

SIEMENS

SINUMERIK

SINUMERIK 840D sl Basisfunktionen

Funktionshandbuch

Gültig für

Steuerung
SINUMERIK 840D sl / 840DE sl

CNC-Software Version 4.92

06/2019

A5E47436268A AA


Vorwort


Grundlegende Sicherheitshinweise	1
K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten	2
K2: Achsen, Koordinatensysteme, Frames	3
K7: Kinematische Kette	4
B1: Bahnsteuerbetrieb, Genauhalt, LookAhead	5
K5: Kanalübergreifende Programmkoordinierung und kanalweises Einfahren	6
K10: Kanalübergreifender Achstausch	7
V2: Vorverarbeitung	8
TE7: Wiederaufsetzen (Retrace Support)	9
N2: Not-Halt	10
A2: Diverse NC/PLC-Nahtstellensignale und Funktionen	11
H2: Hilfsfunktionsausgaben an PLC	12
A4: Digitale und analoge NC-Peripherie	13
B3: Dezentrale Systeme	14
S7: Speicherkonfiguration	15
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/Kataloge
- Anwender-Dokumentation
- Hersteller-/Service-Dokumentation

Weiterführende Informationen

Unter folgender Adresse (<https://support.industry.siemens.com/cs/de/de/view/108464614>) finden Sie Informationen zu den 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 eine E-Mail an folgende Adresse (<mailto:docu.motioncontrol@siemens.com>).

mySupport/Dokumentation

Unter folgender Adresse (<https://support.industry.siemens.com/My/ww/de/documentation>) finden Sie Informationen, wie Sie Ihre Dokumentation auf Basis der Siemensinhalte individuell zusammenstellen und für die eigene Maschinendokumentation anpassen.

Training

Unter folgender Adresse (<http://www.siemens.de/sitrain>) finden Sie Informationen zu SITRAIN - dem Training von Siemens für Produkte, Systeme und Lösungen der Antriebs- und Automatisierungstechnik.

FAQs

Frequently Asked Questions finden Sie in den Service&Support-Seiten unter Produkt Support (<https://support.industry.siemens.com/cs/de/de/ps/faq>).

SINUMERIK

Informationen zu SINUMERIK finden Sie unter folgender Adresse (<http://www.siemens.de/sinumerik>).

Zielgruppe

Die vorliegende Druckschrift wendet sich an:

- Projekteure
- Technologen (von Maschinenherstellern)
- Inbetriebnehmer (von Systemen/Maschinen)
- Programmierer

Nutzen

Das Funktionshandbuch beschreibt die Funktionen, so dass die Zielgruppe die Funktionen kennt und auswählen kann. Es befähigt die Zielgruppe, die Funktionen in Betrieb zu nehmen.

Standardumfang

In der vorliegenden Dokumentation 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 Betriebes und der Instandhaltung berücksichtigen.

Hinweis zur Datenschutzgrundverordnung

Siemens beachtet die Grundsätze des Datenschutzes, insbesondere die Gebote der Datenminimierung (privacy by design). Für dieses Produkt bedeutet dies:

Das Produkt verarbeitet/speichert keine personenbezogenen Daten, lediglich technische Funktionsdaten (z. B. Zeitstempel). Verknüpft der Anwender diese Daten mit anderen Daten (z. B. Schichtpläne) oder speichert er personenbezogene Daten auf dem gleichen Medium (z. B. Festplatte) und stellt so einen Personenbezug her, hat er die Einhaltung der datenschutzrechtlichen Vorgaben selbst sicherzustellen.

Technical Support

Landesspezifische Telefonnummern für technische Beratung finden Sie im Internet unter folgender Adresse (<https://support.industry.siemens.com/sc/ww/de/sc/2090>) im Bereich "Kontakt".

Informationen zu Struktur und Inhalt

Aufbau

Das vorliegende Funktionshandbuch ist wie folgt aufgebaut:

- Innentitel (Seite 3) mit dem Titel des Funktionshandbuchs, den SINUMERIK-Steuerungen sowie der Software und Version, für die diese Ausgabe des Funktionshandbuchs gültig ist, und der Übersicht der einzelnen Funktionsbeschreibungen.
- Funktionsbeschreibungen in der Reihenfolge ihrer alphanumerischen Kurzzeichen (z. B. A2, A3, B1 etc.)
- Anhang mit:
 - Abkürzungsverzeichnis
 - Dokumentationsübersicht
- Begriffsverzeichnis

Hinweis

Ausführliche Daten- und Alarm-Beschreibungen siehe:

- für Maschinen- und Settingdaten:
Ausführliche Maschinendaten-Beschreibung
 - für NC/PLC-Nahtstellensignale:
Listenhandbuch NC-Variablen und Nahtstellensignale
 - für Alarme:
Diagnosehandbuch
-

Schreibweise von Systemdaten

Für Systemdaten gelten in dieser Dokumentation folgende Schreibweisen:

Signal/Datum	Schreibweise	Beispiel
NC/PLC-Nahtstellensignale	... NC/PLC-Nahtstellensignal: <Signaladresse> (<Signalname>)	Ist die neue Getriebestufe eingelegt, dann werden vom PLC-Programm die folgenden NC/PLC-Nahtstellensignale gesetzt: DB31, ... DBX16.0-2 (Istgetriebestufe A bis C) DB31, ... DBX16.3 (Getriebe ist umgeschaltet)
Maschinendatum	... Maschinendatum: <Typ><Nummer> <Vollständiger Bezeichner> (<Bedeutung>)	Masterspindel ist die abgelegte Spindel im Maschinendatum: MD20090 \$MC_SPIND_DEF_MASTER_SPIND (Löschstellung Masterspindel im Kanal)
Settingdatum	... Settingdatum: <Typ><Nummer> <Vollständiger Bezeichner> (<Bedeutung>)	Die logische Masterspindel ist enthalten im Settingdatum: SD42800 \$SC_SPIND_ASSIGN_TAB[0] (Spindelnummernumsetzer)

Mengengerüst

Erläuterungen bezüglich der NC/PLC-Nahtstelle gehen von der absoluten maximalen Anzahl folgender Komponenten aus:

- Betriebsartengruppen (DB11)
- Kanäle (DB21, ...)
- Achsen/Spindeln (DB31, ...)

Datentypen

In der Steuerung stehen zur Programmierung in Teileprogrammen folgenden Datentypen zur Verfügung:

Typ	Bedeutung	Wertebereich
INT	Ganzzahlige Werte mit Vorzeichen	-2.147.483.648 ... +2.147.483.647
REAL	Zahlen mit Dezimalpunkt	$\approx \pm 5,0 \cdot 10^{-324} \dots \approx \pm 1,7 \cdot 10^{308}$
BOOL	Wahrheitswerte	TRUE ($\neq 0$), FALSE (0)
CHAR	Zeichen ASCII bzw. Byte	0 ... 255 bzw. -128 ... 127
STRING	Zeichenkette, Null-terminiert	maximal 400 Zeichen + /0 (keine Sonderzeichen)
AXIS	Achsnamen	alle in der Steuerung vorhandenen Achsnamen
FRAME	geometrische Angaben für Verschieben, Drehen, Skalieren, Spiegeln	---

Felder

Felder können nur aus gleichen elementaren Datentypen gebildet werden. Es sind maximal 3-dimensionale Arrays möglich.

Beispiel: `DEF INT FELD[2, 3, 4]`

Zahlensysteme

Es stehen folgende Zahlensystem zur Verfügung:

- Dezimal: `DEF INT Zahl = 1234` oder `DEF REAL Zahl = 1234.56`
- Hexadezimal: `DEF INT Zahl = 'H123ABC'`
- Binär: `DEF INT Zahl = 'B10001010010'`

Abfrage von REAL-Variablen

Es wird empfohlen, die Abfragen von REAL- bzw. DOUBLE-Variablen in NC-Programmen und Synchronaktionen als Grenzwertbetrachtung zu programmieren.

Beispiel: Abfrage des Istwerts einer Achse auf einen bestimmten Wert

Programmcode	Kommentar
<code>DEF REAL AXPOS = 123.456</code>	
<code>IF (\$VA_IM[<Achse>] - 1ex-6) <= AXPOS <= (\$VA_IM[<Achse>] + 1ex-6)</code>	<code>; Istposition</code>
<code>...</code>	<code>== AXPOS</code>

Programmcode	Kommentar
ELSE ... ENDIF	<> AXPOS

Inhaltsverzeichnis

	Vorwort	3
1	Grundlegende Sicherheitshinweise	25
1.1	Allgemeine Sicherheitshinweise.....	25
1.2	Gewährleistung und Haftung für Applikationsbeispiele.....	25
1.3	Industrial Security.....	26
2	K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten.....	29
2.1	Kurzbeschreibung	29
2.2	Betriebsartengruppe (BAG).....	32
2.2.1	BAG-Stop	34
2.2.2	BAG-Reset	34
2.3	Betriebsarten und Betriebsartenwechsel	35
2.3.1	Überwachungen und Verriegelungen der einzelnen Betriebsarten.....	40
2.3.2	Betriebsartenwechsel.....	40
2.4	Kanal	42
2.4.1	Eigenschaften	42
2.4.2	Startsperrung, global und kanalspezifisch	45
2.5	Programmtest.....	46
2.5.1	Programmbearbeitung im Zustand "Programmtest"	46
2.5.2	Programmbearbeitung im Einzelsatzbetrieb	49
2.5.3	Programmbearbeitung mit Probelaufvorschub.....	51
2.5.4	Teileprogrammsätze ausblenden.....	54
2.6	Werkstücksimulation	55
2.7	Satzsuchlauf Typ 1, 2 und 4.....	56
2.7.1	Funktionsbeschreibung	57
2.7.2	Satzsuchlauf im Zusammenhang mit weiteren NCK-Funktionen.....	60
2.7.2.1	ASUP nach und bei Satzsuchlauf	60
2.7.2.2	PLC-Aktionen nach Satzsuchlauf.....	61
2.7.2.3	Spindelfunktionen nach Satzsuchlauf	61
2.7.2.4	Lesen von Systemvariablen bei Satzsuchlauf.....	62
2.7.3	Automatischer Start eines ASUP nach Satzsuchlauf.....	63
2.7.4	Kaskadierter Satzsuchlauf	65
2.7.5	Beispiele zum Satzsuchlauf mit Berechnung	66
2.7.6	Randbedingungen.....	70
2.7.6.1	Kompressor-Funktionen (COMPON, COMPCURV, COMPCAD, COMPSURF)	70
2.8	Satzsuchlauf Typ 5 (SERUPRO)	70
2.8.1	Funktionsbeschreibung	70
2.8.2	Wiederanfahren an die Kontur (REPOS)	74
2.8.2.1	Wiederanfahren an die Kontur mit gesteuertem REPOS.....	83
2.8.3	Suchlaufvorgang beschleunigen	84
2.8.4	SERUPRO-ASUP	86

2.8.5	Self-Acting SERUPRO	89
2.8.6	Programmabschnitt für Wiederaufsetzen sperren.....	90
2.8.7	Verhalten bei Power On, Betriebsartenwechsel und RESET	94
2.8.8	Randbedingungen	94
2.8.8.1	STOPRE im Zielsatz	94
2.8.8.2	SPOS im Zielsatz	95
2.8.8.3	Fahren auf Festanschlag (FXS)	95
2.8.8.4	Fahren mit begrenztem Moment/Kraft (FOC)	96
2.8.8.5	Synchronspindel.....	97
2.8.8.6	Kopplungen und Master-Slave	97
2.8.8.7	Achsfunktionen.....	100
2.8.8.8	Getriebestufenwechsel.....	102
2.8.8.9	Überlagerte Bewegung	102
2.8.8.10	NC/PLC-Nahtstellensignale	103
2.8.8.11	Flexibilisierung der Grundeinstellung	103
2.8.8.12	Kompressor-Funktionen (COMPON, COMPCURV, COMPCAD, COMPSURF)	104
2.8.9	Systemvariable.....	104
2.9	Programmbetrieb	104
2.9.1	Programmbetrieb	104
2.9.2	Grundstellungen	105
2.9.2.1	Maschinendaten	105
2.9.2.2	Programmierung	106
2.9.3	Anwahl und Start eines NC-Programms	110
2.9.4	Programmunterbrechung	112
2.9.5	Kanal-Reset	113
2.9.6	Programmzustand	114
2.9.7	Kanalzustand	115
2.9.8	Reaktionen auf Bedienungs- und Programmaktionen	116
2.9.9	Zeitdiagramm-Beispiel für einen Programmablauf.....	117
2.9.10	Programmsprünge	118
2.9.10.1	Rücksprung auf Programmanfang (GOTOS).....	118
2.9.11	Programmteilwiederholungen	120
2.9.11.1	Programmierung	120
2.9.12	Ereignisgesteuerter Programmaufruf (PROG_EVENT)	126
2.9.12.1	Funktion	126
2.9.12.2	Parametrierung	130
2.9.12.3	Programmierung	134
2.9.12.4	Randbedingungen	135
2.9.12.5	Beispiele.....	136
2.9.13	Beeinflussung von Stopp-Ereignissen durch Stop-Delay-Bereiche	137
2.9.13.1	Funktion	137
2.9.13.2	Parametrierung	141
2.9.13.3	Programmierung	141
2.9.13.4	Randbedingungen	143
2.9.14	Größenanpassung des Interpolationspuffers	144
2.9.15	Basis-Satzanzeige (nur bei ShopMill/ShopTurn)	146
2.9.15.1	Funktion	146
2.9.15.2	Parametrierung	146
2.9.15.3	Aufbau für einen DIN-Satz	149
2.10	Asynchrone Unterprogramme (ASUPs).....	152
2.10.1	Funktion	152

2.10.1.1	Ablauf eines ASUPs im Programmbetrieb	155
2.10.1.2	ASUP mit REPOSA.....	156
2.10.1.3	NC-Verhalten	157
2.10.2	Inbetriebnahme: Maschinendaten	158
2.10.2.1	NC-spez.: BAG-spezifische NC/PLC-Nahtstellensignale und Betriebsartenumschaltung....	158
2.10.2.2	NC-spez.: ASUP Startfreigabe.....	159
2.10.2.3	NC-spez.: Wirksamkeit der parametrisierten Startfreigaben	160
2.10.2.4	Kanalspez.: Startfreigabe trotz nicht referenzierter Achsen.....	160
2.10.2.5	Kanalspez.: Startfreigabe trotz Einleesperre	160
2.10.2.6	Kanalspez.: Kontinuierliche Abarbeitung trotz Einzelsatz	161
2.10.2.7	Kanalspez.: Aktualisierung der Anzeige.....	161
2.10.3	Programmierung: Systemvariablen.....	162
2.10.3.1	REPOS-Möglichkeit (\$P_REPINF).....	162
2.10.3.2	Aktivierungsereignis (\$AC_ASUP)	163
2.10.4	Programmierung (SETINT, PRIO)	163
2.10.5	Randbedingungen	165
2.10.6	Beispiele.....	166
2.11	Anwenderspezifisches ASUP für RET und REPOS.....	167
2.11.1	Funktion	167
2.11.2	Parametrierung	167
2.11.3	Programmierung	168
2.12	Einzelsatz.....	169
2.12.1	Funktion	169
2.12.2	Parametrierung	170
2.12.3	Programmierung	172
2.12.3.1	Einzelsatzbearbeitung aus/einschalten (SBLOF, SBLON)	172
2.12.3.2	Randbedingungen.....	173
2.12.3.3	Beispiele.....	173
2.12.4	BAG-spezifischer Einzelsatz Typ A / B	176
2.12.5	Randbedingungen.....	177
2.12.5.1	Einzelsatztyp SB2 und satzbezogene Synchronaktionen	177
2.12.5.2	Programmierter Halt (M0), Einzelsatz und Einzelsatztypumschaltung	177
2.13	Programmbeeinflussung	178
2.13.1	Aktivierung	178
2.13.1.1	Anwahl über Bedienoberfläche	178
2.13.1.2	Anwahl über PLC-Anwenderprogramm	178
2.13.1.3	Hinweis zur Auswahl über PLC-Anwenderprogramm.....	178
2.13.1.4	Relevante Nahtstellensignale.....	179
2.13.1.5	Status	180
2.13.2	Programmsätze ausblenden	180
2.14	ASUP-Start bei Anwenderalarmen durchführen	182
2.14.1	Funktion	182
2.14.2	Aktivierung	183
2.14.3	Beispiele.....	184
2.14.3.1	Anwender-ASUP aus Reset - Beispiel 1	184
2.14.3.2	Anwender-ASUP aus Reset - Beispiel 2	185
2.14.3.3	Anwender-ASUP mit M0	186
2.14.3.4	Anwender-ASUP mit Stopp.....	187
2.14.3.5	Anwender-ASUP aus gestoppt	188
2.15	Abarbeiten von Extern.....	189

2.16	Abarbeiten von externen Unterprogrammen (EXTCALL)	191
2.17	Abarbeiten vom externen Speicher (EES) (Option)	194
2.17.1	Funktion	194
2.17.2	Inbetriebnahme	196
2.17.2.1	Projektierung der Laufwerke	196
2.17.2.2	Globaler Teileprogrammspeicher (GDIR)	198
2.17.2.3	Einstellungen für Dateihandling im Teileprogramm bei EES	200
2.17.2.4	Speicherkonfiguration	201
2.17.3	Randbedingungen	202
2.18	Process DataShare - Ausgabe auf ein externes Gerät/Datei.....	202
2.18.1	Funktion	202
2.18.2	Inbetriebnahme	204
2.18.3	Programmierung	206
2.18.4	Randbedingungen	210
2.19	Systemeinstellungen für Hochlauf, RESET/Teileprogrammende und Teileprogramm- Start.....	211
2.19.1	Systemeinstellungen für Hochlauf, RESET/Teileprogramm-Ende und Teileprogramm- Start.....	211
2.19.2	Werkzeugrückzug nach POWER ON mit Orientierungstransformation	215
2.20	Ersetzung von Funktionen durch Unterprogramme	218
2.20.1	Übersicht	218
2.20.2	Ersetzung von M-, T/TCA- und D/DL-Funktionen	219
2.20.2.1	Ersetzung von M-Funktionen	219
2.20.2.2	Ersetzung von T/TCA- und D/DL-Funktionen	221
2.20.2.3	Systemvariable.....	224
2.20.2.4	Beispiel: Ersetzung einer M-Funktion	225
2.20.2.5	Beispiel: Ersetzung einer T- und D-Funktion	228
2.20.2.6	Verhalten im Konfliktfall.....	229
2.20.3	Ersetzung von Spindel-Funktionen	230
2.20.3.1	Allgemeine Informationen	230
2.20.3.2	Ersetzung von M40 - M45 (Getriebestufenwechsel).....	232
2.20.3.3	Ersetzung von SPOS, SPOSA, M19 (Spindelpositionieren)	232
2.20.3.4	Systemvariable.....	233
2.20.3.5	Beispiel: Getriebestufenwechsel.....	234
2.20.3.6	Beispiel: Spindelpositionieren	236
2.20.4	Eigenschaften der Unterprogramme	239
2.20.5	Randbedingungen	241
2.21	Umbenennung/Sperrung von NC-Befehlen	241
2.22	Programmlaufzeit / Werkstückzähler	242
2.22.1	Programmlaufzeit	243
2.22.1.1	Funktion	243
2.22.1.2	Inbetriebnahme	246
2.22.1.3	Randbedingungen	248
2.22.1.4	Beispiele.....	248
2.22.2	Werkstückzähler.....	249
2.22.2.1	Funktion	249
2.22.2.2	Inbetriebnahme	250
2.22.2.3	Randbedingungen	252
2.22.2.4	Beispiele.....	252

2.23	Datenlisten	254
2.23.1	Funktion	254
2.23.2	Maschinendaten	255
2.23.2.1	Allgemeine Maschinendaten	255
2.23.2.2	Kanal-spezifische Maschinendaten	256
2.23.2.3	Achs-/Spindel-spezifische Maschinendaten	259
2.23.3	Settingdaten	260
2.23.3.1	Kanal-spezifische Settingdaten	260
3	K2: Achsen, Koordinatensysteme, Frames	261
3.1	Kurzbeschreibung	261
3.1.1	Achsen	261
3.1.2	Koordinatensysteme	263
3.1.3	Frames	264
3.2	Achsen	268
3.2.1	Übersicht	268
3.2.2	Maschinenachsen	269
3.2.3	Kanalachsen	270
3.2.4	Geometrieachsen	270
3.2.5	Zusatzachsen	270
3.2.6	Bahnachsen	271
3.2.7	Positionierachsen	271
3.2.8	Hauptlaufachsen	272
3.2.9	Synchronachsen	273
3.2.10	Achskonfiguration	275
3.2.11	Link-Achsen	277
3.3	Nullpunkte und Referenzpunkte	278
3.3.1	Bezugspunkte im Arbeitsraum	278
3.3.2	Lage der Koordinatensysteme und Referenzpunkte	280
3.4	Koordinatensysteme	281
3.4.1	Übersicht	281
3.4.2	Maschinenkoordinatensystem (MKS)	284
3.4.2.1	Istwertsetzen mit Verlust des Referenzierstatus (PRESETON)	285
3.4.2.2	Istwertsetzen ohne Verlust des Referenzierstatus (PRESETONS)	290
3.4.3	Basiskoordinatensystem (BKS)	295
3.4.4	Basis-Nullpunktsystem (BNS)	297
3.4.5	Einstellbares Nullpunktsystem (ENS)	299
3.4.6	Werkstückkoordinatensystem (WKS)	300
3.4.7	Additive Korrekturen	301
3.4.7.1	Externe Nullpunktverschiebungen	301
3.4.7.2	DRF-Verschiebung	303
3.4.7.3	Reset-Verhalten	303
3.4.8	Achsspezifische Überlagerung (\$AA_OFF)	304
3.4.8.1	Funktion	304
3.4.8.2	Inbetriebnahme	304
3.4.8.3	Programmierung: Überlagerungen achsspezifisch abwählen (CORROF)	304
3.5	Frames	307
3.5.1	Frame-Arten	307
3.5.2	Frame-Komponenten	309
3.5.2.1	Translation	309

3.5.2.2	Feinverschiebung	309
3.5.2.3	Drehung: Übersicht (nur Geometrieachsen)	311
3.5.2.4	Drehung mit Euler-Winkeln: ZY'X"-Konvention (RPY-Winkel)	312
3.5.2.5	Drehung mit Euler-Winkeln: ZX'Z"-Konvention	317
3.5.2.6	Drehung in beliebiger Ebene	318
3.5.2.7	Skalierung	319
3.5.2.8	Spiegelung	320
3.5.2.9	Verkettungsoperator	320
3.5.2.10	Programmierbare Achsname	320
3.5.2.11	Koordinatentransformation	322
3.5.3	Datenhaltungs-Frames und aktive Frames	322
3.5.3.1	Übersicht	322
3.5.3.2	Aktivierung von Datenhaltungsframes	324
3.5.3.3	NCU-lobale und kanalspezifische Frames	325
3.5.4	Framekette und Koordinatensysteme	326
3.5.4.1	Übersicht	326
3.5.4.2	Relative Koordinatensysteme	327
3.5.4.3	Wählbares ENS	329
3.5.4.4	Manuelle Verfahren von Geometrieachsen wahlweise im WKS oder ENS (\$AC_JOG_COORD)	330
3.5.4.5	Unterdrückung von Frames	331
3.5.5	Frames der Framekette	332
3.5.5.1	Übersicht	332
3.5.5.2	Einstellbare Frames \$P_UIFR[<n>]	333
3.5.5.3	Schleifframes \$P_GFR[<n>]	335
3.5.5.4	Kanal-spezifische Basisframes[<n>]	339
3.5.5.5	NCU-globale Basisframes \$P_NCBFR[<n>]	340
3.5.5.6	Aktiver Gesamt-Basisframe \$P_ACTBFRAME	341
3.5.5.7	Programmierbarer Frame \$P_PFRAME	343
3.5.5.8	Kanalspezifische Systemframes	345
3.5.6	Implizite Frame-Änderungen	347
3.5.6.1	Umschalten von Geometrieachsen	347
3.5.6.2	An- und Abwahl von Transformationen: Allgemein	350
3.5.6.3	An- und Abwahl von Transformationen: TRANSMIT	351
3.5.6.4	An- und Abwahl von Transformationen: TRACYL	357
3.5.6.5	An- und Abwahl von Transformationen: TRAANG	363
3.5.6.6	Adaptionen von aktiven Frames	368
3.5.6.7	Mapped Frames	369
3.5.6.8	Mapped Frames	373
3.5.7	Vordefinierte Frame-Funktionen	376
3.5.7.1	Inverses Frame	376
3.5.7.2	Additives Frame in der Framekette	379
3.5.8	Funktionen	380
3.5.8.1	Setzen von Nullpunkten, Werkstück- und Werkzeugvermessung	380
3.5.8.2	Achsspezifische Externe Nullpunktverschiebung	381
3.5.8.3	Werkzeugträger	381
3.5.9	Unterprogramme mit Attribut SAVE	391
3.5.10	Datensicherung	392
3.5.11	Positionen in den Koordinatensystemen	393
3.5.12	Steuerungsverhalten	393
3.5.12.1	POWER ON	393
3.5.12.2	Betriebsartenwechsel	394

3.5.12.3	Kanal-Reset / Teileprogrammende	394
3.5.12.4	Teileprogrammstart	398
3.5.12.5	Satzsuchlauf.....	398
3.5.12.6	REPOS.....	399
3.6	Werkstücknahes Istwertsystem.....	399
3.6.1	Übersicht	399
3.6.2	Benutzung des werkstücknahen Istwertsystems	399
3.6.3	Besondere Reaktionen.....	402
3.7	Randbedingungen	404
3.8	Beispiele.....	404
3.8.1	Achsen	404
3.8.2	Koordinatensysteme	406
3.8.3	Frames	408
3.9	Datenlisten	410
3.9.1	Maschinendaten	410
3.9.1.1	Anzeige-Maschinendaten.....	410
3.9.1.2	NC-spezifischen Maschinendaten.....	410
3.9.1.3	Kanal-spezifische Maschinendaten.....	411
3.9.1.4	Achs-/Spindel-spezifische Maschinendaten.....	412
3.9.2	Settingdaten	412
3.9.2.1	Kanal-spezifische Settingdaten.....	412
3.9.3	Systemvariablen.....	412
4	K7: Kinematische Kette	415
4.1	Funktionsbeschreibung	415
4.1.1	Merkmale	415
4.2	Inbetriebnahme	419
4.2.1	Allgemein	419
4.2.1.1	Übersicht	419
4.2.1.2	Aufbau der Systemvariablen	420
4.2.2	Maschinendaten	421
4.2.2.1	Maximale Anzahl Elemente.....	421
4.2.2.2	Root-Element	421
4.2.2.3	Maximale Anzahl Schalter.....	421
4.2.3	Systemvariablen.....	422
4.2.3.1	Übersicht	422
4.2.3.2	\$NK_NAME	423
4.2.3.3	\$NK_NEXT	424
4.2.3.4	\$NK_PARALLEL	425
4.2.3.5	\$NK_TYPE	426
4.2.3.6	Typabhängige Variablen bei \$NK_TYPE = "AXIS_LIN"	427
4.2.3.7	Typabhängige Variablen bei \$NK_TYPE = "AXIS_ROT"	430
4.2.3.8	Typabhängige Variablen bei \$NK_TYPE = "ROT_CONST"	433
4.2.3.9	Typabhängige Variablen bei \$NK_TYPE = "OFFSET"	435
4.2.3.10	Typabhängige Variablen bei \$NK_TYPE = "SWITCH"	437
4.2.3.11	\$NK_SWITCH	439
4.3	Programmierung	440
4.3.1	Löschen von Komponenten (DELOBJ)	440
4.3.2	Indexermittlung per Namen (NAMETOINT)	443

4.4	Beispiel.....	444
4.4.1	Vorgaben.....	444
4.4.2	Teileprogramm des Maschinenmodells	446
4.5	Datenlisten	449
4.5.1	Maschinendaten.....	449
4.5.1.1	NC-spezifische Maschinendaten.....	449
4.5.2	Systemvariablen.....	449
5	B1: Bahnsteuerbetrieb, Genauhalt, LookAhead	451
5.1	Kurzbeschreibung	451
5.2	Genauhaltbetrieb.....	454
5.3	Bahnsteuerbetrieb.....	458
5.3.1	Allgemeine Funktionalität	458
5.3.2	Geschwindigkeitsabsenkung gemäß Überlastfaktor	459
5.3.3	Überschleifen	461
5.3.3.1	Überschleifen nach Wegkriterium (G641)	464
5.3.3.2	Überschleifen unter Einhaltung definierter Toleranzen (G642/G643).....	466
5.3.3.3	Überschleifen mit maximal möglicher Achsdynamik (G644).....	469
5.3.3.4	Überschleifen tangentialer Satzübergänge (G645).....	472
5.3.3.5	Überschleifen und Repositionieren (REPOS)	473
5.3.4	LookAhead	474
5.3.4.1	Standardfunktionalität	474
5.3.4.2	Freiformflächenmodus: Erweiterungsfunktion	480
5.4	Dynamikanpassungen.....	484
5.4.1	Glättung der Bahngeschwindigkeit.....	484
5.4.2	Anpassung der Bahndynamik	488
5.4.3	Ermittlung der Dynamikgrenzwerte	492
5.4.4	Zusammenwirken der Funktionen "Glättung der Bahngeschwindigkeit" und "Anpassung der Bahndynamik"	493
5.4.5	Dynamikmodus für Bahninterpolation	496
5.4.6	Freiformflächenmodus: Grundfunktionen.....	499
5.5	Kompressor-Funktionen.....	503
5.5.1	Kompression von Linear-, Kreis- und Eilgangsätzen	503
5.5.1.1	Funktion	503
5.5.1.2	Inbetriebnahme	505
5.5.1.3	Programmierung	507
5.5.1.4	Randbedingungen.....	508
5.5.2	Kompression kurzer Spline-Sätze.....	508
5.6	Kontur-/Orientierungstoleranz	510
5.6.1	Inbetriebnahme	510
5.6.1.1	Parametrierung	510
5.6.2	Programmierung	511
5.6.2.1	Kontur-/Orientierungstoleranz programmieren (CTOL, OTOL, ATOL)	511
5.6.2.2	Kontur-/Orientierungstoleranz programmieren (CTOL, OTOL, ATOL): Weitere Informationen	514
5.7	Eilgangbewegungen.....	515
5.7.1	Funktion	515
5.7.1.1	Eilgang	515

5.7.1.2	Interpolationsverhalten der Bahnachsen bei Eilgangbewegungen	516
5.7.1.3	Toleranzen bei Eilgangbewegungen	518
5.7.1.4	Eilgangkorrektur	519
5.7.2	Inbetriebnahme	519
5.7.2.1	Parametrierung	519
5.7.3	Programmierung	520
5.7.3.1	Eilgang aktivieren (G0).....	520
5.7.3.2	Lineare Interpolation für Eilgangbewegungen ein-/ausschalten (RTLION, RTLIOF)	522
5.7.3.3	Relative G0-Toleranz anpassen (STOLF).....	523
5.8	RESET-Verhalten.....	526
5.9	Randbedingungen.....	526
5.9.1	Satzwechsel und Positionierachsen.....	526
5.9.2	Satzwechselverzögerung	527
5.10	Datenlisten	527
5.10.1	Maschinendaten	527
5.10.1.1	Allgemeine Maschinendaten	527
5.10.1.2	Kanal-spezifische Maschinendaten.....	527
5.10.1.3	Achs-/Spindel-spezifische Maschinendaten.....	529
5.10.2	Settingdaten	529
5.10.2.1	Kanal-spezifische Settingdaten.....	529
6	K5: Kanalübergreifende Programmkoordination und kanalweises Einfahren	531
6.1	Kanalübergreifende Programmkoordination	531
6.1.1	Kanalübergreifende Programmkoordination (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)	531
6.1.2	Bedingtes Warten (WAITMC) im Bahnsteuerbetrieb	537
6.2	Kanalweises Einfahren.....	541
6.2.1	Funktion	541
6.2.2	Ablauf	542
6.2.3	Einkanalansicht	542
6.2.4	Mehrkanalansicht	544
6.2.5	Systemvariablen.....	547
6.2.6	Randbedingungen.....	547
6.2.7	Beispiele.....	548
6.3	Randbedingungen.....	549
6.3.1	Betriebsart MDA: Bahnsteuerbetrieb und WAITMC.....	549
6.3.2	Nicht synchroner Start der Verfahrbewegung nach WAIT-Befehlen.....	550
7	K10: Kanalübergreifender Achstausch	553
7.1	Überblick	553
7.2	Inbetriebnahme	553
7.3	Programmierung: Achse freigeben (RELEASE)	554
7.4	Programmierung: Achse holen (GET, GETD).....	555
7.5	Automatischer Achstausch.....	557
7.6	Achstausch durch PLC.....	558
7.7	Achstausch über Achscontainer-Drehung.....	561

7.8	Achstausch mit und ohne Vorlaufstopp.....	562
7.9	Achse ausschließlich PLC-kontrolliert.....	563
7.10	Achse fest der PLC zugeordnet	564
7.11	Geometrieachse im gedrehten WKS und Achstausch	565
7.12	Achstausch aus Synchronaktionen	567
7.13	Achstausch bei Führungsachsen (Gantry).....	569
7.14	Zustandsdiagramm	569
7.15	Randbedingungen	570
7.15.1	Achstausch von PLC.....	571
7.15.2	Satzsuchlauf mit Berechnung	571
7.16	Beispiel.....	571
7.17	Datenlisten	573
7.17.1	Maschinendaten.....	573
7.17.1.1	Allgemeine Maschinendaten	573
7.17.1.2	Kanal-spezifische Maschinendaten.....	573
7.17.1.3	Achs-/Spindel-spezifische Maschinendaten.....	575
7.17.2	Settingdaten	576
7.17.2.1	Kanal-spezifische Settingdaten.....	576
8	V2: Vorverarbeitung.....	577
8.1	Kurzbeschreibung	577
8.2	Programmhandling.....	579
8.3	Programmaufruf	582
8.4	Randbedingungen	584
8.5	Beispiele.....	585
8.5.1	Vorverarbeitung einzelner Dateien.....	585
8.5.2	Vorverarbeitung im dynamischen NC-Speicher	587
8.6	Datenlisten	587
8.6.1	Maschinendaten.....	587
8.6.1.1	Allgemeine Maschinendaten	587
8.6.1.2	Kanal-spezifische Maschinendaten.....	588
9	TE7: Wiederaufsetzen (Retrace Support)	589
9.1	Kurzbeschreibung	589
9.2	Funktionsbeschreibung.....	590
9.2.1	Funktion	590
9.2.2	Begriffsdefinitionen.....	592
9.2.3	Funktionsablauf (Prinzip)	593
9.2.4	Maximaler RESU-fähiger Konturbereich	596
9.3	Inbetriebnahme	597
9.3.1	Aktivierung	597
9.3.2	Festlegung der RESU-Arbeitsebene.....	597
9.3.3	Speicherkonfiguration: Satzspeicher.....	598
9.3.4	Speicherkonfiguration: Heap-Speicher	599

9.3.5	Speicherbereich des RESU-Hauptprogramms	600
9.3.6	Ablage der RESU-Unterprogramme	601
9.3.7	ASUP-Freigabe	602
9.3.8	PLC-Anwenderprogramm	602
9.4	Programmierung	603
9.4.1	RESU-Start/Stop/Reset (CC_PREPRE)	603
9.5	RESU-spezifische Teileprogramme	604
9.5.1	Übersicht	604
9.5.2	Hauptprogramm (CC_RESU.MPF)	605
9.5.3	INI-Programm (CC_RESU_INI.SPF)	606
9.5.4	END-Programm (CC_RESU_END.SPF)	608
9.5.5	Wiederaufsetz-ASUP (CC_RESU_BS_ASUP.SPF)	609
9.5.6	RESU-ASUP (CC_RESU_ASUP.SPF)	609
9.6	Wiederaufsetzen	610
9.6.1	Allgemeine Informationen	610
9.6.2	Satzsuchlauf mit Berechnung an der Kontur	611
9.6.3	Repositionieren	611
9.6.4	Zeitliche Bedingungen bezüglich NC-Start	612
9.6.5	Satzsuchlauf ab letztem Hauptsatz	613
9.7	Funktionsspezifische Anzeigedaten	614
9.7.1	Kanalspezifische GUD-Variable	614
9.8	Funktionsspezifische Alarmtexte	615
9.9	Randbedingungen	615
9.9.1	Funktionsspezifische Randbedingungen	615
9.9.1.1	Wiederaufsetzen innerhalb von Unterprogrammen	615
9.9.1.2	Wiederaufsetzen innerhalb von Programmschleifen	616
9.9.1.3	Wiederaufsetzen an Vollkreisen	616
9.9.1.4	Automatisch generierte Konturelemente	616
9.9.2	Randbedingungen bezüglich Standardfunktionen	617
9.9.2.1	Achstausch	617
9.9.2.2	Verfahrbewegungen von Kanalachsen	617
9.9.2.3	Satznummern	617
9.9.2.4	Satzsuchlauf	618
9.9.2.5	Transformationen	618
9.9.2.6	Kompensationen	618
9.9.2.7	Frames	619
9.9.2.8	Werkzeugkorrekturen	619
9.10	Datenlisten	620
9.10.1	Maschinendaten	620
9.10.1.1	Allgemeine Maschinendaten	620
9.10.1.2	Kanal-spezifische Maschinendaten	620
10	N2: Not-Halt	621
10.1	Kurzbeschreibung	621
10.2	Normen	621
10.3	Not-Halt-Stellteile	622
10.4	Not-Halt-Ablauf	623

10.5	Not-Halt-Quittierung	624
10.6	Datenlisten	626
10.6.1	Maschinendaten	626
10.6.1.1	Achs-/Spindel-spezifische Maschinendaten.....	626
11	A2: Diverse NC/PLC-Nahtstellensignale und Funktionen	627
11.1	Kurzbeschreibung	627
11.2	Funktionen	627
11.2.1	Verhalten der Bildaktualisierung bei Überlastung	627
11.2.2	DEFAULT-Speicher aktivieren	628
11.2.3	PLC-Variable lesen und schreiben.....	628
11.2.4	Zugriffsschutz über Kennwort und Schlüsselschalter	631
11.2.4.1	Kennwort	633
11.2.4.2	Schlüsselschalter-Stellungen (DB10, DBX56.4 bis 7).....	635
11.2.4.3	Parametrierbare Schutzstufen	636
11.2.5	Umschalten von Motor- / Antriebsdatensätzen	636
11.2.5.1	Allgemeine Informationen	636
11.2.5.2	Formatierungsschnittstellen	637
11.2.5.3	Anforderungsschnittstelle	638
11.2.5.4	Anzeigeschnittstelle	638
11.2.5.5	Beispiel.....	639
11.2.5.6	Übersicht der Schnittstellen	640
11.2.5.7	Randbedingungen	641
11.3	Beispiele.....	642
11.3.1	Parametersatzumschaltung	642
11.4	Datenlisten	643
11.4.1	Maschinendaten	643
11.4.1.1	Anzeige-Maschinendaten.....	643
11.4.1.2	NC-spezifischen Maschinendaten.....	643
11.4.1.3	Kanal-spezifische Maschinendaten.....	644
11.4.1.4	Achs-/Spindel-spezifische Maschinendaten.....	644
11.4.2	Systemvariablen.....	645
12	H2: Hilfsfunktionsausgaben an PLC.....	647
12.1	Kurzbeschreibung	647
12.1.1	Funktion	647
12.1.2	Definition einer Hilfsfunktion.....	648
12.1.3	Übersicht der Hilfsfunktionen	649
12.2	Vordefinierte Hilfsfunktionen	654
12.2.1	Übersicht: vordefinierten Hilfsfunktionen.....	655
12.2.2	Übersicht: Ausgabeverhalten	665
12.2.3	Parametrierung	669
12.2.3.1	Gruppenzuordnung	669
12.2.3.2	Typ, Adresserweiterung und Wert.....	669
12.2.3.3	Ausgabeverhalten	670
12.3	Anwenderdefinierte Hilfsfunktionen.....	674
12.3.1	Parametrierung	676
12.3.1.1	Maximale Anzahl anwenderdefinierter Hilfsfunktionen	676
12.3.1.2	Gruppenzuordnung	676

12.3.1.3	Typ, Adresserweiterung und Wert.....	676
12.3.1.4	Ausgabeverhalten	678
12.4	Assoziierte Hilfsfunktionen	678
12.5	Typ-spezifisches Ausgabeverhalten	680
12.6	Prioritäten des parametrisierten Ausgabeverhaltens	682
12.7	Programmierung einer Hilfsfunktion	683
12.8	Programmierbare Ausgabedauer	684
12.9	Hilfsfunktionsausgabe an die PLC	686
12.10	Hilfsfunktionen ohne Satzwechselferzögerung	686
12.11	M-Funktion mit implizitem Vorlaufstopp	687
12.12	Verhalten bei Überspeichern.....	688
12.13	Verhalten bei Satzsuchlauf	689
12.13.1	Hilfsfunktionsausgabe bei Satzsuchlauf Typ 1,2 und 4.....	689
12.13.2	Zuordnung einer Hilfsfunktion zu mehreren Gruppen	690
12.13.3	Zeitstempel der aktiven M-Hilfsfunktion	692
12.13.4	Ermittlung der Ausgabe-Reihenfolge	692
12.13.5	Ausgabeunterdrückung von Spindel-spezifischen Hilfsfunktionen.....	694
12.13.6	Hilfsfunktionsausgabe bei Satzsuchlauf Typ 5 (SERUPRO)	697
12.13.7	SERUPRO-Ende-ASUP	702
12.14	Implizit ausgegebene Hilfsfunktionen.....	708
12.15	Informationsmöglichkeiten	709
12.15.1	Gruppenspezifische modale M-Hilfsfunktionsanzeige	709
12.15.2	Abfrage von Systemvariablen	710
12.16	Randbedingungen.....	712
12.16.1	Allgemeine Randbedingungen	712
12.16.2	Ausgabeverhalten	713
12.17	Beispiele.....	715
12.17.1	Erweiterung von vordefinierten Hilfsfunktionen.....	715
12.17.2	Definition von Hilfsfunktionen.....	717
12.18	Datenlisten	721
12.18.1	Maschinendaten	721
12.18.1.1	NC-spezifische Maschinendaten.....	721
12.18.1.2	Kanal-spezifische Maschinendaten.....	721
13	A4: Digitale und analoge NC-Peripherie.....	723
13.1	Einleitung	723
13.2	Indirekte Peripherie-Zugriffe über PLC	724
13.2.1	Kurzbeschreibung	724
13.2.2	Parametrierung	725
13.2.3	Systemvariablen.....	728
13.2.4	Komparator-Eingänge	728
13.2.5	Digitale NC-Ein-/Ausgänge	728
13.2.5.1	Digitale NC-Eingänge.....	728
13.2.5.2	Digitale NC-Ausgänge.....	730


13.2.5.3	Durchschalten und Verknüpfen von schnellen digitalen Ein-/Ausgängen.....	733
13.2.6	Analoge NC-Ein-/Ausgänge	735
13.2.6.1	Analoge NC-Eingänge	735
13.2.6.2	Analoge NC-Ausgänge	737
13.2.6.3	Darstellung der analogen Ein-/Ausgabewerte.....	740
13.2.7	Komparator-Eingänge	742
13.3	Direkte Peripherie-Zugriffe über PLC	746
13.3.1	Parametrierung	746
13.3.2	Lesen / Schreiben: Systemvariablen.....	748
13.3.3	Randbedingungen	749
13.3.4	Beispiele.....	749
13.3.4.1	Schreiben auf PLC-Peripherie	749
13.3.4.2	Lesen von PLC-Peripherie	751
13.4	Direkte Peripherie-Zugriffe ohne PLC	752
13.4.1	Kurzbeschreibung	752
13.4.2	Parametrierung	753
13.4.3	Lesen / Schreiben	755
13.4.3.1	Systemvariablen.....	755
13.4.4	Randbedingungen	757
13.4.5	Beispiele.....	757
13.4.5.1	Schreiben auf NC-Peripherie	757
13.4.5.2	Lesen von NC-Peripherie	759
13.4.5.3	Schreiben von NC-Peripherie mit Statusabfrage	761
13.5	Datenlisten	762
13.5.1	Maschinendaten	762
13.5.1.1	Allgemeine Maschinendaten	762
13.5.1.2	Kanal-spezifische Maschinendaten.....	763
13.5.2	Settingdaten	763
13.5.2.1	Allgemeine Settingdaten	763
13.5.3	Systemvariable.....	763
13.5.4	Signale	764
13.5.4.1	Signale an NC.....	764
13.5.4.2	Signale von NC	764
14	B3: Dezentrale Systeme.....	765
14.1	Kurzbeschreibung	765
14.1.1	Mehrere Bedientafeln an mehreren NCU (T:M:N)	765
14.1.2	NCU-Link.....	768
14.1.2.1	Link-Kommunikation.....	768
14.1.2.2	Link-Variablen	769
14.1.2.3	Link-Achsen	769
14.1.2.4	Lead-Link-Achsen	769
14.1.2.5	Abhängigkeiten	770
14.1.2.6	Anwendungsbeispiel: Rundtaktmaschine	771
14.2	NCU-Link.....	773
14.2.1	Link-Kommunikation.....	773
14.2.1.1	Allgemeine Informationen	773
14.2.1.2	Link-Modul.....	777
14.2.1.3	Parametrierung: NC-Systemtakte	777
14.2.1.4	Parametrierung: Link-Kommunikation	780


14.2.1.5	Projektierung	780
14.2.1.6	Verkabelung der NCUs	780
14.2.1.7	Aktivierung	781
14.2.2	Link-Variablen	781
14.2.2.1	Eigenschaften des Link-Variablen-Speichers	782
14.2.2.2	Eigenschaften der Link-Variablen	783
14.2.2.3	Schreibelemente	784
14.2.2.4	Zeitverhalten beim Schreiben	784
14.2.2.5	Systemvariable.....	785
14.2.2.6	Synchronisation eines Schreibauftrags.....	785
14.2.2.7	Beispiel: Aufteilung des Link-Variablen-Speichers.....	786
14.2.2.8	Beispiel: Lesen von Antriebsdaten.....	787
14.2.3	Link-Achsen	788
14.2.3.1	Allgemeine Informationen	788
14.2.3.2	Name einer Link-Achse.....	790
14.2.3.3	Parametrierung	790
14.2.3.4	Hilfsfunktionsausgabe bei Spindeln	791
14.2.3.5	Randbedingungen.....	792
14.2.4	Achscontainer	794
14.2.4.1	Allgemeine Informationen	794
14.2.4.2	Parametrierung	797
14.2.4.3	Programmierung	803
14.2.4.4	Systemvariable.....	805
14.2.4.5	Bearbeitung mit Achscontainer (schematisch).....	806
14.2.4.6	Verhalten in verschiedenen Betriebszuständen.....	806
14.2.4.7	Verhalten bei Rücknahme der Freigabe zur Achscontainer-Drehung.....	807
14.2.4.8	Randbedingungen.....	809
14.2.5	Lead-Link-Achsen	811
14.2.5.1	Allgemeine Informationen	811
14.2.5.2	Parametrierung	813
14.2.5.3	Systemvariablen zur Leitwertvorgabe	814
14.2.5.4	Randbedingungen.....	814
14.2.5.5	Beispiel.....	815
14.2.6	Maßsysteme innerhalb eines Link-Verbunds	815
14.3	Beispiele.....	816
14.3.1	Link-Achse	816
14.3.2	Achscontainer-Koordinierung.....	817
14.3.2.1	Achscontainer-Drehung ohne Warten des Teileprogramms	817
14.3.2.2	Achscontainer-Drehung mit implizitem Warten des Teileprogramms	817
14.3.2.3	Achscontainer-Drehung durch nur einen Kanal (z. B. Hochlauffall).....	818
14.3.3	Achscontainer Systemvariablen auswerten	818
14.3.3.1	Bedingte Verzweigung	818
14.3.3.2	Statische Synchronaktion mit \$AN_AXCTSWA	818
14.3.3.3	Sicher auf Achscontainer-Drehung warten	819
14.3.4	Konfiguration einer Mehrspindel-Drehmaschine	820
14.3.5	Lead-Link-Achse	829
14.3.5.1	Konfiguration	829
14.3.5.2	Programmierung	830
14.4	Datenlisten	832
14.4.1	Maschinendaten	832
14.4.1.1	Allgemeine Maschinendaten	832

14.4.1.2	Kanal-spezifische Maschinendaten.....	832
14.4.1.3	Achs-/Spindel-spezifische Maschinendaten.....	832
14.4.2	Settingdaten.....	833
14.4.2.1	Allgemeine Settingdaten.....	833
14.4.2.2	Achs-/Spindel-spezifische Settingdaten.....	833
14.4.3	Signale.....	833
14.4.3.1	Signale von NC.....	833
14.4.3.2	Signale von HMI/PLC.....	833
14.4.3.3	Allgemein Online-Schnittstelle.....	834
14.4.3.4	Signale von Achse/Spindel.....	835
14.4.4	Systemvariablen.....	835
15	S7: Speicherkonfiguration.....	837
15.1	Einführung.....	837
15.2	Aktives und passives Filesystem.....	837
15.3	Inbetriebnahme.....	839
15.3.1	Konfiguration.....	839
15.3.2	Umkonfiguration.....	839
15.4	Konfiguration des statischen Anwenderspeichers.....	840
15.4.1	Aufteilung des statischen NC-Speichers.....	840
15.4.2	Inbetriebnahme.....	843
15.5	Konfiguration des dynamischen Anwenderspeichers.....	843
15.5.1	Aufteilung des dynamischen NC-Speichers.....	843
15.5.2	Inbetriebnahme.....	845
15.6	Randbedingungen.....	846
15.6.1	Abhängigkeit von Kanal- und Werkzeugträgeranzahl.....	846
15.6.2	Nachträgliche Verringerung der Kanal- und/oder Achsanzahl.....	846
15.7	Datenlisten.....	847
15.7.1	Maschinendaten.....	847
15.7.1.1	Allgemeine Maschinendaten.....	847
15.7.1.2	Kanal-spezifische Maschinendaten.....	850
15.7.1.3	Achs-/Spindel-spezifische Maschinendaten.....	852
A	Anhang.....	853
A.1	Liste der Abkürzungen.....	853
A.2	Verfügbare IPCs.....	861
	Index.....	863

Grundlegende Sicherheitshinweise

1.1 Allgemeine Sicherheitshinweise

 WARNUNG
Lebensgefahr bei Nichtbeachtung von Sicherheitshinweisen und Restrisiken
Bei 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
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 Parametrierung vor unbefugtem Zugriff.• Beherrschen Sie mögliche Fehlfunktionen durch geeignete Maßnahmen, z. B. NOT-HALT oder NOT-AUS.

1.2 Gewährleistung und Haftung für Applikationsbeispiele

Applikationsbeispiele sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten.

Applikationsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern sollen lediglich Hilfestellung bieten bei typischen Aufgabenstellungen.

Als Anwender sind Sie für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Applikationsbeispiele entheben Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung.

1.3 Industrial Security

Hinweis

Industrial Security

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen einen Bestandteil eines solchen Konzepts.

Die Kunden sind dafür verantwortlich, unbefugten Zugriff auf ihre Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Diese Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und nur wenn entsprechende Schutzmaßnahmen (z. B. Nutzung von Firewalls und/oder Netzwerksegmentierung) ergriffen wurden.

Weiterführende Informationen zu möglichen Schutzmaßnahmen im Bereich Industrial Security finden Sie unter:

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

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Produkt-Updates anzuwenden, sobald sie zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter:

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

Weitere Informationen finden Sie im Internet:

Projektierungshandbuch Industrial Security (<https://support.industry.siemens.com/cs/ww/de/view/108862708>)

**WARNUNG****Unsichere Betriebszustände durch Manipulation der Software**

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

- Halten Sie die Software aktuell.
- Integrieren Sie die Automatisierungs- und Antriebskomponenten in ein ganzheitliches Industrial Security-Konzept der Anlage oder Maschine nach dem aktuellen Stand der Technik.
- Berücksichtigen Sie bei Ihrem ganzheitlichen Industrial Security-Konzept alle eingesetzten Produkte.
- Schützen Sie die Dateien in Wechselspeichermedien vor Schadsoftware durch entsprechende Schutzmaßnahmen, z. B. Virens Scanner.
- Prüfen Sie beim Abschluss der Inbetriebnahme alle security-relevanten Einstellungen.
- Schützen Sie den Antrieb vor unberechtigten Änderungen, indem Sie die Umrichterfunktion "Know-How-Schutz" aktivieren.

K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten

2.1 Kurzbeschreibung

Kanal

Ein Kanal der NC stellt die kleinste Einheit für das manuelle Verfahren von Achsen und die automatische Abarbeitung von Teileprogrammen dar. Ein Kanal befindet sich zu einem Zeitpunkt immer in einer bestimmten Betriebsart, z. B. AUTOMATIK, MDA oder JOG. Ein Kanal kann als eigenständige NC betrachtet werden.

Betriebsartengruppe (BAG)

Ein Kanal gehört immer einer Betriebsartengruppe an. Eine Betriebsartengruppe kann auch aus mehreren Kanälen bestehen.

Das Kennzeichen einer Betriebsartengruppe ist, dass sich alle Kanäle der Betriebsartengruppe zu einem Zeitpunkt immer in der gleichen Betriebsart, z. B. AUTOMATIK, MDA oder JOG, befinden. Dies wird durch die NC-interne Betriebsartenlogik sichergestellt.

Eine Betriebsartengruppe kann als eigenständige mehrkanalige NC betrachtet werden.

Kanallücken

Bei der Konfiguration der Kanäle können Platzhalter-Kanäle vorgesehen werden, um die Konfiguration über Maschinen einer Serie möglichst einheitlich anzulegen und nur die tatsächlich benutzten Kanäle zu aktivieren.

Programmtest

Zum Testen oder Einfahren eines neuen Teileprogramms gibt es folgende Möglichkeiten:

- Programmbearbeitung ohne Sollwertausgaben
- Programmbearbeitung im Einzelsatzbetrieb
- Programmbearbeitung mit Probelaufvorschub
- Teileprogrammsätze ausblenden
- Satzsuchlauf mit oder ohne Berechnung

Satzsuchlauf

Über Satzsuchlauf gibt es folgende Programmsimulationen zum Suchen bestimmter Programmstellen:

- Typ 1 ohne Berechnung an Kontur
- Typ 2 mit Berechnung an Kontur

2.1 Kurzbeschreibung

- Typ 4 mit Berechnung an Satzendpunkt
- Typ 5 Selbsttätiger Start der angewählten Programmstelle mit Berechnung aller erforderlichen Daten aus der Vorgeschichte
- Automatischer Start eines ASUPs nach Satzsuchlauf
- Kaskadierter Satzsuchlauf
- Kanalübergreifender Satzsuchlauf im Modus "Programmtest"

Programmbetrieb

Programmbetrieb liegt dann vor, wenn in der Betriebsart AUTOMATIK oder MDA Teileprogramme bzw. Teileprogrammsätze abgearbeitet werden. Während der Abarbeitung kann dabei der Programmablauf durch PLC-Nahtstellensignale und Kommandos beeinflusst werden.

Für jeden Kanal können Grundstellungen über kanalspezifische Maschinendaten vorgegeben werden. Diese Grundstellungen wirken u. a. auf die G-Gruppen sowie auf die Hilfsfunktionsausgabe.

Die Anwahl eines Teileprogramms kann nur erfolgen, wenn sich der betreffende Kanal im Reset-Zustand befindet.

Außerdem werden alle weiteren Programmabläufe mit PLC-Nahtstellensignalen und den entsprechenden Kommandos behandelt.

- Starten des Teileprogramms bzw. des Teileprogrammsatzes
- Teileprogrammrechnung und Programmbeeinflussung
- RESET-Kommando, Programmzustand und Kanalzustand
- Reaktionen auf Bedienungs- und Programmaktionen
- Ereignisgesteuerte Programmaufrufe

Asynchrone Unterprogramme (ASUP), Interruptroutinen

Mit Hilfe von Interrupteingängen ist die NC in der Lage, die aktuelle Teileprogrammbearbeitung zu unterbrechen und auf hochpriorie Ereignisse in Interruptroutinen/ASUPs zu reagieren.

Einzelsatz

In der Funktion Einzelsatz kann der Anwender das Teileprogramm satzweise abarbeiten.

Es gibt 3 Einstellungsarten der Funktion Einzelsatz:

- SLB1: = IPO-Einzelsatz
- SLB2: = Decodiereinzelsatz
- SLB3: = Stopp im Zyklus

Basis-Satzanzeige

Zur bestehenden Satzanzeige können über eine zweite, der Basis-Satzanzeige, alle Sätze angezeigt werden, die eine Aktion an der Maschine bewirken werden.

Die tatsächlich angefahrenen Endpositionen werden als Absolutposition dargestellt. Die Positionswerte beziehen sich wahlweise auf das Werkstückkoordinatensystem (WKS) oder auf das Einstellbare Nullpunkt-System (ENS).

Programm von Extern abarbeiten

Bei der Bearbeitung komplexer Werkstücke kommt es vor, dass der Speicher der NC für die Programme nicht ausreicht. Mit der Funktion "Abarbeiten von Extern" oder "EES (Execution from External Storage)" können Teileprogramme von einem externen Speicher (z. B. von der Festplatte) aufgerufen und abgearbeitet werden.

Verhalten nach POWER ON, Reset, ...

Das Verhalten der Steuerung nach:

- Hochlauf (POWER ON)
- Reset/Teileprogramm-Ende
- Teileprogramm-Start

kann für Funktionen wie z. B. G-Befehle, Werkzeuglängenkorrektur, Transformation, Mitschleppverbände, Tangentiale Nachführung, Programmierbare Synchronspindel, für bestimmte Systemeinstellungen durch Maschinendaten verändert werden.

Unterprogrammaufruf mit M-, T- bzw. D-Funktionen

Für bestimmte Anwendungen kann es vorteilhaft sein, M- T- bzw. D-Funktionen sowie einige NC-Sprachbefehle (*SPOS*, *SPOSA*) durch einen Unterprogrammaufruf zu ersetzen. Dies kann z. B. für den Aufruf der Werkzeugwechselroutine genutzt werden.

Mit entsprechenden Maschinendaten können Unterprogramme mit M- T- bzw. D-Funktionen z. B. für einen Getriebestufenwechsel definiert und beeinflusst werden.

Programmlaufzeit / Werkstückzähler

Zur Unterstützung des Bearbeiters an der Werkzeugmaschine werden Informationen zur Programmlaufzeit und zur Werkstückzählung bereitgestellt.

Die dabei definierte Funktionalität ist **nicht identisch mit Funktionen der Werkzeugverwaltung** und besonders für NC-Systeme ohne Werkzeugverwaltung vorgesehen.

2.2 Betriebsartengruppe (BAG)

Betriebsartengruppe

In einer Betriebsartengruppe (BAG) werden mehrere Kanäle einer NC zu einer Bearbeitungseinheit zusammengefasst. Sie stellt somit im Prinzip eine eigenständige "NC" innerhalb einer NC dar.

Eine BAG ist im Wesentlichen dadurch gekennzeichnet, dass sich alle ihr zugeordneten Kanäle zu einem Zeitpunkt immer in der gleichen Betriebsart (AUTOMATIK, JOG, MDA) befinden.

Hinweis

Im weiteren Verlauf dieser Beschreibung wird von einer BAG und einem Kanal ausgegangen.

Funktionen, die mehrere Kanäle benötigen (z. B. Funktion "Achstausch"), sind beschrieben in:

Weitere Informationen:

Weitere Informationen finden Sie unter K5: Kanalübergreifende Programmkoordination und kanalweises Einfahren (Seite 531).

Weitere Informationen finden Sie unter Parametrierung

Die Zuordnung eines Kanals zu einer BAG erfolgt über das Maschinendatum:

MD10010 \$MN_ASSIGN_CHAN_TO_MODE_GROUP[<Kanal-Index>] = BAG-Nummer

Alle Kanäle mit der gleichen BAG-Nummer bilden eine BAG.

Hinweis

Eine NC umfasst standardmäßig immer eine BAG mit einem Kanal. Die Parametrierung einer NC ohne Kanal ist nicht möglich.

BAG-spezifische NC/PLC-Nahtstellensignale

Die BAG-spezifischen NC/PLC-Nahtstellensignale liegen im Datenbaustein DB11.

Die BAG-spezifische NC/PLC-Nahtstelle umfasst im Wesentlichen folgende Nahtstellensignale:

- Anforderungssignale PLC → NC
 - BAG-Reset
 - BAG-Stop Achsen plus Spindel
 - BAG-Stop
 - Betriebsartenwechsel Sperre
 - Betriebsart: JOG, MDA, AUTOM.
 - Einzelsatz: Typ A, Typ B
 - Maschinenfunktion: REF, REPOS, TEACH IN, INC x
- Statussignale NC → PLC
 - Betriebsart-Strobe: JOG, MDA, AUTOMATIK
 - Maschinenfunktion-Strobe: REF, REPOS, TEACH IN
 - Alle Kanäle im Reset-Zusand
 - BAG reseted
 - BAG betriebsbereit
 - aktive Betriebsart: JOG, MDA, AUTOMATIK
 - aktive Maschinenfunktion: REF, REPOS, TEACH, INC x

Aktivierte und nicht aktivierte Kanäle, Kanallücken

Aktivierter Kanal

Ein Kanal mit einer BAG-Nummer $\neq 0$ ist ein aktiver Kanal.

Nicht aktivierter Kanal

Ein Kanal mit BAG-Nummer 0 ist ein nicht aktivierte Kanal. Er belegt Steuerungs-intern keinen Speicherplatz.

Kanallücken

Ein Kanal mit BAG-Nummer 0 ist nicht nur ein nicht aktivierte Kanal. Er stellt in der Abfolge der Kanäle auch eine sogenannte Kanallücke dar.

Der Vorteil von Kanallücken ist, dass damit innerhalb einer Serie von bauähnliche Maschinen die Konfigurationsdaten weitgehend gleich gehalten werden können. Für die spezifische Inbetriebnahme werden dann nur die Kanäle aktiviert, die für die vorliegende Maschine

2.2 Betriebsartengruppe (BAG)

benötigt werden. Der nicht belegte Speicher steht dann als zusätzlicher Anwenderspeicher zur freien Verfügung.

Tabelle 2-1 Beispiel

Maschinendatum	Bedeutung
MD10010 \$MN_ASSIGN_CHAN_TO_MODE_GROUP[0] = 1	Kanal 1, BAG 1
MD10010 \$MN_ASSIGN_CHAN_TO_MODE_GROUP[1] = 2	Kanal 2, BAG 2
MD10010 \$MN_ASSIGN_CHAN_TO_MODE_GROUP[2] = 0	Kanal 3, nicht vorhanden
MD10010 \$MN_ASSIGN_CHAN_TO_MODE_GROUP[3] = 1	Kanal 4, BAG 1
MD10010 \$MN_ASSIGN_CHAN_TO_MODE_GROUP[4] = 2	Kanal 5, BAG 2

2.2.1 BAG-Stop

Funktion

Über die folgenden NC/PLC-Nahtstellensignale werden in allen Kanälen der BAG die Verfahrbewegungen der Achsen bzw. Achsen und Spindeln angehalten, sowie die Teileprogrammabarbeitung unterbrochen:

DB11 DBX0.5 (BAG-Stop)

DB11 DBX0.6 (BAG-Stop Achsen plus Spindeln)

2.2.2 BAG-Reset

Funktion

Die Anforderung zum BAG-Reset erfolgt über das BAG-spezifische NC/PLC-Nahtstellensignal:

DB11 DBX0.7 = 1 (BAG-Reset)

Wirkung

Wirkung auf die Kanäle der BAG:

- Die Teileprogrammaufbereitung (Vorlauf) wird gestoppt.
- Alle Achsen und Spindeln werden entlang ihrer Beschleunigungskennlinien ohne Konturverletzung auf Stillstand abgebremst.
- Die noch nicht an die PLC ausgegebenen Hilfsfunktionen werden nicht mehr ausgegeben.

- Die Vorlaufzeiger werden auf die Unterbrechungsstelle und die Satzzeiger auf den Anfang der jeweiligen Teileprogramme gesetzt.
- Alle Grundstellungen (z. B. der G-Befehle) werden auf die parametrisierten Werte eingestellt.
- Alle Alarmer mit Löschkriterium "Kanal-Reset" werden gelöscht.

Wenn alle Kanäle der BAG im Reset-Zustand sind, dann:

- werden alle Alarmer mit Löschkriterium "BAG-Reset" gelöscht.
- werden der Abschluss des BAG-Resets und die Betriebsbereitschaft der BAG an der NC/PLC-Nahtstelle angezeigt:
DB11 DBX6.7 = 1 (alle Kanäle im Reset-Zustand)
DB11 DBX6.3 = 1 (BAG betriebsbereit)

2.3 Betriebsarten und Betriebsartenwechsel

Eindeutige Betriebsart

Alle Kanäle einer Betriebsartengruppe (BAG) befinden sich immer in derselben Betriebsart:

- AUTOMATIK
- JOG
- MDA

Sind einzelne Kanäle unterschiedlichen Betriebsartengruppen zugeordnet, so erfolgt über eine Kanalumschaltung auch eine Umschaltung auf die entsprechende BAG. Auf diese Weise ist über eine Kanalumschaltung ein Betriebsartenwechsel möglich.

Betriebsarten

Es stehen folgende Betriebsarten zur Verfügung:

- **AUTOMATIK**
Automatisches Abarbeiten von Teileprogrammen:
 - Teileprogrammtest
 - Alle Kanäle der BAG können gleichzeitig aktiv sein.
- **JOG in AUTOMATIK**
JOG in AUTOMATIK ist eine Erweiterung der Betriebsart AUTOMATIK mit dem Ziel der Bedienungsvereinfachung. JOG kann ausgeführt werden, ohne AUTOMATIK zu verlassen, wenn Randbedingungen dies zulassen.

- **JOG**
Manuelles Verfahren von Achsen über Verfahrstasten der Maschinensteuertafel oder über ein an der Maschinensteuertafel angeschlossenes Handrad:
 - Kanalspezifische Signale und Verriegelungen werden bei Bewegungen, die per ASUP oder über statische Synchronaktionen ausgeführt werden, beachtet.
 - Kopplungen werden berücksichtigt.
 - Jeder Kanal in der BAG kann aktiv sein.
- **MDA**
Manual Data Automatic (Eingabe der Sätze erfolgt über die Bedienoberfläche):
 - Eingeschränktes Abarbeiten von Teileprogrammen und Teileprogrammteilen.
 - Teileprogrammtest
 - Maximal 1 Kanal pro BAG kann aktiv sein (nur bei TEACH IN).
 - Achsen können nur in unterlagerten Maschinenfunktionen wie JOG, REPOS oder TEACHIN manuell verfahren werden.

Gültig für alle Betriebsarten

Betriebsarten-übergreifende Synchronaktionen

Übergreifend für alle Betriebsarten, können modale Synchronaktionen per IDS für folgende Funktionen parallel zum Kanal abgearbeitet werden:

- Kommandoachsfunktionen
- Spindelfunktionen
- Technologiezyklen

Anwahl

Über die Bedienoberfläche kann der Anwender die gewünschte Betriebsart mit Hilfe von Softkeys anwählen.

Diese Anwahl (AUTOMATIK, MDA oder JOG) wird an die NC/PLC-Nahtstelle zur PLC weitergeleitet, aber noch nicht aktiviert:
DB11 DBX4.0, 0.1, 0.2 (Betriebsart-Strobe)

Aktivierung und Prioritäten

Die Betriebsart der BAG wird aktiviert über die NC/PLC-Nahtstelle:

DB11 DBX0.0, 0.1, 0.2 (Betriebsart)

Werden mehrere Betriebsarten zur gleichen Zeit angewählt, besteht folgende Priorität:

Priorität	Betriebsart	BAG-Signal (NC → PLC)
1. Priorität, hoch	JOG	DB11 DBX0.2
2. Priorität, mittel	MDA	DB11 DBX0.1
3. Priorität, niedrig	AUTOMATIK	DB11 DBX0.0

Anzeige

Die aktuelle Betriebsart der BAG wird angezeigt über die NC/PLC-Nahtstelle:

DB11 DBX6.0, 0.1, 0.2 (aktive Betriebsart)

BAG-Signal (NC → PLC)	Aktive Betriebsart
DB11 DBX6.2	JOG
DB11 DBX6.1	MDA
DB11 DBX6.0	AUTOMATIK

Maschinenfunktionen

Innerhalb einer Betriebsart können Maschinenfunktionen angewählt werden, die ebenfalls innerhalb der BAG gelten:

- Maschinenfunktionen innerhalb der Betriebsart **JOG**
 - REF (Referenzpunktfahren)
 - REPOS (Repositionieren)
 - JOG-Retract (Rückzugsbewegung in Werkzeugrichtung)
- Maschinenfunktionen innerhalb der Betriebsart **MDA**
 - REF (Referenzpunktfahren)
 - REPOS (Repositionieren)
 - TEACHIN (Ein-Teachen von Achspositionen)

NC/PLC-Nahtstellensignale

- DB11 DBX5.0, 0.1, 0.2 (Maschinenfunktion-Strobe): Anforderung
- DB11 DBX1.0, 0.1, 0.2 (Maschinenfunktion): Aktivierung
- DB11 DBX7.0-2 (aktive Maschinenfunktion): Rückmeldung

Kanal-Zustände

- **Kanal-Reset**
Die Maschine befindet sich im Grundzustand. Dieser wird über das PLC-Programm vom Maschinenhersteller z. B. nach dem Einschalten oder nach Programmende definiert.
- **Kanal aktiv**
Ein Programm ist gestartet, die Programmabarbeitung oder Referenzpunktfahren läuft.
- **Kanal unterbrochen**
Das laufende Programm oder Referenzpunktfahren sind unterbrochen worden.

Funktionen innerhalb von Betriebsarten

Die Betriebsarten werden ergänzt durch anwenderspezifische Funktionen. Die Funktionen sind technologie- und maschinenunabhängig und können aus den drei Kanalzuständen "Kanal-Reset", "Kanal aktiv" oder "Kanal unterbrochen" gestartet und/oder bearbeitet werden.

Randbedingung für die Maschinenfunktion TEACH IN

TEACH IN ist nicht zulässig für Führungs- oder Leitachsen eines aktiven Achsverbundes, z. B. bei:

- Gantry-Achsverbund oder ein Gantry-Achsenpaar
- Mitschleppverbund von Leitachse mit Folgeachse

JOG in AUTOMATIK

JOG in Betriebsart AUTOMATIK ist zulässig, wenn die BAG im Zustand "RESET" und die Achse JOG-fähig ist. Zustand "RESET" für die BAG bedeutet:

- Alle Kanäle sind im Zustand "RESET"
- Alle Programme sind abgebrochen
- In keinem Kanal ist DRF aktiv

Eine Achse ist **JOG-fähig** wenn sie in keinem der folgenden Zustände ist:

- PLC-Achse als konkurrierende Positionierachse (Anforderung der Achse von der PLC)
- Kommandoachse (die Achse wurde von einer Synchronaktion programmiert und die Bewegung ist noch nicht zu Ende)
- Drehende Spindel (Spindel die über RESET hinaus dreht)
- Asynchrone Pendelachse

Hinweis: Die Eigenschaft "JOG-fähig" ist unabhängig von der Funktion "JOG in AUTOMATIK".

Aktivierung

Die Funktion "JOG in AUTOMATIK" ist aktivierbar mit dem Maschinendatum:

MD10735 \$MN_JOG_MODE_MASK

- Vor POWER ON muss folgendes Maschinendatum gesetzt sein:
MD10735 \$MN_JOG_MODE_MASK, Bit 0 = 1
- Der Anwender schaltet nach AUTO (PLC-Anwenderschnittstelle DB11 DBX0.0 = 0→1 Flanke). "JOG in AUTOMATIK" ist damit aktiv, wenn vorher der NC in allen Kanälen der BAG den Kanalzustand "RESET" und den Programmzustand "Abgebrochen" hatte. Zudem muss die dann betroffene Achse "JOG-fähig" sein. DRF muss abgeschaltet sein oder werden.
- In allen Kanälen der BAG ohne den Kanalzustand "RESET" und den Programmzustand "Abgebrochen" wird RESET ausgelöst oder das laufende Programm kommt mit M30/M2 zu Ende.
- Die betroffene Achse wird automatisch "JOG-fähig" gemacht (z. B. Achstauch: PLC → NC).

Hinweis: In den meisten Anwendungsfällen sind die zu verfahrenen Achsen "JOG-fähig" und mit dem Umschalten nach AUTOMATIK ist damit auch "JOG in AUTOMATIK" aktiv.

Merkmale

- Die Tasten +/- leiten eine JOG-Bewegung ein, die BAG wird **intern** nach JOG geschaltet. (Kurz "Intern-JOG").
- Die Bewegungen der Handräder leiten eine JOG-Bewegung ein, die BAG wird **intern** nach JOG geschaltet, außer es ist DRF aktiv.
- Eine begonnene JOG-Bewegung ist erst zu Ende, wenn die Endposition des Inkrements (falls dies eingestellt worden war) erreicht wurde, oder die Bewegung mit "Restweglöschen" abgebrochen wurde.
Damit kann ein Inkrement mit Stopp angehalten und mit Start wieder bis zum Ende gefahren werden. Solange bleibt der NC im "Intern-JOG". Ein teilweise abgefahrenes Inkrement ist möglich, muss aber nicht mit Stopp unterbrochen sein. Es gibt einen Modus, bei dem eine Wegnahme der Fahrtaste zur Unterbrechung innerhalb eines Inkrements führt.
- Ohne eine JOG-Bewegung verhält sich "JOG in AUTOMATIK" wie "Automatik". Insbesondere startet die Start-Taste das angewählte Teileprogramm und der entsprechende Softkey des HMI löst einen Satzsuchlauf aus.
- Mit einer aktiven JOG-Bewegung ist der NC intern in JOG und eine Satzsuchlaufanforderung wird abgelehnt und der Start startet das Teileprogramm nicht. Der Start startet ggf. das verbliebene Inkrement oder er bleibt ohne Wirkung.
- Solange eine Achse der BAG in JOG verfahren wird, bleibt die BAG intern in JOG.
Bemerkung: Diese Phase kann mit der JOG-Bewegung einer Achse beginnen und mit dem Ende der JOG-Bewegung einer *anderen* Achse enden.
- Eine Achse mit aktiver JOG-Bewegung darf keinem Achstausch unterworfen werden. (Die Achse könnte die BAG wechseln). Der NC verriegelt diesen Achstausch-Versuch.
- Die PLC-Anwenderschnittstelle zeigt die Betriebsart "Automatik" an:
 - DB11 DBX6.0, 6.1, 6.2 = 1
 - DB11 DBX7.0, 7.1, 7.2 = 0
- Die NC/PLC-Nahtstelle zeigt an, wenn in "JOG in AUTOMATIK" sich die BAG in "BAG-RESET" befindet.
 - DB11 DBX6.4 (BAG reseted, BAG 1)
 - DB11 DBX26.4 (BAG reseted, BAG 2)
 - DB11 DBX46.4 (BAG reseted, BAG 3)
- Die NC/PLC-Nahtstelle zeigt an, wenn in "JOG in AUTOMATIK" die NC intern nach "Intern-JOG" geschaltet hat.
 - DB11 DBX6.5 (NC intern JOG aktiv, BAG 1)
 - DB11 DBX26.5 (NC intern JOG aktiv, BAG 2)
 - DB11 DBX46.5 (NC intern JOG aktiv, BAG 3)

Randbedingungen

"JOG in AUTOMATIK" kann intern nur dann nach JOG wechseln, wenn die BAG im Zustand "BAG-RESET" ist. D. h. mitten in einem gestoppten Programm kann nicht unmittelbar gejoggt werden. Der Benutzer kann joggen, wenn er in dieser Situation die JOG-Taste oder die Reset-Taste *in allen Kanälen* der BAG drückt.

2.3 Betriebsarten und Betriebsartenwechsel

Mit der Anwahl von Automatik werden die INC-Tasten abgeschaltet und der Anwender kann/muss die INC-Tasten erneut drücken, um ein von ihm gewünschtes Inkrement anzuwählen. Schaltet der NC nach "Intern-JOG", so bleibt das angewählte Inkrement erhalten.

Versucht der Anwender die Geo- oder Orientierungsachsen zu joggen, so wechselt der NC auch nach "Intern-JOG" und die Bewegung wird ausgeführt. Dabei können mehrere Achsen physisch bewegt werden, die alle "JOG-fähig" sein müssen.

Nach der JOG-Bewegung schaltet der NC "Intern-JOG" wieder ab und wählt damit AUTO wieder an. Der Interne Modewechsel wird verzögert nach dem Bewegungsende durchgeführt. Damit werden unnötig viele Schaltvorgänge vermieden, die z. B. beim Handrad auftreten könnten. Die PLC darf sich nur auf das PLC-Signal "Intern-JOG-Aktiv" verlassen.

Der NC schaltet auch dann nach "Intern-JOG", wenn die Achse keine Freigabe hat.

2.3.1 Überwachungen und Verriegelungen der einzelnen Betriebsarten

Kanalzustand bestimmt Überwachungen

Überwachungen in den Betriebsarten

In den einzelnen Betriebsarten sind unterschiedliche Überwachungen aktiv. Diese Überwachungen sind Technologie- und Maschinen-unabhängig.

In jeder Betriebsart, in Abhängigkeit des Betriebszustandes, ist ein Teil der Überwachungen aktiv. Welche Überwachungen in welcher Betriebsart und in welchem Betriebszustand aktiv sind, wird durch den Kanalzustand bestimmt.

Verriegelungen in den Betriebsarten

In den einzelnen Betriebsarten können unterschiedliche Verriegelungen aktiv sein. Diese Verriegelungen sind Technologie- und Maschinen-unabhängig.

In jeder Betriebsart, in Abhängigkeit des Betriebszustandes, können fast alle Verriegelungen aktiviert werden.

2.3.2 Betriebsartenwechsel

Einführung

Ein Betriebsartenwechsel wird über die BAG-Nahtstelle (DB11) angefordert und aktiviert. Eine Betriebsartengruppe befindet sich entweder in der Betriebsart AUTOMATIK, JOG oder MDA, d. h. mehrere Kanäle einer Betriebsartengruppe können gleichzeitig keine unterschiedlichen Betriebsarten annehmen.

Ob die angeforderte Betriebsart erreichbar ist und wie diese durchgeführt wird, ist maschinenspezifisch über das PLC-Programm projektierbar.

Hinweis

Die Betriebsart wird erst dann steuerungsintern gewechselt, wenn der "Kanalzustand aktiv" nicht mehr vorliegt. Zum fehlerfreien Betriebsartenwechsel müssen aber alle Kanäle einen zulässigen Betriebszustand eingenommen haben.

Mögliche Betriebsartenwechsel

Die möglichen Betriebsartenwechsel können Sie der folgenden Tabelle für einen Kanal entnehmen:

	AUTOMATIK		JOG			MDA			
			Reset	AUTO	MDA	manuelles Verfahren		AUTO	
	Reset	unterbr.	Reset	unterbr.	unterbr.	Reset	unterbr.	aktiv	unterbr.
AUTOMATIK			X	X		X			
JOG	X	X				X	X		X
MDA	X	X	X		X				

Die mit "X" gekennzeichneten Positionen sind mögliche Betriebsartenwechsel.

Sonderfälle

- Fehler bei Betriebsartenwechsel**
 Wenn eine Betriebsartenwechsel-Anforderung vom System abgewiesen wurde, erfolgt die Fehlermeldung "BA-Wechsel erst nach NC-Stop möglich". Diese Fehlermeldung kann gelöscht werden, ohne den Kanalzustand zu ändern.
- Betriebsartenwechsel-Sperre**
 Mit Hilfe des Nahtstellensignals:
 DB11 DBX0.4 (Betriebsartenwechsel-Sperre)
 kann ein Wechseln der Betriebsart verhindert werden.
 Es wird dabei schon die Betriebsartenwechsel-Anforderung unterdrückt.
 Der Anwender muss eine Meldung über die aktive Sperre für den Bediener selber projektieren. Vom System wird dafür nichts abgesetzt.
- Betriebsartenwechsel von MDA nach JOG**
 Sind nach einem Betriebsartenwechsel von MDA nach JOG alle Kanäle der BAG im Reset-Zustand, so wechselt die NC von JOG nach AUTO. In diesem Zustand können die Teileprogrammbeefehle `START` oder `INIT` durchgeführt werden.
 Ist nach dem Betriebsartenwechsel ein Kanal der BAG nicht mehr im Reset-Zustand, so wird der Teileprogrammbeefehl `START` in dieser Situation mit dem Alarm 16952 abgelehnt.

2.4 Kanal

2.4.1 Eigenschaften

In einem Kanal der NC werden die vom Anwender vorgegebenen Teileprogramme abgearbeitet.

Folgende Eigenschaften charakterisieren einen Kanal:

- Ein Kanal ist immer einer Betriebsartengruppe (BAG) zugeordnet.
- Zu einem Zeitpunkt, kann ein Kanal immer nur genau ein Teileprogramm abarbeiten.
- Einem Kanal sind Maschinen-, Geometrie- und Zusatzachsen sowie Spindeln zugeordnet. Nur diese können über die im Kanal abgearbeiteten Teileprogramm verfahren werden.
- Ein Kanal besteht aus den internen Einheiten
 - Vorlauf (Programmdekodierung und Satzaufbereitung)
 - Hauptlauf (Bahn- und Achsinterpolation).
- Ein Kanal hat eine Schnittstelle zur PLC. Über diese NC/PLC-Nahtstelle können vom PLC-Anwenderprogramm aus diverse kanalspezifische Zustandsdaten gelesen und Anforderung an den Kanal geschrieben werden.
- In einem Kanal wirken kanalspezifische (siehe Funktionshandbuch *Werkzeuge* unter *Werkzeugkorrektur*).
- In einem Kanal wirken kanalspezifische Koordinatensysteme.
- Jeder Kanal hat standartmäßig einen eindeutigen Kanalnamen. Der vorgegebene Kanalname kann über Maschinendaten verändert werden:
MD20000 \$MC_CHAN_NAME = "<Kanalname>"

Mehrere Kanäle können zu einer Betriebsartengruppe (BAG) zusammengefasst werden. Die Kanäle einer BAG befinden sich immer in der gleichen Betriebsart (AUTOMATIK, JOG, MDA).

Kanalkonfiguration

Kanäle können über folgendes Maschinendatum mit einem eigenen Kanalnamen belegt werden:

MD20000 \$MC_CHAN_NAME (Kanalname)

Die verschiedenen Achsen werden per Maschinendatum den vorhandenen Kanälen zugeordnet. Für eine Achse/Spindel kann es zu einer Zeit immer nur einen sollwertgebenden Kanal geben. Der Istwert der Achse/Spindel kann von mehreren Kanälen gleichzeitig gelesen werden. Die Achse/Spindel muss dem jeweiligen Kanal bekannt sein.

Des Weiteren können noch folgende kanalspezifische Einstellungen über Maschinendaten definiert werden:

- Löschstellungen bzw. Programmiergrundstellungen von G-Gruppen über das Maschinendatum:
MD20150 \$MC_GCODE_RESET_VALUES (Löschstellung der G-Gruppen)
- Hilfsfunktionsgruppen bezüglich der Zusammenstellung und des Ausgabezeitpunkts

- Transformationsbedingungen zwischen den Maschinenachsen und den Geometrieachsen
- Weitere Einstellungen für die Abarbeitung eines Teileprogramms

Änderung der Kanalzuordnung

Eine Änderung der Kanalkonfiguration kann online nicht auf programmtechnischem Wege in einem Teileprogramm oder über das PLC-Anwenderprogramm erfolgen. Änderungen in der Konfiguration müssen über die Maschinendaten erfolgen. Die Änderungen sind erst nach einem erneuten POWER ON wirksam.

Containerachsen und Linkachsen

Ein Achscontainer fasst eine Gruppe von Achsen eines Containers zusammen. Diese Achsen werden als Containerachsen bezeichnet. Hierbei wird einer Kanalachse ein Zeiger auf einem Containerslot (Ringpuffer-Platz innerhalb des entsprechenden Containers) zugeordnet. In diesem Slot befindet sich zwischenzeitlich eine der in diesem Container zusammengefassten Achsen.

Maschinenachsen im Achscontainer müssen zu jedem Zeitpunkt jeweils genau **einer** Kanalachse zugewiesen sein.

Link-Achsen können fest einem Kanal oder dynamischen (Achscontainer-Switch) mehreren Kanälen der lokalen, oder der anderen NCU zugeordnet werden. Sie sind aus Sicht einer der Kanäle, an deren NCU die Achse nicht physikalisch angeschlossen ist, eine nicht lokale Achse.

Die herbeigeführte Zuordnung der Linkachsen zu einem Kanal erfolgt:

- Für feste Zuordnung über Maschinendatum:
Direktes logisches Maschinenachsbild auf Linkachsen zeigen lassen.
- Für dynamische Zuordnung:
Achscontainerslot - Maschinendatum auf Linkachsen zeigen lassen.

Weitere Informationen zu Link- und Containerachsen finden Sie in:

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme (B3)

Nahtstellensignale

Die Signale des 1. Kanals liegen in der NC/PLC-Nahtstelle im DB21, die vom Kanal 2 im DB22. Der oder die Kanäle können von der PLC bzw. NC überwacht bzw. beeinflusst werden.

Kanalspezifische Technologie-Angabe

Für jeden Kanal kann die verwendete Technologie angegeben werden:

MD27800 \$MC_TECHNOLOGY_MODE

Im Auslieferungszustand sind standardmäßig Maschinendaten für die Technologie Fräsen aktiv.

Spindelfunktionen über PLC

Spindelfunktionen können neben dem Funktionsbaustein FC18 auch über axiale NC/PLC-Nahtstellensignale parallel zu laufenden Teileprogrammen gestartet und gestoppt werden.

Voraussetzungen:

- Kanalzustand: "Unterbrochen" oder "RESET"
- Programmzustand: "Unterbrochen" oder "Abgebrochen"

Folgende Funktionen können von der PLC über Nahtstellensignale gesteuert werden:

- Stopp (entspricht M5)
- Start mit Drehrichtung rechts (entspricht M3)
- Start mit Drehrichtung links (entspricht M4)
- Getriebestufe auswählen
- Positionieren (entspricht M19)

Bei mehreren Kanälen wird die von der PLC gestartete Spindel in dem Kanal aktiv, dem sie zum Startzeitpunkt zugeordnet ist.

Weitere Informationen zur speziellen Spindelschnittstelle siehe Funktionshandbuch *Achsen und Spindeln* unter *Spindeln*.

PLC-kontrollierte Einzelachsvorgänge

Statt von einem Kanal, kann eine Achse auch von der PLC kontrolliert werden. Dazu fordert die PLC über die NC/PLC-Nahtstelle die Achse von der NC an:

DB31, ... DBX28.7 = 1 (PLC kontrolliert Achse)

Folgende Funktionen können vom PLC aus kontrolliert und beeinflusst werden:

- Ablauf von Achse/Spindel abbrechen (entspricht Restweg löschen)
- Achse/Spindel stoppen bzw. unterbrechen
- Achse/Spindel weiterfahren (Bewegungsablauf fortsetzen)
- Achse/Spindel in den Grundzustand zurücksetzen

Weitere Informationen zum kanalspezifischen Signalaustausch (PLC → NC) siehe Funktionshandbuch *PLC und Grundprogramm* unter *PLC-Grundprogramm*.

Die genaue Funktionalität von autarken Einzelachsvorgängen ist beschrieben in:

Weitere Informationen

Funktionshandbuch Achsen und Spindeln; Positionierachsen

2.4.2 Startsperrung, global und kanalspezifisch

Funktion

In ShopMill / ShopTurn kann der Start eines Programms standardmäßig nur im Bereich Maschine ausgeführt werden. Der Start in den anderen Bereichen, z.B. Werkzeuge, wird über eine Startsperrung (PI-Dienst) verriegelt.

Die Startsperrung automatisierte Abläufe kann über ein NC/PLC-Nahtstellensignal aufgehoben werden.

NC/PLC-Nahtstellensignale

Startsperrung aufheben

Mit dem Nahtstellensignal wird eine anstehende Startsperrung aufgehoben:

DB21, ... DBX7.5 (Startsperrung aufheben)

Maschinendaten

Mit dem Maschinendatum wird eingestellt, dass ein Alarm angezeigt wird, falls bei gesetzter Startsperrung ein Start angefordert wird:

MD11411 \$MN_ENABLE_ALARM_MASK, Bit 6

BTSS-Variablen

Variable	Beschreibung
startLockState	Status der globalen Startsperrung
chanStartLockState	Status der kanalspezifischen Startsperrung
startRejectCounter	Startzähler bei globaler Startsperrung
startLockCounter	Startzähler bei kanalspezifischer Startsperrung

PI-Dienste

Setzen der Startsperrung (_N_STRTLK)

Mit dem PI-Dienst _N_STRTLK kann die Startsperrung global bzw. kanalspezifisch gesetzt werden.

- Globale Startsperrung
Ist die globale Startsperrung gesetzt, wird der Start eines NC-Programms in allen Kanälen unterbunden. Ein bereits laufendes NC-Programm bleibt unbeeinflusst. Erst der nächste Start eines NC-Programms wird unterbunden.
- Kanalspezifische Startsperrung
Durch die kanalspezifische Startsperrung wird nur in der Betriebsart AUTOMATIK und im Kanalzustand "Reset" der Start eines NC-Programms unterbunden.

Rücksetzen der globalen bzw. kanalspezifischen Startstperre (_N_STRTUL)

Mit dem PI-Dienst _N_STRTUL kann die Startstperre globale bzw. kanalspezifische Startstperre zurückgesetzt werden.

2.5 Programmtest

Zum Testen bzw. Einfahren eines neuen Teileprogramms gibt es mehrere Steuerungsfunktionen. Durch die Verwendung dieser Funktionen wird eine Gefährdung der Maschine während der Testphase bzw. der Zeitaufwand dafür stark verringert. Es ist möglich, mehrere Programmtestfunktionen gleichzeitig zu aktivieren, um ein besseres Ergebnis zu bekommen.

Testmöglichkeiten

Es werden folgende Testmöglichkeiten beschrieben:

- Programmbearbeitung im Zustand "Programmtest"
- Programmbearbeitung im Einzelsatzbetrieb
- Programmbearbeitung mit Probelaufvorschub
- Teileprogrammsätze ausblenden
- Satzsuchlauf mit oder ohne Berechnung

2.5.1 Programmbearbeitung im Zustand "Programmtest"

Zustand "Programmtest"

Im Zustand "Programmtest" kann ein Teileprogramm abgearbeitet werden, ohne dass die Maschinenachsen real verfahren. Damit kann der Anwender die programmierten Achspositionen sowie die Hilfsfunktionsausgaben eines Teileprogramms kontrollieren. Außerdem kann diese Programmsimulation als erweiterter Syntax-Check verwendet werden.

Hinweis

Verfahrbewegungen der Spindeln

Im Unterschied zu den Maschinenachsen sind die Verfahrbewegungen der Spindeln des Kanals im Zustand "Programmtest" defaultmäßig freigegeben (siehe Abschnitt "Grundeinstellung").

Sollen Spindeln im Zustand "Programmtest" **nicht** verfahren, müssen sie über das PLC-Anwenderprogramm explizit gesperrt werden.

Grundeinstellung

Defaultmäßig gelten für Kanäle im Zustand "Programmtest" folgende Einstellungen:

- Für die Achsen werden zum Normalbetrieb identische Sollwerte erzeugt, aber **nicht** an die Maschinenachsen ausgegeben.
- Für die Spindeln werden Sollwerte an die Maschinenachsen ausgegeben.
- Die angezeigten Istwerte der gesperrten Achsen / Spindeln werden intern aus den Sollwerten erzeugt.
- Die Befehle zur Kanalsynchronisation werden normal bearbeitet.
- Die NC/PLC-Nahtstellensignale werden normal bearbeitet.
- Die Bearbeitungszeit des Programms bleibt unverändert zum Normalbetrieb.

Der Zustand "Programmtest" kann temporär bei Bedarf kanal- oder achsspezifisch verlassen werden. Dadurch werden die Sollwerte wieder an die Maschinenachsen ausgegeben, sodass diese real an der Maschine verfahren.

Anwahl

Die Auswahl von "Programmtest" (PRT) für den im Grundbild "Maschine" angezeigten Kanal erfolgt über die Bedienoberfläche z. B. SINUMERIK Operate:

1. Softkey: Bedienbereich "Maschine" > "Programmbeeinflussung"
2. Menü "Programmbeeinflussung": Kontrollkästchen "Programmtest (PRT)" anwählen.

NC/PLC-Nahtstellensignale

Nach der Auswahl von Programmtest (PRT) sind folgende NC/PLC-Nahtstellensignale gesetzt (siehe unten Hinweis "Automatische Übertragung der Nahtstellensignale"):

- Kanäle
 - DB21, ... DBX25.7 == 1 (von HMI: Programmtest angewählt)
 - DB21, ... DBX1.7 == 1 (von PLC: Programmtest aktivieren)
 - DB21, ... DBX33.7 == 1 (von NC: Programmtest aktiv)
- Achsen
 - DB31, ... DBX128.0 == 0 (von HMI: Programmtest unterdrücken)
 - DB31, ... DBX128.1 == 0 (von HMI: Programmtest aktivieren)
 - DB31, ... DBX14.0 == 0 (von PLC: Programmtest unterdrücken)
 - DB31, ... DBX14.1 == 0 (von PLC: Programmtest aktivieren)
- Spindeln
 - DB31, ... DBX128.0 == 1 (von HMI: Programmtest unterdrücken)
 - DB31, ... DBX128.1 == 0 (von HMI: Programmtest aktivieren)
 - DB31, ... DBX14.0 == 1 (von PLC: Programmtest unterdrücken)
 - DB31, ... DBX14.1 == 0 (von PLC: Programmtest aktivieren)

Hinweis

Defaultzustand der Verfahrbewegungen

Der Defaultzustand bezüglich der Verfahrbewegungen nach Anwahl von "Programmtest" im Kanal ist:

- Achsen: Gesperrt
 - Spindeln: Freigegeben
-

Hinweis

Automatische Übertragung der Nahtstellensignale

Die HMI-Anforderungssignale DB21, ... DBX128.0 / .1 werden nur dann vom PLC-Grundprogramm auf die PLC-Anforderungssignale DB21, ... DBX14.0 / .1 übertragen, wenn der FB1-Parameter **MMCToIF** auf **TRUE** gesetzt ist. Ist der Parameter nicht gesetzt, müssen die PLC-Anforderungssignale vom PLC-Anwenderprogramm gesetzt werden.

Verfahrbewegungen der Spindeln sperren

Sollen Spindeln während "Programmtest" **nicht** verfahren, müssen sie über das PLC-Anwenderprogramm explizit gesperrt werden:

- DB31, ... DBX128.0 = 0 (Programmtest unterdrücken)
- DB31, ... DBX128.1 = 1 (Programmtest aktivieren)
- DB31, ... DBX14.0 = 0 (von PLC: Programmtest unterdrücken)
- DB31, ... DBX14.1 = 1 (von PLC: Programmtest aktivieren)

Verfahrbewegungen von Achsen freigeben

Sollen Achsen des Kanals während "Programmtest" verfahren, müssen sie über das PLC-Anwenderprogramm explizit freigegeben werden:

- DB31, ... DBX14.0 = 1 (von PLC: Programmtest unterdrücken)
- DB31, ... DBX14.1 = 0 (von PLC: Programmtest aktivieren)

Erlaubte Schaltzeitpunkte

- Kanal
Die Nahtstellensignale zum Aus- / Einschalten des kanalspezifischen Zustandes "Programmtest" (DB21, ... DBX25.7 bzw. DBX1.7) dürfen nur im Kanalzustand "Reset" oder "Unterbrochen" geschaltet werden.
- Achsen / Spindel
Die Nahtstellensignale zum Aus- / Einschalten des achsspezifischen Zustandes "Programmtest" (DB31, ... DBX128.0 / .1 bzw. DBX14.0 / .1) dürfen immer geschaltet werden.

Programmstart und Programmablauf

Das Teileprogramm kann bei aktiver Programmtest-Funktion gestartet und abgearbeitet (inkl. Hilfsfunktionsausgaben, Wartezeiten, G-Befehlsausgaben etc.) werden über das Nahtstellensignal:

DB21, ... DBX7.1 (NC-Start)

Sicherheitsfunktionen wie Software-Endschalter, Arbeitsfeldbegrenzungen sind weiterhin gültig.

Die programmierten Geschwindigkeiten bleiben unverändert. Das bedeutet, dass die Positions- und Geschwindigkeitsangaben auf der Bedienoberfläche genau denen einer normalen Teileprogrammbearbeitung entsprechen. Die Lageregelung wird dabei nicht unterbrochen, sodass nach Abschalten der Funktion die Achsen nicht referenziert werden müssen.

Hinweis

Die Signale für Genauhalt DB31, ... DBX60.6/60.7 (Genauhalt grob/fein) spiegeln den tatsächlichen Zustand an der Maschine wieder.

Während des Programmtests würden sie nur weggenommen, wenn die Achse aus ihrer (während des Programmtests konstanten) Sollposition weggedrückt würde.

Mit dem Signal DB21, ... DBX33.7 (Programmtest aktiv) kann sowohl das PLC-Programm als auch das Teileprogramm mit der Variablen \$P_ISTEST entscheiden, wie während des Tests auf diese Signale zu reagieren oder zu verzweigen ist.

Hinweis

Probelaufvorschub

Die "Programmbearbeitung im Zustand Programmtest" kann auch zusammen mit der Funktion "Probelaufvorschub" aktiviert werden. Dadurch werden Teileprogrammabschnitte mit einem kleinen programmierten Vorschubwert in einem kürzeren Zeitabschnitt abgearbeitet.

Hinweis

Werkzeugverwaltung

Aufgrund der Achsensperre wird die Belegung eines Werkzeugmagazins beim Programmtest nicht verändert. Über eine PLC-Applikation muss sichergestellt werden, dass die Konsistenz zwischen den Daten der Werkzeugverwaltung und dem Magazin nicht verloren geht. Auf den Toolbox-Disketten finden Sie dazu beim PLC-Grundprogramm ein Beispiel.

2.5.2 Programmbearbeitung im Einzelsatzbetrieb

Funktion

Bei der "Programmbearbeitung im Einzelsatzbetrieb" stoppt die Teileprogrammbearbeitung nach jedem Programmsatz. Falls eine Fräser- oder Schneiden-Radiuskorrektur angewählt ist, so stoppt die Bearbeitung nach jedem von der Steuerung eingefügten Zwischensatz.

Der Programmzustand wechselt auf "Programmzustand angehalten".

Der Kanalzustand bleibt auf aktiv.

Mit NC-Start gelangt der nächste Teileprogrammsatz zur Abarbeitung.

Anwendung

Der Anwender kann damit ein Teileprogramm Satz für Satz abarbeiten und die einzelnen Bearbeitungsschritte kontrollieren. Wenn er den abgearbeiteten Teileprogrammsatz für korrekt befunden hat, kann er den nächsten Satz anfordern.

Einzelstypen

Es wird zwischen folgenden Einzelstypen unterschieden:

- **"SB1: Einzelsatz grob"** (Stopp nach jedem Satz mit Maschinenfunktion)
Das NC-Programm bzw. die Bearbeitung wird nach jedem vollständig abgearbeiteten Satz mit Maschinenfunktion angehalten bzw. gestoppt.
- **"SB2: Rechensatz"** (Stopp nach jedem Satz inklusive Rechensätzen und Kommentarsätzen)
Das NC-Programm bzw. die Bearbeitung wird nach jedem Satz angehalten bzw. gestoppt. Das Verhalten bei der Verwendung von Synchronaktionen ist als Randbedingung formuliert, Programmbeeinflussung (Seite 178)
- **"SB3: Einzelsatz fein"** (Stopp nach jedem Satz mit Maschinenfunktion auch in einem Zyklus)
Wie SB1, jedoch wird auch in Zyklen nach jedem Satz mit Maschinenfunktion angehalten.



VORSICHT

Funktionsbesonderheit bei Einzelsatztypen-Serie

Bei einer Serie von G33/G34/G35-Sätzen ist Einzelsatz nur dann wirksam, wenn "Probelaufvorschub" angewählt ist.

Rechensätze werden nicht im Einzelschritt bearbeitet (nur beim Dekodier-Einzelsatz).

SB2 ist auch bei G33/G34/G35 unwirksam.

Die Festlegung des Einzelstyps erfolgt über die Bedienoberfläche im Menü "Programmbeeinflussungen".

Anwahl

Die Anwahl des Einzelstypbetriebs ist möglich:

- Über die Maschinensteuertafel (Taste "Single Block")
- Über die Bedienoberfläche
Zur genauen Vorgehensweise siehe:
Weitere Informationen
Bedienungshandbuch der installierten HMI-Applikation

Aktivierung

Die Aktivierung der Funktion erfolgt durch das PLC-Grundprogramm über das Nahtstellensignal:

DB21, ... DBX0.4 (Einzelstyp aktivieren)

Anzeige

Als Rückmeldung des aktiven Einzelsatzbetriebs wird auf der Bedienoberfläche das entsprechende Feld in der Statuszeile invers geschaltet.

Sobald die Teileprogrammbearbeitung wegen des Einzelsatzbetriebs einen Teileprogrammsatz abgearbeitet hat, wird das folgende Nahtstellensignal gesetzt:

DB21, ... DBX35.3 (Programmzustand unterbrochen)

Abarbeiten ohne Einzelsatz-Stopp

Für bestimmte Programmabläufe kann trotz angewählten Einzelsatzbetriebs ein Abarbeiten ohne Einzelsatz-Stopp eingestellt werden, z. B. für:

- Interne ASUPs
- Anwender-ASUPs
- Zwischensätze
- Satzsuchlaufsammelsätze (Aktionssätze)
- Init-Sätze
- Unterprogramme mit `DISPLOF`
- Nicht-reorganisierbare Sätze
- Nicht-repositionierbare Sätze
- Wiederanfahrsatz ohne Verfahrinformation
- Werkzeuganfahrsatz

Die Einstellung erfolgt über das Maschinendatum:

MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK (Einzelsatz-Stopp verhindern)

Weitere Informationen

Listenhandbuch Ausführliche Beschreibung der Maschinendaten

2.5.3 Programmbearbeitung mit Probelaufvorschub

Funktion

Die Funktion "Programmbearbeitung mit Probelaufvorschub" dient zum Testen der Koordination des im Kanal angewählten NC-Programms mit z. B. dem PLC-Anwenderprogramm, externen Signalen und / oder anderen Kanälen der NC. Das NC-Programm wird dabei exakt, entsprechend den programmierten Befehlen, Funktionen und Verfahrgeschwindigkeiten abgearbeitet. Um während des Tests die Programmbearbeitungszeit zu verkürzen, können die Verfahrbewegungen durch Aktivierung des Probelaufvorschubs schneller ausgeführt werden. Statt des programmierten Vorschubs wirkt bei z. B. G01, G02, G03, G33, G34, G35, G95 dann der Probelaufvorschub.

2.5 Programmtest

Der Probelaufvorschub wird auf der Bedienoberfläche in den Betriebsarten AUTOMATIK und MDA angewählt und kann dann bei einer Automatikunterbrechung oder am Ende eines Satzes aktiviert werden. Die Aktivierung erfolgt vom PLC-Anwenderprogramm aus über die NC/PLC-Nahtstelle.

Weitere Informationen zur Vorschubbeeinflussung siehe Funktionshandbuch *Achsen und Spindeln* unter *Vorschübe*.

ACHTUNG
Zerstörung von Werkzeug oder Werkstück durch zu hohe Schnittgeschwindigkeiten
Wird bei aktiver Funktion "Programmbearbeitung mit Probelaufvorschub" eine Werkstückbearbeitung durchgeführt, können die sich dabei ergebenden Schnittgeschwindigkeiten außerhalb des zulässigen Bereichs liegen und zu Zerstörungen am Werkzeug und/oder Werkstück führen.

Parametrierung

Probelaufvorschub

Der im Kanal wirksame Probelaufvorschub wird eingestellt mit:

SD42100 \$SC_DRY_RUN_FEED = <Probelaufvorschub>

Hinweis

Umdrehungsvorschubs

Der Probelaufvorschubwert gilt auch anstelle des programmierten Umdrehungsvorschubs in Programmsätzen mit G95.

Probelaufeinstellungen

Das Auswahlkriterium für den bei "Programmbearbeitung mit Probelaufvorschub" im Kanal wirksamen Vorschub wird eingestellt mit:

SD42101 \$SC_DRY_RUN_FEED_MODE = <Wert>

<Wert >	Bedeutung
0	Als Probelaufvorschub wirkt das Maximum aus SD42100 und dem programmierten Vorschub.
1	Als Probelaufvorschub wirkt das Minimum aus SD42100 und dem programmierten Vorschub.
2	Als Probelaufvorschub wirkt SD42100.
10	Wie Wert 0, außer bei Gewindeschneiden (G33, G34, G35) und Gewindebohren (G331, G332, G63). Bei diesen Funktionen wirkt der programmierte Vorschub.
11	Wie Wert 1, außer bei Gewindeschneiden (G33, G34, G35) und Gewindebohren (G331, G332, G63). Bei diesen Funktionen wirkt der programmierte Vorschub.
12	Wie Wert 2, außer bei Gewindeschneiden (G33, G34, G35) und Gewindebohren (G331, G332, G63). Bei diesen Funktionen wirkt der programmierte Vorschub.

Aktivierung

Anwahl

Die Auswahl der Funktion erfolgt über die Bedienoberfläche über Bedienbereich "Maschine" > Betriebsart "AUTOMATIK" oder "MDA" > "Programmbeeinflussung" > "Probelaufvorschub".

Als Anforderung an das PLC-Anwenderprogramm die Funktion zu aktivieren wird folgendes NC/PLC-Nahtstellensignal gesetzt:

DB21, ... DBX24.6 = 1 (Probelaufvorschub angewählt)

Aktivierung

Zur Anforderung an die NC die Funktion zu aktivieren, muss das PLC-Anwenderprogramm folgendes NC/PLC-Nahtstellensignal setzen:

DB21, ... DBX0.6 = 1 (Probelaufvorschub aktivieren)

Rückmeldung

Als Rückmeldung für den Maschinenbediener, dass die Funktion in der NC aktiv ist, wird auf der Bedienoberfläche in der Statusanzeige des Bedienbereichs "Maschine" **DRY** angezeigt.

Weitere Informationen

Ausführliche Informationen zur Bedienung der Funktion "Programmbearbeitung mit Probelaufvorschub" finden sich in:

- Bedienhandbuch Drehen
- Bedienhandbuch Fräsen

Stichworte: Programmbeeinflussung, Probelaufvorschub, DRY

2.5.4 Teileprogrammätze ausblenden

Funktion

Beim Testen bzw. Einfahren neuer Programme ist es hilfreich, wenn bestimmte Teileprogrammätze für die Programmabarbeitung gesperrt bzw. ausgeblendet werden. Dazu müssen die betreffendenätze mit Schrägstrich gekennzeichnet werden.

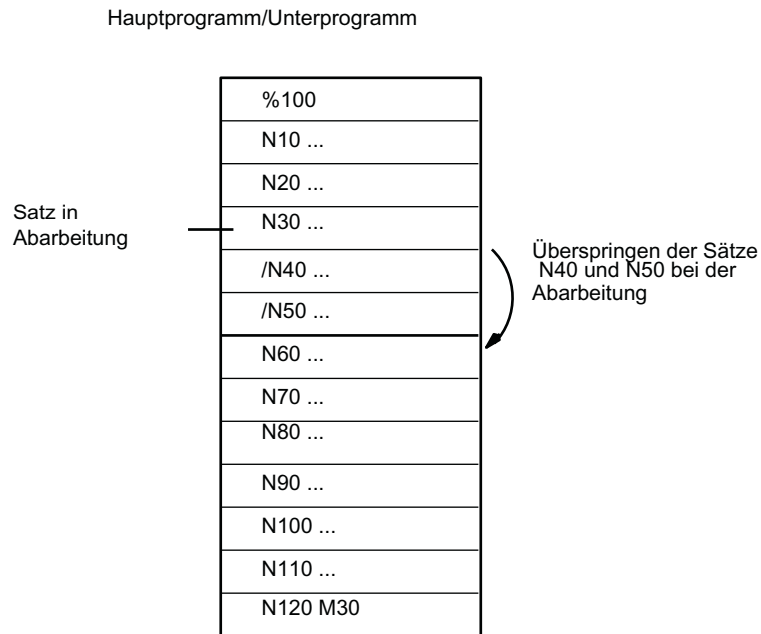


Bild 2-1 Ausblenden von Teileprogrammätzen

Anwahl

Die Anwahl der Funktion erfolgt über die Bedienoberfläche im Menü "Programmbeeinflussungen".

Mit der Anwahl wird folgendes Nahtstellensignal gesetzt:

DB21, ... DBX26.0 (Satz ausblenden angewählt)

Die Funktion wird damit noch nicht aktiviert.

Aktivierung

Die Aktivierung dieser Funktion erfolgt über das Nahtstellensignal:

DB21, ... DBX2.0 (Satz ausblenden aktivieren)

Hinweis

Die Funktion "Teileprogrammätze ausblenden" ist auch bei einem Satzsuchlauf aktiv.

Anzeige

Als Rückmeldung der aktivierten Funktion "Teileprogrammsätze ausblenden" wird auf der Bedienoberfläche das entsprechende Feld in der Statuszeile invers geschaltet.

2.6 Werkstücksimulation

Funktion

In der Werkstücksimulation wird das aktuelle Teileprogramm vollständig berechnet und das Ergebnis in der Bedienoberfläche grafisch dargestellt. Ohne die Maschinenachsen zu verfahren, wird so das Ergebnis der Programmierung kontrolliert. Falsch programmierte Bearbeitungsschritte werden frühzeitig erkannt und Fehlbearbeitungen am Werkstück verhindert.

Simulations-NC

Die Simulation nutzt eine eigene NC-Instanz (Simulations-NC). Deshalb muss vor einem Simulationsstart ein Abgleich des realen NC zum Simulations-NC erfolgen. Bei diesem Abgleich werden unter anderem sämtliche aktive Maschinendaten aus dem NC ausgelesen und in den Simulations-NC eingelesen. Unter den aktiven Maschinendaten befinden sich die NC- und Zyklenmaschinendaten.

Compile-Zyklen in Simulation

Bis SW 4.4 werden keine, ab SW 4.4 nur ausgesuchte Compile-Zyklen (CC) bei der Werkstücksimulation unterstützt. Die Maschinendaten der unterstützten Compile-Zyklen werden einmalig nach Steuerungshochlauf abgeglichen. Ein Abgleich mit "Simulationsstart" findet nicht statt!

Hinweis

In Teileprogrammen können CC-spezifische Sprachbefehle und Maschinendaten von nicht unterstützten CCs **nicht** benutzt werden (siehe auch Absatz "CC-Befehle im Teileprogramm").

Sonderbewegungen von unterstützten CCs (OEM-Transformationen) werden unter Umständen falsch dargestellt.

CC-Befehle im Teileprogramm

Sprachbefehle im Teileprogramm von nicht unterstützten Compile-Zyklen (OMA1 ... OMA5, OEMIPO1/2, G810 ... G829, eigene Prozeduren und Funktionen) führen ohne individuelle Behandlung zur Alarmmeldung und zum Abbruch der Simulation.

Lösung: Die fehlenden CC-spezifischen Sprachelemente im Teileprogramm individuell behandeln (\$P_SIM Abfrage). Beispiel:

Programmcode	Kommentar
N1 G01 X200 F500	

2.7 Satzsuchlauf Typ 1, 2 und 4

Programmcode	Kommentar
IF (1==P_SIM)	
N5 X300	; Bei Simulation CC nicht aktiv.
ELSE	
N5 X300 OMA1=10	
ENDIF	

2.7 Satzsuchlauf Typ 1, 2 und 4

Funktion

Satzsuchlauf bietet die Möglichkeit, die Abarbeitung eines Teileprogramms von einem nahezu beliebigen Teileprogrammsatz aus zu beginnen.

Beim Satzsuchlauf erfolgt ein NC-interner Schnelldurchlauf ohne Verfahrbewegungen durch das Teileprogramm bis zum gewählten Zielsatz. Dabei wird versucht, möglichst exakt den Steuerungszustand zu erzielen, wie er sich am Zielsatz bei der normalen Teileprogrammabarbeitung (z. B. bezüglich Achspositionen, Spindeldrehzahlen, eingewechselte Werkzeuge, NC/PLC-Nahtstellensignale, Variablenwerte) ergibt, um mit möglichst wenigen manuellen Eingriffen ab dem Zielsatz mit der automatischen Teileprogrammabarbeitung fortfahren zu können.

Satzsuchlauf Typen

- **Typ 1: Satzsuchlauf ohne Berechnung**
Satzsuchlauf ohne Berechnung dient zum schnellstmöglichen Auffinden eines Teileprogrammsatzes. Es wird dabei keinerlei Berechnung durchgeführt. Der Steuerungszustand am Zielsatz entspricht unverändert dem vor dem Start des Satzsuchlaufs.
- **Typ 2: Satzsuchlauf mit Berechnung an Kontur**
Satzsuchlauf mit Berechnung an Kontur dient dazu, um in beliebigen Situationen an die programmierte Kontur anfahren zu können. Mit NC-START wird die Anfangsposition des Zielsatzes bzw. die Endposition des Satzes vor dem Zielsatz angefahren. Dieser wird bis zur Endposition abgefahren. Die Abarbeitung erfolgt konturtreu.

- **Typ 4: Satzsuchlauf mit Berechnung an Satzendpunkt**
Satzsuchlauf mit Berechnung an Satzendpunkt dient dazu, um in beliebigen Situationen eine Zielposition (z. B. Werkzeugwechselposition) anfahren zu können. Angefahren wird der Endpunkt des Zielsatzes bzw. die nächste programmierte Position unter Verwendung der im Zielsatz gültigen Interpolationsart. Dies geschieht nicht konturtreu.
Es werden nur die im Zielsatz programmierten Achsen gefahren. Gegebenenfalls muss an der Maschine vor dem Start der weiteren automatischen Teileprogrammbearbeitung manuell über die Betriebsart "JOG-REPOS" eine kollisionsfreie Ausgangssituation geschaffen werden.
- **Typ 5: Satzsuchlauf mit Berechnung im Modus "Programmtest" (SERUPRO)**
SERUPRO (search run by programtest) ist ein Kanal-übergreifender Satzsuchlauf mit Berechnung. Die NC startet dabei das angewählte Teileprogramm im Modus "Programmtest". Mit Erreichen des Zielsatzes wird der Programmtest automatisch wieder abgewählt. Bei diesem Satzsuchlauf sind auch Interaktionen zwischen dem Kanal, in dem der Satzsuchlauf stattfindet, und Synchronaktionen und anderen Kanälen der NC möglich.
Siehe Kapitel "Satzsuchlauf Typ 5 (SERUPRO) (Seite 70)".

Hinweis

Weitere Erläuterungen zum Satzsuchlauf siehe Kapitel "Verhalten bei Satzsuchlauf (Seite 689)".

Folgeaktionen

Nach dem Abschluss eines Satzsuchlaufs können folgende Folgeaktionen erfolgen:

- Typ1 - Typ 5: Automatischer Start eines ASUPS
Mit dem Einwechseln des letzten Aktionssatzes kann ein Anwenderprogramm als ASUP gestartet werden.
- Typ1 - Typ 4: Kaskadierter Satzsuchlauf
Aus dem Zustand "Suchziel gefunden" kann ein weiterer Satzsuchlauf mit geänderter Suchzielangabe gestartet werden.

2.7.1 Funktionsbeschreibung

Prinzipieller Ablauf für Typ 2 oder Typ 4

1. Anwender: Aktivierung des Satzsuchlaufs **Typ 2** oder **Typ 4** (Satzsuchlauf mit **Berechnung** an ...) über die Bedienoberfläche
2. Suchen des Zielsatzes mit Aufsammeln der Hilfsfunktionen
3. Stopp nach "Suchziel gefunden" ⇒ Anzeige des Alarms 10208 "Zur Programmf Fortsetzung NC-Start geben"
4. Anwender: NC-Start zum Abarbeiten der Aktionssätze ⇒ DB21, ... DBX7.1 = 1 (NC-Start)
5. Abarbeiten der Aktionssätze

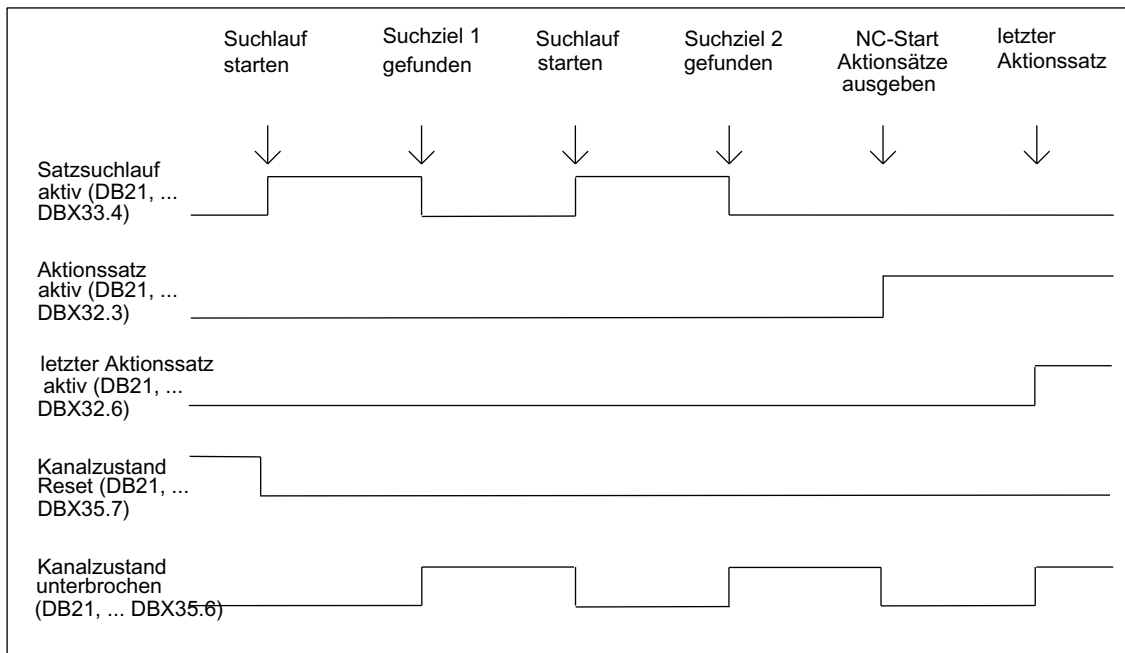
2.7 Satzsuchlauf Typ 1, 2 und 4

6. Letzter Aktionssatz wird eingewechselt ⇒ Automatischer Start von /_N_CMA_DIR/_N_PROG_EVENT_SPF (Voreinstellung) als ASUP
7. Letzter ASUP-Satz (REPOSA) wird eingewechselt ⇒ DB21, ... DBX32.6 = 1 (letzter Aktionssatz aktiv)
8. **Optional:** Abarbeiten der anwenderspezifischen Anforderungen über PLC-Anwenderprogramm
9. Anzeige des Alarms 10208 "Zur Programmfortsetzung NC-Start geben"
10. Anwender: NC-Start zur Programmfortsetzung ⇒ DB21, ... DBX7.1 = 1 (NC-Start)

Suchziel nicht gefunden

Wird das Suchziel nicht gefunden, wird der Alarm 15370 "Suchziel bei Satzsuchlauf nicht gefunden" angezeigt und der Satzsuchlauf abgebrochen.

Zeitlicher Ablauf



Aktionssätze

Während eines Satzsuchlaufs **Typ 2** oder **Typ 4** (Satzsuchlauf mit **Berechnung** an ...) werden Aktionen wie z. B. Werkzeug- (T, D), Spindel- (S), Vorschub-Programmierung oder M-Funktionsausgaben aufgesammelt. Mit NC-Start zum Abarbeiten der Aktionssätze werden die aufgesammelten Aktionen an die PLC ausgegeben.

Hinweis

Mit NC-Start für die Aktionssätze, wird die während Satzsuchlauf **Typ 2** oder **Typ 4** (Satzsuchlauf mit **Berechnung** an ...) aufgesammelte Spindelprogrammierung (S-Wert, M3 / M4 / M5 / M19, SPOS) **aktiv**.

Der Anwender muss über das PLC-Anwenderprogramm sicherstellen, dass das Werkzeug betrieben werden kann oder die Spindelprogrammierung zurückgesetzt bzw. nicht ausgegeben wird:

DB31, ... DBX2.2 = 1 (Spindelreset) .

Randbedingungen

Aufsetzmodus nach Satzsuchlauf Typ 4

Erfolgt die erste Programmierung einer Achse nach einem Satzsuchlauf Typ 4 (Satzsuchlauf mit Berechnung an Satzendpunkt) inkrementell, kann der programmierte inkrementelle Wert auf den bis zum Suchziel aufgesammelten Positionswert oder auf den aktuellen Istwert der Achse addiert werden. Die Einstellung erfolgt über:

SD42444 \$SC_TARGET_BLOCK_INCR_PROG

Das Settingdatum wird mit NC-Start für die Ausgabe der Aktionssätze ausgewertet.

Einzelatz

Soll, nachdem das Suchziel bei Satzsuchlauf Typ 2 oder Typ 4 (Satzsuchlauf mit **Berechnung** an ...) gefunden wurde und aktiver Funktion "Einzelatz" (DB21, ... DBX0.4 == 1 (Einzelatz aktivieren)), nicht nach jedem Aktionssatz angehalten werden, kann dieses Verhalten über das Maschinendatum ausgeschaltet werden:

MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK, Bit 3 = 1 (Einzelatzunterdrückung bei Aktionssätzen)

Nahtstellensignal "Anfahratz aktiv"

Das Nahtstellensignal wird **nur** bei Satzsuchlauf **Typ 2** (Satzsuchlauf mit **Berechnung an Kontur**) gesetzt:

- DB21, ... DBX32.4 = 1 (Anfahratz aktiv)

Bei Satzsuchlauf **Typ 4** (Satzsuchlauf mit **Berechnung an Satzendpunkt**) wird das Nahtstellensignal nicht gesetzt, da hier kein Anfahratz erzeugt wird (Anfahratz gleich Zielsatz).

Interpolationsart des Zielsatzes

Bei Satzsuchlauf **Typ 4** (Satzsuchlauf mit **Berechnung an Satzendpunkt**) wird die Anfahrbewegung in der im Zielsatz gültigen Interpolationsart durchgeführt. Bei anderen

Interpolationsarten als Linearinterpolation (G_0 oder G_1), kann die Anfahrbewegung mit Alarm abgebrochen werden (z. B. Kreisendpunktfehler bei Kreisinterpolation G_2 oder G_3).

2.7.2 Satzsuchlauf im Zusammenhang mit weiteren NCK-Funktionen

2.7.2.1 ASUP nach und bei Satzsuchlauf

Satzsuchlauf Typ 2 und Typ 4: Synchronisation der Kanalachsen

Wenn nach einem Satzsuchlauf Typ 2 oder Typ 4 (Satzsuchlauf mit Berechnung an ...) ein ASUP gestartet wird, werden die Istpositionen aller Kanalachsen synchronisiert.

Auswirkung

Werden die nachfolgenden Systemvariablen im ASUP gelesen, enthalten sie folgende Werte:

- \$P_EP: Aktuelle Istposition der Kanalachse im WKS
- \$AC_RETPOINT: Aufgesammelte Satzsuchlaufposition der Kanalachse im WKS

Satzsuchlauf Typ 2: ASUP Abschluss

Bei Satzsuchlauf Typ 2 (Satzsuchlauf mit Berechnung an Kontur) muss zum Abschluss des ASUP der nachfolgende Befehl `REPOSA` (Wiederanfahren an die Kontur; linear; alle Kanalachsen) programmiert werden.

Auswirkung

- Alle Kanalachsen werden auf ihre im Satzsuchlauf aufgesammelte Suchlaufposition verfahren.
- \$P_EP == "aufgesammelte Satzsuchlaufposition der Kanalachse (WKS)"

Satzsuchlauf Typ 4: REPOS-Verhalten

Nach Satzsuchlauf Typ 4 (Satzsuchlauf mit Berechnung an Satzendpunkt) wird während des durch **Beginn** und **Ende** beschriebenen Zeitraums durch den Befehl `REPOS` kein automatisches Repositionieren ausgelöst:

- **Beginn**: NC/PLC-Nahtstellensignal `DB21,... DBX32.6 == 1` (letzter Aktionssatz aktiv)
- **Ende**: Fortsetzen der Programmbearbeitung mit NC-Start.

Der Startpunkt der Anfahrbewegung sind die aktuellen Positionen der Kanalachsen zum Zeitpunkt zu dem NC-Start gegeben wird. Der Endpunkt ergibt sich durch die weiteren im Teileprogramm programmierten Verfahrbewegungen.

Bei Satzsuchlauf Typ 4 wird durch die NC keine Anfahrbewegung erzeugt.

Auswirkung:

- Die Systemvariable \$P_EP (programmierte Endposition) enthält nach Verlassen des ASUP die Istposition, auf die die Kanalachsen vom ASUP oder manuell (Betriebsart: JOG) positioniert wurden.
\$P_EP == "aktuelle Istposition der Kanalachse im WKS"

2.7.2.2 PLC-Aktionen nach Satzsuchlauf

Wenn alle Aktionssätze seitens der NC abgearbeitet sind und Aktionen seitens der PLC, z.B. Start eines ASUP um einen Werkzeugwechsel durchzuführen, oder des Bedieners, z.B. Überspeichern, möglich sind, wird folgendes kanalspezifisches NC/PLC-Nahtstellensignal gesetzt:

- DB21, ... DB32.6 = 1 (letzter Aktionssatz aktiv)

Zeitpunkt der Alarmausgabe parametrieren

Um den Bediener darauf hinzuweisen, dass zur Fortsetzung der Programmbearbeitung noch NC-Start im Kanal erforderlich ist, wird Alarm 10208 "Zur Programmfortsetzung NC-Start geben" angezeigt.

Wann der Alarm angezeigt wird, ist über das Maschinendatum einstellbar:
MD11450 \$MN_SEARCH_RUN_MODE, Bit 0 = <Wert>

<Wert>	Bedeutung
0	Mit dem Einwechseln des letzten Aktionssatzes nach Satzsuchlauf erfolgt: <ul style="list-style-type: none"> Die Bearbeitung des Teileprogramms wird gestoppt DB21, ... DBB32.6 = 1 (letzter Aktionssatz aktiv) Alarm 10208 anzeigen
1	Mit dem Einwechseln des letzten Aktionssatzes nach Satzsuchlauf erfolgt: <ul style="list-style-type: none"> Die Bearbeitung des Teileprogramms wird gestoppt DB21, ... DBB32.6 = 1 (letzter Aktionssatz aktiv) Alarm 10208 erst anzeigen, wenn DB21, ... DBX1.6 == 1 (PLC-Aktion beendet)

Zusammen mit dem Alarm werden folgende Nahtstellensignale gesetzt:

- DB21, ... DBX36.7 = 1 (NC-Alarm mit Bearbeitungsstillstand steht an)
- DB21, ... DBX36.6 = 1 (NC-Alarm kanalspezifisch steht an)

2.7.2.3 Spindelfunktionen nach Satzsuchlauf**Steuerungsverhalten und Ausgabe**

Ob die beim Satzsuchlauf aufgesammelten Spindel-spezifischen Hilfsfunktionen automatisch im Aktionssatz oder anwenderspezifisch zu einem späteren Zeitpunkt an die PLC ausgegeben werden, ist einstellbar über

MD11450 \$MN_SEARCH_RUN_MODE, Bit 2 = <Wert>

<Wert>	Bedeutung
0	Die Ausgabe der bei Satzsuchlauf aufgesammelten Spindel-spezifischen Hilfsfunktionen (M3, M4, M5, M19, M70) erfolgt im Aktionssatz.
1	Die bei Satzsuchlauf aufgesammelten Spindel-spezifischen Hilfsfunktionen werden im Aktionssatz nicht ausgegeben. Die Ausgabe kann anwenderspezifisch zu einem späteren Zeitpunkt, z.B. in einem ASUP, vorgenommen werden. Die aufgesammelten Spindel-spezifischen Hilfsfunktionen werden dazu in Systemvariablen gespeichert:

Systemvariablen

Die beim Satzsuchlauf aufgesammelten Spindel-spezifischen Hilfsfunktionen werden in folgenden Systemvariablen gespeichert:

Systemvariable	Beschreibung
\$P_SEARCH_S [<n>]	Zuletzt programmierte Spindeldrehzahl bzw. Schnittgeschwindigkeit
\$P_SEARCH_SDIR [<n>]	Zuletzt programmierte Spindeldrehrichtung
\$P_SEARCH_SGEAR [<n>]	Zuletzt programmierte Getriebestufen M-Funktion
\$P_SEARCH_SMODE [<n>]	Zuletzt programmierte Spindelbetriebsart
\$P_SEARCH_SPOS [<n>]	Zuletzt mittels M19, SPOS bzw. SPOSA programmierte Spindelposition bzw. Verfahrenweg
\$P_SEARCH_SPOSMODE [<n>]	Zuletzt mittels M19, SPOS bzw. SPOSA programmierter Positionsanfahrmode
<n>: Spindelnummer	

Zur späteren Ausgabe der Spindel-spezifischen Hilfsfunktionen können die Systemvariablen z.B. in einem ASUP gelesen und nach Ausgabe der Aktionssätze ausgegeben werden:

DB21, ... DBX32.6 == 1 (Letzter Aktionssatz aktiv)

Hinweis

Die Inhalte der Systemvariablen \$P_S, \$P_DIR und \$P_SGEAR können nach Satzsuchlauf durch Synchronisationsvorgänge verloren gehen.

Für weiterführende Informationen zu ASUP, Satzsuchlauf und Aktionssätzen siehe Kapitel "Ausgabeunterdrückung von Spindel-spezifischen Hilfsfunktionen (Seite 694)" und "Programmtest (Seite 46)".

2.7.2.4 Lesen von Systemvariablen bei Satzsuchlauf

In Teilprogrammen können über Systemvariable Werte aus den Bereichen Vorlauf, Hauptlauf oder Servo/Antrieb gelesen werden:

\$P_...	Vorlauf-bezogene Systemvariablen, enthalten programmierte Werte
\$A_...	Hauptlauf-bezogene Systemvariable,n enthalten aktuelle Werte
\$V_...	Servo/Antrieb-bezogene Systemvariablen, enthalten aktuelle Werte

Da während eines Satzsuchlaufs vom Typ 2 und Typ 4 (Satzsuchlauf **mit Berechnung** an ...) keine Sätze in den Hauptlauf gelangen, werden Hauptlauf- und Servo/Antrieb-bezogene Systemvariablen während des Satzsuchlaufs nicht verändert. Gegebenenfalls muss für diese Systemvariablen durch Abfrage ob ein Satzsuchlauf aktiv ist \$P_SEARCH (Satzsuchlauf aktiv) im NC-Programm eine Sonderbehandlung für Satzsuchlauf durchgeführt werden.

Vorlauf-bezogene Systemvariable liefern in allen Suchlaufarten korrekte Werte.

2.7.3 Automatischer Start eines ASUP nach Satzsuchlauf

Parametrierung

Wirksamsetzen der Funktion

Der automatische ASUP-Start nach Satzsuchlauf wird durch folgende MD-Einstellung aktiviert:

MD11450 \$MN_SEARCH_RUN_MODE, Bit 1 = 1

Zu aktivierendes Programm

In der Grundstellung wird nach dem Satzsuchlauf mit dem Einwechseln des letzten Aktionssatzes das Programm **_N_PROG_EVENT_SPF** aus dem Verzeichnis **_N_CMA_DIR** als ASUP aktiviert. Soll ein anderes Programm aktiviert werden, muss im folgenden Maschinendatum der Name dieses Anwenderprogramms eingetragen sein:

MD11620 \$MN_PROG_EVENT_NAME

Verhalten bei gesetzter Einzelsatzbearbeitung

Über das kanalspezifische Maschinendatum wird eingestellt, ob das aktivierte ASUP trotz gesetzter Einzelsatzbearbeitung ohne Unterbrechung abgearbeitet wird, oder die Einzelsatzbearbeitung wirksam sein soll:

MD20106 \$MC_PROG_EVENT_IGN_SINGLEBLOCK, Bit 4 = <Wert>

<Wert>	Bedeutung
0	Einzelsatzbearbeitung ist wirksam.
1	Einzelsatzbearbeitung wird unterdrückt.

Verhalten bei gesetzter Einlesesperre

Über das kanalspezifische Maschinendatum wird eingestellt, ob das ASUP trotz gesetzter Einlesesperre (DB21, ... DBX6.1 == 1) komplett abgearbeitet wird, oder die Einlesesperre wirksam sein soll:

MD20107 \$MC_PROG_EVENT_IGN_INHIBIT, Bit 4 = <Wert>

<Wert>	Bedeutung
0	Einlesesperre ist wirksam.
1	Einlesesperre wird unterdrückt.

Hinweis

Weitere Informationen zur Parametrierung von MD11620, MD20108 und MD20107 siehe Kapitel "Parametrierung (Seite 130)".

Programmierung

Durch welches Ereignis das ASUP gestartet wurde, ist in der Systemvariablen \$P_PROG_EVENT hinterlegt. Bei automatischer Aktivierung nach Satzsuchlauf liefert \$P_PROG_EVENT den Wert "5".

Ablauf

Ablauf zum automatischen Start eines ASUP nach Satzsuchlauf:

1. Anwender: Aktivierung des Satzsuchlaufs **Typ 2** oder **Typ 4** (Satzsuchlauf **mit Berechnung** an ...) über die Bedienoberfläche
2. Suchen des Zielsatzes mit Aufsammeln der Hilfsfunktionen
3. Stopp nach "Suchziel gefunden ⇒ Anzeige des Alarms 10208 "Zur Programmfortsetzung NC-Start geben"
4. Anwender: NC-Start zum Abarbeiten der Aktionssätze ⇒ DB21, ... DBX7.1 = 1 (NC-Start)
5. Abarbeiten der Aktionssätze
6. Letzter Aktionssatz wird eingewechselt ⇒ Automatischer Start von /_N_CMA_DIR/_N_PROG_EVENT_SPF (Voreinstellung) als ASUP
7. Letzter ASUP-Satz (REPOSA) wird eingewechselt ⇒ DB21, ... DBX32.6 = 1 (letzter Aktionssatz aktiv)
8. **Optional:** Abarbeiten der anwenderspezifischen Anforderungen über PLC-Anwenderprogramm
9. Anzeige des Alarms 10208 "Zur Programmfortsetzung NC-Start geben"

Hinweis

Mit MD11450 \$MN_SEARCH_RUN_MODE, Bit 0 == 1, wird Alarm 10208 erst nach Freigabe durch das PLC-Anwenderprogramm (DB21, ... DBX1.6 = 1 (PLC-Aktion beendet)) ausgegeben.

10. Anwender: NC-Start zur Programmfortsetzung ⇒ DB21, ... DBX7.1 = 1 (NC-Start)

2.7.4 Kaskadierter Satzsuchlauf

Funktionalität

Mit der Funktion "Kaskadierter Satzsuchlauf" ist es möglich aus dem Zustand "Suchziel gefunden" einen weiteren Satzsuchlauf zu starten. Die Kaskadierung kann nach jedem gefundenen Suchziel beliebig oft fortgesetzt werden und ist für folgende Satzsuchlauffunktionen anwendbar:

- Satzsuchlauf Typ 1 ohne Berechnung
- Satzsuchlauf Typ 2 mit Berechnung an Kontur
- Satzsuchlauf Typ 3 mit Berechnung an Satzendpunkt

Hinweis

Nur wenn das Suchziel gefunden wurde, kann aus der gestoppten Programmbearbeitung ein weiterer "Kaskadierter Satzsuchlauf" gestartet werden.

Aktivierung

Der "Kaskadierte Satzsuchlauf" wird im bestehenden Maschinendatum projiziert:
MD11450 \$MN_SEARCH_RUN_MODE

- Mit Bit 3 = 0 (FALSE) wird der kaskadierte Satzsuchlauf frei geschaltet (d. h. mehrfache Suchzielvorgaben sind möglich).
- Aus Kompatibilitätsgründen kann der kaskadierte Satzsuchlauf mit Bit 3 = 1 (TRUE) gesperrt werden. Voreingestellt ist der kaskadierte Satzsuchlauf Bit 3 = 0.

Ablaufverhalten

Suchziel gefunden, Satzsuchlauf erneut starten

Mit dem Erreichen des Suchziels wird die Programmbearbeitung gestoppt und das Suchziel als aktueller Satz angezeigt. Nach jedem gefundenen Suchziel ist ein neuer Satzsuchlauf beliebig oft wiederholbar.

Suchzielvorgaben ändern

Vor jeden Satzsuchlaufstart können Suchzielangabe und Satzsuchlauffunktion geändert werden.

Beispiel: Ablauf einer Bearbeitungssequenz mit kaskadiertem Satzsuchlauf

- RESET
- Satzsuchlauf bis Suchziel 1
- Satzsuchlauf bis Suchziel 2 → "Kaskadierter Satzsuchlauf"

2.7 Satzsuchlauf Typ 1, 2 und 4

- NC-Start für Ausgabe der Aktionssätze → Alarm 10208
- NC-Start → Fortsetzen der Programmbearbeitung

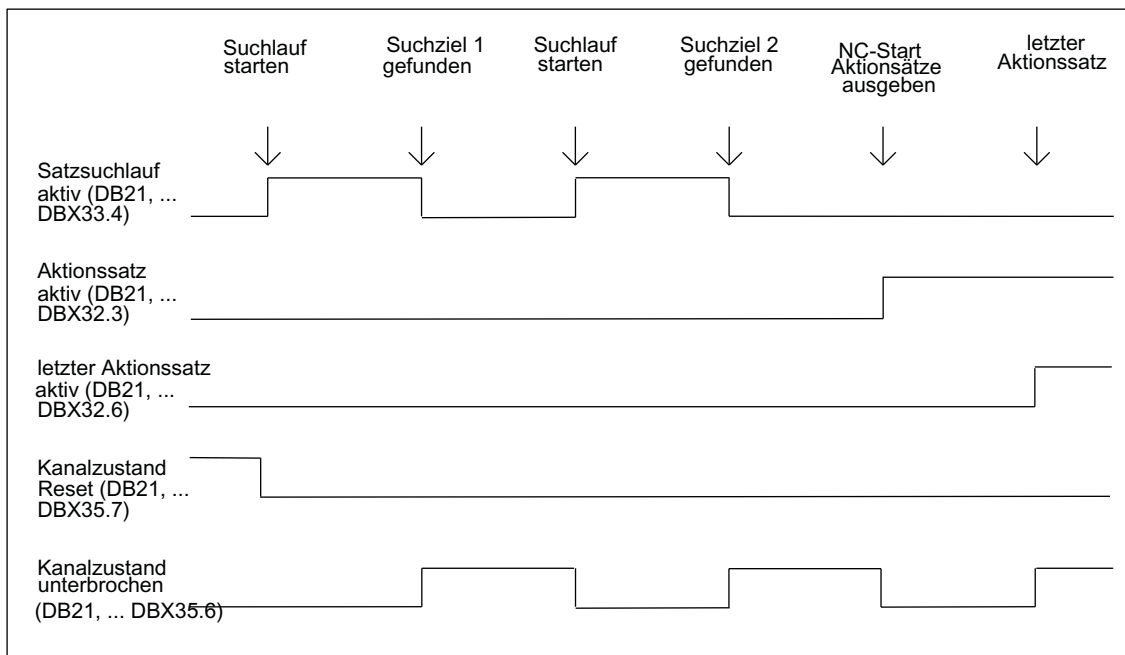


Bild 2-2 Zeitlicher Ablauf Nahtstellensignale

2.7.5 Beispiele zum Satzsuchlauf mit Berechnung

Auswahl

Wählen Sie aus den folgenden Beispielen den Satzsuchlauf typ aus, der Ihrer Problemstellung am Besten entspricht.

Satzsuchlauf Typ 4 mit Berechnung an Satzendpunkt

Beispiel mit automatischem Werkzeugwechsel nach Satzsuchlauf bei aktiver Werkzeugverwaltung:

1. Maschinendaten setzen:
MD11450 \$MN_SEARCH_RUN_MODE auf 1
MD11602 \$MN_ASUP_START_MASK Bit 0 = 1 (ASUP-Start aus gestopptem Zustand)
2. ASUP "SUCHLAUF_ENDE" von PLC über FB4 anwählen (siehe auch Funktionshandbuch "PLC und Grundprogramm", Kapitel "PLC-Grundprogramm").
3. Teileprogramm "WERKSTUECK_1" laden und anwählen.
4. Satzsuchlauf auf Satzendpunkt Satznummer N220.
5. HMI meldet "Suchziel gefunden".
6. NC-Start für die Ausgabe der Aktionssätze.

7. Mit dem PLC-Signal:
DB21... DB32.6 (letzter Aktionssatz aktiv)
startet die PLC über FC9 das ASUP "SUCHLAUF_ENDE" (siehe auch Funktionshandbuch "PLC und Grundprogramm", Kapitel "PLC-Grundprogramm").
8. Nach ASUP-Ende (auswertbar z. B. über die zu definierende M-Funktion M90, siehe Beispiel Satz N1110) setzt die PLC das Signal:
DB21, ... DBX1.6 (PLC-Aktion beendet).
Alternativ kann auch das NC/PLC-Nahtstellensignal:
DB21-DB30 DBB318 Bit 0 (ASUP ist angehalten)
abgefragt werden.
Dadurch wird der Alarm 10208 angezeigt, d. h. es können jetzt weitere Aktionen seitens des Bedieners erfolgen.
9. Manuelle Eingriffe durch Bediener (JOG, JOG-REPOS, Überspeichern).
10. Teileprogramm mit NC-Start fortsetzen.

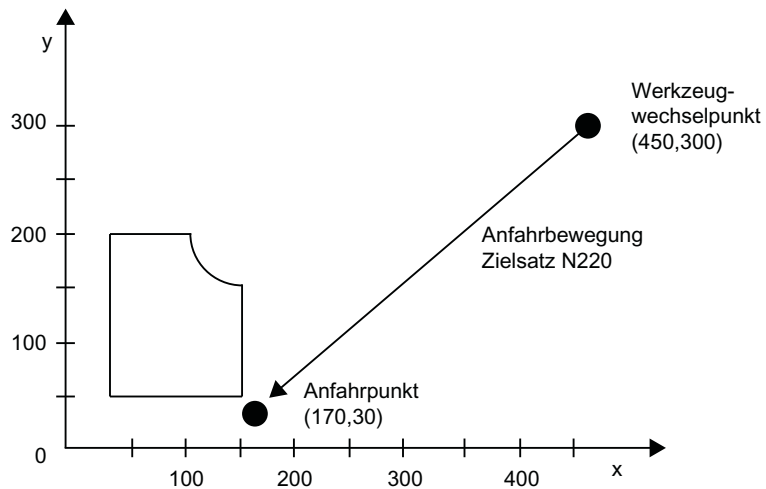


Bild 2-3 Anfahrbewegung bei Satzsuchlauf auf Satzendpunkt (Zielsatz N220)

Hinweis

"Satzsuchlauf an Kontur" mit Zielsatz N220 würde eine Anfahrbewegung zum Werkzeugwechsellpunkt (Startpunkt des Zielsatzes) erzeugen.

Satzsuchlauf Typ 2 mit Berechnung an Kontur

Beispiel mit automatischem Werkzeugwechsel nach Satzsuchlauf bei aktiver Werkzeugverwaltung:

1. bis 3. wie Beispiel zum Satzsuchlauf Typ 4
4. Satzsuchlauf an Kontur Satznummer N260
5. bis 10. wie Beispiel zum Satzsuchlauf Typ 4

2.7 Satzsuchlauf Typ 1, 2 und 4

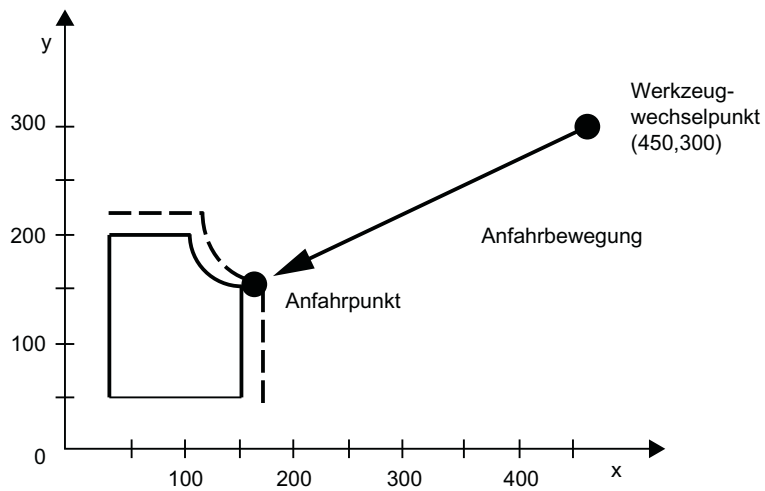


Bild 2-4 Anfahrbewegung bei Satzsuchlauf an Kontur (Zielsatz N260)

Hinweis

"Satzsuchlauf auf Satzendpunkt " mit Zielsatz N260 würde zu Alarm 14040 (Kreisendpunktfehler) führen.

Teileprogramme für Typ 4 und Typ 2

PROC WERKSTUECK_1

Programmcode	Kommentar
;Hauptprogramm	
...	
;Bearbeitung Konturabschnitt 1 mit Werkzeug "FRAESER_1"	
...	
N100 G0 G40 X200 Y200	; Abwahl Radiuskorrektur
N110 Z100 D0	; Abwahl Längenkorrektur
;Ende Konturabschnitt 1	
;	
;Bearbeitung Konturabschnitt 2 mit Werkzeug "FRAESER_2"	
N200 T="FRAESER_2"	; Werkzeug vorwählen
N210 WZW	; Werkzeugwechselroutine aufrufen
N220 G0 X170 Y30 Z10 S3000 M3 D1	; Anfahransatz Konturabschnitt 2
N230 Z-5	; Zustellung
N240 G1 G64 G42 F500 X150 Y50	; Startpunkt der Kontur
N250 Y150	
N260 G2 J50 X100 Y200	
N270 G1 X50	
N280 Y50	
N290 X150	
N300 G0 G40 G60 X170 Y30	; Abwahl Radiuskorrektur

Programmcode	Kommentar
N310 Z100 D0	; Abwahl Längenkorrektur
Ende Konturabschnitt 2	
...	
M30	
PROC WZW	
;Werkzeugwechselroutine	
N500 DEF INT TNR_AKTIV	; Variable für aktive T-Nummer
N510 DEF INT TNR_VORWAHL	; Variable für vorgewählte T-Nummer
N520 TNR_AKTIV = \$TC_MPP6[9998,1]	; T-Nummer des aktiven Werkzeugs lesen
N530 GETSELT(TNR_VORWAHL)	; T-Nummer des vorgewählten Werkzeugs lesen
;	
;Werkzeugwechsel nur ausführen, wenn Werkzeug noch nicht aktiv ist	
N540 IF TNR_AKTIV == TNR_VORWAHL GOTOF ENDE	
N550 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	; Werkzeugwechsellpunkt anfahren
N560 M6	; Werkzeugwechsel ausführen
;	
ENDE: M17	
PROC SUCHLAUF_ENDE SAVE	
;ASUP für den Aufruf der Werkzeugwechselroutine nach Satzsuchlauf	
N1000 DEF INT TNR_AKTIV	; Variable für aktive T-Nummer
N1010 DEF INT TNR_VORWAHL	; Variable für vorgewählte T-Nummer
N1020 DEF INT TNR_SUCHLAUF	; Variable für im Satzsuchlauf ermittelte T-Nummer
N1030 TNR_AKTIV = \$TC_MPP6[9998,1]	; T-Nummer des aktiven Werkzeugs lesen
N1040 TNR_SUCHLAUF = \$P_TOOLNO	; durch Satzsuchlauf ermittelte T-Nummer lesen
N1050 GETSELT(TNR_VORWAHL)	; T-Nummer des vorgewählten Werkzeugs lesen
N1060 IF TNR_AKTIV ==TNR_SUCHLAUF GOTOF ASUP_ENDE	
N1070 T = \$TC_TP2[TNR_SUCHLAUF]	; T-Anwahl über Werkzeugnamen
N1080 WZW	; Werkzeugwechselroutine aufrufen
N1090 IF TNR_VORWAHL == TNR_SUCHLAUF GOTOF ASUP_ENDE	
N1100 T = \$TC_TP2[TNR_VORWAHL]	; T-Vorwahl restaurieren über Werkzeugnamen
ASUP_ENDE:	
N1110 M90	; Rückmeldung an PLC
N1120 REPOSA	; ASUP Ende

2.7.6 Randbedingungen

2.7.6.1 Kompressor-Funktionen (COMPON, COMPCURV, COMPCAD, COMPSURF)

- Wenn bei Satzsuchlauf **Typ 2** oder **Typ 4** (Satzsuchlauf **mit Berechnung** an ...) der Zielsatz in einem Programmabschnitt liegt, in dem eine Kompressor-Funktion (COMPON, COMPCURV, COMPCAD, COMPSURF) aktiv ist, werden beim Wiederanfahren an die Kontur Positionen auf der durch den **Kompressor** berechneten Bahn angefahren. Diese Positionen müssen nicht exakt mit den Positionen auf der im Teileprogramm programmierten Bahn übereinstimmen.
- Wenn bei der Komprimierung im Teileprogramm programmierte Sätze entfallen, werden diese Sätze im Satzsuchlauf nicht als Zielsatz gefunden. Es wird Alarm 15370 "Suchziel bei Satzsuchlauf nicht gefunden" ausgegeben.

2.8 Satzsuchlauf Typ 5 (SERUPRO)

2.8.1 Funktionsbeschreibung

Der Suchlauftyp 5, Satzsuchlauf mit Berechnung im Modus "Programmtest" (SERUPRO, "**Search-Run** by **Programtest**") ermöglicht einen kanalübergreifenden Satzsuchlauf mit Berechnung zu einem wählbaren Unterbrechungspunkt. Dazu werden während SERUPRO, unter Beachtung vorhandener Programmkoordinierungsbefehle, alle zur Programmfortsetzung in den unterbrochenen Kanälen erforderlichen Zustandsdaten ermittelt und anschließend NC und PLC in einen Zustand versetzt, der die Programmfortsetzung erlaubt.

Vor dem Wiederanfahren an die Kontur mit anschließendem Fortsetzen der Programmbearbeitung, können über ein automatisch gestartetes anwenderspezifisches ASUP alle noch eventuell erforderlichen Ausgangszustände erzeugt werden.

Kanäle

In Verbindung mit HMI ist SERUPRO für folgende Kanäle vorgesehen:

- Nur für den aktuellen SERUPRO-Kanal (1)
- Für alle Kanäle mit gleichem Werkstücknamen wie der SERUPRO-Kanal (2)
- Für alle Kanäle mit gleicher BAG wie der SERUPRO-Kanal (3)
- Für alle Kanäle der NCU (4)

Die Auswahl des Kanalumfanges für SERUPRO erfolgt über die Projektierungsdatei **maschine.ini**, in dem Abschnitt [BlockSearch]:

Section [BlockSearch]	Satzsuchlauf-Funktion für HMI freigeben und Satzsuchlauf-Konfiguration auswählen
SeruproEnabled=1	;SERUPRO Softkey bei HMI verfügbar. Standardwert ist (1)
SeruproConfig=1	;Nummer (1) bis (4) der oben genannten Kanalgruppierung. Standardwert ist (1)

Alle anderen mit SERUPRO gestarteten Kanäle werden im Modus "Self-Acting Serupro" betrieben. Nur der Kanal, in dem auch ein Zielsatz ausgewählt wurde, kann mit Satzsuchlauf im Modus SERUPRO gestartet werden.

Aktivierung

Die Aktivierung von SERUPRO erfolgt über HMI. Mit den Softkey "Prog.Test Kontur" wird SERUPRO bedient.

SERUPRO verwendet REPOS, um den Zielsatz anzufahren.

Zeitlicher Ablauf von SERUPRO

- Über HMI wird Softkey "Pog. Test Kontur" und das Suchziel bedient.
- Die NC startet jetzt selbsttätig das angewählte Programm im Modus "Programmtest".
 - Achsen werden dabei nicht verfahren.
 - Hilfsfunktionen \$A_OUT und die direkte PLC-IO werden ausgegeben.
 - Die Hilfsfunktionen des Zielsatzes werden nicht ausgegeben.
- Die NC stoppt am Beginn des Zielsatzes, wählt intern Programmtest ab, und zeigt die Stopp-Bedingung "Warten: Suchziel gefunden" an.
- Liegt das anwenderspezifische ASUP "PROG_EVENT.SPF" vor, wird es automatisch gestartet.
- Mit dem nächsten NC-Start erfolgt ein Wiederanfahren an die Kontur (REPOS). Der REPOS-Vorgang erfolgt über ein System-ASUP und kann über die Funktion "Editierbares ASUP" erweitert werden.

Randbedingungen für Satzsuchlauf SERUPRO

Die Funktion SERUPRO darf nur in der Betriebsart "Automatik" aktiviert und im Programmzustand (Kanalzustand RESET) abgebrochen werden.

Startet im Normalbetrieb **nur** der PLC gemeinsam mehrere Kanäle, so kann dies durch SERUPRO in jedem Kanal simuliert werden.

Bei Maschinendatum-Einstellung:

MD10708 \$MN_SERUPRO_MASK, Bit 1 = 0

bricht der Alarm 16942 "Kanal %1 Start-Programmbefehl Aktion %2<ALNX> nicht möglich" die Simulation dann ab, wenn der Teileprogrammbefehl `START` verwendet wird.

Das Maschinendatum:

MD10707 \$MN_PROG_TEST_MASK

erlaubt das Abschalten im gestoppten Zustand und hat keinen Einfluss auf den SERUPRO-Vorgang. Die Standardvorbelegung erlaubt das Abschalten nur im RESET-Zustand.

Hinweis

Nach dem Abschalten von Programmtest beginnt ein REPOS-Vorgang, für den dieselben Einschränkungen wie beim SERUPRO-Anfahren gelten. Auftretende Beeinträchtigungen lassen sich mit einem ASUP verhindern.

SERUPRO-Verhalten beeinflussen

Für die nachfolgend beispielhaft aufgeführten Funktionen kann das Verhalten von SERUPRO NC-spezifisch vorgegeben werden:

- Programmierter Halt (M0)
- Programmkoordinierungsbefehl START
- Group-SERUPRO
- Kanalübergreifendes Beenden von SERUPRO
- Override

MD10708 \$MN_SERUPRO_MASK = <Verhalten bei SERUPRO>

Kanalspezifische Grundstellungen für SERUPRO

Mit dem nachfolgenden Maschinendatum werden im Normalfall die kanalspezifischen Grundstellungen nach Teileprogrammstart festgelegt:

MD20112 \$MC_START_MODE_MASK= <Grundstellungen>

Für SERUPRO können eigene Grundstellungen welche die Grundstellungen aus MD20112 ersetzen, vorgegeben werden:

MD22620 \$MC_START_MODE_MASK_PRT = <SERUPRO-Grundstellungen>

Die SERUPRO-Grundstellungen müssen explizit freigegeben werden über:

MD22621 \$MC_ENABLE_START_MODE_MASK_PRT = 1

NC/PLC-Nahtstellensignal "Satzsuchlauf via Programmtest ist aktiv"

Der Satzsuchlauf via Programmtest wird angezeigt über das NC/PLC-Nahtstellensignal: DB21, ... DBX318.1 == 1

Das Nahtstellensignal ist gesetzt vom Start des Satzsuchlaufs bis zum Einwechseln des Zielsatzes in den Hauptlauf.

Zum benutzerdefinierten ASUP nach dem SERUPRO-Vorgang

Hinweis

Entschließt sich der Maschinenhersteller nach dem SERUPRO-Vorgang ein ASUP gemäß Punkt 7. zu starten, so muss Folgendes beachtet werden:

Gestoppter Zustand nach Punkt 6.:

Die Maschinendaten:

MD11602 \$MN_ASUP_START_MASK

und

MD11604 \$MN_ASUP_START_PRIO_LEVEL

erlauben es dem NC, das ASUP aus dem gestoppten Zustand selbsttätig über FC9-Baustein zu starten.

Quittierung vom FC9 erst nach Beendigung des REPOS-Satz:

Das ASUP kann erst vom FC9-Baustein mit "Asup Done" als beendet gemeldet werden, wenn auch der REPOS-Satz beendet ist.

Abwahl des vorgesehenen REPOS-Vorgangs nach Punkt 8.:

Der Start des ASUPs wählt den vorgesehenen REPOS-Vorgang ab!

Daher sollte das ASUP mit REPOSA abgeschlossen werden, um den REPOS-Vorgang beizubehalten.

Löschen eines unerwünschten REPOS-Vorgangs:

Der unerwünschte REPOS-Vorgang wird gelöscht, indem das ASUP mit M17 oder RET beendet wird.

Besondere Behandlung von ASUP:

Grundsätzlich wird ein ASUP, das mit REPOS endet und aus einem gestoppten Zustand heraus gestartet wird, besonders behandelt.

Das ASUP stoppt selbsttätig vor dem REPOS-Satz und zeigt dies an über:

DB21, ... DBX318.0 (ASUP angehalten)

Automatischer ASUP-Start

Das unter dem Pfad:

/_N_CMA_DIR/_N_PROG_EVENT_SPF

liegende ASUP, wird automatisch mit dem Maschinendatum:

MD11450 \$MN_SEARCH_RUN_MODE, Bit1 = 1

im SERUPRO-Anfahren nach folgenden Ablauf gestartet:

1. Der SERUPRO-Vorgang ist komplett durchgeführt.
2. Der Anwender löst "NC-Start" aus.
3. Das ASUP wird gestartet.
4. Die NC stoppt vor dem Teileprogrammbefehl REPOS selbsttätig und die Meldung "Zur Programmfortsetzung NC-Start geben" erscheint.

5. Der Anwender drückt zum zweiten Mal "NC-Start".
6. Die NC führt die Wiederanfahrbewegung aus und setzt das Teileprogramm mit dem Zielsatz fort.

Hinweis

Der automatische ASUP-Start mit MD11450 benötigt **Starts**, um das Programm fortzusetzen.

Damit nähert sich das Ablaufverhalten an andere Satzsuchlauf Typen an.

2.8.2 Wiederanfahren an die Kontur (REPOS)

Mit der Funktion "Wiederanfahren an die Kontur" (REPOS) kann eine unterbrochene Bearbeitung an der Unterbrechungsstelle fortgesetzt werden. Im Gegensatz zu REPOS ermöglicht SERUPRO das "Nachholen" oder "Wiederholen" eines Programmabschnittes. Dazu wird, nachdem SERUPRO den Zielsatz gefunden hat, die Kontur an der mit REPOS-Mode wählbaren Stelle angefahren und die Bearbeitung fortgesetzt.

SERUPRO: REPOS Verhalten einstellen

Das REPOS-Verhalten, d. h. das Verhalten im Wiederanfahrtsatz, wird über folgendes Maschinendatum eingestellt:

MD11470 \$MN_REPOS_MODE_MASK = <REPOS-Mode>

<REPOS-Mode>		
Bit	Wert	Bedeutung
0	0	Ein unterbrochene Verweilzeit wird wiederholt
	1	Ein unterbrochene Verweilzeit fortgesetzt
1	-	Reserviert
2	0	DB31, ... DBX10.0 (REPOS Verzögerung) wird im Wiederanfahrtsatz nicht beachtet.
	1	DB31, ... DBX10.0 (REPOS Verzögerung) wird im Wiederanfahrtsatz beachtet.
3	0	SERUPRO: Im Wiederanfahrtsatz nur Bahnachsen verfahren
	1	SERUPRO: Im Wiederanfahrtsatz Bahn- und Positionierachsen gleichzeitig verfahren
4	0	REPOS: Im Wiederanfahrtsatz nur Bahnachsen verfahren
	1	REPOS: Im Wiederanfahrtsatz Bahn- und Positionierachsen gleichzeitig verfahren
5	0	Während der Unterbrechung geänderte Vorschübe und Spindeldrehzahlen wirken erst ab dem ersten Teileprogrammsatz nach der Unterbrechungsstelle
	1	Während der Unterbrechung geänderte Vorschübe und Spindeldrehzahlen wirken ab der Unterbrechungsstelle, d. h. sie werden bereits im Restsatz gültig und werden damit vorgezogen. Dieses Verhalten bezieht sich auf jeden REPOS-Vorgang.

<REPOS-Mode>		
Bit	Wert	Bedeutung
6	0	SERUPRO: Im Wiederanfahrtsatz werden neutrale Achsen und positionierende Spindeln als Bahnachsen verfahren.
	1	SERUPRO: Im Wiederanfahrtsatz werden neutrale Achsen und positionierende Spindeln als Kommandoachsen verfahren. Neutrale Achsen und positionierende Spindeln werden nach SERUPRO repositioniert. Für Neutrale Achsen, die nicht repositioniert werden dürfen, muss die REPOS Verzögerung gesetzt werden: DB31, ... DBX10.0 = 1 (REPOS Verzögerung)
7	0	Die REPOS Verzögerung ist nicht freigegeben.
	1	Die REPOS Verzögerung ist freigegeben. Achsen mit aktiver REPOS Verzögerung (DB31, ... DBX10.0 == 1), die keine Geometrie- oder Orientierungsachsen sind, werden beim Wiederanfahren nicht verfahren.

 **VORSICHT**

Kollisionsgefahr

MD11470 \$MN_REPOS_MODE_MASK, Bit 3 oder Bit 4 = 1

Es liegt in der alleinigen Verantwortung des Anwenders sicherzustellen, dass es beim gleichzeitigen Verfahren der Achsen im Wiederanfahrtsatz zu keiner Kollision an der Maschine kommt.

Wiederanfahren mit gesteuertem REPOS

Der REPOS-Mode kann für die Bahnachsen über die NC/PLC-Nahtstelle vorgegeben werden:
DB21, ... DBX31.0 - 2 (REPOS-Mode)

Der REPOS-Mode wird im NC-Programm programmiert und bestimmt das Anfahrverhalten (siehe Kapitel "Wiederanfahren an die Kontur mit gesteuertem REPOS (Seite 83)").

Das REPOS-Verhalten einzelner Achsen kann auch über NC/PLC-Nahtstellensignale gesteuert werden und muss über das Maschinendatum freigegeben werden:
MD11470 \$MN_REPOS_MODE_MASK.BIT 2 = 1

Bahnachsen können nicht einzeln beeinflusst werden. Bei allen weiteren Achsen, die keine Geometrieachsen sind, kann REPOS einzelner Achsen vorübergehend verhindert und auch verschoben werden. Über NC/PLC-Nahtstellensignal werden die einzelnen Kanalachsen, die

REPOS herausfahren möchte, zu einem späteren Zeitpunkt wieder freigegeben oder weiter gesperrt.



GEFAHR

Kollisionsgefahr

Durch das Signal DB31, ... DBX2.2 (Restweg löschen) ergibt sich folgendes gefährliches Verhalten, wenn die Funktion "Repositionieren einzelner Achsen verhindern" angewählt ist:

MD11470 \$MN_REPOS_MODE_MASK.Bit 2 == 1

Solange eine Achse nach der Unterbrechung inkrementell programmiert wird, fährt die NC andere Positionen an als ohne Unterbrechung.

Siehe nachfolgendes Beispiel: Achse wird inkrementell programmiert

Beispiel: Rundachse A wird inkrementell programmiert

Die Rundachse A ist die vierte Maschinenachse.

- Die Rundachse A steht vor dem REPOS-Vorgang auf 11°
Im Unterbrechungssatz, d. h. im Zielsatz von SERUPRO, soll die Rundachse A auf 27° verfahren.
Beliebig viele Sätze später wird die Rundachse A inkrementell um 5° verfahren:
N1010 POS[A]=IC(5) FA[A]=1000
Bei gesetztem Nahtstellensignal DB34 DBX10.0 = 1 (REPOS Verzögerung) verfährt die Rundachse A im REPOS-Vorgang nicht und wird mit N1010 nach 32° gefahren. Der Benutzer muss evtl. den Weg von 11° nach 27° bewusst quittieren.

**GEFAHR****Kollisionsgefahr**

Da die Achse nach der Unterbrechung inkrementell programmiert wird, verfährt sie nach 16° statt 32°.

- Achsen einzeln starten
Das REPOS-Verhalten für das SERUPRO-Anfahren mit mehreren Achsen wird angewählt mit:
MD11470 \$MN_REPOS_MODE_MASK.BIT 3 = 1
Die NC beginnt SERUPRO-Anfahren mit einem Satz, der **alle** Positionierachsen auf das programmierte Ende und die Bahnachse auf den Zielsatz fährt.
Der Benutzer startet die einzelnen Achsen, indem er die Vorschubfreigaben entsprechend anwählt. Anschließend wird der Zielsatz abgefahren.
- Positionierachsen im Wiederanfahrtsatz repositionieren
Positionierachsen werden nicht im Restsatz, sondern im Wiederanfahrtsatz repositioniert und betreffen nicht nur den Satzsuchlauf über Programmtest beim SERUPRO-Anfahren:
MD11470 \$MN_REPOS_MODE_MASK.Bit 3 = 1: für Satzsuchlauf über Programmtest (SERUPRO)
MD11470 \$MN_REPOS_MODE_MASK.Bit 4 = 1: für jedes REPOS

Hinweis

Ist weder Bit 3 noch Bit 4 gesetzt, werden in dieser Phase Achsen, die keine Bahnachsen sind im Restsatz repositioniert.

Achse mit REPOS-Verschiebung verzögert anfahren

Ist das achsspezifische Nahtstellensignal DB31, ... DBX10.0 (REPOS Verzögerung) gesetzt, wird mit der positiven Flanke von DB21, ... DBX31.4 (REPOS Mode Änderung) die REPOS-Verschiebung für diese Achse erst mit ihrer nächsten Programmierung herausgefahren.

Ob diese Achse gerade einer REPOS-Verschiebung unterliegt, kann über Synchronaktionen mit \$AA_REPOS_DELAY gelesen werden.



VORSICHT

Kollisionsgefahr

DB31, ... DBX10.0 (REPOS Verzögerung) hat auf Maschinenachsen, die eine Bahn bilden, keine Auswirkung.

Ob eine Achse eine Bahnachse ist, kann über DB31, ... DBX76.4 (Bahnachse) gelesen werden.

Übernahmezeitpunkt der REPOS-Signale

Mit der positiven Flanke von DB21, ... DBX31.4 (REPOS Mode Änderung) werden folgende REPOS-Signale in die NC übernommen:

- Kanalspezifisch: DB21, ... DBX31.0 - 2 (REPOS Mode)
- Achsspezifisch: DB31, ... DBX10.0 (REPOS Verzögerung)

Die Pegel der REPOS-Signale beziehen sich auf den aktuellen Hauptlaufsatz. Es werden folgende zwei Fälle unterschieden:

1. Im Hauptlauf befindet sich **ein** Wiederanfahrtsatz eines gerade aktiven REPOS-Vorgangs. Der laufende REPOS-Vorgang wird abgebrochen, neu aufgesetzt, und die REPOS-Verschiebungen werden durch die oben genannten REPOS-Signale beeinflusst.
2. Im Hauptlauf befindet sich **kein** Wiederanfahrtsatz eines gerade aktiven REPOS-Vorgangs. Jeder zukünftige REPOS-Vorgang, der an den aktuellen Hauptlaufsatz wieder anfahren möchte, wird durch die oben genannten REPOS-Signale beeinflusst.

Hinweis

In einem laufenden ASUP wirkt DB21, ... DBX31.4 (REPOS Mode Änderung) nicht auf das abschließende REPOS, außer das Signal wird zufällig zum Zeitpunkt gesetzt zu dem die REPOS-Sätze abgearbeitet werden.

Im 1. Fall ist das Signal nur im gestoppten Zustand erlaubt.

Verhalten bei RESET:

- Die NC hat das PLC-Signal bereits quittiert:
DB21, ... DBX31.4 == 1 (REPOS Mode Änderung) **UND**
DB21, ... DBX319.0 == 1 (REPOS Mode Änderung Quittung)
Tritt in dieser Situation ein Kanal-Reset auf, wird der aktive REPOS-Mode gelöscht:
DB21, ... DBX319.1 - 3 = 0 (Aktiver REPOS-Mode)
- Die NC hat das PLC-Signal noch nicht quittiert:
DB21, ... DBX31.4 == 1 (REPOS Mode Änderung) **UND**
DB21, ... DBX319.0 == 0 (REPOS Mode Änderung Quittung)
Tritt in dieser Situation ein Kanal-Reset auf, wird die Quittierung der REPOS Mode Änderung und der aktive REPOS-Mode gelöscht:
DB21, ... DBX319.0 = 0 (REPOS Mode-Änderung Quittung)
DB21, ... DBX319.1 - 3 = 0 (Aktiver REPOS-Mode)

Mit NC/PLC-Nahtstellensignalen SERUPRO-Anfahren beeinflussen

Das SERUPRO-Anfahren kann mit DB21, ... DBX31.4 (REPOS Mode Änderung) und den dazugehörigen Signalen in den folgenden Phasen eingesetzt werden:

- Von "Suchziel gefunden" bis "Start des SERUPRO-ASUPs"
- Von "SERUPRO-ASUP stoppt selbsttätig vor dem REPOS" bis "Zielsatz wird abgearbeitet"

Während das SERUPRO-ASUP z. B. im Programmteil vor dem REPOS abgearbeitet wird, wirkt das Nahtstellensignal nicht auf SERUPRO-Anfahren.

REPOS Abläufe mit NC/PLC-Nahtstellensignalen**REPOS mit NC/PLC-Nahtstellensignalen steuern**

REPOS-Verschiebungen lassen sich mit den folgenden NC/PLC-Nahtstellensignalen beeinflussen:

- DB21, ... DBX31.0 - 2 (REPOS Mode)
- DB21, ... DBX31.4 (REPOS Mode Änderung)
- DB31, ... DBX10.0 (REPOS Verzögerung)
- DB31, ... DBX72.0 (REPOS Verzögerung)

REPOS-Quittierungssignale

Mit folgenden NC/PLC-Nahtstellensignalen werden Funktionen, die das REPOS-Verhalten über PLC beeinflussen, **von der NC** quittiert:

- DB21, ... DBX319.0 (REPOS Mode-Änderung Quittung)
- DB21, ... DBX319.1 - 3 (Aktiver REPOS Mode)
- DB21, ... DBX319.5 (REPOS Verzögerung Quittung)
- DB31, ... DBX70.0 (REPOS Verschiebung)
- DB31, ... DBX70.1 (REPOS Verschiebung gültig)
- DB31, ... DBX70.2 (REPOS Verzögerung Quittung)
- DB31, ... DBX76.4 (Bahnachse)

Weitere Informationen hierzu finden Sie unter "REPOS-Verschiebung in der Nahtstelle".

REPOS-Quittierungsvorgänge

Wird von der NC eine "REPOS Mode Änderung" erkannt (DB21, ... DBX31.4 == 1), wird diese von der PLC mit DB21, ... DBX319.0 = 1 quittiert.

Hinweis

Hat die NC das Nahtstellensignal DB21, ... DBX31.4 (REPOS Mode Änderung) noch nicht mit dem Nahtstellensignal DB21, ... DBX319.0 (REPOS Mode-Änderung Quittung) quittiert, führte ein Kanal-Reset in dieser Situation zum Programmabbruch, und REPOS, mit dem der REPOS Mode beeinflusst werden soll, wird nicht durchgeführt.

Ein von PLC vorgegebener REPOS Mode wird von der NC mit den folgenden Nahtstellensignalen quittiert:

- DB21, ... DBX319.1 - 3 (Aktiver REPOS Mode)
- DB31, ... DBX10.0 (REPOS Verzögerung)
- DB31, ... DBX70.2 (REPOS Verzögerung Quittung)

Beispiel

- Zeitpunkt ②: Ein NC-Programm wird in Satz N20 mit NC-Stop angehalten. Alle Achsen werden über ihre parametrisierte Bremsrampe bis zum Stillstand abgebremst.
- Zeitpunkt ③: Nachdem vom PLC-Anwenderprogramm der "REPOS Mode" gesetzt wurde, übernimmt die NC den REPOS Mode mit der 0/1-Flanke von "REPOS Mode-Änderung".
- Zeitpunkt ④: "REPOS Mode Änderung Quittung" bleibt so lange gesetzt, bis das ASUP ausgelöst wird
- Zeitpunkt ⑤: Der REPOS-Vorgang wird im ASUP begonnen.
- Zeitpunkt ⑥: Der Restsatz des ASUPs wird wieder eingewechselt.

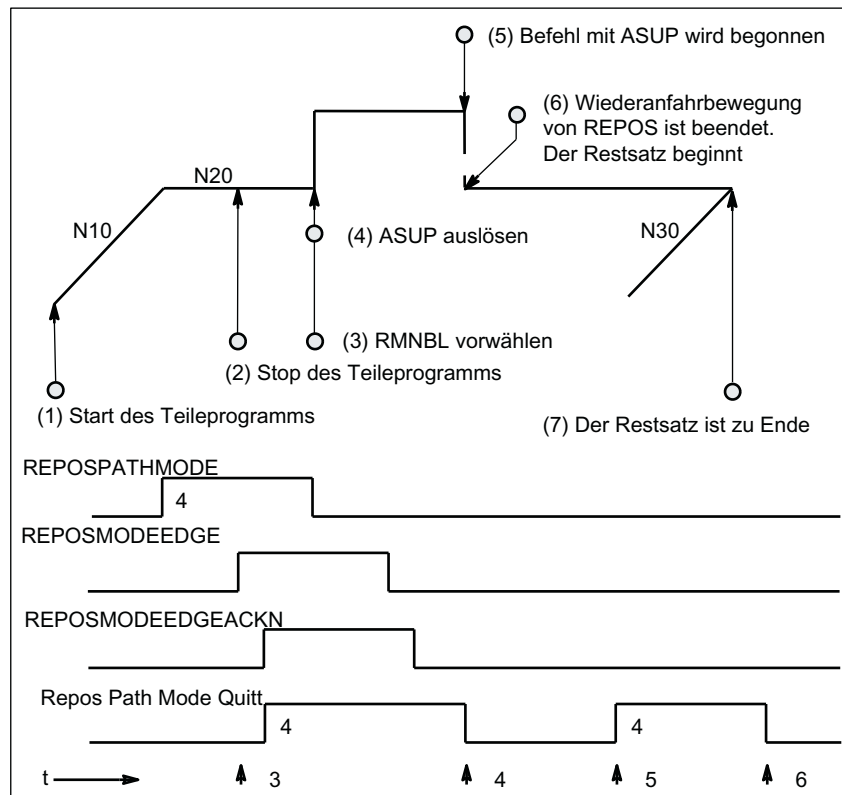


Bild 2-5 REPOS Ablauf im Teilprogramm mit zeitlichen Quittierungssignalen von NC

NC setzt Quittierung erneut

Phase, in der REPOSPATHMODE weiterhin wirkt (Restsatz des im → Zeitpunkt (2) gestoppten Programms ist noch nicht zu Ende ausgeführt).

Sobald die REPOS-Wiederanfahrbewegung des ASUP bearbeitet wird, setzt der NC den "Repos Path Mode Quitt" erneut (→ Zeitpunkt (5)). Sollte kein REPOSPATHMODE über NC/ PLC-Nahtstellensignal vorgewählt worden sein, wird der programmierte REPOS-Mode angezeigt.

"Repos Path Mode Quitt" wird mit dem Einwechseln des Restsatzes (→ Zeitpunkt (6)) zurückgenommen. Der dem im → Zeitpunkt (2) folgende Teilprogrammsatz N30 wird fortgesetzt.

Das Nahtstellensignal:

DB31, ... DBX70.2 (REPOS Verzögerung Quittung) ist analog definiert.

DB31, ... DBX70.1 (REPOS Verschiebung gültig) = 1, wenn:

DB21, ... DBX319.1-319.3 (Aktiver REPOS Mode) = 4 (RMNBL).

Gültige REPOS-Verschiebung

Mit dem Ende des SERUPRO-Vorgangs kann der Anwender die REPOS-Verschiebung über das NC/PLC-Nahtstellensignal Achse/Spindel (NC→PLC):
DB31, ... DBX70.0 (REPOS Verschiebung) auslesen.

Dieses Signal hat folgende Auswirkung für diese Achse:

- Wert 0: Keine REPOS-Verschiebung wird herausgefahren werden.
- Wert 1: Eine REPOS-Verschiebung wird herausgefahren werden.

Gültigkeitsbereich

Das Nahtstellensignal:
DB31, ... DBX70.0 (REPOS Verschiebung)
wird mit dem Ende des SERUPRO-Vorganges versorgt.

Mit dem Start eines SERUPRO-ASUP oder dem automatischen ASUP-Start wird die REPOS-Verschiebung ungültig.

REPOS-Verschiebung im Gültigkeitsbereich aktualisieren

Zwischen SERUPRO-Ende und -Start kann mit Moduswechsel die Achse in JOG verfahren werden.

Der Anwender fährt die REPOS-Verschiebung händisch mit JOG heraus, um das NST DB31, ... DBX70.0 (REPOS Verschiebung) auf den Wert 0 zu setzen.

Im Gültigkeitsbereich kann die Achse auch über FC18 verfahren werden, wobei das NST DB31, ... DBX70.0 (REPOS Verschiebung) ständig aktualisiert wird.

Gültigkeitsbereich anzeigen

Der Gültigkeitsbereich von der REPOS-Verschiebung wird angezeigt mit dem Nahtstellensignal:

DB31, ... DBX70.1 (REPOS Verschiebung gültig)

Es wird angegeben, ob eine gültige Berechnung vorliegt:

- Wert 0: REPOS-Verschiebung dieser Achse ist korrekt berechnet.
- Wert 1: REPOS-Verschiebung dieser Achse ist nicht berechenbar, das REPOS liegt in der Zukunft, z. B. am ASUP-Ende, oder kein REPOS aktiv.

REPOS-Verschiebung nach einem Achstausch

Mit dem Sammelsignal DB21, ... DBX319.5 (REPOS Verzögerung) kann festgestellt werden, ob eine gültige REPOS-Verschiebung stattgefunden hat:

- Wert 0: Alle Achsen, die von diesem Kanal aktuell kontrolliert werden, haben entweder keine REPOS-Verschiebung oder ihre REPOS-Verschiebungen sind ungültig.
- Wert 1: Sonstige.

REPOS-Verschiebung bei synchronisierter Synchronspindelkopplung

Beim Wiederaufstart mit SERUPRO wird an die Unterbrechungsstelle wieder vorgelaufen. War eine Synchronspindelkopplung bereits synchronisiert, dann existiert keine REPOS-Verschiebung der Folgespindel und es steht auch kein Synchronisationsweg an. Die Synchronisationssignale bleiben gesetzt.

Suchziel gefunden beim Satzwechsel

Das achsspezifische NC/PLC-Nahtstellensignal DB31, ... DBX76.4 (Bahnachse) ist 1, wenn die Achse Teil des Bahnverbundes ist.

Dieses Signal zeigt den Zustand des aktuell zu bearbeitenden Satzes beim Satzwechsel an. Spätere Zustandsänderungen werden nicht berücksichtigt.

Wenn der SERUPRO-Vorgang mit "Suchziel gefunden" beendet ist, bezieht sich DB31, ... DBX76.4 (Bahnachse) auf den Zielsatz.

2.8.2.1 Wiederaufstart an die Kontur mit gesteuertem REPOS

Nachdem mit SERUPRO der Zielsatz gefunden wurde, erfolgt vor dem Fortsetzen des unterbrochenen Programms ein REPOS-Vorgang zum Wiederaufstart der Kontur. Standardmäßig ist der REPOS-Mode "Wiederaufstart an den Satzanfangspunkt des Zielsatzes" (RMBBL) aktiv. Über die NC/PLC-Nahtstelle kann der REPOS-Mode anwenderspezifisch vorgegeben werden:

DB21, ... DBX31.0 - .2 (REPOS-Mode)

Weitere Informationen

Eine ausführliche Beschreibung der Nahtstellensignale findet sich im Funktionshandbuch PLC.

REPOS-Mode: Wiederaufstart an den nächstliegenden Bahnpunkt (RMNBL)

Im REPOS-Mode `RMNBL` wird von der REPOS-Startposition aus der nächstliegende Punkt der Kontur angefahren.

Beispiel

Die Programmunterbrechung erfolgte an einer beliebigen Stelle im Satz N110. Die Achsen wurden anschließend, z.B. manuell, auf Position (A) verfahren. Nachdem SERUPRO den Zielsatz N110 gefunden hat, erfolgt der REPOS-Vorgang mit REPOS-Mode `RMNBL`. Von der REPOS-Startposition (A) aus ist Punkt (B) der nächstliegende Punkt der Kontur. Mit Erreichen des Punktes (B) ist der REPOS-Vorgang abgeschlossen. Ab Punkt (B) wird wieder die programmierte Kontur des unterbrochenen Programms abgefahren.

MD22600 \$MC_SERUPRO_SPEED_MODE = <Wert>

<Wert>	Bedeutung
0	Programmtest mit Satzsuchlauf-/Probelaufvorschub-Geschwindigkeit: <ul style="list-style-type: none"> • Achsen: MD22601 \$MC_SERUPRO_SPEED_FACTOR * Probelaufvorschub • Spindeln: MD22601 \$MC_SERUPRO_SPEED_FACTOR * programmierte Drehzahl Dynamische Begrenzungen von Achsen / Spindeln werden nicht beachtet.
1	Programmtest mit programmierter Geschwindigkeit: <ul style="list-style-type: none"> • Achsen: Probelaufvorschub • Spindeln: programmierte Drehzahl Dynamische Begrenzungen von Achsen / Spindeln werden beachtet.
2	Programmtest mit Probelaufvorschub Unter Programmtest wird mit der programmierten Geschwindigkeit / Drehzahl gefahren. Dynamische Begrenzungen von Achsen / Spindeln werden beachtet.
3	Programmtest mit Satzsuchlauf-Geschwindigkeit Unter Programmtest wird mit folgender Geschwindigkeit gefahren: <ul style="list-style-type: none"> • Achsen: MD22601 \$MC_SERUPRO_SPEED_FACTOR * programmierter Vorschub • Spindeln: MD22601 \$MC_SERUPRO_SPEED_FACTOR * programmierte Drehzahl. Dynamische Begrenzungen von Achsen / Spindeln werden nicht beachtet. Hinweis Bei aktivem Umdrehungsvorschub (z.B. G95) wird der programmierte Vorschub nicht mit MD22601 \$MC_SERUPRO_SPEED_FACTOR multipliziert, sondern nur die programmierte Spindeldrehzahl. Dadurch ergibt sich auch hier eine Erhöhung der effektiven Bahngeschwindigkeit um MD22601 \$MC_SERUPRO_SPEED_FACTOR.

Randbedingungen

Hauptlaufachsen

MD22600 \$MC_SERUPRO_SPEED_MODE wirkt bei SERUPRO auf folgende Hauptlaufachsen:

- PLC-Achsen
- Kommando-Achsen
- Positionierachsen
- Pendelachsen

Synchronaktionen

ACHTUNG
Aktionen vom Synchronaktion werden bei SERUPRO u.U. nicht ausgeführt
Da während SERUPRO intern andere Aktualwerte (z.B. Achspositionen) erzeugt werden als im normalen Programmablauf, kann es bei SERUPRO vorkommen, dass Bedingungen von Synchronaktionen die Aktualwerte (z.B. Achspositionen) abprüfen, nicht mehr als wahr (TRUE) erkannt werden und demzufolge der Aktionsteil der Synchronaktion nicht ausgeführt wird.

Umdrehungsvorschub

Auswirkungen von MD22601 \$MC_SERUPRO_SPEED_FACTOR bei DryRun:

- Umschalten von G95/G96/G961/G97/G971 auf G94
- Gewindebohren und Gewindeschneiden: normale DryRun-Geschwindigkeit.

Gewindebohren ohne Ausgleichsfutter

- Bei "Gewindebohren ohne Ausgleichsfutter" (G331/G332), wird die Spindel lagegeregelt im Bahnverbund interpoliert. Dabei ist die Bohrtiefe (Linearachse), die Gewindesteigung und Drehzahl (Spindel) vorgegeben.
Während DryRun ist die Geschwindigkeit von der Linearachse vorgegeben, die Drehzahl bleibt konstant und die Gewindesteigung wird angepasst.
Nach SERUPRO ergibt sich für die Spindel eine zum Normalbetrieb abweichende Position, weil die Spindel bei SERUPRO weniger oft gedreht hat.

2.8.4 SERUPRO-ASUP

SERUPRO-ASUP Besonderheiten

Während SERUPRO-ASUP sind Besonderheiten zu beachten bei:

- Referenzpunktfahren: Referenzieren per Teileprogramm G74
- Werkzeugverwaltung: Werkzeugwechsel und Magazindaten
- Spindelhochlauf: Beim Start eines SERUPRO-ASUP

G74 Referenzpunkt fahren

Befindet sich zwischen dem Programmanfang und dem Suchziel die Anweisung G74 (Referenzpunktfahren), so wird dies von der NC ignoriert.

SERUPRO-Anfahren berücksichtigt diese Anweisung G74 nicht!

Werkzeugverwaltung

Bei aktiver Werkzeugverwaltung (WZV) wird folgende Einstellung empfohlen:

MD20310 \$MC_TOOL_MANAGEMENT_MASK BIT 20 = 0

Damit wird das während des SERUPRO-Vorgangs erzeugte WZV-Kommando **nicht** an die PLC ausgegeben!

Das WZV-Kommando wirkt sich wie folgt aus:

- Die NC quittiert die Kommandos selbst.
- Es werden keine Magazindaten verändert.
- Werkzeug-Daten werden nicht verändert.
Ausnahme:
Der Werkzeug-Zustand des im Testbetrieb aktivierten Werkzeuges kann den Zustand 'aktiv' annehmen. Damit kann nach dem SERUPRO-Vorgang das falsche Werkzeug auf der Spindel sein.
Abhilfe:
Der Anwender startet ein SERUPRO-ASUP, das real verfahren wird. Vor dem Start kann der Anwender ein ASUP starten, welches das korrekte Werkzeug einwechselt.

SERUPRO-Vorgang: Funktionalität: Im Ablauf vom Punkt 2. bis 6.

SERUPRO-ASUP: Funktionalität: Im Ablauf Punkt 7.

Außerdem muss im Maschinendatum MD20310 \$MC_TOOL_MANAGEMENT_MASK das Bit 11 = 1 gesetzt werden, da das ASUP ggf. eine T-Anwahl wiederholen muss.

Anlagen mit Werkzeugverwaltung und Nebenspindel unterstützen SERUPRO nicht!

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	; Im Programmtest-Betrieb wird
N530 IF \$P_ISTEST	; aus dem Programmkontext
N540 TNR_AKTUELL = \$P_TOOLNO	; das "aktuelle" Werkzeug gelesen.
N550 ELSE	; Andernfalls wird das Werkzeug der Spindel ausgelesen.
N560 TNR_AKTUELL = \$TC_MPP6[9998,1]	; T-Nummer des Werkzeugs auf der Spindel lesen
N570 ENDIF	
	; T-Nummer des vorgewählten Werkzeugs der Masterspindel lesen
	; Werkzeugwechsel nur ausführen, wenn Werkzeug noch nicht aktuell ist
N590 IF TNR_AKTUELL <> TNR_VORWAHL	; Werkzeugwechsellpunkt anfahren
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	
N610 M206	; Werkzeugwechsel ausführen
N620 ENDIF	

2.8 Satzsuchlauf Typ 5 (SERUPRO)

Programmcode	Kommentar
N630 M17	

ASUP für den Aufruf der Werkzeugwechselroutine nach Satzsuchlauf-Typ 5

Programmcode	Kommentar
PROC ASUPWZV2	
N1000 DEF INT TNR_SPINDEL	; Variable für aktive T-Nummer
N1010 DEF INT TNR_VORWAHL	; Variable für vorgewählte T-Nummer
N1020 DEF INT TNR_SUCHLAUF	; Variable für im Satzsuchlauf ermittelte T-Nummer
N1030 TNR_SPINDEL = \$TC_MPP6[9998,1]	; T-Nummer des Werkzeugs auf der Spindel lesen
N1040 TNR_SUCHLAUF = \$P_TOOLNO	; Durch Satzsuchlauf ermittelte T-Nummer lesen, ; d. h. dieses Werkzeug bestimmt die ; momentane Werkzeugkorrektur.
N1050 GETSELT(TNR_VORWAHL)	; T-Nummer des vorgewählten Werkzeugs lesen
N1060 IF TNR_SPINDEL == TNR_SUCHLAUF GOTOF AS- UP_ENDE1	
N1070 T = \$TC_TP2[TNR_SUCHLAUF]	; T-Anwahl über Werkzeugnamen
N1080 L6	; Werkzeugwechselroutine aufrufen
N1085 ASUP_ENDE1:	
N1090 IF TNR_VORWAHL == TNR_SUCHLAUF GOTOF AS- UP_ENDE	
N1100 T = \$TC_TP2[TNR_VORWAHL]	; T-Vorwahl restaurieren über Werkzeugnamen
N1110 ASUP_ENDE:	
N1110 M90	; Rückmeldung an PLC
N1120 REPOSA	; ASUP Ende

In beiden Programmen PROC L6 und PROC ASUPWZV2 wird der Werkzeugwechsel mit M206 anstelle mit M6 programmiert.
 Das ASUP-Programm "ASUPWZV2" benutzt verschiedene Systemvariablen, die einerseits den Programmfortschritt (\$P_TOOLNO) kennen und andererseits den aktuellen Zustand der Maschine (\$TC_MPP6[9998,1]) darstellen.

Spindelhochlauf

Beim Start des SERUPRO-ASUPs wird die Spindel nicht auf die im Programm vorgesehene Drehzahl hochgedreht, denn mit SERUPRO-ASUP soll nach dem Werkzeugwechsel, das neue Werkzeug auf die richtige Werkstückposition korrigiert werden.

Ein Spindelhochlauf wird mit SERUPRO-ASUP wie folgt durchgeführt:

- SERUPRO-Vorgang ist komplett beendet.
- Der Anwender startet über den FC-9 Baustein das SERUPRO-ASUP mit dem ggf. die Spindel hochgedreht wird.

- Der Start nach M0 im ASUP verändert den Spindelzustand nicht.
- SERUPRO-ASUP stoppt vor dem REPOS-Teileprogrammsatz selbsttätig.
- Der Anwender drückt START.
- Die Spindel dreht auf den Zielsatzzustand hoch, falls im ASUP die Spindel nicht anders programmiert worden war.

Hinweis

Anpassungen für REPOS der Spindeln:

Bei Anpassungen für SERUPRO-Anfahren und der Spindelfunktionalität sind Übergänge von Drehzahlsteuerbetrieb und Positionierbetrieb zu beachten.

Weitere Informationen zu den Betriebsartenwechsel von Spindeln siehe Kapitel "Betriebsarten und Betriebsartenwechsel (Seite 35)".

2.8.5 Self-Acting SERUPRO

Self-Acting SERUPRO

Die kanalspezifische Funktion "Self-Acting SERUPRO" erlaubt einen SERUPRO-Ablauf **ohne** vorher ein Suchziel in einem Programm der abhängigen SERUPRO-Kanäle definiert zu haben.

Außerdem kann ein spezieller Kanal, der "seruproMasterChan", für **jedes** "Self-Acting SERUPRO" definiert werden. In diesem kann ein Suchziel definiert werden.

Die Funktion "Self-Acting SERUPRO" unterstützt den kanalübergreifenden Satzsuchlauf SERUPRO.

Funktion

Mit den Vorgang "Self-Acting SERUPRO" kann kein Suchziel gefunden werden. Wird das Suchziel nicht erreicht, so wird auch kein Kanal angehalten. In bestimmten Situationen wird aber trotzdem der Kanal vorübergehend angehalten. Dabei wird der Kanal in der Regel auf einen anderen Kanal warten. Beispiele hierfür sind: Wait-Marken, Kopplungen oder Achstausch.

Warte-Phase tritt auf:

In dieser Warte-Phase prüft die NC den Kanal "seruproMasterChan", ob dieser ein Suchziel erreicht hat. Die Warte-Phase wird bei Nichterreichen eines Suchziels wieder verlassen.

Wird das Suchziel erreicht, so wird der SERUPRO-Vorgang auch in diesem Kanal beendet. Der Kanal "seruproMasterChan" muss im normalen SERUPRO-Modus gestartet worden sein.

Keine Warte-Phase tritt auf:

"Self-Acting SERUPRO" wird durch das M30 des Teileprogramms beendet.

Der Kanalzustand befindet sich danach wieder im RESET-Zustand.

Ein SERUPRO-Anfahren findet nicht statt.

Start einer Gruppe von Kanälen

Wird eine Gruppe von Kanälen nur mit "Self-Acting SERUPRO" gestartet, so werden alle Kanäle mit "RESET" beendet.

Ausnahmen:

Ein Kanal wartet auf einen Partnerkanal, der überhaupt nicht gestartet wurde.

Ein kanalübergreifender Satzsuchlauf kann wie folgt durchgeführt werden:

- Der Anwender wählt über HMI die Kanäle aus, die zusammenarbeiten müssen (Kanalgruppe).
- Aus der Kanalgruppe wählt der Anwender einen besonders wichtigen Kanal aus, für den er ein Suchziel explizit anwählen möchte (Zielkanal).
- Der HMI startet dann SERUPRO auf dem Zielkanal und "Self-Acting SERUPRO" auf dem Rest der Kanalgruppe.

Der Vorgang ist beendet, wenn **jeder** betroffene Kanal "seruproActive" gelöscht hat.

"Self-Acting SERUPRO" akzeptiert keinen Masterkanal auf einer anderen NCU.

Aktivierung

Die Aktivierung von "Self-Acting SERUPRO" erfolgt über HMI als Satzsuchlauf Start für Satzsuchlauf-Typ 5 für den Zielkanal "seruproMasterChan".

Für die vom Zielkanal gestarteten abhängigen Kanäle wird kein Suchziel angegeben.

2.8.6 Programmabschnitt für Wiederaufsetzen sperren

Programmierter Unterbrechungszeiger

Ist aufgrund von fertigungs- und/oder prozesstechnischen Gründen abzusehen, dass bei einem Programmabbruch innerhalb eines bestimmten Programmabschnitts kein Wiederaufsetzen möglich ist, kann dieser Programmabschnitt für den Zielsatz eines eventuellen Satzsuchlaufs gesperrt werden.

Erfolgt nach einem Programmabbruch innerhalb des für das Wiederaufsetzen gesperrten Programmabschnitts ein Satzsuchlauf auf die Unterbrechungsstelle, wird von der Steuerung dazu der letzte ausführbare Satz (Hauptlaufsatz) vor dem Beginn des gesperrten Bereichs als Zielsatz (Haltesatz) verwendet.

Programmierung

Syntax

```
IPTRLOCK ( )
```

Funktionalität

Markiert den Anfang des Programmabschnitts, ab dem Wiederaufsetzen gesperrt ist. Als Zielsatz für einen Satzsuchlauf mit "Wiederaufsetzen an der Unterbrechungsstelle" wird ab

jetzt, bis zur Freigabe mit `IPTRUNLOCK`, der nächste ausführbare Satz (Hauptlaufsatz) verwendet, in dem `IPTRLOCK` aktiv wird. Dieser Satz wird im weiteren Verlauf als **Haltesatz** bezeichnet.

Wirksamkeit: modal

Syntax

`IPTRUNLOCK()`

Markiert das Ende des für ein Wiederaufsetzen gesperrten Programmabschnitts. Als Zielsatz für einen Satzsuchlauf mit "Wiederaufsetzen an der Unterbrechungsstelle" wird ab dem nächsten ausführbaren Satz (Hauptlaufsatz) in dem `IPTRLOCK` aktiv wird, wieder der aktuelle Satz verwendet. Dieser Satz wird im weiteren Verlauf als **Freigabesatz** bezeichnet.

Wirksamkeit: modal

Beispiel

Programmcode	Kommentar
...	
N010 IPTRLOCK()	; Gesperrter Bereich: Anfang
N020 R1=R1+1	
N030 G4 F1	; Haltesatz
...	; Gesperrter Bereich
N200 IPTRUNLOCK()	; Gesperrter Bereich: Ende
N220 R1=R1+1	
N230 G4 F1	; Freigabesatz
...	

Randbedingungen

- `IPTRLOCK` wirkt innerhalb eines Programms (*.MPF, *.SPF) maximal bis zum Programmende (M30, M17, RET). Mit dem Programmende wird implizit `IPTRUNLOCK` aktiv.
- Eine mehrfache Programmierung von `IPTRLOCK` innerhalb eines Programms wirkt nicht kumulativ. Mit der ersten Programmierung von `IPTRUNLOCK` innerhalb des Programms oder mit Erreichen des Programmendes, werden alle vorausgegangenen `IPTRLOCK`-Aufrufe abgeschlossen.
- Erfolgt ein Unterprogrammaufruf innerhalb eines gesperrten Bereiches, ist auch für diese und alle eventuell folgenden Unterprogrammebenen das Wiederaufsetzen gesperrt. Die Sperre kann innerhalb des aufgerufenen Unterprogramms auch durch explizites Programmieren von `IPTRUNLOCK` nicht aufgehoben werden.

Beispiel: Verschachtelung gesperrter Programmabschnitte in zwei Programmebenen

Durch die Aktivierung der Wiederaufsetzsperrung in `PROG_1`, ist Wiederaufsetzen auch für `PROG_2` und alle eventuell folgenden Programmebenen gesperrt.

Programmcode	Kommentar
PROC PROG_1	; Programm 1
...	
N010 IPTRLOCK()	

2.8 Satzsuchlauf Typ 5 (SERUPRO)

Programmcode	Kommentar
N020 R1=R1+1	
N030 G4 F1	; Haltesatz
...	; Gesperrter Bereich: Anfang
N040 PROG_2	; Gesperrter Bereich
...	; Gesperrter Bereich: Ende
N050 IPTRUNLOCK()	
N060 R2=R2+2	
N070 G4 F1	; Freigabesatz
...	

Programmcode	Kommentar
PROC PROG_2	; Programm 2
N210 IPTRLOCK()	; Unwirksam aufgrund von Programm 1
...	
N250 IPTRUNLOCK()	; Unwirksam aufgrund von Programm 1
...	
N280 RET	; Unwirksam aufgrund von Programm 1

Beispiel 3: Mehrfache Programmierung von IPTRLOCK

Programmcode	Kommentar
PROC PROG_1	; Programm 1
...	
N010 IPTRLOCK()	
N020 R1=R1+1	
N030 G4 F1	; Haltesatz
...	; Gesperrter Bereich: Anfang
N150 IPTRLOCK()	; Gesperrter Bereich
...	; Gesperrter Bereich
N250 IPTRLOCK()	; Gesperrter Bereich
...	; Gesperrter Bereich: Ende
N360 IPTRUNLOCK()	
N370 R2=R2+2	
N380 G4 F1	; Freigabesatz
...	

Systemvariable

Über die Systemvariable \$P_IPTRLOCK kann der Status des aktuellen Satzes ermittelt werden:

\$P_IPTRLOCK	Bedeutung
FALSE	Der aktuelle Satz befindet sich nicht innerhalb eines für das Wiederaufsetzen gesperrten Programmabschnitts
TRUE	Der aktuelle Satz befindet sich innerhalb eines für das Wiederaufsetzen gesperrten Programmabschnitts

Automatische funktionsspezifische Wiederaufsetzsperrung

Für verschiedene Kopplungen kann das Ein/Ausschalten der Wiederaufsetzsperrung kanalspezifisch automatisch mit dem Ein/Ausschalten der Kopplung erfolgen:

MD22680 \$MC_AUTO_IPTR_LOCK, Bit x

Bit	Wert	Bedeutung
0	Elektronisches Getriebe (EGON / EGOF)	
	1	Automatischen Wiederaufsetzsperrung ist aktiv
	0	Automatischen Wiederaufsetzsperrung ist nicht aktiv
1	Axiale Leitwertkopplung (LEADON / LEADOF)	
	1	Automatischen Wiederaufsetzsperrung ist aktiv
	0	Automatischen Wiederaufsetzsperrung ist nicht aktiv

Dieser Programmbereich beginnt mit dem letzten ausführbaren Satz **vor** dem Einschalten und endet mit dem Ausschalten.

Der automatische Unterbrechungszeiger wird bei Kopplungen, die über Synchronaktionen ein- bzw. ausgeschaltet wurden, nicht aktiv.

Beispiel: Axiale Leitwertkopplung automatisch suchunfähig erklären:

Programmcode	Kommentar
N100 GO X100	
N200 EGON (Y, "NOC", X, 1, 1)	; Suchunfähiger Programmabschnitt beginnt.
N300 LEADON (A, B, 1)	
...	
N400 EGOF (Y)	
...	
N500 LEADOF (A, B)	; Suchunfähiger Programmabschnitt endet.
N600 GO X200	

Ein Programmabbruch im suchunfähigen Programmabschnitt (N200 - N500) versorgt den Unterbrechungszeiger stets mit N100.

ACHTUNG

Unerwünschter Zustand durch Funktionsüberschneidungen

Bei einer Überschneidung der Funktionen "Programmierbarer Unterbrechungszeiger" und "Automatischer Unterbrechungszeiger" über Maschinendatum wählt die NC den größtmöglichen suchunfähigen Bereich.

Ein Programm kann eine Kopplung fast über die gesamte Laufzeit benötigen. Der automatische Unterbrechungszeiger würde damit immer auf den Programmanfang weisen und die Funktion SERUPRO würde de facto nutzlos.

2.8.7 Verhalten bei Power On, Betriebsartenwechsel und RESET

SERUPRO ist bei Power On inaktiv. Während SERUPRO ist der Betriebsartenwechsel erlaubt. RESET bricht SERUPRO ab, der intern gewählte Programmtest ist wieder abgewählt. SERUPRO ist nicht mit anderen Satzsuchlauftypen kombinierbar.

2.8.8 Randbedingungen

2.8.8.1 STOPRE im Zielsatz

Alle satzübergreifenden Einstellungen erhält der STOPRE-Satz aus dem vorangegangenen Satz und kann damit Bedingungen vor dem eigentlichen Satz für die folgenden Fälle berücksichtigen:

- Aktuell bearbeitete Programmzeile mit dem Hauptlauf synchronisieren.
- Satzübergreifende Einstellungen für SERUPRO ableiten, um z. B. beim Anfahren von SERUPRO diese REPOS-Bewegung zu beeinflussen.

Beispiel: Durch Sollwertvorgabe einer X-Achse eine Z-Achse positionieren.

Wenn der Satz "G1 F100 Z=\$AA_IM[X]" interpretiert wird, so sorgt der vorangestellte STOPRE-Satz für die Synchronisation mit dem Hauptlauf. Damit wird über \$AA_IM der korrekte Sollwert der X-Achse gelesen, um die Z-Achse auf die gleiche Position zu fahren.

Beispiel: Externe Nullpunktverschiebung lesen und richtig einrechnen.

```
N10 G1 X1000 F100
N20 G1 X1000 F500
N30 G1 X1000 F1000
N40 G1 X1000 F5000
```

```
N50 SUPA G1 F100 X200          ; externe Nullpunktverschiebung nach 200 fahren
N60 G0 X1000
N70 ...
```

Durch ein implizites STOPRE vor N50 kann der NC die aktuelle Nullpunktverschiebung lesen und richtig einrechnen.

Bei einem SERUPRO-Vorgang auf das Suchziel N50 wird im SERUPRO-Anfahren auf den impliziten STOPRE repositioniert und die Geschwindigkeit ermittelt sich aus N40 mit F5000.

Impliziter Vorlaufstopp

Situationen, in denen ein impliziter Vorlaufstopp abgesetzt wird:

1. In allen Sätzen in denen einer der folgenden Variablenzugriffe vorkommt:
 - Programmierung einer Systemvariablen, die mit \$A... beginnt
 - Programmierung einer redefinierte Variable mit den Attribut SYN R / SYN RW
2. Bei folgenden Befehlen:
 - Teileprogrammbefehl MEACALC, MEASURE
 - Programmierung von SUPA (Frames und Online-Korrekturen unterdrücken)
 - Programmierung CTABDEF (Beginn der Kurventabellendefinition)
 - Teileprogrammbefehl WRITE/DELETE (Datei schreiben/löschen)
 - vor dem ersten WRITE/DELETE-Befehl einer Sequenz solcher Befehle
 - Teileprogrammbefehl EXTCALL
 - Teileprogrammbefehl GETSELT, GETEXET
 - bei Werkzeugwechsel und aktiver Werkzeugfeinkorrektur FTOCON
3. Bei folgender Kommando-Bearbeitung:
 - Endbearbeitung eines Satzsuchlaufs Typ 1 ("Satzsuchlauf ohne Berechnung")
 - Endbearbeitung eines Satzsuchlaufs Typ 2 mit Berechnung ("Satzsuchlauf an Konturendpunkt")

2.8.8.2 SPOS im Zielsatz

Ist eine Spindel mit M3 / M4 programmiert und wird im Zielsatz nach SPOS umgeschaltet, ist am Ende des SERUPRO-Vorganges (Status "Suchziel gefunden") die Spindel nach SPOS umgeschaltet.

DB31, ... DBX84.5 = TRUE (aktive Spindelbetriebsart: Positionierbetrieb)

2.8.8.3 Fahren auf Festanschlag (FXS)

Beim Wiederanfahren an die Kontur (REPOS) wird die Funktion "Fahren auf Festanschlag" (FXS) automatisch wiederholt. Dabei wird jede Achse berücksichtigt. Als Moment wird das zuletzt vor dem Suchziel programmierte Moment angewendet.

Systemvariable

Die Systemvariablen für "Fahren auf Festanschlag" haben bei SERUPRO folgende Bedeutung:

- \$AA_FXS: Fortschritt der Programmsimulation
- \$VA_FXS: Realer Maschinenzustand

Die beiden Systemvariablen haben **außerhalb** der Funktion SERUPRO immer dieselben Werte.

ASUP

Für SERUPRO kann ein anwenderspezifisches ASUP aktiviert werden.

Weitere Informationen

Ausführliche Informationen zum Satzsuchlauf SERUPRO finden sich im Funktionshandbuch Achsen und Spindeln; Fahren auf Festanschlag.

2.8.8.4 Fahren mit begrenztem Moment/Kraft (FOC)

Beim Wiederanfahren an die Kontur (REPOS) wird die Funktion "Fahren mit begrenztem Moment/Kraft" (FOC) automatisch wiederholt. Dabei wird jede Achse berücksichtigt. Als Moment wird das zuletzt vor dem Suchziel programmierte Moment angewendet.

Systemvariable

Die Systemvariablen für "Fahren mit begrenztem Moment/Kraft" haben bei SERUPRO folgende Bedeutung:

- \$AA_FOC: Fortschritt der Programmsimulation
- \$VA_FOC: Realer Maschinenzustand

Randbedingung

Ein wechselnder Momentenverlauf kann beim Wiederanfahren an die Kontur **nicht** realisiert werden.

Beispiel

Ein Programm fährt eine Achse X von Position 0 nach 100 und schaltet alle 20 Incremente für jeweils 10 Incremente "Fahren mit begrenztem Moment/Kraft" (FOC) ein. Dieser Momentenverlauf wird in der Regel mit satzweisem FOC erzeugt und kann beim Wiederanfahren an die Kontur (REPOS) nicht nachvollzogen werden. Stattdessen wird gemäß der letzten Programmierung die Achse X von 0 nach 100 mit oder ohne begrenztem Moment/Kraft fahren.

Weitere Informationen

Ausführliche Informationen zum Satzsuchlauf SERUPRO finden sich im Funktionshandbuch Achsen und Spindeln; Fahren auf Festanschlag.

2.8.8.5 Synchronspindel

Synchronspindel ist simulierbar.

Der Synchronspindelbetrieb mit einer Leitspindel und beliebig vielen Folgespindeln ist in allen vorhandenen Kanälen mit SERUPRO simulierbar.

Weitere Informationen

Weitere Informationen zu Synchronspindeln entnehmen Sie dem Funktionshandbuch Erweiterungsfunktionen; Synchronspindel

2.8.8.6 Kopplungen und Master-Slave

Soll- und Istwert-Kopplungen

Der SERUPRO-Vorgang ist eine Programmsimulation im Modus Programmtest mit dem Soll- und Istwertkopplungen simulierbar sind.

Festlegungen für Simulation von EG

Für die Simulation von EG werden damit folgende Festlegungen getroffen:

1. Es wird immer mit Sollwertkopplung simuliert.
2. Sind nur einige Leitachsen, d. h. nicht alle Leitachsen unter SERUPRO, wird die Simulation mit Alarm 16952 "ResetClear/NoStart" abgebrochen. Dies kann bei kanalübergreifenden Kopplungen auftreten.
3. Achsen, die aus NC-Sicht nur einen Geber haben und von Extern bewegt werden, können nicht korrekt simuliert werden. Insbesondere dürfen diese Achsen nicht in Kopplungen einfließen.



VORSICHT

Fehlerhafte Simulation

Um Kopplungen korrekt simulieren zu können, müssen die Kopplungen vorher abgeschaltet werden.

Dies kann mit Maschinendatum MD10708 \$MA_SERUPRO_MASK erfolgen.

Festlegungen für Achskopplungen

Der SERUPRO-Vorgang simuliert Achskopplungen stets unter der Annahme von Sollwertkopplung. Damit werden Endpunkte für **alle** Achsen berechnet, die als Zielpunkte für SERUPRO-Anfahren benutzt werden. Zugleich ist die Kopplung bereits mit "Suchziel gefunden" aktiv. Der Weg vom aktuellen Punkt bis zum Endpunkt wird bei SERUPRO-Anfahren unter aktiver Kopplung durchgeführt.

LEADON

Für die Simulation von Axialen Leitwertkopplungen ergeben sich damit folgende Festlegungen:

1. Es wird immer mit Sollwertkopplung simuliert.
2. SERUPRO-Anfahren erfolgt mit aktiver Kopplung und einer überlagerten Bewegung der Folgeachse, um den simulierten Zielpunkt zu erreichen.

Die allein durch die Kopplung bewegte Folgeachse kann den Zielpunkt nicht immer erreichen. Es wird im SERUPRO-Anfahren eine Überlagerte lineare Bewegung für die Folgeachse berechnet, um den simulierten Punkt anzufahren!

Simulierten Zielpunkt für LEAD mit JOG erreichen

Zum Zeitpunkt "Suchziel gefunden" ist die Kopplung insbesondere für die JOG-Bewegungen bereits aktiv. Bei nicht erreichten Zielpunkt kann beim SERUPRO-Anfahren die Folgeachse mit aktiver Kopplung und einer überlagerten Bewegung auf den Zielpunkt verfahren werden.

Hinweis

Weitere Informationen zum Wiederanfahren von Achskopplungen siehe Kapitel "Wiederanfahren an die Kontur (REPOS) (Seite 74)".

Master-Slave

Nach Abschluss des Satzsuchlaufs kann ein System ASUP automatisch gestartet werden. In diesem hat der Anwender die Möglichkeit den Kopplungszustand und die zugehörigen Achspositionen nachträglich zu beeinflussen. Die dazu erforderlichen Informationen werden über folgende Systemvariable bereitgestellt:

Systemvariable	Beschreibung
\$P_SEARCH_MASLD[<Slaveachse>]	Positionsversatz zwischen Slave- und Masterachse zum Schließzeitpunkt der Kopplung.
\$AA_MASL_STAT[<Slaveachse>]	Aktueller Zustand einer Master-Slave Kopplung
\$P_SEARCH_MASLC[<Slaveachse>]	Status: der Zustand der Kopplung wurde im Satzsuchlauf verändert
Die Systemvariablen werden beim Einschalten der Kopplung mit MASLON gelöscht.	

Hinweis

Die gekoppelten Achsen müssen sich zum Zeitpunkt des Satzsuchlaufs im gleichen Kanal befinden.

Weitere Informationen

Weitere Informationen zur Master-Slave-Kopplung finden sich im Funktionshandbuch Technologien; Drehzahl-/Drehmomentkopplung, Master-Slave"

Beispiel• **System-ASUP**

- Pfad und Name: /_N_CMA_DIR/PROGEVENT.SPF
- Masterachse: X
- Slaveachse: Y

Programmcode
PROG PROGEVENT
N10 IF(($\$S_SEARCH_MASLC[Y]$ <> 0) AND ($\$AA_MASL_STAT[Y]$ <> 0))
N20 MASLOF(Y)
N30 SUPA Y = $\$AA_IM[X]$ - $\$P_SEARCH_MASLD[Y]$
N40 MASLON(Y)
N50 ENDIF
N60 REPOSA
...
RET

• **Maschinendaten**

Damit das ASUP automatisch startet, müssen folgende Maschinendaten gesetzt werden:

- NC-spezifische Maschinendaten:
 - MD11604 $\$MN_ASUP_START_PRIO_LEVEL$ = 100
 - MD11450 $\$MN_SEARCH_RUN_MODE$ = 'H02'
- Kanal-spezifisch für den Kanal in dem das ASUP gestartet wird, oder allgemein für alle Kanäle:
 - MD20105 $\$MC_PROG_EVENT_IGN_REFP_LOCK$, Bit<n> = TRUE
n: Für alle benötigten ereignisgesteuerten Programmaufrufe (Prog-Events)
 - MD20115 $\$MC_IGNORE_REFP_LOCK_ASUP$, Bit<n> = TRUE
n: Für alle **benötigten Anwender**-Interrupts

ACHTUNG**System-Interrupts**

Mit MD20115 $\$MC_IGNORE_REFP_LOCK_ASUP$, Bit 8 bis 31 werden die System-Interrupts freigegeben.

Durch Bit 8 / Interrupt 9 wird ein ASUP gestartet, welches Verfahrensbewegungen beinhaltet.

Achskopplungen

- Beschleunigung der Abarbeitungsgeschwindigkeit und Leitachse und Folgeachsen in unterschiedlichen Kanälen
Bei einer Leitachse, deren Folgeachsen einem anderen Kanal als dem Kanal der Leitachse zugeordnet sind, wirkt die Einstellung zur Beschleunigung der Abarbeitungsgeschwindigkeit (MD22601 (Seite 84)\$MC_SERUPRO_SPEED_FACTOR) nicht:
- Mitschleppen
Die Funktion Mitschleppen (TRAILON) wird von SERUPRO unterstützt.
Weitere Informationen
Weitere Informationen zu Mitschleppen mit TRAILON, TRAILOF finden sich in:
 - Funktionshandbuch Achsen und Spindeln; Achskopplungen
 - Programmierhandbuch Arbeitvorbereitung; Achskopplungen
- Gantry-Achsen
Die Funktion Gantry-Achsen wird von SERUPRO unterstützt.
Weitere Informationen
Weitere Informationen zur Funktionalität von Gantry-Achsen finden sich im Funktionshandbuch Achsen und Spindeln; Gantry-Achsen
- Tangentialsteuerung
Die Funktion tangentiales Nachführung einzelner Achsen wird von SERUPRO unterstützt.
Weitere Informationen
Weitere Informationen zur Tangentialsteuerung finden sich im Funktionshandbuch Sonderfunktionen; Tangentialsteuerung

2.8.8.7 Achsfunktionen

SERUPRO Bedingungen

Bei Achsfreigabe, Autarken Achsvorgängen und Achstausch müssen die besonderen Bedingungen für SERUPRO berücksichtigt werden.

Achsfreigabe

Das axiale NST DB31, ... DBX3.7 ("Programmtest Achs-/Spindel Freigabe") beeinflusst die Achsfreigaben, wenn an die Maschine keine Reglerfreigabe gegeben werden soll oder kann und wirkt nur während Programmtest oder SERUPRO aktiv ist.

Es besteht die Möglichkeit über das Nahtstellensignal PLC → NC DB31, ... DBX3.7 (Programmtest Achs-/Spindel Freigabe) diese Freigabe zu geben. Fehlt bei

Programmtest oder SERUPRO die reale Reglerfreigabe, so hat das folgenden Einfluss auf Achsen/Spindeln:

- Sobald der simulierte Programmablauf eine Achse/Spindel verfahren möchte, wird die Meldung "Warten auf Achsfreigabe" oder "Warten auf Spindelfreigabe" angezeigt und die Simulation stoppt.
- Wird während einer simulierten Verfahrbewegung das NC/PLC-Nahtstellensignal DB31, ... DBX3.7 (Programmtest Achs-/Spindel Freigabe) wieder weggenommen, wird der Alarm 21612: "Kanal %1 Achse %2 NC/PLC-Nahtstellensignal 'Reglerfreigabe' während der Bewegung zurückgesetzt" ausgelöst.

Autarke Achsvorgänge

Autarke Einzelachsvorgänge sind vom PLC kontrollierte Achsen, die bei SERUPRO auch simuliert werden. Damit wird während SERUPRO wie im normalen Ablauf, die PLC die Kontrolle einer Achse übernehmen oder abgeben. Diese Achse kann gegebenenfalls auch über FC18 verfahren werden. Die PLC übernimmt die Kontrolle der Achse vor dem Anfahrtsatz und ist für die Positionierung dieser Achse verantwortlich. Dies gilt für alle Satzsuchlauftypen.

Weitere Informationen

Weitere Informationen zu autarken Einzelachsvorgängen finden Sie im Funktionshandbuch Achsen und Spindeln; Positionierachsen

Achstausch

Problem: Ein Programm verfährt eine Achse und gibt sie vor dem Zielsatz mit WAITP(X) ab. Damit unterliegt X nicht dem REPOS und die Achse wird beim SERUPRO-Anfahren nicht berücksichtigt.


Über das Maschinendatum MD11470 \$MN_REPOS_MODE_MASK kann für SERUPRO-REPOS folgendes Verhalten erzielt werden:

Die neutralen Achsen werden als "Kommando-Achsen" im SERUPRO-REPOS verfahren. Die Achse interpoliert ohne Bahnzusammenhang, auch wenn sie zuletzt als Bahnachse programmiert worden war. In diesem Fall ergibt sich die Geschwindigkeit aus MD32060 \$MA_POS_AX_VELO. Nach dem SERUPRO-Anfahren ist diese Achse wieder neutral.

Neutrale Achsen, die trotzdem nicht repositioniert werden dürfen, müssen mit dem axialen NC/PLC-Nahtstellensignal "REPOSDELAY" beaufschlagt werden. Damit wird ihre REPOS Bewegung gelöscht.

Beispiel:

Nach SERUPRO wird via Technologiezyklen in der Synchronaktion eine Achse bewusst bewegt. Die Kommando-Achsen werden immer im Anfahrtsatz und nie im Zielsatz bewegt. Der Zielsatz kann erst eingewechselt werden, wenn alle Kommando-Achsen zu Ende gefahren sind.

 VORSICHT
Die PLC kontrollierte Achse wird nicht repositioniert
Achsen, die mit RELEASE(X) vor dem Zielsatz freigegeben worden sind, werden nicht repositioniert.

2.8.8.8 Getriebestufenwechsel

Abläufe

Der Getriebestufenwechsel (GSW) erfordert von der NC physikalische Bewegungen, um einen neuen Gang einlegen zu können.

Im SERUPRO-Vorgang ist ein Getriebestufenwechsel nicht erforderlich und wird wie folgt durchgeführt:

Manche Getriebe können nur NC-geführt gewechselt werden, da entweder die Achse pendeln, oder vorher eine bestimmte Position angefahren werden muss.

Der Getriebestufenwechsel kann im MD35035 \$MA_SPIND_FUNCTION_MASK mit den Bits 0 bis 2 selektiv für DryRun, Programmtest und SERUPRO unterdrückt werden.

Der GSW muss dann im REPOS nachgeholt werden und funktioniert auch, wenn die zugehörige Achse am Zielsatz im "Drehzahlsteuerbetrieb" sein soll. Andernfalls wird der automatische GWS mit einem Alarm abgelehnt, wenn im Teileprogramm zwischen dem GWS und dem Zielsatz die Achse u. a. in eine Transformation oder Kopplung aufgenommen worden war.

Hinweis

Weitere Informationen zum Getriebestufenwechsel bei DryRun, Programmtest und SERUPRO siehe Funktionshandbuch *Achsen und Spindeln* unter *Spindeln*.

2.8.8.9 Überlagerte Bewegung

Nur SERUPRO

Werden "Überlagerte Bewegungen" verwendet, so kann nur der Satzsuchlauf über Programmtest (SERUPRO) verwendet werden, da dabei die überlagerten Bewegungen im Hauptlauf entsprechend interpoliert werden. Dies gilt insbesondere für \$AA_OFF.

Geschwindigkeitsprofil statt maximale Achsgeschwindigkeit

Im Programmtest muss ein Geschwindigkeitsprofil verwendet werden, welches es erlaubt "Überlagerte Bewegungen" im Hauptlauf zu interpolieren. Es kann somit nicht mit der maximalen Achsgeschwindigkeit interpoliert werden.

Die Achsgeschwindigkeit wird im Modus "Probelaufvorschub" über SD42100 \$SC_DRY_RUN_FEED eingestellt.

Die Geschwindigkeit des SERUPRO-Vorgangs wird über MD22600 \$MC_SERUPRO_SPEED_MODE ausgewählt.

2.8.8.10 NC/PLC-Nahtstellensignale

REPOS-Verschiebung vorhanden

Hat sich während SERUPRO für eine Achse eine REPOS-Verschiebung ergeben, wird dies mit Abschluss des SERUPRO-Vorganges über die axiale NC/PLC-Nahtstelle angezeigt:

DB31, ...DBX70.0 == 1 (REPOS-Verschiebung vorhanden)

Gültigkeitsbereich der REPOS-Verschiebung

Mit dem Start eines SERUPRO-ASUP oder NC-Start zum Fortsetzen der Bearbeitung, wird die REPOS-Verschiebung ungültig:

DB31, ... DBX70.1 == 1 (REPOS-Verschiebung ungültig)

Zwischen dem Abschluss des SERUPRO-Vorganges und NC-Start zum Fortsetzen der Bearbeitung, kann die Achse in der Betriebsart JOG manuell oder über das PLC-Anwenderprogramm mittels FC 18 verfahren werden. Wird dabei die REPOS-Verschiebung vollständig herausgefahren, wird das Nahtstellensignal zurückgesetzt.

2.8.8.11 Flexibilisierung der Grundeinstellung

Grundeinstellung / SERUPRO-Grundeinstellung

Mit dem Maschinendatum MD20112 \$MC_START_MODE_MASK wird die Grundstellung der Steuerung bei Teileprogrammstart bezüglich der G-Codes (insbesondere die aktuelle Ebene und einstellbare Nullpunktverschiebung), Werkzeuglängenkorrektur, Transformation und Achskopplungen festgelegt. Speziell für den SERUPRO-Vorgang besteht die Möglichkeit mit MD22620 \$MC_ENABLE_START_MODE_MASK_PRT eine zum normalen Teileprogrammstart abweichende Grundeinstellung zu wählen. Die neue Einstellung muss dabei hinterlegt sein im Maschinendatum:

MD22620 \$MC_START_MODE_MASK_PRT

Die Bedeutung der jeweiligen Bits von MD22620 ist identisch mit denen von MD20112 \$MC_START_MODE_MASK.

Beispiel:

Beim Teileprogrammstart bleibt die Synchronspindelkopplung zu Beginn des SERUPRO Vorgangs erhalten.

\$MC_START_MODE_MASK = 'H400'	; nicht projektierte Syn-
\$MC_START_MODE_MASK_PRT = 'H00'	chronspindelkopplung
\$MC_ENABLE_START_MODE_MASK_PRT = 'H01'	; wird ausgeschaltet
	; bleibt aktiv
	; \$MC_START_MODE_MASK_PRT
	wird bei SERUPRO ausgewertet
	statt \$MC_START_MODE_MASK

2.8.8.12 Kompressor-Funktionen (COMPON, COMPCURV, COMPCAD, COMPSURF)

- Wenn bei Satzsuchlauf **Typ 2** oder **Typ 4** (Satzsuchlauf **mit Berechnung** an ...) der Zielsatz in einem Programmabschnitt liegt, in dem eine Kompressor-Funktion (COMPON, COMPCURV, COMPCAD, COMPSURF) aktiv ist, werden beim Wiederanfahren an die Kontur Positionen auf der durch den **Kompressor** berechneten Bahn angefahren. Diese Positionen müssen nicht exakt mit den Positionen auf der im Teileprogramm programmierten Bahn übereinstimmen.
- Wenn bei der Komprimierung im Teileprogramm programmierte Sätze entfallen, werden diese Sätze im Satzsuchlauf nicht als Zielsatz gefunden. Es wird Alarm 15370 "Suchziel bei Satzsuchlauf nicht gefunden" ausgegeben.

2.8.9 Systemvariable

Übersicht der für SERUPRO relevanten Systemvariablen:

Systemvariable	Bedeutung
\$AC_ASUP, Bit 20	ASUP Aktivierungsgrund: \$AC_ASUP, Bit 20 == 1 ⇒ System-ASUP aktiv, Grund: SERUPRO-Suchziel erreicht
\$AC_SERUPRO	SERUPRO-Status: \$AC_SERUPRO == 1 ⇒ SERUPRO ist aktiv
\$P_ISTEST	Programmtest-Status: SERUPRO aktiv ⇒ \$P_ISTEST == 1
\$P_SEARCHL	Zuletzt aktiver Satzsuchlauf-typ: \$P_SEARCHL == 5 von SERUPRO Anfang bis Reset oder Programmende
\$AC_REPOS_PATH_MODE	REPOS-Mode zum Wiederanfahren der Kontur nach erfolgtem SERUPRO

2.9 Programmbetrieb

2.9.1 Programmbetrieb

Definition

Programmbetrieb liegt vor, wenn in den Betriebsarten AUTOMATIK bzw. MDA NC-Programme bzw. NC-Programmsätze abgearbeitet werden.

NC/PLC-Nahtstellensignale

Der Programmbetrieb kann vom PLC-Anwenderprogramm aus über BAG- und Kanal-spezifische NC/PLC-Nahtstellensignale beeinflusst werden bzw. gibt entsprechende Rückmeldungen an das PLC-Anwenderprogramm zurück.

Weitere Informationen

Eine Übersicht der NC/PLC-Nahtstellensignale findet sich im Funktionshandbuch PLC; Nahtstellensignale

2.9.2 Grundstellungen

Für jeden Kanal können Grundstellungen im über kanalspezifische Maschinendaten vorgegeben werden. Diese Grundstellungen wirken u. a. auf die G-Gruppen sowie auf die Hilfsfunktionsausgabe.

Hilfsfunktionsausgabe

Die zeitliche Ausgabe der Hilfsfunktionen kann über die Maschinendaten AUXFU_x_SYNC_TYPE (MD22200, 22210, 22220, 22230, 22240, 22250, 22260), (Ausgabezeitpunkt der M, S, T, H, F, D, E-Funktionen) vorbestimmt werden. Nähere Erläuterungen dazu siehe Kapitel "H2: Hilfsfunktionsausgaben an PLC (Seite 647)".

G-Gruppen

Für jede der vorhandenen G-Gruppen kann über MD20150 \$MC_GCODE_RESET_VALUES (Löschstellung der G-Gruppen) eine Programmiergrundstellung vorgegeben werden. Diese Programmiergrundstellung ist automatisch beim Programmstart bzw. im Reset-Zustand wirksam, bis sie durch einen G-Befehl derselben G-Gruppe abgewählt wird.

Über das MD22510 \$MC_GCODE_GROUPS_TO_PLC (G-Befehle, die bei Satzwechsel/RESET an Nahtstelle NC-PLC ausgegeben werden) kann die Ausgabe der G-Befehle an die PLC-Nahtstelle aktiviert werden.

Weitere Informationen

Eine Auflistung der G-Gruppen mit den zugehörigen G-Befehlen finden sich im Programmierhandbuch Grundlagen

Grundkonfigurationen des NC-Sprachumfangs

Es können bestimmte Grundkonfigurationen des NC-Sprachumfangs über Maschinendaten projektierbar generiert werden. Damit wird speziell für den Anwender unter Berücksichtigung seiner benötigten Optionen und Funktionen der NC-Sprachumfang einheitlich auf Ihm zugeschnitten konfiguriert.

2.9.2.1 Maschinendaten

NC-Sprachumfang

Wie mit Sprachbefehlen nicht aktiver Optionen und Funktionen verfahren werden soll, wird mit folgendem Maschinendatum eingestellt:

MD10711 \$MN_NC_LANGUAGE_CONFIGURATION = <Wert>

Wert	Bedeutung
0	Alle Sprachbefehle sind bekannt. Erst bei der Ausführung wird erkannt, ob die benötigte Funktion aktiviert ist.
1	Alle Sprachbefehle sind bekannt. Sprachbefehle von nicht freigegebenen Optionen werden bei der Programminterpretation erkannt → Alarm 12553 "Option/Funktion ist nicht aktiv"
2	Es sind nur die Sprachbefehle der freigegebenen Optionen bekannt. Sprachbefehle von nicht freigegebene Optionen werden bei der Programminterpretation erkannt → Alarm 12550 "Name nicht definiert oder Option/Funktion nicht vorhanden"
3	Alle Sprachebefehle sind bekannt. Sprachbefehle nicht aktivierter Funktionen werden bei der Programminterpretation erkannt → Alarm 12553 "Option/Funktion ist nicht aktiv" Beispiel 1. Option für die Zylindermanteltransformation gesetzt 2. Die Zylindermanteltransformation ist im Maschinendatum MD24100 \$MC_TRAOF_TYPE_1 nicht aktiviert 3. Bereits die Programmierung des Befehls TRACYL führt zum Alarm 12553.
4	Es sind nur die Sprachbefehle der aktiven Funktionen bekannt. Sprachbefehle nicht aktiver Funktionen werden nicht erkannt → Alarm 12550 "Name nicht definiert oder Option/Funktion nicht vorhanden" Hinweis Ob der betreffende Sprachbefehle generell in der Siemens Sprache oder nur auf der entsprechenden Anlage nicht vorhanden ist, kann in diesem Fall nicht unterschieden werden.

2.9.2.2 Programmierung

Mit der Funktion "STRINGIS(...)" wird geprüft, ob die angegebene Zeichenkette als Element der NC-Programmiersprache im aktuellen Sprachumfang zur Verfügung steht.

Folgende Elemente der NC-Programmiersprache können geprüft werden:

- G-Befehle aller existierenden G-Gruppen
- DIN- oder NC-Adressen
- Funktionen
- Prozeduren
- Schlüsselworte
- Systemdaten wie 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

Definition

INT STRINGIS (STRING <Name>)

Syntax

<Rückgabewert> = STRINGIS (<Name>)

Bedeutung

STRINGIS ():	Prüffunktion mit Rückgabewert
<Name>:	Zu prüfende Zeichenkette

<Rückgabewert >:	Der Rückgabewert ist in den ersten drei Dezimalstellen codiert y_{xx}		
	000	Der angegebene String ist kein Element des aktuellen Sprachumfangs ¹⁾	
	100	Der angegebene String ist Element des aktuellen Sprachumfangs, aber aktuell nicht programmierbar (Option/Funktion ist inaktiv)	
	2xx	Der angegebene String ist ein programmierbares Element aktuellen Sprachumfangs (Option/Funktion ist aktiv). Detailinformationen sind in der 1. und 2. Dezimalstelle xx enthalten:	
		xx	Bedeutung
		01	DIN-Adresse oder NC-Adresse ²⁾
		02	G-Befehl (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 NC 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-Befehl (ISO Sprachmodus muss aktiv ist)		
400	Der angegebene String ist eine NC-Adresse, die weder als DIN-Adresse bzw. NC-Adresse (xx==01) noch als Makroname (xx==10) erkannt wurde und die nicht G oder R ist ²⁾		
y00	Keine spezifische Zuordnung möglich		

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.

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").

3) Namen von Zyklenparametern können mit STRINGIS **nicht** geprüft werden.

4) Als Makro definierte Adress z.B. G, H, M, L werden als Makro identifiziert

Beispiele

In den folgenden Beispielen wird angenommen, dass das angegebene NC-Sprachelement, sofern nicht besonders vermerkt, in der Steuerung prinzipiell programmierbar ist.

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")
```
4. Der String "INVCW" ist als benannter G-Befehl 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 (Werkzeuglängenkomponente):


```
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 NC bekannter Befehl, GUD-Variable, Makro- oder Zyklus-Name:


```
000 == STRINGIS ("XYZ")
```

Randbedingungen

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 Mode

Die Funktion "ISO Mode" ist aktiv, wenn:

Ist die Funktion "ISO Mode" aktiv, überprüft `STRINGIS()` den angegebenen String zuerst als SINUMERIK G-Befehl:

- MD18800 `$MN_MM_EXTERN_LANGUAGE` (Aktivierung externer NC-Sprachen)
- MD10880 `$MN_MM_EXTERN_CNC_SYSTEM` (zu adaptierendes Steuerungssystem)

Bei aktivem "ISO Mode" überprüft `STRINGIS()` die angegebene Zeichenkette zuerst auf Zugehörigkeit zum SINUMERIK G-Befehl.

Ist der String kein SINUMERIK G-Befehl, wird er anschließend auf Zugehörigkeit zum ISO G-Befehl überprüft.

Programmierte Umschaltung des ISO Mode

Programmierte Umschaltungen durch die Befehle `G290` (SINUMERIK Mode) und `G291` (ISO Mode) haben auf `STRINGIS()` keine Auswirkung.

2.9.3 Anwahl und Start eines NC-Programms

NC/PLC-Nahtstellensignale

Anwahl

Die Anwahl eines NC-Programms kann nur erfolgen, wenn sich der betreffende Kanal im Reset-Zustand befindet.

- DB21, ... DBX35.7 == 1 (Reset)

Start

Ein NC-Programm wird durch zwei verschiedene Ereignisse gestartet:

1. DB21, ... DBX7.1 = 1 (NC-Start), das Signal wird üblicherweise durch Betätigung der Taste "NC-Start" der Maschinensteuertafel ausgelöst.
2. Befehl `START` in einem NC-Programm eines anderen aktiven Kanals. Der Kanal muss sich dazu in der Betriebsart AUTOMATIK oder MDA und im Zustand "Reset" oder "unterbrochen" befinden
 - DB21, ... DBX35.7 == 1 (Reset)
 - DB21, ... DBX35.6 == 1 (unterbrochen)

Ausgangsbedingungen

Ein NC-Programm kann nur gestartet werden, wenn folgende Ausgangsbedingungen erfüllt sind.

- DB11 DBX4.4 == 1 (BAG-betriebsbereit)
- DB11 DBX0.7 == 0 (BAG-Reset)
- DB21, ... DBX1.7 == 0 (Programmtest aktivieren)
- DB21, ... DBX7.0 == 0 (NC-Start-Sperre)
- DB21, ... DBX7.2 == 0 (NC-Stop an Satzgrenze)
- DB21, ... DBX7.3 == 0 (NC-Stop)
- DB21, ... DBX7.4 == 0 (NC-Stop Achsen plus Spindel)
- DB21, ...:DBX7. 7 == 0 (Reset)
- DB10 DBX56.1 == 0 (Not-Halt)
- Es dürfen **keine** Achs- oder NC-spezifischen Alarm anstehen

Ausführung des Kommandos

Das Teileprogramm bzw. der Teileprogrammsatz wird automatisch abgearbeitet und die folgenden Nahtstellensignale werden gesetzt:

- DB21, ... DBX35.5 (Kanalzustand aktiv)
- DB21, ... DBX35.0 (Programmzustand läuft)

Das Programm wird solange bearbeitet, bis das Programmende erreicht bzw. der Kanal durch ein STOP- oder RESET-Kommando unterbrochen bzw. abgebrochen wird.

Weitere Informationen

Eine ausführliche Beschreibung der Nahtstellensignale findet sich im Funktionshandbuch PLC.

Eine ausführliche Beschreibung der NC-Variablen findet sich im Listenhandbuch NC-Variablen

Alarmer

Das START-Kommando ist unter bestimmten Umständen nicht wirksam und es tritt dann einer der folgenden Alarmer auf:

- 10200 "Kein NC-Start bei aktivem Alarm erlaubt"
- 10202 "Kein NC-Start bei aktivem Kommando erlaubt"
- 10203 "Kein NC-Start bei nicht referenzierten Achsen erlaubt"

Weitere Informationen

Diagnosehandbuch Alarmer

2.9.4 Programmunterbrechung

NC/PLC-Nahtstellensignale

Voraussetzungen

Eine Programmunterbrechung wird nur ausgeführt, wenn der Kanal und das NC-Programm aktiv sind:

- DB21, ... D35.5 == 1 (Kanal: aktiv)
- DB21, ... D35.0 == 1 (Programm: läuft)

Programmunterbrechung

Über folgende Ereignisse kann eine Programmbearbeitung unterbrochen werden:

- DB21, ... DBX7.2 == 1 (NC-Stop an Satzgrenze)
- DB21, ... DBX7.3 == 1 (NC-Stop)
- DB21, ... DBX7.4 == 1 (NC-Stop Achsen plus Spindel)
- DB21, ... DBX2.0 == 1 (Einzelsatz)
- Programmierter Befehl M00 oder M01 im abgearbeiteten NC-Programm

Anschließend befindet sich der Kanal und das NC-Programm im Zustand "unterbrochen":

- DB21, ... D35.6 == 1 (Kanal unterbrochen)
- DB21, ... D35.3 == 1 (Programm unterbrochen)

Weitere Informationen

Eine ausführliche Beschreibung der Nahtstellensignale findet sich im Funktionshandbuch PLC

Ablauf

Mit dem Erkennen einer Programmunterbrechung werden folgende Aktionen durchgeführt:

- Unterbrechung der Programmbearbeitung:
 - An der nächsten Satzgrenze bei folgenden Ereignissen: "NC-Stop an Satzgrenze", M00, M01 oder Einzelsatz
 - Sofort: Alle anderen Ereignisse
- Die verfahrenen Achsen des Kanals, werden über Bremsrampe stillgesetzt. Das Abbremsen der Achsen kann sich dabei über mehrere Sätze erstrecken.
- Der Satzzeiger zeigt den aktuellen Satz an der Unterbrechungsstelle an.
- Die bis zur Unterbrechungsstelle noch nicht ausgegebenen Hilfsfunktionen werden nicht mehr ausgegeben.

Möglichkeiten im unterbrochenen Zustand

Während einer Teileprogrammunterbrechung können diverse Funktionen im Kanal ausgeführt werden z.B.:

- **Überspeichern**
- **Satzsuchlauf**
Weitere Informationen finden Sie unter Satzsuchlauf Typ 1, 2 und 4 (Seite 56).
- **Wiederanfahren an die Kontur (REPOS)**
Weitere Informationen finden Sie unter Anwenderspezifisches ASUP für RET und REPOS (Seite 167).
- **Orientierter Werkzeugrückzug**
Weitere Informationen
 - Programmierhandbuch Arbeitsvorbereitung, Werkzeugkorrekturen
 - Funktionsbeschreibung Werkzeuge, Werkzeugkorrekturen (W1) > Orientierbare Werkzeugträger
- **ASUP** (siehe Kapitel "Asynchrone Unterprogramme (ASUPs) (Seite 152)")
- **DRF-Funktion, Verschieben des Werkstücknullpunktes**
Weitere Informationen
Funktionshandbuch Achsen und Spindeln; Manuelles Verfahren
- **Fortsetzen des unterbrochenen NC-Programms**
 - Befehl `START` aus einem anderen Kanal
Weitere Informationen
Programmierhandbuch Arbeitsvorbereitung, Flexible NC-Programmierung > Programmkoordinierung (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)
 - NC/PLC-Nahtstelle:
DB21, ... DBX7.1 = 1 (NC-Start)

Siehe auch

Satzsuchlauf Typ 5 (SERUPRO) (Seite 70)

2.9.5 Kanal-Reset**Funktion**

Durch einen Kanal-Reset wird ein in der Betriebsart Automatik ein in Abarbeitung befindliches NC-Programm bzw. in der Betriebsart MDA ein in Abarbeitung befindlicher Programmsatz abgebrochen.

Das NC-Programm bzw. der Programmsatz kann an der Unterbrechungsstelle nicht mehr fortgesetzt werden. Nach Abschluss des Kanal-Resets befinden sich alle Achsen und Spindeln des Kanals in Zustand "Genauhalt" (Ausnahme: Nachführbetrieb).

Ablauf

Durch NC-Reset werden folgende Aktionen im Kanal durchgeführt:

- Die Programmaufbereitung wird sofort gestoppt.
- Achsen und Spindeln werden über ihre parametrisierten Bremsrampen bis zum Stillstand abgebremst.
- Die noch nicht ausgegebenen Hilfsfunktionen des aktuellen Satzes werden nicht mehr an die PLC ausgegeben.
- Der Satzzeiger wird auf den Anfang des angewählten NC-Programms zurückgesetzt.
- Alle Reset-Alarme des Kanals werden aus der Anzeige gelöscht.

Regeln

- Ein Kanal-Reset wird in jedem Kanalzustand ausgeführt.
- Ein Kanal-Reset wird von keinem anderen Kommando abgebrochen.

NC/PLC-Nahtstellensignale

Anforderung: Kanal-Reset

Ein Kanal-Reset wird über folgende NC/PLC-Nahtstellensignale angefordert:

- DB21, ... DBX7.7 = 1 (Reset)

Anforderung: BAG-Reset

Ein BAG-Reset löst Kanal-Reset in allen Kanälen einer BAG aus.

Ein BAG-Reset wird über folgende NC/PLC-Nahtstellensignale angefordert:

- DB11, ... DBX0.7 = 1 (BAG-Reset)

Rückmeldung: Kanal-Reset abgeschlossen

- DB21, ... DBX35.7 == 1 (Kanalzustand Reset)

Rückmeldung: BAG-Reset abgeschlossen

- DB11, ... DBX6.5 == 1 (BAG reseted)

Weitere Informationen

Eine ausführliche Beschreibung der Nahtstellensignale findet sich im Funktionshandbuch PLC.

2.9.6 Programmzustand

Für jeden Kanal wird der Zustand des angewählten NC-Programms in der Nahtstelle angezeigt.

In den Betriebsarten AUTOMATIK und MDA können alle Programmzustände auftreten. In allen anderen Betriebsarten oder Maschinenfunktionen ist der Programmzustand abgebrochen oder unterbrochen.

NC/PLC-Nahtstellensignale

An der NC/PLC-Nahtstelle (DB21, ...) werden folgende ProgramMZustände angezeigt:

- DB21, ... DBX35.4 ("abgebrochen")
- DB21, ... DBX35.3 ("unterbrochen")
- DB21, ... DBX35.2 ("angehalten")
- DB21, ... DBX35.1 ("warten")
- DB21, ... DBX35.0 ("läuft")

Eine ausführliche Beschreibung der Nahtstellensignale findet sich im Listenhandbuch NC-Variablen und Nahtstellensignale.

Zustandsänderungen

Der ProgramMZustand wird durch Kommandos, NC/PLC-Nahtstellensignale und AlarMe beeinflusst. Ausgehend vom ProgramMZustand "läuft" zeigt die Tabelle den Folgezustand aufgrund des jeweiligen Ereignisses.

Ausgangszustand des Programms: "läuft"					
Ereignis	Folgezustand des Programms				
	ab-gebr.	un-terbr.	angeh.	warten	läuft
DB21, ... DBX7.7 (Reset)	x				
DB21, ... DBX7.3 (NC-Stop)			x		
DB21, ... DBX7.2 (NC-Stop an Satzgrenze)			x		
DB21, ... DBX7.4 (NC-Stop Achsen u. Spindeln)			x		
DB21, ... DBX6.1 (Einlesesperre)					x
DB21, ... DBX6.0 (Vorschub Halt)					x
DB21, ... DBX12.3 / 16.3 / 20.3 (Vorschub Halt, GeoAchse 1 / 2 / 3)					x
DB21, ... DBB4; Vorschuboverride = 0%					x
DB31, ... DBX4.3 (Vorschub Halt/Spindel Halt)					x
DB21, ... DBX194.2 / DBX197.6 (M02 / M30 im Satz)	x				
DB21, ... DBX194.0 / DBX194.1 (M00 / M01 im Satz)			x		
DB21, ... DBX0.4 (Einzelsatz)			x		
DB21, ... DBX6.2 (Restweg löschen)					x
Hilfsfunktion an PLC ausgegeben, aber noch nicht quittiert			x		
WAIT-Befehl im Programm				x	
Alarm mit Systemreaktion "NOREADY"		x			

2.9.7 Kanalzustand

Für jeden Kanal wird in allen Betriebsarten der Kanalzustand an der NC/PLC-Nahtstelle (DB21, ...) angezeigt..

NC/PLC-Nahtstellensignale

An der NC/PLC-Nahtstelle werden die Kanalzustände durch folgende Signale angezeigt:

- DB21, ... DBX35.7 ("Reset")
- DB21, ... DBX35.6 ("unterbrochen")
- DB21, ... DBX35.5 ("aktiv")

Eine ausführliche Beschreibung der Nahtstellensignale findet sich im Listenhandbuch NC-Variablen und Nahtstellensignale.

Zustandsänderungen

Der Kanalzustand wird durch Kommandos und NC/PLC-Nahtstellensignale beeinflusst. Augehend vom Kanalzustand "aktiv" zeigt die Tabelle den Folgezustand aufgrund des jeweiligen Ereignisses.

Ausgangszustand des Kanals: "aktiv"			
Ereignis	Folgezustand des Kanals		
	"Reset"	"unterbrochen"	"aktiv"
DB21, ... DBX7.7 (Reset)	x		
DB21, ... DBX7.3 (NC-Stop)		x	
DB21, ... DBX7.2 (NC-Stop an Satzgrenze)		x	
DB21, ... DBX7.4 (NC-Stop Achsen u. Spindeln)		x	
DB21, ... DBX6.1 (Einlesesperre)			x
DB21, ... DBX6.0 (Vorschub Halt)			x
DB21, ... DBX12/16/20.3 (Vorschub Halt, GeoAchse 1/2/3)			x
DB21, ... DBB4; Vorschuboverride = 0%			x
DB31, ... DBX4.3 (Vorschub Halt/Spindel Halt)			x
DB21, ... DBX194.2/DBX197.6 (M02/M30 im Satz)	x		
DB21, ... DBX194.0/DBX194.1 (M00/M01 im Satz)		x	
DB21, ... DBX0.4 (Einzelsatz)		x	
DB21, ... DBX6.2 (Restweg löschen)			x
Hilfsfunktion an PLC ausgegeben, aber noch nicht quittiert			x
WAIT-Befehl im Programm			x

Der Kanalzustand "aktiv" wird erreicht, wenn ein NC-Programm oder NC-Programmsatz abgearbeitet wird oder wenn in der Betriebsart JOG Achsen verfahren werden.

2.9.8 Reaktionen auf Bedienungs- und Programmaktionen

Zustandsübergänge

Die folgende Tabelle zeigt die Kanal- und Programmzustände, die nach bestimmten Bedien- oder Programmaktionen auftreten.

Im linken Teil der Tabelle sind die verschiedenen Zustände des Kanals und des im Kanal angewählten Programms und die aktive Betriebsart aufgeführt.

Im rechten Teil der Tabelle sind die Bedien-/Programmaktionen und der Folge-Zustand aufgeführt.

Zu- stand	Kanalzustand			Programmzustand					Betriebsart			Bedien- oder Programmaktion => Folge-Zustand
	R	U	A	R	U	S	W	A	A	M	J	
1		x						x	x			RESET => 4
2		x						x		x		RESET => 5
3		x						x			x	RESET => 6
4	x			x					x			NC-Start => 13; BA-Wechsel => 5 bzw. 6
5	x			x						x		NC-Start => 14; BA-Wechsel => 4 bzw. 6
6	x			x							x	Richtungstaste => 15; BA-Wechsel => 4 bzw. 5
7		x		x						x		NC-Start => 14
8		x		x							x	NC-Start => 15
9		x			x				x			NC-Start => 13; BA-Wechsel => 10 bzw. 11
10		x			x					x		NC-Start => 16; BA-Wechsel => 9 bzw. 11
11		x			x						x	Richtungstaste => 17; BA-Wechsel => 9 bzw. 10
12		x				x			x			NC-Start => 13; BA-Wechsel => 10 bzw. 11
13			x					x	x			NC-Stop => 12
14			x	x						x		NC-Stop => 7; bei Satzende => 5
15			x	x							x	NC-Stop => 8; bei JOG-Ende => 6
16			x		x					x		NC-Stop => 10; bei Satzende => 10
17			x		x						x	NC-Stop => 11; bei JOG-Ende => 11
18			x				x		x			RESET => 4; warten auf anderen Kanal => 18

Kanalzustand

R → abgebrochen

U → unterbrochen

A → läuft

Programmzustand

R → abgebrochen

U → unterbrochen

S → angehalten

W → wartet

A → läuft

Betriebsarten

A → Automatik

M → MDA

J → JOG

2.9.9 Zeitdiagramm-Beispiel für einen Programmablauf**Programmcode**

N10 G01 G90 X100 M3 S1000 F1000 M170

N20 M0

2.9 Programmbetrieb

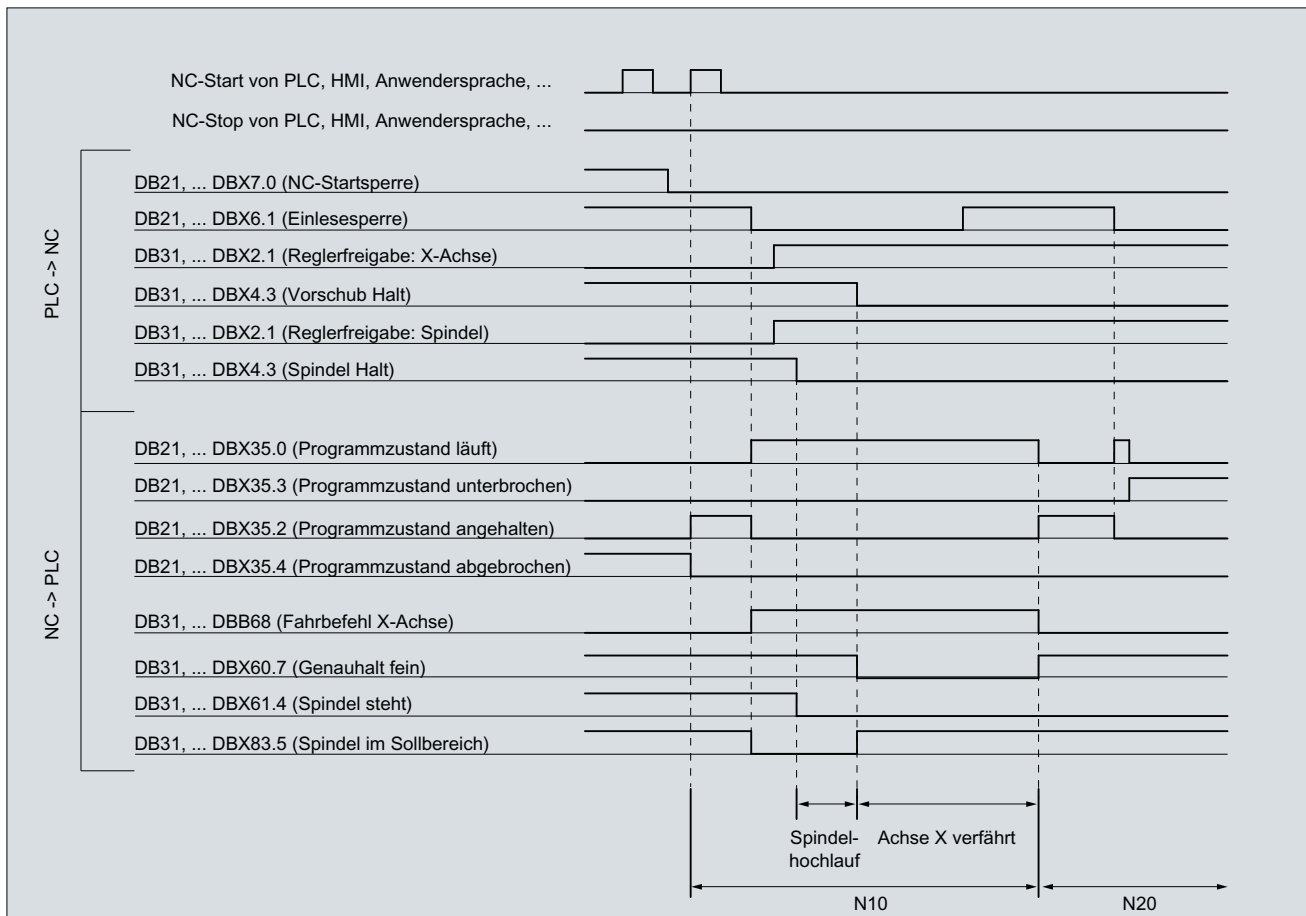


Bild 2-6 Signalverläufe während des Programms

2.9.10 Programmsprünge

2.9.10.1 Rücksprung auf Programmanfang (GOTOS)

Funktion

Mit der Funktion "Rücksprung auf den Programmanfang" kann aus einem Teileprogramm heraus an den Anfang des Programms zurückgesprungen werden. Das Programm wird daraufhin erneut abgearbeitet.

Im Vergleich zur Funktion "Programmsprünge auf Sprungmarken", mit der ebenfalls ein wiederholtes Abarbeiten eines Teileprogramms realisiert werden kann, bietet die Funktion "Rücksprung auf den Programmanfang" folgende Vorteile:

- Die Programmierung einer Sprungmarke am Programmanfang ist nicht notwendig.
- Der Programmneustart ist steuerbar durch das NC/PLC-Nahtstellensignal: DB21, ... DBX384.0 (Programmverzweigung steuern)

- Der Timer für die Programmlaufzeit kann beim Programmneustart auf "0" zurückgesetzt werden.
- Die Timer für die Werkstückzählung können beim Programmneustart um den Wert "1" erhöht werden.

Anwendungsbeispiel

Die Funktion findet Anwendung, wenn die Bearbeitung nachfolgender Werkstücke durch einen automatischen Programmneustart erfolgen soll, z. B. bei Drehmaschinen mit Stangenlader/-wechsler.

NC/PLC-Nahtstellensignale

Der Rücksprung erfolgt nur, wenn das folgende NC/PLC-Nahtstellensignal gesetzt ist:

DB21, ... DBX384.0 (Programmverzweigung steuern) = 1

Parametrierung

Programmlaufzeit

Die Laufzeit des angewählten NC-Programms wird in der Systemvariablen \$AC_CYCLE_TIME gespeichert. Beim Start eines neuen Programms wird die Systemvariable automatisch auf "0" zurückgesetzt (siehe Kapitel " Programmlaufzeit (Seite 243) ")

Über das folgende Maschinendatum kann eingestellt werden, dass die Systemvariable \$AC_CYCLE_TIME auch bei einem Programmneustart durch die Funktion "Rücksprung auf den Programmanfang" auf "0" zurückgesetzt wird:

MD27860 \$MC_PROCESSTIMER_MODE.Bit 8 = <Wert> (Aktivierung der Programmlaufzeit-Messung)

Bit	Wert	Bedeutung
8	0	\$AC_CYCLE_TIME wird durch die Funktion "Rücksprung auf den Programmanfang" nicht auf "0" zurückgesetzt.
	1	\$AC_CYCLE_TIME wird durch die Funktion "Rücksprung auf den Programmanfang" auf "0" zurückgesetzt.

Hinweis

Damit die Einstellung von Bit 8 wirksam werden kann, muss die Messung der aktuellen Programmlaufzeit aktiv sein (MD27860 Bit 1 = 1).

Werkstückzählung

Nachdem das Teileprogrammende (M02 / M30) erreicht ist, werden die aktivierten Werkstückzähler (\$AC_TOTAL_PARTS / \$AC_ACTUAL_PARTS / \$AC_SPECIAL_PARTS) um den Wert "1" erhöht (siehe Kapitel " Werkstückzähler (Seite 254) ").

Über das folgende Maschinendatum kann eingestellt werden, dass die aktivierten Werkstückzähler auch bei einem Programmneustart durch die Funktion "Rücksprung auf den Programmanfang" hochgezählt werden:

MD27880 \$MC_PART_COUNTER.Bit <n> = <Wert> (Aktivierung der Werkstück-Zähler)

Bit	Wert	Bedeutung: Bei einem Programmneustart durch die Funktion "Rücksprung auf den Programmanfang" wird der Werkstückzähler:
7	0	\$AC_TOTAL_PARTS nicht hochgezählt.
	1	\$AC_TOTAL_PARTS hochgezählt.
11	0	\$AC_ACTUAL_PARTS nicht hochgezählt.
	1	\$AC_ACTUAL_PARTS hochgezählt.
15	0	\$AC_SPECIAL_PARTS nicht hochgezählt.
	1	\$AC_SPECIAL_PARTS hochgezählt.

Programmierung

Syntax

GOTOS

Bedeutung

GOTOS:	Rücksprung auf den Anfang des aktuellen Programms	
	Vorlaufstopp:	ja
	Wirksamkeit:	Satzweise

Beispiel

Programmierung	Kommentar
N10 ...	; Programmanfang
...	
IF ...	
N100 GOTOS	Rücksprung auf den Programmanfang (N10)
ENDIF	
...	
RET	

2.9.11 Programmteiwiederholungen

2.9.11.1 Programmierung

Die Programmteiwiederholung ermöglicht die Wiederholung von Programmteilen innerhalb eines NC-Programms.

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. REPEATB: Einzelne Programmzeile wiederholen

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

2. REPEAT + Sprungmarke: Programmbereich zwischen Sprungmarke und REPEAT-Anweisung wiederholen

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

3. REPEAT + Sprungmarke_1 + Sprungmarke_2: 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. REPEAT + Sprungmarke + ENDLABEL: 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"> • REPEATB: Zu wiederholende Programmzeile • REPEAT: Beginn des zu wiederholenden Programmbereichs <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, wird in Richtung Programmende gesucht.</p> <p>Ausnahme: Wenn der Programmbereich zwischen Sprungmarke und REPEAT-Anweisung wiederholt werden soll (siehe 2. unter Syntax), muss die mit der <Sprungmarke> gekennzeichnete Programmzeile vor der REPEAT-Anweisung stehen, da in diesem Fall nur in Richtung Programmanfang gesucht wird.</p> <p>Enthält die Zeile mit der <Sprungmarke> weitere Anweisungen, werden diese bei jeder Wiederholung erneut ausgeführt.</p>
ENDLABEL:	<p>Schlüsselwort, welches das Ende eines zu wiederholenden Programmbereichs markiert</p> <p>Enthält die Zeile mit dem ENDLABEL weitere Anweisungen, werden diese bei jeder Wiederholung erneut ausgeführt.</p> <p>ENDLABEL kann mehrfach im Programm verwendet werden.</p>

P:	Adresse zur Angabe der Wiederholungsanzahl Hinweis: Ist kein P=<n> angegeben, wird der zu wiederholende Programmteil genau einmal wiederholt.
<n>:	Wiederholungsanzahl
	Typ: INT Der zu wiederholende Programmteil wird <n> mal wiederholt. Nach der letzten Wiederholung wird das Programm mit der auf den REPEAT-/REPEATB-Befehl folgenden Programmzeile fortgesetzt.

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	; Anfang des Programmabschnitts
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	
N10 Begin: R10=R10+1	; Anfang des Programmabschnitts
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	
N50 X=-R10	

2.9 Programmbetrieb

Programmcode	Kommentar
N60 Y--R10	
N70 END: Z=10	; Ende des Programmabschnitts
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:	; Anfang Programmabschnitt 1
N30 X10	
N40 Y10	
N50 BEGIN2:	; Anfang Programmabschnitt 2
N60 X20	
N70 Y30	
N80 ENDLABEL: Z10	
N90 X0 Y0 Z0	
N100 Z-10	
N110 BEGIN3: X20	; Anfang Programmabschnitt 3
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:	; Anfang Programmabschnitt 1; Bohrpositionen 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	
N70 ENDLABEL:	
N80 POS_2:	; Anfang Programmabschnitt 2; Bohrpositionen 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	

Programmcode	Kommentar
N120 ENDLABEL:	; Ende der Programmabschnitte 1 und 2
N130 BOHRER()	; Bohrzyklus
N140 GEWINDE(6)	; Gewindezyklus
N150 REPEAT POS_1	; Wiederhole Programmabschnitt ab POS_1 einmal bis ENDLABEL.
N160 BOHRER()	; Bohrzyklus.
N170 GEWINDE(8)	; Gewindezyklus.
N180 REPEAT POS_2	; Wiederhole Programmabschnitt ab POS_2 einmal bis ENDLABEL.
N190 M30	

Weitere Informationen

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

2.9 Programmbetrieb

- 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, wird solange bearbeitet, bis das programmierte Programmteilende gefunden wird.
Beispiel:

```
Programcode
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.

2.9.12 Ereignisgesteuerter Programmaufruf (PROG_EVENT)

2.9.12.1 Funktion

Funktion

Die Funktion "Ereignisgesteuerte Programmaufrufe" (PROG_EVENT) startet beim Auftreten eines ausgewählten Ereignisses in der NC ein anwenderspezifisches NC-Programm (PROG_EVENT-Programm).

Anwendungsbeispiele

Grundeinstellung von Funktionen oder Initialisierungen von System- oder Anwendervariablen.

Ereignisse

Die Auswahl der auslösenden Ereignisse erfolgt über das Maschinendatum MD20108 \$MC_PROG_EVENT_MASK (siehe Kapitel "Parametrierung (Seite 130)").

PROG_EVENT-Programm

Über das Maschinendatum MD11620 \$MN_PROG_EVENT_NAME wird der Name des PROG_EVENT-Programms eingestellt (siehe Kapitel "Parametrierung (Seite 130)").

Das PROG_EVENT-Programm wird in dem Kanal ausgeführt, in dem das Ereignis aufgetreten ist.

Das PROG_EVENT-Programm wird mit niedrigster Priorität ausgeführt und kann daher von einem Anwender-ASUP unterbrochen werden.

Bearbeitungsabläufe**Bearbeitungsablauf bei Aktivierung durch Ereignis: Programm-Start**

Ausgangszustand:

Kanal: Reset-Zustand
 Betriebsart: AUTOMATIK (optional: Überspeichern aktiv)
 MDA
 TEACH IN

1. Im Kanal erfolgt ein Programm-Start
2. Initialisierungssequenz mit Auswertung von:
 - MD20112 \$MC_START_MODE_MASK
3. Impliziter Aufruf des PROG_EVENT-Programms als Unterprogramm
4. Bearbeitung des Datenteils des Hauptprogramms
5. Bearbeitung des Programmteils des Hauptprogramms

Bearbeitungsablauf bei Aktivierung durch Ereignis: Programm-Ende

Ausgangszustand

Kanal: Aktiv
 Betriebsart: AUTOMATIK (optional: Überspeichern aktiv)
 MDA
 TEACH IN

1. Im Kanal wird im ausgeführten Programm der Programmende-Satz eingewechselt.
2. Programm-Ende wird ausgeführt, Auswertung folgender Maschinendaten:
 - MD20110 \$MC_RESET_MODE_MASK
 - MD20150 \$MC_GCODE_RESET_VALUES
 - MD20152 \$MC_GCODE_RESET_MODE

3. Impliziter Aufruf des PROG_EVENT-Programms als ASUP
4. Reset wird im Kanal ausgeführt, Auswertung folgender Maschinendaten:
 - MD20110 \$MC_RESET_MODE_MASK
 - MD20150 \$MC_GCODE_RESET_VALUES
 - MD20152 \$MC_GCODE_RESET_MODE

Ablauf bei Aktivierung durch Ereignis: Kanal-Reset

Ausgangszustand:

Kanal: beliebig
Betriebsart: beliebig

1. Steuerung aktiviert Reset-Sequenz mit Auswertung der Maschinendaten:
 - MD20110 \$MC_RESET_MODE_MASK
 - MD20150 \$MC_GCODE_RESET_VALUES
 - MD20152 \$MC_GCODE_RESET_MODE
2. Impliziter Aufruf des PROG_EVENT-Programms als ASUP
3. Steuerung aktiviert Reset-Sequenz mit Auswertung der Maschinendaten:
 - MD20110 \$MC_RESET_MODE_MASK
 - MD20150 \$MC_GCODE_RESET_VALUES
 - MD20152 \$MC_GCODE_RESET_MODE

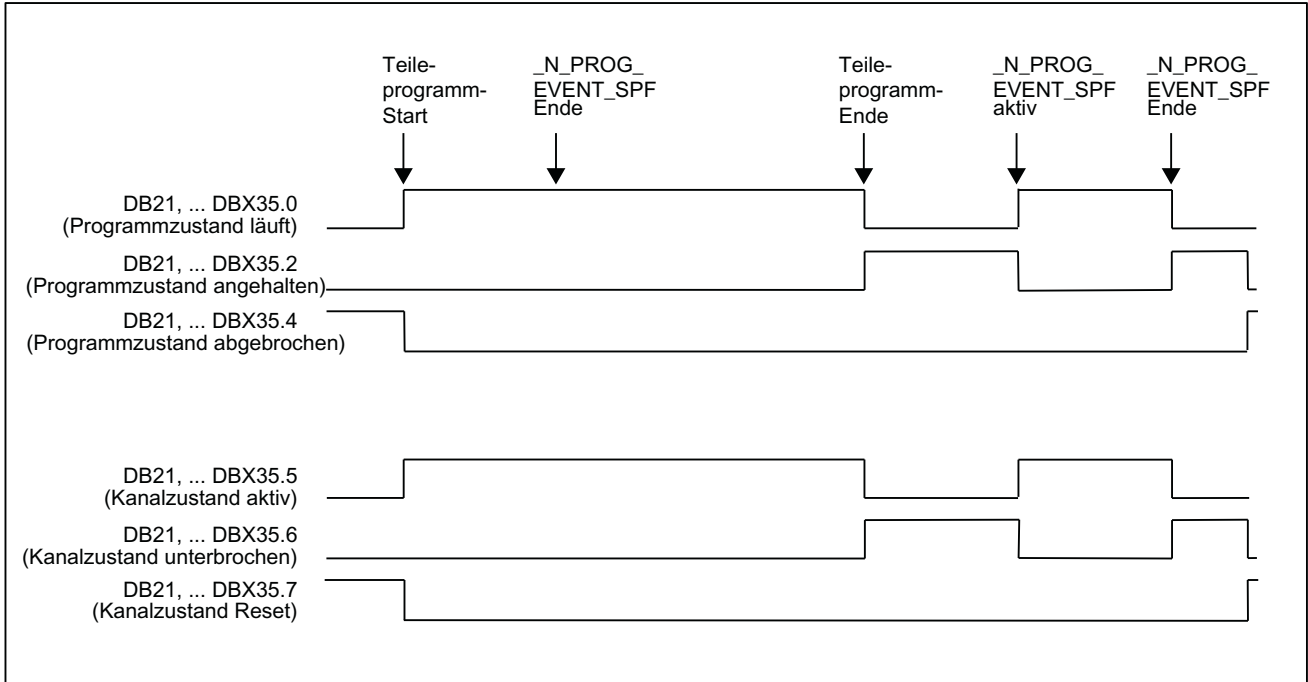
Ablauf bei Aktivierung durch Ereignis: Hochlauf der NC

1. Steuerung aktiviert nach Hochlauf Reset-Sequenz mit Auswertung der Maschinendaten:
 - MD20110 \$MC_RESET_MODE_MASK
 - MD20150 \$MC_GCODE_RESET_VALUES
 - MD20152 \$MC_GCODE_RESET_MODE
2. Impliziter Aufruf des PROG_EVENT-Programms als ASUP
3. Steuerung aktiviert Reset-Sequenz mit Auswertung der Maschinendaten:
 - MD20110 \$MC_RESET_MODE_MASK
 - MD20150 \$MC_GCODE_RESET_VALUES
 - MD20152 \$MC_GCODE_RESET_MODE

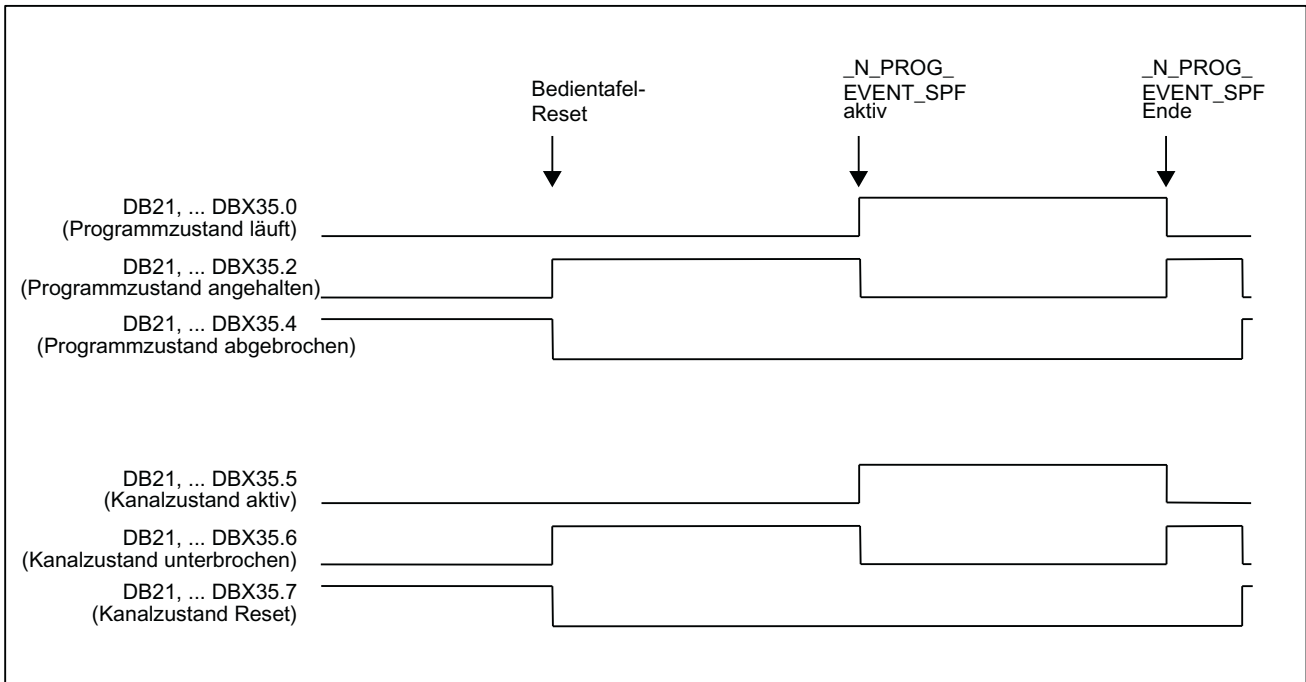
NC/PLC-Nahtstellensignale: Verlauf von "Programmzustand" und "Kanalzustand"

Die nachfolgenden Diagramme zeigen den Verlauf der verschiedenen NC/PLC-Nahtstellensignale für "Programmzustand" und "Kanalzustand" während des Ablaufs des ereignisgesteuerten Programmaufrufs.

Signalverlauf bei Aktivierung durch Programm-Start und Programm-Ende



Signalverlauf bei Aktivierung durch Kanal-Reset



NC/PLC-Nahtstellensignale: DB21, ... DBX35.4 (Programmzustand abgebrochen) und DB21, ... DBX35.7 (Kanalzustand Reset)

- Die Nahtstellensignale werden erst gesetzt, wenn das PROG_EVENT-Programm wieder beendet ist.
- Die Nahtstellensignale werden nicht gesetzt zwischen:
 - Programm-Ende und dem Start des PROG_EVENT-Programms
 - Kanal-Reset und dem Start des PROG_EVENT-Programms

NC/PLC-Nahtstellensignale: DB21, ... DBX376.0 ... 4 (Auslösende Ereignisse)

Das auslösende Ereignis wird über die Nahtstellensignale DB21, ... DBX376.0 ... 4 angezeigt:

Bit	Wert	Ereignis
0	1	Programm-Start aus Kanalzustand "Reset"
1	1	Programm-Ende
2	1	Kanal-Reset
3	1	Hochlauf der NC
4	1	1. Programm-Start nach Satzsuchlauf (siehe "Automatischer Start eines ASUP nach Satzsuchlauf (Seite 63)")

Wird das PROG_EVENT-Programm beendet oder mit Kanal-Reset abgebrochen, werden die Nahtstellensignale wieder gelöscht.

Die Nahtstellensignale stehen mindestens für die Dauer eines PLC-Zyklus an.

2.9.12.2 Parametrierung

Ereignissauswahl

Durch welche Ereignisse das PROG_EVENT-Programm gestartet wird, wird kanalspezifisch eingestellt über das Maschinendatum:

MD20108 \$MC_PROG_EVENT_MASK.Bit <n> = 1

Bit	<Wert>	Bedeutung: Ereignis
0	1	Teileprogramm-Start
1	1	Teileprogramm-Ende
2	1	Kanal-Reset
3	1	Hochlauf
5	1	Safety-Programm im Hochlauf

Hinweis

MD20108 \$MC_PROG_EVENT_MASK wird in der Simulation nicht ausgewertet.

PROG_EVENT-Programm

Das PROG_EVENT-Programm (Voreinstellung: `_N_PROG_EVENT_SPF`) muss geladen und freigegeben sein.

Defaulteinstellung

Defaultmäßig wird bei einem Ereignis das Anwenderprogramm `_N_CMA_DIR/_N_PROG_EVENT_SPF` ausgeführt.

Das PROG_EVENT-Programm muss geladen und freigegeben sein.

Anwenderspezifische Einstellung

Soll bei einem Ereignis ein anderes PROG_EVENT-Programm als das aus der Defaulteinstellung ausgeführt werden, muss der Name die NC-Programms in folgendens Maschinendatum eingetragen werden:

MD11620 \$MN_PROG_EVENT_NAME = <Programmname>

Suchpfad

Das PROG_EVENT-Programm muss sich in einem der folgenden Zyklen-Verzeichnis befinden. Beim Eintreten des parametrisierten Ereignisses wird dann folgender Suchpfad durchlaufen:

1. `/_N_CUS_DIR/` (Anwender-Zyklen)
2. `/_N_CMA_DIR/` (Hersteller-Zyklen)
3. `/_N_CST_DIR/` (Standard-Zyklen)

Das erste gefundene Programm mit dem in MD11620 \$MN_PROG_EVENT_NAME angegebenen Namen wird ausgeführt.

Hinweis

- Der angegebene Programmname wird syntaktisch wie bei einem Unterprogrammname geprüft, d. h. die ersten beiden Zeichen müssen Buchstaben oder Unterstriche sein (keine Ziffern). Präfix (`_N_`) und Suffix (`_SPF`) des Programmnamens werden, wenn nicht angegeben, automatisch ergänzt.
- Es greifen die Schutzmechanismen, wie sie allgemein für Zyklen aktivierbar sind (Schutzstufen für Schreiben, Lesen usw.).

Verhalten beim Starten eines Anwender-ASUP

Das Verhalten der Funktion "Ereignisgesteuerter Programmaufruf" beim Starten eines Anwender-ASUP aus dem Kanalzustand Reset kann kanalspezifisch eingestellt werden mit dem Maschinendatum:

MD20109 \$MC_PROG_EVENT_MASK_PROPERTIES.Bit 0 = <Wert>

Bit	Wert	Bedeutung
0	0	Bei Eintreten eines mit MD20108 parametrisierten Ereignisses wird das PROG_EVENT-Programms gestartet.
	1	Bei Eintreten eines mit MD20108 parametrisierten Ereignisses wird das PROG_EVENT-Programms nicht gestartet.

Verhalten bei gesetzter Einzelsatzbearbeitung

Das Verhalten der Funktion "Ereignisgesteuerter Programmaufruf" bei gesetzter Einzelsatzbearbeitung kann für jedes auslösende Ereignis kanalspezifisch mit folgendem Maschinendatum eingestellt werden:

MD20106 \$MC_PROG_EVENT_IGN_SINGLEBLOCK.Bit <n> = <Wert>

Bit	Wert	Bedeutung: Im PROG_EVENT-Programm wird die Einzelsatzbearbeitung bei Ereignis:
0	0	Teileprogramm-Start: wirksam
	1	Teileprogramm-Start: unterdrückt
1	0	Teileprogramm-Ende: wirksam
	1	Teileprogramm-Ende: unterdrückt
2	0	Kanal-Reset: wirksam
	1	Kanal-Reset: unterdrückt
3	0	Hochlauf: wirksam
	1	Hochlauf: unterdrückt

Ist die Einzelsatzbearbeitung unterdrückt, wird das PROG_EVENT-Programm ohne Unterbrechung abgearbeitet.

Hinweis

- MD20106 \$MC_PROG_EVENT_IGN_SINGLEBLOCK wirkt für alle Einzelsatzbearbeitungstypen.
- Die Einzelsatzbearbeitung im PROG_EVENT-Programm kann durch folgende Einstellung grundsätzlich ausgeschaltet werden:
MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK, Bit 0 = 1 (Einzelsatz-Stopp verhindern)
Die Einstellungen im MD20106 \$MC_PROG_EVENT_IGN_SINGLEBLOCK sind dann unwirksam.

Verhalten bei aktiver Einlesesperre

Das Verhalten der Funktion "Ereignisgesteuerter Programmaufruf" bei gesetzter Einlesesperre (DB21, ... DBX6.1 = 1) kann für jedes auslösende Ereignis kanalspezifisch mit folgendem Maschinendatum eingestellt werden:

MD20107 \$MC_PROG_EVENT_IGN_INHIBIT.Bit <n> = <Wert>

Bit	Wert	Bedeutung: Im PROG_EVENT-Programm wird die Einlesesperre bei Ereignis:
0	0	Teileprogramm-Start: wirksam Wird das PROG_EVENT-Programm mit Befehl <code>RET</code> beendet, wird dafür kein ausführbaren Satz erzeugt
	1	Teileprogramm-Start: unterdrückt Wird das PROG_EVENT-Programm mit Befehl <code>RET</code> beendet, wird dafür ein ausführbaren Satz, analog zu M17, erzeugt.
1	0	Teileprogramm-Ende: wirksam
	1	Teileprogramm-Ende: unterdrückt

Bit	Wert	Bedeutung: Im PROG_EVENT-Programm wird die Einlesesperre bei Ereignis:
2	0	Kanal-Reset: wirksam
	1	Kanal-Reset: unterdrückt
3	0	Hochlauf: wirksam
	1	Hochlauf: unterdrückt

Aktualisierung der Anzeige des Programm- und Kanalzustandes unterdrücken

Um ein Flackern der Anzeige des Programm- und Kanalzustandes an der Bedienoberfläche des HMI zu vermeiden, kann für die Ausführung des im Allgemeinen kurzen PROG_EVENT-Programms die Aktualisierung der Anzeige unterdrückt werden. In der Anzeige bleibt dann der Programm- und Kanalzustand vor Aktivierung des PROG_EVENT-Programms anstehen.

MD20192 \$MC_PROG_EVENT_IGN_PROG_STATE.Bit <n> = <Wert>

Bit	Wert	Bedeutung: Während der Abarbeitung des PROG_EVENT-Programms wird die Aktualisierung der Anzeige des Programm- und Kanalzustands bei Ereignis:
1	0	Teileprogramm-Ende: Nicht unterdrückt
	1	Teileprogramm-Ende: Unterdrückt
2	0	Kanal-Reset: Nicht unterdrückt
	1	Kanal-Reset: Unterdrückt
3	0	Hochlauf: Nicht unterdrückt
	1	Hochlauf: Unterdrückt

Hinweis

Die Systemvariablen \$AC_STAT und \$AC_PROG bleiben von dieser Funktion unbeeinflusst, d. h. im laufenden ereignisgesteuerten Anwenderprogramm werden \$AC_STAT auf "aktiv" und \$AC_PROG auf "läuft" gesetzt sein.

Die NC/PLC-Nahtstellensignale DB21, ... DBX35.0-7 ("Programmzustand ..." und "Kanalzustand ...") bleiben ebenfalls unbeeinflusst.

Verhalten bei DB21, ... DBX7.2 / 3 / 4 (NC-Stop ...)

Das Verhalten der Funktion "Ereignisgesteuerter Programmaufruf" bei "NC-Stop", "NC-Stop an Satzgrenze" und "NC-Stop Achsen plus Spindel" kann für die auslösenden Ereignisse Teileprogramm-Ende, Kanal-Reset und Hochlauf kanalspezifisch mit folgendem Maschinendatum eingestellt werden ::

MD20193 \$MC_PROG_EVENT_IGN_STOP.Bit <n> = <Wert>

Bit	Wert	Bedeutung: Das PROG_EVENT-Programm wird bei NC-Stop und Ereignis
1	0	Teileprogramm-Ende: angehalten/verhindert
	1	Teileprogramm-Ende: vollständig abgearbeitet
2	0	Kanal-Reset: angehalten/verhindert
	1	Kanal-Reset: vollständig abgearbeitet

Bit	Wert	Bedeutung: Das PROG_EVENT-Programm wird bei NC-Stop und Ereignis
3	0	Hochlauf: angehalten/verhindert
	1	Hochlauf: vollständig abgearbeitet

Anwendungsbeispiel

Ein vom Bediener durch Betätigen der Taste NC-Stop während Kanal-Reset oder Hochlauf verursachter Flankenwechsel des Nahtstellensignals DB21, ... DBX7.3 (NC-Stop) wird bei der Abarbeitung des PROG_EVENT-Programms ignoriert und so ein unerwünschtes Anhalten an der Maschine verhindert werden.

Hinweis

Eine Programmierung von `DELAYFSTON` / `DELAYFSTOF` im PROG_EVENT-Programm kann das mit MD20193 einstellbare Verhalten nicht ersetzen. NC-Stop kann vor Ausführung des Befehls `DELAYFSTON` noch eine Unterbrechung bewirken.

2.9.12.3 Programmierung

PROG_EVENT-Programm

Programmende

Soll das Anwenderprogramm durch Teileprogramm-Start aktiviert werden, muss folgendes beachtet werden:

- Das Anwenderprogramm muss mit `M17` bzw. `RET` beendet werden.
- Ein Rücksprung mittels `REPOS`-Befehl ist nicht zulässig.

Satzanzeige

Die Anzeige in der aktuellen Satzanzeige kann durch das `DISPLOF`-Attribut in der `PROC`-Anweisung unterdrückt werden.

Kommunikation zum PLC-Anwenderprogramm

Über Anwender-M-Funktionen, die im PROG_EVENT-Programm geschrieben werden, kann das PLC-Anwenderprogramm, z.B. über den Bearbeitungsstatus des PROG_EVENT-Programms, informiert werden.

Systemvariable

Abfrage des auslösenden Ereignisses

Das auslösenden Ereignis kann im PROG_EVENT-Programm mit folgender Systemvariablen abgefragt werden:

<Wert> = \$P_PROG_EVENT (Ereignisgesteuerter Programmaufruf aktiv)

Wert	Bedeutung: Aktivierung durch
1	Teileprogramm-Start
2	Teileprogramm-Ende
3	Kanal-Reset
4	Hochlauf
5	Nach Ausgabe des letzten Aktionssatzes nach Satzsuchlauf (siehe "Automatischer Start eines ASUP nach Satzsuchlauf (Seite 63)")

Abfrage des aktuellen Kanals

Der Kanal in dem das PROG_EVENT-Programm abgearbeitet wird, kann mit folgender Systemvariablen ermittelt werden:

<Wert> = \$P_CHANNO (Abfrage der aktuellen Kanalnummer)

Hinweis

Das PROG_EVENT-Programm wird in dem Kanal abgearbeitet, in dem das auslösende Ereignis aufgetreten ist.

Hochlauf ist ein Ereignis, das in allen Kanälen gleichzeitig auftritt.

2.9.12.4 Randbedingungen

Not-Halt / Alarm

Steht bei Kanal-Reset oder nach Hochlauf ein Not-Halt oder ein BAG / NC-spezifischer Alarm an, wird das PROG_EVENT-Programm erst nach der Not-Halt- bzw. Fehler-Quittierung in allen betroffenen Kanälen bearbeitet.

Hinweis

Das Ereignis "Hochlauf" tritt in allen Kanälen gleichzeitig auf.

2.9.12.5 Beispiele

Beispiel 1: Aufruf des PROG_EVENT-Programms bei allen Ereignissen

Parametrierung

MD20108 \$MC_PROG_EVENT_MASK = 'H0F'

Aufruf von _N_PROG_EVENT_SPF bei:

- Teileprogramm-Start
- Teileprogramm-Ende
- Kanal-Reset
- Hochlauf

Programmierung

Programmcode	Kommentar
PROC PROG_EVENT DISPLOF	
; Bearbeitung für Teileprogramm-Start	
IF (\$P_PROG_EVENT==1)	
MY_GUD_VAR=0	; GUD-Variable initialisieren.
RET	
ENDIF	
; Bearbeitung für Teileprogramm-Ende und Kanal-Reset	
IF (\$P_PROG_EVENT==2) OR	
(\$P_PROG_EVENT==3)	
DRFOF	; DRF-Verschiebungen ausschalten.
IF \$MC_CHAN_NAME=="CHAN1"	
CANCEL (2)	; Modale Synchronaktion 2 löschen.
ENDIF	
RET	
ENDIF	
; Bearbeitung für Hochlauf	
IF (\$P_PROG_EVENT==4)	
IF \$MC_CHAN_NAME=="CHAN1"	
IDS=1 EVERY \$A_INA[1]>5.0 DO	
\$A_OUT[1]=1	
ENDIF	
RET	
ENDIF	
RET	

Beispiel 2: Aufruf des PROG_EVENT-Programms bei Kanal-Reset**Parametrierung**

MD20108 \$MC_PROG_EVENT_MASK = 'H04'

Aufruf von _N_PROG_EVENT_SPF
bei:

- Bedientafel-Reset

Programmierung

Programmcode	Kommentar
PROC PROG_EVENT DISPLOF	
N10 DRFOF	; DRF-Verschiebungen ausschalten
N20 M17	

Beispiel 3: Initialisierung der Funktion**Parametrierung**

Maschinendatenbelegung, Ausschnitt aus der Inbetriebnahmedatei (_N_INITIAL_INI)

Programmcode	Kommentar
...	
CHANDATA(3)	; Initialisierung für Kanal 3
\$MC_PROG_EVENT_IGN_INHIBIT='H01F'	
\$MC_PROG_EVENT_MASK='H04'	; Ereignis: Kanal-Reset
...	

Bedeutung

Das PROG_EVENT-Programm _N_CMA_DIR / _N_PROG_EVENT_SPF wird bei Kanal-Reset im 3. Kanal gestartet und bis zum Programmende abgearbeitet, unabhängig ob die Einlesesperre ein- oder ausgeschaltet ist.

2.9.13 Beeinflussung von Stopp-Ereignissen durch Stop-Delay-Bereiche**2.9.13.1 Funktion****Stop-Delay-Bereich**

Die Reaktion auf ein Stopp-Ereignis kann durch einen bedingt unterbrechbaren Bereich im NC-Programm beeinflusst werden. Ein solcher Programmbereich wird als Stop-Delay-Bereich bezeichnet.

Innerhalb von Stop-Delay-Bereichen soll nicht angehalten und auch der Vorschub nicht verändert werden. Ein etwaiger Stopp soll erst wirken, nachdem der Programmabschnitt zu Ende bearbeitet worden ist.

Dadurch ergeben sich folgende Vorteile:

- Der Programmabschnitt wird ohne Geschwindigkeitseinbruch bearbeitet.
- Bricht der Anwender, nachdem gestoppt ist, das Programm mit Reset ab, so ist der abgebrochene Programmsatz nach dem geschützten Bereich. Dieser Programmsatz eignet sich dann als Suchziel für einen nachfolgenden Satzsuchlauf.
- Solange ein Stop-Delay-Bereich bearbeitet wird, werden folgende Hauptlaufachsen nicht gestoppt:
 - Kommandoachsen
 - Positionierachsen, die mit POSA verfahren

Anwendung

Beispiel: Bearbeitung eines Gewindes.

Definition

Die Definition eines Stop-Delay-Bereichs erfolgt im Teileprogramm mittels der vordefinierten Prozeduren DELAYFSTON und DELAYFSTOF (siehe "Programmierung (Seite 141)").

Stopp-Ereignisse

Übersicht der Ereignisse, die einen Stopp zur Folge haben:

Ereignis	Reaktion
DB21, ... DBX7.7 (Reset) oder DB11 DBX20.7 (BAG-Reset)	Immediate
NC-Programm: M30	Alarm 16954
Anwender-Interrupt zum Start eines ASUP: FC 9 oder DB10 DBB1 (Setzen der digitalen NC-Eingänge von PLC)	Delayed
DB21, ... DBX6.2 / DB31, ... DBX2.2 (Restweg löschen) kanal- / achsspezifisch	Immediate
DB21, ... DBX6.3 (Unterprogramm-Durchlaufzahl löschen)	Delayed
DB21, ... DBX6.4 (Programmebenen Abbruch)	Delayed
Stopp wegen Einzelsatz <ul style="list-style-type: none"> • Im Stop-Delay-Bereich: NC stoppt am Ende des 1. Satzes außerhalb des Stop-Delay-Bereichs. • Einzelsatz bereits vor Stop-Delay-Bereich aktiv: NC hält an jeder Satzgrenze, d. h. auch im Stop-Delay-Bereich: DB21, ... DBX7.2 (NC-Stop an der Satzgrenze) Der Stop-Delay-Bereich ist damit abgewählt! 	Delayed
DB11 DBX21.7 (Einzelsatz Typ A einschalten)	Delayed
DB11 DBX21.6 (Einzelsatz Typ B einschalten)	Delayed
DB21, ... DBX7.4 (NC-Stop) und DB11 DBX0.6 (BAG-Stop Achsen plus Spindel)	Immediate
DB21, ... DBX7.3 (NC-Stop an Satzgrenze) und DB11 DBX0.5 (BAG-Stop)	Delayed
NC-Programm: Stopp wegen leerem Überspeicher-Puffer	Alarm 16954
NC-Programm: programmierter oder impliziter STOPRE	Alarm 16954
NC-Programm: M0 oder M1	Alarm 16954

Ereignis	Reaktion
DB21, ... DBX7.2 (NC-Stop an Satzgrenze)	Delayed
Unterprogramm-Ende sollte den Stop-Delay-Bereich immer abwählen.	Systemfehler
NC-Programm: WAITM	Alarm 16954
NC-Programm: WAITE	Alarm 16954
NC-Programm: INIT mit Parameter "S"	Alarm 16954
NC-Programm: MMC (STRING, CHAR)	Alarm 16954
DB21, ... DBB26 (Satz ausblenden ein/umschalten)	Delayed
DB21, ... DBB26 (Satz ausblenden ausschalten)	Delayed
DB21, ... DBX0.6 (DryRun einschalten)	Delayed
DB21, ... DBX0.6 (DryRun ausschalten)	Delayed
DB21, ... DBX6.1 (Einlesesperre)	Delayed
Alarm: Alarmprojektierung STOPATENDBYALARM	Immediate
Alarm: Alarmprojektierung STOPBYALARM	Immediate
Intern: Stopp nach Alarm bei leerem IPO-Puffer	Immediate
Intern: Stopp nach Alarm bei leerem IPO-Puffer	Immediate
NC-Programm: Alarm 16954 bei LFON	Alarm 16954
NC-Programm: WAITMC	Alarm 16954
NC-Programm: NEWCONF	Alarm 16954
NC-Programm: NEWCONF	Alarm 16954
BTSS: PI "_N_SETUDT"	Delayed
Erweitertes Stillsetzen und Rückziehen	Delayed
DB31, ... DBX3.0 (Externe Nullpunktverschiebung übernehmen)	Delayed
BTSS: PI "_N_FINDST" STOPRUN	Alarm 16955
Alarmer mit Reaktion NOREADY	Immediate
DB21, ... DBX7.2 == 1 (NC-Stop an Satzgrenze)	Delayed
DB21, ... DBX7.3 == 1 (NC-Stop)	
DB21, ... DBX7.4 == 1 (NC-Stop Achsen plus Spindel)	
DB21, ... DBX6.1 == 1 (Einlesesperre)	Delayed

Ereignis	Reaktion
DB21, ... DBX2.0 == 1 (Einzelsatz)	Delayed
Reaktion <ul style="list-style-type: none"> • Immediate Stoppt sofort, auch im Stop-Delay-Bereich. Wird als "hartes Stopp-Ereignis" bezeichnet. • Delayed Stopp (auch ein kurzfristiger) erfolgt erst nach dem Stop-Delay-Bereich. Wird als "sanftes Stopp-Ereignis" bezeichnet. • Alarm 16954 Programm wird abgebrochen, da im Stop-Delay-Bereich unerlaubte Programmbefehle verwendet worden sind. • Alarm 16955 Programm wird fortgesetzt, im Stop-Delay-Bereich hat eine unerlaubte Aktion stattgefunden. • Alarm 16957 Der Programmbereich (Stop-Delay-Bereich), der durch DELAYFSTON und DELAYFSTOF geklammert ist, konnte nicht aktiviert werden. Damit wirkt jeder Stopp sofort und wird nicht verzögert! Dies geschieht immer dann, wenn man in einen Stop-Delay-Bereich hineinbremst, d. h. vor dem Stop-Delay-Bereich beginnt ein Bremsvorgang, der erst im Stop-Delay-Bereich endet. Betritt man den Stop-Delay-Bereich mit Override 0, so kann der Stop-Delay-Bereich ebenso nicht aktiviert werden (Beispiel: ein G4 vor dem Stop-Delay-Bereich erlaubt es dem Anwender, den Override auf 0 abzusenken. Der nächste Satz im Stop-Delay-Bereich beginnt dann mit Override 0 und die beschriebene Alarmsituation tritt auf.) <p>Hinweis MD11411 \$MN_ENABLE_ALARM_MASK (Aktivierung von Warnungen) Bit 7 schaltet diesen Alarm erst ein.</p>	

Hinweis

Es gibt NC-Ereignisse, die nur kurzfristig stoppen, um einen Umschaltvorgang durchzuführen und anschließend sofort wieder zu starten. Dazu zählt z. B. ein Interrupt, durch welchen das NC-Programm kurzfristig gestoppt wird, um das zugehörige ASUP zu starten. Auch diese Ereignisse sind im Stop-Delay-Bereich zulässig, werden aber auf dessen Ende verschoben und gelten demnach als "sanftes" Stopp-Ereignis.

Hinweis

Trifft ein "hartes" Stopp-Ereignis auf den "Stop-Delay-Bereich", so wird der "Stop-Delay-Bereich" komplett abgewählt! Das heißt, tritt in diesem Programmabschnitt ein weiterer beliebiger Stopp auf, so wird sofort angehalten. Erst eine Neuprogrammierung (erneutes DELAYFSTON) lässt einen neuen Stop-Delay-Bereich beginnen.

2.9.13.2 Parametrierung

Maschinendaten

Stopp-Verhalten bei G331/G332

Für das Gewindebohren ohne Ausgleichsfutter (G331, G332) ist das Stopp-Verhalten wie folgt einstellbar:

MD11550 \$MN_STOP_MODE_MASK

Bit	Wert	Bedeutung
0	0 (Default)	Bewirkt einen impliziten Stop-Delay-Bereich, wenn G331/G332 aktiv ist und zusätzlich eine Bahnbewegung oder eine Verweilzeit (G4) programmiert wurde.
	1	Ein Stopp während aktivem G331/G332 ist in folgenden Situationen möglich: <ul style="list-style-type: none"> • Bei Unterbrechung des Bahnsteuerbetriebs (G64). • Bei Programmierung einer Verweilzeit (G4) zwischen den G33x-Sätzen. Zur Definition eines Stop-Delay-Bereichs müssen die NC-Befehle DELAYFSTON und DELAYFSTOF benutzt werden.

2.9.13.3 Programmierung

Stop-Delay-Bereiche definieren (DELAYFSTON, DELAYFSTOF)

Zur Definition eines bedingt unterbrechbaren Bereichs im Teileprogramm (Stop-Delay-Bereich) dienen die vordefinierten Prozeduren DELAYFSTON und DELAYFSTOF.

Hinweis

In Synchronaktionen sind DELAYFSTON und DELAYFSTOF **nicht** zulässig!

Syntax

```

DELAYFSTON
...
DELAYFSTOF

```

Bedeutung

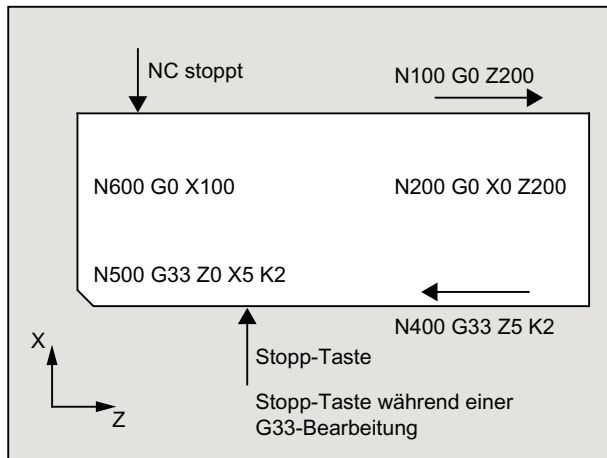
DELAYFSTON:	Beginn eines Stop-Delay-Bereichs definieren	
	Alleine im Satz:	ja
DELAYFSTOF:	Ende eines Stop-Delay-Bereichs definieren	
	Alleine im Satz:	ja

Programmierbeispiel

In einer Schleife wird folgender Programmblock wiederholt:

```
Programmcode
...
N99 MY_LOOP:
N100 G0 Z200
N200 G0 X0 Z200
N300 DELAYFSTON
N400 G33 Z5 K2 M3 S1000
N500 G33 Z0 X5 K3
N600 G0 X100
N700 DELAYFSTOF
N800 GOTOB MY_LOOP
...
```

Im folgenden Bild ist erkennbar, dass der Anwender im Stop-Delay-Bereich "Stopp" drückt, und die NC beginnt den Bremsvorgang außerhalb des Stop-Delay-Bereichs, d. h. im Satz N100. Damit kommt die NC im vorderen Bereich von N100 zum Halten.



Weitere Informationen

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:

Programmcode	Kommentar
N10010 DELAYFSTON	; Sätze mit N10xxx Programmebene 1.
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-Bereichsende in gleicher Ebene.
...	
N10060 R2 = R2 + 2	
N10070 G4 F1	; Stop-Delay-Bereich endet. Stopps wirken ab jetzt unmittelbar.

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	Stop-Delay-Bereich nicht aktiv
1	Stop-Delay-Bereich aktiv

2.9.13.4 Randbedingungen

Überlappung

Überschneiden sich zwei Stop-Delay-Bereiche, einer aus DELAYFSTON/DELAYFSTOF und der andere aus MD11550 \$MN_STOP_MODE_MASK, so wird der größtmögliche Stop-Delay-Bereich gebildet.

Verweilzeit (G4)

G4 ist im Stop-Delay-Bereich zulässig. Andere Teileprogrammbefehle, die zu einem zum vorübergehenden Stopp führen (z. B WAITM), sind nicht zulässig und lösen den Alarm 16954 aus.

Override

Wird der Override **vor** dem Stop-Delay-Bereich verändert, wird der Override **im** Stop-Delay-Bereich wirksam.

Wird der Override **im** Stop-Delay-Bereich verändert, wird die Änderung erst **nach** dem Stop-Delay-Bereich wirksam.

Hinweis

Override = 0

Wird der Override **vor** dem Stop-Delay-Bereich auf 0 gesetzt, kann der Stop-Delay-Bereich nicht aktiviert werden!

Vorschubsperr

DB21, ... DBX6.0 (Vorschubsperr) wirkt im Stop-Delay-Bereich nicht, erst nach dem Verlassen des Stop-Delay-Bereichs wird angehalten.

Einzelsatz

Schaltet man den Einzelsatz im Stop-Delay-Bereich ein, so stoppt der NC am Ende des ersten Satzes außerhalb des Stop-Delay-Bereichs.

Ist der Einzelsatz bereits vor dem Stop-Delay-Bereich angewählt, hält der NC an jeder Satzgrenze, d. h. auch im Stop-Delay-Bereich! **Der Stop-Delay-Bereich ist damit abgewählt.**

2.9.14 Größenanpassung des Interpolationspuffers

MD28060

Der kanalspezifische Interpolator arbeitet beim Teileprogrammablauf vorbereitete Sätze aus dem Interpolationspuffer ab. Die maximale Anzahl Sätze, die zu einem Zeitpunkt im Interpolationspuffer Platz finden sollen, wird durch das speicherkonfigurierende MD28060 \$MM_IPO_BUFFER_SIZE (Anzahl der NC-Sätze im IPO-Puffer(DRAM)) festgelegt. Für einige Anwendungen kann es sinnvoll sein, diesen Puffer nicht voll auszunutzen, um den "Abstand" zwischen Vorbereitung und Interpolation klein zu halten.

SD42990

Mit dem Settingdatum SD42990 \$SC_MAX_BLOCKS_IN_IPOBUFFER (Max. Anzahl Sätze im IPO-Puffer) kann dynamisch die Zahl der Sätze im Interpolationspuffer auf einen **kleineren** Wert als in MD28060 \$MC_MM_IPO_BUFFER_SIZE (Anzahl der NC-Sätze im IPO-Puffer (DRAM)) begrenzt werden, minimal auf 2 Sätze.

Werte des Settingdatums SD42990 \$SC_MAX_BLOCKS_IN_IPOBUFFER:

Wert	Wirkung
< 0	Keine Begrenzung des Interpolationspuffers aktiv. Der max. mögliche IPO-Puffer entsprechend MD 28060: MM_IPO_BUFFER_SIZE wird aktiviert.
oder 1	Der minimal zulässige IPO-Puffer mit 2 Sätzen wird aktiv.

Wert	Wirkung
< MM_IPO_BUFFER_SIZE	Der IPO-Puffer wird maximal mit der angegebenen Anzahl Sätze aktiviert.
>= MM_IPO_BUFFER_SIZE	Der IPO-Puffer wird mit der in MD 28060: MM_IPO_BUFFER_SIZE angegebenen Anzahl Sätze aktiviert.

Hinweis

Wenn das SD42990 \$SC_MAX_BLOCKS_IN_IPOBUFFER im Teileprogramm gesetzt wird, wird die Begrenzung des Interpolationsbuffers sofort wirksam, wenn der Satz mit dem SD vom Interpreter in der Vorbereitung abgearbeitet wird.

D.h. gegebenenfalls wird die Begrenzung des IPO-Puffers einige Sätze vor der beabsichtigten Begrenzung wirksam (siehe auch das MD28070 \$MC_MM_NUM_BLOCKS_IN_PREP).

Will man das vermeiden, und die Begrenzung des IPO-Puffers soll satzsynchron wirksam werden, muss man vor dem Setzen des SD im Teileprogramm ein STOPRE (Vorlaufstopp) programmieren.

Gültigkeit

Das SD42990 \$SC_MAX_BLOCK_IN_IPOBUFFER hat globale, kanalspezifische Gültigkeit und kann auch in einem Teileprogramm verändert werden. Dieser veränderte Wert wird bei Programmende beibehalten. Soll dieses Settingdatum bei definierten Ereignissen wieder zurückgesetzt werden, muss dafür ein so genanntes ereignisgesteuertes Programm eingerichtet werden. Man könnte z. B. dieses Settingdatum bei RESET immer auf einen vordefinierten Wert setzen.

Anwendung

Die IPO-Pufferbegrenzung kann eingesetzt werden, wenn die Anzahl Sätze zwischen Satzaufbereitung und Interpolation möglichst klein sein soll, z. B. wenn im Teileprogramm Istpositionen gelesen und weiterverarbeitet werden sollen.

Beispiel

```

N10 ...
N20 ...
.....
N100 $SC_MAX_BLOCKS_IN_IPOBUFFER = 5           ; Begrenzung des IPO-Buffers auf 5
                                                NC-Sätze
N110 ...
N120 ...
.....
N200 $SC_MAX_BLOCKS_IN_IPOBUFFER = -1         ; Aufhebung der Begrenzung des IPO-
                                                Puffers
N210 ...
.....

```

2.9.15 Basis-Satzanzeige (nur bei ShopMill/ShopTurn)

2.9.15.1 Funktion

Zusätzlich zur bestehenden Satzanzeige können bei ShopMill/ShopTurn über die so genannte Basis-Satzanzeige alle Sätze angezeigt werden, die eine **Aktion an der Maschine** bewirken werden.

Die tatsächlich angefahrenen Endpositionen werden als Absolutposition dargestellt. Die Positionswerte beziehen sich wahlweise auf das Werkstückkoordinatensystem (WKS) oder auf das Einstellbare Nullpunkt-System (ENS).

Die Anzahl der vorausschauend im Anzeigebuffer abgelegten Anzeigesätze hängt von der Anzahl der im jeweiligen Bearbeitungszustand aufbereiteten Sätze im Vorlauf der Steuerung ab. Wird ein Vorlaufstopp bearbeitet, so geht die Anzahl der Anzeigesätze gegen null und steigt nach der Quittierung des Vorlaufstopps wieder an. Bei REORG-Ereignissen (z. B. Betriebsartenwechsel, ASUP-Start) werden die vorausschauend angelegten Anzeigesätze gelöscht und danach wieder aufbereitet.

In der Basis-Satzanzeige aufbereitete Werte stimmen überein mit:

- Angewählten Werkzeugen
 - Vorschub und Spindeldrehzahl
 - Tatsächlich angefahrenen Positionswerten
- Ausnahmen:
Bei aktiver Werkzeugradiuskorrektur können Abweichungen auftreten.
Bei Modulo-Achsen wird in der Basis-Satzanzeige der programmierte Wert angezeigt, der auch außerhalb des Modulo-Bereichs liegen kann.

Basissatzanzeige ein-/ausschalten

Die Basis-Satzanzeige ist über das folgende Settingdatum ein-/ausschaltbar:

SD42750 \$SC_ABSBLOCK_ENABLE (Basis-Satzanzeige freigeben)

2.9.15.2 Parametrierung

Basis-Satzanzeige konfigurieren

Die Basis-Satzanzeige ist über folgende Maschinendaten konfigurierbar:

NC Maschinendaten für Basis-Satzanzeige	Bedeutung:
MD28400 \$MC_MM_ABSBLOCK	Basis-Satzanzeige aktivieren
MD28402 \$MC_MM_ABSBLOCK_BUFFER_CONF[2]	Größe des Anzeigebuffers
Anzeige Maschinendaten	einzustellende Positionswerte:
MD9004 \$MM_DISPLAY_RESOLUTION	für metrische Maßangabe
MD9011 \$MM_DISPLAY_RESOLUTION_INCH	für Inch Maßangabe

MD9010 \$MM_SPIND_DISPLAY_RESOLUTION	für Anzeigefeinheit Spindeln einzustellendes Koordinatensystem
MD9424 \$MM_MA_COORDINATE_SYSTEM	für Istwertanzeige WKS oder ENS

Diese Anzeigemaschinendaten werden in die NC-Maschinendaten MD17200 \$MN_GMMC_INFO_UNIT[0] (Globale HMI-Info) bis MD17200 \$MN_GMMC_INFO_UNIT[3] kopiert. Damit kann vom NC aus auf diese Anzeigemaschinendaten zugegriffen werden.

Aktivierung

Die Basis-Satzanzeige wird aktiviert durch das MD28400 \$MC_MM_ABSBLOCK (Satzanzeige mit Absolutwerten aktivieren) mittels Power On. Wird das MD28400 \$MC_MM_ABSBLOCK mit 1 beschrieben, so wird im Hochlauf ein kanalspezifischer Anzeigepuffer (FIFO) eingerichtet.

Größe des Anzeigepuffer (FIFO) = (MD28060 \$MC_MM_IPO_BUFFER_SIZE (Anzahl der NC-Sätze im IPO-Puffer) + MD28070 \$MC_MM_NUM_BLOCKS_IN_PREP (Anzahl der Sätze für die Satzaufbereitung)) multipliziert mit 128 Byte. Bei Standard-Maschineneinstellung entspricht dies einer Größe von 6 KByte.

Größe des Anzeigepuffers optimieren:

Der Speicherbedarf kann durch Angabe eines Wertes zwischen 128 und 512 optimiert werden. Die im Anzeigepuffer aufbereiteten Anzeigesätze werden über einen konfigurierbaren Upload-Puffer zum HMI transportiert.

Die maximale Größe des Upload-Puffers ergibt sich durch Multiplikation von (MD28402 \$MC_MM_ABSBLOCK_BUFFER_CONF[0] + MD28402 \$MC_MM_ABSBLOCK_BUFFER_CONF[1] + 1) mit der durch MD28400 \$MC_MM_ABSBLOCK projektierten Satzlänge.

Konfiguriert werden hier die Anzahl der Sätze **vor** dem aktuellen Satz im MD28402 \$MC_MM_ABSBLOCK_BUFFER_CONF[0] und **nach** dem aktuellen Satz im MD28402 \$MC_MM_ABSBLOCK_BUFFER_CONF[1].

Randbedingungen

Bei Überschreiten der in MD28400 \$MC_MM_ABSBLOCK projektierten Länge eines Anzeigesatzes wird dieser Anzeigesatz entsprechend abgeschnitten. Um dies darzustellen wird am Satzende der String "..." angefügt.

Für vorübersetzte Zyklen

(MD10700 \$MN_PREPROCESSING_LEVEL > 1 (Programmvorverarbeitungsstufe)) enthält der Anzeigesatz **nur** Achspositionen.

Weitere Randbedingungen für die Basis-Satzanzeige:

- Modale Synchronaktionssätze mit Absolutwerten werden nicht berücksichtigt.
- Die Basis-Satzanzeige ist während Satzsuchlauf mit oder ohne Berechnung deaktiviert.
- Polarkoordinatenprogrammierung wird nicht kartesisch dargestellt.

Radius / Durchmesser-Werte

Auf der Basis-Satzanzeige und der Positionsanzeige dargestellte Durchmesserwerte können zur internen Berechnung als Radius benötigt werden. Es bestehen folgende Möglichkeiten diese Werte für Maßangaben in Radius/Durchmesser gemäß G-Gruppe 29 zu beeinflussen:

- G-Befehl DIAMCYCOF (Erweiterung kanalspezifische Durchmesserprogrammierung)
Dieser G-Befehl schaltet die kanalspezifische Durchmesserprogrammierung während der Zyklusbearbeitung aus. Im Zyklus können somit Berechnungen immer im Radius erfolgen. Die Positionsanzeige und die Basis-Satzanzeige erfolgt weiterhin entsprechend dem Zustand der Durchmesserprogrammierung vor DIAMCYCOF.
In der Basis-Satzanzeige bleibt der zuletzt angezeigte Wert erhalten.
- G-Befehl DIACYCOFA[AX] (achsspezifische Durchmesserprogrammierung)
Dieser G-Befehl schaltet die achsspezifische Durchmesserprogrammierung während der Zyklusbearbeitung aus. Im Zyklus können somit Berechnungen immer im Radius erfolgen. In der Positionsanzeige und in der Basis-Satzanzeige erfolgt weiterhin entsprechend dem Zustand bezüglich Durchmesserprogrammierung vor DIACYCOFA[AX].
In der Basis-Satzanzeige bleibt der zuletzt angezeigte Wert erhalten.
- MD27100 \$MC_ABSBLOCK_FUNCTION_MASK (Satzanzeige mit Absolutwerten parametrieren)

Bit0 = 1 Sollwerte der Planachse werden in der Basis-Satzanzeige grundsätzlich als Durchmesserwerte angezeigt.

Verhalten bei aktivem Kompressor

Bei aktivem Kompressor mit G-Gruppe 30 ungleich COMPOF werden zwei Anzeigesätze generiert. Der

- Erste enthält den G-Befehl des aktiven Kompressors.
- Zweite enthält den String "..." als Zeichen dafür, dass Anzeigesätze fehlen.

Beispiel:

```
G0 X10 Y10 Z10                      ; Satz der noch für die Basis-Satzanzeige aufbereitet wird  
COMPCAD                              ; Kompressor für optimierte Oberflächengüte (CAD-Prog.) ein  
...                                    ; String als Zeichen dafür, das Anzeigesätze fehlen
```

Um Engpässe der NC-Leistung zu vermeiden, wird die Basis-Satzanzeige automatisch abgeschaltet. Als Zeichen dafür, dass Anzeigesätze fehlen, wird ein Anzeigesatz mit dem String "..." generiert.

Im Einzelsatz werden immer alle Anzeigesätze erzeugt.

2.9.15.3 Aufbau für einen DIN-Satz

Aufbau des Anzeigesatzes für einen DIN-Satz

Prinzipieller Aufbau des Anzeigesatzes für einen DIN-Satz

- Satznummer/Label
- G-Befehl der ersten G-Gruppe
(nur bei Änderung gegenüber dem letzten Maschinenfunktionssatz).
- Achspositionen
(Reihenfolge entsprechend MD20070 \$MC_AXCONF_MACHAX_USED
(Maschinenachsnummer gültig im Kanal)).
- Weitere modale G-Befehle
(nur bei Änderungen gegenüber den letzten Maschinenfunktionssatz).
- Weitere Adressen wie programmiert.

Der Anzeigesatz für die Basis-Satzanzeige wird direkt aus dem programmierten Teileprogrammssätzen nach folgenden Regeln abgeleitet:

- Makros werden expandiert.
- Ausblendkennungen und Kommentare entfallen.
- Satznummer und Labels werden aus dem Originalsatz übernommen, entfallen jedoch wenn DISPLOF aktiv ist.
- Die Anzahl der Nachkommastellen wird durch die Anzeigemaschinendaten MD9004, MD9010 und MD9011 über HMI festgelegt.

HMI-Anzeigemaschinendatum	Zugriff im NC Maschinendatum
MD9004 \$MM_DISPLAY_RESOLUTION	MD17200 \$MN_GMMC_INFO_NO_UNIT[0]
MD9011 \$MM_DISPLAY_RESOLUTION_INCH	MD17200 \$MN_GMMC_INFO_NO_UNIT[1]
MD9010 \$MM_SPIND_DISPLAY_RESOLUTION	MD17200 \$MN_GMMC_INFO_NO_UNIT[2]
MD9424 \$MM_MA_COORDINATE_SYSTEM	MD17200 \$MN_GMMC_INFO_NO_UNIT[3]

- Programmierte Achs-Positionen werden als absolute Positionen in dem durch das MD9424 \$MM_MA_COORDINATE_SYSTEM (Koordinatensystem für Istwertanzeige) vorgegebenen Koordinatensystem (WKS / ENS) dargestellt.

Hinweis

Bei Modulo-Achsen entfällt die Modulo-Korrektur. Dadurch sind Positionen außerhalb des Modulo-Bereichs möglich und es ergibt sich zwangsläufig eine Differenz zur Positionsanzeige in der grundsätzlich modulo-gewandelt wird.

Beispiele

Gegenüberstellungen von Anzeigesatz (Originalsatz) zur Basis-Satzanzeige:

- **Programmierte Positionen** werden absolut dargestellt.
Die Adressen AP/RP werden mit ihren programmierten Werten dargestellt.

Originalsatz:	Anzeigesatz:
N10 G90 X10.123	N10 X10.123
N20 G91 X1	N20 X11.123

- **Zuweisungen von Adressen** (nicht DIN-Adressen) werden in der Form <adresse> = <konstante> dargestellt.

Originalsatz:	Anzeigesatz:
N110 R1 = -67.5 R2 = 7.5	
N130 Z = R1 RND = R2	N130 Z-67.5 RND = 7.5

- **Indizes von Adressen** (Adresserweiterungen) werden als Konstanten <adresse> [<konstante>] = <konstante> dargestellt.

Originalsatz:	Anzeigesatz:
N220 DEF AXIS AXIS_VAR = X	
N240 FA[AXIS_VAR] = R2	N240 FA[X] = 1000

- **DIN-Adressen ohne Adresserweiterung** werden in der Form <din_adresse> <konstante> dargestellt.

Originalsatz:	Anzeigesatz:
N410 DEF REAL FEED = 1.5	
N420 F = FEED	N420 F1.5

Für **H-Funktionen** gilt: Unabhängig von der Ausgabeart zur PLC (MD22110 \$MC_AUXFU_H_TYPE_INT (Typ von H-Hilfsfunktionen ist integer)) wird der jeweils programmierte Wert angezeigt.

- Für die **Werkzeug-Anwahl über T-Befehl** wird eine Anzeigeeinformation der Form T<wert> bzw. T=<string> generiert. Wurde eine Adresserweiterung programmiert, so wird diese auch aufgelöst. Wurden mehrere Spindeln projektiert oder ist die Funktion "Werkzeugwechsel über Werkzeughalter" (MD20124 \$MC_TOOL_MANAGEMENT_TOOLHOLDER (Werkzeughalternummer)) aktiv, so wird die T-Nummer grundsätzlich mit Adresserweiterung ausgegeben. Wurde keine Adresserweiterung programmiert, wird die Nummer der Masterspindel bzw. des Mastertoolholders eingesetzt (T<spindel_nummer/tool_holder> =).
- Für die **Spindelprogrammierung** über S, M3, M4, M5, M19, M40 - M45 und M70 (bzw. MD 20094 \$MC_SPIND_RIGID_TAPPING_M_NR (M-Funktion für das Umschalten in den gesteuerten Achsbetrieben)) gilt bezüglich Adresserweiterung folgende Regelung: Wurde eine Adresserweiterung programmiert, so wird diese auch aufgelöst. Wurden mehrere Spindeln projektiert, so wird grundsätzlich die Adresserweiterung mit ausgegeben. Wurde keine Adresserweiterung programmiert, wird die Nummer der Masterspindel eingesetzt (S<spindel_nummer>=).
- **Indirekte G-Befehlprogrammierung** der Form G[<gruppe>] = <ausdruck> wird durch den entsprechenden G-Befehl ersetzt.

Originalsatz:	Anzeigesatz:
N510 R1=2	
N520 G[8]= R1	N520 G54

- **Modale G-Befehle**, die keinen ausführbaren Satz erzeugen werden aufgesammelt und mit dem Anzeigesatz des nächsten ausführbaren Satz angezeigt, sofern dies von der Syntax her erlaubt ist (DIN-Satz). Ist dies nicht der Fall (z. B. vordefinierter Unterprogrammaufruf TRANSMIT) so wird ein eigener Anzeigesatz mit den geänderten G-Befehlen vorangestellt.

Originalsatz:	Anzeigesatz:
N610 G64	G64
N620 TRANSMIT	N620 TRANSMIT

- Für **Teileprogrammzeilen** in denen die Adressen **F** und **FA** vorkommen, wird immer ein Anzeigesatz generiert (auch bei MD22240 \$MC_AUXFU_F_SYNC_TYPE = 3 (Ausgabezeitpunkt der F-Funktionen)).

Originalsatz:	Anzeigesatz:
N630 F1000	N630 F1000
N640 X100	N640 X100

- Die für die **Satzanzeige generierten Anzeigesätze** werden **direkt** von den programmierten Teileprogrammzeilen abgeleitet. Werden durch die Konturaufbereitung Zwischensätze erzeugt (z.B. Werkzeugradiuskorrektur G41/G42, Radius/Fase RNDM, RND, CHF, CHR), so erhalten diese Sätze die Anzeigeeinformation des der Bewegung zugrunde liegenden Teileprogrammsatzes.

2.10 Asynchrone Unterprogramme (ASUPs)

Originalsatz:	Anzeigesatz:
N710 Y157.5 G42	N710 Y157.5 G42
N720 Z-67.5 RND=7.5	N720 Z-67.5 RND=7.5

- Beim **EXECTAB-Befehl** (Abarbeiten einer Tabelle von Konturelementen) wird im Anzeigesatz der durch EXECTAB generierte Satz angezeigt.

Originalsatz:	Anzeigesatz:
N810 EXECTAB (KTAB[5])	N810 G01 X46.147 Z-25.38

- Beim **EXECSTRING-Befehl** wird im Anzeigesatz der durch EXECSTRING generierte Satz angezeigt.

Originalsatz:
N910 DEF STRING[40] PROGSTRING = "N905 M3 S1000 G94 Z100 F1000 G55"
N920 EXECSTRING (PROGSTRING)

Originalsatz:
N905 Z100 G55 G94 M3 S1000 F1000

2.10 Asynchrone Unterprogramme (ASUPs)

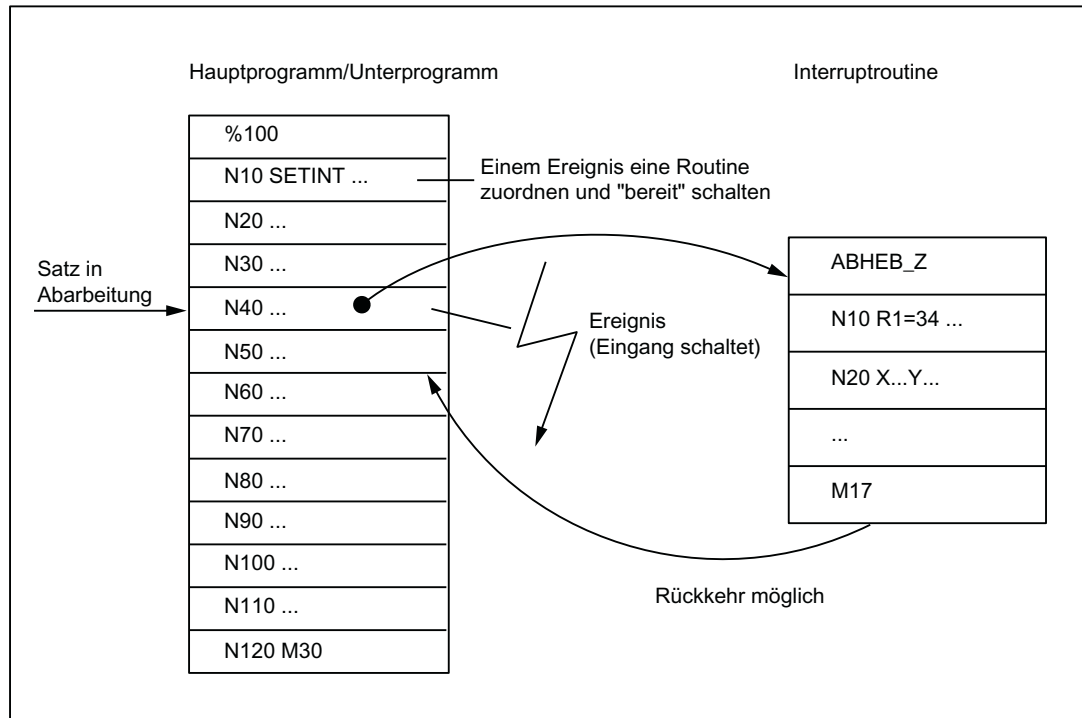
2.10.1 Funktion

Hinweis

Die im Folgenden verwendeten Begriffe "Asynchrones Unterprogramm", "ASUP" und "Interruptroutine" bezeichnen die gleiche Funktionalität.

Allgemein

Asynchrone Unterprogramme (ASUP) sind NC-Programme, die in einem NC-Kanal als Reaktion auf asynchrone Ereignisse (Interrupt-Eingangssignale, Prozess- bzw. Maschinenzustände) gestartet werden. Durch die Aktivierung eines ASUP wird ein in Abarbeitung befindliches NC-Programm unterbrochen. Mit dem ASUP-Ende kann das NC-Programm an der Unterbrechungsposition fortgesetzt werden.



Das im Kanal in Abarbeitung befindliche NC-Programm kann im Ganzen oder abschnittsweise vor einer Unterbrechung durch ein ASUP geschützt werden. Siehe Kapitel "Programmierung (SETINT, PRIO) (Seite 163)", Absatz "Flexible Programmierung".

Definition

Damit aus einem NC-Programm ein ASUP (Interruptroutine) wird, muss dem NC-Programm über den Befehl `SETINT` (siehe Kapitel "Programmierung (SETINT, PRIO) (Seite 163)") oder über den PI-Dienst "ASUP" (siehe Funktionshandbuch *PLC und Grundfunktionen* in den *Bausteinbeschreibungen*) ein Interruptsignal zugeordnet werden.

Interruptsignale

- Als Interruptsignale stehen insgesamt 8 Peripherie-Eingänge zur Verfügung.
- Die Peripherie-Eingangssignale können über das PLC-Anwenderprogramm beeinflusst werden.
- Die ersten vier Peripherie-Eingänge sind die 4 schnellen Eingänge der NCU-Baugruppe. Die Signalzustände können über die NC/PLC-Nahtstelle im Datenbaustein DB10 gelesen werden. Über die NC/PLC-Nahtstelle im Datenbaustein DB10 können die Eingangssignale auch gesperrt werden.

Weitere Informationen

- Weitere Informationen finden Sie unter A4: Digitale und analoge NC-Peripherie (Seite 723).
- Weitere Informationen zur PLC-Beeinflussung der schnellen NC-Eingänge (Interruptsignale) finden Sie im Funktionshandbuch PLC.

Aufruf

Im Programmbetrieb

Im Programmbetrieb, d. h. in der Betriebsart AUTOMATIK oder MDA, ist der Aufruf eines ASUP grundsätzlich immer möglich.

Außerhalb vom Programmbetrieb

Außerhalb des Programmbetriebs ist der Aufruf eines ASUP in folgenden Betriebsarten, Maschinenfunktionen und Zuständen möglich:

- JOG, JOG-REF
- MDA-Teach In, MDA-Teach In-REF, MDA-Teach In-JOG, MDA-REF, MDA-JOG
- AUTOMATIK, Programmzustand "gestoppt", "ready"
- Achszustand "Nicht referenziert"

Wird ein ASUP während JOG oder JOG-REF gestartet, wird die aktuelle Verfahrbewegung abgebrochen.

Aktivierung

Die Aktivierung eines ASUP erfolgt über:

- 0/1-Flanke des Interruptsignals, hervorgerufen durch eine 0/1-Flanke am zugehörigen schnellen NC-Eingang
- Aufruf des "Function Call ASUP" (siehe auch Funktionshandbuch "PLC und Grundprogramm", Kapitel "PLC-Grundprogramm").
- Setzen eines Ausgangs über Synchronaktion, der über Kurzschluss auf einen Interrupt-Eingang parametrisiert ist (siehe "Beispiele (Seite 166)")

Weitere Informationen

Funktionshandbuch Synchronaktionen

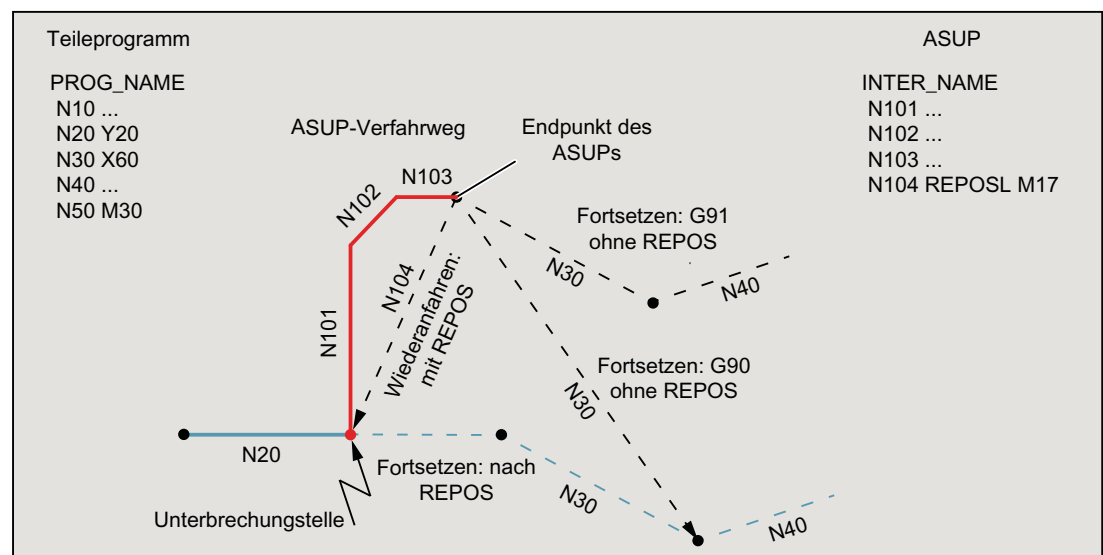
Anzeige

Die Aktivierung eines ASUP wird kanalspezifisch mit dem folgenden NC/PLC-Nahtstellensignal angezeigt:

DB21, ... DBX378.0 == 1 (ASUP aktiv)

2.10.1.1 Ablauf eines ASUPs im Programmbetrieb

1. **Abbremsen der Achsen**
Nach Aktivierung des ASUPs werden alle Maschinenachsen anhand der Bremsrampe (MD32300 \$MA_MAX_AX_ACCEL) bis zum Stillstand abgebremst und die Achspositionen abgespeichert.
2. **Reorganisation**
Zusätzlich zum Abbremsen der Achsen werden die vordekodierten Rechensätze bis zum Unterbrechungssatz zurückgerechnet, d. h. alle Variablen, Frames und G-Befehle bekommen den Wert zugeordnet, den sie ohne Vorausdekodierung des Teileprogramms an der Unterbrechungsstelle besitzen würden. Diese Werte werden auch wieder zwischengespeichert, um nach dem Ende des ASUPs wieder darauf zugreifen zu können. Ausnahmen, bei denen keine Reorganisation möglich ist:
 - Innerhalb von Gewindeschneidsätzen
 - Bei komplexen Geometrien (z. B. Spline oder Radiuskorrektur)
3. **Abarbeiten des ASUPs**
Nach Beendigung der Reorganisation wird automatisch das ASUP gestartet. Das ASUP wird wie ein normales Unterprogramm abgearbeitet (Schachtelungstiefe etc.).
4. **Ende des ASUPs**
Nachdem die Endkennung (M02, M30, M17) des ASUPs bearbeitet wurde, wird standardmäßig auf die Endposition des auf den Unterbrechungssatz folgenden Teileprogramms gefahren.
Falls ein Rückpositionieren auf den Unterbrechungspunkt erfolgen soll, muss am Ende des ASUPs eine REPOS-Anweisung programmiert werden, z. B.: N104 REPOS L M17



2.10.1.2 ASUP mit REPOSA

Wird ein NC-Programm durch NC-Stop oder Alarm angehalten und anschließend vom PLC-Anwenderprogramm über FC9 ein ASUP mit REPOSA ausgelöst, ergibt sich typischerweise folgender Ablauf:

- Das ASUP bzw. die darin programmierten Verfahrbewegungen werden abgearbeitet:
 - Programmzustand: "Angehalten"
 - DB21, ... DBX318.0 (ASUP ist angehalten) = 1
- Vor dem Wiederanfahren an die Kontur (REPOS) wird erneut angehalten.
- Zum Wiederanfahren an die Kontur (REPOS) löst der Bediener NC-Start aus:
 - DB21, ... DBX318.0 (ASUP ist angehalten) = 0
 - Die Wiederanfahrbewegung wird ausgeführt.
- Mit dem Ende der Wiederanfahrbewegung wird das FC9-Quittungssignal "ASUP Done" gesetzt und das unterbrochene NC-Programm fortgesetzt.

Hinweis

Das NC/PLC-Nahtstellensignal DB21, ... DBX318.0 (ASUP ist angehalten) wird nur im folgenden Fall gesetzt:

Interrupt im Programmbetrieb und Kanalzustand "Unterbrochen".

Hinweis

Bei ASUPs ohne REPOSA fallen das FC9-Quittungssignal "ASUP Done" und das Rücksetzen des NC/PLC-Nahtstellensignals DB21, ... DBX318.0 = 0 (ASUP ist angehalten) zeitlich zusammen.

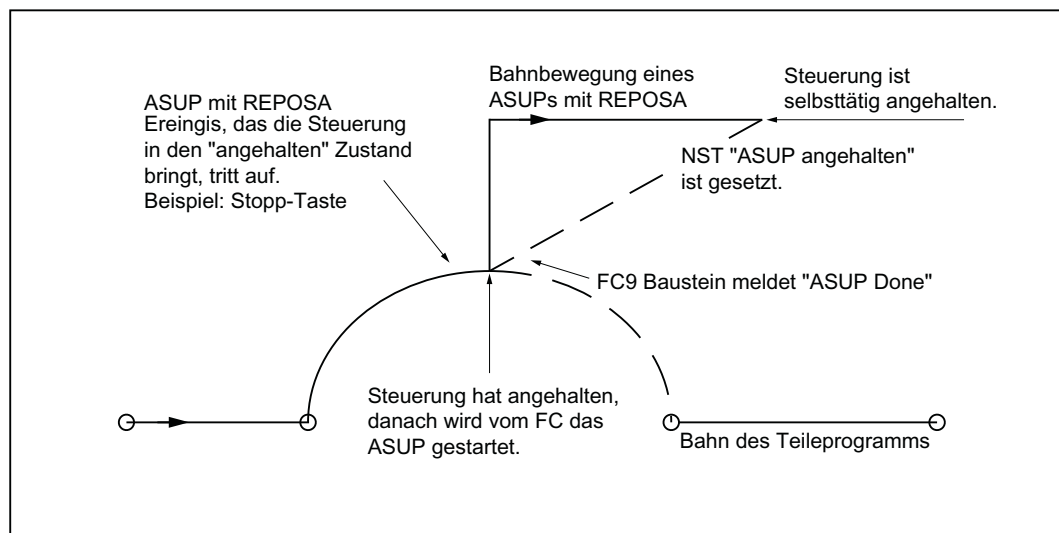


Bild 2-7 Schematischer Ablauf: ASUP mit REPOSA

2.10.1.3 NC-Verhalten

Die unterschiedlichen Reaktionen in den verschiedenen Zuständen von Kanal, BAG oder NC auf ein aktiviertes ASUP sind in der folgenden Tabelle aufgeführt:

Zustand	ASUP-Start	Reaktion der Steuerung
Programm ist aktiv	Interrupt, (PLC)	<ol style="list-style-type: none"> 1. Schnellabheben oder Achsen stoppen 2. Unterbrechung des Programms für die Dauer des ASUP 3. Anfahren der Unterbrechungsstelle, wenn REPOS im ASUP 4. Fortsetzung des Teileprogramms
Reset	Interrupt, (PLC)	<p>Das ASUP läuft ab wie ein Hauptprogramm. Am ASUP-Ende wird wieder Reset (ohne M30) ausgeführt. Der nächste Zustand der Steuerung hängt von folgenden Maschinendaten ab:</p> <p>MD20110 \$MC_RESET_MODE_MASK MD20112 \$MC_START_MODE_MASK</p> <p>Weitere Informationen Werkstücknahes Istwertsystem (Seite 399)</p>
Programmbetrieb (AUTO-MATIK oder MDA) + Kanal gestoppt	Interrupt, (PLC)	<p>ASUP läuft ab. An seinem Ende wird wieder der Stopp-Zustand eingenommen.</p> <p>Wenn REPOS im ASUP:</p> <ul style="list-style-type: none"> • Die ASUP-Abarbeitung wird vor dem Anfahrtsatz gestoppt. • Das Anfahren kann durch Start-Taste ausgelöst werden.
	Start-Taste	Nach ASUP-Ablauf läuft auch das gestoppte Programm weiter ab.
Handbetrieb + Kanal gestoppt	Interrupt, (PLC)	Steuerung nimmt intern für den angesprochenen Kanal den Zustand "interne Programmbearbeitungsbetriebsart" ein (nach außen nicht merkbar) und aktiviert dann das ASUP. Die angewählte Betriebsart bleibt erhalten. Nach ASUP-Ende (M17) wird der ursprüngliche Zustand wieder eingenommen.
JOG AUTO-Teach-In AUTO-Teach-Referenzp.	Interrupt, (PLC)	Stopp der Bearbeitung, Auswerten von: MD11602 \$MN_ASUP_START_MASK MD11604 \$MN_ASUP_START_PRIO_LEVEL
MDA-JOG, MDA-Teach-In, MDA-Teach-Referenzp.	Interrupt, (PLC)	Ggf. internes Umschalten in "interne Programmbearbeitungsbetriebsart", Aktivierung des ASUP, Zustand vor ASUP-Start wieder herstellen. Ein zusammen mit SETINT definiertes LIFTFAST wird bei JOG nicht aktiviert.
Handbetrieb + Kanal läuft	Interrupt, (PLC)	Die gerade aktive Bewegung wird gestoppt. Der Restweg wird gelöscht. Der weitere Ablauf entspricht "Handbetrieb, Kanal gestoppt".
Anwenderalarm 65500 - 65999, Kanal in Reset	Interrupt, (PLC)	Wenn folgendes Maschinendatum eingestellt ist, wird das Anwender-ASUP trotz der Alarmreaktion "Verriegelung NC-Start" abgearbeitet: MD20194 \$MC_IGNORE_NONCSTART_ASUP

2.10 Asynchrone Unterprogramme (ASUPs)

Zustand	ASUP-Start	Reaktion der Steuerung
Abarbeitung von INITIAL.INI Satzsuchlauf	nicht möglich	Es wird das Signal "Interruptanforderung nicht möglich" erzeugt.
Alarm, der nicht durch NC-Start behoben werden kann		
Digitalisieren eingeschaltet		
Kanal im Fehlerzustand		
Anderer Anwenderalarm als 65500 - 65999, Kanal in Reset, MD20194 nicht aktiv	nicht möglich	Es wird das Signal "Die Anforderung wurde wegen eines Alarms abgebrochen" erzeugt.

2.10.2 Inbetriebnahme: Maschinendaten

2.10.2.1 NC-spez.: BAG-spezifische NC/PLC-Nahtstellensignale und Betriebsartenumschaltung

Mit dem Maschinendatum wird die Wirksamkeit der BAG-spezifischen NC/PLC-Nahtstellensignale des DB11 und die Kanäle in denen eine Umschaltung der Betriebsart ausgeführt wird, festgelegt:

MD11600 \$MN_BAG_MASK = <Wert>

Wert	Bedeutung
0	Die BAG-spezifischen NC/PLC-Nahtstellensignale des DB11 wirken . Eine interne Umschaltung der Betriebsart erfolgt in allen Kanälen der BAG.
1	Die BAG-spezifischen NC/PLC-Nahtstellensignale des DB11 wirken nicht . Eine interne Umschaltung der Betriebsart erfolgt nur in dem Kanal, in dem ein ASUP aktiv ist.
2	Die BAG-spezifischen NC/PLC-Nahtstellensignale des DB11 wirken . Eine interne Umschaltung der Betriebsart erfolgt nur in dem Kanal, in dem ein ASUP aktiv ist.
3	Die BAG-spezifischen NC/PLC-Nahtstellensignale des DB11 wirken nicht . Eine interne Umschaltung der Betriebsart erfolgt nur in dem Kanal, in dem ein ASUP aktiv ist.

Hinweis

Mehrkanalige Systeme

Soll bei mehrkanaligen Systemen die Funktion "Manuelles Verfahren während Unterbrechung eines ASUP in der Betriebsart JOG" (siehe unten) möglich sein, muss MD11600 \$MN_BAG_MASK auf den Wert "2" oder "3" gesetzt werden.

Siehe auch

Programmierung (SETINT, PRIO) (Seite 163)

2.10.2.2 NC-spez.: ASUP Startfreigabe

Mit dem Maschinendatum wird festgelegt, welche Stoppgründe bei einem ASUP-Start ignoriert werden:

MD11602 \$MN_ASUP_START_MASK, <Bit> = <Wert>

Bit	Wert	Bedeutung
0	0	Bei anstehendem Stopp wegen Stopp-Taste, M0 oder M01 wird das ASUP nicht gestartet.
	1	Trotz anstehendem Stopp wegen Stopp-Taste, M0 oder M01 wird das ASUP gestartet .
2	0	Beim Start eines ASUPs wirken bei anstehender Einlesesperre im Kanal die Einstellungen in den kanalspezifischen Maschinendaten: <ul style="list-style-type: none"> • MD20107 \$MC_PROG_EVENT_IGN_INHIBIT • MD20116 \$MC_IGNORE_INHIBIT_ASUP
	1	Der Start eines Anwender- oder System-ASUPs ist in allen Kanälen der NC auch bei anstehender Einlesesperre im jeweiligen Kanal freigegeben. Die Einstellungen in den kanalspezifischen Maschinendaten wirken nicht : <ul style="list-style-type: none"> • MD20107 \$MC_PROG_EVENT_IGN_INHIBIT • MD20116 \$MC_IGNORE_INHIBIT_ASUP Achtung Die Startfreigabe trotz anstehender Einlesesperre wirkt in allen Kanälen der NC. Sowohl für Anwender- als auch für System-ASUPs. Es wird daher dringend empfohlen statt der NC-spezifischen Freigabe die kanalspezifischen Freigaben zu verwenden.
3	0	Wurde in der Betriebsart JOG ein ASUP gestartet und durch NC-Stop unterbrochen, können in diesem Zustand Achsen nicht manuell verfahren werden.
	1	Wurde in der Betriebsart JOG ein ASUP gestartet und durch NC-Stop unterbrochen, können in diesem Zustand Achsen manuell verfahren werden. Durch NC-Start kann das ASUP fortgesetzt werden. Dabei erfolgt ein automatischer REPOS-Vorgang. Hinweis Ist in der Steuerung mehr als ein Kanal parametrierbar, muss zusätzlich folgendes Maschinendatum gesetzt werden: MD11600 \$MN_BAG_MASK, Bit 1 = 1

Manuelle Startfreigabe

Wird aufgrund der parametrierbaren Startfreigaben ein ASUP nicht **automatisch** gestartet, kann das ASUP dennoch über NC/PLC-Nahtstellensignal (DB21, ... DBX7.1) vom PLC-Anwenderprogramm oder manueller Betätigung von NC-Start gestartet werden.

Hinweis

Das ASUP für "Schnellabheben von der Kontur" (**LIFTFAST**) wird in jedem Fall gestartet.

2.10.2.3 NC-spez.: Wirksamkeit der parametrisierten Startfreigaben

Mit dem Maschinendatum wird eingestellt bis zu welcher ASUP-Priorität (Seite 163), ausgehend von der höchsten Priorität, die Einstellungen in MD11602 \$MN_ASUP_START_MASK wirksam sind:

MD11604 \$MN_ASUP_START_PRIO_LEVEL = <ASUP-Priorität>

Beispiel

MD11604 \$MN_ASUP_START_PRIO_LEVEL = 5

Die Einstellungen in MD11602 \$MN_ASUP_START_MASK sind wirksam für ASUPs der Prioritäten 1 → 5.

2.10.2.4 Kanalspez.: Startfreigabe trotz nicht referenzierter Achsen

Mit dem Maschinendatum wird eingestellt, bei welchen Interrupts das zugehörige ASUP trotz parametrierter Funktion "NC-Startsperre ohne Referenzpunkt" (MD20700 \$MC_REFP_NC_START_LOCK) gestartet wird:

MD20115 \$MC_IGNORE_REFP_LOCK_ASUP, Bit (1 - <Interrupt>) = TRUE

ACHTUNG

System-Interrupts

Mit MD20115 \$MC_IGNORE_REFP_LOCK_ASUP, Bit 8 - 31 werden die den System-Interrupts zugeordneten ASUPs freigegeben.

Durch Bit 8 / Interrupt 9 wird ein ASUP gestartet, welches Verfahrbewegungen beinhaltet.

NC-spezifische ASUP-Startfreigabe

Ist MD11602 \$MN_ASUP_START_MASK, Bit 2 == TRUE, ist für alle Kanäle der NC die ASUP-Startfreigabe trotz parametrierter kanalspezifischer Funktion "NC-Startsperre ohne Referenzpunkt" (MD20700 \$MC_REFP_NC_START_LOCK) gegeben. Es wird daher **dringend empfohlen** statt der NC-spezifischen Freigabe die **kanalspezifischen** Freigabe zu verwenden.

2.10.2.5 Kanalspez.: Startfreigabe trotz Einlesesperre

Mit dem Maschinendatum wird eingestellt, bei welchen Interrupts das zugehörige ASUP trotz anstehender Einlesesperre im Kanal (DB21, ... DBX6.1) gestartet wird:

MD20116 \$MC_IGNORE_INHIBIT_ASUP, Bit (1 - <Interrupt>) = TRUE

ACHTUNG

System-Interrupts

Mit MD20116 \$MC_IGNORE_INHIBIT_ASUP, Bit 8 - 31 werden die den System-Interrupts zugeordneten ASUPs freigegeben.

Durch Bit 8 / Interrupt 9 wird ein ASUP gestartet, welches Verfahrbewegungen beinhaltet.

NC-spezifische ASUP-Startfreigabe

Ist MD11602 \$MN_ASUP_START_MASK, Bit 2 == TRUE, werden in allen Kanäle der NC die kanalspezifischen Einstellungen in MD20116 \$MC_IGNORE_INHIBIT_ASUP ignoriert. Es wird daher **dringend empfohlen** statt der NC-spezifischen Freigabe die **kanalspezifischen** Freigabe zu verwenden.

2.10.2.6 Kanalspez.: Kontinuierliche Abarbeitung trotz Einzelsatz

Mit dem Maschinendatum wird eingestellt, bei welchen Interrupts das zugehörige ASUP trotz aktiver Einzelsatzbearbeitung im Kanal (DB21, ... DBX0.4) **kontinuierlich**, d.h. ohne satzweise Unterbrechung, abgearbeitet wird:

MD20117 \$MC_IGNORE_SINGLEBLOCK_ASUP, Bit (1 - <Interrupt>) = TRUE

Randbedingungen

Die Einstellungen in MD20117 \$MC_IGNORE_SINGLEBLOCK_ASUP wirken nur bei Einzelsatz SBL1 (Hauptlauf-Einzelsatz).

ACHTUNG

System-Interrupts

Mit MD20117 \$MC_IGNORE_SINGLEBLOCK_ASUP, Bit 8 - 31 werden die den System-Interrupts zugeordneten ASUPs freigegeben.

Durch Bit 8 / Interrupt 9 wird ein ASUP gestartet, welches Verfahrbewegungen beinhaltet.

NC-spezifische ASUP-Startfreigabe

Ist MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK, Bit 1 == TRUE, werden in allen Kanäle der NC die kanalspezifischen Einstellungen in MD20117 \$MC_IGNORE_SINGLEBLOCK_ASUP ignoriert. Es wird daher **dringend empfohlen** statt der NC-spezifischen Freigabe die **kanalspezifischen** Freigabe zu verwenden.

2.10.2.7 Kanalspez.: Aktualisierung der Anzeige

Mit dem Maschinendatum wird eingestellt, dass keine Aktualisierung der Anzeige während der Ausführung von ASUPs durchgeführt wird, damit kein Flackern der Anzeige des Programm- und Kanalzustandes an der Bedienoberfläche bei der Ausführung von sehr kurzen ASUPs auftritt:

MD20191 \$MC_IGN_PROG_STATE_ASUP, Bit (1 - <Interrupt>) = TRUE

Hinweis

NC/PLC-Nahtstellensignal

Bei Ausführung eines ASUPs mit unterdrückter Anzeige wird folgendes NC/PLC-Nahtstellensignal gesetzt:

DB21, ... DBX378.1 = 1 ("Stilles" ASUP aktiv)

Systemvariable und NC/PLC-Nahtstellensignale

Die Systemvariablen und NC/PLC-Nahtstellensignale für Programm- und Kanalzustand werden durch die Unterdrückung der Anzeige während der Abarbeitung eines ASUP nicht beeinflusst:

- \$AC_STAT (Kanalzustand)
- \$AC_PROG (Programmzustand)
- DB21, ... DBX35.5 - 7 (Kanalzustand)
- DB21, ... DBX35.0 - 4 (Programmzustand)

2.10.3 Programmierung: Systemvariablen

2.10.3.1 REPOS-Möglichkeit (\$P_REPINF)

Im Zusammenhang mit ASUPs können Programmabläufe entstehen, für die es keine eindeutige Rückkehr zu einem Wiederanfahrpunkt an die Kontur (REPOS) gibt.

Über die Systemvariable kann im ASUP gelesen werden, ob REPOS möglich ist:

<Wert> = \$P_REPINF

Wert	Bedeutung
0	Repositionieren mit REPOS nicht möglich, weil: <ul style="list-style-type: none">• Nicht im ASUP aufgerufen• ASUP aus Reset-Zustand ablief• ASUP aus JOG ablief
1	Repositionieren mit REPOS im ASUP möglich

2.10.3.2 Aktivierungsereignis (\$AC_ASUP)

Über die Systemvariablen \$AC_ASUP können bezüglich des Ereignisses, das zur Aktivierung des ASUPs geführt hat, folgende Informationen gelesen werden:

- Weswegen wurde ASUP aktiviert, z. B. Bit 0: Anwender-Interrupt "ASUP mit Blsync"
- Wodurch wurde das ASUP aktiviert, z. B. Bit 0: NC/PLC-Nahtstellensignal, Digitale-Analoge Schnittstelle
- Welche Art von Fortsetzung ist möglich, z.B. Bit 0: freiwählbar REORG oder RET

2.10.4 Programmierung (SETINT, PRIO)

Zuordnung: Interruptsignal zu NC-Programm

Die Zuordnung eines NC-Programms zu einem Interruptsignal erfolgt mit dem Befehl `SETINT`. Das NC-Programm wird dadurch zu einem ASUP.

Syntax

```
SETINT(<n>) <NC-Programm>
```

Bedeutung

SETINT:	Zuordnung NC-Programm zu einem Interruptsignal	
<n>:	Nummer des Interruptsignals	
	Wertebereich:	0, 1, 2, ... 8
<NC-Programm>:	Programmname	

Beispiel

Programmcode	Kommentar
N20 SETINT(3) ABHEBEN_Z	; IF Eingang 3 == 1
	; THEN ASUP "ABHEBEN_Z" starten

2.10 Asynchrone Unterprogramme (ASUPs)

Zusammen mit SETINT können zusätzlich folgende Anweisungen programmiert werden:

- LIFTFAST
Beim Eintreffen des Interruptsignals wird vor dem Start des ASUPs ein "Schnellabheben des Werkzeugs von der Kontur" durchgeführt. Die Bewegungsrichtung für das Schnellabheben wird durch die Programmanweisung ALF festgelegt.
- BLSYNC
Beim Eintreffen des Interruptsignals wird der laufende Programmsatz noch abgearbeitet und erst danach das ASUP gestartet.

Hinweis

Die Zuordnung Interruptsignal ↔ Teileprogramm wird unter folgenden Bedingungen gelöscht:

- Reset-Zustand des Kanals
- CLRINT-Anweisung im Teileprogramm

Prioritäten

Falls in einem NC-Programm mehrere Interrupts durch SETINT aktiviert werden, müssen den zugeordneten NC-Programmen bzw. ASUPs unterschiedliche Prioritäten zugewiesen werden.

Syntax

PRIO=<Wert>

Bedeutung

PRIO:	Schlüsselwort zur Festlegung der Priorität des Interrupts
<Wert>:	Priorität: 1, 2, 3 ... 128. 1 entspricht der höchsten Priorität.

Beispiel

Programmcode	Kommentar
N20 SETINT(3) PRIO=2 ABHEBEN_Z	; IF Eingang 3 == 1
	; THEN ASUP "ABHEBEN_Z" starten
N30 SETINT(2) PRIO=3 ABHEBEN_X	; IF Eingang 2 == 1
	; THEN ASUP "ABHEBEN_X" starten

Die ASUPs werden in der Reihenfolge der Prioritätswerte nacheinander abgearbeitet, wenn die Eingänge 2 und 3 gleichzeitig geschaltet haben:

1. "ABHEBEN_Z"
2. "ABHEBEN_X".

Weitere Interrupt-spezifische Befehle

Befehl	Bedeutung
SAVE	Wenn in einem ASUP der Befehl <code>SAVE</code> verwendet wird, dann werden mit Ende des ASUP die vor der Unterbrechung im unterbrochenen NC-Programm aktiven G-Befehle, Frames und Transformationen wieder wirksam.
DISABLE ENABLE	Durch Verwendung des Befehlpaares <code>DISABLE</code> <code>ENABLE</code> können Programmabschnitte vor dem Unterbrechen durch ASUPs geschützt werden. Die über <code>SETINT</code> getroffene Zuordnung von Interruptsignal zu NC-Programm bleibt dabei erhalten. Das ASUP wird dann mit der nächsten 0/1-Flanke des Interruptsignals nach <code>ENABLE</code> gestartet.
CLRINT (<n>)	Zuordnung von Interruptsignals n zum mit <code>SETINT</code> zugeordneten NC-Programm löschen.

Weitere Informationen

Programmierhandbuch Arbeitsvorbereitung; Flexible NC-Programmierung > Interruptroutine (ASUP)

Siehe auch

Programmierung (Seite 168)

2.10.5 Randbedingungen

Betriebsartenübergreifender ASUP-Start

Zu prüfende Einstellungen

- MD11600 \$MN_BAG_MASK
- MD11604 \$MN_ASUP_START_PRIO_LEVEL
- Die Priorität der Interruptzuordnung

Empfohlene Einstellungen

NC-spezifische Maschinendaten:

- MD11600 \$MN_BAG_MASK = 'H3'

Hinweis

Bei dieser Einstellung ist zu beachten, dass die BAG-spezifischen NC/PLC-Nahtstellensignale des DB11 nicht mehr auf den Kanal wirken, in dem das ASUP abgearbeitet wird. Sollte dieses Verhalten nicht erwünscht sein, kann alternativ die Einstellung MD11600 \$MN_BAG_MASK = 'H2' verwendet werden (siehe "Inbetriebnahme: Maschinendaten (Seite 158)").

MD11602 \$MN_ASUP_START_MASK = 'H5'

- MD11604 \$MN_ASUP_START_PRIO_LEVEL = 7

2.10 Asynchrone Unterprogramme (ASUPs)

Kanalspezifische Maschinendaten für den Kanal, in dem das ASUP gestartet wird, oder allgemein für alle Kanäle:

- MD20105 \$MC_PROG_EVENT_IGN_REFP_LOCK, Bit <n> = TRUE
<n>: Für alle benötigten ereignisgesteuerten Programmaufrufe (Prog-Events)
- MD20115 \$MC_IGNORE_REFP_LOCK_ASUP, Bit <n> = TRUE
<n>: Für alle **benötigten Anwender-Interrupts**

ACHTUNG
System-Interrupts
Mit MD20115 \$MC_IGNORE_REFP_LOCK_ASUP, Bit 8 ... 31 werden die System-Interrupts freigegeben.
Durch Bit 8 / Interrupt 9 wird ein ASUP gestartet, welches Verfahrbewegungen beinhaltet.

2.10.6 Beispiele

Aktivierung eines ASUP durch einen Interrupt aus einer Synchronaktion heraus


1. Zwei aktive digitale Ein-/Ausgangs-Bytes parametrieren:
 - MD10350 \$MN_FASTIO_DIG_NUM_INPUTS = 2
 - MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS = 2
2. Einen Kurzschluss mit ODER-Verknüpfung von Ausgang 9 auf Eingang 9 parametrieren:
 - Eingang 1, Eingangsbyte 2 = (Ausgang 1, Ausgangsbyte 2) **ODER** (HW-Eingangs-Signale 1, Eingangsbyte 2):
MD10361 \$MN_FASTIO_DIG_SHORT_CIRCUIT[0] = 'H0102B102'
3. Zuordnen des HW-Eingangsbytes zur Interrupt-Programmierung SETINT:
 - Eingangs-Byte 2:
MD21210 \$MC_SETINT_ASSIGN_FASTIN = 2
4. Eingang als ASUP-Auslöser definieren:
 - Eingang 1 im zweiten Eingangs-Bytes, d.h. absolut Eingang 9, startet Programm SYNCASUP
SETINT(1) PRIO=1 SYNCASUP
5. Synchronaktion zum Setzen des Ausgangs definieren:
 - Synchronaktion mit ID 1 setzt den Ausgang 9 immer dann auf 1, wenn der Wert des normierte Bahnparameters $\geq 0,5$ wird:
IDS=1 EVERY \$\$AC_PATHN ≥ 0.5 DO \$A_OUT[9]=1
Durch den Kurzschluss von Ausgang 9 auf Eingang 9 wird Interrupt 1 ausgelöst und das NC-Programm "SYNCASUP" als ASUP gestartet.

2.11 Anwenderspezifisches ASUP für RET und REPOS

2.11.1 Funktion

Funktion

Die Steuerungs-Software enthält ein Siemens-spezifisches ASUP zur Realisierung der Funktionen NC-Programmende (RET) und Wiederanfahren an die Kontur (REPOS). Das System-ASUPs kann vom Maschinenhersteller durch ein anwenderspezifisches ASUP ersetzt werden.

 GEFAHR
Programmierfehler
Das Sicherstellen des fehlerfreien Inhalts des anwenderspezifischen ASUPs, welches das Siemens-spezifische ASUP ("ASUP.SYF") ersetzt, liegt in der alleinigen Verantwortung des Maschinenherstellers.

2.11.2 Parametrierung

Installation

ASUP-Name

Dem anwenderspezifischen ASUP muss folgender Namen gegeben werden:

- `_N_ASUP_SPF`

ASUP-Verzeichnisse

Das anwenderspezifischen ASUP "`_N_ASUP_SPF`" muss in einem der beiden Verzeichnisse abgelegt werden:

- `_N_CMA_DIR` (Hersteller-Verzeichnis)
- `_N_CUS_DIR` (Anwender-Verzeichnis)

Aktivierung und Suchpfad

Bei welchem Ereignis das anwenderspezifischen ASUP "`_N_ASUP_SPF`" aktiviert wird, wird in folgendem Maschinendatum über Bit 0 und 1 eingestellt.

Wo bei Aktivierung des anwenderspezifischen ASUP "`_N_ASUP_SPF`" mit der Suche begonnen wird, wird über Bit 2 eingestellt.

MD11610 \$MN_ASUP_EDITABLE, Bit 0, 1, 2 = <Wert>

Bit	Wert	Bedeutung
0 und 1	0	Weder bei Programmende (RET) noch bei Wiederanfahren an die Kontur (REPOS) wird das anwenderspezifische ASUP aktiviert.
	1	Bei RET wird das anwenderspezifische ASUP aktiviert. Bei REPOS wird das systemspezifische ASUP aktiviert.
	2	Bei RET wird das systemspezifische ASUP aktiviert. Bei REPOS wird das anwenderspezifische ASUP aktiviert.
	3	Bei RET und bei REPOS wird das anwenderspezifische ASUP aktiviert.
2	0	Das anwenderspezifische ASUP wird zuerst im Anwender-Verzeichnis _N_CUS_DIR gesucht.
	1	Das anwenderspezifische ASUP wird zuerst im Hersteller-Verzeichnis _N_CMA_DIR gesucht.

Festlegung einer Schutzstufe

Wenn ein anwenderspezifisches ASUP für RET und/oder REPOS benutzt werden soll (MD11610 \$MN_ASUP_EDITABLE ≠ 0), kann für die anwenderspezifische Routine "_N_ASUP_SPF" eine Schutzstufe festgelegt werden. Die Schutzstufe kann Werte im Bereich 0 - 7 annehmen. Die Einstellung erfolgt über das Maschinendatum:

MD11612 \$MN_ASUP_EDIT_PROTECTION_LEVEL = <Schutzstufe des anwenderspezifischen ASUP>

Details zu den Schutzstufen finden sich in:

Weitere Informationen

Inbetriebnahmehandbuch; Schutzstufenkonzept

Verhalten bei gesetzter Einzelsatzbearbeitung

Über das folgende Maschinendatum wird eingestellt, ob trotz aktiver Einzelsatzbearbeitung die systemspezifischen ASUP und das anwenderspezifischen ASUP "_N_ASUP_SPF" ohne Unterbrechung abgearbeitet werden:

MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK, Bit 0 = <Wert>

Bit	Wert	Bedeutung
0	0	Bei aktivem Einzelsatzbetrieb wird in jedem ASUP-Satz angehalten .
	1	Bei aktivem Einzelsatzbetrieb wird das ASUP ohne Unterbrechung abgearbeitet .

2.11.3 Programmierung

Ermitteln der Ursache der ASUP-Aktivierung

Die Ursache, die zur Aktivierung des ASUP geführt hat, kann bitcodiert über die Systemvariable \$AC_ASUP gelesen werden.

Fortsetzung

Bei Verwendung der System-ASUP ist das Verhalten für die Fortsetzung nach Abarbeitung der Aktionen innerhalb des ASUP fest vorgegeben:

- System-ASUP1 → Fortsetzung mit `RET` (Unterprogrammrücksprung)
- System-ASUP2 → Fortsetzung mit `REPOS` (Repositionieren)

In der Beschreibung der Systemvariablen ist unter "Fortsetzung durch" für jede Ursache das Verhalten bezüglich der System-ASUP angegeben.

Hinweis

Fortsetzung bei anwenderspezifischem ASUP

Es wird empfohlen bei anwenderspezifischen ASUP die entsprechende Fortsetzung der System-ASUP beizubehalten.

Ursache: Betriebsartenwechsel (`$AC_ASUP`, Bit 9 == 1)

Bei Betriebsartenwechsel ist die Fortsetzung abhängig vom Maschinendatum:

MD20114 `$MC_MODESWITCH_MASK` (Unterbrechung MDA durch Betriebsartwechsel)

- Bit 0 == 0: System-ASUP1 → Fortsetzung mit `RET`
 - Bit 0 == 1: System-ASUP2 → Fortsetzung mit `REPOS`
-

Weitere Informationen

Eine ausführliche Beschreibung der Systemvariable findet sich im Listenhandbuch Systemvariablen

2.12 Einzelsatz

2.12.1 Funktion

Die Funktion "Einzelsatz" ermöglicht auf drei verschiedene Arten das satzweise Abarbeiten eines NC-Programms.

Folgende Einzelsatztypen stehen zur Verfügung:

- **"SB1: Einzelsatz grob"** (Stopp nach jedem Satz mit Maschinenfunktion = Hauptlaufsatz)
Das NC-Programm bzw. die Bearbeitung wird nach jedem vollständig abgearbeiteten Satz mit Maschinenfunktion angehalten bzw. gestoppt.
- **"SB2: Rechensatz"** (Stopp nach jedem Satz inklusive Rechensätzen und Kommentarsätzen)
Das NC-Programm bzw. die Bearbeitung wird nach jedem Satz angehalten bzw. gestoppt. Das Verhalten bei der Verwendung von Synchronaktionen ist als Randbedingung formuliert, Einzelsatztyp SB2 und satzbezogene Synchronaktionen (Seite 177)
- **"SB3: Einzelsatz fein"** (Stopp nach jedem Satz mit Maschinenfunktion auch in einem Zyklus)
Wie SB1, jedoch wird auch in Zyklen nach jedem Satz mit Maschinenfunktion angehalten.

Hinweis

Gewindeschneidsätze

Bei Gewindeschneidsätzen wird das NC-Programm nicht angehalten bzw. die Bearbeitung nicht gestoppt.

Bedienoberfläche: Anwahl des Einzelsatztyps

Die Anwahl des Einzelsatztyps SB1, SB2 oder SB3 erfolgt über die HMI-Bedienoberfläche:

Bedienbereich "Maschine" > "Prog.beeinfl." > Menü: "Programmbeeinflussung"

2.12.2 Parametrierung

Nahtstellensignale

Aktivierung

Die Funktion "Einzelsatz" wird kanalspezifisch über das NC/PLC-Nahtstellensignal aktiviert:

DB21, ... DBX0.4 (Einzelsatz aktivieren)

Maschinendaten

Einzelsatzbearbeitung ausschalten (MD10702, MD20106, MD20117)

- Mit dem Maschinendatum kann für bestimmte Bearbeitungssituationen und Programmtypen eingestellt werden, dass trotz aktiver Einzelsatzfunktion nicht angehalten wird:
MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK

Hinweis

- Durch Programmierung (Seite 172) von `SBLON/SBLOF` innerhalb eines ASUPs oder Unterprogramms kann die Einzelsatzbearbeitung explizit ein/ausgeschaltet werden.
 - Bei Einzelsatztyp "SB2: Rechensatz" wirkt das Maschinendatum nur bei System-ASUPs, Anwender-ASUPs und Unterprogrammen mit dem Attribut `DISPLOF`.
-
- Das Verhalten von ereignisgesteuerten Programmaufrufe (Prog-Events) bezüglich Einzelsatz wird eingestellt über:
MD20106 \$MC_PROG_EVENT_IGN_SINGLEBLOCK
 - Das Verhalten von Interruptprogrammen (ASUP) bezüglich Einzelsatz wird eingestellt über:
MD20117 \$MC_IGNORE_SINGLEBLOCK_ASUP
Wird während des Einzelsatzes ein ASUP aktiviert, wird das ASUP vollständig abgearbeitet. Der Einzelsatz wirkt erst wieder nach dem ASUP bzw. im ersten Hauptlaufsatz, in dem keine Einzelsatz-Unterdrückung aktiviert ist. Ist beim Übergang von ASUP zu NC-Programm die Bahngeschwindigkeit zu groß, als dass noch im Folgesatz bis zum Stillstand abgebremst werden könnte, z. B. bei aktivem Bahnsteuerbetrieb G64, erfolgt der Bremsvorgang eventuell auch über mehrere noch folgende Sätze.

Hinweis

Durch Programmierung (Seite 172) von `SBLON` innerhalb eines ASUPs kann in diesem Fall die Einzelsatzbearbeitung **nicht** wieder eingeschaltet werden.

Settingdaten

Debugmode für Einzelsatz "SB2: Rechensatz" (SD42200)

Bedingt durch die vorauseilende Dekodierung der Sätze, kann auf der Bedienoberfläche der Bezug zwischen der hauptlaufbezogenen aktuellen Satzanzeige und den angezeigten Variablenwerten verloren gehen. Es werden dann u. U. nicht plausible Variablenwerte angezeigt.

Mit dem folgenden kanalspezifischen Settingdatum kann eingestellt werden, dass bei aktivem Einzelsatz "SB2: Rechensatz" bei jedem Satz ein Vorlaufstopp ausgeführt wird. Dadurch wird die vorauseilende Bearbeitung der Teileprogrammsätze unterdrückt und der Bezug zwischen aktueller Satzanzeige und Anzeige der Variablenwerte bleibt erhalten.

SD42200 \$SC_SINGLEBLOCK2_STOPRE (Debugmode für SB2 aktivieren)

Hinweis

Konturabweichung

Beim Abarbeiten von Verfahrssätzen mit Einzelsatztyp "**SB2: Rechensatz**" im **Debugmode** kann es zu Konturabweichungen kommen.

2.12.3 Programmierung

2.12.3.1 Einzelsatzbearbeitung aus/einschalten (SBLOF, SBLON)

Einzelsatzbearbeitung für das gesamte NC-Programm ausschalten

Ist das Ausschalten der Einzelsatzbearbeitung (SBLOF) in der ersten Zeile (PROC) eines **Hauptprogramms** programmiert, gilt dies bis zum Ende oder Abbruch des NC-Programms. Das NC-Programm wird dann bei Einzelsatz ohne anzuhalten abgearbeitet.

Ist das Ausschalten der Einzelsatzbearbeitung (SBLOF) in der ersten Zeile (PROC) eines **Unterprogramms** programmiert, gilt dies bis zum Ende oder Abbruch des NC-Programms. Mit dem programmierten 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

Einzelsatzbearbeitung innerhalb des NC-Programms ausschalten

Ist das Ausschalten der Einzelsatzbearbeitung (SBLOF) in einem Satz innerhalb eines NC-Programms programmiert, wird die Einzelsatzbearbeitung ab diesem Satz bis zum nächsten programmierten Einschalten der Einzelsatzbearbeitung (SBLON) oder dem Ende der aktiven Unterprogrammebene ausgeschaltet.

Syntax

SBLOF
SBLON

Bedeutung

SBLOF:	Vordefinierte Prozedur zum Ausschalten der Einzelsatzbearbeitung	
	Alleine im Satz:	ja, im PROC-Satz möglich
	Wirksamkeit:	Modal
SBLON:	Vordefinierte Prozedur zum Einschalten der Einzelsatzbearbeitung	
	Alleine im Satz:	ja
	Wirksamkeit:	Modal

2.12.3.2 Randbedingungen

Einzelsatzunterdrückung und Satzanzeige

Die aktuelle Satzanzeige kann in Unterprogrammen mit `DISPLOF` unterdrückt werden. Wird `DISPLOF` zusammen mit `SBLOF` programmiert, wird bei Einzelsatz-Stopps innerhalb des Unterprogramms der Aufruf des Unterprogramms angezeigt.

Besonderheiten verschiedener Einzelsatztypen

- "SB2: Rechensatz" UND MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK, Bit 12 == 1: Im `SBLON`-Satz wird nicht angehalten.
- "SB3: Einzelsatz fein": Der Befehl `SBLOF` wird unterdrückt

Einzelsatzunterdrückung für asynchrone Unterprogramme (ASUP)

Damit ein ASUP auch bei aktiver Einzelsatzbearbeitung ohne anzuhalten abgearbeitet wird, muss in der ersten Programmzeile des ASUPs `PROC` zusammen mit `SBLOF` programmiert werden.

Beispiel

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

2.12.3.3 Beispiele

Beispiel 1: Einzelsatzunterdrückung innerhalb eines Programms

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

Der Bereich zwischen `N20` und `N60` wird bei aktivem Einzelsatzbetrieb wie ein Satz behandelt.

Beispiel 2: Unterprogramm ohne anzuhalten

Hauptprogramm

Programmcode	
...	
N100	G1 X10 G90 F200
N120	X-4 Y6
N130	CYCLE1
N140	G1 X0
N150	M30

Unterprogramm

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 wird auch bei aktiver Einzelsatzbearbeitung vollständig abgearbeitet.

Beispiel 3: ASUP mit Einzelsatzunterdrückung und nicht sichtbar

Die Bearbeitung eines vom PLC-Anwenderprogramm gestarteten ASUPs soll auch bei aktiver Einzelsatzbearbeitung nicht unterbrochen werden. Zusätzlich soll das ASUP auch nicht sichtbar sein.

Programmcode	
N100 PROC NV SBLOF DISPLOF	; Einzelsatz- und Anzeigeunterdrückung
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	

Programmcode
N200 END: D=\$P_TOOL T=\$P_TOOLNO
N210 RET

Beispiel 4: Gezieltes Anhalten im Unterprogramm

Annahmen:

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

Hauptprogramm

Programmcode	Kommentar
N10 G0 X0	; Einzelsatz-Stopp
N20 X10	; Einzelsatz-Stopp
N30 CYCLE	; Vom Zyklus generierter Verfahrnsatz.
N50 G90 X20	; Einzelsatz-Stopp
M30	

Unterprogramm

Programmcode	Kommentar
PROC CYCLE SBLOF	; Einzelsatzunterdrückung
N100 R0 = 1	
N110 SBLON	; Kein Einzelsatz-Stopp wegen MD10702, Bit12 = 1
N120 X1	; Einzelsatz-Stopp
N140 SBLOF	
N150 R0 = 2	
RET	

Beispiel 5: Einzelsatzunterdrückung bei Programmschachtelung

Annahme: Einzelsatzbearbeitung ist aktiv.

Programmcode	Kommentar
N10 X0 F1000	; Einzelsatz-Stopp
N20 UP1(0)	
PROC UP1(INT _NR) SBLOF	; Einzelsatz für UP1 ausschalten
N100 X10	
N110 UP2(0)	
PROC UP2(INT _NR)	
N200 X20	
N210 SBLON	; Einzelsatz einschalten
N220 X22	; Einzelsatz-Stopp
N230 UP3(0)	
PROC UP3(INT _NR)	

2.12 Einzelsatz

Programmcode	Kommentar
N300 SBLOF	; Einzelsatz ausschalten
N305 X30	
N310 SBLON	; Einzelsatz einschalten
N320 X32	; Einzelsatz-Stopp
N330 SBLOF	; Einzelsatz ausschalten
N340 X34	
N350 M17	; Einzelsatz-Stopp (M17)
N240 X24	; Einzelsatz-Stopp (N210)
N250 M17	; Einzelsatz-Stopp (M17)
N120 X12	
N130 M17	; Einzelsatz-Stopp (M17)
N30 X0	; Einzelsatz-Stopp
N40 M30	; Einzelsatz-Stopp

2.12.4 BAG-spezifischer Einzelsatz Typ A / B

Beim BAG-spezifischen Einzelsatz wird in einem Kanal (Steuerkanal) das NC-Programm per Einzelsatz satzweise abgearbeitet. Im Steuerkanal muss Einzelsatz per NC/PLC-Nahtstellensignal (DB21 ... DBX0.4) aktiviert sein.

In den übrigen Kanälen der BAG (Abhängigen Kanälen) wird das jeweilige NC-Programm satzweise entsprechend dem BAG-spezifisch per NC/PLC-Nahtstellensignal gewähltem Einzelsatz Typ A oder B (DB11 DBX1.6 / 7) abgearbeitet. In den abhängigen Kanälen darf das NC/PLC-Nahtstellensignal (DB21 ... DBX0.4) nicht gesetzt sein.

Einzelsatz Typ A / B

- Einzelsatz Typ A: Stoppt der Steuerkanal, stoppen sofort auch die abhängigen Kanäle, vergleichbar mit NC-Stop.
- Einzelsatz Typ B: Stoppt der Steuerkanal, stoppen die abhängigen Kanäle am jeweiligen Satzende, vergleichbar mit NC-Stop an der Satzgrenze.

Nahtstellensignale

Steuerkanal

- DB21 ... DBX0.4 (Einzelsatz aktivieren)

Alle Kanäle der BAG

- DB21 ... DBX7.1 (NC-Start)

BAG

- DB11 DBX1.6 (Einzelsatz Typ B)
- DB11 DBX1.7 (Einzelsatz Typ A)

Schematischer Ablauf für Einzelsatz Typ A

Voraussetzung: Alle Kanäle der BAG sind im Zustand "Reset" oder "Unterbrochen".

1. PLC-Anwenderprogramm: Einzelsatz im Steuerkanal anwählen, DB21 ... DBX0.4 = 1
2. PLC-Anwenderprogramm: Einzelsatz Typ A für die BAG anwählen, DB11 DBX1.7 = 1
3. PLC-Anwenderprogramm: Alle Kanäle der BAG starten, DB21 ... DBX0.4 = 1
4. Der Steuerkanal stoppt am Satzende.
5. Alle abhängigen Kanäle erhalten ein internes Signal zum sofortigen Stoppen der Bearbeitung.
6. Alle Kanäle der BAG sind im Zustand "Unterbrochen", wenn alle abhängigen Kanäle das jeweilige Ende der Bremsphase erreicht haben.

Schematischer Ablauf für Einzelsatz Typ B

Voraussetzung: Alle Kanäle der BAG sind im Zustand "Reset" oder "Unterbrochen".

1. PLC-Anwenderprogramm: Einzelsatz im Steuerkanal anwählen, DB21 ... DBX0.4 = 1
2. PLC-Anwenderprogramm: Einzelsatz Typ B für die BAG anwählen, DB11 DBX1.6 = 1
3. PLC-Anwenderprogramm: Alle Kanäle der BAG starten, DB21 ... DBX0.4 = 1
4. Der Steuerkanal stoppt am Satzende.
5. Alle abhängigen Kanäle erhalten ein internes Signal zum Stoppen der Bearbeitung am Satzende.
6. Alle Kanäle der BAG sind Zustand "Unterbrochen", wenn alle abhängigen Kanäle ihr jeweiliges Satzende erreicht haben.

2.12.5 Randbedingungen

2.12.5.1 Einzelsatztyp SB2 und satzbezogene Synchronaktionen

Bei Einzelsatztyp "**SB2: Rechensatz**" wird bei einer satzbezogenen Synchronaktion der nächste Stopp erst nach dem nächsten Hauptlaufsatz (Satz mit einer Maschinenfunktion) ausgeführt. Bei den zwischen der Synchronaktion und dem nächsten Hauptlaufsatz liegenden Vorlaufätzen wird nicht angehalten.

2.12.5.2 Programmierter Halt (M0), Einzelsatz und Einzelsatztypumschaltung

Ausgangssituation: In einem Kanal ist ein NC-Programm durch ein darin programmiertes **M0** angehalten und im Kanal ist **Einzelsatz aktiv** (DB21, ... DBX0.4 == 1)

Wird in dieser Situation der **Einzelsatztyp** an der Bedienoberfläche **mehrfach** zwischen SB1 oder SB3 und SB2 umgeschaltet, wird der Alarm 16922 "Maximale Schachtelungstiefe überschritten" angezeigt.

2.13 Programmbeeinflussung

Die Steuerung bietet verschiedene Funktionen, um den Ablauf eines NC-Programms zu beeinflussen:

- Programmsätze ausblenden (Seite 180)
- Größenanpassung des Interpolationspuffers (Seite 144)
- Basis-Satzanzeige (nur bei ShopMill/ShopTurn) (Seite 146)
- Abarbeiten von Extern (Seite 189)
- Abarbeiten von externen Unterprogrammen (EXTCALL) (Seite 191)

2.13.1 Aktivierung

2.13.1.1 Anwahl über Bedienoberfläche

Die Anwahl einer Funktion zur Programmbeeinflussung über die Bedienoberfläche SINUMERIK Operate erfolgt im Bedienbereich "Automatik" > "Programmbeeinflussung" durch Setzen des entsprechenden Anwahlsignals in der HMI/PLC-Nahtstelle. Die PLC überträgt das Anwahlsignal dann in das entsprechende Aktivierungssignal der NC/PLC-Nahtstelle.

Hinweis

Die Übertragung in das entsprechende Aktivierungssignal ist abhängig vom Wert des FB1-Parameters MMCTolf:

- MMCTolf == "TRUE" ; ⇒ Übertragung (Standardeinstellung)
 - MMCTolf == "FALSE" ; ⇒ Keine Übertragung
-

2.13.1.2 Anwahl über PLC-Anwenderprogramm

Eine Funktion zur Programmbeeinflussung kann anwenderspezifisch direkt durch das PLC-Anwenderprogramm durch Setzen des zugehörigen Aktivierungssignals in der NC/PLC-Nahtstelle aktiviert werden.

2.13.1.3 Hinweis zur Anwahl über PLC-Anwenderprogramm

Hinweis

Der FB1-Parameter MMCTolf ist dabei auf "FALSE" zu setzen, da ansonsten die NC/PLC-Nahtstelle durch die Werte der HMI/PLC-Nahtstelle überschrieben würde.

2.13.1.4 Relevante Nahtstellensignale

Funktion	Anwahl (HMI → PLC)	Aktivierung (PLC → NC)	Rückmeldung (NC → PLC) *)
Programmtest (PRT)	DB21, ... DBX25.7	DB21, ... DBX1.7	DB21, ... DBX33.7
Probelaufvorschub (DRY)	DB21, ... DBX24.6	DB21, ... DBX0.6	DB21, ... DBX318.6
Reduzierter Eilgang (RG0)	DB21, ... DBX25.3	DB21, ... DBX6.6	---
Programmierter Halt 1 (M01)	DB21, ... DBX24.5	DB21, ... DBX0.5	DB21, ... DBX32.5
Assoziierte Hilfsfunktion (M-1)	DB21, ... DBX24.4	DB21, ... DBX30.5	DB21, ... DBX318.5
Handrad-Verschiebung (DRF)	DB21, ... DBX24.3	DB21, ... DBX0.3	DB21, ... DBX33.3
"Satz ausblenden" (SKP) /0 - /7	DB21, ... DBX26.0-7	DB21, ... DBX2.0-7	---
"Satz ausblenden" (SKP) /8 - /9	DB21, ... DBX27.0-1	DB21, ... DBX31.6-7	---
Einzelatz (SBx)	Vorwahl von SB1, SB2 oder SB3 über Programmbeeinflussungsanzeige von HMI		
SB1 "Einzelatz grob"	Maschinensteuertafel (MCP): EB n + 5, Bit 2	DB21, ... DBX0.4	---
SB2 "Rechenatz"			
SB3 "Einzelatz fein"			

*) Bei manchen Funktionen erfolgt nach Aktivierung einer Funktion eine Rückmeldung durch die NC.

Weitere Informationen: Funktionshandbuch PLC

2.13.1.5 Status

Die aktuell angewählten Einstellungen zur Programmbeeinflussung sind über Systemvariablen lesbar:

- Im Teileprogramm über die Vorlaufvariablen:

Systemvariable	Typ	Bedeutung
\$P_ISTEST	BOOL	Programmtest aktiv
\$P_DRYRUN	BOOL	Probelaufvorschub aktiv
\$P_ISRG0	BOOL	Reduzierter Eilgang aktiv
\$P_ISPROGSTOP	BOOL	Programmierter Halt 1 (M01) aktiv
\$P_ISDRF	BOOL	Handrad-Verschiebung aktiv
\$P_ISSKIP	BOOL	"Satz ausblenden" aktiv

- Im Teileprogramm und in Synchronaktionen über die Hauptlaufvariable:

Systemvariable	Typ	Bedeutung	
\$AC_PROGINF	INT	Aktive Programmbeeinflussungen Der Wert ist bitcodiert. Die einzelnen Bits haben folgende Bedeutung:	
		Bit 0 ... 9	Ausblendeebene 0 ... 9 aktiv
		Bit 10	Probelaufvorschub aktiv
		Bit 11	Programmierter Halt 1 (M01) aktiv
		Bit 12	Handrad-Verschiebung aktiv
		Bit 13	Einzelatz aktiv
		Bit 14	Reduzierter Eilgang aktiv
		Bit 15	Vorschub-Halt aktiv
		Bit 16	Programmtest aktiv
Bit 17	Assoziierte Hilfsfunktion (M-1) aktiv		

Weitere Informationen: Listenhandbuch Systemvariablen

2.13.2 Programmsätze ausblenden

Funktion

Sätze, die nicht bei jedem Programmlauf ausgeführt werden sollen, können ausgeblendet werden. Die Sätze, die ausgeblendet werden sollen, werden mit dem Zeichen "/" (Schrägstrich) vor der Satznummer gekennzeichnet.

Im Teileprogramm werden die Ausblendeebenen mit "/0" bis "/9" angegeben.

Pro Teileprogrammsatz kann nur eine Ausblendeebene angegeben werden.

Parametrierung

Die Anzahl der Ausblendeebenen wird festgelegt mit dem Maschinendatum:

MD51029 \$MM_MAX_SKP_LEVEL (Max. Anzahl der Ausblendeebenen im NC-Programm)

Programmierung

Sätze, die nicht bei jedem Programmlauf ausgeführt werden sollen (z. B. Programm-Einfahrtsätze), können nach folgendem Schema ausgeblendet werden:

Programmcode	Kommentar
/N005	; Satz ausgeblendet, (DB21, ... DBX2.0) 1.Ausblendeebene
/0 N005	; Satz ausgeblendet, (DB21, ... DBX2.0) 1.Ausblendeebene
/1 N010	; Satz ausgeblendet, (DB21, ... DBX2.1) 2. Ausblendeebene
/2 N020	; Satz ausgeblendet, (DB21, ... DBX2.2) 3.Ausblendeebene
/3 N030	; Satz ausgeblendet, (DB21, ... DBX2.3) 4.Ausblendeebene
/4 N040	; Satz ausgeblendet, (DB21, ... DBX2.4) 5.Ausblendeebene
/5 N050	; Satz ausgeblendet, (DB21, ... DBX2.5) 6.Ausblendeebene
/6 N060	; Satz ausgeblendet, (DB21, ... DBX2.6), 7.Ausblendeebene
/7 N070	; Satz ausgeblendet, (DB21, ... DBX2.7) 8.Ausblendeebene
/8 N080	; Satz ausgeblendet, (DB21, ... DBX31.6) 9.Ausblendeebene
/9 N090	; Satz ausgeblendet, (DB21, ... DBX31.7) 10.Ausblendeebene

Aktivierung

Die 10 Ausblendeebenen "/0" bis "/9" werden vom PLC durch Setzen der Nahtstellensignale PLC → NC aktiviert.

Die Aktivierung der Funktion von HMI über das Menü "Programmbeeinflussung" im Bedienbereich "Maschine" erfolgt:

- Für die Ausblendeebenen "/0" bis "/7":
über die Schnittstelle HMI → PLC DB21, ... DBB26 (Satzausblenden ausgewählt).
- Für die Ausblendeebenen "/8" bis "/9":
über die Schnittstelle HMI → PLC DB21, ... DBX27.0 bis DBX27.1.

Weitere Informationen

Bedienhandbuch

Hinweis

Änderungen der auszubblendenden Ebenen sind nur im STOP/RESET-Zustand der Steuerung möglich.

2.14 ASUP-Start bei Anwenderalarmen durchführen

2.14.1 Funktion

Beschreibung

Ein ASUP kann in verschiedenen Situationen ausgelöst werden, entweder vom Anwender, vom System oder ereignisgesteuert. Ein Anwenderalarm mit der Alarmreaktion "Verriegelung NC-Start" verhindert in manchen Situationen den ASUP-Start. Bei anstehender Alarmreaktion wird ein Anwender-ASUP aus Reset mit dem Alarm 16906 quittiert.

Durch entsprechendes Setzen eines kanalspezifischen Maschinendatums ist es möglich ein ASUP zu starten und auszuführen, obwohl ein Anwenderalarm mit der Alarmreaktion "Verriegelung NC-Start" aktiv ist. Die Alarmreaktion wird für den ASUP-Start überbrückt und lässt die Ausführung zu.

Hinweis

NC-Alarme mit der Alarmreaktion "Verriegelung NC-Start" werden durch die Überbrückung nicht beeinflusst. Ein Anwender-ASUP aus Reset ist nach wie vor nicht möglich und wird mit dem Alarm 16906 abgelehnt.

Überbrückbare Anwenderalarme

Das Nummerband für die überbrückbaren Anwenderalarme ist dafür wie folgt eingeteilt:

Nummernbereich	Auswirkung	Löschen
65000 - 65499	Anzeige, Verriegelung NC-Start	Reset
65500 - 65999	Anzeige, Verriegelung NC-Start (nicht für ASUPs bei eingestelltem MD20194)	Reset
66000 - 66999	Anzeige, Verriegelung NC-Start, Bewegungsstillstand nach Abarbeitung der vordecodierten Sätze	Reset
67000 - 67999	Anzeige	Cancel
68000 - 68999	Anzeige, Verriegelung NC-Start, sofortiger Interpolator-Stopp	Reset
69000 - 69999	Anzeige, Verriegelung NC-Start, Stopp am nächsten Satzende	Reset

Hinweis

Bei folgenden Alarmnummerbändern ist es zwar möglich die Alarmreaktion "Verriegelung NC-Start" zu überbrücken, die anderen Reaktionen der jeweiligen Alarme sorgen aber für Stopp-Bedingungen. Die Stopp-Bedingungen können mit der in diesem Kapitel beschriebenen Funktion nicht überbrückt werden:

- 66000 – 66999
 - 68000 – 68999
 - 69000 – 69999
-

2.14.2 Aktivierung

Einstellung

Über folgendes kanalspezifische Maschinendatum kann jeder ASUP-Kanal separat eingestellt werden:

MD20194 \$MC_IGNORE_NONCSTART_ASUP (ASUP-Start trotz anstehender Alarmreaktion "Verriegelung NC-Start" bei bestimmten Anwenderalarmen zulässig)

Eine Änderung der MD-Einstellung wird erst durch den Teileprogramm-Befehl `NEWCONF` oder über die Bedienoberfläche per Softkey wirksam.

Ablauf

Der normale Ablauf für den ASUP-Start sieht wie folgt aus:

- Maschinendatum entsprechend MD20194 \$MC_IGNORE_NONCSTART_ASUP setzen und per `NEWCONF` aktivieren.
- Teileprogramm starten.
Es erscheint ein Anwenderalarm aus den überbrückbaren Nummernbändern, z. B. Alarm 65500. Das kann aufgrund einer Synchronaktion oder durch einen Teileprogrammbefehl geschehen.
- Das Teileprogramm wird trotz Alarm bis zum Programmende `M02`, `M30` oder `M17` abgearbeitet.
Der Kanalzustand Reset ist aktiv.
- Ein nun gestartetes Anwender-ASUP aus Reset wird ausgeführt.

Hinweis

Ein ASUP-Start aus einem laufenden oder angehaltenen Teileprogramm ist auch bei anstehender Alarmreaktion "Verriegelung NC-Start" erlaubt. Es ist irrelevant, ob es sich dabei um einen Anwenderalarm oder einen NC-Alarm mit der Alarmreaktion "Verriegelung NC-Start" handelt.

Hinweis

Wenn während der Abarbeitung des Anwender-ASUPs das Teileprogramm angehalten wird, kann das ASUP nicht mehr durch die NC-Start-Taste fortgeführt werden, z. B. bei `M0` im ASUP oder bei Anwender-Stopp. Die Alarmreaktion "Verriegelung NC-Start" wird negativ mit dem Alarm 16906 abgelehnt. Der zuvor erzeugte Anwenderalarm kann nur noch mit Reset quittiert werden.

2.14.3 Beispiele

2.14.3.1 Anwender-ASUP aus Reset - Beispiel 1

In der Anwendung ist MD20194 nicht gesetzt.

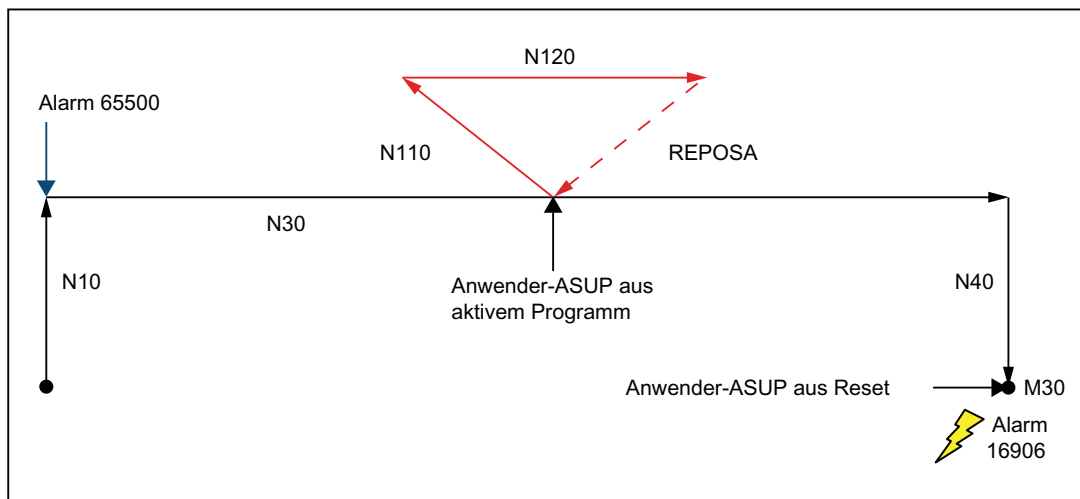
Hauptprogramm

```
Programmcode
N10 G90 G0 Z10
N20 SETAL(65500)
N30 X100
N40 Z0
N50 M30
```

Anwender-ASUP

```
Programmcode
N110 G91 G0 X-10 Z5
N120 X20
N130 REPOSA
```

Ablauf



Der Satz N10 wird abgearbeitet. Es erscheint der Alarm 65500, der die Alarmreaktionen "Anzeige" und "Verriegelung NC-Start" beinhaltet. Das Teileprogramm hält daraufhin nicht an. Der Satz N30 wird eingewechselt und abgefahren. Wenn das Anwender-ASUP in der Satzmitte zum Einsatz kommt, wird es trotz anstehender Alarmreaktion "Verriegelung NC-Start" ausgeführt. Das REPOSA setzt wieder an der Programmunterbrechung an und fährt das Teileprogramm bis zum Programmende M30 ab. Wenn das Anwender-ASUP jetzt zum Einsatz kommt, wird es mit dem Alarm 16906 abgelehnt. Der NC-Zustand befindet sich im Reset.

2.14.3.2 Anwender-ASUP aus Reset - Beispiel 2

Die Anwendung ist der Normalfall, MD20194 ist gesetzt.

Hauptprogramm

Programmcode

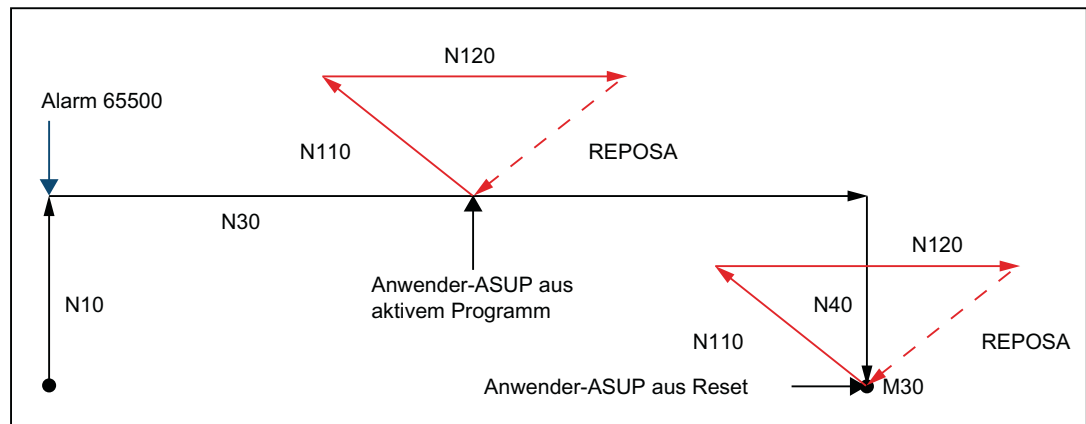
```
N4 $MC_IGNORE_NONCSTART_ASUP=1
N6 NEWCONF
N10 G90 G0 Z10
N20 SETAL(65500)
N30 X100
N40 Z0
N50 M30
```

Anwender-ASUP

Programmcode

```
N110 G91 G0 X-10 Z5
N120 X20
N130 REPOSA
```

Ablauf



Das Maschinendatum MD20194 \$MC_IGNORE_NONCSTART_ASUP wird für ASUP-Kanal 1 gesetzt und per NEWCONF aktiviert. Der Satz N10 wird abgearbeitet. Es erscheint der Alarm 65500, der die Alarmreaktionen "Anzeige" und "Verriegelung NC-Start" beinhaltet. Das Teileprogramm hält daraufhin nicht an. Der Satz N30 wird eingewechselt und abgefahren. Wenn das Anwender-ASUP in der Satzmitte zum Einsatz kommt, wird es trotz anstehender Alarmreaktion "Verriegelung NC-Start" ausgeführt. Das REPOSA setzt wieder an der Programmunterbrechung an und fährt das Teileprogramm bis zum Programmende M30 ab. Wenn das Anwender-ASUP jetzt zum Einsatz kommt, wird es aufgrund des gesetzten Maschinendatums MD20194 wiederholt ausgeführt.

2.14.3.3 Anwender-ASUP mit M0

In der Anwendung ist MD20194 gesetzt.

Hauptprogramm

Programmcode

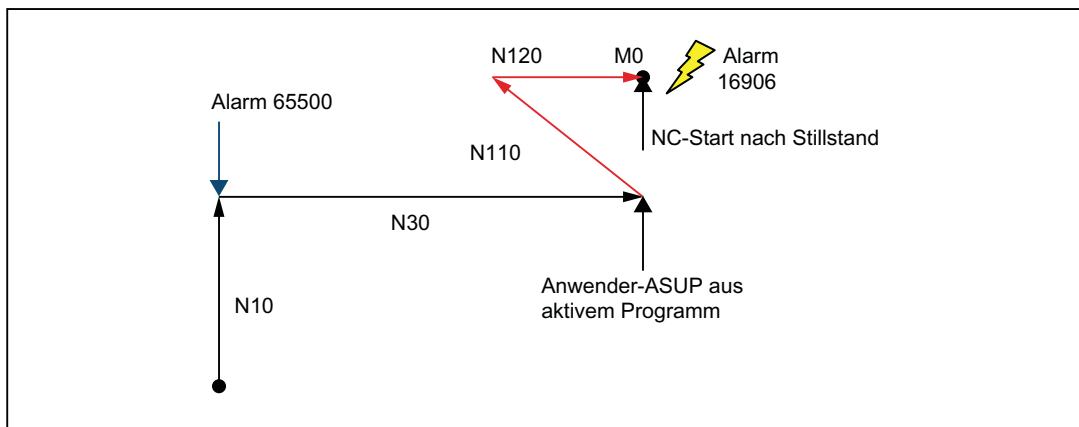
```
N4 $MC_IGNORE_NONCSTART_ASUP=1
N6 NEWCONF
N10 G90 G0 Z10
N20 SETAL(65500)
N30 X100
N40 Z0
N50 M30
```

Anwender-ASUP

Programmcode

```
N110 G91 G0 X-10 Z5
N120 X10
N122 M0
N124 X10
N130 REPOSA
```

Ablauf



Das Maschinendatum MD20194 \$MC_IGNORE_NONCSTART_ASUP wird für ASUP-Kanal 1 gesetzt und per NEWCONF aktiviert. Der Satz N10 wird abgearbeitet. Es erscheint der Alarm 65500, der die Alarmreaktionen "Anzeige" und "Verriegelung NC-Start" beinhaltet. Das Teileprogramm hält daraufhin nicht an. Der Satz N30 wird eingewechselt und abgefahren. Wenn das Anwender-ASUP in der Satzmitte zum Einsatz kommt, wird es trotz anstehender Alarmreaktion "Verriegelung NC-Start" ausgeführt. An Satz N122 hält das Teileprogramm wegen M0 an. Das ASUP kann nicht mehr durch drücken der NC-Start-Taste fortgeführt

werden. Die Alarmreaktion "Verriegelung NC-Start" wird mit dem Alarm 16906 abgelehnt. Der zuvor erzeugte Alarm 65500 kann nur noch mit Reset quittiert werden.

2.14.3.4 Anwender-ASUP mit Stopp

In der Anwendung ist MD20194 gesetzt.

Hauptprogramm

Programmcode

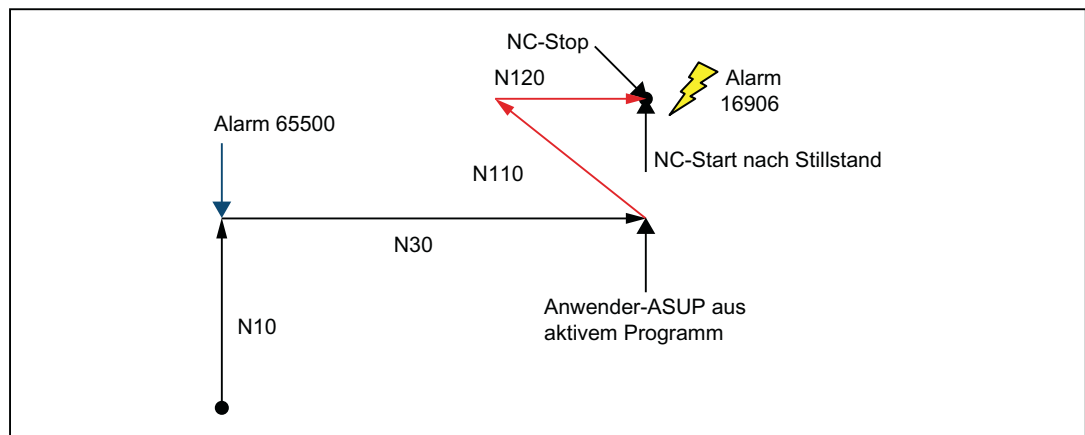
```
N4 $MC_IGNORE_NONCSTART_ASUP=1
N6 NEWCONF
N10 G90 G0 Z10
N20 SETAL(65500)
N30 X100
N40 Z0
N50 M30
```

Anwender-ASUP

Programmcode

```
N110 G91 G0 X-10 Z5
N120 X20
N130 REPOSA
```

Ablauf



Das Maschinendatum MD20194 \$MC_IGNORE_NONCSTART_ASUP wird für ASUP-Kanal 1 gesetzt und per NEWCONF aktiviert. Der Satz N10 wird abgearbeitet. Es erscheint der Alarm 65500, der die Alarmreaktionen "Anzeige" und "Verriegelung NC-Start" beinhaltet. Das Teileprogramm hält daraufhin nicht an. Der Satz N30 wird eingewechselt und abgefahren. Wenn das Anwender-ASUP in der Satzmitte zum Einsatz kommt, wird es trotz anstehender Alarmreaktion "Verriegelung NC-Start" ausgeführt. Wenn bei N120 in der Satzmitte die NC-

Stop-Taste gedrückt wird, hält das ASUP an. Das ASUP kann nicht mehr durch drücken der NC-Start-Taste fortgeführt werden. Die Alarmreaktion "Verriegelung NC-Start" wird mit dem Alarm 16906 abgelehnt. Der zuvor erzeugte Alarm 65500 kann nur noch mit Reset quittiert werden.

2.14.3.5 Anwender-ASUP aus gestoppt

In der Anwendung ist MD20194 gesetzt oder nicht gesetzt.

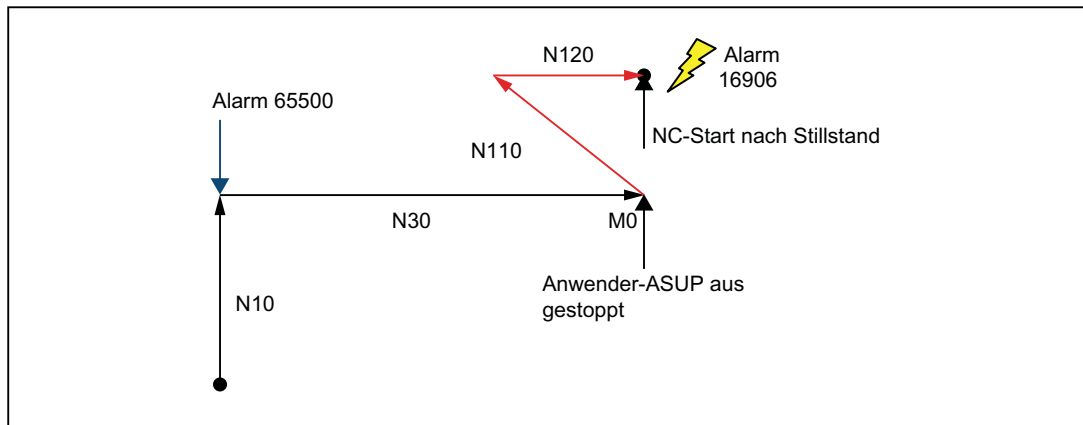
Hauptprogramm

```
Programmcode  
N10 G90 G0 Z10  
N20 SETAL(65500)  
N30 X50  
N35 M0  
N38 X100  
N40 Z0  
N50 M30
```

Anwender-ASUP

```
Programmcode  
N110 G91 G0 X-10 Z5  
N120 X20  
N130 REPOSA
```

Ablauf



Der Satz N10 wird abgearbeitet. Es erscheint der Alarm 65500, der die Alarmreaktionen "Anzeige" und "Verriegelung NC-Start" beinhaltet. Das Teilprogramm hält daraufhin nicht an. Der Satz N30 wird eingewechselt und abgefahren. An Satz N35 hält das Teilprogramm wegen M0 an. Der NC-Zustand befindet sich im Status gestoppt. Wenn das Anwender-ASUP jetzt zum

Einsatz kommt, wird es trotz gestopptem NC-Zustand und anstehender Alarmreaktion "Verriegelung NC-Start" ausgeführt. Es ist dabei egal, ob das Maschinendatum MD20194 \$MC_IGNORE_NONCSTART_ASUP gesetzt ist oder nicht. Nach Satz N120 bleibt das ASUP vor REPOSA stehen. Die Wiederanfahrbewegung ist erst wieder mit dem nächsten NC-Start möglich. Der NC-Start wird aber aufgrund der Alarmreaktion "Verriegelung NC-Start" mit Alarm 16906 abgelehnt. Der zuvor erzeugte Alarm 65500 kann nur noch mit Reset quittiert werden.

2.15 Abarbeiten von Extern

Funktion

Mit der Funktion "Abarbeiten von Extern" können Programme, die aufgrund ihrer Größe nicht mehr im NC-Speicher ablegbar sind, von einem externen Programmspeicher abgearbeitet werden.

Hinweis

Geschützte Zyklen (_CPF Dateien) können **nicht** von einem externen Programmspeicher abgearbeitet werden.

Externe Programmspeicher

Externe Programmspeicher können sich auf folgenden Datenträgern befinden:

- 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
<p>Werkzeug-/Werkstückbeschädigung durch USB-FlashDrive</p> <p>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.</p>

Anwendungen

- **Direktes Abarbeiten von externen Programmen**
Prinzipiell können alle Programme angewählt und abgearbeitet werden, die über die Verzeichnis-Struktur der Bedienoberfläche im HMI-Modus "Abarbeiten von Extern" erreichbar sind.
- **Abarbeiten von externen Unterprogrammen aus dem Teileprogramm**
Aufgerufen wird das "externe" Unterprogramm durch den Teileprogrammbefehl `EXTCALL` mit Angabe eines Aufrufpfades (optional) und des Unterprogrammnamens (→ siehe "Abarbeiten von externen Unterprogrammen (EXTCALL) (Seite 191)").

Parametrierung

Für die Abarbeitung eines Programms im Modus "Abarbeiten von Extern" muss im dynamischen NC-Speicher ein Nachladespeicher (FIFO-Puffer) reserviert werden.

Größe des FIFO-Puffers

Die Größe des FIFO-Puffers wird eingestellt mit dem Maschinendatum:

MD18360 \$MN_MM_EXT_PROG_BUFFER_SIZE (FIFO-Puffer-Größe für Abarbeiten von Extern)

Hinweis

Programme mit Sprunganweisungen

Bei externen Programmen, die Sprunganweisungen enthalten (`GOTOF`, `GOTOB`, `CASE`, `FOR`, `LOOP`, `WHILE`, `REPEAT`, `IF`, `ELSE`, `ENDIF` etc.), müssen die Sprungziele innerhalb des Nachladespeichers liegen.

Hinweis

ShopMill-/ShopTurn-Programme

ShopMill- und ShopTurn-Programme müssen wegen der am Dateieende angefügten Konturbeschreibungen vollständig im Nachladespeicher abgelegt sein.

Anzahl der FIFO-Puffer

Für alle Programme, die gleichzeitig im Modus "Abarbeiten von Extern" abgearbeitet werden, muss jeweils ein FIFO-Puffer zur Verfügung gestellt werden.

Die Anzahl der FIFO-Puffer wird eingestellt mit dem Maschinendatum:

MD18362 \$MN_MM_EXT_PROG_NUM (Anzahl der gleichzeitig von Extern abarbeitbaren Programmebenen)

Verhalten bei Reset, Power-On

Durch Reset und Power-On werden externe Programmaufrufe abgebrochen und die jeweiligen FIFO-Puffer gelöscht.

Ein von einem externen Programmspeicher angewähltes Hauptprogramm wird nach einem Power-On automatisch wieder angewählt, wenn der entsprechende Programmspeicher

weiterhin zur Verfügung steht und die Bearbeitung von EXTCALL-Aufrufen im MD9106 \$MM_SERVER_EXTCALL_PROGRAMS aktiv geschaltet ist.

2.16 Abarbeiten von externen Unterprogrammen (EXTCALL)

Funktion

Bei der Bearbeitung komplexer Werkstücke können sich für die einzelnen Bearbeitungsschritte Programmsequenzen ergeben, die wegen ihres Speicherbedarfs nicht mehr direkt im NC-Speicher ablegbar sind.

In solchen Fällen hat der Anwender die Möglichkeit, mit der Teileprogrammanweisung `EXTCALL` die Programmsequenzen als Unterprogramme von einem externen Programmspeicher im Modus "Abarbeiten von Extern" auszuführen.

Voraussetzungen

Für das Abarbeiten externer Unterprogramme gelten folgende Voraussetzungen:

- Die Unterprogramme müssen über die Verzeichnis-Struktur der Bedienoberfläche erreichbar sein.
- Für jedes Unterprogramm muss im dynamischen NC-Speicher ein Nachladespeicher (FIFO-Puffer) reserviert sein.

Hinweis

Unterprogramme mit Sprunganweisungen

Bei externen Unterprogrammen, die Sprunganweisungen enthalten (`GOTOF`, `GOTOB`, `CASE`, `FOR`, `LOOP`, `WHILE`, `REPEAT`, `IF`, `ELSE`, `ENDIF` etc.), müssen die Sprungziele innerhalb des Nachladespeichers liegen.

Die Größe des Nachladespeichers wird eingestellt über:

MD18360 `MM_EXT_PROG_BUFFER_SIZE`

ShopMill-/ShopTurn-Programme

ShopMill- und ShopTurn-Programme müssen wegen der am Dateiende angefügten Konturbeschreibungen vollständig im Nachladespeicher abgelegt sein.

Parametrierung

Der Pfad zum externen Unterprogrammverzeichnis kann voreingestellt werden mit dem Settingdatum:

SD42700 `$SC_EXT_PROG_PATH` (Programmpfad für externen Unterprogrammaufruf `EXTCALL`)

Zusammen mit dem bei der Programmierung angegebenen Unterprogrammpfad bzw. -name ergibt sich daraus der Gesamtpfad des aufzurufenden Programms.

Hinweis

Soll der Programmpfad nur über die Programmierung angegeben werden, muss SD42700 leer sein!

Programmierung

Der Aufruf eines externen Unterprogramms erfolgt über den Teileprogrammbehehl `EXTCALL`.

Syntax: `EXTCALL("<Pfad/><Programmname>")`

Parameter:

`<Pfad/>`: Absolute oder relative Pfadangabe (**optional**)
Typ: `STRING`

`<Programmname>`: Der Programmname wird ohne Präfix "`_N_`" angegeben.
Die Dateierweiterung ("`MPF`", "`SPF`") kann mit dem Zeichen "`_`" oder "`.`" am Programmnamen angefügt werden (**optional**).
Typ: `STRING`

Hinweis

Pfadangabe: Kurzbezeichnungen

Bei der Pfadangabe können folgende Kurzbezeichnungen verwendet werden:

- **LOCAL_DRIVE**: für lokales Laufwerk
- **CF_CARD**: für CompactFlash-Card
- **USB**: für USB Front-Anschluss

CF_CARD: und **LOCAL_DRIVE**: sind alternativ verwendbar.

EXTCALL-Aufruf mit absoluter Pfadangabe

Wenn das Unterprogramm unter dem angegebenen Pfad existiert, dann wird es nach dem `EXTCALL`-Aufruf ausgeführt. Wenn es nicht existiert, dann wird die Programmausführung 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:

- Wenn in SD42700 \$SC_EXT_PROG_PATH eine Pfadangabe voreingestellt ist, dann 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:
 - der in SD42700 voreingestellten Pfadangabe
 - dem Zeichen "/" als Trennzeichen
 - dem bei EXTCALL angegebenen Unterprogrammpfad bzw. -namen
- Wurde das aufgerufene Unterprogramm unter dem voreingestellten Pfad nicht gefunden, werden als nächstes die Verzeichnisse des Anwenderspeichers nach der Angabe im EXTCALL-Aufruf durchsucht.
- Die Suche endet, wenn das Unterprogramm erstmalig gefunden wurde. Sollte die Suche keinen Treffer ergeben, kommt es zum Programmabbruch.

Beispiel

Abarbeiten von lokalem Laufwerk

Hauptprogramm:

Programmcode
N010 PROC MAIN
N020 ...
N030 EXTCALL ("SCHRUPPEN")
N040 ...
N050 M30

Externes Unterprogramm:

Programmcode
N010 PROC SCHRUPPEN
N020 G1 F1000
N030 X= ... Y= ... Z= ...
N040 ...
...
...
N999999 M17

Das Hauptprogramm "MAIN.MPF" befindet sich im NC-Speicher und ist zur Abarbeitung angewählt.

Das nachzuladende Unterprogramm "SCHRUPPEN.SPF" bzw. "SCHRUPPEN.MPF" befindet sich auf dem lokalen Laufwerk in dem Verzeichnis "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD".

2.17 Abarbeiten vom externen Speicher (EES) (Option)

Der Pfad zu dem Unterprogramm ist im SD42700 voreingestellt:

```
SD42700 $SC_EXT_PROG_PATH = "LOCAL_DRIVE:WKS.DIR/WST1.WPD"
```

Hinweis

Ohne Pfadangabe im SD42700 müsste die `EXTCALL`-Anweisung für dieses Beispiel wie folgt programmiert werden:

```
EXTCALL ("LOCAL_DRIVE:WKS.DIR/WST1.WPD/SCHRUPPEN")
```

2.17 Abarbeiten vom externen Speicher (EES) (Option)

2.17.1 Funktion

Hinweis

Für die Nutzung der Funktion ist die lizenzpflichtige Option "CNC Anwenderspeicher erweitert" oder "Abarbeiten vom externen Speicher (EES)" erforderlich!

Funktion

Mit der Funktion EES (Execution from External Storage) hat der Anwender die Möglichkeit, Programme durch die NC **direkt von einem externen Speicher** abarbeiten zu lassen.

Folgende Laufwerke können als externer Speicher zur Verfügung stehen:

Laufwerk	Symbolischer Name ¹⁾	SINUMERIK 840D sl
NC Extend (ehem. "Lokales Laufwerk")	CF_CARD LOCAL_DRIVE SYS_DRIVE	CF-Card der NCU oder Lokale Festplatte einer PCU
USER CF	CF_CARD LOCAL_DRIVE	-
Netzlaufwerk		
Statisch geführtes USB-Laufwerk		

¹⁾ Bei SINUMERIK 840D sl sind dem Laufwerk NC Extend die symbolischen Namen LOCAL_DRIVE, CF_CARD und SYS_DRIVE fest zugeordnet (⇒ NC Extend kann über LOCAL_DRIVE, CF_CARD und SYS_DRIVE adressiert werden).

ACHTUNG

Werkzeug-/Werkstückbeschädigung durch USB-FlashDrive

Ein USB-FlashDrive kann zum Abarbeiten eines externen Programms **nicht** empfohlen werden. Ein Kommunikationsabbruch zum USB-FlashDrive während der Abarbeitung des Programms durch Kontaktschwierigkeiten, Herausfallen, Abbrechen durch Anstoßen oder versehentliches Abziehen führt zum unkontrollierten Stop der Bearbeitung. Werkzeug und/oder Werkstück könnten dabei beschädigt werden.

Voraussetzungen

Für die Nutzung von EES gelten folgende Voraussetzungen:

- Die lizenzpflichtige Option "CNC Anwenderspeicher erweitert" oder "Abarbeiten vom externen Speicher (EES)" muss gesetzt sein.
- Die Laufwerke, die an der Steuerung als externe Speicher zum Einsatz kommen sollen, müssen als logische Laufwerke projektiert sein (siehe "Inbetriebnahme (Seite 196)").
- Eine Liste verfügbarer IPCs finden Sie unter Verfügbare IPCs (Seite 861).

Arbeitsmodus

Abhängig von der vorhandenen Option und der Laufwerksprojektierung sind verschiedene EES-Arbeitsmodi möglich. Der aktive Arbeitsmodus einer Steuerung wird über das Maschinendatum MD18045 \$MN_EES_MODE_INFO angezeigt:

MD18045	Arbeitsmodus	Option	Externe Speicher
			SINUMERIK 840D sl
= 0	EES nicht aktiv	-	-
= 1	Lokales EES aktiv	6FC5800-0AP77-0YB0 CNC Anwenderspeicher erweitert	Die Nutzung von EES auf einer NCU ist auf den erweiterten Anwenderspeicher (100 MB) der CF-Card begrenzt. Bei Einsatz von EES auf einer PCU kann der gesamte freie Speicher von NC Extend genutzt werden. EES über Netzwerk oder USB ist nicht möglich.
		6FC5800-0AP12-0YB0 zusätzlicher HMI-Anwenderspeicher auf CF-Card der NCU	Bei Einsatz von EES auf einer NCU kann durch die Zusatzoption der lokale Anwenderspeicher auf bis zu 6 GB erweitert werden (abhängig von MD9111). Bei Einsatz von EES auf einer PCU wird die Zusatzoption nicht benötigt.
= 2	Globales EES aktiv	6FC5800-0AP75-0YB0 Abarbeiten vom externen Speicher (EES)	EES ist für alle verfügbaren externen Speicher nutzbar.
		6FC5800-0AP12-0YB0 zusätzlicher HMI-Anwenderspeicher auf CF-Card der NCU	Bei Einsatz von EES auf einer NCU kann durch die Zusatzoption der lokale Anwenderspeicher auf bis zu 6 GB erweitert werden (abhängig von MD9111). Bei Einsatz von EES auf einer PCU wird die Zusatzoption nicht benötigt.

2.17 Abarbeiten vom externen Speicher (EES) (Option)

MD18045	Arbeitsmodus	Option	Externe Speicher
			SINUMERIK 840D sl
= 5	Lokales EES aktiv + Globaler Teileprogramm- speicher	Wie MD18045 = 1, nur dass zusätzlich auf dem erweiterten Anwenderspeicher ein globaler Teileprogramm- speicher (Seite 198) eingerichtet ist.	
= 6	Globales EES aktiv + Globaler Teileprogramm- speicher	Wie MD18045 = 2, nur dass zusätzlich auf einem externen Speicher ein globaler Teile- programm- speicher (Seite 198) eingerichtet ist.	

Eigenschaften

Durch die Funktion EES können die Funktionen "Abarbeiten von Extern (Seite 189)" und "Abarbeiten von externen Unterprogrammen (EXTCALL) (Seite 191)" ersetzt werden.

Die Funktion EES bietet folgende Vorteile:

- Systemweit einheitliches Programmhandling
- Keine Einschränkungen hinsichtlich der verwendbaren Befehle
Die Einschränkungen bei "Abarbeiten von Extern" und "Abarbeiten von externen Unterprogrammen (EXTCALL)", z. B. keine Rückwärtssprünge, Begrenzung der Sprungweite von Sprungbefehlen durch die Größe des Nachladespeichers, sind bei EES aufgehoben.
- Programme können zwischen den verschiedenen Programmspeichern (NC, GDIR, externes Laufwerk) wesentlich einfacher verlagert werden.
- Praktisch keine Einschränkung bezüglich Teileprogrammgröße und Programmanzahl (diese sind nur durch die Kapazität der externen Datenablage limitiert)
- Einheitliche Syntax für den Unterprogramm-Aufruf, unabhängig vom Ablageort des Unterprogramms (kein EXTCALL-Aufruf erforderlich)
- Netzlaufwerke können gemeinsam von mehreren Stationen (PCU/NCU) verwendet werden. Voraussetzung dafür ist eine einheitliche Laufwerksprojektierung für diese Stationen. Daraus ergibt sich eine für alle Stationen einheitliche Sicht auf die Programme.
- Durch die einheitliche Sicht aller Stationen auf die externen Programmspeicher werden Änderungen an den dort abgelegten Programmen konsistent für alle Stationen wirksam.

2.17.2 Inbetriebnahme

2.17.2.1 Projektierung der Laufwerke

Vor Verwendung der Funktion EES müssen die an der Steuerung zum Einsatz kommenden Laufwerke vereinbart werden.

Weitere Informationen

- Inbetriebnahmehandbuch Basesoftware und Bediensoftware
- Bedienhandbuch SINUMERIK Operate

Nach dem Aktivieren der neuen Laufwerksprojektierung können die Programme frei auf die verfügbaren Laufwerke verteilt werden.

Hinweis

In der neu erstellten Laufwerksprojektierung sind eventuell nicht mehr alle vorherigen Laufwerke verfügbar. Der Zugriff auf darauf befindliche Programme ist dann nicht mehr möglich.

Abhilfe: Programme dieser Ablagen vorher auf ein weiterhin zugreifbares Laufwerk kopieren.

Hinweis

Da geschützte Zyklen (_CPF Dateien) systembedingt nur aus dem NC-Teileprogrammspeicher abgearbeitet werden, können sie **nicht** zur Abarbeitung auf einem externen Programmspeicher abgelegt werden.

Der bisherige NC-Teileprogrammspeicher mit den Verzeichnissen MPF.DIR, SPF.DIR und WKS.DIR ist für den EES-Betrieb nicht zwingend erforderlich. Ein System kann auch ohne die Verwendung des NC-Teileprogrammspeichers projektiert werden.

ACHTUNG

Ausführung nicht sichtbarer Programme
--

Selbst wenn der NC-Teileprogrammspeicher aus der Laufwerksprojektierung entfernt wurde, ist er intern im System immer noch vorhanden. Insbesondere bedeutet dies, dass bei der Programmausführung gegebenenfalls dort noch existierende Programme aus dem SPF.DIR unbeabsichtigt zur Ausführung gelangen können.
--

Abhilfe: Wird das System bei der Laufwerksprojektierung ohne NC-Teileprogrammspeicher projektiert, sollten alle dort eventuell noch abgelegten Programme manuell gelöscht werden.

Soll der NC-Teileprogrammspeicher weiterhin genutzt werden, sollte er nicht komplett aus dem System entfernt, sondern bei Bedarf nur mit einer entsprechenden Schutzstufe versehen werden.

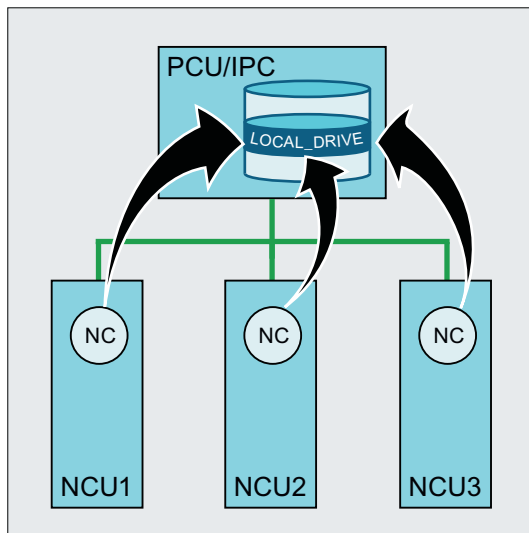
Laufwerke können von mehreren Stationen (PCU/IPC/NCU) gemeinsam genutzt werden. Mit einer einheitlichen Laufwerksprojektierung für diese Stationen erreicht man damit eine stationsunabhängige, einheitliche Sicht auf die Programme.

Hinweis

Die CF-Card einer NCU/PPU kann **nicht** von mehreren Stationen genutzt werden.

Beispiel:

Mehrere NCUs nutzen gemeinsam einen Programmspeicher auf der lokalen Festplatte der PCU (LOCAL_DRIVE).



Hinweis

Werden im EES-Betrieb an verschiedenen Stationen externe Programmspeicher gemeinsam verwendet, sind folgende Regeln zu beachten:

- Ein Programm kann **nicht** gleichzeitig von mehreren Stationen editiert werden.
- Programme, die sich in Abarbeitung befinden, können **nicht** mehr verändert werden.

Bei einem Betrieb mit externem HMI muss die Laufwerksprojektierung auf dem externen HMI erfolgen! Die Laufwerksprojektierung (logdrive.ini) muss aus dem externen HMI in die entsprechende NCU geladen werden. Für die Übertragung steht ein Softkey im Dialog der Laufwerksprojektierung zur Verfügung.

Bei Anlagen, an denen mehrere NCs zusammenarbeiten, muss die Laufwerksprojektierung für alle NCs gleich sein. Dies erreicht man durch Verteilen der Datei logdrive.ini auf alle in der Datei mmc.ini aufgeführten NCUs. Die dort vorhandenen Projektierungen werden dadurch überschrieben.

Weitere Informationen

Eine Liste der PCU-Basesoftware finden Sie unter:

- Inbetriebnahmehandbuch SINUMERIK 840D sl PCU-Basesoftware
- Eine Liste verfügbarer IPCs finden Sie unter Verfügbare IPCs (Seite 861).

2.17.2.2 Globaler Teileprogrammspeicher (GDIR)

Bei der Vereinbarung der Laufwerke kann eines der Laufwerke als globaler Teileprogrammspeicher (GDIR) ausgezeichnet werden.

Weitere Informationen

Bedienhandbuch Universal/Drehen/Fräsen/Schleifen; Programme verwalten > Laufwerke einrichten

Vom System werden automatisch die Verzeichnisse MPF.DIR, SPF.DIR und WKS.DIR auf dem Laufwerk, welches das GDIR beherbergt, angelegt. Diese drei Verzeichnisse bilden das GDIR.

Das GDIR spielt ausschließlich für die Funktion EES eine Rolle. Je nach Laufwerksprojektierung ersetzt oder erweitert das GDIR den NC-Teileprogrammspeicher. Das Einrichten eines GDIR ist für den EES-Betrieb allerdings nicht zwingend erforderlich.

Die Verzeichnisse und Dateien des GDIR können im Teileprogramm 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 GDIR erweitert den Suchpfad für Unterprogramme, die ohne absolute Pfadangabe aufgerufen werden.

GDIR ersetzt den NC-Teileprogrammspeicher

Wenn der NC-Teileprogrammspeicher in den Verzeichnissen MPF.DIR, SPF.DIR und WKS.DIR komplett leer ist, ersetzt das GDIR den NC-Teileprogrammspeicher. Der bisherige NC-Suchpfad wird über das GDIR 1:1 abgebildet.

Anwahl des Hauptprogramms auf einer externen Ablage

Suchreihenfolge für die Unterprogramme:

1. Aktuelles Verzeichnis auf externer Ablage
2. SPF.DIR im GDIR-Speicher
3. Das mit CALLPATH referenzierte Laufwerk
4. Zyklen

GDIR erweitert den NC-Teileprogrammspeicher

Ist der NC-Teileprogrammspeicher in den Verzeichnissen MPF.DIR, SPF.DIR und WKS.DIR nicht leer, so ist die Suchreihenfolge für die Unterprogramme abhängig vom Ablageort des Hauptprogramms (aktives Verzeichnis).

Anwahl des Hauptprogramms im NC-Teileprogrammspeicher (MPF.DIR oder xxx.WPD in WKS.DIR)

Suchreihenfolge für die Unterprogramme:

1. Aktuelles Verzeichnis im NC-Teileprogrammspeicher
2. SPF.DIR im NC-Teileprogrammspeicher
3. Das mit CALLPATH referenzierte Laufwerk
4. Zyklen

Anwahl des Hauptprogramms auf für EES freigegebener externer Ablage

Suchreihenfolge für die Unterprogramme:

1. Aktuelles Verzeichnis auf externer Ablage
2. SPF.DIR im NC-Teileprogrammspeicher

3. SPF.DIR in GDIR
4. Das mit CALLPATH referenzierte Laufwerk
5. Zyklen

Hinweis

Zur Festlegung der Suchreihenfolge siehe auch MD11625
\$MN_FILE_ONLY_WITH_EXTENSION und MD11626 \$MN_CYCLES_ONLY_IN_CYCDIR!

Hinweis

Mit der CALLPATH-Anweisung kann auch auf ein externes Laufwerk verwiesen werden.

2.17.2.3 Einstellungen für Dateihandling im Teileprogramm bei EES

Anlagenweit eindeutige Programmnamen

Werden im EES-Betrieb an verschiedenen Stationen externe Programmspeicher gemeinsam verwendet, kann es bei parallel auf verschiedenen Stationen gleichzeitig durchgeführten Dateioptionen (WRITE, DELETE, ...) zu Zugriffskonflikten kommen. Zur Vermeidung derartiger Zugriffskonflikte wird empfohlen, auf jeder Station einen anlagenweit eindeutigen Namensraum für Dateinamen einzurichten.

Anlagenweit eindeutiger Namensraum

Ein anlagenweit eindeutiger Namensraum für Dateinamen wird z. B. erreicht durch Verknüpfung der Dateinamen mit dem in Maschinendaten parametrierbaren EES-spezifischen Namen der NC und der Kanalnummer des Kanals, in dem das Programm abgearbeitet wird. Die nachfolgende Programmierung erzeugt z. B. bei Abarbeitung des Programms den anlagenweit eindeutigen Dateinamen (MYFILE_NC1_1.SPF) durch Anhängen des EES-spezifischen Namens der NC (NC1) und der Kanalnummer (Kanal 1) an den Programmnamen.

```
$MN_EES_NC_NAME="NC1"  
N10 DEF STRING[31] FILENAME  
N20 FILENAME="MYFILE_" << $MN_EES_NC_NAME << "_" << $P_CHANNO <<  
".SPF"
```

Parametrierung

Der EES-spezifische Name der NC wird eingestellt im NC-spezifischen Maschinendatum:

```
MD10125 $MN_EES_NC_NAME = <NC-Name>
```

Hinweis

Anlagenweit eindeutiger Name der NC

Zur Vermeidung von Zugriffskonflikten muss der EES-spezifische Name der NC anlagenweit eindeutig sein. Die Verantwortung dafür liegt ausschließlich beim Anwender/Maschinenhersteller.

Beim Programmaufruf nur nach Dateien mit Dateikennung suchen

Um im EES-Betrieb die Programmsuche bei Unterprogrammaufrufen zu beschleunigen, wird empfohlen, die Suche auf Dateien mit Dateikennung (z. B. SPF, MPF etc.) zu beschränken:

MD11625 \$MN_FILE_ONLY_WITH_EXTENSION = 1

Hinweis

MD11625 hat keine Auswirkungen auf die Programmsuche beim Abarbeiten von externen Unterprogrammen mit EXTCALL.

Weitere Informationen

Weitere Information zur Beschreibung des Suchpfades für den Unterprogrammaufruf, siehe Programmierhandbuch Arbeitsvorbereitung.

Programme mit Interface nur in den Zyklenverzeichnissen suchen

Um im EES-Betrieb die Programmsuche bei Unterprogrammaufrufen zu beschleunigen, wird empfohlen, die Suche nach Unterprogrammen, die eine Interface-Beschreibung (mittels PROC-Anweisung) angelegt haben und deren Interface-Beschreibung aus einem der Zyklenverzeichnisse (CUS, CMA, CST) erzeugt wurde, auf die Zyklenverzeichnisse zu beschränken:

MD11626 \$MN_CYCLES_ONLY_IN_CYCDIR = 1

Hinweis

Auf Unterprogramme, deren Interface durch eine EXTERN-Erklärung angelegt wurde, hat MD11626 keine Auswirkung. Sie werden in allen Programmverzeichnissen gesucht.

ACHTUNG
Kein Sucherfolg bei Zyklen außerhalb der Zyklenverzeichnisse
Zyklen im aktuellen Verzeichnis und globalen Unterprogrammverzeichnis werden mit der Einstellung MD11626 = 1 nicht mehr gefunden!
Abhilfe: Zyklen immer in den Zyklenverzeichnissen ablegen.

2.17.2.4 Speicherkonfiguration

Endanwender-Programmspeicher im passiven Filesystem verkleinern

Bei aktivem EES kann gegebenenfalls der im passiven Filesystem für den Endanwender vorgesehene Programmspeicher verkleinert werden:

MD18352 \$MN_MM_U_FILE_MEM_SIZE (Endanwenderspeicher für Teileprogramme/Zyklen/Dateien)

Der frei werdende Speicher ist dann z. B. für Werkzeugdaten oder Herstellerzyklen (MD18353 \$MN_MM_M_FILE_MEM_SIZE) nutzbar.

Weitere Informationen

Ausführliche Informationen zur Speicherkonfiguration siehe S7: Speicherkonfiguration (Seite 837).

Nachladespeicher freigeben

Die Funktion EES kann die Funktionen "Abarbeiten von Extern" und "Abarbeiten von externen Unterprogrammen (EXTCALL)" ersetzen.

Um Unterprogramme aus Teileprogrammen mit EXTCALL-Aufrufen statt mit der Funktion "Abarbeiten von externen Unterprogrammen (EXTCALL)" mit EES abarbeiten zu lassen, müssen die EXTCALL-Aufrufe in CALL-Aufrufe umgewandelt und ggf. die Pfadangaben angepasst werden.

Nach vollständiger Umstellung können die für das "Abarbeiten von Extern" und "Abarbeiten von externen Unterprogrammen (EXTCALL)" benötigten Nachladespeicher (FIFO-Puffer) freigegeben werden:

MD18362 \$MN_MM_EXT_PROG_NUM (Anzahl der gleichzeitig von Extern abarbeitbaren Programmebenen) = 0

2.17.3 Randbedingungen

Teach In

Im EES-Betrieb ist die Anwendung der Funktion "Teach In" in der Betriebsart AUTOMATIK **nicht** möglich.

2.18 Process DataShare - Ausgabe auf ein externes Gerät/Datei

2.18.1 Funktion

Mit der Funktion "Process DataShare" ist es möglich, Daten aus einem Teileprogramm heraus auf ein externes Gerät / eine externe Datei zu schreiben, z. B. zum Zweck der Protokollierung von Produktionsdaten oder zur Steuerung von Zusatzaggregaten an einer Steuerung.

Verfügbarkeit

Die Funktion ist verfügbar:

- nur im realen NC (**nicht** in den Simulations-Softwares SNC und VNC).
- nur in Teileprogrammen (**nicht** in Synchronaktionen).
- parallel in allen Bearbeitungskanälen des NC für alle verfügbaren (projektierten) Ausgabegeräte.

Externe Geräte/Dateien

Externe Geräte/Dateien können sein:

- Dateien auf der lokalen CompactFlash Card
Mit lokaler CompactFlash Card ist der Speicher gemeint, auf den mit dem symbolischen Bezeichner LOCAL_DRIVE aus HMI heraus verwiesen wird.

Hinweis

Für die Ausgabe auf das Gerät LOCAL_DRIVE ist bei SINUMERIK 840D sl die Option "Zusätzlich xxx MB HMI-Anwenderspeicher auf CF-Card der NCU" erforderlich.

- Dateien auf einem Netzlaufwerk
- V.24-Schnittstelle

Hinweis

Für die Ausgabe auf die V.24-Schnittstelle ist bei SINUMERIK 840D sl das NCU-Optionsmodul RS232-Schnittstelle erforderlich.

Maximale Anzahl geöffneter externer Geräte

In einem Teileprogramm/Kanal kann auch mehr als ein externes Gerät/Datei belegt werden. Über alle NC-Kanäle hinweg können gleichzeitig maximal 10 Ausgabegeräte geöffnet sein. Zusätzlich gibt es noch zwei reservierte Einträge für Siemens-Zyklen.

Zu den Ausgabegeräten können gleichzeitig maximal 5 Aufträge aktiv sein.

Benutzungsmodus

Pro Ausgabegerät ist beim Öffnen des Geräts vorgebar, ob das Gerät exklusiv nur von einem Kanal aus genutzt werden darf oder ob es geteilt unter den Kanälen, die darauf ausgeben wollen, verwendet werden kann ("Shared"-Modus).

Verhalten bei Teileprogramm-Ende / Kanal-Reset

Mit Teileprogramm-Ende und Kanal-Reset werden alle im Kanal geöffneten externen Geräte/Dateien geschlossen.

Nutzung der Funktion zur Datenübertragung an die Steuerung

ACHTUNG
Datensicherheit
Wird die Funktion Process DataShare verwendet, um von einem externen Gerät über die Ethernet-Schnittstelle X130 Daten an die Steuerung zu schicken, besteht die Möglichkeit, dass die Daten auf der Steuerung durch Dritte verfälscht werden und nicht mehr konsistent sind. Es ist darauf zu achten, dass bei Verwendung der Funktion das Netzwerk gegen den Zugriff durch Dritte gesichert wurde.

2.18.2 Inbetriebnahme

Die Projektierung der zu verwendenden externen Geräte erfolgt in der Datei /oem/sinumerik/nck/extdev.ini bzw. /user/sinumerik/nck/extdev.ini. Sind beide Dateien vorhanden, haben die Einträge im User-Bereich Vorrang. Die Datei kann im Bedienbereich INBETRIEBNAHME unter SYSTEMDATEN/CF-Karte gepflegt werden.

Hinweis

Für die Verwendung von LOCAL_DRIVE und CYC_DRIVE ist keine Projektierung in der Datei extdev.ini erforderlich. Die beiden Geräte sind immer verfügbar, sobald die entsprechende Option gesetzt bzw. die Anwender CompactFlash Card vorhanden ist.

Im Abschnitt [ExternalDevices] der Datei extdev.ini werden die zu verwendenden externen Geräte definiert/aufgezählt. Als Geräte können ein serielles Device (/dev/v24) und bis zu neun Dateien oder Verzeichnisse (/dev/ext/1...9) angegeben werden. Die Notation für die Angabe ist Linux-Schreibweise. Zeilen, die mit ";" beginnen, sind Kommentar und werden überlesen.

Mit Ausnahme von /dev/v24 können die Geräte als Verzeichnispfad – abgeschlossen mit angehängtem "/" – oder als Dateipfad – also mit angehängtem voll qualifizierten Pfad, endend mit einem Dateinamen (ohne endendes "/") – vereinbart werden. Zu einem Gerät mit Verzeichnispfad muss bei Verwendung im Teileprogramm ein Dateiname (Pfad) mit angegeben werden.

Außer für /dev/v24 erfolgt die Definition eines Geräts durch die drei mit Komma getrennten Angaben für "Server", "Pfad" und dem optionalen "Schreibmodus".

Zu Dateien bzw. Verzeichnissen (wirkt dann auf alle Dateien in dem Verzeichnis) kann angegeben werden, ob die Datei nach dem Öffnen überschrieben werden soll ("O" = Overwrite) oder ob die Ausgaben an die Datei angehängt werden sollen ("A" = Append). Standardwert ist "A". Eine nichtexistierende Datei/Verzeichnis wird beim Öffnen neu angelegt.

Für das Gerät V.24-Schnittstelle werden nur die Einstellungen für Baudrate, Daten-Bits, Stopp-Bits, Parität, Protokoll und ggf. Ende in dieser Reihenfolge angegeben.

Für die Dateien, die auf LOCAL_DRIVE erzeugt/abgelegt werden, wird über das Datum LOCAL_DRIVE_MAX_FILESIZE eine maximale Dateigröße in Bytes – gültig einheitlich für alle Dateien – eingestellt. Die Dateigröße wird bei Ausführung eines EXTOPEN-Befehls im Append-Modus überprüft. Optional kann mit dem Datum LOCAL_DRIVE_FILE_MODE der Schreibmodus ("O" = Overwrite, "A" = Append) festgelegt werden. Standardwert ist "A".

Hinweis

Eine Kopiervorlage für die Projektierungsdatei extdev.ini steht im Verzeichnis /siemens/sinumerik/nck zur Verfügung.

Hinweis

Änderungen an der Datei extdev.ini werden erst nach Neustart/Hochlauf des NC wirksam.

Hinweis**USB-Geräte**

Bei SINUMERIK 840D sl können als USB-Geräte nur statisch verbundene USB-Schnittstellen einer TCU projektiert werden. Die Projektierung erfolgt über die Art SERVER:/PATH als Spezifikation für "Server" im obigen Sinne, wobei SERVER der TCU-Name ist, und /PATH die USB-Schnittstelle bezeichnet. Die jeweiligen USB-Schnittstellen einer TCU werden mit "dev0-0", "dev0-1", "dev1-0" angesprochen. Die Pfadangabe beginnt immer mit "/Partition", wobei die Partition durch ihre zweistellige Partitionsnummer oder ihren Partitionsnamen angegeben sein kann und ggf. mit einem Dateipfad bis zum gewünschten Ziel verlängert wird, also z. B.:

```
/dev/ext/8 = "TCU4:/dev0-0, /01/, A"
```

```
/dev/ext/8 = "TCU4:/dev0-0, /01/mydir.dir/"
```

```
/dev/ext/8 = "TCU4:/dev0-0, /myfirstpartition/Mydir.dir/myfile.txt, O"
```

Beispiel

```
[ExternalDevices]
```

```
; Kommentarzeile
```

```
; example for V24
```

```
; /dev/v24 = "9600, 8, 1, none, rts [, etx]"
```

```
; examples for network drives
```

```
; /dev/ext/1 = "[USERNAME[/DOMAIN][%PASSWORD]@]SERVER/SHARE/, /, A"
```

```
; /dev/ext/2 = "[USERNAME[/DOMAIN][%PASSWORD]@]SERVER/SHARE, /myfile.txt, O"
```

```
; /dev/ext/3 = "[USERNAME[/DOMAIN][%PASSWORD]@]SERVER/SHARE, /mydir/, A"
```

```
; /dev/ext/4 = "SERVER:/dev0-0, /01/, A"
```

```
; ...
```

```
; SIEMENS only
```

```
; /dev/cyc/1 = "[USERNAME[/DOMAIN][%PASSWORD]@]SERVER/SHARE, /mydir/, A"
```

```
; /dev/cyc/2 = "[USERNAME[/DOMAIN][%PASSWORD]@]SERVER/SHARE/mydir, /, A"
```

```
LOCAL_DRIVE_MAX_FILESIZE = 50000
```

```
LOCAL_DRIVE_FILE_MODE = "O"
```

Wirksamkeit des EXTOPEN-Parameters <WriteMode>

Durch Angabe des Schreibmodus sowohl bei der Projektierung in der Datei extdev.ini als auch beim EXTOPEN-Aufruf kann es zu Rechtekonflikten kommen, die beim EXTOPEN-Aufruf ggf. mit Fehler quittiert werden:

Wert aus extdev.ini	Wert des EXTOPEN-Parameters		
	"OVR"	"APP"	-
"O"	O	Fehler	O
"A"	Fehler	A	A
-	O	A	A
Erläuterung: O: Modus "Überschreiben" ist wirksam. A: Modus "Anhängen" ist wirksam. Fehler: EXTOPEN-Aufruf wird mit Fehler quittiert.			

LOCAL_DRIVE: Dateiattribute

Die mit EXTOPEN auf LOCAL_DRIVE angelegten Dateien erhalten folgende Dateiattribute:

- Eigentümer: "user" Schreib-/Leserecht gesetzt
- Gruppe: "operator" Schreib-/Leserecht gesetzt

2.18.3 Programmierung

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>)
...
<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	reserviert
	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 SIE-MENS-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 SIE-MENS-Zyklen 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" 		
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.		

<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":	<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>
"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":	Gerät/Datei wird in dem Kanal exklusiv verwendet, kein anderer Kanal kann das Gerät mitverwenden.
"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:	<table border="1"> <tr> <td>"APP":</td> <td> <p>Anhängen</p> <p>Die Datei bleibt in ihrem Inhalt erhalten, Schreibaufrufe fügen an das Ende an.</p> </td> </tr> <tr> <td>"OVR":</td> <td> <p>Überschreiben</p> <p>Der Inhalt der Datei wird gelöscht und durch nachfolgende Schreibaufrufe neu erstellt.</p> </td> </tr> </table>	"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>
	"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>					
<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.</p>						

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
...
    
```

2.18.4 Randbedingungen

Einfluss auf Bahnsteuerbetrieb

Die Befehle EXTOPEN, WRITE und EXTCLOSE lösen jeweils Vorlaufstopp aus und unterbrechen damit den Bahnsteuerbetrieb.

Verhalten bei Satzsuchlauf

Während "Satzsuchlauf mit Berechnung" erfolgt mit WRITE keine Ausgabe. Es werden jedoch die EXTOPEN- und EXTCLOSE-Befehle aufgesammelt und -- nachdem das Suchziel erreicht wurde -- mit NC-Start wirksam gesetzt. Nachfolgende WRITE-Befehle finden damit dieselbe Umgebung wie bei der normalen Programmbearbeitung vor.

Bei Satzsuchlauf mit Berechnung im Modus "Programmtest" (SERUPRO) werden EXTOPEN, WRITE und EXTCLOSE wie bei der normalen Programmbearbeitung ausgeführt.

2.19 Systemeinstellungen für Hochlauf, RESET/Teileprogrammende und Teileprogramm-Start

2.19.1 Systemeinstellungen für Hochlauf, RESET/Teileprogramm-Ende und Teileprogramm-Start

Konzept

Das Verhalten der Steuerung kann über Maschinendaten für folgende Ereignissen eingestellt werden:

- Hochlauf (Power-On)
- Reset / Teileprogrammende
- Teileprogrammstart

Verhalten der Steuerung nach:	Einstellbar mit:
Hochlauf (Power-On) ^{*)}	MD20110 \$MC_RESET_MODE_MASK MD20144 \$MC_TRAFO_MODE_MASK MD20150 \$MC_GCODE_RESET_VALUES
RESET/Teileprogrammende	MD20110 \$MC_RESET_MODE_MASK MD20150 \$MC_GCODE_RESET_VALUES MD20152 \$MC_GCODE_RESET_MODE
Teileprogrammstart	MD20112 \$MC_START_MODE_MASK MD20110 \$MC_RESET_MODE_MASK
^{*)} siehe auch POWER ON (Seite 393)	

Systemeinstellungen nach Hochlauf

MD20110 \$MC_RESET_MODE_MASK, Bit 0 = 0 oder 1

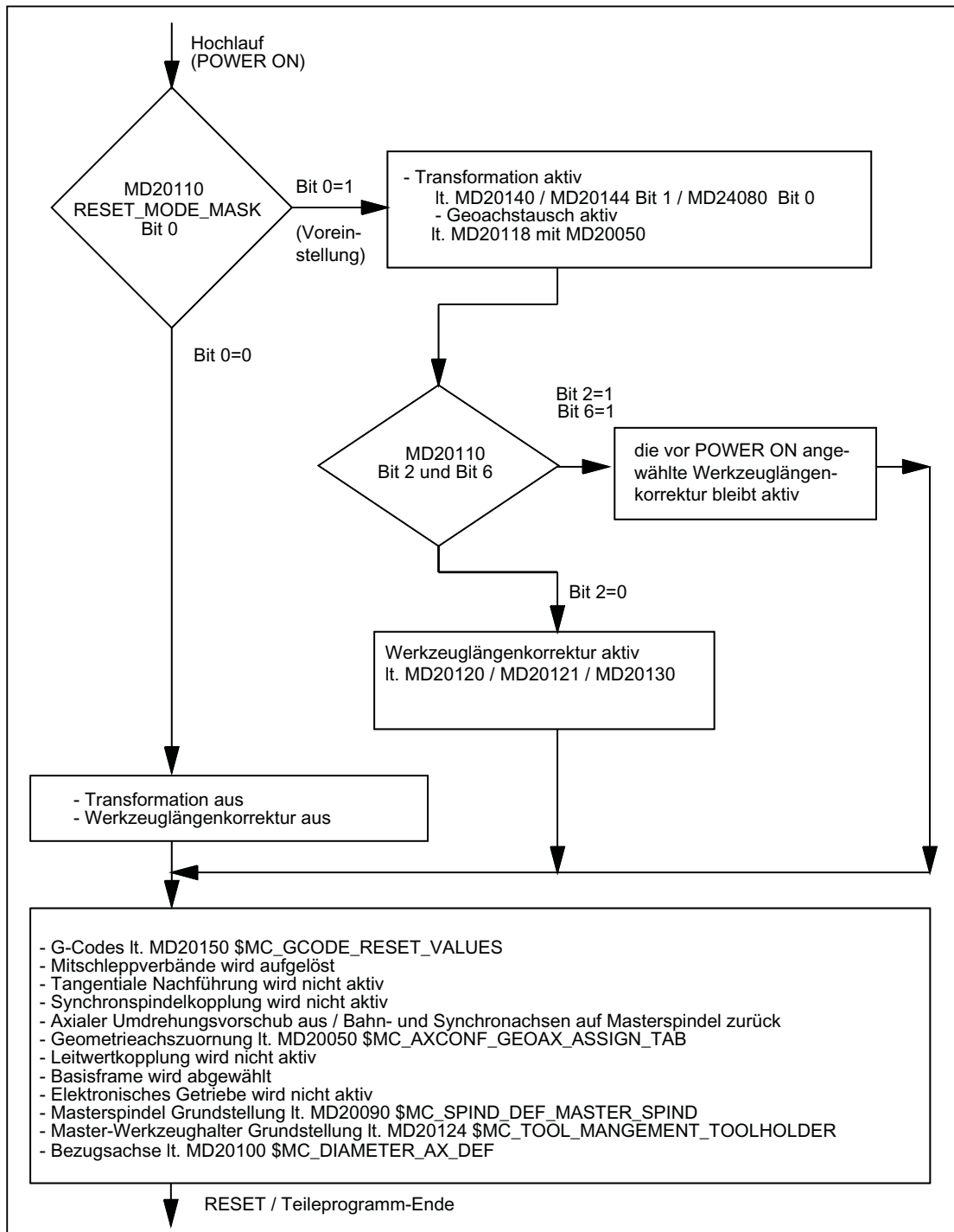


Bild 2-8 Systemeinstellungen nach Hochlauf (Power-On)

Systemeinstellungen nach Reset / Teilprogramm-Ende und Teilprogramm-Start

MD20110 \$MC_RESET_MODE_MASK, Bit 0 = 0 oder 1

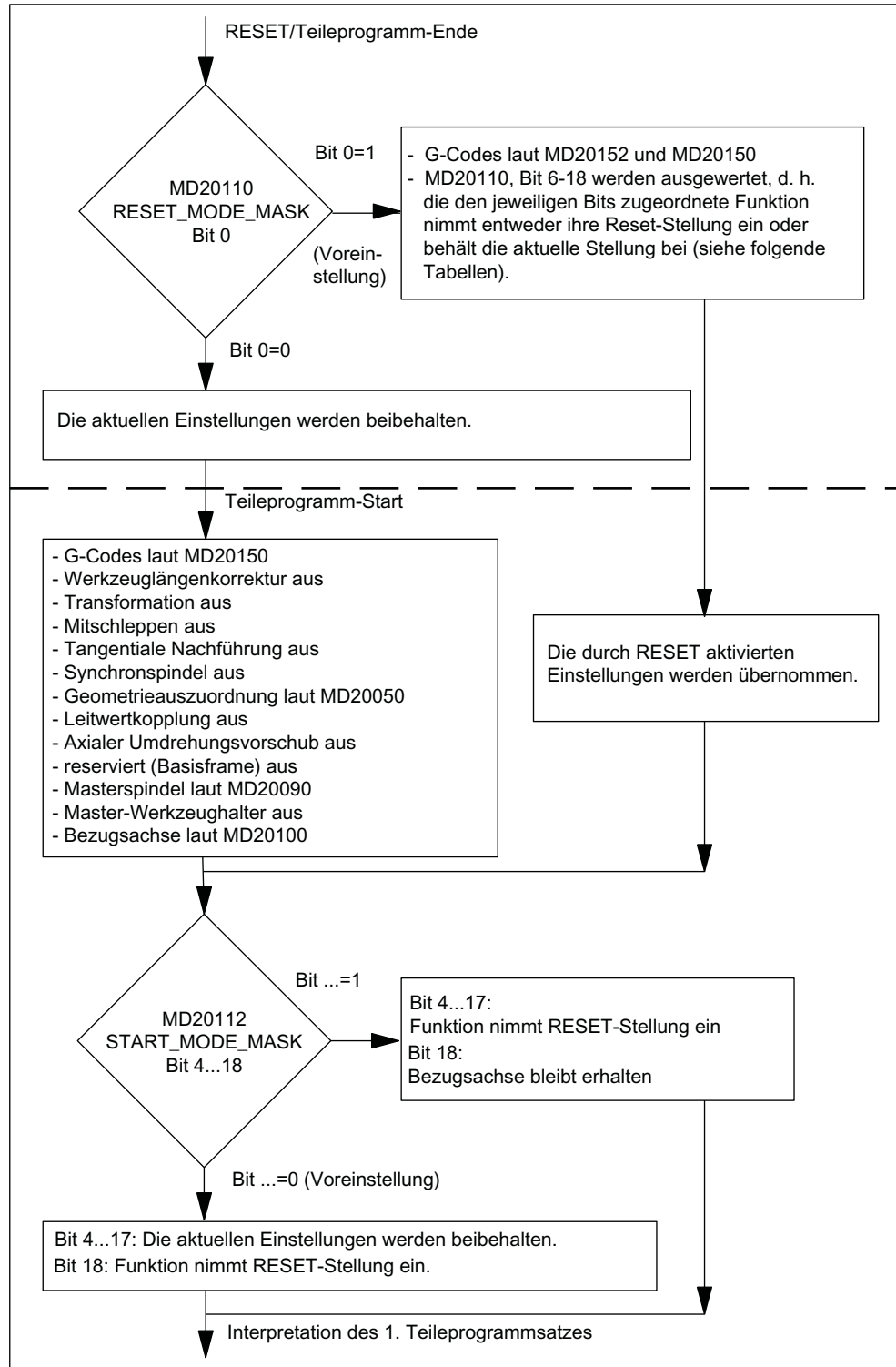


Bild 2-9 Systemeinstellungen nach RESET/Teilprogrammende und Teilprogrammstart

Wirksamer G-Befehl nach Hochlauf und Reset / Teileprogrammende

Die Einstellung des nach Hochlauf (Power-On) und Reset / Teileprogrammende in jeder G-Gruppe wirksamen G-Codes erfolgt über folgende Maschinendaten:

MD20150 \$MC_GCODE_RESET_VALUES[<G-Gruppe>] = <Default-G-Code>

MD20152 \$MC_GCODE_RESET_MODE[<G-Gruppe>] = <Wert>

Wert	Beschreibung: Pro G-Gruppe
0	Der Default-G-Befehl aus MD20150 \$MC_GCODE_RESET_VALUES wird wirksam.
1	Der letzte aktive / aktuelle G-Befehl ist wirksam

Steuerungsgrundstellung nach Hochlauf, Reset / Teileprogrammende und Teileprogrammstart

Die Festlegung der Steuerungsgrundstellung nach Hochlauf (Power-On), Reset / Teileprogrammende und Teileprogrammstart erfolgt über folgende Maschinendaten:

- MD20110 \$MC_RESET_MODE_MASK (Festlegung der Steuerungs-Grundstellung nach **Hochlauf** und **Reset / Teileprogrammende**)
- MD20112 \$MC_START_MODE_MASK (Festlegung der Grundstellung der Steuerung nach **Teileprogrammstart**)

Weitere Literatur

Ausführliche Maschinendaten-Beschreibung

Relevante der Maschinendaten

Maschinendatum	Bedeutung
MD20120 \$MC_TOOL_RESET_VALUE	Werkzeug - Längenkorrektur im Hochlauf, Reset / Teileprogrammende
MD20121 \$MC_TOOL_PRESEL_RESET_VALUE	Vorgewähltes Werkzeug bei Reset
MD20130 \$MC_CUTTING_EDGE_RESET_VALUE	Werkzeugschneide - Längenkorrektur im Hochlauf
MD20140 \$MC_TRAFO_RESET_VALUE	Transformationsdatensatz Hochlauf
MD20144 \$MC_TRAFO_MODE_MASK	Funktionsanwahl der kinematischen Transformation
MD20150 \$MC_GCODE_RESET_VALUES	Löschstellung der G-Gruppen
MD20152 \$MC_GCODE_RESET_MODE	Resetverhalten der G-Gruppen
MD21330 \$MC_COUPLE_RESET_MODE_1	Kopplungs-Abbruch-Verhalten
MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB	Zuordnung Geometrieachse zu Kanalachse
MD20118 \$MC_GEOAX_CHANGE_RESET	Automatischen Geometrieachswechsel erlauben

Beispiel

Reset-Stellung bei Reset aktivieren:

- MD20110, Bit 0 = 1
- MD20112 = 0

Transformation bleibt bei Reset / Teileprogrammstart erhalten:

- MD20110, Bit 0 = 1
- MD20110, Bit 7 = 1
- MD20112 = 0

Werkzeuflängenkorrektur bleibt über Reset / Teileprogrammstart erhalten:

- MD20110, Bit 4 = 1
- MD20110, Bit 6 = 1
- MD20112 = 0

Aktive Ebene (Bit 4) und einstellbarer Frame (Bit 5) bleiben über Reset erhalten und werden bei Teileprogrammstart zurückgesetzt:

- MD20110, Bit 4 = 1
- MD20110, Bit 5 = 1
- MD20112, Bit 4 = 1
- MD20112, Bit 5 = 1

Hinweis

MD20110 / MD20112, Bit 5 und Bit 6

Ist bei Teileprogrammstart in Betriebsart AUTOMATIK bzw. MDA durch entsprechende Parametrierung von MD20110 / MD20112 die Werkzeuflängenkorrektur oder ein Frame aktiv, muss wegen des Verfahrens der Verschiebung, die erste Programmierung der Achsen mit Absolutmaß erfolgen.

Ausnahme: Mit MD42442 / MD42440 wurde das Verfahren der Verschiebung bei G91 unterdrückt.

2.19.2 Werkzeugrückzug nach POWER ON mit Orientierungstransformation

Funktion

Wurde ein Teileprogramm mit einer Bearbeitung mit Werkzeugorientierung durch Spannungsausfall oder Reset abgebrochen, besteht nach dem Hochlauf der Steuerung (Power On) die Möglichkeit, die zuvor aktive Transformation wieder anzuwählen und einen Frame in Richtung der Werkzeugachse zu generieren. Danach kann das Werkzeug in der Betriebsart JOG durch eine Rückzugsbewegung in Richtung der Werkzeugachse freigefahren werden.

Voraussetzung

Für alle an der Transformation beteiligten Maschinenachsen, müssen die aktiven Messsysteme einen Maschinenbezug haben. Siehe Funktionshandbuch "Achsen und SpindelIn", Kapitel "Automatisches Wiederherstellen des Maschinenbezugs".

Parametrierung

Damit die zuletzt aktive Transformation über POWER ON erhalten bleibt, sind folgende Maschinendaten einzustellen:

- MD20144 \$MC_TRAFO_MODE_MASK, Bit 1 = 1
- MD20110 \$MC_RESET_MODE_MASK, Bit 0 = 1
- MD20110 \$MC_RESET_MODE_MASK, Bit 7 = 1

Siehe auch Kapitel "Systemeinstellungen für Hochlauf, RESET/Teileprogramm-Ende und Teileprogramm-Start (Seite 211)".

Programmierung

Warten auf Maschinenbezug WAITENC

Mit dem Befehl WAITENC wird in einem Programm kanalspezifisch gewartet, bis für alle aktiven Messsysteme der parametrierten Achsen ein gültiger Maschinenbezug vorliegt. Siehe oben Abschnitt "Voraussetzung". Die Parametrierung der Achsen erfolgt über:

MD34800 \$MA_WAIT_ENC_VALID = 1

Anwendung

In dem beim Hochlauf (Voraussetzung: MD20108 Bit 3 = 1) aufzurufenden ereignisgesteuerten Anwenderprogramm (.../_N_CMA_DIR/_N_PROG_EVENT_SPF) muss mit dem Befehl WAITENC gewartet werden, bis gültige Achspositionen zur Verfügung stehen.

Danach kann durch den NC-Sprachbefehl TOROTX/TOROTY/TOROTZ ein Frame generiert werden, der die Werkzeugachse in Richtung der X-, Y- oder Z-Achse legt.

Beispiel

Orientierungstransformation und Orientierungsachsen mit inkrementellen Gebern.

Projektiertung:	Bedeutung:
MD10720 \$MN_OPERATING_MODE_DEFAULT [0] = 6	Hochlauf in Betriebsart JOG
MD30240 \$MA_ENC_TYPE [0, <Achse>] = 1	Inkrementelles Messsystem
MD34210 \$MA_ENC_REFP_STATE [0, <Achse>] = 3	Restaurieren der Achspositionen für Inkrementalgeber freigeben.
MD20108 \$MC_PROG_EVENT_MASK = 'H9'	Ereignisgesteuertes Anwenderprogramm (PROG_EVENT) im Hochlauf und bei Teileprogrammstart aktivieren.
MD20152 \$MC_GCODE_RESET_MODE [52] = 1	TOFRAME über Reset erhalten.
MD20110 \$MC_RESET_MODE_MASK = 'HC1'	Transformation und WZK über Reset erhalten.
MD20144 \$MC_TRAFO_MODE_MASK = 'H02'	Transformation über POWER OFF erhalten.

Ereignisgesteuertes Anwenderprogramm (.../_N_CMA_DIR/_N_PROG_EVENT_SPF)

```

; Beispiel mit Aktivierung des Frames, der das WKS in Werkzeugrichtung ausrichtet,
; im Hochlauf und Rücksetzen mit Teileprogrammstart.
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
IF $P_PROG_EVENT == 1 ; Teilprogrammstart.
  TOROTOF ; Werkzeugframe zurücksetzen.
  RET
ENDIF

```

Der Befehl `WAITENC` entspricht prinzipiell folgender Programmsequenz (Beispiel für 5-Achs-Maschine mit AB-Kinematik):

```

WHILE TRUE ; Warten auf Messsystem.
  IF (($AA_ENC_ACTIVE[X]==TRUE) AND ($AA_ENC_ACTIVE[Y]
  ==TRUE) AND ($AA_ENC_ACTIVE[Z]==TRUE) AND ($AA_ENC_ACTIVE[A]
  ==TRUE) AND ($AA_ENC_ACTIVE[B]==TRUE)) GOTOF GET_LABEL
  ENDIF
  G4 F0.5 ; Wartezeit 0,5 s
ENDWHILE
:Position synchronisieren
GET_LABEL: GET(X,Y,Z,A,B,)

```

Bearbeitung fortsetzen**Betriebsart AUTOMATIK**

Zum automatischen Abarbeiten von Programmen in der Betriebsart AUTOMATIK müssen alle Maschinenachsen, deren Istposition des aktive Messsystems restauriert wurde, referenziert werden.

Betriebsart MDA und Überspeichern

In der Betriebsart MDA und für das Überspeichern, kann die Bearbeitung auch ohne Referenzieren der Achsen, mit restaurierten Positionen erfolgen. Dazu muss kanalspezifisch NC-Start mit restaurierten Positionen explizit freigegeben werden:

```
MD20700 $MC_REFP_NC_START_LOCK = 2
```

Randbedingung**Achsen mit inkrementellen Gebern und ohne Istwert-Pufferung**

Es wird davon ausgegangen, dass Achsen mit inkrementellen Gebern und **ohne** Istwert-Pufferung bei Spannungsausfall hinreichend schnell geklemmt werden, um ein Wegdriften von der letzten Sollposition zu verhindern.

2.20 Ersetzung von Funktionen durch Unterprogramme

2.20.1 Übersicht

Funktion

Anwenderspezifische Hilfsfunktionen (z. B. M101) lösen keine Systemfunktionen aus. Sie werden lediglich an die NC/PLC-Nahtstelle ausgegeben. Die Funktionalität der Hilfsfunktion ist vom Anwender/Maschinenhersteller im PLC-Anwenderprogramm zu realisieren. Im weiteren Verlauf wird beschrieben, wie statt der standardmäßigen Ausgabe an die NC/PLC-Nahtstelle der Aufruf eines anwenderspezifischen Unterprogramms (Ersetzungsunterprogramm) projiziert wird.

Im Teileprogramm wird dann weiterhin die Funktion M101 programmiert. Bei der Abarbeitung des Teileprogramms erfolgt aber der Aufruf des Ersetzungsunterprogramms. Die Funktion wird somit von der NC durch einen Unterprogrammaufruf ersetzt. Daraus ergeben sich folgende Vorteile:

- Bei Anpassungen an den Fertigungsprozess kann ein bestehendes, getestetes und bewährtes Teileprogramm unverändert weiter verwendet werden. Die Anpassungen werden in anwenderspezifische Unterprogramme ausgelagert.
- Die Realisierung der Funktionalität kann innerhalb des Ersetzungsunterprogramms mit dem vollen Funktionsumfang der NC-Sprache erfolgen.
- Es entfällt der Kommunikations-Overhead zwischen NC und PLC.

Ersetzbare Funktionen

Folgende Funktionen können durch Unterprogramme ersetzt werden:

Hilfsfunktionen	
M	Schaltfunktionen
T	Werkzeuganwahl
TCA	Werkzeuganwahl unabhängig vom Status des Werkzeugs
D	Werkzeugkorrektur
DL	Additive Werkzeugkorrektur

Spindelbezogene Funktionen bei aktiver Synchronspindelkopplung	
M40	Automatischer Getriebestufenwechsel
M41 - M45	Getriebestufenanwahl 1 ... 5
SPOS	Spindelpositionieren
SPOSA	Spindelpositionieren
M19	Spindelpositionieren

2.20.2 Ersetzung von M-, T/TCA- und D/DL-Funktionen

2.20.2.1 Ersetzung von M-Funktionen

Allgemeine Informationen

Für die Ersetzung von M-Funktionen gelten folgende Bedingungen:

- Pro Satz wird nur eine M-Funktion ersetzt.
- Ein Satz in dem eine M-Funktion ersetzt werden soll, darf folgende Elemente **nicht** enthalten:
 - M98
 - Modalen Unterprogramm-Aufruf
 - Unterprogrammrücksprung
 - Teileprogrammende
- M-Funktionen, die Systemfunktionen auslösen, dürfen nicht durch ein Unterprogramm ersetzt werden (siehe Absatz "M-Funktionen, die nicht ersetzt werden").

Parametrierung

M-Funktion und Unterprogramm

Die Parametrierung der M-Funktionen und der Ersetzungsunterprogramme erfolgt in folgenden Maschinendaten:

- MD10715 \$MC_M_NO_FCT_CYCLE[<Index>] = <M-Funktionsnummer>
- MD10716 \$MC_M_NO_FCT_CYCLE_NAME[<Index>] = "<Unterprogrammname>"

Die M-Funktion und das entsprechende Ersetzungsunterprogramm sind durch den gleichen Index verbunden.

Beispiel: M-Funktion M101 wird durch Unterprogramm SUB_M101 und M-Funktion M102 durch SUB_M102 ersetzt:

MD10715 \$MC_M_NO_FCT_CYCLE[0]	= 101
MD10716 \$MC_M_NO_FCT_CYCLE_NAME[0]	= "SUB_M101"
MD10715 \$MC_M_NO_FCT_CYCLE[1]	= 102
MD10716 \$MC_M_NO_FCT_CYCLE_NAME[1]	= "SUB_M102"

Systemvariable zur Informationsweitergabe

Für eine frei wählbare M-Funktion werden über Systemvariable (siehe Kapitel "Systemvariable (Seite 224)") Informationen bezüglich der ersetzten M-Funktion und weiterer Funktionen (T, TCA, D, DL) zur Auswertung im Ersetzungsunterprogramm zur Verfügung gestellt. Die in den Systemvariablen enthaltenen Daten beziehen sich auf den Satz, in dem die zu ersetzende M-Funktion programmiert wurde.

Die Auswahl der M-Funktion erfolgt mit dem Index des Maschinendatums MD10715 \$MC_M_NO_FCT_CYCLE[<Index>], in dem die zu ersetzende M-Funktion parametrier ist:
MD10718 \$MC_M_NO_FCT_CYCLE_PAR = <Index>

Hinweis

Bei einer M-Funktionsersetzung mit Informationsweitergabe über Systemvariable müssen die Adresserweiterung und der Funktionswert der M-Funktion als konstante Werte programmiert werden.

Zulässige Programmierung:

- M<Funktionswert>
- M=<Funktionswert>
- M[<Adresserweiterung>]=<Funktionswert>

Unzulässige Programmierung:

- M=<Variable1>
 - M[<Variable2>]=<Variable1>
-

Programmierung

Regeln zur Ersetzung von M-Funktionen:

- Das Ersetzungsunterprogramm wird am Satzende aufgerufen
- Innerhalb des Ersetzungsunterprogramms werden keine M-Funktionen ersetzt
- In einem ASUP wird die M-Funktion auch dann ersetzt, wenn das ASUP innerhalb des Ersetzungsunterprogramms gestartet wurde.

M-Funktionen, die nicht ersetzt werden

Folgende M-Funktionen lösen als vordefinierte Hilfsfunktionen Systemfunktionen aus und dürfen **nicht** durch ein Unterprogramm ersetzt werden:

- M0 ... M5
- M17, M30
- M19
- M40 ... M45
- M98, M99 (Nur bei MD18800 \$MN_MM_EXTERN_LANGUAGE ≠ 0)

Die über Maschinendaten parametrieren anwenderspezifischen M-Funktionen dürfen ebenfalls nicht durch ein Unterprogramm ersetzt werden, da auch sie Systemfunktionen auslösen.

Maschinendatum	Bedeutung
MD10714 \$MN_M_NO_FCT_EOP	M-Funktion für Spindel aktiv nach RE-SET
MD10804 \$MN_EXTERN_CHAN_M_NO_SET_INT	M-Funktion für ASUP-Aktivierung (Externmodus)

Maschinendatum	Bedeutung
MD10806 \$MN_EXTERN_CHAN_M_NO_DISABLE_INT	M-Funktion für ASUP-Deaktivierung (Externmodus)
MD10814 \$MN_EXTERN_M_NO_MAC_CYCLE	Makroaufruf über M-Funktion
MD20094 \$MC_SPIND_RIGID_TAPPING_M_NR	M-Funktion für Umschalten in gesteuerten Achsbetrieb
MD20095 \$MC_EXTERN_RIGID_TAPPING_M_NR	M-Funktion für Umschalten in gesteuerten Achsbetrieb (Externmodus)
MD22254 \$MC_AUXFU_ASSOC_M0_VALUE	Zusätzliche M-Funktion für Programm-Halt
MD22256 \$MC_AUXFU_ASSOC_M1_VALUE	Zusätzliche M-Funktion für bedingten Halt
MD26008 \$MC_NIBBLE_PUNCH_CODE	Festlegung der M-Funktionen (für Nibbeln -spezifisch)
MD26012 \$MC_PUNCHNIB_ACTIVATION	Aktivierung der Stanz- und Nibbelfunktionen

Hinweis

Ausnahme

Die über MD22560 \$MC_TOOL_CHANGE_M_CODE (Werkzeugwechsel mit M-Funktion) parametrisierte M-Funktion darf durch ein Unterprogramm ersetzt werden.

2.20.2.2 Ersetzung von T/TCA- und D/DL-Funktionen

Randbedingungen

Für die Ersetzung der Funktionen T, TCA, D und DL gelten folgende Randbedingungen:

- Pro Satz wird maximal eine Funktionsersetzung wirksam.
- Ein Satz mit Funktionsersetzung darf folgende Elemente **nicht** enthalten:
 - M98
 - Modalen Unterprogramm-Aufruf
 - Unterprogrammrücksprung
 - Teileprogrammende
- Erfolgt bei der Multitool-Anwahl mit T = Platznummer die Programmierung der Multitool-Platznummer über die Adresse MTL, so wird mit der T-Ersetzung auch die Adresse MTL ersetzt. Im Ersetzungsunterprogramm kann der programmierte Werte mit den Systemvariablen \$C_MTL_PROG und \$C_MTL abgefragt werden.

Parametrierung: Ersetzungsunterprogramm

Die Angabe des Ersetzungsunterprogramms erfolgt funktionspezifisch im Maschinendatum:

Funktion	Maschinendatum
T	MD10717 \$MN_T_NO_FCT_CYCLE_NAME
TCA	MD15710 \$MN_TCA_CYCLE_NAME
D/DL	MD11717 \$MN_D_NO_FCT_CYCLE_NAME

Hinweis

Es wird empfohlen, für die Ersetzung der T-, TCA- und D/DL-Funktionen dasselbe Unterprogramm zu verwenden.

Parametrierung: Verhalten bezüglich D- bzw. DL-Funktion bei gleichzeitiger T-Funktion

Bei gleichzeitiger Programmierung der Funktionen D bzw. DL und T in einem Satz wird die D- bzw. DL-Nummer entweder als Parameter an das Ersetzungsunterprogramm übergeben oder die D- bzw. DL-Funktion wird vor Aufruf des Ersetzungsunterprogramms ausgeführt. Das Verhalten ist einstellbar über:

MD10719 \$MN_T_NO_FCT_CYCLE_MODE (Parametrierung der T-Funktionsersetzung)

Bit	Wert	Bedeutung
0	0	Die D- bzw. DL-Nummer steht im Unterprogramm per Systemvariable zur Verfügung (Grundstellung).
	1	Die D bzw. DL-Nummer wird direkt im Satz verrechnet. Hinweis: Diese Funktion ist nur aktiv, wenn der Werkzeugwechsel mit M-Funktion projiziert wurde: MD22550 \$MC_TOOL_CHANGE_MODE = 1 Anderweitig werden die D- bzw. DL-Werte immer übergeben.

Systemvariable zur Informationsweitergabe

Dem Ersetzungsunterprogramm werden alle relevanten Informationen zu den im Satz programmierten Funktionen über Systemvariable (siehe Kapitel "Systemvariable (Seite 224)") zur Verfügung gestellt.

Die in den Systemvariablen enthaltenen Daten beziehen sich auf den Satz, in dem die zu ersetzende Funktion programmiert wurde.

Parametrierung: Aufrufzeitpunkt des Ersetzungsunterprogramms

Der Aufrufzeitpunkt des Ersetzungsunterprogramms wird eingestellt über:

MD10719 \$MN_T_NO_FCT_CYCLE_MODE, Bit 1 und Bit 2

Bit 2	Bit 1	Aufrufzeitpunkt des Ersetzungsunterprogramms
0	0	Am Satzende Nach der Bearbeitung des Ersetzungsunterprogramms wird die Interpretation mit der Programmzeile fortgesetzt, die auf diejenige folgt, die den Ersetzungsvorgang ausgelöst hat.
0	1	Am Satzanfang Nach der Bearbeitung des Ersetzungsunterprogramms wird die Programmzeile, die zum Aufruf des Ersetzungsunterprogramms geführt hat, interpretiert. Dabei werden die Adressen T, D bzw. DL und die M-Funktion für den Werkzeugwechsel nicht mehr bearbeitet.
1	-	Am Satzanfang und Satzende Das Ersetzungsunterprogramm wird zweimal aufgerufen.

Systemvariable für den Aufrufzeitpunkt

Über die Systemvariablen \$P_SUB_STAT kann gelesen werden, ob die Ersetzung aktiv ist und wenn ja, wann das Ersetzungsunterprogramm, bezogen auf den Satz, aufgerufen wurde:

Wert	Bedeutung
0	Ersetzung nicht aktiv
1	Ersetzung aktiv, Unterprogrammaufruf am Satzanfang
2	Ersetzung aktiv, Unterprogrammaufruf am Satzende

Beispiel: Ersetzung der T-Funktion

Parametrierung	Bedeutung
MD22550 \$MC_TOOL_CHANGE_MODE = 0	Werkzeugwechsel mit T-Funktion
MD10717 \$MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	Name des Unterprogramms zum Ersetzen der T-Funktion
MD10719 \$MN_T_NO_FCT_CYCLE_MODE = 0	Aufrufzeitpunkt: Satzende

Programmierung	Kommentar
N110 D1	; D1
N120 G90 G0 X100 Y100 Z50	; D1 ist aktiv.
N130 D2 X110 Z0 T5	; D1 bleibt aktiv. Die T-Funktion wird am Satzende durch UP-Aufruf MY_T_CYCLE ersetzt. D2 wird MY_T_CYCLE in einer Systemvariable zur Verfügung gestellt.

Ein ausführliches Beispiel zur Ersetzung der T-Funktion findet sich in Kapitel: "Beispiele zur M/T-Funktionsersetzung beim Werkzeugwechsel (Seite 225)".

2.20.2.3 Systemvariable

Allgemeine Informationen

Dem Ersetzungsunterprogramm werden alle relevanten Informationen zu den im Satz programmierten Funktionen (T bzw. TCA, D bzw. DL, M) über Systemvariable zur Verfügung gestellt.


Ausnahme

Keine Weitergabe der D- bzw. DL-Nummer, wenn:

- MD10719 \$MN_T_NO_FCT_CYCLE_MODE, Bit 0 = 1
- MD22550 \$MC_TOOL_CHANGE_MODE = 1

UND

- D bzw. DL sind zusammen mit T- oder M-Funktion in einem Satz programmiert.

 **VORSICHT**

Werte sind nicht wirksam

Die dem Ersetzungsunterprogramm in den Systemvariablen zur Verfügung gestellten Werte sind noch nicht wirksam. Es liegt in der alleinigen Verantwortung des Anwenders/ Maschinenherstellers dies durch geeignete Programmierung im Ersetzungsunterprogramm "nachzuholen".

Systemvariable

Systemvariable	Bedeutung
\$C_M_PROG	TRUE, wenn M-Funktion programmiert wurde
\$C_M	Enthält bei \$C_M_PROG == TRUE den Wert der Adresse M Dabei sind zwei Fälle zu unterscheiden: <ul style="list-style-type: none"> • Ist für den Werkzeugwechsel mit M-Funktion ein Unterprogramm mit Parameterübergabe projektiert, liefert \$C_M den Wert aus: MD10715 MN_M_NO_FCT_CYCLE • Ist nur ein Unterprogramm für die Adressen T und/oder D/DL projektiert und im Programm ist die M-Funktion für den Werkzeugwechsel zusammen mit einer der zu ersetzenden Adressen programmiert, liefert \$C_M den Wert aus: MD22560 \$MN_TOOL_CHANGE_M_MODE
\$C_AUX_VALUE[0]	Wert der ersetzten M-Funktion
\$C_ME	Enthält bei \$C_M_PROG == TRUE den Wert der Adresserweiterung der M-Funktion
\$C_AUX_EXT[0]	Adresserweiterung der M-Funktion (identisch zu \$C_ME)
\$C_AUX_IS_QUICK[0]	TRUE, wenn M-Funktion mit schneller Ausgabe an PLC programmiert wurde
\$C_T_PROG	TRUE, wenn T-Funktion programmiert wurde
\$C_T	Enthält bei \$C_T_PROG == TRUE den Wert der T-Funktion

Systemvariable	Bedeutung
\$C_TE	Enthält bei: <ul style="list-style-type: none"> • \$C_T_PROG == TRUE • \$C_TS_PROG == TRUE den Wert der Adresserweiterung der T-Funktion
\$C_TS_PROG	TRUE, wenn bei der T- oder TCA-Ersetzung ein Werkzeugname programmiert wurde.
\$C_TS	Enthält bei \$C_TS_PROG == TRUE den bei T- oder TCA-Ersetzung programmierten Werkzeugname
\$C_TCA	TRUE, wenn die TCA-Ersetzung aktiv ist
\$C_DUPLO_PROG	TRUE, wenn die Duplo-Nummer bei der TCA-Ersetzung programmiert wurde
\$C_DUPLO	Enthält bei \$C_DUPLO_PROG == TRUE den Wert der programmierten Duplo-Nummer
\$C_THNO_PROG	TRUE, wenn die Toolholder/Spindel-Nummer bei der TCA-Ersetzung programmiert wurde
\$C_THNO	Enthält bei \$C_THNO_PROG == TRUE den Wert der programmierten Toolholder/Spindel-Nummer
\$C_D_PROG	TRUE, wenn D-Funktion programmiert wurde
\$C_D	Enthält bei \$C_D_PROG == TRUE den Wert der D-Funktion
\$C_DL_PROG	TRUE, wenn DL-Funktion programmiert wurde
\$C_DL	Enthält bei \$C_DL_PROG == TRUE den Wert der DL-Funktion
\$P_SUB_STAT	Satzbezogener Aufrufzeitpunkt des Ersetzungsunterprogramms
\$C_MTL_PROG	TRUE, wenn Adresse MTL programmiert wurde
\$C_MTL	Enthält bei \$C_MTL_PROG == TRUE den Wert der Adresse MTL

2.20.2.4 Beispiel: Ersetzung einer M-Funktion

Beispiel 1

Die Funktion M6 wird durch den Aufruf des Unterprogramms "SUB_M6" ersetzt.

Die für einen Werkzeugwechsel relevanten Informationen sollen über Systemvariable übergeben werden.

Parametrierung

Maschinendatum

MD10715 \$MN_M_NO_FCT_CYCLE[2] = 6

MD10716 \$MN_M_NO_FCT_CYCLE_NAME[2] = "SUB_M6"

MD10718 \$MN_M_NO_FCT_CYCLE_PAR = 2

Bedeutung

Werkzeugwechsel mit M6

Ersetzungsunterprog. für M6

Info.weitergabe über Systemvariable

Hauptprogramm

Programmierung	Kommentar
PROC MAIN	
...	;
N10 T1 D1 M6	; M6 wird durch Unterprogramm "SUB_M6"
	; ersetzt
...	;
N90 M30	

Unterprogramm "SUB_M6"

Programmierung	Kommentar
PROC SUB_M6	
N110 IF \$C_T_PROG==TRUE	; IF Adresse T programmiert
N120 T[\$C_TE]=\$C_T	; T-Anwahl ausführen
N130 ENDIF	; ENDIF
N140 M[\$C_ME]=6	; Werkzeugwechsel ausführen.
N150 IF \$C_D_PROG==TRUE	; IF Adresse D programmiert
N160 D=\$C_D	; D-Anwahl ausführen
N170 ENDIF	; ENDIF
N190 M17	

Beispiele 2

Das neue Werkzeug wird mit der T-Funktion für den Wechsel vorbereitet. Der Werkzeugwechsel erfolgt erst mit der Funktion M6. Die T-Funktion wird durch den Aufruf des Unterprogramms "MY_T_CYCLE " ersetzt. Die D / DL-Nummer wird an das Unterprogramm übergeben.

Parametrierung

Parametrierung	Bedeutung
MD22550 \$MC_TOOL_CHANGE_MODE = 1	Werkzeugwechsel vorbereiten mit T-Funktion
MD10717 \$MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	Ersetzungsunterprogramm
MD10719 \$MN_T_NO_FCT_CYCLE_MODE = 0	Übergabe der D/DL-Nummer

Hauptprogramm

Programmcode	Kommentar
N210 D1	;
N220 G90 G0 X100 Y100 Z50	; D1 ist aktiv.
N230 D2 X110 Z0 T5	; D1 bleibt aktiv, programmiertes D2 wird
	; dem Unterprogramm als Variable übergeben
N240 M6	; Werkzeugwechsel ausführen

Beispiele 3

Das neue Werkzeug wird mit der T-Funktion für den Wechsel vorbereitet. Der Werkzeugwechsel erfolgt erst mit der Funktion M6. Die T-Funktion wird durch den Aufruf des Unterprogramms "MY_T_CYCLE " ersetzt. Die D / DL-Nummer wird **nicht** an das Unterprogramm übergeben.

Parametrierung

Parametrierung	Bedeutung
MD22550 \$MC_TOOL_CHANGE_MODE = 1	Werkzeugwechsel vorbereiten mit T-Funktion
MD10717 \$MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	Ersetzungsunterprogramm
MD10719 \$MN_T_NO_FCT_CYCLE_MODE = 1	Keine Übergabe der D/DL-Nummer

Hauptprogramm

Programmcode	Kommentar
N310 D1	
N320 G90 G0 X100 Y100 Z50	; D1 ist aktiv.
N330 D2 X110 Z0 T5	; D2 wird aktiv und nicht als Variable an ; das Ersetzungsunterprogramm übergeben.
N340 M6	; Werkzeugwechsel ausführen.

Beispiele 4

Die Funktionen T und M6 werden durch das Unterprogramm "MY_T_CYCLE" ersetzt.

Die Parameter werden bei der Ersetzung von M6 an das Unterprogramm übergeben.

Wird M6 zusammen mit D bzw. DL im Satz programmiert, wird die D- bzw. DL-Nummer auch dann als Parameter an das Unterprogramm übergeben, wenn keine Übergabe der D/DL-Nummer parametrier ist:

MD10719 \$MN_T_NO_FCT_CYCLE_MODE = 1

Parametrierung

Projektierung	Bedeutung
MD22550 \$MC_TOOL_CHANGE_MODE = 1	Werkzeugwechsel mit M-Funktion
MD22560 \$MC_TOOL_CHANGE_M_CODE = 6	M-Code für Werkzeugwechsel
MD10715 \$MC_M_NO_FCT_CYCLE[3] = 6	Zu ersetzende M-Funktion
MD10716 \$MC_M_NO_FCT_CYCLE_NAME[3] = "MY_T_CYCLE"	Ersetzungsunterprogramm für M-Funktion
MD10717 \$MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	Ersetzungsunterprogramm für T-Funktion
MD10718 \$MN_M_NO_FCT_CYCLE_PAR = 3	Parameterübergabe an das Ersetzungsunterprog. für M6
MD10719 \$MN_T_NO_FCT_CYCLE_MODE = 1	Keine Übergabe der D/DL-Nummer

Hauptprogramm

Programmcode	Kommentar
N410 D1	
N420 G90 G0 X100 Y100 Z50	; D1 ist aktiv.
N330 D2 X110 Z0 T5 M6	; D1 bleibt aktiv, D2 und T5 werden dem M6-Ersetzungsunterprogramm als Variablen übergeben.

2.20.2.5 Beispiel: Ersetzung einer T- und D-Funktion

Die Funktionen T und D werden durch den Aufruf des Unterprogramms "D_T_SUB_PROG" ersetzt. Für das Beispiel soll weiter gelten:

- Der Werkzeugwechsel erfolgt mit der Adresse T.
- Der Unterprogrammaufruf erfolgt am Satzanfang.
- Die Werkzeugverwaltung ist nicht aktiv.
- Die Achse B ist eine Teilungsachse mit Hirth-Verzahnung.

Parametrierung

Maschinendatum

MD11717 \$MN_D_NO_FCT_CYCLE_NAME = "D_T_SUB_PROG"

MD10717 \$MN_T_NO_FCT_CYCLE_NAME = "D_T_SUB_PROG"

MD10719 \$MN_T_NO_FCT_CYCLE_MODE = 'H2'

MD22550 \$MC_TOOL_CHANGE_MODE = 0

Bedeutung

Ersetzungs-unterprogramm für D-Funktion

Ersetzungs-unterprogramm für M-Funktion

Aufruf am Satzanfang

Werkzeugwechsel mit T-Funktion

Hauptprogramm

Programmierung	Kommentar
PROC MAIN	
...	;
N10 G01 F1000 X10 T1=5 D1	; T- und D-Funktion durch Aufruf von
	; "D_T_SUB_PROG" am Satzanfang ersetzt
...	;
N90 M30	

Unterprogramm "D_T_SUB_PROG"

Programmierung	Kommentar
N1000 PROC D_T_SUB_PROG DISPLOF SBLOF	

Programmierung	Kommentar
N4100 IF \$C_T_PROG==TRUE	; IF Adresse T programmiert
N4120 POS[B]=CAC(\$C_T)	; Teilungsposition anfahren
N4130 T[\$C_TE]=\$C_T	; Werkzeug anwählen (T-Anwahl)
N4140 ENDIF	; ENDIF
N4300 IF \$C_D_PROG==TRUE	; IF Adresse D programmiert
N4320 D=\$C_D	; Korrektur anwählen (D-Anwahl)
N4330 ENDIF	; ENDIF
N4400 IF \$C_DL_PROG==TRUE	; IF Adresse DL programmiert
N4420 D=\$C_DL	; Einsatzkorrektur anwählen
N4430 ENDIF	; ENDIF
N9999 RET	

2.20.2.6 Verhalten im Konfliktfall

Konfliktfall

Ein Konfliktfall liegt vor, wenn in einem Satz mehrere Funktionen programmiert sind und die Funktionen mit unterschiedlichen Unterprogrammen ersetzt werden sollen:

- Adresse D und DL ersetzen mit Unterprogramm:
MD11717 \$MN_FCT_CYCLE_NAME = "D_SUB_PROG"
- Adresse T ersetzen mit Unterprogramm:
MD10717 \$MN_FCT_CYCLE_NAME = "T_SUB_PROG"
- M-Funktion M6 ersetzen mit Unterprogramm:
MD10715 \$MN_M_NO_FCT_CYCLE[0] = 6
MD10716 \$MN_M_NO_FCT_CYCLE_NAME[0] = "M6_SUB_PROG"
MD10718 \$MN_M_NO_FCT_CYCLE_PAR = 0
MD22550 \$MC_TOOL_CHANGE_MODE = 1
MD22560 \$MC_TOOL_CHANGE_M_CODE = 6

Auflösung

Ein Konfliktfall wird entsprechend der nachfolgenden Tabelle aufgelöst:

In einer Programmzeile sind programmiert:			Aufgerufenes Unterprogramm:
D und/oder DL	T oder TCA	M6	
–	–	x	M6_SUB_PROG
–	x	–	T_SUB_PROG
–	x	x	M6_SUB_PROG
x	–	–	D_SUB_PROG
x	–	x	M6_SUB_PROG

In einer Programmzeile sind programmiert:			Aufgerufenes Unterprogramm:
D und/oder DL	T oder TCA	M6	
x	x	–	T_SUB_PROG
x	x	x	M6_SUB_PROG

2.20.3 Ersetzung von Spindel-Funktionen

2.20.3.1 Allgemeine Informationen

Funktion

Bei einer aktiven Kopplung können für Leitspindeln folgende Spindelfunktionen ersetzt werden:

- M40: Automatischer Getriebestufenwechsel
- M41 ... M45: Programmierter Getriebestufenwechsel
- SPOS, SPOSA und M19: Spindelpositionieren

Randbedingungen

- Für die Ersetzung einer Spindelfunktion müssen folgende Bedingungen erfüllt sein:
 - Die programmierte Spindel muss Leitspindel einer aktiven Kopplung sein.
 - Leit- und Folgespindel müssen sich im gleichen Kanal befinden. Dies wird nur erkannt, wenn sich die Leitspindel in dem Kanal befindet, in dem die Kopplung geschlossen wurde. Wird die Leitspindel in einen anderen Kanal getauscht, führt ein Getriebestufenwechsel bzw. ein Positionieren dieser Spindel nicht zum Aufruf des Ersetzungsunterprogramms.
 - Ein programmierter Getriebestufenwechsel muss einen realen Getriebestufenwechsel bewirken. Dazu müssen sich programmierte und aktive Getriebestufe unterscheiden.
- In einem Satz kann nur eine Spindelfunktion ersetzt werden. Mehrfachersetzungen führen zum Abbruch der Programmbearbeitung. Die Spindelfunktionen, die ersetzt werden sollen, müssen dann auf mehrere Sätze aufgeteilt werden.

Parametrierung

Spindelfunktion

Die Spindelfunktionen, die durch das Unterprogramm ersetzt werden sollen, werden ausgewählt im Maschinendatum:

MD30465 \$MA_AXIS_LANG_SUB_MASK

Bit	Bedeutung	
0	Getriebestufenwechsel automatisch (M40) und direkt (M41-M45)	
	Wert	Bedeutung
	0	Keine Ersetzung
1	Ersetzung durch das in MD15700 und MD15702 eingestellte Unterprogramm	
1	Spindelpositionieren mit SPOS / SPOSA / M19	
	Wert	Bedeutung
	0	Keine Ersetzung
1	Ersetzung durch das in MD15700 und MD15702 eingestellte Unterprogramm	

Unterprogramm: Name

Der Name des Ersetzungsunterprogramms wird eingetragen im Maschinendatum:

MD15700 \$MN_LANG_SUB_NAME = "<Unterprogrammname>"

Unterprogramm: Pfad

Der Pfad des Ersetzungsunterprogramms wird eingestellt im Maschinendatum:

MD15702 \$MN_LANG_SUB_PATH = <Wert>

Wert	Bedeutung
0	Hersteller-Zyklenverzeichnis: /_N_CMA_DIR
1	Anwender-Zyklenverzeichnis: /_N_CUS_DIR
2	Siemens-Zyklenverzeichnis: /_N_CST_DIR

Systemvariable: Aufrufzeitpunkt des Ersetzungsunterprogramms

Der Aufrufzeitpunkt des Ersetzungsunterprogramm kann über die Systemvariable \$P_SUB_STAT gelesen werden:

Wert	Bedeutung
0	Ersetzung nicht aktiv
1	Ersetzung aktiv, Unterprogrammaufruf am Satzanfang
2	Ersetzung aktiv, Unterprogrammaufruf am Satzende

Satzbearbeitung

Wird das Ersetzungsunterprogramm am Satzanfang aufgerufen, wird nach der Bearbeitung des Ersetzungsunterprogramms der Satz, der zum Aufruf geführt hat, abgearbeitet. Die ersetzten Befehle werden nicht mehr bearbeitet.

Wird das Ersetzungsunterprogramm am Satzende aufgerufen, wird zuerst der Satz, der zum Aufruf des Ersetzungsunterprogramms führt, ohne die zu ersetzenden Befehle abgearbeitet. Anschließend wird das Ersetzungsunterprogramm aufgerufen.

2.20.3.2 Ersetzung von M40 - M45 (Getriebestufenwechsel)

Funktion

Die Befehle zum Getriebestufenwechsel (M40, M41 ... M45) einer Leitspindel werden bei aktiver Kopplung durch den Aufruf eines anwenderspezifischen Unterprogramms ersetzt.

Parametrierung

Aktivierung

- MD30465 \$MA_AXIS_LANG_SUB_MASK, Bit 0 = 1

Aufrufzeitpunkt des Unterprogramms

- M40
Der Aufrufzeitpunkt kann nicht eingestellt werden. Das Ersetzungsunterprogramm wird immer am Satzanfang aufgerufen.
- M41 ... M45
Der Aufrufzeitpunkt ist abhängig vom projektierten Ausgabeverhalten der Hilfsfunktion an die PLC (siehe unten MD22080):
 - Ausgabe **vor** oder **während** der Bewegung: Unterprogrammaufruf am **Satzanfang**.
 - Ausgabe **nach** der Bewegung: Unterprogrammaufruf am **Satzende**

MD22080 \$MC_AUXFU_PREDEF_SPEC[12 ... 16] (Ausgabeverhalten für M41 ... M45)

Bit	Wert	Bedeutung
5	1	Ausgabe der Hilfsfunktion vor der Bewegung
6	1	Ausgabe der Hilfsfunktion während der Bewegung
7	1	Ausgabe der Hilfsfunktion nach der Bewegung

Systemvariable zu Informationsweitergabe

Dem Ersetzungsunterprogramm werden alle relevanten Informationen zu den im Satz programmierten Funktionen über Systemvariable (siehe Kapitel "Systemvariable (Seite 233)") zur Verfügung gestellt. Die Daten beziehen sich ausschließlich auf den Satz, in dem die zu ersetzende Funktion programmiert wurde.

2.20.3.3 Ersetzung von SPOS, SPOSA, M19 (Spindelpositionieren)

Funktion

Die Positionierbefehle (SPOS, SPOSA oder M19) einer Leitspindel werden bei aktiver Kopplung durch den Aufruf eines anwenderspezifischen Unterprogramms (Ersetzungsunterprogramm) ersetzt.

Anwendungsbeispiel

Bei der Parallelbearbeitung von Werkstücken an einer Doppelspindel-Maschine sind die Spindeln über einen Kopplungsfaktor ungleich 1 gekoppelt. Zum Werkzeugwechsel müssen sie auf dieselbe Position positioniert werden. Das Ersetzungsunterprogramm schaltet dazu die Kopplung aus, positioniert die Spindeln separat auf die Werkzeugwechselposition und schaltet anschließend die Kopplung wieder ein.

Parametrierung

Aktivierung

- MD30465 \$MA_AXIS_LANG_SUB_MASK, Bit 1 = 1

Aufrufzeitpunkt des Ersetzungsunterprogramms

- SPOS, SPOSA
Der Aufrufzeitpunkt kann nicht eingestellt werden. Das Ersetzungsunterprogramm wird immer am Satzanfang aufgerufen.
- M19
Der Aufrufzeitpunkt ist abhängig vom projektierten Ausgabeverhalten der Hilfsfunktion an die PLC (siehe unten MD22080):
 - Ausgabe **vor** oder **während** der Bewegung: Unterprogrammaufruf am **Satzanfang**.
 - Ausgabe **nach** der Bewegung: Unterprogrammaufruf am **Satzende**

MD22080 \$MC_AUXFU_PREDEF_SPEC[9]		
Bit	Wert	Bedeutung
5	1	Ausgabe der Hilfsfunktion vor der Bewegung
6	1	Ausgabe der Hilfsfunktion während der Bewegung
7	1	Ausgabe der Hilfsfunktion nach der Bewegung

Systemvariable zur Informationsweitergabe

Dem Ersetzungsunterprogramm werden alle relevanten Informationen zu den im Satz programmierten Funktionen über Systemvariable (siehe Kapitel "Systemvariable (Seite 233)") zur Verfügung gestellt. Die Daten beziehen sich ausschließlich auf den Satz, in dem die zu ersetzende Funktion programmiert wurde.

2.20.3.4 Systemvariable

Systemvariable	Bedeutung
\$P_SUB_AXFCT	TRUE, wenn M40, M41 ... M45 Ersetzung aktiv ist
\$P_SUB_GEAR	Programmierte bzw. errechnete Getriebestufe Außerhalb des Ersetzungsunterprogramms: Getriebestufe der Masterspindel

2.20 Ersetzung von Funktionen durch Unterprogramme

Systemvariable	Bedeutung
\$P_SUB_AUTOGEAR	TRUE, wenn im Satz der den Ersetzungsvorgang ausgelöst hat, M40 aktiv war. Außerhalb des Ersetzungsunterprogramms: aktuelle Einstellung im Interpreter
\$P_SUB_LA	Enthält den Achsnamen der Leitspindel der aktiven Kopplung, die den Ersetzungsvorgang ausgelöst hat. Hinweis Wird die Variable außerhalb des Ersetzungsunterprogramms verwendet, wird die Programmbearbeitung mit Alarm abgebrochen.
\$P_SUB_CA	Enthält den Achsnamen der Folgespindel der aktiven Kopplung, die den Ersetzungsvorgang ausgelöst hat. Hinweis Wird die Variable außerhalb des Ersetzungsunterprogramms aufgerufen, wird die Programmbearbeitung mit Alarm abgebrochen.
\$P_SUB_AXFCT	Enthält die aktiven Ersetzungsarten entsprechend MD30465 \$MA_AXIS_LANG_SUB_MASK
\$P_SUB_SPOS	TRUE, wenn die SPOS-Ersetzung aktiv ist
\$P_SUB_SPOSA	TRUE, wenn die SPOSA-Ersetzung aktiv ist
\$P_SUB_M19	TRUE, wenn die M19-Ersetzung aktiv ist
\$P_SUB_SPOSIT	Enthält die programmierte Spindelposition Hinweis Wird die Variable außerhalb des Ersetzungsunterprogramms aufgerufen, wird die Programmbearbeitung mit Alarm abgebrochen.
\$P_SUB_SPOSMODE	Enthält den Positionsanfahrmodus für die prog. Spindelposition:
	Wert Bedeutung
	0 keine Änderung des Positionsanfahrmodus
	1 AC
	2 IC
	3 DC
	4 ACP
	5 ACN
	6 OC
7 PC	
Hinweis: Wird die Variable außerhalb des Ersetzungsunterprogramms aufgerufen, wird die Programmbearbeitung mit Alarm abgebrochen.	
\$P_SUB_STAT	Satzbezogener Aufrufzeitpunkt des Ersetzungsunterprogramms

2.20.3.5 Beispiel: Getriebestufenwechsel

Im Unterprogramm werden alle Befehle zum Getriebestufenwechsel M40, M41 ... M45 ersetzt.

Parametrierung

Maschinendatum	Bedeutung
MD15700 \$MN_LANG_SUB_NAME = "LANG_SUB"	Unterprogramm

MD15702 \$MN_LANG_SUB_PATH = 0	Herstellerverzeichnis
MD22080 \$MC_AUXFU_PREDEF_SPEC[12] = 'H21'	M41: Ausgabe vor der Bewegung
MD22080 \$MC_AUXFU_PREDEF_SPEC[13] = 'H21'	M42: Ausgabe vor der Bewegung
MD22080 \$MC_AUXFU_PREDEF_SPEC[13] = 'H21'	M43: Ausgabe vor der Bewegung
MD22080 \$MC_AUXFU_PREDEF_SPEC[15] = 'H21'	M44: Ausgabe vor der Bewegung
MD22080 \$MC_AUXFU_PREDEF_SPEC[16] = 'H21'	M45: Ausgabe vor der Bewegung
MD30465 \$MA_AXIS_LANG_SUB_MASK[AX5] = 'H0001'	Getriebewechselbefehle ersetzen

Hauptprogramm

Programmierung	Kommentar
PROC MAIN	
N110 COUPON(S2,S1)	; Synchronspindelkopplung einschalten
N120 G01 F100 X100 S5000 M3 M43	; Unterprog.Aufruf aufgrund M43
N130 M40	; Automatischer Getriebestufenwechsel einschalten
N140 M3 S1000	; Unterprog.Aufruf aufgrund S1000 ; und dadurch ausgelöstem autom. ; Getriebestufenwechsel
N9999 M30	

Ersetzungsunterprogramm "LANG_SUB", Variante 1

Optimiert auf Einfachheit und Geschwindigkeit durch direkte Adressierung der Spindeln (S1: Leitspindel, S2: Folgespindel).

Programmierung	Kommentar
N1000 PROC LANG_SUB DISPLOF SBLOF	
N1100 IF(\$P_SUB_AXFCT ==1)	; Ersetzung wegen Getriebestufenwechsel
N1140 DELAYFSTON	; Beginn Stopp-Delay-Bereich
N1150 COUPOF(S2,S1)	; Synchronspindelkopplung ausschalten
N1160	;Getriebestufenwechsel getrennt für Leit- und Folgespindel
N1170 M1=\$P_SUB_GEAR M2=\$P_SUB_GEAR	
N1180 DELAYFSTON	; Ende Stopp-Delay-Bereich
N1190 COUPON(S2,S1)	; Synchronspindelkopplung einschalten
N1200 ENDIF	
...	
N9999 RET	

Ersetzungsunterprogramm "LANG_SUB", Variante 2

Flexibilität durch indirekte Adressierung über Systemvariable (Leitspindel: \$P_SUB_LA, Folgespindel: \$P_SUB_CA).

Programmierung	Kommentar
N1000 PROC LANG_SUB DISPLOF SBLOF	
N1010 DEF AXIS _LA	; Merker für Leitachse/-spindel
N1020 DEF AXIS _CA	; Merker für Folgeachse/-spindel
N1030 DEF INT _GEAR	; Merker für Getriebestufe
N1100 IF(\$P_SUB_AXFCT==1)	; Ersetzung wegen Getriebestufenwechsel
N1110 _GEAR=\$P_SUB_GEAR	; zu aktivierende Getriebestufe
N1120 _LA=\$P_SUB_LA	; Achsname der Leitspindel
N1130 _CA=\$P_SUB_CA	; Achsname der Folgespindel
N1140 DELAYFSTON	; Beginn Stopp-Delay-Bereich
N1150 COUPOF(_CA,_LA)	; Synchronspindelkopplung ausschalten
N1160 ;Getriebestufenwechsel für Leit- und Folgespindel	
N1170 M[AXTOSPI(_LA)]=_GEAR M[AXTOSPI(_CA)]=_GEAR	
N1180 DELAYFSTOF	; Ende Stopp-Delay-Bereich
N1190 COUPON(_CA,_LA)	; Synchronspindelkopplung einschalten
N1200 ENDIF	
...	
N9999 RET	

2.20.3.6 Beispiel: Spindelpositionieren

Im Unterprogramm ist nur die Ersetzung der Befehle SPOS und SPOSA explizit ausgeführt. Weitere Ersetzungen sind sinngemäß zu ergänzen.

Parametrierung

Maschinendatum

MD30465 \$MA_AXIS_LANG_SUB_MASK[AX5] = 'H0002'

MD22080 \$MC_AUXFU_PREDEF_SPEC[9] = 'H0021'

Bedeutung

Positionierbefehle ersetzen

Ausgabe von M19 an die PLC vor der Bewegung

Settingdatum

SD43240 \$SA_M19_SPOS[AX5] = 260

SD43250 \$SA_M19_SPOSMODE[AX5] = 4

Bedeutung

Spindelposition bei M19 = 260

Positionsanfahrmodus bei M19 : "Anfahren in positiver Richtung (ACP)"

Hauptprogramm

Programmierung	Kommentar
PROC MAIN	
...	
N210 COUPON(S2,S1)	; Synchronspindelkopplung aktivieren
N220 SPOS[1]=100	; Leitspindel mit SPOS positionieren
...	
N310 G01 F1000 X100 M19	; Leitspindel mit M19 positionieren

Ersetzungsunterprogramm "LANG_SUB", Variante 1

Optimiert auf Einfachheit und Geschwindigkeit durch direkte Adressierung der Spindeln (S1: Leitspindel, S2: Folgespindel).

Programmierung	Kommentar
N1000 PROC LANG_SUB DISPLOF SBLOF	
N2100 IF(\$P_SUB_AXFCT==2)	
N2110 ;Ersetzung von SPOS/SPOSA/M19 bei aktiver Synchronspindelkopplung	
N2185 DELAYFSTON	; Beginn Stopp-Delay-Bereich
N2190 COUPOF(S2,S1)	; Synchronspindelkopplung ausschalten
N2200	; Leit- und Folgespindel positionieren
N2210 IF(\$P_SUB_SPOS==TRUE) OR (\$P_SUB_SPOSA==TRUE)	
N2220 ;SPOS und SPOSA werden auf SPOS abgebildet	
N2230 CASE \$P_SUB_SPOSMODE OF \	
0 GOTOF LABEL1_DC \	
1 GOTOF LABEL1_IC \	
2 GOTOF LABEL1_AC \	
3 GOTOF LABEL1_DC \	
4 GOTOF LABEL1_ACP \	
5 GOTOF LABEL1_ACN \	
DEFAULT GOTOF LABEL_ERR	
LABEL1_DC: SPOS[1]=DC(\$P_SUB_SPOSIT) SPOS[2]=DC(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_IC: DELAYFSTOF	
SPOS[1]=IC(\$P_SUB_SPOSIT) SPOS[2]=IC(\$P_SUB_SPOSIT)	
DELAYFSTON	
GOTOF LABEL1_CONT	
LABEL1_AC: SPOS[1]=AC(\$P_SUB_SPOSIT) SPOS[2]=AC(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_ACP: SPOS[1]=ACP(\$P_SUB_SPOSIT) SPOS[2]=ACP(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_ACN: SPOS[1]=ACN(\$P_SUB_SPOSIT) SPOS[2]=ACN(\$P_SUB_SPOSIT)	
LABEL1_CONT:	
N2250 ELSE	; Spindel mit M19 positionieren
N2270 M1=19 M2=19	; Leit- und Folgespindel

2.20 Ersetzung von Funktionen durch Unterprogramme

Programmierung	Kommentar
N2280 ENDIF	; Ende Ersetzung SPOS, SPOSA
N2285 DELAYFSTOF	; Ende Stopp-Delay-Bereich
N2290 COUPON(S2,S1)	; Synchronspindelkopplung aktivieren
N2410 ELSE	
N2420 ;ab hier Bearbeitung weiterer Ersetzungen	
...	
N3300 ENDIF	; Ende Ersetzungen
...	
N9999 RET	; Normales Programmende
LABEL_ERR: SETAL(61000)	; Fehler aufgetreten

Ersetzungsunterprogramm "LANG_SUB", Variante 2

Flexibilität durch indirekte Adressierung über Systemvariable (Leitspindel: \$P_SUB_LA, Folgespindel: \$P_SUB_CA).

Programmierung	Kommentar
N1000 PROC LANG_SUB DISPLOF SBLOF	
N1010 DEF AXIS _LA	; Leitachse/-spindel
N1020 DEF AXIS _CA	; Folgeachse/-spindel
N1030 DEF INT _LSPI	; Leitspindelnummer (programmierte ; Spindel)
N1040 DEF INT _CSPI	; Folgespindelnummer
...	
N2100 IF(\$P_SUB_AXFCT==2)	
N2110 ;Ersetzung von SPOS/SPOSA/M19 bei aktiver	; Synchronspindelkopplung
N2120 _LA=\$P_SUB_LA	; Achsname der Leitspindel
N2130 _CA=\$P_SUB_CA	; Achsname der Folgespindel
N2140 _LSPI=AXTOSPI(_LA)	; Nummer der Leitspindel
N2180 _CSPI=AXTOSPI(_LA)	; Nummer der Folgespindel
N2185 DELAYFSTON	; Beginn Stopp-Delay-Bereich
N2190 COUPOF(_CA,_LA)	; Synchronspindelkopplung deaktivieren
N2200	; Leit- und Folgespindel positionieren:
N2210 IF(\$P_SUB_SPOS==TRUE) OR (\$P_SUB_SPOSA==TRUE)	
N2220 ;SPOS und SPOSA werden auf SPOS abgebildet	
N2230 CASE \$P_SUB_SPOSMODE OF	
0 GOTOF LABEL1_DC \	
1 GOTOF LABEL1_IC \	
2 GOTOF LABEL1_AC \	
3 GOTOF LABEL1_DC \	
4 GOTOF LABEL1_ACP \	
5 GOTOF LABEL1_ACN \	
DEFAULT GOTOF LABEL_ERR	
LABEL1_DC: SPOS[_LSPI]=DC(\$P_SUB_SPOSIT) SPOS[_CSPI]=DC(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	

Programmierung	Kommentar
LABEL1_IC: DELAYFSTOF	
SPOS[_LSPI]=IC(\$P_SUB_SPOSIT) SPOS[_CSPI]=IC(\$P_SUB_SPOSIT)	
DELAYFSTON	
GOTOF LABEL1_CONT	
LABEL1_AC: SPOS[_LSPI]=AC(\$P_SUB_SPOSIT) SPOS[_CSPI]=AC(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_ACP: SPOS[_LSPI]=ACP(\$P_SUB_SPOSIT) POS[_CSPI]=ACP(\$P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_ACN: SPOS[_LSPI]=ACN(\$P_SUB_SPOSIT) POS[_CSPI]=ACN(\$P_SUB_SPOSIT)	
LABEL1_CONT:	
N2250 ELSE	; Spindeln mit M19 positionieren
N2270 M[_LSPI]=19 M[_CSPI]=19	
N2280 ENDIF	
N2285 DELAYFSTOF	; Ende Stopp-Delay-Bereich
N2290 COUPON(_CA,_LA)	; Synchronspindelkopplung aktivieren
N2410 ELSE	
N2420	; ab hier Bearbeitung weiterer Ersetzungen
...	
N3300 ENDIF	
...	
N9999 RET	; Normales Programmende
LABEL_ERR: SETAL(61000)	; Fehler aufgetreten

2.20.4 Eigenschaften der Unterprogramme

Allgemeine Regeln

- Das bei der Ersetzung aufgerufene Unterprogramm kann den Befehl `PROC` und die Attribute `SBLOF` und `DISPLOF` enthalten.
- Die Ersetzung wird auch im ISO-Sprachmode ausgeführt. Die Ersetzungsunterprogramme werden aber ausschließlich im Standard-Sprachmode (Siemens) abgearbeitet. Dabei erfolgt eine implizite Umschaltung in den Standard-Sprachmode. Mit Rücksprung aus dem Ersetzungsunterprogramm wird wieder in den ursprünglichen Sprachmode zurückgeschaltet.
- Die Weitergabe von Informationen an das Ersetzungsunterprogramm erfolgt ausschließlich über Systemvariablen. Übergabeparameter sind nicht möglich.
- Das Verhalten bei Einzelsatz und Attribut `SBLOF` ist abhängig von der Einstellung in: `MD10702 IGNORE_SINGLEBLOCK_MASK`, Bit 14 (Einzelsatz-Stopp verhindern)

Wert	Bedeutung
0	<p>Das Ersetzungsunterprogramm verhält sich wie ein "normales" Unterprogramm:</p> <ul style="list-style-type: none"> • Rücksprung mit M17: Stopp am Ende des Unterprogramms <p>Hinweis Die Ausgabe der M-Funktion an die PLC ist abhängig von: MD20800 \$MC_SPF_END_TO_VDI, Bit 0 (Unterprogrammende an PLC)</p> <ul style="list-style-type: none"> - Bit 0 = 0: keine Ausgabe - Bit 0 = 1: M17 wird an die PLC ausgegeben. <ul style="list-style-type: none"> • Rücksprung mit RET: Kein Stopp am Ende des Ersetzungsunterprogramms
1	<p>Im Satz, in dem das Ersetzungsunterprogramm aufgerufen wird, wird nur einmal angehalten. Unabhängig davon ob:</p> <ul style="list-style-type: none"> • Das Unterprogramm am Satzanfang und/oder am Satzende aufgerufen wird • Im Unterprogramm weitere Unterprogramme aufgerufen werden • Das Unterprogramm mit M17 oder RET verlassen wird <p>Der Einzelsatz-Stopp erfolgt für die Ersetzung von M-Funktionen am Ende des Ersetzungsunterprogramms.</p> <p>Für die Ersetzung von T- und D/DL-Funktionen ist der Zeitpunkt des Einzelsatz-Stops abhängig vom Aufrufzeitpunkt des Unterprogramms:</p> <ul style="list-style-type: none"> • Aufruf am Satzanfang: Einzelsatz-Stopp am Ende des Satzes • Aufruf am Satzende: Einzelsatz-Stopp am Ende des Ersetzungsunterprogramms

- Bei Ersetzungsunterprogrammen mit dem Attribut DISPLOF wird in der Satzanzeige als aktueller Satz die Programmzeile angezeigt, die zum Aufruf des Unterprogramms geführt hat.
- Im Ersetzungsunterprogramm können mit den Befehlen DELAYFSTON und DELAYFSTOF Bereiche oder das gesamte Ersetzungsunterprogramm gegen Unterbrechungen, wie NC-Stop, Einlesesperre usw., geschützt werden.
- Ersetzungen erfolgen nicht rekursiv. D. h. die Funktion, die zum Aufruf des Ersetzungsunterprogramms geführt hat, wird bei nochmaliger Programmierung im Ersetzungsunterprogramm nicht mehr ersetzt.

Hilfsfunktionsausgabe an PLC

Bei Ersetzungen von Hilfsfunktionen bewirkt der Aufruf des Ersetzungsunterprogramms noch keine Ausgabe der Hilfsfunktion an die PLC. Die Ausgabe der Hilfsfunktion erfolgt erst wenn die Hilfsfunktion im Ersetzungsunterprogramm erneut programmiert wird.

Verhalten bei Satzsuchlauf

Das Ersetzungsunterprogramm wird auch in den Satzsuchlaufmodi "Satzsuchlauf mit Berechnung" und "Satzsuchlauf mit Berechnung im Modus Programmtest" (SERUPRO) aufgerufen. Eventuelle Besonderheiten müssen im Ersetzungsunterprogramm unter Verwendung der Systemvariable: \$P_SEARCH, \$AC_SERUPRO realisiert werden.

Bezüglich des Aufsammelns von Aktionen beim "Satzsuchlauf mit Berechnung" verhalten sich Ersetzungsunterprogramme wie "normale" Unterprogramme.

2.20.5 Randbedingungen

- Funktionsersetzungen sind unzulässig in:
 - Synchronaktionen
 - Technologiezyklen
- Einem Satz, der am Anfang zu ersetzende Funktionen enthält, dürfen keine satzweisen Synchronaktionen vorangestellt werden. Siehe unten Absatz "Beispiel zu: Satzweise Synchronaktionen".
- Im Ersetzungsunterprogramm dürfen nur die für die jeweiligen Ersetzungen notwendigen Aktionen durchgeführt werden.
- In einem Satz, in dem das Ersetzungsunterprogramm am Satzende aufgerufen wird, ist folgendes zu beachten:
 - Es darf kein modaler Unterprogrammaufruf aktiv sein
 - Es darf kein Unterprogrammrücksprung programmiert sein
 - Es darf kein Programmende programmiert sein

Hinweis

Die Steuerung überwacht nicht, ob die zu ersetzende Funktion im Ersetzungsunterprogramm realisiert wurde.

Beispiel zu: Satzweise Synchronaktionen

MD30465 \$MA_AXIS_LANG_SUB_MASK, Bit 0 = 1 (Getriebestufenwechsel)

Programmcode
...
N1000 WHENEVER \$AA_IM[X2] <= \$AA_IM[X1] + 0.5 DO \$AA_OVR[X1]=0
N1010 G1 X100 M43
...

Führt in Satz N1010 die Funktion M43 zum Aufruf eines Ersetzungsunterprogramms, wird die Bearbeitung mit abgebrochen und ein Alarm angezeigt.

2.21 Umbenennung/Sperrung von NC-Befehlen

Funktion

Mit der Funktion "Umbenennung/Sperrung von NC-Befehlen" kann der Maschinenhersteller oder Endanwender den Namen bestehender NC-Befehle ändern.

Anwendung

Die Funktion kann für folgende Zwecke genutzt werden:

- Verbesserung der Lesbarkeit von Teileprogrammen
- Sperrung von NC-Befehlen
- Anwenderspezifische Erweiterung von NC-Funktionen

Parametrierung

Die Umbenennung/Sperrung von NC-Befehlen erfolgt über das Maschinendatum:

MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[<n>] (Liste umprojektierter NC-Codes)

Ein gerader Index [<n>] enthält den ursprünglichen Namen eines Befehls. Der darauf folgende ungerade Index enthält den neuen Namen des Befehls.

Ein Leerstring ("") als neuer Name bedeutet, dass es keinen neuen Namen für den Befehl gibt. Der Befehl ist dadurch gesperrt und kann nicht mehr programmiert werden.

Änderungen von MD10712 werden mit Steuerungshochlauf wirksam.

Beispiel

Befehl G00 umbenennen und G01 sperren

MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[0] = "G00"

MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[1] = "EILGANG"

MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[2] = "G01"

MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[3] = ""

Nach Wirksamsetzen durch Power On kann Eilgang nicht mehr mit dem Befehl G00, sondern nur noch mit dem Befehl EILGANG programmiert werden. G01 kann nicht mehr programmiert werden.

2.22 Programmlaufzeit / Werkstückzähler

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.

2.22.1 Programmlaufzeit

2.22.1.1 Funktion

Die Funktion "Programmlaufzeit" stellt zur Überwachung technologischer Prozesse verschiedene Timer zur Verfügung, die über Systemvariablen im Teileprogramm und Synchronaktionen gelesen werden können. Es gibt zwei verschiedene Arten von Timern:

1. Standard-Timer
Standard-Timer sind immer aktiv.
2. Aktivierbare Timer
Aktivierbare Timer müssen über Maschinendaten aktiviert werden.

Standard-Timer

Zeit seit dem letzten Steuerungshochlauf

Systemvariable	Bedeutung
\$AN_SETUP_TIME	Zeit seit dem letzten Steuerungshochlauf mit Standardwerten in Minuten Wird bei jedem Steuerungshochlauf mit Standardwerten automatisch auf "0" zurückgesetzt.
\$AN_POWERON_TIME	Zeit seit dem letzten Normalhochlauf der Steuerung ("Warmstart") in Minuten Wird bei jedem Normalhochlauf der Steuerung automatisch auf "0" zurückgesetzt.

Programmlaufzeit

Die Timer zur Messung der Programmlaufzeiten sind nur in der Betriebsart AUTOMATIK verfügbar.

Systemvariable (kanalspezifisch)	Beschreibung
\$AC_ACT_PROG_NET_TIME	<p>Aktuelle Netto-Laufzeit des aktuellen Programms in Sekunden</p> <p>Netto-Laufzeit bedeutet, dass die Zeit, in der das Programm gestoppt war, abgezogen ist.</p> <p>Wird in der Betriebsart AUTOMATIK aus dem Kanalzustand RESET heraus ein Teileprogramm neu gestartet, wird \$AC_ACT_PROG_NET_TIME automatisch auf "0" zurückgesetzt.</p> <p>Weitere Eigenschaften:</p> <ul style="list-style-type: none"> • Die Reset-Taste setzt \$AC_ACT_PROG_NET_TIME nicht auf "0" zurück, sondern hält den Timer nur an. • Beim Start eines ASUPs wird \$AC_ACT_PROG_NET_TIME auf "0" gesetzt und zählt die Laufzeit des ASUPs mit. Am Ende eines ASUPs verhält es sich wie bei der RESET-Taste: der Timer wird nur angehalten, aber nicht auf "0" gesetzt. • Beim Start eines ereignisgesteuerten Programms (PROG_EVENT) wird \$AC_ACT_PROG_NET_TIME nicht zurückgesetzt. Die Programmlaufzeit wird nur weitergezählt, wenn es sich um ein Start-, M30- oder Suchlauf-PROG_EVENT handelt. • Das Verhalten von \$AC_ACT_PROG_NET_TIME bei GOTOS und Override = 0% ist über MD27850 parametrierbar (siehe Abschnitt "Parametrierung") <p>Hinweis: Mit \$AC_PROG_NET_TIME_TRIGGER kann \$AC_ACT_PROG_NET_TIME weiter manipuliert werden.</p>
\$AC_OLD_PROG_NET_TIME	<p>Netto-Laufzeit des gerade korrekt beendeten Programms in Sekunden</p> <p>"Korrekt beendet" bedeutet, dass das Programm nicht mit RESET abgebrochen, sondern regulär mit M30 beendet wurde.</p> <p>Wird ein neues Programm gestartet, bleibt \$AC_OLD_PROG_NET_TIME unangetastet, bis M30 erneut erreicht wird.</p> <p>Weitere Eigenschaften:</p> <ul style="list-style-type: none"> • \$AC_OLD_PROG_NET_TIME wird auf "0" gesetzt, wenn das aktuell angewählte Programm editiert wird. • Am Ende eines ASUPs oder eines ereignisgesteuerten Programms (PROG_EVENT) wird \$AC_OLD_PROG_NET_TIME nicht verändert. <p>Hinweis: Der implizite Kopiervorgang von \$AC_ACT_PROG_NET_TIME nach \$AC_OLD_PROG_NET_TIME findet nur statt, wenn \$AC_PROG_NET_TIME_TRIGGER nicht beschrieben wird.</p>

Systemvariable (kanalspezifisch)	Beschreibung												
\$AC_OLD_PROG_NET_TIME_COUNT	<p>Änderungen auf \$AC_OLD_PROG_NET_TIME</p> <p>Nach POWER ON steht \$AC_OLD_PROG_NET_TIME_COUNT auf "0".</p> <p>\$AC_OLD_PROG_NET_TIME_COUNT wird immer dann erhöht, wenn die Steuerung \$AC_OLD_PROG_NET_TIME neu geschrieben hat.</p> <p>Wenn der Anwender das laufende Programm mit RESET abbricht, bleiben \$AC_OLD_PROG_NET_TIME und \$AC_OLD_PROG_NET_TIME_COUNT unverändert.</p> <p>Mit \$AC_OLD_PROG_NET_TIME_COUNT lässt sich daher feststellen, ob \$AC_OLD_PROG_NET_TIME geschrieben wurde.</p> <p>Beispiel:</p> <p>Wenn zwei hintereinander laufende Programme die identische Laufzeit haben und korrekt beendet wurden, dann kann der Anwender dies über den veränderten Wert in \$AC_OLD_PROG_NET_TIME_COUNT erkennen.</p>												
\$AC_PROG_NET_TIME_TRIGGER	<p>Trigger zur selektiven Messung von Programmabschnitten.</p> <p>Durch das Schreiben des Wertes in die Variable wird die Funktion ausgelöst:</p> <table border="1"> <thead> <tr> <th>Wert</th> <th>Funktion</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Inaktiv</td> </tr> <tr> <td>1</td> <td>Beendet die Messung <ul style="list-style-type: none"> • \$AC_OLD_PROG_NET_TIME = \$AC_ACT_PROG_NET_TIME • \$AC_ACT_PROG_NET_TIME = 0 und läuft anschließend weiter </td> </tr> <tr> <td>2</td> <td>Startet die Messung <ul style="list-style-type: none"> • \$AC_ACT_PROG_NET_TIME = 0 • \$AC_OLD_PROG_NET_TIME wird nicht verändert </td> </tr> <tr> <td>3</td> <td>Stoppt die Messung <ul style="list-style-type: none"> • \$AC_OLD_PROG_NET_TIME wird nicht verändert • \$AC_ACT_PROG_NET_TIME wird nicht verändert </td> </tr> <tr> <td>4</td> <td>Fortsetzen der gestoppten Messung <ul style="list-style-type: none"> • \$AC_ACT_PROG_NET_TIME läuft weiter • \$AC_OLD_PROG_NET_TIME wird nicht verändert </td> </tr> </tbody> </table>	Wert	Funktion	0	Inaktiv	1	Beendet die Messung <ul style="list-style-type: none"> • \$AC_OLD_PROG_NET_TIME = \$AC_ACT_PROG_NET_TIME • \$AC_ACT_PROG_NET_TIME = 0 und läuft anschließend weiter 	2	Startet die Messung <ul style="list-style-type: none"> • \$AC_ACT_PROG_NET_TIME = 0 • \$AC_OLD_PROG_NET_TIME wird nicht verändert 	3	Stoppt die Messung <ul style="list-style-type: none"> • \$AC_OLD_PROG_NET_TIME wird nicht verändert • \$AC_ACT_PROG_NET_TIME wird nicht verändert 	4	Fortsetzen der gestoppten Messung <ul style="list-style-type: none"> • \$AC_ACT_PROG_NET_TIME läuft weiter • \$AC_OLD_PROG_NET_TIME wird nicht verändert
Wert	Funktion												
0	Inaktiv												
1	Beendet die Messung <ul style="list-style-type: none"> • \$AC_OLD_PROG_NET_TIME = \$AC_ACT_PROG_NET_TIME • \$AC_ACT_PROG_NET_TIME = 0 und läuft anschließend weiter 												
2	Startet die Messung <ul style="list-style-type: none"> • \$AC_ACT_PROG_NET_TIME = 0 • \$AC_OLD_PROG_NET_TIME wird nicht verändert 												
3	Stoppt die Messung <ul style="list-style-type: none"> • \$AC_OLD_PROG_NET_TIME wird nicht verändert • \$AC_ACT_PROG_NET_TIME wird nicht verändert 												
4	Fortsetzen der gestoppten Messung <ul style="list-style-type: none"> • \$AC_ACT_PROG_NET_TIME läuft weiter • \$AC_OLD_PROG_NET_TIME wird nicht verändert 												
Durch POWER ON werden alle Systemvariablen auf "0" zurückgesetzt!													

Hinweis**Restzeit für ein Werkstück**

Wenn nacheinander gleiche Werkstücke produziert werden, kann mit den folgenden Timerwerten die verbleibende Restzeit für ein Werkstück ermittelt werden:

- Bearbeitungszeit für das zuletzt produzierte Werkstück (siehe \$AC_OLD_PROG_NET_TIME)
- Aktuelle Bearbeitungszeit (siehe \$AC_ACT_PROG_NET_TIME)

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.

Aktivierbare Timer

Programmlaufzeit

Die Timer zur Messung der Programmlaufzeiten sind nur in der Betriebsart AUTOMATIK verfügbar.

Systemvariable (kanalspezifisch)	Beschreibung
\$AC_OPERATING_TIME	Gesamtlaufzeit von NC-Programmen in der Betriebsart Automatik in Sekunden Aufsummiert werden in der Betriebsart Automatik die Laufzeiten aller Programme zwischen NC-Start und Programmende / NC-Reset. Bei NC-Stop und Override = 0% wird standardmäßig nicht gezählt. Das Weiterzählen bei Override 0% kann über MD27860 aktiviert werden. Der Wert wird mit jedem Steuerungshochlauf automatisch auf "0" zurückgesetzt.
\$AC_CYCLE_TIME	Laufzeit des angewählten NC-Programms in Sekunden Im angewählten NC-Programm wird die Laufzeit zwischen NC-Start und Programmende / NC-Reset gemessen. Bei NC-Stop und Override = 0% wird standardmäßig nicht gezählt. Das Weiterzählen bei Override 0% kann über MD27860 aktiviert werden. Der Wert wird mit dem Start eines neuen NC-Programms automatisch auf "0" zurückgesetzt. Über MD27860 ist einstellbar, ob diese Löschung auch bei einem Sprung auf den Programmmanfang mit GOTOS oder bei Start von ASUPs und PROG_EVENTS erfolgen soll.
\$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. Über MD27860 ist einstellbar, ob nur mit aktivem Werkzeug oder werkzeugunabhängig gemessen werden soll. Die Messung wird zusätzlich bei aktiver Verweilzeit unterbrochen. Der Wert wird bei jedem Steuerungshochlauf mit Standardwerten automatisch auf "0" zurückgesetzt.

2.22.1.2 Inbetriebnahme

Aktivierung / Deaktivierung

Das Ein-/Ausschalten der aktivierbaren Timer erfolgt über das Maschinendatum:

MD27860 \$MC_PROCESSTIMER_MODE, Bit 0 - 2 = <Wert>

Bit	Wert	Bedeutung
0	0	Timer für \$AC_OPERATING_TIME nicht aktiv.
	1	Timer für \$AC_OPERATING_TIME aktiv.
1	0	Timer für \$AC_CYCLE_TIME nicht aktiv.
	1	Timer für \$AC_CYCLE_TIME aktiv.
2	0	Timer für \$AC_CUTTING_TIME nicht aktiv.
	1	Timer für \$AC_CUTTING_TIME aktiv.

Parametrierung

Verhalten der immer aktiven Timer

Das Verhalten der immer aktiven Timer bei GOTOS und Override = 0% wird eingestellt mit dem Maschinendatum:

MD27850 \$MC_PROG_NET_TIMER_MODE

Bit	Wert	Bedeutung
0	0	\$AC_ACT_PROG_NET_TIME wird bei einem Sprung mit GOTOS auf den Programm-anfang nicht auf "0" zurückgesetzt (Grundeinstellung).
	1	\$AC_ACT_PROG_NET_TIME wird bei einem Sprung mit GOTOS auf den Programm-anfang auf "0" zurückgesetzt, der Wert vorher in \$AC_OLD_PROG_NET_TIME gespeichert und der Programmzähler \$AC_OLD_PROG_NET_TIME_COUNT erhöht.
1	0	\$AC_ACT_PROG_NET_TIME wird bei Override = 0% nicht erhöht. D. h. die Programmlaufzeit wird ohne die Zeit gemessen, für die der Override auf "0" eingestellt wurde (Grundeinstellung).
	1	\$AC_ACT_PROG_NET_TIME wird auch bei Override = 0% erhöht. D. h. die Programmlaufzeit wird mit der Zeit gemessen, für die der Override auf "0" eingestellt wurde.

Verhalten der aktivierbaren Timer

Das Verhalten der aktivierbaren Timer bei bestimmten Funktionen (z. B. Probelaufvorschub, Programmtest) wird eingestellt mit dem Maschinendatum:

MD27860 \$MC_PROCESSTIMER_MODE

Bit	Wert	Bedeutung
4	0	Keine Messung bei aktivem Probelaufvorschub.
	1	Messung auch bei aktivem Probelaufvorschub.
5	0	Keine Messung bei Programm-Test.
	1	Messung auch bei Programm-Test.
6	Nur bei Bit 1 = 1 (Timer für \$AC_CYCLE_TIME ist aktiv)	
	0	\$AC_CYCLE_TIME wird auch bei Start durch ASUP und PROG_EVENTS auf "0" zurückgesetzt.
	1	\$AC_CYCLE_TIME wird bei Start durch ASUP und PROG_EVENTS nicht auf "0" zurückgesetzt.

Bit	Wert	Bedeutung
7	Nur bei Bit 2 = 1 (Timer für \$AC_CUTTING_TIME ist aktiv)	
	0	Timer für \$AC_CUTTING_TIME zählt nur bei aktivem Werkzeug.
	1	Timer für \$AC_CUTTING_TIME zählt werkzeuginabhängig.
8	Nur bei Bit 1 = 1 (Timer für \$AC_CYCLE_TIME ist aktiv)	
	0	\$AC_CYCLE_TIME wird bei einem Sprung mit GOTOS auf den Programmanfang nicht auf "0" zurückgesetzt.
	1	\$AC_CYCLE_TIME wird bei einem Sprung mit GOTOS auf den Programmanfang auf "0" zurückgesetzt.
9	Nur bei Bit 0, 1 = 1 (Timer für \$AC_OPERATING_TIME und \$AC_CYCLE_TIME sind aktiv)	
	0	Kein Weiterzählen der Programmlaufzeit bei Override = 0%
	1	Weiterzählen der Programmlaufzeit bei Override = 0%

2.22.1.3 Randbedingungen

Satzsuchlauf

Bei Satzsuchlauf werden keine Programmlaufzeiten ermittelt.

REPOS

Die Zeitdauer eines REPOS-Vorgangs wird der aktuellen Bearbeitungszeit (\$AC_ACT_PROG_NET_TIME) angerechnet.

2.22.1.4 Beispiele

Beispiel 1: Parametrierung der Laufzeitmessung über MD27860

Aktivierung der Laufzeitmessung für das aktive NC-Programm, dabei kein Messen bei aktivem Probelaufvorschub und Programmtest:

MD27860 \$MC_PROCESSTIMER_MODE = 'H2'

Aktivierung der Messung für die Werkzeug-Eingriffszeit, dabei auch Messen bei aktivem Probelaufvorschub und Programmtest:

MD27860 \$MC_PROCESSTIMER_MODE = 'H34'

Aktivierung der Messung für die Gesamtlaufzeit und die Bearbeitungszeit mit aktivem Werkzeug, dabei auch Messen bei Programmtest:

MD27860 \$MC_PROCESSTIMER_MODE = 'H25'

Aktivierung der Messung für die Gesamtlaufzeit und die Bearbeitungszeit (werkzeuginabhängig), dabei auch Messen bei Programmtest:

MD27860 \$MC_PROCESSTIMER_MODE = 'Ha5'

Aktivierung der Messung für die Bearbeitungszeit mit aktivem Werkzeug, dabei auch Messen bei Override = 0% , aber nicht bei aktivem Probelauf-Vorschub:


```
MD27860 $MC_PROCESSTIMER_MODE = 'H2'
```

Beispiel 2: 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 3: 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
```

2.22.2 Werkstückzähler

2.22.2.1 Funktion

Mit der Funktion "Werkstückzähler" stehen in Form von kanalspezifische Systemvariablen diverse Zähler mit einem Wertebereich von 0 bis 999.999.999 zur Verfügung. Auf die Systemvariablen kann lesend und schreibend zugegriffen werden.

Über die nachfolgend beschriebenen kanalspezifischen Maschinendaten kann auf die Zähler-Aktivierung, den Zeitpunkt der Nullung und den Zählalgorithmus Einfluss genommen werden.

Systemvariablen zur Werkstückzählung

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. Über MD27880 kann die Generierung des Anzeige-Alarms: "Kanal %1 Werkstueck-Soll = %2 erreicht" und des kanalspezifischen NC/PLC-Nahtstellensignals: DB21, ... DBX317.1 (Werkstück-Soll erreicht) aktiviert werden.
\$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 den Wert "0" gesetzt und können, unabhängig von ihrer Aktivierung, gelesen und geschrieben werden.

2.22.2.2 Inbetriebnahme

Aktivierung

Die Werkstückzähler werden aktiviert mit dem Maschinendatum:

MD27880 \$MC_PART_COUNTER (Aktivierung der Werkstückzähler)

Bit	Wert	Bedeutung
0	1	\$AC_REQUIRED_PARTS ist aktiv.
1	0	Alarm- / Signalausgabe bei: \$AC_ACTUAL_PARTS = \$AC_REQUIRED_PARTS
	1	Alarm- / Signalausgabe bei: \$AC_SPECIAL_PARTS = \$AC_REQUIRED_PARTS
4	1	\$AC_TOTAL_PARTS ist aktiv.

Bit	Wert	Bedeutung
5	0	\$AC_TOTAL_PARTS wird durch M02 / M30 um den Wert "1" erhöht.
	1	\$AC_TOTAL_PARTS wird durch den mit MD27882[0] definierten M-Befehl um den Wert "1" erhöht.
6	0	\$AC_TOTAL_PARTS ist auch bei Programm-Test / Satzsuchlauf aktiv.
7	1	\$AC_TOTAL_PARTS wird bei einem Rücksprung mit GOTOS um den Wert "1" erhöht.
8	1	\$AC_ACTUAL_PARTS ist aktiv.
9	0	\$AC_ACTUAL_PARTS wird durch M02 / M30 um den Wert "1" erhöht.
	1	\$AC_ACTUAL_PARTS wird durch den mit MD27882[1] definierten M-Befehl um den Wert "1" erhöht.
10	0	\$AC_ACTUAL_PARTS ist auch bei Programm-Test / Satzsuchlauf aktiv.
11	1	\$AC_ACTUAL_PARTS wird bei einem Rücksprung mit GOTOS um den Wert "1" erhöht.
12	1	\$AC_SPECIAL_PARTS ist aktiv.
13	0	\$AC_SPECIAL_PARTS wird durch M02 / M30 um den Wert "1" erhöht.
	1	\$AC_SPECIAL_PARTS wird durch den mit MD27882[2] definierten M-Befehl um den Wert "1" erhöht.
14	0	\$AC_SPECIAL_PARTS ist auch bei Programm-Test / Satzsuchlauf aktiv.
15	1	\$AC_SPECIAL_PARTS wird bei einem Rücksprung mit GOTOS um den Wert "1" erhöht.

Parametrierung

Werkstückzählung mit anwenderdefiniertem M-Befehl

Ist das entsprechende Bit in MD27880 gesetzt, wird der Zählimpuls statt über das Programmende M2/M30 über einen über folgendes Maschinendatum parametrierbaren M-Befehl ausgelöst:

MD27882 \$MC_PART_COUNTER_MCODE[<n>] (Werkstückzählung mit anwenderdefiniertem M-Befehl)

<n>	Bedeutung
0	MD27882[0] definiert den M-Befehl, bei dem \$AC_TOTAL_PARTS hochgezählt wird.
1	MD27882[1] definiert den M-Befehl, bei dem \$AC_ACTUAL_PARTS hochgezählt wird.
2	MD27882[2] definiert den M-Befehl, bei dem \$AC_SPECIAL_PARTS hochgezählt wird.

Bei Aufruf eines anwenderdefinierten M-Befehls wird der betreffende Werkstückzähler um den Wert "1" erhöht.

Schutzstufe für Werkstückzählung

Mit dem folgenden Maschinendatum wird die Schutzstufe für das Ein-/Ausschalten der Werkstückzählung über die Bedienoberfläche eingestellt:

MD51074 \$MN_ACCESS_WRITE_WPC_COUNTER = <Schutzstufe>

2.22.2.3 Randbedingungen

Betriebsartenwechsel / NC-RESET

Durch Betriebsartenwechsel oder NC-RESET werden die Werkstückzähler nicht beeinflusst.

\$AC_REQUIRED_PARTS ≤ 0

Bei $\$AC_REQUIRED_PARTS \leq 0$ und MD27880 $\$MC_PART_COUNTER.Bit\ 0 == 1$ wird für alle aktiven Zähler die Zählung und der mit MD27880 eingestellte Identitätsvergleich **nicht** durchgeführt.

2.22.2.4 Beispiele

Aktivierung des Werkstückzählers \$AC_REQUIRED_PARTS:

- MD27880 $\$MC_PART_COUNTER = 'H3'$

$\$AC_REQUIRED_PARTS$ ist aktiv.

Anzeige-Alarm bei: $\$AC_REQUIRED_PARTS == \$AC_SPECIAL_PARTS$

Aktivierung des Werkstückzählers \$AC_TOTAL_PARTS:

- MD27880 $\$MC_PART_COUNTER = 'H10'$
- MD27882 $\$MC_PART_COUNTER_MCODE[0] = 80$

$\$AC_TOTAL_PARTS$ ist aktiv, mit jedem M02 wird der Zähler um den Wert "1" erhöht.

$\$MC_PART_COUNTER_MCODE[0]$ hat keine Bedeutung.

Aktivierung des Werkstückzählers \$AC_ACTUAL_PARTS:

- MD27880 $\$MC_PART_COUNTER = 'H300'$
- MD27882 $\$MC_PART_COUNTER_MCODE[1] = 17$

$\$AC_TOTAL_PARTS$ ist aktiv, mit jedem M17 wird der Zähler um den Wert "1" erhöht.

Aktivierung des Werkstückzählers \$AC_SPECIAL_PARTS:

- MD27880 $\$MC_PART_COUNTER = 'H3000'$
- MD27882 $\$MC_PART_COUNTER_MCODE[2] = 77$

$\$AC_SPECIAL_PARTS$ ist aktiv.

Mit jedem M77 erfolgt: $\$AC_SPECIAL_PARTS + 1$

Abschalten des Werkstückzählers \$AC_ACTUAL_PARTS:

- MD27880 $\$MC_PART_COUNTER = 'H200'$
- MD27882 $\$MC_PART_COUNTER_MCODE[1] = 50$

$\$AC_ACTUAL_PARTS$ ist nicht aktiv.

Aktivierung aller Zähler:

- MD27880 \$MC_PART_COUNTER = 'H3313'
- MD27882 \$MC_PART_COUNTER_MCODE[0] = 80
- MD27882 \$MC_PART_COUNTER_MCODE[1] = 17
- MD27882 \$MC_PART_COUNTER_MCODE[2] = 77

\$AC_REQUIRED_PARTS ist aktiv.

Anzeige-Alarm bei: \$AC_REQUIRED_PARTS == \$AC_SPECIAL_PARTS

\$AC_TOTAL_PARTS ist aktiv, mit jedem M02 wird der Zähler um den Wert "1" erhöht.

\$MC_PART_COUNTER_MCODE[0] hat keine Bedeutung.

\$AC_ACTUAL_PARTS ist aktiv, mit jedem M17 wird der Zähler um den Wert "1" erhöht.

\$AC_SPECIAL_PARTS ist aktiv, mit jedem M77 wird der Zähler um den Wert "1" erhöht.

Keine Bearbeitung des Werkstückzählers \$AC_ACTUAL_PARTS bei Programm-Test / Satzsuchlauf:

- MD27880 \$MC_PART_COUNTER = 'H700'
- MD27882 \$MC_PART_COUNTER_MCODE[1] = 75

\$AC_ACTUAL_PARTS ist aktiv, mit jedem M75 wird der Zähler um den Wert "1" erhöht, außer bei Programm-Test und Satzsuchlauf.

Aufhebung der im MD27880 \$MC_PART_COUNTER getroffenen Zählmodi bei Bit 0 = 1:

- MD27882 \$MC_PART_COUNTER_MCODE[0] = 41
- MD27882 \$MC_PART_COUNTER_MCODE[1] = 42
- MD27882 \$MC_PART_COUNTER_MCODE[2] = 43

Programmcode	Kommentar
...	
N100 \$AC_REQUIRED_PARTS=-10	; Wert < 0: Zählung einstellen.
N200 M41 M43	; Keine Zählung.
N300 M42	
...	
N500 \$AC_REQUIRED_PARTS=52	; Wert > 0: Zählung entsprechend MD27880 aktiv.
N501 M43	; Zählung.
N502 M42 M41	; Zählung.
...	

2.23 Datenlisten

2.23.1 Funktion

Mit der Funktion "Werkstückzähler" stehen in Form von kanalspezifische Systemvariablen diverse Zähler mit einem Wertebereich von 0 bis 999.999.999 zur Verfügung. Auf die Systemvariablen kann lesend und schreibend zugegriffen werden.

Über die nachfolgend beschriebenen kanalspezifischen Maschinendaten kann auf die Zähler-Aktivierung, den Zeitpunkt der Nullung und den Zählalgorithmus Einfluss genommen werden.

Systemvariablen zur Werkstückzählung

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. Über MD27880 kann die Generierung des Anzeige-Alarms: "Kanal %1 Werkstueck-Soll = %2 erreicht" und des kanalspezifischen NC/PLC-Nahtstellensignals: DB21, ... DBX317.1 (Werkstück-Soll erreicht) aktiviert werden.
\$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 den Wert "0" gesetzt und können, unabhängig von ihrer Aktivierung, gelesen und geschrieben werden.

2.23.2 Maschinendaten

2.23.2.1 Allgemeine Maschinendaten

Anzeige-Maschinendaten

Nummer	Bezeichner: \$MM_	Beschreibung
SINUMERIK Operate		
9421	MA_AXES_SHOW_GEO_FIRST	Geo-Achsen des Kanals zu erst anzeigen
9422	MA_PRESET_MODE	PRESET / Basisverschiebung in JOG
9423	MA_MAX_SKP_LEVEL	Maximale Anzahl von Ausblendebenen

NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10010	ASSIGN_CHAN_TO_MODE_GROUP	Kanal gültig in BAG
10125	EES_NC_NAME	NCU-Name für die Generierung eindeutiger NC-Programmnamen im EES-Betrieb
10280	PROG_FUNCTION_MASK	Vergleichsbefehle ">" und "<"
10700	PREPROCESSING_LEVEL	Programmvorverarbeitungsstufe
10702	IGNORE_SINGLEBLOCK_MASK	Einzelatz-Stopp verhindern
10707	PROG_TEST_MASK	Programmtest Modi
10708	SERUPRO_MASK	Satzsuchlauf Modi
10710	PROG_SD_RESET_SAVE_TAB	Zu aktualisierende Settingdaten
10711	NC_LANGUAGE_CONFIGURATION	Art und Weise, wie mit Sprachbefehlen verfahren wird, deren zugehörige Option bzw. Funktion nicht aktiviert ist.
10713	M_NO_FCT_STOPRE	M-Funktion mit Vorlaufstopp
10715	M_NO_FCT_CYCLE	Durch UP zu ersetzende M-Funktion
10716	M_NO_FCT_CYCLE_NAME	UP-Name für M-Funktions-Ersetzung
10717	T_NO_FCT_CYCLE_NAME	Name des Werkzeugwechselzyklus für T-Funktionsersetzung
10718	M_NO_FCT_CYCLE_PAR	M-Funktionsersetzung mit Parametern
10719	T_NO_FCT_CYCLE_MODE	Parametrierung der T-Funktionsersetzung
11450	SEARCH_RUN_MODE	Suchlauf Parametrierung
11470	REPOS_MODE_MASK	Repositioniereigenschaften
11600	BAG_MASK	BAG-Verhalten bezüglich ASUP
11602	ASUP_START_MASK	Stoppgründe für ASUP ignorieren
11604	ASUP_START_PRIO_LEVEL	Prioritäten, ab denen \$MN_ASUP_START_MASK wirksam ist
11610	ASUP_EDITABLE	Aktivierung eines anwenderspezifischen ASUP Programmes

2.23 Datenlisten

Nummer	Bezeichner: \$MN_	Beschreibung
11612	ASUP_EDIT_PROTECTION_LEVEL	Schutzstufe des anwenderspezifischen ASUP Programms
11620	PROG_EVENT_NAME	Programmname für PROG_EVENT
11625	FILE_ONLY_WITH_EXTENSION	Beim Programmaufruf nur nach Dateien mit einer Dateikennung suchen
11626	CYCLES_ONLY_IN_CYCDIR	Programme mit Interface nur in den Zyklenverzeichnissen suchen
11717	D_NO_FCT_CYCLE_NAME	Unterprogrammname für D-Funktionsersetzung
15700	LANG_SUB_NAME	Name für Substitutionsunterprogramm
15702	LANG_SUB_PATH	Aufrufpfad für Substitutionsunterprogramm
17200	GMMC_INFO_NO_UNIT	globale HMI Info (ohne physikalische Einheit)
17201	GMMC_INFO_NO_UNIT_STATUS	globale HMI Statusinfo (ohne physik. Einheit)
18045	EES_MODE_INFO	Modus, in dem die Funktion EES arbeitet
18360	MM_EXT_PROG_BUFFER_SIZE	FIFO-Buffer Größe für Abarbeiten von Extern (DRAM)
18362	MM_EXT_PROG_NUM	Anzahl der gleichzeitig von 'Extern abarbeitbaren Programmebenen

2.23.2.2 Kanal-spezifische Maschinendaten

Grundmaschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20000	CHAN_NAME	Kanalname
20050	AXCONF_GEOAX_ASSIGN_TAB	Zuordnung Geometrieachse zu Kanalachse
20060	AXCONF_GEOAX_NAME_TAB	Geometrieachsname im Kanal
20070	AXCONF_MACHAX_USED	Maschinenachsnummer gültig im Kanal
20080	AXCONF_CHANAX_NAME_TAB	Kanalachsname im Kanal [Kanalachsnr.]: 0...7
20090	SPIND_DEF_MASTER_SPIND	Löschstellung der Masterspindel im Kanal
20100	DIAMETER_AX_DEF	Geometrieachse mit Planachsfunktion
20106	PROG_EVENT_IGN_SINGLEBLOCK	Prog-Events ignorieren den Einzelsatz
20107	PROG_EVENT_IGN_INHIBIT	Prog-Events ignorieren die Einlesesperre
20108	PROG_EVENT_MASK	Ereignisgesteuerte Prorammaufrufe
20109	PROG_EVENT_MASK_PROPERTIES	Eigenschaften Prog-Events
20114	MODESWITCH_MASK	Einstellung für Repos
20116	IGNORE_INHIBIT_ASUP	Anwender-ASUPs trotz Einlesesperre komplett abarbeiten
20117	IGNORE_SINGLEBLOCK_ASUP	Anwender-ASUPs trotz Einzelsatzbearbeitung komplett abarbeiten
20160	CUBIC_SPLINE_BLOCKS	Anzahl der Sätze beim C-Spline
20170	COMPRESS_BLOCK_PATH_LIMIT	Maximale Verfahrenlänge eines NC-Satzes bei Kompression

Nummer	Bezeichner: \$MC_	Beschreibung
20191	IGN_PROG_STATE_ASUP	Ausführung der Interruptroutine auf der Bedienoberfläche nicht anzeigen
20192	PROG_EVENT_IGN_PROG_STATE	Ausführung des Prog-Events auf der Bedienoberfläche nicht anzeigen
20193	PROG_EVENT_IGN_STOP	Prog-Events ignorieren die Stopp-Taste
20194	IGNORE_NONCSTART_ASUP	ASUP-Start trotz anstehender Alarmreaktion "Verriegelung NC-Start" bei bestimmten Anwenderalarmen zulässig
20210	CUTCOM_CORNER_LIMIT	Maximalwinkel für Ausgleichssätze bei WRK
20220	CUTCOM_MAX_DISC	Maximaler Wert für DISC
20230	CUTCOM_CURVE_INSERT_LIMIT	Maximalwinkel für Schnittpunktberechnung bei WRK
20240	CUTCOM_MAXNUM_CHECK_BLOCKS	Sätze für vorausschauende Konturberechnung bei WRK
20250	CUTCOM_MAXNUM_DUMMY_BLOCKS	Maximale Satzanzahl ohne Verfahrbewegung bei WRK
20270	CUTTING_EDGE_DEFAULT	Grundstellung der Werkzeugschneide ohne Programmierung
20400	LOOKAH_USE_VELO_NEXT_BLOCK	Lookahead auf programmierte Folgesatzgeschwindigkeit
20430	LOOKAH_NUM_OVR_POINTS	Anzahl der Override-Eckwerte bei Lookahead
20440	LOOKAH_OVR_POINTS	Override-Schalter-Eckwerte bei Lookahead
20500	CONST_VELO_MIN_TIME	Minimale Zeit mit konstanter Geschwindigkeit
20600	MAX_PATH_JERK	Bahnbezogener Maximalruck
20610	ADD_MOVE_ACCEL_RESERVE	Beschleunigungsreserve für überlagerte Bewegungen
20700	REFP_NC_START_LOCK	NC-Startsperre ohne Referenzpunkt
20750	ALLOW_GO_IN_G96	G0-Logik bei G96, G961
20800	SPF_END_TO_VDI	Unterprogrammende an PLC
21000	CIRCLE_ERROR_CONST	Kreisendpunktüberwachung Konstante
21010	CIRCLE_ERROR_FACTOR	Kreisendpunktüberwachung Faktor
21100	ORIENTATION_IS_EULER	Winkeldefinition bei Orientierungsprogrammierung
21110	X_AXIS_IN_OLD_X_Z_PLANE	Koordinatensystem bei automatischer Framedefinition
21200	LIFTFAST_DIST	Verfahrstrecke bei Schnellabheben von der Kontur
21210	SETINT_ASSIGN_FASTIN	HW-Zuordnung des ext. NC-Eingangsbytes für NC-Programm-Interrupt
21202	LIFTFAST_WITH_MIRROR	Schnellabheben mit Spiegeln

Satzsuchlauf

Nummer	Bezeichner: \$MC_	Beschreibung
20128	COLLECT_TOOL_CHANGE	Werkzeugwechselbefehle an PLC nach Satzsuchlauf
22600	SERUPRO_SPEED_MODE	Geschwindigkeit bei Satzsuchlauf-Typ 5
22601	SERUPRO_SPEED_FACTOR	Geschwindigkeitsfaktor bei Satzsuchlauftyp 5
22621	ENABLE_START_MODE_MASK_PRT	Schaltet MD22620: START_MODE_MASK_PRT bei Satzsuchlauf SERUPRO frei

2.23 Datenlisten

Nummer	Bezeichner: \$MC_	Beschreibung
22622	DISABLE_PLC_START	Teileprogrammstart via PLC erlauben
22680	AUTO_IPTR_LOCK	Unterbrechungszeiger sperren

Reset-Verhalten

Nummer	Bezeichner: \$MC_	Beschreibung
20110	RESET_MODE_MASK	Grundstellung nach RESET/Teileprogrammende
20112	START_MODE_MASK	Grundstellung bei NC-Start nach Teileprogrammstart
20118	GEOAX_CHANGE_RESET	Automatischen Geometrieachswechsel erlauben
20120	TOOL_RESET_VALUE	Werkzeug-Längenkorrektur im Hochlauf (RESET/ Teileprogrammende)
20121	TOOL_PRESEL_RESET_VALUE	Vorgewähltes Werkzeug bei RESET
20130	CUTTING_EDGE_RESET_VALUE	Werkzeugschneiden-Längenkorrektur im Hochlauf (RESET/Teileprogrammende)
20140	TRAFO_RESET_VALUE	Transformationsdatensatz im Hochlauf (RESET/Teileprogrammende)
20150	GCODE_RESET_VALUES	Löschstellung der G-Gruppen
20152	GCODE_RESET_MODE	Resetverhalten der G-Gruppen
20156	MAXNUM_GCODES_EXT	Resetverhalten der externen G-Gruppen
22620	START_MODE_MASK_PRT	Grundstellung bei speziellem NC-Start nach Hochlauf und bei RESET

Hilfsfunktionseinstellungen

Nummer	Bezeichner: \$MC_	Beschreibung
22000	AUXFU_ASSIGN_GROUP	Hilfsfunktionsgruppe
22010	AUXFU_ASSIGN_TYPE	Hilfsfunktionsart
22020	AUXFU_ASSIGN_EXTENSION	Hilfsfunktionserweiterung
22030	AUXFU_ASSIGN_VALUE	Hilfsfunktionswert
22200	AUXFU_M_SYNC_TYPE	Ausgabezeitpunkt der M-Funktionen
22210	AUXFU_S_SYNC_TYPE	Ausgabezeitpunkt der S-Funktionen
22220	AUXFU_T_SYNC_TYPE	Ausgabezeitpunkt der T-Funktionen
22230	AUXFU_H_SYNC_TYPE	Ausgabezeitpunkt der H-Funktionen
22240	AUXFU_F_SYNC_TYPE	Ausgabezeitpunkt der F-Funktionen
22250	AUXFU_D_SYNC_TYPE	Ausgabezeitpunkt der D-Funktionen
22400	S_VALUES_ACTIVE_AFTER_RESET	S-Funktion über RESET hinaus wirksam
22410	F_VALUES_ACTIVE_AFTER_RESET	F-Funktion über RESET hinaus wirksam
22510	GCODE_GROUPS_TO_PLC	G-Befehle, die bei Satzwechsel/Reset an die Nahtstelle NC/PLC ausgegeben werden
22550	TOOL_CHANGE_MODE	Neue Werkzeugkorrektur bei M-Funktion
22560	TOOL_CHANGE_M_CODE	M-Funktion für Werkzeugwechsel

Speichereinstellungen

Nummer	Bezeichner: \$MC_	Beschreibung
27900	REORG_LOG_LIMIT	Prozentsatz des IPO-Puffers für Freigabe des Logfiles
28000	MM_REORG_LOG_FILE_MEM	SpeichergroÙe für REORG (DRAM)
28010	MM_NUM_REORG_LUD_MODULES	Anzahl der Bausteine für lokale Anwendervariablen bei REORG
28020	MM_NUM_LUD_NAMES_TOTAL	Anzahl der lokalen Anwendervariablen (DRAM)
28040	MM_LUD_VALUES_MEM	SpeichergroÙe für lokale Anwendervariablen (DRAM)
28050	MM_NUM_R_PARAM	Anzahl der kanalspezifischen R-Parameter (SRAM)
28060	MM_IPO_BUFFER_SIZE	Anzahl der NC-Sätze im IPO-Puffer (DRAM)
28070	MM_NUM_BLOCKS_IN_PREP	Anzahl der Sätze für die Satzaufbereitung. (DRAM)
28080	MM_NUM_USER_FRAMES	Anzahl der einstellbaren Frames (SRAM)
28090	MM_NUM_CC_BLOCK_ELEMENTS	Anzahl Satzelemente für Compile-Zyklen (DRAM)
28100	MM_NUM_CC_BLOCK_USER_MEM	GröÙe des Satzspeichers für Compile-Zyklen (DRAM)
28400	MM_ABSBLOCK	Basissätze mit Absolutwerten aktivieren
28402	MM_ABSBLOCK_BUFFER_CONF	GröÙe des Upload-Buffers dimensionieren
28500	MM_PREP_TASK_STACK_SIZE	StackgröÙe der Präparationstask (DRAM)

Programmlaufzeit und Werkstückzähler

Nummer	Bezeichner: \$MC_	Beschreibung
27860	PROCESSTIMER_MODE	Aktivierung der Laufzeit-Messung
27880	PART_COUNTER	Aktivierung der Werkstück-Zähler
27882	PART_COUNTER_MCODE[]	Werkstückzählung über einen M-Befehl

2.23.2.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30465	AXIS_LANG_SUB_MASK	Substituierung von NC-Sprachbefehlen
30550	AXCONF_ASSIGN_MASTER_CHAN	Löschstellung des Kanals für Achswechsel
30600	FIX_POINT_POS	Festwertpositionen der Achsen bei G75
33100	COMPRESS_POS_TOL	Maximale Abweichung bei Kompression

2.23.3 Settingdaten

2.23.3.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42000	THREAD_START_ANGLE	Startwinkel bei Gewinde
42010	THREAD_RAMP_DISP	Beschleunigungsverhalten der Achse beim Gewinde-schneiden
42100	DRY_RUN_FEED	Probelaufvorschub
42200	SINGLEBLOCK2_STOPRE	Debugmode für SBL2 aktivieren
42444	TARGET_BLOCK_INCR_PROG	Aufsetzmodus nach Satzsuchlauf mit Berechnung
42700	EXT_PROG_PATH	Programmpfades für externen Unterprogrammaufruf EXTCALL
42750	ABSBLOCK_ENABLE	Basis-Satzanzeige freigeben
42990	MAX_BLOCKS_IN_IPOBUFFER	Maximale Anzahl der Sätze im Interpolationspuffer

K2: Achsen, Koordinatensysteme, Frames

3.1 Kurzbeschreibung

3.1.1 Achsen

Maschinenachsen

Maschinenachsen sind die real an der (Werkzeug-)Maschine vorhandenen Achsen.

Kanalachsen

Jede Geometrieachse und jede Zusatzachse wird einem Kanal und somit einer Kanalachse zugewiesen. Geometrieachsen und Zusatzachsen werden immer in "ihrem" Kanal verfahren.

Geometrieachsen

Die drei Geometrieachsen bilden immer ein fiktives rechtwinkliges Koordinatensystem, das Basiskoordinatensystem (BKS).

Durch Verwendung von FRAMES (Verschiebung, Drehung, Skalierung, Spiegelung) können Geometrieachsen des Werkstückkoordinatensystems (WKS) auf das BKS abgebildet werden.

Zusatzachsen

Bei Zusatzachsen ist, im Gegensatz zu Geometrieachsen, kein geometrischer Zusammenhang zwischen den Achsen definiert.

Bahnachsen

Bahnachsen zeichnen sich dadurch aus, dass sie gemeinsam interpoliert werden (alle Bahnachsen eines Kanals haben gemeinsam einen Bahninterpolator).

Alle Bahnachsen eines Kanals haben eine gemeinsame Beschleunigungsphase, eine Konstantfahrphase und eine Verzögerungsphase.

Positionierachsen

Positionierachsen zeichnen sich dadurch aus, dass sie getrennt interpoliert werden (jede Positionierachse hat einen eigenen Achsinterpolator). Jede Positionierachse hat einen eigenen Vorschub und eine eigene Beschleunigungskennlinie.

3.1 Kurzbeschreibung

Synchronachsen

Synchronachsen werden gemeinsam mit Bahnachsen interpoliert (alle Bahnachsen und Synchronachsen eines Kanals haben einen gemeinsamen Bahninterpolator).

Alle Bahnachsen und alle Synchronachsen eines Kanals haben eine gemeinsame Beschleunigungsphase, eine Konstantfahrphase und eine Verzögerungsphase.

Achskonfiguration

Die Zuordnung zwischen den Geometrieachsen, Zusatzachsen, Kanalachsen und Maschinenachsen, sowie die Festlegung der Namen der einzelnen Achstypen wird über folgende Maschinendaten getroffen:

MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB (Zuordnung Geometrieachse zu Kanalachse)

MD20060 \$MC_AXCONF_GEOAX_NAME_TAB (Geometrieachsname im Kanal)

MD20070 \$MC_AXCONF_MACHAX_USED (Maschinenachsnummer gültig im Kanal)

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB (Kanalachsname im Kanal)

MD10000 \$MN_AXCONF_MACHAX_NAME_TAB (Maschinenachsname)

MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX (Zuordnung Spindel zu Maschinenachse)

Umschaltbare Geometrieachsen

Mit der Funktion "Umschaltbare Geometrieachsen" kann aus dem Teileprogramm heraus der Verbund der Geometrieachsen aus anderen Kanalachsen zusammengesetzt werden.

Achsen, die in einem Kanal zunächst als synchrone Zusatzachsen projektiert sind, können durch einen Programmbefehl eine beliebige Geometrieachse ersetzen.

Link-Achse

Link-Achsen sind Achsen, die an einer anderen NCU physikalisch angeschlossen sind und deren Lageregelung unterliegen. Link-Achsen können dynamisch Kanälen einer **anderen** NCU zugeordnet werden. Link-Achsen sind aus Sicht einer bestimmten NCU nicht-lokale Achsen.

Der dynamischen Änderung der Zuordnung zu einer NCU dient das Konzept der **Achscontainer**. Achstausch mit `GET` und `RELEASE` aus dem Teileprogramm ist für Link-Achsen über NCU-Grenzen hinweg nicht verfügbar.

Weitere Informationen

Die Link-Achsen sind beschrieben unter B3: Dezentrale Systeme (Seite 765).

Achscontainer

Ein Achscontainer ist eine Ringpuffer-Datenstruktur, in der die Zuordnung von lokalen Achsen und/oder Link-Achsen zu Kanälen erfolgt. Die Einträge im Ringpuffer sind **zyklisch verschiebbar**.

Die Link-Achsen Konfiguration lässt im logischen Maschinenachs-Abbild neben dem direkten Verweis auf lokale Achsen oder Link-Achsen den Verweis auf Achscontainer zu.

Ein solcher Verweis besteht aus:

- Achscontainer-Nummer
- Slot (Ringpuffer-Platz innerhalb des entsprechenden Achscontainers)

Als Eintrag in einem Ringpuffer-Platz steht:

- Eine lokale Achse
oder
- Eine Link-Achse

Weitere Informationen

Die Achscontainer sind beschrieben unter B3: Dezentrale Systeme (Seite 765).

3.1.2 Koordinatensysteme

MKS

Das Maschinenkoordinatensystem (MKS) zeichnet sich durch folgende Eigenschaften aus:

- Es wird durch die Maschinenachsen gebildet.
- Die Maschinenachsen können rechtwinkelig kartesisch oder beliebig zueinander angeordnet sein.
- Die Maschinenachsen haben einstellbare Namen.
- Die Maschinenachsen können Linearachsen oder Rundachsen sein.

BKS

Das Basiskoordinatensystem (BKS) zeichnet sich durch folgende Eigenschaften aus:

- Die Geometrieachsen bilden ein rechtwinkeliges kartesisches Koordinatensystem.
- Das BKS geht durch kinematische Transformation aus dem MKS hervor.

BNS

Das Basis-Nullpunktsystem (BNS) ist das Basis-Koordinatensystem mit Basis-Verschiebung.

ENS

Das Einstellbare Nullpunktsystem (ENS) ist das Werkstückkoordinatensystem mit programmierbarem Frame aus Sicht vom WKS. Der Werkstücknullpunkt wird durch die einstellbaren Frames G54 ... G599 festgelegt.

WKS

Das Werkstückkoordinatensystem (WKS) zeichnet sich durch folgende Eigenschaften aus:

- Im WKS werden alle Koordinaten der Achsen programmiert (Teileprogramm).
- Es wird durch Geometrieachsen und Zusatzachsen gebildet.
- Geometrieachsen bilden immer ein rechtwinkeliges kartesisches Koordinatensystem
- Zusatzachsen bilden ein Koordinatensystem ohne geometrischen Zusammenhang zwischen den Zusatzachsen.
- Geometrieachsen und Zusatzachsen haben einstellbare Namen.
- Durch FRAMES kann das WKS verschoben, gedreht, skaliert oder gespiegelt werden (TRANS, ROT, SCALE, MIRROR).
Auch Mehrfachverschiebungen, Mehrfachdrehungen, etc. sind möglich.

Externe Nullpunktverschiebung

Die externe Nullpunktverschiebung hat folgende Eigenschaften:

- Zu einem von PLC bestimmten Zeitpunkt wird eine zuvor festgelegte zusätzliche Nullpunktverschiebung zwischen Basis- und Werkstück-Koordinatensystem aktiviert.
- Die Beträge der Verschiebungen können pro beteiligte Achse gesetzt werden durch:
 - PLC
 - Bedientafel
 - Teileprogramm
- Aktivierte Verschiebungen wirken sich ab dem Abarbeiten des ersten Bewegungssatzes dieser Achsen nach der Aktivierung aus. Die Verschiebungen werden dem programmierten Weg überlagert (keine Interpolation).
Die Geschwindigkeit, mit der die externe Nullpunktverschiebung herausgefahren wird, beträgt:
Programmierter F-Wert + 1/2 JOG-Geschwindigkeit
In G0-Sätzen wird die externe Nullpunktverschiebung am Satzende gefahren.
- Die aktivierten Verschiebungen werden bei RESET und Programmende beibehalten.
- Nach Power On ist die zuletzt aktive Verschiebung weiterhin in der Steuerung gespeichert, muss aber über die PLC wieder aktiviert werden.

3.1.3 Frames

Ein Frame stellt eine geschlossene Rechenvorschrift dar, die kartesische Koordinatensysteme ineinander überführt.

Frame-Komponenten

Ein Frame setzt sich aus folgenden Komponenten zusammen:

Frame-Komponente		Programmierbar mit:
Verschiebung	Grobverschiebung	TRANS ATRANS (additiver Translationsanteil) CTRANS (Nullpunktverschiebung für mehrere Achsen) G58 (axiale Nullpunktverschiebung)
	Feinverschiebung	CFINE G59 (axiale Nullpunktverschiebung)
Drehung		ROT / ROTs AROT / AROTs CROTs
Skalierung		SCALE ASCALE
Spiegelung		MIRROR AMIRROR

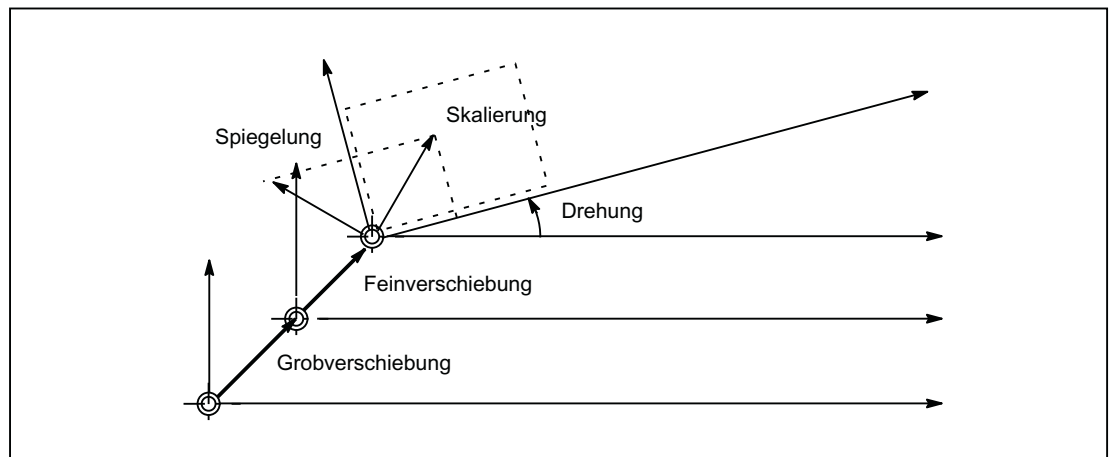


Bild 3-1 Frame-Komponenten

Grob- und Feinverschiebung

Da die Zuordnung von Maschinenachsen zu Kanalachsen und speziell zu Geometrieachsen in allen Kanälen unterschiedlich sein kann, gibt es demzufolge keine eindeutigen kanalübergreifenden geometrischen Zusammenhang zwischen den Kanalachsen. Bei NCU-globalen Frames sind daher nur Verschiebung, Skalierung und Spiegelung möglich. Drehungen sind nicht möglich.

G58: Absolute achsspezifische programmierbare Nullpunktverschiebung (Grobverschiebung)

Mit G58 wird achsspezifisch der absolute Anteil der translatorischen Verschiebung (Grobverschiebung) programmiert. Der additive Anteil der translatorischen Verschiebung (Feinverschiebung) bleibt erhalten.

3.1 Kurzbeschreibung

G59: Additive achsspezifische programmierbare Nullpunktverschiebung (Feinverschiebung)

Mit G59 wird achsspezifisch der additive Anteil der translatorischen Verschiebung (Feinverschiebung) programmiert. Der absolute Anteil der translatorischen Verschiebung (Grobverschiebung) bleibt erhalten.

G59 kann nur verwendet werden, wenn die Feinverschiebung freigegeben wurde:

- MD18600 \$MN_MM_FRAME_FINE_TRANS = TRUE
- MD24000 \$MC_FRAME_ADD_COMPONENTS = TRUE

Drehung

Orientierungen im Raum werden über Frame-Drehungen wie folgt festgelegt:

- Drehung mit ROT legt die Einzeldrehungen für alle Geometrieachsen fest.
- Raumwinkel mit ROTs, AROTs, CROTs legen die Orientierung einer Ebene im Raum fest.
- Framedrehung mit TOFRAME definiert einen Frame, dessen Z-Achse in Werkzeugrichtung zeigt.

Skalierung

Mit SCALE werden die programmierbaren Skalierungen (Maßstabfaktor) für alle Geometrieachsen und Zusatzachsen programmiert.

Soll eine neue Skalierung auf einer anderen Skalierung, Drehung, Verschiebung oder Spiegelung aufbauen, muss ASCALE programmiert werden.

Spiegeln

Mit folgendem Maschinendatum wird eingestellt, wie das Spiegeln ausgeführt wird:

MD10610 \$MN_MIRROR_REF_AX

Verkettung

Frames und Frame-Komponenten lassen sich über den Verkettungsoperator ":" verketteten. Der kanalspezifische Aktive Frame \$P_ACTFRAME ergibt sich z.B. aus der Verkettung aller aktiven Frames des Kanals:

```
$P_ACTFRAME = $P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME :  
               $P_ISO1FRAME : $P_ISO2FRAME : $P_ISO3FRAME :  
               $P_ACTBFRAME : $P_IFRAME : $P_GFRAME :  
               $P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME :  
               $P_PFRAME : $P_ISO4FRAME : $P_CYCFRAME
```

Randbedingungen

Kettenmaßangabe G91

Kettenmaß-Programmierung mit G91 ist so definiert, dass bei Anwahl einer Nullpunktverschiebung der Korrekturwert additiv zum inkrementell programmierten Wert gefahren wird.

Das Verhalten ist abhängig von der Einstellung im Settingdatum:

SD42440 \$SC_FRAME_OFFSET_INCR_PROG (Nullpunktverschiebungen in Frames)

Wert	Bedeutung
1	Nullpunktverschiebung wird bei FRAME und inkrementeller Programmierung einer Achse herausgefahren (= Standardeinstellung).
0	Nur der programmierte Weg wird gefahren.

Konsistenz

Es liegt in der alleinigen Verantwortung des Anwenders beim Schreiben, Lesen und Aktivieren von Frames z. B. mittels Kanalkoordinierung, ein konsistentes Verhalten innerhalb der Kanäle zu erzielen. Eine kanalübergreifende Aktivierung von Frames ist nicht möglich.

3.2 Achsen

3.2.1 Übersicht

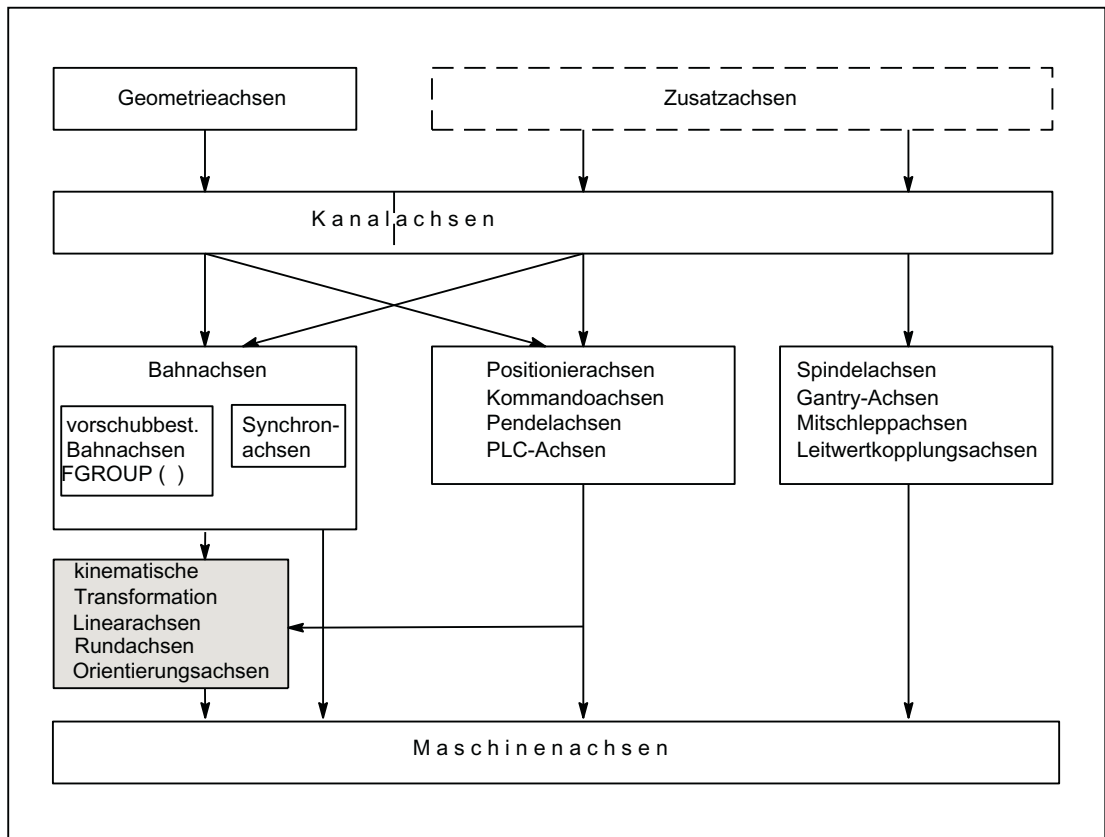


Bild 3-2 Zusammenhang zwischen Geometrie-, Zusatz- und Maschinenachsen

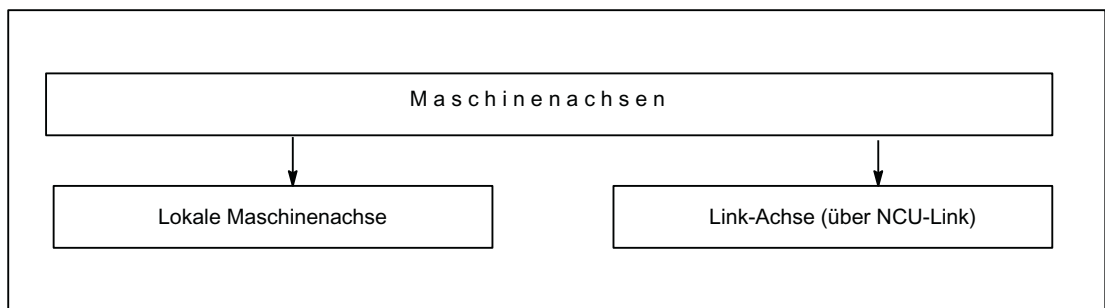


Bild 3-3 Lokale und externe Maschinenachsen (Link-Achsen)

3.2.2 Maschinenachsen

Bedeutung

Maschinenachsen sind die real an der (Werkzeug-)Maschine vorhandenen Achsen.

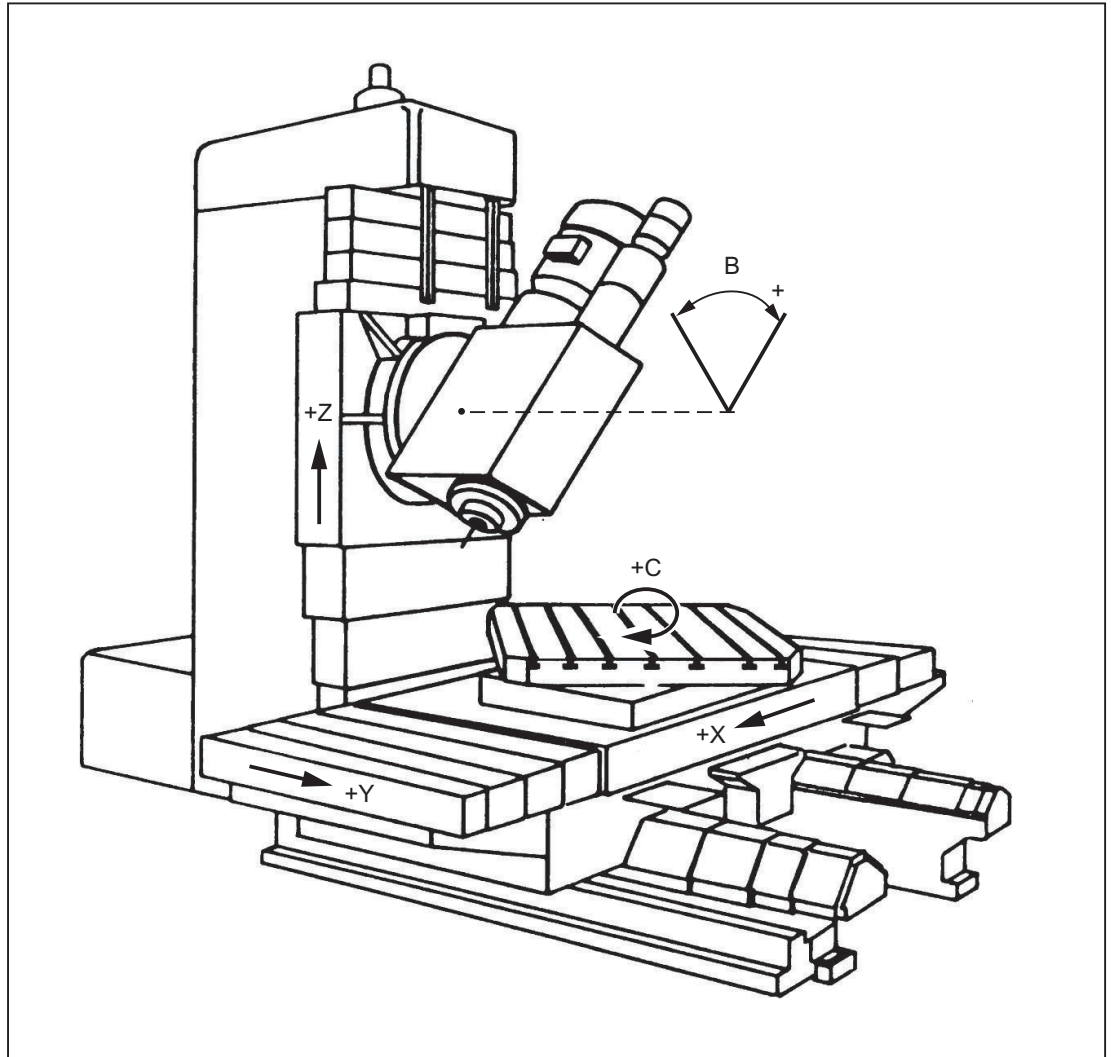


Bild 3-4 Maschinenachsen X, Y, Z, B, S einer kartesischen Maschine

Anwendung

Maschinenachsen können sein:

- Geometrieachsen X, Y, Z
- Orientierungsachsen A, B, C
- Laderachsen
- Werkzeugrevolver

3.2 Achsen

- Achsen für Werkzeugmagazin
- Achsen für Werkzeugwechsler
- Pinole
- Achsen für Palettenwechsler
- etc.

3.2.3 Kanalachsen

Bedeutung

Jede Geometrieachse und jede Zusatzachse wird einem Kanal zugewiesen. Geometrieachsen und Zusatzachsen werden immer in "ihrem" Kanal verfahren.

3.2.4 Geometrieachsen

Bedeutung

Die drei Geometrieachsen bilden immer ein fiktives rechtwinkliges Koordinatensystem.

Durch Verwendung von FRAMES (Verschiebung, Drehung, Skalierung, Spiegelung) können Geometrieachsen des Werkstückkoordinatensystems (WKS) auf das BKS abgebildet werden.

Anwendung

Geometrieachsen werden für Programmierung der Werkstückgeometrie (der Kontur) verwendet.

Die Ebenenanwahl G17, G18 und G19 (DIN 66217) bezieht sich immer auf die drei Geometrieachsen. Deshalb ist es vorteilhaft, die drei Geometrieachsen X, Y und Z zu nennen.

3.2.5 Zusatzachsen

Bedeutung

Bei Zusatzachsen ist, im Gegensatz zu Geometrieachsen, kein geometrischer Zusammenhang zwischen den Achsen definiert.

Hinweis

Geometrieachsen haben einen exakt definierten Zusammenhang in Form eines rechtwinkligen Koordinatensystems.

Zusatzachsen sind Bestandteile des Basiskoordinatensystems (BKS). Durch Verwendung von FRAMES (Verschiebung, Skalierung, Spiegelung) können Zusatzachsen des Werkstückkoordinatensystems (WKS) auf das BKS abgebildet werden.

Anwendung

Typische Zusatzachsen sind:

- Rundachsen
- Werkzeugmagazinachsen
- Werkzeugrevolverachsen
- Laderachsen

3.2.6 Bahnachsen

Bedeutung

Bahnachsen zeichnen sich dadurch aus, dass sie gemeinsam interpoliert werden (alle Bahnachsen eines Kanals haben gemeinsam einen Bahninterpolator).

Alle Bahnachsen eines Kanals haben eine gemeinsame Beschleunigungsphase, eine Konstantfahrphase und eine Verzögerungsphase.

Der unter der Adresse `F` programmierte Vorschub (Bahnvorschub) gilt für alle im Satz programmierten Bahnachsen, mit folgenden Ausnahmen:

- Es wurde eine Achse programmiert, die mit der Anweisung `FGROUP` als nicht bahngeschwindigkeitsbestimmend festgelegt wurde.
- Achsen, die mit den Anweisungen `POS` oder `POSA` programmiert wurden, besitzen einen eigenen Vorschub (Achsinterpolator).

Anwendung

Bahnachsen dienen zum Bearbeiten des Werkstücks mit der programmierten Kontur.

3.2.7 Positionierachsen

Bedeutung

Positionierachsen zeichnen sich dadurch aus, dass sie getrennt interpoliert werden (jede Positionierachse hat einen eigenen Achsinterpolator). Jede Positionierachse hat einen eigenen Vorschub und eine eigene Beschleunigungskennlinie. Positionierachsen können zusätzlich zu Bahnachsen (auch im gleichen Satz) programmiert werden. Die Interpolation der Bahnachsen (Bahninterpolator) wird durch Positionierachsen nicht beeinflusst. Bahnachsen und die einzelnen Positionierachsen erreichen nicht notwendigerweise gleichzeitig ihre Satzendpunkte.

3.2 Achsen

Durch die Anweisungen `POS` und `POSA` werden Positionierachsen programmiert und das Satzwechselkriterium festgelegt:

- `POS`
Der Satzwechsel erfolgt, wenn Bahnachsen und Positionierachsen ihre Satzendpunkte erreicht haben.
- `POSA`
Der Satzwechsel erfolgt, wenn Bahnachsen ihre Satzendpunkte erreicht haben. Positionierachsen laufen über Satzgrenzen hinweg bis zu ihrem Satzendpunkt weiter.

Konkurrierende Positionierachsen unterscheiden sich von Positionierachsen durch folgende Punkte:

- Sie bekommen ihre Satzendpunkte ausschließlich von der PLC.
- Sie können zu jedem beliebigen Zeitpunkt (nicht an Satzgrenzen) gestartet werden.
- Sie beeinflussen das laufende Teileprogramm in ihrer Bearbeitung nicht.

Anwendung

Typische Positionierachsen sind:

- Lader für Werkstücktransport
- Werkzeugmagazin / -revolver

Weitere Informationen

- Funktionshandbuch Achsen und Spindeln
- Funktionshandbuch Synchronaktionen
- Funktionshandbuch PLC und Grundprogramm

3.2.8 Hauptlaufachsen

Bedeutung

Als Hauptlaufachse wird eine Achse bezeichnet, die vom Hauptlauf interpoliert wird.

Diese Interpolation kann folgendermaßen gestartet werden:

- Aus Synchronaktionen
(als Kommandoachsen auf Grund eines Ereignisses über satzbezogene, modale oder statische Synchronaktionen)
- Vom PLC über spezielle Funktionsbausteine im PLC-Grundprogramm
(als konkurrierende Positionierachse oder auch PLC-Achse genannt)
- Durch Settingdatum oder aus dem Teileprogramm
(als asynchrone oder satzsynchrone Pendelachse)

Beeinflussung

Eine vom Hauptlauf interpolierte Achse reagiert bezüglich:

- NC-Stop
- Alarmbehandlung
- Programmbeeinflussungen
- Programmende
- RESET

Hinweis

Das Verhalten am Programmende ist unterschiedlich. Die Achsbewegung muss nicht immer am Programmende abgeschlossen sein und kann damit auch über das Programmende erfolgen.

Anwendung

Bestimmte Achsen können im Hauptlauf von dem durch den NC-Programmablauf getriggerten Kanalverhalten entkoppelt und vom PLC aus kontrolliert werden. Auch diese Achsen werden im Hauptlauf interpoliert und verhalten sich unabhängig vom Kanal- und Programmablauf.

Eine vom PLC kontrollierte Achse kann dann autark von der NC beeinflusst werden. Dies betrifft folgende Aktionen:

- Ablauf der Achse abbrechen (entspricht Restweg löschen)
- Achse stoppen bzw. unterbrechen
- Achse weiterfahren (Bewegungsablauf fortsetzen)
- Achse in den Grundzustand zurücksetzen

3.2.9 Synchronachsen

Bedeutung

Synchronachsen sind Teil der Bahnachsen, die nicht zur Berechnung der Bahngeschwindigkeit herangezogen werden. Sie werden gemeinsam mit Bahnachsen interpoliert (alle Bahnachsen und Synchronachsen eines Kanals haben einen gemeinsamen Bahninterpolator).

Alle Bahnachsen und alle Synchronachsen eines Kanals haben eine gemeinsame Beschleunigungsphase, eine Konstantfahrphase und eine Verzögerungsphase.

Der unter der Adresse F programmierte Vorschub (Bahnvorschub) gilt für alle im Satz programmierten Bahnachsen, jedoch nicht für die Synchronachsen.

Synchronachsen benötigen für ihren programmierten Weg die gleiche Zeit wie die Bahnachsen.

Befehl FGROUP

Mit dem Befehl `FGROUP` wird festgelegt, ob die Achse eine vorschubbestimmende **Bahnachse** (wird zur Berechnung der Bahngeschwindigkeit herangezogen) oder eine **Synchronachse** (wird zur Berechnung der Bahngeschwindigkeit nicht herangezogen) ist.

Beispiel

Programmcode	Kommentar
N05 G00 G94 G90 M3 S1000 X0 Y0 Z0	;
N10 FGROUP (X,Y)	; Achsen X und Y sind Bahnachsen Achse Z ist Synchronachse
N20 G01 X100 Y100 F1000	; progr. Vorschub 1000 mm/min Vorschub der Achse X = 707 mm/min Vorschub der Achse Y = 707 mm/min
N30 FGROUP (X)	; Achse X ist eine Bahnachse Achse Y ist eine Synchronachse
N20 X200 Y150	; progr. Vorschub 1000 mm/min Vorschub der Achse X = 1000 mm/min Vorschub der Achse Y stellt sich auf 500 mm/min ein, da nur der halbe Weg zu fahren ist.

Hinweis

Beim Befehl `FGROUP` ist der Kanalachsname zu verwenden.

Dies wird definiert durch das Maschinendatum:

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB (Kanalachsname im Kanal)

Anwendung

Bei Schraubenlinieninterpolation (Helixinterpolation) kann durch `FGROUP` wahlweise eingestellt werden:

- Ob der programmierte Vorschub auf der Bahn gelten soll (alle 3 programmierten Achsen sind Bahnachsen)
- Ob der programmierte Vorschub auf dem Kreis gelten soll (2 Achsen sind Bahnachsen und die Zustellachse ist eine Synchronachse)

3.2.10 Achskonfiguration

Zuordnung von Geometrie-, Zusatz-, Kanal- und Maschinenachsen und Antrieben

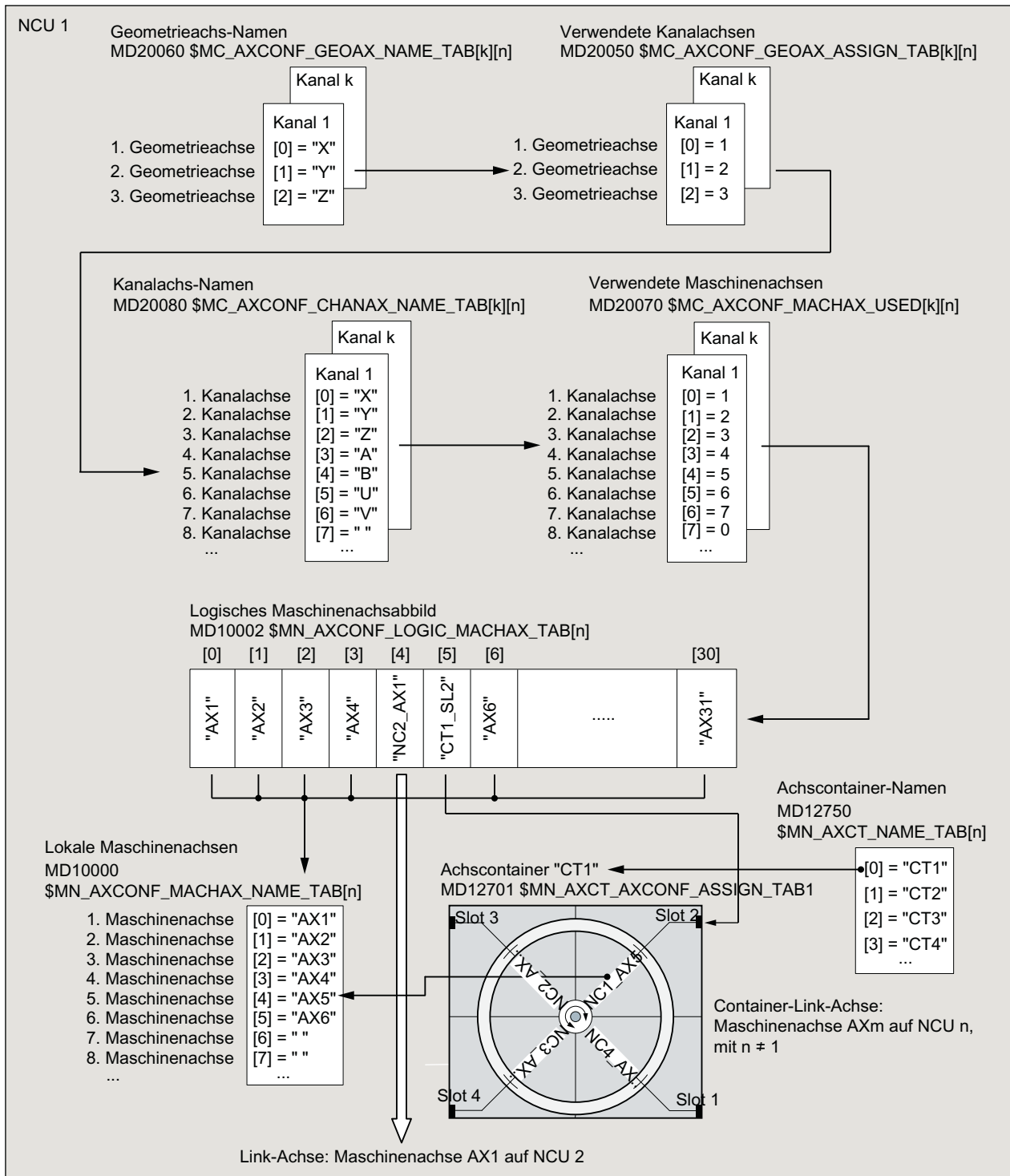


Bild 3-5 Achszuordnung

3.2 Achsen

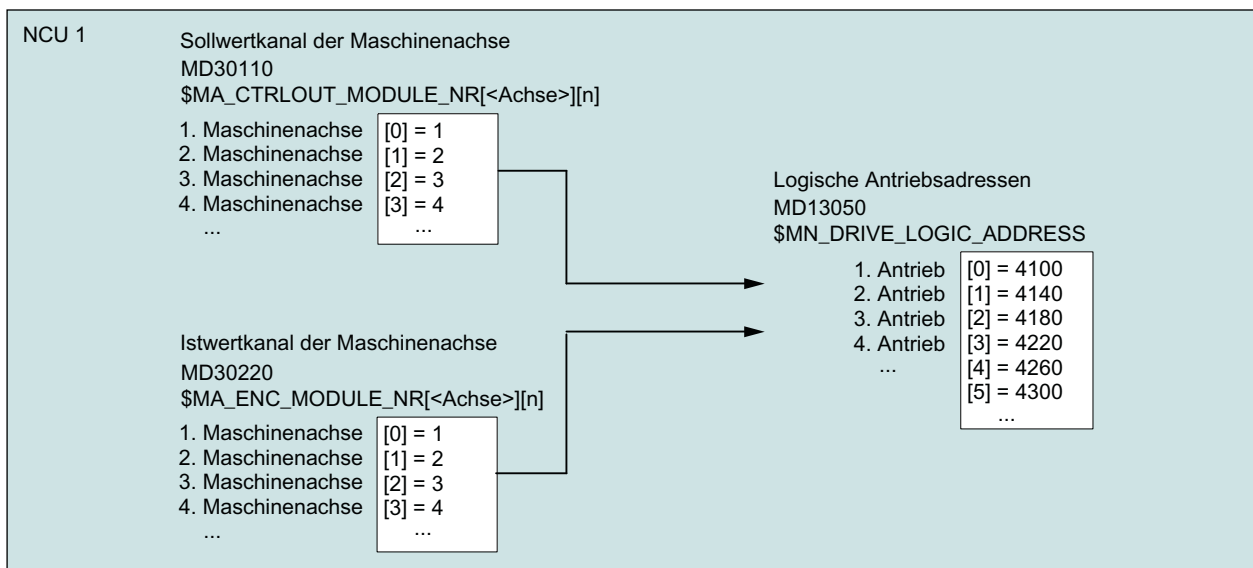


Bild 3-6 Antriebszuordnung

Randbedingungen

- Führende Nullen bei anwenderdefinierten Achsnamen werden ignoriert:
MD10000 ` \$MN_AXCONF_MACHAX_NAME_TAB[0] = X01 entspricht X1
- Die Zuordnung der Geometrieachsen zu den Kanalachsen muss lückenlos und in aufsteigender Reihenfolge erfolgen.
- Alle Kanalachsen, die keine Geometrieachsen sind, sind Zusatzachsen.

Kanalachslücken

Im Normalfall muss jeder im Maschinendatum MD20080 \$MC_AXCONF_CHANAX_NAME_TAB definierten Kanalachse über MD20070 \$MC_AXCONF_MACHAX_USED eine Maschinenachse zugeordnet werden.

Zur Vereinfachung der Inbetriebnahme von Maschinenbaureihen mit unterschiedlicher Anzahl von Maschinenachsen, dürfen auch Kanalachsen definiert sein, denen keine Maschinenachse zugeordnet ist. In der Reihenfolge der Aufzählung der Kanalachsen können dadurch Lücken entstehen.

Die Zulässigkeit von Kanalachslücken muss explizit freigegeben werden:

MD11640 \$MN_ENABLE_CHAN_AX_GAP = 1

Ohne Freigabe beendet ein Wert von 0 im Maschinendatum MD20070 \$MC_AXCONF_MACHAX_USED die Zuweisung weiterer Kanalachsen zu Maschinenachsen.

Beispiel

Der Kanalachse B ist keine Maschinenachse zugeordnet.

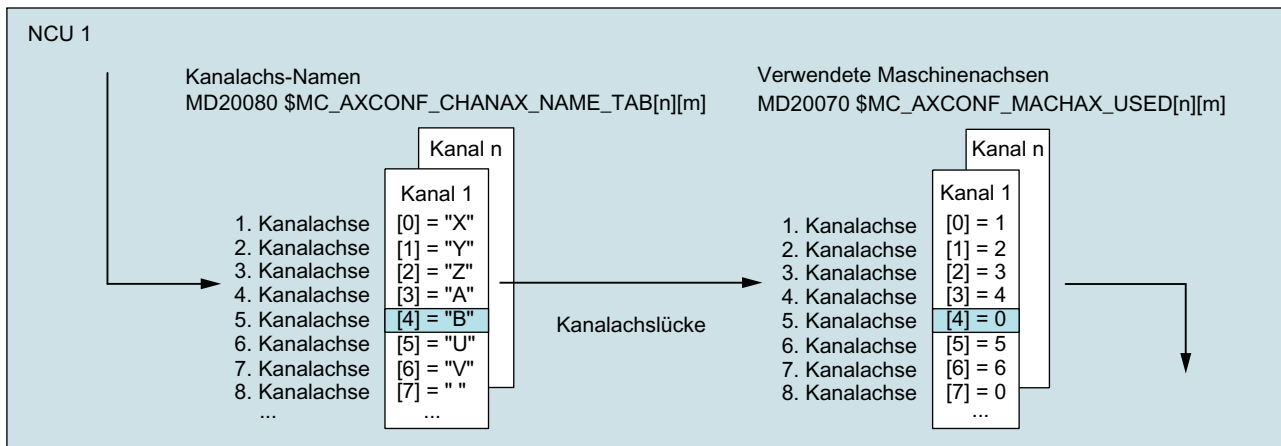


Bild 3-7 Achskonfiguration mit Kanalachslücke (Ausschnitt)

Randbedingungen

- Kanalachsen ohne zugeordnete Maschinenachsen (Kanalachslücken) werden bezüglich Anzahl und Indizierung der Kanalachsen wie normale Kanalachsen mit zugeordneten Maschinenachsen behandelt.
- Wird eine Kanalachse ohne zugeordnete Maschinenachse (Kanalachslücke) als Geometrieachse definiert, wird dies ohne Alarm abgewiesen.

3.2.11 Link-Achsen

Bedeutung

Eine Link-Achse ist eine Maschinenachse, die nicht auf der NCU liegt von der aus sie verfahren wird. Im Maschinendatum für das logische Maschinenachsabbild der verfahrenen NCU wird dazu nicht der Name einer lokalen Maschinenachse eingetragen, sondern der NCU- und der Maschinenachsname der NCU an der sie physikalisch angeschlossen ist.

Beispielhaft soll die Maschinenachse AX1 der NCU2 von NCU1 aus verfahren werden:

- NCU1: MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB[n] = NC2_AX1

Voraussetzung

Als Voraussetzung für die Verwendung von Link-Achsen müssen die beteiligten NCUs per Link-Kommunikation verbunden sein. Die Funktionen Link-Achsen und Link-Kommunikation sind ausführlich beschrieben in B3: Dezentrale Systeme (Seite 765).





3.3 Nullpunkte und Referenzpunkte

3.3.1 Bezugspunkte im Arbeitsraum

Nullpunkte und Referenzpunkte

Aus den Koordinatenachsen und den konstruktiven Merkmalen der Maschine ergibt sich deren Nullstellung. Den Nullpunkt des Koordinatensystems erhält man durch Festlegung eines zweckmäßigen Bezugspunktes an der Maschine in ihrer Nullstellung.

Die Lage der Koordinatensysteme (MKS, BKS, BNS, ENS, WKS) wird durch Nullpunkte festgelegt.

Nullpunkte		Referenzpunkte	
	M = Maschinennullpunkt		R = Referenzpunkt
	W = Werkstücknullpunkt		T = Werkzeugträgerbezugspunkt

Maschinennullpunkt M

Mit dem Maschinennullpunkt M wird das Maschinen-Koordinatensystem MKS festgelegt. Auf den Maschinennullpunkt beziehen sich alle anderen Bezugspunkte.

Werkstücknullpunkt W

Der Werkstücknullpunkt W legt das Werkstück-Koordinatensystem in Bezug auf den Maschinennullpunkt M fest. Im Werkstück-Koordinatensystem WKS werden die programmierten Teileprogrammsätze abgefahren.

Referenzpunkt R

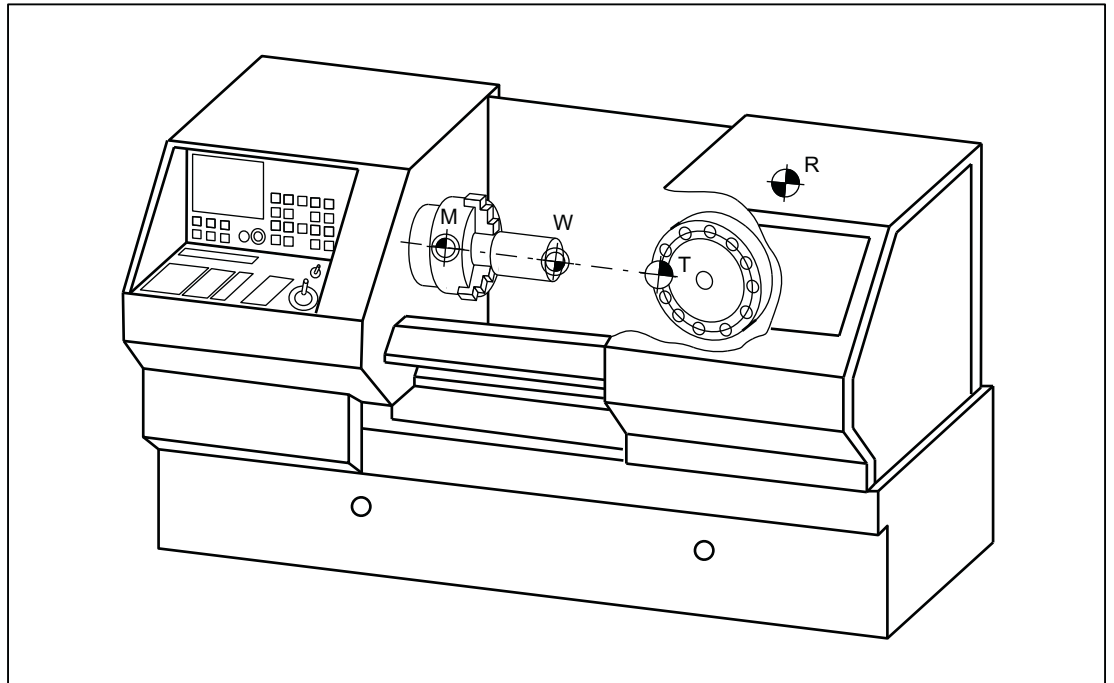
Die Position des Referenzpunktes R wird durch Nockenschalter vorgegeben. Durch ihn wird das Wegmesssystem geeicht.

Bei inkrementellen Messgebern muss der Referenzpunkt nach jedem Einschalten der Steuerung angefahren werden. Erst dann kann die Steuerung mit dem Messsystem arbeiten und alle Positionswerte auf die Koordinatensysteme übertragen.

Werkzeugträgerbezugspunkt T

Der Werkzeugträgerbezugspunkt T befindet sich an der Werkzeughalteraufnahme. Durch Eingabe der Werkzeuglängen berechnet die Steuerung den Abstand der Werkzeugspitze (TCP-Tool Center Position) vom Werkzeugträgerbezugspunkt.

Beispiel: Nullpunkte und Referenzpunkte bei einer Drehmaschine



3.3.2 Lage der Koordinatensysteme und Referenzpunkte

Einschalten der Steuerung

Bei inkrementellen Messgebern muss der Referenzpunkt nach jedem Einschalten der Steuerung angefahren werden, damit die Steuerung alle Positionswerte auf das Koordinatensystem übertragen kann.

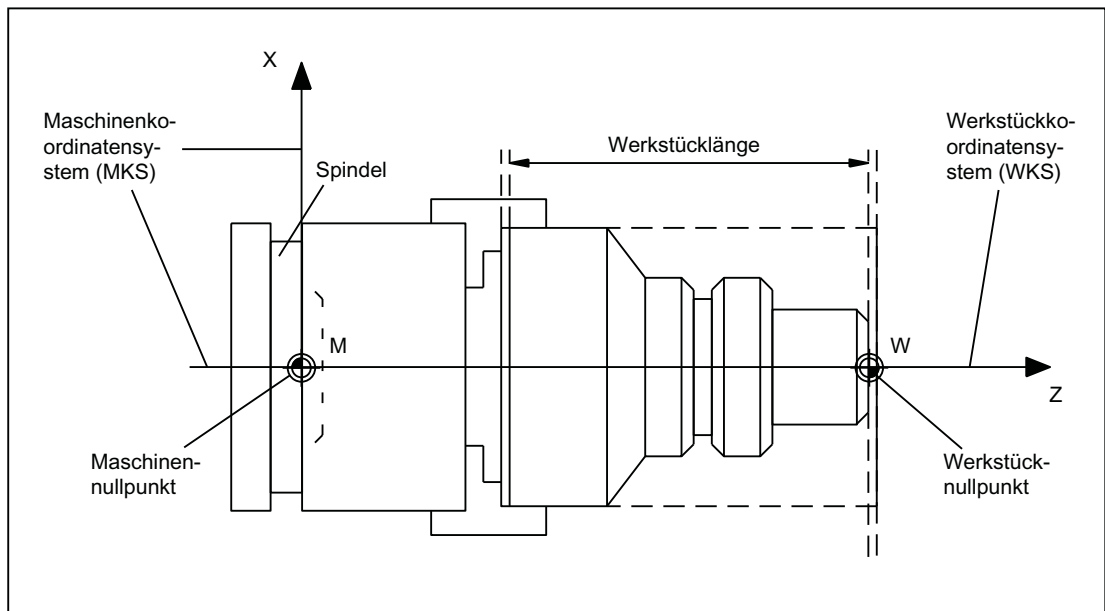


Bild 3-8 Lage der Koordinatensysteme durch Maschinennullpunkt M und Werkstücknullpunkt W

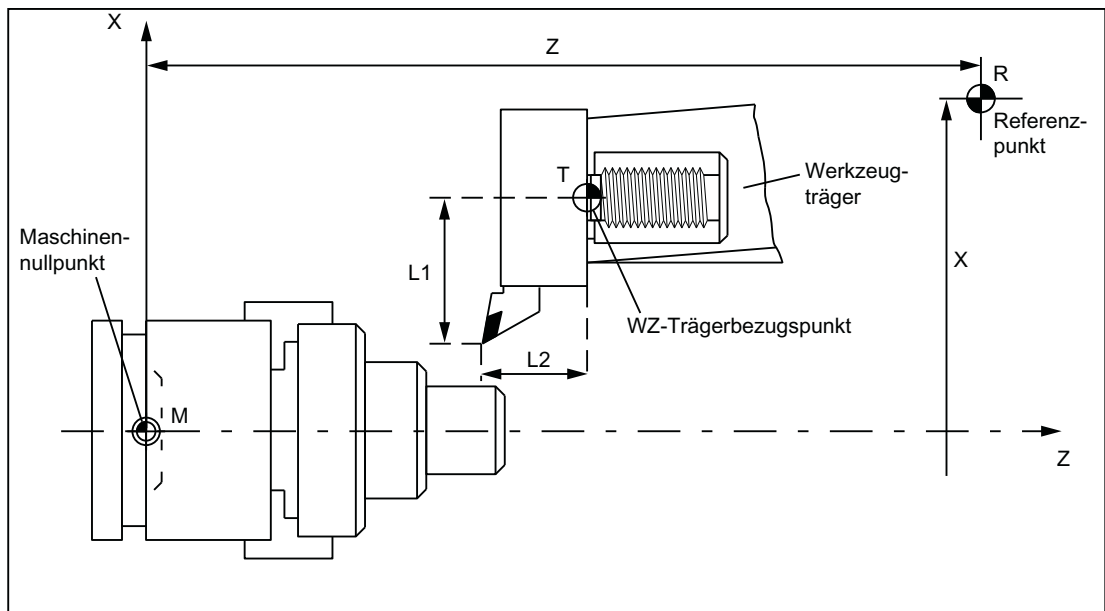


Bild 3-9 Lage des Referenzpunktes zum Maschinennullpunkt

3.4 Koordinatensysteme

3.4.1 Übersicht

Definitionen

Nach DIN 66217 werden bei der Programmierung von Werkzeugmaschinen rechtwinkelige (kartesische) Koordinatensysteme benutzt. Mit der "Rechten-Hand-Regel" können die positiven Richtungen der Koordinatenachsen bestimmt werden.

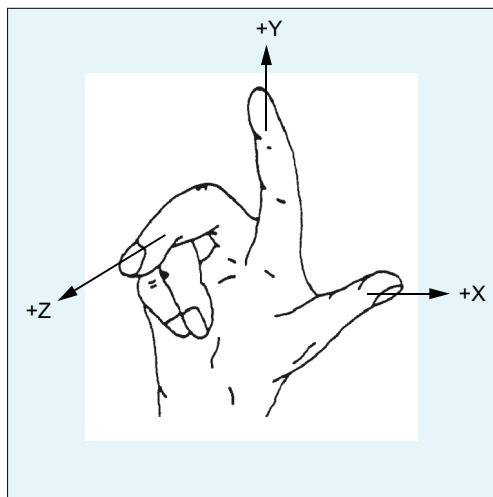


Bild 3-10 Rechte_Hand_Regel

Das Koordinatensystem in dem programmiert wird, wird auf das Werkstück bezogen. Die Programmierung erfolgt unabhängig davon, ob das Werkzeug oder das Werkstück bewegt wird. Bei der Programmierung wird immer davon ausgegangen, dass sich das Werkzeug relativ zum Koordinatensystem des stillstehend gedachten Werkstücks bewegt.

Mit der "Rechten-Hand-Regel" kann auch die positive (rechtsdrehende) Drehrichtung der Rundachsen bestimmt werden. Zeigt der Daumen der rechten Hand in Richtung der positiven Richtung einer Koordinatenachse (Linearachse), zeigen die Finger in Richtung der positiven Drehrichtung der zugehörigen Rundachse.

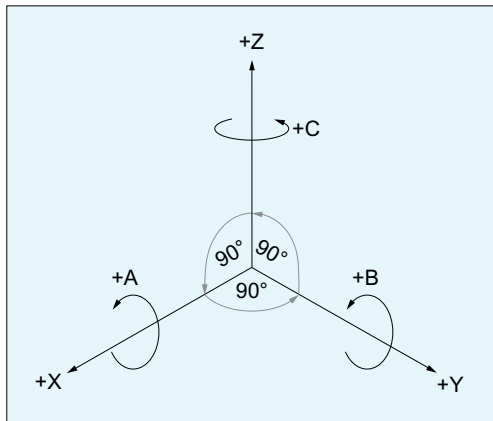


Bild 3-11 Rechtsdrehendes, rechtwinkliges kartesisches Koordinatensystem

Koordinatensysteme

Für eine Werkzeugmaschine sind folgende Koordinatensysteme definiert:

Koordinatensystem	Abkürzung	Bemerkung
Werkstück-KoordinatenSystem	WKS	Im WKS erfolgt die Programmierung der Verfahrensbewegungen der Geometrieachsen für die Bearbeitung des Werkstücks.
Einstellbares NullpunktSystem	ENS	Koordinatentransformation über Frames: WKS \Rightarrow ENS
Basis-NullpunktSystem	BNS	Koordinatentransformation über Frames: ENS \Rightarrow BNS
Basis-KoordinatenSystem	BKS	Koordinatentransformation über Frames: BNS \Rightarrow BKS
Maschinen-KoordinatenSystem	MKS	Koordinatentransformation über Frames und kinematische Transformation: BKS \Rightarrow MKS Die programmierten Verfahrensbewegungen der Geometrieachsen im WKS werden über Frames und kinematische Transformationen auf die Maschinenachsen des MKS abgebildet. Durch eine kinematische Transformation wird das MKS in das BKS überführt. Damit wirken Funktionen wie z. B. das manuelle Verfahren nicht im MKS, sondern im BKS.

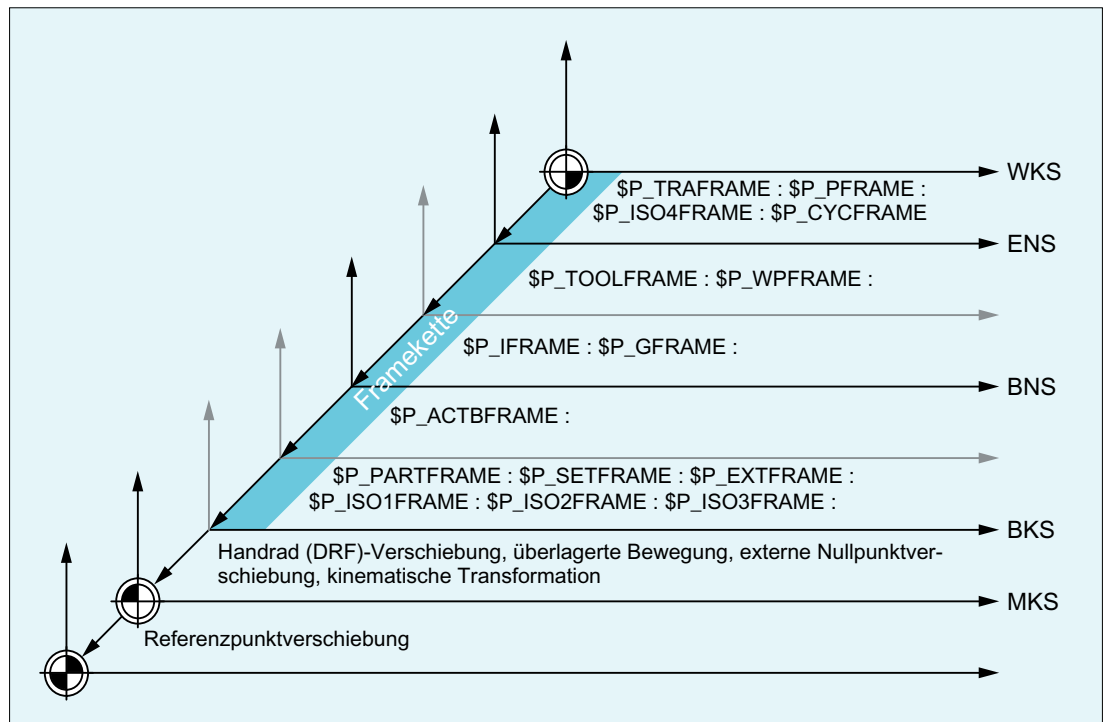


Bild 3-12 Koordinatensysteme, Frames und kinematische Transformationen

3.4.2 Maschinenkoordinatensystem (MKS)

Maschinenkoordinatensystem (MKS)

Das Maschinenkoordinatensystem (MKS) wird aus allen physikalisch vorhandenen Maschinenachsen gebildet.

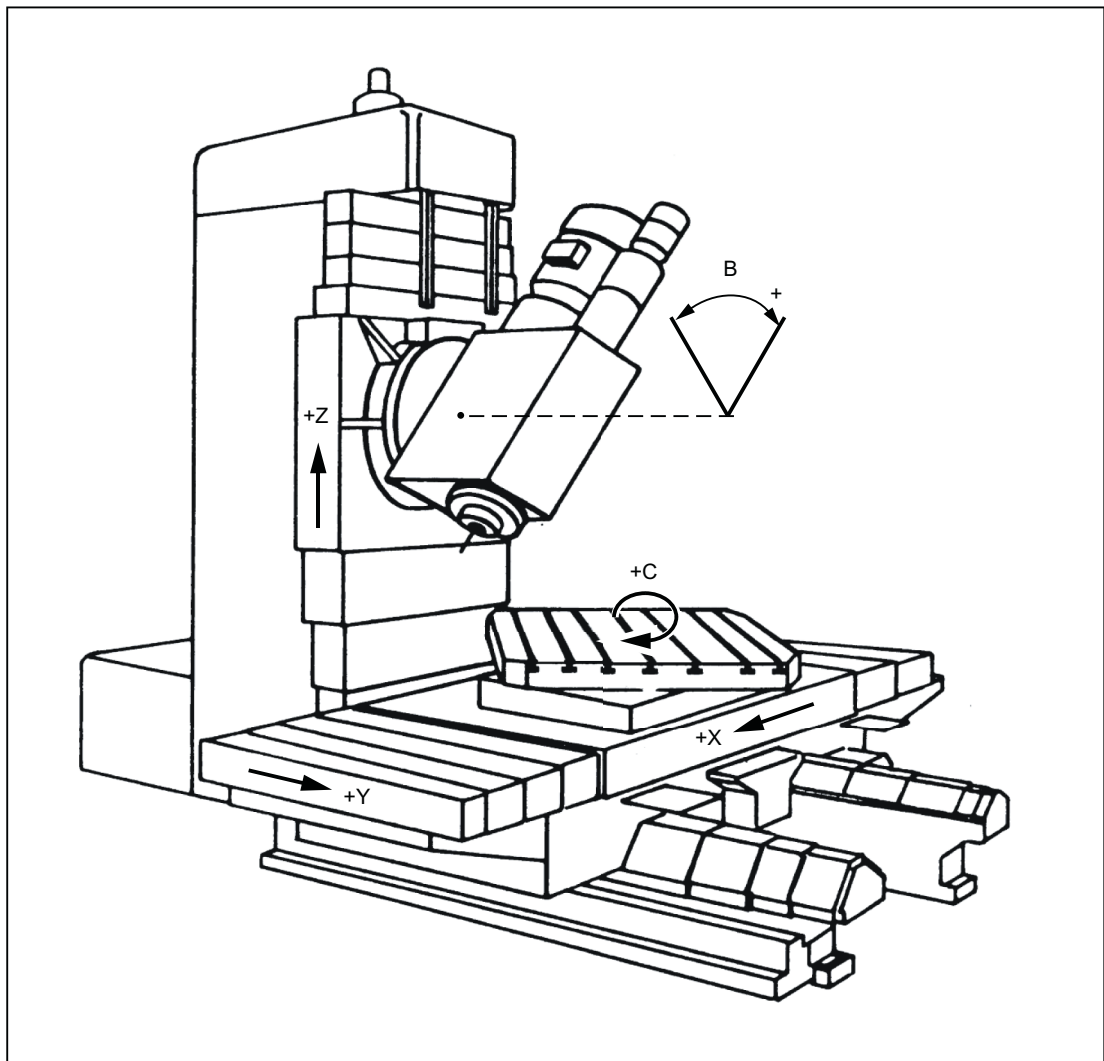


Bild 3-13 MKS mit den Maschinenachsen X, Y, Z, B, C (5-Achs-Fräsmaschine)

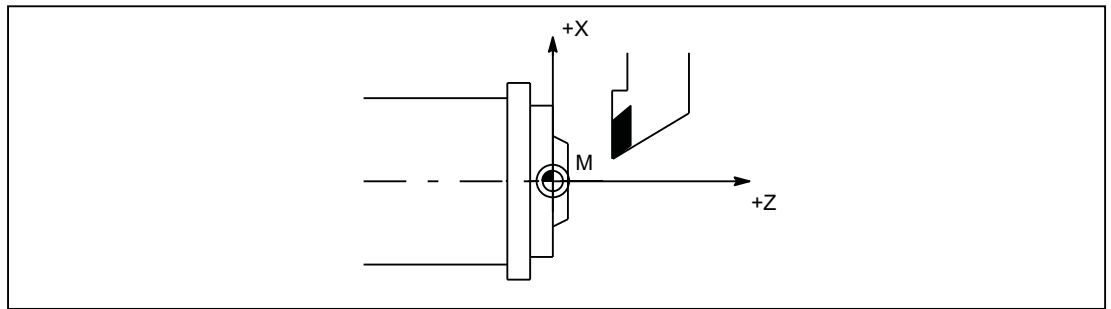


Bild 3-14 MKS mit den Maschinenachsen X, Z (Drehmaschine)

Axiale Preset-Verschiebung

Über die Funktion "Preset-Verschiebung (PRESETON)" kann der Bezugspunkt der Steuerung im Maschinenkoordinatensystem (Maschinennullpunkt) neu gesetzt werden.

VORSICHT

Verlust der Geber-Justage

Nach einer Preset-Verschiebung ist die entsprechende Maschinenachse im Status "nicht referenziert"! Das bedeutet bei Verwendung von Absolutgebern, dass die Geber-Justage verlorengegangen und neu durchgeführt werden muss (z. B. durch Einmessen mit einem Laser-Interferometer). Die Verwendung von PRESETON in Kombination mit Absolutgebern ist deshalb nicht zu empfehlen.

Hinweis

Es wird empfohlen, die Funktion **nur** für Maschinenachsen ohne Referenzpunktpflicht zu verwenden.

Zum Wiederherstellen des ursprünglichen Maschinenkoordinatensystems muss die Maschinenachse z. B. mit G74 (Referenzpunktfahren) erneut referenziert werden.

Bei der Preset-Verschiebung findet keine Bewegung der Maschinenachsen statt.

Weitere Informationen

- Programmierhandbuch Grundlagen, Ergänzende Befehle
- Programmierhandbuch Arbeitsvorbereitung, Koordinatentransformationen (FRAMES)

3.4.2.1 Istwertsetzen mit Verlust des Referenzierstatus (PRESETON)

Funktion

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 dem in folgenden Maschinendatum parametrisierten Achstauschverhalten:

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 PRESETON 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 (DIAMON) wird berücksichtigt Typ: REAL

Systemvariable

\$AC_PRESET

Die achsspezifische Systemvariable \$AC_PRESET liefert den Vektor vom Nullpunkt des aktuellen verschobenen MKS' zum Nullpunkt des ursprünglichen MKS₀ nach dem Referenzieren der Maschinenachse.

$\$AC_PRESET\langle Achse \rangle = \$AC_PRESET\langle Achse \rangle + \text{"aktuelle Istposition der Achse im MKS"} - \text{"PRESETON-Istposition"}$

Mit Hilfe der Systemvariablen kann die Nullpunktverschiebung wieder rückgängig gemacht werden:

`PRESETON(<Achse>, $VA_IM + $AC_PRESET[<Achse>]) ; "Aktueller Istwert der Achse im MKS'" + "Verschiebungen"`

Beispiel

Programmcode

```
N10 G1 X10 F5000
N20 PRESETON(X, $AA_IM[X]+70) ; Istwert = 10 + 70 = 80 =>
                          ; $AC_PRESET = $AC_PRESET - 70
```

Randbedingungen

Achsen bei denen PRESETON nicht angewandt werden darf

- Fahrende Bahnachsen
- Fahrende Positionierachsen
- Fahrende Kommandoachsen im Spindelbetrieb
- Fahrende konkurrierende Positionierachsen (FC18)
- Achsen, die an einer Transformation beteiligt sind
- Achsen, bei denen eine oder mehrere der folgenden Safety Integrated Funktionen aktiv ist:
 - Freigabe "Sichere Software-Endschalter"
 - MD36901 \$MA_SAFE_FUNCTION_ENABLE[<SafeAxis>], Bit 1 = 1
 - Freigabe "Sichere Software-Nocken", Paar 1 ... 4, Nocke +/-
 - MD36901 \$MA_SAFE_FUNCTION_ENABLE[<SafeAxis>], Bit 8 ... 15 = 1
 - bzw.
 - Freigabe "Sichere Nockenspur", Nocke 1 ... 30
 - MD36903 \$MA_SAFE_CAM_ENABLE[<SafeAxis>], Bit 0 ... 29 = 1
- Pendelachsen
- Hirth-Achsen
- Gleichlaufachsen eines Gantry-Verbunds
- Achsen für die Referenzpunktfahren aus dem Teileprogramm (G74) aktiv ist
- Slave-Achse einer Drehzahl-/Drehmomentkopplung (Master-Slave)

Geometrieachsen

- PRESETON kann auf eine stehende Geometrieachse angewandt werden, wenn im Kanal nicht gleichzeitig eine weitere Geometrieachse verfährt.
- PRESETON kann auf eine stehende Geometrieachse angewandt werden, auch wenn im Kanal gleichzeitig eine weitere Geometrieachse verfährt, sich diese aber im Zustand "Neutrale Achse" befindet oder als Kommandoachse verfährt.

Programmcode	Kommentar
N10 G0 X0 Y0	; X, Y: Geometrieachsen
N15 RELEASE(Y)	; Neutrale Achse
N20 PRESETON(Y,20)	; Istposition Y im MKS = 20
N30 G0 X40	; Geometrieachse X verfährt
N40 M30	

Beispiel: Eine andere Geometrieachse (X) verfährt gleichzeitig im Zustand "Neutrale Achse"

Programmcode	Kommentar
N10 G0 X0 Y0	; X, Y: Geometrieachsen
N20 POS[X]=40 FA[X]=1000	; Kommandoachse X
N30 DO PRESETON(Y,20)	; Istposition Y im MKS = 20
N40 M30	

Beispiel: Geometrieachse (X) verfährt gleichzeitig als **Kommandoachse**

PLC-kontrollierte Achsen

PRESETON kann auf PLC-kontrollierte Achsen entsprechend ihres aktuellen Typs angewandt werden.

Spindelzustände

In der nachfolgenden Tabelle sind die Reaktionen aufgezeigt, die erfolgen, wenn PRESETON in einer Synchronaktion auf eine Spindel angewandt wird:

PRESETON im NC-Programm			
Spindelbetriebsart	Verfahrstatus	dem NC-Programm zugeordnet	Hauptlaufachse
Drehzahlsteuerbetrieb	in Bewegung	Alarm 22324	Alarm 22324
	steht	+	+
Positionierbetrieb SPOS	Bewegung	-	+
	steht	+	+
Positionieren über Satzgrenzen SPOSA	in Bewegung	Alarm 10610	-
Achsbetrieb	in Bewegung	-	+
	steht	+	+
+: möglich -: nicht möglich			

Achskopplungen

- Leitachsen: Die durch PRESETON verursachte sprungförmige Änderung der Leitachsposition wird in den Folgeachsen nicht herausgefahren. Die Kopplung bleibt unverändert erhalten.
- Folgeachsen: Durch PRESETON wird nur der überlagerte Positionsanteil der Folgeachse beeinflusst.

Gantry-Verbund

Wird PRESETON auf die Führung Achse eines Gantry-Verbunds angewandt, wird die Nullpunktverschiebung auch in allen Gleichlaufachsen des Gantry-Verbunds durchgeführt.

Teilungsachsen

PRESETON kann auf Teilungsachsen angewandt werden.

Software-Endschalter, Arbeitsfeldbegrenzung, Schutzbereiche

Liegt nach einer Nullpunktverschiebung durch PRESETON die Achsposition außerhalb einer der genannten Begrenzungen, wird erst dann ein Alarm angezeigt, wenn versucht wird die Achse zu verfahren.

Satzsuchlauf mit Berechnung

PRESETON-Befehle werden während des Satzsuchlaufs aufgesammelt und mit NC-Start zum Fortsetzen des NC-Programms ausgeführt.

Positionsabhängige NC/PLC-Nahtstellensignale

Der Status der positionsabhängigen NC/PLC-Nahtstellensignale wird aufgrund der neuen Istposition neu bestimmt.

Beispiel: Festpunktpositionen

- Parametrierte Festpunktpositionen: MD30600 \$MA_FIX_POINT_POS[0...3] = <Festpunktposition 1...4>
- NC/PLC-Nahtstellensignale: DB31, ... DBX75.3 ... 5 (JOG Festpunkt anfahren: erreicht)

Steht die Achse innerhalb der Genauhalttoleranz auf einer Festpunktposition, wird das zugehörige NC/PLC-Nahtstellensignal gesetzt. Mit dem Setzen des Istwertes durch PRESETON auf einen anderen Wert ausserhalb der Genauhalttoleranz um die Festpunktposition, wird das Nahtstellensignal zurückgesetzt.

DRF-Verschiebung

Durch PRESETON wird eine DRF-Verschiebung der Achse gelöscht.

Überlagerte Bewegung \$AA_OFF

Eine überlagerte Bewegung aus einer Synchronaktion mit \$AA_OFF wird durch PRESETON nicht beeinflusst.

Online-Werkzeugkorrektur FTOC

Eine aktive Online-Werkzeugkorrektur aus einer Synchronaktion mit FTOC bleibt auch nach PRESETON weiter aktiv.

Achsspezifische Kompensationen

Achsspezifische Kompensationen bleiben nach PRESETON weiter aktiv.

Betriebsart JOG

PRESETON darf nur auf eine stehende Achsen angewandt werden.

Betriebsart JOG, Maschinenfunktion REF

PRESETON darf **nicht** angewandt werden..

3.4.2.2 Istwertsetzen ohne Verlust des Referenzierstatus (PRESETONS)

Funktion

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 Vorlaufstopp 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 Istwertes 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-Modi 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)

Inbetriebnahme

Achsspezifische Maschinendaten

Istwertsetzen ohne Verlust des Referenzierstatus (PRESETONS) muss achsspezifisch aktiviert werden:

MD30455 \$MA_MISC_FUNCTION_MASK, Bit 9 = 1

Hinweis

PRESETON deaktiviert

Mit dem Aktivieren der Funktion "Istwertsetzens **ohne** Verlust des Referenzierstatus PRESETONS" wird die Funktion "Istwertsetzen **mit** Verlust des Referenzierstatus PRESETON" deaktiviert. Beide Funktionen schließen sich gegenseitig aus.

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) Der Wert bezieht sich auf das aktive Maßsystem (inch / metrisch) Eine aktive Durchmesserprogrammierung (DIAMON) wird berücksichtigt Typ: REAL

Systemvariable

\$AC_PRESET

Die achsspezifische Systemvariable \$AC_PRESET liefert den Vektor vom Nullpunkt des aktuellen verschobenen MKS' zum Nullpunkt des ursprünglichen MKS₀ nach dem Referenzieren der Maschinenachse.

```
$AC_PRESET<Achse> = $AC_PRESET<Achse> + "aktuelle Istposition der Achse im MKS" - "PRESETONS-Istposition"
```

Mit Hilfe der Systemvariablen kann die Nullpunktverschiebung wieder rückgängig gemacht werden:

```
PRESETONS(<Achse>, $VA_IM + $AC_PRESET[<Achse>]) ; "Aktueller Istwert der Achse im MKS'" + "Verschiebungen"
```

Beispiel

Nullpunktverschiebung des MKS der Achse X um 70 Einheiten.

Die programmierte Endposition der Achse X (Kommandoachse) wird mit PRESETONS in das neue MKS transformiert.

Programmcode

```
N10 G1 X10 F5000
N20 PRESETONS(X, $AA_IM[X]+70) ; Istwert = 10 + 70 = 80 =>
                               ; $AC_PRESET = $AC_PRESET - 70
```

Randbedingungen

Achsen bei denen PRESETONS nicht angewandt werden darf

- Fahrende Positionierachsen
- Fahrende Kommandoachsen im Spindelbetrieb
- Fahrende konkurrierende Positionierachsen (FC18)
- Achsen, die an einer Transformation beteiligt sind
- Fahrende Bahnachsen
- Pendelachsen
- Achsen, bei denen eine oder mehrere der folgenden Safety Integrated Funktionen aktiv sind:
 - Freigabe "Sichere Software-Endschalter"
MD36901 \$MA_SAFE_FUNCTION_ENABLE[<SafeAxis>], Bit 1 = 1
 - Freigabe "Sichere Software-Nocken", Paar 1 ... 4, Nocke +/-
MD36901 \$MA_SAFE_FUNCTION_ENABLE[<SafeAxis>], Bit 8 ... 15 = 1
bzw.
Freigabe "Sichere Nockenspur", Nocke 1 ... 30
MD36903 \$MA_SAFE_CAM_ENABLE[<SafeAxis>], Bit 0 ... 29 = 1
- Hirth-Achsen
- Gleichlaufachsen eines Gantry-Verbunds
- Achsen für die Referenzpunktfahren aus dem Teileprogramm (G74) aktiv ist
- Slave-Achse einer Drehzahl-/Drehmomentkopplung (Master-Slave)

Geometrieachsen

- PRESETONS kann auf eine stehende Geometrieachse angewandt werden, wenn im Kanal nicht gleichzeitig eine weitere Geometrieachse verfährt.
- PRESETONS kann auf eine stehende Geometrieachse angewandt werden, auch wenn im Kanal gleichzeitig eine weitere Geometrieachse verfährt, sich diese aber im Zustand "Neutrale Achse" befindet oder als Kommandoachse verfährt.
Beispiel: Eine andere Geometrieachse (X) verfährt gleichzeitig im Zustand "**Neutrale Achse**"

Programmcode	Kommentar
N10 G0 X0 Y0	; X, Y: Geometrieachsen
N15 RELEASE(Y) ¹⁾	; Neutrale Achse
N20 PRESETONS(Y,20)	; Istposition Y im MKS = 20
N30 G0 X40	; Geometrieachse X verfährt
N40 M30	
1) Hinweis Bei einer Freigabe der Achse im Aktionsteil der Synchronaktion ist nicht gewährleistet, dass die Freigabe rechtzeitig erfolgt. N20 ID=1 WHEN 20.0 < \$AA_IM[X] DO RELEASE(Y) PRESETONS(Y,20) ; NICHT empfohlen!	

Beispiel: Eine andere Geometrieachse (X) verfährt gleichzeitig als **Kommandoachse**

Programmcode	Kommentar
N10 G0 X0 Y0	; X, Y: Geometrieachsen
N20 POS[X]=40 FA[X]=1000	; Kommandoachse X
N30 PRESETONS(Y,20)	; Istposition Y im MKS = 20
N40 M30	

PLC-kontrollierte Achsen

PRESETONS kann auf PLC-kontrollierte Achsen entsprechend ihres aktuellen Typs angewandt werden.

Spindelzustände

In der nachfolgenden Tabelle sind die Reaktionen aufgezeigt, die erfolgen, wenn PRESETONS in einer Synchronaktion auf eine Spindel angewandt wird:

PRESETONS im NC-Programm			
Spindelbetriebsart	Verfahrstatus	dem NC-Programm zugeordnet	Hauptlaufachse
Drehzahlsteuerbetrieb	in Bewegung	Alarm 22324	Alarm 22324
	steht	+	+
Positionierbetrieb SPOS	Bewegung	-	+
	steht	+	+
Positionieren über Satzgrenzen SPOSA	in Bewegung	Alarm 10610	-

PRESETONS im NC-Programm			
Spindelbetriebsart	Verfahrstatus	dem NC-Programm zugeordnet	Hauptlaufachse
Achsbetrieb	in Bewegung	-	+
	steht	+	+
+: möglich -: nicht möglich			

Achskopplungen

- Leitachsen: Die durch PRESETONS verursachte sprungförmige Änderung der Leitachsposition wird in den Folgeachsen nicht herausgefahren. Die Kopplung bleibt unverändert erhalten.
- Folgeachsen: Durch PRESETONS wird nur der überlagerte Positionsanteil der Folgeachse beeinflusst.

Gantry-Verbund

Wird PRESETONS auf die Führungachse eines Gantry-Verbunds angewandt, wird die Nullpunktverschiebung auch in allen Gleichlaufachsen des Gantry-Verbunds durchgeführt.

Teilungsachsen

PRESETONS kann auf Teilungsachsen angewandt werden.

Software-Endschalter, Arbeitsfeldbegrenzung, Schutzbereiche

Liegt nach einer Nullpunktverschiebung durch PRESETONS die Achsposition außerhalb einer der genannten Begrenzungen, wird erst dann ein Alarm angezeigt, wenn versucht wird die Achse zu verfahren.

Satzsuchlauf mit Berechnung

PRESETONS-Befehle werden während des Suchlaufs aufgesammelt und mit NC-Start zum Fortsetzen des NC-Programms ausgeführt.

Positionsabhängige NC/PLC-Nahtstellensignale

Der Status der positionsabhängigen NC/PLC-Nahtstellensignale wird aufgrund der neuen Istposition neu bestimmt.

Beispiel: Festpunktpositionen

- Parametrierte Festpunktpositionen: MD30600 \$MA_FIX_POINT_POS[0...3] = <Festpunktposition 1...4>
- NC/PLC-Nahtstellensignale: DB31, ... DBX75.3 ... 5 (JOG Festpunkt anfahren: erreicht)

Steht die Achse innerhalb der Genauhalttoleranz auf einer Festpunktposition, wird das zugehörige NC/PLC-Nahtstellensignal gesetzt. Mit dem Setzen des Istwertes durch PRESETONS auf einen anderen Wert außerhalb der Genauhalttoleranz um die Festpunktposition, wird das Nahtstellensignal zurückgesetzt.

DRF-Verschiebung

Durch PRESETONS wird eine DRF-Verschiebung der Achse gelöscht.

Überlagerte Bewegung \$AA_OFF

Eine überlagerte Bewegung aus einer Synchronaktion mit \$AA_OFF wird durch PRESETONS nicht beeinflusst.

Online-Werkzeugkorrektur FTOC

Eine aktive Online-Werkzeugkorrektur aus einer Synchronaktion mit FTOC bleibt auch nach PRESETONS weiter aktiv.

Achsspezifische Kompensationen

Achsspezifische Kompensationen bleiben nach PRESETONS weiter aktiv.

Betriebsart JOG

PRESETONS darf nur auf eine stehende Achsen angewandt werden.

Betriebsart JOG, Maschinenfunktion REF

PRESETONS darf **nicht** angewandt werden..

Synchronaktionen

Sind in einer Synchronaktion für eine Achse mehrere PRESETONS-Anweisungen enthalten, wird nur die letzte in der Reihenfolge von links nach rechts stehende Anweisung ausgeführt.

Beispiel:

Programmcode

```
N10 ID=1 WHEN TRUE DO PRESETONS (X, 40) PRESETONS (X, 39) PRESETONS (X, 38)
; Gleichbedeutend mit:
N10 ID=1 WHEN TRUE DO PRESETONS (X, 38)
```

3.4.3 Basiskoordinatensystem (BKS)

Basiskoordinatensystem (BKS)

Das Basiskoordinatensystem (BKS) besteht aus drei rechtwinklig angeordneten Achsen (Geometrieachsen), sowie aus weiteren Achsen (Zusatzachsen) ohne geometrischen Zusammenhang.

WZ-Maschinen ohne kinematische Transformation

Das BKS und das MKS fallen immer dann zusammen, wenn das BKS ohne kinematische Transformation (z. B. TRANSMIT / Stirnflächentransformation, 5-Achstransformation und max. drei Maschinenachsen) auf das MKS abgebildet werden kann.

Bei diesen Maschinen können Maschinenachsen und Geometrieachsen den gleichen Namen haben.

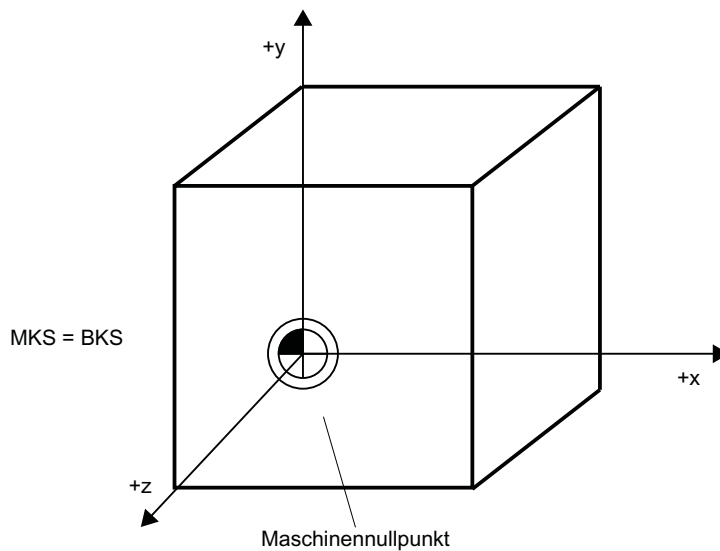


Bild 3-15 MKS=BKS ohne kinematische Transformation

WZ-Maschinen mit kinematischer Transformation

Das BKS und das MKS fallen nicht zusammen, wenn das BKS mit kinematischer Transformation (z. B. TRANSMIT / Stirnflächen-Transformation, 5-Achstransformation oder mehr als drei Achsen) auf das MKS abgebildet wird.

Bei diesen Maschinen müssen Maschinenachsen und Geometrieachsen unterschiedliche Namen haben.

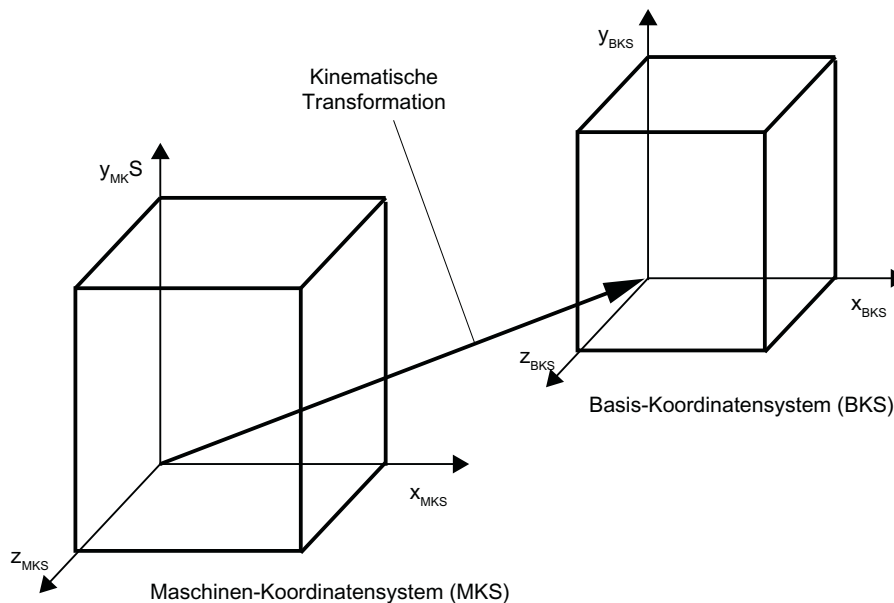


Bild 3-16 Kinematische Transformation zwischen MKS und BKS

Maschinenkinematik

Das Werkstück wird immer in einem zwei- oder dreidimensionalen rechtwinkligen Koordinatensystem (WKS) programmiert. Zur Fertigung dieser Werkstücke werden aber immer häufiger Werkzeugmaschinen mit Rundachsen oder nicht rechtwinklig angeordneten Linearachsen eingesetzt. Zur Abbildung der im WKS programmierten Koordinaten (rechtwinklig) in reale Maschinenachsbewegungen dient die kinematische Transformation.

Weitere Informationen

Funktionshandbuch Transformationen; Mehrachstransformationen

3.4.4 Basis-Nullpunktsystem (BNS)

Basis-Nullpunktsystem (BNS)

Das Basis-Nullpunktsystem (BNS) ergibt sich aus dem Basis-Koordinatensystem durch die Basisverschiebung.

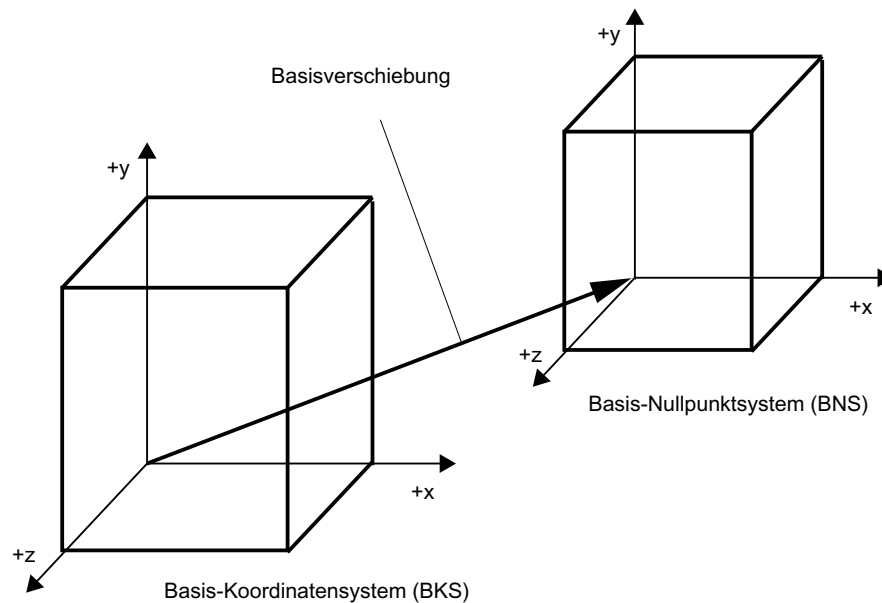


Bild 3-17 Basisverschiebung zwischen BKS und BNS

Basisverschiebung

Die Basisverschiebung beschreibt die Koordinatentransformation zwischen dem BKS und BNS. Mit ihr kann z. B. der Paletten-Nullpunkt festgelegt werden.

Die Basisverschiebung setzt sich zusammen aus:

- Externe Nullpunktverschiebung
- DRF-Verschiebung

- Überlagerte Bewegung
- Verkettete Systemframes
- Verkettete Basisframes

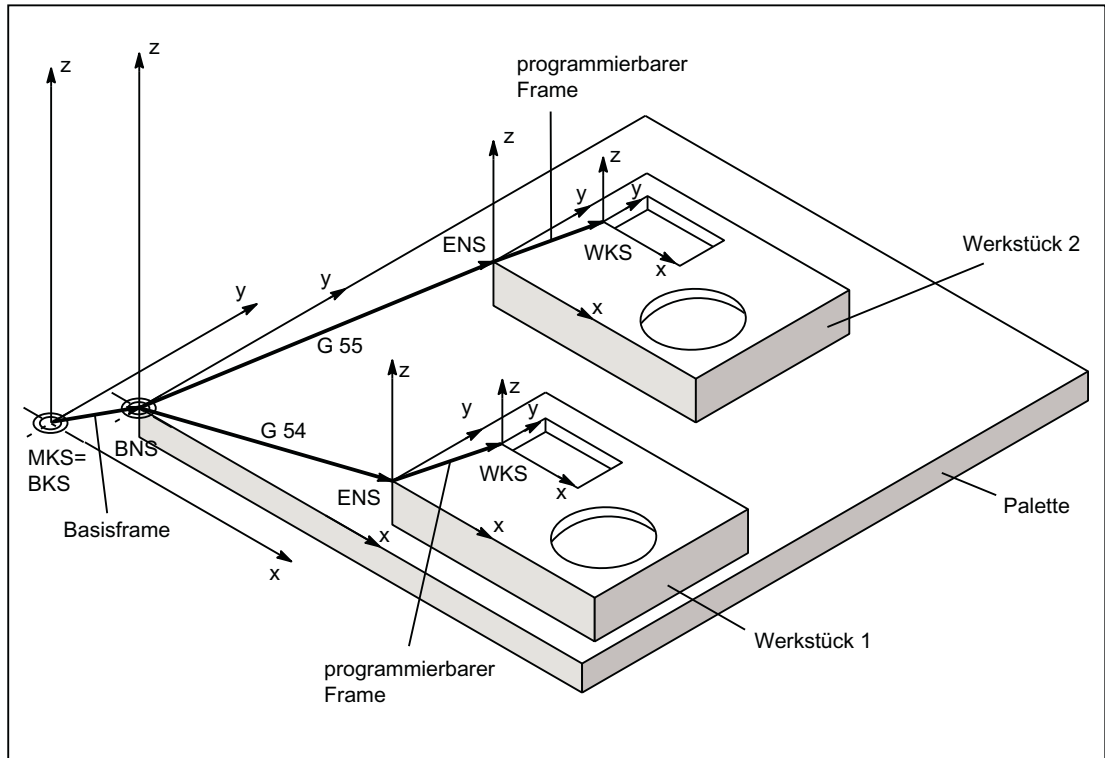


Bild 3-18 Beispiel für Anwendung der Basisverschiebung

Es gilt:

- Der Anwender kann die Basisverschiebung aus dem Teileprogramm, der Bedienung und von der PLC verändern.
- Soll die Basisverschiebung sofort wirksam werden, so kann über PLC mit FC9 ein ASUP gestartet werden, der den entsprechenden G-Befehl ausführt.

Hinweis

Empfehlung an den Maschinenhersteller

Verwenden Sie für eigene Anwendungen Verschiebungen ab der 3. Basisverschiebung.

Die 1. und 2. Basisverschiebung sind für das "Istwertsetzen" und die "Externe Nullpunktverschiebung" vorgesehen.

3.4.5 Einstellbares Nullpunktsystem (ENS)

Einstellbare Nullpunktsystem (ENS)

Das "Einstellbare Nullpunktsystem" (ENS) ist das Werkstückkoordinatensystem WKS mit programmierbarem FRAME (gesehen aus der Perspektive WKS). Der Werkstücknullpunkt wird durch die einstellbaren FRAMES G54...G599 festgelegt.

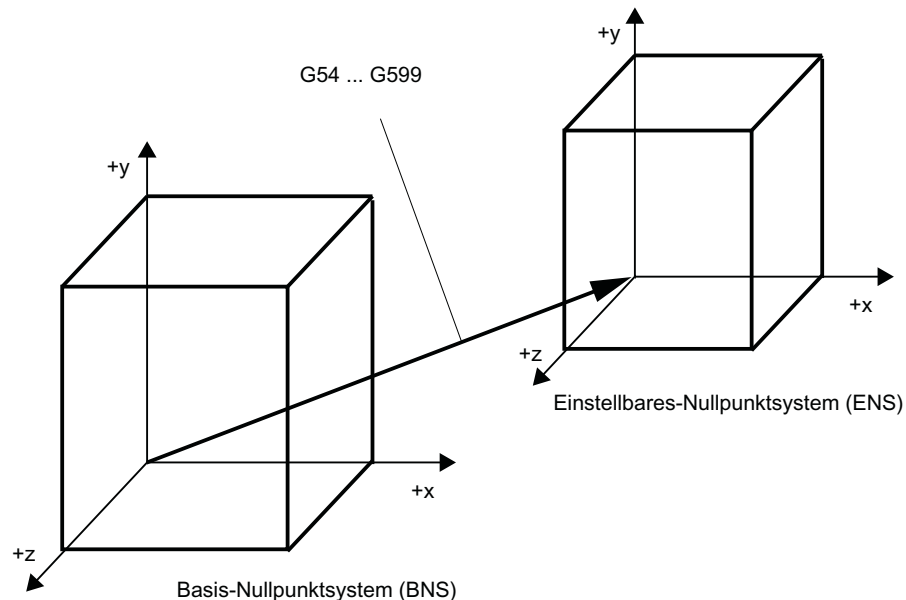


Bild 3-19 Einstellbarer FRAME G54 ... G599 zwischen BNS und ENS

Vom "Einstellbaren-Nullpunktsystem" aus wirken programmierbare Verschiebungen. Alle programmierbaren Verschiebungen beziehen sich auf das "Einstellbare-Nullpunktsystem".

WKS-Istwertanzeige im WKS oder ENS

Auf der HMI-Bedienoberfläche können die Istwerte der Achsen im Maschinenkoordinatensystem (MKS) oder im WKS angezeigt werden. Bei Anzeige im WKS können die Istwerte auch bezogen auf das ENS angezeigt werden. Die entsprechende Parametrierung erfolgt über das Maschinendatum:

MD9424 \$MM_MA_COORDINATE_SYSTEM (Koordinatensystem für Istwertanzeige)

Wert	Bedeutung
0	Istwertanzeige bezogen auf das WKS
1	Istwertanzeige bezogen auf das ENS

Hinweis

Anzeige des aktuellen Koordinatensystems

Auf der HMI-Bedienoberfläche wird auch bei aktiver "Istwertanzeige bezogen auf das ENS" das WKS als Koordinatensystem angezeigt, auf das sich die Istwertanzeige bezieht.

Beispiel

Istwertanzeige bezogen auf das WKS bzw. ENS

Code (Ausschnitt)	Istwertanzeige: Achse X (WKS)	Istwertanzeige: Achse X (ENS)
N10 X100	100	100
N20 X0	0	0
N30 \$P_PFRAME = CTRANS(X,10)	0	10
N40 X100	100	110

3.4.6 Werkstückkoordinatensystem (WKS)

Werkstückkoordinatensystem WKS

Das Werkstückkoordinatensystem (WKS) ist die Basis für die Programmierung.

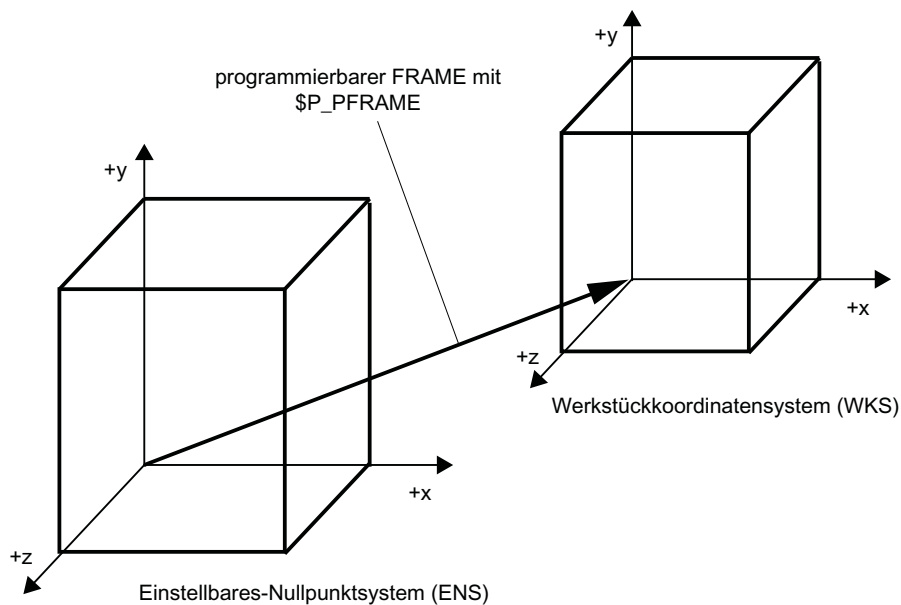
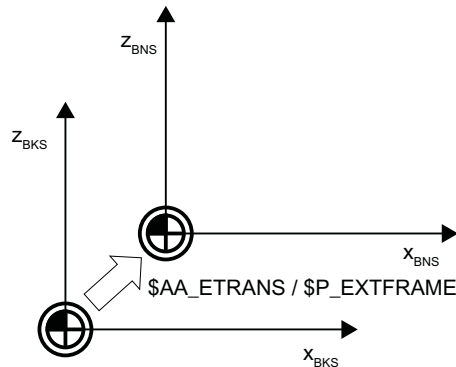


Bild 3-20 Programmierbarer FRAME zwischen ENS und WKS

3.4.7 Additive Korrekturen

3.4.7.1 Externe Nullpunktverschiebungen

Die Externen Nullpunktverschiebung ist eine lineare Verschiebung zwischen Basiskoordinatensystem (BKS) und Basisnullpunktsystem (BNS).



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 Verfahrssatz: 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

Aktivierung der Externe Nullpunktverschiebung:

DB31, ... DBX3.0 = 0 → 1 ⇒ \$P_EXTFRAME[<Achse>] = \$P_EXTFR[<Achse>] = \$AA_ETRANS[<Achse>]

Unterdrückung: Externe Nullpunktverschiebung

- Die Befehl SUPA unterdrückt die Externe Nullpunktverschiebung für die Dauer der Bearbeitung des Satzes.
- Aktive Externe Nullpunktverschiebungen werden für die Dauer des Referenzpunktfahrens durch Befehl G74 (Referenzpunktfahren), sowie die entsprechende Bedienung in der Betriebsart Referenzpunktfahren, unterdrücken die Externe Nullpunktverschiebung für die Dauer des Referenzpunktfahrens.

- Bei G74, d. h. Betriebsart "Automatik" oder "MDA", wird die zuvor aktive Externe Nullpunktverschiebung mit der nächsten Verfahrbewegung im Satz wieder automatisch aktiv.
- Nach einem Betriebsartenwechsel aus der Betriebsart Referenzpunktfahren muss für die referenzierten Achsen das NC/PLC-Nahtstellensignal zur erneuten Aktivierung gesetzt werden.

3.4.7.2 DRF-Verschiebung

Die DRF-Verschiebung ermöglicht das Einstellen einer additiven inkrementellen Nullpunktverschiebung für Geometrie- und Zusatzachsen im Basiskoordinatensystem **über Handrad**.

Systemvariable

Die DRF-Verschiebung kann über die achsspezifische Systemvariable gelesen werden:

\$AC_DRF[<Achse>]

Weitere Informationen

Funktionshandbuch Achsen und Spindeln; Manuelles Verfahren

3.4.7.3 Reset-Verhalten

Das Reset- und Power On-Verhalten des aktiven Frames \$P_EXTFRAME und des Datenhaltungsframes \$P_EXTFR kann über Maschinendaten eingestellt werden:

Maschinendaten

- Das Reset-Verhalten bezüglich des im Kanal aktiven Systemframes der Externen Nullpunktverschiebung \$P_EXTFRAME wird über folgendes Maschinendatum eingestellt:
- MD24006 \$MC_CHSFRAME_RESET_MASK, Bit1 = <Wert>

Wert	Bedeutung
	Das aktive Systemframe der Externen Nullpunktverschiebung ist nach Kanal- / Programmende-Reset:
0	nicht aktiv
1	aktiv

- Das Power On-Verhalten bezüglich des kanalspezifischen Systemframes der Externen Nullpunktverschiebung \$P_EXTFR der Datenhaltung wird über folgendes Maschinendatum eingestellt:
MD24008 \$MC_CHSFRAME_POWERON_MASK, Bit1 = <Wert>

Wert	Bedeutung
	Das Systemframe \$P_EXTFR der Externen Nullpunktverschiebung wird bei Power On
0	nicht gelöscht
1	gelöscht

3.4.8 Achsspezifische Überlagerung (\$AA_OFF)

3.4.8.1 Funktion

Über die achsspezifische Systemvariable \$AA_OFF[<Achse>] kann z.B. in einer Synchronaktion für die angegebene Achse eine absolute Position oder ein inkrementeller Weg vorgegeben. Die daraus resultierende Verfahrbewegung wird dann parallel zu den Verfahrbewegungen des Kanals der Achse ausgeführt.

\$AA_OFF[<Achse>] = <Wert>

3.4.8.2 Inbetriebnahme

Maschinendaten

Parametrierung der überlagerten Bewegung

Über das achsspezifische Maschinendatum werden folgende Einstellungen für die überlagerte Bewegung vorgenommen:

36750 \$MA_AA_OFF_MODE, Bit<n> = <Wert>

Bit	Wert	Bedeutung
0	0	Interpretation des Wertes von \$AA_OFF als absolute Position
	1	Interpretation des Wertes von \$AA_OFF als inkrementellen Weg
1	0	Die überlagerte Bewegung wird bei Kanal-Reset abgewählt.
	1	Die überlagerte Bewegung bleibt über Kanal-Reset hinaus erhalten.
2	0	In der Betriebsart JOG wird eine überlagerte Bewegung nicht herausgefahren.
	1	In der Betriebsart JOG wird eine überlagerte Bewegung herausgefahren.
3	0	Eine überlagerte Bewegung wird bei NC-Stopp unterbrochen.
	1	Eine überlagerte Bewegung wird bei NC-Stopp nicht unterbrochen.

Systemvariable

Integrierter Weg der Achsüberlagerung

Über die achsspezifische Systemvariable kann der integrierte Wert der überlagerten Bewegung gelesen werden.

<Wert> = \$AA_OFF_VAL[<Achse>]

3.4.8.3 Programmierung: Überlagerungen achsspezifisch abwählen (CORROF)

Mit der Prozedur CORROF werden folgende achsspezifischen Überlagerungen gelöscht:

- Über Handradverfahren eingestellte additive Nullpunktverschiebung (DRF-Verschiebungen)
- Über die Systemvariable \$AA_OFF programmierte Positionsoffsets

Durch das Löschen eines Überlagerungswertes wird ein Vorlaufstopp ausgelöst und der Positionsanteil der abgewählten überlagerten Bewegung in die Position im Basiskoordinatensystem übernommen. Dabei wird keine Achse verfahren.

Der über die Systemvariable \$AA_IM (Aktueller **MKS**-Sollwert der Achse) lesbare Positionswert im **Maschinen**koordinatensystem **ändert sich nicht**.

Der über die Systemvariable \$AA_IW (Aktueller **WKS**-Sollwert der Achse) lesbare Positionswert im **Werkstück**koordinatensystem **ändert sich**, da er nun den abgewählten Anteil der überlagerten Bewegung enthält.

Hinweis

In einem **NC-Programm** darf `CORROF` programmiert werden.

In einer **Synchronaktion** darf `CORROF` **nicht** programmiert werden.

Syntax

```
CORROF(<Axis>,"<String>"[,<Axis>,"<String>"])
```

Bedeutung

CORROF:	Prozedur zur Abwahl folgender Verschiebungen bzw. Überlagerungen einer Achse:	
	<ul style="list-style-type: none"> • DRF-Verschiebung • Positionsoffsets (\$AA_OFF) 	
	Wirksamkeit:	modal
<Axis>:	Achsenbezeichner (Kanal-, Geometrie- oder Maschinenachsbezeichner)	
	Datentyp:	AXIS
<String>:	Zeichenkette zur Definition der Überlagerungsart	
	Datentyp:	BOOL
	Wert	Bedeutung
	DRF	DRF-Verschiebung
	AA_OFF	Positionsoffset (\$AA_OFF)

Beispiele

Beispiel 1: Achsspezifische Abwahl einer DRF-Verschiebung (1)

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

Programmcode	Kommentar
N10 CORROF(X,"DRF")	; CORROF wirkt hier wie DRFOF.
...	

Beispiel 2: Achsspezifische Abwahl einer DRF-Verschiebung (2)

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X- und in der Y-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

Programmcode	Kommentar
	; Nur die DRF-Verschiebung der X-Achse wird abgewählt, die DRF-Verschiebung der Y-Achse bleibt erhalten.
	; Bei DRFOF wären beide Verschiebungen abgewählt worden.
N10	CORROF(X,"DRF")
...	

Beispiel 3: Achsspezifische Abwahl eines \$AA_OFF-Positionsoffsets

Programmcode	Kommentar
	; Für die X-Achse wird ein Positionsoffset == 10 interpoliert.
N10	WHEN TRUE DO \$AA_OFF[X]=10 G4 F5
...	
	; Der Positionsoffset der X-Achse wird abgewählt: \$AA_OFF[X]=0
	; Die X-Achse wird nicht verfahren.
	; Zur aktuellen Position der X-Achse wird der Positionsoffset hinzugerechnet.
N80	CORROF(X,"AA_OFF")
...	

Beispiel 4: Achsspezifische Abwahl einer DRF-Verschiebung und eines \$AA_OFF-Positionsoffsets (1)

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

Programmcode	Kommentar
	; Für die X-Achse wird ein Positionsoffset von 10 interpoliert.
N10	WHEN TRUE DO \$AA_OFF[X]=10 G4 F5
...	
	; Nur die DRF-Verschiebung und der Positionsoffset der X-Achse wird abgewählt.
	; Die DRF-Verschiebung der Y-Achse bleibt erhalten.
N70	CORROF(X,"DRF",X,"AA_OFF")
...	

Beispiel 5: Achsspezifische Abwahl einer DRF-Verschiebung und eines \$AA_OFF-Positionsoffsets (2)

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X-Achse und in der Y-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

Programmcode	Kommentar
	; Für die X-Achse wird ein Positionsoffset == 10 interpoliert.
N10	WHEN TRUE DO \$AA_OFF[X]=10 G4 F5
...	

Programmcode	Kommentar
	; Die DRF-Verschiebung der Y-Achse und der Positionsoffset der X-Achse werden abgewählt.
	; Die DRF-Verschiebung der X-Achse bleibt erhalten.
N70	CORROF(Y, "DRF", X, "AA_OFF")
...	

Weitere Informationen

\$AA_OFF_VAL

Nach der Abwahl des Positionsoffsets aufgrund von \$AA_OFF ist die Systemvariable \$AA_OFF_VAL (Integrierter Weg der Achsüberlagerung) der entsprechenden Achse gleich Null.

\$AA_OFF in der Betriebsart JOG

Auch in der Betriebsart JOG findet bei einer Änderung von \$AA_OFF eine Interpolation des Positionsoffsets als überlagerte Bewegung statt, wenn die Freischaltung dieser Funktion über das Maschinendatum MD36750 \$MA_AA_OFF_MODE erfolgt ist.

\$AA_OFF in Synchronaktion

Ist bei der Abwahl des Positionsoffsets über CORROF (<Achse>, "AA_OFF") eine Synchronaktion aktiv, die \$AA_OFF sofort wieder setzt (DO \$AA_OFF[<Achse>]=<Wert>), dann wird \$AA_OFF abgewählt und nicht wieder gesetzt und der Alarm 21660 angezeigt. Wird die Synchronaktion jedoch später aktiv, z. B. im Satz nach CORROF, dann wird \$AA_OFF gesetzt und ein Positionsoffset interpoliert.

Automatischer Kanalachstausch

Falls eine Achse, für die ein CORROF programmiert wurde, in einem anderen Kanal aktiv ist, dann wird sie mit Achstausch in den Kanal geholt (Voraussetzung: MD30552 \$MA_AUTO_GET_TYPE > 0) und dann der Positionsoffset und/oder die DRF-Verschiebung abgewählt.

3.5 Frames

3.5.1 Frame-Arten

Ein Frame ist eine Datenstruktur die Werte für Verschiebung (TRANS), Feinverschiebung (FINE), Drehung (ROT), Spiegelung (MIRROR) und Skalierung (SCALE) für Achsen enthält.

Bei Aktivierung des Frames wird anhand der Frame-Werte über eine definierte Rechenvorschrift eine statische Koordinatentransformation für die im Frame enthaltenen Achsen durchgeführt.

Achsspezifischer Frame

Ein achsspezifischer Frame enthält die Frame-Werte einer Achse.

Beispielhafte Datenstruktur eines achsspezifischen Frames für Geometrieachse X:

Achse	TRANS	FINE	ROT	MIRROR	SCALE
X	10.0	0.1	0.0	0	1

Kanalspezifischer Frame

Ein kanalspezifischer Frame enthält die Frame-Werte für alle Kanalachsen (Geometrie-, Zusatz- und Maschinenachsen).

Drehungen (ROT) werden nur bei Geometrieachsen eingerechnet.

Ein kanalspezifischer Frame wirkt nur in dem Kanal in dem der Frame definiert ist.

Beispielhafte Datenstruktur eines kanalspezifischen Frames:

- Geometrieachsen: X, Y, Z
- Zusatzachsen: A
- Maschinenachsen: AX1

Achse	TRANS	FINE	ROT	MIRROR	SCALE
X	10.0	0.1	0.0	0	1
Y	0.0	0.0	0.0	1	1
Z	0.0	0.0	45.0	0	1
A	2.0	0.1	0.0	0	2
AX1	0.0	0.0	0.0	0	0

Globaler Frame

Ein globaler Frame enthält die Frame-Werte für alle Maschinenachsen.

Ein globaler Frame wirkt in allen Kanälen der NC.

Beispielhafte Datenstruktur eines globalen Frames:

- Maschinenachsen: AX1, ... AX5

Achse	TRANS	FINE	ROT	MIRROR	SCALE
AX1	10.0	0.1	-	0	1
AX2	0.0	0.0	-	1	1
AX3	0.0	0.0	-	0	1
AX4	2.0	0.1	-	0	2
AX5	0.0	0.0	-	1	1

3.5.2 Frame-Komponenten

3.5.2.1 Translation

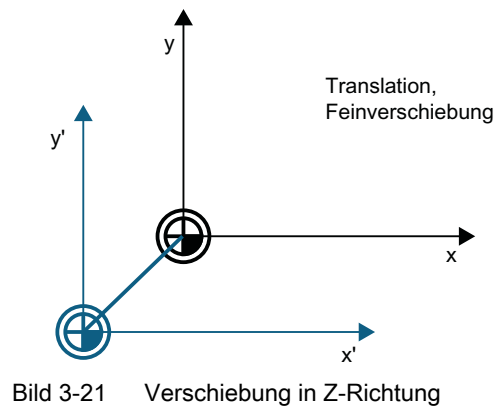
Programmierung

Die Programmierung der Translation bzw. Grobverschiebung kann über folgende Befehle erfolgen:

- Beispiel Datenhaltungsframes \$P_UIFR
 - Gesamtframe: $\$P_UIFR[\langle n \rangle] = CTRANS(\langle K1 \rangle, \langle V1 \rangle [, \langle K2 \rangle, \langle V2 \rangle] [, \langle K3 \rangle, \langle V3 \rangle])$
mit $K_m =$ Koordinate x, y oder z und $V_m =$ Verschiebung m
 - Frame-Komponente: $\$P_UIFR[\langle n \rangle, \langle K \rangle, TR] = \langle V \rangle$
mit $K =$ Koordinate x, y oder z und $V =$ Verschiebung
- Beispiel Programmierbarer Frame
 - $TRANS \langle K1 \rangle \langle V1 \rangle [\langle K2 \rangle \langle V2 \rangle] [\langle K3 \rangle \langle V3 \rangle]$
mit $K_m =$ Koordinate x, y oder z und $V_m =$ Verschiebung m

Programmbeispiele:

Programmcode	Bemerkung
$\$P_UIFR[1] = CTRANS(X,10,Y,10)$	Gesamtframe
$\$P_UIFR[1,X,TR] = 10$	Frame-Komponente
$TRANS X=10 Y=10$	Programmierbarer Frame



3.5.2.2 Feinverschiebung

Parametrierung

Die Freigabe der Feinverschiebung erfolgt über das Maschinendatum:

MD18600 \$MN_MM_FRAME_FINE_TRANS = <Wert>

Wert	Bedeutung
0	Feinverschiebung kann nicht eingegeben bzw. nicht programmiert werden.
1	Feinverschiebung für einstellbare Frames, Basisframes und das Programmierbare Frame ist von der Bedienung oder über Programm möglich.

Programmierung

Die Programmierung der Translation bzw. Grobverschiebung kann über folgende Befehle erfolgen:

- Beispiel Datenhaltungsframes \$P_UIFR
 - Gesamtframe: \$P_UIFR[<n>] = CFINE (<K1>, <V1> [, <K2>, <V2>] [, <K3>, <V3>])
mit Km = Koordinate x, y oder z und Vm = Verschiebung m
 - Frame-Komponente: \$P_UIFR[<n>, <K>, FI] = <V>
mit K = Koordinate x, y oder z und V = Verschiebung
- Beispiel Programmierbarer Frame
 - TRANS <K1> <V1> [, <K2> <V2>] [, <K3> <V3>]
mit Km = Koordinate x, y oder z und Vm = Verschiebung m

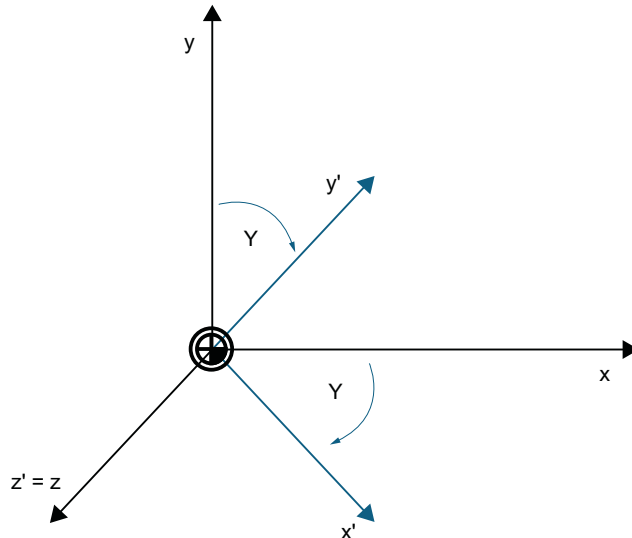
Programmierbeispiele:

Programmcode	Bemerkung
\$P_UIFR[1] = CTRANS (X, 10, Y, 10)	Gesamtframe
\$P_UIFR[1, X, TR] = 10	Frame-Komponente
TRANS X=10 Y=10	Programmierbarer Frame

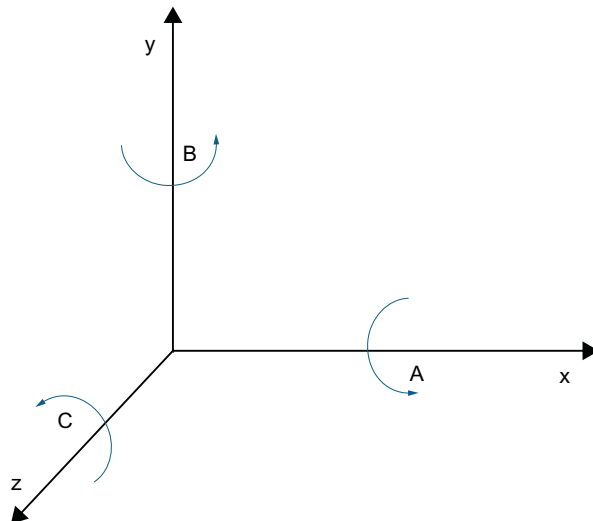
3.5.2.3 Drehung: Übersicht (nur Geometrieachsen)

Funktion

Die Drehrichtung um die Koordinatenachsen wird durch ein rechtshändiges, rechtwinkliges Koordinatensystem mit den Achsen x , y und z bestimmt. Der Drehsinn der Drehung ist positiv, wenn die Drehbewegung bei Blick in die positive Richtung der Koordinatenachse im Uhrzeigersinn erfolgt. A , B und C bezeichnen Drehungen, deren Achsen parallel zu den Koordinatenachsen sind.



Das folgende Bild zeigt die neue Lage des Koordinatensystems x' , y' und z' nach der Drehung um z mit $\gamma = -45^\circ$



Parametrierung der Drehreihenfolge

Über das folgende Maschinendatum wird eingestellt, um welche Koordinatenachsen und in welcher Reihenfolge die Drehungen ausgeführt werden, wenn mehr als ein Drehwinkel programmiert ist:

MD10600 \$MN_FRAME_ANGLE_INPUT_MODE = <Wert>

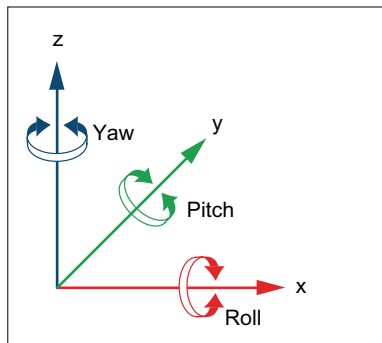
Wert	Bedeutung
1	Euler-Winkel in zy'x"-Konvention (RPY-Winkel)
2	Euler-Winkel in zx'z"-Konvention

Hinweis

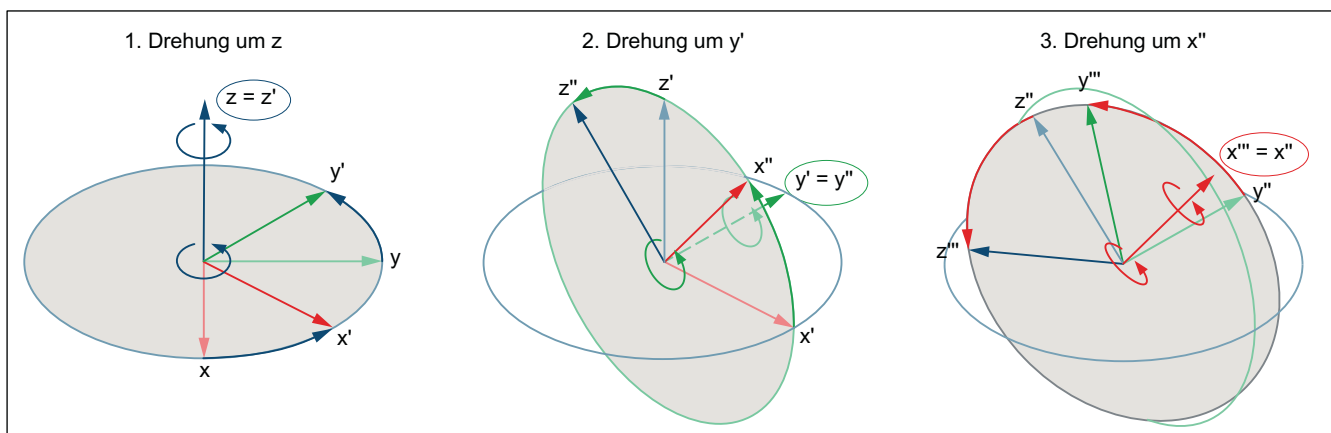
Aus historischen Gründen ist die Möglichkeit der Verwendung von Euler-Winkeln in zx'z"-Konvention vorhanden. Es wird aber dringend empfohlen ausschließlich Euler-Winkel in zy'x"-Konvention (RPY-Winkel) (siehe Kapitel "Drehung mit Euler-Winkeln: ZY'X"-Konvention (RPY-Winkel) (Seite 312)") zu verwenden.

3.5.2.4 Drehung mit Euler-Winkeln: ZY'X"-Konvention (RPY-Winkel)

Euler-Winkel in ZY'X"-Konvention werden auch als RPY-Winkel bezeichnet. RPY ist abgeleitet aus dem Englischen Roll (Rollen), Pitch (Nicken), und Yaw (Gieren):



Bei ZY'X"-Konvention erfolgen die Drehungen in folgender Reihenfolge:



Wertebereich

Bei RPY-Winkeln können programmierte Werte nur innerhalb folgender Wertebereiche eindeutig zurückgerechnet werden:

$$\begin{array}{rclcl} -180 & \leq & \mathbf{x} & \leq & 180 \\ -90 & < & \mathbf{y} & < & 90 \\ -180 & \leq & \mathbf{z} & \leq & 180 \end{array}$$

Programmierung: Schreiben aller Drehkomponenten

Bei der Programmierung der Drehkomponenten eines Frames mittels CROT, ROT oder AROT werden immer alle Drehkomponenten geschrieben. Nicht explizit programmierte Drehkomponenten werden mit dem Wert 0° belegt.

Syntax

```
<Frame> = CROT([<1. GAx>, <Winkel>], [<2. GAx>, <Winkel>], [<3. GAx>, <Winkel>])
```

```
ROT [<1. GAx> <Winkel>] [<2. GAx> <Winkel>] [<3. GAx> <Winkel>]
```

```
AROT [<1. GAx> <Winkel>] [<2. GAx> <Winkel>] [<3. GAx> <Winkel>]
```

Bedeutung

CROT:	Drehung absolut	
<Frame>:	Beliebiger aktiver oder Datenhaltungsframe	
ROT:	Drehung absolut Bezugsframe: Programmierbarer Frame \$P_PFRAME, Bezugspunkt: Nullpunkt des aktuellen mit G54 ... G57, G505 ... G599 eingestellten Werkstückkoordinatensystems	
AROT:	Drehung additiv Bezugsframe: Programmierbarer Frame \$P_PFRAME, Bezugspunkt: Nullpunkt des aktuellen mit G54 ... G57, G505 ... G599 eingestellten Werkstückkoordinatensystems	
<n-te GAx>:	Name der n-ten Geometrieachse um die um den angegebenen Winkel gedreht werden soll. Für nicht programmierte Geometrieachse wird als Drehwinkel implizit der Wert 0° gesetzt.	
	Zuordnung von Geometrieachse zu Drehachse:	
	Geometrieachse	Drehachse
	1. Geometrieachse	x''
	2. Geometrieachse	y'
	3. Geometrieachse	z
<Winkel>:	Winkelangabe in Grad	
[...]:	Die Angaben in eckigen Klammern sind optional.	

Programmierung: Schreiben einer Drehkomponente

Bei der expliziten Programmierung einer Drehkomponente eines Frames wird nur die programmierte Drehkomponente geschrieben. Nicht programmierte Drehkomponenten bleiben unverändert.

Syntax

<Frame> [<Index>, <GAx>, RT] = <Winkel>

Bedeutung

<Frame>:	Beliebiger aktiver Frame oder Datenhaltungsframe
<Index>:	Feldindex des Frames, z.B. \$P_UIFR[0 ... n]
<GAx>:	Name der Geometrieachse um die um den angegebenen Winkel gedreht werden soll.
RT:	Schlüsselwort für Drehung "RoTation"
<Winkel>	Winkelangabe in Grad

Zurücklesen der Drehkomponenten

Im Allgemeinen erhält man beim Zurücklesen der Drehkomponenten eines Frames die gleichen Werte wie die, welche programmiert wurden:

Programmiert	Werte beim Zurücklesen		
	x, RT	y, RT	z, RT
<Frame> = CROT (X, 45, Y, 30, Z, -20)	45	30	-20

Werte außerhalb des Wertebereichs

Programmierte Werte außerhalb eines Wertebereichs werden auf die Bereichsgrenzen abgebildet:

Programmiert	Werte beim Zurücklesen		
	x, RT	y, RT	z, RT
<Frame> = CROT (X, 190, Y, 0, Z, -200)	-170	0	160

Hinweis

Es wird empfohlen, beim Schreiben der Drehkomponenten des Frames die angegebenen Wertebereiche einzuhalten, um beim Rücklesen der Drehkomponenten wieder die gleichen Werte zu erhalten.

Kardanische Blockade (engl. Gimbal-Lock)

Kardanische Blockade (engl. Gimbal-Lock) bezeichnet ein geometrisches Problem, bei dem die Drehkomponenten nicht mehr eindeutig aus dem Ortsvektor zurückgerechnet werden können. Die Kardanische Blockade tritt bei RPY-Winkeln bei einer Winkelstellung der Drehkomponente y = 90° auf. In diesem Fall werden von der Steuerung die Drehkomponenten nach dem Schreiben so umgerechnet, dass gilt:

- Drehkomponente z = Drehkomponente z - Drehkomponente x
- Drehkomponente x = 0°
- Drehkomponente y = 90°

Programmiert	Werte beim Zurücklesen		
	x, RT	y, RT	z, RT
<Frame> = CROT (X, 30, Y, 90, Z, 40)	0	90	40 - 30 = 10

 **VORSICHT**

Unterschiedliche Werte beim Zurücklesen der Drehkomponente z

Aufgrund des unterschiedlichen Umrechnungszeitpunktes können nach dem Schreiben des **Gesamtframes** bzw. dem Schreiben **einzelner Drehkomponenten** eines **Datenhaltungsframes** und dem Schreiben **einzelner Drehkomponenten** eines **aktiven Frames** für die Drehkomponente z unterschiedliche Werte zurückgelesen werden.

Unterschiede beim Schreiben von Gesamtframe und Framekomponenten

Beim Schreiben der Drehkomponenten eines Frames sind zwei Fälle zu unterscheiden:

1. Schreiben des Gesamtframes: $\langle \text{Frame} \rangle = \text{CROT}(X, a, Y, b, Z, c)$
Beim Schreiben des Gesamtframes, erfolgt die Umrechnung sofort zum Zeitpunkt des Schreibens.
2. Schreiben einzelner Drehkomponenten z.B. Drehung um X: $\langle \text{Frame} \rangle[0, X, \text{RT}] = \langle a \rangle$
Beim Schreiben einzelner Drehkomponenten, erfolgt die Umrechnung abhängig vom Speicherort des Frames:
 - Datenhaltungsframes
Bei Datenhaltungsframes erfolgt die Umrechnung zum Zeitpunkt des Aktivierens des Frames aufgrund der bis dahin geschriebenen Drehkomponenten. Bezüglich der Umrechnung der Drehkomponenten verhält sich somit ein Datenhaltungsframe nach dem Schreiben einzelner Drehkomponenten wie beim Schreiben des Gesamtframes.
 - Aktive Frames
Bei aktiven Frames erfolgt die Umrechnung sofort zum Zeitpunkt des Schreibens der Drehkomponente.

Beispiel

- Schreiben des Gesamtframes
Die **Umrechnung** erfolgt in **jedem Satz** nach dem Schreiben des **Gesamtframes**.

Programmiert	Werte beim Zurücklesen		
	x, RT	y, RT	z, RT
N10 <Frame> = CROT(X,0,Y,90,Z,90)	0	90	90
N20 <Frame> = CROT(X,90,Y,90)	0	90	-90 ¹⁾
N30 <Frame> = CROT(X,90,Y,90,Z,90)	0	90	0 ¹⁾
1) unterschiedliche Werte gegenüber dem Schreiben einzelner Drehkomponenten eines aktiven Frames			

- Schreiben einzelner Drehkomponenten eines **Datenhaltungsframes**
Die **Umrechnung** erfolgt **bei Aktivierung des Datenhaltungsframes**. Im Beispiel zu einem beliebigen Zeitpunkt nach N30.

Programmiert	Werte beim Zurücklesen		
	x, RT	y, RT	z, RT
N10 <Datenhaltungsframe>[0,X,RT] = 0 N20 <Datenhaltungsframe>[0,Y,RT] = 90 N30 <Datenhaltungsframe>[0,Z,RT] = 90	0	90	90
N10 <Datenhaltungsframe>[0,X,RT] = 90 N20 <Datenhaltungsframe>[0,Y,RT] = 90 N30 <Datenhaltungsframe>[0,Z,RT] = 0	0	90	-90 ¹⁾
N10 <Datenhaltungsframe>[0,X,RT] = 90 N20 <Datenhaltungsframe>[0,Y,RT] = 90 N30 <Datenhaltungsframe>[0,Z,RT] = 90	0	90	0 ¹⁾
1) unterschiedliche Werte gegenüber dem Schreiben einzelner Drehkomponenten eines aktiven Frames			

- Schreiben einzelner Drehkomponenten eines **aktiven Frames**
Eine eventuell notwendige **Umrechnung** erfolgt **sofort beim Schreiben der jeweiligen Drehkomponente**.
Die gespeicherten Ausgangswerte des aktiven Frames seien: x = 0, y = 0, z = 0.

Programmiert	Werte beim Zurücklesen		
	x, RT	y, RT	z, RT
N10 <aktiver Frame>[0,X,RT] = 0 N20 <aktiver Frame>[0,Y,RT] = 90 N30 <aktiver Frame>[0,Z,RT] = 90	0	90	90
N10 <aktiver Frame>[0,X,RT] = 90 N20 <aktiver Frame>[0,Y,RT] = 90 N30 <aktiver Frame>[0,Z,RT] = 0	0	90	0 ¹⁾

Programmiert	Werte beim Zurücklesen		
	x, RT	y, RT	z, RT
N10 <aktiver Frame>[0,X,RT] = 90	0	90	90 ¹⁾
N20 <aktiver Frame>[0,Y,RT] = 90			
N30 <aktiver Frame>[0,Z,RT] = 90			
1) unterschiedliche Werte gegenüber dem Schreiben des Gesamtframes bzw. dem Schreiben einzelner Drehkomponenten eines Datenhaltungsframes			

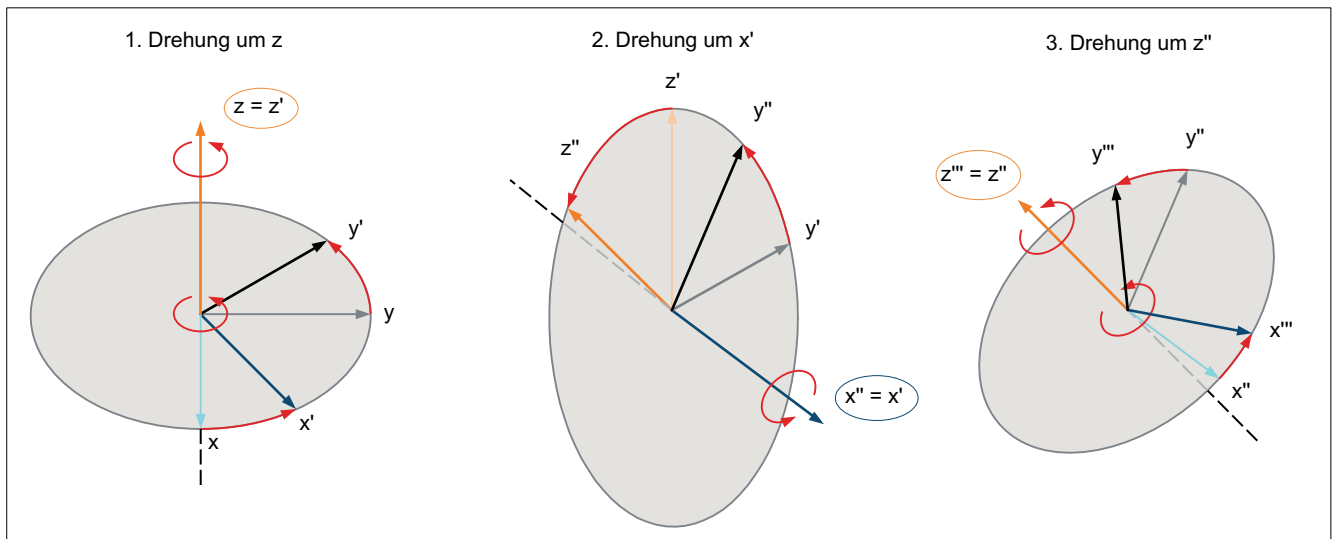
3.5.2.5 Drehung mit Euler-Winkeln: ZX'Z''-Konvention

Mit Euler-Winkeln erfolgen die Drehungen in der Reihenfolge z, x', z''.

Hinweis

Verwendungsempfehlung

Aus historischen Gründen ist die Möglichkeit der Verwendung von Euler-Winkeln in zx'z''-Konvention vorhanden. Es wird aber dringend empfohlen ausschließlich Euler-Winkel in zy'x''-Konvention (RPY-Winkel) (siehe Kapitel "Drehung mit Euler-Winkeln: ZY'X''-Konvention (RPY-Winkel) (Seite 312)") zu verwenden.



Zuordnung Drehachse zu Geometrieachse

Drehachse	Geometrieachse im Kanal
z	3. Geometrieachse
x'	1. Geometrieachse
z''	3. Geometrieachse

Wertebereich

Angaben von Euler-Winkeln können nur innerhalb folgender Wertebereiche eindeutig rückgerechnet werden:

0	<=	x	<	180
-180	<=	y	<=	180
-180	<=	z	<=	180

Angaben außerhalb der angegebenen Wertebereiche werden modulo der Bereichsgrenzen gerechnet.

Hinweis

Es wird empfohlen, beim Schreiben der Drehkomponenten des Frames die angegebenen Wertebereiche einzuhalten, um beim Zurücklesen der Drehkomponenten wieder die gleichen Werte zu erhalten.

3.5.2.6 Drehung in beliebiger Ebene

CRPL - Constant Rotation Plane

Mit der vordefinierten Funktion "**Constant Rotation Plane**" kann für einen Frame eine Drehung in einer beliebigen Ebene (G17, G18, G19), ohne Angabe des Namens einer Geometrieachse, programmiert werden. Somit können auch Drehungen in der dritten Ebene programmiert werden, wenn aufgrund der spezifischen Maschinenkonstellation nur zwei Geometrieachsen im Kanal vorhanden sind.

Syntax

CRPL (<Drehachse>, <Drehwinkel>)

Bedeutung

CRPL: Drehung in beliebiger Ebene

<Drehachse>: Achse um die gedreht wird

Typ: INT

Wert	Bedeutung
0	Drehung in der aktiven Ebene
1	Drehung um Z
2	Drehung um Y
3	Drehung um X

<Drehwinkel>: Winkel in Grad um den gedreht wird

Typ: REAL

Es wird dringend empfohlen die angegebenen Winkelbereiche einzuhalten. Werden die Grenzen nicht eingehalten, ist keine eindeutige Rückrechnung möglich. Winkelangaben außerhalb der Grenzen werden nicht abgelehnt.

RPY-Winkel: X -180 <= <Drehwinkel> <= 180

ZX'Z"-Konvention:	Y	-90 <= <Drehwinkel> <= 90
	Z	-180 <= <Drehwinkel> <= 180
	X	-180 <= <Drehwinkel> <= 180
	Y	0 <= <Drehwinkel> <= 180
	Z	-180 <= <Drehwinkel> <= 180

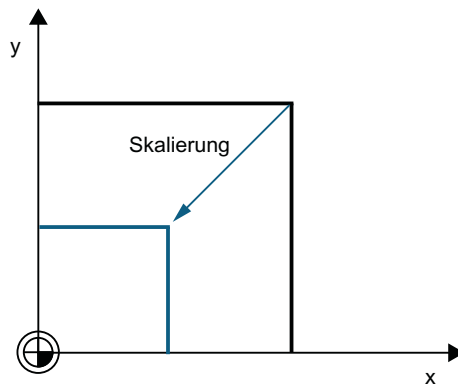
Verkettung mit Frames

CRPL () kann mit Frames und Frame-Funktionen wie CTRANS (), CROT (), CMIRROR (), CSCALE (), CFINE () etc. verkettet werden.

Beispiele:

```
$P_PFRAME = $P_PFRAME : CRPL(0,30.0)
$P_PFRAME = CTRANS(X,10) : CRPL(1,30.0)
$P_PFRAME = CROT(X,10) : CRPL(2,30.0)
$P_PFRAME = CRPL(3,30.0) : CMIRROR(Y)
```

3.5.2.7 Skalierung

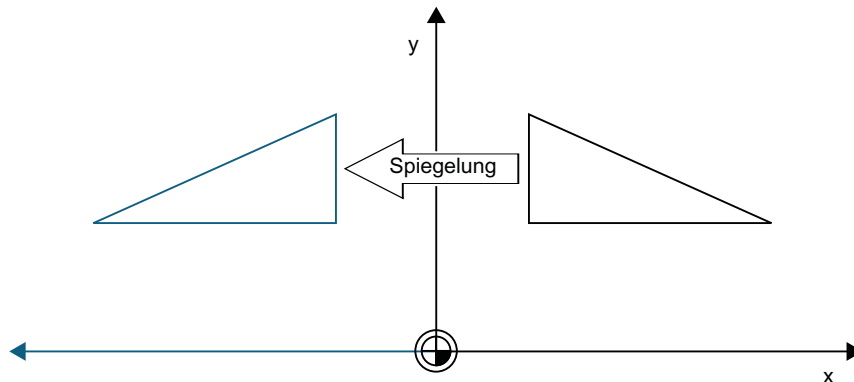


Programmierung

Die Programmierung einer Skalierung erfolgt über folgende Programmbefehle:

```
$P_UIFR[1] = CSCALE(x,1,y,1)
SCALE x = 1y = 1
$P_UIFR[1,x,sc] = 1
```

3.5.2.8 Spiegelung



Programmierung

Die Programmierung einer Spiegelung erfolgt über folgende Programmbefehle:

```
$P_UIFR[1] = CMIRROR(x,1,y,1)
MIRROR x = 1y = 1
$P_UIFR[1,x,mi] = 1
```

3.5.2.9 Verkettungsoperator

Framekomponenten oder gesamte Frames lassen sich über den Verkettungsoperator (:) zu einem Gesamtframe zusammenfassen.

3.5.2.10 Programmierbare Achsname

In den Frame-Befehlen können Geo-, Kanal- und Maschinen-Achsname verwendet werden. Die programmierte Achse muss bei den kanalspezifischen Frames im Kanal bekannt sein.

SPI

Bei der Programmierung von Frame-Anweisungen kann an Stelle eines Achsnamens auch die Achsfunktion `SPI(<spindelnummer>)` verwendet werden.

`SPI(<spindelnummer>)` stellt dabei den Bezug der Spindel zu der Kanalachse.
→ siehe MD35000 `$MA_SPIND_ASSIGN_TO_MACHAX[]` (Zuordnung Spindel zur Maschinenachse)

Folgende Frame-Anweisungen können mit `SPI(spino)` programmiert werden:

```
CTRANS ( )
CFINE ( )
CMIRROR ( )
CSCALE ( )
```


Eine Spindel kann immer nur einer Rundachse zugewiesen werden. Deshalb kann die Funktion `CROT (. .)` nicht mit `SPI ()` programmiert werden, da für `CROT ()` nur Geometrie-Achsen erlaubt sind.

Bei der Rückübersetzung von Frames wird immer der Kanalachsname bzw. der Maschinenachsname der zur Spindel gehörenden Achse ausgegeben, auch wenn im Teileprogramm Achsname mit `SPI (. .)` programmiert worden sind.

Ist die Spindel z. B. der Kanalachse "A" zugeordnet, dann wird aus der Programmierung:

```
N10 $P_UIFR[1] =
CTTRANS (SPI (1) , 33.33, X, 1) : CSCALE (SPI (1) , 33.33) : CMIRROR (SPI (1) )
```

bei der Rückübersetzung:

```
$P_UIFR[1]=CTTRANS (X, 1, A, 33.33) : CSCALE (A, 33.33) : CMIRROR (A)
```

Wird in einer Frame-Anweisung eine Spindel und die zugeordnete Achse programmiert, wird der Alarm 16420 "Achse % mehrfach programmiert" ausgegeben.

Beispiel:

```
$P_UIFR[1] = CTRANS (SPI (1) , 33.33, X, 1, A, 44)
```

(Die Spindel ist der Achse A zugeordnet.)

Programmierbeispiele

```
$P_PFRAME [SPI (1) , TR]=22.22
```

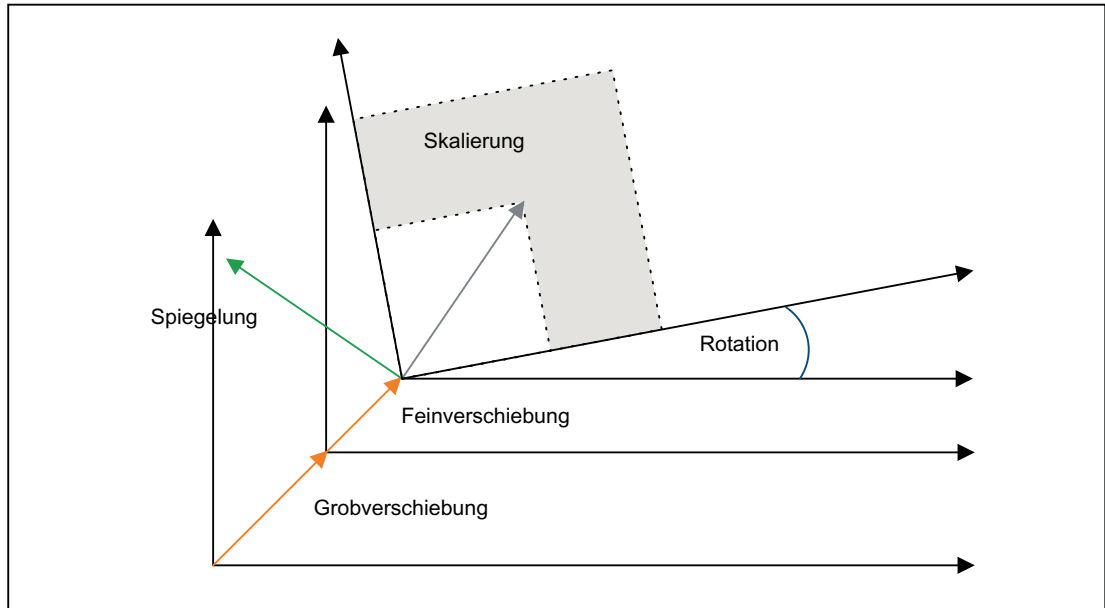
```
$P_PFRAME=CTTRANS (X, Achswert, Y, Achswert, SPI (1) , Achswert)
```

```
$P_PFRAME=CSCALE (X, Maßstab, Y, Maßstab, SPI (2) , Maßstab)
```

```
$P_PFRAME=CMIRROR (S1, Y, Z)
```

```
$P_UBFR=CTTRANS (A, 10) : CFINE (SPI (1) , 0.1)
```

3.5.2.11 Koordinatentransformation



Die Koordinatentransformation für Geometrieachsen ergibt sich anhand folgender Formeln:

WKS → BKS	$\vec{v} = R * \underline{S} * \underline{M} * \vec{v}' + \underline{t}$
BKS → WKS	$\vec{v}' = \text{inv}(\underline{M}) * \text{inv}(\underline{S}) * \text{inv}(\underline{R}) + (\underline{v} - \underline{t})$

V: Positionsvektor im BKS

V': Positionsvektor im WKS

3.5.3 Datenhaltungs-Frames und aktive Frames

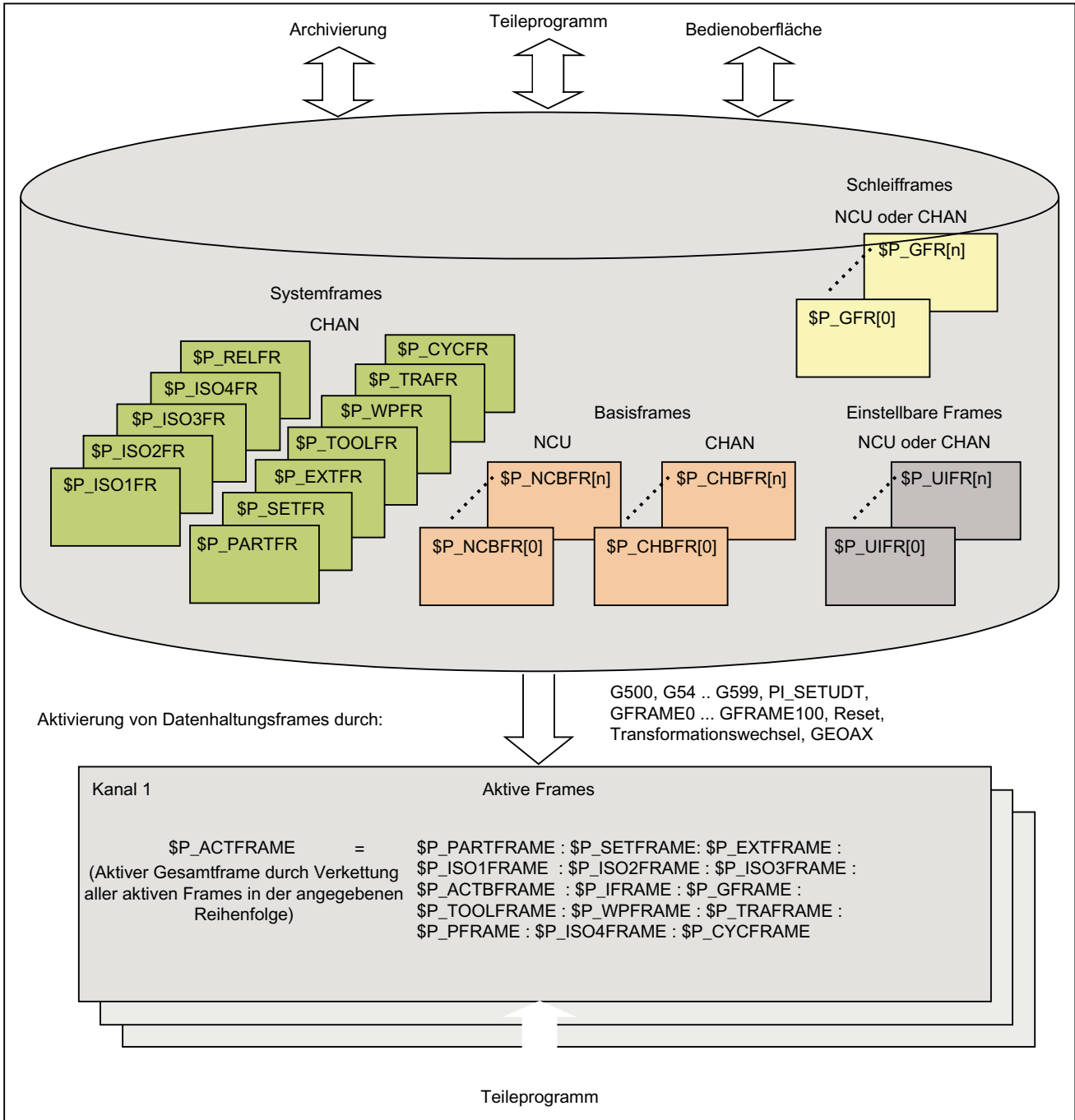
3.5.3.1 Übersicht

Frame-Typen

Es gibt folgende Frames-Typen:

- Systemframes ($\$P_PARTFR$, ... siehe Bild)
- Basisframes ($\$P_NCBFR[<n>]$, $\$P_CHBFR[<n>]$)
- Schleiframes ($\$P_GFR[<n>]$)
- Einstellbare Frames ($\$P_UIFR[<n>]$)
- Programmierbarer Frame ($\$P_PFRAME$)

Bei allen Frame-Typen, außer dem Programmierbaren Frame, existiert neben dem im Kanal aktiven Frame ein oder mehrere Frames in der Datenhaltung (Datenhaltungsframes). Beim Programmierbaren Frame existiert nur der im Kanal aktive Frame.



Schreiben von Frames

Aus dem Teileprogramm heraus können Datenhaltungsframes und aktive Frames geschrieben werden. Über die Bedienoberfläche können nur Datenhaltungsframes geschrieben werden.

Archivieren von Frames

Es können nur Datenhaltungsframes archiviert werden.

3.5.3.2 Aktivierung von Datenhaltungsframes

Datenhaltungsframes werden zu aktiven Frames durch folgende Aktionen:

- G-Gruppe "Einstellbare Frames": G54 ... G57, G500, G505 ... G599
- G-Gruppe "Schleifframes": GFRAME0 ... GFRAME100
- RESET und MD20110 \$MC_RESET_MODE_MASK, Bit14 == 1 (Die aktuelle Einstellung des Basisframes bleibt erhalten)
- Transformationswechsel
- Änderung der Geometrieachsordnung GEOAX
- Von HMI aus mit PI-Dienst "_N_SETUDT"

Aktivierung von HMI aus

Die Aktivierung eines Datenhaltungsframes von HMI aus mit dem PI-Dienst "_N_SETUDT" wirkt im Kanal erst nach einem Fortsetzstart für das aktuelle NC-Programm.

Die Aktivierung wirkt im Reset-Zustand sofort, wenn folgendes Maschinendatum gesetzt ist:
MD9440 \$MM_ACTIVATE_SEL_USER_DATA (Aktive Verschiebung sofort wirksam setzen)

Aktivierung von Systemframes

Die Aktivierung von Systemframes erfolgt durch:

- Programmierung der entsprechenden Systemfunktion im Teileprogramm
- Bedienhandlung an SINUMERIK Operate

Hinweis

Modifizieren von Systemframes der Datenhaltung

Systemframes der Datenhaltung können zwar prinzipiell durch den Zyklen-Programmierer modifiziert und über eine G500, G54...G599-Anweisung aktiviert werden. Diese Möglichkeit sollte aber nur unter Vorbehalt verwendet werden.

Aktivierung von Datenhaltungsframes

Das Verhalten bei Aktivierung von Datenhaltungsframes wird über folgendes Maschinendatum eingestellt:

MD24050 \$MC_FRAME_SAA_MODE (Speichern und aktivieren von Datenhaltungsframes)

Bit	Wert	Bedeutung
0	0	@@@
	1	<ul style="list-style-type: none"> Datenhaltungsframes werden nur durch Programmierung der Systemvariablen \$P_CHBFRMASK, \$P_NCBFRMASK und \$P_CHSFRMASK aktiv. G500...G599 aktiviert nur das entsprechende Einstellbare Frame.
1	0	Datenhaltungsframes werden durch Funktionen wie TOROT, PAROT, externe Nullpunktverschiebung, Transformationen, implizit beschrieben.
	1	Datenhaltungsframes werden durch Funktionen wie TOROT, PAROT, externe Nullpunktverschiebung, Transformationen nicht implizit beschrieben.

Aktivierung von Systemframes über Systemvariable \$P_CHSFRMASK

Die Systemframes der Datenhaltung können über die Systemvariable \$P_CHSFRMASK aktiviert werden. Der Wert der Variablen wird bitcodiert angegeben, entsprechend dem Maschinendatum:

MD28082 \$MC_MM_SYSTEM_FRAME_MASK (Systemframes der Datenhaltung)

Durch Setzen eines Bits der Systemvariable \$P_CHSFRMASK auf den Wert 1, wird der entsprechende Systemframe der Datenhaltung im Kanal aktiv. Bei einem Wert 0 bleibt der im Kanal aktuell aktive Systemframe weiterhin aktiv.

Aktivierung von Systemframes nach RESET

Nach RESET werden im Kanal die Systemframes aktiviert, deren Bits im folgenden Maschinendatum gesetzt sind:

MD24006 \$MC_CHSFRAME_RESET_MASK (Aktive Systemframes nach Reset)

Aktivierung der Systemframes für TCARR, PAROT und TOROT, TOFRAME

Die Systemframes für TCARR, PAROT und TOROT, TOFRAME werden entsprechend der Einstellung im folgenden Maschinendatum aktiviert:

MD20150 \$MC_GCODE_RESET_VALUES (Löschstellung der G-Gruppen)

Beim Umschalten von Geometrieachsen durch Transformations-An/Abwahl oder Befehl GEOAX, wird das aktuelle Gesamtframe \$P_ACTFRAME entweder gelöscht oder anhand der neuen Geometrieachs-Konstellation neu berechnet und aktiviert. Die Systemframes werden, ebenso wie alle anderen Frames, bezüglich der Geometrieachsen neu aufbereitet.

3.5.3.3 NCU-lobale und kanalspezifische Frames

- Einstellbare Frames und Schleifframes können nur alternativ NCU-global **oder** kanalspezifisch projiziert werden.
- Basisframes können NCU-global **und** kanalspezifisch projiziert werden.
- Ein NCU-globaler Frame wirkt in allen Kanäle einer NCU gleich.

- Alle Kanäle einer NCU können NCU-globale Frames gleichberechtigt lesen und schreiben.
- Da die Zuordnung von Maschinenachsen zu Kanalachsen und speziell zu Geometrieachsen in allen Kanälen unterschiedlich sein kann, gibt es demzufolge keine eindeutigen kanalübergreifenden geometrischen Zusammenhang zwischen den Kanalachsen. Bei NCU-globalen Frames sind daher nur Verschiebung, Skalierung und Spiegelung möglich. Drehungen sind nicht möglich.

Hinweis

Programmkoordinierung

Die Koordination von kanalspezifischen Zugriffen auf NCU-globale Frames steht allein in der Verantwortung des Anwenders. Es wird dazu die Verwendung der Befehle zur Programmkoordinierung empfohlen.

Weitere Informationen

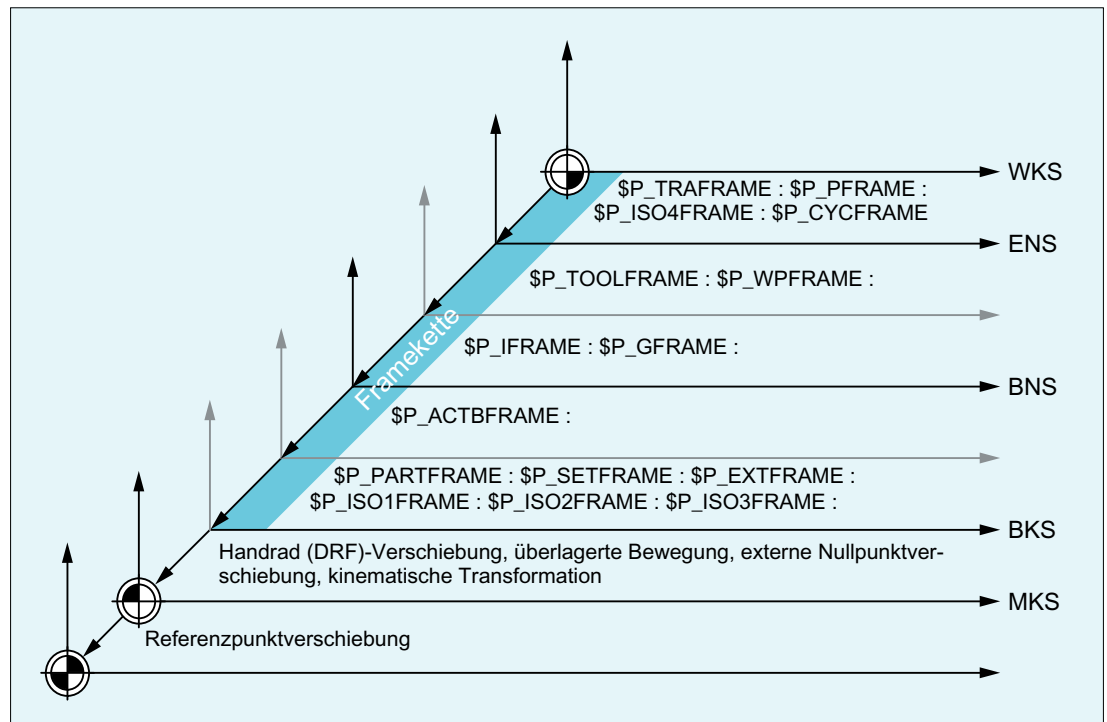
Programmierhandbuch Arbeitsvorbereitung; Flexible NC-Programmierung > Programmkoordinierung (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

3.5.4 Framekette und Koordinatensysteme

3.5.4.1 Übersicht

Im folgenden Bild ist die Framekette für das aktuelle Gesamtframe abgebildet. Die Framekette liegt zwischen dem Basiskoordinatensystem (BKS) und dem Werkstückkoordinatensystem (WKS).

Das Einstellbare Nullpunkt-System (ENS) entspricht dem durch das Programmierbare Frame transformierte WKS. Das Basis-Nullpunkt-System (BNS) beinhaltet noch das aktuelle Einstellbare Frame. Das Systemframe für die externe Nullpunktverschiebung ist nur vorhanden, wenn es projiziert wurde. Sonst wird die externe Nullpunktverschiebung als überlagerte Bewegung der Achse heraus gefahren.



- WKS: **W**erkstück **K**oordinaten **S**ystem
 ENS: **E**instellbares **N**ullpunkt **S**ystem
 BNS: **B**asis **N**ullpunkt **S**ystem
 BKS: **B**asis **K**oordinaten **S**ystem
 MKS: **M**aschinen **K**oordinaten **S**ystem

Gesamtframe

Das aktuelle Gesamtframe $\$P_ACTFRAME$ ergibt sich aus der Verkettung aller aktiven Frames der Framekette:

$$\begin{aligned} \$P_ACTFRAME = & \quad \$P_PARTFRAME : \$P_SETFRAME : \$P_EXTFRAME : \\ & \quad \$P_ISO1FRAME : \$P_ISO2FRAME : \$P_ISO3FRAME : \\ & \quad \$P_ACTBFRAME : \$P_IFRAME : \$P_GFRAME : \\ & \quad \$P_TOOLFRAME : \$P_WPFRAME : \$P_TRAFRAME : \\ & \quad \$P_PFRAME : \$P_ISO4FRAME : \$P_CYCFRAME \end{aligned}$$

3.5.4.2 Relative Koordinatensysteme

Relative Koordinatensysteme dienen nur zur Anzeige von aktuellen Achs-Sollpositionen, die relativ zu einem vorgegebenen Referenzpunkt im aktiven angezeigten Koordinatensystem liegen. Bezüglich den relativen Koordinatensystemen kann nicht programmiert werden. Es können nur Achspositionen in diesen Systemen über Systemvariablen gelesen werden.

Die neuen Anzeige-Koordinatensysteme liegen relativ zum WKS- und ENS-Koordinatensystem und ergeben sich durch Transformation der WKS- oder ENS-Achspositionen mit dem aktiven Systemframe $\$P_RELF$. Die relativen

3.5 Frames

Koordinatensysteme können nicht nur linear verschoben sein, sondern auch noch gedreht, gespiegelt, gestaucht bzw. gedehnt werden.

Die Positionsanzeige für Achssollwerte erfolgt im WKS oder im ENS. Die Projektierung erfolgt über HMI-Maschinendaten. Es ist immer nur ein Anzeige-Koordinatensystem im Kanal aktiv. Deshalb wird auch nur ein relatives Frame zur Verfügung gestellt, welches die beiden relativen Koordinatensysteme im gleichen Verhältnis erzeugt. Das HMI zeigt die relativen Koordinaten entsprechend der Projektierung an.

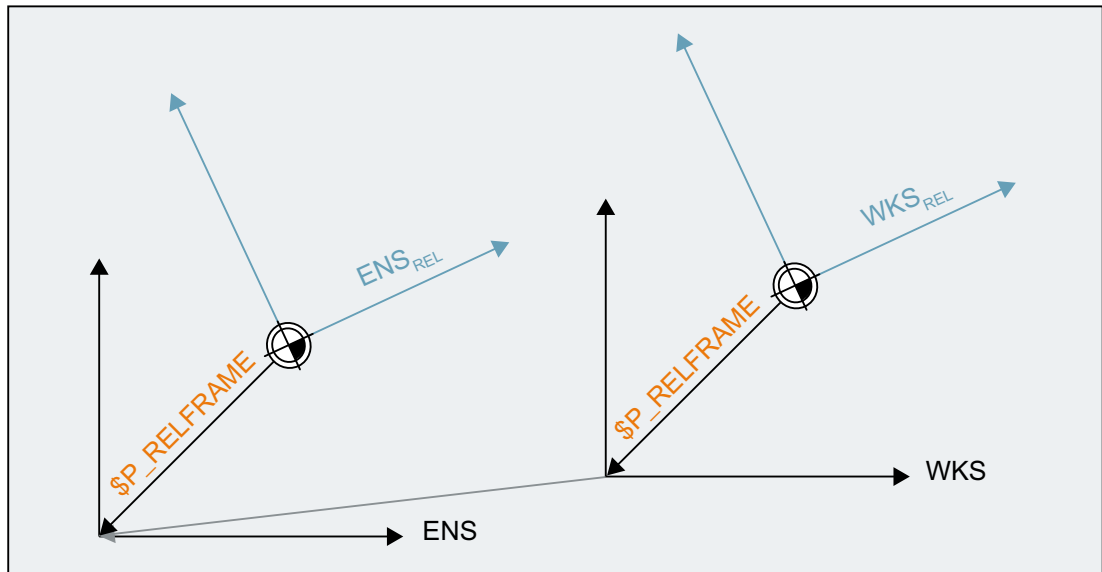


Bild 3-22 Relative Koordinatensysteme

Die Funktion "Relative Koordinatensysteme" wird mit MD51036 \$MNS_ENABLE_COORDINATE_REL=1 aktiviert.

Das Datenhaltungsframe \$P_RELFR kann im Teileprogramm und über BTSS beschrieben werden. Es können alle Framekomponenten verändert werden.

Das aktive Systemframe \$P_RELFRAME kann im Teileprogramm beschrieben und über BTSS gelesen werden.

Die Projektierung des Systemframes \$P_RELFR erfolgt über folgende Maschinendaten:

Maschinendatum	Bit	Bedeutung
MD28082 \$MC_MM_SYSTEM_FRAME_MASK	11	Anlegen von \$P_RELFR; damit werden relative Koordinatensysteme existent.
MD28083 \$MC_MM_SYSTEM_DATAFRAME_MASK	11	Datenhaltungsframe \$P_RELFR
MD24006 \$MC_CHSFRAME_RESET_MASK.	11	\$P_RELFR wird bei Reset aktiv
MD24007 \$MC_CHSFRAME_RESET_CLEAR_MASK	11	\$P_RELFR wird bei Reset gelöscht
MD24008 \$MC_CHSFRAME_POWERON_MASK	11	\$P_RELFR wird bei PowerOn gelöscht

Die Achspositionen im relativen Koordinatensystem WKS_{Rel} können über die Variable \$AA_PCS_REL[ax] gelesen werden. Die Variable ist im Teileprogramm, BTSS und über Synchronaktionen lesbar.

Die Achspositionen im relativen Koordinatensystem ENS_{Rel} können über die Variable \$AA_ACS_REL[ax] gelesen werden. Die Variable ist im Teileprogramm, BTSS und über Synchronaktionen lesbar.

Das Setzen eines relativen Bezugspunktes über die Bedienoberfläche erfolgt über die allgemeine Kommandoschnittstelle für die Werkstück- und Werkzeugvermessung. Das Systemframe $\$P_RELFR$ für relative Koordinatensysteme wird folgendermaßen berechnet und aktiviert:

- $\$AC_MEAS_TYPE = 14$
- PI-Dienste $_N_SETUDT(6, 7)$

Ein Beispiel zum Setzen von relativen Achspositionen findet sich in:

Weitere Informationen

Funktionshandbuch Technologien; Messen (M5)

3.5.4.3 Wählbares ENS

Innerhalb eines Zyklus erfolgt die Bearbeitung in einem Zyklus-spezifischen Werkstückkoordinatensystem (WKS). Das Zyklus-spezifische WKS entsteht dabei aus dem ENS transformiert durch für den Zyklus programmierten Frames Programmierbaren Frame $\$P_PFRAME$ und/oder Zyklen-Frame $\$P_CYCFRAME$.

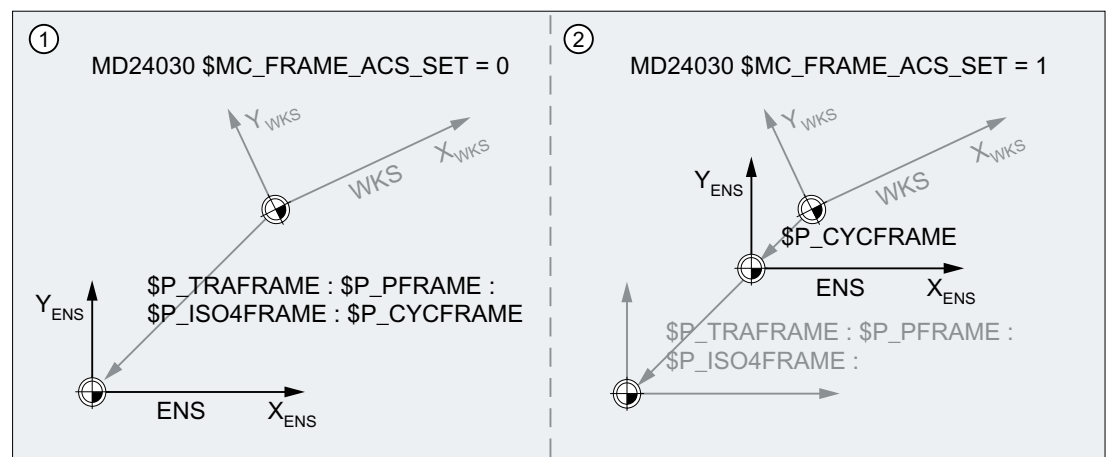
Wird ein Zyklus von einem Maschinenbediener, z.B. durch NC-Stop unterbrochen, soll anschließend im ursprünglichen, vor Aktivierung des Zyklus gültigen Koordinatensystem (ENS), verfahren werden.

Maschinendaten

Die Vorgabe, wie das ENS aus dem Zyklus-spezifischen WKS zu berechnen ist, kann über folgendes Maschinendatum eingestellt werden.

MD24030 $\$MC_FRAME_ACS_SET = \langle \text{Wert} \rangle$

$\langle \text{Wert} \rangle$	Bedeutung
0	ENS = WKS transformiert mit $\$P_TRAFFRAME$, $\$P_PFRAME$, $\$P_ISO4FRAME$ und $\$P_CYCFRAME$
1	ENS = WKS transformiert nur mit $\$P_CYCFRAME$



- ① ENS = WKS transformiert mit $\$P_TRAFFRAME$, $\$P_PFRAME$, $\$P_ISO4FRAME$ und $\$P_CYCFRAME$
- ② ENS = WKS transformiert nur mit $\$P_CYCFRAME$

Auswirkungen

Die Umprojektierung des ENS hat Rückwirkung auf:

- ENS-bezogenen Istwerte: Istwertanzeigen, Systemvariablen z.B. \$AA_IEN etc.
- Manuelles Verfahren (JOG) von Geometrieachsen im ENS

3.5.4.4 Manuelle Verfahren von Geometrieachsen wahlweise im WKS oder ENS (\$AC_JOG_COORD)

Die Geometrieachsen werden bisher beim manuellen Verfahren in der Betriebsart JOG im WKS verfahren. Zusätzlich dazu gibt es die Möglichkeit, das manuelle Verfahren im ENS-Koordinatensystem durchzuführen. Dazu gibt es die Variable \$AC_JOG_COORD, mit der zwischen manuellem Verfahren im WKS und ENS umgeschaltet werden kann. Der Anwender hat jetzt die Möglichkeit auszuwählen, ob er im ENS oder im WKS verfahren möchte.

Die Geometrieachsen können beim manuell Verfahren in der Betriebsart JOG wahlweise im Werkstückkoordinatensystem (WKS) oder im Einstellbaren Nullpunktsystem (ENS) verfahren werden.

Systemvariablen

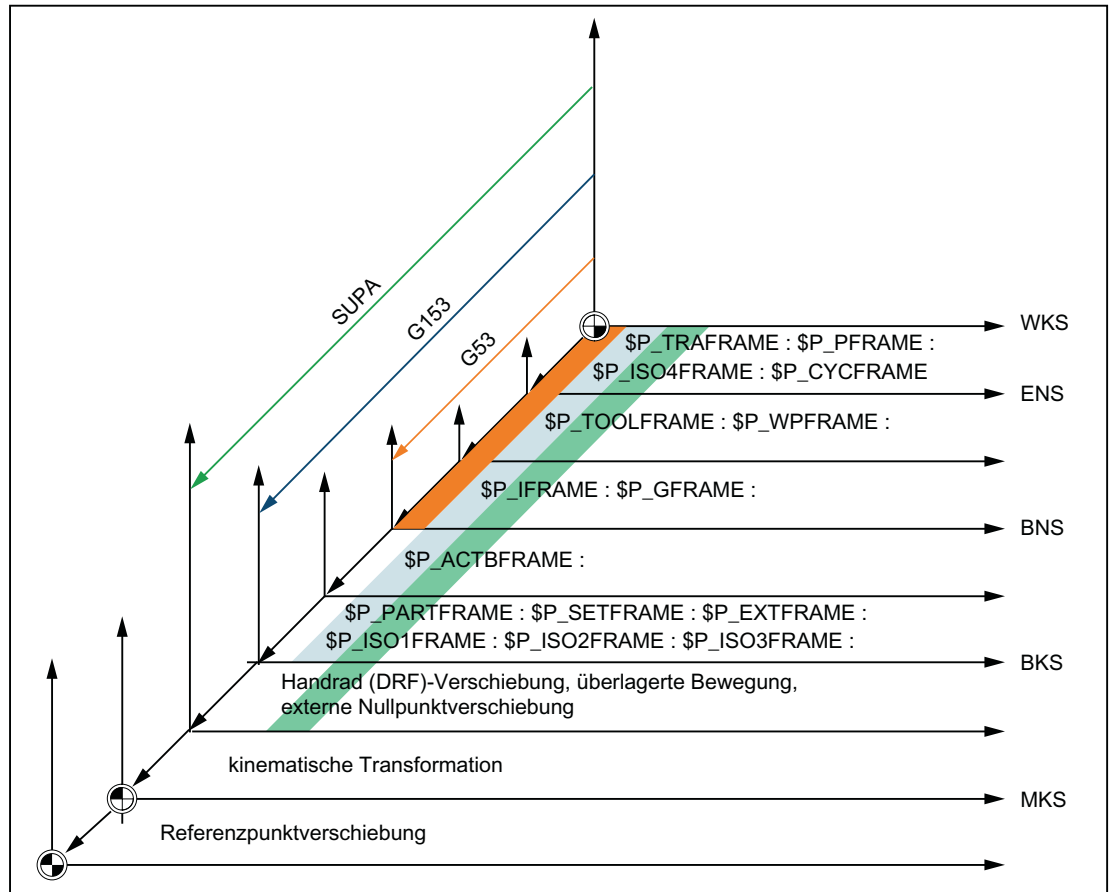
Die Geometrieachsen können beim manuellen Verfahren in der Betriebsart JOG wahlweise im Werkstückkoordinatensystem (WKS) oder in Einstellbaren Nullpunktsystem (ENS) verfahren werden. Die Auswahl erfolgt über die Systemvariable \$AC_JOG_COORD:

\$AC_JOG_COORD = <Wert>

<Wert>	Bedeutung
0	Werkstückkoordinatensystem (WKS)
1	Einstellbarens Nullpunktsystem (ENS)

3.5.4.5 Unterdrückung von Frames

Die Unterdrückung von Frames erfolgt kanalspezifisch über die nachfolgend beschriebenen Befehle `G53`, `G135` und `SUPA`. Eine Aktivierung der Frame-Unterdrückungen führt dazu, dass Positionsanzeigen (HMI) sowie Positionsangaben in Systemvariablen, die sich auf das WKS, ENS oder BNS beziehen, springen. Das Verhalten kann über Maschinendaten eingestellt werden.



Maschinendaten

Über das folgende Maschinendatum wird das Verhalten von Positionsanzeigen (HMI) und Positionsangaben in Systemvariablen festgelegt:

MD24020 `$MC_FRAME_SUPPRESS_MODE, Bit<n> = <Wert>` (Positionen bei Frameunterdrückung)

Bit	Wert	Bedeutung
0	0	Positionsangaben (BTSS) mit Frame-Unterdrückung ¹⁾
	1	Positionsangaben (BTSS) ohne Frame-Unterdrückung ²⁾

3.5 Frames

Bit	Wert	Bedeutung
1	0	Positionsangaben in Systemvariablen mit Frame-Unterdrückung ¹⁾
	1	Positionsangaben in Systemvariablen ohne Frame-Unterdrückung ²⁾
1) Springen des Positionswertes		
2) kein Springen des Positionswertes		

Programmierung

Befehl	Bedeutung
G53:	Satzweises Unterdrücken folgender Frames: <code>\$P_TRAFRAME : \$P_PFRAME : \$P_ISO4FRAME : \$P_CYCFRAME</code> <code>\$P_IFRAME : \$P_GFRAME : \$P_TOOLFRAME : \$P_WPFRAME :</code>
G153:	Satzweises Unterdrücken der Frames wie bei G53 plus folgender Frames: <code>\$P_PARTFRAME : \$P_SETFRAME : \$P_EXTFRAME : \$P_ACTBFRAME</code> <code>\$P_ISO1FRAME : \$P_ISO2FRAME : \$P_ISO3FRAME :</code>
SUPA:	Impliziter Vorlaufstopp und satzweises Unterdrücken der Frames wie bei G53 und G135 plus folgender Frames: <ul style="list-style-type: none"> • Handradverschiebungen (DRF) • Überlagerte Bewegung • Externe Nullpunktverschiebung
G500:	Modales Aktivieren des G500-Frames. Der G500-Frame sollte normalerweise ein Nullframe sein.
DRFOF:	Ausschalten (Löschen) der Handradverschiebungen (DRF)

3.5.5 Frames der Framekette

3.5.5.1 Übersicht

Es stehen folgende Frames zur Verfügung:

- Einstellbare Frames (G500, G54 ... G57, G505 ... G599)
- Schleifframes (GFRAME0 ... GFRAME100)
- Basisframes
- Programmierbares Frame
- Systemframes

3.5.5.2 Einstellbare Frames \$P_UIFR[<n>]

Maschinendaten

Kanalspezifische Einstellbare Frames

Die Anzahl der kanalspezifischen Einstellbaren Frames wird mit folgendem Maschinendatum eingestellt:

MD28080 \$MC_MM_NUM_USER_FRAMES = <Anzahl>

Systemvariablen-Index n = 0, 1, 2, ... <Anzahl> - 1

NCU-globale Einstellbare Frames

Die Anzahl der NCU-globalen Einstellbare Frames wird mit folgendem Maschinendatum eingestellt:

MD18601 \$MN_MM_NUM_GLOBAL_USER_FRAMES = <Anzahl>

Systemvariablen-Index n = 0, 1, 2, ... <Anzahl> - 1

Hat das Maschinendatum einen Wert > 0, existieren **keine kanalspezifischen** Einstellbaren Frames. Das Maschinendatum zum Einstellen der kanalspezifischen Einstellbaren Frames wird dann nicht ausgewertet.

Löschstellung der G-Gruppe der Einstellbare Frames

Die Löschstellung bzw. welche der G-Befehle der Einstellbare Frames-spezifischen **8. G-Gruppe** nach Kanal-Reset bzw. Power On wirksam wird, wird eingestellt in:

MD20150 \$MC_GCODE_RESET_VALUES[7] = <Wert>

Wert	G-Befehl
1	G500
2	G54
3	G55
4	G56
5	G57
6	G505
...	...
100	G599

Resetverhalten der schleifenspezifischen G-Gruppe

Das Resetverhalten der Einstellbaren Frames-spezifischen **8. G-Gruppe** wird eingestellt in:

MD20152 \$MC_GCODE_RESET_MODE[7] = <Wert>

Wert	Bedeutung Nach Kanal-Reset oder Teileprogrammende:
0	wird der Einstellbare Frames-spezifische G-Befehl entsprechend MD20150 aktiv.
1	bleibt der aktuell aktive Einstellbare Frames-spezifische G-Befehl weiter aktiv.

Hinweis

MD20110 \$MC_RESET_MODE_MASK

Das Maschinendatum MD20152 \$MC_GCODE_RESET_MODE wird nur ausgewertet bei:
MD20110 \$MC_RESET_MODE_MASK, Bit0 == 1

Systemvariablen

\$P_UIFR[<n>] (Einstellbare Frames der Datenhaltung)

Über die Systemvariable \$P_UIFR[<n>] können die Einstellbaren Frame der Datenhaltung gelesen und geschrieben werden. Beim Schreiben eines Einstellbaren Frames der Datenhaltung werden die neuen Werte nicht sofort im Kanal aktiv. Die Aktivierung im Kanal erfolgt erst mit Programmierung einer Nullpunktverschiebung G500, G54...G599.

Bei NCU-globalen Frames wird das geänderte Einstellbaren Frame der Datenhaltung in jedem Kanal der NCU der einen G500, G54...G599-Befehl ausführt, aktiv.

Die Einstellbaren Frames in der Datenhaltung werden bei einer Datensicherung mit gesichert.

\$P_IFRAME (Aktives Einstellbares Frame)

Über die Systemvariable \$P_IFRAME kann der im Kanal **aktive** Einstellbare Frame gelesen und geschrieben werden. Beim Schreiben des Einstellbare Frame werden die neuen Werte sofort im Kanal wirksam.

Bei NCU-globalen Einstellbaren Frames wirkt der geänderte aktive Einstellbaren Frame nur in dem Kanal, in dem die neuen Werte programmiert wurde. Soll das geänderte NCU-globale Einstellbare Frame für alle Kanäle der NCU wirksam werden, muss das im Kanal aktive Einstellbare Frame **und** das entsprechende Einstellbare Frame der Datenhaltung gemeinsam geschrieben werden:

\$P_UIFR[<n>] = \$P_IFRAME = <neuer Wert>

- \$P_UIFR[<n>] (Einstellbares Frame in der Datenhaltung)
- \$P_IFRAME (im Kanal aktiver Einstellbarer Frame) .

Damit das geänderte Einstellbare Frame in einem anderen Kanal wirksam wird, muss es in diesem Kanal noch mit dem entsprechenden Befehl, z. B. G54, aktiviert werden.

\$P_UIFRNUM (Nummer des aktiven Einstellbaren Frames)

Über die Systemvariable \$P_UIFRNUM kann der Index <n> des im Kanal aktiven Einstellbaren Frames der Datenhaltung gelesen werden:

Im Kanal aktiver Einstellbarer Frame \$P_IFRAME == \$P_UIFR[\$P_UIFRNUM]

\$P_UIFRNUM	\$P_IFRAME == \$P_UIFR[<n>], mit n =
0	0
1	1
2	2
...	...
99	99

Programmierung

Befehle zur Aktivierung eines Einstellbarer Frames im Kanal

Durch Programmierung eines Befehles G500, G54...G599 wird der Einstellbarer Frame der Datenhaltung \$P_UIFR[<n>] im Kanal aktiv bzw. der aktive Einstellbarer Frame \$P_IFRAME wird gleich dem Einstellbaren Frame der Datenhaltung \$P_UIFR[<n>] gesetzt:

$G<x> \Rightarrow \$P_IFRAME = \$P_UIFR[<n>]$

Befehl	Aktiver Einstellbarer Frame \$P_IFRAME =
G500	\$P_UIFR[0]
G54	\$P_UIFR[1]
G55	\$P_UIFR[2]
G56	\$P_UIFR[3]
G57	\$P_UIFR[4]
G505	\$P_UIFR[5]
...	...
G599	\$P_UIFR[99]

Randbedingungen

Schreiben von Einstellbaren Frames durch HMI / PLC

Von HMI oder dem PLC-Anwenderprogramm können nur die Einstellbaren Frames der **Datenhaltung** geschrieben werden.

3.5.5.3 Schleiframes \$P_GFR[<n>]

Schleiframes sind speziell für die Technologie Schleifen zur Verfügung stehende zusätzliche Nullpunktverschiebungen oder auch „Sitzabhängige Korrekturen“. Sie wirken additiv zu den Nullpunktverschiebungen der Einstellbaren Frames (Seite 333).

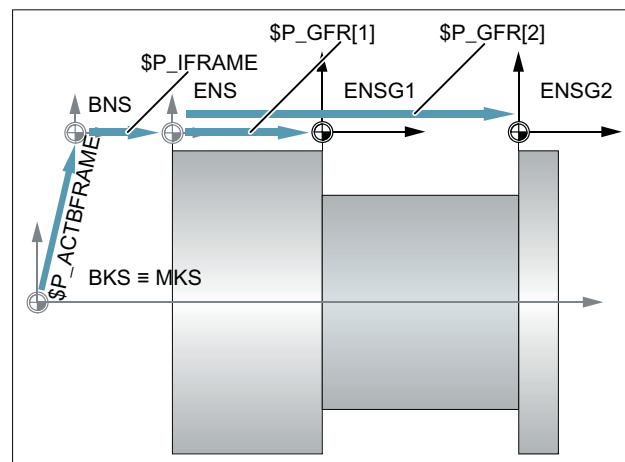


Bild 3-23 Schleiframes

Ausgehend vom Basis-Nullpunktsystem (BNS) ergibt sich somit das Einstellbare Nullpunktsystem (ENS) aus der Verkettung der im Kanal aktiven Frames:

\$P_IFRAME : \$P_GFRAME : \$P_TOOLFRAME : \$P_WPFRAME

Maschinendaten

Anzahl kanalspezifischer Schleiframes

Die Anzahl der kanalspezifischen Schleiframes wird eingestellt in:

MD28079 \$MN_MM_NUM_G_FRAMES = <Anzahl>

mit <Anzahl> = 0, 1, 2, ... maximale Anzahl

Anzahl NCU-globaler Schleiframes

Die Anzahl der NCU-globalen Schleiframes wird eingestellt in:

MD18603 \$MN_MM_NUM_GLOBAL_G_FRAMES = <Anzahl>

mit <Anzahl> = 0, 1, 2, ... maximale Anzahl

Hat das Maschinendatum einen Wert > 0, existieren **keine kanalspezifischen** Schleiframes. Das Maschinendatum zum Einstellen der kanalspezifischen Schleiframes wird dann nicht ausgewertet.

Löschstellung der schleifspezifischen G-Gruppe (64)

Die Löschstellung bzw. welche der G-Befehle der schleifspezifischen **64. G-Gruppe** nach Kanal-Reset bzw. Power On wirksam wird, wird eingestellt in:

MD20150 \$MC_GCODE_RESET_VALUES[63] = <Wert>

Wert	G-Befehl
1	GFRAME0 (Nullframe)
2	GFRAME1
3	GFRAME2
...	...
101	GFRAME100

Resetverhalten der schleifspezifischen G-Gruppe

Das Resetverhalten der schleifspezifischen **64. G-Gruppe** wird eingestellt in:

MD20152 \$MC_GCODE_RESET_MODE[63] = <Wert>

Wert	Bedeutung Nach Kanal-Reset oder Teileprogrammende:
0	wird der schleifspezifische G-Befehl entsprechend MD20150 aktiv.
1	bleibt die aktuell aktive schleifspezifische G-Befehl weiter aktiv.

Hinweis**MD20110 \$MC_RESET_MODE_MASK**

Das Maschinendatum MD20152 \$MC_GCODE_RESET_MODE wird nur ausgewertet bei:

MD20110 \$MC_RESET_MODE_MASK, Bit0 == 1

Systemvariablen**\$P_GFR[<n>] (Schleiframes der Datenhaltung)**

Über die Systemvariable \$P_GFR[<n>] können die Schleiframes der Datenhaltung gelesen und geschrieben werden. Beim Schreiben eines Schleiframes werden die neuen Werte nicht sofort im Kanal aktiv. Die Aktivierung im Kanal erfolgt erst mit Programmierung des entsprechenden Befehls GFRAME0 ... GFRAME100. Bei NCU-globalen Frames wird das geänderte Frame in jedem Kanal der NCU, der einen GFRAME0 ... GFRAME100-Befehl ausführt, aktiv.

Die Schleiframes der Datenhaltung werden bei einer Datensicherung mit gesichert.

Hinweis**Anzeige (SINUMERIK Operate)**

Die Schleiframes der Datenhaltung werden auf der Bedienoberfläche von SINUMERIK Operate in einem eigenen Fenster angezeigt.

- Weitere Informationen
Bedienhandbuch Schleifen; Kapitel "Maschine einrichten" > "Nullpunktverschiebungen" > "Sitzbezogene Feinverschiebung anzeigen und bearbeiten"

Löschen (SINUMERIK Operate)

Die Schleiframes der Datenhaltung können einzeln oder, z. B. nach einem Werkstückwechsel, alle gemeinsam über die Bedienoberfläche von SINUMERIK Operate gelöscht werden.

- Weitere Informationen
Bedienhandbuch Schleifen; Kapitel "Maschine einrichten" > "Nullpunktverschiebungen" > "Sitzbezogene Feinverschiebungen löschen"

\$P_GFRAME (Aktiver Schleiframe)

Über die Systemvariable \$P_GFRAME kann der im Kanal **aktive** Schleiframe gelesen und geschrieben werden. Beim Schreiben des aktiven Schleiframes werden die neuen Werte sofort im Kanal wirksam.

Bei NCU-globalen Schleiframes wirkt das geänderte Frame nur in dem Kanal, in dem die neuen Frame-Werte programmiert wurde.

Soll das geänderte NCU-globale Schleiframe von einem Kanal ausgehend für alle Kanäle der NCU wirksam werden, muss das im Kanal aktive Schleiframe \$P_GFRAME und das Schleiframe in der Datenhaltung \$P_GFR[<n>] gemeinsam geschrieben werden:

\$P_GFRAME = \$P_GFR[<n>] = <neuer Wert>

3.5 Frames

Damit das geänderte Schleifframe in der Datenhaltung \$P_GFR[<n>] in einem anderen Kanal wirksam wird, muss es in diesem Kanal noch mit dem entsprechenden Befehl GFRAME<n>, aktiviert werden.

\$P_GFRNUM (Nummer des aktiven Schleifframes)

Über die Systemvariable \$P_GFRNUM kann der Index <n> des im Kanal aktiven Schleifframes der Datenhaltung gelesen werden:

Im Kanal aktiver Schleifframe \$P_GFRAME == \$P_GFR[\$P_GFRNUM]

Durch G-Befehl GFRAME<n> aktivierter Schleifframe \$P_GFRAME:	\$P_GFRNUM
GFRAME0	0
GFRAME1	1
GFRAME2	2
...	...
GFRAME100	100

Programmierung

Befehl zur Aktivierung eines Schleifframes im Kanal

Durch Programmierung des Befehles GFRAME<n> wird der entsprechende Schleifframe der Datenhaltung \$P_GFR[<n>] im Kanal aktiv. Dazu wird der aktive Schleifframe \$P_GFRAME gleich dem Schleifframe der Datenhaltung \$P_GFR[<n>] gesetzt:

GFRAME<n> => \$P_GFRAME = \$P_GFR[<n>]

Befehl	Im Kanal aktivierter Schleifframe
GFRAME0	\$P_GFR[0] (Nullframe)
GFRAME1	\$P_GFR[1]
...	...
GFRAME100	\$P_GFR[100]

Syntax

GFRAME<n>

Bedeutung

\$P_GFRAME<n>:	Aktivierung des Schleifframes <n> der Datenhaltung	
	G-Gruppe:	64
	Grundstellung:	MD20150 \$MC_GCODE_RESET_VALUES[63]
	Wirksamkeit:	Modal
<n>:	Nummer des Schleifframes	
	Wertebereich:	0, 1, 2, ... 100

Randbedingungen

Schreiben von Schleifframes durch HMI / PLC

Von HMI oder dem PLC-Anwenderprogramm können nur die Schleifframes der **Datenhaltung** geschrieben werden.

3.5.5.4 Kanal-spezifische Basisframes[<n>]

Maschinendaten

Anzahl der kanalspezifischen Basisframes

Die Anzahl der kanalspezifischen Basisframes wird mit folgendem Maschinendatum eingestellt:

MD28081 \$MC_MM_NUM_BASE_FRAMES = <Anzahl>

Systemvariablen-Index n = 0, 1, 2, ... <Anzahl> - 1

Systemvariablen

\$P_CHBFR[<n>] (Kanalspezifische Basisframes der Datenhaltung)

Über die Systemvariablen \$P_CHBFR[<n>] können die kanalspezifischen Basisframes der Datenhaltung gelesen und geschrieben werden. Beim Schreiben eines kanalspezifischen Basisframes werden die neuen Werte nicht sofort im Kanal aktiv. Die Aktivierung im Kanal erfolgt erst mit Programmierung des entsprechenden Befehls G500, G54..G599.

Die kanalspezifischen Basisframes der Datenhaltung werden bei einer Datensicherung mit gesichert.

\$P_CHBFRAME[<n>] (Aktive kanalspezifische Basisframes)

Über die Systemvariablen \$P_CHBFRAME[<n>] können die im Kanal **aktiven** kanalspezifischen Basisframes gelesen und geschrieben werden. Beim Schreiben eines aktiven kanalspezifischen Basisframes werden die neuen Werte im Kanal sofort durch Neuberechnung des aktiven Gesamt-Basisframe \$P_ACTBFRAME wirksam.

Systemvariablen aus Kompatibilitätsgründen

\$P_UBFR (Erster kanalspezifischer Basisframe der Datenhaltung)

Die Systemvariable bleibt aus Kompatibilitätsgründen erhalten, obwohl sie redundant zu der Variablen \$P_CHBFR[0] ist.

Ein Schreiben auf die vordefinierte Variable \$P_UBFR aktiviert das Basisframe mit dem Feldindex 0 nicht gleichzeitig, sondern die Aktivierung erfolgt erst mit der Ausführung einer G500, G54, ..G599-Anweisung. Bei NCU-globalen Frames wird das geänderte Frame in jedem Kanal der NCU, der eine G500, G54..G599-Anweisung ausführt, aktiv. Die Variable dient vorwiegend als Speicher für Schreibvorgänge auf das Basisframe von HMI oder PLC. Die Variable kann auch im Programm geschrieben und gelesen werden.

\$P_UBFR ist identisch mit \$P_CHBFR[0]. Standardmäßig gibt es immer ein Basisframe im Kanal, so dass die Systemvariable kompatibel zu älteren Ständen ist. Gibt es kein kanalspezifisches Basisframe, so wird beim Schreiben oder Lesen der Alarm "Frame: Anweisung unzulässig" ausgegeben.

\$P_BFRAME (Erster aktiver kanalspezifischer Basisframe)

Die Systemvariable bleibt aus Kompatibilitätsgründen erhalten, obwohl sie redundant zu der Variablen \$P_CHBFRAME[0] ist.

Über die vordefinierte Framevariable \$P_BFRAME kann das aktuelle Basisframe mit dem Feldindex 0, welches im Kanal gültig ist, im Teileprogramm gelesen und geschrieben werden. Das geschriebene Basisframe wird sofort eingerechnet. Bei NCU-globalen einstellbaren Frames wirkt das geänderte Frame nur in dem Kanal, in dem das Frame programmiert wurde. Soll das Frame für alle Kanäle einer NCU geändert werden, so muss gleichzeitig \$P_UBFR und \$P_BFRAME beschrieben werden. Die anderen Kanäle müssen dann noch das entsprechende Frame mit z.B. G54 aktivieren.

\$P_BFRAME ist identisch mit \$P_CHBFRAME[0]. Die Systemvariable hat standardmäßig immer einen gültigen Wert. Gibt es kein kanalspezifisches Basisframe, so wird beim Schreiben oder Lesen der Alarm "Frame: Anweisung unzulässig" ausgegeben.

Randbedingungen

Schreiben von Basisframes durch HMI / PLC

Von HMI oder dem PLC-Anwenderprogramm können nur die Basisframes der **Datenhaltung** geschrieben werden.

3.5.5.5 NCU-globale Basisframes \$P_NCBFR[<n>]

Maschinendaten

Anzahl NCU-globale Basisframes

Die Anzahl der NCU-globalen Basisframes wird mit folgendem Maschinendatum eingestellt:

MD18602 \$MN_MM_NUM_GLOBAL_BASE_FRAMES = <Anzahl>

Systemvariablen-Index n = 0, 1, 2, ... <Anzahl> - 1

Systemvariablen

\$P_NCBFR[<n>] (NCU-globale Basisframes der Datenhaltung)

Über die Systemvariablen \$P_NCBFR[<n>] können die NCU-globalen Basisframes der Datenhaltung gelesen und geschrieben werden. Beim Schreiben eines NCU-globalen Basisframes werden die neuen Werte nicht sofort im Kanal aktiv. Die Aktivierung im Kanal erfolgt erst mit Programmierung des entsprechenden Befehls G500, G54..G599.

Die NCU-globalen Basisframes der Datenhaltung werden bei einer Datensicherung mit gesichert.

\$P_NCBFRAME[<n>] (Aktuelle NCU-globale Basisframes)

Über die Systemvariablen \$P_NCBFRAME[<n>] können die im Kanal **aktiven** NCU-globalen Basisframes gelesen und geschrieben werden. Beim Schreiben eines aktiven NCU-globalen Basisframes werden die neuen Werte im Kanal sofort durch Neuberechnung des aktiven Gesamt-Basisframe \$P_ACTBFRAME wirksam.

Soll das geänderte NCU-globale Basisframe von einem Kanal ausgehend für alle Kanäle der NCU wirksam werden, muss das im Kanal aktive NCU-globale Basisframe und das NCU-globale Basisframe der Datenhaltung gemeinsam geschrieben werden:

\$P_NCBFR[<n>] = \$P_NCBFRAME = <neuer Wert>

- \$P_NCBFR[<n>] (NCU-globaler Basisframe der Datenhaltung)
- \$P_NCBFRAME (im Kanal aktiver NCU-globaler Basisframe)

Damit das geänderte NCU-globale Basisframe in einem anderen Kanal wirksam wird, muss es in diesem Kanal noch mit dem entsprechenden Befehl G500, G54..G599, aktiviert werden.

Programmierung

Ein kanalspezifischer Einstellbarer Frame der Datenhaltung \$P_UIFR[<n>] wird durch den entsprechenden Befehl (G54 ... G57, G505 ... G599 und G500) zum im Kanal aktiven Einstellbarer Frame \$P_IFRAME.

Befehl	Aktivierung der NCU-globalen und kanalspezifischen Basisframes der Datenhaltung
G500	\$P_CHBFR[0] : \$P_NCBFR[0]
G54	\$P_CHBFR[1] : \$P_NCBFR[1]
G55	\$P_CHBFR[2] : \$P_NCBFR[2]
G56	\$P_CHBFR[3] : \$P_NCBFR[3]
G57	\$P_CHBFR[4] : \$P_NCBFR[4]
G505	\$P_CHBFR[5] : \$P_NCBFR[5]
...	...
G599	\$P_CHBFR[99] : \$P_NCBFR[99]

3.5.5.6 Aktiver Gesamt-Basisframe \$P_ACTBFRAME**Funktion**

Im Gesamt-Basisframe \$P_ACTBFRAME sind alle aktiven NCU-globalen und kanalspezifischen Basisframes zusammengefasst:

\$P_ACTBFRAME = \$P_NCBFRAME[0] : ... : \$P_NCBFRAME[<n>] :
\$P_CHBFRAME[0] : ... : \$P_CHBFRAME[<n>]

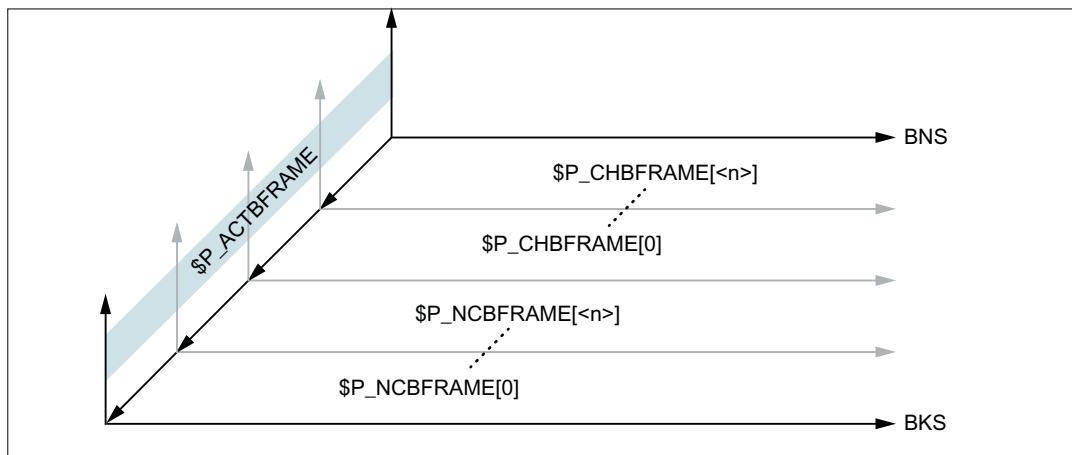


Bild 3-24 Gesamt-Basisframe

Maschinendaten

Reset-Verhalten

Welche Basisframes nach Reset (Kanal-Reset, Programmende-Reset bzw. Power On) aktiv sind, wird eingestellt über das Maschinendatum:

MD20110 \$MC_RESET_MODE_MASK, Bit0 = 1 und Bit14 = 1

- Bit0 = 1: Standardwert \Rightarrow Reset-Verhalten entsprechend der Einstellung der weiteren Bits
- Bit14 = 0: Mit Reset werden die Basisframes vollständig abgewählt.
- Bit14 = 1: Mit Reset werden die Maschinendateneinstellungen in die Systemvariablen übernommen und die darin angewählten Basisframes aktiv:
 - \$P_NCBFRMASK = MD10613 \$MN_NCBFRAME_RESET_MASK
 - \$P_CHBFRMASK = MD24002 \$MC_CHBFRAME_RESET_MASK

Beispiel

Maschinendateneinstellung	Aktive Basisframes
\$P_NCBFRMASK = MD10613 \$MN_NCBFRAME_RESET_MASK = 'H81'	\$P_NCBFRAME[0] : \$P_NCBFRAME[7]

Programmierung

Basisframemasken

Über die Basisframemasken \$P_NCBFRMASK und \$P_CHBFRMASK werden die Basisframes ausgewählt, die zum Gesamt-Basisframe verkettet werden.

Durch Setzen eines Bits in der Basisframemaske wird der entsprechende Basisframe ausgewählt:

- \$P_NCBFRMASK, Bit0, 1, 2, ... n \Rightarrow \$P_NCBFRAME[0, 1, 2, ... n]
- \$P_CHBFRMASK, Bit0, 1, 2, ... n \Rightarrow \$P_NCHFRAME[0, 1, 2, ... n]

Die Basisframemasken `$P_NCBFRMASK` und `$P_CHBFRMASK` können nur im NC-Programm gelesen / geschrieben werden. Über BTSS können die Basisframemasken gelesen werden.

Nach dem Schreiben einer Basisframemaske wird das aktive Gesamtbasisframe `$P_ACTBFRAME` und Gesamtframe `$P_ACTFRAME` neu berechnet.

Beispiel

Programmcode	Kommentar
<code>\$P_NCBFRMASK = 'H81'</code>	<code>; Aktive NCU-globale Basisframes: \$P_NCBFR-</code> <code>ME[0] : \$P_NCBFRAME[7]</code>

3.5.5.7 Programmierbarer Frame `$P_PFRAME`

Den programmierbaren Frame gibt es nur als aktiven Frame. Dieser Frame ist für den Programmierer reserviert.

Das programmierbare Frame bleibt bei RESET erhalten, wenn:

`MD24010 $MC_PFRAME_RESET_MODE` (Reset-Modus für programmierbaren Frame) = 1

Diese Funktionalität ist vor allem wichtig, wenn man nach RESET noch aus einer schrägen Bohrung herausfahren möchte.

MIRROR

Spiegelungen einer Geometrieachse wurden bisher (bis SW-P4) anhand des Maschinendatums:

`MD10610 $MN_MIRROR_REF_AX` (Bezugsachse für das Spiegeln) auf eine festgelegte Bezugsachse bezogen.

Diese Festlegung ist aus Anwendersicht schwer nachvollziehbar. Bei Spiegelung der z-Achse wurde in der Anzeige übermittelt, dass die x-Achse gespiegelt ist und die y-Achse um 180° gedreht wurde. Bei Spiegelungen von zwei Achsen wird das noch komplexer und es ist nicht mehr leicht nachvollziehbar, welche Achsen gespiegelt wurden und welche nicht.

Ab SW-P5 gibt es die zusätzliche Möglichkeit das Spiegeln einer Achse eindeutig darzustellen. Ein Spiegeln wird dann nicht auf ein Spiegeln einer Bezugsachse und Verdrehungen von anderen Achsen abgebildet.

Projektiert werden kann diese Einstellung durch:

`MD10610 $MN_MIRROR_REF_AX = 0`

Die Programmierung des programmierbaren Frames mit `MIRROR` und `AMIRROR` wird erweitert. Bisher wurde der angegebene Wert der Koordinatenachse, wie z.B. bei `MIRROR X0` der Wert 0, nicht ausgewertet, sondern `AMIRROR` hatte eine Toggle-Funktion, d. h. `MIRROR X0` schaltet die Spiegelung ein und ein weiteres `AMIRROR X0` schaltet sie wieder aus. `MIRROR` wirkt immer absolut und `AMIRROR` additiv.

Mit der Maschinendatum-Einstellung:

`MD10612 $MN_MIRROR_TOGGLE = 0` ("Mirror Toggle")

kann festgelegt werden, dass die programmierten Werte ausgewertet werden.

Bei einem Wert von 0, wie bei `AMIRROR X0`, wird die Spiegelung der Achse ausgeschaltet, und bei Werte ungleich 0 wird die Achse gespiegelt, wenn sie noch nicht gespiegelt ist.

3.5 Frames

Das komponentenweise Lesen oder Schreiben von Spiegelungen ist unabhängig vom Maschinendatum:

MD10612 \$MN_MIRROR_TOGGLE

Ein Wert = 0 bedeutet, dass danach die Achse nicht gespiegelt ist und ein Wert = 1 heißt, dass die Achse danach immer gespiegelt wird, egal, ob die Achse schon gespiegelt war oder nicht.

```
$P_NCBFR[0,x,mi]=1           ; x-Achse wird immer gespiegelt.
$P_NCBFR[0,x,mi]=0           ; x-Achse spiegeln aus.
```

Achsspezifisches Ersetzen G58, G59

Der Translationsanteil des programmierbaren Frames wird in einen absoluten Anteil und in einen Anteil für die Summe aller additiv-programmierten Translationen aufgespalten. Der absolute Anteil kann über TRANS, CTRANS oder durch Schreiben der Translationskomponente geändert werden, wobei der additive Anteil zu Null gesetzt wird. G58 verändert nur den absoluten Translationsanteil für die angegebene Achse, die Summe der additiv-programmierten Translationen bleibt erhalten.

G58 X... Y... Z... A... ..

G59 dient zum achsspezifischen Überschreiben der additiv programmierten Translationen für die angegebenen Achsen, die mit ATRANS programmiert wurden.

G59 X... Y... Z... A... ..

Beispiel

```
TRANS X10 Y10 Z10
ATrans X5 Y5           ; Gesamttranslationen X15 Y15 Z10
G58 X20                ; Gesamttranslationen X25 Y15 Z10
G59 X10 Y10           ; Gesamttranslationen X30 Y20 Z10
```

G58 und G59 ist nur einsetzbar, wenn:

MD24000 \$MC_FRAME_ADD_COMPONENTS (Framekomponenten für G58 / G59) == TRUE

Nachfolgende Tabelle beschreibt die Wirkung von diversen Programmbefehlen auf die absolute und die additive Translation.

	Grob- bzw. absolute Verschiebung	Fine- bzw. additive Verschiebung
TRANS X10	10	0
ATrans X10	unverändert	alt_fine + 10
CTrans (X, 10)	10	0
CTrans ()	0	0
CFine (X, 10)	0	10
\$P_PFRAME [X, TR] = 10	10	unverändert
\$P_PFRAME [X, FI] = 10	unverändert	10
G58 X10	10	unverändert
G59 X10	unverändert	10

3.5.5.8 Kanalspezifische Systemframes

Kanalspezifische Systemframes werden nur von Systemfunktionen wie Istwertsetzen, Ankratzen, externe Nullpunktverschiebung, Schrägbearbeitung etc. geschrieben.

Maschinendaten

Parametrierung der kanalspezifischen Systemframes

Aus Speicherplatzgründen sollten nur die kanalspezifische Systemframes projiziert werden, deren Systemfunktionen tatsächlich verwendet werden.

Jedes Systemframe belegt pro Kanal ca. 1 kByte statischen und ca. 6 kByte dynamischen Speicher.

Die Parametrierung der kanalspezifischen Systemframes erfolgt in folgenden Maschinendatum:

MD28082 \$MC_MM_SYSTEM_FRAME_MASK, Bit<n>

Bit	Wert	Systemframe vorhanden:
0	1	\$P_SETFR: Istwertsetzen und Ankratzen
1	1	\$P_EXTFR: Externe Nullpunktverschiebung über Systemframe
2	1	\$P_PARTFR: TCARR und PAROT bei orientierbarem Werkzeugträger
3	1	\$P_TOOLFR: TOROT und TOFRAME
4	1	\$P_WPFR: Frame für Werkstückbezugspunkte
5	1	\$P_CYCFR: Frame für Zyklen
6	1	\$P_TRAFR: Frame für An- und Abwahl von Transformationen
7	1	\$P_ISO1FRAME: Frame für G51 . 1 Spiegeln (ISO)
8	1	\$P_ISO2FRAME: Frame für G68 2DROT (ISO)
9	1	\$P_ISO3FRAME: Frame für G68 3DROT (ISO)
10	1	\$P_ISO4FRAME: Frame für G51 Scale (ISO)
11	1	\$P_RELFR: Frame für relative Koordinatensysteme

Parametrierung des ENS (ACS)-Koordinatensystems

Über das folgende Maschinendatum wird festgelegt, welche Systemframes das ENS (ACS)-Koordinatensystem bilden

MD24030 \$MC_FRAME_ACS_SET = <Wert>

<Wert>	Bedeutung: Das ENS (ACS)-Koordinatensystem besteht aus
0	\$P_PARTFRAME : \$P_SETFRAME : \$P_EXTFRAME : \$P_ISO1FRAME : \$P_ISO2FRAME : \$P_ISO3FRAME : \$P_ACTBFRAME : \$P_IFRAME : \$P_GFRAME : \$P_TOOLFRAME : \$P_WPFRAME
1	\$P_PARTFRAME : \$P_SETFRAME : \$P_EXTFRAME : \$P_ISO1FRAME : \$P_ISO2FRAME : \$P_ISO3FRAME : \$P_ACTBFRAME : \$P_IFRAME : \$P_GFRAME : \$P_TOOLFRAME : \$P_WPFRAME : \$P_TRAFRAME : \$P_PFRAME : \$P_ISO4FRAME

Systemvariablen

Kanalspezifische Systemframes der Datenhaltung

Die kanalspezifischen Systemframes der Datenhaltung können über folgende Framevariablen geschrieben / gelesen werden:

Systemvariable	Bedeutung: Systemframe der Datenhaltung für
\$P_SETFR	Istwertsetzen und Ankratzen (Set-Frame)
\$P_EXTFR	Externe Nullpunktverschiebung (Ext-Frame)
\$P_PARTFR	TCARR und PAROT bei orientierbarem Werkzeugträger (Part-Frame)
\$P_TOOLFR	TOROT und TOFRAME (Tool-Frame)
\$P_WPFR	Werkstückbezugspunkte (Work-Piece-Frame)
\$P_CYCFR	Zyklen (Cycle-Frame)
\$P_TRAFR	Transformationen (Transformation-Frame)
\$P_ISO1FR	G51 .1 Spiegeln (ISO)
\$P_ISO2FR	G68 2DROT (ISO)
\$P_ISO3FR	G68 3DROT (ISO)
\$P_ISO4FR	G51 Scale (ISO)
\$P_RELFR	relative Koordinatensysteme

Hinweis

Zyklenprogrammierung

Die Framevariablen der Systemframes sind ausschließlich für die Zyklen-Programmierung vorgesehen. In NC-Programmen sollten die Systemframes daher vom Anwender nicht direkt, sondern nur über Systemfunktionen wie TOROT, PAROT, etc. geschrieben werden.

Kanalspezifische aktive Systemframes

Systemvariablen der aktiven kanalspezifischen Systemframes:

Systemvariable	Bedeutung: Aktiver Systemframe für
\$P_SETFRAME	Istwertsetzen und Ankratzen (Set-Frame)
\$P_EXTFRAME	Externe Nullpunktverschiebung (Ext-Frame)
\$P_PARTFRAME	TCARR und PAROT bei orientierbarem Werkzeugträger (Part-Frame)
\$P_TOOLFRAME	TOROT und TOFRAME (Tool-Frame)
\$P_WPFRAME	Werkstückbezugspunkte (Work-Piece-Frame)
\$P_CYCFRAME	Zyklen (Cycle-Frame)
\$P_TRAFRAME	Transformationen (Transformation-Frame)
\$P_ISO1FRAME	G51 .1 Spiegeln (ISO)
\$P_ISO2FRAME	G68 2DROT (ISO)
\$P_ISO3FRAME	G68 3DROT (ISO)
\$P_ISO4FRAME	G51 Scale (ISO)
\$P_RELFRAME	relative Koordinatensysteme

Ist ein kanalspezifisches Systemframe der Datenhaltung nicht parametrierbar, gilt für das entsprechende aktive Systemframe: \$P_<Systemframe> == Nullframe.

Kanalspezifische aktives ENS (ACS)-Gesamtframe

Über die Systemvariable \$P_ACSFRAME kann gelesen werden, welche Systemframes das ENS (ACS)-Koordinatensystem bilden. Die Festlegung erfolgt über das oben beschriebenen Maschinendatum MD24030 \$MC_FRAME_ACS_SET. Siehe Absatz "Maschinendaten" > "Parametrierung des ENS (ACS)-Koordinatensystems"

Systemvariable	Bedeutung: Aktiver Systemframe für
\$P_ACSFRAME	Systemframes, die das ENS (ACS)-Koordinatensystem entsprechend der Parametrierung in MD24030 \$MC_FRAME_ACS_SET bilden

3.5.6 Implizite Frame-Änderungen**3.5.6.1 Umschalten von Geometrieachsen**

Welche Kanalachsen die Geometrieachsen des Kanals sind, kann sich sowohl beim Ein-/ Ausschalten einer Transformation und beim Befehl `GEOAX()` ändern.

Über das folgende Maschinendatum können vier verschiedenen Einstellungen zur Behandlung des aktuelle Gesamtframe \$P_ACTFRAME vorgenommen werden:

MD10602 \$MN_FRAME_GEOAX_CHANGE_MODE = <Wert>

<Wert>	Bedeutung
0	<p>Löschen</p> <p>Bei Umschaltungen von Geometrieachsen, wie bei An- und Abwahl von Transformationen und <code>GEOAX()</code> wird das aktuelle Gesamtframe gelöscht. Erst nach Aktivieren eines neuen Frames wird dann die geänderte Geometrieachskonstellation berücksichtigt.</p>
1	<p>Neuberechnung mit Erhalt der Drehungen</p> <p>Das aktuelle Gesamtframe wird beim Umschalten von Geometrieachsen neu berechnet, wobei die Frame-Anteile der neuen Geometrieachsen wirksam werden. Die Drehungen der Geometrieachsen, die vor dem Umschalten programmiert wurden, bleiben auch für die neuen Geometrieachsen wirksam.</p> <p>Für <code>TRANSMIT</code>, <code>TRACYL</code> und <code>TRAANG</code> siehe Kapitel "An- und Abwahl von Transformationen: Allgemein (Seite 350)".</p>

3.5 Frames

<Wert>	Bedeutung
2	<p>Neuberechnung nur, wenn keine Drehungen aktiv waren</p> <p>Das aktuelle Gesamtframe wird beim Umschalten von Geometrieachsen neu berechnet, wobei die Frame-Anteile der neuen Geometrieachsen wirksam werden. Sind vor der Umschaltung in den aktuellen Basisframes, dem aktuellen Einstellbaren Frame oder im Programmierbaren Frame, Drehungen aktiv, wird die Umschaltung mit Alarm "Frame: Umschaltung der Geometrieachsen unzulässig" abgebrochen.</p> <p>Für TRANSMIT, TRACYL und TRAANG siehe Kapitel "An- und Abwahl von Transformationen: Allgemein (Seite 350)".</p>
3	<p>Transformation: Löschen / GEOAX(): Neuberechnung mit Erhalt der Drehungen</p> <ul style="list-style-type: none"> Transformation: Das aktuelle Gesamtframe wird bei An- und Abwahl von Transformationen gelöscht. GEOAX(): Bei GEOAX() wird das Gesamtframe neu berechnet, wobei die Translationen, Skalierungen und Spiegelungen der neuen achsspezifischesachsspezifisches wirksam werden. Die Drehungen der Geometrieachsen, die vor dem Umschalten programmiert wurden, bleiben auch für die neuen Geometrieachsen wirksam.

Die Werkstückgeometrie wird von einem Koordinatensystem beschrieben, das von den Geometrieachsen aufgespannt wird. Jeder Geometrieachse ist eine Kanalachse und jeder Kanalachse eine Maschinenachse zugeordnet. Für jedes Frame (Systemframe, Basisframes, einstellbare Frames, prog. Frame) gibt es für jede Maschinenachse ein achsspezifisches Frame. Wird einer Geometrieachse eine andere Maschinenachse zugeordnet, bringt die Maschinenachse ihren eigenen achsspezifischen Frameanteile mit. Die neue Geometrie im Kanal wird dann von den neuen Konturframes gebildet, die sich aus den bis zu drei neuen Geometrieachsen bilden.

Die aktuell gültigen Frames werden bei der Umschaltung von Geometrieachsen neu berechnet und ein resultierendes Gesamtframe daraus gewonnen. Die Datenhaltungsframes werden erst nach Aktivierung berücksichtigt.

Beispiel

Die Kanalachse A soll durch einen GeoAx-Tausch zur Geometrieachse X werden. Durch den Tausch soll das Programmierbare Frame einen Trans-Anteil von 10 in der X-Achse haben. Das aktuelle einstellbare Frame soll erhalten bleiben.

MD10602 \$MN_FRAME_GEOAX_CHANGE_MODE = 1

Programmcode	Kommentar
\$P_UIFR[1] = CROT(X,10,Y,20,Z,30)	; Frame bleibt nach GeoAx-Tausch erhalten.
G54	; Einstellbares Frame wird aktiv.
TRANS A10	; Achsiale Verschiebung von A wird mit getauscht.
GEOAX(1, A)	; A wird zur X-Achse
	; \$P_ACTFRAME = CROT(X,10,Y,20,Z,30) : CTRANS(X10)

Bei Transformationswechsel können gleichzeitig mehrere Kanalachsen zu Geometrieachsen werden.

Beispiel

Durch eine 5-Achs-Orientierungstranformation werden die Kanalachsen 4, 5 und 6 zu Geometrieachsen der Transformation. Die Geometrieachsen vor der Transformation werden

also alle ersetzt. Beim Einschalten der Transformation ändern sich alle aktuellen Frames. Zur Berechnung des neuen WKS-Systems werden die achsialen Frameanteile der Kanalachsen, die zu Geometrieachsen werden, berücksichtigt. Programmierte Drehungen vor der Transformation werden beibehalten. Nach dem Ausschalten der Transformation wird das alte WKS wieder hergestellt. Der häufigste Anwendungsfall wird wohl der sein, dass sich die Geometrieachsen vor und nach der Transformation nicht ändern und die Frames so beibehalten werden sollen, wie sie vor der Transformation waren.

Maschinendaten

Programmcode

```
$MC_AXCONF_CHANAX_NAME_TAB[0] = "CAX"
$MC_AXCONF_CHANAX_NAME_TAB[1] = "CAY"
$MC_AXCONF_CHANAX_NAME_TAB[2] = "CAZ"
$MC_AXCONF_CHANAX_NAME_TAB[3] = "A"
$MC_AXCONF_CHANAX_NAME_TAB[4] = "B"
$MC_AXCONF_CHANAX_NAME_TAB[5] = "C"
```

```
$MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1
$MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 2
$MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3
```

```
$MC_AXCONF_GEOAX_NAME_TAB[0] = "X"
$MC_AXCONF_GEOAX_NAME_TAB[1] = "Y"
$MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"
```

```
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=4
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=5
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=6
```

```
$MC_TRAFO_AXES_IN_1[0]=4
$MC_TRAFO_AXES_IN_1[1]=5
$MC_TRAFO_AXES_IN_1[2]=6
$MC_TRAFO_AXES_IN_1[3]=1
$MC_TRAFO_AXES_IN_1[4]=2
```

Programm:

Programmcode

```
$P_NCBFRAME[0] = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6)
$P_CHBFRAME[0] = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6)
$P_IFRAME = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6):CROT(Z,45)
$P_PFRAME = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6):CROT(X,10,Y,20,Z,30)
```

Programmcode	Kommentar
TRAORI	; Trafo setzt GEOAX(4,5,6)
	; \$P_NCBFRAME[0] = CTRANS(X,4,Y,5,Z,6,CAX,1,CAY,2,CAZ,3)

Programmcode	Kommentar
	; \$P_ACTBFRAME =CTTRANS (X, 8, Y, 10, Z, 12, CAX, 2, CAY, 4, CAZ, 6)
	; \$P_PFRAME =
	CTTRANS (X, 4, Y, 5, Z, 6, CAX, 1, CAY, 2, CAZ, 3) :CROT (X, 10, Y, 20, Z, 30)
	; \$P_IFRAME = CTRANS (X, 4, Y, 5, Z, 6, CAX, 1, CAY, 2, CAZ, 3) :CROT (Z, 45)
TRAFOOF	; Ausschalten der Transformation setzt GEOAX(1,2,3)
	; \$P_NCBFRAME[0] = CTRANS (X, 1, Y, 2, Z, 3, A, 4, B, 5, C, 6)
	; \$P_CHBFRAME[0] = CTRANS (X, 1, Y, 2, Z, 3, A, 4, B, 5, C, 6)
	; \$P_IFRAME = CTRANS (X, 1, Y, 2, Z, 3, A, 4, B, 5, C, 6) :CROT (Z, 45)
	; \$P_PFRAME =
	CTTRANS (X, 1, Y, 2, Z, 3, A, 4, B, 5, C, 6) :CROT (X, 10, Y, 20, Z, 30)

3.5.6.2 An- und Abwahl von Transformationen: Allgemein

Bei An- und Abwahl von Transformationen ändert sich in der Regel die Zuordnung der Geometrieachsen zu den Kanalachsen. Bei Transformationen, bei denen Rundachsen zu Linearachsen und Linearachsen zu Rundachsen werden, ist keine eindeutige Zuordnung von achsspezifischen Frameanteilen zu geometrischen Konturframeanteilen möglich. Bei diesen nichtlinearen Transformationen muss eine Sonderbehandlung zur Aufbereitung des Konturframes erfolgen.

Es wird der Modus, der durch MD10602 \$MN_FRAME_GEOAX_CHANGE_MODE = 1 und 2 eingestellt wird, so erweitert, dass auch die o. g. Transformationen berücksichtigt werden.

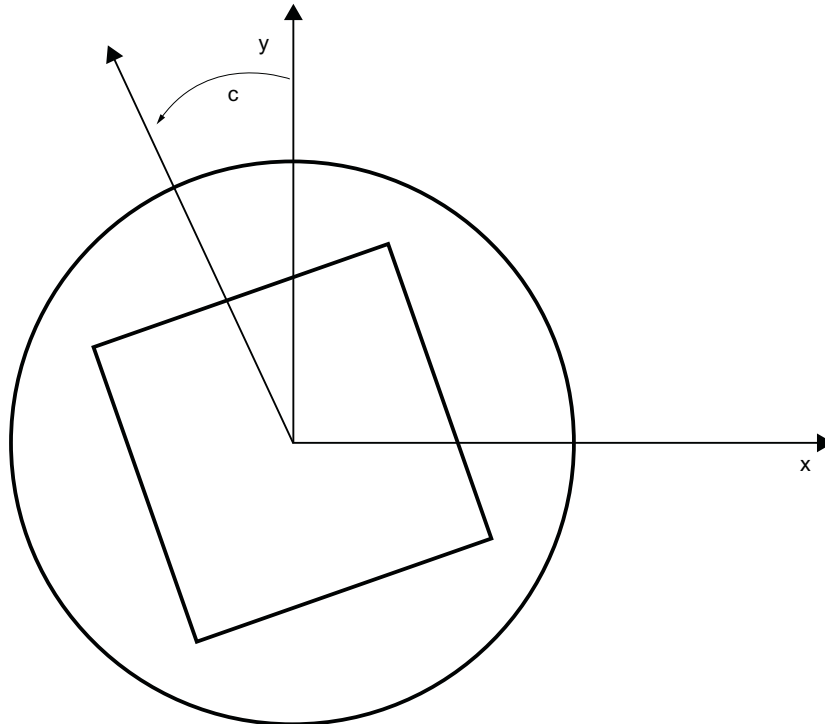
Bei Transformationsanwahl wird das Konturframe anhand der achsspezifischen Frames aufgebaut. Die virtuelle Geometrieachse der Transformationen TRANSMIT, TRACYL und TRAANG unterliegt dabei einer Sonderbehandlung.

Hinweis

Transformationen mit virtuellen Achsen

Bei der Anwahl von TRANSMIT oder TRACYL werden Verschiebungen, Skalierungen und Spiegelungen der realen Y-Achse nicht in die virtuelle Y-Achse übernommen. Verschiebungen, Skalierungen und Spiegelungen der virtuellen Y-Achse werden bei TRAFOOF gelöscht.

3.5.6.3 An- und Abwahl von Transformationen: TRANSMIT

**Transmit-Erweiterungen**

Der achsspezifische Gesamtframe der TRANSMIT-Rundachse, d. h. Translation, Spiegelung und Skalierung, kann über folgende Maschinendaten in der Transformation berücksichtigt werden:

- MD24905 \$MC_TRANSMIT_ROT_AX_FRAME_1 = 1
- MD24955 \$MC_TRANSMIT_ROT_AX_FRAME_2 = 1

Eine Verschiebung der Rundachse kann z. B. durch eine Kompensation der Schräglage eines Werkstückes in einem Frame innerhalb der Framekette eingetragen werden. Diese Verschiebung soll in der Regel auch in der Transformation als Offset der Rundachse berücksichtigt werden. Eine Verschiebung der C-Achse, wie im obigen Bild, führt dann zu entsprechenden X- und Y-Werten.

- MD24905 \$MC_TRANSMIT_ROT_AX_FRAME_1 = 2
- MD24955 \$MC_TRANSMIT_ROT_AX_FRAME_2 = 2

Mit dieser Einstellung wird die achsspezifische Verschiebung der Rundachse bis zum ENS in der Transformation berücksichtigt. Die in den ENS-Frames enthaltenen achsspezifischen Verschiebungen der Rundachse werden in das Transformationsframe als Drehung eingetragen. Diese Einstellung ist nur wirksam, wenn das Transformationsframe projiziert ist.

Frame-Erweiterungen

Nachfolgend beschriebene Erweiterungen gelten nur für folgende Maschinendaten-Einstellungen:

- MD10602 \$MN_FRAME_GEOAX_CHANGE_MODE = 1
- MD10602 \$MN_FRAME_GEOAX_CHANGE_MODE = 2

Mit Anwahl der Transformation TRANSMIT entsteht, gekoppelt über die Rundachse, eine virtuelle Geometrieachse, die keinen Bezug zu einem achsspezifischen Frame hat, sondern nur im Konturframe berücksichtigt wird. Der geometrische Wert ergibt sich aus der Drehung einer Rundachse. Alle anderen Geometrieachsen übernehmen bei Trafo-Anwahl ihre achsspezifischen Anteile.

Komponenten:

- Translationen
Die Translationen der virtuellen Achse werden bei TRANSMIT-Anwahl gelöscht. Die Translationen der Rundachse können in der Transformation berücksichtigt werden.
- Drehungen
Die Drehungen vor der Transformation werden übernommen.
- Spiegelungen
Die Spiegelung der virtuellen Achse wird gelöscht. Die Spiegelung Rundachse kann in der Transformation berücksichtigt werden.
- Skalierungen
Die Skalierung der virtuellen Achse wird gelöscht. Die Skalierung der Rundachse kann in der Transformation berücksichtigt werden.

Beispiel: Maschinendaten

```

; FRAME-Projektierungen

$MC_MM_SYSTEM_FRAME_MASK='H41'           ; TRAFRAME, SETFRAME
$MC_CHSFRAME_RESET_MASK='H41'           ; Frames sind nach Reset aktiv.
$MC_CHSFRAME_POWERON_MASK='H41'        ; Frames werden bei Power On gelöscht.

$MN_FRAME_GEOAX_CHANGE_MODE=1           ; Frames werden nach GeoAx-Umschaltung
                                         umgerechnet.
$MC_RESET_MODE_MASK='H4041'            ; Basisframe wird nach RESET nicht abge-
                                         wählt.
; $MC_RESET_MODE_MASK='H41'            ; Basisframe wird nach RESET abgewählt.

; $MC_GCODE_RESET_VALUES[7]=2          ; G54 ist Voreinstellung.
$MC_GCODE_RESET_VALUES[7]=1           ; G500 ist Voreinstellung.

$MN_MM_NUM_GLOBAL_USER_FRAMES=0
$MN_MM_NUM_GLOBAL_BASE_FRAMES=3

$MC_MM_NUM_USER_FRAMES=10              ; von 5 bis 100
$MC_MM_NUM_BASE_FRAMES=3              ; von 0 bis 8

```



```

$MN_NCBFRAME_RESET_MASK='HFF'
$MC_CHBFRAME_RESET_MASK='HFF'

$MN_MIRROR_REF_AX=0 ; Keine Normierung bei der Spiegelung.
$MN_MIRROR_TOGGLE=0
$MN_MM_FRAME_FINE_TRANS=1 ; Feinverschiebung
$MC_FRAME_ADD_COMPONENTS=TRUE ; G58, G59 ist möglich.

```

; TRANSMIT ist 1. Trafo

```

$MC_TRAFO_TYPE_1=256

$MC_TRAFO_AXES_IN_1[0]=1
$MC_TRAFO_AXES_IN_1[1]=6
$MC_TRAFO_AXES_IN_1[2]=3
$MC_TRAFO_AXES_IN_1[3]=0
$MC_TRAFO_AXES_IN_1[4]=0

$MA_ROT_IS_MODULO[AX6]=TRUE;

$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=6
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3

$MC_TRANSMIT_BASE_TOOL_1[0]=0.0
$MC_TRANSMIT_BASE_TOOL_1[1]=0.0
$MC_TRANSMIT_BASE_TOOL_1[2]=0.0

$MC_TRANSMIT_ROT_AX_OFFSET_1=0.0
$MC_TRANSMIT_ROT_SIGN_IS_PLUS_1=TRUE

$MC_TRANSMIT_ROT_AX_FRAME_1=1

```

; TRANSMIT ist 2. Trafo

```

$MC_TRAFO_TYPE_2=256

$MC_TRAFO_AXES_IN_2[0]=1
$MC_TRAFO_AXES_IN_2[1]=6
$MC_TRAFO_AXES_IN_2[2]=2
$MC_TRAFO_AXES_IN_2[3]=0
$MC_TRAFO_AXES_IN_2[4]=0

$MC_TRAFO_GEOAX_ASSIGN_TAB_2[0]=1
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[1]=6

```

3.5 Frames

```

$MC_TRAFO_GEOAX_ASSIGN_TAB_2[2]=2

$MC_TRANSMIT_BASE_TOOL_2[0]=4.0
$MC_TRANSMIT_BASE_TOOL_2[1]=0.0
$MC_TRANSMIT_BASE_TOOL_2[2]=0.0

$MC_TRANSMIT_ROT_AX_OFFSET_2=19.0
$MC_TRANSMIT_ROT_SIGN_IS_PLUS_2=TRUE

$MC_TRANSMIT_ROT_AX_FRAME_2=1

```

Beispiel: Teileprogramm

```

; Frameeinstellungen
N820 $P_UIFR[1] = ctrans(x,1,y,2,z,3,c,4)
N830 $P_UIFR[1] = $P_UIFR[1] : crot(x,10,y,20,z,30)
N840 $P_UIFR[1] = $P_UIFR[1] : cmirror(x,c)
N850
N860 $P_CHBFR[0] = ctrans(x,10,y,20,z,30,c,15)
N870

; Werkzeuganwahl, Aufspannkompensation, Ebenenanwahl
N890 T2 D1 G54 G17 G90 F5000 G64 SOFT
N900

; Anfahren der Ausgangsstellung
N920 G0 X20 Z10
N930
N940 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,C,15)
N950 setal(61000)
N960 endif
N970 if $P_BFRAME <> $P_CHBFR[0]
N980 setal(61000)
N990 endif
N1000 if $P_IFRAME <>
CTRANS(X,1,Y,2,Z,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1010 setal(61000)
N1020 endif
N1030 if $P_IFRAME <> $P_UIFR[1]
N1040 setal(61000)
N1050 endif
N1060 if $P_ACTFRAME <>
CTRANS(X,11,Y,22,Z,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1070 setal(61000)
N1080 endif

```

```
N1090
N1100 TRANSMIT(2)
N1110
N1120 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,20,CAZ,30,C,15)
N1130 setal(61000)
N1140 endif
N1180 if $P_IFRAME <>
CTRANS(X,1,Y,0,Z,2,CAZ,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1190 setal(61000)
N1200 endif
N1240 if $P_ACTFRAME <>
CTRANS(X,11,Y,0,Z,22,CAZ,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1250 setal(61001)
N1260 endif
N1270
N1280
N1290 $P_UIFR[1,x,tr] = 11
N1300 $P_UIFR[1,y,tr] = 14
N1310
N1320 g54
N1330

; Frame einstellen
N1350 ROT RPL=-45
N1360 ATRANS X-2 Y10
N1370

; Vierkant schrappen
N1390 G1 X10 Y-10 G41 OFFN=1; Aufmass 1mm
N1400 X-10
N1410 Y10
N1420 X10
N1430 Y-10
N1440

; Werkzeugwechsel
N1460 G0 Z20 G40 OFFN=0
N1470 T3 D1 X15 Y-15
N1480 Z10 G41
N1490

; Vierkant schlichten
N1510 G1 X10 Y-10
```

3.5 Frames

```
N1520 X-10
N1530 Y10
N1540 X10
N1550 Y-10
N1560

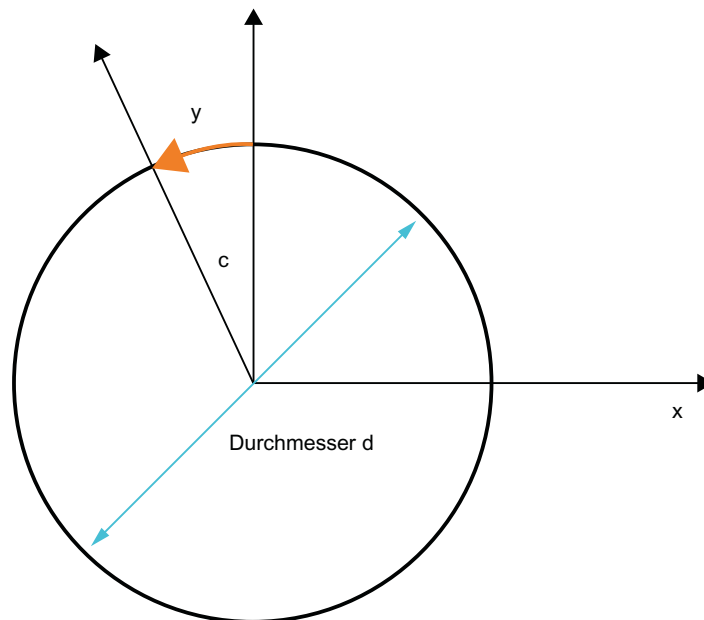
; Frame abwählen
  N2950 m30 N1580 Z20 G40
N1590 TRANS
N1600
N1610 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,20,CAZ,30,C,15)
N1620 setal(61000)
N1630 endif
N1640 if $P_BFRAME <> $P_CHBFR[0]
N1650 setal(61000)
N1660 endif
N1670 if $P_IFRAME <>
TRANS(X,11,Y,0,Z,2,CAZ,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1680 setal(61000)
N1690 endif
N1730 if $P_ACTFRAME <>
TRANS(X,21,Y,0,Z,22,CAZ,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1740 setal(61001)
N1750 endif
N1760
N1770 TRAF00F
N1780
N1790 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,C,15)
N1800 setal(61000)
N1810 endif
N1820 if $P_BFRAME <> $P_CHBFR[0]
N1830 setal(61000)
N1840 endif
N1850 if $P_IFRAME <>
TRANS(X,11,Y,2,Z,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1860 setal(61000)
N1870 endif
N1880 if $P_IFRAME <> $P_UIFR[1]
N1890 setal(61000)
N1900 endif
N1910 if $P_ACTFRAME <>
TRANS(X,21,Y,22,Z,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1920 setal(61002)
N1930 endif
N1940
```

```

N2010 $P_UIFR[1] = ctrans()
N2011 $P_CHBFR[0] = ctrans()
N2020 $P_UIFR[1] = ctrans(x,1,y,2,z,3,c,0)
N2021 G54
N2021 G0 X20 Y0 Z10 C0
N2030 TRANSMIT(1)
N2040 TRANS x10 y20 z30
N2041 ATRANS y200
N2050 G0 X20 Y0 Z10
N2051 if $P_IFRAME <> CTRANS(X,1,Y,0,Z,3,CAY,2)
N2052 setal(61000)
N2053 endif
N2054 if $P_ACTFRAME <> CTRANS(X,11,Y,20,Z,33,CAY,2):CFINE(Y,200)
N2055 setal(61002)
N2056 endif
N2060 TRAFOOF
N2061 if $P_IFRAME <> $P_UIFR[1]
N2062 setal(61000)
N2063 endif
N2064 if $P_ACTFRAME <> CTRANS(X,11,Y,2,Z,33):CFINE(Y,0)
N2065 setal(61002)
N2066 endif

```

3.5.6.4 An- und Abwahl von Transformationen: TRACYL



TRACYL-Erweiterungen

Der achsspezifische Gesamtframe der TRACYL-Rundachse, d. h. die Translation, die Feinverschiebung, die Spiegelung und die Skalierung, kann über folgende Maschinendaten in der Transformation berücksichtigt werden:

- MD24805 \$MC_TRACYL_ROT_AX_FRAME_1 = 1
- MD24855 \$MC_TRACYL_ROT_AX_FRAME_2 = 1

Eine Verschiebung der Rundachse kann z. B. durch eine Kompensation der Schräglage eines Werkstückes in einem Frame innerhalb der Framekette eingetragen werden. Diese Verschiebung soll in der Regel auch in der Transformation als Offset der Rundachse oder als y-Verschiebung berücksichtigt werden. Eine Verschiebung der C-Achse (wie im obigen Bild) führt dann zu entsprechenden X- und Y-Werten.

- MD24805 \$MC_TRACYL_ROT_AX_FRAME_1 = 2
- MD24855 \$MC_TRACYL_ROT_AX_FRAME_2 = 2

Mit dieser Einstellung wird die achsspezifische Verschiebung der Rundachse bis zum ENS in der Transformation berücksichtigt. Die in den ENS-Frames enthaltenen axialen Verschiebungen der Rundachse werden in das Transformationsframe als Verschiebung auf der Manteloberfläche eingetragen. Diese Einstellung ist nur wirksam, wenn das Transformationsframe projiziert ist.

Frame-Erweiterungen

Nachfolgend beschriebene Erweiterungen gelten nur für folgende Maschinendaten-Einstellungen:

- MD10602 \$MN_FRAME_GEOAX_CHANGE_MODE = 1
- MD10602 \$MN_FRAME_GEOAX_CHANGE_MODE = 2

Mit Anwahl der Transformation TRACYL entsteht gekoppelt über die Rundachse eine virtuelle Geometrieachse auf der Manteloberfläche, die keinen Bezug zu einem achsspezifischen Frame hat, sondern nur im Konturframe berücksichtigt wird. Alle Komponenten der virtuellen Geometrieachse werden abgelöscht. Alle anderen Geometrieachsen übernehmen bei Trafo-Anwahl ihre achsspezifischen Anteile.

Komponenten:

- Translationen
Die Translationen der virtuellen Achse werden bei TRACYL-Anwahl gelöscht. Die Translationen der Rundachse können in der Transformation berücksichtigt werden.
- Drehungen
Die Drehungen vor der Transformation werden übernommen.
- Spiegelungen
Die Spiegelung der virtuellen Achse wird gelöscht. Die Spiegelung Rundachse kann in der Transformation berücksichtigt werden.
- Skalierungen
Die Skalierung der virtuellen Achse wird gelöscht. Die Skalierung der Rundachse kann in der Transformation berücksichtigt werden.

Beispiel: Maschinendaten

```
| ; FRAME-Projektierungen
```

```

$MC_MM_SYSTEM_FRAME_MASK = 'H41'           ; TRAFRAME, SETFRAME
$MC_CHSFRAME_RESET_MASK = 'H41'           ; Frames sind nach Reset aktiv.
$MC_CHSFRAME_POWERON_MASK = 'H41'        ; Frames werden bei Power On gelöscht.

$MN_FRAME_GEOAX_CHANGE_MODE = 1           ; Frames werden nach GeoAx-Umschaltung um-
                                           gerechnet.

$MC_RESET_MODE_MASK = 'H4041'            ; Basisframe wird nach Reset nicht abge-
                                           wählt.
; $MC_RESET_MODE_MASK = 'H41'            ; Basisframe wird nach Reset abgewählt.

; $MC_GCODE_RESET_VALUES[7] = 2           ; G54 ist Voreinstellung.
$MC_GCODE_RESET_VALUES[7] = 1           ; G500 ist Voreinstellung.

$MN_MM_NUM_GLOBAL_USER_FRAMES = 0
$MN_MM_NUM_GLOBAL_BASE_FRAMES = 3

$MC_MM_NUM_USER_FRAMES = 10              ; von 5 bis 100
$MC_MM_NUM_BASE_FRAMES = 3              ; von 0 bis 8

$MN_NCBFRAME_RESET_MASK = 'HFF'
$MC_CHBFRAME_RESET_MASK = 'HFF'

$MN_MIRROR_REF_AX = 0                    ; Keine Normierung bei der Spiegelung.
$MN_MIRROR_TOGGLE = 0

$MN_MM_FRAME_FINE_TRANS = 1              ; Feinverschiebung
$MC_FRAME_ADD_COMPONENTS = TRUE          ; G58, G59 ist möglich

; TRACYL mit Nutwandkorrektur ist 3. Trafo

$MC_TRAFO_TYPE_3 = 513; TRACYL

$MC_TRAFO_AXES_IN_3[0] = 1
$MC_TRAFO_AXES_IN_3[1] = 5
$MC_TRAFO_AXES_IN_3[2] = 3
$MC_TRAFO_AXES_IN_3[3] = 2

$MC_TRAFO_GEOAX_ASSIGN_TAB_3[0] = 1
$MC_TRAFO_GEOAX_ASSIGN_TAB_3[1] = 5
$MC_TRAFO_GEOAX_ASSIGN_TAB_3[2] = 3

$MC_TRACYL_BASE_TOOL_1[0] = 0.0
$MC_TRACYL_BASE_TOOL_1[1] = 0.0
$MC_TRACYL_BASE_TOOL_1[2] = 0.0

```

```
$MC_TRACYL_ROT_AX_OFFSET_1 = 0.0  
$MC_TRACYL_ROT_SIGN_IS_PLUS_1 = TRUE  
  
$MC_TRACYL_ROT_AX_FRAME_1 = 1
```

Beispiel: Teileprogramm

```
;Einfacher Verfahrtest mit Nutwandkorrektur  
N450 G603  
N460  
  
; Frameeinstellungen  
N500 $P_UIFR[1] = CTRANS(x,1,y,2,z,3,b,4)  
N510 $P_UIFR[1] = $P_UIFR[1] : CROT(x,10,y,20,z,30)  
N520 $P_UIFR[1] = $P_UIFR[1] : CMIRROR(x,b)  
N530  
N540 $P_CHBFR[0] = CTRANS(x,10,y,20,z,30,b,15)  
N550  
N560 G54  
N570  
  
; Bahnsteuerbetrieb mit angewähltem Überschleifen  
N590 G0 x0 y0 z-10 b0 G90 F50000 T1 D1 G19 G641 ADIS=1 ADISPOS=5  
N600  
N610 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,B,15)  
N620 setal(61000)  
N630 endif  
N640 if $P_BFRAME <> $P_CHBFR[0]  
N650 setal(61000)  
N660 endif  
N670 if $P_IFRAME <>  
TRANS(X,1,Y,2,Z,3,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)  
N680 setal(61000)  
N690 endif  
N700 if $P_IFRAME <> $P_UIFR[1]  
N710 setal(61000)  
N720 endif  
N730 if $P_ACTFRAME <>  
TRANS(X,11,Y,22,Z,33,B,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)  
N740 setal(61000)  
N750 endif  
N760
```



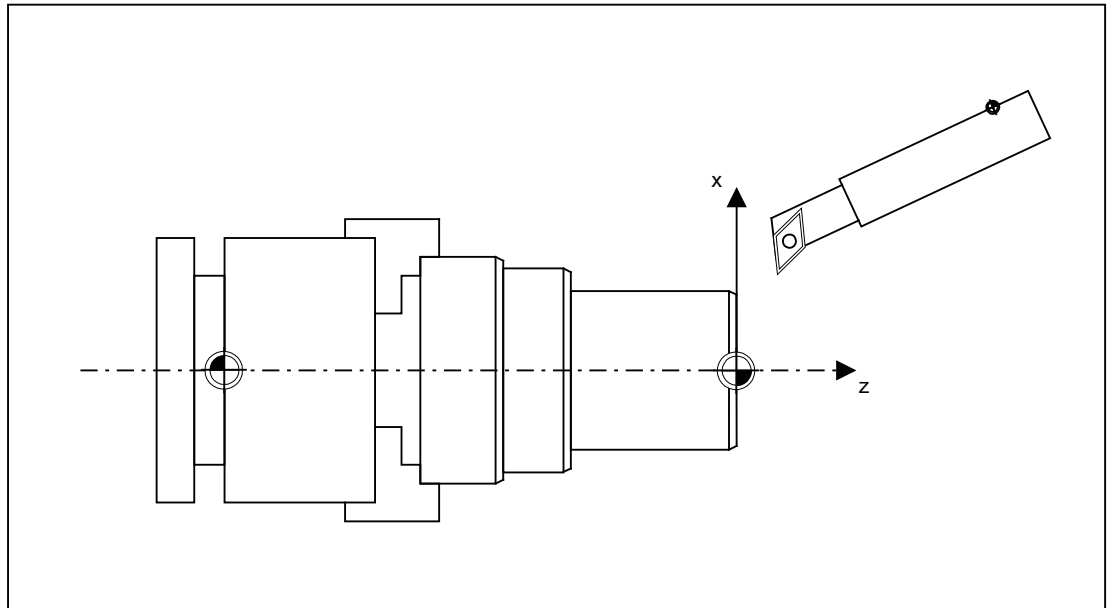
```
; Transformation ein
N780 TRACYL(40.)
N790
N800 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,30,CAY,20,B,15)
N810 setal(61000)
N820 endif
N830 if $P_CHBFR[0] <> CTRANS(X,10,Y,0,Z,30,CAY,20,B,15)
N840 setal(61000)
N850 endif
N860 if $P_IFRAME <>
TRANS(X,1,Y,0,Z,3,CAY,2,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N870 setal(61000)
N880 endif
N890 if $P_UIFR[1] <>
TRANS(X,1,Y,0,Z,3,CAY,2,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N900 setal(61000)
N910 endif
N920 if $P_ACTFRAME <>
TRANS(X,11,Y,0,Z,33,CAY,22,B,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N930 setal(61001)
N940 endif
N950
N960 $P_UIFR[1,x,tr] = 11
N970 $P_UIFR[1,y,tr] = 14
N980
N990 g54
N1000
N1010 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,30,CAY,20,B,15)
N1020 setal(61000)
N1030 endif
N1040 if $P_BFRAME <> $P_CHBFR[0]
N1050 setal(61000)
N1060 endif
N1070 if $P_IFRAME <>
TRANS(X,11,Y,0,Z,3,CAY,2,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N1080 setal(61000)
N1090 endif
N1100 if $P_IFRAME <> $P_UIFR[1]
N1110 setal(61000)
N1120 endif
N1130 if $P_ACTFRAME <>
TRANS(X,21,Y,0,Z,33,CAY,22,B,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N1140 setal(61001)
N1150 endif
```

3.5 Frames

```
N1160

; Transformation aus
N1180 TRAFOOF
N1190
N1200 if $P_BFRAME <> CTRANS (X,10,Y,20,Z,30,B,15)
N1210 setal(61000)
N1220 endif
N1230 if $P_BFRAME <> $P_CHBFR[0]
N1240 setal(61000)
N1250 endif
N1260 if $P_IFRAME <>
TRANS (X,11,Y,2,Z,3,B,4):CROT (X,10,Y,20,Z,30):CMIRROR (X,B)
N1270 setal(61000)
N1280 endif
N1290 if $P_IFRAME <> $P_UIFR[1]
N1300 setal(61000)
N1310 endif
N1320 if $P_ACTFRAME <>
TRANS (X,21,Y,22,Z,33,B,19):CROT (X,10,Y,20,Z,30):CMIRROR (X,B)
N1330 setal(61002)
N1340 endif
N1350
N1360 G00 x0 y0 z0 G90
N1370
N1380 m30
```

3.5.6.5 An- und Abwahl von Transformationen: TRAANG



Frame-Erweiterungen:

Nachfolgend beschriebene Erweiterungen gelten nur für folgende Maschinendaten-Einstellungen:

- MD10602 \$MN_FRAME_GEOAX_CHANGE_MODE = 1
- MD10602 \$MN_FRAME_GEOAX_CHANGE_MODE = 2

Komponenten:

- Translationen
Die Translationen der virtuellen Achse werden bei TRAANG-Anwahl beibehalten.
- Drehungen
Die Drehungen vor der Transformation werden übernommen.
- Spiegelungen
Die Spiegelung der virtuellen Achse wird übernommen.
- Skalierungen
Die Skalierung der virtuellen Achse wird übernommen.

Beispiel: Maschinendaten

```

; FRAME - Projektierungen

$MC_MM_SYSTEM_FRAME_MASK = 'H1'           ; SETFRAME
$MC_CHSFRAME_RESET_MASK = 'H41'          ; Frames sind nach RESET aktiv.
$MC_CHSFRAME_POWERON_MASK = 'H41'        ; Frames werden bei "Power On" gelöscht.

$MN_FRAME_GEOAX_CHANGE_MODE = 1           ; Frames werden nach GeoAx-Umschaltung um-
                                           gerechnet.

```

3.5 Frames

```

$MC_RESET_MODE_MASK = 'H4041' ; Basisframe wird nach RESET nicht abge-
                               wahlt.
; $MC_RESET_MODE_MASK = 'H41' ; Basisframe wird nach RESET abgewahlt.

; $MC_GCODE_RESET_VALUES[7] = 2 ; G54 ist Voreinstellung.
$MC_GCODE_RESET_VALUES[7] = 1 ; G500 ist Voreinstellung.

$MN_MM_NUM_GLOBAL_USER_FRAMES = 0
$MN_MM_NUM_GLOBAL_BASE_FRAMES = 3

$MC_MM_NUM_USER_FRAMES = 10 ; von 5 bis 100
$MC_MM_NUM_BASE_FRAMES = 3 ; von 0 bis 8

$MN_NCBFRAME_RESET_MASK = 'HFF'
$MC_CHBFRAME_RESET_MASK = 'HFF'

$MN_MIRROR_REF_AX = 0 ; Keine Normierung bei der Spiegelung.
$MN_MIRROR_TOGGLE = 0
$MN_MM_FRAME_FINE_TRANS = 1 ; Feinverschiebung
$MC_FRAME_ADD_COMPONENTS = TRUE ; G58, G59 ist moglich.

```

; TRAANG ist 1. Trafo

```

$MC_TRAFO_TYPE_1 = 1024

$MC_TRAFO_AXES_IN_1[0] = 4 ; schrage Achse
$MC_TRAFO_AXES_IN_1[1] = 3 ; Achse parallel zu z
$MC_TRAFO_AXES_IN_1[2] = 2
$MC_TRAFO_AXES_IN_1[3] = 0
$MC_TRAFO_AXES_IN_1[4] = 0

$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0] = 4
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1] = 2
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2] = 3

$MC_TRAANG_ANGLE_1 = 85.
$MC_TRAANG_PARALLEL_VELO_RES_1 = 0.
$MC_TRAANG_PARALLEL_ACCEL_RES_1 = 0.

$MC_TRAANG_BASE_TOOL_1[0] = 0.0
$MC_TRAANG_BASE_TOOL_1[1] = 0.0
$MC_TRAANG_BASE_TOOL_1[2] = 0.0

```

```

; TRAANG ist 2. Trafo

$MC_TRAFO_TYPE_2 = 1024

$MC_TRAFO_AXES_IN_2[0] = 4
$MC_TRAFO_AXES_IN_2[1] = 3
$MC_TRAFO_AXES_IN_2[2] = 0
$MC_TRAFO_AXES_IN_2[3] = 0
$MC_TRAFO_AXES_IN_2[4] = 0

$MC_TRAFO_GEOAX_ASSIGN_TAB_2[0] = 4
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[1] = 0
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[2] = 3

$MC_TRAANG_ANGLE_2 = -85.
$MC_TRAANG_PARALLEL_VELO_RES_2 = 0.2
$MC_TRAANG_PARALLEL_ACCEL_RES_2 = 0.2

$MC_TRAANG_BASE_TOOL_2[0] = 0.0
$MC_TRAANG_BASE_TOOL_2[1] = 0.0
$MC_TRAANG_BASE_TOOL_2[2] = 0.0

```

Beispiel: Teileprogramm

```

; Frameeinstellungen
N820 $P_UIFR[1] = ctrans(x,1,y,2,z,3,b,4,c,5)
N830 $P_UIFR[1] = $P_UIFR[1] : crot(x,10,y,20,z,30)
N840 $P_UIFR[1] = $P_UIFR[1] : cmirror(x,c)
N850
N860 $P_CHBFR[0] = ctrans(x,10,y,20,z,30,b,40,c,15)
N870

; Werkzeuganwahl, Aufspannkompensation, Ebenenanwahl
N890 T2 D1 G54 G17 G90 F5000 G64 SOFT
N900

; Anfahren der Ausgangsstellung
N920 G0 X20 Z10
N930
N940 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,B,40,C,15)
N950 setal(61000)
N960 endif
N970 if $P_BFRAME <> $P_CHBFR[0]
N980 setal(61000)

```

3.5 Frames

```

N990 endif
N1000 if $P_IFRAME <>
TRANS (X,1,Y,2,Z,3,B,4,C,5):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1010 setal(61000)
N1020 endif
N1030 if $P_IFRAME <> $P_UIFR[1]
N1040 setal(61000)
N1050 endif
N1060 if $P_ACTFRAME <>
TRANS (X,11,Y,22,Z,33,B,44,C,20):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1070 setal(61000)
N1080 endif
N1090
N1100 TRAANG(,1)
N1110
N1120 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,CAX,10,B,40,C,15)
N1130 setal(61000)
N1140 endif
N1150 if $P_BFRAME <> $P_CHBFR[0]
N1160 setal(61000)
N1170 endif
N1180 if $P_IFRAME <>
CTRANS(X,1,Y,2,Z,3,CAX,1,B,4,C,5):CROT(X,10,Y,20,Z,30):CMIRROR(X,CAX,C)
N1190 setal(61000)
N1200 endif
N1210 if $P_IFRAME <> $P_UIFR[1]
N1220 setal(61000)
N1230 endif
N1240 if $P_ACTFRAME <>
TRANS (X,11,Y,22,Z,33,CAX,11,B,44,C,20):CROT(X,10,Y,20,Z,30):CMIRROR(X,CAX,
C)
N1250 setal(61001)
N1260 endif
N1270
N1280
N1290 $P_UIFR[1,x,tr] = 11
N1300 $P_UIFR[1,y,tr] = 14
N1310
N1320 g54
N1330

; Frame einstellen
N1350 ROT RPL=-45
N1360 ATRANS X-2 Y10
N1370

```

```
; Vierkant schrappen
N1390 G1 X10 Y-10 G41 OFFN=1; Aufmass 1mm
N1400 X-10
N1410 Y10
N1420 X10
N1430 Y-10
N1440

; Werkzeugwechsel
N1460 G0 Z20 G40 OFFN=0
N1470 T3 D1 X15 Y-15
N1480 Z10 G41
N1490

; Vierkant schlichten
N1510 G1 X10 Y-10
N1520 X-10
N1530 Y10
N1540 X10
N1550 Y-10
N1560

; Frame abwaehlen
N1580 Z20 G40
N1590 TRANS
N1600
N1610 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,CAX,10,B,40,C,15)
N1620 setal(61000)
N1630 endif
N1640 if $P_BFRAME <> $P_CHBFR[0]
N1650 setal(61000)
N1660 endif
N1670 if $P_IFRAME <>
TRANS(X,11,Y,14,Z,3,CAX,1,B,4,C,5):CROT(X,10,Y,20,Z,30):CMIRROR(X,CAX,C)
N1680 setal(61000)
N1690 endif
N1700 if $P_IFRAME <> $P_UIFR[1]
N1710 setal(61000)
N1720 endif
N1730 if $P_ACTFRAME <>
TRANS(X,21,Y,34,Z,33,CAX,11,B,44,C,20):CROT(X,10,Y,20,Z,30):CMIRROR(X,CAX,
C)
N1740 setal(61001)
```

3.5 Frames

```

N1750 endif
N1760
N1770 TRAFOOF
N1780
N1790 if $P_BFRAME <> CTRANS (X,10,Y,20,Z,30,B,40,C,15)
N1800 setal(61000)
N1810 endif
N1820 if $P_BFRAME <> $P_CHBFR[0]
N1830 setal(61000)
N1840 endif
N1850 if $P_IFRAME <>
TRANS (X,1,Y,14,Z,3,B,4,C,5):CROT (X,10,Y,20,Z,30):CMIRROR (X,C)
N1860 setal(61000)
N1870 endif
N1880 if $P_IFRAME <> $P_UIFR[1]
N1890 setal(61000)
N1900 endif
N1910 if $P_ACTFRAME <>
TRANS (X,11,Y,34,Z,33,B,44,C,20):CROT (X,10,Y,20,Z,30):CMIRROR (X,C)
N1920 setal(61002)
N1930 endif
N1940
N1950 m30
    
```

3.5.6.6 Adaptionen von aktiven Frames

Die Geometrie-Achskonstellation kann sich während der Programmbearbeitung oder bei RESET ändern. Die Anzahl der vorhandenen Geometrieachsen können dabei von null bis drei variieren. Bei nicht-vorhandenen Geometrieachsen können Komponenten in den aktiven Frames (z.B. Drehungen) dazu führen, dass die aktiven Frames für diese Achskonstellation ungültig werden. Dies wird durch Alarm "Kanal %1 Satz %2 Rotation fuer nicht vorhandene Geometrieachse programmiert" angezeigt. Der Alarm bleibt solange anstehen, bis die Frames entsprechend geändert wurden.

Mit folgendem Maschinendatum kann die automatische Anpassung von aktiven Frames eingeschaltet werden:

```
MD24040 $MC_FRAME_ADAPT_MODE, Bit<n> = <Wert>
```

Bit	<Wert>	Bedeutung
0	1	Drehungen in aktiven Frames, die Koordinatenachsen verdrehen, fuer die es keine Geometrieachsen gibt, werden aus den aktiven Frames geloescht.
1	1	Scherungswinkel in aktiven Frames werden orthogonalisiert.
2	1	Skalierungen aller Geometrieachsen in den aktiven Frames werden auf den Wert 1 gesetzt.

Über folgendes Maschinendatum werden alle Drehungen in den aktiven Frames gelöscht, die zu Koordinaten-Achsbewegungen für nicht vorhandene Geometrieachsen führen könnten:

MD24040 \$MC_FRAME_ADAPT_MODE = 1

Die Datenhaltungsframes werden dabei nicht verändert. Bei der Aktivierung von Datenhaltungsframes, werden dann auch nur die möglichen Drehungen übernommen.

Beispiel

Es existiert keine Y-Achse:

- MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1
- MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0
- MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3
- \$P_UIFR[1] = CROT(X,45,Y,45,Z,45)

Programmcode	; Kommentar
N390 G54 G0 X10 Z10 F10000	
IF \$P_IFRAME <> CROT(Y,45)	; es wird nur die Drehung um Y übernommen
SETAL(61000)	
ENDIF	

3.5.6.7 Mapped Frames

Übersicht

Die Funktion "Mapped Frames" unterstützt die kanalübergreifende konsistente Änderung achsspezifischer Frames innerhalb kanalspezifischer oder globaler Datenhaltungsframes. In achsspezifischen Maschinendaten wird dazu festgelegt zwischen welchen Achsen das Mapping erfolgen soll.

Ist das Frame-Mapping z.B. für die Maschinenachsen AX1 und AX4 aktiv und es wird in einem kanalspezifischen Datenhaltungsframe (z.B. Basisframe \$P_CHBFR[x]) der achsspezifische Frame der Achse AX1 verändert (Translation, Fine-Translation, Skalierung, Spiegelung), werden diese Frame-Daten für AX1 und AX4 auf alle kanalspezifischen Datenhaltungsframes (z.B. Basisframe \$P_CHBFR[x]) in allen Kanälen in denen sie als Kanalachsen parametrisiert sind, übertragen.

Kein Frame-Mapping erfolgt bei Änderung der achsspezifischen Frame-Daten für die Drehung.

Voraussetzungen

Für das Frame-Mapping müssen folgende Voraussetzungen erfüllt sein:

- Die Datenhaltungsframes die für das Mapping verwendet werden, müssen projiziert sein: MD28083 \$MC_MM_SYSTEM_DATAFRAME_MASK (Systemframes)
- **Kanalspezifische** Datenhaltungsframes müssen für das Mapping explizit freigegeben werden: MD10616 \$MN_MAPPED_FRAME_MASK (Freigabe Frame-Mapping)

Hinweis

Bei **globalen** Datenhaltungsframes wird das Mapping immer durchgeführt. Eine Freigabe ist nicht erforderlich.

Parametrierung

Die Parametrierung der Mapping-Beziehungen erfolgt in den achsspezifischen Maschinendaten:

MD32075 \$MA_MAPPED_FRAME[<AXn>] = "AXm"

AXn, AXm: Maschinenachsname mit n, m = 1, 2, ... max. Anzahl Maschinenachsen

Mapping-Regeln

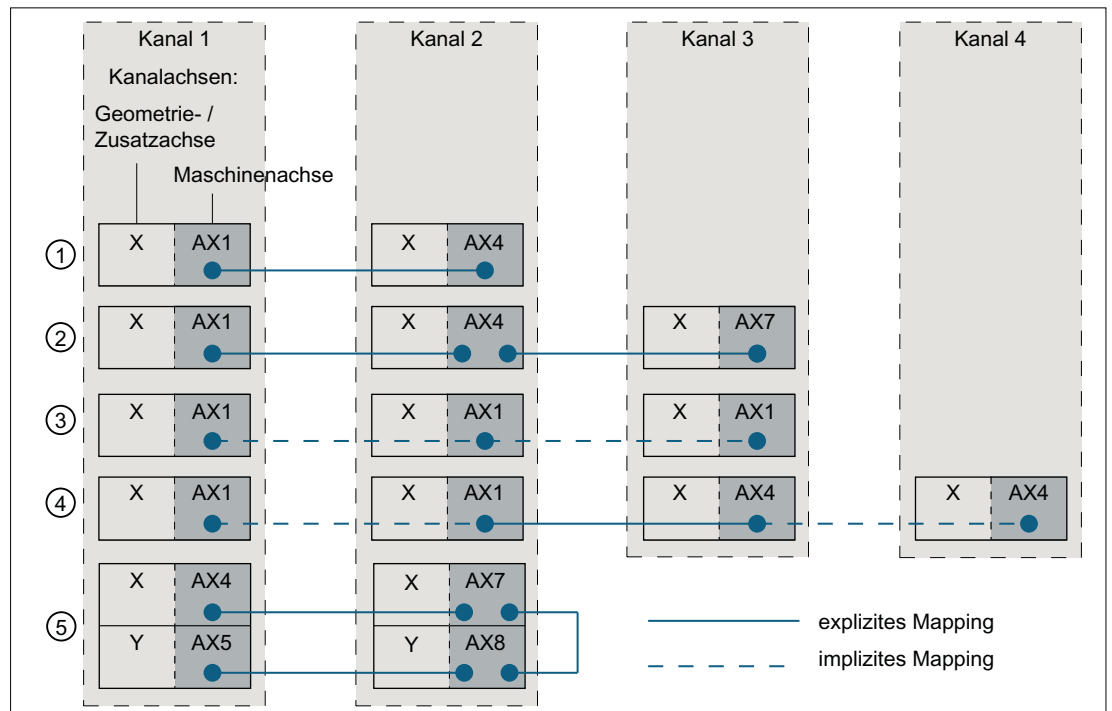
Für das Frame-Mapping gelten folgende Regeln:

- Das Mapping ist bidirektional.
Ein achsspezifischer Frame kann für die Achse AXn oder AXm geschrieben werden. Die Frame-Daten werden immer für die jeweils andere Achse übernommen.
- Es werden immer alle parametrierten Mapping-Beziehungen ausgewertet.
Beim Schreiben eines achsspezifischen Frames einer Achsen AXn werden alle Mapping-Beziehungen ausgewertet und die Frame-Daten für alle direkt und indirekt betroffenen Achsen übernommen.
- Das Mapping ist kanalglobal.
Beim Schreiben eines achsspezifischen Frames der Achse AXn oder AXm für einen kanalspezifischen Frame, werden die Frame-Daten für alle Kanäle übernommen in denen AXn oder AXm als Kanalachsen parametrier sind.
- Beim Schreiben eines achsspezifischen Frames mittels Geometrie- oder Zusatzachsname, erfolgt die Auswertung der Mapping-Beziehungen über die der Geometrie- oder Zusatzachse aktuell zugeordneten Maschinenachse.
- Das Mapping ist Frame-spezifisch.
Beim Schreiben eines achsspezifischen Frames erfolgt das Mapping der Frame-Daten nur innerhalb des gleichen kanalspezifischen oder globalen Datenhaltungsframes.

Hinweis

Datenkonsistenz

Es liegt ausschließlich in der Verantwortung des Anwenders/Maschinenhersteller, z.B. durch Kanalsynchronisation, dass nach dem Schreiben eines Frames in allen Kanälen konsistente Frame-Daten vorliegen.

**Beschreibung**

- ① Einfache Mapping-Beziehung:
AX1(K1) ↔ AX4(K2)
- ② Verkettete Mapping-Beziehungen:
AX1(K1) ↔ AX4(K2) ↔ AX7(K3)
- ③ Mapping-Beziehung auf sich selbst, mit AX1 als
Kanalachse von Kanal 1, 2 und 3:
AX1(K1+K2+K3)
- ④ Mapping-Beziehung zwischen zwei Achsen, die
Kanalachsen in jeweils zwei Kanälen sind:
AX1(K1+K2) ↔ AX4(K3+K4)
- ⑤ Verkettete Mapping-Beziehungen bei der meh-
rere Kanalachsen im gleichen Kanal geschrie-
ben werden:
AX4(K1) ↔ AX7(K2) ↔ AX8(K2) ↔ AX5(K1)

Parametrierung: \$MA_

- MAPPED_FRAME[<AX1>] = "AX4"
- MAPPED_FRAME[<AX1>] = "AX4"
MAPPED_FRAME[<AX4>] = "AX7"
- MAPPED_FRAME[<AX1>] = "AX1"
- MAPPED_FRAME[<AX1>] = "AX4"
- MAPPED_FRAME[<AX4>] = "AX7"
MAPPED_FRAME[<AX7>] = "AX8"
MAPPED_FRAME[<AX8>] = "AX5"

Bild 3-25 Mapping Beispiele

Aktivieren der Datenhaltungsframes

Die Datenhaltungsframes können im Teileprogramm und über die Bedienoberfläche von SINUMERIK Operate geschrieben werden. Bei der Aktivierung der direkt und über Frame-Mapping geschriebenen Datenhaltungsframes in den Kanälen ist folgendes zu beachten:

- Schreiben im Teileprogramm
Die Datenhaltungsframes müssen in jedem Kanal explizit (G500, G54 ... G599) aktiviert werden
- Schreiben über Bedienoberfläche
Datenhaltungsframes werden über die Bedienoberfläche z.B. durch die Eingabe neuer Nullpunktverschiebungen geschrieben. Ein veränderter Datenhaltungsframe wird in allen betroffenen Kanälen sofort aktiv, wenn sich keiner dieser Kanäle im Zustand "Kanal aktiv" befindet. Ist einer der betroffenen Kanäle im Zustand "Kanal aktiv", wird der Datenhaltungsframe in keinem Kanal aktiv. Die Aktivierung muss dann in jedem Kanal explizit im Teileprogramm (G500, G54 ... G599) programmiert werden. Oder er wird mit dem nächsten Wechsel des Kanalzustandes nach "Kanal-Reset" aktiv.

Beispiel

An einer Steuerung sind folgende Kanäle und Kanalachsen parametrier:

- Kanal 1
 - Z: Geometrieachse
 - AX1: Maschinenachse der Geometrieachse Z
- Kanal 2
 - Z: Geometrieachse
 - AX4: Maschinenachse der Geometrieachse Z

Der Nullpunkt der Z-Achse soll in beiden Kanäle immer gleich sein:

- Mapping-Beziehung: \$MA_MAPPED_FRAME[AX1] = "AX4"

Teileprogramme im Kanal 1 und 2

Kanal 1	Kanal 2
...	...
N100 WAIT (10,1,2)	N200 WAIT (10,1,2)
N110 \$P_UIFR[1] = CTRANS(Z, 10)	
N120 WAIT (20,1,2)	N220 WAIT (20,1,2)
N130 G54	N230 G54
N140 IF (\$P_IFRAME[0, Z, TR] <> 10)	N230 IF (\$P_IFRAME[0, Z, TR] <> 10)
N150 SETAL(61000)	N250 SETAL(61000)
N160 ENDIF	N260 ENDIF
...	...

Beschreibung:	
N100 / N200	Kanalsynchronisation für konsistentes Schreiben und Mapping der Frame-Daten
N110	Schreiben des einstellbaren Datenhaltungs-Frames \$P_UIFR[1]: Verschieben des Nullpunktes der Z-Achse auf 10 mm Mapping der achsspezifischen Frame-Daten: Kanal1: $Z \triangleq AX1 \Leftrightarrow$ Kanal2: $Z \triangleq AX4$
N120 / N220	Kanalsynchronisation für konsistentes Aktivieren der neuen Frame-Daten
N130 / N230	Aktivieren der neuen Frame-Daten
N140 / N240	Überprüfung des Nullpunktes der Z-Achse auf: 10 mm

3.5.6.8 Mapped Frames

Übersicht

Die Funktion "Mapped Frames" unterstützt die kanalübergreifende konsistente Änderung achsspezifischer Frames innerhalb kanalspezifischer oder globaler Datenhaltungsframes. In achsspezifischen Maschinendaten wird dazu festgelegt zwischen welchen Achsen das Mapping erfolgen soll.

Ist das Frame-Mapping z.B. für die Maschinenachsen AX1 und AX4 aktiv und es wird in einem kanalspezifischen Datenhaltungsframe (z.B. Basisframe \$P_CHBFR[x]) der achsspezifische Frame der Achse AX1 verändert (Translation, Fine-Translation, Skalierung, Spiegelung), werden diese Frame-Daten für AX1 und AX4 auf alle kanalspezifischen Datenhaltungsframes (z.B. Basisframe \$P_CHBFR[x]) in allen Kanälen in denen sie als Kanalachsen parametrier sind, übertragen.

Kein Frame-Mapping erfolgt bei Änderung der achsspezifischen Frame-Daten für die Drehung.

Voraussetzungen

Für das Frame-Mapping müssen folgende Voraussetzungen erfüllt sein:

- Die Datenhaltungsframes die für das Mapping verwendet werden, müssen projiziert sein: MD28083 \$MC_MM_SYSTEM_DATAFRAME_MASK (Systemframes)
- **Kanalspezifische** Datenhaltungsframes müssen für das Mapping explizit freigegeben werden:
MD10616 \$MN_MAPPED_FRAME_MASK (Freigabe Frame-Mapping)

Hinweis

Bei **globalen** Datenhaltungsframes wird das Mapping immer durchgeführt. Eine Freigabe ist nicht erforderlich.

Parametrierung

Die Parametrierung der Mapping-Beziehungen erfolgt in den achsspezifischen Maschinendaten:

MD32075 \$MA_MAPPED_FRAME[<AXn>] = "AXm"

AXn, AXm: Maschinenachsname mit n, m = 1, 2, ... max. Anzahl Maschinenachsen

Mapping-Regeln

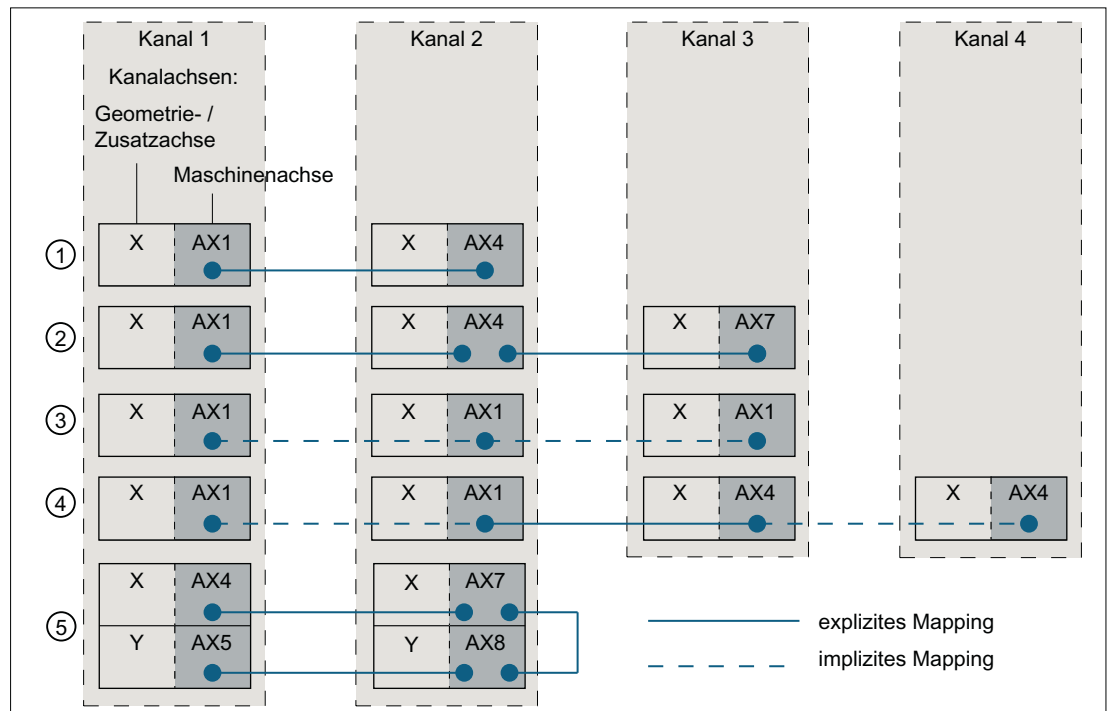
Für das Frame-Mapping gelten folgende Regeln:

- Das Mapping ist bidirektional.
Ein achsspezifischer Frame kann für die Achse AXn oder AXm geschrieben werden. Die Frame-Daten werden immer für die jeweils andere Achse übernommen.
- Es werden immer alle parametrisierten Mapping-Beziehungen ausgewertet.
Beim Schreiben eines achsspezifischen Frames einer Achsen AXn werden alle Mapping-Beziehungen ausgewertet und die Frame-Daten für alle direkt und indirekt betroffenen Achsen übernommen.
- Das Mapping ist kanalglobal.
Beim Schreiben eines achsspezifischen Frames der Achse AXn oder AXm für einen kanalspezifischen Frame, werden die Frame-Daten für alle Kanäle übernommen in denen AXn oder AXm als Kanalachsen parametrisiert sind.
- Beim Schreiben eines achsspezifischen Frames mittels Geometrie- oder Zusatzachsname, erfolgt die Auswertung der Mapping-Beziehungen über die der Geometrie- oder Zusatzachse aktuell zugeordneten Maschinenachse.
- Das Mapping ist Frame-spezifisch.
Beim Schreiben eines achsspezifischen Frames erfolgt das Mapping der Frame-Daten nur innerhalb des gleichen kanalspezifischen oder globalen Datenhaltungsframes.

Hinweis

Datenkonsistenz

Es liegt ausschließlich in der Verantwortung des Anwenders/Maschinenhersteller, z.B. durch Kanalsynchronisation, dass nach dem Schreiben eines Frames in allen Kanälen konsistente Frame-Daten vorliegen.

**Beschreibung**

- ① Einfache Mapping-Beziehung:
AX1(K1) ↔ AX4(K2)
- ② Verkettete Mapping-Beziehungen:
AX1(K1) ↔ AX4(K2) ↔ AX7(K3)
- ③ Mapping-Beziehung auf sich selbst, mit AX1 als Kanalachse von Kanal 1, 2 und 3:
AX1(K1+K2+K3)
- ④ Mapping-Beziehung zwischen zwei Achsen, die Kanalachsen in jeweils zwei Kanälen sind:
AX1(K1+K2) ↔ AX4(K3+K4)
- ⑤ Verkettete Mapping-Beziehungen bei der mehrere Kanalachsen im gleichen Kanal geschrieben werden:
AX4(K1) ↔ AX7(K2) ↔ AX8(K2) ↔ AX5(K1)

Parametrierung: \$MA_

- MAPPED_FRAME[<AX1>] = "AX4"
- MAPPED_FRAME[<AX1>] = "AX4"
MAPPED_FRAME[<AX4>] = "AX7"
- MAPPED_FRAME[<AX1>] = "AX1"
- MAPPED_FRAME[<AX1>] = "AX4"
- MAPPED_FRAME[<AX4>] = "AX7"
MAPPED_FRAME[<AX7>] = "AX8"
MAPPED_FRAME[<AX8>] = "AX5"

Bild 3-26 Mapping Beispiele

Aktivieren der Datenhaltungsframes

Die Datenhaltungsframes können im Teileprogramm geschrieben werden. Bei der Aktivierung der direkt und über Frame-Mapping geschriebenen Datenhaltungsframes in den Kanälen ist folgendes zu beachten:

- Schreiben im Teileprogramm
Die Datenhaltungsframes müssen in jedem Kanal explizit (G500, G54 ... G599) aktiviert werden

Beispiel

An einer Steuerung sind folgende Kanäle und Kanalachsen parametrier:

- Kanal 1
 - Z: Geometrieachse
 - AX1: Maschinenachse der Geometrieachse Z
- Kanal 2
 - Z: Geometrieachse
 - AX4: Maschinenachse der Geometrieachse Z

Der Nullpunkt der Z-Achse soll in beiden Kanäle immer gleich sein:

- Mapping-Beziehung: \$MA_MAPPED_FRAME[AX1] = "AX4"

Teileprogramme im Kanal 1 und 2

Kanal 1	Kanal 2
...	...
N100 WAIT (10,1,2)	N200 WAIT (10,1,2)
N110 \$P_UIFR[1] = CTRANS(Z, 10)	
N120 WAIT (20,1,2)	N220 WAIT (20,1,2)
N130 G54	N230 G54
N140 IF (\$P_IFRAME[0, Z, TR] <> 10)	N230 IF (\$P_IFRAME[0, Z, TR] <> 10)
N150 SETAL(61000)	N250 SETAL(61000)
N160 ENDIF	N260 ENDIF
...	...

Beschreibung:

- N100 / N200 Kanalsynchronisation für konsistentes Schreiben und Mapping der Frame-Daten
- N110 Schreiben des einstellbaren Datenhaltungs-Frames \$P_UIFR[1]:
Verschieben des Nullpunktes der Z-Achse auf 10 mm
Mapping der achsspezifischen Frame-Daten:
Kanal1: Z ≙ AX1 ⇔ Kanal2: Z ≙ AX4
- N120 / N220 Kanalsynchronisation für konsistentes Aktivieren der neuen Frame-Daten
- N130 / N230 Aktivieren der neuen Frame-Daten
- N140 / N240 Überprüfung des Nullpunktes der Z-Achse auf: 10 mm

3.5.7 Vordefinierte Frame-Funktionen

3.5.7.1 Inverses Frame

Die Funktion INVFRAME() berechnet aus einem Frame den entsprechenden inversen Frame.

Funktionsbeschreibung

Die Frame-Verkettung eines Frames mit seinem inversen Frame ergibt immer einen Nullframe.

`FRAME : INVFRAME (FRAME) ⇒ Null-Frame`

Die Frame-Invertierung ist ein Hilfsmittel für die Koordinatentransformationen. Die Berechnung von Messframes erfolgt meist im WKS. Möchte man dieses berechnete Frame in ein anderes Koordinatensystem transformieren, d. h., das berechnete Frame soll in einen beliebigen Frame innerhalb der Frame-Kette eingetragen werden, so ist dies mit folgenden Berechnungen möglich:

Das neue Gesamtframe ergibt sich als Verkettung des alten Gesamtframes mit dem berechneten Frame:

`$P_ACTFRAME = $P_ACTFRAME : $AC_MEAS_FRAME`

Das neue Frame in der Framekette ergibt sich danach:

- Zielframe ist `$P_SETFRAME`:

`$P_SETFRAME = $P_ACTFRAME : $AC_MEAS_FRAME :
INVFRAME ($P_ACTFRAME) : $P_SETFRAME`

- Ziel-Frame ist n-tes Kanalbasisframe `$P_CHBFRAME[<n>]`:

`k = $MN_MM_NUM_GLOBAL_BASE_FRAMES`

- Bei `n = 0` ergibt sich TMP zu:

`TMP = $P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME :
$P_NCBFRAME [<0...k>]`

- Bei `n ≠ 0` ergibt sich TMP zu:

`TMP = $P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME :
$P_NCBFRAME [<0...k>] : $P_CHBFRAME [<0...n> - 1]`

`$P_CHBFRAME [<n>] = INVFRAME (TMP) : $P_ACTFRAME : $AC_MEAS_FRAME :
INVFRAME ($P_ACTFRAME) : TMP : $P_CHBFRAME [<n>]`

- Ziel-Frame ist `$P_IFRAME`:

`TMP = $P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_BFRAME
$P_IFRAME = INVFRAME (TMP) : $P_ACTFRAME : $AC_MEAS_FRAME :
INVFRAME ($P_ACTFRAME) : TMP : $P_IFRAME`

Beispiel

Ein Frame, der z. B. über eine Messfunktion ermittelt wurde, soll im aktuellen `SETFRAME` so eingetragen werden, dass das neue Gesamtframe sich als Verkettung des alten Gesamtframes mit dem Mess-Frame ergibt. Das `SETFRAME` wird mit Hilfe von Frame-Invertierungen entsprechend umgerechnet.

Programmcode	Kommentar
<code>DEF INT RETVAL</code>	
<code>DEF FRAME TMP</code>	
<code>\$TC_DP1 [1,1]=120</code>	; Typ
<code>\$TC_DP2 [1,1]=20.</code>	; 0
<code>\$TC_DP3 [1,1]= 10.</code>	; (z) Laengenkorrekturvektor
<code>\$TC_DP4 [1,1]= 0.</code>	; (y)
<code>\$TC_DP5 [1,1]= 0.</code>	; (x)

3.5 Frames

Programmcode	Kommentar
\$TC_DP6[1,1]= 2.	; Radius
T1 D1	
g0 x0 y0 z0 f10000	
G54	
\$P_CHBFRAME[0] = CROT(Z,45)	
\$P_IFRAME[X,TR] = -SIN(45)	
\$P_IFRAME[Y,TR] = -SIN(45)	
\$P_PFRAME[Z,TR] = -45	
\$AC_MEAS_VALID = 0	; Ecke mit 4 Messpunkten vermessen
G1 X-1 Y-3	; 1. Messpunkt anfahren
\$AC_MEAS_LATCH[0] = 1	; 1. Messpunkt abspeichern
G1 X5 Y-3	; 2. Messpunkt anfahren
\$AC_MEAS_LATCH[1] = 1	; 2. Messpunkt abspeichern
G1 X-4 Y4	; 3. Messpunkt anfahren
\$AC_MEAS_LATCH[2] = 1	; 3. Messpunkt abspeichern
G1 X-4 Y1	; 4. Messpunkt anfahren
\$AC_MEAS_LATCH[3] = 1	; 4. Messpunkt abspeichern
\$AA_MEAS_SETPOINT[X] = 0	; Sollposition der Ecke setzen
\$AA_MEAS_SETPOINT[Y] = 0	
\$AA_MEAS_SETPOINT[Z] = 0	
\$AC_MEAS_CORNER_SETANGLE = 90	; Sollschnittwinkel vorgeben
\$AC_MEAS_WP_SETANGLE = 30	
\$AC_MEAS_ACT_PLANE = 0	; Ebene fuer die Messung ist G17
\$AC_MEAS_T_NUMBER = 1	; Werkzeug auswaehlen
\$AC_MEAS_D_NUMBER = 1	
\$AC_MEAS_TYPE = 4	; Messtype auf Ecke 1 setzen
RETVAl = MEASURE()	; Messvorgang starten
IF RETVAL <> 0	
SETAl(61043, << RETVAL)	
ENDIF	
IF \$AC_MEAS_WP_ANGLE <> 30	
SETAl(61043, << \$AC_MEAS_WP_ANGLE)	
ENDIF	
IF \$AC_MEAS_CORNER_ANGLE <> 90	
SETAl(61043, << \$AC_MEAS_CORNER_ANGLE)	
ENDIF	
; Gemessenes Frame so transformieren und nach \$P_SETFRAME schreiben, dass ein Gesamtframe entsteht,	
; welches aus dem alten Gesamtframe verkettet mit dem Messframe ergibt.	
\$P_SETFRAME = \$P_ACTFRAME : \$AC_MEAS_FRAME : INVFRAME(\$P_ACTFRAME) : \$P_SETFRAME	
\$P_SETFR = \$P_SETFRAME	; Systemframe in der Datenhaltung beschreiben
G1 X0 Y0	; Fahre die Ecke an
G1 X10	; Um 30 Grad gedrehtes Rechteck abfahren
Y10	

Programmcode	Kommentar
X0	
Y0	
M30	

3.5.7.2 Additives Frame in der Framekette

Durch Messungen am Werkstück oder durch Berechnungen im Teileprogramm oder Zyklus ergibt sich oftmals ein Frame, der additiv zum aktiven Gesamtframe wirken sollen. Dadurch sollen z.B. das WKS und damit der Nullpunkt der Programmierung verschoben und/oder gedreht werden. Der gemessene Frame liegt dabei in einer Frame-Variablen vor und ist noch nicht in die Framekette eingerechnet worden.

Funktionsbeschreibung

Die Funktion `ADDFRAME()` berechnet aus dem temporären Frame den angegebenen Zielframe, der durch den Parameter `<STRING>` spezifiziert ist, so, dass sich das neue aktive Gesamtframe `$P_ACTFRAME` aus der Verkettung des alten aktiven Gesamtframes mit dem temporären Frame ergibt:

`ERG = ADDFRAME(TMPFRAME, "$P_SETFRAME") => $P_SETFRAMEneu = $P_SETFRAMEalt
 ADD TMPFRAME und $P_ACTFRAMEneu = $P_ACTFRAMEalt : TMPFRAME`

Wurde als Zielframe ein aktiver Frame angegeben, wird das neue Gesamtframe **im Vorlauf** aktiv.

Ist das Zielframe ein Datenhaltungs-Frame, wird der Frame erst aktiv, wenn er im Kanal explizit, z.B. Teileprogramm, Zyklus, aktiviert wird.

Die Funktion gibt einen Rückgabewert zurück, auf den anwenderspezifisch, z.B. in einem Anwender-Zyklus, reagiert werden kann.

Programmierung

Syntax

`INT ADDFRAME (<FRAME>, <STRING>)`

Bedeutung

<FRAME>:	Frame-Variable mit den additiv einzurechnenden Werten	
	Typ	FRAME

<STRING>:	Name eines aktiven oder Datenhaltungs-Frames:	
	<ul style="list-style-type: none"> • Aktive Frames "\$P_CYCFRAME", "\$P_ISO4FRAME", "\$P_PFRAME", "\$P_WPFRAME", "\$P_TOOLFRAME", "\$P_IFRAME", "\$P_GFRAME", "\$P_CHBFRAME[<n>]", "\$P_NCBFRAME[<n>]", "\$P_ISO1FRAME", "\$P_ISO2FRAME", "\$P_ISO3FRAME", "\$P_EXTFRAME", "\$P_SETFRAME", "\$P_PARTFRAME" • Datenhaltungs-Frames "\$P_CYCFR", "\$P_ISO4FR", "\$P_TRAFR", "\$P_WPFR", "\$P_TOOLFR", "\$P_UIFR[<n>]", "\$P_GFR", "\$P_CHBFR[<n>]", "\$P_NCBFR[<n>]", "\$P_ISO1FR", "\$P_ISO2FR", "\$P_ISO3FR", "\$P_EXTFR", "\$P_SETFR", "\$P_PARTFR" 	
	Typ	STRING
Rückgabewert:	Mögliche Rückgabewerte:	
	<ul style="list-style-type: none"> • 0: OK • 1: Zielangabe (String) ist falsch • 2: Zielframe ist nicht projiziert • 3: Drehung im Frame ist nicht erlaubt 	
	Typ	INT

3.5.8 Funktionen

3.5.8.1 Setzen von Nullpunkten, Werkstück- und Werkzeugvermessung

Das Iswertsetzen erfolgt über die HMI-Bedienung oder über Messzyklen. Das berechnete Frame wird in das Systemframe `SETFRAME` geschrieben werden. Beim Istwertsetzen kann die Sollposition einer Achse im WKS geändert werden.

Unter dem Begriff "Ankratzen" verstehen wir Werkstück- und Werkzeugvermessung. Bei der Werkstückvermessung kann die Lage des Werkstückes bzgl. einer Kante, einer Ecke oder einer Bohrung vermessen werden. Zur Festlegung der Nulllage des Werkstückes oder der Bohrung können die gemessenen Positionen dann mit Sollpositionen im WKS beaufschlagt werden. Die resultierenden Verschiebungen können dabei in ein ausgewähltes Frame eingetragen werden. Bei der Werkzeugvermessung kann die Länge oder der Radius eines Werkzeuges anhand eines vermessenen Referenzteiles gemessen werden.

Die Messungen können über die Bedienung oder über Messzyklen erfolgen. Als Kommunikation mit dem NC dienen vordefinierte Systemvariablen. Die Berechnung erfolgt im NC durch Aktivierung eines PI-Dienstes über die HMI-Bedienung oder über einen Teileprogrammbehehl aus den Messzyklen. Als Grundlage für die Berechnung kann ein Werkzeug und eine Ebene ausgewählt werden. Der berechnete Frame wird in das Ergebnis-Frame eingetragen.

3.5.8.2 Achsspezifische Externe Nullpunktverschiebung

Maschinendaten

Die Aktivierung der Externe Nullpunktverschiebung bzw. Systemframe \$P_EXTFRAME erfolgt über das folgende Maschinendatum:

MD28082 \$MC_MM_SYSTEM_FRAME_MASK, Bit1 = TRUE

Der Betrag für die Externe Nullpunktverschiebung kann manuell über die HMI-Bedienoberfläche und das PLC-Anwenderprogramm über BTSS vorgegeben oder im Teileprogramm über die achsiale Systemvariable \$AA_ETRANS[<Achse>] programmiert werden.

Aktivierung

Die Aktivierung der Externen Nullpunktverschiebung erfolgt über das Nahtstellensignal: DB31, ... DBX3.0 (Externe Nullpunktverschiebung übernehmen)

Verhalten

Bei Aktivierung der Externen Nullpunktverschiebung werden die Verfahrbewegungen aller Achsen, außer Kommando- und PLC-Achsen, sofort gestoppt und der Vorlauf reorganisiert. Die Grobverschiebung des aktuellen Systemframe und des Systemframe in der Datenhaltung wird auf den Wert der achsialen Systemvariablen \$AA_ETRANS[<Achse>] gesetzt. Anschließend wird zuerst die Verschiebung herausgefahren und dann die unterbrochene Bewegung fortgesetzt.

Verhalten bei Kettenmaßangabe

Bei aktiver Kettenmaßangabe G91 und Maschinendatum:

MD42440 \$MC_FRAME_OFFSET_INCR_PROG (Nullpunktverschiebungen in Frames) = 0 wird im Rahmen der Externe Nullpunktverschiebung über Systemframe die Verschiebung, trotz gegenteiliger Projektierung des Maschinendatums, mit dem Anfahrtsatz herausgefahren, obwohl sie durch einen Frame vorgegeben wird.

Hinweis

Die Externe Nullpunktverschiebung wirkt immer absolut.

3.5.8.3 Werkzeugträger

Translationen

Bei Kinematiken vom Typ "P" und "M" wird bei der Anwahl eines Werkzeugträgers ein additiver Frame aktiviert (Tischoffset des orientierbaren Werkzeugträgers), der die Verschiebung des Nullpunktes als Folge der Drehung des Tisches berücksichtigt. Die Nullpunktverschiebung wird in das Systemframe \$P_PARTFR eingetragen. Dabei wird der translatorische Anteil dieses Frames überschrieben. Die anderen Framekomponenten bleiben erhalten.

Das Systemframe \$P_PARTFR muss über folgendes Maschinendatum freigegeben werden:

MD28082 \$MC_MM_SYSTEM_FRAME_MASK, Bit 2 = 1 (Systemframe für TCARR und PAROT)

Hinweis

Alternativ kann die Verschiebung auch über das Maschinendatum zur Aufnahme des Tischoffsets parametrisiert werden:

MD20184 \$MC_TOCARR_BASE_FRAME_NUMBER = <Nummer des Basisframes>

Diese Möglichkeit besteht nur aus Kompatibilitätsgründen zu älteren Softwareständen. Es wird dringend empfohlen diese Vorgehensweise **nicht** mehr zu verwenden.

Eine Frameverschiebung als Folge eines Werkzeugträgerwechsels wird sofort bei der Anwahl mit TCARR=... wirksam. Dagegen wird eine Änderung der Werkzeuglänge nur dann sofort wirksam, wenn ein Werkzeug aktiv ist.

Eine Framedrehung wird mit der Aktivierung nicht ausgeführt, bzw. eine bereits wirksame Drehung wird nicht verändert. Die Stellung der Rundachsen, die für die Berechnung verwendet werden, wird wie im Fall T (nur das Werkzeug ist drehbar) abhängig vom Befehl TCOFR / TCOABS aus dem Drehenteil eines aktiven Frames bzw. aus den Einträgen \$TC_CARR ermittelt. Durch die Aktivierung eines Frames verändert sich die Position im Werkstückkoordinatensystem entsprechend, ohne dass es zu einer Ausgleichsbewegung der Maschine kommt.

Die Verhältnisse sind im folgenden Bild dargestellt:

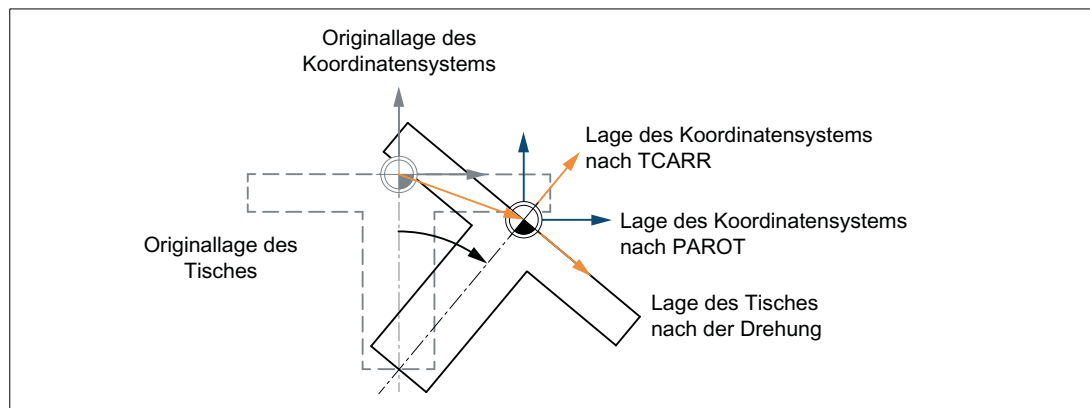


Bild 3-27 Frame bei Aktivierung eines drehbaren Werkzeugträgers mit TCARR

Bei Kinematiken des Typs M (Werkzeug und Tisch sind jeweils um eine Achse drehbar), bewirkt die Aktivierung eines Werkzeugträgers mit TCARR gleichzeitig eine entsprechende Änderung der effektiven Werkzeuglänge (falls ein Werkzeug aktiv ist) und der Nullpunktverschiebung.

Drehungen

Je nach Bearbeitungsaufgabe muss beim Einsatz drehbarer Werkzeugträger bzw. Tische nicht nur eine Nullpunktverschiebung (sei es als Frame oder als Werkzeuglänge), sondern auch eine Drehung berücksichtigt werden. Die Aktivierung eines orientierbaren Werkzeugträgers führt jedoch in keinem Fall unmittelbar zu einer Drehung des Koordinatensystems.

Ist nur das Werkzeug drehbar, kann dafür ein Frame mittels `TOFRAME` bzw. `TOROT` usw. definiert werden.

Bei drehbaren Tischen (Kinematiktypen P und M) hat die Aktivierung mit `TCARR` zunächst ebenfalls keine Drehung des Koordinatensystems zur Folge, d.h. der Nullpunkt des Koordinatensystems verschiebt sich zwar in Bezug auf die Maschine und bleibt fest in Bezug auf den Nullpunkt des Werkstücks, die Orientierung bleibt jedoch unverändert raumfest.

Wird ein Koordinatensystem benötigt, das in Bezug auf das Werkstück fest ist, d.h. gegenüber der Originallage nicht nur verschoben sondern auch entsprechend der Tischdrehung gedreht ist, so kann analog zur Situation bei drehbarem Werkzeug mit `PAROT` eine entsprechende Drehung aktiviert werden.

Mit `PAROT` bleiben Translationen, Skalierungen und Spiegelungen im aktiven Frame erhalten, der Rotationsanteil wird jedoch durch den Rotationsanteil eines orientierbaren Werkzeugträgers, der dem Tisch entspricht gedreht. Dabei bleibt der gesamte programmierbare Frame, einschließlich seines Drehanteils, unverändert.

Der Rotationsanteil, der die Drehung des Werkzeugetisches beschreibt, wird dann entweder in den Systemframe `$PARTFR` oder in den durch MD20184 `$MC_TOCARR_BASE_FRAME_NUMBER` parametrisierten Basisframe eingetragen:

`$MC_MM_SYSTEM_FRAME_MASK`, Bit 2 = <Wert>

Wert	Bedeutung
1	Rotationsanteil → <code>\$PARTFR</code>
0	Rotationsanteil → MD20184 <code>\$MC_TOCARR_BASE_FRAME_NUMBER</code>

Entsprechend dem Hinweis bei der Beschreibung der Tischverschiebung gilt auch hier, dass empfohlen wird, die zweite Alternative für Neuanlagen nicht mehr zu verwenden.

Der Rotationsanteil des Partframes kann mit `PAROTOF` gelöscht werden, unabhängig davon, ob dieser Frame in einem Basis- oder in einem Systemframe steht.

Der Translationsanteil wird gelöscht, indem ein Werkzeugträger aktiviert wird, der keine Verschiebung bewirkt, bzw. durch Abwahl eines eventuell aktiven orientierbaren Werkzeugträgers mit `TCARR=0`.

`PAROT` bzw. `TOROT` berücksichtigen in den Fällen, in denen der Tisch bzw. das Werkzeug mit zwei Drehachsen orientiert werden, die gesamte Orientierungsänderung. Bei gemischten Kinematiken wird jeweils nur der entsprechende, durch eine Drehachse verursachte Anteil berücksichtigt. Damit ist es z. B. bei der Verwendung von `TOROT` möglich, ein Werkstück so zu drehen, dass eine schräge Ebene parallel zur raumfesten X-Y-Ebene liegt, bei der Bearbeitung aber eine Drehung des Werkzeugs zu berücksichtigen, wenn beispielsweise Bohrungen angebracht werden müssen, die nicht senkrecht zu dieser Ebene sind.

Beispiel

Bei einer Maschine zeige die Drehachse des Tisches in die positive Y-Richtung. Der Tisch ist um +45 Grad gedreht. Mit `PAROT` wird dann ein Frame definiert, der ebenfalls eine Drehung von 45 Grad um die Y-Achse beschreibt. Das gegenüber der Außenwelt nicht gedrehte Koordinatensystem (im Bild mit "Lage des Koordinatensystems nach `TCARR`" gekennzeichnet) ist gegenüber dem mitgeführten Koordinatensystem (Lage nach `PAROT`) aber um -45 Grad gedreht. Wird dieses Koordinatensystem z. B. mit `ROT Y-45` definiert und anschließend der Werkzeugträger bei aktivem `TCOFR` angewählt, wird für die Drehachse des Werkzeugträgers ein Winkel von +45 Grad ermittelt.

Der Sprachbefehl `PAROT` wird nicht abgelehnt, wenn kein orientierbarer Werkzeugträger aktiv ist. Allerdings bewirkt ein solcher Aufruf dann keine Frameänderung.

Bearbeitung in Richtung der Werkzeugorientierung

Vor allem an Maschinen mit orientierbarem Werkzeug soll mitunter in Werkzeugrichtung verfahren werden (typischerweise beim Bohren), ohne dass ein Frame aktiviert wird (z. B. mittels `TOFRAME` oder `TOROT`), bei dem eine der Achsen in Richtung des Werkzeugs zeigt. Das Problem stellt sich auch dann, wenn bei einer Schrägbearbeitung ein Frame aktiv ist, der die schräge Ebene definiert, das Werkzeug aber nicht exakt senkrecht eingestellt werden kann, weil wegen eines indexierten Werkzeugträgers (Hirth-Verzahnung) die Werkzeugorientierung nicht beliebig einstellbar ist. In diesen Fällen muss dann - abweichend von der eigentlich verlangten Bewegung senkrecht zur Ebene - in Werkzeugrichtung gebohrt werden, da andernfalls der Bohrer nicht in Richtung seiner Längsachse geführt würde (Werkzeugbruch).

Inkrementelles Verfahren

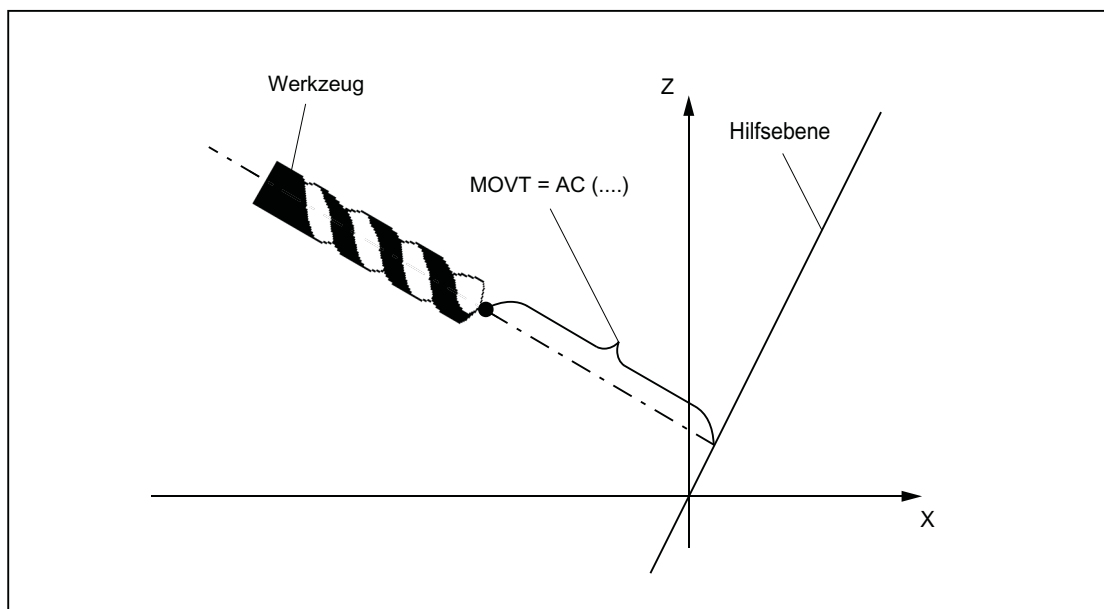
Der Endpunkt für eine inkrementelle Verfahrbewegung in Werkzeugrichtung wird mit `MOVT = <Wert>` oder `MOVT=IC (<Wert>)` programmiert.

Die positive Verfahrrichtung ist dabei von der Werkzeugspitze zur Werkzeugaufnahme definiert. Entsprechend der achsparallelen Bearbeitung z. B. mit `G91 Z...`

Absolutes Verfahren

Der Endpunkt für eine absolute Verfahrbewegung in Werkzeugrichtung wird mit `MOVT=AC (<Wert>)` programmiert.

Dazu wird eine Ebene definiert, die durch den aktuellen Nullpunkt verläuft, und deren Flächennormalenvektor parallel zur Werkzeugorientierung ist. `MOVT` gibt dann die Lage bezüglich dieser Ebene an (siehe Bild). Die Bezugsebene dient nur zur Berechnung der Endposition. Aktive Frames werden durch diese interne Berechnung nicht beeinflusst.



Die Programmierung mit `MOV`T ist unabhängig vom Vorhandensein eines orientierbaren Werkzeugträgers. Die Richtung der Bewegung ist von der aktiven Ebene abhängig. Sie verläuft in Richtung der Applikate, d.h. bei `G17` in Z-Richtung, bei `G18` in Y-Richtung und bei `G19` in X-Richtung. Dies gilt sowohl für den Fall, dass kein orientierbarer Werkzeugträger aktiv ist, als auch für den Fall eines orientierbaren Werkzeugträgers ohne drehbares Werkzeug oder mit drehbarem Werkzeug in Grundstellung.

`MOV`T wirkt bei aktiver Orientierungstransformation (3-4-5-Achs-Transformation) in gleicher Weise.

Wird in einem Satz mit `MOV`T gleichzeitig die Werkzeugorientierung verändert (z. B. bei aktiver 5-Achstransformation durch gleichzeitige Interpolation der Rundachsen), ist die Orientierung am Satzanfang für die Bewegungsrichtung von `MOV`T maßgebend.

Bei aktiver 5-Achstransformation wird die Bahn des "tool center points" (TCP) durch die Orientierungsänderung nicht beeinflusst, d. h. die Bahn bleibt eine Gerade, deren Richtung durch die Werkzeugorientierung im Satzangang bestimmt ist.

Wird `MOV`T programmiert, muss Linear- oder Spline-Interpolation (`G0`, `G1`, `ASPLINE`, `BSPLINE`, `CSPLINE`) aktiv sein. Andernfalls wird ein Alarm ausgegeben.

Ist eine Spline-Interpolationsart aktiv, ist die resultierende Bahn im allgemeinen keine Gerade, da der von `MOV`T ermittelte Endpunkt so behandelt wird, als sei er explizit mit X, Y, Z programmiert worden.

In einem Satz mit `MOV`T dürfen keine Geometrieachsen programmiert werden.

Definition von Framedrehungen mit Raumwinkeln

Soll ein Frame, der eine Drehung um mehr als eine Achse beschreibt, definiert werden, so geschieht das durch die Verkettung von Einzeldrehungen. Dabei erfolgt die nachfolgende Drehung im neuen gedrehten Koordinatensystem.

Das gilt sowohl bei Programmierung in einem Satz als auch beim Aufbau eines Frames in mehreren aufeinander folgenden Sätzen:

- Ein Satz: `N10 ROT X... Y... Z...`
- Mehrere aufeinander folgenden Sätze:


```
N10 ROT Y...
N20 AROT X...
N30 AROT Z...
```

Raumwinkel

In Werkstückzeichnungen werden zur Beschreibung schräger Flächen oftmals Raumwinkel angegeben. Raumwinkel sind dabei die Winkel welche die Schnittgeraden der schrägen Ebene mit den Hauptebenen (X-Y, Y-Z, Z-X-Ebene) des Werkstückkoordinatensystems bilden (siehe nachfolgendes Bild). Die Orientierung einer Ebene im Raum ist durch die Angabe zweier Raumwinkel eindeutig bestimmt. Der dritte Raumwinkel ergibt sich aus den beiden ersten.

Mit den Befehlen `ROTS`, `AROTS` und `CROTS` können die Drehungen unmittelbar als Raumwinkel beschrieben werden.

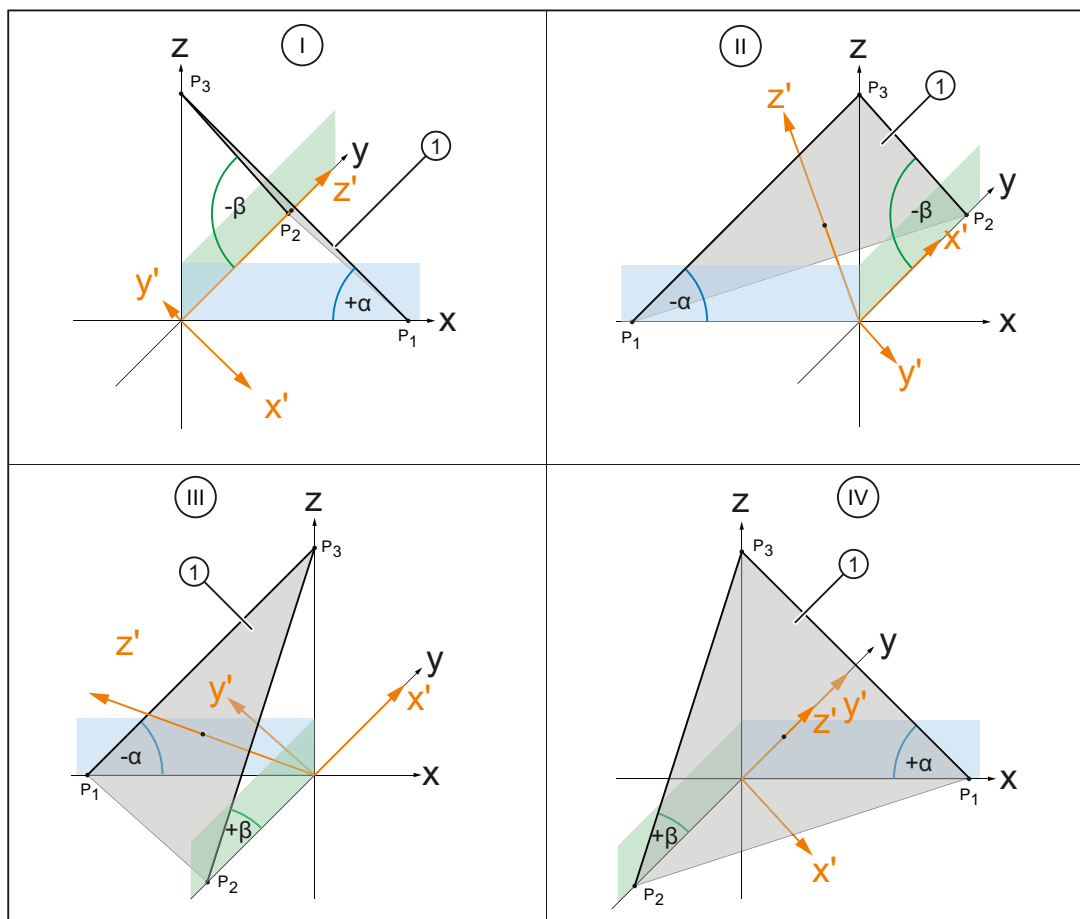
Die Angabe eines einzelnen Raumwinkels ist zulässig. Die Drehung, die in diesem Fall bei `ROTS` bzw. `AROTS` ausgeführt wird, ist identisch zu denen bei `ROT` bzw. `AROT`.

Die beiden im Befehl programmierten Achsen definieren eine Ebene. Die nicht programmierte Achse definiert die zugehörige dritte Achse des kartesischen Koordinatensystems. Damit ist für

3.5 Frames

die beiden programmierten Achsen auch eindeutig definiert, welches die erste und welches die zweite Achse ist. Die Definition entspricht dabei den Festlegungen der Ebenendefinition bei G17/G18/G19.

Der mit dem Achsbuchstaben einer Achse der Ebene programmierte Winkel gibt dann die Achse an, um die man die andere Achse der Ebene drehen muss, um diese in die Schnittgerade zu überführen, die die gedrehte Ebene mit der aus der anderen und der dritten Achse aufgespannten Ebene bildet. Mit dieser Definition ist gewährleistet, dass für den Fall, dass einer der beiden programmierten Winkel gegen Null geht, die derart definierte Ebene in die Ebene übergeht, die sich ergibt, wenn nur eine Achse (z. B. auch mit ROT oder AROT) programmiert ist.



I, ..., Quadrant 1 bis 4

IV

① Schräge Ebene als Vorgabe für die neue G17-Ebene

α, β Raumwinkel der schrägen Ebene

Bild 3-28 Drehung um Raumwinkel

Im Bild sind die Raumwinkel für eine beispielhafte Ebene in den Quadranten I bis IV aufgezeigt. Die schräge Ebene definiert die Ausrichtung der G17-Ebene nach der Drehung des Werkstück-

Koordinatensystems WKS. Die Vorzeichen der Raumwinkel geben die Richtung an, um die das Koordinatensystem um die jeweilige Achse gedreht wird:

1. Drehung um y:
Drehung des Werkstück-Koordinatensystems WKS um die y-Achse um den vorzeichenbehafteten Winkel $\alpha \Rightarrow$
x'-Achse ist parallel (kollinear) zur Schnittgeraden der xz-Ebene mit der schrägen Ebene ausgerichtet
2. Drehung um x':
Drehung des neuen Werkstück-Koordinatensystems WKS' um x'-Achse um den vorzeichenbehafteten Winkel $\beta \Rightarrow$
 - y'-Achse ist parallel (kollinear) zur Schnittgeraden der zy-Ebene mit der schrägen Ebene ausgerichtet
 - z'-Achse steht senkrecht auf der schrägen Ebene
 - G17' liegt parallel zur schrägen Ebene

Die entsprechende Programmierung zur Ausrichtung der G17-Ebene des Werkstück-Koordinatensystems WKS auf die schräge Ebene lautet pro Quadrant:

- Quadrant I: ROTs X<+ α > Y<- β >
- Quadrant II: ROTs X<- α > Y<- β >
- Quadrant III: ROTs X<- α > Y<+ β >
- Quadrant IV: ROTs X<+ α > Y<+ β >

Orientierung

Mit der Angabe der Raumwinkel ist die Orientierung des zweidimensionalen Koordinatensystems innerhalb der Ebene (d.h. der Drehwinkel um den Flächennormalenvektor) nicht definiert. Die Lage des Koordinatensystems wird deshalb so festgelegt, dass die gedrehte erste Achse in der Ebene liegt, die von der ersten und dritten Achse des nicht gedrehten Koordinatensystems aufgespannt wird.

Das bedeutet:

- Bei Programmierung von X und Y liegt die neue X-Achse in der ursprünglichen Z-X-Ebene.
- Bei Programmierung von Z und X liegt die neue Z-Achse in der ursprünglichen Y-Z-Ebene.
- Bei Programmierung von Y und Z liegt die neue Y-Achse in der ursprünglichen X-Y-Ebene.

Wird eine von dieser Voreinstellung abweichende Lage des Koordinatensystems benötigt, muss eine zusätzliche Drehung mit `AROT . . .` ausgeführt werden.

Parametrierung: zy'x"- (RPY-Winkel) oder zx'z"-Konvention

Die programmierten Raumwinkel werden bei der Eingabe abhängig von folgendem Maschinendatum in die äquivalenten Euler-Winkel nach zy'x"-Konvention (RPY-Winkel) oder zx'z"-Konvention umgerechnet:

MD10600 \$MN_FRAME_ANGLE_INPUT_MODE

Framedrehung in Werkzeugrichtung

Mit dem bereits in älteren Softwareständen vorhandenen Sprachbefehl `TOFRAME` besteht die Möglichkeit, einen Frame zu definieren, dessen Z-Achse in Werkzeugrichtung zeigt. Ein vorhandener programmierter Frame wird dabei durch einen Frame überschrieben, der eine reine Drehung beschreibt. Im vorher aktiven Frame eventuell vorhandene Nullpunktverschiebungen, Spiegelungen oder Skalierungen werden gelöscht. Mitunter ist dieses Verhalten störend. Insbesondere ist es oft sinnvoll, eine Nullpunktverschiebung, mit der der Bezugspunkt im Werkstück definiert wird, beizubehalten.

Es wird deshalb zusätzlich der Sprachbefehl `TOROT` eingeführt, der im programmierten Frame nur den Rotationsanteil überschreibt, die übrigen Komponenten aber unverändert lässt. Die durch `TOROT` definierte Drehung ist die gleiche wie bei `TOFRAME`.

`TOROT` ist ebenso wie `TOFRAME` unabhängig vom Vorhandensein eines orientierbaren Werkzeugträgers. Insbesondere ist dieser Sprachbefehl auch bei 5-Achs-Transformationen verwendbar.

Mit dem neuen Sprachbefehl `TOROT` wird außerdem eine konsistente Programmierung bei aktiven orientierbaren Werkzeugträgern für jeden Kinematiktyp erreicht.

Mit `TOFRAME` bzw. `TOROT` werden Frames definiert, deren Z-Richtung in Werkzeugrichtung zeigt. Diese Definition passt zu Fräsbearbeitungen, bei denen typischerweise `G17` aktiv ist. Insbesondere bei Drehbearbeitungen oder allgemein bei aktivem `G18` oder `G19` ist es jedoch wünschenswert, dass Frames definiert werden können, bei denen die Ausrichtung an der X- bzw. Y-Achse erfolgt. Dazu sind in der G-Gruppe 53 folgende Befehle vorhanden:

- `TOFRAMEX`, `TOFRAMEY`, `TOFRAMEZ`
- `TOROTX`, `TOROTY`, `TOROTZ`

Mit diesen Befehlen ist es möglich, entsprechende Frames zu definieren. Dabei ist die Funktionalität von `TOFRAME` und `TOFRAMEZ` bzw. von `TOROT` und `TOROTZ` jeweils identisch.

Die durch `TOROT` bzw. `TOFRAME` entstehenden Frames können in einen eigenen Systemframe (`$P_TOOLFR`) geschrieben werden. Der programmierbare Frame bleibt dann unverändert erhalten.

- Voraussetzung: MD28082 `$MC_MM_SYSTEM_FRAME_MASK`, Bit 3 = 1

Bei der Programmierung von `TOROT` bzw. `TOFRAME` usw. ist das Verhalten mit bzw. ohne Systemframe identisch. Unterschiede ergeben sich dann, wenn der programmierbare Frame weiter bearbeitet wird.

Hinweis

Es wird empfohlen, für die von den Befehlen der G-Gruppe 53 erzeugten Frames bei Neuanlagen nur noch den dafür vorgesehenen Systemframe zu verwenden.

Beispiel

Nach `TOROT` wird `TRANS` programmiert. `TRANS` ohne Angabe von Parametern löscht den programmierbaren Frame. In der Variante ohne Systemframe wird damit auch der durch `TOROT` verursachte Frameanteil des programmierbaren Frames gelöscht, steht der `TOROT`-Anteil im Systemframe, bleibt er erhalten.

`TOROT` bzw. `TOFRAME` usw. werden mit dem Sprachbefehl `TOROTOF` ausgeschaltet. `TOROTOF` löscht den gesamten Systemframe `$P_TOOLFR`. Beschreiben die Befehle `TOFRAME` usw.

nicht das Systemframe, sondern das programmierbare Frame (alte Variante), löscht `TOROT` nur den Rotationsanteil und lässt die übrigen Frameanteile unverändert.

Ist vor der Aktivierung der Sprachbefehle `TOFRAME` oder `TOROT` bereits ein drehender Frame aktiv, besteht oft die Forderung, dass der neu definierte Frame vom alten Frame möglichst wenig abweicht. Das ist z.B. dann der Fall, wenn eine Framedefinition leicht modifiziert werden muss weil die Werkzeugorientierung wegen Hirth-verzahnter Rundachsen nicht beliebig eingestellt werden kann. Durch die genannten Sprachbefehle wird die Z-Richtung des neuen Frames eindeutig definiert.

Parametrierung: Frame-Definition bei `TOFRAME`, `TOROT` und `PAROT` (SD42980)

Mit dem folgenden Settingdatum wird die Richtung der Geometrieachsen der aktuellen Bearbeitungsebene (G17: X- und Y-Achse) bei der Frame-Definition mittels `TOFRAME`, `TOROT` (`TOROTY`, `TOROTX`) oder `PAROT` festgelegt.

Bei einer Frameberechnung wird die Werkzeugrichtung so festgelegt, dass Werkzeugrichtung und Applikate (G17: Z-Achse) des Frames parallel sind und senkrecht auf der Bearbeitungsebene stehen.

Die Drehung um die Werkzeugachse ist zunächst beliebig. Mit dem Settingdatum kann diese freie Drehung so bestimmt werden, dass der neu definierte Frame von einem vorher aktiven Frame möglichst wenig abweicht.

In allen Fällen, in denen das Settingdatum ungleich Null ist, bleibt ein aktiver Frame unverändert, wenn die Werkzeugrichtung des alten und des neuen Frame übereinstimmen.

SD42980 \$SC_TOFRAME_MODE

TCARR (Werkzeugträger anfordern) und PAROT (Werkstückkoordinatensystem am Werkstück ausrichten)

`TCARR` nutzt das durch das folgende Maschinendatum bezeichnete Basisframe:
MD20184 \$MC_TOCARR_BASE_FRAME_NUMBER.

Um Konflikte mit Systemen zu vermeiden, die bereits alle Basisframes nutzen, kann für `TCARR` und `PAROT` ein eigenes Systemframe angelegt werden.

`PAROT`, `TOROT` und `TOFRAME` verändern bisher den Rotationsanteil des programmierbaren Frames. Ein separates Abschalten von `PAROT` oder `TOROT` ist in diesem Falle nicht möglich. Bei `RESET` wird das programmierbare Frame gelöscht, was bedeutet, dass nach Betriebsartenwechsel nach `JOG` der Rotationsanteil von `PAROT` und `TOROT` nicht mehr vorhanden ist. Auch muss das programmierbare Frame dem Anwender uneingeschränkt zur Verfügung stehen. Die durch `PAROT` und `TOROT` erzeugten Frames müssen über die Datensicherung archivierbar und wieder ladbar sein.

Das Systemframe für `TCARR` und `PAROT` wird projiziert mit:
MD28082 \$MC_MM_SYSTEM_FRAME_MASK, Bit 2 = 1

Folgendes Maschinendatum wird dann nicht mehr ausgewertet:
MD20184 \$MC_TOCARR_BASE_FRAME_NUMBER

Ist das Systemframe für `TCARR` projiziert, so beschreibt `TCARR` und `PAROT` das entsprechende Systemframe, ansonsten wird das durch Maschinendatum MD20184 bezeichnete Basisframe beschrieben.

TCARR trägt bei Kinematiken des Typs P und des Typs M den Tischoffset des orientierbaren Werkzeugträgers (Verschiebung des Nullpunktes als Folge der Drehung des Tisches), als Translation in das Systemframe ein. PAROT rechnet das Systemframe so um, dass sich ein werkstückbezogenes Werkstückkoordinatensystem ergibt.

Die Systemframes werden remanent gespeichert und bleiben deshalb nach Reset erhalten. Auch bei Betriebsartenwechsel bleiben die Systemframes aktiv.

Für die Anzeige werden die Befehle PAROT und TOROT, TOFRAME jeweils einer eigenen G-Gruppe zugeordnet.

PAROTOF

PAROTOF ist der Ausschaltbefehl zu PAROT. Dieser Befehl löscht die Drehungen im Systemframe für PAROT ab. Es werden dabei die Drehungen im aktuellen \$P_PARTFRAME und im Datenhaltungsframe \$P_PARTFR gelöscht. Damit wird die Lage des Koordinatensystems nach TCARR wieder hergestellt. PAROTOF ist in der gleichen G-Gruppe, wie PAROT und erscheint deshalb in der G-Befehlsanzeige.

TOROT (Z-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten) und TOFRAME (dto.)

Das Systemframe für TOROT und TOFRAME wird über folgendes Maschinendatum aktiviert: MD28082 \$MC_MM_SYSTEM_FRAME_MASK, Bit 3 = 1

Dieses Systemframe liegt in der Framekette vor dem programmierbaren Frame. Das ENS-Koordinatensystem liegt entsprechend vor dem programmierbaren Frame.

TOROTOF

TOROTOF ist der Ausschaltbefehl zu TOROT und TOFRAME. Dieser Befehl löscht den entsprechenden Systemframe. Dabei wird der aktuelle \$P_TOOLFRAME und der Datenhaltungsframe \$P_TOOLFR gelöscht. TOROTOF ist in der gleichen G-Gruppe wie TOROT und TOFRAME und erscheint deshalb in der G-Befehlsanzeige.

Beispiel

Verwendung eines orientierbaren Werkzeugträgers mit aufgelöster Kinematik.

```
N10 $TC_DP1[1,1]= 120
N20 $TC_DP3[1,1]= 13           ; Werkzeuglänge 13mm
; Definition des Werkzeugträgers 1:
N30 $TC_CARR1[1] = 0           ; X-Komponente des 1. Offsetvektors
N40 $TC_CARR2[1] = 0           ; Y-Komponente des 1. Offsetvektors
N50 $TC_CARR3[1] = 0           ; Z-Komponente des 1. Offsetvektors
N60 $TC_CARR4[1] = 0           ; X-Komponente des 2. Offsetvektors
N70 $TC_CARR5[1] = 0.          ; Y-Komponente des 2. Offsetvektors
N80 $TC_CARR6[1] = -15.        ; Z-Komponente des 2. Offsetvektors
N90 $TC_CARR7[1] = 1           ; X-Komponente der 1. Achse
N100 $TC_CARR8[1] = 0          ; Y-Komponente der 1. Achse
```

```

N110 $TC_CARR9[1] = 0 ; Z-Komponente der 1. Achse
N120 $TC_CARR10[1] = 0 ; X-Komponente der 2. Achse
N130 $TC_CARR11[1] = 1 ; Y-Komponente der 2. Achse
N140 $TC_CARR12[1] = 0 ; Z-Komponente der 2. Achse
N150 $TC_CARR13[1] = 30. ; Drehwinkel der 1. Achse
N160 $TC_CARR14[1] = -30. ; Drehwinkel der 2. Achse
N170 $TC_CARR15[1] = 0 ; X-Komponente des 3. Offsetvektors
N180 $TC_CARR16[1] = 0 ; Y-Komponente des 3. Offsetvektors
N190 $TC_CARR17[1] = 0 ; Z-Komponente des 3. Offsetvektors
N200 $TC_CARR18[1] = 0 ; X-Komponente des 4. Offsetvektors
N210 $TC_CARR19[1] = 0 ; Y-Komponente des 4. Offsetvektors
N220 $TC_CARR20[1] = 15. ; Z-Komponente des 4. Offsetvektors
N230 $TC_CARR21[1] = A ; Bezug fuer 1. Achse
N240 $TC_CARR22[1] = B ; Bezug fuer 2. Achse
N250 $TC_CARR23[1] = "M" ; Typ des Werkzeugtraegers
N260 X0 Y0 Z0 A0 B45 F2000
N270 TCARR=1 X0 Y10 Z0 T1 TCOABS ; Anwahl des orient. Werkzeugträgers
N280 PAROT ; Drehung des Tisches
N290 TOROT ; Drehung der z-Achse in WZ-Orient.
N290 X0 Y0 Z0
N300 G18 MOVT=AC(20) ; Bearbeitung in G18 Ebene
N310 G17 X10 Y0 Z0 ; Bearbeitung in G17 Ebene
N320 MOVT=-10
N330 PAROTOF ; Drehung des Tisches abschalten
N340 TOROTOF ; WKS nicht mehr am WZ ausrichten.
N400 M30

```

3.5.9 Unterprogramme mit Attribut SAVE

Bei verschiedenen Frames ist das Verhalten bezüglich Unterprogrammen mit dem Attribut SAVE einstellbar.

Einstellbare Frames G54 - G599

Mit MD10617 \$MN_FRAME_SAVE_MASK.BIT0 kann das Verhalten der einstellbaren Frames eingestellt werden:

- BIT0 = 0
Werden durch das Unterprogramm nur die Werte des aktiven einstellbaren Frames über die Systemvariable \$P_IFRAME verändert, der G-Befehl aber beibehalten, bleibt die Veränderung auch nach Unterprogrammende erhalten.
- BIT0 = 1
Mit dem Unterprogrammende wird der vor dem Unterprogrammaufruf aktive einstellbare Frame, G-Befehl und Werte, reaktiviert.

Basisframes \$P_CHBFR[] und \$P_NCBFR[]

Mit MD10617 \$MN_FRAME_SAVE_MASK.BIT1 kann das Verhalten der Basisframes eingestellt werden:

- BIT1 = 0
Wird durch das Unterprogramm der aktive Basisframe verändert, bleibt die Veränderung auch nach Unterprogrammende erhalten.
- BIT1 = 1
Mit dem Unterprogrammende wird der vor dem Unterprogrammaufruf aktive Basisframe reaktiviert.

Programmierbarer Frame

Mit dem Unterprogrammende wird der vor dem Unterprogrammaufruf aktive programmierbare Frame reaktiviert.

Systemframes

Werden durch das Unterprogramm Systemframes verändert, bleibt die Veränderung nach Unterprogrammende erhalten.

3.5.10 Datensicherung

Datenbausteine

Die Datensicherung der Frames erfolgt in folgende Datenbausteine:

- Kanalspezifische Frames ⇒ Datenbausteine _N_CHAN<x>_UFR
- Globale Frames ⇒ Datenbaustein _N_NC_UFR

Hinweis

- Es wird dringen empfohlen, zwischen dem Sichern und Wiedereinspielen der gesicherten Systemframes das folgende Maschinendatum nicht zu ändern. Ansonsten können die gesicherte Systemframes eventuell nicht mehr geladen werden.
MD28082 \$MC_MM_SYSTEM_FRAME_MASK (Projektierung von kanalspezifischen Systemframes, die im Kanal eingerechnet werden)
 - Die Datensicherung erfolgt immer nach den aktuell im Kanal gültigen Geometrieachs-Zuordnungen und nicht nach den in den Maschinendaten ursprünglich eingestellten Achskonstellationen.
-

Datensicherung von Systemframes

Bei der Datensicherung von Systemframes, werden nur die in der Datenhaltung angelegten Systemframes berücksichtigt. Diese Systemframes wurden bei der Inbetriebnahme der Steuerung über das folgende Maschinendatum ausgewählt:

MD28083 \$MC_MM_SYSTEM_DATAFRAME_MASK

Nicht in der Datenhaltung angelegte Frames werden nicht gesichert.

Datensicherung von NC-globalen Frames

Eine Datensicherung von NC-globalen Frames erfolgt nur, wenn in einem der folgenden Maschinendaten mindestens ein NC-globaler Frame parametrier ist:

- MD18601 \$MN_MM_NUM_GLOBAL_USER_FRAMES
- MD18602 \$MN_MM_NUM_GLOBAL_BASE_FRAMES
- MD18603 \$MN_MM_NUM_GLOBAL_G_FRAMES

3.5.11 Positionen in den Koordinatensystemen

Die aktuellen Sollwert-Positionen in den Koordinatensystemen können über folgende Systemvariablen gelesen werden. Über PLC können wahlweise die Istwerte im WKS, ENS, BNS oder MKS angezeigt werden. Es gibt dazu einen Softkey Istwertanzeige im MKS/WKS. Der Maschinenhersteller kann von PLC-Seite vorgeben, welches Koordinatensystem bei seiner Maschine dem Werkstückkoordinatensystem entspricht. Der HMI fordert die entsprechenden Istwerte von NC an.

\$AA_IM[Achse]

Die Sollwerte im Maschinenkoordinatensystem können für jede Achse mit der Variablen \$AA_IM[Achse] gelesen werden.

\$AA_IEN[Achse]

Die Sollwerte im einstellbaren Nullpunkt-Koordinatensystem ENS können für jede Achse mit der Variablen \$AA_IEN[Achse] gelesen werden.

\$AA_IBN[Achse]

Mit \$AA_IBN[Achse] können die Sollwerte im Basisnullpunkt-Koordinatensystem BNS gelesen werden.

\$AA_IW[Achse]

Mit \$AA_IW[Achse] können die Sollwerte im Werkstück-Koordinatensystem WKS gelesen werden.

3.5.12 Steuerungsverhalten

3.5.12.1 POWER ON

Frame	Zustand nach POWER ON
Programmierbarer Frame \$P_PFRAME	Gelöscht.
Einstellbare Frame \$P_IFRAME	Bleibt erhalten, abhängig von: <ul style="list-style-type: none"> • MD24080 \$MC_USER_FRAME_POWERON_MASK, Bit 0 • MD20152 \$MC_GCODE_RESET_MODE[7]

3.5 Frames

Frame	Zustand nach POWER ON
Schleifframe \$P_GFRAME	Bleibt erhalten, abhängig von: <ul style="list-style-type: none"> • MD24080 \$MC_USER_FRAME_POWERON_MASK, Bit 0 • MD20152 \$MC_GCODE_RESET_MODE[63]
Gesamt-Basisframe \$P_ACTBF-RAME	Bleibt erhalten, abhängig von: <ul style="list-style-type: none"> • MD20110 \$MC_RESET_MODE_MASK Bit 0 und Bit 14 Über Maschinendaten können Basisframes gelöscht werden: <ul style="list-style-type: none"> • MD10615 \$MN_NCBFRAME_POWERON_MASK • MD24004 \$MC_CHBFRAME_POWERON_MASK
Systemframes: \$P_PARTFRAME, \$P_SETFR- AME, \$P_ISO1FRAME, \$P_ISO2FRAME, \$P_ISO3FRAME, \$P_TOOLFR- AME, \$P_WPFRAME, \$P_TRAFR- AME, \$P_ISO4FRAME, \$P_REL- FRAME, \$P_CACFRAME	Bleiben erhalten. Über das Maschinendatum können einzelne Systemframes ge- löscht werden: <ul style="list-style-type: none"> • MD24008 \$MC_CHSFRAME_POWERON_MASK Das Löschen von Systemframes wird erstrangig in der Datenhal- tung durchgeführt.
Externe Nullpunktverschiebung \$P_EXTFRAME	Bleibt erhalten, muss aber erneut aktiviert werden.
DRF-Verschiebung	Gelöscht.

3.5.12.2 Betriebsartenwechsel

Systemframes

Die Systemframes bleiben erhalten und aktiv.

Betriebsart JOG

In JOG werden nur die Framekomponenten des aktuellen Frames für die Geometrieachsen berücksichtigt, wenn eine Rotation aktiv ist. Alle anderen achsspezifischen Frames werden nicht berücksichtigt.

PLC- und Kommando-Achsen

Für PLC- und Kommando-Achsen ist das Verhalten einstellbar über das Maschinendatum:

MD32074 \$MA_FRAME_OR_CORRPOS_NOTALLOWED (Frame oder HL-Korrektur sind unzulässig)

3.5.12.3 Kanal-Reset / Teileprogrammende

Reset-Verhalten der Basisframes

Das Reset-Verhalten der Basisframes wird eingestellt über das Maschinendatum:

MD20110 \$MC_RESET_MODE_MASK (Festlegung der Steuerungs-Grundstellung nach Kanal-Reset / Teileprogrammende)

Reset-Verhalten der Systemframes

Die Systemframes bleiben auch nach Kanal-Reset / Teileprogrammende in der Datenhaltung erhalten.

Die Aktivierung der einzelnen Systemframes kann über die folgenden Maschinendaten projiziert werden:

MD24006 \$MC_CHSFRAME_RESET_MASK, Bit<n> = <Wert> (Aktive Systemframes nach Kanal-Reset / Teileprogrammende)

Bit	Wert	Bedeutung
0	1	Systemframe für Istwertsetzen und Ankratzen ist nach Kanal-Reset / Teileprogrammende aktiv.
1	1	Systemframe für externe Nullpunktverschiebung ist nach Kanal-Reset / Teileprogrammende aktiv.
2	---	Wird nicht ausgewertet.
3	---	Wird nicht ausgewertet.
4	1	Systemframe für Werkstückbezugspunkte ist nach Kanal-Reset / Teileprogrammende aktiv.
5	1	Systemframe für Zyklen ist nach Kanal-Reset / Teileprogrammende aktiv.
6	---	Reserviert, RESET-Verhalten abhängig von MD20110 \$MC_RESET_MODE_MASK.
7	1	Systemframe \$P_ISO1FR ist nach Kanal-Reset / Teileprogrammende aktiv.
8	1	Systemframe \$P_ISO2FR ist nach Kanal-Reset / Teileprogrammende aktiv.
9	1	Systemframe \$P_ISO3FR ist nach Kanal-Reset / Teileprogrammende aktiv.
10	1	Systemframe \$P_ISO4FR ist nach Kanal-Reset / Teileprogrammende aktiv.
11	1	Systemframe \$P_RELFR ist nach Kanal-Reset / Teileprogrammende aktiv.
Bei Bit<n> = 0 wird der entsprechende Systemframe nicht aktiv.		

Reset-Verhalten der Systemframes für TCARR, PAROT, TOROT und TOFRAME

Das Reset-Verhalten der Systemframes für TCARR, PAROT, TOROT und TOFRAME nach Kanal-Reset / Teileprogrammende ist von der Reset-Einstellung der G-Befehle abhängig.

Die Einstellung erfolgt mit den Maschinendaten:

- MD20110 \$MC_RESET_MODE_MASK, Bit<n> = <Wert>

Bit	Wert	Bedeutung	
0	0	Der aktuelle Systemframe für TCARR und PAROT bleibt erhalten.	
0	1	Weitere relevante Maschineneinstellungen	Auswirkung
		MD20152 \$MC_GCODE_RESET_MODE[51] = 0 UND MD20150 \$MC_GCODE_RESET_VALUES[51] = 1	PAROTOF
		MD20152 \$MC_GCODE_RESET_MODE[51] = 0 UND MD20150 \$MC_GCODE_RESET_VALUES[51] = 2	PAROT
		MD20152 \$MC_GCODE_RESET_MODE[51] = 1	TCARR und PAROT bleiben erhalten
		MD20152 \$MC_GCODE_RESET_MODE[52] = 0 UND MD20150 \$MC_GCODE_RESET_VALUES[52] = 1	TOROTOF
		MD20152 \$MC_GCODE_RESET_MODE[52] = 0 UND MD20150 \$MC_GCODE_RESET_VALUES[52] = 2	TOROT
		MD20152 \$MC_GCODE_RESET_MODE[52] = 0 UND MD20150 \$MC_GCODE_RESET_VALUES[52] = 3	TOFRAME
		MD20152 \$MC_GCODE_RESET_MODE[52] = 1	TOROT und TOFRAME bleiben erhalten
Beschreibung der weiteren Bits bei Bit 0 == 1			
14	0	Verkettetes Gesamtbasisframe wird gelöscht.	
	1	Das Gesamtbasisframe ergibt sich anhand folgender weiterer Maschinendaten:: MD24002 \$MC_CHBFRAME_RESET_MASK, Bit<n> = 1 n: der n-te kanalspezifischer Basisframe wird in das verkettete Gesamtbasisframe eingerechnet.	
		MD10613 \$MN_NCBFRAME_RESET_MASK, Bit<n> = 1 n: der n-te NCU-globaler Basisframe wird in das verkettete Gesamtbasisframe eingerechnet.	

TCARR und PAROT sind zwei eigenständige Funktionen, die das gleiche Frame beschreiben. Bei Kanal-Reset / Teileprogrammende wird mit PAROTOF auch der Anteil von TCARR nicht aktiviert.

- MD20152 \$MC_GCODE_RESET_MODE[] (RESET-Verhalten der G-Gruppen)
- MD20150 \$MC_GCODE_RESET_VALUES[] (Löschstellung der G-Gruppen)

Framezustände nach Kanal-Reset / Teileprogrammende

Frame	Zustand nach Kanal-Reset / Teileprogrammende
Programmierbarer Frame: \$P_PFRAME	Gelöscht.
Einstellbare Frame \$P_IFRAME	Bleibt erhalten, abhängig von: <ul style="list-style-type: none"> • MD20110 \$MC_RESET_MODE_MASK • MD20152 \$MC_GCODE_RESET_MODE[7]
Schleifframe \$P_GFRAME	Bleibt erhalten, abhängig von: <ul style="list-style-type: none"> • MD20110 \$MC_RESET_MODE_MASK • MD20152 \$MC_GCODE_RESET_MODE[63]
Gesamt-Basisframe \$P_ACTBFRAME	Bleibt erhalten, abhängig von: <ul style="list-style-type: none"> • MD20110 \$MC_RESET_MODE_MASK Bit 0 und Bit 14 • MD10613 \$MN_NCBFRAME_RESET_MASK • MD24002 \$MC_CHBFRAME_RESET_MASK
Systemframes: \$P_PARTFRAME, \$P_SETFRAME, \$P_ISO1FRAME, \$P_ISO2FRAME, \$P_ISO3FRAME, \$P_TOOLFRAME, \$P_WPFRAME, \$P_TRAFRAME, \$P_ISO4FRAME, \$P_RELFRAME, \$P_CACFRAME	Bleiben erhalten, abhängig von: <ul style="list-style-type: none"> • MD24006 \$MC_CHSFRAME_RESET_MASK • MD20150 \$MC_GCODE_RESET_VALUES[<n>]
Externe Nullpunktverschiebung \$P_EXTFRAME	Bleibt erhalten.
DRF-Verschiebung	Bleibt erhalten.

Löschen von Systemframes

Die Systemframes in der Datenhaltung können bei Kanal-Reset / Teileprogrammende über folgendes Maschinendatum gelöscht werden:

MD24007 \$MC_CHSFRAME_RESET_CLEAR_MASK, Bit<n> = <Wert>

Bit	Wert	Bedeutung
0	1	Systemframe für Istwertsetzen und Ankratzen wird bei Kanal-Reset / Teileprogrammende gelöscht.
1	1	Systemframe für externe Nullpunktverschiebung wird bei Kanal-Reset / Teileprogrammende gelöscht.
2	---	Reserviert, TCARR und PAROT siehe MD20150 \$MC_GCODE_RESET_VALUES[].
3	---	Reserviert, TOROT und TOFRAME siehe MD20150 \$MC_GCODE_RESET_VALUES[].
4	1	Systemframe für Werkstückbezugspunkte wird bei Kanal-Reset / Teileprogrammende gelöscht.
5	1	Systemframe für Zyklen wird bei Kanal-Reset / Teileprogrammende gelöscht.
6	---	Reserviert, Reset-Verhalten abhängig von MD20110 \$MC_RESET_MODE_MASK.
7	1	Systemframe \$P_ISO1FR wird bei Kanal-Reset / Teileprogrammende gelöscht.

3.5 Frames

Bit	Wert	Bedeutung
8	1	Systemframe \$P_ISO2FR wird bei Kanal-Reset / Teileprogrammende gelöscht.
9	1	Systemframe \$P_ISO3FR wird bei Kanal-Reset / Teileprogrammende gelöscht.
10	1	Systemframe \$P_ISO4FR wird bei Kanal-Reset / Teileprogrammende gelöscht.
11	1	Systemframe \$P_RELFR wird bei Kanal-Reset / Teileprogrammende gelöscht.
Bei Bit<n> = 0 wird der entsprechende Systemframe nicht gelöscht.		

3.5.12.4 Teileprogrammstart

Framezustände nach Teileprogrammstart

Frame	Zustand nach Teileprogrammstart
Programmierbarer Frame \$P_PFRAME	Gelöscht.
Einstellbare Frame \$P_IFFRAME	Bleibt erhalten, abhängig von: MD20112 \$MC_START_MODE_MASK
Schleifframe \$P_GFRAME	Bleibt erhalten, abhängig von: MD20112 \$MC_START_MODE_MASK
Gesamt-Basisframe \$P_ACTBFRAME	Bleibt erhalten.
Systemframes: \$P_PARTFRAME, \$P_SETFRAME, \$P_ISO1FRAME, \$P_ISO2FRAME, \$P_ISO3FRAME, \$P_TOOLFRAME, \$P_WPFRAME, \$P_TRAFRAME, \$P_ISO4FRAME, \$P_RELFRAME, \$P_CACFRAME	Bleiben erhalten.
Externe Nullpunktverschiebung \$P_EXTFRAME	Bleibt erhalten.
DRF-Verschiebung	Bleibt erhalten.

3.5.12.5 Satzsuchlauf

Satzsuchlauf mit Berechnung

Beim Satzsuchlauf mit Berechnung werden auch Datenhaltungsframes modifiziert.

Abbruch des Satzsuchlaufs

Wird ein Satzsuchlauf mit Kanal-Reset abgebrochen, so kann mit dem folgenden Maschinendatum eingestellt werden, dass alle Datenhaltungsframes auf den Wert vor dem Satzsuchlauf zurückgesetzt werden:

MD28560 \$MC_MM_SEARCH_RUN_RESTORE_MODE, Bit<n>

Bit	Wert	Bedeutung
0	1	Alle Frames in der Datenhaltung werden restauriert.

Bei kaskadiertem Satzsuchlauf, werden die Frames auf den Zustand des Vorgänger-Satzsuchlaufes gesetzt.

SERUPRO

Die Funktion "SERUPRO" wird nicht unterstützt.

3.5.12.6 REPOS

Es gibt keine Sonderbehandlung für Frames. Wird ein Frame in einem ASUP geändert, so bleibt es im Programm erhalten. Beim Wiederanfahren mit REPOS wird ein geändertes Frame berücksichtigt, sofern die Änderung im ASUP aktiviert wurde.

3.6 Werkstücknahes Istwertsystem

3.6.1 Übersicht

Definition

Unter dem Begriff "Werkstücknahes Istwertsystem" werden eine Reihe von Funktionen zusammengefasst, die dem Anwender folgendes Vorgehen ermöglichen:

- Nach Hochlauf auf ein über Maschinendaten definiertes Werkstückkoordinatensystem aufsetzen.
Merkmale:
 - keine zusätzlichen Bedienhandlungen nötig
 - in der Betriebsart JOG und AUTOMATIK wirksam
- Gültige Einstellungen bei Teileprogrammende für das folgende Teileprogramm beibehalten bezüglich:
 - aktiver Ebene
 - einstellbarer Frame (G54-G57)
 - kinematischer Transformation
 - aktiver Werkzeugkorrektur
- Über die Bedienoberfläche zwischen Werkstückkoordinatensystem und Maschinenkoordinatensystem wechseln.
- Das Werkstückkoordinatensystem per Bedienung ändern (z.B. Änderung des einstellbaren Frame oder der Werkzeugkorrektur).

3.6.2 Benutzung des werkstücknahen Istwertsystems

Voraussetzungen, Grundsätzliches

Für das System wurden die im vorigen Abschnitt beschriebenen Einstellungen getroffen. Die Voreinstellung nach Hochlauf der HMI-Software ist MKS.

Umschalten auf WKS

Die Umschaltung auf das WKS über die Bedienoberfläche bewirkt, dass die Achspositionen bezüglich des Ursprungs des WKS angezeigt werden.

Umschalten auf MKS

Die Umschaltung auf das MKS über die Bedienoberfläche bewirkt, dass die Achspositionen bezüglich des Ursprungs des MKS angezeigt werden.

Zusammenhänge zwischen Koordinatensystemen

Das folgende Bild stellt die Zusammenhänge vom Maschinenkoordinatensystem MKS bis zum Werkstückkoordinatensystem WKS dar.

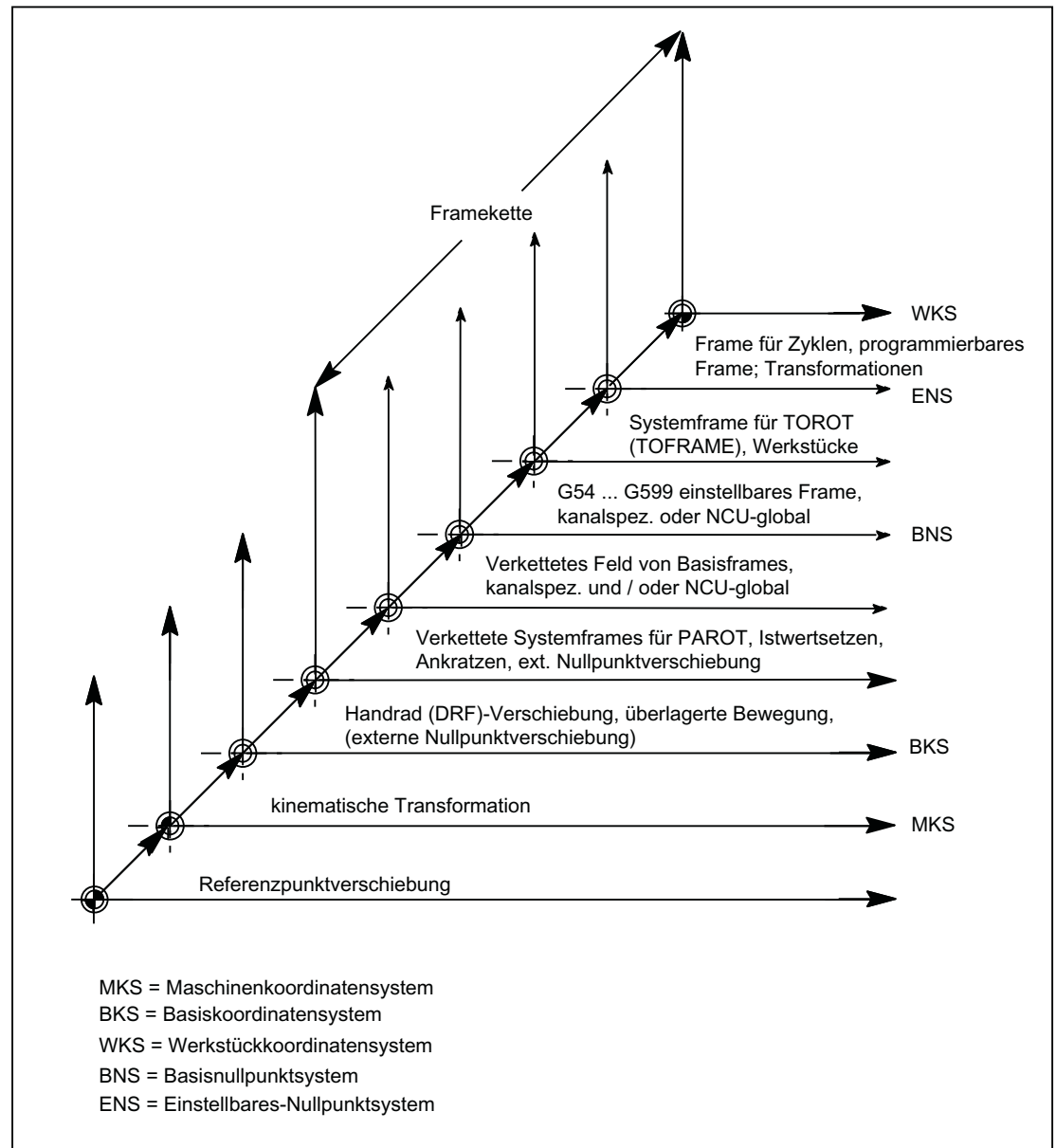


Bild 3-29 Zusammenhang Koordinatensysteme

Für weitere Informationen siehe "H2: Hilfsfunktionsausgaben an PLC (Seite 647)" und Funktionshandbuch *Werkzeuge* unter *Werkzeugkorrektur*.

Weitere Informationen

- Programmierhandbuch Grundlagen
- Funktionshandbuch Transformationen; Kinematische Transformation

- Funktionshandbuch Achsen und Spindeln; Achskopplungen
- Funktionshandbuch Technologien; Tangentialsteuerung

3.6.3 Besondere Reaktionen

Überspeichern

Überspeichern im `RESET`-Zustand von:

- Frames (Nullpunktverschiebungen)
- Aktiver Ebene
- Aktivierter Transformation
- Werkzeugkorrektur

wirkt sofort auf die Istwert-Anzeige aller Achsen im Kanal.

Eingabe über Bedientafelfront

Werden die Werte für:

"Aktiver Frame" (Nullpunktverschiebungen, Bedienbereich Parameter)

und

"Aktive Werkzeuglängenkorrektur" (Bedienbereich Parameter)

durch Bedienung an der Bedientafelfront geändert, so werden sie in der Anzeige durch eine der folgenden Maßnahmen wirksam:

- Betätigen der `RESET`-Taste
- Neuanwahl von:
 - Nullpunktverschiebung durch das Teileprogramm
 - Werkzeugkorrektur durch das Teileprogramm
- Neueinstellung von:
 - Nullpunktverschiebung durch Überspeichern
 - Werkzeugkorrektur durch Überspeichern
- Teileprogramm-Start

MD9440

Ist das HMI-Maschinendatum für die Bedientafelfront:

MD9440 `ACTIVATE_SEL_USER_DATA`

gesetzt, so werden die eingegebenen Werte im Reset-Zustand sofort aktiv.

Bei erfolgter Eingabe in gestoppter Teileprogrammbearbeitung werden die Werte mit dem Fortsetz-Start aktiv.

Istwertlesen

Wird aus \$AA_IW nach dem Aktivieren eines Frames (Nullpunktverschiebung) oder einer Werkzeugkorrektur der Istwert im WKS gelesen, so sind die aktivierten Änderungen im gelesenen Ergebnis bereits enthalten, auch wenn die Achsen noch nicht mit den aktivierten Änderungen bewegt wurden.

Die Istwerte im Einstellbaren-Nullpunktsystem ENS können für jede Achse mit der Variablen \$AA_IEN[Achse] aus dem Teileprogramm gelesen werden.

Mit \$AA_IBN[Achse] können die Istwerte im Basisnullpunkt-Koordinatensystem BNS aus dem Teileprogramm gelesen werden.

Istwertanzeige

Im WKS wird immer die programmierte Kontur angezeigt.

Auf das MKS werden folgende Verschiebungen aufgerechnet:

- Kinematische Transformation
- DRF-Verschiebung / Externe Nullpunktverschiebung
- Aktiver Frame
- Aktive Werkzeugkorrektur des aktuellen Werkzeugs

Umschalten durch PLC

Über die PLC können wahlweise die Istwerte im WKS, ENS, BNS oder MKS angezeigt werden. Von PLC-Seite kann vorgegeben werden, welches Koordinatensystem bei einer Maschine dem Werkstückkoordinatensystem entspricht.

Nach Hochlauf ist standardmäßig das MKS ausgewählt.

Durch das Signal DB19 DBB0.7 "MKS/WKS-Umschaltung" kann auch von der PLC aus auf das WKS umgeschaltet werden.

Übergaben an PLC

In Abhängigkeit vom Maschinendatum:

MD20110 / MD20112, Bit 1

werden bei der Anwahl der Werkzeuglängenkorrektur die Hilfsfunktionen (D, T, M) an die PLC ausgegeben (oder nicht).

Hinweis

Ist von der PLC das WKS ausgewählt, kann per Bedienung für die jeweilige Betriebsart dennoch zwischen WKS und MKS umgeschaltet werden.

Bei Betriebsarten- oder/und Bereichswechsel wird jedoch das von der PLC angewählte WKS ausgewertet und aktiviert (siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 29)").

3.7 Randbedingungen

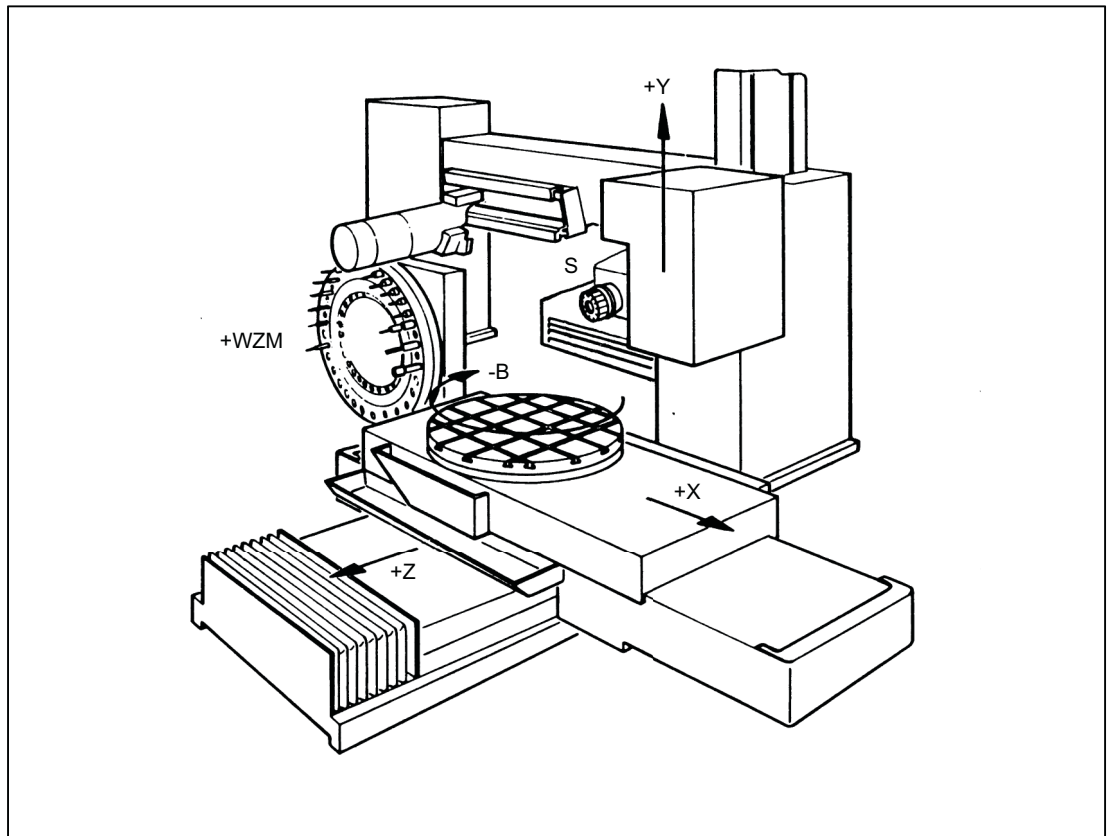
Es sind keine Randbedingungen zu beachten.

3.8 Beispiele

3.8.1 Achsen

Achskonfiguration für eine 3-Achs-Fräsmaschine mit Rundtisch

1. Maschinenachse: X1	Linearachse
2. Maschinenachse: Y1	Linearachse
3. Maschinenachse: Z1	Linearachse
4. Maschinenachse: B1	Rundtisch (zum Drehen für Mehrseitenbearbeitung)
5. Maschinenachse: W1	Rundachse für Werkzeugmagazin (WZ-Teller)
6. Maschinenachse: C1	(Spindel)
1. Geometrieachse: X	(1. Kanal)
2. Geometrieachse: Y	(1. Kanal)
3. Geometrieachse: Z	(1. Kanal)
1. Zusatzachse: B	(1. Kanal)
2. Zusatzachse: WZM	(1. Kanal)
1. Spindel: S1/C	(1. Kanal)



Parametrierung der Maschinendaten

Maschinendatum	Wert
MD10000 AXCONF_MACHAX_NAME_TAB[0]	= X1
MD10000 AXCONF_MACHAX_NAME_TAB[1]	= Y1
MD10000 AXCONF_MACHAX_NAME_TAB[2]	= Z1
MD10000 AXCONF_MACHAX_NAME_TAB[3]	= B1
MD10000 AXCONF_MACHAX_NAME_TAB[4]	= W1
MD10000 AXCONF_MACHAX_NAME_TAB[5]	= C1
MD20050 AXCONF_GEOAX_ASSIGN_TAB[0]	= 1
MD20050 AXCONF_GEOAX_ASSIGN_TAB[1]	= 2
MD20050 AXCONF_GEOAX_ASSIGN_TAB[2]	= 3
MD20060 AXCONF_GEOAX_NAME_TAB[0]	= X
MD20060 AXCONF_GEOAX_NAME_TAB[1]	= Y
MD20060 AXCONF_GEOAX_NAME_TAB[2]	= Z
MD20070 AXCONF_MACHAX_USED[0]	= 1
MD20070 AXCONF_MACHAX_USED[1]	= 2

3.8 Beispiele

Maschinendatum	Wert
MD20070 AXCONF_MACHAX_USED[2]	= 3
MD20070 AXCONF_MACHAX_USED[3]	= 4
MD20070 AXCONF_MACHAX_USED[4]	= 5
MD20070 AXCONF_MACHAX_USED[5]	= 6
MD20080 AXCONF_CHANAX_NAME_TAB[0]	= X
MD20080 AXCONF_CHANAX_NAME_TAB[1]	= Y
MD20080 AXCONF_CHANAX_NAME_TAB[2]	= Z
MD20080 AXCONF_CHANAX_NAME_TAB[3]	= B
MD20080 AXCONF_CHANAX_NAME_TAB[4]	= WZM
MD20080 AXCONF_CHANAX_NAME_TAB[5]	= S1
MD30300 IS_ROT_AX[3]	= 1
MD30300 IS_ROT_AX[4]	= 1
MD30300 IS_ROT_AX[5]	= 1
MD30310 ROT_IS_MODULO[3]	= 1
MD30310 ROT_IS_MODULO[4]	= 1
MD30310 ROT_IS_MODULO[5]	= 1
MD30320 DISPLAY_IS_MODULO[3]	= 1
MD30320 DISPLAY_IS_MODULO[4]	= 1
MD20090 SPIND_DEF_MASTER_SPIND	= 1
MD35000 SPIND_ASSIGN_TO_MACHAX[AX1]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX2]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX3]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX4]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX5]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX6]	= 1

3.8.2 Koordinatensysteme

Projektierung eines globalen Basisframes

Vorausgesetzt ist eine NC mit 2 Kanälen. Dabei gilt:

- Beide Kanäle können den globalen Basisframe schreiben.
- Der jeweils andere Kanal erkennt die Änderung nach erneuter Aktivierung des globalen Basisframes.

- Beide Kanäle können den globalen Basisframe lesen.
- Beide Kanäle können für sich den globalen Basisframe aktivieren

Maschinendaten

Maschinendatum	Wert
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[0]	= X1
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[1]	= X2
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[2]	= X3
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[3]	= X4
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[4]	= X5
MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[5]	= X6
MD18602 \$MN_MM_NUM_GLOBAL_BASE_FRAMES	= 1
MD28081 \$MC_MM_NUM_BASE_FRAMES	= 1

Maschinendaten für Kanal 1	Wert	Maschinendaten für Kanal 1	Wert
\$MC_AXCONF_CHANAX_NAME_TAB[0]	= X	\$MC_AXCONF_CHANAX_NAME_TAB[0]	= X
\$MC_AXCONF_CHANAX_NAME_TAB[1]	= Y	\$MC_AXCONF_CHANAX_NAME_TAB[1]	= Y
\$MC_AXCONF_CHANAX_NAME_TAB[2]	= Z	\$MC_AXCONF_CHANAX_NAME_TAB[2]	= Z
\$MC_AXCONF_MACHAX_USED[0]	= 1	\$MC_AXCONF_MACHAX_USED[0]	= 4
\$MC_AXCONF_MACHAX_USED[1]	= 2	\$MC_AXCONF_MACHAX_USED[1]	= 5
\$MC_AXCONF_MACHAX_USED[2]	= 3	\$MC_AXCONF_MACHAX_USED[2]	= 6
\$MC_AXCONF_GEOAX_NAME_TAB[0]	= X	\$MC_AXCONF_GEOAX_NAME_TAB[0]	= X
\$MC_AXCONF_GEOAX_NAME_TAB[1]	= Y	\$MC_AXCONF_GEOAX_NAME_TAB[1]	= Y
\$MC_AXCONF_GEOAX_NAME_TAB[2]	= Z	\$MC_AXCONF_GEOAX_NAME_TAB[2]	= Z
\$MC_AXCONF_GEOAX_ASSIGN_TAB[0]	= 1	\$MC_AXCONF_GEOAX_ASSIGN_TAB[0]	= 4
\$MC_AXCONF_GEOAX_ASSIGN_TAB[1]	= 2	\$MC_AXCONF_GEOAX_ASSIGN_TAB[1]	= 5
\$MC_AXCONF_GEOAX_ASSIGN_TAB[2]	= 3	\$MC_AXCONF_GEOAX_ASSIGN_TAB[2]	= 6

Teileprogramm im 1. Kanal

Code (Ausschnitt)	; Kommentar
...	
N100 \$P_NCBFR[0] = CTRANS(x, 10)	; Aktivierung des NC-globalen Basisframes
...	
N130 \$P_NCBFRAME[0] = CROT(X, 45)	; Aktivierung des NC-globalen Basisframes mit Drehung => ; Alarm 18310, da Drehungen bei NC-globalen Frames ; nicht erlaubt sind
...	

3.8 Beispiele

Teileprogramm im 2. Kanal

Code (Ausschnitt)	; Kommentar
. . . N100 \$P_NCBFR[0] = CTRANS(x, 10) . . .	; Der NC-globale Basisframe wirkt auch im 2.Kanal
N510 G500 X10	; Basisframe aktivieren
N520 \$P_CHBFRAME[0] = CTRANS(x, 10) . . .	; Aktueller Frame des 2. Kanals wird mit einer Verschiebung aktiviert

3.8.3 Frames

Beispiel 1

Die Kanalachse soll durch einen Geometrieachsentausch zur Geometrieachse werden.

Durch den Tausch soll der programmierbare Frame einen Translationsanteil von 10 in der x-Achse haben.

Der aktuelle einstellbare Frame soll erhalten bleiben:

FRAME_GEOAX_CHANGE_MODE = 1

Programmcode	Kommentar
\$P_UIFR[1] = CROT(X,10,Y,20,Z,30)	; Frame bleibt nach Geometrieachsen-Tausch erhalten.
G54	; Einstellbarer Frame wird aktiv.
TRANS A10	; Achsspezifische Verschiebung von A wird getauscht.
GEOAX(1,A)	; A wird zur X-Achse.
	; \$P_ACTFRAME = CROT(X,10,Y,20,Z,30) : CTRANS(X10)

Bei Transformationswechsel können gleichzeitig mehrere Kanalachsen zu Geometrieachsen werden.

Beispiel 2

Durch eine 5-Achs-Orientierungs-Transformation werden die Kanalachsen 4, 5 und 6 zu Geometrieachsen der Transformation. Die Geometrieachsen vor der Transformation werden also alle ersetzt.

Beim Einschalten der Transformation ändern sich alle aktuellen Frames.

Zur Berechnung des neuen WKS werden die achsspezifischen Frame-Anteile der Kanalachsen, die zu Geometrieachsen werden, berücksichtigt. Programmierte Drehungen vor der Transformation werden beibehalten. Nach dem Ausschalten der Transformation wird das alte WKS wieder hergestellt.

Der häufigste Anwendungsfall wird wohl der sein, dass sich die Geometrieachsen vor und nach der Transformation nicht ändern und die Frames so beibehalten werden sollen, wie sie vor der Transformation waren.

Maschinendaten:

```
$MN_FRAME_GEOAX_CHANGE_MODE = 1
```

```
$MC_AXCONF_CHANAX_NAME_TAB [0] = "CAX"
```

```
$MC_AXCONF_CHANAX_NAME_TAB [1] = "CAY"
```

```
$MC_AXCONF_CHANAX_NAME_TAB [2] = "CAZ"
```

```
$MC_AXCONF_CHANAX_NAME_TAB [3] = "A"
```

```
$MC_AXCONF_CHANAX_NAME_TAB [4] = "B"
```

```
$MC_AXCONF_CHANAX_NAME_TAB [5] = "C"
```

```
$MC_AXCONF_GEOAX_ASSIGN_TAB [0] = 1
```

```
$MC_AXCONF_GEOAX_ASSIGN_TAB [1] = 2
```

```
$MC_AXCONF_GEOAX_ASSIGN_TAB [2] = 3
```

```
$MC_AXCONF_GEOAX_NAME_TAB [0] = "X"
```

```
$MC_AXCONF_GEOAX_NAME_TAB [1] = "Y"
```

```
$MC_AXCONF_GEOAX_NAME_TAB [2] = "Z"
```

```
$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0] = 4
```

```
$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1] = 5
```

```
$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2] = 6
```

```
$MC_TRAFO_AXES_IN_1 [0] = 4
```

```
$MC_TRAFO_AXES_IN_1 [1] = 5
```

```
$MC_TRAFO_AXES_IN_1 [2] = 6
```

```
$MC_TRAFO_AXES_IN_1 [3] = 1
```

```
$MC_TRAFO_AXES_IN_1 [4] = 2
```

Programm:

Programmcode	Kommentar
\$P_NCBFRAME [0] = CTRANS (X,1,Y,2,Z,3,A,4,B,5,C,6)	
\$P_CHBFRAME [0] = CTRANS (X,1,Y,2,Z,3,A,4,B,5,C,6)	
\$P_IFRAME = CTRANS (X,1,Y,2,Z,3,A,4,B,5,C,6) : CROT (Z,45)	
\$P_PFRAME = CTRANS (X,1,Y,2,Z,3,A,4,B,5,C,6) : CROT (X,10,Y,20,Z,30)	
TRAORI	; Trafo setzt GEOAX (4,5,6)
	; \$P_NCBFRAME [0] = CTRANS (X,4,Y,5,Z,6,CAX,1,CAY,2,CAZ,3)
	; \$P_ACTBFRAME = CTRANS (X,8,Y,10,Z,12,CAX,2,CAY,4,CAZ,6)
	; \$P_PFRAME = CTRANS (X,4,Y,5,Z,6,CAX,1,CAY,2,CAZ,3) : CROT (X,10,Y,20,Z,30)
	; \$P_IFRAME = CTRANS (X,4,Y,5,Z,6,CAX,1,CAY,2,CAZ,3) : CROT (Z,45)

3.9 Datenlisten

Programmcode	Kommentar
TRAFOOF	; Ausschalten der Transformation setzt GEOAX (1,2,3) ; \$P_NCBFRAME[0] = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6) ; \$P_CHBFRAME[0] = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6) ; \$P_IFRAME = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6) : CROT(Z,45) ; \$P_PFRAME = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6) : CROT(X,10,Y,20,Z,30)

3.9 Datenlisten

3.9.1 Maschinendaten

3.9.1.1 Anzeige-Maschinendaten

Nummer	Bezeichner: \$MM_	Beschreibung
SINUMERIK Operate		
9242	MA_STAT_DISPLAY_BASE	Zahlenbasis für die Anzeige der Gelenkstellung <i>STAT</i>
9243	MA_TU_DISPLAY_BASE	Zahlenbasis für die Anzeige der Rundachsstellung <i>TU</i>
9244	MA_ORIAXES_EULER_ANGLE_NAME	Anzeige der Orientierungsachsen als Euler-Winkel
9245	MA_PRESET_FRAMEIDX	Wertablage Ankratzen und Istwertsetzen
9247	USER_CLASS_BASE_ZERO_OFF_PA	Verfügbarkeit der Basisverschiebung in Bedienbereich "Parameter"
9248	USER_CLASS_BASE_ZERO_OFF_MA	Verfügbarkeit der Basisverschiebung in Bedienbereich Maschine
9424	MA_COORDINATE_SYSTEM	Koordinatensystem für Istwertanzeige
9440	ACTIVE_SEL_USER_DATA	Aktive Daten (Frames) werden nach Änderung sofort wirksam
9449	WRITE_TOA_LIMIT_MASK	Wirksamkeit von MD9203 bezüglich Schneidendaten und ortsabhängige Korrekturen
9450	MM_WRITE_TOA_FINE_LIMIT	Grenzwert für Verschleiß fein
9451	MM_WRITE_ZOA_FINE_LIMIT	Grenzwert für Verschiebung fein

3.9.1.2 NC-spezifischen Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10000	AXCONF_MACHAX_NAME_TAB	Maschinenachsname
10600	FRAME_ANGLE_INPUT_MODE	Drehreihenfolge in Frame

Nummer	Bezeichner: \$MN_	Beschreibung
10602	FRAME_GEOAX_CHANGE_MODE	Frames beim Umschalten von Geometrieachsen
10610	MIRROR_REF_AX	Bezugsachse für das Spiegeln
10612	MIRROR_TOGGLE	Mirror umschalten
10613	NCBFRAME_RESET_MASK	Aktive NCU-globalen Basisframes nach Reset
10615	NCBFRAME_POWERON_MASK	Globalen Basisframes nach Power On zurücksetzen
10617	FRAME_SAVE_MASK	Verhalten von Frames bei SAVE- Unterprogrammen
10650	IPO_PARAM_NAME_TAB	Name der Interpolationsparameter
10660	INTERMEDIATE_POINT_NAME_TAB	Name der Zwischenpunktkoordinaten bei G2/G3
11640	ENABLE_CHAN_AX_GAP	Kanalachslücken werden erlaubt
18600	MM_FRAME_FINE_TRANS	Feinverschiebung bei FRAME (SRAM)
18601	MM_NUM_GLOBAL_USER_FRAMES	Anzahl der globalen vordefinierten Anwender-Frames (SRAM)
18602	MM_NUM_GLOBAL_BASE_FRAMES	Anzahl der globalen Basisframes (SRAM)

3.9.1.3 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20050	AXCONF_GEOAX_ASSIGN_TAB	Zuordnung Geometrieachse zu Kanalachse
20060	AXCONF_GEOAX_NAME_TAB	Geometrieachsname im Kanal
20070	AXCONF_MACHAX_USED	Maschinenachsnummer gültig im Kanal
20080	AXCONF_CHANAX_NAME_TAB	Kanalachsname im Kanal
20110	RESET_MODE_MASK	Festlegung der Steuerungs-Grundstellung nach RESET / TP-Ende
20118	GEOAX_CHANGE_RESET	Automatischen Geometrieachswechsel erlauben
20126	TOOL_CARRIER_RESET_VALUE	Wirksamer Werkzeugträger bei RESET
20140	TRAFO_RESET_VALUE	Transformationsdatensatz im Hochlauf (RESET / TP-Ende)
20150	GCODE_RESET_VALUES	Löschstellung der G-Gruppen
20152	GCODE_RESET_MODE	RESET-Verhalten der G-Gruppen
20184	TOCARR_BASE_FRAME_NUMBER	Nummer des Basiframe für Aufnahme des Tischoffsets
21015	INVOLUTE_RADIUS_DELTA	Endpunktüberwachung bei Evolvente
22532	GEOAX_CHANGE_M_CODE	M-Code bei Umschaltung der Geometrieachsen
22534	TRAFO_CHANGE_M_CODE	M-Code bei Transformationswechsel
24000	FRAME_ADD_COMPONENTS	Framekomponenten für G58 und G59
24002	CHBFRAME_RESET_MASK	RESET-Verhalten der kanalspezifischen Basisframes
24004	CHBFRAME_POWERON_MASK	Kanalspezifische Basisframes nach Power On zurücksetzen
24006	CHSFRAME_RESET_MASK	Aktive Systemframes nach Reset
24007	CHSFRAME_RESET_CLEAR_MASK	Löschen von Systemframes bei RESET
24008	CHSFRAME_POWERON_MASK	Systemframes nach POWER ON zurücksetzen
24010	PFRAME_RESET_MODE	RESET-Mode für programmierbaren Frame

3.9 Datenlisten

Nummer	Bezeichner: \$MC_	Beschreibung
24020	FRAME_SUPPRESS_MODE	Positionen bei Frameunterdrückung
24030	FRAME_ACT_SET	Einstellung des ENS-Koordinatens
24040	FRAME_ADAPT_MODE	Anpassungen von aktiven Frames
24050	FRAME_SAA_MODE	Speichern und Aktivieren von Datenhaltungsframes
24805	TRACYL_ROT_AX_FRAME_1	Rundachs-Verschiebung TRACYL 1
24855	TRACYL_ROT_AX_FRAME_2	Rundachs-Verschiebung TRACYL 2
24905	TRANSMIT_ROT_AX_FRAME_1	Rundachsverschiebung TRANSMIT 1
24955	TRANSMIT_ROT_AX_FRAME_2	Rundachsverschiebung TRANSMIT 2
28080	MM_NUM_USER_FRAMES	Anzahl der einstellbaren Frames (SRAM)
28081	MM_NUM_BASE_FRAMES	Anzahl Basisframes
28082	MM_SYSTEM_FRAME_FRAMES	Systemframes (SRAM)
28560	MM_SEARCH_RUN_RESTORE_MODE	Restore von Daten nach einer Simulation

3.9.1.4 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
32074	FRAME_OR_CORRPOS_NOTALLOWED	FRAME oder HL-Korrektur ist unzulässig
35000	SPIND_ASSIGN_TO_MACHAX	Zuordnung Spindel zu Maschinenachse

3.9.2 Settingdaten

3.9.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42440	FRAME_OFFSET_INCR_PROG	Nullpunktverschiebungen in Frames
42980	TOFRAME_MODE	Framedefinition bei TOFRAME, TOROT und PAROT

3.9.3 Systemvariablen

Bezeichner	Beschreibung
\$AA_ETRANS[Achse]	Externe Nullpunktverschiebung
\$AA_IBN [Achse]	Istwert im Basisnullpunkt-Koordinatensystem (BNS)
\$AA_IEN [Achse]	Aktueller Istwert im einstellbaren Nullpunkt-Koordinatensystem (ENS)
\$AA_OFF[Achse]	Überlagerte Bewegung für die programmierte Achse
\$AC_DRF[Achse]	Handradüberlagerung einer Achse

Bezeichner	Beschreibung
\$AC_JOG_COORD	Koordinatensystem für das manuelle Verfahren
\$P_ACSFRAME	Aktives Frame zwischen BKS und ENS
\$P_ACTBFRAME	Aktives Gesamt-Basisframe
\$P_ACTFRAME	Aktives Gesamtframe
\$P_BFRAME	1. aktives Basisframe. Entspricht \$P_CHBFRAME[0]
\$P_CHBFR[<n>]	Datenhaltungsframe: Basisframe über G500, G54...G599 aktivierbar
\$P_CHBFRAME[<n>]	Aktives Basisframe
\$P_CHBFRMASK	Basisframemaske im Kanal
\$P_CHSFRMASK	Systemframe-Maske
\$P_CYCFR	Datenhaltungsframe: Systemframe für Zyklen
\$P_CYCFRAME	Aktives Systemframe für Zyklen
\$P_EXTFR	Datenhaltungsframe: Systemframe für Externe Nullpunktverschiebung
\$P_EXTFRAME	Aktives Systemframe für Externe Nullpunktverschiebung
\$P_IFRAME	Aktives einstellbares Frame
\$P_ISO1FR	Datenhaltungsframe: Systemframe für ISO G51 . 1 Spiegeln
\$P_ISO2FR	Datenhaltungsframe: Systemframe für ISO G68 2DROT
\$P_ISO3FR	Datenhaltungsframe: Systemframe für ISO G68 3DROT
\$P_ISO4FR	Datenhaltungsframe: Systemframe für ISO G51 Scale
\$P_ISO1FRAME	Aktives Systemframe für ISO G51 . 1 Spiegeln
\$P_ISO2FRAME	Aktives Systemframe für ISO G68 2DROT
\$P_ISO3FRAME	Aktives Systemframe für ISO G68 3DROT
\$P_ISO4FRAME	Aktives Systemframe für ISO G51 Scale
\$P_NCBFR[n]	Datenhaltungsframe: NCU-globales Basisframe in der Datenhaltung, über G500, G54...G599 aktivierbar
\$P_NCBFRAME[n]	Aktives NCU-globales Basisframe
\$P_NCBFRMASK	Globale Basisframe-Maske
\$P_PARTFR	Datenhaltungsframe: Systemframe für TCARR und PAROT
\$P_PARTFRAME	Aktives Systemframe für TCARR und PAROT bei orientierbarem Werkzeugträger
\$P_PFRAME	Aktiver Programmierbarer Frame
\$P_SETFR	Datenhaltungsframe: Systemframe für Istwertsetzen
\$P_SETFRAME	Aktives Systemframe für Istwertsetzen
\$P_TOOLFR	Datenhaltungsframe: Systemframe für TOROT und TOFRAME
\$P_TOOLFRAME	Aktives Systemframe für TOROT und TOFRAME
\$P_TRAFR	Datenhaltungsframe: Systemframe für Transformationen
\$P_TRAFRAME	Aktives Systemframe für Transformationen
\$P_UBFR	1. Basisframe im Kanal in der Datenhaltung, das nach G500, G54...G599 aktiviert wird. Entspricht \$P_CHBFR[0].
\$P_UIFR[n]	Datenhaltungsframe: Einstellbarer Frame, über G500, G54...G599 aktivierbar
\$P_UIFRNUM	Nummer des aktiven Einstellbaren Frames
\$P_WPFR	Datenhaltungsframe: Systemframe für das Werkstück
\$P_WPFRAME	Aktives Systemframe für das Werkstück

K7: Kinematische Kette

4.1 Funktionsbeschreibung

4.1.1 Merkmale

Im vorliegenden Kapitel wird beschrieben, wie für NC-Funktionen wie "Kollisionsvermeidung" oder "Kinematische Transformation" die kinematische Struktur einer Maschine mittels einer kinematischen Kette abgebildet und in der Steuerung über Systemvariablen parametrisiert wird.

Die Systemvariablen werden in der NC remanent gespeichert und können über SINUMERIK Operate mittels Inbetriebnahme-archiv als "NC-Daten" archiviert bzw. eingelesen werden.

Für Funktionen wie z.B. die "Kollisionsvermeidung", die auch die Beschreibung der Maschinengeometrie benötigen, siehe Funktionshandbuch *Überwachen und kompensieren*, Kapitel *Geometrische Maschinenmodellierung*.

Hinweis

Grafischer Editor

Die Maschinenmodellierung kann, alternativ zum Schreiben der Systemvariablen in einem Teileprogramm, über die Bedienoberfläche SINUMERIK Operate erfolgen:

Bedienbereich: "Inbetriebnahme" > "NC" > "Maschinenmodell"

Änderungen am Maschinenmodell

Direkt an den Systemvariablen vorgenommene Änderungen am Maschinenmodell werden erst nach einer expliziten Anforderung zum Neuberechnen des Maschinenmodells durch Aufruf der Funktion PROTA() bzw. PROTS() an der Bedienoberfläche sichtbar.

Über die Bedienoberfläche vorgenommene Änderungen am Maschinenmodell werden sofort in die Systemvariablen der NC übernommen. Die Änderungen werden aber erst nach einer expliziten Anforderung zum Neuberechnen des Maschinenmodells durch Aufruf der Funktion PROTA() bzw. PROTS() aktiv.

Kinematische Struktur

Die kinematische Struktur einer Maschine setzt sich zusammen aus:

- Anzahl und Typ der Maschinenachsen: Linear- oder Rundachsen
- Anordnung der Maschinenachsen: Lage und Orientierung
- Abhängigkeiten der Maschinenachsen untereinander: Welche Maschinenachse verfährt mit welcher anderen Maschinenachse mit.

Kinematische Kette

Die Beschreibung der kinematischen Struktur einer Maschine erfolgt mittels einer kinematischen Kette mit folgenden Eigenschaften:

- Eine kinematische Kette besteht aus einer beliebigen Anzahl miteinander verbundener Elemente.
- Von einer kinematischen Kette können parallele Teilketten abzweigen.
- In der Steuerung liegt immer nur eine aktive kinematische Kette vor.
- Die aktive kinematische Kette beginnt mit dem Root-Element.
- Parametrierte Elemente oder Teilketten, die nicht mit dem Root-Element verbunden sind oder deren Verbindung zum Root-Element hin zeigt, sind nicht Bestandteil der aktuell wirksamen kinematischen Kette.
- Eine kinematische Kette wird in den raumfesten Koordinaten des Weltkoordinatensystems definiert.

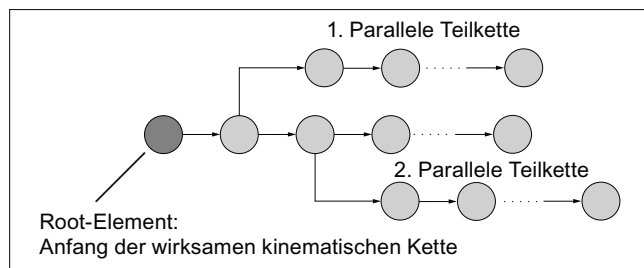


Bild 4-1 Beispiel einer kinematischen Kette

Element

Ein Element einer kinematischen Kette beschreibt im Wesentlichen die Transformation, mit der das lokale Koordinatensystem des vorhergehenden Elements in das lokale Koordinatensystem des aktuellen Elements abgebildet wird:

$$K_{n-1} \Rightarrow T_n \Rightarrow K_n$$

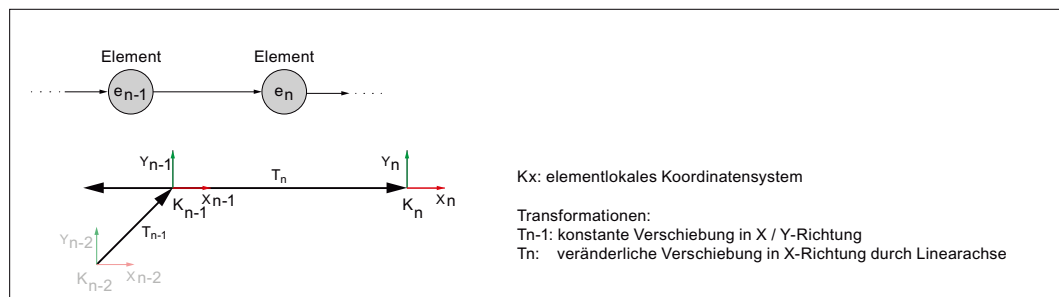


Bild 4-2 Elementlokale Koordinatensysteme

Folgende konstante Transformationen sind möglich:

- Verschiebung (Typ: OFFSET (Seite 435))
- Drehung (Typ: ROT_CONST (Seite 433))

Folgende veränderliche Transformationen, basierend auf den aktuellen Positionswerten der dem Element zugeordneten Maschinenachse (Linearachse / Rundachse), sind möglich:

- Verschiebung (Typ: AXIS_LIN (Seite 427))
- Drehung (Typ: AXIS_ROT (Seite 430))

Eine Positions- oder Orientierungsänderung in einem Element, z.B. durch Positionsänderung der zugehörigen Maschinenachse, wirkt sich auf alle **nachfolgenden** Elemente der Kette oder parallelen Teilketten aus.

Die maximale Anzahl möglicher Elemente ist über Maschinendaten (Seite 421) parametrierbar.

Parallele Teilketten

Zweigt von einem Element e_n eine parallele Teilkette ab, erfolgt das Abzweigen der Teilkette kinematisch immer **vor** dem Element. Eine Veränderung im Element e_n , z.B. eine Positionsänderung der zugehörigen Maschinenachse, wirkt sich dadurch **nicht** auf die abzweigende Teilkette aus.

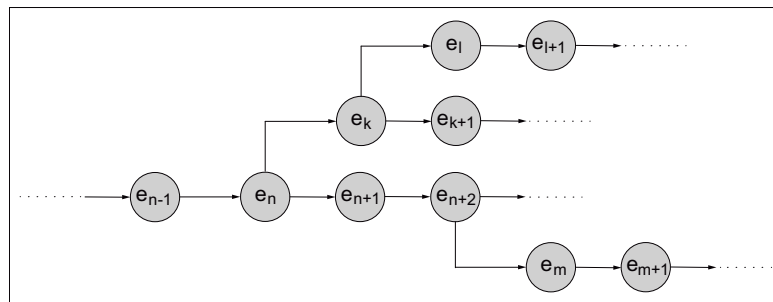


Bild 4-3 Von den Elementen e_n und e_{n+2} abzweigende parallele Teilketten

Schalter

Ein Schalter ist ein spezielles, kinematisch nicht wirksames Element (Typ: SWITCH (Seite 437)) in einer kinematischen Kette, das die Zustände EIN und AUS einnehmen kann.

Im Zustand **AUS** ist die Verbindung vom vorhergehenden zum nachfolgenden Element unterbrochen.

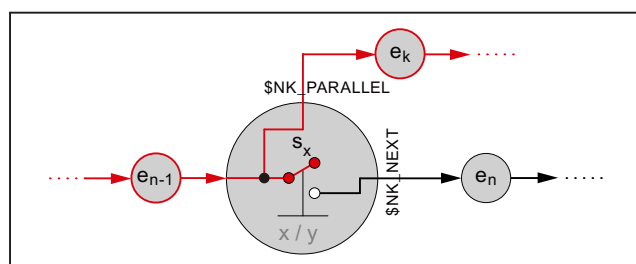


Bild 4-4 Zustand AUS

Im Zustand **EIN** ist das vorhergehende Element mit dem nachfolgenden Element verbunden.

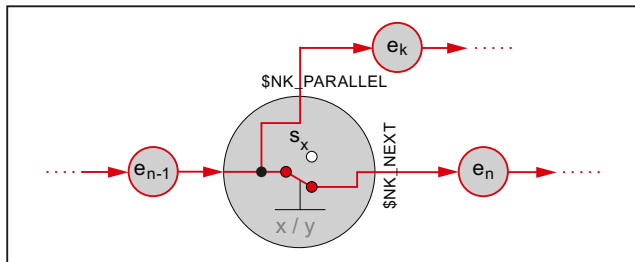


Bild 4-5 Zustand EIN

Durch den Schalter wird die Verbindung zu einem parallelen Element nicht beeinflusst.

Die maximale Anzahl möglicher Schalter ist über Maschinendaten (Seite 421) parametrierbar.

Hinweis

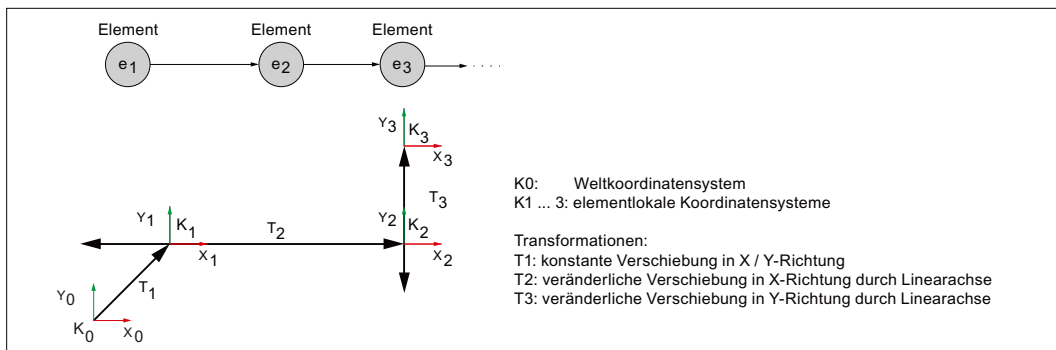
Lokales Koordinatensystem

Das lokale Koordinatensystem eines Schalters ist gegenüber dem Weltkoordinatensystem nicht gedreht.

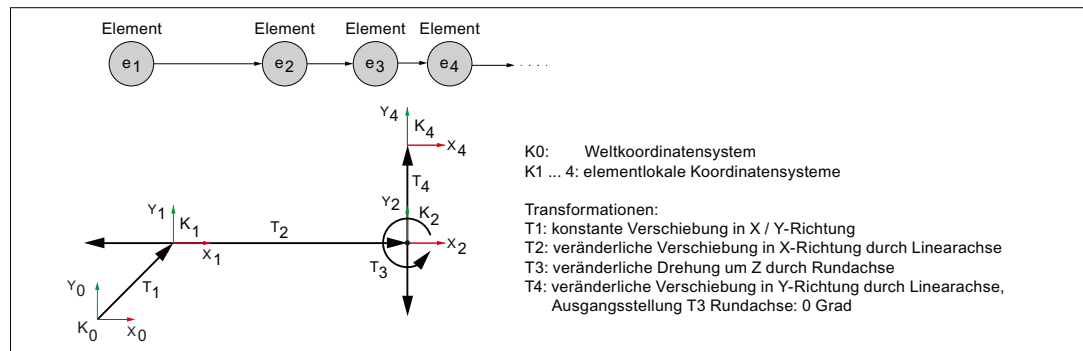
Weltkoordinatensystem

Um die kinematische Struktur einer Maschine eindeutig beschreiben zu können, werden für alle Elemente der kinematischen Kette sowohl die Orientierungs- und Verschiebungsvektoren der Maschinenachsen als auch der konstanten Drehungen / Verschiebungen, auf das Weltkoordinatensystem bezogen.

Elemente mit **Linearachsen** ändern die Orientierung der nachfolgenden Elemente nicht. Daher können die Orientierungs- und Verschiebungsvektoren der nachfolgenden Elemente ohne weitere Randbedingungen auf das Weltkoordinatensystem bezogen werden.



Elemente mit **Rundachsen** ändern die Orientierung der nachfolgenden Elemente. Daher müssen die Orientierungs- und Verschiebungsvektoren der nachfolgenden Elemente bei einer definierten Ausgangsstellung des vorhergehenden orientierungsverändernden Elementes auf das Weltkoordinatensystem bezogen werden.



Ursprung und Orientierung des Weltkoordinatensystems sind frei wählbar in Ketten-Elementen, die vor dem Root-Element definiert werden. Für das wirksame Koordinatensystem ab dem Root-Element ist folgende Anordnung erforderlich:

- Ursprung des Weltkoordinatensystems im Maschinennullpunkt
- Orientierung des Weltkoordinatensystems so, dass die Koordinatenachsen in positiver Verfahrrichtung der linearen Hauptachsen der Maschine angeordnet sind

Richtungsvektoren

Innerhalb einer kinematischen Kette werden die Richtungsvektoren, über die die Ausrichtung der Maschinenachsen angegeben wird, immer absolut, d.h. bezogen auf das Weltkoordinatensystem angegeben.

Siehe auch

Maschinendaten (Seite 421)

4.2 Inbetriebnahme

4.2.1 Allgemein

4.2.1.1 Übersicht

Die Inbetriebnahme der Funktion "Kinematische Kette" erfolgt mittels:

- Maschinendaten
 - Vorgabe des Mengengerüsts
 - Festlegung des ersten Elements der kinematischen Kette
- Systemvariablen
 - Festlegung der kinematischen Eigenschaften eines Elements
 - Verbindung der Elemente zur kinematischen Kette

4.2.1.2 Aufbau der Systemvariablen

Die Systemvariablen sind nach folgendem Schema aufgebaut:

- **\$NK_<Name>[<Index_1>]**
- **\$NK_<Name>[<Index_1>, <Index_2>]**

Allgemein

Die Systemvariablen zur Beschreibung der Elemente von kinematischen Ketten haben folgende Eigenschaften:

- Der Präfix für alle Systemvariablen der kinematischen Kette ist **\$NK_**, (N für NC, K für Kinematik).
- Die Systemvariablen sind über NC-Programme les- und schreibbar.
- Die Systemvariablen können über Archive gesichert und wieder in die NC eingelesen werden.

Datentyp

STRING

Alle Systemvariablen vom Datentyp STRING haben folgende Eigenschaften:

- Maximale String-Länge: 31 Zeichen
- Es wird keine Unterscheidung zwischen Groß- und Kleinschreibung gemacht
Beispiel: "Achse1" identisch mit "ACHSE1"
- Leer- und Sonderzeichen sind zulässig
Beispiel: "Achse1" nicht identisch mit " Achse 1"
- Namen, die mit **zwei** Unterstrichen "___" **beginnen**, sind für Systemzwecke reserviert und dürfen **nicht** für anwenderdefinierte Namen verwendet werden.

Hinweis

Führendes Leerzeichen

Da Leerzeichen gültige und der Unterscheidung dienende Zeichen sind, dürfen Namen, die mit einem **Leerzeichen**, gefolgt von **zwei** Unterstrichen "___" **beginnen**, prinzipiell für anwenderdefinierte Namen verwendet werden. Aufgrund der Verwechslungsgefahr mit Systemnamen wird dieses Vorgehen **nicht** empfohlen.

Index_1

Über Index_1 werden die einzelnen Elemente adressiert. Index 0 → 1. Element, Index 1 → 2. Element, ... n → (n+1) Element, mit n = (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)

Alle Systemvariablen eines Elements haben den gleichen Index.

Index_2

Bei Systemvariablen, die einen Vektor enthalten, werden über Index_2 die Koordinaten des Vektors adressiert.

- 0 → X-Achse
- 1 → Y-Achse
- 2 → Z-Achse

4.2.2 Maschinendaten

4.2.2.1 Maximale Anzahl Elemente

Mit dem Maschinendatum wird die maximale Anzahl von Elementen für kinematische Ketten eingestellt:

MD18880 \$MN_MM_MAXNUM_KIN_CHAIN_ELEM = <Anzahl>

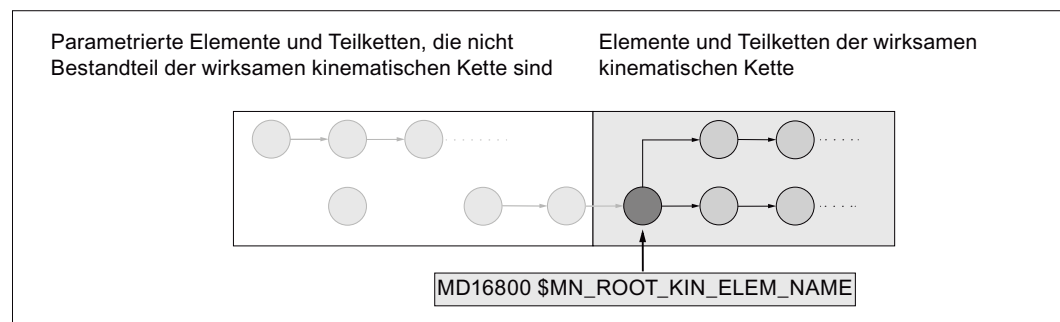
4.2.2.2 Root-Element

Mit dem Maschinendatum wird das Root-Element, d.h. das erste Element der aktuell wirksamen kinematischen Kette, festgelegt.

MD16800 \$MN_ROOT_KIN_ELEM_NAME = "<Element_Name>"

Alle parametrisierten Elemente und Teilketten, die so mit dem Root-Element verbunden sind, dass die Verbindung vom Root-Element ausgeht, sind Bestandteil der aktuell wirksamen kinematischen Kette.

Parametrisierte Elemente oder Teilketten, die nicht mit dem Root-Element verbunden sind oder deren Verbindung zum Root-Element hin zeigt, sind nicht Bestandteil der aktuell wirksamen kinematischen Kette.



4.2.2.3 Maximale Anzahl Schalter

Mit dem Maschinendatum wird die maximale Anzahl von Schaltern für kinematische Ketten eingestellt

MD18882 \$MN_MM_MAXNUM_KIN_SWITCHES = <Anzahl>

4.2.3 Systemvariablen

4.2.3.1 Übersicht

Elementunabhängige Systemvariablen

Systemvariable	Bedeutung
\$NK_SWITCH	Schaltvariable zum Ein- und Ausschalten der Schalter

Elementspezifische Systemvariablen

Die elementspezifischen Systemvariablen unterteilen sich in typunabhängige und typabhängige Variablen:

- Typunabhängige Variablen

Systemvariable	Bedeutung
\$NK_NAME	Name des aktuellen Elements e_n
\$NK_NEXT	Name des nächsten Elements e_{n+1}
\$NK_PARALLEL	Name eines vor dem aktuellen Element e_n abzweigenden parallelen Elements e_p
\$NK_TYPE	Typ des Elements

- Typabhängige Variablen

Systemvariable	Bedeutung
\$NK_OFF_DIR	Verschiebungs- oder Richtungsvektor
\$NK_AXIS	Maschinenachs- bzw. Objektname
\$NK_A_OFF	Nullpunktverschiebung bei Linear - bzw. Rundachsen
\$NK_SWITCH_INDEX	Schalterindex
\$NK_SWITCH_POS	Schalterstellung "EIN"

Die typabhängigen Variablen werden bei folgenden Typen ausgewertet:

Systemvariable	Typ				
	OFFSET	AXIS_LIN	AXIS_ROT	ROT_CON ST	SWITCH
\$NK_OFF_DIR	x	x	x	x	-
\$NK_AXIS	-	x	x	-	-
\$NK_A_OFF	-	x	x	x	-
\$NK_SWITCH_INDEX	-	-	-	-	x
\$NK_SWITCH_POS	-	-	-	-	x

Die Systemvariablen sind in den nachfolgenden Kapiteln ausführlich beschrieben.

Hinweis

Definierten Ausgangszustand herstellen

Es wird empfohlen, vor Parametrierung der kinematischen Kette einen definierten Ausgangszustand zu erzeugen. Dazu sind die Systemvariablen der kinematischen Kette mit der Funktion DELOBJ() auf ihren Defaultwert zu setzen.

Ändern von Systemvariablenwerten

Wird der Wert einer der oben aufgeführten Systemvariablen geändert, wird die Änderung auf der Bedienoberfläche, z.B. SINUMERIK Operate, sofort sichtbar. Das Maschinenmodell der NC wird aber erst nach einer expliziten Anforderung zum Neuberechnen des Maschinenmodells durch Aufruf der Funktion PROTA() bzw. PROTS() aktualisiert.

Siehe auch

Löschen von Komponenten (DELOBJ) (Seite 440)

4.2.3.2 \$NK_NAME

Funktion

In die Systemvariable ist der NC-weit eindeutige Name des Elements einzutragen. Über diesen Namen wird das Element, z.B. innerhalb von kinematischen Ketten, referenziert. Der Name wird auch im grafischen Editor von SINUMERIK Operate angezeigt.

Syntax

```
$NK_NAME [<n>] = "<Name>"
```

Bedeutung

\$NK_NAME:	Name des Elements	
	Datentyp:	STRING
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Name>:	Element-Name, max. Stringlänge: 31 Zeichen	
	Datentyp:	STRING

Beispiel

Dem 9. kinematischen Elementes wird der Name "B-Achse" zugewiesen:

Programmcode	Kommentar
N100 \$NK_NAME[8] = "B-Achse"	; 9. kin. Element, ; Name = "B-Achse"

4.2.3.3 \$NK_NEXT

Funktion

Ist das Element Bestandteil einer kinematischen Kette, ist in die Systemvariable der Namen des nachfolgenden Elements einzutragen.

Syntax

\$NK_NEXT [<n>] = "<Name>"

Bedeutung

NK_NEXT:	Name des nachfolgenden Elements	
	Datentyp:	STRING
	Wertebereich:	Alle in \$NK_NAME (Seite 423) enthaltenen Namen
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Name>:	Element-Name, max. Stringlänge: 31 Zeichen	
	Datentyp:	STRING

Beispiel

Das 9. kinematischen Elementes hat kein nachfolgendes Element:

Programmcode	Kommentar
N100 \$NK_NEXT[8] = ""	; 9. kin. Element, ; Nachfolgendes Element = ""

4.2.3.4 \$NK_PARALLEL

Funktion

In die Systemvariable ist der Name des Elements einzutragen, das **vor** dem aktuellen Element abzweigt. Das abzweigende Element liegt parallel zum aktuellen Element. Änderungen im aktuellen Element, z.B. Positionsänderungen der zugeordneten Maschinenachse, haben keine Auswirkung auf das parallele Element.

Syntax

```
$NK_PARALLEL[<n>] = "<Name>"
```

Bedeutung

\$NK_PARALLEL:	Name des parallelen Elements	
	Datentyp:	STRING
	Wertebereich:	Alle in \$NK_NAME (Seite 423) enthaltenen Namen
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Name>:	Element-Name, max. Stringlänge: 31 Zeichen	
	Datentyp:	STRING

Beispiel

Das Element "Offset zum Rundtisch" ist parallel zum 9. kinematischen Element angeordnet:

Programmcode	Kommentar
N100 \$NK_PARALLEL[8] = "Offset zum Rundtisch"	; Parallel zum 9. kin. Element, ; Name = "Offset zum Rundtisch"

4.2.3.5 \$NK_TYPE

Funktion

In die Systemvariable ist der Typ des Elements einzutragen:

Typ	Beschreibung
AXIS_LIN (Seite 427)	Das Element beschreibt eine lineare Maschinenachse (Linearachse) mit dem Richtungsvektor \$NK_OFF_DIR und der Nullpunktverschiebung \$NK_A_OFF
AXIS_ROT (Seite 430)	Das Element beschreibt eine rotatorische Maschinenachse (Rundachse) mit dem Richtungsvektor \$NK_OFF_DIR und der Nullpunktverschiebung \$NK_A_OFF oder eine Spindel. Hinweis Die Position einer Spindel wird, abhängig von der Funktion welche die kinematische Kette verwendet, unterschiedlich berücksichtigt: <ul style="list-style-type: none"> • Kollisionsvermeidung: unbestimmte Position • Kinematische Transformation: abhängig von der Einstellung in \$NT_CNTRL, Bit 1-3, wird die Position entsprechend \$NK_A_OFF oder der Drehzahlsollwert berücksichtigt
ROT_CONST (Seite 433)	Das Element beschreibt eine konstante Drehung um den Richtungsvektor \$NK_OFF_DIR und dem Winkel \$NK_A_OFF
OFFSET (Seite 435)	Das Element beschreibt eine konstante lineare Verschiebung mit dem Verschiebungsvektor \$NK_OFF_DIR
SWITCH (Seite 437)	Das Element beschreibt einen Schalter der über die Systemvariable \$NK_SWITCH (Seite 439) geschaltet wird.

Syntax

\$NK_TYPE [<n>] = "<Typ>"

Bedeutung

\$NK_TYPE:	Typ des Elements	
	Datentyp:	STRING
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Typ>:	Element-Name, max. Stringlänge: 31 Zeichen	
	Datentyp:	STRING
	Wertebereich:	"AXIS_LIN", "AXIS_ROT", "ROT_CONST", "OFFSET", "SWITCH"

Beispiel

Das 9. kinematischen Element ist eine Rundachse:

Programmcode	Kommentar
N100 \$NK_TYPE[8] = "AXIS_ROT"	; 9. kin. Element ; Type = Rundachse

4.2.3.6 Typabhängige Variablen bei \$NK_TYPE = "AXIS_LIN"

\$NK_OFF_DIR

Funktion

In die Systemvariable ist der Richtungsvektor einzutragen, entlang dessen sich die dem Element zugeordnete Linearachse \$NK_AXIS bewegt. Das Ausgangskordinatensystem ergibt sich somit aus dem Eingangskordinatensystem, verschoben um den aktuellen Positionswert der Linearachse und der in \$NK_A_OFF angegebenen Nullpunktverschiebung.

Randbedingungen:

- Der Richtungsvektor ist absolut, d.h. bezogen auf das **Weltkoordinatensystem**, anzugeben.
- Der Richtungsvektor ist so anzugeben, dass er in die positive Verfahrrichtung der Maschinenachse zeigt.
- Der Betrag des Richtungsvektors muss größer $1 \cdot 10^{-6}$ sein.

Syntax

\$NK_OFF_DIR[<n>,<k>] = <Wert>

Bedeutung

\$NK_OFF_DIR:	Richtungsvektor (X; Y; Z)	
	Datentyp:	REAL
	Wertebereich:	Richtungsvektor: $1 \cdot 10^{-6} < \text{Vektor} \leq \text{max. REAL-Wert}$
	Defaultwert:	(0.0, 0.0, 0.0)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<k>:	Koordinatenindex	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
<Wert>:	Koordinatenwert	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert $\leq x \leq$ + max. REAL-Wert

Beispiel

Die Linearachse des 9. Elements bewegt sich entlang des Richtungsvektors. Der Richtungsvektor ist der Einheitsvektor (1; 0; 0), gedreht um Z mit $\gamma=90^\circ$ in der X/Y-Ebene und um X mit $\alpha=10^\circ$ in der Y/Z Ebene, bezogen auf das Weltkoordinatensystem. Daraus ergeben sich folgende Werte für die Komponenten (x, y, z) des Richtungsvektors:

- $x = \cos(\gamma) * \cos(\alpha) = \cos(90) * \cos(10) = 0,0$
- $y = \sin(\gamma) * \cos(\alpha) = \sin(90) * \cos(10) \approx 0,985$
- $z = \sin(\alpha) = \sin(10) \approx 0,174$

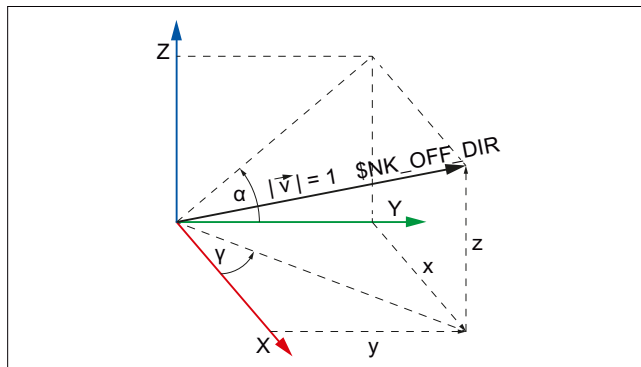


Bild 4-6 Richtungsvektor, allgemein

Programmcode	Kommentar
; 9. kinematisches Element	
N100 \$NK_OFF_DIR[8,0] = COS(90)*COS(10)	; 0 = X-Komponente
N110 \$NK_OFF_DIR[8,1] = SIN(90)*COS(10)	; 1 = Y-Komponente
N120 \$NK_OFF_DIR[8,2] = SIN(10)	; 2 = Z-Komponente

\$NK_AXIS

Funktion

In die Systemvariable ist der Name der Maschinenachse (MD10000 \$MN_AXCONF_MACHAX_NAME_TAB) einzutragen, die dem Element zugeordnet wird.

Das Ausgangskordinatensystem des Elements ergibt sich aus dem Eingangskordinatensystem, verschoben um die aktuelle Sollposition der Maschinenachse im MKS und dem in \$NK_A_OFF angegebenen Offset. In der Sollposition der Maschinenachse sind alle aktiven Nullpunktverschiebungen und Überlagerungen enthalten.

Die Maschinenachse muss, entsprechend dem in \$NK_TYPE eingetragenen Typ AXIS_LIN, eine Linearachse sein:

MD30300 \$MA_IS_ROT_AX = 0

Syntax

\$NK_AXIS [<n>] = <Name>

Bedeutung

\$NK_AXIS:	Maschinenachsname	
	Datentyp:	STRING
	Wertebereich:	Maschinenachsnamen
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Wert>:	Maschinenachsname	
	Datentyp:	STRING
	Wertebereich:	Maschinenachsenamen

Beispiel

Dem 9. kinematischen Element ist als Linearachse die Maschinenachse mit dem Namen V1 zugeordnet

Programmcode	Kommentar
N100 \$NK_AXIS[8] = "V1"	; 9. kin. Element ; Achse = Maschinenachse V1

\$NK_A_OFF**Funktion**

In die Systemvariable kann für die zugeordnete Maschinenachse (\$NK_AXIS) eine zusätzliche Nullpunktverschiebung eingetragen werden. Diese Nullpunktverschiebung ist nur innerhalb der kinematischen Kette wirksam.

Syntax

\$NK_A_OFF[<n>] = <Wert>

Bedeutung

\$NK_A_OFF:	Nullpunktverschiebung	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ ± max. REAL-Wert
	Defaultwert:	0.0
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Wert>:	Verschiebungswert	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ ± max. REAL-Wert

Beispiel

Der Nullpunkt der Linearachse des 9. kinematischen Elements ist um 30.0 mm gegenüber der modellierten Kinematik verschoben.

Die Einheit, in der der Zahlenwert interpretiert wird, ist abhängig vom aktuellen Eingabesystem (inch / metrisch).

Programmcode	Kommentar
N100 \$NK_A_OFF[8] = 30.0	; 9. kin. Element ; Nullpunktversch. = 30.0 mm

4.2.3.7 Typabhängige Variablen bei \$NK_TYPE = "AXIS_ROT"

\$NK_OFF_DIR

Funktion

In die Systemvariable ist der Richtungsvektor einzutragen, um den sich die dem Element zugeordnete Rundachse \$NK_AXIS dreht. Das Ausgangskordinatensystem berechnet sich somit aus dem Eingangskordinatensystem, gedreht um den aktuellen Positionswert der Rundachse und dem in \$NK_A_OFF angegebenen Offset, um den Richtungsvektor \$NK_OFF_DIR.

Randbedingungen:

- Der Richtungsvektor ist absolut, d.h. bezogen auf das **Weltkoordinatensystem**, anzugeben.
- Der Richtungsvektor ist so anzugeben, dass, entsprechend der "Rechten-Hand-Regel", bei positiver Drehrichtung der Rundachse der Daumen in Richtung des Vektors zeigt.
- Der Betrag des Richtungsvektors muss größer $1 \cdot 10^{-6}$ sein.

Hinweis

Spindel

Ist die zugeordnete Maschinenachse eine Spindel, wird ihre Position funktionspezifisch unterschiedlich berücksichtigt:

- Kollisionsvermeidung: unbestimmte Position
- Kinematische Transformation: abhängig von der Einstellung in \$NT_CNTRL, Bit 1-3
 - Bit x == 0 → unbestimmte Position
 - Bit x == 1 → aktueller Positionswert + \$NK_A_OFF,

Syntax

\$NK_OFF_DIR[<n>, <k>] = <Wert>

Bedeutung

\$NK_OFF_DIR:	Richtungsvektor (X; Y; Z)	
	Datentyp:	REAL
	Wertebereich:	Richtungsvektor: $1 \cdot 10^{-6} < \text{Vektor} \leq \text{max. REAL-Wert}$
	Defaultwert:	(0.0, 0.0, 0.0)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<k>:	Koordinatenindex	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
<Wert>:	Koordinatenwert	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert $\leq x \leq$ + max. REAL-Wert

Beispiel

Die Rundachse des 9. Elements dreht sich um den Richtungsvektor. Der Richtungsvektor ist der Einheitsvektor (1; 0; 0), gedreht um Z mit $\gamma=90^\circ$ in der X/Y-Ebene und um X mit $\alpha=10^\circ$ in der Y/Z Ebene, bezogen auf das Weltkoordinatensystem. Daraus ergeben sich folgende Werte für die Komponenten (x, y, z) des Richtungsvektors:

- $x = \cos(\gamma) * \cos(\alpha) = \cos(90) * \cos(10) = 0,0$
- $y = \sin(\gamma) * \cos(\alpha) = \sin(90) * \cos(10) \approx 0,985$
- $z = \sin(\alpha) = \sin(10) \approx 0,174$

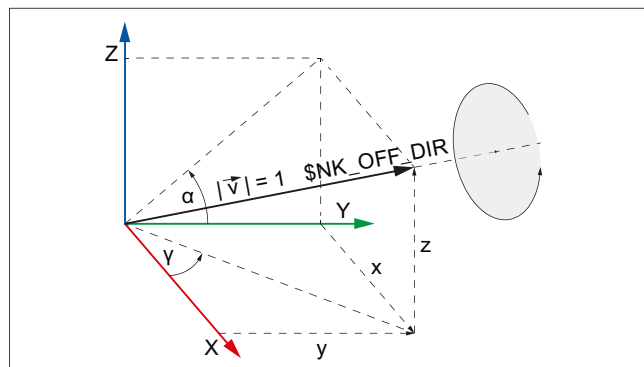


Bild 4-7 Richtungsvektor, allgemein

Programmcode	Kommentar
; 9. kinematisches Element	
N100 \$NK_OFF_DIR[8,0] = COS(90)*COS(10)	; 0 = X-Komponente
N110 \$NK_OFF_DIR[8,1] = SIN(90)*COS(10)	; 1 = Y-Komponente
N120 \$NK_OFF_DIR[8,2] = SIN(10)	; 2 = Z-Komponente

\$NK_AXIS

Funktion

In die Systemvariable ist der Name der Maschinenachse (MD10000 \$MN_AXCONF_MACHAX_NAME_TAB) einzutragen, die dem Element zugeordnet wird.

Das Ausgangskoordinatensystem des Elements ergibt sich aus dem Eingangskoordinatensystem, gedreht um die aktuelle Sollposition der Maschinenachse im MKS und dem in \$NK_A_OFF angegebenen Offset. In der Sollposition der Maschinenachse sind alle aktiven Nullpunktverschiebungen und Überlagerungen enthalten.

Der Typ der Maschinenachse muss dem in \$NK_TYPE eingetragenen Typ entsprechen:

Die Maschinenachse muss, entsprechend dem in \$NK_TYPE eingetragenen Typ AXIS_ROT, eine Rundachse sein:

MD30300 \$MA_IS_ROT_AX = 1

Syntax

\$NK_AXIS [<n>] = <Name>

Bedeutung

\$NK_AXIS:	Maschinenachsname	
	Datentyp:	STRING
	Wertebereich:	Maschinenachsamen
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Wert>:	Maschinenachsname	
	Datentyp:	STRING
	Wertebereich:	Maschinenachsenamen

Beispiel

Dem 9. kinematischen Element ist als Rundachse die Maschinenachse mit dem Namen B1 zugeordnet.

Programmcode	Kommentar
N100 \$NK_AXIS[8] = "B1"	; 9. kin. Element ; Achse = Maschinenachse B1

\$NK_A_OFF

Funktion

In die Systemvariable kann für die zugeordnete Maschinenachse (\$NK_AXIS) eine zusätzliche Nullpunktverschiebung eingetragen werden. Diese Nullpunktverschiebung ist nur innerhalb der kinematischen Kette wirksam.

Syntax

```
$NK_A_OFF[<n>] = <Wert>
```

Bedeutung

\$NK_A_OFF:	Nullpunktverschiebung	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ ± max. REAL-Wert
	Defaultwert:	0.0
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Wert>:	Verschiebungswert	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ ± max. REAL-Wert

Beispiel

Der Nullpunkt der Rundachse des 9. kinematischen Elements ist um 30.0° gegenüber der modellierten Kinematik verschoben.

Programmcode	Kommentar
N100 \$NK_A_OFF[8] = 30.0	; 9. kin. Element ; Nullpunktversch. = 30.0°

4.2.3.8 Typabhängige Variablen bei \$NK_TYPE = "ROT_CONST"**\$NK_OFF_DIR****Funktion**

In die Systemvariable ist der Richtungsvektor einzutragen, um den die konstante Drehung ausgeführt wird. Das Ausgangskordinatensystem berechnet sich somit aus dem Eingangskordinatensystem, gedreht um den in \$NK_A_OFF angegebenen Winkel um den Richtungsvektor \$NK_OFF_DIR.

Randbedingungen:

- Der Richtungsvektor ist absolut, d.h. bezogen auf das **Weltkoordinatensystem**, anzugeben.
- Der Richtungsvektor ist so anzugeben, dass, entsprechend der "Rechten-Hand-Regel", bei positiver Drehrichtung der Rundachse der Daumen in Richtung des Vektors zeigt.
- Der Betrag des Richtungsvektors muss größer $1 \cdot 10^{-6}$ sein.

Syntax

```
$NK_OFF_DIR[<n>, <k>] = <Wert>
```

Bedeutung

\$NK_OFF_DIR:	Richtungsvektor (X; Y; Z)	
	Datentyp:	REAL
	Wertebereich:	Richtungsvektor: $1 \cdot 10^{-6} < \text{Vektor} \leq \text{max. REAL-Wert}$
	Defaultwert:	(0.0, 0.0, 0.0)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<k>:	Koordinatenindex	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
<Wert>:	Koordinatenwert	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert $\leq x \leq$ + max. REAL-Wert

Beispiel

Die Ausgangskoordinatensystem des 9. Elements ergibt sich aus dem Eingangskoordinatensystem, gedreht um den in \$NK_A_OFF angegebenen Winkel um den Richtungsvektor. Der Richtungsvektor ist der Einheitsvektor (1; 0; 0), gedreht um Z mit $\gamma=90^\circ$ in der X/Y-Ebene und um X mit $\alpha=10^\circ$ in der Y/Z Ebene, bezogen auf das Weltkoordinatensystem. Daraus ergeben sich folgende Werte für die Komponenten (x, y, z) des Richtungsvektors:

- $x = \cos(\gamma) \cdot \cos(\alpha) = \cos(90) \cdot \cos(10) = 0,0$
- $y = \sin(\gamma) \cdot \cos(\alpha) = \sin(90) \cdot \cos(10) \approx 0,985$
- $z = \sin(\alpha) = \sin(10) \approx 0,174$

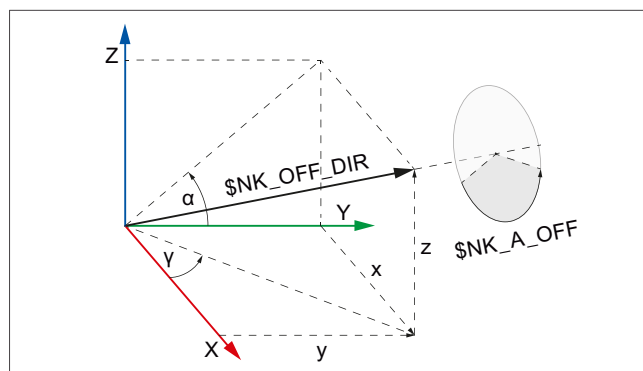


Bild 4-8 ROT_CONST

Programmcode	Kommentar
; 9. kinematisches Element	
N100 \$NK_OFF_DIR[8,0] = COS(90)*COS(10)	; 0 = X-Komponente
N110 \$NK_OFF_DIR[8,1] = SIN(90)*COS(10)	; 1 = Y-Komponente

Programmcode	Kommentar
N120 \$NK_OFF_DIR[8,2] = SIN(10)	; 2 = Z-Komponente

\$NK_A_OFF

Funktion

In die Systemvariable ist der Winkel einzutragen um den das Ausgangs- gegenüber dem Eingangskordinatensystem um den Richtungsvektor \$NK_OFF_DIR gedreht wird.

Syntax

\$NK_A_OFF[<n>] = <Wert>

Bedeutung

\$NK_A_OFF:	Drehwinkel	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ + max. REAL-Wert
	Defaultwert:	0.0
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Wert>:	Winkel	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ + max. REAL-Wert

Beispiel

Der Drehwinkel des 9. kinematischen Elements beträgt 30.0°.

Programmcode	Kommentar
N100 \$NK_A_OFF[8] = 30.0	; 9. kin. Element ; Drehwinkel. = 30.0°

4.2.3.9 Typabhängige Variablen bei \$NK_TYPE = "OFFSET"

\$NK_OFF_DIR

Funktion

In die Systemvariable ist der Verschiebungsvektor einzutragen, um den das Ausgangskordinatensystem gegenüber dem Eingangskordinatensystem verschoben ist.

Der Verschiebungsvektor ist absolut, d.h. bezogen auf das **Weltkordinatensystem**, anzugeben.

Syntax

\$NK_OFF_DIR[<n>,<k>] = <Wert>

Bedeutung

\$NK_OFF_DIR:	Verschiebungsvektor (X; Y; Z)	
	Datentyp:	REAL
	Wertebereich:	$(- \text{max. REAL-Wert}) \leq x \leq (+ \text{max. REAL-Wert})$
	Defaultwert:	(0.0, 0.0, 0.0)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<k>:	Koordinatenindex	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
	<Wert>:	Koordinatenwert
<Wert>:	Datentyp:	REAL
	Wertebereich:	$(- \text{max. REAL-Wert}) \leq x \leq (+ \text{max. REAL-Wert})$

Beispiel

Das Ausgangskordinatensystem des 9. Elements ergibt sich aus dem Eingangskordinatensystem, verschoben um den Verschiebungsvektor (x, y, z) mit den folgenden, auf das Weltkordinatensystem bezogenen Koordinaten:

- x = 10,0
- y = 20,0
- z = 30,0

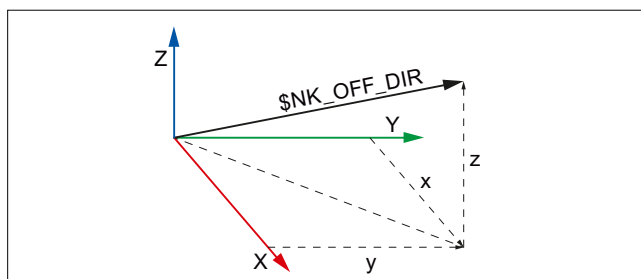


Bild 4-9 Verschiebungsvektor

Programmcode	Kommentar
; 9. kinematisches Element	
N100 \$NK_OFF_DIR[8,0] = 10.0	; 0 = X-Komponente
N110 \$NK_OFF_DIR[8,1] = 20.0	; 1 = Y-Komponente
N120 \$NK_OFF_DIR[8,2] = 30.0	; 2 = Z-Komponente

4.2.3.10 Typabhängige Variablen bei \$NK_TYPE = "SWITCH"

\$NK_SWITCH_INDEX

Funktion

Ein Schalter wird durch die Systemvariablen \$NK_SWITCH_INDEX und \$NK_SWITCH_POS (siehe nachfolgenden Absatz) gebildet.

In \$NK_SWITCH_INDEX ist der Index *i* einzutragen mit dem der Schalter über die Systemvariable \$NK_SWITCH[<i>] ein- und ausgeschaltet wird.

Syntax

```
$NK_SWITCH_INDEX[<n>] = <i>
```

Bedeutung

\$NK_SWITCH_INDEX:	Index <i>i</i> über den der Schalter durch die Systemvariable \$NK_SWITCH[<i>] angesprochen wird	
	Datentyp:	INT
	Wertebereich:	-1, 0, 1, 2, ... (\$MN_MAXNUM_KIN_SWITCHES - 1)
		-1: Der Zustand des Schalters ist konstant EIN
Defaultwert:	-1	
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<i>:	Indexwert	
	Datentyp:	INT

Beispiel

Das 9. Element der kinematischen Kette ist ein Schalter, der durch die Systemvariable \$NK_SWITCH[2] geschaltet wird.

Programmcode	Kommentar
; 9. Element	
N100 \$NK_SWITCH_INDEX[8] = 2	; Index 2

\$NK_SWITCH_POS

Funktion

Ein Schalter wird durch die Systemvariablen \$NK_SWITCH_INDEX (siehe vorhergehenden Absatz) und \$NK_SWITCH_POS gebildet.

In \$NK_SWITCH_POS ist der Wert *p* (Einschaltwert) einzutragen, bei dem der Schalter durch die Systemvariable \$NK_SWITCH[<i>] in Stellung EIN geschaltet wird.

Der Schalter hat abhängig von \$NK_SWITCH[<i>] folgende Zustände:

- EIN:** Der in \$NK_SWITCH_POS angegebene Einschaltwert p des Schalters ist gleich dem aktuellen Wert von \$NK_SWITCH[<i>].
 $\$NK_SWITCH_POS[<n>] == \$NK_SWITCH[<i>]$
 Das vorhergehende Element der kinematischen Kette ist mit dem Ausgang d.h. dem in \$NK_NEXT angegebenen nachfolgenden Element des Schalters verbunden.
- AUS:** Die in \$NK_SWITCH_POS angegebene Einschaltwert des Schalters ist ungleich dem aktuellen Wert von \$NK_SWITCH[<i>].
 $\$NK_SWITCH_POS[<n>] \neq \$NK_SWITCH[<i>]$
 Das vorhergehende Element der kinematischen Kette ist **nicht** mit dem Ausgang d.h. dem in \$NK_NEXT angegebenen nachfolgenden Element des Schalters verbunden.

Der Einschaltwert ist frei wählbar

Hinweis

Paralleles Element \$NK_PARALLEL

Die Verbindung zu einem in \$NK_PARALLEL angegebenen parallelen Element wird durch den Schalter nicht beeinflusst. D.h. das vorhergehende Element ist immer mit dem parallel vom Schalter abzweigenden Element verbunden.

Syntax

$\$NK_SWITCH_POS[<n>] = <p>$

Bedeutung

\$NK_SWITCH_POS:	Einschaltwert	
	Datentyp:	INT
	Wertebereich:	$0 \leq x \leq$ positiver max. INT-Wert
	Defaultwert:	0
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<p>:	Wert für die Schalterstellung EIN	
	Datentyp:	INT

Beispiel

Das 9. Element der kinematischen Kette ist ein Schalter, der eingeschaltet ist, wenn $\$NK_SWITCH[3] == 1$

Programmcode	Kommentar
; 9. Element	
N110 \$NK_SWITCH_POS[8] = 1	; Schalterstellung = 1

Siehe auch

\$NK_SWITCH (Seite 439)

\$MN_MAXNUM_KIN_SWITCHES (Seite 421)

\$MN_MM_MAXNUM_KIN_CHAIN_ELEM (Seite 421)

4.2.3.11 \$NK_SWITCH

Funktion

Die Schaltvariable besteht aus einem Feld von Schaltpositionen i . In diese sind die aktuellen Schaltstellungen p einzutragen.

Funktion

Zur Parametrierung (Seite 437) eines Schalters $_i$ in einer kinematischen Kette muss der Schalter $_i$ mit dem Index i der Schaltvariablen verbunden und ihm seine Schalterstellung p für den Zustand EIN zugeordnet werden.

\$NK_SWITCH_INDEX[<n>] = <i>

\$NK_SWITCH_POS[<n>] = <p>

Der Schalter $_i$ kann dann über den Index i der Schaltvariablen ein- und ausgeschaltet werden:

Einschalten: \$NK_SWITCH[<i>] = <p>

Ausschalten: \$NK_SWITCH[<i>] \neq <p>

Es können beliebig viele Schalter mit einem Index der Schaltvariablen verbunden sein.

Syntax

\$NK_SWITCH[<i>] = <p>

Bedeutung

\$NK_SWITCH<i>:	Schaltvariable mit Schaltpositionen i	
	Datentyp:	INT
	Wertebereich:	-1 $\leq x \leq$ positiver max. INT-Wert Hinweis -1: Grundzustand AUS
<i>:	Schalter-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (MD18882 \$MN_MM_MAX_NUM_KIN_SWITCHES - 1)
<p>:	Schalterstellung	
	Datentyp:	INT
	Wertebereich:	-1 $\leq x \leq$ positiver max. INT-Wert -1: Grundzustand AUS

Beispiel

Dem 9. kinematischen Elementes wird der Name "B-Achse" zugewiesen:

Programmcode	Kommentar
N100 \$NK_SWITCH[3] = 1	; Aktuelle Schalterstellung ; für Schalter[3] = 1

4.3 Programmierung**4.3.1 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>]
	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 projizierte 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 projizierte Anzahl Komponenten	

4.3.2 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:	$-1 \leq x \leq (\text{max. Anzahl projektierter Komponenten} - 1)$
	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

4.4 Beispiel

4.4.1 Vorgaben

Allgemeines

Anhand einer 5-Achs-Maschine mit drei unterschiedlichen Werkzeugköpfen, die wechselweise zum Einsatz kommen, wird beispielhaft das prinzipielle Vorgehen zur Parametrierung der kinematischen Kette mit drei Schaltern über ein Teileprogramm gezeigt. Im Teileprogramm werden alle für die kinematische Kette relevanten Systemvariablen geschrieben:

- Kinematische Kette \$NK_...

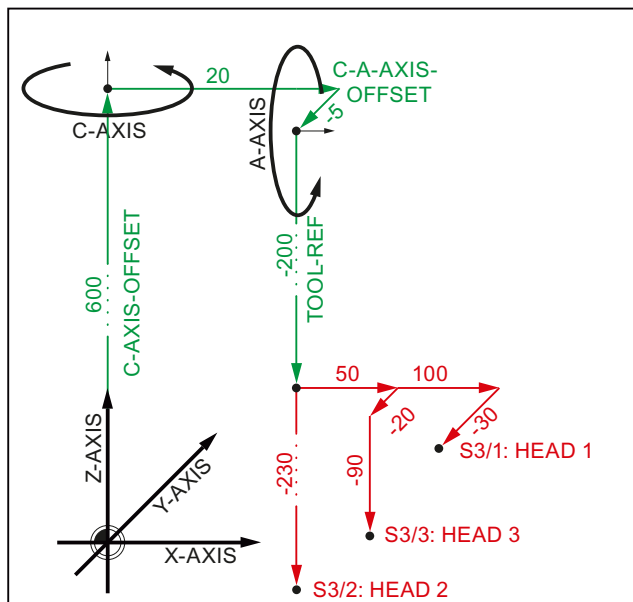
Options- und Maschinendaten

Für das Beispiel sind folgende Options- und Maschinendaten einzustellen:

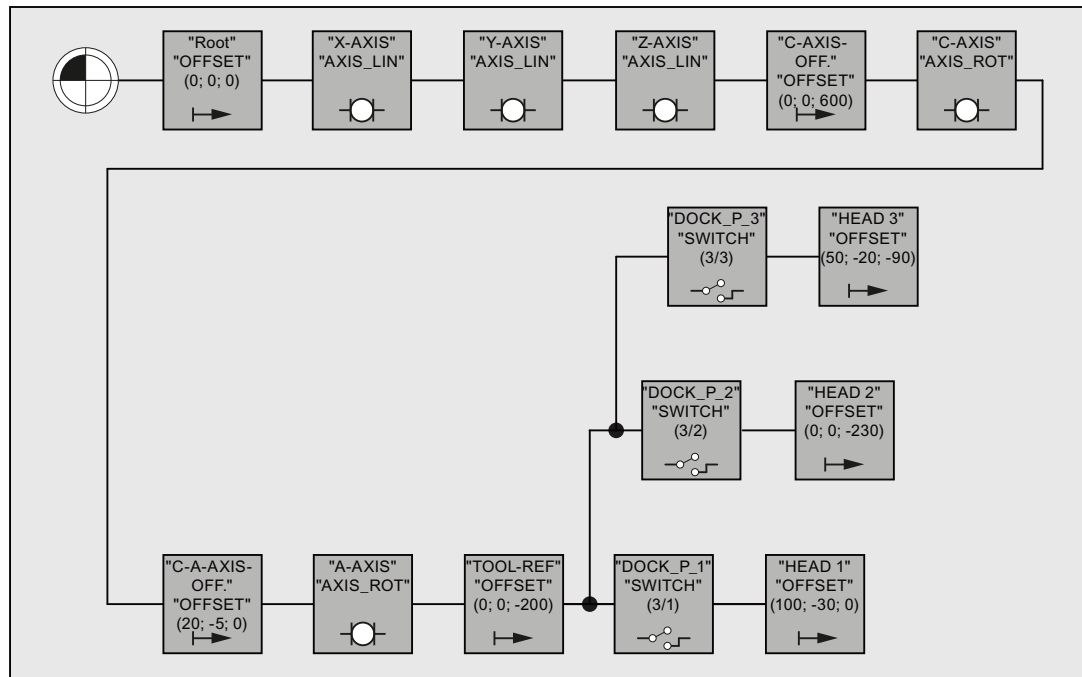
- MD19830 \$ON_COLLISION_MASK.Bit 0 = 1
- MD18880 \$MN_MM_MAXNUM_KIN_CHAIN_ELEM = 15

Kinematische Kette

Schematische Darstellung der Maschinenkinematik



Elemente der kinematischen Kette



Die kinematische Kette beginnt mit einem Element vom Typ "Offset". Diesem werden bei einer vollständigen Parametrierung der Kollisionsvermeidung alle statischen Schutzbereiche der Maschine zugeordnet.

Auf das Offset-Element folgen die kinematischen Elemente der linearen Maschinenachsen X, Y und Z. Sowie die Offset- und kinematischen Elemente der Rundachsen C und A.

Nach dem Offset-Element für den Referenzpunkt der Werkzeugköpfe folgen die drei Schalter zum Ein- und Ausschalten der Werkzeugköpfe. Alle drei Schalter beziehen sich durch $\$NK_INDEX = 3$ auf dieselbe Schaltervariable $\$NK_SWITCH[3]$. Durch die unterschiedlichen Schalterpositionen (1, 2, 3) wird nur die Teilkette des jeweils aktiven Werkzeugkopfs wirksam.

4.4.2 Teileprogramm des Maschinenmodells

Programmcode

```

;=====
; Definitionen
;=====
N10 DEF INT KIE_CNTR ; ZAEHLER FÜR ELEMENTE DER KIN. KETTEN
N20 DEF INT RETVAL
;
;=====
; Initialisierung der Kollisionsdaten
;=====
; Alle Parameter auf ihre Grundstellungswerte zuruecksetzen:
N30 RETVAL = DELOBJ("KIN_CHAIN_ELEM")
N40 KIE_CNTR = 0
;
;=====
; Kinematische Kette
;=====
; KE1: OFFSET: Root
; -----
N50 $NK_TYPE[KIE_CNTR] = "OFFSET"
N60 $NK_NAME[KIE_CNTR] = "ROOT"
N70 $NK_NEXT[KIE_CNTR] = "X-AXIS"
N80 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: LINEARACHSE: X-Achse
; -----
N90 $NK_TYPE[KIE_CNTR] = "AXIS_LIN"
N100 $NK_NAME[KIE_CNTR] = "X-AXIS"
N110 $NK_NEXT[KIE_CNTR] = "Y-AXIS"
N120 $NK_AXIS[KIE_CNTR] = "X1"
;
N130 $NK_OFF_DIR[KIE_CNTR,0] = 1.0 ; X
N140 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: LINEARACHSE: Y-Achse
; -----
N150 $NK_TYPE[KIE_CNTR] = "AXIS_LIN"
N160 $NK_NAME[KIE_CNTR] = "Y-AXIS"
N170 $NK_NEXT[KIE_CNTR] = "Z-AXIS"
N180 $NK_AXIS[KIE_CNTR] = "Y1"
;
N190 $NK_OFF_DIR[KIE_CNTR,1] = 1.0 ; Y
N200 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: LINEARACHSE: Z-Achse
; -----
N210 $NK_TYPE[KIE_CNTR] = "AXIS_LIN"
N220 $NK_NAME[KIE_CNTR] = "Z-AXIS"
N230 $NK_NEXT[KIE_CNTR] = "C-AXIS-OFFSET"
N240 $NK_AXIS[KIE_CNTR] = "Z1"
;

```

Programmcode

```

N250 $NK_OFF_DIR[KIE_CNTR,2] = 1.0      ; Z
N260 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: OFFSET: C-Achs
; -----
N270 $NK_TYPE[KIE_CNTR] = "OFFSET"
N280 $NK_NAME[KIE_CNTR] = "C-AXIS-OFFSET"
N290 $NK_NEXT[KIE_CNTR] = "C-AXIS"
;
N300 $NK_OFF_DIR[KIE_CNTR,2] = 600.0    ; Z-Richtung
N310 KIE_CNTR = KIE_CNTR + 1 ;
;
; -----
; Kinematisches Element: RUNDACHSE: C-Achse
; -----
N320 $NK_TYPE[KIE_CNTR] = "AXIS_ROT"
N330 $NK_NAME[KIE_CNTR] = "C-AXIS"
N340 $NK_NEXT[KIE_CNTR] = "C-A-OFFSET"
N350 $NK_AXIS[KIE_CNTR] = "C1"
;
N360 $NK_OFF_DIR[KIE_CNTR,2] = 1.0      ; Z-Richtung
N370 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: OFFSET: C-A-Achs
; -----
N380 $NK_TYPE[KIE_CNTR] = "OFFSET"
N390 $NK_NAME[KIE_CNTR] = "C-A-OFFSET"
N400 $NK_NEXT[KIE_CNTR] = "A-AXIS"
;
N410 $NK_OFF_DIR[KIE_CNTR,0] = 20.0     ; X-Richtung
N420 $NK_OFF_DIR[KIE_CNTR,1] = -5.0    ; Y-Richtung
N430 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: RUNDACHSE: A-Achse
; -----
N440 $NK_TYPE[KIE_CNTR] = "AXIS_ROT"
N450 $NK_NAME[KIE_CNTR] = "A-AXIS"
N460 $NK_NEXT[KIE_CNTR] = "TOOL-REF"
N470 $NK_AXIS[KIE_CNTR] = "A1"
;
N480 $NK_OFF_DIR[KIE_CNTR,0] = 1.0      ; X-Richtung
N490 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: OFFSET: Werkzeug-Referenz
; -----
N500 $NK_TYPE[KIE_CNTR] = "OFFSET"
N510 $NK_NAME[KIE_CNTR] = "TOOL-REF"
N520 $NK_NEXT[KIE_CNTR] = "DOCKING_POINT 1"
;
N530 $NK_OFF_DIR[KIE_CNTR,2] = -200.0   ; Z-Richtung
N540 KIE_CNTR = KIE_CNTR + 1
;

```

Programmcode

```

; -----
; Kinematisches Element: Schalter 3/1
; -----
N550 $NK_TYPE[KIE_CNTR] = "SWITCH"
N560 $NK_NAME[KIE_CNTR] = "DOCKING_POINT 1"
N570 $NK_NEXT[KIE_CNTR] = "HEAD 1"
N580 $NK_PARALLE[KIE_CNTR] = "DOCKING_POINT 2"
;
N590 $NK_SWITCH_INDEX[KIE_CNTR] = 3      ; Index 3
N600 $NK_SWITCH_POS[KIE_CNTR] = 1       ; Schalterstellung 1
N610 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: Schalter 3/2
; -----
N620 $NK_TYPE[KIE_CNTR] = "SWITCH"
N630 $NK_NAME[KIE_CNTR] = "DOCKING_POINT 2"
N640 $NK_NEXT[KIE_CNTR] = "HEAD 2"
N650 $NK_PARALLE[KIE_CNTR] = "DOCKING_POINT 3"
;
N660 $NK_SWITCH_INDEX[KIE_CNTR] = 3      ; Index 3
N670 $NK_SWITCH_POS[KIE_CNTR] = 2       ; Schalterstellung 2
N680 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: Schalter 3/3
; -----
N690 $NK_TYPE[KIE_CNTR] = "SWITCH"
N700 $NK_NAME[KIE_CNTR] = "DOCKING_POINT 3"
N710 $NK_NEXT[KIE_CNTR] = "HEAD 3"
N720 $NK_PARALLE[KIE_CNTR] = ""
N730 $NK_SWITCH_INDEX[KIE_CNTR] = 3      ; Index 3
N740 $NK_SWITCH_POS[KIE_CNTR] = 3       ; Schalterstellung 3
N750 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: OFFSET: HEAD 1
; -----
N760 $NK_TYPE[KIE_CNTR] = "OFFSET"
N770 $NK_NAME[KIE_CNTR] = "HEAD 1"
N780 $NK_NEXT[KIE_CNTR] = ""
;
N790 $NK_OFF_DIR[KIE_CNTR,0] = 100.      ; X-Richtung
N800 $NK_OFF_DIR[KIE_CNTR,1] = -30.     ; Y-Richtung
N810 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: OFFSET: HEAD 2
; -----
N820 $NK_TYPE[KIE_CNTR] = "OFFSET"
N830 $NK_NAME[KIE_CNTR] = "HEAD 2"
N840 $NK_NEXT[KIE_CNTR] = ""
;
N850 $NK_OFF_DIR[KIE_CNTR,2] = -230.    ; Z-Richtung
N860 KIE_CNTR = KIE_CNTR + 1
;

```


Programmcode

```

; -----
; Kinematisches Element: OFFSET: HEAD 3
; -----
N870 $NK_TYPE[KIE_CNTR] = "OFFSET"
N880 $NK_NAME[KIE_CNTR] = "HEAD 3"
N890 $NK_NEXT[KIE_CNTR] = ""
;
N900 $NK_OFF_DIR[KIE_CNTR,0] = 50.      ; X-Richtung
N910 $NK_OFF_DIR[KIE_CNTR,1] = -20.    ; Y-Richtung
N920 $NK_OFF_DIR[KIE_CNTR,2] = -90.    ; Z-Richtung
N930 KIE_CNTR = KIE_CNTR + 1
;===== ENDE =====

```

4.5 Datenlisten

4.5.1 Maschinendaten

4.5.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
16800	ROOT_KIN_ELEM_NAME	Name des ersten Elements der aktiven kinematischen Kette
18880	MM_MAXNUM_KIN_CHAIN_ELEM	Maximale Anzahl Elemente für kinematische Ketten
18882	MM_MAX_NUM_KIN_SWITCHES	Maximale Anzahl Schalter für kinematische Ketten

4.5.2 Systemvariablen

Bezeichner	Beschreibung
\$NK_NAME	Name des kinematischen Elements
\$NK_NEXT	Name des nächsten Elements in der kinematischen Kette
\$NK_PARALLEL	Name des ersten Elements einer vor dem Element parallel abzweigenden kinematischen Kette
\$NK_TYPE	Typ des kinematischen Elements
\$NK_OFF_DIR	Abhängig von \$NK_TYPE: Verschiebungs- oder Richtungsvektor
\$NK_AXIS	Name der zugeordneten Maschinenachse bzw. Objektname
\$NK_A_OFF	Nullpunktverschiebung im Element bei Linear - bzw. Rundachsen
\$NK_SWITCH_INDEX	Index i, über den der Schalter durch die Systemvariable \$NK_SWITCH[<i>] angesprochen wird
\$NK_SWITCH_POS	Schalterstellung EIN
\$NK_SWITCH	Schaltvariable

B1: Bahnsteuerbetrieb, Genauhalt, LookAhead

5.1 Kurzbeschreibung

Genauhalt oder Genauhaltbetrieb

Genauhaltbetrieb ist ein Verfahrensmodus, bei dem am Ende eines jeden Verfahrnsatzes alle an der Verfahrnbewegung beteiligten Achsen (außer Achsen von satzübergreifenden Verfahrnbewegungen) bis zum Stillstand abgebremst werden. Der Satzwechsel zum nachfolgenden Verfahrnsatz erfolgt erst, wenn alle an der Verfahrnbewegung beteiligten Achsen ihre programmierte Zielposition in Abhängigkeit des gewählten Genauhaltkriteriums erreicht haben.

Bahnsteuerbetrieb

Bahnsteuerbetrieb ist ein Verfahrensmodus, bei dem die NC versucht, die programmierte Bahngeschwindigkeit möglichst konstant beizubehalten. Insbesondere soll dabei das Abbremsen der Bahnachsen an den Satzgrenzen des Teileprogramms vermieden werden.

LookAhead

LookAhead ist eine Funktion zur Optimierung des Bahnsteuerbetriebs.

Um an Werkstücken hohe Oberflächengüten zu erzielen, ist eine gleichmäßige Bearbeitung notwendig. Deshalb sollte es während der Bearbeitung möglichst nicht zu Schwankungen der Bahngeschwindigkeit kommen. Ohne LookAhead betrachtet die NC zur Ermittlung der möglichen Bahngeschwindigkeit nur den unmittelbar auf den aktuellen Verfahrnsatz folgenden Verfahrnsatz. Beinhaltet der folgende Verfahrnsatz nur einen kurzen Verfahrweg, muss die NC die Bahngeschwindigkeit vermindern (Bremsen im aktuellen Verfahrnsatz), um eventuell rechtzeitig am Satzende des Folgesatzes anhalten zu können.

Durch "Vorausschauen" über eine parametrierbare Anzahl von dem aktuellen Verfahrnsatz folgenden Verfahrnsätzen kann mit LookAhead eine unter Umständen wesentlich höhere Bahngeschwindigkeit erzielt werden, da der NC jetzt wesentlich mehr Verfahrnsätze bzw. Verfahrweg zur Berechnung zur Verfügung steht.

Dadurch ergeben sich folgende Vorteile:

- Bearbeitung mit durchschnittlich höherer Bahngeschwindigkeit
- Verbesserung der Oberflächengüte durch Vermeidung von Brems- und Beschleunigungsvorgängen

Glättung der Bahngeschwindigkeit

"Glättung der Bahngeschwindigkeit" ist eine Funktion speziell für Anwendungen, die eine möglichst gleichmäßige Bahngeschwindigkeit erfordern (z. B. Hochgeschwindigkeitsfräsen im Formenbau). Dazu wird bei der Glättung der Bahngeschwindigkeit auf Brems- und Beschleunigungsvorgänge verzichtet, die zu hochfrequenten Anregungen von Maschinenresonanzen führen würden.

Dadurch ergeben sich folgende Vorteile:

- Verbesserung der Oberflächengüte und der Stückzeit durch Vermeidung von Anregungen von Maschinenresonanzen
- Konstanter Verlauf der Bahngeschwindigkeit bzw. Schnittgeschwindigkeiten durch Vermeidung von "überflüssigen" Beschleunigungsvorgängen, d. h. Beschleunigungsvorgänge, die keinen großen Gewinn hinsichtlich der Programmlaufzeit bewirken.

Anpassung der Bahndynamik

Die "Anpassung der Bahndynamik" ist neben der "Glättung der Bahngeschwindigkeit" eine weitere Funktion zur Vermeidung hochfrequenter Anregungen von Maschinenresonanzen bei gleichzeitiger Optimierung der Bahndynamik. Dazu werden hochfrequente Änderungen der Bahngeschwindigkeit automatisch mit geringeren Ruck- oder Beschleunigungswerten als den in den Maschinendaten parametrisierten Dynamikgrenzwerten durchgeführt.

Somit wirken bei niederfrequenten Änderungen der Bahngeschwindigkeit die vollen Dynamikgrenzwerte, während bei hochfrequenten Änderungen durch die automatische Dynamikanpassung nur die reduzierten Dynamikgrenzwerte wirken.

Dynamikmodus für Bahninterpolation

Zur Optimierung der Bahndynamik zählen auch die Technologie-spezifischen Dynamikeinstellungen, die für unterschiedliche Bearbeitungstechnologien (u. a. Gewindebohren, Schruppen, Schlichten) voreingestellt sind und im Teileprogramm durch Aufruf des jeweiligen Dynamikmodus aktiviert werden können.

Freiformflächenmodus

Jegliche Schwankung der Krümmung bzw. Torsion führt zu einer Veränderung der Bahngeschwindigkeit. Dadurch erhält man in der Regel bei der Bearbeitung von Freiformflächen-Werkstücken unnötige Brems- und Beschleunigungsvorgänge, die sich negativ auf die Qualität der Werkstückoberflächen auswirken können.

Für die Freiformflächenbearbeitung stehen daher folgende Funktionen zur Verfügung:

- Funktion "Freiformflächenmodus: Grundfunktionen"
Damit wird die Festlegung des Verlaufs der Bahngeschwindigkeit "unempfindlicher" gegenüber Schwankungen der Krümmung und Torsion.
- Funktion "Freiformflächenmodus: Erweiterungsfunktion"
Diese Erweiterung der LookAhead-Standardfunktionalität dient zur Berechnung des Bahngeschwindigkeitsprofils bei der Freiformflächenbearbeitung.

Die Vorteile des Freiformflächenmodus liegen in einer gleichmäßigeren Werkstückoberfläche und einer geringeren Belastung der Maschine.

Kompression von Linearsätzen

Nach Abschluss der Konstruktion eines Werkstücks mit einem CAD/CAM-System übernimmt dieses gewöhnlich auch die Generierung des entsprechenden Teileprogramms zur Erzeugung der Werkstückoberfläche. Dabei verwenden die meisten CAD/CAM-Systeme zur Beschreibung auch gekrümmter Abschnitte der Werkstückoberfläche Linearsätze. Zur Einhaltung der erforderlichen Konturgenauigkeit sind dabei gewöhnlich sehr viele Stützstellen notwendig. Demzufolge ergeben sich viele lineare Verfahrssätze mit meist sehr kurzen Verfahrwegen.

Durch den Einsatz einer "Kompressor-Funktion" erfolgt eine nachträgliche Annäherung an die durch die Linearsätze vorgegebene Kontur durch Polynom-Sätze. Dabei wird eine parametrierbare Anzahl von Linearsätzen durch einen Polynom-Satz ersetzt.

Vorteile:

- Reduzierung der Anzahl von Verfahrssätzen
- Erhöhung der Bahngeschwindigkeit
- Erhöhung der Oberflächengüte
- Stetige Satzübergänge

Kompression kurzer Spline-Sätze

Ein Spline definiert eine Kurve, die aus Polynomen 2. oder 3. Grades zusammengesetzt wird. Mit der Spline-Interpolation kann die Steuerung aus nur wenigen vorgegebenen Stützpunkten einer Sollkontur einen glatten Kurvenverlauf erzeugen.

Die Vorteile der Spline-Interpolation gegenüber der Linear-Interpolation sind:

- Reduzierung der Anzahl von benötigten Teileprogrammsätzen zur Beschreibung einer gekrümmten Kontur.
- Weicher, Mechanik-schonender Kurvenverlauf auch beim Übergang zwischen den Teileprogrammsätzen.

Die Nachteile der Spline-Interpolation gegenüber der Linear-Interpolation sind:

- Für eine Spline-Kurve kann kein exakter Kurvenverlauf, sondern nur ein Toleranzband vorgegeben werden, innerhalb dessen die Spline-Kurve liegen soll.

Wie bei der Linear-Interpolation können auch bei der Aufbereitung von Splines so kurze Sätze entstehen, dass zur Interpolation dieser Spline-Sätze die Bahngeschwindigkeit reduziert werden muss. Dies ist auch dann der Fall, wenn der Spline eigentlich eine lange, glatte Kurve darstellt. Mit der Funktion "Kompression kurzer Spline-Sätze" können die Spline-Sätze so zusammengefasst werden, dass die resultierende Satzlänge ausreichend groß ist und nicht zu einer Verringerung der Bahngeschwindigkeit führt.

5.2 Genauhaltbetrieb

Genauhalt oder Genauhaltbetrieb

Genauhalt oder Genauhaltbetrieb ist ein Verfahrensmodus, bei dem am Ende eines jeden Verfahrssatzes alle an der Verfahrbewegung beteiligten Bahnachsen und Zusatzachsen, die nicht satzübergreifend verfahren, zum Satzende bis zum Stillstand abgebremst werden. Der Satzwechsel zum nachfolgenden Verfahrssatz erfolgt erst, wenn alle an der Verfahrbewegung beteiligten Achsen ihre programmierte Zielposition in Abhängigkeit der gewählten Genauhaltbedingung erreicht haben.

Daraus ergibt sich folgendes Verhalten:

- Durch das Abbremsen der Achsen und die Wartezeit bis zum Erreichen des Zustandes "Genauhalt" für alle beteiligten Maschinenachsen verlängert sich die Programmlaufzeit gegenüber dem Bahnsteuerbetrieb erheblich.
- Bei Genauhaltbetrieb während der Bearbeitung kann es zu Freischneidmarken auf der Werkstückoberfläche kommen.

Genauhaltbedingung

Folgende Genauhaltbedingung können eingestellt werden:

- "Genauhalt grob"
- "Genauhalt fein"
- "Interpolator-Ende"

Anwendung

Der Genauhaltbetrieb ist immer dann zu verwenden, wenn die programmierte Kontur exakt abgefahren werden muss.

Aktivierung

In einem Programm kann der Genauhaltbetrieb über folgende Befehle modal oder satzweise vorgegeben werden:

Befehl	Bedeutung
G60	Der Genauhaltbetrieb ist ab dem aktuellen Satz modal wirksam.
G9	Der Genauhaltbetrieb ist im aktuellen Satz wirksam.

Genauhaltbedingung "Genauhalt grob" und "Genauhalt fein"

Die Genauhaltbedingung "grob" bzw. "fein" ist von eine Maschinenachse erfüllt, wenn ihr aktueller Schleppabstand kleiner bzw. gleich dem im Maschinendatum parametrisierten Toleranzbereich um die Sollposition ist:

- MD36000 \$MA_STOP_LIMIT_COARSE (Genauhalt grob)
- MD36010 \$MA_STOP_LIMIT_FINE (Genauhalt fein)

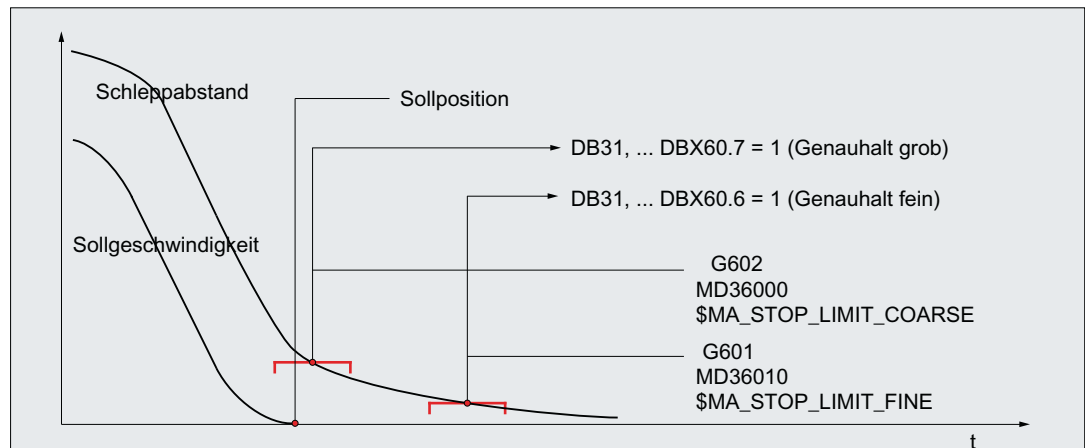


Bild 5-1 Toleranzfenster der Genauhaltbedingungen

Hinweis

Die Toleranzfenster der Genauhaltbedingungen "Genauhalt grob" und "Genauhalt fein" sollten so parametrisiert werden, dass folgende Forderung erfüllt ist:

"Genauhalt grob" > "Genauhalt fein"

Genauhaltbedingung "Interpolator-Ende"

Bei Genauhaltbedingung "Interpolator-Ende" erfolgt der Satzwechsel zum nachfolgenden Verfahrtsatz, sobald alle an der Verfahrbewegung beteiligten Bahnachsen und Zusatzachsen, die nicht satzübergreifend verfahren, Sollwertbezogen ihre im Verfahrtsatz programmierte Position erreicht haben. D. h., der Interpolator hat den Verfahrtsatz abgefahren.

Die Istposition bzw. der Schleppabstand der beteiligten Maschinenachsen wird bei Genauhaltbedingung "Interpolator-Ende" nicht betrachtet. Dadurch kann es, abhängig von der Dynamik der Maschinenachsen, zu einem im Vergleich zu den Genauhaltbedingungen "Genauhalt grob" und "Genauhalt fein" größeren Verschleifen der Kontur an den Satzübergängen kommen.

Aktivierung der programmierbaren Genauhaltbedingungen

Die programmierbaren Genauhaltbedingungen werden durch folgende Befehle aktiviert:

Befehl	Genauhaltbedingung
G601	Genauhalt fein
G602	Genauhalt grob
G603	Interpolator-Ende

Satzwechsel in Abhängigkeit der aktiven Genauhaltbedingung

Das nachfolgende Bild veranschaulicht den Zeitpunkt des Satzwechsels in Abhängigkeit von der gewählten Genauhaltbedingung.

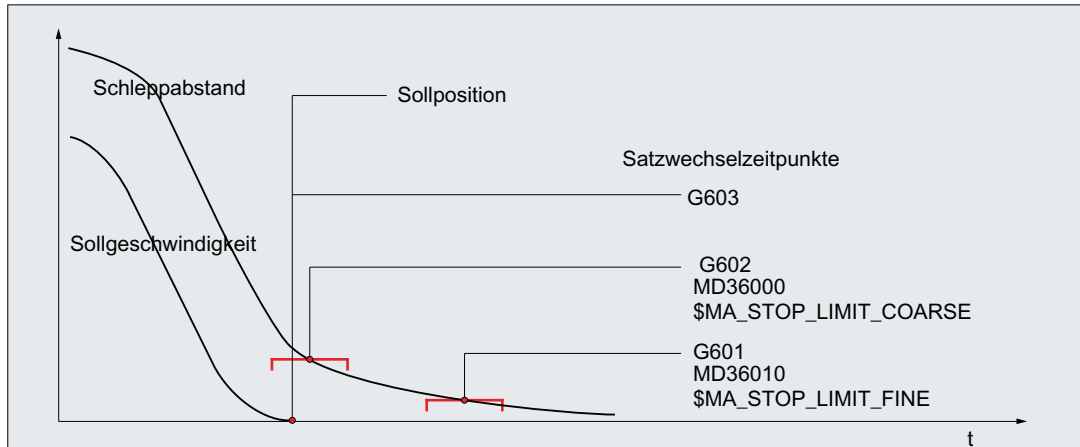


Bild 5-2 Satzwechsel in Abhängigkeit der aktiven Genauhaltbedingung

Bewertungsfaktor für Genauhaltbedingungen

Eine Parametersatz-abhängige Bewertung der Genauhaltbedingungen kann über das folgende achsspezifische Maschinendatum vorgegeben werden:

MD36012 \$MA_STOP_LIMIT_FACTOR[<Parametersatz>] = <Wert>

Mit dem Bewertungsfaktor werden die Werte folgender Maschinendaten beaufschlagt:

- MD36000 \$MA_STOP_LIMIT_COARSE
- MD36010 \$MA_STOP_LIMIT_FINE
- MD36030 \$MA_STANDSTILL_POS_TOL

Anwendungsbeispiele

- Anpassen des Positionierverhaltens an veränderte Massenverhältnisse, z. B. nach Getriebeumschaltung.
- Verkürzen der Positionierzeit in Abhängigkeit von verschiedenen Bearbeitungszuständen, z. B. Schruppen, Schlichten.

Parametrierbare Vorgabe der wirksamen Genauhaltbedingungen

Die Genauhaltbedingungen für die Befehle der 1. G-Gruppe können fest vorgegeben werden. Programmierte Genauhaltbedingungen werden dadurch unwirksam.

Die Genauhaltbedingungen können **unabhängig voneinander** für folgende Befehle eingestellt werden:

- Eilgang G0
- Alle anderen Befehle der 1. G-Gruppe

Das Einstellen der Genauhaltbedingung erfolgt kanalspezifisch über das nachfolgend dezimalcodierte Maschinendatum:

MD20550 \$MC_EXACT_POS_MODE = <Z><E>

Z	E	Wirksame Genauhaltbedingung
0	0	Programmierte Genauhaltbedingung
1	1	G601 (Genauhaltfenster fein)
2	2	G602 (Genauhaltfenster grob)
3	3	G603 (Interpolator-Ende)
E (Einerstelle): Einstellen der Genauhaltbedingung für Eilgang.		
Z (Zehnerstelle): Einstellen der Genauhaltbedingung für alle anderen Befehle der 1. G-Gruppe.		

Beispiel

MD20550 \$MC_EXACT_POS_MODE = 02

- <E> = 2: Beim Verfahren mit Eilgang wirkt unabhängig von der Programmierung immer Genauhaltbedingung G602 (Genauhaltfenster grob).
- <Z> = 0: Beim Verfahren mit allen anderen Befehlen der 1. G-Gruppe wirkt die programmierte Genauhaltbedingung.

Satzwechselverhalten bei Satzübergängen von G0 ↔ Nicht-G0 im Bahnsteuerbetrieb

Bei Bahnsteuerbetrieb kann das Satzwechselverhalten zwischen Eilgang- und Nicht-Eilgangsätzen (G0 ↔ Nicht-G0) über das nachfolgende Maschinendatum eingestellt werden:

MD20552 \$MC_EXACT_POS_MODE_G0_TO_G1 = <Wert>

<Wert>	Bedeutung
0	Kein zusätzlicher Stopp am Satzübergang.
1	Stopp am Satzübergang. Das Verhalten entspricht G601 (Genauhaltfenster fein)
2	Stopp am Satzübergang. Das Verhalten entspricht G602 (Genauhaltfenster grob)
3	Stopp am Satzübergang. Das Verhalten entspricht G603 (Interpolator-Ende)
4	Kein Stopp am Satzübergang. Bei Bahnsteuerbetrieb wird bei Satzwechseln von G0 → Nicht-G0 im G0-Satz vorausschauend der aktuelle Wert der Vorschubkorrektur des nachfolgenden Nicht-G0-Satzes berücksichtigt. Abhängig von der Achsdynamik und der Bahnlänge des aktuellen Satzes erfolgt der Satzwechsel mit der exakten bzw. bestmöglich angepassten Geschwindigkeit des Folgesatzes.
5	Kein Stopp am Satzübergang. Bei Bahnsteuerbetrieb wird bei Satzwechseln von G0 → Nicht-G0 und Nicht-G0 → G0 vorausschauend der aktuelle Wert der Vorschubkorrektur (G0 → Nicht-G0) bzw. der Eilgangkorrektur (Nicht-G0 → G0) des nachfolgenden Satzes berücksichtigt. Abhängig von der Achsdynamik und der Bahnlänge des aktuellen Satzes erfolgt der Satzwechsel mit der exakten bzw. bestmöglich angepassten Geschwindigkeit des Folgesatzes.

5.3 Bahnsteuerbetrieb

5.3.1 Allgemeine Funktionalität

Bahnsteuerbetrieb

Im Bahnsteuerbetrieb wird die Bahngeschwindigkeit am Satzende zum Satzwechsel **nicht** auf eine Geschwindigkeit abgebremst, die ein Erreichen des Genauhaltkriteriums ermöglicht. Ziel ist dagegen, ein größeres Abbremsen der Bahnachsen am Satzwechsellpunkt zu vermeiden, um mit möglichst gleicher Bahngeschwindigkeit in den nächsten Satz zu wechseln. Um dieses Ziel zu erreichen, wird mit Anwahl des Bahnsteuerbetriebs zusätzlich die Funktion "LookAhead" aktiviert.

Bahnsteuerbetrieb bewirkt, dass knickförmige Satzübergänge durch lokale Änderungen des programmierten Verlaufs tangential gestaltet bzw. geglättet werden. Das Ausmaß der Änderung relativ zum programmierten Verlauf kann durch die Kriterien des Überlastfaktors oder Überschleifens eingegrenzt werden.

Bahnsteuerbetrieb bewirkt:

- eine Verrundung der Kontur.
- kürzere Bearbeitungszeiten durch fehlende Brems- und Beschleunigungsvorgänge, die für das Erreichen des Genauhaltkriteriums benötigt werden.
- bessere Schnittbedingungen durch den gleichmäßigeren Geschwindigkeitsverlauf.

Bahnsteuerbetrieb ist sinnvoll, wenn:

- eine Kontur möglichst zügig abgefahren werden soll (z. B. mit Eilgang).
- der exakte Verlauf im Rahmen eines Fehlerkriteriums vom programmierten abweichen darf, um einen durchgehend stetigen Verlauf zu erzeugen.

Bahnsteuerbetrieb ist nicht sinnvoll, wenn eine Kontur exakt abgefahren werden soll.

Impliziter Genauhalt

In einigen Fällen muss im Bahnsteuerbetrieb ein Genauhalt erzeugt werden, um Folgeaktionen ausführen zu können. In diesen Situationen wird die Bahngeschwindigkeit auf Null abgebremst.

- Werden Hilfsfunktionen vor der Verfahrbewegung ausgegeben, so wird der vorhergehende Satz erst mit dem Erreichen des angewählten Genauhaltkriteriums beendet.
- Werden Hilfsfunktionen nach der Verfahrbewegung ausgegeben, so werden diese nach Interpolator-Ende des Satzes ausgegeben.
- Enthält ein ausführbarer Satz (z. B. Starten einer Positionierachse) keine Verfahrinformation für die Bahnachsen, so wird der vorhergehende Satz mit dem Erreichen des angewählten Genauhaltkriteriums beendet.
- Wird eine Positionierachse zur Geometrieachse deklariert, so wird mit Programmierung der Geometrieachse der vorhergehende Satz mit Interpolator-Ende abgeschlossen.

- Wird eine Synchronachse programmiert, die zuletzt als Positionierachse oder als Spindel programmiert war (Grundstellung der Zusatzachse ist Positionierachse), so wird der vorhergehende Satz mit Interpolator-Ende beendet.
- Wird die Transformation gewechselt, so wird der zuvor bearbeitete Satz mit dem aktiven Genauhaltkriterium abgeschlossen.
- Ein Satz wird mit Interpolator-Ende beendet, falls im Folgesatz die Umschaltung des Beschleunigungsprofils BRISK/SOFT erfolgt (siehe Funktionshandbuch *Achsen und Spindeln*, Kapitel *Beschleunigung*).
- Wird im Teileprogramm die Funktion "Zwischenspeicher leeren" programmiert, so wird der vorhergehende Satz mit dem Erreichen des angewählten Genauhaltkriteriums beendet.

Geschwindigkeit = 0 im Bahnsteuerbetrieb

Unabhängig vom impliziten Genauhalt wird die Bahnbewegung am Satzende auf Geschwindigkeit Null abgebremst, wenn:

- Positionierachsen mit der Anweisung `POS` programmiert sind und deren Verfahrzeit länger als die Verfahrzeit der Bahnachsen ist. Der Satzwechsel erfolgt mit Erreichen des "Genauhalt fein" der Positionierachse.
- die Zeit für die Positionierung einer Spindel, die mit der Anweisung `SPOS` programmiert wurde, länger dauert als die Verfahrzeit der Bahnachsen. Der Satzwechsel erfolgt mit Erreichen des "Genauhalt fein" der positionierenden Spindel.
- im aktuellen Satz Geometrieachsen verfahren und im folgenden Satz stattdessen Synchronachsen verfahren, bzw. im aktuellen Satz Synchronachsen verfahren und im nächsten Satz stattdessen Geometrieachsen verfahren werden.
- eine Synchronisation erforderlich wird.

5.3.2 Geschwindigkeitsabsenkung gemäß Überlastfaktor

Funktion

Die Funktion senkt im Bahnsteuerbetrieb die Bahngeschwindigkeit soweit ab, dass unter Wahrung der Beschleunigungsgrenze und unter Berücksichtigung eines Überlastfaktors der nichttangente Satzübergang in einem Interpolatortakt überfahren werden kann.

Mit der Geschwindigkeitsabsenkung werden bei nichttangente Konturverlauf am Satzübergang axiale Geschwindigkeitssprünge erzeugt. Diese werden auch von den mitfahrenden Synchronachsen nachvollzogen. Mit dem Geschwindigkeitssprung wird vermieden, dass die Bahngeschwindigkeit zu null abgesenkt wird. Der Sprung wird dann ausgeführt, wenn die axiale Geschwindigkeit mit der Achsbeschleunigung auf eine Geschwindigkeit reduziert wurde, ab der dann mit dem Sprung auf den neuen Sollwert gelangt werden kann. Die Sprunghöhe des Sollwerts kann mittels des Kriteriums Überlastfaktor begrenzt werden. Da die Sprunghöhe achsbezogen ist, wird am Satzübergang die kleinste Sprunghöhe der beim Satzwechsel aktiven Bahnachsen berücksichtigt.

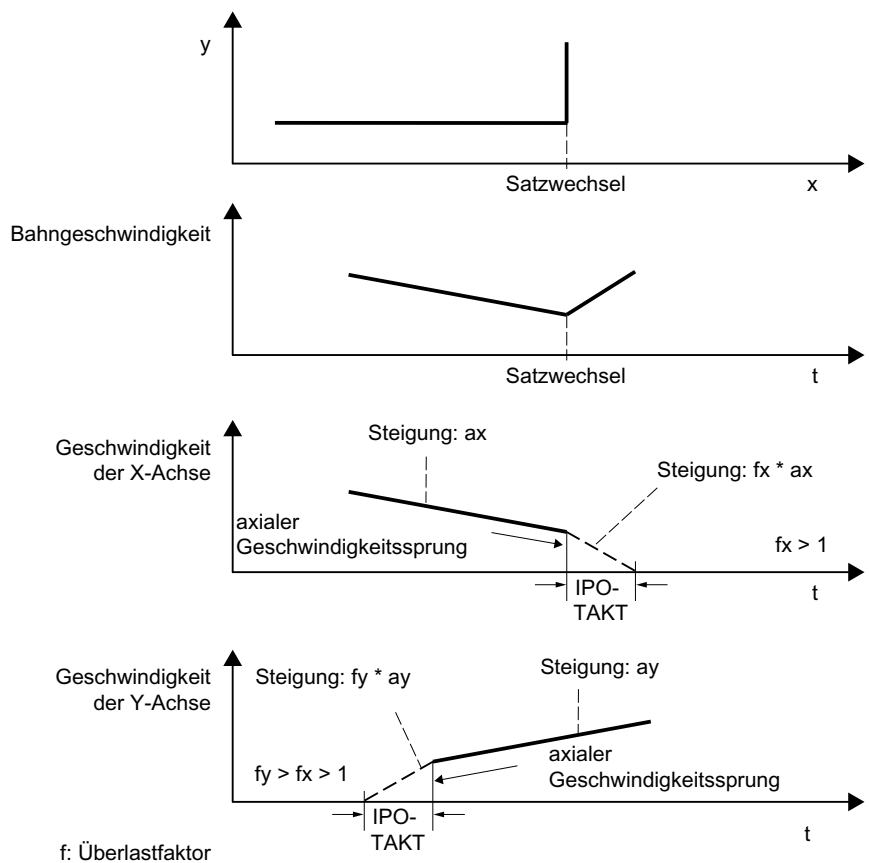


Bild 5-3 Axiale Geschwindigkeitsänderung am Satzübergang

Bei nahezu tangentialem Satzübergang wird die Bahngeschwindigkeit nicht abgesenkt, falls die zulässigen Achsbeschleunigungen nicht überschritten werden. Damit wird erreicht, dass sehr kleine Knicke in der Kontur (z. B. 0,5°) direkt überfahren werden.

Überlastfaktor

Der Überlastfaktor begrenzt den Geschwindigkeitssprung der Maschinenachse am Satzübergang. Damit der Geschwindigkeitssprung die Achsbelastbarkeit nicht überschreitet, wird der Sprung aus der Beschleunigung der Achse abgeleitet.

Der Überlastfaktor gibt an, um welches Maß die Beschleunigung der Maschinenachse (MD32300 \$MA_MAX_AX_ACCEL) für einen IPO-Takt überschritten werden darf.

Der Geschwindigkeitshub ergibt sich wie folgt:

$$\text{Geschwindigkeitshub} = \text{Achsbeschleunigung} * (\text{Überlastfaktor} - 1) * \text{Interpolator-Takt}$$

Der Überlastfaktor ist hinterlegt im Maschinendatum:

MD32310 \$MA_MAX_ACCEL_OVL_FACTOR (Überlastfaktor für axiale Geschwindigkeitssprünge)

Faktor 1.0 bedeutet, dass nur tangentielle Übergänge mit endlicher Geschwindigkeit überfahren werden können. Bei allen anderen Übergängen wird sollwertseitig auf Geschwindigkeit 0 abgebremst. Das Verhalten entspricht der Funktionalität "Genauhalt mit

Interpolator-Ende". Da dies für den Bahnsteuerbetrieb nicht erwünscht ist, ist der Faktor größer 1.0 einzustellen.

Hinweis

Für die Inbetriebnahme ist zu beachten, dass der Faktor dann herabzusetzen ist, wenn die Maschine bei knickförmigem Satzübergang zu Schwingungen angeregt wird und Überschleifen nicht verwendet werden soll.

Durch Setzen des folgenden Maschinendatums werden Satzübergänge unabhängig vom eingestelltem Überlastfaktor mit G641 / G642 überschleifen:

MD20490 \$MC_IGNORE_OVL_FACTOR_FOR_ADIS

Aktivierung / Deaktivierung

Der Bahnsteuerbetrieb mit Geschwindigkeitsabsenkung gemäß Überlastfaktor kann in jedem NC-Teileprogramm durch den modal wirksamen Befehl G64 aktiviert werden.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G9.

Der Bahnsteuerbetrieb G64 kann deaktiviert werden durch Anwahl von:

- Modalen Genauhalt G60
- Überschleifen G641, G642, G643, G644 oder G645

Impliziter Bahnsteuerbetrieb

Können im Bahnsteuerbetrieb mit Überschleifen G641 wegen sehr kurzer Weglängen der Sätze (z. B. nulltaktige Sätze) keine Überschleifsätze eingefügt werden, wird auf Bahnsteuerbetrieb G64 umgeschaltet.

5.3.3 Überschleifen

Funktion

Die Funktion "Überschleifen" fügt entlang einer programmierten Kontur (Bahnachsen) an nicht stetigen (knickförmigen) Satzübergängen Zwischensätze (Überschleifsätze) ein, sodass der sich daraus ergebende neue Satzübergang stetig (tangential) verläuft.

Synchronachsen

Überschleifen berücksichtigt neben den Geometrie- auch alle Synchronachsen. Allerdings kann bei parallelen Verfahren von Bahn- und Synchronachsen nicht für beide Achstypen gleichzeitig ein stetiger Satzübergang erzeugt werden. Für Synchronachsen wird dann zu Gunsten der Bahnachsen, die immer exakt verfahren werden, nur ein annähernd stetiger Satzübergang erzeugt.

Überschleifen bei G64

Überschleifen erfolgt auch, wenn zum Einhalten der Dynamikgrenzen am Satzübergang eine Geschwindigkeit erforderlich wird, welche bei G64 die zulässige Geschwindigkeit am Satzübergang unterschreitet. Siehe Kapitel "Geschwindigkeitsabsenkung gemäß Überlastfaktor (Seite 459)" Absatz "Überlastfaktor".

Auswirkung auf Synchronisationsbedingungen

Durch Überschleifen werden die programmierten Sätze, zwischen denen der oder die Überschleifsätze eingefügt werden, verkürzt. Die programmierte Satzgrenze verschwindet dabei und steht dann für etwaige Synchronisierungsbedingungen (z. B. Hilfsfunktionsausgabe parallel zur Bewegung, Stopp am Satzende) nicht mehr als Kriterium zur Verfügung.

Hinweis

Es wird empfohlen, bei Verwendung der Funktion "Überschleifen" Synchronisierungsbedingungen auf das Satzende des Satzes vor der Überschleifstelle zu beziehen, nicht auf das Ende des eingefügten Überschleifsatzes. Der Folgesatz wurde dann noch nicht begonnen und bei einem Stopp am Satzende kann die Kontur des Folgesatzes noch manuell geändert werden.

Ausnahmen

In folgenden Fällen erfolgt am Satzübergang, hier beispielhaft zwischen den Sätzen N10 nach N20, **kein** Überschleifen, d.h. es wird kein Überschleifsatz eingefügt:

Implizites Anhalten der Verfahrbewegung

Mögliche Ursachen:

- Hilfsfunktionsausgabe vor der Verfahrbewegung von N20 aktiv
- N20 enthält keine Verfahrbewegung von Bahnachsen
- In N20 verfährt zum ersten Mal eine Achse als Bahnachse, die zuvor Positionierachse war
- In N20 verfährt zum ersten Mal eine Achse als Positionierachse, die zuvor Bahnachse war
- In N10 werden Geometrieachsen verfahren, in N20 nicht
- In N20 werden Geometrieachsen verfahren, in N10 nicht
- Aktivierung von Gewindeschneiden G33 in N20
- Wechsel von BRISK und SOFT
- Transformationsrelevante Achsen sind nicht vollständig der Bahnbewegung zugeordnet (z. B. Pendelachsen, Positionierachsen).

Das Einfügen des Überschleifsatzes würde die Teileprogrammbearbeitung überproportional verlangsamen

Mögliche Ursachen:

- Ein Programm bzw. Programmabschnitt besteht aus einer Vielzahl sehr kurzer Verfahrssätze (≈ 1 Interpolatortakt / Verfahrssatz; da jeder Verfahrssatz mindestens einen Interpolatortakt benötigt, würden die eingefügten Zwischensätze die Bearbeitungszeit nahezu verdoppeln)
- G64 (Bahnsteuerbetrieb ohne Überschleifen) ohne Geschwindigkeitsreduzierung beim Satzwechsel aktiv
- Der parametrisierte Überlastfaktor (MD32310 \$MA_MAX_ACCEL_OVL_FACTOR) erlaubt das Verfahren der programmierten Kontur ohne dass die Bahngeschwindigkeit reduziert werden muss. Siehe dazu auch: MD20490 \$MC_IGNORE_OVL_FACTOR_FOR_ADIS

Bahnparameter verhindern das Überschleifen

Mögliche Ursachen:

- G641 (Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium) ist aktiv, aber Eilgang ist aktiv (G0) UND ADISPOS == 0 (Überschleifabstand bei G0)
- G641 (Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium) ist aktiv, aber Eilgang ist **nicht** aktiv UND ADIS == 0 (Überschleifabstand für Bahnfunktionen G1, G2, G3, ...)
- G642 oder G643 (Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen) ist aktiv, aber alle Toleranzen == Null

N10 oder N20 enthält keine Verfahrbewegung (Nullsatz)

Normalerweise werden keine Nullsätze erzeugt. Ausnahmen:

- Aktive Synchronaktion
- Programmsprünge

Auswirkung auf Synchronisationsbedingungen

Beim Überschleifen werden die programmierten Sätze, zwischen denen die Überschleifkontur eingefügt wird, verkürzt. Die ursprünglich programmierte Satzgrenze verschwindet dabei und steht dann für etwaige Synchronisierungsbedingungen (z. B. Hilfsfunktionsausgabe parallel zur Bewegung, Stopp am Satzende) nicht mehr zur Verfügung.

Hinweis

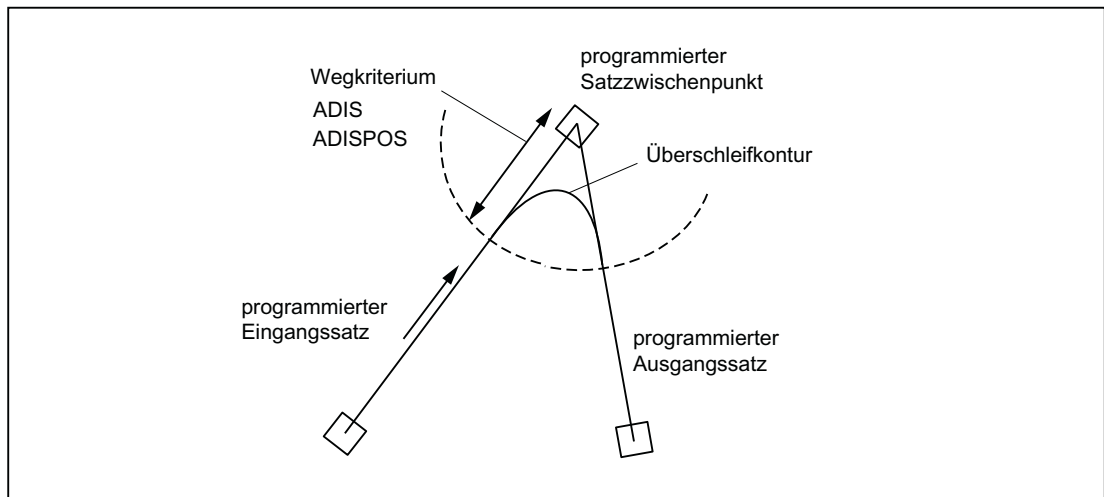
Es wird empfohlen, bei Verwendung der Funktion "Überschleifen" Synchronisierungsbedingungen auf das Satzende des Satzes vor der Überschleifstelle zu beziehen, nicht auf das Ende des eingefügten Überschleifsatzes. Der Folgesatz wurde dann noch nicht begonnen und bei einem Stopp auf Satzende kann die Kontur des Folgesatzes noch geändert werden.

5.3.3.1 Überschleifen nach Wegkriterium (G641)

Funktion

Beim Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium wird die Größe des Verrundungsbereichs durch die Wegkriterien ADIS und ADISPOS beeinflusst.

Die Wegkriterien ADIS und ADISPOS beschreiben die Strecke, die der Überschleifsatz vor dem Satzende frühestens beginnen darf, bzw. die Strecke nach Satzende, in der der Überschleifsatz beendet sein muss.



Hinweis

Spitze Ecken erzeugen stark gekrümmte Überschleifkurven und haben damit auch eine entsprechende Geschwindigkeitsreduzierung zur Folge.

Hinweis

ADISPOS wird wie ADIS gehandhabt, ist aber speziell nur für die Bewegungsart Eilgang G0 einzusetzen.

Wirksamkeit des Wegkriteriums

- ADIS bzw. ADISPOS müssen programmiert werden. Ist die Voreinstellung "Null" verhält sich G641 wie G64.
- Sind nicht beide aufeinanderfolgende Sätze Eilgang G0, so gilt der kleinere Überschleifabstand.

- Wird ein sehr kleiner Wert für ADIS verwendet, so ist zu beachten, dass die Steuerung sicherstellt, dass jeder interpolierte Satz - auch ein Überschleifzwischenatz - mindestens einen Interpolationspunkt enthält. Die maximale Bahngeschwindigkeit wird damit auf ADIS/ Interpolatortakt begrenzt.
- Unabhängig von ADIS und ADISPOS wird der Überschleifbereich durch die Satz-Weglänge begrenzt.
Bei Sätzen mit kurzen Wegstrecken (Strecke $< 4 * ADIS$ bzw. $4 * ADISPOS$) wird der Überschleifabstand reduziert, damit ein abfahrbarer Teil des ursprünglichen Satzes noch erhalten bleibt. Die verbleibende Länge ist vom Achsverlauf abhängig und beträgt ca. 60% des Wegs, welcher in dem Satz noch abgefahren werden soll. ADIS bzw. ADISPOS werden somit auf die verbleibenden 40% des zu verfahrenen Wegs reduziert. Durch diesen Algorithmus wird weiterhin verhindert, dass bei einer sehr kleinen Konturänderung ein Überschleifsatz eingefügt wird. In diesem Fall wird solange auf Bahnsteuerbetrieb G64 umgeschaltet, bis Überschleifsätze wieder eingefügt werden können.

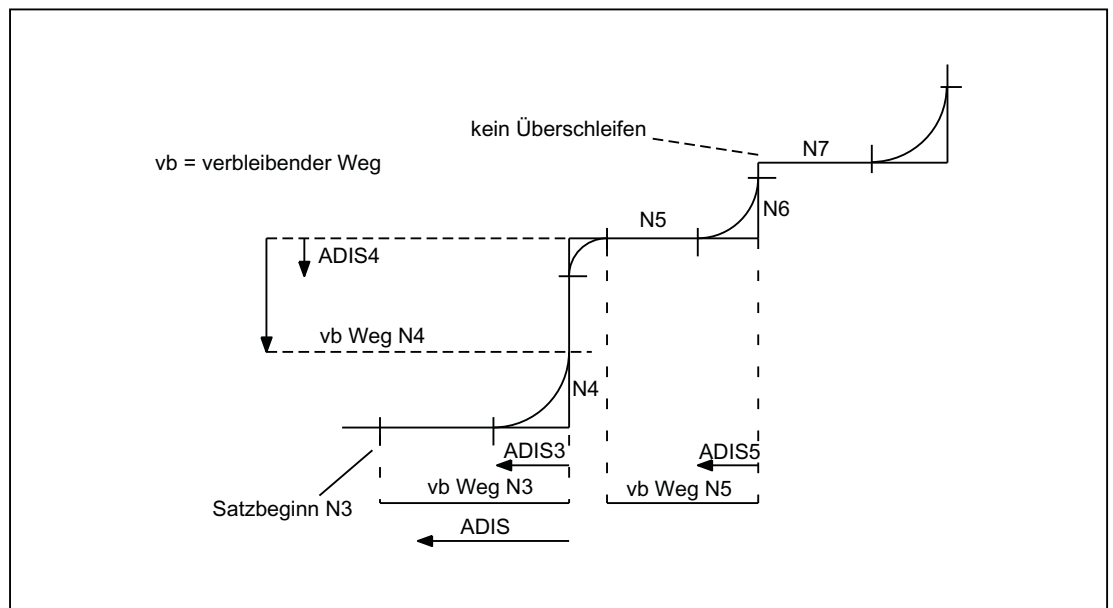


Bild 5-4 Bahnverlauf mit Begrenzung von ADIS

Aktivierung / Deaktivierung

Der Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium kann in jedem NC-Teileprogrammsatz durch den modal wirksamen Befehl G641 aktiviert werden. Vor oder mit der Aktivierung sollten die Wegkriterien ADIS/ADISPOS angegeben werden.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G9.

Der Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium (G641) kann deaktiviert werden durch Anwahl von:

- Modalen Genauhalt (G60)
- Bahnsteuerbetrieb G64, G642, G643, G644 oder G645

Programmbeispiel

Programmcode	Kommentar
N1 G641 Y50 F10 ADIS=0.5	; Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium aktivieren (Überschleifabstand: 0,5 mm).
N2 X50	
N3 X50.7	
N4 Y50.7	
N5 Y51.4	
N6 Y51.0	
N7 X52.1	

5.3.3.2 Überschleifen unter Einhaltung definierter Toleranzen (G642/G643)

Funktion

Beim Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen erfolgt das Überschleifen im Normalfall unter Einhaltung der maximal erlaubten Bahnabweichung. Anstelle dieser achsspezifischen Toleranz kann aber auch die Einhaltung der maximalen Konturabweichung (Konturtoleranz) oder der maximalen Winkelabweichung der Werkzeugorientierung (Orientierungstoleranz) konfiguriert werden.

Aktivierung

Der Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen kann in jedem NC-Teileprogrammsatz durch den modal wirksamen Befehl G642 bzw. G643 aktiviert werden.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G9.

Der Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen (G642/G643) kann deaktiviert werden durch Anwahl von:

- Modalen Genauhalt (G60)
- Bahnsteuerbetrieb G64, G641, G644 oder G645

Unterschiede G642 - G643

Die Funktionen G642 und G643 weisen im Überschleifverhalten folgende Unterschiede auf:

G642	G643
Bei G642 wird der Überschleifweg aus dem kürzesten Überschleifweg aller Achsen bestimmt. Dieser Wert wird bei der Erzeugung eines Überschleifsatzes berücksichtigt.	Bei G643 kann der Überschleifweg jeder Achse unterschiedlich sein. Die Überschleifwege werden achsspezifisch satzintern berücksichtigt (⇒ kein eigener Überschleifsatz).
Bei G642 bestimmt sich der Überschleifbereich aus der kleinsten Toleranzvorgabe.	Sehr unterschiedliche Vorgaben für die Konturtoleranz und die Orientierungstoleranz können sich nur bei G643 auswirken.

Parametrierung

Maximale Bahnabweichung

Die beim Überschleifen mit G642/G643 maximal erlaubte Bahnabweichung wird für jede Achse eingestellt im Maschinendatum:

MD33100 \$MA_COMPRESS_POS_TOL

Konturtoleranz und Orientierungstoleranz

Die Konturtoleranz und die Orientierungstoleranz werden eingestellt in den kanalspezifischen Settingdaten:

SD42465 \$SC_SMOOTH_CONTUR_TOL (Maximale Konturabweichung)

SD42466 \$SC_SMOOTH_ORI_TOL (Maximale Winkelabweichung der Werkzeugorientierung)

Die Settingdaten können im NC-Programm programmiert und dadurch für jeden Satzübergang anders vorgegeben werden.

Hinweis

Das Settingdatum SD42466 \$SC_SMOOTH_ORI_TOL ist nur bei aktiver Orientierungstransformation wirksam.

Überschleifverhalten

Das Überschleifverhalten mit G642 und G643 wird konfiguriert über das Maschinendatum:

MD20480 \$MC_SMOOTHING_MODE (Verhalten des Überschleifens mit G64x)

Die Einerstellen (**E**) definieren das Verhalten bei G643, die Zehnerstellen (**Z**) das Verhalten bei G642:

Wert E bzw. Z	Bedeutung
0	Alle Achsen: Überschleifen unter Einhaltung der maximal erlaubten Bahnabweichung: MD33100 \$MA_COMPRESS_POS_TOL
1	Geometrieachsen: Überschleifen unter Einhaltung der Konturtoleranz: SD42465 \$SC_SMOOTH_CONTUR_TOL Restliche Achsen: Überschleifen unter Einhaltung der maximal erlaubten Bahnabweichung: MD33100 \$MA_COMPRESS_POS_TOL
2	Geometrieachsen: Überschleifen unter Einhaltung der Orientierungstoleranz: SD42466 \$SC_SMOOTH_ORI_TOL Restliche Achsen: Überschleifen unter Einhaltung der maximal erlaubten Bahnabweichung: MD33100 \$MA_COMPRESS_POS_TOL

Wert E bzw. Z	Bedeutung
3	<p>Geometrieachsen: Überschleifen unter Einhaltung der Konturtoleranz und Orientierungstoleranz: SD42465 \$SC_SMOOTH_CONTUR_TOL SD42466 \$SC_SMOOTH_ORI_TOL</p> <p>Restliche Achsen: Überschleifen unter Einhaltung der maximal erlaubten Bahnabweichung: MD33100 \$MA_COMPRESS_POS_TOL</p>
4	<p>Alle Achsen: Es wird die mit ADIS bzw. ADISPOS programmierte Überschleiflänge verwendet (wie bei G641). Die Vorgaben der achsspezifischen Toleranz bzw. der Kontur- und Orientierungstoleranz werden ignoriert.</p>

Profil für die Grenzgeschwindigkeit

Die Benutzung eines Geschwindigkeitsprofils beim Überschleifen unter Einhaltung definierter Toleranzen wird über die Hunderterstelle von MD20480 gesteuert:

Wert	Bedeutung
< 100:	<p>Innerhalb des Überschleifbereichs wird ein Profil der Grenzgeschwindigkeit berechnet, wie es sich aus den vorgegebenen maximalen Werten für Beschleunigung und Ruck der beteiligten Achsen bzw. der Bahn ergibt. Dies kann zu einem Ansteigen der Bahngeschwindigkeit in dem Überschleifbereich führen, und damit zu einem Beschleunigen der beteiligten Achsen.</p>
≥100:	<p>Für Überschleifsätze mit G641/G642 wird kein Profil der Grenzgeschwindigkeit berechnet. Es wird nur eine konstante Grenzgeschwindigkeit festgelegt. Damit wird verhindert, dass beim Überschleifen mit G641/G642 die beteiligten Achsen im Überschleifbereich eventuell beschleunigt werden. Diese Einstellung kann jedoch unter Umständen, insbesondere bei großen Überschleifbereichen, dazu führen, dass in Überschleifsätzen mit zu kleiner Geschwindigkeit gefahren wird.</p>
1xx:	Kein Geschwindigkeitsprofil für G641
2xx:	Kein Geschwindigkeitsprofil für G642

Hinweis

Siehe auch MD28530 \$MC_MM_PATH_VELO_SEGMENTS (Anzahl der Speicherelemente z. Begrenzung der Bahngeschwindigkeit)

Randbedingungen

Einschränkung für Schutzbereiche bei aktiver Radius-Korrektur und einer Werkzeugorientierung:

Eine Werkzeugorientierung, die nicht senkrecht zu einer der drei Grundebenen des Basis-Koordinatensystems steht, berücksichtigt zwar die Radius-Korrektur, die Schutzbereiche werden jedoch nicht in die entsprechende Ebene gedreht.

Für G643 hat unbedingt zu gelten:

MD28530 \$MC_MM_PATH_VELO_SEGMENTS > 0 (Anzahl der Speicherelemente zur Begrenzung der Bahngeschwindigkeit)

Ist diese Bedingung erfüllt, so muss für alle Achsen gelten:

MD35240 \$MC_ACCEL_TYPE_DRIVE = FALSE (Beschleunigungskennlinie DRIVE für Achsen Ein / Aus)

5.3.3.3 Überschleifen mit maximal möglicher Achsdynamik (G644)

Funktion

Bei diesem Modus des Bahnsteuerbetriebs mit Überschleifen steht die maximal mögliche Dynamik der Achsen im Vordergrund.

Hinweis

Das Überschleifen mit G644 ist nur möglich, wenn:

- alle beteiligten Achsen nur eine lineare Bewegung in den beiden betrachteten Sätzen enthalten.
- **keine** kinematische Transformation aktiv ist.

Falls eine beteiligte Achse ein Polynom enthält (Polynom programmiert, Spline aktiv, Kompressor aktiv) oder eine kinematische Transformation aktiv ist, wird der Satzübergang mit G642 überschliften.

Aktivierung

Der Bahnsteuerbetrieb mit Überschleifen mit maximal möglicher Achsdynamik kann in jedem NC-Teileprogrammsatz durch den modal wirksamen Befehl G644 aktiviert werden.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G9.

Der Bahnsteuerbetrieb mit Überschleifen mit maximal möglicher Achsdynamik (G644) kann deaktiviert werden durch Anwahl von:

- Modalen Genauhalt (G60)
- Bahnsteuerbetrieb G64, G641, G642, G643 oder G645

Parametrierung

Das Überschleifverhalten mit G644 wird konfiguriert über die Tausender- und Zehntausenderstelle im Maschinendatum:

MD20480 \$MC_SMOOTHING_MODE (Verhalten des Überschleifens mit G64x)

Wert	Bedeutung
Tausenderstelle:	
0xxx:	Beim Überschleifen mit G644 werden die mit dem folgenden Maschinendatum angegebenen maximalen Abweichungen jeder Achse eingehalten: MD33100 \$MA_COMPRESS_POS_TOL Falls die Dynamik der Achse es zulässt, wird dabei evtl. die vorgegebene Toleranz nicht ausgenutzt.
1xxx:	Vorgabe des maximalen Überschleifwegs durch Programmierung von ADIS=... bzw. ADISPOS=... (wie bei G641).
2xxx:	Vorgabe der maximal auftretenden Frequenzen jeder Achse im Überschleifbereich mit dem Maschinendatum: MD32440 \$MA_LOOKAH_FREQUENCY (Glättungsfrequenz bei Look Ahead) Der Überschleifbereich wird so festgelegt, dass bei der Überschleifbewegung keine Frequenzen auftreten, welche die vorgegebene maximale Frequenz überschreiten.
3xxx:	Jede Achse, die einen Geschwindigkeitssprung an einer Ecke hat, fährt mit maximal möglicher Dynamik (maximale Beschleunigung und maximaler Ruck) um die Ecke. SOFT: Wenn SOFT aktiv ist, wird die maximale Beschleunigung und der maximale Ruck jeder Achse eingehalten. BRISK: Wenn BRISK aktiv ist, wird nur die maximale Beschleunigung eingehalten, der Ruck wird nicht begrenzt. Bei dieser Einstellung werden also weder die maximal auftretenden Abweichungen noch der maximale Überschleifweg kontrolliert. Die sich ergebenden Abweichungen bzw. Überschleifwege bestimmen sich allein aus den dynamischen Grenzwerten der jeweiligen Achse und der aktuellen Bahngeschwindigkeit.
4xxx:	Wie bei 0xxx werden die mit dem folgenden Maschinendatum angegebenen maximalen Abweichungen jeder Achse eingehalten: MD33100 \$MA_COMPRESS_POS_TOL Im Unterschied zu 0xxx wird die vorgegebene Toleranz nach Möglichkeit auch ausgenutzt. Die Achse erreicht daher nicht ihre maximal mögliche Dynamik.
5xxx:	Wie bei 1xxx wird der maximale Überschleifweg durch Programmierung von ADIS=... bzw. ADISPOS=... vorgegeben. Im Unterschied zu 1xxx wird der vorgegebene Überschleifweg nach Möglichkeit auch ausgenutzt. Die beteiligten Achsen erreichen dann eventuell nicht ihre maximal mögliche Dynamik.
Zehntausenderstelle	
0xxxx	Die Geschwindigkeitsprofile der Achsen werden im Überschleifbereich bei BRISK ohne Ruckbegrenzung und bei SOFT mit Ruckbegrenzung bestimmt.
1xxxx	Die Geschwindigkeitsprofile der Achsen werden im Überschleifbereich immer mit Ruckbegrenzung bestimmt, unabhängig davon ob BRISK oder SOFT aktiv ist.

Bei der Vorgabe der maximalen axialen Abweichungen (MD33100 \$MA_COMPRESS_POS_TOL) bzw. des maximalen Überschleifabstandes (ADIS/ADISPOS) wird der zur Verfügung stehende Überschleifweg normalerweise nicht ausgenutzt, wenn es die Dynamik der beteiligten Achsen erlauben. Dadurch hängt die Länge des Überschleifwegs vom aktiven Bahnvorschub ab. Bei kleineren Bahngeschwindigkeiten erhält

man geringere Abweichungen von der programmierten Kontur. Es kann jedoch eingestellt werden, dass in diesen Fällen nach Möglichkeit die vorgegebenen maximalen axialen Abweichungen bzw. der vorgegebene Überschleifabstand ausgenutzt wird. In diesem Fall sind dann die Abweichungen von der programmierten Kontur unabhängig vom programmierten Bahnvorschub.

Hinweis

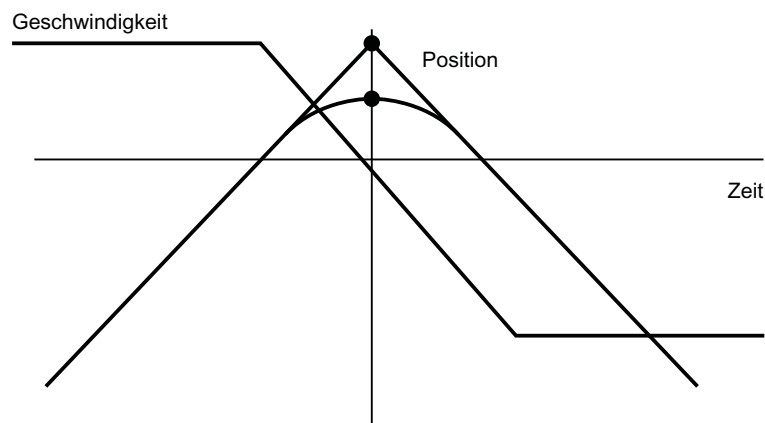
Außer den bisher genannten kann zusätzlich folgende Begrenzung wirksam werden:

Der Überschleifabstand kann maximal die Hälfte der Länge der beteiligten Originalsätze betragen.

Ruckbegrenzung

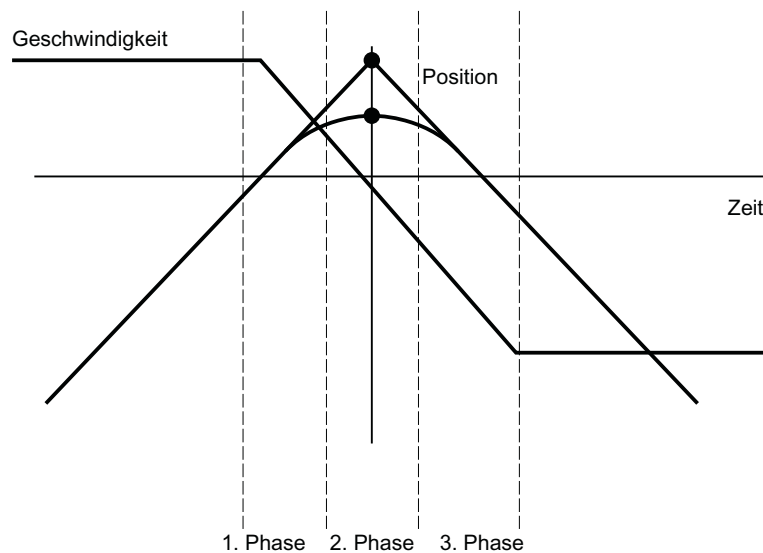
Die Glättung des Geschwindigkeitssprungs jeder Achse und damit die Form des Überschleifwegs hängt davon ab, ob eine Interpolation mit oder ohne Ruckbegrenzung durchgeführt wird.

Ohne Ruckbegrenzung erreicht die Beschleunigung jeder Achse im gesamten Überschleifbereich ihren Maximalwert:



Mit Ruckbegrenzung wird der Ruck jeder Achse im Überschleifbereich auf ihren jeweiligen Maximalwert begrenzt. Demzufolge besteht die Überschleifbewegung im Allgemeinen aus 3 Phasen:

- **1. Phase**
In der 1. Phase wird die maximale Beschleunigung jeder Achse aufgebaut. Dabei ist der Ruck konstant und gleich dem maximal möglichen Ruck der jeweiligen Achse.
- **2. Phase**
Die 2. Phase wird mit der maximal erlaubten Beschleunigung durchfahren.
- **3. Phase**
In der 3. Phase schließlich wird die Beschleunigung jeder Achse mit dem maximal erlaubten Ruck wieder auf Null abgebaut.



5.3.3.4 Überschleifen tangentialer Satzübergänge (G645)

Funktion

Bei diesem Modus des Bahnsteuerbetriebs mit Überschleifen werden auch bei tangentialen Satzübergängen Überschleifsätze gebildet, wenn der Krümmungsverlauf der Originalkontur in mindestens einer Achse einen Sprung aufweist.

Die Überschleifbewegung wird hierbei so festgelegt, dass alle beteiligten Achsen keinen Sprung in der Beschleunigung erfahren und die parametrisierten maximalen Abweichungen zur Originalkontur (MD33120 \$MA_PATH_TRANS_POS_TOL) nicht überschritten werden.

Bei knickförmigen, nicht-tangentialen Satzübergängen ist das Überschleifverhalten wie bei G642 (siehe Kapitel "Überschleifen unter Einhaltung definierter Toleranzen (G642/G643) (Seite 466)").

Aktivierung / Deaktivierung

Der Bahnsteuerbetrieb mit Überschleifen tangentialer Satzübergänge kann in jedem NC-Teileprogrammsatz durch den modal wirksamen Befehl G645 aktiviert werden.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G9.

Der Bahnsteuerbetrieb mit Überschleifen tangentialer Satzübergänge (G645) kann deaktiviert werden durch Anwahl von:

- Modalen Genauhalt (G60)
- Bahnsteuerbetrieb G64, G641, G642, G643 oder G644

Vergleich G642 - G645

Beim Überschleifen mit G642 werden nur Satzübergänge überschleifen, die eine Ecke darstellen, d. h. die Geschwindigkeit mindestens einer Achse hat einen Sprung. Ist ein Satzübergang jedoch tangential, aber die Krümmung macht einen Sprung, wird mit G642 kein Überschleifensatz eingefügt. Wird dieser Satzübergang mit endlicher Geschwindigkeit überfahren, so erfahren die Achsen einen mehr oder weniger großen Sprung der Beschleunigung, der (bei aktiver Ruckbegrenzung!) den parametrisierten Grenzwert (MD32432 \$MA_PATH_TRANS_JERK_LIM) nicht überschreiten darf. Abhängig von der Höhe des Grenzwerts wird die Bahngeschwindigkeit am Satzübergang dadurch evtl. stark verringert. Diese Einschränkung wird durch die Verwendung von G645 vermieden, da hier die Überschleifbewegung so festgelegt wird, dass keine Beschleunigungssprünge auftreten.

Parametrierung

Mit dem folgenden Maschinendatum wird für jede Achse die maximal erlaubte Bahnabweichung beim Überschleifen mit G645 angegeben:

MD33120 \$MA_PATH_TRANS_POS_TOL

Dieser Wert ist nur relevant für tangentiale Satzübergänge, die nicht beschleunigungsstetig sind. Beim Überschleifen von knickförmigen, nicht-tangentialen Satzübergängen wird (wie bei G642) der Toleranzwert aus MD33100 \$MA_COMPRESS_POS_TOL wirksam.

Siehe auch

Freiformflächenmodus: Grundfunktionen (Seite 499)

5.3.3.5 Überschleifen und Repositionieren (REPOS)

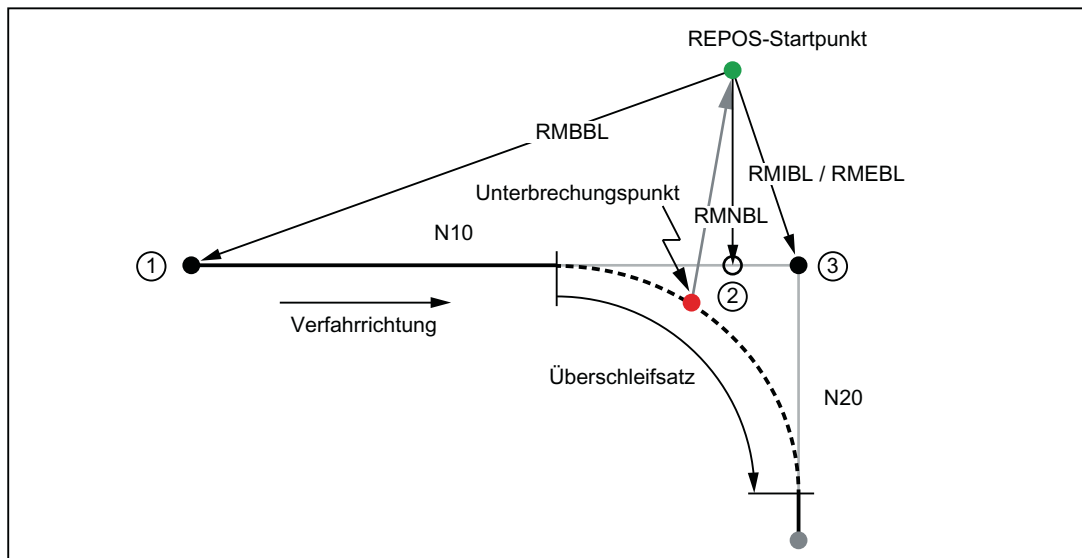
Wird eine Bearbeitung im Bereich der Überschleifkontur unterbrochen, kann durch einen REPOS-Vorgang **nicht** wieder direkt an die Überschleifkontur positioniert werden. In diesem Fall kann nur an die **programmierte** Kontur positioniert werden.

Beispiel

Programmiert: Zwei Verfahrsätze N10 und N20 mit programmiertem Überschleifen G641.

Im Überschleifbereich erfolgt eine Unterbrechung der Verfahrbewegung. Anschließend werden die Achsen z. B. manuell auf den REPOS-Startpunkt verfahren. Abhängig vom

gewählten REPOS-Mode erfolgt das Wiederanfahren an die Kontur an den Punkten ①, ② oder ③.



- RMBBL Wiederanfahren an den Anfang des unterbrochenen Verfahrssatzes
- RMIBL Wiederanfahren an die Unterbrechungsstelle
- RMEBL Wiederanfahren an das Ende des unterbrochenen Verfahrssatzes
- RMNBL Wiederanfahren an den nächstliegenden Konturpunkt
- ① Satzanfang N10
- ② Zum REPOS-Startpunkt nächstliegender Konturpunkt
- ③ Satzende N10 / Satzanfang N20

5.3.4 LookAhead

5.3.4.1 Standardfunktionalität

Funktion

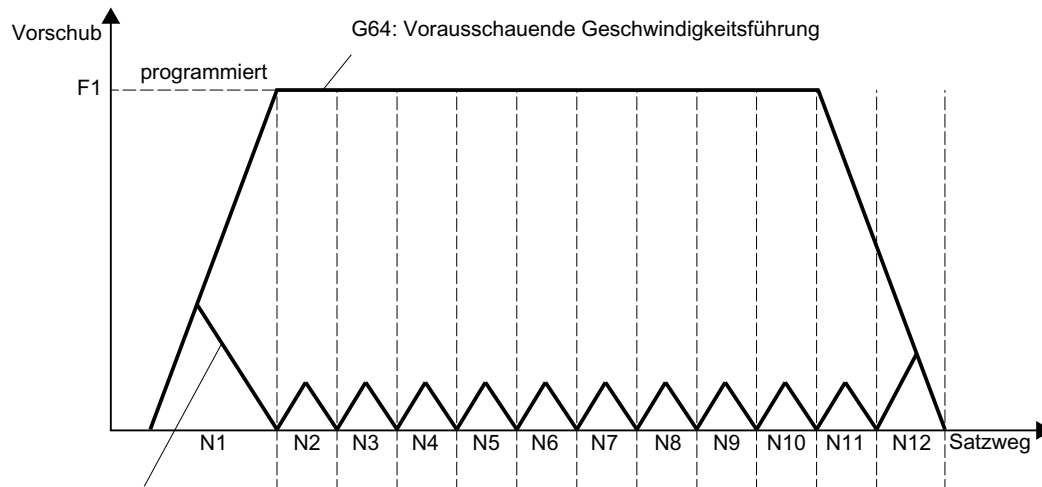
LookAhead ist eine im Bahnsteuerbetrieb (G64, G64x) aktive Funktion, die über den aktuellen Satz hinaus für mehrere NC-Teileprogrammätze eine vorausschauende Geschwindigkeitsführung ermittelt.

Hinweis

LookAhead ist nur für Bahnachsen verfügbar, **nicht** für Spindeln und Positionierachsen.

Beinhaltet ein Teileprogramm aufeinanderfolgende Sätze mit sehr kleinen Bahnwegen, dann wird ohne LookAhead pro Satz nur eine Geschwindigkeit erreicht, die zum Satzende ein Abbremsen der Achsen unter Wahrung der Beschleunigungsgrenzen ermöglicht. Dies bedeutet, dass die programmierte Geschwindigkeit überhaupt nicht erreicht wird. Mit

LookAhead ist es dagegen möglich, bei annähernd tangentialen Satzübergängen die Beschleunigungs- und Bremsphase über mehrere Sätze zu realisieren und somit bei kleinen Wegen einen höheren Vorschub zu erzielen.



G60: Phase konstanter Geschwindigkeit kann nicht erreicht werden

Bild 5-5 Geschwindigkeitsführung bei kurzen Wegstrecken und Genauhalt G60 bzw. Bahnsteuerbetrieb G64 mit LookAhead

Auf Geschwindigkeitsbeschränkungen wird vorausschauend so abgebremst, dass eine Verletzung der Beschleunigungs- und der Geschwindigkeitsgrenze vermieden wird.

LookAhead berücksichtigt planbare Geschwindigkeitsbeschränkungen wie:

- Genauhalt am Satzende
- Geschwindigkeitsbegrenzung im Satz
- Beschleunigungsbegrenzung im Satz
- Geschwindigkeitsbegrenzung am Satzübergang
- Synchronisieren mit Satzwechsel am Satzübergang

Funktionsweise

LookAhead analysiert satzbezogen die planbaren Geschwindigkeitsbeschränkungen und legt dementsprechend die benötigten Bremsrampenprofile fest. Die Vorausschau wird automatisch an Satzlänge, Bremsvermögen und zulässige Bahngeschwindigkeit angepasst.

Aus Sicherheitsgründen wird die Geschwindigkeit am Satzende des letzten vorbereiteten Satzes zunächst zu 0 angenommen, da der anschließende Satz sehr klein oder ein Genauhaltsatz sein könnte und die Achsen zum Satzende Stillstand erreicht haben sollen.

Bei einer Folge von Sätzen mit hoher Sollgeschwindigkeit und sehr kurzen Wegstrecken kann in den einzelnen Sätzen die Geschwindigkeit, abhängig vom aktuellen vorausgeschauten Geschwindigkeitswert, erhöht werden, um die geforderte Sollgeschwindigkeit zu erreichen und anschließend wieder reduziert werden, damit die Geschwindigkeit am Satzende des letzten vorausschauenden Folgesatzes 0 werden kann. Man erhält dadurch ein sägezahnförmiges Geschwindigkeitsprofil (siehe folgendes Bild), welches durch Herabsetzen der Sollgeschwindigkeit oder durch Vergrößern der vorausschauenden Satzanzahl vermieden werden kann.

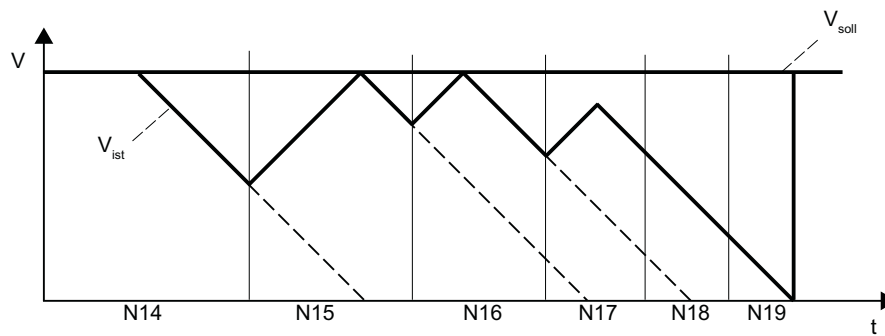


Bild 5-6 Beispiel für satzübergreifende Geschwindigkeitsführung (Zahl der vorausschauenden Sätze = 2)

Aktivierung / Deaktivierung

LookAhead wird aktiviert durch Anwahl des Bahnsteuerbetriebs G64, G641, G642, G643, G644 oder G645.

Eine Unterbrechung ist möglich durch Anwahl des satzweise wirksamen Genauhalts G09.

LookAhead wird deaktiviert durch Anwahl des modalen Genauhalt (G60).

Parametrierung

Satzanzahl

Um im Bahnsteuerbetrieb sicher zu fahren, muss der Vorschub über mehrere Sätze angepasst werden. Die Anzahl der vorausschauenden Sätze wird steuerungsintern automatisch ermittelt, wird optional aber über ein Maschinendatum begrenzt. Die Standardeinstellung ist "1" und bedeutet, dass LookAhead die Geschwindigkeitsführung nur für den Folgesatz berücksichtigt.

Da LookAhead vor allem bei (relativ zum Bremsweg) kurzen Sätzen wichtig ist, ist für das vorausschauende Bremsen die benötigte Satzanzahl von Interesse. Es ist ausreichend, eine Weglänge gleich dem Bremsweg zu betrachten, der nötig ist, um aus der maximalen Geschwindigkeit Stillstand zu erreichen.

Für eine Maschine mit einer niedrigen Achsbeschleunigung von $a = 1 \text{ m/s}^2$ und einem dazu hohen Vorschub von $v_{\text{Bahn}} = 10 \text{ m/min}$ erhält man mit einer erreichbaren Blockzykluszeit der Steuerung von $TB = 10 \text{ ms}$ eine für die Steuerung folgende Satzanzahl $n_{\text{LookAhead}}$ von:

$$n_{\text{LookAhead}} = \text{Bremsweg} / \text{Satzlänge} = (v_{\text{Bahn}}^2 / (2a)) / (v_{\text{Bahn}} * TB) = 9$$

Unter den angegebenen Bedingungen ist es nicht sinnvoll, den Vorschub über 10 Sätze anzupassen. Die angegebene Satzanzahl für die Vorausschau der LookAhead-Funktion verändert den LookAhead-Algorithmus und den Speicheraufwand nicht.

Da in einem Programm sehr oft die Bearbeitungsgeschwindigkeit kleiner der Maximalgeschwindigkeit ist, würde man mehr Sätze vorausschauen wie nötig ist, was

zusätzliche unnötige Rechenleistung beanspruchen würde. Daher wird die benötigte Satzanzahl aus der Geschwindigkeit abgeleitet, die sich aus folgender Multiplikation ergibt:

- Programmierte Geschwindigkeit * MD12100 \$MN_OVR_FACTOR_LIMIT_BIN
(bei Verwendung eines **binär**kodierten Vorschub-Override-Schalters)
- Programmierte Geschwindigkeit * MD12030 \$MN_OVR_FACTOR_FEEDRATE[30]
(bei Verwendung eines **gray**kodierten Vorschub-Override-Schalters)

Der Wert von MD12100 bzw. der 31. Override-Wert von MD12030 legt dabei die Dynamik-Reserven fest, die die Geschwindigkeitsführung für eine Überhöhung des Bahnvorschubs bereithält.

Hinweis

Der 31. Override-Wert von MD12030 sollte dem höchsten tatsächlich verwendeten Override-Faktor entsprechen.

Hinweis

Die vorausschauende Satzanzahl wird durch die mögliche Anzahl der NC-Sätze im IPO-Puffer begrenzt.

Geschwindigkeitsprofile

Neben den festen planbaren Geschwindigkeitsbeschränkungen kann LookAhead zusätzlich auch die programmierte Geschwindigkeit miteinbeziehen. Damit ist es möglich, über den aktuellen Satz hinaus vorausschauend die geringere Geschwindigkeit zu erreichen.

- **Ermittlung der Folgesatzgeschwindigkeit**
Ein mögliches Geschwindigkeitsprofil enthält die Ermittlung der Folgesatzgeschwindigkeit. Anhand von Informationen aus dem aktuellen und dem folgenden NC-Satz wird ein Geschwindigkeitsprofil berechnet, aus dem wiederum die erforderlichen Geschwindigkeitsreduzierungen für den aktuellen Override abgeleitet werden. Der ermittelte Maximalwert des Geschwindigkeitsprofils wird durch die maximale Bahngeschwindigkeit begrenzt. Mit der Funktion ist es möglich, Override-berücksichtigend im aktuellen Satz eine Geschwindigkeitsreduzierung zu starten, so dass zu Beginn des Folgesatzes dessen geringere Geschwindigkeit erreicht sein kann. Dauert die Reduzierung der Geschwindigkeit länger als die Verfahrzeit des aktuellen Satzes, so wird im Folgesatz die Geschwindigkeit weiter verringert. Die Geschwindigkeitsführung wird immer nur für den Folgesatz berücksichtigt.
Aktiviert wird die Funktion über das Maschinendatum:
MD20400 \$MC_LOOKAH_USE_VELO_NEXT_BLOCK

Wert	Bedeutung
TRUE	Funktion aktiv
FALSE	Funktion nicht aktiv

- **Festlegung von Override-Eckwerten**

Ist das Geschwindigkeitsprofil der Folgesatzgeschwindigkeit nicht ausreichend, weil z. B. sehr hohe Override-Werte (z. B. 200 %) verwendet werden bzw. konstante Schnittgeschwindigkeit G96/G961 aktiv ist und somit die Geschwindigkeit im Folgesatz immer noch reduziert werden muss, so bietet LookAhead eine Möglichkeit an, die programmierte Geschwindigkeit über mehrere NC-Sätze vorausschauend zu reduzieren: Mittels Festlegung von Override-Eckwerten berechnet sich LookAhead für jeden Eckwert ein begrenzendes Geschwindigkeitsprofil. Von diesen Profilen werden die erforderlichen Geschwindigkeitsreduzierungen für den aktuellen Override abgeleitet.

Der ermittelte Maximalwert des Geschwindigkeitsprofils wird durch die maximale Bahngeschwindigkeit begrenzt.

Der obere Eckwert sollte den Geschwindigkeitsbereich abdecken, der durch den Maximalwert des folgenden Maschinendatums erreicht wird:

MD12030 \$MN_OVR_FACTOR_FEEDRATE[n] (Bewertung des Bahnvorschub-Override-Schalters)

Er kann auch erreicht werden durch den Wert des Maschinendatums:

MD12100 \$MN_OVR_FACTOR_LIMIT_BIN (Begrenzung bei binärcodiertem Override-Schalter)

Damit kann eine Reduzierung der Geschwindigkeit in den Satz hinein, in dem sie programmiert ist, vermieden werden.

Sind auch schon bei 100 %-Override deutliche satzübergreifende Geschwindigkeitsreduzierungen erforderlich, so sollte auch im unteren Override-Bereich ein Eckwert gesetzt werden.

Die Anzahl der verwendeten Override-Eckwerte pro Kanal werden angegeben im Maschinendatum:

MD20430 \$MC_LOOKAH_NUM_OVR_POINTS (Anzahl der Override-Schalter-Eckwerte bei LookAhead)

Die dazugehörigen Eckwerte werden hinterlegt im Maschinendatum:

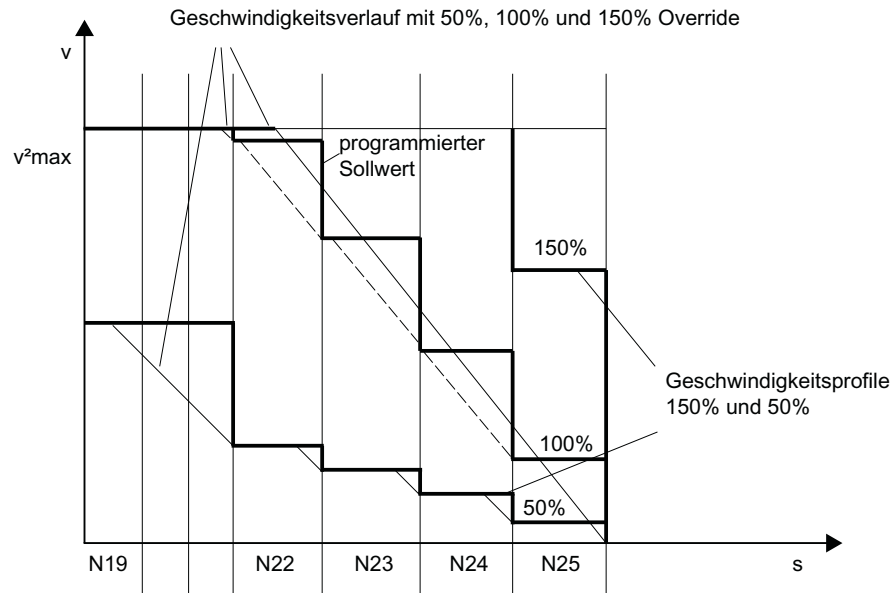
MD20440 \$MC_LOOKAH_OVR_POINTS (Override-Schalter-Eckwerte bei LookAhead)

Beispiel:

Begrenzende Geschwindigkeitsverläufe mit:

- Override = 50 %, 100 % bzw. 150 %
- vorausschauender Satzanzahl = 4
- MD20430 \$MC_LOOKAH_NUM_OVR_POINTS = 2
- MD20440 \$MC_LOOKAH_OVR_POINTS = 1.5, 0.5

– MD20400 \$MC_LOOKAH_USE_VELO_NEXT_BLOCK = 1



Eine Kombination beider Verfahren (Ermittlung der Folgesatzgeschwindigkeit und Festlegung von Override-Eckwerten) zur Ermittlung der Geschwindigkeitsprofile ist möglich und in der Regel auch sinnvoll, weil bereits mit den vorbesetzten Maschinendaten für diese Funktionen der größte Bereich der Override-abhängigen Geschwindigkeitsbeschränkungen abgedeckt ist.

Hinweis

Ist keines der beiden Verfahren aktiviert, so wird die Sollgeschwindigkeit immer erst im aktuellen Satz angefahren.

Hinweis

Planbare Geschwindigkeitsbeschränkungen begrenzen die Override-abhängigen Geschwindigkeitsbeschränkungen.

Entlastungsfaktor bei Blockzyklus-Problemen

Blockzyklusprobleme treten auf, wenn die Verfahrenlänge der abzuarbeitenden NC-Sätze so kurz ist, dass die LookAhead-Funktion die Maschinengeschwindigkeit reduzieren muss, um der Satzaufbereitung genügend Zeit zur Verfügung zu stellen. In dieser Situation kann ein ständiges Abbremsen und Beschleunigen der Bahnbewegung auftreten.

Derartige Geschwindigkeits-Schwankungen können durch Angabe eines Entlastungsfaktors gedämpft werden:

MD20450 \$MC_LOOKAH_RELIEVE_BLOCK_CYCLE (Entlastungsfaktor für die Blockzykluszeit)

Randbedingungen

Achsspezifischer Vorschub-Halt / Achssperre

5.3 Bahnsteuerbetrieb

Achsspezifischer Vorschub-Halt und achsspezifische Achssperre werden von LookAhead nicht berücksichtigt.

Soll eine Achse interpoliert werden, die aber andererseits per achsspezifischem Vorschub-Halt oder Achsen-Sperre stehenbleiben soll, so hält LookAhead die Bahnbewegung nicht vor dem betreffenden Satz an, sondern bremst **im** Satz ab.

Stört dieses Verhalten, so kann über die PLC ein **achsspezifischer** Vorschub-Halt auf eine **kanalspezifische** Vorschubsperre umgelegt und damit die Bahn sofort angehalten werden (siehe auch Funktionshandbuch *Überwachen und kompensieren*, Kapitel *Klemmungsüberwachung*).

5.3.4.2 Freiformflächenmodus: Erweiterungsfunktion

Funktion

Die Funktion "Freiformflächenmodus: Erweiterungsfunktion" ist eine Erweiterung der LookAhead-Standardfunktionalität und dient zur Berechnung des Bahngeschwindigkeitsprofils bei der Freiformflächenbearbeitung (siehe auch Kapitel "Freiformflächenmodus: Grundfunktionen (Seite 499)").

Ihr Einsatz optimiert den Bahnsteuerbetrieb wie folgt:

- Symmetrie zwischen Beschleunigungs- und Bremsprofil
- Gleichmäßiger Beschleunigungsvorgang auch bei sich ändernder Ruck- bzw. Beschleunigungsbegrenzung
- Gleichmäßiger Beschleunigungsvorgang von Sollgeschwindigkeitsprofilen unabhängig davon, inwieweit sie mit der vorgegebenen Dynamikbegrenzung anfahrbar sind oder nicht
- Vorausschauendes Bremsen auf niedrigere Sollgeschwindigkeiten

Die Gleichmäßigkeit und die Einhaltung der Dynamikbegrenzung gewährleisten, dass die Sollgeschwindigkeitsprofile zu einem homogenen Geschwindigkeitsprofil auf dem Teil geglättet werden. Damit wird sichergestellt, dass sich Schleppabstände möglichst gering auf die Oberflächengüte auswirken.

Vorteile

- Gleichmäßigere Oberfläche des Werkstücks
- Geringere Belastung der Maschine

Anwendungen

Die Funktion "Freiformflächenmodus: Erweiterungsfunktion" wird zur Bearbeitung von Werkstücken verwendet, die vorwiegend aus Freiformflächen bestehen.

Hinweis

Bei Standardbearbeitungen werden keine besseren Ergebnisse erzielt, weshalb in solchen Fällen die Standardfunktionalität von LookAhead verwendet werden sollte.

Wirksamkeit

Die Funktion ist nur unter folgenden Bedingungen wirksam:

- Betriebsart AUTOMATIK aktiv
- Beschleunigungsmodus "Beschleunigung mit Ruckbegrenzung (SOFT)" aktiv

Parametrierung

Arbeitsspeicher

Der Speicher für die Funktion "Freiformflächenmodus: Erweiterungsfunktion" wird konfiguriert durch das Maschinendatum:

MD28533 \$MC_MM_LOOKAH_FFORM_UNITS = <Wert>

Der benötigte Speicher ist abhängig vom Teileprogramm, den Satzlängen, der Achsdynamik sowie einer aktiven kinematischen Transformation.

Als Richtwert für die Freiformflächenbearbeitung gilt folgende Einstellung: MD28533 = 18

Hinweis

MD28533 sollte wegen dem zusätzlichen Speicherbedarf nur für die Kanäle gesetzt werden, in denen auch Freiformflächen bearbeitet werden.

Anzahl der NC-Sätze im IPO-Puffer

In der Regel empfiehlt es sich, für die Verwendung der Funktion "Freiformflächenmodus: Erweiterungsfunktion" die konfigurierte Anzahl der NC-Sätze im Interpolationspuffer deutlich zu erhöhen:

MD28060 \$MC_MM_IPO_BUFFER_SIZE > 100

Ein zu geringer Satzspeicher verringert ggf. die Gleichmäßigkeit des Bahngeschwindigkeitsprofils.

Aktivierung / Deaktivierung

Die Funktion kann für jeden Dynamikmodus (siehe Kapitel "Dynamikmodus für Bahninterpolation (Seite 496)") unabhängig ein- oder ausgeschaltet werden:

MD20443 \$MC_LOOKAH_FFORM[<n>]= <Wert>

Index <n>	Dynamikmodus	<Wert>	Freiformflächenmodus: Erweiterungsfunktion
0	Standard-Dynamikeinstellungen (DYNNORM)	0	aus
		1	ein
1	Positionierbetrieb, Gewindebohren (DYNPOS)	0	aus
		1	ein
2	Schruppen (DYNROUGH)	0	aus
		1	ein

5.3 Bahnsteuerbetrieb

Index <n>	Dynamikmodus	<Wert>	Freiformflächenmodus: Erweiterungsfunktion
3	Vorschlichten (DYNSEMIFIN)	0	aus
		1	ein
4	Schlichten (DYNFINISH)	0	aus
		1	ein
5	Feinschlichten (DYNPREC)	0	aus
		1	ein

Typischerweise ist die Funktion "Freiformflächenmodus: Erweiterungsfunktion" nur aktiv, wenn auch die Funktion "Freiformflächenmodus: Grundfunktionen" aktiv ist. Die Einstellungen in MD20443 \$MC_LOOKAH_FFORM[<n>] sollten daher mit den Einstellungen in MD20606 \$MC_PREPDYN_SMOOTHING_ON[<n>] übereinstimmen.

In den Dynamikmodi, in denen die Funktion "Freiformflächenmodus: Erweiterungsfunktion" ausgeschaltet ist, ist die LookAhead-Standardfunktionalität aktiv.

Programmierung

In der Regel wird die Funktion "Freiformflächenmodus: Erweiterungsfunktion" durch einen Wechsel des Dynamikmodus im Teileprogramm wirksam.

Beispiel

Folgende Parametrierung ist gegeben:

- MD20443 \$MC_LOOKAH_FFORM[0] = 0
- MD20443 \$MC_LOOKAH_FFORM[1] = 0
- MD20443 \$MC_LOOKAH_FFORM[2] = 1
- MD20443 \$MC_LOOKAH_FFORM[3] = 1
- MD20443 \$MC_LOOKAH_FFORM[4] = 1
- MD20443 \$MC_LOOKAH_FFORM[5] = 1

Wechsel des Dynamikmodus im Teileprogramm:

Programmcode	Kommentar
N10 DYNPOS	; Dynamikmodus DYNPOS einschalten. Im Dynamikmodus DYNPOS ist die LookAhead-Standardfunktionalität aktiv.
...	
N100 G17 G54 F10000	
N101 DYNFINISH	; Dynamikmodus DYNFINISH einschalten. Im Dynamikmodus DYNFINISH ist die Funktion "Freiformflächenmodus: Erweiterungsfunktionen" aktiv.
N102 SOFT G642	
N103 X-0.274 Y149.679 Z100.000 G0	
N104 COMPCAD	

Programmcode	Kommentar
...	
N1009 Z4.994 G01	
N10010 X.520 Y149.679 Z5.000	
N10011 X10.841 Y149.679 Z5.000	
N10012 X11.635 Y149.679 Z5.010	
N10013 X12.032 Y149.679 Z5.031	
M30	

Hinweis

Beim Wechsel zwischen der LookAhead-Standardfunktionalität und der Funktion "Freiformflächenmodus: Erweiterungsfunktion" bzw. umgekehrt wird der Bahnsteuerbetrieb durch einen Interpolator-Stopp unterbrochen.

Randbedingungen

Automatische Umschaltung

Folgende Funktionen bewirken eine automatische Umschaltung auf die LookAhead-Standardfunktionalität:

- Gewindegewindebohren (G33, G34, G35, G331, G332, G63)
- Bahn-Leitwertkopplung
- Stanzen, Nibbeln
- Kartesisches PTP-Fahren

Danach wird die Funktion "Freiformflächenmodus: Erweiterungsfunktion" automatisch wieder eingeschaltet.

Verwendung der Befehle der G-Gruppe 15 (Vorschubtypen)

Folgende Vorschubtypen sind in Verbindung mit der Funktion "Freiformflächenmodus: Erweiterungsfunktion" nicht zu empfehlen:

- Umdrehungsvorschub (G95, G96, G97, ...)
- Zeitreziproker Vorschub (G93)

Verwendung von FLIN

Das Vorschubprofil FLIN ist in Verbindung mit der Funktion "Freiformflächenmodus: Erweiterungsfunktion" nicht nutzbar.

Einfluss von Vorschubkorrekturen

Vorschubkorrekturen (über Maschinensteuertafel, \$AC_OVR, ...) können die Verfahrzeit gegenüber der LookAhead-Standardfunktionalität merklich verlängern.

Wechselwirkung mit Eilgangbewegung (G0)

In die Freiformflächenbearbeitung eingestreute G0-Sätze schalten die LookAhead-Funktionalität nicht um (von der Funktion "Freiformflächenmodus: Erweiterungsfunktion" auf

die LookAhead-Standardfunktionalität oder umgekehrt). D. h., obwohl mit G0 die Standard-Dynamikeinstellung (DYNNORM) wirksam ist, wird dadurch nicht automatisch auch die für DYNNORM voreingestellte LookAhead-Standardfunktionalität (→ MD20443 \$MC_LOOKAH_FFORM) wirksam. Durch das Beibehalten der aktuell aktiven LookAhead-Funktionalität wird ein homogeneres Geschwindigkeitsprofil erreicht, zumal in der Regel die G0- und Polynomsätze durch Überschleifen glatt verbunden sind.

5.4 Dynamikanpassungen

5.4.1 Glättung der Bahngeschwindigkeit

Einleitung

Die Geschwindigkeitsführung nutzt die vorgegebene Achsdynamik aus. Wenn der programmierte Vorschub nicht erreicht werden kann, wird die Bahngeschwindigkeit an den parametrisierten axialen Grenzwerten und den Grenzwerten der Bahn (Geschwindigkeit, Beschleunigung, Ruck) geführt. Damit kann es zu häufigen Brems- und Beschleunigungsvorgängen auf der Bahn kommen.

Findet bei einer Bearbeitung mit hoher Bahngeschwindigkeit ein kurzzeitiger Beschleunigungsvorgang statt, der nach sehr kurzer Zeit wieder zu einem Bremsvorgang führt, so wird die Bearbeitungszeit hierdurch nicht deutlich reduziert. Diese Beschleunigungsvorgänge können jedoch zu unerwünschten Erscheinungen führen, z. B. wenn Maschinenresonanzen angeregt werden.

Bei einigen Anwendungen im Formenbau, speziell beim Hochgeschwindigkeitsfräsen, ist dagegen eine gleichmäßige Bahngeschwindigkeit wünschenswert. In diesen Fällen kann es daher sinnvoll sein, zugunsten einer ruhigeren Bahngeschwindigkeit auf kurzzeitige Beschleunigungsvorgänge zu verzichten.

Funktion

Bei aktiver Funktion "Glättung der Bahngeschwindigkeit" ist zugunsten einer ruhigeren Bahngeschwindigkeitsführung ein Glättungsfaktor wirksam, der den maximal zulässigen Produktivitätsverlust bestimmt: Beschleunigungsvorgänge, die weniger als dieser Faktor zu einer kürzeren Programmlaufzeit beitragen, werden nicht durchgeführt. Betrachtet werden dabei nur Beschleunigungsvorgänge, deren Frequenzen oberhalb der parametrierbaren Grenzfrequenzen der beteiligten Achsen liegen.

Vorteile:

- Vermeidung von Anregungen möglicher Maschinenresonanzen aufgrund von ständigen, kurzzeitigen Brems- und Beschleunigungsvorgängen (im Bereich weniger IPO-Takte).
- Vermeidung ständig variierender Schnittgeschwindigkeiten durch Beschleunigungsvorgänge, die keinen großen Gewinn für die Programmlaufzeit bewirken.

Hinweis

Die Glättung der Bahngeschwindigkeit bewirkt keinen Konturfehler.

Schwankungen der Achsgeschwindigkeit aufgrund von Krümmungen in der Kontur bei konstanter Bahngeschwindigkeit können weiterhin auftreten und werden mit dieser Funktion nicht reduziert.

Schwankungen der Bahngeschwindigkeit aufgrund der Vorgabe eines neuen Vorschubs werden ebenfalls nicht verändert. Dies liegt in der Verantwortung des Teileprogrammerstellers.

Voraussetzungen

- Die Glättung der Bahngeschwindigkeit ist nur wirksam im Bahnsteuerbetrieb mit LookAhead über mehrere Sätze bei SOFT und BRISK. Die Glättung ist **nicht** wirksam bei GO.
- Die Taktzeiten der Steuerung müssen so parametrisiert sein, dass der Vorlauf genügend Sätze aufbereiten kann, um einen Beschleunigungsvorgang analysieren zu können.

Aktivierung / Deaktivierung

Die Funktion "Glättung der Bahngeschwindigkeit" wird aktiviert/deaktiviert mit dem Maschinendatum:

MD20460 \$MC_LOOKAH_SMOOTH_FACTOR (Glättungsfaktor bei LookAhead)

Wert	Bedeutung
0.0	Glättung der Bahngeschwindigkeit nicht aktiv (Standardeinstellung)
> 0	Glättung der Bahngeschwindigkeit aktiv

Eine Änderung der MD-Einstellung wird erst durch NEW CONF wirksam.

Parametrierung**Glättungsfaktor**

Der Glättungsfaktor wird eingestellt über das kanalspezifische Maschinendatum:

MD20460 \$MC_LOOKAH_SMOOTH_FACTOR (Glättungsfaktor bei LookAhead)

Der Prozentwert definiert, wieviel ein Bearbeitungsschritt ohne Beschleunigungen/Verzögerungen länger sein darf als der entsprechende Schritt mit Ausführung der Beschleunigungen/Verzögerungen.

Dies wäre ein "worst case"-Wert, wenn alle Beschleunigungsvorgänge innerhalb des Teileprogramms, außer der ersten Anfahrbewegung, weggeglättet werden würden. Die tatsächliche Verlängerung wird in jedem Fall kleiner sein, möglicherweise sogar 0, falls das Kriterium für keinen Beschleunigungsvorgang anspricht. Es können also durchaus Werte von 50 bis 100 % eingetragen werden, ohne eine deutliche Verlängerung der Bearbeitungszeit zu erhalten.

Berücksichtigung des programmierten Vorschubs

5.4 Dynamikanpassungen

Die Glättung der Bahngeschwindigkeit kann mit oder ohne Berücksichtigung des programmierten Vorschubs durchgeführt werden. Die Auswahl erfolgt über das Maschinendatum:

MD20462 \$MC_LOOKAH_SMOOTH_WITH_FEED (Bahnglättung mit programmiertem Vorschub)

Wert	Bedeutung
0	Programmierter Vorschub wird nicht berücksichtigt.
1	Programmierter Vorschub wird berücksichtigt (Standardeinstellung).

Bei Berücksichtigung des programmierten Vorschubs wird der vorgegebene Glättungsfaktor (siehe MD20460) besser eingehalten, wenn der Override auf 100 % steht.

Achsspezifische Grenzfrequenzen

Die achsspezifischen Grenzfrequenzen werden festgelegt über das Maschinendatum:

MD32440 \$MA_LOOKAH_FREQUENCY (Glättungsfrequenz bei LookAhead)

Beschleunigungs- und Bremsvorgänge, die mit einer höheren Frequenz ablaufen, werden abhängig von der Parametrierung der folgenden Maschinendaten geglättet oder in der Dynamik reduziert:

MD20460 \$MC_LOOKAH_SMOOTH_FACTOR (Glättungsfaktor bei LookAhead)

MD20465 \$MC_ADAPT_PATH_DYNAMIC (Adaption der Bahndynamik)

Weitere Informationen zu MD20465 siehe Kapitel "Anpassung der Bahndynamik (Seite 488)".

Hinweis

Werden Schwingungen in der Mechanik einer Achse angeregt und ist deren Frequenz bekannt, so sollte MD32440 kleiner als diese Frequenz eingestellt werden.

Die benötigten Resonanzfrequenzen können z. B. über die integrierten Messfunktionen ermittelt werden.

Funktionsweise

Aus den an der Bahn beteiligten Achsen wird intern das Minimum von MD32440 ermittelt (= f_{Bahn}). Für die Glättung werden nur Beschleunigungsvorgänge betrachtet, bei denen die Anfangs- oder Endgeschwindigkeit dieser Bewegung innerhalb folgender Zeit wieder erreicht wird:

$$t = t_2 - t_1 = 2 / f_{\text{Bahn}}$$

Auf diese Beschleunigungsvorgänge wird verzichtet, wenn die sich dadurch ergebende Verlängerung der Bearbeitungszeit den über den Glättungsfaktor (MD20460) vorgegebenen Grenzwert nicht überschreitet.

Beispiel

Folgende Parametrierung ist gegeben:

MD20460 \$MC_LOOKAH_SMOOTH_FACTOR = 10 %

MD32440 \$MA_LOOKAH_FREQUENCY[AX1] = 20 Hz

MD32440 \$MA_LOOKAH_FREQUENCY[AX2] = 20 Hz

MD32440 \$MA_LOOKAH_FREQUENCY[AX3] = 10 Hz

An der Bahn sind die 3 Achsen X = AX1, Y = AX2, Z = AX3 beteiligt.

Das Minimum vom MD32440 dieser 3 Achsen beträgt somit 10 Hz. Daraufhin werden Beschleunigungsvorgänge untersucht, die in einem Zeitraum $t_2 - t_1 = 2 / 10 \text{ Hz} = 200 \text{ ms}$ ablaufen. Die Zeit t_2 ist diejenige Zeit, zu der nach einem Beschleunigungsvorgang von der Geschwindigkeit v_1 ausgehend diese Geschwindigkeit v_1 wieder erreicht wird. Nur dieser Bereich wird auch für die Verlängerung der Bearbeitungszeit betrachtet.

Wenn der Zeitraum $t_2 - t_1$ größer als 200 ms ist oder wenn die zusätzliche Programmbearbeitungszeit $t_3 - t_2$ mehr als 10 % (= MD20460) von $t_2 - t_1$ beträgt, ergibt sich folgender Zeitverlauf:

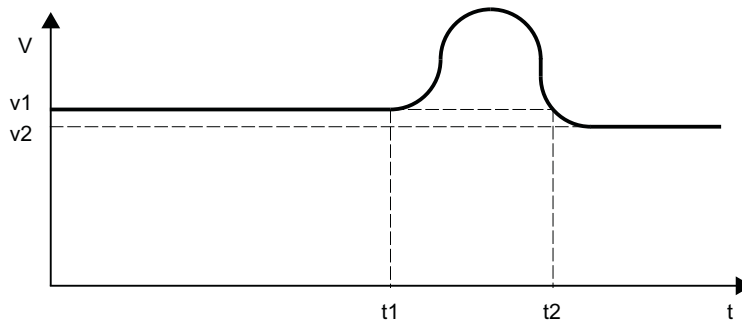


Bild 5-7 Verlauf der zeitoptimalen Bahngeschwindigkeit (ohne Glättung)

Wenn dagegen der Zeitraum $t_2 - t_1$ kleiner als 200 ms ist und wenn die zusätzliche Programmbearbeitungszeit $t_3 - t_2$ maximal 10 % von $t_2 - t_1$ beträgt, dann ergibt sich dieser Zeitverlauf:

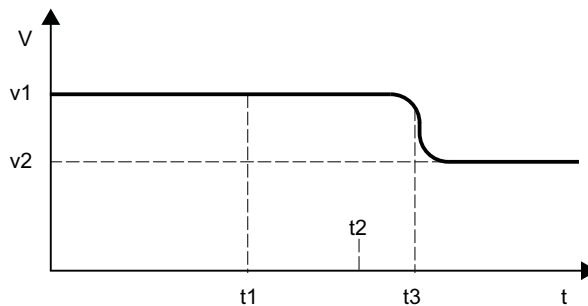


Bild 5-8 Verlauf der geglätteten Bahngeschwindigkeit

5.4.2 Anpassung der Bahndynamik

Funktion

Hochdynamische Beschleunigungs- und Bremsvorgänge während der Bearbeitung können zur Anregung von mechanischen Schwingungen von Maschinenelementen und in Folge zu einer Verminderung der Oberflächengüte des Werkstücks führen.

Mit der Funktion "Anpassung der Bahndynamik" kann die Dynamik der Beschleunigungs- und Bremsvorgänge an die Maschinengegebenheiten angepasst werden.

Hinweis

Die Funktion "Anpassung der Bahndynamik" betrachtet nicht die Brems- und Beschleunigungsvorgänge der einzelnen an der Bahn beteiligten Achsen, sondern immer der resultierenden Bahn. Daher kann es auch bei einem konstanten Bahngeschwindigkeitsverlauf durch unetstetige Konturverläufe oder kinematische Transformationen zu bezüglich der Anregung von mechanischen Schwingungen kritischen Brems- und Beschleunigungsvorgängen der Achsen kommen.

Wirksamkeit

Die Funktion "Anpassung der Bahndynamik" ist nur während Bahnbewegungen wirksam:

- Bahnsteuerbetrieb (G64, G64x)
Im Bahnsteuerbetrieb wird die optimale Wirkung der Dynamikanpassung bei einem wirksamen Override von 100% erreicht. Deutliche Abweichungen von diesem Wert oder Funktionen, durch die die Bahnachsen abgebremst werden müssen (z. B. Hilfsfunktionsausgaben an die PLC), vermindern die gewünschte Wirkung stark.
- Genauhalt (G60)

Unter folgenden Bedingungen ist die Funktion "Anpassung der Bahndynamik" auch bei Bahnbewegungen **nicht** wirksam:

- Programmierem Eilgang (G0)
- Veränderungen des Override-Werts
- Stoppanforderungen während der Bewegung (z. B. NC-Stop, NC-Reset)
- Funktion "Geschwindigkeitsabhängige Bahnbeschleunigung" (DRIVE) ist aktiv

Aktivierung / Deaktivierung

Die Funktion wird aktiviert/deaktiviert mit dem Maschinendatum:

MD20465 \$MC_ADAPT_PATH_DYNAMIC (Adaption der Bahndynamik)

Wert	Bedeutung
= 1.0	Dynamikanpassung nicht aktiv (Standardeinstellung)
> 1.0	Dynamikanpassung aktiv

Durch die Aktivierung wird im Bahnsteuerbetrieb intern **immer die Funktion "Glättung der Bahngeschwindigkeit" mitaktiviert** (siehe Kapitel "Glättung der Bahngeschwindigkeit (Seite 484)").

Falls der Glättungsfaktor (MD20460 \$MC_LOOKAH_SMOOTH_FACTOR) auf 0 % eingestellt ist (= Funktion deaktiviert; Voreinstellung!), wird als Ersatz ein Glättungsfaktor von 100 % verwendet. Bei einem von 0 % verschiedenen Glättungsfaktor wird der eingestellte Wert wirksam.

Parametrierung

Anpassfaktor der Bahndynamik

Durch den Anpassfaktor der Bahndynamik werden kurzzeitige Veränderungen der Bahngeschwindigkeit mit verminderten Dynamikgrenzwerten durchgeführt.

Der Anpassfaktor ist kanalspezifisch einstellbar:

- Für Verfahrbewegungen mit Beschleunigung ohne Ruckbegrenzung (**BRISK**) über:
MD20465 \$MC_ADAPT_PATH_DYNAMIC[0]
→ Der Anpassfaktor wirkt auf die Beschleunigung.
- Für Verfahrbewegungen mit Beschleunigung mit Ruckbegrenzung (**SOFT**) über:
MD20465 \$MC_ADAPT_PATH_DYNAMIC[1]
→ Der Anpassfaktor wirkt auf den Ruck.

Achsspezifische Grenzfrequenzen

Die Dynamikbegrenzung soll nur bei den Brems- und Beschleunigungsvorgängen wirken, die mechanische Schwingungen größer einer bestimmten Grenzfrequenz auslösen und dadurch zu einer Anregung von Maschinenresonanzen führen.

Die Grenzfrequenz, ab der die Dynamikbegrenzung einsetzen soll, wird achsspezifisch vorgegeben über das Maschinendatum:

MD32440 \$MA_LOOKAH_FREQUENCY (Glättungsfrequenz bei LookAhead)

Weitere Informationen siehe Kapitel "Glättung der Bahngeschwindigkeit (Seite 484)".

Funktionsweise

Bei der Bearbeitung ermittelt die Steuerung zyklisch über alle an der Bahn beteiligten Achsen das Minimum aller Grenzfrequenzen als die für die Dynamikanpassung relevante Grenzfrequenz (f) und berechnet daraus das relevante Zeitfenster (t_{adapt}):

$$t_{\text{adapt}} = 1 / f$$

Die Größe des relevanten Zeitfensters t_{adapt} bestimmt das weitere Verhalten:

1. Die erforderliche Zeit für die Geschwindigkeitsänderung ist kleiner als t_{adapt} :
Die Beschleunigungen werden reduziert um einen Faktor > 1 und \leq dem Wert im Maschinendatum:
MD20465 ADAPT_PATH_DYNAMIC (Adaption der Bahndynamik)
Durch die geringere Beschleunigung verlängert sich die Zeit für die Geschwindigkeitsänderung.
Folgende Fälle werden unterschieden:
 - Die Beschleunigung wird mit einem Wert kleiner als MD20465 so verringert, dass der Vorgang t_{adapt} [s] dauert. Die erlaubte Reduktion muss nicht voll ausgenutzt werden.
 - Die Beschleunigungszeit wird mit dem Wert im MD20465 verringert. Der Vorgang dauert trotz der geringeren Beschleunigung kürzer als t_{adapt} . Die erlaubte Reduktion wurde voll ausgenutzt.
2. Die erforderliche Zeit für die Geschwindigkeitsänderung ist größer als t_{adapt} :
Es ist keine Dynamikanpassung erforderlich.

Beispiel

Das folgende Beispiel soll die Auswirkung der Funktion "Anpassung der Bahndynamik" auf Verfahrbewegungen mit Beschleunigung ohne Ruckbegrenzung (BRISK) veranschaulichen.

Folgende Parametrierung ist gegeben:

MD20465 \$MC_ADAPT_PATH_DYNAMIC[0] = 1.5
MD20460 \$MC_LOOKAH_SMOOTH_FACTOR = 1.0
MD32440 \$MA_LOOKAH_FREQUENCY[AX1] = 20 Hz $T_{AX1} = 1/20 \text{ Hz} = 50 \text{ ms}$
MD32440 \$MA_LOOKAH_FREQUENCY[AX2] = 10 Hz $T_{AX2} = 1/10 \text{ Hz} = 100 \text{ ms}$
MD32440 \$MA_LOOKAH_FREQUENCY[AX3] = 20 Hz $T_{AX3} = 1/20 \text{ Hz} = 50 \text{ ms}$

Hinweis

Zur Verdeutlichung der Auswirkung der Dynamikanpassung ist der Wert für den Glättungsfaktor (MD20460) auf "1" gesetzt, wodurch die mitaktivierte Funktion "Glättung der Bahngeschwindigkeit" praktisch nicht aktiv ist.

An der Bahn sind die 3 Achsen X = AX1, Y = AX2, Z = AX3 beteiligt.

Bei Bahnbewegungen, an denen die Achse AX2 beteiligt ist, werden alle Brems- und Beschleunigungsvorgänge angepasst, die weniger als T_{AX2} dauern würden.

Sind bei Bahnbewegungen nur die Achsen AX1 und/oder AX3 beteiligt, werden alle Brems- und Beschleunigungsvorgänge angepasst, die weniger als $T_{AX1} = T_{AX3}$ dauern würden.

Das relevante Zeitfenster wird in den folgenden Bildern mit $t_{adapt...}$ bezeichnet.

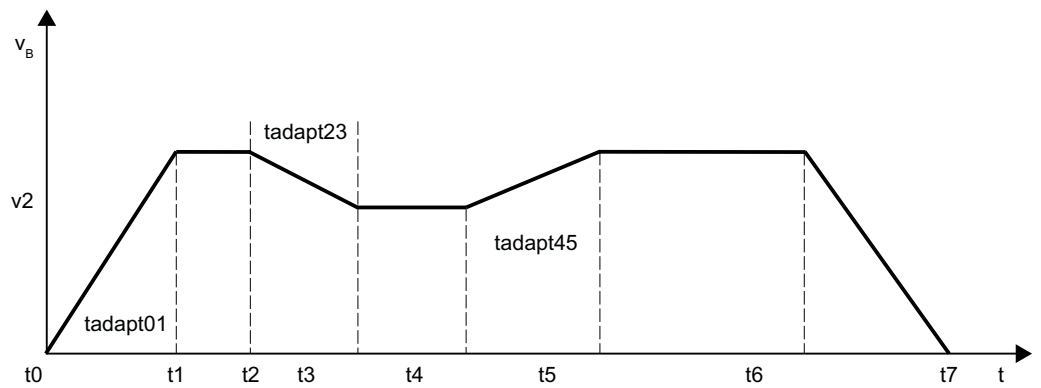
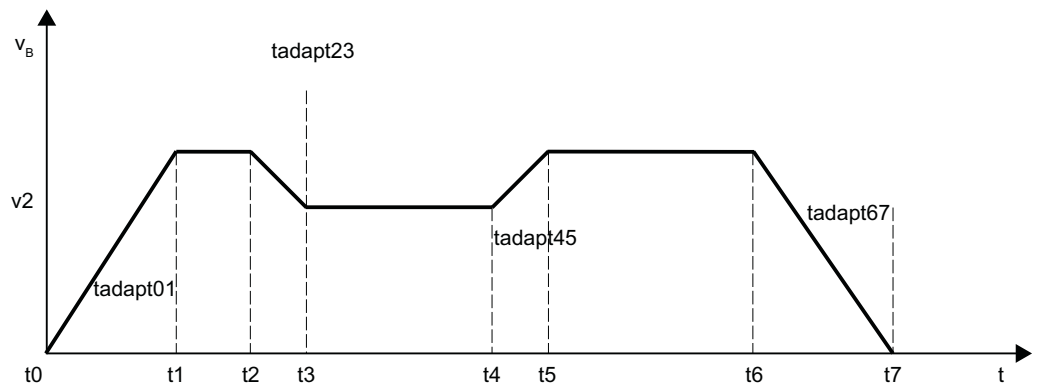


Bild 5-9 Zeitoptimaler Bahngeschwindigkeitsverlauf ohne Glättung und Dynamikanpassung

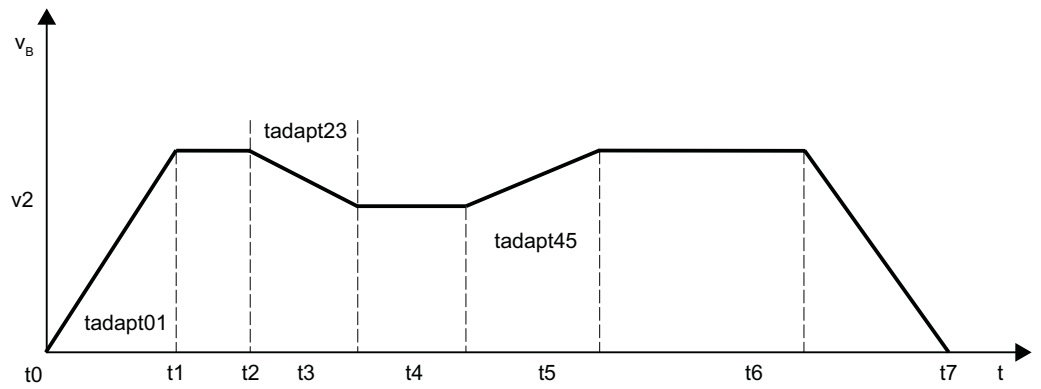


Bild 5-10 Bahngeschwindigkeitsverlauf mit Anpassung der Bahndynamik

Intervalle $t_0 - t_1$ und $t_2 - t_3$:	Der Beschleunigungsvorgang zwischen $t_0 - t_1$ und der Bremsvorgang zwischen $t_2 - t_3$ werden durch eine Anpassung der Beschleunigung auf die Zeit $t_{\text{adapt}01}$ bzw. $t_{\text{adapt}23}$ zeitlich verlängert.
Intervall $t_4 - t_5$:	Der Beschleunigungsvorgang zwischen $t_4 - t_5$ wird mit einer um den maximalen Anpassfaktor 1,5 reduzierten Beschleunigung durchgeführt. Dennoch ist der Beschleunigungsvorgang vor der Zeit $t_{\text{adapt}45}$ beendet.
Intervall $t_6 - t_7$:	Der Bremsvorgang zwischen $t_6 - t_7$ bleibt unverändert, da er länger als $t_{\text{adapt}67}$ dauert.

5.4.3 Ermittlung der Dynamikgrenzwerte

Die Inbetriebnahme der Funktion "Anpassung der Bahndynamik" erfordert neben der Ermittlung der Eigenfrequenz der Bahnachsen zur Parametrierung der achsspezifischen Grenzfrequenzen (MD32440 \$MA_LOOKAH_FREQUENCY) auch die Ermittlung der Dynamikgrenzwerte für Geschwindigkeit, Beschleunigung und Ruck.

Vorgehensweise

Im Folgenden ist die Ermittlung der Dynamikgrenzwerte für das Verfahren der Bahnachsen mittels Beschleunigung mit Ruckbegrenzung (SOFT) beschrieben. Das Vorgehen kann sinngemäß auf den Fall der Beschleunigung ohne Ruckbegrenzung (BRISK) übertragen werden.

1. Deaktivieren Sie die Funktion "Anpassung der Bahndynamik":
MD20465 \$MC_ADAPT_PATH_DYNAMIC [1] = 1
2. Untersuchen Sie das Positionierverhalten der einzelnen Bahnachsen bei unterschiedlichen Verfahrgeschwindigkeiten. Stellen Sie dabei den Ruck so ein, dass die gewünschte Positioniertoleranz eingehalten wird.

Hinweis

Je höher die Verfahrgeschwindigkeit ist, von der aus der Positioniervorgang begonnen wird, desto höher kann im Allgemeinen der Ruck eingestellt werden.

3. Übernehmen Sie den für die unkritischste Verfahrgeschwindigkeit ermittelten maximal zulässigen Ruck:
MD32431 \$MA_MAX_AX_JERK (Maximaler Ruck)

4. Ermitteln Sie für alle Bahnachsen den Faktor F_{APD} mit:
 $F_{APD} = (\text{größter ermittelter Ruck}) / (\text{kleinster ermittelter Ruck})$

Hinweis

Der kleinste ermittelte Ruck ist der Wert für den Ruck bei der kritischsten
 Verfahrensgeschwindigkeit.

5. Tragen Sie den über alle Bahnachsen ermittelten größten Faktor F_{APD} als Wert für den
 Anpassfaktor der Bahndynamik ein:
 MD20465 \$MC_ADAPT_PATH_DYNAMIC [1] = F_{APD}

5.4.4 Zusammenwirken der Funktionen "Glättung der Bahngeschwindigkeit" und "Anpassung der Bahndynamik"

Die folgenden Beispiele sollen das Zusammenwirken der Funktionen "Glättung der Bahngeschwindigkeit" und "Anpassung der Bahndynamik" im Bahnsteuerbetrieb veranschaulichen.

Beispiel 1

Beschleunigungsmodus: BRISK

An der Bahn sind die 3 Achsen X = AX1, Y = AX2, Z = AX3 beteiligt.

Folgende Parametrierung ist gegeben:

MD20465 \$MC_ADAPT_PATH_DYNAMIC[0] = 3
 MD20460 \$MC_LOOKAH_SMOOTH_FACTOR = 80.0
 MD32440 \$MA_LOOKAH_FREQUENCY[AX1] = 20 $T_{AX1} = 1/20 \text{ Hz} = 50 \text{ ms}$
 MD32440 \$MA_LOOKAH_FREQUENCY[AX2] = 20 $T_{AX2} = 1/20 \text{ Hz} = 50 \text{ ms}$
 MD32440 \$MA_LOOKAH_FREQUENCY[AX3] = 20 $T_{AX3} = 1/20 \text{ Hz} = 50 \text{ ms}$

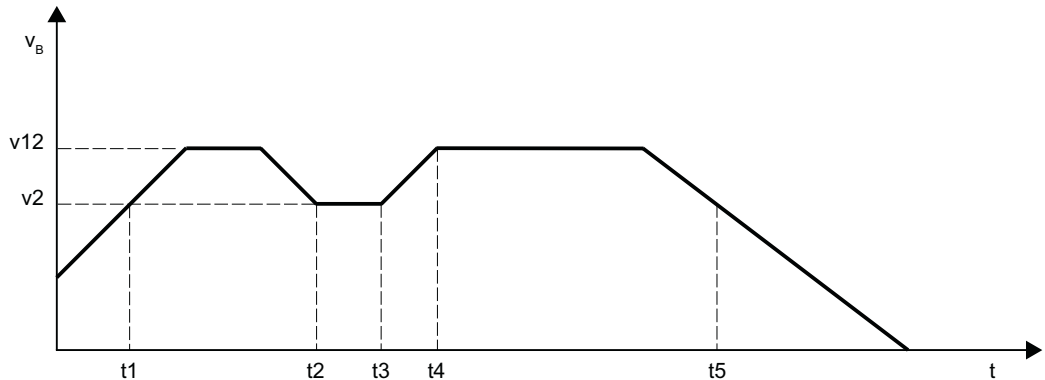


Bild 5-11 Zeitoptimaler Bahngeschwindigkeitsverlauf ohne Glättung und Dynamikanpassung

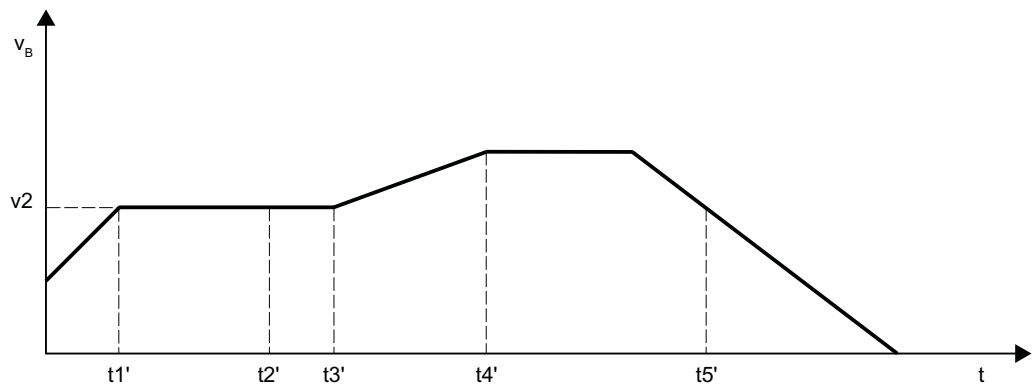


Bild 5-12 Bahngeschwindigkeitsverlauf mit Glättung der Bahngeschwindigkeit und Anpassung der Bahndynamik

Auswirkungen der Glättung der Bahngeschwindigkeit:

- Intervall $t_1 - t_2$: Der Beschleunigungs- und Bremsvorgang zwischen $t_1 - t_2$ entfällt, da die Verlängerung der Bearbeitungszeit ohne den Beschleunigungsvorgang auf v_{12} kleiner als die sich mittels Glättungsfaktor von 80 % ergebende Zeit ist.
- Intervall $t_3 - t_5$: Der Beschleunigungs- und Bremsverlauf zwischen $t_3 - t_5$ erfüllt diese Bedingung nicht oder dauert länger als die parametrisierte Glättungszeit $T_{AXn} = 2/20 \text{ Hz} = 100 \text{ ms}$.

Auswirkungen der Dynamikanpassung:

- Intervall $t_3 - t_4$: Der Beschleunigungsvorgang zwischen $t_3 - t_4$ ist kürzer als $\text{MIN}(T_{AXn}) = 1/20 \text{ Hz} = 50 \text{ ms}$ und wird daher mit einer um den Anpassfaktor 3 geringeren Beschleunigung durchgeführt.
- Intervall bis t_1 : Der nach der Bahnglättung verbleibende Beschleunigungsvorgang bis t_1 wird durch die Dynamikanpassung auf den Zeitraum bis t_1' gedehnt.

Hinweis

Das Beispiel zeigt, dass diejenigen Beschleunigungs- oder Bremsvorgänge, die von der Glättung der Bahngeschwindigkeit nicht eliminiert werden, anschließend noch durch die Anpassung der Bahndynamik optimiert werden können. Daher sollten immer möglichst beide Funktionen aktiviert werden.

Beispiel 2

Beschleunigungsmodus: SOFT

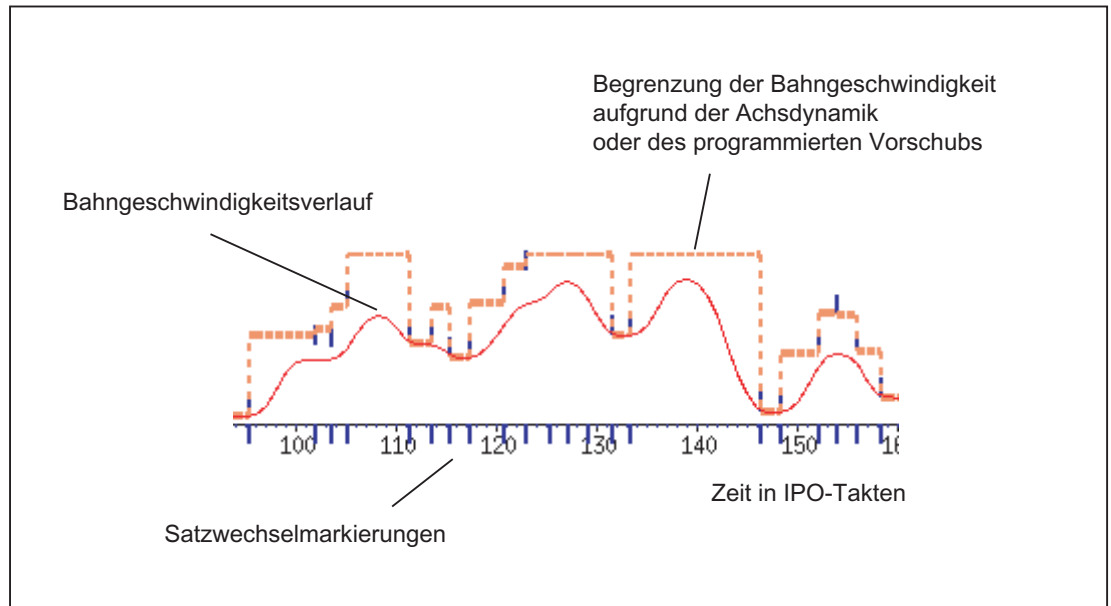
An der Bahn sind die 3 Achsen $X = AX1$, $Y = AX2$, $Z = AX3$ beteiligt.

Folgende Parametrierung ist gegeben:

```
MD20465 $MC_ADAPT_PATH_DYNAMIC[1] = 1
MD20460 $MC_LOOKAH_SMOOTH_FACTOR = 0.0
MD32440 $MA_LOOKAH_FREQUENCY[AX1] = 10      TAX1 = 1/20 Hz = 100 ms
```

MD32440 \$MA_LOOKAH_FREQUENCY[AX2] = 10 $T_{AX2} = 1/20 \text{ Hz} = 100 \text{ ms}$
 MD32440 \$MA_LOOKAH_FREQUENCY[AX3] = 20 $T_{AX3} = 1/20 \text{ Hz} = 50 \text{ ms}$

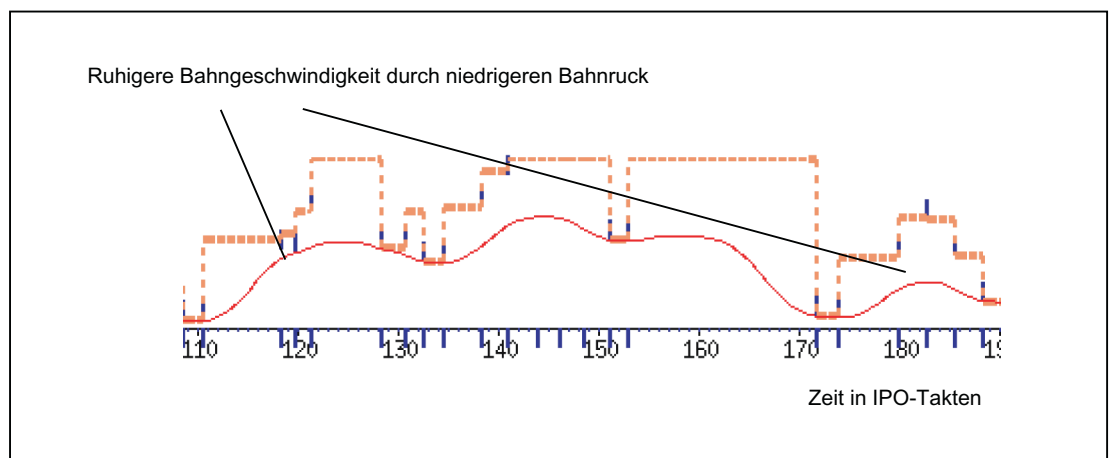
Dies führt zu einem zeitoptimalen Bahngeschwindigkeitsverlauf ohne Glättung der Bahngeschwindigkeit und ohne Anpassung der Bahndynamik:



Die Parametrierung wird wie folgt geändert:

MD20465 \$MC_ADAPT_PATH_DYNAMIC[1] = 4
 MD20460 \$MC_LOOKAH_SMOOTH_FACTOR = 1.0

Daraus ergibt sich ein Bahngeschwindigkeitsverlauf mit Anpassung der Bahndynamik und minimaler und damit fast abgeschalteter Glättung der Bahngeschwindigkeit:



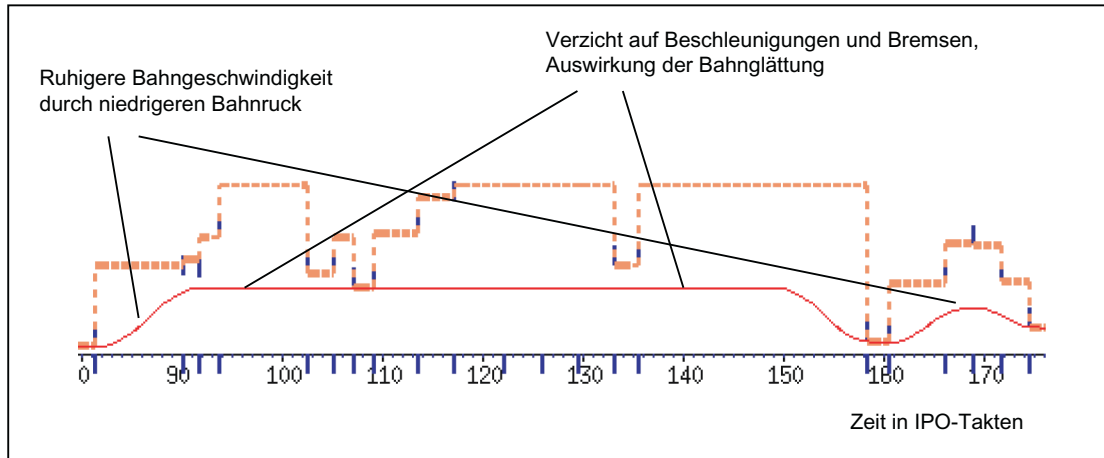
Der Glättungsfaktor wird statt auf 1 % auf 0 % eingestellt (entspricht der Voreinstellung!):

MD20460 \$MC_LOOKAH_SMOOTH_FACTOR = 0.0

5.4 Dynamikanpassungen

Mit dieser Parametrierung wird ein Glättungsfaktor von 100 % wirksam.

Daraus ergibt sich ein Bahngeschwindigkeitsverlauf mit Glättung der Bahngeschwindigkeit und Anpassung der Bahndynamik:



5.4.5 Dynamikmodus für Bahninterpolation

Funktion

Technologie-spezifische Dynamikeinstellungen können in Maschinendaten hinterlegt und im Teileprogramm über die Befehle der G-Gruppe 59 (Dynamikmodus für Bahninterpolation) aktiviert werden.

Befehl	Aktiviert die Dynamikeinstellungen für:
DYNNORM	Standard-Dynamikeinstellungen
DYNPOS	Positionierbetrieb, Gewindebohren
DYNROUGH	Schruppen
DYNSEMIFIN	Vorschlichten
DYNFINISH	Schlichten
DYNPREC	Feinschlichten

Hinweis

Mit den Befehlen der G-Gruppe 59 (Dynamikmodus für Bahninterpolation) wird ausschließlich die Dynamik der **Bahnachsen** bestimmt. Sie haben keinen Einfluss auf:

- Positionierachsen
- PLC-Achsen
- Kommandoachsen
- Bewegungen aufgrund von Achskopplungen
- Überlagerte Bewegungen mit Handrad
- JOG-Bewegungen
- Referenzpunktfahren (G74)
- Festpunktfahren (G75)
- Eilgangbewegung (G0)

Für diese Achsbewegungen ist immer die Standard-Dynamikeinstellung (DYNNORM) wirksam.

Anwendung

Durch Umschalten der Dynamikeinstellungen ist es z. B. möglich, ein Werkstück beim Schruppen zeitoptimiert und beim Schlichten oberflächenoptimiert zu bearbeiten.

Parametrierung**Achsspezifische Dynamikeinstellungen:**

- MD32300 \$MA_MAX_AX_ACCEL[<n>] (Achsbeschleunigung)
- MD32310 \$MA_MAX_ACCEL_OVL_FACTOR[<n>] (Überlastfaktor für axiale Geschwindigkeitssprünge)
- MD32431 \$MA_MAX_AX_JERK[<n>] (maximaler axialer Ruck bei Bahnbewegung)
- MD32432 \$MA_PATH_TRANS_JERK_LIM[<n>] (maximaler axialer Ruck am Satzübergang im Bahnsteuerbetrieb)
- MD32433 \$MA_SOFT_ACCEL_FACTOR[<n>] (Skalierung der Beschleunigungsbegrenzung bei SOFT)

Kanalspezifische Dynamikeinstellungen:

- MD22450 \$MC_DYN_LIM_MODE[<n>] (Achssiale oder geometrische Geschwindigkeitsbegrenzungen)
- MD20600 \$MC_MAX_PATH_JERK[<n>] (Bahnbezogener Maximalruck)
- MD20602 \$MC_CURV_EFFECT_ON_PATH_ACCEL[<n>] (Einfluss der Bahnkrümmung auf die Bahnbeschleunigung)
- MD20603 \$MC_CURV_EFFECT_ON_PATH_JERK[<n>] (Einfluss der Bahnkrümmung auf den Bahnruck)

mit Index <n> = 0 für DYNNORM
1 für DYNPOS
2 für DYNROUGH
3 für DYNSEMIFIN
4 für DYNFINISH
5 für DYNPREC

Hinweis

Das **Beschreiben** der Maschinendaten **ohne Index** setzt den gleichen Wert in alle Feldelemente des betreffenden Maschinendatums.

Das **Lesen** der Maschinendaten **ohne Index** liefert immer den Wert des Feldes mit dem Index 0.

G-Befehle unterdrücken

Es wird empfohlen, nicht zur Verwendung vorgesehene G-Befehle der G-Gruppe 59 (Dynamikmodus für Bahninterpolation) über das folgende Maschinendatum zu unterdrücken:

MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[<n>] (Liste umprojektierter NC-Befehle)

Bei Verwendung eines unterdrückten G-Befehls wird ein Alarm angezeigt. Dadurch wird verhindert, dass nicht-parametrierte Maschinendaten wirksam werden.

Beispiel

Mit folgenden Einstellungen können die G-Befehle `DYNPOS` und `DYNSEMIFIN` unterdrückt werden:

- MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[0]="DYNPOS"
- MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[1]=" "
- MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[2]="DYNSEMIFIN"
- MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[3]=" "

Weitere Informationen

Ausführliche Informationen zur Programmierung der G-Befehle der G-Gruppe 59 (Dynamikmodus für Bahninterpolation) finden sich in:
Programmierhandbuch Arbeitsvorbereitung; Bahnverhalten

5.4.6 Freiformflächenmodus: Grundfunktionen

Einleitung

Bei Anwendungen im Werkzeug- und Formenbau ist es wichtig, dass möglichst gleichmäßige Oberflächen auf dem Werkstück entstehen. Diese Anforderung ist meistens wichtiger als die Genauigkeit der Werkstückoberfläche.

Ungleichmäßige Werkstückoberflächen können z. B. folgende Ursachen haben:

- Das Teileprogramm zur Fertigung des Werkstücks enthält eine ungleichmäßige Geometrie. Dies betrifft vor allem den Verlauf der Krümmung und der Torsion.

Hinweis

Die Krümmung k einer Kontur ist das Inverse des Radius r des anschmiegenden Kreises in einem Konturpunkt ($k = 1/r$). Die Torsion ist die Änderung der Krümmung (1. Ableitung).

Durch die ungleichmäßige Geometrie werden bei der Abarbeitung des Teileprogramms die Dynamikgrenzen der Maschine erreicht und es kommt zu unnötigen Brems- und Beschleunigungsvorgängen. Dies führt je nach Größe des effektiven Nachlaufs der Achsen zu unterschiedlichen Konturabweichungen.

- Unnötige Brems- und Beschleunigungsvorgänge können zur Anregung von Schwingungen der Maschine führen, die als unerwünschte Marken auf dem Werkstück sichtbar werden.

Zur Behebung dieser Ursachen gibt es verschiedene Möglichkeiten:

- CAD/CAM-System verwenden
Von CAD/CAM-Systemen generierte Teileprogramme enthalten einen sehr gleichmäßigen Verlauf der Krümmung und der Torsion, sodass es nicht zu unnötigen Reduzierungen der Bahngeschwindigkeit kommen kann.
- Maximale Bahngeschwindigkeit so bestimmen, dass ungewollte geometrische Schwankungen des Verlaufs der Krümmung und Torsion keinen Einfluss haben.

Funktion

Mit der Funktion "Freiformflächenmodus: Grundfunktionen" kann **für die aktiven Geometrieachsen und, bei aktiver Orientierungstransformation, den an der Transformation beteiligten Rundachsen** die Festlegung der Begrenzung der Bahngeschwindigkeit unempfindlicher gegenüber kleinen geometrischen Schwankungen der Krümmung bzw. Torsion gemacht werden, ohne dass die dynamischen Begrenzungen der Maschine bezüglich der Beschleunigung und des Rucks der Achsen überschritten werden.

Dadurch ergeben sich folgende Vorteile:

- Gleichmäßigerer Verlauf der Bahngeschwindigkeit
- Gleichmäßigere Oberfläche des Werkstücks
- Verringerung der Bearbeitungszeit (falls die Dynamik der Maschine dies zulässt)

Anwendungen

Die Funktion wird zur Bearbeitung von Werkstücken verwendet, die vorwiegend aus Freiformflächen bestehen.

Voraussetzungen

Die Funktion ist nur aktivierbar, wenn bei der Speicherkonfiguration der notwendige Speicher reserviert wird:

MD28610 \$MC_MM_PREPDYN_BLOCKS = 10

Der eingetragene Wert gibt die Anzahl von Sätzen vor, die bei der Festlegung der Bahngeschwindigkeit (Geschwindigkeitspräparation) berücksichtigt werden sollen. Ein sinnvoller Wert ist "10".

Hat MD28610 den Wert "0", dann werden zur Festlegung der maximalen Bahngeschwindigkeit eines Satzes nur die jeweiligen Bewegungen der Achsen in diesem Satz berücksichtigt. Wird bei der Festlegung der Bahngeschwindigkeit auch die Geometrie in benachbarten Sätzen berücksichtigt (Wert > 0), erhält man einen gleichmäßigeren Verlauf der Bahngeschwindigkeit.

Aktivierung / Deaktivierung

Die Funktion kann für jeden Dynamikmodus (siehe Kapitel "Dynamikmodus für Bahninterpolation (Seite 496)") unabhängig ein- oder ausgeschaltet werden:

MD20606 \$MC_PREPDYN_SMOOTHING_ON[<n>] = <Wert>

Index <n>	Dynamikmodus	<Wert>	Freiformflächenmodus: Grundfunktionen
0	Standard-Dynamikeinstellungen (DYNORM)	0	aus
		1	ein
1	Positionierbetrieb, Gewindebohren (DYNPOS)	0	aus
		1	ein
2	Schruppen (DYNROUGH)	0	aus
		1	ein
3	Vorschlichten (DYNSEMIFIN)	0	aus
		1	ein
4	Schlichten (DYNFINISH)	0	aus
		1	ein
5	Feinschlichten (DYNPREC)	0	aus
		1	ein

Hinweis

Die Funktion sollte wegen des zusätzlichen Speicherbedarfs nur in den relevanten Bearbeitungskanälen aktiviert werden.

Parametrierung

Veränderung des Konturabtastfaktors

Bei der Interpolation von gekrümmten Konturen ist der dabei entstehende Sekantenfehler von folgenden Faktoren abhängig:

- Krümmung
- Interpolatortakt (Anzeige im MD10071 \$MN_IPO_CYCLE_TIME)
- Geschwindigkeit, mit der die betreffende Kontur abgefahren wird

Der maximal mögliche Sekantenfehler ist für jede Achse festgelegt im Maschinendatum:

MD33100 \$MA_COMPRESS_POS_TOL (Maximal Abweichung bei Kompression)

Wenn der eingestellte Interpolatortakt nicht ausreichend klein ist, kann es vorkommen, dass bei stärker gekrümmten Konturen die maximale Bahngeschwindigkeit reduziert wird. Dies ist notwendig, damit auch in diesem Fall die Oberfläche des Werkstücks noch ausreichend genau hergestellt wird.

Durch die Veränderung des Konturabtastfaktors ist es möglich, das Zeitintervall, mit dem eine gekrümmte Kontur im Interpolator abgetastet wird (Konturabtastzeit), unterschiedlich zum Interpolatortakt einzustellen. Eine Konturabtastzeit niedriger als der Interpolatortakt kann so bei stärker gekrümmten Konturen eine Reduzierung der Bahngeschwindigkeit verhindern.

Der Konturabtastfaktor wird eingestellt mit dem Maschinendatum:

MD10682 \$MN_CONTOUR_SAMPLING_FACTOR

Die wirksame Konturabtastzeit berechnet sich daraus wie folgt:

$$T_s = f * T_1$$

mit: T_s = wirksame Konturabtastzeit
 T_1 = Interpolatortakt
 f = Konturabtastfaktor (Wert aus MD10682)

Der Standardwert des Konturabtastfaktors ist "1", d. h., die Konturabtastzeit ist gleich dem Interpolatortakt.

Der Konturabtastfaktor kann sowohl größer als auch kleiner "1" sein.

Wird ein Wert kleiner "1" eingestellt, kann man erreichen, dass die Überprüfung der Genauigkeit der Konturabtastung außer Kraft gesetzt wird.

Die eingestellte Abtastzeit darf die projektierte minimale Konturabtastzeit nicht unterschreiten:

MD10680 \$MN_MIN_CONTOUR_SAMPLING_TIME

Hinweis

MD10680 ist für jedes Steuerungsmodell spezifisch eingestellt und nicht veränderbar.

Geschwindigkeitsbegrenzung

Der Modus der Geschwindigkeitsbegrenzung aufgrund von Krümmungen kann für jeden Dynamikmodus über das folgende Maschinendatum vorgegeben werden:

MD22450 \$MC_DYN_LIM_MODE [<n>] = <Wert>

mit Index <n> = 0 für DYNNORM
 1 für DYNPOS

5.4 Dynamikanpassungen

- 2 für DYNROUGH
- 3 für DYNSEMIFIN
- 4 für DYNFINISH
- 5 für DYNPREC

<Wert>	Bedeutung
= 0	Geschwindigkeitsbegrenzung erfolgt aufgrund der achsspezifischen Beschleunigungen der an der Bahn beteiligten Geometrieachsen Diese Einstellung führt zu einer höheren Bahngeschwindigkeit, weil dann in diagonaler Richtung schneller gefahren werden kann.
= 1	Geschwindigkeitsbegrenzung erfolgt aufgrund der geometrischen Krümmung der an der Bahn beteiligten Geometrieachsen Mit dieser Einstellung wird die Bahngeschwindigkeit an Stellen mit diagonaler und achspareller Tangente gleich begrenzt. Der Verlauf der Bahngeschwindigkeit ist dann ruhiger. Diese Einstellung sollte man daher beim Schlichten wählen.

MD22450 hat Auswirkungen auf die Begrenzung der Bahngeschwindigkeit durch:

- Funktion "Programmierbare Konturgenauigkeit"
- Maximale Achsbeschleunigung (MD32300 \$MA_MAX_AX_ACCEL)
- Konturabtastfaktor (MD10682 \$MN_CONTOUR_SAMPLING_FACTOR)

Programmierung

Abhängig von der Einstellung im Maschinendatum

MD20606 \$MC_PREPDYN_SMOOTHING_ON kann die Funktion "Freiformflächenmodus: Grundfunktionen" im Teileprogramm durch einen Wechsel des aktiven Dynamikmodus ein- oder ausgeschaltet werden.

Beispiel:

Durch die Parametrierung von MD20606 \$MC_PREPDYN_SMOOTHING_ON[2-5] = 1 und MD20606 \$MC_PREPDYN_SMOOTHING_ON[0-1] = 0 kann die Funktion über die Befehle DYNROUGH, DYNSEMIFIN, DYNFINISH und DYNPREC eingeschaltet und über die Befehle DYNNORM und DYNPOS ausgeschaltet werden.

Siehe auch

Überschleifen tangentialer Satzübergänge (G645) (Seite 472)

Freiformflächenmodus: Erweiterungsfunktion (Seite 480)

5.5 Kompressor-Funktionen

5.5.1 Kompression von Linear-, Kreis- und Eilgangsätzen

5.5.1.1 Funktion

CAD/CAM-Systeme erzeugen zur Beschreibung von komplexen Konturen eine große Anzahl von Linear- und Kreissätzen mit zum Teil sehr kurzen Bahnlängen. Die maximal mögliche Bahngeschwindigkeit wird dabei häufig durch die Satzwechselzeit begrenzt. Ab einer bestimmten Bahngeschwindigkeit können dann vom Vorlauf nicht mehr schnell genug neue Verfahrsätze aufbereitet und in den Hauptlauf eingewechselt werden.

Kompression von Linearsätzen

Kompressor-Funktionen ersetzen aufeinander folgende Linearsätze durch Polynom-Sätze mit möglichst großen Bahnlängen unter Einhaltung einer parametrierbaren Konturgenauigkeit. Dadurch ergeben sich folgende Vorteile:

- Reduzierung der Anzahl von Verfahrätzen
- Erhöhung der Bahngeschwindigkeit
- Erhöhung der Oberflächengüte
- Stetige Satzübergänge

In der folgenden Tabelle sind die zur Verfügung stehenden Kompressor-Funktionen mit ihren wichtigsten Eigenschaften aufgelistet:

Kompressor	Funktion	Stetigkeit an Satzübergängen	Hinweise zur Anwendung
COMPON	COMPON erzeugt aus bis zu 10 aufeinander folgenden Linear-sätzen der Form "G01 X... Y... Z... F..." einen Polynom-Satz.	geschwindigkeitsstetig	
COMPCURV	wie COMPON	geschwindigkeits- und beschleunigungsstetig	

5.5 Kompressor-Funktionen

Kompressor	Funktion	Stetigkeit an Satzübergängen	Hinweise zur Anwendung
COMPCAD	COMPCAD kann aus beliebig vielen aufeinander folgenden Linearsätzen einen Polynom-Satz erzeugen.	geschwindigkeits- und beschleunigungsstetig	COMPCAD ist sehr rechenzeit- und speicherplatzintensiv. Es wird daher empfohlen, COMPCAD nur dort einzusetzen, wo Maßnahmen zur Oberflächenverbesserung im CAD/CAM-Programm nicht zum Erfolg führen.
COMPSURF	wie COMPCAD	geschwindigkeits- und beschleunigungsstetig	Mit COMPSURF können noch bessere Ergebnisse als mit COMPCAD erzielt werden. Ebenso wie COMPCAD ist COMPSURF jedoch sehr rechenzeit- und speicherplatzintensiv. Durch den Einsatz von COMPSURF können z. B. bei schräg abgezeilten Schlichtprogrammen, schlechter Datenqualität und/oder unregelmäßiger Punkteverteilung im CAD/CAM-Programm deutlich bessere Werkstückoberflächen erhalten werden. Außerdem ermöglicht COMPSURF eine richtungsunabhängige identische abschaltbare Glättung der Fräsbahnen, was beim bidirektionalen Fräsen die Oberflächengüte deutlich erhöht.

Wenn in und zwischen den zu komprimierenden Verfahransätzen Befehle programmiert werden, die keine Verfahrensanweisungen sind (z. B. Hilfsfunktionsausgaben), wird der Kompressionsvorgang unterbrochen.

Die maximal tolerierbare Abweichung der berechneten Bahn zu den programmierten Positionen kann für alle Kompressor-Funktionen über Maschinendaten vorgegeben werden (siehe "Parametrierung (Seite 505)"). Anders als bei COMPON und COMPCURV werden die parametrisierten Toleranzen bei COMPCAD und COMPSURF in benachbarten Bahnen nicht in unterschiedlichen Richtungen ausgenutzt.

Kompression von Kreis- und Eilgangsätzen

Außer der Kompression von Linearsätzen bieten alle Kompressor-Funktionen die Möglichkeit, auch Eilgangsätze (G0-Sätze) zu komprimieren. Die Kompression von Kreissätzen ist dagegen nur mit der Kompressor-Funktion COMPCAD möglich.

Die Kompression von Kreis- und/oder Eilgangsätzen wird kanalspezifisch eingestellt über die **Hunderterstelle** im Maschinendatum:

MD20482 \$MC_COMPRESSOR_MODE = <Wert>

Wert	Bedeutung
0xx	Kreissätze und G0-Sätze werden nicht komprimiert. Dies ist kompatibel zu früheren SW-Versionen.
1xx	Kreissätze werden von COMPCAD linearisiert und komprimiert. Vorteil: Die Kompressor-Funktion arbeitet genauer und erzeugt dadurch i. d. R. bessere Oberflächen. Nachteil: Die Kompressor-Funktion wird empfindlicher gegenüber Defekten in den NC-Programmen. Aus Kompatibilitätsgründen kann es deshalb notwendig sein, die Einstellung 0xx beizubehalten.

Wert	Bedeutung
2xx	G0-Sätze werden komprimiert, evtl. wird dabei eine andere Toleranz wirksam (siehe "Toleranzen bei Eilgangbewegungen (Seite 518)"). Vorteil: Durch die mögliche Einstellung einer größeren Toleranz und der Kompression von G0-Zustellbewegungen können diese schneller und flüssiger abgefahren werden.
3xx	Kombination der beiden vorhergehenden Möglichkeiten: sowohl Kreissätze als auch G0-Sätze werden komprimiert.

Aktivierung/Deaktivierung

Die Kompressor-Funktionen COMPON, COMPCURV, COMPCAD und COMPSURF werden mit den G-Befehlen der G-Gruppe 30 ein-/ausgeschaltet (siehe "Programmierung (Seite 507)").

5.5.1.2 Inbetriebnahme

Parametrierung

Achsspezifische Maschinendaten

Nummer	Bezeichner \$MA_	Bedeutung
MD33100	COMPRESS_POS_TOL	Maximal erlaubte achsspezifische Bahnabweichung bei Kompression

Kanalspezifische Maschinendaten

Nummer	Bezeichner \$MC_	Bedeutung
MD20170	COMPRESS_BLOCK_PATH_LIMIT	Maximale Verfahrlänge eines komprimierbaren NC-Satzes
MD20171	SURF_BLOCK_PATH_LIMIT	Maximale Verfahrlänge eines komprimierbaren NC-Satzes bei Kompression mit COMPSURF
MD20172	COMPRESS_VELO_TOL	Maximal erlaubte Abweichung des Bahnvorstubs bei Kompression
MD20173	SURF_VELO_TOL	Maximal erlaubte Abweichung des Bahnvorstubs bei Kompression mit COMPSURF
MD20482	COMPRESSOR_MODE	Arbeitsweise des Kompressors
MD20485	COMPRESS_SMOOTH_FACTOR	Glättungsfaktor bei Kompression mit COMPCAD für den jeweiligen Dynamikmodus
MD20486	COMPRESS_SPLINE_DEGREE	Spline-Grad bei Kompression mit COMPCAD für den jeweiligen Dynamikmodus
MD20487	COMPRESS_SMOOTH_FACTOR_2	Glättungsfaktor für Rundachsen bei Kompression mit COMPCAD für den jeweiligen Dynamikmodus

Nummer	Bezeichner \$MC_	Bedeutung
MD28071	MM_NUM_SURF_LEVELS	Dimensionierung der Funktion COMPSURF (DRAM)
MD28072	MM_MAXNUM_SURF_GROUPS	Dimensionierung der Funktion COMPSURF bzgl. Achsgruppen (DRAM)

Kanalspezifische Settingdaten

Nummer	Bezeichner \$SC_	Bedeutung
SD42470	CRIT_SPLINE_ANGLE	Ecken-Grenzwinkel bei Kompression mit COMP-CAD
SD42471	MIN_CURV_RADIUS	Minimaler Krümmungsradius bei Kompression mit COMPCAD
SD42472	MIN_SURF_RADIUS	Minimaler Krümmungsradius bei Kompression mit COMPSURF
SD42473	ACTNUM_SURF_GROUPS	Anzahl Achsgruppen bei COMPSURF
SD42475	COMPRESS_CONTUR_TOL	Maximal erlaubte Konturabweichung bei Kompression
SD42476	COMPRESS_ORI_TOL	Maximale Abweichung der Werkzeugorientierung bei Kompression Hinweis: Nur bei aktiver Orientierungstransformation!
SD42477	COMPRESS_ORI_ROT_TOL	Maximale Abweichung der Werkzeugdrehung bei Kompression Hinweis: Nur bei 6-Achs-Maschinen mit drehbarem Werkzeug!

Hinweis

Ecken-Grenzwinkel und Kompressorfunktion COMPCAD

Der über das Settingdatum SD42470 \$SC_CRIT_SPLINE_ANGLE eingestellte Ecken-Grenzwinkel für die Kompressor-Funktion COMPCAD dient nur als ungefähres Maß zur Eckenerkennung. Der Kompressor erkennt aufgrund von Plausibilitätsbetrachtungen auch flachere Satzübergänge als Ecken und größere Winkel als Ausreißer.

Einstellempfehlungen für den Werkzeug- und Formenbau mit Advanced Surface / Top Surface

Kompressor-Funktionen haben eine wichtige Bedeutung beim Fräsen von Freiformflächen im Werkzeug- und Formenbau. Für den Einsatz im Rahmen der lizenzpflichtigen Option "Advanced Surface" bzw. "Top Surface" sind Einstellempfehlungen zu beachten!

Zur Überprüfung der eingestellten Maschinen- und Settingdaten stehen über das SIOS-Portal spezielle Prüfprogramme zur Verfügung:

- Prüfprogramme für Advanced Surface (<https://support.industry.siemens.com/cs/ww/de/view/78956392>)
- Prüfprogramme für Top Surface (<https://support.industry.siemens.com/cs/ww/de/view/109738423>)

5.5.1.3 Programmierung

NC-Satz-Kompression ein-/ausschalten (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF)

Die Funktionen zur Kompression von Linearsätzen (und abhängig von der Parametrierung auch Kreis- und/oder Eilgangsätzen) 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 Programmanfang 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. Verfahrssatz
N24051 X41.365 Z32.500	; 2. Verfahrssatz
...	
N99999 X... Z...	; letzter Verfahrssatz
COMPOF	; Kompressor-Funktion aus.
...	

5.5.1.4 Randbedingungen

Orientierungstransformation (TRAORI)

Die Kompressor-Funktionen können bei aktiver Orientierungstransformation auch Bewegungssätze zur Werkzeugorientierung und Werkzeugdrehung komprimieren.

Weitere Informationen

Funktionshandbuch Transformationen; Mehrachstransformationen > Orientierung > Komprimierung der Orientierung

Satzsuchlauf mit Berechnung

Wenn bei Satzsuchlauf **Typ 2** oder **Typ 4** (Satzsuchlauf **mit Berechnung** an ...) der Zielsatz in einem Programmabschnitt liegt, in dem eine Kompressor-Funktion aktiv ist, werden beim Wiederanfahren an die Kontur Positionen auf der durch den Kompressor berechneten Bahn angefahren. Diese Positionen müssen nicht exakt mit den Positionen auf der im Teileprogramm programmierten Bahn übereinstimmen.

Teileprogrammsätze, die durch die Komprimierung ersetzt werden, können im Satzsuchlauf nicht als Zielsatz gefunden werden. Es wird Alarm 15370 "Suchziel bei Satzsuchlauf nicht gefunden" ausgegeben.

5.5.2 Kompression kurzer Spline-Sätze

Funktion

Bei der Erzeugung von Spline-Sätzen zur Beschreibung von komplexen Konturen durch CAD/CAM-Systeme entstehen zwischen Spline-Sätzen mit großen Bahnlängen auch immer wieder Spline-Sätze mit sehr kurzen Bahnlängen. Diese zwingen die Steuerung zu einer signifikanten Reduzierung der Bahngeschwindigkeit. Die Funktionen zur Kompression kurzer Spline-Sätze erzeugen neue Spline-Sätze so, dass möglichst große Bahnlängen entstehen.

Inbetriebnahme

Aktivierung

Die Kompression kurzer Spline-Sätze kann für folgende Spline-Arten aktiviert werden:

- BSPLINE
- BSPLINE / ORICURVE
- CSPLINE

Die Aktivierung erfolgt über das kanalspezifische Maschinendatum:

MD20488 \$MC_SPLINE_MODE, Bit <n> = <Wert>

Bit	<Wert>	Spline-Art	Kompression kurzer Spline-Sätze
0	0	BSPLINE	nicht aktiv
	1	BSPLINE	aktiv
1	0	BSPLINE / ORICURVE	nicht aktiv
	1	BSPLINE / ORICURVE	aktiv
2	0	CSPLINE	nicht aktiv
	1	CSPLINE	aktiv

Randbedingungen

- Wenn in und zwischen den zu komprimierenden Verfahrssätzen Befehle programmiert sind, die keine Verfahrensanweisungen sind, z. B. Hilfsfunktionsausgaben, können die Spline-Sätze nicht zusammengefasst werden.
- Die maximale Anzahl hintereinander liegender Spline-Sätzen, die in einem Programmabschnitt zusammengefasst werden können, hängt von der parametrierten Größe des Satzspeichers in der Satzaufbereitung ab:
MD28070 \$MC_MM_NUM_BLOCKS_IN_PREP (Anzahl der Sätze für die Satzaufbereitung)

Beispiel

Um beim Abarbeiten der Verfahrssätze eine höhere Bahngeschwindigkeit erreichen zu können, wird die Kompression für kurze Spline-Sätze für die BSPLINE-Interpolation aktiviert:

MD20488 \$MC_SPLINE_MODE, Bit 0 = 1

Programmcode	Kommentar
N10 G1 G64 X0 Y0 Z0 F1000	; Grundstellung
N20 G91 F10000 BSPLINE	; Einschalten: B-Spline
N30 X0.001 Y0.001 Z0.001	; Ab hier: Zusammenfassung kurzer Spline-Sätze
N40 X0.001 Y0.001 Z0.001	
...	

5.6 Kontur-/Orientierungstoleranz

5.6.1 Inbetriebnahme

5.6.1.1 Parametrierung

Maschinendaten

Konturtoleranz / Orientierungstoleranz

MD33100 \$MA_COMPRESS_POS_TOL[<Achse>] = <Wert> (Maximale Abweichung bei Kompression)

Über das achsspezifische Maschinendatum wird die maximal erlaubte Konturabweichung (Konturtoleranz) bzw. Winkelabweichung der Werkzeugorientierung (Orientierungstoleranz) der jeweiligen Achse eingestellt. Das Maschinendatum wirkt bei folgenden Funktionen:

- Überschleifen: G642, G643, G644, G645
- Kompressor: COMON, COMPCURV, COMPCAD, COMPSURF
- Je größer der Wert ist, umso mehr kurze Sätze können in einen langen Satz komprimiert werden.

Das Maschinendatum wirkt nicht bei Überschleiffunktion G641. Bei G641 wirkt die mit ADIS bzw. ADISPOS programmierbare Wegdistanz zum Satzübergang.

Überschleif-Modus

MD20480 \$MC_SMOOTHING_MODE (Verhalten des Überschleifens mit G64x)

Kompressor-Modus

MD20482 \$MC_COMPRESSOR_MODE (Modus des Kompressors)

Überschleifen G645

MD33120 \$MA_PATH_TRANS_POS_TOL (Maximale Konturabweichung bei Überschleifen G645)

Wirksam beim Überschleifen von tangentialen-, aber nicht krümmungsstetigen Satzübergängen (z. B. Kreis - Gerade)

Settingdaten

Kanalspezifische Konturtoleranz

SD42465 \$SC_SMOOTH_CONTUR_TOL (Maximale Konturabweichung)

Kanalspezifische Orientierungstoleranz

SD42466 \$SC_SMOOTH_ORI_TOL (Maximale Winkelabweichung der Werkzeugorientierung)

Kanalspezifische Orientierungstoleranz bei Überschleifen mit OST

SD42676 \$SSC_ORI_SMOOTH_TOL (Toleranz zur Glättung der Orientierung beim Überschleifen)

Kanalspezifische Orientierungstoleranz bei Glättung der Orientierung mit ORISON

SD42678 \$SSC_ORISON_TOL (Toleranz zur Glättung der Orientierung)

5.6.2 Programmierung

5.6.2.1 Kontur-/Orientierungstoleranz programmieren (CTOL, OTOL, ATOL)

Mit den Adressen CTOL, OTOL und ATOL können die über Maschinen- und Settingdaten parametrisierten Bearbeitungstoleranzen für Kompressor-Funktionen, Überschleifen und Orientierungsglättung im Teileprogramm angepasst werden.

Die programmierten Toleranzwerte gelten, bis sie neu programmiert oder durch Zuweisung eines negativen Werts gelöscht werden. Sie werden ferner gelöscht bei Programmende bzw. Reset. Nach dem Löschen sind wieder die parametrisierten Toleranzwerte wirksam.

Syntax

```
CTOL=<Value>
OTOL=<Value>
ATOL[<Axis>]=<Value>
```

Bedeutung

CTOL:	Adresse zum Programmieren der Konturtoleranz			
	Anwendungsbereich:	<ul style="list-style-type: none"> • alle Kompressor-Funktionen • alle Überschleifarten außer G641 und G644 		
	Vorlaufstopp:	nein		
	Wirksamkeit:	modal		
	<Value>:	Der Wert für die Konturtoleranz ist eine Längenangabe.		
		Typ:	REAL	
		Einheit:	Inch/mm (abhängig von der aktuellen Einstellung der Maßangabe)	
		Wertebereich:	≥ 0:	Toleranzwert
< 0:	Löschen des programmierten Toleranzwerts ⇒ Es gilt wieder der im Maschinen- bzw. Settingdatum parametrisierte Toleranzwert.			

5.6 Kontur-/Orientierungstoleranz

OTOL:	Adresse zum Programmieren der Orientierungstoleranz		
	Anwendungsbereich:	<ul style="list-style-type: none"> • alle Kompressor-Funktionen • Orientierungsglättung ORISON • alle Überschleifarten außer G641, G644 und OSD 	
	Vorlaufstopp:	nein	
	Wirksamkeit:	modal	
	<Value>:	Der Wert für die Orientierungstoleranz ist eine Winkelangabe.	
	Typ:	REAL	
	Einheit:	Grad	
	Wertebereich:	≥ 0:	Toleranzwert
		< 0:	Löschen des programmierten Toleranzwerts ⇒ Es gilt wieder der im Maschinen- bzw. Settingdatum parametrisierte Toleranzwert.
	ATOL: Adresse zum Programmieren einer achsspezifischen Toleranz		
	Anwendungsbereich:	<ul style="list-style-type: none"> • alle Kompressor-Funktionen • Orientierungsglättung ORISON • alle Überschleifarten außer G641, G644 und OSD 	
Vorlaufstopp:	nein		
Wirksamkeit:	modal		
<Axis>:	Name der Kanalachse, für welche die programmierte Toleranz wirken soll		
<Value>:	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
	Wertebereich:	≥ 0:	Toleranzwert
< 0:		Löschen des programmierten Toleranzwerts ⇒ Es gilt wieder der im Maschinen- bzw. Settingdatum parametrisierte Toleranzwert.	

Hinweis

Die mit CTOL und OTOL programmierten kanalspezifischen Toleranzwerte sind höherprior als die mit ATOL programmierten achsspezifischen Toleranzwerte.

Hinweis

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...	

5.6.2.2 Kontur-/Orientierungstoleranz programmieren (CTOL, OTOL, ATOL): Weitere Informationen

Systemvariablen

Lesen mit Vorlaufstopp

Über folgende Systemvariablen sind im Teileprogramm und Synchronaktion die aktuell wirksamen Toleranzen lesbar:

- **\$AC_CTOL**
Kanalspezifische Konturtoleranz, die bei der Aufbereitung des aktuellen Hauptaufsatzes wirksam war.
Wenn keine Konturtoleranz wirksam ist, liefert \$AC_CTOL die Wurzel aus der Summe der Quadrate der Toleranzen der Geometrieachsen.
- **\$AC_OTOL**
Kanalspezifische Orientierungstoleranz, die bei der Aufbereitung des aktuellen Hauptaufsatzes wirksam war.
Wenn 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>]**
Achsspezifische Konturtoleranz, die bei der Aufbereitung des aktuellen Hauptaufsatzes wirksam war.
Wenn eine Konturtoleranz aktiv ist, liefert \$AA_ATOL[<Geometrieachse>] die Konturtoleranz geteilt durch die Wurzel aus der Anzahl der Geometrieachsen.
Wenn eine Orientierungstoleranz und eine Orientierungstransformation aktiv ist, 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 Toleranzen der einzelnen Funktionen zu unterscheiden.

Solche Fälle können auftreten, wenn die Maschinen- und Settingdaten unterschiedliche Toleranzen für Kompressor-Funktionen, Überschleifen und Orientierungsglättung einstellen. Die Systemvariablen liefern dann den größten Wert, der bei den gerade aktiven Funktionen auftritt. Wenn z. B. eine Kompressor-Funktion mit Orientierungstoleranz $0,1^\circ$ und eine Orientierungsglättung ORISON mit 1° aktiv ist, liefert \$AC_OTOL den Wert "1". Wird die Orientierungsglättung ausgeschaltet, liefert \$AC_OTOL den Wert "0,1".

Lesen ohne Vorlaufstopp

Über folgende Systemvariablen sind im Teileprogramm die aktuell wirksamen Toleranzen lesbar:

- \$P_CTOL
Aktuell wirksame kanalspezifische Konturtoleranz.
- \$P_OTOL
Aktuell wirksame kanalspezifische Orientierungstoleranz.
- \$PA_ATOL
Aktuell wirksame achsspezifische Konturtoleranz.

Randbedingungen

Die mit CTOL, OTOL und ATOL programmierten Toleranzen wirken auch auf Funktionen, die indirekt von diesen Toleranzen abhängen:

- Die Begrenzung des Sehnenfehlers bei der Sollwertberechnung
- Die Grundfunktionen des Freiformflächenmodus

Folgende Überschleif-Funktionen werden durch die Programmierung von CTOL, OTOL und ATOL **nicht** beeinflusst:

- Überschleifen der Orientierung mit OSD
OSD verwendet keine Toleranz, sondern eine Distanz zum Satzübergang.
- Überschleifen mit G644
G644 dient nicht der Bearbeitung, sondern der Optimierung von Werkzeugwechseln und anderen Bewegungen ohne Bearbeitung.
- Überschleifen mit G645
G645 verhält sich fast immer wie G642 und verwendet somit die programmierten Toleranzen. Nur an tangentialen Satzübergängen mit einem Krümmungssprung, z. B. einem tangentialen Kreis-Gerade-Übergang, wird der Toleranzwert aus dem Maschinendatum MD33120 \$MA_PATH_TRANS_POS_TOL verwendet. Der Überschleifweg kann sich an diesen Stellen nämlich auch auf der Außenseite der programmierten Kontur befinden, wo viele Anwendungen weniger tolerant sind. Außerdem genügt in der Regel eine kleine, fest eingestellte Toleranz zum Ausgleich der Krümmungsänderungen, über die sich der NC-Programmierer keine Gedanken machen muss.

5.7 Eilgangbewegungen

5.7.1 Funktion

5.7.1.1 Eilgang

Im Eilgang werden programmierte Werkzeugbewegungen mit der größtmöglichen Verfahrensgeschwindigkeit ausgeführt.

Die Eilganggeschwindigkeit ist für jede Achse getrennt festgelegt (siehe "Parametrierung (Seite 519)").

Anwendung

Eilgangbewegungen werden z. B. für folgende Aufgaben eingesetzt:

- Schnelles Positionieren des Werkzeugs
- Umfahren des Werkstücks
- Anfahren von Werkzeugwechsellpunkten
- Freifahren des Werkzeugs

Hinweis

Eilgangbewegungen eignen sich nicht zur Werkstückbearbeitung!

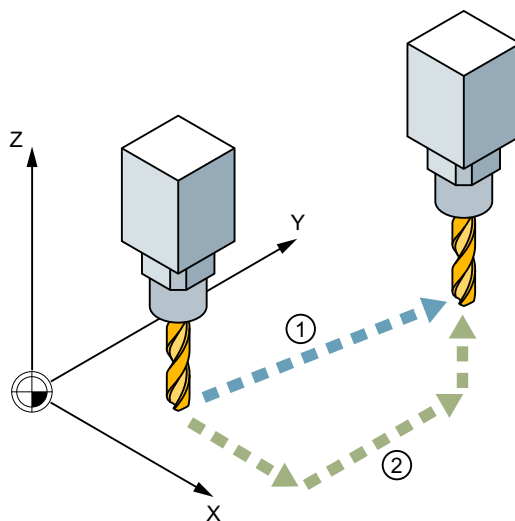
Aktivierung

Eilgang wird durch Programmierung von G0 im Teileprogramm aktiviert (siehe "Programmierung (Seite 520)").

5.7.1.2 Interpolationsverhalten der Bahnachsen bei Eilgangbewegungen

Lineare / nicht-lineare Interpolation

Bahnachsen im Eilgang können wahlweise mit linearer oder nicht-linearer Interpolation bewegt werden.



- ① Bahnweg bei Eilgang mit linearer Interpolation
- ② Einzelachsenbewegungen bei Eilgang mit nicht-linearer Interpolation

Lineare Interpolation

Eigenschaften:

- Die Bahnachsen werden gemeinsam interpoliert.
- Die mit G0 programmierte Werkzeugbewegung wird mit der größtmöglichen Verfahrgeschwindigkeit (Eilgang) ausgeführt.
- Die Eilganggeschwindigkeit ist für jede Achse getrennt festgelegt.
- Wird die Eilgangbewegung gleichzeitig in mehreren Achsen ausgeführt, so wird die Eilganggeschwindigkeit durch die Achse bestimmt, die für ihren Bahnweganteil die meiste Zeit benötigt.

In folgenden Fällen wird immer eine lineare Interpolation ausgeführt:

- Bei einer G-Befehl-Kombination mit G0, die eine Positionierachsbewegung **nicht** zulässt, z. B.:
G40, G41, G42, G96, G961 und MD20750 \$MC_ALLOW_G0_IN_G96 == FALSE
- Bei der Kombination G0 mit G64.
- Bei aktivem Kompressor oder bei einer aktiven Transformation.
- Bei Punkt-zu-Punkt(PTP)-Fahren.
- Bei angewähltem Konturhandrad (FD = 0).
- Bei aktivem Frame mit Rotation der Geometrieachsen.
- Bei Nibbeln aktiv mit Geometrieachsen.

Nicht-lineare Interpolation

Eigenschaften:

- Jede Bahnachse interpoliert als Einzelachse (Positionierachse) unabhängig von den anderen Achsen mit der achsspezifischen Eilganggeschwindigkeit.
- Kanalspezifisches "Restweg löschen" über PLC und über Synchronaktion wirkt auf alle Positionierachsen, die als Bahnachsen programmiert wurden.

Bei nicht-linearer Interpolation gilt bezüglich des achsspezifischen Rucks alternativ eine der beiden folgenden Einstellungen:

- Positionierachsbefehle BRISKA, SOFTA, DRIVEA
- Maschinendaten:
 - MD32420 \$MA_JOG_AND_POS_JERK_ENABLE
 - MD32430 \$MA_JOG_AND_POS_MAX_JERK

Die Systemvariablen, die sich auf den Bahnrestweg beziehen (\$AC_PATH, \$AC_PLTBB und \$AC_PLTEB), werden unterstützt.

 **VORSICHT**

Kollisionsgefahr

Da sich die Werkzeugbewegung bei nicht-linearer Interpolation von der Werkzeugbewegung bei linearer Interpolation unterscheiden kann, werden Synchronaktionen, die sich auf Koordinaten der Bahnbewegung beziehen, ggf. nicht aktiv.

Auswahl der Interpolationsart

Die Interpolationsart, die bei Eilgangbewegungen wirksam sein soll, wird über ein Maschinendatum voreingestellt (siehe "Parametrierung (Seite 519)").

Unabhängig von der Voreinstellung kann das gewünschte Interpolationsverhalten auch im Teileprogramm eingestellt werden (siehe "Lineare Interpolation für Eilgangbewegungen ein-/ ausschalten (RTLION, RTLIOF) (Seite 522)").

5.7.1.3 Toleranzen bei Eilgangbewegungen

Toleranzen für Eilgangbewegungen (G0-Toleranzen) können unterschiedlich zu den Werkstückbearbeitungstoleranzen eingestellt werden.

Vorteil

Größere G0-Toleranzen ermöglichen ein schnelleres Verfahren von G0-Sätzen.

Voraussetzungen

G0-Toleranzen werden nur wirksam, wenn folgende Bedingungen erfüllt sind:

- Eine der folgenden Funktionen ist aktiv:
 - Kompressorfunktion COMPON, COMPCURV, COMPCAD oder COMPSURF
 - Überschleiffunktion G642 oder G645
 - Orientierungsüberschleifen OST
 - Orientierungsglättung ORISON
 - Glättung bei bahnrelativer Orientierung ORIPATH
- Im Teileprogramm folgen mehrere (≥ 2) G0-Sätze aufeinander.
Bei einem einzelnen G0-Satz werden G0-Toleranzen nicht wirksam, da beim Übergang von einer Nicht-G0-Bewegung zu einer G0-Bewegung (und umgekehrt) grundsätzlich die "**kleinere Toleranz**" (Werkstückbearbeitungstoleranz) wirkt!

Einstellung der G0-Toleranzen

G0-Toleranzen werden kanalspezifisch über Maschinendaten voreingestellt (siehe "Parametrierung (Seite 519)").

Die voreingestellten G0-Toleranzen können durch Programmierung im Teileprogramm vorübergehend angepasst werden (siehe "Relative G0-Toleranz anpassen (STOLF) (Seite 523)").

5.7.1.4 Eilgangkorrektur

Mit dem Eilgang-Override-Schalter an der Maschinensteuertafel kann der Bediener vor Ort und mit sofortiger Wirkung die Eilganggeschwindigkeit prozentual verringern.

Eine wirksame Eilgangkorrektur wirkt auf alle Bahnachsen, die im Eilgang mit linearer oder nicht-linearer Interpolation verfahren und dem aktuellen Kanal zugeordnet sind.

Weitere Informationen siehe Funktionshandbuch *Achsen und Spindeln*, Kapitel *Vorschübe*.

5.7.2 Inbetriebnahme

5.7.2.1 Parametrierung

Eilganggeschwindigkeit

Die Eilganggeschwindigkeit entspricht der maximal zulässigen Achsgeschwindigkeit, die über das folgende Maschinendatum für jede Achse festgelegt wird:

MD32000 \$MA_MAX_AX_VELO (maximale Achsgeschwindigkeit)

Interpolationsverhalten bei Eilgangbewegungen

Das Interpolationsverhalten bei Eilgangbewegungen wird kanalspezifisch voreingestellt über das Maschinendatum:

MD20730 \$MC_G0_LINEAR_MODE = <Wert> (Interpolationsverhalten bei G0)

<Wert>	Bedeutung
0	Bei Eilgang (G0) ist die nicht-lineare Interpolation aktiv. Bahnachsen werden als Positionierachsen verfahren.
1	Bei Eilgang (G0) ist die lineare Interpolation aktiv. Die Bahnachsen werden gemeinsam interpoliert.

G0-Toleranzen

Relative G0-Toleranz

Toleranzen für Eilgangbewegungen können über einen kanalspezifischen Toleranzfaktor relativ zu den Werkstückbearbeitungstoleranzen eingestellt werden:

MD20560 \$MC_G0_TOLERANCE_FACTOR (Toleranzfaktor für G0)

5.7 Eilgangbewegungen

Der Toleranzfaktor kann sowohl größer als auch kleiner 1.0 sein. Ist der Faktor gleich 1.0 (Standardwert), sind für Eilgangbewegungen dieselben Toleranzen wirksam wie für Nicht-Eilgangbewegungen. Im Normalfall wird der Toleranzfaktor > 1.0 eingestellt.

Absolute G0-Toleranzen

Alternativ sind die kanalspezifischen Toleranzen für Eilgangbewegungen auch absolut und getrennt für Kontur- und Orientierungstoleranz einstellbar:

- MD20561 \$MC_G0_TOLERANCE_CTOL_ABS (Konturtoleranz bei G0-Bewegungen) = <Wert>

<Wert>	Bedeutung
= 0 (Default)	Die Konturtoleranz bei G0-Bewegungen wird durch die relative G0-Toleranz bestimmt.
> 0	Die Konturtoleranz bei G0-Bewegungen wird durch den Wert von MD20651 festgelegt.

- MD20562 \$MC_G0_TOLERANCE_OTOL_ABS (Orientierungstoleranz bei G0-Bewegungen) = <Wert>

<Wert>	Bedeutung
= 0 (Default)	Die Orientierungstoleranz bei G0-Bewegungen wird durch die relative G0-Toleranz bestimmt.
> 0	Die Orientierungstoleranz bei G0-Bewegungen wird durch den Wert von MD20652 festgelegt.

5.7.3 Programmierung

5.7.3.1 Eilgang aktivieren (G0)

Das Verfahren der Bahnachsen mit Eilganggeschwindigkeit wird eingeschaltet mit dem G-Befehl G0.

Syntax

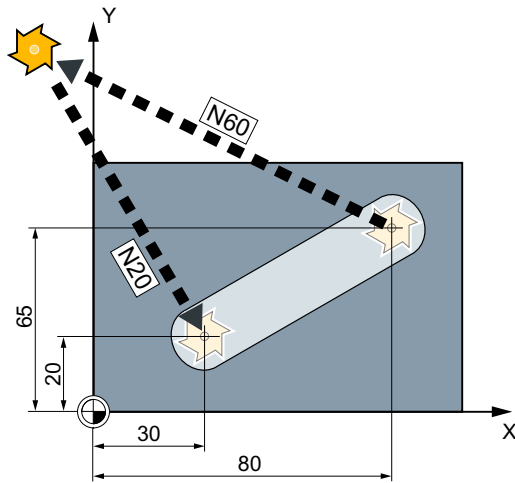
```
G0 X... Y... Z...
G0 RP=... AP=...
```

Bedeutung

G0:	Verfahren der Achsen mit Eilganggeschwindigkeit	
	Wirksamkeit:	modal
X... Y... Z...:	Angabe des Endpunkts in kartesischen Koordinaten	
RP=... AP=... :	Angabe des Endpunkts in Polarkoordinaten	

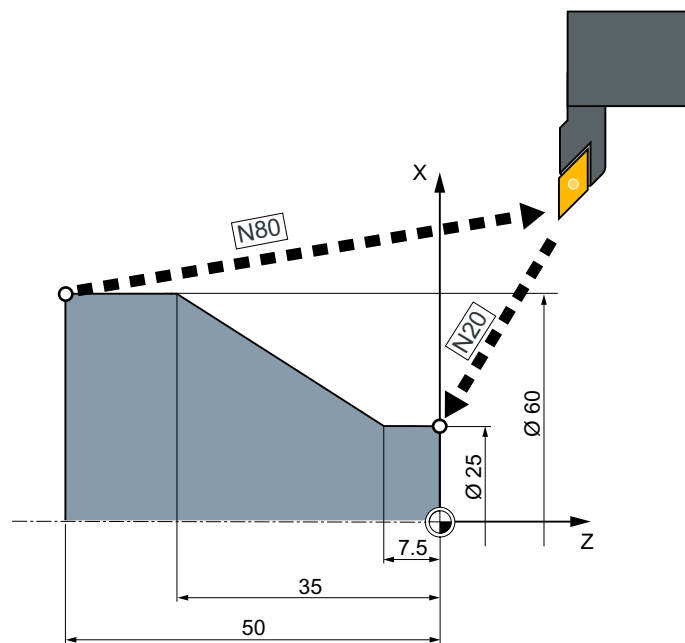
Beispiele

Beispiel 1: Fräsen



Programmcode	Kommentar
N10 G90 S400 M3	; Absolutmaßeingabe, Spindel rechts
N20 G0 X30 Y20 Z2	; Anfahren der Startposition
N30 G1 Z-5 F1000	; Zustellen des Werkzeugs
N40 X80 Y65	; Fahren auf einer Geraden
N50 G0 Z2	
N60 G0 X-20 Y100 Z100 M30	; Werkzeug freifahren, Programmende

Beispiel 2: Drehen



5.7 Eilgangbewegungen

Programmcode	Kommentar
N10 G90 S400 M3	; Absolutmaßeingabe, Spindel rechts
N20 G0 X25 Z5	; Anfahren der Startposition
N30 G1 G94 Z0 F1000	; Zustellen des Werkzeugs
N40 G95 Z-7.5 F0.2	
N50 X60 Z-35	; Fahren auf einer Geraden
N60 Z-50	
N70 G0 X62	
N80 G0 X80 Z20 M30	; Werkzeug Freifahren, Programmende

5.7.3.2 Lineare Interpolation für Eilgangbewegungen ein-/ausschalten (RTLION, RTLIOf)

Unabhängig von der Voreinstellung (MD20730 \$MC_G0_LINEAR_MODE) kann das Interpolationsverhalten bei Eilgangbewegungen mit den Befehlen der G-Gruppe 55 auch im Teileprogramm eingestellt werden.

Syntax

```

RTLIOf
...
RTLION
    
```

Bedeutung

RTLIOf:	G-Befehl zum Ausschalten der linearen Interpolation ⇒ Bei Eilgang (G0) ist die nicht-lineare Interpolation aktiv. Alle Bahnachsen erreichen unabhängig voneinander ihren Endpunkt.	
	Wirksamkeit:	modal
RTLION:	G-Befehl zum Einschalten der linearen Interpolation ⇒ Bei Eilgang (G0) ist die lineare Interpolation aktiv. Alle Bahnachsen erreichen gleichzeitig ihren Endpunkt.	
	Wirksamkeit:	modal

Hinweis**Voraussetzungen für RTLIOF**

Damit bei RTLIOF **nicht-linear** interpoliert wird, müssen folgende Voraussetzungen erfüllt sein:

- Keine Transformation (TRAORI, TRANSMIT, ...) aktiv.
- G60 aktiv (Anhalten am Satzende).
- Kein Kompressor aktiv (COMPOF).
- Keine Werkzeugradiuskorrektur aktiv (G40).
- Kein Konturhandrad angewählt.
- Kein Nibbeln aktiv.

Ist eine dieser Voraussetzungen nicht erfüllt, wird wie bei RTLION linear interpoliert.

Beispiel

Programmcode	Kommentar
	; Lineare Interpolation ist voreingestellt:
	; MD20730 \$MC_GO_LINEAR_MODE == TRUE
...	
N30 RTLIOF	; Lineare Interpolation ausschalten.
N40 G0 X0 Y10	; G0-Sätze werden mit nicht-linearer Interpolation verfahren.
N50 G41 X20 Y20	; WRK aktiv ⇒ G0-Sätze werden mit linearer Interpolation verfahren.
N60 G40 X30 Y30	; WRK nicht aktiv ⇒ G0-Sätze werden mit nicht-linearer Interpolation verfahren.
N70 RTLION	; Lineare Interpolation einschalten.
...	

Weitere Informationen**Lesen des aktuellen Interpolationsverhaltens**

Über die Systemvariablen \$AA_G0MODE kann das aktuelle Interpolationsverhalten gelesen werden.

5.7.3.3 Relative G0-Toleranz anpassen (STOLF)

Der über das Maschinendatum MD20560 \$MC_GO_TOLERANCE_FACTOR projizierte Toleranzfaktor für Eilgangbewegungen (G0-Toleranzfaktor) kann im Teileprogramm temporär angepasst werden. Die Einstellung im Maschinendatum wird dabei nicht verändert. Nach Kanal- bzw. Programmende-Reset wird wieder die projizierte Toleranz wirksam.

Voraussetzungen

Die relative G0-Toleranz wird nur wirksam, wenn folgende Bedingungen erfüllt sind:

- Eine der folgenden Funktionen ist aktiv:
 - Kompressorfunktion COMPON, COMPCURV, COMPCAD oder COMPSURF
 - Überschleiffunktion G642 oder G645
 - Orientierungsüberschleifen OST
 - Orientierungsglättung ORISON
 - Glättung bei bahnrelativer Orientierung ORIPATH
- Im Teileprogramm folgen mehrere (≥ 2) G0-Sätze aufeinander.
Bei einem einzelnen G0-Satz werden G0-Toleranzen nicht wirksam, da beim Übergang von einer Nicht-G0-Bewegung zu einer G0-Bewegung (und umgekehrt) grundsätzlich die "**kleinere Toleranz**" (Werkstückbearbeitungstoleranz) wirkt!
- Es sind keine absoluten G0-Toleranzen projiziert (\neq Standardeinstellung):
MD20561 \$MC_G0_TOLERANCE_CTOL_ABS (Konturtoleranz bei G0-Bewegungen) = 0
MD20562 \$MC_G0_TOLERANCE_OTOL_ABS (Orientierungstoleranz bei G0-Bewegungen) = 0

Syntax

STOLF=<Value>

Bedeutung

STOLF:	Adresse zum Programmieren eines temporär wirksamen G0-Toleranzfaktors		
	<Value>:	G0-Toleranzfaktor	
		Typ:	REAL
		Wert:	> 0:
	≤ 0:	Löschen des programmierten Toleranzfaktors => Es gilt wieder der im Maschinendatum voreingestellte Toleranzfaktor.	

Beispiel

Programmcode	Kommentar
COMPCAD G645 G1 F10000	; Kompressor-Funktion COMPCAD
X... Y... Z...	; Hier wirken die Maschinen- und Settingdaten.

Programmcode	Kommentar
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 0,02 mm.
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	
X... Y... Z...	; Ab hier wirkt ein G0-Toleranzfaktor von 4, also eine Konturtoleranz von 0,08 mm.
...	

Weitere Informationen

Lesen des G0-Toleranzfaktors

Der im Teileprogramm bzw. im aktuellen IPO-Satz wirksame Toleranzfaktor für Eilgangbewegungen 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 im Maschinendatum projektierten Wert.

Ist in einem Satz kein Eilgang (G0) aktiv, dann liefern diese Systemvariablen immer den Wert 1.

Ist ein absoluter Wert für die Konturtoleranz bei G0 aktiv, dann geben diese beiden Variablen den Faktor zwischen der Konturtoleranz bei G0-Bewegungen und der Konturtoleranz bei Nicht-G0-Bewegungen zurück.

Lesen der absoluten G0-Toleranzen

Die im Teileprogramm bzw. im aktuellen IPO-Satz wirksame absolute Kontur- und Orientierungstoleranz für Eilgangbewegungen sind über Systemvariable lesbar.

- In Synchronaktionen oder mit Vorlauf-Stopp im Teileprogramm über die Systemvariablen:
 - \$AC_CTOL_G0_ABS
 - \$AC_OTOL_G0_ABS
- Ohne Vorlauf-Stopp im Teileprogramm über die Systemvariablen:
 - \$P_CTOL_G0_ABS
 - \$P_OTOL_G0_ABS

5.8 RESET-Verhalten

MD20150

Durch Reset (Kanal- oder BAG-Reset) wird für alle G-Gruppen die kanalspezifisch parametrisierte Grundstellung wirksam:

MD20150 \$MC_GCODE_RESET_VALUES (Löschstellung der G-Gruppen)

Bezüglich "Bahnsteuerbetrieb, Genauhalt, LookAhead" sind folgende G-Gruppen relevant:

- Gruppe 10: Genauhalt - Bahnsteuerbetrieb
- Gruppe 12: Satzwechselkriterium bei Genauhalt
- Gruppe 21: Beschleunigungsprofil
- Gruppe 30: NC-Satz-Kompression
- Gruppe 59: Dynamikmodus für Bahninterpolation

Ausführliche Informationen zur Einstellung der Grundstellungen siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 29)".

5.9 Randbedingungen

5.9.1 Satzwechsel und Positionierachsen

Werden in einem Teileprogramm Bahnachsen im Bahnsteuerbetrieb verfahren, können parallel dazu verfahrenende Positionierachsen sowohl das Verhalten der Bahnachsen als auch den Satzwechsel beeinflussen.

Eine ausführliche Beschreibung der Positionierachsen findet sich in:

Weitere Informationen

Funktionshandbuch Achsen und Spindeln; Positionierachsen

5.9.2 Satzwechselverzögerung

Auch wenn für alle im Teileprogrammsatz verfahrenen Bahn- und Zusatzachsen die jeweils spezifischen Satzwechselkriterien erfüllt sind, kann es aufgrund anderweitiger nicht erfüllter Bedingungen und / oder aktiver Funktionen zu zeitlichen Verzögerungen des Satzwechsels kommen.

Beispiele:

- Fehlende Hilfsfunktionsquittierung durch die PLC
- Nicht vorhandene Folgesätze
- Aktive Funktion "Zwischenspeicher leeren"

Auswirkungen

Wenn während des Bahnsteuerbetriebs ein Satzwechsel nicht durchgeführt werden kann, dann werden alle in diesem Teileprogrammsatz programmierten Achsen (außer satzübergreifend verfahrenende Zusatzachsen) angehalten. Dabei treten keine Konturfehler auf.

Durch das Anhalten der Bahnachsen **während der Bearbeitung** kann es zum Entstehen von Freischneidmarken auf der Werkstückoberfläche kommen.

5.10 Datenlisten

5.10.1 Maschinendaten

5.10.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10110	PLC_CYCLE_TIME_AVERAGE	Mittlere PLC-Quittierungszeit
10680	MIN_CONTOUR_SAMPLING_TIME	Minimale Kontur-Abtastzeit
10682	CONTOUR_SAMPLING_FACTOR	Konturabtastfaktor
10712	NC_USER_CODE_CONF_NAME_TAB	Liste umprojektierter NC-Befehle
12030	OVR_FACTOR_FEEDRATE	Bewertung des Bahnvorschub-Override-Schalters
12100	OVR_FACTOR_LIMIT_BIN	Begrenzung bei binärkodiertem Override-Schalter
18360	MM_EXT_PROG_BUFFER_SIZE	FIFO-Buffer Größe für Abarbeiten von extern (DRAM)

5.10.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20150	GCODE_RESET_VALUES	Löschstellung der Gruppen
20170	COMPRESS_BLOCK_PATH_LIMIT	Maximale Verfahränge eines komprimierbaren NC-Satzes

5.10 Datenlisten

Nummer	Bezeichner: \$MC_	Beschreibung
20171	SURF_BLOCK_PATH_LIMIT	Maximale Verfahrnlänge eines komprimierbaren NC-Satzes bei Kompression mit COMPSURF
20172	COMPRESS_VELO_TOL	Maximal erlaubte Abweichung des Bahnvorschubs bei Kompression
20173	SURF_VELO_TOL	Maximal erlaubte Abweichung des Bahnvorschubs bei Kompression mit COMPSURF
20400	LOOKAH_USE_VELO_NEXT_BLOCK	LookAhead Folgesatzgeschwindigkeit
20430	LOOKAH_NUM_OVR_POINTS	Anzahl der Override-Schalter-Eckwerte bei LookAhead
20440	LOOKAH_OVR_POINTS	Override-Schalter-Eckwerte bei LookAhead
20443	LOOKAH_FFORM	Aktivierung des erweiterten LookAhead
20450	LOOKAH_RELIEVE_BLOCK_CYCLE	Entlastungsfaktor für die Blockzykluszeit
20455	LOOKAH_FUNCTION_MASK	Sonderfunktionen des LookAhead
20460	LOOKAH_SMOOTH_FACTOR	Glättungsfaktor bei LookAhead
20462	LOOKAH_SMOOTH_WITH_FEED	Glättung mit programmierten Vorschub
20465	ADAPT_PATH_DYNAMIC	Adaption der Bahndynamik
20480	SMOOTHING_MODE	Verhalten des Überschleifens mit G64x
20482	COMPRESSOR_MODE	Arbeitsweise des Kompressors
20484	COMPRESSOR_PERFORMANCE	Kompressorleistung
20485	COMPRESS_SMOOTH_FACTOR	Glättungsfaktor bei Kompression mit COMPCAD für den jeweiligen Dynamikmodus
20486	COMPRESS_SPLINE_DEGREE	Spline-Grad bei Kompression mit COMPCAD für den jeweiligen Dynamikmodus
20487	COMPRESS_SMOOTH_FACTOR_2	Glättungsfaktor für Rundachsen bei Kompression mit COMPCAD für den jeweiligen Dynamikmodus
20488	SPLINE_MODE	Einstellung für Spline-Interpolation
20490	IGNORE_OVL_FACTOR_FOR_ADIS	G641/G642 unabhängig vom Überlastfaktor
20550	EXACT_POS_MODE	Genauhaltbedingungen bei G0/G1
20552	EXACT_POS_MODE_G0_TO_G1	Genauhaltkriterium bei Eilgangübergängen im Bahnsteuerbetrieb
20560	G0_TOLERANCE_FACTOR	G0-Toleranzfaktor
20561	G0_TOLERANCE_CTOL_ABS	Konturtoleranz bei G0-Bewegungen
20562	G0_TOLERANCE_OTOL_ABS	Orientierungstoleranz bei G0-Bewegungen
20600	MAX_PATH_JERK	Bahnbezogener Maximalruck
20602	CURV_EFFECT_ON_PATH_ACCEL	Einfluss der Bahnkrümmung auf Bahndynamik
20603	CURV_EFFECT_ON_PATH_JERK	Einfluss der Bahnkrümmung auf Bahnruck
20605	PREPDYN_SMOOTHING_FACTOR	Faktor zur Glättung der Krümmung
20606	PREPDYN_SMOOTHING_ON	Aktivierung der Glättung der Krümmung
20730	G0_LINEAR_MODE	Interpolationsverhalten bei G0
28060	MM_IPO_BUFFER_SIZE	Anzahl der NC-Sätze im IPO-Buffer (DRAM)
28070	MM_NUM_BLOCKS_IN_PREP	Anzahl der NC-Sätze für die Satzaufbereitung (DRAM)
28071	MM_NUM_SURF_LEVELS	Dimensionierung der Funktion COMPSURF (DRAM)
28072	MM_MAXNUM_SURF_GROUPS	Dimensionierung der Funktion COMPSURF bzgl. Achsgruppen (DRAM)

Nummer	Bezeichner: \$MC_	Beschreibung
28520	MM_MAX_AXISPOLY_PER_BLOCK	Maximale Anzahl Achspolynome pro Satz
28530	MM_PATH_VELO_SEGMENTS	Anzahl der Speicherelemente zur Begrenzung der Bahngeschwindigkeit im Satz
28533	MM_LOOKAH_FFORM_UNITS	Speicher für den erweiterten LookAhead
28540	MM_ARCLENGTH_SEGMENTS	Anzahl der Speicherelemente zur Darstellung der Bogenlängenfunktion pro Satz
28610	MM_PREPDYN_BLOCKS	Anzahl Sätze zur Geschwindigkeitspräparation

5.10.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
32000	MAX_AX_VELO	Maximale Achsgeschwindigkeit
32310	MAX_ACCEL_OVL_FACTOR	Überlastfaktor für axiale Geschwindigkeitssprünge
32431	MAX_AX_JERK	Maximaler axialer Ruck bei Bahnbewegung
32432	PATH_TRANS_JERK_LIM	Maximaler axialer Ruck am Satzübergang im Bahnsteuerbetrieb
32433	SOFT_ACCEL_FACTOR	Skalierung der Beschleunigungsbegrenzung bei SOFT
32434	G00_ACCEL_FACTOR	Skalierung der Beschleunigungsbegrenzung bei G00
32435	G00_JERK_FACTOR	Skalierung der axialen Ruckbegrenzung bei G00
32440	LOOKAH_FREQUENCY	Glättungsgrenzfrequenz bei LookAhead
33100	COMPRESS_POS_TOL	Maximale Abweichung bei Kompression
33120	PATH_TRANS_POS_TOL	Maximale Abweichung beim Überschleifen mit G645
35240	ACCEL_TYPE_DRIVE	Beschleunigungskennlinie DRIVE für Achsen Ein/Aus
36000	STOP_LIMIT_COARSE	Genauhalt grob
36010	STOP_LIMIT_FINE	Genauhalt fein
36012	STOP_LIMIT_FACTOR	Faktor Genauhalt grob/fein und Stillstandsüberwachung
36020	POSITIONING_TIME	Verzögerungszeit Genauhalt fein

5.10.2 Settingdaten

5.10.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42465	SMOOTH_CONTUR_TOL	Maximale Konturabweichung beim Überschleifen
42466	SMOOTH_ORI_TOL	Maximale Abweichung der Werkzeugorientierung beim Überschleifen
42470	CRIT_SPLINE_ANGLE	Ecken-Grenzwinkel bei Kompression mit COMPCAD
42471	MIN_CURV_RADIUS	Minimaler Krümmungsradius bei Kompression mit COMPCAD

5.10 Datenlisten

Nummer	Bezeichner: \$SC_	Beschreibung
42472	MIN_SURF_RADIUS	Minimaler Krümmungsradius bei Kompression mit COMPSURF
42473	ACTNUM_SURF_GROUPS	Anzahl Achsgruppen bei COMPSURF
42475	COMPRESS_CONTUR_TOL	Maximal erlaubte Konturabweichung bei Kompression
42476	COMPRESS_ORI_TOL	Maximale Abweichung der Werkzeugorientierung bei Kompression Hinweis: Nur bei aktiver Orientierungstransformation!
42477	COMPRESS_ORI_ROT_TOL	Maximale Abweichung der Werkzeugdrehung bei Kompression Hinweis: Nur bei 6-Achs-Maschinen mit drehbarem Werkzeug!
42676	ORI_SMOOTH_TOL	Toleranz zur Glättung der Orientierung beim Überschleifen
42678	ORISON_TOL	Toleranz zur Glättung der Orientierung

K5: Kanalübergreifende Programmkoordinierung und kanalweises Einfahren

6

6.1 Kanalübergreifende Programmkoordinierung

6.1.1 Kanalübergreifende 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 mit den nachfolgenden Koordinierungsbefehlen in den unterschiedlichen Kanälen koordiniert werden.

Voraussetzung

Alle an der Programmkoordinierung beteiligten Kanäle müssen zur **selben** Betriebsartengruppe (BAG) gehören:

```
MD10010 $MC_ASSIGN_CHAN_TO_MODE_GROUP[<Kanal>] = <BAG-Nummer>
```

Kanalname statt Kanalnummer

Als Parameter der vordefinierten Prozeduren der Programmkoordinierung können anstelle von Kanalnummern auch die in MD20000 \$MC_CHAN_NAME[<Kanalindex>] eingetragenen Kanalnamen verwendet werden. Die Verwendung der Kanalnamen in NC-Programmen muss zuerst freigegeben werden:

```
MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE
```

Hinweis

Mindestabstand zwischen Befehlen

Zwischen den Befehlen `INIT`, `START`, `WAITE`, `WAITM`, `SETM`, `CLEARM` und dem Befehl `WAITMC` müssen mindestens zwei Verfahrssätze Abstand gehalten werden. `WAITMC` ist ein ausführbarer Satz, wird aber zur Optimierung in den vorherigen Satz verschoben und dann als Satz gelöscht. `SETM` als Beispiel ist kein ausführbarer Satz und wird in den nächsten Satz verschoben, so dass bei einem Satz Abstand beide Befehle im mittleren Satz stehen würden. Da nur ein Satz möglich ist, wird bei einem Satz Abstand für `WAITMC` die Optimierung nicht durchgeführt.

Damit bremst das Programm ab und die Bearbeitung wird kurz angehalten.

Syntax

```
INIT(<ChanNr>, <Prog>, <AckMode>)  
START(<ChanNr>, <ChanNr>, ...)
```

6.1 Kanalübergreifende Programmkoordinierung

WAITM(<MarkNr>, <ChanNr>, <ChanNr>, ...)
 WAITE(<ChanNr>, <ChanNr>, ...)
 WAITMC(<MarkNr>, <ChanNr>, <ChanNr>, ...)
 SETM(<MarkNr>, <MarkNr>, ...)
 CLEARM(<MarkNr>, <MarkNr>, ...)

Bedeutung

INIT():	Vordefinierte Prozedur zur Anwahl des NC-Programms, das im angegebenen Kanal abgearbeitet werden soll
START():	Vordefinierte Prozedur zum Starten des im jeweiligen Kanal angewählten Programms
WAITM():	Vordefinierte Prozedur zum Warten auf das Erreichen einer Wartemarke in den angegebenen Kanälen Im eigenen Kanal wird die angegebene Wartemarke durch <code>WAITM</code> gesetzt. Der vorhergehenden Satz wird mit Genauhalt beendet. Die Wartemarke wird nach Synchronisation gelöscht. Gleichzeitig können max. 10 Marken pro Kanal gesetzt werden.
WAITE():	Vordefinierte Prozedur zum Warten auf das Programmende in einem oder mehreren anderen Kanälen
WAITMC(): ¹⁾	Vordefinierte Prozedur zum Warten auf das Erreichen einer Wartemarke in den angegebenen Kanälen Im Gegensatz zu <code>WAITM</code> wird das Abbremsen der Achsen auf Genauhalt nur eingeleitet, wenn die anderen Kanäle die Wartemarke noch nicht erreicht haben.
SETM(): ¹⁾	Vordefinierte Prozedur zum Setzen einer oder mehrerer Wartemarken für die Kanalkoordinierung Die Bearbeitung im eigenen Kanal wird davon nicht beeinflusst. <code>SETM</code> behält seine Gültigkeit über Kanal-Reset und NC-Start hinweg.
CLEARM(): ¹⁾	Vordefinierte Prozedur zum Löschen einer oder mehrerer Wartemarken der Kanalkoordinierung Die Bearbeitung im eigenen Kanal wird davon nicht beeinflusst. <code>CLEARM()</code> löscht alle Wartemarken im Kanal. <code>CLEARM(0)</code> löscht nur die Wartemarke "0". <code>CLEARM</code> behält seine Gültigkeit über Kanal-Reset und NC-Start hinweg.
<ChanNr>:	Kanalnummer Die Nummer des eigenen Kanals muss nicht angegeben werden. Typ: INT
<Prog>:	Absolute oder relative Pfadangabe (optional) + Programmname Typ: STRING Zu Pfadangabe siehe: Weitere Informationen Programmierhandbuch Arbeitsvorbereitung; Datei- und Programmverwaltung > Programmspeicher > Adressierung von Dateien des Programmspeichers

<AckMode>:	Quittierungsmodus (optional)	
	Typ:	CHAR
	Werte:	<p>"N" Ohne Quittung Die Programmbearbeitung wird nach Absenden des Kommandos fortgeführt. Der Absender wird nicht benachrichtigt, wenn das Kommando nicht erfolgreich ausgeführt werden kann.</p> <p>"S" Synchroner 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.</p>
<MarkNr>:	Nummer der Wartemarke Hinweis In einem mehrkanaligen System stehen maximal 100 Wartemarken zur Verfügung (Wartemarke 0 ... 99). In einem einkanaligen System steht nur die Wartemarke 0 zur Verfügung.	
1) Zur anwenderspezifischen Kommunikation und / oder Koordination von Kanälen können Wartemarken mittels <code>SETM / CLEARM</code> auch ohne Verwendung des bedingten Wartebefehles <code>WAITMC</code> eingesetzt werden. Die Wartemarken behalten dabei über Kanal-Reset und NC-Start hinaus ihre Werte bei.		

Beispiele

START mittels Kanalnamen aus MD20000

- Parametrierung

```
MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE
$MC_CHAN_NAME[ 0 ] = "BEARBEITUNG" ; Name für Kanal 1
$MC_CHAN_NAME[ 1 ] = "ZUFUEHRUNG" ; Name für Kanal 2
```

- Programmierung

Programmcode	Kommentar
START (BEARBEITUNG)	; Start von Kanal 1
START (ZUFUEHRUNG)	; Start von Kanal 2

START über lokale "Kanalnamen" und Anwendervariable

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

6.1 Kanalübergreifende Programmkoordination

Programmcode	Kommentar
chanNo1 = CHAN_1	; Zuweisung parametrierter Kanalnamen Kanal 1
chanNo2 = CHAN_2	; Zuweisung parametrierter 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")
```

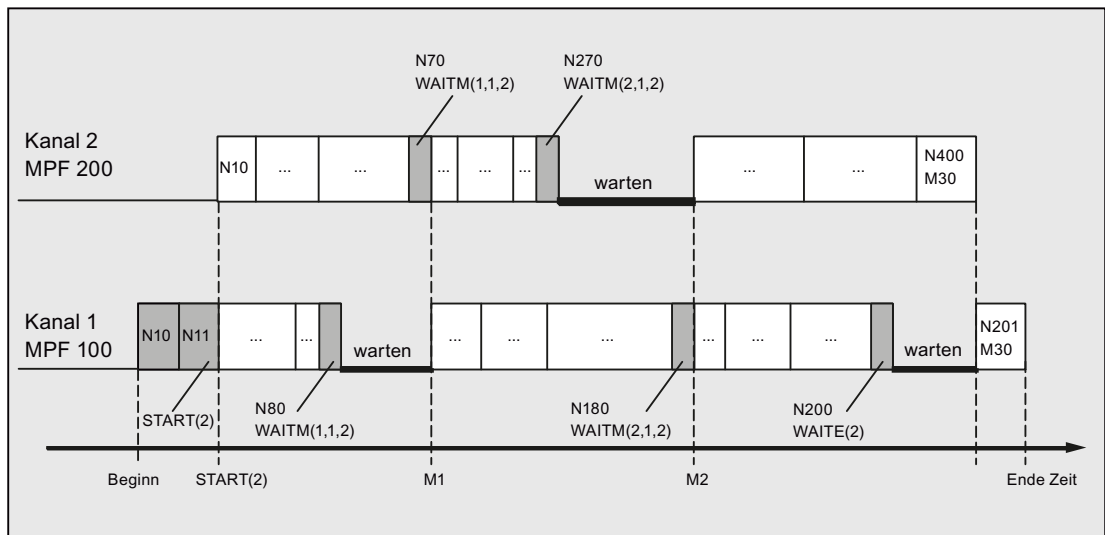
Programmkoordination mit WAITM

- 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 Kanalkoordination 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 6-1 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 GO X2=100	Kanal 2: Beginn des Abarbeitens von N220
• REPOSA		
<i>Ende</i>		
N110 GO X1=100		Kanal 1: Zeitversetzter Beginn des Abarbeitens von N110

Siehe auch

Bedingtes Warten (WAITMC) im Bahnsteuerbetrieb (Seite 537)

Programmierung: Achse holen (GET, GETD) (Seite 555)

6.1.2 Bedingtes Warten (WAITMC) im Bahnsteuerbetrieb

Sind in einem Kanal zum Bearbeitungszeitpunkt von `WAITMC` alle erforderlichen Wartemarken der anderen Kanäle bereits eingetroffen, wird in diesem Kanal die Verfahrbewegung nicht gebremst bzw. die weitere Programmbearbeitung nicht angehalten, sondern mit der Bearbeitung des Folgesatzes fortgefahren.

Als Funktionsaufruf steht `WAITMC` in einem eigenen Satz. Für ein korrektes Ausführen von `WAITMC` ohne Bremsen ist es erforderlich, dass unmittelbar vor `WAITMC` ein Verfahrersatz steht. Bei Verfahrersatzen mit `G64x` wird der Satz mit `WAITMC` dem Vorgängersatz hinzugefügt, so dass ein Geschwindigkeitseinbruch vermieden wird. Befinden sich keine Verfahrätze, z. B. "IF-Schleifen", unmittelbar vor `WAITMC`, kann der Befehl nicht konsistent zugeordnet werden und ein Abbremsen tritt auf.

Voraussetzung

Folgende Funktionen müssen im Kanal aktiv sein:

- Bahnsteuerbetrieb (G64, G641, G642, G643, G644 oder G645)
- Funktion "LookAhead"

Bremsverhalten**Kein Bremsen**

Beginnend mit dem Bewegungssatz vor dem Aufruf von `WAITMC` werden die Wartemarken der anderen zu synchronisierenden Kanäle geprüft. Liegen diese bereits alle vor, wird nicht gebremst:

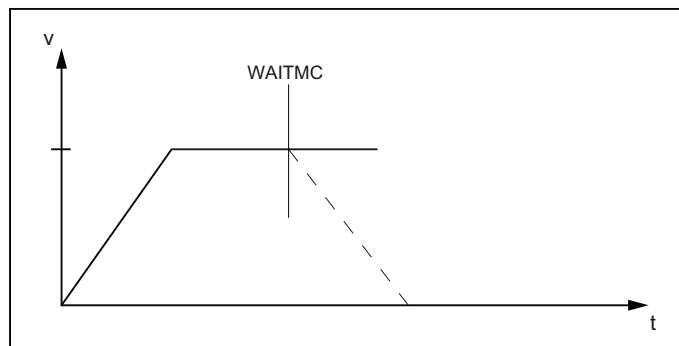


Bild 6-1 Verlauf der Bahngeschwindigkeit beim bedingten Warten mit `WAITCM`: Wartemarken aller Kanäle bereits vorhanden

Kurzzeitiges Bremsen

Fehlt die Wartemarke eines zu synchronisierenden Kanals, wird mit dem Bremsen begonnen. Während des Bremsens wird in jedem Interpolatorakt geprüft, ob die noch fehlenden

Wartemarken inzwischen eingetroffen sind. Ist dies der Fall, wird wieder auf die programmierte Geschwindigkeit beschleunigt:

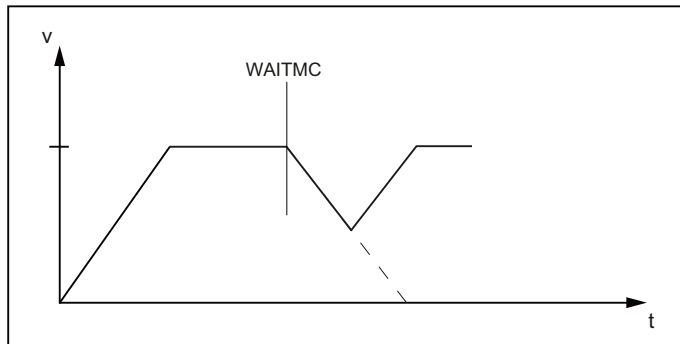


Bild 6-2 Verlauf der Bahngeschwindigkeit beim bedingten Warten mit WAITCM: Letzte Wartemarke kommt während des Bremsens

Bremsen bis Stillstand

Trifft die Wartemarke nicht während der Bremsphase ein, wird bis zum Stillstand gebremst und gewartet. Mit dem Eintreffen der Wartemarke wird wieder auf die programmierte Geschwindigkeit beschleunigt.

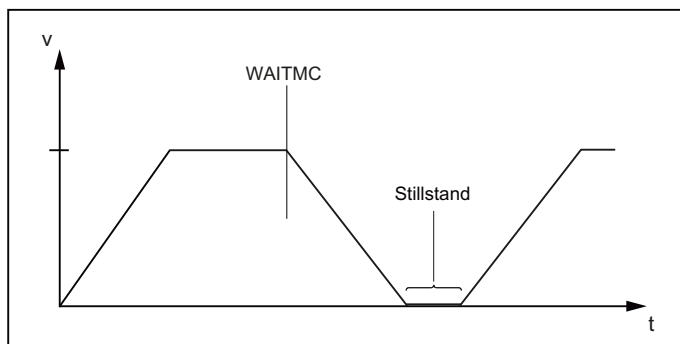


Bild 6-3 Verlauf der Bahngeschwindigkeit beim bedingten Warten mit WAITCM: Letzte Wartemarke kommt nach dem Bremsen

Satzwechsel während des Bremsens (IPOBRKA)

Bei aktivem Bewegungsendekriterium IPOBRKA (Satzwechselkriterium "Bremsrampe") wird bei Eintreffen der letzten fehlenden Wartemarke der nächste Satz eingewechselt und die Achse beschleunigt.

Beispiel 1: Bedingtes Warten im Bahnsteuerbetrieb

Das Beispiel ist schematisch und zeigt nur die für die Synchronisation relevanten Befehle:

Kanal 1

Programmcode	Kommentar
N10 INIT(2, "_N_200_MPF", "n")	; Anwählen Partnerprogramm Kanal 2.
N11 INIT(3, "_N_300_MPF", "n")	; Anwählen Partnerprogramm Kanal 3.
N15 START(2,3)	; Starten Programme in Kanal 2, 3.

Programmcode	Kommentar
...	; Bearbeitung in Kanal 1.
N20 WAITMC(7,2,3)	; Auf Marke 7 aus Kanälen 2 und 3 bedingt warten.
...	; Weitere Bearbeitung in Kanal 1.
N40 WAITMC(8,2)	; Auf Marke 8 aus Kanal 2 bedingt warten.
...	; Weitere Bearbeitung in Kanal 1.
N70 M30	; Ende Kanal 1.

Kanal 2

Programmcode	Kommentar
N200 ...	; Bearbeitung in Kanal 2.
N210 SETM(7)	; Kanal 2 setzt Wartemarke 7.
...	; Weitere Bearbeitung in Kanal 2.
N250 SETM(8)	; Kanal 2 setzt Wartemarke 8.
N260 M30	; Ende Kanal 2.

Kanal 3

Programmcode	Kommentar
N300 ...	; Bearbeitung in Kanal 3.
...	
N350 WHEN <Bedingung> DO SETM(7)	; Wartemarke in einer Synchronaktion setzen.
...	; Weitere Bearbeitungen in Kanal 3.
N360 M30	; Ende Kanal 3.

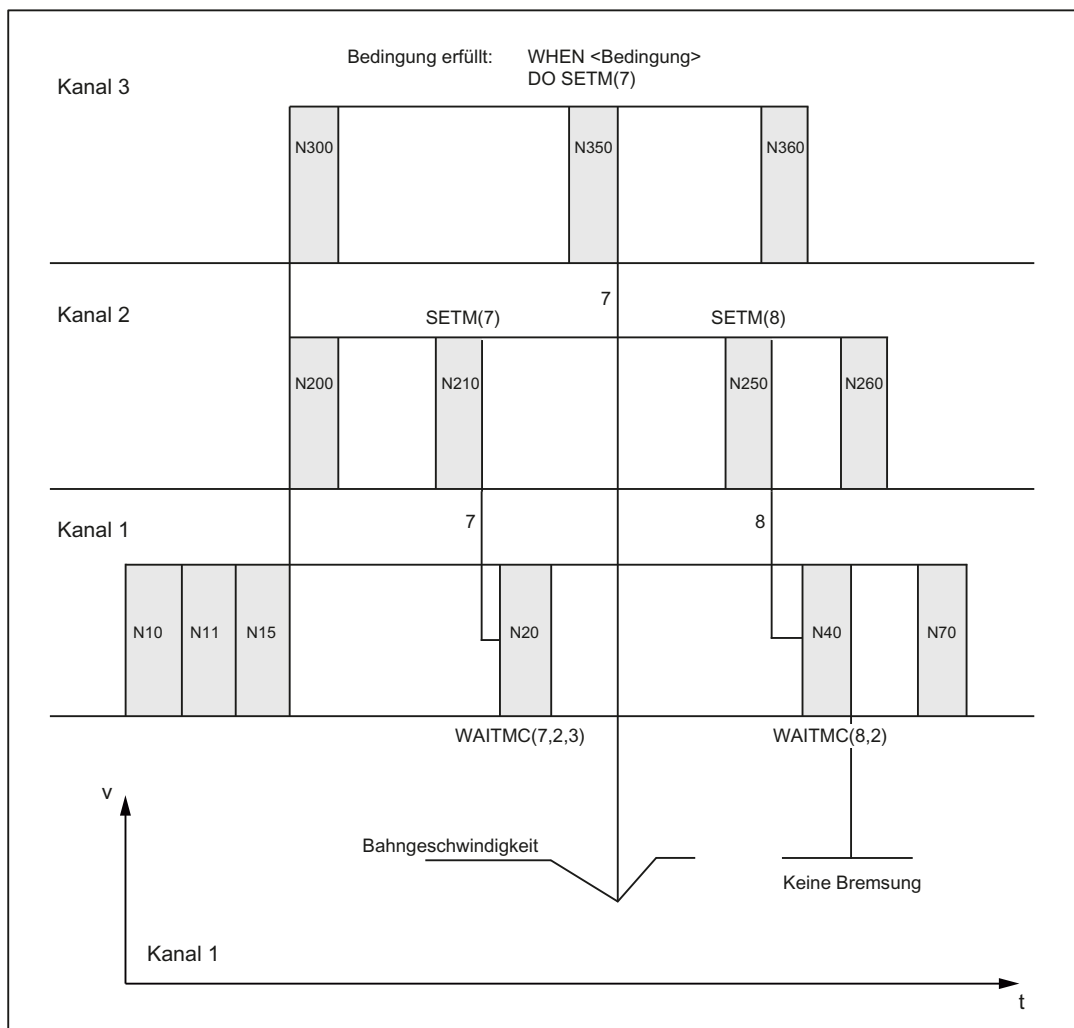


Bild 6-4 Bedingtes Warten im Bahnsteuerbetrieb mit drei beteiligten Kanälen (schematisch)

Beispiel 2: WAITMC und Einlesesperre

Die Hilfsfunktion M555 wird in Kanal 3 während des Fahrens ausgegeben und erzeugt eine Einlesesperre. Da WAITMC Satz N312 zugeordnet wird, ist die Wartemarke gesetzt und Kanal 2 fährt weiter. Die Programmbearbeitung im Kanal 3 bleibt wegen der Einlesesperre stehen.

Hinweis

WAITMC erzeugt bei aktivem G64 **keinen eigenen Satz**, sondern wird dem Vorgängersatz hinzugefügt. Bei aktivem Bahnsteuerbetrieb soll ein Geschwindigkeitseinbruch vermieden werden. Damit ist ein WAITMC schon erfüllt, wenn der Vorgängersatz z. B. mit Einlesesperre angehalten wird.

Kanal 2

Programmcode	Kommentar
N112 G18 G64 X200 Z200 F567	; Bearbeitung in Kanal 2

Programmcode	Kommentar
N120 WAITMC(1,2,3)	; Auf Wartemarke 1 aus Kanälen 2 und 3
	; bedingt warten
...	; Weitere Bearbeitung in Kanal 2, da das
	; WAITMC dem Satz N312 hinzugefügt wird
...	; Weitere Bearbeitung in Kanal 2
N170 M30	; Ende Kanal 2

Kanal 3

	Kommentar
	; Während des Fahrens Einleesperre M555
N300	; Bearbeitung in Kanal 3
N312 G18 G64 D1 X180 Z300 M555	
N320 WAITMC(1,2,3)	; Warten wegen Einleesperre

6.2 Kanalweises Einfahren

6.2.1 Funktion

Die Funktion "Kanalweises Einfahren" dient zum Testen bzw. Einfahren der NC-Programme eines von mehreren Kanälen, die untereinander synchronisiert sind. Die programmierten Verfahrbewegungen der Kanalachsen bzw. -spindeln in Kanälen, die sich im Zustand "Normalbetrieb" befinden, führen zu Verfahrbewegungen der Maschineachsen. Für die Kanäle, die sich im Zustand "Programmtest" befinden, gilt:

- Standardmäßig werden für die **Achsen** zum Normalbetrieb identische Sollwerte erzeugt, aber **nicht** an die Maschinenachsen **ausgegeben**.
- Standardmäßig werden für die **Spindeln** Sollwerte an die Maschinenachsen **ausgegeben**.
- Die angezeigten Istwerte der gesperrten Achsen / Spindeln werden intern aus den Sollwerten erzeugt.
- Die Befehle zur Kanalsynchronisation werden normal bearbeitet.
- Die NC/PLC-Nahtstellensignale werden normal bearbeitet.
- Die Bearbeitungszeit des Programms bleibt unverändert zum Normalbetrieb.

Der Zustand "Programmtest" kann temporär bei Bedarf kanal- oder achsspezifisch verlassen werden. Dadurch werden die Sollwerte wieder an die Maschinenachsen ausgegeben, sodass diese real an der Maschine verfahren.

6.2.2 Ablauf

Ein Kanal bewegt im Normalfall ein Werkzeug im Arbeitsraum. Bewegen mehrere Kanäle je ein Werkzeug im gleichen Arbeitsraum müssen die Werkzeugbewegungen zueinander synchronisiert werden. Folgende Synchronisationen sind möglich:

- Kanalsynchronisation über Programmkoordinierungsbefehle `WAITM`, `WAITMC`, `WAITE`, `START`.
- Kanalsynchronisation über PLC-Anwenderprogramm und NC/PLC-Nahtstellensignale. Beispielsweise über M-Funktionsausgabe von Kanal an PLC und Einlesesperre von PLC an Kanal.
- Achstausch: Der Kanal wartet, bis der andere Kanal die Achse abgibt.
- Synchronisation mittels globaler Variablen im NC-Programm.
- Kanalübergreifende Kopplungen
- Achscontainer-Drehung
- Test des Programms inkl. der parallelen Synchronaktionen im Hauptlauf und Synchronisation der Synchronaktionen mit dem Kanal.

Unter diesen Rahmenbedingungen ist es fast unmöglich, nur einen Kanal zu starten, er würde an der ersten Synchronisationsstelle stehen bleiben.

Mit der Funktion "Kanalweises Einfahren" können alle erforderlichen Kanäle so gestartet werden, dass nur der oder die Kanäle Fahrfreigaben bekommen, deren Zusammenspiel der programmierten Verfahrbewegungen getestet werden soll. Die anderen Kanäle befinden sich im Zustand "Programmtest".

Dazu müssen vom Maschinenbediener vor dem Start der Kanäle die Kanäle definiert werden, die unter "Programmtest" verfahren sollen. Dies erfolgt über die Bedienoberfläche im Menü "Programmbeeinflussung".

6.2.3 Einkanalansicht

Anwahl

Die Anwahl von "Programmtest" (PRT) für den im Grundbild "Maschine" angezeigten Kanal erfolgt über die Bedienoberfläche z. B. SINUMERIK Operate:

1. Softkey: Bedienbereich "Maschine" > "Programmbeeinflussung"
2. Menü "Programmbeeinflussung": Kontrollkästchen "Programmtest (PRT)" anwählen.

NC/PLC-Nahtstellensignale

Nach der Anwahl von Programmtest (PRT) sind folgende NC/PLC-Nahtstellensignale gesetzt (siehe unten Hinweis "Automatische Übertragung der Nahtstellensignale"):

- Kanäle
 - DB21, ... DBX25.7 == 1 (von HMI: Programmtest angewählt)
 - DB21, ... DBX1.7 == 1 (von PLC: Programmtest aktivieren)
 - DB21, ... DBX33.7 == 1 (von NC: Programmtest aktiv)
- Achsen
 - DB31, ... DBX128.0 == 0 (von HMI: Programmtest unterdrücken)
 - DB31, ... DBX128.1 == 0 (von HMI: Programmtest aktivieren)
 - DB31, ... DBX14.0 == 0 (von PLC: Programmtest unterdrücken)
 - DB31, ... DBX14.1 == 0 (von PLC: Programmtest aktivieren)
- Spindeln
 - DB31, ... DBX128.0 == 1 (von HMI: Programmtest unterdrücken)
 - DB31, ... DBX128.1 == 0 (von HMI: Programmtest aktivieren)
 - DB31, ... DBX14.0 == 1 (von PLC: Programmtest unterdrücken)
 - DB31, ... DBX14.1 == 0 (von PLC: Programmtest aktivieren)

Hinweis

Defaultzustand der Verfahrbewegungen

Der Defaultzustand bezüglich der Verfahrbewegungen nach Anwahl von "Programmtest" im Kanal ist:

- Achsen: Gesperrt
 - Spindeln: Freigegeben
-

Hinweis

Automatische Übertragung der Nahtstellensignale

Die HMI-Anforderungssignale DB21, ... DBX128.0 / .1 werden nur dann vom PLC-Grundprogramm auf die PLC-Anforderungssignale DB21, ... DBX14.0 / .1 übertragen, wenn der FB1-Parameter **MMCToIF** auf **TRUE** gesetzt ist. Ist der Parameter nicht gesetzt, müssen die PLC-Anforderungssignale vom PLC-Anwenderprogramm gesetzt werden.

Verfahrbewegungen der Spindeln sperren

Sollen Spindeln während "Programmtest" **nicht** verfahren, müssen sie über das PLC-Anwenderprogramm explizit gesperrt werden:

- DB31, ... DBX128.0 = 0 (Programmtest unterdrücken)
- DB31, ... DBX128.1 = 1 (Programmtest aktivieren)

- DB31, ... DBX14.0 = 0 (von PLC: Programmtest unterdrücken)
- DB31, ... DBX14.1 = 1 (von PLC: Programmtest aktivieren)

ACHTUNG

Verfahrbewegungen der Spindeln

Defaultmäßig sind die Verfahrbewegungen der Spindeln des Kanals im Zustand "Programmtest" freigegeben.

Verfahrbewegungen von Achsen freigeben

Sollen Achsen des Kanals während "Programmtest" verfahren, müssen sie über das PLC-Anwenderprogramm explizit freigegeben werden:

- DB31, ... DBX14.0 = 1 (von PLC: Programmtest unterdrücken)
- DB31, ... DBX14.1 = 0 (von PLC: Programmtest aktivieren)

Erlaubte Schaltzeitpunkte

- Kanal
Die Nahtstellensignale zum Aus- / Einschalten des kanalspezifischen Zustandes "Programmtest" (DB21, ... DBX25.7 bzw. DBX1.7) dürfen nur im Kanalzustand "Reset" oder "Unterbrochen" geschaltet werden.
- Achsen / Spindel
Die Nahtstellensignale zum Aus- / Einschalten des achsspezifischen Zustandes "Programmtest" (DB31, ... DBX128.0 / .1 bzw. DBX14.0 / .1) dürfen immer geschaltet werden.

6.2.4 Mehrkanalansicht

Voraussetzungen

Damit in der Mehrkanalansicht im Menü "Programmbeeinflussung" die Kanäle und Spindeln angezeigt werden, müssen folgende Voraussetzungen erfüllt sein:

- Kanäle
 - Es ist mehr als ein Kanal parametrier (MD10010 \$MN_ASSIGN_CHAN_TO_MODE_GROUP[<Kanal>] ≠ 0)
 - Option "programSYNC" (6FC5800-0AP05-0YB0) ist gesetzt / lizenziert
- Spindeln
 - Die Bedeutung der im Kanal parametrieren Spindel ist definiert (MD52206 \$MC_AXIS_USAGE[<Spindel>] ≠ 0)

Anwahl

Die Auswahl von "Programmtest" (PRT) für einen oder mehrere der in der Steuerung parametrisierten Kanäle erfolgt über die Bedienoberfläche z. B. SINUMERIK Operate:

1. Softkey: Bedienbereich "Maschine" > "Programmbeeinflussung"
2. Softkey: "Einfahren"
3. Menü "Programmbeeinflussung": Gruppe "Einfahren" > Kontrollkästchen "Kanal x" **abwählen**.

Kontrollkästchen Kanal / Spindel

Im Menü "Programmbeeinflussung": Gruppe "Einfahren" kann über die Kontrollkästchen "Kanal x" bzw. "Spindel x" folgendes Verhalten eingestellt werden:

Kontrollkästchen	Angewählt	Abgewählt
"Kanal x"	Normalbetrieb	Programm wird nur berechnet (Kanal im Testlauf)
"Spindel x"	Normalbetrieb	Keine Bearbeitung an Spindel x (Spindel im Testlauf)

NC/PLC-Nahtstellensignale

Nach Auswahl des Kontrollkästchens für den "Kanal x" befindet sich der Kanal im Zustand "Programmtest (PRT)". Folgende NC/PLC-Nahtstellensignale sind gesetzt (siehe unten Hinweis "Automatische Übertragung der Nahtstellensignale"):

- Kanäle
 - DB21, ... DBX25.7 == 1 (von HMI: Programmtest angewählt)
 - DB21, ... DBX1.7 == 1 (von PLC: Programmtest aktivieren)
 - DB21, ... DBX33.7 == 1 (von NC: Programmtest aktiv)
- Achsen
 - DB31, ... DBX128.0 == 0 (von HMI: Programmtest unterdrücken)
 - DB31, ... DBX128.1 == 0 (von HMI: Programmtest aktivieren)
 - DB31, ... DBX14.0 == 0 (von PLC: Programmtest unterdrücken)
 - DB31, ... DBX14.1 == 0 (von PLC: Programmtest aktivieren)

Defaultmäßig sind die Kontrollkästchen der Spindeln angewählt, D. h., die Spindeln sind im **Normalbetrieb**:

- Spindeln
 - DB31, ... DBX128.0 == 1 (von HMI: Programmtest unterdrücken)
 - DB31, ... DBX128.1 == 0 (von HMI: Programmtest aktivieren)
 - DB31, ... DBX14.0 == 1 (von PLC: Programmtest unterdrücken)
 - DB31, ... DBX14.1 == 0 (von PLC: Programmtest aktivieren)

Hinweis

Defaultzustand der Verfahrbewegungen

Der Defaultzustand bezüglich der Verfahrbewegungen nach Anwahl von "Programmtest" im Kanal ist:

- Achsen: Gesperrt
 - Spindeln: Freigegeben
-

Hinweis

Automatische Übertragung der Nahtstellensignale

Die HMI-Anforderungssignale DB21, ... DBX128.0 / .1 werden nur dann vom PLC-Grundprogramm auf die PLC-Anforderungssignale DB21, ... DBX14.0 / .1 übertragen, wenn der FB1-Parameter **MMCToIF** auf **TRUE** gesetzt ist. Ist der Parameter nicht gesetzt, müssen die PLC-Anforderungssignale vom PLC-Anwenderprogramm gesetzt werden.

Verfahrbewegungen der Spindeln sperren

Sollen Spindeln des Kanals während "Programmtest" **nicht** verfahren, müssen sie über das PLC-Anwenderprogramm explizit gesperrt werden:

- DB31, ... DBX128.0 = 0 (Programmtest unterdrücken)
- DB31, ... DBX128.1 = 1 (Programmtest aktivieren)
- DB31, ... DBX14.0 = 0 (von PLC: Programmtest unterdrücken)
- DB31, ... DBX14.1 = 1 (von PLC: Programmtest aktivieren)

ACHTUNG
Verfahrbewegungen der Spindeln
Defaultmäßig sind die Verfahrbewegungen der Spindeln des Kanals im Zustand "Programmtest" freigegeben.

Verfahrbewegungen von Achsen freigeben

Sollen Achsen des Kanals während "Programmtest" verfahren, müssen sie über das PLC-Anwenderprogramm explizit freigegeben werden:

- DB31, ... DBX14.0 = 1 (von PLC: Programmtest unterdrücken)
- DB31, ... DBX14.1 = 0 (von PLC: Programmtest aktivieren)

Erlaubte Schaltzeitpunkte

- Kanal
Die Nahtstellensignale zum Aus- / Einschalten des kanalspezifischen Zustandes "Programmtest" (DB21, ... DBX25.7 bzw. DBX1.7) dürfen nur im Kanalzustand "Reset" oder "Unterbrochen" geschaltet werden.
- Achsen / Spindel
Die Nahtstellensignale zum Aus- / Einschalten des achsspezifischen Zustandes "Programmtest" (DB31, ... DBX128.0 / .1 bzw. DBX14.0 / .1) dürfen immer geschaltet werden.

6.2.5 Systemvariablen

Der Zustand "Programmtest" ist über Systemvariablen abfragbar:

- Für die Anzeige in der Bedienoberfläche, in Synchronaktionen oder mit Vorlaufstopp im Teileprogramm über die Systemvariablen:

\$AC_ISTEST	Zustand "Programmtest" für den Kanal Liefert TRUE (1), wenn der Zustand "Programmtest" für den Kanal aktiv ist.
\$AA_ISTEST[<n>]	Zustand "Programmtest" für die Achse <n> Liefert TRUE (1), wenn der Zustand "Programmtest" für die Achse <n> aktiv ist.

- Ohne Vorlaufstopp im Teileprogramm über die Systemvariable:

\$P_ISTEST	Liefert TRUE (1), wenn der Zustand "Programmtest" für den Kanal aktiv ist.
------------	--

Beispiel

Der Kanal läuft unter "Programmtest" und die Achse "C" wurde mit "Programmtest unterdrücken" herausgenommen. Eine Abfrage über Systemvariablen liefert dann folgendes Ergebnis:

- \$AC_ISTEST == TRUE
- \$P_ISTEST == TRUE
- \$AA_ISTEST[C] == FALSE

6.2.6 Randbedingungen**Achstausch**

Die Funktion "Achstausch" ermöglicht es, dass eine Achse/Spindel in mehreren Kanälen bekannt ist und wechselweise von diesen programmiert werden kann (siehe Kapitel "K10: Kanalübergreifender Achstausch (Seite 553)").

Im Zusammenhang mit den Funktionen "Programmtest" und "Kanalweises Einfahren" ist beim Achstausch auf Folgendes zu achten:

- Befindet sich nur einer der Kanäle im Zustand "Programmtest", so wird die Tauschachse aus diesem Kanal genommen und in einen Kanal eingebracht, der sich nicht im Zustand "Programmtest" befindet. Für eine Tauschachse mit aktiver Achsensperre findet also beim Wechsel über die Kanäle mit/ohne Kanalzustand "Programmtest" kein Zustandswechsel in der Achse selbst statt (siehe Beispiel 3).
- Bei Programmtest werden bei Teileprogrammende/Reset alle Achsen/Spindeln, die nicht interpolieren, wieder auf die aktuelle Istposition synchronisiert. Dies hat zur Folge, dass bei einem Achstausch, der erst nach dem Programmende erfolgt, da die Achse erst beim Programmende den Kanal verlassen darf, die simulierte erreichte Position nicht an den aufnehmenden Kanal weitergegeben wird.

Hinweis

Die Programme sollten auch am Ende eine WAIT-Marke enthalten, um sich gleichzeitig zu beenden.

6.2.7 Beispiele

Beispiel 1: In einer 3-kanaligen Anlage soll Kanal 2 erprobt werden.

Testmöglichkeit 1: Programmtest ohne SERUPRO

1. Der Anwender überlegt, welche Achsen/Spindeln real verfahren werden sollen. Für diese Achsen wird "Programmtest unterdrücken" gesetzt.
2. Für den Kanal 1 und 3 wird der Zustand "Programmtest" angewählt.
3. Die Kanäle 1, 2, 3 werden über die PLC gestartet.
4. Nach dem Programmende kann "Programmtest" wieder abgewählt werden.
5. Falls die jetzt aktuelle Einstellung von "Programmtest unterdrücken" auch für andere Situationen (Kanal 1 oder Kanal 3 soll erprobt werden) sinnvoll ist, kann dieses Signal gesetzt bleiben. Dies wird sicher in vielen Fällen sinnvoll sein.

Testmöglichkeit 2: Programmtest mit SERUPRO

1. Der Anwender überlegt, welche Achsen/Spindeln real verfahren werden sollen. Für diese Achsen wird "Programmtest unterdrücken" gesetzt.
2. Für den Kanal 1 und 3 wird der Zustand "Programmtest" angewählt.
3. Die Kanäle 1, 2, 3 werden über die PLC gestartet.
4. Ein Fehler oder Alarm tritt auf, der Anwender bricht mit RESET ab.
5. SERUPRO auf die Unterbrechungsstelle aller 3 Kanäle.
6. Suchziel in allen 3 Kanälen ist erreicht.
7. Start aller 3 Kanäle.
8. Jetzt befinden sich Kanal 1 und 3 erneut in "Programmtest" und das "Kanalweise Einfahren" wird fortgesetzt.

Beispiel 2: Einschalten von "Programmtest unterdrücken"

Ein Kanal befindet sich im Programmtest. Im laufenden Betrieb soll "Programmtest unterdrücken" für die Achse "Y" ausgelöst werden (auf dem Satz N1010).

Programmcode	Kommentar
N1000 G0 Y1000	
N1010 G4 F10	
N1020 G0 G91 Y=10	; Inkrementell verfahren.
N1030 M30	

Das Programm fährt mit diesem Ablauf auf die Position 1010, d. h. der simulierte Anteil "1000" dieser Achse wird nach dem Einschalten von "Programmtest unterdrücken" verfahren.

Beispiel 3: Programmtest und Achstausch

Achse X1 aus Kanal 1 und Achse X2 aus Kanal 2 sind der 1. Maschinenachse AX1 der NC zugeordnet.

Kanal 1 mit "Programmtest"	Kanal 2 ohne "Programmtest"
N10010 G0 G90 X0	
N10020 X1=100	
N10030 WAITM(91,1,2)	
N10040 WAITM(92,1,2)	
N10050 M0	
N10060 M30	
	N20010 WAITM(91,1,2)
	N20020 G91 G0 X2=10
	N20030 WAITM(92,1,2)
	N20040 M0
	N20050 M30

Im Satz N20040 wird die Maschinenachse AX1 in den Kanal 2 getauscht, die letzte Position der Achse aus Kanal 1 übernommen und anschließend auf der Position 110 verfahren.

6.3 Randbedingungen

6.3.1 Betriebsart MDA: Bahnsteuerbetrieb und WAITMC

Betriebsart MDA: Bahnsteuerbetrieb und WAITMC

In der Betriebsart MDA darf beim Start der Abarbeitung des MDA-Satzpuffers der Befehl `WAITMC` im Zusammenhang mit Bahnsteuerbetrieb (G64 / G604), nicht im letzten Satz des MDA-Satzpuffers stehen. Ansonsten hält die Programmbearbeitung mit dem vorletzten Verfahrtsatz an und kann nur noch mit Reset abgebrochen werden.

6.3 Randbedingungen

Beispiel

Der MDA-Satzpuffer enthält vor NC-Start folgende Sätze:

Programmcode	Kommentar
N10 G64 G1 G94 F5000 X100	; Bahnsteuerbetrieb im ersten Satz des ; MDA-Satzpuffers
N20 X200	
N30 X300	; Vorletzter Verfahrersatz
N40 X400	; Die Programmbearbeitung hält hier
N50 WAITMC(...)	; wegen WAITMC in N50 an. ; Abbruch mit Reset notwendig!

6.3.2 Nicht synchroner Start der Verfahrbewegung nach WAIT-Befehlen

Bei einer Kanalkoordination mittels WAIT-Befehlen kann es zu einem nicht synchronen Beginn des Abarbeitens der Folgesätze kommen. Dieses Verhalten tritt auf, wenn unmittelbar vor Erreichen der gemeinsamen Synchronisationsstelle in einem der 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 Z1 und W
- Kanal 2: Achse Z2

Beispiel 1: Zeitversetzter Start der Verfahrbewegungen

Die folgenden Befehlsfolgen in den Kanälen 1 und 2 führen zu einem eventuell problematischen zeitversetzten Start der Verfahrbewegungen nach der Synchronisationsstelle N110 / N210:

Tabelle 6-2 Zeitlicher Ablauf in den Kanälen 1 und 2

Kanal 1	Kanal 2	Beschreibung
...	...	Beliebige Bearbeitung in Kanal 1 und 2
N100 W100		Kanal 1: Verfahren der Achse W
...	...	Beliebige Bearbeitung in Kanal 1 und 2
N110 WAITM(2,1,2)	N210 WAITM(2,1,2)	Kanal 1 und 2: warten auf Synchronisation
<i>Beginn "AUTO-GET(W)":</i> <ul style="list-style-type: none"> • Achstausch • Restweglöschen • REPOSA <i>Ende</i>	<i>"AUTOGET(W)" und warten auf Achse W von Kanal 1</i> N220 Z2=100 W100	Kanal 2: Zeitversetzter Start der Verfahrbewegung Kanal 1: Start der Verfahrbewegung
N120 Z1=100		

Beispiel 2: Zeitgleicher Start der Verfahrbewegungen

Der zeitversetzte Start der Verfahrbewegung im Beispiel 1 kann durch explizites Freigeben und Anfordern der zu tauschenden Achse W und durch Einfügen einer weiteren Synchronisationsstelle vermieden werden. Die folgenden Befehlsfolgen in den Kanälen 1 und 2 führen zu einem zeitgleichen Start der Verfahrbewegungen nach der Synchronisationsstelle N120 / N220:

Tabelle 6-3 Zeitlicher Ablauf in den Kanälen 1 und 2

Kanal 1	Kanal 2	Beschreibung
...	...	Beliebige Bearbeitung in Kanal 1 und 2
N100 W100		Kanal 1: Verfahren der Achse W
...	...	Beliebige Bearbeitung in Kanal 1 und 2
RELEASE (W)		Kanal 1: Freigabe der Achse W ⇒ Der Achstausch in den Kanal 2 hat keine Rückwirkung mehr auf Kanal 1
N110 WAITM(2,1,2)	N210 WAITM(2,1,2)	Kanal 1 und 2: warten auf Synchronisation
	N215 GET (W)	Kanal 2: Anfordern der Achse W. Der Achstausch erfolgt zeitoptimal, da Achse W bereits freigegeben ist
N120 WAITM(9,1,2)	N220 WAITM(9,1,2)	Kanal 1 und 2: warten auf Synchronisation
N130 Z1=100	N230 Z2=100 W100	Kanal 1 und 2: zeitgleicher Start der Verfahrbewegungen

K10: Kanalübergreifender Achstausch

7.1 Überblick

Hinweis Spindeln

Die nachfolgenden, im Zusammenhang mit der Funktion "Achstausch" für **Achsen** gemachte Aussagen und Funktionen, gelten auch für **Spindeln**.

Jede Achse muss im Rahmen der Steuerungsinbetriebnahme einem Kanal zugeordnet werden. Nur von diesem Kanal aus, dem die Achse zugeordnet ist, kann sie z.B. über Teileprogramme oder Synchronaktionen verfahren werden. Mit der Funktion "Achstausch" ist es möglich, eine Achse freizugeben und einem anderen Kanal zuzuordnen, d. h. die Achse zu tauschen. Erst dann kann die Achse vom anderen Kanal verfahren werden.

Achszustände

Im Rahmen der Funktion "Achstausch" kann eine Achse folgende Zuständen einnehmen:

- "Kanal-Achse"
Eine Kanal-Achse ist eine Achse die einem Kanal zugeordnet ist. Sie kann über ein Teileprogramm oder manuell verfahren werden.
- "PLC-Achse"
Eine PLC-Achse ist eine Achse, die der PLC zugeordnet ist. Sie kann nur vom PLC-Anwenderprogramm über Funktionsbaustein FC18 verfahren werden.
- "Neutrale Achse"
Eine Neutrale Achse ist eine Achse, die aktuell keinem Kanal oder der PLC zugeordnet ist. Vor dem Verfahren muss sie erst von einem Kanal oder der PLC angefordert werden.
- "Achse in anderem Kanal"
Eine Achse ist in diesem Zustand, wenn ein Kanal die Achse angefordert hat. Sie konnte ihm aber noch nicht zugeordnet werden, da sie noch von einem anderen Kanal belegt ist.

7.2 Inbetriebnahme

Parametrierung

NC-spezifische Maschinendaten

- Allgemeine Parameterierung des Achstauschverhaltens
MD10722 \$MN_AXCHANGE_MASK

Kanalspezifische Maschinendaten

- Parametrierung welche Achsen zum Kanal gehören bzw. Kanalachsen sind:
MD20070 \$MC_AXCONF_MACHAX_USED[<Kanalachse>] = <Maschinenachse>
Hinweis: Die Zuordnung aller in der NC verwendeten Achsen als Kanalachsen eines oder mehreren Kanälen muss, unabhängig von der Funktion "Achstausch", in jedem Fall durchgeführt werden.

Achsspezifische Maschinendaten

- Ist eine Achse Kanalachse in mehreren Kanälen, wird über das Maschinendatum festgelegt, welchem Kanal die Achse nach dem Hochlauf der Steuerung (Power On-Reset) zugeordnet ist:
MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN[<Kanal>] = <Kanalnummer>
Der parametrierte Kanal ist Masterkanal der Achse.
- Parametrierung der Reaktion, wenn eine Achse in einem Teileprogramm programmiert wird und die Achse ist dem Kanal aktuell nicht zugeordnet:
 - Einen Alarm anzeigen und die Achse nicht verfahren.
 - Die Achse entsprechend dem Befehl GET (Seite 555) automatisch anfordern
 - Die Achse entsprechend dem Befehl GETD (Seite 555) automatisch anfordernMD30552 \$MA_AUTO_GET_TYPE[<Achse>] = <Reaktion>

Systemvariable

Achsspezifische Systemvariable

- Zustand der Achse bezüglich Achstausch:
\$AA_AXCHANGE_TYP[<Achse>]
- Status der Achse bezüglich Achstausch:
\$AA_AXCHANGE_STAT[<Achse>]

7.3 Programmierung: Achse freigeben (RELEASE)

Funktion

Durch die vordefinierte Prozedur `RELEASE()` wird eine dem aktuellen Kanal zugeordnete Achse für einen Achstausch freigegeben und dazu in den Zustand "Neutrale Achse" versetzt.

Syntax

```
RELEASE (<Achse1>[, Achse2 ... Achse15])
```

Bedeutung

RELEASE:	Achse für Achstausch freigeben	
	Vorlaufstopp:	ja
	Alleine im Satz:	ja
<Achse>:	Achse: Kanalachsname der freigegebenen Achse	
	Spindel: Kanalachsname der freigegebenen Spindel oder Konvertierung der Spindelnummer in den Kanalachsnamen mittels SPI (<Spindelnummer>)	
	Typ:	AXIS

Randbedingungen**Keine Freigabe möglich**

- Die Achse ist an einer Transformation beteiligt.
- Die Achse gehört zu einem Achsverbund.

Führungsschse eines Gantry-Verbunds

Bei Freigabe der Führungsschse eines Gantry-Verbunds, werden auch alle Gleichlaufachsen freigegeben.

7.4 Programmierung: Achse holen (GET, GETD)**Achstausch anfordern (GET)****Funktion**

Durch die vordefinierten Prozedur `GET()` wird eine Achse für den eigenen Kanal angefordert.

Im Kanal, dem die Achse aktuell zugeordnet ist, muss die Achse mit `RELEASE()` in einem Teileprogramm oder Synchronaktion für den Achstausch freigegeben werden.

Nach dem Achstausch ist die Achse im Zustand "Kanal-Achse".

Syntax

```
GET(<Achse1>[, Achse2 ... Achse15])
```

Bedeutung

GET:	Anforderung einer Achse für den aktuellen Kanal	
	Vorlaufstopp:	ja
	Alleine im Satz:	ja
<Achse>:	Achse: Kanalachsname der angeforderten Achse	
	Spindel: Kanalachsname der angeforderten Spindel oder Konvertierung der Spindelnummer in den Kanalachsnamen mittels SPI (<Spindelnummer>)	
	Typ:	AXIS

Randbedingungen

Der Achstausch wird in folgenden Situationen verzögert:

- Die Achse wurde vom Kanal, dem sie aktuell zugeordnet ist, noch nicht mittels RELEASE freigegeben.
- Ein Messsystemwechsel ist noch nicht abgeschlossen
- Änderung des Status der Reglerfreigabe ist noch nicht abgeschlossen (Übergang von Regeln in Nachführen/Halten und umgekehrt).
- Das NC/PLC-Nahtstellensignal "Achs- bzw. Spindelsperre" steht an (DB31, ... DBX1.3 == 1)
- Die aktuelle Verfahrbewegung (Interpolation) der Achse ist noch nicht abgeschlossen.

Achse direkt holen (GETD)

Funktion

Eine Achse, die nicht dem aktuellen Kanal zugeordnet ist, wird für den nachfolgenden Bearbeitungsabschnitt eines Teileprogramms benötigt. Mit der vordefinierten Prozedur GETD () wird die Achse direkt aus dem Kanal dem die Achse zugeordnet ist geholt. Die Achse muss dazu **nicht** mit RELEASE () von diesem Kanal freigegeben werden.

Nach dem Achstausch ist die Achse im Zustand "Kanal-Achse".

Abhängig von Zustand der Achse im abgebenden Kanal wird in diesem ein Vorlaufstopp ausgelöst (STOPRE):

- Zustand "Kanal-Achse" ⇒ Vorlaufstopp
- Zustand "Neutrale Achse" ⇒ **kein** Vorlaufstopp

Um die Übergabe der Achse zwischen den Kanälen mit GETD () zu koordinieren, wird empfohlen, eine Kanalsynchronisation (Seite 531) zwischen anfordernden und abgebenden Kanal zu verwenden.

ACHTUNG
Vorlaufstopp im abgebenden Kanal
Wenn die Achse im abgebenden Kanal im Zustand "Kanal-Achse" ist, wird in diesem Kanal ein Vorlaufstopp ausgelöst (STOPRE):

Syntax

GETD (<Achse>)

Bedeutung

GETD:	Achse direkt in den aktuellen Kanal holen	
	Vorlaufstopp:	ja
	Alleine im Satz:	ja
<Achse>:	Achsen: Kanalachsname der angeforderten Achse Spindeln: Kanalachsname der angeforderten Spindel oder Konvertierung der Spindelnummer in den Kanalachsnamen mittels SPI (<Spindelnummer>)	

Randbedingung

Wenn die Achse im abgebenden Kanal im Zustand "PLC-Achse" ist, muss die Achse vom PLC-Anwenderprogramm aus für den Achstausch freigegeben werden.

Randbedingungen**Kanal-Reset**

- Wird im Kanal, der eine Achse angefordert hat, ein Kanal-Reset ausgelöst, wird der Achstausch abgebrochen.
- Eine getauschte Achse bleibt auch nach einem Kanal-Reset dem Kanal zugeordnet, der die Achse zuletzt angefordert hatte.

7.5 Automatischer Achstausch

Funktion

Ein automatischer Achstausch bzw. das automatische Holen einer Achse in den aktuellen Kanal erfolgt, wenn die Achse im Teileprogramm oder Synchronaktion programmiert wird, sie dem Kanal aktuell aber nicht zugeordnet ist.

Voraussetzung

Parametrierung des automatischen Achstauschs durch `GET ()` oder `GETD ()`:

MD30552 \$MA_AUTO_GET_TYPE (Seite 553)

Randbedingungen

Siehe Beschreibung (Seite 555) zu `GETD ()`.

Beispiele**Beispiel 1**

Programmcode	Kommentar
N1 M3 S1000	; Verfahren der Hauptspindel
N2 RELEASE (SPI(1))	; Freigabe in den neutralen Zustand

Programmcode	Kommentar
N3 S3000	; Programmierung der Hauptspindel => Automatisches Holen ; Verhalten abhängig von MD30552 \$MA_AUTO_GET_TYPE: ; 0 => Alarm "Falscher Achstyp" ; 1 => implizites GET(SPI(1)) ; 2 => implizites GETD(SPI(1))

Beispiel 2

Programmcode	Kommentar
N1 RELEASE (AX1)	; Maschinenachse AX1 ≙ Kanalachse X ; Freigabe in den neutralen Zustand
N2 G04 F2	; Verweilzeit
N3 G0 X100 Y100	; Programmierung der Achse X als Bahnachse ; Verhalten abhängig von MD30552 \$MA_AUTO_GET_TYPE: ; 0 => Alarm "Falscher Achstyp" ; 1 => implizites GET(AX1) ; 2 => implizites GETD(AX1)

Beispiel 3

Programmcode	Kommentar
N1 RELEASE (AX1)	; Maschinenachse AX1 ≙ Kanalachse X ; Freigabe in den neutralen Zustand
N2 G04 F2	; Verweilzeit
N3 POS(X) = 100	; Programmierung der Achse X als Positionierachse ; Verhalten abhängig von MD30552 \$MA_AUTO_GET_TYPE: ; 0 => Alarm "Falscher Achstyp" ; 1 => implizites GET(AX1) *) ; 2 => implizites GETD(AX1) *)

*) Wenn die Achse noch synchronisiert ist, wird für das automatische Holen der Achse mittels (GET () oder GETD ()) kein eigener Satz erzeugt.

7.6 Achstausch durch PLC

Funktion

Vom PLC-Anwenderprogramm aus kann über die NC/PLC-Nahtstelle ein Achstausch angefordert werden:

- Von einem NC-Kanal zur PLC
- Von der PLC zu einem NC-Kanal
- Von einem NC-Kanal zu einem anderen NC-Kanal

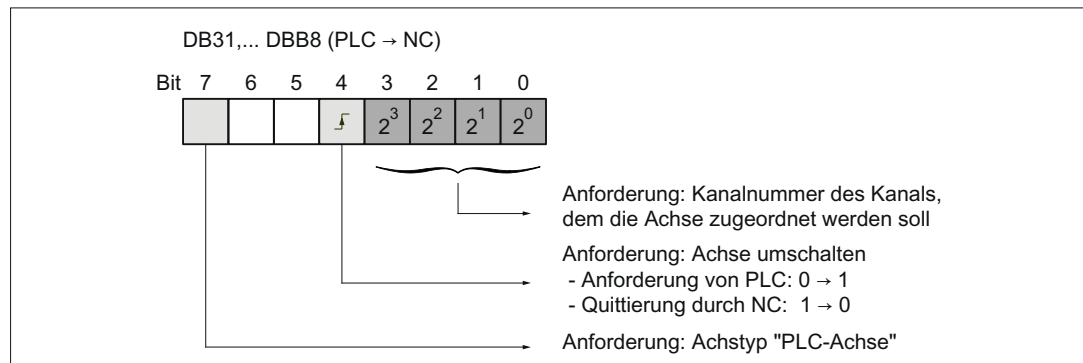


Bild 7-1 Achstausch-Anforderung: DB31, ... DBB8 (PLC → NC)

Achstausch-Status

Der aktuelle Status einer Achse bezüglich des Achstauschs kann vom PLC-Anwenderprogramm aus über die NC/PLC-Nahtstelle gelesen werden.

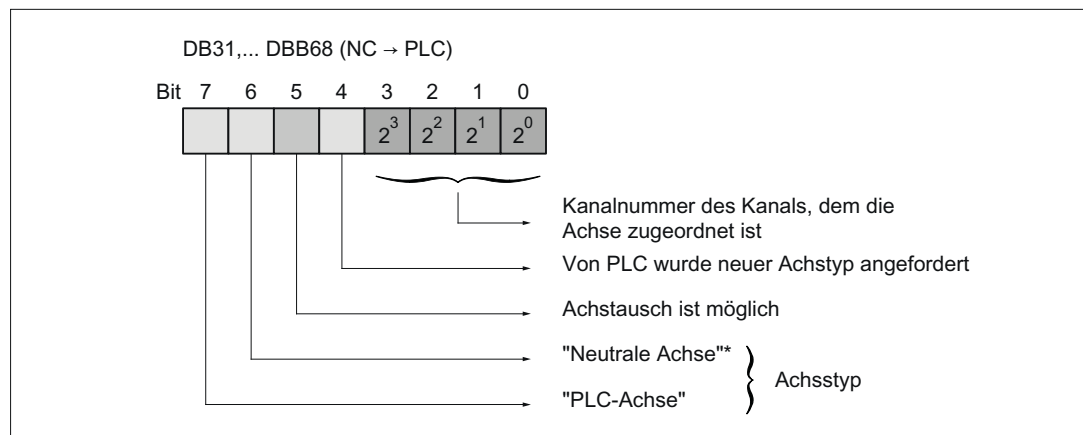



Bild 7-2 Achstausch-Status: DB31, ... DBB68 (NC → PLC)

Beispiele

Beispiel 1


Achstausch einer Achse von Kanal 1 nach Kanal 2 mittels `RELEASE()` und `GET()` in Teileprogrammen, die im jeweiligen Kanal bearbeitet werden:

- Kanal 1: `RELEASE(<Achse>)`
- Kanal 2: `GET(<Achse>)`

	DB31,... DBB68	DB31,... DBB8	
Nach Power On: "Kanal-Achse" in K1	0 0 1 0 0 0 0 1	0 0 0 0 0 0 0 0	
K1: RELEASE(Ax1)	0 1 1 0 0 0 0 1	0 0 0 0 0 0 0 0	
	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	
K2: GET(Ax1)	0 0 1 0 0 0 1 0	0 0 0 0 0 0 0 0	
	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	
	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	

Beispiel 2

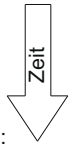
Zustandswechsel einer dem Kanal 1 zugeordneten Achse von "NC-Achse" zu "PLC-Achse" durch das PLC-Anwenderprogramm.

	DB31,... DBB68	DB31,... DBB8			
Nach Power On: "Kanal-Achse" in K1	0 0 1 0 0 0 0 1	0 0 0 0 0 0 0 0			
	0 0 1 0 0 0 0 1	1 0 0 0 0 0 0 0		Anforderung von PLC: Neuer TYP "PLC-Achse"	
Anforderung erkannt Achstausch möglich	0 0 1 1 0 0 0 1	1 1 0 0 0 0 0 0			
	0 0 1 1 0 0 0 1	1 0 0 0 0 0 0 0			Anforderung von NC zurückgesetzt
"PLC-Achse" in K1	1 0 0 0 0 0 0 1	1 0 0 0 0 0 0 0			

Beispiel 3

Zustandswechsel einer dem Kanal 1 zugeordneten Achse von "NC-Achse" über "PLC-Achse" zu "Neutrale Achse" durch das PLC-Anwenderprogramm.

	DB31,... DBB68	DB31,... DBB8	
Nach Power On: "Kanal-Achse" in K1	0 0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	Anforderung von PLC: Neuer TYP "PLC-Achse"
Anforderung erkannt Achstausch möglich	0 0 1 1 0 0 0 0 1	1 0 0 0 0 0 0 0	
"PLC-Achse" in K1	1 0 0 0 0 0 0 0 1	1 0 0 0 0 0 0 0	Anforderung von NC zurückgesetzt
Anforderung erkannt Achstausch möglich	0 0 1 1 0 0 0 0 1	1 0 0 0 0 0 0 0	
"Neutrale Achse" in K1	1 0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 0	Anforderung von PLC: Neuer TYP "Neutrale Achse"
Anforderung erkannt Achstausch möglich	1 0 1 1 0 0 0 0 1	0 1 0 0 0 0 0 0	
"Neutrale Achse" in K1	0 1 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	Anforderung von NC zurückgesetzt
Anforderung erkannt Achstausch möglich	0 1 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	



7.7 Achstausch über Achscontainer-Drehung

Achscontainer-Drehung freigeben

Bei der Freigabe einer Achscontainer-Drehung werden alle einem Kanal zuordenbaren Container-Achsen mittels implizit erzeugtem `GET` oder `GETD` diesem Kanal zugeordnet. Eine Abgabe der Achsen, z.B. an einen anderen Kanal, ist erst wieder nach der Containerdrehung möglich.

Hinweis

Die implizite Zuordnung einer Achse zum einem Kanal kann **nicht** erfolgen, wenn die Achse im Zustand "Hauptlauf-Achse" (z. B. PLC-Achse) ist. Um an der Achscontainer-Drehung teilnehmen zu können, müsste die Achse den Zustand erst verlassen.

Weitere Erläuterungen zum Achstausch von Container-Achsen (siehe Kapitel "B3: Dezentrale Systeme").

Beispiel: Achscontainer-Drehung mit implizitem `GET` oder `GETD`

Aktion Kanal 1

AXCTSWE (CT 1)

Aktion Kanal 2

SPOS = 180 positioniert

; holt Spindel in Kanal 1

; und erlaubt Achs-Container-Drehung

Annahme:

Die Spindel wird in beiden Kanälen verwendet und ist auch Achse im Achscontainer CT 1.

Aktivierung

Der Achstausch mittels Achscontainer-Drehung und implizitem GET/GETD wird aktiviert mit Maschinendatum MD10722 \$MN_AXCHANGE_MASK, Bit 1=1.

7.8 Achstausch mit und ohne Vorlaufstopp

Achstauscherweiterung ohne Vorlaufstopp

Statt eines GET-Satzes mit Vorlaufstopp wird nur ein Zwischensatz mit dieser GET-Anforderung erzeugt. Im Hauptlauf wird bei Abarbeitung dieses Satzes überprüft, ob die Zustände der Achse im Satz mit den aktuellen Achszuständen übereinstimmen. Bei nicht Übereinstimmung kann ein Zwangsreorganisieren ausgelöst werden.

Folgende Zustände einer **Achse bzw. positionierende Spindel** werden überprüft auf:

- Den Mode, ob Achse oder positionierende Spindel
- Die Sollposition

Folgende Zustände einer **Spindel im Drehzahlmode** werden überprüft auf:

- Spindelbetriebsart: Drehzahlmode
- Spindeldrehzahl S
- Drehrichtung M3, M4
- Getriebestufe M40, M41, M42, M43, M44, M45
- Masterspindel auf konstante Schnittgeschwindigkeit

Gegebenenfalls ist ein Zwangsreorganisieren möglich. Folgeachsen werden in jedem Fall zwangsreorganisiert.

Aktivierung

Der Achstausch ohne Vorlauf und Überprüfung der aktuellen Zustände wird aktiviert mit Maschinendatum MD10722 \$MN_AXCHANGE_MASK, Bit 2=1.

Beispiel

Aktivierung eines Achstauschs ohne Vorlaufstopp

Tabelle 7-1

```
N010 M4 S1000
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
N041 G4 F2
N050 M5
N099 M30

```

Wird die Spindel (Achse B) unmittelbar nach dem Satz N023 als PLC-Achse z. B. auf 180° und zurück auf 1° verfahren und wieder zur neutralen Achse, so löst der Satz N040 keinen Vorlaufstopp und kein Reorganisieren aus.

Sonderfall: Achstausch mit Vorlaufstopp

Ohne dass eine GET oder GETD Anweisung vorher im Hauptlauf angekommen ist, kann die Spindel oder Achse z. B. mit RELEASE(Achse) oder WAITP(Achse) wieder freigegeben werden. Ein nachfolgender GET führt zu einem GET **mit** Vorlaufstopp.

7.9 Achse ausschließlich PLC-kontrolliert

Funktion

Nach dem Hochlauf der Steuerung befindet sich die Achse im Zustand "Neutrale Achse". Die Kontrolle erfolgt durch die PLC. Zum Verfahren der Achse als konkurrierende Positionierachse (von PLC aus über Funktionsbaustein FC18), muss die Achse zuerst explizit von der PLC angefordert werden.

Hinweis

Der Achstausch zur PLC kann per Maschinendatum auf ausschließlich von PLC kontrollierte Achsen eingeschränkt werden: MD10722 \$MN_AXCHANGE_MASK, Bit 3 = 1

Die Achse kann **nicht** von einem NC-Teileprogramm aus verfahren werden.

Parametrierung

Die Parametrierung einer Achse als ausschließlich von PLC kontrollierte Achse erfolgt über das achsspezifische Maschinendatum:

MD30460 \$MA_BASE_FUNCTION_MASK, Bit 4 = 1

Kontrolle durch PLC

Eine ausschließlich von PLC kontrollierte Achse wird in ihrem Verfahrverhalten nur durch die axialen NC/PLC-Nahtstellensignale beeinflusst:

- DB31, ... DBX28.1 (Reset)
- DB31, ... DBX28.2 (Fortsetzen)
- DB31, ... DBX28.6 (Halt mit Bremsrampe)

Mögliche Verfahrfunktionen

Folgende Verfahrfunktionen sind bei einer ausschließlich PLC-kontrollierten Achse möglich:

1. Verfahren in der Betriebsart JOG über Verfahrtasten und Handrad
2. Referenzieren der Achse
3. Verfahren als Kommandoachse über statische Synchronaktionen
4. Verfahren als asynchrone Pendelachse
5. Verfahren als konkurrierende Positionierachse von der PLC aus über FC18

Nach Abschluss der Verfahrfunktionen 1. bis 4. befindet sich die Achse automatisch wieder im Zustand "Neutrale Achse". Nach Abschluss der Verfahrfunktion 5. von PLC aus bleibt die Achse im Zustand "PLC-Achse". In den Zustand "Neutrale Achse" wechselt die Achse erst nach expliziter Freigabe durch die PLC.

7.10 Achse fest der PLC zugeordnet

Funktion

Nach dem Hochlauf der Steuerung befindet sich die Achse im Zustand "Neutrale Achse" und wird vom NC-Kanal kontrolliert. Zum Verfahren der Achse als konkurrierende Positionierachse (von PLC aus über Funktionsbaustein FC18), muss die Achse **nicht** zuerst explizit von der PLC angefordert werden. Der Achstausch zur PLC erfolgt automatisch mit der Verfahranforderung über FC18. Nach Abschluss der über FC18 angeforderten Verfahrbewegung wechselt die Achse wieder automatisch in den Zustand "Neutrale Achse".

Nach Anforderung durch die PLC kann nach erfolgtem Achstausch die Kontrolle der Achse auch von der PLC aus erfolgen: Zustand "PLC-Achse".

Hinweis

Der Achstausch zur PLC kann per Maschinendatum auf ausschließlich fest der PLC zugeordnete Achsen eingeschränkt werden: MD10722 \$MN_AXCHANGE_MASK, Bit 3 = 1

Parametrierung

Die Parametrierung einer Achse als fest der PLC zugeordnete Achse erfolgt über das achsspezifische Maschinendatum:

MD30460 \$MA_BASE_FUNCTION_MASK, Bit 5 = 1

Kontrolle durch PLC oder NC-Kanal

Eine fest der PLC zugeordnete Achse kann in ihrem Verfahrverhalten entweder durch den NC-Kanal oder durch die PLC beeinflusst werden:

NC-Kanal: kanalspezifische NC/PLC-Nahtstellensignale (Auswahl)

- DB21, ... DBXDBX7.1 (NC-Start)
- DB21, ... DBXDBX7.3 (NC-Stop)
- DB21, ... DBXDBX7.7 (Reset)

PLC: axiale NC/PLC-Nahtstellensignale

- DB31, ... DBX28.1 (Reset)
- DB31, ... DBX28.2 (Fortsetzen)
- DB31, ... DBX28.6 (Halt mit Bremsrampe)

Mögliche Verfahrfunktionen

Folgende Verfahrfunktionen sind bei einer fest der PLC zugeordneten Achse möglich:

1. Verfahren in der Betriebsart JOG über Verfahrtasten und Handrad
2. Referenzieren der Achse
3. Verfahren als konkurrierende Positionierachse von der PLC aus über FC18

Nach Abschluss der Verfahrfunktionen 1. bis 3. befindet sich die Achse automatisch wieder im Zustand "Neutrale Achse".

7.11 Geometrieachse im gedrehten WKS und Achstausch

Achstauscherweiterung über Frame mit Rotation

In der Betriebsart JOG kann eine Geometrieachse im gedrehten WKS als PLC-Achse oder Kommandoachse über statische Synchronaktionen verfahren werden. Dazu muss folgendes Maschinendatum gesetzt sein:

MD32074 \$MA_FRAME_OR_CORRPOS_NOTALLOWED, Bit 10 == 1

Hinweis

Betriebsartenwechsel in der Betriebsart JOG

Vor einem Wechsel aus Betriebsart JOG in eine andere Betriebsart, müssen Verfahrbewegungen von **allen PLC- und Kommandoachsen** abgeschlossen sein, die als Geometrieachsen im gedrehten WKS verknüpft sind. Diese Achsen müssen zumindest wieder neutrale Achse geworden sein, anderenfalls wird beim Betriebsartenwechsel der Alarm 16908 angezeigt. Der Alarm wird auch dann angezeigt, wenn nur eine einzige Geometrieachse im gedrehten Koordinatensystem als PLC- bzw. Kommandoachse verfahren wird.

Eine solche Achse kann nur innerhalb des Kanals zur PLC- oder Kommandoachse werden. Ein Achstausch in einen anderen Kanal ist nicht möglich.

Voraussetzung beim Wechsel von JOG nach AUTOMATIK

Bei einem Betriebsartenwechsel von JOG nach AUTOMATIK, wird im Zustand "Programm unterbrochen" der Endpunkt der Geometrieachsbewegungen nur übernommen, wenn Maschinendatum gesetzt ist:

MD 32074: FRAME_OR_CORRPOS_NOTALLOWED, Bit 11 == 1

Damit werden PLC- bzw. Kommandoachsen entsprechend der Drehungen im WKS positioniert.

Es werden **alle** im gedrehten WKS beeinflussten Achsen als Geometrieachs-Verbund betrachtet und gemeinsam behandelt. Damit werden alle Achsen des Verbundes

- dem NC-Programm zugeordnet oder
- alle Achsen sind neutral oder
- alle Achsen sind als Hauptlauf-Achsen (PLC-, Kommando-, oder Pendel-Achse) aktiv

Zum Beispiel wird bei einer mit einem WAITP programmierten Achse, auf alle weiteren Achsen dieses Geometrieachs-Verbundes gewartet, um diese Achsen gemeinsam zu neutralen Achsen werden zu lassen. Wird eine dieser Achsen im Hauptlauf zur PLC-Achse, werden alle anderen Achsen dieses Verbundes zu neutralen Achsen.

Randbedingungen

Ist MD32074 \$MA_FRAME_OR_CORRPOS_NOTALLOWED, Bit 10 == 0 und im NC-Programm `ROT Z45` programmiert, dann ist für die X- und Y-Achse **kein Achstausch** möglich. Dies gilt analog auch für die Z-Achse bei z.B. `ROT X45` oder `ROT Y45` und auch in der Betriebsart JOG, wenn ein Satz mit solch einer Programmierung unterbrochen wurde. In diesem Fall sind zwar für die X- und Y-Achse die NC/PLC-Nahtstellesignale "Achstausch möglich" (DB31, ...DBX68.5) gesetzt, diese werden aber zurückgesetzt.

Nur wenn MD32074 \$MA_FRAME_OR_CORRPOS_NOTALLOWED, Bit 10 == 1 und kein Satz mit dieser Programmierung aktuell verfahren wird, ist in der Betriebsart JOG ein Achstausch für solche Achsen möglich.

7.12 Achstausch aus Synchronaktionen

Funktion

Als Aktion einer Synchronaktion kann eine Achse für den Kanal angefordert (`GET (<Achse>)`) oder für den Achstausch freigegeben (`RELEASE (<Achse>)`) werden.

Hinweis

Die Achse muss dem Kanal über Maschinendaten als Kanalachse zugeordnet sein.

Mit dem Befehl `AXTOCHAN` kann über Synchronaktionen oder im Teileprogramm eine Achse zwischen den Kanälen direkt an einen bestimmten Kanal übergeben werden. Dies muss nicht der eigene Kanal sein, und es ist auch nicht erforderlich, dass dieser Kanal das aktuelle Interpolationsrecht für die Achse besitzt.

Aktueller Zustand und Interpolationsrecht der Achse

Mit welchem Achstyp und Interpolationsrecht ein möglicher Achstausch erfolgen soll, kann über folgende Systemvariable gelesen werden:

`<Wert> = $AA_AXCHANGE_TYP[<Achse>]`

<Wert>	Bedeutung
0	Die Achse ist dem NC-Programm zugeordnet.
1	Achse der PLC zugeordnet oder als Kommandoachse/Pendelachse aktiv.
2	Ein anderer Kanal hat das Interpolationsrecht.
3	Achse ist neutrale Achse.
4	Neutrale Achse wird von der PLC kontrolliert.
5	Ein anderer Kanal hat das Interpolationsrecht, die Achse ist für das NC-Programm angefordert.
6	Ein anderer Kanal hat das Interpolationsrecht, die Achse ist als neutrale Achse angefordert.
7	Achse ist PLC-Achse oder als Kommandoachse/Pendelachse aktiv, die Achse ist für das NC-Programm angefordert.
8	Achse ist PLC-Achse oder als Kommandoachse/Pendelachse aktiv, die Achse ist als neutrale Achse angefordert.
9	Fest zugeordnete PLC-Achse, im Zustand neutrale Achse.
10	Fest zugeordnete PLC-Achse von der PLC kontrolliert, im Zustand neutrale Achse.

Eine "Fest zugeordnete PLC-Achse im Zustand neutrale Achse" (9) und eine "Fest zugeordnete PLC-Achse von der PLC kontrolliert im Zustand neutrale Achse" (10) wird **unabhängig von GET und RELEASE** fest der PLC zugeordnet. Ob die Achse getauscht werden kann, wird über die Systemvariable `$AA_AXCHANGE_STAT[Achse]` angezeigt.

Zustandsübergänge GET, RELEASE aus Synchronaktionen und wenn GET erfüllt ist

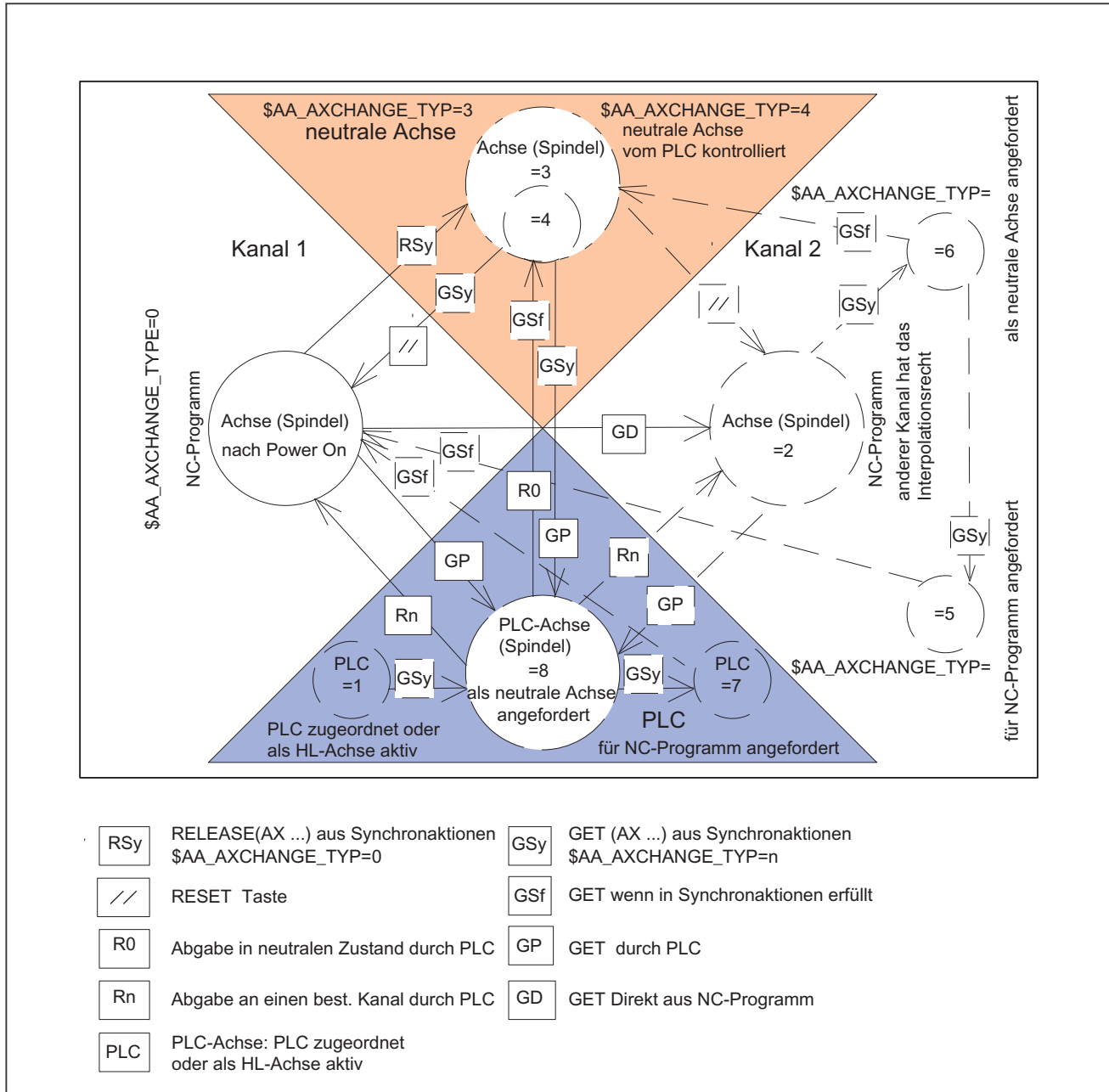


Bild 7-3 Übergänge aus Synchronaktionen

Weitere Informationen

Funktionshandbuch Synchronaktionen; Aktionen in Synchronaktionen

7.13 Achstausch bei Führungsachsen (Gantry)

Funktion

Ein geschlossener Gantry-Verbund wird bei einem Achstausch bezüglich seiner Achsen immer als Einheit behandelt. Daher erfolgt bei einem Achstausch der Führungsachse gleichzeitig auch ein Achstausch für alle Gleichlaufachsen des Gantry-Verbundes. Dazu müssen neben den in den vorausgehenden Kapiteln beschriebenen Voraussetzungen für die Führungsachse auch die entsprechenden Voraussetzungen für alle Gleichlaufachsen des Gantry-Verbundes erfüllt sein.

Axiale Maschinendaten

Folgende axiale Maschinendaten müssen bei einem Achstausch für alle Achsen eines geschlossenen Gantry-Verbundes gleich eingestellt sein:

- MD30460 \$MA_BASE_FUNCTION_MASK, Bit 4 (Kontrolle ausübende Komponente)
- MD30460 \$MA_BASE_FUNCTION_MASK, Bit 5 (Zuordnung zu Komponente)

Axiale NC/PLC-Nahtstellensignale

Folgende axialen NC/PLC-Nahtstellensignale zeigen im Rahmen der Funktion Achstausch für alle Achsen eines geschlossenen Gantry-Verbundes immer die gleichen Werte:

- DB31, ... DBX63.0 (Reset ausgeführt)
- DB31, ... DBX63.1 (PLC kontrolliert Achse)
- DB31, ... DBX63.2 (Achsstopp aktiv)

Axiale Systemvariable

Folgende axialen Systemvariablen zeigen im Rahmen der Funktion Achstausch für alle Achsen eines geschlossenen Gantry-Verbundes immer die gleichen Werte:

- \$AA_AXCHANGE_TYP (Achstyp bezüglich Achstausch)
- \$AA_AXCHANGE_STAT (Achstatus bezüglich Achstausch)
- \$AA_SNLAX_STAT (Achstyp der Einzelachse)

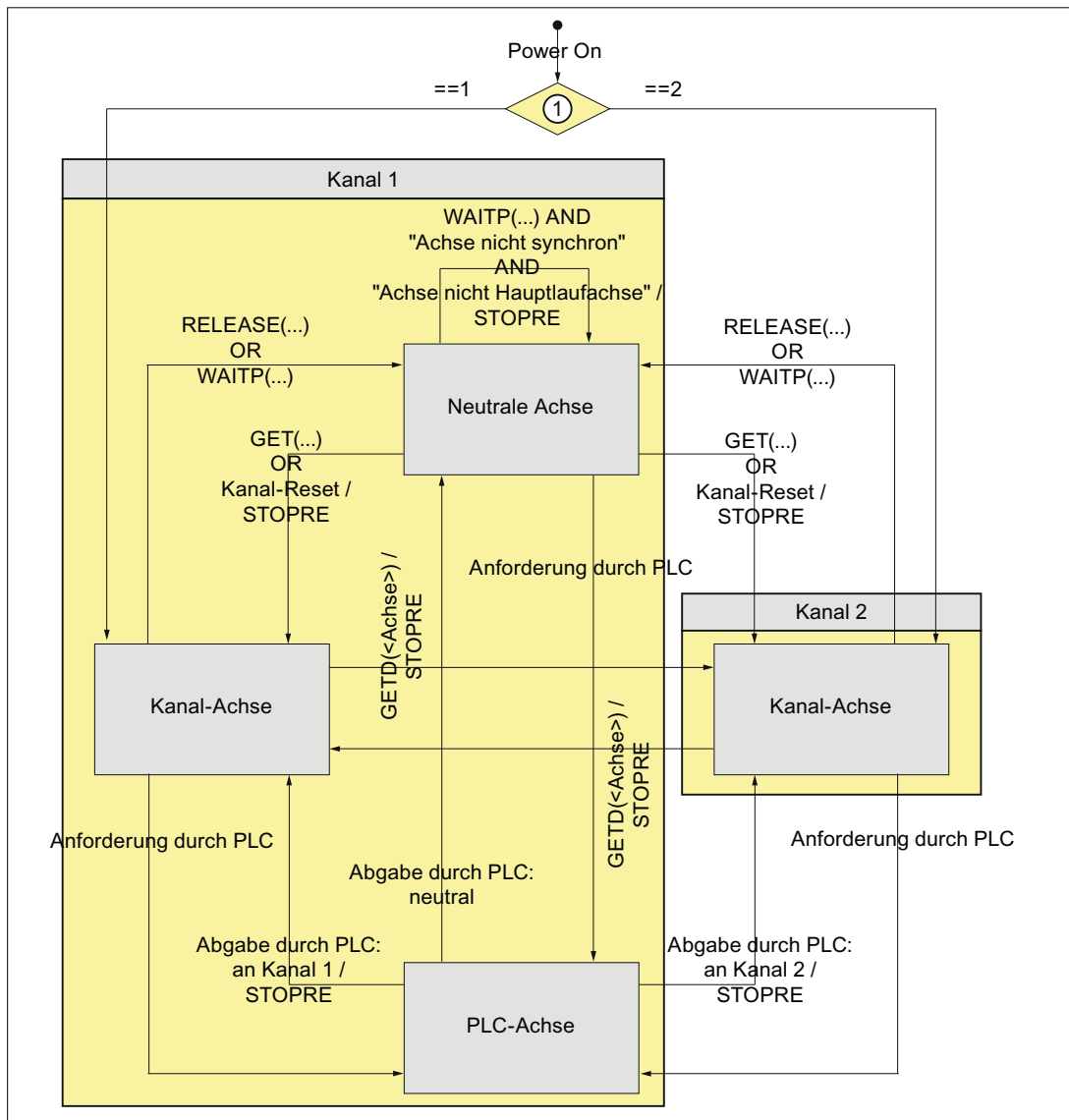
7.14 Zustandsdiagramm

Das folgende Bild zeigt für Kanal 1 die Zustände, Ereignisse, Aktionen und die Zustandsübergänge für eine Achse bezogen auf die Funktion "Achstausch".

Für Kanal 2 sind der Übersicht halber die Sub-Zustände "Neutrale Achse" und "PLC-Achse" nicht dargestellt.

Annahme: Die Achse ist defaultmäßig dem Kanal 1 zugeordnet:

MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN[<Achse>] = 1



① MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN[<Achse>]

Bild 7-4 Zustandsdiagramm: Achstausch

7.15 Randbedingungen

Beim Wechsel einer Achse vom Zustand "PLC-Achse", "Neutrale Achse" oder "Achse im anderen Kanal" in den Zustand "Kanal-Achse" erfolgt eine Synchronisation mit Vorlaufstopp und Synchronisation im holenden Kanal. Dabei erfolgt:

- Achse: Übernahme der aktuellen Achsposition
- Spindel: Übernahme der aktuellen Drehzahl und der aktuellen Getriebestufe

7.15.1 Achstausch von PLC

Befindet sich das Teileprogramm des Kanals zum Zeitpunkt zu dem der Achstausch (PLC → Kanal oder Kanal → PLC) von der PLC aus angefordert wird in einem der folgenden Bearbeitungsabschnitte, wird der Achstausch erst nach dem Verlassen des Bearbeitungsabschnitts ausgeführt:

- Bahnsteuerbetrieb (G64/G640)
- Gewindeschneiden/-bohren (G33/G331/G332)

7.15.2 Satzsuchlauf mit Berechnung

Bei Satzsuchlauf mit Berechnung werden nur die sich nicht gegenseitig aufhebenden Befehle GET, GETD und RELEASE im Aktionssatz ausgegeben.

Beispiel

Satzvorlauf mit Berechnung auf Zielsatz N700:

Programmcode	Kommentar
...	
N100 RELEASE (AX1)	; RELEASE(AX1) wird aufgesammelt
N110 GET (AX2)	; GET(AX2) wird aufgesammelt
...	
N400 GET (AX1)	; GET(AX1) hebt RELEASE(AX1) auf => ; RELEASE(AX1) und GET(AX1) werden nicht gespeichert
...	
N700 ...	; Zielsatz ⇒ Ausgabe: GET(AX2)
N710 RELEASE (AX2)	
...	

Weitere Informationen

Weitere Informationen finden Sie unter Satzsuchlauf Typ 1, 2 und 4 (Seite 56).

7.16 Beispiel

Annahmen

- Kanal 1: Kanalachsen sind folgende Achsen: 1, 2, 3, 4
- Kanal 2: Kanalachsen sind folgende Achsen: 4, 5, 6
- Defaultzuordnung: Achse 4 (AX4) wird defaultmäßig Kanal 1 zugeordnet

Parametrierung

Kanal 1

Achsnamen im Kanal: MD20080

- \$MC_AXCONF_CHANAX_NAME_TAB[<Kanal 1>][0] = "X" ; 1. Kanalachse
- \$MC_AXCONF_CHANAX_NAME_TAB[<Kanal 1>][1] = "Y" ; 1. Kanalachse
- \$MC_AXCONF_CHANAX_NAME_TAB[<Kanal 1>][2] = "Z" ; 1. Kanalachse
- \$MC_AXCONF_CHANAX_NAME_TAB[<Kanal 1>][3] = "U" ; 1. Kanalachse

Verwendete Maschinenachsen: MD20070

- \$MC_AXCONF_MACHAX_USED[<Kanal 1>][0] = 1 ; 1. Kanalachse → Achse 1
- \$MC_AXCONF_MACHAX_USED[<Kanal 1>][1] = 2 ; 2. Kanalachse → Achse 2
- \$MC_AXCONF_MACHAX_USED[<Kanal 1>][2] = 3 ; 3. Kanalachse → Achse 3
- \$MC_AXCONF_MACHAX_USED[<Kanal 1>][3] = 4 ; 4. Kanalachse → Achse 4

Kanal 2

Achsnamen im Kanal: MD20080

- \$MC_AXCONF_CHANAX_NAME_TAB[<Kanal 2>][0] = "X" ; 1. Kanalachse
- \$MC_AXCONF_CHANAX_NAME_TAB[<Kanal 2>][1] = "Y" ; 1. Kanalachse
- \$MC_AXCONF_CHANAX_NAME_TAB[<Kanal 2>][2] = "U" ; 1. Kanalachse

Verwendete Maschinenachsen: MD20070

- \$MC_AXCONF_MACHAX_USED[<Kanal 2>][0] = 5 ; 1. Kanalachse → Achse 5
- \$MC_AXCONF_MACHAX_USED[<Kanal 2>][1] = 6 ; 2. Kanalachse → Achse 6
- \$MC_AXCONF_MACHAX_USED[<Kanal 2>][2] = 4 ; 3. Kanalachse → Achse 4

Defaultzuordnung

Masterkanal von Achse 4 (AX4) → Kanal 1

AX4 ist der Standardname der 4. Maschinenachse entsprechend MD10000

\$MN_AXCONF_MACHAX_NAME_TAB[3]

- MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN[AX4] = 1

Programmbeispiel

Programm im Kanal 1	Programm im Kanal 2
...	...
; Verfahren der Achse 4 (AX4)	; Synchronisationspunkt mit Kanal 1
G01 F1000 U100	; WAITM(1,1,2)
; Freigabe von AX4	; Anfordern von AX4
RELEASE(AX4)	GET(AX4)
; Anwahl Programm TAUSH2 in Kanal 2	; Verfahren von Achse 4 (AX4)
INIT (2, "_N_MPF_DIR_N_TAUSH2_MPF", "S")	G0 U0

Programm im Kanal 1	Programm im Kanal 2
; Start Programm TAUSH2 in Kanal 2	...
START(2)	
; Synchronisationspunkt mit Kanal 2	; Freigabe von AX4
WAITM(1,1,2)	RELEASE(AX4)
...	...
M30	M30

7.17 Datenlisten

7.17.1 Maschinendaten

7.17.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10010	ASSIGN_CHAN_TO_MODE_GROUP[n]	Kanal gültig in BAG [Kanalnr.]: 0, 1
10722	AXCHANGE_MASK	Parametrierung des Achstausch-Verhaltens

7.17.1.2 Kanal-spezifische Maschinendaten

Grundmaschinendaten des Kanals

Nummer	Bezeichner: \$MC_	Beschreibung
20000	CHAN_NAME	Kanalname
20050	AXCONF_GEOAX_ASSIGN_TAB[n]	Zuordnung Geometrieachse zu Kanalachse [GEOAchsnr.]: 0...2
20060	AXCONF_GEOAX_NAME_TAB[n]	Geometrieachsname im Kanal [GEOAchsnr.]: 0...2
20070	AXCONF_MACHAX_USED[n]	Maschinenachsnummer gültig im Kanal [Kanalachsnr.]: 0...7
20080	AXCONF_CHANAX_NAME_TAB[n]	Kanalachsname im Kanal [Kanalachsnr.]: 0...7
20090	SPIND_DEF_MASTER_SPIND	Löschstellung der Masterspindel im Kanal
20100	DIAMETER_AX_DEF	Geometrieachse mit Planachsfunktion
20110	RESET_MODE_MASK	Festlegung der Steuerungs-Grundstellung nach Reset/TP-Ende
20112	START_MODE_MASK	Festlegung der Steuerungs-Grundstellung bei NC-START

Nummer	Bezeichner: \$MC_	Beschreibung
20150	GCODE_RESET_VALUES[n]	Löschstellung der G-Gruppen [G-Gruppennr.]: 0...59
20160	CUBIC_SPLINE_BLOCKS	Anzahl der Sätze beim C-Spline
20170	COMPRESS_BLOCK_PATH_LIMIT	Maximale Verfahrlänge eines NC-Satzes bei Kompression
20200	CHFRND_MAXNUM_DUMMY_BLOCKS	Leersätze bei Phase/Radien
20210	CUTCOM_CORNER_LIMIT	Maximalwinkel für Ausgleichssätze bei WRK
20220	CUTCOM_MAX_DISC	Maximaler Wert für DISC
20230	CUTCOM_CURVE_INSERT_LIMIT	Maximalwinkel für Schnittpunktberechnung bei WRK
20240	CUTCOM_MAXNUM_CHECK_BLOCKS	Sätze für vorausschauende Konturberechnung bei WRK
20250	CUTCOM_MAXNUM_DUMMY_BLOCKS	Satzanzahl ohne Verfahrbewegung bei WRK
20270	CUTTING_EDGE_DEFAULT	Grundstellung der Werkzeugschneide ohne Programmierung
20400	LOOKAH_USE_VELO_NEXT_BLOCK	Lookahead auf programmierte Folgesatzgeschwindigkeit
20430	LOOKAH_NUM_OVR_POINTS	Anzahl der Override-Schalter-Eckwerte bei Lookahead
20440	LOOKAH_OVR_POINTS[n]	Override-Schalter-Eckwerte bei Lookahead [Eckwertnummer]: 0...1
20500	CONST_VELO_MIN_TIME	Minimale Zeit mit konstanter Geschwindigkeit
20600	MAX_PATH_JERK	Bahnbezogener Maximalruck
20610	ADD_MOVE_ACCEL_RESERVE	Beschleunigungsreserve für überlagerte Bewegungen
20650	THREAD_START_IS_HARD	Beschleunigungsverhalten der Achse beim Gewindeschneiden
20700	REFP_NC_START_LOCK	NC-Startsperre ohne Referenzpunkt
20750	ALLOW_GO_IN_G96	G0-Logik bei G96
20800	SPF_END_TO_VDI	Unterprogrammende an PLC
21000	CIRCLE_ERROR_CONST	Kreisendpunktüberwachung Konstante
21010	CIRCLE_ERROR_FACTOR	Kreisendpunktüberwachung Faktor
21100	ORIENTATION_IS_EULER	Winkeldefinition bei Orientierungsprogrammierung
21110	X_AXIS_IN_OLD_X_Z_PLANE	Koordinatensystem bei automatischer Framedefinition
21200	LIFTFAST_DIST	Verfahrstrecke bei Schnellabheben von der Kontur
21250	START_INDEX_R_PARAM	Nummer des ersten kanalspezifischen R-Parameters

Hilfsfunktionseinstellungen des Kanals

Nummer	Bezeichner: \$MC_	Beschreibung
22000	AUXFU_ASSIGN_GROUP[n]	Hilfsfunktionsgruppe [HiFunr. im Kanal]: 0...49
22010	AUXFU_ASSIGN_TYPE[n]	Hilfsfunktionsart [HiFunr. im Kanal]: 0...49
22020	AUXFU_ASSIGN_EXTENSION[n]	Hilfsfunktionserweiterung [HiFunr. im Kanal]: 0...49

Nummer	Bezeichner: \$MC_	Beschreibung
22030	AUXFU_ASSIGN_VALUE[n]	Hilfsfunktionswert [HiFunr. im Kanal]: 0...49
22200	AUXFU_M_SYNC_TYPE	Ausgabezeitpunkt der M-Funktionen
22210	AUXFU_S_SYNC_TYPE	Ausgabezeitpunkt der S-Funktionen
22220	AUXFU_T_SYNC_TYPE	Ausgabezeitpunkt der T-Funktionen
22230	AUXFU_H_SYNC_TYPE	Ausgabezeitpunkt der H-Funktionen
22240	AUXFU_F_SYNC_TYPE	Ausgabezeitpunkt der F-Funktionen
22250	AUXFU_D_SYNC_TYPE	Ausgabezeitpunkt der D-Funktionen
22260	AUXFU_E_SYNC_TYPE (in Vorbereitung)	Ausgabezeitpunkt der E-Funktionen
22400	S_VALUES_ACTIVE_AFTER_RESET	S-Funktion über RESET wirksam
22410	F_VALUES_ACTIVE_AFTER_RESET	F-Funktion über RESET wirksam
22500	GCODE_OUTPUT_TO_PLC	G-Befehle an PLC
22550	TOOL_CHANGE_MODE	Neue Werkzeugkorrektur bei M-Funktion
22560	TOOL_CHANGE_M_CODE	M-Funktion für Werkzeugwechsel

Kanal-spezifische Speichereinstellungen

Nummer	Bezeichner: \$MC_	Beschreibung
25000	REORG_LOG_LIMIT	Prozentsatz des IPO-Puffers für Freigabe des Logfiles
28000	MM_REORG_LOG_FILE_MEM	Speichergöße für REORG (DRAM)
28010	MM_NUM_REORG_LUD_MODULES	Anzahl der Bausteine für lokale Anwendervariablen bei REORG (DRAM)
28020	MM_NUM_LUD_NAMES_TOTAL	Anzahl der lokalen Anwendervariablen (DRAM)
28030	MM_NUM_LUD_NAMES_PER_PROG	Anzahl der lokalen Anwendervariablen pro Programm (DRAM)
28040	MM_LUD_VALUES_MEM	Speichergöße für lokale Anwendervariablen (DRAM)
28050	MM_NUM_R_PARAM	Anzahl der kanalspezifischen R-Parameter (SRAM)
28060	MM_IPO_BUFFER_SIZE	Anzahl der NC-Sätze im IPO-Puffer (DRAM)
28070	MM_NUM_BLOCKS_IN_PREP	Anzahl der Sätze für die Satzaufbereitung. (DRAM)
28080	MM_NUM_USER_FRAMES	Anzahl der einstellbaren Frames (SRAM)
28090	MM_NUM_CC_BLOCK_ELEMENTS	Anzahl Satzelemente für Compile-Zyklen (DRAM)
28100	MM_NUM_CC_BLOCK_USER_MEM	Größe des Satzspeichers für Compile-Zyklen (DRAM)
28500	MM_PREP_TASK_STACK_SIZE	Stackgröße der Präparationstask (DRAM)
28510	MM_IPO_TASK_STACK_SIZE	Stackgröße der Ipo-Task (DRAM)

7.17.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30460	BASE_FUNCTION_MASK	Achsfunktionen
30550	AXCONF_ASSIGN_MASTER_CHAN	Löschstellung des Kanals für Achswechsel
30552	AUTO_GET_TYPE	Festlegung für automatische GET
30600	FIX_POINT_POS	Festwertpositionen der Achsen bei G75

Nummer	Bezeichner: \$MA_	Beschreibung
32074	FRAME_OR_CORRPOS_NOTALLOWED	Frame oder HL-Korrektur sind unzulässig
33100	COMPRESS_POS_TOL	Maximale Abweichung bei Kompression

7.17.2 Settingdaten

7.17.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42000	THREAD_START_ANGLE	Startwinkel bei Gewinde
42100	DRY_RUN_FEED	Probelaufvorschub

V2: Vorverarbeitung

8.1 Kurzbeschreibung

Vorverarbeitung

Die in den Verzeichnissen für Standard- und Anwenderzyklen befindlichen Programme können zur schnellen Abarbeitung vorverarbeitet werden.

Die Vorverarbeitung wird über Maschinendatum aktiviert.

Die Standard- und Anwenderzyklen werden bei Power On vorverarbeitet, d. h. das Teileprogramm wird in einen bearbeitungsoptimalen binären Zwischencode steuerungsintern übersetzt (kompiliert).

Alle Programmfehler, die mit Korrektursatz korrigiert werden können, werden bereits zum Zeitpunkt der Vorverarbeitung erkannt. Zusätzlich wird bei Verwendung von Sprüngen und Kontrollstrukturen überprüft, ob die Sprungziele vorhanden sind und ob die Schachtelung von Kontrollstrukturen korrekt ist.

Es steht die volle Steuerungsfunktionalität zur Verfügung:

- Override-Beeinflussung
- Reaktionen auf Daten und Signale, die von der PLC oder Bedienung vorgegeben werden
- Aktuelle Satzanzeige
- Die Programme können im Einzelsatz (SBL1 und SBL2) bearbeitet werden. Satzsuchlauf ist möglich. Das Kompilat kann nicht archiviert werden, es ist dem Anwender verborgen und wird bei Power On jeweils neu erzeugt.

Die Vorverarbeitung ist geeignet für:

- Laufzeitoptimierung von Teileprogrammen mit Hochsprachenanteilen (Sprüngen, Kontrollstrukturen, Bewegungssynchronaktionen)
- Rechenintensive Teileprogramme (z. B. Abspannzyklen)
- Zeitkritische Stellen (z. B. die Programmfortsetzung nach Vorlaufstopp bei schnellem Restweglöschen oder Rückhub oder im Werkzeugwechselzyklus) werden schneller bearbeitet.

Allgemeines

Die Vorverarbeitung von Standard- und Anwenderzyklen ist möglich. Damit kann die Bearbeitungszeit von Teileprogrammen ohne Einschränkung der Steuerungsfunktionalität reduziert werden.

Die Standard- und Anwenderzyklen werden vorverarbeitet, bei entsprechender Setzung des Maschinendatums:

MD10700 \$MN_PREPROCESSING_LEVEL (Programmvorverarbeitungsstufe)

Die Vorverarbeitung erfolgt programmspezifisch. Die Mischung von vorverarbeiteten und im ASCII-Format interpretierten Teileprogrammen ist möglich. Die Vorverarbeitung dient zur Verkürzung von Nebenzeiten.

Für die Vorverarbeitung von Zyklen wird Speicherplatz benötigt. Zur besseren Speicherausnutzung haben Sie zwei Möglichkeiten:

- Mit dem Befehl `DISPLOF` (Anzeige aus) wird das ausführbare Programm verkürzt.
- Das MD10700 `$MN_PREPROCESSING_LEVEL` wurde um Bit 2 und 3 erweitert. Dadurch kann die Zyklen-Vorverarbeitung gezielt für einzelne Verzeichnisse angewählt werden (z. B. Anwenderzyklen).

Das MD10700 `$MN_PREPROCESSING_LEVEL` wurde erweitert um Bit 4. Damit kann die Vorverarbeitung der Anwenderzyklen aus dem Directory `_N_CMA_DIR` angewählt werden.

Das MD10700 `$MN_PREPROCESSING_LEVEL` wurde erweitert um Bit 5. Damit können einzelne Anwenderzyklen gezielt vorverarbeitet werden, welche den Befehl `PREPRO` nach der `PROC`-Anweisung haben.

Vorübersetzte Zyklen werden standardmäßig im dynamischen NC-Speicher abgelegt. Das MD10700 `$MN_PREPROCESSING_LEVEL` wurde erweitert um Bit 6. Damit kann festgelegt werden, dass die jetzt im dynamischen NC-Speicher abgelegten kompilierten Programme, die dort nicht mehr genug Platz haben, im statischen NC-Speicher abgelegt werden dürfen.

Funktionalität

Die in den Verzeichnissen für Standard- und Anwenderzyklen befindlichen Programme werden bei Power On vorverarbeitet, d. h. das Teileprogramm wird in einen bearbeitungsoptimalen binären Zwischencode übersetzt (kompiliert). Beim Aufruf wird dann das Kompilat bearbeitet.

Laufzeitoptimierung

Die Vorverarbeitung ist vor allem zur Laufzeitoptimierung von Teileprogrammen mit Hochsprachenanteilen (Sprüngen, Kontrollstrukturen, Bewegungssynchronaktionen) geeignet.

Während bei der standardmäßig aktiven Interpretation des ASCII-Teileprogramms Sprünge und Kontrollstrukturen durch Suchen in allen Sätzen (Satzanfang) aufgelöst werden, erfolgt beim vorverarbeiteten Teileprogramm der Sprung direkt auf den Zielsatz.

Somit sind die Laufzeitunterschiede zwischen Sprüngen und Kontrollstrukturen aufgehoben.

Beispiel für Laufzeit einer Vorverarbeitung:

Laufzeitreduzierung um 30 % bei aktivem Kompressor

```
DEF INT ZAEHLER
Ziel: G1 G91 COMPON
G1 X0.001 Y0.001 Z0.001 F100000
ZAEHLER=ZAEHLER +1
ZAEHLER=ZAEHLER -1
ZAEHLER=ZAEHLER +1
IF ZAEHLER<= 100000 GOTOB ZIEL
```

Rechenintensive Programme sowie Programme mit symbolischen Namen werden schneller bearbeitet.

Laufzeitkritische Stellen (z. B. die Fortsetzung der Bearbeitung nach Restweglöschen oder Vorlaufstopp in Zyklen) können schneller bearbeitet werden.

Wenn die Interruptroutine als vorverarbeiteter Zyklus vorliegt, kann die Bearbeitung nach der Programmunterbrechung schneller fortgesetzt werden.

8.2 Programmhandling

Aktivierung/Deaktivierung

Die Vorverarbeitung von Zyklen erfolgt bei Power On, wenn das folgende Maschinendatum gesetzt ist:

MD10700 \$MN_PREPROCESSING_LEVEL, Bit1 (Programmvorverarbeitungsstufe)

Bit	Wert	Bedeutung
0		Keine Vorverarbeitung
	0	Aufrufbeschreibung von Zyklen ist nicht standardmäßig bekannt. Zyklen müssen wie normale Unterprogramme vor dem Zyklenaufwurf als Extern erklärt werden. Das ist eine sinnvolle Einstellung, wenn keine Zyklen mit Aufrufparametern verwendet werden.
	1	Im Steuerungshochlauf wird die Aufrufbeschreibung der Zyklen gebildet. Alle Anwenderzyklen (Directory_N_CUS_DIR) und Siemenszyklen (Directory_N_CST_DIR) mit Übergabeparametern können ohne Extern-Erklärung aufgerufen werden. Änderungen an der Zyklen-Aufrufchnittstelle werden erst mit Power On wirksam. Es sind folgende Maschinendaten einzustellen: MD18170 \$MN_MM_NUM_MAX_FUNC_NAMES (Anzahl von Zusatzaktionen) MD18180 \$MN_MM_NUM_MAX_FUNC_PARAM (Anzahl von zusätzlichen Parametern für Zyklen)
1	1	Im Steuerungshochlauf werden alle Zyklen in ein bearbeitungsoptimales Kompilat vorverarbeitet. Alle Anwender-Zyklen (Directory_N_CUS_DIR) und Standard-Zyklen (Directory_N_CST_DIR) werden schnell abgearbeitet. Änderungen an den Zyklen-Programmen werden erst beim nächsten Power On wirksam.
2	1	Im Steuerungshochlauf werden die Standardzyklen aus dem Verzeichnis _N_CST_DIR in ein bearbeitungsoptimales Kompilat vorverarbeitet.
3	1	Im Steuerungshochlauf werden die Anwenderzyklen aus dem Verzeichnis _N_CUS_DIR in ein bearbeitungsoptimales Kompilat vorverarbeitet.
4	1	Vorverarbeitung der Anwenderzyklen aus dem Directory_N_CMA_DIR

5	1	Vorverarbeitung der Anwenderzyklen mit dem Befehl <code>PREPRO</code> in der PROC-Anweisungszeile. Nicht gekennzeichnete Dateien der durch Bit 1-4 bezeichneten Verzeichnisse werden nicht vorverarbeitet. Ist das Bit 0, dann erfolgt die Steuerung des Vorverarbeitens ausschließlich nach den Vorgaben der Bits 0-4.
6	0	Das Kompilat wird im dynamischen NC-Speicher abgelegt, solange noch freier Speicher vorhanden ist. Reicht dieser nicht, so wird die Vorverarbeitung abgebrochen. Bemessung des dynamischen NC-Speichers mit dem Maschinendatum: MD18351 \$MN_MM_DRAM_FILE_MEM_SIZE.

Die durch das Kompilat im dynamischen NC-Speicher belegten Bereiche sind für den Anwender sichtbar.

Kombinationen der Bits sind zulässig.

Kompilieren

Die in den Directories Standard-Zyklen: `_N_CST_DIR`, `_N_CMA_DIR` und Anwender-Zyklen: `_N_CUS_DIR` befindlichen Unterprogramme (Extension `_SPF`) und ggf. die mit `PREPRO` gekennzeichneten Unterprogramme werden kompiliert. Das Kompilat hat den Namen des Original-Zyklus mit Extension `_CYC`.

Hinweis

Programmänderungen an vorkompilierten Programmen werden erst nach dem nächsten Power On wirksam!

Zugriffsrecht

Das vorverarbeitete Programm ist nur ausführbar, nicht les- und schreibbar. Das Kompilat kann nicht verändert oder archiviert werden. Die Original-Zyklen `_SPF` Dateien bleiben erhalten.

Bei Änderung des ASCII-Zyklus wird das Kompilat nicht verändert, d. h. Änderungen werden erst nach dem nächsten Power On wirksam.

Speicherbedarf

Der Speicherbedarf für kompilierte Zyklen ist etwa Faktor 2 zusätzlich zu dem ASCII-Teilprogramm.

Der Speicherbedarf für die in den Teileprogrammen definierte Variable wird über die folgenden bestehenden Maschinendaten festgelegt:

MD28020 \$MC_MM_NUM_LUD_NAMES_TOTAL (Anzahl der lokalen Anwendervariablen)

MD28010 \$MC_MM_NUM_REORG_LUD_MODULES (Anzahl der Bausteine für lokale Anwendervariablen bei REORG)

MD28040 \$MC_MM_LUD_VALUES_MEM (Speichergröße für lokale Anwendervariablen)

MD18242 \$MC_MM_MAX_SIZE_OF_LUD_VALUE (Speicherblockgröße für LUD-/GUD-Werte)

Weitere Informationen finden Sie unter Speicherkonfiguration (Seite 837).

Der Speicherbedarf zum Zeitpunkt der Vorverarbeitung ist so groß, als würde das vorverarbeitete Programm in der ersten Unterprogrammebene aufgerufen.

Zum Zeitpunkt der Vorverarbeitung bei Power On wird für jedes Sprungziel/Label sowie für jedes Kontrollstrukturelement ein Name wie für eine Variable gezählt und muss in dem folgenden Maschinendatum berücksichtigt werden:

MD28020 \$MC_MM_NUM_LUD_NAMES_TOTAL (Anzahl der lokalen Anwendervariablen)

Beispiel

Programmcode	Kommentar
PROC NAMES	;
DEF INT VARIABLE, FELD[2]	; 2 Namen
ANFANG:	; 1 Name, nur für Vorverarbeitung
FOR VARIABLE = 1 TO 9	; 1 Name, nur für Vorverarbeitung
G1 F10 X=VARIABLE*10-56/86EX4+4*SIN(VARIABLE/3)	
ENDFOR	; 1 Name, nur für Vorverarbeitung
M17	

Um dieses Programm normal abzuarbeiten, muss das folgende Maschinendatum mindestens 3 Namen vereinbaren:

MD28020 \$MC_MM_NUM_LUD_NAMES_TOTAL

Um dieses Programm bei Power On zu kompilieren, sind 6 Namen notwendig.

Vorverarbeitete Programme/Zyklen werden im dynamischen NC-Speicher abgelegt. Der Platzbedarf je Programm ist unverändert wie oben skizziert zu überschlagen. Abstimmung auf die Platzbelegung im statischen NC-Speicher ist nur dann erforderlich, wenn Bit 6 = 1 im folgenden Maschinendatum gesetzt ist:

MD10700 \$MN_PREPROCESSING_LEVEL (Programmverarbeitungsstufe)

In diesem Falle werden die Kompilate der Programme, die im dynamischen NC-Speicher keinen Platz mehr vorfinden, im statischen NC-Speicher untergebracht.

Beispiele für entsprechende Maschinendateneinstellungen finden Sie unter "Beispiele" im Kapitel "Vorverarbeitung im dynamischen NC-Speicher".

8.3 Programmaufruf

Übersicht

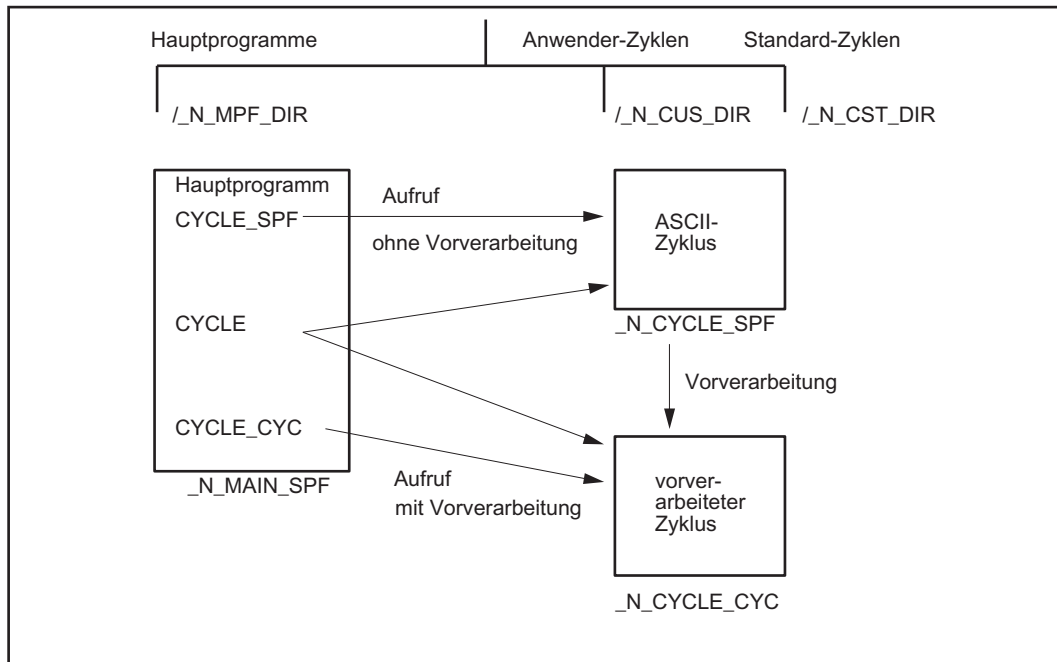


Bild 8-1 Erzeugung und Aufruf vorverarbeiteter Zyklen ohne Parameter

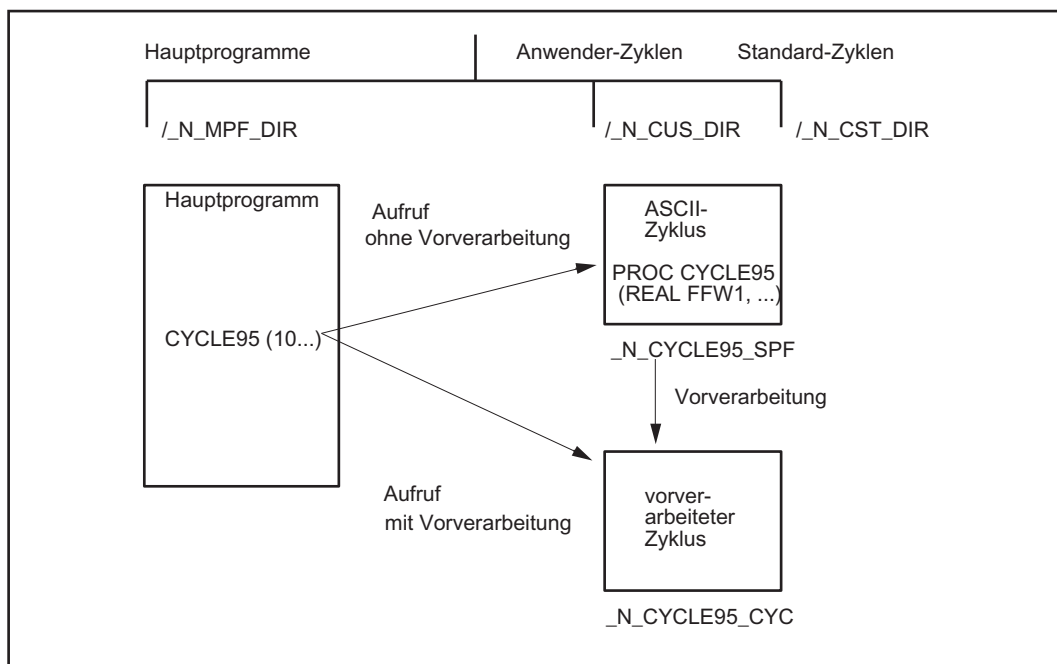


Bild 8-2 Erzeugung und Aufruf vorverarbeiteter Zyklen mit Parameter

Aufruf

- Kompilierter Zyklus: Der Aufruf des kompilierten Zyklus unterscheidet sich nicht vom normalen Unterprogrammaufruf.
Beispiel: CYCLE
- Vorverarbeitung ist aktiviert: Anstelle des ASCII-Zyklus wird der kompilierte Zyklus aufgerufen.
 - Erfolgt der Unterprogrammaufruf explizit mit der Extension `_SPF`, so wird der ASCII-Zyklus aufgerufen, auch wenn ein Kompilat vorhanden ist.
Beispiel: `CYCLE_SPF`; Aufruf des ASCII-Zyklus
 - Erfolgt der Unterprogrammaufruf explizit mit der Extension `_CYC`, so wird der vorverarbeitete Zyklus, sofern vorhanden, aufgerufen. Ist kein Kompilat vorhanden erfolgt Fehlermeldung.
Beispiel: `CYCLE_CYC`; Aufruf des vorkompilierten Zyklus
 - Wird bei aktiviertem Bit 5 eine Datei, die nicht mit `PREPRO` gekennzeichnet ist, explizit mit der Extension `_CYC` aufgerufen, so erfolgt eine Fehlermeldung mit Alarm 14011.
- Wenn ein Unterprogramm aufgerufen wird ohne explizite Extension, so wird zuerst versucht, das kompilierte Programm zu laden, wenn dies nicht möglich ist (wenn nicht mit `PREPRO` gekennzeichnet), so wird versucht das SPF-Programm zu laden.
- Der Wechsel in den externen Sprachmodus durch `G291` wird mit Alarm abgelehnt. Beim Aufruf eines vorkompilierten Zyklus wird explizit in den Siemens-Sprachmodus gewechselt.
- Beim Unterprogrammaufruf wird überprüft, ob das Kompilat älter ist als der Zyklus. Wenn dies der Fall ist, so wird das Kompilat gelöscht und ein Alarm abgesetzt, so dass der Anwender die Zyklen neu vorverarbeiten lassen muss.

Hinweis

Der Aufruf von Zyklen mit der Extension `_SPF` oder `_CYC` ist nur für Zyklen ohne Parameter zulässig

In Zyklen die vorverarbeitet werden, dürfen keine PUDs verwendet werden. Die PUDs werden im aufrufenden Hauptprogramm angelegt. Zum Übersetzungszeitpunkt nach Power On sind diese Daten in den Zyklen nicht bekannt.

In der aktuellen Programmanzeige ist erkennbar, ob der aktuelle ASCII-Zyklus oder das Kompilat aufgerufen wurde (Extension `_SPF` oder `_CYC`).

Aufrufbedingung

Wenn Vorverarbeitung aktiviert ist, müssen alle Zyklen in den Zyklendirectories kompiliert sein. Nicht kompilierte Zyklen in `_N_CUS_DIR` und `_N_CST_DIR`, die z. B. erst nach Power On eingespielt wurden, können nur mit expliziter Angabe Extension `_SPF` aufgerufen werden.

Bei aktiver Vorverarbeitung und Bit 5 werden alle Programme, die nicht mit der Proc-Anweisung `PREPRO` beginnen, nicht vorübersetzt.

Syntax-Check

Alle Programmfehler, die mit Korrektursatz korrigiert werden können, werden bereits zum Zeitpunkt der Vorverarbeitung erkannt. Zusätzlich wird bei Verwendung von Sprüngen und Kontrollstrukturen überprüft, ob die Sprungziele vorhanden sind und ob die Schachtelung von Kontrollstrukturen korrekt ist.

Sprungziele/Labels müssen im Programm eindeutig sein.

Nach Korrektur der bei der Vorverarbeitung erkannten Fehler muss die Vorverarbeitung durch NC Power On erneut gestartet werden.

8.4 Randbedingungen

Verfügbarkeit der Funktion "Vorverarbeitung"

Die Funktion ist eine Option ("Programmvorverarbeitung"), die über das Lizenzmanagement der Hardware zugeordnet werden muss.

Sprachumfang

Es steht der volle Sprachumfang der NC-Sprache im Teileprogramm zur Verfügung.

Die Verrechnung gemessener Prozessgrößen und die Reaktion auf Signale aus dem Prozess und aus anderen Kanälen (Override, Restweglöschen, Bewegungssynchronaktionen, Kanalkoordinierung, Interruptbearbeitung usw.) ist uneingeschränkt möglich.

Achsname

Das Kompilieren von Teileprogrammen erfolgt kanalunabhängig. Deshalb müssen die über die folgenden Maschinendaten eingestellten Geometrie- und Kanalnamen in allen Kanälen **gleich** sein, wenn sie in den **vorkompilierten Zyklen direkt verwendet werden**:

MD20060 \$MC_AXCONF_GEOAX_NAME_TAB (Geometrieachsname im Kanal)

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB (Kanalachsname im Kanal)

Typischerweise werden in Bearbeitungszyklen Achsnamen nicht direkt verwendet, da Zyklen folgendermaßen geschrieben werden:

- kanalunabhängig und
- unabhängig von den an der Maschine definierten Achsnamen.

Die zu verfahrenen Achsen werden indirekt über Maschinendaten angesprochen oder als Parameter übergeben:

- Indirekte Achsprogrammierung:
 - IF \$AA_IM[AXNAME(\$MC_AXCONF_CHANAX_NAME_TAB[4])] > 5
; Dieser Zweig wird durchlaufen, wenn der Istwert der 5. Kanalachse
; bezogen auf das Maschinenkoordinatensystem größer als 5 ist.
 - G1 AX[AXNAME(\$MC-AXCONF-GEOAX-NAME-TAB[0])] = 10
F1000 G90.
; Verfahre die 1. Geometrieachse auf den Wert 10.
ENDIF
- Übergabe der zu verfahrenen Achse aus dem Hauptprogramm:
 - Zyklendefinition
PROC BOHRE(Axis BOHRACHSE)
WHILE \$AA_IW[BOHRACHSE] > -10
G1 G91 F250 AX[BOHRACHSE] = -1
ENDWHILE
 - Aufruf aus dem Hauptprogramm
BOHRE(Z)

8.5 Beispiele

8.5.1 Vorverarbeitung einzelner Dateien

Programmcode	Kommentar
PROC UP1 PREPRO	; Vorverarbeitung, wenn Bit 5 = 1
	; in PREPROCESSING_LEVEL
N1000 DEF INT ZAEHLER	
N1010 ZIEL: G1 G91 COMPON	
N1020 G1 X0.001 Y0.001 Z0.001 F100000	
N1030 ZAEHLER=ZAEHLER+1	
N1040 ZAEHLER=ZAEHLER-1	
N1050 ZAEHLER=ZAEHLER+1	
N1060 IF ZAEHLER <=10 GOTOB ZIEL	
N1070 M30	

Programmcode	Kommentar
PROC UP2	
N2000 DEF INT VARIABLE, FELD[2]	
N2010 IF \$AN_NCK_Version < 3.4	
N2020 SETAL(61000)	
N2030 ENDIF	
N2040 ANFANG:	
N2050 FOR VARIABLE = 1 TO 5	
N2060 G1 F1000 X=VARIABLE*10-56/86EX4+4*SIN(VARIABLE/3)	
N2070 ENDFOR	
N2080 M17	
PROC MAIN	
N10 G0 X0 Y0 Z0	
N20 UP1	
N30 G0 X10 Y10 Z10	
N40 UP2	
N50 G0 X100 Y100	
N60 UP3	
N70 G0 X10 Y10	
N80 M30	

Beispielkonstellationen:

a) Bit 5 = 1

MD10700 \$MN_PREPROCESSING_LEVEL=45 ; Bit 0, 2, 3, 5

Unterprogramm UP1 wird vorübersetzt, die Aufrufbeschreibung wird gebildet.

Unterprogramm UP2 wird nicht vorübersetzt, die Aufrufbeschreibung wird aber gebildet.

b) Bit 5 = 0

MD10700 \$MN_PREPROCESSING_LEVEL=13 ; Bit 0, 2, 3,

Beide Unterprogramme werden vorübersetzt, die Aufrufbeschreibung wird gebildet.

c) Beispiel für ein **ungültiges** Unterprogramm bei aktiviertem Kompilieren:

Programmcode	Kommentar
PROC SUB1 PREPRO	
G291	; ← Alarm beim Kompilieren, G291 nicht möglich
G0 X0 Y0 Z0	
M17	

8.5.2 Vorverarbeitung im dynamischen NC-Speicher

Maschinendaten für Vorverarbeitung nur im dynamischen NC-Speicher mit selektiver Auswahl:

Programmcode	Kommentar
	; Bit 5 =1 Selektive Programmauswahl
	; Bit 6 =0 Kein Ausweichen auf
	; statischen NC-Speicher, wenn
	; dynamischer NC-Speicher voll
N30 \$MN_MM_DRAM_FILE_MEM_SIZE = 800	; Platz vorhalten
N40 \$MN_PREPROCESSING_LEVEL = 63	; Bit 0-5 =1
M17	

Maschinendaten für Vorverarbeitung im dynamischen NC-Speicher mit Option, den statischen NC-Speicher zu nutzen und mit selektiver Auswahl:

Programmcode	Kommentar
	; Bit 5 =1 Selektive Programmauswahl
	; Bit 6 =1 Ausweichen auf statischen
	; NC-Speicher, wenn dynamischer NC-
	; Speicher voll
N30 \$MN_MM_DRAM_FILE_MEM_SIZE = 800	; Platz vorhalten
N40 \$MN_PREPROCESSING_LEVEL = 127	; Bit 0-6 = 1
M17	

8.6 Datenlisten

8.6.1 Maschinendaten

8.6.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10700	PREPROCESSING_LEVEL	Programmvorverarbeitungsstufe
18242	MM_MAX_SIZE_OF_LUD_VALUE	Maximale Feldgröße der LUD-Variablen

8.6.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
28010	MM_NUM_REORG_LUD_MODULES	Anzahl der Bausteine für lokale Anwendervariablen bei REORG (DRAM)
28020	MM_NUM_LUD_NAMES_PER_PROG	Anzahl der lokalen Anwendervariablen (DRAM)
28040	MM_LUD_VALUES_MEM	Speichergröße für lokale Anwendervariablen (DRAM)

TE7: Wiederaufsetzen (Retrace Support)

9.1 Kurzbeschreibung

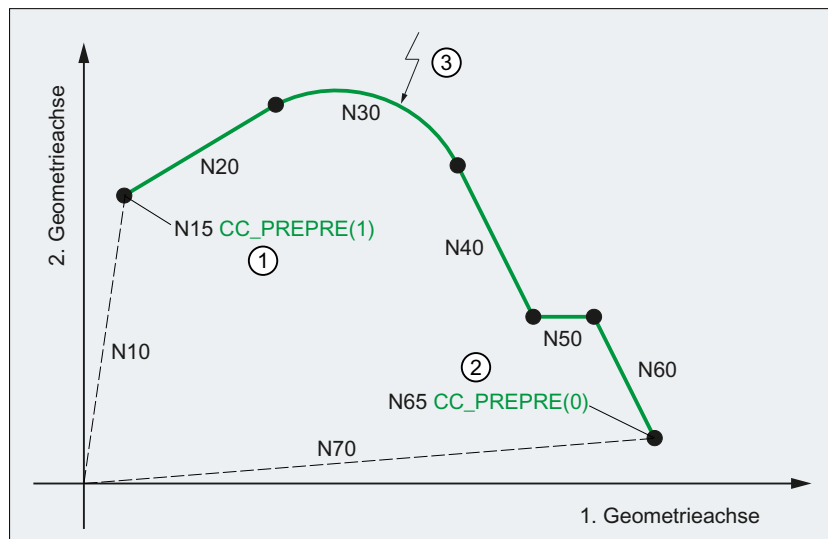
Funktion

Die Technologie-Funktion "Wiederaufsetzen - Retrace Support (RESU)" unterstützt das Wiederaufnehmen von unterbrochenen 2-dimensionalen Bearbeitungsvorgängen, wie z. B. Laserschneiden, Wasserstrahlschneiden etc.

RESU ermöglicht es dem Maschinenbediener, bei einer Störung des Bearbeitungsvorgangs, z. B. Ausfall des Laserstrahls, ohne genaue Kenntnis des aktiven Teileprogramms die Bearbeitung zu unterbrechen und vom Unterbrechungspunkt bis zu einem bearbeitungstechnisch notwendigen Wiederaufsetzpunkt entlang der Kontur zurückzufahren.

Nach Erreichen des Wiederaufsetzpunkts löst der Maschinenbediener das Wiederaufsetzen aus. Im Rahmen des Wiederaufsetzens erfolgt implizit ein Satzsuchlauf mit Berechnung an der Kontur mit anschließendem Repositionieren an die Kontur und automatischer Wiederaufnahme der Teileprogrammbearbeitung.

Die Möglichkeit zum Wiederaufsetzen innerhalb eines Bearbeitungsprogramms wird durch die Programmierung von RESU-Start- und Endpunkten über die Prozedur CC_PREPRE(...) erreicht. Innerhalb eines so definierten Konturbereichs kann der Wiederaufsetzpunkt beliebig gewählt werden.



- ① RESU-Startpunkt bzw. Anfang des RESU-fähigen Konturbereichs
- ② RESU-Endpunkt bzw. Ende des RESU-fähigen Konturbereichs
- ③ Beispielhafter Wiederaufsetzpunkt

Bild 9-1 Programmierte Kontur mit Wiederaufsetz- und Unterbrechungspunkt

9.2 Funktionsbeschreibung

Ein konturgenaues Rückwärtsfahren ist auf allen Konturen möglich, die sich aus programmierten Geraden- und Kreiselementen zusammensetzen. Andere Konturelemente wie z. B. Splines oder automatisch eingefügte nichtlineare Konturelemente (Kreis, Parabel etc. z. B. durch Werkzeugradiuskorrektur), werden beim Rückwärtsfahren als Gerade zwischen Anfangs- und Endpunkt des entsprechenden Konturelements abgebildet und erlauben daher kein konturgenaues Rückwärtsfahren.

Funktionskürzel

Das Kürzel der Technologie-Funktion "Wiederaufsetzen - Retrace Support" für funktionsspezifische Bezeichner von Programmbefehlen, Maschinendaten etc. ist:

RESU (= REtrace SUpport)

Einschränkungen

Für die Anwendung der Technologie-Funktion "Wiederaufsetzen - Retrace Support" gelten folgende Einschränkungen:

- **Die Technologie-Funktion ist nur im 1. Kanal der NC verfügbar.**
- Wiederaufsetzen bzw. Rückwärtsfahren ist nur für Teileprogrammsätze möglich, die Verfahransätze in der projektierten RESU-Arbeitsebene (z. B. 1. und 2. Geometrieachse des Kanals, siehe Kapitel "Festlegung der RESU-Arbeitsebene (Seite 597)") enthalten.

Weitere Informationen

Die Technologie-Funktion "Wiederaufsetzen - Retrace Support" ist ein Compile-Zyklus. Für die Handhabung von Compile-Zyklen (siehe Funktionshandbuch *Technologien*, Kapitel *Installation und Aktivierung ladbarer Compile- Zyklen*).

9.2 Funktionsbeschreibung

9.2.1 Funktion

Satzsuchlauf mit Berechnung an der Kontur

Um an einer bestimmten Stelle eines Teileprogramms mit der unterbrochenen Bearbeitung wieder aufsetzen zu können, besteht die Möglichkeit, einen Satzsuchlauf mittels der Standard-Funktion "Satzsuchlauf mit Berechnung an der Kontur" zu verwenden. Dazu ist aber die genaue Kenntnis des Teileprogramms notwendig, um die zum Satzsuchlauf erforderliche Satznummer des Teileprogrammsatzes, auf den der Satzsuchlauf erfolgen soll, angeben zu können.

Wiederaufsetzen - Retrace Support

Die Technologie-Funktion "Wiederaufsetzen - Retrace Support" unterstützt das Wiederaufsetzen des Bearbeitungsvorgangs durch einen impliziten Satzsuchlauf mit Berechnung an der Kontur, ohne dass der Maschinenbediener den dafür notwendigen Teileprogrammsatz kennen muss.

Ein Wiederaufsetzen ist zum Beispiel erforderlich, wenn beim Laserschneiden der Laserstrahl während des Bearbeitungsvorganges ausfällt und an der Unterbrechungsstelle mit der Bearbeitung wieder aufgesetzt werden soll.

RESU unterstützt das Wiederaufsetzen durch folgende, automatisch ablaufende Teilfunktionen:

- Funktionsspezifisches Rückwärtsfahren auf der Kontur bis zum gewünschten Wiederaufsetzpunkt
- Automatische Ermittlung des zum Wiederaufsetzpunkt gehörenden Teileprogrammsatzes
- Satzsuchlauf mit Berechnung an der Kontur auf den ermittelten Teileprogrammsatz
- Repositionieren an die Kontur im Wiederaufsetzpunkt
- Fortsetzen der Teileprogrammbearbeitung

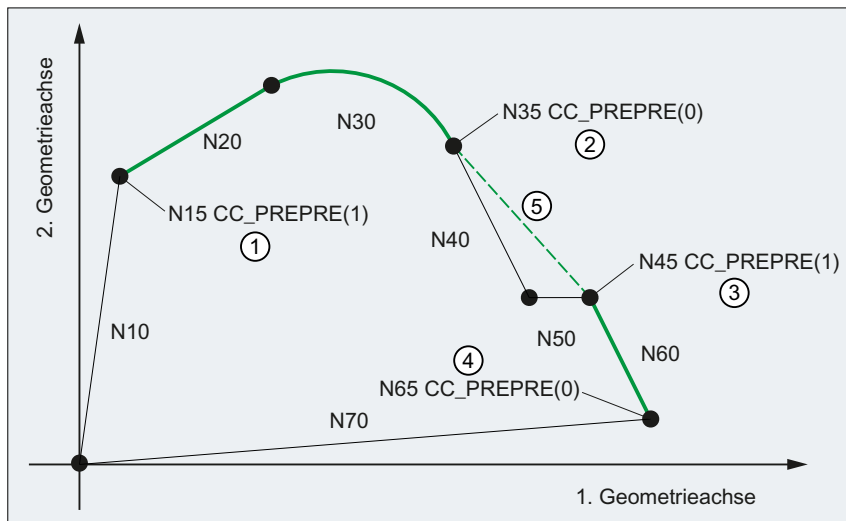
Um den erforderlichen Wiederaufsetzpunkt exakt anfahren zu können, ist während des Wiederaufsetzens auch ein mehrfacher Wechsel zwischen Rückwärts- und Vorwärtsfahren auf der Kontur möglich.

RESU-fähige Konturbereiche

RESU wird durch Programmierung des funktionsspezifischen Teileprogrammbefehls `CC_PREPRE (1)` eingeschaltet. Nur der Konturbereich ist im Sinne von RESU wiederaufsetzfähig (RESU-fähig), der zwischen dem RESU-Start (`CC_PREPRE (1)`) und dem Unterbrechungspunkt (NC-Stopp) liegt.

Nach RESU-Start werden alle Teileprogrammsätze, in denen Verfahrbewegungen programmiert sind, von RESU zum eventuellen späteren Rückwärtsfahren protokolliert. Konturbereiche, für die ein Wiederaufsetzen keinen Sinn macht, können mittels RESU-Stopp `CC_PREPRE (0)` vom Protokollieren ausgenommen werden.

Nicht protokollierte Konturbereiche werden beim Rückwärts- / Vorwärtsfahren durch Geraden zwischen Anfangs- und Endpunkt der protokollierten Konturbereiche überbrückt.



- ① RESU-Startpunkt 1 bzw. Anfang des RESU-fähigen Konturbereichs 1
- ② RESU-Endpunkt 1 bzw. Ende des RESU-fähigen Konturbereichs 1
- ③ RESU-Startpunkt 2 bzw. Anfang des RESU-fähigen Konturbereichs 2
- ④ RESU-Endpunkt 2 bzw. Ende des RESU-fähigen Konturbereichs 2
- ⑤ RESU-Gerade als Ersatzkontur des **nicht** RESU-fähigen Konturbereichs

Bild 9-2 RESU-fähige Konturbereiche

9.2.2 Begriffsdefinitionen

Unterbrechungspunkt

Der Unterbrechungspunkt ist der Punkt der Kontur, an dem die Verfahrbewegung nach NC-Stopp zum Stillstand kommt und das Rückwärtsfahren ausgelöst wird.

Wiederaufsetzpunkt

Der Wiederaufsetzpunkt ist der Punkt der Kontur, auf dem das Rückwärtsfahren beendet und das Wiederaufsetzen ausgelöst wird.

RESU-fähiger Konturbereich

RESU-fähige Konturbereiche setzen sich aus Verfahrstrichen in der projektierten RESU-Arbeitsebene (z. B. 1. und 2. Geometrieachse des Kanals) zusammen, die im Teileprogramm zwischen dem RESU-Start-Befehl `CC_PREPRE(1)` und dem RESU-Stopp-Befehl `CC_PREPRE(0)` programmiert sind (siehe "Bild 9-2 RESU-fähige Konturbereiche (Seite 592)").

9.2.3 Funktionsablauf (Prinzip)

Im Folgenden ist der prinzipielle Ablauf der Funktion RESU zwischen Unterbrechungspunkt, Wiederaufsetzpunkt und dem Fortsetzen der Teileprogrammbearbeitung beschrieben.

Voraussetzungen

Ein Teileprogramm mit Verfahrshätzen in der projektierten RESU-Arbeitsebene (Seite 597) sowie dem Befehl für den RESU-Start ist im 1. Kanal gestartet.

Funktionsablauf

1. Teileprogrammbearbeitung unterbrechen:
Die Teileprogrammbearbeitung bzw. die Verfahrbewegung kann eine beliebige Anzahl von Verfahrshätzen nach RESU-Start mittels NC-Stop unterbrochen werden.
2. Rückwärtsfahren anwählen:
Die Anwahl des Rückwärtsfahrens erfolgt per NC/PLC-Nahtstellensignal:
DB21, ... DBX0.1 = 1
3. Rückwärtsfahren:
Mit NC-Start wird die Kontur in der RESU-Arbeitsebene (Seite 597) rückwärts abgefahren. RESU wählt dazu an Stelle des aktuellen Bearbeitungsprogramms das automatisch erzeugte RESU-Hauptprogramm (Seite 605) an.
4. Rückwärtsfahren beenden:
Ist der gewünschte Wiederaufsetzpunkt auf der Kontur erreicht, wird das Rückwärtsfahren mittels NC-Stop beendet.
5. Vorwärtsfahren anwählen (optional):
Zum Vorwärtsfahren ist das Rückwärtsfahren abzuwählen per NC/PLC-Nahtstellensignal:
DB21, ... DBX0.1 = 0
6. Vorwärtsfahren (optional):
Mit NC-Start wird die Kontur in der RESU-Arbeitsebene (Seite 597) in Vorwärtsrichtung abgefahren.
7. Vorwärtsfahren beenden (optional):
Ist der gewünschte Wiederaufsetzpunkt auf der Kontur erreicht, wird das Vorwärtsfahren mittels NC-Stop beendet.

9.2 Funktionsbeschreibung

8. Wiederaufsetzen:
Das Wiederaufsetzen wird ausgelöst per PLC-Nahtstellensignal:
DB21, ... DBX0.2 = 1 (Wiederaufsetzen starten)
Zum Wiederaufsetzen wählt RESU automatisch das ursprüngliche Bearbeitungsprogramm an und veranlasst einen Satzvorlauf mit Berechnung bis zum Wiederaufsetzpunkt.
9. Fortsetzen der Teileprogrammbearbeitung:
Das Fortsetzen der Teileprogrammbearbeitung ab dem Wiederaufsetzpunkt erfolgt entsprechend der Standardfunktion "Satzvorlauf mit Berechnung" mittels zweier aufeinander folgender NC-Start-Kommandos.
Mit dem ersten NC-Start-Kommando werden die Aktionssätze ausgeführt. Das Wiederaufsetz-ASUP CC_RESU_BS_ASUP.SPF (Seite 609) wird ausgelöst mit Erreichen des letzten Aktionssatzes:
DB21, ... DBX32.6 == 1 (letzter Aktionssatz aktiv)
Mit dem zweiten NC-Start-Kommando wird der Anfahrsatz ausgeführt und anschließend die Teileprogrammbearbeitung fortgesetzt.

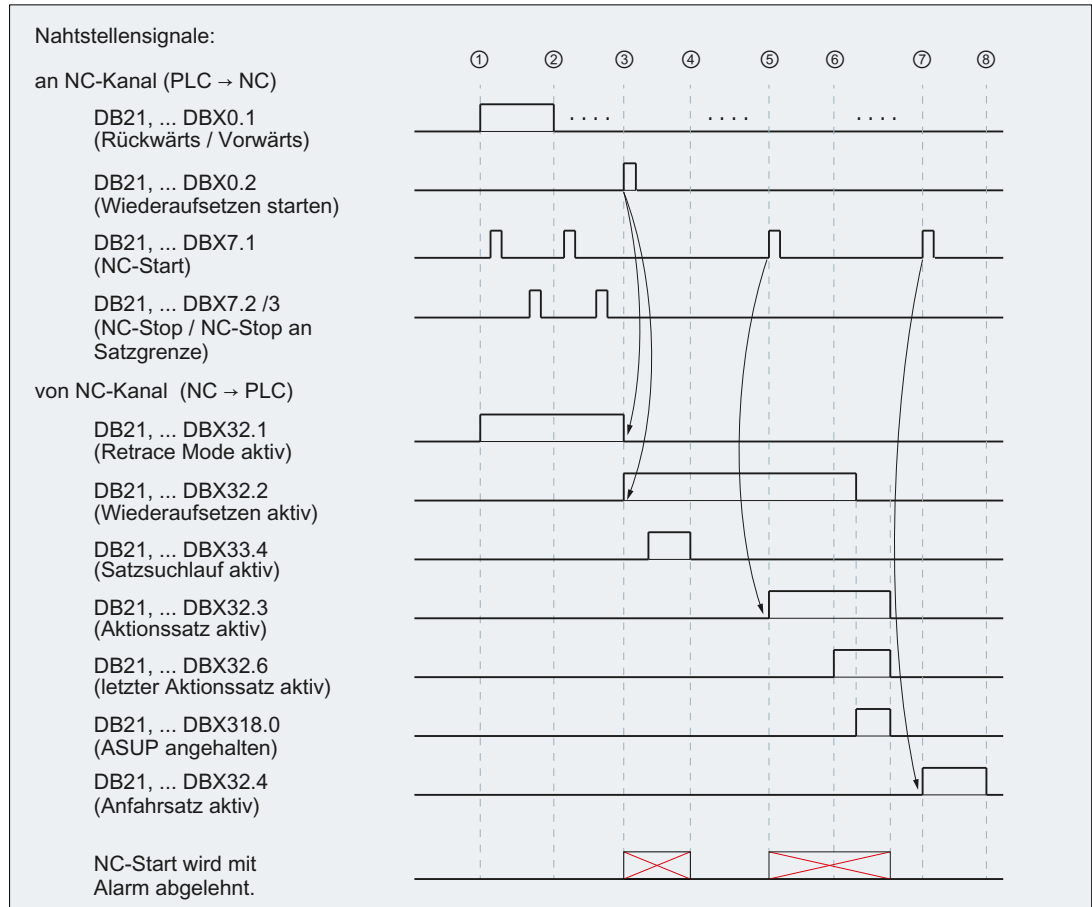
Hinweis

Die Punkte 3. bis 8. können beliebig oft wiederholt werden.

Nach dem Wiederaufsetzen ist ein erneutes Rückwärtsfahren maximal bis zum letzten Wiederaufsetzpunkt möglich.

Signalverlauf der NC/PLC-Nahtstellensignale

Der prinzipielle Ablauf der Funktion RESU ist im folgenden Bild als Signalverlauf der beteiligten NC/PLC-Nahtstellensignale dargestellt:



- ① Rückwärtsfahren wird gestartet.
- ② Vorwärtsfahren wird gestartet (optional).
- ③ Wiederaufsetzen wird gestartet (Satzsuchlauf).
- ④ Suchlaufziel (Zielsatz) wurde gefunden.
- ⑤ 1. NC-Start → Aktionssätze werden ausgegeben.
- ⑥ Letzter Aktionssatz wird aktiv.
Mit dem Aktivieren des letzten Aktionssatzes wird das RESU-ASUP CC_RESU_BS_ASUP.SPF (Seite 609) ausgelöst.
- ⑦ 2. NC-Start → Der Anfahrsatz zum Wiederaufsetzpunkt wird abgefahren.
- ⑧ Die Teileprogrammbearbeitung (Zielsatz) wird fortgesetzt

Bild 9-3 Signalverlauf

9.2.4 Maximaler RESU-fähiger Konturbereich

Beim mehrmaligen Wiederaufsetzen innerhalb eines Konturbereichs ist das Rückwärtsfahren auf der Kontur immer nur bis zum letzten Wiederaufsetzpunkt (**W**) möglich. Beim erstmaligen Rückwärtsfahren nach RESU-Start kann bis zum Anfang des Konturbereichs zurückgefahren werden.

Dieses Verhalten soll durch folgende Grafik veranschaulicht werden. Der Einfachheit halber sei der Unterbrechungspunkt (**U**) immer derselbe:

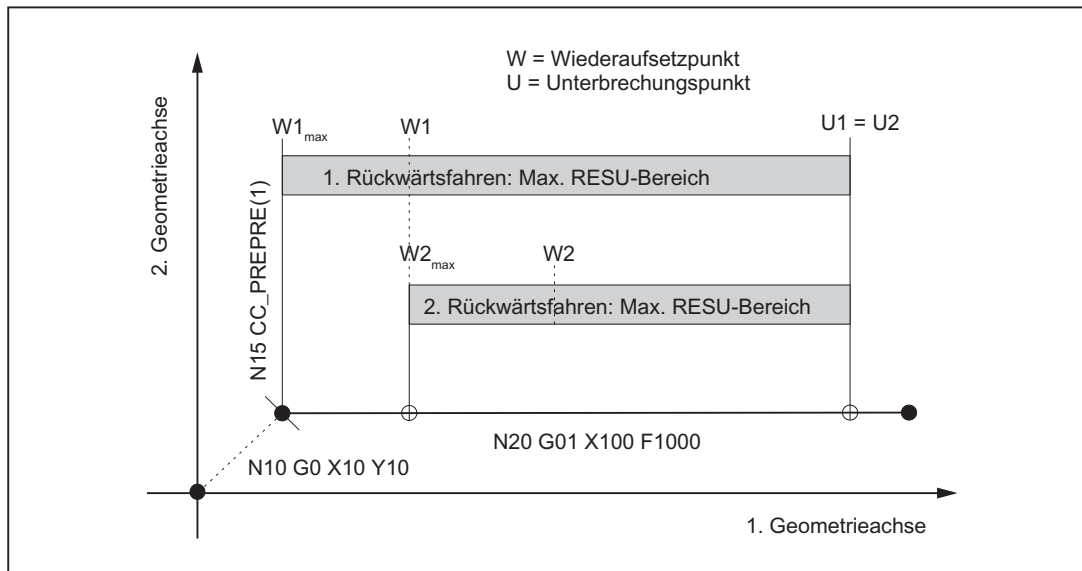


Bild 9-4 Maximaler RESU-fähiger Konturbereich

1. Rückwärtsfahren

Beim erstmaligen Rückwärtsfahren kann maximal bis zum Anfang des ersten Konturelements (N20) nach RESU-Start (N15) rückwärts gefahren werden ($W1_{max}$).

Wird bis zum Wiederaufsetzpunkt W1 zurückgefahren, legt W1 nach Wiederaufsetzen und Vorwärtsfahren den maximalen RESU-Bereich für ein eventuell weiteres Rückwärtsfahren fest.

2. Rückwärtsfahren

Ein erneutes Rückwärtsfahren kann jetzt nur noch maximal bis zum letzten Wiederaufsetzpunkt $W2_{max} = W1$ erfolgen.

Wird bis zum Wiederaufsetzpunkt W2 zurückgefahren, wird der maximale RESU-Bereich weiter eingeschränkt.

9.3 Inbetriebnahme

9.3.1 Aktivierung

Vor Inbetriebnahme der Technologie-Funktion ist sicherzustellen, dass der entsprechende Compile-Zyklus geladen und aktiviert ist (siehe auch Funktionshandbuch *Technologien*, Kapitel *Installation und Aktivierung ladbarer Compile- Zyklen*).

Aktivierung

Die Technologie-Funktion "Wiederaufsetzen - Retrace Support" wird über folgendes Maschinendatum aktiviert:

MD60900+i \$MN_CC_ACTIVE_IN_CHAN_RESU[0], Bit 0 = 1

Hinweis

Die Technologie-Funktion "Wiederaufsetzen - Retrace Support" ist nur im **1. Kanal** der NC verfügbar.

9.3.2 Festlegung der RESU-Arbeitsebene

Wiederaufsetzen bzw. Rückwärtsfahren ist nur für Teileprogrammsätze möglich, die Verfahrssätze in der projektierten RESU-Arbeitsebene enthalten.

Die Festlegung der RESU-Arbeitsebene erfolgt mit dem Maschinendatum:

MD62580 \$MC_RESU_WORKING_PLANE

Wert	Bedeutung
1	Die RESU-Arbeitsebene wird von der 1. und 2. Geometrieachse des 1. Kanals gebildet (für G17).
2	Die RESU-Arbeitsebene wird von der 1. und 3. Geometrieachse des 1. Kanals gebildet (für G18).
3	Die RESU-Arbeitsebene wird von der 2. und 3. Geometrieachse des 1. Kanals gebildet (für G19).

9.3.3 Speicherkonfiguration: Satzspeicher

Die Technologiefunktion benötigt **zusätzliche** Daten im NC-internen Satzspeicher. Für folgende speicherkonfigurierende kanalspezifischen Maschinendaten sind die Werte zu erhöhen:

- MD28090 \$MC_MM_NUM_CC_BLOCK_ELEMENTS += **4** (Anzahl Satzelemente für Compile-Zyklen)
- MD28100 \$MN_MM_NUM_CC_BLOCK_USER_MEM += **20** (Größe des Satzspeichers für Compile-Zyklen (DRAM) in kByte)

9.3.4 Speicherkonfiguration: Heap-Speicher

Speicherbedarf

RESU benötigt Compile-Zyklen-Heap-Speicher für folgende funktionspezifische Puffer:

- **Satzpuffer**
Je größer der Satzpuffer (siehe "Bild 9-6 RESU-spezifische Teileprogramme (Seite 605)") ist, desto mehr Teileprogrammätze können rückwärts gefahren werden.
Pro Teileprogrammätze werden 32 Byte benötigt.
Der Satzpuffer ist direkt parametrierbar.
- **Satzsuchlaufpuffer**
Im Satzsuchlaufpuffer liegen die zur Bearbeitung von Unterprogrammaufrufen im Rahmen von RESU benötigten Informationen.
Pro Unterprogrammaufruf werden 180 Bytes benötigt. Für den Satzsuchlaufpuffer werden daher mindestens 2880 Bytes benötigt (16 Unterprogrammebenen à 180 Bytes).
Der Satzsuchlaufpuffer ist nicht direkt parametrierbar.
Über eine funktionspezifische GUD-Variable wird die Größe des Satzsuchlaufpuffers angezeigt (zum Anlegen der GUD-Variablen, siehe Kapitel "Kanalspezifische GUD-Variable (Seite 614)").

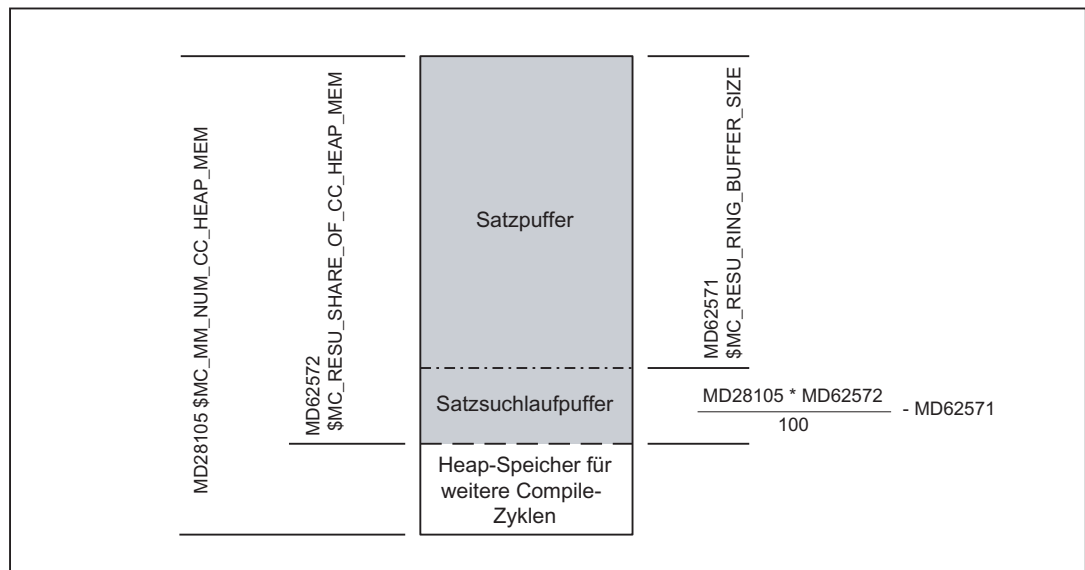


Bild 9-5 Aufteilung des Heap-Speichers für Compile-Zyklen

Speicherkonfiguration

Größe des Compile-Zyklen-Heap-Speichers

Die Größe des vom Anwender für Compile-Zyklen nutzbaren Heap-Speichers in kByte wird eingestellt über das speicherkonfigurierende kanalspezifische Maschinendatum:

MD28105 \$MC_MM_NUM_CC_HEAP_MEM

Für RESU wird der bereits vorhandene Maschinendatenwert (x) wie folgt angepasst:

MD28105 \$MC_MM_NUM_CC_HEAP_MEM = x + 50

Größe des Satzpuffers

Die Größe des Satzpuffers wird eingestellt über das Maschinendatum:

MD62571 \$MC_RESU_RING_BUFFER_SIZE

Standardmäßige Einstellung:

MD62571 \$MC_RESU_RING_BUFFER_SIZE = 1000

RESU-Anteil am gesamten Heap-Speicher

Der RESU-Anteil am gesamten, vom Anwender für Compile-Zyklen nutzbaren Heap-Speichers wird eingestellt über das Maschinendatum:

MD62572 \$MC_RESU_SHARE_OF_CC_HEAP_MEM

Standardmäßige Einstellung:

MD62572 \$MC_RESU_SHARE_OF_CC_HEAP_MEM = 100

Fehlermeldungen

Für den Satzsuchlaufpuffer müssen mindestens 2880 Bytes (entsprechend 16 Unterprogrammebenen à 180 Bytes) vorhanden sein. Ansonsten wird im Hochlauf der NC folgender Alarm angezeigt:

Alarm 75600 "Kanal 1 Retrace Support: falsche MD-Konfiguration, Fehler-Nr. 5"

Ist der Satzsuchlaufpuffer zur Laufzeit nicht ausreichend, wird folgender Alarm angezeigt:

Alarm 75606 "Kanal 1 retrace-fähige Kontur wurde verkürzt"

9.3.5 Speicherbereich des RESU-Hauptprogramms**Speicherkonfiguration**

Über das folgende Maschinendatum kann der Ablageort des RESU-Hauptprogramms CC_RESU.MPF (siehe Kapitel "Hauptprogramm (CC_RESU.MPF) (Seite 605)") eingestellt werden:

MD62574 \$MC_RESU_SPECIAL_FEATURE_MASK (Zusätzlich RESU-Eigenschaften)

Bit	Wert	Bedeutung
1	0	Das RESU-Hauptprogramm wird im dynamischen NC-Speicher abgelegt (Voreinstellung).
	1	Das RESU-Hauptprogramm wird im statischen NC-Speicher abgelegt.

Ablage im dynamischen NC-Speicher (Voreinstellung)

Wenn das RESU-Hauptprogramm im dynamischen NC-Speicher angelegt wird, dann muss der dem Anwender für die Dateiablage zur Verfügung stehende Speicherbereich wie folgt vergrößert werden:

MD18351 \$MN_MM_DRAM_FILE_MEM_SIZE = x^1 + 100

¹⁾ bereits vorhandener Maschinendatenwert

Ablage im statischen NC-Speicher

Wenn das RESU-Hauptprogramm im statischen NC-Speicher angelegt wird, dann bleibt es über POWER OFF hinaus erhalten. Da RESU das RESU-Hauptprogramm aber bei jedem Wiederaufsetzen neu erzeugt, wird diese Parametrierung nicht empfohlen.

9.3.6 Ablage der RESU-Unterprogramme

Ablage als Anwender- oder Hersteller-Zyklen

Die folgenden RESU-spezifischen Unterprogramme können als Anwender- oder Hersteller-Zyklen abgelegt werden:

- INI-Programm: CC_RESU_INI.SPF
- END-Programm CC_RESU_END.SPF
- Wiederaufsetz-ASUP CC_RESU_BS_ASUP.SPF
- RESU-ASUP CC_RESU_BS_ASUP.SPF

Die Einstellung erfolgt über das Maschinendatum:

MD62574 \$MC_RESU_SPECIAL_FEATURE_MASK (Zusätzlich RESU-Eigenschaften)

Bit	Wert	Bedeutung
2	0	Die RESU-spezifischen Unterprogramme werden als Anwender-Zyklen abgelegt (Voreinstellung).
	1	Die RESU-spezifischen Unterprogramme werden als Hersteller-Zyklen abgelegt.

Serieninbetriebnahme

Aufgrund der Voreinstellung von MD62574 Bit 2 werden im erstmaligen Hochlauf der NC nach Aktivierung der Technologie-Funktion die RESU-spezifischen Unterprogramme mit ihren voreingestellten Inhalten als Anwender-Zyklen abgelegt.

Wird anschließend eingestellt, dass die RESU-spezifischen Unterprogramme als Hersteller-Zyklen abgelegt werden, bleiben die bereits als Anwender-Zyklen angelegten RESU-spezifischen Unterprogramme auch nach einem erneuten Hochlauf erhalten und müssen gelöscht werden.

Zur Unterstützung der Serieninbetriebnahme kann eingestellt werden, dass als Anwender-Zyklen vorhandene RESU-spezifische Unterprogramme im Hochlauf der NC ohne Nachfrage gelöscht werden:

MD62574 \$MC_RESU_SPECIAL_FEATURE_MASK, Bit 3 = 1

9.3.7 ASUP-Freigabe

Hinweis

Voraussetzung für den Einsatz von ASUPs ist die Verfügbarkeit der Option "Betriebsartübergreifende Aktionen".

Zur Startfreigabe des RESU-spezifischen ASUP "CC_RESU_ASUP.SPF" während sich der Kanal im NC-Stopp-Zustand befindet, sind Maschinendaten wie folgt zu parametrieren:

MD11602 \$MN_ASUP_START_MASK, Bit 0 = 1 (Stoppgründe für ASUP ignorieren)

MD11604 \$MN_ASUP_START_PRIO_LEVEL = 1 (Prioritäten, ab der MD11602 wirksam ist)

9.3.8 PLC-Anwenderprogramm

Anforderungen

Zur Ablaufkoordination von RESU ist im PLC-Anwenderprogramm folgende Funktionalität vorzusehen:

```
IF      DB21, ... DBX32.2      "Wiederaufsetzen aktiv" == 1
THEN   DB21, ... DBX0.1      "Vorwärts/Rückwärts" = 0
       DB21, ... DBX0.2      "Wiederaufsetzen starten" = 0
```

```
IF      DB11, ... DBX0.7      "BAG-Reset" == 1
OR      DB21, ... DBX7.7      "Reset" == 1
THEN   DB21, ... DBX0.1      "Vorwärts/Rückwärts" = 0
       DB21, ... DBX0.2      "Wiederaufsetzen starten" = 0
```

Zur Sicherheit sollten folgende Signale zurückgesetzt werden:

```
IF      DB21, ... DBX0.2      "Wiederaufsetzen starten" == 1
THEN   DB21, ... DBX0.1      "Vorwärts/Rückwärts" = 0

IF      DB21, ... DBX0.1      "Vorwärts/Rückwärts" == 1
THEN   DB21, ... DBX0.2      "Wiederaufsetzen starten" = 0
```

Programmbeispiel

Der folgende Programmausschnitt realisiert die oben beschriebenen Anforderungen:

```
U      DB21, ... DBX32.2      // IF      "Wiederaufsetzen aktiv" == 1
R      DB21, ... DBX0.1      // THEN   "Vorwärts/Rückwärts" = 0
R      DB21, ... DBX0.2      //          "Wiederaufsetzen starten" = 0
```

```

O   DB11, ... DBX0.7           // IF     "BAG-Reset" == 1
O   DB21, ... DBX7.7           // OR     "Reset" == 1
R   DB21, ... DBX0.1           // THEN  "Vorwärts/Rückwärts" = 0
R   DB21, ... DBX0.2           //       "Wiederaufsetzen starten" = 0
U   DB21, ... DBX0.2           // IF     "Wiederaufsetzen starten" == 1
R   DB21, ... DBX0.1           // THEN  "Vorwärts/Rückwärts" = 0
U   DB21, ... DBX0.1           // IF     "Vorwärts/Rückwärts" == 1
R   DB21, ... DBX0.2           // THEN  "Wiederaufsetzen starten" = 0

```

9.4 Programmierung

9.4.1 RESU-Start/Stop/Reset (CC_PREPRE)

Über die Prozedur `CC_PREPRE` (**Prepare Retrace**) können für das Wiederaufsetzen die Modi "Starten", "Stoppen" und "Rücksetzen" aktiviert werden:

Syntax

```
CC_PREPRE (<Modus>)
```

Bedeutung

CC_PREPRE:	Wiederaufsetz-Modus	
	Alleine im Satz:	ja

<Modus>:	Mode	
	Datentyp:	INT
	Wertebereich:	-1, 0, 1
	Wert	Bedeutung
	-1	Startet die Protokollierung der Verfahrssätze. Die zum Rückwärtsfahren benötigten Informationen werden satzspezifisch in einem RESU-internen Satzpuffer protokolliert. Die Verfahrinformationen beziehen sich dabei auf die beiden Geometriachsen der RESU-Arbeitsebene, z. B. die 1. und 2. Geometrieachse des Kanals: MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[x]; mit x = 0 und 1 Bzw. bei aktiver Transformation: MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB[x]; mit x = 0 und 1
0	Stoppt die Protokollierung der Verfahrssätze. Nicht relevante Konturbereiche können somit von der Protokollierung ausgenommen werden. Derart ausgesparte Konturbereiche werden beim Rückwärtsfahren durch eine Gerade zwischen Anfangs- und Endpunkt überbrückt.	
1	Schaltet die Protokollierung der Verfahrssätze aus und löscht den funktionsinternen Satzpuffer. Vor dem Ausschaltzeitpunkt liegende Konturbereiche des Teileprogramms stehen für RESU damit nicht mehr zur Verfügung.	

RESET-Verhalten

Bei folgenden Reset-Ereignissen wird implizit `CC_PREPRE (-1)` ausgeführt:

- Power On (Warmstart)
- Kanal-Reset
- Programmende-Reset (M30)

9.5 RESU-spezifische Teileprogramme

9.5.1 Übersicht

Im Rahmen des Wiederaufsetzens werden folgende Programme automatisch generiert. In den gekennzeichneten Programmen können anwenderspezifische Inhalte eingebracht werden.

Funktion	Programmname	Änderbar
Hauptprogramm	CC_RESU.MPF	Nein
INI-Programm	CC_RESU_INI.SPF	Ja
END-Programm	CC_RESU_END.SPF	Ja

Funktion	Programmname	Änderbar
Wiederaufsetz-ASUP	CC_RESU_BS_ASUP.SPF	Ja
RESU-ASUP	CC_RESU_ASUP.SPF	Nein

Interner Aufbau

Das folgende Bild gibt einen Überblick über den internen Aufbau der Technologie-Funktion und den Zusammenhang der verschiedenen Teileprogramme.

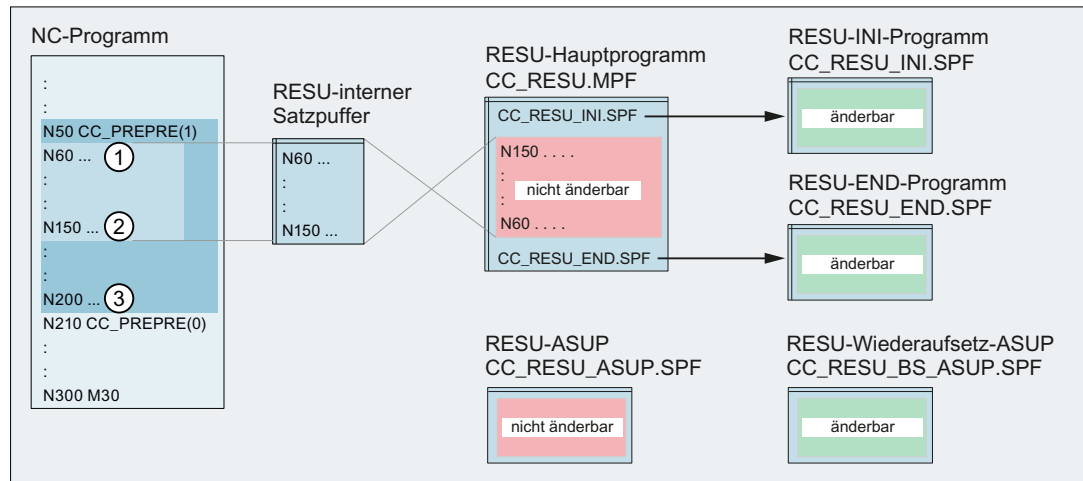


Bild 9-6 RESU-spezifische Teileprogramme

9.5.2 Hauptprogramm (CC_RESU.MPF)

Funktion

Das RESU-Hauptprogramm "CC_RESU.MPF" enthält neben den Aufrufen der RESU-spezifischen Unterprogramme die aus den protokollierten Verfahrssätzen des Satzpuffers erzeugten Verfahrssätze zum Rückwärts- / Vorwärtsfahren auf der Kontur. Es wird von RESU automatisch immer dann neu erzeugt, wenn nach Unterbrechung des Teileprogramms der Status des folgenden Nahtstellensignals wechselt:

DB21, ... DBX0.1 (Rückwärts / Vorwärts)

Hinweis

CC_RESU.MPF darf nicht verändert werden. Anwenderspezifische Anpassungen sind im entsprechenden RESU-spezifischen Unterprogramm durchzuführen.

Fehlermeldungen

Standardmäßig generiert RESU Verfahrssätze für den gesamten im Satzpuffer protokollierten RESU-fähigen Konturbereich. Ist zur Generierung aller Verfahrssätze im parametrisierten Speicherbereich des RESU-Hauptprogramms (siehe Kapitel "Speicherbereich des RESU-Hauptprogramms (Seite 600)") zuwenig Speicherplatz vorhanden, reduziert RESU die Anzahl der generierten Verfahrssätze.

Der fehlende Speicher bzw. die Reduzierung der generierten Verfahrssätze wird mit einem Alarm angezeigt:

RESU-Alarm 75608 "Kanal *Nummer* NC-Speichergrenze erreicht, RAM-Typ *Typ*"

Wird das RESU-Hauptprogramm im statischen Anwenderspeicher angelegt, wird gleichzeitig mit dem oben genannten RESU-Alarm folgender System-Alarm angezeigt:

Alarm 6500 "NC-Speichergrenze erreicht"

Hinweis

Wird aufgrund fehlenden Speichers die Anzahl der generierten Verfahrssätze reduziert, kann dennoch die gesamte RESU-fähige Kontur zum Wiederaufsetzen abgefahren werden. Dazu ist folgendes Vorgehen notwendig:

- Rückwärtsfahren bis zum Ende des RESU-Hauptprogramms
- Zweimaliger Wechsel des Nahtstellensignals:
DB21, ... DBX0.1 (Rückwärts / Vorwärts)

RESU generiert dadurch, ausgehend von der aktuellen Position als neuer Unterbrechungspunkt, ein neues RESU-Hauptprogramm.

Anschließend kann bis zum Ende des RESU-fähigen Konturbereichs oder bei erneuter Begrenzung bis zum Anfangspunkt des letzten generierbaren Verfahrssatzes gefahren werden.

Das beschriebene Vorgehen ist sowohl zum Rückwärts- als auch zum Vorwärtsfahren beliebig oft wiederholbar.

9.5.3 INI-Programm (CC_RESU_INI.SPF)

Funktion

Das RESU-spezifische Unterprogramm "CC_RESU_INI.SPF" enthält die für das Rückwärtsfahren notwendigen Voreinstellungen:

- Metrisches Eingabesystem: G71
- Maßangaben absolut: G90
- Ausschalten der einstellbaren Nullpunktverschiebungen / Frames (siehe Kapitel "Frames (Seite 619)"): G500
- Ausschalten der aktiven Werkzeugkorrekturen (siehe T0 Kapitel "Werkzeugkorrekturen (Seite 619)"): T0

- Ausschalten der Werkzeugradiuskorrektur: G40
- Verfahrgeschwindigkeit: F200

Programmstruktur

CC_RESU_INI.SPF hat folgenden voreingestellten Inhalt:

```

PROC CC_RESU_INI
    G71 G90 G500 T0 G40 F200
    ;vorhandene Systemframes werden deaktiviert
    ;Ist-Wert und Ankratzen
    if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H01'
    $P_SETFRAME = ctrans()
    endif
    ;externe Nullpunktverschiebung
    if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H02'
    $P_EXTFRAME = ctrans()
    endif
    ;Werkzeugtraeger
    if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H04'
    PAROTOF
    endif
    if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H08'
    TOROTOF
    endif
    ;Werkstueckbezugspunkte
    if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H10'
    $P_WPFRAME = ctrans()
    endif
    ;Zyklen
    if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H20'
    $P_CYCFRAME = ctrans()
    endif
    ;Transformationen
    if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H40'
    $P_TRAFRAME = ctrans()
    endif
    ; Bitmaske fuer globale Basisframes
    $P_NCBFRMASK = 0
    ;Bitmaske für kanalspezifische Basisframes
    $P_CHBFRMASK = 0
    ;programmierbare Frame
    $P_PFRAME = ctrans()

```

M17

 **VORSICHT**

Programmänderungen

Mit der Veränderung des Inhalts des RESU-spezifischen Unterprogramms "CC_RESU_INI.SPF" übernimmt der Anwender (Maschinenhersteller) die Verantwortung für den korrekten Ablauf der Technologie-Funktion.

Hinweis

CC_RESU_INI.SPF darf verändert werden.

CC_RESU_INI.SPF darf keine RESU-Teileprogrammbefehle CC__PREPRE (x) enthalten.

9.5.4 END-Programm (CC_RESU_END.SPF)

Funktion

Das RESU-spezifische Unterprogramm "CC_RESU_END.SPF" hat die Aufgabe, das Rückwärtsfahren anzuhalten, wenn das Ende der RESU-fähigen Kontur erreicht wurde. Bei geeigneter Parametrierung von RESU wird dieser Fall in der Regel nicht eintreten.

Programmstruktur

CC_RESU_END.SPF hat folgenden voreingestellten Inhalt:

```
PROC CC_RESU_END
      M0
M17
```

 **VORSICHT**

Programmänderungen

Mit der Veränderung des Inhalts des RESU-spezifischen Unterprogramms "CC_RESU_END.SPF" übernimmt der Anwender (Maschinenhersteller) die Verantwortung für den korrekten Ablauf der Technologie-Funktion.

Hinweis

CC_RESU_END.SPF darf verändert werden.

CC_RESU_END.SPF darf keine RESU-Teileprogrammbefehle CC__PREPRE (x) enthalten.

9.5.5 Wiederaufsetz-ASUP (CC_RESU_BS_ASUP.SPF)

Funktion

Mittels des RESU-spezifischen ASUP "CC_RESU_BS_ASUP.SPF" wird die NC veranlasst, beim Wiederaufsetzen an den aktuellen Bahnpunkt anzufahren:

- Wiederanfahren an den nächstliegenden Bahnpunkt: RMN
- Anfahren auf einer Geraden mit allen Achsen: REPOSA

Programmstruktur

CC_RESU_BS_ASUP.SPF hat folgenden voreingestellten Inhalt:

```
PROC CC_RESU_BS_ASUP SAVE
    RMN
    REPOSA
M17
```



VORSICHT

Programmänderungen

Mit der Veränderung des Inhalts des RESU-spezifischen Unterprogramms "CC_RESU_BS_ASUP.SPF" übernimmt der Anwender (Maschinenhersteller) die Verantwortung für den korrekten Ablauf der Technologie-Funktion.

Hinweis

CC_RESU_BS_ASUP.SPF darf verändert werden.

Anwenderspezifische Anpassungen sind vor dem Teileprogrammsatz RMN einzufügen.

9.5.6 RESU-ASUP (CC_RESU_ASUP.SPF)

Funktion

Das RESU-spezifische ASUP "CC_RESU_ASUP.SPF" wird funktionsintern benötigt. Das ASUP wird ausgelöst, wenn im NC-Stop-Zustand das folgende RESU-Nahtstellensignal umgeschaltet wird:

DB21, ... DBX0.1 (Vorwärts / Rückwärts)

Programmstruktur

CC_RESU_ASUP.SPF hat folgenden Inhalt:

PROC CC_RESU_ASUP	
	; siemens system asup - do not change G4 F0.001 M0 REPOSA
M17	

Hinweis

CC_RESU_ASUP.SPF darf nicht verändert werden.

9.6 Wiederaufsetzen

9.6.1 Allgemeine Informationen

Wiederaufsetzen bezeichnet den gesamten Vorgang vom Auslösen des Wiederaufsetzens über das Nahtstellensignal DB21, ... DBX0.2 = 1 (Wiederaufsetzen starten) bis zum Fortsetzen der Teileprogrammbearbeitung auf der programmierten Kontur.

Voraussetzung

Als Voraussetzung für das Wiederaufsetzen muss der Retrace Mode, ausgelöst durch die Anforderung zum Rückwärtsfahren, im Kanal aktiv sein:

DB21, ... DBX32.1 = 1 (Retrace Mode aktiv)

(Siehe Kapitel "Funktionsablauf (Prinzip) (Seite 593)")

Teilfunktionen

Die beiden wesentlichen Teilfunktionen des Wiederaufsetzens sind die Standard-NC-Funktionen:

- Satzsuchlauf mit Berechnung an der Kontur
- Repositionieren an die Kontur auf kürzestem Weg (REPOS RMN)

9.6.2 Satzsuchlauf mit Berechnung an der Kontur

Funktion

Der im Rahmen des Wiederaufsetzens implizit von RESU ausgelöste Satzsuchlauf mit Berechnung an der Kontur hat folgende Aufgaben:

- Programmzeiger auf den Teileprogrammsatz stellen, auf den mittels Rückwärts- / Vorwärtsfahren zurückpositioniert wurde.
- Berechnung der Achspositionen aufgrund der programmierten Verfahrssätze vom Teileprogrammanfang bis zum Zielsatz.
- Aufsammeln der vom Teileprogrammanfang bis zum Zielsatz programmierten Anweisungen, die im Aktionssatz ausgeführt werden, wie z. B.:
 - Hilfsfunktionen
 - Werkzeugwechsel
 - Spindelfunktionen
 - Vorschubprogrammierung

Alle Teileprogrammanweisungen, die nicht im Aktionssatz ausgeführt werden, aber zum Wiederaufsetzen im Teileprogramm benötigt werden, müssen manuell in das RESU-spezifische Wiederaufsetz-ASUP "CC_RESU_BS_ASUP.SPF" eingetragen werden, z. B.:

- Synchronaktionen
- M-Funktionen

Weitere Informationen siehe Funktionshandbuch Basisfunktionen.

Siehe auch

K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 29)

9.6.3 Repositionieren

Funktion

Nach dem Ende des letzten Aktionssatzes (letzter Verfahrssatz vor dem Repositionieren) erfolgt mit NC-Start die Ausgabe des Anfahrsatzes zum Repositionieren aller im Teileprogramm bis zum Zielsatz programmierten Kanalachsen.

Geometrieachsen

Im Anfahrtsatz verfahren die Geometrieachsen der RESU-Arbeitsebene (z. B. die 1. und 2. Geometrieachse des Kanals) auf kürzestem Weg zum Wiederaufsetzpunkt an die Kontur.

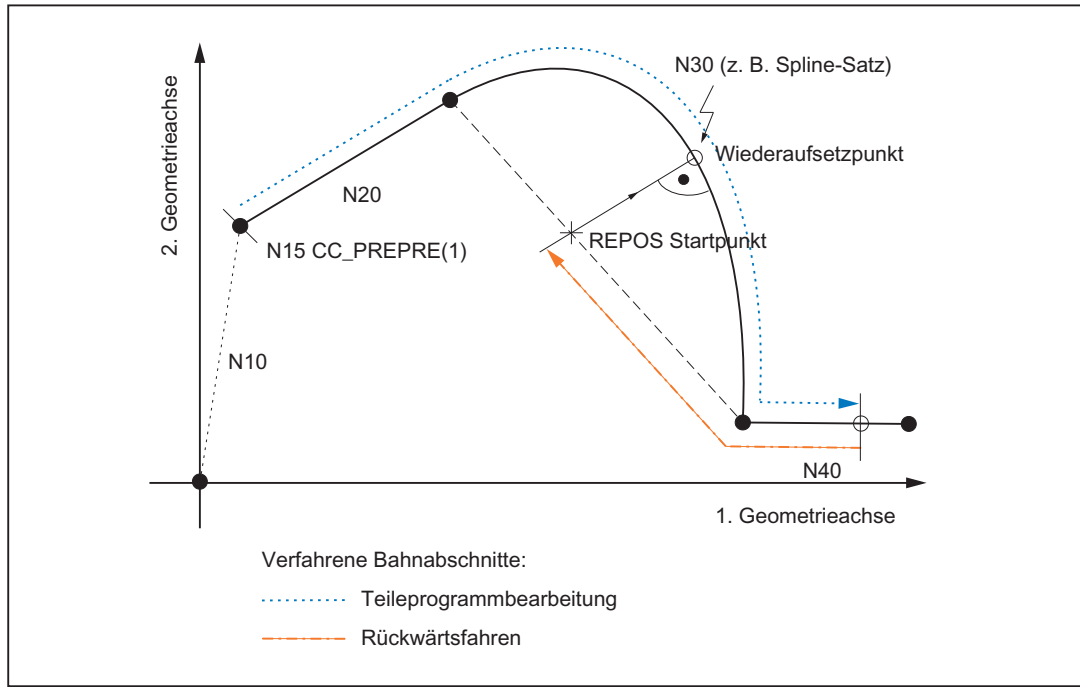


Bild 9-7 RESU-fähige Konturbereiche und REPOS

Kanalachsen

Alle anderen im Teileprogramm programmierten Kanalachsen verfahren auf ihre jeweilige im Satzsuchlauf berechnete Position.

9.6.4 Zeitliche Bedingungen bezüglich NC-Start

Im Rahmen des Wiederaufsetzens ist vom Maschinenbediener zweimal NC-Start auszulösen (siehe Kapitel "Funktionsablauf (Prinzip) (Seite 593)").

Dabei sind folgende Bedingungen zu beachten:

- beim NC-Start zum Ausgeben der Aktionssätze:
 - Der Satzsuchlauf muss abgeschlossen sein:
DB21, ... DBX33.4 = 0 (Satzsuchlauf aktiv)
- beim NC-Start zum Ausgeben des Anfahrtsatzes:
 - Das RESU-ASUP "CC_RESU_BS_ASUP" muss beendet sein:
DB21, ... DBX318.0 = 1 (ASUP angehalten)

9.6.5 Satzsuchlauf ab letztem Hauptsatz

Der im Rahmen des Wiederaufsetzens durchgeführte Satzsuchlauf mit Berechnung an der Kontur kann selbst bei Verwendung der leistungsfähigsten NCU bei sehr großen Teileprogrammen zu Rechenzeiten von mehreren Minuten bis zum Erreichen des Zielsatzes führen.

Eine signifikante Verkürzung dieser Wartezeit ist möglich durch die Verwendung der Funktion "Satzsuchlauf ab dem letzten Hauptsatz".

Funktionalität

Beim Wiederaufsetzen mit Satzsuchlauf ab dem letzten Hauptsatz erfolgt das Auffinden des Zielsatzes in 2 Schritten:

1. Satzsuchlauf ohne Berechnung vom Anfang des Bearbeitungsprogramms bis zum letzten Hauptsatz vor dem Zielsatz. Während dieses Satzsuchlaufs werden keine Unterprogramme berücksichtigt. Diese Suche erfolgt somit ausschließlich im Hauptprogramm.
2. Satzsuchlauf mit Berechnung an der Kontur ab dem Hauptsatz bis zum Zielsatz. Dieser Satzsuchlauf berücksichtigt auch Unterprogramme.

Voraussetzung

Voraussetzung dafür, dass zum Wiederaufsetzen ein Satzsuchlauf ab dem letzten Hauptsatz verwendet werden kann, ist die Programmierung mindestens eines Hauptsatzes nach dem RESU-Start `CC_PREPRE (1)`.

Hauptsatz

In einem Hauptsatz müssen alle Anweisungen programmiert sein, die zum Abarbeiten des nachfolgenden Teileprogrammabschnitts erforderlich sind.

Hauptsätze sind durch eine Hauptsatznummer zu kennzeichnen, die aus dem Zeichen ":" und einer positiven ganzen Zahl (Satznummer) besteht.

Weitere Informationen:

Programmierhandbuch Grundlagen; Grundlagen der NC-Programmierung

Aktivierung

Das Aktivieren des Satzsuchlaufs ab dem letzten Hauptsatz erfolgt über das RESU-spezifische Maschinendatum:

MD62575 \$MC_RESU_SPECIAL_FEATURE_MASK_2, Bit 0 (Zusätzlich RESU-Eigenschaften)

Bit	Wert	Bedeutung
0	0	Das Wiederaufsetzen erfolgt mittels Satzsuchlauf mit Berechnung an der Kontur.
	1	Das Wiederaufsetzen erfolgt mittels Satzsuchlauf ab dem letzten Hauptsatz.

Randbedingungen

Damit nach einem Wiederaufsetzen mit Satzsuchlauf ab dem letzten Hauptsatz ein erneutes Wiederaufsetzen erfolgen kann, muss der RESU-Start `CC_PREPRE(1)` im Wiederaufsetz-ASUP "CC_RESU_BS_ASUP.SPF" programmiert sein.

Programmierbeispiel:

```
PROC CC_RESU_BS_ASUP SAVE
    ; (zum Wiederaufsetzen benötigte
    ; Synchronaktionen, M-Funktionen, etc.)
    CC_PREPRE(1)
    RMN
    REPOSA
M17
```

9.7 Funktionsspezifische Anzeigedaten

9.7.1 Kanalspezifische GUD-Variable

Als Anzeigedatum für die Größe des Satzsuchlaufpuffers stellt RESU folgende kanalspezifische GUD-Variable zur Verfügung:

GUD-Variable	Bedeutung	Einheit	Zugriff
CLC_RESU_LENGTH_BS_BUFFER	Größe des Satzsuchlaufpuffers	Byte	nur lesen

Nach erfolgter Inbetriebnahme der Technologie-Funktion wird die GUD-Variable nicht automatisch auf der Bedienoberfläche angezeigt.

Anlegen und Anzeigen der GUD-Variable

Zum Anlegen und Anzeigen der GUD-Variablen sind auf der Bedienoberfläche folgende Handlungsschritte durchzuführen:

1. Kennwort setzen:
Es ist das Kennwort der Schutzstufe 1 (Maschinenhersteller) einzugeben.
2. Anzeige der "Definitionen" aktivieren.
3. Falls noch keine Datei "SGUD.DEF" vorhanden ist, wird die Datei neu angelegt:
Name: SGUD
Typ: GlobaleDaten / System
Eingaben mit **OK** bestätigen.
Die Datei wird daraufhin im Editor geöffnet.
4. GUD-Variablen-Definitionen editieren:
DEF CHAN REAL CLC_RESU_LENGTH_BS_BUFFER
M30

5. Datei speichern und Editor schließen.

6. Datei "SGUD.DEF" aktivieren.

Die GUD-Variable wird jetzt auf der Bedienoberfläche angezeigt.

Hinweis

Die neu angelegte und bereits angezeigte GUD-Variable wird von RESU erst nach einem NC-POWER ON-Reset erkannt und mit dem aktuellen Wert versorgt. Nach dem Anlegen muss daher ein NC-POWER ON-Reset ausgelöst werden.

Weitere Informationen

Das genaue Vorgehen beim Anlegen und Anzeigen von GUD-Variablen ist abhängig von der Software-Version der vorhandenen Bedienoberfläche und beschrieben im:

- Bedienhandbuch Universal/Drehen/Fräsen/Schleifen

9.8 Funktionsspezifische Alarmtexte

Zum Vorgehen beim Anlegen von funktionsspezifischen Alarmtexten siehe Funktionshandbuch Technologien, Installation und Aktivierung ladbarer Compile- Zyklen.

9.9 Randbedingungen**9.9.1 Funktionsspezifische Randbedingungen****9.9.1.1 Wiederaufsetzen innerhalb von Unterprogrammen****Unterprogrammaufruf außerhalb oder innerhalb einer Programmschleife**

Ein eindeutiges Wiederaufsetzen innerhalb von Unterprogrammen ist davon abhängig, ob der Unterprogrammaufruf außerhalb oder innerhalb einer Programmschleife erfolgt:

- Außerhalb
Wird ein Unterprogramm außerhalb einer Programmschleife aufgerufen, ist ein eindeutiges Wiederaufsetzen möglich.
- Innerhalb
Wird ein Unterprogramm innerhalb einer Programmschleife aufgerufen, ist unter Umständen kein eindeutiges Wiederaufsetzen möglich (siehe Kapitel "Wiederaufsetzen innerhalb von Programmschleifen (Seite 616)").

Durchlaufzahl P

Unterprogramm-Wiederholungen durch Verwendung der Durchlaufzahl P werden beim Wiederaufsetzen berücksichtigt. D. h. das Wiederaufsetzen im Teileprogramm erfolgt mit dem korrekten Bezug von Teileprogrammsatz und Durchlaufzahl P zum Wiederaufsetzpunkt der Kontur.

9.9.1.2 Wiederaufsetzen innerhalb von Programmschleifen

Innerhalb der NC-Hochsprache sind Programmschleifen programmierbar mittels:

- LOOP ENDLOOP
- FOR ENDFOR
- WHILE ENDWHILE
- REPEAT UNTIL
- CASE/IF-ELSE-ENDIF in Zusammenhang mit GOTOB

WARNUNG

Kollisionsgefahr

Wenn der Wiederaufsetzpunkt an der programmierten Kontur das Ergebnis eines Schleifendurchlaufs ungleich dem ersten Schleifendurchlauf ist, können sich im weiteren Verlauf der Bearbeitung erhebliche Konturabweichungen ergeben.

Bei einem Wiederaufsetzen innerhalb von Programmschleifen erfolgt das Wiederaufsetzen immer im ersten Schleifendurchlauf.

9.9.1.3 Wiederaufsetzen an Vollkreisen

Bei Vollkreisen fallen Satzanfangs- und -endpunkt in einem Konturpunkt zusammen. Da in diesem Fall keine eindeutige Unterscheidung möglich ist, wird beim Wiederaufsetzen an einem derartigen Konturpunkt immer vom Satzanfangspunkt ausgegangen. Der erste Teileprogrammsatz nach Wiederaufsetzen ist dann der Kreissatz.

Um zu vermeiden, dass nach dem Wiederaufsetzen der Kreissatz gefahren wird, ist als Wiederaufsetzpunkt ein Konturpunkt kurz vor dem Satzendpunkt des Kreissatzes zu wählen.

9.9.1.4 Automatisch generierte Konturelemente

Die automatische Generierung von nicht-linearen / nicht-kreisförmigen Konturelementen durch die NC erfolgt z. B. bei Programmierung folgender NC-Funktionen im Teileprogramm:

- RND
- G641/G642
- Werkzeugradiuskorrektur

Für das Rückwärts- / Vorwärtsfahren im Rahmen von RESU werden diese Konturelemente durch Geraden zwischen Satzanfangs- und Satzendpunkt ersetzt.

9.9.2 Randbedingungen bezüglich Standardfunktionen

9.9.2.1 Achstausch

Solange RESU aktiv ist, dürfen die beiden Geometrieachsen der RESU-Arbeitsebene (z. B. 1. und 2. Geometrieachse des Kanals) nicht per Achstausch (`RELEASE (x) /GET (x)`) an einen anderen Kanal übergeben werden.

Die RESU-Aktivität:

- beginnt:
 - mit dem Teileprogrammbefehl `CC_PREPRE (1)`
- endet mit:
 - dem Programmende
oder
 - dem Teileprogrammbefehl `CC_PREPRE (-1)`

9.9.2.2 Verfahrbewegungen von Kanalachsen

Andere Kanalachsen außer den beiden Geometrieachsen der RESU-Arbeitsebene werden durch RESU nicht betrachtet.

Sind zum Wiederaufsetzen bzw. Rückwärtsfahren Verfahrbewegungen in anderen Kanalachsen erforderlich, so können diese entweder vom Maschinenbediener von Hand vorgenommen werden, oder im RESU-spezifischen Unterprogramm "CC_RESU_INI.SPF" als Verfahratz programmiert werden.

 WARNUNG
--

Kollisionsgefahr bei Verfahrbewegung

Während des gesamten Wiederaufsetzvorgangs im Rahmen der Technologie-Funktion RESU ist vom Maschinenbediener die Kollisionsfreiheit der damit verbundenen Verfahrbewegungen sicherzustellen.
--

9.9.2.3 Satznummern

Die folgenden RESU-spezifischen Unterprogramme und deren Unterprogramme dürfen keine Satznummern enthalten:

- `CC_RESU_INI.SPF`
- `CC_RESU_END.SPF`

Im Fehlerfall wird folgender Alarm angezeigt:

Alarm 75604 "Rückwärtsfahren nicht möglich, Fehler-Nr. *Nummer*"

9.9.2.4 Satzsuchlauf

Satzsuchlauf mit Berechnung

Im Rahmen der Standardfunktion "Satzsuchlauf mit Berechnung (an der Kontur / am Satzende)" gelten bezüglich RESU folgende Randbedingungen:

- Im Zielsatz ist der letzte während des Satzsuchlaufs durchlaufene RESU-Teileprogrammbefehl `CC_PREPRE (x)` wirksam.
- Der RESU-fähige Konturbereich beginnt mit dem `REPOS`-Anfahrtsatz.

Satzsuchlauf ohne Berechnung

Bei der Funktion "Satzsuchlauf ohne Berechnung" werden die RESU-Teileprogrammbefehle `CC_PREPRE (x)` **nicht** wirksam.

9.9.2.5 Transformationen

RESU kann mit Einschränkungen auch bei aktiver kinematischer Transformation (z. B. 5-Achstransformation) verwendet werden, da die Verfahrbewegungen der beiden Geometrieachsen der RESU-Arbeitsebene im Basis-Koordinatensystem (BKS) und somit vor der Transformation protokolliert werden (siehe auch Funktionshandbuch *Technologien*, Kapitel *Transformationspaket - Handling*).

Transformationswechsel

Während RESU aktiv ist, darf kein Wechsel der Transformation bzw. kein Ein- / Ausschalten der Transformation erfolgen.

Die RESU-Aktivität:

- beginnt:
 - mit dem Teileprogrammbefehl `CC_PREPRE (1)`
- endet mit:
 - dem Programmende
oder
 - dem Teileprogrammbefehl `CC_PREPRE (-1)`

Weitere Informationen

Funktionshandbuch Transformationen; Kinematische Transformation

9.9.2.6 Kompensationen

RESU kann im Zusammenhang mit Kompensationen verwendet werden, da die Verfahrbewegungen der beiden Geometrieachsen der RESU-Arbeitsebene im Basis-Koordinatensystem (BKS) und somit vor der Kompensation protokolliert werden.

Eine vollständige Beschreibung der Kompensationen findet sich in:

Weitere Informationen:

Funktionshandbuch Überwachen und Kompensieren; Kompensationen

9.9.2.7 Frames

RESU kann im Zusammenhang mit Frames verwendet werden.

Da die Verfahrbewegungen der beiden Geometrieachsen der RESU-Arbeitsebene aber im Basis-Koordinatensystem (BKS) und somit nach Einrechnung der Frames protokolliert werden, müssen während des Wiederaufsetzens (Rückwärts- / Vorwärtsfahren) die Frame-Korrekturen ausgeschaltet sein.

Das Ausschalten der Frame-Korrekturen während des Wiederaufsetzens erfolgt durch die standardmäßigen Voreinstellungen des RESU-spezifischen Unterprogramms "CC_RESU_INI.SPF" (siehe Kapitel "INI-Programm (CC_RESU_INI.SPF) (Seite 606)").

Weitere Informationen

Funktionshandbuch Basisfunktionen; Achsen, Koordinatensysteme, Frames

Siehe auch

K2: Achsen, Koordinatensysteme, Frames (Seite 261)

9.9.2.8 Werkzeugkorrekturen

RESU kann im Zusammenhang mit Werkzeugkorrekturen verwendet werden.

Da die Verfahrbewegungen der beiden Geometrieachsen der RESU-Arbeitsebene aber im Basis-Koordinatensystem (BKS) und somit nach Einrechnung der Werkzeugkorrekturen protokolliert werden, müssen während des Wiederaufsetzens (Rückwärts- / Vorwärtsfahren) die Werkzeugkorrekturen ausgeschaltet sein.

Das Ausschalten der Werkzeugkorrekturen während des Wiederaufsetzens erfolgt durch die standardmäßigen Voreinstellungen des RESU-spezifischen Unterprogramms "CC_RESU_INI.SPF" (siehe Kapitel "INI-Programm (CC_RESU_INI.SPF) (Seite 606)").

In spezifischen Fällen der Werkzeugradiuskorrektur, z. B. Korrekturen an Außenecken $G450\ DISC=x$, kann es zu Konturabweichungen zwischen der während des Wiederaufsetzens gefahrenen Kontur und der im Bearbeitungsprogramm programmierten Kontur kommen.

Konturabweichungen ergeben sich immer dann, wenn die Werkzeugradiuskorrektur Konturelemente erzeugt, die nicht linear oder kreisförmig sind. Z. B. erzeugt $G450\ DISC=x$ mit $x > 0$ parabel- oder hyperbelförmige Konturelemente.

Weitere Informationen zum Thema Werkzeugkorrekturen siehe Funktionshandbuch Werkzeuge.

9.10 Datenlisten

9.10.1 Maschinendaten

9.10.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
11602	ASUP_START_MASK	Ignoriere Stoppgründe, wenn ein Asup läuft.
11604	ASUP_START_PRIO_LEVEL	Legt fest, ab welcher Asup-Priorität MD11602 wirksam ist.
18351	MM_DRAM_FILE_MEM_SIZE	Größe des Teileprogrammspeicher im DRAM (in kByte)

9.10.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20050	AXCONF_GEOAX_ASSIGN_TAB	Zuordnung Geometrie - Kanalachse
24120	TRAFO_GEOAX_ASSIGN_TAB_1	Zuordnung Geometrie - Kanalachse für Transformati- on 1
28090	MM_NUM_CC_BLOCK_ELEMENTS	Anzahl Blockelemente für Compile-Zyklen (CC)
28100	MM_NUM_CC_BLOCK_USER_MEM	Größe Satzspeicher für CC in kByte
28105	MM_NUM_CC_HEAP_MEM	Heap-Speicher in kByte für CC-Applikationen (DRAM)
62571	RESU_RING_BUFFER_SIZE	Größe des Ringpuffers (RESU-interner Satzpuffer)
62572	RESU_SHARE_OF_CC_HEAP_MEM	RESU-Anteil am gesamten CC-Heap-Speicher
62573	RESU_INFO_SA_VAR_INDEX	Indizes der benutzten Synchronaktionsvariablen
62574	RESU_SPECIAL_FEATURE_MASK	Zusätzlich RESU-Eigenschaften
62575	RESU_SPECIAL_FEATURE_MASK_2	Zusätzlich RESU-Eigenschaften
62580	RESU_WORKING_PLANE	Anwahl der RESU-Arbeitsebene

N2: Not-Halt

10.1 Kurzbeschreibung

Funktion

Die Steuerung unterstützt den Maschinenhersteller bei der Realisierung der Not-Halt-Funktion durch folgende Funktionen:

- An allen SINUMERIK-Maschinensteuertafeln ist ein Not-Halt-Taster für den Maschinenbediener leicht erreichbar angebracht. Der Funktionalität des Not-Halt-Tasters umfasst die Zwangsöffnung der elektrischen Schaltkontakte und eine mechanisch selbsttätige Verrastung/Verriegelung.
- Die Not-Halt-Anforderung an die NC erfolgt über die PLC über die NC/PLC-Nahtstelle.
- Bei Not-Halt muss ein Stillsetzen nach Stopp-Kategorie 0 oder 1 (EN 60204) erfolgen.
- Alle durch die PLC gesteuerten Maschinenfunktionen können bei Not-Halt einen, durch den Maschinenhersteller einstellbaren, sicheren Zustand einnehmen.
- Ein Entriegeln des Not-Halt-Tasters bewirkt kein Aufheben des Not-Halt-Zustandes bzw. löst keinen Wiederanlauf aus.
- Nach dem Aufheben des Not-Halt-Zustandes ist kein Referenzieren der Maschinenachsen bzw. Synchronisieren der Spindeln nötig. Die Istpositionen der Maschinenachsen werden während des Not-Halt-Ablaufs kontinuierlich nachgeführt.

10.2 Normen

Relevante Normen

Im Rahmen der Not-Halt-Funktion sind folgende Normen unbedingt zu beachten:

- EN ISO 12000-1
- EN ISO 12000-2
- EN 418
- EN 60204

Not-Halt

Entsprechend EN 418 ist Not-Halt eine Funktion, die:

- aufkommende oder bestehende Gefahren für Personen, Schäden an der Maschine oder dem Arbeitsgut abwenden oder vermindern soll.
- durch eine einzige Handlung einer Person ausgelöst wird, wenn die normale Haltefunktion dafür nicht angemessen ist.

Gefahren

Gefahren im Sinne der EN 418 sind solche, die herrühren können von:

- funktionalen Unregelmäßigkeiten (Fehlfunktionen der Maschine, nicht hinnehmbare Eigenschaften des bearbeiteten Materials, menschliche Fehler, ...).
- normalem Betrieb.

Norm EN ISO 12000-2

Gemäß einer grundlegenden Sicherheitsanforderung der EG-Richtlinie Maschinen hinsichtlich Not-Halt müssen Maschinen mit einer Not-Halt-Einrichtung versehen sein.

Ausnahmen

Keine Not-Halt-Einrichtung wird bei Maschinen benötigt:

- an denen eine Not-Halt-Einrichtung das Risiko nicht verringern würde, weil dadurch entweder die Stillsetzzeit nicht verringert würde oder weil die dafür zu ergreifenden Maßnahmen nicht geeignet wären, das Risiko zu beeinflussen.
- die von Hand getragen und geführt werden.

Hinweis

Der Maschinenhersteller wird auf die Einhaltung der nationalen und internationalen Normen ausdrücklich hingewiesen. Die SINUMERIK-Steuerungen unterstützen den Maschinenhersteller bei der Realisierung der Not-Halt-Funktion entsprechend den Festlegungen der folgenden Funktionsbeschreibung. Die Verantwortung für die Not-Halt-Funktion (Auslösung, Ablauf, Quittierung) liegt aber ausschließlich beim Maschinenhersteller selbst.

10.3 Not-Halt-Stellteile

Not-Halt-Stellteile

Nach EN 418 müssen Not-Halt-Stellteile so konstruiert sein, dass sie mechanisch selbsttätig verrasten und für die Bedienperson und andere im Notfall leicht zu betätigen sind.

Folgende Typen von Stellteilen können unter anderem eingesetzt werden:

- Pilztaster (drucktastenbetätigter Schalter)
- Drähte/Drahtseile, Leinen, Stangen
- Griffe
- In besonderen Fällen: Fußschalter ohne Schutzhaube

Not-Halt-Taster und Steuerung

Die Betätigung des Not-Halt-Tasters oder ein direkt daraus abgeleitetes Signal muss als PLC-Eingang zur Steuerung (PLC) geführt werden. Im PLC-Anwenderprogramm muss dieser PLC-Eingang weitergeleitet werden an die NC auf das Nahtstellensignal:

DB10 DBX56.1 (Not-Halt)

Das Rückstellen des Not-Halt-Tasters oder ein direkt daraus abgeleitetes Signal muss als PLC-Eingang zur Steuerung (PLC) geführt werden. Im PLC-Anwenderprogramm muss dieser PLC-Eingang weitergeleitet werden an die NC auf das Nahtstellensignal:

DB10 DBX56.2 (Not-Halt quittieren)

Anschlussbedingungen

Zum Anschluss des Not-Halt-Tasters siehe:

Weitere Informationen:

Gerätehandbuch Bedienkomponenten

10.4 Not-Halt-Ablauf

Nach Betätigung des Not-Halt-Stellteils muss die Not-Halt-Einrichtung in einer Weise arbeiten, dass die Gefahr automatisch auf die bestmögliche Weise abgewendet oder verringert wird.

"Auf bestmögliche Weise" bedeutet, dass die günstigste Verzögerungsrate gewählt und die richtige Stopp-Kategorie (definiert in EN 60204) entsprechend einer Risikoabschätzung festgelegt werden kann.

Not-Halt-Ablauf in der NC

Der nach EN 418 vorbestimmte Ablauf interner Funktionen zum Not-Halt-Zustand sieht in der Steuerung wie folgt aus:

1. Die Teileprogrammbearbeitung wird unterbrochen.
Alle Maschinenachsen werden abgebremst in der jeweiligen achsspezifisch parametrisierten Zeit:
MD36610 \$MA_AX_EMERGENCY_STOP_TIME (Zeit der Bremsrampe bei Fehlern)
Die maximale Bremsrampe, die dabei erzielt werden kann, wird definiert durch den maximalen Bremsstrom des jeweiligen Antriebs. Der maximale Bremsstrom wird erreicht durch Vorgabe von Sollwert = 0 (Schnellbremsung).
2. Zurücksetzung des Nahtstellensignals:
DB11 DBX6.3 (BAG-betriebsbereit)
3. Setzen des Nahtstellensignals:
DB10 DBX106.1 (Not-Halt aktiv)
4. Alarm 3000 "Not-Halt" wird angezeigt.
5. Nach Ablauf einer parametrierbaren Verzögerungszeit werden die Reglerfreigaben der Maschinenachsen zurückgesetzt.
Die Einstellung der Verzögerungszeit erfolgt im Maschinendatum:
MD36620 \$MA_SERVO_DISABLE_DELAY_TIME (Abschaltverzögerung Reglerfreigabe)
Folgende Einstellregel ist zu beachten: MD36620 ≥ MD36610
6. Alle Maschinenachsen werden steuerungsintern in den Nachführbetrieb geschaltet.
Die Maschinenachsen sind dabei nicht mehr in Lageregelung.

Not-Halt-Ablauf an der Maschine

Der Not-Halt-Ablauf an der Maschine wird ausschließlich vom Maschinenhersteller bestimmt.

Dabei ist in Verbindung mit dem Ablauf in der NC Folgendes zu beachten:

- Der Ablauf in der NC wird gestartet mit dem Nahtstellensignal:
DB10 DBX56.1 (Not-Halt)
Nachdem die Maschinenachsen im Stillstand sind, muss nach EN 418 die Energiezufuhr unterbrochen werden.

Hinweis

Das Unterbrechen der Energiezufuhr liegt in der Verantwortung des Maschinenherstellers.

- Die digitalen und analogen Ausgänge der PLC-Peripherie werden vom Not-Halt-Ablauf in der NC nicht beeinflusst.
Sollen einzelne Ausgänge bei Not-Halt einen bestimmten Zustand oder Spannungspegel einnehmen, ist dies vom Maschinenhersteller im PLC-Anwenderprogramm vorzunehmen.
- Die schnellen digitalen Ausgänge der NCK-Peripherie werden vom Not-Halt-Ablauf in der NC nicht beeinflusst.
Sollen einzelne Ausgänge bei Not-Halt einen bestimmten Zustand einnehmen, muss der Maschinenhersteller im PLC-Anwenderprogramm den gewünschten Zustand an die NC übergeben über die Nahtstellensignale:
DB10 DBB4-7

Hinweis

Soll bei Not-Halt der Ablauf in der NC nicht wie oben beschrieben ablaufen, darf bis zum Erreichen eines durch den Maschinenhersteller im PLC-Anwenderprogramm festgelegten Not-Halt-Zustandes das Nahtstellensignal DB10 DBX56.1 (Not-Halt) nicht gesetzt werden.

Solange das Nahtstellensignal nicht gesetzt ist und kein anderer Alarm ansteht, sind in der NC alle Nahtstellensignale wirksam. Dadurch kann jeder herstellerspezifische Not-Halt-Zustand (auch Achs-, Spindel- und Kanal-spezifisch) eingenommen werden.

10.5 Not-Halt-Quittierung

Nach EN 418 darf das Rückstellen des Not-Halt-Stellteils nur als Ergebnis einer von Hand ausgeführten Handlung am Not-Halt-Stellteil möglich sein.

Das Rückstellen des Not-Halt-Stellteils allein darf keinen Wiederanlauf-Befehl auslösen.

Der Wiederanlauf der Maschine darf nicht möglich sein, bis alle betätigten Not-Halt-Stellteile von Hand einzeln und bewusst rückgestellt worden sind.

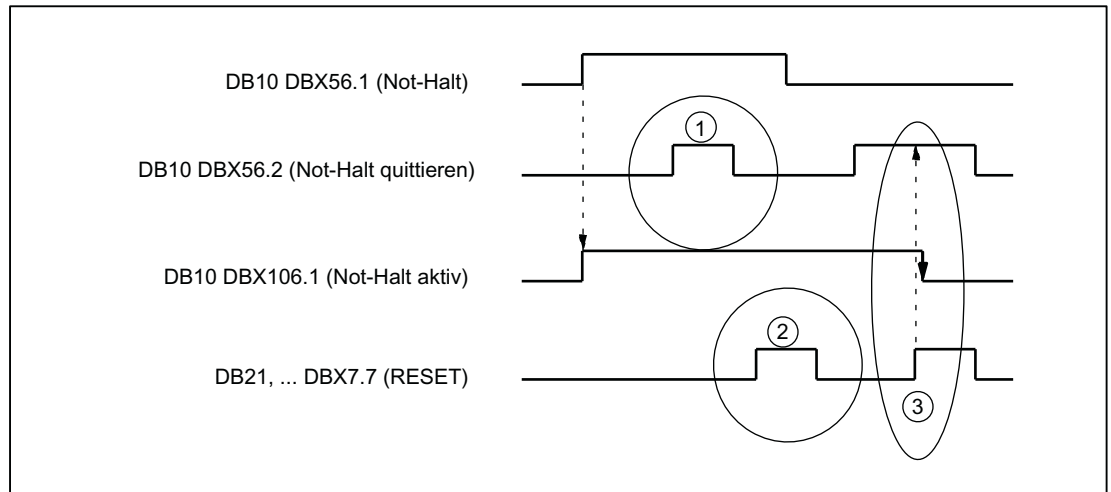
Not-Halt-Quittierung

Der Not-Halt-Zustand wird nur dann wieder zurückgesetzt, wenn zuerst das Nahtstellensignal DB10 DBX56.2 (Not-Halt quittieren) und anschließend das Nahtstellensignal DB11 DBX0.7 (BAG-Reset) gesetzt wird.

Dabei ist zu beachten, dass das Nahtstellensignal DB10 DBX56.2 (Not-Halt quittieren) und das Nahtstellensignal DB21, ... DBX7.7 (Reset) gemeinsam mindestens so lange gesetzt sind, bis das Nahtstellensignal DB10 DBX106.1 (Not-Halt aktiv) zurückgesetzt wurde.

Hinweis

Allein mit dem Nahtstellensignal DB21, ... DBX7.7 (Reset) kann der Not-Halt-Zustand nicht zurückgesetzt werden.



- (1) DB10 DBX56.2 (Not-Halt quittieren) ist wirkungslos
- (2) DB21, ... DBX7.7 (Reset) ist wirkungslos
- (3) DB10 DBX56.2 und DB21, ... DBX7.7 setzen DB10 DBX106.1 (Not-Halt aktiv) zurück

Bild 10-1 Not-Halt-Zustand rücksetzen

Auswirkungen

Durch Rücksetzen des Not-Halt-Zustandes werden:

- steuerungsintern für alle Maschinenachsen:
 - die Reglerfreigaben gesetzt.
 - der Nachführbetrieb aufgehoben.
 - die Lageregelung aktiviert.
- die folgenden Nahtstellensignale gesetzt:
 - DB31, ... DBX60.5 (Lageregelung aktiv)
 - DB11 DBX6.3 (BAG betriebsbereit)
- das folgende Nahtstellensignal zurückgesetzt:
 - DB10 DBX106.1 (Not-Halt aktiv)
- der Alarm 3000 "Not-Halt" gelöscht.
- die Teileprogrammabarbeitung in allen Kanälen der NC abgebrochen.

PLC- und NC-Peripherie

Die PLC- und NC-Peripherie müssen durch das PLC-Anwenderprogramm wieder in den Zustand zum Betrieb der Maschine versetzt werden.

POWER OFF/ON (Netz aus/ein)

Der Not-Halt-Zustand kann auch durch Aus- und Einschalten der Steuerung (POWER OFF/ON) zurückgesetzt werden.

Voraussetzung:

Beim Hochlauf der Steuerung darf das Nahtstellensignal DB10 DBX56.1 (Not-Halt) nicht gesetzt sein.

10.6 Datenlisten

10.6.1 Maschinendaten

10.6.1.1 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
36610	AX_EMERGENCY_STOP_TIME	Zeitdauer der Bremsrampe bei Fehlerzuständen
36620	SERVO_DISABLE_DELAY_TIME	Abschaltverzögerung Reglerfreigabe

A2: Diverse NC/PLC-Nahtstellensignale und Funktionen

11

11.1 Kurzbeschreibung

Inhalt

Die Nahtstelle PLC/NC wird einerseits durch eine Datenschnittstelle und andererseits durch eine Funktionsschnittstelle gebildet. In der Datenschnittstelle sind Status- und Steuersignale, Hilfs- und G-Befehl enthalten, während über die Funktionsschnittstelle Aufträge von der PLC an den NC übergeben werden.

In der vorliegenden Beschreibung wird die Funktionalität von Nahtstellensignalen beschrieben, die von allgemeiner Bedeutung sind und die in den funktionsspezifischen Beschreibungen nicht beschrieben sind:

- Asynchrone Ereignisse
- Zustandssignale
- PLC-Variable (lesen und schreiben)

11.2 Funktionen

11.2.1 Verhalten der Bildaktualisierung bei Überlastung

Es gibt Teileprogramme, bei denen der Hauptlauf (HL) warten muss, bis der Vorlauf (VL) neue Sätze zur Verfügung stellt.

Vorlauf und Anzeige-Aktualisierung konkurrieren um die NC-Rechenzeit.

Über das folgende Maschinendatum kann eingestellt werden, wie sich die NC verhalten soll, wenn der Vorlauf zu langsam ist:

MD10131 \$MN_SUPPRESS_SCREEN_REFRESH

Wert	Bedeutung
0	Wenn der VL eines Kanals zu langsam ist, wird die Aktualisierung der Anzeige in allen Kanälen unterdrückt.
1	Wenn der VL eines Kanals zu langsam ist, wird die Aktualisierung der Anzeige nur in den zeitkritischen Kanälen unterdrückt, um Rechenzeit für den Vorlauf zu gewinnen.
2	Die Aktualisierung der Anzeige wird grundsätzlich nicht unterdrückt.

11.2.2 DEFAULT-Speicher aktivieren

GUD Startwerte

Mit den Sprachbefehlen DEF... / REDEF... können globalen Anwendervariablen (GUD) Defaultwerte zugewiesen werden. Damit diese Defaultwerte zum parametrierten Initialisierungszeitpunkt, z. B. durch das Attribut `INIPO` nach Power On, verfügbar sind, müssen diese im System dauerhaft gespeichert werden. Der dafür erforderliche Speicherplatz muss über folgendes Maschinendatum freigegeben werden:

MD18150 \$MM_GUD_VALUES_MEM (remanenter Speicherplatz für GUD-Werte)

Weitere Informationen

- Speicherkonfiguration (Seite 837)
- Programmierhandbuch Arbeitsvorbereitung

11.2.3 PLC-Variable lesen und schreiben

Schneller Datenkanal

Für den schnellen Austausch von Informationen zwischen PLC und NC wird in dem Koppelspeicher dieser Baugruppen (DPR) ein Speicherbereich reserviert. In diesem Speicherbereich können beliebige PLC-Variablen (E/A, DB, DW, Merker) ausgetauscht werden.

Auf diesen Speicher wird von der PLC mit 'FunctionCalls' (FC), von der NC mit Systemvariablen zugegriffen.

Organisation des Speicherbereichs

Für die Organisation (Struktur) des Speicherbereiches ist der Anwenderprogrammierer (NC und PLC) selbst verantwortlich.

Dabei kann jede beliebige Speicherstelle angesprochen werden, jedoch muss dabei die Grenze entsprechend dem Datenformat gewählt werden (ein DWORD also auf einer 4-Byte-Grenze, ein WORD auf einer 2-Byte-Grenze ...).

Der Zugriff in den Speicherbereich erfolgt über den Datentyp und den Positionsoffset innerhalb des Speicherbereichs.

Zugriff von NC

Für den schnellen Zugriff aus einem Teileprogramm oder Synchronaktion auf PLC-Variable, werden in der NC Systemvariablen zur Verfügung gestellt. Das Schreiben / Lesen der Daten durch die NC erfolgt direkt. Der Datentyp ergibt sich aus dem Bezeichner der Systemvariablen. Die Position innerhalb des Speicherbereichs wird als Index in Byte angegeben.

Systemvariable	Datentyp	Wertebereich
\$A_DBB(<Index>)	Byte (8 Bit)	$0 \leq x \leq 255$
\$A_DBW(<Index>)	Wort (16 Bit)	$-32768 \leq x \leq 32767$
\$A_DBD(<Index>)	Doppelwort (32 Bit)	$-2147483648 \leq x \leq 2147483647$
\$A_DBR(<Index>)	Fließkomma (32 Bit)	$\pm(1,5 \cdot 10^{-45} \leq x \leq 3,4 \cdot 10^{38})$

Zugriff von PLC

Der Zugriff von der PLC erfolgt mittels "FunctionCall" (FC). Im FC werden die Daten sofort und nicht erst bei Zyklusbeginn der PLC, im DPR gelesen bzw. geschrieben. Datentyp und Position im Speicherbereich werden dem FC als Parameter übergeben.

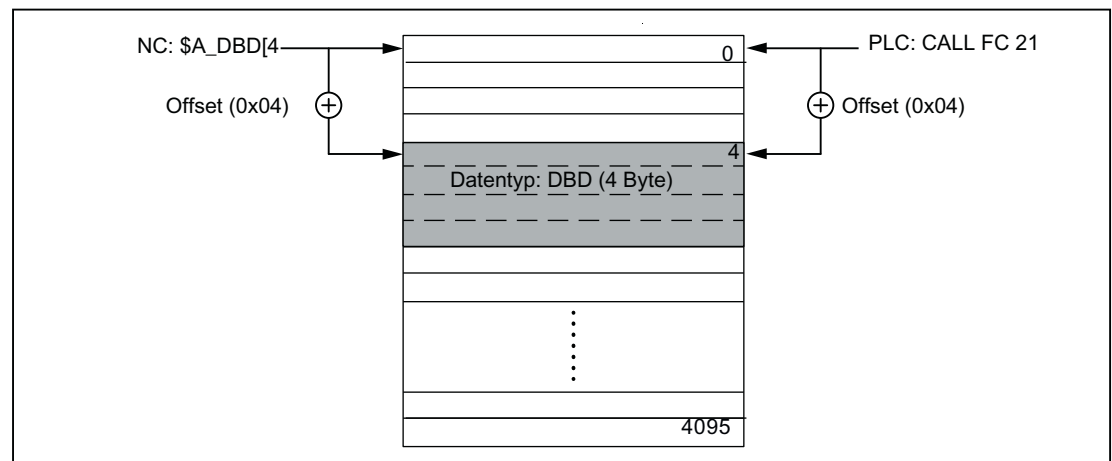


Bild 11-1 Koppelspeicher DPR für die NC/PLC-Kommunikation

Randbedingungen

- Die Strukturierung des DPR-Speicherbereichs liegt ausschließlich in der Verantwortung der Anwenders. Es werden keine Überprüfungen auf übereinstimmende Projektierung vorgenommen.
- In Ein- und Ausgaberrichtung stehen in Summe 4096 Bytes zur Verfügung.
- Einzelbitoperationen werden nicht unterstützt und müssen vom Anwender auf Byte-Operationen zurückgeführt werden.
- Da die Manipulation der Inhalte der Variablen im Koppelspeicher unmittelbar erfolgt, ist durch den Anwenderprogrammierer darauf zu achten, dass bei mehrfacher Auswertung einer Variablen oder bei der Verknüpfung von Variablen sich die Werte zwischenzeitlich geändert haben (ggf. Werte in lokaler Variable oder R-Parameter zwischenspeichern oder eine Semaphore einrichten).

11.2 Funktionen

- Für die Koordination der Zugriffe von verschiedenen Kanälen auf den Koppelspeicher ist der Anwenderprogrammierer verantwortlich.
- Es wird nur für die Datenzugriffe bis 16 Bit (Byte und Word) Datenkonsistenz zugesichert. Für konsistente Übertragung von 32 Bit-Variablen (Double und Real) ist ausschließlich der Anwender verantwortlich. Auf der PLC-Seite steht dazu ein einfacher Semaphor-Mechanismus zur Verfügung.
- Die Daten werden von der PLC im 'Little Endian'-Format im DPR abgelegt.
- Mit \$A_DBR transferierte Werte unterliegen einer Datenwandlung und damit einem Genauigkeitsverlust. Das Datenformat für Fließkommazahlen ist in der NC DOUBLE (64Bit), in der PLC jedoch nur FLOAT (32Bit). Die Ablage im DPR ist FLOAT. Die Wandlung erfolgt jeweils vor/nach der Ablage im DPR.
Wird z.B. von der NC auf eine Variable im DPR geschrieben und wieder zurückgegeben, wird die Wandlung zweimal durchlaufen. Eine Differenz des geschriebenen/gelesenen Wertes ist aufgrund der Datenablage in den beiden Formaten nicht zu verhindern.

Beispiel

Umgehung des Problems durch Vergleich auf "EPSILON" (kleine Abweichung)

Programmcode	
N10	DEF REAL DBR
N12	DEF REAL EPSILON = 0.00001
N20	\$A_DBR[0]=145.145
N30	G4 F2
N40	STOPRE
N50	DBR=\$A_DBR[0]
N60	IF (ABS(DBR/145.145-1.0) < EPSILON) GOTOF ENDE
N70	MSG ("Fehler")
N80	M0
N90	ENDE:
N99	M30

Aktivierung

Die maximale Anzahl der gleichzeitig schreibbaren Ausgangsvariablen ist einstellbar über: MD28150 \$MC_MM_NUM_VDIVAR_ELEMENTS (Anzahl Elemente für das schreiben von PLC-Variablen)

Beispiel

Es soll eine Variable vom Typ WORD von der PLC an die NC übergeben werden.

Der Positions-Offset innerhalb des NC-Eingangs (PLC-Ausgangsbereich) soll das 4. Byte sein. Der Positions-Offset muss ein ganzzahliges Vielfaches der Datenbreite sein.

Schreiben von der PLC:

Programmcode	Kommentar
...	
CALL FC21 (

Programmcode	Kommentar
Enable :=M10.0,	; wenn TRUE, dann FC21 aktiv
Funct :=B#16#4,	
S7Var :=P#M 104.0 WORD1,	
IVAR1 :=04,	
IVAR2 :=-1,	
Error :=M10.1,	
ErrCode :=MW12);	
. . .	
)	

Lesen im Teileprogramm

Programmcode	Kommentar
. . .	
PLCDATA = \$A_DBW[4];	; Lesen eines Wortes
. . .	

Verhalten bei POWER ON, Satzsuchlauf

Bei "POWER ON" wird der Koppelspeicher DPR initialisiert.

Bei "Satzsuchlauf" wird die Ausgabe der PLC-Variablen aufgesammelt und mit dem Anfahr Satz in den Koppelspeicher DPR ausgegeben (analog dem Schreiben von analogen und digitalen Ausgängen).

Andere Zustandsübergänge haben hier keine Auswirkung.

Weitere Informationen

Eine ausführliche Beschreibung zum Datenaustausch seitens der PLC mit FC 21 findet sich in:
Funktionshandbuch PLC und Grundprogramm

11.2.4 Zugriffsschutz über Kennwort und Schlüsselschalter

Zugriffsrechte

Der Zugriff auf Funktionen, Programme und Daten und ist benutzerorientiert über 8 hierarchische Schutzstufen geschützt. Diese sind unterteilt in:

- Kennwort-Stufen für Siemens, Maschinenhersteller und Endanwender
- Schlüsselschalter-Stellungen für Endanwender

Mehrstufiges Sicherheitskonzept

Über die Kennwort-Stufen und Schlüsselschalter-Stellungen steht ein mehrstufiges Sicherheitskonzept zur Regelung der Zugriffsrechte zur Verfügung.

Schutzstufe	Art	Benutzer	Zugriff auf (Beispiele)
0	Kennwort	Siemens	alle Funktionen, Programme und Daten
1	Kennwort	Maschinenhersteller: Entwicklung	definierte Funktionen, Programme und Daten; z.B.: Optionen eingeben
2	Kennwort	Maschinenhersteller: Inbetriebnehmer	definierte Funktionen, Programme und Daten; z.B.: Großteil der Maschinendaten
3	Kennwort	Endanwender: Service	zugeordnete Funktionen, Programme und Daten
4	Schlüsselschalter-Stellung 3	Endanwender: Programmierer Einrichter	weniger als Schutzstufe 0 bis 3; festgelegt vom Maschinenhersteller oder Endanwender
5	Schlüsselschalter-Stellung 2	Endanwender: Qualifizierter Bediener, der nicht programmiert	weniger als Schutzstufe 0 bis 3; festgelegt vom Endanwender
6	Schlüsselschalter-Stellung 1	Endanwender: Ausgebildeter Bediener, der nicht programmiert	Beispiel: nur Programmanwahl, Werkzeugverschleißeingabe und Eingabe von Nullpunktverschiebungen
7	Schlüsselschalter-Stellung 0	Endanwender: Angelernter Bediener	Beispiel: keine Eingaben und Programmanwahl möglich, nur Maschinensteuertafel bedienbar

Zugriffsmerkmale

- Die Schutzstufe 0 besitzt die höchsten Zugriffsrechte, die Schutzstufe 7 hat die geringsten Zugriffsrechte.
- Wird einer Schutzstufe ein Zugriffsrecht erteilt, so schließt die höhere Schutzstufe automatisch dieses Zugriffsrecht mit ein.
- Umgekehrt kann ein Zugriffsrecht für eine bestimmte Schutzstufe nur aus einer höheren Schutzstufe heraus geändert werden.
- Die Zugriffsrechte für die Schutzstufen 0 bis 3 werden von Siemens standardmäßig vorgegeben (Default).
- Die Zugriffsberechtigung wird durch Abfrage der aktuellen Schlüsselschalterstellung und durch Vergleich der eingegebenen Kennwörter gesetzt. Dabei überschreibt ein eingegebenes Kennwort die Zugriffsrechte der Schlüsselschalterstellung.

- In jeder Schutzstufe können Optionen gesichert werden. Aber nur in Schutzstufe 0 und 1 können Optionsdaten eingegeben werden.
- Die Zugriffsrechte für die Schutzstufen 4 bis 7 sind Vorschlagswerte und können vom Maschinenhersteller oder Endanwender geändert werden.

11.2.4.1 Kennwort

Kennwort setzen

Das Kennwort einer Schutzstufe (0 - 3) wird über die HMI-Bedienoberfläche eingegeben:

Bedienbereich "Inbetriebnahme" > "Kennwort" > "Kennwort setzen"

Kennwort löschen

Die Zugriffsberechtigung durch ein gesetztes Kennwort bleibt so lange wirksam, bis sie explizit durch Löschen des Kennworts zurückgenommen wird:

Bedienbereich "Inbetriebnahme" > "Kennwort" > "Kennwort löschen"

Kennwort ändern

Das Kennwort einer Schutzstufe (0 - 3) wird über die HMI-Bedienoberfläche geändert:

Bedienbereich "Inbetriebnahme" > "Kennwort" > "Kennwort ändern"

Hinweis

Warmstart

Eine Zugriffsberechtigung bzw. der Status des Kennworts (gesetzt / gelöscht) wird durch Warmstart nicht beeinflusst!

Neues Kennwort

Ein Kennwort kann maximal aus acht Zeichen bestehen. Bei der Wahl des Kennworts wird empfohlen, sich auf den Zeichenvorrat der Bedientafelfront zu beschränken.

Bei einem Kennwort mit weniger als 8 Zeichen werden die restlichen Zeichen als Leerzeichen (Blank) interpretiert.

Bei der Vergabe eines neuen Kennworts müssen die Regeln zur Vergabe von sicheren Passwörtern beachtet werden.

Hinweis

Sichere Passwörter vergeben

Beachten Sie bei der Vergabe von neuen Passwörtern die folgenden Regeln:

- Beachten Sie bei der Vergabe von neuen Passwörtern, dass Sie niemals leicht zu erratende Passwörter vergeben, z. B. einfache Wörter, leicht zu erratende Tastenfolgen auf der Tastatur, etc.
- Passwörter müssen immer eine Kombination aus Groß- und Kleinbuchstaben enthalten sowie Zahlen und Sonderzeichen. Passwörter müssen aus mindestens 8 Zeichen bestehen. PINS müssen aus einer willkürlichen Abfolge von Ziffern bestehen.
- Sie müssen – wo immer es möglich ist und wo es von den IT-Systemen und der Software unterstützt wird – stets die höchste Komplexität von Zeichen für die Wahl eines Passworts ansetzen.

Weitere Regeln zur Vergabe von sicheren Passwörtern finden Sie beim Bundesamt für Sicherheit in der Informationstechnik (BSI). (<https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/content/m/m02/m02011.html>)

Zur Unterstützung im Umgang mit Passwörtern können Sie ein Programm zur Passwortverwaltung nutzen. Mit dessen Hilfe können Sie Passwörter und Geheimzahlen verschlüsselt speichern, verwalten sowie sichere Passwörter erzeugen.

Standardeinstellungen

Standardmäßig sind für die Schutzstufen 1 - 3 folgende Kennwörter wirksam:

- Schutzstufe 1: SUNRISE
- Schutzstufe 2: EVENING
- Schutzstufe 3: CUSTOMER

ACHTUNG
Kaltstart bewirkt ein Rücksetzen der Kennwörter auf die Standardeinstellungen
Nach einem Kaltstart (Hochlauf der NC mit Laden der Standard-Maschinendaten) sind die Kennwörter der Schutzstufen 1 - 3 wieder auf die Standardeinstellungen zurückgesetzt. Aus Sicherheitsgründen wird daher nach einem Kaltstart dringend empfohlen, die Standardeinstellungen der Kennwörter zu ändern.

Weitere Informationen

- Inbetriebnahmehandbuch Inbetriebnahme CNC: NC, PLC, Antrieb; Voraussetzungen für die Inbetriebnahme > Einschalten und Hochlauf und Zugriffsstufen

11.2.4.2 Schlüsselschalter-Stellungen (DB10, DBX56.4 bis 7)

Schlüsselschalter

Der Schlüsselschalter verfügt über vier Schalterstellungen, denen die Schutzstufen 4 bis 7 zugeordnet sind. Zum Schlüsselschalter gehören verschiedenfarbige Schlüssel, die in unterschiedlichen Schalterstellungen abgezogen werden können.





Schalterstellung	Abzugsstellung	DB10, DBB56	Schutzstufe
Stellung 0 	-	Bit 4	7
Stellung 1 	0 oder 1 schwarzer Schlüssel	Bit 5	6
Stellung 2 	0 oder 1 oder 2 grüner Schlüssel	Bit 6	5
Stellung 3 	0 oder 1 oder 2 oder 3 roter Schlüssel	Bit 7	4

Bild 11-2 Schalterstellungen 0 bis 3

Schalterstellungen

Schalterstellung 0 hat die geringsten Zugriffsrechte. Schalterstellung 3 hat die höchsten Zugriffsrechte:

DB10, DBX56.4 / .5 / .6 / .7 (Schalterstellungen 0 / 1 / 2 / 3)

Den Schalterstellungen können maschinenspezifische Freigaben für den Zugriff auf Programme, Daten und Funktionen zugeordnet werden. Ausführliche Informationen dazu finden sich in:

Weitere Informationen

- Inbetriebnahmehandbuch Inbetriebnahme CNC: NC, PLC, Antrieb; Voraussetzungen für die Inbetriebnahme > Einschalten und Hochlauf und Zugriffsstufen

Vorgaben durch das PLC-Anwenderprogramm

Die Schalterstellungen des Schlüsselschalters werden durch PLC-Grundprogramm in die NC/PLC-Nahtstelle übertragen. Die entsprechenden Nahtstellensignale können durch das PLC-Anwenderprogramm verändert werden. Dabei sollte aus Sicht der NC immer nur eine Schalterstellung aktiv, d.h. das entsprechende Nahtstellensignal auf den Wert 1 gesetzt sein. Sind aus Sicht der NC gleichzeitig mehrere Schalterstellungen aktiv, wird NC-intern die Schalterstellung 3, d.h. die Schlüsselschalterstellung mit den höchsten Zugriffsrechten aktiv.

11.2.4.3 Parametrierbare Schutzstufen

Parametrierbare Schutzstufe

Für verschiedene Funktionen und Datenbereiche kann die Schutzstufe frei parametriert werden. Die Einstellung der Schutzstufe erfolgt über Bedientafel-Maschinendaten mit folgender Bezeichnungssystematik: \$MM_USER_CLASS_<Funktion_Datenbereich>

Beispiele:

\$MM_USER_CLASS_READ_TOA	Werkzeugkorrekturen lesen
\$MM_USER_CLASS_WRITE_TOA	Werkzeugkorrekturen schreiben
\$MM_USER_CLASS_READ_PROGRAM	Teileprogramme lesen
\$MM_USER_CLASS_WRITE_PROGRAM	Teileprogramme schreiben/editieren

Standardwerte

Bei Auslieferung oder nach einer Standard-Inbetriebnahme ist der Standardwert für die Schutzstufe bis auf wenige Ausnahmen auf 7, d.h. niedrigste Schutzstufe eingestellt.

11.2.5 Umschalten von Motor- / Antriebsdatensätzen

11.2.5.1 Allgemeine Informationen

Motor- und Antriebsdatensätzen

Zur optimalen Anpassung an die jeweilige Bearbeitungssituation oder aufgrund unterschiedlicher Maschinenkonfigurationen kann es erforderlich sein, dass in einem Antrieb für Motoren, Antriebsparameter und Geber jeweils mehrere unterschiedliche Datensätze vorliegen. Die Erstellung der grundlegenden Datensätze der Antriebsobjekte erfolgt während der Inbetriebnahme mit Hilfe des "Antriebsassistenten".

Hinweis

Weitere Informationen

Inbetriebnahmehandbuch: CNC: NC, PLC- Antrieb, Inbetriebnahme NC-gesteuerte Antriebe

Die nachfolgende Duplizierung und Verwaltung der Datensätze erfolgt über die Bedienoberfläche:

SINUMERIK Operate: Bedienbereich "Inbetriebnahme" > "Antriebssystem" > "Antriebe" > "Datensätze"

Die Aktivierung des für eine Maschinenachse in der konkreten Bearbeitungssituation erforderlichen Motor- (MDS) bzw. Antriebsdatensatzes (DDS), muss vom PLC-Anwenderprogramm über die im Folgenden beschriebenen Schnittstellen vorgenommen werden.

Axiale NC/PLC-Nahtstelle

Die Schnittstellen in der axialen NC/PLC-Nahtstelle zur Umschaltung der Motor- und Antriebsdatensätze ist in drei Bereiche unterteilt:

- Formatierungsschnittstelle (Seite 637)
- Anforderungsschnittstelle (Seite 638)
- Anzeigeschnittstelle (Seite 638)

11.2.5.2 Formatierungsschnittstellen

Formatierung

Über die Formatierungsschnittstelle wird eingestellt, welche welche Bits der Anforderungs- und Anzeigeschnittstelle zur Adressierung der Motordatensätze (MDS) und welche zur Adressierung der Antriebsdatensätze (DDS) verwendet werden:

DB31, ... DBX130.0 - 4, mit Bit x = <Wert>

<Wert>	Bedeutung
0	Bit-Stelle für Motordatensatz (MDS) bzw. ungültige Bit-Stelle
1	Bit-Stelle für Antriebsdatensatz (DDS)

Motor- und Antriebsdatensätzen im Antrieb

Die Formatierung ist abhängig von der Anzahl der im Antrieb vorliegenden Motor- (MDS) und Antriebsdatensätzen (DDS). Die jeweilige Anzahl kann über folgende Antriebsparameter ermittelt werden:

- p0130 (Anzahl Motordatensätze)
- p0180 (Anzahl Antriebsdatensätze)

Gültigkeit der Schnittstellen

Sobald im Hochlauf der Steuerung alle erforderlichen Informationen vom Antrieb übermittelt und von der NC ausgewertet wurden, werden die Anforderungs (Seite 638)- und Anzeigeschnittstelle (Seite 638) als gültig angezeigt:

DB31, ... DBX130.7 == 1 (Anforderungs- und Anzeigeschnittstelle gültig)

Werden vom Antrieb keine oder inkompatible Informationen übertragen, werden die Anforderungs- und Anzeige-Schnittstellen als ungültig angezeigt.

Hinweis

Bei ungültigen Anforderungs- und Anzeigeschnittstellen liegt es in der alleinigen Verantwortung des Anwenders / Maschinenhersteller über die ungültigen Schnittstellen dennoch eine Datensatzumschaltung vorzunehmen.

Siehe auch

Beispiel (Seite 639)

Übersicht der Schnittstellen (Seite 640)

11.2.5.3 Anforderungsschnittstelle

Das Umschalten auf einen neuen Motor- (MDS) und/oder Antriebsdatensatz (DDS) wird angefordert über die Schnittstelle :

DB31, ... DBX21.0 - .4 = <MDS / DDS-Index>

Wertebereich

Die Adressierung eines Motor- oder Antriebsdatensatzes n, mit $n = 1, 2, 3, \dots$, erfolgt anhand seines Index i, mit $i = n - 1 = 0, 1, 2, \dots$

- Motordatensätze: MDS[0, 1, 2, ... 15]
- Antriebsdatensätze: DDS[0, 1, 2, ... 31]

Schnittstellenformatierung

Die Formatierung der Anforderungsschnittstelle, d. h. welche Bits zur Adressierung der Motordatensätze (MDS) und welche zur Adressierung der Antriebsdatensätze (DDS) verwendet werden, wird über die Formatierungsschnittstelle (Seite 637) eingestellt.

Motor- und Antriebsdatensätzen im Antrieb

Die Anzahl der im Antrieb vorliegenden Motor- (MDS) und Antriebsdatensätzen (DDS) kann über folgende Antriebsparameter ermittelt werden:

- p0130 (Anzahl Motordatensätze)
- p0180 (Anzahl Antriebsdatensätze)

Motordatensatz (MDS) bei Hauptspindelantrieben

Bei Hauptspindelantrieben gilt folgender Zusammenhang:

- MDS[0] → Stern-Betrieb
- MDS[1] → Dreieck-Betrieb

11.2.5.4 Anzeigeschnittstelle

Der aktive Motor- (MDS) und Antriebsdatensatzes (DDS) wird angezeigt über die Schnittstelle:

DB31, ... DBX93.0 - .4 == <MDS / DDS-Index>

Wertebereich und Formatierung sind identisch zur Anforderungsschnittstelle (Seite 638).

11.2.5.5 Beispiel

Im Antrieb sind zwei Motordatensätze (MDS) und zwei Antriebsdatensätze (DDS) pro Motordatensatz vorhanden. Dies entspricht "Nr.": 9 der im Bild 11-3 Prinzip der Motor- / Antriebsdatensatz-Umschaltung (Seite 640) dargestellten möglichen Datensatzkombinationen.

Format

Bit-Stellen für Antriebsdatensatz-Umschaltung (DDS):

- DB31, ... DBX130.0 == 1

Bit-Stellen für Motordatensatz -Umschaltung (MDS):

- DB31, ... DBX130.1 == 0

Ungültige Bit-Stellen:

- DB31, ... DBX130.2 == 0
- DB31, ... DBX130.3 == 0
- DB31, ... DBX130.4 == 0

Schnittstellen der Antriebsdatensätze (DDS)

Relevante Bit-Stellen der Anforderungs- und Anzeige-Schnittstellen:

- DB31, ... DBX21.0 / DBX93.0
 - DB31, ... DBX21.0 / DBX93.0 == 0 \Rightarrow 1. Antriebsdatensatz DDS[0]
 - DB31, ... DBX21.0 / DBX93.0 == 1 \Rightarrow 2. Antriebsdatensatz DDS[1]

Schnittstellen der Motordatensätze (MDS)

Relevante Bit-Stellen der Anforderungs- und Anzeige-Schnittstellen:

- DB31, ... DBX21.1 / DBX93.1
 - DB31, ... DBX21.1 / DBX93.1 == 0 \Rightarrow 1. Motordatensatz MDS[0]
 - DB31, ... DBX21.1 / DBX93.1 == 1 \Rightarrow 2. Motordatensatz MDS[1]

Ungültige Bit-Stellen (MDS / DDS)

Ungültige Bit-Stellen der Anforderungs- und Anzeige-Schnittstellen:

- DB31, ... DBX21.1 / DBX93.2 == 0
- DB31, ... DBX21.1 / DBX93.3 == 0
- DB31, ... DBX21.1 / DBX93.4 == 0

Siehe auch

Übersicht der Schnittstellen (Seite 640)

Bild 11-3 Prinzip der Motor- / Antriebsdatensatz-Umschaltung

11.2.5.7 Randbedingungen

Variable Anzahl Antriebsdatensätze für den "letzten" Motordatensatz

Der "letzte" Motordatensatz ist der Motordatensatz mit der höchsten Nummer bzw. Index.

Im Allgemeinen gilt, dass im Antrieb für jeden Motordatensatz die gleiche Anzahl von Antriebsdatensätzen (Anzahl "DDS pro MDS") angelegt wird. Nur für den "letzten" Motordatensatz kann davon abweichend, eine beliebige Anzahl von Antriebsdatensätzen a parametrierbar werden kann:

$$1 \leq a \leq (\text{Anzahl "DDS pro MDS"})$$

Beispiel

Es sollen 4 Motordatensätze (MDS) und 8 Antriebsdatensätze (DDS) pro Motordatensatz (DDS pro MDS) parametrierbar werden. Dies entspricht "Nr.": 22 der im Bild 11-3 Prinzip der Motor- / Antriebsdatensatz-Umschaltung (Seite 640) dargestellten möglichen Datensatzkombinationen:

- Motordatensätze: MDS[0], MDS[1], ... MDS[3] ("letzter" Motordatensatz)
- Antriebsdatensätzen pro Motordatensatz: DDS[0] ... DDS[7]

Die Anzahl Antriebsdatensätze für die einzelnen Motordatensätze ist somit:

Motordatensatz (MDS)	Anzahl Antriebsdatensätze (DDS) pro Motordatensatz (MDS)
MDS[0] ... MDS[2]	8
MDS[3]	1 - 8

Umschaltzeitpunkt: Antriebsparametersatz

Prinzipiell ist das Umschalten der Antriebsparametersätze zu jedem beliebigen Zeitpunkt möglich. Während eine Achse verfährt, können insbesondere bei der Umschaltung von Drehzahlreglerparameter und Motordrehzahlnormierung Momentensprünge auftreten. Es wird daher empfohlen, eine Umschaltung des Antriebsparametersatzes nur bei stationären Zuständen, insbesondere bei Achsstillstand, vorzunehmen.

Siehe auch

Übersicht der Schnittstellen (Seite 640)

11.3 Beispiele

11.3.1 Parametersatzumschaltung

Parametersatzumschaltung

Über eine Parametersatzumschaltung wird für die Maschinenachse X1 der Verstärkungsfaktor der Lageregelung (K_V -Faktor) von $K_V = 4.0$ auf $K_V = 0.5$ umgeschaltet.

Voraussetzungen

Die Parametersatzumschaltung muss freigegeben sein durch das Maschinendatum:

MD35590 \$MA_PARAMSET_CHANGE_ENABLE [AX1] = 1 oder 2 (Parametersatzwechsel möglich)

Angewählt ist der 1. Parametersatz der Maschinenachse X1, entsprechend den Maschinendaten mit dem Index "0". NC/PLC-Nahtstelle:

DB31, ... DBX9.0 - DBX9.2 = 0 (Regler Parametersatz)

Parametersatz-abhängige Maschinendaten

Die Parametersatz-abhängigen Maschinendaten sind wie folgt eingestellt:

Maschinendatum	Bemerkung
MD32200 \$MA_POSCTRL_GAIN [0, AX1] = 4.0	K_V -Einstellung für Parametersatz 1
MD32200 \$MA_POSCTRL_GAIN [1, AX1] = 2.0	K_V -Einstellung für Parametersatz 2
MD32200 \$MA_POSCTRL_GAIN [2, AX1] = 1.0	K_V -Einstellung für Parametersatz 3
MD32200 \$MA_POSCTRL_GAIN [3, AX1] = 0.5	K_V -Einstellung für Parametersatz 4
MD32200 \$MA_POSCTRL_GAIN [4, AX1] = 0.25	K_V -Einstellung für Parametersatz 5
MD32200 \$MA_POSCTRL_GAIN [5, AX1] = 0.125	K_V -Einstellung für Parametersatz 6
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [0, AX1] = 3	Nenner Lastgetriebe für Parametersatz 1
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [1, AX1] = 3	Nenner Lastgetriebe für Parametersatz 2
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [2, AX1] = 3	Nenner Lastgetriebe für Parametersatz 3
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [3, AX1] = 3	Nenner Lastgetriebe für Parametersatz 4
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [4, AX1] = 3	Nenner Lastgetriebe für Parametersatz 5
MD31050 \$MA_DRIVE_AX_RATIO_DENOM [5, AX1] = 3	Nenner Lastgetriebe für Parametersatz 6
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [0, AX1] = 5	Zähler Lastgetriebe für Parametersatz 1
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [1, AX1] = 5	Zähler Lastgetriebe für Parametersatz 2
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [2, AX1] = 5	Zähler Lastgetriebe für Parametersatz 3
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [3, AX1] = 5	Zähler Lastgetriebe für Parametersatz 4
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [4, AX1] = 5	Zähler Lastgetriebe für Parametersatz 5
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA [5, AX1] = 5	Zähler Lastgetriebe für Parametersatz 6
MD35130 \$MA_AX_VELO_LIMIT [0..5, AX1]	Einstellung für jeden Parametersatz*)
MD32800 \$MA_EQUIV_CURRCTRL_TIME [0..5, AX1]	Einstellung für jeden Parametersatz*)

Maschinendatum	Bemerkung
MD32810 \$MA_EQUIV_SPEEDCTRL_TIME [0..5, AX1]	Einstellung für jeden Parametersatz*)
MD32910 \$MA_DYN_MATCH_TIME [0...5, AX1]	Einstellung für jeden Parametersatz*)
*) Die entsprechende Zeile ist für jeden Parametersatz nach den Syntaxregeln gesondert anzugeben.	

Umschaltung

Zur Umschaltung des Verstärkungsfaktor der Lageregelung wird vom PLC-Anwenderprogramm für die Maschinenachse X1 der 4. Parametersatz angewählt.

- Anforderung durch PLC-Anwenderprogramm:
 - DB31, ... DBX9.0 – DBX9.2 = 3 (Parametersatz Servo)
 - Für die Maschinenachse AX1 wird die Umschaltung auf den 4. Parametersatz angefordert.
 - Nach einer Verzögerungszeit wird der Parametersatz umgeschaltet.
 - Aktiv ist jetzt der Parametersatz 4, entsprechend den Maschinendaten mit dem Index "3"
- Rückmeldung durch NC:
 - DB31, ... DBX69.0 – DBX69.2 = 3 (Parametersatz Servo)
 - Die Umschaltung des Parametersatzes wird von der NC bestätigt / quittiert.

11.4 Datenlisten

11.4.1 Maschinendaten

11.4.1.1 Anzeige-Maschinendaten

Nummer	Bezeichner: \$MM_	Beschreibung
SINUMERIK Operate		
9000	LCD_CONTRAST	Kontrast
9001	DISPLAY_TYPE	Monitortyp
9004	DISPLAY_RESOLUTION	Anzeigefeinheit
9006	DISPLAY_SWITCH_OFF_INTERVAL	Zeit für Bildschirmdunkelschaltung

11.4.1.2 NC-spezifischen Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10350	FASTIO_DIG_NUM_INPUTS	Anzahl der aktiven digitalen NC-Eingangsbytes
10360	FASTIO_DIG_NUM_OUTPUTS	Anzahl der aktiven digitalen NC-Ausgangsbytes
10361	FASTIO_DIG_SHORT_CIRCUIT	Kurzschluss digitaler ein- und Ausgänge

11.4 Datenlisten

Nummer	Bezeichner: \$MN_	Beschreibung
11120	LUD_EXTENDED_SCOPE	Programmglobale Variablen aktivieren (PUD)
11270	DEFAULT_VALUES_MEM_MSK	Aktiv. Funktion: DEFAULT-Werte von GUD speichern.
18150	MM_GUD_VALUES_MEM	Speicherplatz für GUD reservieren

11.4.1.3 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
21015	INVOLUTE_RADIUS_DELTA	NC-Startsperre ohne Referenzpunkt
21016	INVOLUTE_AUTO_ANGLE_LIMIT	Automatische Winkelbegrenzung bei Evolventen-Interpolation
27800	TECHNOLOGY_MODE	Technologie im Kanal
28150	MM_NUM_VDIVAR_ELEMENTS	Anzahl Schreibelemente für PLC-Variable
28530	MM_PATH_VELO_SEGMENTS	Anzahl von Speicherelementen zur Begrenzung der Bahngeschwindigkeit im Satz

11.4.1.4 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30350	SIMU_AX_VDI_OUTPUT	Ausgabe der Achssignale bei Simulationsachsen
33050	LUBRICATION_DIST	Schmierimpulsdistanz
35590	PARAMSET_CHANGE_ENABLE	Parametersatzvorgabe durch PLC möglich
36060	STANDSTILL_VELO_TOL	Maximale Geschwindigkeit/Drehzahl bei Achse/Spindel steht
36610	AX_EMERGENCY_STOP_TIME	Zeitdauer der Bremsrampe bei Fehlerzuständen
36620	SERVO_DISABLE_DELAY_TIME	Abschaltverzögerung Reglerfreigabe

11.4.2 Systemvariablen

Bezeichner	Beschreibung
\$P_FUMB	freier Teileprogrammspeicher (F ree U ser M emory B uffer)
\$A_DBB[n]	Datum auf der PLC (Daten vom Type BYTE)
\$A_DBW[n]	Datum auf der PLC (Daten vom Type WORD)
\$A_DBD[n]	Datum auf der PLC (Daten vom Type DWORD)
\$A_DBR[n]	Datum auf der PLC (Daten vom Type REAL)

H2: Hilfsfunktionsausgaben an PLC

12.1 Kurzbeschreibung

12.1.1 Funktion

Hilfsfunktionen bieten die Möglichkeit, Systemfunktionen des NC und PLC-Anwenderfunktionen zu aktivieren. Hilfsfunktionen können programmiert werden in:

- Teileprogrammen
- Synchronaktionen
- Anwenderzyklen

Ausführliche Informationen zur Verwendung von Hilfsfunktionsausgaben in Synchronaktionen siehe:

Weitere Informationen

Funktionshandbuch Synchronaktionen

Vordefinierte Hilfsfunktionen

Vordefinierte Hilfsfunktionen aktivieren Systemfunktionen. Zusätzlich wird die Hilfsfunktion an die NC/PLC-Nahtstelle ausgegeben.

Vordefiniert sind folgende Hilfsfunktionen:

Typ	Funktion	Beispiel	Bedeutung
M	Zusatzfunktion	M30	Programmende
S	Spindelfunktion	S100	Spindeldrehzahl 100 (z. B. 1/min)
T	Werkzeugnummer	T3	Werkzeugnummer 3
D, DL	Werkzeugkorrektur	D1	Werkzeugschneidenummer 1
F	Vorschub	F1000	Vorschub 1000 (z. B. mm/min)

Anwenderdefinierte Hilfsfunktionen

Anwenderdefinierte Hilfsfunktionen sind entweder erweiterte vordefinierte Hilfsfunktionen oder anwenderspezifische Hilfsfunktionen.

Erweiterung von vordefinierten Hilfsfunktionen

Die Erweiterung von vordefinierten Hilfsfunktionen bezieht sich auf den Parameter "Adresserweiterung". Über die Adresserweiterung wird die Nummer der Spindel definiert, auf die sich die Hilfsfunktion bezieht. Vordefiniert ist z. B. die Spindelfunktion M3 (Spindel rechts) für die Masterspindel eines Kanals. Ist einem Kanal eine 2. Spindel zugeordnet, ist eine

entsprechende anwenderdefinierte Hilfsfunktionen zu definieren, welche die vordefinierte Hilfsfunktion erweitert.

Typ	Funktion	Beispiel	Bedeutung
M	Zusatzfunktion	M2=3	2. Spindel: Spindel rechts
S	Spindelfunktion	S2=100	2. Spindel: Spindeldrehzahl = 100 (z. B. 1/min)
T	Werkzeugnummer	T2=3	

Anwenderspezifische Hilfsfunktionen

Über anwenderspezifische Hilfsfunktionen werden keine Systemfunktionen aktiviert. Anwenderspezifische Hilfsfunktionen werden lediglich an die NC/PLC-Nahtstelle ausgegeben. Die Funktionalität der Hilfsfunktion ist vom Maschinenhersteller/Anwender im PLC-Anwenderprogramm zu realisieren.

Typ	Funktion	Beispiel	Bedeutung
H ¹⁾	Hilfsfunktion	H2=5	Anwenderspezifische Funktion

¹⁾ Empfehlung

12.1.2 Definition einer Hilfsfunktion

Eine Hilfsfunktion ist über folgende Parameter definiert:

- **Typ, Adresserweiterung und Wert**
Die 3 Parameter werden an die NC/PLC-Nahtstelle ausgegeben.
- **Ausgabeverhalten**
Über das Hilfsfunktionsspezifische Ausgabeverhalten wird festgelegt, wie lange eine Hilfsfunktion an die NC/PLC-Nahtstelle ausgegeben wird und wann die Ausgabe, bezogen auf die im gleichen Teileprogrammsatz programmierte Verfahrensbewegung, erfolgt.
- **Gruppenzuordnung**
Eine Hilfsfunktion kann einer bestimmten Hilfsfunktionsgruppe zugeordnet werden. Für jede Hilfsfunktionsgruppe kann ein eigenes Ausgabeverhalten festgelegt werden. Dieses wird wirksam, wenn kein Hilfsfunktionsspezifisches Ausgabeverhalten festgelegt wurde. Darüber hinaus beeinflusst die Gruppenzugehörigkeit die Ausgabe einer Hilfsfunktion nach Satzsuchlauf.

Weitere Informationen

Funktionshandbuch PLC und Grundprogramm, Grundprogramm

12.1.3 Übersicht der Hilfsfunktionen

M-Funktionen

M (Zusatzfunktion)					
Adresserweiterung		Wert			
Wertebereich	Bedeutung	Wertebereich ¹⁰⁾	Typ	Bedeutung	Anzahl ⁸⁾
0 (implizit)	---	0 ... 99	INT	Funktion	5
Wertebereich	Bedeutung	Wertebereich ¹¹⁾	Typ	Bedeutung	Anzahl ⁸⁾
1 ... 20	Spindelnummer	1 ... 99	INT	Funktion	5
Wertebereich	Bedeutung	Wertebereich ¹²⁾	Typ	Bedeutung	Anzahl ⁸⁾
0 ... 99	beliebig	100 ... 2147483647	INT	Funktion	5

⁸⁾ Siehe "Bedeutung der Fußnoten" am Ende der Übersicht

¹⁰⁾ Für den Wertebereich 0 ... 99 ist die Adresserweiterung 0. Zwingend ohne Adresserweiterung: M0, M1, M2, M17, M30

¹¹⁾ M3, M4, M5, M19, M70: Die Adresserweiterung ist die Spindel-Nummer, z. B. M2=5 ⇒ Spindel-Halt (M5) für Spindel 2. Ohne Adresserweiterung wirkt die M-Funktion auf die Masterspindel.

¹²⁾ Anwenderspezifische M-Funktionen.

Verwendung

Maschinenfunktionen synchron zum Teileprogramm steuern.

Weitere Informationen

- Folgende M-Funktionen haben vordefinierte Bedeutung: M0, M1, M2, M17, M30, M3, M4, M5, M6, M19, M70, M40, M41, M42, M43, M44, M45.
- Den M-Funktionen (M0 - M99) ist ein dynamisches NC/PLC-Nahtstellensignal zur Anzeige der Gültigkeit zugeordnet. Zusätzlich können 64 weitere Signale für eigene M-Funktionen zugeordnet werden (siehe Funktionshandbuch *PLC und Grundprogramm*, Kapitel *Grundprogramm*).
- Für Unterprogramme ist über Maschinendatum einstellbar, ob eine Ausgabe der M-Funktion für das Teileprogrammende M17, M2 oder M30 an die PLC erfolgen soll: MD20800 \$MC_SPF_END_TO_VDI (Unterprogrammende an PLC)
- Für die vordefinierten M-Funktionen M40 ... M45 kann die Ausgabespezifikation nur eingeschränkt umprojektiert werden.

12.1 Kurzbeschreibung

- Die vordefinierten Hilfsfunktionen M0, M1, M17, M30, M6, M4, M5 lassen sich nicht umprojektieren.
- M-Funktionsspezifische Maschinendaten:
 - MD10800 \$MN_EXTERN_CHAN_SYNC_M_NO_MIN
 - MD10802 \$MN_EXTERN_CHAN_SYNC_M_NO_MAX
 - MD10804 \$MN_EXTERN_M_NO_SET_INT
 - MD10806 \$MN_EXTERN_M_NO_DISABLE_INT
 - MD10814 \$MN_EXTERN_M_NO_MAC_CYCLE
 - MD10815 \$MN_EXTERN_M_NO_MAC_CYCLE_NAME
 - MD20094 \$MC_SPIND_RIGID_TAPPING_M_NR
 - MD20095 \$MC_EXTERN_RIGID_TAPPING_M_NR
 - MD20096 \$MC_T_M_ADDRESS_EXT_IS_SPINO
 - MD22200 \$MC_AUXFU_M_SYNC_TYPE
 - MD22530 \$MC_TOCARR_CHANGE_M_CODE
 - MD22532 \$MC_GEOAX_CHANGE_M_CODE
 - MD22534 \$MC_TRAFO_CHANGE_M_CODE
 - MD22560 \$MC_TOOL_CHANGE_M_CODE

S-Funktionen

S (Spindelfunktion)					
Adresserweiterung ¹⁰⁾		Wert			
Wertebereich	Bedeutung	Wertebereich	Typ	Bedeutung	Anzahl ⁸⁾
0 ... 20	Spindelnummer ⁵⁾	0 ... ± 3.4028 exp38 ³⁾	REAL	Spindeldrehzahl	3

³⁾ ⁵⁾ ⁸⁾ Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

¹⁰⁾ Ohne Angabe einer Adresserweiterung wird die Master-Spindel des Kanals adressiert.

Verwendung

Spindeldrehzahl.

Weitere Informationen

- Die S-Funktionen sind standardmäßig der 3. Hilfsfunktionsgruppe zugeordnet.
- Ohne Adresserweiterung beziehen sich die S-Funktionen auf die Masterspindel des Kanals.
- S-Funktionsspezifische Maschinendaten:
MD22210 \$MC_AUXFU_S_SYNC_TYPE (Ausgabezeitpunkt der S-Funktionen)

H-Funktionen

Die Funktionalität einer H-Funktion ist im PLC-Anwenderprogramm zu realisieren.

H (Hilfsfunktion) ¹⁰⁾					
Adresserweiterung		Wert			
Wertebereich	Bedeutung	Wertebereich	Typ	Bedeutung	Anzahl ⁸⁾
0 ... 99	beliebig	- 2147483648 ... + 2147483647	INT	beliebig	3
		0 ... ± 3.4028 exp38 ^{2) 3) 4)}	REAL		

^{2) 3) 4) 8)} Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

Verwendung

Anwenderspezifische Hilfsfunktionen.

Weitere Informationen

H-Funktionsspezifische Maschinendaten:

- MD22110 \$MC_AUXFU_H_TYPE_INT (Typ von H-Hilfsfunktionen ist Integer)
- MD22230 \$MC_AUXFU_H_SYNC_TYPE (Ausgabezeitpunkt der H-Funktionen)

T-Funktionen

Werkzeugnamen werden nicht an die NC/PLC-Nahtstelle ausgegeben.

T (Werkzeugnummer) ^{1) 5) 6)}					
Adresserweiterung		Wert			
Wertebereich	Bedeutung	Wertebereich	Typ	Bedeutung	Anzahl ⁸⁾
1 ... 12	Spindelnummer (bei aktiver Werkzeugverwaltung)	0 ... 32000 (auch symbolische Werkzeugnamen bei aktiver Werkzeugverwaltung)	INT	Anwahl des Werkzeugs	1

^{1) 5) 6) 8)} Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

Verwendung

Werkzeuganwahl.

Weitere Informationen

- Identifikation der Werkzeuge, wahlweise über Werkzeugnummer oder Platznummer (siehe Funktionshandbuch *Werkzeuge*, Kapitel *Werkzeugkorrektur*).
Funktionshandbuch Werkzeugverwaltung
- Mit T0 wird das aktuelle Werkzeug aus der Werkzeughalterung entfernt und kein neues eingewechselt (Grundeinstellung).
- T-Funktionsspezifische Maschinendaten:
MD22220 \$MC_AUXFU_T_SYNC_TYPE (Ausgabezeitpunkt der T-Funktionen)

D-Funktionen

Die Abwahl der Werkzeugkorrektur erfolgt mit D0. Vorbesetzung ist D1.

D (Werkzeugkorrektur)					
Adresserweiterung		Wert			
Wertebereich	Bedeutung	Wertebereich	Typ	Bedeutung	Anzahl ⁸⁾
---	---	0 ... 9	INT	Anwahl der Werkzeugkorrektur	1

⁸⁾ Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

Verwendung

Anwahl der Werkzeugkorrektur.

Weitere Informationen

- Grundstellung: D1
- Die Default-Werkzeugschneide nach Werkzeugwechsel ist parametrierbar über: MD20270 \$MC_CUTTING_EDGE_DEFAULT (Grundstellung der Werkzeugschneide ohne Programmierung)
- Abwahl der Werkzeugkorrektur: D0
- D-funktionsspezifische Maschinendaten: MD22250 \$MC_AUXFU_D_SYNC_TYPE (Ausgabezeitpunkt der D-Funktionen)

DL-Funktionen

Die mit DL angewählte Werkzeugsummenkorrektur bezieht sich auf aktive D-Nummer.

DL (Werkzeugsummenkorrektur)					
Adresserweiterung		Wert			
Wertebereich	Bedeutung	Wertebereich	Typ	Bedeutung	Anzahl ⁸⁾
---	---	0 ... 6	INT	Anwahl der Werkzeugsummenkorrektur	1

⁸⁾ Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

Verwendung

Anwahl der Werkzeugsummenkorrektur bezüglich einer aktiven Werkzeugkorrektur.

Weitere Informationen

- Grundstellung: DL = 0
- DL-Werte können nicht über Synchronaktionen an die PLC ausgegeben werden.
- Voreinstellung der Werkzeugsummenkorrektur ohne aktive DL-Funktion: MD20272 \$MC_SUMCORR_DEFAULT (Grundstellung Summenkorrektur ohne Programm)

- Abwahl der Werkzeugsummenkorrektur: DL = 0
- DL-Funktionsspezifische Maschinendaten:
MD22252 \$MC_AUXFU_DL_SYNC_TYPE (Ausgabezeitpunkt DL-Funktionen)

F-Funktionen

F (Bahnvorschub)					
Adresserweiterung		Wert			
Wertebereich	Bedeutung	Wertebereich	Typ	Bedeutung	Anzahl ⁸⁾
---	---	0.001 ... 999 999.999	REAL	Bahnvorschub	6

⁸⁾ Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

Verwendung

Bahngeschwindigkeit.

Weitere Informationen

F-funktionsspezifische Maschinendaten:

- MD22240 \$MC_AUXFU_F_SYNC_TYPE (Ausgabezeitpunkt der F-Funktionen)

FA-Funktionen

FA (achsspezifischer Vorschub)					
Adresserweiterung		Wert			
Wertebereich	Bedeutung	Wertebereich	Typ	Bedeutung	Anzahl ⁸⁾
1 - 31	Achsnummer	0.001 ... 999 999.999	REAL	Achsvorschub	6

⁸⁾ Siehe "Bedeutung der Fußnoten" am Ende der Übersicht.

Verwendung

Achsspezifische Geschwindigkeit.

Weitere Informationen

F-funktionsspezifische Maschinendaten:

- MD22240 \$MC_AUXFU_F_SYNC_TYPE (Ausgabezeitpunkt der F-Funktionen)

Bedeutung der Fußnoten

- ¹⁾ Bei aktiver Werkzeugverwaltung wird weder ein T-Änderungssignal noch ein T-Wort an die kanalspezifische NC/PLC-Nahtstelle ausgegeben.
- ²⁾ Der Typ für die Werte kann durch MD22110 \$MC_AUXFU_H_TYPE_INT vom Anwender gewählt werden.

- 3) Aufgrund der begrenzten Anzeigemöglichkeiten auf den Bildschirmen der Bediengeräte sind die angezeigten Werte des Typs REAL begrenzt auf:
–999 999 999.9999 bis 999 999 999.9999
Die NC rechnet intern aber mit der vollen Genauigkeit.
- 4) Die REAL–Werte werden gerundet an die PLC ausgegeben, bei der Einstellung des Maschinendatums:
MD22110 \$MC_AUXFU_H_TYPE_INT = 1 (Typ von H-Hilfsfunktionen ist Integer)
Das PLC-Anwenderprogramm muss den übergebenen Wert entsprechend der Maschinendaten-Einstellung interpretieren.
- 5) Bei aktiver Werkzeugverwaltung kann die Bedeutung der Adresserweiterung parametrierbar werden. Adresserweiterung = 0 bedeutet, der Wert ist durch den der Masterspindelnummer zu ersetzen; ist somit gleichbedeutend mit Nichtprogrammierung der Adresserweiterung.
Bei Satzsuchlauf aufgesammelte Hilfsfunktionen M19 "Spindel positionieren" werden nicht an die PLC ausgegeben.
- 6) M6: Wertebereich der Adresserweiterung:
- ohne Werkzeugverwaltung: 0 ... 99
- mit Werkzeugverwaltung: 0 ... maximale Spindelnummer
0: Ersetzen durch Wert der Masterspindelnummer bzw. des Masterwerkzeughalters
- 7) Mit aktiver Werkzeugverwaltung kann die Hilfsfunktion M6 "Werkzeugwechsel" nur einmal in einem Teileprogrammsatz programmiert werden, unabhängig davon, welche Adresserweiterungen programmiert werden.
- 8) Maximale Anzahl von Hilfsfunktionen pro Teileprogrammsatz.

12.2 Vordefinierte Hilfsfunktionen

Funktion

Jeder vordefinierten Hilfsfunktion ist nicht änderbar eine Systemfunktion zugeordnet. Ist in einem Teileprogramm / Zyklus eine vordefinierte Hilfsfunktion programmiert, wird diese über die NC/PLC-Nahtstelle an die PLC ausgegeben und im NC die entsprechende Systemfunktion ausgeführt.

Definition einer vordefinierten Hilfsfunktion

Die Parameter der vordefinierten Hilfsfunktionen sind in Maschinendaten hinterlegt und können teilweise geändert werden. Alle Maschinendaten die zu einer Hilfsfunktion gehören, haben den gleichen Index <n>.

- MD22040 \$MC_AUXFU_PREDEF_GROUP[<n>] (Gruppenzuordnung von vordefinierten Hilfsfunktionen)
- MD22050 \$MC_AUXFU_PREDEF_TYPE[<n>] (Typ von vordefinierten Hilfsfunktionen)
- MD22060 \$MC_AUXFU_PREDEF_EXTENSION[<n>] (Adresserweiterung für vordefinierte Hilfsfunktionen)

- MD22070 \$MC_AUXFU_PREDEF_VALUE[<n>] (Wert von vordefinierten Hilfsfunktionen)
- MD22080 \$MC_AUXFU_PREDEF_SPEC[<n>] (Ausgabeverhalten von vordefinierten Hilfsfunktionen)

12.2.1 Übersicht: vordefinierten Hilfsfunktionen

Bedeutung der in den nachfolgenden Tabellen aufgeführten Parameter:

Parameter	Bedeutung
Index <n>	Maschinendatenindex der Parameter einer Hilfsfunktion
Typ	MD22050 \$MC_AUXFU_PREDEF_TYPE[<n>]
Adresserweiterung	MD22060 \$MC_AUXFU_PREDEF_EXTENSION[<n>]
Wert	MD22070 \$MC_AUXFU_PREDEF_VALUE[<n>]
Gruppe	MD22040 \$MC_AUXFU_PREDEF_GROUP[<n>]

Vordefinierte Hilfsfunktionen

Allgemeine Hilfsfunktionen, Teil 1					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Halt	0	M	0	0	1
bedingter Halt	1	M	0	1	1
Unterprogramm Ende	2	M	0	2	1
	3	M	0	17	1
	4	M	0	30	1
Werkzeugwechsel	5	M	(0)	6 ¹⁾	(1)

Spindel-spezifische Hilfsfunktionen, Spindel 1					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	6	M	1	3	(2)
Spindel links	7	M	1	4	(2)
Spindel halt	8	M	1	5	(2)
Spindel positionieren	9	M	1	19	(2)
Achsbetrieb	10	M	1	70 ²⁾	(2)
automatische Getriebestufe	11	M	1	40	(4)
Getriebestufe 1	12	M	1	41	(4)
Getriebestufe 2	13	M	1	42	(4)
Getriebestufe 3	14	M	1	43	(4)
Getriebestufe 4	15	M	1	44	(4)
Getriebestufe 5	16	M	1	45	(4)
Spindel-Drehzahl	17	S	1	-1	(3)

Allgemeine Hilfsfunktionen, Teil 2					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Vorschub	18	F	0	-1	(1)
Schneidenwahl	19	D	0	-1	(1)
DL	20	L	0	-1	(1)
Werkzeugwahl	21	T	(0)	-1	(1)
Halt (assoziiert)	22	M	0	-1 ³⁾	1
bedingter Halt (assoziiert)	23	M	0	-1 ⁴⁾	1
Unterprogramm Ende	24	M	0	-1 ⁵⁾	1
Nibbeln	25	M	0	20 ⁶⁾	(10)
Nibbeln	26	M	0	23 ⁶⁾	(10)
Nibbeln	27	M	0	22 ⁶⁾	(11)
Nibbeln	28	M	0	25 ⁶⁾	(11)
Nibbeln	29	M	0	26 ⁶⁾	(12)
Nibbeln	30	M	0	122 ⁶⁾	(11)
Nibbeln	31	M	0	125 ⁶⁾	(11)
Nibbeln	32	M	0	27 ⁶⁾	(12)

Spindel-spezifische Hilfsfunktionen, Spindel 2					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	33	M	2	3	(72)
Spindel links	34	M	2	4	(72)
Spindel halt	35	M	2	5	(72)
Spindel positionieren	36	M	2	19	(72)
Achsbetrieb	37	M	2	70 ²⁾	(72)
automatische Getriebestufe	38	M	2	40	(74)
Getriebestufe 1	39	M	2	41	(74)
Getriebestufe 2	40	M	2	42	(74)
Getriebestufe 3	41	M	2	43	(74)
Getriebestufe 4	42	M	2	44	(74)
Getriebestufe 5	43	M	2	45	(74)
Spindel-Drehzahl	44	S	2	-1	(73)

Spindel-spezifische Hilfsfunktionen, Spindel 3					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	45	M	3	3	(75)
Spindel links	46	M	3	4	(75)
Spindel halt	47	M	3	5	(75)
Spindel positionieren	48	M	3	19	(75)
Achsbetrieb	49	M	3	70 ²⁾	(75)

Spindel-spezifische Hilfsfunktionen, Spindel 3					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
automatische Getriebestufe	50	M	3	40	(77)
Getriebestufe 1	51	M	3	41	(77)
Getriebestufe 2	52	M	3	42	(77)
Getriebestufe 3	53	M	3	43	(77)
Getriebestufe 4	54	M	3	44	(77)
Getriebestufe 5	55	M	3	45	(77)
Spindel-Drehzahl	56	S	3	-1	(76)

Spindel-spezifische Hilfsfunktionen, Spindel 4					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	57	M	4	3	(78)
Spindel links	58	M	4	4	(78)
Spindel halt	59	M	4	5	(78)
Spindel positionieren	60	M	4	19	(78)
Achsbetrieb	61	M	4	70 ²⁾	(78)
automatische Getriebestufe	62	M	4	40	(80)
Getriebestufe 1	63	M	4	41	(80)
Getriebestufe 2	64	M	4	42	(80)
Getriebestufe 3	65	M	4	43	(80)
Getriebestufe 4	66	M	4	44	(80)
Getriebestufe 5	67	M	4	45	(80)
Spindel-Drehzahl	68	S	4	-1	(79)

Spindel-spezifische Hilfsfunktionen, Spindel 5					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	69	M	5	3	(81)
Spindel links	70	M	5	4	(81)
Spindel halt	71	M	5	5	(81)
Spindel positionieren	72	M	5	19	(81)
Achsbetrieb	73	M	5	70 ²⁾	(81)
automatische Getriebestufe	74	M	5	40	(83)
Getriebestufe 1	75	M	5	41	(83)
Getriebestufe 2	76	M	5	42	(83)
Getriebestufe 3	77	M	5	43	(83)
Getriebestufe 4	78	M	5	44	(83)
Getriebestufe 5	79	M	5	45	(83)
Spindel-Drehzahl	80	S	5	-1	(82)

Spindel-spezifische Hilfsfunktionen, Spindel 6					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	81	M	6	3	(84)
Spindel links	82	M	6	4	(84)
Spindel halt	83	M	6	5	(84)
Spindel positionieren	84	M	6	19	(84)
Achsbetrieb	85	M	6	70 ²⁾	(84)
automatische Getriebestufe	86	M	6	40	(86)
Getriebestufe 1	87	M	6	41	(86)
Getriebestufe 2	88	M	6	42	(86)
Getriebestufe 3	89	M	6	43	(86)
Getriebestufe 4	90	M	6	44	(86)
Getriebestufe 5	91	M	6	45	(86)
Spindel-Drehzahl	92	S	6	-1	(85)

Spindel-spezifische Hilfsfunktionen, Spindel 7					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	93	M	7	3	(87)
Spindel links	94	M	7	4	(87)
Spindel halt	95	M	7	5	(87)
Spindel positionieren	96	M	7	19	(87)
Achsbetrieb	97	M	7	70 ²⁾	(87)
automatische Getriebestufe	98	M	7	40	(89)
Getriebestufe 1	99	M	7	41	(89)
Getriebestufe 2	100	M	7	42	(89)
Getriebestufe 3	101	M	7	43	(89)
Getriebestufe 4	102	M	7	44	(89)
Getriebestufe 5	103	M	7	45	(89)
Spindel-Drehzahl	104	S	7	-1	(88)

Spindel-spezifische Hilfsfunktionen, Spindel 8					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	105	M	8	3	(90)
Spindel links	106	M	8	4	(90)
Spindel halt	107	M	8	5	(90)
Spindel positionieren	108	M	8	19	(90)
Achsbetrieb	109	M	8	70 ²⁾	(90)
automatische Getriebestufe	110	M	8	40	(92)
Getriebestufe 1	111	M	8	41	(92)
Getriebestufe 2	112	M	8	42	(92)

Spindel-spezifische Hilfsfunktionen, Spindel 8					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Getriebestufe 3	113	M	8	43	(92)
Getriebestufe 4	114	M	8	44	(92)
Getriebestufe 5	115	M	8	45	(92)
Spindel-Drehzahl	116	S	8	-1	(91)

Spindel-spezifische Hilfsfunktionen, Spindel 9					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	117	M	9	3	(93)
Spindel links	118	M	9	4	(93)
Spindel halt	119	M	9	5	(93)
Spindel positionieren	120	M	9	19	(93)
Achsbetrieb	121	M	9	70 ²⁾	(93)
automatische Getriebestufe	122	M	9	40	(95)
Getriebestufe 1	123	M	9	41	(95)
Getriebestufe 2	124	M	9	42	(95)
Getriebestufe 3	125	M	9	43	(95)
Getriebestufe 4	126	M	9	44	(95)
Getriebestufe 5	127	M	9	45	(95)
Spindel-Drehzahl	128	S	9	-1	(94)

Spindel-spezifische Hilfsfunktionen, Spindel 10					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	129	M	10	3	(96)
Spindel links	130	M	10	4	(96)
Spindel halt	131	M	10	5	(96)
Spindel positionieren	132	M	10	19	(96)
Achsbetrieb	133	M	10	70 ²⁾	(96)
automatische Getriebestufe	134	M	10	40	(98)
Getriebestufe 1	135	M	10	41	(98)
Getriebestufe 2	136	M	10	42	(98)
Getriebestufe 3	137	M	10	43	(98)
Getriebestufe 4	138	M	10	44	(98)
Getriebestufe 5	139	M	10	45	(98)
Spindel-Drehzahl	140	S	10	-1	(97)

Spindel-spezifische Hilfsfunktionen, Spindel 11					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	141	M	11	3	(99)
Spindel links	142	M	11	4	(99)
Spindel halt	143	M	11	5	(99)
Spindel positionieren	144	M	11	19	(99)
Achsbetrieb	145	M	11	70 ²⁾	(99)
automatische Getriebestufe	146	M	11	40	(101)
Getriebestufe 1	147	M	11	41	(101)
Getriebestufe 2	148	M	11	42	(101)
Getriebestufe 3	149	M	11	43	(101)
Getriebestufe 4	150	M	11	44	(101)
Getriebestufe 5	151	M	11	45	(101)
Spindel-Drehzahl	152	S	11	-1	(100)

Spindel-spezifische Hilfsfunktionen, Spindel 12					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	153	M	11	3	(102)
Spindel links	154	M	12	4	(102)
Spindel halt	155	M	12	5	(102)
Spindel positionieren	156	M	12	19	(102)
Achsbetrieb	157	M	12	70 ²⁾	(102)
automatische Getriebestufe	158	M	12	40	(104)
Getriebestufe 1	159	M	12	41	(104)
Getriebestufe 2	160	M	12	42	(104)
Getriebestufe 3	161	M	12	43	(104)
Getriebestufe 4	162	M	12	44	(104)
Getriebestufe 5	163	M	12	45	(104)
Spindel-Drehzahl	164	S	12	-1	(103)

Spindel-spezifische Hilfsfunktionen, Spindel 13					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	165	M	13	3	(105)
Spindel links	166	M	13	4	(105)
Spindel halt	167	M	13	5	(105)
Spindel positionieren	168	M	13	19	(105)
Achsbetrieb	169	M	13	70 ²⁾	(105)
automatische Getriebestufe	170	M	13	40	(107)
Getriebestufe 1	171	M	13	41	(107)
Getriebestufe 2	172	M	13	42	(107)

Spindel-spezifische Hilfsfunktionen, Spindel 13					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Getriebestufe 3	173	M	13	43	(107)
Getriebestufe 4	174	M	13	44	(107)
Getriebestufe 5	175	M	13	45	(107)
Spindel-Drehzahl	176	S	13	-1	(106)

Spindel-spezifische Hilfsfunktionen, Spindel 14					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	177	M	14	3	(108)
Spindel links	178	M	14	4	(108)
Spindel halt	179	M	14	5	(108)
Spindel positionieren	180	M	14	19	(108)
Achsbetrieb	181	M	14	70 ²⁾	(108)
automatische Getriebestufe	182	M	14	40	(110)
Getriebestufe 1	183	M	14	41	(110)
Getriebestufe 2	184	M	14	42	(110)
Getriebestufe 3	185	M	14	43	(110)
Getriebestufe 4	186	M	14	44	(110)
Getriebestufe 5	187	M	14	45	(110)
Spindel-Drehzahl	188	S	14	-1	(109)

Spindel-spezifische Hilfsfunktionen, Spindel 15					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	189	M	15	3	(111)
Spindel links	190	M	15	4	(111)
Spindel halt	191	M	15	5	(111)
Spindel positionieren	192	M	15	19	(111)
Achsbetrieb	193	M	15	70 ²⁾	(111)
automatische Getriebestufe	194	M	15	40	(113)
Getriebestufe 1	195	M	15	41	(113)
Getriebestufe 2	196	M	15	42	(113)
Getriebestufe 3	197	M	15	43	(113)
Getriebestufe 4	198	M	15	44	(113)
Getriebestufe 5	199	M	15	45	(113)
Spindel-Drehzahl	200	S	15	-1	(112)

Spindel-spezifische Hilfsfunktionen, Spindel 16					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	201	M	16	3	(114)
Spindel links	202	M	16	4	(114)
Spindel halt	203	M	16	5	(114)
Spindel positionieren	204	M	16	19	(114)
Achsbetrieb	205	M	16	70 ²⁾	(114)
automatische Getriebestufe	206	M	16	40	(116)
Getriebestufe 1	207	M	16	41	(116)
Getriebestufe 2	208	M	16	42	(116)
Getriebestufe 3	209	M	16	43	(116)
Getriebestufe 4	210	M	16	44	(116)
Getriebestufe 5	211	M	16	45	(116)
Spindel-Drehzahl	212	S	16	-1	(115)

Spindel-spezifische Hilfsfunktionen, Spindel 17					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	213	M	17	3	(117)
Spindel links	214	M	17	4	(117)
Spindel halt	215	M	17	5	(117)
Spindel positionieren	216	M	17	19	(117)
Achsbetrieb	217	M	17	70 ²⁾	(117)
automatische Getriebestufe	218	M	17	40	(119)
Getriebestufe 1	219	M	17	41	(119)
Getriebestufe 2	220	M	17	42	(119)
Getriebestufe 3	221	M	17	43	(119)
Getriebestufe 4	222	M	17	44	(119)
Getriebestufe 5	223	M	17	45	(119)
Spindel-Drehzahl	224	S	17	-1	(118)

Spindel-spezifische Hilfsfunktionen, Spindel 18					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	225	M	18	3	(120)
Spindel links	226	M	18	4	(120)
Spindel halt	227	M	18	5	(120)
Spindel positionieren	228	M	18	19	(120)
Achsbetrieb	229	M	18	70 ²⁾	(120)
automatische Getriebestufe	230	M	18	40	(122)
Getriebestufe 1	231	M	18	41	(122)
Getriebestufe 2	232	M	18	42	(122)

Spindel-spezifische Hilfsfunktionen, Spindel 18					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Getriebestufe 3	233	M	18	43	(122)
Getriebestufe 4	234	M	18	44	(122)
Getriebestufe 5	235	M	18	45	(122)
Spindel-Drehzahl	236	S	18	-1	(121)

Spindel-spezifische Hilfsfunktionen, Spindel 19					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	237	M	19	3	(123)
Spindel links	238	M	19	4	(123)
Spindel halt	239	M	19	5	(123)
Spindel positionieren	240	M	19	19	(123)
Achsbetrieb	241	M	19	70 ²⁾	(123)
automatische Getriebestufe	242	M	19	40	(125)
Getriebestufe 1	243	M	19	41	(125)
Getriebestufe 2	244	M	19	42	(125)
Getriebestufe 3	245	M	19	43	(125)
Getriebestufe 4	246	M	19	44	(125)
Getriebestufe 5	247	M	19	45	(125)
Spindel-Drehzahl	248	S	19	-1	(124)

Spindel-spezifische Hilfsfunktionen, Spindel 20					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Spindel rechts	249	M	20	3	(126)
Spindel links	250	M	20	4	(126)
Spindel halt	251	M	20	5	(126)
Spindel positionieren	252	M	20	19	(126)
Achsbetrieb	253	M	20	70 ²⁾	(126)
automatische Getriebestufe	254	M	20	40	(128)
Getriebestufe 1	255	M	20	41	(128)
Getriebestufe 2	256	M	20	42	(128)
Getriebestufe 3	257	M	20	43	(128)
Getriebestufe 4	258	M	20	44	(128)
Getriebestufe 5	259	M	20	45	(128)
Spindel-Drehzahl	260	S	20	-1	(127)

Werkzeughalter-spezifische Hilfsfunktionen, T-Hilfsfunktionen					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Werkzeuganwahl	261	T	1	-1	129
Werkzeuganwahl	262	T	2	-1	130
Werkzeuganwahl	263	T	3	-1	131
Werkzeuganwahl	264	T	4	-1	132
Werkzeuganwahl	265	T	5	-1	133
Werkzeuganwahl	266	T	6	-1	134
Werkzeuganwahl	267	T	7	-1	135
Werkzeuganwahl	268	T	8	-1	136
Werkzeuganwahl	269	T	9	-1	137
Werkzeuganwahl	270	T	10	-1	138
Werkzeuganwahl	271	T	11	-1	139
Werkzeuganwahl	272	T	12	-1	140
Werkzeuganwahl	273	T	13	-1	141
Werkzeuganwahl	274	T	14	-1	142
Werkzeuganwahl	275	T	15	-1	143
Werkzeuganwahl	276	T	16	-1	144
Werkzeuganwahl	277	T	17	-1	145
Werkzeuganwahl	278	T	18	-1	146
Werkzeuganwahl	279	T	19	-1	147
Werkzeuganwahl	280	T	20	-1	148

Werkzeughalter-spezifische Hilfsfunktionen, M6-Hilfsfunktionen					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Werkzeugwechsel	281	M	1	6 ¹⁾	149
Werkzeugwechsel	282	M	2	6 ¹⁾	150
Werkzeugwechsel	283	M	3	6 ¹⁾	151
Werkzeugwechsel	284	M	4	6 ¹⁾	152
Werkzeugwechsel	285	M	5	6 ¹⁾	153
Werkzeugwechsel	286	M	6	6 ¹⁾	154
Werkzeugwechsel	287	M	7	6 ¹⁾	155
Werkzeugwechsel	288	M	8	6 ¹⁾	156
Werkzeugwechsel	289	M	9	6 ¹⁾	157
Werkzeugwechsel	290	M	10	6 ¹⁾	158
Werkzeugwechsel	291	M	11	6 ¹⁾	159
Werkzeugwechsel	292	M	12	6 ¹⁾	160
Werkzeugwechsel	293	M	13	6 ¹⁾	161
Werkzeugwechsel	294	M	14	6 ¹⁾	162
Werkzeugwechsel	295	M	15	6 ¹⁾	163
Werkzeugwechsel	296	M	16	6 ¹⁾	164

Werkzeughalter-spezifische Hilfsfunktionen, M6-Hilfsfunktionen					
Systemfunktion	Index <n>	Typ	Adresserw.	Wert	Gruppe
Werkzeugwechsel	297	M	17	6 ¹⁾	165
Werkzeugwechsel	298	M	18	6 ¹⁾	166
Werkzeugwechsel	299	M	19	6 ¹⁾	167
Werkzeugwechsel	300	M	20	6 ¹⁾	168

Legende:

() Der Wert kann geändert werden.

- 1) Der Wert ist abhängig von Maschinendatum:
MD22560 \$MC_TOOL_CHANGE_M_MODE (M-Funktion für Werkzeugwechsel)
- 2) Der Wert lässt sich über folgende Maschinendaten mit einem anderen Wert vorbesetzen:
MD20095 \$MC_EXTERN_RIGID_TAPPING_M_NR (M-Funktion für das Umschalten in den gesteuerten Achsbetrieb (Ext. mode))
MD20094 \$MC_SPIND_RIGID_TAPPING_M_NR (M-Funktion für das Umschalten in den gesteuerten Achsbetrieb)
Hinweis
An die PLC wird immer der Wert 70 ausgegeben.
- 3) Der Wert wird eingestellt über Maschinendatum:
MD22254 \$MC_AUXFU_ASSOC_M0_VALUE (Zusätzliche M-Fkt. für Programm-Halt)
- 4) Der Wert wird eingestellt über Maschinendatum:
MD22256 \$MC_AUXFU_ASSOC_M1_VALUE (Zusätzliche M-Fkt. für bedingten Halt)
- 5) Der Wert wird eingestellt über Maschinendatum:
MD10714 \$MN_M_NO_FCT_EOP (M-Funktion für Spindel aktiv nach Reset)
- 6) Der Wert wird eingestellt über Maschinendatum:
MD26008 \$MC_NIBBLE_PUNCH_CODE (Festlegung der M-Funktionen)

12.2.2 Übersicht: Ausgabeverhalten

Bedeutung der in der nachfolgenden Tabelle aufgeführten Parameter:

Parameter	Bedeutung
Index <n>	Maschinendatenindex der Parameter einer Hilfsfunktion
Ausgabeverhalten	MD22080 \$MC_AUXFU_PREDEF_SPEC[<n>], Bit 0 ... 18 Bit 19 ... 31: reserviert

Ausgabeverhalten der vordefinierten Hilfsfunktionen

Systemfunktion	Index <n>																			
	Ausgabeverhalten, Bit																			
18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Halt	0	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
bedingter Halt	1	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
Unterprogramm Ende	2	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
	3	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
	4	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
Werkzeugwechsel	5	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
Spindel rechts	6	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
Spindel links	7	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
Spindel halt	8	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
Spindel positionieren	9	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
Achsbetrieb	10	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
automatische Getriebe- stufe	11	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Getriebestufe 1	12	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Getriebestufe 2	13	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Getriebestufe 3	14	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Getriebestufe 4	15	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Getriebestufe 5	16	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
Spindel-Drehzahl	17	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(1)	(0)	(0)	(0)	0	(0)	(1)
Vorschub	18	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(1)	(0)	0	(1)	0	(0)	(1)
Schneidenwahl	19	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(0)	(1)	0	(0)	0	(0)	(1)
DL	20	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(0)	(1)	0	(0)	0	(0)	(1)
Werkzeugwahl	21	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(0)	(1)	0	(0)	0	(0)	(1)
Halt (assoziiert)	22	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
bedingter Halt (assozii- ert)	23	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
Unterprogramm Ende	24	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
Nibbeln	25	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	26	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	27	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	28	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	29	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	30	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	31	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
Nibbeln	32	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)

() Der Wert kann geändert werden.

Bedeutung der Bits

Bit	Bedeutung
0	<p>Quittierung "normal" nach einen OB1-Takt</p> <p>Eine Hilfsfunktion mit normaler Quittierung wird zu Beginn des OB1-Zyklus in die NC/PLC-Nahtstelle ausgegeben. Über das hilfsfunktionsspezifische Änderungssignal wird dem PLC-Anwenderprogramm angezeigt, dass die Hilfsfunktion gültig ist.</p> <p>Die Quittierung der Hilfsfunktion erfolgt, sobald der Organisationsbaustein OB1 einmal komplett durchlaufen wurde. Dies entspricht einem vollständigen PLC-Anwenderzyklus.</p> <p>Die Ausgabe einer Hilfsfunktion mit normaler Quittierung erfolgt synchron zum Teileprogrammsatz, in dem sie programmiert ist. Ist die Abarbeitung des Teileprogrammsatzes, z. B. Bahn- und/oder Positionierachsbewegungen, vor der Quittierung der Hilfsfunktion beendet, wird der Satzwechsel so lange verzögert, bis die Quittierung durch die PLC erfolgt.</p> <p>Bei Bahnsteuerbetrieb kann eine konstante Bahngeschwindigkeit im Zusammenhang mit Hilfsfunktion mit normaler Quittierung nur dann eingehalten werden, wenn die Hilfsfunktionsausgabe während der Verfahrbewegung erfolgt und vor dem Erreichen des Satzendes durch die PLC quittiert wurde.</p>
1	<p>Quittierung "quick" mit OB40</p> <p>Eine Hilfsfunktion mit schneller Quittierung wird vor dem nächsten OB1-Zyklus in die NC/PLC-Nahtstelle ausgegeben. Über das hilfsfunktionsspezifische Änderungssignal wird dem PLC-Anwenderprogramm angezeigt, dass die Hilfsfunktion gültig ist.</p> <p>Die Quittierung der Hilfsfunktion erfolgt sofort durch das PLC-Grundprogramm im nächsten OB40-Takt. Die Quittierung der Hilfsfunktion ist daher keine Bestätigung für die Ausführung der entsprechenden PLC-Anwenderfunktion. Die Bearbeitung der Hilfsfunktion erfolgt weiterhin im OB1-Zyklus. Die nächste Ausgabe von Hilfsfunktionen an die PLC kann daher erst nach dem vollständigen Durchlauf dieses OB1-Zyklus erfolgen. Dies macht sich bei Bahnsteuerbetrieb vor allem dann bemerkbar (Absenken der Bahngeschwindigkeit), wenn in mehreren aufeinander folgenden Teileprogrammsätzen Hilfsfunktionen schneller Quittierung ausgegeben werden.</p> <p>Bei Hilfsfunktionen mit schneller Quittung ist nicht gewährleistet, dass die Reaktion im PLC-Anwenderprogramm satzsynchron erfolgt.</p> <p>Hinweis Die Parametrierung des Ausgabeverhaltens von Hilfsfunktionen als "Schnelle Hilfsfunktionen" ist nur in Verbindung mit anwenderdefinierten Hilfsfunktionen möglich.</p>
2	<p>Keine vordefinierte Hilfsfunktion</p> <p>Mit dieser Einstellung wird eine vordefinierte Hilfsfunktion wie eine anwenderdefinierte Hilfsfunktion behandelt. Die Hilfsfunktion löst dann nicht mehr die entsprechende Systemfunktion aus, sondern wird nur noch an die PLC ausgegeben.</p> <p>Beispiel: Umprojektierung der Hilfsfunktion "Spindel positionieren" (Index 9) zu einer anwenderdefinierten Hilfsfunktion mit normaler Quittierung und Ausgabe vor der Verfahrbewegung. MD22080 \$MC_AUXFU_PREDEF_SPEC [9] = 'H25' (100101B)</p>
3	<p>Keine Ausgabe an die PLC</p> <p>Die Hilfsfunktion wird nicht an die PLC ausgegeben.</p>
4	<p>Spindelreaktion nach der Quittung durch die PLC</p> <p>Die zugehörige Systemfunktion wird erst nach der Quittierung durch die PLC ausgeführt.</p>
5	<p>Ausgabe vor der Bewegung</p> <p>Die Ausgabe der Hilfsfunktion an die PLC erfolgt vor den im Teileprogrammsatz programmierten Verfahrbewegungen (Bahn- und/oder satzbezogene Positionierachsbewegungen).</p>
6	<p>Ausgabe während der Bewegung</p> <p>Die Ausgabe der Hilfsfunktion an die PLC erfolgt während der im Teileprogrammsatz programmierten Verfahrbewegungen (Bahn- und/oder satzbezogene Positionierachsbewegungen).</p>

Bit	Bedeutung
7	Ausgabe am Satzende
	Die Ausgabe der Hilfsfunktion an die PLC erfolgt nach Abschluss der im Teileprogrammsatz programmierten Verfahrbewegungen (Bahn- und/oder satzbezogene Positionierachsbewegungen).
8	Keine Ausgabe nach Satzsuchlauf Type 1, 2, 4
	Satzsuchlauf Typ 1, 2, 4: Die während des Satzsuchlaufs aufgesammelte Hilfsfunktion wird nicht ausgegeben.
9	Aufsammlung während Satzsuchlauf mit Programmtest (Type 5, SERUPRO)
	Die Hilfsfunktion wird bei Satzsuchlauf mit Programmtest gruppenspezifisch in folgenden Systemvariablen aufgesammelt: <ul style="list-style-type: none"> • \$AC_AUXFU_M_VALUE[<n>] • \$AC_AUXFU_M_EXT[<n>] • \$AC_AUXFU_M_STATE[<n>]
10	Keine Ausgabe während Satzsuchlauf mit Programmtest (Type 5, SERUPRO)
	Die Hilfsfunktion wird bei Satzsuchlauf mit Programmtest nicht an die PLC ausgegeben.
11	Kanalübergreifende Hilfsfunktion (SERUPRO)
	Die Hilfsfunktion wird bei Satzsuchlauf mit Programmtest (SERUPRO) kanalübergreifend in der globalen Liste der Hilfsfunktionen aufgesammelt. Hinweis Pro Hilfsfunktionsgruppe wird immer nur die letzte Hilfsfunktion der Gruppe aufgesammelte.
12	Ausgabe erfolgte über Synchronaktion (nur lesbar)
	Das Bit wird gesetzt wenn die Hilfsfunktion über eine Synchronaktion an die PLC ausgegeben wurde.
13	Implizite Hilfsfunktion(nur lesbar)
	Das Bit wird gesetzt wenn die Hilfsfunktion implizit an die PLC ausgegeben wurde.
14	Aktives M01(nur lesbar)
	Das Bit wird gesetzt wenn die Hilfsfunktion bei aktivem M01 an die PLC ausgegeben wurde.
15	Keine Ausgabe während Einfahr-Testlauf
	Die Hilfsfunktion wird während des Einfahr-Testlaufs nicht an die PLC ausgegeben.
16	Nibbeln aus
17	Nibbeln ein
18	Nibbeln

Hinweis

Bei Hilfsfunktionen, für die kein Ausgabeverhalten festgelegt ist, wird folgendes Standard-Ausgabeverhalten wirksam:

- Bit 0 = 1: Ausgabedauer einen OB1-Takt
 - Bit 7 = 1: Ausgabe am Satzende
-

12.2.3 Parametrierung

12.2.3.1 Gruppenzuordnung

Über die Gruppenzuordnung einer Hilfsfunktion wird die Behandlung der Hilfsfunktion bei Satzsuchlauf festgelegt. Die 168 zur Verfügung stehenden Hilfsfunktionsgruppen sind in vordefinierte und anwenderdefinierbare Gruppen aufgeteilt:

vordefinierte Gruppen:	1 ... 4	10 ... 12	72 ... 168
anwenderdefinierbare Gruppen:	5 ... 9	13 ... 71	

Jede vordefinierte Hilfsfunktion ist standardmäßig einer Hilfsfunktionsgruppe zugeordnet. Diese Zuordnung kann für die meisten vordefinierte Hilfsfunktion über folgendes Maschinendatum geändert werden:

MD22040 \$MC_AUXFU_PREDEF_GROUP[<n>] (Gruppenzuordnung von vordefinierten Hilfsfunktionen)

Soll eine Hilfsfunktion keiner Gruppe zugeordnet sein, ist im Maschinendatum der Wert "0" einzutragen.

Für die vordefinierten Hilfsfunktionen mit folgenden Indizes <n> kann die Gruppenzuordnung nicht geändert werden: 0, 1, 2, 3, 4, 22, 23, 24

Hinweis

1. Hilfsfunktionsgruppe und Satzsuchlauf

Hilfsfunktionen der 1. Hilfsfunktionsgruppe werden bei Satzsuchlauf nur aufgesammelt, aber nicht ausgegeben.

12.2.3.2 Typ, Adresserweiterung und Wert

Über die Parameter Typ, Adresserweiterung und Wert erfolgt die Programmierung einer Hilfsfunktion (siehe Kapitel "Programmierung einer Hilfsfunktion (Seite 683)").

Typ

Über den "Typ" wird der Bezeichner einer Hilfsfunktion festgelegt, z. B.:

"M"	für Zusatzfunktion
"S"	für Spindelfunktion
"F"	für Vorschub

Die Einstellung erfolgt über das Maschinendatum:

MD22050 \$MC_AUXFU_PREDEF_TYPE[<n>] (Typ von vordefinierten Hilfsfunktionen)

Hinweis

Bei vordefinierten Hilfsfunktionen kann der "Typ" nicht verändert werden.

Adresserweiterung

Die "Adresserweiterung" einer Hilfsfunktion dient zur Adressierung unterschiedlicher Komponenten des gleichen Typs. Bei vordefinierten Hilfsfunktionen entspricht der Wert der "Adresserweiterung" der Spindelnummer, auf die sich die Hilfsfunktion bezieht.

Die Einstellung erfolgt über das Maschinendatum:

MD22060 \$MC_AUXFU_PREDEF_EXTENSION[<n>] (Adresserweiterung für vordefinierte Hilfsfunktionen)

Zusammenfassen von Hilfsfunktionen

Soll eine Hilfsfunktion für alle Spindeln eines Kanals der gleichen Hilfsfunktionsgruppe zugeordnet werden, ist für den Parameter "Adresserweiterung" ein Wert von "-1" einzugeben.

Beispiel:

Für alle Spindeln des Kanals wird die Hilfsfunktion M3 (Maschinendatenindex = 6) der 2. Hilfsfunktionsgruppe zugeordnet.

MD22040 \$MC_AUXFU_PREDEF_GROUP[6] = 2
MD22050 \$MC_AUXFU_PREDEF_TYPE[6] = "M"
MD22060 \$MC_AUXFU_PREDEF_EXTENSION[6] = -1
MD22070 \$MC_AUXFU_PREDEF_VALUE[6] = 3

Wert

Die Parameter "Wert" und "Typ" definieren die Bedeutung einer Hilfsfunktion. D. h. die Systemfunktion, die aufgrund dieser Hilfsfunktion aktiviert wird.

Der "Wert" einer Hilfsfunktion ist festgelegt im Maschinendatum:

MD22070 \$MC_AUXFU_PREDEF_VALUE[<n>] (Wert von vordefinierten Hilfsfunktionen)

Hinweis

Bei vordefinierter Hilfsfunktion kann der "Wert" nicht verändert werden. Für einige vordefinierte Hilfsfunktionen kann der "Wert" über zusätzliche Maschinendaten umprojektiert werden (siehe Kapitel "Assoziierte Hilfsfunktionen (Seite 678)").

12.2.3.3 Ausgabeverhalten

Über den Parameter "Ausgabeverhalten" wird festgelegt, wann eine vordefinierte Hilfsfunktion an die NC/PLC-Nahtstelle ausgegeben und wann sie von der PLC quittiert wird.

Die Einstellung erfolgt über das Maschinendatum:

MD22080 \$MC_AUXFU_PREDEF_SPEC[<n>] (Ausgabeverhalten von vordefinierten Hilfsfunktionen)

Ausgabeverhalten bezüglich der Bewegung

Ausgabe vor der Bewegung

- Die Verfahrbewegungen (Bahn- und/oder satzbezogene Positionierachsbewegungen) des vorausgehenden Teileprogrammsatzes werden mit Genauhalt beendet.
- Die Ausgabe der Hilfsfunktionen erfolgt mit Beginn des aktuellen Teileprogrammsatzes.
- Die Verfahrbewegungen des aktuellen Teileprogrammsatzes (Bahn- und/oder Positionierachsbewegungen) werden erst nach Quittierung der Hilfsfunktionen durch die PLC gestartet:
 - Ausgabedauer einen OB1-Zyklus (Normale Quittierung): nach einem OB1-Zyklus
 - Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung): nach einem OB40-Zyklus

Ausgabe während der Bewegung

- Die Ausgabe der Hilfsfunktionen erfolgt mit dem Start der Verfahrbewegungen (Bahn- und/oder Positionierachsbewegungen).
- Die Bahngeschwindigkeit des aktuellen Teileprogrammsatzes wird so reduziert, dass die Zeit bis zum Satzende größer als die Zeit zum Quittieren der Hilfsfunktionen durch die PLC ist:
 - Ausgabedauer einen OB1-Zyklus (Normale Quittierung): ein OB1-Zyklus
 - Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung): ein OB40-Zyklus

Ausgabe nach der Bewegung

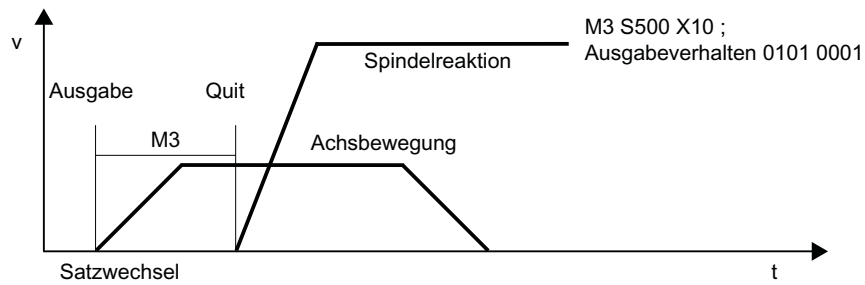
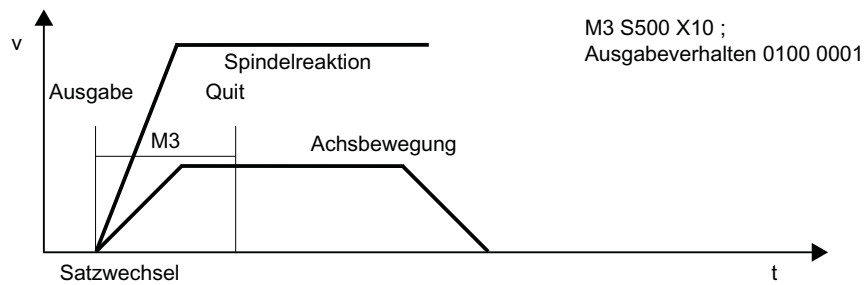
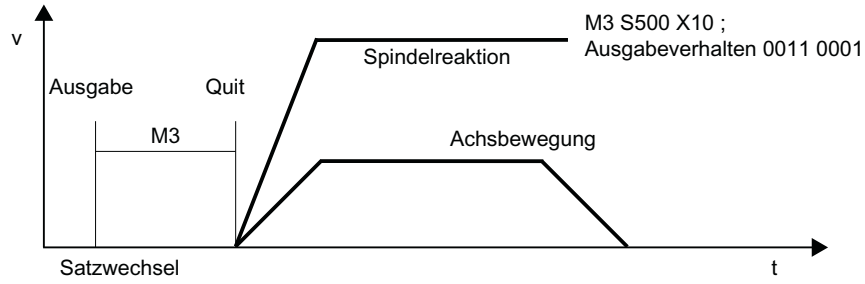
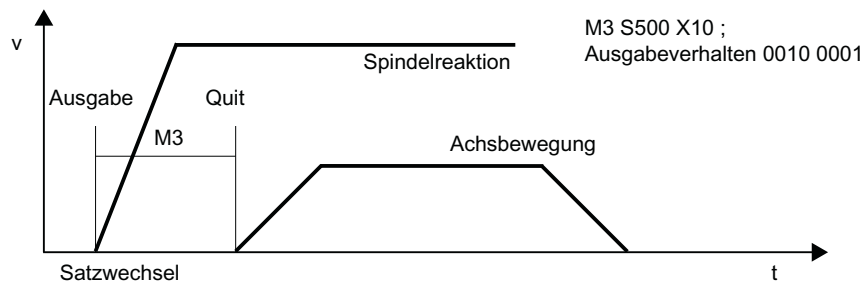
- Die Verfahrbewegungen (Bahn- und/oder satzbezogene Positionierachsbewegungen) des aktuellen Teileprogrammsatzes werden mit Genauhalt beendet.
- Die Ausgabe der Hilfsfunktionen erfolgt nach dem Beenden der Verfahrbewegungen.
- Der Satzwechsel erfolgt nach Quittierung der Hilfsfunktionen durch die PLC:
 - Ausgabedauer einen OB1-Zyklus (Normale Quittierung): nach einem OB1-Zyklus
 - Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung): nach einem OB40-Zyklus

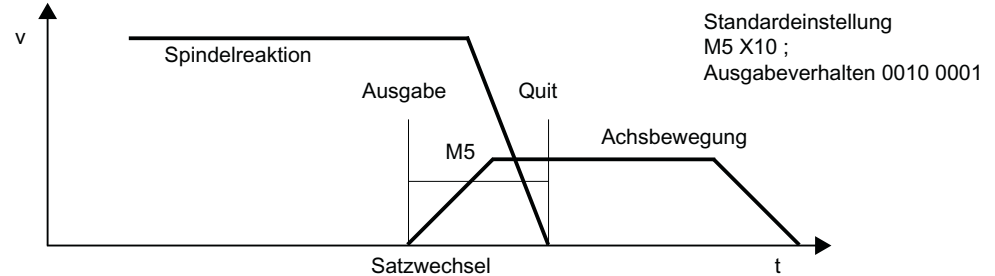
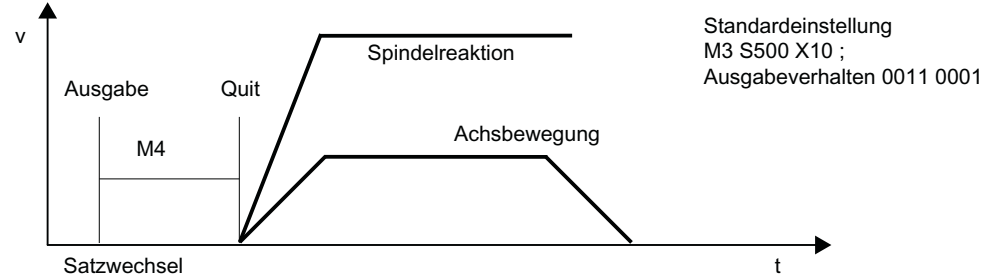
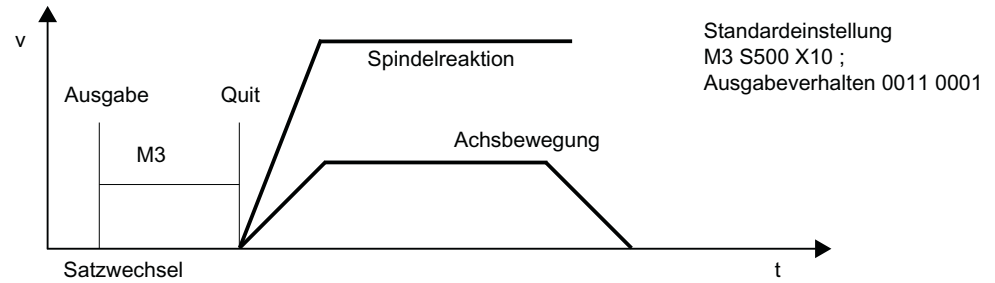
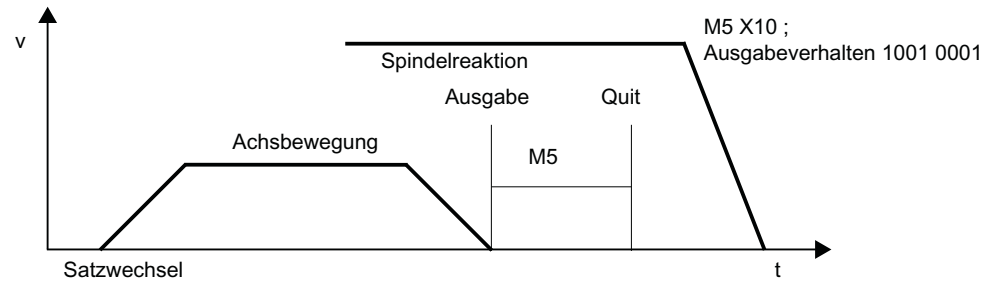
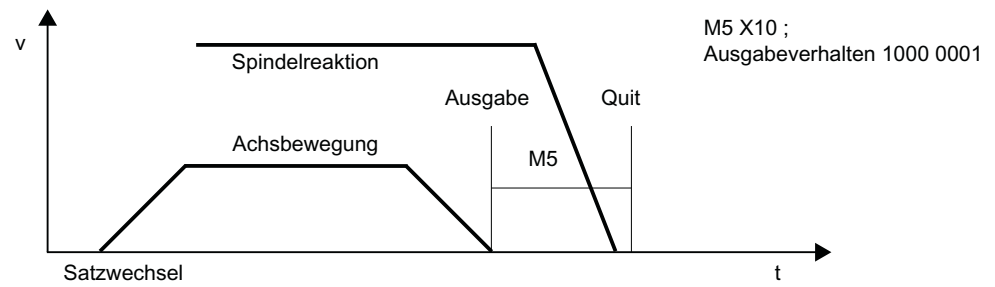
Beispiele für unterschiedliches Ausgabeverhalten

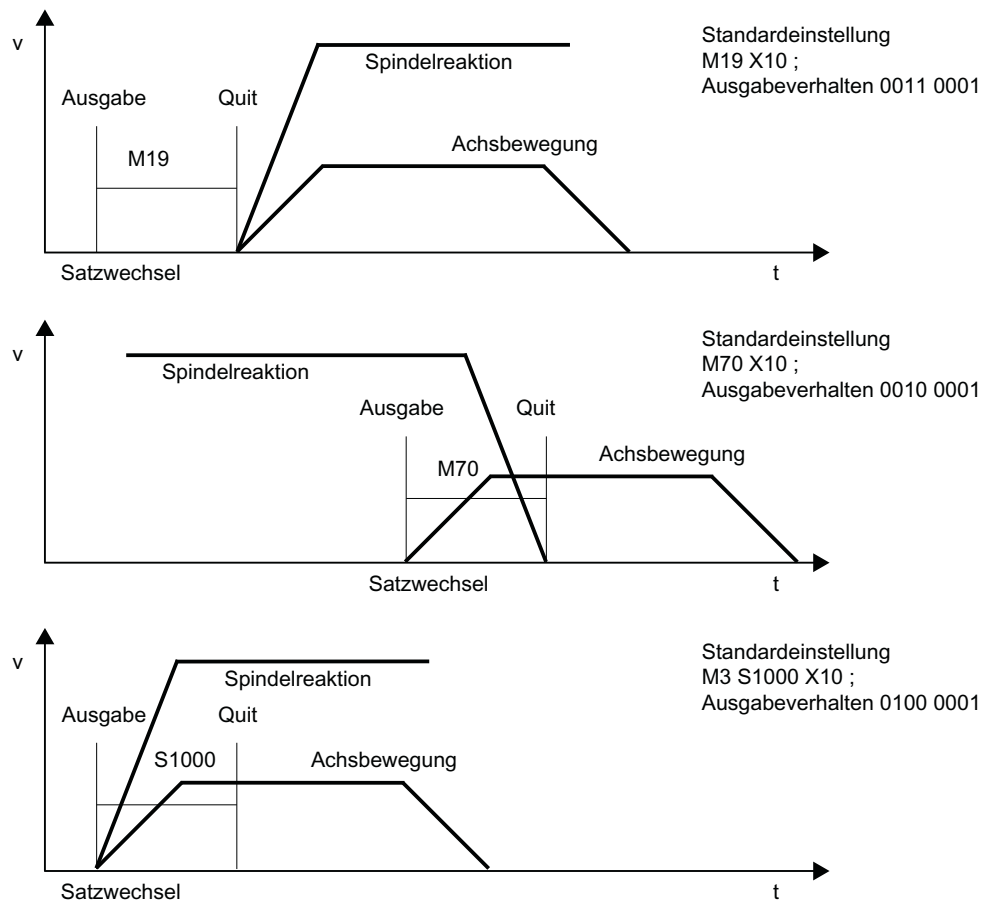
Die folgenden Abbildungen veranschaulichen das unterschiedliche Verhalten bezüglich:

- Ausgabe und Quittierung der Hilfsfunktion
- Spindelreaktion (Drehzahländerung)
- Verfahrbewegung (Geschwindigkeitsänderung)

Die in den Abbildungen unter "Ausgabeverhalten" angegebenen binären Werte beziehen sich auf das parametrisierte Ausgabeverhalten (MD22080).







12.3 Anwenderdefinierte Hilfsfunktionen

Die Verwendung von anwenderdefinierten Hilfsfunktionen lässt sich in zwei Bereiche unterteilen:

- Erweiterung von vordefinierten Hilfsfunktionen
- Anwenderspezifische Hilfsfunktionen

Erweiterung von vordefinierten Hilfsfunktionen

Da die Maschinendaten der vordefinierten Hilfsfunktionen nur einmal vorhanden sind, kann darüber immer nur eine Spindel des Kanals adressiert werden. Zur Adressierung weiterer Spindeln müssen anwenderdefinierte Hilfsfunktionen zur Erweiterung der vordefinierten Hilfsfunktionen parametrisiert werden.

Die Erweiterung von vordefinierten Hilfsfunktionen bezieht sich ausschließlich auf den Parameter "Adresserweiterung". In den Parameter "Adresserweiterung" wird die Nummer der Spindel eingetragen, auf die sich die Hilfsfunktion bezieht.

Für folgende Systemfunktionen können die entsprechenden vordefinierten Hilfsfunktionen erweitert werden:

Systemfunktion	Typ		
		Adresserweiterung ¹⁾	
			Wert
Werkzeugwechsel	M	1	6
Spindel rechts	M	1	3
Spindel links	M	1	4
Spindel halt	M	1	5
Spindel positionieren	M	1	19
Achsbetrieb	M	1	70
automatische Getriebestufe	M	1	40
Getriebestufe 1	M	1	41
Getriebestufe 2	M	1	42
Getriebestufe 3	M	1	43
Getriebestufe 4	M	1	44
Getriebestufe 5	M	1	45
Spindel-Drehzahl	S	1	-1
Werkzeuganwahl	T	1	-1

¹⁾ Adresserweiterung = 1 ist der in den Maschinendaten der vordefinierten Hilfsfunktionen verwendete Standardwert

Beispiel:

Erweiterung der vordefinierten Hilfsfunktion für die Systemfunktion "Spindel rechts" für die 2. und 3. Spindel des Kanals.

Hilfsfunktion "Spindel rechts" für die 2. Spindel des Kanals:

MD22010 \$MC_AUXFU_ASSIGN_TYPE [n] = "M"
MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [n] = 2
 MD22030 \$MC_AUXFU_ASSIGN_VALUE [n] = 3

Hilfsfunktion "Spindel rechts" für die 3. Spindel des Kanals:

MD22010 \$MC_AUXFU_ASSIGN_TYPE [m] = "M"
MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [m] = 3
 MD22030 \$MC_AUXFU_ASSIGN_VALUE [m] = 3

Anwenderspezifische Hilfsfunktionen

Anwenderspezifische Hilfsfunktionen haben folgende Eigenschaften:

- Über anwenderspezifische Hilfsfunktionen werden ausschließlich Anwenderfunktionen aktiviert.
- Über anwenderspezifische Hilfsfunktionen können keine Systemfunktionen aktiviert werden.

- Eine anwenderspezifische Hilfsfunktion wird entsprechend dem parametrisierten Ausgabeverhalten an die PLC ausgegeben.
- Die Funktionalität einer anwenderspezifischen Hilfsfunktion wird durch den Maschinenhersteller/Anwender im PLC-Anwenderprogramm realisiert.

12.3.1 Parametrierung

12.3.1.1 Maximale Anzahl anwenderdefinierter Hilfsfunktionen

Die maximale Anzahl von anwenderdefinierten Hilfsfunktionen pro Kanal ist parametrierbar über das Maschinendatum:

MD11100 \$MN_AUXFU_MAXNUM_GROUP_ASSIGN (Maximale Anzahl von anwenderdefinierten Hilfsfunktionen)

12.3.1.2 Gruppenzuordnung

Über die Gruppenzuordnung einer Hilfsfunktion wird die Behandlung der Hilfsfunktion bei Satzsuchlauf festgelegt. Die 168 zur Verfügung stehenden Hilfsfunktionsgruppen sind in vordefinierte und anwenderdefinierbare Gruppen aufgeteilt:

vordefinierte Gruppen:	1 ... 4	10 ... 12	72 ... 168
anwenderdefinierbare Gruppen:	5 ... 9	13 ... 71	

Jede anwenderdefinierte Hilfsfunktion ist standardmäßig der 1. Hilfsfunktionsgruppe zugeordnet. Die Zuordnung kann über das folgende Maschinendatum geändert werden:

MD22000 \$MC_AUXFU_ASSIGN_GROUP[<n>] (Gruppenzuordnung von anwenderdefinierten Hilfsfunktionen)

Soll eine Hilfsfunktion keiner Gruppe zugeordnet sein, ist im Maschinendatum der Wert "0" einzutragen.

Hinweis

1. Hilfsfunktionsgruppe und Satzsuchlauf

Hilfsfunktionen der 1. Hilfsfunktionsgruppe werden bei Satzsuchlauf nur aufgesammelt, aber nicht ausgegeben.

12.3.1.3 Typ, Adresserweiterung und Wert

Über die Parameter Typ, Adresserweiterung und Wert erfolgt die Programmierung einer Hilfsfunktion (siehe Kapitel "Programmierung einer Hilfsfunktion (Seite 683)").

Typ

Über den "Typ" wird der Bezeichner einer Hilfsfunktion festgelegt.

Bezeichner für anwenderdefinierte Hilfsfunktionen sind:

Typ	Bezeichner	Bedeutung
"H"	Hilfsfunktion	Anwenderspezifische Hilfsfunktionen
"M"	Zusatzfunktion	Erweiterung von vordefinierten Hilfsfunktionen
"S"	Spindelfunktion	
"T"	Werkzeugnummer	

Die Einstellung erfolgt über das Maschinendatum:

MD22010 \$MC_AUXFU_ASSIGN_TYPE[<n>] (Typ von anwenderdefinierten Hilfsfunktionen)

Adresserweiterung

MD22020 \$MC_AUXFU_ASSIGN_EXTENSION[<n>] (Adresserweiterung für anwenderdefinierte Hilfsfunktionen)

Bei anwenderspezifischen Hilfsfunktionen ist die Funktionalität der Adresserweiterung nicht festgelegt. Sie dient allgemein der Unterscheidung von Hilfsfunktionen mit dem gleichen "Wert".

Zusammenfassen von Hilfsfunktionen

Sollen alle Hilfsfunktionen vom gleichen Typ und Wert der gleichen Hilfsfunktionsgruppe zugeordnet werden, ist für den Parameter "Adresserweiterung" ein Wert von "-1" einzugeben.

Beispiel:

Alle anwenderspezifischen Hilfsfunktionen mit Wert = 8 werden der 10. Hilfsfunktionsgruppe zugeordnet.

```
MD22000 $MC_AUXFU_ASSIGN_GROUP [ 1 ]           = 10
MD22010 $MC_AUXFU_ASSIGN_TYPE [ 1 ]           = "H"
MD22020 $MC_AUXFU_ASSIGN_EXTENSION [ 1 ]      = -1
MD22030 $MC_AUXFU_ASSIGN_VALUE [ 1 ]         = 8
```

Wert

MD22030 \$MC_AUXFU_ASSIGN_VALUE[<n>] (Wert von anwenderdefinierten Hilfsfunktionen)

Bei anwenderspezifischen Hilfsfunktionen ist die Funktionalität des Parameters "Wert" nicht festgelegt. Allgemein wird über den Wert die entsprechende PLC-Anwenderfunktion aktiviert.

Zusammenfassen von Hilfsfunktionen

Sollen alle Hilfsfunktionen vom gleichen Typ und Adresserweiterung der gleichen Hilfsfunktionsgruppe zugeordnet werden, ist für den Parameter "Wert" ein Wert von "-1" einzugeben.

Beispiel:

Alle anwenderspezifischen Hilfsfunktionen mit Adresserweiterung = 2 werden der 11. Hilfsfunktionsgruppe zugeordnet.

MD22000 \$MC_AUXFU_ASSIGN_GROUP [2]	= 11
MD22010 \$MC_AUXFU_ASSIGN_TYPE [2]	= "H"
MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [2]	= 2
MD22030 \$MC_AUXFU_ASSIGN_VALUE [2]	= -1

12.3.1.4 Ausgabeverhalten

Das "Ausgabeverhalten" von anwenderdefinierten Hilfsfunktionen ist parametrierbar über das Maschinendatum:

MD22035 \$MC_AUXFU_ASSIGN_SPEC[<n>] (Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen)

Zur Beschreibung der einzelnen Ausgabeparameter siehe Kapitel "Ausgabeverhalten (Seite 670)" der vordefinierten Hilfsfunktionen. Die dort gemachten Aussagen gelten sinngemäß auch für das Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen.

12.4 Assoziierte Hilfsfunktionen

Funktion

Assoziierte Hilfsfunktionen sind anwenderdefinierte Hilfsfunktionen, mit der gleichen Wirkung wie die entsprechenden vordefinierten Hilfsfunktionen. Für folgende vordefinierten Hilfsfunktionen können anwenderdefinierte Hilfsfunktionen assoziiert werden:

- M0 (Programmierer Halt)
- M1 (Wahlweiser Halt)

G-Gruppe

Einer assoziierten Hilfsfunktion wird die G-Gruppe der entsprechenden vordefinierten Hilfsfunktion zugeordnet.

Parametrierung

Die Assoziierung einer anwenderdefinierten Hilfsfunktion zu einer der oben genannten vordefinierten Hilfsfunktionen erfolgt in den Maschinendaten:

- MD22254 \$MC_AUXFU_ASSOC_M0_VALUE (Assoziierte M-Funktion für "Programmierer Halt")
- MD22256 \$MC_AUXFU_ASSOC_M1_VALUE (Assoziierte M-Funktion für "Wahlweiser Halt")

Anwahl

Die Auswahl von "Assoziierte Hilfsfunktion" (M-1) erfolgt über die Bedienoberfläche SINUMERIK Operate im Bedienbereich "Automatik" > "Programmbeeinflussung" durch Setzen des HMI/PLC-Nahtstellensignals DB21, ... DBX24.4.

Das Nahtstellensignal wird, abhängig vom Wert des FB1-Parameters `MMCToIF`, vom PLC-Grundprogramm in das NC/PLC-Nahtstellensignal DB21, ... DBX30.5 übertragen:

- "TRUE": Übertragung
- "FALSE": Keine Übertragung

Standardmäßig ist der Wert des Parameters "TRUE".

Hinweis

Anwahlmöglichkeit über Bedienoberfläche SINUMERIK Operate

Die Auswahl einer assoziierten Hilfsfunktion ist im Bedienbereich "Automatik" > "Programmbeeinflussung" nur sichtbar, wenn sie im Maschinendatum parametrierbar ist.

Anwendung

Assoziierte Hilfsfunktionen dürfen angewendet werden in:

- Hauptprogramm
 - Unterprogramm
 - Zyklus
-

Hinweis

Assoziierte Hilfsfunktionen dürfen in Synchronaktionen **nicht** angewendet werden.

NC/PLC-Nahtstellensignale

Bei einer assoziierten anwenderdefinierten Hilfsfunktion werden an die NC/PLC-Nahtstelle dieselben Signale ausgegeben wie bei der entsprechenden vordefinierten Hilfsfunktion. Zur Unterscheidung, welche Hilfsfunktion tatsächlich programmiert wurde, wird aber als Wert der Hilfsfunktion der Wert der anwenderdefinierten Hilfsfunktionen (Parameter "Wert") ausgegeben. Im PLC-Anwenderprogramm besteht damit die Möglichkeit, zwischen vordefinierter und anwenderdefinierter Hilfsfunktion zu unterscheiden.

NC/PLC-Nahtstellensignale für assoziierte Hilfsfunktionen:

- DB21, ... DBX24.4 (Assoziierte Hilfsfunktion ausgewählt)
- DB21, ... DBX30.5 (Assoziierte Hilfsfunktion aktivieren)
- DB21, ... DBX318.5 (Assoziierte Hilfsfunktion aktiv)

Randbedingungen

Folgenden Randbedingungen sind zu beachten:

- Eine anwenderdefinierte Hilfsfunktion darf nicht mehrfach assoziiert werden.
- Vordefinierte Hilfsfunktionen (z. B. M3, M4, M5 etc.) dürfen nicht assoziiert werden.

Beispiele

1. Assoziieren der anwenderdefinierten Hilfsfunktion M111 zu M0:
MD22254 \$MC_AUXFU_ASSOC_M0_VALUE = 111
Die anwenderdefinierte Hilfsfunktion M111 hat damit die gleiche Funktionalität wie M0.
2. Assoziieren der anwenderdefinierten Hilfsfunktion M222 zu M1:
MD22256 \$MC_AUXFU_ASSOC_M1_VALUE = 222
Die anwenderdefinierte Hilfsfunktion M222 hat damit die gleiche Funktionalität wie M1.

12.5 Typ-spezifisches Ausgabeverhalten

Funktion

Das Ausgabeverhalten der Hilfsfunktionen bezüglich einer im Teileprogrammsatz programmierten Verfahrbewegung kann Typ-spezifisch festgelegt werden.

Parametrierung

Die Parametrierung des Typ-spezifischen Ausgabeverhaltens erfolgt über die Maschinendaten:

MD22200 \$MC_AUXFU_M_SYNC_TYPE (Ausgabezeitpunkt M-Funktionen)

MD22210 \$MC_AUXFU_S_SYNC_TYPE (Ausgabezeitpunkt S-Funktionen)

MD22220 \$MC_AUXFU_T_SYNC_TYPE (Ausgabezeitpunkt T-Funktionen)

MD22230 \$MC_AUXFU_H_SYNC_TYPE (Ausgabezeitpunkt H-Funktionen)

MD22240 \$MC_AUXFU_F_SYNC_TYPE (Ausgabezeitpunkt F-Funktionen)

MD22250 \$MC_AUXFU_D_SYNC_TYPE (Ausgabezeitpunkt D-Funktionen)

MD22252 \$MC_AUXFU_DL_SYNC_TYPE (Ausgabezeitpunkt DL-Funktionen)

Folgende Ausgabeverhalten können parametriert werden:

MD \$MC_AUXFU_xx_SYNC_TYPE = <Wert>

Wert	Ausgabeverhalten
0	Ausgabe vor der Bewegung
1	Ausgabe während der Bewegung
2	Ausgabe am Satzende
3	Keine Ausgabe an PLC
4	Ausgabe entsprechend des mit MD22080 festgelegten Ausgabeverhaltens

Zur Beschreibung der verschiedenen Ausgabeverhalten siehe Kapitel "Ausgabeverhalten (Seite 670)".

Hinweis

Die für den jeweiligen Hilfsfunktionstyp einstellbaren Ausgabeverhalten sind dem Listenhandbuch "Ausführliche Maschinendaten-Beschreibung" zu entnehmen.

Beispiel

Ausgabe von Hilfsfunktionen mit unterschiedlichem Ausgabeverhalten in einem Teileprogrammsatz mit Verfahrbewegung.

Parametriertes Ausgabeverhalten:

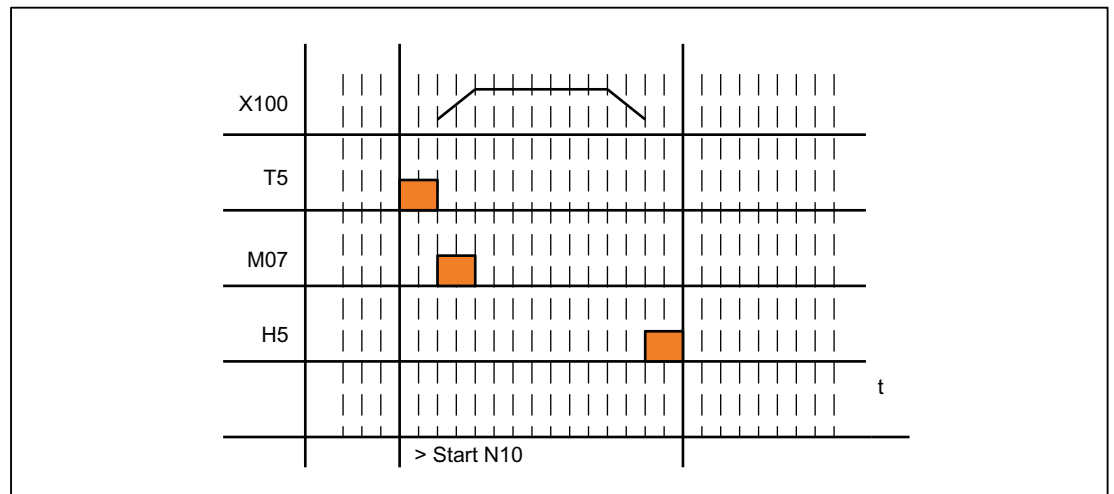
- MD22200 \$MC_AUXFU_M_SYNC_TYPE = 1 ⇒ M-Funktion:
Ausgabe **während** der Bewegung
- MD22220 \$MC_AUXFU_T_SYNC_TYPE = 0 ⇒ T-Funktion:
Ausgabe **vor** der Bewegung
- MD22230 \$MC_AUXFU_H_SYNC_TYPE = 2 ⇒ H-Funktion:
Ausgabe **am Satzende**

Teileprogrammsatz:

```

Programmcode
...
N10 G01 X100 M07 H5 T5
...
    
```

Zeitlicher Ablauf der Hilfsfunktionsausgabe:



12.6 Prioritäten des parametrisierten Ausgabeverhaltens

Die Prioritäten bezüglich des parametrisierten Ausgabeverhaltens einer Hilfsfunktion müssen für folgende Kriterien getrennt beachtet werden:

- Ausgabedauer (normale / schnelle Quittierung)
- Ausgabe bezüglich der Bewegung (vor / während / nach der Bewegung)

Allgemein gilt, dass das parametrisierte Ausgabeverhalten mit der niedrigeren Priorität immer dann wirksam wird, wenn kein höher priorisiertes Ausgabeverhalten parametrisiert wurde.

Ausgabedauer

Für die Ausgabedauer gelten folgende Prioritäten:

Priorität	Ausgabeverhalten	Festlegung über:
Höchste	Hilfsfunktionsspezifisch	Teileprogrammanweisung: QU(...) (siehe Kapitel "Programmierbare Ausgabedauer (Seite 684)")
↓	Hilfsfunktionsspezifisch	MD22035 \$MC_AUXFU_ASSIGN_SYNC[<n> MD22080 \$MC_AUXFU_PREDEF_SYNC[<n>
↓	Gruppenspezifisch	MD11110 \$MC_AUXFU_GROUP_SPEC[<n>
Niedrigste	Keine Festlegung	Standard-Ausgabeverhalten: Ausgabedauer einen OB1-Takt

Ausgabe bezüglich der Bewegung

Für die Ausgabe bezüglich der Bewegung gelten folgende Prioritäten:

Priorität	Ausgabeverhalten	Festlegung über:
Höchste	Hilfsfunktionsspezifisch	MD22035 \$MC_AUXFU_ASSIGN_SYNC[<n> MD22080 \$MC_AUXFU_PREDEF_SYNC[<n>
↓	Gruppenspezifisch	MD11110 \$MC_AUXFU_GROUP_SPEC[<n>
↓	Typspezifisch	MD22200 \$MC_AUXFU_M_SYNC_TYPE MD22210 \$MC_AUXFU_S_SYNC_TYPE MD22220 \$MC_AUXFU_T_SYNC_TYPE MD22230 \$MC_AUXFU_H_SYNC_TYPE MD22240 \$MC_AUXFU_F_SYNC_TYPE MD22250 \$MC_AUXFU_D_SYNC_TYPE MD22252 \$MC_AUXFU_DL_SYNC_TYPE
Niedrigste	Keine Festlegung	Standard-Ausgabeverhalten: Ausgabe am Satzende

Hinweis

Teileprogrammsätze ohne Bahnbewegung

In einem Teileprogrammsatz ohne Bahnbewegung (also auch bei Positionierachsen und Spindeln) werden die Hilfsfunktionen sofort in einem Block ausgegeben.

12.7 Programmierung einer Hilfsfunktion

Syntax

Die Programmierung einer Hilfsfunktion erfolgt in einem Teileprogrammsatz mit folgender Syntax:
<Typ> [<Adresserweiterung>=] <Wert>

Hinweis

Wird keine Adresserweiterung programmiert, wird implizit die Adresserweiterung = 0 gesetzt.
Vordefinierte Hilfsfunktionen mit der Adresserweiterung = 0 beziehen sich immer auf die Masterspindel des Kanals.

Symbolische Adressierung

Die Werte für die Parameter "Adresserweiterung" und "Wert" können auch symbolisch angegeben werden. Der symbolische Name für die Adresserweiterung muss dann in eckigen Klammern angegeben werden.

Beispiel:

Symbolische Programmierung der Hilfsfunktion M3 (Spindel rechts) für die 1. Spindel:

Programmcode	Kommentar
DEF SPINDEL_NR=1	; 1.Spindel im Kanal
DEF DREHRICHTUNG=3	; Drehrichtung rechts
N100 M[SPINDEL_NR]=DREHRICHTUNG	; entsprechend: M1=3

Hinweis

Bei der Verwendung von symbolischen Namen zur Programmierung einer Hilfsfunktion wird bei der Ausgabe der Hilfsfunktion an die PLC nicht der symbolische Name übergeben, sondern der entsprechende Zahlenwert.

Beispiele

Beispiel 1: Programmierung vordefinierter Hilfsfunktionen

Programmcode	Kommentar
N10 M3	; "Spindel rechts" für die Master-Spindel des Kanals.
N20 M0=3	; "Spindel rechts" für die Master-Spindel des Kanals.
N30 M1=3	; "Spindel rechts" für die 1.Spindel des Kanals.
N40 M2=3	; "Spindel rechts" für die 2.Spindel des Kanals.

Beispiel 2: Programmierbeispiele von Hilfsfunktionen mit den entsprechenden Werten zur Ausgabe an die PLC

Programmcode	Kommentar
DEF Kühlmittel=12	; Ausgabe an die PLC: - - -
DEF Schmiermittel=130	; Ausgabe an die PLC: - - -
H[Kühlmittel]=Schmiermittel	; Ausgabe an die PLC: H12=130
H=Kühlmittel	; Ausgabe an die PLC: H0=12
H5	; Ausgabe an die PLC: H0=5
H=5.379	; Ausgabe an die PLC: H0=5.379
H17=3.5	; Ausgabe an die PLC: H17=3.5
H[Kühlmittel]=13.8	; Ausgabe an die PLC: H12=13.8
H='HFF13'	; Ausgabe an die PLC: H0=65299
H='B1110'	; Ausgabe an die PLC: H0=14
H5.3=21	; Fehler

12.8 Programmierbare Ausgabedauer

Funktion

Anwenderspezifischen Hilfsfunktionen, für die das Ausgabeverhalten "Ausgabedauer einen OB1-Zyklus (Langsame Quittierung)" parametrisiert wurde, können für einzelne Ausgaben über die Teileprogrammanweisung QU (Quick) zu Hilfsfunktionen mit schneller Quittierung definiert werden.

Syntax

Die Definition einer Hilfsfunktion mit schneller Quittierung erfolgt in einem Teileprogrammsatz mit folgender Syntax:

<Typ> [<Adresserweiterung>]=QU (<Wert>)

Beispiel

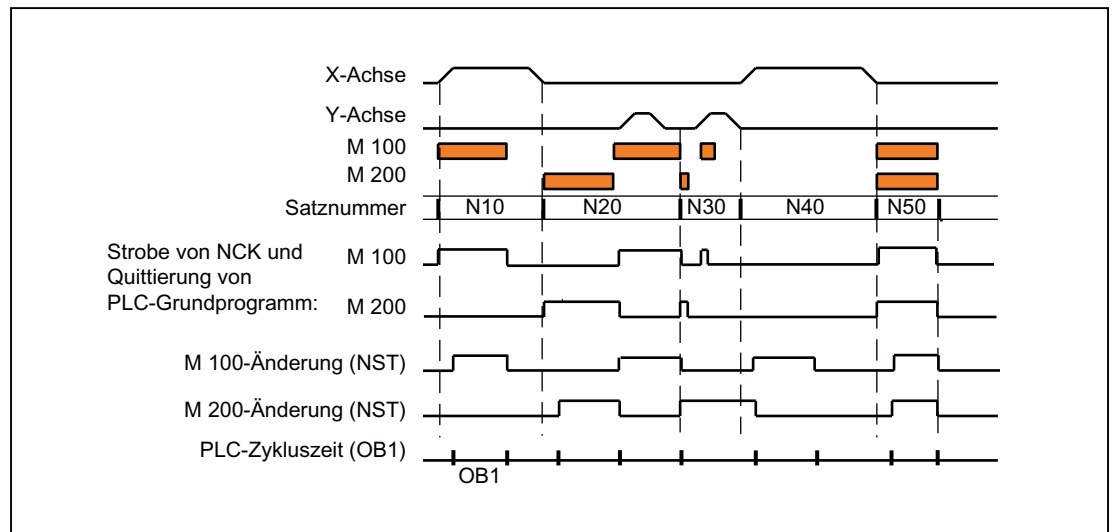
Unterschiedliches Verhalten bei der Ausgabe der Hilfsfunktionen M100 und M200 in einem Teileprogramm. Das Ausgabeverhalten der Hilfsfunktionen ist folgendermaßen parametrisiert:

- M100
 - Ausgabedauer einen OB1-Zyklus (Langsame Quittierung)
 - Ausgabe während der Bewegung
- M200
 - Ausgabedauer einen OB1-Zyklus (Langsame Quittierung)
 - Ausgabe vor der Bewegung

Programmcode	Kommentar
N10 G94 G01 X50 M100	; Ausgabe von M100: während der Bewegung ; Quittierung: langsam
N20 Y5 M100 M200	; Ausgabe von M200: vor der Bewegung ; Ausgabe von M100: während der Bewegung ; Quittierung: langsam
N30 Y0 M=QU(100) M=QU(200)	; Ausgabe von M200: vor der Bewegung ; Ausgabe von M100: während der Bewegung ; Quittierung: schnell
N40 X0	
N50 M100 M200	; Ausgabe von M200: sofort 1) ; Ausgabe von M100: sofort 1) ; Quittierung: langsam
M17	

1) Ohne Verfahrbewegung erfolgt die Ausgabe von Hilfsfunktionen an die PLC immer sofort.

Das folgende Bild zeigt den zeitlichen Ablauf des Teileprogramms. Besonders zu beachten ist der zeitliche Unterschied bei der Abarbeitung der Teileprogrammsätze N20 und N30.



12.9 Hilfsfunktionsausgabe an die PLC

Funktion

Bei der Ausgabe einer Hilfsfunktion an die PLC werden folgende Signale und Werte an die NC/PLC-Nahtstelle übergeben:

- Änderungssignale
- Parameter "Adresserweiterung"
- Parameter "Wert"

Datenbereiche in der NC/PLC-Nahtstelle

Die Änderungssignale und Werte der Hilfsfunktionen liegen in der NC/PLC-Nahtstelle in folgenden Datenbereichen:

- Änderungssignale bei Hilfsfunktionsübergabe von NC-Kanal:
DB21, ... DBB58 - DBB67
- Übergebene M-/ und S-Funktionen:
DB21, ... DBB68 - DBB112
- Übergebene T-/ D-/ DL-Funktionen:
DB21, ... DBB116 - DBB136
- Übergebene H-/ F-Funktionen:
DB21, ... DBB140 - DBB190
- Dekodierte M-Signale (M0 - M99):
DB21, ... DBB194 - DBB206 (dynamische M-Funktionen)

Weitere Informationen

- Funktionshandbuch PLC und Grundprogramm, PLC-Grundprogramm
- Funktionshandbuch PLC und Grundprogramm, Referenz zum PLC-Grundprogramm

12.10 Hilfsfunktionen ohne Satzwechselferzögerung

Funktion

Auch bei Hilfsfunktionen mit einem parametrisierten und/oder programmierten Ausgabeverhalten:

- "Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung)"
- "Ausgabe vor der Bewegung" oder "Ausgabe während der Bewegung"

kann es bei Bahnsteuerbetrieb (kurze Verfahwege und hohen Geschwindigkeiten) zu Geschwindigkeitseinbrüchen kommen, weil zum Satzende hin auf die Quittierung der Hilfsfunktion durch die PLC gewartet werden muss. Um derartige Geschwindigkeitseinbrüche

zu vermeiden, kann der Satzwechsel unabhängig bezüglich der Quittierung derartiger Hilfsfunktionen gemacht werden.

Parametrierung

Das Unterdrücken der Satzwechselverzögerung bei schnellen Hilfsfunktionen wird eingestellt über das Maschinendatum:

MD22100 \$MC_AUXFU_QUICK_BLOCKCHANGE (Satzwechselverzögerung bei schnellen Hilfsfunktionen)

Wert	Bedeutung
0	Bei der schnellen Hilfsfunktionsausgabe an die PLC wird der Satzwechsel bis zur Quittierung durch die PLC (OB40) verzögert.
1	Bei der schnellen Hilfsfunktionsausgabe an die PLC wird der Satzwechsel nicht verzögert.

Randbedingungen

Die Synchronität von Hilfsfunktionen, die ohne Satzwechselverzögerung ausgegeben werden, ist bezüglich des Teileprogramm��atzes, in dem sie programmiert sind, nicht mehr gewährleistet. Im "Worst-Case"-Szenario erfolgt die Quittierung einen OB40-Takt und die Bearbeitung der Hilfsfunktion einen OB1-Takt nach dem Satzwechsel zum nächsten Teileprogramm��atz.

12.11 M-Funktion mit implizitem Vorlaufstopp

Funktion

Soll im Zusammenhang mit einer Hilfsfunktion ein Vorlaufstopp ausgelöst werden, kann dieser explizit über den Teileprogramm��atzbefehl `STOPRE` programmiert werden. Soll bei Programmierung der M-Funktion immer ein Vorlaufstopp ausgelöst werden, kann dieses Verhalten M-Funktionsspezifisch über folgendes Maschinendatum parametrierbar werden:

MD10713 \$MN_M_NO_FCT_STOPRE[<n>] (M-Funktion mit Vorlaufstopp)

Beispiel

Die anwenderdefinierte M-Funktion M88 soll einen Vorlaufstopp auslösen.

Parametrierung:

MD10713 \$MN_M_NO_FCT_STOPRE [0] = 88

Anwendung:

Teileprogramm (Auszug)

Programmcode	Kommentar
...	
N100 G0 X10 M88	; Verfahrbewegung und impliziter Vorlaufstopp durch M88.

Programmcode	Kommentar
N110 Y=R1	; N110 wird erst nach Abschluss der Verfahrbewegung und Quittierung der M-Funktion interpretiert.
...	

Randbedingungen

Wird in einem Teileprogramm ein Unterprogramm durch eine der beiden folgenden Möglichkeiten indirekt über eine M-Funktion aufgerufen, erfolgt dabei kein Vorlaufstopp:

- MD10715 \$MN_M_NO_FCT_CYCLE (Durch Unterprogramm zu ersetzende M-Funktion)
- M98 (ISO-Dialekt-T / ISO-Dialekt-M)

12.12 Verhalten bei Überspeichern

Überspeichern

Über die SINUMERIK-Bedienoberfläche können vor dem Start folgender Funktionen:

- NC-START eines Teileprogramms
- NC-START zur Wiederaufnahme eines unterbrochenen Teileprogramms

die Hilfsfunktionen, die mit dem Start ausgegeben werden, durch die Funktion "Überspeichern" verändert werden.

Mögliche Anwendungsfälle sind:

- Ergänzen von Hilfsfunktionen nach Satzsuchlauf
- Herstellen des Ausgangszustandes zum Einfahren eines Teileprogramms

Überspeicherbare Hilfsfunktionstypen

Folgende Hilfsfunktionstypen können überspeichert werden:

- M (Zusatzfunktion)
- S (Spindeldrehzahl)
- T (Werkzeugnummer)
- H (Hilfsfunktion)
- D (Werkzeugkorrekturnummer)
- DL (Summenkorrektur)
- F (Vorschub)

Gültigkeitsdauer

Eine überspeicherte Hilfsfunktion, z. B. M3 (Spindel rechts), ist so lange gültig, bis sie durch eine andere Hilfsfunktion der gleichen Hilfsfunktionsgruppe, durch erneute Überspeicherung oder durch Programmierung in einem Teileprogrammsatz überschrieben wird.

12.13 Verhalten bei Satzsuchlauf

12.13.1 Hilfsfunktionsausgabe bei Satzsuchlauf Typ 1,2 und 4

Ausgabeverhalten

Bei Satzsuchlauf Typ 1, 2 und 4 werden die Hilfsfunktionen gruppenspezifisch aufgesammelt. Die jeweils letzte Hilfsfunktion einer Hilfsfunktionsgruppe wird nach NC-START in einem eigenen Teileprogrammsatz vor dem eigentlichen Wiedereinstiegssatz mit folgendem Ausgabeverhalten ausgegeben:

- Ausgabedauer einen OB1-Zyklus (normale Quittierung)
- Ausgabe vor der Bewegung

Ausgabesteuerung

Ob eine Hilfsfunktion nach Satzsuchlauf an die PLC ausgegeben wird oder nicht, lässt sich projektieren über das Bit 8 der Maschinendaten:

- MD22080 \$MC_AUXFU_PREDEF_SPEC[<n>]
(Ausgabeverhalten von vordefinierten Hilfsfunktionen)
mit <n> = Systemfunktionsindex (0 ... 32)
- MD22035 \$MC_AUXFU_ASSIGN_SPEC[<n>]
(Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen)
mit <n> = Hilfsfunktionsindex (0 ... 254)
- MD11110 \$MN_AUXFU_GROUP_SPEC[<n>]
(Ausgabeverhalten der Hilfsfunktionen einer Gruppe)
mit <n> = Gruppenindex (0 ... 63)

Bit	Wert	Bedeutung
10	0	Ausgabe während Satzsuchlauf Typ 1,2,4
	1	Keine Ausgabe während Satzsuchlauf Typ 1,2,4

Dieses Verhalten hat keinen Einfluss auf die Anzeige oder auf die Variablen \$AC_AUXFU_STATE[<n>], \$AC_AUXFU_VALUE[<n>] und \$AC_AUXFU_EXT[<n>]. Die Hilfsfunktionen gelten nach Satzsuchlauf immer noch als aufgesammelt, obwohl sie nicht an die PLC ausgegeben werden.

Eine nach Satzsuchlauf nicht ausgegebene Hilfsfunktion überschreibt beim Aufsammeln auch eine Hilfsfunktion, deren Bit 8 nicht gesetzt ist.

Der Anwender kann nach Satzsuchlauf die aufgesammelten Hilfsfunktionen abfragen und unter Umständen diese selber nochmals über das Teileprogramm oder über Synchronaktionen ausgeben.

Hinweis

Folgende Hilfsfunktionen werden nicht aufgesammelt:

- Hilfsfunktionen, die keiner Hilfsfunktionsgruppe zugeordnet sind.
 - Hilfsfunktionen, die der 1. Hilfsfunktionsgruppe zugeordnet sind.
-

Überspeichern von Hilfsfunktionen

Nach erfolgtem Satzsuchlauf werden mit dem nächsten NC-START die aufgesammelten Hilfsfunktionen ausgegeben. Besteht die Anforderung, zusätzliche Hilfsfunktionen auszugeben, können diese über die Funktion "Überspeichern" hinzugefügt werden (siehe Kapitel "Verhalten bei Überspeichern (Seite 688)").

Verhalten bezüglich M19 (Spindel positionieren)

Nach Satzvorlauf wird immer die letzte mit M19 programmierte Spindelpositionierung durchgeführt, auch wenn vom Teileprogrammsatz mit M19 bis zum Zielsatz noch andere spindelspezifische Hilfsfunktionen programmiert sind. Das Setzen der erforderlichen Spindelfreigaben muss im PLC-Anwenderprogramm daher abgeleitet werden von den Nahtstellensignalen der Fahrbefehle:

DB31, ... DBX64.6 / 64.7 (Fahrbefehl minus / plus)

Die spindelspezifischen Hilfsfunktionen M3, M4, M5 eignen sich in diesem Fall nicht dazu, da sie unter Umständen erst nach der Spindelpositionierung an die PLC ausgegeben werden.

Für ausführliche Informationen zum Satzsuchlauf siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 29)".

12.13.2 Zuordnung einer Hilfsfunktion zu mehreren Gruppen

Funktion

Über die Gruppenzuordnung (MD22000 \$MC_AUXFU_ASSIGN_GROUP) können anwenderdefinierte Hilfsfunktionen auch mehreren Gruppen zugeordnet werden. Beim Satzsuchlauf werden diese Hilfsfunktionen bezüglich aller projektierten Gruppen aufgesammelt.

Hinweis

Vordefinierte Hilfsfunktionen können nur einer Gruppe zugeordnet werden.

Beispiel

In der DIN sind folgende M-Befehle zur Kühlmittelausgabe vorgesehen:

- M7: Kühlmittel 2 EIN
- M8: Kühlmittel 1 EIN
- M9: Kühlmittel 1 und 2 AUS

Damit beide Kühlmittel auch gemeinsam aktiv sein können:

- Müssen M7 und M8 in zwei getrennten Gruppen aufgesammelt werden (z. B. Gruppe 5 und 6)
- Muss M9 diesen beiden Gruppen zugeordnet sein, z. B.:
 - Gruppe 5: M7, M9
 - Gruppe 6: M8, M9

Parametrierung:

MD11100 \$MN_AUXFU_MAXNUM_GROUP_ASSIGN = 4

MD22000 \$MC_AUXFU_ASSIGN_GROUP [0] = 5

MD22000 \$MC_AUXFU_ASSIGN_GROUP [1] = 5

MD22000 \$MC_AUXFU_ASSIGN_GROUP [2] = 6

MD22000 \$MC_AUXFU_ASSIGN_GROUP [3] = 6

MD22010 \$MC_AUXFU_ASSIGN_TYPE [0] = M

MD22010 \$MC_AUXFU_ASSIGN_TYPE [1] = M

MD22010 \$MC_AUXFU_ASSIGN_TYPE [2] = M

MD22010 \$MC_AUXFU_ASSIGN_TYPE [3] = M

MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [0] = 0

MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [1] = 0

MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [2] = 0

MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [3] = 0

MD22030 \$MC_AUXFU_ASSIGN_VALUE [0] = 7

MD22030 \$MC_AUXFU_ASSIGN_VALUE [1] = 9

MD22030 \$MC_AUXFU_ASSIGN_VALUE [2] = 8

MD22030 \$MC_AUXFU_ASSIGN_VALUE [3] = 9

MD22035 \$MC_AUXFU_ASSIGN_SPEC [0] = 'H121'

MD22035 \$MC_AUXFU_ASSIGN_SPEC [1] = 'H121'

MD22035 \$MC_AUXFU_ASSIGN_SPEC [2] = 'H121'

MD22035 \$MC_AUXFU_ASSIGN_SPEC [3] = 'H121'

Teileprogramm (Ausschnitt):

```

Programmcode
-----
...
N10 ... M8
N20 ... M9
N30 ... M7
...
    
```

Beim Satzsuchlauf wird die Hilfsfunktion M9 bezüglich der Gruppen 5 und 6 aufgesammelt.

Abfrage der aufgesammelten M-Hilfsfunktionen:

M-Funktion der 5. Gruppe: \$AC_AUXFU_M_VALUE [4] = 7

M-Funktion der 6. Gruppe: \$AC_AUXFU_M_VALUE [5] = 9

12.13.3 Zeitstempel der aktiven M-Hilfsfunktion

Bei der Ausgabe aufgesammelter Hilfsfunktionen nach Satzsuchlauf muss die Reihenfolge beim Aufsammeln beachtet werden. Es wird deshalb zu jeder Gruppe ein Zeitstempel mitgeführt, der über die folgende Systemvariable gruppenspezifisch abgefragt werden kann:

\$AC_AUXFU_M_TICK[<n>] (Zeitstempel der aktiven M-Hilfsfunktion)

12.13.4 Ermittlung der Ausgabe-Reihenfolge

Funktion

Um die Ermittlung der Ausgabe-Reihenfolge für M-Hilfsfunktionen für den Programmierer einfach zu machen, wird folgende vordefinierte Prozedur zur Verfügung gestellt:

```

AUXFUMSEQ(VAR INT _NUM_IN, VAR INT _M_IN[], VAR INT _EXT_IN[], VAR
INT _NUM_OUT, VAR INT _M_OUT[], VAR INT _EXT_OUT[])
    
```

Eingangsparameter:

- VAR INT _NUM_IN: Anzahl relevanter M-Befehle
- VAR INT _M_IN[]: Feld von relevanten M-Codes
- VAR INT _EXT_IN[]: Feld von relevanten M-Adresserweiterungen

Ausgangsparameter:

- VAR INT _NUM_OUT: Anzahl ermittelter M-Codes
- VAR INT _M_OUT[]: Feld von ermittelten M-Codes
- VAR INT _EXT_OUT[]: Feld von ermittelten M-Adresserweiterungen

Die Funktion ermittelt für die vorgegebenen M-Codes die Ausgabe-Reihenfolge der gruppenspezifisch aufgesammelten M-Hilfsfunktionen. Die Reihenfolge ergibt sich aus den

Aufsammlzeitpunkten \$AC_AUXFU_M_TICK[<n>] (siehe Kapitel "Zeitstempel der aktiven M-Hilfsfunktion (Seite 692)").

Ein bestimmter M-Code wird immer nur einmal berücksichtigt, auch wenn er mehreren Gruppen angehört. Ist die Anzahl der relevanten M-Befehle kleiner oder gleich 0, so werden alle aufgesammelten M-Codes ausgegeben. Die Anzahl der relevanten M-Befehle ist auf 64 begrenzt.

Beispiel

M-Befehle zur Kühlmittelausgabe:

- M7: Kühlmittel 2 EIN
- M8: Kühlmittel 1 EIN
- M9: Kühlmittel 1 und 2 AUS

Gruppenzuordnung:

- Gruppe 5: M7, M9
- Gruppe 6: M8, M9

Teileprogramm (Ausschnitt):

Programmcode
...
N10 ... M8
N20 ... M9
N30 ... M7
...

Bei Satzsuchlauf werden die Hilfsfunktionen gruppenspezifisch aufgesammelt. Die jeweils letzte Hilfsfunktion einer Hilfsfunktionsgruppe wird nach Satzsuchlauf an die PLC ausgegeben:

- Gruppe 5: M7
- Gruppe 6: M9

Werden diese in der Reihenfolge M7 → M9 ausgegeben, so ist anschließend kein Kühlmittel aktiv. Im Programmablauf wäre jedoch Kühlmittel 2 aktiv. Die korrekte Ausgabe-Reihenfolge für die M-Hilfsfunktionen wird deshalb mit einem ASUP ermittelt, dass die vordefinierte Prozedur AUXFUMSEQ (...) enthält:

Programmcode
DEF INT _I, _M_IN[3], _EXT_IN[3], _NUM_OUT, _M_OUT[2], _EXT_OUT[2]
_M_IN[0]=7 _EXT_IN[0]=0
_M_IN[1]=8 _EXT_IN[1]=0
_M_IN[2]=9 _EXT_IN[2]=0
AUXFUMSEQ(3, _M_IN, _EXT_IN, _NUM_OUT, _M_OUT, _EXT_OUT)
FOR _I = 0 TO _NUM_OUT-1
M[_EXT_OUT[_I]]=_M_OUT[_I]
ENDFOR

12.13.5 Ausgabeunterdrückung von Spindel-spezifischen Hilfsfunktionen

Funktion

In Verbindung mit bestimmten Situationen, z. B. einem Werkzeugwechsel, kann es erforderlich sein, die bei Satzsuchlauf aufgesammelten spindelspezifischen Hilfsfunktionen nicht in den Aktionssätzen, sondern erst zu einem späteren Zeitpunkt, z. B. nach einem Werkzeugwechsel, auszugeben. Die automatische Ausgabe der spindelspezifischen Hilfsfunktionen nach Satzsuchlauf kann dazu unterdrückt werden. Die Ausgabe kann dann zu einem späteren Zeitpunkt manuell durch Überspeichern oder durch ein ASUP erfolgen.

Parametrierung

Die Unterdrückung der automatischen Ausgabe der spindelspezifischen Hilfsfunktionen nach Satzsuchlauf wird eingestellt über das Maschinendatum:

MD11450 \$MN_SEARCH_RUN_MODE (Verhalten nach Satzsuchlauf)

Bit	Wert	Bedeutung
2	0	Die Ausgabe der spindelspezifischen Hilfsfunktionen erfolgt in den Aktionssätzen
	1	Die Ausgabe der Hilfsfunktionen in den Aktionssätzen wird unterdrückt.

Systemvariablen

Die spindelspezifischen Hilfsfunktionen werden bei Satzsuchlauf, unabhängig von der oben genannten Parametrierung, immer in den folgenden Systemvariablen gespeichert:

Systemvariable	Beschreibung
\$P_SEARCH_S [<n>]	Aufgesammelte Spindeldrehzahl
	Wertebereich: 0 ... Smax
\$P_SEARCH_SDIR [<n>]	Aufgesammelte Spindeldrehrichtung
	Wertebereich: 3, 4, 5, -5, -19, 70
\$P_SEARCH_SGEAR [<n>]	Aufgesammelte Spindelgetriebestufe-M-Funktion
	Wertebereich: 40 ... 45
\$P_SEARCH_SPOS [<n>]	Aufgesammelte Spindelposition
	Wertebereich: 0 ... MD30330 \$MA_MODULO_RANGE (Größe des Modulbereichs)
	bzw.
	Aufgesammelter Verfahrensweg
\$P_SEARCH_SPOSMODE [<n>]	Aufgesammelter Positionsanfahrmodus
	Wertebereich: 0 ... 5

Zur späteren Ausgabe der spindelspezifischen Hilfsfunktionen können die Systemvariablen z. B. in einem ASUP gelesen und nach Ausgabe der Aktionssätze ausgegeben werden:

DB21, ... DBX32.6 = 1 (Letzter Aktionssatz aktiv)

Hinweis

Die Inhalte der Systemvariablen \$P_S, \$P_DIR und \$P_SGEAR können nach Satzsuchlauf durch Synchronisationsvorgänge verloren gehen.

Für weiterführende Informationen zu ASUP, Satzsuchlauf und Aktionssätzen siehe Kapitel "K1: BAG, Kanal, Programmbetrieb, Reset-Verhalten (Seite 29)".

Beispiel

Satzsuchlauf auf Kontur mit Unterdrückung der Ausgabe der spindelspezifischen Hilfsfunktionen und Start eines ASUP nach der Ausgabe der Aktionssätze.

Parametrierung: MD11450 \$MN_SEARCH_RUN_MODE, Bit 2 = 1

Nach dem Satzsuchlauf auf N55 wird das ASUP gestartet.

Teileprogramm:

Programmcode	Kommentar
N05 M3 S200	; Spindel 1
N10 G4 F3	
N15 SPOS=111	; Spindel 1 wird im ASUP auf 111 Grad positioniert
N20 M2=4 S2=300	; Spindel 2
N25 G4 F3	
N30 SPOS[2]=IC(77)	; Spindel 2 verfährt inkrementell um 77 Grad
N55 X10 G0	; Zielsatz
N60 G4 F10	
N99 M30	

ASUP:

Programmcode	Kommentar
PROC ASUP_SAVE	
MSG ("Ausgabe der Spindelfunktionen")	
DEF INT SNR=1	
AUSG_SPI:	
M[SNR]=\$P_SEARCH_SGEAR[SNR]	; Getriebestufe ausgeben.
S[SNR]=\$P_SEARCH_S[SNR]	; Drehzahl ausgeben (bei M40 wird passende Getriebestufe ermittelt).
M[SNR]=\$P_SEARCH_SDIR[SNR]	; Drehrichtung, Positionierung oder Achsbetrieb ausgeben.
SNR=SNR+1	; Nächste Spindel.
REPEAT AUSG_SPI P=\$P_NUM_SPINDLES-1	; Für alle Spindeln.
MSG ("")	
REPOSA	
RET	

Erläuterungen zum Beispiel

Ist die Anzahl der Spindeln bekannt, können zur Reduzierung der Programmbearbeitungszeit gleichartige Ausgaben in einem Teileprogrammsatz geschrieben werden.

Die Ausgabe von \$P_SEARCH_SDIR sollte in einem separaten Teileprogrammsatz erfolgen, da die Spindelpositionierung bzw. die Umschaltung in den Achsbetrieb zusammen mit dem Getriebestufenwechsel zu einer Alarmmeldung führen kann.

Wird das gestartete ASUP mit REPOSA abgeschlossen, bleibt Spindel 1 auf der Position 111 Grad stehen, während Spindel 2 auf die Position 77 Grad repositioniert wird.

Ist ein anderes Verhalten erforderlich, muss die Programmsequenz von z. B. "N05 M3 S..." und "N30 SPOS[2]=IC(...)" für Satzsuchlauf besonders behandelt werden.

Ob Satzsuchlauf aktiv ist, kann im ASUP über die Systemvariablen \$P_SEARCH ermittelt werden:

```
$P_SEARCH==1 ; Satzsuchlauf aktiv
```

Bei einer inkrementellen Positionierung nach Drehzahlsteuerbetrieb ist zwar der zu verfahrenende Weg definiert, jedoch ergibt sich die erreichte Endposition in einigen Fällen erst während des Positioniervorganges. Dies ist z. B. beim Positionsabgleich während des Überfahrens der Nullmarke oder beim Einschalten der Lageregelung der Fall. Aus diesem Grund wird als REPOS-Position (REPOSA im ASUP) der programmierte Weg ab der Position Null angenommen.

Randbedingungen

Aufgesammelte S-Werte

Die Bedeutung eines S-Werts im Teileprogramm ist abhängig vom aktuell aktiven Vorschubtyp:

G93, G94, G95, G97, G971 : S-Wert wird als Drehzahl interpretiert

G96, G961 : S-Wert wird als konstante Schnittgeschwindigkeit interpretiert

Wird die Vorschubart (z. B. für einen Werkzeugwechsel) vor der Ausgabe der Systemvariablen \$P_SEARCH_S geändert, muss, um zu vermeiden, dass ein falscher Vorschubtyp zugrunde gelegt wird, die Vorschubart wieder auf die ursprüngliche Einstellung aus dem Zielsatz des Teileprogramms restauriert werden.

Aufgesammelte Drehrichtung

Bei der Ausgabe der Drehrichtung wird die Systemvariable \$P_SEARCH_SDIR zum Zeitpunkt des Satzsuchlauf-Starts mit dem Wert "-5" vorbesetzt. Dieser Wert ist bei der Ausgabe wirkungslos.

Damit wird sichergestellt, dass bei Satzsuchlauf über Bereiche, in denen Spindeln nicht mit einer Drehrichtung, Positionierung oder Achsbetrieb programmiert werden, der letzte Spindelbetriebsmode erhalten bleibt.

Die Programmierung von M19, SPOS und SPOSA wird als "M-19" (interner M19) in der Systemvariablen \$P_SEARCH_SDIR alternativ zu M3, M4, M5 und M70 aufgesammelt.

Bei der Ausgabe von "M-19" werden die Positionierdaten intern aus der Systemvariablen \$P_SEARCH_SPOS und \$P_SEACH_SPOSMODE gelesen. Beide Systemvariable sind auch beschreibbar, um z. B. Korrekturen vornehmen zu können.

Hinweis

Die Werte "-5" und "19" bleiben dem Anwender aufgrund der oben genannten Zuweisungen (z. B. M[<n>] = \$P_SEARCH_SDIR[<n>]) prinzipiell verborgen und müssen nur bei spezieller Auswertung der Systemvariablen im ASUP beachtet werden.

12.13.6 Hilfsfunktionsausgabe bei Satzsuchlauf Typ 5 (SERUPRO)

Ausgabeverhalten

Beim Satzsuchlauf Typ 5 (SERUPRO) kann eine Hilfsfunktion während des Satzsuchlaufs an die PLC ausgegeben und/oder in den folgenden Systemvariablen gruppenspezifisch aufgesammelt werden:

- \$AC_AUXFU_PREDEF_INDEX[<n>] (Index einer vordefinierten Hilfsfunktion)
- \$AC_AUXFU_TYPE[<n>] (Typ der Hilfsfunktion)
- \$AC_AUXFU_STATE[<n>] (Ausgabestatus der Hilfsfunktion)
- \$AC_AUXFU_EXT[<n>] (Adresserweiterung der Hilfsfunktion)
- \$AC_AUXFU_VALUE[<n>] (Wert der Hilfsfunktion)

Zur Beschreibung der Systemvariablen siehe Kapitel "Abfrage von Systemvariablen (Seite 710)".

Ausgabesteuerung

Ob eine Hilfsfunktion während Satzsuchlauf Typ 5 (SERUPRO) an die PLC ausgegeben und/oder in den genannten Systemvariablen gruppenspezifisch aufgesammelt wird, lässt sich projektieren über Bit 9 und 10 der Maschinendaten:

- MD22080 \$MC_AUXFU_PREDEF_SPEC[<n>]
(Ausgabeverhalten von vordefinierten Hilfsfunktionen)
mit <n> = Systemfunktionsindex (0 ... 32)
- MD22035 \$MC_AUXFU_ASSIGN_SPEC[<n>]
(Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen)
mit <n> = Hilfsfunktionsindex (0 ... 254)
- MD11110 \$MN_AUXFU_GROUP_SPEC[<n>]
(Ausgabeverhalten der Hilfsfunktionen einer Gruppe)
mit <n> = Gruppenindex (0 ... 63)

Bit	Wert	Bedeutung
9	0	Keine Aufsammlung während Satzsuchlauf Typ 5 (SERUPRO)
	1	Aufsammlung während Satzsuchlauf Typ 5 (SERUPRO)

Bit	Wert	Bedeutung
10	0	Ausgabe während Satzsuchlauf Typ 5 (SERUPRO)
	1	Keine Ausgabe während Satzsuchlauf Typ 5 (SERUPRO)

Ausgabezähler

Der Anwender kann die aufgesammelten Hilfsfunktionen kanalweise im Satzsuchlauf-ASUP an die PLC ausgeben. Zum Zwecke der serialisierten Ausgabe über mehrere Kanäle werden die drei Ausgabezähler bei jeder Ausgabe einer Hilfsfunktion über alle Kanäle verändert:

Systemvariable	Bedeutung		
\$AC_AUXFU_TICK[<n>,<m>]	Ausgabezähler der aktiven Hilfsfunktion		
	Index	Bedeutung	
	<n>	Gruppenindex (0 ... 63)	
	<m>	Ausgabezähler (0 ... 2)	
		Wert	Bedeutung
		0	Ausgabesequenzzähler (Alle Ausgaben innerhalb eines IPO-Takts)
1		Paket-Zähler innerhalb einer Ausgabesequenz im IPO-Takt	
2	Hilfsfunktionszähler innerhalb eines Pakets		

Erläuterung

- Ein Hilfsfunktionspaket besteht aus max. 10 Hilfsfunktionen.
- Pro Kanal können während SERUPRO zwei Pakete pro IPO-Takt abgearbeitet werden, da Synchronaktionen in diesem Takt bearbeitet werden.
- Innerhalb eines IPO-Takts kann über alle Kanäle eine Ausgabesequenz von bis zu max. 20 Paketen (2 Pakete pro Kanal * 10 Kanäle) abgearbeitet werden.

Über die Kodierung wird angezeigt, wie viele Hilfsfunktionspakete und wieviele darin enthaltene Hilfsfunktionen im gleichen IPO-Takt abgearbeitet wurden:

- Alle Hilfsfunktionen, die in einem IPO-Takt aufgesammelt wurden, haben den gleichen Sequenzzähler.
- Alle Hilfsfunktionen, die in einem Paket (Satz oder Synchronaktion) aufgesammelt wurden, haben den gleichen Paketzähler.

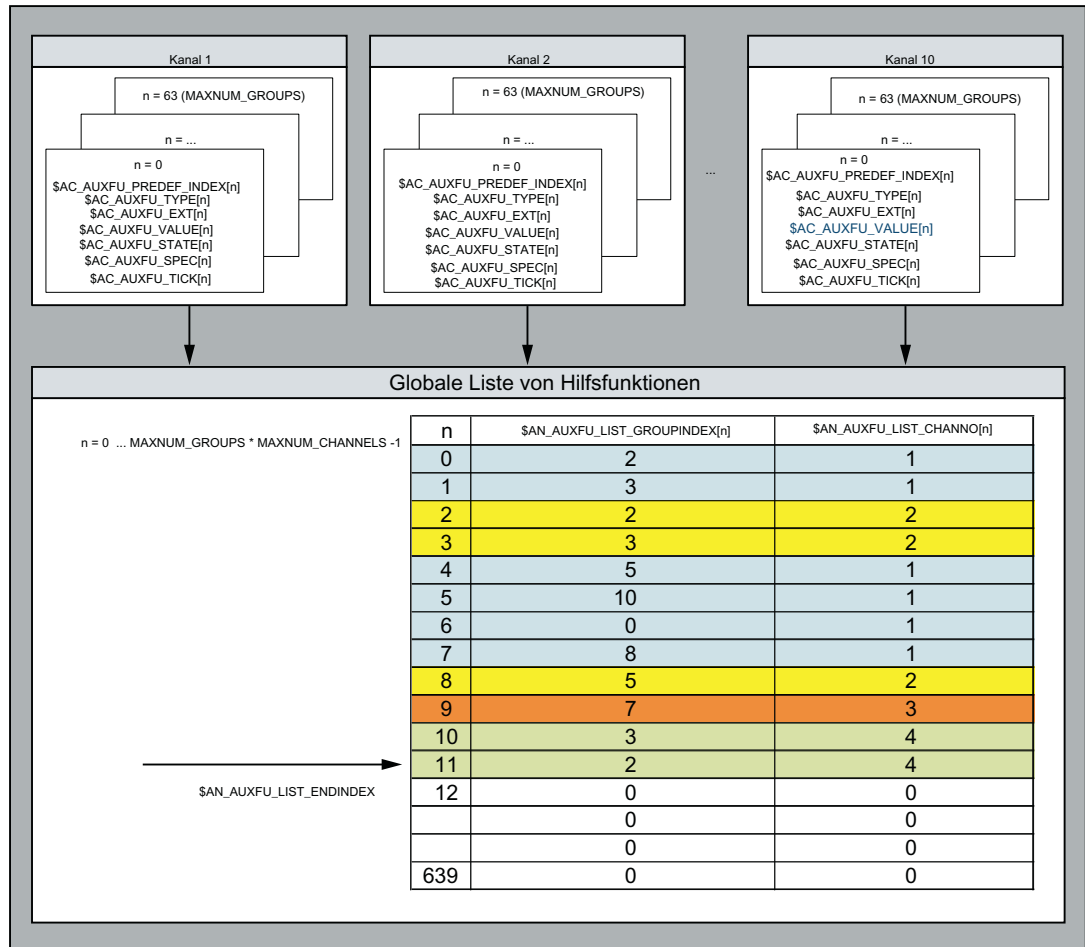
Der Hilfsfunktionszähler wird bei jeder aufgesammelten Hilfsfunktion inkrementiert.

Globale Liste von Hilfsfunktionen

Am Ende von SERUPRO werden die in den einzelnen Kanälen gruppenspezifisch aufgesammelten Hilfsfunktionen entsprechend ihrem Zählerstand in eine kanalübergreifende, globale Liste mit Kanalnummer und Gruppenindex eingetragen.

Systemvariable ¹⁾	Bedeutung
\$AC_AUXFU_TICK[<n>,<m>]	Zählerstand
\$AN_AUXFU_LIST_CHANNO[<n>] ²⁾	Kanalnummer

Systemvariable ¹⁾	Bedeutung
\$AN_AUXFU_LIST_GROUPINDEX[<n>] ²⁾	Gruppenindex
1) Wertebereich Index <n>: 0 ... MAXNUM_GROUPS * MAXNUM_CHANNELS - 1	
2) Die Systemvariablen sind schreib- und lesbar.	



Die globale Liste wird aufgebaut, nachdem das Suchziel gefunden wurde. Sie soll als Systemvorschlag für im nachfolgenden SERUPRO-Ende-ASUP auszugebende Hilfsfunktionen dienen. Soll eine Hilfsfunktion nicht ausgegeben werden, so ist der entsprechende Gruppenindex auf "0" zu setzen.

Verhalten bezüglich Spindel-Hilfsfunktionen

Nach Satzsuchlaufstart sammeln alle Kanäle die Hilfsfunktionen gruppenspezifisch in den Kanal-Variablen auf. Zur weitgehenden Restaurierung des Spindelzustandes im SERUPRO-Zielsatz durch aufgesammelte Hilfsfunktionen ist es notwendig, dass die letzte aktive Hilfsfunktion einer jeden Spindel-Hilfsfunktionsgruppe den Zustand der Spindel im Zielsatz repräsentiert. Bei Spindelzustandsübergängen werden dabei obsolete Hilfsfunktionen gelöscht oder, wenn notwendig, implizite Hilfsfunktionen eingetragen.

Alle Spindel-Hilfsfunktionen der globalen Hilfsfunktionsliste müssen im Zielsatz den erreichten Spindelzuständen entsprechen, damit die Hilfsfunktionen bei Ausgabe der Liste

abarbeitungsfähig sind und keine Alarmer oder ungewollte Spindelzustände angefordert werden, die eine Fortsetzung des Teileprogrammes verhindern können.

Betroffen davon sind die Hilfsfunktionsgruppen einer jeden im System projektierten Spindel, wobei die Spindelnummer der Adresserweiterung einer Hilfsfunktion entspricht.

Gruppe a: M3, M4, M5, M19, M70

Gruppe b: M40, M41, M42, M43, M44, M45

Gruppe c: S

Löschen obsoleter Hilfsfunktionen

Bei folgenden Funktionen werden für die betroffene Spindel die Hilfsfunktionen der Gruppe a gelöscht:

- Für die Folgespindel bei Einschalten einer generischen Kopplung wie COUPON, TRAILON, EGON, ...

Erzeugen von impliziten Hilfsfunktionen der Gruppe a

Bei folgenden Funktionen werden für die betroffene Spindel Hilfsfunktionen der Gruppe a implizit erzeugt:

- Für die Folgespindel beim Ausschalten der Synchronspindelkopplung
 - COUPOF erzeugt im Hauptlauf je nach Kopplungssituation M3, M4 und S oder M5.
 - COUPOF(S<n>, S<m>, POS) und COUPOFS(S<n>, S<m>, POS, POS) erzeugen M3, M4 und S.
 - COUPOFS erzeugt im Hauptlauf M5.
 - COUPOFS(S<n>, S<m>, POS) erzeugt im Hauptlauf M19.
Das implizite M19 (im ASUP "SPOS[<Adresserweiterung>] = IC(0)") schaltet den Positionierbetrieb ein ohne eine Verfahrbewegung.
- Bei einer Verfahrbewegung als Achse oder beim Übergang in den Achsbetrieb durch Anwahl einer Transformation, bei der die Spindel als Achse eingeht, wird M70 erzeugt.
- Bei SPCOF wird M5 erzeugt.

Hinweis

Die Hilfsfunktionen zur Spindelprogrammierung müssen in Zusammenhang mit den Funktionen "Achstausch" und "Achscontainer-Drehung" immer passend zum tatsächlichen Zustand (Motor) bei Tausch/Drehung mitgegeben werden. Dabei unterscheiden sich die Mechanismen von Achstausch und Achscontainer.

Beispiel für Achscontainer-Drehung:

Ein Achscontainer enthält 4 Spindeln. Jede Spindel ist jeweils einem Kanal (1 - 4) zugewiesen. Im Kanal 1 wird immer M3 S1000 programmiert und dann eine Achscontainer-Drehung ausgeführt. Die anderen Kanäle führen keine Spindelprogrammierung aus. Nach der 3. Achscontainer-Drehung und der 4. Spindelprogrammierung M3 drehen alle 4 Spindeln mit 1000 U/min rechts herum. Liegt das SERUPRO-Ende nun in diesem Bereich, so wird erwartet, dass jedes ASUP eines Kanals für die dort befindliche Spindel ein M3 S1000 enthält.

Beim Achstausch hingegen müssen die aufgesammelten Hilfsfunktionen nur dem Kanal zugeordnet werden, in dem sich die Spindel aktuell befindet.

Kanalübergreifende Hilfsfunktion

Eine Hilfsfunktion kann bei Satzsuchlauf Typ 5 (SERUPRO) auch kanalübergreifend in der globalen Hilfsfunktionsliste aufgesammelt werden. Es wird dann nur die letzte aufgesammelte Hilfsfunktion dieser Gruppe (höchster Zählerstand) in die globale Liste eingetragen.

Die entsprechende Projektierung erfolgt über folgende Maschinendaten:

- MD22080 \$MC_AUXFU_PREDEF_SPEC[<n>], Bit 11
(Ausgabeverhalten von vordefinierten Hilfsfunktionen)
mit <n> = Systemfunktionsindex
- MD22035 \$MC_AUXFU_ASSIGN_SPEC[<n>], Bit 11
(Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen)
mit <n> = Hilfsfunktionsindex
- MD11110 \$MN_AUXFU_GROUP_SPEC[<n>], Bit 11
(Ausgabeverhalten der Hilfsfunktionen einer Gruppe)
mit <n> = Gruppenindex

Bit	Wert	Bedeutung
11	0	Aufsammlung kanalspezifisch
	1	Aufsammlung kanalübergreifend

Die Spindel-Hilfsfunktionen werden vorab je nach Spindelzustand am Ende des Satzsuchlaufs herausgefiltert. Die Kanaldaten werden entsprechend aktualisiert. Die globale Hilfsfunktionsliste kann in den SERUPRO-Ende-ASUPs der Reihe nach abgearbeitet und die sortierten Hilfsfunktionen kanalsynchron ausgegeben werden.

Abfrage der letzten aufgesammelten Hilfsfunktion

Der Index der letzten aufgesammelten Hilfsfunktion in der globalen Liste kann mit der Systemvariablen \$AN_AUXFU_LIST_ENDINDEX abgefragt werden.

12.13.7 SERUPRO-Ende-ASUP

Funktion

Nach Abschluss des Satzsuchlaufs mit Programmtest (SERUPRO) müssen vor dem Start der nachfolgenden Bearbeitung die während des Satzsuchlaufs aufgesammelten Hilfsfunktionen ausgegeben werden. Während des Satzsuchlaufs werden dazu die Hilfsfunktionen in einer globalen Liste aufgesammelten. Aus dieser Liste generieren kanalspezifisch die SERUPRO-Ende-ASUPs die entsprechenden Teileprogrammätze. Damit wird sicher gestellt, dass die aufgesammelten Hilfsfunktionen sowohl kanalspezifisch als auch kanalübergreifend in zeitlich korrekter Reihenfolge ausgegeben werden. Ein vollständiges funktionsfähiges SERUPRO-Ende-ASUP ist Bestandteil der NC-Software.

Das SERUPRO-Ende-ASUP kann vom Anwender/Maschinenhersteller verändert werden. Die nachfolgend beschriebenen Funktionen unterstützen die Bearbeitung der globalen Liste der Hilfsfunktionen und die Generierung der für die synchronisierte Hilfsfunktionsausgabe erforderlichen Teileprogrammätze.

Funktion AUXFUSYNC(...)

Funktion:

Die Funktion `AUXFUSYNC` generiert aus der globalen Liste der Hilfsfunktionen pro Aufruf einen vollständigen Teileprogrammatz als String. Der Teileprogrammatz enthält entweder Hilfsfunktionen oder Befehle zur Synchronisation der Hilfsfunktionsausgaben (`WAITM`, `G4`, etc.).

Die Funktion löst Vorlaufstopp aus.

Syntax:

```
PROC AUXFUSYNC (VAR INT <NUM>, VAR INT <GROUPINDEX>[10], VAR  
STRING[400] <ASSEMBLED>)
```

Parameter:

<NUM>: Enthält Informationen über den im Parameter <ASSEMBLED> gelieferten Teileprogrammatz bzw. den darin enthaltenen Hilfsfunktionen.

Wertebereich: -1, 0, 1 ... 10

Wert Bedeutung

≥1 Anzahl der im Teileprogrammatz enthaltenen Hilfsfunktionen

0 Teileprogrammatz ohne Hilfsfunktionen, z.B. `WAITM`, `G4`

-1 Ende-Kennung. Die globale Liste der Hilfsfunktionen ist für den aktuellen Kanal vollständig bearbeitet.

<GROUPINDEX>: Enthält die Indizes der im Teileprogrammatz enthaltenen Hilfsfunktionsgruppen. Mit Index = Nummer der Hilfsfunktionsgruppe - 1

<ASSEMBLED>: Enthält den vollständigen Teileprogrammatz für das kanalspezifische SERUPRO-Ende-ASUP als String.

Weitere Informationen:

Wurden Hilfsfunktionen über eine Synchronaktion aufgesammelt, werden zwei NC-Sätze erzeugt. Ein NC-Satz zur Ausgabe der Hilfsfunktionen. Ein ausführbarer NC-Satz über den der NC-Satz zur Ausgabe der Hilfsfunktionen zum Hauptlauf transportiert wird:

1. Ausgabe der Hilfsfunktionen über Synchronaktion, z. B.: `WHEN TRUE DO M100 M102`
2. Ausführbaren NC-Satz, z. B.: `G4 F0.001`

Funktion AUXFUDEL(...)

Funktion:

Die Funktion AUXFUDEL löscht kanalspezifisch für den aufrufenden Kanal die angegebene Hilfsfunktion aus der globalen Liste der Hilfsfunktionen. Das Löschen erfolgt durch Setzen des entsprechenden Gruppenindex ...GROUPINDEX[n] auf 0.

Der Aufruf der Funktion muss vor dem Aufruf von AUXFUSYNC erfolgen.

Die Funktion löst Vorlaufstopp aus.

Syntax:

```
PROC AUXFUDEL (CHAR <TYPE>, INT <EXTENSION>, REAL <VALUE>, INT
<GROUP>)
```

Parameter:

<TYPE>:	Typ der zu löschenden Hilfsfunktion
<EXTENSION>:	Adresserweiterung der zu löschenden Hilfsfunktion
<VALUE>:	Wert der zu löschenden Hilfsfunktion
<GROUP>:	Nummer der Hilfsfunktionsgruppe

Funktion AUXFUDELG(...)

Funktion:

Die Funktion AUXFUDELG löscht kanalspezifisch für den aufrufenden Kanal alle Hilfsfunktionen der angegebenen Hilfsfunktionsgruppe aus der globalen Liste der Hilfsfunktionen. Das Löschen erfolgt durch Setzen des entsprechenden Gruppenindex ...GROUPINDEX[n] auf 0.

Der Aufruf der Funktion muss vor dem Aufruf von AUXFUSYNC erfolgen.

Die Funktion löst Vorlaufstopp aus.

Syntax:

```
PROC AUXFUDELG (INT <GROUP>)
```

Parameter:

<GROUP>:	Nummer der Hilfsfunktionsgruppe
----------	---------------------------------

Mehrkanaliger Satzsuchlauf

ACHTUNG

Mehrkanaliger Satzsuchlauf und AUXFUDEL / AUXFUDELG

Werden bei einem mehrkanaligen Satzsuchlauf in den SERUPRO-Ende-ASUPs Hilfsfunktionen mit AUXFUDEL / AUXFUDELG aus der globalen Liste der Hilfsfunktionen gelöscht, muss vor dem Aufruf der Funktion AUXFUSYNC, eine Synchronisation der beteiligten Kanäle erfolgen. Durch die Synchronisation ist dann sicher gestellt, dass vor dem Aufruf der Funktion AUXFUSYNC alle Löschaufträge bearbeitet wurden und eine konsistente Liste vorliegt.

Beispiele

Zwei Beispiele für den Aufbau eines anwenderspezifischen SERUPRO-Ende ASUPs.

Beispiel 1: Löschen von Hilfsfunktionen und Erzeugen der Hilfsfunktionsausgabe mit AUXFUSYNC(...)

Programmcode	Kommentar
N10 DEF STRING[400] ASSEMBLED=""	
N20 DEF STRING[31] FILENAME="/_N_CST_DIR/_N_AUXFU_SPF"	
N30 DEF INT GROUPINDEX[10]	
N40 DEF INT NUM	
N60 DEF INT ERROR	
N90	
N140 AUXFUDEL("M",2,3,5)	; M2=3 (5. Hilfsfunktionsgruppe) löschen
N150	
N170 AUXFUDELG(6)	; die aufgesammelte Hilfsfunktion der ; 6. Gruppe löschen.
N180	
N190 IF ISFILE(FILENAME)	
N210 DELETE(ERROR,FILENAME)	; Datei FILENAME löschen
N220 IF (ERROR<>0)	; Fehlerauswertung
N230 SETAL(61000+ERROR)	
N240 ENDIF	
N250 ENDIF	
; VORSICHT!	
; Werden bei einem mehrkanaligen Satzsuchlauf Hilfsfunktionen mit AUXFUDEL/AUXFUDELG	
; aus der globalen Liste der Hilfsfunktionen gelöscht, muss vor der Schleife zum	
; Erzeugen des Unterprogramms FILENAME mit AUXFUSYNC eine Synchronisation der Kanäle	
; erfolgen. Durch die Synchronisation wird sicher gestellt, dass alle Löschaufträge	
; in allen Kanälen bearbeitet wurden und eine konsistente Liste vorliegt.	
; Beispiel: WAITM(99,1,2,3)	
N270 LOOP	
N300 AUXFUSYNC(NUM,GROUPINDEX,ASSEMBLED)	; Teileprogrammsatz erzeugen lassen

Programmcode	Kommentar
N310	
N320 IF (NUM==--1)	; Alle Hilfsfunktionen des Kanals
	; sind abgearbeitet.
N340 GOTO LABEL1	
N350 ENDIF	
N380 WRITE (ERROR, FILENAME, ASSEMBLED)	; Teileprogrammsatz in Datei FILENAME schreiben.
N390 IF (ERROR<>0)	; Fehlerauswertung
N400 SETAL (61000+ERROR)	
N410 ENDIF	
N430 ENDLOOP	
N440	
N450 LABEL1:	
N460	
N480 CALL FILENAME	; Erzeugtes Unterprogramm abarbeiten.
N490	
N510 DELETE (ERROR, FILENAME)	; Datei nach Ausführung wieder löschen.
N520 IF (ERROR<>0)	
N530 SETAL (61000+ERROR)	
N540 ENDIF	
N550	
N560 M17	

Beispiel 2: Löschen von Hilfsfunktionen und Erzeugen der Hilfsfunktionsausgabe ohne AUXFUSYNC(...)

Programmcode	Kommentar
N0610 DEF STRING[400] ASSEMBLED=""	
N0620 DEF STRING[31] FILENAME="/_N_CST_DIR/_N_AUXFU_SPF"	
N0630 DEF INT GROUPINDEX[10]	
N0640 DEF INT NUM	
N0650 DEF INT LAUF	
N0660 DEF INT ERROR	
N0670 DEF BOOL ISQUICK	
N0680 DEF BOOL ISSYNACT	
N0690 DEF BOOL ISIMPL	
...	
N0760 AUXFUDEL ("M", 2, 3, 5)	; M2=3 (5. Hilfsfunktionsgruppe) löschen
N0770	
N0790 AUXFUDELG (6)	; die aufgesammelte Hilfsfunktion der
	; 6. Gruppe löschen.
N0800	
N0810 IF ISFILE (FILENAME)	
N0830 DELETE (ERROR, FILENAME)	; Datei ist bereits vorhanden und muss
	; gelöscht werden.

12.13 Verhalten bei Satzsuchlauf

Programmcode	Kommentar
N0840	IF (ERROR<>0)
N0850	SETAL(61000+ERROR)
N0860	ENDIF
N0870	ENDIF
N0880	
	; VORSICHT!
	; Werden bei einem mehrkanaligen Satzsuchlauf Hilfsfunktionen mit AUXFUDEL/AUXFUDELG
	; aus der globalen Liste der Hilfsfunktionen gelöscht, muss vor der Schleife zum
	; Erzeugen des Unterprogramms FILENAME mit AUXFUSYNC eine Synchronisation der Kanäle
	; erfolgen. Durch die Synchronisation wird sicher gestellt, dass alle Löschaufträge
	; in allen Kanälen bearbeitet wurden und eine konsistente Liste vorliegt.
	; Beispiel: WAITM(99,1,2,3)
N0890	LOOP
N0920	AUXFUSYNC(NUM,GROUPINDEX,ASSEMBLED)
	; Prozedur zur Erzeugung von
	; Hilfsfunktionssätzen aus der globalen
	; Hilfsfunktionsliste.
N0930	
N0940	IF (NUM==--1)
	; Alle Hilfsfunktionen des Kanals sind
	; abgearbeitet.
N0960	GOTOF LABEL1
N0970	ENDIF
N0980	
N1000	IF (NUM>0)
	; Wenn Hilfsfunktionen zur Ausgabe kommen
	; wird der Satz erzeugt.
N1010	
N1020	ASSEMBLED=""
N1030	
N1050	FOR LAUF=0 TO NUM-1
	; Aufgesammelte Hilfsfunktionen für einen
	; Satz.
N1060	
N1080	IF GROUPINDEX[LAUF]<>0
	; Aus der globalen Liste gelöschte
	; Hilfsfunktionen haben den Gruppenindex 0.
N1090	
N1100	ISQUICK=\$AC_AUXFU_SPEC[GROUPINDEX[LAUF]] BAND'H2'
N1110	
N1120	ISSYNACT=\$AC_AUXFU_SPEC[GROUPINDEX[LAUF]] BAND'H1000'
N1130	
N1140	ISIMPL=\$AC_AUXFU_SPEC[GROUPINDEX[LAUF]] BAND'H2000'
N1150	
N1180	IF ISSYNACT
	; Satz für die M-Hilfsfunktionsausgabe
	; zusammenbauen
N1190	ASSEMBLED= ASSEMBLED << "WHEN TRUE DO "
N1200	ENDIF
N1210	; Implizit erzeugtes M19 wird auf SPOS[SPI(<Spindelnr.>)] = IC(0) abgebildet.

Programmcode	Kommentar
N1230	IF (ISIMPL AND (\$AC_AUXFU_VALUE[GROUPINDEX[LAUF]]==19))
N1240	ASSEMBLED= ASSEMBLED << "SPOS[SPI(" << \$AC_AUXFU_EXT[GROUPINDEX[LAUF]] << ")=IC(0) "
N1260	ELSE
N1270	ASSEMBLED= ASSEMBLED << "M[" << \$AC_AUXFU_EXT[GROUPINDEX[LAUF]] << "]="
N1280	
N1290	IF ISQUICK
N1300	ASSEMBLED= ASSEMBLED << "QU("
N1310	ENDIF
N1320	
N1330	ASSEMBLED= ASSEMBLED << \$AC_AUXFU_VALUE[GROUPINDEX[LAUF]]
N1340	
N1350	IF ISQUICK
N1360	ASSEMBLED= ASSEMBLED << ") "
N1370	ENDIF
N1380	ENDIF
N1400	ENDIF
N1420	ENDFOR
N1430	
N1450	WRITE (ERROR, FILENAME, ASSEMBLED) ; Hilfsfunktionssatz in Datei schreiben.
N1460	
N1470	IF ISSYNACT
N1480	ASSEMBLED="G4 F0.001"
N1490	WRITE (ERROR, FILENAME, ASSEMBLED)
N1500	ENDIF
N1510	
N1520	ELSE
N1540	WRITE (ERROR, FILENAME, ASSEMBLED) ; Hilfsfunktionssatz in Datei schreiben.
N1550	ENDIF
N1560	
N1570	ENDLOOP
N1580	
N1590	LABEL1:
N1600	
N1620	CALL FILENAME ; Erzeugtes Unterprogramm abarbeiten.
N1630	
N1650	DELETE (ERROR, FILENAME) ; Datei nach Ausführung wieder löschen.
N1660	IF (ERROR<>0)
N1670	SETAL (61000+ERROR)
N1680	ENDIF
N1690	
N1700	M17

12.14 Implizit ausgegebene Hilfsfunktionen

Funktion

Implizit ausgegebene Hilfsfunktionen sind Hilfsfunktionen, die nicht explizit programmiert wurden und zusätzlich von anderen Systemfunktionen (z. B. Transformationsanwahl, Werkzeuganwahl, etc.) ausgegeben werden. Diese impliziten Hilfsfunktionen führen zu keiner Systemfunktion, sondern die M-Codes werden entsprechend ihres parametrisierten Ausgabeverhaltens aufgesammelt und/oder an die PLC ausgegeben.

Parametrierung

Die M-Codes für implizit auszugebende Hilfsfunktionen werden festgelegt mit den Maschinendaten:

- MD22530 \$MC_TOCARR_CHANGE_M_CODE (M-Code bei Werkzeugträgerwechsel)
Der Betrag dieses Maschinendatums gibt die Nummer des M-Codes an, der bei Aktivierung eines Werkzeugträgers an der NC/PLC-Nahtstelle ausgegeben wird.
Ist der Wert positiv, wird immer der unveränderte M-Code ausgegeben.
Ist der Wert negativ, wird die Nummer des Werkzeugträgers zum Betrag des Maschinendatums addiert und diese Nummer ausgegeben.
- MD22532 \$MC_GEOAX_CHANGE_M_CODE (M-Code bei Umschaltung der Geometrieachsen)
Nummer des M-Codes, der bei einer Umschaltung der Geometrieachsen an der NC/PLC-Nahtstelle ausgegeben wird.
- MD22534 \$MC_TRAFO_CHANGE_M_CODE (M-Code bei Transformationswechsel)
Nummer des M-Codes, der bei einer Transformationsumschaltung der Geometrieachsen an der NC/PLC-Nahtstelle ausgegeben wird.

Hinweis

Hat die Nummer des auszugebenden M-Codes oder der Betrag von MD22530 / MD22532 / MD22534 selbst einen der Werte 0 bis 6, 17 oder 30, wird kein M-Code ausgegeben. Es wird nicht überwacht, ob ein derart erzeugter M-Code zu Konflikten mit anderen Funktionen führt.

Ausgabeverhalten

Bei implizit ausgegebenen Hilfsfunktionen ist das Bit 13 im Maschinendatum MD22080 bzw. MD22035 (Ausgabeverhalten von vordefinierten bzw. anwenderdefinierten Hilfsfunktionen) gesetzt.

Über die Systemvariable \$AC_AUXFU_SPEC[<n>] kann dieses Bit abgefragt werden.

Implizit ausgegebene Hilfsfunktion M19

Um eine Durchgängigkeit von M19 und SPOS bzw. SPOSA bezüglich des Verhaltens an der NC/PLC-Nahtstelle zu erreichen, kann bei SPOS und SPOSA die Hilfsfunktion M19 an die NC/PLC-Nahtstelle ausgegeben werden (siehe Funktionshandbuch *Achsen und Spindeln*, Kapitel *Spindeln*).

Die Implizit ausgegebene Hilfsfunktion M19 wird beim Satzsuchlauf aufgesammelt.

12.15 Informationsmöglichkeiten

Informationen zu Hilfsfunktionen (z. B. über den Ausgabestatus) sind möglich über:

- die gruppenspezifische modale M-Hilfsfunktionsanzeige an der Bedienoberfläche.
- die Abfrage von Systemvariablen in Teileprogrammen und Synchronaktionen.

12.15.1 Gruppenspezifische modale M-Hilfsfunktionsanzeige

Funktion

An der Bedienoberfläche kann der Ausgabe- und Quittungsstatus von M-Hilfsfunktionen gruppenspezifisch angezeigt werden.

Voraussetzungen

Vorraussetzung für die Realisierung einer funktionsorientierten Quittierung und Anzeige von M-Hilfsfunktionen ist, dass die Verwaltung der Hilfsfunktionen in der PLC, also im Anwenderprogramm selbst, erfolgen muss. Deshalb obliegt es den PLC-Programmierer, die Quittierung dieser Hilfsfunktionen selbst vorzunehmen. Er muss wissen, welche Hilfsfunktionen in welcher Gruppe sind, die er quittieren muss.

Standard

M-Hilfsfunktionen, die nicht über die PLC verwaltet werden, werden vom NC an die PLC ausgegeben und als "übergeben" gekennzeichnet. Eine funktionale Quittierung gibt es für diese Hilfsfunktionen nicht. Alle nach einem Satzsuchlauf aufgesammelten M-Hilfsfunktionen werden zusätzlich zur Anzeige gebracht, damit der Bediener weiß, welche Hilfsfunktionen nach dem Start nach Satzsuchlauf ausgegeben werden.

PLC-Aktivitäten

Bei Hilfsfunktionsgruppen, die von der PLC selbst verwaltet werden, muss das PLC-Anwenderprogramm alle Hilfsfunktionen dieser Gruppe bei **Übernahme** und **Funktionsende** quittieren. Der PLC-Programmierer muss alle Hilfsfunktionen dieser Gruppen kennen.

Sonstiges

Es werden nur die M-Hilfsfunktionen **gruppenspezifisch** angezeigt. Die satzweise Anzeige bleibt zusätzlich erhalten. Es können bis zu 15 Gruppen angezeigt werden, wobei je Gruppe immer **nur die letzte M-Funktion einer Gruppe**, die entweder aufgesammelt oder an die PLC ausgegeben wurde, angezeigt wird. Die M-Funktionen werden ihrem Status entsprechend in verschiedener Anzeigeart dargestellt:

Status	Anzeigeart
Hilfsfunktion ist aufgesammelt	Invertiert mit gelber Schrift
Hilfsfunktion ist von NC an PLC ausgegeben	Invertiert
Hilfsfunktion ist von NC an PLC übergeben worden und Transportquittierung ist erfolgt	Schwarze Schrift auf grauem Hintergrund
Hilfsfunktion wird von PLC verwaltet und ist von PLC direkt übernommen worden.	Schwarze Schrift auf grauem Hintergrund
Hilfsfunktion wird von PLC verwaltet und die Funktionsquittierung ist erfolgt.	Schwarze Schrift auf grauem Hintergrund

Anzeigeaktualisierung

Die Anzeige wird so organisiert, dass zuerst immer die aufgesammelten Hilfsfunktionen vor den PLC-verwalteten und vor den NC-verwalteten ausgegeben werden. Eine aufgesammelte Hilfsfunktion bleibt solange als aufgesammelt markiert, bis sie vom NC an die PLC ausgegeben wurde. PLC-verwaltete Hilfsfunktionen bleiben solange bestehen, bis diese von einer anderen Hilfsfunktion abgelöst werden. Bei Reset werden nur die aufgesammelten und die NC-verwalteten gelöscht.

12.15.2 Abfrage von Systemvariablen

Funktion

Im Teileprogramm und über Synchronaktionen können Hilfsfunktionen gruppenspezifisch über Systemvariablen abgefragt werden:

\$AC_AUXFU_... [<n>] = <Wert>

Systemvariable	Bedeutung	
\$AC_AUXFU_PREDEF_INDEX[<n>]	<Wert>:	Index der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Satzsuchlauf) oder ausgegebenen vordefinierten Hilfsfunktion
		Typ: INT
		Ist für die spezifizierte Gruppe noch keine Hilfsfunktion ausgegeben worden oder ist die Hilfsfunktion eine anwenderdefinierte Hilfsfunktion, so liefert die Variable den Wert "-1".
	<n>:	Gruppenindex (0 ... 63)
	Hinweis: Über diese Variable lässt sich eine vordefinierte Hilfsfunktion eindeutig identifizieren.	
\$AC_AUXFU_TYPE[<n>]	<Wert>:	Typ der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Satzsuchlauf) oder ausgegebenen Hilfsfunktion
		Typ: CHAR
	<n>:	Gruppenindex (0 ... 63)
\$AC_AUXFU_EXT[<n>] bzw. M-Funktionsspezifisch: \$AC_AUXFU_M_EXT[<n>]	<Wert>:	Adresserweiterung der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Satzsuchlauf) oder ausgegebenen Hilfsfunktion
		Typ: INT
	<n>:	Gruppenindex (0 ... 63)
\$AC_AUXFU_VALUE[<n>] bzw. M-Funktionsspezifisch: \$AC_AUXFU_M_VALUE[<n>]	<Wert>:	Wert der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Satzsuchlauf) oder ausgegebenen Hilfsfunktion
		Typ: REAL
	<n>:	Gruppenindex (0 ... 63)
\$AC_AUXFU_SPEC[<n>]	Wert:	Bit-kodiertes Ausgabeverhalten entsprechend MD22080/MD22035 (oder QU-Programmierung) der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Satzsuchlauf) oder ausgegebenen Hilfsfunktion
		Typ: INT
	<n>:	Gruppenindex (0 ... 63)
		Hinweis: Über diese Variable lässt sich ermitteln, ob die Hilfsfunktionsausgabe mit schneller Quittierung erfolgen soll.

Systemvariable	Bedeutung	
\$AC_AUXFU_STATE[<n> bzw. M-Funktionsspezifisch: \$AC_AUXFU_M_STATE[<n>]	<Wert>:	Ausgabestatus der zuletzt für eine Hilfsfunktionsgruppe aufgesammelten (Satzsuchlauf) oder ausgegebenen Hilfsfunktion
	Typ:	INT
	Wertebereich:	0 ... 5
	0:	Hilfsfunktion ist nicht vorhanden
	1:	M-Hilfsfunktion wurde per Satzsuchlauf aufgesammelt
	2:	M-Hilfsfunktion wurde an die PLC ausgegeben
	3:	M-Hilfsfunktion wurde an die PLC ausgegeben und die Transportquittierung ist erfolgt
	4:	M-Hilfsfunktion wird von der PLC verwaltet und ist von der PLC übernommen worden
	5:	M-Hilfsfunktion wird von der PLC verwaltet und die Funktionsquittierung ist erfolgt
<n>:	Gruppenindex (0 ... 63)	

Beispiel

Alle M-Hilfsfunktionen der 1. Gruppe sollen der Reihe ihrer Ausgabe nach abgespeichert werden:

```
id=1 every $AC_AUXFU_M_STATE[0]==2 do $AC_FIFO[0,0]=
$AC_AUXFU_M_VALUE[0]
```

Weitere Informationen

Listenhandbuch Systemvariablen

12.16 Randbedingungen

12.16.1 Allgemeine Randbedingungen

Spindeltausch

Da die Parametrierung der Hilfsfunktionen kanalspezifisch erfolgt, müssen bei Verwendung der Funktion: "Spindeltausch" die Spindel-spezifischen Hilfsfunktionen in allen Kanälen in denen die Spindel verwendet wird, gleich parametriert werden.

Werkzeugverwaltung

Bei aktiver Werkzeugverwaltung gelten folgende Randbedingungen:

- T- und M<k>-Funktionen werden nicht an die PLC ausgegeben.
Hinweis
k ist der parametrisierte Wert der Hilfsfunktion für den Werkzeugwechsel (Default: 6):
MD22560 \$MC_TOOL_CHANGE_M_CODE (Hilfsfunktion für Werkzeugwechsel)
- Wird keine Adresserweiterung programmiert, bezieht sich die Hilfsfunktion auf die Master-Spindel bzw. des Master-Werkzeughalters des Kanals.
Definition der Master-Spindel:
 - MD20090 \$MC_SPIND_DEF_MASTER_SPIND
 - Teileprogrammanweisung: SETMS
 Definition des Master-Werkzeughalters
 - MD20124 \$MC_TOOL_MANAGEMENT_TOOLHOLDER
 - Teileprogrammanweisung: SETMTH

Maximale Anzahl von Hilfsfunktionen pro Teileprogrammsatz

In einem Teileprogrammsatz dürfen maximal 10 Hilfsfunktionen programmiert werden.

DL (Summenkorrektur)

Für die DL-Funktion gelten folgende Einschränkungen:

- Pro Teileprogrammsatz kann nur eine DL-Funktion programmiert werden.
- Bei Verwendung von DL-Funktionen in Synchronaktionen wird der Parameter: "Wert" nicht an die PLC ausgegeben.

12.16.2 Ausgabeverhalten

Gewindeschneiden

Während aktivem Gewindeschneiden G33, G34 und G35 wirkt für die Spindel-spezifischen Hilfsfunktionen:

- M3 (Spindel rechts)
- M4 (Spindel links)

immer mit folgendes Ausgabeverhalten:

- Ausgabedauer einen OB40-Takt (Schnelle Quittierung)
- Ausgabe während der Bewegung

Die Spindel-spezifische Hilfsfunktion M5 (Spindel halt) wird immer am Satzende ausgegeben. Der Teileprogrammsatz, der M5 beinhaltet, wird immer d. h. auch bei aktivem Bahnsteuerbetrieb, mit Genauhalt beendet.

Synchronaktionen

Bei der Hilfsfunktionsausgabe aus Synchronaktionen wird das parametrierte Ausgabeverhalten bis auf folgende Parameter ignoriert:

- Bit0: Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
- Bit1: Ausgabedauer einen OB40-Takt (Schnelle Quittierung)

Hilfsfunktionen: M17 bzw. M2 / M30 (Unterprogrammende)

Allein in einem Teileprogrammsatz

Steht eine der Hilfsfunktionen M17, M2 oder M30 allein in einem Teileprogrammsatz und es ist noch eine Achse in Bewegung, erfolgt die Ausgabe der Hilfsfunktion an die PLC erst, nachdem die Achse steht.

Überlagerung des parametrierten Ausgabeverhaltens

Das parametrierte Ausgabeverhalten der Hilfsfunktionen M17 bzw. M2/M30 wird durch das im folgenden Maschinendatum festgelegte Ausgabeverhalten überlagert:

MD20800 \$MC_SPF_END_TO_VDI, Bit 0 (Unterprogrammende / Halt an PLC)

Bit	Wert	Bedeutung
0	0	Die Hilfsfunktionen M17 bzw. M2/M30 (Unterprogrammende) werden nicht an die PLC ausgegeben. Bahnsteuerbetrieb wird am Unterprogrammende nicht unterbrochen.
	1	Die Hilfsfunktionen M17 bzw. M2/M30 (Unterprogrammende) werden an die PLC ausgegeben.

Hilfsfunktion: M1 (Bedingter Halt)

Überlagerung des parametrierten Ausgabeverhaltens

Das parametrierte Ausgabeverhalten der Hilfsfunktion M1 wird durch das im folgenden Maschinendatum festgelegte Ausgabeverhalten überlagert:

MD20800 \$MC_SPF_END_TO_VDI, Bit 1 (Unterprogrammende / Halt an PLC)

Bit	Wert	Bedeutung
1	0	Die Hilfsfunktionen M01 (Bedingter Halt) wird immer an PLC ausgegeben. Eine schnelle Quittierung ist unwirksam, da M01 fest der 1. Hilfsfunktionsgruppe zugeordnet ist und damit immer am Satzende ausgegeben wird.
	1	Die Hilfsfunktionen M01 (Bedingter Halt) wird nur dann an PLC ausgegeben, wenn die Funktion: "Programmierter Halt" über die HMI-Bedienoberfläche aktiviert ist. Bei schneller Quittierung wird die M1 während der Bewegung an die PLC ausgegeben. Solange die Funktion nicht aktiv ist, wird dadurch ein Bahnsteuerbetrieb nicht unterbrochen.

Teileprogrammsätze ohne Verfahrbewegung

In einem Teileprogrammsatz ohne Verfahrbewegung werden alle Hilfsfunktionen, unabhängig von ihrem jeweiligen parametrierten Ausgabeverhalten, sofort in einem Block ausgegeben.

Spindelspezifische Hilfsfunktionsausgabe nur als Information für das PLC-Anwenderprogramm

In bestimmten Steuerungssituationen, z.B. zum Abschluss eines Satzsuchlaufs, werden die aufgesammelten spindelspezifischen Hilfsfunktionen (z.B. M3, M4, M5, M19, M40...M45, M70) nur zur Information für das PLC-Anwenderprogramm an die NC/PLC-Nahtstelle ausgegeben. Die Steuerung erzeugt dazu einen Teileprogrammsatz (Aktionssatz) in welchem die aufgesammelten Hilfsfunktionen mit **negativer** Adresserweiterung eingetragen sind. Die entsprechenden Systemfunktionen werden dann **nicht** ausgeführt.

Beispiel: M(-2) = 41 Anforderung Getriebestufenwechsel für die 2. Spindel

12.17 Beispiele

12.17.1 Erweiterung von vordefinierten Hilfsfunktionen

Aufgabe

Parametrierung der Hilfsfunktionen M3, M4 und M5 für die 2. Spindel des Kanals.

Parametrierung: M3

Anforderungen:

- Maschinendaten-Index: 0 (erste anwenderdefinierte Hilfsfunktion)
- Hilfsfunktionsgruppe: 5
- Typ und Wert: M3 (Spindel rechts)
- Adresserweiterung: 2 entsprechend der 2. Spindel des Kanals
- Ausgabeverhalten:
 - Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
 - Ausgabe vor der Bewegung

Parametrierung:

```
MD22000 $MC_AUXFU_ASSIGN_GROUP [ 0 ]      = 5
MD22010 $MC_AUXFU_ASSIGN_TYPE [ 0 ]       = "M"
MD22020 $MC_AUXFU_ASSIGN_EXTENSION [ 0 ]  = 2
MD22030 $MC_AUXFU_ASSIGN_VALUE [ 0 ]     = 3
MD22035 $MC_AUXFU_ASSIGN_SPEC [ 0 ]      = 'H21'
```

Parametrierung: M4

Anforderungen:

- Maschinendaten-Index: 1 (zweite anwenderdefinierte Hilfsfunktion)
- Hilfsfunktionsgruppe: 5
- Typ und Wert: M4 (Spindel links)
- Adresserweiterung: 2 entsprechend der 2. Spindel des Kanals
- Ausgabeverhalten:
 - Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
 - Spindelreaktion nach der Quittierung
 - Ausgabe während der Bewegung

Parametrierung:

MD22000 \$MC_AUXFU_ASSIGN_GROUP [1]	= 5
MD22010 \$MC_AUXFU_ASSIGN_TYPE [1]	= "M"
MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [1]	= 2
MD22030 \$MC_AUXFU_ASSIGN_VALUE [1]	= 4
MD22035 \$MC_AUXFU_ASSIGN_SPEC [1]	= 'H51'

Parametrierung: M5

Anforderungen:

- Maschinendaten-Index: 2 (dritte anwenderdefinierte Hilfsfunktion)
- Hilfsfunktionsgruppe: 5
- Typ und Wert: M5 (Spindel halt)
- Adresserweiterung: 2 entsprechend der 2. Spindel des Kanals
- Ausgabeverhalten:
 - Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
 - Spindelreaktion nach der Quittierung
 - Ausgabe am Satzende

Parametrierung:

MD22000 \$MC_AUXFU_ASSIGN_GROUP [2]	= 5
MD22010 \$MC_AUXFU_ASSIGN_TYPE [2]	= "M"
MD22020 \$MC_AUXFU_ASSIGN_EXTENSION [2]	= 2
MD22030 \$MC_AUXFU_ASSIGN_VALUE [2]	= 5
MD22035 \$MC_AUXFU_ASSIGN_SPEC [2]	= 'H91'

12.17.2 Definition von Hilfsfunktionen

Aufgabe

Parametrierung der Hilfsfunktionsspezifischen Maschinendaten für eine Maschine mit folgender Konfiguration:

Spindeln

- Spindel 1: Masterspindel
- Spindel 2: Zweite Spindel

Getriebestufen

- Spindel 1: 5 Getriebestufen
- Spindel 2: keine Getriebestufen

Schaltfunktionen für Kühlwasser Ein/Aus

- Spindel 1
 - "Ein" = M50
 - "Aus" = M51
- Spindel 2
 - "Ein" = M52
 - "Aus" = M53

Anforderungen

Spindel 1 (Masterspindel)

Hinweis

Standardzuordnungen

- Die Hilfsfunktionen M3, M4, M5, M70 und M1=3, M1=4, M1=5, M1=70 der Spindel 1 (Masterspindel) sind standardmäßig der 2. Hilfsfunktionsgruppe zugeordnet.
 - Alle S- und S1-Werte der Spindel 1 (Masterspindel) sind standardmäßig der 3. Hilfsfunktionsgruppe zugeordnet.
-

12.17 Beispiele

- Nach Satzsuchlauf soll die zuletzt programmierte Getriebestufe ausgegeben werden. Die folgenden Hilfsfunktionen werden dazu der 9. Hilfsfunktionsgruppe zugeordnet:
 - M40, M41, M42, M43, M44, M45
 - M1=40, M1=41, M1=42, M1=43, M1=44, M1=45
- Die Hilfsfunktionen M3, M4, M5, M70 und M1=3, M1=4, M1=5, M1=70 (2. Hilfsfunktionsgruppe) sowie S- und S1-Werte (3. Hilfsfunktionsgruppe) sollen folgendes Ausgabeverhalten haben:
 - Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung)
 - Ausgabe vor der Bewegung
- Die Hilfsfunktionen zum Getriebeumschalten M40 bis M45 und M1=40 bis M1=45 (9. Hilfsfunktionsgruppe) sollen folgendes Ausgabeverhalten haben:
 - Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
 - Ausgabe vor der Bewegung

Spindel 2

- In einem Satz darf nur eine M-Funktion zur Drehrichtungsumschaltung programmiert werden. Nach Satzsuchlauf soll die zuletzt programmierte Drehrichtung ausgegeben werden. Die folgenden Hilfsfunktionen werden dazu der 10. Hilfsfunktionsgruppe zugeordnet:
 - M2=3, M2=4, M2=5, M2=70
- Alle S2-Werte werden der 11. Hilfsfunktionsgruppe zugeordnet.
- Die Hilfsfunktionen M2=3, M2=4, M2=5, M2=70 (10. Hilfsfunktionsgruppe) sowie S2-Werte (11. Hilfsfunktionsgruppe) sollen folgendes Ausgabeverhalten haben:
 - Ausgabedauer einen OB40-Zyklus (Schnelle Quittierung)
 - Ausgabe vor der Bewegung

Kühlwasser

- Ein- und Ausschalten in einem Teileprogrammsatz ist nicht zulässig. Nach Satzsuchlauf soll das Kühlwasser ein- bzw. ausgeschaltet werden. Die folgenden Hilfsfunktionen werden dazu z. B. der 12. bzw. 13. Hilfsfunktionsgruppe zugeordnet:
 - 12. Hilfsfunktionsgruppe: M50, M51
 - 13. Hilfsfunktionsgruppe: M52, M53
- Die Hilfsfunktionen M50, M51 (12. Hilfsfunktionsgruppe) und M52, M53 (13. Hilfsfunktionsgruppe) sollen folgendes Ausgabeverhalten haben:
 - Ausgabedauer einen OB1-Zyklus (Normale Quittierung)
 - Ausgabe vor der Bewegung

Parametrierung der Maschinendaten

Die Parametrierung der Maschinendaten erfolgt über eine entsprechende Programmierung innerhalb eines Teileprogramms:

Programmcode	Kommentar
\$MN_AUXFU_MAXNUM_GROUP_ASSIGN=21	; Anzahl anwenderdefinierter Hilfsfunktionen pro Kanal
\$MN_AUXFU_GROUP_SPEC[1]='H22'	; Ausgabeverhalten der 2.Hilfsfunktionsgruppe
\$MN_AUXFU_GROUP_SPEC[2]='H22'	; Ausgabeverhalten der 3.Hilfsfunktionsgruppe
\$MN_AUXFU_GROUP_SPEC[8]='H21'	; Ausgabeverhalten der 9.Hilfsfunktionsgruppe
\$MC_AUXFU_ASSIGN_TYPE[0]="M"	; Beschreibung der 1.Hilfsfunktion: M40
\$MC_AUXFU_ASSIGN_EXTENSION[0]=0	
\$MC_AUXFU_ASSIGN_VALUE[0]=40	
\$MC_AUXFU_ASSIGN_GROUP[0]=9	
	; ... (entsprechend für 2.-5.Hilfsfunktion)
\$MC_AUXFU_ASSIGN_TYPE[5]="M"	; Beschreibung der 6.Hilfsfunktion: M45
\$MC_AUXFU_ASSIGN_EXTENSION[5]=0	
\$MC_AUXFU_ASSIGN_VALUE[5]=45	
\$MC_AUXFU_ASSIGN_GROUP[5]=9	
\$MC_AUXFU_ASSIGN_TYPE[6]="M"	; Beschreibung der 7.Hilfsfunktion: M1=40
\$MC_AUXFU_ASSIGN_EXTENSION[6]=1	
\$MC_AUXFU_ASSIGN_VALUE[6]=40	
\$MC_AUXFU_ASSIGN_GROUP[6]=9	
	; . . . (entsprechend für 8.-11.Hilfsfunktion)
\$MC_AUXFU_ASSIGN_TYPE[11]="M"	; Beschreibung der 12.Hilfsfunktion: M1=45
\$MC_AUXFU_ASSIGN_EXTENSION[11]=1	
\$MC_AUXFU_ASSIGN_VALUE[11]=45	
\$MC_AUXFU_ASSIGN_GROUP[11]=9	
\$MN_AUXFU_GROUP_SPEC[9] = 'H22'	; Ausgabeverhalten der 10.Hilfsfunktionsgruppe
\$MC_AUXFU_ASSIGN_TYPE[12]="M"	; Beschreibung der 13.Hilfsfunktion: M2=3
\$MC_AUXFU_ASSIGN_EXTENSION[12]=2	
\$MC_AUXFU_ASSIGN_VALUE[12]=3	
\$MC_AUXFU_ASSIGN_GROUP[12]=10	
\$MC_AUXFU_ASSIGN_TYPE[13]="M"	; Beschreibung der 14.Hilfsfunktion: M2=4
\$MC_AUXFU_ASSIGN_EXTENSION[13]=2	
\$MC_AUXFU_ASSIGN_VALUE[13]=4	
\$MC_AUXFU_ASSIGN_GROUP[13]=10	
\$MC_AUXFU_ASSIGN_TYPE[14]="M"	; Beschreibung der 15.Hilfsfunktion: M2=5
\$MC_AUXFU_ASSIGN_EXTENSION[14]=2	

12.17 Beispiele

Programmcode	Kommentar
\$MC_AUXFU_ASSIGN_VALUE[14]=5 \$MC_AUXFU_ASSIGN_GROUP[14]=10	
\$MC_AUXFU_ASSIGN_TYPE[15]="M" \$MC_AUXFU_ASSIGN_EXTENSION[15]=2 \$MC_AUXFU_ASSIGN_VALUE[15]=70 \$MC_AUXFU_ASSIGN_GROUP[15]=10	; Beschreibung der 16.Hilfsfunktion: M2=70
\$MN_AUXFU_GROUP_SPEC[10]='H22'	; Spezifikation der 11.Hilfsfunktionsgruppe
\$MC_AUXFU_ASSIGN_TYPE[16]="S" \$MC_AUXFU_ASSIGN_EXTENSION[16]=2 \$MC_AUXFU_ASSIGN_VALUE[16]=-1 \$MC_AUXFU_ASSIGN_GROUP[16]=11	; Beschreibung der 17.Hilfsfunktion: S2=<alle Werte>
\$MN_AUXFU_GROUP_SPEC[11]='H21'	; Spezifikation der 12.Hilfsfunktionsgruppe
\$MC_AUXFU_ASSIGN_TYPE[17]="M" \$MC_AUXFU_ASSIGN_EXTENSION[17]=0 \$MC_AUXFU_ASSIGN_VALUE[17]=50 \$MC_AUXFU_ASSIGN_GROUP[17]=12	; Beschreibung der 18.Hilfsfunktion: M50
\$MC_AUXFU_ASSIGN_TYPE[18]="M" \$MC_AUXFU_ASSIGN_EXTENSION[18]=0 \$MC_AUXFU_ASSIGN_VALUE[18]=51 \$MC_AUXFU_ASSIGN_GROUP[18]=12	; Beschreibung der 19.Hilfsfunktion: M51
\$MN_AUXFU_GROUP_SPEC[12]='H21'	; Spezifikation der 13.Hilfsfunktionsgruppe
\$MC_AUXFU_ASSIGN_TYPE[19]="M" \$MC_AUXFU_ASSIGN_EXTENSION[19]=0 \$MC_AUXFU_ASSIGN_VALUE[19]=52 \$MC_AUXFU_ASSIGN_GROUP[19]=13	; Beschreibung der 20.Hilfsfunktion: M52
\$MC_AUXFU_ASSIGN_TYPE[20]="M" \$MC_AUXFU_ASSIGN_EXTENSION[20]=0 \$MC_AUXFU_ASSIGN_VALUE[20]=53 \$MC_AUXFU_ASSIGN_GROUP[20]=13	; Beschreibung der 21.Hilfsfunktion: M53

12.18 Datenlisten

12.18.1 Maschinendaten

12.18.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10713	M_NO_FCT_STOPRE	M-Funktion mit Vorlaufstopp
10714	M_NO_FCT_EOP	M-Funktion für Spindel aktiv nach NC-RESET
10715	M_NO_FCT_CYCLE	Durch Unterprogramm zu ersetzende M-Funktion
11100	AUXFU_MAXNUM_GROUP_ASSIGN	Maximale Anzahl anwenderdefinierbarer Hilfsfunktionen pro Kanal
11110	AUXFU_GROUP_SPEC	Gruppenspezifisches Ausgabeverhalten
11450	SEARCH_RUN_MODE	Verhalten nach Satzsuchlauf

12.18.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20110	RESET_MODE_MASK	Festlegung der Steuerungsgrundstellung nach Teileprogramm-Start
20112	START_MODE_MASK	Festlegung der Steuerungs-Grundstellung nach Hochlauf und bei Reset bzw. Teileprogrammende
20270	CUTTING_EDGE_DEFAULT	Grundstellung der Werkzeugschneide ohne Programmierung
20800	SPF_END_TO_VDI	Unterprogrammende / Halt an PLC
22000	AUXFU_ASSIGN_GROUP	Gruppenzuordnung von anwenderdefinierten Hilfsfunktionen
22010	AUXFU_ASSIGN_TYPE	Typ von anwenderdefinierten Hilfsfunktionen
22020	AUXFU_ASSIGN_EXTENSION	Adresserweiterung für anwenderdefinierte Hilfsfunktionen
22030	AUXFU_ASSIGN_VALUE	Wert von anwenderdefinierten Hilfsfunktionen
22035	AUXFU_ASSIGN_SPEC	Ausgabeverhalten von anwenderdefinierten Hilfsfunktionen
22040	AUXFU_PREDEF_GROUP	Gruppenzuordnung von vordefinierten Hilfsfunktionen
22050	AUXFU_PREDEF_TYPE	Typ von vordefinierten Hilfsfunktionen
22060	AUXFU_PREDEF_EXTENSION	Adresserweiterung für vordefinierte Hilfsfunktionen
22070	AUXFU_PREDEF_VALUE	Wert von vordefinierten Hilfsfunktionen
22080	AUXFU_PREDEF_SPEC	Ausgabeverhalten von vordefinierten Hilfsfunktionen
22100	AUXFU_QUICK_BLOCKCHANGE	Satzwechselverzögerung bei schnellen Hilfsfunktionen

Nummer	Bezeichner: \$MC_	Beschreibung
22110	AUXFU_H_TYPE_INT	Typ von H-Hilfsfunktionen
22200	AUXFU_M_SYNC_TYPE	Ausgabezeitpunkt M-Funktionen
22210	AUXFU_S_SYNC_TYPE	Ausgabezeitpunkt S-Funktionen
22220	AUXFU_T_SYNC_TYPE	Ausgabezeitpunkt T-Funktionen
22230	AUXFU_H_SYNC_TYPE	Ausgabezeitpunkt H-Funktionen
22240	AUXFU_F_SYNC_TYPE	Ausgabezeitpunkt F-Funktionen
22250	AUXFU_D_SYNC_TYPE	Ausgabezeitpunkt D-Funktionen
22252	AUXFU_DL_SYNC_TYPE	Ausgabezeitpunkt DL-Funktionen
22254	AUXFU_ASSOC_M0_VALUE	Zusätzliche M-Funktion für Programm-Halt
22256	AUXFU_ASSOC_M1_VALUE	Zusätzliche M-Funktion für bedingten Halt
22530	TOCARR_CHANGE_M_CODE	M-Code bei Werkzeugträgerwechsel
22532	GEOAX_CHANGE_M_CODE	M-Code bei Umschaltung der Geometrieachsen
22534	TRAFO_CHANGE_M_CODE	M-Code bei Transformationswechsel
22560	TOOL_CHANGE_M_CODE	Hilfsfunktion für Werkzeugwechsel

A4: Digitale und analoge NC-Peripherie

13.1 Einleitung

Funktionen

An eine SINUMERIK 840D sl können über PROFIBUS oder PROFINET Peripheriebaugruppen angeschlossen werden. Auf die entsprechenden digitalen und analogen Ein- bzw. Ausgänge wird im Normalfall vom PLC-Anwenderprogramm aus zugegriffen. Die Funktion "Digitale und analoge NC-Peripherie für SINUMERIK 840D sl" ermöglicht über Systemvariablen oder Compile-Zyklen direkt von der NC aus (Teileprogramme, Synchronaktionen und Compile-Zyklen) auf die Ein-/Ausgänge der Peripheriebaugruppen zuzugreifen. Diese Peripherie wird im weiteren Verlauf als **NC-Peripherie** bezeichnet.

Aus Kompatibilitätsgründen stehen drei verschiedene Funktionen zur Verfügung:

1. Direkte Peripherie-Zugriffe ohne PLC (Seite 752)
Von der NC aus wird, unter Umgehung der PLC, direkt auf die steuerungsinternen Abbilder der Ein-/Ausgänge der Peripheriebaugruppen zugegriffen.
Dies ist sowohl vom Mengengerüst als auch von der Reaktionszeit die derzeit performanteste Funktion.
2. Direkte Peripherie-Zugriffe über PLC (Seite 746)
Von der NC aus werden die Anforderungen zum Lesen und Schreiben der Ein-/Ausgänge der Peripheriebaugruppen in die Schnittstelle zur PLC geschrieben. Anschließend wird Interrupt auf der PLC ausgelöst. Im Rahmen der Interrupt-Bearbeitung werden die Anforderungen von der PLC bearbeitet.
Vom Mengengerüst her ist die Funktion gleichwertig mit der unter 1. genannten Funktion. Die Reaktionszeit ist aber weniger performant.
3. Indirekte Peripherie-Zugriffe über PLC (Seite 724)
Von der NC aus werden die Anforderungen zum Lesen und Schreiben der Ein-/Ausgänge der Peripheriebaugruppen in die Schnittstelle zur PLC geschrieben. Die Anforderungen werden zyklisch im OB1-Takt bearbeitet.
Sowohl vom Mengengerüst als auch von der Reaktionszeit ist dies die am wenigsten performante Funktion.

Voraussetzungen

- Die PROFIBUS / PROFINET-Peripheriebaugruppen der NC-Peripherie müssen angeschlossen und betriebsbereit sein.
- Die HW-Konfiguration der PROFIBUS / PROFINET-Peripheriebaugruppen der NC-Peripherie wurde mit SIMATIC S7-Manager bzw. HW-Konfig durchgeführt und in die PLC geladen.

Überwachungen

Bezüglich der NC-E/A-Peripherie sind folgende Überwachungen aktiv:

- Hochlauf:
 - Überprüfung auf Übereinstimmung der PLC-seitig erkannten E/A-Peripherie mit der in den Maschinendaten parametrisierten NC-Peripherie.
- Zyklischer Betrieb:
 - Lebenszeichenüberwachung im Interpolatortakt
 - Baugruppen-Überwachung im Interpolatortakt
 - Temperatur-Überwachung

Im Fehlerfall wird das Signal DB10.DBX104.7 "NC-Ready" zurückgesetzt und ein Alarm angezeigt.

Verhalten bei Störungen

Die digitalen und analogen Ausgänge der NC-Peripherie werden bei Störungen (z. B. NC-Ready = 0), bei Fehlern in der NCU bzw. bei Spannungsausfall in den sicheren Zustand (0 V) geschaltet.

Anwendung

Die NC-Peripherie wird z.B. von folgenden NC-Funktionen verwendet:

- Mehrere Vorschubwerte oder Hilfsfunktionen pro Satz
- Schnellrückzug bei Fertigmaß
- Achsspezifisches Restweglöschen
- Programmverzweigungen
- Schneller NC-Start
- Analoge Messzange
- Wegschaltssignale
- Stanz-/Nibbelfunktionen
- Analogwertsteuerung

13.2 Indirekte Peripherie-Zugriffe über PLC

13.2.1 Kurzbeschreibung

Auf der SINUMERIK 840D sl **NCU** befinden sich drei digitale E/A-Schnittstellen (X122, X132 und X142).

Vier digitale Ein- und Ausgänge der Schnittstelle X142 stehen als sogenannte schnelle NC-Peripherie zur Verfügung. Diese können sowohl über das erste Adressbyte als auch über die Systemvariablen \$A_IN[1...4] und \$A_OUT[1...4] gelesen bzw. geschrieben werden.

An den PROFIBUS DP/MPI-Schnittstellen X126 und X136 können Peripherie-Baugruppen angeschlossen werden. Dadurch ist es möglich, das Mengengerüst der digitalen und analogen NC-Ein-/Ausgänge um jeweils **32 digitale** bzw. **8 analoge** Ein- und Ausgänge zu erweitern. Diese NC-Ein-/Ausgänge werden nachfolgenden als **externe** NC-Peripherie bezeichnet.

Tabelle 13-1 Maximale Anzahl an digitalen und analogen NC-Ein-/Ausgängen

	On-Board	NC-Peripherie	Gesamt
digitale Eingänge	4	32	36
digitale Ausgänge	4	32	36
analoge Eingänge	-	8	8
analoge Ausgänge	-	8	8

Weitere Informationen

- Gerätehandbuch SINUMERIK 840D sl
- Betriebsanleitung SIMATIC ET 200S FC

Siehe auch

Direkte Peripherie-Zugriffe über PLC (Seite 746)

13.2.2 Parametrierung

Maschinendaten

Anzahl der aktiven NC-E/A-Peripherie

Die Anzahl der aktiven bzw. von der NC nutzbaren **digitalen** NC-Ein-/Ausgangsbytes werden in folgenden Maschinendaten eingestellt:

- Anzahl der digitalen NC-Eingangsbytes
MD10350 \$MN_FASTIO_DIG_NUM_INPUTS = <Anzahl>
- Anzahl der digitalen NC-Ausgangsbytes
MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS = <Anzahl>

Die Anzahl der aktiven bzw. von der NC nutzbaren **analogen** Ein-/Ausgänge werden in folgenden Maschinendaten eingestellte:

- Anzahl der analogen NC-Eingänge
MD10300 \$MN_FASTIO_ANA_NUM_INPUTS = <Anzahl>
- Anzahl der analogen NC-Ausgänge
MD10310 \$MN_FASTIO_ANA_NUM_OUTPUTS = <Anzahl>

Die Ein-/Ausgänge müssen hardwaremäßig nicht vorhanden sein. In diesem Fall werden NC-intern die Signalzustände bzw. Analogwerte definiert auf "Null" gesetzt.

Vom PLC-Anwenderprogramm aus können die an den Peripherie-Eingängen anliegenden Werte verändert werden, bevor sie von der NC gelesen werden.

Slot-Adressen

Adressierung der **digitalen** Ein- / Ausgänge:

- HW-Zuordnung für externe digitale Eingänge
MD10366 \$MN_HW_ASSIGN_DIG_FASTIN[<n>] = <Adresse_H>
- HW-Zuordnung für externe digitale Ausgänge
MD10368 \$MN_HW_ASSIGN_DIG_FASTOUT[<n>] = <Adresse_H>

Adressierung der **analogen** Ein- / Ausgänge

- HW-Zuordnung für externe analoge Eingänge
MD10362 \$MN_HW_ASSIGN_ANA_FASTIN[<n>] = <Adresse_H>
- HW-Zuordnung für externe analoge Ausgänge
MD10364 \$MN_HW_ASSIGN_ANA_FASTOUT[<n>] = <Adresse_H>

<n>: Index zur Adressierung der externen **digitalen** E/A-Bytes (0 ... 3) bzw. der externen **analogen** Ein-/Ausgänge (0 ... 7)

<Adresse>: Slot-Adresse der PROFIBUS / PROFINET-Baugruppe mit 05 00 xxxx

05 Kennung für PROFIBUS / PROFINET-Baugruppe

00 fest vorgegeben

xxxx_H Hexadezimale logische Startadresse des Slots

0000 = kein aktiver Slot

Bei logischen Startadressen innerhalb des PLC-Prozessabbildes ist folgendes zu beachten:

- Eingangsslots: Lesen von der NC aus möglich
- Ausgangsslots: Schreiben von der NC aus verboten ⇒ Alarm nach Hochlauf der NC

Weitere Informationen

Gerätehandbuch NCU 7x0.3 PN, NCU 7x0.3B PN; Technische Daten
> PLC > Prozessabbildgröße

Ausführliche Informationen finden sich unter der Adresse (<http://support.automation.siemens.com/WW/view/de/54058408>)

Bewertungsfaktoren für analogen Ein-/Ausgänge

Mit dem Bewertungsfaktor kann für jeden einzelnen **analogen** NC-Ein-/Ausgang eine Anpassung an die AD- bzw. DA-Wandler der verwendeten Analog-Peripheriebaugruppe vorgenommen werden:

- Bewertungsfaktor für die analogen NC-Eingänge (siehe "Analoge NC-Eingänge (Seite 735)"):
MD10320 \$MN_FASTIO_ANA_INPUT_WEIGHT[<n>]
- Bewertungsfaktor für die analogen NC-Ausgänge (siehe "Analoge NC-Ausgänge (Seite 737)"):
MD10330 \$MN_FASTIO_ANA_OUTPUT_WEIGHT[<n>]

<n>: Index zur Adressierung der externen **analogen** Ein-/Ausgänge (0 ... 7)

Zuordnung zu NC-Funktionen

Bei mehreren NC-Funktionen wird E/A-Peripherie benötigt. Die Zuordnung der verwendeten Ein-/Ausgänge erfolgt funktionspezifisch über Maschinendaten, z. B. für die Funktion "Mehrere Vorschübe in einem Satz" über das Maschinendatum:

MD21220 \$MC_MULTFEED_ASSIGN_FASTIN = <Byte-Adresse>

<Byte-Adresse>	digitale NC-Ein-/Ausgänge	
0	keine	
1	1 ... 4	On-Board Ein-/Ausgänge
	5 ... 8	NC-Ausgang ohne Hardware
2	9 ... 16	Externe NC-Peripherie
3	17 ... 24	Externe NC-Peripherie
4	25 ... 32	Externe NC-Peripherie
5	33 ... 40	Externe NC-Peripherie
128	Eingänge 1 bis 8 von Komparator-Byte 1	
129	Eingänge 9 bis 16 von Komparator-Byte 2	

Hinweis

Mehrfachzuordnungen

Mehrfachzuordnungen von **Eingängen** werden nicht als Fehlparametrierung betrachtet.

Mehrfachzuordnungen von **Ausgängen** werden als Fehlparametrierung betrachtet, im Hochlauf überprüft und mit einem Alarm angezeigt.

Beispiel: Hardwarezuordnung von externer E/A-Peripherie

Zum Lesen / Schreiben von digitalen Ein-/Ausgängen einer PROFIBUS-Baugruppe durch die NC werden zwei Eingangs- und ein Ausgangsbyte parametrierung.

Anzahl der aktiven NC-Ein-/Ausgangsbytes

MD10350 \$MN_FASTIO_DIG_NUM_INPUTS ; 2 digitale Eingangs-Bytes + 1 On-Board Byte
 = 2 + 1 = 3

MD10360 \$MN_FASTIO_DIG_NUM_OUT-PUTS = 1 + 1 = 2 ; 1 digitales Ausgangs-Bytes + 1 On-Board Byte

Hardwarezuordnung

MD10366 \$MN_HW_ASSIGN_DIG_FASTIN[0] = 'H5000**200**' ; \$A_IN[9] ... [16] aus 1. Peripherie-Eingangsbyte

MD10366 \$MN_HW_ASSIGN_DIG_FASTIN[1] = 'H5000**201**' ; \$A_IN[17] ... [24] aus 2. Peripherie-Eingangsbyte

MD10368 \$MN_HW_ASSIGN_DIG_FASTOUT[0] = 'H5000**200**' ; \$A_OUT[9] ... [16] in 1. Peripherie-Ausgangsbyte

Die in den Maschinendaten eingetragenen hexadezimalen Adressen 200_H und 201_H entsprechen den bei der Projektierung im SIMATIC S7-Manager vergebenen, dezimalen logischen Basisadresse 512_D und 513_D

13.2.3 Systemvariablen

Eingangsdaten

Systemvariable	Index bzw. Eingangsnummer <n>
\$A_IN[<n>]	1 ... 4 und 9 ... 40, siehe Digitale NC-Eingänge (Seite 728)
\$A_INA[<n>]	1 ... 8, siehe Analoge NC-Eingänge (Seite 735)

Beim Lesen von Eingangsdaten aus einem Teileprogramm wird im Kanal ein Vorlaufstopp ausgelöst.

Ausgangsdaten

Systemvariable	Index bzw. Ausgangsnummer <n>
\$A_OUT[<n>]	1 ... 4 und 9 ... 40, siehe Digitale NC-Ausgänge (Seite 730)
\$A_OUTA[<n>]	1 ... 8, siehe Analoge NC-Ausgänge (Seite 737)

Beim Lesen von Ausgangsdaten aus einem Teileprogramm wird im Kanal ein Vorlaufstopp ausgelöst.

13.2.4 Komparator-Eingänge

Zusätzlich zu den Peripherie-Eingängen stehen 16 Steuerungs-interne Komparator-Eingänge zur Verfügung.

Der aktuelle Signalzustand eines Komparator-Eingangs ergibt sich aus einem Vergleich eines analogen Peripherie-Eingangs mit einem in einem Settingdatum vorgegebenem Schwellwert.

Siehe "Komparator-Eingänge (Seite 742)".

13.2.5 Digitale NC-Ein-/Ausgänge

13.2.5.1 Digitale NC-Eingänge

Funktion

Über die Systemvariablen \$A_IN können die Werte der digitalen NC-Eingänge in einem NC-Programm oder Synchronaktion gelesen werden. Über NC/PLC-Nahtstellensignale kann der gelesene Wert beeinflusst werden.

Anwendungsbeispiele

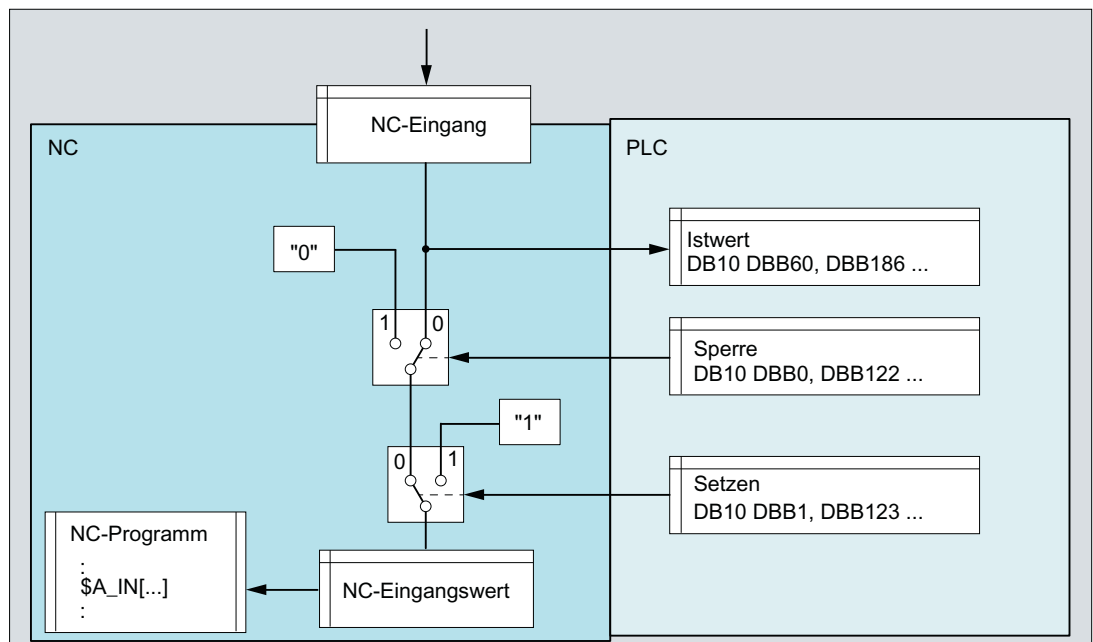
Digitale NC-Eingänge werden z. B. für folgende NC-Funktionen verwendet:

- Restweglöschen bei Positionierachsen
- Schnelle Programmverzweigungen am Satzende
- Programmierte Einlesesperre
- Mehrere Vorschübe in einem Satz

Weitere Informationen

Funktionshandbuch Synchronaktionen

NC/PLC-Nahtstellensignale



Istwert

Über die Istwert-Schnittstelle kann im PLC-Anwenderprogramm der aktuelle Wert des NC-Eingangs gelesen werden.

Hinweis

Unterschiedliche Werte

Der in der Nahtstelle "Istwert" vorliegende Wert kann aufgrund der verschiedenen nachfolgenden Einflussmöglichkeiten unterschiedlich zum NC-Eingangswert sein, der über die Systemvariable \$A_IN gelesen wird.

Sperre

Wird in der Nahtstelle ein Bit gesetzt, wird für den entsprechenden Eingang der Wert 0 weitergegeben.

Setzen

Wird in der Nahtstelle ein Bit gesetzt, wird für den entsprechenden Eingang der Wert 1 weitergegeben.

Übersicht

Sperre	Setzen	Istwert	NC-Eingang
DBX0.0 - 7	DBX1.0 - 7	DBX60.0 - 7	1 - 8
DBX122.0 - 7	DBX123.0 - 7	DBX186.0 - 7	9 - 16
DBX124.0 - 7	DBX125.0 - 7	DBX187.0 - 7	17 - 24
DBX126.0 - 7	DBX127.0 - 7	DBX188.0 - 7	25 - 32
DBX128.0 - 7	DBX129.0 - 7	DBX189.0 - 7	33 - 40

Randbedingungen

Verhalten nach Warmstart und Kanal-Reset

Nach Warmstart und Kanal-Reset wird der Wert am Hardware-Eingang unverändert übertragen.

13.2.5.2 Digitale NC-Ausgänge

Funktion

Über die Systemvariablen \$A_OUT können in einem NC-Programm oder Synchronaktion die Werte der digitalen NC-Ausgänge geschrieben werden. Über NC/PLC-Nahtstellensignale kann der geschriebene Wert beeinflusst werden.

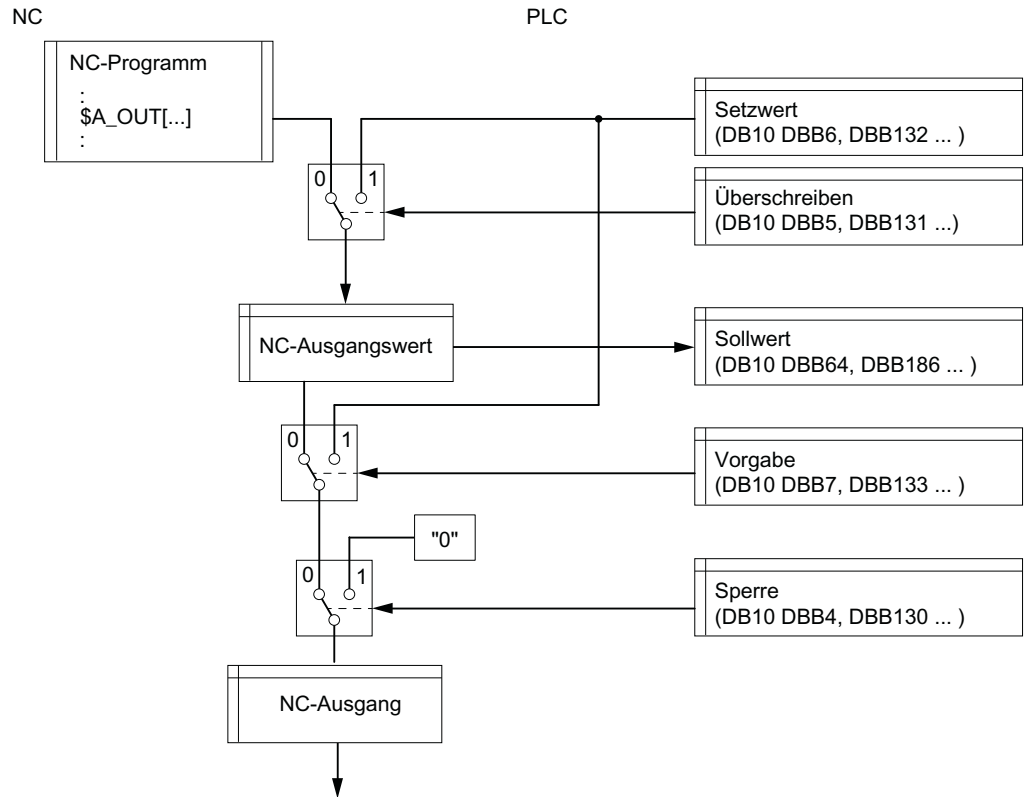
Anwendungsbeispiele

- Wegschaltsignale (siehe Funktionshandbuch *Achsen und Spindeln*, Kapitel *Softwareknocken*)
- Ausgabe der Komparatorsignale

Weitere Informationen

Funktionshandbuch Synchronaktionen

NC/PLC-Nahtstellensignale



Überschreiben

Wird in der Nahtstelle ein Bit gesetzt, wird für den entsprechenden Ausgang statt dem mit der Systemvariablen \$A_OUT geschriebene Wert der vom PLC-Anwenderprogramm vorgegebene Setzwert verwendet. Der über die Systemvariablen \$A_OUT geschriebene Wert geht dabei verloren.

Wird in der Nahtstelle ein Bit zurückgesetzt, bleibt der aktuelle Wert am Hardware-Ausgang erhalten.

Setzwert

Über den Setzwert kann vom PLC-Anwenderprogramm ein definierter Ausgangswert vorgegeben werden. Damit der Setzwert wirksam wird, muss er über die Nahtstellen für "Überschreiben" oder "Vorgabe" aktiviert werden.

Sollwert

Über den Sollwert kann im PLC-Anwenderprogramm der aktuelle NC-Ausgangswert gelesen werden.

Hinweis

Unterschiedliche Werte

Der in der Nahtstelle "Sollwert" vorliegende Wert kann aufgrund der verschiedenen nachfolgenden Einflussmöglichkeiten unterschiedlich zum Wert sein, der am NC-Ausgang anliegt.

Vorgabe

Wird in der Nahtstelle ein Bit gesetzt, wird für den entsprechenden Ausgang statt dem NC-Ausgangswert der vom PLC-Anwenderprogramm vorgegebene Setzwert verwendet. Der aktuelle NC-Ausgangswert bleibt dabei erhalten.

Wird in der Nahtstelle ein Bit zurückgesetzt, wird wieder der letzte NC-Ausgangswert wirksam.

Sperre

Wird in der Nahtstelle ein Bit gesetzt, wird für den entsprechenden Ausgang der Wert 0 ausgegeben.

Übersicht

Sperre	Überschreiben	Setzwert	Vorgabe	Sollwert	NC-Ausgang
DBX4.0 - 7	DBX5.0 - 7	DBX6.0 - 7	DBX7.0 - 7	DBX64.0 - 7	1 - 8
DBX130.0 - 7	DBX131.0 - 7	DBX132.0 - 7	DBX133.0 - 7	DBX190.0 - 7	9 - 16
DBX134.0 - 7	DBX135.0 - 7	DBX136.0 - 7	DBX137.0 - 7	DBX191.0 - 7	17 - 24
DBX138.0 - 7	DBX139.0 - 7	DBX140.0 - 7	DBX141.0 - 7	DBX192.0 - 7	25 - 32
DBX142.0 - 7	DBX143.0 - 7	DBX144.0 - 7	DBX145.0 - 7	DBX193.0 - 7	33 - 40

Randbedingungen

NC-Ausgänge ohne Hardware

Werden NC-Ausgänge beschrieben, die definiert wurden (MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS), Hardware-mäßig aber nicht vorhanden sind, wird kein Alarm angezeigt. Der Sollwert kann über das PLC-Anwenderprogramm gelesen werden.

Verhalten bei Programmende-Reset / Kanal-Reset

Vom PLC-Anwenderprogramm aus kann über die Nahtstellen für "Überschreiben", "Vorgabe" oder "Sperre" jeder NC-Ausgang bei Programmende- oder Kanal-Reset anwendungsspezifisch definiert gesetzt werden.

Verhalten bei Warmstart

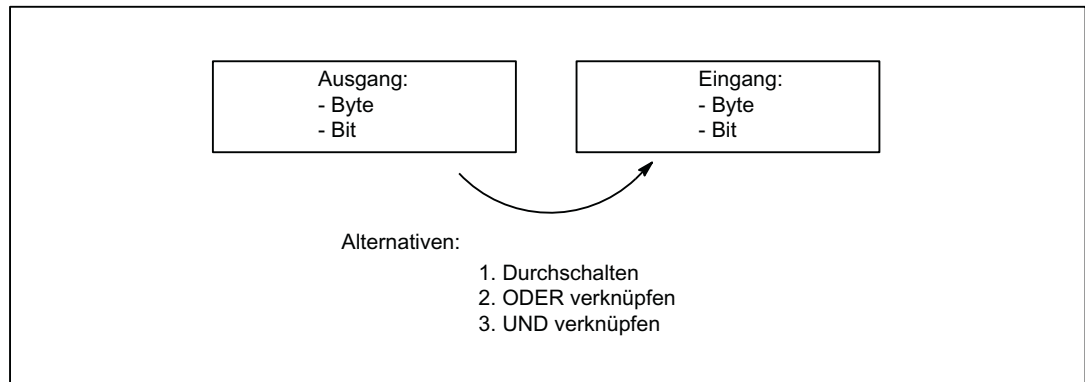
Nach Warmstart werden alle NC-Ausgänge definiert auf "0" gesetzt. Vom PLC-Anwenderprogramm aus kann über die Überschreib- oder Vorgabemaske jeder NC-Ausgang anwendungsspezifisch gesetzt werden.

13.2.5.3 Durchschalten und Verknüpfen von schnellen digitalen Ein-/Ausgängen

Funktion

Schnelle Eingänge der NC-Peripherie können abhängig von Signalzuständen schneller Ausgänge softwaremäßig gesetzt werden.

Übersicht:



Durchschalten

Der schnelle Eingang der NC-Peripherie wird auf den Signalzustand gesetzt, den der zugeordnete schnelle Ausgang hat.

ODER-Verknüpfung

Der schnelle Eingang der NC-Peripherie nimmt den Signalzustand ein, der sich aus der ODER-Verknüpfung des Ausgangssignals mit dem zugeordneten Eingangssignal ergibt.

UND-Verknüpfung

Der schnelle Eingang der NC-Peripherie nimmt den Signalzustand ein, der sich aus der UND-Verknüpfung des Ausgangssignals mit dem zugeordneten Eingangssignal ergibt.

Sonderfälle

- Werden mehrere Ausgangsbits dem gleichen Eingangsbit zugeordnet, so wird die Vorgabe mit dem höchsten MD-Index wirksam.
- Werden Eingänge oder Ausgänge angegeben, die nicht vorhanden oder nicht aktiviert sind, wird die Zuordnung ohne Alarm ignoriert. Die Überprüfung der aktivierten Bytes der NC-Peripherie erfolgt anhand der Einträge in den beiden folgenden Maschinendaten:
MD10350 \$MN_FASTIO_DIG_NUM_INPUTS
MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS.

Zuordnungen definieren

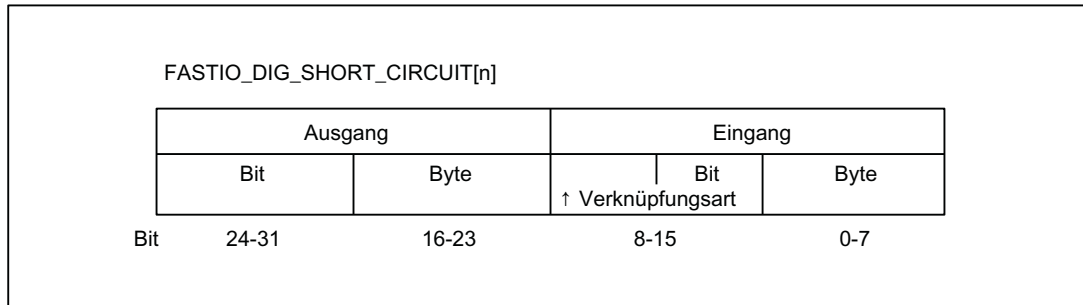
Die Zuordnungen werden angegeben durch das Maschinendatum:
 MD10361 \$MN_FASTIO_DIG_SHORT_CIRCUIT[n]

n: kann Werte 0 bis 9 annehmen, es sind also bis zu **10** Zuordnungen angebbbar.

Je 2 Hexa-Zeichen sind für die Angabe von Byte und Bit eines Ausgangs und eines Eingangs vorgesehen.

Durch Angabe von 0, A und B in Bit 12 - 15 des Einganges wird die **Verknüpfungsart** angegeben:

- 0 Durchschalten
- A UND-Verknüpfung
- B ODER-Verknüpfung



Beispiele

Durchschalten:

MD10361 \$MN_FASTIO_DIG_SHORT_CIRCUIT = '04010302H'

Ausgang 4, Byte 1, durchschalten auf

Eingang 3, Byte 2

UND-Verknüpfung:

MD10361 \$MN_FASTIO_DIG_SHORT_CIRCUIT = '0705A201H'

Ausgang 7, Byte 5 UND-verknüpfen mit

Eingang 2, Byte 1

ODER-Verknüpfung:

MD10361 \$MN_FASTIO_DIG_SHORT_CIRCUIT = '0103B502H'

Ausgang1, Byte 3 ODER-verknüpfen mit

Eingang 5, Byte 2

13.2.6 Analoge NC-Ein-/Ausgänge

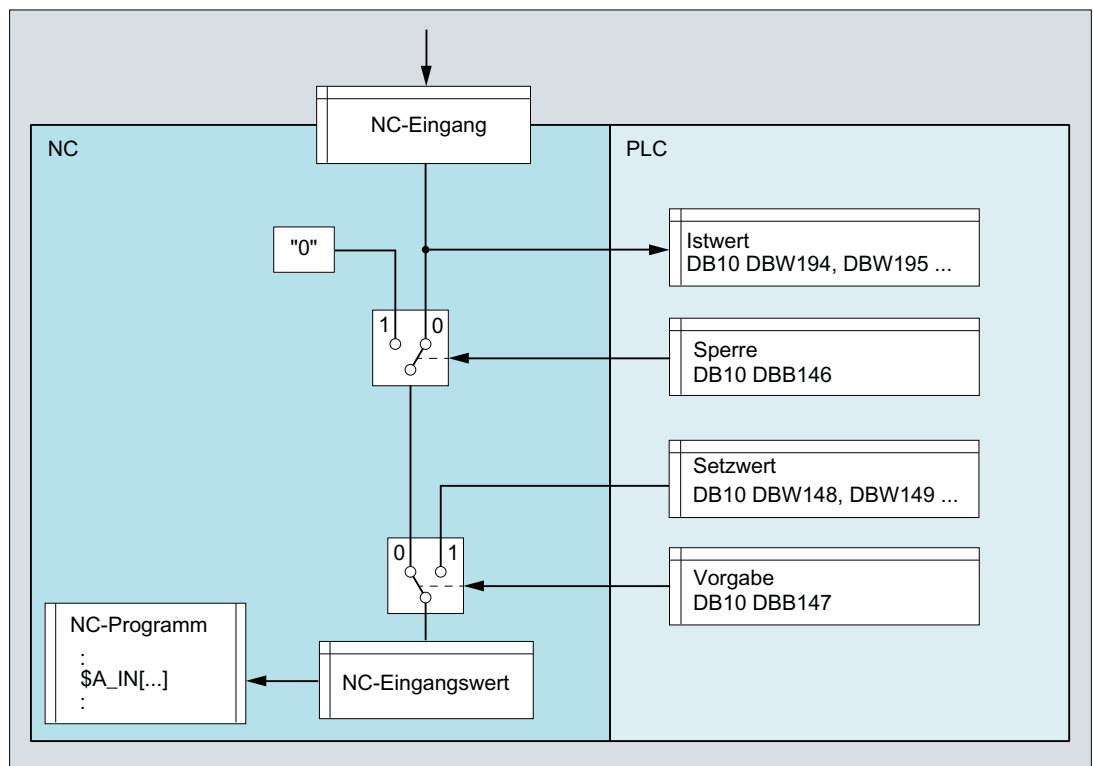
13.2.6.1 Analoge NC-Eingänge

Funktion

Über die Systemvariablen `$A_INA` können die Werte der analogen NC-Eingänge in einem NC-Programm oder Synchronaktion gelesen werden. Über NC/PLC-Nahtstellensignale kann der gelesene Wert beeinflusst werden.

Binäre Analogwertdarstellung

NC/PLC-Nahtstellensignale



Istwert

Über die Istwert-Schnittstelle kann im PLC-Anwenderprogramm der aktuelle Wert des NC-Eingangs gelesen werden.

Der Istwert wird als Festpunktzahl (16 Bitwert einschließlich Vorzeichen) im 2er-Komplement dargestellt.

Siehe "Darstellung der analogen Ein-/Ausgabewerte (Seite 740)".

Hinweis

Unterschiedliche Werte

Der in der Nahtstelle "Istwert" vorliegende Wert kann aufgrund der verschiedenen nachfolgenden Einflussmöglichkeiten unterschiedlich zum NC-Eingangswert sein, der über die Systemvariable \$A_INA gelesen wird.

Sperre

Wird in der Nahtstelle ein Bit gesetzt, wird für den entsprechenden Eingang der Wert 0 weitergegeben.

Setzwert

Über den Setzwert kann vom PLC-Anwenderprogramm ein definierter Eingangswert vorgegeben werden. Damit der Setzwert wirksam wird, muss er über die Nahtstelle "Vorgabe" aktiviert werden.

Der Setzwert muss als Festpunktzahl (16 Bitwert einschließlich Vorzeichen) im 2er-Komplement vorgegeben werden.

Siehe "Darstellung der analogen Ein-/Ausgabewerte (Seite 740)".

Vorgabe

Wird in der Nahtstelle ein Bit gesetzt, wird für den entsprechenden Eingang der zugehörige "Setzwert" weitergegeben.

Übersicht

Sperre	Vorgabe	Setzwert	Istwert	NC-Eingang
DBX146.0 - 7	DBX147.0 - 7	DBW148 - 162	DBW194 - 209	1 - 8

Maschinendaten

Bewertungsfaktor

Mit dem Bewertungsfaktor kann für jeden analogen NC-Eingang eine Anpassung an die DA-Wandler der verwendeten Peripherie-Baugruppe parametrieren werden:

MD10320 \$MN_FASTIO_ANAINTPUT_WEIGHT[<Ausgang>]

Hinweis

Analoge NC-Eingänge ohne Hardware

Bei einem Bewertungsfaktor von 32767 sind die digitalisierten Analogwerte für NC-Programm und PLC-Anwenderprogramm gleich. Dadurch kann der NC-Ausgang für eine 1:1-Kommunikation zwischen NC-Programm und PLC-Anwenderprogramm verwendet werden.

Randbedingungen

Analoge NC-Eingänge ohne Hardware

Werden NC-Ausgänge beschrieben, die definiert wurden (MD10300 \$MN_FASTIO_ANA_NUM_INPUTS), Hardware-mäßig aber nicht vorhanden sind, wird kein Alarm angezeigt. Der Istwert kann über das PLC-Anwenderprogramm gelesen werden.

Verhalten bei Warmstart, Programmende-Reset und Kanal-Reset

Nach Warmstart, Programmende-Reset oder Kanal-Reset wird für alle NC-Eingänge der anliegende Analogwert weitergegeben. Vom PLC-Anwenderprogramm aus kann jeder NC-Eingang anwendungsspezifisch gesetzt werden.

13.2.6.2 Analoge NC-Ausgänge

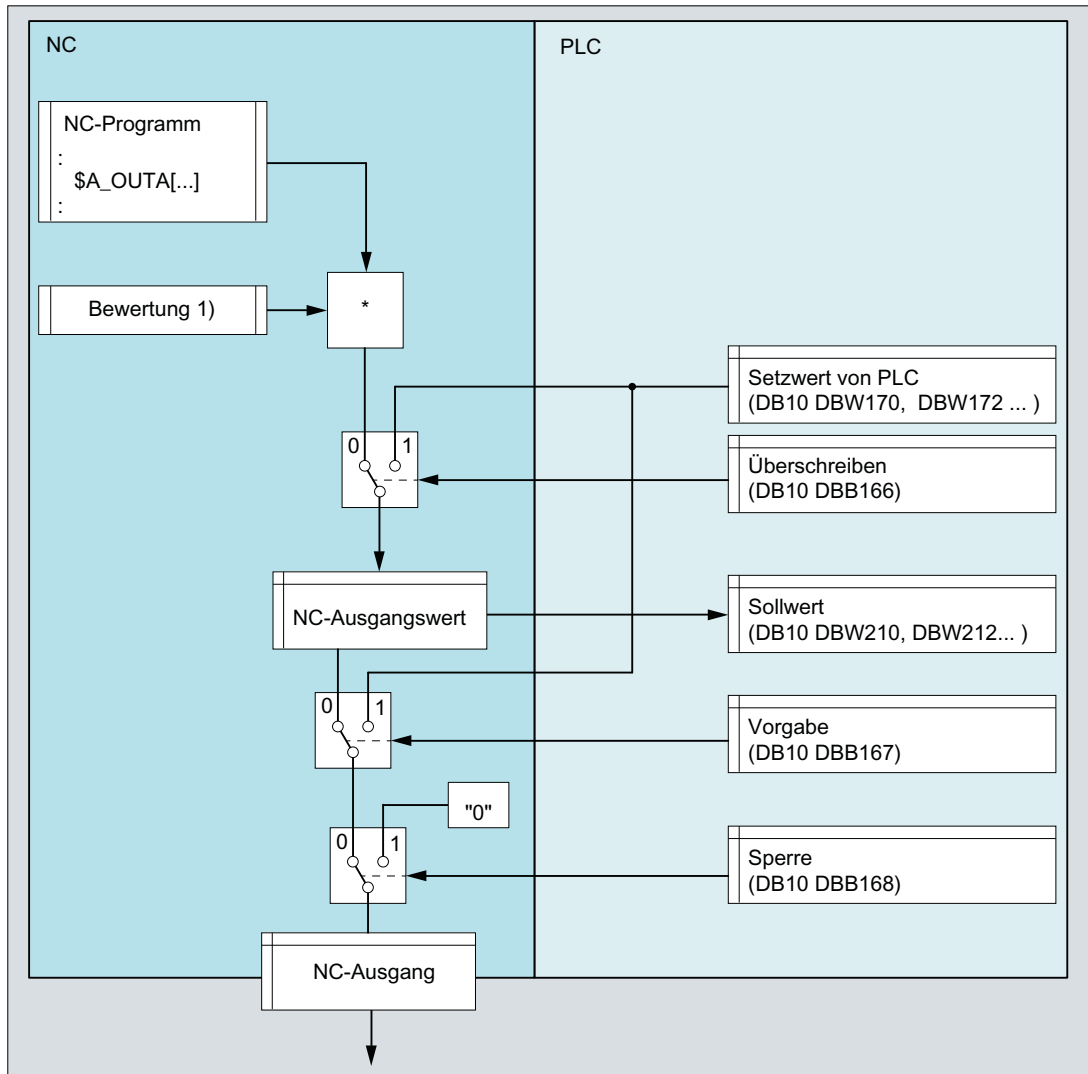
Funktion

Über die Systemvariablen \$A_OUTA können in einem NC-Programm oder Synchronaktion die Werte der analogen NC-Ausgänge geschrieben werden. Über Maschinendaten und NC/PLC-Nahtstellensignale kann der geschriebene Wert beeinflusst werden.

Weitere Informationen

Funktionshandbuch Synchronaktionen

NC/PLC-Nahtstellensignale



1) MD10330 \$MN_FASTIO_ANA_OUTPUT_WEIGHT[<Ausgang>]

Setzwert

Über den Setzwert kann vom PLC-Anwenderprogramm ein definierter Ausgangswert vorgegeben werden. Damit der Setzwert wirksam wird, muss er über die Nahtstellen für "Überschreiben" oder "Vorgabe" aktiviert werden.

Der Setzwert muss als Festpunktzahl (16 Bitwert einschließlich Vorzeichen) im 2er-Komplement vorgegeben werden.

Siehe "Darstellung der analogen Ein-/Ausgabewerte (Seite 740)".

Überschreiben

Wird ein Bit gesetzt, wird für den entsprechenden Ausgang statt dem mit der Systemvariablen \$A_OUTA geschriebene Wert, der vom PLC-Anwenderprogramm vorgegebene Setzwert verwendet. Der über die Systemvariablen \$A_OUTA geschriebene Wert geht dabei verloren.

Wird ein Bit zurückgesetzt, bleibt für den entsprechenden Ausgang der aktuelle Wert am Hardware-Ausgang erhalten.

Vorgabe

Wird ein Bit gesetzt, wird für den entsprechenden Ausgang statt dem NC-Ausgangswert der vom PLC-Anwenderprogramm vorgegebene Setzwert verwendet. Der aktuelle NC-Ausgangswert bleibt dabei erhalten.

Wird ein Bit zurückgesetzt, wird für den entsprechenden Ausgang wieder der letzte NC-Ausgangswert wirksam.

Sperrern

Wird in der Nahtstelle ein Bit gesetzt, wird am entsprechenden NC-Ausgang der Wert 0 Volt ausgegeben.

Sollwert

Über den Sollwert kann im PLC-Anwenderprogramm der aktuelle NC-Ausgangswert gelesen werden.

Der Sollwert wird als Festpunktzahl (16 Bitwert einschließlich Vorzeichen) im 2er-Komplement dargestellt.

Siehe "Darstellung der analogen Ein-/Ausgabewerte (Seite 740)".

Hinweis

Unterschiedliche Werte

Der in der Nahtstelle "Sollwert" vorliegende Wert kann aufgrund der verschiedenen nachfolgenden Einflussmöglichkeiten unterschiedlich zum Wert sein, der am NC-Ausgang anliegt.

Übersicht

Sperre	Überschreiben	Setzwert	Vorgabe	Sollwert	NC-Ausgang
DBX168.0 - 7	DBX166.0 - 7	DBW170 - 184	DBX167.0 - 7	DBW210 - 225	1 - 8

Maschinendaten

Bewertungsfaktor

Mit dem Bewertungsfaktor kann für jeden analogen NC-Ausgang eine Anpassung an die DA-Wandler der verwendeten Peripherie-Baugruppe parametrisiert werden:

MD10330 \$MN_FASTIO_ANA_OUTPUT_WEIGHT[<Ausgang>]

Programmcode	Kommentar
; Annahme: Analogwertbereich ± 10 V	
; Parametrierung: MD10330 \$MN_FASTIO_ANA_OUTPUT_WEIGHT[<Ausgang>] = 10000	
\$A_OUTA[<Ausgang>] = 9500	; NC-Ausgang: 9,5 V
\$A_OUTA[<Ausgang>] = -4120	; NC-Ausgang: -4,12 V

Hinweis

Analoge NC-Ausgängen ohne Hardware

Bei einem Bewertungsfaktor von 32767 sind die digitalisierten Analogwerte für NC-Programm und PLC-Anwenderprogramm gleich. Dadurch kann der NC-Ausgang für eine 1:1-Kommunikation zwischen NC-Programm und PLC-Anwenderprogramm verwendet werden.

Randbedingungen

Analoge NC-Ausgänge ohne Hardware

Werden NC-Ausgänge beschrieben, die definiert wurden (MD10310 \$MN_FASTIO_ANA_NUM_OUTPUTS), Hardware-mäßig aber nicht vorhanden sind, wird kein Alarm angezeigt. Der Sollwert kann über das PLC-Anwenderprogramm gelesen werden.

Verhalten bei Warmstart

Nach Warmstart werden alle NC-Ausgänge definiert auf "0" gesetzt. Vom PLC-Anwenderprogramm aus kann jeder NC-Ausgang anwendungsspezifisch gesetzt werden.

Verhalten bei Programmende-Reset / Kanal-Reset

Vom PLC-Anwenderprogramm aus kann über die Nahtstellen für "Überschreiben", "Vorgabe" oder "Sperrung" jeder NC-Ausgang bei Programmende- oder Kanal-Reset anwendungsspezifisch definiert gesetzt werden.

13.2.6.3 Darstellung der analogen Ein-/Ausgabewerte

Die Darstellung der digitalisierten Analogwerte an der NC/PLC-Nahtstelle erfolgt als Festpunktzahl (16 Bit einschließlich Vorzeichen) im 2er-Komplement.

Bitnummer	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Wertigkeit	VZ	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰

VZ: Vorzeichen

Minimalwert	Maximalwert
-32768 _D	32767 _D
8000 _H	7FFF _H

Schrittweite

Die Schrittweite beträgt bei einer Auflösung von 16 Bit und einem Nennbereich von ±10 V:

$$20 \text{ V} / 2^{16} = 20 \text{ V} / 65536 \approx 0,305 \text{ mV}$$

Auflösungen < 16 Bit

Ist die Auflösung einer Analogbaugruppe kleiner als 16 Bit einschließlich Vorzeichen, wird der digitalisierte Analogwert ausgehend von Bit 14 in die Schnittstelle eingetragen. Die nicht besetzten niederwertigen Stellen werden mit "0" aufgefüllt.

14 Bit-Auflösung

Bei einer Auflösung von 14 Bit inklusive Vorzeichen und einem Nennbereich von ±10 V beträgt die Schrittweite:

$$20 \text{ V} / 2^{14} = 20 \text{ V} / 16384 \approx 1,22 \text{ mV}$$

Bit 0 ... 1 sind immer "0".

12 Bit-Auflösung

Bei einer Auflösung von 12 Bit inklusive Vorzeichen und einem Nennbereich von ±10 V beträgt die Schrittweite:

$$20 \text{ V} / 2^{12} = 20 \text{ V} / 4096 \approx 4,88 \text{ mV}$$

Bit 0 ... 3 sind immer "0".

Darstellung des Maximalwerts bei unterschiedlichen Auflösungen

Bitnummer	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Wertigkeit der Bits	VZ	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
16-Bit-Auflösung: 32767 _D = 7FFF _H	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14-Bit- Auflösung: 8191 _D = 1FFF _H	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
12-Bit- Auflösung: 2047 _D = 7FF _H	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

Hinweis

Die Daten (Auflösung, Nennbereich) der verwendeten analogen Ein-/Ausgabebaugruppe sind der Dokumentation der jeweiligen Baugruppe zu entnehmen.

Beispiele

Digitale Analogwertdarstellung bei einer Auflösung von 14 Bit inklusive Vorzeichen und einem Nennbereich von ±10 V.

Beispiel 1: Analogwert = 9,5 V

digitalisierter Analogwert (dezimal):	9,5 V / 20 V * 16384 = 7782
digitalisierter Analogwert 14 Bit (binär):	01 1110 0110 0110
digitalisierter Analogwert 16 Bit (binär):	0111 1001 1001 1000
digitalisierter Analogwert 16 Bit (hex.):	7998 _H

Beispiel 2: Analogwert = -4,12 V

digitalisierter Analogwert (dezimal):	-4,12 V / 20 V * 16384 = -3375
digitalisierter Analogwert 14 Bit (binär):	11 0010 1101 0001

digitalisierter Analogwert 16 Bit (binär): 1100 1011 0100 0100
digitalisierter Analogwert 16 Bit (hex.): CB44_H

13.2.7 Komparator-Eingänge

Funktion

Zusätzlich zu den digitalen und analogen NC-Eingängen stehen noch 2 interne Komparator-Eingangsbites mit je 8 Komparator-Eingängen zur Verfügung. Der Signalzustand der Komparator-Eingänge wird durch den Vergleich zwischen den an den schnellen Analogeingängen anliegenden Analogwerten mit in Settingdaten parametrierbaren Schwellwerten gebildet.

Mit der Systemvariablen \$A_INCO[<n>] kann der Signalzustand (bzw. das Ergebnis des Vergleichs) des Komparator-Eingangs <n> direkt im Teileprogramm abgefragt werden.

Für Index <n> gilt:

<n> = 1 ... 8 für Komparator-Byte 1
<n> = 9 ... 16 für Komparator-Byte 2

Begriffe

In der vorliegenden Beschreibung werden die Begriffe **Komparator-Eingänge** (mit Index <n>; Wertebereich von <n>: 1 ... 8 bzw. 9 ... 16) und **Komparator-Eingangsbits** (mit Index ; Wertebereich von : 0 ... 7) verwendet.

Dabei gelten folgende Zusammenhänge:

Für <n> = 1 ... 8: Komparator-Eingang <n> entspricht Komparator-Eingangsbite = <n> - 1
Für <n> = 9 ... 16: Komparator-Eingang <n> entspricht Komparator-Eingangsbite = <n> - 9

Beispiel: Komparator-Eingang 1 entspricht Komparator-Eingangsbite 0.

Zuordnung der Analogeingänge

Mit dem folgenden Maschinendatum wird dem Eingangsbit des Komparator-Bytes 1 ein Analogeingang zugeordnet:

MD10530 \$MN_COMPAR_ASSIGN_ANA_INPUT_1[]

Beispiel:

MD10530 \$MN_COMPAR_ASSIGN_ANA_INPUT_1[0] = 1

MD10530 \$MN_COMPAR_ASSIGN_ANA_INPUT_1[1] = 1

MD10530 \$MN_COMPAR_ASSIGN_ANA_INPUT_1[7] = 7

Analogeingang 1 wirkt auf Eingangsbit 0 und 1 des Komparatorbytes 1.

Analogeingang 7 wirkt auf Eingangsbit 7 des Komparatorbytes 1.

Die Zuordnung der Analogeingänge für das Komparator-Byte 2 erfolgt mit dem Maschinendatum:

MD10531 \$MN_COMPAR_ASSIGN_ANA_INPUT_2[]

Komparator-Einstellungen

Die Einstellungen für die einzelnen Bits (0 bis 7) von Komparatorbyte 1 bzw. 2 werden parametrierbar über das Maschinendatum:

MD10540 \$MN_COMPAR_TYPE_1 (Parametrierung für Komparatorbyte 1)

bzw.

MD10541 \$MN_COMPAR_TYPE_2 (Parametrierung für Komparatorbyte 2)

Folgende Einstellungen sind möglich:

- Vergleichstyp-Maske (Bit 0 ... 7)
Für jedes Komparator-Eingangsbit wird die Art der Vergleichsbedingung festgelegt:
 - Bit = 1: Zugehöriges Komparator-Eingangsbit wird auf "1" gesetzt, wenn:
Analogwert \geq Schwellwert
 - Bit = 0: Zugehöriges Komparator-Eingangsbit wird auf "0" gesetzt, wenn:
Analogwert $<$ Schwellwert
- Ausgabe des Komparator-Eingangsbytes über digitale NC-Ausgänge (Bit 16 ... 23)
Die Komparatorbits können zusätzlich über die digitalen NC-Ausgänge byteweise direkt ausgegeben werden. Dazu muss in diesem Byte (Bit 16 ... 23) angegeben werden, welches digitale NC-Ausgangsbyte dafür verwendet wird.
 - Byte = 0: keine Ausgabe über digitale NC Ausgänge
 - Byte = 1: Ausgabe über digitale Onboard-NC-Ausgänge 9 ... 16
 - Byte = 2: Ausgabe über externe digitale NC-Ausgänge 17 ... 24
 - Byte = 3: Ausgabe über externe digitale NC-Ausgänge 25 ... 32
 - Byte = 4: Ausgabe über externe digitale NC- Ausgänge 33 ... 40
- Invertiermaske für die Ausgabe des Komparator-Eingangsbytes (Bit 24 ... 31)
Für jedes Komparatorsignal kann zusätzlich festgelegt werden, ob der an den digitalen NC-Ausgang auszugebende Signalzustand invertiert werden soll.
 - Bit = 1: Zugehöriges Komparator-Eingangsbit wird nicht invertiert.
 - Bit = 0: Zugehöriges Komparator-Eingangsbit wird invertiert.

Schwellwerte

Die beim Komparatorbyte 1 bzw. 2 für den Vergleich verwendeten Schwellwerte sind als Settingdaten zu hinterlegen. Für jedes Komparator-Eingangsbit (mit = 0 ... 7) ist ein eigener Schwellwert einzutragen:

SD41600 \$SN_COMPAR_THRESHOLD_1[]

bzw.

SD41601 \$SN_COMPAR_THRESHOLD_2[]

Komparatorsignale als digitale NC-Eingänge

Alle NC-Funktionen, deren Ablauf abhängig von digitalen NC-Eingängen bestimmt wird, können auch von den Signalzuständen der Komparatoren gesteuert werden. Dabei ist in dem der NC-Funktion zugehörigen Maschinendatum ("Zuordnung des verwendeten Hardware-Bytes") die Byteadresse für das Komparatorbyte 1 (HW-Byte 128) oder 2 (HW-Byte 129) einzutragen.

Beispiel:

NC-Funktion "Mehrere Vorschübe in einem Satz".

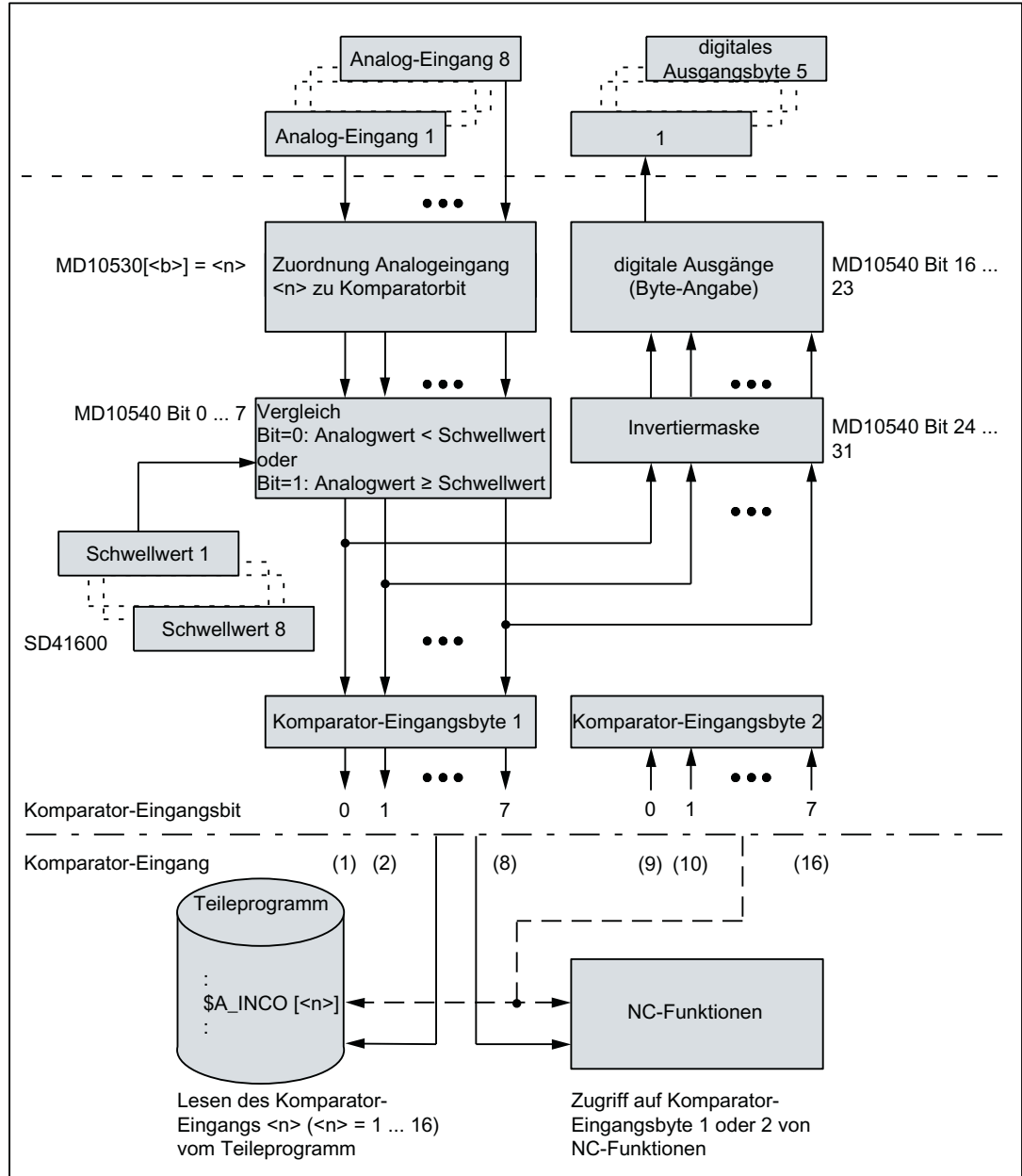
Eintrag im kanalspezifischen Maschinendatum:

MD21220 \$MC_MULTFEED_ASSIGN_FASTIN = 129

Damit werden abhängig vom Zustand des Komparatorbytes 2 verschiedene Vorschubwerte aktiviert.

Funktionsablauf

In der folgenden Abbildung ist der Funktionsablauf für Komparator-Eingangsbyte 1 schematisch dargestellt.



13.3 Direkte Peripherie-Zugriffe über PLC

13.3.1 Parametrierung

Maschinendaten

Länge der E/A-Bereiche

- Anzahl der PLC-Peripherie Eingang-Bytes, die von der NC direkt gelesen werden:
MD10394 \$MN_PLCIO_NUM_BYTES_IN
- Anzahl der PLC-Peripherie Ausgang-Bytes, die von der NC direkt beschrieben werden:
MD10396 \$MN_PLCIO_NUM_BYTES_OUT

Logische Anfangsadressen

- Logische Anfangsadresse ab der Daten von der PLC-Eingangs-Peripherie gelesen werden. Der bei der späteren Adressierung verwendete Offset, z.B. \$A_PBB_IN[< offset >], bezieht sich auf die im Maschinendatum festgelegte Anfangsadresse:
MD10395 \$MN_PLCIO_LOGIC_ADDRESS_IN
- Logische Anfangsadresse ab der Daten auf die PLC-Eingangs-Peripherie geschrieben werden. Der bei der späteren Adressierung verwendete Offset, z.B. \$A_PBB_OUT[< offset >], bezieht sich auf die im Maschinendatum festgelegte Anfangsadresse:
MD10397 \$MN_PLCIO_LOGIC_ADDRESS_OUT

Aktualisierungszeit

- Zeitdauer, innerhalb der die über \$A_PBx_IN lesbaren Daten aktualisiert werden (siehe unten Absatz "Übertragungszeitpunkte"):
MD10398 \$MN_PLCIO_IN_UPDATE_TIME

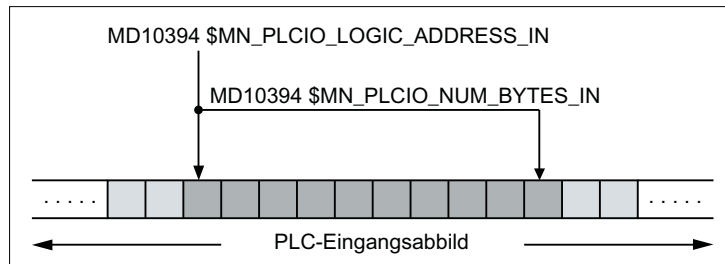
Formatdarstellung

- Formatdarstellung der Systemvariablen \$A_PBx_OUT und \$A_PBx_IN (siehe unten Absatz "Auswahl des Speicherformats (Little/Big-Endian)"):
MD10399 \$MN_PLCIO_TYPE_REPRESENTATION

Hinweis

Die in den Maschinendaten eingetragenen logischen Adressen der PLC-Peripherie und die Anzahl der zu übertragenden Bytes müssen mit der PLC-HW-Konfiguration konsistent sein. Es darf in den konfigurierten Bereichen keine "Adresslücken" im PLC-Peripherieausbau geben.

Prinzip der Parametrierung der NC-Peripherie im Eingangsbereich



Übertragungszeitpunkte

- Übertragung der Ausgangsdaten von der NC zu den Ausgangs-Baugruppen
 - Die Übertragung der Ausgangsdaten zu den Ausgangs-Baugruppen erfolgt am Ende des aktuellen Interpolatortakts.
 - Die Übertragung erfolgt nur, wenn im aktuellen Interpolatortakt mindestens ein Ausgangsdatum geschrieben wurde
- Übertragung der Eingangsdaten von den Eingangs-Baugruppen zur NC
 - Die Zeit, innerhalb der die zyklische Anforderung zur Aktualisierung der Eingangsdaten (Eingangsabbild der Baugruppen → NC-Eingangsdaten) von der NC an die PLC gestellt wird, ist einstellbar über das Maschinendatum:
 - MD10398 \$MN_PLCIO_IN_UPDATE_TIME = <Aktualisierungszeit>
 - Die parametrierte Aktualisierungszeit wird intern auf das nächsthöhere Vielfache des Interpolatortakts aufgerundet. Bei Aktualisierungszeit = 0, wird die Anforderung in jedem Interpolatortakt an die PLC übertragen.
 - Die Anforderung zu Aktualisierung der Eingangsdaten wird am Ende des parametrierten Interpolatortakts an die PLC gestellt.
 - Die aktualisierten Eingangsdaten stehen NC-seitig frühestens im darauf folgenden Interpolatortakt zur Verfügung.

Auswahl des Speicherformats (Little / Big-Endian)

Für den Datentransfer zwischen NC und PLC steht für alle NC-Kanäle insgesamt 16 Byte zur Verfügung. Diese Bereiche sind durch den Anwender selbst zu verwalten (d. h. keine Überschneidung der Variablen, auch nicht über Kanalgrenzen hinweg!).

Die Darstellung der Variablen innerhalb dieser Bereiche erfolgt abhängig von der Einstellung im Maschinendatum:

MD10399 \$MN_PLCIO_TYPE_REPRESENTATION = <Wert>

<Wert >	Bedeutung
0	Little-Endian-Format (Standardeinstellung) Die Abbildung der Systemvariablen erfolgt im Little-Endian-Format ⇒ niederwertigstes Daten-Byte an niederwertigster Adresse
1	Big-Endian-Format (PLC-Standardformat, empfohlene Einstellung) Die Abbildung der Systemvariablen erfolgt im Big-Endian-Format ⇒ höchstwertigstes Daten-Byte an niederwertigster Adresse Hinweis Das Big-Endian-Format ist das in der PLC und PLC-Peripherie gebräuchliche Format. Es wird daher empfohlen diese Einstellung zu verwenden.

13.3.2 Lesen / Schreiben: Systemvariablen

Eingangsdaten

Systemvariable	Typ	Offset <n>
\$A_PBB_IN[<n>]	Byte	0, 1, 2, ... (MD10394 \$MN_PLCIO_NUM_BYTES_IN - 1)
\$A_PBW_IN[<n>]	Word	0, 2, 4, ... (MD10394 \$MN_PLCIO_NUM_BYTES_IN - 2)
\$A_PBD_IN[<n>]	Double	0, 4, 8, ... (MD10394 \$MN_PLCIO_NUM_BYTES_IN - 4)
\$A_PBR_IN[<n>]	Real	0, 4, 8, ... (MD10394 \$MN_PLCIO_NUM_BYTES_IN - 4)

Beim Lesen aus einem Teileprogramm wird im Kanal ein Vorlaufstopp ausgelöst.

Ausgangsdaten

Systemvariable	Typ	Offset <n>
\$A_PBB_OUT[<n>]	Byte	0, 1, 2, ... (MD10396 \$MN_PLCIO_NUM_BYTES_OUT - 1)
\$A_PBW_OUT[<n>]	Word	0, 2, 4, ... (MD10396 \$MN_PLCIO_NUM_BYTES_OUT - 2)
\$A_PBD_OUT[<n>]	Double	0, 4, 8, ... (MD10396 \$MN_PLCIO_NUM_BYTES_OUT - 4)
\$A_PBR_OUT[<n>]	Real	0, 4, 8, ... (MD10396 \$MN_PLCIO_NUM_BYTES_OUT - 4)

Beim Lesen von Ausgangsdaten aus einem Teileprogramm wird im Kanal ein Vorlaufstopp ausgelöst.

Wertebereiche der Ausgangsdaten

Typ	Wertebereich	
Byte	mit Vorzeichen:	-128 ... +127
	ohne Vorzeichen:	0 ... 255
Word	mit Vorzeichen:	-32768 ... +32767
	ohne Vorzeichen:	0 ... 65535
Double	mit Vorzeichen:	-2147483648 ... +2147483647
	ohne Vorzeichen:	0 ... 4294967295
Real	-3.402823466*10 ⁺³⁸ ... +3.402823466*10 ⁺³⁸	

13.3.3 Randbedingungen

Mehrere Slots

Wenn mehrere Slots, einen Ein- oder Ausgangsbereich der direkt von der NC verwendet PLC-Peripherie bilden, müssen sie Adress-mäßig als ein zusammenhängender Bereich ohne Lücken konfiguriert werden.

Paralleles Schreiben von NC und PLC

Ein **paralleles Schreiben von Peripherieausgängen** durch die NC über Direktzugriff über die PLC und aus dem PLC-Anwenderprogramm führt zu einem zufälligen gegenseitigen Überschreiben der Ausgangswerte. Die Anwendung ist daher **unzulässig**, kann Steuerungsseitig aber nicht verhindert werden.

Zeitverhalten

Der Zeitpunkt, zu dem die Daten von der PLC-Peripherie eingelesen werden, und der Zeitpunkt, zu dem die Daten durch die Systemvariablen dem Teileprogramm zur Verfügung gestellt werden, ist nicht synchronisiert!

Datentransfer

- Die Ausgabe der Ausgangsdaten an die PLC-Peripherie erfolgt immer für alle parametrisierten Ausgangsdaten, auch per Systemvariable nur ein Ausgangsdatum geschrieben wurde.
- Werden mehreren Systemvariablen gleichzeitig Werte zugewiesen, ist nicht sichergestellt, dass diese im selben Interpolortakt übertragen werden.

13.3.4 Beispiele

13.3.4.1 Schreiben auf PLC-Peripherie

Vorgaben

- Dezimale logische Adressen der zu schreibenden Ausgangsdaten innerhalb der PLC-Peripherie
 - 521: 1 Byte Integer-Wert
 - 522: 2 Byte Integer-Wert
- Die Ausgangsdaten werden in einem Teileprogramm über die R-Parameter R10 - R11 geschrieben.
- Um die Abarbeitungsgeschwindigkeit des PLC-Anwenderprogramms (OB1) nur gering zu belasten, soll der Aktualisierungstakt für schreibende Zugriffe das 3fache des Interpolortakts von 12 ms betragen.
- Es sollen Daten direkt auf folgende PLC-Peripherie ausgegeben werden:

Parametrierung

Die Maschinendaten sind folgendermaßen zu setzen:

- Länge des Ausgangsbereichs der NC-Peripherie: 2 + 1 = 3 Bytes
 MD10396 \$MN_PLCIO_NUM_BYTES_OUT = 3
- Logische Anfangsadresse des Eingangsbereichs: 521
 MD10397 \$MN_PLCIO_LOGIC_ADRESS_OUT = 521
- Aktualisierungstakt: Peripherie-Eingangsdaten → Systemvariablen = 3 * Interpolortakt = 3 * 12 ms = 36 ms
 MD10398 \$MN_PLCIO_OUT_UPDATE_TIME = 3 * 0.012 = 0.036
- Datenformat: Big-Endian
 MD10399 \$MN_PLCIO_TYPE_REPRESENTATION = 1

Aktualisierung

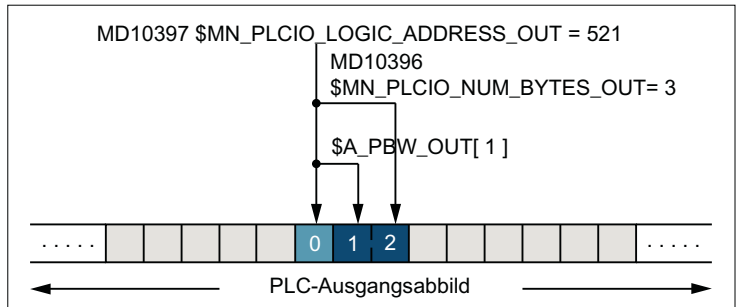
Die Übertragung der Systemdaten zur PLC-Peripherie erfolgt erst ab dem Zeitpunkt, wenn nach dem Hochlauf der Steuerung die PLC die entsprechenden PROFIBUS / PROFINET-Peripherie fehlerfrei erkannt hat. Ab dann erfolgt die Übertragung in jedem dritten Interpolortakt.

Programmierung

Schreiben der NC-Peripherie in Synchronaktionen mit R-Parametern:

Programmcode	Kommentar
R10=123	
R11='Habcd'	
ID=1 WHENEVER TRUE DO \$A_PBB_OUT[0]=R10	; 1 Byte Integer-Wert
ID=2 WHEN \$AA_IW[x] >= 5 DO \$A_PBW_OUT[1]=R11	; 2 Byte Integer-Wert

Adressierungsbeispiel: \$PBW_OUT[1] = R11



13.3.4.2 Lesen von PLC-Peripherie

Vorgaben

- Dezimale logische Adressen der zu lesenden Eingangsdaten innerhalb der PLC-Peripherie
 - 420: 2 Byte Integer-Wert
 - 422: 4 Byte Integer-Wert
 - 426: 4 Byte Real-Wert
 - 430: 1 Byte Integer-Wert
- Die Eingangsdaten sollen in einem Teileprogramm in den R-Parametern R1 - R4 abgelegt werden.
- Um die Abarbeitungsgeschwindigkeit des PLC-Anwenderprogramms (OB1) nur gering zu belasten, soll der Aktualisierungstakt für lesende Zugriffe das 3fache des Interpolatortakts von 12 ms betragen.

Parametrierung

- Länge des Eingangsdatenbereichs der NC-Peripherie: $2 + 4 + 4 + 1 = 11$ Bytes
MD10394 \$MN_PLCIO_NUM_BYTES_IN = 11
- Logische Anfangsadresse des Eingangsdatenbereichs: 420
MD10395 \$MN_PLCIO_LOGIC_ADRESS_IN = 420
- Aktualisierungstakt: Peripherie-Eingangsdaten → Systemvariablen = $3 * \text{Interpolatortakt} = 3 * 12 \text{ ms} = 36 \text{ ms}$
MD10398 \$MN_PLCIO_IN_UPDATE_TIME = $3 * 0.012 = 0.036$
- Datenformat: Big-Endian
MD10399 \$MN_PLCIO_TYPE_REPRESENTATION = 1

Aktualisierung

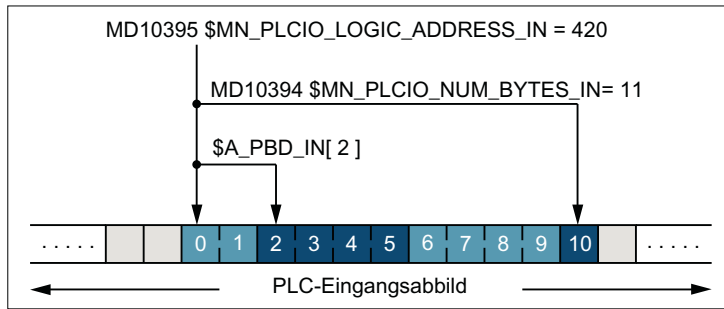
Die Aktualisierung der Systemvariablen erfolgt nach dem Hochlauf von NC und PLC, in jedem dritten Interpolatortakt.

Programmierung

Einlesen der NC-Peripherie in R-Parameter:

Programmcode	Kommentar
R1=\$A_PBW_IN[0]	; 2 Byte Integer-Wert
R2=\$A_PBD_IN[2]	; 4 Byte Integer-Wert
R3=\$A_PBR_IN[6]	; 4 Byte Real-Wert
R4=\$A_PBB_IN[10]	; 1 Byte Integer-Wert

Adressierungsbeispiel: R2 = \$PBD_IN[2]



13.4 Direkte Peripherie-Zugriffe ohne PLC

13.4.1 Kurzbeschreibung

Isochrone und nicht isochrome PROFIBUS / PROFINET

Das Lesen / Schreiben der PROFIBUS / PROFINET-Peripherie ist sowohl bei isochroner und nicht isochroner PROFIBUS / PROFINET-Projektierung möglich.

E/A-Bereich

Werden Slots eines für die NC-Peripherie verwendeten PROFIBUS / PROFINET-Slaves so projektiert, dass sie in aufsteigender Reihenfolge **lückenlos** hintereinander liegen, werden sie nachfolgend als E/A-Bereich bezeichnet.

Ein E/A-Bereich ist somit gekennzeichnet durch:

- Logische Startadresse: Logische Startadresse des ersten Slots des PROFIBUS / PROFINET-Slaves
- Länge: Gesamtlänge aller verwendeten Slots des PROFIBUS / PROFINET-Slaves

Die Parametrierung der logischen Startadresse und der Länge des E/A-Bereichs erfolgt in der NC über Maschinendaten (siehe "Parametrierung (Seite 753)").

Lesen / Schreiben

Teileprogramme / Synchronaktionen: Systemvariablen

Das Lesen / Schreiben der NC-Peripherie erfolgt in der NC über Systemvariablen im Interpolatortakt. Das Schreiben auf die NC-Peripherie erfolgt nach dem Interpolatortakt.

Datenkonsistenz: Interpolatortakt

Siehe "Systemvariablen (Seite 755)".

Paralleles Lesen / Schreiben

Lesen

Paralleles Lesen durch Compile-Zyklen und Teileprogrammen / Synchronaktionen auf Eingangsdaten des gleichen E/A-Bereichs ist **möglich**. Die Lesezugriffe erfolgen auf verschiedenen Wegen auf die NC-Peripherie.

- Die **Datenkonsistenz** ist dabei **sichergestellt**.
- Die **Datengleichheit** während eines Interpolatortaktes ist **nicht sichergestellt**.

Schreiben

Paralleles Schreiben durch Compile-Zyklen und Teileprogramme / Synchronaktionen auf Ausgangsdaten des gleichen E/A-Bereichs ist **nicht möglich**.

13.4.2 Parametrierung

Maschinendaten

Logische Startadressen der E/A-Bereiche

Die logischen Startadressen der verwendeten E/A-Bereiche werden über folgende Maschinendaten eingestellt:

- Logische Startadresse des Eingangsbereichs 1, 2, ... m:
MD10500 \$MN_DPIO_LOGIC_ADDRESS_IN[<n>] = <logische Startadresse> ; mit <n> = 0, 1, 2, ... (m - 1)
 - Logische Startadresse des Ausgangsbereichs 1, 2, ... m:
MD10510 \$MN_DPIO_LOGIC_ADDRESS_OUT[<n>] = <logische Startadresse> ; mit <n> = 0, 1, 2, ... (m - 1)
- Bei logischen Startadressen innerhalb des PLC-Prozessabbildes ist Folgendes zu beachten (siehe Literatur):

- Eingangsslots: Lesen von der NC aus möglich
- Ausgangsslots: Schreiben von der NC aus verboten ⇒ Alarm nach Hochlauf der NC

Weitere Informationen

Gerätehandbuch NCU 7x0.3 PN, NCU 7x0.3B PN; Technische Daten > PLC > Prozessabbildgröße

Ausführliche Informationen finden sich unter der Adresse (<http://support.automation.siemens.com/WW/view/de/54058408>)

Hinweis

E/A-Bereiche für den **schreibenden** Zugriff (MD10510) auf die PROFIBUS-Peripherie dürfen nicht im Bereich des Prozessabbilds, z. B. PLC 317-3, Ausgangsadressen 0 - 1023, liegen.

Bei PROFINET-Peripherie müssen diese E/A-Bereiche im Prozessabbild dem "TPA 2" zugeordnet werden, die Projektierung erfolgt über den Taktsynchronalarm "NC".

Längen der E/A-Bereiche

Die Längen der verwendeten E/A-Bereiche werden über folgende Maschinendaten eingestellt:

- Länge des Eingangsbereichs 1, 2, ... m:
MD10501 \$MN_DPIO_RANGE_LENGTH_IN[<n>] = <Länge>; mit <n> = 0, 1, 2, ... (m - 1)
- Länge des Ausgangsbereichs 1, 2, ... m:
MD10511 \$MN_DPIO_RANGE_LENGTH_OUT[<n>] = <Länge> ; mit <n> = 0, 1, 2, ... (m - 1)

Wird als Länge der Wert 0 eingetragen, wird intern als Länge des E/A-Bereichs die Länge des ersten, unter der angegebenen logischen Startadresse (MD10500 / MD10510) gefundenen Nutzdaten-Slots gesetzt.

E/A-Bereichs-Attribute

- Attribute des Eingangsbereichs 1, 2, ... m:
MD10502 \$MN_DPIO_RANGE_ATTRIBUTE_IN[<n>] ; mit <n> = 0, 1, 2, ... (m - 1)

Bit	Wert	Bedeutung
0	Formatdarstellung der Systemvariablen \$A_DPx_IN[<n>,<m>]	
	0	Little-Endian Format
	1	Big-Endian Format
2	Lesen von Eingangsdaten	
	0	Lesen über Systemvariable und CC-Binding möglich
	1	Lesen nur für CC-Binding möglich
3	Ausgabe von Slot-Lebenszeichen-Alarmen	
	0	Slot-Lebenszeichen-Alarme werden ausgegeben
	1	Slot-Lebenszeichen-Alarme werden unterdrückt

- Attribute des Ausgangsbereichs 1, 2, ... m:
MD10512 \$MN_DPIO_RANGE_ATTRIBUTE_OUT[<n>] ; mit <n> = 0, 1, 2, ... (m - 1)

Bit	Wert	Bedeutung
0	Formatdarstellung der Systemvariablen \$A_DPx_OUT[<n>,<m>]	
	0	Little-Endian Format
	1	Big-Endian Format
1	Schreiben von Ausgangsdaten	
	0	Schreiben nur über Systemvariable
	1	Schreiben nur über CC-Binding
3	Ausgabe von Slot-Lebenszeichen-Alarmen	
	0	Slot-Lebenszeichen-Alarme werden ausgegeben
	1	Slot-Lebenszeichen-Alarme werden unterdrückt

13.4.3 Lesen / Schreiben

13.4.3.1 Systemvariablen

Eingangsdaten

Systemvariable	Typ	Bedeutung
\$A_DPB_IN[<n>,]	8 Bit unsigned	Lesen eines Datenbytes (8 Bit)
\$A_DPW_IN[<n>,]	16 Bit unsigned	Lesen eines Datenwortes (16 Bit)
\$A_DPSB_IN[<n>,]	8 Bit signed	Lesen eines Datenbytes (8 Bit)
\$A_DPSW_IN[<n>,]	16 Bit signed	Lesen eines Datenwortes (16 Bit)
\$A_DPSD_IN[<n>,]	32 Bit signed	Lesen eines Datendoppelwortes (32 Bit)
\$A_DPR_IN[<n>,]	32 Bit REAL	Lesen von Eingangs-Daten (32 Bit REAL)

<n> = Eingangsbereich 1, 2, ... m; = Byte-Index innerhalb des Eingangsbereichs: 0, 1, ... (Länge - 1)

Ausgangsdaten

Systemvariable	Typ	Bedeutung
\$A_DPB_OUT[<n>,]	8 Bit unsigned	Schreiben eines Datenbytes (8 Bit)
\$A_DPW_OUT[<n>,]	16 Bit unsigned	Schreiben eines Datenwortes (16 Bit)
\$A_DPSB_OUT[<n>,]	8 Bit signed	Schreiben eines Datenbytes (8 Bit)
\$A_DPSW_OUT[<n>,]	16 Bit signed	Schreiben eines Datenwortes (16 Bit)
\$A_DPSD_OUT[<n>,]	32 Bit signed	Schreiben eines Datendoppelwortes (32 Bit)
\$A_DPR_OUT[<n>,]	32 Bit REAL	Schreiben von Ausgangs-Daten (32 Bit REAL)

<n> = Ausgangsbereichs 1, 2, ... m; = Byte-Index innerhalb des Ausgangsbereichs: 0, 1, ... (Länge - 1)

Konfigurierte und parametrisierte E/A-Bereiche für Teileprogramme / Synchronaktionen

Jede Systemvariable ist ein 32-Bit Bitfeld. Jedes Bit ist einem E/A-Bereich zugeordnet.

Bit <n> \triangleq Maschinendaten-Index <n> \triangleq E/A-Bereich <n+1>

Bit <n> == 1 \Rightarrow Der E/A-Bereich <n+1> ist konfiguriert / parametrisiert.

Systemvariable	Typ	Bedeutung
\$A_DP_IN_CONF	32 Bit Bitfeld	Lesen aller konfigurierten Eingangsbereiche
\$A_DP_OUT_CONF	32 Bit Bitfeld	Lesen aller konfigurierten Ausgangsbereiche

Gültige E/A-Bereiche für Teileprogramme / Synchronaktionen

Jede Systemvariable ist ein 32-Bit Bitfeld. Jedes Bit ist einem E/A-Bereich zugeordnet.

Bit <n> \triangleq Maschinendaten-Index <n> \triangleq E/A-Bereich <n+1>

Bit <n> == 1 ⇒ Der E/A-Bereich <n+1> ist gültig. Lesen / Schreiben über Teileprogramme / Synchronaktionen ist möglich.

Systemvariable	Typ	Bedeutung
\$A_DP_IN_VALID	32 Bit Bitfeld	Lesen aller gültigen Eingangsbereiche
\$A_DP_OUT_VALID	32 Bit Bitfeld	Lesen aller gültigen Ausgangsbereiche

Zustand eines E/A-Bereichs

Über die folgenden Systemvariablen kann der Zustand eines E/A-Bereichs gelesen werden.

Systemvariable	Typ	Bedeutung
\$A_DP_IN_STATE[<n>]	INT	Lesen des Zustandes des Eingangsbereichs
\$A_DP_OUT_STATE[<n>]		Lesen des Zustandes des Ausgangsbereichs
<n> = Index des E/A-Bereichs		
Zustand		
0: E/A-Bereich wurde nicht konfiguriert		
1: E/A-Bereich konnte noch nicht aktiviert werden		
2: E/A-Bereich ist verfügbar		
3: E/A-Bereich ist momentan nicht verfügbar		

Länge eines E/A-Bereichs

Über die folgenden Systemvariablen kann die konfigurierte Länge eines E/A-Bereichs gelesen werden.

Systemvariable	Bedeutung
\$A_DP_IN_LENGTH[<n>]	Lesen der Länge des Eingangs-Datenbereichs
\$A_DP_OUT_LENGTH[<n>]	Lesen der Länge des Ausgangs-Datenbereichs
<n> = Index des Ein-/Ausgangsbereichs	

Randbedingungen

- Beim Lesen / Schreiben aus einem Teileprogramm wird im Kanal ein Vorlaufstopp ausgelöst.
- Um Datenkonsistenz bei der Programmierung aus dem Teileprogramm und den Synchronaktionen zu gewährleisten, erfolgt der Zugriff auf PROFIBUS-Peripherie-Daten, die für den jeweiligen IPO-Takt konsistent gehalten werden.
- Sollte innerhalb eines Interpolatorstaktes mehrfach auf dieselben PROFIBUS-Peripherie-Daten schreibend zugegriffen werden (z. B. Synchronaktionen, Zugriff aus verschiedenen Kanälen etc.), so sind die Daten des jeweils letzten schreibenden Zugriffs gültig.
- Zu schreibende PROFIBUS-Peripherie-Daten werden erst nach dem entsprechenden IPO-Takt an die PROFIBUS-Peripherie ausgegeben.

- Mit `` (RangeOffset) wird auf die Stelle (Byte-Offset) innerhalb des E/A-Bereichs verwiesen, ab dem der Datenzugriff geschehen soll. Datentypen können an beliebigen Byte-Offsets innerhalb des E/A-Bereichs gelesen/geschrieben werden. Lese-/Schreibzugriffe, die die konfigurierten Grenzen des jeweiligen E/A-Bereichs überschreiten, werden mit der Ausgabe eines Alarms (17030) abgelehnt.
- Über die Maschinendaten MD10502 `$MN_DPIO_RANGE_ATTRIBUTE_IN` und MD10512 `$MN_DPIO_RANGE_ATTRIBUTE_OUT` (siehe "Parametrierung (Seite 753)") kann sowohl für die Lese-/Schreibrichtung als auch für jeden einzelnen E/A-Bereich das Darstellungsformat (Little-/Big-Endian) für `$A_DPx_IN[<n>,]`- bzw. `$A_DPx_OUT[<n>,]`-Systemvariablen festgelegt werden.

13.4.4 Randbedingungen

Paralleles Schreiben von NC und PLC

Ein **paralleles Schreiben von Peripherieausgängen** durch die NC über Direktzugriff und aus dem PLC-Anwenderprogramm führt zu einem zufälligen gegenseitigen Überschreiben der Ausgangswerte. Die Anwendung ist daher **unzulässig**, kann Steuerungs-seitig aber nicht verhindert werden.

Lebenszeichenüberwachung

Am Anfang eines IPO-Takts wird für jeden E/A-Bereich überprüft, ob das Lebenszeichen des zugehörigen Slots bzw. E-A-Bereichs ausgefallen ist. Ist das der Fall, wird Alarm 9050 bzw. 9052 angezeigt.

Auswirkungen:

- Die Teileprogrammbearbeitung wird **nicht** angehalten
- Mit Rückkehr des Lebenszeichens wird der Alarm automatisch wieder gelöscht.

13.4.5 Beispiele

13.4.5.1 Schreiben auf NC-Peripherie

Voraussetzung

Eine gültige Konfiguration muss in der PLC bereits geladen sein.

Parametrierung für Teileprogramme / Synchronaktionen

Vorgaben

- Parametrierung 6. Datensatz: Maschinendaten / Systemvariablen-Index = 5
- Konfigurationsdaten:
 - logische Startadresse = 334
 - Slotlänge = 8 Byte
- Darstellung: Little-Endian-Format

Parametrierung in Maschinendaten

- MD10510 \$MN_DPIO_LOGIC_ADDRESS_OUT[5] = 334 (logische Startadresse)
- MD10511 \$MN_DPIO_LENGTH_OUT[5] = 8 (Länge des E/A-Bereichs in Byte)
- MD10512 \$MN_DPIO_ATTRIBUTE_OUT[5]
 - Bit0 = 0 (Little-Endian-Format)
 - Bit1 = 0 (Schreiben nur über Systemvariable)
 - Bit3 = 0 (Slot-Lebenszeichen-Alarme ausgeben)

Beispiele

Programmcode	Kommentar
\$A_DPB_OUT[5,6]=128	; Byte $\hat{=}$ 8 Bit, Index=5, Offset=6
\$A_DPW_OUT[5,5]='B0110'	; Word $\hat{=}$ 16 Bit, Index=5, Offset=5
\$A_DPSD_OUT[5,3]='H8F'	; Double $\hat{=}$ 32 Bit, Index=5, Offset=3
	; Achtung: Daten auf Offset 4,5,6 werden überschrieben
\$AC_MARKER[0]=5	; Index=5
\$AC_MARKER[1]=3	; Offset=3
\$A_DPSD_OUT[\$AC_MARKER[0],\$AC_MARKER[1]]='H8F'	; indirekte Adressierung
	; Double $\hat{=}$ 32 Bit, auf Index=5, Offset=3
R1 = \$A_DPB_OUT[5,6]	; Zuweisung an R-Parameter, Byte $\hat{=}$ 8 Bit, Index=5, Offset=6
	; Ergebnis: R1 == 'HFF'
ID=1 WHENEVER TRUE DO \$A_DPB_OUT[5,0]=123	; zyklisches Schreiben pro IPO-Takt
	; Byte $\hat{=}$ 8 Bit, Index=5, Offset=0
; Fehlerhafte Programmierungen	
\$A_DPB_OUT[5,255]=128	; \Rightarrow Alarm 17030: Offset 255 > E/A-Bereich
\$A_DPB_OUT[6,10]=128	; \Rightarrow Alarm 17020: Index 6 reserviert für Compilezyklus s.u.
\$A_DPB_OUT[7,10]=128	; \Rightarrow Alarm 17020: Index 7 in Maschinendaten nicht definiert
\$A_DPB_OUT[16,6]=128	; \Rightarrow Alarm 17020: Index 16 ausserhalb des Wertebereichs

Konfiguration für Programmierung über CompileZyklen

Vorgaben

- Parametrierung 7. Datensatz: Maschinendaten / Systemvariablen-Index = 6
- Konfigurationsdaten:
 - logische Startadresse = 444
 - Slotlänge = 10 Byte
- Darstellung: Little-Endian-Format

Parametrierung in Maschinendaten

- MD10510 \$MN_DPIO_LOGIC_ADDRESS_OUT[6] = 444 (logische Startadresse)
- MD10511 \$MN_DPIO_LENGTH_OUT[6] = 0 (es wird nur der erste Nutzdaten-Slot verwendet)
- MD10512 \$MN_DPIO_ATTRIBUTE_OUT[6]
 - Bit0 = 0 (Little-Endian-Format)
 - Bit1 = 1 (Schreiben nur über CC-Binding)
 - Bit3 = 1 (Slot-Lebenszeichen-Alarme werden unterdrückt)

13.4.5.2 Lesen von NC-Peripherie

Voraussetzung

Eine gültige Konfiguration muss in der PLC bereits geladen sein.

Parametrierung für Teileprogramme / Synchronaktionen

Vorgaben

- Parametrierung 1. Datensatz: Maschinendaten / Systemvariablen-Index = 0
- Konfigurationsdaten:
 - logische Startadresse = 456
 - Slotlänge = 32 Byte
- Darstellung: Big-Endian-Format

Parametrierung in Maschinendaten

- MD10500 \$MN_DPIO_LOGIC_ADDRESS_IN[0] = 456 (logische Startadresse)
- MD10501 \$MN_DPIO_LENGTH_IN[0] = 32 (Länge des E/A-Bereichs in Byte)
- MD10502 \$MN_DPIO_ATTRIBUTE_IN[0]
 - Bit0 = 1 (Big-Endian-Format)
 - Bit2 = 0 (Lesen über Systemvariable und CC-Binding möglich)
 - Bit3 = 0 (Slot-Lebenszeichen-Alarme werden ausgegeben)

Beispiele

Programmcode	Kommentar
\$AC_MARKER[0]= \$A_DPW_IN[0,0]	; Byte ≙ 8 Bit, Index=0, Offset=0
\$AC_MARKER[1]= \$A_DPSD_IN[0,1]	; signed Double ≙ 32 Bit, Index=0, Offset=1
\$AC_MARKER[1]= \$A_DPSD_IN[0,8]	; signed Double ≙ 32 Bit, Index=0, Offset=8
\$AC_MARKER[2]=0	; Index=0
\$AC_MARKER[3]=8	; Offset=8
\$AC_MARKER[1]=\$A_DPSD_IN[\$AC_MARKER[2],\$AC_MARKER[3]]	; indirekte Adressierung
	; signed Double ≙ 32 Bit, Index=0, Offset=8
ID=2 WHEN \$A_DPB_IN[0,11]>=5 DO \$AC_MARKER[2]=\$A_DPSD_IN[0,8]	; IF Index 0, Offset 11 >= 5
	; THEN signed Double ≙ 32 Bit, Index=0, Offset=8
; Fehlerhafte Programmierungen	
R1=\$A_DPB_IN[0,255]	; ⇒ Alarm 17030: Offset 255 > E/A-Bereich
R1=\$A_DPB_IN[2,6]	; ⇒ Alarm 17020: Index 2 in Maschinendaten nicht definiert
R1=\$A_DPB_IN[1,10]	; ⇒ Alarm 17020: Index 1 reserviert für Compilezyklus s.u.
R1=\$A_DPB_IN[16,6]	; ⇒ Alarm 17020: Index 16 ausserhalb des Wertebereichs

Parametrierung für Programmierung über CompileZyklen

Vorgaben

- Parametrierung 2. Datensatz: Maschinendaten / Systemvariablen-Index = 1
- Konfigurationsdaten:
 - logische Startadresse = 312
 - Slotlänge = 32 Byte
- Darstellung: Big-Endian-Format

Parametrierung in Maschinendaten

- MD10500 \$MN_DPIO_LOGIC_ADDRESS_IN[01] = 312 (logische Startadresse)
- MD10501 \$MN_DPIO_LENGTH_IN[1] = 32 (Länge des E/A-Bereichs in Byte)
- MD10502 \$MN_DPIO_ATTRIBUTE_IN[1]
 - Bit0 = 1 (Big-Endian-Format)
 - Bit2 = 1 (Lesen nur für CC-Binding möglich)
 - Bit3 = 1 (Slot-Lebenszeichen-Alarme unterdrücken)

13.4.5.3 Schreiben von NC-Peripherie mit Statusabfrage

Voraussetzung

Eine gültige Konfiguration muss in der PLC bereits geladen sein.

Parametrierung für Teileprogramme / Synchronaktionen

Vorgaben

- Parametrierung 6. Datensatz: Maschinendaten / Systemvariablen-Index = 5
- Konfigurationsdaten:
 - logische Startadresse = 1200
 - Slotlänge = 32 Byte
- Darstellung: Big-Endian-Format

Parametrierung in Maschinendaten

- MD10510 \$MN_DPIO_LOGIC_ADDRESS_OUT[5] = 1200 (logische Startadresse)
- MD10511 \$MN_DPIO_LENGTH_OUT[5] = 0 (Länge des E/A-Bereichs in Byte)
- MD10512 \$MN_DPIO_ATTRIBUTE_OUT[5]
 - Bit0 = 1 (Little-Endian-Format)
 - Bit1 = 0 (Schreiben nur über Systemvariable)
 - Bit3 = 0 (Slot-Lebenszeichen-Alarme ausgeben)

Beispiele

```
check:                                     ; Sprungmarke
IF $A_DP_OUT_STATE[5]==2 GOTOF write      ; wenn Datenbereich gültig; =>
                                           Sprung zu N15
GOTOB check                               ; Rücksprung zu check
write:                                     ; Sprungmarke
$A_DPB_OUT[5,6]=128                       ; Schreiben des Datenbytes
```

13.5 Datenlisten

```

check:                                     ; Sprungmarke
IF $A_DP_OUT_CONF==$A_DP_OUT_VALID GOTOF write ; wenn Datenbereich gültig
                                                ; => Sprung zu N15

SETAL(61000)                               ; Alarm Nr. 61000 setzen

write:                                     ; Sprungmarke
$A_DPB_OUT[5,6]=128                        ; Schreiben des Datenbytes

check:                                     ; Sprungmarke
IF $A_DP_OUT_VALID B_AND 'B100000' GOTOF write ; wenn Datenbereich gültig
                                                ; => Sprung zu N15

SETAL(61000)                               ; Alarm Nr. 61000 setzen

write:                                     ; Sprungmarke
$A_DPB_OUT[5,6]=128                        ; Schreiben des Datenbytes

R1=$A_DP_OUT_LENGTH[5]                    ; Länge des E/A-Bereichs (Slots) in Byte
                                                ; Ergebnis: R1 = 32
    
```

13.5 Datenlisten

13.5.1 Maschinendaten

13.5.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10300	FASTIO_ANA_NUM_INPUTS	Anzahl der aktiven analogen NC-Eingänge
10310	FASTIO_ANA_NUM_OUTPUTS	Anzahl der aktiven analogen NC-Ausgänge
10320	FASTIO_ANA_INPUT_WEIGHT	Bewertungsfaktor für analoge NC-Eingänge
10330	FASTIO_ANA_OUTPUT_WEIGHT	Bewertungsfaktor für analoge NC-Ausgänge
10350	FASTIO_DIG_NUM_INPUTS	Anzahl der aktiven digitalen NC-Eingangsbytes
10360	FASTIO_DIG_NUM_OUTPUTS	Anzahl der aktiven digitalen NC-Ausgangsbytes
10362	HW_ASSIGN_ANA_FASTIN	Hardware-Zuordnung der externen analogen NC-Eingänge
10364	HW_ASSIGN_ANA_FASTOUT	Hardware-Zuordnung der externen analogen NC-Ausgänge
10366	HW_ASSIGN_DIG_FASTIN	Hardware-Zuordnung der externen digitalen NC-Eingänge
10368	HW_ASSIGN_DIG_FASTOUT	Hardware-Zuordnung der externen digitalen NC-Ausgänge
10394	PLCIO_NUM_BYTES_IN	Anzahl der direkt lesbaren Eingangsbytes der PLC-Peripherie
10395	PLCIO_LOGIC_ADDRESS_IN	Startadresse der direkt lesbaren Eingangsbytes der PLC-Peripherie
10396	PLCIO_NUM_BYTES_OUT	Anzahl der direkt schreibbaren Ausgangsbytes der PLC-Peripherie

Nummer	Bezeichner: \$MN_	Beschreibung
10397	PLCIO_LOGIC_ADDRESS_OUT	Startadresse der direkt schreibbaren Ausgangsbytes der PLC-Peripherie
10398	PLCIO_IN_UPDATE_TIME	Updatetime für PLCIO-Input-Zyklus
10399	PLCIO_TYPE_REPRESENTATION	Little-/Big-Endian Datendarstellung für PLCIO
10500	DPIO_LOGIC_ADDRESS_IN	logische Slotadresse der PROFIBUS-Peripherie
10501	DPIO_RANGE_LENGTH_IN	Länge des PROFIBUS-Peripherie Bereiches
10502	DPIO_RANGE_ATTRIBUTE_IN	Attribute der PROFIBUS-Peripherie
10510	DPIO_LOGIC_ADDRESS_OUT	logische Slotadresse der PROFIBUS-Peripherie
10511	DPIO_RANGE_LENGTH_OUT	Länge des PROFIBUS-Peripherie-Bereiches
10512	DPIO_RANGE_ATTRIBUTE_OUT	Attribute der PROFIBUS-Peripherie
10530	COMPAR_ASSIGN_ANA_INPUT_1	Hardware-Zuordnung der NC-Analogueingänge für Komparatorbyte 1
10531	COMPAR_ASSIGN_ANA_INPUT_2	Hardware-Zuordnung der NC-Analogueingänge für Komparatorbyte 2
10540	COMPAR_TYPE_1	Parametrierung für Komparatorbyte 1
10541	COMPAR_TYPE_2	Parametrierung für Komparatorbyte 2

13.5.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
21220	MULTFEED_ASSIGN_FASTIN	Zuordnung der Eingangsbytes der NC-Peripherie für "Mehrere Vorschübe in einem Satz"

13.5.2 Settingdaten

13.5.2.1 Allgemeine Settingdaten

Nummer	Bezeichner: \$SN_	Beschreibung
41600	COMPAR_THRESHOLD_1	Schwellwerte für Komparatorbyte 1
41601	COMPAR_THRESHOLD_2	Schwellwerte für Komparatorbyte 2

13.5.3 Systemvariable

Bezeichner	Beschreibung
\$A_IN	digitaler NC-Eingang
\$A_OUT	digitaler NC-Ausgang
\$A_INA	analoger NC-Eingang
\$A_OUTA	analoger NC-Ausgang

13.5 Datenlisten

Bezeichner	Beschreibung
\$A_PBB_IN	digitale NC-Eingangsdaten, Byte
\$A_PBW_IN	digitale NC-Eingangsdaten, Word
\$A_PBD_IN	digitale NC-Eingangsdaten, Double
\$A_PBR_IN	digitale NC-Eingangsdaten, Real
\$A_PBB_OUT	digitale NC-Ausgangsdaten, Byte
\$A_PBW_OUT	digitale NC-Ausgangsdaten, Word
\$A_PBD_OUT	digitale NC-Ausgangsdaten, Double
\$A_PBR_OUT	digitale NC-Ausgangsdaten, Real

13.5.4 Signale

13.5.4.1 Signale an NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Sperre der digitalen NC-Eingänge	DB10.DBB0/122/124/126/128	DB2800.DBB0/1000
Setzen von PLC der digitalen NC-Eingänge	DB10.DBB1/123/125/127/129	DB2800.DBB1/1001
Sperre der digitalen NC-Ausgänge	DB10.DBB4/130/134/138/142	DB2800.DBB4/1008
Überschreibmaske der digitalen NC-Ausgänge	DB10.DBB5/131/135/139/143	DB2800.DBB5/1009
Setzwert von PLC der digitalen NC-Ausgänge	DB10.DBB6/132/136/140/144	DB2800.DBB6/1010
Vorgabemaske der digitalen NC-Ausgänge	DB10.DBB7/133/137/141/145	DB2800.DBB7/1011
Sperre der analogen NC-Eingänge	DB10.DBB146	-
Vorgabemaske der analogen NC-Eingänge	DB10.DBB147	-
Setzwert von PLC für analoge NC-Eingänge	DB10.DBB148-163	-
Überschreibmaske der analogen NC-Ausgänge	DB10.DBB166	-
Vorgabemaske der analogen NC-Ausgänge	DB10.DBB167	-
Sperre der analogen NC-Ausgänge	DB10.DBB168	-
Setzwert von PLC für analoge NC-Ausgänge	DB10.DBB170-185	-

13.5.4.2 Signale von NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Istwert der digitalen NCK-Eingänge	DB10.DBB60/186-189	DB2900.DBB0/1000
Sollwert der digitalen NCK-Ausgänge	DB10.DBB64/190-193	DB2900.DBB4/1004
Istwert der analogen NCK-Eingänge	DB10.DBB194-209	-
Sollwert der analogen NCK-Ausgänge	DB10.DBB210-225	-

B3: Dezentrale Systeme

14.1 Kurzbeschreibung

14.1.1 Mehrere Bedientafeln an mehreren NCU (T:M:N)

Zum Bedienen und Beobachten komplexer Anlagen und Maschinen ist eine einzelne Bedien- und Beobachtungsstation unter Umständen nicht ausreichend. Daher können in einem SINUMERIK Anlagennetz (Ethernet) mehrere Bedien- und Beobachtungsstation über eine PCU mit mehreren numerischen Steuerungen (NCU) so verschaltet werden, dass sie zusammen ein flexibles und verteiltes Bedienen und Beobachten der Gesamtanlage ermöglichen.

Das folgende Bild gibt einen Überblick über die Komponenten, die aktuell in einem Anlagennetz zu einem Bedien- und Beobachtungssystem T:1:N zusammenschaltet werden können. Dabei sind:

- **T:** Thin Client Unit (TCU) bzw. Bedienhandgerät HT8/HT10 (mit der PCU verbunden)
- **M:** Man Machine Control (MMC), z.B. PCU 50.x mit SINUMERIK Operate
- **N:** Numeric Control Unit (NCU), z.B. NCU 7x0.3 PN

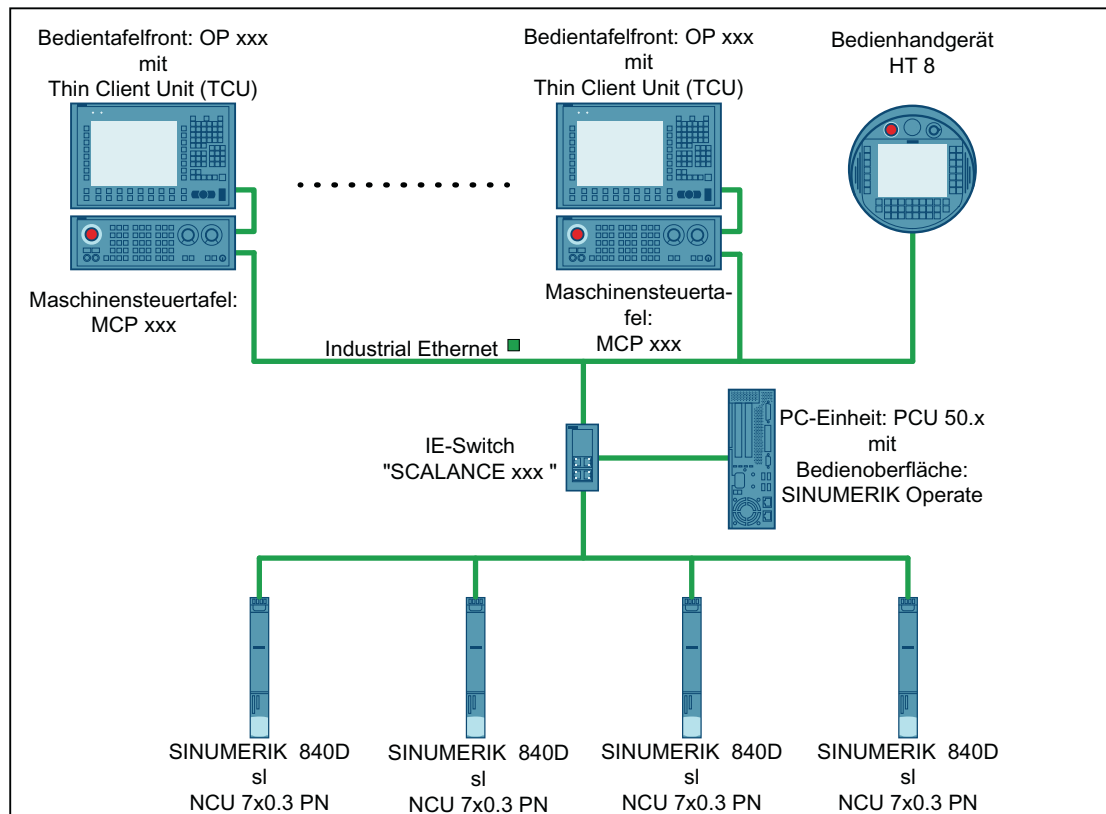


Bild 14-1 Beispiel für einen T:1:N Verbund

T: Thin Client Unit (TCU) bzw. Bedienhandgerät HT8

Über die TCU werden die Graphikinformationen der Bedienoberfläche von der PCU auf die Bedientafelfront (OP) übertragen und dort visualisiert. TCU und Bedientafelfront ergeben zusammen eine Bedienstation.

- Maximale Anzahl
Im Anlagennetz kann eine prinzipiell beliebige, nur durch den zur Verfügung stehenden Adressraum begrenzte Anzahl von Bedienstationen betrieben werden.
- Online
Pro PCU können gleichzeitig maximal vier Bedienstationen online sein.
- Bedienfokus
Von den Bedienstationen die an der PCU online sind, kann zu einem Zeitpunkt nur eine Bedienstation den Bedienfokus haben. Nur über diese Bedienstation können Eingaben bezüglich der Bedienoberfläche gemacht werden. An den anderen Bedienstationen die online sind, wird die Bedienoberfläche, als Kennzeichen, dass diese nicht den Bedienfokus haben, abgedunkelt angezeigt. An Bedienstationen die nicht online sind, wird nur ein Hinweisbild angezeigt.
Der Bedienfokus kann zwischen den Bedienstationen die online sind umgeschaltet werden.
- Bedienhandgerät HT8
Ein Bedienhandgerät HT8 vereint eine TCU, eine Bedientafelfront und ein Maschinensteuertafel in einem Gerät.

M: PCU/IPC mit SINUMERIK Operate

An eine PCU mit Bedienoberfläche SINUMERIK Operate können mehrere Bedienstationen und mehrere NCUs angeschlossen werden. Durch entsprechende Projektierung des Kanalmenüs, können auf den Bedienstationen die Daten der Kanäle von unterschiedlichen NCU angezeigt werden.

N: SINUMERIK 840D sl

- Interne Bedienoberfläche SINUMERIK Operate
Aufgrund der im Anlagennetz vorhandenen externen Bedienoberfläche SINUMERIK Operate auf der PCU, müssen die NCU-integrierten internen Bedienoberfläche SINUMERIK Operate abgeschaltet werden. Die interne Bedienoberfläche SINUMERIK Operate muss nur dann nicht abgeschaltet werden, wenn sie mit eingeschränktem Funktionsumfang, ausschließlich im Rahmen der Werkzeugverwaltung verwendet wird. Ausführliche Informationen dazu finden sich im Inbetriebnahmehandbuch "Bedienkomponenten und Vernetzung (IM5)", Kapitel "Netzkonfiguration" > "Anwendungsbeispiel"
- Maschinensteuertafel (MCP)
Für eine flexible Umschaltung der Maschinensteuertafeln, kann nur eine MCP mit einer NCU verbunden sein. Die Umschaltung der MCP erfolgt anwenderspezifisch über das PLC-Anwenderprogramm (Funktionsbaustein FB 9).

Aufbauen und Anschließen**Weitere Informationen**

- TCU, MCP, PCU:
Gerätehandbuch SINUMERIK 840D sl Bedienkomponenten und Vernetzung
- NCU:
Gerätehandbuch SINUMERIK Steuerung
- Maschinensteuertafel (MCP)
FB 9: MzuN Bedieneinheitenumschaltung
- Switch SCALANCE:
Systemhandbuch Kommunikation mit SIMATIC
- PCU-Basesoftware kompatible Industrie-PCs
Inbetriebnahmehandbuch SINUMERIK 840D sl PCU-Basesoftware

Projektierung, Inbetriebnahme und Parametrierung**Weitere Informationen**

- Aufbau des Anlagennetzes und Inbetriebnahme einer TCU:
Inbetriebnahmehandbuch Bedienkomponenten und Vernetzung (IM5)
- Projektierung des Kanalmenüs
Inbetriebnahmehandbuch SINUMERIK Operate (IM9), Kapitel "Kanalmenü"

Programmierung

Weitere Informationen

- MCP:
Funktionshandbuch Grundfunktionen, PLC-Grundprogramm

Randbedingungen

Aktuell kann nur **eine** PCU (M) pro T:M:N Verbund projektiert werden ⇒ T:1:N

14.1.2 NCU-Link

14.1.2.1 Link-Kommunikation

Die NCU-Link-Kommunikation ermöglicht einen Interpolatortakt-synchronen NCU-übergreifenden Datenaustausch für folgende Applikationsaufgaben:

- NCU-übergreifender Datenaustausch über Link-Variablen \$A_DLx
- NCU-übergreifendes Verfahren von Achsen und Spindeln mittels Link-Achsen und Container-Link-Achsen
- NCU-übergreifende Leitwertkopplung von Achsen und Spindeln mittels Lead-Link-Achsen

Safety Integrated

Sind achsspezifische Safety Integrated Funktionen für eine Linkachse aktiv, werden über NCU-Link nicht sicherheitsrelevante Daten zwischen der lokalen Safety-Achse (Linkachse auf NCUx) und dem sie verfahrenen Kanal (NCUy) ausgetauscht. Dies ermöglicht auch für Linkachsen Safety Integrated-unterstützende Funktionen, wie z. B. Geschwindigkeits-/Drehzahlüberwachungsabhängige Sollwertbegrenzung und höherwertige Stopp-Funktionen (STOP D/E).

Link-Modul

Zur Link-Kommunikation werden optionale Link-Module benötigt. Ein Link-Modul ist eine optionale PROFINET-Baugruppe für isochrone Echtzeit-Kommunikation (IRT) über Ethernet. Das Link-Modul belegt den Option Slot der NCU.

Mengengerüst

Standardmäßig können maximal 3 NCU zu einem Link-Verbund zusammengeschaltet werden.

Hinweis

Projektspezifisch können auf Anfrage bei ihrem regionalen Siemens-Ansprechpartner weitere NCU in einen Link-Verbund eingebunden werden.

14.1.2.2 Link-Variablen

Link-Variablen sind systemglobale Anwendervariablen, die bei projektierte Link-Kommunikation von allen NCU des Link-Verbundes verwendet werden können. Link-Variablen können in Teileprogrammen, Zyklen und Synchronaktionen gelesen und geschrieben werden.

Das Lesen und Schreiben von Link-Variablen erfolgt im Hauptlauf. Dadurch erfolgt, bezogen auf die Link-Variablen, implizit eine Kanalsynchronisation, so dass alle Kanäle im gleichen Interpolator-Takt denselben Wert lesen. Im Vergleich dazu können globale Anwendervariablen (GUD) zwar auch wie die Link-Variablen kanalübergreifend verwendet werden. Da GUD-Variablen aber im Vorlauf verarbeitet werden, erfolgt hier keine implizite Kanalsynchronisation. GUD-Variablen müssen daher gegebenenfalls vom Anwender durch spezifische Programmierung synchronisiert werden.

Hinweis

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

14.1.2.3 Link-Achsen

Link-Achsen ermöglichen den Aufbau verteilter Steuerungssysteme, bei denen die Sollwerte der Maschinenachsen nicht von Kanälen der NCU, an welcher die Maschinenachsen physikalisch angeschlossen sind, erzeugt werden, sondern von Kanälen einer beliebig anderen NCU des Link-Verbundes.

Feste Zuordnung bei Link-Achsen

Die Zuordnung, auf welcher NCU die Sollwerte für welche Maschinenachse erzeugt werden, ist bei Link-Achsen durch die Maschinendatenprojektierung fest vorgegeben.

Variable Zuordnung bei Container-Linkachsen

Die Zuordnung, auf welcher NCU die Sollwerte für welche Maschinenachse erzeugt werden, ist bei Container-Linkachsen variabel. Eine Container-Linkachse kann somit wechselweise von jeder NCU des Link-Verbundes aus verfahren werden. Siehe Kapitel "Achsccontainer (Seite 794)".

Unabhängig von der oben beschriebenen Zuordnung kann auf der Sollwert-generierenden NCU die logische Maschinenachse der Link-Achse mehreren Kanälen bekannt gemacht und dann funktional wie eine lokale logische Maschinenachse gehandhabt werden, z. B. Leitwert für Lage-/Vorschubkopplungen, NCU-lokaler Achstausch.

14.1.2.4 Lead-Link-Achsen

Befinden sich im Zusammenhang mit den Kopplungsfunktionen der generischen Kopplung alle Interpolatoren, d. h. die Sollwert-erzeugenden/-verarbeitenden Kanäle, der Leit- und Folgeachsen/-spindeln auf der gleichen NCU, ist der Einsatz einer Lead-Link-Achse nicht erforderlich. Die Maschinenachsen der Leit- oder/und Folgeachsen/spindeln können dabei auch als Link-Achsen an unterschiedlichen NCU angeschlossen sein.

Anwendungsfall

Eine Lead-Link-Achse wird immer dann erforderlich, wenn sich nicht alle Interpolatoren eines Kopplungsverbandes auf derselben NCU befinden. Durch die Lead-Link-Achse wird ein Abbild der Leitachse/-spindel, bzw. der Leitwert für die lokalen Interpolatoren, auf einer anderen NCU bereitgestellt. Ausgehend von diesem Leitwert werden dann die Sollwerte der Folgeachsen/-spindeln von den auf dieser NCU liegenden Interpolatoren erzeugt.

Weitere Anwendung

Eine Lead-Link-Achse kommt auch zum Einsatz, wenn an einer Kopplung mehr Maschinenachsen beteiligt sind als auf einer NCU absolut oder relativ für die Kopplung noch zur Verfügung stehen.

Weitere Informationen

Funktionshandbuch Achsen und Spindeln, Achskopplungen

14.1.2.5 Abhängigkeiten

Damit mehrere Achsen in einem interpolatorischen Zusammenhang verfahren, ist es essenziell notwendig, dass die vom Interpolator des verfahrenenden Kanals generierten Sollwerte zum selben Zeitpunkt an die Lageregelung übergeben werden.

Bei der Übertragung der Sollwerte über den NCU-Link entsteht z. B. bei einer Link-Achse eine Verzögerung von einem Interpolator-Takt, bezogen auf den Sollwert einer lokalen Achse. Die Verzögerung muss durch eine entsprechende Vergrößerung des Puffers zwischen Interpolator und Lageregler kompensiert werden:

MD18720 \$MN_MM_SERVO_FIFO_SIZE = <Wert> = <Standardwert + Kompensationswert>

Funktion	Kompensationswert
Link-Achse	Auf allen NCUs mit Achsen, die mit einer Link- oder Container-Link-Achse interpolieren, muss ein Interpolator-Takt kompensiert werden. Kompensationswert = 1
Container-Link-Achse	
Lead-Link-Achse	Auf der NCU der Leitachse, welche die Sollwerte für die Lead-Link-Achse generiert, müssen zwei Interpolator-Takte kompensiert werden. Kompensationswert = 2

Hinweis

Link- und Container-Link-Achsen benötigen im Gegensatz zu Lead-Link-Achsen einen anderen Kompensationswert (MD18720 \$MN_MM_SERVO_FIFO_SIZE). Bei gleichzeitiger Verwendung von Link- bzw. Container-Link-Achsen und Lead-Link-Achsen innerhalb eines Link-Verbundes können die Verzögerungen nicht kompensiert werden.

14.1.2.6 Anwendungsbeispiel: Rundtaktmaschine

Anhand einer Rundtaktmaschine mit zwei NCU soll beispielhaft die Anwendung der Funktion "NCU-Link" aufgezeigt werden. Die dabei wesentlichen Einheiten der Rundtaktmaschine sind:

- Rundachse MTR für den Rundtisch
- Spindeln MS1 - MS4
- Bearbeitungsstation 1 und 2 jeweils mit den Linearachsen X1 und Z1
- Lade- und Entladestation

Alle Maschinenachsen bleiben fest ihrer jeweiligen NCU zugeordnet. Im Bearbeitungsprogramm der jeweiligen NCU werden immer dieselben Achsen (X und Z) und Spindeln (S1) adressiert.

Der Werkstückzustand ist schematisch nach dem jeweiligen Bearbeitungsschritt dargestellt.

Für jeden Bearbeitungsschritt wird der Rundtisch (MTR) gegen den Uhrzeigersinn um eine Position weitergeschaltet. Die Spindeln werden dadurch bei jedem Bearbeitungsschritt einer anderen Station zugeordnet. Die wechselnden Beziehungen der in den Kanälen definierten Spindeln zu den Maschinenachsen werden in den NCU durch einen **Achscontainer** abgebildet.

Liegt die Maschinenachse einer Spindel nicht auf der eigenen NCU, werden die Spindelsollwerte per **Link-Kommunikation** auf die zugehörige NCU übertragen und dort an die Maschinenachse ausgegeben.

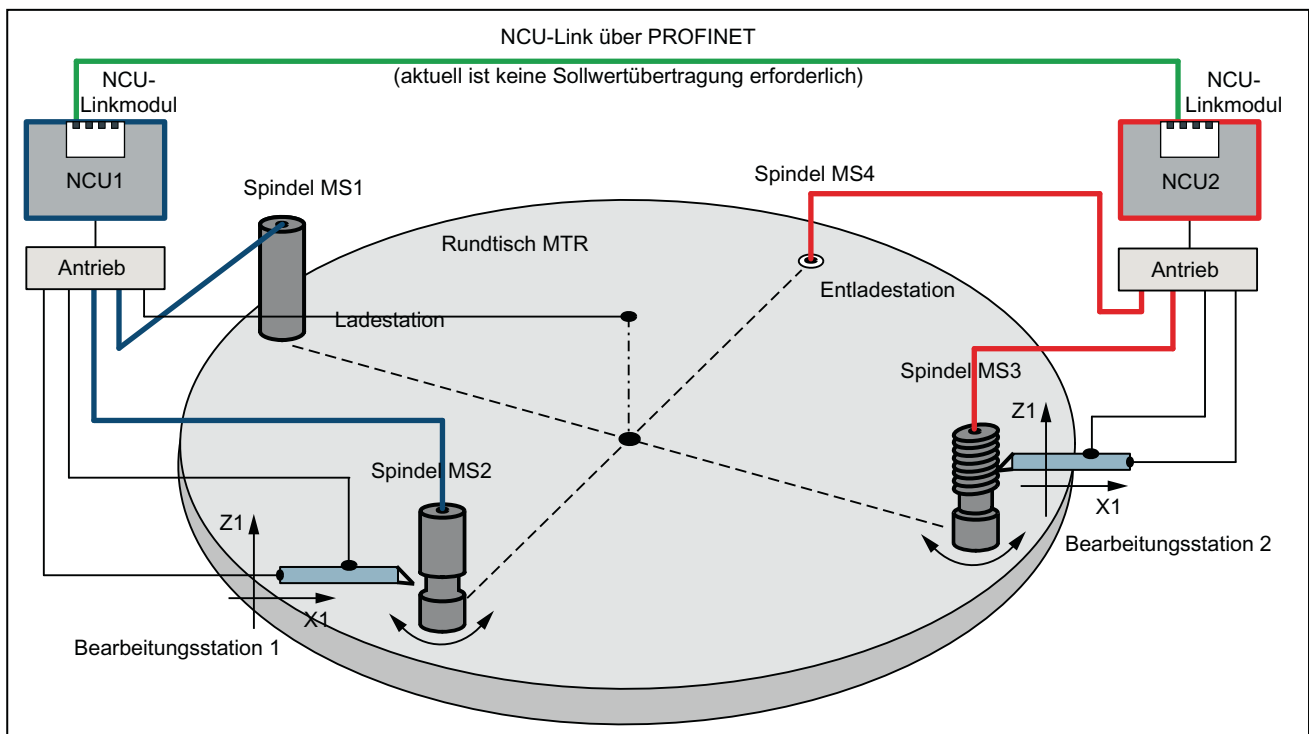


Bild 14-2 Bild 1: Ausgangssituation

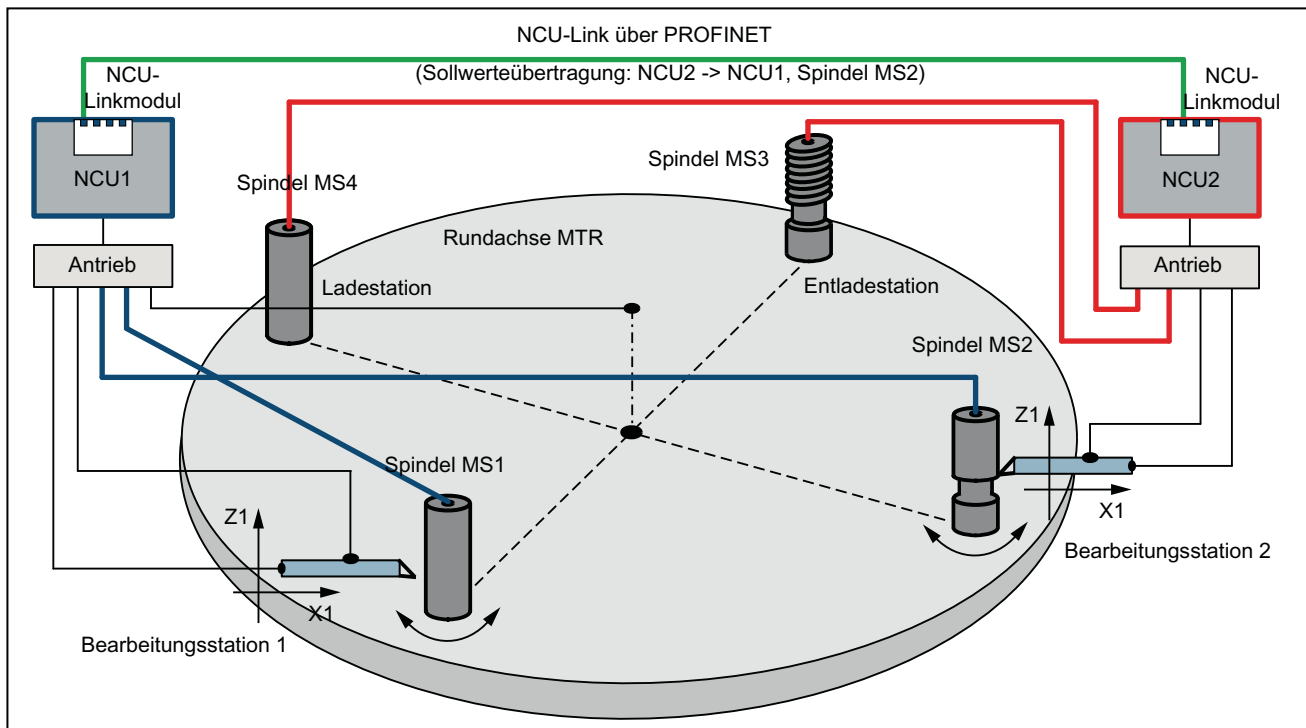


Bild 14-3 Bild 2: Stellung nach Drehung um eine Position

Parametrierung (schematisch)

Allgemein	
In den Teileprogrammen beider NCU programmierte Kanalachsen: X, Z, S1	
NCU 1	In der NCU definierte Maschinenachsen:
	Lokal: X1, Z1
	Achscontainer: MS1, MS2
NCU 2	In der NCU definierte Maschinenachsen:
	Lokal: X1, Z1
	Achscontainer: MS3, MS4

	Ausgangsstellung (Bild 1)	Drehung der Rundachse MTR (Rundtisch) um eine Position (Bild 2)
NCU 1	Bearbeitungsstation 1: X1, Z1, MS2	
	Abbildung der im Teileprogramm programmierten Kanalachsen:	
	X → X1 und Z → Z1	X → X1 und Z → Z1
	S1 → MS2	S1 → MS1
NCU 2	Bearbeitungsstation 2: X1, Z1, MS2	
	Abbildung der im Teileprogramm programmierten Kanalachsen:	
	X → X1 und Z → Z1	X → X1 und Z → Z1
	S1 → MS3	S1 → MS2 (Link-Achse)

14.2 NCU-Link

14.2.1 Link-Kommunikation

14.2.1.1 Allgemeine Informationen

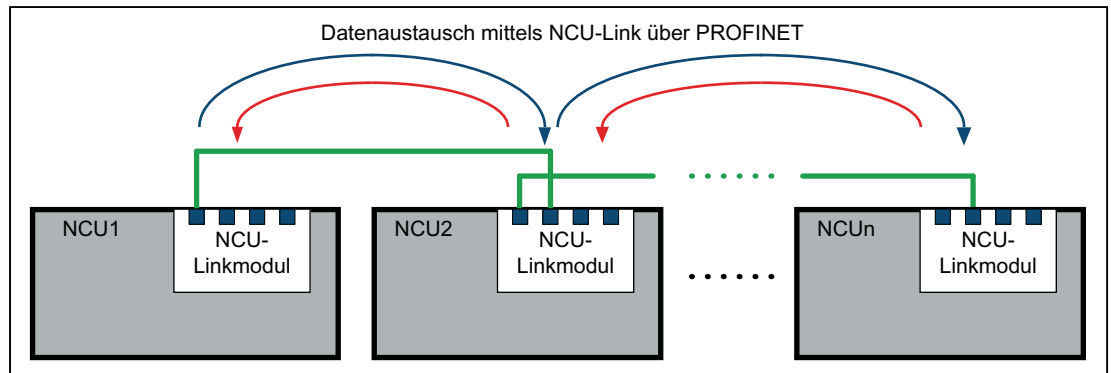


Bild 14-4 Link-Kommunikation (Prinzip)

Die NCU-Link-Kommunikation ermöglicht einen Interpolatortakt-synchronen NCU-übergreifenden Datenaustausch für folgende Applikationsaufgaben:

- NCU-übergreifende Link-Variablen \$A_DLx
Alle an der NCU-Link-Kommunikation beteiligten NCU haben eine gemeinsame Sicht auf die Link-Variablen, da diese über NCU-Link Interpolatortakt-synchron zwischen den NCU des Link-Verbundes ausgetauscht werden.
- NCU-übergreifendes Verfahren von Achsen und Spindeln mittels Link-Achsen und Container-Link-Achsen
Werden von einem Kanal einer NCU Achsen einzeln oder interpolatorisch verfahren, können über NCU-Link Sollwerte auch an Achsen übertragen werden, die physikalisch an anderen NCU des Link-Verbundes angeschlossen sind. Diese Achsen werden als Link-Achsen bezeichnet.
- NCU-übergreifende Leitwertkopplung von Achsen und Spindeln mittels Lead-Link-Achsen
NCU1 verfährt Achse X1 (Leitachse), die Sollwerte werden über NCU-Link auf eine Link-Achse der NCU2 übertragen (Lead-Link-Achse). Die Kopplung der Achse X2 erfolgt in NCU2 auf diese Lead-Link-Achse. Achse X2 ist somit indirekt Folgeachse von X1.

Datenübertragung

Zwischen den an der Link-Kommunikation beteiligten NCU finden, abhängig von den aktiven Funktionen, folgende zyklischen und nicht zyklischen Datenübertragungen statt:

- Hochpriore, zyklische Datenübertragung:
 - Sollwerte, Istwerte und Statussignale der Link-Achsen
 - NCU-Statussignale
- Hochpriore, **nicht** zyklische, **nicht** verdrängbare Datenübertragung:
 - Nicht sicherheitsrelevante Daten im Rahmen der Funktion Safety Integrated
 - Zustände der Containerachsen während einer Achscontainer-Drehung
- Niederpriore, **nicht** zyklische, verdrängbare Datenübertragung (Aufzählung mit abnehmender Priorität):
 - Link-Variablen
 - Warmstartanforderungen
 - Aktivierung von Achscontainer-Drehungen
 - Änderungen von NCU-globalen Maschinen- und Settingdaten
 - Wirksamsetzen axialer Maschinendaten von Link-Achsen
 - Alarme

Verdrängung

Können bei der niederprioren, nicht zyklischen, verdrängbaren Datenübertragung nicht alle Anforderungen in einem Interpolator-Takt gesendet werden, verdrängt die Anforderung mit höherer Priorität diejenige mit niederer Priorität. Die Anforderung mit niederer Priorität wird dann zu einem späteren Zeitpunkt gesendet.

Mengengerüst: NCU 7x0.3 PN

Standardmäßig können maximal 3 NCU zu einem Link-Verbund zusammengeschaltet werden.

Hinweis

Ein NCU-Link-Verbund mit mehr als 3 NCU ist projektspezifisch auf Anfrage bei ihrem regionalen Siemens Ansprechpartner möglich. Ohne projektspezifische Ergänzungen werden mehr als 3 NCU mit Alarm 380020 abgelehnt.

NCU-Link und Safety Integrated

Das folgende Bild zeigt eine Konstellation mit zwei NCU und zwei Maschinenachsen, wovon die Maschinenachse MA2 der NCU2 als Link-Achse von NCU1 aus verfahren wird. Beide Achsen werden durch die Funktion Safety Integrated sicherheitsgerichtet überwacht.

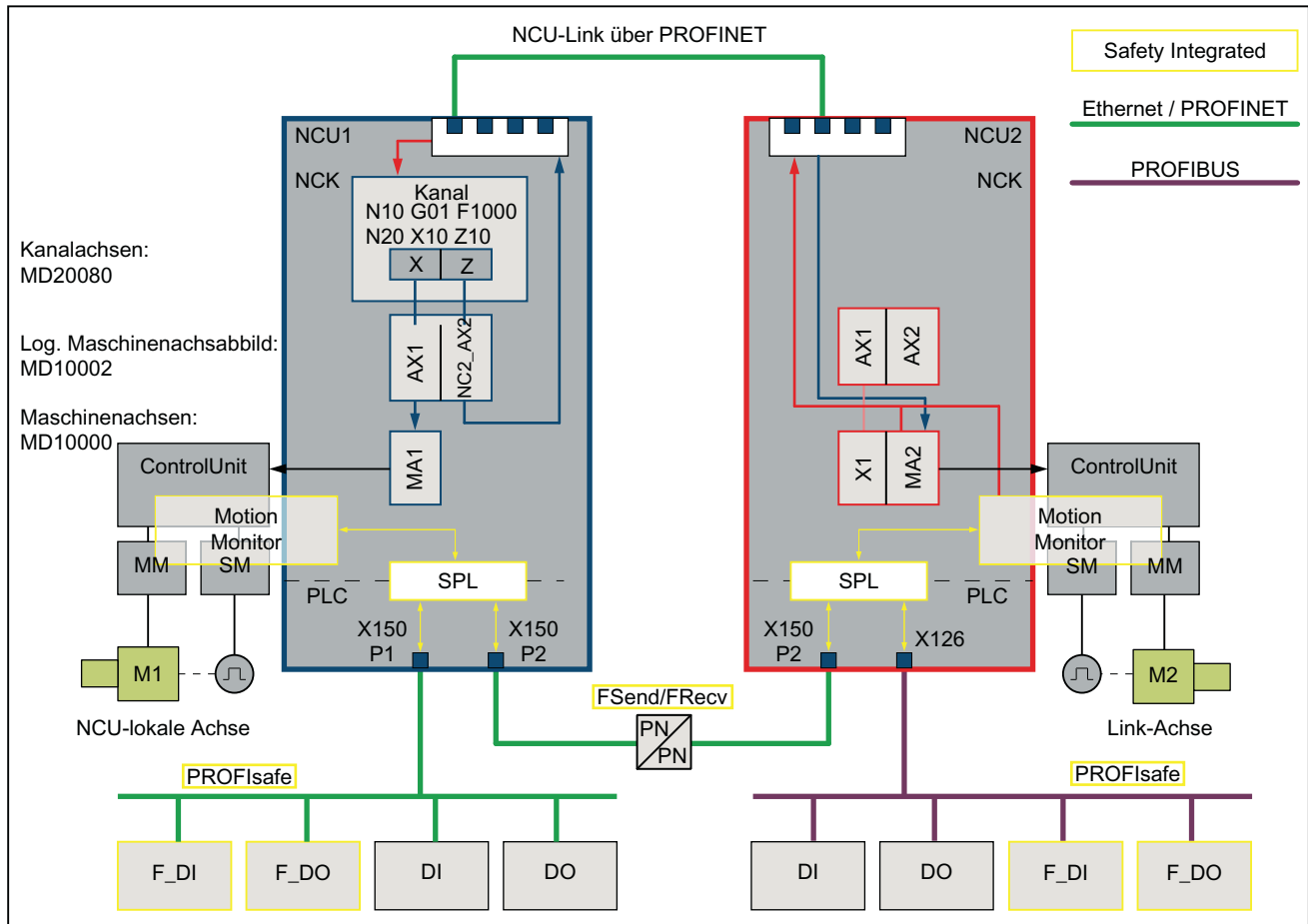


Bild 14-5 NCU-Link und Safety Integrated

Sicherheitsgerichtete Eingangssignale (F_DI) können von beiden NCU erfasst, über die sichere programmierbare Logik (SPL) verknüpft und über die sicherheitsgerichtete CPU-CPU-Kommunikation (FSend/FRecv) ausgetauscht werden.

Im Rahmen der Funktion Safety Integrated ermöglicht die NCU-Link-Kommunikation die einkanalige **nicht sicherheitsrelevante** Übertragung von Link-Achsdaten für Safety Integrated-unterstützende Funktionen.

Beispiele:

- interpolatorisches Abbremsen bei implizitem Stop D aller im Kanal verfahrenen Achsen, einschließlich Link-Achsen.
- reduzieren der Geschwindigkeit aller im Kanal verfahrenen Achsen, einschließlich Link-Achsen, bei einem Wechsel der sicheren Geschwindigkeit (SG).

Safety Integrated Abnahmetest und NCU-Link

Mit dem Acceptance Test Wizzard (ATW) wird der Abnahmetest prinzipiell für jede NCU separat durchgeführt. Bei Link-Achsen werden Alarme aber nur auf der Heimat-NCU der Achse angezeigt. Damit der ATW auch Alarme von Link-Achsen überprüfen kann, müssen dem ATW die über NCU-Link verbundenen Safety-relevanten NCU bekannt gemacht werden. Dazu müssen diese in die Datei NETNAMES.INI auf dem PG/PCU, auf welcher der ATW abläuft, eingetragen werden.

Beispiel: Einträge in Datei NETNAMES.INI für NCU-Link mit zwei Safety-relevanten NCU

Datei: NETNAMES.INI (Auszug)

```
[own]
owner= HMI_1
[conn HMI_1]
conn_1= NCU_1
conn_2= NCU_2
[param HMI_1]
mmc_address= 1
[param network]
[param NCU_1]
nck_address= 192.168.214.2,LINE=10,NAME=/NC,SAP=030d,PROFILE=CLT1__CP_L4_INT
plc_address= 192.168.214.2,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_L4_INT
ADDRESS2= 192.168.214.2,LINE=10,NAME=/CP,SAP=0501,PROFILE=CLT2__CP_L4_INT
ADDRESS10= 192.168.214.2,LINE=10,NAME=/
DRIVE_00_000,SAP=0201,SUBNET=0000-00000000:000,PROFILE=CLT2__CP_L4_INT
ADDRESS11= 192.168.214.2,LINE=10,NAME=/DRIVE_03_003,SAP=0900,PROFILE=CLT2__CP_L4_INT
ADDRESS12= 192.168.214.2,LINE=10,NAME=/DRIVE_03_011,SAP=0B00,PROFILE=CLT2__CP_L4_INT
ADDRESS13= 192.168.214.2,LINE=10,NAME=/DRIVE_03_012,SAP=0C00,PROFILE=CLT2__CP_L4_INT
ADDRESS14= 192.168.214.2,LINE=10,NAME=/DRIVE_03_013,SAP=0D00,PROFILE=CLT2__CP_L4_INT
ADDRESS15= 192.168.214.2,LINE=10,NAME=/DRIVE_03_014,SAP=0E00,PROFILE=CLT2__CP_L4_INT
ADDRESS16= 192.168.214.2,LINE=10,NAME=/DRIVE_03_015,SAP=0F00,PROFILE=CLT2__CP_L4_INT
name=Machine_1
[param NCU_2]
nck_address= 192.168.214.1,LINE=10,NAME=/NC,SAP=030d,PROFILE=CLT1__CP_L4_INT
plc_address= 192.168.214.1,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_L4_INT
ADDRESS2= 192.168.214.1,LINE=10,NAME=/CP,SAP=0501,PROFILE=CLT2__CP_L4_INT
ADDRESS10= 192.168.214.1,LINE=10,NAME=/
DRIVE_00_000,SAP=0201,SUBNET=0000-00000000:000,PROFILE=CLT2__CP_L4_INT
ADDRESS11= 192.168.214.1,LINE=10,NAME=/DRIVE_03_003,SAP=0900,PROFILE=CLT2__CP_L4_INT
ADDRESS12= 192.168.214.1,LINE=10,NAME=/DRIVE_03_011,SAP=0B00,PROFILE=CLT2__CP_L4_INT
ADDRESS13= 192.168.214.1,LINE=10,NAME=/DRIVE_03_012,SAP=0C00,PROFILE=CLT2__CP_L4_INT
ADDRESS14= 192.168.214.1,LINE=10,NAME=/DRIVE_03_013,SAP=0D00,PROFILE=CLT2__CP_L4_INT
ADDRESS15= 192.168.214.1,LINE=10,NAME=/DRIVE_03_014,SAP=0E00,PROFILE=CLT2__CP_L4_INT
ADDRESS16= 192.168.214.1,LINE=10,NAME=/DRIVE_03_015,SAP=0F00,PROFILE=CLT2__CP_L4_INT
name=Machine_2
```

Weitere Informationen

Die Funktion "Safety Integrated" ist ausführlich beschrieben in:

Funktionshandbuch SINAMICS S120 Safety Integrated

14.2.1.2 Link-Modul

Die NCU-Link-Kommunikation erfolgt über Link-Module. Ein Link-Modul ist eine optionale PROFINET-Baugruppe für isochrone Echtzeit-Kommunikation (IRT) über Ethernet. Das Link-Modul kann nur für die NCU-Link-Kommunikation verwendet werden. Eine Verwendung des Link-Moduls zur allgemeinen PROFINET-Kommunikation ist nicht möglich.

Für das Link-Modul wird an der NCU-Baugruppe der Option Slot benötigt.

Hinweis

An einer NCU-Baugruppe ist nur ein Option Slot vorhanden. Die parallele Verwendung eines NCU-Linkmoduls und anderer optionaler Baugruppen schließen sich daher aus.

Link-Modul und NCU-Baugruppen

Für die NCU-Baugruppen "NCU710.3 PN", "NCU720.3 PN" und "NCU730.3 PN" stehen als Link-Modul das Communication Board Ethernet CBE30-2 zur Verfügung. Für das Link-Modul wird an der NCU-Baugruppe der Option Slot benötigt.

Weitere Informationen

Gerätehandbuch NCU7x0.3 PN; Anschließbare Komponenten > CBE30-2

14.2.1.3 Parametrierung: NC-Systemtakte

Als grundlegende Voraussetzung für die Link-Kommunikation müssen folgende Systemtakte in **allen** am NCU-Verbund beteiligten NCUs gleich eingestellt werden:

- Systemgrundtakt
- Lagereglertakt
- Interpolatortakt

Systemgrundtakt

Als Systemgrundtakt wird der im STEP7-Projekt für die isochrone Kommunikation eingestellte DP-Takt übernommen. Der aktuelle Systemgrundtakt wird angezeigt in Maschinendatum:

MD10050 \$MN_SYSCLOCK_CYCLE_TIME

Hinweis

Manueller Abgleich über mehrere Kommunikationsbusse

Sind an einer NCU mehrere isochrone Kommunikationsbusse (PROFIBUS 1 ... n, PROFINET 1 ... m) projektiert, muss in STEP7 HW-Konfig für jeden Kommunikationsbus die gleiche Zykluszeit eingestellt werden.

Abhängigkeit zum Lagereglertakt

Bei SINUMERIK 840D sl ist das Verhältnis zwischen Systemgrundtakt und Lagereglertakte fest (1:1) und kann nicht geändert werden. Da im Zusammenhang mit NCU-Link nur bestimmte Lagereglertakte eingestellt werden können, dürfen auch nur diese Lagereglertakte als Systemgrundtakt bzw. DP-Zykluszeit eingestellt werden. Siehe nächsten Absatz "Lagereglertakt".

Lagereglertakt

Der Lagereglertakt wird im Verhältnis zum Systemgrundtakt eingestellt. Das Verhältnis ist für SINUMERIK 840D sl fest (1:1) und kann nicht geändert werden. Der aktuelle Lagereglertakt wird angezeigt im Maschinendatum:

MD10061 \$MN_POSCTRL_CYCLE_TIME

Hinweis

Erlaubte Lagereglertakte

Für NCU-Link dürfen, abhängig von der Anzahl der NCUs im Link-Verbund, nur folgende Lagereglertakte [ms] eingestellt werden:

- **2 NCU:** 2.0, 2.5, 3.0, 3.5, 4.0
- **3 NCU:** 3.0, 3.5, 4.0

Siehe dazu Kapitel "Projektierung (Seite 780)".

Interpolatortakt

Der Interpolatortakt wird im Verhältnis zum Systemgrundtakt eingestellt. Die Einstellung erfolgt über das Maschinendatum:

MD10070 \$MN_IPO_SYSCLOCK_TIME_RATIO

Der aktuelle Interpolatortakt wird angezeigt in Maschinendatum:

MD10071 \$MN_IPO_CYCLE_TIME

Einstellhinweise

Takteinstellungen

Es wird empfohlen, folgende Einstellungen vorzunehmen:

- Die Standardeinstellung für den Rechenzeitanteil des NC von 90% sollte beibehalten werden:
MD10185 \$MN_NCK_PCOS_TIME_RATIO
- Die Systemtakte sind so einzustellen, dass die **durchschnittliche** Systemauslastung durch Interpolator und Lageregler im normalen Programmbetrieb nicht mehr als 60% beträgt. Als **Maximalwert** sollten 90% nicht überschritten werden.

Hinweis

In **Ausnahmefällen** können Maximalwerte >100% angezeigt werden.

Geringste zyklische Kommunikationslast

Die geringste zyklische Kommunikationslast im Rahmen der Link-Kommunikation ergibt sich bei einem Taktverhältnis von Interpolator- zu Lagereglertakt von 1 : 1. Mit aktivierter Antriebsfunktion "Dynamic Servo Control (DSC)" ist dies im Allgemeinen die effektivste Einstellung.

Nachteil: keine Telegrammwiederholungen im Rahmen der Link-Kommunikation möglich.

Zeitraster der Aktualisierung der NC/PLC-Nahtstelle

Für alle am Link-Verbund beteiligten NCU wird im Hochlauf folgende Einstellung aktiv:

MD18000 \$MN_VDI_UPDATE_IN_ONE_IPO_CYCLE = 1

Dadurch wird die NC/PLC-Nahtstelle in nur einem Interpolatortakt vollständig gelesen und geschrieben.

Systemauslastung

Die Systemauslastung wird auf der Bedienoberfläche SINUMERIK Operate angezeigt unter: Bedienbereich "Diagnose" > "ETC-Taste (>)" > "Systemauslastung"

Interpolatorakt und Link-Variablen

Bei der Parametrierung der Systemtakte ist neben den Maximalwerten der Systemauslastung im "worst case", die voraussichtliche maximale Rechenzeitbelastung der Interpolator-Ebene durch die Übertragung der Link-Variablen im Produktivbetrieb zu berücksichtigen.

WARNUNG

Steuerungsausfall wegen Rechenzeitüberlauf auf der Interpolator-Ebene

Wird bei der Parametrierung der NC der Interpolatortakt zu kurz eingestellt, kann es im Produktivbetrieb aufgrund der zusätzlichen Rechenzeitbelastung durch die Übertragung von Link-Variablen zu einem Rechenzeitüberlauf auf der Interpolatorebene kommen.

Weitere Informationen: Diagnosehandbuch; NC-Alarme > Alarm 4240 Rechenzeitüberlauf auf der IPO- oder Lageregler-Ebene.

14.2.1.4 Parametrierung: Link-Kommunikation

NC-spezifischen Maschinendaten

Nummer	Bezeichner \$MN_	Bedeutung
MD12510	NCU_LINKNO	Eindeutige numerische Kennung der NCU innerhalb des Link-Verbundes. Die Kennungen müssen von 1 beginnend lückenlos in aufsteigender Reihenfolge vergeben werden. Wertebereich: 1, 2, ... maximale NCU-Nummer Hinweis Die NCU, der als NCU-Kennung der Wert 1 zugewiesen wird, ist die Master-NCU des Link-Verbundes. Die Parametrierung von Link-Achsen und Achscontainern sind ausschließlich über die Maschinen- und Settingdaten) der Master-NCU vorzunehmen.
MD18780	MM_NCU_LINK_MASK.Bit 0	Aktivierung der Link-Kommunikation
MD18781	NCU_LINK_CONNECTIONS	Anzahl interner Link-Verbindungen Hinweis Es wird empfohlen, den Standardwert 0 (Ermittlung der Anzahl durch die NC) beizubehalten.
MD18782	MM_LINK_NUM_OF_MODULES	Anzahl der über NCU-Link miteinander verbundenen NCUs.

14.2.1.5 Projektierung

Für jede unterstützte Kombination aus NCU-Anzahl und Lagereglertakt eines Link-Verbundes (siehe Kapitel "Parametrierung: NC-Systemtakte (Seite 777)") werden mit der NC-Systemsoftware spezifische Konfigurationen mitgeliefert.

Im Systemhochlauf wird, in Abhängigkeit von den in den Maschinendaten parametrierten Werten, die entsprechende Konfiguration geladen:

- MD18782 \$MN_MM_LINK_NUM_OF_MODULES (NCU-Anzahl des Link-Verbundes)
- MD10061 \$MN_POSCTRL_CYCLE_TIME (Lagereglertakt)

Hinweis

Bei Anwendungsfällen in denen die mitgelieferten Standardkonfigurationen nicht verwendet werden können, wenden sie sich bitte an ihren regionalen Siemens Ansprechpartner.

14.2.1.6 Verkabelung der NCUs

Die numerische Reihenfolge der NCUs innerhalb eines Link-Verbunds ist in jeder NCU über folgendes Maschinendatum festgelegt:

MD12510 \$MN_NCU_LINKNO = <NCU-Nummer>, mit NCU-Nummer = 1 ... max. NCU-Nummer

Verkabelung

Die Verkabelung der NCU-Linkmodule ist in der Reihenfolge der NCU-Nummern, ausgehend von NCU1, nach folgendem Schema vorzunehmen: NCU(n), Port 0 → NCU(n+1), Port 1

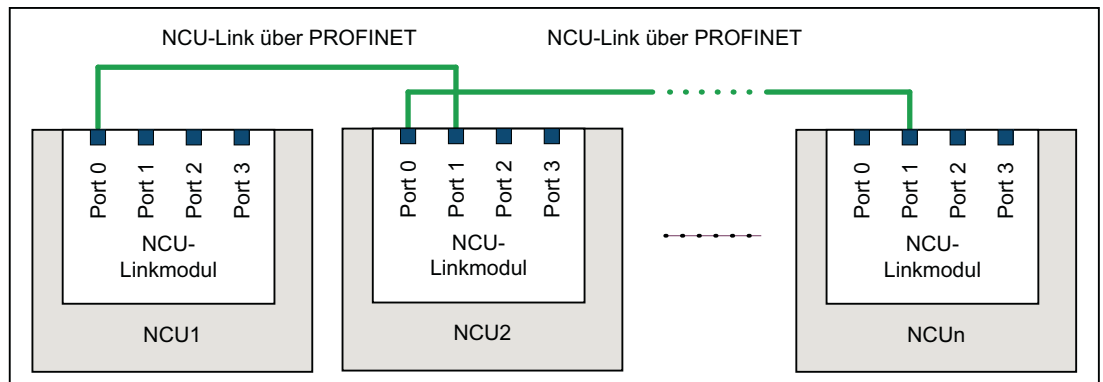


Bild 14-6 Verkabelungsschema NCU-Link

14.2.1.7 Aktivierung

Die Aktivierung der Link-Kommunikation erfolgt über folgendes Maschinendatum:

MD18780 \$MN_MM_NCU_LINK_MASK, Bit 0 = 1

Hinweis

Aktivierungszeitpunkt

Es wird empfohlen, die Aktivierung der Link-Kommunikation erst nach vollständiger Inbetriebnahme der gesamten Funktion auf allen beteiligten NCU vorzunehmen.

14.2.2 Link-Variablen

Komplexe Anlagen mit mehreren NCU erfordern zur systemweiten Koordinierung der Fertigungsabläufe einen zyklischen Austausch anwenderspezifischer Daten zwischen den NCU. Der Datenaustausch erfolgt über die Link-Kommunikation und einem speziellen Speicherbereich, dem pro NCU vorhandenen Link-Variablen-Speicher.

Sowohl die Größe und als auch die Datenstruktur des Link-Variablen-Speichers können anwenderspezifisch festgelegt werden. Die Adressierung der im Link-Variablen-Speicher abgelegten Daten erfolgt über spezielle Link-Variablen \$A_DLx.

Link-Variablen sind somit systemglobale Anwendervariablen, die bei projektierte Link-Kommunikation von allen NCU 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

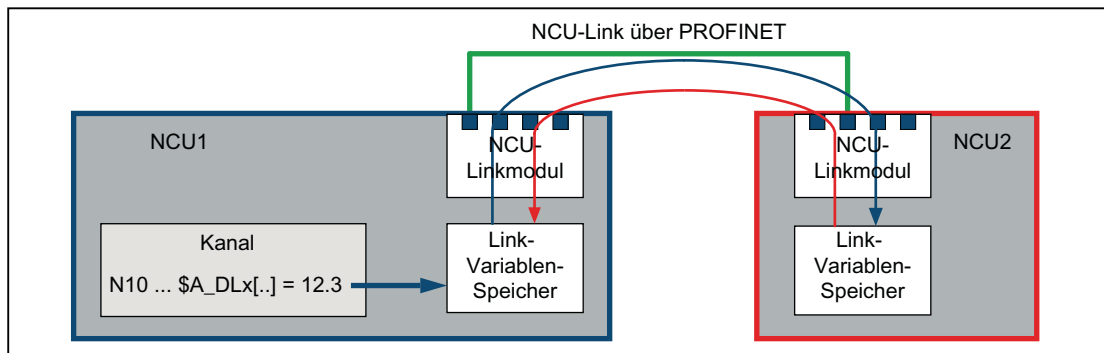


Bild 14-7 Link-Variablen

Link-Variablen als NCU-globale Anwendervariablen

Hinweis

Ist eine NCU kein Teilnehmer eines Link-Verbundes bzw. werden die Link-Variablen nicht zur Kommunikation über NCU-Link benötigt, können die Link-Variablen als NCU-globale Anwendervariablen verwendet werden.

14.2.2.1 Eigenschaften des Link-Variablen-Speichers

Parametrieren der Speichergröße

Die Größe des Link-Variablen-Speichers in Byte wird über folgendes Maschinendatum eingestellt:

MD18700 \$MN_MM_SIZEOF_LINKVAR_DATA (Größe des Link-Variablen-Speichers)

Die Größe des Link-Variablen-Speichers sollte für alle am Link-Verbund beteiligten NCUs gleich eingestellt sein. Bei unterschiedlichen Speichergrößen, wird der größte parametrierte Wert verwendet.

Initialisierung

Nach dem Hochlauf einer NCU ist der Link-Variablen-Speicher mit 0 initialisiert.

Struktur

Der Link-Variablen-Speicher ist vom System aus gesehen ein unstrukturierter Speicherbereich der für die Link-Kommunikation zur Verfügung steht. Die Strukturierung des Link-Variablen-Speichers erfolgt ausschließlich durch den Anwender / Maschinenhersteller. Entsprechend der definierten Datenstruktur erfolgt der Zugriff auf den Link-Variablen-Speicher über Datenformat-spezifische Link-Variablen.

Systemweiter Abgleich

Nach einem Schreibvorgang auf einen Link-Variablen-Speicher wird die Datenänderung in die Link-Variablen-Speicher alle anderen am Link-Verbund beteiligten NCU übertragen. Die Zeitverzögerung bis zur Aktualisierung der Link-Variablen-Speicher durch die Link-Kommunikation beträgt im Normalfall zwei Interpolator-Takte.

14.2.2.2 Eigenschaften der Link-Variablen

Der Zugriff auf den Link-Variablen-Speicher erfolgt über die folgenden Datenformat-spezifischen Link-Variablen:

Datentyp ¹⁾	Bezeichnung	Datenformat ²⁾	Bytes ²⁾	Index i ³⁾	Wertebereich
UINT	\$A_DLB[i]	BYTE	1	$i = n * 1$	0 ... 255
INT	\$A_DLW[i]	WORD	2	$i = n * 2$	-32768 ... 32767
INT	\$A_DLD[i]	DWORD	4	$i = n * 4$	-2147483648 ... 2147483647
REAL	\$A_DLR[i]	REAL	8	$i = n * 8$	$\pm(2,2*10^{-308} \dots 1,8*10^{+308})$

1) Datentyp der Link-Variablen bei der Verwendung im Teileprogramm / Zyklus

2) Datenformat der Link-Variablen bzw. die Anzahl Bytes die von der Link-Variablen im Link-Variablen-Speicher adressiert werden.

3) Bezüglich des Index i ist folgendes zu beachten:

- Der Index i ist ein Byte-Index bezogen auf den Anfang des Link-Variablen-Speichers.
- Der Index muss so gewählt werden, dass die im Link-Variablen-Speicher adressierten Bytes auf einer Datenformatgrenze liegen \Rightarrow Index $i = n * \text{Bytes}$, mit $n = 0, 1, 2, \dots$
 - $\$A_DLB[i]: i = 0, 1, 2, \dots$
 - $\$A_DLW[i]: i = 0, 2, 4, \dots$
 - $\$A_DLD[i]: i = 0, 4, 8, \dots$
 - $\$A_DLR[i]: i = 0, 8, 16, \dots$

Schreiben

Das Schreiben einer Link-Variablen wird Hauptlauf-synchron ausgeführt.

Lesen

Beim Lesen einer Link-Variablen wird Vorlaufstopp ausgelöst.

Überprüfungen

Bezüglich der Link-Variablen und des Link-Variablen-Speichers finden folgende Überprüfungen statt:

- Einhaltung der Wertebereichsgrenzen
- Zugriff auf Formatgrenze
- Einhaltung des definierten Speicherbereichs des Link-Variablen-Speichers

Die Vermeidung der folgenden Fehler liegt ausschließlich in der Verantwortung des Anwenders / Maschinenherstellers:

- Zugriff mit falschem Datenformat
- Zugriff auf falsche Adresse (Index i)

- Gegenseitiges Überschreiben des gleichen Datums von mehreren Kanälen einer NCU oder von unterschiedlichen NCU
 - Lesen eines Datums bevor es von einem Kanal der eigenen NCU oder einer anderen NCU aktualisiert wurde
-

Hinweis

Datenkonsistenz

Die Sicherstellung der Datenkonsistenz innerhalb des Link-Variablen-Speichers, sowohl NCU-lokal als auch NCU-übergreifend, liegt ausschließlich in der Verantwortung des Anwenders / Maschinenherstellers.

14.2.2.3 Schreibelemente

Bei Schreibzugriffen auf den Link-Variablen-Speicher (z. B. \$A_DLB[4] = 21) wird für die systeminterne Verwaltung des Schreibvorganges ein sogenanntes Link-Variablen-Schreibelement benötigt. Die maximale Anzahl von Schreibelementen die pro Interpolator-Takt zur Verfügung stehen, wird über folgendes Maschinendatum eingestellt:

MD28160 \$MC_MM_NUM_LINKVAR_ELEMENTS

Die maximale Anzahl von Schreibelementen begrenzt somit die Anzahl von Link-Variablen die pro Interpolator-Takt geschrieben werden können.

14.2.2.4 Zeitverhalten beim Schreiben

Das Schreiben der Link-Variablen erfolgt Hauptlauf-synchron. Den neuen Wert können die anderen Kanäle der eigenen NCU spätestens im nächsten Interpolator-Takt lesen. Im eigenen Kanal kann der neue Wert im nächsten Satz gelesen werden.

Die Kanäle der anderen NCU des Link-Verbundes sehen den neuen Wert nach zwei Interpolator-Takten. Aufgrund der begrenzten Bandbreite kann es aber zu Verzögerungen bei der Übertragung von Schreibaufträgen auf die anderen NCU des Link-Verbundes (Message-Delay) kommen. Ursachen für einen Message-Delay können sein:

- Schreiben einer großen Anzahl von Link-Variablen in einem Interpolator-Takt
- Schreiben von Link-Variablen und die Anforderung einer Achscontainer-Drehung im gleichen Interpolator-Takt
- Schreiben von Link-Variablen und die Übertragung eines Alarms im gleichen Interpolator-Takt

14.2.2.5 Systemvariable

NC-spezifische Systemvariable

Bezeichner	Bedeutung
\$AN_LINK_TRANS_RATE_LAST	Anzahl Schreibaufträge, die im letzten Interpolator-Takt noch frei waren.
\$AN_LINK_TRANS_RATE_LAST_SUM[<n>]	Anzahl Schreibaufträge, die im letzten Interpolator-Takt in Senderichtung zur angegebenen NC <n> (NCU-Nummer) noch frei waren.
\$AN_LINK_CONN_SIZE_LINKVAR	Anzahl Bytes, die bei einem Schreibauftrag für eine Link-Variablen übertragen werden.
\$AN_LINK_CONN_SND[<n>]	Max. Byte-Anzahl, die pro Interpolator-Takt von der aktuellen zur angegebenen NCU übertragen werden können
\$AN_LINK_CONN_RCV[<n>]	Max. Byte-Anzahl, die pro Interpolator-Takt von der angegebenen zur aktuellen NCU übertragen werden können
<n>: NCU-Nummer entsprechend MD12510 \$MN_NCU_LINKNO der jeweiligen NCU	

Hinweis zu: \$AN_LINK_CONN_SIZE_LINKVAR

Durch das Schreiben einer Link-Variablen werden über die nicht-zyklische Link-Kommunikation die in \$AN_LINK_CONN_SIZE_LINKVAR angezeigte Anzahl an Bytes übertragen. Die Anzahl ist unabhängig vom Format der Link-Variablen.

Die max. Anzahl von Schreibaufträgen, die pro Interpolator-Takt zur angegebenen NCU übertragen werden können, berechnen sich zu:

$$\text{Max. Anzahl Schreibaufträge} = \frac{\$AN_LINK_CONN_SND[<n>]}{\$AN_LINK_CONN_SIZE_LINKVAR}$$

Kanal-spezifische Systemvariable

Bezeichner	Bedeutung
\$A_LINK_TRANS_RATE ¹⁾	Anzahl Schreibaufträge, die noch im aktuellen Interpolator-Takt übertragen werden können.
1) Anwendungsbeispiel siehe Kapitel: "Synchronisation eines Schreibauftrags (Seite 785)"	

14.2.2.6 Synchronisation eines Schreibauftrags

Ist es in bestimmten Anwendungsfällen erforderlich, dass der neue Wert einer Link-Variable in genau zwei Interpolator-Takten auf die anderen NCU des Link-Verbundes übertragen wird, muss das Schreiben der Link-Variable in einer Synchronaktion erfolgen. In der Synchronaktion wird das Schreiben der Link-Variable nur dann ausgeführt, wenn im aktuellen Interpolator-Takt der Schreibauftrag noch durchgeführt werden kann. Die Systemvariable \$A_LINK_TRANS_RATE enthält dazu die Anzahl der Schreibaufträge, die noch im aktuellen Interpolator-Takt durchgeführt werden können.

Im folgenden Beispiel soll eine Link-Variable vom Datentyp WORD (2 Byte) und eine Link-Variable vom Datentyp DWORD (4 Byte) geschrieben werden:

Programmbeispiel

```
N120 WHEN $A_LINK_TRANS_RATE > 0 DO $A_DLW[0] = 9
N125 WHEN $A_LINK_TRANS_RATE > 0 DO $A_DLD[2] = 7
N130 G4 F1
```

Die Synchronaktion in N120 wird nur dann ausgeführt, wenn der Schreibauftrag im gleichen Interpolator-Takt auf die anderen NCU des Link-Verbundes übertragen werden kann. Dabei wird im selben Interpolator-Takt auch die Systemvariable \$A_LINK_TRANS_RATE dekrementiert, so dass für die Synchronaktion im Folgensatz N125 der aktualisierte Wert zur Verfügung steht.

14.2.2.7 Beispiel: Aufteilung des Link-Variablen-Speichers

Zur Link-Kommunikation werden folgende Daten eingerichtet:

Datenformat	Anzahl	Bytes pro Datum	benötigte Bytes
BYTE	2	1	2
WORD	1	2	2
DWORD	3	4	12
REAL	1	8	8
benötigte Größe des Link-Variablen-Speichers:			24

Speicherstruktur

Die Daten werden unter Berücksichtigung der Datenformatgrenzen im Link-Variablen-Speicher folgendermaßen angeordnet:

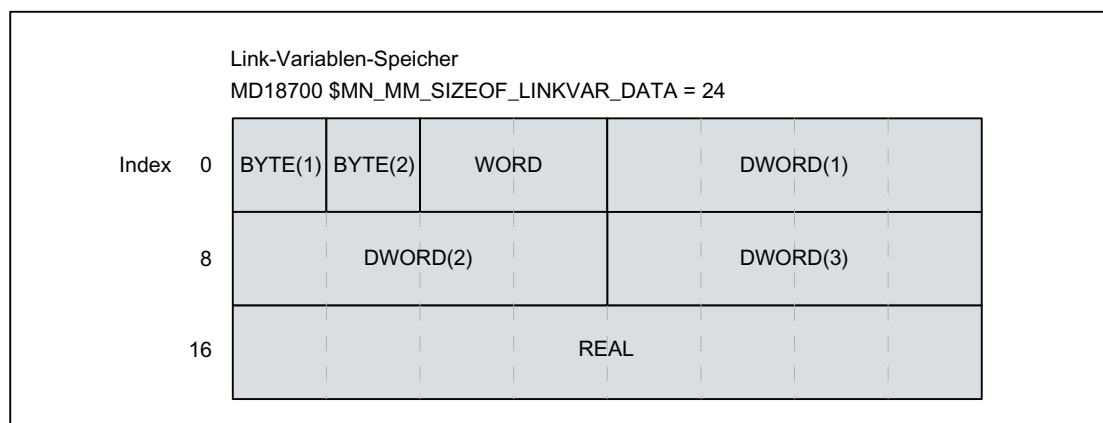


Bild 14-8 Beispiel: Speicheraufteilung des Link-Variablen-Speichers

Hinweis

Speicherstruktur

Die Anordnung der Daten im Link-Variablen-Speicher ist prinzipiell beliebig und könnte unter Berücksichtigung der Datenformatgrenzen auch in anderer Form erfolgen.

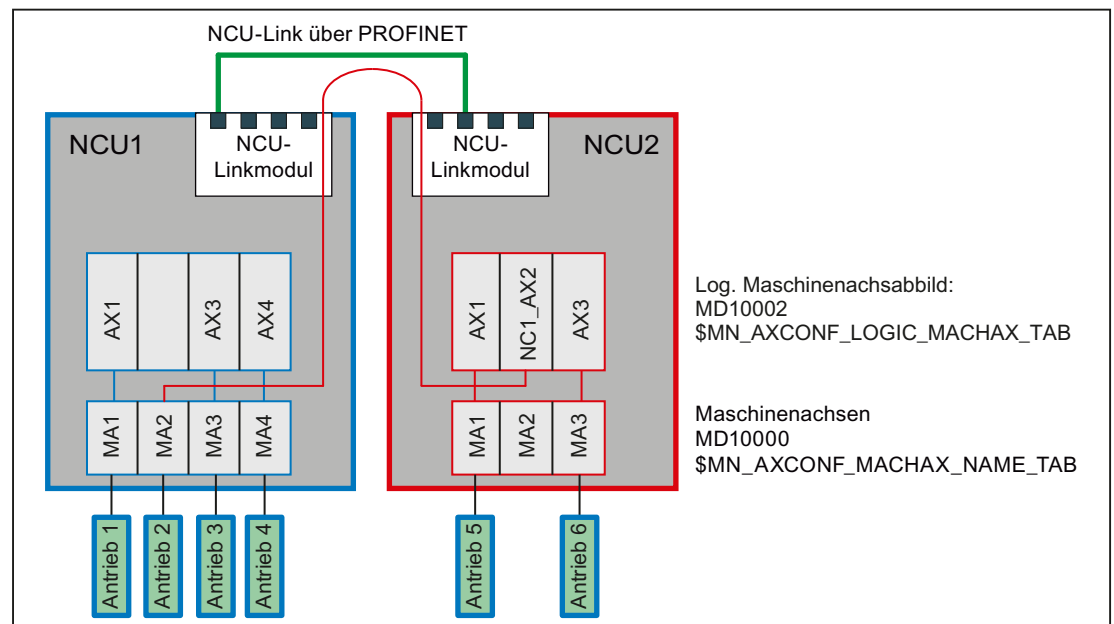
Der Zugriff auf die Link-Variable ist entsprechend der definierten Speicherstruktur folgendermaßen zu programmieren:

Programmcode	Beschreibung
\$A_DLB[0]	; BYTE (1)
\$A_DLB[1]	; BYTE (2)
\$A_DLW[2]	; WORD
\$A_DLD[4]	; DWORD (1)
\$A_DLD[8]	; DWORD (2)
\$A_DLD[12]	; DWORD (3)
\$A_DLR[16]	; REAL

14.2.2.8 Beispiel: Lesen von Antriebsdaten

Aufgabe

In einer Anlagen sind zwei NCU (NCU1 / NCU2) vorhanden. Die NCU sind über NCU-Link verbunden. Maschinenachse MA2 der NCU1 (Antrieb 2) wird als Link-Achse von NCU2 aus interpolatorisch verfahren. Der Stromistwert von Antrieb 2 soll von NCU1 zur Auswertung an NCU2 übertragen werden. Das folgende Bild zeigt den prinzipiellen Systemaufbau.



Voraussetzung

Der Stromistwert von Antrieb 2 (NCU1/MA2) kann über die Systemvariable \$VA_CURR gelesen werden. Bei PROFIdrive-basierten Antrieben muss dazu folgendes Maschinendatum gesetzt werden:

MD36730 \$MA_DRIVE_SIGNAL_TRACKING = 1 (Erfassung zusätzlicher Antriebsistwerte)

Durch Setzen des Maschinendatums werden folgende Antriebsistwerte bereitgestellt:

- \$AA_LOAD, \$VA_LOAD (Antriebsauslastung in %)
- \$AA_POWER, \$VA_POWER (Antriebswirkleistung in W)
- \$AA_TORQUE, \$VA_TORQUE (Antriebsmomentensollwert in Nm)
- \$AA_CURR, \$VA_CURR (Stromistwert der Achse oder Spindel in A)

Programmierung

NCU1

Über eine statische Synchronaktion wird zyklisch im Interpolator-Takt der Stromistwert \$VA_CURR des Antrieb 2 (NCU1/MA2) über die Link-Variable \$A_DLR[0] (REAL-Wert) in die ersten 8 Bytes des Link-Variablen-Speichers geschrieben:

Programmcode

```
IDS=1 WHENEVER TRUE DO $A_DLR[0]=$VA_CURR[MA2]
```

NCU2

Über eine statische Synchronaktion wird zyklisch im Interpolator-Takt der übertragene Stromistwert über die Link-Variable \$A_DLR[0] gelesen. Ist der Stromistwert größer als 23 A, wird der Alarm 61000 angezeigt.

Programmcode

```
IDS=1 WHEN $A_DLR[0] > 23.0 DO SETAL(61000)
```

14.2.3 Link-Achsen

14.2.3.1 Allgemeine Informationen

Als Link-Achse wird eine Maschinenachse bezeichnet, bei der die Sollwerte von einer anderen NCU erzeugt werden, als der, an welcher die Maschinenachse physikalisch angeschlossen ist. Damit ermöglichen Link-Achsen im Zusammenhang mit Achscontainern (siehe Kapitel "Achscontainer (Seite 794)") bei komplexen Anlagen, wie z.B. Rundtaktmaschinen mit mehreren NCU, die wechselweise Verwendung von Maschinenachsen der NCU des Link-Verbundes.

Wie im nachfolgenden Bild beispielhaft dargestellt, ist die Maschinenachse MA1 an der NCU1 angeschlossen. Die Maschinenachse MA2 ist an der NCU2 angeschlossen. Über ein Teileprogramm das in einem Kanal von NCU1 abgearbeitet wird, werden die Kanalachsen X und Z interpolatorisch verfahren. Im Interpolator der NCU1 werden die Sollwerte erzeugt. Für Maschinenachse MA1 werden sie an die Lageregelung der NCU1 weiter gegeben. Für Maschinenachse MA2 werden sie über NCU-Link an die Lageregelung der NCU2 übertragen und von dort an den Antrieb ausgegeben.

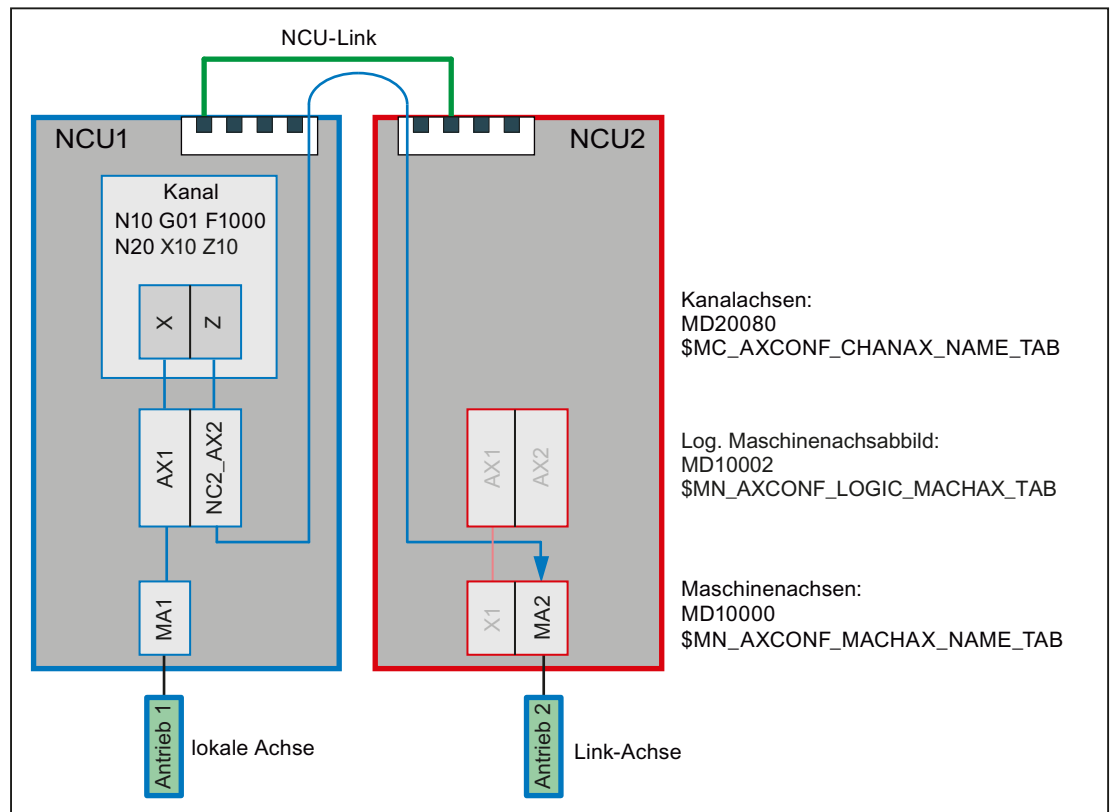


Bild 14-9 Linkachsen

Voraussetzung

Voraussetzung für die Verwendung von Link-Achsen ist eine gemäß Kapitel "Link-Kommunikation (Seite 773)" eingerichtete Link-Kommunikation.

Heimat-NCU

Die Heimat-NCU einer Link-Achse ist die NCU an welcher sie als Maschinenachse physikalisch angeschlossen ist. Die Lagereglung und der Austausch der axialen NC/PLC-Nahtstellensignale einer Link-Achse erfolgt immer auf der Heimat-NCU. Die Erzeugung der Sollwert kann prinzipiell auf jeder NCU des Link-Verbundes erfolgen.

Im obigen Bild:

- NCU1: Heimat-NCU von Maschinenachse MA1
- NCU2: Heimat-NCU von Maschinenachse MA2

14.2.3.2 Name einer Link-Achse

Der Name einer Link-Achse setzt sich zusammen aus der Kennung für die Heimat-NCU an der die Maschinenachse physikalisch angeschlossen ist, und dem allgemeinen Maschinenachsennamen AXn:
 NC<ID>_<Achse>

- <ID>: Kennung der NCU des Link-Verbundes entsprechend:
 MD12510 \$MN_NCU_LINKNO
 Siehe Kapitel "Parametrierung: Link-Kommunikation (Seite 780)"
- <Achse>: Allgemeiner Maschinenachsname: AX1, AX2, AX3, ...

14.2.3.3 Parametrierung

Zuordnung: Geometrie- oder Zusatzachse zu Link-Achse

Direkte Zuordnung

Die Zuordnung einer Geometrie- oder Zusatzachse zu einer Link-Achse kann im logischen Maschinenachsabbild direkt durch Angabe des Namens der Link-Achsen erfolgen:

MD10002 \$MA_AXCONF_LOGIC_MACHAX_TAB[<Achse>] = <Link-Achse>

Parameter	Bedeutung
<Achse>:	Maschinenachse-Index: 0, 1, 2, ... (max. Anzahl Maschinenachsen – 1)
<Link-Achse>:	Name der Link-Achse: NCx_AXy, siehe Kapitel "Name einer Link-Achse (Seite 790)"

Indirekt Zuordnung

Die Zuordnung einer Geometrie- oder Zusatzachse zu einer Link-Achse kann im logischen Maschinenachsabbild indirekt durch die Angabe eines Container-Slots erfolgen. Der Container-Slot enthält dann den eigentlichen, oben beschriebenen, Namen der Link-Achse. In diesem Fall wird von einer Container-Link-Achse gesprochen. Siehe dazu auch Kapitel: "Achscontainer (Seite 794)".

Synchrone Sollwertausgabe

Bei der Übertragung der Sollwerte einer Link-Achse von der Sollwert-erzeugenden NCU zur Heimat-NCU entsteht eine Verzögerung von einem Interpolator-Takt. Damit die Sollwerte bei der Interpolation von lokalen Achsen und Link-Achsen exakt zeitgleich an die Antriebe ausgegeben werden, muss diese Verzögerung kompensiert werden. Dazu ist auf der Sollwert-erzeugenden NCU, die Anzahl der Pufferelemente des Zwischenspeichers zwischen Interpolator und Lageregler um ein Element höher zu setzen, als die Anzahl der Pufferelemente der Heimat-NCU:

MD18720 \$MN_MM_SERVO_FIFO_SIZE = 3

Beispiel

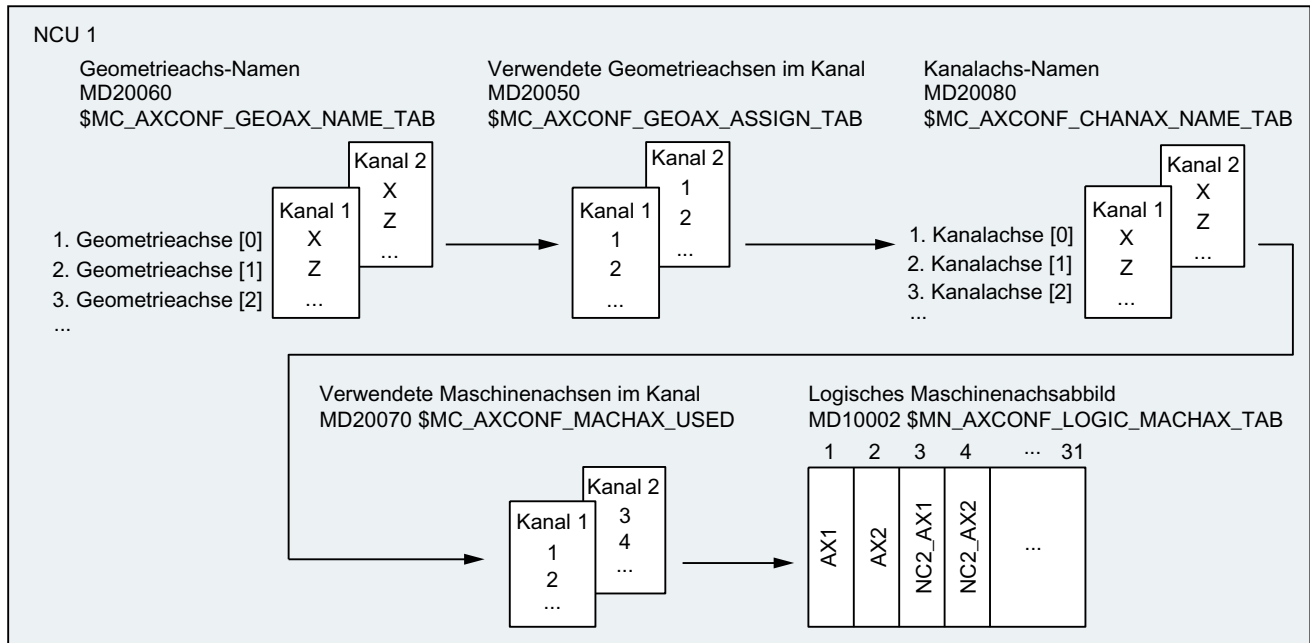


Bild 14-10 Beispiel: Parametrierung von Link-Achsen

Kanal 1

Den Geometrieachsen X / Z sind die lokalen Maschinenachsen AX1 / AX2 der NCU1 zugeordnet.

Kanal 2

Den Geometrieachsen X / Z sind die Link-Achsen NC2_AX1 / NC2_AX2 der NCU2 zugeordnet.

14.2.3.4 Hilfsfunktionsausgabe bei Spindeln

Beschreibung

Während der Programmbearbeitung und nach Satzsuchlauf mit "Suchlauf über Programmtest" (SERUPRO) werden die vordefinierten Hilfsfunktionen S, M3, M4, M5, M19 und M70 kanalspezifisch auf der NCU, auf der die Spindel programmiert wurde und zusätzlich achsspezifisch auf der Heimat-NCU ausgegeben.

Ausgabe der kanalspezifischen Hilfsfunktionen

- DB21, ... DBW68 (erweiterte Adresse der M-Funktion)
- DB21, ... DBD70 (M-Funktion 1)
- DB21, ... DBW98 (erweiterte Adresse der S-Funktion 1)
- DB21, ... DBD100 (S-Funktion 1)

Ausgabe der achsspezifischen Hilfsfunktionen

- DB31, ... DBD78 (F-Funktion für Achse)
- DB31, ... DBW86 (M-Funktion für Spindel)
- DB31, ... DBD88 (S-Funktion für Spindel)

Weitere Informationen

Ausführliche Informationen zur Hilfsfunktionsausgabe findet sich in:
Funktionshandbuch Basisfunktionen, Hilfsfunktionsausgaben an PLC

14.2.3.5 Randbedingungen

Maximale Anzahl von Maschinenachsen

Auch bei der Verwendung von Link-Achsen stehen nach wie vor pro NCU-Typ die maximale Anzahl von gleichzeitig nutzbaren Geometrie- und Zusatzachsen sowie Maschinenachsen zur Verfügung.

Funktionen "Lead-Link-Achsen" und "Link-Achsen"

Da die Funktionen "Lead-Link-Achsen" und "Link-Achsen" unterschiedliche Einstellungen im Maschinendatum MD18720 \$MN_MM_SERVO_FIFO_SIZE benötigen, können sie innerhalb eines Link-Verbundes nicht gleichzeitig verwendet werden.

Alarmer: Allgemeines Verhalten

Wird auf der Lagereglerebene der Heimat-NCU einer Link-Achse ein Fehler festgestellt und der entsprechende Alarm hat als Reaktion **nicht** "NC nicht betriebsbereit" zur Folge, wird der Alarm über NCU-Link an die Sollwert-erzeugende NCU übertragen und dort ausgegeben.

Alarmer: Verhalten bei Not-Halt

Wird bei einer NCU über die NC/PLC-Nahtstelle eine Not-Halt-Anforderung aktiviert, werden alle Achsen, die physikalisch an dieser NCU angeschlossen sind, in den Zustand "Nachführen" geschaltet. Davon sind auch Link-Achsen betroffen, deren Sollwerte aktuell von anderen NCUs erzeugt werden. Unter der Annahme, dass auf diesen NCUs ab diesem Zeitpunkt keine weiteren sinnvollen Bearbeitungen mehr möglich sind, wird hier ein zusätzlicher Alarm erzeugt, der alle abhängigen Achsbewegungen stoppt.

Alarmquittierung

Der zusätzlich erzeugte Alarm muss mit NC-Reset quittiert werden. Steht zu diesem Zeitpunkt noch der verursachende Alarm an, kann zwar der zusätzlich erzeugte Alarm quittiert werden, aber es wird ein weiterer selbstlöschender Alarm erzeugt, der ein Verfahren der Achsen bzw. einen Programmstart so lange verhindert, bis der verursachende Alarm quittiert wurde.

Alarmer: Verhalten bei Alarmreaktion "NC nicht betriebsbereit"

Wird auf der Lagereglerebene der Heimat-NCU einer Link-Achse ein Fehler festgestellt und der entsprechende Alarm hat als Reaktion "NC nicht betriebsbereit" zur Folge, wird der Alarm über NCU-Link an die Sollwert-erzeugende NCU übertragen und dort ausgegeben. Zusätzlich erfolgt die Alarmausgabe auf der Heimat-NCU.

Unter der Annahme, dass auf der Sollwert-erzeugenden NCU ab diesem Zeitpunkt keine weitere sinnvolle Bearbeitung mehr möglich ist, wird hier ein zusätzlicher Alarm erzeugt, der alle abhängigen Achsbewegungen stoppt.

Alarmquittierung

Siehe Alarmquittierung unter "Alarmer: Verhalten bei Not-Halt".

Alarmer: Verhalten bei Alarmreaktion "BAG nicht betriebsbereit"

Wird innerhalb einer BAG mit mehreren Kanälen ein Fehler festgestellt und der entsprechende Alarm hat als Reaktion "BAG nicht betriebsbereit", werden dadurch die Verfahrbewegungen in allen Kanälen der BAG stillgesetzt. Sind die Verfahrbewegungen prinzipiell voneinander unabhängig, kann die Reaktion über folgendes Maschinendatum auf "Kanal nicht betriebsbereit" umprojektiert werden:

```
MD11412 $MN_ALARM_REACTION_CHAN_NOREADY = TRUE
```

Auswirkung

In der NC/PLC-Nahtstelle wird statt des Signals DB11 DBX26.3 (BAG betriebsbereit) das Signal DB21, ... DBX36.5 (Kanal betriebsbereit) zurückgesetzt.

Vorteil

Die Alarmreaktion bleibt auf den Kanal, in dem der Fehler erkannt wird, begrenzt. Durch das PLC-Anwenderprogramm können bei Bedarf weitere Reaktionen ausgelöst werden.

Voraussetzung

Es tritt keine höherpriorige Alarmreaktion als "BAG nicht betriebsbereit" auf.

Kompensationen

Folgenden Kompensationen stehen **nicht** zur Verfügung:

- Link-Achsen: Quadrantenfehlerkompensation (QEC)
- Container-Link-Achsen: Durchhangkompensation (CEC)

Ausschalten einer NCU eines Link-Verbundes

Wird eine NCU eines Link-Verbundes ausgeschaltet, wird auf allen anderen NCUs des Link-Verbundes die Bearbeitung abgebrochen und ein Alarm angezeigt.

Hochlauf eines NCU-Verbundes

Wird auf einer NCU eines Link-Verbundes ein NC-Reset ausgelöst, wird dieser auch auf alle anderen NCUs des Link-Verbundes übertragen, so dass alle NCUs des Link-Verbundes einen Warmstart ausführen.

Technologien Nibbeln und Stanzen

Die für das Nibbeln und Stanzen benötigten schnellen Ein/Ausgänge müssen auf der NCU angeschlossen und parametrisiert werden, auf der das Teileprogramm abgearbeitet und die Achsen interpoliert werden. Die Befehle für "Schnelles Nibbeln und Stanzen", z. B. PONS, SONS, stehen für Link-Achsen nicht zur Verfügung.

Frames

In einem Frame-Befehl ist eine Link-Achse nur zulässig, wenn sie Geometrieachse ist. Der Frame-Befehl ändert nur die Geometrie in dem Kanal, dem die Link-Achse aktuell zugeordnet ist.

Drehzahl-/Drehmomentkopplung, Master-Slave

Die Antriebe aller Achsen/Spindeln eines Master-Slave-Verbundes müssen an derselben NCU angeschlossen sein. Die Master-Achse kann aber als Link-Achse vom Kanal einer anderen NCU aus verfahren werden.

14.2.4 Achscontainer

14.2.4.1 Allgemeine Informationen

Ein Achscontainer ist eine ringförmig Datenstruktur mit einer parametrisierbaren Anzahl von Elementen. Im Zusammenhang mit Achscontainern werden diese Elemente als Slots (Slot 1, Slot 2, ... Slot n) bezeichnet. Die Slots ermöglichen eine variable Zuordnung von Geometrie- und/oder Zusatzachsen zu Maschinenachsen. Der Eintrag in einem Slot kann auf eine NCU-lokale Maschinenachse (Container-Achse) oder eine Link-Achse (Container-Link-Achse) verweisen.

Im nachfolgenden Bild ist ein Achscontainer mit vier Slots dargestellt. Die Containerachsen verweisen in der aktuell Stellung des Achscontainers auf folgende Maschinenachsen:

Containerachse	Maschinenachse
CT1_SL1	NCU 1: AX1
CT1_SL2	NCU 1: AX2
CT1_SL3	NCU 2: AX1
CT1_SL4	NCU 2: AX2

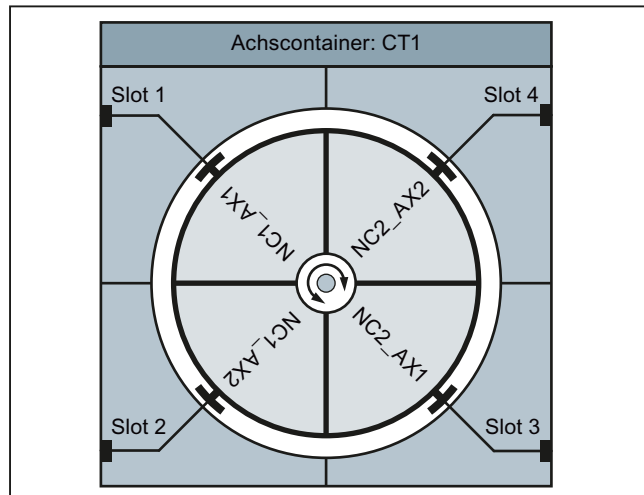


Bild 14-11 Beispiel: Achscontainer CT1 mit vier Slots

Regeln

Bei der Verwendung von Achscontainern sind folgende Regeln zu beachten:

- Alle Maschinenachsen eines Achscontainers dürfen zu einem Zeitpunkt immer nur genau einer Kanalachse zugeordnet sein.
- Es dürfen nicht mehrere Slots eines Achscontainers auf die gleiche Maschinenachse verweisen.
- Zu einem Zeitpunkt darf nur ein Kanal das Schreibrecht auf eine Maschinenachse, direkt oder über eine Containerachse, haben.
- Es können auch mehrere Geometrie- und/oder Zusatzachsen eines Kanals Containerachsen eines Achscontainers zugeordnet sein.

Zuordnung: Geometrie- oder Zusatzachse → Containerachse

Im logischen Maschinenachsabbild MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB kann einer Geometrie- oder Zusatzachse über Container und Slot eine Container-Achse als Maschinenachse zugeordnet werden z.B. Container "CT1", Slot "1":

```
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ n ] = CT1_SL1
```

Beim Verfahren der Geometrie- oder Zusatzachse, verfährt dann die Maschinenachse, die zu diesem Zeitpunkt dem Slot 1 zugeordnet ist.

Achscontainer-Drehung

Eine Achscontainer-Drehung wird immer dann ausgeführt, wenn alle am Achscontainer beteiligten Kanäle über Programmbefehl ihre Freigabe erteilt haben. Nach der Achscontainer-Drehung sind den Geometrie- oder Zusatzachsen der Kanäle andere Maschinenachsen zugeordnet.

Die Schrittweite einer Achscontainer-Drehung wird über Settingdatum vorgegeben.

Siehe Kapitel "Parametrierung (Seite 797)".

Achscontainername

Über den Achscontainername (<Achscontainer>) können folgende Programmbefehle adressiert werden:

- Programmbefehle:
 - AXCTSWE (<Achscontainer>)
 - AXCTSWED (<Achscontainer>)
 - AXCTSWEK (<Achscontainer>)

Als Namen sind möglich:

- CT<Containernummer>: An die Buchstabenkombination CT wird die Nummer des Achscontainers angehängt.
Beispiel: CT3
- <Containername>: Mit MD12750 \$MN_AXCT_NAME_TAB eingestellter individueller Name des Achscontainers.
Beispiel: A_CONT3
- <Achsname>: Achsname einer Containerachse, die im betreffenden Kanal bekannt ist.

Implizites Warten

Implizit wird auf den Vollzug einer angeforderten Achscontainer-Drehung gewartet, wenn eines der folgenden Ereignisse vorliegt:

- Teileprogramm-Sprachbefehle, die eine Achsbewegung einer zu diesem Achscontainer gehörenden Containerachse in diesem Kanal zur Folge haben
- GET (<Kanalachsname>) auf eine entsprechende Containerachse
- der nächste AXCTSWE (<Achscontainer>) für diesen Achscontainer

Hinweis

Auch ein IC (0) hat ein Warten mit ggf. notwendiger Synchronisation zur Folge (satzweise Änderung der Adressierung nach Kettenmaß, obwohl übergreifend Absolutmaß eingestellt ist).

Synchronisation auf Achsposition

Wenn nach der Drehung die neue dem Kanal zugeordnete Achscontainer-Achse nicht dieselbe absolute Maschinenposition hat wie die bisherige, so erfolgt eine Synchronisation auf die neue Position (internes REORG).

Hinweis

SD41700 \$SN_AXCT_SWWIDTH[<Achscontainer>] wird nur bei Neukonfiguration aktualisiert. Wenn nach schrittweisen Drehungen der RTM/MS die Stellung eine Schaltung vor der Ausgangslage erreicht ist, kann der Container normal weiter **vorwärts** gedreht werden, um die Ausgangslage des Containers wieder zu erreichen. Die Trommel oder der Rundtisch muss jedoch in die Ausgangslage **zurück**gedreht werden, damit Mess- und Versorgungsleitungen nicht abgedreht werden.

Siehe auch

Systemvariable (Seite 805)

Programmierung (Seite 803)

14.2.4.2 Parametrierung

Maschinendaten

NC-spezifische Maschinendaten

Nummer	Bezeichner \$MN_	Bedeutung
MD12750	AXCT_NAME_TAB	Name der Achscontainer
MD12760	AXCT_FUNCTION_MASK.Bit x	Achscontainer-spezifische Funktionen
MD1270x	AXCT_AXCONF_ASSIGN_TABx	Zuordnung von Maschinenachsen zu den Slots eines Achscontainers
MD18720	MM_SERVO_FIFO_SIZE	Größe des IPO/SERVO-Datenpuffers Hinweis Bei NCU-übergreifenden Achscontainern ist auf allen beteiligten NCU der Wert 3 einzustellen.

Name der Achscontainer

MD12750 \$MN_AXCT_NAME_TAB[<Index>] = "<Name>"

Parameter	Bedeutung
<Index>:	0, 1, ... max. Achscontainerindex
<Name>:	Name des Achscontainers (z.B. CT1)

Achscontainer-spezifische Funktionen

MD12760 \$MN_AXCT_FUNCTION_MASK.Bit x = <Wert>

Parameter	<Wert>	Bedeutung
Bit 0:	0	Bei einer direkten Achscontainerschaltung (AXCTSWED), müssen alle andere Kanäle im RESET-Zustand sein.
	1	Bei einer direkten Achscontainerschaltung (AXCTSWED) müssen nur andere Kanäle, die das Interpolationsrecht auf Achsen des Achscontainers haben, im RESET-Zustand sein.

Über das Maschinendatum werden die Achscontainer-spezifischen Funktionen aktiviert.

Zuordnung von Maschinenachsen zu den Slots eines Achscontainers

MD1270x \$MN_AXCT_AXCONF_ASSIGN_TABx[<Index>] = <Achse>

Parameter	Bedeutung
x:	1 ... max. Anzahl von Achscontainern
<Index>:	0, 1, ... max. Slotindex
<Achse>:	Name einer lokalen Maschinenachse (z.B. AX1)
	Name einer Link-Achse. Siehe Kapitel "Allgemeine Informationen (Seite 788)".

Innerhalb eines Achscontainers müssen die Slots lückenlos, beginnend mit Slotindex 0, in aufsteigender Reihenfolge belegt werden.

Settingdaten

Schrittweite einer Achscontainer-Drehung

SD41700 \$SN_AXCT_SWWIDTH[<Index>] = <Schrittweite>

Parameter	Bedeutung
<Index>:	0, 1, ... max. Achscontainerindex
<Schrittweite>:	Anzahl der Slots um die der Achscontainer gedreht wird

Veranschaulichung der Achscontainerdrehung

Die Achscontainer-Drehung wird über Programmbefehle freigegeben. Siehe Kapitel "Programmierung (Seite 803)".

Im linken Teil von Bild "Achscontainer-Drehung, **Bild 1**" ist in der **Grundstellung** des Achscontainers dem Slot 1 die Link-Achse NCU1_AX1 zugeordnet.

Nach der Drehung mit Schrittweite 1 (rechter Teil) ist dem Slot 1 die Link-Achse NCU2_AX2 zugeordnet.

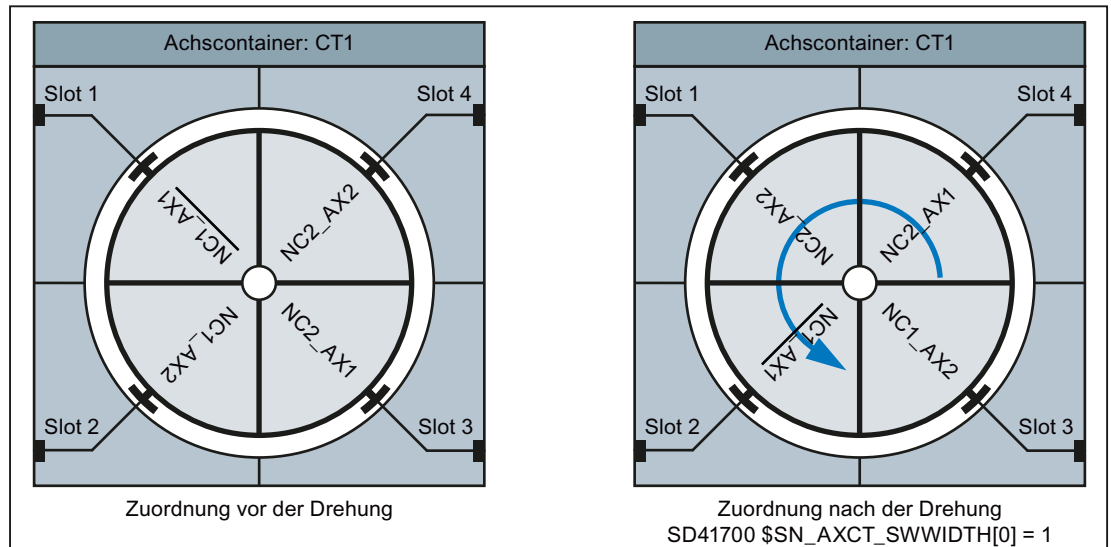


Bild 14-12 Achscontainer-Drehung, Bild 1

Ausgehend von der oben genannten **Grundstellung** ist nach einer Drehung mit Schrittweite 2, (Bild "Achscontainer-Drehung, **Bild 2**", linker Teil) dem Slot 1 die Link-Achse NCU2_ AX1 zugeordnet.

Ausgehend von der oben genannten **Grundstellung** ist nach einer Drehung mit Schrittweite -1 (Bild "Achscontainer-Drehung, **Bild 2**", rechter Teil) dem Slot 1 die Link-Achse NCU1_ AX2 zugeordnet.

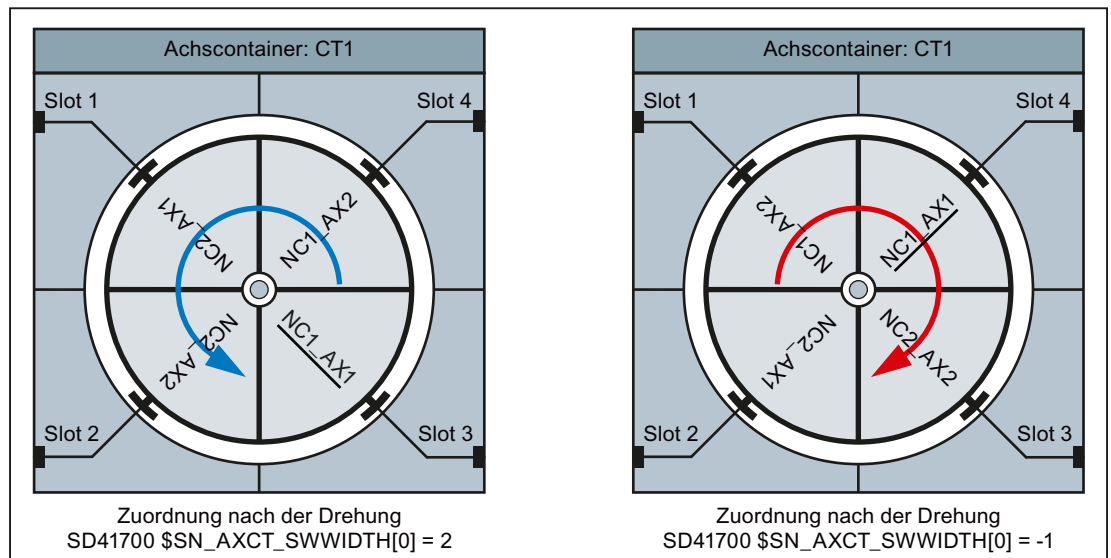


Bild 14-13 Achscontainer-Drehung, Bild 2

Achscontainer mit Container-Link-Achsen

Die Parametrierung eines Achscontainers, der Container-Link-Achsen enthält, muss auf der Master-NCU des Link-Verbundes ($MD12510 \$MN_NCU_LINKNO == 1$) vorgenommen werden.

Abgleich von axialen Maschinendaten

Bei Containerachsen müssen alle mit dem Attribut "CTEQ" (ConTainer EQual) gekennzeichneten axialen Maschinendaten für alle Containerachsen den gleichen Wert haben. Bei unterschiedlichen Werten, werden diese automatisch abgeglichen.

Steuerungshochlauf

Im Hochlauf der Steuerung werden alle Maschinendaten auf die Werte der Containerachse des ersten Slots abgeglichen. Wird dabei der Wert eines Maschinendatums geändert, wird folgende Meldung angezeigt: "Die axialen Maschinendaten der Achsen im Achscontainer <n> wurden angepasst"

Maschinendatenänderung

Bei Änderung eines Maschinendatums einer beliebigen Containerachse, wird der neu Wert sofort auch in alle anderen Containerachsen übertragen. Dabei wird folgende Meldung angezeigt: "Achtung dieses MD wird für alle Containerachsen gesetzt"

Slot-Änderung

Wird einem Slot eines Achscontainers eine andere Maschinenachse zugewiesen, (MD127xx AXCT_AXCONF_ASSIGN_TAB<x>), wird folgender Hinweis angezeigt: "Beim nächsten Hochlauf werden die Maschinendaten der Achsen im Achscontainer <n> angepasst".

Hinweis

Container-Link-Achsen

Bei Container-Link-Achsen erfolgt der Maschinendatenabgleich auf allen am Achscontainer beteiligten NCU des Link-Verbundes.

Parametrierbeispiel

Annahmen

NCU	Komponenten
NCU1:	Kanal 1, Geometrieachsen X / Z → 1. / 2. Kanalachse Kanal 2, Geometrieachsen X / Z → 1. / 2. Kanalachse Maschinenachsen: AX1, AX2 Achscontainer CT1 mit 4 Slots
NCU2:	Maschinenachsen: AX1, AX2

Parametrierung: NCU1

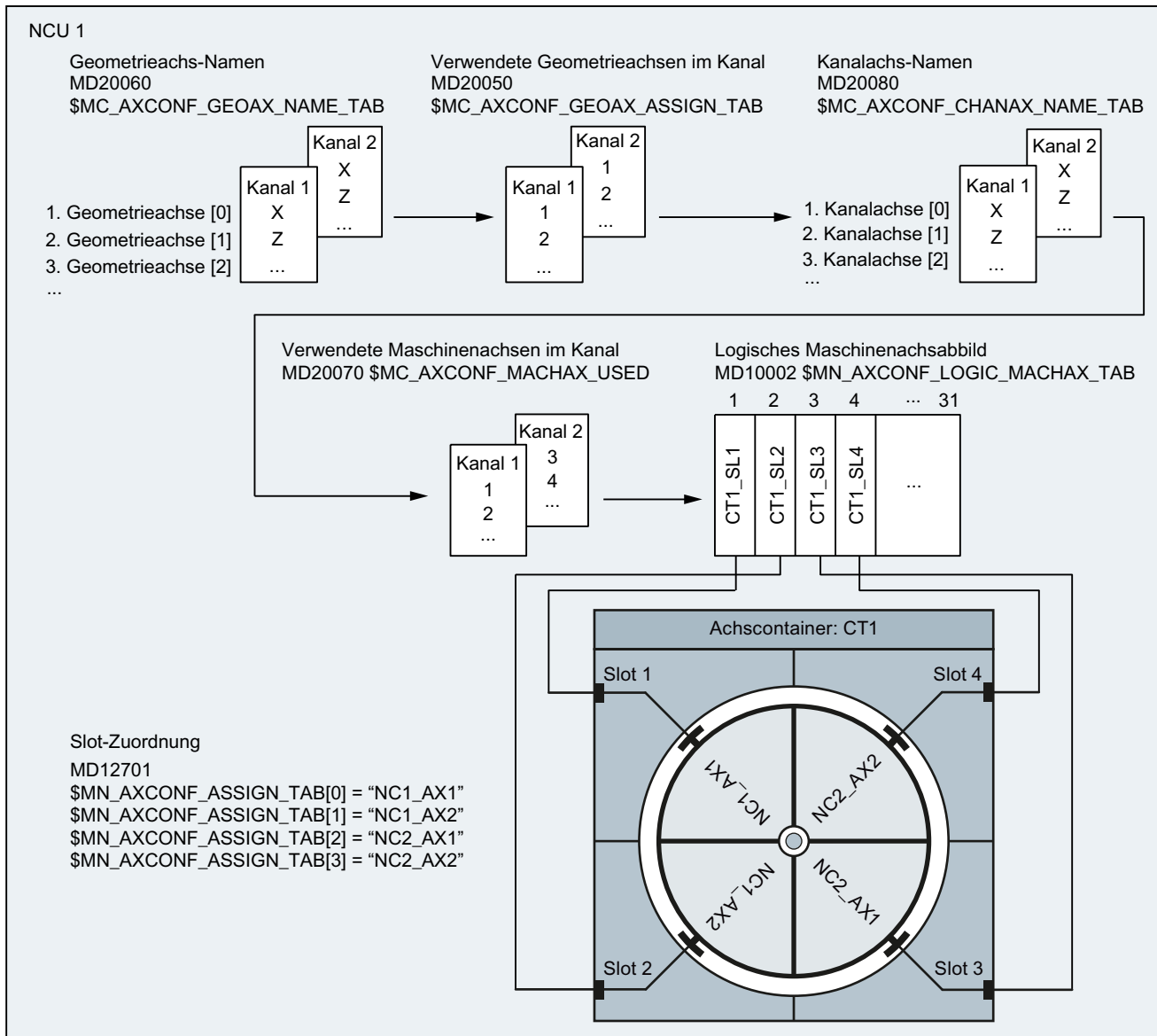


Bild 14-14 Beispiel: Parametrierung von Kanalachsen und Achscontainer

Wirkung

Durch Programmierung der Geometrieachsen X und Z im 1. und 2. Kanal der NCU1, verfahren in der aktuellen Stellung des Containers:

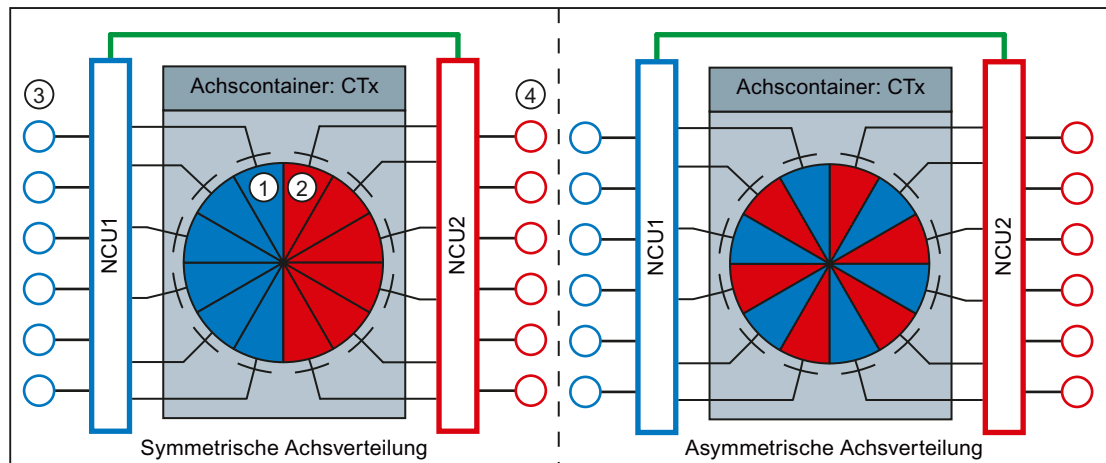
- Die lokalen Maschinenachsen AX1 und AX2 der NCU1.
- Als Container-Link-Achsen die Maschinenachsen AX1 und AX2 der NCU2.

Zur Achscontainer-Drehung siehe Kapitel "Programmierung (Seite 803)".

Hinweise zur Parametrierung

Containerachsverteilung und Kommunikationsauslastung

Bei einer Anlage mit mehreren NCU, die im Zusammenhang mit Achscontainern wechselweise Achsen anderer NCU verfahren (Link-Achsen), entscheidet die Art und Weise wie die Link-Achsen innerhalb des Achscontainers verteilt sind über die Auslastung der Link-Kommunikation.



- ① Blauer Slot: Verweist auf einen an der NCU1 angeschlossenen Antrieb
- ② Roter Slot: Verweist auf einen an der NCU2 angeschlossenen Antrieb
- ③ An NCU1 angeschlossene Antriebe
- ④ An NCU2 angeschlossene Antriebe

Bild 14-15 Achsverteilung

- **Symmetrische Achsverteilung**
Bei einer symmetrischen Achsverteilung verfährt zuerst jede NCU nur lokale Achsen. Es findet daher keine Link-Kommunikation statt. Mit jedem Weiterschalten des Achscontainers erhöht sich die Auslastung der Link-Kommunikation bis zu einem Maximum, wenn alle NCU ausschließlich Link-Achsen verfahren.
- **Asymmetrische Achsverteilung**
Bei einer asymmetrischen Achsverteilung verfährt jede NCU von Anfang an lokale und Link-Achsen. Im Vergleich zur symmetrischen Achsverteilung ergibt sich dadurch eine gleichbleibende mittlere Auslastung der Link-Kommunikation.

Antriebsverteilung und Kommunikationsauslastung

Bei einer Anlage mit mehreren NCU, die im Zusammenhang mit Achscontainern wechselweise Achsen anderer NCU verfahren (Link-Achsen), entscheidet die Verteilung der an die NCU angeschlossenen Antriebe über die Auslastung der Link-Kommunikation.

- **Symmetrische Antriebsverteilung**
Bei einer symmetrischen Antriebsverteilung sind die über Achscontainer angesprochenen Antriebe auf beide NCU verteilt angeschossen. Durch diese Anordnung können über die Logischen Maschinenachsen Abbilder (LAI) auf beiden NCU weiterhin die maximal mögliche Anzahl von Antrieben adressiert werden.

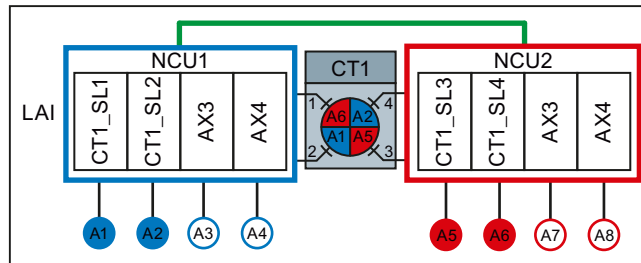


Bild 14-16 Symmetrische Antriebsverteilung

- **Asymmetrische Antriebsverteilung**
Bei einer asymmetrischen Antriebsverteilung sind die über Achscontainer angesprochenen Antriebe nur an NCU1 angeschossen. Durch diese Anordnung kann nur über das Logische Maschinenachsen Abbild (LAI) auf NCU2 weiterhin die maximal mögliche Anzahl von Antrieben adressiert werden. Über das LAI von NCU1 kann nur noch die maximale Anzahl minus der von NCU2 genutzten Antriebe adressiert werden. Um auch auf NCU1 die maximale Anzahl von Antrieben nutzen zu können, müssten diese an NCU2 angeschossen und von NCU1 aus über NCU-Link angesprochen werden. Was eine höhere zyklische Link-Kommunikationslast zur Folge hat.

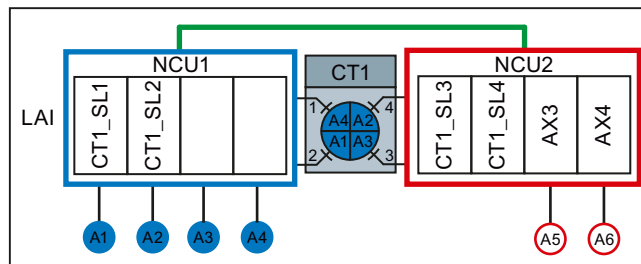


Bild 14-17 Asymmetrische Antriebsverteilung

14.2.4.3 Programmierung

Funktion

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.
Die Schrittweite einer Achscontainer-Drehung wird eingestellt über das Settingdatum SD41700 \$SN_AXCT_SWWIDTH (siehe Kapitel "Parametrierung (Seite 797)")

AXCTSWED: Freigabe zum Drehen des Achscontainers ohne Berücksichtigung der anderen am Achscontainer beteiligten Kanäle
Die Schrittweite einer Achscontainer-Drehung wird eingestellt über das Settingdatum SD41700 \$SN_AXCT_SWWIDTH (siehe Kapitel "Parametrierung (Seite 797)")

Hinweis

- 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
Siehe Kapitel " Parametrierung (Seite 797)".

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
Siehe Kapitel "Systemvariable (Seite 805)"

<ID>: Name des Achscontainers oder einer Containerachse:

CT<Nummer>: Defaultbezeichner eines Achscontainers:
MD12750 \$MN_AXCT_NAME_TAB
Beispiel: CT1

<Container>: Anwenderspezifischer Name eines Achscontainers:
MD12750 \$MN_AXCT_NAME_TAB
Beispiel: CONTAINER_1

<Achse>: Name einer im Kanal bekannten Containerachse

Weitere Informationen

Die Anwendung des Befehls AXCTSWEC in Synchronaktionen ist ausführlich beschrieben in:

Funktionshandbuch Synchronaktionen; Ausführliche Beschreibung > Aktionen in Synchronaktionen > Freigabe für Achscontainer-Drehung zurücknehmen (AXCTSWE)

14.2.4.4 Systemvariable

Beschreibung

Container-spezifische Systemvariable

Systemvariable	Beschreibung
\$AC_AXCTSWA[<ID>]	Kanal-spezifischer Status de Achscontainer-Drehung
\$AN_AXCTSWA[<ID>]	NCU-spezifischer Status des Achscontainer-Drehung
\$AN_AXCTSWE[<ID>]	Slot-spezifischer Status der Achscontainer-Drehung
\$AN_AXCTAS[<ID>]	Anzahl der Slots um die der Achscontainer aktuell weitergeschaltet wurde
ID: Achscontainername oder Name einer Containerachse	

NC-spezifische Systemvariable

Systemvariable	Beschreibung
\$AN_LAI_AX_IS_AXCTAX ¹⁾	Status: LAI-Achse == Containerachse der Maschinenachsen im logischen Maschinenachsabbild (MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB) bezüglich Achscontainer.
\$AN_LAI_AX_IS_LEADLINKAX ¹⁾	Status: LAI-Achse == Lead-Link-Achse
\$AN_LAI_AX_IS_LINKAX ¹⁾	Status: LAI-Achse == Link-Achse
\$AN_LAI_AX_TO_IPO_NC_CHANAX[<ID>]	Kanal- und Kanalachsnummer bzw. NCU-ID und globale Achsnummer
\$AN_LAI_AX_TO_MACHAX[<ID>]	NCU-ID und Achsnummer der Maschinenachse
¹⁾ Bitmaske: Bit $n \pm$ LAI-Achse ($n+1$) aus MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB ID: LAI-Achsnummer NCU-ID: Wert aus MD12510 \$MN_NCU_LINKNO	

Weitere Informationen

Eine ausführliche Beschreibung der Systemvariable findet sich in:
 Listenhandbuch Systemvariable

Siehe auch

Achscontainer Systemvariablen auswerten (Seite 818)

14.2.4.5 Bearbeitung mit Achscontainer (schematisch)

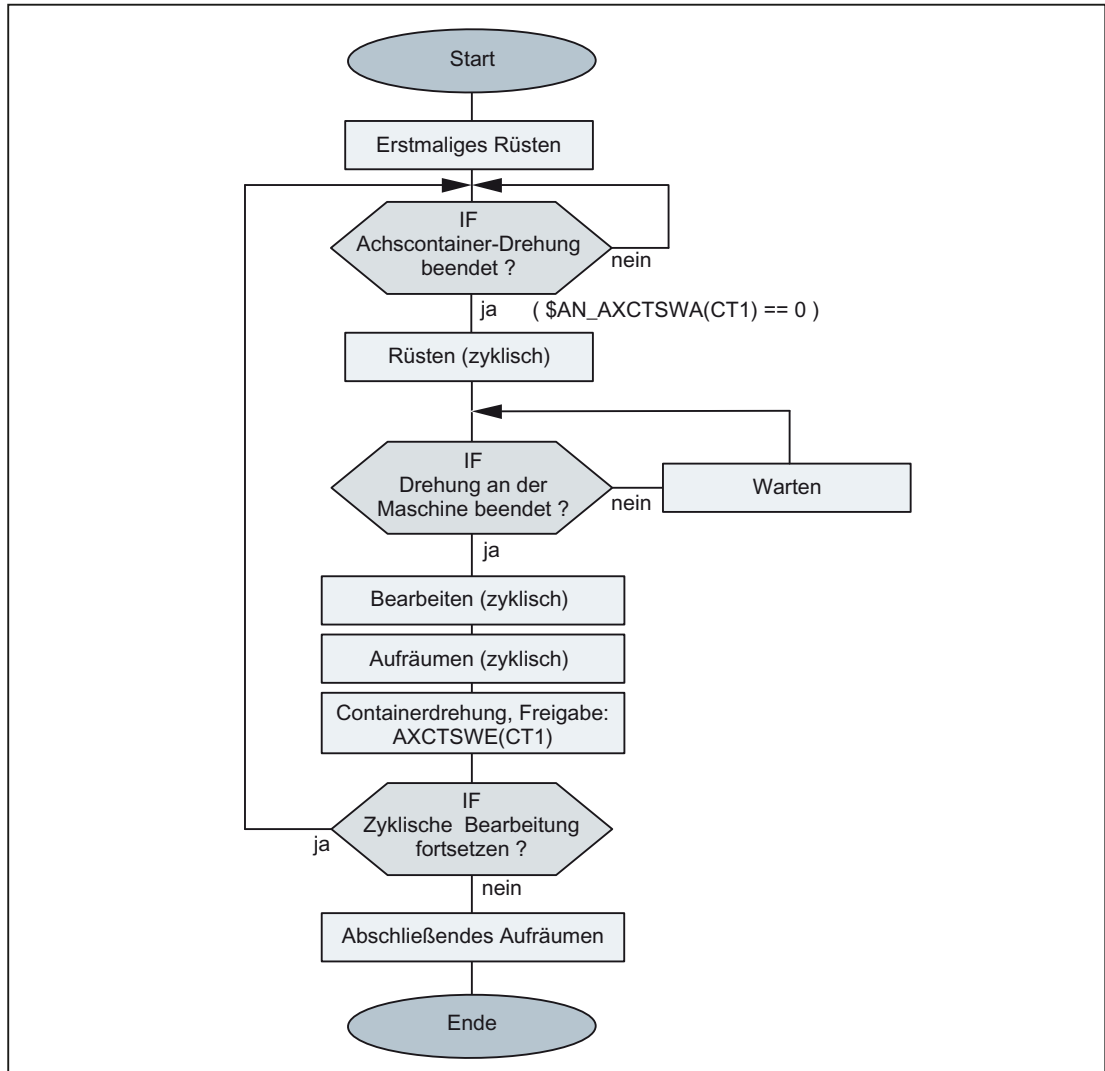


Bild 14-18 Beispiel: schematische Bearbeitungsfolge für eine Station einer Rundtaktmaschine

14.2.4.6 Verhalten in verschiedenen Betriebszuständen

Hochlauf (Power On)

Im Hochlauf der Steuerung wird bezüglich der Slot-Zuordnung immer der in den Maschinendaten festgelegte Grundzustand eingenommen, unabhängig davon, in welchem Zustand des Achscontainers die Steuerung ausgeschaltet wurde:

MD1270x \$MN_AXCT_AXCONF_ASSIGN_TABx

Hinweis

Ausgleich zwischen Soll- und Istzustand

Es liegt in der alleinigen Verantwortung des Anwenders / Maschinenherstellers nach dem Hochlauf der Steuerung eine Differenz zwischen dem Zustand des Achscontainers und dem Maschinenzustand zu erkennen und diesen durch eine geeignete Achscontainer-Drehung auszugleichen.

Betriebsartenwechsel

Eine Containerachse, deren Achscontainer in der Betriebsart AUTOMATK für die Drehung freigegeben wurde, kann nach einem Wechsel in der Betriebsart JOG nicht verfahren werden.

Kanal-spezifischer Reset-Zustand

Sobald ein Kanal am Achscontainer beteiligter Kanal im Reset-Zustand ist, wird von diesem Kanal keine Freigabe zur Achscontainer-Drehung benötigt. Es genügen die Freigaben der verbleibenden aktiven Kanäle.

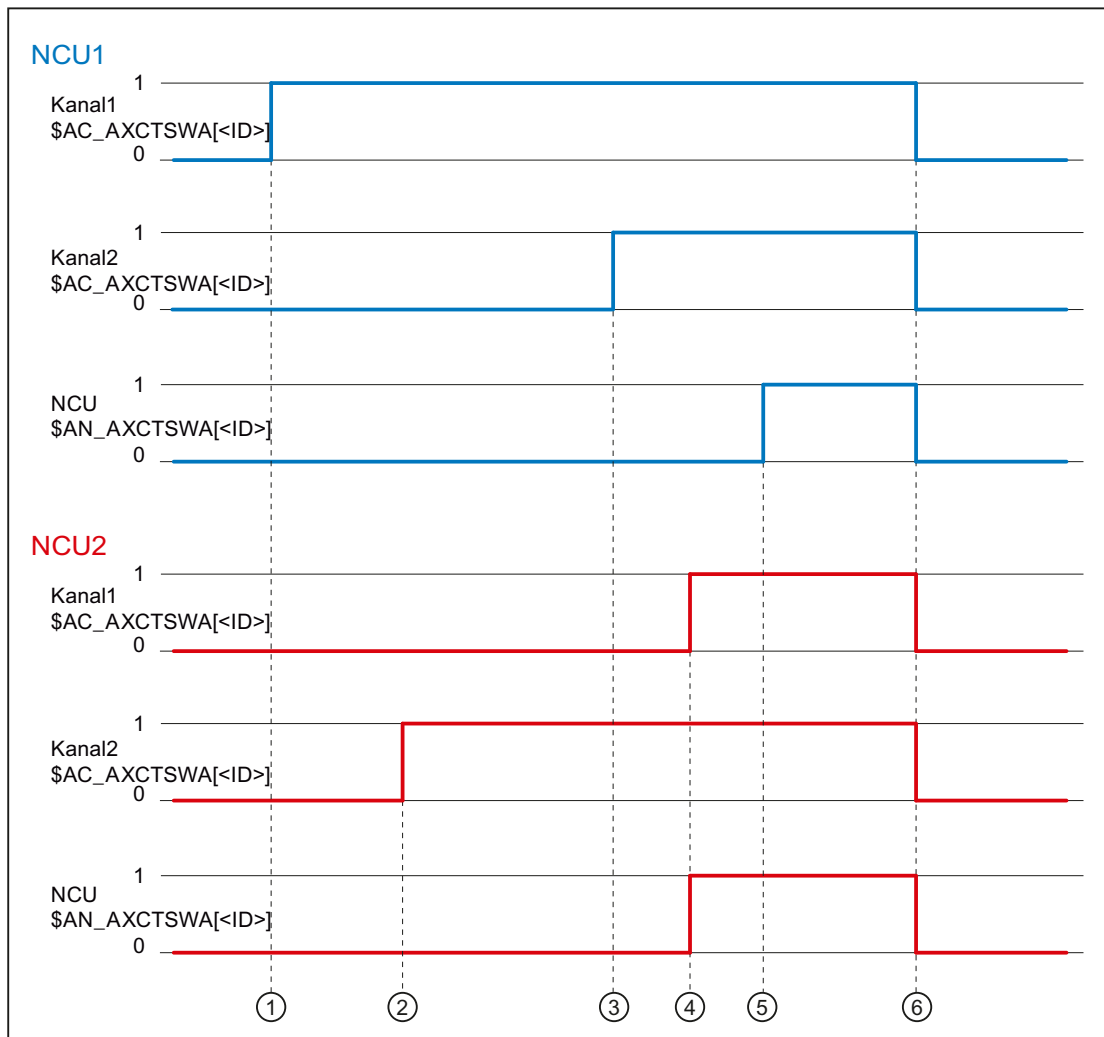
Satzsuchlauf

Die Freigabe und Aktivierung der Achscontainer-Drehung (`AXCTSWE`) kann nicht zusammengefasst werden, sondern muss jeweils in einem Aktionssatz ausgegeben werden. D. h. der Achscontainer-Zustand ändert sich abhängig vom Zustand anderer Kanäle für jeden Drehbefehl gesondert.

14.2.4.7 Verhalten bei Rücknahme der Freigabe zur Achscontainer-Drehung

Mit dem Befehl `AXCTSWE` wird für einen Achscontainer Kanal-spezifisch die Freigabe zur Achscontainer-Drehung gegeben. Über den Befehl `AXCTSWEC` kann die Freigabe wieder zurückgenommen werden.

Das folgende Bild zeigt beispielhaft den Ablauf einer Achscontainer-Drehung wie er sich in den Achscontainer-spezifischen Systemvariablen darstellt. Am Achscontainer sind jeweils zwei Kanäle von zwei NCU beteiligt.



- ① NCU1, Kanal1: Freigabe über Befehl AXCTSWE erteilt
- ② NCU2, Kanal2: Freigabe über Befehl AXCTSWE erteilt
- ③ NCU1, Kanal2: Freigabe über Befehl AXCTSWE erteilt → in NCU1 liegen jetzt alle Freigaben aller Kanäle vor → der Gesamt-Freigabestatus von NCU1 wird über Link-Kommunikation an NCU2 übertragen
- ④ NCU2, Kanal1: Freigabe über Befehl AXCTSWE erteilt → in NCU2 liegen jetzt alle Freigaben aller Kanäle vor → der Gesamt-Freigabestatus von NCU2 wird über Link-Kommunikation an NCU1 übertragen
in NCU2 liegen jetzt alle Freigaben von allen Kanälen (NCU2 und NCU1) vor → die Achscontainer-Drehung wird in NCU2 ausgeführt
- ⑤ NCU1: in NCU1 liegen jetzt alle Freigaben von allen Kanälen (NCU1 und NCU2) vor → die Achscontainer-Drehung wird in NCU1 ausgeführt
- ⑥ NCU1 / NCU2: die Achscontainer-Drehung ist beendet

Bild 14-19 NCU-übergreifende Freigabe und Achscontainer-Drehung

Damit in einem Kanal eine einmal erteilte Freigabe wieder zurückgenommen werden kann, muss zum Zeitpunkt der Rücknahme die Freigabe von mindestens einem der am Achscontainer beteiligten Kanäle (NCU1 oder NCU2) noch ausstehen. Eine Rücknahme muss daher vor Zeitpunkt ④ erfolgen.

Sobald alle Freigaben von allen Kanälen aller NCUs vorliegen (Zeitpunkt ④), ist eine Rücknahme nicht mehr möglich. In diesem Fall bleibt der Befehl `AXCTSWEC` wirkungslos. Es erfolgt keine Rückmeldung an den Anwender.

Siehe auch

Programmierung (Seite 803)

14.2.4.8 Randbedingungen

Achsbetrieb

Verfährt eine Containerachse im Achsbetrieb oder als positionierende Spindel (POSA, SPOSA), wird eine Drehung des Achscontainers erst nach Erreichen der programmierten Endposition ausgeführt.

Spindel

- Eine Containerachse, die als Spindel aktiv ist, dreht während einer Achscontainer-Drehung weiter.
- Die Regelungsart einer Spindel (Drehzahl- / Lageregelung) bezieht sich auf die zugehörige Maschinenachse. Bei einer Achscontainer-Drehung "wandert" die eingestellte Regelungsart mit der Maschinenachse mit.
- Bei Befehlen, die sich auf die Masterspindel des Kanals beziehen, muss zum Ausführungszeitpunkt des Befehls eine Maschinenachse mit der entsprechenden Spindelnummer im Kanal vorhanden sein:
`MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[<Achse>] == Nummer der Masterspindel`

Hinweis

Es liegt in der alleinigen Verantwortung des Anwenders/Maschinenhersteller dafür zu sorgen, dass bei Spindeln als Containerachsen nach einer Achscontainer-Drehung für die Masterspindel weiterhin eine entsprechende Maschinenachse im Kanal vorhanden ist.

Nullpunktverschiebungen

Hinweis

Es liegt in der alleinigen Verantwortung des Anwenders / Maschinenhersteller dafür zu sorgen, dass nach einer Achscontainer-Drehung die im Kanal wirksamen Nullpunktverschiebungen an die geänderten Maschinenachsen-Zuordnungen angepasst werden.

Bahnsteuerbetrieb

Ist im Kanal Bahnsteuerbetrieb aktiv und es erfolgt eine Achscontainer-Drehung, unterbricht eine nachfolgende Programmierung einer Containerachse den Bahnsteuerbetrieb. Die Unterbrechung erfolgt auch dann, wenn die Containerachse keine Bahnachse ist.

PLC-Achse

Soll eine Containerachse, deren Achscontainer zur Drehung freigegeben wurde, zur PLC-Achse werden, erfolgt der Statuswechsel erst nach Abschluss der Achscontainer-Drehung.

Kommandoachse

Soll eine Containerachse, deren Achscontainer zur Drehung freigegeben wurde, als Kommandoachse verfahren werden, wird die Verfahrbewegung erst nach Abschluss der Achscontainer-Drehung ausgeführt.

Pendelachse

Soll eine Containerachse, deren Achscontainer zur Drehung freigegeben wurde, zur Pendelachse werden, erfolgt der Statuswechsel erst nach Abschluss der Achscontainer-Drehung.

Externe Nullpunktverschiebung

Die "externe Nullpunktverschiebung" bezieht sich auf das Maschinenkoordinatensystem (MKS). Daher wird bei einer aktiven "externen Nullpunktverschiebung" einer der Containerachsen die Achscontainer-Drehung mit Alarm 4022 abgelehnt.

Axiale Frames

Der axiale Frame einer Kanalachse, die Containerachse ist, bleibt über die Achscontainer-Drehung hinweg nicht erhalten. Da durch die Achscontainer-Drehung der Kanalachse eine neue Maschinenachse zugeordnet wird, der axiale Frame aber den Bezug zur Maschinenachse hat, ändert sich durch die Drehung auch der axiale Frame. Falls die beiden Frames nicht übereinstimmen, so erfolgt eine Synchronisation (internes REORG).

Hinweis

Die Zuordnung einer Kanalachse zu einer Maschinenachse ändert sich durch die Achscontainer-Drehung. Die aktuellen Frames bleiben nach der Drehung erhalten. Der Anwender muss selbst dafür sorgen, dass die richtigen Frames nach der Drehung angewählt werden. Dies kann z. B. durch die Programmierung der Basisframe-Masken erreicht werden.

Transformation

Ist eine Containerachse als Spindel an einer Transformation beteiligt, muss vor Freigabe der Achscontainer-Drehung die Transformation abgewählt werden.

Achskopplungen

Ist für eine Containerachse eine Achskopplung aktiv, muss vor Freigabe der Achscontainer-Drehung die Kopplung mit `COUPOF` abgewählt werden. Nach Abschluss der Drehung kann die Kopplung sofort mit `COUPON` wieder angewählt werden. Ein erneutes Definieren der Kopplung ist nicht erforderlich.

Gantry-Achse

Eine Gantry-Achse darf keine Containerachse sein.

Fahren auf Festanschlag

Steht eine Containerachse am Festanschlag, kann keine Achscontainer-Drehung ausgeführt werden.

Antriebsalarme

Steht für eine Containerachse ein Antriebsalarm an, wird die Achscontainer-Drehung nicht durchgeführt.

14.2.5 Lead-Link-Achsen

14.2.5.1 Allgemeine Informationen

Sind bei einer Achskopplung die Maschinenachsen der Leit- und Folgeachse nicht an derselben NCU angeschlossen, muss die Kopplung über eine Link-Achse der NCU der Folgeachse erfolgen. Die Link-Achse wird in diesem Fall als Lead-Link-Achse bezeichnet.

Die Sollwerte der Leitachse werden synchron im Interpolator-Takt über NCU-Link an die Lead-Link-Achse übertragen. Entsprechend erfolgt in Gegenrichtung die Übertragung der Istwerte und Zustandsdaten der Lead-Link- und Folgeachse an die Leitachse.

Die Lead-Link-Achse wird als lokale Leitachse der Folgeachse parametrierbar.

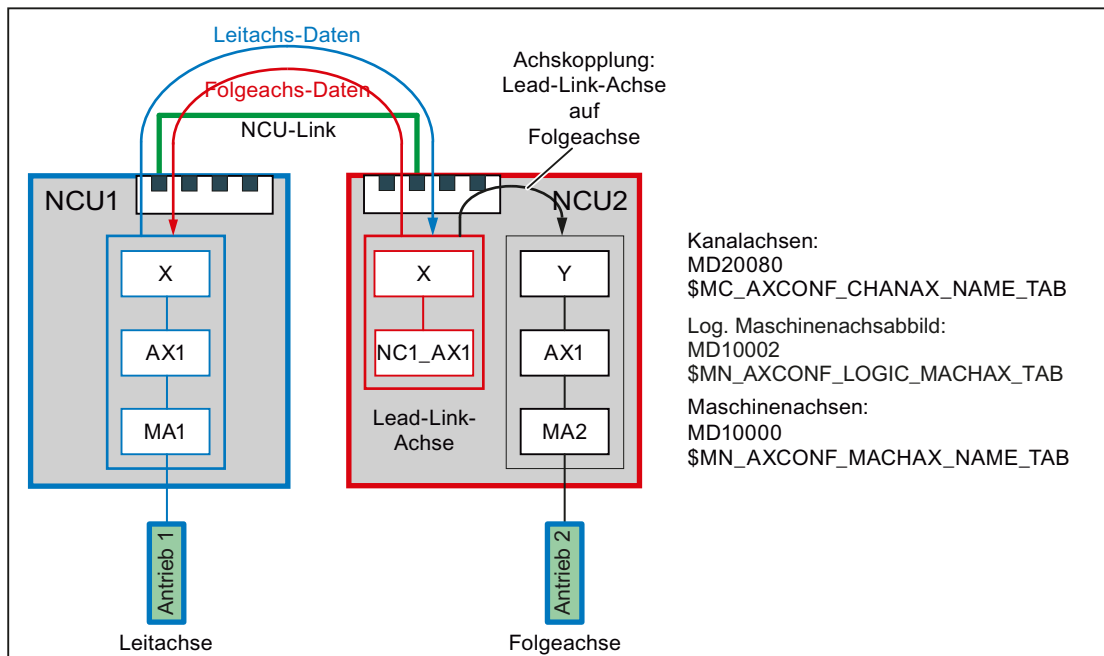


Bild 14-20 Lead-Link-Achse

Achskopplungen

Lead-Link-Achsen können im Zusammenhang mit folgenden Achskopplungen verwendet werden:

- Leitwertkopplung
- Mitschleppen
- Tangentiales Nachführen
- Elektronisches Getriebe (ELG)
- Synchronspindel

Voraussetzung

Die NCU müssen über NCU-Link kommunizieren. Siehe Kapitel "Link-Kommunikation (Seite 773)"

14.2.5.2 Parametrierung

Link-Kommunikation

NC-spezifische Maschinendaten

Nummer	Bezeichner \$MN_	Bedeutung
MD12510	NCU_LINKNO	Eindeutige numerische Kennung der NCU innerhalb des Link-Verbundes. Die Kennungen müssen von 1 beginnend lückenlos in aufsteigender Reihenfolge vergeben werden. Wertebereich: 1, 2, ... maximale NCU-Nummer Hinweis Die NCU, der als NCU-Kennung der Wert 1 zugewiesen wird, ist die Master-NCU des Link-Verbundes. Die Parametrierung von Link-Achsen und Achscontainern sind ausschließlich über die Maschinen- und Settingdaten) der Master-NCU vorzunehmen.
MD18780	MM_NCU_LINK_MASK.Bit 0	Aktivierung der Link-Kommunikation
MD18781	NCU_LINK_CONNECTIONS	Anzahl interner Link-Verbindungen Hinweis Es wird empfohlen, den Standardwert 0 (Ermittlung der Anzahl durch die NC) beizubehalten.
MD18782	MM_LINK_NUM_OF_MODULES	Anzahl der über NCU-Link miteinander verbundenen NCUs.

Sollwert-Synchronisation

NC-spezifische Maschinendaten

Nummer	Bezeichner \$MN_	Bedeutung
MD18720	MM_SERVO_FIFO_SIZE	Größe des IPO/SERVO-Datenpuffers Durch die Übertragung der Sollwerte der Leitachse über den NCU-Link auf die NCU der Lead-Link-Achse ergibt sich eine Totzeit von zwei Interpolator-Takten. Ein Totzeitausgleich erfolgt durch Parametrierung der Größen des IPO/SERVO-Datenpuffers auf der NCU der Leitachse und der NCU der Lead-Link-Achse von: <ul style="list-style-type: none"> • NCU der Leitachse: 4 • NCU der Lead-Link-Achse: 2

Hinweis

Bei gleichzeitiger Verwendung von **Lead-Link-Achsen** und **NCU-übergreifenden Achscontainern**, muss aufgrund der Achscontainer das Maschinendatum MD18720 \$MN_MM_SERVO_FIFO_SIZE = **3** eingestellt werden. Eine synchrone Ausgabe der Sollwerte für Leit- und Folgeachse ist dadurch nicht möglich. Der Versatz beträgt dann einen Interpolator-Takt.

Leit-, Lead-Link- und Folgeachse

NC-spezifische Maschinendaten

Nummer	Bezeichner \$MN_	Bedeutung
MD10000	AXCONF_MACHAX_NAME_TAB	Maschinenachsname
MD10002	AXCONF_LOGIC_MACHAX_TAB	logisches Maschinenachsabbild

Kanalspezifische Maschinendaten

Nummer	Bezeichner \$MC_	Bedeutung
MD20070	AXCONF_MACHAX_USED	Verwendete Maschinenachse

Achsspezifische Maschinendaten

Nummer	Bezeichner \$MA_	Bedeutung
MD30554	AXCONF_ASSIGN_MASTER_NCU	Master-NCU Kann eine Maschinenachse von mehreren NCU aus verfahren werden, muss eine NCU als Master-NCU festgelegt werden. Die Sollwert-erzeugung erfolgt nach dem Hochlauf der Steuerung von dieser NCU aus.

14.2.5.3 Systemvariablen zur Leitwertvorgabe

Über folgende Systemvariable können auf der NCU der Leitachse Leitwerte vorgegeben werden:

- Positionsleitwert: \$AA_LEAD_SP[<Leitachse>]
- Geschwindigkeitsleitwert: \$AA_LEAD_SV[<Leitachse>]

Bei Änderung werden die Werte per NCU-Link auch auf die NCU der Folgeachse übertragen.

Hinweis

Die Übertragungspriorität von Systemvariablen ist geringer als die von Link-Variablen.

14.2.5.4 Randbedingungen

Es sind folgende Randbedingungen zu beachten:

- die Leitachse darf keine Link-Achse sein
- die Leitachse darf keine Container-Achse sein
- die Leitachse darf keine Gantry-Achse sein
- die Leitachse darf nur innerhalb ihrer eigenen NCU getauscht werden (siehe Kapitel "K10: Kanalübergreifender Achstausch (Seite 553)")

- Kopplungen mit Lead-Link-Achsen dürfen nicht kaskadiert werden
- eine Lead-Link-Achse darf nicht unabhängig von der Leitachse verfahren werden

Hinweis

Funktionen "Lead-Link-Achsen" und "Link-Achsen"

Da die Funktionen "Lead-Link-Achsen" und "Link-Achsen" unterschiedliche Einstellungen im Maschinendatum: MD18720 \$MN_MM_SERVO_FIFO_SIZE benötigen, können sie innerhalb eines Link-Verbundes nicht gleichzeitig verwendet werden.

14.2.5.5 Beispiel

Ein ausführliches Beispiel zur Parametrierung und Programmierung einer Achskopplung mit Lead-Link-Achse findet sich im Kapitel: "Beispiele" > "Lead-Link-Achse (Seite 829)".

14.2.6 Maßsysteme innerhalb eines Link-Verbunds

Für eine NCU-übergreifende Interpolation muss auf allen NCU des Link-Verbunds das gleiche Maßsystem aktiv sein.

Gemeinsame Maßsystemumschaltung über HMI

Folgende Bedingungen müssen bei allen NCU des Link-Verbundes erfüllt sein, damit eine Maßsystemumschaltung über die HMI-Bedienoberfläche einer NCU des Link-Verbunds auch auf allen anderen NCU des Link-Verbunds durchgeführt wird:

- MD10260 \$MN_CONVERT_SCALING_SYSTEM = 1
- Für alle Kanäle:
MD20110 \$MC_RESET_MODE_MASK, Bit 0 = 1
- Alle Kanäle sind im Reset-Zustand
- Keine Achse wird in der Betriebsart JOG oder DRF oder über die PLC verfahren
- Die Funktion "konstante Scheibenumfangsgeschwindigkeit (SUG)" ist nicht aktiv

Ist auf einer NCU des Link-Verbunds eine der genannten Bedingungen nicht erfüllt, wird die Maßsystemumschaltung auf keiner NCU des Link-Verbunds durchgeführt.

Unterschiedliche Maßsysteme

Unterschiedliche Maßsysteme sind trotz aktivem Link-Verbund möglich, solange keine NCU-übergreifende Interpolation stattfindet. Die Maßsystem-Einstellungen sind dazu NCU-spezifisch im Teileprogramm oder Synchronaktion über G-Befehle (G70, G71, G700, G710) vorzunehmen.

Weitere Informationen

Funktionshandbuch Basisfunktionen; Geschwindigkeiten, Soll-/Istwertsysteme, Regelung

14.3 Beispiele

14.3.1 Link-Achse

Parametrierbeispiel für 2 NCU mit jeweils einer Link-Achse

NCU 1

Maschinendatum	Anmerkung
Allgemeine Link-Daten:	
\$MN_NCU_LINKNO = 1	Master-NCU
\$MN_MM_NCU_LINK_MASK = 1	NCU-Link aktiv setzen
\$MN_MM_SERVO_FIFO_SIZE = 3	Größe des Datenpuffers zwischen Interpolation und Lageregelung
\$MN_MM_LINK_NUM_OF_MODULES = 2	Anzahl der Link-Module
Logisches Maschinenachsabbild (LAI):	
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "AX1"	Lokale Maschinenachse
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "AX2"	Lokale Maschinenachse
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC2_AX3"	Link-Achse
Maschinenachsname, durch NCU-Kennung systemweit eindeutig:	
\$MN_AXCONF_MACHAX_NAME_TAB[0] = "NC1_A1"	
\$MN_AXCONF_MACHAX_NAME_TAB[1] = "NC1_A2"	
\$MN_AXCONF_MACHAX_NAME_TAB[2] = "NC1_A3"	
Zuordnung von Kanalachse zu Maschinenachse:	
\$MC_AXCONF_MACHAX_USED[0] = 1	1. Kanalachse zur Maschinenachse von LAI[0]
\$MC_AXCONF_MACHAX_USED[1] = 2	2. Kanalachse zur Maschinenachse von LAI[1]
\$MC_AXCONF_MACHAX_USED[2] = 3	3. Kanalachse zur Maschinenachse von LAI[2]

NCU 2

Maschinendatum	Anmerkung
Allgemeine Link-Daten:	
\$MN_NCU_LINKNO = 2	Slave-NCU
\$MN_MM_NCU_LINK_MASK = 1	NCU-Link aktiv setzen
\$MN_MM_SERVO_FIFO_SIZE = 3	Größe des Datenpuffers zwischen Interpolation und Lageregelung
\$MN_MM_LINK_NUM_OF_MODULES = 2	Anzahl der Link-Module
Logisches Maschinenachsabbild (LAI):	
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "AX1"	Lokale Maschinenachse
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "AX2"	Lokale Maschinenachse
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC1_AX3"	Link-Achse
Maschinenachsname, durch NCU-Kennung systemweit eindeutig:	
\$MN_AXCONF_MACHAX_NAME_TAB[0] = "NC2_A1"	

Maschinendatum	Anmerkung
\$MN_AXCONF_MACHAX_NAME_TAB[1] = "NC2_A2"	
\$MN_AXCONF_MACHAX_NAME_TAB[2] = "NC2_A3"	
Zuordnung von Kanalachse zu Maschinenachse:	
\$MC_AXCONF_MACHAX_USED[0] = 1	1. Kanalachse zur Maschinenachse von LAI[0]
\$MC_AXCONF_MACHAX_USED[1] = 2	2. Kanalachse zur Maschinenachse von LAI[1]
\$MC_AXCONF_MACHAX_USED[2] = 3	3. Kanalachse zur Maschinenachse von LAI[2]

14.3.2 Achscontainer-Koordinierung

Der zeitliche Verlauf wird in den folgenden Tabellen von oben nach unten dargestellt. Es wird vorausgesetzt, dass nur zwei Kanäle Achsen im Container haben.

14.3.2.1 Achscontainer-Drehung ohne Warten des Teileprogramms

Kanal 1	Kanal 2	Kommentar
AXCTWE(C1)	Teileprogramm ...	Kanal 1 gibt den Achscontainer zur Drehung frei
Teileprogramm ohne Bewegung einer Containerachse	Teileprogramm ...	
	AXCTSWE(C1)	Kanal 2 gibt Achscontainer zur Drehung frei, Drehung erfolgt, weil beide Kanäle Drehung freigegeben haben
Teileprogramm mit Bewegung einer Containerachse	Teileprogramm mit Bewegung einer Containerachse	ohne Warten

14.3.2.2 Achscontainer-Drehung mit implizitem Warten des Teileprogramms

Kanal 1	Kanal 2	Kommentar
AXCTWE(C1)	Teileprogramm ...	Kanal 1 gibt den Achscontainer zur Drehung frei
Teileprogramm mit Bewegung einer Containerachse	Teileprogramm ...	Kanal 1 wartet implizit auf Achscontainer-Drehung
	AXCTSWE(C1)	Kanal 2 gibt Achscontainer zur Drehung frei, Drehung erfolgt. Kanal 1 wird fortgesetzt.

14.3.2.3 Achscontainer-Drehung durch nur einen Kanal (z. B. Hochlauffall)

Kanal 1	Kanal 2	Kommentar
AXCTWED(C1)	Ist im RESET-Zustand	Drehung erfolgt unmittelbar

14.3.3 Achscontainer Systemvariablen auswerten

14.3.3.1 Bedingte Verzweigung

Kanal 1	Kommentar
N100 AXCTWE(CT1)	Kanal 1: Freigabe der Drehung von Achscontainer CT1
MARKE1:	
N200 ...	Teileprogramm ohne Bewegung einer Containerachse
IF \$AC_AXCTSWA[CT1] == 1 GOTOB MARKE1	IF Drehung von Achscontainer CT1 noch aktiv THEN weiter bei MARKE1 ELSE (Drehung von Achscontainer CT1 abgeschlossen)
N300 ...	Teileprogramm mit Bewegung einer Containerachse

14.3.3.2 Statische Synchronaktion mit \$AN_AXCTSWA

Kanal 1	Kommentar
IDS =1 EVERY \$AN_AXCTSWA[CT1] == 1 DO M99	Statische Synchronaktion: Immer zu Beginn einer Achscontainer-Drehung die Hilfsfunktion M99 ausgeben.
	Weitere Informationen: Funktionshandbuch Synchronaktionen

14.3.3.3 Sicher auf Achscontainer-Drehung warten

Hinweis

Obwohl die Programmierung der Systemvariable \$AN_AXCTSWA einen impliziten Vorlaufstopp auslöst, kann die nachfolgende Befehlsfolge nicht dazu verwendet werden, den Abschluss einer angeforderten Achscontainer-Drehung im Kanal eindeutig festzustellen:

```
WHILE ($AN_AXCTSWA[<ID>] == TRUE) ; Warten: bis die Freigabe der
Achscontainer-Drehung gelöscht wurde

ENDWHILE
```

Während der Kanal in der Befehls-Sequenz auf den Abschluss der Achscontainer-Drehung wartet, könnte, ausgelöst von einem anderen Kanal, im Hintergrund ein Reorganisieren durchgeführt werden. Beispielsweise durch das Übergeben einer angeforderten Achse an einen anderen Kanal. Dadurch würde die Systemvariable dann den Wert FALSE liefern. Auch wenn die Achscontainer-Drehung noch nicht abgeschlossen wurde.

Empfohlene Befehls-Sequenz:

```
RL = $AN_AXCTAS[<ID>] ; Lesen: aktuelle Achscontainer-Stellung
AXCTSWE(<ID>) ; Freigabe: Achscontainer-Drehung
WHILE (RL == $AN_AXCTAS[<ID>]) ; Warten: bis die gelesene ungleich
der aktuellen Achscontainer-
; Stellung ist

ENDWHILE
```

RL ist dabei eine beliebige Anwendervariable.

Beispiel 1: Explizites Warten mittels Systemvariable \$AN_AXCTAS

Programmcode	Kommentar
RL = \$AN_AXCTAS[<ID>]	; Lesen: aktuelle Achscontainer-Stellung
AXCTSWE(<ID>)	; Freigabe: Achscontainer-Drehung
WHILE (RL == \$AN_AXCTAS[<ID>])	; Warten: Änderung der Achscontainer-
ENDWHILE	; Stellung

Beispiel 2: Explizites Warten mittels WAIT-Marke

Programmcode	Kommentar
CLEARM(9)	; Löschen: Marker 9 im Kanal
AXCTSWE(<ID>)	; Freigabe: Achscontainer-Drehung
; SynAct: Warten auf Abschluss der Achscontainer-Drehung => Setzen: Marker 9 im Kanal	
WHEN \$AN_AXCTSWA[<ID>] == TRUE DO SETM(9)	
WAITMC(9, 1)	; Warten: Marker 9 von Kanal 1

Beispiel 3: Implizites Warten mittels Container-Spindel

Programmcode	Kommentar
<pre>; Container-Spindel (erneut) programmieren ; Intern wird vor dem Start der Container-Spindel auf das Ende ; der Achscontainer-Drehung gewartet M3 S100</pre>	<pre>; Master-Spindel des Kanals</pre>

Beispiel 4: Implizites Warten mittels Container-Achse

Programmcode	Kommentar
<pre>; Container-Achse inkrementell ohne Verfahrweg programmieren ; Intern wird vor dem Start der Container-Achse auf das Ende ; der Achscontainer-Drehung gewartet X=IC(0)</pre>	<pre>; Kanalachse X</pre>

14.3.4 Konfiguration einer Mehrspindel-Drehmaschine

Einführung

Das folgende Beispiel enthält die Nutzung von:

- Mehreren NCUs im NCU-Link-Verband
- Flexible Konfiguration mit Achscontainern

Maschinenbeschreibung

- Die Maschine hat auf den Umfang einer Trommel A (Vorderseitenbearbeitung) verteilt:
 - 4 Hauptspindeln HS1 bis HS4
Jede Hauptspindel verfügt über die Möglichkeit der Materialzuführung (Stangen, Stangennachschub hydraulisch, Achsen: STN1-STN4).
 - 4 Kreuzschlitten
 - Jeder Schlitten hat zwei Achsen.
 - Optional kann auf jedem Schlitten ein angetriebenes Werkzeug S1-S4 arbeiten.
- Die Maschine hat auf den Umfang einer Trommel B (Hinterseitenbearbeitung) verteilt:
 - 4 Gegenspindeln GS1 bis GS4
 - 4 Kreuzschlitten
 - Jeder Schlitten hat zwei Achsen.
 - Optional kann auf jedem Schlitten ein angetriebenes Werkzeug S5-S8 arbeiten.
 - Jede Gegenspindel kann durch eine Linearachse in ihrer Lage verschoben werden zum Beispiel für die Übernahme von Teilen aus der Hauptspindel für die Hinterseitenbearbeitung in der Trommel B. (Übernahme-Achsen. Achsen: ZG1-ZG4).
- Kopplungen:
 - Wenn Trommel A sich dreht, werden **alle** Hauptspindeln dieser Trommel einer anderen Gruppe von Schlitten untergeordnet.
 - Wenn Trommel B sich dreht, werden **alle** Gegenspindeln und alle Übernahmeachsen dieser Trommel einer anderen Gruppe von Schlitten untergeordnet.
 - Die Drehungen der Trommeln A und B sind autark.
 - Die Drehungen der Trommeln A und B sind auf 270 Grad begrenzt. (Reichweite und Verdrehung von Versorgungsleitungen)

Begriff: Lage

Hauptspindel HS_i und Gegenspindel GS_i mit ihren Schlitten kennzeichnen eine Lage.

NCU-Zuordnung

Die Achsen und Spindeln einer Lage werden (für dieses Beispiel) jeweils einer NCU zugeordnet. Eine der NCUs, die Master-NCU, steuert zusätzlich die Achsen für die Drehung der Trommeln A und B. Es ergeben sich 4 NCUs mit maximal folgenden Achsen:

Achszahl

Je NCU_i sind folgende Achsen/Spindeln zu konfigurieren:

Schlitten 1: X_{i1} , Z_{i1}

2: X_{i2} , Z_{i2}

Spindeln: HS_i , GS_i , angetriebene Werkzeuge: S1, S2

Übernahme-Achse: ZG_i

Stangenzuführung: STN_i

Für die Master-NCU kommen die zwei Achsen für die Drehung der Trommeln A und B noch zu den oben genannten Achsen hinzu. Die Aufstellung zeigt, dass die Achszahl für insgesamt 4 Lagen durch eine NCU nicht konfiguriert werden könnte. (Grenze 31 Achsen, benötigt werden $4 + 10 + 2$ Achsen).

Achscontainer

HS_i , GS_i , ZG_i und STN_i sind bei Drehung der Trommeln A/B einer anderen NCU zuzuordnen und müssen deshalb als Link-Achsen in Achscontainern konfiguriert werden.

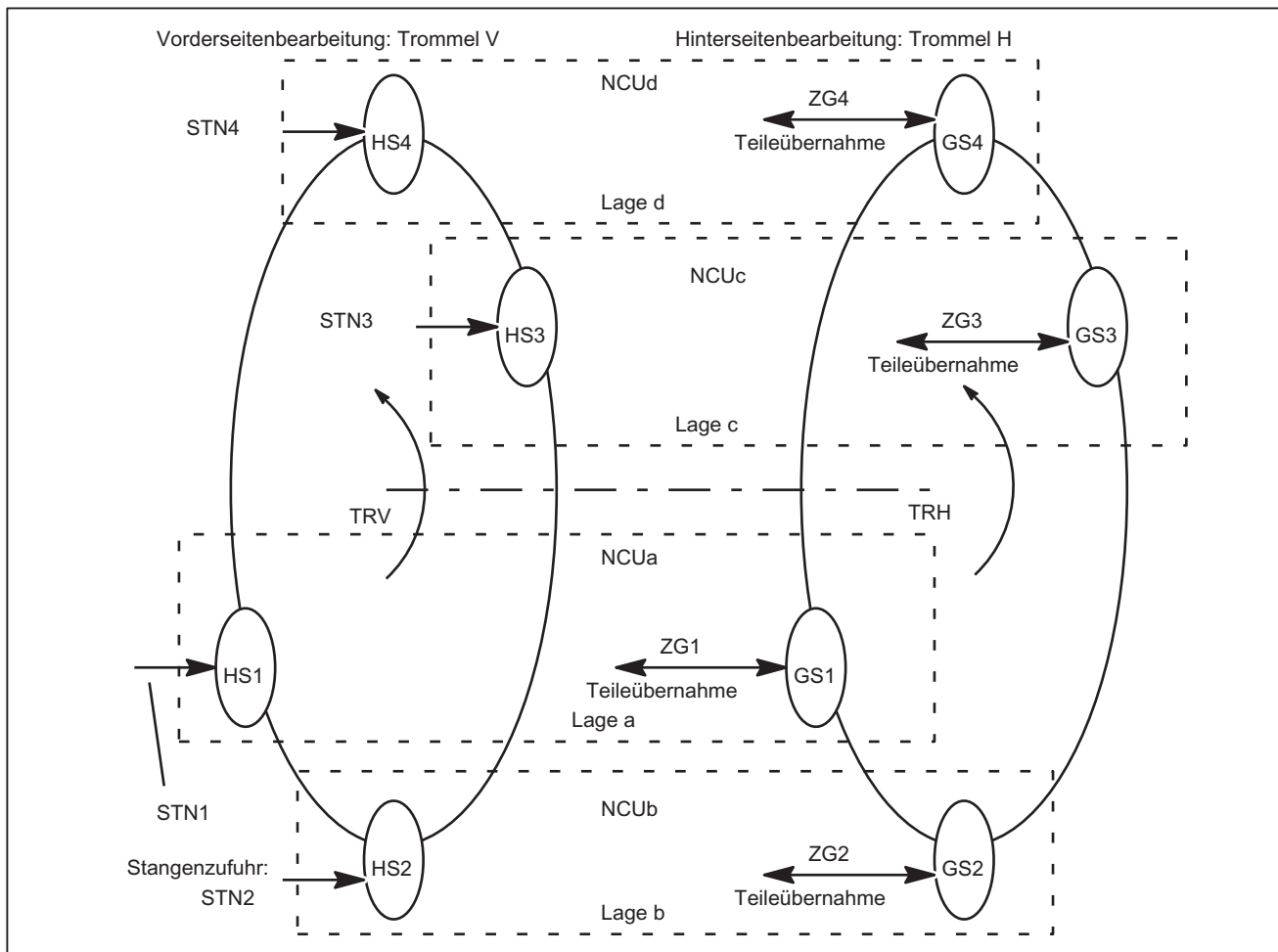


Bild 14-21 Hauptspindeln HS_i , Gegensp. GS_i , Stangenzufuhrachse STN_i und Übernahme-Achsen ZG_i schematisch

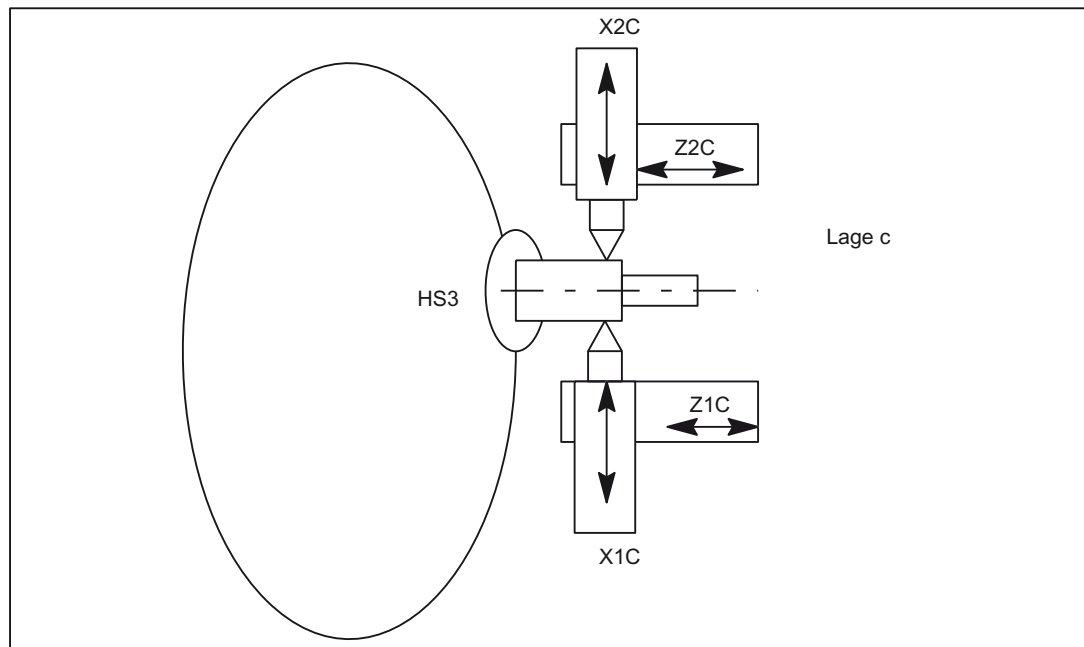


Bild 14-22 Zwei Schlitten je Lage können auch gemeinsam an einer Spindel arbeiten.

Hinweis

Zur Verdeutlichung der Zuordnung von Achsen zu Schlitten und Lagen werden die Achsen wie folgt benannt:

Xij mit i Schlitten (1, 2), j Lage (A-D)

Zij mit i Schlitten (1, 2), j Lage (A-D)

Lagen und ihre Schlitten bleiben ortsfest, während Hauptspindeln, Gegenspindeln, Stangenzführungsachsen STN und Übernahme-Achsen ZG sich durch Drehungen der Trommeln V oder H in neue Lagen bewegen.

Aus obigen Bildern folgen mit Berücksichtigung der Schlitten z. B. je NCU folgende zu verwaltende Achsen:

Achsen der Master-NCU

Tabelle 14-1 Achsen der Master-NCU: NCUa

gemeinsame Achsen	lokale Achsen	Bemerkung
	TRV (Trommel V)	nur Master-NCU
	TRH (Trommel H)	nur Master-NCU
	X1A	Schlitten 1
	Z1A	Schlitten 1
	X2A	Schlitten 2

gemeinsame Achsen	lokale Achsen	Bemerkung
	Z2A	Schlitten 2
	S1	Schlitten 1
	S2	Schlitten 2
HS1		Achscontainer nötig
GS1		Achscontainer nötig
ZG1		Achscontainer nötig
STN1		Achscontainer nötig
4	8	

Achsen der NCUb bis NCUd

Die NCUs, die nicht Master-NCU sind, haben die gleichen Achsen mit Ausnahme der Achsen für den Antrieb der Trommeln TRV und TRH. Der für die Lage kennzeichnende Buchstabe ist bei NCU und den Achsnamen entsprechend zu ersetzen (a, A → b, B bis d, D).

Konfigurationsregeln

Für die unten angegebene Konfiguration wurden folgende Regeln berücksichtigt:

- Hauptspindel, Gegenspindeln und Achsen, die während ihrer Nutzung gemäß obigem Bild "Hauptspindel ..." verschiedenen NCUs durch Trommeldrehung zugeordnet werden, müssen in einem Achscontainer konfiguriert werden. (HS_i, GS_i, ZG_i, STN_i).
- Alle Hauptspindeln der Trommel A stehen im gleichen Container (Nr. 1).
- Alle Stangenzuführungsachsen der Trommel A stehen im gleichen Container (Nr. 2).
- Alle Gegenspindeln der Trommel B stehen im gleichen Container (Nr. 3).
- Alle Übernahmeachsen der Trommel B stehen im gleichen Container (Nr. 4).
- Hauptspindeln HS_i und ihre Gegenspindel GS_i, sowie die Übernahme-Achsen der Gegenspindel ZG_i und die Stangenzuführungsachsen STN_i der Hauptspindel sind aus Gründen der NCU-Belastungsgleichverteilung wie folgt zugeordnet:
NCUa HS1-STN1,
NCUb HS2-STN2, ...usw.
- Schlittenachsen X_{ij}, Z_{ij} sind rein lokale Achsen mit fester NCU-Zuordnung.
- Schlitten sind jeweils einem eigenständigen Kanal einer NCU zugeordnet. Schlitten können damit autark bewegt werden.

Möglichkeiten der Konfiguration

- Haupt- oder Gegenspindeln sind flexibel auf den Schlitten zuzuordnen.
- In jeder Lage kann die Drehzahl der Hauptspindel und der Gegenspindel eigenständig bestimmt werden.
Ausnahmen:
Während des Teilewechsels von der Vorderseitenbearbeitung in Trommel V zur Hinterseitenbearbeitung in Trommel H müssen Hauptspindel und Gegenspindel auf gleiche Drehzahl gebracht werden (Synchronspindelkopplung).
Wenn Schlitten 2 zur "Unterstützung" von Schlitten 1 ebenfalls an der Vorderseitenbearbeitung mitwirkt, gilt für diesen Fall die Hauptspindeldrehzahl auch für Schlitten 2. Entsprechend, wenn Schlitten 1 an der Hinterseitenbearbeitung mitwirkt, gilt auch für Schlitten 1 die Gegenspindeldrehzahl.

Kleine Drehzahländerungen

Bei NCU-übergreifenden Bearbeitungen sollten wegen der notwendigen Zeitausgleiche für die Zusammenführung von Istwerten starke Drehzahländerungen vermieden werden. Vergl. Achsdaten und Signale.

Konfiguration für NCU1

Einheitliche Benutzung der Kanal-Achsamen in den Teileprogrammen:

S4: Hauptspindel

S3: Gegenspindel

X1: Zustellachse

Z1: Längsachse

S1: Angetriebenes Werkzeug

Z3: Übernahme-Achse

TRV: Trommel V für Hauptspindel

TRH: Trommel H für Gegenspindel

STN: Stangenzuführung hydraulisch

Durch **Fett**druck hervorgehobene Achsen kennzeichnen den aktuellen Kanal als Heimatkanal für die Achse im Zusammenhang mit Achstausch.

Tabelle 14-2 NCUa, Lage: a, Kanal: 1, Schlitten: 1

Kanalachs-Name	..._MA- CHAX_USE D	\$MN_ AXCONF_LOGIC_MA- CHAX_TAB	Container, Slot Eintrag (String)	Maschinenachs-Name
S4	1	AX1: CT1_SL1	1 1 NC1_AX1	HS1
S3	2	AX2: CT3_SL1	3 1 NC1_AX2	GS1
X1	3	AX3:		X1A

14.3 Beispiele

Kanalachs-Name	..._MA- CHAX_USE D	\$MN_ AXCONF_LOGIC_MA- CHAX_TAB	Container, Slot Eintrag (String)	Maschinenachs-Name
Z1	4	AX4:		Z1A
Z3	5	AX5: CT4_SL1	4 1 NC1_AX5	ZG1
S1	6	AX6:		WZ1A
STN	7	AX7: CT2_SL1	2 1 NC1_AX7	STN1
TRV	11	AX11:		TRV
TRH	12	AX12:		TRH
x2 *				
z2 *				

Tabelle 14-3 NCUa, Lage: a, Kanal: 2, Schlitten: 2

Kanalachs-Name	..._MA- CHAX_USE D	\$MN_ AXCONF_LOGIC_MA- CHAX_TAB	Container, Slot Eintrag (String)	Maschinenachs-Name
S4	1	AX1: CT1_SL1	1 1 NC1_AX1	HS1
S3	2	AX2: CT3_SL1	3 1 NC1_AX2	GS1
Z3	5	AX5: CT4_SL1	4 1 NC1_AX5	ZG1
STN	7	AX7: CT2_SL1	2 1 NC1_AX7	STN1
X2	8	AX8:		X2A
Z2	9	AX9:		Z2A
S1	10	AX10:		WZ2A
x1 *				
z1 *				

Hinweis

* wegen Programmkoordinierung über Achspositionen und 4-Achsbearbeitung in einer Lage.

Für den Eintrag in einem Platz des Achscontainers ist die Form: "NC1_AX.." erforderlich, mit der Bedeutung NC1 = NCU 1. In den obigen Tabellen wird NCUa abgebildet auf NC1_..., NCUb auf NC2_... usw.

Weitere NCUs

Die oben angegebenen Konfigurationsdaten müssen entsprechend für NCUb bis NCUd angegeben werden. Dabei ist folgendes zu beachten:

- Die Achsen TRA und TRB gibt es nur für NCUa , Kanal 1.
- Die Container-Nummern bleiben für die weiteren NCUs so erhalten, wie für die einzelnen Achsen angegeben
- Die Slot-Nummern werden für:
NCUb → 2
NCUc → 3
NCUd → 4.
- Die Maschinenachsnamen werden für:
NCUb → HS2, GS2, ZG2, STN2
NCUc → HS3, GS3, ZG3, STN3
NCUd → HS4, GS4, ZG4, STN4.

Achscontainer

Die in der Tabelle 7-17, enthaltenen Informationen bezüglich der Container sowie die Container-Einträge der gleichartig konfigurierten NCUs NCUb bis NCUd werden in den folgenden Tabellen nach Containern und Slots geordnet so angegeben, wie sie im Maschinendatum:

MD12701 \$MN_AXCT_AXCONF_ASSIGN_TAB1[slot]

...

MD12716 \$MN_AXCT_AXCONF_ASSIGN_TAB16[slot]

mit slot: 1-4 für die 4 Lagen einer Mehrspindel-Drehmaschine gesetzt werden müssen:

Hinweis

Für den Maschinendaten-Eintrag

\$MN_AXC_AXCONF_ASSIGN_TAB_i[slot]

sind die in den obigen Tabellen unter Ausgangslage eingetragenen Werte (ohne Komma und Maschinenachsnamen) zu setzen.

Tabelle 14-4 Achscontainer und ihre lageabhängigen Inhalte für Trommel A

Container	Slot	Ausgangslage (TRA 0°)	Switch 1 (TRA 90°)	Switch 2 (TRA 180°)	Switch 3 (TRA 270°)	Switch 4 = (TRA 0°)
1	1	NC1_AX1, HS1	NC2_AX1, HS2	NC3_AX1, HS3	NC4_AX1, HS4	NC1_AX1, HS1
	2	NC2_AX1, HS2	NC3_AX1, HS3	NC4_AX1, HS4	NC1_AX1, HS1	NC2_AX1, HS2
	3	NC3_AX1, HS3	NC4_AX1, HS4	NC1_AX1, HS1	NC2_AX1, HS2	NC3_AX1, HS3
	4	NC4_AX1, HS4	NC1_AX1, HS1	NC2_AX1, HS2	NC3_AX1, HS3	NC4_AX1, HS4

14.3 Beispiele

Container	Slot	Ausgangslage (TRA 0°)	Switch 1 (TRA 90°)	Switch 2 (TRA 180°)	Switch 3 (TRA 270°)	Switch 4 = (TRA 0°)
2	1	NC1_AX7, STN1	NC2_AX7, STN2	NC3_AX7, STN3	NC4_AX7 STN4	NC1_AX7, STN1
	2	NC2_AX7, STN2	NC3_AX7, STN3	NC4_AX7, STN4	NC1_AX7, STN1	NC2_AX7, STN2
	3	NC3_AX7, STN3	NC4_AX7, STN4	NC1_AX7, STN1	NC2_AX7, STN2	NC3_AX7, STN3
	4	NC4_AX7, STN4	NC1_AX7, STN1	NC2_AX7, STN2	NC3_AX7, STN3	NC4_AX7, STN4
Trommelbewegung		0°	+ 90°	+ 90°	+ 90°	- 270°

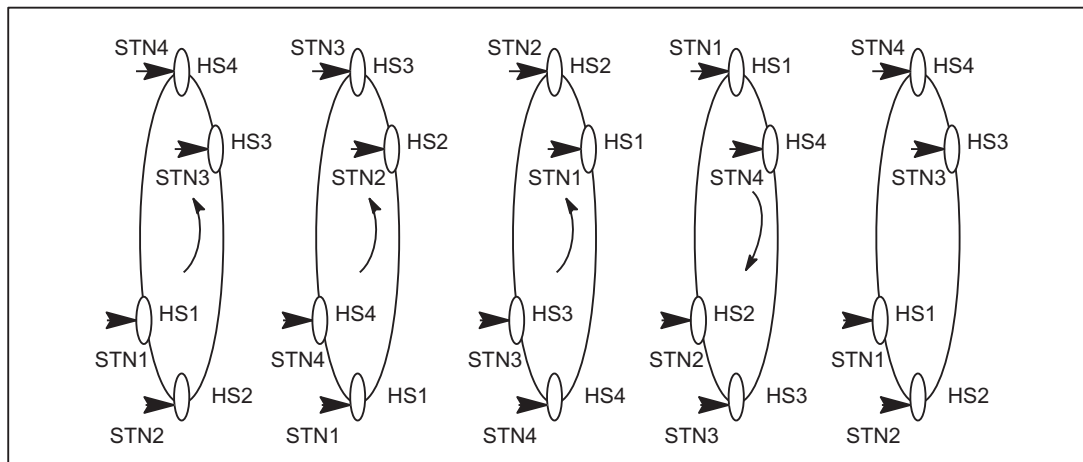


Bild 14-23 Positionen der Trommel A

Tabelle 14-5 Achscontainer und ihre lageabhängigen Inhalte für Trommel B

Container	Slot	Ausgangslage (TRB 0°)	Switch 1 (TRB 90°)	Switch 2 (TRB 180°)	Switch 3 (TRB 270°)	Switch 4 = (TRB 0°)
3	1	NC1_AX2, GS1	NC2_AX2, GS2	NC3_AX2, GS3	NC4_AX2, GS4	NC1_AX2, GS1
	2	NC2_AX2, GS2	NC3_AX2, GS3	NC4_AX2, GS4	NC1_AX2, GS1	NC2_AX2, GS2
	3	NC3_AX2, GS3	NC4_AX2, GS4	NC1_AX2, GS1	NC2_AX2, GS2	NC3_AX2, GS3
	4	NC4_AX2, GS4	NC1_AX2, GS1	NC2_AX2, GS2	NC3_AX2, GS3	NC4_AX2, GS4
4	1	NC1_AX5, ZG1	NC2_AX5, ZG2	NC3_AX5, ZG3	NC4_AX5 ZG4	NC1_AX5, ZG1
	2	NC2_AX5, ZG2	NC3_AX5, ZG3	NC4_AX5, ZG4	NC1_AX5, ZG1	NC2_AX5, ZG2
	3	NC3_AX5, ZG3	NC4_AX5, ZG4	NC1_AX5, ZG1	NC2_AX5, ZG2	NC3_AX5, ZG3
	4	NC4_AX5, ZG4	NC1_AX5, ZG1	NC2_AX5, ZG2	NC3_AX5, ZG3	NC4_AX5, ZG4

14.3.5 Lead-Link-Achse

14.3.5.1 Konfiguration

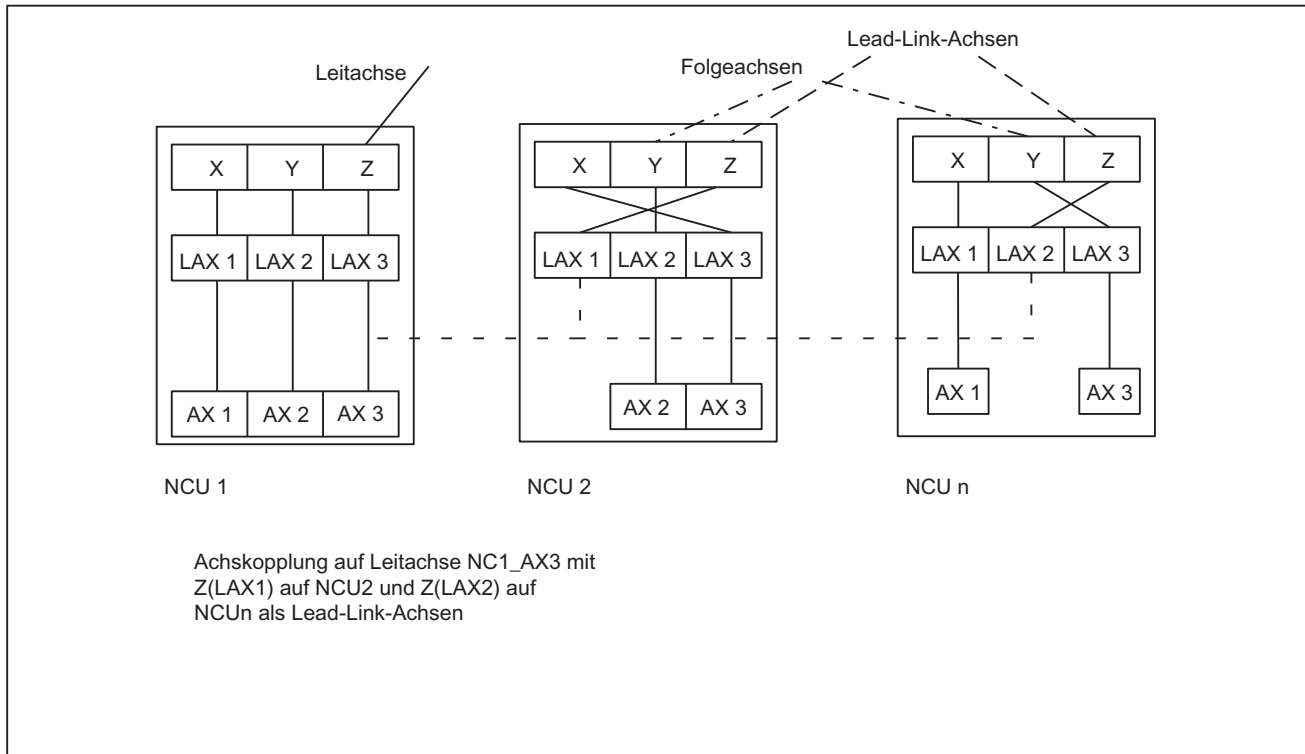


Bild 14-24 NCU2 bis NCU n benutzen eine Lead-Link-Achse, um eine Kopplung die auf Maschinenachse an NCU1. (NCU1-AX3) zu ermöglichen

Das folgende Beispiel bezieht sich auf den Ausschnitt der Kopplung zwischen Y(LAX2, AX2) als Folgeachse auf NCU2 und Z(LAX3, NC1_AX3) als Lead-Link-Achse.

Maschinendaten

- Das Laden der Maschinendaten einer Leitwert-Achse muss nur auf der Heimat-NCU erfolgen. Von dieser aus werden die relevanten Maschinendaten an die anderen NCU, bei denen eine Lead-Link-Achse definiert ist, verteilt.
- Jede Lead-Link-Achse reduziert die maximale Anzahl von Achsen die auf dieser NCU verfahren können um eine Achse.

Maschinendaten für NCU1 (Leitachse)

Maschinendatum	Bedeutung
\$MN_NCU_LINKNO = 1	1. bzw. Master-Ncu
\$MN_MM_NCU_LINK_MASK = 1	NCU-Link aktiv
\$MN_MM_LINK_NUM_OF_MODULES= 2	Anzahl der Link-Module

Maschinendatum	Bedeutung
\$MN_MM_SERVO_FIFO_SIZE = 4	Größe des Datenpuffers zwischen Interpolation und Lageregelung auf 4 erhöht
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "AX1"	
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "AX2"	
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "AX3"	
\$MN_AXCONF_MACHAX_NAME_TAB[0] = "XM1"	
\$MN_AXCONF_MACHAX_NAME_TAB[2] = "YM1"	
\$MA_AXCONF_ASSIGN_MASTER_NCK[AX3] = 1	
\$MC_AXCONF_MACHAX_USED[0]=1 ; X	
\$MC_AXCONF_MACHAX_USED[1]=2 ; Y	
\$MC_AXCONF_MACHAX_USED[2]=3 ; Z	

Maschinendaten für NCU2 (Folgeachse)

Maschinendatum	Bedeutung
\$MN_NCU_LINKNO = 2	2. NCU-Nummer
\$MN_MM_NCU_LINK_MASK = 1	Link aktivieren
\$MN_MM_NUM_CURVE_TABS = 5	Anzahl Kurventabellen
\$MN_MM_LINK_NUM_OF_MODULES= 2	Anzahl der Link-Module
\$MN_MM_NUM_CURVE_SEGMENTS= 50	
\$MN_MM_NUM_CURVE_POLYNOMS = 100	
\$MN_MM_SERVO_FIFO_SIZE = 2	Größe des Datenpuffers zwischen Interpolation und Lagerregelung (Standard)
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "NC1_AX3"	Lead-Link auf NCU1/AX3
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "AX2"	
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "AX3"	
\$MC_AXCONF_MACHAX_USED[0]=3	AX3 ist Maschinenachse der 1. Kanalachse
\$MC_AXCONF_MACHAX_USED[1]=2	AX2 ist Maschinenachse der 2. Kanalachse
\$MC_AXCONF_MACHAX_USED[2]=1	AX3 auf NCU1 ist Maschinenachse der 3. Kanalachse

14.3.5.2 Programmierung

Programm für NCU1 (Leitachse)

NCU1 verfährt Leitachse Z.

Kennung für NCU2, dass die Leitachse von NCU1 belegt ist: Link-Variable \$A_DLB[0] = 1

Kennung für NCU2, dass die Leitachse von NCU1 freigegeben ist: Link-Variable \$A_DLB[0] = 0

Programmcode	Kommentar
N1000 R1 = 0	; Schleifenzähler initialisieren
N1004 G1 Z0 F1000	; Achse Z in Ausgangsposition fahren
N1005 \$A_DLB[0] = 1	; Kennung für NCU2: Achse Z ist belegt
LOOP10:	;
N1005 R1=R1+1	; Schleifenzähler erhöhen
N1006 Z0.01 G91	; Verfahren der Leitwert-Achse Z
N1008 Z0.02	; Verfahren der Leitwert-Achse Z
N1010 Z0.03	; Verfahren der Leitwert-Achse Z
N1012 IF R1 < 10 GOTOB LOOP10	;
N1098 \$A_DLB[0] = 0	; Kennung für NCU2: Achse Z ist frei

Programm für NCU2 (Folgeachse)

Das Programm stellt durch eine Kurventabelle einen Zusammenhang zwischen Leitachsbewegung auf NCU1 und Folgeachsbewegung auf NCU2 her. Ist die Tabelle definiert, geht NCU2 in Wartestellung (N2006) bis NCU1 die Achse Z als Leitachse belegt hat (N1005). Sobald Achse Z als Leitachse belegt ist, wird die Kopplung aktiviert (N2010). Die Kopplung wird aufrecht erhalten, bis NCU1 die Achse Z als Leitachse freigegeben hat.

Programmcode	Kommentar
N2000 CTABDEL(1)	; Tabelle 1 initialisieren
N2001 G04 F.1	;
N2003 G0 Y0 Z0	; Achsen Y, Z in Ausgangsposition fahren
N2002 CTABDEF(Y, Z, 1, 0)	; Tabellendefinition EIN
N2003 G1 X0 Y0	; Stützstelle 1
N2004 G1 X100 Y200	; Stützstelle 2
N2005 CTABEND	; Tabellendefinition AUS
LOOP20:	;
N2006 IF (\$A_DLB[0] == 0) GOTOB LOOP20	; Warten auf NCU1
N2010 LEADON(Y,Z,1)	; => Kopplung einschalten
LOOP25:	;
N2030 IF (\$A_DLB[0] > 0) GOTOB LOOP25	; Kopplung halten bis NCU1 die Leitwert-Achse nicht mehr verfährt
N2090 LEADOF(Y,Z)	; => Kopplung ausschalten

14.4 Datenlisten

14.4.1 Maschinendaten

14.4.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10002	AXCONF_LOGIC_MACHAX_TAB	Logisches NCU-Maschinenachsenabbild
10065	POCTRL_DESVAL_DELAY	Lagesollwert-Verzögerung
10134	MM_NUM_MMC_UNITS	Anzahl gleichzeitig möglicher MMC-Kommunikationspartner
12510	NCU_LINKNO	NCU-Nummer in einem NCU-Verband
12520	LINK_TERMINATION	NCU-Nummern, bei denen Busabschlusswiderstände aktiviert sind
12530	LINK_NUM_OF_MODULES	Anzahl der Link-Module
12701	AXCT_AXCONF_ASSIGN_TAB1	Liste der Achsen im Achscontainer
...	...	
12716	AXCT_AXCONF_ASSIGN_TAB16	
12750	AXCT_NAME_TAB	Liste der Achscontainernamen
12760	AXCT_FUNCTION_MASK	Funktionen zum Achscontainer
18700	MM_SIZEOF_LINKVAR_DATA	Größe des Link-Variablen-Speichers
18720	MM_SERVO_FIFO_SIZE	Größe des Datenpuffers zwischen Interpolator und Lageregler
18780	MM_NCU_LINK_MASK, Bit 0	Aktivierung Link-Kommunikation

14.4.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20000	CHAN_NAME	Kanalname
20070	AXCONF_MACHAX_USED	Maschinenachsnummer gültig im Kanal
28160	MM_NUM_LINKVAR_ELEMENTS	Anzahl Schreibelemente für die NCU-Link-Variablen

14.4.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30550	AXCONF_ASSIGN_MASTER_CHAN	Defaultmäßige Zuordnung einer Achse zu einem Kanal
30554	AXCONF_ASSIGN_MASTER_NCU	Löschstellung, welche NCU für die Achse Sollwerte erzeugt.
30560	IS_LOCAL_LINK_AXIS	Achse ist eine lokale Linkachse
32990	POCTRL_DESVAL_DELAY_INFO	Aktuelle Lagesollwertverzögerung

14.4.2 Settingdaten

14.4.2.1 Allgemeine Settingdaten

Nummer	Bezeichner: \$SA	Beschreibung
41700	AXCT_SWWIDTH[Containernummer]	Achscontainer-Drehungsvorgabe

14.4.2.2 Achs-/Spindel-spezifische Settingdaten

Nummer	Bezeichner: \$SA_	Beschreibung
43300	ASSIGN_FEED_PER_REV_SOURCE	Umdrehungsvorschub für Positionierachsen/ Spindeln

14.4.3 Signale

14.4.3.1 Signale von NC

Signalname	SINUMERIK 840D sl
MCP1 bereit	DB10.DBX104.0
MCP2 bereit	DB10.DBX104.1
BHG bereit	DB10.DBX104.2
NCU-Link aktiv	DB10.DBX107.6
HMI2-CPU ready (HMI an BTSS oder MPI)	DB10.DBX108.1
HMI1-CPU an MPI ready	DB10.DBX108.2
HMI1-CPU an BTSS ready (Standard-Anschluss)	DB10.DBX108.3

14.4.3.2 Signale von HMI/PLC

Signalname	SINUMERIK 840D sl
ONL_REQUEST Online-Anforderung von HMI	DB19.DBB100
ONL_CONFIRM Quittung von PLC auf Online-Anforderung	DB19.DBB102
PAR_CLIENT_IDENT HMI schreibt seine Client-Identifikation (Bustyp, HMI-Bus- adresse)	DB19.DBB104
PAR_MMC_TYP Typ des HMI gemäß NETNAMES.INI: Haupt-/Nebenbe- dienfeld/Alarmserver	DB19.DBB106

Signalname	SINUMERIK 840D sl
PAR_MSTT_ADR HMI schreibt Adresse der zu aktivierenden MCP	DB19.DBB107
PAR_STATUS PLC schreibt die Online-Freigabe für den HMI (Verbindungszustand)	DB19.DBB108
PAR_Z_INFO PLC schreibt Zusatz-Info zum Verbindungszustand	DB19.DBB109
M_TO_N_ALIVE Lebenszeichen von PLC an HMI durch M zu N Baustein	DB19.DBB110

14.4.3.3 Allgemein Online-Schnittstelle

Signalname	SINUMERIK 840D sl
MMC1_CLIENT_IDENT PLC schreibt PAR_CLIENT_IDENT nach MMCx_CLIENT_IDENT, wenn HMI online geht.	DB19.DBB120
MMC1_TYP PLC schreibt PAR_MMC_TYP nach MMCx_TYP, wenn HMI online geht.	DB19.DBB122
MMC1_MSTT_ADR PLC schreibt PAR_MSTT_ADR nach MMCx_MSTT_ADR, wenn HMI online geht	DB19.DBB123
MMC1_STATUS Verbindungszustand, HMI und PLC schreiben wechselseitig ihre Anforderungen/Quittungen.	DB19.DBB124
MMC1_Z_INFO Zusatz-Info Verbindungszustand (pos./neg. Quittung, Fehlermeldungen, ...)	DB19.DBB125
MMC1_SHIFT_LOCK HMI-Umschalt Sperre	DB19.DBX126.0
MMC1_MSTT_SHIFT_LOCK MCP-Umschalt Sperre	DB19.DBX126.1
MMC1_ACTIVE_REQ HMI fordert aktiven Bedienmodus	DB19.DBX126.2
MMC1_ACTIVE_PERM Freigabe von PLC zum Wechsel des Bedienmodus	DB19.DBX126.3
MMC1_ACTIVE_CHANGED HMI hat Bedienmodus gewechselt	DB19.DBX126.4
MMC1_CHANGE_DENIED HMI-Aktiv-Passiv-Umschaltung abgelehnt	DB19.DBX126.5
MMC2_CLIENT_IDENT PLC schreibt PAR_CLIENT_IDENT nach MMCx_CLIENT_IDENT, wenn HMI online geht.	DB19.DBB130

Signalname	SINUMERIK 840D sl
MMC2_TYP PLC schreibt PAR_MMC_TYP nach MMCx_TYP, wenn HMI online geht.	DB19.DBB132
MMC2_MSTT_ADR PLC schreibt PAR_MSTT_ADR nach MMCx_MSTT_ADR, wenn HMI online geht.	DB19.DBB133
MMC2_STATUS Verbindungszustand, HMI und PLC schreiben wechselseitig ihre Anforderungen/ Quittungen.	DB19.DBB134
MMC2_Z_INFO Zusatz-Info Verbindungszustand (pos./neg. Quittung, Fehlermeldungen, ...)	DB19.DBB135
MMC2_SHIFT_LOCK HMI-Umschaltsperr	DB19.DBX136.0
MMC2_MSTT_SHIFT_LOCK MCP-Umschaltsperr	DB19.DBX136.1
MMC2_ACTIVE_REQ HMI fordert aktiven Bedienmodus	DB19.DBX136.2
MMC2_ACTIVE_PERM Freigabe von PLC zum Wechsel des Bedienmodus	DB19.DBX136.3
MMC2_ACTIVE_CHANGED HMI hat Bedienmodus gewechselt	DB19.DBX136.4
MMC2_CHANGE_DENIED HMI-Aktiv-Passiv-Umschaltung abgelehnt	DB19.DBX136.5

14.4.3.4 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
NCU-Link: Achse aktiv	DB31, ... DBX60.1	-
Achsspezifischer Alarm	DB31, ... DBX61.1	DB390x, DBX1.1
Achse betriebsbereit	DB31, ... DBX61.2	DB390x, DBX1.2
Achscontainer: Drehung aktiv	DB31, ... DBX62.7	-

14.4.4 Systemvariablen

Systemvariable	Beschreibung
\$AN_AXCTSWE[Achse]	Liefert die Slots des Achscontainers der angegebenen Achse, welche für die nächste Achscontainer-Drehung freigegeben sind
\$AN_LAI_AX_IS_AXCTAX	Enthält als Bitfeld die Container-Achsen des logischen Maschinenachs-Abbildes
\$AN_LAI_AX_IS_LINKAX	Enthält als Bitfeld die Link-Achsen des logischen Maschinenachs-Abbildes

Systemvariable	Beschreibung
\$AN_LAI_AX_IS_LEADLINKAX	Enthält als Bitfeld die Lead-Link-Achsen des logischen Maschinenachs-Abbildes
\$AN_LAI_AX_TO_MACHAX[Achse]	Liefert für die angegebene Achse des logischen Maschinenachs-Abbildes die NCU-ID und die Achsnummer der zugeordneten Maschinenachse
\$AN_LAI_AX_TO_IPO_NC_CHANAX[Achse]	Liefert für die angegebenen Achse des logischen Maschinenachs-Abbildes die Kanal- und Kanalachs-Nummer bzw. NCU- und globale Achsnummer
\$AN_IPO_CHANAX[globale Achsnummer]	Liefert für die angegebenen globale Achsnummer die Kanal- und Kanalachs-Nummer
\$AA_MACHAX[Achse]	Liefert für die angegebenen Achse die NCU-ID und Maschinenachs-Nummer
\$AA_IPO_NC_CHANAX[Achse]	Liefert für die angegebenen Achse die Kanal- und Kanalachs-Nummer oder NCU-ID und globale Achsnummer
\$VA_IPO_NC_CHANAX[Achse]	Liefert für die angegebenen Maschinenachse die Kanal- und Kanalachs-Nummer oder NCU-ID und globale Achsnummer

Ausführliche Erläuterungen zu den Systemvariablen finden sich in:

Weitere Informationen:

Listenhandbuch Systemvariablen

S7: Speicherkonfiguration

15.1 Einführung

Speicherbereiche

Zur Datenhaltung und -verwaltung der **lokalen** persistenten und nicht-persistenten Daten der NC sind auf der CF-Card der NCU zwei Speicherbereiche vorhanden:

- **Statischer NC-Speicher**
Der statische NC-Speicher enthält die **persistenten** NC-Daten des aktiven und passiven Filesystems (Seite 837).
- **Dynamischer NC-Speicher**
Der dynamische NC-Speicher enthält die von der NC dynamisch erzeugten, **nicht-persistenten** NC-Daten wie z. B. Makros, Lokale User-Daten, Zwischenspeicher, etc.

Deterministisches Verhalten

Um in allen Bearbeitungssituationen ein deterministisches Verhalten der NC sicherzustellen, haben alle Speicherbereiche des lokalen statischen und dynamischen NC-Speichers eine einstellbare aber feste Größe.

Speicherkonfiguration

Die Speicherbereiche des lokalen statischen und dynamischen NC-Speichers werden im erstmaligen Systemhochlauf anhand der Standardeinstellungen der speicherkonfigurierenden Maschinendaten eingerichtet. Im Normalfall sind diese Einstellungen ausreichend.

Umkonfiguration

Bei Bedarf kann die Speicherkonfiguration an anwenderspezifische Anforderungen angepasst werden. Durch die dazu notwendige Umkonfiguration des Speichers kann es zum Verlust sämtlicher Anwenderdaten im statischen NC-Speichers kommen. Es wird daher nach einer Änderung eines speicherkonfigurierenden Maschinendatums der Alarm 4400: "MD-Änderung bewirkt Reorganisation des gepufferten Speichers, (Datenverlust!)" angezeigt. Vor der Aktivierung der Maschinendatenänderung bzw. der Umkonfiguration des Speichers durch NC-Reset sollte ein Inbetriebnahmearchiv erstellt werden.

Mit dem nächsten Hochlauf der NC wird Speicherkonfiguration geändert. Anschließend kann dann zur Wiederherstellung der Anwenderdaten das erstellte Inbetriebnahmearchiv eingelesen werden.

15.2 Aktives und passives Filesystem

Im lokalen statischen NC-Speicher sind die anwenderspezifischen Daten der NC abgelegt. In diesem Speicherbereich liegen die Daten des aktiven und passiven Filesystems.

Aktives Filesystem

Das aktive Filesystem beinhaltet die Systemdaten zur Parametrierung der NC. Im Wesentlichen sind dies:

- Maschinendaten
- Settingdaten
- Optionsdaten
- Globale Anwenderdaten (GUD)
- Werkzeugkorrektur-/Magazin-Daten
- Schutzbereiche
- R-Parameter
- Nullpunktverschiebungen/FRAME
- Durchhangkompensationen
- Quadrantenfehlerkompensation
- Spindelsteigungsfehler-Kompensation

Die Daten des aktiven Filesystems repräsentieren die aktuellen Arbeitsdaten der NC.

Die Sicht auf das aktive Filesystem ist Daten-orientiert.

Passives Filesystem

Das passive Filesystem beinhaltet alle lokal in der NC geladenen Dateien:

- Hauptprogramme
- Unterprogramme
- Werkstücke
- Definitionsdateien (*.DEF) der globalen Anwenderdaten und Makros
- Standard-Zyklen
- Anwender-Zyklen
- Kommentare
- Binärdateien (z. B. Bilder, PDF-Dokumente)

Die Sicht auf das passive Filesystem ist Datei-orientiert.

15.3 Inbetriebnahme

15.3.1 Konfiguration

Die Konfiguration des lokalen statischen und dynamischen NC-Speichers wird durch folgende Maschinendaten eingestellt bzw. beeinflusst:

- Speicherkonfigurierende Maschinendaten:
 - \$MN_MM... (NC-spezifische speicherkonfigurierende Maschinendaten)
 - \$MC_MM... (kanalspezifische speicherkonfigurierende Maschinendaten)
 - \$MA_MM... (achsspezifische speicherkonfigurierende Maschinendaten)
- Anzahl parametrierter Kanäle:
 - MD10010 \$MN_ASSIGN_CHAN_TO_MODE_GROUP (Kanal gültig in Betriebsartengruppe)

15.3.2 Umkonfiguration

Beim erstmaligen Hochlauf der Steuerung werden die Größen der Speicherbereiche des lokalen NC-Speichers anhand der Standardeinstellungen der speicherkonfigurierenden Maschinendaten eingestellt. Im Allgemeinen sind diese Einstellungen für den Betrieb der Steuerung ausreichend. Werden die Speicherbereiche des lokalen NC-Speichers aufgrund besonderer anwenderspezifischer Anforderungen geändert, führt dies zu einer Umkonfiguration des lokalen NC-Speichers. Ist dabei das aktive und/oder passive Filesystem des **statischen** NC-Speichers betroffen, wird der Alarm 4400 angezeigt. Vom Anwender sollten daraufhin die relevanten Daten durch Erstellen eines Inbetriebnahmearchivs auf ein externes Speichermedium gesichert werden.

In folgenden Fällen wird der Alarm 4400 angezeigt:

- ein speicherkonfigurierendes Maschinendatum des passiven Filesystems wurde geändert
- ein speicherkonfigurierendes Maschinendatum des aktiven Filesystems wurde geändert und die Funktion "Automatic Memory Reconfiguration" (AMR) ist **nicht aktiv**
- ein speicherkonfigurierendes Maschinendatum des aktiven Filesystems wurde geändert, die Funktion AMR ist **aktiv**, aber die Daten des aktiven Filesystems können nicht lokal zwischengespeichert werden

ACHTUNG

Datenverlust durch Umkonfiguration

Eine Umkonfiguration des lokalen **statischen** NC-Speichers kann den Verlust der Anwenderdaten des aktiven und/oder passiven Filesystems zur Folge haben. Es wird daher dringend empfohlen nach der Anzeige des Alarms 4400, **vor** Aktivierung der geänderten Speicherkonfiguration durch NC-Reset, alle relevanten Daten durch Erstellen eines Inbetriebnahmearchivs zu sichern.

Funktion "Automatic Memory Reconfiguration" (AMR)

Aktives Filesystem

Die Funktion AMR ermöglicht das Umkonfigurieren von Speicherbereichen des aktiven Filesystems (Seite 837), ohne dass dabei, um den Verlust der Anwenderdaten zu vermeiden, ein Inbetriebnahmearchiv erstellt und anschließend wieder eingelesen werden muss.

Ist die Funktion aktiv, wird bei einer Änderung von speicherkonfigurierenden Maschinendaten die das aktive Filesystem betreffen zuerst geprüft, ob alle Daten des aktiven Filesystems lokal zwischengespeichert werden können. Ist dies der Fall, werden mit der Aktivierung der geänderten Speicherkonfiguration durch Auslösen von Warmstart die Daten des aktiven Filesystems zuerst in einen lokalen Zwischenpuffer geschrieben. Danach wird der Speicher des aktiven Filesystems umkonfiguriert. Abschließend werden die zwischengespeicherten Daten wieder in das neu konfigurierte aktive Filesystem zurückgelesen.

Können die Daten des aktiven Filesystems nicht zwischengespeichert werden, wird der Alarm 4400 angezeigt. Vor Aktivierung der Maschinendatenänderung sollte dann ein Inbetriebnahmearchiv mit den relevanten Anwenderdaten erstellt werden. Siehe unten Absatz "Weitere Informationen".

Passives Filesystem

Werden Maschinendaten geändert, die eine Umkonfiguration des passiven Filesystems erfordern, wird, unabhängig von der Funktion AMR, immer der Alarm 4400 angezeigt. Vor Aktivierung der Maschinendatenänderung sollte daraufhin ein Inbetriebnahmearchiv mit den relevanten Anwenderdaten erstellt werden.

Werden Maschinendaten geändert, die nur eine Umkonfiguration des aktiven Filesystems erfordern, bleiben die Daten des passiven Filesystems erhalten.

Aktivierung

Die Aktivierung der Funktion "Automatic Memory Reconfiguration" (AMR) erfolgt über das Maschinendatum:

```
MD17950 $MN_IS_AUTOMATIC_MEM_RECONFIG = TRUE
```

Weitere Informationen

Inbetriebnahmehandbuch CNC: NC, PLC, Antrieb; Daten sichern und verwalten

15.4 Konfiguration des statischen Anwenderspeichers

15.4.1 Aufteilung des statischen NC-Speichers

Der statische NC-Speicher wird vom System und vom Anwender gemeinsam genutzt. Der dem Anwender zur Verfügung stehende Bereich wird als statischer Anwenderspeicher bezeichnet. Er enthält die Daten des aktiven und passiven Filesystems.

Das folgende Bild zeigt die prinzipielle Aufteilung des statischen NC-Speichers:

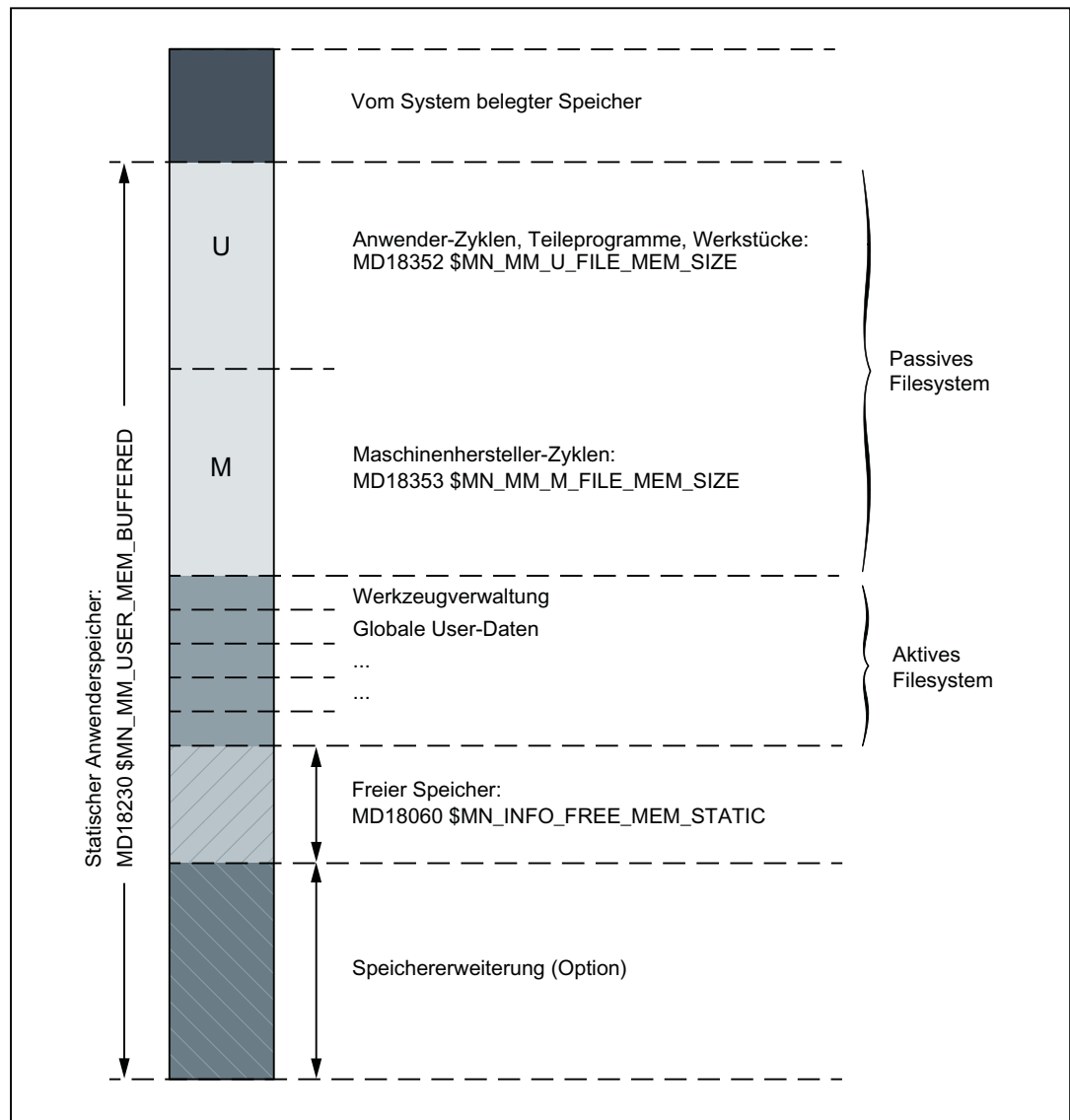


Bild 15-1 Statischer NC-Speicher

Größe des statischen Anwenderspeichers

Die Größe des statischen Anwenderspeichers wird angezeigt im Maschinendatum:

MD18230 \$MN_MM_USER_MEM_BUFFERED

Bestandteile des statischen Anwenderspeichers

Passives Filesystem

Im statischen Anwenderspeicher liegen folgende Partitionen des passiven Filesystems:

Partition	Speicherung von:
U (User = Anwender)	<ul style="list-style-type: none"> • Dateien aus dem Verzeichnis _N_CUS_DIR (Anwender-Zyklen) • Teileprogramme • Werkstücke
M (Manufacturer = Maschinenhersteller)	Dateien aus dem Verzeichnis _N_CMA_DIR (Maschinenhersteller-Zyklen)

Der im MD18060 \$MN_INFO_FREE_MEM_STATIC angezeigte freie Speicher des passiven Filesystems kann beliebig auf die Partitionen **U** und **M** aufgeteilt werden. Die Einstellung erfolgt über die Maschinendaten:

- MD18352 \$MN_MM_U_FILE_MEM_SIZE = <Speichergröße für Anwenderdaten>
- MD18353 \$MN_MM_M_FILE_MEM_SIZE = <Speichergröße für Maschinenherstellereinstellungen>

Hinweis

Partition S

Die Partition S (Siemens = Steuerungshersteller) des passiven Filesystems liegt im dynamischen Speicher (Seite 843).

Aktives Filesystem

Der Speicher des aktiven Filesystems ist in verschiedene Bereiche unterteilt (Werkzeugverwaltung, Globale User-Daten, ...). Über speicherkonfigurierende Maschinendaten (...MM...) kann für jeden Bereich die jeweilige Größe eingestellt werden.

Freier Speicher

Die Größe des freien Speichers wird angezeigt im Maschinendatum:

MD18060 \$MN_INFO_FREE_MEM_STATIC

Hinweis

Auf der Bedienoberfläche im Bedienbereich "Inbetriebnahme" wird der Speicherbedarf für die Änderung von Speicherbereichen angezeigt. Mit diesen Angaben kann der Inbetriebnehmer den tatsächlichen Speicherbedarf für die geplanten Änderungen abschätzen.

Speichererweiterung (Option)

Bei Bedarf von zusätzlichem statischen Anwenderspeicher kann eine Speichererweiterung als Option erworben werden.

Der zusätzliche Speicher kann je nach Bedarf zur Vergrößerung der Partitionen U und/oder M oder zur Vergrößerung des Speicherbereichs des aktiven Filesystems verwendet werden.

15.4.2 Inbetriebnahme

Die standardmäßige Speicheraufteilung kann durch Vergrößern/Verkleinern einzelner Speicherbereiche anwendungsspezifisch angepasst werden.

Prinzipielle Vorgehensweise:

1. Standardmaschinendaten laden.
Weitere Informationen:
 Inbetriebnahmehandbuch CNC: NC, PLC, Antrieb; Voraussetzungen für die Inbetriebnahme
2. Die maximal mögliche Größe des statischen Anwenderspeichers (inklusive der optionalen Speichererweiterung) ermitteln:
 MD18230 \$MN_MM_USER_MEM_BUFFERED
3. Optional: die Größe des statischen Anwenderspeichers verändern:
 - MD19250 \$ON_USER_MEM_BUFFERED
 - Power On-Reset ausführen
4. Optional: die Größe der Partitionen U und M einstellen:
 - MD18352 \$MN_MM_U_FILE_MEM_SIZE
 - MD18353 \$MN_MM_M_FILE_MEM_SIZE
5. Optional: zusätzliche Kanäle der Steuerung einrichten:
 MD10010 \$MC_ASSIGN_CHAN_TO_MODE_GROUP
6. Optional: die Größe der Speicherbereiche des aktiven Filesystems (Werkzeugverwaltung, Globale User-Daten, ...) einstellen:
 - Den freien statischen Anwenderspeicher ermitteln:
 MD18060 \$MN_INFO_FREE_MEM_STATIC
 - Die Größe der Speicherbereiche des aktiven Filesystems über die speicherkonfigurierenden Maschinendaten (...MM...) einstellen.
7. Power On-Reset ausführen.
 Im anschließenden Hochlauf der Steuerung wird der Speicher neu konfiguriert.

Weitere Informationen

- Listenhandbuch Maschinendaten und Parameter

15.5 Konfiguration des dynamischen Anwenderspeichers

15.5.1 Aufteilung des dynamischen NC-Speichers

Der dynamische NC-Speicher wird vom System und vom Anwender gemeinsam genutzt. Der dem Anwender zur Verfügung stehende Bereich wird als dynamischer Anwenderspeicher bezeichnet.

Das folgende Bild zeigt die prinzipielle Aufteilung des dynamischen NC-Speichers:

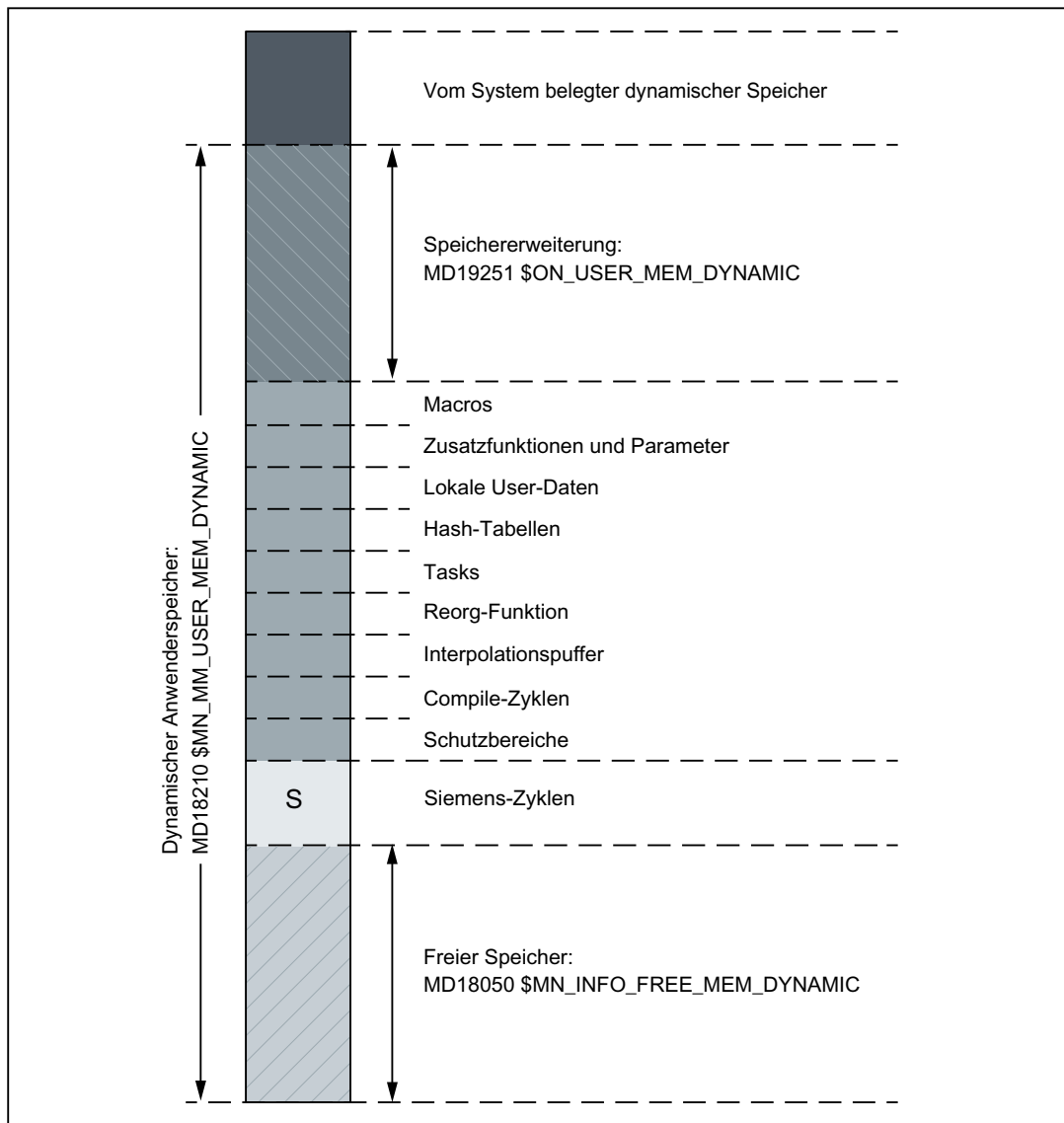


Bild 15-2 Dynamischer NC-Speicher

Größe des dynamischen Anwenderspeichers

Die Größe des dynamischen Anwenderspeichers wird angezeigt im Maschinendatum:
 MD18210 \$MN_MM_USER_MEM_DYNAMIC

Freier Anwenderspeicher

Der freie Anwenderspeicher wird angezeigt im Maschinendatum:
 MD18050 \$MN_INFO_FREE_MEM_DYNAMIC

Der Wert des Maschinendatums gibt an, wie viel Speicher für die Vergrößerung der Anwenderdatenbereiche (Lokale User-Daten, IPO-Puffer, ...) pro Kanal zur Verfügung steht.

Bestandteile des dynamischen Anwenderspeichers

Passives Filesystem

Im dynamischen Anwenderspeicher liegt die Partitionen S des **passiven** Filesystems:

Partition	Speicherung von:
S (Siemens = Steuerungshersteller)	Dateien aus dem Verzeichnis _N_CST_DIR (Siemens-Zyklen)

Die Größen der Partition **S** ist voreingestellt und nicht veränderbar.

Speichererweiterung

Beim Einsatz einer NCU mit mindestens 2 GB Arbeitsspeicher besteht die Möglichkeit, einen Teil des Arbeitsspeichers zur Erweiterung des dynamischen Anwenderspeichers zu nutzen.

Die Erweiterung des zusätzlichen dynamischen Anwenderspeichers wird in 4 MB-Schritten eingestellt im Maschinendatum:

OD19240 \$ON_USER_MEM_DYNAMIC

Gesamtgröße des dynamischen Anwenderspeicher

"Gesamtgröße des dynamischen Anwenderspeichers" = Grundausbau + OD19240
\$ON_USER_MEM_DYNAMIC * 4MB

Anzeige

Der aktuell zur Verfügung stehende dynamische Anwenderspeicher wird angezeigt im Maschinendatum:

MD18210 \$MN_MM_USER_MEM_DYNAMIC

15.5.2 Inbetriebnahme

Die standardmäßige Speicheraufteilung kann durch Vergrößern/Verkleinern einzelner Speicherbereiche anwendungsspezifisch angepasst werden.

Prinzipielle Vorgehensweise:

1. Die Größe des freien dynamischen Anwenderspeicher ermitteln:
MD18050 \$MN_INFO_FREE_MEM_DYNAMIC
2. Optional: den dynamischen Anwenderspeicher vergrößern:
 - MD18210 \$MN_MM_USER_MEM_DYNAMIC
 - Power On-Reset ausführen
3. Optional: die Größe der Speicherbereiche des dynamischen Anwenderspeichers über die speicherkonfigurierenden Maschinendaten (...MM_...) einstellen:
4. Power On-Reset ausführen.
Im anschließenden Hochlauf der Steuerung wird der Speicher neu konfiguriert.

Weitere Informationen:

- Listenhandbuch Maschinendaten und Parameter

15.6 Randbedingungen

15.6.1 Abhängigkeit von Kanal- und Werkzeugträgeranzahl

Wird die Anzahl der Kanäle erhöht, muss eventuell die Anzahl der Werkzeugträger ebenfalls erhöht werden.

Die Anzahl der in der Steuerung verfügbaren Werkzeugträger ist nach folgender Regel einzustellen:

$$\text{MD18088 \$MN_NUM_TOOL_CARRIER} = \\ \text{<Größte Anzahl von Werkzeugträgern in einem Kanal> *} \\ \text{<Anzahl parametrierter Kanäle>}$$

Die Werkzeugträger werden steuerungsintern gleichmäßig auf die vorhandenen Kanäle aufgeteilt.

Beispiel

In der NC sind 3 Kanäle parametrierter. Die Anzahl von benötigten Werkzeugträgern in den Kanälen ist:

1. Kanal: 3 \Rightarrow Maximalwert
2. Kanal: 2
3. Kanal: 1

$$\text{MD18088} = \text{<Maximalwert von Werkzeugträgern in einem Kanal> * <Kanalanzahl>} = 3 * 3 = 9$$

Steuerungsintern werden somit jedem Kanal 3 Werkzeugträger zugeordnet.

15.6.2 Nachträgliche Verringerung der Kanal- und/oder Achszahl

Eine einmal vorgenommene und in der Steuerung durch Warmstart wirksam gemachte Vergrößerung der Kanal- und/oder Achszahl darf anschließend nicht mehr rückgängig gemacht werden, wenn von dieser Steuerung ein Inbetriebnahmearchiv erstellt werden soll.

Wird ein derartiges Inbetriebnahmearchiv in eine Steuerung geladen, wird der Vorgang mit einem Alarm abgebrochen.

Unterschiede bezüglich des dynamischen und statischen Speichers der NC

- **Dynamischer Speicher:** Ein erhöhter interner Speicherbedarf entsteht bei Erhöhung sowohl der **Kanal-** als auch der **Achszahl**.
- **Statischer Speicher:** Ein erhöhter interner Speicherbedarf entsteht nur bei Erhöhung der **Kanalanzahl**.

15.7 Datenlisten

15.7.1 Maschinendaten

15.7.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10134	MM_NUM_MMC_UNITS	Anzahl gleichzeitig möglicher HMI-Kommunikationspartner
10850	MM_EXTERN_MAXNUM_OEM_GCODES	Maximale Anzahl der OEM-G-Befehle
10880	MM_EXTERN_CNC_SYSTEM	Definition des zu adaptierenden Steuerungssystems
10881	MM_EXTERN_GCODE_SYSTEM	ISO_3 Mode: GCodeSystem
18050	INFO_FREE_MEM_DYNAMIC	Anzeigedatum des freien dynamischen Speichers
18060	INFO_FREE_MEM_STATIC	Anzeigedatum des freien statischen Speichers
18070	INFO_FREE_MEM_DPR	Anzeigedatum des freien Speichers im Dualport-RAM
18072	INFO_FREE_MEM_CC_MD	Anzeige freier Speicher CC-MD-Speicher
18078	MM_MAX_NUM_OF_HIERARCHIES	Maximale Anzahl definierbarer Hierarchien für Magazinplatztypen
18079	MM_MAX_HIERARCHY_ENTRIES	Max. erlaubte Anzahl von Einträgen in einer Mag.pl.typ-Hierarch.
18080	MM_TOOL_MANAGEMENT_MASK	Maske für Speicherreservierung der WZV
18082	MM_NUM_TOOL	Anzahl der von NC verwalteten Werkzeuge
18084	MM_NUM_MAGAZINE	Anzahl der von der NC verwalteten Magazine
18086	MM_NUM_MAGAZINE_LOCATION	Anzahl der Magazinplätze
18088	MM_NUM_TOOL_CARRIER	Maximale Anzahl definierbarer Werkzeugträger
18090	MM_NUM_CC_MAGAZINE_PARAM	Compile-Zyklen-WZV: Anzahl der Magazin-Daten
18092	MM_NUM_CC_MAGLOC_PARAM	Compile-Zyklen-WZV: Anzahl der Magazinplatz-Daten
18094	MM_NUM_CC_TDA_PARAM	Compile-Zyklen-WZV: Anzahl der TDA-Daten
18096	MM_NUM_CC_TOA_PARAM	Compile-Zyklen-WZV: Anzahl der TOA-Daten
18098	MM_NUM_CC_MON_PARAM	Compile-Zyklen-WZV: Anzahl der Monitor-Daten
18100	MM_NUM_CUTTING_EDGES_IN_TOA	Anzahl der Werkzeugkorrekturen in NC
18102	MM_TYPE_OF_CUTTING_EDGE	Art der D-Nummer-Programmierung
18104	MM_NUM_TOOL_ADAPTER	WZ-Adapter im TO-Bereich
18105	MM_MAX_CUTTING_EDGE_NO	Maximaler Wert der D-Nummer
18106	MM_MAX_CUTTING_EDGE_PERTOOL	Maximale Anzahl der D-Nummern pro Werkzeug
18108	MM_NUM_SUMCORR	Summenkorrekturen im TO-Bereich
18110	MM_MAX_SUMCORR_PER_CUTTEDGE	Maximale Anzahl der Summenkorrekturen pro Schneide

Nummer	Bezeichner: \$MN_	Beschreibung
18112	MM_KIND_OF_SUMCORR	Eigenschaften der Summenkorrekturen im TO-Bereich
18114	MM_ENABLE_TOOL_ORIENT	Werkzeugschneiden-Orientierung zuordnen
18116	MM_NUM_TOOL_ENV	Anzahl der Werkzeugumgebungen im TO-Bereich
18118	MM_NUM_GUD_MODULES	Anzahl der GUD-Bausteine
18120	MM_NUM_GUD_NAMES_NCK	Anzahl der globalen Anwendervariablen
18130	MM_NUM_GUD_NAMES_CHAN	Anzahl der kanalspezifischen Anwendervariablen
18150	MM_GUD_VALUES_MEM	Speicherplatz für globale Anwendervariablen
18160	MM_NUM_USER_MACROS	Anzahl von Makros
18170	MM_NUM_MAX_FUNC_NAMES	Anzahl von Zusatzfunktionen
18180	MM_NUM_MAX_FUNC_PARAM	Anzahl von zusätzlichen Parametern
18190	MM_NUM_PROTECT_AREA_NCK	Anzahl der Schutzbereiche in NC
18200	MM_NUM_CCS_MAGAZINE_PARAM	Anzahl der Siemens-OEM-Magazindaten
18201	MM_TYPE_CCS_MAGAZINE_PARAM	Typ der Siemens-OEM-Magazindaten
18202	MM_NUM_CCS_MAGLOC_PARAM	Anzahl der Siemens-OEM-Magazinplatzdaten
18203	MM_TYPE_CCS_MAGLOC_PARAM	Typ der Siemens-OEM-Magazinplatzdaten
18204	MM_NUM_CCS_TDA_PARAM	Anzahl der Siemens-OEM-Werkzeugdaten
18205	MM_TYPE_CCS_TDA_PARAM	Typ der Siemens-OEM-Werkzeugdaten
18206	MM_NUM_CCS_TOA_PARAM	Anzahl der Siemens-OEM-Daten je Schneide
18207	MM_TYPE_CCS_TOA_PARAM	Typ der Siemens-OEM-Daten je Schneide
18208	MM_NUM_CCS_MON_PARAM	Anzahl der Siemens-OEM-Monitordaten
18209	MM_TYPE_CCS_MON_PARAM	Typ der Siemens-OEM-Monitordaten
18210	MM_USER_MEM_DYNAMIC	Anwenderspeicher für nicht persistente Daten
18220	MM_USER_MEM_DPR	Anwenderspeicher im Dualport-RAM
18230	MM_USER_MEM_BUFFERED	Anwenderspeicher für persistente Daten
18231	MM_USER_MEM_BUFFERED_TYPEOF	Technologie für die Datenpufferung
18232	MM_ACTFILESYS_LOG_FILE_MEM	System: Logdateigröße
18238	MM_CC_MD_MEM_SIZE	Speichergröße für Compilezyklen-Maschinendaten
18240	MM_LUD_HASH_TABLE_SIZE	Hash-Tabellengröße für Anwendervariablen
18242	MM_MAX_SIZE_OF_LUD_VALUE	Maximale Feldgröße der LUD-Variablen
18250	MM_CHAN_HASH_TABLE_SIZE	Hash-Tabellengröße für kanalspezifische Daten
18260	MM_NCK_HASH_TABLE_SIZE	Hash-Tabellengröße für globale Daten
18270	MM_NUM_SUBDIR_PER_DIR	Anzahl von Unterverzeichnissen
18280	MM_NUM_FILES_PER_DIR	Anzahl von Dateien pro Verzeichnis
18290	MM_FILE_HASH_TABLE_SIZE	Hash-Tabellengröße für Dateien eines Verzeichnisses
18300	MM_DIR_HASH_TABLE_SIZE	Hash-Tabellengröße für Unterverzeichnisse
18310	MM_NUM_DIR_IN_FILESYSTEM	Anzahl von Verzeichnissen im passiven Filesystem
18320	MM_NUM_FILES_IN_FILESYSTEM	Anzahl von Dateien im passiven Filesystem
18332	MM_FLASH_FILE_SYSTEM_SIZE	Größe des Flash-File-Systems auf der PCNC
18342	MM_CEC_MAX_POINTS	Maximale Tabellengröße für Durchhangkompensation
18350	MM_USER_FILE_MEM_MINIMUM	Minimaler Teileprogrammspeicher

Nummer	Bezeichner: \$MN_	Beschreibung
18352	MM_U_FILE_MEM_SIZE	Endanwenderspeicher für Teileprogramme/Zyklen/Dateien
18353	MM_M_FILE_MEM_SIZE	Speichergröße für Zyklen/Dateien des Maschinenherstellers
18354	MM_S_FILE_MEM_SIZE	Speichergröße für Zyklen/Dateien des NC-Herstellers
18355	MM_T_FILE_MEM_SIZE	Speichergröße für temporäre Dateien
18356	MM_E_FILE_MEM_SIZE	Speichergröße für externe Dateien
18360	MM_EXT_PROG_BUFFER_SIZE	FIFO-Buffer Größe für Abarbeiten von Extern
18362	MM_EXT_PROG_NUM	Anzahl der gleichzeitig von Extern abarbeitbaren Programmebenen
18370	MM_PROTOC_NUM_FILES	Maximale Anzahl von Protokoll-Files
18371	MM_PROTOC_NUM_ETPD_STD_LIST	Anzahl von Standard-Datenlisten ETPD
18372	MM_PROTOC_NUM_ETPD_OEM_LIST	Anzahl von OEM-Datenlisten ETPD
18373	MM_PROTOC_NUM_SERVO_DATA	Anzahl von Servo-Daten für Protokoll
18374	MM_PROTOC_FILE_BUFFER_SIZE	Größe des Puffers eines Protokollfiles
18375	MM_PROTOC_SESS_ENAB_USER	Für Sessions freigegebene User
18390	MM_COM_COMPRESS_METHOD	Unterstützte Komprimierverfahren
18400	MM_NUM_CURVE_TABS	Anzahl der Kurventabellen (persistent)
18402	MM_NUM_CURVE_SEGMENTS	Anzahl der Kurvensegmente (persistent)
18403	MM_NUM_CURVE_SEG_LIN	Anzahl der linearen Kurvensegmente (persistent)
18404	MM_NUM_CURVE_POLYNOMS	Anzahl der Kurventabellenpolynome (persistent)
18406	MM_NUM_CURVE_TABS_DRAM	Anzahl der Kurventabellen (nicht persistent)
18408	MM_NUM_CURVE_SEGMENTS_DRAM	Anzahl der Kurvensegmente (nicht persistent)
18409	MM_NUM_CURVE_SEG_LIN_DRAM	Anzahl der linearen Kurvensegmente (nicht persistent)
18410	MM_NUM_CURVE_POLYNOMS_DRAM	Anzahl der Kurventabellenpolynome (nicht persistent)
18450	MM_NUM_CP_MODULES	Maximale Anzahl der CP-Module
18452	MM_NUM_CP_MODUL_LEAD	Maximale Anzahl der Leitwerte pro CP-Koppelmodul
18500	MM_EXTCOM_TASK_STACK_SIZE	Stackgröße für externe Kommunikationstask (nicht persistent)
18502	MM_COM_TASK_STACK_SIZE	Stackgröße in kByte für Kommunikationstask (nicht persistent)
18510	MM_SERVO_TASK_STACK_SIZE	Stackgröße der Servotask (nicht persistent)
18512	MM_IPO_TASK_STACK_SIZE	Stackgröße der Ipo-Task (nicht persistent)
18520	MM_DRIVE_TASK_STACK_SIZE	Stackgröße der Antriebtask (nicht persistent)
18540	MM_PLC_TASK_STACK_SIZE	Stackgröße der PLC-Task (nicht persistent)
18600	MM_FRAME_FINE_TRANS	Feinverschiebung bei FRAME (persistent)
18601	MM_NUM_GLOBAL_USER_FRAMES	Anzahl der globalen vordefinierten Anwender-Frames (persistent)
18602	MM_NUM_GLOBAL_BASE_FRAMES	Anzahl der globalen Basisframes (persistent)
18660	MM_NUM_SYNACT_GUD_REAL	Anzahl der projektierbaren GUD-Variablen vom Typ Real
18661	MM_NUM_SYNACT_GUD_INT	Anzahl der projektierbaren GUD-Variablen vom Typ Integer

Nummer	Bezeichner: \$MN_	Beschreibung
18662	MM_NUM_SYNACT_GUD_BOOL	Anzahl der projektierbare GUD-Variablen vom Typ Boolean
18663	MM_NUM_SYNACT_GUD_AXIS	Anzahl der projektierbaren GUD-Variablen vom Typ Axis
18664	MM_NUM_SYNACT_GUD_CHAR	Projektierbare GUD-Variable Typ Char
18665	MM_NUM_SYNACT_GUD_STRING	Projektierbare GUD-Variable Typ STRING
18700	MM_SIZEOF_LINKVAR_DATA	Größe des NCU-Link-Variablen-Speichers
18710	MM_NUM_AN_TIMER	Anzahl der globalen Zeitvariablen für Synchronaktionen
18720	MM_SERVO_FIFO_SIZE	Sollwert für Puffergröße zwischen IPO und Lageregelung
18780	MM_NCU_LINK_MASK	Aktivierung der NCU-Link-Kommunikation
18781	NCU_LINK_CONNECTIONS	Anzahl interner Linkverbindungen
18782	MM_LINK_NUM_OF_MODULES	Anzahl der NCU-Link-Module
18790	MM_MAX_TRACE_LINK_POINTS	Größe des Trace-Datenbuffers für NCU-Link
18792	MM_TRACE_LINK_DATA_FUNCTION	Spezifiziert die Inhalte des NCU-Link-Files
18794	MM_TRACE_VDI_SIGNAL	Trace-Spezifikation der Vdi-Signale
18800	MM_EXTERN_LANGUAGE	Aktivierung externer NC-Sprachen
18860	MM_MAINTENANCE_MON	Aktivierung der Aufzeichnung von Wartungsdaten
18870	MM_MAXNUM_KIN_CHAINS	Maximale Anzahl kinematischer Ketten
18880	MM_MAXNUM_KIN_CHAIN_ELEM	Maximale Anzahl der Elemente kinematischer Ketten
18890	MM_MAXNUM_3D_PROT_AREAS	Maximale Anzahl der Elemente in 3D-Schutzbereichen
18892	MM_MAXNUM_3D_PROT_AREA_ELEM	Maximale Anzahl der Schutzbereichselemente
18894	MM_MAXNUM_3D_PROT_GROUPS	Maximale Anzahl Schutzbereichsgruppen
18896	MM_MAXNUM_3D_COLLISION	Maximale Anzahl temporärer Speicherplätze für Kollisionscheck

15.7.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20096	T_M_ADDRESS_EXIT_SPINO	Spindelnummer als Adresserweiterung
27900	REORG_LOG_LIMIT	Prozentsatz des IPO-Puffers für Freigabe des Logfiles
28000	MM_REORG_LOG_FILE_MEM	Speichergröße für REORG
28010	MM_NUM_REORG_LUD_MODULES	Anzahl der Bausteine für lokale Anwender-variablen bei REORG
28020	MM_NUM_LUD_NAMES_TOTAL	Anzahl der lokalen Anwendervariablen
28040	MM_LUD_VALUES_MEM	Speichergröße für lokale Anwendervariablen
28050	MM_NUM_R_PARAM	Anzahl der Kanal-spezifischen R-Parameter
28060	MM_IPO_BUFFER_SIZE	Anzahl der NC-Sätze im IPO-Puffer
28070	MM_NUM_BLOCKS_IN_PREP	Anzahl der Sätze für die Satzaufbereitung
28080	MM_NUM_USER_FRAMES	Anzahl der einstellbaren Frames
28081	MM_NUM_BASE_FRAMES	Anzahl Basisframes (persistent)
28082	MM_SYSTEM_FRAME_MASK	Systemframes (persistent)

Nummer	Bezeichner: \$MC_	Beschreibung
28083	MM_SYSTEM_DATAFRAME_MASK	Systemframes (persistent)
28085	MM_LINK_TOA_UNIT	Zuordnung einer TO-Einheit zu einem Kanal
28090	MM_NUM_CC_BLOCK_ELEMENTS	Anzahl Satzelemente für Compile-Zyklen
28100	MM_NUM_CC_BLOCK_USER_MEM	Größe des Satzspeichers für Compile-Zyklen
28105	MM_NUM_CC_HEAP_MEM	Heap-Speicher in kByte für Compile-Zyklen Applikationen (nicht persistent)
28150	MM_NUM_VDIVAR_ELEMENTS	Anzahl Elemente für das Schreiben von PLC-Variablen
28160	MM_NUM_LINKVAR_ELEMENTS	Anzahl Elemente zum Schreiben der NCU-Link-Variablen
28180	MM_MAX_TRACE_DATAPOINTS	Größe des Trace-Datenpuffers
28200	MM_NUM_PROTECT_AREA_CHAN	Anzahl der Bausteine für Kanal-spezifische Schutzbereiche
28210	MM_NUM_PROTECT_AREA_ACTIVE	Anzahl der gleichzeitig aktiven Schutzbereiche
28212	MM_NUM_PROTECT_AREA_CONTOUR	Elemente für aktive Schutzbereiche (nicht persistent)
28250	MM_NUM_SYNC_ELEMENTS	Anzahl Elemente für Ausdrücke in Synchronaktionen
28252	MM_NUM_FCTDEF_ELEMENTS	Anzahl der FCTDEF-Elemente
28254	MM_NUM_AC_PARAM	Dimension von \$AC_PARAM.
28255	MM_BUFFERED_AC_PARAM	\$AC_PARAM[...] (persistent)
28256	MM_NUM_AC_MARKER	Dimension von \$AC_MARKER
28257	MM_BUFFERED_AC_MARKER	\$AC_MARKER[...] (persistent)
28258	MM_NUM_AC_TIMER	Anzahl Zeitvariablen \$AC_TIMER (nicht persistent)
28274	MM_NUM_AC_SYSTEM_PARAM	Anzahl \$AC_SYSTEM_PARAM für Bewegungssynchronaktionen
28276	MM_NUM_AC_SYSTEM_MARKER	Anzahl \$AC_SYSTEM_MARKER für Bewegungssynchronaktionen
28290	MM_SHAPED_TOOLS_ENABLE	Werkzeugradiuskorrektur für Konturwerkzeuge freigeben
28300	MM_PROTOC_USER_ACTIVE	Aktivierung der Protokollierung für einen User
28301	MM_PROTOC_NUM_ETP_OEM_TYP	Anzahl von OEM-Event-Typen ETP
28302	MM_PROTOC_NUM_ETP_STD_TYP	Anzahl von Standard-Event-Typen ETP
28400	MM_ABSBLOCK	Satzanzeige mit Absolutwerten aktivieren
28402	MM_ABSBLOCK_BUFFER_CONF	Größe des Upload-Buffers dimensionieren
28450	MM_TOOL_DATA_CHG_BUFF_SIZE	Puffer für Werkzeugdaten-Änderung (nicht persistent)
28500	MM_PREP_TASK_STACK_SIZE	Stack-Größe der Präparation-Task
28520	MM_MAX_AXISPOLY_PER_BLOCK	Maximale Anzahl der Achspolynome pro Satz
28530	MM_PATH_VELO_SEGMENTS	Anzahl Speicherelemente zur Begrenzung der Bahngeschwindigkeit
28535	MM_FEED_PROFILE_SEGMENTS	Anzahl der Speicherelemente für Vorschubprofile
28540	MM_ARCLENGTH_SEGMENTS	Anzahl Speicherelementen zur Darstellung der Bogenlängenfunktion
28560	MM_SEARCH_RUN_RESTORE_MODE	Restore von Daten nach einer Simulation
28580	MM_ORIPATH_CONFIG	Einstellung für bahnrelative Orientierung ORIPATH

15.7.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
38000	MM_ENC_COMP_MAX_POINTS	Anzahl der Stützpunkte bei interpolatorischer Kompensation
38010	MM_QEC_MAX_POINTS	Anzahl der Werte für Quadrantenfehler-Kompensation
38020	MM_CPREC_FIR_POINTS	Anzahl Punkte für Kennlinienapproximation von FIR-Filtern für CPRECON

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
CEC	Cross Error Compensation
CI	Connector Input
CF-Card	Compact Flash-Card

C	
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung
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/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

Anhang

A.1 Liste der Abkürzungen

G	
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	Hauptspindelantrieb
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

L	
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 mit Satzaufbereitung, Verfahrbereich usw.
NCU	Numerical Control Unit: Hardware-Einheit des NC
NRK	Bezeichnung des Betriebssystems des NC
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
PCMCIA	Personal Computer Memory Card International Association: Speichersteckkarten-Normierung
PCU	PC Unit: PC-Box (Rechereinheit)
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

R	
ROV	Rapid Override: Eingangskorrektur
RP	R-Parameter, Rechenparameter, vordefinierte Anwendervariable
RPA	R-Parameter Active: Speicherbereich in NC für R-Parameternummern
RPY	Roll Pitch Yaw: Drehungsart eines Koordinatensystems
RTL	Rapid Traverse Linear Interpolation: Lineare Interpolation bei Eilgangbewegung
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: Satzsuchlauf 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

S	
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 NC 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

W	
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 Verfügbare IPCs

Für die SINUMERIK empfohlene IPCs

Panel-IPC	
IPC 477E 22" Win 7	6AV7241-3YA04-0FA0
IPC 477E 24" Win 7	6AV7241-5SB04-0FA0
IPC 477E 15" Win10	6AV7241-1WA07-0FA0
IPC 477E 19" Win10	6AV7241-3XB07-0FA0
IPC 477E 22" Win10	6AV7241-3YA07-0FA0
IPC 477E 24" Win10	6AV7241-5SB07-0FA0

Box-IPC	
IPC 427E (Standard) Win7	6AG4141-1AA14-0FA0
IPC 427E (High) Win7	6AG4141-5AB14-0FA0
IPC 427E (Standard) Win10	6AG4141-1AA17-0FA0
IPC 427E (High) Win10	6AG4141-5AB17-0FA0

Index

\$

\$A_DP_IN_CONF, 755
\$A_DP_IN_STATE, 756
\$A_DP_IN_VALID, 756
\$A_DP_OUT_CONF, 755
\$A_DP_OUT_STATE, 756
\$A_DP_OUT_VALID, 756
\$A_IN, 728
\$A_INA, 728, 735
\$A_INCO, 742
\$A_OUT, 728, 730
\$A_OUTA, 728, 737
\$AA_ATOL, 514
\$AA_G0MODE, 523
\$AA_ISTEST, 547
\$AC_ACT_PROG_NET_TIME, 244
\$AC_ACTUAL_PARTS, 250, 254
\$AC_ASUP, 168
\$AC_AUXFU_EXT, 711
\$AC_AUXFU_M_EXT, 711
\$AC_AUXFU_M_STATE, 712
\$AC_AUXFU_M_TICK, 692
\$AC_AUXFU_M_VALUE, 711
\$AC_AUXFU_PREDEF_INDEX, 697, 711
\$AC_AUXFU_SPEC, 708, 711
\$AC_AUXFU_STATE, 712
\$AC_AUXFU_TYPE, 711
\$AC_AUXFU_VALUE, 711
\$AC_AXCTSWA, 805
\$AC_AXCTSWE, 805
\$AC_CTOL, 514
\$AC_CTOL_G0_ABS, 526
\$AC_CUTTING_TIME, 246
\$AC_CYCLE_TIME, 246
\$AC_DELAYFST, 143
\$AC_ISTEST, 547
\$AC_OLD_PROG_NET_TIME, 244
\$AC_OLD_PROG_NET_TIME_COUNT, 245
\$AC_OPERATING_TIME, 246
\$AC_OTOL, 514
\$AC_OTOL_G0_ABS, 526
\$AC_PROG_NET_TIME_TRIGGER, 245
\$AC_PROGINF, 180
\$AC_REQUIRED_PARTS, 250, 254
\$AC_SPECIAL_PARTS, 250, 254
\$AC_STOLF, 525
\$AC_TOTAL_PARTS, 250, 254
\$AN_AUXFU_LIST_ENDINDEX, 701
\$AN_AXCTAS, 805
\$AN_AXCTSWA, 805
\$AN_LAI_AX_IS_AXCTAX, 805
\$AN_LAI_AX_IS_LEADLINKAX, 805
\$AN_LAI_AX_IS_LINKAX, 805
\$AN_LAI_AX_TO_IPO_NC_CHANAX, 805
\$AN_LAI_AX_TO_MACHAX, 805
\$AN_POWERON_TIME, 243
\$AN_SETUP_TIME, 243
\$C_AUX_EXT, 224
\$C_AUX_IS_QUICK, 224
\$C_AUX_VALUE, 224
\$C_D, 225
\$C_D_PROG, 225
\$C_DL, 225
\$C_DL_PROG, 225
\$C_DUPLO, 225
\$C_DUPLO_PROG, 225
\$C_M, 224
\$C_M_PROG, 224
\$C_ME, 224
\$C_MTL, 221, 225
\$C_MTL_PROG, 221, 225
\$C_T, 224
\$C_T_PROG, 224
\$C_TCA, 225
\$C_TE, 225
\$C_THNO, 225
\$C_THNO_PROG, 225
\$C_TS, 225
\$C_TS_PROG, 225
\$NK_A_OFF, 429, 433, 435
\$NK_AXIS, 428, 432
\$NK_NAME, 423
\$NK_NEXT, 424
\$NK_OFF_DIR, 427, 430, 433, 435
\$NK_PARALLEL, 425
\$NK_SWITCH, 439
\$NK_SWITCH_INDEX, 437
\$NK_SWITCH_POS, 438
\$NK_TYPE, 426
\$P_CHANNO, 135
\$P_CTOL, 515
\$P_CTOL_G0_ABS, 526
\$P_DELAYFST, 143
\$P_DRYRUN, 180
\$P_GFRNUM, 338
\$P_IFRAME, 334

\$P_ISDRF, 180
 \$P_ISPROGSTOP, 180
 \$P_ISRG0, 180
 \$P_ISSKIP, 180
 \$P_ISTEST, 49, 180, 547
 \$P_OTOL, 515
 \$P_OTOL_G0_ABS, 526
 \$P_PROG_EVENT, 135
 \$P_REPINF, 162
 \$P_SEARCH_S, 62, 694
 \$P_SEARCH_SDIR, 62, 694
 \$P_SEARCH_SGEAR, 62, 694
 \$P_SEARCH_SMODE, 62
 \$P_SEARCH_SPOS, 62, 694
 \$P_SEARCH_SPOSMODE, 62, 694
 \$P_SIM, 55
 \$P_STOLF, 525
 \$P_SUB_AUTOGEAR, 234
 \$P_SUB_AXFCT, 233, 234
 \$P_SUB_CA, 234
 \$P_SUB_GEAR, 233
 \$P_SUB_LA, 234
 \$P_SUB_M19, 234
 \$P_SUB_SPOS, 234
 \$P_SUB_SPOSA, 234
 \$P_SUB_SPOSIT, 234
 \$P_SUB_SPOSMODE, 234
 \$P_SUB_STAT, 223, 225
 \$P_UIFR, 334
 \$P_UIFRNUM, 334
 \$PA_ATOL, 515

—

_N_STRTLK, 45
 _N_STRTUL, 46

A

Achsbezeichner, 584
 Achscontainer
 -Bezeichner, 796
 Achskonfiguration, 275
 Achstausch, 617
 Achs-Containerdrehung freigeben, 561
 Achstausch über Synchronaktionen, 567
 Geometrieachse im gedrehten WKS, 566
 ohne Vorlaufstopp, 562
 ADDFRAME, 379
 Aktionssätze, 59
 Aktives Filesystem, 838

Aktivierung eines Achstauschs ohne
 Vorlaufstopp, 562
 Aktivierung/Deaktivierung, 579
 ALF, 164
 Allgemeine Funktionalität, 577
 Allgemeines, 577
 Anpassfaktor
 der Bahndynamik, 489
 Anzeigesatz, Aufbau (DIN), 149
 Arbeitsebene, 597
 ASUP
 Aktivierung, 154
 bei Anwenderalarmen, 182
 interne, 167
 Priorität, 164
 Reorganisation, 155
 SERUPRO-Ende, 702
 ASUP-Freigabe, 602
 ATOL, 511
 ATRANS, 265
 Aufruf, 583
 Aufrufbedingung, 583
 Ausblendeebenen, 180
 Ausführliche Beschreibung, 577
 Ausgabe
 auf externes Gerät/Datei, 206
 -sequenz, 698
 -verhalten einer Hilfsfunktion, 670
 -zähler, 698
 Autarke Einzelachsvorgänge, 101
 automatischer Achstausch, 557
 AUXFUDEL, 703
 AUXFUDELG, 703
 AUXFUMSEQ, 692
 AUXFUSYNC, 702
 AXCTSWE, 803
 AXCTSWEC, 803
 AXCTSWED, 803
 AXTOCHAN, 567

B

Bahnachsen, 271
 Bahnsteuerbetrieb, 458
 Impliziter, 461
 Basis-Anzeige
 Größe des Anzeigebuffers, 147
 Basiskoordinatensystem (BKS), 263, 295
 Basis-Satzanzeige
 aktivieren, 147
 konfigurieren, 146
 Bearbeitungszeit, 246

- Benutzerdefinierten ASUP
nach SERUPRO-Vorgang, 72
- Besonderheiten im Zielsatz
STOPPRE-Satz, 94
- Betriebsart
AUTOMATIC, 35
JOG, 36
JOG in AUTOMATIC, 35
MDA, 36
- Betriebsarten
der Betriebsartengruppe, 35
Prioritäten, 36
-übergreifende Synchronaktionen, 36
-Überwachungen, 40
-Verriegelungen, 40
-wechsel, 35, 40
- Betriebsartengruppe (BAG), 32
- Betriebsartenwechsel
von/nach Betriebsarten AUTOMATIK, JOG,
MDA, 41
- Betriebszustände, 37
- BLSYNC, 164
- C**
- CC_PREPRE, 591, 603
CC_RESU.MPF, 605
CC_RESU_ASUP.SPF, 609
CC_RESU_BS_ASUP.SPF, 609
CC_RESU_END.SPF, 608
CC_RESU_INI.SPF, 606
CFINE, 265
CLC_RESU_LENGTH_BS_BUFFER, 614
CLEARARM, 531
CLRINT, 165
COMPCAD, 504, 507
COMPCURV, 503, 507
COMPOF, 507
COMPON, 503, 507
COMPSURF, 504, 507
CORROF, 304
CT, 803
CTOL, 511
CTRANS, 265
- D**
- D/DL-Funktionsersetzung, 221
- Datenkanal, schneller, 628
- DB10
DBB4-7, 624
- DBX106.1, 623, 625
DBX56.1, 623, 624, 626
DBX56.2, 623, 624
DBX56.4, 635
DBX56.5, 635
DBX56.6, 635
DBX56.7, 635
- DB11
DBX0.0, 36, 38
DBX0.1, 36
DBX0.2, 36
DBX0.4, 41
DBX0.5, 34
DBX0.6, 34
DBX0.7, 34, 114, 624
DBX1.0 - DBX1.2, 37
DBX26.4, 39
DBX26.5, 39
DBX4.0 - DBX4.2, 36
DBX46.4, 39
DBX46.5, 39
DBX5.0 - DBX5.2, 37
DBX6.0, 37, 39
DBX6.1, 37
DBX6.2, 37
DBX6.3, 35, 623, 625
DBX6.4, 39
DBX6.5, 39
DBX6.7, 35
DBX7.0, 39
DBX7.0 - DBX7.2, 37
- DB11, ...
DBX0.7, 602
- DB21
DBX24.4, 679
DBX378.1, 162
- DB21, ...
D35.0, 112
D35.5, 112
DBB116 - DBB136, 686
DBB140 - DBB190, 686
DBB194 - DBB206, 686
DBB376, 130
DBB58 - DBB67, 686
DBB68 - DBB112, 686
DBX0.4, 50
DBX0.6, 53
DBX1.6, 58, 61
DBX2.0, 54, 112, 139, 140
DBX2.0 - 7, 181
DBX24.6, 53
DBX26.0, 54

DBX30.5, 679
 DBX31.0 - 2, 78
 DBX31.0 - DBX31.2, 79
 DBX31.0-31.2, 84
 DBX31.4, 77, 79, 80
 DBX31.6 - 7, 181
 DBX318.0, 156
 DBX318.5, 679
 DBX319.0, 80
 DBX319.1 - DBX319.3, 79, 80, 81
 DBX319.5, 80, 82
 DBX32.3, 58
 DBX32.4, 58, 59
 DBX32.6, 58, 695
 DBX33.4, 58
 DBX35.0, 111, 115
 DBX35.0 - DBX35.4, 162
 DBX35.1, 115
 DBX35.2, 115
 DBX35.3, 51, 115
 DBX35.4, 115
 DBX35.5, 111, 116
 DBX35.5 - DBX35.7, 162
 DBX35.6, 116
 DBX35.7, 114, 116
 DBX36.7, 61
 DBX384.0, 119
 DBX7.1, 49, 113
 DBX7.2, 112, 139
 DBX7.3, 112, 139
 DBX7.4, 112, 139
 DBX7.5, 45
 DBX7.7, 114, 625
 DB21, ... DB32.6, 61
 DB21, ... DBX0.4, 59, 161
 DB21, ... DBX1.7, 47
 DB21, ... DBX25.7, 47
 DB21, ... DBX33.7, 47
 DB21, ... DBX36.6, 61
 DB21, ... DBX6.1, 160
 DB21, ...
 DBX0.1, 593, 602, 605, 606, 609
 DBX0.2, 594, 602, 610
 DBX318.0, 612
 DBX32.1, 610
 DBX32.2, 602
 DBX32.6, 594
 DBX33.4, 612
 DBX378.0, 155
 DBX7.7, 602
 DB21, ...
 DBX317.1, 250, 254

DB31, ...
 DBX10.0, 75, 78, 79, 80, 84
 DBX2.2, 76
 DBX28.7, 44
 DBX3.7, 101
 DBX60.5, 625
 DBX64.6, 690
 DBX64.7, 690
 DBX69.0, 643
 DBX69.1, 643
 DBX69.2, 643
 DBX70.0, 80, 82
 DBX70.1, 80, 81, 82
 DBX70.2, 80, 81
 DBX72.0, 79
 DBX76.4, 80, 83
 DBX9.0, 642
 DBX9.1, 642
 DBX9.2, 642
 DB31, ... DBX128.0, 47
 DB31, ... DBX128.1, 47
 DB31, ... DBX14.0, 47
 DB31, ... DBX14.1, 47
 Debugmode für SB2, 171
 Default-Kennwort, 634
 DELAYFSTOF, 141
 DELAYFSTON, 141
 DELOBJ, 440
 D-Funktionen, 652
 DISABLE, 165
 DL-Funktionen, 652
 Dynamik
 -anpassung, 488
 Dynamischer NC-Speicher, 837
 DYNFINISH, 496
 DYNNORM, 496
 DYNPOS, 496
 DYNPREC, 496
 DYNROUGH, 496
 DYNSEMIFIN, 496

E

E/A-Bereich, 752
 EES, 194
 Eilgang
 Interpolationsarten, 516
 Einzelsatz
 -betrieb, 49
 SB1, 50, 169
 SB2, 50, 169

SB3, 50, 169
 -unterdrückung, 172
 Element, 416
 ENABLE, 165
 ENDLABEL, 121
 End-Programm, 608
 Ereignisgesteuerte Programmabläufe, 126
 Ersetzungsunterprogramm, 218
 EXTCLOSE, 206
 Externe Nullpunktverschiebung, 301
 Externe Programmspeicher, 189
 Externes Unterprogramm Abarbeiten, 190
 EXTOPEN, 206

F

FA-Funktionen, 653
 Feinverschiebung, 265
 F-Funktionen, 653
 FGROUP, 271, 274
 FIFO-Puffer, 190
 FRAME, 270
 Framedrehungen, 266
 in Werkzeugrichtung, 388, 389
 mit Raumwinkeln, 385
 Freiformflächen, 499
 -modus, 452, 499
 Führungssachse freigeben, 555
 Funktionalität, 578

G

G0-Toleranz, 523
 G0-Toleranzen, 518, 519
 G0-Toleranzfaktor, 519
 G58, 265
 G59, 265
 G60, 454
 G601, 455
 G602, 455
 G603, 455
 G64, 461
 G642, 466
 G643, 466
 G644, 469
 G645, 473
 G9, 454
 Genauhalt, 454
 Impliziter, 458
 Genauhaltbedingungen, 454, 455, 456
 Geometrieachsen, 270, 295

Geometrieachs-Verbund, 566
 GET, 555
 GETD, 556
 GFRAME0 ... GFRAME100, 338
 G-Gruppen, 105
 Glättung
 der Bahngeschwindigkeit, 484
 Grobverschiebung, 265
 GUD-Variablen, 614

H

Hauptlauf, 42
 Hauptlaufachsen, 272
 Hauptprogramm, 605
 Heimat-NCU, 789
 Helixinterpolation, 274
 H-Funktionen, 651
 Hilfsfunktion
 Adresserweiterung, 670
 anwenderdefiniert, 647
 Assoziierte, 678
 Ausgabeverhalten, 670
 Definition, 648
 Typ, 669
 vordefiniert, 647
 Wert, 670
 Hilfsfunktionen
 Anwenderspezifische, 675
 Vordefinierte, 654
 Hilfsfunktions
 -zähler, 698
 Hilfsfunktionsausgabe, 105

I

Identitätsvergleich, 252
 Impliziter Bahnsteuerbetrieb, 461
 Impliziter Genauhalt, 458
 impliziter Vorlaufstopp, 95
 Inbetriebnahmearchiv, 839
 INI-Programm, 606
 INIT, 531
 Interpolation
 Bei G0, 516
 Lineare, 517
 Nicht-lineare, 517
 Interpolator-Ende, 455
 Interrupt
 -routine, 153

- signal, 154
- sperre, 165
- Interruptroutine
 - Ende, 155
- Istwertsystem
 - werkstücknahes, 399

J

- Joggen
 - in Betriebsart AUTOMATIK, 38

K

- Kanal
 - Aktueller, 135
 - Eigenschaften, 42
 - Grundeinstellungen, 105
 - Konfiguration, 42
 - Zustand anzeigen, 115
 - Zustände, 116
- Kanalachsen, 270
- Kanalzustand
 - Kanal aktiv, 37
 - Kanal unterbrochen, 37
 - Kanal-Reset, 37
- Kaskadierter Satzsuchlauf, 65
- Kennwort, 633
- Kinematische Ketten, 416
- Kinematische Struktur, 415
- kinematische Transformation, 295
- Kollisionsvermeidung
 - Beispiel Grundlagen, 444
- Komparator-Eingänge, 742
- Kompilieren, 580
- Kontur
 - abtastfaktor, 500
 - abtastzeit, 500
 - toleranz, 510
- Krümmung, 499

L

- Label, 121
- Laderachsen, 271
- Lage der Koordinatensysteme und Referenzpunkte, 280
- Laufzeiten
 - Programm-, 243
- Laufzeitoptimierung, 578
- LIFTFAST, 164

- LookAhead, 474
 - An- und Abwahl, 476

M

- M1, 714
- M17, 714
- M2, 714
- M30, 714
- Maschinenachsen, 269
- Maschinenkoordinatensystem (MKS), 263, 284
- Maschinennullpunkt M, 278
- MD10000, 262
- MD10010, 32, 531
- MD10050, 778
- MD10061, 778
- MD10070, 778
- MD10071, 778
- MD10125, 200
- MD10131, 627
- MD10185, 779
- MD10260, 815
- MD10280, 531
- MD10300, 725, 737
- MD10310, 725, 740
- MD10320, 726, 736
- MD10330, 726, 739
- MD10350, 725, 733
- MD10360, 725, 732, 733
- MD10361, 734
- MD10362, 726
- MD10364, 726
- MD10366, 726
- MD10368, 726
- MD10398, 747
- MD10399, 748
- MD10500, 753
- MD10501, 754
- MD10502, 754
- MD10510, 753
- MD10511, 754
- MD10512, 754
- MD10530, 742
- MD10531, 743
- MD10540, 743
- MD10541, 743
- MD10600, 312, 387
- MD10602, 348, 350, 358, 363
- MD10610, 266, 343
- MD10612, 343, 344
- MD10615, 394
- MD10680, 501

MD10682, 501
MD10700, 147, 577, 579, 581
MD10702, 51, 59, 161, 171
MD10707, 72
MD10708, 71
MD10712, 242, 498
MD10713, 687
MD10714, 220, 665
MD10715, 219
MD10716, 219
MD10719, 222
MD10722, 562
MD10735, 38
MD10804, 220
MD10806, 221
MD10814, 221
MD11100, 676
MD11110, 682
MD11411, 45
MD11450, 61, 62, 63, 65, 73, 99, 694
MD11470, 74, 76, 101
MD11550, 141
MD11600, 158
MD11602, 157, 159, 160, 161, 602
MD11604, 99, 157, 160, 602
MD11610, 168
MD11620, 131
MD11625, 201
MD11626, 201
MD12030, 477, 478
MD12100, 477, 478
MD12701, 800, 827
MD12702, 800, 827
MD12703, 800, 827
MD12704, 800, 827
MD12705, 800, 827
MD12706, 800, 827
MD12707, 800, 827
MD12708, 800, 827
MD12709, 800, 827
MD12710, 800, 827
MD12711, 800, 827
MD12712, 800, 827
MD12713, 800, 827
MD12714, 800, 827
MD12715, 800, 827
MD12716, 800, 827
MD12717, 800
MD12750, 796
MD15700, 231
MD15702, 231
MD16800, 421
MD17200, 147
MD17950, 840
MD18000, 779
MD18050, 844, 845
MD18060, 842, 843
MD18150, 628
MD18170, 579
MD18180, 579
MD18210, 844, 845
MD18230, 841, 843
MD18242, 580
MD18351, 580, 600
MD18352, 842, 843
MD18353, 842, 843
MD18360, 190
MD18362, 190
MD18600, 310
MD18602, 333, 336, 339, 340
MD18720, 770
MD18880, 421
MD18882, 421
MD19250, 843
MD19251, 845
MD20000, 42
MD20050, 214, 262, 369, 604
MD20060, 262, 584
MD20070, 262
MD20080, 262, 274, 584
MD20090, 713
MD20094, 221, 665
MD20095, 221, 665
MD20105, 99, 166
MD20106, 63, 132, 171
MD20107, 63, 132
MD20108, 130
MD20109, 131
MD20110, 157, 393, 394, 396, 403, 815
MD20112, 72, 157, 214, 398, 403
MD20115, 99, 160, 166
MD20116, 161
MD20117, 161, 171
MD20118, 214
MD20120, 214
MD20121, 214
MD20124, 713
MD20130, 214
MD20140, 214
MD20144, 216
MD20150, 42, 105, 214, 325, 396, 397, 526
MD20152, 214, 396
MD20170, 505
MD20171, 505

MD20172, 505
MD20173, 505
MD20184, 389
MD20191, 162
MD20192, 133
MD20193, 133
MD20194, 157, 183
MD20270, 652
MD20272, 652
MD20310, 87
MD20400, 477
MD20430, 478
MD20440, 478
MD20443, 481
MD20450, 479
MD20460, 485, 486
MD20462, 486
MD20465, 489, 490
MD20480, 467, 470, 510
MD20482, 505, 510
MD20485, 505
MD20486, 505
MD20487, 505
MD20488, 509
MD20490, 461
MD20550, 457
MD20552, 457
MD20560, 519
MD20561, 520
MD20562, 520
MD20606, 500
MD20700, 160
MD20730, 519
MD20750, 517
MD20800, 649, 714
MD21220, 727, 744
MD21330, 214
MD22000, 676
MD22010, 677
MD22020, 677
MD22030, 677
MD22035, 678
MD22040, 669, 676
MD22050, 669
MD22060, 670
MD22070, 670
MD22080, 232, 670
MD22100, 687
MD22110, 651, 653, 654
MD22200, 680
MD22210, 650, 680
MD22220, 651, 680
MD22230, 651, 680
MD22240, 653, 680
MD22250, 652, 680
MD22252, 653, 680
MD22254, 221, 665, 678
MD22256, 221, 665, 678
MD22510, 105
MD22530, 708
MD22532, 708
MD22534, 708
MD22550, 222
MD22560, 221, 665, 713
MD22600, 85
MD22620, 103
MD24004, 394
MD24006, 303, 325, 395
MD24007, 397
MD24008, 303, 394
MD24010, 343
MD24020, 331
MD24040, 368, 369
MD24120, 604
MD24805, 358
MD24855, 358
MD24905, 351
MD24955, 351
MD26008, 221, 665
MD26012, 221
MD27100, 148
MD27800, 43
MD27850, 247
MD27860, 119, 247
MD27880, 120, 250
MD27882, 251
MD28010, 580
MD28020, 580
MD28040, 580
MD28060, 144, 147
MD28070, 509
MD28071, 506
MD28072, 506
MD28082, 325, 345, 381, 382, 389
MD28090, 598
MD28100, 598
MD28105, 599
MD28150, 630
MD28400, 147
MD28402, 147
MD28530, 469
MD28533, 481
MD28560, 398
MD28610, 500

MD30552, 557
 MD31050, 642
 MD31060, 642
 MD32000, 519
 MD32060, 101
 MD32074, 394, 566
 MD32200, 642
 MD32310, 460
 MD32420, 517
 MD32430, 517
 MD32440, 486, 489
 MD32800, 642
 MD32810, 643
 MD32910, 643
 MD33100, 467, 501, 505, 510
 MD33120, 473, 510
 MD35000, 262, 320
 MD35130, 642
 MD35240, 469
 MD35590, 642
 MD36000, 454
 MD36010, 454
 MD36012, 456
 MD36610, 623
 MD36620, 623
 MD51029, 181
 MD51074, 251
 MD60900+i, 597
 MD62571, 600
 MD62572, 600
 MD62574, 601
 MD62575, 613
 MD62580, 597
 MD9004, 149
 MD9010, 149
 MD9011, 149
 MD9424, 149, 299
 MD9440, 402
 M-Funktionsersetzung, 219
 MTL, 221

N

NAMETOINT, 443
 NC
 -Sprachumfang, 105
 -Start, 110
 NC/PLC-Nahtstelle, 42
 negative Adresserweiterung, 715
 Normen, 621
 Not-Halt
 Ablauf, 623

Nahtstelle, 622
 Quittierung, 624
 Not-Halt-Stellteile, 622
 Nullpunkte, 278
 Nullpunktverschiebung
 Externe Nullpunktverschiebung, 301

O

Orientierungs
 -toleranz, 510
 OTOL, 511

P

Paket
 -zähler, 698
 Parallele Teilketten, 417
 Passives Filesystem, 838
 Passwörter, 634
 PLC
 -Achsen, 272
 -Variable lesen und schreiben, 628
 POS, 271, 272
 POSA, 271, 272
 Positionierachsen, 271
 PRESETON, 285
 PRESETONS, 290
 Probelaufvorschub, 52
 Process DataShare, 202, 206
 Programm
 -aktion, 117
 -laufzeiten, 243
 -test, 46
 -zustand anzeigen, 114
 -zustände, 115
 Programmbetrieb, 104
 Programmhandling, 579
 Programmteil
 -wiederholung, 121

R

Randbedingungen, 584
 RangeOffset, 757
 Referenzpunkt R, 278
 Referenzpunkte, 278
 Regler-Parametersatz-Umschaltung, 642
 RELEASE, 554
 REPEAT, 121
 REPEATB, 121

- Reset
 - Kommando,
 - verhalten, 211
- RESET-Verhalten, 604
- Restzeit
 - für ein Werkstück, 245
- RESU-Arbeitsebene, 597
- RESU-ASUP, 609
- RESU-fähiger Konturbereich, 592
- RESU-Hauptprogramm, 605
- Rotationsanteil, 383
- Ruckbegrenzung, 471
- Rundachsen, 271

- S**
- Satz
 - ausblenden, 180
- Satzsuchlauf
 - Kaskadierter, 57
 - mit Berechnung an der Kontur (Typ 2), 56
 - mit Berechnung an Satzendpunkt (Typ 4), 57
 - mit Berechnung im Modus Programmtest SERUPRO (Typ 5), 57
 - ohne Berechnung (Typ 1), 56
- Satzsuchlauf mit Berechnung
 - aufgesammelte Spindelfunktionen, 62
- Satzsuchlauf SERUPRO
 - Bahnachsen, 78
 - Bedingungen für Achsfunktionen, 100
 - Getriebestufenwechsel, 102
 - Grundeinstellung, 103
 - Positionierachsen Wiederanfahren, 77
 - REPOS mit NC/PLC-Nahtstellensignale steuern, 79
 - REPOS-Quittungen, 80
 - REPOS-Verhalten einstellen, 74
 - REPOS-Verschiebung bei Synchronspindelkopplung, 83
 - REPOS-Verschiebung im Gültigkeitsbereich, 82
 - REPOS-Verschiebung nach einem Achstausch, 82
 - Soll- und Istwertkopplungen, 97
 - Überlagerte Bewegungen, 102
 - Zeitlicher Ablauf, 71
- SAVE, 165
- SBLOF, 172
- SBLON, 172
- Schlüsselschalter, 635
- Schneller Datenkanal, 628
- Schraubenlinieninterpolation, 274
- Schutzstufe
 - bei Anwender-ASUP, 168
- Schutzstufen, 631
 - parametrierbare, 636
- SD41600, 744
- SD41601, 744
- SD42100, 52
- SD42200, 172
- SD42440, 267
- SD42444, 59
- SD42465, 467, 510
- SD42466, 467, 510
- SD42470, 506
- SD42471, 506
- SD42472, 506
- SD42473, 506
- SD42475, 506
- SD42476, 506
- SD42477, 506
- SD42676, 511
- SD42678, 511
- SD42700, 191
- SD42750, 146
- SD42990, 144
- Sekantenfehler, 501
- Self-Acting SERUPRO, 89
- Serieninbetriebnahme, 601
- SERUPRO
 - automatischer Unterbrechungszeiger, 93
 - Ende-ASUP, 702
 - programmierbarer Unterbrechungszeiger, 90
- SERUPRO-Anfahren
 - vom PLC beeinflussen, 79
- SERUPRO-ASUP
 - Besonderheiten, 86
- seruproMasterChan, 89
- SETINT, 153, 163
- SETM, 531, 538
- S-Funktionen, 650
- Simulation, 55
- Simulierten Zielpunkt für LEAD mit JOG erreichen, 98
- Speicherbedarf, 580
- Speichererweiterung, 842
- Spiegelung
 - Frames, 320
- Spindelfunktionen über PLC, 44
- Spline, 453
- Sprachumfang, 584
- Sprungmarke
 - bei Programmteilwiederholungen, 121
- START, 531

Statischer NC-Speicher, 837
 Steuerungsverhalten
 bei Hochlauf, 211
 bei Reset, 211
 bei Teileprogrammende, 211
 bei Teileprogrammstart, 211
 STOLF, 523
 Stop-Delay-Bereich, 137
 Stopp-Ereignisse, 138
 STRINGIS, 106
 Synchronachsen, 273
 Syntax-Check, 584

T

TCP-Tool Center Position, 278
 TEACH IN, 36
 Teileprogramm
 Anwahl, 110
 -sätze ausblenden, 54
 T-Funktion, 651
 T-Funktionsersetzung, 221
 Tick, 698
 Toleranz
 Bei G0, 518
 Torsion, 499
 TRANS, 265
 Transformation, 618
 TRANSMIT, 295

U

Überlastfaktor, 460
 Überschleifen, 461
 Unterbrechungspunkt, 592

V

V2 Vorverarbeitung
 Kurzbeschreibung, 577
 Verriegelbare Datenbereiche, 636
 Vorlauf, 42

W

WAITE, 531
 WAITENC, 216
 WAITM, 531
 WAITMC, 531, 537
 Wegkriterium, 464

Weltkoordinatensystem, 418
 Werkstück
 -simulation, 55
 -zähler, 249, 254
 Werkstückkoordinatensystem (WKS), 263, 264, 300
 Werkstücknullpunkt W, 278
 Werkzeug
 -rückzug, 215
 -verwaltung, 31
 Werkzeugkorrekturen, 619
 Werkzeugmagazinachsen, 271
 Werkzeugrevolverachsen, 271
 Werkzeugträgerbezugspunkt T, 278
 Wiederaufsetz-ASUP, 609
 Wiederaufsetzpunkt, 592

X

XE * MERGEFORMAT,

Z

Zählimpuls, 252
 Zugriffsmerkmale, 632
 Zugriffsrecht, 580
 Zugriffsrechte, 631
 Zugriffsschutz, 631
 Zusammenfassen von Hilfsfunktionen, 670
 Zusatzachsen, 270

