SIEMENS

SINUMERIK

SINUMERIK 840D sl/ SINUMERIK 828D Erweiterungsfunktionen

Funktionshandbuch

Gültig für

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

Software	*Version*
NCU Systemsoftware für 840D sl/840DE sl	2.6
NCU Systemsoftware für 828D	2.6

09/2009
6FC5397-1BP20-0AA0

Vorwort
A4: Digitale und analoge NCK-Peripherie 1
B3: Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme - nur 840D sl 2
B4: Bedienung über PG/PC - nur 840D sl 3
H1: Handfahren und Handradfahren 4
K3: Kompensationen 5
K5: BAGs, Kanäle, Achstausch 6
M1: Kinematische Transformation 7
M5: Messen 8
N3: Softwarenocken, Wegschaltsignale - nur 840D sl 9
N4: Stanzen und Nibbeln - nur 840D sl 10
P2: Positionierachsen 11
P5: Pendeln - nur 840D sl 12
R2: Rundachsen 13
S3: Synchronspindel 14
S7: Speicherkonfiguration 15
T1: Teilungsachsen 16
W3: Werkzeugwechsel 17
W4: Schleifspezifische Werkzeugkorrektur und Überwachungen - nur 840D sl 18
Z2: NC/PLC-Nahtstellensignale 19
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
mit Warndreieck bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

VORSICHT
ohne Warndreieck bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

ACHTUNG
bedeutet, dass ein unerwünschtes Ergebnis oder Zustand eintreten kann, wenn der entsprechende Hinweis nicht beachtet wird.

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.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
DEUTSCHLAND

Dokumentbestellnummer: 6FC5397-1BP20-0AA0
℗ 10/2009

Copyright © Siemens AG 2009.
Änderungen vorbehalten

Vorwort

SINUMERIK-Dokumentation

Die SINUMERIK-Dokumentation ist in 3 Kategorien gegliedert:

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

Unter dem Link http://www.siemens.com/motioncontrol/docu gibt es Informationen zu folgenden Themen:

- Dokumentation bestellen
  Hier finden Sie die aktuelle Druckschriftenübersicht.
- Dokumentation downloaden
  Weiterführende Links für den Download von Dateien aus Service & Support.
- Dokumentation online recherchieren
  Informationen zur DOConCD und direkten Zugriff auf die Druckschriften im DOConWEB.
- Dokumentation auf Basis der Siemens Inhalte individuell zusammenstellen mit dem My Documentation Manager (MDM), siehe http://www.siemens.com/mdm

  Der My Documentation Manager bietet Ihnen eine Reihe von Features zur Erstellung Ihrer eigenen Maschinendokumentation.
- Training und FAQs

  Informationen zum Trainingsangebot und zu FAQs (frequently asked questions) finden Sie über die Seitennavigation.

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 Betriebs und der Instandhaltung berücksichtigen.

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
  - Faxvorlage für Feedback zur Dokumentation
  - Dokumentationsübersicht
- Begriffsverzeichnis

Hinweis

Ausführliche Daten- und Alarm-Beschreibungen siehe:

- für Maschinen- und Settingdaten:
  Ausführliche Maschinendaten-Beschreibung (nur elektronisch auf DOConCD oder DOConWEB)
- für NC/PLC-Nahtstellensignale:
  Funktionshandbuch Grundfunktionen; NC/PLC-Nahtstellensignale (Z1)
  Funktionshandbuch Erweiterungsfunktionen; NC/PLC-Nahtstellensignale (Z2)
  Funktionshandbuch Sonderfunktionen; NC/PLC-Nahtstellensignale (Z3)
- 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)

Hinweis

Signaladresse

Die Funktionsbeschreibungen enthalten als <Signaladresse> eines NC/PLC-Nahtstellensignals nur die für SINUMERIK 840D sl gültige Adresse. Die Signaladresse für SINUMERIK 828D ist den Datenlisten "Signale an/von ..." am Ende der jeweiligen Funktionsbeschreibung zu entnehmen.

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 werden die folgenden elementaren Datentypen verwendet:

Typ	Bedeutung	Wertebereich
INT	Ganzzahlige Werte mit Vorzeichen	-2147483648 ... +2147483647
REAL	Zahlen mit Dezimalpunkt nach IEEE	$\pm(2{,}2*10^{-308} \ldots 1{,}8*10^{+308})$
BOOL	Wahrheitswerte TRUE (1) und FALSE (0)	1, 0
CHAR	Zeichen ASCII	entsprechend Code 0 ... 255
STRING	Zeichenkette, Zeichenzahl in [...]	maximal 200 Zeichen (keine Sonderzeichen)
AXIS	nur Achsnamen (Achsadressen)	alle im Kanal vorhandenen Achsadressen
FRAME	geometrische Angaben für Verschieben, Drehen, Skalieren, Spiegeln	
Arrays können nur aus gleichen elementaren Datentypen gebildet werden. Es sind maximal 3-dimensionale Arrays möglich.		

Technical Support

Bei Fragen wenden Sie sich bitte an folgende Hotline:

	Europa / Afrika
Telefon	+49 180 5050 - 222
Fax	+49 180 5050 - 223
0,14 €/Min. aus dem deutschen Festnetz, abweichende Mobilfunkpreise möglich	
Internet	http://www.siemens.de/automation/support-request

	Amerika
Telefon	+1 423 262 2522
Fax	+1 423 262 2200
E-Mail	mailto:techsupport.sea@siemens.com

	Asien / Pazifik
Telefon	+86 1064 757575
Fax	+86 1064 747474
E-Mail	mailto:support.asia.automation@siemens.com

Hinweis

Landesspezifische Telefonnummern für technische Beratung finden Sie im Internet:
http://www.automation.siemens.com/partner

Fragen zur Dokumentation

Bei Fragen zur Dokumentation (Anregungen, Korrekturen) senden Sie bitte ein Fax oder eine E-Mail an folgende Adresse:

Fax: +49 9131- 98 2176
E-Mail: mailto:docu.motioncontrol@siemens.com

Eine Faxvorlage finden Sie im Anhang dieses Dokuments.

Internetadresse für SINUMERIK

http://www.siemens.com/sinumerik

SINUMERIK 828D Systemperformance (Region)

	PPU 260 / 261		PPU 280 / 281	
	T	M	T	M
Systemperformance				
Grundumfang Achsen/Spindeln	3	4	3	4
max. Anzahl Achsen/Spindeln	6	6	8	6
max. Anzahl interpolierender Achsen	4	4	4	4
max. Anzahl Kanäle/Betriebsartengruppen	1/1	1/1	1/1	1/1
min. Satzwechselzeit	~ 6 ms	~ 2 ms	~ 6 ms	~ 1 ms
Drehzahl-/Stromregeltakt	125 µs	125 µs	125 µs	125 µs
CNC-Anwenderspeicher (gepuffert)	3 MB	3 MB	5 MB	5 MB
CNC-Funktionen				
Werkzeugverwaltung	●	●	●	●
Anzahl Werkzeuge/Schneiden	128/256	128/256	256/512	256/512
Safety Integrated: Safe Torque Off, Safe Brake Control	●	●	●	●
Safety Integrated: Sichere Geschwindigkeit	○	○	○	○
max. Anzahl ASUPs (fest eingestellt)	2	2	2	2
TRANSMIT / TRACYL	○	○	○	○
Schräge Y-Achse	-	-	○	-
Synchronspindel für Gegenspindel	-	-	○	-
Synchronspindel für Mehrkantschlagen	○	-	○	-
Gantry	○	○	○	○

	PPU 260 / 261		PPU 280 / 281	
	T	M	T	M
Temperaturkompensation	●	●	●	●
HMI, CNC-Programmierung				
ShopMill / ShopTurn HMI-Funktionen	●	●	●	●
DIN/ISO-Programmierung mit programGUIDE	●	●	●	●
Online ISO Dialekt Interpreter	●	●	●	●
ShopMill / ShopTurn Arbeitsschrittprogrammierung	○	○	○	○
Messzyklen	○	○	○	○
Simulation in Flächendarstellung	●	●	●	●
Simulation in 3D-Darstellung	○	○	○	○
PLC				
SIMATIC S7-200 (integriert)	●	●	●	●
Einfachperipheriemodule: PP 72/48D PN PP 72/48D 2/2A PN (in Vorbereitung)	○	○	○	○
PLC-Zykluszeit	6 ms	6 ms	6 ms	6 ms
max. Anzahl PLC-Anweisungen	24000	24000	24000	24000
Servo-synchrone High Speed PLC-Task	●	●	●	●
Reaktionszeit auf Prozessereignisse (Klemme zu Klemme)	7,5 ms	7,5 ms	4,5 ms	4,5 ms
max. Anzahl digitaler Ein-/Ausgänge	288/192	288/192	360/240	360/240
max. Anzahl analoger Ein-/Ausgänge	8/8	8/8	10/10	10/10
Inbetriebnahmefunktionen				
Service Planner (Wartungsplaner)	●	●	●	●
Easy Extend (Verwalten von Maschinenaggregaten)	●	●	●	●
Easy Archive (Datenarchivierung)	●	●	●	●

T Drehen
M Fräsen
● Standard (Grundumfang)
○ CNC-Option
\- nicht verfügbar

Inhaltsverzeichnis

Vorwort 3

1 A4: Digitale und analoge NCK-Peripherie 27

1.1 Kurzbeschreibung 27

1.2 NCK-Peripherie über PLC 28
1.2.1 Allgemeine Funktionalität 28
1.2.2 Digitale Ein-/Ausgänge des NCK 35
1.2.2.1 Digitale Eingänge des NCK 35
1.2.2.2 Digitale Ausgänge des NCK 37
1.2.3 Durchschalten und Verknüpfen von schnellen NCK-Ein-/Ausgängen 41
1.2.4 Analoge Ein-/Ausgänge des NCK 43
1.2.4.1 Analoge Eingänge des NCK 43
1.2.4.2 Analoge Ausgänge des NCK 45
1.2.5 Direkte, von NC adressierbare PLC-Peripherie 49
1.2.6 Analogwertdarstellung der Analog-Ein-/Ausgabewerte des NCK 52
1.2.7 Komparator-Eingänge 54

1.3 NCK-Peripherie über PROFIBUS - nur 840D sl 58
1.3.1 Funktionalität 58
1.3.2 Parametrierung 59
1.3.3 Programmierung 61
1.3.3.1 Kommunikation über Teileprogramm/Synchronaktionen 61
1.3.3.2 Kommunikation über Compile-Zyklen 64

1.4 Randbedingungen 66
1.4.1 NCK-Peripherie über PLC 66
1.4.2 NCK-Peripherie über PROFIBUS - nur 840D sl 67

1.5 Beispiele 68
1.5.1 NCK-Peripherie über PLC 68
1.5.1.1 Schreiben auf PLC-Peripherie 68
1.5.1.2 Lesen von PLC-Peripherie 69
1.5.2 NCK-Peripherie über PROFIBUS- nur 840D sl 70
1.5.2.1 PROFIBUS-Peripherie in Schreibrichtung 70
1.5.2.2 PROFIBUS-Peripherie in Leserichtung 71
1.5.2.3 Abfrage des RangeIndex bei "PROFIBUS-Peripherie in Schreibrichtung" 73

1.6 Datenlisten 75
1.6.1 Maschinendaten 75
1.6.1.1 Allgemeine Maschinendaten 75
1.6.1.2 Kanal-spezifische Maschinendaten 76
1.6.2 Settingdaten 76
1.6.2.1 Allgemeine Settingdaten 76
1.6.3 Signale 76
1.6.3.1 Signale an NC 76
1.6.3.2 Signale von NC 77

2 B3: Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme - nur 840D sl 79

2.1 Kurzbeschreibung 79
2.1.1 Topologie dezentraler Systemaufbauten 79

2.1.2 Mehrere Bedientafeln und NCUs mit Bedieneinheitenmanagement (Option) ... 84
2.1.2.1 Allgemeines ... 84
2.1.2.2 Systemmerkmale ... 85
2.1.2.3 Hardware ... 86
2.1.2.4 Funktionen ... 88
2.1.2.5 Projektierbarkeit ... 89
2.1.3 Mehrere Bedientafelfronten und NCUs Standardlösung ... 90
2.1.3.1 Systemmerkmale ... 90
2.1.3.2 Funktionen ... 91
2.1.3.3 Projektierbarkeit ... 93
2.1.3.4 MPI/BTSS-Netzwerkregeln ... 97
2.1.4 NCU-Link ... 98
2.1.4.1 Allgemeines ... 98
2.1.4.2 Ausführungsformen dezentraler Maschinen ... 99
2.1.4.3 Link-Achsen ... 101
2.1.4.4 Flexible Konfiguration ... 102
2.1.4.5 NCU-übergreifende Anwenderkommunikation ... 103
2.1.4.6 Lead-Link-Achsen ... 104
2.1.4.7 NCU-Link mit unterschiedlichem Interpolationstakt ... 104

2.2 Mehrere Bedientafelfronten und NCUs mit Option Bedieneinheitenmanagement ... 107
2.2.1 Hardware-Struktur ... 107
2.2.2 Eigenschaften ... 107
2.2.3 Konfigurationsdatei NETNAMES.INI ... 109
2.2.4 Aufbau der Konfigurationsdatei ... 109
2.2.5 Konfigurationsdatei erstellen und anwenden ... 114
2.2.6 Hochlauf ... 115
2.2.7 HMI-Umschaltung ... 118
2.2.8 Verdrängung ... 118
2.2.9 An- und Umschaltbedingungen ... 120
2.2.10 Realisierung der Bedieneinheitenumschaltung ... 121
2.2.11 Bedienoberfläche ... 121
2.2.12 Bedienmodus-Umschaltung ... 122
2.2.13 MSTT-Umschaltung ... 124
2.2.14 Das PLC-Programm "Bedieneinheitenumschaltung" ... 125

2.3 Mehrere Bedientafelfronten und NCUs Standardlösung ... 130
2.3.1 Konfigurationen ... 131
2.3.2 Umschalten der Verbindung auf eine andere NCU ... 135
2.3.3 Konfigurationsdatei erstellen und anwenden ... 135
2.3.4 Hochlauf ... 136
2.3.5 NCU-Tausch ... 137

2.4 Einschränkungen beim Umschalten von Bedienkomponenten ... 138

2.5 NCU-Link ... 139
2.5.1 Einführung ... 139
2.5.2 Technologische Beschreibung ... 140

2.6 Link-Achsen ... 142
2.6.1 Konfiguration von Link-Achsen und Containerachsen ... 144
2.6.2 Achsdaten und Signale ... 148
2.6.3 Ausgabe vordefinierter Hilfsfunktionen bei NCU-Link ... 151
2.6.4 Randbedingungen für Link-Achsen ... 152
2.6.5 Programmierung mit Kanal- und Maschinenachsbezeichnern ... 153
2.6.6 Flexible Konfiguration ... 154

2.7 Achscontainer ... 155

2.7.1 Systemvariablen für Achscontainer 161
2.7.2 Bearbeitung mit Achscontainer (schematisch) 163
2.7.3 Verhalten der Achscontainer bei Power On....164
2.7.4 Verhalten der Achscontainer bei Betriebsartenwechsel 164
2.7.5 Verhalten der Achscontainer bei ASUPs 164
2.7.6 Verhalten der Achscontainer bei RESET....164
2.7.7 Verhalten der Achscontainer bei Satzsuchlauf 164
2.7.8 Randbedingungen für Achscontainer/-Drehungen 164

2.8 Link-Kommunikation 168
2.8.1 Link-Variablen 168
2.8.2 Lesen von Antriebsdaten über Link-Variablen....172

2.9 Konfiguration eines Link-Verbundes....174

2.10 Kommunikation im Link-Verbund....177

2.11 Lead-Link-Achse 182
2.11.1 Programmierung einer Lead-Link-Achse 186

2.12 NCU-Link bei unterschiedlichem Interpolationstakt....187
2.12.1 Allgemeine Lösung schematisch 189
2.12.2 Unterschiedlicher Lageregeltakt 192
2.12.3 Randbedingungen....194
2.12.4 Aktivierung von NCU-Link mit unterschiedlichem Interpolationstakt 195
2.12.5 Unterschiedliche IPO-Takte, Verhalten bei Power On, RESET ... .195
2.12.6 Systemvariable bei unterschiedlichem Interpolationstakt....195

2.13 Maßsystem eines Link-Verbundes 196

2.14 Randbedingungen....197
2.14.1 Mehrere Bedientafeln und NCUs Option Bedieneinheitenmanagement 197
2.14.2 Mehrere Bedientafelfronten und NCUs Standardlösung 197
2.14.3 Link-Achsen 198
2.14.4 Achscontainer 198
2.14.5 Lead-Link-Achse 198
2.14.6 NCU-Link bei unterschiedlichem Interpolationstakt....199

2.15 Beispiele....199
2.15.1 Konfigurationsdatei NETNAMES.INI mit Option Bedieneinheitenmanagement....199
2.15.2 Anwenderspezifische Umgestaltung des PLC-Programms Bedieneinheitenumschaltung 202
2.15.2.1 Beschreibung der Abläufe (Übersicht)....202
2.15.2.2 Beschreibung der Abläufe (Details) 203
2.15.2.3 Festgelegte logische Funktionen/Defines....213
2.15.2.4 Graphiken der Funktionssequenzen....213
2.15.3 Konfigurationsdatei NETNAMES.INI Standardlösung 221
2.15.3.1 Zwei Bedientafelfronten und eine NCU 221
2.15.3.2 Eine Bedientafelfront und drei NCUs....221
2.15.4 Kurzinbetriebnahme M:N anhand von Beispielen....223
2.15.4.1 Beispiel 1....223
2.15.4.2 Beispiel 2....226
2.15.4.3 Beispiel 3....232
2.15.4.4 Beschreibung von FB9....236
2.15.4.5 Aufrufbeispiel für FB9 238
2.15.4.6 Beispiel für Override-Umschaltung 240
2.15.4.7 Umschaltung zwischen MSTT und HT6 241
2.15.4.8 Allgemeine Hinweise....242
2.15.5 Link-Achse 244
2.15.6 Achscontainer-Koordinierung....245

2.15.6.1 Achscontainer-Drehung ohne Warten des Teileprogramms ... 245
2.15.6.2 Achscontainer-Drehung mit implizitem Warten des Teileprogramms ... 246
2.15.6.3 Achscontainer-Drehung durch nur einen Kanal (z. B. Hochlauffall) ... 246
2.15.7 Achscontainer Systemvariablen auswerten ... 246
2.15.7.1 Bedingte Verzweigung ... 246
2.15.7.2 Statische Synchronaktion mit $AN_AXCTSWA ... 246
2.15.7.3 Sicher auf Achscontainer-Drehung warten ... 246
2.15.8 Konfiguration einer Mehrspindel-Drehmaschine ... 248
2.15.9 Lead-Link-Achse ... 257
2.15.9.1 Konfiguration ... 257
2.15.9.2 Programmierung ... 259
2.15.10 NCU-Link mit unterschiedlichem Interpolationstakt ... 260
2.15.10.1 Beispiel Unrund-Drehen ... 260

2.16 Datenlisten ... 262
2.16.1 Maschinendaten ... 262
2.16.1.1 Allgemeine Maschinendaten ... 262
2.16.1.2 Kanal-spezifische Maschinendaten ... 262
2.16.1.3 Achs-/Spindel-spezifische Maschinendaten ... 263
2.16.2 Settingdaten ... 263
2.16.2.1 Allgemeine Settingdaten ... 263
2.16.2.2 Achs-/Spindel-spezifische Settingdaten ... 263
2.16.3 Signale ... 263
2.16.3.1 Signale von NC ... 263
2.16.3.2 Signale von HMI/PLC ... 264
2.16.3.3 Allgemein Online-Schnittstelle ... 264
2.16.3.4 Signale von Achse/Spindel ... 266

3 B4: Bedienung über PG/PC - nur 840D sl ... 267

3.1 Kurzbeschreibung ... 267

3.2 Software-Installation ... 269
3.2.1 Systemvoraussetzungen ... 269
3.2.2 Installation ... 270
3.2.3 SW-Randbedingungen ... 275
3.2.4 Programm starten ... 275
3.2.5 Programm beenden ... 276

3.3 Bedienung über PG/PC ... 277
3.3.1 Allgemeine Bedienung ... 277
3.3.2 Zusatzinformationen ... 279
3.3.3 Betrieb von Bedientafelfronten ... 280

3.4 Simulation von Teileprogrammen ... 281

3.5 Randbedingungen ... 281

3.6 Datenlisten ... 281

4 H1: Handfahren und Handradfahren ... 283

4.1 Kurzbeschreibung ... 283
4.1.1 Übersicht ... 283
4.1.2 Allgemeine Eigenschaften beim Handfahren in JOG ... 284
4.1.3 Steuerung der Handfahr-Funktionen über PLC-Nahtstelle ... 287
4.1.4 Steuerungsverhalten bei Power On, Betriebsartenwechsel, Reset, Satzsuchlauf, Repos ... 288

4.2 Kontinuierliches Verfahren ... 289
4.2.1 Allgemeine Funktionalität ... 289
4.2.2 Unterscheidung Tippbetrieb / Dauerbetrieb ... 289

4.2.3 Besonderheiten beim kontinuierlichen Verfahren 291

4.3 Inkrementelles Verfahren (INC) 292
4.3.1 Allgemeine Funktionalität 292
4.3.2 Unterscheidung im Tippbetrieb / Dauerbetrieb 292
4.3.3 Besonderheiten beim inkrementellen Verfahren 294

4.4 Handradfahren im JOG 295
4.4.1 Allgemeine Funktionalität 295
4.4.2 Fahranforderung 299
4.4.3 Doppelverwendung des Handrades 303

4.5 Handradüberlagerung in Automatik 305
4.5.1 Allgemeine Funktionalität 305
4.5.2 Programmierung und Aktivierung der Handradüberlagerung 309
4.5.3 Besonderheiten bei Handradüberlagerung in Automatik 311

4.6 Konturhandrad/Wegvorgabe mit Handrad (Option) 312

4.7 DRF-Verschiebung 315

4.8 Inbetriebnahme: Handräder 318
4.8.1 Allgemeines 318
4.8.2 Anschluss über PPU - nur 828D 319
4.8.3 Anschluss über PROFIBUS - nur 840D sl 320
4.8.4 Anschluss über Ethernet - nur 840D sl 322

4.9 Besonderheiten beim Handfahren 326
4.9.1 Geometrieachsen beim Handfahren 326
4.9.2 Besonderheiten beim Handfahren der Spindel 327
4.9.3 Überwachungen 328
4.9.4 Sonstiges 330

4.10 Festpunkt anfahren in JOG 331
4.10.1 Einführung 331
4.10.2 Funktionalität 332
4.10.3 Parametrierung 334
4.10.4 Programmierung 335
4.10.5 Randbedingungen 335
4.10.6 Anwendungsbeispiel 336

4.11 Datenlisten 337
4.11.1 Maschinendaten 337
4.11.1.1 Allgemeine Maschinendaten 337
4.11.1.2 Kanal-spezifische Maschinendaten 337
4.11.1.3 Achs-/Spindel-spezifische Maschinendaten 338
4.11.2 Settingdaten 338
4.11.2.1 Allgemeine Settingdaten 338
4.11.3 Signale 339
4.11.3.1 Signale von NC 339
4.11.3.2 Signale an BAG 339
4.11.3.3 Signale von BAG 339
4.11.3.4 Signale an Kanal 339
4.11.3.5 Signale von Kanal 340
4.11.3.6 Signale an Achse/Spindel 340
4.11.3.7 Signale von Achse/Spindel 341

5 K3: Kompensationen 343

5.1 Einführung 343

5.2 Temperaturkompensation ... 345
5.2.1 Funktionsbeschreibung ... 345
5.2.2 Inbetriebnahme ... 348
5.2.2.1 Temperaturabhängige Parameter ... 348
5.2.2.2 Temperaturkompensationstyp und Aktivierung ... 348
5.2.2.3 Maximaler Kompensationswert pro IPO-Takt ... 349
5.2.3 Beispiel ... 349
5.2.3.1 Inbetriebnahme der Temperaturkompensation für die Z-Achse einer Drehmaschine ... 349

5.3 Losekompensation ... 352
5.3.1 Funktionsbeschreibung ... 352
5.3.2 Inbetriebnahme ... 353
5.3.2.1 Umkehrlose ... 353
5.3.2.2 Bewertungsfaktor für Umkehrlose ... 354
5.3.2.3 Schrittweises Aufschalten der Losekompensation ... 354

5.4 Interpolatorische Kompensation ... 355
5.4.1 Allgemeine Eigenschaften ... 355
5.4.2 Kompensation von Spindelsteigungsfehler und Messsystemfehler ... 358
5.4.2.1 Funktionsbeschreibung ... 358
5.4.2.2 Inbetriebnahme ... 360
5.4.2.3 Beispiel ... 363
5.4.3 Kompensation von Durchhang und Winkligkeitsfehler ... 364
5.4.3.1 Funktionsbeschreibung ... 364
5.4.3.2 Inbetriebnahme ... 368
5.4.3.3 Beispiele ... 372
5.4.4 Richtungsabhängige Spindelsteigungsfehler-Kompensation ... 381
5.4.4.1 Funktionsbeschreibung ... 381
5.4.4.2 Inbetriebnahme ... 382
5.4.4.3 Beispiel ... 386
5.4.5 Erweiterung der Durchhangkompensation mit NCU-Link - nur 840D sl ... 389
5.4.6 Besonderheiten der Interpolatorischen Kompensation ... 397

5.5 Dynamische Vorsteuerung (Schleppfehler-Kompensation) ... 399
5.5.1 Allgemeine Eigenschaften ... 399
5.5.2 Drehzahlvorsteuerung ... 401
5.5.3 Momentenvorsteuerung - nur 840D sl (Option) ... 403
5.5.4 Dynamikanpassung ... 405
5.5.5 Vorsteuerung bei Kommando- und PLC-Achsen ... 406
5.5.6 Randbedingungen ... 408

5.6 Reibkompensation (Quadrantenfehler-Kompensation) ... 409
5.6.1 Allgemeine Eigenschaften ... 409
5.6.2 Konventionelle Reibkompensation ... 410
5.6.2.1 Konventionelle Reibkompensation ... 410
5.6.2.2 Inbetriebnahme der konventionellen Reibungskompensation ... 412
5.6.3 Quadrantenfehler-Kompensation mit neuronalen Netzen - nur 840D sl ... 421
5.6.3.1 Grundlagen ... 421
5.6.3.2 Parametrierung der neuronalen QFK ... 423
5.6.3.3 Lernen des neuronalen Netzes ... 429
5.6.3.4 Inbetriebnahme der neuronalen QFK ... 434
5.6.3.5 Weitere Optimierungs- und Eingriffsmöglichkeiten ... 437
5.6.3.6 Schnellinbetriebnahme ... 443

5.7 Kreisformtest ... 446

5.8 Maßnahmen bei hängenden Achsen ... 451
5.8.1 Elektronischer Gewichtsausgleich ... 451

5.8.2 Reboot-Verzögerung....452

5.9 Datenlisten....455
5.9.1 Maschinendaten....455
5.9.1.1 Allgemeine Maschinendaten....455
5.9.1.2 Kanal-spezifische Maschinendaten....455
5.9.1.3 Achs-/Spindel-spezifische Maschinendaten....455
5.9.2 Settingdaten....456
5.9.2.1 Allgemeine Settingdaten....456
5.9.2.2 Achs-/Spindel-spezifische Settingdaten....457
5.9.3 Signale....457
5.9.3.1 Signale von NC....457
5.9.3.2 Signale von BAG....457
5.9.3.3 Signale von Kanal....457
5.9.3.4 Signale von Achse/Spindel....457

6 K5: BAGs, Kanäle, Achstausch....459

6.1 Kurzbeschreibung....459

6.2 Betriebsartengruppen (BAGs) - nur 840D sl....461

6.3 Kanäle - nur 840D sl....462
6.3.1 Kanalsynchronisation (Programmkoordination)....462
6.3.2 Bedingtes Warten im Bahnsteuerbetrieb WAITMC....465

6.4 Achs-/Spindeltausch....470
6.4.1 Einführung....470
6.4.2 Beispiel eines Achstausches....472
6.4.3 Achstausch Möglichkeiten....473
6.4.4 Achstauschverhalten NC-Programm....475
6.4.5 Achse in den neutralen Zustand überführen (abgeben)....476
6.4.6 Achse oder Spindel im Teileprogramm übernehmen....477
6.4.7 Automatischer Achstausch....478
6.4.8 Achstausch durch PLC....480
6.4.9 Achstauschverhalten veränderbar einstellen....483
6.4.10 Achstausch über Achs-Containerdrehung....484
6.4.11 Achstausch mit und ohne Vorlaufstopp....485
6.4.12 Ausschließlich PLC-kontrollierte Achse und fest zugeordnete PLC-Achse....486
6.4.13 Geometrieachse im gedrehten WKS und Achstausch....488
6.4.14 Achstausch aus Synchronaktionen....489

6.5 Randbedingungen....491

6.6 Datenlisten....493
6.6.1 Maschinendaten....493
6.6.1.1 Allgemeine Maschinendaten....493
6.6.1.2 Kanal-spezifische Maschinendaten....493
6.6.1.3 Achs-/Spindel-spezifische Maschinendaten....495
6.6.2 Settingdaten....496
6.6.2.1 Kanal-spezifische Settingdaten....496
6.6.3 Signale....496
6.6.3.1 Signale an/von BAG....496
6.6.3.2 Signale an/von Kanal....496

7 M1: Kinematische Transformation....497

7.1 Kurzbeschreibung....497
7.1.1 TRANSMIT (Option)....497
7.1.2 TRACYL (Option)....498

7.1.3 TRAANG (Option) ... 500
7.1.4 Verkettete Transformationen ... 500
7.1.5 Transformations-MD über Teileprogramm/Softkey wirksam setzen ... 501

7.2 TRANSMIT (Option) ... 502
7.2.1 Voraussetzungen für TRANSMIT ... 503
7.2.2 TRANSMIT-spezifische Einstellungen ... 506
7.2.3 Aktivieren von TRANSMIT ... 510
7.2.4 Ausschalten der TRANSMIT-Funktion ... 510
7.2.5 Besondere Reaktionen bei TRANSMIT ... 510
7.2.6 Bearbeitungsmöglichkeiten für TRANSMIT ... 514
7.2.7 Arbeitsraumbegrenzungen ... 520
7.2.8 Überlagerte Bewegungen bei TRANSMIT ... 521
7.2.9 Kontrolle bei Rundachsdrehungen über 360 Grad ... 521
7.2.10 Randbedingungen ... 522

7.3 TRACYL (Option) ... 524
7.3.1 Voraussetzungen für TRACYL ... 526
7.3.2 TRACYL-spezifische Einstellungen ... 530
7.3.3 Aktivieren von TRACYL ... 534
7.3.4 Ausschalten der TRACYL-Funktion ... 535
7.3.5 Besondere Reaktionen bei TRACYL ... 535
7.3.6 Jog ... 538

7.4 TRAANG (Option) ... 539
7.4.1 Voraussetzungen für TRAANG (Schräge Achse) ... 540
7.4.2 TRAANG-spezifische Einstellungen ... 543
7.4.3 Aktivieren von TRAANG ... 546
7.4.4 Ausschalten von TRAANG ... 546
7.4.5 Besondere Reaktionen bei TRAANG ... 547
7.4.6 Schräge Achse programmieren (G05, G07) ... 548

7.5 Verkettete Transformationen ... 550
7.5.1 Aktivierung verketteter Transformationen ... 552
7.5.2 Ausschalten der verketteten Transformation ... 552
7.5.3 Besonderheiten für verkettete Transformationen ... 552
7.5.4 Persistente Transformation ... 553
7.5.5 Achspositionen in der Transformationskette ... 558

7.6 Kartesisches PTP-Fahren ... 561
7.6.1 Programmierung der Stellung ... 564
7.6.2 Überlappbereiche der Achswinkel ... 565
7.6.3 Beispiele für Mehrdeutigkeiten in der Stellung ... 566
7.6.4 Beispiel für Mehrdeutigkeit in der Rundachsposition ... 567
7.6.5 PTP/CP-Umschaltung in der Betriebsart JOG ... 567

7.7 Kartesisches Handverfahren (Option) ... 568

7.8 Transformations-MD über Teileprogramm/Softkey wirksam setzen ... 577
7.8.1 Funktionalität ... 577
7.8.2 Randbedingungen ... 578
7.8.3 Steuerungsverhalten bei Power On, Betriebsartenwechsel, Reset, Satzsuchlauf, REPOS ... 579
7.8.4 Liste der betroffenen Maschinendaten ... 580

7.9 Randbedingungen ... 583
7.9.1 Verkettete Transformationen ... 583

7.10 Beispiele ... 583
7.10.1 TRANSMIT ... 583
7.10.2 TRACYL ... 585

7.10.3 TRAANG 590
7.10.4 Verkettete Transformationen 591
7.10.5 Wirksamsetzen von Transformations-MD über Teileprogramm 595
7.10.6 Achspositionen in der Transformationskette 596

7.11 Datenlisten 599
7.11.1 Maschinendaten 599
7.11.1.1 TRANSMIT 599
7.11.1.2 TRACYL 600
7.11.1.3 TRAANG 602
7.11.1.4 Verkettete Transformationen 603
7.11.1.5 Nicht-transformationsspezifische Maschinendaten 603
7.11.2 Signale 603
7.11.2.1 Signale von Kanal 603

8 M5: Messen 605

8.1 Kurzbeschreibung 605

8.2 Hardwarevoraussetzungen 606
8.2.1 Verwendbare Messtaster 606
8.2.2 Messtasteranschluss 608

8.3 Kanalspezifisches Messen 613
8.3.1 Messmodus 613
8.3.2 Messergebnisse 614

8.4 Setzen von Nullpunkten, Werkstückvermessung und Werkzeugvermessung 615
8.4.1 Istwertsetzen und Ankratzen 615
8.4.2 Werkstückvermessung 616
8.4.2.1 Eingangswerte 616
8.4.2.2 Auswahl der Messung 623
8.4.2.3 Ausgangswerte 624
8.4.2.4 Berechnungsmethode 625
8.4.2.5 Maßeinheiten und Meßgrößen für die Berechnung 628
8.4.2.6 Diagnose 629
8.4.3 Messtypen der Werkstückvermessung 630
8.4.3.1 Messen einer Kante (Mess Type 1, 2, 3) 630
8.4.3.2 Messung eines Winkels (Mess Type 4, 5, 6, 7) 634
8.4.3.3 Messen einer Bohrung (Mess Type 8) 637
8.4.3.4 Messen einer Welle (Mess Type 9) 640
8.4.3.5 Messen einer Nut (Mess Type 12) 641
8.4.3.6 Messen eines Steg (Mess Type 13) 644
8.4.3.7 Messen von Geo- und Zusatzachsen (Mess Type 14, 15) 645
8.4.3.8 Messen einer schrägen Kante (Mess Type 16) 647
8.4.3.9 Messen eines schrägen Winkels in einer Ebene (Mess Type 17) 649
8.4.3.10 Messung um ein WKS-Bezugssystem neu definieren (Mess Type 18) 652
8.4.3.11 Messen einer 1-, 2- und 3-dimensionalen Sollwertvorgabe (Mess Type 19, 20, 21) 655
8.4.3.12 Messen eines schrägen Winkel (Mess Type 24) 659
8.4.3.13 Messen eines Rechtecks (Mess Type 25) 663
8.4.3.14 Messung zum Sichern von Datenhaltungsframes (Mess Type 26) 664
8.4.3.15 Messung zum Zurückschreiben gesicherter Datenhaltungsframes (Mess Type 27) 665
8.4.3.16 Messung zur Vorgabe einer additiven Drehung für Kegeldrehen (Mess Type 28) 666
8.4.4 Werkzeugvermessung 667
8.4.5 Messtypen der Werkzeugvermessung 668
8.4.5.1 Messen der Werkzeuglänge (Mess Type 10) 668
8.4.5.2 Messen des Werkzeugdurchmessers (Mess Type 11) 670
8.4.5.3 Messen von Werkzeuglängen mit Lupe (Mess Type 22) 671

8.4.5.4 Messen einer Werkzeuglänge mit gemerkter oder aktueller Position (Mess Type 23) 672
8.4.5.5 Messen einer Werkzeuglänge zweier Werkzeuge mit Orientierung 673

8.5 Messgenauigkeit und Prüfung 685
8.5.1 Messgenauigkeit 685
8.5.2 Messtaster-Funktionsprüfung 685

8.6 Beispiele - nur 840D sl 686
8.6.1 Messmodus 1 686
8.6.2 Messmodus 2 687
8.6.3 Kontinuierliches Messen 687
8.6.3.1 Kontinuierlilches Messen nach Abschluss der programmierten Verfahrbewegung 687
8.6.3.2 Kontinuierliches Messen mit Restweglöschen 688
8.6.3.3 Kontinuierliches Messen modal über mehrere Sätze 688
8.6.4 Funktionsprüfung und Wiederholgenauigkeit 689

8.7 Datenlisten 691
8.7.1 Maschinendaten 691
8.7.1.1 Allgemeine Maschinendaten 691
8.7.1.2 Kanal-spezifische Maschinendaten 691
8.7.2 Systemvariablen 691

9 N3: Softwarenocken, Wegschaltsignale - nur 840D sl 693

9.1 Kurzbeschreibung 693

9.2 Nockensignale und Nockenpositionen 694
9.2.1 Erzeugung der Nockensignale für getrennte Ausgabe 694
9.2.2 Erzeugung der Nockensignale bei verknüpfter Ausgabe 698
9.2.3 Nockenpositionen 702
9.2.4 Vorhalte-/Verzögerungszeit (dynamischer Nocken) 704

9.3 Ausgabe der Nockensignale 706
9.3.1 Aktivierung 706
9.3.2 Ausgabe der Nockensignale an die PLC 706
9.3.3 Ausgabe der Nockensignale im Lageregeltakt an die NCK-Peripherie 707
9.3.4 Timergesteuerte Nockensignalausgabe 708
9.3.5 Unabhängige, timergesteuerte Ausgabe der Nockensignale 710

9.4 Weg-Zeit-Nocken 712

9.5 Randbedingungen 714

9.6 Datenlisten 715
9.6.1 Maschinendaten 715
9.6.1.1 Allgemeine Maschinendaten 715
9.6.2 Settingdaten 715
9.6.2.1 Allgemeine Settingdaten 715
9.6.3 Signale 716
9.6.3.1 Signale an Achse/Spindel 716
9.6.3.2 Signale von Achse/Spindel 716

10 N4: Stanzen und Nibbeln - nur 840D sl 717

10.1 Kurzbeschreibung 717

10.2 Hubsteuerung 717
10.2.1 Allgemeines 717
10.2.2 Schnelle Signale 718
10.2.3 Kriterien für die Hubauslösung 719
10.2.4 Achsstart nach Stanzen 722

10.2.5 Stanz- und nibbelspezifische PLC-Signale 723
10.2.6 Stanz- und nibbelspezifische Reaktionen auf Standard-PLC-Signale 723
10.2.7 Signalüberwachung 724

10.3 Aktivierung und Deaktivierung 725
10.3.1 Sprachbefehle 725
10.3.2 Funktionserweiterungen 729
10.3.3 Kompatibilität zu älteren Systemen 733

10.4 Automatische Wegaufteilung 735
10.4.1 Allgemeines 735
10.4.2 Verhalten bei Bahnachsen 736
10.4.3 Verhalten im Zusammenhang mit Einzelachsen 741

10.5 Drehbares Werkzeug 746
10.5.1 Allgemeines 746
10.5.2 Mitschleppen Ober- und Unterwerkzeug 747
10.5.3 Tangentialsteuerung 748

10.6 Schutzbereiche 753

10.7 Randbedingungen 753

10.8 Beispiele 754
10.8.1 Beispiele zum definierten Nibbelbeginn 754

10.9 Datenlisten 760
10.9.1 Maschinendaten 760
10.9.1.1 Allgemeine Maschinendaten 760
10.9.1.2 Kanal-spezifische Maschinendaten 760
10.9.2 Settingdaten 760
10.9.2.1 Kanal-spezifische Settingdaten 760
10.9.3 Signale 761
10.9.3.1 Signale an Kanal 761
10.9.3.2 Signale von Kanal 761
10.9.4 Sprachbefehle 761

11 P2: Positionierachsen 763

11.1 Kurzbeschreibung 763

11.2 Eigener Kanal, Positionierachse oder konkurrierende Positionierachse 766
11.2.1 Eigener Kanal - nur 840D sl 766
11.2.2 Positionierachse 767
11.2.3 Konkurrierende Positionierachse 769

11.3 Bewegungsverhalten und Interpolationsvorgänge 771
11.3.1 Bahninterpolator und Achsinterpolator 771
11.3.2 Interpolationsverhalten der Bahnachsen bei G0 771
11.3.3 Autarke Einzelachsvorgänge 773
11.3.4 Autarke Einzelachsvorgänge mit NC-geführtem ESR 781

11.4 Geschwindigkeit 783

11.5 Programmierung 784
11.5.1 Allgemein 784
11.5.2 Umdrehungsvorschub bei externer Programmierung 787

11.6 Satzwechsel 788
11.6.1 Satzwechselzeitpunkt einstellbar 791
11.6.2 Bewegungsendekriterien bei Satzsuchlauf 798

11.7 Beeinflussung durch die PLC ... 799
11.7.1 Konkurrierende Positionierachsen von der PLC starten ... 801
11.7.2 PLC kontrollierte Achsen ... 801
11.7.3 Steuerungsverhalten PLC-kontrollierter Achsen ... 803

11.8 Verhalten bei Sonderfunktionen ... 804
11.8.1 Probelaufvorschub (DRY RUN) ... 804
11.8.2 Einzelsatz ... 804

11.9 Beispiele ... 805
11.9.1 Bewegungsverhalten und Interpolationsvorgänge ... 805
11.9.1.1 Bahnachsen fahren bei G0 nicht interpolierend ... 806

11.10 Datenlisten ... 807
11.10.1 Maschinendaten ... 807
11.10.1.1 Kanal-spezifische Maschinendaten ... 807
11.10.1.2 Achs-/Spindel-spezifische Maschinendaten ... 807
11.10.2 Settingdaten ... 807
11.10.2.1 Achs-/Spindel-spezifische Settingdaten ... 807
11.10.3 Signale ... 807
11.10.3.1 Signale an Kanal ... 807
11.10.3.2 Signale von Kanal ... 808
11.10.3.3 Signale an Achse/Spindel ... 808
11.10.3.4 Signale von Achse/Spindel ... 808

12 P5: Pendeln - nur 840D sl ... 809

12.1 Kurzbeschreibung ... 809

12.2 Asynchrones Pendeln ... 811
12.2.1 Einflüsse auf das asynchrone Pendeln ... 812
12.2.2 Asynchrones Pendeln unter PLC-Regie ... 818
12.2.3 Besondere Reaktionen beim asynchronen Pendel ... 819

12.3 Pendeln über Synchronaktionen gesteuert ... 822
12.3.1 Zustellung im Umkehrpunkt 1 oder 2 ... 825
12.3.2 Zustellung im Umkehrbereich ... 826
12.3.3 Zustellung in beiden Umkehrpunkten ... 827
12.3.4 Anhalten der Pendelbewegung im Umkehrpunkt ... 828
12.3.5 Pendelbewegung wieder starten ... 829
12.3.6 Teilzustellung nicht zu früh starten ... 830
12.3.7 Zuordnung von Pendel- und Zustellachse OSCILL ... 831
12.3.8 Festlegung der Zustellungen POSP ... 831
12.3.9 Pendelumkehr von Extern ... 832

12.4 Randbedingungen ... 834

12.5 Beispiele ... 834
12.5.1 Beispiel Asynchrones Pendeln ... 834
12.5.2 Beispiel 1 Pendeln mit Synchronaktionen ... 835
12.5.3 Beispiel 2 Pendeln mit Synchronaktionen ... 838
12.5.4 Beispiele Startposition ... 840
12.5.4.1 Startposition über Sprachbefehl definieren ... 840
12.5.4.2 Pendeln über Settingdaten starten ... 840
12.5.4.3 Satzbezogenes Pendeln (Startposition = Umkehrpunkt 1) ... 841
12.5.5 Beispiel Pendelumkehr von Extern ... 843
12.5.5.1 Umkehrposition mit "Pendelumkehr von Extern" über Synchronaktion ändern ... 843

12.6 Datenlisten ... 844
12.6.1 Maschinendaten ... 844

12.6.1.1 Allgemeine Maschinendaten 844
12.6.2 Settingdaten 844
12.6.2.1 Achs-/Spindel-spezifische Settingdaten 844
12.6.3 Signale 844
12.6.3.1 Signale an Achse/Spindel 844
12.6.3.2 Signale von Achse/Spindel 845
12.6.4 Systemvariablen 845
12.6.4.1 Hauptlaufvariablen für Bewegungssynchronaktionen 845

13 R2: Rundachsen 849

13.1 Kurzbeschreibung 849

13.2 Modulo 360 Grad 854

13.3 Programmierung von Rundachsen 857
13.3.1 Allgemeines 857
13.3.2 Rundachse bei aktiver Modulo-Wandlung (endlos drehende Rundachse) 857
13.3.3 Rundachse ohne Modulo-Wandlung 863
13.3.4 Sonstige Programmiereigenschaften bei Rundachsen 865

13.4 Inbetriebnahme von Rundachsen 866

13.5 Besonderheiten von Rundachsen 868

13.6 Beispiele 869

13.7 Datenlisten 870
13.7.1 Maschinendaten 870
13.7.1.1 Allgemeine Maschinendaten 870
13.7.1.2 Achs-/Spindel-spezifische Maschinendaten 870
13.7.2 Settingdaten 870
13.7.2.1 Allgemeine Settingdaten 870
13.7.2.2 Achs-/Spindel-spezifische Settingdaten 870
13.7.3 Signale 871
13.7.3.1 Signale an Achse/Spindel 871
13.7.3.2 Signale von Achse/Spindel 871

14 S3: Synchronspindel 873

14.1 Kurzbeschreibung 873
14.1.1 Funktion 873
14.1.2 Voraussetzungen 875
14.1.3 Synchronbetrieb 875
14.1.4 Voraussetzungen für Synchronbetrieb 881
14.1.5 Anwahl des Synchronbetriebes vom Teileprogramm 882
14.1.6 Abwahl des Synchronbetriebes vom Teileprogramm 885
14.1.7 Synchronspindelkopplung durch PLC beeinflussen 886
14.1.8 Überwachungen des Synchronbetriebs 889

14.2 Programmierung von Synchronspindelkopplungen 892
14.2.1 Vorbereitende Programmieranweisungen 892
14.2.2 Programmieranweisungen für Ein- und Ausschalten der Kopplung 895
14.2.3 Axiale Systemvariablen für Synchronspindel 897
14.2.4 Automatische An- und Abwahl der Lageregelung 898

14.3 Projektierung eines Synchronspindelpaares über Maschinendaten 900
14.3.1 Projektierung des Verhaltens bei NC-Start 901
14.3.2 Projektierung des Verhaltens bei Reset 902

14.4 Besonderheiten des Synchronbetriebs 903

14.4.1 Allgemeine Besonderheiten des Synchronbetriebs ... 903
14.4.2 Synchronität der Folgespindel wieder herstellen ... 905
14.4.3 Einfluss über PLC-Nahtstelle auf den Synchronbetrieb ... 907
14.4.4 Differenzdrehzahl zwischen Leit- und Folgespindel ... 910
14.4.5 Verhalten der Synchronlaufsignale bei Synchronlaufkorrektur ... 915
14.4.6 Synchronlaufkorrektur ablöschen und NC-Reset ... 915
14.4.7 Besonderheiten bei der Inbetriebnahme der Synchronspindelkopplung ... 916
14.5 Beispiele ... 921
14.6 Datenlisten ... 922
14.6.1 Maschinendaten ... 922
14.6.1.1 NC-spezifische Maschinendaten ... 922
14.6.1.2 Kanal-spezifische Maschinendaten ... 922
14.6.1.3 Achs-/Spindel-spezifische Maschinendaten ... 922
14.6.2 Settingdaten ... 923
14.6.2.1 Kanal-spezifische Settingdaten ... 923
14.6.3 Signale ... 923
14.6.3.1 Signale an Kanal ... 923
14.6.3.2 Signale von Kanal ... 923
14.6.3.3 Signale an Achse/Spindel ... 923
14.6.3.4 Signale von Achse/Spindel ... 924
14.6.4 Systemvariablen ... 924
14.6.4.1 Systemvariablen ... 924
15 S7: Speicherkonfiguration ... 925
15.1 Kurzbeschreibung ... 925
15.2 Speicherorganisation ... 925
15.2.1 Aktives und passives Filesystem ... 925
15.2.2 Neukonfiguration ... 927
15.3 Konfiguration des statischen Anwenderspeichers ... 928
15.3.1 Aufteilung des statischen NC-Speichers ... 928
15.3.2 Inbetriebnahme ... 930
15.4 Konfiguration des dynamischen Anwenderspeichers ... 932
15.4.1 Aufteilung des dynamischen NC-Speichers ... 932
15.4.2 Inbetriebnahme ... 933
15.5 Datenlisten ... 934
15.5.1 Maschinendaten ... 934
15.5.1.1 Allgemeine Maschinendaten ... 934
15.5.1.2 Kanal-spezifische Maschinendaten ... 937
15.5.1.3 Achs-/Spindel-spezifische Maschinendaten ... 939
16 T1: Teilungsachsen ... 941
16.1 Kurzbeschreibung ... 941
16.2 Verfahren von Teilungsachsen ... 942
16.2.1 Allgemeines ... 942
16.2.2 Verfahren von Teilungsachsen in der Betriebsart JOG ... 942
16.2.3 Verfahren von Teilungsachsen in der Betriebsart AUTOMATIK ... 944
16.2.4 Verfahren von Teilungsachsen von PLC ... 945
16.3 Parametrierung der Teilungsachsen ... 946
16.4 Programmierung von Teilungsachsen ... 948
16.5 Äquidistante Teilungsintervalle ... 953

16.5.1 Funktion 953
16.5.2 Hirth-Verzahnung 955
16.5.3 Verhalten der Hirth-Achsen in besonderen Situationen 956
16.5.4 Einschränkungen 957
16.5.5 Geänderte Wirksamkeit von Maschinendaten 958
16.6 Inbetriebnahme von Teilungsachsen 959
16.7 Besonderheiten von Teilungsachsen 962
16.8 Beispiele 963
16.8.1 Beispiele mit äquidistanten Teilungen 963
16.9 Datenlisten 965
16.9.1 Maschinendaten 965
16.9.1.1 Allgemeine Maschinendaten 965
16.9.1.2 Achs-/Spindel-spezifische Maschinendaten 965
16.9.2 Settingdaten 965
16.9.2.1 Allgemeine Settingdaten 965
16.9.3 Signale 966
16.9.3.1 Signale von Achse/Spindel 966
16.9.4 Systemvariablen 966
16.9.4.1 Systemvariablen 966

17 W3: Werkzeugwechsel 967

17.1 Kurzbeschreibung 967
17.2 Werkzeugmagazine und Wechseleinrichtungen 968
17.3 Werkzeugwechselzeiten 968
17.4 Span-zu-Span-Zeit 968
17.5 Ansteuerung des Werkzeugwechsels 969
17.6 Werkzeugwechselpunkt 970
17.7 Randbedingungen 970
17.8 Beispiele 971
17.9 Datenlisten 973
17.9.1 Maschinendaten 973
17.9.1.1 Allgemeine Maschinendaten 973
17.9.1.2 Kanal-spezifische Maschinendaten 973
17.9.1.3 Achs/Spindel-spezifische Maschinendaten 973
17.9.2 Signale 973
17.9.2.1 Signale von Kanal 973

18 W4: Schleifspezifische Werkzeugkorrektur und Überwachungen - nur 840D sl 975

18.1 Kurzbeschreibung 975
18.2 Schleifspezifische Werkzeugkorrektur 976
18.2.1 Struktur der Werkzeugdaten 976
18.2.2 Schneidenspezifische Korrekturdaten 978
18.2.3 Werkzeugspezifische Schleifdaten 981
18.2.4 Schleifwerkzeuge - Beispiele 986
18.3 Online-Werkzeugkorrektur 991
18.3.1 Allgemeines 991
18.3.2 Online-Werkzeugkorrektur schreiben: kontinuierlich 993
18.3.3 Online-Werkzeugkorrektur ein-/ausschalten 996

18.3.4 Beispiel für Online-Werkzeugkorrektur schreiben kontinuierlich ... 997
18.3.5 Online-Werkzeugkorrektur schreiben diskret ... 998
18.3.6 Hinweise zur Online-Korrektur ... 999

18.4 Online-Werkzeugradiuskorrektur ... 1001

18.5 Schleifspezifische Werkzeugüberwachung ... 1002
18.5.1 Allgemeines ... 1002
18.5.2 Geometrieüberwachung ... 1003
18.5.3 Drehzahlüberwachung ... 1004
18.5.4 An-/Abwahl der WZ-Überwachung ... 1005

18.6 Konstante Scheibenumfangsgeschwindigkeit (SUG) ... 1006
18.6.1 Allgemeines ... 1006
18.6.2 An-/Abwahl und Programmierung der SUG, Systemvariable ... 1007
18.6.3 SUG in allen Betriebsarten ... 1008
18.6.4 Programmierbeispiel für SUG ... 1009

18.7 Randbedingungen ... 1011
18.7.1 Werkzeugwechsel mit Online-Werkzeugkorrektur ... 1011

18.8 Datenlisten ... 1011
18.8.1 Maschinendaten ... 1011
18.8.1.1 Allgemeine Maschinendaten ... 1011
18.8.1.2 Kanal-spezifische Maschinendaten ... 1011
18.8.1.3 Achs-/Spindel-spezifische Maschinendaten ... 1011
18.8.2 Signale ... 1012
18.8.2.1 Signale von Achse/Spindel ... 1012

19 Z2: NC/PLC-Nahtstellensignale ... 1013

19.1 Digitale und analoge NCK-Peripherie ... 1013
19.1.1 Signale an NC (DB10) ... 1013
19.1.2 Signale von NC (DB10) ... 1021

19.2 Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme ... 1025
19.2.1 Festgelegte logische Funktionen/Defines ... 1025
19.2.2 Schnittstellen in DB19 für M:N ... 1028
19.2.3 Signale von NC (DB10) ... 1034
19.2.4 Signale von Achse/Spindel (DB31, ...) ... 1035

19.3 Bedienung über PG/PC (B4) ... 1035

19.4 Handfahren und Handradfahren ... 1036
19.4.1 Signale von NC (DB10) ... 1036
19.4.2 Signale an Kanal (DB21, ...) ... 1039
19.4.3 Signale von Kanal (DB21, ...) ... 1044
19.4.4 Signale bei Konturhandrad ... 1050
19.4.5 Signale an Achse/Spindel (DB31, ...) ... 1052
19.4.6 Signale von Achse/Spindel (DB31, ...) ... 1057

19.5 Kompensationen (K3) ... 1061

19.6 BAGs, Kanäle, Achstausch ... 1062
19.6.1 Signale an Achse/Spindel (DB31, ...) ... 1062
19.6.2 Signale von Achse/Spindel (DB31, ...) ... 1062

19.7 Kinematische Transformation ... 1063
19.7.1 Signale von Kanal (DB21, ...) ... 1063

19.8 Messen ... 1064
19.8.1 Signale von NC (DB10) ... 1064

19.8.2 Signale von Achse/Spindel (DB31, ...) 1064

19.9 Softwarenocken, Wegschaltsignale 1065
19.9.1 Signalübersicht 1065
19.9.2 Signale von NC (DB10) 1065
19.9.3 Signale an Achse/Spindel (DB31, ...) 1066
19.9.4 Signale von Achse/Spindel (DB31, ...) 1067

19.10 Stanzen und Nibbeln 1068
19.10.1 Signalübersicht 1068
19.10.2 Signale an Kanal (DB21, ...) 1068
19.10.3 Signale von Kanal (DB21, ...) 1070

19.11 Positionierachsen 1072
19.11.1 Signale an Achse/Spindel (DB31, ...) 1072
19.11.2 Function Call - nur 840D sl 1075

19.12 Pendeln 1076
19.12.1 Signale Achse/Spindel (DB31, ...) 1076
19.12.2 Signale von Achse/Spindel (DB31, ...) 1077

19.13 Rundachsen 1079
19.13.1 Signale an Achse/Spindel (DB31, ...) 1079
19.13.2 Signale von Achse/Spindel (DB31, ...) 1079

19.14 Synchronspindel 1080
19.14.1 Signale an Achse/Spindel (DB31, ...) 1080
19.14.2 Signale von Achse/Spindel (DB31, ...) 1080

19.15 Speicherkonfiguration (S7) 1083

19.16 Teilungsachsen 1083
19.16.1 Signale von Achse/Spindel (DB31, ...) 1083

19.17 Werkzeugwechsel (W3) 1084

19.18 Schleifspezifische Werkzeugkorrektur und Überwachungen 1084
19.18.1 Signale von Achse/Spindel (DB31, ...) 1084

A Anhang 1087

A.1 Liste der Abkürzungen 1087

A.2 Feedback zur Dokumentation 1095

A.3 Dokumentationsübersicht 1097

Glossar 1099

Index 1121

A4: Digitale und analoge NCK-Peripherie

1

1.1 Kurzbeschreibung

Allgemeines

Über die "Digitale und analoge NCK-Peripherie" können Signale im Interpolationstakt gelesen bzw. ausgegeben werden. Mit diesen Signalen können u. a. folgende Funktionen ausgeführt werden:

- Mehrere Vorschubwerte pro Satz
- Mehrere Hilfsfunktionen im Satz
- Schnellrückzug bei Fertigmaß
- Achsspezifisches Restweglöschen
- Programmverzweigungen
- Schneller NC-Start
- Analoge Messzange
- Wegschaltsignale
- Stanz-/Nibbelfunktionen
- Analogwertsteuerung

Zugriff auf NCK-Peripherie

- NCK-Peripherie über PLC: für On-Board-/ und PROFIBUS-Peripherie
- NCK-Peripherie über PROFIBUS: für PROFIBUS-Peripherie

Vorteile der Funktion "NCK-Peripherie über PROFIBUS"

In Bezug auf die Funktion "NCK-Peripherie über PLC" ergeben sich für die direkte Kommunikation des NCK mit der PROFIBUS-Peripherie folgende Vorteile:

- Direkter Zugriff auf die PROFIBUS-Peripherie, ohne Umweg über die PLC.
- Das PLC-Betriebssystem wird nicht belastet.
- Lesen ist gleichzeitig von Teileprogramm/Synchronaktionen und Compile-Zyklen aus möglich.
- Die konfigurierten E/A-Bereiche müssen nicht lückenlos aufeinander folgen.

1.2 NCK-Peripherie über PLC

1.2.1 Allgemeine Funktionalität

Allgemein

Zum Steuern oder Beeinflussen von zeitkritischen NC-Funktionen werden schnelle NCK-Peripherie-Schnittstellen bzw. eine schnelle Ansprache bestimmter PLC-E/As benötigt.

Folgendes ist daher bei SINUMERIK 840D, 840Di möglich:

- Einsatz von digitalen und analogen NCK-Ein-/Ausgängen
- Direkte Ansprache bestimmter PLC-E/As

Die Hardware-Ein-/Ausgänge können via Systemvariablen über das Teileprogramm oder Synchronaktionen gelesen oder beschrieben werden.

Über die PLC-Nahtstelle können sowohl die Signalzustände der digitalen E/A-Peripherie als auch die Werte der externen analogen E/A-Peripherie vom PLC-Anwenderprogramm anwendungsspezifisch verändert werden.

Hardware 840D

Bei SINUMERIK 840D befinden sich auf der **NCU On-Board** jeweils 4 digitale NCK-Eingänge (Eingang 1 bis 4) und 4 digitale NCK-Ausgänge (Ausgang 1 bis 4).

Diese je 4 digitalen On-Board-Ein-/Ausgänge liegen im ersten Adressbyte. Bei den NCK-Ausgängen können die verbleibenden Signale dieses Bytes (NCK-Ausgänge 5 - 8) über die PLC-Nahtstelle genutzt werden (digitale NCK-Ausgänge ohne Hardware).

Mit Hilfe des am Antriebsbus ankoppelbaren "NCU-Terminalblock" können weitere digitale NCK-Ein-/Ausgänge sowie analoge NCK-Ein-/Ausgänge angeschlossen werden (nachfolgend als **externe NCK-E/A-Peripherie** bezeichnet).

Der "NCU-Terminalblock" dient als Trägerbaugruppe für maximal 8 DP-Kompakt-Aufsteckmodule. Je NCU können bis zu 2 "NCU-Terminalblöcke" angeschlossen werden.

Maximaler Ausbaugrad der externen NCK-E/A-Peripherie:

- 32 digitale NCK-Eingänge (digitaler Eingang 9 bis 40)
- 32 digitale NCK-Ausgänge (digitaler Ausgang 9 bis 40)
- 8 analoge NCK-Eingänge (analoger Eingang 1 bis 8)
- 8 analoge NCK-Ausgänge (analoger Ausgang 1 bis 8)

Weitere Informationen bzgl. Spezifikation der Hardware siehe:

Literatur:

/PHD/ SINUMERIK 840D Handbuch Projektierung (HW)

Hardware 840 Di digitale I/Os

Für die SINUMERIK 840Di stehen über die Baugruppe "MCI-Board-Extension" digitale Ein-/Ausgänge zur Verfügung. Es sind folgende Anschlüsse vorhanden:

- 2 Handräder
- 2 Messtaster
- 4 digitale Ein-/Ausgänge

Hinweis

Die MCI-Board-Extension-Baugruppe ist eine Option für die SINUMERIK 840Di. Die PIN-Belegung der Kabelverteiler-Schnittstelle (X121) entspricht der des Kabelverteilers von SINUMERIK 840D.

SINUMERIK 840Di analoge u. digitale Ein-/Ausgänge

Analoge und digitale Ein-/Ausgänge können an der SINUMERIK 840Di mittels SIMATIC-S7-Busanschaltungs- und Signalbaugruppen über den PROFIBUS-DP betrieben werden.

PLC-Peripherie direkt von NCK ansprechbar

Bis zu 16 Byte für digitale Eingabe-Signale und analoge Eingabewerte sowie bis zu 16 Bytes für digitale Ausgabe-Signale und analoge Ausgabewerte können direkt vom Teileprogramm adressiert werden. Die Bytes sind bei der PLC-Projektierung zu berücksichtigen. Die Bytes müssen lückenlos projektiert sein. Sie werden direkt durch das PLC-Betriebssystem bearbeitet. Dadurch liegt die Transportzeit der Signale zwischen NC und den PLC-I/O-Modulen in der Größenordnung von 0,5 ms.

Hinweis

Die für den NCK spezifizierten Ausgangsbytes dürfen nicht vom PLC-Anwenderprogramm schreibend verwendet werden, weil die Zugriffe zwischen NCK und PLC unkoordiniert wären.

Komparator-Eingänge

Zusätzlich zu den digitalen und analogen NCK-Eingängen stehen noch 16 interne Komparatoreingänge (Komparator-Eingangsbyte 1 und 2) zur Verfügung.

Der Signalzustand eines Komparatoreingangs wird durch den Vergleich eines Analog-Eingangssignals mit einem Schwellwert, der in einem Settingdatum steht, gebildet.

Anzahl

Mit allgemeinen Maschinendaten muss jeweils die Anzahl der ansprechbaren digitalen NCK-Ein-/Ausgangsbytes und der analogen Ein-/Ausgänge festgelegt werden.

Maschinendatum	Anzahl der aktiven ...	Max. Anzahl
MD10350 $MN_FASTIO_DIG_NUM_INPUTS	... digitalen NCK-Eingangs**bytes**	5
MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS	... digitalen NCK-Ausgangs**bytes**	5
MD10300 $MN_FASTIO_ANA_NUM_INPUTS	... analogen NCK-Eingänge	8
MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS	... analogen NCK-Ausgänge	8

Hinweis

Per Definition wird bei der SINUMERIK 840Di das 1. Byte immer den 4 digitalen E/As der MCI-Board-Extension-Baugruppe zugewiesen. Auch wenn Sie keine MCI-Board-Extension-Baugruppe an die SINUMERIK 840Di angeschlossen haben, wird das 1. Byte immer von dieser Baugruppe belegt.

Aus diesem Grund müssen in den Maschinendaten:

$MN_FASTIO_DIG_NUM...

mindestens 2 Bytes eingetragen werden, wenn Sie über den PROFIBUS weitere E/A-Peripherie betreiben.

Falls Ein-/Ausgänge vom Teileprogramm angesprochen werden, die in diesen Maschinendaten nicht definiert wurden, werden entsprechende Alarme generiert.

Diese NCK-Ein-/Ausgänge müssen hardwaremäßig nicht vorhanden sein.

In diesem Fall werden NCK-intern die Signalzustände bzw. die binären Analogwerte definiert auf "Null" gesetzt. Von der PLC können diese Werte jedoch verändert werden.

Hardwarezuordnung der externen NCK-Peripherie

Für die Zuordnung der E/A-Signalbaugruppen bzw. E/A-Module zur externen NCK-Peripherie gibt es folgende allgemeine Maschinendaten:

MD10366 $MN_HW_ASSIGN_DIG_FASTIN[hw]

(HW-Zuordnung für externe digitale Eingänge)

MD10368 $MN_HW_ASSIGN_DIG_FASTOUT[hw]

(HW-Zuordnung für externe digitale Ausgänge)

MD10362 $MN_HW_ASSIGN_ANA_FASTIN[hw]

(HW-Zuordnung für externe analoge Eingänge)

MD10364 $MN_HW_ASSIGN_ANA_FASTOUT[hw]

(HW-Zuordnung für externe analoge Ausgänge)

[hw]: Index für Adressierung der externen digitalen E/A-Bytes (0 bis 3) bzw. der externen analogen Ein-/Ausgänge (0 bis 7)

Hinweis

Die HW-Zuordnung ist steuerungsspezifisch unterschiedlich zwischen SINUMERIK 840D und 840Di.

Die **Zuordnung** der E/As für die SINUMERIK 840Di ist über die Maschinendaten:

MD10362 bis MD10368

mit folgenden Defaultwerten festgelegt:

Maschinendatum	Bedeutung	Default
MD10362 $MN_HW_ASSIGN_ANA_FASTIN[0] ...	Zuordnung für Analog-Eingang (16-Bit-Zugriff)	050000A0
MD10364 $MN_HW_ASSIGN_ANA_FASTOUT[0] ...	Zuordnung für Analog-Ausgang (16-Bit-Zugriff)	050000A0
MD10366 $MN_HW_ASSIGN_DIG_FASTIN[0] ...	Zuordnung für Digital-Eingang (8-Bit-Zugriff)	05000090
MD10368 $MN_HW_ASSIGN_DIG_FASTOUT[0] ...	Zuordnung für Digital-Ausgang (8-Bit-Zugriff)	05000090

Änderung MD für PROFIBUS-DP

Die folgenden Maschinendaten sind für den Betrieb der Hardware am PROFIBUS-DP der SINUMERIK 840Di neu spezifiziert:

$MN_HW_ASSIGN_...

Die Zuweisung vom Byte 1 bis Byte 4 ist neu festgelegt worden. Für den PROFIBUS-DP-Betrieb gilt folgende Belegung dieser Maschinendaten:

Byte	Neu für PROFIBUS-DP	Bisherige Bedeutung
4.Byte	Segment-Nummer = 5	Segment-Nummer
3.Byte	nicht verwendet = 0	Modul-Nummer
2.Byte	logische Adresse high	Submodul-Nummer
1.Byte	logische Adresse low	Ein-/Ausgangs-Nummer

Richtlinien für die Maschinendaten $MN_HW_ASSIGN_...:

- Logische Adresse im 1. und 2. Byte wird hexadezimal angegeben.
  Beispiel: 050001A2 (Hex) entspricht logischer Adresse 418 (Dec)
- Adresse 0 ist für die PLC reserviert und kann nicht als NC-E/A genutzt werden.
- Der Wert 05000000 im Maschinendatum:
- $MN_HW_ASSIGN_...
- gilt als Signal "Slot physikalisch nicht vorhanden".
- Der Eingang wird dann als Simulationseingang betrachtet.

Systemvariable

In der nachfolgenden Tabelle sind die Systemvariablen aufgelistet, mit denen die NCK-E/A-Peripherie vom Teileprogramm direkt gelesen bzw. beschrieben werden kann.

Adressiert wird mit der Nummer des NCK-Ein-/Ausgangs.

Für n gilt:

1 ≤ n ≤ 8 * MD10350 $MN_FASTIO_DIG_NUM_INPUTS

1 ≤ n ≤ 8 * MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS

1 ≤ n ≤ MD10300 $MN_FASTIO_ANA_NUM_INPUTS

1 ≤ n ≤ MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS

Systemvariable	Bedeutung	Bereich v. [n]
$A_IN[n]	Digitalen NCK-Eingang [n] lesen	1 bis 3, 9 bis 40
$A_INA[n]	Analogen NCK-Eingang [n] lesen	1 bis 8
$A_INCO[n]	Komparator-Eingang [n] lesen	1 bis 16
PBB		
$A_OUT[n]	Digitalen NCK-Ausgang [n] schreiben/lesen	1 bis 40
$A_OUTA[n]	Analogen NCK-Ausgang [n] schreiben/lesen	1 bis 8

Hinweis

Beim Lesen dieser Systemvariablen vom Teileprogramm wird steuerungsintern Vorlaufstopp (`STOPRE`-Befehl) ausgelöst.

Bewertungsfaktor

Mit den Bewertungsfaktoren in den folgenden, allgemeinen Maschinendaten kann für jeden einzelnen analogen NCK-Ein-/Ausgang eine Anpassung an die AD- bzw. DA-Wandler der verwendeten Analog-Peripheriebaugruppe vorgenommen werden.:

MD10320 $MN_FASTIO_ANA_INPUT_WEIGHT[hw]

MD10330 $MN_FASTIO_ANA_OUTPUT_WEIGHT[hw]

Durch Vorgabe des richtigen Bewertungsfaktors erzeugt der mit der Systemvariable $A_OUTA[n] vorgegebene Wert am Analogausgang [n] den entsprechenden Spannungswert in Millivolt.

Beispiel für 840D

Analogwertbereich ist 10 V (maximale Aussteuerung);

MD10330 $MN_FASTIO_ANA_OUTPUT_WEIGHT[hw] = 10000

(Standardwert bei 840D)

$A_OUTA[1] = 9500 ; am analogen NCK-Ausgang 1 wird 9,5 V ausgegeben

$A_OUTA[3] = -4120 ; am analogen NCK-Ausgang 3 wird -4,12 V ausgegeben

Anwendung bei analogen NCK-Ein-/Ausgängen ohne Hardware:

Bei einem Bewertungsfaktor = 32767 sind die digitalisierten Analogwerte für Teileprogramm- und PLC-Zugriffe gleich. Somit kann das zugehörige Eingangs- bzw. Ausgangswort für eine 1:1-Kommunikation zwischen Teileprogramm und PLC verwendet werden.

Zuordnung zu NC-Funktionen

Bei mehreren NC-Funktionen wird die Funktionalität der NCK-E/A-Peripherie vorausgesetzt.

Die Zuordnung der für diese NC-Funktionen verwendeten NCK-Ein-/ Ausgänge

erfolgt funktionsspezifisch über Maschinendaten, z. B.:

MD21220 $MC_MULTFEED_ASSIGN_FASTIN (Mehrere Vorschübe in einem Satz).

Für die digitalen Ein-/Ausgänge ist in den Maschinendaten eine Byte-Adresse anzugeben. Die Zuordnung ist hierbei immer byteweise.

Byte-Adresse	Zuordnung für die digitalen NCK-Ein-/Ausgänge			
0	keine			
1	1 bis 4 (Onboard-E/A)	und	5 bis 8	(NCK-A ohne Hardware)
2	9	bis	16	(externe NCK-Peripherie)
3	17	bis	24	(externe NCK-Peripherie)
4	25	bis	32	(externe NCK-Peripherie)
5	33	bis	40	(externe NCK-Peripherie)
128	Eingänge 1 bis 8 von Komparatorbyte 1			
129	Eingänge 9 bis 16 von Komparatorbyte 2			

Taktsynchrone Bearbeitung

Bei SINUMERIK 840D können die E/A-Module der externen NCK-Peripherie wie folgt betrieben werden:

- **Asynchron**

  Die Ein- und Ausgangswerte werden in einem vom Terminalblock vorgegebenen Takt asynchron zu den NC-internen Bearbeitungstakten zur Verfügung gestellt.

- **Synchron**

  Die Ein- und Ausgangswerte werden synchron zu einem einstellbaren NC-internen Bearbeitungstakt zur Verfügung gestellt.

Die Anwahl der Bearbeitungsart erfolgt modulspezifisch mit dem allgemeinen Maschinendatum:

MD10384 $MN_HW_CLOCKED_MODULE_MASK[tb]

[tb] = Index für Terminalblock (0 bis 1)

Bei synchroner Bearbeitung kann zwischen folgenden Taktraten ausgewählt werden

(MD10380 $MN_HW_UPDATE_RATE_FASTIO[tb]):

- Synchrone Ein-/Ausgaben im Lageregeltakt (Standardeinstellung)
- Synchrone Ein-/Ausgaben im Interpolatortakt

Für die getaktete NCK-E/A-Peripherie kann eine Vorhaltezeit in Mikrosekunden mit dem allgemeinen Maschinendatum:

MD10382 $MN_HW_LEAD_TIME_FASTIO[tb]

festgelegt werden.

Damit kann beispielsweise die Wandlungszeit des AD-Wandlers berücksichtigt werden, so dass der digitalisierte Eingangswert zum Taktzeitpunkt verfügbar ist.

Die festgelegte Taktrate bzw. Vorhaltezeit ist für alle taktsynchron betriebenen E/A-Module des mit [tb] adressierten Terminalblocks wirksam.

Überwachungen

Bei SINUMERIK 840D werden für die externe E/A-Peripherie folgende Funktionsüberwachungen vorgenommen:

- Beim Hochlauf:
  - Überprüfung, ob E/A-Modul-Bestückung der Terminalblöcke mit den MD-Zuordnungen übereinstimmt.
- Im zyklischen Betrieb:
  - Lebenszeichenüberwachung im Interpolatortakt
  - Baugruppen-Überwachung im Interpolatortakt
  - Temperatur-Überwachung

Im Fehlerfall wird NC-Ready weggenommen und ein Alarm gemeldet.

Verhalten bei Störungen

Die digitalen und analogen NCK-Ausgänge werden bei Störungen

(z. B. NC-Ready = 0), bei Fehlern in der NCU bzw. bei Spannungsausfall in den sicheren Zustand geschaltet (0 V am Ausgang).

1.2.2 Digitale Ein-/Ausgänge des NCK

1.2.2.1 Digitale Eingänge des NCK

Anzahl

Mit dem allgemeinen Maschinendatum werden die nutzbaren digitalen NCK-Eingänge (in 8er Gruppen) festgelegt:

MD10350 $MN_FASTIO_DIG_NUM_INPUTS

(Anzahl der aktiven digitalen NCK-Eingangsbytes)

Funktion

Mit Hilfe der digitalen NCK-Eingänge kann z. B. der Programmablauf der Werkstückbearbeitung durch externe Signale beeinflusst werden.

Mit der Systemvariablen **$A_IN[n]** kann der Signalzustand des digitalen Eingangs [n] direkt im Teileprogramm abgefragt werden.

Der am Hardware-Eingang anstehende Signalzustand kann vom PLC-Anwenderprogramm verändert werden (siehe Bild).

Eingang sperren

Vom PLC-Anwenderprogramm können die digitalen NCK-Eingänge mit dem folgenden Nahtstellensignal einzeln gesperrt werden:

DB10 DBB0 bzw. DBB122 ... (Sperre der digitalen NCK-Eingänge)

In diesem Fall werden sie steuerungsintern definiert auf "0" gesetzt.

Eingang von PLC setzen

Darüber hinaus kann von der PLC mit dem folgenden Nahtstellensignal jeder digitale Eingang definiert auf 1-Signal gesetzt werden (siehe Bild):
DB10 DBB1 bzw. DBB123 ... (Setzen von PLC der digitalen NCK-Eingänge)

Sobald dieses Nahtstellensignal auf "1" gesetzt wird, ist der am Hardware-Eingang anstehende Signalzustand bzw. das Sperren des Eingangs wirkungslos.

Istwert lesen

Der Signalzustand der digitalen NCK-Eingänge wird zur PLC gemeldet:

DB10 DBB60 bzw. DBB186 ... (Istwert der digitalen NCK-Eingänge)

Der Istwert spiegelt dabei den tatsächlichen Zustand des Signals am Hardware-Eingang wieder. Der Einfluss der PLC bleibt somit beim "Istwert" unberücksichtigt (siehe Bild).

RESET-/POWER ON-Verhalten

Nach POWER ON und RESET wird der am jeweiligen Eingang anliegende Signalpegel weitergereicht. Bei Bedarf können vom PLC-Anwenderprogramm die einzelnen Eingänge wie oben beschrieben gesperrt bzw. definiert auf "1" gesetzt werden.

Anwendungen

Abhängig vom Signalzustand eines externen Hardware-Signals kann im Teileprogramm der Programmablauf mit bedingten Sprunganweisungen beeinflusst werden.

Beispielsweise werden für folgende NC-Funktionen digitale NCK-Eingänge verwendet:

- Restweglöschen bei Positionierachsen
- Schnelle Programmverzweigungen am Satzende
- Programmierte Einlesesperre
- Mehrere Vorschübe in einem Satz

Literatur:
/FBSY/ Funktionshandbuch Synchronaktionen

Die Zuordnung der NCK-Eingänge zu den NC-Funktionen erfolgt funktionsspezifisch und byteweise über Maschinendaten. Mehrfachzuordnungen von Eingängen werden dabei nicht überwacht.

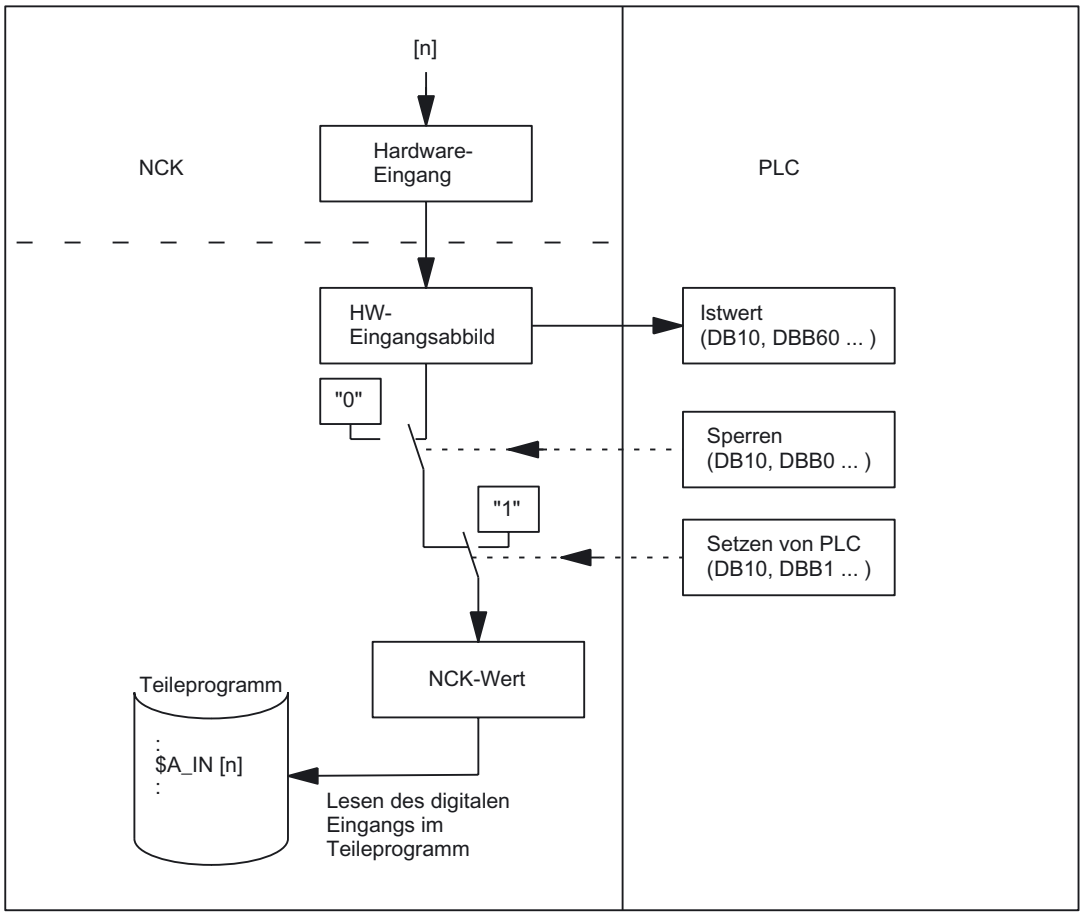


Bild 1-1 Signalfluss für die digitalen NCK-Eingänge

1.2.2.2 Digitale Ausgänge des NCK

Anzahl

Mit dem folgenden, allgemeinen Maschinendatum werden die nutzbaren digitalen NCK-Ausgänge (in 8er-Gruppen) festgelegt (Anzahl der aktiven digitalen NCK-Ausgangsbytes) :

MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS

Funktion

Mit Hilfe der digitalen NCK-Ausgänge besteht die Möglichkeit, abhängig vom Bearbeitungszustand notwendige Schaltbefehle schnell auszugeben.

Mit der Systemvariablen **$A_OUT[n]** kann der Signalzustand des digitalen Ausgangs [n] im Teileprogramm direkt gesetzt bzw. auch wieder gelesen werden.

Zusätzlich bestehen mehrere Möglichkeiten, diesen gesetzten Signalzustand über die PLC zu verändern (siehe Bild).

Ausgang sperren

Vom PLC-Anwenderprogramm können die digitalen NCK-Ausgänge mit dem Nahtstellensignal:
DB10 DBB4 bzw. DBB130 ... (Sperre der digitalen NCK-Ausgänge)
einzeln gesperrt werden. In diesem Fall wird am Hardware-Ausgang "0-Signal" ausgegeben (siehe Bild).

Überschreibmaske

Jeder einzelne vom NC-Teileprogramm setzbare Ausgang kann mit Hilfe der Überschreibmaske von der PLC **überschrieben** werden. Der bisherige 'NCK-Wert' geht dadurch verloren (siehe Bild).

Für das Überschreiben des NCK-Wertes mit der PLC ist folgender Ablauf erforderlich:

1. An der PLC-Nahtstelleist der betroffene Ausgang mit dem gewünschten Signalzustand vorzubesetzen. :
   DB10 DBB6 bzw. DBB132 ... (Setzwert von PLC der digitalen NCK-Ausgänge)

2. Mit Aktivierung der Überschreibmaske (Flankenwechsel 0 → 1) für den betroffenen Ausgang (DB10 DBB5 bzw. DBB131 ...) wird der 'Setzwert' zum neuen 'NCK-Wert'. Dieser Wert bleibt bis zur nächsten Programmierung (von der PLC oder NC-Teileprogramm) wirksam.

Vorgabemaske

Des Weiteren kann von der PLC für jeden Ausgang festgelegt werden, ob der momentane 'NCK-Wert' (z. B. vom NC-Teileprogramm vorgegeben) oder der über die Vorgabemaske (DB10 DBB7 bzw. DBB133 ...) vorgegebene 'PLC-Wert' an den Hardware-Ausgang ausgegeben werden soll (siehe Bild).

Für die Vorgabe des 'PLC-Wertes' ist folgender Ablauf erforderlich:

1. An der PLC-Nahtstelle:
   DB10 DBB6 (Setzwert von PLC der digitalen NCK-Ausgänge)
   ist der betroffene Ausgang mit dem gewünschten Signalzustand vorzubesetzen.
2. Für den betroffenen Ausgang ist die Vorgabemaske auf "1" zu setzen.

Im Gegensatz zur Überschreibmaske geht bei der Vorgabemaske der NCK-Wert nicht verloren. Sobald von der PLC bei der entsprechenden Vorgabemaske "0" vorgegeben wird, wirkt wieder der NCK-Wert.

Hinweis

Für die Überschreibmaske und Vorgabemaske wird an der PLC-Nahtstelle der gleiche Setzwert (DB10 DBB6) verwendet. Deshalb bewirkt ein **gleichzeitiges** Verändern des Signalzustandes über die Überschreibmaske **und** Vorgabemaske stets einen identischen Signalzustand des Ausgangssignals!

Sollwert lesen

Der momentane 'NCK-Wert' der digitalen Ausgänge kann vom PLC-Anwenderprogramm gelesen werden:

DB10 DBB64 bzw. DBB186 ... (Sollwert der digitale NCK-Ausgänge).

Dabei ist zu beachten, dass dieser Sollwert das Sperren bzw. die Vorgabemaske von PLC nicht berücksichtigt. Somit kann sich der Sollwert vom tatsächlichen Signalzustand am Hardware-Ausgang unterscheiden (siehe Bild).

RESET/Programmende

Vom PLC-Anwenderprogramm kann mit Hilfe der Überschreibmaske, Vorgabemaske oder des Sperrsignals jeder digitale Ausgang bei Programmende bzw. bei RESET entsprechend den Anforderungen definiert gesetzt werden.

POWER ON

Nach POWER ON werden die digitalen Ausgänge definiert auf "0" gesetzt. Dies kann vom PLC-Anwenderprogramm mit Hilfe der o. g. Masken anwendungsspezifisch überschrieben werden.

Digitale NCK-Ausgänge ohne Hardware

Werden vom Teileprogramm digitale NCK-Ausgänge beschrieben, die in dem folgenden, allgemeinen Maschinendatum definiert wurden, jedoch als Hardware nicht vorhanden sind, erfolgt keine Alarmmeldung. Der NCK-Wert kann von PLC gelesen werden (NST "Sollwert ..."). :

MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS

Anwendungen

Mit dieser Funktion können digitale Hardware-Ausgänge unter Umgehung von PLC-Zykluszeiten sehr schnell gesetzt werden. Somit lassen sich zeitkritische Schaltvorgänge bearbeitungsnah und programmgesteuert (z. B. mit dem Satzwechsel) auslösen.

Beispielsweise sind für folgende NC-Funktionen digitale NCK-Ausgänge erforderlich:

- Wegschaltsignale

  Literatur:
  /FB2/ Funktionshandbuch Erweiterungsfunktionen; Softwarenocken, Wegschaltsignale (N3)

- Ausgabe der Komparatorsignale

Die Zuordnung der NCK-Ausgänge zu diesen NC-Funktionen erfolgt funktionsspezifisch über Maschinendaten. Mehrfachzuordnungen von Ausgängen werden im Hochlauf überprüft und mit einem Alarm angezeigt.

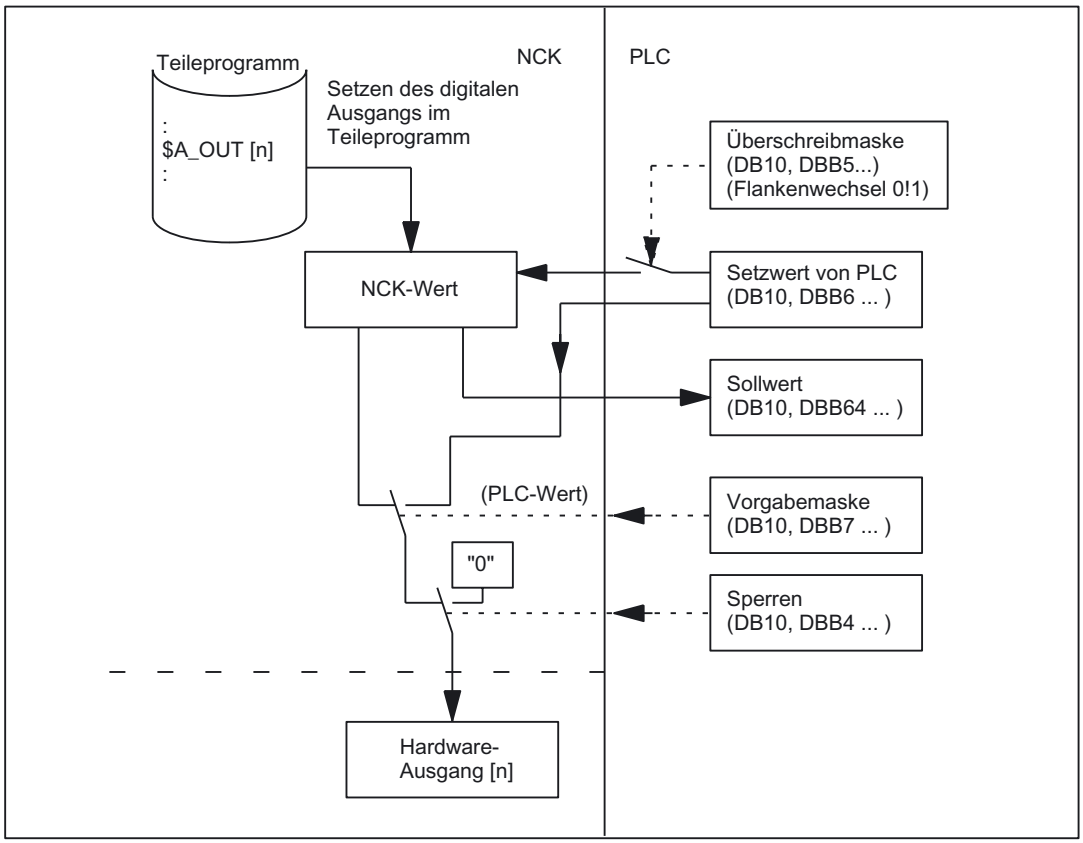


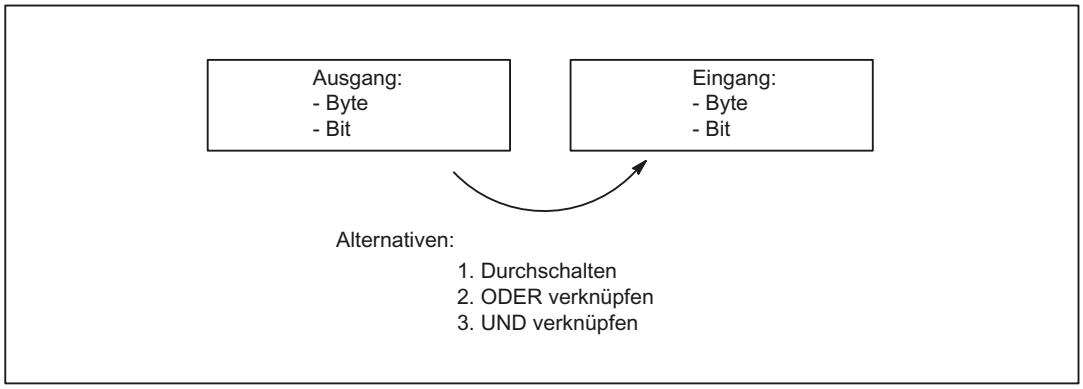
Bild 1-2 Signalfluss für die digitalen Ausgänge der NCK

1.2.3 Durchschalten und Verknüpfen von schnellen NCK-Ein-/Ausgängen

Funktion

Schnelle Eingänge der NCK-Peripherie können abhängig von Signalzuständen schneller Ausgänge softwaremäßig gesetzt werden.

Übersicht:



Durchschalten

Der schnelle Eingang der NCK-Peripherie wird auf den Signalzustand gesetzt, den der zugeordnete schnelle Ausgang hat.

ODER-Verknüpfung

Der schnelle Eingang der NCK-Peripherie nimmt den Signalzustand ein, der sich aus der ODER-Verknüpfung des Ausgangssignals mit dem zugeordneten Eingangssignal ergibt.

UND-Verknüpfung

Der schnelle Eingang der NCK-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 NCK-Peripherie erfolgt anhand der Einträge in den beiden folgenden Maschinendaten:

  MD10350 $MN_FASTIO_DIG_NUM_INPUTS

  MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS.

Zuordnungen definieren

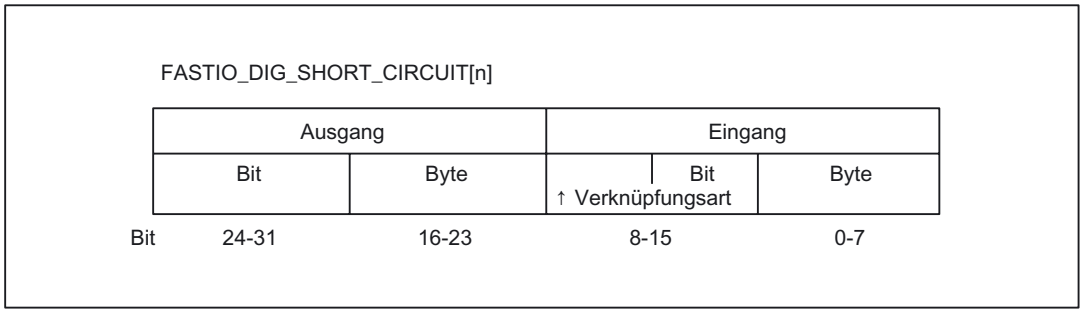
Die Zuordnungen werden angegeben durch das Maschinendatum:
MD10361 $MN_FASTIO_DIG_SHORT_CIRCUIT[n]

n: kann Werte 0 bis 9 annehmen, es sind also bis zu **10** Zuordnungen angebbar.

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

1.2.4 Analoge Ein-/Ausgänge des NCK

1.2.4.1 Analoge Eingänge des NCK

Anzahl

Mit dem folgenden allgemeinen Maschinendatum
werden die nutzbaren analogen NCK-Eingänge festgelegt.:

MD10300 $MN_FASTIO_ANA_NUM_INPUTS(Anzahl der analogen NCK-Eingänge)

Funktion

Mit der Systemvariablen **$A_INA[n]** kann direkt im Teileprogramm auf den Wert des analogen NCK-Eingangs [n] zugegriffen werden.

Zusätzlich kann der am Hardware-Eingang anstehende Analogwert vom PLC-Anwenderprogramm beeinflusst werden (siehe Bild).

Eingang sperren

Vom PLC-Anwenderprogramm können die analogen NCK-Eingänge mit dem Nahtstellensignal:
DB10 DBB146 (Sperre der analogen NCK-Eingänge)
einzeln gesperrt werden.
In diesem Fall werden sie steuerungsintern definiert auf "0" gesetzt.

Eingang von PLC setzen

Darüber hinaus kann von der PLC mit dem Nahtstellensignal:
DB10 DBB147 (Vorgabemaske der analogen NCK-Eingänge)
für jeden analogen NCK-Eingang ein Wert vorgegeben werden (siehe Bild).

Sobald dieses Nahtstellensignal auf "1" gesetzt ist, wird für den entsprechenden Analogeingang der von der PLC vorgegebene Setzwert (DB10 DBB148-163) wirksam. Damit ist der am Hardware-Eingang anstehende Analogwert bzw. das Sperren des Analogeingangs wirkungslos.

Istwert lesen

Über das folgende Nahtstellensignal werden die an den Hardware-Eingängen tatsächlich anstehenden Analogwerte zur PLC übertragen:
DB10 DBB194-209 (Istwert des analogen Eingangs der NCK)
Der mögliche Einfluss von der PLC bleibt somit beim Istwert unberücksichtigt (siehe Bild).

RESET-/POWER ON-Verhalten

Nach `POWER ON` und `RESET` wird der am jeweiligen Eingang anliegende Analogwert weitergereicht. Bei Bedarf können über das PLC-Anwenderprogramm die einzelnen analogen NCK-Eingänge, wie oben beschrieben, manipuliert werden.

Bewertungsfaktor

Mit dem Bewertungsfaktor im allgemeinen Maschinendatum kann für die analogen NCK-Eingänge eine Anpassung an die hardwaremäßig verschiedenen AD-Wandler für das Lesen im Teileprogramm vorgenommen werden (siehe Bild):

MD10320 $MN_FASTIO_ANA_INPUT_WEIGHT[hw]

In dieses Maschinendatum ist der Wert x einzutragen, der im Teileprogramm mit der Systemvariablen x = $A_INA[n] gelesen werden soll, wenn der zugehörige Analogeingang [n] maximal ausgesteuert wird, bzw. über die PLC-Nahtstelle für diesen Eingang der Wert 32767 vorgegeben wird. Dann wird der am Analogeingang anliegende Spannungswert mit der Systemvariablen $A_INA[n] als Zahlenwert mit der Einheit Millivolt gelesen.

Binäre Analogwertdarstellung

Siehe Thema "Analogwertdarstellung der Analog-Ein-/Ausgabewerte des NCK".

Analoger NCK-Eingang ohne Hardware

Bei Teileprogrammzugriffen auf analoge NCK-Eingänge, die über folgendes Maschinendatum definiert wurden, jedoch als Hardware-Eingänge nicht vorhanden sind, werden folgende Werte gelesen:

MD10300 $MN_FASTIO_ANA_NUM_INPUTS

- Der von PLC vorgegebene Sollwert,
  falls von der PLC das NST "Setzwert von PLC der analogen NCK-Eingänge"
  auf 1-Signal gesetzt ist (siehe Bild).
- Sonst 0 Volt.

Damit besteht die Möglichkeit, die Funktionalität der analogen NCK-Eingänge vom PLC-Anwenderprogramm ohne Peripherie-Hardware zu nutzen.

Anwendungen

Die analogen NCK-Eingänge werden insbesondere bei Schleif- und Lasermaschinen angewendet (z. B. bei der NC-Funktion "Analoge Messzange").

Schnelle analoge NCK-Eingänge

Die schnellen analogen Eingänge müssen taktsynchron betrieben werden.
Die Zuordnung wird durch das Maschinendatum angegeben:

MD10384 $MN_HW_CLOCKED_MODULE_MASK

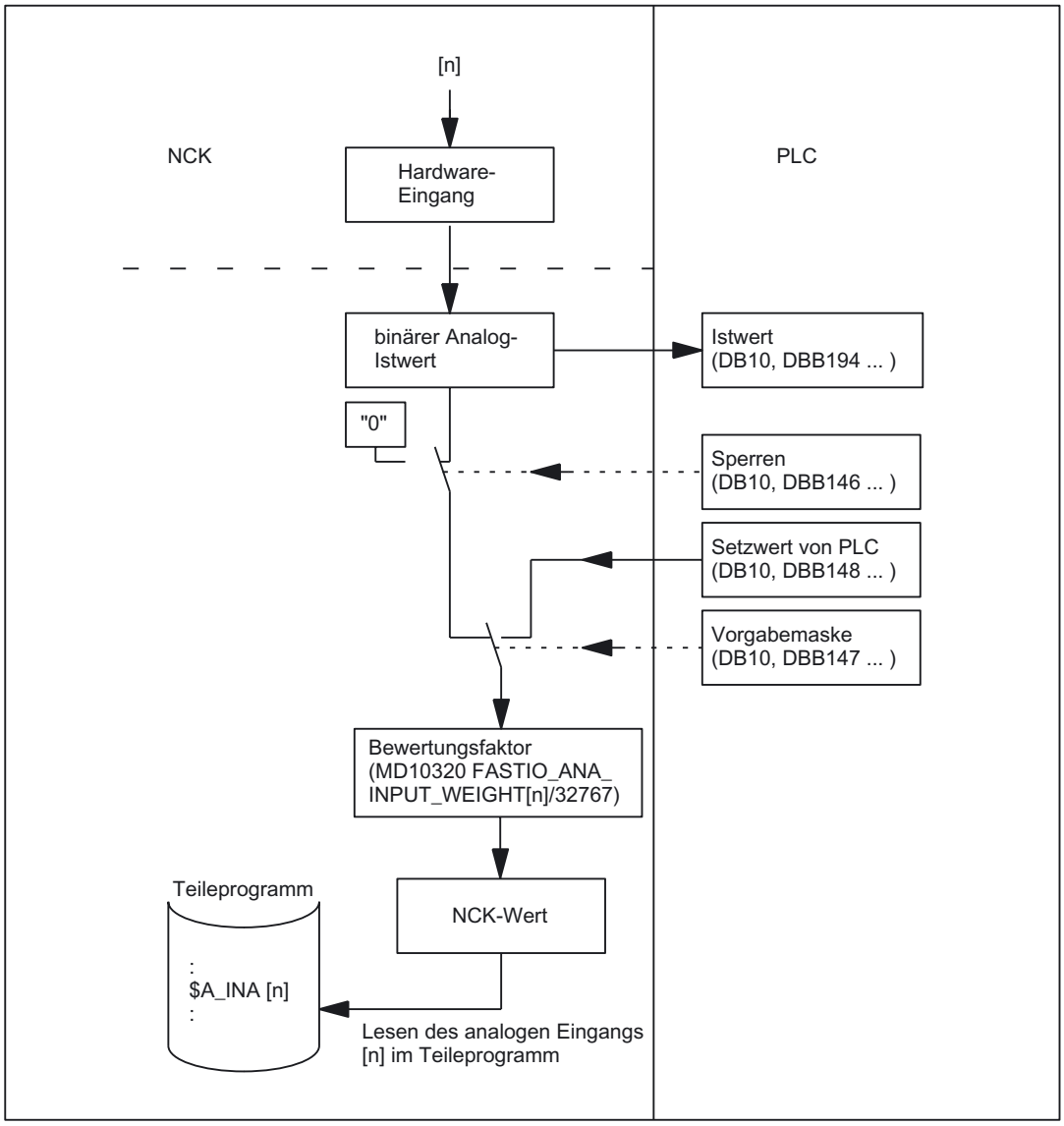


Bild 1-3 Signalfluss für die analogen NCK-Eingänge

1.2.4.2 Analoge Ausgänge des NCK

Anzahl

Mit dem allgemeinen Maschinendatum:
MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS
(Anzahl der analogen NCK-Ausgänge)
werden die nutzbaren analogen NCK-Ausgänge festgelegt.

Funktion

Mit der Systemvariablen **$A_OUTA[n]** kann der Wert des analogen Ausgangs [n] im Teileprogramm direkt vorgegeben werden.

Bevor die Ausgabe an die Hardware-Peripherie erfolgt, ist der von der NCK vorgegebene Analogwert von der PLC veränderbar (siehe Bild).

Ausgang sperren

Vom PLC-Anwenderprogramm können die analogen NCK-Ausgänge über das Nahtstellensignal:
DB10 DBB168 (Sperre der analogen NCK-Ausgänge)
einzeln gesperrt werden.
In diesem Fall wird am Analogausgang **0 Volt** ausgegeben (siehe Bild).

Überschreibmaske

Jeder einzelne vom NC-Teileprogramm vorgegebene NCK-Analogwert kann mit Hilfe der Überschreibmaske von der PLC **überschrieben** werden. Der bisherige 'NCK-Wert' geht dadurch verloren (siehe Bild).

Für das Überschreiben des NCK-Wertes von der PLC ist folgender Ablauf erforderlich:

1. An der PLC-Nahtstelle:
   DB10 DBB170-185 (Setzwert von PLC für analogen Ausgang n der NCK)
   ist der betroffene Ausgang n mit dem gewünschten Analogwert vorzubesetzen.
2. Mit Aktivierung der Überschreibmaske (Flankenwechsel 0 → 1) für den betroffenen Analogausgang (DB10 DBB166) wird der 'Setzwert von PLC' zum neuen 'NCK-Wert'.

Dieser Wert behält seine Gültigkeit bis z. B. vom Teileprogramm der NCK ein neuer Analogwert vorgegeben wird.

Vorgabemaske

Des Weiteren kann von der PLC für jeden Ausgang festgelegt werden, ob der momentane 'NCK-Wert' (z. B. vom NC-Teileprogramm vorgegeben) oder der über die Vorgabemaske (DB10 DBB167) vorgegebene 'PLC-Wert' an den Hardware-Analogausgang ausgegeben werden soll (siehe Bild).

Für die Vorgabe des 'PLC-Wertes' ist folgender Ablauf erforderlich:

1. An der PLC-Nahtstelle:
   DB10 DBB170-185 (Setzwert von PLC für analogen Ausgang n der NCK)
   ist der betroffene Ausgang n mit dem gewünschten Analogwert vorzubesetzen.
2. Für den betroffenen Analogausgang ist die Vorgabemaske (DB10 DBB167) auf "1" zu setzen.

Im Gegensatz zur Überschreibmaske geht bei der Vorgabemaske der NCK-Vorgabewert nicht verloren. Sobald von der PLC bei der entsprechenden Vorgabemaske "0" vorgegeben wird, wirkt wieder der NCK-Wert.

Hinweis

Für die Überschreibmaske und Vorgabemaske wird an der PLC-Nahtstelle der gleiche Setzwert (DB10 DBB170-185) verwendet.

Sollwert lesen

Der momentane 'NCK-Wert' der analogen Ausgänge kann vom PLC-Anwenderprogramm gelesen werden:

DB10 DBB210-225 (Sollwert des analogen Ausgangs n der NCK)

Dabei ist zu beachten, dass dieser Sollwert das Sperren bzw. die Vorgabemaske von der PLC nicht berücksichtigt. Somit kann der Sollwert sich vom tatsächlichen Analogwert am Hardware-Ausgang unterscheiden (siehe Bild).

RESET/Programmende

Vom PLC-Anwenderprogramm kann mit Hilfe der Überschreibmaske, Vorgabemaske oder des Sperrsignals jeder analoge Ausgang bei Programmende bzw. bei RESET entsprechend den Anforderungen definiert vorbesetzt werden.

POWER ON

Nach `POWER ON` werden die analogen Ausgänge definiert auf "0" gesetzt. Nach dem Hochlauf kann vom PLC-Anwenderprogramm mit Hilfe der o. g. Masken anwendungsspezifisch ein anderer Wert vorgegeben werden.

Bewertungsfaktor

Mit dem Bewertungsfaktor im allgemeinen Maschinendatum:
MD10330 $MN_FASTIO_ANA_OUTPUT_WEIGHT[hw]
kann für die analogen NCK-Ausgänge eine Anpassung an die hardwaremäßig verschiedenen DA-Wandler für die Programmierung im Teileprogramm vorgenommen werden (siehe Bild).

In dieses Maschinendatum ist der Wert x einzutragen, der bei Programmierung von $A_OUTA[n] = x die maximale Aussteuerung des zugehörigen Analog-Ausgangs [n] bewirken soll bzw. in der PLC-Nahtstelle für diesen Ausgang den Wert 32767 erzeugen soll. Dann erzeugt der mit der Systemvariablen $A_OUTA[n] vorgegebene Wert am Analogausgang den gleichen Spannungswert in Millivolt.

Binäre Analogwertdarstellung

Siehe Thema "Analogwertdarstellung der Analog-Ein-/Ausgabewerte des NCK".

Sonderfall

Falls im Teileprogramm Werte für NCK-Analogausgänge programmiert werden, die zwar im Maschinendatum:
MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS
definiert wurden, jedoch als Hardware nicht vorhanden sind, erfolgt keine Alarmmeldung. Der NCK-Wert kann von PLC gelesen werden (NST "Sollwert ...").

Anwendung

Mit dieser Funktion können Analogwerte unter Umgehung von PLC-Zykluszeiten sehr schnell ausgegeben werden.

Die analogen NCK-Ausgänge werden insbesondere bei Schleif- und Lasermaschinen angewendet.

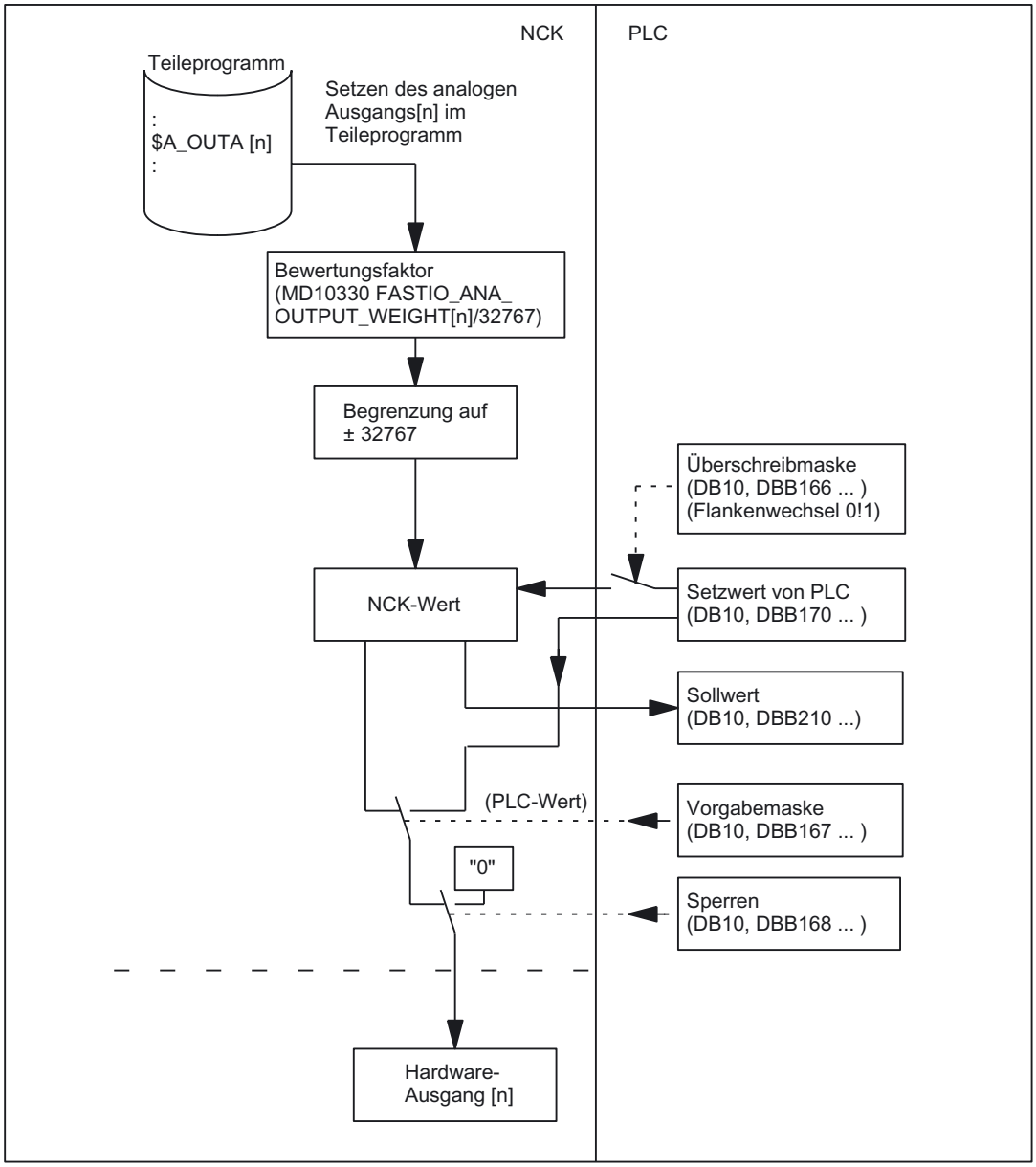


Bild 1-4 Signalfluss für die analogen NCK-Ausgänge

1.2.5 Direkte, von NC adressierbare PLC-Peripherie

Einführung

Der hier vorgestellte **schnelle Datenkanal** zwischen NCK und PLC-Peripherie wird **direkt** durch das PLC-Betriebssystem und damit schnell verarbeitet.

Die Einflussnahme des PLC-Grund- und -Anwendungsprogramms ist nicht vorgesehen. Konkurrierender Zugriff zwischen NCK und PLC auf dieselbe PLC-Peripherie ist nicht sinnvoll und kann zu Störungen führen.

Systemvariablen

Seitens der NC erfolgt der Zugriff über Systemvariablen aus **Teileprogrammen** und **Synchronaktionen**.

Für das **Lesen** von PLC:

$A_PBB_IN[n]	; Byte (8 Bit) lesen
$A_PBW_IN[n]	; Word (16 Bit) lesen
$A_PBD_IN[n]	; DWord (32 Bit) lesen
$A_PBR_IN[n]	; Real (32 Bit float) lesen
n	Byteoffset innerhalb des PLC-Eingabebereiches

Beim Lesen aus dem Teileprogramm erfolgt Vorlaufstopp.

Für das **Schreiben** zur PLC:

$A_PBB_OUT[n]	; Byte (8 Bit) schreiben
$A_PBW_OUT[n]	; Word (16 Bit) schreiben
$A_PBD_OUT[n]	; DWord (32 Bit) schreiben
$A_PBR_OUT[n]	; Real (32 Bit float) schreiben
n	Byteoffset innerhalb des PLC-Ausgabebereiches

Die Ausgabedaten können vom Teileprogramm und von Synchronaktionen auch gelesen werden. Beim Lesen aus dem Teileprogramm wird automatisch ein Vorlaufstopp ausgelöst (um die Synchronisation mit dem Echtzeitkontext zu erreichen).

Variablen-Wertebereiche

Die Variablen können Werte in folgenden Bereichen annehmen:

$A_PBB_OUT[n]	;(-128 ... +127) oder (0 ... 255)
$A_PBW_OUT[n]	;(-32768 ... +32767) oder (0 ... 65535)
$A_PBD_OUT[n]	;(-2147483648 ... +2147483647) oder (0 ... 4294967295)
$A_PBR_OUT[n]	;(-3.402823466E+38 ... +3.402823466E+38)

Übergabezeitpunkte

Die Ausgabe von Daten von NCK ⇒ PLC (Schreiben) erfolgt, wenn wenigstens ein Datum geschrieben wurde, am Ende des Interpolationstaktes.

Das Einlesen von Daten erfolgt, in Abhängigkeit des Maschinendatums durch das Absetzen einer Request-Anforderung am Ende des Interpolationstaktes:

MD10398 $MN_PLCIO_IN_UPDATE_TIME

Frühestens im darauf folgenden Interpolationstakt stehen die neuen Daten zur Verfügung.

Mit dem folgenden Maschinendatum kann die Zeitdauer, innerhalb der eine Request-Anforderungen an die PLC abgesetzt wird, vorgegeben werden.

MD10398 $MN_PLCIO_IN_UPDATE_TIME

Die eingetragene Zeitdauer wird intern auf das nächst höhere Vielfache eines Interpolationstaktes aufgerundet. Wird der Wert dieses Maschinendatums auf 0 gesetzt, so wird die Request-Anforderung weiterhin in jedem Interpolationstakt an die PLC abgesetzt.

Projektierung

Zur Aktivierung der Funktionalität müssen in der NC folgende Maschinendaten (Power On aktiv) projektiert werden:

MD10394 $MN_PLCIO_NUM_BYTES_IN

Anzahl der PLC-Peripherie Eingang-Bytes, die von der NC direkt gelesen werden.

MD10395 $MN_PLCIO_LOGIC_ADDRESS_IN

Logische Start-Adresse der PLC-Eingangs-Peripherie, ab der Daten gelesen werden.

MD10396 $MN_PLCIO_NUM_BYTES_OUT

Anzahl der PLC-Peripherie Ausgang-Bytes, die von der NC direkt beschrieben werden.

MD10397 $MN_PLCIO_LOGIC_ADDRESS_OUT

Logische Start-Adresse der PLC-Ausgangs-Peripherie, ab der Daten geschrieben werden.

MD10398 $MN_PLCIO_IN_UPDATE_TIME

Zeitdauer, innerhalb der die über $A_PBx_IN lesbaren Daten aktualisiert werden. Die Zeitdauer wird intern auf das nächst höhere Vielfache der durch den Interpolationstakt vorgegebenen Zeit aufgerundet. Bei Eingabe von 0 (Standardwert) erfolgt die Aktualisierung der Daten in jedem Interpolationstakt.

MD10399 $MN_PLCIO_TYPE_REPRESENTATION

Little-/Big-Endian Formatdarstellung der $A_PBx_OUT, $A_PBx_IN Systemvariablen für direkt von NCK ansteuerbare PLC-Peripherie.

value = 0 (Default)
Darstellung der Systemvariablen erfolgt im Little-Endian-Format
(d. h. niederwertigstes Byte an niederwertigster Adresse)

value = 1 (PLC-Standardformat, empfohlen)
Darstellung der Systemvariablen erfolgt im Big-Endian-Format
(d. h. höchstwertiges Byte an niederwertigster Adresse)

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.

Speicherordnung

Für den Datenaustausch von/zur PLC stehen jeweils 16 Byte (über alle Kanäle) 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 geschieht,
je nach Einstellung des Maschinendatums entweder im Little-Endian-Format (=0) oder im Big-Endian-Format (=1).:

MD10399 $MN_PLCIO_TYPE_REPRESENTATION

Da das Big-Endian-Format in der PLC die allgemein gebräuchliche Darstellungsart ist (d. h. auch für die PLC-Peripherie gültig ist), sollte diese generell zur Anwendung kommen.

Ausrichtung

Die Belegung des Eingabe- und Ausgabebereiches für direkte PLC-I/O muss folgenden Bedingungen genügen:

$A_PBB_IN[j]	; j < ([MD10394 $MN_PLCIO_NUM_BYTES_IN])
$A_PBW_IN[j]	; j < ([MD10394 $MN_PLCIO_NUM_BYTES_IN] - 1)
$A_PBD_IN[j]	; j < ([MD10394 $MN_PLCIO_NUM_BYTES_IN] - 3)
$A_PBR_IN[j]	; j < ([MD10394 $MN_PLCIO_NUM_BYTES_IN] - 3)
$A_PBB_OUT[k]	; k < ([MD10396 $MN_PLCIO_NUM_BYTES_OUT])
$A_PBW_OUT[k]	; k < ([MD10396 $MN_PLCIO_NUM_BYTES_OUT] - 1)
$A_PBD_OUT[k]	; k < ([MD10396 $MN_PLCIO_NUM_BYTES_OUT] - 3)
$A_PBR_OUT[k]	; k < ([MD10396 $MN_PLCIO_NUM_BYTES_OUT] - 3)

Weiterhin darf die maximale Anzahl der für den Datenaustausch zur Verfügung stehenden Bytes nicht überschritten werden.

1.2.6 Analogwertdarstellung der Analog-Ein-/Ausgabewerte des NCK

Umwandlung von Analogwerten

Die Analogwerte werden nur in digitaler Form von der NCU verarbeitet.

Analog-Eingabebaugruppen wandeln das analoge Prozess-Signal in einen Digitalwert um.

Analog-Ausgabebaugruppen wandeln den digitalen Ausgabewert in einen Analogwert um.

Analogwertdarstellung

Der digitalisierte Analogwert ist für die Ein- und Ausgabewerte bei gleichem Nennbereich (z. B. Spannungsbereich ± 10V DC) derselbe.

Die Darstellung der Analogwerte an der PLC-Nahtstelle erfolgt als Festpunktzahl (16 Bit einschließlich Vorzeichen) im 2er-Komplement (siehe folgende Tabelle).

Tabelle 1- 1 Digitale Analogwertdarstellung an der PLC-Nahtstelle

Auflösung	**Binärer Analogwert**															
	High-Byte								Low-Byte							
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}

Vorzeichen

Das Vorzeichen (VZ) des Analogwertes steht immer im Bit 15.

Für VZ gilt: "0" → +
"1" → -

Auflösung weniger als 15 Bit

Abhängig von der Auflösung des Digital-Analog-Wandlers kann der Analogwert feinstufig verändert werden.

Beträgt die Auflösung der Analogbaugruppe weniger als 15 Bit, wird der Analogwert linksbündig eingetragen. Die nicht-besetzten niederwertigen Stellen werden mit "0" beschrieben.

Die folgende Tabelle zeigt am Beispiel eines Bitmusters, wie bei Auflösung eines 14-Bit- und eines 12-Bit-Analogwertes die nicht-besetzbaren Bitstellen mit "0" beschrieben sind.

Bei einer Auflösung von 14 Bit (inklusive Vorzeichen) beträgt die minimale Schrittweite 1,22 mV (10 V : 8192). In diesem Fall sind die beiden niederwertigen Bits des Analogwertes (Bit 0 und Bit 1) immer 0.

Bei einer Auflösung von 12 Bit (inklusive Vorzeichen) ergibt das Schritte von 4,8 mV (10 V : 2048). Bit 0 bis 3 sind immer 0.

Tabelle 1- 2 Beispiele für digitale Analogwertdarstellung

Auflösung	**Binärer Analogwert**															
	High-Byte								Low-Byte							
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}
14-Bit-Analogwert	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0
12-Bit-Analogwert	0	1	1	1	1	0	0	1	1	0	0	1	0	0	0	0

Die Auflösung und der Nennbereich der verwendeten Analog-Ein-/Ausgabebaugruppen sind zu entnehmen:
Literatur:
/PHD/ SINUMERIK 840D Handbuch Projektierung NCU(HW)
/S7H/ SIMATIC S7-300 Installationshandbuch Technologische Funktionen

Beispiele

Nachfolgend sind zwei Beispiele für die digitale Analogwertdarstellung bei einem Nennbereich von 10 V und 14 Bit-Auflösung dargestellt.

Beispiel 1:

Analogwert:	9,5V
Betrag (Dezimalzahl):	7782 = j9,5(V)j:10(V) * 8192
Betrag (Dualzahl):	0111 1001 1001 10
Wort (Dualzahl):	0111 1001 1001 10**00**
Wort (Hexadezimalzahl):	7998

Beispiel 2:

Analogwert:	-4,12V
Betrag (Dezimalzahl):	3375 = j-4,12(V)j:10(V) * 8192
Betrag (Dualzahl):	0011 0100 1011 11
2er-Komplement:	1100 1011 0100 01
Wort (Dualzahl):	1100 1011 0100 01**00**
Wort (Hexadezimalzahl):	CB44

1.2.7 Komparator-Eingänge

Funktion

Zusätzlich zu den digitalen und analogen NCK-Eingängen stehen noch 2 interne Komparator-Eingangsbytes (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 (siehe Bild).

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 bis 8	für Komparatorbyte 1
n = 9 bis 16	für Komparatorbyte 2

Begriffe

In der vorliegenden Beschreibung werden die Begriffe "Komparator-Eingänge" (mit Index [n]; Bereich n: 1 bis 8 bzw. 9 bis 16) und "Komparator-Eingangsbits" (mit Index [b]; Bereich b: 0 bis 7) verwendet.

Dabei gelten folgende Zusammenhänge:

Für n = 1 bis 8:	Komparator-Eingang n entspricht Komparator-Eingangsbit b = n - 1.
Für n = 9 bis 16:	Komparator-Eingang n entspricht Komparator-Eingangsbit b = n - 9.

Beispiel

Komparator-Eingang 1 entspricht Komparator-Eingangsbit 0.

Zuordnung der Analog-Eingänge

Mit dem folgenden, allgemeinen Maschinendatum wird dem Eingangsbit [b] des Komparatorbytes 1 ein Analogeingang zugeordnet.:

MD10530 $MN_COMPAR_ASSIGN_ANA_INPUT_1[b]

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

Für das Komparatorbyte 2 ist die Zuordnung entsprechend mit dem folgenden Maschinendatum vorzunehmen:
MD10531 $MN_COMPAR_ASSIGN_ANA_INPUT_2[b]

Komparator-Parametrierung

Mit dem allgemeinen Maschinendatumwerden für die einzelnen Bits (0 bis 7) des Komparatorbytes 1 folgende Parameter vorgegeben:
MD10540 $MN_COMPAR_TYPE_1

- Vergleichstyp-Maske (Bit 0 bis 7)

  Für jedes Komparator-Eingangsbit wird die Art der Vergleichsbedingung festgelegt.

Bit = 1:	Zugehöriges Komparator-Eingangsbit wird auf "1" gesetzt, wenn der Analogwert ≥ dem Schwellwert ist.
Bit = 0:	Zugehöriges Komparator-Eingangsbit wird auf "0" gesetzt, wenn der Analogwert ≤ dem Schwellwert ist.

- Ausgabe des Komparator-Eingangsbytes über digitale NCK-Ausgänge (Bit 16 bis 23)

  Die Komparatorbits können zusätzlich über die digitalen NCK-Ausgänge byteweise direkt ausgegeben werden. Dazu muss in diesem Byte (Bit 16 bis 23) angegeben werden, welches digitale NCK-Ausgangsbyte dafür verwendet wird (siehe MD10540 $MN_COMPARE_TYPE_1).

- Invertiermaske für die Ausgabe des Komparator-Eingangsbytes (Bit 24 bis 31)

  Für jedes Komparatorsignal kann zusätzlich festgelegt werden, ob der an den digitalen NCK-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 [b] ist ein eigener Schwellwert einzutragen:

SD41600 $SN_COMPAR_THRESHOLD_1[b]
(Schwellwerte für Eingangsbit [b] des Komparatorbytes 1); b = 0 bis 7

Komparatorsignale als digitale NCK-Eingänge

Alle NC-Funktionen, deren Ablauf abhängig von digitalen NCK-Eingängen bestimmt wird, können auch von den Signalzuständen der Komparatoren gesteuert werden. Dabei ist in dem der NC-Funktion zugehörigen MD ("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.

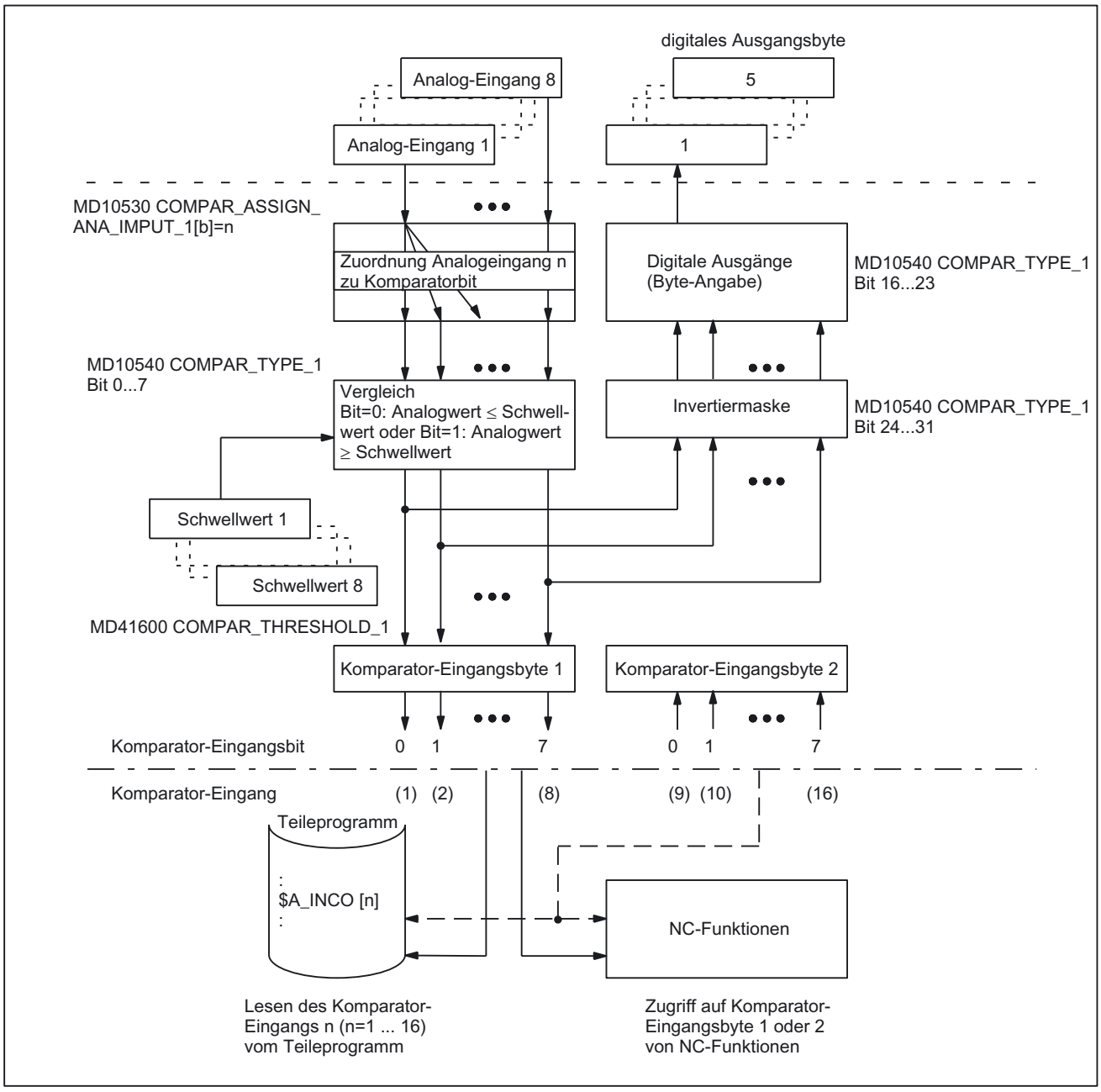


Bild 1-5 Funktionsablauf für Komparator-Eingangsbyte 1 (bzw. 2)

1.3 NCK-Peripherie über PROFIBUS - nur 840D sl

1.3.1 Funktionalität

Allgemein

Durch die Funktion "NCK-Peripherie über PROFIBUS" wird ein direkter Datenaustausch zwischen NCK und PROFIBUS-Peripherie realisiert.

Die PROFIBUS-Peripherie ist an der Steuerung angeschlossen. Wie für jede andere PLC-Peripherie muss vor der Benutzung dieser PROFIBUS-Peripherie eine S7-HW-Konfiguration (PLC) durchgeführt werden.

Werden einzelne Nutzdaten-Slots eines PROFIBUS-Slaves bei der S7-HW-Konfiguration (PLC) so konfiguriert, dass sie einen lückenlos hintereinander liegenden PROFIBUS-Peripherie-Abschnitt bilden, mit logischen Startadressen in aufsteigender Reihenfolge, wird dieser Abschnitt nachfolgend als E/A-Bereich bezeichnet.

Ein E/A-Bereich ist gekennzeichnet durch:

- eine logische Startadresse (dies entspricht der logischen Startadresse des ersten Nutzdaten-Slots dieses E/A-Bereichs)
- eine konfigurierte Länge (dies entspricht der Länge in Byte der für den Zugriff vorgesehenen zusammenhängenden PROFIBUS-Peripherie)

Die logischen Startadressen des E/A-Bereichs müssen dem NCK bekannt gemacht werden, damit dieser über eine NCK-interne PROFIBUS-Kommunikationsschnittstelle die entsprechenden Daten der PROFIBUS-Peripherie lesen bzw. schreiben kann. Die Anmeldung des projektierten E/A-Bereichs erfolgt über Maschinendaten. Die Kommunikation mit der PROFIBUS-Peripherie ist deshalb nur E/A-Bereichs-orientiert möglich.

Datenaustausch

Der Datenaustausch mit der PROFIBUS-Peripherie erfolgt über eine NCK-interne PROFIBUS-Kommunikationsschnittstelle. Folgende Möglichkeiten zum Datenaustausch mit der PROFIBUS-Peripherie stehen dem NCK-Anwender zur Verfügung:

- Lesen/Schreiben von NCK-Systemvariablen ($A_DPx_IN[n,m] bzw. $A_DPx_OUT[n,m]) über Teileprogramm/Synchronaktionen im IPO-Takt (Datenkonsistenz). Zu schreibende PROFIBUS-Peripherie-Daten werden erst nach dem entsprechenden IPO-Takt an die PROFIBUS-Peripherie ausgegeben.
- Einlesen/Ausgeben von Datenblöcken über die Compile-Zyklen-Schnittstelle (Datenkonsistenz für Servo-Takt)

Paralleler Datenzugriff

Ein **paralleler lesender Zugriff** durch Compile-Zyklen und Teileprogramm/Synchronaktion auf Daten des gleichen E/A-Bereichs ist **möglich**, sofern der entsprechende E/A-Bereich dafür konfiguriert wurde. Es ist jedoch zu beachten, dass die Lesezugriffe auf verschiedene Abbilder der PROFIBUS-Peripherie-Daten zugreifen. Die Datenkonsistenz innerhalb dieser Abbilder ist sichergestellt. Die Datengleichheit zwischen diesen Abbildern kann jedoch während eines IPO-Taktes nicht sichergestellt werden.

Ein **paralleler schreibender Zugriff** durch Compile-Zyklen und Teileprogramm/Synchronaktion auf Daten des gleichen E/A-Bereichs ist **nicht möglich**. Bei der Konfiguration des NCK muss festgelegt werden, ob ein bestimmter E/A-Bereich der PROFIBUS-Peripherie den Systemvariablen oder den Compile-Zyklen zugeordnet wird.

Aktivierung

Die Aktivierung der Funktion geschieht während des NCK-Hochlaufs.

1.3.2 Parametrierung

Die Konfiguration der E/A-Bereiche erfolgt über Maschinendaten. Die eingestellten Parameter können während des normalen Betriebs des NCK nicht mehr verändert werden.

Es werden 16 E/A-Bereiche in Leserichtung und 16 E/A-Bereiche in Schreibrichtung zur Verfügung gestellt. Die max. Größe der E/A-Bereiche wird vom NCK auf jeweils 128 Bytes begrenzt.

Ein einmal aktivierter E/A-Bereich wird zu Beginn jedes IPO-Taktes darauf überprüft, ob er noch verfügbar ist. Dazu wird das Lebenszeichen eines Nutzdaten-Slots innerhalb eines E/A-Bereichs ausgewertet. Wenn das Lebenszeichen am Beginn eines IPO-Taktes nicht gesetzt ist, wird ein Alarm (9050 bzw. 9052) ausgegeben. Dieser Alarm führt nicht zum Anhalten der Teileprogrammbearbeitung, sondern wird lediglich angezeigt und steht so lange an, bis der betroffene E/A-Bereich wieder ein gültiges Lebenszeichen liefert.

Logische Startadresse

Um bestimmte E/A-Bereiche verfügbar zu machen, müssen deren logische Startadressen dem NCK mitgeteilt werden. Dafür sind folgende Maschinendaten zu konfigurieren:

MD10500 $MN_ DPIO_LOGIC_ADDRESS_IN[n]

MD10510 $MN_ DPIO_LOGIC_ADDRESS_OUT[n]

Länge eines E/A-Bereichs

Damit der NCK überprüfen kann, ob ein E/A-Bereich vollständig projektiert wurde, muss die erwartete Länge (Bytes) des entsprechenden E/A-Bereichs eingetragen werden. Dafür sind folgende Maschinendaten zu konfigurieren:

MD10501 $MN_ DPIO_RANGE_LENGTH_IN[n]

MD10501 $MN_ DPIO_RANGE_LENGTH_OUT[n]

Wenn die Länge "0" eingetragen ist, wird nur der unter der entsprechenden logischen Startadresse gefundene Nutzdaten-Slot als E/A-Bereich konfiguriert. Die Länge des E/A-

Bereichs wird dann in einem solchen Fall mit der Länge des gefundenen Nutzdaten-Slots gleichgesetzt.

Weitere Attribute

Mit folgenden Maschinendaten können jedem E/A-Bereich weitere Attribute zugewiesen werden:

MD10502 $MN_ DPIO_RANGE_ATTRIBUTE_IN[n]

Bit	Wert	Bedeutung
0	0	Little-Endian Format
	1	Big-Endian Format
1		reserviert
2	0	Lesen über Systemvariable und CC-Binding möglich.
	1	Lesen nur für CC-Binding möglich.
3	0	Slot-Lebenszeichen-Alarme werden ausgegeben.
	1	Slot-Lebenszeichen-Alarme werden unterdrückt.

MD10512 $MN_ DPIO_RANGE_ATTRIBUTE_OUT[n]

Bit	Wert	Bedeutung
0	0	Little-Endian Format
	1	Big-Endian Format
1	0	Schreiben nur über Systemvariable.
	1	Schreiben nur über CC-Binding.
2		reserviert
3	0	Slot-Lebenszeichen-Alarme werden ausgegeben.
	1	Slot-Lebenszeichen-Alarme werden unterdrückt.

Zu beachten

- Die Konfiguration der E/A-Bereiche über die entsprechenden Maschinendaten muss nicht durchgehend erfolgen. D. h., die Zuordnung der E/A-Bereiche zu den entsprechenden Maschinendaten-Indices ist beliebig wählbar.
- Wenn die Anmeldung eines/mehrerer E/A-Bereiche während der NCK-Hochlaufphase nicht erfolgreich verläuft, wird deren Anmeldung mit einem Alarm (4700 oder 4702) abgebrochen.

1.3.3 Programmierung

Voraussetzungen

- Korrekte Konfiguration der entsprechenden E/A-Bereiche.
- Die angeforderten E/A-Bereiche (Nutzdaten-Slots) müssen auch tatsächlich von der PLC zur Verfügung gestellt werden können.
- Die konfigurierten E/A-Bereiche werden für die Benutzung erst dann freigegeben, wenn die PROFIBUS-Kommunikationsschnittstelle erstmalig einen Datenaustausch mit der entsprechenden PROFIBUS-Peripherie durchführen konnte.

1.3.3.1 Kommunikation über Teileprogramm/Synchronaktionen

Allgemein

Den Zugriff auf die PROFIBUS-Peripherie, ob lesend oder schreibend, aus dem Teileprogramm/Synchronaktionen ermöglichen die NCK-globalen Systemvariablen:

- $A_DPx_IN [n,m]
- $A_DPx_OUT [n,m]

Bitte beachten Sie dabei Folgendes:

- Beim Lesen bzw. Schreiben auf diese Variable aus einem Teileprogramm heraus wird 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 IPO-Takt 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.

Auf E/A-Bereich-Daten zugreifen

Folgende Systemvariable stehen für den Zugriff auf E/A-Bereich-Daten zur Verfügung:

Tabelle 1- 3 NCK → PROFIBUS-Peripherie

Systemvariable	Wert	Beschreibung
$A_DPB_OUT[n,m]	8 Bit unsigned	Schreiben eines Datenbytes (8 Bit) auf PROFIBUS-IO
$A_DPW_OUT[n,m]	16 Bit unsigned	Schreiben eines Datenwortes (16 Bit) auf PROFIBUS-IO
$A_DPSB_OUT[n,m]	8 Bit signed	Schreiben eines Datenbytes (8 Bit) auf PROFIBUS-IO
$A_DPSW_OUT[n,m]	16 Bit signed	Schreiben eines Datenwortes (16 Bit) auf PROFIBUS-IO
$A_DPSD_OUT[n,m]	32 Bit signed	Schreiben eines Datendoppelwortes (32 Bit) auf PROFIBUS-IO
$A_DPR_OUT[n,m]	32 Bit REAL	Schreiben von Ausgangs-Daten (32 Bit REAL) auf PROFIBUS-IO

n = Index für den Ausgangs-Datenbereich; m = Byte-Index für die Daten

Tabelle 1- 4 PROFIBUS-Peripherie → NCK

Systemvariable	Wert	Beschreibung
$A_DPB_IN[n,m]	8 Bit unsigned	Lesen eines Datenbytes (8 Bit) von PROFIBUS-IO
$A_DPW_IN[n,m]	16 Bit unsigned	Lesen eines Datenwortes (16 Bit) von PROFIBUS-IO
$A_DPSB_IN[n,m]	8 Bit signed	Lesen eines Datenbytes (8 Bit) von PROFIBUS-IO
$A_DPSW_IN[n,m]	16 Bit signed	Lesen eines Datenwortes (16 Bit) von PROFIBUS-IO
$A_DPSD_IN[n,m]	32 Bit signed	Lesen eines Datendoppelwortes (32 Bit) von PROFIBUS-IO
$A_DPR_IN[n,m]	32 Bit REAL	Lesen von Ausgangs-Daten (32 Bit REAL) von PROFIBUS-IO

n = Index für den Eingangs-Datenbereich; m = Byte-Index für die Daten

Konfiguration von E/A-Bereichen prüfen

Über die folgenden Systemvariablen kann die Konfiguration von E/A-Bereichen überprüft werden. Jedes Bit dieser Bitfelder entspricht einem E/A-Bereich. Es wird gesetzt, wenn der E/A-Bereich für den Zugriff über Teileprogramm/Synchronaktionen konfiguriert wurde.

Systemvariable	Wert	Beschreibung
$A_DP_IN_CONF	32 Bit Bitfeld	Lesen aller konfigurierter Eingangs-Datenbereiche des PROFIBUS-IO
$A_DP_OUT_CONF	32 Bit Bitfeld	Lesen aller konfigurierter Ausgangs-Datenbereiche des PROFIBUS-IO

Verfügbarkeit von E/A-Bereichen prüfen

Über die folgenden Systemvariablen kann die Verfügbarkeit von E/A-Bereichen überprüft werden. Jedes Bit dieser Bitfelder entspricht einem E/A-Bereich. Es wird gesetzt, wenn der E/A-Bereich für den Zugriff über Teileprogramm/Synchronaktionen einsatzbereit ist.

Systemvariable	Wert	Beschreibung
$A_DP_IN_VALID	32 Bit Bitfeld	Lesen aller gültigen Eingangs-Datenbereiche des PROFIBUS-IO
$A_DP_OUT_VALID	32 Bit Bitfeld	Lesen aller gültigen Ausgangs-Datenbereiche des PROFIBUS-IO

Zustand eines E/A-Bereichs abfragen

Über die folgenden Systemvariablen kann der genaue Zustand eines E/A-Bereichs abgefragt werden.

Systemvariable	Wert	Beschreibung
$A_DP_IN_STATE[n] n = Index für den Eingangs-Datenbereich	0: Datenbereich wurde nicht konfiguriert 1: Datenbereich konnte noch nicht aktiviert werden 2: Datenbereich ist verfügbar 3: Datenbereich ist momentan nicht verfügbar	Lesen des Zustands des Eingangs-Datenbereiches
$A_DP_OUT_STATE[n] n = Index für den Ausgangs-Datenbereich		Lesen des Zustands des Ausgangs-Datenbereiches

Länge eines E/A-Bereiches abfragen

Über die folgenden Systemvariablen kann die konfigurierte Länge eines E/A-Bereiches abgefragt werden.

Systemvariable	Beschreibung
$A_DP_IN_LENGTH[n] n = Index für den Eingangs-Datenbereich	Lesen der Länge des Eingangs-Datenbereiches
$A_DP_OUT_LENGTH[n] n = Index für den Ausgangs-Datenbereich	Lesen der Länge des Ausgangs-Datenbereiches

Zu beachten

- Über <n> (RangeIndex) wird der entsprechende NCK-konfigurierte E/A-Bereich ausgewählt. Sollte der angeforderte E/A-Bereich nicht konfiguriert sein, wird dies durch die Ausgabe eines Alarms (17020) angezeigt.
- Mit <m> (RangeOffset) wird auf die Stelle (Byte-Offset) innerhalb des E/A-Bereich 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 das Maschinendatum:
  MD10502 $MN_DPIO_RANGE_ATTRIBUTE_IN bzw.
  MD10512 $MN_DPIO_RANGE_ATTRIBUTE_OUT
  kann sowohl für die Lese-/Schreibrichtung, als auch für jeden einzelnen E/A-Bereich, das Darstellungsformat (Little-/Big-Endian) für $A_DPx_IN[n,m]- bzw. $A_DPx_OUT[n,m]-Systemvariablen festgelegt werden.

1.3.3.2 Kommunikation über Compile-Zyklen

Allgemein

Für das Einlesen/Ausgeben von Datenblöcken über die Compile-Zyklen-Schnittstelle stehen CC-Bindings zur Verfügung. Der Zugriff auf Daten der E/A-Bereiche findet in der Servo-Task-Ebene statt. Die Daten werden in jedem Servo-Takt aktualisiert.

Datenkonsistenz ist deshalb nur für den jeweiligen Servo-Takt gegeben.

Um über die CC-Bindings schreibend auf Daten der E/A-Bereiche zugreifen zu können, müssen die betreffenden E/A-Bereiche bei der NCK-Konfiguration durch das Maschinendatum:
MD10512 $MN_DPIO_SLOT_ATTRIBUTE_OUT (Attribute der PROFIBUS-Peripherie)
für die Programmierung über Compile-Zyklen freigeschaltet werden.

Eine gleichzeitige Programmierung dieser E/A-Bereiche über Teileprogramm/Synchronaktionen wird unter Ausgabe eines Alarms (17020) unterbunden.

Es ist zu beachten, dass die Darstellung von Daten in der PLC im Allgemeinen im Big-Endian-Format erfolgt. Dies trifft natürlich auch auf die PROFIBUS-Peripherie zu. Da die Bindings nur den Byte orientierten Zugriff auf Datenbereiche (Byte-Offset, Anzahl zu übertragender Bytes) innerhalb eines E/A-Bereichs unterstützen, ist auf die korrekte Darstellung von Datentypen (16 Bit, 32 Bit, etc.) zu achten.

CC-Bindings

Folgende CC-Bindings stehen zur Verfügung:

CCDataOpi: getDpIoRangeConfiguration()

CCDataOpi: getDpIoRangeValid()

CCDataOpi: getDpIoRangeInInformation()

CCDataOpi: getDpIoRangeOutInformation()

CCDataOpi: getDpIoRangeInState()

CCDataOpi: getDpIoRangeOutState()

CCDataOpi: getDataFromDpIoRangeIn()

CCDataOpi: putDataToDpIoRangeOut()

Zu beachten

- Die Bindings
  CCDataOpi: getDataFromDpIoRangeIn() bzw.
  CCDataOpi: putDataToDpIoRangeOut()
  überwachen bei Lese-/Schreibzugriffen die Einhaltung der NCK- und PLC-seitig konfigurierten Grenzen des jeweiligen E/A-Bereichs. Zugriffe auf Daten/Datenbereiche, die nicht vollständig innerhalb der konfigurierten E/A-Bereichsgrenzen liegen, werden unter Rückgabe des Enumerator CCDATASTATUS_RANGE_LENGTH_LIMIT abgelehnt.
- Sollte der Zugriff auf einen nicht (bzw. nicht für den Compile-Zyklus) konfigurierten E/A-Bereich versucht werden, wird dies durch den Rückgabe-Enumerator CCDATASTATUS_RANGE_NOT_AVAILABLE mitgeteilt.

ACHTUNG
Der Compile-Zyklen-Programmierer ist selbst für den richtigen Gebrauch der CC-Bindings verantwortlich! Es ist darauf zu achten, dass der zusätzliche Performance-Bedarf, der für die Bereitstellung von Daten der konfigurierten E/A-Bereiche auf Servo-Task-Ebene benötigt wird, nicht zu einem Servo-Ebenen-Rechenzeitüberlauf führt. Nähere Informationen über den Gebrauch dieser Bindings sind den OEM-Unterlagen zu entnehmen.

1.4 Randbedingungen

1.4.1 NCK-Peripherie über PLC

Verfügbarkeit von "digitalen und analogen NC-Ein-/Ausgängen"

Digitale und analoge CNC-Ein-/Ausgänge (DE, DA, AE, AA) sind wie folgt verfügbar:

- SINUMERIK 840D mit NCU 571

  4 DE / 4 DA (On-Board)

  32 DE / 32 DA mit Erweiterung über NCU-Terminalblock
- SINUMERIK 840D mit NCU 572/573, ab SW 2

  4 DE / 4 DA (On-Board)

  32 DE / 32 DA und 8 AE / 8 AA mit Erweiterung über NCU-Terminalblock

Analoge Peripherie 840Di

Die analoge Peripherie wird über den PROFIBUS-DP an die SINUMERIK 840Di angeschlossen.

Konfiguration

- Die PLC-Peripherie muss, wenn sie durch den schnellen Datenkanal beschrieben/gelesen werden soll, immer als zusammenhängender Block konfiguriert sein (d. h. keine Adresslücken innerhalb dieses Blocks).
- Die Anzahl der zu übertragenden Bytes muss sich lückenlos auf der PLC-Peripherie abbilden lassen.

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 (NCK <--> PLC)

- Die Ausgabe des Datenpuffers an die PLC-Peripherie erfolgt immer vollständig, auch wenn nur eine Systemvariable innerhalb dieses Datenpuffers zugewiesen wurde.
- Werden mehreren Systemvariablen gleichzeitig Werte zugewiesen (z. B. zur Initialisierung von PLC-Peripherie), so ist nicht sichergestellt, dass diese im selben Interpolationstakt übertragen werden.

1.4.2 NCK-Peripherie über PROFIBUS - nur 840D sl

System

Die Funktion ist im System SINUMERIK 840D sl für isochron und nicht isochron projektierte PROFIBUS-Peripherie verfügbar.

Hardware

- Die erforderliche PROFIBUS-Peripherie muss vorhanden und betriebsbereit sein.
- Es muss eine korrekte S7-HW-Konfiguration (PLC-Seite) mit der benötigten PROFIBUS-Peripherie durchgeführt und in die PLC geladen worden sein.
- Ein E/A-Bereich muss sich auf dem gleichen PROFIBUS-Slave befinden.
- Es werden nur PROFIBUS-Slaves am ersten realen PROFIBUS-Strang der PLC (Stecker mit der Bezeichnung DP1) unterstützt.

Software

- Der NCK muss über die entsprechenden Maschinendaten korrekt konfiguriert worden sein.
- Ein gleichzeitiges Beschreiben der PROFIBUS-Peripherie aus einem PLC-Anwenderprogramm ist unzulässig und kann technisch nicht verhindert werden.
- E/A-Bereiche für den schreibenden Zugriff (MD10510 $MN_DPIO_LOCIG_ADDRESS_OUT[]) auf die PROFIBUS-Peripherie dürfen nicht im Peripherie-Abbildbereich der PLC (z. B. PLC 317, Adressen von 0 - 255) liegen, da dieser Bereich vom PLC-Betriebssystem beansprucht wird.

1.5 Beispiele

1.5.1 NCK-Peripherie über PLC

1.5.1.1 Schreiben auf PLC-Peripherie

Für dieses Beispiel werden folgende Annahmen getroffen:

- Es sollen Daten direkt auf folgende PLC-Peripherie ausgegeben werden:

  - log. Adr. 521: ;8Bit Digital-Ausgabebaugruppe
  - log. Adr. 522: ;16Bit Digital-Ausgabebaugruppe

- Die Ausgabe geschieht über $A_PBx_OUT aus **Synchronaktionen** heraus.

Parametrierung

Die Maschinendaten sind folgendermaßen zu setzen:

MD10397 $MN_PLCIO_LOGIC_ADRESS_OUT= 521	;ab log. Adr. 521 werden Daten ausgegeben
MD10396 $MN_PLCIO_NUM_BYTES_OUT= 3	;es müssen insgesamt 3 Byte ausgegeben werden
MD10399 $MN_PLCIO_TYPE_REPRESENTATION = 1	;Darstellung der Daten erfolgt im Big-Endian-Format

Hochlauf von NCK und PLC

Nach dem Hochlauf von NCK und PLC findet **kein** zyklischer Datentransfer (für schreibende Zugriffe) zur PLC-Peripherie statt.

Programmierung

Laden und Starten des Teileprogramms mit folgendem Inhalt:

```
...
ID = 1 WHENEVER TRUE DO $A_PBB_OUT[0] = 123                ;zyklische Ausgabe
                                                           ;(pro
                                                           Interpolationstakt)
...
ID = 2 WHEN $AA_IW[x] >= 5 DO $A_PBW_OUT[1] = 'Habcd'      ;Ausgabe eines
                                                           ;HEX-Wertes
...
```

1.5.1.2 Lesen von PLC-Peripherie

Für dieses Beispiel werden folgende Annahmen getroffen:

- PLC-Peripherie:

  - log. Adr. 420: 16 Bit Analog-Eingabebaugruppe
  - log. Adr. 422: 32 Bit Digital-Eingabebaugruppe
  - log. Adr. 426: 32 Bit Input DP-Slave
  - log. Adr. 430: 8 Bit Digital-Eingabebaugruppe

- Das Einlesen geschieht über $A_PBx_IN in R-Parameter aus einem Teileprogramm heraus.
- Um die Abarbeitungsgeschwindigkeit des PLC-Anwenderprogramms (OB1) nicht unnötig zu verlangsamen, wurde über das Maschinendatum:
  MD10398 $MN_PLCIO_IN_UPDATE_TIME
  die Update-Zeit (für lesende Zugriffe) so gewählt, dass nur jeden 3ten Interpolationstakt ein Update ausgeführt wird.

Parametrierung

Die Maschinendaten sind folgendermaßen zu setzen:

MD10395 $MN_PLCIO_LOGIC_ADRESS_IN = 420	;ab log. Adr. 420 werden Daten eingelesen
MD10394 $MN_PLCIO_NUM_BYTES_IN = 11	;es müssen insgesamt 11 Byte eingelesen werden
MD10398 $MN_PLCIO_IN_UPDATE_TIME = 0.03	;Update Zeitdauer = 30 msec (Interpolationstakt = 12 msec)
MD10399 $MN_PLCIO_TYPE_REPRESENTATION = 1	;Darstellung der Daten erfolgt im Big-Endian-Format

Hochlauf von NCK und PLC:

Der Update (für lesende Zugriffe) wird bereits jetzt, nach dem Hochlauf von NCK und PLC, in jedem 3ten Interpolationstakt durchgeführt.

Programmierung

Laden und Starten des Teileprogramms mit folgendem Inhalt:

```
...
R1 = $A_PBW_IN[0]          ;Einlesen 16Bit Integer
R2 = $A_PBD_IN[2]          ;Einlesen 32Bit Integer
R3 = $A_PBR_IN[6]          ;Einlesen 32Bit Float
R4 = $A_PBB_IN[10]         ;Einlesen 8Bit Integer
...
```

1.5.2 NCK-Peripherie über PROFIBUS- nur 840D sl

1.5.2.1 PROFIBUS-Peripherie in Schreibrichtung

Voraussetzung

Die S7-HW-Konfiguration hat bereits stattgefunden.

Konfiguration für Programmierung über Teileprogramm/Synchronaktionen

- RangeIndex = 5 (NCK-interne Konfiguration)
- lt. S7-HW-Konfiguration:
  - log. Startadresse = 334
  - Slotlänge = 8 Byte
  - Darstellung soll im Little-Endian-Format erfolgen

Daraus ergibt sich folgende Konfiguration der Maschinendaten:

MD10510 $MN_DPIO_LOGIC_ADDRESS_OUT[5] = 334 (log. Startadresse E/A-Bereich)

MD10511 $MN_DPIO_RANGE_LENGTH_OUT[5] = 8 (Länge des E/A-Bereichs)

MD10512 $MN_DPIO_RANGE_ATTRIBUTE_OUT[5]
Bit0 = 0 (Little-Endian-Format)
Bit1 = 0 (Schreiben nur über Systemvariable)
Bit3 = 0 (Slot-Lebenszeichen-Alarme werden ausgegeben)

Konfiguration für Programmierung über CompileZyklen

- RangeIndex = 6 (NCK-interne Konfiguration)
- lt. S7-HW-Konfiguration:
  - log. Startadresse = 444
  - Slotlänge = 10 Byte
  - Darstellung soll im Little-Endian-Format erfolgen

Daraus ergibt sich folgende Konfiguration der Maschinendaten:

MD10510 $MN_DPIO_LOGIC_ADDRESS_OUT[6] = 444 (log. Startadresse E/A-Bereich)

MD10511 $MN_DPIO_RANGE_LENGTH_OUT[6] = 0
(es soll ein einzelner Nutzdaten-Slot benutzt werden)

MD10512 $MN_DPIO_RANGE_ATTRIBUTE_OUT[6]
Bit0 = 0 (Little-Endian-Format)
Bit1 = 1 (Schreiben nur über CC-Binding)
Bit3 = 1 (Slot-Lebenszeichen-Alarme werden unterdrückt)

Programmierung

```
$A_DPB_OUT[5,6]=128          ; schreiben (8 Bit) auf RangeIndex=5, RangeOffset=6
$A_DPW_OUT[5,5]='B0110'      ; schreiben (16 Bit) auf RangeIndex=5, RangeOffset=5
                             ; Little-Endian-Format
                             ; Achtung: RangeDaten Byte 6 werden überschrieben
$A_DPSD_OUT[5,3]='8FHex'     ; schreiben (32 Bit) auf RangeIndex=5, RangeOffset=3
                             ; Little-Endian-Format
                             ; Achtung: RangeDaten Byte 5,6 werden überschrieben
$AC_MARKER[0]=5
$AC_MARKER[1]=3
$A_DPSD_OUT[$AC_MARKER[0],$AC_MARKER[1]]='8FHex'
                             ; schreiben (32 Bit) auf RangeIndex=5, RangeOffset=3
                             ; Little-Endian-Format
                             ; indirekte Programmierung
R1=$A_DPB_OUT[5,6]           ; lesen (8 Bit) auf RangeIndex=5, RangeOffset=6
                             ; Little-Endian-Format
                             ; Ergebnis: '0xFF'
ID=1 WHENEVER TRUE DO $A_DPB_OUT[5,0]=123
                             ; zyklische Ausgabe (pro IPO-Takt)
$A_DPB_OUT[5,255]=128        ; Alarm 17030 weil der RangeOffset ungültig ist.
$A_DPB_OUT[6,10]=128         ; Alarm 17020 weil dieser Range vom Teileprogramm
                             ; nicht beschrieben werden kann.
$A_DPB_OUT[7,10]=128         ; Alarm 17020 weil dieser Range nicht definiert ist.
$A_DPB_OUT[16,6]=128         ; Alarm 17020 weil RangeIndex >= max. verfügbare
                             ; Anzahl von Ranges.
```

1.5.2.2 PROFIBUS-Peripherie in Leserichtung

Voraussetzung

Die S7-HW-Konfiguration hat bereits stattgefunden.

Konfiguration für Programmierung über Teileprogramm/Synchronaktionen

- RangeIndex = 0 (NCK-interne Konfiguration)
- lt. S7-HW-Konfiguration:
  - log. Startadresse = 456
  - Slotlänge = 32 Byte
  - Darstellung soll im Big-Endian-Format erfolgen

Daraus ergibt sich folgende Konfiguration der Maschinendaten:

MD10500 $MN_DPIO_LOGIC_ADDRESS_IN[0] = 456 (log. Startadresse E/A-Bereich)

MD10501 $MN_DPIO_RANGE_LENGTH_IN[0] = 32 (Länge des E/A-Bereichs)

MD10502 $MN_DPIO_RANGE_ATTRIBUTE_IN[0]
Bit0 = 1 (Big-Endian-Format)
Bit2 = 0 (Lesen über Systemvariable und CC-Binding möglich)
Bit3 = 0 (Slot-Lebenszeichen-Alarme werden ausgegeben)

Konfiguration für Programmierung über CompileZyklen

- RangeIndex = 1 (NCK-interne Konfiguration)
- lt. S7-HW-Konfiguration:
  - log. Startadresse = 312
  - Slotlänge = 32 Byte
  - Darstellung soll im Little-Endian-Format erfolgen

Daraus ergibt sich folgende Konfiguration der Maschinendaten:

MD10500 $MN_DPIO_LOGIC_ADDRESS_IN[1] = 312 (log. Startadresse E/A-Bereich)

MD10501 $MN_DPIO_RANGE_LENGTH_IN[1] = 32 (Länge des E/A-Bereichs)

MD10502 $MN_DPIO_RANGE_ATTRIBUTE_IN[1]
Bit0 = 1 (Big-Endian-Format)
Bit2 = 1 (Lesen nur für CC-Binding möglich)
Bit3 = 1 (Slot-Lebenszeichen-Alarme werden unterdrückt)

Programmierung

```
$AC_MARKER[0]=$A_DPW_IN[0,0]       ; lesen (16 Bit) auf RangeIndex=0, RangeOffset=0
                                   ; Big-Endian-Format
$AC_MARKER[1]=$A_DPSD_IN[0,1]      ; lesen (32 Bit) auf RangeIndex=0, RangeOffset=1
                                   ; Big-Endian-Format
$AC_MARKER[1]=$A_DPSD_IN[0,8]      ; lesen (32 Bit) auf RangeIndex=0, RangeOffset=8
                                   ; Big-Endian-Format
$AC_MARKER[2]=0
$AC_MARKER[3]=8
$AC_MARKER[1]=$A_DPSD_IN[$AC_MARKER[2],$AC_MARKER[3]]
                                   ; lesen (32 Bit) auf RangeIndex=0, RangeOffset=8
                                   ; Big-Endian-Format
                                   ; indirekte Programmierung
ID=2 WHEN $A_DPB_IN[0,11]>=5 DO $AC_MARKER[2]='ABCDHex'
                                   ; zyklisches Einlesen (pro IPO-Takt)
R1=$A_DPB_IN[0,255]                ; Alarm 17030 weil der RangeOffset ungültig ist.
R1=$A_DPB_IN[2,6]                  ; Alarm 17020 weil dieser Range nicht definiert
                                   ist.
R1=$A_DPB_IN[1,10]                 ; Alarm 17020 weil dieser Range vom Teileprogramm
                                   ; nicht beschrieben werden kann.
R1=$A_DPB_IN[16,6]                 ; Alarm 17020 weil RangeIndex >= max. verfügbare
                                   ; Anzahl von Ranges.
```

1.5.2.3 Abfrage des RangeIndex bei "PROFIBUS-Peripherie in Schreibrichtung"

Voraussetzung

Die S7-HW-Konfiguration hat bereits stattgefunden.

Konfiguration für Programmierung über Teileprogramm/Synchronaktionen

- RangeIndex = 5 (NCK-interne Konfiguration)
- lt. S7-HW-Konfiguration:
  - log. Startadresse = 1200
  - Slotlänge = 32 Byte
  - Darstellung soll im Big-Endian-Format erfolgen

Daraus ergibt sich folgende Konfiguration der Maschinendaten:

MD10510 $MN_DPIO_LOGIC_ADDRESS_OUT[5] = 1200 (log. Startadresse E/A-Bereich)

MD10511 $MN_DPIO_RANGE_LENGTH_OUT[5] = 0
(es soll ein einzelner Nutzdaten-Slot benutzt werden)

MD10512 $MN_DPIO_RANGE_ATTRIBUTE_OUT[5]
Bit0 = 1 (Big-Endian-Format)
Bit1 = 0 (Schreiben nur über Systemvariable)
Bit3 = 0 (Slot-Lebenszeichen-Alarme werden ausgegeben)

Programmierung

vor einem Zugriff den Zustand von RangeIndex = 5 abgefragen

```
N3     check:                                    ; Sprungmarke
N5     IF $A_DP_OUT_STATE[5]==2 GOTOF write      ; wenn Datenbereich gültig
                                                 ; => Sprung zu N15
N10    GOTOB check                               ; Rücksprung zu check
N15    write:                                    ; Sprungmarke
N20    $A_DPB_OUT[5,6]=128                       ; Schreiben des Datenbytes
```

Abfrage ob alle konfigurierten Ranges/Slots gültig sind

```
N3     check:                                    ; Sprungmarke
N5     IF $A_DP_OUT_CONF==$A_DP_OUT_VALID GOTOF  ; wenn Datenbereich gültig
       write                                     ; => Sprung zu N15
N10    SETAL(61000)                              ; Alarm Nr. 61000 setzen
N15    write:                                    ; Sprungmarke
N20    $A_DPB_OUT[5,6]=128                       ; Schreiben des Datenbytes
```

Abfrage ob der konfigurierte RangeIndex = 5 gültig ist

```
N3      check:                                      ; Sprungmarke
N5      IF $A_DP_OUT_VALID B_AND 'B100000' GOTOF    ; wenn Datenbereich gültig
        write                                       ; => Sprung zu N15
N10     SETAL(61000)                                ; Alarm Nr. 61000 setzen
N15     write:                                      ; Sprungmarke
N20     $A_DPB_OUT[5,6]=128                         ; Schreiben des Datenbytes
```

Abfrage der Länge des konfigurierten, gültigen E/A-Bereichs mit RangeIndex = 5

```
N100    R1=$A_DP_OUT_LENGTH[5]          ; Länge des E/A-Bereichs (Slots) in Byte
                                        ; Ergebnis: R1 = 32
```

1.6 Datenlisten

1.6.1 Maschinendaten

1.6.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
10300	FASTIO_ANA_NUM_INPUTS	Anzahl der aktiven analogen NCK-Eingänge
10310	FASTIO_ANA_NUM_OUTPUTS	Anzahl der aktiven analogen NCK-Ausgänge
10320	FASTIO_ANA_INPUT_WEIGHT	Bewertungsfaktor für analoge NCK-Eingänge
10330	FASTIO_ANA_OUTPUT_WEIGHT	Bewertungsfaktor für analoge NCK-Ausgänge
10350	FASTIO_DIG_NUM_INPUTS	Anzahl der aktiven digitalen NCK-Eingangsbytes
10360	FASTIO_DIG_NUM_OUTPUTS	Anzahl der aktiven digitalen NCK-Ausgangsbytes
10362	HW_ASSIGN_ANA_FASTIN	Hardware-Zuordnung der externen analogen NCK-Eingänge
10364	HW_ASSIGN_ANA_FASTOUT	Hardware-Zuordnung der externen analogen NCK-Ausgänge
10366	HW_ASSIGN_DIG_FASTIN	Hardware-Zuordnung der externen digitalen NCK-Eingänge
10368	HW_ASSIGN_DIG_FASTOUT	Hardware-Zuordnung der externen digitalen NCK-Ausgänge
10380	HW_UPDATE_RATE_FASTIO	Aktualisierungsrate der taktsynchronen externen NCK-Peripherie
10382	HW_LEAD_TIME_FASTIO	Vorhaltezeit der taktsynchronen externen NCK-Peripherie
10384	HW_CLOCKED_MODULE_MASK	Taktsychrone Bearbeitung der externen NCK-E/A-Peripherie
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
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 NCK-Analogeingänge für Komparatorbyte 1
10531	COMPAR_ASSIGN_ANA_INPUT_2	Hardware-Zuordnung der NCK-Analogeingänge für Komparatorbyte 2

Nummer	Bezeichner: $MN_	Beschreibung
10540	COMPAR_TYPE_1	Parametrierung für Komparatorbyte 1
10541	COMPAR_TYPE_2	Parametrierung für Komparatorbyte 2

1.6.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
21220	MULTFEED_ASSIGN_FASTIN	Zuordnung der Eingangsbytes der NCK-Peripherie für "Mehrere Vorschübe in einem Satz"

1.6.2 Settingdaten

1.6.2.1 Allgemeine Settingdaten

Nummer	Bezeichner: $SN_	Beschreibung
41600	COMPAR_THRESHOLD_1	Schwellwerte für Komparatorbyte 1
41601	COMPAR_THRESHOLD_2	Schwellwerte für Komparatorbyte 2

1.6.3 Signale

1.6.3.1 Signale an NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Sperre der digitalen NCK-Eingänge	DB10.DBB0/122/124/126/128	DB2800.DBB0/1000
Setzen von PLC der digitalen NCK-Eingänge	DB10.DBB1/123/125/127/129	DB2800.DBB1/1001
Sperre der digitalen NCK-Ausgänge	DB10.DBB4/130/134/138/142	DB2800.DBB4/1008
Überschreibmaske der digitalen NCK-Ausgänge	DB10.DBB5/131/135/139/143	DB2800.DBB5/1009
Setzwert von PLC der digitalen NCK-Ausgänge	DB10.DBB6/132/136/140/144	DB2800.DBB6/1010
Vorgabemaske der digitalen NCK-Ausgänge	DB10.DBB7/133/137/141/145	DB2800.DBB7/1011
Sperre der analogen NCK-Eingänge	DB10.DBB146	-
Vorgabemaske der analogen NCK-Eingänge	DB10.DBB147	-
Setzwert von PLC für analoge NCK-Eingänge	DB10.DBB148-163	-
Überschreibmaske der analogen NCK-Ausgänge	DB10.DBB166	-
Vorgabemaske der analogen NCK-Ausgänge	DB10.DBB167	-
Sperre der analogen NCK-Ausgänge	DB10.DBB168	-
Setzwert von PLC für analoge NCK-Ausgänge	DB10.DBB170-185	-

1.6.3.2 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: Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme - nur 840D sl

2

2.1 Kurzbeschreibung

2.1.1 Topologie dezentraler Systemaufbauten

Merkmale

Rundtaktmaschinen, Mehrspindeldrehmaschinen und komplexe NC-Fertigungszentren weisen alle oder einige der folgenden Merkmale auf:

- Mehr als eine NCU wegen hoher Achs- und Kanalanzahl
- Große Abmessungen und räumliche Trennungen erfordern mehrere Bedieneinheiten (Bedientafelfronten OP/TP mit PCU/TCU, Maschinensteuertafeln MSTT, Handheld Terminal HT8)
- Modulares Maschinenkonzept, z. B. durch dezentrale Schaltschränke

Dezentraler Systemaufbau

Die im Topologiebild hervorgehobenen Bereiche kennzeichnen zwei für Konfigurierung und Benutzung gesondert zu betrachtende Kommunikationsfunktionen.

Eine Bedientafel besteht in der Regel aus einer OP/TP mit einer PCU, auf der die HMI-Software läuft. Sollen mehrere OPs/TPs mit einer PCU verbunden werden, werden zusätzlich TCUs benötigt.

Literatur:
Gerätehandbuch Bedienkomponenten und Vernetzung

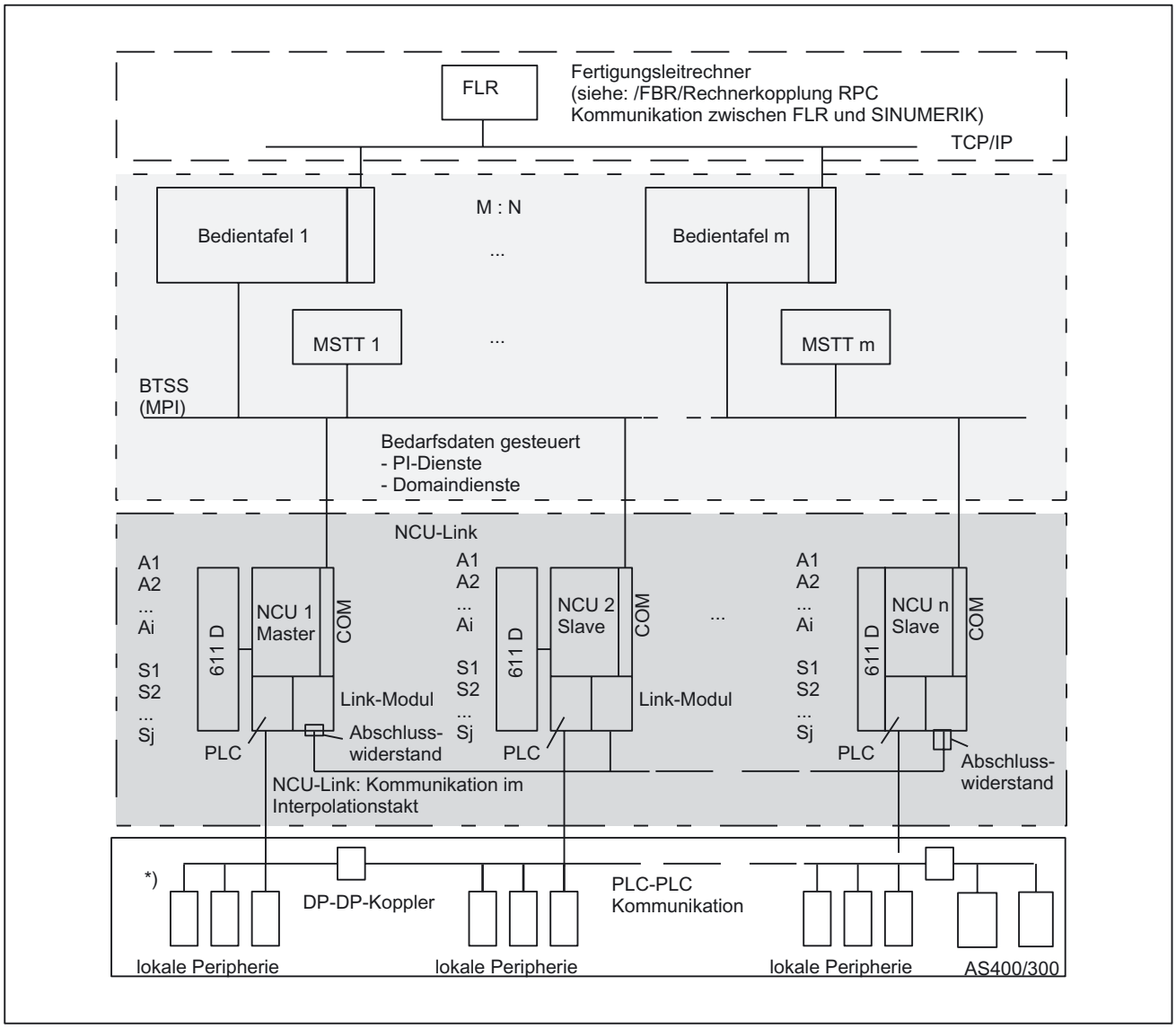


Bild 2-1 Topologie Dezentrale Systemaufbauten

*) Bei PLC-PLC Kommunikation liegt entweder vor:
- PLC-PLC-Querkommunikation Master-, Slave-K.)
oder
- PLC-lokale Peripherie

M : N

Zuordnung mehrerer Bedieneinheiten (M) zu mehreren NCUs (N):

- Busadressen, Busart
- Eigenschaften der Bedieneinheiten:
  - Hauptbedienfeld/Nebenbedienfeld
- Dynamische Umschaltungen von PCUs/MSTTs bzw. HT6s zu anderen NCUs

Für die Nutzung von M:N sind Maßnahmen erforderlich bei:

- Projektierung der Hardware

  Literatur:
  Leitfaden für die Maschinenprojektierung
- Parametrierung in Dateien
- Gestaltung des PLC-Programms

  Literatur:
  Funktionshandbuch Grundfunktionen; PLC-Grundprogramm (P3)
- Bedienung

  Literatur:
  Bedienhandbuch

Für Anwendungen/Konfigurationen, die den beschriebenen Beispielen entsprechen, können die Beispielnotationen direkt übernommen oder leicht angepasst werden. Es sind jeweils die Aspekte Projektierung in Dateien, PLC-Programm, Bedienung als Kurzinbetriebnahmehilfe angegeben.

NCU-Link

Die Funktionen zu NCU-Link basieren auf einer zusätzlichen Kommunikation zwischen NCUs im Interpolationstakt.

NCU-Link ermöglicht:

- Unterstellung einer physikalischen Achse unter verschiedene NCUs
- NCU-übergreifende Interpolation
- Erhöhung der Anzahl benutzbarer Achsen für einen NCU-Verband
- Erhöhung der Anzahl der Kanäle für einen NCU-Verband
- Bereitstellung der Achsdaten und Signale an der NCU, der eine nicht lokale Achse temporär zugeordnet ist
- Anwenderkommunikation über den NCU-Verband durch Linkvariablen

Lead-Link-Achsen

Folgeachsen können von einer NCU verfahren werden, wenn die zugehörige Leitachse von einer anderen NCU verfahren wird. Der erforderliche Austausch der Achsdaten erfolgt über NCU-Link-Kommunikation.

NCU-Link mit unterschiedlichem IPO-Takt

Es ist möglich, eine NCU-Link-Verbindung zwischen NCUs mit unterschiedlichem Interpolationstakt für besondere Anwendungen, wie z. B. Unrunddrehen, zu benutzen.

Fertigungsleitrechner

Die Kommunikation zwischen Leitrechnern und Bedieneinheiten ist beschrieben in:
Literatur:
Funktionsbeschreibung Rechnerkopplung RPC SINUMERIK

PLC-PLC-Kommunikation

DP-Master, DP-Slave, DP-DP-Koppler, Querkommunikation via PBK

Busleistungen

Die im obigen Bild dargestellten Busse sind für ihre Übertragungsaufgaben spezialisiert. Daraus folgen die im nächsten Bild dargestellten Kommunikationskenndaten:

- Anzahl Busteilnehmer
- Baudrate
- Synchronisation

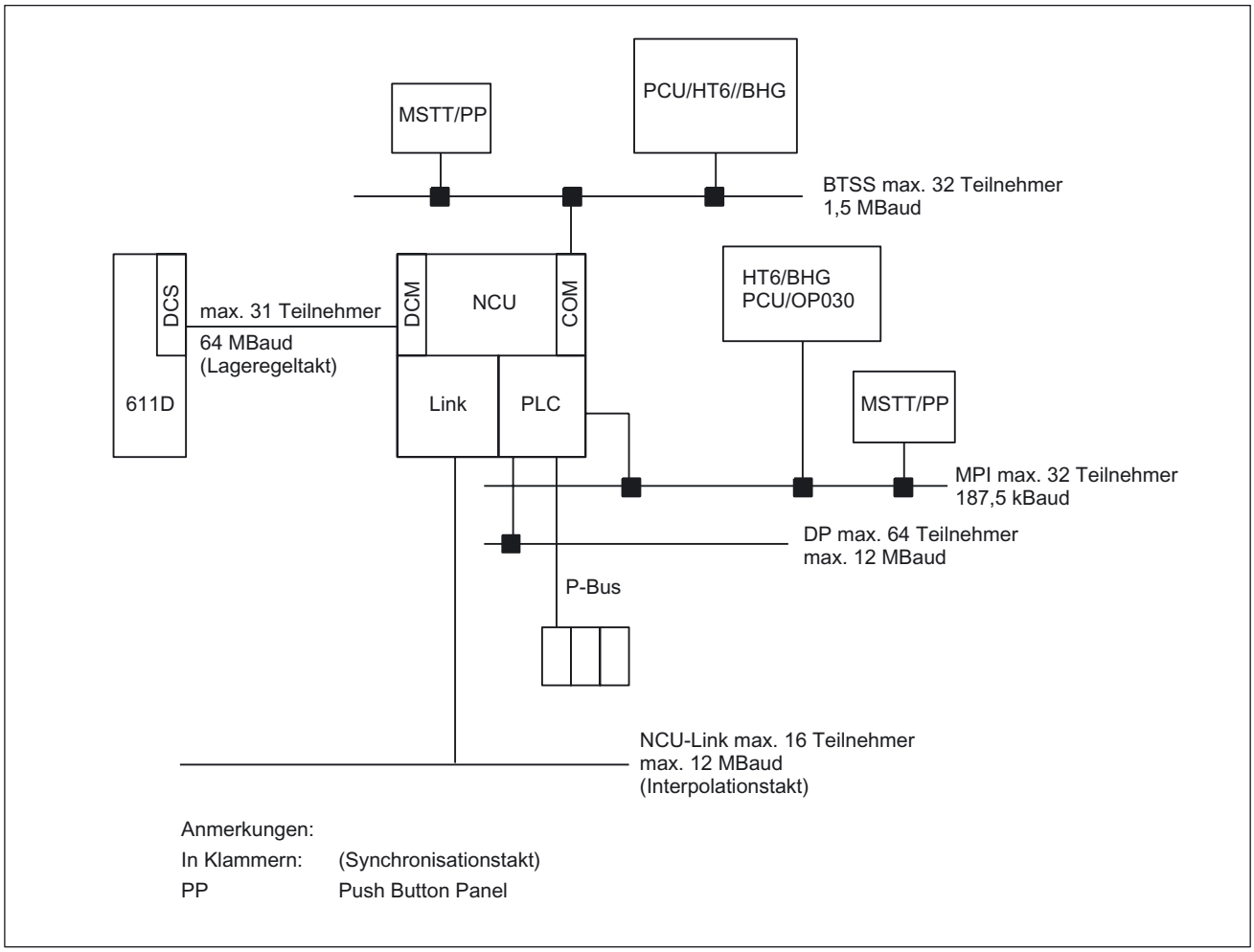


Bild 2-2 Buseigenschaften

Einstufung im 7-Schichten-Modell

Die Kommunikation ist auf folgenden Protokollebenen realisiert:

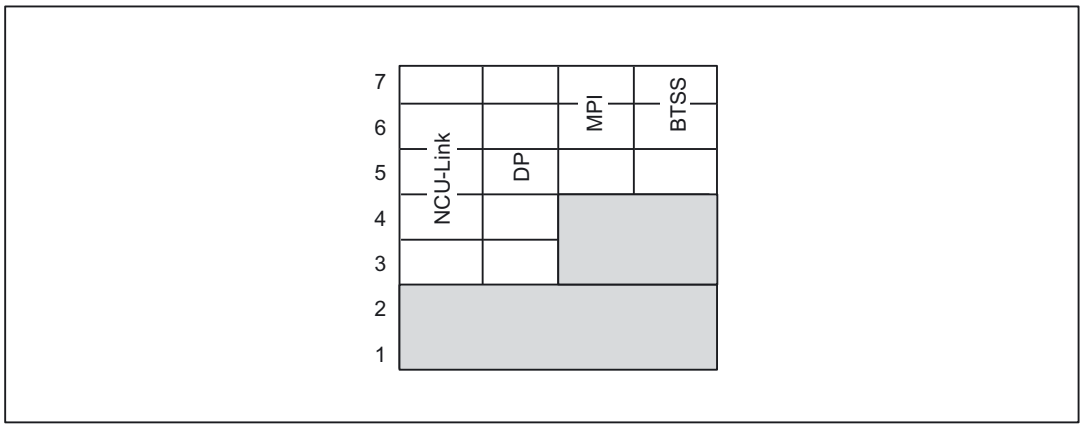


Bild 2-3 Protokollebenen des 7-Schichten-Modells

NCU-Link und DP beziehen Geschwindigkeitsvorteile aus dem unmittelbaren Aufsetzen auf Ebene 2.

2.1.2 Mehrere Bedientafeln und NCUs mit Bedieneinheitenmanagement (Option)

2.1.2.1 Allgemeines

Einführung

Komplexe Maschinen wie z.B. Rundtaktmaschinen, Mehrspindeldrehmaschinen und komplexe NC-Fertigungszentren stellen hohe Anforderungen an die Flexibilität der Anlagenkonfiguration.

Häufig benötigen sie:

- mehrere Bedieneinheiten (M) wegen großer Abmessungen der Maschine und räumlicher Trennung der Bedienplätze
- mehrere NCUs (N) wegen hoher Achsen- und Kanalanzahl

Abgrenzung

Die Standardlösung gilt für alle SW-Stände ohne die Option. Dabei sind noch Leistungsabstufungen in Abhängigkeit von SW-Ständen zu berücksichtigen (siehe "Randbedingungen").

Während die Standardlösung nur bestimmte, eingeschränkte M:N-Kombinationen erlaubt, ist mit der Option "Bedieneinheitenmanagement" jetzt eine flexible allgemeine Lösung für die oben genannten Anforderungen verfügbar.

2.1.2.2 Systemmerkmale

M:N-Konzept

Dieses Konzept erlaubt es dem Anwender, innerhalb der durch die Hardware gesetzten Grenzen beliebige Bedieneinheiten über den Bus mit beliebigen NCUs des Systems zu verbinden und nach Bedarf umzuschalten.

NCU-Link

NCU-Link ist eine zusätzliche Direktverbindung zwischen den NCUs, die schnelle Kommunikation ermöglicht.

Neue Merkmale

Neue Merkmale im Zusammenhang mit der Option "Bedieneinheitenmanagement" sind:

- Unabhängige Verbindung von PCU und MSTT
- Je NCU zwei eigenständige HMI-Verbindungen:
  - PCU und MSTT zusammen, oder PCU allein umschaltbar.
  - HMI-Zustände:
    - online/aktiv: bedienen und beobachten
    - online/passiv: zu sehen ist ein Bildrahmen mit Header und Alarmzeile und einem Hinweis auf "passiv"
    - offline
- Verschiedene Bussysteme (MPI/BTSS) zwischen PCU/MSTT/HT6 und NCUs

  (Änderungen nur mit Hochlauf möglich)
- HMI-Funktion als Server/als Haupt-, Nebenbedienfeld
- Feste und umschaltbare Bedientafeln können gemischt werden
- Verdrängungsmechanismus (prioritätsgesteuert), wenn mehr als zwei PCUs sich um eine NCU-Verbindung bewerben
- Bis zu 32 Busteilnehmer (PCUs, MSTTs, HT6s und/oder NCUs)
- PLC steuert den Umschaltvorgang (Bedieneinheitenumschaltung auf Toolbox, Verzeichnis PSP_PROG\m_zu_n.zip)
- Konfigurationsdatei NETNAMES.INI mit neuen Parametern

Randbedingungen

Zu einem Zeitpunkt:

- können maximal zwei PCU/HT6 an einer NCU online sein
- kann maximal einer davon im aktiven Zustand sein
- muss beim HT6 für die PCU- und die MSTT-Adresse derselbe Wert eingetragen werden (da die MSTT-Adressen nur Werte bis 15 annehmen können, ergibt sich daraus auch eine Begrenzung für die PCU-Adresse)

2.1.2.3 Hardware

Bedientafelfronten

Die Bedientafelfronten OP/TP enthalten Flachbildschirm, Softkeys, Tastenfeld, Schnittstellen und Stromversorgung.

Maschinensteuertafel

Die Maschinensteuertafel (MSTT/MCP) enthält Tastenfeld, Drehknopffeld und Schnittstellen.

HT6

Das Handheld Terminal HT6 enthält Flachbildschirm, Softkeys, Tastenfeld, Override-Drehschalter, Notaus- und Zustimmungstaster, sowie Schnittstellen und Stromversorgung. Im HT6 sind Funktionalitäten von Bedientafel und MSTT zusammengefasst.

Unterschied zwischen OP030 und HT6

OP030 und HT6 unterscheiden sich in der Möglichkeit der Zuordnung zu einer NCU:

- OP030

  Kann nur einer NCU fest zugeordnet werden. Für diese kann sie als zweite Bedientafelfront genutzt werden. Hierzu sind die Adressen der angeschlossenen Partner einstellbar.

- HT6

  Kann aktiv über HT6/HMI-Bedienung einer anderen NCU zugeordnet werden.

Literatur

Die Bedienoberflächen sind in den Bedienungsanleitungen zu den verwendeten Bedientafelfronten beschrieben:

/FBO/ Funktionshandbuch Projektierung Bedienoberfläche OP030

/BHsl/ Gerätehandbuch Bedienkomponenten

Busse

Die Bedieneinheiten (PCUs und/oder MSTTs, HT6s) und die NCUs werden verbunden über:

- MPI-Bus (Multi Point Interface; 187,5 kBaud)

oder

- BTSS-Bus (Bedientafelfront-SchnittStelle; 1,5 MBaud).

Mischung der verschiedenen Bussysteme innerhalb der Anlage ist möglich.

Adressenbelegung

Busteilnehmer haben eine eindeutige Adresse am Bus.

Eine NCU belegt:

- an der **BTSS** eine gemeinsame Adresse für NC und PLC
- an der **MPI-Schnittstelle** zwei Adressen (getrennt für NC und PLC)

  Dabei gilt:

  – PLC-Adresse kann mit STEP7 umkonfiguriert werden.
  Voreingestellte Adresse für die PLC an der MPI ist 2.

  – Für die Adressen an der MPI-Schnittstelle gilt:
  NC-Adresse = PLC-Adresse + 1

Voreinstellungen für BTSS

Werksseitig sind die Adressen 0 (PG-Diagnose) bzw. 14 (HT6) reserviert und 13 (NCK) voreingestellt und sollten bei M:N-Anlagen nicht mit Busteilnehmern belegt werden.

- **Adresse 0**

  Ist für PG-Diagnose reserviert.

  Hinweis

  Die M:N-Umschaltung kann gestört sein, wenn ein PG online ist.

  Abhilfe:

  Entweder PG vor der Einheitenumschaltung offline schalten oder an die MPI-Schnittstelle anschließen.

- **Adresse 13**
  Ist für Service-/Inbetriebnahme voreingestellt.

  Umkonfiguration dieser Adresse über HMI-Bedienung ist möglich. Adresse für den Servicefall "NCU-Tausch" möglichst freihalten.

Voreinstellungen für MPI

- **Adresse 2**

  für PLC

- **Adresse 3**

  für NCU

Anzahl aktiver PCUs/HT6 an 1 NCU

An einer NCU dürfen maximal zwei PCUs/HT6 (inkl. COROS-OPs) ständig aktiv verbunden sein. Dabei zählen PCUs/HT6s an BTSS bzw. MPI in gleicher Weise.

Anzahl MSTTs/BHGs an 1 NCU

An einer NCU können standardmäßig zwei MSTTs und ein BHG wahlfrei an der BTSS oder MPI-Schnittstelle angeschlossen werden.

Hinweis

Die MPI/BTSS Netzwerkregeln entsprechend der "SINUMERIK 840D Inbetriebnahmeanleitung" sind unbedingt zu beachten.

Insbesondere darf eine M:N-Anlage nur mit Kabeln aufgebaut werden, die mit Abschlusswiderständen versehen sind (erkennbar am Schalter, mit dem diese an- und abschaltbar sind).

2.1.2.4 Funktionen

Festlegen der Eigenschaften

Die Zuordnung der HMI-Eigenschaften kann **statisch** oder **dynamisch** sein.

Statische Eigenschaften

Statische Systemeigenschaften werden in der Datei NETNAMES.INI projektiert. Sie werden beim Hochlauf wirksam und können während der Laufzeit nicht geändert werden.

Statische Eigenschaften sind:

- Zuordnung Busteilnehmer - Bussystem
- Mischung verschiedener Bussysteme (BTSS, MPI)
- Zuordnung HMI - NCU (welche PCUs/HT6s können auf welche NCUs sehen)
- MSTT-Umschaltung
- Verdrängungs-Prioritäten beim Umschalten (s. u.)
- Nutzungseigenschaften:
  - Bedieneinheit ist Alarm-/Datenhaltungs-**Server**
  - Bedieneinheit ist Haupt- bzw. Neben-Bedienfeld

Dynamische Eigenschaften

Die dynamischen Eigenschaften können während der Laufzeit geändert werden.

Es sind die Zustände:

<table>
<tr><th colspan="2">Online</th><th>Offline</th></tr>
<tr><td colspan="2">Normale HMI-Betriebsart mit Kommunikation zwischen PCU/HT6 und NCU: Bedienen und/oder Beobachten möglich.</td><td rowspan="3">Keine Kommunikation zwischen PCU/HT6 und NCU:
Bedienen und Beobachten nicht möglich.</td></tr>
<tr><td>aktiv</td><td>passiv</td></tr>
<tr><td>Bediener kann bedienen und beobachten.</td><td>Bediener kann nicht bedienen. Er sieht einen Bildrahmen mit Header und Alarmzeile und einem Hinweis auf "passiv".</td></tr>
</table>

Bedieneinheiten-Umschaltung ist **freigegeben**.	Bedieneinheiten-Umschaltung ist **gesperrt**.

Bedienung der Funktion M:N

Die Funktion M:N wird über die Option "Kanalmenü" bedient.

Das Kanalmenü wird über die Taste "Kanal umschalten" angewählt.

Mit den horizontalen Softkeys wählen Sie eine Kanalgruppe an (HMI Embedded/HT6: max. 8, HMI Advanced: max. 24 Kanalgruppen). In einer Kanalgruppe können acht Verbindungen zu Kanälen verschiedener NCUs aufgenommen werden.

Im Bild "Kanalmenü" werden alle aktuellen Verbindungen und die zugehörigen Symbolnamen angezeigt.

Verdrängungs-Strategie

Sind zwei PCUs/HT6s an einer NCU online, und ein dritter PCU/HT6 möchte online gehen, so kann er einen von den beiden verdrängen. Der Betroffene unterbricht dann die Kommunikation zur NCU.

Der dafür zuständige Verdrängungsalgorithmus wird von Prioritäten gesteuert, die in der Datei NETNAMES.INI projektiert sind.

2.1.2.5 Projektierbarkeit

NETNAMES.INI

Dem M:N-System müssen beim Hochlauf die existierenden Bedieneinheiten, NCUs und Kommunikationsverbindungen sowie deren Eigenschaften bekannt sein.

Diese Informationen sind in der vor dem Hochlauf projektierten Konfigurationsdatei NETNAMES.INI hinterlegt.

Hauptziel der vorliegenden Beschreibung ist es, die erforderlichen Kenntnisse zu vermitteln, diese Konfigurationsdatei bezüglich des M:N-Konzepts korrekt aufzubauen.

Das bedeutet, dass:

- die HW-Konfiguration abgebildet ist
- die Eigenschaften der Komponenten definiert sind
- die gewünschten Umschaltungen/Zuordnungen möglich sind

2.1.3 Mehrere Bedientafelfronten und NCUs Standardlösung

2.1.3.1 Systemmerkmale

Allgemeines

Das Folgende gilt für alle M:N-Anwendungen, die nicht die Option "Bedieneinheitenmanagement" benutzen.

Hinweis

Dieser Abschnitt trifft auf das HT6 nicht zu, da ohne Bedieneinheitenmanagement immer nur ein HT6 an einer NCU betrieben werden kann.

Systemmerkmale

- Die Verbindung zwischen PCU und NCU erfolgt über:
  - MPI-Bus (187,5 kBaud)
  - BTSS (1,5 MBaud)
- Folgende Konfigurationen sind möglich:
  - "Eine Bedientafelfront und bis zu drei NCUs"
  - "Eine Bedientafelfront und bis zu vier NCUs"
  - "Mehrere Bedientafelfronten und NCUs"
    Anschluss der Bedientafeln über die BTSS-Schnittstelle (X101 auf NCU) und die MPI-Schnittstelle (X122 auf NCU).
- Eine von den Bedientafelfronten muss OP030 sein.
- An einer NCU können zwei MSTTs und ein BHG an MPI oder BTSS angeschlossen werden.
- Die notwendige Konfiguration in der NC für den Anschluss der MSTTs/BHGs erfolgt mit dem PLC-Grundprogramm (siehe Funktionsbeschreibung, P3: PLC-Grundprogramm).
- Adressenvorgabe beim Datenaustausch zwischen PLCs über Profibus-DP bzw. bei Globale Daten (Doppeladressierung) ab PLC-CPU 315.
  Es gilt: NC-Adresse = PLC-Adresse + 1.
- Die maximale Zahl der Busteilnehmer beträgt 32.

2.1.3.2 Funktionen

Umschaltung der Verbindung auf andere NCU mit dem Softkey "Verbindungen"

Ein Menü erscheint, in dem Sie die Verbindungen conn_1, ... conn_n (in NETNAMES.INI vereinbart) über Softkeys auswählen können.
Auf den Softkeys wird der in NETNAMES.INI der Verbindung zugeordnete Name (name= ...) dargestellt.

Betätigen eines Softkeys stellt entsprechende Verbindung mit einer neuen NCU her.

Umschaltverhalten OP030

Keine Online-Umschaltung auf anderen Busteilnehmer möglich. In NETNAMES.INI ist eine Verbindung fest projektiert.

Umschaltverhalten HMI Embedded

Softkey "Verbindungen" wird nur angeboten, wenn in NETNAMES.INI mehr als eine Verbindung projektiert ist. Beim Wechsel auf die neue NCU wird die bisher bestehende Verbindung zu einer anderen NCU abgebrochen.

HMI-Anwendungen dürfen im Umschaltzeitpunkt keine Verbindung zur bisherigen NCU mehr benötigen (z.B. aktive Datensicherung über V.24). Ansonsten gibt die Steuerung eine entsprechende Meldung aus.

Bezüglich der NCU, auf die umgeschaltet wird, verhält sich HMI wie nach einem Neustart und befindet sich in dem Bedienbereich, der als Start-Bedienbereich eingestellt ist.

Umschaltverhalten HMI Advanced

Softkey "Verbindungen" wird nur angeboten, wenn an der Steuerung die Funktion "M:N" aktiviert ist. Aktivieren von " M:N " erfolgt im Menü "Inbetriebnahme/HMI/Bedientafel". Verbindungen bleiben bei Umschaltvorgängen bestehen und damit auch die Anwendungen aktiv, die diese Verbindungen benutzt haben. Nach Umschaltung befindet sich HMI bezüglich der neuen NCU im gleichen Bedienbereich, in dem er sich zuvor bezüglich einer anderen NCU befand.

Mögliche Störungen

NCU, mit der die Verbindung aufgenommen werden soll, kann dies ablehnen. Grund: NCU gestört oder NCU kann zu dem Zeitpunkt keine weitere Bedieneinheit mehr bedienen.

Im Maschinendatum MD10134 $MN_MM_NUM_MMC_UNITS (Anzahl gleichzeitig möglicher HMI-Kommunikationspartner) ist projektiert, wieviele Bedieneinheiten eine NCU zu einem Zeitpunkt bearbeiten kann.

OP030 verbraucht eine Einheit. Eine PCU verbraucht im Auslieferungszustand zwei Einheiten. Bei größeren OEM-Paketen sind weitere Einheiten (bis zu 12) erforderlich.

Alarme, Meldungen

HMI Embedded, OP030	HMI Advanced
Es können nur die Alarme der NCU ausgegeben werden, mit der momentan Verbindung besteht.	Alarme und Meldungen aller verbundener NCUs können gleichzeitig verarbeitet werden.

Alarmtexthaltung

HMI Embedded, OP030	HMI Advanced
Auf der Bedienkomponente können nur die Alarmtexte in einer Ausprägung gehalten werden. Die Standardalarmtexte sind für alle NCUs gleich und einmalig vorhanden. In dem Bereich für Anwenderalarme müssen die möglichen Alarme aller angeschlossenen NCUs hinterlegt werden.	NCU-spezifische Anwenderalarmtexte können nicht angelegt werden.

Verbindungskontrolle HMI

Adresse einer angeschlossenen NCU (nur am BTSS-Bus) kann im Menü "Verbindungen/Service" verändert werden. Die neue Adresse der NCU wird auf der NCU gespeichert.

Softkey "Service" wird nur angeboten, wenn Kennwort für Schutzstufe Service" eingegeben ist. "

Bei Inbetriebnahme der Funktion zum Ändern der Adresse immer eine Direktverbindung HMI zur entsprechenden NCU aufbauen, um sicherzugehen, dass die Adresse nur einmal am Bus vorhanden ist.

Hinweis

Beim Tausch der NCU (Servicefall) bzw. beim Ausfall der Pufferbatterie geht die Speicherung der Adresse verloren.

Durch Urlöschen der NCU geht die Adresse der NCU nicht verloren.

Eine Änderung der Adresse ist nur über die HMI-Software möglich.

Die Anzeige der aktuellen Verbindung im Grundbild sollte durch eindeutige Vergabe des Kanalnamens im folgenden Maschinendatum erfolgen:
MD20000 $MC_CHAN_NAME (Kanalname)

Funktion M:N

Bedienen der Funktion M:N über die Option "Bedieneinheitenmanagement".

Voraussetzung: Projektierung über die Datei NETNAMES.INI

Literatur:
/IAD/ Inbetriebnahmeanleitung 840D

Das Kanalmenü wird über die Taste "Kanal umschalten" angewählt. Mit den horizontalen Softkeys wählen Sie eine Kanalgruppe an (HMI Embedded: max. 8, HMI Advanced: max. 24 Kanalgruppen), in einer Kanalgruppe können acht Verbindungen zu Kanälen verschiedener NCUs aufgenommen werden. Im Bild "Kanalmenü" werden alle aktuellen Verbindungen und die zugehörigen Symbolnamen angezeigt.

Hinweis

Bei Fehlern im Hochlauf (wenn z.B. keine Verbindung aufgebaut wird), siehe Kapitel "Hochlauf".

2.1.3.3 Projektierbarkeit

2 Bedieneinheiten : 1 NCU

Die Verbindung zwischen zwei Bedieneinheiten und einer NCU zeigt das folgende Bild. Die MSTT ist dabei fest der NCU zugeordnet.

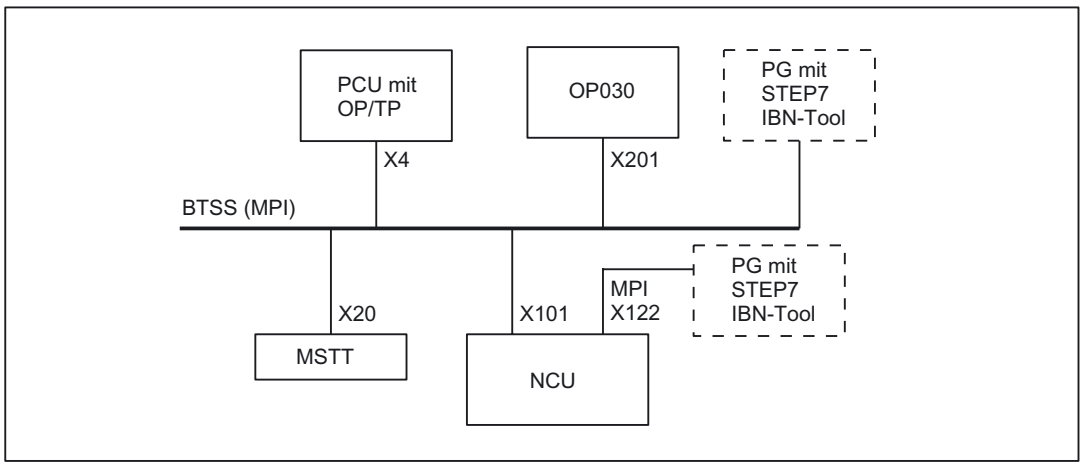


Bild 2-4 Beispiel-Konfiguration M:N entspricht 2:1

Die Bedieneinheiten, die NCU und die Maschinensteuertafel hängen entweder alle am BTSS-Bus oder alle am MPI-Bus. Es muss bezüglich dieser Komponenten ein **homogenes** Netz vorliegen.

Die skizzierte Konfiguration ermöglicht es, z. B. eine große Werkzeugmaschine jeweils an der Front– und Rückseite mit einer Bedieneinheit auszustatten.

Merkmale

Im Betrieb von zwei Bedieneinheiten in der skizzierten Konfiguration stellt der Benutzer folgendes Verhalten fest:

- Die Eingaben von den Bedieneinheiten sind gleichwertig gegenüber der NCU.
- Jede Bedieneinheit bekommt unabhängig von der anderen die Anzeigen zu sehen, die sie sich anwählt.
- Spontane Ereignisse wie Alarme werden an beiden Bedieneinheiten angezeigt.
- Die an einer Bedieneinheit eingestellte Schutzstufe ist auch für den zweite gültig.
- Das System leistet keine weiterführenden Koordinierungen der Bedieneinheiten.

Verwendet der Benutzer die skizzierte Standardkonfiguration, so sind keine besonderen Einstellungen erforderlich.

1 Bedientafel : 3 NCUs

Eine Bedientafel kann mit bis zu drei NCUs verbunden werden (siehe folgendes Bild). Die MSTT ist dabei fest der jeweiligen NCU zugeordnet.

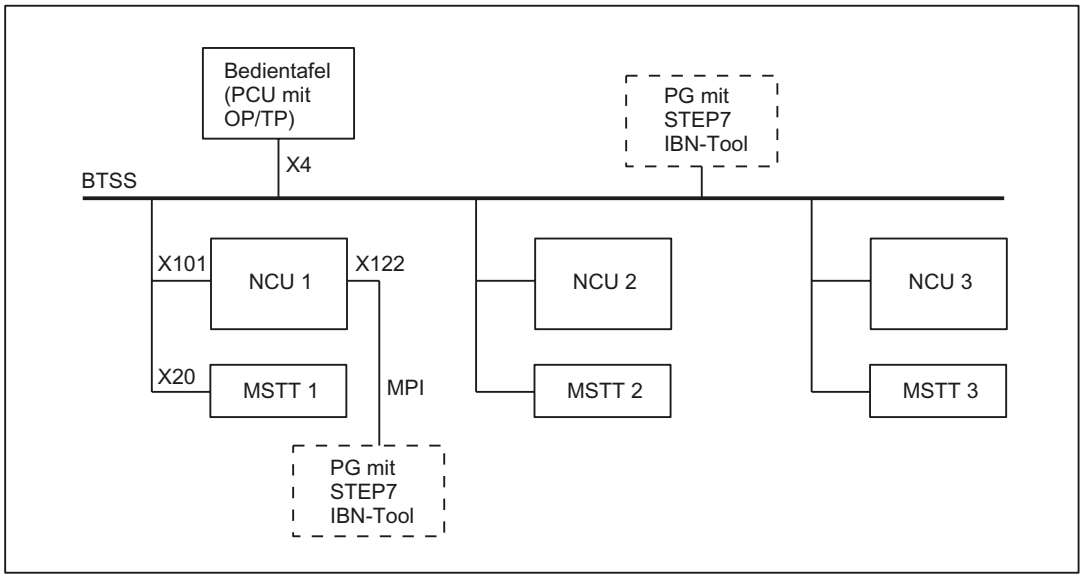


Bild 2-5 Beispiel-Konfiguration M:N entspricht 1:3

Es können mehrere NCUs von einer Bedientafel aus bedient werden (mehrere autonome Maschinen oder eine große Maschine mit mehreren NCUs). Dabei ist zu einem gegebenen Zeitpunkt immer nur eine zuvor angewählte NCU für Bedienungen mit der Bedientafel verbunden.

- HMI Embedded unterhält auch für Alarme nur eine Verbindung.
- HMI Advanced: Die PCU bleibt für Alarme mit allen angeschlossenen NCUs verbunden.

Merkmale

Merkmale für den Betrieb mehrerer NCUs und einer Bedientafel sind:

- NCU-Bedienung:

  Der Benutzer muss die NCU, die er bedienen will, durch einen Softkey anwählen.

  Er sieht dann am Bedienbild im Bedienbereich "Verbinden" die Bezeichnung der Verbindung und der NCU, mit der aktuell Verbindung besteht.

- HMI Embedded:
  - Auf der Verbindung, die beim Umschalten auf eine andere NCU unterbrochen wird, soll keine Anwendung mehr aktiv sein (Beispiel: Datensichern über V24). Wird der Umschaltversuch bei aktiver Anwendung unternommen, erfolgt die Systemmeldung "V24 aktiv".
  - Für die neu hergestellte Verbindung befindet sich HMI im voreingestellten Start-Bedienbereich (wie nach Neustart HMI).
- HMI Advanced:

  Nach Aufnahme der Verbindung mit einer anderen NCU steht für diese sofort der zuletzt bediente Bedienbereich zur Verfügung (wie bei der NCU, deren Verbindung getrennt wurde).

OEM-Lösung

Als OEM-Lösung kann eine PCU mit HMI-Advanced über eine BTSS an bis zu drei NCUs als Programm- und Alarm-Server angeschlossen werden (m=1, n=3).

Zusätzlich kann noch ein PG mit einem IBN-Tool angeschlossen werden.

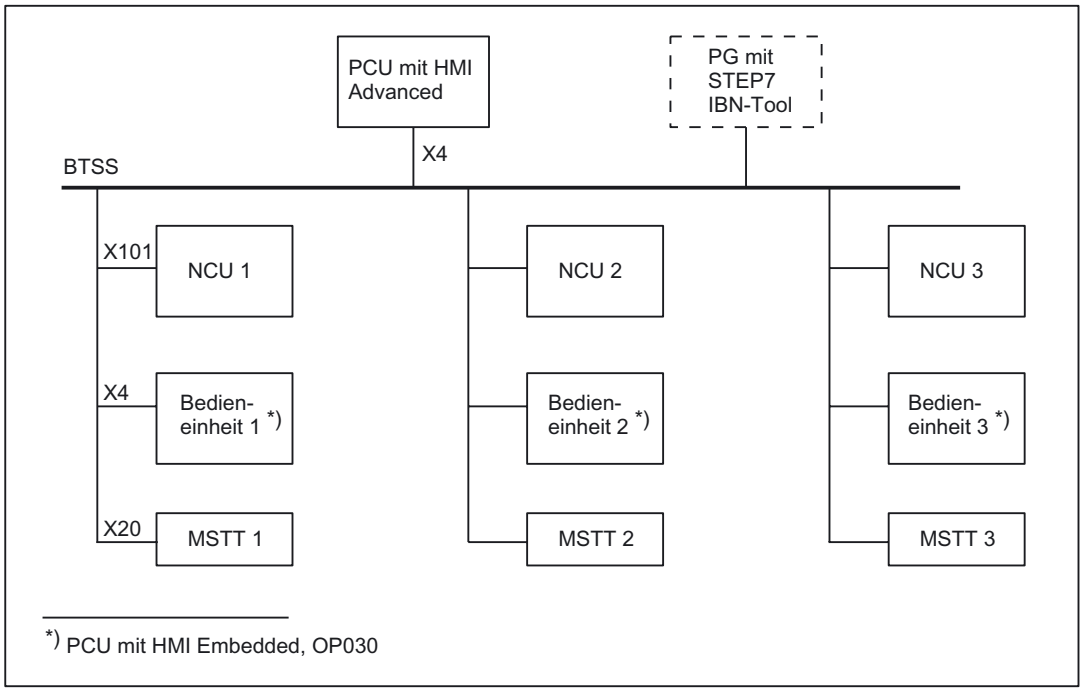


Bild 2-6 Beispiel-Konfiguration für OEM-Lösung

Merkmale

Folgende Merkmale kennzeichnen die im obigen Bild skizzierte OEM-Lösung:

- NCU-Bedienung:

  Der Benutzer muss die NCU, die er bedienen will, durch einen Softkey anwählen.

  Er sieht dann am Bedienbild die Bezeichnung der Verbindung und der NCU, mit der aktuell Verbindung besteht.

- HMI Embedded

  Kann nur an eine lokale NCU verbunden werden.

- HMI Advanced

  Nach Aufnahme der Verbindung mit einer anderen NCU steht für diese sofort der zuletzt bediente Bedienbereich zur Verfügung (wie bei der NCU, deren Verbindung getrennt wurde).

1 Bedientafel : 4 NCUs

Zusätzlich zu den oben genannten Möglichkeiten kann die Verbindung zwischen einer Bedientafel mit HMI Advanced und bis zu vier NCUs realisiert werden, wie es das folgende Bild zeigt. Die MSTT und die lokale Bedientafel mit HMI Embedded sind dabei fest der jeweiligen NCU zugeordnet.

Eine zweite Bedieneinheit kann an der BTSS angeschlossen werden.

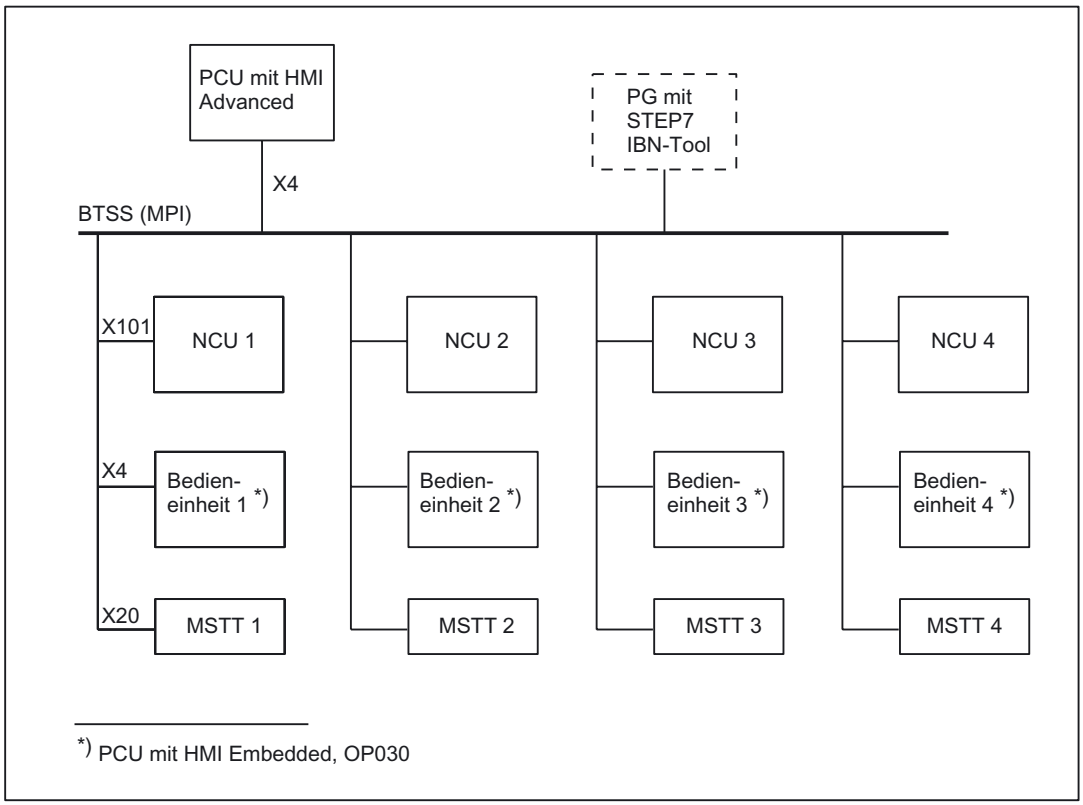


Bild 2-7 Beispiel-Konfiguration M:N entspricht 1:4

Zu einem gegebenen Zeitpunkt kann immer nur eine zuvor angewählte NCU mit der HMI Advanced-Bedientafel für Bedienungen verbunden werden:

- HMI Embedded unterhält auch für Alarme nur eine Verbindung.
- HMI Advanced: Die PCU bleibt für Alarme mit allen angeschlossenen NCUs verbunden.

Literatur

/BH/ Gerätehandbuch Bedienkomponenten

/IAD/ Inbetriebnahmeanleitung

/FB1/ Funktionshandbuch Grundfunktionen; PLC-Grundprogramm (P3)

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Speicherkonfiguration (S7)

Hier finden Sie beschrieben:

- Busaufbau MPI/BTSS, Busadressen (IAD)
- Busabschluss (IAD, FB2)
- Anschluss der MSTTs über PLC-Grundprogramm (FB1)
- DIP-FIX Einstellungen der MSTT (IAD)

2.1.3.4 MPI/BTSS-Netzwerkregeln

Netzinstallationen

Folgende Grundregeln sind bei Netzinstallationen zu beachten:

- Die Buslinie muss an **beiden Enden** abgeschlossen werden. Hierfür schalten Sie den Abschlusswiderstand im MPI-Stecker des ersten und letzten Teilnehmers ein, die übrigen Abschlusswiderstände aus.

  Hinweis

  Nur zwei eingelegte Abschlüsse sind erlaubt.

  Beim BHG/HT6 sind Busabschlusswiderstände im Gerät **fest** eingebaut.

- Mindestens 1 Abschluss muss mit 5V-Spannung versorgt werden.
  Dies ist automatisch gegeben, sobald der MPI-Stecker mit eingelegtem Abschlusswiderstand an einem eingeschalteten Gerät angeschlossen ist.
- Stichleitungen (zuführendes Kabel vom Bussegment zum Teilnehmer) sollten möglichst kurz sein.

  Hinweis

  Nicht belegte Stichleitungen sollten, wenn möglich, entfernt werden.

- Jeder MPI-Teilnehmer muss erst angesteckt, dann aktiviert werden.
  Beim Trennen eines MPI-Teilnehmers muss erst die Verbindung deaktiviert, dann der Stecker abgezogen werden.

- Pro Bussegment können je ein BHG und ein HT6 oder zwei BHGs bzw. HT6s angeschlossen werden. An den Verteilerboxen eines BHGs bzw. HT6s dürfen **keine** Busabschlüsse eingelegt werden.

  Falls nötig, kann der Anschluss von mehr als einem BHG/PHG an ein Bussegment mit zwischengeschaltetem Repeater erfolgen.

- Folgende Kabellängen für MPI bzw. BTSS für den Standardfall ohne Repeater dürfen nicht überschritten werden:

  MPI (187,5 kBaud): max. Kabellänge in Summe 1000 m

  BTSS (1.5 MBaud): max. Kabellänge in Summe 200 m

Hinweis

Huckepack-Stecker werden bei Netzverbindungen nicht empfohlen.

Weitere Informationen zur Bus-Kommunikation siehe:
Literatur:
/IAD/ Inbetriebnahmeanleitung 840D

2.1.4 NCU-Link

2.1.4.1 Allgemeines

Einsatz

NCU-Link, die Verbindung zwischen mehreren NCU-Einheiten einer Anlage, findet ihren Einsatz in Anlagen mit dezentralem Systemaufbau.

Bei großem Bedarf an Achsen und Kanälen z. B. bei Rundtaktmaschinen und Mehrspindelmaschinen können die Rechenleistung, die Konfigurationsmöglichkeiten und Speicherbereiche mit einer einzigen NCU an praktische Grenzen stoßen. Mehrere mit einem NCU-Link-Modul vernetzte NCUs bieten eine nach oben offene Lösung, die insgesamt die Aufgaben solcher Werkzeugmaschinen erfüllt. Das NCU-Link-Modul realisiert eine schnelle NCU-NCU-Kommunikation auf Basis einer synchronisierten 12 MBaud PROFIBUS-Schnittstelle.

Hinweis

NCU-Link ist verfügbar bei SINUMERIK 840D mit Link-Modul und HMI Advanced.

⚠WARNUNG

Wird an Maschinen mit NCU-Link Safety Integrated (SI) genutzt, so ist dies möglich, wenn:

- die Projektierung der systemintegrierten SI-Funktionen die Link-, Containerlink- und/oder Lead-Linkachsen nicht mit einschließt.
- bei der Projektierung der sicherheitsgerichteten Ein-/Ausgänge (SGE/SGA) kein sicherheitsgerichteter Signalaustausch zwischen den beteiligten Steuerungen benötigt wird, bzw. der Signalaustausch zwischen den beteiligten Steuerungen ausschließlich über sicherheitsgerichtete Ein-/Ausgänge (SGE/SGA) entsprechend der Funktionsbeschreibung Safety Integrated erfolgt (z. B. Berücksichtigung des 3-Klemmenkonzepts).

Werden axiale systemintegrierte SI-Funktionen von Link- oder Containerlinkachsen dennoch genutzt, so erfolgt dies in Verantwortung des Maschinenherstellers. Dabei existieren u. a. Randbedingungen bei den SI-Funktionen Stop D/E, SG, SE, SN, sowie bei Inbetriebnahme, Diagnose und beim SI-Abnahmetest.

Unterstützung kann das Centre of Competence Service (CoCS) - SINUMERIK Safety Integrated leisten. Bitte kontaktieren Sie die im Vorwort unter Technical Support angegebenen Telefonnummern und Adressen.

2.1.4.2 Ausführungsformen dezentraler Maschinen

Merkmale der Maschinen

Rundtaktmaschinen/Mehrspindelmaschinen weisen folgende Besonderheiten auf:

- Globale, übergreifende (nicht einer Station zuordenbare) Einheiten:
  - Trommel-/Rundschaltachse und
  - von Station (Lage) zu Station (Lage) wandernde Einheiten:

    bei Rundtaktmaschinen: → die Rundachse der Werkstückaufspannung für Mehrseitenbearbeitung

    bei Mehrspindelmaschine:
    → Spindel, Pinole
- stationsbezogene (lagebezogene), ortsfeste Einheiten:
  - Schlitten, Fräs- und Bohreinheiten, die für eine Bearbeitungsaufgabe an dem von Station zu Station wandernden Teil eingesetzt werden.

Einsatzgebiete

Rundtaktmaschinen (**RTM**) und Mehrspindelmaschinen (**MS**) werden als hochproduktive Maschinen in der Mittel- und Großserienfertigung eingesetzt.
Ihr wesentlicher Vorteil ist, dass das Werkstück in einer Aufspannung viele Bearbeitungsschritte erfahren kann.

NCU-Zuordnungen

Abhängig vom Ausbau der RTM/MS werden die zahlreichen Achsen solcher Maschinen verschiedenen NCUs zugeordnet (z. B. für jede Bearbeitungseinheit oder eine Gruppe von Bearbeitungseinheiten eine NCU). Die globalen Einheiten werden einer getrennten NCU zugeordnet oder entsprechend verteilt.

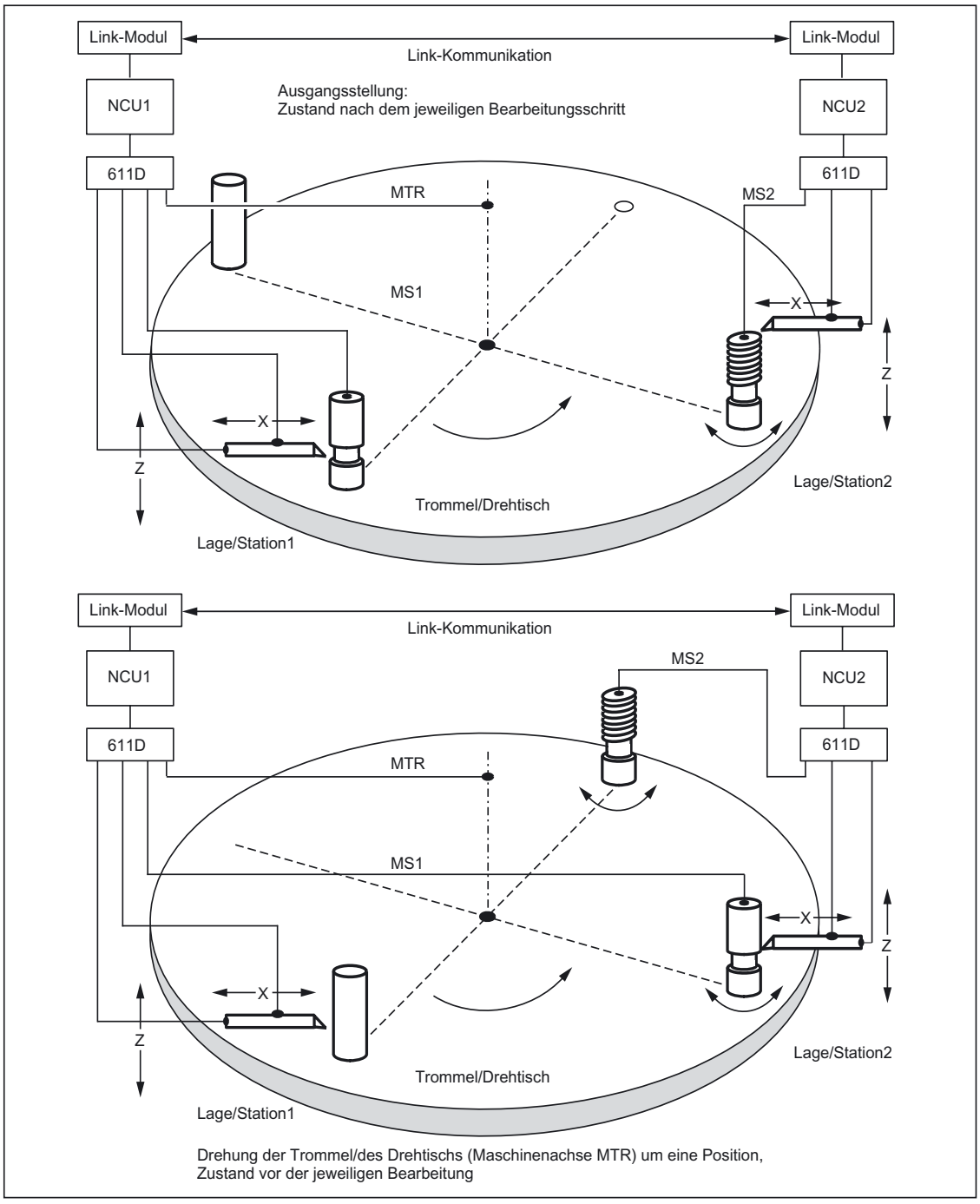


Bild 2-8 Trommelumschaltung, schematisch, Ausschnitt

Beim Weiterschalten des Rundtisches bei RTM oder der Trommel bei MS bewegt sich die Werkstück haltende Achse zur nächsten Bearbeitungseinheit.

Die Werkstück haltende Achse wird nun dem Kanal der Bearbeitungseinheit zugeordnet. Das ist im Beispiel auf einer anderen NCU, muss aber nicht so sein.

Wie obiges Bild zeigt, wird die Maschinenachse MS1, die physikalisch der NCU1 unterstellt ist, durch Drehung der Trommel/des Drehtisches in Lage/Station2 gebracht. Damit in Lage/Station2 nun eine koordinierte Bearbeitung zwischen Schlitten und Spindel möglich ist, werden in dieser Position die Befehle für die Spindel MS1 mit Hilfe der Link-Kommunikation übergeben. Die Spindeln MS1, MS2, ... sind **Link-Achsen**.

Physikalische Achsen können zu einem Zeitpunkt immer nur der Bewegungsführung eines NCU-Kanals unterstellt sein. Die Initiative der Bewegungsführung für eine Achse kann jedoch nacheinander verschiedenen NCU-Kanälen zugewiesen werden.

Lösungen

Für die Bereitstellung einer physikalischen Achse unter verschiedenen NCUs ist die Eigenschaft **Link-Achse** verfügbar. Im Schemabild oben wird MS1 nach der Drehung in Lage/Station2 zur Link-Achse aus Sicht der NCU2 (unteres Teilbild).

Für veränderliche, achsgruppenweise Zuordnung von Kanalachsen zu Maschinenachsen ist das Konfigurationskonzept **Achscontainer** verfügbar.

Alle Link-Achsen, die von der gleichen Trommel/Drehtisch weitergeschaltet werden, sind im gleichen Achscontainer zu verwalten.

2.1.4.3 Link-Achsen

Link-Achsen

- Definition:

  Eine Link-Achse ist eine Achse, deren Antriebssteuerung und Lageregelung einer anderen NCU untersteht oder die lokale Achse der betrachteten NCU ist, aber von einer anderen NCU angesprochen werden kann.
- Es muss die SW-Option "Link-Achse" vorliegen.

Koordinierungen

- Die wechselweise Nutzung einer physikalischen Achse durch mehrere NCUs setzt voraus, dass sich die beteiligten NCUs über den Status und Daten einer solchen Achse informieren und die Benutzungen der Achse koordinieren können.

Interpolation

- Lokale Achsen und Link-Achsen können unter der Bewegungsführung einer NCU gemeinsam interpoliert werden.

Hardware

- Die an der wechselseitigen Nutzung von Achsen über NCU-Grenzen hinweg beteiligten NCUs müssen mit einem **Link-Modul** ausgestattet sein. Das NCU-Linkmodul realisiert eine schnelle NCU-NCU-Kommunikation auf Basis einer synchronisierten 12-MBaud-Profibus-Schnittstelle.

Literatur:
/PHD/ Handbuch Projektierung NCU 840D

Die vorliegende Beschreibung gibt für die skizzierten dezentralen Maschinen die erforderlichen Informationen für Konfigurierung, Programmierung und Koordinierung.

2.1.4.4 Flexible Konfiguration

Achscontainer

Bei Rundtaktmaschinen/Mehrspindelmaschinen bewegen sich die Werkstück tragenden Achsen von einer Bearbeitungseinheit zur nächsten. Weil die Bearbeitungseinheiten verschiedenen NCU-Kanälen unterstehen, müssen bei einem Stations-/Lagewechsel die Werkstück tragenden Achsen dem entsprechenden NCU-Kanal dynamisch neu zugeordnet werden. Diesem Zweck dienen **Achscontainer**.

Zu einem Zeitpunkt ist immer nur eine Werkstück-Aufspannachse/Spindel an der lokalen Bearbeitungseinheit. Der Achscontainer stellt die Verbindungsmöglichkeiten zu allen Aufspannachsen/Spindeln zusammen, von denen immer nur genau **eine** für die Bearbeitungseinheit **aktiviert** ist.

Über Achscontainer können zugeordnet werden:

- lokale Achsen und/oder
- Link-Achsen

Der Wechsel der über einen Achscontainer definiert benutzbaren Achsen erfolgt durch zyklische Verschiebung (Drehung) der Einträge im Achscontainer.

Die Verschiebung kann durch das **Teileprogramm** ausgelöst werden.

Gültigkeitsbereich

Achscontainer mit Link-Achsen sind ein NCU-übergreifendes Betriebsmittel (NCU global), das durch die Steuerung koordiniert wird.

Achscontainer, in denen ausschließlich lokale Achsen verwaltet werden, sind möglich.

2.1.4.5 NCU-übergreifende Anwenderkommunikation

Link-Variablen

- Jede über Link-Modul verbundene NCU kann für alle verbundenen NCUs einheitlich ansprechbare **globale Link-Variablen** ansprechen. Link-Variablen sind als **Systemvariablen** programmierbar. Die Bedeutung dieser Variablen wird in der Regel durch den Maschinenhersteller festgelegt und dokumentiert.
- Anwendungen für Link-Variablen:
  - globale Maschinenzustände
  - Werkstück-Aufspannung offen/geschlossen
  - ...
- Datenvolumen vergleichsweise klein.
- Übertragungsgeschwindigkeit sehr hoch, weil für Austausch Hauptlauf-relevanter Informationen vorgesehen.
- Der Zugriff auf diese Systemvariablen ist aus dem **Teileprogramm** und aus **Synchronaktionen** heraus möglich. Die Größe des Speicherbereiches für globale Systemvariablen ist projektierbar.
- Bis alle beteiligten NCUs einen neu geschriebenen Wert in einer globalen Systemvariablen konsistent lesen können, vergeht ein Interpolationstakt.

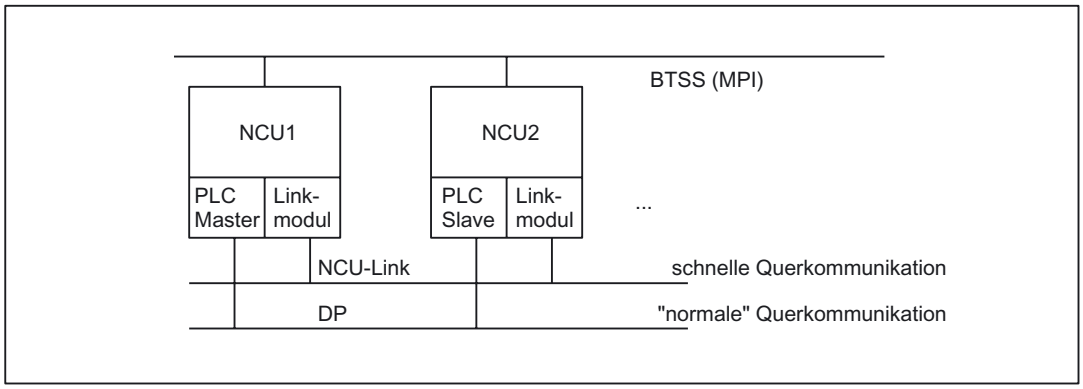


Bild 2-9 Übersicht Kommunikation

Hinweis

Bei Anlagen ohne NCU-Link können die Linkvariablen auch NCU-lokal als zusätzliches kanalübergreifendes Kommunikationsmittel genutzt werden.
In diesem Falle vergeht kein Interpolationstakt zwischen Schreiben und Lesen.

Literatur:
/FBSY/ Funktionshandbuch Synchronaktionen

2.1.4.6 Lead-Link-Achsen

Folgeachsbewegungen

Mit der unten gezeigten Konstellation wird es möglich, Folgeachsen einiger NCUs (im Bild als NCU2 bis NCU n gewählt) in Abhängigkeit von der Bewegung einer Leitachse an einer anderen NCU (im Beispiel an NCU 1) zu verfahren.

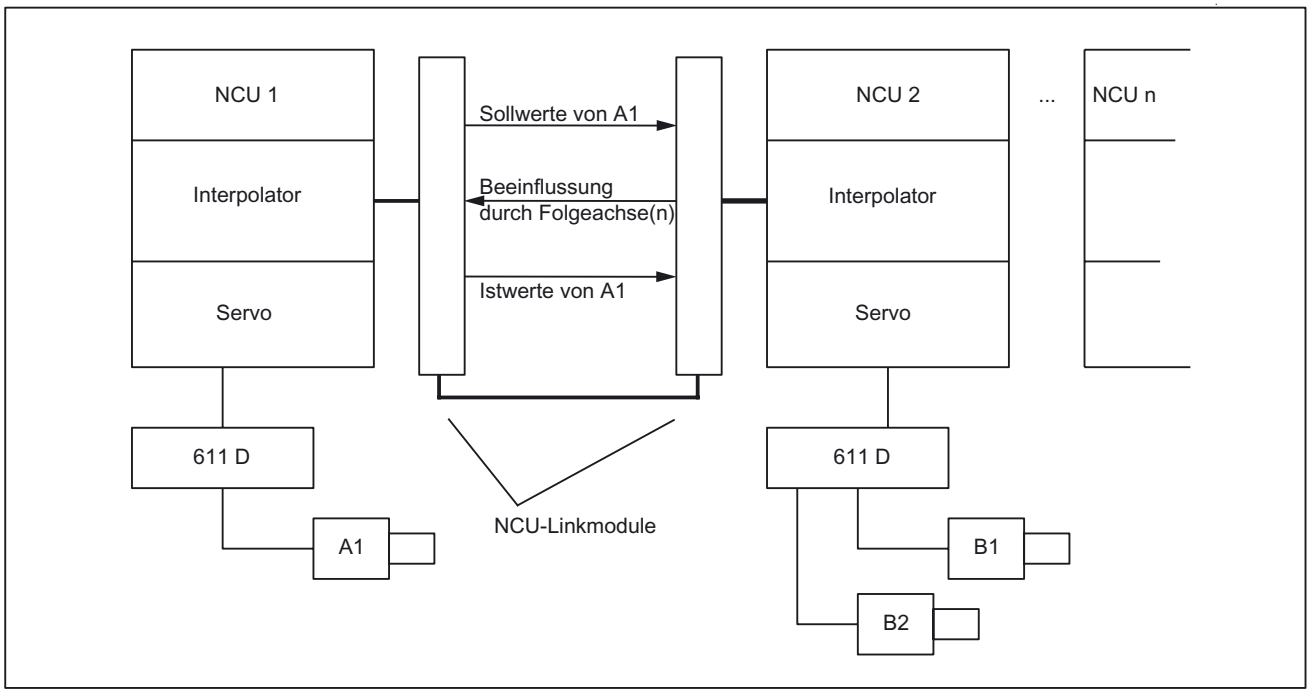


Bild 2-10 NCU 2 bewegt Folgeachsen durch Kopplung an eine Leitachse an NCU1

Bei Einhaltung der in der "Ausführlichen Beschreibung" genannten Randbedingungen können im gesamten NCU-Link-Verbund mehrere NCUs Leitachsen verfahren und andere NCUs Folgeachsbewegungen davon ableiten. Die Bewegung der Leitachsen darf nur von den NCUs erfolgen, an denen sie physikalisch angeschlossen sind. (Heimat-NCUs).

2.1.4.7 NCU-Link mit unterschiedlichem Interpolationstakt

Funktion

Die Erweiterung des Link-Konzeptes in der Weise, dass NCUs mit Link-Modulen verbunden sind, für die unterschiedliche Interpolationstakte parametriert sind, erschließt weitere Anwendungsmöglichkeiten. Weil bei unterschiedlichem Interpolationstakt eine der verbundenen NCUs den schnelleren Interpolationstakt hat, wird die Funktionalität auch "Fast-IPO-Link" genannt.

Anwendungsrichtung

Die NCU/NCUs mit normalem Interpolationstakt betreiben Achsen und Spindeln mit Standard-Anforderungen an Dynamik und Genauigkeit, während die NCU/NCUs mit dem schnelleren Interpolationstakt eine oder wenige Achsen mit hoher Dynamik und Genauigkeitsanforderung betreiben.

Beispiele:

Unrunddrehen (Nocken, Kolben o. Ä.)

- C und Z Achse normale Anforderungen
- X-Achse hohe Anforderungen an Dynamik und Genauigkeit

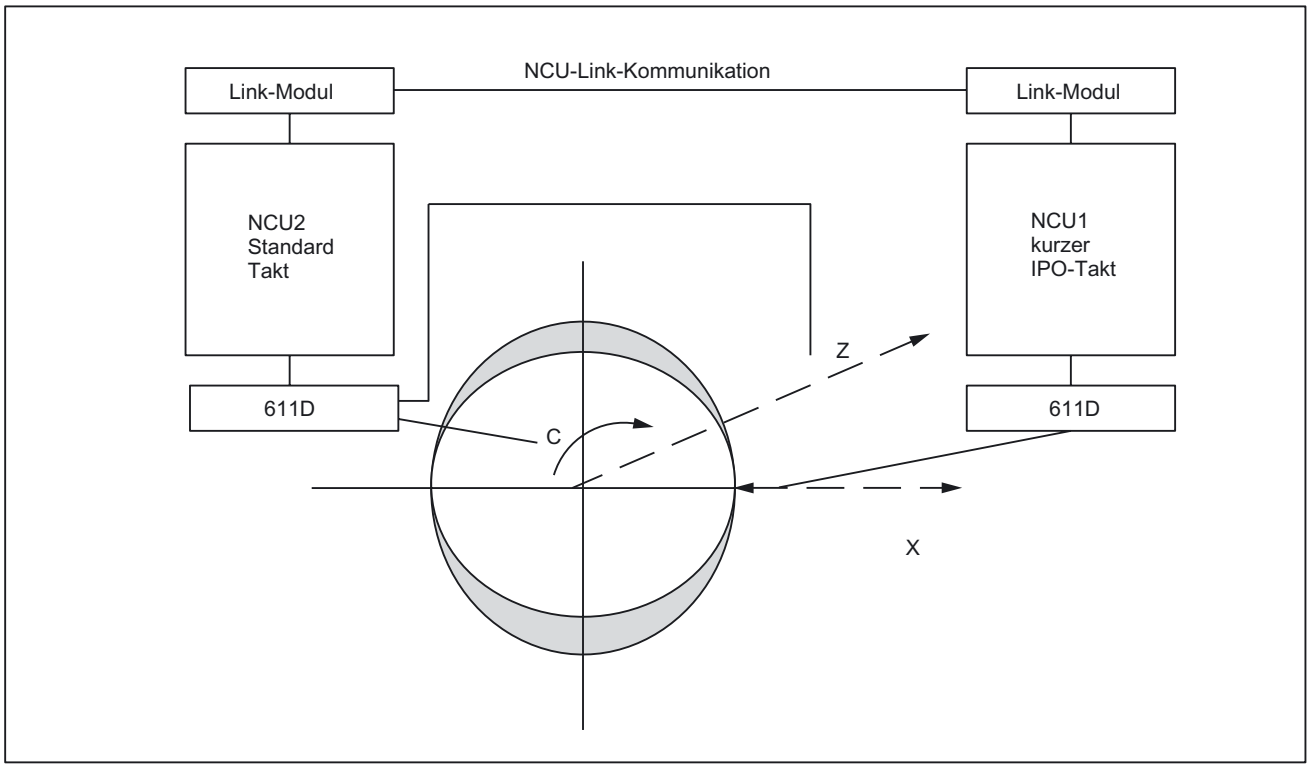


Bild 2-11 NCU-Link bei unterschiedlichem Interpolationstakt

Wesentliche Merkmale

- NCU-übergreifende Interpolation schneller (X) und normaler (C, Z) Achsen/Spindeln (siehe Bild).
- Das Teileprogramm läuft auf der NCU mit dem schnelleren Interpolationstakt und "sieht" die anderen Achsen als Link-Achsen bzw. Container-Link-Achsen.
- Die Kommunikation der beteiligten NCUs erfolgt über die Link-Module im Takt der langsameren NCU/NCUs (Link-Takt).
- Der langsamere Interpolationstakt ist ein ganzzahliges Vielfaches des schnelleren Interpolationstaktes.
- Aufgrund der unterschiedlichen Taktung sind die in der "Ausführlichen Beschreibung" angegebenen Randbedingungen einzuhalten.
- Konturgenauigkeit zwischen den schnellen und langsam interpolierten Achsen ist gegeben, wenn die langsam interpolierten Achsen nicht während des Bearbeitungseingriffes beschleunigt werden.

2.2 Mehrere Bedientafelfronten und NCUs mit Option Bedieneinheitenmanagement

Im folgenden Kapitel werden Vorbereitung und Realisierung der Bedienschritte des M:N-Konzepts im Einzelnen beschrieben.

Vorgehensweise

1. Festlegen der Konfiguration:
   - hardwaremäßig: durch Zusammenschalten über die Bussysteme
   - softwaremäßig: Projektierung der statischen Eigenschaften mit Hilfe der Konfigurationsdatei NETNAMES.INI (siehe folgende Absätze). Diese statischen Eigenschaften werden beim Hochlauf wirksam und können zur Laufzeit nicht geändert werden.
2. Bereitstellung der Bedieneinheitenumschaltung in den PLCs der betroffenen NCUs. Die Bedieneinheitenumschaltung ist Liefergegenstand in der Tool-Box und besteht aus mehreren Bausteinen. Sie leisten:
   - Überprüfung der Umschaltbedingungen
   - Prioritätsgemäße Verdrängung
   - Umschaltung
3. Dynamische Eigenschaften (wie online/offline-Zustände) können während der Laufzeit geändert werden, und zwar innerhalb der Grenzen, die durch die Datei NETNAMES.INI festgelegt sind.

2.2.1 Hardware-Struktur

Wie in Kapitel "Kurzbeschreibung" bereits dargestellt, kann eine komplexe Anlage aus M Bedieneinheiten und N NCUs bestehen.

Über den Bus (MPI und/oder BTSS) werden die Hardware-Komponenten miteinander verbunden. Die Beziehungen der Busteilnehmer untereinander (Identifikation, Eigenschaften, Zuordnung und Umschaltung) sind Software-gesteuert.

2.2.2 Eigenschaften

Client-Identifikation

Die Zuordnung Busteilnehmer zum Bussystem ist statisch und kann zur Laufzeit nicht geändert werden. Sie wird einmal in der Datei NETNAMES.INI projektiert.

Bustyp und Busadresse ergeben zusammen die Client-Identifikation (CLIENT_IDENT), mit der sich die Bedieneinheit an einer NCU anmeldet, um online zu gehen.

Eigenschaften

In einer M:N-Anlage gibt es Bedieneinheiten mit den folgenden Eigenschaften:

Server		Bedienfeld	
Hält ständige 1:N-Verbindung		Ist auf die verschiedenen NCUs umschaltbar und hält ständige 1:1-Verbindung (gleichzeitig nur eine!). Bediener kann bedienen und beobachten. Verbindung wird aufgebaut, wenn HMI online geht, und wieder abgebaut, wenn er offline geht.	
Alarmserver (HMI Advanced)	Datenhaltungsserver (HMI Advanced)	Hauptbedienfeld	Nebenbedienfeld
Empfängt die Alarme aller NCUs einer M:N-Anlage. Er hält aus seiner Sicht eine ständige 1:N-Verbindung. Der Prozess "Alarme empfangen" ist immer aktiv und läuft im Hintergrund ab.	Geht im Hochlauf alle für ihn in NETNAMES.INI projektierten Verbindungen ein und hält ständige 1:N-Verbindung. Kann im Rahmen des Joblisten-Konzepts Daten empfangen, verwalten und verteilen.	Beispiel: Hauptbedienfeld für Rundtaktmaschinen kann auf alle Bearbeitungsstationen geschaltet werden.	Beispiel: Nebenbedienfeld für Rundtaktmaschinen kann nur auf eine von zwei benachbarten Bearbeitungsstationen geschaltet werden.
Kann nicht verdrängt werden (siehe Kapitel "Verdrängung")	Kann nicht verdrängt werden.	Kann nicht verdrängt werden.	Kann von Haupt- oder Nebenbedienfeld verdrängt werden.

Verteilung der Eigenschaften auf die HMI-Typen:

Eigenschaft	HMI Advanced	HMI Embedded/HT6
Server	x	
Hauptbedienfeld	x	x
Nebenbedienfeld	x	x

HMI ist zugleich Server und Hauptbedienfeld

Als Server unterhält HMI ständig 1:N-Verbindungen und als Hauptbedienfeld eine umschaltbare 1:1-Verbindung.

Wird HMI als Bedienfeld auf eine andere NCU umgeschaltet, belegt er dieselbe Verbindung, die er als Server bereits unterhält. Es wird keine neue Verbindung hergestellt.

Zulässige Kombinationen in einer Anlage

Falls in einer M:N-Anlage ein Server (Alarm-/Datenhaltungsserver) existiert, ist er zugleich Hauptbedienfeld.

In einer M:N-Anlage darf es nur eine Bedieneinheit mit den folgenden Eigenschaften geben:

- Windows-HMI (HMI Advanced): Server und Hauptbedienfeld

oder

- Non-Windows-HMI (HMI Embedded/HT6): Hauptbedienfeld

Es kann beliebig viele Nebenbedienfelder geben.

Hinweis

Damit die Funktion **Abarbeiten von Extern** zur Verfügung steht, muss **eine** Bedientafel im System als **Server** ausgewiesen sein.

2.2.3 Konfigurationsdatei NETNAMES.INI

Die hardwareseitig freie Kombinierbarkeit der Komponenten (siehe Kapitel "HW-Struktur") macht es erforderlich, dem System mitzuteilen, welche Komponenten auf welche Weise miteinander verbunden sind und zusammenarbeiten.

Insbesondere muss die Konkurrenz der verschiedenen Bedieneinheiten um die begrenzte Zahl der verfügbaren Schnittstellen geregelt sein (Verdrängung, siehe Kapitel "Verdrängung").

Zu diesem Zweck besitzt jede PCU/HT6 eine Konfigurationsdatei NETNAMES.INI, in der die Konfigurationsparameter abgelegt werden müssen.

2.2.4 Aufbau der Konfigurationsdatei

Die Konfigurationsdatei NETNAMES.INI ist wie folgt aufgebaut:

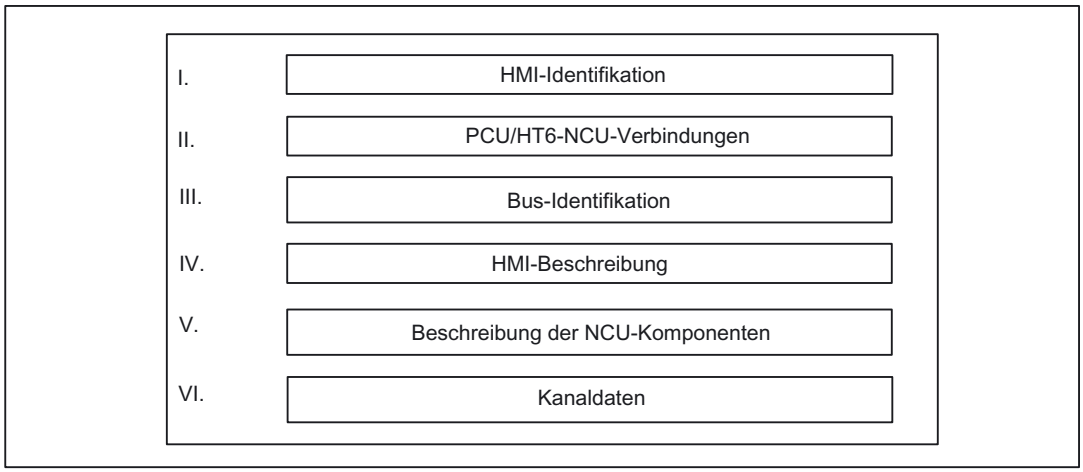


Bild 2-12 Struktur der Konfigurationsdatei NETNAMES.INI

In den folgenden Tabellen sind

- vom Anwender ggf. anzupassende Parameter *kursiv* gesetzt,
- alternativ zu verwendende Parameter mit | getrennt angegeben.

I. HMI-Identifikation

Kennzeichnung des PCU/HT6, für den NETNAMES.INI gilt:

Element	Erläuterung	Beispiel
[own]	Kopf	[own]
owner = *Bezeichner*	HMI-Identifikation	owner = MMC_1

II. PCU/HT6-NCU-Verbindungen

Projektierung der Verbindungen von PCU/HT6 zu den NCUs:

Element	Erläuterung	Beispiel
[conn *Bezeichner*]	Kopf	[conn MMC_1]
conn_i = *NCU_Bez*	Projektierung der NCU-Verbindung(en) i = 1, ..., 15	conn_1 = NCU_1 conn_2 = NCU_2 ... conn_i = NCU_i

III. Bus-Identifikation

Festlegung, an welchem Bus HMI hängt:

Element	Erläuterung	Beispiel
[param network]	Kopf	[param network]
bus = *BTSS* | *MPI*	Bus-Bezeichnung	bus = BTSS

btss: Bedientafelfrontschnittstelle mit 1,5Mbaud

mpi: Multi Point Interface mit 187,5 kBaud

Hinweis

Bei HMI Embedded/HT6 wird die Baudrate automatisch erkannt.

IV. HMI-Beschreibung

Charakterisierung der Bedieneinheit:

Element	Erläuterung	Beispiel
[param *Bezeichner*]	Kopf	[param MMC_1]
mmc_typ = *Typ-/Verbindungskennung*	HMI-Eigenschaften (s.u.)	mmc_typ = 0x40 HMI ist Server und Hauptbedienfeld Erklärungen s. unten
mmc_bustyp = *BTSS* | *MPI*	Bus, an dem HMI hängt	mmc_bustyp = BTSS
mmc_address = *Adresse*	HMI_Adresse	mmc_address = 2
mstt_address bzw. mcp_address = *Adresse*	Adresse der simultan umzuschaltenden MSTT. Wenn nicht vorhanden, gibt es keine umzuschaltende MSTT.	mstt_address = 6 bzw. mcp_address = 6
name = *Bezeichner*	beliebiger, vom Anwender zu vergebender Name (optional, max. 32 Zeichen)	name = HMI_LINKS
start_mode = *ONLINE* | *OFFLINE*	Zustand nach Hochlauf. Falls ONLINE, wird Verbindung über DEFAULT_Kanal-Eintrag zur zugehörigen NCU hergestellt. OFFLINE: Nach Hochlauf wird zunächst keine Verbindung hergestellt. Wichtig: In mmc.ini Sektion [GLOBAL] muss zusätzlich der Eintrag NcddeDefaultMachineName = local gesetzt werden.	start_mode = ONLINE (HMI geht im Hochlauf online an die NCU, auf welcher der Kanal durch Kanaldaten (s. VI)DEFAULT_logChanGrp, DEFAULT_ log_Chan festgelegt ist).

Beim HT6 wird immer die integrierte MSTT simultan umgeschaltet. Für die Adressvergabe des HT6 ist die integrierte MSTT verantwortlich, so dass nur Werte zwischen 1 und 15 erlaubt sind.

Hinweis

Zu beachten ist, dass die über den DEFAULT-Kanal projektierte NCU mit der in **MMC.INI** unter NcddeDefaultMachineName eingetragene NCU übereinstimmen muss.

Erläuterungen zu **mmc_typ**:

mmc_typ enthält Typ- und Verbindungskennungen für Bedieneinheiten und wird beim Umschaltwunsch an die PLC übergeben. mmc_typ wird als Priorität für die Verdrängungsstrategie ausgewertet. Siehe Kapitel "Verdrängung".

Bit 7 = --	(reserviert)
Bit 6 = TRUE:	HMI ist Server (HMI Advanced) und kann nicht verdrängt werden.
Bit 5 = TRUE:	Bedientafel/HT6 ist Hauptbedienfeld.
Bit 4 = TRUE:	Bedientafel/HT6 ist Nebenbedienfeld.

Bedientafel = PCU (incl. HMI Advanced/Embedded) mit OP/TP

Der Anwender kann noch vier zusätzliche HMI-Typen festlegen, die von der Bedieneinheitenumschaltung der PLC ebenfalls gemäß Verdrängungsstrategie berücksichtigt werden:

Bit 3 = TRUE:	OEM_MMC_3
Bit 2 = TRUE:	OEM_MMC_2
Bit 1 = TRUE:	OEM_MMC_1
Bit 0 = TRUE:	OEM_MMC_0

Existiert kein mmc_typ Eintrag in der Datei NETNAMES.INI, so fährt HMI/HT6 gemäß Standardlösung hoch.

V. Beschreibung der NCU-Komponente(n)

Für jede am Bus angeschlossene NCU-Komponente muss ein eigener Eintrag erstellt werden.

Element	Erläuterung	Beispiel
[param *NCU_Bez*]	Kopf	[param NCU_1]
name= *bel_name*	beliebiger, vom Anwender zu vergebender Name, wird in Alarmzeile ausgegeben (optional, max. 32 Zeichen)	name= NCU1
type= *NCU_561*|*NCU_571*|*NCU_572*|*NCU_573*	NCU-Typ	type= NCU_572
nck_address = *j*	Adresse der NCU-Komponente am Bus: *j* = 1, 2, ..., 30 *)	nck_address = 14
plc_address = *p*	Adresse der PLC-Komponente am Bus: *p* = 1, 2, ..., 30 *) (Nur beim MPI-Bus nötig, da bei BTSS-Bus: j = p)	plc_address = 14

*) Bei MPI-Bus: Weil die zugehörige NCU immer die nächst höhere Adresse als die PLC belegt, darf die PLC-Adresse nicht 31 sein. Auf Adresse 31 kann z. B. eine PCU gelegt werden.

Hinweis

Werden die Busteilnehmer-Adressen am MPI-Bus SIMATIC-konform projektiert, kann der Projekteur mit einem SIMATIC-Programmiergerät die vergebenen Adressen auslesen und für die Erstellung der Datei NETNAMES.INI verwenden.

VI. Kanaldaten

Für Bedieneinheitenumschaltung ist es unerlässlich, dass der Bedieneinheit die Zuordnung von Kanälen zu NCUs bekannt ist, um Verbindungen zwischen der Bedieneinheit und NCUs herstellen zu können. (Kanalmenü).

Konzept

Folgende Schritte sind erforderlich:

1. Definition technologischer Kanalgruppen
2. Zuweisung von Kanälen zu den Gruppen
3. Zuweisung der NCUs zu den Kanälen
4. Festlegung der Hochlaufverbindung

NCUs werden auf der Bedieneinheit anhand von Kanalgruppe und Kanal indirekt angefordert. S. Kapitel "Bedienoberfläche".

Literatur:

/IAM/ Inbetriebnahmeanleitung HMI

/FB1/ Funktionshandbuch Grundfunktionen; P3 "PLC-Grundprogramm"

Element	Erläuterung	Beispiel
[chan *Bezeichner*]	Kopf (Kanalmenü der MMC_1)	[chan MMC_1]
DEFAULT_logChanGrp = *Gruppe*	Kanalgruppe des Kanals bei Hochlauf (4.)	DEFAULT_logChanGrp = Fräsen1
DEFAULT_logChan = *Kanal*	angewählter Kanal bei Hochlauf (4.)	DEFAULT_logChan = Kanal11
ShowChanMenu = *TRUE* | *FALSE*	TRUE Kanalmenü anzeigen	ShowChanMenu = TRUE
logChanSetList = *Gruppenliste*	Auflistung der Kanalgruppen (1.)	logChanSet = Fräsen1, Fräsen2
[*Gruppe*]	Kopf (2.)	[Fräsen1]
logChanList = *Kanal1, Kanal2, ...*	Kanäle der Gruppe, getrennt durch Komma (2.)	logChanList = Kanal11, Kanal12, Kanal13
[*Kanal*]	Kopf (3.)	[Kanal11]
logNCName = *Bezeichner*	log. Bezeichner einer NCU (3.)	logNCName = NCU_1

Element	Erläuterung	Beispiel
ChanNum = i (i = 1, 2, 3,...)	Nummer des Kanals in der Projektierung seiner NCU (3.)	ChanNum = 1
Weiter für alle Kanäle der Gruppe		
Weiter nächste Gruppe und ihre Kanäle		

Ein vollständiges Beispiel zur Kanalmenü-Projektierung finden Sie in "Konfigurationsdatei NETNAMES.INI mit Option Bedieneinheitenmanagement".

2.2.5 Konfigurationsdatei erstellen und anwenden

Syntax

Die Konfigurationsdatei muss als ASCII-Datei erzeugt werden. Die Syntax entspricht der von Windows *.ini-Dateien.

Insbesondere gilt:

- Kennworte sind in Kleinbuchstaben zu schreiben.
- Kommentar kann in die Parameterdatei eingefügt werden (nach links von ";" und nach rechts vom Zeilenende begrenzt).
- Blanks sind als Trennzeichen an beliebiger Stelle außer in Bezeichnern und Kennworten erlaubt.

HMI Embedded, OP030, HT6

Die auf PC oder einem PG erstellte Datei NETNAMES.INI wird, wie in

Literatur: FBO/IK/ Projektierung Bedienoberfläche OP 030 / Installation Kit

beschrieben, über die V.24 Schnittstelle geladen und im FLASH-Speicher der Bedieneinheiten dauerhaft hinterlegt.

HMI Advanced

Die Datei NETNAMES.INI kann direkt mit einem Editor (Menü "Inbetriebnahme/HMI/Editor" oder DOS_SHELL) auf der Festplatte der Bedienkomponente bearbeitet werden. Die Datei NETNAMES.INI steht im Installationsverzeichnis C:\USER\.

Beispiel

Eine Beispiel-Konfigurationsdatei finden Sie im Kapitel "Beispiele".

2.2.6 Hochlauf

Vorbesetzungen Standardlösung

Wird keine Projektierdatei NETNAMES.INI in HMI Embedded/OP030/HT6 eingespielt oder kann diese nicht interpretiert werden, so werden folgende Voreinstellungen benutzt (Standard M:N = 1:1):

- Die vorliegende Busart wird automatisch ermittelt.
- HMI hat die Adresse 1.
- OP030 hat die Adresse 10.
- Für Bus BTSS haben NCU und PLC je die Adresse 13.
- Für Bus MPI hat die NCU die Adresse 3 und die PLC die Adresse 2.

Mit Option

Wird jedoch eine spezielle NETNAMES.INI-Datei erstellt, so muss sie wegen der im Folgenden beschriebenen Besonderheiten exakt der Vernetzung entsprechen.

Gelingt es bei einer M:N fähigen Bedieneinheit nicht, im Hochlauf eine Verbindung zur NCU aufzubauen oder liegt ein Projektierfehler vor, so geht die Bedieneinheit in den OFFLINE-Modus über. In diesem Zustand kann der Bediener mit der Recall-Taste zur Bereichsapplikation und anschließend zur Inbetriebnahme umschalten.

Kompatibilität

Die Verwendung der obigen Vorbesetzungen garantiert die Kompatibilität zu allen SW-Ständen für den Betrieb der Bedientafeln.

Hochlauf mit HMI Embedded/HT6

Die Bedieneinheit HMI Embedded/HT6 kann mit der NCU nur dann eine aktive Verbindung aufbauen, wenn die laut Kapitel "Aufbau der Konfigurationsdatei" erstellte Konfiguration in NETNAMES.INI korrekt ist. HMI Embedded, HT6 und OP030 können parallel an einer NCU hochlaufen, da sie als Busteilnehmer verschiedene Adressen haben. OP030 kann als zweite Bedientafelfront genutzt werden, die fest einer NCU zugeordnet ist.

Stimmen die projektierten Adressen mit den realen Adressen (NC-/PLC-Adresse) nicht überein, kann der Inbetriebnehmer über die folgende Tastenfolge den Hochlauf an einer nicht projektierten NCU herbeiführen.

Ablauf

1. HMI/HT6 bootet an der NCU mit der Busadresse 13, falls die Datei NETNAMES.INI nicht verändert wurde (Originalzustand ab Werk).
2. Die Datei NETNAMES.INI wurde verändert, es erscheint die Meldung

   "HMI Embedded version xx.xx.xx: waiting for connection ..."

   – Taste "1" drücken, es erscheint die Meldung:

     "choice: '1'=set new start-address, '^' =boot"

   – Taste "1" drücken, es werden die Busadressen aller am Bus angeschlossenen Teilnehmer angezeigt. Es erscheint die Meldung:

     "Please try one of the shown addresses or press '^' to reboot

     '1',_,_,_,_,'6',_,..._,'D',_,..."

   – Taste "D" und INPUT drücken

   – HMI/HT6 bootet an der NCU mit Busadresse 13 (falls unter der gefundenen Adresse eine NCU ist).
3. Im Bedienbereich Inbetriebnahme/NC/"NC-Adresse" neue NC-Adresse eingeben und mit "Ja" bestätigen.
4. NC-Reset (neue Busadresse wird erst nach NC-Reset wirksam)
5. Verbindung bzw. Kanalmenü in der Datei NETNAMES.INI projektieren und in HMI/HT6 übertragen.
6. Nachdem die NCU-Adressen vergeben wurden, kann der Bus für den M:N-Betrieb verdrahtet werden.

Hinweis

Eine OP030 und eine PCU (HMI Embedded)/HT6 können Sie ohne Parametrierung an einer Schnittstelle betreiben (ab Werk haben sie verschiedene Busadressen).

Hochlauf mit HMI Advanced

Der Hochlauf der HMI Advanced ist auch dann möglich, wenn die Verbindung zur NCU aufgrund fehlerhafter Konfiguration nicht gelingt.

Bei der Vorgabe einer "1:1"-Verbindung im Menü "Inbetriebnahme/HMI/Bedientafel" kann explizit eine NCU-Adresse vorgegeben werden. Nach einem neuen HMI-Hochlauf ist dann die Kommunikation zwischen HMI und NCU/PLC in Ordnung.

Ablauf

1. HMI bootet an der NCU mit der Busadresse 13, falls die Datei NETNAMES.INI nicht verändert wurde (Originalzustand ab Werk).
2. Busadresse der NCU wurde verändert, es erscheint der Alarm

   "120201 name: Kommunikation ausgefallen"

   – Im Bedienbereich Inbetriebnahme/HMI/Bedientafel die Verbindung auf 1:1 setzen und in die NC-Adresse "13" eintragen

   – Mit OK bestätigen und HMI booten
3. Punkt 6. wie bei HMI Embedded

Hinweis

Überprüfen Sie im Fehlerfalle die aktiven Busteilnehmer im Menü:
- Inbetriebnahme/NC/NCK-Adressen (HMI Embedded, HT6 und HMI Advanced)
- Inbetriebnahme/HMI/Bedientafel (HMI Advanced)

Hochlauf mit HMI Advanced Option

Ist die Option Bedieneinheitenumschaltung vorhanden, so kann mit folgenden Schritten ein Konfigurationsproblem behoben werden:

1. Wählen Sie mit der Inputtaste das Kanalmenü an
2. Mit Recall gelangen Sie in die Bereichsumschaltung
3. Wählen Sie Inbetriebnahme.

Benötigte Dokumentation

Literatur:

/BH/ Bedienkomponenten-Handbuch

/IAD/ Inbetriebnahmeanleitung

/FB1/ Funktionshandbuch Grundfunktionen; P3, PLC-Grundprogramm

Hier finden Sie beschrieben:

- Busaufbau MPI/BTSS, Busadressen, /IAD/
- Busabschluss, /IAD/, /FB/S7
- Anschluss der MSTTs über PLC-Grundprogramm, /FB/, P3
- DIP-FIX Einstellungen der MSTT, /IAD/

Hinweis

Nach Durchführung einer Serieninbetriebnahme muss ein Power On an der PCU durchgeführt werden, damit sich die Busteilnehmer (PLC, NC, PCU) wieder synchronisieren können.

2.2.7 HMI-Umschaltung

Das M:N-Konzept erlaubt es, während der Laufzeit Eigenschaften und Zustände der Bedieneinheiten innerhalb der in NETNAMES.INI festgelegten Konfiguration zu ändern.

Z. B. kann der Anwender eingreifen, um

- Bedieneinheiten umzuschalten (siehe Kapitel "An- und Abschaltbedingungen"),
- MSTTs umzuschalten (siehe Kapitel "MSTT-Umschaltung").

An einer NCU dürfen höchstens zwei Bedieneinheiten gleichzeitig online sein. Zur Vermeidung von Konfliktsituationen, gibt es eine Verdrängungsstrategie (siehe Kapitel "Verdrängung").

Die HMI-Eigenschaften werden in der NETNAMES.INI für jede Bedieneinheit projektiert. Wenn eine Bedieneinheit über das Umschaltprotokoll an eine NCU online gehen möchte, werden seine Parameter an die PLC der betreffenden NCU übergeben. Das PLC-Programm **Bedieneinheitenumschaltung** wertet die Parameter aus:

- Überprüfung der Verdrängungsbedingungen
- ggf. Umschaltung

2.2.8 Verdrängung

An einer NCU können maximal zwei Bedieneinheiten online sein. Wenn das der Fall ist, und eine weitere PCU/HT6 möchte online gehen, muss sichergestellt sein, dass Konflikte vermieden werden. Erreicht wird das durch den im Folgenden beschriebenen Verdrängungsalgorithmus.

Ablauf

- Die PLC stellt eine Offline-Anforderung an die zu verdrängende Bedieneinheit.
- Diese gibt der PLC eine positive oder negative Quittung:
  - Im positiven Fall wird die Bedieneinheit verdrängt (s.u.), beendet die Kommunikation mit der NCU und geht in die Betriebsart Offline.

    Eine evtl. der PCU zugeordnete MSTT wird von der PLC deaktiviert.

    Beim HT6 ist die integrierte MSTT immer zugeordnet und wird somit ebenfalls deaktiviert.
  - Eine negative Quittung wird gegeben, wenn auf der Bedieneinheit Prozesse ablaufen, die nicht unterbrochen werden dürfen, z. B. Betrieb über V24 oder Datenübertragung zwischen NCU und PCU. In diesem Falle wird die Bedieneinheit nicht verdrängt und bleibt online.

Verdrängungs-Strategie

Das PLC-Programm Bedieneinheitenumschaltung handelt anhand der

- Prioritäten der Bedieneinheiten und
- der aktiven Prozesse

Die Priorität hängt vom Parameter **mmc_typ** in der Projektierungsdatei NETNAMES.INI ab (siehe Kapitel "Aufbau der Konfigurationsdatei"). Das PLC-Programm Bedieneinheitenumschaltung wertet diese gemäß folgender Tabelle aus:

HMI-Eigenschaft	Priorität
Server	6
Hauptbedienfeld	5
Nebenbedienfeld	4
OEM-MMC 3	3
OEM-MMC 2	2
OEM-MMC 1	1
OEM-MMC 0	0

Verdrängungs-Regeln

Für die Bedieneinheiten-Verdrängung gelten folgende Regeln:

- Große Priorität verdrängt kleinere.
- Bei geicher Priorität der Teilnehmer wird die aktive Bedieneinheit verdrängt.

Einschränkend gilt:

- Server können nicht verdrängt werden, da sie zu jeder NCU permanent Verbindung halten müssen.
- Bedieneinheiten, auf denen die folgenden Prozesse aktiv sind, können nicht verdrängt werden:
  - Datentransfer z. B. von/zu NCU
  - Bedieneinheit befindet sich in der Umschaltphase auf diese NCU
  - Bedieneinheit wechselt gerade seinen Bedienmodus
  - OEM sperrt Umschaltung

2.2.9 An- und Umschaltbedingungen

Um eine bisher offline arbeitende Bedieneinheit an einer bestimmten NCU online gehen zu lassen oder eine bereits online arbeitenden Bedieneinheit auf eine andere NCU umzuschalten, gehen Sie folgendermaßen vor:

PCU

1. Rufen Sie an der PCU durch Drücken der Kanalumschalttaste das Kanalmenü auf.
2. Wählen Sie über einen horizontalen Softkey die Kanalgruppe aus.
3. Betätigen Sie den gewünschten vertikalen Softkey des Kanals. Siehe "Realisierung der Bedieneinheitenumschaltung".

HT6

1. "Panel Function" durch Drücken der gleichnamigen Taste auslösem.
2. Drücken Sie den Softkey "Kanal".
3. Wählen Sie die Kanalgruppe aus.
4. Wählen Sie den Kanal aus.

   Falls der gewünschte Kanal nicht in dieser Gruppe enthalten ist, so können Sie durch Drücken der Taste "Recall" zu Punkt 2 gelangen.

Dann wird PCU/HT6 an- bzw. umgeschaltet, wenn dies nicht von einer der folgenden Bedingungen (in der Meldezeile abzulesen) verhindert wird.

Meldungen bei PCU-Umschaltung	
HMI Embedded	Meldungstext
109001	keine Umschaltung: Umschaltsperre in aktueller PLC gesetzt
109002	keine Umschaltung: Ziel-PLC belegt, erneut versuchen
109003	keine Umschaltung: Umschaltsperre in Ziel-PLC gesetzt
109004	keine Umschaltung: PLC von höherprioren PCUs belegt
109005	keine Umschaltung: an Ziel-PLC keine PCU verdrängbar
109006	keine Umschaltung: gewählter Kanal ungültig
109007	Kanalumschaltung läuft
109009	Umschaltung: innerer Zustand fehlerhaft
109010	Verdrängung: innerer Zustand fehlerhaft
109012	Bedieneinheitenumschaltung, PLC Timeout: 002
109013	Aktivierung abgelehnt

Hinweis

Bei HMI Advanced werden die entsprechenden Meldungen ohne Meldungsnummer ausgegeben.

Darüberhinaus können in HMI Embedded/HT6 und HMI Advanced weitere Meldungen auftreten, die Hinweise auf den aktuellen Zustand oder Fehler bei Projektierung oder im Ablauf hinweisen.

Näheres siehe

Literatur: /DA/ Diagnosehandbuch, Kap. 1

2.2.10 Realisierung der Bedieneinheitenumschaltung

Die Bedieneinheitenumschaltung ist eine Erweiterung der Kanalumschaltung.

Kanalumschaltung

Die Kanalprojektierung bietet die Möglichkeit, Kanäle beliebiger NCUs individuell zu gruppieren und zu benennen. Im Rahmen der Kanalumschaltung ist eine HMI-Umschaltung auf eine andere NCU möglich.

Als Basis für die Kanalprojektierung dient die Datei NETNAMES.INI. S. "Aufbau der Konfigurationsdatei".

2.2.11 Bedienoberfläche

Funktion

Über die Bedienoberfläche können Sie in jedem Bedienbereich zwischen der Bedieneinheit und einer der angeschlossenen NCU/PLC-Einheiten eine Verbindung herstellen.

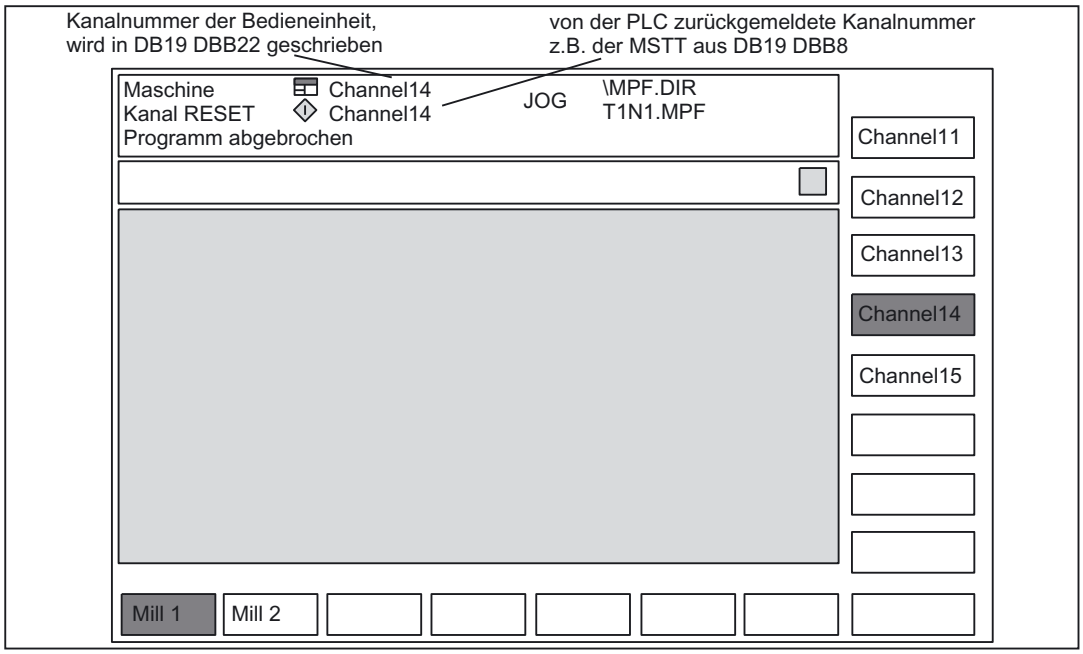


Bild 2-13 Kanalmenü (Die Kommentare beziehen sich auf die 1. HMI-Schnittstelle)

Es werden nur die Kanäle der jeweiligen Gruppe angezeigt.

Betätigen Sie die Kanalumschalttaste. Die aktuell bestehende Verbindung wird durch die markierten Softkeys (horizontal, vertikal) angezeigt, wenn das Kanalmenü aktiv ist.

Kanal umschalten

Über die vertikal projektierten Softkeys kann auf andere Kanäle umgeschaltet werden.

Gruppe umschalten

Über die horizontal projektierten Softkeys schalten Sie auf eine andere Gruppe (siehe Kapitel "Realisierung der Bedieneinheitenumschaltung") um, auf den vertikalen Softkeys werden nun die Kanäle der aktuellen Gruppe angezeigt. Eine Umschaltung auf einen anderen Kanal (und ggf. damit auf eine andere NC) findet erst mit Betätigung eines vertikalen Softkeys statt.

NC umschalten

Über die vertikal projektierten Softkeys kann auf eine andere NC umgeschaltet werden, wenn der Kanal nicht auf der aktuellen NC liegt.

Vorgehensweise: ggf. einen Kanalbereich NCs (Horiz. Softkeys 1-8) projektieren und aus jeder NCU einen Kanal auf vertikale Softkeys legen.

Hinweis

Es werden nur die Verbindungen über Softkey angeboten, die auch tatsächlich belegt sind und deren Kanäle in der jeweiligen NC aktiv sind.

HT6

Beim HT6 ist das Kanalmenü zweistufig aufgebaut. In der 1. Stufe wählen Sie die Kanalgruppe und in der 2.Stufe wählen Sie den Kanal. Einzelheiten entnehmen Sie bitte der Bedienungsanleitung des HT6.

2.2.12 Bedienmodus-Umschaltung

An einer NCU können gleichzeitig zwei PCUs/HT6 online sein. Um zu vermeiden, dass beide gleichzeitig schreibend auf dieselben Daten bzw. Dateien zugreifen, gibt es zwei Bedienmodi:

- den aktiven und
- den passiven Bedienmodus.

Nur einer von zwei PCUs/HT6 kann aktiv sein, der andere ist passiv.

Das Zusammenspiel läuft nach folgenden Regeln ab:

Aktiver Bedienmodus

- Der aktive Bedienmodus wird vom Bediener per Tastendruck an der Bedientafelfront angefordert.

  Er besitzt folgende Merkmale:

  – Alle Bedienungen und Bedienbereiche sind aktiviert.

  – Bediener kann bedienen und beobachten.

  – Die der Bedieneinheit zugeordnete MSTT ist aktiviert.

  – Wenn noch Datenaustauschprozesse (z. B. Serieninbetriebnahme, verschiedene Dienste der Werkzeugverwaltung, Inbetriebnahme Antriebskonfiguration) zwischen der anderen Bedieneinheit und der gemeinsamen NCU laufen, kann die PCU/HT6 nicht unmittelbar aktiv werden.

Passiver Bedienmodus

- Der passive Bedienmodus wird dann wirksam, wenn die andere PCU/HT6 den Aktiv-Modus angefordert hat.

  Er besitzt folgende Merkmale:

  – Die Verbindung zur NCU bleibt weiterhin bestehen.

  – Alle Bedienungen sind deaktiviert.

  – Bediener kann nicht bedienen: Am Bildschirm ist ein Bildrahmen mit Header und Alarmzeile und einem Hinweis auf "passiv" zu sehen.

  – Das globale Menü ist aktivi.

  – Evtl. vorher (im aktiven Modus) angestoßene Dienste bleiben aktiv (z. B. Betrieb über V24, Nachladen von Teileprogrammen, Abarbeiten der Jobliste, Alarme)

  – Die der Bedieneinheit zugeordnete MSTT ist deaktiviert.

  – Das Applikationsfenster und die Softkeys sind deaktiviert.

Der Wechsel zum aktiven Bedienmodus kann auf 2 Arten erfolgen:

- Input-Taste
- Kanalumschalttaste und Kanalauswahl

Regeln für die Bedienmodus-Umschaltung

Für den Wechsel der Bedienmodi gelten folgende Regeln (siehe auch Kapitel "Verdrängung", "Verdrängungs-Strategie"):

- Eine PCU/HT6, die an einer NCU online geht, bekommt an dieser den aktiven Bedienmodus.

  Wenn vorher bereits eine andere PCU/HT6 an dieser NCU aktiv war, geht diese in den Passivmodus, wenn die PLC dies zulässt.

- Wenn zwei PCUs/HT6s online sind, erfolgt der Wechsel der Bedienmodi über Tastendruck ("Input", ENTER, RETURN), mit dem der aktive Bedienmodus angefordert wird.

- Der Wechsel vom aktiven in den passiven Bedienmodus kann von der PCU/HT6 abgelehnt werden, wenn die aktuelle HMI-Anwendung nicht abbrechbar oder fertig ist. Andererseits bekommt eine PCU/HT6 den aktiven Bedienmodus nicht, falls sich die an der NCU gerade aktive PCU/HT6 nicht passiv schalten lässt.

- Wenn eine Online-Anforderung von einer PCU/HT6 kommt
  - und bisher keine PCU/HT6 online ist:

    Die anfordernde PCU/HT6 geht online und schaltet sich in den Aktivmodus.

    Eine evtl. zugeordnete MSTT wird von der PLC aktiviert.
  - und eine PCU/HT6 bereits online ist:

    Diese PCU/HT6 schaltet sich in den Passivmodus und wird verdrängt.

    Die anfordernde PCU/HT6 geht online.

- Wenn zwei PCUs an einer NCU online sind und die bisher aktive PCU/HT6 geht offline, dann wechselt diese zuerst in den Modus passiv. Dann geht die zweite PCU/HT6 in den Modus aktiv und die erste bricht die Verbindung mit der NCU ab.

Hinweis

Der HMI-Typ wird als Priorität für die Verdrängungsstrategie ausgewertet. Siehe Kapitel "Verdrängung".

Kann die aktive PCU/HT6 nicht in den passiven Bedienmodus versetzt werden, so bekommt die anfordernde PCU/HT6 den passiven Bedienmodus.

2.2.13 MSTT-Umschaltung

Eine MSTT kann nicht unabhängig von der PCU umgeschaltet werden, der sie zugeordnet ist. Die Umschaltung erfolgt nur, wenn

- die PCU umschaltet und
- die MSTT-Adresse im HMI-Parameterblock der NETNAMES.INI vorhanden ist (siehe Kapitel "Aufbau der Konfigurationsdatei").
- in der Bedieneinheitenumschaltung der PLC MSTT_enable gesetzt ist.

MSTT aktivieren/deaktivieren

Falls der PCU in der NETNAMES.INI eine MSTT zugeordnet ist, wird sie im Rahmen der Bedienmodi-Wechsel aktiviert bzw. deaktiviert. Die MSTT-Umschaltung in der PLC wird vom Bedienmodus-Wechsel als Unterfunktion aufgerufen.

PCU wechselt Bedienmodus	MSTT wird
aktiv -> passiv	deaktiviert
passiv -> aktiv	aktiviert

2.2.14 Das PLC-Programm "Bedieneinheitenumschaltung"

Einführung

Im Gesamtkonzept M:N nimmt die Bedieneinheitenumschaltung eine wichtige steuernde Funktion wahr:

- Die Bedieneinheitenumschaltung auf PLC prüft die Prioritäten der Anforderungen und die Zustände der beteiligten Komponenten und schaltet gegebenenfalls um.
- Die NCU setzt Signale und wertet Signale aus, die im Zusammenhang der Bedieneinheitenumschaltung benötigt werden.

Die Bedieneinheitenumschaltung ist ein SW-Paket in der Toolbox. Sie steht mit ihrer **Standardfunktion** unmittelbar zur Verfügung, kann jedoch auch für besondere Einsatzfälle nach individuellen Bedürfnissen modifiziert werden.

Es sind zwei Klassen von Modifikationen vorgesehen:

1. Einfache Parametrierung der Standardfunktion
2. Weitergehende Umgestaltung der Bedieneinheitenumschaltung

Gründe für eine weitergehende (anwenderspezifische) Umgestaltung können sein:

- von der Standardlösung abweichendes Verdrängungsverhalten
- von der Standardlösung abweichende Bedienmodusumschaltung
- eigenständige Behandlung des Override-Schalters im Umschaltfall
- Existenz einer 2. Maschinensteuertafel an einer PCU.

Programmstruktur

Bedieneinheitenumschaltung besteht aus:

1. FB101/DB101: Betriebsartenumschaltung online/offline
2. FB102/DB102: Bedienmodusumschaltung aktiv/passiv
3. FC103: Maschinensteuertafel-Umschaltung

Jeder Programmteil ist in einem eigenen Funktionsbaustein (FB) oder in einer Funktion (FC) realisiert. Die Variablen sind für jeden FB in einem eigenen Instanz-Datenbaustein (DB) abgelegt. In **FB101** befindet sich das Bedieneinheitenumschaltungs-**Hauptprogramm.** FB101 muss in einem Organisationsbaustein (OB) aufgerufen werden, um die Funktionen zu aktivieren. FB102 und FC103 werden in FB101 mehrfach aufgerufen.

MSTT-Umschaltung

Die MSTT-Umschaltung ist nicht zwingend. Sie kann über die FB101/DB101 Variable **MSTT_enable** ein- oder ausgeschaltet werden. Zuweisung in der Deklarationstabelle:
TRUE: MSTT-Umschaltung ist wirksam
FALSE: MSTT-Umschaltung ist unwirksam, kein FC103 Aufruf

Es wird immer die MSTT aktiviert, welche der aktiven PCU zugeordnet ist. Wenn die Bedienmodusumschaltung ausgeschaltet ist, dann wird die MSTT aktiviert, welche der zuletzt online gegangenen PCU zugeordnet ist.

Hochlaufbedingung:

Um zu verhindern, dass bei einem Neustart der NCU, die davor zuletzt angewählte MSTT aktiviert wird, sollte beim Aufruf von FB1 in OB100 der Eingangsparameter **MCP1BusAdr** auf den Wert 255 (Adresse 1. MSTT) und **MCP1Stop** auf TRUE (1. MSTT ausschalten) gesetzt werden.

Freigaben:

Wenn von einer MSTT auf eine andere umgeschaltet wird, dann bleiben eventuell eingeschaltete Vorschub- sowie Achsfreigaben erhalten.

Hinweis

Nach einer MSTT-Umschaltung ist der Override-Schalter der neuen MSTT sofort wirksam.

Die zur Zeit der Umschaltung betätigten Tasten wirken weiter. Ist an der PCU, auf die umgeschaltet wird, keine MSTT vorhanden, kann die Wirkung der Tasten von da aus nicht mehr aufgehoben werden. Maßnahmen für diese Fälle sind im PLC-Anwenderprogramm zu realisieren.

Bedienmodus-Umschaltung

Die Bedienmodus-Umschaltung ist nicht zwingend. Sie kann über die FB101/DB101 Variable **aktiv_enable** ein- oder ausgeschaltet werden. Zuweisung in der Deklarationstabelle:
TRUE: Bedienmodus-Umschaltung ist wirksam
FALSE: Bedienmodus-Umschaltung ist unwirksam, kein FB102 Aufruf

Fehlermeldungen

Kommt es während des Programmablaufes zu Störungen (z. B. ausbleibendes Nahtstellensignal), so werden entsprechende Alarme/Fehlermeldung an den Datenbaustein DB2 übergeben. Es gibt 6 **Alarme**:

1. Fehler HMI-Busadresse, HMI-Bustyp
2. Keine Bestätigung HMI 1 offline
3. HMI 1 geht nicht offline
4. Keine Bestätigung HMI 2 offline
5. HMI 2 geht nicht offline
6. Anklopf-HMI geht nicht online (anfordernde HMI)

und eine **Fehlermeldung**: Fehler Lebenszeichenüberwachung

Bei unveränderten Standardeinstellungen in FB101 beginnen die Alarme bei

DB2.DBX188.0 (1. Alarm) und enden bei

DB2.DBX188.(6. Alarm)

Mit der Variablen:

DBX_Byte_alarm lässt sich der Bytewert für die 6 Alarme anders als Standardeinstellung 188 einstellen. Mit der Variablen:

DBX_Byte_report lässt sich der Bytewert der Betriebsmeldung anders als Standardeinstellung 192 einstellen.

Mischbetrieb

Definition:

Mischbetrieb liegt vor, wenn an der ersten HMI-Schnittstelle der NCU ein herkömmliches OP ohne Bedieneinheitenumschaltung angeschlossen ist. Die Bedieneinheitenumschaltung arbeitet dann ausschließlich an der 2. HMI-Schnittstelle.

Mit Hilfe des Parameters "MMC_mixed_mode" (Variable in FB101/DB101) kann die Betriebsart vom reinen HMI-Betrieb auf den Mischbetrieb umgeschaltet werden, indem folgende Einstellungen vorgenommen werden:

Bei Aufruf von FB101 muss zugewiesen werden:
TRUE Mischbetrieb ist wirksam. Die Bedieneinheitenumschaltung wirkt auf der 2. Schnittstelle
FALSE Mischbetrieb ist unwirksam. Die Bedieneinheitenumschaltung wirkt auf beideSchnittstellen

Randbedingungen:

- Für die erste HMI-Schnittstelle darf keine Maschinensteuertafel (MSTT) projektiert werden. Die erste HMI-Schnittstelle erhält bei Mischbetrieb immer den **aktiven** Modus.
- Damit die zweite HMI-Schnittstelle auch aktiven Bedienmodus erhalten kann, ist es bei Mischbetrieb möglich, dass beide Online-Schnittstellen aktiven Bedienmodus erhalten. Es sind jedoch Randbedingungen einzuhalten.

⚠ WARNUNG
Bei gleichzeitiger Eingabe von beiden Bedieneinheiten besteht die Gefahr, dass inkonsistente Daten in die Steuerung eingegeben werden.

Serverbetrieb

Wenn ein Server eine HMI-Schnittstelle einer NCU belegt hat, kann er nicht mehr verdrängt werden.

Bedienerlaubnis für Server bearbeiten

Im Bedieneinheitenumschaltprogramm gibt es drei für Server relevante Programmzweige zur Behandlung von HMI-Anforderungen:

1. Anforderung für das Abgeben des Bedienfocus
2. Anforderung für das Setzen des Bedienfocus
3. Bedienfocus abgeben

Jeder Zweig prüft/bearbeitet die erste und zweite HMI-Schnittstelle.

Die Anforderungen werden positiv quittiert, wenn keine Umschaltsperren bestehen. Das "Bedienfocus abgeben" schließt die Deaktivierung der zugehörigen Maschinensteuertafel ein.

Siehe auch Bilder im Kapitel "Beispiele/Graphiken der Funktionssequenzen".

Wartezeiten für Quittungssignale

Im Programm Bedieneinheitenumschaltung sind zwei Wartezeiten durch mehrfaches Lesen der Systemzeit über SFC64 realisiert, um das Programm unabhängig von Timern zu halten. Im Bedarfsfall können die Wartezeiten auf Quittungssignale verändert werden durch:

FB101/DB101 Variablen:

waiting_period_1: Warten auf Aktivierung/Online HMI

waiting_period_2: Warten auf Passivierung/Offline HMI

Warten auf HMI-Lebenszeichen:

Den FB101-Variablen können Werte zugewiesen werden zwischen:

0-32 (Sekunden). Die Eingabe erfolgt in ms.

Programmintegration

Soll das PLC-Programm Bedieneinheitenumschaltung als Funktion in einem übergeordneten PLC-Programm aufgerufen werden, so ist darauf zu achten, dass FB101, FB102 und FC103 sowie die zugehörigen Instanzbausteine DB101, DB102 nicht schon anderweitig verwendet werden.

Initialisierung

Bei einem Neustart der NCU werden alle für Bedieneinheitenumschaltung relevanten Signale der PLC-Nahtstelle in DB19 auf Null gesetzt.

Hinweis

Bevor an einer NCU eine Initialisierung vorgenommen wird, muss darauf geachtet werden, dass an dieser NCU **keine** PCU/HT6 im Modus **Online** angeschaltet ist.

Gegebenenfalls muss ein HMI-Restart ausgeführt werden.

Rücksetzen der Nahtstelle durch PLC

Die für die Bedieneinheitenumschaltung relevanten Signale der Nahtstelle können wie folgt gezielt zurückgesetzt werden (ohne RESET an der NCU):

FB101/DB101 Variable:

initialization

TRUE: Einmaliges Rücksetzen der Signale in DB19. Nach Ausführung stellt sich der Parameter Initialisierung selbständig wieder auf FALSE.

Lebenszeichenüberwachung

Sobald eine PCU/HT6 online geschaltet wird, sendet sie ein Lebenszeichen in DB10 DBB108. Sendet eine PCU/HT6 im Onlinebetrieb länger als die in waiting_period_2 definierte Zeit kein Lebenszeichen, so wird vom PLC-Programm die Meldung: "Fehler Lebenszeichenüberwachung" erzeugt. Die Meldung wird erst wieder gelöscht, wenn irgendeine PCU/HT6 wieder von Offline- in den Online-Betrieb wechselt.

Kennung für PCU/HT6 "Bedieneinheitenumschaltung existiert"

PCUs/HT6s müssen in bestimmten Betriebszuständen erkennen können, ob die Bedieneinheitenumschaltung existiert. Diesen Zweck erfüllt das Anklopf-Schnittstellensignal in DB19.DBW110 **m_to_n_alive**. Sobald der Baustein FB101 in der PLC aufgerufen wird, sendet er auch ein Lebenszeichen. Dieses Lebenszeichen besteht aus der zyklischen Erhöhung von **m_to_n_alive** (Umlaufzähler).

Generierung nach Anpassungen

Nachdem statische Parameter in FB101 verändert wurden, muss:

- DB101 gelöscht,
- neu generiert,
- aufgerufen und
- gespeichert werden.

Verwendete Bausteine und Funktionen

Funktionsbausteine	**FB101, FB102**
Instanzdatenbausteine	DB101, DB102
Funktionen	FC103
DB der Schnittstelle	DB19
Globaler Datenbaustein für Fehlermeldungen	DB2, (DB3)
Timer-Hilfsfunktion	SFC64

2.3 Mehrere Bedientafelfronten und NCUs Standardlösung

Das M:N-Konzept **ohne** die Option **Bedieneinheitenmanagement** ist im folgenden beschrieben.

Hinweis

Dieses Kapitel trifft auf das HT6 nicht zu, da ohne Bedieneinheitenmanagement immer nur ein HT6 an einer NCU betrieben werden kann.

2.3.1 Konfigurationen

Konfigurationsparameter

Die Kombinierbarkeit der Komponenten macht es erforderlich, dem System mitzuteilen, welche Komponenten auf welche Weise miteinander verbunden sind. Dies geschieht bei HMI Advanced durch einen Bediendialog im Bereich Inbetriebnahme. Bei HMI Embedded/OP030 erfolgt die Eingabe der Konfigurationsparameter durch die Erstellung einer Konfigurationsdatei, die zur Inbetriebnahme geladen wird. Sie muss, wie im Folgenden beschrieben, aufgebaut werden.

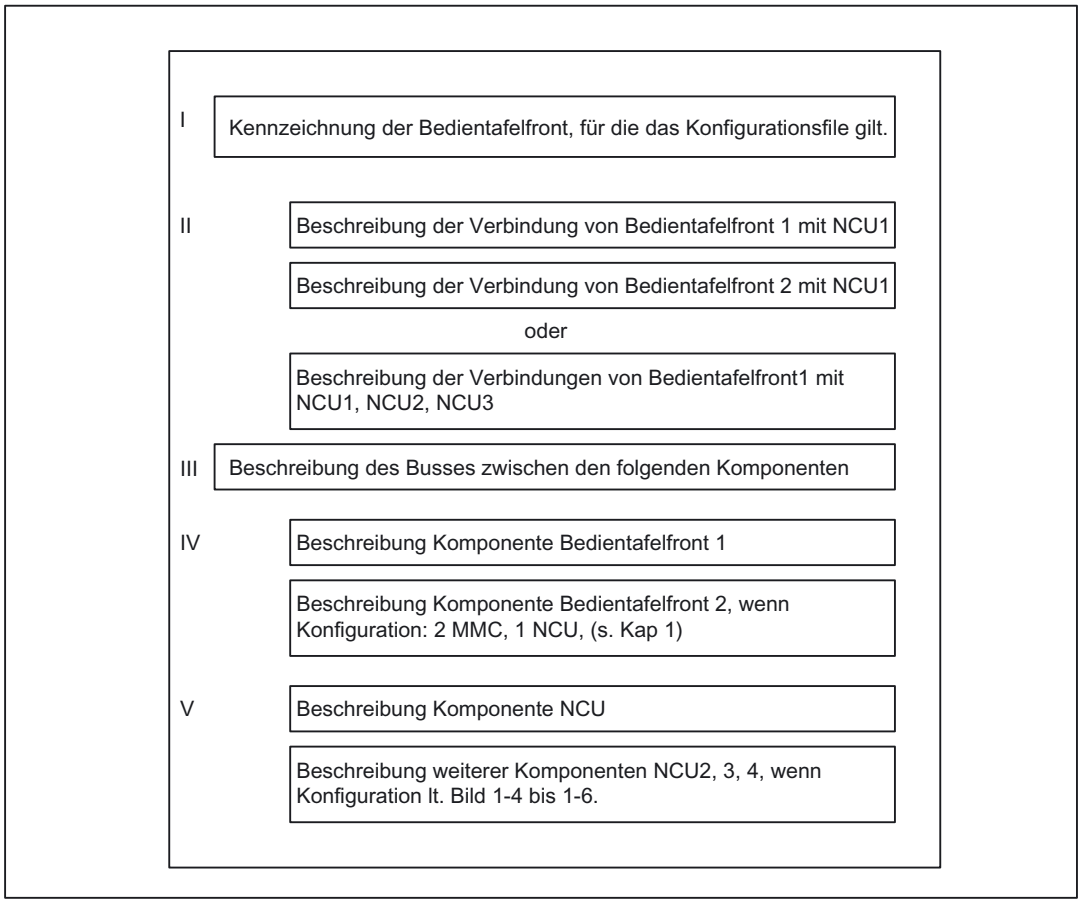


Bild 2-14 Struktur des Konfigurationsfiles NETNAMES.INI

Beispiele

Vollständige Beispiele für Konfigurationsdateien finden Sie im Kapitel "Beispiele" dieser Beschreibung.

Syntaktische Vereinbarungen

Die Konfigurationsdatei muss als ASCII-Datei erzeugt werden. Die Syntax entspricht der von Windows *.ini"-Files.

In den folgenden Tabellen sind vom Anwender ggf. anzupassende Anteile *kursiv* gesetzt. Alternativ zu verwendende Parameter sind mit | getrennt angegeben. Parameter sind in Kleinbuchstaben zu schreiben. Eingeleitet von ";" und nach hinten durch Zeilenende begrenzt können Kommentare eingefügt werden. Blanks sind als Trennzeichen an beliebiger Stelle außer Bezeichnern und Parametern erlaubt.

Anzahl der Konfigurationsdateien

Für jede angeschlossene Bedientafel ist ein Konfigurationsfile erforderlich.

Die Konfigurationsdateien verschiedener Bedientafeln einer Konfiguration unterscheiden sich nur im ersten Eintrag, der die Zuordnung der Datei zu einer bestimmten Bedientafel enthält ([own] siehe unten). Der Kern der Datei wird zweckmäßigerweise einmal erstellt und für die zweite Bedientafel eine Kopie erstellt. In der Kopie wird dann die Kennzeichnung der Bedientafel, für die die Datei gilt, hinzugefügt.

I. Kennzeichnung der Bedientafel

Kennzeichnung der Bedientafel, für die die Konfigurations gilt.

Tabelle 2- 1 Kennzeichnung der Bedientafelfront

Beschreibungsteil	formal	Beispiel
Kopf	[own]	[own]
Folgezeile	owner = *Bezeichner*	owner = MMC_2

Bezeichner: Mit dem gewählten Bezeichner muss ein Beschreibungsteil einer Bedientafel laut IV. erzeugt werden.

Schlüsselworte:
own: Einleitung Erkennungsteil
owner: Eigner

II. Verbindungen

Beschreibung der Verbindungen von den Bedientafelkomponenten zu der anzusprechenden NCU. Je Bedientafel ist ein Eintrag der folgenden Art erforderlich.

Tabelle 2- 2 Beschreibung der Verbindungen

Beschreibungsteil	formal	Beispiel
Kopf	[conn *Bezeichner*]	[conn MMC_1]
Folgezeile(n)	conn_i = *NCU_Bez*	conn_1 = NCU_1

Bezeichner: Mit dem gewählten Bezeichner muss ein Beschreibungsteil einer Bedientafelfront laut IV. erzeugt werden.

NCU_Bez: Mit dem gewählten NCU-Bezeichner muss ein Beschreibungsteil der NCU laut V. erzeugt werden.

Schlüsselworte:
conn: Einleitung Verbindungsteil
conn_i: Kennwort für Verbindung, i = 1, 2, ..., 8.

III. Beschreibung des Busses

Die Hardware gestattet die Verbindungen über verschiedene Busse zu realisieren, die sich im Wesentlichen in der Baudrate unterscheiden. Die benutzte Busart muss angegeben werden.

Tabelle 2- 3 Beschreibung des Busses

Beschreibungsteil	formal	Beispiel
Kopf	[param network]	[param network]
Folgezeile	bus = btss | mpi	bus = btss

Schlüsselworte:
param network Einleitung des Netzbeschreibungsteils
bus: Bus
btss: Bedientafelfrontschnittstelle mit 1,5 Mbaud
mpi: Multi Point Interface mit 187,5 KBaud

Hinweis

Bei HMI Embedded wird die Baudrate automatisch erkannt.

IV. Beschreibung der Bedienkomponente(n)

Je am Bus angeschlossene Bedientafel muss ein eigener Eintrag erstellt werden. In SW 3.x maximal zwei Einträge.

Tabelle 2- 4 Beschreibung der Bedienkomponente

Beschreibungsteil	formal	Beispiel
Kopf	[param *Bezeichner*]	[param MMC_1]
Folgezeilen (optional)	name= *bel_name*	name= MMC_A
(optional)	type= *mmc_100* | *mmc_102* | *op_030*	type = mmc_100
	mmc_address = j	mmc_address = 1

Bezeichner: Eintrag für erste oder zweite Bedientafel

bel_name: beliebiger Name von max. 32 Zeichen

mmc_100 | *mmc_102* | *op_030*: Type der Bedienkomponente

j: Adresse der Bedienkomponente am Bus: *j* = 1, 2, ... 31

Schlüsselworte:
param: Einleitung Parameter für eine Bedienkomponente
name: beliebiger Name für die zu beschreibende Bedienkomponente
type: Typ der Bedienkomponente
mmc_address: Busadresse der Bedienkomponente

V. Beschreibung der NCU-Komponente(n)

Für jede am Bus angeschlossene NCU-Komponente muss ein eigener Eintrag erstellt werden.

Tabelle 2- 5 Beschreibung der NCU-Komponente

Beschreibungsteil	formal	Beispiel
Kopf	[param *NCU_Bez*]	[param NCU_1]
Folgezeilen (optional)	name= *bel_name*	name= NCU1
(optional)	type= *ncu_561* | *ncu_571* | *ncu_572* | *ncu_573*	type= ncu_572
*)	nck_address = *j*	nck_address = 13
*)	plc_address = *p*	plc_address = 13

NCU_Bez: Eintrag für NCU-Komponente.

bel_name: beliebiger Name von max. 32 Zeichen; bei HMI Advanced wird der hier eingetragene Name (z. B. NCU1) in der Alarmzeile mit ausgegeben

ncu_561|*ncu_571*|*ncu_572*|*ncu_573:* NCU-Typ, (ncu_561 nicht bei Konfiguration 1 PCU, 3 NCU)

j: Adresse der NCU-Komponente am Bus: *j* = 1, 2, ... 31 *)

p: Adresse der PLC-Komponente am Bus: *p* = 1, 2, ... 31 *)

Bei bus = btss sind *j* und *p* gleich zu besetzen.

***)** Bei bus = mpi gilt: Weil die zugehörige NCU immer die nächst höhere Adresse als die PLC belegt, darf die PLC-Adresse nicht 31 sein. Auf Adresse 31 kann z. B. eine PCU gelegt werden.

Schlüsselworte:
param: Einleitung Parameter für eine NCU-Komponente
name: beliebiger Name für die zu beschreibende NCU
type: Typ der NCU
nck_address: Busadresse der NCU
plc_address: Busadresse der PLC.

Hinweis

Werden die Busteilnehmer-Adressen am MPI-Bus SIMATIC-konform projektiert, kann der Projekteur mit einem SIMATIC-Programmiergerät die vergebenen Adressen auslesen und für die Erstellung der Datei NETNAMES.INI verwenden.

Vorbesetzungen

Wird keine Konfigurationsdatei NETNAMES.INI in HMI Embedded/OP030 eingespielt oder kann diese nicht interpretiert werden, so gelten folgende Voreinstellungen:

- Die vorliegende Busart wird automatisch ermittelt.
- PCU hat die Adresse 1.
- OP030 hat die Adresse 10.
- Für Bus BTSS haben NCU und PLC je die Adresse 13.
- Für Bus MPI hat die NCU die Adresse 3 und die PLC die Adresse 2.

Entspricht die tatsächliche Vernetzung diesen Vorbesetzungen, so kann auf die explizite Erstellung und das Laden der NETNAMES.INI verzichtet werden. Wird jedoch eine spezielle Datei erstellt, so muss es exakt der Vernetzung entsprechen.

2.3.2 Umschalten der Verbindung auf eine andere NCU

Hinweis

Die Funktion Kanalmenü ist eine Option und muss in der Datei "NETNAMES.INI" projektiert werden.

Sie können in allen Bedienbereichen mit der Kanalumschalttaste in das Kanalmenü umschalten. Dabei wechseln nur die horizontalen und vertikalen Softkeys.

Mit den horizontalen Softkeys wählen Sie eine Kanalgruppe an (max. 24), in einer Kanalgruppe können 8 Verbindungen zu Kanälen verschiedener NCUs aufgenommen werden.

Im Bild "Kanalmenü" werden alle aktuellen Kommunikationsverbindungen und die zugehörigen Symbolnamen angezeigt.

2.3.3 Konfigurationsdatei erstellen und anwenden

HMI Embedded, OP030

Die auf PC oder einem Programmiergerät erstellte ASCII-Datei NETNAMES.INI wird über die V.24 Schnittstelle geladen und im FLASH-Speicher der Bedieneinheiten dauerhaft hinterlegt.

Literatur: FBO/IK/, Installation Kit

HMI Advanced

Die Datei NETNAMES.INI kann direkt mit einem Editor (Menü "Inbetriebnahme/HMI/Editor" oder DOS_SHELL) auf der Festplatte der Bedienkomponente bearbeitet werden. Die Datei NETNAMES.INI steht im Installationsverzeichnis:

C:\USER\NETNAMES.INI.

2.3.4 Hochlauf

Unterschiede HMI Embedded und HMI Advanced

Durch das unterschiedliche Verhalten beim Hochlauf unterscheidet sich das Vorgehen bei der Inbetriebnahme.

- HMI Embedded läuft immer im Mode "M:N", wenn in der Datei NETNAMES.INI "M:N" projektiert ist.
- Beim HMI Advanced ist der Mode im Menü "Inbetriebnahme/HMI/Bedientafel" einstellbar. Der HMI Advanced läuft standardmäßig in einer "1 : 1"-Kopplung zu einer NCU, dabei ist es möglich die Adresse der NCU direkt vorzugeben. Wird der Mode "M:N" eingestellt, so sieht HMI Advanced in der Datei NETNAMES.INI nach, welche Partner hier vorgegeben werden. Die Adressen sind frei belegbar.

  Empfehlung:

  Adresse 0 freihalten (für PG)

  Adresse 13 freihalten (für Service-Fall: NCU-Tausch)
- Die OP030 ist funktionell nicht "M:N" fähig. Sie kann als zweite Bedientafelfront genutzt werden, die fest einer NCU zugeordnet ist ("1 : 1"-Kopplung). Hierzu sind die Adressen der angeschlossenen Partner einstellbar.

Hinweis

Es wird empfohlen, die Vorgehensweise vorher schriftlich (Adressierungen etc.) festzulegen.

Inbetriebnahme

Ab Werk werden die NCUs mit der Busadresse 13 ausgeliefert. Jede NCU im Bus muss mit einer eigenen Busadresse versehen werden.

Adressen werden vergeben in:

- HMI: Datei NETNAMES.INI
- NCK: Im Menü "Inbetriebnahme/NC/NCK-Adresse"
- MSTT: Schalter... (Adresse und evt. Baud-Rate , siehe auch /IBN/)
  OB100 Parameter: ...(siehe auch FB1/P3).

Hinweis

Eine Adresse des NCK wird durch "SRAM löschen" (Schalter S3= Stellung "1" auf NCU) nicht gelöscht.

Hochlauf mit HMI Embedded/HMI Advanced

Siehe Kapitel "Mehrere Bedientafelfronten und NCUs mit Option Bedieneinheitenmanagement/Hochlauf".

2.3.5 NCU-Tausch

Beim Tausch bzw. Hinzufügen einer NCU wird wie bei der Inbetriebnahme (siehe "Hochlauf") verfahren.

Variante 1

1. 1:1-Verbindung zwischen PCU und NCU herstellen
2. HMI an NCU mit Busadresse "13" hochlaufen lassen (siehe oben)
3. Über Bedienbereich Inbetriebnahme/NC/NC-Adresse neue NC-Adresse eingeben und NCU booten
4. Bus wieder für M:N-Betrieb verdrahten

Variante 2

1. Die NCU, die für einer am Bus beteiligten PCU die "Hochlauf-NCU" ist, wird abgeschaltet. (HMI läuft an der ersten in der Datei NETNAMES.INI projektierten Verbindung hoch)
2. HMI an der NCU mit Busadresse 13 hochlaufen lassen (siehe oben)
3. Über Bedienbereich Inbetriebnahme/NC/NC-Adresse neue NC-Adresse eingeben und NCU booten
4. "Hochlauf-NCU" wieder einschalten

Hinweis

Bitte beachten Sie Folgendes:

- Die Busadresse 13 sollte für den Service-Fall freigehalten werden (nicht von einem Busteilnehmer belegt sein).
- HMI Embedded:

  Die Namenslänge in der Datei NETNAMES.INI (Projektierung Kanalmenü) ist auf 5 Zeichen begrenzt.
- HMI Advanced:

  Das Datum "mstt_address" wird nicht ausgewertet, es dient zur Dokumentation der Busteilnehmer. Falls die Kanäle auf verschiedenen NCUs liegen, muss im Bedienbereich Inbetriebnahme/HMI/Bedientafel "m:n" eingestellt werden.

Datenaustausch zwischen NC<->PLC

Bei Konfigurationen aus 1 x PCU und n x NCU ist es oft notwendig, dass Synchronisationen zwischen den NCUs stattfinden.

Dafür stehen folgende Möglichkeiten zur Verfügung:

- NCK-Peripherie am Antriebsbus (digital, analog, schreiben von NC und PLC).
- PLC-Peripherie (E/A-Kopplung).
- Kopplung über PROFIBUS-DP.
- Kopplung über die Funktion Globale Daten der SIMATIC S7.

2.4 Einschränkungen beim Umschalten von Bedienkomponenten

Abgelehnte Verbindung

Beim Umschalten kann es vorkommen, dass die NCU, mit der Verbindung aufgenommen werden soll, dies ablehnt. Es könnte eine Störung der NCU vorliegen oder es kann keine weitere Bedientafel mehr angenommen werden. HMI Embedded schaltet in diesem Fall nach ca. fünf Sekunden automatisch auf die Verbindung 1 um. HMI Advanced zeigt für die Variablen "#" an.

Alarme, Meldungen

Das Verhalten ist vom HMI-Typ abhängig:

1. HMI Embedded/OP 030

   Es können sowohl aufgrund von Betriebsmitteleinschränkungen auf Treiberebene, als auch durch begrenzten Arbeitsspeicher gleichzeitig nur Alarme/Meldungen jeweils einer NCU verarbeitet werden.

2. HMI Advanced

   Es wird nur ein Alarmtextfile verwaltet. Als NCU-Kennung wird vor jedem Alarm bzw. jeder Meldung, der in der Datei NETNAMES.INI vergebene NCU-Name angezeigt. Um NCU-spezifische Anwendertexte zu bekommen kann man für bestimmte NCU's in der PLC Anwenderbereiche definieren. Die Alarme/Meldungen aller verbundenen Komponenten können gleichzeitig verarbeitet und angezeigt werden.

Bedienoberfläche

Das Verhalten ist vom HMI-Typ abhängig:

1. HMI Embedded

   Es können in einem Fenster gleichzeitig nur Felder und Variablen einer NCU angezeigt werden. Alarme und Meldungen werden nur von der NCU angezeigt, die gerade mit der PCU verbunden ist.

   Über Anwenderprojektierung (OEM) können bis zu 4 Verbindungen (eine aktive Verbindung, drei weitere Verbindungen) gleichzeitig angezeigt werden, wobei alle Variablen (Alarme und Meldungen) einer Verbindung in einem Fenster enthalten sein müssen (fensterspezifische Verbindungen).

2. HMI Advanced

   Es können in einem Fenster Felder und Variablen unterschiedlicher NCUs angezeigt werden (als OEM-Anwendung). Alarme und Meldungen können von allen NCUs angezeigt werden, zu denen die PCU eine Verbindung hat.

3. OP030

   OP030 kann nur als "1 : 1"-Verbindung zu einer NCU projektiert werden.

Bei standardmäßiger Nutzung der HMI Embedded und HMI Advanced bezüglich Konfiguration (Kapitel "Projektierbarkeit") ist keine Projektierung der Bedienoberfläche nötig. Wenn in einem Bild gleichzeitig Variablen verschiedener NCUs angezeigt werden sollen, ist Projektierung erforderlich.

Literatur:
/BEM/ SINUMERIK HMI Embedded / EBF Projektierungspaket

2.5 NCU-Link

2.5.1 Einführung

Wegen der Begrenzung der Betriebsmittel Speicher und Rechenleistung ist die Zahl der Kanäle bzw. Achsen pro NCU beschränkt. Bei komplexen und verteilten Maschinen, wie z. B. Mehrspindler- und Rundtaktmaschinen, reicht eine NCU nicht mehr aus, um die Anforderungen zu erfüllen. Deshalb werden die Steuerung bzw. die Achsregelungen über mehrere NCUs verteilt.

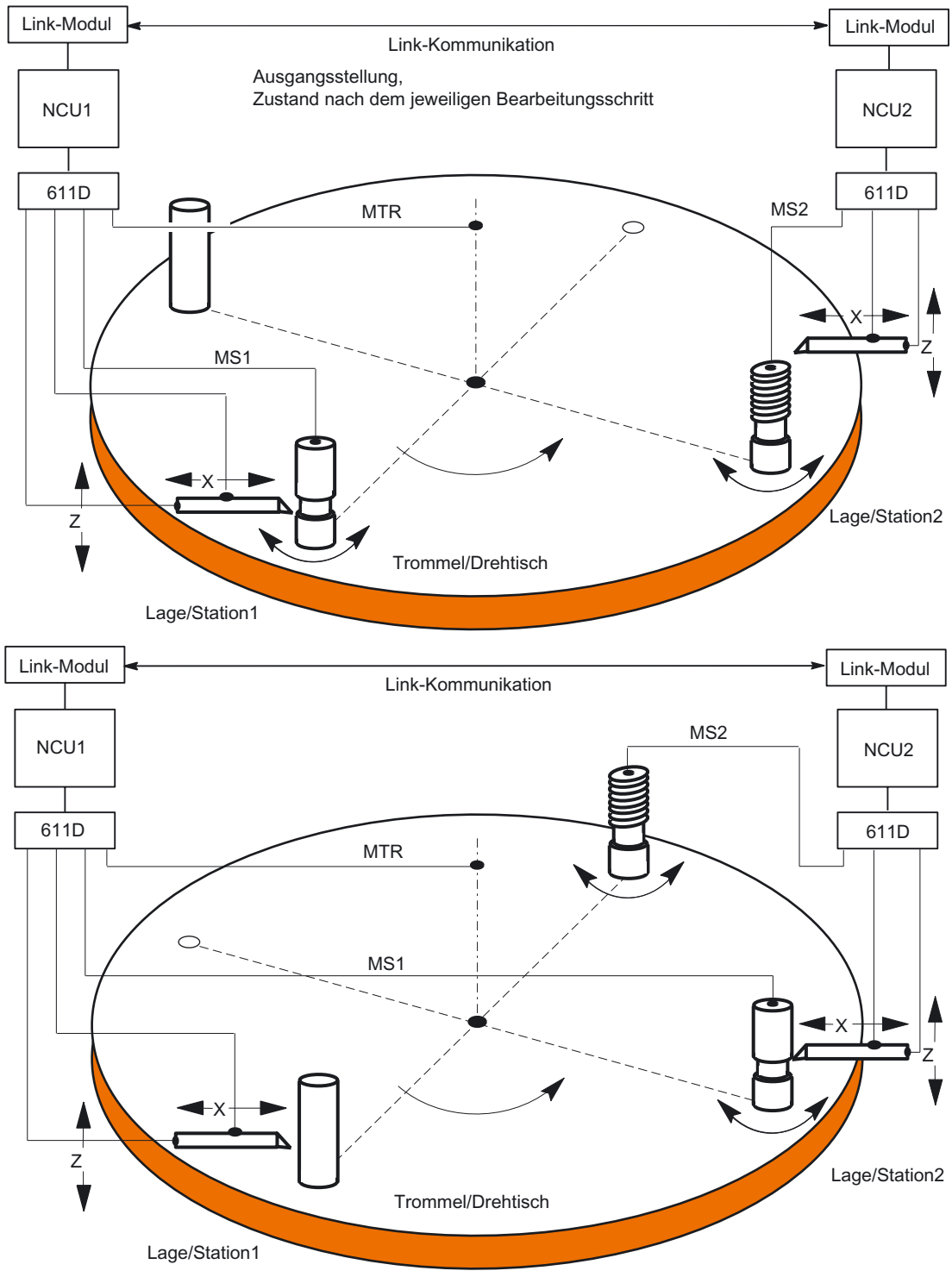
Damit aber die Stationen, Kanäle und Achsen nach wie vor übergreifende bzw. zusammenhängende Dinge verrichten können, wird die NCU-Link Funktionalität vom System bereitgestellt.

Funktionserweiterungen

Folgende Anwendungen sind möglich:

- NCU-übergreifende Interpolation (Soll-, Istwert und VDI-Signalkopplung)
- Echtes Tauschen von Achsen
- NCU-übergreifender Zugriff auf Achswerte und axiale Systemvariable
- NCU-Anwender-Kommunikation mit Hilfe von NCU-Link-Variablen
- Erzeugung von Alarmen auf der NCU, die von einer Unregelmäßigkeit betroffen ist, auch wenn die Ursache auf einer anderen NCU liegt.

2.5.2 Technologische Beschreibung



Drehung der Trommel/des Drehtischs (Maschinenachse MTR) um eine Position, Zustand vor der jeweiligen Bearbeitung

Bild 2-15 Trommelumschaltung, schematisch, Ausschnitt

Das Bild stellt die wichtigsten Teile einer einfachen Mehrspindleranlage dar. Auf der Trommel sind mechanisch mehrere Spindeln angebracht. Jede Spindel kann für eine unterschiedliche Bearbeitung verwendet werden. Zusammen mit dem Schlitten (X- und Z-Achse) bildet sie eine Bearbeitungsstation. Diese ist einem Kanal zugeordnet. Ein Werkstück wird durch eine Spindel gedreht.

Das zu bearbeitende Werkstück wird nur einmal beladen bzw. entladen. Das Werkzeug befindet sich auf dem Schlitten (z. B. X-, Z-Achse). Für jeden Bearbeitungsvorgang können unterschiedliche Werkzeuge eingesetzt werden.

Das Werkzeug bleibt der Bearbeitungsstation zugeordnet. Die Spindeln mit Werkstück werden von einer Bearbeitungsstation zur nächsten bewegt.

Die Spindel kann nur für einem Bearbeitungsprozess zur Zeit kontrolliert werden. Der Kanal muss zu einem Zeitpunkt die Achsen des Schlttens und die aktuelle Spindel adressieren können. Nach jedem Weiterschalten jedoch muss eine andere Maschinenachse als Spindel vom Kanal angesprochen werden. Das Konzept "**Achscontainer**" löst die veränderliche Abbildung von Kanalachsen auf Maschinenachsen. Die Maschinenachsen können einer mit NCU-Link verbundenen anderen NCU angehören. Eine ansprechbare Maschinenachse einer anderen NCU wird als **Link-Achse** bezeichnet (siehe "Link-Achsen").

Die folgenden Punkte stehen in enger Beziehung zu NCU-Link und werden in eigenen Unterkapiteln behandelt.

- Link-Achsen
- Achscontainer
- NCU-übergreifende Anwenderkommunikation
- Konfiguration des Link-Verbundes

2.6 Link-Achsen

Einführung

In diesem Unterkapitel wird beschrieben, wie eine Achse (z. B. B1 gemäß Bild "Überblick Link-Achsen"), die physikalisch an der Antriebssteuerung der NCU2 angeschlossen ist, außer von NCU2 auch von NCU1 aus angesprochen werden kann.

Voraussetzungen

- Die beteiligten NCUs, NCU1 und NCU2, müssen über das Link-Modul miteinander verbunden sein.

Literatur: /PHD/ Handbuch Projektierung NCU 840D, Link-Modul

- Die Achse muss durch Maschinendaten entsprechend konfiguriert werden.
- Die Option Link-Achse muss vorhanden sein.
- Die Link-Kommunikation muss mit dem Maschinendatum:

  MD18780 $MN_MM_NCU_LINK_MASK

  aktiviert sein. Der Link-Verbund muss gemäß "Konfiguration von Link-Achsen" konfiguriert sein.

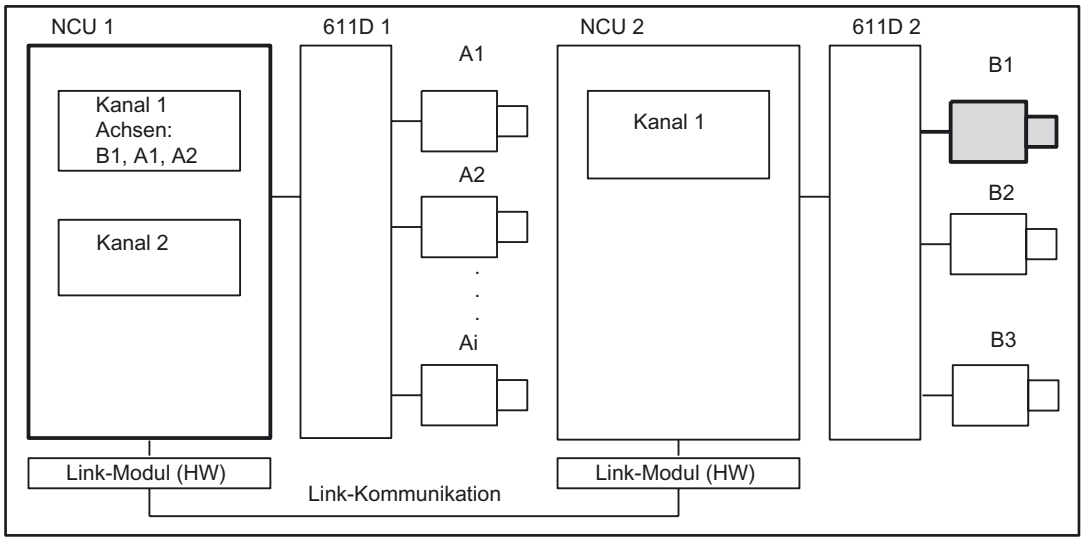


Bild 2-16 Überblick Link-Achsen

Begriffe

Folgende Begriffe sind für das Verständnis des weiteren Textes wichtig:

- **Link-Achse**

  Link-Achsen sind Maschinenachsen, die an einer anderen NCU physikalisch angeschlossen sind und deren Lageregelung unterliegen. Link-Achsen können dynamisch Kanälen einer **anderen** NCU zugeordnet werden. Sie sind aus Sicht einer bestimmten NCU nicht → lokale Achsen. Der dynamischen Änderung der Zuordnung zu einem Kanal (Sonderfall: Kanal auf einer anderen NCU) dient das in "Achscontainer" beschriebene Konzept der **Achscontainer**.

  Achstausch mit `GET` und `RELEASE` aus dem Teileprogramm ist für Link-Achsen nur innerhalb einer NCU verfügbar. Um die NCU-Grenze zu überschreiten, muss zuerst die Achse mit dem Achscontainer in die NCU bzw. in einen Kanal gebracht werden, um sie dann wie jede andere Achse beliebig hin- und hertauschen zu können.

- **Lokale Achse**

  Eine lokale Achse wird ausschließlich von der NCU angesprochen, an deren Antriebsbus sie angeschlossen ist.

- **Link-Kommunikation**

  Die Link-Kommunikation wird durch Link-Module an den beteiligten NCUs realisiert. Die Link-Kommunikation umfasst Sollwerte, Istwerte, Alarmbehandlung, globale Variablen (Daten) und Signale (Achssignale, PLC-Signale).

- **Heimat-NCU**

  Die NCU, welche die Antriebsbusverbindung für eine → Link-Achse herstellt und welche die Lageregelung durchführt, wird als Heimat-NCU der Link-Achse bezeichnet.

  Im Bild "Überblick Link-Achsen" ist NCU 2 Heimat-NCU für → Link-Achse B1.

- **Interpolation**

  Mit der Option **Link-Achse** ist für NCUs mit → Link-Kommunikation das Interpolieren zwischen → lokalen Achsen und Achsen an weiteren NCUs möglich.

  Bei nicht rein lokaler Interpolation erfolgt innerhalb eines Interpolationstaktes der zyklische Datenaustausch (Sollwerte, Istwerte, ...). Dieses wirkt sich insbesondere beim Warten auf externe Ereignisse als Totzeit aus.

- **Achs-Wechsel**

  Die Benutzung einer → Link-Achse durch eine bestimmte NCU kann sich dynamisch ändern. Hierfür ist der Mechanismus **Achscontainer** gemäß "Achscontainer" vorgesehen. Für Link-Achsen steht der Teileprogrammbefehl `GET` gar nicht zur Verfügung, der Teileprogrammbefehl `GETD` nur innerhalb einer NCU.

  Bis SW 4 war es lediglich möglich, Achsen zwischen verschiedenen Kanälen einer NCU zu tauschen.

- **Konfiguration von Link-Achsen**

  NCUs, die → Link-Achsen benutzen wollen, müssen die **NCU-Bezeichner** für die Heimat-NCU der Link-Achsezusätzlich zu den üblichen Kanal- und Achs-Maschinendaten konfigurieren.

- **Heimat-Kanal**

  Kanal, in dem für die Achse nach dem Einschalten der Anlage die sollwerterzeugende Teileprogrammbearbeitung erfolgt.

- **Lead-Link-Achse**

  Aus Sicht einer NCU (2), die Folgeachsen verfährt, eine *Leitachse*, die von einer anderen NCU (1) verfahren wird. Die erforderlichen Daten der Leitwertachse werden über →Link-Kommunikation für NCU (2) zur Verfügung gestellt. Zwischen Leitachse und Folgeachse(n) besteht Achskopplung z. B. über Kurventabelle.

2.6.1 Konfiguration von Link-Achsen und Containerachsen

Funktion

Über die Funktion Link ist es möglich die Anzahl der Achsen und Kanäle an der Anlage zu erhöhen. Der notwendige Linktakt steigt mit der Anzahl der Link-Achsen und Containerachsen an.

Typische Linktakte sind:

- 2 NCUs mit 2 Containern mit je 6 Achsen bzw. Spindeln: 8ms
- 3 NCUs mit 2 Containern mit je 6 Achsen bzw. Spindeln: 17ms
- 4 NCUs mit 2 Containern mit je 4 Achsen bzw. Spindeln: 17ms

Hinweis

Die Zeiten erhöhen sich, wenn zusätzlich noch mehrere Linkvariable übertragen werden müssen.

Maschinenachsabbild

Die Kanäle arbeiten mit einem aus 31 logischen Achsen bestehenden logischen Maschinenachsabbild. Dieses Abbild zeigt auf:

- lokale Achsen
- Link-Achsen
- Containerslots

Containerslots wiederum zeigen auf:

- lokale Achsen oder
- Link-Achsen.

Das folgende Bild verdeutlicht die Zusammenhänge:

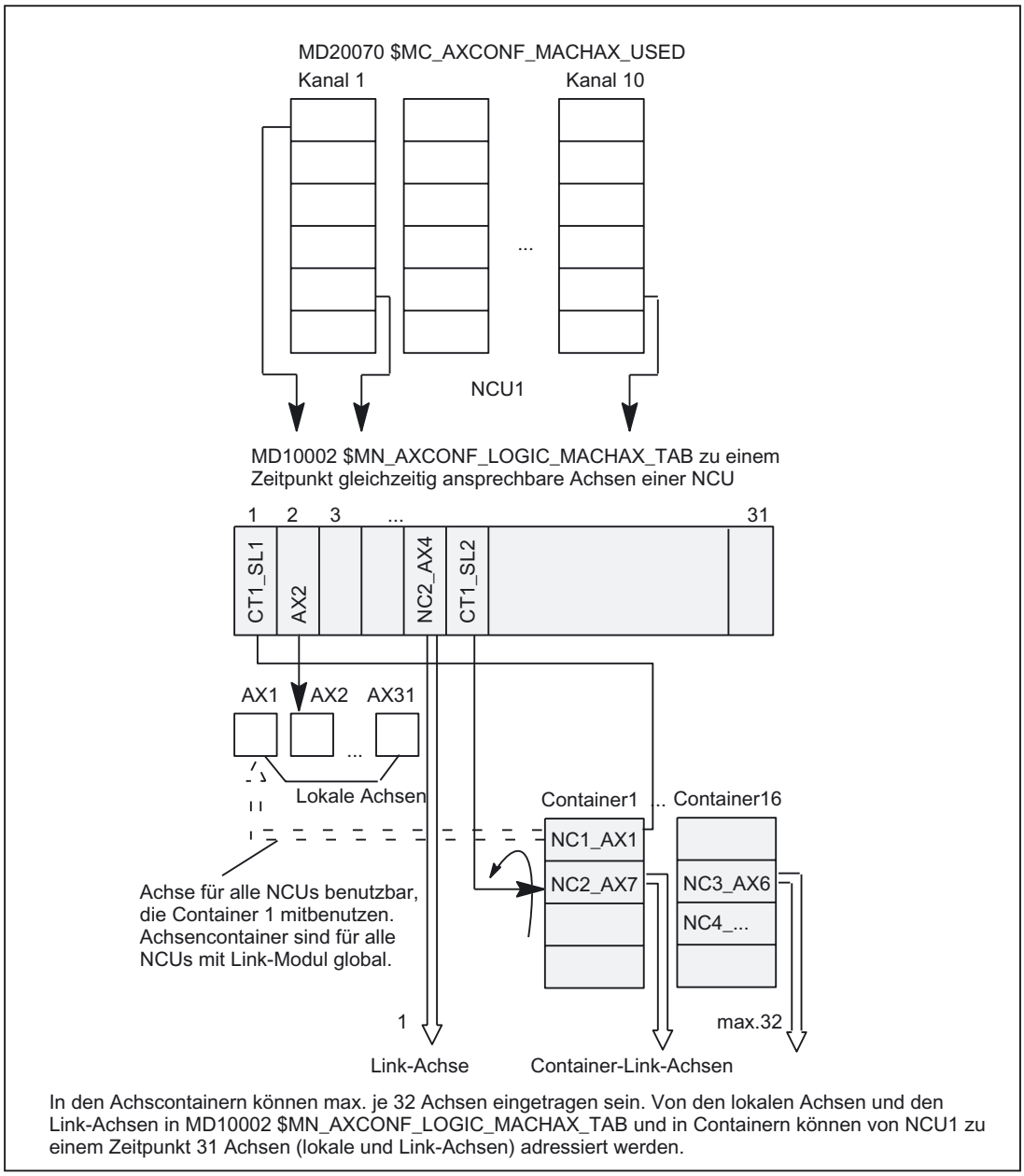


Bild 2-17 Konfiguration von Link-Achsen

Unterscheidung lokale/Link-Achsen

Wenn im Gesamtsystem Link-Achsen angesprochen werden sollen, muss die Konfiguration Angaben zu den NCUs der Achsen enthalten. Es wird unterschieden zwischen lokalen Achsen und Link-Achsen. Zur Unterscheidung dient die durch das Maschinendatum anzulegende Tabelle:

MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB

.

Hinweis

Die Funktionen der **Achscontainer** sind im Unterkapitel "Achscontainer" beschrieben.

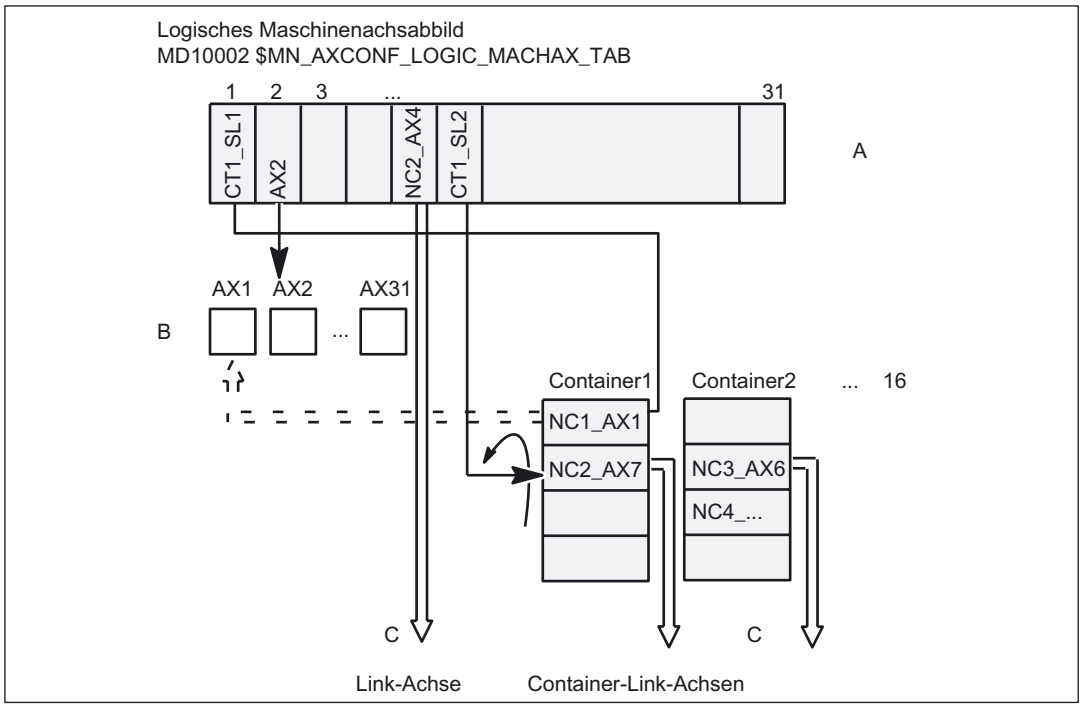


Bild 2-18 Zuordnung der Kanalachsen zu lokalen Maschinenachsen und Link-Achsen

Erklärung

Das **logische Maschinenachsabbild** A adressiert lokale Maschinenachsen B und Link-Achsen C.

Die Anzahl der lokalen Maschinenachsen in B ist begrenzt. Die zulässige höchste Anzahl für ein bestimmtes System finden Sie im Katalog NC60.1.

In B und C zusammen stehen alle Achsen, welche die NCU ansprechen kann.

Die **Einträge in A** haben folgende Form:

$MN_AXCONF_LOGIC_MACHAX_TAB[n] = NCj_AXimit

n: Index in der Tabelle A

NC: steht für **NCU** mit

j: **NCU-Nummer**, 1 <= j <= 16

i: Achs-Nummer, 1 <= i <= 31

Kanalachsen werden mit dem folgenden Maschinendatum auf das **logische Maschinenachsabbild A** zugewiesen:

MD20070 $MC_AXCONF_MACHAX_USED

Vom Teileprogramm aus betrachtet sind nur die Maschinenachsen zugänglich, die vom Kanal über das logische Maschinenachsabbild im betrachteten Zeitpunkt (ggf. über Achscontainer, s. u.) adressiert sind.

Vorbesetzung

Die Einträge des logischen Maschinenachsabbildes A sind so vorbesetzt, dass auf Eintrag 1 der lokale Achsname AX1, auf Eintrag 2 der lokale Achsname AX2, ... steht.

Beispiele

Im logischen Maschinenachsabbild können zum Beispiel folgende Ausdrücke stehen:

NC2_AX7: Maschinenachse 7 der NCU 2

AX2: lokale Maschinenachse 2

Werden ausschließlich Ausdrücke der letzten Form AXi in das logische Maschinenachsenabbild eingetragen, so entspricht das einer Konfiguration, bei der nur lokale Achsen angesprochen werden.

Achtung: Die Vorbesetzungen lauten:

MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[0] = AX1

MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[1] = AX2

...

Hinweis

Eine weitere gültige Form für Einträge im logischen Maschinenachsabbild A ist:

MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[n] = CTx_SLy mit

CT: steht für **Container**

x: Container-Nummer, 1 <= x <= 16

SL: steht für **Slot**

y: Slot-Nummer, 1 <= y <= 32

Achscontainer stellen eine Zusammenfassung von Achsen in Gruppen dar, die dynamisch verändert werden können. Achscontainer sind beschrieben im Kapitel "Achscontainer".

2.6.2 Achsdaten und Signale

Einführung

Achsdaten und Signale für eine Link-Achse fallen an der Heimat-NCU der Link-Achse an. Die NCU, welche die Bewegung einer Link-Achse veranlasst hat, erhält Achsdaten und Signale vom System zur Verfügung gestellt:

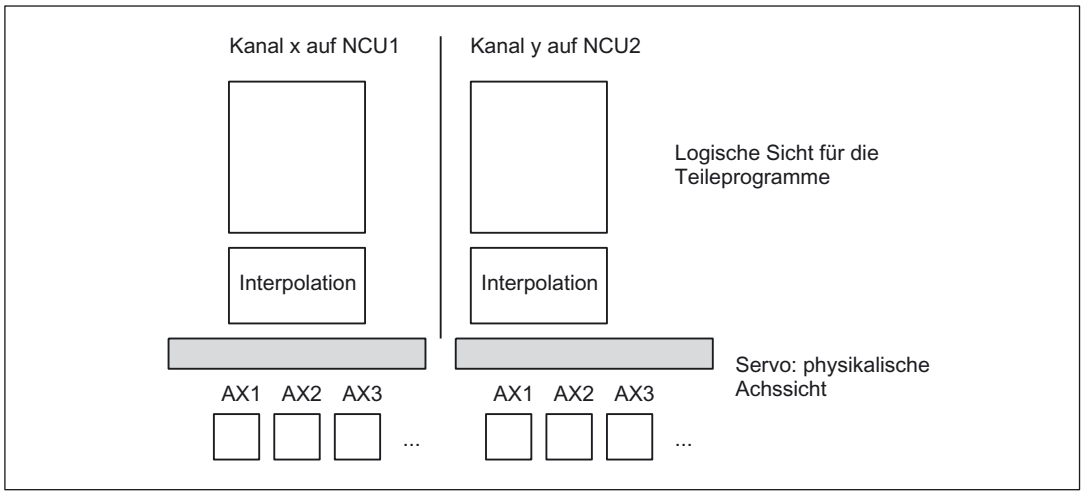


Bild 2-19 Sichtweisen auf Achsen

Implizit aktive Link-Kommunikation

Während der Interpolation werden für Achsen, die physikalisch einem nicht lokalen Servo unterstellt sind (erkenntlich an den Einträgen in dem Maschinendatum:

MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB bzw. Achscontainer),

die Daten über die Link-Kommunikation so bereitgestellt, wie es für lokale Achsen in der logischen Sicht der Teileprogramme erfolgt. Der Vorgang bleibt für die Anwendungen verdeckt.

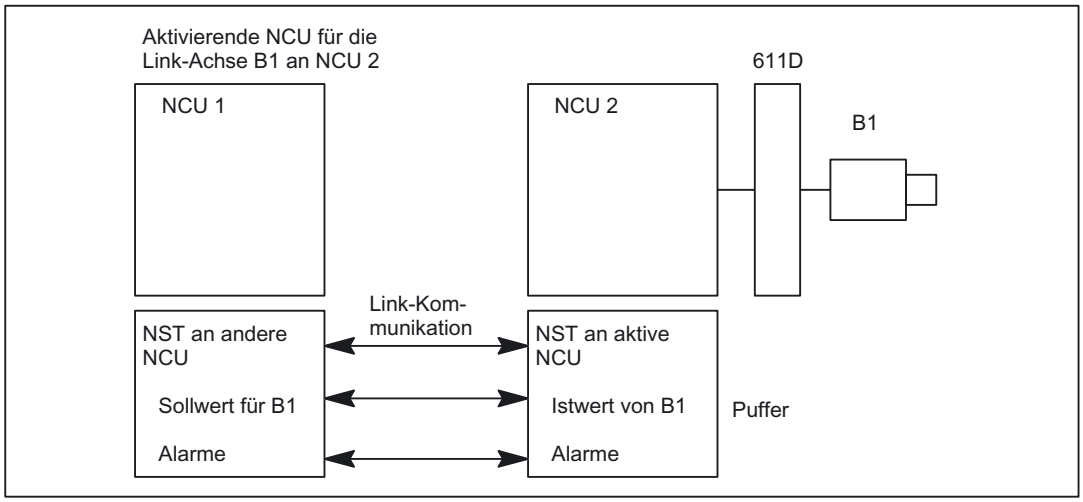


Bild 2-20 Austausch von Betriebsdaten und Signalen einer Link-Achse

Lageregelung

Die Lagereglung erfolgt auf der NCU, auf der die Achse physikalisch mit dem Antrieb verbunden ist. Dort befindet sich auch die zugehörige Achs-Nahtstelle. Die Lagesollwerte werden bei Link-Achsen auf der aktiven NCU erzeugt und über NCU-Link übermittelt.

Kommunikationsarten

Die Link-Kommunikation umfasst zwei Kommunikationsarten:

- Zyklische Kommunikation
- Nicht zyklische Kommunikation

Zyklische Kommunikation

- Sollwerte für die Link-Achse,
- Istwerte von der Link-Achse
- Statussignale der Link-Achse
- Statussignale der NCUs

werden zyklisch übertragen. Istwerte und Statussignale einer Link-Achse werden der NCU aktuell bereitgestellt, die diese Achse interpoliert.

Nicht zyklische Kommunikation

- Austausch von Linkvariablen
- Warmstartanforderungen
- Aktivierung der Achscontainerdrehung
- Änderungen an NCU-globalen Maschinen- und Settingdaten.
- Wirksamsetzen von axialen Maschinendaten für Link-Achsen
- Alarme

Übertragungszeitbedarf

Für die Übertragung der **Sollwerte** zur Heimat-NCU einer Link-Achse und die Rückmeldung der **Istwerte** dieser Achse entstehen Verzögerungen. Bei einem Interpolationsverbund aus lokalen Achsen und Link-Achsen werden die Sollwerte für lokale Achsen der interpolierenden NCU durch die Steuerung um einen Interpolationstakt verzögert, so dass für die Interpolation konsistente Werte berücksichtigt werden.

Ist ein Kanal auf Istwerte einer Achse einer anderen NCU angewiesen, z. B. einer Spindel beim Gewindeschneiden, vergehen zwei Interpolationstakte, bis diese zur Verfügung stehen. Die damit erzeugten Sollwerte werden aus dem oben genannten Grund erst einen Interpolationstakt später an die Lageregler gegeben.

Verhalten des AXIS-VAR-Servers im Fehlerfall

Wenn für eine Achse keine Werte bereitgestellt werden können (z. B. weil die Achse eine Link-Achse ist), so wird ein Default-Wert (i.d.R. 0) zurückgegeben.

Für Testzwecke kann mit dem unten stehenden Maschinendatum der Achsdaten-Server sensitiv eingestellt werden, so dass er anstatt von Defaultwerten eine Fehlermeldung zurückgibt:

MD11398 AXIS_VAR_SERVER_SENSITIVE

0: Default-Wert

1: Fehlermeldung

2.6.3 Ausgabe vordefinierter Hilfsfunktionen bei NCU-Link

Vordefinierte Hilfsfunktionen M, S, F

Für Link-Achsen und Containerlink-Achsen wird eine vordefinierte Hilfsfunktion M, S, und F von NCK über NCU-Link zur Heimat-NCU der Linkachsen übertragen und von dort als System-Hilfsfunktionen an PLC ausgegeben. Diese System-Hilfsfunktionen werden vom PLC ausgewertet und wie folgt ausgegeben:

DB31, ... DBW86 (M-Funktion (`INT`) für Spindel)

DB31, ... DBD88 (S-Funktion (`REAL`) für Spindel)

DB31, ... DBD78 (F-Funktion (`REAL`) für Achse)

Hinweis

Nur für die vordefinierten Spindelhilfsfunktionen M3, M4, M5, M19 und M70 ist die Übertragung von NCU-Link zur Heimat-NCU relevant.

Weitere Informationen zu "Vordefinierte Hilfsfunktionen" siehe

Literatur /FB1/ Funktionshandbuch Grundfunktionen; Hilfsfunktionen an PLC(H2)

Alarm 14768

Kann die NCU-Link empfangene System-Hilfsfunktionen nicht über die VDI-Nahtstelle ausgegeben werden, weil z. B. der Übergabepuffer voll ist, so wird der Alarm 14768 "Kann die über NCU-Link empfangene axiale Hifsfunktion nicht ausgegeben" abgesetzt.

Beispiele

Auf der NCU_2 wird für den 7. Kanal ein NC-Programm mit M3 S1000 abgearbeitet. Diese Spindel entspricht der 5. Maschinenachse der NCU_1 und ist damit Link-Achse. Die Hilfsfunktionensausgabe erfolgt hier also für NCU_2 im Kanal 7 mit der Achsnummer 0, da die Link-Achse auf einer anderen NCU (hier NCU_1) liegt. Dies führt an der PLC der NCU_2 zur Ausgabe von:

DB21, ... DBW68 (erweiterte Adresse der M-Funktion (16 Bit-`INT`))

DB21, ... DBD70 (M-Funktion 1 (`INT` 3))

DB21, ... DBW98 (erweiterte Adresse der S-Funktion 1)

DB21, ... DBD100 (S-Funktion 1 (`REAL` 100))

Parallel dazu wird die Information der System-Hilfsfunktionen von NCU_2 über NCU-Link an NCU_1 (Heimat-NCU der Linkachse) übertragen. Von hier aus werden die System-Hilfsfunktionen M3 S1000 für die 5. Maschinenachse ausgegeben.

Die PLC auf NCU_1 versorgt dann die folgenden axialen Signale an der PLC-Anwenderschnittstelle für Maschinenachse 5:

DB35, ... DBW86 (M-Funktion (`INT` 3) für Spindel)

DB35, ... DBD88 (S-Funktion (`REAL` 100) für Spindel).

2.6.4 Randbedingungen für Link-Achsen

Alarmausgabe von Lageregler oder Antrieb

Ein Achsalarm wird immer auf der NCU ausgegeben, die den Interpolationswert erzeugt. Wird bei einer Link-Achse durch den Lageregler ein Alarm ausgegeben, wird dieser Alarm an die NCU gesendet, die aktuell für die Interpolation zuständig ist.

In der Annahme, dass Achsalarme, die ein Abfallen des NCK-Ready-Relais bewirken (Nck-NoReady), auf eine Störung des Antriebsbusses zurückzuführen sind, erfolgt die Alarmausgabe zusätzlich auf der NCU, an der die Achse bzw. der Antriebsbus physikalisch betrieben wird. Auch die Reaktion "Abfallen des Ready-Relais" wird nur hier aktiviert.

Alarmausgabe wegen NOT-AUS

Wird bei einer NCU eine NOT-AUS-Anforderung von PLC aktiviert, werden alle Achsen, die auf dieser NCU physikalisch mit den Antrieben verbunden sind, ins Nachführen geschaltet. Das bedeutet: Auch Achsen, die durch eine andere NCU interpoliert werden, sind davon betroffen. Da dieser Umstand auch auf den anderen NCUs dann keine weitere sinnvolle Bearbeitung zulässt, wird hier ein zusätzlicher Alarm erzeugt, der die Aufgabe hat, alle Achsbewegungen sofort zu stoppen.

Der zusätzlich erzeugte Alarm muss mit NC-Reset quittiert werden. Steht zu diesem Zeitpunkt noch der verursachende Alarm an, kann zwar der zu quittierende Alarm erfolgreich gelöscht werden, aber es wird ein weiterer Alarm (selbstlöschend) erzeugt, der ein Achsverfahren bzw. einen neuen Programmstart so lange verhindert, bis der verursachende Alarm gelöscht wurde.

Alarmausgabe mit der Alarmreaktion NCK-NoReady

Wird auf einer NCU ein schwerwiegender Alarm aktiviert, der ein Abfallen des NCK-Ready-Relais zufolge hat, so wirkt sich dies auch auf alle anderen NCUs aus, die eine Achse über Link-Kommunikation auf dieser NCU ansprechen. Auf den anderen NCUs wird jeweils ein weiterer Alarm aktiviert, der das sofortige Stoppen aller weiterer Achsen zufolge hat.

Alarm-Quittierung siehe NOT-AUS.

Kompensationen

Folgenden Kompensationen stehen für Link-Achsen **nicht** zur Verfügung:

- Durchhangkompensation (CEC)
- Spindelsteigungsfehlerkompensation (EEC)
- Quadrantenfehlerkompensation (QEC)

Ausschalten verbundener NCUs

Wird eine NCU eines Link-Verbunds ausgeschaltet oder durch NCK-Reset neu gestartet, sind auch alle anderen NCUs des Link-Verbundes davon betroffen. Auf diesen wird ein Alarm erzeugt und die aktuelle Bearbeitung abgebrochen.

Siehe "Konfiguration eines Link-Verbundes (Seite 174)".

Hochlauf eines NCU-Verbundes

Wird eine NCU eines Link-Verbundes über NCK-Reset neu gestartet, führen auch die anderen NCUs des Verbundes einen NCK-Reset aus.

Nibbeln und Stanzen

Für Nibbeln und Stanzen müssen Schnelle ein- und Ausgänge auf der NCU angeschlossen und parametriert werden, auf der interpoliert wird (auf der das Teileprogramm abgearbeitet wird). Die Befehle "Schnelles Nibbeln und Stanzen" z. B. `PONS`, `SONS` stehen nicht für Link-Achsen zur Verfügung.

Fahren auf Festanschlag

Wenn eine Achscontainer-Achse auf Festanschlag gehalten wird, kann nicht eines Achscontainerdrehung ausgeführt werden. Ein Anfahren von Anschlägen auf verschiedenen NCUs und anschließendes Klemmen ist ohne Einschränkung möglich.

Frames

In den Programmbefehlen für Frames sind Link-Achsen nur zulässig, wenn sie zugleich Geometrieachsen sind. Der Befehl ändert nur die Geometrie für den Kanal, in dem die Achse aktuell zugeordnet ist. Ein Frame-Befehl für eine Achse, die nicht Geometrieachse ist wird mit Alarm 14092 abgelehnt.

Umdrehungsvorschub

Das Settingdatum SD43300 SA_ASSIGN_FEED_PER_REV_SOURCE bezieht sich auf das logische Maschinenachsabbild und über dieses dann auf eine Maschinenachse (lokale oder Link-Achse).

2.6.5 Programmierung mit Kanal- und Maschinenachsbezeichnern

Kanalachsbezeichner

Beispiel:

WHENEVER $AA_IW[Z] < 10 DO ... ;Aktuelle Position der Z-Achse

Maschinenachsbezeichner

Beispiel:

WHENEVER $AA_IW[AX3] < 10 DO ... ;Abfrage nach aktueller Position der Maschinenachse AX3

Diese Form ist nur erlaubt, wenn zum Abfragezeitpunkt die Maschinenachse AX3 im Kanal bekannt ist.

Hinweis

Die mit Kanalachsbezeichnern benutzbaren Systemvariablen sind in der Programmieranleitung Arbeitsvorbereitung (Anhang) besonders gekennzeichnet.

2.6.6 Flexible Konfiguration

Einführung

Rundtaktmaschinen (RTM) und Mehrspindeldrehmaschinen (MS) stellen besondere Anforderungen an die flexible Zuordnung von Kanalachsen zu Maschinenachsen.

Anforderungsprofil

Bei der Weiterschaltung des Tisches der Rundtaktmaschine oder der Trommel der Mehrspindelmaschinen werden die Achsen bzw. Spindeln in eine neue Station bzw. Lage gebracht. Die NCU, die die Achsen einer Station als lokale Achsen steuert, muss die neu eingewechselten Achsen bzw. Spindel adressieren können. Die bisher adressierbare Achsen bzw. Spindeln können dafür abgegeben werden.

Lösung

Eine durch Maschinendaten spezifizierte Konfiguration der betroffenen Achsen in einem Achscontainer ermöglicht es, dass sich hinter einer konstant bleibenden Kanalachse nacheinander verschiedene Maschinenachsen befinden. Die Weiterschaltung von Drehtisch oder Trommel wird damit synchron mit dem Weiterschalten der im Achscontainer eingetragenen Achsen durchgeführt.

Achsen in einem Achscontainer können auch als Geometrieachsen konfiguriert sein.

Hinweis

Der Achscontainer hat keinen BAG-(Betriebsartengruppen-)Bezug.

D. h. die mitlaufende (werkstücktragende) Achse kann an den verschiedenen Bearbeitungsstationen von einer BAG in die andere wechseln.

2.7 Achscontainer

Achscontainer

Ein **Achscontainer** kann als Ringpuffer aufgefasst werden, in dem die Zuordnung

- von lokalen Achsen und / oder
- von Link-Achsen

zu Kanälen erfolgt.

Containerachsen

Die Achsen im Achscontainer werden auch als **Containerachsen** bezeichnet.

Die Zuordnungen können per Programmbefehl verschoben werden ("Drehung" des Ringpuffers).

Der Achsbegriff umfasst hier sowohl Achsen als auch Spindeln.

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

Hinweis

Die Drehung der Trommel bzw. des Rundtisches und die Drehung des Ringpuffers mit den zugeordneten Achseinträgen sind analog.

Container-Link-Achsen

Einträge in einem Achscontainer können auf Maschinenachsen verweisen, die auf einer anderen NCU eingerichtet sind als die restlichen Containerachsen. Derartige Containerachsen werden auch als Container-Link-Achsen bezeichnet.

Verweis auf Achscontainer

Die im Kapitel "Konfiguration von Link-Achsen" beschriebene Link-Achsen-Konfiguration lässt im logischen Maschinenachsabbild neben dem direkten Verweis auf lokale Achsen oder Link-Achsen den Verweis auf Achscontainer zu.

Ein solcher Verweis besteht aus:

- Container-Nummer
- Slot (Ringpuffer-Platz innerhalb des entsprechenden Containers)

Als Eintrag in einem Ringpuffer-Platz steht:

- eine lokale Achse **oder**
- eine Link-Achse

(jeweils Achse oder Spindel).

Achscontainer-Namen

Die Achscontainer können benannt werden mit dem Maschinendatum:

MD12750 $MN_AXCT_NAME_TAB

Die Achscontainer-Namen können benutzt werden:

- in den Achscontainer-Drehbefehlen `AXCTSWE()` und `AXCTSWED()` zur Adressierung der zu drehenden Container.
- bei der Abfrage der Zustände der Achscontainer mit den Systemvariablen:

$AC_AXCTSWA[ ]	(Freigabe der Achs-Container-Drehung im Kanal)
$AN_AXCTSWA[ ]	(Achs-Container-Drehung aktiv?)
$AN_AXCTAS[ ]	(Achs-Container aktuelle Verdrehung)

Festlegung des Containerinhalts

Für den Achscontainer 1 ... n wird die Hochlauf-Zuordnung von einem Achscontainer-Slot zu der jeweiligen Maschinenachse innerhalb eines NCU-Verbundes festgelegt mit dem Maschinendatum:

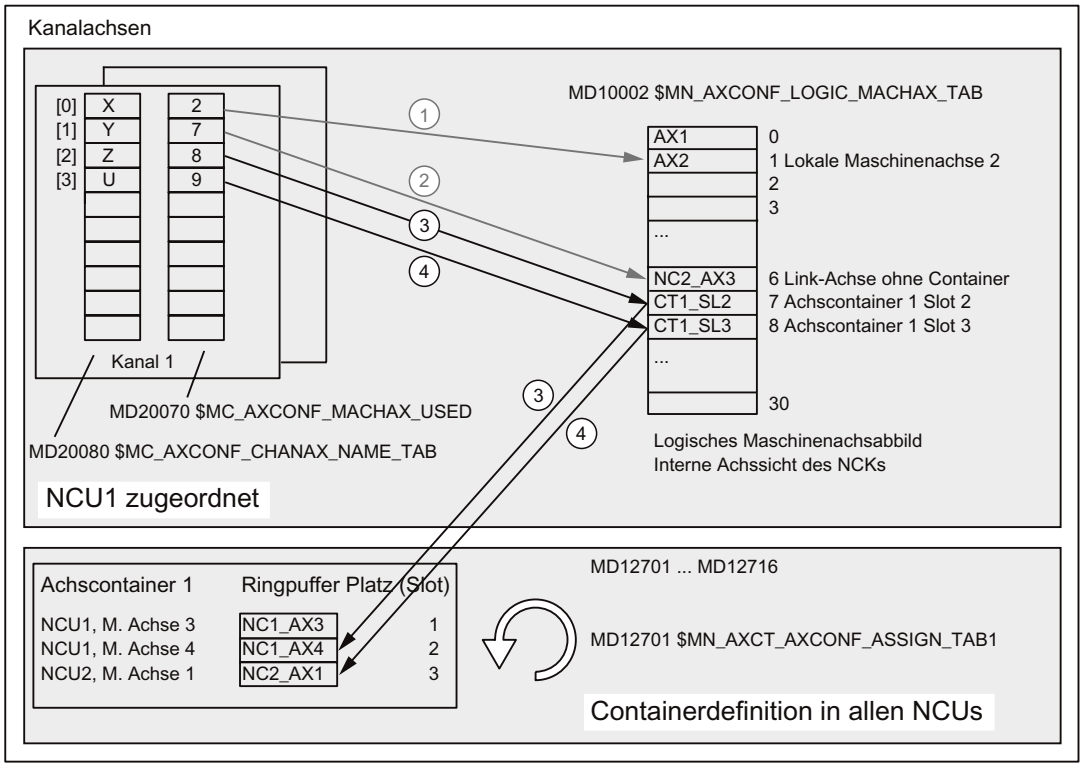
MD12701 ... MD12716 $MN_AXCT_AXCONF_ASSIGN_TAB1...n

Die Zuordnung eines Achscontainer-Slots zu dem jeweiligen Kanal erfolgt über die Maschinendaten:

MD20070 $MC_AXCONF_MACHAX_USED (Maschinenachsnummer gültig im Kanal)

MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB (Logisches NCK-Maschinenachsabbild)

Beispiel:



Nach dem Hochlauf der Steuerung ergibt sich für die Containerachsen folgende Zuordnung (Grundzustand vor einer ersten Container-Drehung):

③ **3. Kanalachse Z des Kanals 1 = 4. Maschinenachse der NCU1**

Erklärung:

Die 3. Kanalachse (MD20070 $MC_AXCONF_MACHAX_USED[2]) zeigt auf die 8. Machinenachse im logischen NCK-Maschinenachsabbild (MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[7]).

Dort ist die Containerslot-Projektierung CT1_SL2 hinterlegt.

Durch den Eintrag "NC1_AX4" in MD12701 $MN_AXCT_AXCONF_ASSIGN_TAB1[1] ist dem 2. Slot des 1. Achscontainers die 4. Maschinenachse der NCU1 zugeordnet.

④ **4. Kanalachse U des Kanals 1 = 1. Maschinenachse der NCU2**

Erklärung:

Die 4. Kanalachse (MD20070 $MC_AXCONF_MACHAX_USED[3]) zeigt auf die 9. Machinenachse im logischen NCK-Maschinenachsabbild (MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[8]).

Dort ist die Containerslot-Projektierung CT1_SL3 hinterlegt.

Durch den Eintrag "NC2_AX1" in MD12701 $MN_AXCT_AXCONF_ASSIGN_TAB1[2] ist dem 3. Slot des 1. Achscontainers die 1. Maschinenachse der NCU2 zugeordnet (⇒ **Container-Link-Achse**).

Für die restlichen im Bild gezeigten Kanalachsen gilt:

① **1. Kanalachse X des Kanals 1 = Lokale Maschinenachse 2**

② **2. Kanalachse Y des Kanals 1 = 3. Maschinenachse der NCU2 (⇒ Link-Achse)**

Container-Drehung

Die Inhalte der Achscontainer-Plätze sind veränderlich in der Weise, dass die Inhalte des Ringpuffers Achscontainer gemeinsam um ± n Schritte verschoben werden können.

Die Schrittzahl n wird durch das folgende Settingdatum je Achscontainer definiert:

SD41700 $SN_AXCT_SWWIDTH

Die Schrittzahl n wird modulo der tatsächlich belegten Anzahl Slots im Achscontainer ausgewertet. Dabei entsteht für alle Plätze eines Achscontainers ein neuer Inhalt (Ausnahmen: 0 und Platzzahl = Schrittzahl).

Über den aktuellen Zustand eines Achscontainers geben Systemvariablen Auskunft, die aus dem Teileprogramm und aus Synchronaktionen heraus lesend angesprochen werden können (siehe).

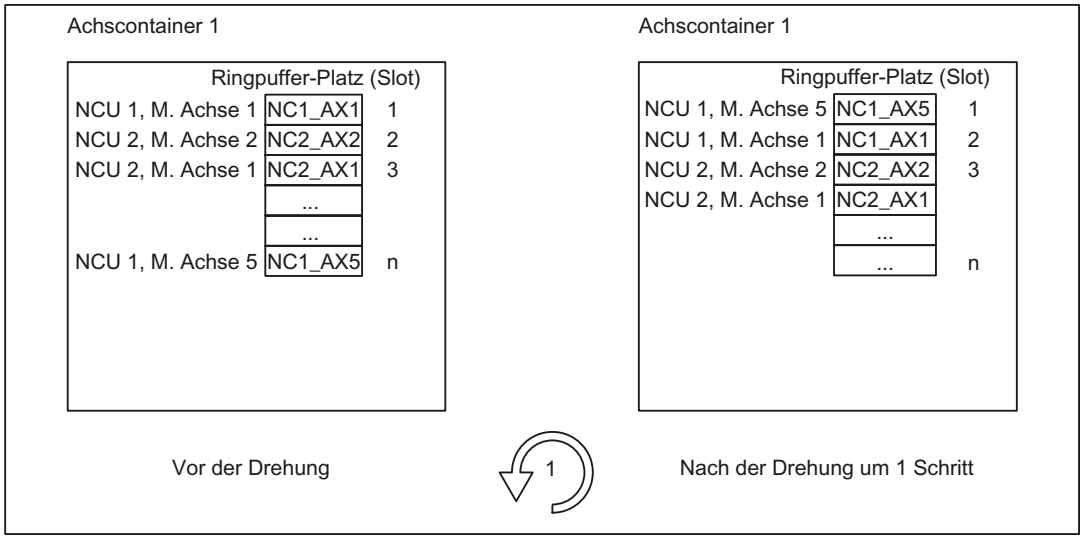


Bild 2-21 Verschiebung der Einträge auf den Achscontainer-Plätzen

Das Modell Achscontainer hat folgende Merkmale:

- Ein Kanal sieht immer eine feste Anzahl Achsen mit definierten Kanalachsnamen (logisches Maschinenachsabbild)
- Die "Drehung" des Achscontainers stellt für **alle** Kanäle, die Achsen im gleichen Achscontainer haben, neue Maschinenachsen ein.

Frames bei Achscontainer-Drehungen

Die Kanalachszuordnung zu Maschinenachsen kann sich bei Achscontainer- Drehungen ändern. 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 er z. B. durch die Programmierung der Basisframe-Masken erreichen.

Aktivierung der Achscontainer-Drehung

Die Anwendung muss durch Befehle im Teileprogramm zur Drehung des Achscontainers auf eine bestimmte Position dafür sorgen, dass die gewünschten lokalen oder Link-Achsen angesprochen werden.

Zum Beispiel muss bei einer Drehung der Trommel einer Mehrspindelmaschine in eine neue Lage durch Drehung des Achscontainers dafür gesorgt werden, dass jede Lage die neu eingewechselte Spindel adressiert.

Hinweis

Achscontainer können sowohl von verschiedenen Kanälen einer NCU als auch von Kanälen anderer NCUs gemeinsam benutzt werden.

Zeigen Achsen verschiedener Kanäle über das logische Maschinenachs-Abbild auf den gleichen Achscontainer, so sehen **alle** betroffenen Kanäle nach einer Drehung **andere Achsen**. Daraus folgt: Der Zeitpunkt für eine Drehung muss zwischen den Kanälen abgestimmt werden. Dies leisten die verfügbaren Sprachbefehle.

Jeder Eintrag im Achscontainer muss zu jeder Zeit dem richtigen Kanal zugeordnet sein. Die Systemvariablen (siehe) geben die Möglichkeit für Teileprogramm oder Synchronaktion, sich über den aktuellen Achscontainerzustand zu informieren.

Befehle für die Achscontainer-Drehung

Die oben skizzierte Pflicht zur Koordinierung der Kanäle, die einen Achscontainer gemeinsam benutzen, spiegelt sich in den Leistungen des Befehls `AXCTSWE` wider.

Notation:

Programmierung	Kommentar

`AXCTSWE(CTi)`	`; Der Funktionsname steht für:`
	`AXis ConTainer SWitch Enable`
	`CTi ist der Bezeichner des Achscontainers, der weitergeschaltet werden sollen.`

Die Schrittweite einer Schaltung muss containerspezifisch in dem folgenden Settingdatum hinterlegt werden:

SD41700 $SN_AXCT_SWWIDTH[Containernummer]

SD41700 ist über Link-Modul NCU-global verfügbar (alle NCUs, die über Link-Modul verbunden sind, sehen die gleichen Werte).

Funktion:

Jeder Kanal, dessen Achsen im angegebenen Container eingetragen sind, gibt die **Freigabe für eine Container-Drehung** (enable), wenn er mit der Bearbeitung der Lage / Station fertig ist. Wenn in der Steuerung die Freigaben **aller** Kanäle für die Achsen des Containers eingetroffen sind, erfolgt die Container-Drehung mit der im SD41700 hinterlegten Schrittweite (durch Beachtung eines ggf. existierenden Vorzeichens wird auch die Drehrichtung ausgewertet).

Für die Vereinfachung der Inbetriebnahme ist die folgende Variante vorgesehen:

Programmierung	Kommentar
`AXCTSWED(CT1)`	`; Der Funktionsname steht für:` `AXis ConTainer SWitch Enable Direct`

Der Achscontainer dreht sich gemäß der Vorgabe im Settingdatum:

SD41700 $SN_AXCT_SWWIDTH[Containernummer]

Der Aufruf darf nur verwendet werden, wenn sich die übrigen Kanäle, die Achsen im Container haben, im **RESET**-Zustand befinden.

Hinweis

Für die Befehle `AXCTSWE` und `AXCTSWED` können die über das folgende Maschinendatum vergebenen Achscontainer-Namen benutzt werden:

MD12750 $MN_AXCT_NAME_TAB

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(CTi)` 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 AXCT_SWWIDTH[Containernummer] 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.

Heimat-Kanal einer Achscontainer-Achse

Hat mehr als ein Kanal ein Zugriffsrecht (einen "Bezug") durch das Maschinendatum auf die Achse, so kann das Schreibrecht auf die Achse (Sollwertvorgabe) weitergegeben werden:

MD20070 $MC_AXCONF_MACHAX_USED

Durch das folgende Maschinendatum wird eine Achse standardmäßig einem Kanal zugeordnet:

MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN

Über das folgende Maschinendatum wird festgelegt, welche NCU beim Einschalten die Achse besitzt bzw. die Interpolationsvorgabe macht:

MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB

Da axiale Maschinendaten von Link-Achsen auf allen NCUs gleich sind, wird das folgende Maschinendatum nur ausgewertet, wenn auch die NCU das Schreibrecht auf die Achse hat (siehe MD10002):

MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN

Achstausch

Die Weitergabe des Schreibrechts auf die Achse (Sollwertvorgabe) durch `GET`, `RELEASE`, ..., funktioniert für eine Containerachse in gleicher Weise wie für eine gewöhnliche Achse. Das Schreibrecht kann nur zwischen den Kanälen **einer** NCU getauscht werden. Die Weitergabe des Schreibrechts kann nicht über NCU-Grenzen erfolgen.

2.7.1 Systemvariablen für Achscontainer

Zustände der Achscontainer

Die folgenden Systemvariablen erlauben es Teileprogrammen und Synchronaktionen, sich über den augenblicklichen Zustand eines Achscontainers zu informieren.

Legende:

r: Lesen

TP: Teileprogramm

SA: Synchronaktion

SW: Software Stand

n: Achscontainerbezeichner

Name	Typ/SW	Beschreibung/Werte	Index	Zugriff TP	Zugriff SA
$AC_AXCTSWA[n] (AXis ConTainer SWitch Active)	BOOLEAN /5	**Kanal**zustand der Achscontainer-Drehung/ 1: Der Kanal hat für den Achscontainer n die Achscontainerdrehung freigegeben und diese ist noch nicht beendet. 0: Die Achscontainerdrehung ist beendet Beispiele: s. Kap. "Achscontainer-Koordinierung"	Bezeichner	r	r
$AN_AXCTSWA[n]	BOOLEAN /5	**Achscontainer-Drehung**/ 1: Eine Achscontainer-Drehung wird augenblicklich auf dem Achscontainer n ausgeführt 0: Es ist keine Achscontainer-Drehung aktiv Beispiele: s. Kap. "Achscontainer-Koordinierung"	Bezeichner	r	r
$AN_AXCTAS[n] (AXis ConTainer Actual State)	INT /5	**Aktuelle Verdrehung Achscontainer** Für den Achscontainer n wird angegeben, um wie viele Plätze (slots) der Achscontainer aktuell weitergeschaltet wurde./ 0 bis (max. Zahl belegter Plätze im Achscontainer - 1) Die ursprüngliche Belegung steht nach Power On an. Sie hat den Wert 0.	Bezeichner	r	r

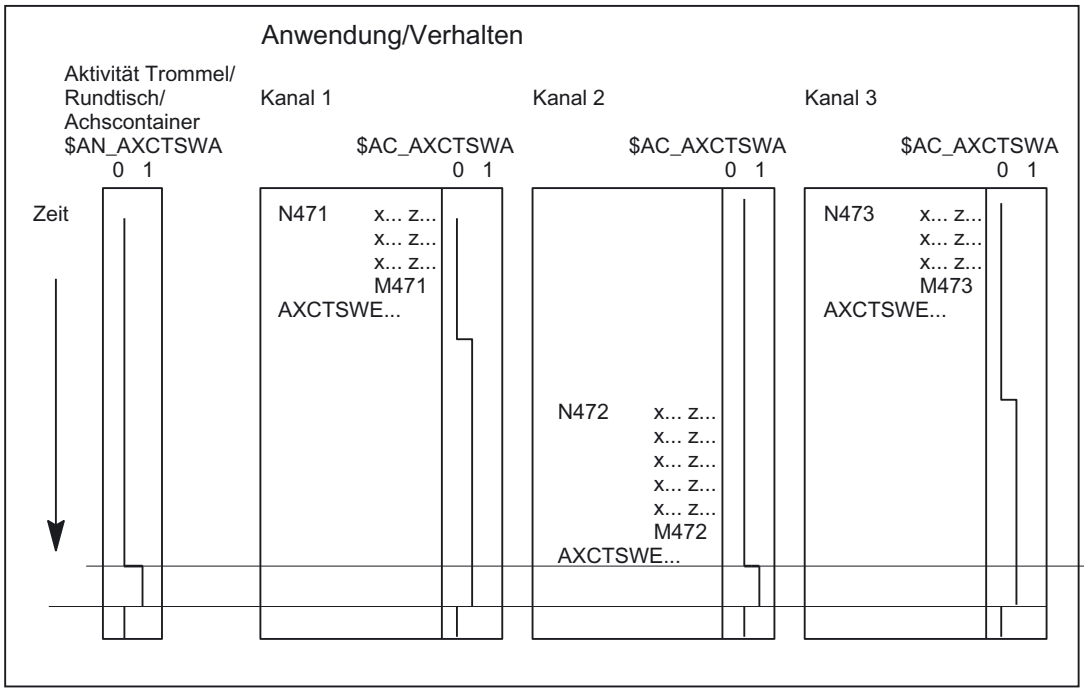


Bild 2-22 Achscontainerdrehung abhängig von der Freigabe durch betreffende Kanäle

2.7.2 Bearbeitung mit Achscontainer (schematisch)

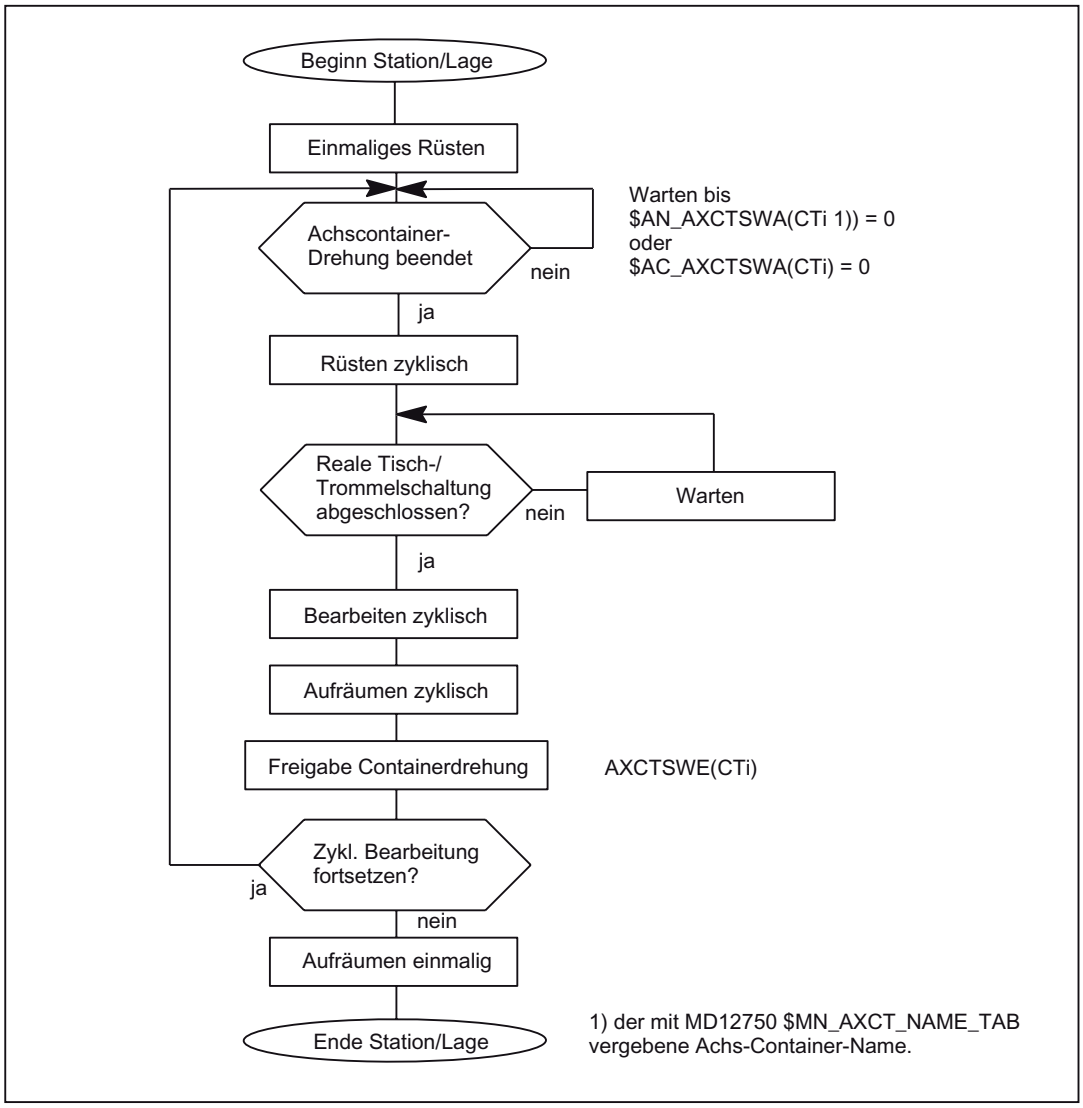


Bild 2-23 Schematische Bearbeitung einer Station/Lage

Hinweis

Ein Bearbeitungszyklus der NCU, der die Drehung des Rundtisches oder der Trommel bei Mehrspindelmaschinen obliegt, enthält die Abfrage der Freigaben für Containerdrehung aller betroffenen NCUs. Liegen alle Freigaben vor, wird in die nächste Lage/Station geschaltet. Die Achscontainer werden entsprechend gedreht.

2.7.3 Verhalten der Achscontainer bei Power On

Bei `Power On` wird immer der in den Maschinendaten festgelegte Zustand eingenommen, unabhängig davon, in welchem Zustand des Achscontainers ausgeschaltet wurde. D. h. der Anwender muss eine Differenz zwischen dem Istzustand der Maschine und der Voreinstellung erkennen und diese durch Vorgabe entsprechender Achscontainer-Drehungen ausgleichen. Dies kann z. B. durch ein ASUP mit `AXCTSWED` in einem Kanal erfolgen, während sich die anderen Kanäle noch im `RESET`-Zustand befinden.

2.7.4 Verhalten der Achscontainer bei Betriebsartenwechsel

Eine Containerachse, deren Achscontainer für die Drehung freigegeben wurde, kann nicht in JOG verfahren werden. In der Betriebsart JOG ist eine Achscontainer-Drehung nur über ASUP möglich.

2.7.5 Verhalten der Achscontainer bei ASUPs

Die Freigabe zur Achscontainer-Drehung kann nicht rückgängig gemacht werden, d. h. ist in einem ASUP eine Freigabe zur Achscontainer-Drehung erfolgt, bleibt sie über das Ende des ASUPs hinaus wirksam.

2.7.6 Verhalten der Achscontainer bei RESET

Ein Reset hebt die Freigabe der Achscontainer-Drehung auf. Der zurückgesetzte Kanal verzichtet damit auf Beteiligung an der Achscontainer-Drehung. Die Freigaben verbleibender aktiver Kanäle können eine Drehung bewirken. Sind alle Kanäle außer einem zurückgesetzt, so kann dieser eine mit `AXCTSWED` direkt die Drehposition einstellen.

2.7.7 Verhalten der Achscontainer bei 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.

2.7.8 Randbedingungen für Achscontainer/-Drehungen

Hinweis

Der Anwender muss durch Programmierung dafür sorgen, dass

- nach der Containerschaltung die richtigen Nullpunktverschiebungen wirken
- und während des Containerswitches keine Transformationen aktiv sind.

Axiale Maschinendaten

Befindet sich eine Achse in einem Achscontainer, so müssen bestimmte axiale Maschinendaten zum Zeitpunkt des Maschinendaten-Wirksamsetzens für alle Achsen des Achscontainers gleich sein. Dies wird dadurch erreicht, dass bei der Änderung eines solchen Maschinendatums die Änderung auf allen Containerachsen und auf allen NCUs, die diese Achse sehen, erfolgt. Dabei wird der Hinweis: "Achtung dieses MD wird für alle Containerachsen gesetzt" ausgegeben.

Beim Hochlauf werden alle solche axialen Maschinendaten auf die Werte der Maschinenachse auf **Platz 1 des Achscontainers** synchronisiert. D. h. alle Achsen im Achscontainer übernehmen die entsprechenden Maschinendaten von der Maschinenachse, die auf Platz 1 des Achscontainers eingetragen ist. Werden dadurch Maschinendaten mit einem anderen Wert überschrieben, so wird der Hinweis ausgegeben: "Die axialen MDs der Achsen im Achscontainer <n> wurden angepasst".

Wird ein Platz des Achscontainers neu belegt (Beschreiben der Maschinendaten: MD12701-12716 AXCT_AXCONF_ASSIGN_TAB<n>), so wird der Hinweis ausgegeben: "Beim nächsten Hochlauf werden die MDs der Achsen im Achscontainer <n> angepasst".

Die entsprechenden axialen Maschinendaten sind mit dem Attribut *containerEqual*(gleich für alle Achsen im Achscontainer) gekennzeichnet. Bei NCU-Link erfolgt die Achscontainer-Definition **auf der Master-NCU** (siehe Kapitel "NCU-Link").

Achszustände

Ist eine Containerachse im Achsbetrieb bzw. als positionierende Spindel aktiv (`POSA`, `SPOSA`) und es soll eine Achscontainer-Drehung erfolgen, so kann die Drehung erst erfolgen, wenn die Containerachse ihre Endposition erreicht hat.

Eine Containerachse, die als **Spindel** aktiv ist, **dreht** während Achscontainer-Drehung **weiter**.

`SPCON` (Umschalten in Lageregelung) haftet an der physikalischen Spindel, d. h. dieser Zustand wird bei einer Achscontainer-Drehung mit weitergegeben. `SETMS` (Masterspindel) dagegen bezieht sich auf den Kanal und bleibt bei einer Achscontainer-Drehung im Kanal erhalten.

Bahnsteuerbetrieb G64

Eine Achscontainer-Drehung unterbricht den G64-Betrieb eines Kanals, der eine zugehörige Containerachse als Kanalachse hat, auch wenn diese nicht dem Bahnverbund angehört. Diese Unterbrechung findet allerdings erst statt, wenn eine Achse des gedrehten Achscontainers wieder programmiert wird.

PLC-Achsen

Soll eine Containerachse, deren Achscontainer für die Drehung freigegeben wurde, zur PLC-Achse werden, so wird diese Anforderung gespeichert, der Wechsel zur PLC-Achse erfolgt jedoch erst **nach** Abschluss der Achscontainer-**Drehung**.

Kommandoachsen

Eine Containerachse, deren Achscontainer für die Drehung freigegeben wurde, kann nicht zur Kommandoachse werden. Die Verfahranordnung wird im Kanal gespeichert und nach dem Beenden der Achscontainer-Drehung ausgeführt.

Eine Ausnahme bilden die Synchronaktionen M3, M4, M5 und eine bewegungsändernde S-Funktion: Wenn eine Achscontainer-Drehung aktiv ist und die Spindel an eine andere NCU abgegeben wird, wird der Alarm 20142 (Kanal %1 Kommandoachse %2: ungültiger Achstyp) gemeldet. Bei diesen Synchronaktionen wird eine Kanalachse nicht zur Kommandoachse, sondern ihr Achszustand bleibt erhalten. Diese Synchronaktionen können nicht gespeichert werden.

Literatur: /FBSY/ Funktionshandbuch Synchronaktionen

Pendelachsen

Eine Containerachse, deren Achscontainer für die Drehung freigegeben wurde, kann nicht zur Pendelachse werden; dies geschieht erst nach Beenden der Achscontainer-Drehung. Die Aktivierung zur Pendelachse bleibt bestehen.

Achskopplungen

Solange eine Achskopplung aktiv ist, an der eine Containerachse beteiligt ist, kann keine Achscontainer-Drehung erfolgen. Die Kopplung muss vor der Achscontainer-Drehung abgewählt (`COUPOF`) werden und kann danach wieder angewählt (`COUPON`) werden. Es ist jedoch kein neues `COUPDEF` notwendig.

Hauptlauf Korrekturwerte

Die Hauptlauf-Korrekturwerte (DRF-Verschiebung, Online-Werkzeugkorrektur, Synchronaktions-Offset, Compilezyklen-Offset) einer Kanalachse, die einen Container-Slot belegt, bleiben über die Achscontainer-Drehung hinweg erhalten. Die Externe Nullpunktverschiebung kann nicht über die Achscontainer-Drehung hinweg erhalten bleiben, da sie sich auf eine Maschinenachse bezieht. Ist eine Externe Nullpunktverschiebung aktiv, so wird die Achscontainer-Drehung mit dem Alarm 4022 abgelehnt.

Axialer Frame

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, so **ändert sich durch die Drehung auch der axiale Frame**. Falls die beiden Frames nicht übereinstimmen, so erfolgt eine Synchronisation (internes `REORG`).

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.

Transformationen

Ist die Achscontainer-Achse eine Spindel und diese an der Transformation beteiligt, so muss die **Transformation** vor der Freigabe zur Achscontainer-Drehung **abgewählt werden**. Andernfalls wird der Alarm 17605 gemeldet.

Gantry-Verbund

Gantry-Achsen können nicht Achsen eines Achscontainers sein.

Antriebsalarme

Bei einem anstehenden Antriebsalarm für eine Achscontainer-Achse kann keine Drehung des zugehörigen Achscontainers durchgeführt werden; der Alarm muss erst beseitigt werden.

2.8 Link-Kommunikation

2.8.1 Link-Variablen

Funktion

In komplexen Anlagen sind oftmals mehrere NCUs mit jeweils mehreren Kanälen vorhanden. Zur systemweiten Koordinierung der Fertigungsabläufe besitzt jede NCU einen Link-Kommunikationskanal. Über diesen kann jede NCU mit allen anderen am Link-Verbund beteiligten NCUs zyklisch Daten austauschen.

Der Link-Kommunikationskanal basiert auf einem auf jeder NCU vorhandenen Speicherbereich, dem sogenannten Link-Variablen-Speicher. Sowohl die Größe und als auch die Datenstruktur des Link-Variablen-Speichers können vom Anwender / Maschinenhersteller anlagenspezifisch festgelegt werden. Die Adressierung der im Link-Variablen-Speicher abgelegten Daten erfolgt über sogenannte Link-Variablen.

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

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

Voraussetzungen

Zur Verwendung von Link-Variablen für einen NCU-übergreifende Datenaustausch müssen folgende Voraussetzungen erfüllt sein:

- Der Link-Verbund muss installiert und konfiguriert sein. Siehe "Konfiguration eines Link-Verbundes (Seite 174)".
- Die Link-Kommunikation muss aktiviert sein:

  MD18780 $MN_MM_NCU_LINK_MASK (Aktivierung der NCU-Link Kommunikation)

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 NCU-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 NCUs übertragen. Die Zeitverzögerung bis zur Aktualisierung der Link-Variablen-Speicher durch die Link-Kommunikation beträgt im Normalfall zwei IPO-Takte.

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	±(2,2*10^{-308} … 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 ⇒ Index i = n * Bytes, mit n = 0, 1, 2, ...
  - `$A_DLB[i]`: i = 0, 1, 2, ...
  - `$A_DLW[i]`: i = 0, 2, 4, ...
  - `$A_DLD[i]`: i = 0, 4, 8, ...
  - `$A_DLR[i]`: i = 0, 8, 16, ...

Schreiben
Das Schreiben einer Link-Variablen wird Hauptlauf-synchron ausgeführt.

Lesen
Beim Lesen einer Link-Variablen wird Vorlaufstop 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 NCUs
- Lesen eines Datums bevor es von einem Kanal der eigenen NCU oder einer anderen NCU aktualisiert wurde

ACHTUNG
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.

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 IPO-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 IPO-Takt geschrieben werden können.

Zeitverhalten beim Schreiben

Das Schreiben der Link-Variablen erfolgt Hauptlauf-synchron. Den neuen Wert können die anderen Kanäle der eigenen NCU spätenstens im nächsten IPO-Takt lesen. Im eigenen Kanal kann der neue Wert im nächsten Satz gelesen werden.

Die Kanäle der anderen NCUs des Link-Verbundes sehen den neuen Wert normalerweise nach zwei IPO-Takten. Aufgrund der begrenzten Bandbreite kann es aber zu Verzögerungen bei der Übertragung von Schreibaufträgen auf die anderen NCUs des Link-Verbundes (Message-Delay) kommen. Mögliche Situationen sind:

- Schreiben einer großen Anzahl von Link-Variablen in einem IPO-Takt
- Schreiben von Link-Variablen und zusätzlich eine Anforderung einer Achscontainer-Drehung im gleichen IPO-Takt
- Schreiben von Link-Variablen und zusätzlich Übertragung eines Alarms im gleichen IPO-Takt

Synchronisation eines Schreibauftrags

Ist es in bestimmten Anwendungsfällen erforderlich, dass der neue Wert einer Link-Variable in genau zwei IPO-Takten auf die anderen NCUs des Link-Verbundes übertragen wird, kann das Schreiben über eine Synchronaktion und der Systemvariablen $A_LINK_TRANS_RATE abhängig von der Anzahl freier Schreibelemente ausgeführt werden. Die Systemvariable enthält den dabei Wert wieviel Schreibaufträge im aktuellen IPO-Takt noch auf die anderen NCUs des Link-Verbundes übertragen werden können.

Programmbeispiel

```
N120 WHEN $A_LINK_TRANS_RATE > 0 DO $A_DLB[0] = 9
N125 WHEN $A_LINK_TRANS_RATE > 0 DO $A_DLB[1] = 7
```

Das von der Link-Variable `$A_DLB[0]` adressiert Byte wird in `N120` erst dann geschrieben, wenn der Schreibauftrag im gleichen IPO-Takt auf die anderen NCUs des Link-Verbundes übertragen werden kann. Ist dies der Fall, wird die Link-Variable geschrieben und gleichzeitig die Systemvariable dekrementiert. Anschließend erfolgt ein Satzwechsel und `N125` wird in gleicher Weise bearbeitet.

Kann in `N120` der Schreibauftrag nicht sofort ausgeführt werde, wird versucht den Schreibvorgang im nächsten IPO-Takt auszuführen.

Beispiel

Für die Link-Kommunikation werden folgende Daten benötigt:

Datenformat	Anzahl	benötigte Bytes
BYTE	2	2
WORD	1	2
DWORD	3	12
REAL	1	8
		24 (benötigte Größe des Link-Variablen-Speichers)

Die Daten werden unter Berücksichtigung der Datenformatgrenzen im Link-Variablen-Speicher folgendermaßen angeordnet:

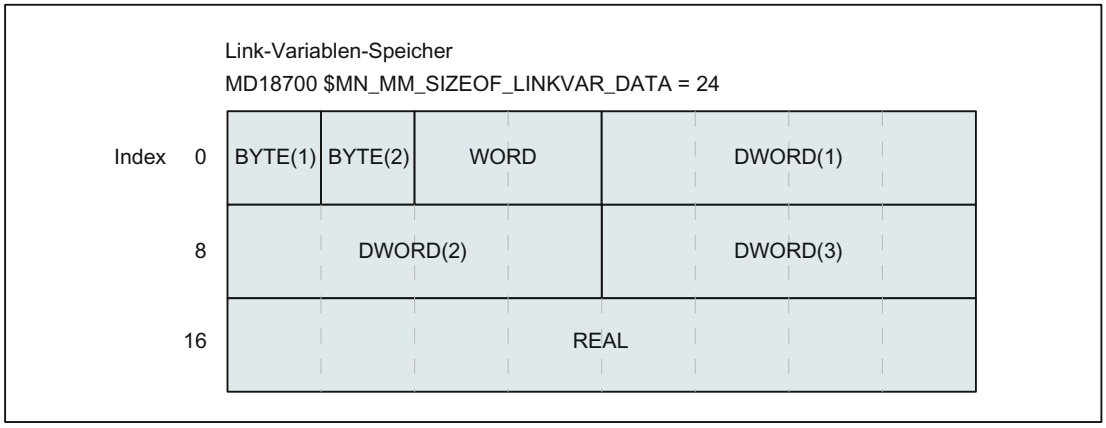


Bild 2-24 Beispiel: Speicheraufteilung des Link-Variablen-Speichers

Hinweis

Speicherstrukturierung

Die Anordung der Daten im Link-Variablen-Speicher ist prinzipiell beliebig und könnte unter Berücksichtigung der Datenformatgrenzen auch in anderer Form erfolgen.

Der Zugriff über 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

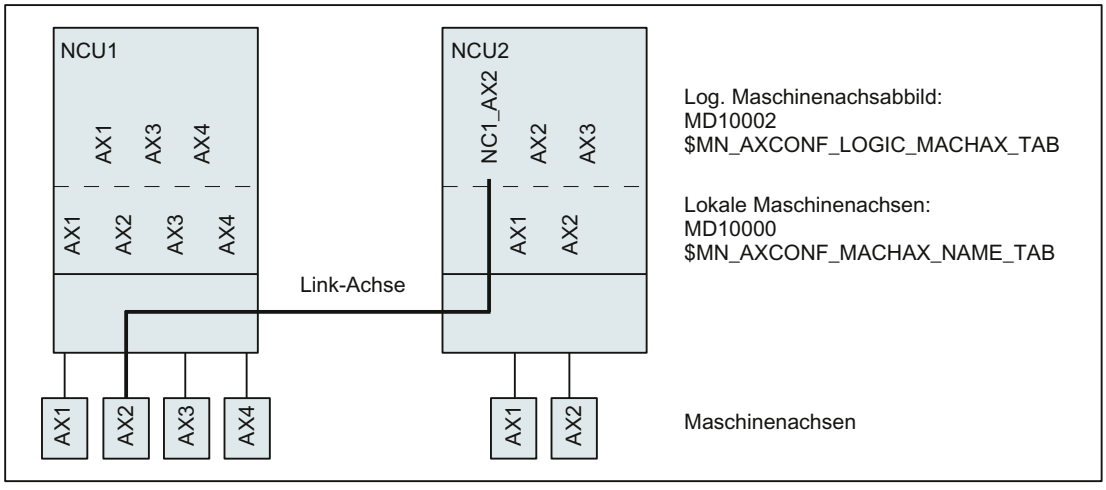
Siehe auch

Lesen von Antriebsdaten über Link-Variablen (Seite 172)

2.8.2 Lesen von Antriebsdaten über Link-Variablen

Aufgabe

In einer Anlagen sind 2 NCUs mit den Bezeichnungen NCU1 und NCU2 vorhanden. Die beiden NCUs sind über NCU-Link miteinander verbunden. An der NCU1 sind mehrere Maschinenachsen angeschlossen, von denen die Achse AX2 als Link-Achse von NCU2 interpolatorisch verfahren wird. Über Link-Kommunikation soll der Stromistwert der Achse AX2 von der NCU1 zu Auswertung an die NCU2 übertragen werden. Das folgende Bild zeigt den prinzipiellen Systemaufbau.



Voraussetzung

Der Stromistwert der Achse AX2 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 Antriebs-Istwerte)

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 IPO-Takt der Stromistwert $VA_CURR der Achse AX2 über die Link-Variable $A_DLR[0] (REAL-Wert) in die ersten 8 Bytes des Link-Variablen-Speichers geschrieben:

Programmcode

```
N111 IDS=1 WHENEVER TRUE DO $A_DLR[0]=$VA_CURR[AX2]
```

NCU2

Über eine statische Synchronaktion wird zyklisch im IPO-Takt der per Link-Kommunikation übertragene Stromistwert $VA_CURR der Achse AX2 über die Link-Variable $A_DLR[0] gelesen. Ist der Stromistwert größer als 23 A, wird der Alarm 61000 angezeigt.

Programmcode

```
N222 IDS=1 WHEN $A_DLR[0] > 23.0 DO SETAL(61000)
```

2.9 Konfiguration eines Link-Verbundes

Einführung

In den vorausgegangenen Kapiteln wurden die Konfiguration von Link-Achsen und Achscontainern beschrieben. Beide bedingen, dass zwischen den betreffenden NCUs eine Link-Kommunikation besteht. Die Einrichtung der Link-Kommunikation erfolgt durch:

- die Hardware der Link-Module
- **Literatur:** /PHD/ Handbuch Projektierung NCU 840D
- Maschinendaten

Das folgende Kapitel erläutert die Verwendung der erforderlichen Maschinendaten.

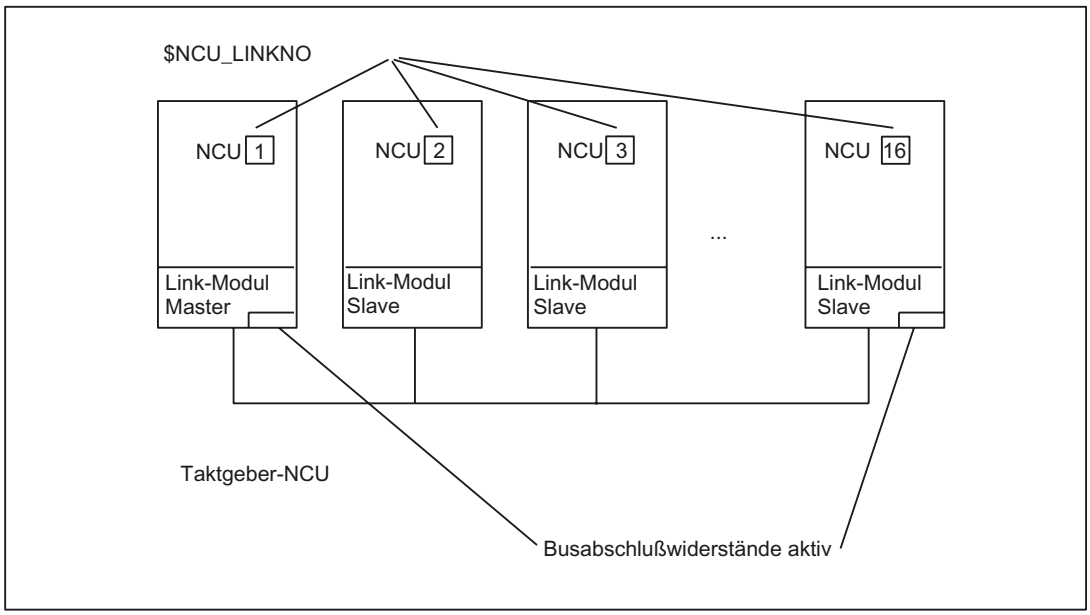


Bild 2-25 Link-Verbund

Link-Verbund

Ein Link-Verbund besteht aus wenigstens 2 und höchstens 16 mit Link-Modulen verbundenen NCUs.

- Das Link-Modul Master

  (MD12510 $MN_NCU_LINKNO = 1)

  nimmt eine ausgezeichnete Rolle ein. Es synchronisiert den Interpolationstakt und baut im Hochlauf die Kommunikation der Slaves auf. Es wird empfohlen, die NCU-Nummern aufsteigend stetig für die Slave-Module zu vergeben.
- Das erste und das letzte Modul im physikalischen Strang müssen die Busabschlusswiderstände aktivieren.
- Alle NCUs eines Link-Verbundes müssen den gleichen Software-Stand haben.

Maschinendaten

MD18780 $MN_MM_NCU_LINK_MASK

Das oben stehende Maschinendatum sorgt dafür, dass die Link-Kommunikation aufgebaut wird. Es stellt in den mit Link-Modulen ausgestatteten NCUs den für die Kommunikation erforderlichen dynamischen Speicherplatz bereit.

Das folgende Maschinendatumnennt die Datenübertragungsrate der Link-Kommunikation mit folgender Zuordnung: :

MD12540 $MN_LINK_BAUDRATE_SWITCH

Eingestellter Wert/Rate

0: 9,600 KBd

1: 19,200 KBd

2: 45,450 KBd

3: 93,750 KBd

4: 187,500 KBd

5: 500,000 KBd

6: 1,500 MBd

7: 3,000 MBd

8: 6,000 MBd

9: 12,000 MBd (Standardeinstellung)

Das Maschinendatum:

MD12550 $MN_LINK_RETRY_CTR

nennt die maximale Anzahl der automatischen Wiederholungen bei der Link-Kommunikation, wenn bei der Telegrammübertragung ein Fehler aufgetreten war.

Das folgende Maschinendatum nennt die Anzahl der an der Link-Kommunikation beteiligten Link-Module. :

MD12530 $MN_LINK_NUM_OF_MODULS

Das folgende Maschinendatum ordnet einer NCU eine logische Link-Nummer zu, unter der sie im Zusammenhang mit Link-Achsen und Link-Kommunikation identifiziert wird:

MD12510 $MN_NCU_LINKNO

Die Identifikationen sind unabhängig von der physikalischen Reihenfolge der Module im Link-Strang vergebbar. Der Modul mit NCU_LINKNO = 1 ist **Master**.

⚠ WARNUNG
Die Vergabe der NCU_LINKNO muss eindeutig sein. Im Fehlerfall tritt ein Alarm auf.

Das folgende Maschinendatum legt für die SW fest, welche NCUs die Busabschlusswiderstände haben:

MD12520 $MN_LINK_TERMINATION

Die gesetzten Werte beziehen sich auf die mit folgendem Maschinendatum festegelegten Einträge:

MD12510 $MN_NCU_LINKNO

0 entspricht der ersten Festlegung, 1 der zweiten, usw. aus dem Maschinendatum:

MD12510 $MN_NCU_LINKNO.

Hinweis

Die Besetzung des folgende Maschinendatums ist nur für die Prototyp-HW des Linkmoduls erforderlich:

MD12520 $MN_LINK_TERMINATION

Es hat nur für die Software Bedeutung.

Die NCUs, die **physikalisch** am **Anfang** und am **Ende** des Busses angeschlossen sind, müssen die Abschlusswiderstände im Stecker aktivieren. Die Maßnahme ist für das Funktionieren der Link-Kommunikation zwingend.

Das folgende Maschinendatumlegt für den Hochlauf fest, welche NCU eines NCU-Verbundes die Sollwerterzeugung für die Achse übernimmt (Master NCU) :

MD30554 $MA_AXCONF_ASSIGN_MASTER_NCU

Das folgende Maschinendatum legt für den Hochlauf fest, dass der Antrieb der Achse in Betrieb genommen werden soll, obwohl die Achse von einer anderen NCU verfahren wird:

MD30560 $MA_IS_LOCAL_LINK_AXIS

Es wird nur ausgewertet, wenn die für einen Link-Verbund erforderlichen Maschinendaten gesetzt sind, die Link-Kommunikation aber aufgrund eines Fehlers nicht zustande kommt.

Hinweis

Die Anzahl der Link-Achsen und die Anzahl der Schreibelemente kann dazu führen, dass der Interpolationstakt erhöht werden muss.

2.10 Kommunikation im Link-Verbund

Obwohl die Kummunikation über Link-Module eine schnelle Kommunikation ist, müssen die folgenden Aspekte bei der Projektierung berücksichtigt werden.

Datentransport

Zur Kommunikation der Daten wird ein zyklischer und ein nichtzyklischer Dienst verwendet. Der zyklische Datenbereich stellt sicher, dass die Daten in jedem Interpolationstakt übertragen werden.

Im nichtzyklischen Dienst werden z. B. Maschinendaten, Link-Variable und Daten für eine Container-Drehung verwendet. Wenn gleichzeitig zuviele Daten für die nichtzyklische Link-Kommunikation ausgewählt wurden, entscheidet eine intern festgelegte Priorität die Reihenfolge der Übertragung. Die Istwert- bzw. die Sollwertdaten der Lead-Link-Achse werden mit dem zyklischen Dienst übertragen.

Abhängigkeiten

Die Zeiten für die Kommunikation werden bestimmt durch:

- Anzahl der zyklisch auszutauschenden Daten

  (Anzahl Linkachsen, Anzahl Lead-Link-Achsen (s. unten), Anzahl Achsen in einem NCU-übergreifenden Container mit hoher Priorität und Anzahl Link-Variablen, die pro IPO-Takt ausgetauscht werden sollen)
- Typ und Geschwindigkeit der NCUs
- Geschwindigkeit der Linkmodule (z. Z. 12 MBaud)
- Punkt zu Punkt-Protokoll: Die zyklische Kommunikationslast steigt massiv mit jeder weiteren NCU, wie das schematische Bild "Anstieg der Kommunikationszeit mit der Anzahl der über Link verbundenen NCUs (Skalierung s. u. Abhängigkeiten)" veranschaulicht.

Der Interpolationstakt muss für alle im Lead-Link-Achsverbund stehenden NCUs gleich eingestellt sein!

Link-Resourcen

Das unten stehende Maschinendatum regelt die Vergabe der Send-/Empfangspuffer für Link-Verbindungen:

MD18781 $MN_NCU_LINK_CONNECTIONS

Es stehen je NCU maximal 32 solcher Puffer zur Verfügung.

Für die Soll-/istwertübertragung einer Linkachse wird ein solcher Puffer vorrangig reserviert. Nur die verbleibenden können für nicht zyklischen Datenaustausch (Alarme-, Containerswitches-, Linkvariablen-Übertragungen) benutzt werden. Dabei darf die Maximalzahl 32 nicht überschritten werden.

Ist das unten stehende Maschinendatum mit 0 gesetzt, bestimmt die Software den Bedarf mit 25 Verbindungen selbst.:

MD18781 $MN_NCU_LINK_CONNECTIONS

Ein Wert ungleich 0 legt die Anzahl der nicht zyklischen Verbindungen zu anderen NCUs explizit fest.

Beispiele

Ein Achscontainer enthalte 12 Slots. 3 Achsen sind lokal auf NCU1, je 3 Achsen sind Link-Achsen, die auf NCU 2, NCU3, NCU4 liegen. Das MD ist mit Wert 0 gesetzt.

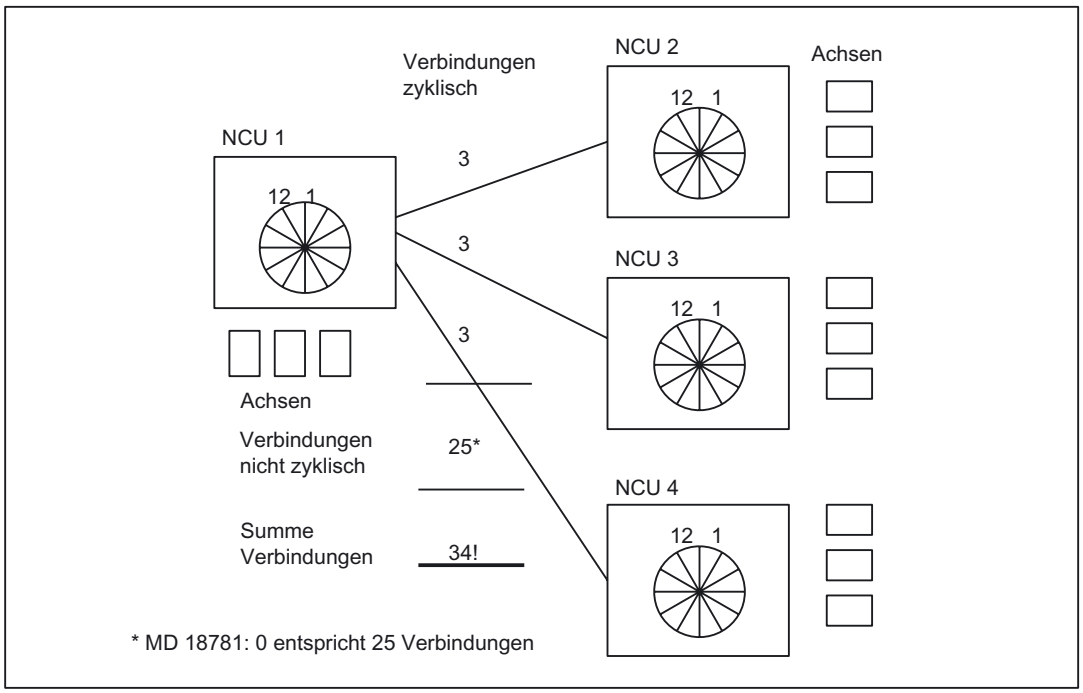


Bild 2-26 Resourcen reichen nicht

Ein Achscontainer enthalte 12 Slots. 4 Achsen sind lokal auf NCU1, je 4 Achsen sind Link-Achsen auf NCU 2, NCU3. Das Maschinendatum ist mit Wert 9 gesetzt:

MD18781 $MN_NCU_LINK_CONNECTIONS

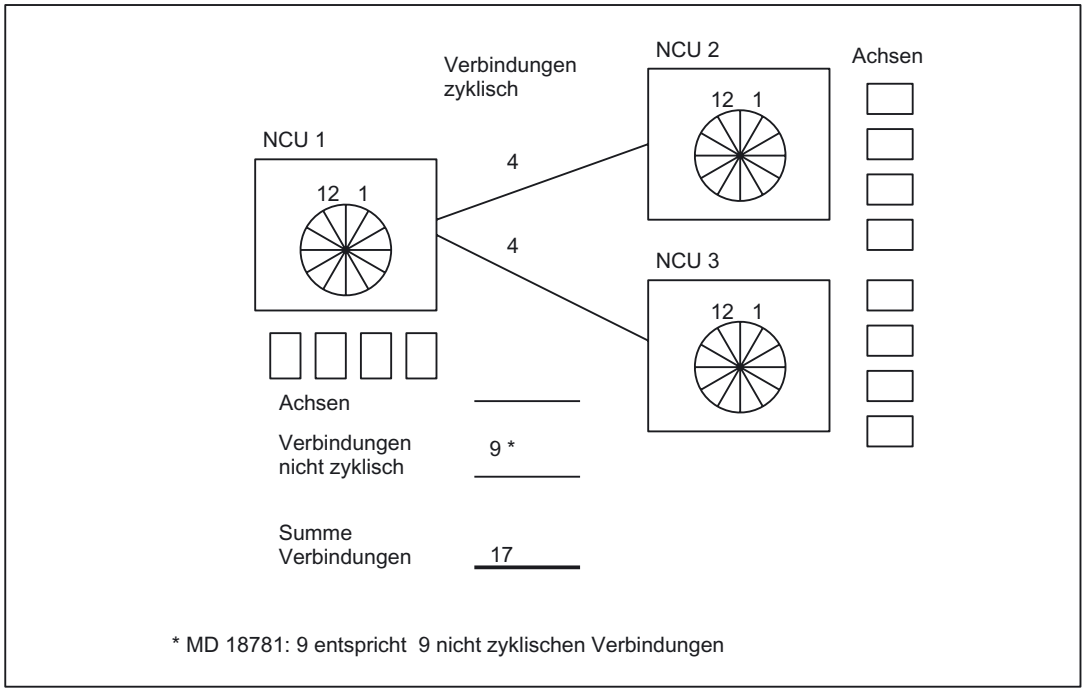


Bild 2-27 Resourcen reichen

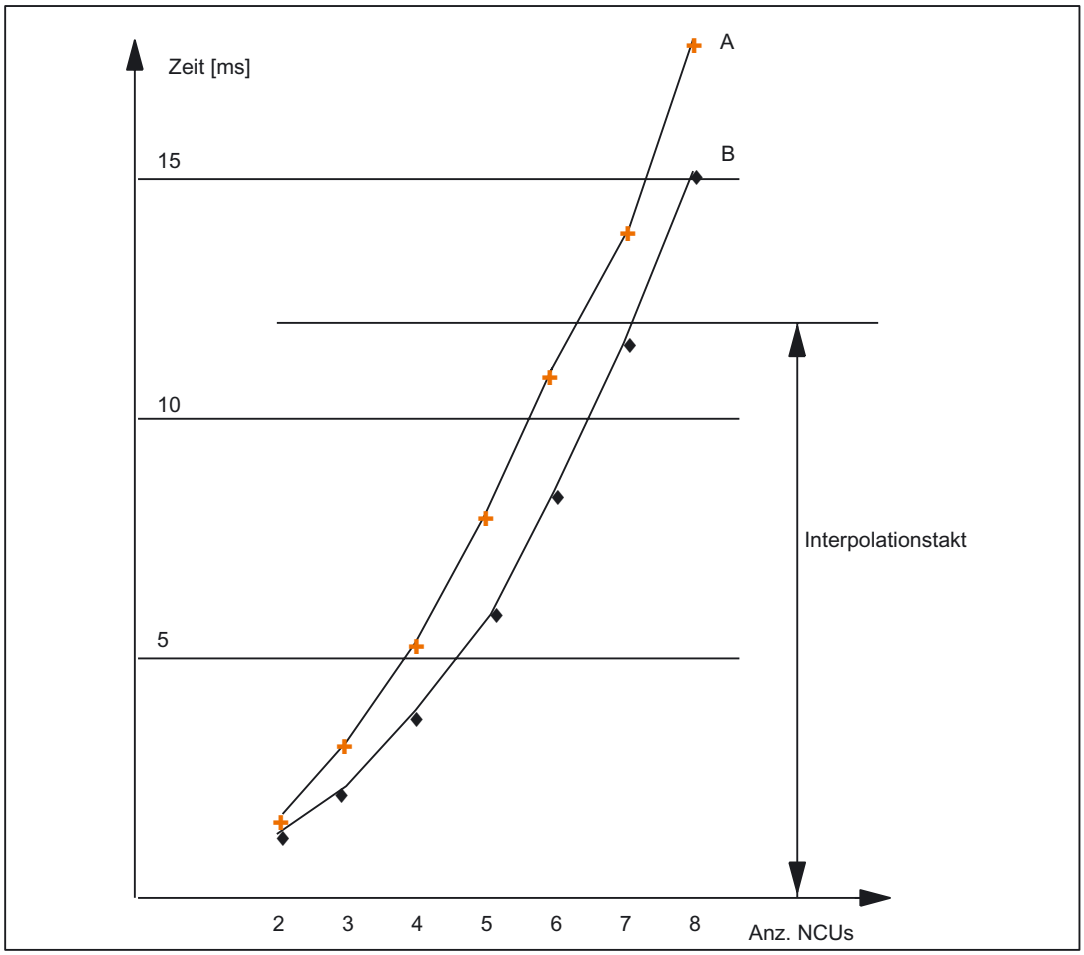


Bild 2-28 Anstieg der Kommunikationszeit mit der Anzahl der über Link verbundenen NCUs (Skalierung s. u. Abhängigkeiten)

Projektierungsgrenze

Im obigen Bild wird schematisch dargestellt, wie der Kommunikationsaufwand mit der Anzahl der NCUs wächst.

Kurvenzug A:

Zeitbedarf für den Austausch von Linkvariablen-/Maschinendaten-Information und der Lead-Link-Achs-Information (eine Lead-Link-Achse) zwischen der Leitwert gebenden NCU und weiteren NCUs, die Folgeachsen abhängig von der Leitachse (Lead-Link-Achse) interpolieren.

Kurvenzug B:

Zeitbedarf für den Austausch von Linkvariablen-/Maschinendaten- Information zwischen der Leitwert gebenden NCU und weiteren NCUs, die Folgeachsen abhängig von der Leitachse (Lead-Link-Achse) interpolieren.

Regel

Bei einer Konfiguration muss der Zeitbedarf gemäß Kurvenzug A unter dem Interpolationstakt bleiben. Wenn über die Anzahl der erforderlichen NCUs kein Freiheitsgrad mehr besteht, muss ggf. der Interpolationstakt angepasst werden.

Wenn der Interpolationstakt unverändert bleiben soll muss ggf. die Anzahl der NCUs im Lead-Link-Achsverbund reduziert werden.

2.11 Lead-Link-Achse

Begriff

Eine Lead-Link-Achse ermöglicht den lesenden Zugriff auf die achsialen Daten (Sollwert, Istwert, ...) einer Achse, die sich auf einer anderen NCU befindet.

Einführung

Für folgende Probleme bietet das Konzept der Lead-Link-Achse eine Lösung:

Bei sogenannten Taktstraßen sollen sich die einzelnen Bearbeitungs- und Handlingsstationen im Takt bzw. abhängig von einem gemeinsamen Leitwert bewegen. Die abhängigen Achsen werden dabei von einer anderen NCU interpoliert als die Leitachse.

Merkmale einer Lead-Link-Achse

Es besteht die Möglichkeit der ncu-übergreifenden Kopplung.

Es können sich mehrere NCUs auf eine Leitachse einer anderen NCU koppeln.

Der Link-Verband kann sich über bis zu 8 mit Link-Kommunikation verbundene NCUs erstrecken. Zwischen Leitachse und Folgeachsen werden Achswerte und weitere Informationen über Link-Kommunikation ausgetauscht.

Achse oder Spindel sind in der Regel gleichwertig als führende Einheit einer Achskopplung.

Bei den an der Kopplung beteiligten NCUs muss der gleiche Interpolationstakt eingestellt werden.

Einschränkungen

- Eine Achse (Leitachse), die von einer Lead-Link-Achse referenziert wird, kann nicht Link-Achse sein, d. h. von anderen NCUs als ihrer Heimat-NCU verfahren werden. S. "Link-Achsen".
- Eine Achse (Leitachse), die als Lead-Link-Achse referenziert wird, kann nicht Container-Achse sein, d. h. wechselweise von verschiedenen NCUs angesprochen werden. S. "Achscontainer".
- Eine Lead-Link-Achse darf selber keine Container-Achse sein.
- Eine Lead-Link-Achse kann nicht programmierte Führungsachse eines Gantry-Verbandes sein.
- Kopplungen mit Lead-Link-Achsen können nicht mehrstufig hintereinandergeschaltet werden (Kaskadierung).
- Achstausch ist nur innerhalb der Heimat-NCU der Lead-Link-Achse möglich. S.u.

Mit Lead-Link-Achsen werden NCU übergreifende Kopplungen ermöglicht, bei denen die Leitwertachse und die Folgeachsen auf **verschiedenen NCUs** programmiert bzw. interpoliert werden. (siehe Randbedingung).

Kopplungen

Folgende Kopplungsarten können verwendet werden:

- Leitwert (Soll-, Ist-, Simulierter-Leitwert)
- Mitschleppen
- Tangentiale Nachführung
- Elektronisches Getriebe (ELG)
- Synchronspindel

Konfiguration Leitachs-NCU

Auf der NCU, welche die Lead-Link-Achse als Leitachse interpoliert, ist die Achse eine normale lokale Achse. Sie wird mit Kanalachs- und Maschinenachs-Maschinendaten sowie im logischen Maschinenachsabbild konfiguriert. Zusätzlich sind erforderlich:

- Kennzeichnung der Maschinenachse als Lead-Link-Achse mit dem Maschinendatum:

  MD30554 $MA_AXCONF_ASSIGN_MASTER_NCU
- Nummer des Linkmoduls:

  MD12510 $MN_NCU_LINKNO
- Aktivierung der Link-Funktionalität:

  MD18780 $MN_MM_NCU_LINK_MASK = 1
- Anzahl der Linkmodule:

  MD18782 $MN_MM_LINK_NUM_OF_MODULES
- Größe des Servo-Puffers:

  MD18720 $MN_MM_SERVO_FIFO_SIZE = **4**

Konfiguration Folgeachs-NCU

Auf der NCU, welche von der nicht lokalen Leitachse (Lead-Link-Achse) Folgeachsbewegungen ableitet, sind neben den üblichen Kanal- und Maschinenachs-Maschinendaten zu konfigurieren:

- Nummer des Linkmoduls:

  MD12510 $MN_NCU_LINKNO
- Aktivierung der Link-Funktionalität:

  MD18780 $MN_MM_NCU_LINK_MASK = 1
- Anzahl der Linkmodule:

  MD18782 $MN_MM_LINK_NUM_OF_MODULES
- Evtl. MD18402 $MN_MM_NUM_CURVE_SEGMENTS
- Evtl. MD18404 $MN_MM_NUM_OF_CURVE_POLYNOMS
- Größe des Servo-Puffers:

  MD18720 $MN_MM_SERVO_FIFO_SIZE = **2**

- Im logischen Maschinenachsabbild muss die Lead-Link-Achse berücksichtigt werden mit dem Maschinendatum:

  MD AXCONF_LOGIC_MACHAX_TAB[i] = "NC_m_AX_n"

  Hierdurch wird der Bezug zu der NCU hergestellt, welche die Lead-Link-Achse interpoliert.
- In den Kanalachsen muss auch die Lead-Link-Achse konfiguriert werden.

Hinweis

Weitere Hinweise zu den Link-spezifischen Maschinendaten finden Sie in den vorangegangenen Unterkapiteln, ein Beispiel zur Konfiguration und Programmierung im Kapitel "Beispiele".

Schematischer Ablauf

Die auch im Beispiel verwendete Konstellation ist im folgenden Bild mit den wesentlichen Datenflüssen dargestellt.

NCU 1 leistet:

Interpolation und Lageregelung der Leitwertachse bzw. versorgt die Lead-Link-Achse von NCU2 mit Sollwert und Istwert.

NCU 2 leistet:

Kopplung der Lead-Link-Achse auf die Leitwertachse (von NCU1) und Ableitung einer abhängigen Bewegung einer Achse (Folgeachse).

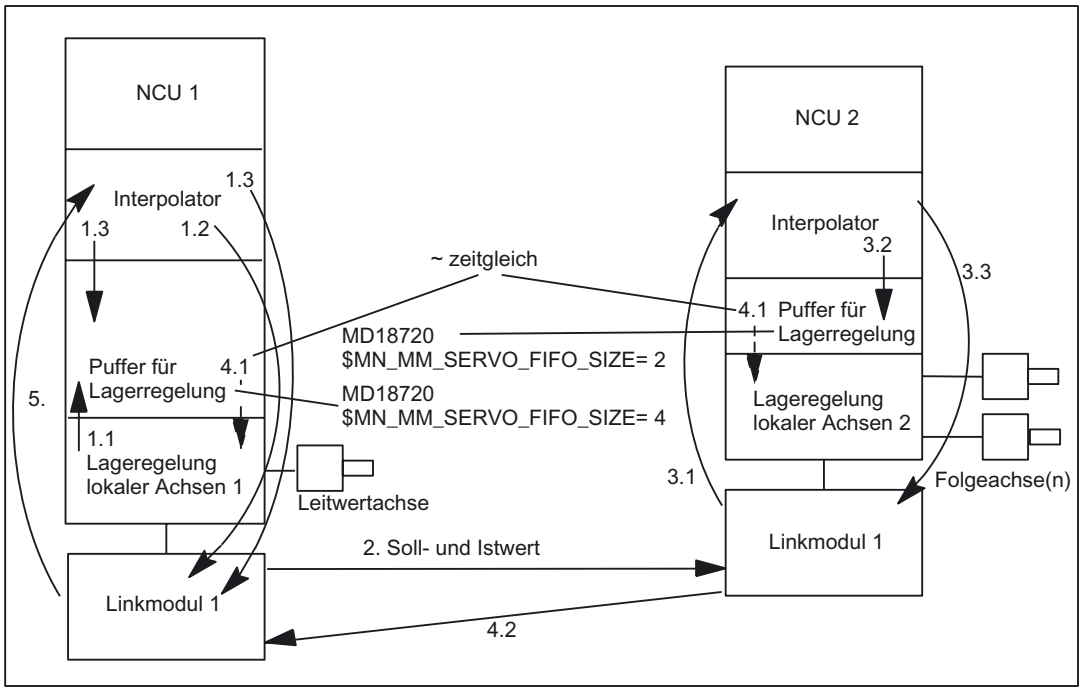


Bild 2-29 Datenflüsse für die Leitwwert-Achse, Lead-Link-Achse und abhängige Folgeachse(n)

Folgende Schritte sind dargestellt:

- 1.1 Lagereglung von NCU1 liest Istwerte der Leitwert-Achse vom Antrieb ein und schreibt sie in den Kommunikations-Puffer zur Interpolation.
- 1.2 Im selben Takt wie (1.1) werden die Istwerte vom NCU1-Interpolator auf das Link-Modul geschrieben.
- 1.3 Die Interpolation von NCU1 schreibt die erzeugten Sollwerte der Leitwert-Achse auf das Link-Modul und in den lokalen Puffer zur Lagereglung
- 2. Soll- und Istwerte werden durch das Link-Modul von NCU1 nach NCU2 übertragen.
- 3.1 Interpolation von NCU2 empfängt Sollwerte der Leitwert-Achse über das Linkmodul und berechnet die Sollwerte der Folgeachse.
- 3.2 Die Interpolation von NCU2 schreibt die erzeugten Sollwerte der Folge-Achse in den Puffer zur Lagereglung
- 3.3 Interpolator von NCU2 trägt Rückwirkungen von Folgeachse an Leitwert-Achse auf das Link-Modul.
- 4.1 Lagereglung von NCU1 empfängt Sollwert für Leitachse (Verzögerung durch: MM_SERVO_FIFO_SIZE=4). Die Lagereglung von NCU2 empfängt Sollwert für die Folgeachse.
- 4.2 Rückwirkungen werden von NCU2 nach NCU1 übertragen.
- 5. Interpolator von NCU1 empfängt Rückwirkungen und kann diese beim Berechnen der Sollwerte für die Leitwertachse mit berücksichtigen.

Transport der Lead-Link-Daten

Die Systemvariable $AA_LEAD_SP und $AA_LEAD_SV (s. unten) werden bei Änderung mit dem nichtzyklischen Dienst übertragen. Die Übertragungspriorität dieser Systemvariablen ist geringer, als die der Link-Variablen.

Zur Versorgung der Lead-Link-Achse (Soll- und Istwerte) werden ungefähr 320 Byte ausgetauscht. Die benötigte Kommunikationszeit entspricht ungefähr der einer Link-Achse.

Im Gegensatz zu einer NCU-lokalen Kopplung, werden bei einer NCU-weiten Kopplung, um eine gleichgetakte bzw. synchrone Vorgabe von Sollwerten für Leitachse bzw. Folgeachse zu erreichen, die Sollwerte der Leitachse (auf NCU 1) verzögert. (Siehe Schritt 4.1)

Der Interpolationstakt muss für alle im Lead-Link-Achsverbund stehenden NCUs gleich eingestellt sein!

Achstausch

Die Kanäle der NCU, die Folgeachsen verfährt, dürfen eine Lead-Link-Achse nicht bewegen und nicht tauschen. Die reale Leitachse kann auf ihrer Heimat-NCU getauscht werden.

Die Befehle GET, GETD und auch der Auto-Get ($MA_AUTO_GET_TYPE) werden durch den Alarm "Kanal %1 Achstausch fuer Achse %1 nicht erlaubt" abgelehnt.

2.11.1 Programmierung einer Lead-Link-Achse

Sicht Leitwert-Achse

Nur die NCU, der die Leitwert-Achse physikalisch zugeordnet ist, kann Verfahrbewegungen für diese Achse programmieren. Die Programmierung muss darüberhinaus keine Besonderheiten berücksichtigen. Durch die in den vorigen Kapiteln skizzierte Konfigurierung ist die Steuerung in der Lage, die erforderliche Sollwertverzögerung selbständig vorzunehmen, so dass Leitachse und Folgeachsen auf anderen NCUs taktsynchron interpoliert werden.

Sicht Folgeachsen

Die Programmierung auf der NCU der Folgeachsen darf keine Verfahrbefehle für die Leitwert-Achse enthalten. Verstöße gegen diese Regel lösen einen Alarm aus.

DieLeitwert-Achse wird über Kanalachs-Bezeichner in gewohnter Weise angesprochen. Die Zustände der Leitwert-Achse werden durch ausgewählte Systemvariablen zugänglich.

Systemvariablen

Folgende Systemvariablen können mit dem Kanalachsbezeichner der Leitwert-Achse benutzt werden:

$AA_LEAD_SP ; Simulierter Leitwert - Position

SAA_LEAD_SV ; Simulierter Leitwert - Geschwindigkeit

Werden diese Systemvariablen durch die NCU der Leitachse aktualisiert, so werden die neuen Werte auch an die NCUs übertragen, die Folgeachsen abhängig von dieser Leitachse verfahren wollen, übertragen.

2.12 NCU-Link bei unterschiedlichem Interpolationstakt

Problembeschreibung

In der Technik kommen Teile vor, die von einer exakt runden/zylindrischen Form geringfügig abweichen. (Beispiel: Kolben, die im Fertigungszustand unrund hergestellt werden. Sie erreichen die gewünschte annähernd runde Form erst durch die Betriebstemperatur im Einsatz). Die unrunden Formen sind z.B:

- Oval
- Kleeblatt
- Nocken
- komplexere Formen

Sollen Teile dieser Kategorie auf Drehmaschinen wirtschaftlich gefertigt werden, so kann die im Folgenden beschriebene Erweiterung zu den ab Kapitel "NCU-Link" dokumentierten Link-Funktionen benutzt werden, sofern für die Einhaltung der entsprechenden Randbedingungen gesorgt werden kann.

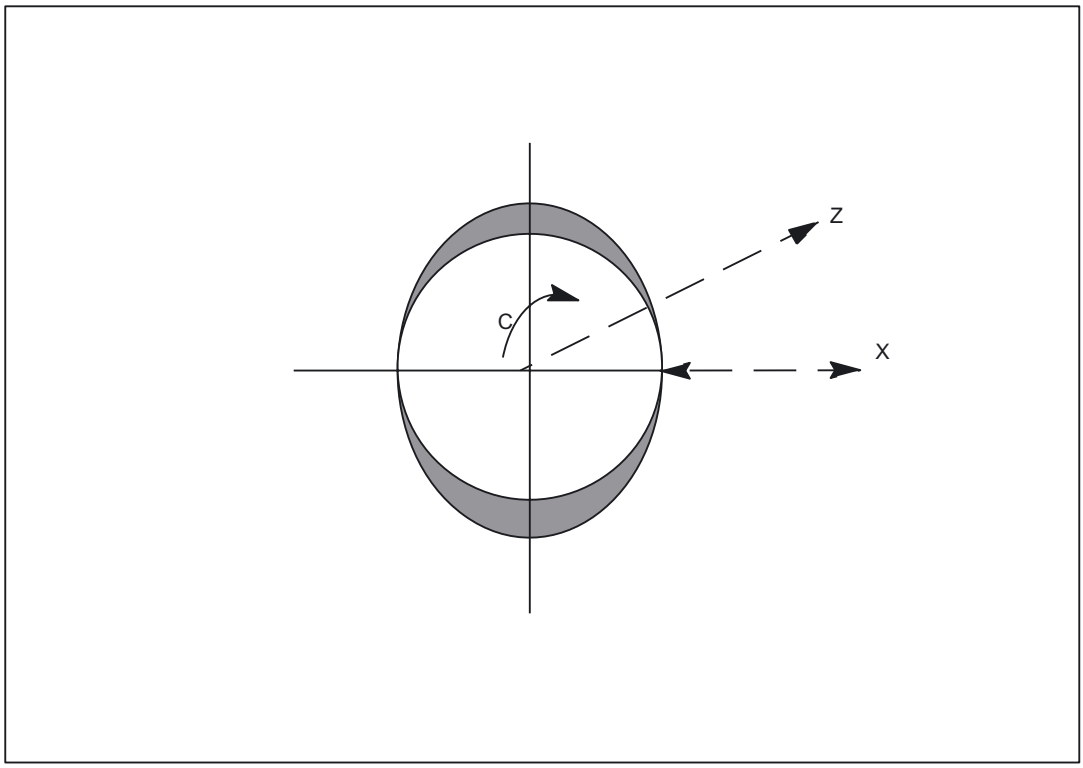


Bild 2-30 Beispiel Unrunddrehen Oval

Bewegungsabläufe

Währen das Werkstück sich um die C-Achse dreht, muss die Zustellung der X-Achse sich mit hoher Genauigkeit zwischen kleinstem und größtem Radius/Durchmesser entsprechend der gewünschten Form bewegen. (Sinus, Doppelsinus usw.). In Z-Richtung genügt in der Regel eine geringe lineare Bewegung je Umdrehung. Wenn das Werkstück sich mit üblichen Drehzahlen (z. B. 3000 Umdrehungen pro Minute) bewegen soll, muss die X-Achse hochdynamisch sein. Das bedeutet:

- kurzer Interpolationstakt
- kurzer Lageregeltakt

im Vergleich zu den Anforderungen an die C- und Z-Achse.

Lösung

Die bei den Bewegungsabläufen beschriebenen Anforderungen werden erfüllt von einem NCU-Link-Verband, wobei eine NCU mit kurzem Interpoationstakt die sehr dynamische X-Achse steuert und eine andere NCU mit standardmäßigem Interpolationstakt die weniger dynamischen Achsen C und Z steuert. Die Vorgabe der Bewegung für die X-Achse erfolgt durch Polynome in allgemeiner Form.

Hinweis

Im Zusammenhang mit Alarmmeldungen wird die Konstellation von NCUs mit unterschiedlichem Interpolationstakt, in der einer zwangsläufig der schnellere ist, "**FAST-IPO-LINK**" genannt.

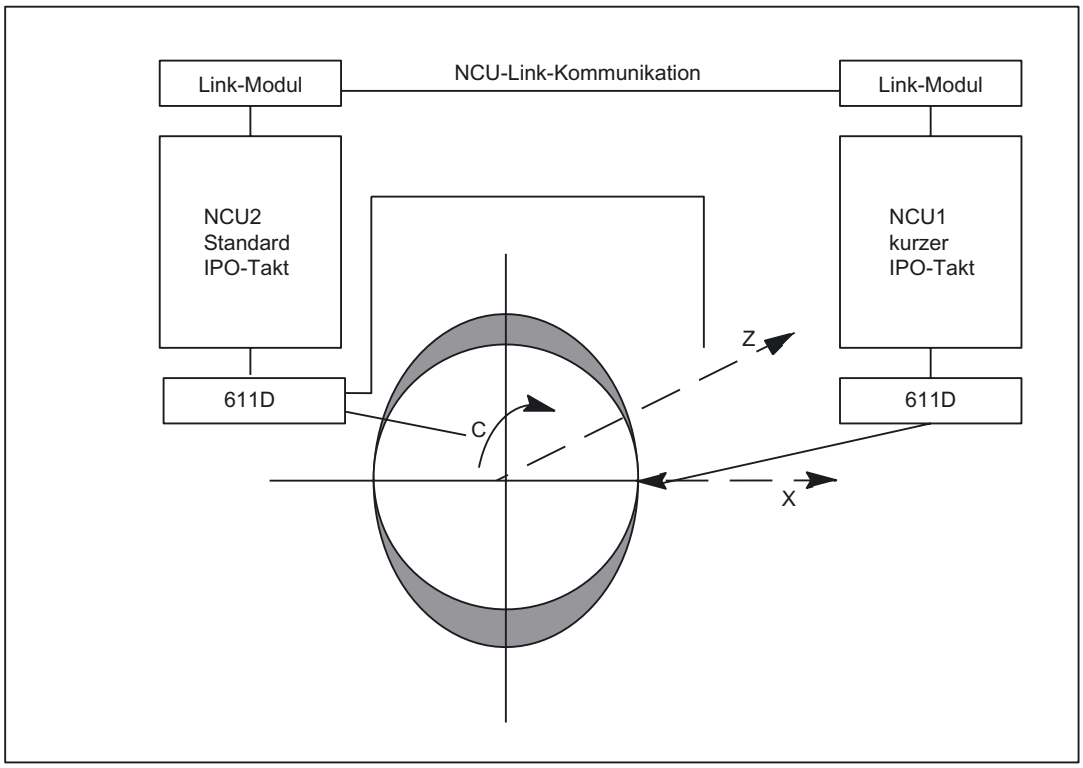


Bild 2-31 NCU-Link bei unterschiedlichem Interpolationstakt

Verallgemeinerte Lösung

Im Link-Verband von mehreren (max. 8) NCUs werden einige NCUs mit kurzem Interpolationstakt einige mit Standard-Interpolationstakt eingerichtet und die Achsen in Analogie zu obigem Bild konfiguriert. Auf diese Weise kann der Bedarf an NCUs optimiert werden. In Verbindung mit dem Konzept Achscontainer (s. "Achscontainer") wird die Unrund-Bearbeitung für Mehrspindel-Drehmaschinen möglich.

Hinweis

Für die Vorgabe der Bewegung der X-Achse existiert eine zweite Möglichkeit:

Leitwertkopplung der X-Achse an die Spindel (LEADON). Diese Möglichkeit ist weiterhin gegeben, und für Aufgaben mit geringen Dynamikanforderungen hinreichend. Sie hat aber keine spezielle Funktionserweiterung für das Unrunddrehen. Insbesondere die Vorgabe von Sollwerten über NCU-Link auf eine NCU mit schnellerem IPO-Takt wird nicht unterstützt.

Literatur: /PGA/ Programmieranleitung Arbeitsvorbereitung

2.12.1 Allgemeine Lösung schematisch

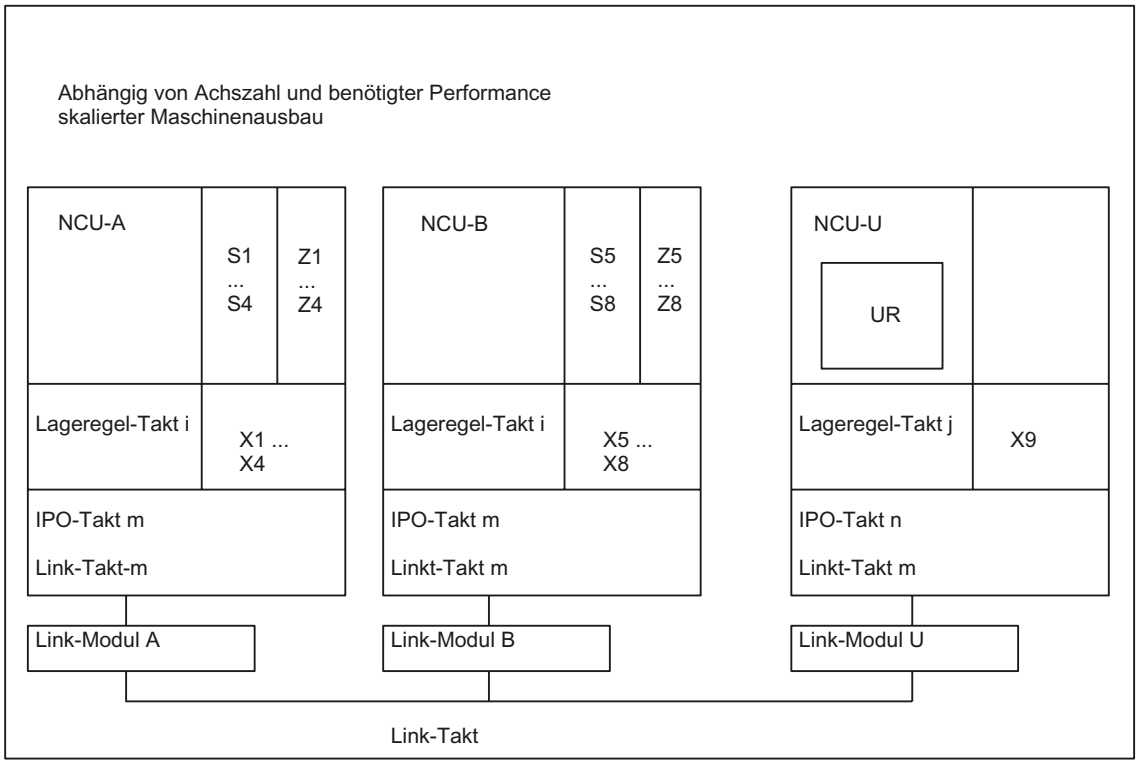


Bild 2-32 Schematisches Beispiel einer Konfiguration mit mehreren NCUs und Unrund-Einheit

Abkürzungen und Begriffe

NCU-A, NCU-B	NCUs mit Standard-Interpolationstakt
NCU-U	Unrund-NCU mit schnellem Interpolationstakt
Lageregeltakt i	"langsamer"-Lageregeltakt
Lageregeltakt j	"schneller"-Lageregeltakt
IPO-Takt m	"langsamer"-Interpolationstakt
IPO-Takt n	"schneller"-Interpolationstakt
Link-Takt m	einheitlicher Link-Takt von NCU-A bis NCU-U
S1-S8	Spindeln 1-8
Z1-Z8	Z-Achsen 1-8
X1-X8	X-Achsen für Rundbearbeitungen durch NCU-A/NCU-B
X9	X-Achse für die Unrundbearbeitung
UR	Auf NCU-U sind die Kanäle definiert, welche die Spindeln und Z-Achsen von NCU-A/NCU-B als Link-/Containerachsen betreiben und die hochdynamische x-Achse mit schnellem Interpolations- und Lageregeltakt interpolieren.

Takte

Der IPO-Takt m ist ein ganzzahliges Vielfaches von IPO-Takt n.

Der Lageregeltakt i darf größer als der schnelle IPO-Takt n sein.

Der langsame IPO-Takt m wird als Kommunikationstakt zwischen den NCU-A/NCU-B und NCU-U benutzt.

Spindelgeschwindigkeit

Die Berechnung, welche maximale Spindeldrehzahl eingehalten und programmiert werden muss, um die beteiligten Achsen (vor allem die X-Achse) nicht bei Geschwindigkeit und Beschleunigung zu überlasten, muss vorher `OFFLINE` vom Teileprogrammersteller durchgeführt werden. War dies nicht korrekt, so wird die Spindeldrehzahl an den Stellen abgesenkt, wo eine der beteiligten Achsen überlastet werden würde. Dies beeinträchtigt allerdings die Konturgenauigkeit und sollte daher vermieden werden.

Konturabweichungen

Verzögerungen oder Beschleunigungen der Spindel führen aufgrund der verschiedenen Takte zwischen NCU-A/NCU-B und NCU-U zu Abweichungen von der Sollkontur, deshalb sollten die Beschleunigungsvorgänge vor Beginn der Bearbeitung der Unrundkontur bereits beendet sein.

Achszuordnungen

Link-Achsen dürfen nur an NCUs angeschlossen sein, deren IPO-Takt gleich dem Link-Takt ist.

Beispiel: Die Achse X9 auf der schnellen NCU-U aus obigem Bild kann nicht als Link-Achse von NCU-A oder NCU-B interpoliert werden. Die Achse Z4 kann dagegen sowohl von der NCU-A, als auch der NCU-B und NCU-U interpoliert werden.

Sollwertverzögerung

Aus Sicht der NCU-U liegt bei Unrundbearbeitung eine Interpolation mit der lokalen X-Achse und den zwei Link-Achsen C und Z vor. Der Zeitverzug, der bei der Übertragung der Sollwerte an die Linkachsen entsteht, und die unterschiedliche Taktung müssen zur Erhaltung einer korrekten Kontur ausgeglichen werden durch eine Verzögerung der Sollwerte an die lokale X-Achse. Dies geschieht durch Einstellung des Maschinendatums:

MD18720 $MN_MM_SERVO_FIFO_SIZE.

Die Formel für die Belegung von MM_SERVO_FIFO_SIZE lautet:

MM_SERVO_FIFO_SIZE = 2 * Link-Takt / IPO-Takt +1

Die Formel gilt für NCU-U und NCU-A/NCU-B gemäß Bild "Schematisches Beispiel einer Konfiguration mit mehreren NCUs und Unrund-Einheit" (Kap. "Allgemeine Lösung schematisch").

Beispiele:

NCU-U:

Link-Takt = 16 ms

IPO-Takt n = 2 ms

Das Taktverhältnis ist damit 8, der Wert der Formel 2 * 8 + 1 = 17

NCU-A/NCU-B:

Linkt-Takt = 16 ms

IPO-Takt m = 16 ms

Das Taktverhältnis ist damit 1, der Wert der Formel 2 * 1 + 1 = 3

Auf fehlerhafte Einstellungen wird in einigen Fällen durch einen Alarm hingewiesen im Maschinendatum:

MD18720 $MN_MM_SERVO_FIFO_SIZE

Konsequenzen aus SERVO_FIFO_SIZE Einstellungen

- Bei allen Funktionen, die in der Interpolation zur Sollwertgenerierung mit Istwerten arbeiten, wirkt hier als effektive Totzeit der langsame Link-Takt und nicht der eigentlich sehr schnelle IPO-Takt. Dies betrifft auch das Verhalten bei Störungen (Alarme, die eine BAG stillsetzen sollen oder interpolatorisch bremsen).
- Alarmreaktionen, die direkt im Servo ausgelöst werden, weisen keine zusätzliche Verzögerung auf.

- Beschleunigungsvorgänge auf den Link-Achsen werden immer mit dem Link-Takt ausgegeben und sind damit nicht exakt synchron zu den Beschleunigungsvorgängen auf den an der schnellen NCU physikalisch angeschlossenen Achsen. Diese Konfiguration mit interpolierenden Achsen auf NCUs mit unterschiedlichem IPO-Takt ist daher nur dann sinnvoll, wenn die Link-Achsen bei der Bearbeitung nur wenig oder überhaupt nicht beschleunigt werden. Die dynamisch beanspruchten Achsen müssen also auf der schnellen NCU physikalisch angeschlossen sein.
- Beschleunigungs- und Bremsvorgänge auf den Achsen mit langsamerem Interpolationstakt sollen daher bei Bearbeitungen mit hohen Genauigkeitsanforderungen ohne Eingriff des Werkzeuges auf das Werkstück durchgeführt werden. Treten sie dennoch innerhalb der Kontur auf, so sind Konturverletzungen die Folge.
- Eine Weiterschaltung von Achscontainern mit sich **drehenden** Spindeln ist nur auf den NCUs erlaubt, deren IPO-Takt gleich dem Link-Takt ist. Wenn eine Verzögerung der Containerdrehung möglich ist, wird dies bis zum Stillstand der Spindel durchgeführt, im anderen Fall wird Alarm 4019 abgesetzt. Achscontainerumschaltungen mit **stehenden** Achsen und Spindeln sind auch auf den schnelleren NCUs zulässig. (S. "Achscontainer")

2.12.2 Unterschiedlicher Lageregeltakt

Die im Kapitel "Allgemeine Lösung schematisch" beschriebene allgemeine Lösung erlaubt auch unterschiedlichen Lageregeltakt bei NCU-A/NCU-B und NCU-U gemäß Bild "Schematisches Beispiel einer Konfiguration mit mehreren NCUs und Unrund-Einheit" (Kap. "Allgemeine Lösung schematisch").

Die im Folgenden beschriebene Parametriermöglichkeit erlaubt Qualitätsverbesserungen für bestimmte Reglerstrukturen.

Bei der Interpolation werden die unterschiedlichen Laufzeiten in der Feininterpolation und der Lageregelung aufgrund der unterschiedlichen Lageregeltakte erkannt und intern ausgeglichen durch Verzögerung der Sollwerte für die Achsen mit dem schnelleren Lageregeltakt.

Standardergebnis

Wird die Achse mit dem langsameren Lageregeltakt nicht oder nur sehr langsam beschleunigt, wird ein Konturfehler weitestgehend vermieden. Bei Beschleunigungsvorgängen wird jedoch immer ein Konturfehler auftreten. Beschleunigungen sollten daher nicht in der Bearbeitung erfolgen.

Spezielle Einstellung durch MD

In einigen Fällen kann es wünschenswert sein, die Verzögerung anpassen zu können (siehe Beispiele unten). Dazu wurde folgendes Maschinendatum eingeführt:

MD10065 $MN_POSCTRL_DESVAL_DELAY

Damit kann die Verzögerung des Sollwerts im Lageregler **für die gesamte NCU** angepasst werden. Es sind positive und negative Werte möglich. Die eingestellten Werte (Zeiten in Sekunden, max. +/- 0.1 s) werden additive zu den automatisch ermittelten Werten berücksichtigt. Siehe unten "Auskunft".

Einstellhinweise

Die notwendige Verzögerung des Lagesollwerts ist von der vorliegenden Reglerstruktur abhängig (DSC (Dynamic Stiffness Control), Vorsteuerung) und wird bei den entsprechenden Reglerumschaltungen (z. B. `FFWON` (Fahren mit Vorsteuerung ein)) berücksichtigt. Die **automatische** Einstellung sucht sich aus den Lageregeltakten der einzelnen NCUs den Maximalwert und verzögert auf allen NCUs soweit, dass mit jeder Reglerstruktur die unterschiedlichen Laufzeiten bei konstanter Geschwindigkeit ausgeglichen werden können. Damit kann auch auf der Achse mit dem langsamsten Lageregeltakt eine zusätzliche Verzögerung aktiv sein.

Die Reglerstruktur mit dem

- höchstem Kompensationsbedarf ist der Betrieb mit Momentvorsteuerung oder mit Drehzahlvorsteuerung und aktiviertem Rampengeber im Antrieb,
- geringste Kompensationsbedarf entsteht bei DSC oder deaktivierter Vorsteuerung.

Auskunft

Die Sollwertverzögerung für die 3 Reglerstrukturen

- ohne Vorsteuerung (Index 0)
- Drehzahlvorsteuerung (Index 1)
- Momentenvorsteuerung (Index 2)

wird im folgenden nur lesbaren Maschinendatum angezeigt:

MD32990 $MA_POSCTRL_DESVAL_DELAY_INFO

Durch negative Werte im Maschinendatum:

MD10065 $MN_POSCTRL_DESVAL_DELAY

werden die Werte für **alle 3** Reglerstrukturen verringert, durch positive Werte vergrößert.

Sinnvolle Umparametrierungen

Eine Umparametrierung vom Maschinendatum:

MD10065 $MN_POSCTRL_DESVAL_DELAY

ist in folgende Fällen sinnvoll:

- Es wird immer DSC aktiviert oder im Achsbetrieb immer ohne Vorsteuerung gefahren.

  In diesen Fällen ist nur auf der NCU mit dem schnellen Lagereglertakt eine zusätzliche Verzögerung notwendig. Folgendes Vorgehen ist dann sinnvoll:

  1.) MD32990 $MA_POSCTRL_DESVAL_DELAY_INFO[0] auf einer beliebigen Achse auf der NCU mit dem langsamsten Lagereglertakt auslesen.

  2.) Diese Zeit von dem aktuellen Wert im Maschinendatum:

  MD10065 $MN_POSCTRL_DESVAL_DELAY

  auf allen NCUs abziehen.

- in der Achse mit dem schnellen Lageregeltakt ergibt sich bei der hauptsächlich genutzten Reglerstruktur eine Verzögerung im Maschinendatum:

  MD32990 $MA_POSCTRL_DESVAL_DELAY_INFO,

  die kein ganzzahliges Vielfaches des Lageregeltaktes ist.

  Da gerade an die Achse mit dem schnellen Lageregeltakt hohe Anforderungen an die Dynamik bestehen, kann hier die notwendige Mittelung für die Erzeugung dieser Verzögerung bereits zu Problemen führen.

  Beispiel: Die Bearbeitung wird überwiegend mit Drehzahlvorsteuerung durchgeführt. In der Achse X aus NCU-U von Bild "Schematisches Beispiel einer Konfiguration mit mehreren NCUs und Unrund-Einheit" (Kap. "Allgemeine Lösung schematisch") wird im Maschinendatum:

  MD32990 $MA_POSCTRL_DESVAL_DELAY_INFO[1]

  ein Wert von 7 ms ausgelesen, der Lageregeltakt auf dieser NCU beträgt 2 ms.

  In diesem Fall kann es sinnvoll sein, das folgende Maschinendatum auf allen NCUs um 1 ms zu erhöhen:

  MD10065 $MN_POSCTRL_DESVAL_DELAY

  Der sich dann ergebende Wert von 8 ms in MD32990 POSCTRL_DESVAL_DELAY_INFO[1] ist ein **ganzzahliges** Vielfaches des Lageregeltaktes von 2 ms.

2.12.3 Randbedingungen

- Die Option "unterschiedlicher Interpolationstakt" kann nur im Zusammenhang mit NCU-Link (Optionen, achszahlabhängig) genutzt werden. Die verbundenen NCUs müssen alle mit den Hardware-Komponenten Link-Modul ausgestattet sein.
  - Es gelten die Konfigurationsregeln nach "Konfiguration eines Link-Verbundes".
  - Link-Achsen müssen nach "Konfiguration von Link-Achsen" konfiguriert werden.
  - Achscontainer können nach der Beschreibung unter "Achscontainer" benutzt werden.
- Link-Achsen sind nur auf NCUs zulässig, deren IPO-Takt gleich dem Link-Takt ist,

- Achscontainer
  - Die Weiterschaltung von Achscontainern mit sich drehenden Spindeln ist nur auf den NCUs erlaubt, deren IPO-Takt gleich dem Link-Takt ist.
  - Die Weiterschaltung von Achscontainern mit stehenden Achsen und Spindel ist auch auf den NCUs mit schnellerem IPO-Takt zulässig.
- Beschleunigungs- und Bremsvorgänge auf den Achsen mit langsameren Interpolations- oder Lagereglertakt führen zu Konturfehlern.

2.12.4 Aktivierung von NCU-Link mit unterschiedlichem Interpolationstakt

- Einstellung von unterschiedlichen IPO-Taktzeiten in einem Link-Verbund
- Setzen von Bit 1 im Maschinendatum:

  MD18780 $MN_MM_NCU_LINK_MASK
- Existenz der Option "NCU-Link mit unterschiedlichem Interpolationstakt"

Beispiel

Ein Beispiel Unrunddrehen unter Verwendung der Funktionalität "NCU-Link mit unterschiedlichem Interpolationstakt" finden Sie im Kapitel "Beispiele".

2.12.5 Unterschiedliche IPO-Takte, Verhalten bei Power On, RESET ...

Aufgrund der notwendigen Verzögerung der Sollwerte an den Servo tritt eine Verzögerung im Hochlauf, bei Reset, bei New Config usw. um 2 Link-Takte auf. Beim Reset tritt wegen der notwendigen Synchronisation eine zusätzliche Verzögerung von mindestens 2 Link-Takten auf.

2.12.6 Systemvariable bei unterschiedlichem Interpolationstakt

Es werden keine neuen Systemvariablen eingeführt.

Die bestehende allgemeine Systemvariable $A_LINK_TRANS_RATE zeigt auf einer NCU mit einem kleinerem IPO-Takt als dem Link-Kommunikationstakt nur im **Link-Kommunikationstakt** einen Wert ungleich 0 an. Der dann angezeigte Wert steht dem Anwender real zur Verfügung, nur in diesem Takt wird auch wirklich die Übertragung durchgeführt.

Wenn nicht alle Linkvariablen in einem Takt übertragbar sind, wird wie bisher der unterdrückbare Alarm 14764, "NCU-Link kann nicht alle Link-Variablen sofort uebertragen" angezeigt.

2.13 Maßsystem eines Link-Verbundes

Einführung

Im Link-Verbund sind NCU-übergreifende Interpolationen möglich mit:

- Link-Achsen (S. "Link-Achsen")
- Lead-Link-Achsen (S. "Lead-Link-Achse")
- Link-Achsen auf NCUs mit unterschiedlichem Interpolationstakt (S. "NCU-Link bei unterschiedlichem Interpolationstakt")

In allen Konstellationen ist ein korrektes Ergebnis nur dann möglich, wenn die mit NCU-Link verbundenen NCUs das gleiche Maßsystem verwenden.

Link-Verbund globale Einstellung

Um der Anforderung nach einheitlichem Maßsystem zu genügen, wirkt ab SW-Stand 5 die **Bedienung** Maßsystemumschaltung metrisch/inch auf den gesamten NCU-Verbund, wenn sie am Bedienfeld einer NCU des Verbundes durchgeführt wird.

Sind die Bedingungen:

- MD10260 $MN_CONVERT_SCALING_SYSTEM=1.
- Bit 0 des Maschinendatums:

  MD20110 $MC_RESET_MODE_MASK

  in jedem Kanal gesetzt
- Alle Kanäle im Reset Zustand
- Achsen nicht über JOG, DRF oder die PLC verfahren
- Konstante Scheibenumfangsgeschwindigkeit (SUG) nicht aktiv

auf allen verbundenen NCUs erfüllt, findet die Umschaltung auf allen NCUs statt. Ist auf wenigstens einer NCU eine Bedingung für Maßsystemumschaltung verletzt, so findet die Umschaltung auf keiner NCU statt.

Für die Dauer der Umschaltung werden Aktionen, wie Teileprogrammstart oder BA-Wechsel gesperrt.

Unterschiedliche Maßsysteme

Unterschiedliche Maßsysteme können nur in Zeitintervallen benutzt werden, in denen trotz installierten Link-Verbundes keine NCU-übergreifenden Interpolation stattfindet. Die erforderlichen Einstellungen müssen dann, wie in

Literatur: /FB1/ Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (G2)

beschrieben, NCU-spezifisch über G-Codes durchgeführt werden.

2.14 Randbedingungen

2.14.1 Mehrere Bedientafeln und NCUs Option Bedieneinheitenmanagement

Projektierung

Die Anzahl der projektierbaren Bedieneinheiten/NCUs ist nur durch die Verfügbarkeit der Busadressen an den einzelnen Bussegmenten der verschiedenen Bustypen begrenzt.

Die Option Bedieneinheitenmanagement ermöglicht die Kopplung von bis zu **9 Bedieneinheiten** an bis zu **9 NCU**s.

2.14.2 Mehrere Bedientafelfronten und NCUs Standardlösung

Mögliche Konfigurationen

- Konfiguration "2 Bedientafeln und 1 NCU"
  Von den Bedientafeln muss eine OP030 sein.
- Konfiguration "1 Bedientafel und bis zu 4 NCUs"
  Zusätzlich kann pro NCU lokal 1 Bedientafel betrieben werden. Beim Aufbau der Kopplung über MPI können nur NCUs eingesetzt werden, bei denen die NC-Adresse vorgegeben werden kann.

Bedienung der m:n-Verbindung

über das Kanalmenü (siehe Kapitel "Bedienoberfläche"), das über die Taste "Kanal umschalten" angewählt wird.

Voraussetzung für das Kanalmenü ist eine Projektierung über die Datei NETNAMES.INI (siehe /IAD/, Inbetriebnahmeanleitung 840D).

Die Funktion Kanalmenü ist eine Option.

Busanbindung

Adressraum: 0, ..., 31

Hinweis

Wird eine Adresse > 15 verwendet, müssen alle am Bus beteiligten Komponenten Adressen von 0 ... 31 verarbeiten können.

2.14.3 Link-Achsen

Verfügbarkeit

1. Voraussetzung ist die Vernetzung der NCUs mit Link-Modulen.
2. Die Funktion **Link-Achse** ist eine Option, die pro Linkachse (max. 32) notwendig ist.
3. Die Funktion **Achscontainer** ist eine Option, die pro Container notwendig ist.

Literatur: /PHD/ Handbuch Projektierung NCU 840D

⚠ WARNUNG
Im Zusammenhang mit Safety Integrated (SI) sind Randbedingungen zu berücksichtigen, siehe Kurzbeschreibung, Kap. NCU-Link.

2.14.4 Achscontainer

Verfügbarkeit

Achscontainer ist eine Option. Für den Fall, dass Achscontainer für Link-Achsen eingesetzt werden, gelten zusätzlich deren Randbedingungen gemäß Kapitel "Link-Achsen".

⚠ WARNUNG
Im Zusammenhang mit Safety Integrated (SI) sind Randbedingungen zu berücksichtigen, siehe Kurzbeschreibung, Kap. NCU-Link.

2.14.5 Lead-Link-Achse

Verfügbarkeit

Die NCU-übergreifende Kopplung mit Lead-Link-Achse setzt auf NCU-Link auf. Die NCU-Link Optionen müssen deshalb vorhanden sein.

⚠ WARNUNG
Im Zusammenhang mit Safety Integrated (SI) sind Randbedingungen zu berücksichtigen, siehe Kurzbeschreibung, Kap. NCU-Link.

2.14.6 NCU-Link bei unterschiedlichem Interpolationstakt

Verfügbarkeit

NCU-Link bei unterschiedlichem Interpolationstakt ist eine Option.

Es müssen alle Voraussetzungen erfüllt sein, die für Link-Achsen gelten.

Wenn die nicht lokalen Achsen aus Sicht der NCU mit dem schnellen Interpolationstakt Containerachsen sind, (z. B. bei Mehrspindel-Drehmaschinen), muss auch die entsprechende Option existieren.

2.15 Beispiele

2.15.1 Konfigurationsdatei NETNAMES.INI mit Option Bedieneinheitenmanagement

Für ein System mit vier NCUs an der BTSS wird im folgenden eine Beispiel-Konfigurationsdatei NETNAMES.INI für die Bedieneinheit MMC 1 angegeben.

Erläuterungen siehe Kapitel "Aufbau der Konfigurationsdatei".

Hinweis

Die Marginalien (fett) am linken Seitenrand dienen der Gliederung und sind nicht Bestandteil der Datei.

; NETNAMES.INI Beispiel 1 Anfang

HMI-Identifizierung

; Erkennungsteil

[own]

owner = MMC_1

PCU-NCU-Verbindungen

; Verbindungsteil

[conn MMC_1]

conn_1 = NCU_1 ; NCU 1

conn_2 = NCU_2 ; NCU 2

conn_3 = NCU_3 ; NCU 3

conn_4 = NCU_4 ; NCU 4

Bus-Identifizierung

```
[param network]
bus = BTSS ; BTSS-Bus (1,5 MBaud)
```

HMI-Beschreibung

```
[param MMC_1]

mmc_typ      = 40         ; = 0100 0000: HMI ist Server u. Hauptbedienfeld
mmc_bustyp   = BTSS       ; Bus, an dem HMI hängt
mmc_address  = 10         ; HMI-Adresse
mstt_address = 6          ; Adresse der simultan umzuschaltenden MSTT
name         = MMC_LINKS  ; Name der Bedientafel
start_mode   = ONLINE     ; HMI geht im Hochlauf in die Betriebsart Online an
                          die
                          ; DEFAULT-NCU gem. Kanaldaten s. unten
```

Beschreibung der NCU-Komponenten

```
[param NCU_1]
type = NCU_572 ; NCU-Typ
nck_address = 20 ; Adresse j der NCU-Komponente am Bus
plc_address = 20 ; Adresse p der PLC-Komponente am Bus
name = NCU1 ; Name der NCU
[param NCU_2]
type = NCU_572 ; NCU-Typ
nck_address = 21 ; Adresse j der NCU-Komponente am Bus
plc_address = 21 ; Adresse p der PLC-Komponente am Bus
name = NCU2 ; Name der NCU
[param NCU_3]
type = NCU_572 ; NCU-Typ
nck_address = 22 ; Adresse j der NCU-Komponente am Bus
plc_address = 22 ; Adresse p der PLC-Komponente am Bus
name = NCU3 ; Name der NCU
[param NCU_4]
type = NCU_572 ; NCU-Typ
nck_address = 23 ; Adresse j der NCU-Komponente am Bus
plc_address = 23 ; Adresse p der PLC-Komponente am Bus
name = NCU4 ; Name der NCU
; Ende Beschreibungsteil
```

Kanaldaten

Muster für eine Kanalmenü-Projektierung mit der Möglichkeit der M:N Zuordnung:

```
[chan MMC_1]

DEFAULT_logChanSet = G_1      ; Bei Hochlauf einzustellende Gruppe
DEFAULT_logChan = K_1_1       ; Bei Hochlauf einzustellender Kanal
ShowChanMenu = TRUE           ; Kanalmenü anzeigen
```

Liste der Kanalgruppen:

```
logChanSetList      = G_1, G_2, G_3, G_4
[G_1]
logChanList         = K_1_1, K_1_2        ; Gruppe G_1 Kanäle
[G_2]
logChanList         = K_2_1, K_2_2        ; Gruppe G_2 Kanäle
[G_3]
logChanList         = K_3_1, K_3_2        ; Gruppe G_3 Kanäle
[G_4]
logChanList         = K_4_1, K_4_2        ; Gruppe G_4 Kanäle
[K_1_1]
logNCName           = NCU_1               ; 1. Kanal der 1. Gruppe
ChanNum             = 1
[K_1_2]
logNCName           = NCU_1               ; 2. Kanal der 1. Gruppe
ChanNum             = 2
[K_2_1]
logNCName           = NCU_2               ; 1. Kanal der 2. Gruppe
ChanNum             = 1
[K_2_2]
logNCName           = NCU_2               ; 2. Kanal der 2. Gruppe
ChanNum             = 2
[K_3_1]
logNCName           = NCU_3               ; 1. Kanal der 3. Gruppe
ChanNum             = 1
[K_3_2]
logNCName           = NCU_3               ; 2. Kanal der 3. Gruppe
ChanNum             = 2
[K_4_1]
logNCName           = NCU_4               ; 1. Kanal der 4. Gruppe
ChanNum             = 1
[K_4_2]
logNCName           = NCU_4               ; 2. Kanal der 4. Gruppe
ChanNum             = 2
```

Ende der NETNAMES.INI Beispiel 1.

2.15.2 Anwenderspezifische Umgestaltung des PLC-Programms Bedieneinheitenumschaltung

Einführung

Die hier in groben Zügen beschriebene Lösung sollte nur gewählt werden, wenn wenigstens einer der folgenden Gründe/Wünsche vorliegt:

- von der Standardlösung abweichendes Verdrängungsverhalten
- von der Standardlösung abweichende Bedienmodusumschaltung
- eigenständige Behandlung des Override-Schalters im Umschaltfall
- Existenz einer 2. Maschinensteuertafel an einer MMC

Beschreibungsweise:

1. Beschreibung der Abläufe
2. Beschreibung der verfügbaren Funktionalität (Defines)
3. Graphische Darstellung der Abläufe in Diagrammform

Implementierungsdetails können darüberhinaus aus der Standardlösung, die Bestandteil der Toolbox ist, abgelesen werden.

2.15.2.1 Beschreibung der Abläufe (Übersicht)

Übersicht

PCU klopft an

Eine PCU möchte mit einer NCU Verbindung aufnehmen und teilt diesen Wunsch der PLC der betreffenden NCU mit.

PCU kommt

Eine PCU geht online an eine NCU und nimmt Verbindung mit ihr auf.

PCU geht

Eine PCU bricht die Verbindung zu einer NCU ab.

Verdrängung

Eine PCU muss die Verbindung zu einer NCU abbrechen, weil eine andere Bedientafel an diese NCU Online gehen will.

Bedienfocuswechsel im Serverbetrieb

Ein Server hält zu den ihm zugeordneten NCUs permanent Verbindung. Der Bediener kann den Bedienfocus von einer auf die andere NCU umschalten, ohne dass die bestehende Verbindung aufgelöst wird.

Aktiver/Passiver Bedienmodus

Eine Online-Bedientafel kann zwei verschiedene Bedienmodi haben:

Aktiver Modus: Bediener kann bedienen und beobachten

Passiver Modus: Bediener sieht eine Header-Information und die Kennzeichnung "passiv".

MSTT Umschaltung

Zusammen mit der Bedientafel kann optional eine ihm zugeordnete MSTT mit umgeschaltet werden.

2.15.2.2 Beschreibung der Abläufe (Details)

Einführung

Bei der Beschreibung der Abläufe werden Bezeichner für festgelegte logische Funktionen benutzt (Beispiel: `OFFL_REQ_OP/OK`), deren programmiertechnische Anwendung weiter hinten im Kapitel beschrieben werden. Die Codierung der Funktionen erfolgt gemäß Kapitel "Festgelegte logische Funktionen/Defines". Die Funktionen hinterlegen Werte in der Nahtstelle, die von PLC und von HMI aus angesprochen werden können. Eine Bedientafel benutzt, während sie sich um die Nutzung einer Online-Schnittstelle bewirbt, die Anklopfschnittstelle. Bedientafeln, die bereits eine Verbindung mit einer NCU halten, benutzen eine der zwei vorhandenen Online-Schnittstellen der Nahtstelle. Die Details der Nahtstelle sind im Kapitel "Signalbeschreibungen" und in

Literatur: Listen

beschrieben.

Um die Gesamtabläufe zu verdeutlichen, werden sowohl die nicht beeinflussbaren HMI-Aktivitäten als auch die **modifizierbaren PLC-Aktivitäten** beschrieben.

Bedientafel klopft an

Falls die Bedientafel bereits an einer NCU online ist (Online-NCU) und Kontakt mit einer anderen NCU aufnehmen möchte (Ziel-NCU), muss sie der PLC der Online-NCU zuerst mitteilen, dass sie auf die Ziel-NCU umschalten will.

Er stellt den Offline-Request `OFFL_REQ_OP/OK` an die Online-PLC.

`OFFL_CONF_OP/OK`:

Online-PLC hat den Offline-Request erhalten. HMI kann anschließend an der Ziel-PLC anklopfen.

OFFL_CONF_OP/PLC_LOCKED

Online-PLC hat den Offline-Request erhalten. Die Bedientafel-Umschaltung ist in der HMI-PLC-Schnittstelle gesperrt. Die Bedientafel kann keine Verbindung mit einer anderen NCU aufnehmen und muss online bleiben.

Nach der positiven Quittung OFFL_CONF_OP/OK teilt die Bedientafel ihren Online-Wunsch, mittels *Client-Identifikation*, der Ziel-PLC der betreffenden NCU mit.

Client-Identifikation: Eindeutige HMI-Kennung, die aus Bustyp und Busadresse besteht. (ONL_REQUEST DB19, DBW100)

Die Ziel-PLC gibt der Bedientafel eine positive oder negative Quittung:

pos. Quittung: Ziel-PLC gibt der Bedientafel seine Client-Identifikation zurück. (ONL_CONFIRM, DB19, DBW102), Bedientafel belegt die Anklopf-Schnittstelle mit seinen Parametern. (Client-Ident, MMC-Typ, MSTT-Adresse). Bedientafel kann, nachdem er die Online-Permission durch die Ziel-PLC erhalten hat, online gehen.

neg. Quittung: Ziel-PLC gibt der Bedientafel seine Client-Identifikation nicht zurück. (ONL_CONFIRM, DB19, DBW102 nicht gleich der eigenen Client-Identifikation). Bedientafel kann nicht online gehen.

Beispiel:

Eine andere Bedientafel befindet sich gerade in der Umschaltphase auf diese NCU. Diese Phase darf nicht unterbrochen werden. Die Bedientafel bleibt online an der Online-NCU.

Nachdem die Ziel-PLC die Bedientafel positiv quittiert hat, muss sie u. U. eine Online-Bedientafel verdrängen. Anschließend gibt die PLC der Bedientafel die positive/negative Online-Permission.

Positiv:

ONL_PERM/OK

Nach Erhalt der positiven Online-Permission (DB19, DBB 108, 109) kann die Bedientafel online gehen. Zusammen mit der Online-Permission wird der Bedientafel seine HMI-PLC-Schnittstelle zugewiesen. (1 oder 2, Details verdeutlicht die Nahtstellenbeschreibung im Kapitel "Signalbeschreibungen").

Negativ:

ONL_PERM/MMC_LOCKED

Die anklopfende Bedientafel kann nicht Online gehen. An dieser NCU sind zwei Bedientafeln online, auf denen nicht unterbrechbare Prozesse laufen. Die PLC kann keine der beiden Bedientafen verdrängen.

ONL_PERM/PLC_LOCKED

Die anklopfende Bedientafel kann nicht Online gehen. Die Bedientafel-Umschaltung ist in der HMI-PLC-Schnittstelle gesperrt.

ONL_PERM/PRIO_H

Die anklopfende Bedientafel kann nicht Online gehen. An dieser NCU sind zwei Bedientafeln online, die höherprior sind als die anklopfende Bedientafel. Die PLC kann keine der beiden Bedientafeln verdrängen.

Bedientafel kommt

Nachdem die Bedientafel an der Ziel-PLC angeklopft und von der PLC die Online-Permission erhalten hat, kann sie sich mit der Ziel-NCU verbinden.

Die Bedientafel geht online und gibt der PLC mit (Station-active) `S_ACT/CONNECT` Mitteilung, dass sie Verbindung mit der NCU aufgenommen hat.

Die Bedientafel richtet ihr Lebenszeichen gemäß der ihm zugewiesenen Schnittstelle ein.

Anschließend fordert die Bedientafel

- bei Bedientafelfront: den aktiven Bedienmodus auf der Ziel-NCU an,
- bei Server: den Bedienfocus auf der Ziel-NCU an.

Die PLC schaltet anschließend für diese Bedientafel die HMI-Lebenszeichenüberwachung ein.

siehe: Aktiver/Passiver Bedienmodus

siehe: Bedienfocuswechsel im Serverbetrieb

Bedientafel geht

Eine Bedientafel bricht die Kommunikation mit einer NCU ab.

Dabei sind zwei Fälle zu unterscheiden:

1. Der Bediener möchte die Bedientafel auf eine andere NCU umschalten. Die Bedientafel hat an der Ziel-PLC angeklopft und die positive Online-Permission (`ONL_PERM/OK`) erhalten. Vorher hat sie der Online-PLC seinen Umschaltwunsch mit `OFFL_REQ_OP/OK` mitgeteilt und die positive Quittung erhalten (`OFFL_CONF_OP/OK`). Durch den Umschaltvorgang auf die Ziel-NCU wird das HMI-Lebenszeichen in der Online-PLC von TRUE auf FALSE gesetzt. Die fallende Flanke in Verbindung mit der vorher beschriebenen Sequenz ist für die Online-PLC das Signal, dass die Bedientafel die Verbindung zur NCU gelöst hat. Eine eventuell der Bedientafel zugeordnete und aktivierte MSTT wird jetzt von der PLC deaktiviert. Für die gegangene Bedientafel wird in der PLC der passive Bedienmodus eingestellt.

   siehe: Aktiver/Passiver Bedienmodus

2. Die Bedientafel wird von der PLC durch die Online-Anforderung einer anderen Bedientafel verdrängt. Siehe Verdrängung.

Verdrängung

Zwei Bedientafeln sind online an einer NCU, jeder belegt eine HMI-PLC-Schnittstelle und eine dritte Bedientafel möchte online gehen.

Die PLC muss nach einer festgelegten Strategie eine der beiden Bedientafeln verdrängen. Sie fordert mit dem Offline-Request (`OFFL_REQ_OP/OK`) die zu verdrängende Bedientafel auf, die Kommunikation mit der NCU abzubrechen. Die Bedientafel gibt der PLC eine positive oder negative Quittung:

Positiv:

`OFFL_CONF_PLC/OK`

Bedientafel bricht die Verbindung zu dieser NCU ab und geht in den Zustand Offline.

Das HMI-Lebenszeichen in der PLC wechselt von TRUE auf FALSE.

Die fallende Flanke in Verbindung mit der vorher beschriebenen Sequenz ist für die Online-PLC das Signal, dass die Bedientafel die Verbindung zur NCU gelöst hat. Eine eventuell der Bedientafel zugeordnete und aktivierte MSTT muss jetzt von der PLC deaktiviert werden.

Die HMI-Lebenszeichenüberwachung wird von der PLC ebenfalls eingestellt.

Für diese Bedientafel wird in der PLC der passive Bedienmodus eingestellt.

siehe "Aktiver/Passiver Bedienmodus" weiter unten.

Negativ: OFFL_CONF_PLC/MMC_LOCKED

Auf der Bedientafel laufen Prozesse ab, die nicht unterbrochen werden dürfen (z. B. Betrieb über V24, Datenübertragung zwischen NCU und PCU).

Die Bedientafel bleibt online an dieser NCU.

Bedienfocuswechsel im Serverbetrieb

Ein Server hält zu den ihm zugeordneten NCUs permanent Verbindung. Der Bediener kann den Bedienfocus von einer auf die andere NCU umschalten, ohne dass die bestehende Verbindung aufgelöst wird.

Falls vom Bediener der Bedienfocus auf eine andere NCU umgeschaltet werden soll, muss in der Focus-PLC und in der Ziel-PLC zuerst nachgefragt werden, ob die PLCs eine Umschaltung erlauben.

Die Bedientafel stellt zuerst den Offline-Request-Focus (OFFL_REQ_FOC/OK) an die Focus-PLC.

Nach positiver Quitung (OFFL_CONF_FOC/OK) durch die Focus-PLC stellt die Bedientafel mit ONL_REQ_FOC/OK die Anfrage auf Focuswechsel in der Ziel-PLC.

Nachdem die Bedientafel von der Ziel-PLC die positive Permission auf den Focuswechsel erhalten hat (ONL_PERM_FOC, OK), meldet sich die Bedientafel mit S_ACT/DISC_FOCUS in der Focus-PLC ab und wechselt den Focus auf die Ziel-PLC.

In der Ziel-NCU muss die Bedientafel noch anschließend den aktiven Bedienmodus anfordern. Die ehemalige Focus-PLC muss nach Erhalt von S_ACT/DISC_FOCUS für diese HMI-PLC-Schnittstelle den passiven Bedienmodus einstellen und eine eventuell für diese Bedientafel aktivierte MSTT deaktivieren.

siehe: Aktiver/Passiver Bedienmodus

Aktiver/Passiver Bedienmodus

Nachdem eine Bedientafel an eine NCU online gegangen ist, kann sie zwei verschiedene Bedienmodi einnehmen:

Aktiv-Modus: Bediener kann bedienen und beobachten

Passiv-Modus: der Bediener sieht Header-Information und Statushinweis "passiv".

Nach Umschaltung auf eine NCU fordert sie in der Online PLC zuerst den aktiven Bedienmodus an.

Falls zwei Bedientafeln zu einer Zeit an einer NCU online sind, ist eine der beiden immer im aktiven, die andere im passiven Bedienmodus.

Der Bediener kann den aktiven Bedienmodus an der passiven Bedientafel per Tastendruck anfordern.

Falls zu den Online-Bedientafeln eine MSTT projektiert worden ist, wird die MSTT der aktiven Bedientafel aktiviert. Die MSTT der passiven Bedientafel ist deaktiviert, es gibt also an einer NCU nur jeweils eine aktive MSTT.

Für jeden der beiden Online-Bedientafeln gibt es in der HMI-PLC-Schnittstelle vier Signale, über die der Bedienmodus-Wechsel von der PLC gesteuert wird.

Tabelle 2- 6 Signale (x = 1, 2: 1. oder 2. HMI-PLC-Schnittstelle)

HMI-PLC-Schnittstelle	Wert	Bedeutung
MMCx_ACTIVE_REQ	FALSE → TRUE	HMI an PLC: passive Bedientafel fordert aktiven Bedienmodus an
	TRUE → FALSE	PLC an HMI: Anforderung erhalten
MMCx_ACTIVE_PERM	FALSE → TRUE	PLC an HMI: passive Bedientafel kann in aktiven Bedienmodus wechseln
	TRUE → FALSE	PLC an HMI: aktive Bedientafel soll in passiven Bedienmodus wechseln
MMCx_ACTIVE_CHANGED	FALSE → TRUE	HMI an PLC: Bedientafel hat Wechsel von Passiv- nach Aktiv-Modus vollzogen
	TRUE → FALSE	HMI an PLC: Bedientafel hat Wechsel von Aktiv- nach Passiv-Modus vollzogen
MMCx_CHANGE_DENIED	FALSE → TRUE	HMI an PLC oder PLC an HMI, je nach Schnittstelle: Bedienmoduswechsel nicht möglich, nicht unterbrechbare Prozesse auf aktiver Bedientafel.
	TRUE → FALSE	HMI an PLC oder PLC an HMI, je nach Schnittstelle: Quittung auf MMCx_CHANGE_DENIED(FALSE->TRUE)

Die Umschaltung der Bedienmodi ist in der nachfolgenden Sequenz beispielhaft beschrieben.

Zwei Bedientafeln an einer NCU online, MMC_1 im aktiven Bedienmodus, MMC_2 im passiven Bedienmodus, Bediener fordert an MMC_2 den aktiven Bedienmodus an.

Signalzustand für diesen Fall:

MMC_1	WERT	MMC_2	Wert
MMC1_ACTIVE_REQ	FALSE	MMC2_ACTIVE_REQ	FALSE
MMC1_ACTIVE_PERM	TRUE	MMC2_ACTIVE_PERM	FALSE
MMC1_ACTIVE_CHANGED	TRUE	MMC2_ACTIVE_CHANGED	FALSE
MMC1_CHANGE_DENIED	FALSE	MMC2_CHANGE_DENIED	FALSE

MMC_2 fordert den aktiven Bedienmodus und setzt MMC_2_AKTIVE_REQ = TRUE.

Die PLC quittiert die Anforderung von MMC_2 mit MMC_2_ACTIVE_REQ = FALSE.

Anschließend fordert sie MMC_1 mit MMC1_ACTIVE_PERM = FALSE auf, in den Passiven Bedienmodus zu wechseln.

Dabei sind zwei Fälle zu unterscheiden:

1. MMC_1 kann in den passiven Bedienmodus wechseln:

   MMC_1 wechselt vom aktiven in den passiven Bedienmodus und quittiert den Wechsel mit

   MMC1_ACTIVE_CHANGED = FALSE.

   Eine dem MMC evtl. zugeordnete und aktivierte MSTT wird jetzt von der PLC deaktiviert.

   Die PLC gibt mit MMC2_ACTIVE_PERM = TRUE dem MMC_2 die Erlaubnis in den aktiven Bedienmodus zu wechseln.

   MMC_2 wechselt den Zustand und quittiert mit MMC2_ACTIVE_CHANGED = TRUE. Falls MMC_2 eine MSTT zugeordnet wurde, wird sie jetzt von der PLC aktiviert.

2. MMC_1 kann nicht in den passiven Bedienmodus wechseln (auf MMC_1 laufen Prozesse, die einen Wechsel nicht zulassen):

   MMC_1 setzt MMC1_CHANGE_DENIED = TRUE, Zustandswechsel kann nicht vollzogen werden.

   Die PLC quittiert mit MMC1_CHANGE_DENIED = FALSE und gibt mit MMC1_ACTIVE_PERM = TRUE dem MMC_1 die Erlaubnis im Aktiv-Modus zu bleiben. Mit MMC2_CHANGE_DENIED = TRUE, teilt sie MMC_2 mit, dass MMC_1 nicht in den passiven Bedienmodus wechseln kann.

   MMC_2 quittiert anschließend mit MMC2_CHANGE_DENIED = FALSE und bleibt im passiven Bedienmodus.

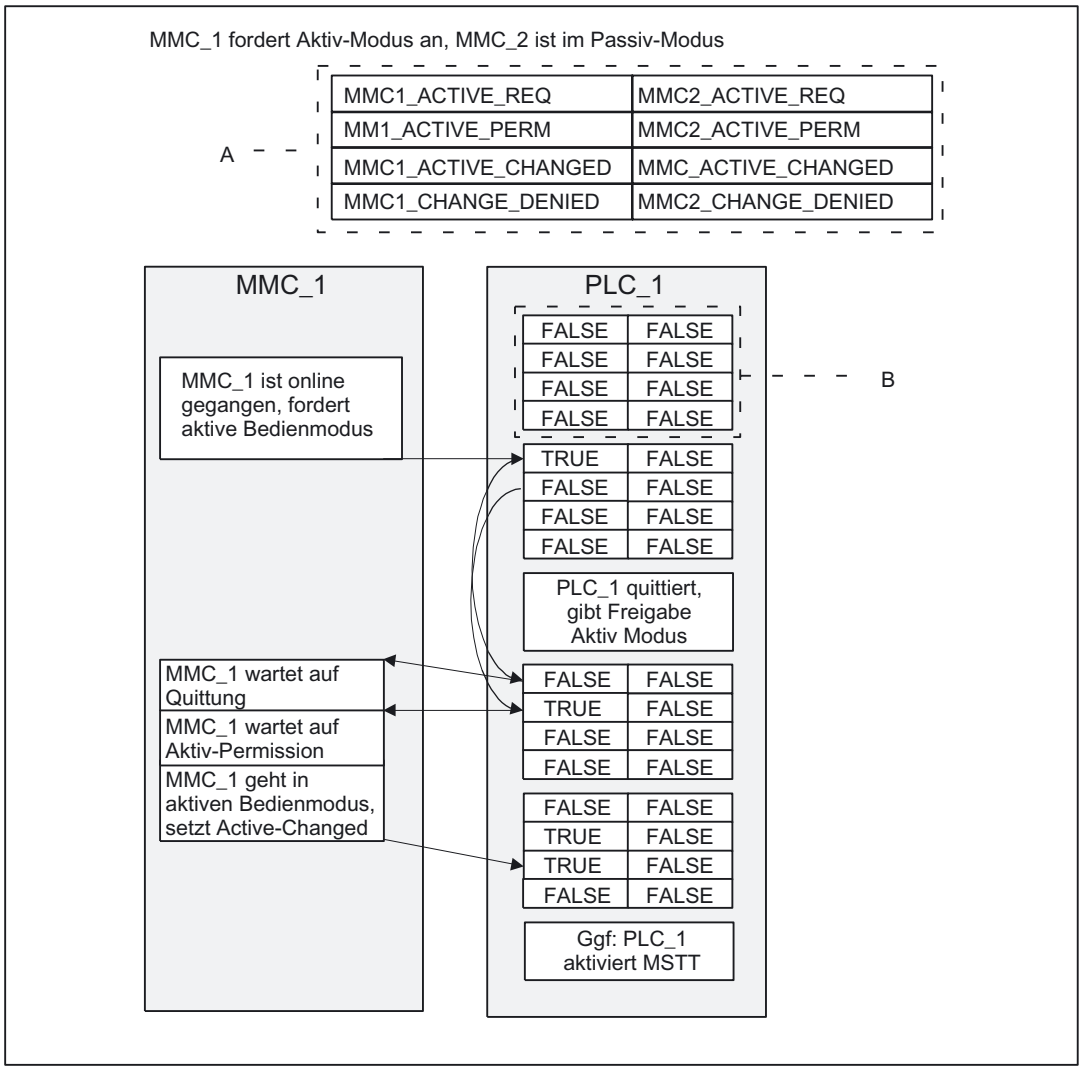


Bild 2-33 MMC_1 fordert Aktiv-Modus an, MMC_2 ist im Passiv-Modus

Lesehinweis

Die Anordnung der Signale eines Blockes im Kasten PLC_x (markiert als B) entspricht der Anordnung der Signalnamen im Kopfteil (markiert als A). Die Blöcke B wiederholen sich im Kasten PLC_x mit dem Verlauf der Zeit von oben nach unten.

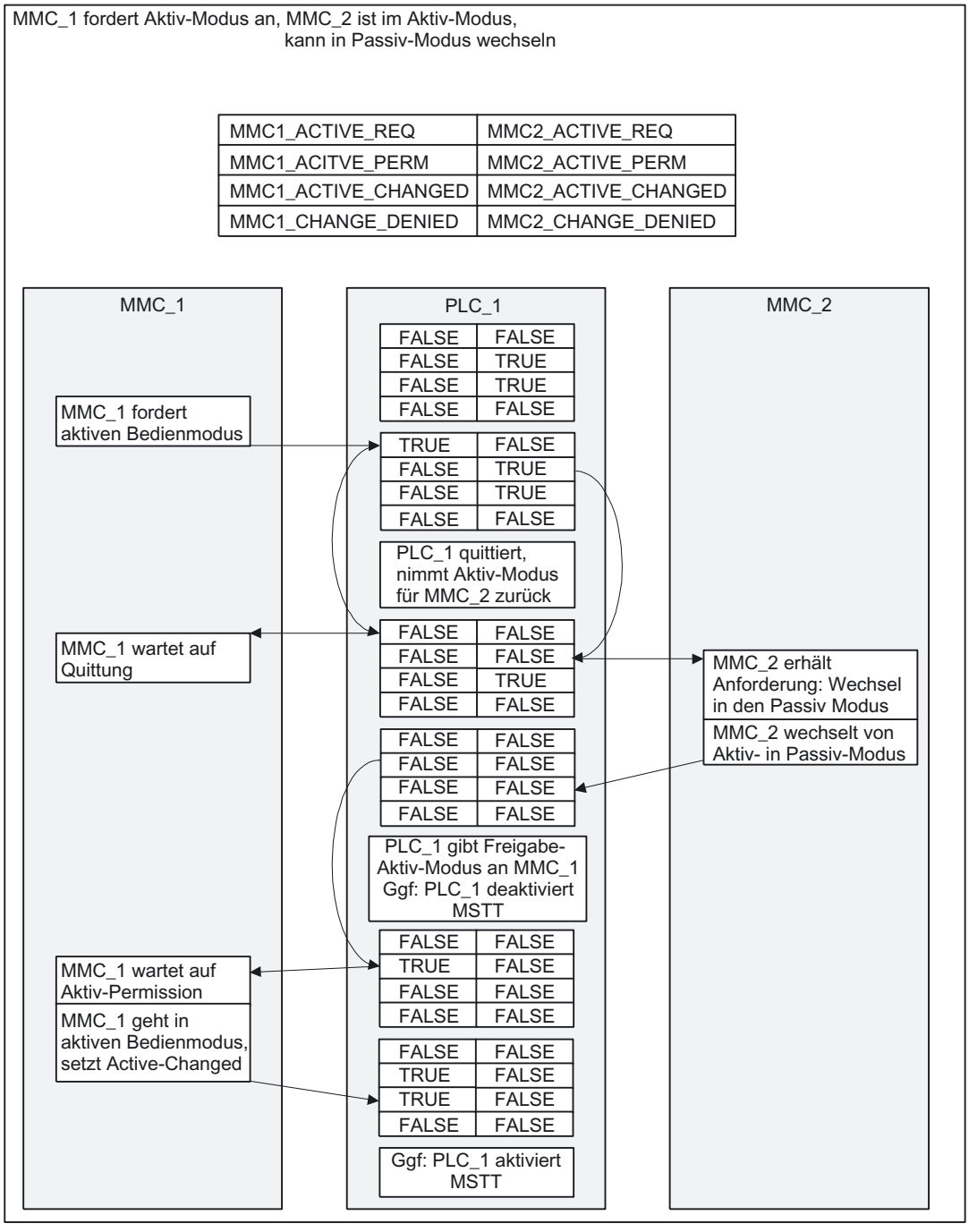


Bild 2-34 MMC_1 fordert Aktiv-Modus an, MMC_2 ist im Aktiv-Modus, kann in Passiv-Modus wechseln

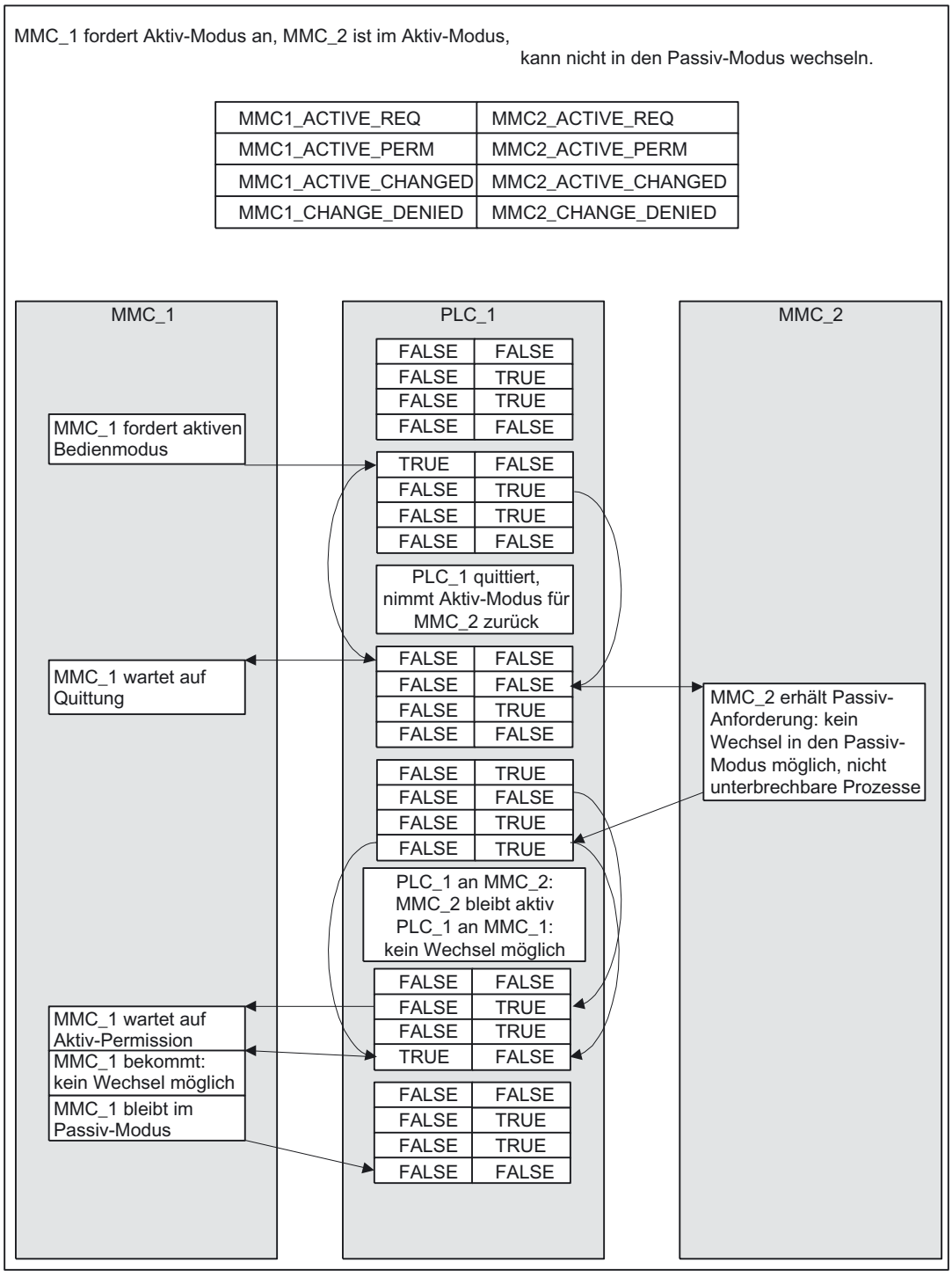


Bild 2-35 MMC_1 fordert Aktiv-Modus an, MMC_2 ist im Aktiv-Modus, kann nicht in den Passiv-Modus wechseln

MSTT_UMSCHALTUNG

Eine Bedieneinheit besteht aus Bedientafel und MSTT, beide können als Einheit zusammen umgeschaltet werden.

Falls zur Bedientafel in der Projektierungsdatei NETNAMES.INI eine MSTT projektiert wurde, wird sie zusammen mit der Bedientafel aktiviert bzw. deaktiviert.

Die MSTT derjenigen Bedientafel, die sich im aktiven Bedienmodus befindet, ist aktiviert.

An einer NCU ist somit immer nur **eine** MSTT aktiv.

Die MSTT wird von der PLC aktiviert:

- Bedientafel wechselt in den aktiven Bedienmodus. (Signal MMCx_ACTIVE_CHANGED: FALSE -> TRUE, x = 1,2 erste oder zweite HMI-PLC-Schnittstelle)

Die MSTT wird von der PLC deaktiviert

- Bedientafel wechselt in den passiven Bedienmodus

  (Signal MMCx_ACTIVE_CHANGED: TRUE -> FALSE, x = 1,2 erste oder zweite HMI-PLC-Schnittstelle)

- Bedientafel geht offline durch Umschaltung oder Verdrängung

  Das HMI-Lebenszeichen wechselt von TRUE auf FALSE, wenn eine Bedientafel offline geht. Die PLC deaktiviert nach dem Flankenwechsel die zugeordnete MSTT.

- Server-HMI löst Bedienfocus von dieser NCU und schaltet auf eine andere NCU um. Als letztes Signal sendet der Server S_ACT/DIS_FOCUS an seine eigene HMI-PLC-Schnittstelle. Die PLC deaktiviert anschließend die entsprechende MSTT.

2.15.2.3 Festgelegte logische Funktionen/Defines

Hinweis

Die zulässigen Werte für Bustyp, Funktionen/Status sowie Zusatzinformationen und die zulässigen Kombinationen von Status und Zusatzinformationen finden Sie unter "Festgelegte logische Funktionen/Defines". In den folgenden Diagrammen sind die logischen Bezeichner der Funktionen verwendet.

2.15.2.4 Graphiken der Funktionssequenzen

Übersicht

Die nachfolgenden Bilder beschreiben die Umschaltung einer Bedienstation (Umschaltung von NCU_1 auf NCU_2).

Die ersten fünf Bilder beschreiben das Umschalten als Bedienstation, die drei danach folgenden Bilder beschreiben das Umschalten als Server.

Falls eine Bedientafel (MMC) im Zustand Offline an eine NCU Online gehen möchte (z. B. Hochlauf) entfällt die Sequenz OFFL_REQ_OP (...), bzw OFFL_CONF_OP(...).

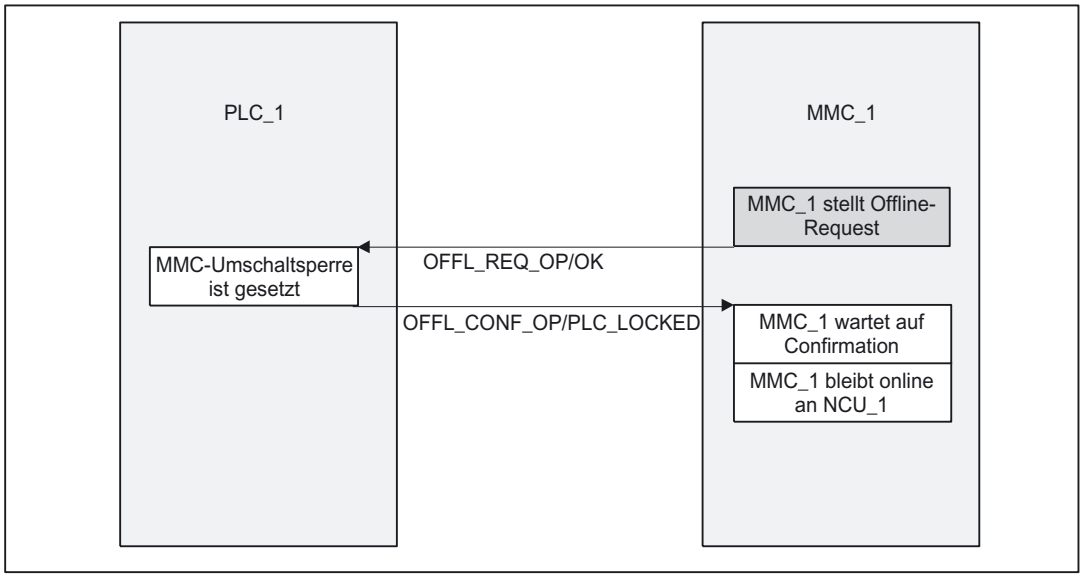


Bild 2-36 MMC_1 ist online an NCU_1 und möchte auf NCU_2 umschalten, in PLC_1 besteht Umschaltsperre

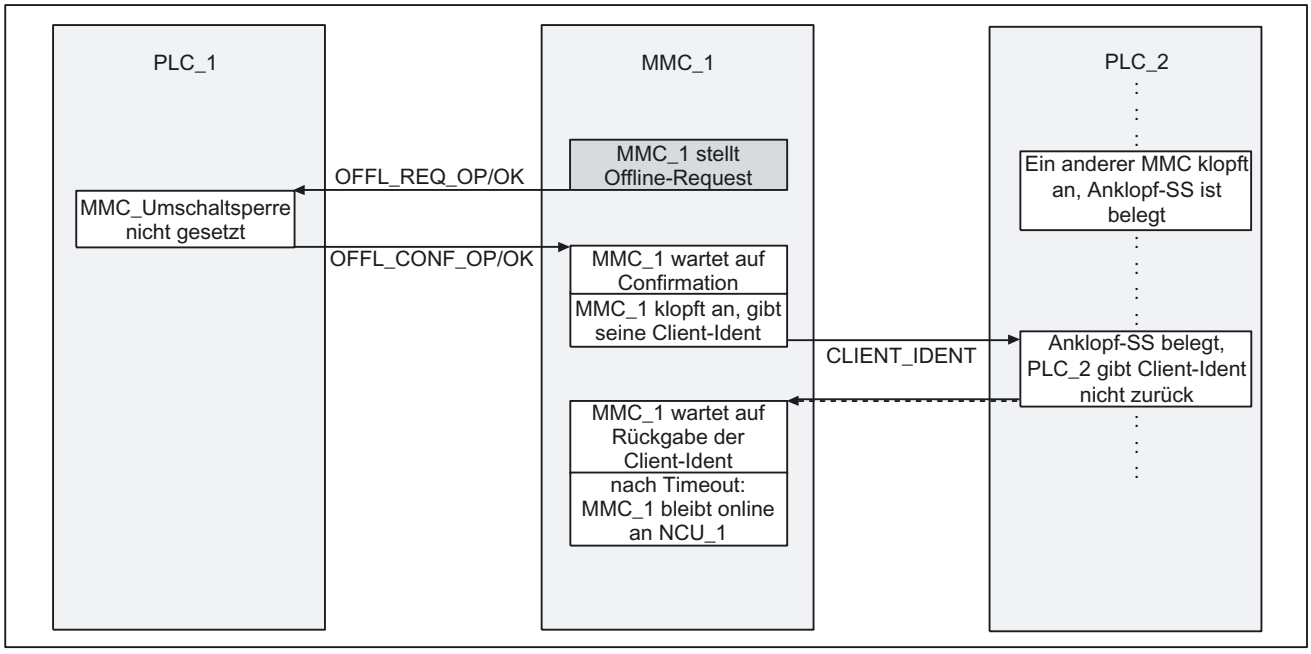


Bild 2-37 MMC_1 online an NCU_1, MMC_1 möchte auf NCU_2 umschalten, Anklopf-Schnittstelle in PLC_2 von anderen MMC belegt

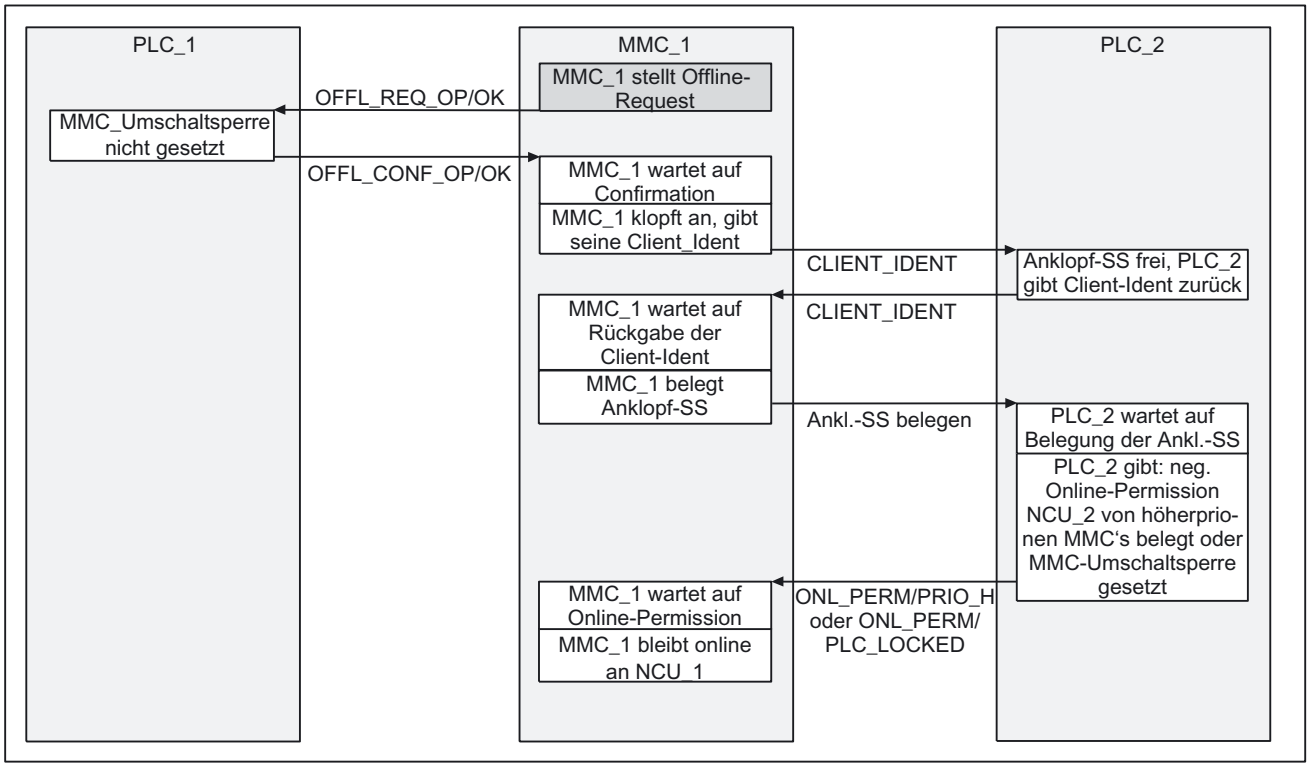


Bild 2-38 MMC_1 online an NCU_1, MMC_1 möchte auf NCU_2 umschalten, bekommt von PLC_2 negative Permission

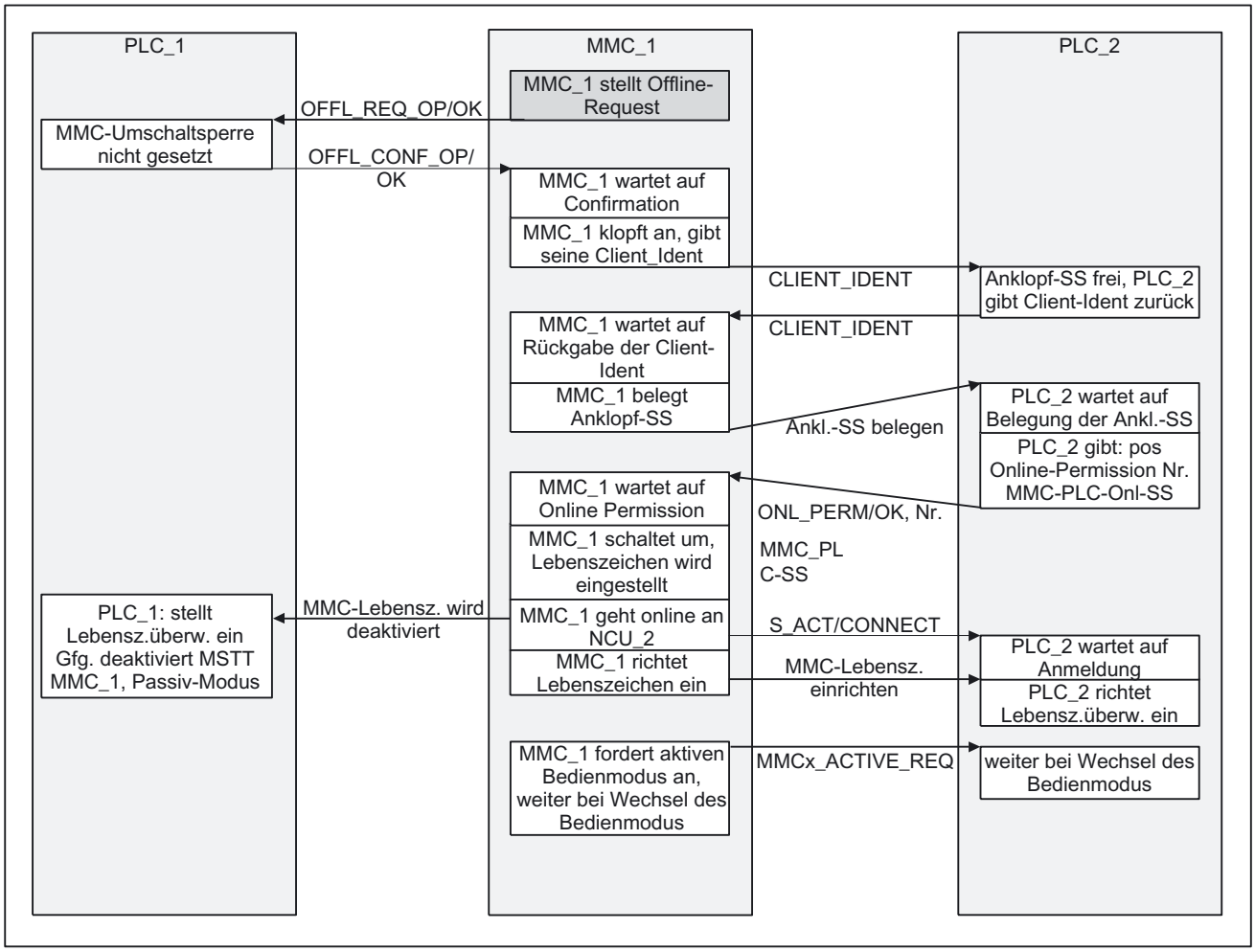


Bild 2-39 MMC_1 online an NCU_1, MMC_1 schaltet auf NCU_2 um (keine Verdrängung)

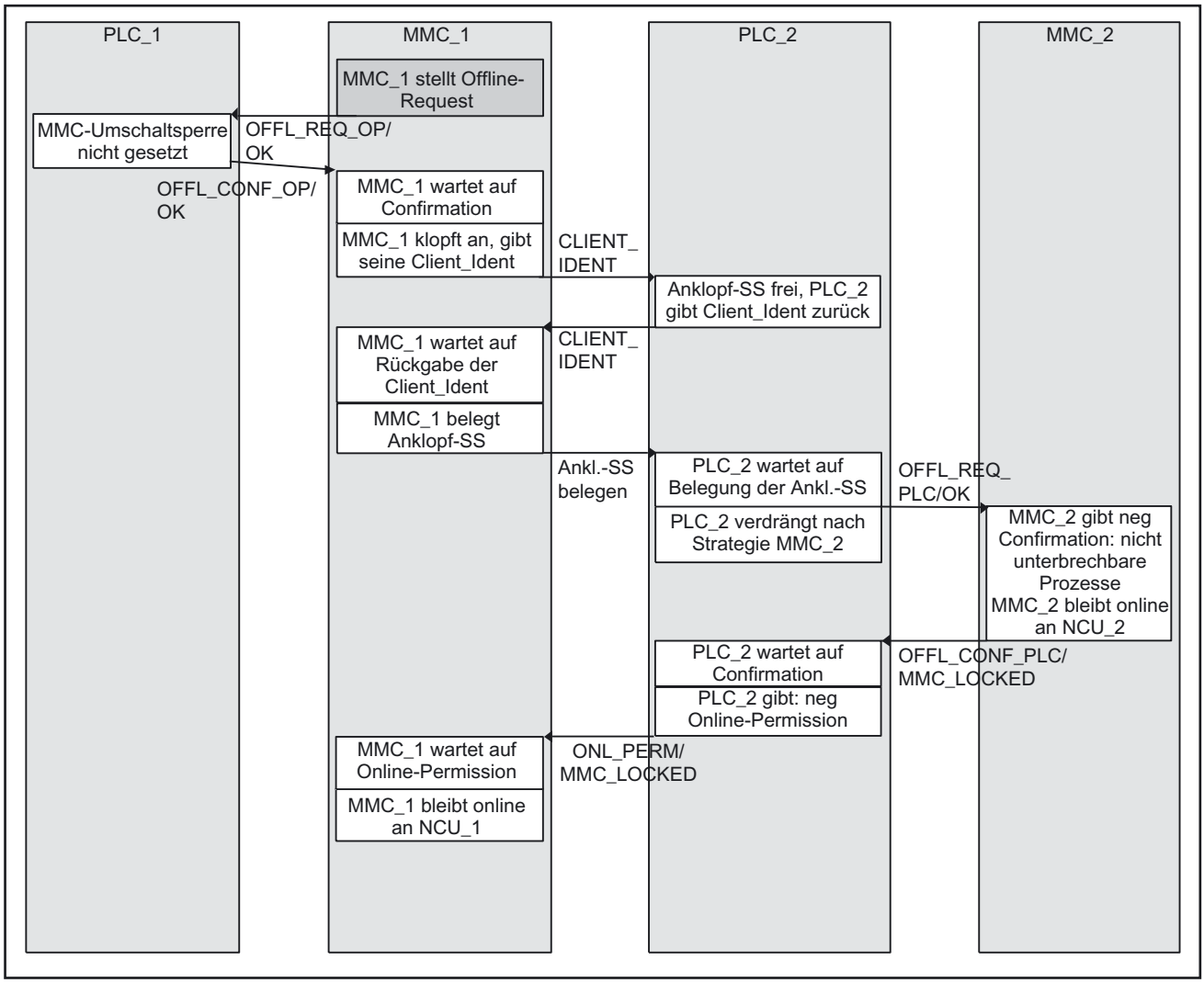


Bild 2-40 MMC_1 online an NCU_1, MMC_2 online an NCU_2, MMC_1 möchte auf NCU_2 schalten, an NCU_2 sind MMCs mit nicht unterbrechbaren Prozessen online

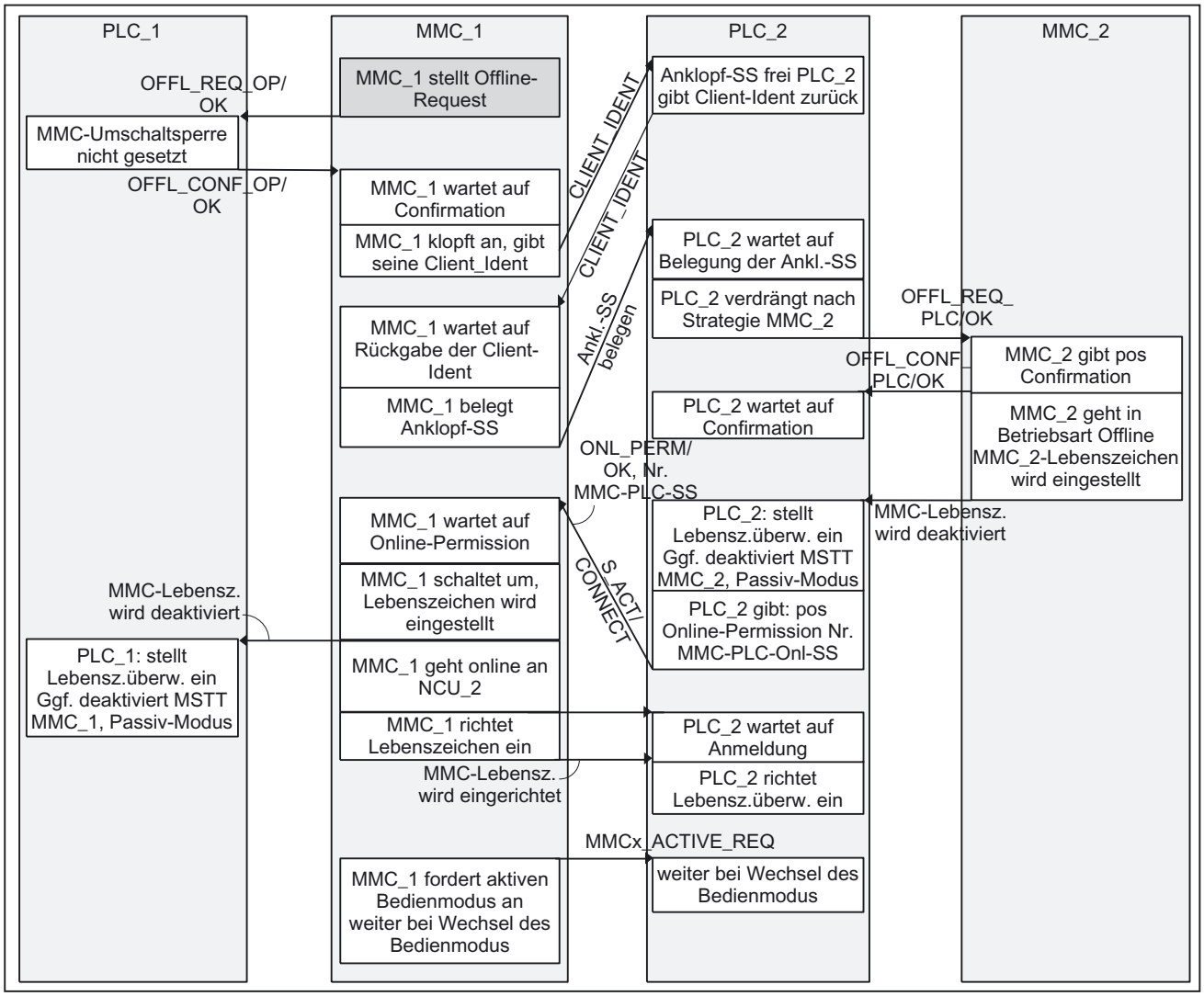


Bild 2-41 MMC_1 online an NCU_1, MMC_2 online an NCU_2, MMC_1 schaltet von NCU_1 auf NCU_2, MMC_2 wird verdrängt

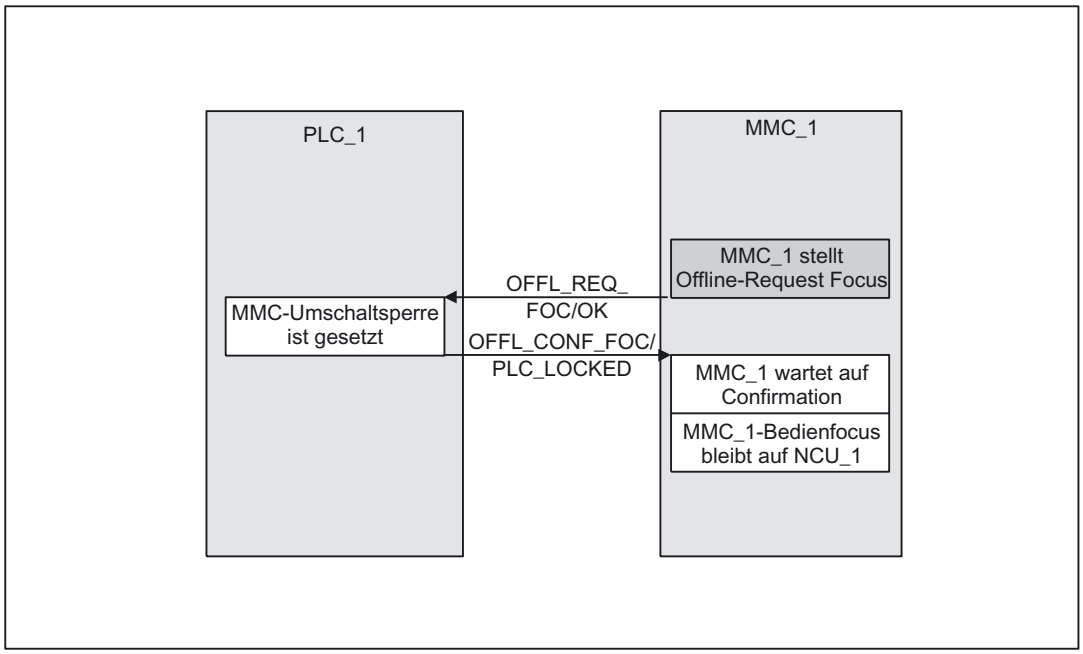


Bild 2-42 MMC_1 Server, möchte Bedienfocus von NCU_1 auf NCU_2 umschalten, in PLC_1 besteht Umschaltsperre

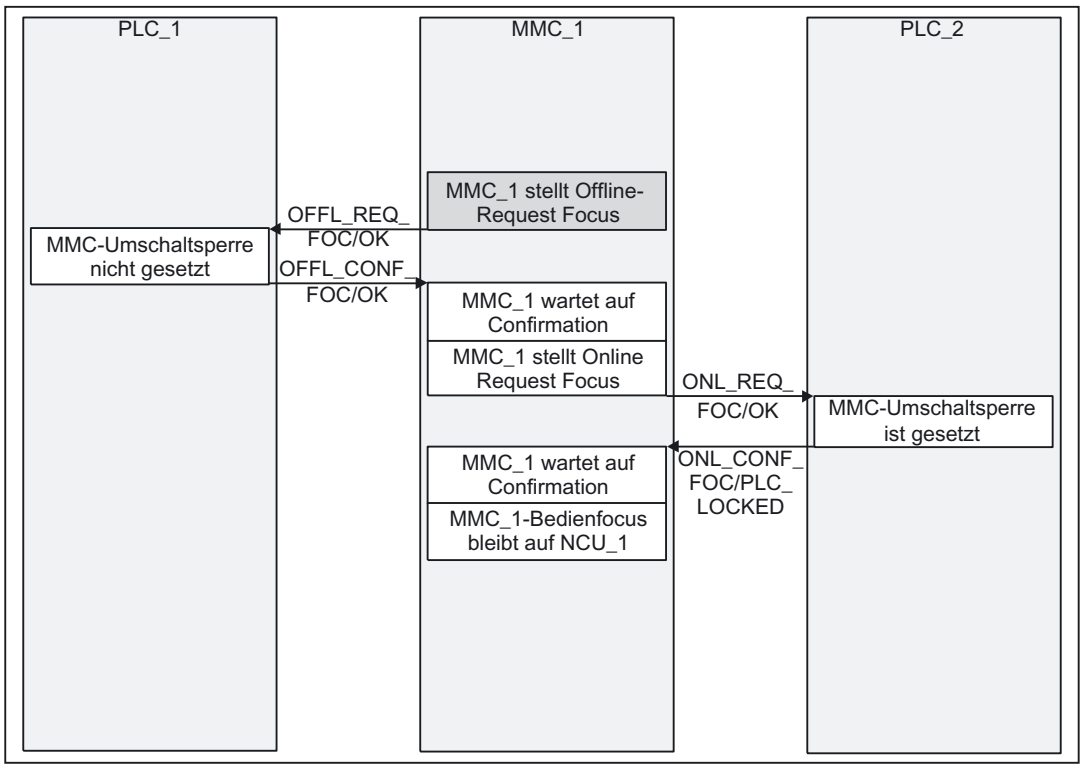


Bild 2-43 MMC_1 ist Server, möchte Bedienfocus von NCU_1 auf NCU_2 umschalten, in PLC_2 ist Umschaltsperre gesetzt

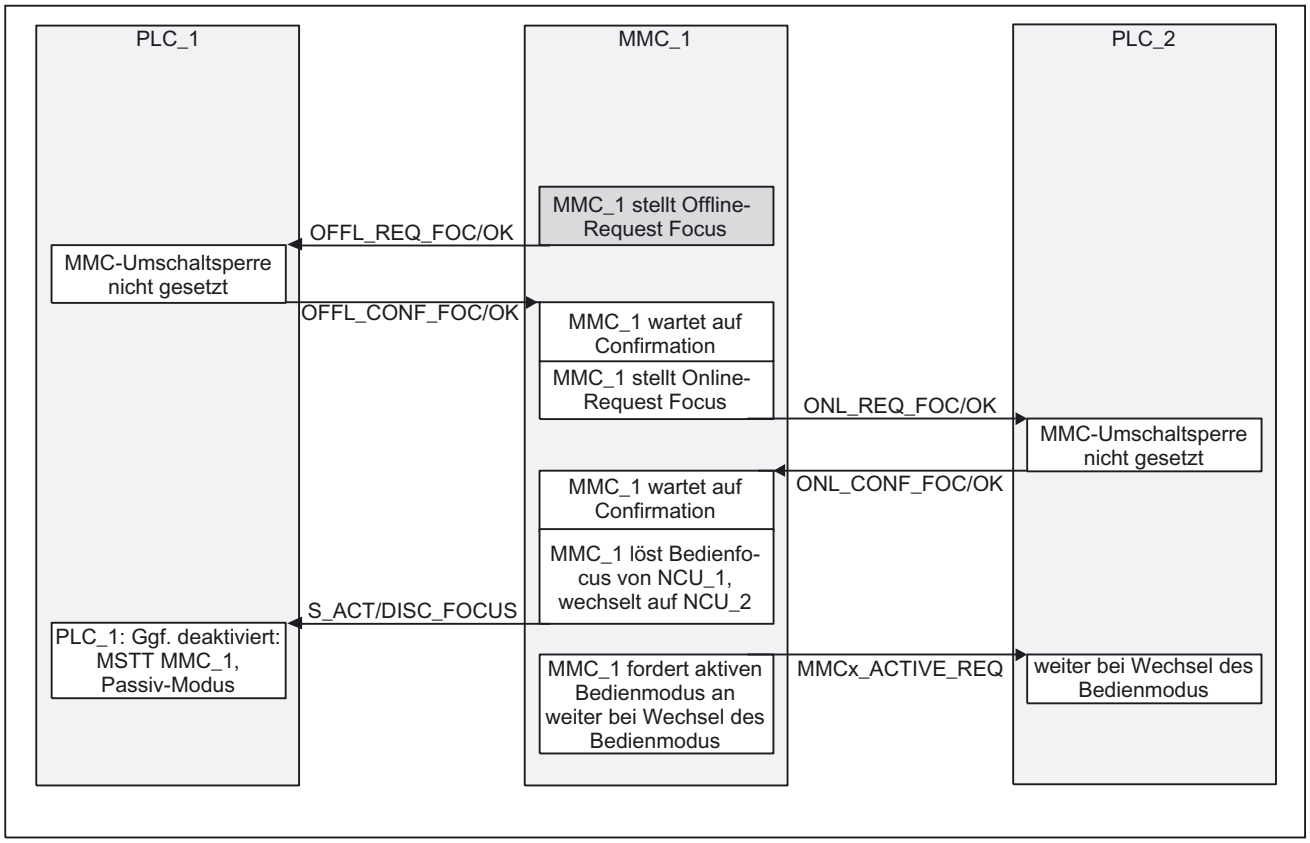


Bild 2-44 MMC_1 ist Server, möchte Bedienfocus von NCU_1 auf NCU_2 umschalten, keine Umschaltsperre in den PLCs, MMC_1 kann Bedienfocus wechseln

2.15.3 Konfigurationsdatei NETNAMES.INI Standardlösung

2.15.3.1 Zwei Bedientafelfronten und eine NCU

Für ein System bestehend aus zwei Bedieneinheiten und einer NCU an der BTSS wird im folgenden eine Beispiel-Konfigurationsdatei für die zweite Bedieneinheit angegeben.

```
; NETNAMES.INI Beispiel 2 Anfang

; Erkennungsteil
[own]
owner = MMC_2
; Verbindungsteil
[conn MMC_1]
conn_1 NCU_1

[conn MMC_2]
conn_1 NCU_1
; Beschreibungsteil
[param network]
bus = btss

[param MMC_1]
mmc_address = 1

[param MMC_2]
mmc_address = 3

[param NCU_1]
nck_address = 13
plc_address = 13

; NETNAMES.INI Beispiel 2 Ende
```

2.15.3.2 Eine Bedientafelfront und drei NCUs

Für ein System bestehend aus einer Bedieneinheit und drei NCUs an der BTSS wird im folgenden eine Beispiel-Konfigurationsdatei angegeben.

Die ggf. erforderlichen Anpassungen sind im Kapitel "Konfigurationen" beschrieben.

```
; NETNAMES.INI, Beispiel 3 Anfang
; Erkennungsteil:
[own]
owner= MMC_1

; Verbindungsteil: Für die vorgesehenen bis zu 3 Verbindungen
[conn MMC_1]
conn_1= NCU_1
conn_2= NCU_2
conn_3= NCU_3

; Beschreibungsteil: Es wird das Netz eindeutig bezeichnet
[param network]
bus= btss

[param MMC_1]
name= beliebiger_name
type= MMC_100
mmc_address= 1

[param NCU_1]
name= beliebiger_name1
type= ncu_572
nck_address= 12
plc_address= 12

[param NCU_2]
name= beliebiger_name2
type= ncu_573
nck_address= 14
plc_address= 14

[param NCU_3]
name= beliebiger_name3
```

type= ncu_573

nck_address= 15

plc_address= 15

; NETNAMES.INI, Beispiel 3 Ende

2.15.4 Kurzinbetriebnahme M:N anhand von Beispielen

Einführung

Es wird nicht auf Netzwerkregeln MPI/BTSS - Bus eingegangen.

Siehe **Literatur**:

/BH/ Bedienkomponenten-Handbuch

Anhand von drei Beispielen wird gezeigt, wie schrittweise ein M:N-Verbund in Betrieb zu nehmen ist. Die Beschreibung beginnt jeweils mit der Darstellung einer Konfiguration.

2.15.4.1 Beispiel 1

Konfiguration HW

Die HW besteht aus den Komponenten:

- 1 Bedientafel (PCU50 mit HMI Advanced, Operator Panel, Maschinensteuertafel)
- 2 NCUs mit je 2 Kanälen

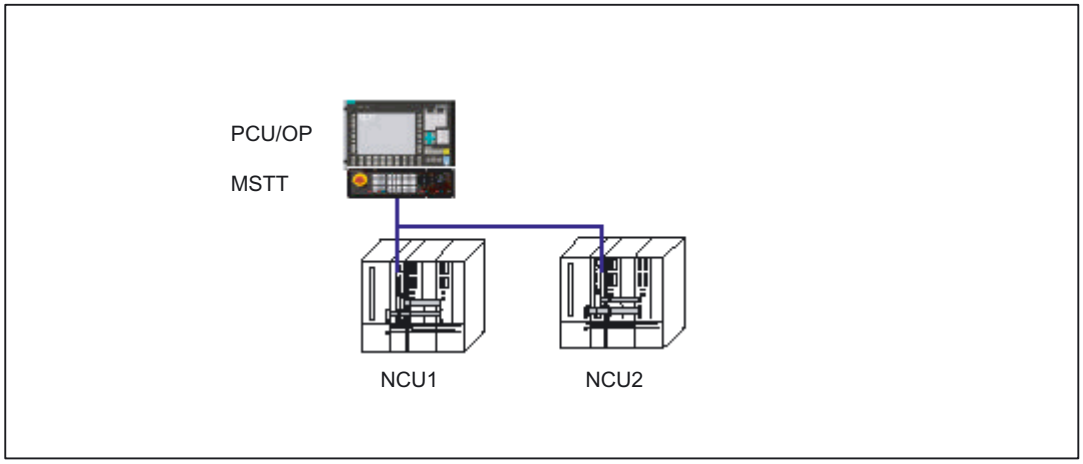


Bild 2-45 Eine Bedientafel für 2 NCUs

Schritt 1:

Konfigurationsdatei NETNAMES.INI

Für dieses Beispiel werden folgende Einträge gesetzt:

```
[own]
owner= MMC_1
; Verbindungsteil
[conn MMC_1]
conn_1 = NCU_1
conn_2 = NCU_2
; Extcall bei einer PCU nicht notwendig
; Netzwerkparameter
[param network]
bus = btss
; HMI-Beschreibungen
[param MMC_1]
mmc_address = 1
; alle anderen Angaben sind nicht notwendig
; Beschreibungsteil NCU-Komponenten
[param NCU_1]
type = NCU_573
nck_address = 22
plc_address = 22
name = NCU1

[param NCU_2]
type = NCU_573
nck_address = 23
plc_address = 23
name = NCU2

; Kanaldaten
[chan MMC_1]
DEFAULT_logChanSet = Station_1
DEFAULT_logChan = N1_K1
ShowChanMenu = True
```

```
logChanSetList = Station_1, Station_2

[Station_1]
logChanList = N1_K1, N1_K2
[N1_K1]
logNCName = NCU_1
ChanNum = 1
[N1_K2]
logNCName = NCU_1
ChanNum = 2

[Station_2]
logChanList = N2_K1, N2_K2
[N2_K1]
logNCName = NCU_2
ChanNum = 1
[N2_K2]
logNCName = NCU_2
ChanNum = 2
; Ende
```

Schritt 2:

Laden von Datei NETNAMES.INI

HMI Advanced/PCU50: Nach der Erstellung der Datei NETNAMES.INI wird diese in das USER Verzeichnis der PCU übertragen.

Schritt 3:

Einstellung der NCK Busadressen

1. Über "Inbetriebnahme → HMI → Bedientafel" geben Sie in die Eingabemaske "Bedientafelfront-Schnittstellenparameter" Folgendes ein:
   Verbindung: M:N (Select M:N statt 1:1
   NCK-Adresse: 22
   PLC-Adresse: 22
   entsprechend NETNAMES.INI für NCU2 Adresse 23
2. "Speichern"
3. PCU neu starten

Schritt 4:

PLC

Ein FB9 Aufruf ist für diese Konfiguration nicht notwendig, da hier kein Verdrängen bzw. Aktiv-/Passiv-Schalten stattfindet.

Softkey Beschriftung

Die Texte werden aus der Datei NETNAMES.INI übernommen. Für das vorliegende Beispiel sind keine Textangaben über NETNAMES.INI hinaus nötig.

2.15.4.2 Beispiel 2

Konfiguration HW

Die HW besteht aus den Komponenten:

- Bedientafel 1 (PCU50 mit HMI Advanced, Operator Panel, Maschinensteuertafel)
- Bedientafel 2 (PCU20 mit HMI Embedded, Operator Panel, Maschinensteuertafel)
- 2 NCUs mit je 2 Kanälen

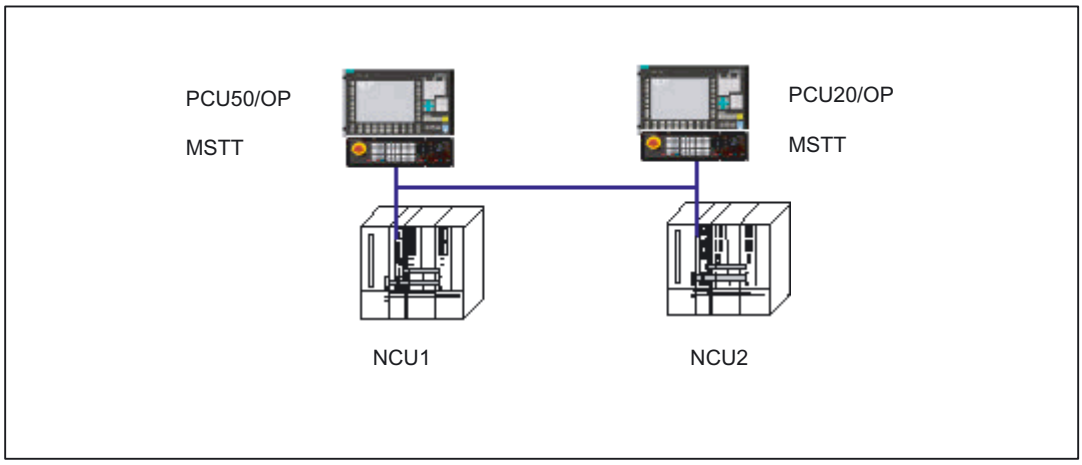


Bild 2-46 Zwei Bedientafeln für 2 NCUs

Nutzung

Bedientafel 1 (Server) und Bedientafel 2 können auf NCU1 und NCU2 zugreifen.

Schritt 1a):

Konfigurationsdateien NETNAMES.INI

Für dieses Beispiel werden für die Bedientafeln in den Dateien NETNAMES.INI eigenständige Einträge gesetzt:

Bedientafel 1

Einträge für HMI Advanced/PCU50:

```
[own]
owner= MMC_1
; Verbindungsteil
[conn MMC_1]
conn_1 = NCU_1
conn_2 = NCU_2
EXTCALL_conns = conn_1, conn_2
; Netzwerkparameter
[param network]
bus = btss
; HMI-Beschreibungen
[param MMC_1]
mmc_typ = 0x40
mmc_bustyp = BTSS
mmc_address= 1
mstt_address = 6
name = MMC_Serv
start_mode = ONLINE
; Beschreibungsteil NCU-Komponenten
[param NCU_1]
type = NCU_573
nck_address = 22
plc_address = 22
name = NCU1

[param NCU_2]
type = NCU_573
nck_address = 23
plc_address = 23
```

```
name = NCU2

; Kanaldaten
[chan MMC_1]
DEFAULT_logChanSet = Station_1
DEFAULT_logChan = N1_K1
ShowChanMenu = True
logChanSetList = Station_1, Station_2

[Station_1]
logChanList = N1_K1, N1_K2
[N1_K1]
logNCName = NCU_1
ChanNum = 1
[N1_K2]
logNCName = NCU_1
ChanNum = 2

[Station_2]
logChanList = N1_K1, N1_K2
[N1_K1]
logNCName = NCU_2
ChanNum = 1
[N1_K2]
logNCName = NCU_2
ChanNum = 2
; Ende
```

Schritt 2a):

Laden von Datei NETNAMES.INI

HMI Advanced/PCU50: Nach der Erstellung der Datei NETNAMES.INI wird diese in das USER Verzeichnis der PCU übertragen.

Schritt 1b):

Bedientafel 2

Einträge für HMI Embedded/PCU20:

```
[own]
owner= PCU20
; Verbindungsteil
[conn PCU20]
conn_1 = NCU_1
conn_2 = NCU_2
; Netzwerkparameter
[param network]
bus = btss
; HMI-Beschreibungen
[param PCU20]
mmc_typ = 0x10
mmc_bustyp = BTSS
mmc_address = 2
mstt_address = 7
name = MMC_Neben
start_mode = OFFLINE
; Beschreibungsteil NCU-Komponenten
[param NCU_1]
type =NCU_573
nck_address = 22
plc_address = 22
name = NCU1
 [param NCU_2]
type = NCU_573
nck_address = 23
```

```
plc_address = 23
name = NCU2
; Kanaldaten
[chan PCU20]
DEFAULT_logChanSet = Station_2
DEFAULT_logChan = N1_K1
ShowChanMenu = True
logChanSetList = Station_1, Station_1
 [Station_1]
logChanList = N1_K1, N1_K2
[N1_K1]
logNCName = NCU_1
ChanNum = 1
[N1_K2]
logNCName = NCU_1
ChanNum = 2
 [Station_2]
logChanList = N1_K1, N1_K2
[N1_K1]
logNCName = NCU_2
ChanNum = 1
[N1_K2]
logNCName = NCU_2
ChanNum = 2
; Ende
```

Softkey Beschriftung

Zur Unterscheidung der anzusprechenden NCU müssen für die Beschriftung der OP-Softkeys Texte in der Datei **chan.txt** vorgegeben werden:

```
//*Länge des Textes max. 2*9 Zeichen*/
//* Zeilenwechsel mit %n am Ende der ersten Zeile erzeugen*/
//* Name des Kanalbereiches 1 und Namen der Kanäle dieses Bereiches */
T_CHAN_AREA_1 "Stat_1"
T_CHAN_AREA_1_CHANNEL_1 "N1_K1"
T_CHAN_AREA_1_CHANNEL_2 "N1_K2"
//* Name des Kanalbereiches 2 und Namen der Kanäle dieses Bereiches */
T_CHAN_AREA_2 "Stat_2"
```

T_CHAN_AREA_2_CHANNEL_1 "N2_K1"

T_CHAN_AREA_2_CHANNEL_2 "N2_K2"

Schritt 2b:

PCU20

Nach der Erstellung derDateien NETNAMES.INI und der Datei chan.txt, werden diese mit der Applikation in das *.abb-File eingebunden.

Schritt 3:

Einstellung der NCK Busadressen

HMI Advanced/PCU50:

1. Über "Inbetriebnahme → HMI → Bedientafel" in die Eingabemaske "Bedientafelfront-Schnittstellenparameter" geben Sie Folgendes ein:
   Verbindung: M:N (mit Taste Select statt 1:1)
   NCK-Adresse: 22
   PLC-Adresse: 22
   entsprechend NETNAMES.INI für NCU2 Adresse 23
2. "Speichern"
3. PCU neu starten

HMI Embedded/PCU20:

Übertragen Sie *.abb mit PC-Card auf das System und führen Sie einen SW-Update aus.

Hinweis

Sollte die Datei "chan.txt" im *.abb vergessen worden sein, sind nach Anwahl der Kanalmenü -Taste, keine beschrifteten Softkeys zu sehen. Die Auswahl-Funktion ist aber vorhanden.

Schritt 4:

PLC

Binden Sie FB9 in das Anwender-PLC-Programm ein. Erläuterungen hierzu finden Sie nach diesen Beispielen.

2.15.4.3 Beispiel 3

Konfiguration HW

Die HW besteht aus den Komponenten:

- 1 Bedientafel (PCU50 mit HMI Advanced, Operator Panel)
- 1 HT6
- 2 NCUs mit je zwei Kanälen

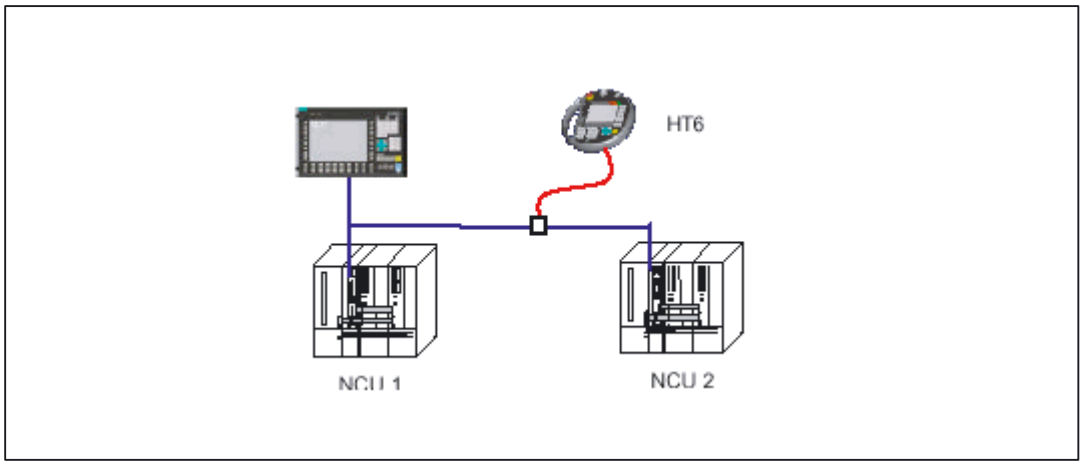


Bild 2-47 Bedientafel und HT6 für 2 NCUs

Nutzung

Die Bedientafel (Server) kann auf NCU1 und NCU2 zugreifen.

Das HT6 kann nur auf NCU2 zugreifen.

Schritt 1a:

Erstellung der Datei NETNAMES.INI für HMI Advanced/PCU50

[own)

owner= MMC_1

; Verbindungsteil

[conn MMC_1]

conn_1= NCU_1

conn_2= NCU_2

EXTCALL_conns = conn_1, conn_2

; Netzwerkparameter

[param network]

bus= btss

```
; HMI-Beschreibungen
[param MMC_1]
mmc_typ = 0x40
mmc_bustyp = BTSS
mmc_address= 1
mstt_address = 255 ; 255 ist erforderlich, wenn keine MSTT
; vorhanden ist.
name = MMC_Serv
start_mode = ONLINE
; Beschreibung der NCU-Komponenten
[param NCU_1]
type = NCU_573
nck_address = 22
plc_address = 22
name = NCU1
 [param NCU_2]
type =NCU_573
nck_address = 23
plc_address = 23
name = NCU2
; Kanaldaten
[chan MMC_1]
DEFAULT_logChanSet = Station_1
DEFAULT_logChan = N1_K1
ShowChanMenu = True
logChanSetList = Station_1, Station_2
[Station_1]
logChanList = N1_K1, N1_K2
[N1_K1]
logNCName = NCU_1
ChanNum = 1
[N1_K2]
logNCName = NCU_1
ChanNum = 2
 [Station_2]
logChanList = N1_K1, N1_K2
```

```
[N1_K1]
logNCName = NCU_2
ChanNum = 1
[N1_K2]
logNCName = NCU_2
ChanNum = 2
; Ende
```

Schritt 1b:

```
Erstellung der Datei NETNAMES.INI für HT6
[own]
owner = HT_6
; Verbindungsteil
[conn HT_6]
conn_1 = NCU_2
; Netzwerkparameter
[param network]
bus = btss
; HMI-Beschreibungen
[param HT_6]
mmc_typ = 0x10
mmc_bustyp = BTSS
mmc_address = 14
mstt_address = 14 ; ist immer gleich mit
; mmc address
name = MMC_Neben
start_mode = OFFLINE
; Beschreibung der NCU-Komponenten
[param NCU_2]
type =NCU_573
nck_address= 23
plc_address= 23
name = NCU2
; Kanaldaten
[chan HT_6]
DEFAULT_logChanSet = Station_2
```

DEFAULT_logChan = N1_K1

ShowChanMenu = True

logChanSetList = Station_2

[Station_2]

logChanList = N2_K1, N2_K2

[N2_K1]

logNCName = NCU_2

ChanNum = 1

[N2_K2]

logNCName = NCU_2

ChanNum = 2

;End of file

Schritt 2a:

PCU50:

Nach der Erstellung der Datei NETNAMES.INI wird diese in das USER Verzeichnis übertragen bzw. kopiert.

Schritt 2b:

HT6:

Erstellung der Softkeytexte siehe Beispiel 2.

Nach der Erstellung der Datei NETNAMES.INI und der Datei chan.txt, werden diese mit derApplikation in das *.abb-File eingebunden.

Schritt 3:

Einstellung der NCK Busadressen

HMI Advanced/PCU50:

1. Über "Inbetriebnahme → HMI → Bedientafel" geben Sie in die Eingabemaske "Bedientafelfront-Schnittstellenparameter" Folgendes ein:
   Verbindung: M:N (Select statt 1:1)
   NCK-Adresse: 22
   PLC-Adresse: 22
   entsprechend NETNAMES.INI für NCU2 Adresse 23
2. "Speichern"
3. PCU neu starten

Schritt 4:

Binden Sie FB9 in das PLC-Anwenderprogramm ein. Erläuterungen hierzu finden Sie im folgenden Kapitel.

2.15.4.4 Beschreibung von FB9

Funktionsbeschreibung

Dieser Baustein ermöglicht die **Umschaltung** mehrerer **Bedientafeln** (PCU mit Operator Panel und/oder Maschinensteuertafel), die über ein Bussystem mit einer/mehreren Steuerungsbaugruppen NCU verbunden sind.

Die **Schnittstelle** zwischen den einzelnen Bedientafeln und der NCU (PLC) ist die M:N Schnittstelle im **Datenbaustein DB19** (siehe Kapitel "Signalbeschreibungen" und /LIS/, Listen).

Der FB 9 arbeitet mit den Signalen dieser Schnittstelle.

Folgende **Grundfunktionen** werden neben der Initialisierung, der Lebenszeichenüberwachung und der Fehlerroutinen vom Baustein für die Bedieneinheitenumschaltung ausgeführt:

Tabelle 2- 7 Übersicht der Funktionen

Grundfunktion	Bedeutung
PCU klopft an	PCU hat den Wunsch an eine NCU online zu gehen
PCU kommt	PCU verbindet sich mit einer NCU
PCU geht	PCU trennt sich von einer NCU
Verdrängung	PCU muss Verbindung zu einer NCU lösen
Bedienfocuswechsel im Serverbetrieb	Bedienfocus von einer auf die andere NCU umschalten
Aktiver/Passiver Bedienmodus	Bedienen und beobachten/nur beobachten
MSTT-Umschaltung	MSTT kann optional mit dem PCU umgeschaltet werden

Die folgenden Erläuterungen ergänzen die Angaben aus "Beschreibung der Abläufe (Übersicht)" und "Beschreibung der Abläufe (Details)" mit Schwerpunkt der in den vorangehenden 3 Beispielen verwendeten Sachverhalte.

Kurzbeschreibung wichtiger Funktionen

Aktiver/Passiver Bedienmodus:

Ein Online-PCU kann zwei verschiedene Bedienmodi haben:

Aktiv-Modus: Bediener kann bedienen und beobachten

Passiv-Modus: Bediener kann beobachten (nur Kopfzeile PCU)

Nach Umschaltung auf eine NCU fordert sie in der PLC der Online-NCU zuerst den aktiven Bedienmodus an. Falls zwei PCU's zu einer Zeit an einer NCU online sind, ist eine der beiden immer im aktiven, die andere im passiven Bedienmodus. Der Bediener kann den aktiven Bedienmodus an der passiven PCU per Tastendruck anfordern.

MSTT-Umschaltung:

Zusammen mit der PCU kann optional eine zugeordnete MSTT mit umgeschaltet werden. Voraussetzung ist, dass die MSTT-Adresse im Parameter mstt_adress der Konfigurationsdatei NETNAMES.INI der PCUs eingetragen und MCPEnable true gesetzt ist. Die MSTT des passiven PCU's ist deaktiviert. Dadurch gibt es an einer NCU immer nur eine aktive MSTT.

Hochlaufbedingung:

Um zu verhindern, dass bei einem Neustart der NCU, die davor zuletzt angewählte MSTT aktiviert wird, muss beim Aufruf von FB1 in OB100 der Eingangsparameter MCP1BusAdr = 255 (Adresse 1. MSTT) und MCP1STOP = TRUE (1.MSTT ausschalten) gesetzt werden.

Freigaben:

Wenn von einer MSTT auf eine andere umgeschaltet wird, dann bleiben eventuell eingeschaltete Vorschub- sowie Achsfreigaben erhalten.

ACHTUNG
Die zur Zeit der Umschaltung betätigten Tasten wirken bis zur Aktivierung der neuen MSTT (vom HMI, der anschließend aktiviert wird) weiter. Auch die Overridestellungen für Vorschub, Spindel bleiben erhalten. Um die betätigten Tasten zu deaktivieren, ist bei fallender Flanke des DB10.DBX104.0 das Eingangsabbild der Maschinensteuer-Signale auf nicht betätigte Signalpegel zu legen. Die Overridestellungen sollten unverändert bleiben.

Maßnahmen zur Deaktivierung der Tasten sind im PLC-Anwenderprogramm zu realisieren. (s.u.: Beispiel Override-Umschaltung)

Deklaration der Funktion

FUNCTION_BLOCK FB9

```
VAR_INPUT
Quit : BOOL; // Quittierung Alarme
OPMixedMode : BOOL:= FALSE ; // Mischbetrieb mit nicht M:N fähigem
// OP deaktiviert!
AktivEnable : BOOL:= TRUE ; // Aktiv/passiv Umschaltung aktivieren.
MCPEnable : BOOL:= TRUE ; // MSTT Umschaltung aktivieren
END_VAR
VAR_OUTPUT
Alarm1 : BOOL ; // Alarm: Fehler PCU-Busadresse, Bustyp!
Alarm2 : BOOL ; // Alarm: Keine Bestätigung MMC1 offline!
Alarm3 : BOOL ; // Alarm: MMC1 geht nicht offline!
Alarm4 : BOOL ; // Alarm: Keine Bestätigung MMC2 offline!
Alarm5 : BOOL ; // Alarm: MMC2 geht nicht offline!
Alarm6 : BOOL ; // Alarm: Anklopf-PCU geht nicht online!
```

Report : BOOL ; // Meldung: Lebenszeichenüberwachung

ErrorMMC : BOOL ; // Fehlererkennung HMI

END_VAR

Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion FB9

Tabelle 2- 8 Formalparameter FB9

Signal	Art	Typ	Bemerkung
Quit	E	BOOL	Quittierung Alarme
OPMixedMode	E	BOOL	Mischbetrieb mit nicht M:N fähigen OP
AktivEnable	E	BOOL	Bedientafel aktiv/passiv Umschaltung aktivieren TRUE = Bedientafel kann aktiv/passiv geschaltet werden. FALSE = Bedientafel kann nicht aktiv geschaltet werden und bleibt in seinemZustand.
MCPEnable	E	BOOL	MSTT Umschaltung aktivieren TRUE = MSTT wird mit Bedientafel umgeschaltet FALSE: = MSTT wird nicht mit Bedientafel umgeschaltet.
Alarm1	A	BOOL	Alarm: Fehler PCU-Busadresse, Bustyp!
Alarm2	A	BOOL	Alarm: Keine Bestätigung PCU1 offline!
Alarm3	A	BOOL	Alarm: PCU1 geht nicht offline!
Alarm4	A	BOOL	Alarm: Keine Bestätigung PCU2 offline!
Alarm5	A	BOOL	Alarm: PCU2 geht nicht offline!
Alarm6	A	BOOL	Alarm: Anklopf-PCU geht nicht online!
Report	A	BOOL	Meldung: Lebenszeichenüberwachung
ErrorMMC	A	BOOL	Fehlererkennung HMI

Hinweis

Der Baustein ist vom Anwenderprogramm aufzurufen. Hierbei ist ein Instanz-DB mit beliebiger Nummer vom Anwender beizustellen. Der Aufruf ist nicht multiinstanzfähig.

2.15.4.5 Aufrufbeispiel für FB9

CALL FB 9 , DB 109 (

Quit := Fehler_Quitt, // z. B. MSTT-Reset

OPMixedMode := FALSE,

AktivEnable := TRUE, // Freigabe PCU Umschaltung

MCPEnable := TRUE, // Freigabe MSTT Umschaltung

Alarm1 := DB2.dbx188.0, // Fehlermeldung 700.100

Alarm2 := DB2.dbx188.1, // Fehlermeldung 700.101

Alarm3 := DB2.dbx188.2, // Fehlermeldung 700.102

Alarm4 := DB2.dbx188.3, // Fehlermeldung 700.103

Alarm5 := DB2.dbx188.4, // Fehlermeldung 700.104

Alarm6 := DB2.dbx188.5, // Fehlermeldung 700.105

Report := DB2.dbx192.0, // Betriebsmeldung 700.132

ErrorMMC := DB2.dbx192.1) // Betriebsmeldung 700.133

Hinweis

AktivEnable := true ermöglicht die PCU aktiv/passiv Umschaltung.

MCPEnable := true erlaubt die MSTT Umschaltung.

Der Defaultwert dieserParameter ist so geschaltet und muss beim Aufruf der Funktion nicht zusätzlich beschaltet werden.

Alarme, Fehler

Die Ausgangsparameter "Alarm1" bis "Alarm6" und "Report" können für die Alarm- und Fehlermeldungen der HMI-Software in den Bereichen des DB2 übergeben werden.

Konnte eine HMI-Funktion nicht ausgeführt werden (bei der die Fehlermeldung nicht angezeigt werden kann), wird dies am Zustandsparameter ErrorMMC mit 'logisch 1' angezeigt (z B. Fehler im Hochlauf, wenn keine Verbindung aufgebaut wird).

Aufrufbeispiel für FB1

(Aufruf im OB100):

CALL "RUN_UP" , "gp_par" (

MCPNum := 1,

MCP1In := P#E 0.0,

MCP1Out := P#A 0.0,

MCP1StatSend := P#A 8.0,

MCP1StatRec := P#A 12.0,

MCP1BusAdr := 255, // Adresse 1. MSTT

MCP1Timeout := S5T#700MS,

MCP1Cycl := S5T#200MS,

MCP1Stop := TRUE, // MSTT ausgeschaltet

NCCyclTimeout := S5T#200MS,

NCRunupTimeout := S5T#50S);

2.15.4.6 Beispiel für Override-Umschaltung

Das Beispiel verwendete Hilfsmerker M100.0, M100.1, M100.2, M100.3.

Die positive Flanke von MCP1Ready muss auf Override prüfen und Maßnahmen für Aktivierung des MSTT Bausteins einleiten.

Dieses Beispiel gilt für Vorschub-Override. Für Spindel-Override sind Nahtstellen- und Eingangsbyte auszutauschen.

```
U DB10.DBX 104.0; // MCP1Ready
 FN M 100.0; // Flankenmerker 1
 SPBN wei1;
 S M 100.2; // Hilfsmerker 1 setzen
 R M 100.3; // Hilfsmerker 2 rücksetzen
// Override speichern
 L DB21.DBB 4; // Nahtstelle Vorschub-Override
 T EB 28; // Zwischenspeicher (freies Eingangs-
// oder Merkerbyte)
wei1:
 U M 100.2; //Umschaltung erfolgt
 O DB10.DBX 104.0; //MCP1Ready
 SPBN wei2;
 U DB10.DBX 104.0; //MCP1Ready
 FP M 100.1; //Flankenmerker 2
 SPB wei2;
 U M 100.2; //Umschaltung erfolgt
 R M 100.2; //Hilfsmerker 1 rücksetzen
 SPB wei2;
 U M 100.3; //Vergleich ist erfolgt
 SPB MCP; //MSTT-Programm aufrufen
// gespeicherten Override auf Nahtstelle der umgeschalteten MSTT lenken
// bis die Override-Werte übereinstimmen
 L EB28; //Zwischenspeicher auf
 T DB21.DBB 4; //Nahtstelle Override lenken
 L EB 3; //Override-Eingangsbyte fuer Vorschub
 <>i; //Übereinstimmung?
 SPB wei2; //nein, Absprung
 S M100.3; //ja, Hilfsmerker 2 setzen
// nach Übereinstimmung der Override-Werte MSTT-Programm wieder aufrufen
```

```
MCP: CALL "MCP_IFM"( //FC 19
 BAGNo := B#16#1,
 ChanNo := B#16#1,
 SpindleIFNo := B#16#0,
 FeedHold := M 101.0,
 SpindleHold := M 101.1);
wei2: NOP 0;
```

2.15.4.7 Umschaltung zwischen MSTT und HT6

```
CALL FCxx
 L DB7.DBB 27 // akt. MSTT
 L 6 // Maschinensteuertafel
 ==I
 SPB MSTT // Aufruf FC 19
 L DB7.DBB 27 // akt. MSTT
 L 14 // HT 6
 ==I
 SPB HT6 // Aufruf FC 26
 SPA ENDE
HT6: NOP 0
 L B#16#40 // Verschiebung der Eingänge des HT6 auf EB 8+n
 T DB7.DBB7
 L B#16#40 // Verschiebung der Ausgänge des HT6 auf AB 8+n
 T DB7.DBB13
 CALL FC26 // Baustein Aufruf HT6
 SPA ENDE
MSTT: NOP 0
 L 0
 T DB7.DBB7
 T DB7.DBB13

 CALL FC19 // Baustein Aufruf Maschinensteuertafel
ENDE: NOP 0
```

2.15.4.8 Allgemeine Hinweise

- Bei einer Konfiguration mit nur **einer** NCU muss in der Datei NETNAMES.INI in der Sektion [param NCU_xx] bei der PLC-Adresse der zusätzliche Eintrag : " ,SAP=202 " gesetzt werden.

Beispiel:

[param NCU_1]

type =NCU_573

nck_address= 11

plc_address= 11**, SAP=202**

name = NCU1

- Bei einer Konfiguration ohne Maschinensteuertafel (Bedientafel ohne MSTT) muss in der jeweiligen Datei NETNAMES.INI in der Sektion [param MMC_xx] im Parametersatz "mstt_address = **255**" eingetragen werden.
- Dies gilt nicht bei HMI Embedded/HT6 , da hier die bt_conf einen Fehler meldet.
- Im PLC Programm wird die Parametierung vom FB1 (Aufruf OB100) standardmäßig parametriert, siehe "FB9 Beschreibung".

Beispiel:

[param MMC_1]

mmc_typ = 0x40

mmc_bustyp = BTSS

mmc_address = 1

mstt_address = **255**

name = MMC_Serv

start_mode = ONLINE

- Empfehlung: Die BTSS/MPI-Adressen 0 (für PG) und 13 (für Service-Fall: NC-Tausch) sollten freigehalten werden.
- OFFLINE-Modus bei HMI Advanced: Ein Server kann nicht mit der Hochlaufeigenschaft start_mode = Offline projektiert werden.

  Soll ein Haupt- bzw. Nebenbedienfeld im Offline-Modus hochlaufen, ist in der Datei MMC.INI folgender Eintrag zu machen.

  Setzen Sie in der Sektion [Global]

  NcddeDefaultMachineName = LOCAL.

  Danach darf nicht mehr im Menü "Bedientafelfront Schnittstellenparameter" auf "Speichern" gedrückt werden, da sonst dieser Eintrag wieder überschrieben wird.

HT6 Abziehen/Stecken

Ein störungsfreies Abziehen und Stecken des HT 6 **im laufenden Maschinenbetrieb** erfordert:

- ein Freigeben bzw. Überbrücken des NOT-AUS des HT 6,
- den Anschluss des HT 6 am BTSS/MPI mittels Profibus-Repeater.

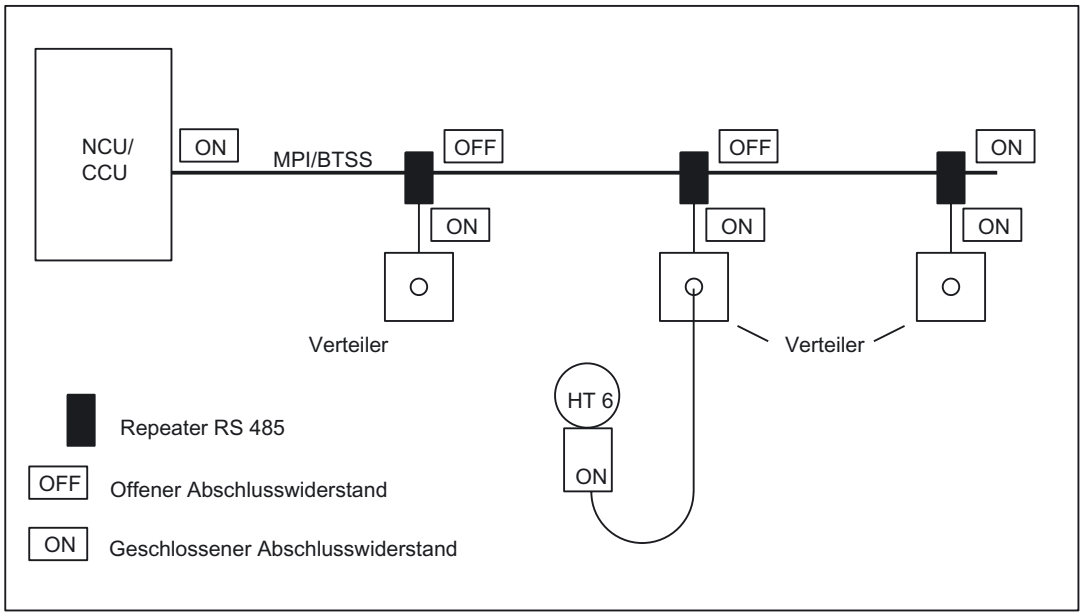


Bild 2-48 Anschluss von HT 6 mittels Profibus-Repeater

Bei jedem Abzweig ist ein Profibus-Repeater vor den HT 6-Verteiler zu schalten. Die einzelnen Bussegmente (MPI/BTSS-Leitung bzw. die lokalen Segmente zwischen Repeater und HT 6) sind an den Busenden mit Abschlusswiderständen zu terminieren.

Repeater RS 485

Der Repeater ist unter der Bestell-Nr. 6ES7972-0AA01-0XA0 bestellbar. Weitere Informationen siehe Katalog

/IK10/ Industrielle Kommunikationsnetze SIMATIC-NET

Hinweis

Bitte beachten Sie Folgendes:

- Das HT 6 beinhaltet bereits einen eingelegten Busabschlusswiderstand.
- Die Leitung vom Repeater zum Verteiler sollte 2 m Länge nicht überschreiten.

Schaltungsvorschläge für NOT-AUS finden Sie in:

Literatur: /BH/, Bedienkomponenten-Handbuch.

2.15.5 Link-Achse

Annahme

NCU1 und NCU2 haben jeweils eine Link-Achse,
Maschinendaten z. B.:

```
; Maschinendaten von NCU1:
$MN_NCU_LINKNO = 1 ; NCU-Nummer auf 1 setzen
 ; (Master-NCU)
$MN_MM_NCU_LINK_MASK = 1 ; Funktion Link aktiv setzen
$MN_MM_SERVO_FIFO_SIZE = 3 ; Größe des Datenpuffers 1)
 ; zwischen Interpolation
 ; und Lageregelung
$MN_MM_LINK_NUM_OF_MODULES = 2 ; Anzahl der Link-Module

$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "AX1"
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "AX2"
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC2_AX3" ; Link-Achse

; Eindeutige NCU-Achsnamen
$MN_AXCONF_MACHAX_NAME_TAB[0] = "NC1_A1"
$MN_AXCONF_MACHAX_NAME_TAB[1] = "NC1_A2"
$MN_AXCONF_MACHAX_NAME_TAB[2] = "NC1_A3"

CHANDATA(1)
$MC_AXCONF_MACHAX_USED[0] = 1
$MC_AXCONF_MACHAX_USED[1] = 2
$MC_AXCONF_MACHAX_USED[2] = 3

...

; Maschinendaten von NCU2:
$MN_NCU_LINKNO = 2 ; NCU-Nummer auf 2 setzen (Slave-NCU)
$MN_MM_NCU_LINK_MASK = 1
$MN_MM_SERVO_FIFO_SIZE = 3 ; 1)
$MN_MM_LINK_NUM_OF_MODULES = 2
```

```
$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "AX1"
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "AX2"
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC1_AX3" ; Link-Achse

; Eindeutige NCU-Achsnamen
$MN_AXCONF_MACHAX_NAME_TAB[0] = "NC2_A1"
$MN_AXCONF_MACHAX_NAME_TAB[1] = "NC2_A2"
$MN_AXCONF_MACHAX_NAME_TAB[2] = "NC2_A3"

CHANDATA(1)
$MC_AXCONF_MACHAX_USED[0] = 1
$MC_AXCONF_MACHAX_USED[1] = 2
$MC_AXCONF_MACHAX_USED[2] = 3
...
```

1) Bei SW-Stand 5 ist das entsprechende Maschinendatum:

MD10087 $MN_SERVO_FIFO_SIZE.

2.15.6 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.

2.15.6.1 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**

2.15.6.2 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.

2.15.6.3 Achscontainer-Drehung durch nur einen Kanal (z. B. Hochlauffall)

Kanal 1	Kanal 2	Kommentar
AXCTWE**D**(C1)	Ist im RESET-Zustand	Drehung erfolgt unmittelbar

2.15.7 Achscontainer Systemvariablen auswerten

2.15.7.1 Bedingte Verzweigung

Kanal 1	Kommentar
AXCTWE(CT1)	Kanal 1 gibt den Achscontainer zur Drehung frei
MARKE1: Teileprogramm **ohne** Bewegung einer Containerachse	
IF $AC_AXCTSWA[CT1] == 1 GOTOB MARKE1	**bedingte Verzweigung**, abhängig vom Vollzug der Achscontainer-Drehung.
Teileprogramm **mit** Bewegung einer Containerachse	

2.15.7.2 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.
	Literatur: /FPSY/, FB Synchronaktionen

2.15.7.3 Sicher auf Achscontainer-Drehung warten

Will man sicher auf das Ende der Achscontainer-Drehung warten, so kann eines der folgenden Beispiele je nach Umfeld herangezogen werden.

Beispiel 1

```
rl = $AN_AXCTAS[ctl] ; Lesen der aktuell. Achscontainer-Stellung
AXCTSWE(ctl) ; Achscontainer-Drehung zulassen
WHILE (rl == $AN_AXCTAS[ctl]) ; Warten, bis Achscontainer-Stellung
ENDWHILE ; geändert
```

Beispiel 2 für 1. Kanal

```
CLEARM(9) ; Synchronisationsmarker 9 löschen
AXCTSWE(ctl) ; Achscontainer-Drehung zulassen
; mit Synchronaktion warten, dass
; Achscontainer-Drehung beendet ist
WHEN $AN_AXCTSWA[ctl] == TRUE DO SETM(9) ; Marke 9 setzen und
WAITMC(9, 1) ; Warten auf Synchronisationsmarke 9
; im 1. Kanal
```

Beispiel 3.1 Internes Warten nutzen

```
M3 S100 ; Achscontainer-Spindel erneut programmieren
; Es wird intern auf das Ende der Achscontainer-Drehung
; gewartet
```

Beispiel 3.2 Internes Warten nutzen

```
x=IC(0) ; Achscontainer-Achse x erneut programmieren
; Es wird intern auf das Ende der Achscontainer-Drehung
; gewartet
```

Beispiel 3.3 Internes Warten nutzen

```
AXCTSWE(CTL) ; Wird ein Achscontainer erneut zur Drehung freigegeben,
; so wird intern auf das Ende der vorausgegangenen
; Achscontainer-Drehung gewartet
```

N2150 WHILE (rl == $AN_AXCTAS[ctl])

Hinweis

Die Programmierung im NC-Programm:

WHILE ($AN_AXCTSWA[n] == 0)

ENDWHILE

kann **nicht** sicher dazu verwendet werden festzustellen, ob eine vorausgegangene Achcontainer-Drehung beendet ist. Obwohl ab SW 7.x $AN_AXCTSWA einen impliziten Vorlaufstopp ausführt, kann diese Programmierung nicht verwendet werden, da der Satz durch ein Reorganisieren unterbrochen werden kann und anschließend die Systemvariable wieder 0 liefert, da die Achscontainer-Drehung dann beendet wurde.

2.15.8 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 Γραδ 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:

Achsanzahl

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.

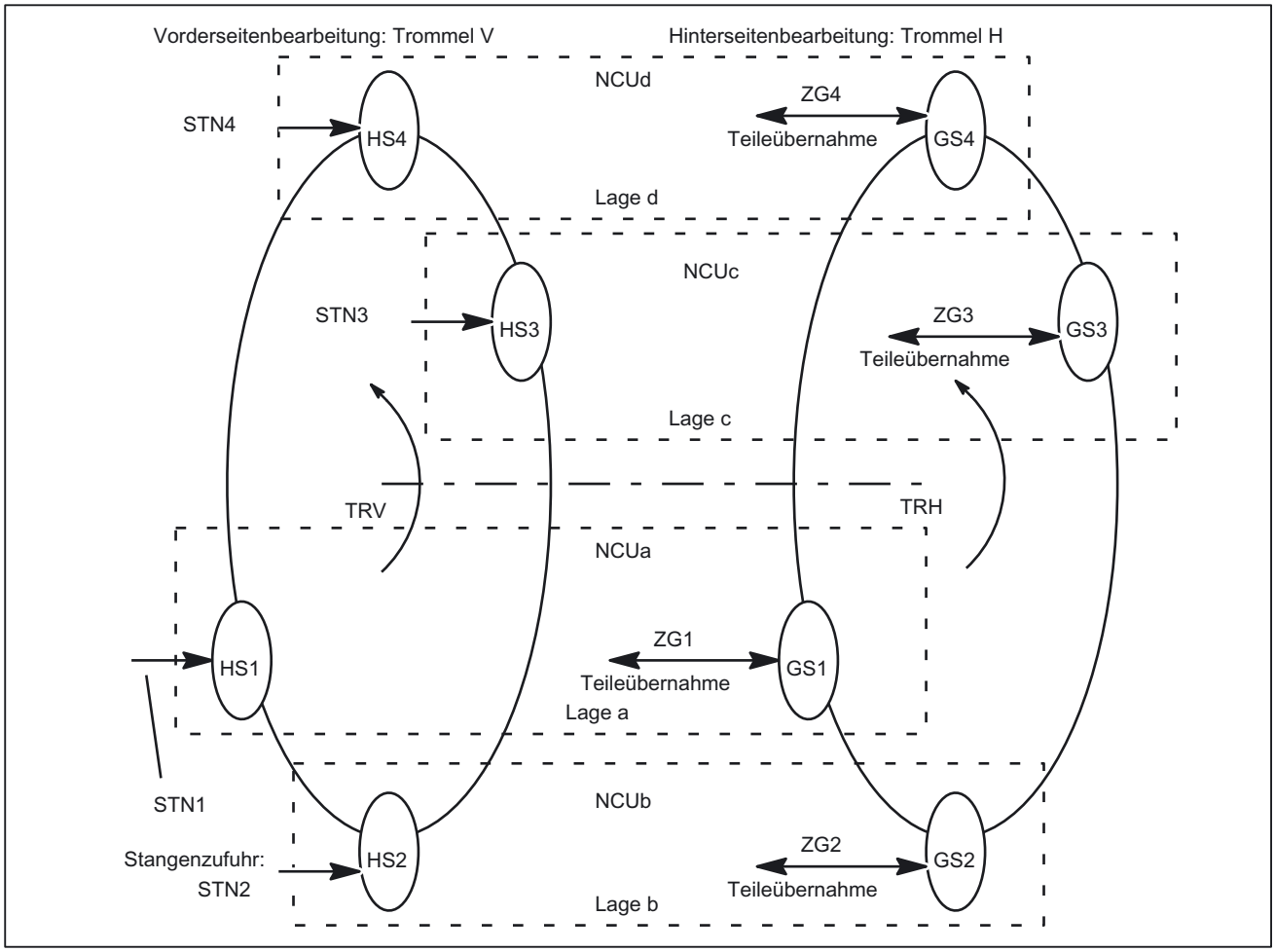


Bild 2-49 Hauptspindeln HSi, Gegensp. GSi, Stangenzufuhrachse STNi und Übernahme-Achsen ZGi schematisch

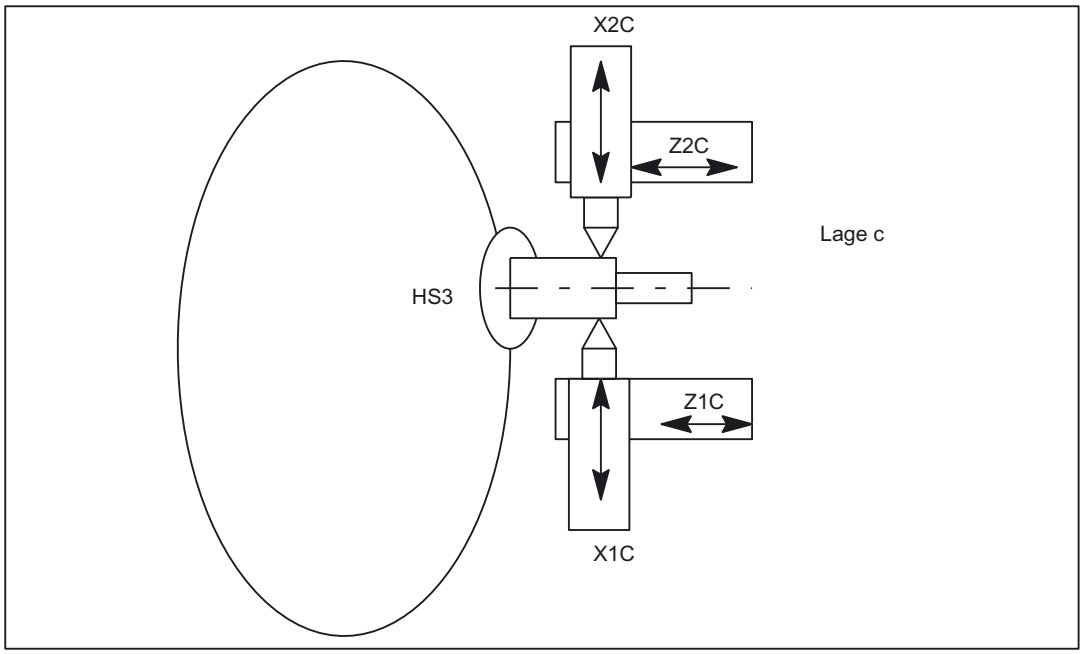


Bild 2-50 Zwei Schlitten je Lage können auch gemeinsam an einer Spindel arbeiten.

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, Stangenzuführungsachsen 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 2- 9 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 Xij, Zij 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-Achsnamen 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 2- 10 NCUa, Lage: a, Kanal: 1, Schlitten: 1

Kanalachs-Name	..._MACHAX _USED	$MN_ AXCONF_LOGIC_MACH AX_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

Kanalachs-Name	..._MACHAX_USED	$MN_AXCONF_LOGIC_MACHAX_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 2- 11 NCUa, Lage: a, Kanal: 2, Schlitten: 2

Kanalachs-Name	..._MACHAX_USED	$MN_AXCONF_LOGIC_MACHAX_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 Maschinenachsname) zu setzen.

Tabelle 2- 12 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	N4C_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

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
Trommel-bewegung		0°	+ 90°	+ 90°	+ 90°	- 270°

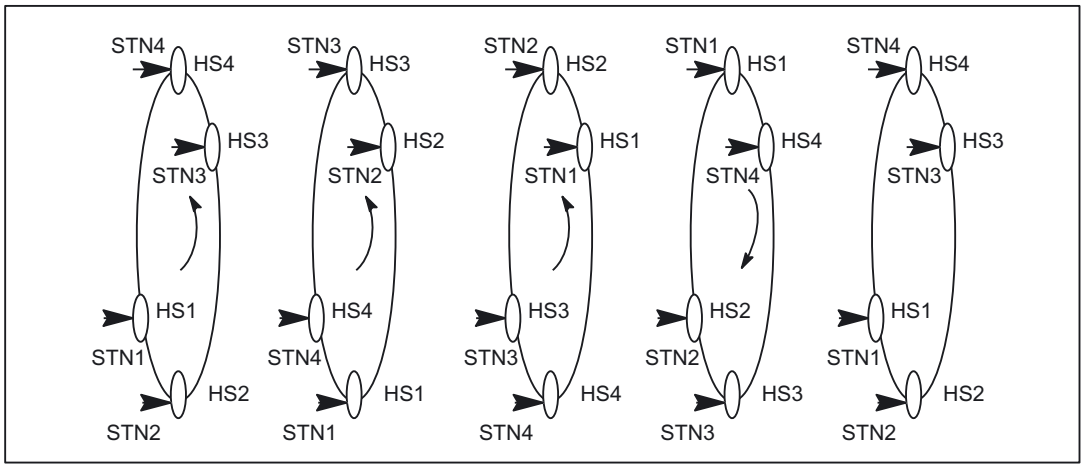


Bild 2-51 Positionen der Trommel A

Tabelle 2- 13 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

2.15.9 Lead-Link-Achse

2.15.9.1 Konfiguration

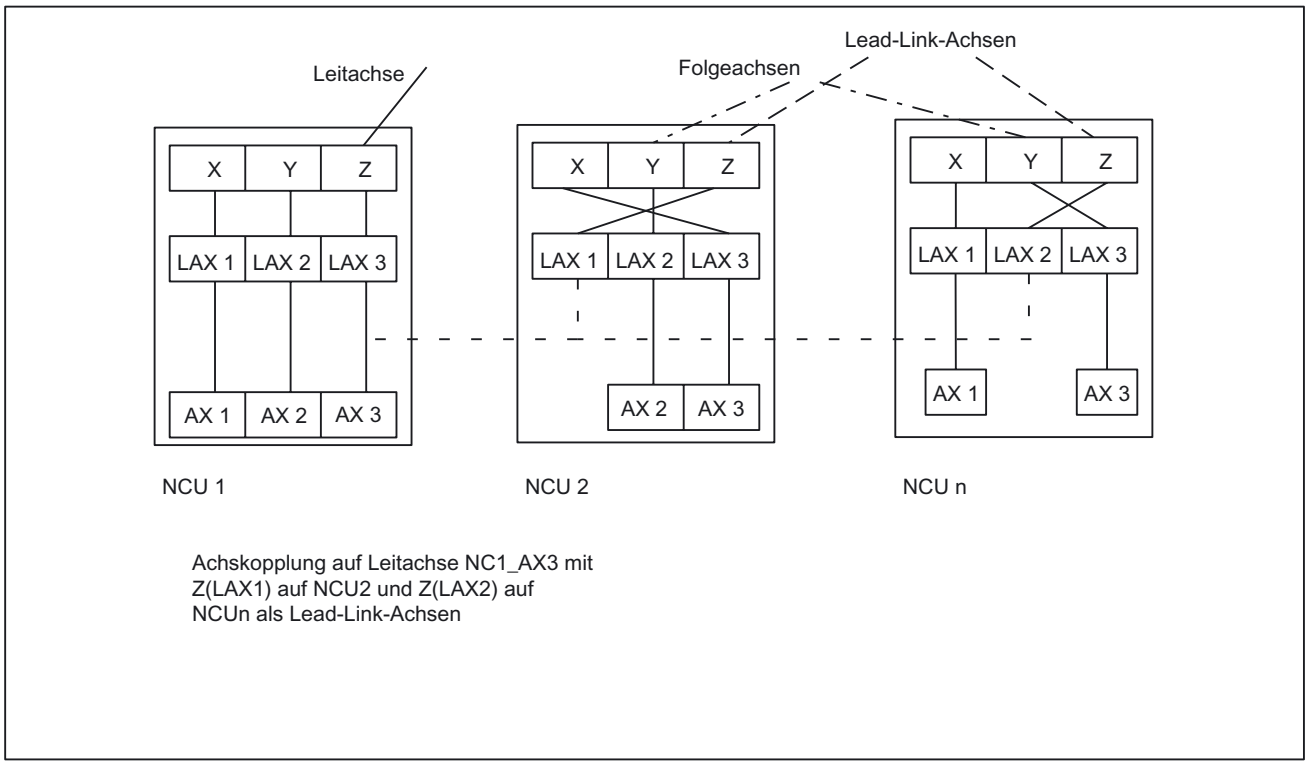


Bild 2-52 NCU2 bis NCUn benutzen eine Lead-Link-Achse, um eine Kopplung die auf Maschinenachse an NCU1. (NCU1-AX3) zu ermöglichen

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.

Laden der Maschinendaten

1. Das Laden der Maschinendaten/Settingdaten einer Leitwert-Achse muss nur auf der Heimat-NCU erfolgen. Von hier aus werden die Maschinendaten intern an die anderen NCUs, bei denen eine Lead-Link-Achse definiert ist, weiterverteilt.
2. Die Lead-Link-Achse muss in der Konfiguration der NCU berücksichtigt werden, welche die Folgeachsen verfährt (NCU2). Die Lead-Link-Achse belegt im logischen Maschinenachs-Abbild (LAI) der (NCU2) einen Platz. Damit reduziert sich die maximal von dieser NCU zu interpolierenden Anzahl Achsen um 1 für die Lead-Link-Achse.

   Zusätzlich zu der LAI-Achsplatz-Definition muss die Lead-Link-Achse noch in jedem Kanal, in dem sie zusammen mit der Folge-Achse verwendet werden soll, als Kanal-Achse ($MC_AXCONF_MACHAX_USED) definiert werden und reduziert auch hier die max. Anzahl der möglichen Kanal-Achsen.

Maschinendaten für NCU1

Leitachse verfahrende NCU
$MN_NCU_LINKNO = 1 ; Master-Ncu
$MN_MM_NCU_LINK_MASK = 1 ; NCU-Link aktiv
$MN_MM_LINK_NUM_OF_MODULES= 2 ; Anzahl der Link-Module
$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"
$MA_AXCONF_ASSIGN_MASTER_NCK[AX3] = 1
$MN_AXCONF_MACHAX_NAME_TAB[0] = "XM1"
$MN_AXCONF_MACHAX_NAME_TAB[2] = "YM1"
CHANDATA(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 verfahrende NCU(s)
$MN_NCU_LINKNO = 2 ; NCU-Nummer auf 2 setzen
$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 Lagereglung (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
CHANDATA(1)
$MC_AXCONF_MACHAX_USED[0]=3 ; X
$MC_AXCONF_MACHAX_USED[1]=2 ; Y
$MC_AXCONF_MACHAX_USED[2]=1 ; Z ; Zuordnung zu LAI AX1
; bzw. NCU1/AX3

2.15.9.2 Programmierung

Programm auf NCU 1

NCU1 verfährt die Leitachse Z. Solange NCU2 mit Bewegungen der Leitachse rechnen muss (Mitteilung über Link-Variable $A_DLB[0]), ist die Variable 1, nach Bewegungsabschluss 0.

```
N3000 R1 = 1 ; Zähler für Bewegungsschleife
N3004 G1 Z0 F1000
N3005 $A_DLB[0] = 1 ; Starten an NCU1
LOOP30:
N3005 R1=R1+1
N3006 G91 Z0.01 ; Jetzt wird die Leitwert-Achse verfahren
N3008 Z0.02
N3010 Z0.03
N3012 IF R1 < 10 GOTOB LOOP30
N3098 $A_DLB[0] = 0 ; Beenden an NCU1
N3099 GOTOF TESTE
```

NC-Programm auf NCU2

Das Programm stellt durch eine Kurventabelle einen Zusammenhang zwischen Leitachsbewegung auf NCU1 und Folgeachsbewegung auf NCU2 her. Wenn die Tabelle definiert ist geht die NCU2 in Wartestellung bis NCU1 die Leitachse startet. Dann wird die Kopplung aktiviert und aufrecht erhalten, bis die Leitachsbewegung beendet ist.

```
N2800 CTABDEL(1)
N2801 G04 F.1
N2803 G0 Y0 Z0
;*****************************************************
; Tabelle 1 erstellen
;*****************************************************
N2802 CTABDEF(Y, Z, 1, 0)
N2803 G1 X0 Y0
N2804 G1 X100 Y200
N2805 CTABEND
LOOP29:
N2806 IF ($A_DLB[0]== 0) GOTOB LOOP29 ; Warten auf NCU1
N2810 LEADON(Y,Z,1)
LOOP292: ; Kopplung aktiv!!!
N2830 IF ($A_DLB[0] > 0) GOTOB LOOP292 ; Kopplung halten bis
```

;NCU1 die Leitwert-Achse nicht mehr verfährt

N2890 LEADOF(Y,Z)

2.15.10 NCU-Link mit unterschiedlichem Interpolationstakt

2.15.10.1 Beispiel Unrund-Drehen

Aufgabenstellung

Es soll eine Ovalität mit folgenden Eigenschaften erreicht werden:

Ovalität: 0.2 mm

Grundkreisdurchmesser: 50 mm

Z-Weg pro Umdrehung: 0.1 mm

Spindeldrehzahl: 3000 Upm

Für die geforderte Genauigkeit soll eine Sinus-Approximation durch ein kubisches Polynom pro 45 Grad Spindeldrehung ausreichend sein.

Hinweis

Es können Polynome bis 5. Grades benutzt werden. S. Programmieranleitung Arbeitsvorbereitung.

Das ausschnittweise dargestellte Teileprogramm läuft in einem Kanal der NCU mit dem schnelleren Interpolationstakt, an dem die X-Achse lokale Achse und die C- und Z-Achse als Link-Achsen projektiert sind.

Für die Bestimmung der Polynomkoeffizienten werden CAD-Systeme verwendet, die Koeffizienten aus Konturpunkten und dem gewünschten Grad des Polynoms berechnen.

Das folgende Teileprogramm beschreibt die notwendigen Befehle für die erste Spindelumdrehung. Es muss dann für den gesamten geforderten Z-Weg entsprechend fortgesetzt werden:

```
G0 C0 X24.95 Z0 ; Startposition
FGROUP(C) ; sorgt für eine konstante Spindeldrehzahl
G1 G642 F1080000 ; Spindeldrehzahl 3000 UPM
POLY ;Vorgabe von Polynomen
C=DC(45.0000000) PO[X]=(25.0,.0750000,-0.0250000)
PO[Z]=(.2125000,0,0)
;1/8-Kreis, lineare Z-Bewegung, 1/4-Sinus in X
C=DC(90.0000000) PO[X]=(25.0500000,0,-0.0250000)
PO[Z]=(.2250000,0,0)
C=DC(135.0000000) PO[X]=(25.0,-0.0750000,.0250000)
PO[Z]=(.2375000,0,0)
C=DC(180.0000000) PO[X]=(24.9500000,0,.0250000) PO[Z]=(.2500000,0,0)
```

```
C=DC(225.0000000) PO[X]=(25.0,.0750000,-0.0250000)
PO[Z]=(.2625000,0,0)
C=DC(270.0000000) PO[X]=(25.0500000,0,-0.0250000)
PO[Z]=(.2750000,0,0)
C=DC(315.0000000) PO[X]=(25.0,-0.0750000,.0250000)
PO[Z]=(.2875000,0,0)
C=DC(0) PO[X]=(24.9500000,0,.0250000) PO[Z]=(.3000000,0,0)
...
```

Maschinendaten

Hinweise zur Projektierung der Maschinendaten finden Sie in "Konfiguration von Link-Achsen", "Achscontainer".

2.16 Datenlisten

2.16.1 Maschinendaten

2.16.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
10002	AXCONF_LOGIC_MACHAX_TAB[n]	Logisches NCU-Maschinenachsenabbild
10065	POSCTRL_DESVAL_DELAY	Lagesollwert-Verzögerung
10087	SERVO_FIFO_SIZE	Größe des Datenpuffers zwischen Interpolations- und Lagereglertask (bis SW 5, dann MD18720 s. u.)
10134	MM_NUM_MMC_UNITS	Anzahl gleichzeitig möglicher MMC-Kommunikationspartner
11398	AXIS_VAR_SERVER_SENSITIVE	Verhalten des AXIS-VAR-Servers im Fehlerfall
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 NCU-Link-Module
12540	LINK_BAUDRATE_SWITCH	Link Bus Baudrate
12550	LINK_RETRY_CTR	Maximale Anzahl der Telegrammwiederholungen im Fehlerfall
12701 ... 12716	AXCT_AXCONF_ASSIGN_TAB1[s] ... AXCT_AXCONF_ASSIGN_TAB16[s]	Liste der Achsen im Achscontainer
12750	AXCT_NAME_TAB[n]	Liste der Achscontainer-Namen
18700	MM_SIZEOF_LINKVAR_DATA	Größe des NCU-Link-Variablen-Speichers
18720	MM_SERVO_FIFO_SIZE	Größe des Datenpuffers zwischen Interpolations- und Lagereglertask
18780	MM_NCU_LINK_MASK	Aktivierung NCU-Link-Kommunikation

2.16.1.2 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

2.16.1.3 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

2.16.2 Settingdaten

2.16.2.1 Allgemeine Settingdaten

Nummer	Bezeichner. $SA	Beschreibung
41700	AXCT_SWWIDTH[Containernummer]	Achscontainer-Drehungsvorgabe

2.16.2.2 Achs-/Spindel-spezifische Settingdaten

Nummer	Bezeichner: $SA_	Beschreibung
43300	ASSIGN_FEED_PER_REV_SOURCE	Umdrehungsvorschub für Positionierachsen/ Spindeln

2.16.3 Signale

2.16.3.1 Signale von NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
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	DB2700.DBX2.3

2.16.3.2 Signale von HMI/PLC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
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-Busadresse)	DB19.DBB104	-
PAR_MMC_TYP Typ des HMI gemäß NETNAMES.INI: Haupt-/Nebenbedienfeld/Alarmserver	DB19.DBB106	-
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	-

2.16.3.3 Allgemein Online-Schnittstelle

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
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 wechselweise ihre Anforderungen/Quittungen.	DB19.DBB124	-
MMC1_Z_INFO Zusatz-Info Verbindungszustand (pos./neg. Quittung, Fehlermeldungen, ...)	DB19.DBB125	-
MMC1_SHIFT_LOCK HMI-Umschaltsperre	DB19.DBX126.0	-

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
MMC1_MSTT_SHIFT_LOCK MCP-Umschaltsperre	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	-
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 wechselweise ihre Anforderungen/ Quittungen.	DB19.DBB134	-
MMC2_Z_INFO Zusatz-Info Verbindungszustand (pos./neg. Quittung, Fehlermeldungen, ...)	DB19.DBB135	-
MMC2_SHIFT_LOCK HMI-Umschaltsperre	DB19.DBX136.0	-
MMC2_MSTT_SHIFT_LOCK MCP-Umschaltsperre	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	-

2.16.3.4 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
NCU-Link Achse aktiv	DB31, ...DBX60.1	-
Achscontainer-Drehung aktiv	DB31, ...DBX61.1	DB390x.DBX1.1
Achse betriebsbereit	DB31, ...DBX61.2	DB390x.DBX1.2

B4: Bedienung über PG/PC - nur 840D sl

3

3.1 Kurzbeschreibung

Anwendungen

Die Bedienung über PG/PC

- muss eingesetzt werden, wenn keine Bedientafelfront vorgesehen ist.
- kann eingesetzt werden bei OP030 zur Unterstützung des Handlings,

Hardware

Folgende HW-Voraussetzungen müssen gegeben sein:

- PG/PC mit mindestens 486DX33-Prozessor und 8 MB Arbeitsspeicher
- MS-Windows muss im ENHANCED-Mode (386-Mode) laufen
- PG/PC mit MPI/BTSS-Schnittstelle (bei PG 720/720C/740/760 vorhanden).
  Für PC mit freiem ISA-Steckplatz ist eine MPI-Karte (6ES7793-2AA00-0AA0) lieferbar.
- VGA-Monitor mit einer Auflösung 640x480 oder besser.

Realisierung Variante 1

Die Maschinensteuertafel (MSTT) und die Bedientafel OP030 sind fest der NCU zugeordnet.

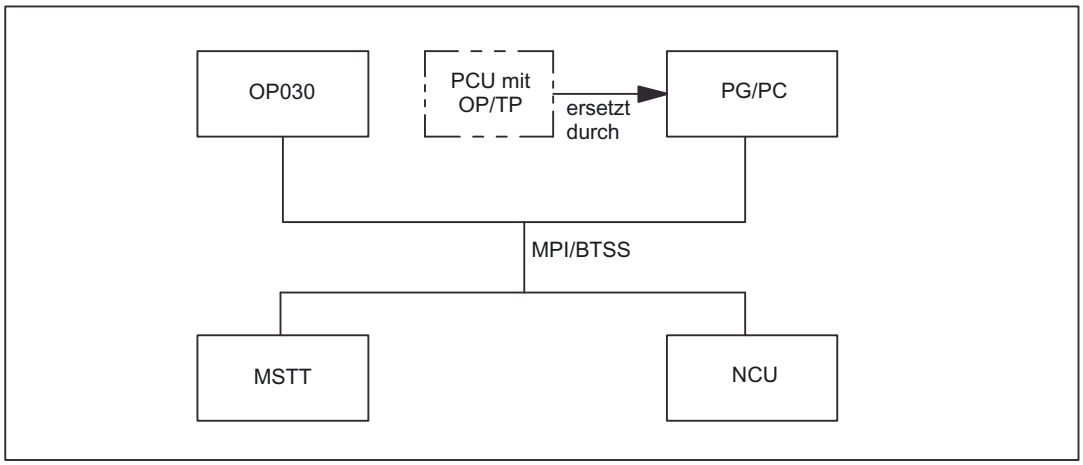


Bild 3-1 Konfiguration mit OP030 und PG/PC

Maschinensteuertafel, OP030 und NCU hängen entweder alle am BTSS-Bus oder alle am MPI-Bus. Es muss bezüglich dieser Komponenten ein **homogenes** Netz vorliegen.

Realisierung Variante 2

Bedientafelfront und bis zu drei NCUs. Die Maschinensteuertafel ist dabei fest der jeweiligen NCU zugeordnet.

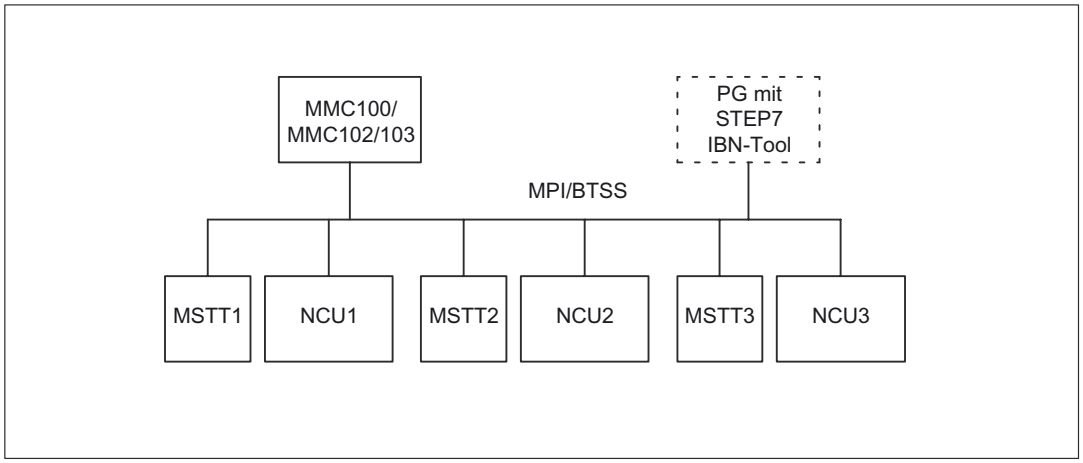


Bild 3-2 Konfiguration m:n entspricht 1:3

Literatur:

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafelfronten und NCUs (B3)

Bedienoberflächen

Die Bedienoberflächen sind in den Bedienungsanleitungen zu den verwendeten Bedientafelfronten beschrieben.

Literatur:

/BA/ Bedienungsanleitung

/FBO/Projektierung Bedienoberfläche OP 030; BA, Bedienungsanleitung OP030

Einschränkungen

Wenn die Bedienung vom PG/PC zusätzlich zu einer Bedientafelfront OP030 benutzt wird, müssen die in der Funktionsbeschreibung /FB2/, B3, "Mehrere Bedientafelfronten und NCUs" beschriebenen Bedingungen bezüglich Konfiguration und Koordination beachtet werden.

Literatur:

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafelfronten und NCUs (B3)

3.2 Software-Installation

3.2.1 Systemvoraussetzungen

Hardwareanforderungen

Die Bedienung über PG/PC benötigt folgende Hardware-Anforderungen:

- IBM® AT-kompatibler PG/PC mit 486DX33-Mikroprozessor
- Hauptspeicher mindestens 8 MB
- Diskettenlaufwerk (3 ½ Zoll)
- Festplattenlaufwerk für die Datenhaltung
- Monochrom- bzw. Farb-Monitor
- Tastatur
- PG/PC mit MPI-Schnittstelle (bei PG 720/720C/740/760 vorhanden)

  Ein eingeschränkter Betrieb ohne MPI-Karte ist möglich (z. B.: interaktiver Bedienung).

  Hinweis: RS232-MPI-Adapter wird nicht unterstützt.
- Maus
- Verbindungskabel für Kopplung zwischen PG/PC und NCU-Modul

Hinweis

Die Bedientafelfronten und die NCU hängen entweder

- alle am BTSS-Bus oder
- alle am MPI-Bus.

Es muss bezüglich dieser Komponenten ein **homogenes** Netz vorliegen.

Softwareanforderungen

Software-Konfiguration für die Bedienung über PG/PC:

- Betriebssystem MS-DOS® ab Version 6.x
- WINDOWS™-Bedienoberfläche ab Version 3.1
- Schnittstellentreiber MPI (ist in der mitgelieferten SW enthalten).
- WINDOWS™ 32s, ab Version 1.30.166.0

  (Die aktuelle Version können Sie unter "windows\system\win32s.ini" nachschauen.)

  Falls WINDOWS™ 32s nicht installiert ist, kann es über 2 mitgelieferte Disketten installiert werden (setup.exe aufrufen).

3.2.2 Installation

Speicherbereich MPI-Karte

Der Speicherbereich der MPI-Karte muss von der Nutzung durch Speichermanager ausgeschlossen werden (Dateien: CONFIG.SYS, SYSTEM.INI).

Beispiel für den Eintrag in die SYSTEM.INI:

[386enh]

EmmExclude=....<Bereich der Karte>

(siehe HW-Beschreibung der Karte)

Lieferumfang

System-Software:

- ca.10 Disketten mit komprimierter MMC102/103-SW und Installationstools
- 2 Disketten mit WINDOWS 32s Subsystem (= Microsoft Setup)

Um die Software zu installieren, ist wie folgt vorzugehen:

Aufruf

1. SETUP.EXE starten

Erste Installationsdiskette einlegen und mittels dem WINDOWS™-Dateimanager die Datei SETUP.EXE starten.

Das Installationsprogramm fordert alle weiter benötigten Eingaben bzw. Diskettenwechsel im Benutzerdialog an.

2. Installationspfad eingeben

Wählen Sie das Verzeichnis samt Installationspfad aus (siehe Bild), in das die SW kopiert werden soll.

Mit "**Continue**" setzen Sie die Installation fort, mit "**Exit Setup**" brechen Sie die Installation ab.

Dies gilt auch für die weiteren Schritte.

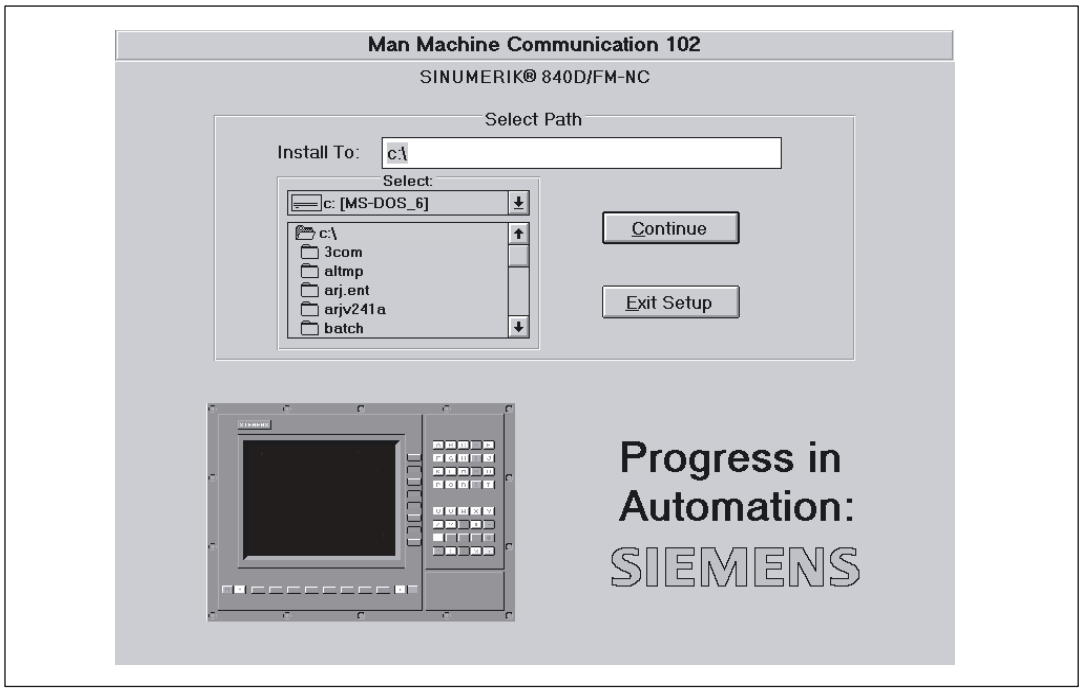


Bild 3-3 Installationspfad eingeben

3. Betrieb mit MPI/ohne MPI auswählen



Bild 3-4 Betrieb mit MPI/ohne MPI

4. Drehen (turn) oder Fräsen (mill) auswählen



Bild 3-5 Drehen/Fräsen auswählen

Hinweis

Wenn Sie später Ihre Auswahl ändern wollen, wählen Sie im Installationspfad das Verzeichnis "mmc2" und kopieren dort "dpturn.exe" (Drehen) bzw. "dpmill.exe" (Fräsen) ins Verzeichnis "dp.exe".

5. Laufwerk auswählen

Nur wenn mehrere lokale Plattenlaufwerke zur Verfügung stehen.

Wählen Sie das Laufwerk für das tmp-Verzeichnis aus (siehe Bild)

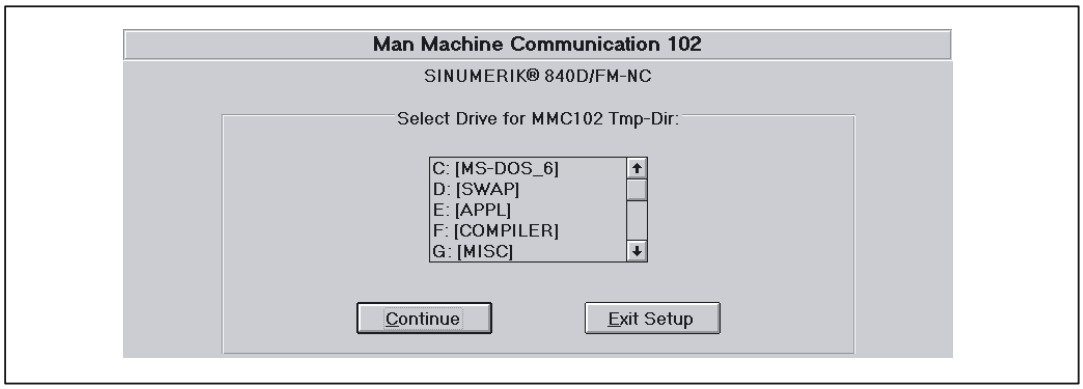


Bild 3-6 Laufwerk auswählen

Ansonsten wird Laufwerk C:\ ausgewählt.

Hinweis

Bei jedem Neustart von MMC 102 wird der Inhalt des Verzeichnisses "tmp" auf dem Installationslaufwerk gelöscht.

Nach der Anwahl erscheint eine Statusanzeige mit den von Ihnen vorgenommenen Eingaben.

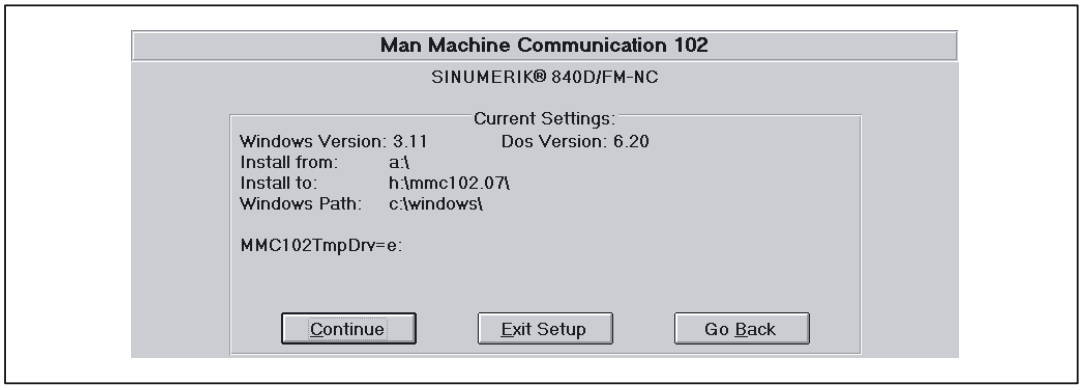


Bild 3-7 Statusanzeige der erfolgten Installation

6. Continue

Betätigen Sie **Continue**, es werden die Installationsdisketten angefordert.

Hinweis

Beachten Sie bitte die Aufforderungen am Bildschirm.

Die Programmgruppe "SINUMERIK 840D MMC V3.2" wird erzeugt.

Bei erfolgreicher Installation erscheint die Meldung:

"MMC 102 Installation is complete"

Falls Sie den Installationspfad ändern wollen, betätigen Sie **Go back**.

7. Einstellungen vornehmen

7a:

Schnittstelle **BTSS** (1,5 MBaud), Konfiguration: **1 MMC** an **1 NCU** (Lieferform)

Zusätzliche Einstellungen sind nicht erforderlich.

7b:

Schnittstelle **MPI** (187,5 kBaud), Konfiguration: **1 MMC** an **1 NCU** (Lieferform)

1. Feststellen der NCK-/PLC-Busadresse
   - wenn PLC < SW 3.2,

     dann NC-Adresse = 13

     PLC-Adresse = 2
   - wenn PLC ≥ SW 3.2 und Baugruppe PLC 314,

     dann NC-Adresse = 13

     PLC-Adresse = 2
   - wenn PLC ≥ SW 3.2 und Baugruppe PLC 315,

     dann NC-Adresse = 3

     PLC-Adresse = 2
2. Eintrag der Adressen in Dateien
   - Datei "S7CFGPGX.DAT"

     In der Datei "S7CFGPGX.DAT" auf dem MPI-Treiber-Verzeichnis (<Installationspfad>\MMC2\DRV.ID) müssen mit einem ASCII-Editor folgende Einträge an die vorliegende Hardware-Konfiguration angepasst werden:

     # Einstellung des Interrupts

     "hwint_vector": Einstellung des Interrupts für die MPI-Karte. Dieser Interrupt darf von keiner anderen Karte (z. B. Netzwerkadapter) verwendet werden.

     Voreinstellung: 10.

     # Einstellungen für die Baudrate

     "baudrate", "tslot" und "tgap": Einstellungen für die Baudrate. Diese 3 Einstellungen müssen immer gemeinsam durch Entfernen/Voranstellen des führenden "#" (Kommentar) aktiviert/deaktiviert werden.

     Bei Änderung der Baudrate muss zusätzlich in der Datei Installationspfad>\MMC2\MMC.ini, Sektion [840D] die Einstellung "ADDRESS1=\PLC, 10000d01" für 1,5 MBaud bzw. "ADDRESS1=\PLC, 10000201" für 187,5 kBaud angepasst werden.

     Voreinstellung: 1,5 MBaud.
   - Datei "netnames.ini"

     Folgende Zeilen in der Datei müssen geändert werden:

     # bus = btss ersetzen durch = **mpi**

     # nck_address = 13 ersetzen durch = **3** (wenn PLC ≥ SW3.2)

     = 13 (wenn PLC < SW3.2)

     # plc_address = 13 ersetzen durch = **2**

Parallele Step7/AS300-Applikation

Eine parallele Installation mit der Step7/AS300-SW kann zu Problemen führen. Evtl. sind eine Umkonfigurierung der Treiber und ein Neustart nötig.

3.2.3 SW-Randbedingungen

- **Funktionstasten**

  In allen Bildern dürfen Funktionstasten erst betätigt werden, nachdem der Bildaufbau beendet ist.

- **Monochrombildschirm**

  Bei Verwendung eines Monochrombildschirms müssen die von MMC verwendeten Farben entsprechend angepasst werden. Dazu stellen Sie im Bild "Inbetriebnahme\MMC\Farbeinstellung" das Farbschema "Monochrom" oder "Mono positiv" ein.

- **Komfortable Parametrierung**

  Das Bild "Inbetriebnahme\MMC\BTSS-Parameter" kann ab jetzt auch dann aufgerufen werden, wenn keine Verbindung zum NC-Kernel besteht. Dadurch können die BTSS-Parameter für Baudrate und Netzadresse komfortabel eingestellt werden.

3.2.4 Programm starten

Programmaufruf

Auf einem PG/PC wird die MMC102/103-SW entweder

- vom Programmmanager aus durch Anwahl der Programmgruppe "SINUMERIK 840D MMC V2.3" und danach durch **Doppelklicken** des Symbols **"MMC Startup"** gestartet oder

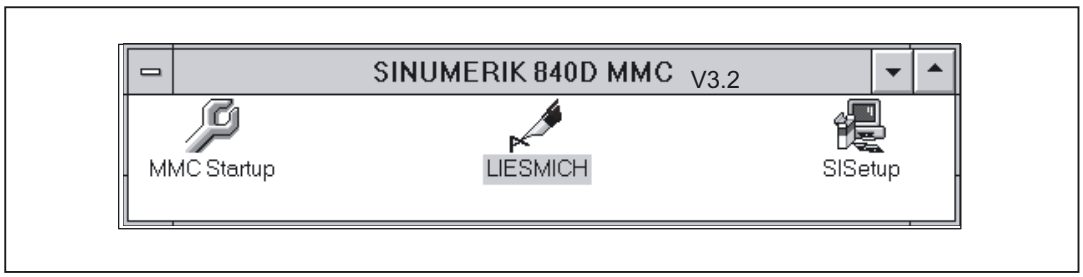


Bild 3-8 Programmgruppe SINUMERIK 840D MMC

- vom Dateimanager aus durch **Doppelklicken** der Datei **REG_CMD.EXE**.

Kommunikation

Kann keine Kommunikation zum NCK bzw. 611D stattfinden, so erscheint die Meldung "Keine Kommunikation zu NCK". Wird die Kommunikation unterbrochen, z. B. durch NCK-Reset, so versucht die MMC102/103-SW selbständig den Wiederaufbau der Verbindung.

3.2.5 Programm beenden

Programm abwählen

Die MMC102/103-SW wird durch folgende Schritte abgewählt:

1. Funktionstaste **F10** betätigen
   Es wird eine horizontale Softkeyleiste eingeblendet.
2. Funktionstaste **Shift + F9** betätigen.
3. Mit Anwahl des Softkeys **Exit** beenden Sie das Programm.

3.3 Bedienung über PG/PC

3.3.1 Allgemeine Bedienung

Bedienphilosophie

Die speziellen Funktionstasten der Bedientastatur können mit der Volltastatur benutzt werden. Die Bedienung ist mausgesteuert oder per Tastatur möglich.

Tastaturbedienung

Die folgende Tabelle zeigt, auf welchen Tasten die horizontalen/vertikalen Softkeys und die Sondertasten abgebildet sind:

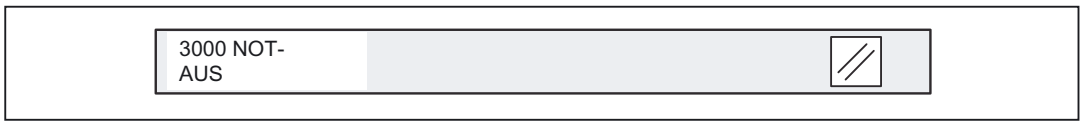
Hinweis

Im Editor werden nur die über die Bedientafelfronttastatur eingebbaren Zeichen angezeigt.

Volltastatur	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
mit SHIFT	vertik Soft. 1	vertik Soft. 2	vertik Soft. 3	vertik Soft. 4	vertik Soft. 5	vertik Soft. 6	vertik Soft. 7	vertik Soft. 8				
ohne SHIFT	horiz Soft. 1	horiz Soft. 2	horiz Soft. 3	horiz Soft. 4	horiz Soft. 5	horiz Soft. 6	horiz Soft. 7	horiz Soft. 8				
Volltastatur	5	Esc	Insert	Home	Page Up	Page Down	Enter					
ohne SHIFT												

Bild 3-9 Entsprechung der Bedientastatur und Volltastatur

Alarm- bzw. Meldungszeile



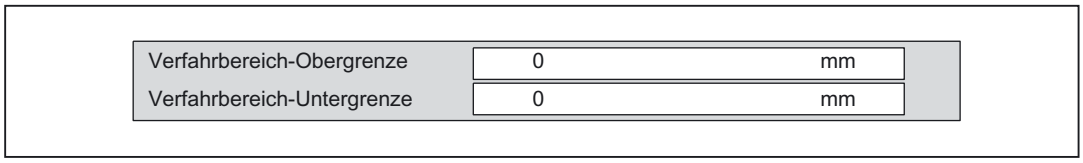
Alarm- bzw. Meldungszeile zur Anzeige von Hinweisen für den Bediener

i-R

Die Anwahlfelder **i** und **R**, die in jedem Bild angeboten werden, haben folgende Bedeutung:

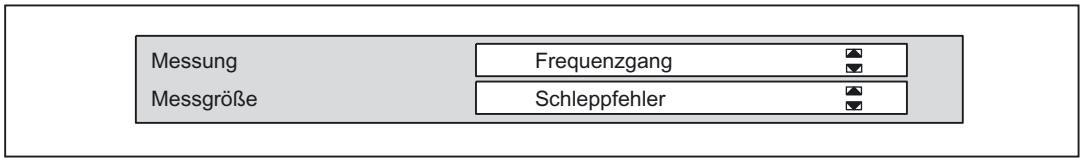
- Das **i**-Feld wird mit der **Help**-Taste bzw. mit **Maus-Klick** angewählt.
- Das **R**-Feld wird mit der **F9**-Taste bzw. mit **Maus-Klick** angewählt. Bei Anwahl erfolgt die Recall-Funktion (Rücksprung) in die vorangehende Ebene.

Eingabefelder



Um eine Eingabe zu ermöglichen, wird der Eingabe-Cursor mit den Tasten **TAB** bzw. **SHIFT + TAB** oder **Maus-Klick** im entsprechenden Eingabefeld platziert. Der Editier-Modus ist immer auf **Überschreiben** voreingestellt. Mit der **Insert**-Taste kann zwischen Überschreib-Modus und Einfüge-Modus hin- und hergeschaltet werden.

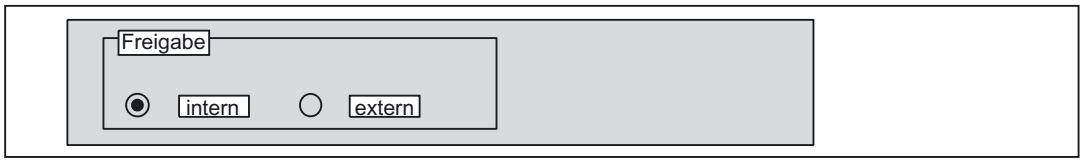
Listenfelder



Die angebotenen Funktionen werden mit den Cursor-Tasten **UP (")** und **DOWN (#)** bzw. **Maus-Klick** ausgewählt. Die angezeigte Funktion ist gültig.

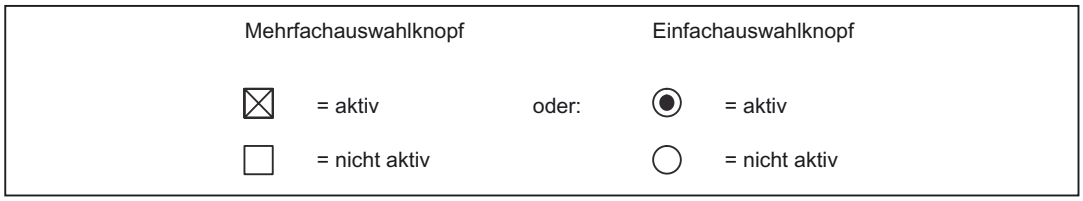
Die Anwahl der Listenfelder erfolgt mit den Tasten **TAB** bzw. **SHIFT + TAB** oder **Maus-Klick**.

Einfach-/Mehrfachauswahlknopf



Die gewünschte Funktion wird mit den Cursor-Tasten **LEFT (z)** und **RIGHT (!)** bzw. **Maus-Klick** aktiviert.

Die Anwahl der Funktionsfelder erfolgt mit den Tasten **TAB** bzw. **SHIFT + TAB** oder **Maus-Klick**.



Aktivieren der Felder

Zur Änderung von Werten und Funktionen muss das Fenster mit dem Eingabefeld über die Tasten **CTRL + TAB** bzw. die Taste **HOME** aktiviert werden (gelber Rahmen = Fokus).

3.3.2 Zusatzinformationen

Achsauswahl

Die Achsweiterschaltung/-auswahl in achsspezifischen Bildern erfolgt stets über die einheitlich angeordneten vertikalen Softkeys **ACHSE+** bzw. **ACHSE-**.

Funktionsanwahl/-abwahl

Sämtliche Funktionen werden mit dem Softkey **START** aktiviert und mit dem Softkey **STOP** beendet.

Kennwort

Mit dem Softkey **Kennwort setzen** erscheint eine Dialogbox mit der Aufforderung zur Eingabe des Kennwortes. Die Kennworteingabe erfolgt wie beschrieben in:

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; Diverse Nahtstellensignale (A2)

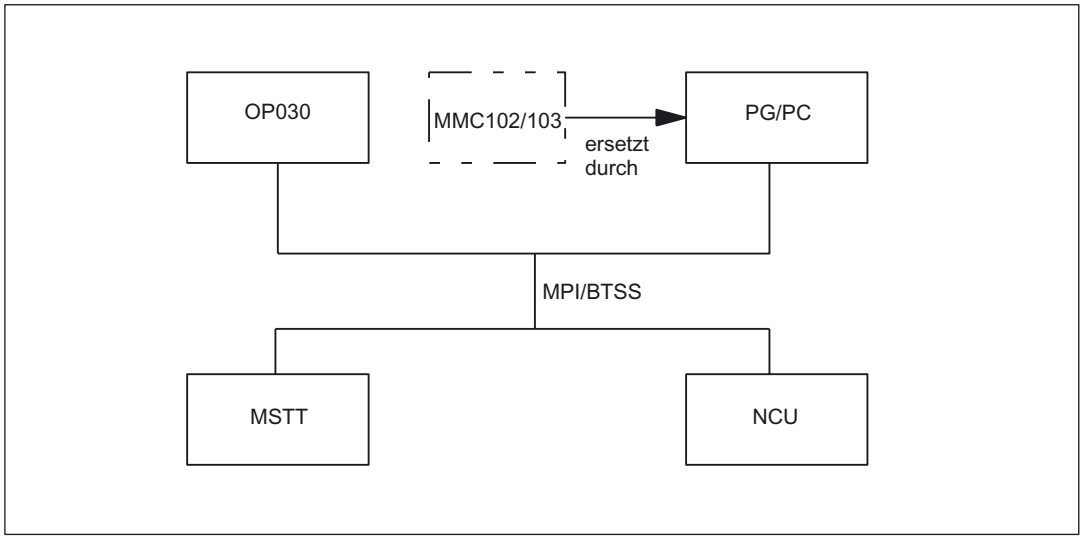
Tastaturbelegung

Bis auf die Belegung der Tasten **F1** bis **F12** bzw. **SHIFT + F1** bis **F10** gelten die Bedingungen und Tastenbelegungen wie unter WINDOWS™ 3.1.

Mit der Tastenkombination **ALT + TAB** kann jederzeit aus der Bedienung von PG/PC in andere WINDOWS™-Applikationen umgeschaltet werden.

3.3.3 Betrieb von Bedientafelfronten

Beim Betrieb mit z. B. zwei Bedientafelfronten in der skizzierten Konfiguration ist folgendes Verhalten zu beachten:



1. Die Eingaben von der Bedientafelfront MMC bzw. OP030 sind gleichwertig gegenüber der NCU.
2. Jede Bedienungseinheit bekommt unabhängig von der anderen die Anzeigen zu sehen, die sie sich anwählt.
3. Spontane Ereignisse wie Alarme werden an beiden Bedieneinheiten angezeigt.
4. Es gilt die Schutzstufe mit der höchsten Berechtigung entsprechend der kleinsten aktivierten Schutzstufen-Nummer für beide Bedientafelfronten.
5. Das System leistet keine weiterführenden Koordinierungen der Bedieneinheiten.

Weitere Informationen erhalten Sie in

Literatur:

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafelfronten und NCUs (B3)

3.4 Simulation von Teileprogrammen

Für einen Betrieb der Teileprogrammsimulation ist ein installiertes Windows 32s ab Version 1.30.166.0 erforderlich.

Bedienung siehe

Literatur:

/BA/ Bedienungsanleitung

3.5 Randbedingungen

Die Funktion "Bedienung über PG/PC" ist ab SW-Stand 3.1 in der Grundausführung verfügbar. Bei SW-Stand 3.1 ist die Anzahl der anschließbaren NCUs auf 1 beschränkt und die Anzahl der Bedientafelfronten auf zwei. Von diesen muss eine OP030 sein.

Ab SW-Stand 3.2 kann auch eine Bedientafelfront mit MMC 100 oder MMC 102/103 mit bis zu drei NCUs verbunden werden.

3.6 Datenlisten

Für die Funktion werden keine Signale oder Maschinendaten benötigt.

H1: Handfahren und Handradfahren

4

4.1 Kurzbeschreibung

4.1.1 Übersicht

Anwendungen

Auch bei modernen numerisch gesteuerten Werkzeugmaschinen müssen die Achsen vom Bediener von Hand verfahren werden können.

Maschine einrichten

Insbesondere beim Einrichten eines neuen Bearbeitungsprogramms ist ein Bewegen der Maschinenachsen mit Verfahrtasten auf der Maschinensteuertafel oder mit dem elektronischen Handrad erforderlich. Das Verfahren von Hand kann bei angewählter Koordinatenverschiebung oder -drehung auch im transformierten Werkstückkoordinatensystem erfolgen.

Freifahren des Werkzeugs

Nach Programmunterbrechung durch Ereignisse wie NC-STOP, RESET oder Netzausfall muss vom Maschinenbediener das Werkzeug manuell aus der aktuellen Bearbeitungsposition frei gefahren werden. Dies erfolgt in der Regel mit Hilfe der Verfahrtasten in der Betriebsart JOG. Dabei müssen die während der Bearbeitung gültigen Transformationen und Koordinatensysteme wirksam bleiben.

Varianten des manuellen Verfahrens

In der vorliegenden Funktionsbeschreibung werden folgende Varianten des manuellen Verfahrens aufgezeigt:

- Kontinuierliches Verfahren im Tipp- oder Dauerbetrieb in JOG
- Inkrementelles Verfahren (INC) im Tipp- oder Dauerbetrieb in JOG
- Verfahren von Achsen mit Hilfe von elektronischen Handrädern (Zubehör) in JOG
- Handradüberlagerung in AUTOMATIK (Wegvorgabe und Geschwindigkeitsüberlagerung)

DRF

Die "Differential-Resolver-Function" (DRF) erzeugt eine zusätzliche inkrementelle Nullpunktverschiebung im AUTOMATIK-Betrieb über das elektronische Handrad. Mit dieser Funktion kann z. B. der Werkzeugverschleiß innerhalb eines programmierten Satzes korrigiert werden.

Festpunktfahren

Die Funktion "Festpunktfahren in JOG" ermöglicht das manuelle Anfahren fester Achspositionen, die über ein Maschinendatum definiert werden.

4.1.2 Allgemeine Eigenschaften beim Handfahren in JOG

Nachfolgend werden die Eigenschaften beschrieben, die generell beim Handfahren in JOG (unabhängig der angewählten Variante) gültig sind:

Betriebsart JOG

Für das Verfahren von Achsen durch Handbedienung (nachfolgend Handfahren bezeichnet) muss die Betriebsart JOG aktiv sein.

Die jeweils wirksame Betriebsart wird an die PLC über das Nahtstellensignal:
DB11 DBX4.2 (aktive Betriebsart-Strobe: JOG)
gemeldet.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb (K1)

Maschinenfunktionen

Innerhalb der Betriebsart JOG unterscheidet man mehrere JOG-Varianten (die sog. Maschinenfunktionen):

- Kontinuierliches Verfahren (JOG-CONT)
- Inkrementelles Verfahren (JOG-INC)
- Verfahren mit dem Handrad

Handradverfahren

Mit folgenden Funktionen ist das Handradfahren außerdem aktiv:

- Betriebsart JOG-REPOS, zum Verfahren der Geometrie- und Maschinenachsen
- Betriebsart AUTOMATIK, zum Herausfahren einer DRF-Verschiebung
- bei der Wegüberlagerung
- bei der Verschiebung des Umkehrpunktes einer Pendelbewegung

Die Anwahl der jeweils wirkenden Maschinenfunktion erfolgt über die PLC-Nahtstelle. Dabei gibt es eine eigene PLC-Nahtstelle für die Maschinenachsen (achsspezifisch) und für die Geometrieachsen (kanalspezifisch).

Simultanes Verfahren

Es können bei JOG gleichzeitig alle Achsen verfahren werden. Bei simultaner Bewegung von mehreren Achsen besteht kein interpolatorischer Zusammenhang.

Geschwindigkeit

Die Geschwindigkeit der Verfahrbewegung bei JOG wird, abhängig von der Vorschubart, durch folgende Wertvorgaben festgelegt:

- bei Linearvorschub (G94) aktiv

  (SD41100 $SN_JOG_REV_IS_ACTIVE (JOG: Umdrehungs.- / Linearvorschub) = 0):

  – mit allgemeinem Settingdatum:
  SD41110 $SN_JOG_SET_VELO (Achsgeschwindigkeit bei JOG)
  bzw. bei Rundachsen mit dem allgemeinen Settingdatum:
  SD41130 $SN_JOG_ROT_AX_SET_VELO
  (JOG-Geschwindigkeit bei Rundachsen)

  oder

  – mit axialem Maschinendatum:
  MD32020 $MA_JOG_VELO (Konventionelle Achsgeschwindigkeit),
  nur falls SD41110 = 0.

- bei Umdrehungsvorschub (G95) aktiv
  (SD41100 $SN_JOG_REV_IS_ACTIVE = 1):

  – mit allgemeinem Settingdatum:
  SD41120 $SN_JOG_REV_SET_VELO
  (Umdrehungsvorschub der Achsen bei JOG)

  oder

  – mit axialem Maschinendatum:
  MD32050 $MD_JOG_REV_VELO (Umdrehungsvorschub bei JOG),
  nur falls SD41120 = 0.

Standardeinstellung für die Vorschubgeschwindigkeit ist mm/min bzw. bei Umdrehungsvorschub oder Rundachsen Umdr./min.

Eilgangüberlagerung

Wird zusätzlich mit den Verfahrtasten die Eilgangsüberlagerungstaste betätigt, so erfolgt die Bewegung mit der über das achsspezifische Maschinendatum festgelegten Eilgangsgeschwindigkeit (bzw. bei Umdrehungsvorschub mit MD32040 $MA_JOG_REV_VELO_RAPID):

MD32010 $MA_JOG_VELO_RAPID (Konventioneller Eilgang)

Vorschub-Korrektur

Die bei JOG verfahrene Geschwindigkeit kann zusätzlich mit Hilfe des axialen Vorschubkorrekturschalters beeinflusst werden, sofern das Nahtstellensignal:
DB31, ... DBX1.7 (axiale Vorschubkorrektur wirksam)
wirksam gesetzt ist.

Die Zuordnung der Prozentanteile zu den einzelnen Schalterstellungen des Vorschubkorrekturschalters erfolgt über Maschinendaten. Bei Schalterstellung 0% wird die Achse nicht verfahren, sofern im zugehörigen Maschinendatum 0 eingetragen ist.

Das Nahtstellensignal:
DB31, ... DBX1.7 (axiale Vorschubkorrektur wirksam)
hat bei Schalterstellung 0% keine Bedeutung.

Wahlweise kann von der PLC anstatt der Schalterstellung des Vorschubkorrekturschalters (Gray-Code) direkt der Prozentwert (0% bis 200%) vorgegeben werden. Die Auswahl erfolgt ebenfalls über Maschinendaten.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Vorschübe (V1)

Beschleunigung

Auch beim Handfahren erfolgt die Beschleunigung entsprechend einer vorgegebenen Kennlinie. Die bei JOG wirkende Beschleunigungskennlinie für die einzelne Achse wird mit dem axialen folgenden Maschinendatum festgelegt (Grundeinstellung der axialen Ruckbegrenzung):

MD32420 $MA_ JOG_AND_POS_JERK_ENABLE

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Beschleunigung (B2)

Anzeige

Bei Anwahl der Betriebsart JOG wird das JOG-Grundbild angezeigt. Dieses Grundbild enthält Positions-, Vorschub-, Spindel- und Werkzeugwerte.

Erläuterungen zu den einzelnen Anzeigewerten siehe:
Literatur:
/BA/ SINUMERIK 840D Bedienungsanleitung HMI

Koordinatensysteme

In der Betriebsart JOG hat der Bediener die Möglichkeit, Achsen in unterschiedlichen Koordinatensystemen zu verfahren.

Folgende Koordinatensysteme sind möglich:

- Basiskoordinatensystem

  Jede Achse manuell verfahrbar.
- Werkstückkoordinatensystem

  Nur Geometrieachsen manuell verfahrbar (kanalspezifisch).

Geometrieachsen

Es ist beim Handfahren zu unterscheiden, ob die betroffene Achse als Maschinenachse (achsspezifisch) oder als Geometrieachse (kanalspezifisch) verfahren werden soll.

Zunächst wird auf die Merkmale als Maschinenachse eingegangen. Die Besonderheiten beim manuellen Verfahren von Geometrieachsen sind unter dem Thema "Geometrieachsen beim Handfahren" näher beschrieben.

Handfahren der Spindel

In der Betriebsart JOG können auch die Spindeln manuell verfahren werden. Dabei gelten im Wesentlichen die gleichen Bedingungen wie beim Handfahren von Achsen. Die Spindeln können bei JOG über Verfahrtasten kontinuierlich bzw. inkrementell sowohl im Tipp- als

auch im Dauerbetrieb oder mit dem Handrad verfahren werden. Die Anwahl und Aktivierung erfolgt über die achs-/spindelspezifische PLC-Nahtstelle analog den Achsen. Ebenso wirken die achsspezifischen Maschinendaten auch für die Spindeln. Die Besonderheiten beim manuellen Verfahren von Spindeln sind unter dem Thema "Besonderheiten beim Handfahren der Spindel" näher beschrieben.

4.1.3 Steuerung der Handfahr-Funktionen über PLC-Nahtstelle

HMI/NCK/PLC-Nahtstelle

Die Aktivierung der einzelnen Funktionen beim Handfahren in JOG erfolgt überwiegend über die PLC-Anwendernahtstelle. Der Maschinenhersteller kann mit Hilfe des PLC-Anwenderprogramms die Handfahr-Funktionalität abhängig vom Ausbaugrad an die jeweilige Werkzeugmaschine anpassen.

Maschinensteuertafel

Die Signale zwischen der Maschinensteuertafel und den einzelnen PLC/NCK-Nahtstellen-Datenbausteinen können maschinenspezifisch vom PLC-Anwenderprogramm transferiert werden. Mit dem PLC-Anwenderprogramm wird u. a. die Zuordnung zwischen den Richtungstasten der MSTT und den Verfahrtasten der Achsen/Spindeln (Maschinenachsen, Geometrieachsen) festgelegt.

Für das Handfahren sind insbesondere folgende Signale der MSTT relevant:

- Betriebsart JOG (Anwahl)
- Maschinenfunktion INC1 ...
- Richtungstasten
- Vorschubkorrektur bzw. Spindelkorrektur

Weitere Informationen bzgl. Signalübertragung der MSTT siehe:

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; PLC-Grundprogramm (P3)

Anwahl der Maschinenfunktion

Die in der Betriebsart JOG möglichen Maschinenfunktionen können von folgenden Stellen angewählt werden:

über Maschinensteuertafel (MSTT)	→ z. B. Anwender-DB-Nahtstelle
über PLC-Anwenderprogramm	→ PLC/NCK-Nahtstelle

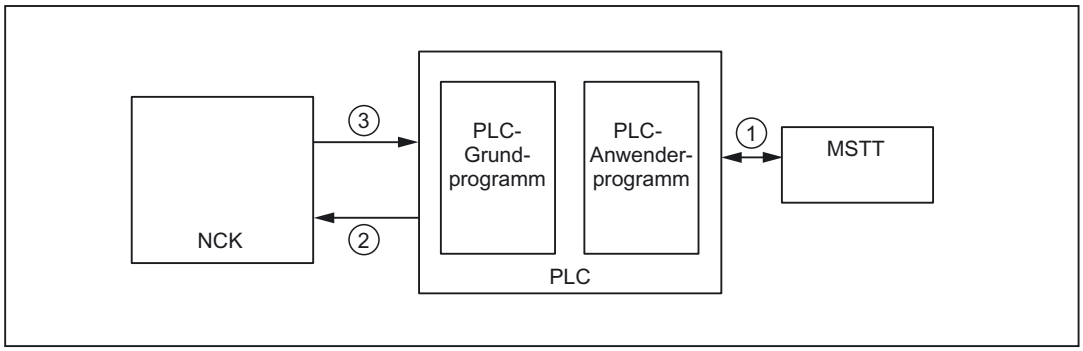
Vom PLC-Anwenderprogramm ist die an der MSTT-Nahtstelle anstehende Maschinenfunktion auf die entsprechende PLC/NCK-Nahtstelle umzusetzen. Dabei ist bei einer Maschinenachse/Spindel die achsspezifische NCK/PLC-Nahtstelle und bei einer Geometrieachse die kanalspezifische NCK/PLC-Nahtstelle zu verwenden.

4.1.4 Steuerungsverhalten bei Power On, Betriebsartenwechsel, Reset, Satzsuchlauf, Repos

Ein RESET führt immer zu einem Abbruch mit Bremsrampe einer durch Handradfahren ausgelösten Verfahrbewegung.

Anwahl von MSTT

Im nachfolgenden Beispiel wird der Ablauf für die Anwahl der Maschinenfunktion "kontinuierlich" für eine Maschinenachse der Maschinensteuertafel aufgezeigt.



① Der Bediener wählt an der Maschinensteuertafel für eine Maschinenachse die Maschinenfunktion "JOG-kontinuierlich" an.

② NST "Maschinenfunktion"

Das PLC-Programm (Grund- oder Anwenderprogramm) verknüpft dieses NST und gibt die Anforderung an die NCK-Nahtstelle:
DB31, ... DBX5.6. (Maschinenfunktion kontinuierlich)

Zuvor wird vom PLC-Anwenderprogramm abhängig vom momentanen Maschinenzustand überprüft, ob die Anforderung zulässig ist.

③ NST "aktive Maschinenfunktion"

Steuerungsintern wird die Anwahl der Maschinenfunktion vorgenommen.

Sobald die Maschinenfunktion "JOG-kontinuierlich" (DB31, ... DBX65.6) aktiv ist, wird dies von dem NCK an die PLC zurückgemeldet.

Bild 4-1 Ablauf bei Anwahl der Maschinenfunktionen von MSTT

Erläuterungen zur Signalübertragung zwischen der MSTT und der PLC siehe:
Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; PLC-Grundprogramm (P3)

4.2 Kontinuierliches Verfahren

4.2.1 Allgemeine Funktionalität

Anwahl

In der Betriebsart JOG muss das kontinuierliche Verfahren über die PLC-Nahtstelle aktiviert werden:
DB21, ... DBX13.6, ff (Maschinenfunktion: kontinuierlich)

Sobald das kontinuierliche Verfahren wirksam ist, wird das an die PLC mit dem Nahtstellensignal:
DB21, … DBX41.6, ff (aktive Maschinenfunktion: kontinuierlich)
zurückgemeldet.

Verfahrtasten +/-

Mit den Verfahrtasten Plus und Minus wird die zugehörige Achse in die entsprechende Richtung verfahren. Werden beide Verfahrtasten gleichzeitig betätigt, so erfolgt keine Verfahrbewegung bzw. wird eine in Bewegung befindliche Achse gestoppt.

Hinweis

Nach dem Einschalten der Steuerung können Achsen bis in den Grenzbereich der Maschine bewegt werden, da Referenzpunkte noch nicht angefahren sind. Dabei können Notendschalter ausgelöst werden.

Die Software-Endschalter und die Arbeitsfeldbegrenzung sind nicht wirksam.

Fahrbefehl +/-

Sobald eine Fahranforderung für eine Achse ansteht (z. B. mit Betätigung der Verfahrtaste), wird an die PLC das Nahtstellensignal:
DB21, ... DBX40.7 (Fahrbefehl+)
bzw.
DB21, … DBX40.6 (Fahrbefehl-)
abhängig von der Bewegungsrichtung ausgegeben.

4.2.2 Unterscheidung Tippbetrieb / Dauerbetrieb

Auswahl

Beim konventionellen Verfahren im JOG-Betrieb wird zwischen Verfahren im Tippbetrieb und im Dauerbetrieb unterschieden.

Die Auswahl erfolgt mit dem allgemeinen Settingdatum:
SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD

(Tipp-/ Dauerbetrieb bei JOG kontinuierlich)
und ist gemeinsam für alle Achsen gültig.

Grundeinstellung

Das Verfahren im Tippbetrieb ist die Grundeinstellung.

Kontinuierliches Verfahren im Tippbetrieb

Funktion

Beim Tippbetrieb (Grundeinstellung) verfährt die Achse so lange, wie die Verfahrtaste gedrückt wird, sofern zuvor keine Achsbegrenzung erreicht wird. Bei Loslassen der Verfahrtaste wird die Achse bis zum Stillstand abgebremst, und die Bewegung gilt als beendet.

Kontinuierliches Verfahren im Dauerbetrieb

Funktion

Mit kurzer Betätigung der Verfahrtaste (erste steigende Flanke) startet die Achse die Verfahrbewegung mit der eingestellten Geschwindigkeit in die gewünschte Richtung. Diese Verfahrbewegung wird auch nach Loslassen der Verfahrtaste fortgesetzt. Die Bewegung der Achse wird entweder vom Bediener oder aufgrund von steuerungsinternen Reaktionen (z. B. SW-Endschalter erreicht) gestoppt.

⚠WARNUNG

Falls "Dauerbetrieb" angewählt ist, können mehrere Achsen durch Antippen der jeweiligen Richtungstaste gestartet werden. Eventuelle Verriegelungen sind über die PLC zu realisieren!

Verfahrbewegung unterbrechen

Der Bediener hat folgende Möglichkeiten, die Verfahrbewegung zu unterbrechen:

- Vorschub-Override auf 0% stellen
- Axiale Vorschubsperre (PLC-Nahtstellensignal)
- NC-STOP bzw. NC-STOP Achse/Spindel

Wird die Unterbrechungsursache aufgehoben, so fährt die Achse weiter.

Verfahrbewegung abbrechen

Die Verfahrbewegung wird durch folgende Bedienhandlungen bzw. Überwachungen gestoppt und abgebrochen:

- gleiche Verfahrtaste wird erneut betätigt (zweite steigende Flanke)
- Verfahrtaste der Gegenrichtung wird betätigt
- `RESET`

- bei Abwahl des kontinuierlichen Betriebs
- bei Erreichen der ersten gültigen Begrenzung
- bei Störungen

⚠ VORSICHT
Software-Endschalter und Arbeitsfeldbegrenzungen sind erst nach Anfahren des Referenzpunktes wirksam.

Hinweis

Während eine Achse verfährt, wird steuerungsintern ein Betriebsartenwechsel von JOG auf AUT bzw. auf MDA nicht zugelassen.

4.2.3 Besonderheiten beim kontinuierlichen Verfahren

Teilungsachsen

Bei einer als Teilungsachse deklarierten Achse verfährt diese auch beim kontinuierlichen Verfahren stets auf Teilungspositionen. Beispielsweise fährt die Achse im Tippbetrieb nach Loslassen der Verfahrtaste weiter auf die nächste in Fahrtrichtung liegende Teilungsposition.

Literatur:
/FB2/ Funktionshandbuch Erweiterungsfunktionen; Teilungsachsen (T1)

4.3 Inkrementelles Verfahren (INC)

4.3.1 Allgemeine Funktionalität

Inkremente vorgeben

Der von der Achse abzufahrende Verfahrweg wird durch sog. Inkremente(auch Schrittmaß genannt) festgelegt. Bevor der Maschinenbediener die Achse verfährt, muss er das gewünschte Inkrement einstellen.

Die Einstellung erfolgt beispielsweise über die Maschinensteuertafel. Vom PLC-Anwenderprogramm ist nach entsprechender Verknüpfung das dem gewünschten Inkrement zugehörige Nahtstellensignal:
DB31, ... DBB5 ff (Maschinenfunktion: INC1 bis INCvar)
zu setzen.

Einstellbare Inkremente

Der Bediener hat die Möglichkeit, bis zu sechs unterschiedliche Inkrementgrößen einzustellen:

- **Fünf feste Inkremente**

  Die Größen dieser Inkremente werden gemeinsam für alle Achsen mit dem folgenden, allgemeinen Maschinendatum festgelegt:

  MD11330 $MN_JOG_INCR_SIZE_TAB (Inkrementgröße INC/Handrad)

  Als Standardeinstellung wird INC1, INC10, INC100, INC1000 und INC10000 vorbesetzt.

- **Ein variables Inkrement (INCvar)**

  Die Inkrementvorgabe beim variablen Inkrement ist ebenfalls gemeinsam für alle Achsen mit dem allgemeinen Settingdatum (Größe des variablen Inkrements bei INC/Handrad) möglich.

Inkrementbewertung

Mit dem axialen Maschinendatum:
MD31090 $MA_JOG_INCR_WEIGHT
(Bewertung eines Inkrements einer Maschinenachse bei INC/Handrad)
wird die Wegbewertung **eines** JOG-Inkrements festgelegt.

4.3.2 Unterscheidung im Tippbetrieb / Dauerbetrieb

Auswahl

Auch beim inkrementellen Verfahren der Maschinenachsen wird zwischen Verfahren im Tippbetrieb und im Dauerbetrieb unterschieden.

Die Auswahl erfolgt mit dem allgemeinen Maschinendatum:
MD11300 $MN_JOG_INC_MODE_LEVELTRIGGRD (INC und REF im Tippbetrieb).

Der Tippbetrieb ist die Grundeinstellung.

Inkrementelles Verfahren im Tippbetrieb

Funktion

Mit Drücken der Verfahrtaste in die gewünschte Richtung (z. B. +) beginnt die Achse, das eingestellte Inkrement zu verfahren. Wird die Verfahrtaste losgelassen, bevor das Inkrement vollständig abgefahren wurde, so wird die Bewegung unterbrochen, und die Achse bleibt stehen. Mit erneuter Betätigung der gleichen Verfahrtaste verfährt die Achse den noch verbleibenden Restweg, bis dieser Null ist. Zuvor kann die Bewegung wiederum durch Loslassen der Verfahrtaste unterbrochen werden.

Ein Drücken der Verfahrtaste der entgegengesetzen Richtung bleibt wirkungslos, solange das Inkrement nicht restlos abgefahren wurde bzw. ein Abbruch der Bewegung erfolgte.

Verfahrbewegung abbrechen

Soll das Inkrement nicht zu Ende gefahren werden, so kann mit `RESET` oder dem Nahtstellensignal:
DB31, ... DBX2.2 (Restweg löschen)
abgebrochen werden.

Inkrementelles Verfahren im Dauerbetrieb

Funktion

Die Achse fährt das eingestellte Inkrement mit Betätigung der Verfahrtaste (erste steigende Flanke) vollständig ab. Wird die gleiche Verfahrtaste ein zweites Mal betätigt (zweite steigende Flanke) bevor die Achse das Inkrement abgefahren hat, so wird die Verfahrbewegung abgebrochen (d.h. nicht mehr zu Ende gefahren).

Verfahrbewegung unterbrechen

Verhalten entsprechend wie beim kontinuierlichenVerfahren.

Verfahrbewegung abbrechen

Diese Verfahrbewegung wird durch folgende Bedienhandlungen bzw. Überwachungen gestoppt und abgebrochen:

- gleiche Verfahrtaste wird erneut betätigt (zweite steigende Flanke)
- Verfahrtaste der Gegenrichtung wird betätigt
- `RESET`
- axiales Restweg löschen (PLC-Nahtstellensignal)
- bei Erreichen der ersten gültigen Begrenzung

⚠ VORSICHT
Software-Endschalter und Arbeitsfeldbegrenzungen sind erst nach Anfahren des Referenzpunktes wirksam.

- bei Abwahl oder Wechsel des aktuellen Inkrements (z. B. Wechsel von INC100 auf INC10)
- bei Störungen (z. B. bei Wegnahme der Reglerfreigabe)

Hinweis

Während eine Achse verfährt, wird steuerungsintern ein Betriebsartenwechsel von JOG auf AUT bzw. auf MDA nicht zugelassen.

⚠ WARNUNG

Falls "Dauerbetrieb" angewählt ist, können mehrere Achsen durch Antippen der jeweiligen Richtungstaste gestartet werden. Eventuelle Verriegelungen sind über die PLC zu realisieren!

4.3.3 Besonderheiten beim inkrementellen Verfahren

Teilungsachsen

Unabhängig vom aktuell eingestellten Inkrementwert fährt die als Teilungsachse deklarierte Achse (MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB (Achse ist Teilungsachse)) nach Betätigung der Verfahrtaste "+" die nächsthöhere Teilungsposition an.
Analog wird mit Betätigung der Verfahrtaste "-" die nächstniedrigere Teilungsposition angefahren.

Literatur:
/FB2/ Funktionshandbuch Erweiterungsfunktionen; Teilungsachsen (T1)

4.4 Handradfahren im JOG

4.4.1 Allgemeine Funktionalität

Anwahl

Die Betriebsart JOG muss aktiv sein. Vom Bediener ist zusätzlich das beim Handradfahren wirkende Inkrement INC1, INC10, ... einzustellen. An der PLC-Nahtstelle ist entsprechend wie beim inkrementellen Verfahren die gewünschte Maschinenfunktion zu setzen.

Verfahren

Durch Verdrehen des elektronischen Handrads wird die zugehörige Maschinenachse je nach Drehrichtung in positiver oder negativer Richtung verfahren.

Verfahrweg

Der beim Drehen des Handrades resultierende Verfahrweg ist von folgenden Faktoren abhängig:

- Anzahl der an der Schnittstelle empfangenen Handradpulse
- Aktives Inkrement (Maschinenfunktion INC1, INC10, INC100, ... INCvar)
- Pulsbewertung des Handrades mit dem allgemeinen Maschinendatum: MD11320 HANDWH_IMP_PER_LATCH (Handradimpulse pro Raststellung)
- Bewertung eines Inkrements bei INC/Handrad (MD31090 $MA_JOG_INCR_WEIGHT).

Fahrbefehl +/-

Während der Achsbewegung wird an die PLC das Nahtstellensignal:
DB21, ... DBX40.7 (Fahrbefehl+)
bzw.
DB21, … DBX40.6 (Fahrbefehl-)
abhängig von der Bewegungsrichtung ausgegeben.

Wird die Achse bereits über die Verfahrtasten verfahren, so ist ein zusätzliches Handradfahren nicht möglich.

Handradanschluss

Es sind bis zu 2 Handräder simultan anschließbar. Somit können gleichzeitig bis zu 2 Achsen mit Handrädern bewegt werden.

Ausnahme: Falls mehrere Achsen gleichzeitig einem Handrad zugeordnet sind, können mehr als 2 Achsen mit Handrädern verfahren werden.

Handradzuordnung

Einer Geometrie- bzw. Maschinenachse wird über jeweils ein achsspezifisches VDI-Nahtstellensignal ein Handrad zugeordnet.

Welche Achse durch Verdrehen des Handrades 1 oder 2 bewegt wird, ist folgendermaßen einstellbar:

- über die PLC-Anwenderschnittstelle mit dem Nahtstellensignal:
  DB31, ... DBX4.0 - DBX4.2 (Handrad aktivieren).

  (bei Geometrieachse: DB21, ... DBX12.0 - 12.2 ff)

oder

- durch menügeführte Bedienung (HMI)

  Bei Betätigen des Softkeys "Handrad" im Grundmenü der Betriebsart JOG wird das Fenster "Handrad" eingeblendet. Hiermit kann jedem Handrad eine Achse zugeordnet sowie das Handrad freigegeben oder gesperrt werden.

Die Verknüpfung auf die PLC-Nahtstelle erfolgt durch das PLC-Anwenderprogramm. Dabei können mehrere Achsen gleichzeitig einem Handrad zugeordnet werden.

Funktion

Mit Hilfe von elektronischen Handrädern (Zubehör) können die angewählten Achsen im Handbetrieb simultan verfahren werden. Die Bewertung der Teilstriche der Handräder wird über die Schrittmaß-Bewertung festgelegt. Das Verfahren von Hand kann bei angewählter Koordinatenverschiebung oder -drehung auch im transformierten Werkstückkoordinatensystem erfolgen.

Handradanwahl von HMI

Für die Aktivierung des Handrades von der Bedientafelfront gibt es eine eigene Anwendernahtstelle zwischen HMI und PLC. Diese vom PLC-Grundprogramm bereitgestellte Nahtstelle für Handrad 1 und 2 enthält folgende Informationen:

- die dem Handrad zugeordneten Achsnummern:
  DB10 DBB100 ff (Achsnummer Handrad n)
- die Zusatzinformation Maschinen- oder Geometrieachse:
  DB10 DBX100.7 ff (Maschinenachse)
- die dem Handrad zugeordnete Kanalnummer, falls bei der Handradanwahl eine Geometrieachse angewählt wurde:
  DB10 DBX97 ff (Kanalnummer Geometrieachse Handrad n)
- die Information, dass das Handrad freigegeben bzw. gesperrt ist:
  DB10 DBX100.6 ff (Handrad abgewählt)

Vom PLC-Grundprogramm wird für die vorgegebene Achse das zugehörige Nahtstellensignal:
DB31, ... DBX4.0 - DBX4.2 (Handrad aktivieren)
(bei Geometrieachse: DB21, ... DBX12.0 - 12.2 ff)
entweder auf "0" (Sperren) oder auf "1" (Freigabe) gesetzt.

Eingangsfrequenz

Die Handradanschlüsse können Handradpulse mit einer maximalen Eingangsfrequenz von 100 kHz empfangen.

Geschwindigkeit

Auch beim Handradfahren werden die folgenden bei JOG wirksamen Achsgeschwindigkeiten verwendet:

- SD41110 $SN_JOG_SET_VELO (Achsgeschwindigkeit bei JOG)
- SD41130 $SN_JOG_ROT_AX_SET_VELO (Achsgeschwindigkeit der Rundachsen bei JOG-Betrieb)
- MD32020 $MA_JOG_VELO (Konventionelle Achsgeschwindigkeit)

Bedingt durch den begrenzten Vorschub kann insbesondere bei großer Pulsbewertung die Achse der Drehung des Handrades nicht zeitsynchron folgen, so dass sich ein Nachlaufen der Achse ergibt.

Abbruch der Verfahrbewegung

Ein RESET oder das Nahtstellensignal:
DB31, ... DBX2.2 (Restweg löschen/Spindel-Reset)
bewirken einen Abbruch der Verfahrbewegung.
Die anstehende Soll-/Ist-Differenz wird gelöscht.

Mit STOP wird die Verfahrbewegung lediglich unterbrochen.
Eine anstehende Soll-/Ist-Differenz bleibt erhalten.

Mit START wird der Restweg anschließend abgefahren.

Inkrementwert-Begrenzung

Über das kanalspezifische Maschinendatum:
MD20620 $MC_HANDWH_GEOAX_MAX_INCR_SIZE
(Begrenzung Handrad Inkrement für Geo-Achsen)
kann der Bediener die Größe des angewählten Inkrements für Geometrieachsen begrenzen.

Mit dem folgenden axialen Maschinendatum kann die Größe des angewählten Inkrements für Maschinenachsen begrenzt werden:

MD32080 $MA_HANDWH_MAX_INCR_SIZE (Begrenzung des angewählten Inkrements)

Eine durch das Handrad bestimmte Verfahrbewegung für eine **Geometrieachse** wird bestimmt durch:

- Verfahrweg
- Größe des variablen Inkrements
  (SD41010 $SN_JOG_VAR_INCR_SIZE)
- Geometrieachsenzuordnung
  (MD20620 $MC_HANDWH_GEOAX_MAX_INCR_SIZE)

Eine durch das Handrad bestimmte Verfahrbewegung für eine **Maschinenachse** wird bestimmt durch:

- Verfahrweg
- Größe des variablen Inkrements
  (SD41010 $SN_JOG_VAR_INCR_SIZE)
- Maschinenachsenzuordnung
  (MD32080 $ HANDWH_MAX_INCR_SIZE)

Fahren in Gegenrichtung

Abhängig vom Maschinendatum:
MD11310 $MN_HANDWH_REVERSE (Schwelle für Richtungswechsel Handrad)
ist das Verhalten bei einer Umkehr der Verfahrrichtung wie folgt:

- Wird das Handrad in Gegenrichtung bewegt, so wird die resultierende Wegstrecke berechnet und der so berechnete Endpunkt schnellstmöglich angefahren.

  Liegt dieser Endpunkt vor dem Punkt, auf den die fahrende Achse bei der augenblicklichen Fahrtrichtung bremsen kann, so wird abgebremst und dann der Endpunkt durch Fahren in Gegenrichtung angefahren. Andernfalls wird der neu berechnete Endpunkt sofort angefahren.
- Wird das Handrad um mindestens die im Maschinendatum angegebene Anzahl von Pulsen in Gegenrichtung bewegt, so wird die Achse schnellstmöglich abgebremst und alle bis zum Ende der Interpolation eintreffenden Impulse werden ignoriert.

  D. h. erst nach dem Stillstand (sollwertseitig) der Achse wird erneut verfahren.

Beschleunigung

Auch beim Handfahren erfolgt die Beschleunigung entsprechend einer vorgegebenen Kennlinie. Die bei JOG wirkende Beschleunigungskennlinie für die einzelne Achse wird mit dem folgenden axialen Maschinendatum festgelegt:

MD32420 $MA_ JOG_AND_POS_JERK_ENABLE (Grundeinstellung der axialen Ruckbegrenzung)

Literatur:
Funktionshandbuch Grundfunktionen; Beschleunigung (B2)

Verhalten am Softwareendschalter, Arbeitsfeldbegrenzung

Beim Verfahren in der Betriebsart JOG wird nur bis zur jeweils ersten aktiven Begrenzung gefahren und der entsprechende Alarm ausgegeben.

Abhängig vom Maschinendatum:
MD11310 $MN_HANDWH_REVERSE (Schwelle für Richtungswechsel Handrad)
ist das Verhalten dann wie folgt (so lange die Achse noch nicht sollwertseitig zum Endpunkt gekommen ist):

- Die aus den Handradimpulsen resultierende Wegstecke bildet einen fiktiven Endpunkt, der für die nachfolgenden Berechnungen verwendet wird.

  Liegt dieser fiktive Endpunkt z. B. 10 mm hinter der Begrenzung, so müssen diese 10 mm erst wieder in Gegenrichtung verfahren werden, bevor die Achse tatsächlich wieder verfährt. Soll an einer Begrenzung sofort wieder in Gegenrichtung verfahren werden, so kann der fiktive Restweg über Restweglöschen oder Abwahl der Handradzuordnung gelöscht werden.
- Alle Handradimpulse, die zu einem Endpunkt hinter der Begrenzung führen, werden ignoriert. Eine Bewegung des Handrades in Gegenrichtung führt unmittelbar zu einem Verfahren in Gegenrichtung, d. h. von der Begrenzung weg.

Begrenzungen

Die Begrenzungen wirken auch beim Verfahren mit Handrad.

Weitere Informationen siehe Thema "Überwachungen".

Umdrehungsvorschub

In der Betriebsart JOG ist das Verhalten der Achse/Spindel auch von der Einstellung des folgenden Settingdatums abhängig:

SD41100 $SN_JOG_REV_IS_ACTIVE (JOG: Umdrehungs.- / Linearvorschub)

SD41100 $SN_JOG_REV_IS_ACTIVE (JOG: Umdrehungs.- / Linearvorschub)	
aktiv	Eine Achse/Spindel wird immer mit Umdrehungsvorschub: MD32050 $MA_JOG_REV_VELO (Umdrehungsvorschub bei JOG) bzw. MD32040 $MA_JOG_REV_VELO_RAPID (Umdrehungsvorschub bei JOG mit Eilgangsüberlagerung) in Abhängigkeit von der Masterspindel verfahren.
nicht aktiv	Das Verhalten der Achse/Spindel ist abhängig vom Settingdatum: SD43300 $SA_ASSIGN_FEED_PER_REV_SOURCE (Umdrehungsvorschub für Positionsachsen/-spindeln)
	Das Verhalten einer Geometrieachse, auf die ein Frame mit Rotation wirkt, ist abhängig vom kanalspezifischen Settingdatum: SD42600 $SC_JOG_FEED_PER_REV_SOURCE (Steuerung Umdrehungsvorschub in JOG). (In der Betriebsart JOG Umdrehungsvorschub für Geometrieachsen, auf die ein Frame mit Rotation wirkt).

4.4.2 Fahranforderung

Mit den Signalen "Fahranforderung" ergeben sich gegenüber dem bisherigen Verhalten Erweiterungen, die im Folgenden beschrieben werden.

Signale "Fahranforderung"

DB21, … DBX40.5	Fahranforderung +	Geometrieachse 1
DB21, … DBX40.4	Fahranforderung -	Geometrieachse 1
DB21, … DBX46.5	Fahranforderung +	Geometrieachse 2
DB21, … DBX46.4	Fahranforderung -	Geometrieachse 2
DB21, … DBX52.5	Fahranforderung +	Geometrieachse 3
DB21, … DBX52.4	Fahranforderung -	Geometrieachse 3
DB21, … DBX332.5	Fahranforderung +	Orientierungsachse 1
DB21, … DBX332.4	Fahranforderung -	Orientierungsachse 1
DB21, … DBX336.5	Fahranforderung +	Orientierungsachse 2
DB21, … DBX336.4	Fahranforderung -	Orientierungsachse 2
DB21, … DBX340.5	Fahranforderung +	Orientierungsachse 3
DB21, … DBX340.4	Fahranforderung -	Orientierungsachse 3
DB31, … DBX64.5	Fahranforderung +	von Achse/Spindel
DB31, … DBX64.4	Fahranforderung -	von Achse/Spindel

Handradfahren mit Wegvorgabe

Ist beim Handradfahren mit Wegvorgabe
(MD11346 $MN_HANDWH_TRUE_DISTANCE == 1 oder == 3)
eine anstehende Haltebedingung **kein Abbruch-Kriterium**
(siehe:
MD32084 $MA_HANDWH_STOP_COND
bzw.
MD20624 $MC_HANDWH_CHAN_STOP_COND)
so entspricht die Ausgabe der PLC-Signale "Fahranforderung" und "Fahrbefehl" dem allgemeinen Verhalten (siehe folgende Bilder).

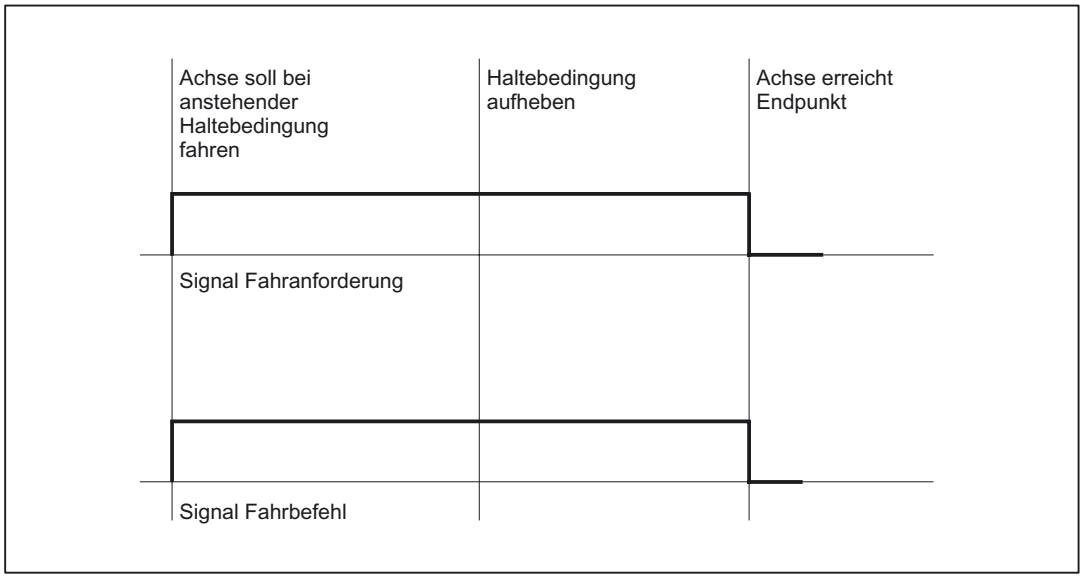


Bild 4-2 Signal / Zeitdiagramm MD17900 $MN_VDI_FUNCTION_MASK Bit 0 = 0

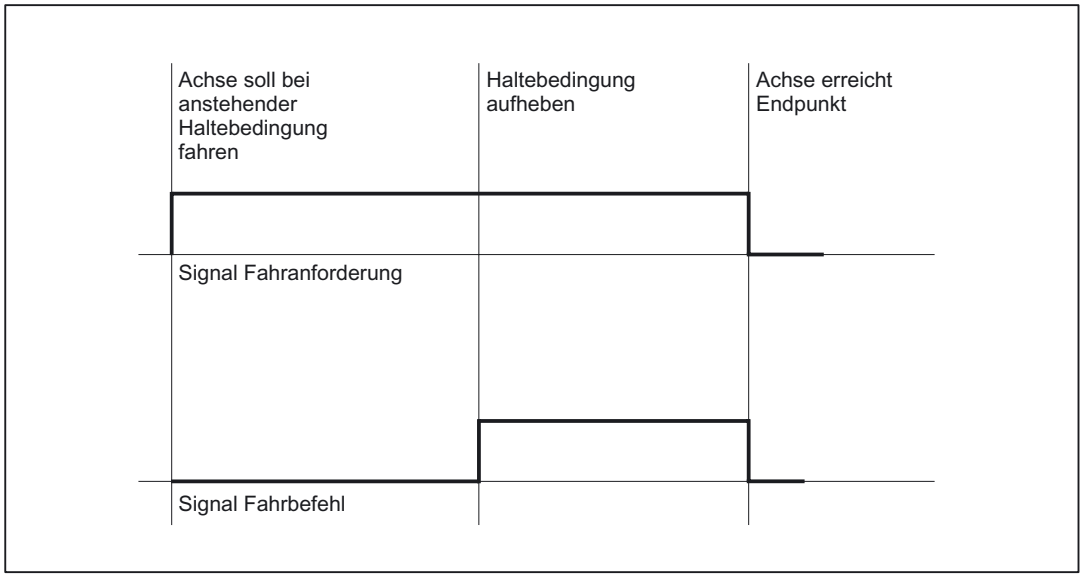


Bild 4-3 Signal / Zeitdiagramm MD17900 $MN_VDI_FUNCTION_MASK Bit 0 = 1

Beim Handradfahren wird, bei einer anstehenden Haltebedingung, die über das Maschinendatum:
MD32084 $MA_HANDWH_STOP_COND
bzw.
MD20624 $MC_HANDWH_CHAN_STOP_COND
als **Abbruch-Kriterium** angewählt ist, wie bisher **kein Fahrbefehl** ausgegeben (Kompatibilität), **aber** die entsprechende **Fahranforderung**.

Beim Aufheben der Haltebedingung wird das entsprechende PLC-Signal "Fahranforderung" zurückgesetzt, da ein Abbruch vorliegt. Es ist nun keine Haltebedingung mehr aktiv, die Achse kann aber nicht verfahren, da die Haltebedingung einen Abbruch verursacht hat.

Zusätzlich gilt, dass entweder Wegvorgabe
(MD11346 $MN_HANDWH_TRUE_DISTANCE == 1 oder == 3)
aktiv ist, oder das Handrad dauernd bewegt wird, d. h. Impulse liefert.

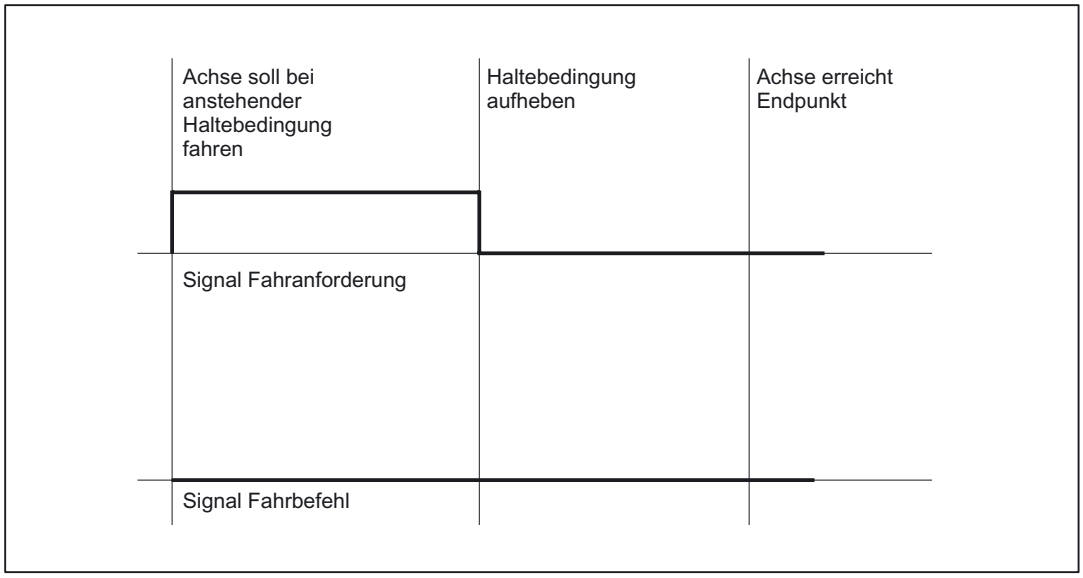


Bild 4-4 Signal / Zeitdiagramm Handradfahren mit Haltebedingung ist Abbruchkriterium

Wird während der Handradfahrbewegung eine Haltebedingung aktiviert, so wird die Bewegung abgebrochen und "Fahranforderung" und "Fahrbefehl" zurückgesetzt.

Mit Geschwindigkeitvorgabe

Wird bei Geschwindigkeitsvorgabe
(MD11346 $MN_HANDWH_TRUE_DISTANCE == 0 oder == 2)
das Handrad nicht mehr bewegt, so wird das PLC-Signal "Fahranforderung" zurückgesetzt.

Ebenso wird bei Handradabwahl das PLC-Signal "Fahranforderung" zurückgesetzt.

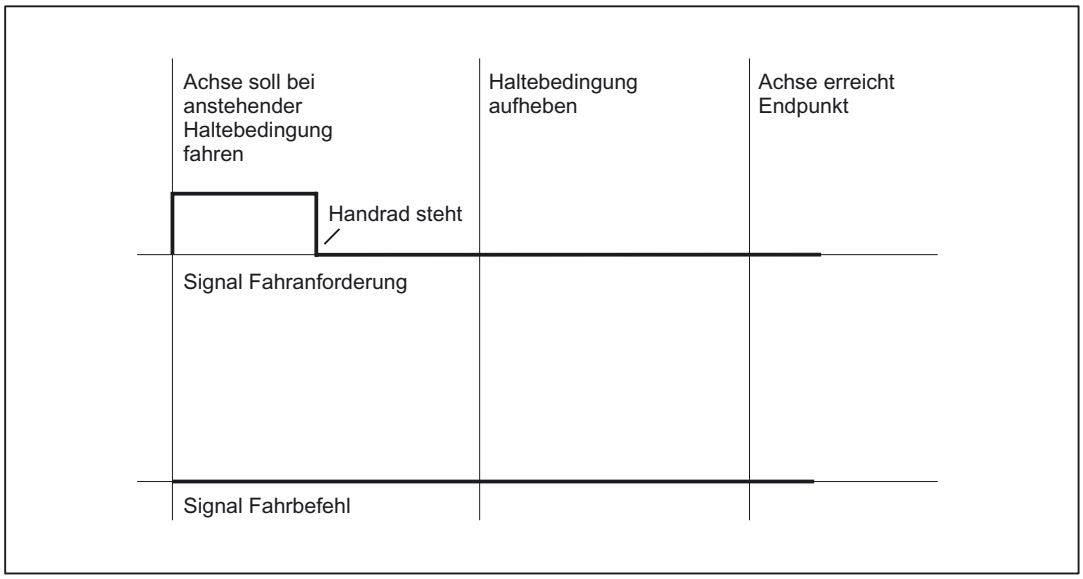


Bild 4-5 Signal / Zeitdiagramm Handradfahren Geschwindigkeitsvorgabe und Haltebedingung ist Abbruchkriterium

Randbedingungen

Bei anstehendem NC-Stopp wird kein Fahrbefehl und damit auch keine Fahranforderung ausgegeben. Eine Ausnahme davon gibt es beim DRF-Verfahren:

Wenn über das Maschinendatum:
MD20624 $MC_HANDWH_CHAN_STOP_COND (Bit 13 == 1)
ein DRF-Verfahren im Zustand NC-Stopp erlaubt ist, so entspricht das Verhalten dem des Handradfahrens.

Wie für den Fahrbefehl, so ergibt sich die Fahranforderung aus der Summe aller Teilbewegungen, d. h. auch der Anteil aus Kopplungen und Korrekturwerten wird berücksichtigt.

Beispiele

Im Maschinendatum:
MD32084 $MA_HANDWH_STOP_COND (Steuerung der VDI-Signale bzgl. Handrad)
ist Vorschub Halt als Abbruchkriterium eingestellt.
Das PLC-Signal "Vorschub Halt" steht an. Handradfahren ist angewählt (Betriebsart JOG, DRF-Verfahren in der Betriebsart AUTOMATIK).

Am Handrad wird in Plusrichtung gedreht:	Es wird das PLC-Signal "Fahranforderung +" von Achse/Spindel ausgegeben, es wird kein Fahrbefehl + von Achse/Spindel ausgegeben.
Das PLC-Signal "Vorschub Halt" wird zurückgesetzt:	keine Fahranforderung, kein Fahrbefehl

4.4.3 Doppelverwendung des Handrades

Alarm 14320

Die Doppelverwendung eines Handrades für DRF und Geschwindigkeits- bzw. Wegüberlagerung, einschließlich Konturhandrad, wird unterbunden und durch den selbstlöschenden Alarm 14320 (Handrad %1 doppelt verwendet (%2) in Kanal %3 Achse %4) angezeigt, falls auf eine Achse unterschiedliche Beeinflussungen durch das Handrad einwirken.

Dies bedeutet, dass eine überlagerte Bewegung erst ausgeführt werden kann, wenn für die an der Bewegung beteiligten Achsen im Basiskoordinatensystem keine DRF-Verschiebung, ausgelöst vom gleichen Handrad, mehr aktiv ist; d. h. die DRF-Bewegung muss beendet sein.

Ist eine überlagerte Bewegung gestartet, so kann für keine der daran beteiligten Achsen eine DRF-Verschiebung gestartet werden, die vom gleichen Handrad gespeist wird. Eine solche DRF-Bewegung ist erst möglich, wenn die Bewegung mit Überlagerung ihren Endpunkt erreicht hat oder mit Restweglöschen oder RESET abgebrochen wurde.

Soll die Handradüberlagerung und DRF-Verschiebung gleichzeitig aktiv sein, so ist dies mit Aktivierung von **zwei** getrennten Handrädern möglich.

Beispiel: Wegüberlagerung der Bahn

Annahme:
Kanal 1 und Geometrieachse X entspricht der Maschinenachse 3 und Geometrieachse Y entspricht Maschinenachse 5 und für die 1. Geometrieachse ist das Handrad 2 angewählt.

Wird der Satz `X10 Y10 FD=0` im Hauptlauf bearbeitet, so kann weder Maschinenachse 3 noch Maschinenachse 5 über das Handrad 2 mit DRF verfahren werden. Wird der Maschinenachse 3 das Handrad 2 zugeordnet während das kanalspezifische Signal DRF aktiv ist, so wird der Alarm 14320 (Handrad 2 doppelt verwendet (8) in Kanal 1 Achse X) gemeldet.

Wird Maschinenachse 3 oder Maschinenachse 5 unter Verwendung des 2. Handrades mit DRF verfahren, so kann die Bewegung `X10 Y10 FD=0` nicht ausgeführt werden und es wird der Alarm 14320 (Handrad 2 doppelt verwendet (3) in Kanal 1 Achse X) bzw. 14320 (Handrad 2 doppelt verwendet (3) in Kanal 1 Achse Y) gemeldet.

Beispiel: Geschwindigkeitsüberlagerung Positionierachse

Annahme:
Kanal 1: Kanalachse A entspricht Maschinenachse 4 und dieser Achse ist das Handrad 1 zugeordnet.

Wird der Satz `POS[A]=100 FDA[A]=0` im Hauptlauf bearbeitet, so kann die Maschinenachse 4 nicht mit DRF verfahren werden. D. h. ist das kanalspezifische Signal DRF aktiv, so wird der Alarm 14320 (Handrad 1 doppelt verwendet (6) in Kanal 1 Achse A) gemeldet.

Wird die Maschinenachse 4 mit DRF verfahren, so kann während einer aktuell stattfindenden DRF-Bewegung keine Bewegung `POS[A]=100 FDA[A]=0` ausgeführt werden. Es wird der Alarm 14320 (Handrad 1 doppelt verwendet (1) in Kanal 1 Achse A) gemeldet.

Beispiel: Wegüberlagerung PLC-Achse (840D sl)

Annahme:
Kanal 1: Maschinenachse 4 ist das Handrad 2 zugeordnet.

Wird eine vom FC18 ausgelöste Achsbewegung mit Wegüberlagerung der 4. Maschinenachse im Hauptlauf bearbeitet, so kann die Maschinenachse 4 nicht mit DRF verfahren werden. D. h. ist das kanalspezifische Signal DRF aktiv, so wird der Alarm 14320 (Handrad 2 doppelt verwendet (9) in Kanal 1 Achse A) gemeldet.

Wird die Maschinenachse 4 mit DRF verfahren, so kann während einer aktuell stattfindenden DRF-Bewegung keine vom FC18 ausgelöste Achsbewegung mit Wegüberlagerung ausgeführt werden. Es wird der Alarm 14320 (Handrad 2 doppelt verwendet (4) in Kanal 1 Achse A) gemeldet.

4.5 Handradüberlagerung in Automatik

4.5.1 Allgemeine Funktionalität

Funktion

Mit dieser Funktion können Achsen im Automatikbetrieb (Automatik, MDA) mit dem Handrad direkt verfahren bzw. deren Achsgeschwindigkeit verändert werden.

Die Aktivierung der Handradüberlagerung erfolgt im NC-Teileprogramm über die NC-Sprachenelemente `FD` (für Bahnachsen) und `FDA`(für Positionierachsen) und ist **satzweise wirksam**.

Bei Positionierachsen kann die Handradüberlagerung mit der Verfahranweisung `POSA` auch über die Satzgrenze hinaus wirken. Nach Erreichen der programmierten Zielposition ist die Handradüberlagerung wieder inaktiv.

Im gleichen NC-Satz können weitere Achsen interpolatorisch oder simultan verfahren werden.

Desweiteren kann die Funktion für konkurrierende Positionierachsen vom PLC-Anwenderprogramm aktiviert werden.

Unterscheidung

Abhängig vom programmierten Vorschub wird bei der Handradüberlagerung unterschieden zwischen:

- **Wegvorgabe**

  Achsvorschub = 0 (FDA = 0)

- **Geschwindigkeitsüberlagerung**

  Achsvorschub > 0 (FD bzw. FDA > 0)

In der folgenden Tabelle ist dargestellt, welche Achstypen durch die Funktion "Handradüberlagerung in Automatik" beeinflusst werden können.

Beeinflussbare Achsen bei "Handradüberlagerung in Automatik"		
Achstyp	**Geschwindigkeitsüberlagerung**	**Wegvorgabe**
Positionierachse	FDA[AXi] > 0 ; wirkt axial	FDA[AXi] = 0
konkurrierende Positionierachse	Parameter "Handradüberlagerung aktiv" = 1 und Achsvorschub > 0 von FC18	Parameter "Handradüberlagerung aktiv" = 1 und Achsvorschub = 0 von FC18
Bahnachse	FD > 0 ; wirkt auf die Bahngeschwindigkeit	nicht möglich

Wegvorgabe

Bei Achsvorschub = 0 (z. B. Programmierung von `FDA[AXi] = 0`) wird die Verfahrbewegung der Positionierachse auf die programmierte Zielposition ausschließlich vom Bediener durch Drehen des zugeordneten Handrades vorgenommen.

Die Drehrichtung des Handrades bestimmt die Verfahrrichtung der Achse. Die programmierte Zielposition kann bei der Handradüberlagerung nicht überschritten werden. Die Achse kann auch entgegen der programmierten Zielposition bewegt werden, wobei die Bewegung in die entgegengesetzte Richtung lediglich durch die axialen Positionsbegrenzungen eingeschränkt wird.

Der Satzwechsel wird vorgenommen, wenn:

- die Achse die programmierte Zielposition erreicht hat

oder

- der Restweg durch das axiale Nahtstellensignal:
  DB31, ... DBX2.2 (Restweg löschen)
  gelöscht wird.

Ab diesen Zeitpunkt wird die Wegvorgabe ausgeschaltet und weitere Handradimpulse sind wirkungslos.

Im Anschluss daran beziehen sich inkrementell programmierte Positionen auf den Unterbrechungspunkt und nicht auf die zuletzt programmierte Position.

Geschwindigkeitsüberlagerung

Bei der Geschwindigkeitsüberlagerung wird zwischen Achs- und Bahnvorschub unterschieden.

- **Überlagerung der Achsgeschwindigkeit** (FDA[AXi] > 0):

  Die Positionierachse wird mit dem programmierten axialen Vorschub auf die Zielposition gefahren. Mit dem zugeordneten Handrad kann die Achsgeschwindigkeit abhängig von der Drehrichtung erhöht bzw. bis auf maximal Null reduziert werden. Der resultierende Achsvorschub wird durch die Maximalgeschwindigkeit begrenzt. Die Achse kann jedoch nicht entgegen der programmierten Zielrichtung verfahren werden.

  Sobald die Achse die programmierte Zielposition erreicht hat, wird der Satzwechsel vorgenommen. Damit ist automatisch die Geschwindigkeitsüberlagerung wieder ausgeschaltet und weitere Handrandimpulse sind wirkungslos.

  Dies gilt analog auch für konkurrierende Positionierachsen, wobei die Zielposition und die Geschwindigkeit von der PLC vorgegeben werden.

- **Überlagerung der Bahngeschwindigkeit** (FD > 0):

  Die im NC-Satz programmierten Bahnachsen verfahren mit dem programmierten Bahnvorschub auf die Zielposition. Bei aktiver Geschwindigkeitsüberlagerung wird der programmierten Bahngeschwindigkeit die mit dem **Handrad der 1. Geometrieachse** erzeugte Geschwindigkeit überlagert. Sobald die programmierte Zielposition erreicht wird, erfolgt der Satzwechsel.

  Abhängig von der Drehrichtung am Handrad wird die Bahngeschwindigkeit erhöht oder bis auf maximal Null reduziert. Eine Umkehr der Bewegungsrichtung ist mit der Handradüberlagerung jedoch nicht möglich.

Anwendungsbeispiel

Die Funktion "Handradüberlagerung in Automatik" wird häufig bei Schleifmaschinen angewendet. Beispielsweise wird vom Bediener die pendelnde Schleifscheibe mit dem Handrad (Wegvorgabe) an das Werkstück positioniert. Nach Anfeuern wird die Verfahrbewegung beendet und der Satzwechsel eingeleitet (durch Aktivieren von DB31, ... DBX2.2 (Restweg löschen)).

Voraussetzungen

Bei Aktivierung der "Handradüberlagerung in Automatik" sind folgende Voraussetzungen erforderlich:

- der betroffenen Achse muss ein Handrad zugeordnet sein
- für das zugeordnete Handrad existiert eine Pulsbewertung

Handradzuordnung

Die Zuordnung der angeschlossenen Handräder zu den Achsen erfolgt analog dem "Handradfahren im JOG" über die PLC-Anwenderschnittstelle mit Nahtstellensignal:
DB31, ... DBX4.0 bis DBX4.2 (Handrad aktivieren)
oder durch menügeführte Bedienung.

Falls die Handradüberlagerung für eine Achse programmiert wird, für die kein Handrad zugeordnet ist, werden folgende Fälle unterschieden:

- **Bei Geschwindigkeitsüberlagerung:**

  Die Achsen verfahren mit der programmierten Geschwindigkeit.
  Es wird eine selbstlöschende Alarmmeldung (ohne Reaktion) ausgegeben.

- **Bei Wegvorgabe:**

  Es erfolgt keine Verfahrbewegung, da die Geschwindigkeit Null ist.
  Es wird eine selbstlöschende Alarmmeldung (ohne Reaktion) ausgegeben.

Hinweis

Bei der Geschwindigkeitsüberlagerung bei Bahnachsen wirkt lediglich das **Handrad der 1. Geometrieachse** auf die Bahngeschwindigkeit.

Handradbewertung

Der beim Drehen des Handrades um eine Rasterstellung resultierende Verfahrweg der Achse ist von mehreren Faktoren abhängig (siehe Thema "Handradfahren im JOG"):

- angewählte Inkrementgröße:

  MD11330 $MN_JOG_INCR_SIZE_TAB[5] (Inkrementgröße bei INC/Handrad)
  bzw.
  SD41010 $SN_JOG_VAR_INCR_SIZE (Größe des variablen Inkrements bei JOG)

- Bewertung eines Inkrements:

  MD31090 $MA_JOG_INCR_WEIGHT

- Anzahl der Handradimpulse je Rasterstellung:

  MD11320 $MN_HANDWH_IMP_PER_LATCH

Beispielsweise verfährt die Achse bei angewählter Maschinenfunktion INC1 und Standardeinstellung der o. g. Maschinendaten um 0,001 mm je Handradrasterstellung.

Bei der Geschwindigkeitsüberlagerung ergibt sich die Geschwindigkeit aus dem mit dem Handrad innerhalb einer Zeitdauer erzeugten Verfahrweg.

Beispiel

Annahmen:
Der Bediener dreht das Handrad mit 100 Impulse/Sekunde.

Die angewählte Maschinenfunktion ist INC100.

O. g. Maschinendaten für Handradbewertung mit Standardeinstellung.

⇒ Handrad-Verfahrweg pro Sekunde: 10 mm
⇒ Geschwindigkeitsüberlagerung: 0,6 m/min

PLC-Nahtstellensignale

Sobald die Handradüberlagerung wirksam ist, werden folgende Nahtstellensignale an die PLC auf 1-Signal gesetzt:

- **bei Positionierachsen:**

  DB31, ... DBX62.1 (Handradüberlagerung aktiv)

- **bei konkurrierenden Positionierachsen:**

  DB31, ... DBX62.1 (Handradüberlagerung aktiv)

- **bei Bahnachsen:**

  DB21, ... DBX33.3 (Handradüberlagerung aktiv)

Bei der Wegvorgabe werden abhängig von der Verfahrrichtung die entsprechenden Nahtstellensignale:
DB31, ... DBX64.6 und 64.7 (Fahrbefehle +/-)
an die PLC ausgegeben.

Begrenzungen

Bei Handradüberlagerung sind die axialen Begrenzungen (SW-Endschalter, HW-Endschalter, Arbeitsfeldbegrenzung) wirksam. Bei der Wegvorgabe kann die Achse in der programmierten Verfahrrichtung mit dem Handrad maximal bis zur programmierten Zielposition verfahren werden.

Die resultierende Geschwindigkeit wird durch das axiale Maschinendatum begrenzt.:

MD32000 $MA_MAX_AX_VELO (Maximale Achsgeschwindigkeit)

NC-STOP/Override = 0

Falls bei aktiver Handradüberlagerung der Vorschub-Override auf 0% eingestellt bzw. ein NC-STOP ausgelöst wird, gilt:

- **bei Wegvorgabe:**

  Die zwischenzeitlich eingehenden Handradimpulse werden aufsummiert und gespeichert. Mit NC-Start bzw. Vorschub-Override > 0% werden die gespeicherten Handradimpulse wirksam (d. h. abgefahren).

  Wird jedoch zuvor das Handrad inaktiv geschaltet
  (DB21, ... DBB12/16/20 (Handrad n aktivieren)),
  werden die gespeicherten Handradimpulse gelöscht.

- **bei Geschwindigkeitsvorgabe:**

  Die zwischenzeitlich eingehenden Handradimpulse werden nicht aufsummiert und sind unwirksam.

4.5.2 Programmierung und Aktivierung der Handradüberlagerung

Allgemeines

Bei der Programmierung der Handradüberlagerung mit den NC-Sprachenelementen `FD` (für Bahnachsen) und `FDA` (für Positionierachsen) sind folgende Punkte zu beachten:

- `FDA` bzw. `FD` wirken **satzweise.**

  Ausnahme bei Positionierachsen: Bei Programmierung der Verfahranweisung `POSA` kann auch die Handradüberlagerung über die Satzgrenze hinaus wirken, da die Satzweiterschaltung durch diese Positionierachse nicht beeinflusst wird.

- Bei Aktivierung der Handradüberlagerung mit `FDA` bzw. `FD` muss im NC-Satz für die Positionierachse bzw. für eine Bahnachse eine **Zielposition** programmiert werden. Nach Erreichen der programmierten Zielposition ist die Handradüberlagerung wieder inaktiv.
- Die Programmierung von `FDA` und `FD` bzw. `FA` und `F` im gleichen NC-Satz ist nicht möglich.
- Die Positionierachse darf keine Teilungsachse sein.

Positionierachse

Syntax für Handradüberlagerung: `FDA[AXi]=[Vorschubwert]`

Beispiel 1:

Geschwindigkeitsüberlagerung aktivieren

```
N10 POS[U]=10 FDA[U]=100 POSA[V]=20 FDA[V]=150 . . .
```

`POS[U]=10`	Zielposition der Positionierachse U
`FDA[U]=100`	Geschwindigkeitsüberlagerung für Positionierachse U aktivieren; Achsgeschwindigkeit von U ist 100 mm/min
`POSA[V]=20`	Zielposition der Positionierachse V (über Satzgrenze)
`FDA[V]=150`	Geschwindigkeitsüberlagerung für Positionierachse V aktivieren; Achsgeschwindigkeit von V ist 150 mm/min

Beispiel 2:

Wegvorgabe und Geschwindigkeitsüberlagerung im gleichen NC-Satz aktivieren

```
N20 POS[U]=100 FDA[U]= 0 POS[V]=200 FDA[V]=150 . . .
```

`POS[U]=100`	Zielposition der Positionierachse U
`FDA[U]= 0`	Wegvorgabe für Positionierachse U aktivieren;
`POS [V]=200`	Zielposition der Positionierachse V
`FDA[V]=150`	Geschwindigkeitsüberlagerung für Positionierachse V aktivieren, Achsgeschwindigkeit von V ist 150 mm/min

Bahnachse

Syntax für Handradüberlagerung: `FD = [Vorschubwert]`

Für die Programmierung der "Handradüberlagerung in Automatik" bei Bahnachsen sind folgende Voraussetzungen erforderlich:

- aktive Bewegungsbefehle der Gruppe 1: `G01`, `G02`, `G03`, `CIP`
- Genauhalt aktiv (`G60`)
- Linearvorschub in mm/min bzw. inch/min aktiv (`G94`)

Von der Steuerung werden diese Voraussetzungen überprüft und bei Nichteinhaltung wird ein Alarm gemeldet.

Beispiel 3:

Geschwindigkeitsüberlagerung aktivieren

```
N10 G01 X10 Y100 Z200 FD=1500 . . .
```

`X10 Y100 Z200`	Zielposition der Bahnachsen X, Y und Z
`FD=1500`	Geschwindigkeitsüberlagerung für Bahnachsen aktivieren, Bahngeschwindigkeit ist 1500 mm/min

Konkurrierende Positionierachse

Die Aktivierung der Handradüberlagerung bei konkurrierenden Positionierachsen erfolgt von der PLC über den FC18, indem das entsprechende Nahtstellensignal:

DB31, ... DBX62.1(Handradüberlagerung ein)
gesetzt wird.

Wird dabei der Parameter Geschwindigkeit (F_Wert) mit dem Wert 0 übergeben, so wirkt die aktivierte Handradüberlagerung als Wegvorgabe, d. h. der Vorschub wird in diesem Fall nicht vom axialen Maschinendatum:
MD32060 $MA_POS_AX_VELO (Löschstellung für Positionierachsgeschwindigkeit) abgeleitet.

Literatur:

- Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)
- Funktionshandbuch Grundfunktionen; PLC-Grundprogramm (P3)

4.5.3 Besonderheiten bei Handradüberlagerung in Automatik

Geschwindigkeitsanzeige

Bei der Handradüberlagerung werden bei der Geschwindigkeitsanzeige folgende Werte angezeigt:

- **Sollgeschwindigkeit**

  = programmierte Geschwindigkeit

- **Istgeschwindigkeit**

  = resultierende Geschwindigkeit einschließlich der Handradüberlagerung

Wirkung bei Planachsen

Ist die Achse als Planachse definiert und `DIAMON` aktiv, so werden bei aktiver Handradüberlagerung die Handradimpulse als Durchmesserwerte interpretiert und verfahren.

Probelaufvorschub

Bei aktivem Probelauf:
DB21, ... DBX0.6 (Probelaufvorschub aktivieren) = 1
wirkt grundsätzlich der Probelaufvorschub:
SD42100 $SC_DRY_RUN_FEED.

Somit wird trotz aktiver Handradüberlagerung mit Wegvorgabe (`FDA[AXi] = 0`) die Achse auch ohne Handradbeeinflussung mit Probelaufvorschub auf die programmierte Zielposition gefahren (d. h. die Wegvorgabe ist unwirksam).

DRF aktiv

Bei Aktivierung der "Handradüberlagerung in Automatik" muss darauf geachtet werden, ob die Funktion "DRF" aktiv ist (DB21, ... DBX0.3 = 1).

In diesem Fall würden die Handradimpulse auch eine DRF-Verschiebung der Achse bewirken. Vom Bediener sollte daher zuvor DRF abgewählt werden.

Vorschub-Override

Der Vorschub-Override wirkt nicht auf die vom Handrad erzeugten Geschwindigkeitsbewegungen (Ausnahme: 0%). Er wirkt nur auf den programmierten Vorschub.

Bei Wegvorgabe und schnellen Handradbewegungen kann u. U. (insbesondere bei großer Handradpulsbewertung) die Achse der Drehung des Handrades nicht zeitsynchron folgen, so dass sich ein Nachlaufen der Achse ergibt.

4.6 Konturhandrad/Wegvorgabe mit Handrad (Option)

Funktion

Bei aktivierter Funktion kann in den Betriebsarten AUTOMATIK und MDA die Vorschubbeeinflussung der Bahn- und Synchronachsen über ein Handrad vorgegeben werden.

Verfügbarkeit

Die Funktion "Konturhandrad" ist bei den Systemen SINUMERIK 840D sl und SINUMERIK 828D als lizenzpflichtige Option verfügbar.

Verhalten der Funktion

MD11346 $MN_HANDWH_TRUE_DISTANCE

Es kann folgendes Verhalten beim Konturhandrad über das obige Maschinendatum(Handrad Weg- oder Geschwindigkeitsvorgabe) eingestellt werden:

- **Wegvorgabe**

  Durch die Begrenzung auf die maximal zulässige Geschwindigkeit kommt es zu einem Nachlaufen der Achsen. Der durch das Handrad vorgegebene Weg wird gefahren. Es gehen **keine Impulse verloren**.

- **Geschwindigkeitsvorgabe**

  Über das Handrad wird nur die Geschwindigkeit vorgegeben, mit der verfahren werden soll. Sobald das Handrad steht, bleiben auch die Achsen stehen. Die Bewegung wird sofort abgebremst, wenn in einem IPO-Takt keine Impulse vom Handrad kommen. Dadurch kommt es zu **keinem Nachlaufen der Achsen**. Die Handradimpulse liefern keine Wegvorgabe.

Vorschub

Der Vorschub in mm/min ist **abhängig** von:

- der Anzahl von Impulsen des angewählten Handrads innerhalb eines Zeitraums
- der Pulsbewertung des Handrads über das Maschinendatum:
  MD11322 $MN_CONTOURHANDWH_IMP_PER_LATCH
  (Konturhandradimpulse pro Raststellung)

- dem aktivierten Inkrement (INC1, 10, 100, ...)
- der Weggewichtung eines Inkrements:
  MD31090 $MA_JOG_INCR_WEIGHT
  der ersten verfügbaren Geometrieachse

Der Vorschub ist **unabhängig** von:

- der programmierten Vorschubart (mm/min, mm/Umdr.)
- dem programmierten Vorschub (resultierende Geschwindigkeit kann auch höher sein)
- der Eilganggeschwindigkeit bei G0-Sätzen
- dem Override (die Stellung 0% wirkt, d. h. Stillstand)

Verfahrrichtung

Die Verfahrrichtung ist abhängig von der Drehrichtung:

- **Im Uhrzeigersinn**

  → Ergibt ein Verfahren in programmierter Richtung.

  Wird das Satzwechselkriterium (IPO-Ende) erreicht, dann wird der nächste Satz eingewechselt (Verhalten wie bei G60).

- **Gegen Uhrzeigersinn**

  → Ergibt ein Verfahren in programmierter Richtung.

  Es kann hier nur bis zum jeweiligen Satzanfang gefahren werden. Die Impulse werden nicht aufgesammelt, wenn weitergedreht wird.

Aktivierung der Funktion

Die Funktion kann über Nahtstellsignale oder über das NC-Programm aktiviert werden:

- Aktivierung über Nahtstellensignal

  Das Ein-/Ausschalten erfolgt über das Nahtstellensignal:
  DB21, ... DBX30.0, 30.1, 30.2 (Handrad x als Konturhandrad aktivieren)

- Aktivierung über NC-Programm

  Das Konturhandrad kann mit FD=0 im NC-Programm satzweise eingeschalten werden, d. h. im folgenden Satz gilt **ohne** erneute Programmierpflicht die Geschwindigkeit F aus dem Satz vor dem Konturhandrad.

Hinweis

Wurde in vorhergehenden NC-Sätzen kein Vorschub programmiert, so wird ein entsprechender Alarm ausgegeben.

FD und F in einem NC-Satz schließen sich aus (führt zu einem Alarm).

Simulation des Konturhandrads

Bei aktiviertem Konturhandrad kann das Konturhandrad auch simuliert werden.

Nach der Aktivierung über Nahtstellensignal:
DB21, ... DBX30.3 (Simulation Konturhandrad)

wird der Vorschub dann nicht mehr vom Konturhandrad vorgegeben, sondern der programmierte Vorschub verwendet.

Die Richtung wird auch über Nahtstellensignal vorgegeben:
DB21, ... DBX30.4 (Negative Richtung Simulation Konturhandrad)

Wenn die Simulation abgewählt wird oder die Richtung umgeschaltet wird, dann wird die laufende Bewegung mit Bremsrampe abgebremst.

Hinweis

Der Override wirkt wie beim Abarbeiten des NC-Programms.

Randbedingungen

- **Voraussetzungen**

  Festvorschub, Probelaufvorschub, Gewindeschneiden oder Gewindebohren darf nicht angewählt sein.

- **Grenzwerte**

  Auf die über Maschinendaten festgelegten Grenzwerte für Beschleunigung und Geschwindigkeit der beteiligten Achsen wird begrenzt.

- **Unterbrechung der Verfahrbewegung**

  Bei NC-STOP bleibt die Funktion angewählt, die Handradimpulse werden jedoch nicht aufsummiert und sind unwirksam.

  Voraussetzung: MD32084 $MA_HANDWH_CHAN_STOP_COND Bit 2 = 1

  DRF

  Eine angewählte DRF-Funktion wirkt zusätzlich wegüberlagernd.

- **Kanalspezifisches Restweglöschen**

  Dies führt zu einem Abbruch der durch das Konturhandrad ausgelösten Bewegung, die Achsen werden abgebremst und das Programm mit dem nächsten NC-Satz wieder gestartet. Danach ist das Konturhandrad wieder wirksam.

4.7 DRF-Verschiebung

Funktion

Mit Hilfe der Funktion "DRF-Verschiebung" (Differential Resolver Function) kann in der Betriebsart AUTOMATIK über ein elektronisches Handrad eine additive inkrementelle Nullpunktverschiebung von Geometrie- und Zusatzachsen im Basiskoordinatensystem eingestellt werden.

Die Handradzuordnung, d. h. die Zuordnung des Handrades von dem die Inkremente für die DRF-Verschiebung abgeleitet werden zur Geometrie- bzw. Zusatzachse die darüber verschoben werden soll, muss über die entsprechende Maschinenachsen erfolgen. Die entsprechenden Maschinenachsen sind dabei diejenigen Maschinenachsen, auf welche die Geometrie- oder Zusatzachse abgebildet wird.

Die DRF-Verschiebung wird in der Istwertanzeige der Achsen nicht angezeigt.

Literatur:
/BAD/ Bedienungsanleitung HMI Advanced
/BEM/ Bedienungsanleitung HMI Embedded

Anwendungen

Die DRF-Verschiebung ist beispielhaft in folgenden Anwendungsfällen einsetzbar:

- Korrektur des Werkzeugverschleißes innerhalb eines NC-Satzes

  Bei NC-Sätzen mit sehr langen Bearbeitungszeiten besteht die Notwendigkeit, innerhalb des NC-Satzes den Werkzeugverschleiß von Hand zu korrigieren (z. B. großen Flächenfräsmaschinen).
- Feinstkorrekturen beim Schleifen
- Einfache Temperaturkompensationen

⚠ VORSICHT

Die über die DRF-Verschiebung eingebrachte Nullpunktverschiebung ist in allen Betriebsarten und über RESET hinaus immer wirksam. Sie kann jedoch im Teileprogramm satzweise unterdrückt werden.

Geschwindigkeitsreduzierung

Die bei DRF mit dem Handrad erzeugte Geschwindigkeit kann über das axiale Maschinendatum:
MD32090 $MA_HANDWH_VELO_OVERLAY_FACTOR
(Verhältnis JOG -Geschwindigkeit zu Handradgeschwindigkeit)
gegenüber der JOG-Geschwindigkeit reduziert werden.

DRF wirksam

Damit die DRF-Verschiebung durch Verfahren mit dem Handrad verändert werden kann, muss DRF aktiv sein. Dazu sind folgende Voraussetzungen erforderlich:

- Betriebsart AUTOMATIK
- DB21, ... DBX0.3 (DRF aktivieren) = 1

Über die Funktion "Programmbeeinflussung" der HMI-Bedienoberfläche kann die DRF-Verschiebung kanalspezifisch ein- bzw. ausgeschaltet werden.

Die HMI-Software setzt daraufhin Nahstellensignal:
DB21, ... DBX24.3 (DRF angewählt) =1

Das PLC-Programm (PLC-Grundprogramm oder Anwenderprogramm) überträgt dieses Nahtstellensignal nach entsprechender Verknüpfung zum Nahtstellensignal:
DB21, ... DBX0.3 (DRF aktivieren)

Beeinflussung der DRF-Verschiebung

Die DRF-Verschiebung kann verändert, gelöscht oder gelesen werden:

Bediener:	• durch Verfahren mit dem Handrad
Teileprogramm:	• Lesen über die achsspezifische Systemvariable $AC_DRF[<Achse>] • Löschen über Teileprogrammbefehl (`DRFOF`) bei allen Achsen des Kanals • Satzweises unterdrücken über Teileprogrammbefehl (`SUPA`) **Literatur:** /PG/ Programmieranleitung Grundlagen
PLC-Anwenderprogramm:	• Lesen der DRF-Verschiebung (achsspezifisch) **Literatur:** /FB1/ Funktionsbeschreibung Grundmaschine; PLC-Grundprogramm (P3)
HMI-Bedienoberfläche:	• Anzeige der DRF-Verschiebung (achsspezifisch)

Hinweis

Beim Löschen der DRF-Verschiebung wird die Achse nicht verfahren!

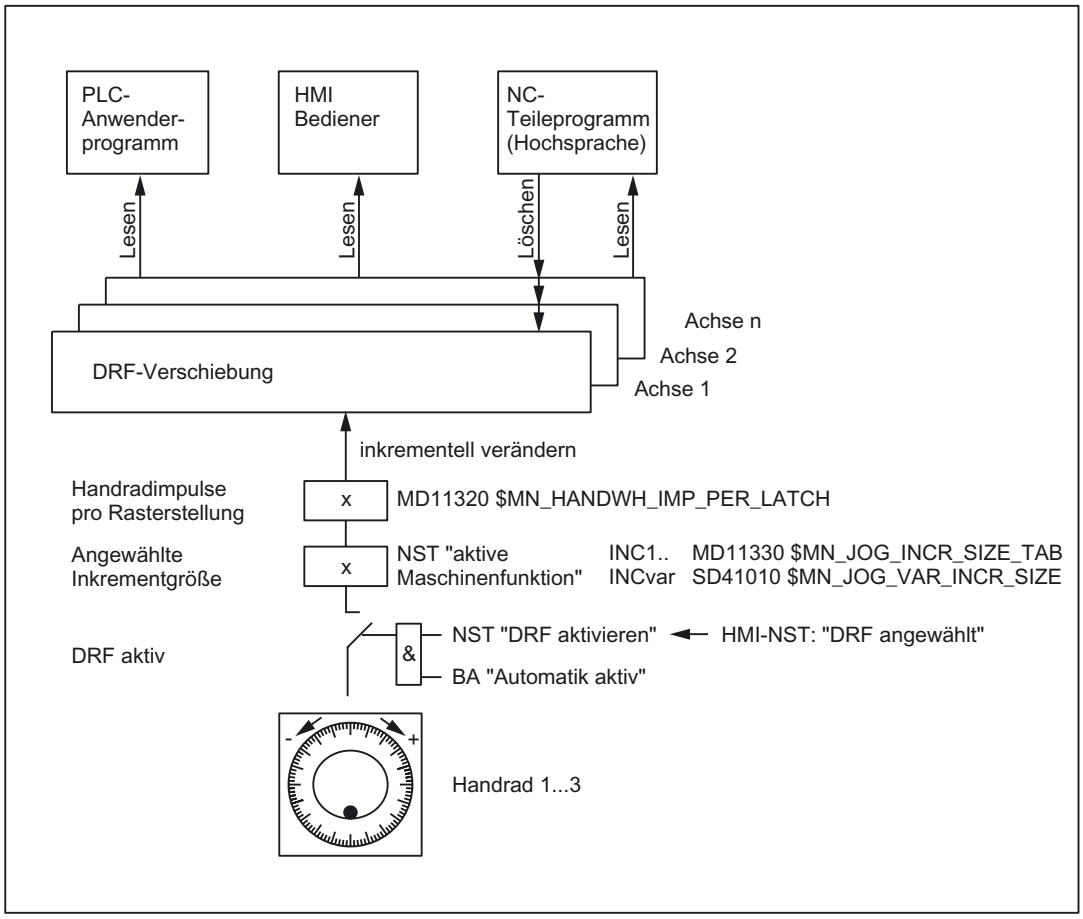


Bild 4-6 Beeinflussung der DRF-Verschiebung

Anzeige

Bei Verfahren einer Achse mit dem Handrad über DRF ändert sich die Achsistpositionsanzeige (ACTUAL POSITION) nicht. Die aktuelle DRF-Verschiebung der Achse kann im Fenster DRF zur Anzeige gebracht werden.

Referenzpunktfahren

In Phase 1 des Referenzpunktfahrens der Maschinenachse wird die DRF-Verschiebung für die entsprechende Geometrie- oder Zusatzachse gelöscht.

Während des Referenzpunktfahrens der Maschinenachse darf nicht gleichzeitig eine DRF-Verschiebung für die entsprechende Geometrie- oder Zusatzachse erfolgen.

Reset-Verhalten

`PowerOn-Reset`: die DRF-Verschiebung wird gelöscht

4.8 Inbetriebnahme: Handräder

4.8.1 Allgemeines

Um Handräder an einer SINUMERIK-Steuerung betreiben zu können, müssen diese über NCK-Maschinendaten parametriert werden.

Wenn die Handräder nicht direkt an der Steuerung angeschlossen sind, werden weitere Maßnahmen erforderlich, z. B. bei Anschluss über PROFIBUS-MCP bzw. -Handradmodul das Einfügen und Konfigurieren der Baugruppe mit SIMATIC STEP 7, HW-Konfig.

Hinweis

An einer SINUMERIK-Steuerung können maximal 6 Handräder parametriert werden.

Anschlussmöglichkeiten

SINUMERIK 840D sl

Bei SINUMERIK 840D sl können Handräder über folgende Komponenten angeschlossen werden:

- PROFIBUS-Baugruppe
- Ethernet-Baugruppe

Hinweis

An einer SINUMERIK 840D sl -Steuerung können gleichzeitig mehrere Handräder betrieben werden, die über unterschiedliche Komponenten angeschlossen sind.

SINUMERIK 828D

Bei SINUMERIK 828D werden Handräder direkt mit der Klemme X143 der PPU verdrahtet.

Hinweis

An einer SINUMERIK 828D -Steuerung können maximal 2 Handräder angeschlossen werden.

4.8.2 Anschluss über PPU - nur 828D

Parametrierung

Die Parametrierung der direkt an der Klemme X143 der PPU angeschlossenen Handräder erfolgt über folgende NCK-Maschinendaten:

- MD11350 $MN_HANDWHEEL_SEGMENT[< *Handradnummer_im_NCK - 1* >] = 2

  Bei direktem Anschluss an die PPU ist als Hardware-Segment immer 2 einzutragen.
- MD11351 $MN_HANDWHEEL_MODULE[< *Handradnummer_im_NCK - 1* >] = 1

  Bei direktem Anschluss an die PPU ist als Hardware-Modul immer 1 einzutragen.
- MD11352 $MN_HANDWHEEL_INPUT[< *Handradnummer_im_NCK - 1* >] = <*Handradanschluss* >

  Verwendeter Handradanschluss: 1 oder 2

Hinweis

An der Klemme X143 der PPU sind maximal 2 elektronische Handräder anschließbar.

Beispiel

Parametrierung von 2 Handrädern, die über die Klemme X143 direkt an die PPU angeschlossen sind.

Maschinendatum	Wert	Bedeutung
		1. Handrad:
MD11350 $MN_HANDWHEEL_SEGMENT[0]	2	Bei Anschluss über PPU immer 2
MD11351 $MN_HANDWHEEL_MODULE[0]	1	Bei Anschluss über PPU immer 1
MD11352 $MN_HANDWHEEL_INPUT[0]	1	1. Handradanschluss an PPU
		2. Handrad:
MD11350 $MN_HANDWHEEL_SEGMENT[1]	2	Bei Anschluss über PPU immer 2
MD11351 $MN_HANDWHEEL_MODULE[1]	1	Bei Anschluss über PPU immer 1
MD11352 $MN_HANDWHEEL_INPUT[1]	2	2. Handradanschluss an PPU

4.8.3 Anschluss über PROFIBUS - nur 840D sl

Parametrierung

Die Parametrierung der über PROFIBUS-Baugruppen (z. B. Maschinensteuertafel "MCP 483") angeschlossenen Handräder erfolgt in folgenden NCK-Maschinendaten:

- MD11350 $MN_HANDWHEEL_SEGMENT[<*Handradnummer_im_NCK - 1*>] = 5

  Bei Anschluss über eine PROFIBUS-Baugruppen ist als Hardware-Segment immer 5 (PROFIBUS) einzutragen.

- MD11351 $MN_HANDWHEEL_MODULE[<*Handradnummer_im_NCK - 1*>] = <*Index + 1*>

  Es ist der Verweis auf das MD11353 $MN_HANDWHEEL_LOGIC_ADDRESS[<*Index*>]

  einzutragen, das die logische Basisadresse des Handradslots enthält.

- MD11352 $MN_HANDWHEEL_INPUT[<*Handradnummer_im_NCK - 1*>] = <*Nummer_im_Handradslot*>

  Ein Handradslot kann mehrere Handräder umfassen. Es ist die Nummer des Handrades innerhalb des Handradslots einzutragen: 1, 2, ...

- MD11353 $MN_HANDWHEEL_LOGIC_ADDRESS[<*Index*>] = <*logischen Basisadresse*>

  Es ist die in SIMATIC STEP 7, HW-Konfig festgelegte logische Basisadresse des Handradslots einzutragen.

Handradslot

Neben der Parametrierung der Handräder in den NCK-Maschinendaten muss die PROFIBUS-Baugruppe in STEP 7 konfiguriert werden. Dabei wird unter anderem die logische Adresse des Handradslots festgelegt.

Der Handradslot liegt bei der jeweiligen PROFIBUS-Baugruppe auf folgendem Steckplatz:

PROFIBUS-Baugruppe	Steckplatz
Maschinensteuertafel MCP 438	2
Maschinensteuertafel MCP 310	2
Handradanschluss-Modul	1

Beispiel

Parametrierung von 5 Handrädern, angeschlossen über 4 Maschinensteuertafeln "MCP 483". An eine Maschinensteuertafel "MCP 483" können 2 Handräder angeschlossen werden.

Handradnummer im NCK	Maschinendatensatz (Index)	Anschluss
1	0	1. MCP, 1. Handrad im Handradslot
2	1	1. MCP, 2. Handrad im Handradslot
3	2	2. MCP, 1. Handrad im Handradslot

Handradnummer im NCK	Maschinendatensatz (Index)	Anschluss
5	4	3. MCP, 1. Handrad im Handradslot
6	5	4. MCP, 2. Handrad im Handradslot

Das 4. Handrad im NCK wird nicht verwendet (Maschinendaten-Lücke).

Hinweis

Bei der Parametrierung von Handrädern in den NCK-Maschinendaten sind Maschinendaten-Lücken zulässig.

Die Maschinensteuertafeln wurden in SIMATIC STEP 7, HW-Konfig folgendermaßen konfiguriert:

	Steckplatz	DP-Kennung	Bestellnummer / Bezeich...	E-Adresse	A-Adresse
1. MCP	1	55	Standard+Handrad	0 ... 7	0 ... 7
	2	*2AE*	*→ Standard+Handrad*	*288 ... 291*	
	3	*1*	*→ Standard+Handrad*		
2. MCP	1	55	Standard+Handrad	8 ... 15	8 ... 15
	2	*2AE*	*→ Standard+Handrad*	*304 ... 307*	
	3	*1*	*→ Standard+Handrad*		
3. MCP	1	55	Standard+Handrad	16 ... 23	16 ... 23
	2	*2AE*	*→ Standard+Handrad*	*320 ... 323*	
	3	*1*	*→ Standard+Handrad*		
4. MCP	1	55	Standard+Handrad	24 ... 29	24 ... 29
	2	*2AE*	*→ Standard+Handrad*	*330 ... 333*	
	3	*1*	*→ Standard+Handrad*		

Parametrierung in den NCK-Maschinendaten:

Maschinendatum	Wert	Bedeutung
		1. Handrad im NCK
MD11350 $MN_HANDWHEEL_SEGMENT[0]	5	Hardware-Segment: PROFIBUS
MD11351 $MN_HANDWHEEL_MODULE[0]	1	Verweis auf log. Basisadresse des Handradslot der 1. MCP
MD11352 $MN_HANDWHEEL_INPUT[0]	1	1. Handrad im Handradslot
		2. Handrad im NCK
MD11350 $MN_HANDWHEEL_SEGMENT[1]	5	Hardware-Segment: PROFIBUS
MD11351 $MN_HANDWHEEL_MODULE[1]	1	Verweis auf log. Basisadresse des Handradslot der 1. MCP
MD11352 $MN_HANDWHEEL_INPUT[1]	2	2. Handrad im Handradslot
		3. Handrad im NCK
MD11350 $MN_HANDWHEEL_SEGMENT[2]	5	Hardware-Segment: PROFIBUS

Maschinendatum	Wert	Bedeutung
MD11351 $MN_HANDWHEEL_MODULE[2]	2	Verweis auf log. Basisadresse des Handradslot der 2. MCP
MD11352 $MN_HANDWHEEL_INPUT[2]	1	1. Handrad im Handradslot
		4. Handrad im NCK
MD11350 $MN_HANDWHEEL_SEGMENT[3]	0	Kein Handrad parametriert
MD11351 $MN_HANDWHEEL_MODULE[3]	0	Kein Handrad parametriert
MD11352 $MN_HANDWHEEL_INPUT[3]	0	Kein Handrad parametriert
		5. Handrad im NCK
MD11350 $MN_HANDWHEEL_SEGMENT[4]	5	Hardware-Segment: PROFIBUS
MD11351 $MN_HANDWHEEL_MODULE[4]	6	Verweis auf log. Basisadresse des Handradslot der 3. MCP
MD11352 $MN_HANDWHEEL_INPUT[4]	1	1. Handrad im Handradslot
		6. Handrad im NCK
MD11350 $MN_HANDWHEEL_SEGMENT[5]	5	Hardware-Segment: PROFIBUS
MD11351 $MN_HANDWHEEL_MODULE[5]	5	Verweis auf log. Basisadresse des Handradslot der 4. MCP
MD11352 $MN_HANDWHEEL_INPUT[5]	2	2. Handrad im Handradslot

Logische Basisadressen:

Maschinendatum	Wert	Bedeutung
MD11353 $MN_HANDWHEEL_LOGIC_ADDRESS [0]	288	Logische Basisadresse Handradslot 1. MCP
MD11353 $MN_HANDWHEEL_LOGIC_ADDRESS [1]	304	Logische Basisadresse Handradslot 2. MCP
MD11353 $MN_HANDWHEEL_LOGIC_ADDRESS [4]	330	Logische Basisadresse Handradslot 4. MCP
MD11353 $MN_HANDWHEEL_LOGIC_ADDRESS [5]	320	Logische Basisadresse Handradslot 3. MCP

4.8.4 Anschluss über Ethernet - nur 840D sl

Parametrierung

Die Parametrierung der über Ethernet-Baugruppen, z. B. Maschinensteuertafel "MCP 483C IE", "HT 8" oder "HT 2" angeschlossenen Handräder erfolgt in folgenden NC-Maschinendaten:

- MD11350 $MN_HANDWHEEL_SEGMENT[< x - 1 >] = 7

  Bei Anschluss über Ethernet-Baugruppen ist als Segment immer 7 (Ethernet) einzutragen.

- MD11351 $MN_HANDWHEEL_MODULE[< x - 1 >] = 1

  Bei Anschluss über Ethernet-Baugruppen ist als Module immer 1 einzutragen.

- MD11352 $MN_HANDWHEEL_INPUT[< x - 1 >] = y

  mit y = 1, 2, 3, ... (Handradschnittstelle am Ethernet-Bus)

mit x = 1, 2, 3, ... (Handradnummer in der NC)

Handradschnittstellen am Ethernet-Bus

Die Nummerierung der Handradschnittstellen am Ethernet-Bus ergibt sich aus folgenden Festlegungen:

- Die Reihenfolge der Bedienkomponenten-Schnittstellen ist: MCP1, MCP2, BHG
- Jede Bedienkomponenten-Schnittstellen verfügt über zwei Handradschnittstellen
- Bedienkomponenten: MCP 483C IE

  An eine MCP 483C IE können über die Anschlüsse X60 und X61 maximal zwei Handräder angeschlossen werden. Die Zuordnung der Anschlüsse in der Bedienkomponenten-Schnittstelle ist:

  – Anschluss X60: 1. Handrad in der Bedienkomponenten-Schnittstelle MCP1 / MCP2
  – Anschluss X61: 2. Handrad in der Bedienkomponenten-Schnittstelle MCP1 / MCP2
- Bedienkomponenten: HT 8

  Das Handrad des HT 8 ist immer dem 1. Handrad der Bedienkomponenten-Schnittstelle MCP1 / MCP2 zugeordnet.
- Bedienkomponenten: HT 2

  Das Handrad des HT 2 ist immer dem 1. Handrad in der Bedienkomponenten-Schnittstellen BHG zugeordnet.

Bedienkomponenten-Schnittstelle ->	**MCP1**		**MCP2**		**BHG**	
Handradschnittstelle [1]	1	2	1	2	1	2
FB1-Parameter [2]	MCP1BusAdr		MCP2BusAdr		BHGRecGDNo	
Zuordnung der Handräder [3]						
MCP 483C IE	X60	X61	X60	X61	-	-
HT 8	x	-	x	-	-	-
HT 2	-	-	-	-	x	-
Handradschnittstelle am Ethernet-Bus (y) [4] ->	1	2	3	4	5	6

1) Nummerierung der Handradschnittstellen innerhalb einer Bedienkomponenten-Schnittstelle
2) Zuordnung der Bedienkomponente zur Schnittstelle über den entsprechenden FB1-Parameter
3) Zuordnung der Handräder der jeweiligen Bedienkomponente zu den Handradschnittstellen
4) Nummerierung der Handradschnittstellen am Ethernet-Bus -> MD11352 $MN_HANDWHEEL_INPUT[< x - 1 >] = **y**

Beispiel

Parametrierung von 3 Handrädern, angeschlossen über folgende Bedienkomponenten:

Bedienkomponenten-Schnittstelle ->	**MCP1**		**MCP2**		**BHG**	
Bedienkomponente	HT 8		MCP 483C		HT 2	
FB1-Parameter	MCP1BusAdr := 39		MCP2BusAdr := 192		BHGRecGDNo := 40	
Handradschnittstelle	**x**	-	-	**X61**	**x**	-
Handradschnittstelle am Ethernet-Bus ->	**1**	2	3	**4**	**5**	6

Tabelle 4- 1 NCK-Maschinendaten für die Handradzuordnung

Maschinendatum	Wert	Beschreibung
		HT 8: Handradnummer in der NC = 1
MD11350 $MN_HANDWHEEL_SEGMENT[0]	7	Segment: Ethernet
MD11350 $MN_HANDWHEEL_MODULE[0]	1	Modul: Ethernet
MD11350 $MN_HANDWHEEL_INPUT[0]	1	Handradschnittstelle am Ethernet-Bus
		MCP 483C IE: Handradnummer in der NC = 2
MD11350 $MN_HANDWHEEL_SEGMENT[1]	7	Segment: Ethernet
MD11350 $MN_HANDWHEEL_ MODULE [1]	1	Modul: Ethernet
MD11350 $MN_HANDWHEEL_ INPUT [1]	4	Handradschnittstelle am Ethernet-Bus
		HT 2: Handradnummer in der NC = 3
MD11350 $MN_HANDWHEEL_SEGMENT[2]	7	Segment: Ethernet
MD11350 $MN_HANDWHEEL_ MODULE [2]	1	Modul: Ethernet
MD11350 $MN_HANDWHEEL_ INPUT [2]	5	Handradschnittstelle am Ethernet-Bus

Tabelle 4- 2 FB1-Parameter (Auszug)

```
Parameter           Wert              Bemerkung
MCPNum              := 2              // Anzahl angeschlossener MCP
                                      // MCP1 = HT 8
MCP1In              ...               // MCP1-Parameter ...
...                 ...
MCP1BusAdr          := 39             // Über Schalter S1 und S2 am Anschlussgerät
                                      // eingestellte "IP-Adresse"
                                      // MCP2 = MCP 483C IE
MCP2In              ...               // MCP2-Parameter ...
...                 ...
MCP2BusAdr          := 192            // Über Schalter S2 an der MCP 483C
                                      // eingestellte "IP-Adresse"
MCPBusType          := b#16#55        // Bustyp: Ethernet
                                      // BHG = HT 2
BHG                 := 5              // Bustyp: Ethernet
BHGIn               ...               // BHG-Parameter ...
...                 ...
BHGRecGDNo          := 40             // Über Schalter S1 und S2 am Anschlussgerät
                                      // eingestellte "IP-Adresse"
```

Filterzeit

Aufgrund der nicht deterministischen Übertragung der Handradpulse am Ethernet-Bus, kann eine Filterung (Glättung) der Handradimpulsübertragung bei hochdynamischen Antrieben notwendig sein. Die Filterzeit wird über folgendes Maschinendatum parametriert:

- MD11354 $MN_HANDWHEEL_FILTER_TIME[< x - 1 >] = <Filterzeit>

  mit x = 1, 2, 3, ... (Handradnummer in der NC) und Filterzeit = 0.0 ... 2.0 s

Die Filterzeit gibt an, in welcher Zeit die an die Steuerung übertragenen Handradpulse zum Verfahren an den Interpolator weitergegeben werden. Bei einer Filterzeit von 0.0 s werden die Handradpulse innerhalb eines einzigen Interpolatortaktes an den Interpolator weitergegeben. Dies kann zu einem ruckartigen Verfahren der betreffenden Achse führen.

Empfohlene Filterzeit: 0.2 – 0.5 s

Stillstandserkennung

Die Stillstandserkennung erfolgt durch die Ethernet-Baugruppen an der das Handrad angeschlossen ist. Werden von einem Handrad für eine definierte Zeit keine Handradpulse übertragen, wird dies von der Baugruppe als Stillstand des Handrads erkannt und in die NC/PLC-Nahtstelle übertragen:

NC/PLC-Nahtstellensignal	Wert	Bedeutung
DB10, DBX245.0	0	Handrad 1 wird bewegt
	1	Handrad 1 steht
DB10, DBX245.1	0	Handrad 2 wird bewegt
	1	Handrad 2 steht
DB10, DBX245.2	0	Handrad 3 wird bewegt
	1	Handrad 3 steht

Durch Auswertung des Signals kann der Nachlauf einer über das Handrad verfahrenden Achse aufgrund in der Steuerung aufgesammelter, aber noch nicht an den Interpolator zum Verfahren ausgegebener Handradpulse verkürzt werden. Dazu ist mit dem Erkennen des Stillstands Restweglöschen für die entsprechende Achse bzw. im Kanal auszulösen:

- DB31,... DBX2.2 = 1 (axiales Restweglöschen)
- DB21,... DBX6.2 = 1 (kanalspez. Restweglöschen)

4.9 Besonderheiten beim Handfahren

4.9.1 Geometrieachsen beim Handfahren

Koordinatensysteme bei JOG

In der Betriebsart JOG hat der Bediener die Möglichkeit, die als Geometrieachsen deklarierten Achsen manuell auch im Werkstückkoordinatensystem (WKS) zu verfahren. Dabei bleiben evtl. angewählte Koordinatenverschiebungen oder -drehungen wirksam.

Hinweis

Bei SINUMERIK 840D kann mit "Transformationspaket Handling" im JOG-Betrieb die Translation von Geometrieachsen in mehreren gültigen Bezugssystemen getrennt voneinander einstellt werden.

Literatur:
/FB3/ Funktionshandbuch Sonderfunktion; 3- bis 5-Achs-Transformation (F2), Kap.: "Kartesisches Handverfahren"

Anwendung

Handfahrbewegungen bei denen Transformationen und Frames wirksam sein müssen. Die Geometrieachsen werden mit dem zuletzt gültigen Koordinatensystem verfahren. Nachfolgend werden die speziellen Eigenschaften für das Handfahren von Geometrieachsen beschrieben.

Simultanes Verfahren

Beim kontinuierlichen und inkrementellen Verfahren über Verfahrtasten ist zu beachten, dass simultan nur eine Geometrieachse verfahren werden kann. Beim Versuch, mehr als eine Geometrieachse zu verfahren, wird der Alarm 20062 "Achse bereits aktiv" gemeldet. Über die Handräder 1 bis 3 lassen sich jedoch 3 Geometrieachsen gleichzeitig verfahren. Der entsprechende Alarm 20060 wird gemeldet, wenn jeweils nur eine Achse nicht als Geometrieachse definiert ist.

PLC-Nahtstelle

Für Geometrieachsen gibt es eine eigene PLC-Nahtstelle:
DB21, ... DBB12-23 und DBB40-56,
die die gleichen Signale wie die achsspezifische PLC-Nahtstelle enthält.

Vorschub-/Eilgangkorrektur

Für das Handfahren von Geometrieachsen wirkt der kanalspezifische Vorschub-Korrekturschalter bzw. der Eilgangkorrekturschalter bei Eilgangüberlagerung.

Alarmmeldungen

Beim Handfahren einer Geometrieachse wird der Alarm 20062 "Achse bereits aktiv" unter folgenden Bedingungen gemeldet:

- Die Achse wird bereits über die axiale PLC-Schnittstelle im JOG-Betrieb verfahren.
- Ein Frame für ein gedrehtes Koordinatensystem ist aktiv, und es wird eine andere daran beteiligte Geometrieachse über Verfahrtasten im JOG-Betrieb verfahren.

Falls die Achse nicht als Geometrieachse definiert ist, wird beim Versuch, sie als Geometrieachse im JOG zu verfahren, der Alarm 20060 "Achse kann nicht als Geometrieachse verfahren werden" gemeldet.

4.9.2 Besonderheiten beim Handfahren der Spindel

Handfahren der Spindel

In der Betriebsart JOG können auch die Spindeln manuell verfahren werden. Dabei gelten im Wesentlichen die gleichen Bedingungen wie beim Handfahren von Achsen. Die Spindeln können bei JOG über Verfahrtasten kontinuierlich bzw. inkrementell sowohl im Tipp- als auch im Dauerbetrieb oder mit dem Handrad verfahren werden. Die Anwahl und Aktivierung erfolgt über die achs-/spindelspezifische PLC-Nahtstelle analog den Maschinenachsen. Ebenso wirken die achsspezifischen Maschinendaten auch für die Spindeln.

Spindelbetriebsart

Das Handfahren der Spindel ist sowohl im Positionierbetrieb (Spindel ist in Lageregelung) als auch im Steuerbetrieb möglich.

JOG-Geschwindigkeit

Die beim Handfahren der Spindeln verwendete Geschwindigkeit kann wie folgt festgelegt werden:

- mit dem gemeinsam für alle Spindeln gültigen allgemeinen Settingdatum:
  SD41200 $SN_JOG_SPIND_SET_VELO (Drehzahl für Spindel-JOG-Betrieb)

bzw.

- mit dem Maschinendatum:
  MD32020 $_MA_JOG_VELO (Konventionelle Achsgeschwindigkeit)

  Das Maschinendatum wirkt jedoch nur, falls:
  SD41110 $SN_JOG_SET_VELO (Achsgeschwindigkeit bei JOG) = 0

Auch beim Verfahren der Spindel mit JOG werden die Maximaldrehzahlen der aktiven Getriebestufe berücksichtigt.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Spindeln (S1)

Geschwindigkeitskorrektur

Bei den Spindeln wirkt für die bei JOG verfahrene Geschwindigkeit der Spindel-Korrekturschalter.

JOG-Beschleunigung

Da eine Spindel häufig sowohl im Drehzahlsteuerbetrieb als auch im Lageregelbetrieb mehrere Getriebestufen besitzt, wirkt bei JOG jeweils die aktuelle zur Getriebestufe zugehörige Beschleunigung der Spindel.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Spindeln (S1)

PLC-Nahtstellensignale

Beim Handfahren der Spindeln wirken die PLC-Nahtstellensignale zwischen NCK und PLC analog wie bei den Maschinenachsen.

Die Nahtstellensignale:
DB31, ... DBX60.7 bzw. DBX60.6 (Position erreicht mit Genauhalt fein bzw. grob)
werden nur gesetzt, falls die Spindel sich in Lageregelung befindet.

Bei den rein spindelspezifischen Nahtstellensignalen ist während des Verfahrens der Spindeln bei JOG folgendes zu beachten:

- Folgende PLC-Nahtstellensignale an Spindel wirken nicht:
  - DB31, ... DBX17.6 (M3/M4 invertieren)
  - DB31, ... DBX18.7 bzw. DBX18.6 (Solldrehrichtung links bzw. Solldrehrichtung rechts)
  - DB31, ... DBX18.5 (Pendeldrehzahl)
  - DB31, ... DBX16.7 (Spindel-RESET)
- Folgende PLC-Nahtstellensignale von Spindel werden nicht gesetzt:
  - DB31, ... DBX83.7 (Istdrehung rechts)
  - DB31, ... DBX83.5 (Spindel im Sollbereich)

4.9.3 Überwachungen

Begrenzungen

Beim Handfahren wirken folgende Begrenzungen:

- Arbeitsfeldbegrenzung (Achse muss referiert sein)
- Software-Endschalter 1 und 2 (Achse muss referiert sein)
- Hardware-Endschalter

Steuerungsintern wird sichergestellt, dass die Verfahrbewegung abgebrochen wird, sobald die erste gültige Begrenzung erreicht wird. Die Geschwindigkeitsführung sorgt dafür, dass der Bremsvorgang frühzeitig eingeleitet wird, so dass die Achse genau auf die

Begrenzungsposition (z. B. Software-Endschalter) zum Stehen kommt. Lediglich beim Ansprechen des Hardware-Endschalters wird die Achse mit "Schnellstopp" abgebremst.

Mit Erreichen der jeweiligen Begrenzung erfolgt eine Alarmmeldung (Alarme 16016, 16017, 16020, 16021). Anschließend wird steuerungsintern ein Weiterfahren in diese Richtung verhindert. Die Verfahrtasten sowie das Handrad für diese Richtung bleiben wirkungslos.

Hinweis

Damit die Software-Endschalter und Arbeitsfeldbegrenzungen wirksam werden, muss die Achse zuvor referiert gewesen sein.

Ist eine NPV (DRF-Verschiebung) über Handrad für Achsen aktiv, werden in der Betriebsart JOG die Softwareendschalter dieser Achse im Hauptlauf überwacht. D. h., beim Anfahren der Softwareendschalter wirkt die Ruckbegrenzung nicht. Es wird nach der Beschleunigung aus dem Maschinendatum:
MD32300 $MA_MAX_AX_ACCEL (Achsbeschleunigung)
die Geschwindigkeit am Software-Endschalter reduziert.

Weitere Informationen bzgl. Arbeitsfeldbegrenzungen sowie Hard- und Software-Endschalter siehe:
Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Achsüberwachungen, Schutzbereiche (A3)

Achse freifahren

Die Achse kann von einer Begrenzungsposition in die entgegengesetzte Richtung freigefahren werden.

Hinweis

Maschinenhersteller

Das Freifahren einer Achse, die die Begrenzungsposition angefahren hat, ist abhängig vom Maschinenhersteller. Bitte die Dokumentation des Maschinenherstellers beachten!

Maximale Geschwindigkeit und Beschleunigung

Die beim Handfahren verwendete Geschwindigkeit und Beschleunigung wird achsspezifisch vom Inbetriebnehmer über Maschinendaten festgelegt. Die Steuerung begrenzt die an den Achsen wirkenden Werte auf die maximalen Geschwindigkeits- und Beschleunigungsvorgaben.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (G2)
/FB1/ Funktionshandbuch Grundfunktionen; Beschleunigung (B2)

4.9.4 Sonstiges

Betriebsartenwechsel von JOG nach AUT oder von JOG nach MDA

Ein Betriebsartenwechsel von JOG nach AUT oder nach MDA wird nur dann ausgeführt, wenn alle Achsen des Kanals "Genauhalt Grob"erreicht haben.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)

Umdrehungsvorschub wirksam bei JOG

In der Betriebsart JOG kann auch eine Achse manuell mit Umdrehungsvorschub (entsprechend wie `G95`), bezogen auf die aktuelle Drehzahl der Master-Spindel, verfahren werden.

Die Aktivierung erfolgt über das Settingdatum:
SD41100 **$SN_**JOG_REV_IS_ACTIVE (JOG: Umdrehungs.- / Linearvorschub).

Der dabei verwendete Vorschubwert (in mm/U) ist auf folgende Weise vorgebbar:

- mit dem allgemeinen Settingdatum:
  SD41120 $SN_JOG_REV_SET_VELO (JOG-Geschwindigkeit für G95).
- durch die axialen Maschinendaten:
  MD32050 $MA_JOG_REV_VELO (Umdrehungsvorschub bei JOG)
  bzw. bei Eilgangüberlagerung:
  MD32040 $MA_JOG_REV_VELO_RAPID
  (Umdrehungsvorschub bei JOG mit Eilgangsüberlagerung),
  falls SD41120 = 0.

Falls keine Master-Spindel definiert ist und die Achse in JOG mit Umdrehungsvorschub verfahren werden soll, wird der Alarm 20055 bzw. bei Geometrieachsen der Alarm 20065 gemeldet.

Planachsen

Falls eine Geometrieachse als Planachse definiert ist und Radiusprogrammierung angewählt ist:
MD20100 $MC_DIAMETER_AX_DEF (Geometrieachsen mit Planachsfunktion),
sind beim Verfahren in JOG folgende Merkmale zu beachten:

- Kontinuierliches Verfahren:

  Beim kontinuierlichen Verfahren einer Planachse gibt es keine Unterschiede.

- Inkrementelles Verfahren:

  Es wird nur die **halbe Wegstrecke** der angewählten Inkrementgröße verfahren. Beispielsweise verfährt die Achse bei INC10 bei Betätigung der Verfahrtaste um 5 Inkrementwerte.

- Verfahren mit dem Handrad:

  Entsprechend wird beim inkrementellen Verfahren auch mit dem Handrad je Handradpuls nur die halbe Wegstrecke verfahren.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Planachsen (P1)

4.10 Festpunkt anfahren in JOG

4.10.1 Einführung

Funktion

Mit der Funktion "Festpunkt anfahren in JOG" kann der Maschinenbediener mit den Verfahrtasten der Maschinensteuertafel oder dem Handrad feste, über Maschinendaten definierte Achspositionen anfahren. Die fahrende Achse kommt auf dem definierten Festpunkt automatisch zum Stehen.

Anwendungen

Typische Anwendungen sind z. B.:

- das Anfahren einer Grundposition vor dem Start eines NC-Programms.
- das Anfahren von Werkzeugwechselpunkten, Beladepunkten und Palettenwechselpunkten.

Voraussetzungen

- Die Funktion "Festpunkt anfahren in JOG" ist nur aktivierbar in der Betriebsart JOG.

  In den Unterbetriebsarten JOG-REPOS und JOG-REF und in JOG in AUTOMATIK ist die Funktion nicht aktivierbar.
- Die zu verfahrende Achse muss referenziert sein.
- Es darf keine kinematische Transformation aktiv sein.
- Die zu verfahrende Achse darf keine Folgeachse einer aktiven Kopplung sein.

Festpunkt anfahren mit G75/G751

Das Anfahren der definierten Festpunkte kann über den Befehl `G75`/`G751` auch aus dem Teileprogramm heraus aufgerufen werden.

Weitere Informationen zum Festpunktfahren mit `G75`/`G751` siehe:
Literatur:
Programmierhandbuch Grundlagen; Kapitel: "Ergänzende Befehle" > "Festpunkt anfahren (G75, G751)"

4.10.2 Funktionalität

Vorgehensweise

Vorgehensweise beim "Festpunkt anfahren in JOG":

- Anwahl der Betriebsart JOG
- Aktivierung der Funktion "Festpunktfahren in JOG"
- Verfahren der Maschinenachse über Verfahrtasten oder Handrad

Aktivierung

Nach Anwahl der Funktion "Festpunkt anfahren in JOG" setzt die PLC das Nahtstellensignal:

DB31, ... DBX13.0-2 (JOG - Festpunkt anfahren)

Die Nummer des anzufahrenden Festpunkts wird durch Bit 0 - 2 binärcodiert ausgegeben.

Sobald die Funktion wirksam ist, bestätigt die NC die Aktivierung mit dem Nahtstellensignal:

DB31, ... DBX75.0-2 (JOG - Festpunkt anfahren aktiv)

Hinweis

Eine Aktivierung ist nicht möglich:

- während eines NCK-Reset
- bei anstehendem Not-Halt
- bei der Abarbeitung eines ASUP

Es erfolgt keine Alarmmeldung. Die Aktivierung erfolgt verzögert nach Abschluss bzw. nach Quittierung der aktiven Funktion.

Ablauf

Die eigentliche Verfahrbewegung wird mit den Verfahrtasten oder dem Handrad in Richtung des anzufahrenden Festpunkts gestartet.

Die angewählte Maschinenachse verfährt bis zum automatischen Stillstand auf dem anzufahrenden Festpunkt.

Bei Erreichen des Festpunkts mit "Genauhalt fein" wird das entsprechende NC/PLC-Nahtstellensignal gemeldet:

DB31, ... DBX75.3-5 (JOG - Festpunkt anfahren erreicht)

Dieses Anzeigesignal wird auch gemeldet, wenn die Achse über andere Methoden, wie z. B. NC-Programm, FC18 (bei 840D sl) oder Synchronaktion, sollwertseitig die Festpunktposition im Maschinenkoordinatensystem erreicht und istwertseitig innerhalb des Toleranzfensters "Genauhalt fein" (MD36010 $MA_STOP_LIMIT_FINE) zum Stehen kommt.

Fahren in Gegenrichtung

Das Verhalten beim Fahren in Gegenrichtung, d. h. entgegen der Richtung des anzufahrenden Festpunkts, ist abhängig von der Einstellung von Bit 2 im Maschinendatum:

MD10735 $MN_JOG_MODE_MASK (Einstellungen für Betriebsart JOG)

Ein Fahren in Gegenrichtung ist nur möglich, wenn das Bit gesetzt ist.

Wenn das Bit nicht gesetzt ist, dann ist das Fahren in Gegenrichtung gesperrt und bei dem Versuch, mit den Verfahrtasten oder dem Handrad die Richtung hin zu dem anzufahrenden Festpunkt zu verlassen, wird folgende Kanalzustandsmeldung ausgegeben:

"JOG: Richtung gesperrt <Achse>"

Anderen Festpunkt anfahren

Bei der Anwahl eines anderen Festpunkts während des Festpunktfahrens wird die Achsbewegung abgebrochen und folgender Alarm gemeldet:

Alarm 17812 "Kanal %1 Achse %2 Festpunktfahren in JOG: Festpunkt geändert"

Das Meldesignal DB31, ... DBX75.0-2 (JOG - Festpunkt anfahren aktiv) zeigt die Nummer des neu angewählten Festpunkts an. Zum Weiterfahren muss die JOG-Bewegung erneut ausgelöst werden.

Hinweis

Um die Alarmmeldung zu vermeiden, kann der Maschinenbediener wie folgt vorgehen:

1. Aktuelle Verfahrbewegung mit Restweglöschen abbrechen.
2. Nach Achsstillstand Festpunktfahren auf anderen Festpunkt aktivieren und starten.

Wegfahren vom Festpunkt / Deaktivierung

Zum Wegfahren von einer Festpunktposition muss die Funktion "Festpunkt anfahren in JOG" deaktiviert werden. Dies erfolgt durch Rücksetzen des Aktivierungssignals auf "0":

DB31, ... DBX13.0-2 = 0

Die Meldesignale DB31, ... DBX75.0-2 (JOG - Festpunkt anfahren aktiv) und DB31, ... DBX75.3-5 (JOG - Festpunkt anfahren erreicht) werden beim Verlassen der Festpunktposition gelöscht.

Sonderfall: Achse steht bereits auf Festpunkt

Wenn die Achse beim Starten des Festpunktfahrens bereits auf der Position des anzufahrenden Festpunkts steht, dann kann die Achse nicht verfahren werden. Dies wird über die folgende Kanalzustandsmeldung angezeigt:

"JOG: Position erreicht <Achse>"

Zum Wegfahren von der Festpunktposition muss die Funktion "Festpunkt anfahren in JOG" deaktiviert werden.

Besonderheiten beim inkrementellen Verfahren

Wird beim inkrementellen Verfahren der Festpunkt erreicht, bevor das Inkrement vollständig abgefahren ist, so wird das Inkrement als vollständig abgefahren betrachtet. Dies ist auch der Fall, wenn nur ganze Inkremente gefahren werden:

MD11346 $MN_HANDWH_TRUE_DISTANCE = 2 oder 3

Besonderheiten bei Modulo-Rundachsen

Modulo-Rundachsen können in beide Richtungen den Festpunkt anfahren. Es erfolgt keine Betrachtung zum Anfahren über den kürzesten Weg (DC).

Besonderheiten bei Spindeln

Eine Spindel wechselt bei Aktivierung der Funktion "Festpunkt anfahren in JOG" in den Positionierbetrieb. Dadurch wird die Lageregelung aktiv und es kann auf den Festpunkt gefahren werden.

Wenn keine Nullmarke erkannt wurde, erfolgt wie im Achsbetrieb die Alarmmeldung:

Alarm 17810 "Kanal %1 Achse %2 nicht referenziert"

Da eine Spindel auch immer eine Modulo-Rundachse sein muss, gilt für die Richtungsbetrachtung das Gleiche wie für Modulo-Rundachsen (siehe Absatz " Besonderheiten bei Modulo-Rundachsen").

4.10.3 Parametrierung

Fahren in Gegenrichtung

Das Verhalten beim Fahren in Gegenrichtung, d. h. entgegen der Richtung des anzufahrenden Festpunkts, ist abhängig von der Einstellung von Bit 2 im Maschinendatum:

MD10735 $MN_JOG_MODE_MASK (Einstellungen für Betriebsart JOG)

Bit	Wert	Bedeutung
2	0	Fahren in Gegenrichtung **nicht** möglich (Grundeinstellung).
	1	Fahren in Gegenrichtung möglich.

Festpunktpositionen

Über das folgende Maschinendatum können für jede Achse maximal 4 Festpunktpositionen definiert werden:

MD30600 $MA_FIX_POINT_POS[n]

Anzahl gültiger Festpunktpositionen

Die Anzahl gültiger Festpunktpositionen einer Achse wird festgelegt über das Maschinendatum:

MD30610 $MA_NUM_FIX_POINT_POS

Hinweis

Eine Ausnahme gibt es beim "Festpunkt anfahren mit `G75`/`G751`". Hier können auch bei einer Einstellung MD30610 = 0 zwei Festpunktpositionen angefahren werden.

4.10.4 Programmierung

Systemvariablen

Für die Funktion "Festpunkt anfahren" stehen folgende, im Teileprogramm und in Synchronaktionen lesbare Systemvariablen zur Verfügung:

Systemvariable	Bedeutung
$AA_FIX_POINT_SELECTED [<Achse>]	Nummer des anzufahrenden Festpunkts
$AA_FIX_POINT_ACT [<Achse>]	Nummer des Festpunkts, auf dem die Achse aktuell steht

4.10.5 Randbedingungen

Achse ist Teilungsachse

Wenn die zu verfahrende Achse eine Teilungsachse ist und die anzufahrende Festpunktposition nicht mit einer Teilungsposition übereinstimmt, dann wird die Achse nicht verfahren und es wird ein Alarm gemeldet.

Frames aktiv

Alle aktiven Frames werden ignoriert. Es wird im Maschinenkoordinatensystem verfahren.

Korrekturwerte aktiv

Aktive Korrekturwerte (DRF, externe Nullpunktverschiebung, Synchronaktionsoffset $AA_OFF, Online-Werkzeugkorrektur) werden mit herausgefahren. Der Festpunkt ist eine Position im Maschinenkoordinatensystem.

Wenn während des Festpunktfahrens in JOG eine Korrekturbewegung (DRF, externe Nullpunktverschiebung, Synchronaktionsoffset $AA_OFF, Online-Werkzeugkorrektur) aktiviert wird, dann wird ein Alarm gemeldet. Die Position des anzufahrenden Festpunkts im Maschinenkoordinatensystem wird nicht erreicht, sondern es wird auf eine Position gefahren, die ohne aktive Korrekturbewegung erreicht werden würde. Das dem Festpunkt entsprechende NC/PLC-Nahtstellensignal DB31, ... DBX75.3-5 wird nicht gemeldet.

Arbeitsfeldbegrenzungen

Arbeitsfeldbegrenzungen (im BKS und im WKS) werden berücksichtigt und die Achsbewegung wird bei Erreichen der Begrenzungen abgebrochen.

4.10.6 Anwendungsbeispiel

Ziel

Eine Rundachse (Maschinenachse 4 [AX4]) soll mit der Funktion "Festpunkt anfahren in JOG" auf den Festpunkt 2 (90 Grad) gefahren werden.

Parametrierung

Die Maschinendaten für das "Festpunkt anfahren" von Maschinenachse 4 sind wie folgt parametriert:

MD30610 $MA_NUM_FIX_POINT_POS[AX4] = 4	Für Maschinenachse 4 werden 4 Festpunkte definiert.
MD30600 $MA_FIX_POINT_POS[0,AX4] = 0	1. Festpunkt von AX4 = 0 Grad
MD30600 $MA_FIX_POINT_POS[1,AX4] = 90	2. Festpunkt von AX4 = 90 Grad
MD30600 $MA_FIX_POINT_POS[2,AX4] = 180	3. Festpunkt von AX4 = 180 Grad
MD30600 $MA_FIX_POINT_POS[3,AX4] = 270	4. Festpunkt von AX4 = 270 Grad

Ausgangssituation

Maschinenachse 4 ist referiert und steht auf Position 0 Grad. Dies entspricht dem 1. Festpunkt und wird gemeldet über das NC/PLC-Nahtstellensignal:

DB31 DBX75.0 = 1 (Bit 0-2 = 1)

Festpunkt 2 anfahren

Die Steuerung wird in die Betriebsart JOG geschaltet.

Das "Festpunkt anfahren" auf Festpunkt 2 wird aktiviert über das NC/PLC-Nahtstellensignal:

DB31 DBX13.1 = 1 (Bit 0-2 = 2)

Die Aktivierung wird bestätigt über das NC/PLC-Nahtstellensignal:

DB31 DB75.1 = 1 (Bit 0-2 = 2)

Über kontinuierliches Verfahren wird mit der Plus-Verfahrtaste der Maschinensteuertafel Festpunkt 2 angefahren.

Die Maschinenachse 4 stoppt auf Position 90 Grad. Dies wird gemeldet über das NC/PLC-Nahtstellensignal:

DB31 DBX75.4 = 1 (Bit 3-5 = 2)

4.11 Datenlisten

4.11.1 Maschinendaten

4.11.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
10000	AXCONF_MACHAX_NAME_TAB[n]	Maschinenachsname
10735	JOG_MODE_MASK	Einstellungen für Betriebsart JOG
11300	JOG_INC_MODE_LEVELTRIGGRD	INC und REF im Tippbetrieb
11310	HANDWH_REVERSE	Bestimmt Fahren in Gegenrichtung
11320	HANDWH_IMP_PER_LATCH[n]	Handradimpulse pro Rasterstellung
11324	HANDWH_VDI_REPRESENTATION	Codierung der Handradnummer (NCK/PLC-Nahtst.)
11330	JOG_INCR_SIZE_TAB[n]	Inkrementgröße bei INC / Handrad
11340	ENC_HANDWHEEL_SEGMENT_NR	Drittes Handrad: Bussegment
11342	ENC_HANDWHEEL_MODULE_NR	Drittes Handrad: logische Antriebsnummer
11344	ENC_HANDWHEEL_INPUT_NR	Drittes Handrad: Geberschnittstelle
11346	HANDWH_TRUE_DISTANCE	Handrad Weg- oder Geschwindigkeitsvorgabe
11350	HANDWHEEL_SEGMENT[n]	Handradsegment
11351	HANDWHEEL_MODULE[n]	Handradmodul
11352	HANDWHEEL_INPUT[n]	Handradanschluss
11353	HANDWHEEL_LOGIC_ADDRESS[n]	Logische Adresse des Handradslots (STEP 7)
17900	VDI_FUNCTION_MASK	Funktionsmaske für VDI Signale

4.11.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
20060	AXCONF_GEOAX_NAME_TAB	Geometrieachse im Kanal [n = Geoachsnummer]
20100	DIAMETER_AX_DEF	Geometrieachsen mit Planachsfunktion
20360	TOOL_PARAMETER_DEF_MASK	Definition der Werkzeug-Parameter
20620	HANDWH_GEOAX_MAX_INCR_SIZE	Begrenzung der Geometrieachsen
20622	HANDWH_GEOAX_MAX_INCR_VSIZE	Bahngeschwindigkeitsüberlagerung
20624	HANDWH_CHAN_STOP_COND	Diverse Parameter für das Handradfahren
21150	JOG_VELO_RAPID_ORI	Konventioneller Eilgang für Orientierungsachsen
21165	JOG_VELO_GEO	Konventionelle Geschwindigkeit für Geometrieachsen

4.11.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: $MA_	Beschreibung
30450	IS_CONCURRENT_POS_AX	Voreinstellung bei Reset: neutrale Achse oder Kanalachse
30600	FIX_POINT_POS[n]	Festpunktpositionen der Achse
30610	NUM_FIX_POINT_POS	Anzahl der Festpunktpositionen einer Achse
31090	JOG_INCR_WEIGHT	Bewertung eines Inkrements bei INC/Handrad
32000	MAX_AX_VELO	Maximale Achsgeschwindigkeit
32010	JOG_VELO_RAPID	Konventioneller Eilgang
32020	JOG_VELO	Konventionelle Achsgeschwindigkeit
32040	JOG_REV_VELO_RAPID	Umdrehungsvorschub bei JOG mit Eilgangüberlagerung
32050	JOG_REV_VELO	Umdrehungsvorschub bei JOG
32060	POS_AX_VELO	Löschstellung für Positionierachsgeschwindigkeit
32080	HANDWH_MAX_INCR_SIZE	Begrenzung der Größe des angewählten Inkrements
32082	HANDWH_MAX_INCR_VELO_SIZE	Begrenzung des angewählten Inkrements für Geschwindigkeitsüberlagerung
32084	HANDWH_STOP_COND	Wirkung der achsspezifischen VDI-Nahtstellensignale Bit 0...5 auf das Handrad
32090	HANDWH_VELO_OVERLAY_FACTOR	Verhältnis JOG- zu Handradgeschwindigkeit (bei DRF)
32300	MAX_AX_ACCEL	Maximale Achsbeschleunigung
32430	JOG_AND_POS_MAX_JERK	Maximaler axialer Ruck für JOG und Positionierachsbetrieb
35130	GEAR_STEP_MAX_VELO_LIMIT[n]	Maximaldrehzahl für Getriebestufe

4.11.2 Settingdaten

4.11.2.1 Allgemeine Settingdaten

Nummer	Bezeichner: $SN_	Beschreibung
41010	JOG_VAR_INCR_SIZE	Größe des variablen Inkrements bei INC/Handrad
41050	JOG_CONT_MODE_LEVELTRIGGRD	JOG-kontinuierlich im Tippbetrieb
41100	JOG_REV_IS_ACTIVE	Umdrehungsvorschub bei JOG aktiv
41110	JOG_SET_VELO	JOG-Geschwindigkeit bei Linearachsen (für `G94`)
41120	JOG_REV_SET_VELO	JOG-Geschwindigkeit (für `G95`)
41130	JOG_ROT_AX_SET_VELO	JOG-Geschwindigkeit bei Rundachsen
41200	JOG_SPIND_SET_VELO	JOG-Geschwindigkeit für die Spindel

4.11.3 Signale

4.11.3.1 Signale von NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Handrad 1 wird bewegt	DB10.DBB68	DB2700.DBB12
Handrad 2 wird bewegt	DB10.DBB69	DB2700.DBB13
Handrad 3 wird bewegt	DB10.DBB70	-
Kanalnummer Geometrie-Achse Handrad 1, 2, 3	DB10.DBB97/98/99	DB1900.DBB1003
Achsnummer für Handrad 1, 2, 3, Handrad angewählt und Maschinenachse	DB10.DBB100/101/102	DB1900.DBB1004

4.11.3.2 Signale an BAG

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Betriebsart JOG	DB11.DBX0.2	DB3000.DBX0.2

4.11.3.3 Signale von BAG

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Aktive Betriebsart JOG	DB11.DBX4.2	DB3100.DBX0.2

4.11.3.4 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
DRF aktivieren	DB21, ...DBX0.3	DB3200.DBX0.3
Handrad aktivieren (3, 2, 1)	DB21, ...DBX12.0-2 DB21, ...DBX16.0-2 DB21, ...DBX20.0-2	DB3200.DBX1000.0-1 DB3200.DBX1004.0-1 DB3200.DBX1008.0-1
Verfahrtastensperre	DB21, ...DBX12.4 DB21, ...DBX16.4 DB21, ...DBX20.4	DB3200.DBX1000.4 DB3200.DBX1004.4 DB3200.DBX1008.4
Eilgangüberlagerung	DB21, ...DBX12.5 DB21, ...DBX16.5 DB21, ...DBX20.5	DB3200.DBX1000.5 DB3200.DBX1004.5 DB3200.DBX1008.5
Verfahrtasten plus und Verfahrtasten minus	DB21, ...DBX12.6/7 DB21, ...DBX16.6/7 DB21, ...DBX20.6/7	DB3200.DBX1000.6/7 DB3200.DBX1004.6/7 DB3200.DBX1008.6/7
Geometrieachse Maschinenfunktion INC1 ... kontinuierlich	DB21, ...DBB13/17/21	DB3200.DBB1001/1005/ 1009

4.11.3.5 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
DRF angewählt	DB21, ...DBX24.3	DB1700.DBX0.3
Handrad aktiv (3, 2, 1)	DB21, ...DBX40.0-2 DB21, ...DBX46.0-2 DB21, ...DBX52.0-2	DB3300.DBX1000.0-1 DB3300.DBX1004.0-1 DB3300.DBX1008.0-1
Fahranforderung Geometrieachse	DB21, ...DBX40.4/5 DB21, ...DBX46.4/5 DB21, ...DBX52.4/5	DB3300.DBX1000.4/5 DB3300.DBX1004.4/5 DB3300.DBX1008.4/5
Fahrbefehl plus und Fahrbefehl minus	DB21, ...DBX40.6/7 DB21, ...DBX46.6/7 DB21, ...DBX52.6/7	DB3300.DBX1000.6/7 DB3300.DBX1004.6/7 DB3300.DBX1008.6/7
Geometrieachse aktive Maschinenfunktion INC1 ... kontinuierlich	DB21, ...DBB41/47/53	DB3300.DBB1001/1005/ 1009
Handradüberlagerung aktiv, für Bahnachsen	DB21, ...DBX33.3	DB3300.DBX1.3
Handrad 1 als Konturhandrad aktivieren Handrad 2 als Konturhandrad aktivieren Handrad 3 als Konturhandrad aktivieren	DB21, ...DBX30.0 DB21, ...DBX30.1 DB21, ...DBX30.2	DB3200.DBX14.0 DB3200.DBX14.1
Simulation Konturhandrad ein	DB21, ...DBX30.3	DB3200.DBX14.3
Negative Richtung Simulation Konturhandrad	DB21, ...DBX30.4	DB3200.DBX14.4
Handrad 1 als Konturhandrad aktiv Handrad 2 als Konturhandrad aktiv Handrad 3 als Konturhandrad aktiv	DB21, ...DBX37.0 DB21, ...DBX37.1 DB21, ...DBX37.2	DB3300.DBX5.0
Handrad 1 als Konturhandrad aktiv Handrad 2 als Konturhandrad aktiv Handrad 3 als Konturhandrad aktiv	DB21, ...DBX100.5 DB21, ...DBX101.5 DB21, ...DBX102.5	DB3300.DBX5.1
Fahranforderung Orientierungsachse	DB21, ...DBX332.4/5 DB21, ...DBX336.4/5 DB21, ...DBX340.4/5	-

4.11.3.6 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Vorschub-/Spindelkorrektur	DB21, ...DBB0	DB380x.DBB0
Korrektur wirksam	DB21, ...DBX1.7	DB380x.DBX1.7
Axiales Restweglöschen	DB21, ...DBX2.2	DB380x.DBX2.2
Handrad aktivieren (1, 2, 3)	DB21, ...DBX4.0-2	DB380x.DBX4.0-1
Verfahrtastensperre	DB21, ...DBX4.4	DB380x.DBX4.4
Eilgangüberlagerung	DB21, ...DBX4.5	DB380x.DBX4.5
Verfahrtasten plus und Verfahrtasten minus	DB21, ...DBX4.6/7	DB380x.DBX4.6/7
Maschinenfunktion: 1 INC, 10 INC, 100 INC, 1000 INC, 10000 INC, Var. INC, kontinuierlich	DB21, ...DBX5.0-6	DB380x.DBX5.0-6
Aktiviere Festpunktfahren in JOG (binärcodiert)	DB21, ...DBX13.0-2	DB380x.DBX1001.0-2

4.11.3.7 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Position erreicht mit Genauhalt grob/fein	DB21, ...DBX60.6/7	DB390x.DBX0.6/7
Handrad aktiv (1, 2, 3)	DB21, ...DBX64.0-2	DB390x.DBX4.0/1
Fahranforderung plus und minus Achse/Spindel	DB21, ...DBX64.4/5	DB390x.DBX4.4/5
Fahrbefehl plus und Fahrbefehl minus	DB21, ...DBX64.6/7	DB390x.DBX4.6/7
Aktive Maschinenfunktion INC1 ... kontinuierlich	DB21, ...DBB65	DB390x.DBB5
Handradüberlagerung aktiv, für Positionierachsen und konkurrierende Positionierachsen	DB21, ...DBX62.1	DB390x.DBX2.1
Festpunktfahren in JOG aktiv (binärcodiert)	DB21, ...DBX75.0-2	DB390x.DBX1001.0-2
Festpunkt erreicht (binärcodiert)	DB21, ...DBX75.3-5	DB390x.DBX1001.3-5

K3: Kompensationen

5

5.1 Einführung

Genauigkeitsfehler

Die Genauigkeit von Werkzeugmaschinen wird durch Abweichungen von der idealen Geometrie, Fehler in der Kraftübertragung und in den Messsystemen beeinträchtigt. Bei der Bearbeitung großer Werkstücke führen Temperaturunterschiede und mechanische Kräfte häufig zu hohem Präzisionsverlust.

Kompensationsfunktionen

Ein Teil dieser Abweichungen lässt sich in der Regel bei der Inbetriebnahme der Maschine messen und während des Betriebs, gestützt auf die Lageistwert-Geber und zusätzliche Sensorik, kompensieren. Dazu besitzen moderne CNC-Steuerungen achsspezifisch wirksame Kompensationsfunktionen.

Bei SINUMERIK 840D sl stehen folgende Kompensationsfunktionen zur Verfügung:

- Temperaturkompensation
- Losekompensation
- Interpolatorische Kompensation
  - Kompensation von Spindelsteigungsfehler und Messsystemfehler
  - Kompensation von Durchhang und Winkligkeitsfehler
- Dynamische Vorsteuerung (Schleppfehler-Kompensation)
- Reibkompensation (Quadrantenfehler-Kompensation)
  - Konventionelle Reibkompensation
  - Quadrantenfehlerkompensation mit neuronalen Netzen
- Elektronischer Gewichtsausgleich

Parametrierung

Die Kompensationsfunktionen lassen sich für jede Maschine mit Hilfe von achsspezifischen Maschinendaten individuell einstellen.

Wirksamkeit

Die Kompensationen wirken in allen Betriebsarten der Steuerung, sobald die Eingangsdaten zur Verfügung stehen. Alle Kompensationen, die von der Kenntnis des absoluten Lageistwerts abhängen, werden erst mit Erreichen des Referenzpunkts aktiviert.

Positionsanzeige

Die normale Ist- und Sollpositionsanzeige berücksichtigt die Kompensationswerte nicht und zeigt die Positionswerte einer "idealen Maschine" an. Die Kompensationswerte werden im Bedienbereich "Diagnose" im Bild "Service Achsen" angezeigt.

5.2 Temperaturkompensation

5.2.1 Funktionsbeschreibung

Verformung durch Temperatureinfluss

Durch Wärmeeinfluss von den Antrieben oder aus der Umgebung (z. B. Sonneneinstrahlung, Luftzug) kommt es zu einer Ausdehnung des Maschinenbetts und der Maschinenteile. Diese Ausdehnung ist u. a. abhängig von der Temperatur und von der Wärmeleitfähigkeit der Maschinenteile.

Auswirkungen

Aufgrund der Wärmeausdehnung der Maschinenteile ändern sich abhängig von der Temperatur die Istpositionen der einzelnen Achsen. Dies wirkt sich negativ auf die Genauigkeit der bearbeiteten Werkstücke aus.

Temperaturkompensation

Durch Aktivierung der Funktion "Temperaturkompensation" können Istwertveränderungen aufgrund von Temperatureinflüssen achsspezifisch kompensiert werden.

Sensorik

Für die Temperaturkompensation werden in der Regel neben den Lageistwerten von den vorhandenen Messgebern noch mehrere Temperatursensoren zur Erfassung eines Temperaturprofils benötigt.

Da die temperaturbedingten Änderungen relativ langsam ablaufen, kann die Erfassung und Vorverarbeitung des Temperaturprofils durch die PLC beispielsweise im Minutentakt erfolgen.

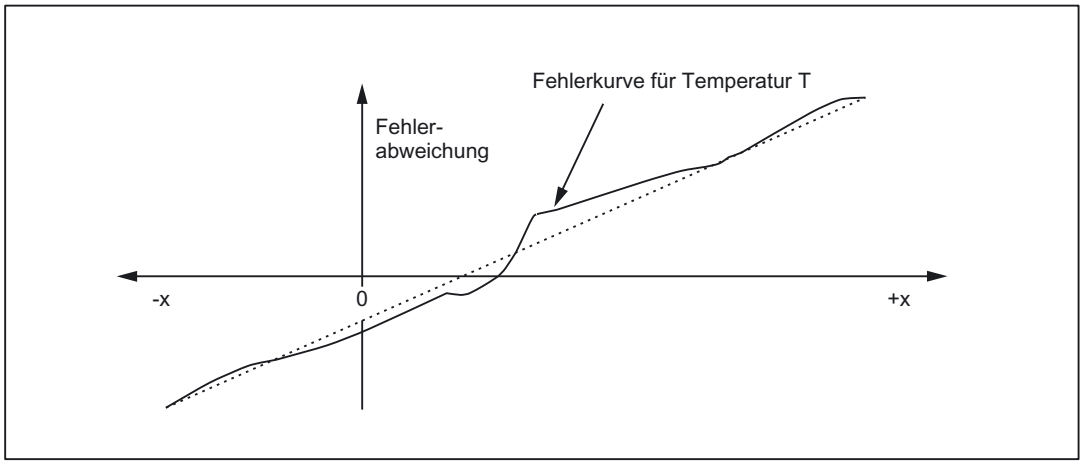
Fehlerkurven

Für die Temperaturkompensation sind bei gegebener Temperatur (T) die Istwertverschiebungen über den Positionierbereich der Achse zu messen und grafisch aufzutragen. Damit erhält man für diesen Temperaturwert eine zugehörige Fehlerkurve. Derartige Fehlerkurven sind für verschiedene Temperaturen zu ermitteln.

Fehlerkurvenverlauf

Wählt man einen Achspositionsbezugspunkt P_0, so beobachtet man bei Temperaturänderung eine Verschiebung des Bezugspunkts (entspricht dem "positionsunabhängigen Anteil" der Temperaturkompensation) und aufgrund der Längenänderungen eine zusätzliche Verschiebung der anderen Positionspunkte, die mit dem Abstand zum Bezugspunkt wächst (entspricht dem "positionsabhängigen Anteil" der Temperaturkompensation).

Für eine gegebene Temperatur T ist die Fehlerkurve in der Regel ausreichend genau durch eine Gerade darstellbar, deren Geradensteigung und Lage der Bezugsposition von der Temperatur abhängen:



Kompensationsgleichung

Der Kompensationswert ΔK_x berechnet sich aus der aktuellen Istposition P_x dieser Achse und der Temperatur T nach folgender Gleichung:

$\Delta K_x = K_0 (T) + \tan\beta (T) * (P_x - P_0)$

Dabei bedeuten:

ΔK_x:	Temperaturkompensationswert der Achse an der Position P_x
K_0:	positionsunabhängiger Temperaturkompensationswert der Achse
P_x:	Istposition der Achse
P_0:	Bezugsposition der Achse
tanβ:	Koeffizient für die positionsabhängige Temperaturkompensation (entspricht der Steigung der angenäherten Fehlergerade)

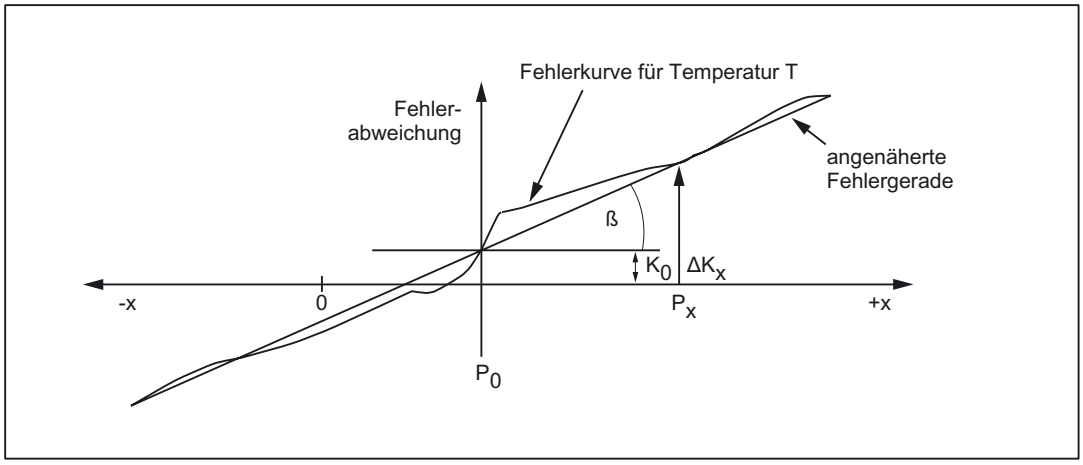


Bild 5-1 Angenäherte Fehlergerade für Temperaturkompensation

Wirksamkeit

Damit die Temperaturkompensation wirksam ist, müssen folgende Bedingungen erfüllt sein:

1. Der Kompensationstyp ist angewählt (MD32750, siehe "Temperaturkompensationstyp und Aktivierung (Seite 348)").
2. Die zum Kompensationstyp zugehörigen Parameter sind vorgegeben (siehe "Temperaturabhängige Parameter (Seite 348)").
3. Die Achse ist referiert:

   DB31, ... DBX60.4 bzw. 60.5 =1 (Referiert/Synchronisiert 1 bzw. 2)

Sobald diese Bedingungen erfüllt sind, wird in allen Betriebsarten der zum aktuellen Positionsistwert zugehörige Temperaturkompensationswert additiv auf den Sollwert aufgeschaltet und von der Maschinenachse verfahren. Bei positivem Kompensationswert ΔK_x fährt die Achse in negativer Richtung.

Falls anschließend die Referenzposition z. B. wegen Überschreiten der Encoderfrequenz wieder verloren geht (DB31, ... DBX60.4 bzw. 60.5 = 0), wird die Kompensationsverarbeitung ausgeschaltet.

Takt

Die Ermittlung der Kompensationswerte erfolgt im Interpolationstakt.

Anzeige

Der zur aktuellen Istposition gehörige Summen-Kompensationswert aus Temperatur- und Durchhangkompensation wird im Bedienbereich "Diagnose" im Bild "Service Achsen" angezeigt.

Parameteranpassungen bei Temperaturänderungen

Da die angenäherte Fehlergerade nur für den momentanen Temperaturwert gilt, müssen bei steigender oder fallender Temperatur die Parameter der sich neu ergebenden Fehlergeraden wieder an den NCK übergeben werden. Nur so ist sichergestellt, dass die Wärmeausdehnungen immer richtig kompensiert werden.

Da sich bei Änderung der Temperatur T die von ihr abhängigen Parameter (K_0, tanβ und P_0) ändern, können diese von der PLC oder über Synchronaktionen stets überschrieben werden.

Der Maschinenhersteller hat somit die Möglichkeit, die mathematischen und technologischen Zusammenhänge zwischen den Achspositionen und den Temperaturwerten mit Hilfe eines PLC-Anwenderprogramms abzubilden und somit die jeweiligen Parameter der Temperaturkompensation zu berechnen. Die Übertragung der Temperaturparameter zur NCK erfolgt mit Hilfe der variablen Dienste FB2 (GET) "Daten lesen" und FB3 (PUT) "Daten schreiben".

Nähere Informationen bzgl. Handhabung und Versorgung von FB2 und FB3 siehe:
Literatur:
Funktionshandbuch Grundfunktionen; PLC-Grundprogramm (P3)

Kompensationswert glätten

Damit es bei sprungförmigen Änderungen der Temperaturkompensationsparameter weder zu einer Überlastung der Maschine noch zu einem Ansprechen von Überwachungen kommt, werden die Kompensationswerte steuerungsintern auf mehrere IPO-Takte verteilt, sobald sie den maximalen Kompensationswert pro IPO-Takt (MD32760, siehe "Maximaler Kompensationswert pro IPO-Takt (Seite 349)") überschreiten.

5.2.2 Inbetriebnahme

5.2.2.1 Temperaturabhängige Parameter

SD43900, SD43910, SD43920

Für jede Achse können Fehlerkurven für verschiedene Temperaturen definiert werden. Für jede Fehlerkurve sind die folgenden Parameter zu bestimmen und als Settingdaten zu hinterlegen:

- Positionsunabhängiger Temperaturkompensationswert K_0:

  SD43900 $SA_TEMP_COMP_ABS_VALUE
- Bezugsposition P_0 für die positionsabhängige Temperaturkompensation:

  SD43920 $SA_TEMP_COMP_REF_POSITION
- Steigungswinkel tanβ für die positionsabhängige Temperaturkompensation:

  SD43910 $SA_TEMP_COMP_SLOPE

5.2.2.2 Temperaturkompensationstyp und Aktivierung

MD32750

Die Auswahl des Temperaturkompensationstyps und die Aktivierung der Temperaturkompensation erfolgt über das achsspezifische Maschinendatum:

MD32750 $MA_TEMP_COMP_TYPE (Temperaturkompensationstyp)

Bit	Wert	Bedeutung	zugehörige Parameter
0	Positions**un**abhängige Temperaturkompensation		SD43900
	0	nicht aktiv	
	1	aktiv	
1	Positionsabhängige Temperaturkompensation		SD43920, SD43910
	0	nicht aktiv	
	1	aktiv	
2	Temperaturkompensation in Werkzeugrichtung		MD20390 $MC_TOOL_TEMP_COMP_ON (Aktivierung Temperaturkomp. Werkzeugl.)
	0	nicht aktiv	
	1	aktiv	

5.2.2.3 Maximaler Kompensationswert pro IPO-Takt

MD32760

Der pro IPO-Takt maximal mögliche Kompensationswert, d. h. die durch die Temperaturkompensation in einem IPO-Takt maximal verfahrbare Strecke, wird begrenzt durch das Maschinendatum:

MD32760 $MA_COMP_ADD_VELO_FACTOR (Geschwindigkeitsüberhöhung durch Kompensation)

Der angegebene Wert wirkt als Faktor und bezieht sich auf die maximale Achsgeschwindigkeit (MD32000 $MA_MAX_AX_VELO).

Durch MD32760 wird auch die Steigung der Fehlergeraden (tanß) der Temperaturkompensation maximal begrenzt.

5.2.3 Beispiel

5.2.3.1 Inbetriebnahme der Temperaturkompensation für die Z-Achse einer Drehmaschine

Nachfolgend wird die Vorgehensweise für die Inbetriebnahme der Temperaturkompensation anhand eines Beispiels (Z-Achse einer Drehmaschine) erläutert.

Fehlerverlauf der Z-Achse ermitteln

Um den temperaturbedingten Fehlerverlauf der Z-Achse zu ermitteln, wird wie folgt vorgegangen:

- Gleichmäßiges Erwärmen durch Verfahren im gesamten Verfahrbereich der Z-Achse (im Beispiel: von 500 mm bis 1500 mm)
- Vermessen der Achsposition in Schritten von 100 mm
- Messen der aktuellen Temperatur an der Kugelrollspindel
- Durchführung des Verfahr-Messzyklus alle 20 Minuten

Anhand der aufgenommenen Daten können die mathematischen und technologischen Zusammenhänge und daraus die Parameter der Temperaturkompensation abgeleitet werden. Im folgenden Bild sind die ermittelten Fehlerabweichungen für eine bestimmte Temperatur, bezogen auf die von der NC angezeigte Istposition der Z-Achse, grafisch dargestellt.

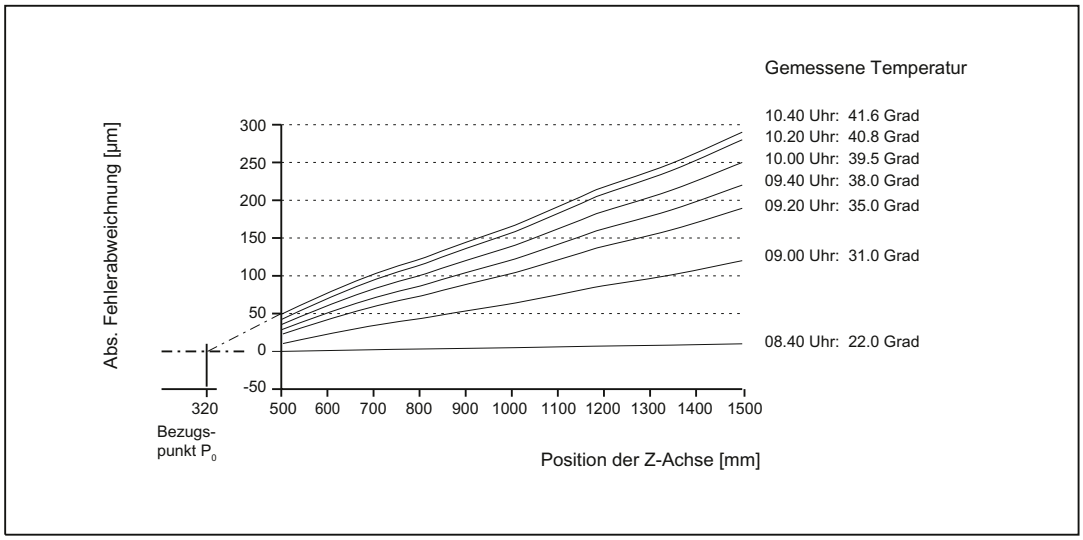


Bild 5-2 Ermittelte Fehlerkurven der Z–Achse

Parameter bestimmen

Anhand der ermittelten Messergebnisse (siehe vorhergehendes Bild) sind nun die Parameter der Temperaturkompensation zu bestimmen.

Bezugsposition P_0

Wie vorhergehendes Bild zeigt, gibt es prinzipiell zwei Varianten für die Parametrierung der Bezugsposition P_0:

1. $P_0 = 0$ mit positionsunabhängigem Temperaturkompensationswert $K_0 \neq 0$
2. $P_0 \neq 0$ mit positionsunabhängigem Temperaturkompensationswert $K_0 = 0$

In diesen Fall wird die Variante 2 gewählt, womit der positionsunabhängige Temperaturkompensationswert immer 0 ist. Somit besteht der Temperaturkompensationswert nur aus der positionsabhängigen Komponente.

Es ergeben sich folgende Parameter:

- MD32750 $MA_TEMP_COMP_TYPE = 2

  (nur positionsabhängige Temperaturkompensation aktiv)

- P_0 = 320 mm → SD43920 $SA_TEMP_COMP_REF_POSITION = 320

  (Bezugsposition für positionsabhängige Temperaturkompensation)

Koeffizient tanβ (T)

Um die Abhängigkeit des Koeffizienten tanβ der positionsabhängigen Temperaturkompensation von der Temperatur zu ermitteln, wird die Fehlerkurvensteigung über der gemessenen Temperatur grafisch aufgetragen:

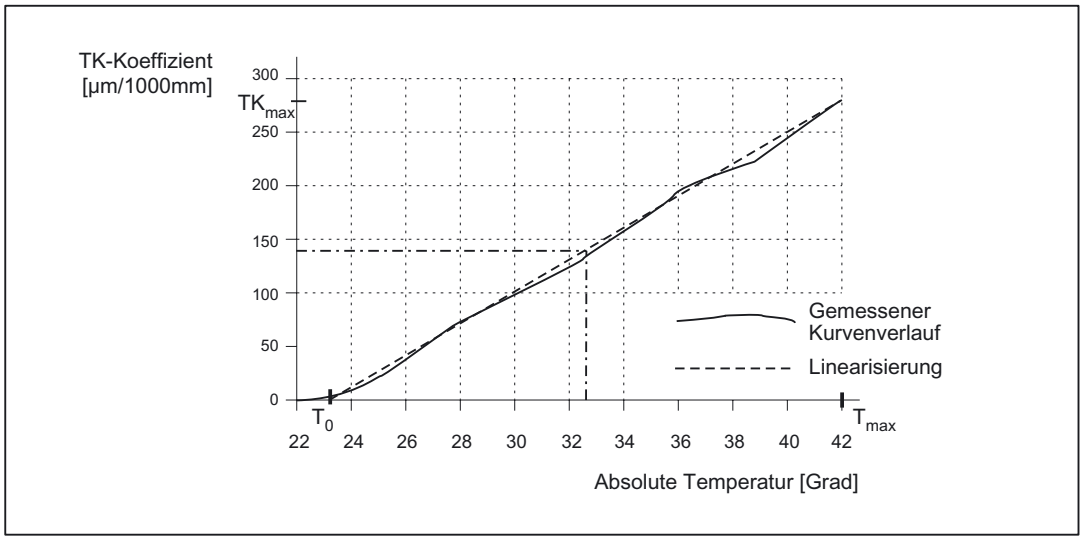


Bild 5-3 Verlauf des Koeffizienten tanβ in Abhängigkeit von der gemessenen Temperatur T

Bei entsprechender Linearisierung ergibt sich für den Koeffizienten tanβ folgende Abhängigkeit von T:

$\tan\beta(T) = (T - T_0) * TK_{max} * 10^{-6} / (T_{max} - T_0)$

mit

T_0 = Temperatur, bei der der positionsabhängige Fehler = 0 ist; [Grad]

T_{max} = maximal gemessene Temperatur; [Grad]

TK_{max} = Temperaturkoeffizient bei T_{max}; [µm/1000 mm]

Entsprechend den Werten aus obigem Bild somit:

T_0 = 23°

T_{max} = 42°

TK_{max} = 270 µm/1000 mm

Daraus ergibt sich tanß (T) zu:

$\tan\beta(T) = (T - 23)\ [Grad] * 270\ [µm/1000\ mm] * 10^{-6} / (42 - 23)\ [Grad]$
$= (T - 23)\ [Grad] * 14{,}21\ [µm/1000\ mm] * 10^{-6}$

Beispiel:

Bei einer Temperatur T = 32,3 Grad ergibt sich: tanβ = 0,000132

PLC-Anwenderprogramm

Im PLC-Anwenderprogramm ist nach obiger Formel der der gemessenen Temperatur entsprechende Koeffizient tanβ (T) zu berechnen und in folgendes Settingdatum der NCK zu schreiben:

SD43910 $SA_TEMP_COMP_SLOPE (Steigungswinkel für positionsabhängige Temperaturkompensation)

Nach obigem Beispiel:

SD43910 $SA_TEMP_COMP_SLOPE = 0,000132

5.3 Losekompensation

5.3.1 Funktionsbeschreibung

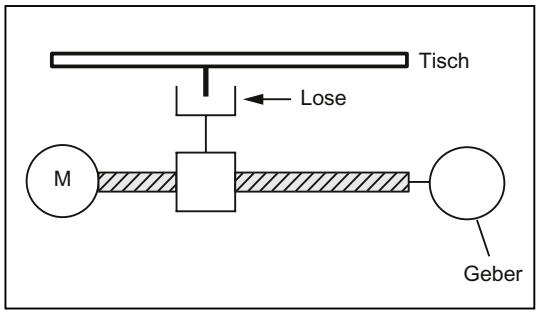
Mechanische Lose

Bei der Kraftübertragung zwischen einem bewegten Maschinenteil und seinem Antrieb (z. B. Kugelrollspindel) treten in der Regel kleine Lose auf, da eine völlig spielfreie Einstellung der Mechanik einen zu hohen Maschinenverschleiß zur Folge hätte. Außerdem kann zwischen dem Maschinenteil und dem Messsystem eine Lose auftreten.

Auswirkungen

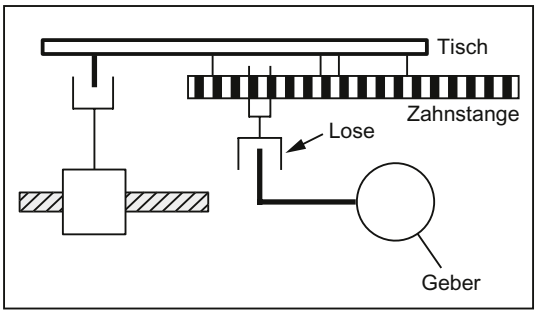
Bei Achsen/Spindeln mit indirekten Messsystemen führt mechanische Lose zu einer Verfälschung des Verfahrwegs. Eine Achse fährt beispielsweise bei Richtungsumkehr um den Betrag der Lose zu wenig oder zu viel:

Positive Lose (Normalfall)



Der Geber eilt dem Maschinenteil (z. B. Tisch) voraus. Da damit auch die vom Geber erfasste Istposition der tatsächlichen Istposition des Tisches vorauseilt, fährt der Tisch zu kurz.

Negative Lose



Der Geber hinkt dem Maschinenteil (z. B. Tisch) nach. Da damit auch die tatsächliche Istposition des Tisches der vom Geber erfassten Istposition vorauseilt, fährt der Tisch zu weit.

Losekompensation

Zur Kompensation der Lose wird der achsspezifische Istwert bei jedem Richtungswechsel der Achse/Spindel um den bei der Inbetriebnahme vorgegebenen achsspezifischen Losebetrag (MD32450, siehe "Umkehrlose (Seite 353)") korrigiert.

Wirksamkeit

Die Losekompensation ist nach dem Referenzpunktfahren immer in allen Betriebsarten aktiv.

Anzeige

Der zur aktuellen Istposition gehörige Kompensationswert wird im Bedienbereich "Diagnose" im Bild "Service Achsen" als Summen-Kompensationswert aus "SSFK" und "Losekompensation" angezeigt.

5.3.2 Inbetriebnahme

5.3.2.1 Umkehrlose

MD32450

Der Korrekturwert für die Losekompensation wird für jede Achse/Spindel eingetragen in das Maschinendatum:

MD32450 $MA_BACKLASH (Umkehrlose)

Bei positiver Lose (Normalfall) ist der Korrekturwert positiv einzugeben, bei negativer Lose negativ.

2. Messsystem

Sofern ein 2. Messsystem für die Achse/Spindel vorhanden ist, ist dafür ebenfalls ein Losekompensationswert einzugeben. Aufgrund der unterschiedlichen Anbringung des 2. Messsystems können hier andere Umkehrlose gegenüber dem 1. Messsystem bestehen.

Bei Umschaltung des Messsystems wird immer automatisch der zugehörige Kompensationswert aktiviert.

5.3.2.2 Bewertungsfaktor für Umkehrlose

MD32452

Die Umkehrlose kann parametersatzabhängig durch einen Faktor bewertet werden.

Dieser Bewertungsfaktor wird durch das folgende Maschinendatum zwischen 0.01 und 100.0 eingestellt:

MD32452 $MA_BACKLASH_FACTOR (Bewertungsfaktor für Umkehrlose)

Die Vorbesetzung ist 1.0.

Anwendung: z. B. Ausgleich der getriebestufenabhängigen Lose.

5.3.2.3 Schrittweises Aufschalten der Losekompensation

MD36500

Dem Anwender wird die Möglichkeit geboten, den Losekompensationswert bei Richtungsumkehr der betreffenden Achse in mehreren Teilstücken aufzuschalten. Damit wird vermieden, dass ein Sollwertsprung auf den Achsen zu entsprechenden Fehlern führt.

Der Inhalt des folgenden achsspezifischen Maschinendatums bestimmt die Schrittweite, mit der der Losekompensationswert (MD32450) aufgeschaltet wird:

MD36500 $MA_ENC_CHANGE_TOL (Maximale Toleranz bei Lageistwertumschaltung)

Es ist zu beachten, dass die Losekompensation erst nach <n> Servotakten eingerechnet ist (<n> = MD32450 / MD36500). Eine zu große Zeitspanne kann zur Auflösung von Stillstandsüberwachungsalarm führen.

Ist MD36500 > MD32450, wird die Kompensation in einem Servotakt ausgeführt.

5.4 Interpolatorische Kompensation

5.4.1 Allgemeine Eigenschaften

Funktion

Mit der "Interpolatorischen Kompensation" können positionsabhängige Maßabweichungen (Spindelsteigungs- und Messystemfehler, Durchhang- und Winkligkeitsfehler) korrigiert werden.

Die Korrekturwerte werden bei der Inbetriebnahme messtechnisch ermittelt und positionsbezogen in eine Kompensationstabelle hinterlegt. Im Betrieb wird die entsprechende Achse zwischen den Stützpunkten linear interpolierend korrigiert.

Methoden

Innerhalb der "Interpolatorischen Kompensation" wird zwischen den beiden folgenden Kompensationsmethoden unterschieden:

- Kompensation von Spindelsteigungsfehler und Messsystemfehler
- Kompensation von Durchhang und Winkligkeitsfehler

Viele Eigenschaften sind bei beiden Kompensationsmethoden identisch und daher nachfolgend für beide Methoden beschrieben.

Begriffe

Wichtige Begriffe der "Interpolatorischen Kompensation" sind:

- Kompensationswert

  Differenz zwischen der durch den Lageistwertgeber gemessenen Achsposition und der gewünschten programmierten Achsposition (= Achsposition der idealen Maschine). Häufig wird der Kompensationswert auch Korrekturwert genannt.

- Basisachse

  Achse, deren Positionssoll- oder -istwert für die Berechnung eines Kompensationswerts herangezogen wird.

- Kompensationsachse

  Achse, deren Positionssoll- oder -istwert durch einen Kompensationswert modifiziert wird.

- Stützpunkt

  Eine Position der Basisachse und der zugehörige Korrekturwert der Kompensationsachse.

- Kompensationstabelle

  Tabelle von Stützpunkten und Kompensationswerten (s. u.)

- Kompensationsbeziehung

  Zuordnung der Basisachse und der davon abhängigen Kompensationsachse sowie Verweis auf die zugehörige Korrekturtabelle.

Kompensationstabellen

Da sich die genannten Maßabweichungen auf die Genauigkeit der Werkstückbearbeitung direkt auswirken, sind sie durch entsprechende positionsabhängige Korrekturwerte zu kompensieren. Die Korrekturwerte werden anhand der gemessenen Fehlerkurve ermittelt und bei der Inbetriebnahme in Form von sog. Kompensationstabellen in die Steuerung eingegeben. Dabei ist für jede Kompensationsbeziehung eine eigene Tabelle zu erstellen.

Eingabe von Kompensationstabellen

Zuerst ist die Größe der Kompensationstabelle, d. h. die Anzahl der Stützpunkte, über Maschinendaten festzulegen. Nach dem nächsten POWER ON werden die Kompensationstabellen von der NC generiert und mit dem Wert "0" vorbesetzt.

Die Eingabe von Korrekturwerten sowie zusätzlichen Tabellenparametern in den Kompensationstabellen erfolgt mit speziellen Systemvariablen. Das Laden kann auf zwei verschiedene Arten durchgeführt werden:

- Durch Starten eines NC-Programms mit den Parameterwerten.
- Durch Übertragen der Kompensationstabellen aus einem PC über die serielle Schnittstelle der PCU.

Hinweis

Kompensationstabellen können über die serielle Schnittstelle der PCU aus dem Bedienbereich "Dienste" ausgegeben und nach dem Editieren zurückgeladen werden.

Hinweis

Das Laden der Kompensationstabellen ist nur möglich, wenn:

MD32700 $MA_ENC_COMP_ENABLE (Interpolatorische Kompensation) = 0

bzw.

MD32710 $MA_CEC_ENABLE (Freigabe der Durchhangkompensation) = 0

Da die Kompensationsdaten im statischen Anwenderspeicher abgelegt werden, bleiben die Werte auch bei ausgeschalteter Steuerung erhalten. Bei Bedarf können sie aktualisiert werden (z. B. nach Neuvermessung aufgrund von Alterung der Maschine).

⚠ VORSICHT

Bei Änderung des Maschinendatums:

- MD18342 $MN_MM_CEC_MAX_POINTS[<t>] (maximale Anzahl der Stützpunkte bei Durchhangkompensation); <t> = Index der Kompensationstabelle

oder

- MD38000 $MA_MM_ENC_COMP_MAX_POINTS (Anzahl der Stützpunkte bei interpolatorischer Kompensation)

wird bei Systemhochlauf automatisch der statische Anwenderspeicher neu eingerichtet.

Dabei werden alle Anwenderdaten des statischen NC-Speichers gelöscht (z. B. Antriebs- und HMI-Maschinendaten, Werkzeugkorrekturen, Teileprogramme, Kompensationstabellen usw.).

Literatur:
Funktionshandbuch Erweiterungsfunktionen; Speicherkonfiguration (S7)

Archivierung

Kompensationstabellen werden nicht mit der Serieninbetriebnahmedatei gesichert.

Kompensationstabellen müssen zur Archivierung über die serielle Schnittstelle der PCU ausgegeben werden. Dazu wird im Bedienbereich "Dienste" > "Daten-Aus" zwischen folgenden Kompensationen unterschieden:

- SSFK/Messsystemfehlerkompensation (%_N_AX_EEC_INI)
- Durchhang/Winkligkeitskompensation (%_N_AX_CEC_INI)
- Quadrantenfehlerkompensation (%_N_AX_QEC_INI)

Bei HMI-Advanced können die Kompensationstabellen auch als Archiv-Datei gesichert werden.

Linearinterpolation zwischen Stützpunkten

Die mit der Anfangs- und Endposition festgelegte zu kompensierende Verfahrstrecke wird in mehrere (Anzahl abhängig von der Fehlerkurvenform) gleichgroße Teilstrecken unterteilt (siehe folgendes Bild). Die Istpositionen, die diese Teilstrecken begrenzen, werden nachfolgend als "Stützpunkte" bezeichnet. Für jeden Stützpunkt ist bei der Inbetriebnahme der zugehörige Korrekturwert einzutragen. Zwischen 2 Stützpunkten wird der dort wirkende Korrekturwert durch eine **Linearinterpolation** aus den jeweiligen Korrekturwerten der benachbarten Stützpunkte gebildet (d. h. benachbarte Stützpunkte werden durch eine Strecke verbunden).

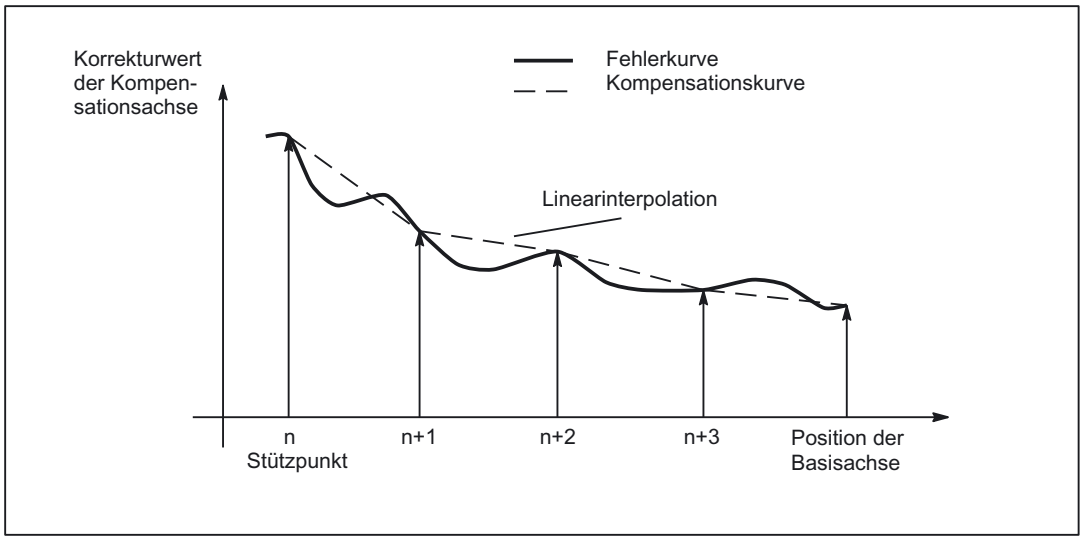


Bild 5-4 Linearinterpolation zwischen den Stützpunkten

Kompensationswert am Referenzpunkt

Die Kompensationstabelle sollte so aufgebaut sein, dass am Referenzpunkt der Kompensationswert den Wert "0" hat.

5.4.2 Kompensation von Spindelsteigungsfehler und Messsystemfehler

5.4.2.1 Funktionsbeschreibung

Spindelsteigungs- und Messsystemfehler

Das Messprinzip der "indirekten Messung" bei NC-gesteuerten Maschinen geht davon aus, dass an jeder beliebigen Stelle innerhalb des Verfahrbereichs die Steigung der Kugelrollspindel konstant ist, so dass die Istposition der Achse von der Position der Antriebsspindel abgeleitet werden kann (Idealfall). Durch Fertigungstoleranzen bei Spindeln kommt es jedoch zu mehr oder weniger großen Maßabweichungen (sog. Spindelsteigungsfehler).

Hierzu addieren sich noch die vom verwendeten Messsystem (unterschiedliche Teilungen) sowie dessen Anbringung an die Maschine bedingten Maßabweichungen (sog. Messsystemfehler) und weitere evtl. maschinenabhängige Fehlerquellen.

Kompensation

Bei der "Messsystemfehler-Kompensation" (nachfolgend als **MSFK** bezeichnet) sind Basis- und Kompensationsachse immer identisch. Sie ist daher eine **axiale Kompensation,** bei der eine Definition der Basis- und Kompensationsachse innerhalb der Kompensationstabelle nicht erforderlich ist.

Hinweis

Die Spindelsteigungsfehler-Kompensation ist ein Teil der Messsystemfehler-Kompensation.

Bei der MSFK wird im Interpolationstakt der achsspezifische Lageistwert um den zugehörigen Korrekturwert verändert und von der Maschinenachse unmittelbar verfahren. Ein positiver Korrekturwert führt zu einer Bewegung der zugehörigen Maschinenachse in negativer Richtung.

Die Größe des Korrekturwerts ist nicht begrenzt und wird auch nicht überwacht. Um infolge der Kompensation unzulässig hohe Geschwindigkeiten und Beschleunigungen der Maschinenachse zu vermeiden, sollten die Korrekturwerte entsprechend klein gewählt werden. Ansonsten können bei großen Korrekturwerten andere Achsüberwachungen zu Alarmmeldungen führen (z. B. Konturüberwachung, Drehzahlsollwertbegrenzung).

Falls die zu kompensierende Achse ein 2. Lagemesssystem besitzt, so ist für jedes Messsystem eine eigene Kompensationstabelle zu erstellen und zu aktivieren. Bei Umschaltung zwischen den beiden Messsystemen wird automatisch die jeweils zugehörige Tabelle verwendet.

Voraussetzungen / Wirksamkeit

Die MSFK ist erst unter folgenden Voraussetzungen wirksam:

- Die Kompensationswerte sind im statischen Anwenderspeicher abgelegt und wirksam (nach POWER ON).
- Die Funktion wurde für die jeweilige Maschinenachse aktiviert:

  MD32700 $MA_ENC_COMP_ENABLE [<e>] = 1

  mit: <e> = Lagemesssystem
  <e> = **0** Messsystem **1**
  <e> = **1** Messsystem **2**

- Die Achse wurde referiert:

  DB31, ... DBX60.4 bzw. 60.5 = 1 (Referiert/Synchronsiert 1 bzw. 2)

Sobald diese Bedingungen erfüllt sind, wird in allen Betriebsarten der achsspezifische Lageistwert um den zugehörigen Korrekturwert verändert und von der Maschinenachse unmittelbar verfahren.

Falls anschließend die Referenz z. B. wegen Überschreiten der Encoderfrequenz wieder verloren geht (DB31, ... DBX60.4 bzw. 60.5 = 0), wird die Kompensationsverarbeitung ausgeschaltet.

5.4.2.2 Inbetriebnahme

Anzahl der Kompensations-Stützpunkte (MD38000)

Für jede Maschinenachse sowie für jedes Messsystem (falls ein 2. Messsystem vorhanden ist) ist die Anzahl der reservierten Stützpunkte der Kompensationstabelle festzulegen und mit dem folgenden Maschinendatum der dafür notwendige Speicher zu reservieren:

MD38000 $MA_MM_ENC_COMP_MAX_POINTS[<e>,<AXi>]

mit: <e> = Lagemesssystem
<e> = 0 Messsystem 1
<e> = 1 Messsystem 2

<AXi> = Achsname (z. B. X1, Y1, Z1)

Die erforderliche Anzahl der Kompensations-Stützpunkte errechnet sich wie folgt:

$$\text{MM_ENC_COMP_MAX_POINTS [e,AXi]} = \frac{\$AA_ENC_COMP_MAX[e,AXi] - \$AA_ENC_COMP_MIN[e,AXi]}{\$AA_ENC_COMP_STEP[e,AXi]} + 1$$

Messsystemspezifische Parameter der Kompensationstabelle

Für jede Maschinenachse sowie für jedes Messsystem (falls ein 2. Messsystem vorhanden ist) sind die positionsbezogenen Korrekturen sowie zusätzliche Tabellenparameter in Form von Systemvariablen zu hinterlegen:

- **$AA_ENC_COMP[<e>,<N>,<AXi>]**

  (Korrekturwert für Stützpunkt N der Kompensationstabelle)

  <N> = Stützpunkt (Achsposition)

  Für jeden einzelnen Stützpunkt ist der jeweilige Korrekturwert in die Tabelle einzutragen.

  <N> ist durch die Anzahl der maximal möglichen Stützpunkte der jeweiligen Kompensationstabelle (MD38000 $MA_MM_ENC_COMP_MAX_POINTS) begrenzt:

  0 ≤ N ≤ MD38000 -1

  Die Größe des Korrekturwerts ist nicht begrenzt.

  Hinweis

  Der erste und letzte Korrekturwert bleibt über den gesamten Verfahrbereich aktiv, d. h. diese Korrekturwerte sollten den Wert "0" haben, falls sich die Kompensationstabelle nicht über den gesamten Verfahrbereich erstreckt.

- **$AA_ENC_COMP_STEP[<e>,<AXi>] (Stützpunktabstand)**

  Der Stützpunktabstand legt die Distanz zwischen den Korrekturwerten der jeweiligen Kompensationstabelle fest.

- **$AA_ENC_COMP_MIN[<e>,<AXi>] (Anfangsposition)**

  Die Anfangsposition ist die Achsposition, bei der die Kompensationstabelle für die betroffene Achse beginnt (≙ Stützpunkt 0).

  Der zur Anfangsposition zugehörige Korrekturwert ist $AA_ENC_COMP[<e>,**0**,<AXi>)].

  Für alle Positionen kleiner der Anfangsposition wird der Korrekturwert des Stützpunkts 0 verwendet (gilt nicht für Tabelle mit Modulofunktion).

- **$AA_ENC_COMP_MAX[<e>,<AXi>] (Endposition)**

  Die Endposition ist die Achsposition, bei der die Kompensationstabelle für die betroffene Achse endet (≙ Stützpunkt <k>).

  Der zur Endposition zugehörige Korrekturwert ist $AA_ENC_COMP[<e>,**<k>**,<AXi>)].

  Für alle Positionen größer der Endposition wird der Korrekturwert des Stützpunkts <k> verwendet (Ausnahme bei Tabelle mit Modulofunktion).

  Für den Stützpunkt <k> gelten folgende Randbedingungen:

  – bei k = MD38000 - 1:

    Die Kompensationstabelle wird voll genutzt!

  – bei k < MD38000 - 1:

    Die Kompensationstabelle wird nicht voll genutzt. Die in der Tabelle eingetragenen Korrekturwerte größer k sind wirkungslos.

  – bei k > MD38000 - 1:

    Die Kompensationstabelle wird steuerungsintern begrenzt, indem die Endposition verkleinert wird. Die Korrekturwerte größer k sind wirkungslos.

- **$AA_ENC_COMP_IS_MODULO[<e>,<AXi>] (Kompensation mit Modulofunktion)**

  Systemvariable zur Aktivierung/Deaktivierung der Kompensation mit Modulofunktion:

  – $AA_ENC_COMP_IS_MODULO[<e>,<AXi>] **= 0**: Kompensation **ohne** Modulofunktion

  – $AA_ENC_COMP_IS_MODULO[<e>,<AXi>] **= 1**: Kompensation **mit** Modulofunktion

  Bei Aktivierung der Kompensation mit Modulofunktion wird die Kompensationstabelle zyklisch wiederholt, d. h. auf den Korrekturwert an der Stelle $AA_ENC_COMP_MAX (≙ Stützpunkt $AA_ENC_COMP[<e>,<k>,<AXi>]) folgt unmittelbar sofort der Korrekturwert an der Stelle $AA_ENC_COMP_MIN (≙ Stützpunkt $AA_ENC_COMP[<e>,<0>,<AXi>]).

  Sinnvollerweise sollte bei Rundachsen mit Modulo 360° als Anfangsposition 0° ($AA_ENC_COMP_MIN) und als Endposition 360° ($AA_ENC_COMP_MAX) vorgegeben werden.

  Dabei sind diese beiden Korrekturwerte gleich einzugeben, sonst springt der Kompensationswert beim Übergang von MAX auf MIN und umgekehrt.

  ⚠VORSICHT

  Bei Eintrag der Korrekturwerte ist darauf zu achten, dass allen Stützpunkten innerhalb des festgelegten Bereichs ein Korrekturwert zugeordnet wird (d. h. keine Lücken entstehen). Ansonsten würde für diese Stützpunkte der Korrekturwert verwendet, der von früheren Einträgen an diesen Stellen übrig geblieben ist.

Hinweis

Tabellenparameter, welche Positionsangaben enthalten, werden beim Maßsystemwechsel (Änderung von MD10240 $MN_SCALING_SYSTEM_IS_METRIC) automatisch umgerechnet.

Die Positionsangaben werden immer im aktuellen Maßsystem interpretiert. Die Umrechnung muss extern vorgenommen werden.

Eine automatische Umrechnung der Positionsangaben kann wie folgt projektiert werden:

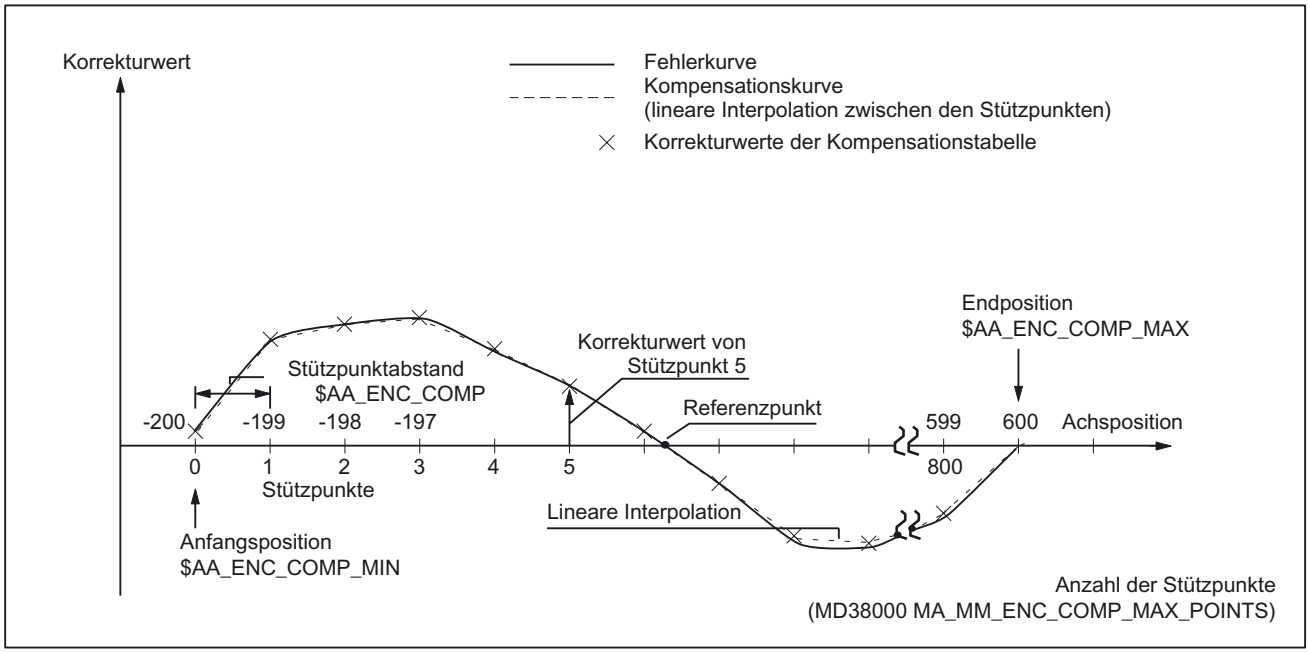
MD10260 $MN_CONVERT_SCALING_SYSTEM = 1

Die externe Umrechnung entfällt damit.

Literatur:
Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (G2)

5.4.2.3 Beispiel

Das nachfolgende Beispiel zeigt die Vorgabe der Kompensationswerte für die Maschinenachse X1.



Programmcode	Kommentar
%_N_AX_EEC_INI	
CHANDATA(1)	
$AA_ENC_COMP[0,0,X1]=0.0	; 1. Korrekturwert (Stützpunkt 0)
	; +0µm
$AA_ENC_COMP[0,1,X1]=0.01	; 2. Korrekturwert (Stützpunkt 1)
	; +10µm

```
Programmcode                                Kommentar
$AA_ENC_COMP[0,2,X1]=0.012                  ; 3. Korrekturwert (Stützpunkt 2)
                                            ; +12µm
$AA_ENC_COMP[0,800,X1]=-0.0                 ; Letzter Korrekturwert (Stützpunkt 800)
$AA_ENC_COMP_STEP[0,X1]=1.0                 ; Stützpunktabstand 1.0 mm
$AA_ENC_COMP_MIN[0,X1]=-200.0               ; Kompensation beginnt bei -200.0 mm
$AA_ENC_COMP_MAX[0,X1]=600.0                ; Kompensation endet bei +600.0 mm
$AA_ENC_COMP_IS_MODULO[0,X1]=0              ; Kompensation ohne Modulofunktion
M17
```

Für dieses Beispiel muss die projektierte Anzahl der Stützpunkte ≥ 801 sein:

MD38000 $MM_ENC_COMP_MAX_POINTS ≥ 801

Der Speicherbedarf im statischen Anwenderspeicher beträgt 6,4 kByte (8 Byte je Kompensationswert).

5.4.3 Kompensation von Durchhang und Winkligkeitsfehler

5.4.3.1 Funktionsbeschreibung

Durchhangfehler

Der Gewichtseinfluss kann zu einer stellungsabhängigen Verlagerung und Neigung der bewegten Teile führen, da sich Maschinenteile einschließlich der Führungen durchbiegen.

Auch große Werkstücke (z. B. Walzen) hängen unter ihrem eigenen Gewicht durch.

Winkligkeitsfehler

Falls Bewegungsachsen nicht genau im gewünschten Winkel (z. B. senkrecht) zueinander stehen, führt dies mit zunehmender Auslenkung aus der Null-Lage zu wachsenden Positionierfehlern.

Kompensation

Im Gegensatz zur MSFK muss bei der "Durchhangkompensation" bzw. "Winkligkeitsfehlerkompensation" Basis- und Kompensationsachse nicht identisch sein, weshalb in jeder Kompensationstabelle eine Achszuordnung erforderlich ist.

Um den Durchhang einer Achse (Basisachse) aufgrund des Eigengewichts zu korrigieren, ist die absolute Position einer anderen Achse (Kompensationsachse) zu beeinflussen. Die "Durchhangkompensation" ist daher eine **achsübergreifende Kompensation**.

Wie im folgenden Bild dargestellt, biegt sich die Auslegerachse desto mehr in die negative Z1-Achsrichtung durch, je weiter der Bearbeitungskopf in die negative Y1-Achsrichtung verfährt.

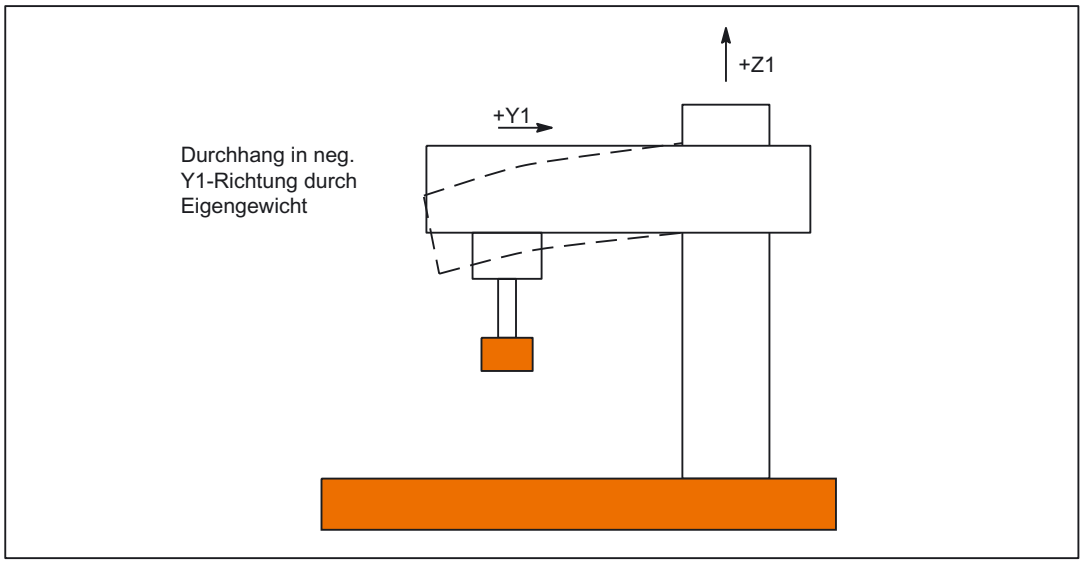


Bild 5-5 Beispiel für Durchhang durch Eigengewicht

Der Fehler ist in Form einer Kompensationstabelle aufzunehmen, die zu jeder Lageistposition der Y1-Achse einen Kompensationswert für die Z1-Achse enthält. In der Tabelle werden die Kompensationswerte lediglich für die Stützpunkte eingetragen.

Beim Verfahren der Y1-Achse ermittelt die Steuerung im Interpolationstakt den zugehörigen Korrekturwert der Z1-Achse, indem bei Positionen zwischen den Stützpunkten linear interpoliert wird. Dieser Korrekturwert wird als zusätzlicher Sollwert an die Lageregelung gegeben. Ein positiver Korrekturwert führt zu einer Bewegung der zugehörigen Maschinenachse in negativer Richtung.

Abhängig vom Anwendungsfall können für eine Achse auch mehrere Kompensationsbeziehungen definiert werden. Der Gesamtkorrekturwert ergibt sich dabei aus der Summe aller Kompensationswerte dieser Achse.

Einstellmöglichkeiten

Nachfolgend sind für die Durchhangkompensation die vielen Möglichkeiten und Einflüsse auf die Bildung des Korrekturwerts aufgelistet (siehe folgendes Bild).

1. Eine Achse kann als Eingangsgröße (Basisachse) für **mehrere** Kompensationstabellen definiert werden (einstellbar über Systemvariable).
2. Eine Achse kann als Empfänger der Ausgangsgröße (Kompensationsachse) von **mehreren** Kompensationstabellen definiert werden (einstellbar über Systemvariable). Der Gesamtkompensationswert ergibt sich aus der Summe der Einzelkorrekturwerte.

   Für die Anzahl der maximal möglichen Kompensationstabellen gelten folgende Festlegungen:

   – Maximale Anzahl der insgesamt für alle Achsen verfügbaren Tabellen:

     2 * maximale Achsanzahl des Systems

   – Anzahl der maximal auf eine Kompensationsachse wirkenden Tabellen:

     1 * maximale Achsanzahl des Systems
3. Eine Achse kann gleichzeitig Basis- und Kompensationsachse sein. Für die Berechnung der Korrekturwerte wird stets der programmierte (gewünschte) Positionssollwert verwendet.
4. Der Wirkungsbereich der Kompensation (Anfangs- und Endposition der Basisachse) sowie der Stützpunktabstand sind für jede Kompensationstabelle definierbar (einstellbar über Systemvariable).
5. Die Kompensation kann richtungsabhängig wirken (einstellbar über Systemvariable).
6. Jede Kompensationstabelle verfügt über eine Modulofunktion für zyklische Auswertung (einstellbar über Systemvariable).
7. Für jede Kompensationstabelle kann ein Wichtungsfaktor berücksichtigt werden, mit dem der Tabellenwert multipliziert wird (vorgebbar als Settingdatum; damit vom Teileprogramm, von der PLC oder Bediener jederzeit änderbar).
8. Mit der "Tabellenmultiplikation" kann jede Kompensationstabelle mit jeder beliebigen Kompensationstabelle (d. h. auch mit sich selbst) paarweise multipliziert werden. Die Verknüpfung der Multiplikation erfolgt über eine Systemvariable. Das Produkt wird additiv zum Summenkompensationswert der Kompensationsachse verrechnet.
9. Für die Aktivierung der Kompensation gibt es folgende Möglichkeiten:

   – Mit dem Maschinendatum:

     MD32710 $MA_CEC_ENABLE[<AXi>] (Freigabe der Durchhangkompensation)

     wird für die Maschinenachse <AXi> die Summe aller Kompensationsbeziehungen freigegeben.

   – Mit dem Settingdatum:

     SD41300 $SN_CEC_TABLE_ENABLE[<t>] (Vorbesetzung für Kompensationstabelle)

     wird die Auswertung der Kompensationstabelle[<t>] freigegeben.

     Hiermit können z. B. abhängig von der Bearbeitung die Kompensationsbeziehungen vom Teileprogramm oder PLC-Anwenderprogramm abgeändert werden (z. B. Tabellen umschalten).

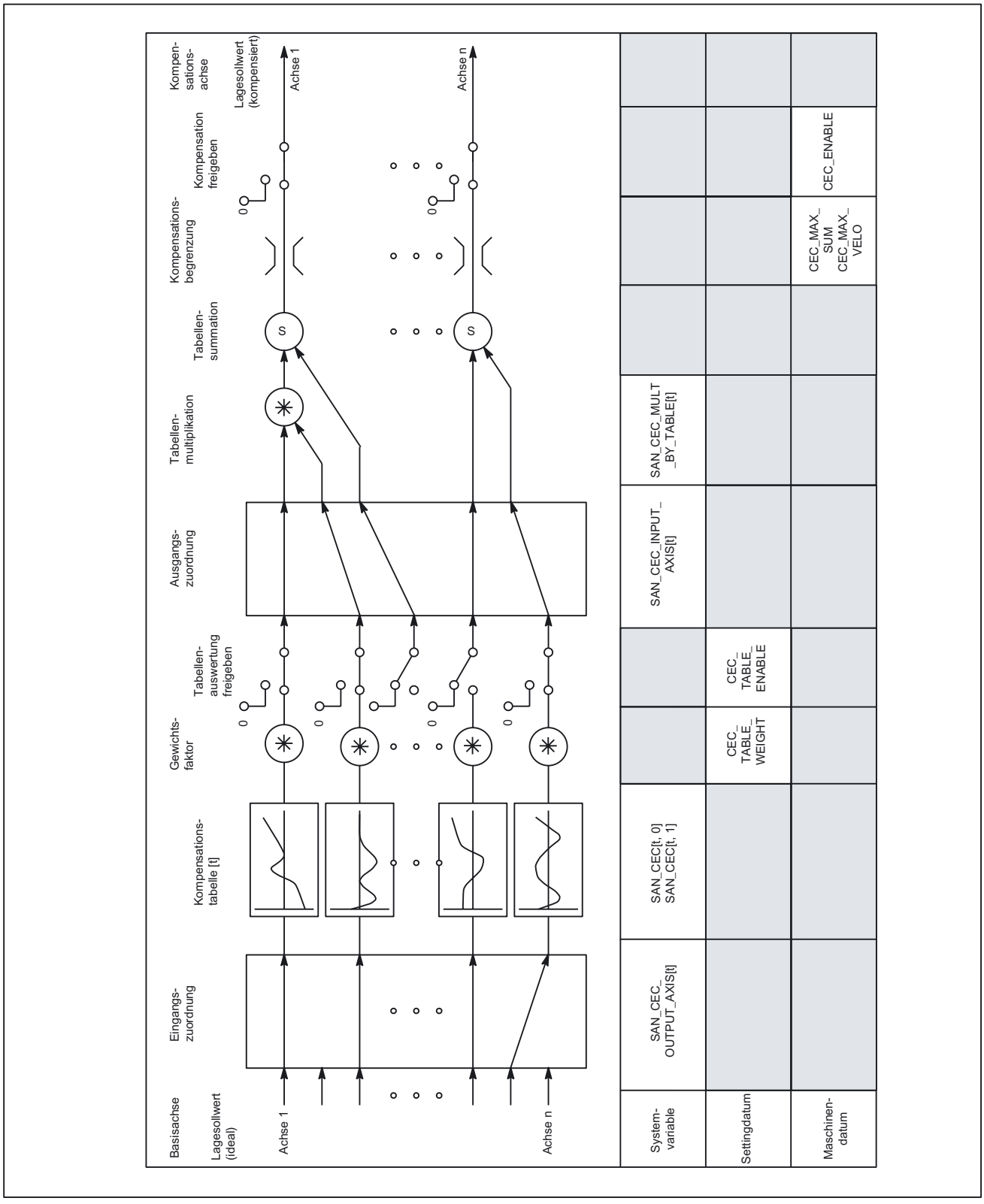


Bild 5-6 Bildung des Kompensationswertes bei Durchhangkompensation

Komplexe Kompensation

Mit der Möglichkeit, die Position einer Achse als Eingangsgröße (Basisachse) für mehrere Tabellen zu verwenden bzw. den Gesamt-Kompensationswert einer Achse von mehreren Kompensationsbeziehungen (Tabellen) abzuleiten sowie Tabellen zu multiplizieren, lassen sich vollwertige und komplexe Durchhang- und Winkligkeitsfehler-Kompensationen durchführen.

Außerdem sind damit auch unterschiedliche Fehlerquellen effizient zu behandeln. So kann z. B. eine Tabelle mit Modulo-Funktion für einen kurzwelligen sich wiederholenden Fehleranteil zusammen mit einer zweiten Tabelle ohne Modulo-Funktion für einen aperiodischen Fehleranteil auf dieselbe Achse wirken.

Auch sind mit dieser Funktion Spindelsteigungsfehler kompensierbar, indem für die Basis- und Kompensationsachse die identische Achse parametriert wird. Allerdings werden hier im Gegensatz zur MSFK Messsystemumschaltungen nicht automatisch berücksichtigt.

Voraussetzungen / Wirksamkeit

Die "Durchhangkompensation" ist erst unter folgenden Voraussetzungen wirksam:

- Die Option "Durchhangkompensation, mehrdimensional" ist freigegeben.
- Die Funktion wurde für die jeweilige Maschinenachse (Kompensationsachse) aktiviert:

  MD32710 $MA_CEC_ENABLE[<AXi>] = 1
- Die Kompensationswerte sind im statischen Anwenderspeicher abgelegt und wirksam (nach POWER ON).
- Die Auswertung der entsprechenden Kompensationstabelle wurde freigegeben:

  SD41300 $SN_CEC_TABLE _ENABLE[<t>] = 1
- Das aktuelle Messsystem der Basis- und Kompensationsachse wurde referenziert:

  DB31, ... DBX60.4 bzw. 60.5 = 1 (Referenziert/Synchronsiert 1 bzw. 2)

Sobald diese Bedingungen erfüllt sind, wird in allen Betriebsarten die Sollposition der Kompensationsachse abhängig von der Sollposition der Basisachse um den zugehörigen Korrekturwert verändert und von der Maschinenachse unmittelbar verfahren.

Falls anschließend die Referenz z. B. wegen Überschreiten der Encoderfrequenz wieder verloren geht (DB31, ... DBX60.4 bzw. 60.5 = 0), wird die Kompensationsverarbeitung ausgeschaltet.

5.4.3.2 Inbetriebnahme

Anzahl der Kompensations-Stützpunkte (MD18342)

Für jede Kompensationsbeziehung ist die Anzahl der benötigten Stützpunkte der Kompensationstabelle festzulegen und mit dem folgenden Maschinendatum der dafür notwendige Speicher zu reservieren:

MD18342 $MN_MM_CEC_MAX_POINTS[<t>] (maximale Anzahl der Stützpunkte bei Durchhangkompensation)

mit: <t> = Index der Kompensationstabelle
Zulässiger Bereich: 0 ≤ t < 2 * maximale Achsanzahl
t = 0: 1. Kompensationstabelle
t = 1: 2. Kompensationstabelle
…

Die erforderliche Anzahl der Kompensations-Stützpunkte errechnet sich wie folgt:

$$MM_CEC_MAX_POINTS\ [t] = \frac{\$AN_CEC_MAX\ [t] - \$AN_CEC_MIN\ [t]}{\$AN_CEC_STEP\ [t]} + 1$$

Tabellenparameter

Für jede Kompensationsbeziehung sind die positionsbezogenen Korrekturen sowie zusätzliche Tabellenparameter in Form von Systemvariablen zu hinterlegen:

- **$AN_CEC[<t>,<N>] (Korrekturwert für Stützpunkt <N> der Kompensationstabelle[<t>])**

  <N> = Stützpunkt (Position der Basisachse)

  Für jeden einzelnen Stützpunkt ist der jeweilige Korrekturwert der Kompensationsachse in die Tabelle einzutragen.

  <N> ist durch die Anzahl der maximal möglichen Stützpunkte der jeweiligen Kompensationstabelle (MD18342 $MN_MM_CEC_MAX_POINTS) begrenzt:

  0 ≤ N ≤ MD18342 - 1

- **$AN_CEC_INPUT_AXIS[<t>] (Basisachse)**

  Name der Maschinenachse, deren Sollwert als Eingang für die Kompensationstabelle[<t>] verwendet wird.

- **$AN_CEC_OUTPUT_AXIS[<t>] (Kompensationsachse)**

  Name der Maschinenachse, auf die der Ausgang der Kompensationstabelle[<t>] wirkt.

  Hinweis

  In mehrkanaligen Systemen müssen die "allgemeinen Achsbezeichner" AX1 ... vorgegeben werden, falls die Bezeichner von Maschinenachse und Kanalachse identisch sind.

- **$AN_CEC_STEP[<t>] (Stützpunktabstand)**

  Der Stützpunktabstand legt die Distanz der Eingangswerte für die Kompensationstabelle[<t>] fest.

- **$AN_CEC_MIN[<t>] (Anfangsposition)**

  Die Anfangsposition ist die Position der Basisachse, bei der die Kompensationstabelle[<t>] beginnt (≙ Stützpunkt 0).

  Der zur Anfangsposition zugehörige Korrekturwert ist $AN_CEC [<t>,0].

  Für alle Positionen kleiner als die Anfangsposition wird der Korrekturwert des Stützpunkts 0 verwendet (Ausnahme bei Tabelle mit Modulofunktion).

- **$AN_CEC_MAX[<t>] (Endposition)**

  Die Endposition ist die Position der Basisachse, bei der die Kompensationstabelle[<t>] endet (≙ Stützpunkt <k>).

  Der zur Endposition zugehörige Korrekturwert ist $AN_CEC [<t>,**<k>**].

  Für alle Positionen größer als die Endposition wird der Korrekturwert des Stützpunkts <k> verwendet (Ausnahme bei Tabelle mit Modulofunktion).

  Für den Stützpunkt k gelten folgende Randbedingungen:

  – bei k = MD18342 - 1:

    Die Kompensationstabelle wird voll genutzt!

  – bei k < MD18342 - 1:

    Die Kompensationstabelle wird nicht voll genutzt. Die eingetragenen Korrekturwerte größer k sind wirkungslos.

  – bei k > MD18342 – 1:

    Die Kompensationstabelle wird steuerungsintern begrenzt, indem die Endposition verkleinert wird. Die Korrekturwerte größer k sind wirkungslos.

- **$AN_CEC_DIRECTION[<t>] (Richtungsabhängige Kompensation)**

  Mit dieser Systemvariablen kann eingestellt werden, ob die Kompensationstabelle[<t>] für beide Verfahrrichtungen oder nur bei positiver bzw. negativer Verfahrrichtung der Basisachse wirkt:

  – $AN_CEC_DIRECTION[<t>] = 0:

    Tabelle wirkt für beide Verfahrrichtungen der Basisachse

  – $AN_CEC_DIRECTION[<t>] = 1:

    Tabelle wirkt nur in positiver Verfahrrichtungen der Basisachse

  – $AN_CEC_DIRECTION[<t>] = -1:

    Tabelle wirkt nur in negativer Verfahrrichtungen der Basisachse

  Anwendungsmöglichkeit:

  Mit Hilfe zweier Tabellen, von denen die eine in positiver, die andere in negativer Verfahrrichtung der gleichen Achse wirkt, lässt sich eine positionsabhängige Losekompensation verwirklichen.

- **$AN_CEC_MULT_BY_TABLE [<t>] (Tabellenmultiplikation)**

  Damit können die Kompensationswerte jeder Tabelle mit denen einer beliebigen Tabelle (auch mit sich selbst) multipliziert werden. Das Produkt wird als zusätzlicher Korrekturwert additiv zum Summenkompensationswert der Kompensationstabelle verrechnet.

  Syntax:

  $AN_CEC_MULT_BY_TABLE[<$t_1$>] = <$t_2$>

  mit:

  <t_1>= Index der Tabelle 1 der Kompensationsachse

  <t_2>= Nummer der Tabelle 2 der Kompensationsachse

  Es ist zu beachten, dass sich Nummer und Index der gleichen Tabelle unterscheiden! Allgemein gilt: Tabellennummer = Tabellenindex + 1

- **$AN_CEC_IS_MODULO[<t>] (Kompensation mit Modulofunktion)**

  Systemvariable zur Aktivierung/Deaktivierung der Kompensation mit Modulofunktion:

  – $AA_CEC_COMP_IS_MODULO[<t>] = **0**: Kompensation **ohne** Modulofunktion

  – $AA_CEC_COMP_IS_MODULO[<t>] = **1**: Kompensation **mit** Modulofunktion

  Bei Aktivierung der Kompensation mit Modulofunktion wird die Kompensationstabelle zyklisch wiederholt, d. h. auf den Korrekturwert an der Stelle $AN_CEC_MAX[<t>] (Stützpunkt $AN_CEC[<t>,<k>]) folgt unmittelbar der Korrekturwert an der Stelle $AN_CEC_MIN[<t>] (Stützpunkt $AN_CEC[<t>,0]).

  Sinnvollerweise sind diese beiden Korrekturwerte gleich einzugeben, sonst springt der Kompensationswert beim Übergang von MAX auf MIN und umgekehrt.

  Soll eine Modulokompensation mit einer Modulo-Rundachse als Basisachse realisiert werden, so muss die verwendete Kompensationstabelle ebenfalls modulo gerechnet werden.

  Beispiel:

  MD30300 $MA_IS_ROT_AX[AX1]=1 ; Rundachse

  MD30310 $MA_ROT_IS_MODULO[AX1]=1 ; Modulo 360°

  $AN_CEC_INPUT_AXIS[0]=AX1

  $AN_CEC_MIN[0]=0.0

  $AN_CEC_MAX[0]=360.0

  $AN_CEC_IS_MODULO[0]=1

Maßsystem

Tabellenparameter, welche Positionsangaben enthalten, werden beim Maßsystemwechsel (Änderung von MD10240 $MN_SCALING_SYSTEM_IS_METRIC) automatisch umgerechnet.

Die Positionsangaben werden immer im aktuellen Maßsystem interpretiert. Die Umrechnung muss extern vorgenommen werden.

Eine automatische Umrechnung der Positionsangaben kann wie folgt projektiert werden:

MD10260 $MN_CONVERT_SCALING_SYSTEM = 1

Mit dieser Einstellung wird das folgende axiale Maschinendatum wirksam:

MD32711 $MA_CEC_SCALING_SYSTEM_METRIC (Maßsystem der Durchhangkompensation)

Mit diesem Maschinendatum wird das Maßsystem für alle in dieser Achse wirksamen Tabellen festgelegt. Damit werden alle Positionsangaben zusammen mit dem sich errechnenden Gesamtkompensationswert in dem projektierten Maßsystem ausgewertet. Externe Umrechnungen der Positionsangaben sind bei einem Maßsystemwechsel nicht mehr erforderlich.

Überwachung

Um infolge der Durchhangkompensation unzulässig hohe Geschwindigkeiten und Beschleunigungen der Maschinenachse zu vermeiden, wird der Summenkompensationswert überwacht und auf einen Maximalwert begrenzt.

Der maximal mögliche Summenkompensationswert bei Durchhangkompensation wird achsspezifisch festgelegt mit dem Maschinendatum:

MD32720 $MA_CEC_MAX_SUM (Maximaler Kompensationswert bei Durchhangkompensation)

Ist der ermittelte Summenkompensationswert größer als der Maximalwert, wird ein entsprechender Alarm gemeldet. Die Programmbearbeitung wird nicht unterbrochen. Der als zusätzlicher Sollwert ausgegebene Kompensationswert wird auf den Maximalwert begrenzt.

Außerdem ist zusätzlich die Änderung des Summenkompensationswerts axial begrenzt:

MD32730 $MA_CEC_MAX_VELO (Geschwindigkeitsänderung bei Durchhangkompensation)

Der angegebene Wert wirkt als Faktor und bezieht sich auf die maximale Achsgeschwindigkeit (MD32000 $MA_MAX_AX_VELO).

Bei Überschreitung des Grenzwerts wird ein entsprechender Alarm gemeldet. Die Programmbearbeitung wird nicht unterbrochen. Die infolge der Begrenzung nicht abgefahrene Strecke wird nachgeholt, sobald sich der Kompensationswert wieder aus der Begrenzung löst.

5.4.3.3 Beispiele

Kompensationstabelle für Durchhangkompensation der Y1-Achse

Im nachfolgenden Beispiel ist die Kompensationstabelle für die Durchhangkompensation der Y1-Achse dargestellt. Abhängig von der Position der Y1-Achse wird auf die Z1-Achse ein Korrekturwert geschaltet. Hierzu wird die 1. Korrekturtabelle (<t> = 0) verwendet.

```
Programmcode                        Kommentar
%_N_NC_CEC_INI
CHANDATA(1)
$AN_CEC[0,0]=0                      ; 1.Korrekturwert (Stützpunkt 0)
                                    ; für Z1: ±0µm
$AN_CEC[0,1]=0.01                   ; 2.Korrekturwert (Stützpunkt 1)
                                    ; für Z1: +10µm
```

```
Programmcode                          Kommentar
$AN_CEC[0,2]=0.012                    ; 3.Korrekturwert (Stützpunkt 2)
                                      ; für Z1: +12µm
...
$AN_CEC[0,100]=0                      ; Letzter Korrekturwert (Stützpunkt 101)
                                      ; für Z1: ±0µm
$AN_CEC_INPUT_AXIS[0]=(AX2)           ; Basisachse Y1
$AN_CEC_OUTPUT_AXIS[0]=(AX3)          ; Kompensationsachse Z1
$AN_CEC_STEP[0]=8                     ; Stützpunktabstand 8.0mm
$AN_CEC_MIN[0]=-400.0                 ; Kompensation beginnt bei Y1=-400mm
$AN_CEC_MAX[0]=400.0                  ; Kompensation endet bei Y1=+400mm
$AN_CEC_DIRECTION[0]=0                ; Tabelle wirkt für beide Verfahrrichtungen von
                                        Y1.
$AN_CEC_MULT_BY_TABLE[0]=0
$AN_CEC_IS_MODULO[0]=0                ; Kompensation ohne Modulofunktion
M17                                   ;
```

Für dieses Beispiel muss die projektierte Anzahl der Stützpunkte ≥ 101 sein:

MD18342 $MN_MM_CEC_MAX_POINTS [0] ≥ 101

Der Speicherbedarf im statischen Anwenderspeicher beträgt mindestens 808 Byte (8 Byte je Kompensationswert).

Anwendungsfall für Tabellenmultiplikation

Nachfolgendes Beispiel für die Kompensation der Durchbiegung eines Fundaments zeigt einen Anwendungsfall für die Tabellenmultiplikation:

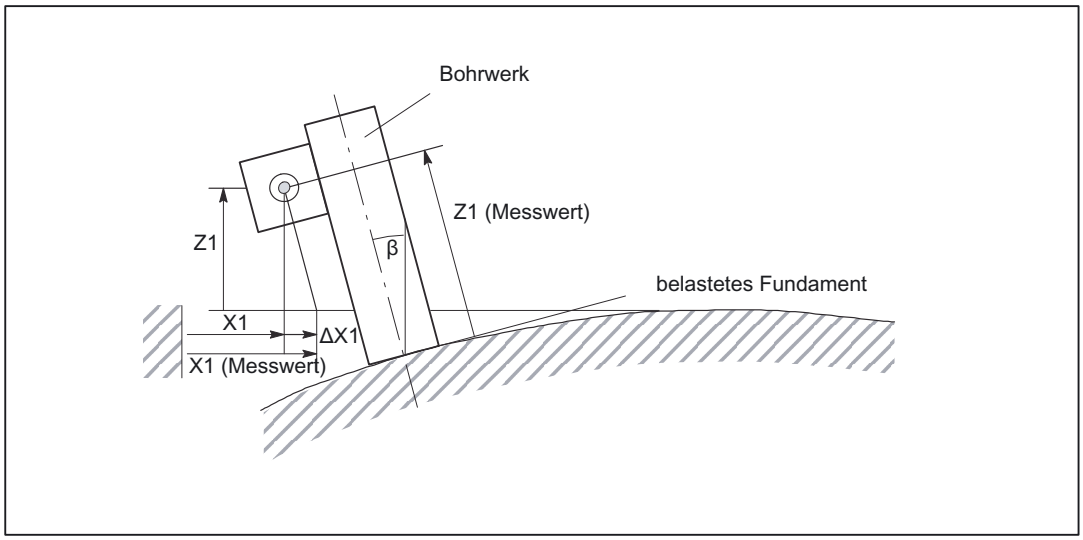


Bild 5-7 Kompensation der Durchbiegung eines Fundaments

Bei Großmaschinen kann die Durchbiegung des Fundaments zu Neigung der gesamten Maschine führen. Für das Bohrwerk in der Abbildung wird beispielsweise festgestellt, dass die Kompensation der X1-Achse sowohl von der Stellung der X1-Achse selbst abhängt (da diese den Neigungswinkel β bestimmt) als auch von der Höhe, in der sich der Bohrer befindet (d. h. Stellung der Z1-Achse).

Für eine Kompensation sind die Korrekturwerte der X1- und Z1-Achse nach folgender Gleichung multiplikativ zu verknüpfen:

$\Delta X1 = Z1 * \sin\beta(X1) \approx Z1 * \beta(X1)$

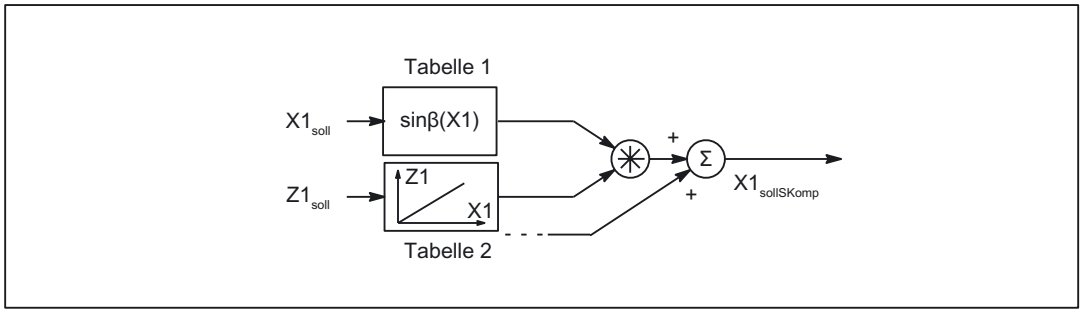


Bild 5-8 Tabellenmultiplikation

Kompensationstabelle 1 (Tabellenindex = 0) beschreibt die Rückwirkung der Achse X1 auf die Achse X1 (Sinus des positionsabhängigen Kippwinkels β(X1)).

Kompensationstabelle 2 (Tabellenindex = 1) beschreibt die Rückwirkung der Achse Z1 auf die Achse X1 (linear).

In Tabelle 1 ist die Multiplikation der Tabelle 1 (Index = 0) mit Tabelle 2 anzuwählen:

$AN_CEC_MULT_BY_TABLE[0] = 2

Eingabe der Kompensationswerte in Gitterstruktur

Für Flachbettmaschinen ergibt sich in der Praxis oft der Anwendungsfall, dass die Kompensationswerte des Durchhangs der Z-Achse an verschiedenen Punkten in Abhängigkeit von den X- und Y-Koordinaten gemessen werden. Unter diesen Voraussetzungen bietet sich die Eingabe der gemessenen Kompensationswerte nach einer gitterartigen Aufteilung an. An den Schnittpunkten des Gitters (X-Y-Ebene) befinden sich die Stützpunkte mit den jeweiligen Kompensationswerten. Kompensationswerte zwischen diesen Stützpunkten werden von der Steuerung linear interpoliert.

Im folgenden Beispiel wird die Vorgehensweise der Durchhang- und Winkligkeitskompensation an einem Gitter mit einer Größe von 4 x 5 (Zeilen x Spalten) näher erläutert. Die Größe des gesamten Gitters beträgt 2000 x 900 mm^2. Die Kompensationswerte werden jeweils in Schritten von 500 mm auf der X-Achse und in Schritten von 300 mm auf der Y-Achse ermittelt.

Hinweis

Für die maximale Abmessung des Gitters (Anzahl der Zeilen und Spalten) gelten folgende Abhängigkeiten:

- Die Zeilenzahl ist abhängig von der Anzahl der Achsen im System (abhängig vom NCU-Typ).
- Die Spaltenzahl ist abhängig von der maximalen Anzahl von Werten, die in einer Korrekturtabelle eingetragen werden können (bis max. 2000 Werte).

⚠ VORSICHT

Die Zeilen- und Spaltenanzahl wird über folgendes Maschinendatum eingestellt:

MD18342 $_MN_MM_CEC_MAX_POINTS (Maximale Anzahl der Stützpunkte bei Durchhangkompensation)

Dieses Maschinendatum ist speicherkonfigurierend!

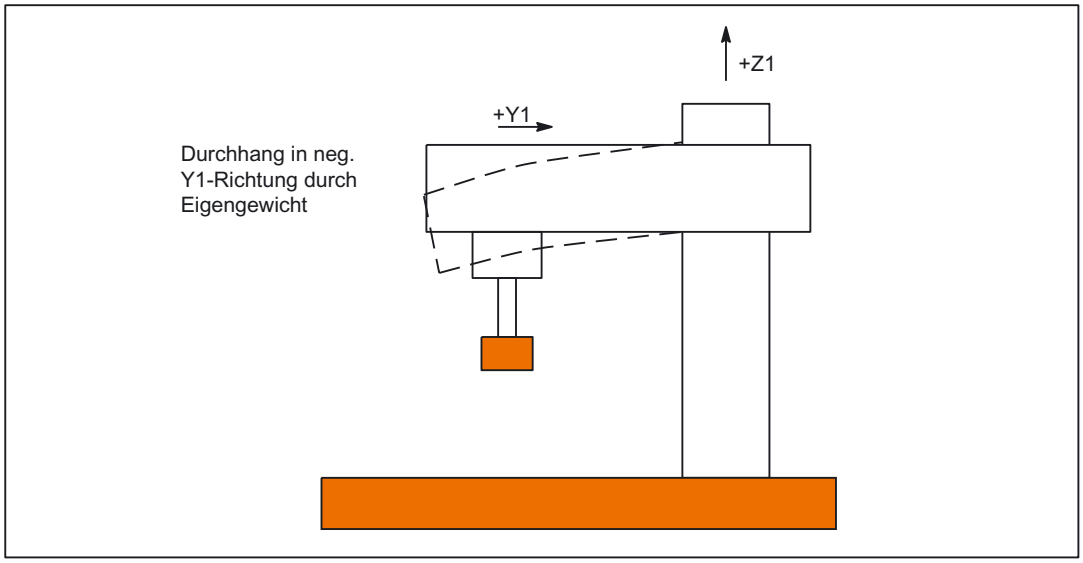


Bild 5-9 Kompensationswerte der Z-Achse bei schachbrettartiger Aufteilung der X-Y-Ebene

Das beschriebene Anwendungsbeispiel kann mit folgendem Teileprogrammcode realisiert werden:

```
$MA_CEC_ENABLE[Z1]          = FALSE      ; deaktivieren der Kompensation
                                         ; durch Setzen auf FALSE.
                                         ; Damit können die Tabellenwerte
                                         ; geändert werden, ohne dass
                                         ; der Alarm 17070 auftritt.
NEWCONF                                  ; $MA_CEC_ENABLE wirksam setzen
;Festlegen der Werte f_i(x) in den f-Tabellen:
;Funktionswerte f_1(x) für Tabelle mit Index [0]
$AN_CEC[0,0]                =0.1
$AN_CEC[0,1]                =0.2
$AN_CEC[0,2]                =0.3
$AN_CEC[0,3]                =0.4
$AN_CEC[0,4]                =0.5

;Funktionswerte f_2(x) für Tabelle mit Index [1]
$AN_CEC[1,0]                =0.6
$AN_CEC[1,1]                =0.7
$AN_CEC[1,2]                =0.8
$AN_CEC[1,3]                =0.9
$AN_CEC[1,4]                =1.0

;Funktionswerte f_3(x) für Tabelle mit Index [2]
$AN_CEC[2,0]                =1.1
$AN_CEC[2,1]                =1.2
$AN_CEC[2,2]                =1.3
$AN_CEC[2,3]                =1.4
$AN_CEC[2,4]                =1.5

;Funktionswerte f_4(x) für Tabelle mit Index [3]
$AN_CEC[3,0]                =1.6
$AN_CEC[3,1]                =1.7
$AN_CEC[3,2]                =1.8
$AN_CEC[3,3]                =1.9
$AN_CEC[3,4]                =2.0

;Auswertung der f-Tabellen mit den Kompensationswerten freigeben
$SN_CEC_TABLE_ENABLE[0]     =TRUE
$SN_CEC_TABLE_ENABLE[1]     =TRUE
$SN_CEC_TABLE_ENABLE[2]     =TRUE
$SN_CEC_TABLE_ENABLE[3]     =TRUE

;Gewichtsfaktor der f-Tabellen festlegen
$SN_CEC_TABLE_WEIGHT[0]     =1.0
$SN_CEC_TABLE_WEIGHT[1]     =1.0
$SN_CEC_TABLE_WEIGHT[2]     =1.0
```

```
$SN_CEC_TABLE_WEIGHT[3]      =1.0

;Änderungen der folgenden Tabellenparameter werden erst
;nach Power On wirksam
;Basisachse X1 festlegen
$AN_CEC_INPUT_AXIS[0]        =(X1)
$AN_CEC_INPUT_AXIS[1]        =(X1)
$AN_CEC_INPUT_AXIS[2]        =(X1)
$AN_CEC_INPUT_AXIS[3]        =(X1)

;Kompensationsachse Z1 festlegen
$AN_CEC_OUTPUT_AXIS[0]       =(Z1)
$AN_CEC_OUTPUT_AXIS[1]       =(Z1)
$AN_CEC_OUTPUT_AXIS[2]       =(Z1)
$AN_CEC_OUTPUT_AXIS[3]       =(Z1)

;Stützpunktabstand für die Kompensationswerte der f-Tabellen festlegen
$AN_CEC_STEP[0]              =500.0
$AN_CEC_STEP[1]              =500.0
$AN_CEC_STEP[2]              =500.0
$AN_CEC_STEP[3]              =500.0

;Kompensation beginnt bei X1=0
$AN_CEC_MIN[0]               =0.0
$AN_CEC_MIN[1]               =0.0
$AN_CEC_MIN[2]               =0.0
$AN_CEC_MIN[3]               =0.0

;Kompensation endet bei X1=2000
$AN_CEC_MAX[0]               =2000.0
$AN_CEC_MAX[1]               =2000.0
$AN_CEC_MAX[2]               =2000.0
$AN_CEC_MAX[3]               =2000.0
;Werte der f-Tabellen mit Index [t1] werden mit Werten der g-Tabellen
;mit der Nummer [t2] multipliziert
;entspricht der oben ausgeführten Rechenvorschrift
$AN_CEC_MULT_BY_TABLE[0]        =5
$AN_CEC_MULT_BY_TABLE[1]        =6
$AN_CEC_MULT_BY_TABLE[2]        =7
$AN_CEC_MULT_BY_TABLE[3]        =8

;Festlegen der g-Tabellenwerte für g_i(y) :
:Funktionswerte g_1(x) für Tabelle mit Index [4]
$AN_CEC[4,0]                    =1.0
$AN_CEC[4,1]                    =0.0
```

```
$AN_CEC[4,2]                    =0.0
$AN_CEC[4,3]                    =0.0

;Funktionswerte g_2(x) für Tabelle mit Index [5]
$AN_CEC[5,0]                    =0.0
$AN_CEC[5,1]                    =1.0
$AN_CEC[5,2]                    =0.0
$AN_CEC[5,3]                    =0.0

;Funktionswerte g_3(x) für Tabelle mit Index [6]
$AN_CEC[6,0]                    =0.0
$AN_CEC[6,1]                    =0.0
$AN_CEC[6,2]                    =1.0
$AN_CEC[6,3]                    =0.0

;Funktionswerte g_4(x) für Tabelle mit Index [7]
$AN_CEC[7,0]                    =0.0
$AN_CEC[7,1]                    =0.0
$AN_CEC[7,2]                    =0.0
$AN_CEC[7,3]                    =1.0

;Auswertung der g-Tabellen mit den Kompensationswerten freigeben
$SN_CEC_TABLE_ENABLE[4]         =TRUE
$SN_CEC_TABLE_ENABLE[5]         =TRUE
$SN_CEC_TABLE_ENABLE[6]         =TRUE
$SN_CEC_TABLE_ENABLE[7]         =TRUE

;Gewichtsfaktor für die g-Tabellen festlegen
$SN_CEC_TABLE_WEIGHT[4]         =1.0
$SN_CEC_TABLE_WEIGHT[5]         =1.0
$SN_CEC_TABLE_WEIGHT[6]         =1.0
$SN_CEC_TABLE_WEIGHT[7]         =1.0

;Änderungen der folgenden Tabellenparameter werden
;erst nach Power On wirksam
;Basisachse Y1 festlegen
$AN_CEC_INPUT_AXIS[4]                   =(Y1)
$AN_CEC_INPUT_AXIS[5]                   =(Y1)
$AN_CEC_INPUT_AXIS[6]                   =(Y1)
$AN_CEC_INPUT_AXIS[7]                   =(Y1)

;Kompensationsachse Z1 festlegen
$AN_CEC_OUTPUT_AXIS[4]                  =(Z1)
$AN_CEC_OUTPUT_AXIS[5]                  =(Z1)
$AN_CEC_OUTPUT_AXIS[6]                  =(Z1)
```

```
$AN_CEC_OUTPUT_AXIS[7]                  =(Z1)
;Stützpunktabstand für die Kompensationswerte der g-Tabellen festlegen
$AN_CEC_STEP[4]                         =300.0
$AN_CEC_STEP[5]                         =300.0
$AN_CEC_STEP[6]                         =300.0
$AN_CEC_STEP[7]                         =300.0

;Kompensation beginnt bei Y1=0
$AN_CEC_MIN[4]                          =0.0
$AN_CEC_MIN[5]                          =0.0
$AN_CEC_MIN[6]                          =0.0
$AN_CEC_MIN[7]                          =0.0

;Kompensation endet bei Y1=900
$AN_CEC_MAX[4]                          =900.0
$AN_CEC_MAX[5]                          =900.0
$AN_CEC_MAX[6]                          =900.0
$AN_CEC_MAX[7]                          =900.0
$MA_CEC_ENABLE[Z1]           =TRUE      ;Kompensation wieder aktivieren
NEWCONF

;Durchführung eines Programmtests, um die Wirksamkeit
;der Kompensation zu überprüfen
G01 F1000 X0 X0 Z0 G90
R1=0 R2=0
LOOP_Y:
LOOP_X:
STOPRE
X=R1 Y=R2
M0                          ; warten, um den CEC-Wert zu prüfen
R1=R1+500
IF R1 <=2000 GOTOB LOOP_X
R1=0
R2=R2+300
IF R2<=900 GOTOB LOOP_Y
```

Hinweis

Den Kompensationswert können Sie an der Bedienoberfläche unter der Variable "Kompensation Durchhang + Temperatur" ansehen. Wählen Sie dazu zuerst den Softkey "Diagnose" dann "Service Achse". Den aktuell wirksamen Kompensationswert können Sie neben der Variable "Kompensation Durchhang + Temperatur" ablesen.

```
;zur Vorbereitung der Tabellen-Konfiguration werden die Power On
```

```
;Maschinendaten gesetzt
;cec.md:
;Optionsdatum setzen zur Inbetriebnahme
;Festlegen der Anzahl von Stützpunkten in den Kompensationstabellen
;Maschinendatum ist speicherkonfigurierend
$MN_MM_CEC_MAX_POINTS[0]=5
$MN_MM_CEC_MAX_POINTS[1]=5
$MN_MM_CEC_MAX_POINTS[2]=5
$MN_MM_CEC_MAX_POINTS[3]=5
$MN_MM_CEC_MAX_POINTS[4]=4
$MN_MM_CEC_MAX_POINTS[5]=4
$MN_MM_CEC_MAX_POINTS[6]=4
$MN_MM_CEC_MAX_POINTS[7]=4
$MA_CEC_MAX_SUM[AX3]=10.0          ; Festlegen des max.
                                   ; Summenkompensationswerts
$MA_CEC_MAX_VELO[AX3]=100.0        ; Begrenzen der max. Änderungen des
                                   ; Summenkompensationswerts
M17
```

Erläuterung:

Die Kompensationswerte können nicht direkt als 2-dimensionalen Gitterfeld eingegeben werden. Es müssen Korrekturtabellen erstellt werden, in die die Kompensationswerte eingetragen werden.

Eine Korrekturtabelle beinhaltet die Kompensationswerte einer Zeile (im Beispiel vier Zeilen, d. h. vier Korrekturtabellen). In der ersten Tabelle des Beispiels werden die Kompensationswerte 0,1 bis 0,5 eingetragen und in der zweiten die Kompensationswerte von 0,6 bis 1,0 usw. Die Korrekturtabellen werden im Weiteren mit f-Tabellen und die Tabellenwerte als f_i(x) bezeichnet (i=Nummer der Tabelle).

Zur Auswertung der Kompensationswerte der f-Tabellen ist eine Multiplikation mit weiteren Tabellen nötig. Diese Tabellen werden im Folgenden als g-Tabellen und die Tabellenwerte als g_i(y) bezeichnet. Die Anzahl der f-Tabellen und g-Tabellen sind gleich (im Beispiel vier).

In den g-Tabellen wird jeweils ein Kompensationswert in jeder Tabelle auf 1 gesetzt und alle weiteren auf 0. Die Position des Kompensationswerts 1 innerhalb der Tabelle richtet sich nach der Tabellennummer. Bei der ersten g-Tabelle befindet sich der Kompensationswert 1 an der ersten Stützpunkt-Stelle und bei der zweiten g-Tabelle befindet sich der Kompensationswert 1 an der zweiten Stützpunkt-Stelle usw. Durch die Multiplikation der g-Tabellen mit den f-Tabellen wird jeweils der richtige Kompensationswert der f-Tabelle durch Multiplikation mit 1 ausgewählt. Die nicht relevanten Kompensationswerte werden durch die Multiplikation mit 0 ausgeblendet.

Der Kompensationswert D_z an der Stelle (x/y) wird dabei nach folgender Rechenvorschrift ermittelt:

$D_z(x/y)=f_1(x)*g_1(y) + f_2(x)*g_2(y) + ...$

Bei der Berechnung des Kompensationswerts für die aktuelle Position der Maschinenspindel werden nach dieser Rechenvorschrift die f-Tabellenwerte mit den g-Tabellenwerten multipliziert.

Übertragen auf das Beispiel bedeutet das z. B. für die Ermittlung des Kompensationswerts $D_z(500/300)$, dass jeweils die Funktionswerte f_i(500) der f-Tabellen mit den Funktionswerten g_i(300) der g-Tabellen multipliziert werden:

$D_z(500/300)$ = f_1(1000)*g_1(300) + f_2(1000)*g_2(300) + f_3(1000)*g_3(300) + f_4(1000)*g_4(300)

$D_z(500/300)$ = 0,2*0 + 0,7*1 + 1,2*0 + 1,7*0 = 0,7

5.4.4 Richtungsabhängige Spindelsteigungsfehler-Kompensation

5.4.4.1 Funktionsbeschreibung

Bei zu großen richtungsabhängigen Differenzen in den Kompensationspunkten, bei einer nicht konstanten Lose bzw. bei sehr hohen Anforderungen an die Genauigkeit kann eine richtungsabhängige Kompensation der Spindelsteigungsfehler oder Messsystemfehler (bei direkter Positionserfassung) erforderlich sein.

"Richtungsabhängige Spindelsteigungsfehler-Kompensation" ("Richtungsabhängige SSFK" oder auch "Bidirektionale SSFK")

Bei der "Richtungsabhängigen SSFK" werden je Achse zwei Kompensationstabellen verwendet: eine Tabelle für die Plus-Verfahrrichtung und eine für die Minus-Verfahrrichtung. In die Tabelle wird direkt die fehlerhafte Abweichung im jeweiligen Kompensationspunkt eingetragen (Fehler = Korrekturwert). Die Steuerung kompensiert automatisch mit dem entgegengesetzten Vorzeichen. Sie führt in jedem Fall automatisch eine lineare Interpolation zwischen den benachbarten Stützpunkten aus, so dass es nicht zu sprunghaften Positionsänderungen durch die Kompensation kommen kann.

Voraussetzungen / Wirksamkeit

Die "Richtungsabhängigen SSFK" wird in der SINUMERIK-Steuerung als Sonderfall der "Durchhangkompensation" realisiert. Es gelten daher die Voraussetzungen und Bedingungen der "Durchhangkompensation" (siehe "Kompensation von Durchhang und Winkligkeitsfehler (Seite 364)").

Die Wirksamkeit der Kompensation kann durch Referenzmessung z. B. mit dem Laserinterferometer oder im einfachsten Fall über die Service-Anzeige der jeweiligen Achse überprüft werden.

ACHTUNG
Wird die "Richtungsabhängige SSFK" parallel zur Durchhangkompensation und Korrektur der Winkligkeit verwendet, müssen die Randbedingungen dieser Funktionen zusammen betrachtet werden, z. B. die Zuordnung der Tabellen <t> zur jeweiligen Funktion.

5.4.4.2 Inbetriebnahme

Messung der Fehler- bzw. Korrekturwerte

In Anlehnung an die Vorgehensweise bei der richtungs**un**abhängigen SSFK (siehe "Kompensation von Spindelsteigungsfehler und Messsystemfehler (Seite 358)") werden bei der richtungsabhängigen SSFK mit einem geeigneten Messgerät (z. B. Laserinterferometer) **richtungsabhängige Fehlerkurven je Achse** ermittelt. Dazu muss ein entsprechendes NC-Programm mit Messpunkten und Wartezeiten erstellt werden (siehe Programm "BI_SSFK_MESS_AX1_X.MPF" im "Beispiel (Seite 386)"). Die Software der verschiedenen Messgeräte bietet hier unterschiedliche Unterstützung bei der praktischen Umsetzung mit der SINUMERIK-Steuerung, deshalb soll dieser Vorgang hier nur allgemein und steuerungsbezogen beschrieben werden.

Aus den richtungsabhängigen Fehlerkurven werden die Steigungsfehler (= Sollposition - Istposition) den frei gewählten Stützpunkten (Messpunkten) zugeordnet.

Die positive Richtung ergibt sich mit steigenden Positionswerten.

Die Messung ist bei Erstinbetriebnahme erst durchzuführen, wenn die Verfahrrichtung der Achsen dem Koordinatensystem der Maschine bereits angepasst ist.

Inbetriebnahme der richtungsabhängigen SSFK

1. Option "Durchhangkompensation, mehrdimensional" freigeben.

   Hinweis

   Lizenzierungsverfahren beachten!

2. Anzahl der Kompensations-Stützpunkte festlegen (siehe auch "Kompensation von Durchhang und Winkligkeitsfehler: Inbetriebnahme (Seite 368)")

   Jeder Achse sind zwei Kompensationstabellen für positive **und** negative Verfahrrichtung zuzuordnen und mit dem folgenden Maschinendatum die Anzahl der Kompensations-Stützpunkte festzulegen:

   MD18342 $MN_MM_CEC_MAX_POINTS[<t>] (maximale Anzahl der Stützpunkte bei Durchhangkompensation)

   mit: <t> = Index der Kompensationstabelle

   Zulässiger Bereich: 0 ≤ t < 2 * maximale Achsanzahl

   Beispiel:

   MD18342 [0] = 11; 11 Stützpunkte für 1. Tabelle, z. B. positive Verfahrrichtung, X-Achse

   MD18342 [1] = 11; 11 Stützpunkte für 2. Tabelle, z. B. negative Verfahrrichtung, X-Achse

   MD18342 [2] = 21; 21 Stützpunkte für die 3. Tabelle, z. B. positive Verfahrrichtung, Y-Achse

   MD18342 [3] = 21; 21 Stützpunkte für die 4. Tabelle, z. B. positive Verfahrrichtung, Y-Achse

   …

   MD18342 [61] = ...; Anzahl Stützpunkte für die 62. Tabelle

   ACHTUNG

   Bei einer Änderung von MD18342 wird der ALARM 4400 gemeldet:

   "Reorganisation des gepufferten Speichers!"

   Damit eine automatische Speicherkonfiguration unter Beibehaltung aller bisher eingetragenen Daten möglich wird, darf **kein** Systemhochlauf (POWER ON) ohne eine vorherige Serieninbetriebnahme ausgeführt werden.

3. Serieninbetriebnahme durchführen:
   - NC-Archiv mit den Eintragungen in MD18342 [<t>] erstellen.
   - Archiv wieder einlesen.

     → Der NC-Speicher wird automatisch konfiguriert.

   Die Speicherplätze für die Tabellen stehen nun zur Beschreibung mit den Kompensationswerten zur Verfügung.

4. Tabellen mit Kompensationswerten für die jeweiligen Achsen und Verfahrrichtungen als NC-Programme erstellen (siehe Abschnitt "Messung der Fehler- bzw. Korrekturwerte").

5. Programm mit Kompensationswerten in der Steuerung ausführen.

   Betriebsart AUTOMATIK > Anwahl Programm > NC-Start

   Hinweis

   Vor jedem Einlesen der Kompensationstabellen sind folgende Parameter immer auf **0** und anschließend zur Aktivierung immer auf **1** zu setzen:

   MD32710 $MA_CEC_ENABLE[<AXi>] (Freigabe der Durchhangkompensation) = **0** → **1**

   SD41300 $SN_CEC_TABLE _ENABLE[<t>] (Freigabe der Kompensationstabelle) = **0** → **1**

   Die Umkehrlose ist immer auf **0** zu setzen:

   MD32450 $MA_BACKLASH [<e>] (Umkehrlose) = **0**

   mit: <e> = Lagemesssystem

   Durch Nutzung der Programm-Vorlage "BI_SSFK_TAB_AX1_X.MPF" (siehe "Beispiel (Seite 386)") werden diese Vorgänge automatisiert. Bei Handeingabe von Maschinendaten ist das allgemein übliche "MD wirksam setzen" bzw. "Reset" zu beachten.

6. POWER ON (Warmstart) ausführen.
7. Es können nun Vergleichsmessungen mit dem Laserinterferometer durchgeführt werden.
8. Zur weiteren Verbesserung des Kompensationsergebnisses ist es denkbar, einzelne Kompensationswerte im Programm zu korrigieren. Nach wiederholtem Einlesen der Tabelle ist kein POWER ON mehr notwendig.

Hinweis

Ablauf bei SINUMERIK 828D

Bei SINUMERIK 828D entfallen die Schritte **2** und **3**, da sofort mit Freigabe der Option "Durchhangkompensation, mehrdimensional" 8 Tabellen mit je 128 Stützpunkten pro Tabelle für die Kompensation zur Verfügung stehen. Eine Erweiterung ist nicht möglich!

Hinweis

Wie in Schritt **5** beschrieben, wird die Kompensationstabelle als ausführbares Programm in den Programmspeicher geladen und durch NC-Start in die vorher konfigurierten Speicherbereiche der Steuerung übertragen. Dieser Vorgang kann aus Gründen der Übersichtlichkeit für jede Tabelle wiederholt werden. Es ist aber auch möglich, alle Tabellen in einem Initialisierungsschritt zu laden. Wirksam werden die Korrekturwerte nach MD32710[<AXi>] = 1 und zwingendem POWER ON.

Hinweis

NC_CEC.INI

Die über "Inbetriebnahme" > "Systemdaten" (aus dem Ordner "NC-Aktive-Daten" > "Durchhang-Winkligkeit Komp") kopierte Datei "NC_CEC.INI" beinhaltet alle vereinbarten Durchhang-/Winkligkeits- und richtungsabhängigen SSFK-Tabellen.

Tabellenparameter

In der Kompensationstabelle sind für die jeweilige Richtung die positionsbezogenen Korrekturen sowie zusätzliche Tabellenparameter in Form von Systemvariablen zu hinterlegen:

- $AN_CEC[<t>,<N>] (Korrekturwert für Stützpunkt <N> der Kompensationstabelle[<t>])
- $AN_CEC_INPUT_AXIS[<t>] (Basisachse)
- $AN_CEC_OUTPUT_AXIS[<t>] (Kompensationsachse)

  Hinweis

  Basis- und Kompensationsachse sind bei der "Richtungsabhängigen SSFK" **identisch**.

- $AN_CEC_STEP[<t>] (Stützpunktabstand)
- $AN_CEC_MIN[<t>] (Anfangsposition)
- $AN_CEC_MAX[<t>] (Endposition)
- $AN_CEC_DIRECTION[<t>] (Richtungsabhängige Kompensation)

  Mit dieser Systemvariablen wird eingestellt, ob die Kompensationstabelle[<t>] nur bei positiver bzw. negativer Verfahrrichtung der Basisachse wirkt:

  – $AN_CEC_DIRECTION[<t>] = 1:

    Tabelle wirkt nur in positiver Verfahrrichtungen der Basisachse

  – $AN_CEC_DIRECTION[<t>] = -1:

    Tabelle wirkt nur in negativer Verfahrrichtungen der Basisachse

    Hinweis

    Die Einstellung $AN_CEC_DIRECTION[<t>] = 0 (Tabelle wirkt für beide Verfahrrichtungen der Basisachse) ist bei der "Richtungsabhängigen SSFK" **nicht** relevant.

- $AN_CEC_IS_MODULO[<t>] (Kompensation mit Modulofunktion)

Hinweis

Für die ausführliche Beschreibung dieser Systemvariablen siehe "Kompensation von Durchhang und Winkligkeitsfehler: Inbetriebnahme (Seite 368)".

Maßsystem

Siehe "Kompensation von Durchhang und Winkligkeitsfehler: Inbetriebnahme (Seite 368)".

Überwachung

Siehe "Kompensation von Durchhang und Winkligkeitsfehler: Inbetriebnahme (Seite 368)".

5.4.4.3 Beispiel

Im folgenden Beispiel für eine 3-Achs-Maschine sind die richtungsabhängigen Kompensationstabellen der X-Achse ausführlich dargestellt.

Projektierung

Anzahl der Kompensations-Stützpunkte:

MD18342 $MN_MM_CEC_MAX_POINTS[**0**] = 11 (Tabelle **1**: Achse **X**, **positive** Verfahrrichtung)

MD18342 $MN_MM_CEC_MAX_POINTS[**1**] = 11 (Tabelle **2**: Achse **X**, **negative** Verfahrrichtung)

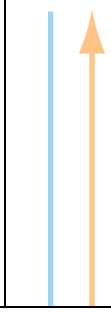
Hinweis

Die Festlegung der Stützpunktanzahl ist bei SINUMERIK 828D nicht erforderlich, da sofort mit Freigabe der Option "Durchhangkompensation, mehrdimensional" 8 Tabellen mit je 128 Stützpunkten pro Tabelle für die Kompensation zur Verfügung stehen. Eine Erweiterung ist nicht möglich!

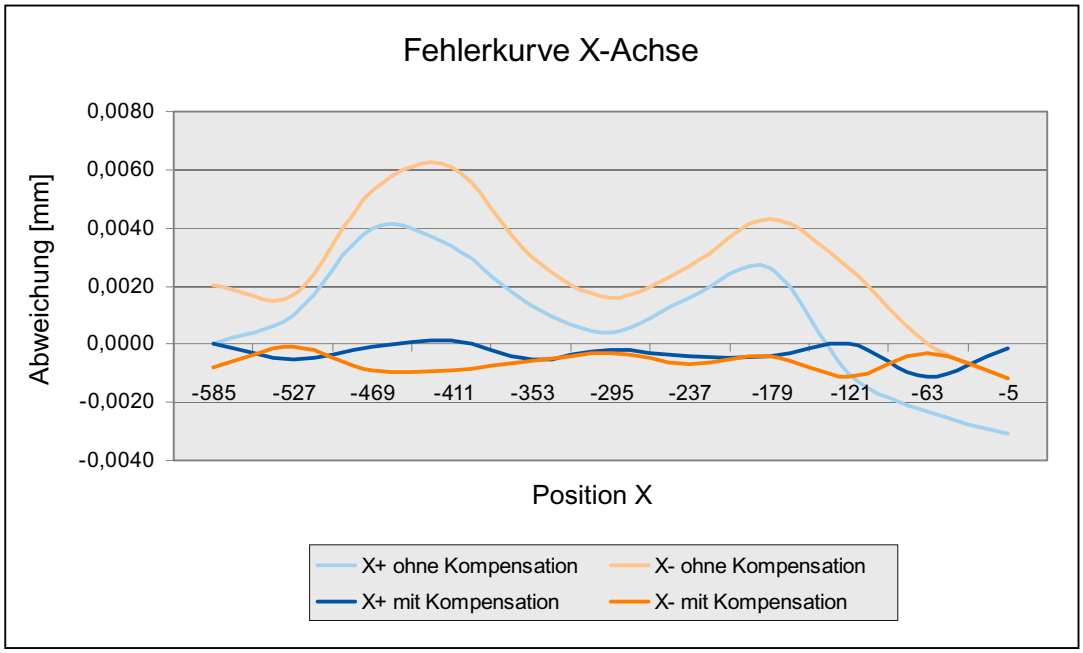
Stützpunkte

Tabelle <t>	[0,<N>]										
Anzahl Stützpunkte	MD18342 $MN_MM_CEC_MAX_POINTS[0] = **11**										
Stützpunkt <N>	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Position X	-585	-527	-469	-411	-353	-295	-237	-179	-121	-63	-5

Messung

			Sollpostion	Abweichung		Kontrollmessung	
	Position	Korr.-Nr.	Messposition [mm]	Richtung + [mm]	Richtung - [mm]	Richtung + [mm]	Richtung - [mm]
$AC_CEC_MIN[<t>]	-585	0	-585	0,0000	0,0020	0,0000	-0,0008
		1	-527	0,0010	0,0017	-0,0005	-0,0001
		2	-469	0,0040	0,0053	-0,0001	-0,0009
		3	-411	0,0034	0,0061	0,0001	-0,0009
		4	-353	0,0013	0,0030	-0,0005	-0,0006
		5	-295	0,0004	0,0016	-0,0002	-0,0003
		6	-237	0,0016	0,0027	-0,0004	-0,0007
		7	-179	0,0026	0,0043	-0,0004	-0,0004
		8	-121	-0,0010	0,0026	0,0000	-0,0011

		9	-63		-0,0023		0,0000		-0,0011	-0,0003
$AC_CEC_MAX[<t>]	-5	10	-5		-0,0031		-0,0012		-0,0001	-0,0012



Programmierung

Das folgende Programm "BI_SSFK_TAB_AX1_X.MPF" enthält die Wertzuweisungen für die Parameter der beiden Kompensationstabellen (positive und negative Verfahrrichtung) der X-Achse:

```
;Richtungsabhängige SSFK
;1.Achse MX1
;Tabelle 1- positive Verfahrrichtung
;Tabelle 2 - negative Verfahrrichtung
;------------------------------------------------------------------------------
```

```
CHANDATA(1)
$MA_CEC_ENABLE[AX1]=0                  ;Kompensation AUS
$SN_CEC_TABLE_ENABLE[0]=0              ;Tabelle 1 sperren
$SN_CEC_TABLE_ENABLE[1]=0              ;Tabelle 2 sperren
NEWCONF
;-------------------------------------------------------------------------------------
$AN_CEC[0,0]=0                         ;1.Korrekturwert (Stützpunkt 0)
$AN_CEC[0,1]=0.001                     ;2.Korrekturwert (Stützpunkt 1)
$AN_CEC[0,2]=0.004                     ;3.Korrekturwert (Stützpunkt 2)
$AN_CEC[0,3]=0.0034                    ;4.Korrekturwert (Stützpunkt 3)
$AN_CEC[0,4]=0.0013                    ;5.Korrekturwert (Stützpunkt 4)
$AN_CEC[0,5]=0.0004                    ;6.Korrekturwert (Stützpunkt 5)
$AN_CEC[0,6]=0.0016                    ;7.Korrekturwert (Stützpunkt 6)
$AN_CEC[0,7]=0.0026                    ;8.Korrekturwert (Stützpunkt 7)
$AN_CEC[0,8]=-0.001                    ;9.Korrekturwert (Stützpunkt 8)
$AN_CEC[0,9]=-0.0023                   ;10.Korrekturwert (Stützpunkt 9)
$AN_CEC[0,10]=-0.0031                  ;letzter Korrekturwert (Stützpunkt 10)
$AN_CEC_INPUT_AXIS[0]=(AX1)            ;Basisachse
$AN_CEC_OUTPUT_AXIS[0]=(AX1)           ;Kompensationsachse
$AN_CEC_STEP[0]=58.0                   ;Stützpunktabstand
$AN_CEC_MIN[0]=-585.0                  ;Kompensation beginnt
$AN_CEC_MAX[0]=-5.0                    ;Kompensation endet
$AN_CEC_DIRECTION[0]=1                 ;Tabelle wirkt für positive Verfahrrichtungen
$AN_CEC_MULT_BY_TABLE[0]=0             ;keine Multiplikation (hier nicht relevant)
$AN_CEC_IS_MODULO[0]=0                 ;Kompensation ohne Modulofunktion
;-------------------------------------------------------------------------------------
$AN_CEC[1,0]=0.002                     ;(Stützpunkt 0)
$AN_CEC[1,1]=0.0017                    ;(Stützpunkt 1)
$AN_CEC[1,2]=0.0053                    ;(Stützpunkt 2)
$AN_CEC[1,3]=0.0061                    ;(Stützpunkt 3)
$AN_CEC[1,4]=0.003                     ;(Stützpunkt 4)
$AN_CEC[1,5]=0.0016                    ;(Stützpunkt 5)
$AN_CEC[1,6]=0.0027                    ;(Stützpunkt 6)
$AN_CEC[1,7]=0.0043                    ;(Stützpunkt 7)
$AN_CEC[1,8]=0.0026                    ;(Stützpunkt 8)
$AN_CEC[1,9]=0.000                     ;(Stützpunkt 9)
$AN_CEC[1,10]=-0.0012                  ;(Stützpunkt 10)
$AN_CEC_INPUT_AXIS[1]=(AX1)            ;Basisachse
$AN_CEC_OUTPUT_AXIS[1]=(AX1)           ;Kompensationsachse
$AN_CEC_STEP[1]=58.                    ;Stützpunktabstand
$AN_CEC_MIN[1]=-585.0                  ;Kompensation beginnt
$AN_CEC_MAX[1]=-5.0                    ;Kompensation endet
$AN_CEC_DIRECTION[1]=-1                ;Tabelle wirkt für negative Verfahrrichtungen
$AN_CEC_MULT_BY_TABLE[1]=0             ;keine Multiplikation (hier nicht relevant)
$AN_CEC_IS_MODULO[1]=0                 ;Kompensation ohne Modulofunktion (nur bei Rundachsen)
```

```
;-------------------------------------------------------------------------------
$MA_CEC_ENABLE[AX1]=1                  ;Kompensation EIN
$SN_CEC_TABLE_ENABLE[0]=1              ;Tabelle 1 freigeben
$SN_CEC_TABLE_ENABLE[1]=1              ;Tabelle 2 freigeben
NEWCONF
M17
```

Es können weitere Tabellen angelegt werden, z. B. für die Achsen Y und Z:

MD18342 $MN_MM_CEC_MAX_POINTS[**2**] = 90 (Tabelle **3**: Achse **Y**, positive Verfahrrichtung)

MD18342 $MN_MM_CEC_MAX_POINTS[**3**] = 90 (Tabelle **4**: Achse **Y**, negative Verfahrrichtung)

MD18342 $MN_MM_CEC_MAX_POINTS[**4**] = 50 (Tabelle **5**: Achse **Z**, positive Verfahrrichtung)

MD18342 $MN_MM_CEC_MAX_POINTS[**5**] = 50 (Tabelle **6**: Achse **Z**, negative Verfahrrichtung)

5.4.5 Erweiterung der Durchhangkompensation mit NCU-Link - nur 840D sl

Anwendung

Wird eine Anlage mit NCU-Link betrieben, können beliebige Achsen des NCU-Link-Verbundes miteinander kompensiert werden. Die beiden Achsen, die über Durchhangkompensation gekoppelt werden, müssen beide auf einer NCU interpoliert werden.

Siehe auch

- Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme (B3); Kapitel: NCU-Link
- Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme (B3); Kapitel: Achscontainer

Funktion

Die Parametrierung der Funktion Durchhangkompensation erfolgt über das Setzten von Systemvariablen der Form: $AN_CEC ...

Diese Systemvariablen werden in der Regel über ein Teileprogramm gesetzt, das der NCK in einem bestimmten Kanal abarbeitet. In der Variablen $AN_CEC_OUTPUT_AXIS oder $AN_CEC_INPUT_AXIS kann der Kanal-Achsbezeichner verwendet werden. Damit kann jede Achse dieses Kanals angesprochen werden, auch wenn sie auf einer anderen NCU liegt.

Ein Programm in Kanal-2 kann mit folgender Einstellung die Achse ZZ an die Achse XX koppeln (Die Einstellung gilt für die Tabelle mit der Nummer 0):

$AN_CEC_INPUT_AXIS[0] = (XX)

$AN_CEC_OUTPUT_AXIS[0] = (ZZ)

Damit wird AX3 auf NCU-1 mit AX2 auf NCU-2 "gekoppelt" (siehe Konfiguration 1).

Sind die zu koppelnden Achsen auf zwei unterschiedlichen Kanälen, gibt es folgende Varianten für die Parametrierung:

- **Variante 1: "Programmieren mit Kanal-Achsbezeichner"**:

  Es werden 2 unterschiedliche Teileprogramme TP1 und TP2 erstellt, die in unterschiedlichen Kanälen abgearbeitet werden.

  Achse "ZZ" wird an "XR" gekoppelt:

  Ausschnitt aus dem Teileprogramm TP1 im Kanal-1:

  ```
  $AN_CEC_INPUT_AXIS[0] = (XR)
  ```

  Ausschnitt aus dem Teileprogramm TP2 im Kanal-2:

  ```
  $AN_CEC_OUTPUT_AXIS[0] = (ZZ)
  ```

  Nachdem TP1 im Kanal 1 und TP2 im Kanal 2 ausgeführt worden sind, wird nach dem Neustart die Achse AX2 auf der NCU-2 mit der Achse AX1 auf der NCU-1 "gekoppelt".

- **Variante 2: "Programmieren mit Maschinen-Achsbezeichner"**:

  Es wird ein Teileprogramm erstellt, das in einem beliebigen Kanal der NCU-1 abläuft und die Maschinen-Achsnamen zusammen mit der NCU-Nummer angibt.

  Achse "ZZ" wird an "XR" gekoppelt:

  ```
  $AN_CEC_INPUT_NCU[0]=1
  $AN_CEC_INPUT_AXIS[0] = (AX1)
  $AN_CEC_OUTPUT_NCU[0]=2
  $AN_CEC_OUTPUT_AXIS[0] = (AX2)
  ```

  Der NCK überwacht, ob die Achsen auf der lokalen NCU wirklich interpoliert werden, d.h. gibt es einen Kanal, der diese Achse programmieren kann. Die "lokale NCU" ist immer die NCU, auf der das Teileprogramm abläuft.

  Folgende Achsen sind für die NCU-1 als Eingangs- oder Ausgangsachsen in Konfiguration 1 erlaubt: NC1_AX1, NC1_AX3, NC1_AX4, NC1_AX5, NC2_AX2, NC2_AX6

  Die Datensicherung aus dem NCK liefert die Kompensationsdaten immer mit der Sichtweise "Maschinen-Achsbezeichner".

ACHTUNG
Die NCU-Nummer ist vor dem Achsbezeichner zu programmieren. Eine Durchhangkompensation zwischen NC1_AX1 und NC1_AX2 ist nicht möglich.

Zuordnung der Achsen

Die Zuordnung von Eingangs- und Ausgangsachse erfolgt durch die Systemvariablen:

$AN_CEC_INPUT_NCU	und	$AN_CEC_INPUT_AXIS
$AN_CEC_OUTPUT_NCU	und	$AN_CEC_OUTPUT_AXIS

Die Systemvariablen werden erst nach einem Neustart wirksam.

Die Datensicherung erfolgt immer mit Maschinen-Achsbezeichnern.

Hinweis

Die Durchhangkompensation kann nur die Achsen auf einer NCU koppeln, die auch von dieser NCU entweder über das Teileprogramm oder eine Synchronaktion verfahrbar sind.

Diese Variablen werden optional gesetzt, wenn die Achsen (Eingang oder Ausgang) nicht auf der lokalen NCU vorhanden sind. Verwendet man bei der Programmierung von $AN_CEC_INPUT_AXIS und $AN_CEC_OUTPUT_AXIS einen Kanal-Achsbezeichner, so sind die Systemvariablen $AN_CEC_INPUT_NCU und $AN_CEC_OUTPUT_NCU ohne Bedeutung.

Die Steuerung prüft, ob beide Achsen von dieser NCU aus interpoliert werden können, d.h. ein Programm auf dieser NCU die Achse verfahren kann. Die Achsen können unterschiedlichen Kanälen zugeordnet sein. Zwei Achsen, die zu unterschiedlichen NCUs gehören, können auch kompensiert werden. Die Steuerung lehnt dies andernfalls mit Alarm 17040 ab.

Beide Achsen der Kompensation müssen auf einer NCU interpoliert werden, d.h. es kann ein oder es können zwei Teileprogramme existieren, die die Eingangs- und Ausgangsachse auf einer NCU verfahren.

Achscontainer

Der Achscontainer ist ein Verbund gleichartiger Achsen. Einer Kanalachse kann eine Achse aus dem Verbund zugeordnet werden. Diese Zuordnung ist variabel, so dass die Achse im Kanal nach und nach immer eine neue Achse aus dem Verbund zugewiesen bekommt. Damit kann das Teileprogramm mit einer Achse programmiert werden und nach und nach verschiedene Achsen verfahren.

Beispiel:

4 Spindeln sind auf einer Trommel angeordnet. Jede Spindel trägt ein Werkstück der Drehmaschine und in jedem Takt dreht sich die Trommel um 90 Grad. Damit werden die Werkstücke von einer Bearbeitungsstation zur nächsten transportiert. Der Kanal der Bearbeitungsstation muss nur eine Spindel programmieren, obwohl immer eine neue Spindel eingewechselt wird. Es handelt sich um eine Achscontainer-Rotation.

Die Durchhangkompensation kann mit dem Achscontainer kombiniert werden, falls er in Grundstellung ist, d.h. $AN_AXCTAS == 0. Anderenfalls wird die Programmierung mit Alarm 17040 abgelehnt.

"YY" soll an "XX" gekoppelt werden (siehe Konfiguration 2):

1. Programmierung mit Kanal-Achsbezeichner

```
$AN_CEC_INPUT_AXIS[0] = (XX)
$AN_CEC_OUTPUT_AXIS[0] = (YY)
```

2. Programmierung mit Maschinen-Achsbezeichner

```
$AN_CEC_INPUT_NCU[0]=1          ; optional ...
$AN_CEC_INPUT_AXIS[0] = (AX2)
$AN_CEC_OUTPUT_NCU[0]=2
$AN_CEC_OUTPUT_AXIS[0] = (AX2
```

Damit wird die Achse AX2 der NCU-1 an die Achse-2 der NCU-2 gekoppelt.

ACHTUNG
Mit jeder Container-Rotation bleibt YY an XX gekoppelt, allerdings ist jetzt eine andere Achse hinter YY: In Konfiguration 3 ist YY "AX5 der NCU-1".
Damit sind nach der Rotation andere reale Achsen gekoppelt: In diesem Beispiel ist AX-5 der NCU-1 mit AX-2 der NCU-1 gekoppelt.

Grundsätzlich gilt:

Die Kopplung wird zwischen zwei Achsen aus der LAI-Schicht aufgebaut, damit sind nach jeder Achscontainer-Drehung andere Achsen an der Kopplung beteiligt. Um eine Kopplung genau zwischen zwei realen Achsen vorzunehmen, muss bei jeder Container-Drehung eine neue Tabelle aktiviert werden.

Konfigurationsbeispiel

Die nachfolgenden Abbildungen (Konfiguration 1, Konfiguration 2 und Konfiguration 3) zeigen die Achskonfigurationen eines NCU-Links, der aus zwei NCUs zusammengebaut worden ist.

In Konfiguration 1 werden die beiden Kanäle CHAN-1 und CHAN-2 der NCU-1 dargestellt. Dabei sind die Kanalachsnamen, die über das Maschinendatum $MC_AXCONF_CHANAX_NAME_TAB definiert werden, jeweils eingetragen. Die Kanalkonfiguration der zweiten NCU ist nicht dargestellt.

Alle von dieser NCU interpolierbaren Achsen werden im "Logischen NCK Maschinenachsabbild" (LAI-Schicht) zusammengefasst. Die Zuordnung zwischen Kanal und MKS-Achsschicht erfolgt durch $MC_AXCONF_MACHAX_USED.

Die Zuordnung zwischen dem "Logischen NCK Maschinenachsabbild" und den realen Achsen erfolgt über das Maschinendatum $MN_AXCONF_LOGICMACHAX_TAB. Verfolgt man die Verbindungslinie, die bei der Kanalachse ZZ startet, so endet man auf NCU-2 bei der Achse AX-2, d.h. um die 2. Achse der NCU-2 zu verfahren, ist im 2. Kanal der NCU-1 folgende Anweisung zu programmieren: `"N2040 POS[ZZ]=10 FA[ZZ]=1000"`

Konfiguration 2 und Konfiguration 3 erweitern die Abbildung von Konfiguration 1 um einen Achscontainer (CT1), der mit dem Maschinendatum $MN_AXCT_AXCONF_ASSIGN_TAB1 eingestellt wird. Der Achscontainer ist ein übergreifendes Objekt, d.h. jeder Achscontainer existiert nur einmal für das ganze NCU Cluster.

Für die NCU-1 sind die Kanalachsen YR und YY die Teilnehmer am Achscontainer, die beiden Kanalachsen aus der NCU-2 sind nicht dargestellt. Der Container enthält die realen Achsen NC1_AX5, NC1_AX6, NC2_AX1 und NC2_AX2. Während des Hochlaufs verbindet der Container YR mit NC2_AX1 und YY mit NC2_AX2. In Konfiguration 3 ist der Container rotiert, d.h. die Verbindungsstruktur hat sich geändert. Jetzt ist YR mit NC2_AX2 und YY mit NC1_AX5 verbunden.

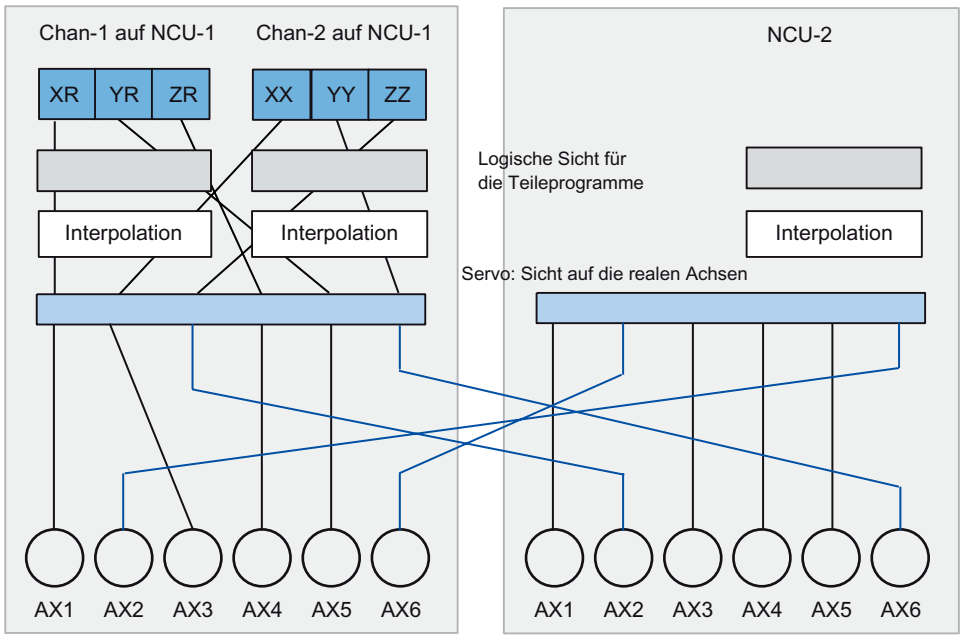


Bild 5-10 Konfiguration 1: NCU-Link vom Kanal zur realen Achse

Maschinendaten zu Konfiguration 1

```
; ########## NCU1 ##########
$MN_NCU_LINKNO = 1
$MN_MM_NCU_LINK_MASK = 1
$MN_MM_LINK_NUM_OF_MODULES= 2
$MN_MM_SERVO_FIFO_SIZE = 3

$MN_ASSIGN_CHAN_TO_MODE_GROUP[1]=1

$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "NC1_AX1"
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "NC1_AX3"
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC2_AX2"
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = "NC1_AX4"
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = "NC1_AX5"
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = "NC2_AX6"

CHANDATA(1)
$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=5
$MC_AXCONF_MACHAX_USED[2]=4
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0
$MC_AXCONF_CHANAX_NAME_TAB[0] = "XR"
```

```
$MC_AXCONF_CHANAX_NAME_TAB[1] = "YR"
$MC_AXCONF_CHANAX_NAME_TAB[2] = "ZR"

CHANDATA(2)
$MC_REFP_NC_START_LOCK=0
$MC_AXCONF_MACHAX_USED[0]=2
$MC_AXCONF_MACHAX_USED[1]=6
$MC_AXCONF_MACHAX_USED[2]=3
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0
$MC_AXCONF_CHANAX_NAME_TAB[0] = "XX"
$MC_AXCONF_CHANAX_NAME_TAB[1] = "YY"
$MC_AXCONF_CHANAX_NAME_TAB[2] = "ZZ"
M30

; ########## NCU-2 ##########
$MN_NCU_LINKNO = 2
$MN_MM_NCU_LINK_MASK = 1
$MN_MM_LINK_NUM_OF_MODULES= 2
$MN_MM_SERVO_FIFO_SIZE = 3
;
;
;
$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "NC2_AX1"
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "NC1_AX6"
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC2_AX3"
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = "NC2_AX4"
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = "NC2_AX5"
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = "NC1_AX2"

CHANDATA(1)
$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=2
$MC_AXCONF_MACHAX_USED[2]=3
$MC_AXCONF_MACHAX_USED[3]=4
$MC_AXCONF_MACHAX_USED[4]=5
$MC_AXCONF_MACHAX_USED[5]=6
$MC_AXCONF_MACHAX_USED[6]=0
M30
```

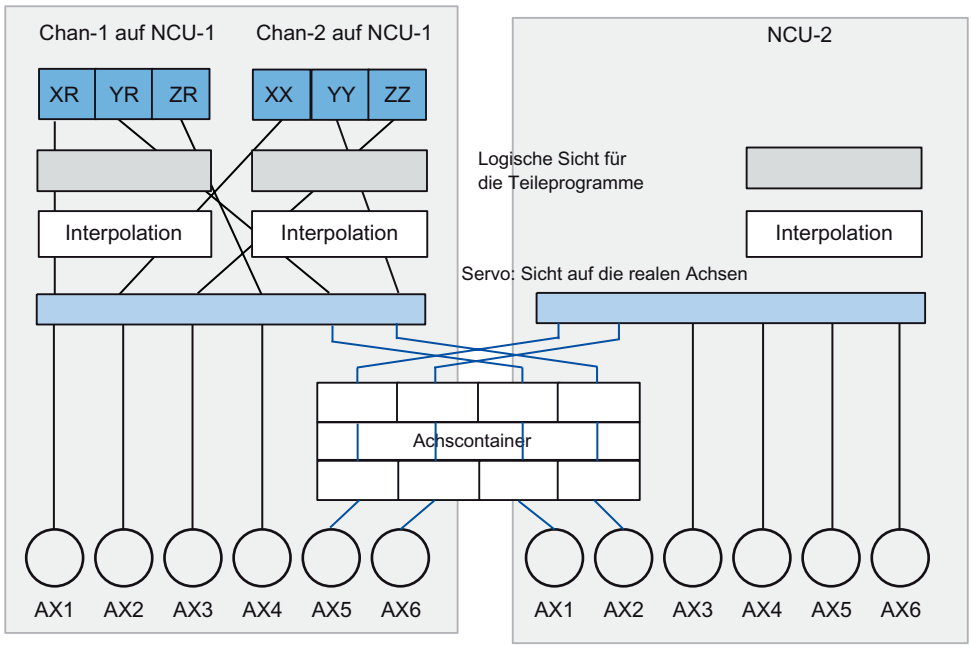


Bild 5-11 Konfiguration 2: NCU-Link mit Achscontainer im Ausgangszustand

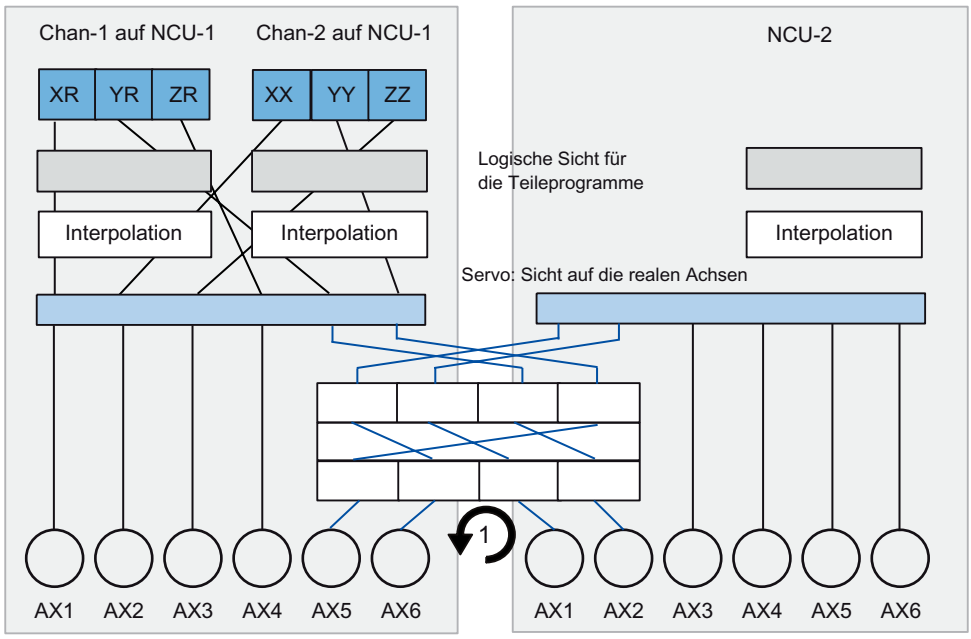


Bild 5-12 Konfiguration 3: NCU-Link mit Achscontainer im rotierten Zustand

Maschinendaten zu Konfiguration 2

```
; ########## NCU1 ##########
$MN_NCU_LINKNO = 1
$MN_MM_NCU_LINK_MASK = 1
$MN_MM_LINK_NUM_OF_MODULES= 2
$MN_MM_SERVO_FIFO_SIZE = 3

$MN_ASSIGN_CHAN_TO_MODE_GROUP[1]=1

$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "NC1_AX1"
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "NC1_AX3"
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC2_AX2"
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = "NC1_AX4"
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = "CT1_SL3"
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = "CT1_SL4"

$MN_AXCT_AXCONF_ASSIGN_TAB1[0] = "NC1_AX5"
$MN_AXCT_AXCONF_ASSIGN_TAB1[1] = "NC1_AX6"
$MN_AXCT_AXCONF_ASSIGN_TAB1[2] = "NC2_AX1"
$MN_AXCT_AXCONF_ASSIGN_TAB1[3] = "NC2_AX2"

$SN_AXCT_SWWIDTH[0] = 1

CHANDATA(1)
$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=5
$MC_AXCONF_MACHAX_USED[2]=4
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0
$MC_AXCONF_CHANAX_NAME_TAB[0] = "XR"
$MC_AXCONF_CHANAX_NAME_TAB[1] = "YR"
$MC_AXCONF_CHANAX_NAME_TAB[2] = "ZR"

CHANDATA(2)
$MC_REFP_NC_START_LOCK=0
$MC_AXCONF_MACHAX_USED[0]=2
$MC_AXCONF_MACHAX_USED[1]=6
$MC_AXCONF_MACHAX_USED[2]=3
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0
$MC_AXCONF_CHANAX_NAME_TAB[0] = "XX"
$MC_AXCONF_CHANAX_NAME_TAB[1] = "YY"
```

```
$MC_AXCONF_CHANAX_NAME_TAB[2] = "ZZ"
M30

; ########## NCU-2 ##########
$MN_NCU_LINKNO = 2
$MN_MM_NCU_LINK_MASK = 1
$MN_MM_LINK_NUM_OF_MODULES= 2
$MN_MM_SERVO_FIFO_SIZE = 3

$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "CT1_SL1"
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "CT1_SL2"
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC2_AX3"
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = "NC2_AX4"
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = "NC2_AX5"
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = "NC2_AX6"

CHANDATA(1)
$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=2
$MC_AXCONF_MACHAX_USED[2]=3
$MC_AXCONF_MACHAX_USED[3]=4
$MC_AXCONF_MACHAX_USED[4]=5
$MC_AXCONF_MACHAX_USED[5]=6
$MC_AXCONF_MACHAX_USED[6]=0
M30
```

5.4.6 Besonderheiten der Interpolatorischen Kompensation

Messen

Bei der Funktion "Messen" werden die vom Bediener bzw. Programmierer benötigten kompensierten Istpositionen (ideale Maschine) geliefert.

TEACH IN

Auch bei der Funktion "TEACH IN" werden die abzuspeichernden Istpositionen aus kompensierten Positionswerten bestimmt.

Software-Endschalter

Bei den Software-Endschaltern werden ebenfalls die idealen Positionswerte (d. h. die von MSFK und Losekompensation korrigierten Lageistwerte) überwacht.

Positionsanzeige

Die Positions-Istwertanzeige im Maschinenkoordinatensystem zeigt den idealen (programmierten) Lageistwert der Achse (ideale Maschine) an.

In der Service-Anzeige "Achse/Spindel" (Bedienbereich "Diagnose") wird der vom Messsystem erfasste Positionsistwert zuzüglich der Summe von MSFK und Losekompensation angezeigt (= Lageistwert Messsystem 1/2).

Kompensationswert-Anzeige

In der Service-Anzeige "Achse/Spindel" (Bedienbereich "Diagnose") werden folgende Kompensationswerte angezeigt:

- Absoluter Kompensationswert Messsystem 1 bzw. 2

  Anzeigewert ist die zur aktuellen Istposition der Achse (Messsystem 1 oder 2) gehörige Summe der Korrekturwerte aus MSFK und Losekompensation.

- Kompensation Durchhang + Temperatur

  Anzeigewert ist die zur aktuellen Istposition der Achse gehörige Summe der Korrekturwerte aus Durchhang- und Temperaturkompensation.

Literatur:
Funktionshandbuch Grundfunktionen; Diagnosehilfsmittel (D1)

Referenzpunktverlust

Geht bei der Basisachse der Referenzpunkt verloren (DB31, ... DBX60.4 bzw. 60.5 = 0), werden MSFK und Durchhangkompensation in den betroffenen Achsen ausgeschaltet. Mit Erreichen des Referenzpunkts werden sie automatisch wieder aktiviert.

Zugriffsschutz

Derzeit existiert für die Kompensationstabellen kein Zugriffsschutz.

Setzen von Reglerfreigaben

Da infolge der Kompensationsbeziehung die Verfahrbewegung einer Basisachse auch zu Bewegungen der Kompensationsachse führen kann, müssen auch für diese Achsen die Reglerfreigaben gesetzt werden (PLC-Anwenderprogramm). Ansonsten funktioniert die Kompensation nur eingeschränkt.

Ausgabe der Verfahrsignale

Die Verfahrsignale in der Kompensationsachse werden beim Ein-/Ausschalten der Kompensation und bei jeder Änderung der Anzahl aktiver Kompensationstabellen ausgegeben.

Eine aus der Bewegung der Basisachse resultierende Kompensationswertänderung führt nicht zur Ausgabe von Verfahrsignalen in der Kompensationsachse.

5.5 Dynamische Vorsteuerung (Schleppfehler-Kompensation)

5.5.1 Allgemeine Eigenschaften

Axialer Schleppfehler

Als axialer Schleppfehler wird die bleibende Regelabweichung des Lagereglers beim Verfahren einer Maschinenachse bezeichnet. Anders ausgedrückt, der axiale Schleppfehler ist die Differenz zwischen der Sollposition und der Istposition der Maschinenachse.

Auswirkungen

Der Schleppfehler führt insbesondere bei Beschleunigungsvorgängen an Konturkrümmungen, z. B. Kreisen und Ecken, zu einem unerwünschten, geschwindigkeitsabhängigen Konturfehler.

Kompensation

Mit Hilfe der "Dynamischen Vorsteuerung" kann der axiale Schleppfehler nahezu auf Null reduziert werden. Die Funktion wird daher auch als "Schleppfehler-Kompensation" bezeichnet.

Methoden

Es gibt zwei Methoden der "Dynamischen Vorsteuerung":

- Drehzahlvorsteuerung(geschwindigkeitsabhängig)
- Momentenvorsteuerung(beschleunigungsabhängig)

Hinweis

Die Momentenvorsteuerung ist eine lizenzpflichtige Option und steht nur für SINUMERIK 840D sl zur Verfügung. Für SINUMERIK 828D ist die Funktion nicht verfügbar.

Aktivierung

Die Auswahl und Aktivierung der Vorsteuerungsmethode erfolgt über das Maschinendatum:

MD32620 $MA_FFW_MODE (Vorsteuerungsart)

Wert	Bedeutung
0	keine Vorsteuerung
1	Drehzahlvorsteuerung mit PT1-Symmetrierung
2	Momentenvorsteuerung mit PT1-Symmetrierung
3	Drehzahlvorsteuerung mit Tt-Symmetrierung (Standard-Einstellung für 840D sl / 840Di sl)
4	Momentenvorsteuerung mit Tt-Symmetrierung

Hinweis

Hochrüstung auf 840D sl und 840Di sl

Bei Hochrüstung auf SINUMERIK 840D sl und 840Di sl müssen Inbetriebnahme-Einstellungen neu vorgenommen werden.

Wurde die Vorsteuer-Variante MD32620 = 3 bereits genutzt, so ist bei Software-Hochrüstung die Inbetriebnahme-Einstellung von MD32810 $MA_EQUIV_SPEEDCTRL_TIME (Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung) neu vorzunehmen, da die Werte Ti und To automatisch berücksichtigt werden. Diese Werte müssen im MD32810 manuell korrigiert werden.

Ein-/Ausschalten im Teileprogramm

Über das folgende achsspezifische Maschinendatum kann festgelegt werden, dass die Vorsteuerung für diese Achse/Spindel vom Teileprogramm ein- und ausschaltbar ist:

MD32630 $MA_FFW_ACTIVATION_MODE (Vorsteuerung aktivieren von Programm)

Wert	Bedeutung
0	Die Vorsteuerung kann nicht über das Teileprogramm ein- bzw. ausgeschaltet werden. Für die Achse/Spindel wirkt somit stets der mit MD32620 $MA_FFW_MODE vorgegebene Zustand.
1	Die Vorsteuerung kann vom Teileprogramm ein- bzw. ausgeschaltet werden. Die Anweisung wird sofort wirksam.
2	Die Vorsteuerung kann vom Teileprogramm ein- bzw. ausgeschaltet werden. Die Anweisung wird erst im nächsten Achsstillstand wirksam.

Das Ein-/Ausschalten der Vorsteuerung über das Teileprogramm erfolgt über die Anweisungen:

`FFWON`: Vorsteuerung EIN

`FFWOF`: Vorsteuerung AUS

Die Standardeinstellung (d. h. auch nach Reset, M30) wird vorgegeben mit dem kanalspezifischen Maschinendatum:

MD20150 $MC_GCODE_RESET_VALUES (Löschstellung der G-Gruppen)

`FFWON`/`FFWOF` wird für alle Achsen/Spindeln im Achsbetrieb wirksam, bei denen:

MD32630 $MA_FFW_ACTIVATION_MODE = 1 (bzw. 2)

und

MD32620 $MA_FFW_MODE = 1, 2, 3 oder 4

Bei miteinander interpolierenden Achsen sollte die MD32630-Einstellung identisch eingestellt sein.

Das Ein- bzw. Ausschalten der Vorsteuerung sollte nur während des Stillstandes der Achse/Spindel im Achsbetrieb erfolgen, um ein Rucken zu vermeiden. Deshalb wird die Umschaltung automatisch bis zum nächsten Stillstand mittels Vorlaufstopp verzögert.

Dabei bitte Folgendes beachten:

Für asynchron zur Teileprogrammabarbeitung fahrende Kommando- oder PLC-Achsen hat ein Vorlaufstopp keine Wirkung. Um sicherzustellen, dass `FFWON/FFWOF` erst beim nächsten Stillstand der Achse/Spindel im Achsbetrieb wirkt, muss explizit für jede Achse/Spindel im Achsbetrieb MD32630 = 2 eingestellt werden (siehe auch "Vorsteuerung bei Kommando- und PLC-Achsen (Seite 406)").

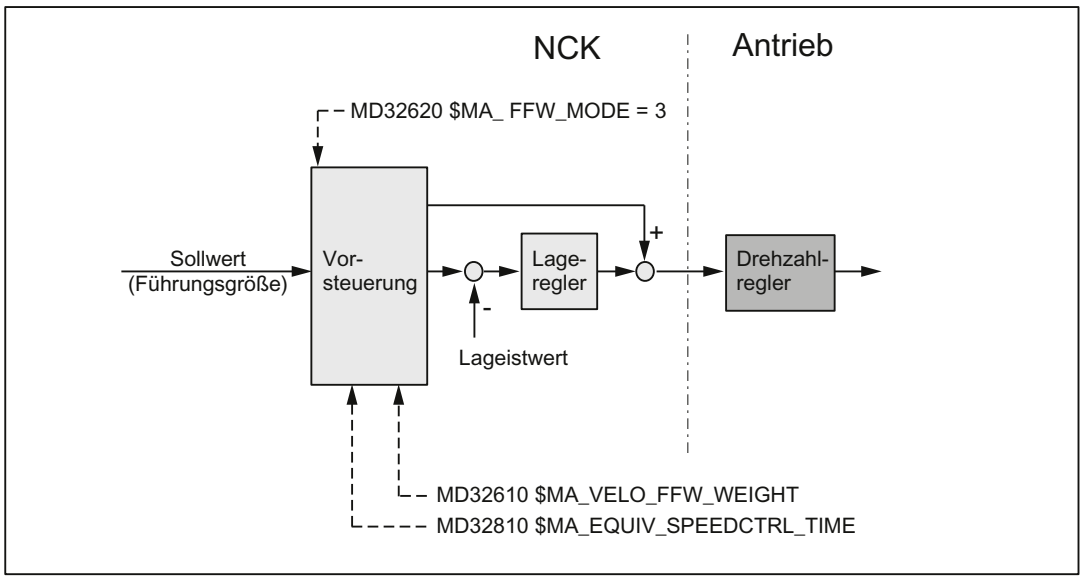
5.5.2 Drehzahlvorsteuerung

Funktion

Bei der Drehzahlvorsteuerung wird zusätzlich ein Geschwindigkeitssollwert auf den Eingang des Drehzahlreglers gegeben. Damit kann bei konstanter Geschwindigkeit der Schleppabstand fast vollständig reduziert werden (d. h. die Regeldifferenz wird 0).

Inbetriebnahme

Für die Drehzahlvorsteuerung sind folgende achsspezifische Parameter bei der Inbetriebnahme festzulegen:



Ersatzzeitkonstante des Drehzahlregelkreises (MD32810)

Für eine korrekt eingestellte Drehzahlvorsteuerung ist die Ersatzzeitkonstante des Drehzahlregelkreises genau zu bestimmen (z. B. grafisch aus einer Drehzahlsollwert-Sprungantwort) und in das folgende Maschinendatum einzutragen:

MD32810 $MA_EQUIV_SPEEDCTRL_TIME (Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung)

Literatur:
Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (G2), Kapitel: "Optimierung der Regelung"

Vorsteuerfaktor für Drehzahlvorsteuerung (MD32610)

Der zusätzliche Geschwindigkeitssollwert kann mit einem Faktor gewichtet werden:

MD32610 $MA_VELO_FFW_WEIGHT

Wertebereich: 0 ... 1

"0" bedeutet: keine Vorsteuerung. Standardmäßig hat der Faktor den Wert 1 (≙ 100%).

Der Faktor sollte auf 100% eingestellt bleiben, da dieser Wert bei einem optimal eingestellten Regelkreis für die Achse/Spindel sowie einer exakt ermittelten Ersatzzeitkonstanten des Drehzahlregelkreises die optimale Einstellung ist.

Feinabstimmung

Durch geringfügige Veränderungen (Feinabstimmung) der Ersatzzeitkonstanten des Drehzahlregelkreises (MD32810) kann die Drehzahlvorsteuerung für die jeweilige Achse/Spindel optimiert werden.

Zur Überprüfung ist die Achse/Spindel mit konstanter Geschwindigkeit zu verfahren und die Reaktion der Maschinendatenänderungen in der Service-Anzeige "Achse/Spindel" die "Regeldifferenz" zu kontrollieren.

Für das Abgleichkriterium der Drehzahlvorsteuerung gilt:

- Bei positiver Verfahrrichtung der Achse zeigt die "Regeldifferenz" einen **positiven** Wert an.

  ⇒ Die Ersatzzeitkonstante des Drehzahlregelkreises ist **zu klein**.

- Bei positiver Verfahrrichtung der Achse zeigt die "Regeldifferenz" einen **negativen** Wert an.

  ⇒ Die Ersatzzeitkonstante des Drehzahlregelkreises ist **zu groß**.

Um dies an der Service-Anzeige gut feststellen zu können, wählt man am besten eine kleine Beschleunigung und einen großen Vorschub. Man erhält dadurch sehr lange Beschleunigungsphasen, in denen sich die Regeldifferenz gut ablesen lässt.

Literatur:

- Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (G2), Kapitel "Optimierung der Regelung"
- Funktionshandbuch Grundfunktionen; Diagnosehilfsmittel (D1)

Vorhaltezeit

Für die Ausgabe der Drehzahlsollwerte kann eine Vorhaltezeit festgelegt werden.

Literatur:
SINAMICS S120 Funktionshandbuch

Beispiel

Teileprogramm zum Einstellen der Ersatzzeitkonstanten für die Achse X

```
Programmcode                                    Kommentar
MD32300 $MA_MAX_AX_ACCEL=0,1                    ; Einheit: m/s2
MD32000 $MA_MAX_AX_VELO=20000,0                 ; Einheit: mm/min
; Teileprogramm zum Einstellen der Ersatzzeitkonstanten
G1 F20000
FFWON
LOOP:
X1000
X0
GOTOB LOOP
M30
```

5.5.3 Momentenvorsteuerung - nur 840D sl (Option)

Funktion

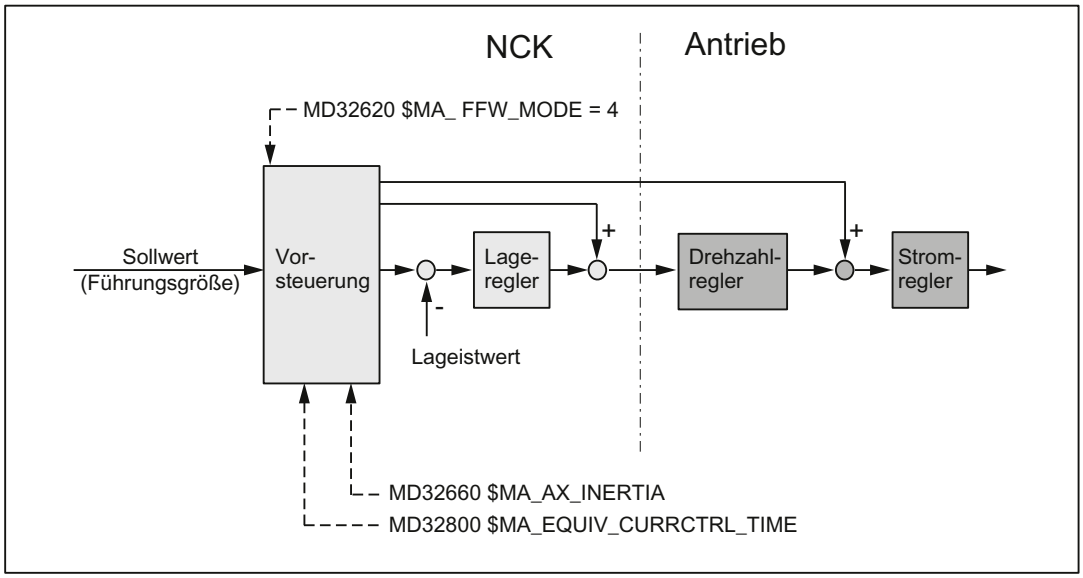
Bei der Momentenvorsteuerung wird ein dem Moment proportionaler zusätzlicher Stromsollwert direkt auf den Eingang des Stromreglers aufgeschaltet. Dieser Wert wird mit Hilfe der Beschleunigung und des Trägheitsmoments gebildet.

Anwendung

Die Momentenvorsteuerung wird zum Erreichen hoher Konturgenauigkeiten bei höchsten Dynamikanforderungen benötigt. Damit kann bei richtiger Einstellung der Schleppabstand auch bei Beschleunigungsvorgängen fast vollständig kompensiert werden.

Inbetriebnahme

Für die Momentenvorsteuerung sind folgende achsspezifische Parameter bei der Inbetriebnahme festzulegen:



Ersatzzeitkonstante des Stromregelkreises (MD32800)

Für eine korrekt eingestellte Momentenvorsteuerung ist die Ersatzzeitkonstante des Stromregelkreises genau zu bestimmen (z. B. grafisch aus der Sprungantwort des Stromregelkreises) und in das folgende Maschinendatum einzutragen:

MD32800 $MA_EQUIV_CURRCTRL_TIME (Ersatzzeitkonstante Stromregelkreis für Vorsteuerung)

Literatur:
Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (G2), Kapitel: "Optimierung der Regelung"

Gesamt-Trägheitsmoment der Achse (MD32650)

Für die Momentenvorsteuerung ist das Gesamt-Trägheitsmoment (Trägheitsmoment von Antrieb + Last bezogen auf die Motorwelle) der Achse zu ermitteln und in das folgende Maschinendatum einzugeben:

MD32650 $MA_AX_INERTIA (Trägheit für Drehmomentvorsteuerung)

Feinabstimmung

Durch geringfügige Veränderungen (Feinabstimmung) der Werte in MD32800 und MD32650 kann die Momentenvorsteuerung für die jeweilige Achse/Spindel optimiert werden.

Zur Überprüfung sollte der Schleppabstand über einen analogen Sollwertausgang aufgezeichnet werden. Dabei ist der Schleppabstand neben der Konstantfahrt insbesondere auch bei Beschleunigungsvorgängen der Achse/Spindel zu beobachten.

Für das Abgleichkriterium der Momentenvorsteuerung gilt:

- Bei positiver Verfahrrichtung der Achse hat der aufgezeichnete Schleppabstand einen **positiven** Wert.

  ⇒ Der eingegebene Wert für die Ersatzzeitkonstante des Stromregelkreises oder für das Trägheitsmoment der Achse ist **zu klein**.

- Bei positiver Verfahrrichtung der Achse hat der aufgezeichnete Schleppabstand einen **negativen** Wert.

  ⇒ Der eingegebene Wert für die Ersatzzeitkonstante des Stromregelkreises oder für das Trägheitsmoment der Achse ist **zu groß**.

Hinweis

Aufgrund der sehr schnellen Zeitabläufe bei den Beschleunigungsvorgängen kann bei der Inbetriebnahme der Momentenvorsteuerung die Service-Anzeige nicht zur Überprüfung der Feinabstimmung verwendet werden.

5.5.4 Dynamikanpassung

Funktion

Bei miteinander interpolierenden Achsen mit unterschiedlichem Zeitverhalten der axialen Regelkreise kann über die Dynamikanpassung das gleiche Zeitverhalten erzielt werden, um eine optimale Konturgenauigkeit ohne Verlust von Regelgüte zu erhalten.

Inbetriebnahme

Zeitkonstante der Dynamikanpassung (MD32910)

Als Zeitkonstante der Dynamikanpassung ist die Differenz der Ersatzzeitkonstanten des "langsamsten" Drehzahl- bzw. Stromregelkreises zu der jeweiligen Achse in das folgende Maschinendatum einzugeben.

MD32910 $MA_DYN_MATCH_TIME (Zeitkonstante der Dynamikanpassung)

Beispiel:

Ersatzzeitkonstanten des Drehzahlregelkreises (MD32810) bei aktiver Drehzahlvorsteuerung der Achsen 1, 2 und 3:

- Achse 1: 2 ms
- Achse 2: 4 ms (ist dynamisch langsamste Achse)
- Achse 3: 1 ms

Damit ergeben sich für die Zeitkonstante der Dynamikanpassung (MD32910) folgende Werte:

- Achse 1: 2 ms
- Achse 2: 0 ms
- Achse 3: 3 ms

Aktivierung (MD32900)

Die Dynamikanpassung ist nur wirksam, wenn das folgende Maschinendatum gesetzt ist:

MD32900 $MA_DYN_MATCH_ENABLE = 1

Literatur

Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (G2), Kapitel: "Optimierung der Regelung"

5.5.5 Vorsteuerung bei Kommando- und PLC-Achsen

Funktion

Bei Kommando- und PLC-Achsen muss wie folgt verhindert werden, dass die Vorsteuerung bei höheren Geschwindigkeiten ein-/ausgeschaltet wird:

MD32630 $MA_FFW_ACTIVATION_MODE = 2

Mit dieser Einstellung wird die `FFWON/FFWOF`-Anweisung ausschließlich unterhalb der für diese Achse projektierten Stillstandsgeschwindigkeit (MD36060 $MA_STANDSTILL_VELO_TOL) wirksam.

Fällt die Umschaltungsanweisung mit einer Achsbewegung zusammen, wird die angeforderte Umschaltung erst im nachfolgenden Stillstand der Achse ausgeführt. Damit wird ein schlagartiger Auf-/Abbau des Schleppfehlers vermieden.

ACHTUNG
Eine zu hoch eingestellte Stillstandsgeschwindigkeit kann zum Umschalten der Vorsteuerung in der Bewegung führen. Abhängig vom vorhandenen Schleppfehler können Überwachungen ansprechen.

Inbetriebnahme

Um die Vorsteuerung bei Kommando- und PLC-Achsen zu überprüfen, empfiehlt sich folgendes Vorgehen:

1. Stillstandsgeschwindigkeit im MD36060 kontrollieren.
2. Vorhandenen Schleppfehler der Achse im Stillstand kontrollieren.
3. Umschaltbedingung einstellen und wirksam setzen:

   MD32630 = 2
4. Achse im Teileprogramm über `POSA`-Anweisung verfahren.
5. `FFWON` während der Achsbewegung ausführen.

6. Der in der Service-Anzeige "Achse/Spindel" angezeigte K_v-Faktor und Schleppfehler darf dabei nicht springen.
7. Ein höherer K_v-Faktor und ein kleinerer Schleppfehler stellen sich erst bei einer dem Stillstand folgenden Verfahrbewegung ein. Die Vorsteuerung ist erst ab dem Stillstand aktiv.

Analog zum Einschalten der Vorsteuerung gilt für den Ausschaltvorgang:

1. Achse im Teileprogramm über `POSA`-Anweisung verfahren.
2. `FFWOF` während der Achsbewegung ausführen.
3. Der in der Service-Anzeige "Achse/Spindel" angezeigte K_v-Faktor und Schleppfehler darf dabei nicht springen.
4. Ein niedrigerer K_v-Faktor und ein größerer Schleppfehler stellen sich erst bei einer dem Stillstand folgenden Verfahrbewegung ein. Die Vorsteuerung ist erst ab dem Stillstand inaktiv.

Beispiel

Im nachfolgenden Programmbeispiel wird die Achse A asynchron zur Bahn verfahren. Während der Verfahrbewegung wird versucht, die Vorsteuerung im Kanal einzuschalten. Im Gegensatz zu den Geometrieachsen X, Y und Z wird die Vorsteuerung bei der Achse A aber nicht sofort wirksam. Hier wird der Stillstand nach dem Satz N60 abgewartet. Im Satz N70 fährt die Achse A dann mit Vorsteuerung.

Programmcode

```
N10 FFWOF
N20 POSA[A]=1000 FA[A]=10000
N30 G4 F1
N40 FFWON
N50 G0 X10 Y10 Z10
N60 WAITP(A)
N70 POSA[A]=1500 FA[A]=10000
N80 WAITP(A)
M30
```

5.5.6 Randbedingungen

Miteinander interpolierende Achsen

Auch bei miteinander interpolierenden Achsen sind die Vorsteuer-Parameter **für jede Achse** optimal einzustellen, d. h. auch miteinander interpolierende Achsen können unterschiedliche Vorsteuer-Parameter haben.

Konturüberwachung überprüfen

Da die beiden Ersatzzeitkonstanten:

- MD32810 $MA_EQUIV_SPEEDCTRL_TIME (Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung)

und

- MD32800 $MA_EQUIV_CURRCTRL_TIME) (Ersatzzeitkonstante Stromregelkreis für Vorsteuerung)

auch die Konturüberwachung beeinflussen, ist diese anschließend zu überprüfen.

Literatur:
Funktionshandbuch Grundfunktionen; Achsüberwachungen, Schutzbereiche (A3)

Einfluss auf Kv-Faktor

Bei richtig eingestellter Vorsteuerung erhält die Regelstrecke für das Verhalten der Führungsgröße bei Drehzahlvorsteuerung die Dynamik des Drehzahlregelkreises bzw. bei Momentenvorsteuerung die Dynamik des Stromregelkreises; d. h. der in MD32200 $MA_POS_CTRLGAIN eingetragene K_v-Faktor hat kaum Einfluss auf das Führungsverhalten (z. B. Eckenfehler, Überschwingen, Kreis-/Radiusfehler).

Die Vorsteuerung wiederum hat keinen Einfluss auf das Störverhalten (Gleichlauf). Hierbei ist der im MD32200 vorgegebene K_v-Faktor der wirksame Faktor.

Service-Anzeige "Kv-Faktor"

Bei aktiver Vorsteuerung wird der in der Service-Anzeige "Achse/Spindel" angezeigte K_v-Faktor der Achse (entspricht dem für das Führungsverhalten wirksamen K_v-Faktor) sehr groß.

5.6 Reibkompensation (Quadrantenfehler-Kompensation)

5.6.1 Allgemeine Eigenschaften

Funktion

Reibung tritt vor allem in den Getrieben und Führungsbahnen auf. Besonders macht sich bei den Maschinenachsen die Haftreibung bemerkbar. Da das Auslösen einer Bewegung eine höhere Kraft erfordert (zum Losbrechen) als das Weiterfahren, erhält man vorübergehend zu Beginn einer Bewegung einen erhöhten Schleppfehler.

Entsprechendes ist auch bei einer Richtungsumkehr feststellbar, bei der die Haftreibung einen Sprung der Reibkraft bewirkt. Wird eine Achse beispielsweise von einer negativen auf eine positive Geschwindigkeit beschleunigt, so bleibt sie beim Nulldurchgang der Geschwindigkeit wegen der wechselnden Reibverhältnisse kurzzeitig hängen. Bei miteinander interpolierenden Achsen führen somit wechselnde Reibverhältnisse zu Konturfehlern.

Quadrantenfehler

Besonders deutlich tritt dieses Verhalten bei Kreiskonturen auf, wo sich bei den Quadrantenübergängen eine Achse mit der maximalen Bahngeschwindigkeit bewegt, während die zweite Achse steht. Mit Hilfe der Reibungskompensation können diese so genannten "Quadrantenfehler" an Kreiskonturen fast vollständig kompensiert werden.

Prinzip

Messungen an Maschinen haben ergeben, dass die durch Haftreibung verursachten Konturfehler durch Aufschaltung eines zusätzlichen Sollwertimpulses mit entsprechendem Vorzeichen und Amplitude gut kompensiert werden können.

Reibkompensationsarten

Bei SINUMERIK 840D kann zwischen zwei Reibkompensationsmethoden ausgewählt werden

(MD32490 $MA_FRICT_COMP_MODE (Art der Reibkompensation)):

- konventionelle Reibkompensation

  (MD32490 = 1)

Hierbei kann die Intensität des Korrekturimpulses entsprechend einer Kennlinie in Abhängigkeit von der Beschleunigung eingestellt werden. Diese Kennlinie muss bei der Inbetriebnahme mit Hilfe des Kreisformtests bestimmt und parametriert werden. Das dazu notwendige Vorgehen ist relativ aufwendig und erfordert daher einige Erfahrung.

- Quadrantenfehlerkompensation mit neuronalen Netzen

  (Option bei SINUMERIK 840D)

  (MD32490 = 2)

Zur Vereinfachung der Inbetriebnahme muss die Kompensationskennlinie nicht mehr durch den Inbetriebnehmer von Hand eingestellt werden, sondern sie wird während einer Lernphase automatisch ermittelt und im gepufferten Anwenderspeicher abgelegt.

Das neuronale Netz kann den Verlauf der Kompensationskennlinie erheblich besser nachbilden, wodurch eine verbesserte Genauigkeit erzielt wird.

Desweiteren gestattet es jederzeit eine einfache automatische Nachoptimierung vor Ort.

Kreisformtest

Die Inbetriebnahme der Reibungskompensation (sowohl konventionelle als auch neuronale Reibkompensation) erfolgt am einfachsten mit einem Kreisformtest. Hierbei werden während des Abfahrens einer Kreiskontur die Istposition gemessen und die Abweichungen vom programmierten Radius (insbesondere an den Quadrantenübergängen) graphisch dargestellt. Die Erfassung erfolgt mit Hilfe eines sogenannten "Trace", der im passiven Dateisystem abgespeichert wird.

Der Kreisformtest ist eine Funktion des "Inbetriebnahme-Tools" und kann auch im Inbetriebnahme-Bereich der Bedienoberfläche unter HMI angewählt werden. Weitere Informationen siehe Kapitel "Kreisformtest".

5.6.2 Konventionelle Reibkompensation

5.6.2.1 Konventionelle Reibkompensation

Art der Reibkompensation

Die Anwahl der konventionellen Reibkompensation erfolgt durch Eintrag des Wertes = 1 in dem Maschinendatum

MD32490 $MA_FRICT_COMP_MODE (Art der Reibkompensation).

Amplituden-Adaption

In vielen Fällen ist die Aufschaltamplitude des Reibkompensationswertes über den gesamten Beschleunigungsbereich nicht konstant. So muss für eine optimale Kompensation bei höheren Beschleunigungen ein kleinerer Kompensationswert aufgeschaltet werden als bei kleineren Beschleunigungen. Deshalb kann bei hohen Genauigkeitsanforderungen die Reibkompensation **mit adaptierter Aufschaltamplitude** aktiviert werden (siehe folgendes Bild). Die Aktivierung erfolgt achsspezifisch mit dem Maschinendatum

MD32510 $MA_FRICT_COMP_ADAPT_ENABLE = 1

(Adaption Reibkompensation aktiv).

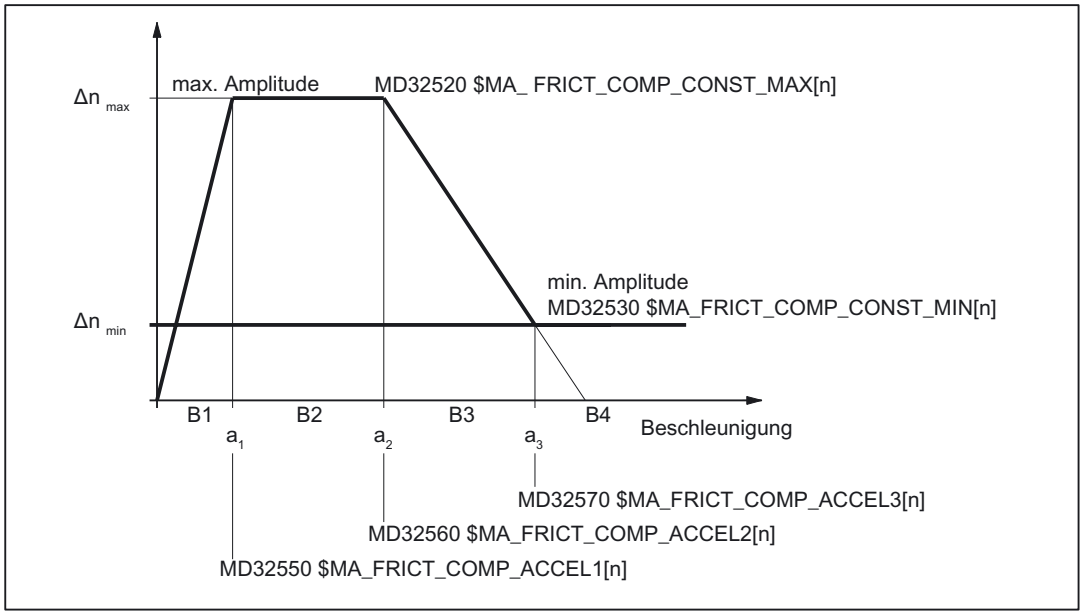


Bild 5-13 Typische Kennlinie für die Reibkompensation mit Amplituden-Adaption

Die Adaptionskennlinie ist dabei in folgende vier Bereiche unterteilt (je Bereich wirkt eine andere Aufschaltamplitude Δn):

B1:	für $a < a_1$	$\Delta n = \Delta n_{max} * a / a_1$
B2:	für $a_1 \leq a \leq a_2$	$\Delta n = \Delta n_{max}$
B3:	für $a_2 < a < a_3$	$\Delta n = \Delta n_{max} * (1 - (a - a_2) / (a_3 - a_2))$
B4:	für $a \geq a_3$	$\Delta n = \Delta n_{min}$

Kennlinienparameter

Die Parameter der Adaptionskennlinie vom vorhergehenden Bild sind als Maschinendaten achsspezifisch einzugeben.

Δn	=	Aufschaltamplitude des Reibkompensationswertes
Δn_{max}	=	maximaler Reibkompensationswert MD32520 $FRICT_COMP_CONST_MAX[n] (Maximaler Reibkompensationswert)
Δn_{min}	=	minimaler Reibkompensationswert MD32530 $FRICT_COMP_CONST_MIN[n] (Minimaler Reibkompensationswert)
a_1	=	Adaptions-Beschleunigungswert 1 bei Reibkompensation MD32550 $FRICT_COMP_ACCEL1[n] (Adaptions-Beschleunigungswert 1)

a_2 = Adaptions- Beschleunigungswert 2 bei Reibkompensation
MD32560 $FRICT_COMP_ACCEL2[n]
(Adaptions-Beschleunigungswert 2)

a_3 = Adaptions- Beschleunigungswert 3 bei Reibkompensation
MD32570 $FRICT_COMP_ACCEL3[n]
(Adaptions-Beschleunigungswert 3)

Anmerkung zum Kennlinienverlauf

In Sonderfällen kann die ermittelte Kennlinie von dem im vorhergehenden Bild typischen Verlauf abweichen.

Dabei kann sogar der Wert für Δn_{min} (MD32530) größer sein als Δn_{max} (MD32520).

5.6.2.2 Inbetriebnahme der konventionellen Reibungskompensation

Kreisformtest

Die Inbetriebnahme der Reibungskompensation erfolgt am einfachsten mit einem Kreisformtest. Hierbei lassen sich während des Abfahrens einer Kreiskontur die Abweichungen vom programmierten Radius (insbesondere an den Quadrantenübergängen) messtechnisch erfassen und graphisch darstellen.

Schrittweise Inbetriebnahme

Zunächst ist die konventionelle Reibkompensation anzuwählen.

(MD32490 $MA_FRICT_COMP_MODE=1 (Art der Reibkompensation)).

Der Reibkompensationswert ist im wesentlichen von der Maschinenkonfiguration abhängig. Bei der Inbetriebnahme wird in zwei Stufen vorgegangen:

- Stufe 1: Ermittlung der Kompensationswerte ohne Adaption
- Stufe 2: Ermittlung der Adaptionskennlinie (falls die Reibkompensation beschleunigungsabhängig ist und die Ergebnisse von Stufe 1 nicht befriedigend sind).

Inbetriebnahme-Stufe 1: Reibkompensation ohne Adaption

1. Kreisformtest ohne Reibkompensation

Zunächst sollte ein Kreisformtest ohne Reibkompensation gefahren werden

(MD32500 $MA_FRICT_COMP_ENABLE = 0)

Das Vorgehen für den Kreisformtest ist in Kapitel "Kreisformtest" beschrieben.

Ein typisches Aussehen der Quadrantenübergänge ohne Reibkompensation ist im folgenden Bild dargestellt.

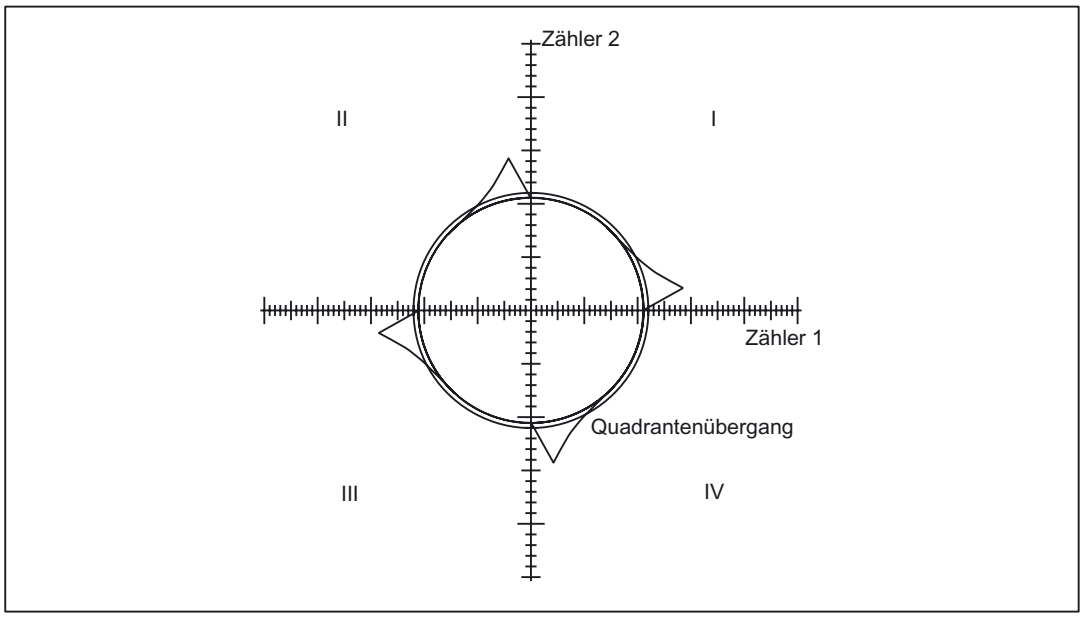


Bild 5-14 Nicht kompensierte Radiusabweichung an den Quadrantenübergängen

2. Freigabe der Reibkompensation

Danach ist die für die entsprechende Achse/Spindel die Reibkompensation zu aktivieren.

Reibkompensation aktiv setzen mit Maschinendatum

→ MD32500 $MA_FRICT_COMP_ENABLE[n] = 1

(Reibkompensation aktiv)

3. Adaption ausschalten

Um die Reibkompensation ohne Adaption in Betrieb zu nehmen, ist die Adaption auszuschalten.

Adaption ausschalten mit dem Maschinendatum

→ MD32510 $MA_FRICT_COMP_ADAPT_ENABLE[n] = 0

(Adaption Reibkompensation aktiv)

4. Kompensationsparameter ermitteln

Die Reibkompensation ohne Adaption wird von folgenden Parametern bestimmt:

1. MD32520 $MA_FRICT_COMP_CONST_MAX[n] (Maximaler Reibkompensationswert) (Amplitude) in [mm/min]
2. MD32540 $MA_FRICT_COMP_TIME[n] (Reibkompensations-Zeitkonstante) in [s]

Diese beiden Parameter werden jeweils so lange verändert, bis beim Kreisformtest an den Quadrantenübergängen die Abweichungen vom programmierten Radius minimal werden bzw. völlig verschwinden. Dabei sind die Tests mit verschiedenen Radien und Geschwindigkeiten (typische Werte für den Anwendungsbereich der Maschine) durchzuführen.

Startwert

Zu Beginn der Messungen sollte als Startwert eine relativ kleine Aufschaltamplitude sowie Zeitkonstante von wenigen Lageregler-Takten vorgegeben werden.

Beispiel:

MD32520 $MA_FRICT_COMP_CONST_MAX[n] = 10 (mm/min)

MD32540 $FRICT_COMP_TIME[n] = 0,008 (8 ms)

Die Wirkung der Parameteränderungen ist anhand der ermittelten und graphisch dargestellten Kreisformen zu kontrollieren.

Mittelwertbildung

Falls sich für die verschiedenen Radien und Geschwindigkeiten keine einheitliche Kompensationszeitkonstante finden lässt, bildet man den Mittelwert aus den ermittelten Zeitkonstanten.

Gut eingestellte Reibkompensation

Bei gut eingestellter Reibkompensation sind keine Quadrantenübergänge mehr erkennbar.

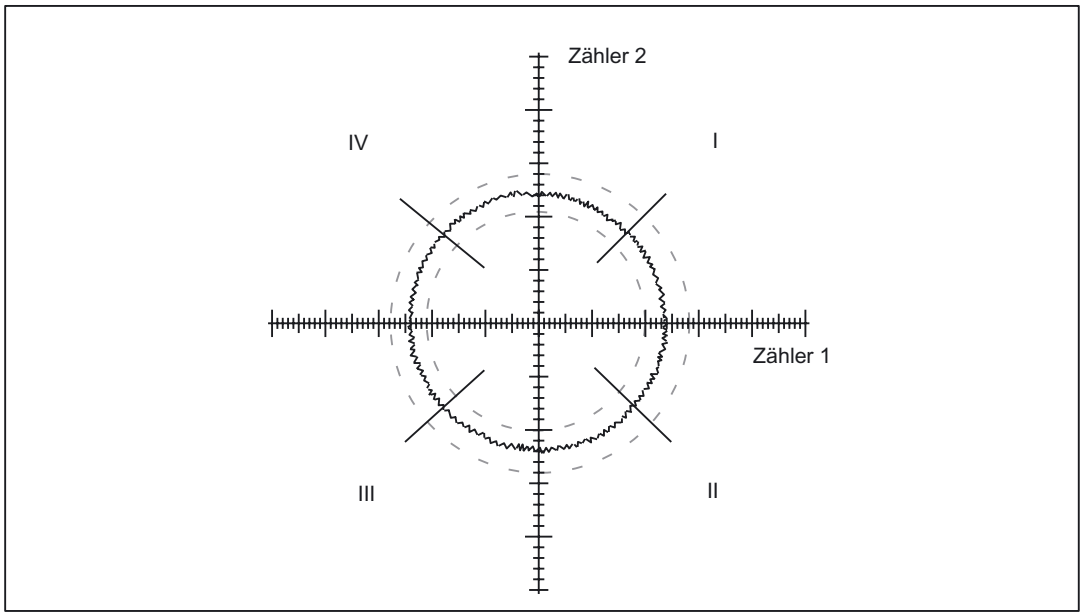


Bild 5-15 Quadrantenübergänge bei gut eingestellter Reibkompensation

Amplitude zu klein

Eine zu klein eingestellte Aufschaltamplitude erkennt man im Kreisformtest daran, dass die Radiusabweichungen vom programmierten Radius an den Quadrantenübergängen nicht genügend kompensiert sind.

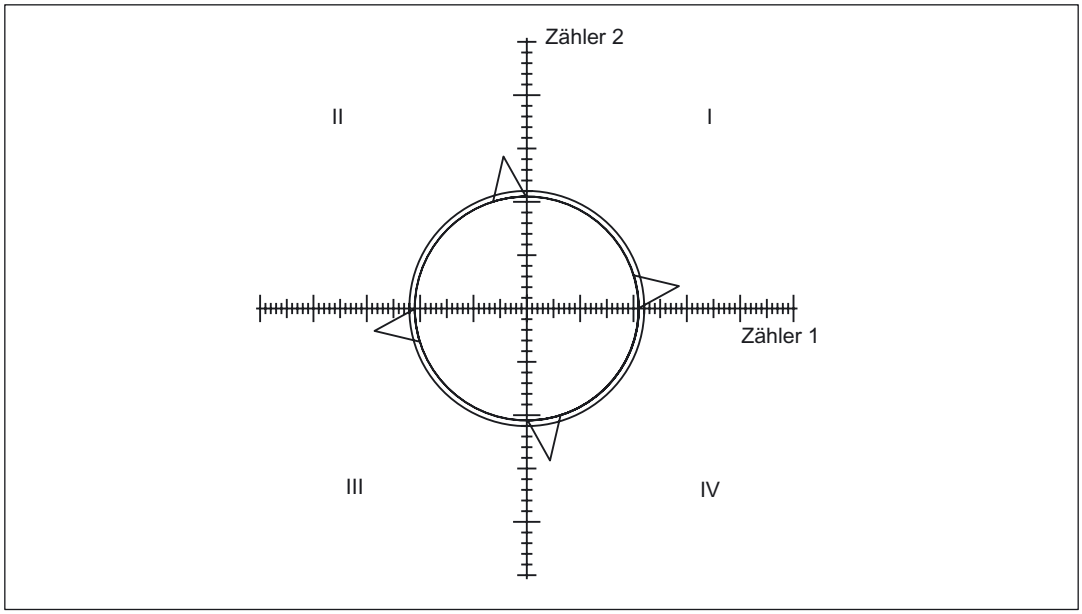


Bild 5-16 Zu kleine Aufschaltamplitude

Amplitude zu groß

Eine zu groß eingestellte Aufschaltamplitude erkennt man im Kreisformtest daran, dass eine deutliche Überkompensation der Radiusabweichungen an den Quadrantenübergängen erfolgt.

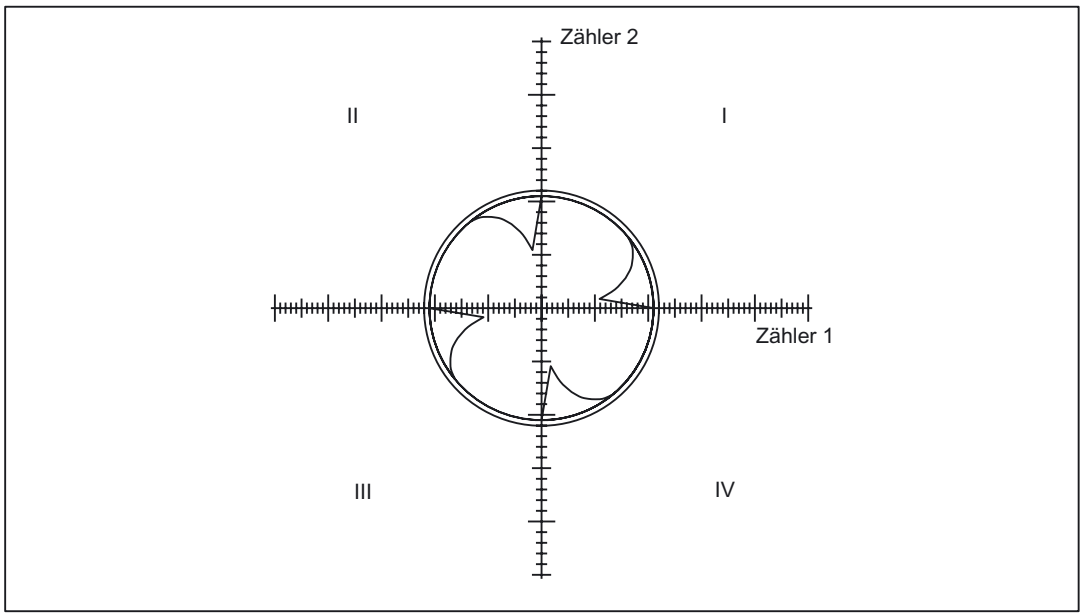


Bild 5-17 Zu große Aufschaltamplitude

Zeitkonstante zu klein

Zu klein eingestellte Kompensationszeitkonstanten sind im Kreisformtest daran erkennbar, dass bei den Quadrantenübergängen kurzzeitig die Radiusabweichung kompensiert wird, jedoch unmittelbar danach wieder größere Radiusabweichungen vom programmierten Radius auftreten.

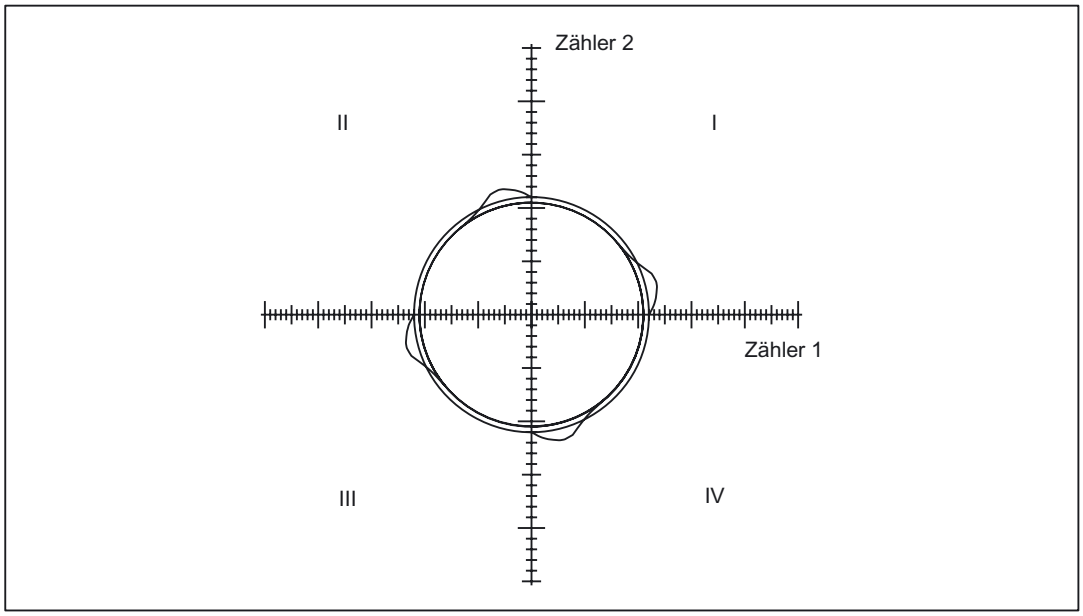


Bild 5-18 Zu kleine Kompensationszeitkonstante

Zeitkonstante zu groß

Zu große Werte der Kompensationszeitkonstanten sind im Kreisformtest daran erkennbar, dass bei den Quadrantenübergängen zwar die Radiusabweichung kompensiert wird (dabei ist vorausgesetzt, dass die optimale Aufschaltamplitude bereits ermittelt wurde), jedoch nach den Quadrantenübergängen die Radiusabweichung in Richtung Kreismitte deutlich zunimmt.

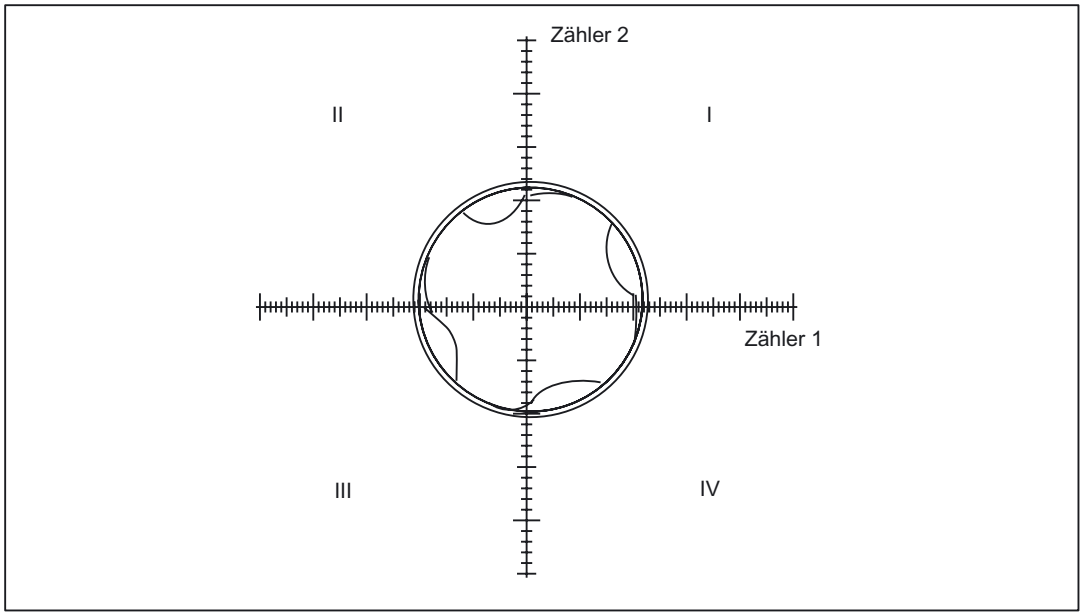


Bild 5-19 Zu große Kompensationszeitkonstante

Adaption ja/nein?

Lässt sich mit der so ermittelten Zeitkonstante und der konstanten Aufschaltamplitude im gesamten gewünschten Arbeitsbereich (d. h. für alle interessierenden Radien und Geschwindigkeiten) beim Kreisformtest sowie beim Positionieren ein gutes Ergebnis erzielen, kann die Kennlinien-Adaption entfallen.

Erweist sich jedoch die Reibkompensation als beschleunigungsabhängig, so muss in einem zweiten Schritt die Adaptionskennlinie ermittelt werden.

Inbetriebnahme-Stufe 2: Reibkompensation mit Adaption

Anwendung

Sobald die Reibkompensation beschleunigungsabhängig ist und mit der konstanten Aufschaltamplitude nicht die gewünschten Ergebnisse erreicht werden können, ist die Adaption anzuwenden.

Um bei hohen Genauigkeitsanforderungen eine optimale Kompensation im gesamten Arbeitsbereich der Reibvorsteuerung zu erhalten, muss die Beschleunigungsabhängigkeit des Kompensationswertes berücksichtigt werden. Dazu ist an verschiedenen Arbeitspunkten im Bereich zwischen Beschleunigung 0 und maximal vorgesehener Beschleunigung die entsprechende Abhängigkeit messtechnisch zu ermitteln. Die aus den Messergebnissen abzuleitende Adaptionskennlinie ist dann in den o. g. Maschinendaten achsspezifisch zu parametrieren.

1. Adaptionskennlinie ermitteln

Für verschiedene Radien und Geschwindigkeiten sind ...

1. ... die erforderlichen Aufschaltamplituden zu erfassen,
2. ... die Kompensationswirkung der Aufschaltamplituden mit Hilfe des Kreisformtests zu kontrollieren
3. ... und die optimalen Aufschaltamplituden zu protokollieren.

Die Adaptionskennlinie wird durch die Ermittlung der im Kapitel "Konventionelle Reibkompensation" genannten Parameter vollständig bestimmt. Allerdings sollten zur Kontrolle erheblich mehr Messwerte ermittelt werden. Insbesondere ist auf eine ausreichende Anzahl von Stützpunkten für hohe Geschwindigkeiten bei kleinen Radien zu achten. Die Kennliniengrößen sind am besten aus einer graphischen Darstellung zu ermitteln.

2. Beschleunigungswerte bestimmen

Bei der Kreisbewegung berechnen sich die axialen Beschleunigungswerte mit dem Radius r und der gefahrenen Geschwindigkeit v nach der Formel

$a = v^2/r$.

Mit Hilfe des Vorschub-Overrideschalters lässt sich die Geschwindigkeit und somit der axiale Beschleunigungswert a auf einfache Weise variieren.

Bei der Eingabe der Beschleunigungswerte a_1, a_2 und a_3 für die Adaptionskennlinie in die zugehörigen Maschinendaten muss die Bedingung $a_1 < a_2 < a_3$ eingehalten werden:

MD32550 $MA_FRICT_COMP_ACCEL1(Adaptions-Beschleunigungswert 1)

bis

MD32570 MA_FRICT_COMP_ACCEL3 (Adaptions-Beschleunigungswert 3)

Bei fehlerhafter Parametrierung der Kennlinie wird die Alarmmeldung 26001 Parametrierfehler "Reibkompensation" ausgegeben.

Beispiel für Einstellung der Kennlinie

1. Ermittlung der vorliegenden Beschleunigung

   Die axiale Beschleunigung errechnet sich beim Drehzahlnulldurchgang einer Kreisfahrt mit der Formel $a = v^2/r$.

   Bei Radius r = 10 mm und einer Kreisgeschwindigkeit von v = 1 m/min (=16,7 mm/s) tritt somit eine Beschleunigung a = 27,8 mm/s² auf.

2. Eingabe der Kennlinienknickpunkte

   Als Kennlinienknickpunkte wurden folgende Beschleunigungen ermittelt:

   a_1 = 1,1 mm/s² ; a_2 = 27, 8 mm/s² ; a_3 = 695 mm/s²

In den Maschinendaten sind dann folgende Werte in dieser Reihenfolge einzugeben:

MD32550 $MA_FRICT_COMP_ACCEL1 [n] (Adaptions-Beschleunigungswert 1) = 0,0011 [m/s²]

MD32560 $MA_FRICT_COMP_ACCEL2 [n] (Adaptions-Beschleunigungswert 2) = 0,0278 [m/s²]

MD32570 $MA_FRICT_COMP_ACCEL3 [n] (Adaptions-Beschleunigungswert 3) = 0,695 [m/s²]

Für die Aufschaltamplituden wurden beispielsweise folgende Werte ermittelt:

MD32520 $FRICT_COMP_CONST_MAX [n] = 30 [mm/min]

MD32530 $FRICT_COMP_CONST_MIN [n] = 10 [mm/min]

Hinweis

Werden für niedrigste Geschwindigkeiten keine befriedigenden Ergebnisse erzielt, so ist ggf. die Rechenfeinheit zu erhöhen, für Linearpositionen im Maschinendatum:

MD10200 $MA_INT_INCR_ PER_MM (Rechenfeinheit für Linearpositionen)

bzw. für Winkelpositionen im Maschinendatum:

MD10210 $MA_INT_INCR_PER_DEG (Rechenfeinheit für Winkelpositionen)

Siehe auch Maschinendatum

MD32580 $MA_FRICT_COMP_INC_FACTOR

(Wichtungsfaktor des Reibkompensationswertes bei kurzen Verfahrbewegungen)

5.6.3 Quadrantenfehler-Kompensation mit neuronalen Netzen - nur 840D sl

5.6.3.1 Grundlagen

Prinzip der QFK

Die Quadrantenfehlerkompensation (QFK reduziert) die beim Reversieren infolge von Richtung, Lose oder Torsion entstehenden Konturfehler. Die Kompensation erfolgt durch die zeitrichtige Aufschaltung eines zusätzlichen Drehzahlsollwertes .Neuronale

Die Intensität des Korrekturimpulses kann bei der konventionellen QFK entsprechend einer Kennlinie in Abhängigkeit von der Beschleunigung eingestellt werden. Diese Kennlinie muss bei der Inbetriebnahme mit Hilfe des Kreisformtests bestimmt und parametriert werden. Das dazu notwendige Vorgehen ist relativ aufwendig und erfordert daher einige Erfahrung.

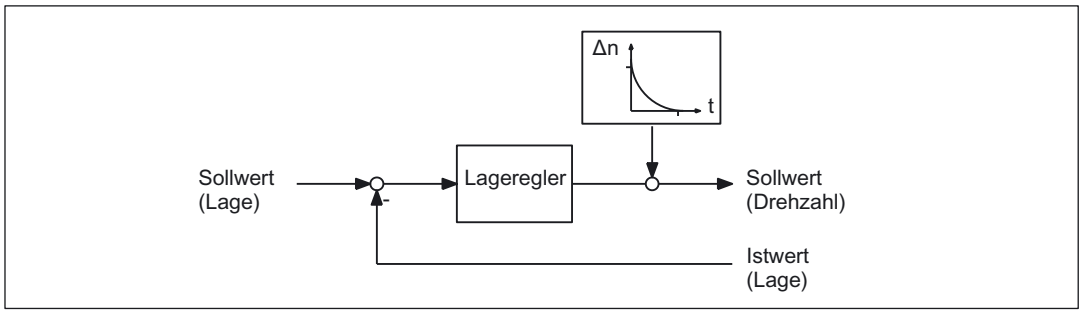


Bild 5-20 Aufschaltung eines zusätzlichen Drehzahlsollwertimpulses

Vorteile der QFK mit neuronalem Netz

Bei SINUMERIK 840D kann der bisher manuell parametrierte Kennlinienbaustein durch ein neuronales Netz ersetzt werden. Dieses bietet folgende Vorteile:

- Zur Vereinfachung der Inbetriebnahme muss die Kompensationskennlinie nicht mehr durch den Inbetriebnehmer von Hand eingestellt werden, sondern sie wird während einer Lernphase **automatisch ermittelt**.
- Bei der manuell parametrierten Reibkompensation wird die Kompensationskennlinie durch einen Polygonzug mit 4 Geradenstücken angenähert. Für eine verbesserte Genauigkeit kann das neuronale Netz den tatsächlichen Verlauf **erheblich besser nachbilden**.

  Die Auflösung der Kennlinie kann an die Genauigkeitsanforderungen angepasst werden, und eine Richtungsabhängigkeit der Korrekturamplitude kann berücksichtigt werden.

  Neben der Korrekturamplitude kann in besonderen Fällen auch die Abklingzeit an die Beschleunigung angepasst werden.

- Das System gestattet jederzeit eine **einfache automatische Nachoptimierung vor Ort**.

Voraussetzung für neuronale QFK

Voraussetzung für die QFK mit neuronalen Netzen ist allerdings, dass die am Werkstück auftretenden Fehler beim Quadrantenübergang durch das Messsystem auch erkannt werden. Dies setzt entweder ein direktes Messsystem, ein indirektes Messsystem mit deutlichen Rückwirkungen der Last auf den Motor (d. h. starre Mechanik, wenig Lose) oder entsprechende Kompensationen voraus. Bei indirekten Messsystemen sollte eventuell vorhandene Lose durch die Losekompensation kompensiert werden.

Lern-/Arbeitsphase

Bei der QFK mit neuronalem Netz sind folgende Phasen zu unterscheiden:

- **Lernphase**

  In der Lernphase wird dem neuronalen Netz ein bestimmtes Verhalten eingeprägt. Dabei wird der Zusammenhang zwischen seinen Eingangs- und Ausgangsgrößen erlernt. Ergebnis ist die gelernte Kompensationskennlinie, die im gepufferten Anwenderspeicher abgelegt ist. Das Ein- und Aus-

  schalten des Lernvorgangs erfolgt vom NC-Teileprogramm mit speziellen Hochsprachenbefehlen.

- **Arbeitsphase**

  In der Arbeitsphase werden anhand der gelernten Kennlinie zusätzliche Drehzahlsollwertimpulse aufgeschaltet. Eine Veränderung der gespeicherten Kennlinie findet in dieser Phase nicht statt.

Die Lernphase kann für mehrere (bis zu 4) Achsen gleichzeitig durchgeführt werden. Weitere Informationen bzgl. Lernen des neuronalen Netzes siehe Kap. "Lernen des neuronalen Netzes".

Lern-, Arbeitsphase sowie die resultierende neuronale QFK wirken rein axial. Es gibt keine wechselseitige Beeinflussung zwischen verschiedenen Achsen.

Kennlinienwerte sichern

Nach dem Lernvorgang müssen die ermittelten Kompensationsdaten (Kennlinienwerte im Anwenderspeicher) einschließlich der Netzwerkparameter (QFK-Systemvariablen) in einer vom Bediener festgelegten Datei gesichert werden. Die Dateien werden standardmäßig "AXn_QEC.INI" benannt.

Kennlinienwerte laden

Diese gesicherten, gelernten Kompensationsdaten können, wie Teileprogramme, wieder direkt in den Anwenderspeicher geladen werden.

Mit Start des Teileprogramms, welches die Tabellen enthält, werden die Kompensationswerte in den NC-Anwenderspeicher übertragen. Diese Kennlinienwerte werden erst nach Freigabe der Kompensation wirksam.

Kennlinienwerte können **bei aktiver** Kompensation **nicht** geschrieben werden (es muss Maschinendatum

MD32500 $MA_FRICT_COMP_ENABLE = 0 (Reibkompensation aktiv)

sein und wirksam sein).

Bei QFK:

Die QFK ist mit folgendem Maschinendatum freizugeben (und wirksam zu machen):

MD32500 $MA_FRICT_COMP_ENABLE = 1 (QFK aktiv)

Empfohlene Vorgehensweise bei Inbetriebnahme

Wie bereits erwähnt, adaptiert das in die Steuerung integrierte neuronale Netz während der Lernphase automatisch die optimalen Kompensationsdaten.

Dabei müssen die gewünschten Achsen Reversiervorgänge mit abschnittsweise konstanten Beschleunigungswerten ausführen. Vor Aktivierung der Lernphase müssen die Parameter des neuronalen Netzes (QFK-Systemvariablen) entsprechend den Anforderungen vorbesetzt werden.

Um eine möglichst einfache Inbetriebnahme zu ermöglichen, werden dafür NC-Programme als Musterbeispiele bereitgestellt.

Wie im Kapitel "Inbetriebnahme der neuronalen QFK" dargestellt, sollte der Inbetriebnehmer zunächst mit Hilfe dieser Musterbeispiele und den empfohlenen QFK-Parameterwerten die Kennlinie für die gewünschten Achsen erlernen und mit Hilfe des Kreisformtests (siehe Kapitel "Kreisformtest") die erzielte Konturgenauigkeit kontrollieren. Falls die Ergebnisse den Anforderungen noch nicht genügen, ist eine Nachoptimierung mit entsprechend veränderten Parametern (siehe Kapitel "Parametrierung der neuronalen QFK", "Lernen des neuronalen Netzes" und "Weitere Optimierungs- und Eingriffsmöglichkeiten") vorzunehmen (sog. Nachlernen).

5.6.3.2 Parametrierung der neuronalen QFK

Maschinendaten

Die grundlegenden Projektierungsdaten für die neuronale QFK werden als Maschinendaten hinterlegt.

- MD32490 $MA_FRICT_COMP_MODE (Art der Reibkompensation)

  (2 = neuronale QFK)
- MD32500 $MA_FRICT_COMP_ENABLE

  (Reibkompensation aktiv)
- MD32580 $MA_FRICT_COMP_INC_FACTOR

  (Wichtungsfaktor Reibkompensationswert b. kurzen Verfahrbew.)
- MD38010 $MA_MM_QEC_MAX_POINTS

  (Maximale Anzahl der Korrekturwerte für QFK mit neuronalen Netzen)

Mit diesen Maschinendaten wird die neuronale QFK aktiviert sowie der Speicherplatz im gepufferten RAM reserviert. Die Vorgehensweise und Belegung ist im Kap. "Inbetriebnahme der neuronalen QFK" beschrieben.

Alle anderen Daten werden mit Hilfe von Systemvariablen vorgegeben.

QFK-Systemvariablen

Die Daten zur Parametrierung des neuronalen Netzes werden als Systemvariablen definiert, die von einem NC-Programm gelesen und geschrieben werden können. Folgende Systemvariablen dienen zur Parametrierung des neuronalen Netzes:

- $AA_QEC_COARSE_STEPS

  "Grobquantisierung der Kennlinie"

  Dieser Parameter bestimmt die Grobquantisierung der Eingangsgröße und damit die Auflösung der Kennlinie. Je größer die Wert gewählt wird, desto höher ist der benötigte Speicherbedarf und desto größer ist die Zeitdauer für die Lernphase. Weitere Informationen siehe am Ende dieses Kapitels.

  Wertebereich: 1 bis 1024; Empfohlener Wert: 49

- $AA_QEC_FINE_STEPS

  "Feinquantisierung der Kennlinie"

  Dieser Parameter bestimmt die Feinquantisierung der Eingangsgröße und damit die Auflösung der Kennlinie. Je größer der Wert gewählt wird, desto höher ist der benötigte Speicherbedarf.

  Wertebereich: 1 ... 16; Empfohlener Wert: 8

- $AA_QEC_DIRECTIONAL

  "Richtungsabhängigkeit"

  Mit diesem Parameter wird festgelegt, ob die Kompensation richtungsabhängig aufgeschaltet werden soll. Bei Aktivierung wird jeweils getrennt eine Kennlinie für jede Richtung der Beschleunigung ermittelt und abgelegt. Da hierbei zwei Kennlinien verwendet werden, muss der doppelte Speicherbedarf im gepufferten Anwenderspeicher reserviert werden.

  Wertebereich: `TRUE/FALSE`; Empfohlener Wert: `FALSE`

- $AA_QEC_LEARNING_RATE

  "Lernrate für aktive Lernphase"

  Mit der Lernrate kann bestimmt werden, wie schnell in der aktiven Lernphase der neuronalen QFK die optimale Kennlinie gelernt wird. Dieser Wert ist ein Wichtungsfaktor, mit dem festgelegt wird, wie intensiv sich Abweichungen auf die Aufschaltamplitude auswirken. Bei größeren Werten (>100%) wird die Kennlinie schneller gelernt, wobei zu beachten ist, dass zu hohe Lernratenwerte (Wichtungsfaktoren) zur Instabilität führen können (Zweipunktverhalten).

  Für Nachlernvorgänge während des normalen Betriebes ist auf jeden Fall eine kleine Lernrate zu empfehlen (< 50%), da sonst bei jeder kleinsten Störung im Drehzahlnulldurchgang die Kennlinie verändert wird.

  Wertebereich: > 0%; ≤ 500%; Empfohlener Wert: 50%

- $AA_QEC_ACCEL_1/_2/_3

  "Beschleunigungsgrenzwerte für die Kennlinienbereiche 1/2/3"

  Die Beschleunigungskennlinie ist in drei Bereiche aufgeteilt. In jedem Bereich gilt eine unterschiedliche Quantisierung der Beschleunigungsstufen. Im Bereich kleiner Beschleunigungen ist eine besonders hohe Auflösung der Kennlinie erforderlich, um die dort stark differierenden Korrekturwerte nachzubilden. Daher erfolgt die Quantisierung der Eingangsgröße umso feiner, je kleiner die Beschleunigung ist.

  Empfohlene Werte für

  – $AA_QEC_ACCEL_1: 20 mm/s² (= 2% von $AA_QEC_ACCEL_3)
  – $AA_QEC_ACCEL_2: 600 mm/s² (= 60% von $AA_QEC_ACCEL_3)
  – $AA_QEC_ACCEL_3: 1000 mm/s²(Maximalbeschleunigung des Arbeitsbereiches)

  Der Wert des Parameters $AA_QEC_ACCEL_3 ist entsprechend den Anforderungen einzugeben; d. h. nur in dem Arbeitsbereich mit dieser maximalen Beschleunigung arbeitet und lernt das neuronale Netz optimal. Wird eine höhere Beschleunigung als der parametrierte Arbeitsbereich erkannt, so wird die Aufschaltamplitude verwendet, die bei der festgelegten Maximalbeschleunigung des Arbeitsbereiches ermittelt wurde. Bei hohen Beschleunigungen ist dieser Aufschaltwert relativ konstant.

  Eine Änderung der empfohlenen Werte sollte nur vorgenommen werden, wenn die Kompensation in diesen Beschleunigungsbereichen unzureichend ist. Weitere Informationen dazu siehe Kap. "Weitere Optimierungs- und Eingriffsmöglichkeiten".

- $AA_QEC_TIME_1

  "Zeitkonstante für die neuronale QFK (Abklingzeit)"

  Damit wird die Abklingzeit des Korrektursollwertimpulses eingestellt, wenn ohne dessen Adaption gearbeitet wird.

  Die optimale Abklingzeit ist in einem Arbeitspunkt im mittleren Beschleunigungsbereich manuell mit Hilfe des Kreisformtests zu ermitteln. Das Vorgehen ist bei der konventionellen Reibkompensation (Kap. "Inbetriebnahme der konventionellen Reibkompensation")beschrieben (analog wie bei Maschinendatum

  MD32540 $FRICT_COMP_TIME (Reibkompensations-Zeitkonstante)).

  Mit dem empfohlenen Wert (15ms) können in der Regel bereits gute Ergebnisse erzielt werden.

  Wertebereich: ≥ 0; Empfohlener Wert: 0,015s

  Falls die Adaption der Abklingzeit wirksam ist, so bestimmt $AA_QEC_TIME_1 die Filterzeitkonstante in der Mitte des Arbeitsbereichs (d. h. bei 0,5 * $AA_QEC_ACCEL_3).

- **$AA_QEC_TIME_2**

  "Kompensationszeitkonstante zur Adaption der Abklingzeit des Korrekturwertes"

  Bei einem Wert von Null oder kleiner/gleich $AA_QEC_TIME_1 findet keine Adaption statt.

  Die Abklingzeit ist üblicherweise im gesamten Arbeitsbereich konstant. In seltenen Fällen kann jedoch im Bereich kleinster Beschleunigungen eine Anhebung der Abklingzeit von Vorteil sein bzw. eine Absenkung bei höheren Beschleunigungen. Weitere Informationen dazu siehe Kap. "Weitere Optimierungs- und Eingriffsmöglichkeiten".

  Wertebereich: ≥ 0; Empfohlener Wert: 0,015s (= identisch mit $AA_QEC_TIME_1)

- **$AA_QEC_MEAS_TIME_1/_2/_3**

  "Messdauer zur Ermittlung des Fehlerkriteriums im Beschleunigungsbereich 1/2/3"

  Die Messzeit wird gestartet, sobald das Kriterium zur Aufschaltung des Kompensationswertes erfüllt ist (d. h. die Sollgeschwindigkeit wechselt das Vorzeichen). Das Ende der Messzeit wird durch die vorgegebenen Parameterwerte festgelegt.

  Für die drei Kennlinienbereiche sind i. a. unterschiedliche Messzeiten erforderlich.

  Empfohlene Werte für

  – $AA_QEC_MEAS_TIME_1: 0,090s (= 6 *$AA_QEC_TIME_1)
  – $AA_QEC_MEAS_TIME_2: 0,045s (= 3 * $AA_QEC_TIME_1)
  – $AA_QEC_MEAS_TIME_3: 0,030s (= 2 * $AA_QEC_TIME_1)

  Eine Änderung der empfohlenen Werte sollte nur vorgenommen werden, wenn die Kompensation in diesen Beschleunigungsbereichen unzureichend ist oder falls $AA_QEC_TIME_1 verändert wird. Weitere Informationen dazu siehe Kap. "Weitere Optimierungs- und Eingriffsmöglichkeiten".

Parameterübernahme

Die QFK-Systemvariablen werden nach Starten des NC-Programms im gepufferten Anwenderspeicher abgelegt und bleiben bis zum Löschen oder Neuformatieren des gepufferten Speichers, Neu- oder Nachlernen oder Schreiben durch NC-Programm erhalten.

Vor dem Aufruf des Lernzyklus müssen alle Systemvariablen mit den für den Lernvorgang gültigen Werten besetzt werden. Dies kann z. B. in einem Unterprogramm erfolgen. Nach Ablauf dieses NC-Programms und anschließendem Reset sind die QFK-Daten wirksam.

Kennlinienwerte

Die beim Lernvorgang ermittelten Kennlinienwerte werden als Systemvariablen in den reservierten Anwenderspeicher abgelegt.

Format: **$AA_QEC[n]** Bereich von n: 0 bis 1024

Diese Werte beschreiben die gelernte Kennlinie in steuerungsinternen Formaten und dürfen daher **nicht geändert** werden!

Quantisierung der Kennlinie

Die Festlegung der Quantisierung und damit die Auflösung der Kennlinie wird über die beiden Größen **Feinquantisierung** ($AA_QEC_FINE_STEPS)und **Grobquantisierung** ($AA_QEC_COARSE_STEPS)vorgenommen. Je feiner die Auflösung gewählt wird, umso höher ist der benötigte Speicherbedarf und umso größer ist die Zeitdauer für die Lernphase.

Die Anzahl der benötigten Speicherplätze bzw. die Gesamtzahl der Quantisierungsintervalle errechnet sich nach der Formel:

Anzahl Speicherplatz = $AA_QEC_FINE_STEPS * ($AA_QEC_COARSE_STEPS+1)

Pro Achse sind maximal 1025 Speicherplätze reservierbar. Damit wird eine ausreichende Auflösung auch für hohe Genauigkeitsanforderungen ermöglicht.

Anhand der folgenden 3 Bilder wird die Bedeutung der Kennwerte für Grob- und Feinquantisierung sowie deren Einfluss auf die Lerndauer in Abhängigkeit vom Parameter "Ausführliches Lernen aktiv j/n" erläutert. Zum besseren Verständnis werden hierbei drei Fälle unterschieden.

Fall 1:

Grobquantisierung > 1; Feinquantisierung = 1 (Sonderfall; i.d.R. liegt die Feinquantisierung im Bereich von 8):

Die Stützpunkte der Kennlinie werden hierbei ausschließlich durch die Grobquantisierung festgelegt (siehe folgendes Bild).

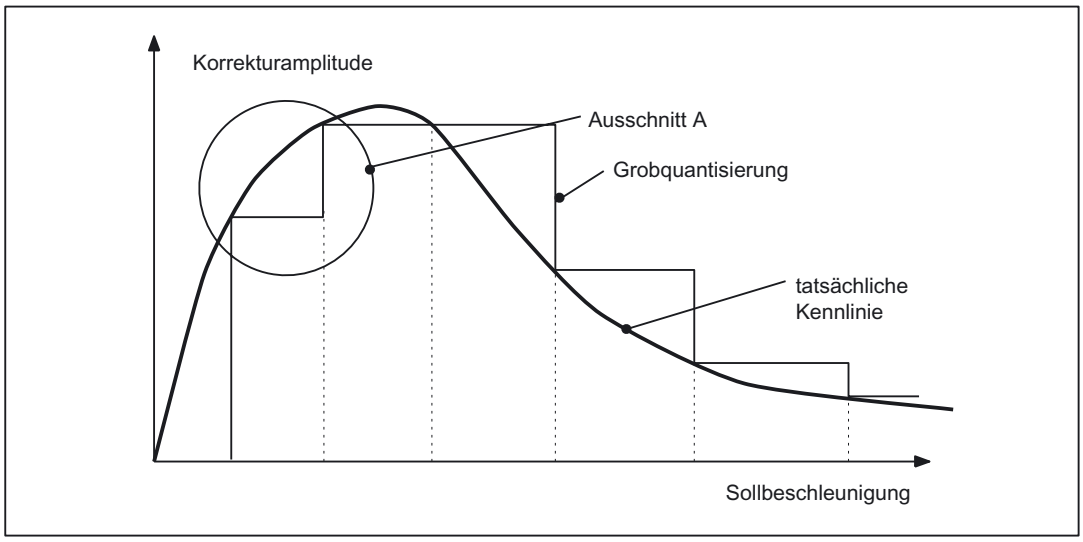


Bild 5-21 Grobquantisierung der Kennlinie

Fall 2:

Grobquantisierung > 1; Feinquantisierung > 1; "Ausführliches Lernen" abgewählt (diese Einstellung entspricht dem Standardfall):

In diesem Fall wird bei der Feinquantisierung zwischen den durch die Grobquantisierung festgelegten Stützpunkten diskret linear interpoliert.

Die Lernzeitdauer ist hier identisch wie bei Fall 1, da nur an den Stützpunkten der Grobquantisierung gelernt wird.

Im folgenden Bild ist die Wirkung der Feinquantisierung für einen Kennlinienausschnitt innerhalb einer Grobquantisierung dargestellt (siehe Ausschnitt A des vorhergehenden Bildes).

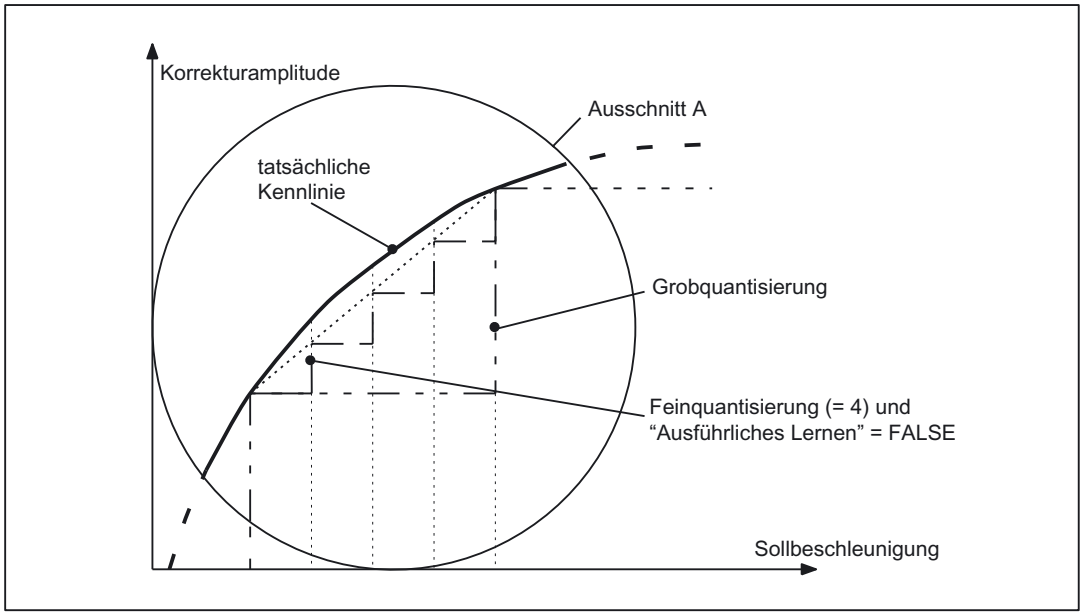


Bild 5-22 Wirkung der Feinquantisierung bei "Ausführliches Lernen" nicht aktiv

Fall 3:

Grobquantisierung > 1; Feinquantisierung > 1; "Ausführliches Lernen aktiv" angewählt (Anwendung wird nur bei sehr hohen Genauigkeitsanforderungen empfohlen):

Beim "Ausführlichen Lernen" wird sowohl an den Stützpunkten der Grobquantisierung als auch der Feinquantisierung gelernt.

Die Lernzeitdauer ist hier somit wesentlich höher.

Im folgenden Bild ist ein stark schwankender Kennlinienverlauf dargestellt, bei dem die Wirkung bei An- bzw. Abwahl des "Ausführlichen Lernens" deutlich erkennbar ist.

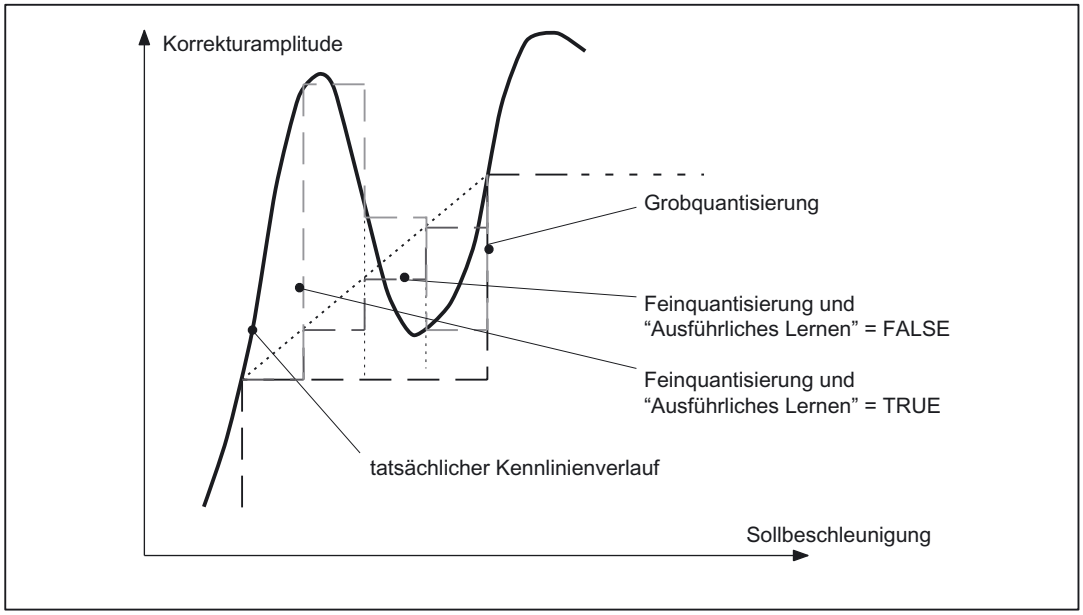


Bild 5-23 Wirkung der Feinquantisierung bei "Ausführliches Lernen" = aktiv

5.6.3.3 Lernen des neuronalen Netzes

Ablauf der Lernphase

In der Lernphase wird dem neuronalen Netz ein bestimmtes Verhalten eingeprägt. Dabei wird der Zusammenhang zwischen seinen Eingangs- und Ausgangsgrößen erlernt.

Die Steuerung des Lernvorganges erfolgt komplett über NC-Programme und gliedert sich in folgende Bereiche:

1. Vorbesetzung der QFK-Systemvariablen für den Lernvorgang
2. QFK-Systemvariablen aktivieren (durch Starten des NC-Programms)
3. Parametrierung des Lernzyklusses
4. Lernzyklus starten

Ergebnis ist die gelernte Kompensationskennlinie, die im gepufferten Anwenderspeicher abgelegt ist.

Die dabei erzielten Ergebnisse sind mit Hilfe des Kreisformtests (Kapitel "Kreisformtest") zu kontrollieren.

Lern-NC-Programme

Um die Inbetriebnahme der QFK mit neuronalen Netzen für den Inbetriebnehmer möglichst einfach zu machen, werden NC-Programme als Musterbeispiele für die Lernbewegungen sowie für die Belegung der QFK-Systemvariablen (empfohlene Werte) zur Verfügung gestellt.

Dabei handelt es sich um folgende Muster-NC-Programme:

- QECLRNP.SPF

  Lernen mit POLY-Stützen (Option "POLY" erforderlich)

- QECLRNC.SPF

  Lernen mit Kreisen

- QECDAT.MPF

  Muster-NC-Programm für Belegen der Systemvariablen und für Parametrierung des Lernzyklus

- QECSTART.MPF

  Muster-NC-Programm, das den Lernzyklus aufruft

Diese NC-Programme sind auf der Diskette des PLC-Grundprogramms für die SINUMERIK 840D enthalten.

Die Möglichkeit, den Lernvorgang ausschließlich über NC-Programme durchführen zu können, bietet folgende Vorteile:

- Das Lernen kann vollautomatisch ohne Bedienereingriffe erfolgen.Dies ist sehr vorteilhaft bei Serieninbetriebnahmen, wenn die Lernparameter für einen Maschinentyp optimal ermittelt wurden und nur die Kennlinie dann für jede einzelne Maschine ermittelt oder nachgelernt werden soll.
- Das Lernen kann für mehrere Achsen (bis zu 4) gleichzeitig erfolgen. Dadurch verkürzt sich die Lernphase für die Maschine erheblich.
- Die Verfahrbewegungen können sehr leicht den speziellen Anforderungen angepaßt werden.

Hinweis

Der Kreisformtest ist bei HMI Advanced integriert. Bei HMI Embedded ist das Inbetriebnahme-Tool zu verwenden.

Lernbewegung

Die beim Lernen erforderlichen Verfahrbewegungen der Achse werden mit Hilfe eines NC-Programms generiert. Die Lernbewegung des Muster-Lernzyklusses besteht jeweils aus Gruppen von NC-Sätzen mit parabolischen Bewegungen (damit nach dem Nulldurchgang der Solldrehzahl mit möglichst konstanter Beschleunigung gefahren wird; siehe folgendes Bild), in denen die Achsen mit abschnittsweise konstanter Beschleunigung hin- und herschwingen. Dabei nimmt die Beschleunigung von Gruppe zu Gruppe ab. Im folgenden Bild bilden jeweils die NC-Sätze 2 bis 3, 5 bis 6, 8 bis 9 eine solche Gruppe; in den Sätzen 1, 4, 7 und 10 finden die Übergangsbewegungen auf kleinere Beschleunigungswerte statt.

Hinweis

Damit die Lernparameter entsprechend den Vorbesetzungen wirken, muss der Vorschub-Korrekturschalter während der Lernphase auf 100% eingestellt werden.

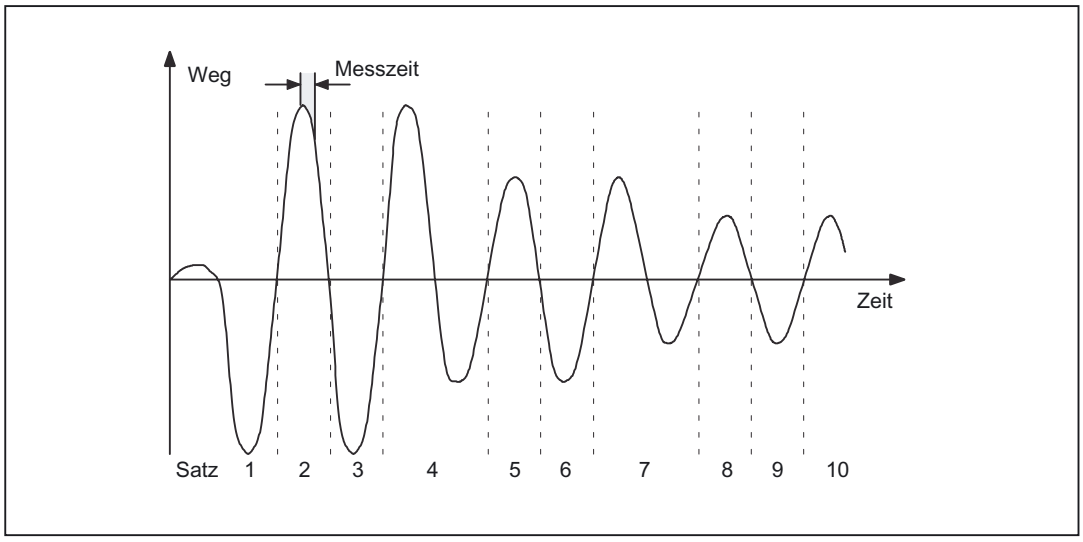


Bild 5-24 Typische Verfahrbewegung einer Achse beim Lernen der QFK-Kennlinie

Systemvariable belegen

Vor dem Aufruf des Lernzyklusses müssen alle QFK-Systemvariablen mit den für den Lernvorgang gültigen Werten besetzt werden. Dabei sind die im Muster-NC-Programm empfohlenen Werte zu kontrollieren und ggf. abzuändern (siehe Kapitel "Parametrierung der neuronalen QFK").

Lernen ein/aus

Desweiteren wird im Muster-NC-Programm der eigentliche Lernvorgang des neuronalen Netzes aktiviert. Dies erfolgt mit folgendem Hochsprachenbefehl:

QECLRNON(Achsname 1, ... 4) Lernen ein (für angegebene Achsen)

Nur während dieser Phase werden die Kennlinien verändert.

Nachdem die Lernbewegungen der gewünschten Achsen abgeschlossen ist, wird der Lernvorgang wieder für alle Achsen gleichzeitig ausgeschaltet. Dies erfolgt mit dem Hochsprachenbefehl

QECLRNOF Lernen aus (gleichzeitig für alle Achsen)

Nach Power-On-Reset, Programmende (M02/M30) oder Bedientafelfront-Reset ist das Lernen ebenfalls ausgeschaltet.

Der aktuelle Status bzgl. "Lernen Ein/Aus" wird im Servicebild "Achsen" mit "QFK-Lernen aktiv" (1 = aktiv; 0 = nicht aktiv) angezeigt.

Lernzyklus-Aufruf

Vom Muster-NC-Programm wird nach der Aktivierung des Lernens der Lernzyklus mit folgenden Eingangsparametern aufgerufen:

- **Anzahl der Achsen**, für die gelernt werden soll (maximal 4).

  Voraussetzung:

  Falls mehr als eine Achse gleichzeitig lernen soll, müssen alle QFK-Systemvariablen der beteiligten Achsen die gleichen Werte aufweisen. Diese Werte werden überwacht, und bei Ungleichheit wird eine Fehlermeldung ausgelöst.

- **Achsnamen der lernenden Achsen**

  Ausgangsnummer (für alle Achsen gleich) Wert immer 0 (Sollwertzweig)

  Lernmodus (Neulernen = 0; Nachlernen = 1)0: Neulernen aktiv. Damit werden vor dem Start des Lernens alle Werte des Netzwerkes mit dem Wert 0 vorbesetzt.

  1: Nachlernen aktiv. Das Lernen wird mit den bereits gelernten Werten fortgesetzt in der festgelegten Schrittweite.

- **Ausführliches Lernen aktiv** ja/nein (`TRUE`/`FALSE`)

  FALSE: Das ausführliche Lernen ist nicht aktiv. Die Kennlinie wird somit in der Schrittweite der Grobquantisierung der Beschleunigung gelernt.

  TRUE: Das "ausführliche Lernen" ist aktiv. Die Kennlinie wird in der Schrittweite der Feinquantisierung der Beschleunigung gelernt. D. h. bei einer Feinquantisierung von 10 Schritten pro Grobschritt dauert die Ermittlung der Kennlinie auch zehnmal länger. Daher sollte dieser Parameter nur für extrem hohe Genauigkeitsanforderungen gesetzt werden.

  Hinweis

  Wenn das "ausführliche Lernen" angewählt ist, kann und sollte wegen Reduzierung der Lerndauer die Anzahl der Lerndurchläufe reduziert werden (empfohlener Bereich: zwischen 1 bis 5).

- **Anzahl der Lerndurchläufe**

  Defaultwert = 15; Bereich > 0

  Die Wirkungsweise dieses Parameters ist davon abhängig, ob "ausführliches Lernen aktiv" gesetzt ist oder nicht.

  a) Ausführliches Lernen nicht aktiv (= `FALSE`):

  Damit wird die Anzahl der Testbewegungen (Hin- und Herbewegungen) bei jeder Beschleunigungsstufe festgelegt. Je höher die Anzahl, desto genauer wird gelernt, aber umso länger ist die Lerndauer.

  Bei richtungsabhängiger Kompensation ($AA_QEC_DIRECTION = TRUE) wird die parametrierte Anzahl der Testbewegungen für jede Richtung generiert.

  b) Ausführliches Lernen aktiv (= `TRUE`):

  Damit wird die Anzahl der kompletten Durchläufe von Maximal- zu Minimalbeschleunigung und umgekehrt mit der Feinschrittweite aktiviert.D. h. bei einem Wert von 1 werden einmal alle Beschleunigungsstufen beginnend beim Maximalwert durchlaufen. Für jede Beschleunigungsstufe werden zwei Testbewegungen generiert, sofern keine Richtungsabhängigkeit besteht ($AA_QEC_DIRECTION= FALSE), ansonsten vier Testbewegungen pro Beschleunigungsstufe.

  Eine Reduzierung der "Anzahl der Lerndurchläufe" kann insbesondere dann vorgenommen werden, wenn bereits Datensätze für den betreffenden Maschinentyp vorliegen (Serienmaschinen) und davon ausgehend nur noch eine Nachoptimierung durchgeführt werden soll.

- **Abschnittsweises Lernen aktiv** ja/nein (`TRUE`/`FALSE`)

  Das "abschnittsweise Lernen" in bestimmten Beschleunigungsbereichen ist insbesondere beim "ausführlichen Lernen" z. B. in technologisch wichtigen Bereichen der Maschine interessant. Durch entsprechende Bereichseingrenzung kann damit die Lerndauer reduziert werden.

  Defaultwert = `FALSE`

- **Bereichsgrenzen für "abschnittsweises Lernen"** (minimale Beschleunigung, maximale Beschleunigung); nur bei "abschnittsweises Lernen aktiv" relevant.

  Defaultwert = 0; Format: mm/s^2

- **Zeitdauer für eine Testbewegung** (hin und her)

  Defaultwert = 0.5; Format: s (Sekunden) (entspricht einer Frequenz von 2 Hz)

Voraussetzung

Die neuronale QFK erfordert in der Lernphase eine Drehzahlvorsteuerung und keine Ruckbegrenzung (BRISK):

(MD32620 $MA_FFW_MODE=1; FFWON (Vorsteuerungsart))

Die Vorsteuerung muss daher richtig parametriert und optimiert sein. Beim Starten des Lernvorgangs wird geprüft, ob die Drehzahlvorsteuerung aktiviert ist. Falls dies nicht der Fall ist, wird der Lernvorgang abgebrochen und eine Fehlermeldung generiert.

5.6.3.4 Inbetriebnahme der neuronalen QFK

Allgemeines

Im Folgenden wird der Ablauf der Inbetriebnahme der QFK mit neuronalen Netzen kurz beschrieben. Wie bereits mehrfach erwähnt, werden die Kompensationskennlinien während der Lernphase automatisch ermittelt.

Dabei müssen die gewünschten Achsen Reversiervorgänge mit abschnittsweise konstanten Beschleunigungswerten ausführen. Desweiteren müssen die QFK-Systemvariablen zur Parametrierung des neuronalen Netzes entsprechend den Anforderungen vorbesetzt werden.

Um eine möglichst einfache Inbetriebnahme zu ermöglichen, werden dafür NC-Programme als Musterbeispiele bereitgestellt (siehe Kapitel "Lernen des neuronalen Netzes").

Beim Lernen wird zwischen "Neulernen" (insbesondere für Erstinbetriebnahme) und "Nachlernen" (insbesondere für Nachoptimierung bereits vorgelernter Kennlinien) unterschieden. Nachfolgend sind die Abläufe für "Neulernen" bzw. "Nachlernen" beschrieben.

Falls die Kompensationskennlinien für die Maschine zum ersten Mal gelernt werden sollen, empfiehlt es sich, die im Kapitel "Lernen des neuronalen Netzes" angegebenen Muster-NC-Programme zu verwenden.

Ablauf "Neulernen"

"Neulernen" -> Zyklusparameter "Lernmodus" = 0

1. a) QFK mit neuronalen Netzen für die gewünschten Achsen aktivieren mit Maschinendatum:

   MD32490 $MA_FRICT_COMP_MODE = 2 (Art der Reibkompensation)

   Hinweis

   QFK mit neuronalen Netzen ist eine Option!

   b) Speicherplatz für die Kompensationspunkte reservieren mit Maschinendatum

   MD38010 $MA_MM_QEC_MAX_POINTS

   (Anzahl der Werte für Quadrantenfehlerkomp. mit neuro. Netz)

   Falls die benötigte Anzahl noch nicht bekannt ist, sollte hier zunächst großzügig reserviert werden (siehe auch Punkt 12).

   c) Drehzahlvorsteuerung parametrieren und optimieren (wird für Lernphase benötigt)

   d) Hardware-Reset geben (wegen Neueinrichtung des gepufferten Anwenderspeichers)

2. QFK-Systemvariablen aktivieren:

   Muster-NC-Programm QECDAT.MPF zur Belegung der QFK-Systemvariablen für alle betroffenen Achsen an die gewünschten Werte anpassen (ggf. zunächst die empfohlenen Werte verwenden) und NC-Programm starten. Bei Fehlermeldungen Werte korrigieren und NC-Programm erneut starten.

3. NC-Programm erstellen, das die Maschinenachsen an die gewünschten Positionen fährt und den Muster-Lernzyklus QECLRN.SPF parametriert und aufruft (wie im Beispielprogramm QECSTART.MPF). Der Vorschubkorrekturschalter sollte während der

Lernphase auf 100% eingestellt werden, damit die Parameter entsprechend den Vorbesetzungen wirken.

4. Lernphase aktivieren durch Starten dieses NC-Programms. Die Kompensationskennlinie wird für alle parametrierten Achsen gleichzeitig gelernt. Die Lerndauer ist abhängig von den vorgegebenen Lernparametern. Bei Anwendung der Standardwerte beträgt diese einige Minuten. Im Servicebild "Achse" kann in der Anzeige "QFK-Lernen aktiv" der Status für die betroffenen Achsen beobachtet werden.
5. Aktivierung der Aufschaltung der Kompensationswerte für die gewünschten Achsen mit Maschinendatum:

   MD32500 $MA_FRICT_COMP_ENABLE = 1(Reibkompensation aktiv).
6. Trace für Kreisformtest über Menü "Kreisformtest-Messung"parametrieren (mit HMI Advanced oder Inbetriebnahme-Tool). Parameterwerte für Muster-NC-Programm:

   Radius[mm]:

   Vorschub[mm/min].

   Anschließend ist die Messfunktion mit dem vertikalen Softkey "Start" freizugeben.
7. NC-Programm mit Testbewegung (Kreis) starten. Die Lageistwerte während der Kreisbewegung werden aufgezeichnet und im passivem Dateisystem hinterlegt. Nach Beendigung der Datenaufzeichnung wird die aufgezeichnete Kontur graphisch dargestellt.
8. Die Quadrantenübergänge bei der aufgezeichneten Kontur kontrollieren.
9. Je nach Ergebnis sind ggf. die Punkte 2, 4, 7, 8 und 9 zu wiederholen. Ggf. sind zuvor einzelne QFK-Systemvariablen abzuändern (siehe auch Kap. "Lernen des neuronalen Netzes").
10. Sobald die erreichte Konturgenauigkeit den Anforderungen entspricht, sind die Kompensationskennlinien zu sichern (siehe Kapitel "Lernen des neuronalen Netzes").
11. Ggf. kann der bisher für die Kompensationswerte reservierte Speicherbereich auf den tatsächlich benötigten Speicherbedarf reduziert werden.

ACHTUNG
Bei Änderung des folgenden Maschinendatums wird bei Systemhochlauf automatisch der gepufferte NC-Anwenderspeicher neu eingerichtet: MD38010 $MA_MM_QEC_MAX_POINTS (Anzahl der Werte für Quadrantenfehlerkomp. mit neuro. Netz) Dabei werden alle Anwenderdaten des batteriegepufferten Anwenderspeichers gelöscht! Daher sind diese zuvor unbedingt zu sichern. Nach POWER ON der Steuerung sind u. a. die gesicherten Kennlinien wieder zu laden.

Ablauf "Nachlernen"

"Nachlernen" -> Zyklusparameter "Lernmodus" = 1

Mit dem "Nachlernen" ist eine einfache automatische Nachoptimierung der bereits gelernten Kennlinien möglich. Dabei wird auf den bisherigen im Anwenderspeicher befindlichen Werten aufgesetzt.

Beim "Nachlernen" sind für die Lernphase die für die Maschine angepaßten Lern-NC-Programme (z. B. vom "Neulernen") zu verwenden. In der Regel können dafür die bisherigen Werte der QFK-Systemvariablen weiter verwendet werden. Vor Aufruf des Lernzyklusses ist der Parameter "Lernmodus" auf 1 (für Nachlernen) zu setzen. Desweiteren kann u.U. die "Anzahl der Lerndurchläufe" reduziert werden.

Ablaufschritte für Nachlernen

Nachfolgend sind die Ablaufschritte für Nachlernen beschrieben.

1. Falls im Anwenderspeicher (RAM) noch keine Kennlinienwerte enthalten sind (z. B. Inbetriebnahme einer Serienmaschine), ist der voroptimierte Datensatz zu laden (siehe Kapitel "Grundlagen").
2. NC-Programm anpassen, das die Maschinenachsen an die gewünschten Positionen fährt und den Lernzyklus parametriert und aufruft. Die Parameter des Lernzyklus (z. B. QECLRN.SPF) sind ggf. für das "Nachlernen" abzuändern.
   - "Lernmodus" = 1 setzen
   - "Anzahl der Lerndurchläufe" ggf. reduzieren (z. B. auf 5)
   - "Abschnittsweises Lernen" ggf. aktiveren sowie die zugehörigen Bereichsgrenzen festlegen
3. Lernphase aktivieren durch Starten dieses NC-Programms. Die Kompensationskennlinie wird für alle parametrierten Achsen gleichzeitig gelernt.
4. Trace für Kreisformtest über Menü "Kreisformtest-Messung"parametrieren (mit HMI Advanced oder Inbetriebnahme-Tool). Anschließend ist die Messfunktion mit dem vertikalen Softkey "Start" freizugeben.
5. NC-Programm mit Testbewegung für Kreisformtest starten. Die Lageistwerte während der Kreisbewegung werden aufgezeichnet und im passivem Dateisystem hinterlegt. Nach Beendigung der Datenaufzeichnung wird die aufgezeichnete Kontur auf der Bedienoberfläche dargestellt.
6. Die Quadrantenübergänge bei der aufgezeichneten Kontur kontrollieren.
7. Je nach Ergebnis sind ggf. die Punkte 3, 4, 5 und 6 zu wiederholen. Ggf. sind zuvor einzelne QFK-Systemvariablen abzuändern (siehe Kapitel "Weitere Optimierungs- und Eingriffsmöglichkeiten").
8. Sobald die erreichte Konturgenauigkeit den Anforderungen entspricht, ist die nachgelernte Kompensationskennlinie zu sichern (siehe Kapitel "Grundlagen").

5.6.3.5 Weitere Optimierungs- und Eingriffsmöglichkeiten

Optimierungsmöglichkeiten

Falls die Ergebnisse des Kreisformtests nicht den gewünschten Genauigkeitsanforderungen entsprechen, können durch gezielte Veränderungen der QFK-Systemvariablen weitere Verbesserungen erzielt werden. Nachfolgend werden einige Möglichkeiten zur Optimierung der neuronalen QFK erläutert.

Ändern der Grob- und Feinquantisierung

Die Quantisierung der Eingangsgröße erfolgt durch die beiden Größen "Grobquantisierung" und "Feinquantisierung".

Ein großer Wert bei der Feinquantisierung bewirkt, dass für benachbarte Intervalle der Eingangsgröße ein "ähnlicher" Ausgangswert ermittelt wird. Damit können z. B. Messfehler, die nur bei einer bestimmten Beschleunigung auftreten, ausgemittelt werden.

Mit einem kleinen Feinquantisierungswert werden hingegen stark schwankende Kennlinienverläufe besser nachgebildet.

Für die neuronale Reibkompensation sollte die größere Fehlertoleranz durch eine hohe Feinquantisierung ($AA_QEC_FINE_STEPS im Bereich von etwa 5 bis 10) ausgenutzt werden.

Richtungsabhängige Kompensation

Die richtungsabhängige Reibkompensation ist dann erforderlich, wenn bei richtungsunabhängiger Aufschaltung der Kompensationswerte die Korrektur an gegenüberliegenden Quadranten nicht gleichwertig gelingt (siehe folgendes Bild).

Die richtungsabhängige Aufschaltung wird über die Systemvariable $AA_QEC_DIRECTIONAL = TRUE aktiviert.

Es sind dabei allerdings folgende Punkte zu beachten:

- Da für jede Beschleunigungsrichtung jeweils eine Kennlinie gelernt und abgelegt wird, verdoppelt sich der Speicherbedarf im gepufferten Anwenderspeicher. Das folgende Maschinendatum ist entsprechend anzupassen.

  MD38010 $MA_MM_QEC_MAX_POINTS

  (Anzahl der Werte für Quadrantenfehlerkomp. mit neuro. Netz)

- Die Anzahl der Lerndurchläufe sollte erhöht werden, da nur jeder zweite Lernvorgang wieder an derselben Stelle stattfindet.
- Bei gleicher Kennlinien-Auflösung erhöht sich die Inbetriebnahmezeit.

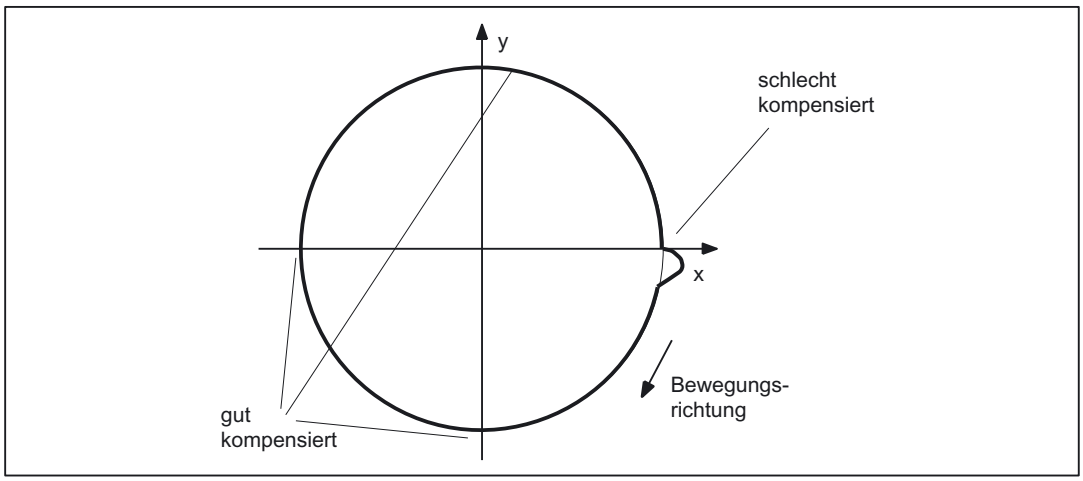


Bild 5-25 Beispiel für eine richtungsabhängige Reibkompensation (Kreisformtest)

Verändern der Kennlinienbereiche

Die Beschleunigungskennlinie ist in drei Bereiche aufgeteilt. Im Bereich kleiner Beschleunigungen ist eine besonders hohe Auflösung der Kennlinie erforderlich, um die dort stark differierenden Korrekturwerte nachzubilden. Daher erfolgt die Quantisierung der Eingangsgröße umso feiner, je kleiner die Beschleunigung ist (siehe folgendes Bild).

Im Bereich hoher Beschleunigungen ergeben sich nur geringfügige Änderungen der Korrekturwerte, so dass hier eine geringe Auflösung ausreichend ist.

Die in Kap. "Parametrierung der neuronalen QFK" empfohlenen Prozentwerte für $AA_QEC_ACCEL_1 (2% von $AA_QEC_ACCEL_3) und für $AA_QEC_ACCEL_2 (60% von $AA_QEC_ACCEL_3) liegen Erfahrungswerte zugrunde, die an Maschinen mit einer maximalen Beschleunigung (= Arbeitsbereich) bis zu ca. 1 m/s^2 gewonnen wurden.

Wird der Arbeitsbereich deutlich kleiner gewählt, so sind die Grenzwerte für a_1 und a_2 prozentual zu a_3 etwas größer festzulegen. $AA_QEC_ACCEL_1 sollte dennoch den Bereich von ca. 5% der Maximalbeschleunigung nicht überschreiten. Sinnvolle Grenzen für $AA_QEC_ACCEL_2 sind etwa die Werte von 40% bis 75% der Maximalbeschleunigung.

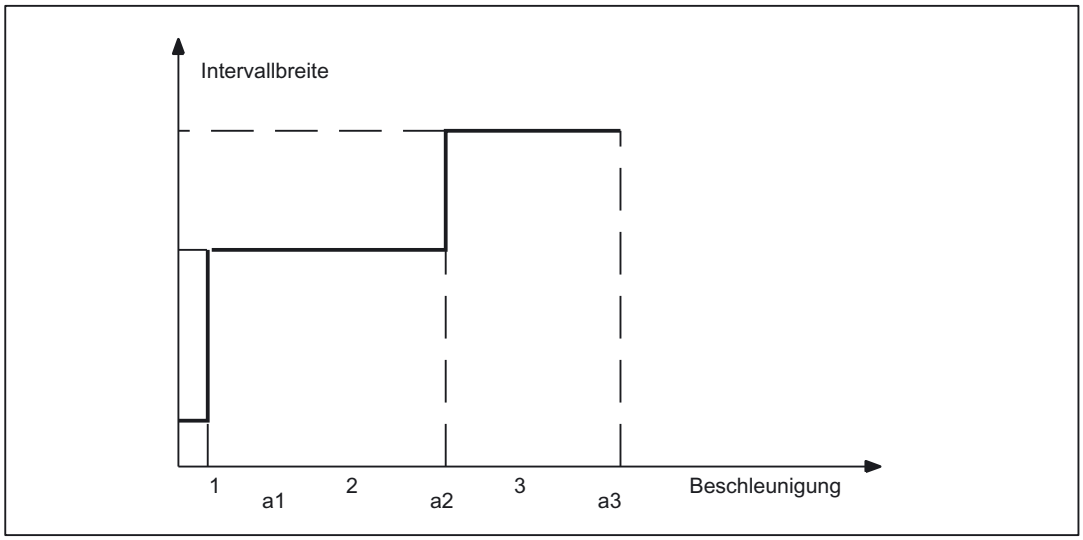


Bild 5-26 Intervallbreite in den Beschleunigungsbereichen

Adaption der Abklingzeit

In besonderen Fällen kann neben der Korrekturamplitude auch die Adaption der Abklingzeit des Korrektursollwertimpulses vorgenommen werden.

Falls beispielsweise anhand des Kreisformtests ermittelt wird, dass im Bereich von kleinen Beschleunigungen (a_1) zwar bei den Quadrantenübergängen gute Kompensationsergebnisse erzielt werden, jedoch unmittelbar danach wieder Radiusabweichungen vom programmierten Radius auftreten, kann mit Hilfe der Adaption der Abklingzeit eine Verbesserung erzielt werden.

Die ohne Adaption wirksame Zeitkonstante ($AA_QEC_TIME_1) gilt im mittleren Beschleunigungsbereich (50%).

Die Adaption der Abklingzeit des Korrektursollwertimpulses entsprechend der im folgenden Bild dargestellten Kennlinie wird mit der Systemvariablen $AA_QEC_TIME_2 parametriert (für Beschleunigung = 0). Die Adaption wird entsprechend einer e^{-x}-Funktion durch diese beiden Punkte gebildet (siehe folgendes Bild).

Die Adaption findet nur bei folgender Bedingung statt: $AA_QEC_TIME_2 > $AA_QEC_TIME_1

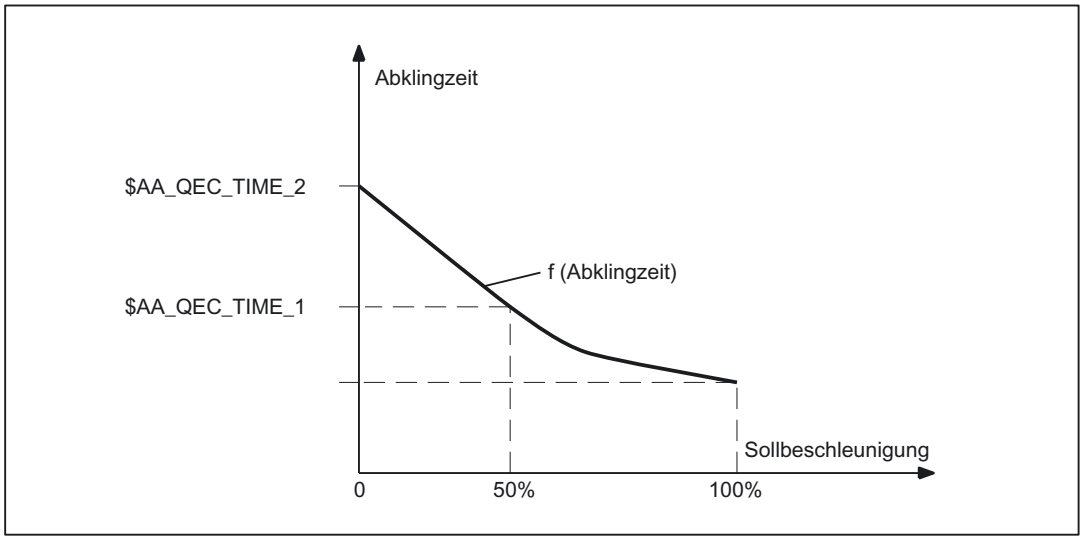


Bild 5-27 Adaption der Abklingzeit

Änderung der Fehler-Messdauer

Die Fehlermesszeit bestimmt in der Lernphase des neuronalen Netzes das Zeitfenster, während dessen nach dem Drehzahlnulldurchgang der Konturfehler überwacht wird.

Erfahrungsgemäß ist für die Fehler-Messdauer bei mittleren Beschleunigungen (ca. 2 bis 50mm/s^2) der dreifache Wert der Abklingzeit zu verwenden ($AA_QEC_MEAS_TIME_2 = 3 * $AA_QEC_ TIME_1).

Im Bereich großer und sehr kleiner Beschleunigung muss die Fehler-Messdauer angepaßt werden. Dies erfolgt automatisch nach der Kennlinie im folgenden Bild. Dabei wird für die Fehler-Messdauer bei kleinen Beschleunigungen der sechsfache Wert der Abklingzeit eingestellt ($AA_QEC_MEAS_TIME_1 = 6 * $AA_QEC_ TIME_1), für die Fehler-Messdauer bei großen Beschleunigungen der doppelte Wert der Abklingzeit ($AA_QEC_MEAS_TIME_3 = 2 * $AA_QEC_ TIME_1).

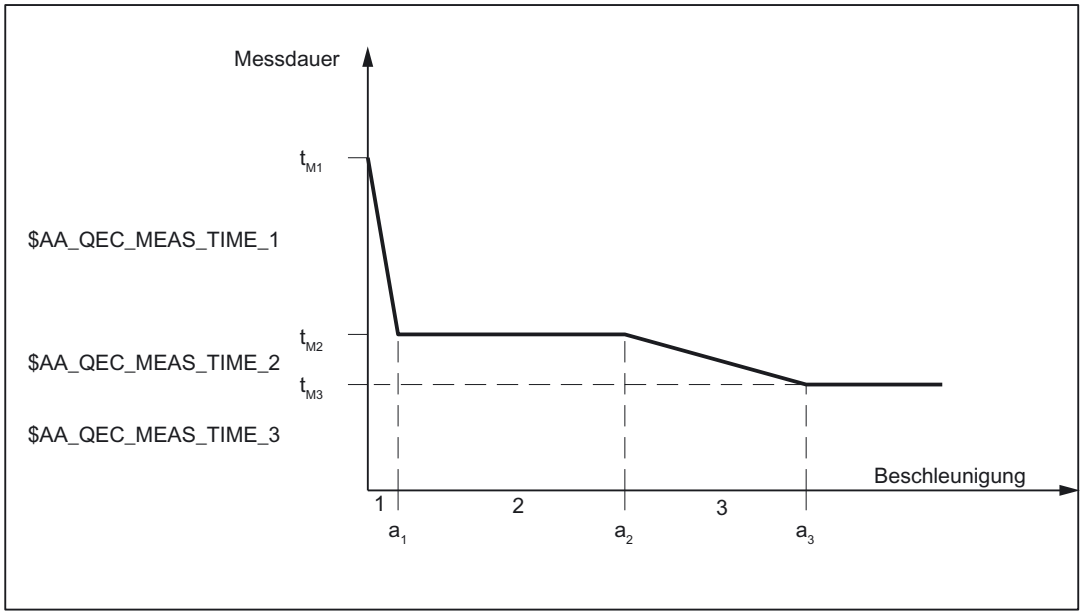


Bild 5-28 Abhängigkeit der Fehlermesszeit von der Beschleunigung

In besonderen Fällen kann dennoch eine Umparametrierung der Fehlermesszeiten notwendig werden:

- Einstellung sehr extremer Werte für die Kompensationszeitkonstante der QFK. Erfahrungsgemäß ist eine Fehler-Messdauer kleiner 10ms und größer 200ms nicht sinnvoll.
- Parametrierung der Fehlermesszeiten bei Adaption der Abklingzeit des Korrekturwertes

  Ist die Adaption der Abklingzeit des Korrekturwertes aktiv (siehe oben), so gilt für die Parametrierung der Fehler-Messdauer für Beschleunigungsbereich 1 folgende Faustregel:

  $AA_QEC_MEAS_TIME_1 = 3 * $AA_QEC_ TIME_2

  Beispiel:

  Abklingzeit ($AA_QEC_TIME_1) = 10ms

  Adaption der Abklingzeit ($AA_QEC_TIME_2) = 30ms

  Für die Fehler-Messdauer für Beschleunigungsbereich 1 ergibt sichnach der o.g. Faustregel:

  $AA_QEC_MEAS_TIME_1 = 3 * 30ms = 90ms

  Ohne Adaption der Abklingzeit würde sich lediglich für $AA_QEC_MEAS_TIME_1 = 6 * 10ms = 60ms ergeben.

Überkompensation bei kurzen Verfahrbewegungen

Erfahrungen haben gezeigt, dass der aus dem Kreisformtest ermittelte optimale Reibkompensationswert unter Umständen bei kurzen axialen Positioniervorgängen (z. B. bei Zustellungen im mm-Bereich) zu einer Überkompensation in dieser Achse führen kann.

Um die Genauigkeit auch für diese Fälle zu verbessern, kann bei kurzen Verfahrbewegungen die Korrekturamplitude reduziert werden.

MD32580 $MA_FRICT_COMP_INC_FACTOR

(Wichtungsfaktor Reibkompensationswert b. kurzen Verfahrbew.)

Dieser mit dem obigen Maschinendatum vorgebbare Wichtungsfaktor wirkt automatisch bei eingeschalteter Reibkompensation (konventionelle oder QFK mit neuronalen Netzen) und allen Positionierbewegungen, die innerhalb eines Interpolatortaktes von der Steuerung abgefahren werden.

Der Eingabebereich liegt zwischen 0 bis 100% des ermittelten Kompensationswertes.

Beeinflussung der Lerndauer

Wie bereits in den vorliegenden Kapiteln erläutert wurde, ist die Lerndauer von mehreren Parametern abhängig. Im wesentlichen wird sie von folgenden Werten bestimmt:

- Grobquantisierung ($AA_QEC_COARSE_STEPS)
- Messdauer zur Ermittlung des Fehlerkriteriums ($AA_QEC_MEAS_TIME_1 bis $AA_QEC_MEAS_TIME_3)
- Anzahl der Lerndurchläufe
- Ausführliches Lernen aktiv [ja/nein]?
- Feinquantisierung ($AA_QEC_FINE_STEPS)(nur wenn "Ausführliches Lernen aktiv = ja" angewählt ist)
- Richtungsabhängige Kompensation aktiv [ja/nein]? ($AA_QEC_DIRECTIONAL)
- Dauer der Reversierbewegung

Eine deutliche Erhöhung der benötigten Lerndauer bewirkt die Einstellung "Ausführliches Lernen aktiv = ja". Sie sollte daher nur bei sehr hohen Genauigkeitsanforderungen eingesetzt werden. Es sollte auch überprüft werden, ob diese hohen Anforderungen nur bei bestimmten Beschleunigungsbereichen erforderlich sind, und ggf. das ausführliche Lernen nur abschnittsweise durchgeführt wird (siehe "Abschnittsweise Lernen aktiv j/n?"). Auf jeden Fall sollte dann die Anzahl der Lerndurchläufe reduziert werden.

Bei Anwendung der o.g. Muster-NC-Programme mit den empfohlenen Parameterwerten wurden folgende Zeiten für die Lerndauer ermittelt:

- Ausführliches Lernen nicht aktiv: ca. 6,5 min.
- Ausführliches Lernen aktiv: ca. 13 min.

5.6.3.6 Schnellinbetriebnahme

Vorbereitung für "Lernen"

- Optimale Reibkompensationszeitkonstante

  (MD32540 $MA_FRICT_COMP_TIME (Umkehrlose)) mit der konventionellen Reibkompensation ermitteln.

- **Folgende Maschinendaten ohne POWER ON eingeben:**

Maschinendaten	Standard	Ändern auf	Bedeutung
MD19330 NC-CODE_CONF_NAME_TAB[8]	0		Option "IPO_FUNKTION_MASK" aktivieren. **Nur bei Lernprogramm "Polynom"! Bit4 = 1**
MD19300 COMP_MASK	0		Option setzen
MD32490 $MA_FRIC_COMP_MODE (Art der Reibkompensation)	1	2	"Art der Reibkompensation" neuro QFK
MD32500 $MA_FRIC_COMP_ENABLE (Reibkompensation aktiv)	0	0	"Reibkompensation aktiv" zum Lernen **"AUS"**
MD32580 $MA_FRIC_COMP_INC_FACTOR (Wichtungsfaktor Reibkompensationswert b. kurzen Verfahrbew.)	0	0	"Wichtungsfaktor des Reibkompensationswertes bei kurzen Verfahrbewegungen" (mm-Schritte)
MD38010 $MA_MM_QEC_MAX_POINTS (Anzahl der Werte für Quadrantenfehlerkomp. mit neuro. Netz)	0	400	"Anzahl der Werte für QFK" = $AA_QEC_FINE_STEPS * ($AA_QEC_COARSE_STEPS + 1)
MD32620 $MA_FFW_MODE (Vorsteuerungsart)	1	1	Drehzahlvorsteuerung
MD32610 $MA_VELO_FFW_WEIGHT (Vorsteuerfaktor für Drehzahlvorsteuerung)	1	1	Aufschaltung 100%
MD32630 $MA_FFW_ACTIVATION_MODE (Vorsteuerung aktivieren von Programm)	1	0	Vorsteuerung dauernd EIN
MD32810 $MA_EQUIV_SPEEDCTRL_TIME (Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung)	0,004	Ausgangswert t_lage + n_sollGl.*	Ersatzzeitk. n-Regelkreis anpassen

* t_lage ... Lageregeltakt (=Systemgrundtakt*Faktor für Lageregeltakt), n_sollGl. ... Drehzahlsollwertglättung (MD1500 bis 1521)

- **Einlesen der Maschinendaten aufgrund der Speicheränderung (MD38010).**
  - **HMI Embedded:**

    "Dienste" "Daten aus" "Inbetriebnahmedaten, NCK-Daten" und falls vorhanden "SSFK-, Messsystemfehler-, Durchhang- und Winkligkeitsfehlerkompensationstabellen" über PCIN sichern. `POWER ON`-Reset durchführen und anschließend mit PCIN und "Daten ein" die gesicherten Daten einlesen. (= Serieninbetriebnahme).
  - **HMI Advanced:**

    "SERIEN IBN" und falls vorhanden "SSFK-, Messsystemfehler-, Durchhang- und Winkligkeitsfehlerkompensationstabellen" sichern. `POWER ON`-Reset durchführen und Archiv "IBN" einlesen (gesicherte Daten werden wieder geladen).
- **Die in der Toolbox enthaltenen Programme in die NC überspielen (mit Archiv!)**

  QECDAT.MPF

  QECSTART.MPF

  QECLRNP.SPF (Lernprogramm Polynom) **oder** QECLRNC.SPF (Lernprogramm Kreis) wird als QECLRN.SPF auf der NC gespeichert!

  Bei GEO-Achsen sollte vorzugsweise das Lernprogramm Kreis, bei allen anderen Achsen nur das lernprogramm Polynom verwendet werden.
- **Die folgenden Programme anpassen:**
  - **Im Teileprogramm QECDAT**

    ggf. Reibkompensations-Zeitkonstante anpassen (siehe Punkt 1)

    `N1340 $AA_QEC_TIME_1[outNr,axNr] = 0.0xx`

    `N1040 def int numAxes =` ..... die Anzahl der zu lernenden Achsen eingeben.

    `N1150 axisName[0] =` ..... Achsname 1. Achse eingeben.

    `N1160 axisName[1] =` ..... Achsname 2. Achse eingeben.

    `N1170 axisName[2] =` ..... Achsname 3. Achse eingeben.

    `N1180 axisName[3] =` ..... Achsname 4. Achse eingeben.

    (Beim Lernprogramm "Kreis" kann als Achsname AX1 .. AX8 bzw. der Maschinen oder Kanalachsname verwendet werden. Beim Lernprogramm "Polynom" hingegen nur der Kanalachsname)
  - **Im Teileprogramm QECSTART**

    (Beim Lernprogramm "Kreis" kann als Achsname AX1 .. AX8 bzw. der Maschinen oder Kanalachsname verwendet werden. Beim Lernprogramm "Polynom" hingegen nur der Kanalachsname)

    `N1080 def int numAxes =` ..... die Anzahl der zu lernenden Achsen eingeben.

    `N1310 axisName[0] =` ..... Achsname 1. Achse eingeben.

    `N1320 axisName[1] =` ..... Achsname 2. Achse eingeben.

    `N1330 axisName[2] =` ..... Achsname 3. Achse eingeben.

    `N1340 axisName[3] =` ..... Achsname 4. Achse eingeben.

"Lernen" durchführen

Folgende Programme starten

- QECDAT anwählen und starten. Systemvariablen werden belegt.
- QECSTART anwählen, Override 100% und starten. Das Lernprogramm läuft ca. 15 min. mit ca. 30 cm Verfahrbewegung. Eine evt. Meldung "REORG nicht möglich" kann ignoriert werden. Die Meldung steht ca. 10 sec. an. Danach verschwindet sie und der Lernvorgang läuft mit Verfahrbewegungen ab.

QFK aktivieren

Maschinendaten	Standard	Ändern auf	Bedeutung
MD32500 $MA_FRIC_COMP_ENABLE (Reibkompensation aktiv)	0	1	"Reibkompensation aktiv" einschalten

"Kreisformtest"

Mit "Kreisformtest" das Ergebnis überprüfen!

Kompensationsdaten sichern

Kompensationsdaten sichern (QFK-Daten werden mit "SERIEN IBN" nicht mitgesichert):

HMI Embedded:

Unter DIENSTE Daten\Kreisfehler-Kompensation\alle mit PCIN sichern

HMI Advanced:

Unter DIENSTE im Verzeichnis NC-Aktive Daten \ Quadrantenfehlerkompensation die Datei Quadrantenfehlerkomp-komplett sichern. Diese Datei enthält alle Kompensationswerte.

Hinweis

In DIENSTE "Systemeinstellungen" "für Anzeige" "angezeigte Namenslänge" auf "20" ändern damit der komplette Name sichtbar wird.

5.7 Kreisformtest

Funktion

Der Kreisformtest dient u. a. als Kontrollmittel für die mit der Reibkompensation (konventionelle oder neuronale QFK) erzielte Konturgenauigkeit. Dabei werden bei einer Kreisbewegung die Istpositionen gemessen und graphisch die Abweichungen vom programmierten Radius (insbesondere an den Quadrantenübergängen) dargestellt.

Vorgehen

Die Kreiskontur für die beteiligten Achsen wird über ein NC-Programm vorgegeben. Um den Kreisformtest für den Inbetriebnehmer möglichst einfach zu gestalten, wird ein NC-Programme als Musterbeispiel für die Kreisformtestbewegung zur Verfügung gestellt (Datei QECTEST.MPF auf Diskette vom PLC-Grundprogramm). Dieses NC-Programm ist vom Inbetriebnehmer für seinen Anwendungsfall anzupassen.

Um festzustellen, ob die gelernte Korrekturkennlinie (bei neuronaler QFK) bzw. die festgelegten Kompensationswerte (bei konventioneller Reibkompensation) den Genauigkeitsanforderung entspricht, sind beim Kreisformtest mehrere Messungen mit unterschiedlichen Beschleunigungswerten vorzunehmen.

Das Abfahren der Kreisbewegung mit verschiedenen Beschleunigungen ist auf einfache Weise möglich, indem man bei unveränderter Kreiskontur die Vorschubgeschwindigkeit mit Hilfe des Vorschubkorrekturschalters verändert. Die tatsächliche Vorschubgeschwindigkeit muss bei der Messung im Eingabefeld "Vorschub" berücksichtigt werden.

Für die Wahl des Kreisradius sollten typische Bearbeitungsfälle für die Maschine gewählt werden (z. B. Radius im Bereich von 10 bis 200mm).

Für die Zeitdauer der Kreisbewegung werden die Lageistwerte der Achsen aufgenommen und in einen sog. "Trace" im passiven Dateisystem abgespeichert. Der Kreisformtest ist somit eine reine Messfunktion.

Parametrierung des Kreisformtests

In diesem Menü werden die **Achsnamen** oder Achsnummern ausgewählt, mit denen der Kreis gefahren wird und deren Lageistwerte aufzuzeichnen sind. Ob die angewählten Achsen mit den im NC-Teileprogramm programmierten Achsen übereinstimmen, wird nicht überprüft.

Bei der Parametrierung der Eingabefelder "Radius" und "Vorschub" sind dieentsprechenden Werte aus dem Teileprogramm, das die Kreisbewegung der Achsen steuert, unter Berücksichtigung des Vorschubkorrekturschalters einzutragen. Die Übereinstimmung zwischen den Werten im Teileprogramm (einschließlich Vorschubkorrektur) und den Eingabewerten wird nicht überprüft.

Im Anzeigefeld "Messzeit" wird die aus den Werten "Radius" und "Vorschub" errechnete Messdauer für die Aufzeichnung der Lageistwerte bei der Kreisfahrt angezeigt.

Falls nur Teile vom Kreis dargestellt werden (d. h. Messzeit ist nicht ausreichend) kann durch Reduzierung des Vorschubwertes im Menü die Messzeit erhöht werden. Dies trifft auch zu, wenn der Kreisformtest aus dem Stillstand gestartet wird.

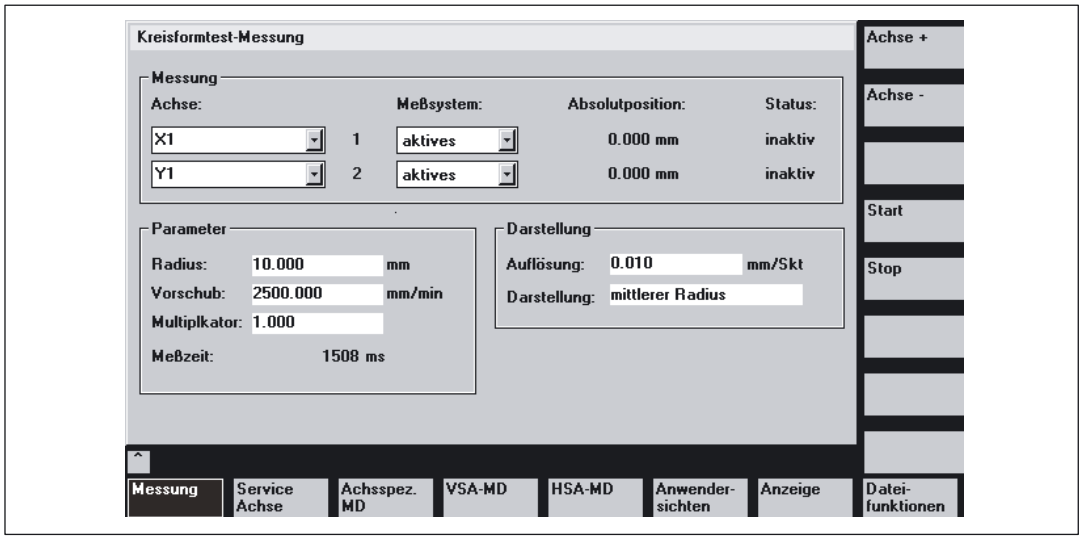


Bild 5-29 Menü Kreisformtest-Messung

Darstellungsart

Außerdem ist folgende Parametrierung für die Darstellungsart der Messergebnisse möglich:

- Darstellung über mittleren Radius
- Darstellung über programmierten Radius
- Auflösung (Skalierung) der Diagrammachsen

Überschreitet die ermittelte Messzeit den mit den Tracepuffern darstellbaren Zeitbereich (maximale Messzeit = Lagereglertakt * 2048), so wird für die Aufzeichnung eine entsprechende gröbere Abtastung (n * Lagereglertakt) vorgenommen, damit ein vollständiger Kreis darstellbar ist.

Messung starten

Vom Bediener muss das Teileprogramm, in dem für die angewählten Achsen die Kreisbewegung hinterlegt ist, mit NC-Start gestartet werden (Betriebsart `AUTOMATIC` oder MDA).

Die Messfunktion wird mit dem vertikalen Softkey **Start** gestartet.

Die Bedienreihenfolge (NC-Start des Teileprogramms und Messung starten) kann vom Bediener je nach Anwendungsfall beliebig gewählt werden.

Sobald der Kreisformtest für die vorgegebenen Achsen wirksam ist, wird im Anzeigefeld "Status" die Information "aktiv" angezeigt.

Messung stoppen

Mit dem vertikalen Softkey **Stop** kann die Messung jederzeit unterbrochen werden. Dabei möglicherweise unvollständig aufgezeichnete Messungen werden unter dem Softkey **Anzeige** bestmöglich angezeigt. Diesbezüglich sind keine Überwachungen wirksam.

Damit ein direkter Zugriff auf die benötigten Reglerparameter besteht, werden die Softkeys **Achsspezifische MD, VSA-MD** und **HSA-MD** angeboten. Mit den vertikalen Softkeys **Achse+** und **Achse-** kann die gewünschte Achse angewählt werden.

Nach Betätigung des Softkeys **ServiceAchse** wird das Bild "Service Achse" angezeigt. Hiermit werden u.a. für die Inbetriebnahme der Reibkompensation folgende Servicedaten zyklisch angezeigt:

- QFK-Lernen aktiv ja/nein?
- aktuelle Lage- und Drehzahlistwerte

Anzeige

Bei Betätigung des Softkeys **Anzeige** wird auf die graphische Darstellung des aufgezeichneten Kreisdiagramms umgeschaltet.

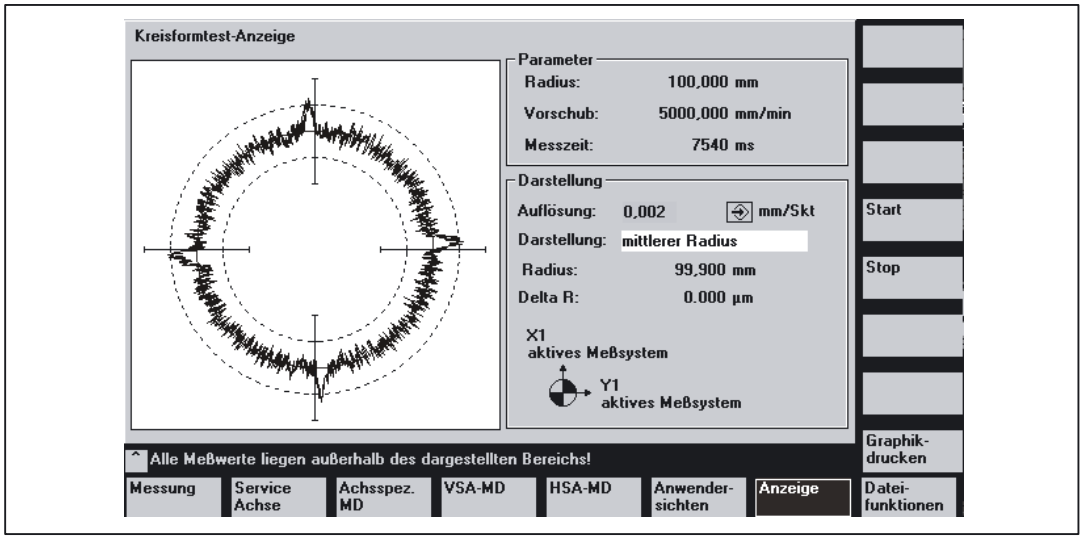


Bild 5-30 Menü Kreisformtest-Anzeige

In diesem Bild wird der gemessene Verlauf der beiden Lageistwerte mit der eingestellten Auflösung als Kreis dargestellt.

Desweiteren werden zu Dokumentationszwecken (für die spätere Speicherung der gemessenen Kreiskennlinien als Datei) der programmierte Radius, der programmierte Vorschub und die daraus abgeleitete Messßzeit angezeigt.

Der Bediener kann im Eingabefeld **Auflösung** eine feinere Skalierung der Diagrammachsen eingeben, um z. B. die Übergänge an den Quadranten noch besser hervorzuheben. Eine Neudarstellung des gesamten Kreisdiagramms mit veränderter Auflösung erfolgt nach der Betätigung des Softkeys **Anzeige**.

Dateifunktionen

Die dargestellten Messergebnisse sowie die vorgegebenen Parameter können als Datei auf der PCU mit dem Softkey **Dateifunktionen** gespeichert werden.

Druckereinstellung

Über die Softkeys **HMI** \ **Druckerauswahl** erreicht man aus dem Grundbild die Druckerauswahl.

Mittels der Toggle-Taste erfolgt die Auswahl, ob die angezeigte Graphik nach Betätigung des Softkeys **Graphik drucken** direkt auf den Drucker ausgegeben wird oder die Ausgabe in eine Bitmapdatei erfolgt.

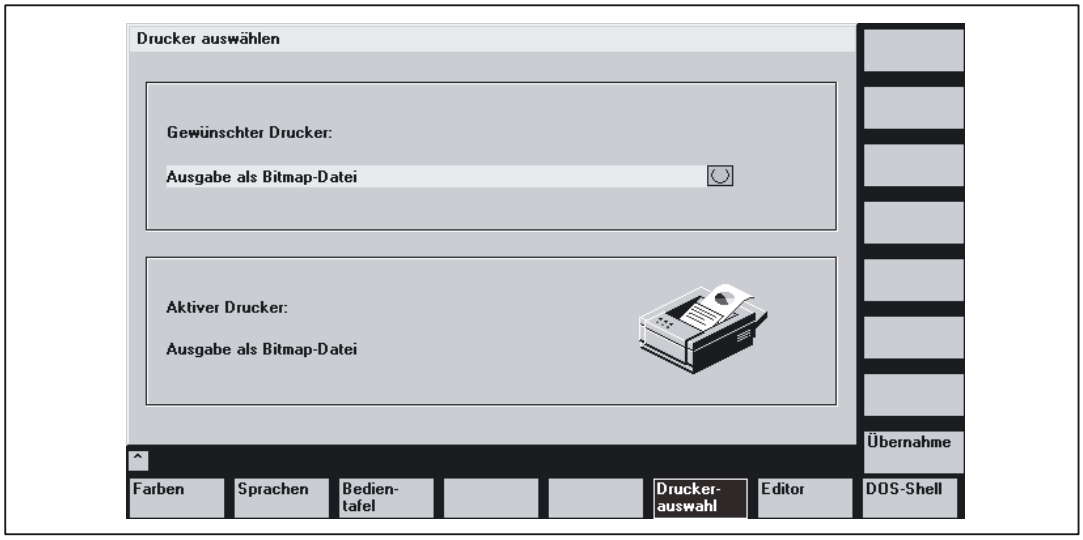


Bild 5-31 Grundbild der Druckerauswahl

Direkte Ausgabe auf Drucker

Der Drucker muss unter MS-WINDOWS eingerichtet sein.

Im Auswahlfeld wird "Ausgabe auf Drucker" eingestellt. Nach Betätigung des Softkeys **Graphik drucken** wird die angezeigte Graphik auf den angeschlossenen Drucker ausgegeben.

Ausgabe in Bitmapdatei

Die Graphik wird in einer Bitmap-Datei (*.bmp) gespeichert.

Im Auswahlfeld der Druckereinstellung wird "Ausgabe als Bitmap Datei" eingestellt.

Im Bild "Kreisformtest Anzeige" wird dann bei Betätigung des Softkeys **Graphik drucken** die Maske zur Vergabe eines Dateinamens aufgeblendet. In der Drop-Down Liste kann ein neuer Dateiname eingegeben werden bzw. ein bereits vorhandener Dateiname zum Überschreiben ausgewählt werden.

Mit dem Softkey **Ok** wird die Datei abgespeichert. Mit dem Softkey **Abbruch** erfolgt der Rücksprung in das aktuelle Graphikbild.

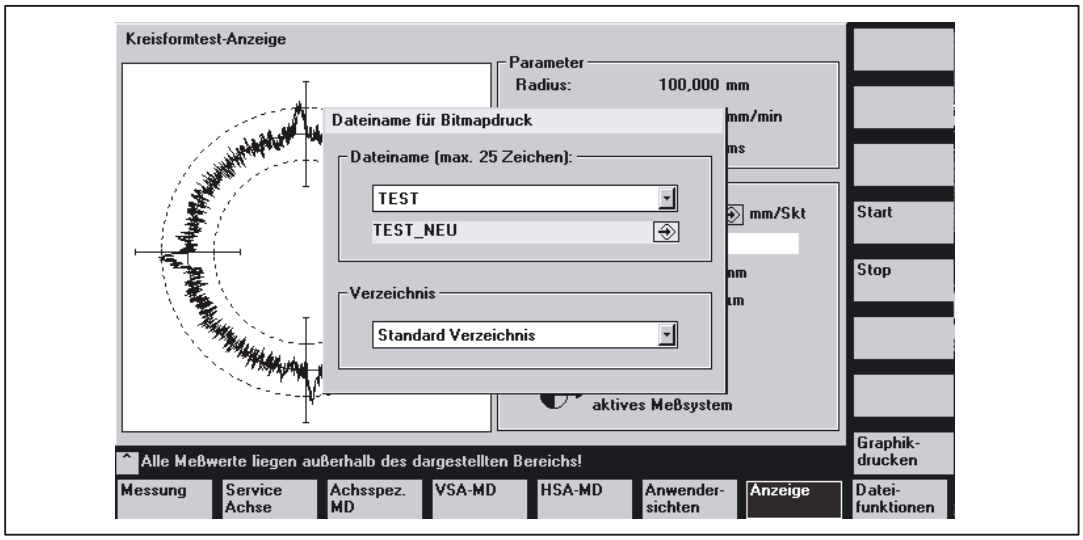


Bild 5-32 Dateinamenvergabe für die Ausgabe in eine Bitmap-Datei.

5.8 Maßnahmen bei hängenden Achsen

5.8.1 Elektronischer Gewichtsausgleich

Achse ohne Gewichtsausgleich

Bei gewichtsbelasteten Achsen ohne einen Gewichtsausgleich senkt sich die hängende Achse unerwünscht nach dem Lösen der Bremse und das folgende Verhalten stellt sich ein:

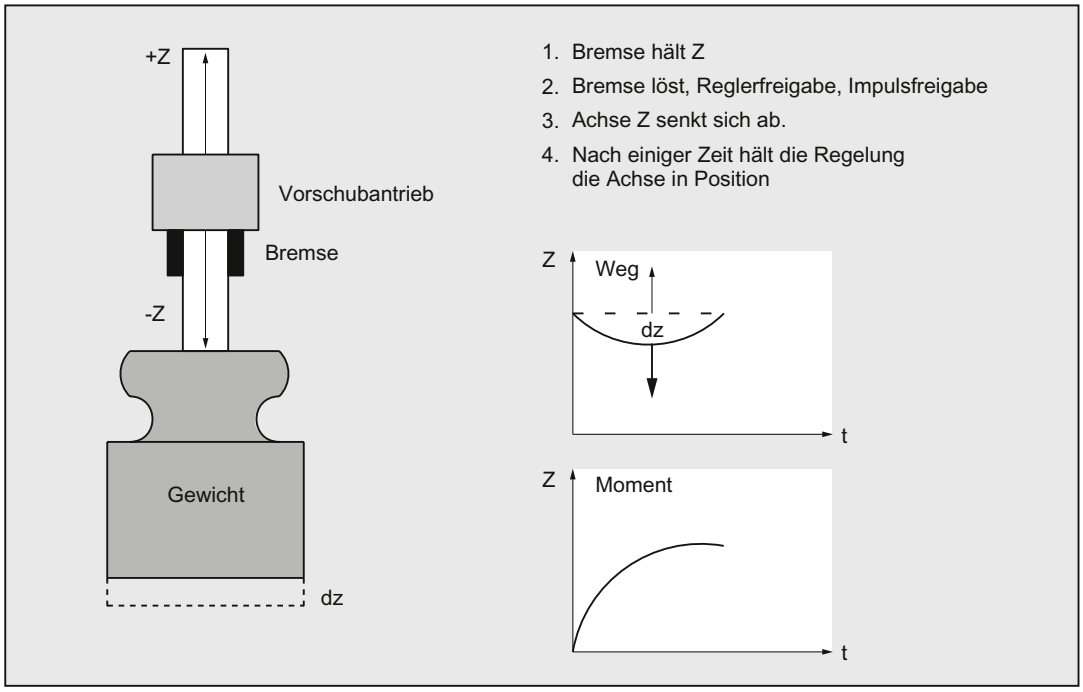


Bild 5-33 Absenkung einer hängenden Achse ohne Gewichtsausgleich

Funktion "Elektronischer Gewichtsausgleich"

Eine Absenkung der hängenden Achse kann mit der Funktion "Elektronischer Gewichtsausgleich" nahezu vollständig vermieden werden.

Der elektronische Gewichtsausgleich vermindert das Durchsacken gewichtsbelasteter Achsen beim Einschalten der Regelung. Nach dem Lösen der Bremse hält das anstehende konstante Gewichtsausgleichsmoment die Position der hängenden Achse:

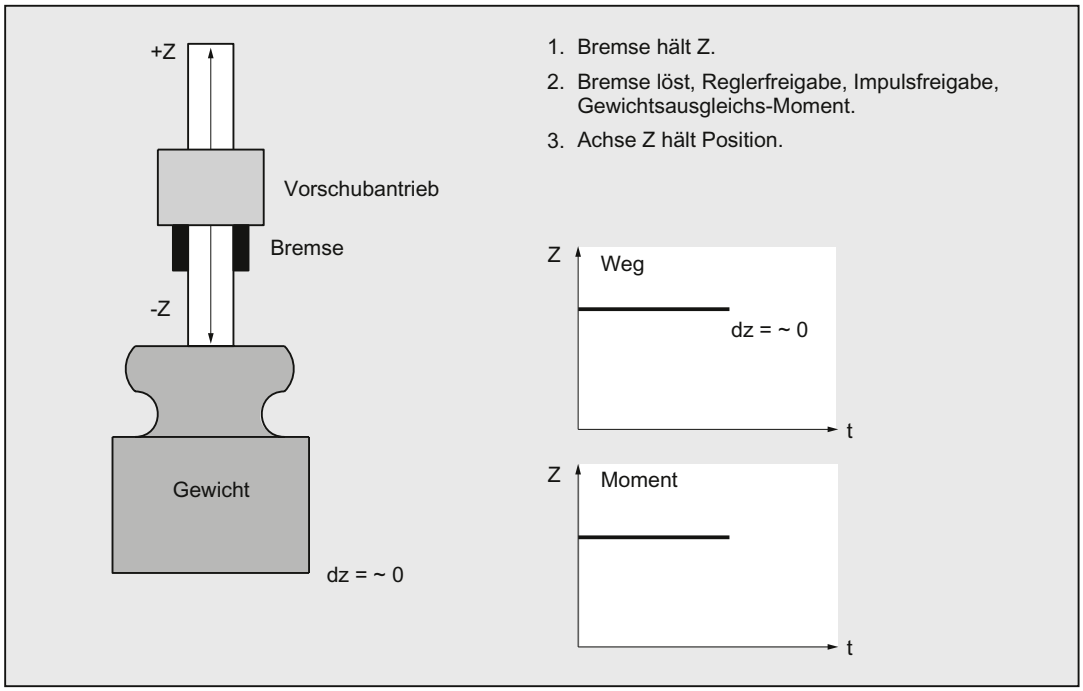


Bild 5-34 Absenkung einer hängenden Achse mit elektronischem Gewichtsausgleich

Inbetriebnahme

Hinweis

Die Inbetriebnahme des "Elektronischen Gewichtsausgleichs" erfolgt über den Antrieb!

Literatur

Weitere Informationen siehe:

SINAMICS S120 Funktionshandbuch Antriebsfunktionen

5.8.2 Reboot-Verzögerung

Nebenwirkung eines Reboots via HMI

HMI besitzt die Möglichkeit, den NCK zu booten, um z. B. Maschinendaten wirksam zu setzen. Das würde dazu führen, dass hängende Achsen ein Stück fallen. Um dies zu vermeiden, kann von der Funktion "Reboot-Verzögerung" Gebrauch gemacht werden.

Reboot-Verzögerung

Die Reboot-Verzögerung führt zu einer verzögerten Abschaltung von NCK und PLC und teilt die bevorstehende Abschaltung mit, um ein Fallen hängender Achsen verhindern zu können.

Hinweis

Die Reboot-Verzögerung wirkt nur beim kontrollierten POWER ON via HMI.

Ein POWER-FAIL (Stromausfall) bzw. Hardware-Reboot aktiviert die Reboot-Verzögerung nicht.

Ablauf des Reboot

Mit dem PI-Dienst "_N_IBN_SS" löst die HMI ein Reboot des NCK und der PLCs aus.

Unmittelbar mit dem PI-Dienst löst die NCK den Alarm 2900 aus.

In der Zeit, die der NCK vom PI-Dienst an bis zum Reboot verstreichen lässt (Reboot-Verzögerungszeit, siehe MD10088 $MN_REBOOT_DELAY_TIME), können mechanische **Achsbremsen aktiviert werden**.

Reaktionen mit Alarm 2900

- Folgende NC/PLC-Nahtstellensignale werden **gelöscht**, d. h. auf Null gesetzt:
  - DB11 DBX 6.3 (BAG betriebsbereit) ; alle BAGs
  - DB21, … DBX 36.5 (Kanal betriebsbereit) ; alle Kanäle
  - DB31, … DBX 61.2 (Achse betriebsbereit) ; alle Achsen
- Die Meldung "Betriebsbereit" an den Relaiskontakten 72 73.1 73.2 74 wird **nicht** weggenommen.
- Der NCK bremst an der Stromgrenze.

  Siehe dazu die Maschinendaten:
  - MD36610 $MA_AX_EMERGENCY_STOP_TIME (Zeitdauer der Bremsrampe bei Fehlerzuständen)
  - MD36620 $MA_SERVO_DISABLE_DELAY_TIME (Abschaltverzögerung Reglerfreigabe)

  Hinweis

  Der NCK nimmt nach der Abschaltverzögerung der Reglerfreigabe (MD36620) die Lageregelung weg.

- Folgende NC/PLC-Nahtstellensignale bleiben auf 1:

  DB10 DBX108.7 (NC ready)

Durch die Verwendung des Maschinendatums:
MD11410 $MN_SUPPRESS_ALARM_MASK (Maske zur Unterstützung spezieller Alarme) (BIT20)
wird der Alarm 2900 unterdrückt, der NCK löst aber die gleichen Reaktionen aus.

Da Alarm 2900 die Lageregelung der Achse wegnimmt, muss dieser Alarm zum **Schließen der mechanischen Bremsen durch den PLC** führen. Mit dem Reboot des PLC werden die PLC-Ausgänge auf definiert Null gezwungen. Die Bremsen müssen so verschaltet sein, dass sie **bei Null geschlossen** bleiben, d. h. ein 1-Signal des PLC lässt die Bremse offen.

Hinweis

Der Alarm verhält sich in seinen Reaktionen wie der Not-Halt-Alarm (3000). Aus internen Gründen kann die Reboot-Verzögerungszeit vom NCK etwas verlängert werden.

Aktivierung

Die Reboot-Verzögerung wird wie folgt aktiviert:

MD10088 $MN_REBOOT_DELAY_TIME (Reboot-Verzögerung) **> 0**

Der eingetragene Wert liefert die Reboot-Verzögerungszeit in Sekunden.

Auswertung mit einer Systemvariablen

Die Systemvariable $AN_REBOOT_DELAY_TIME kann in einer Synchronaktion gelesen werden. Ein Wert größer als Null zeigt an, dass die von HMI ausgelöste Reboot-Aufforderung gegeben wurde und wieviel Zeit (in Sekunden) der NCK bis zum Reboot (POWER OFF und anschließend POWER ON) noch vorsieht. Der Anwender kann in einer Synchronaktion den bevorstehenden Reboot erkennen und darauf reagieren (z. B. mit "Sicherer Halt" bei einer Safety-Integrated-Anwendung). $AN_REBOOT_DELAY_TIME ist 0.0, solange keine von HMI ausgelöste Reboot-Aufforderung ansteht.

5.9 Datenlisten

5.9.1 Maschinendaten

5.9.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
10050	SYSCLOCK_CYCLE_TIME	Systemgrundtakt
10070	IPO_SYSCLOCK_TIME_RATIO	Faktor für Interpolatortakt
10082	CTRLOUT_LEAD_TIME	Verschiebung des Sollwertübernahmezeitpunkts
10083	CTRLOUT_LEAD_TIME_MAX	Maximal einstellbare Verschiebung Sollwertübernahmezeitpunkt
10088	REBOOT_DELAY_TIME	Reboot-Verzögerung
18342	MM_CEC_MAX_POINTS[t]	Maximale Anzahl der Stützpunkte einer Durchhangkompensation

5.9.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
20150	GCODE_RESET_VALUES	Löschstellung der G-Gruppen

5.9.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: $MA_	Beschreibung
32450	BACKLASH	Umkehrlose
32452	BACKLASH_FACTOR	Bewertungsfaktor für Umkehrlose
32490	FRICT_COMP_MODE	Art der Reibkompensation
32500	FRICT_COMP_ENABLE	Reibkompensation aktiv
32510	FRICT_COMP_ADAPT_ENABLE	Adaption Reibkompensation aktiv
32520	FRICT_COMP_CONST_MAX	Maximaler Reibkompensationswert
32530	FRICT_COMP_CONST_MIN	Minimaler Reibkompensationswert
32540	FRICT_COMP_TIME	Reibkompensations-Zeitkonstante
32550	FRICT_COMP_ACCEL1	Adaptions-Beschleunigungswert 1
32560	FRICT_COMP_ACCEL2	Adaptions-Beschleunigungswert 2
32570	FRICT_COMP_ACCEL3	Adaptions-Beschleunigungswert 3
32580	FRICT_COMP_INC_FACTOR	Wichtungsfaktor für Reibkompensationswert bei kurzen Verfahrbewegungen
32610	VELO_FFW_WEIGHT	Vorsteuerfaktor für Geschwindigkeits-/Drehzahlvorsteuerung

Nummer	Bezeichner: $MA_	Beschreibung
32620	FFW_MODE	Vorsteuerungsart
32630	FFW_ACTIVATION_MODE	Vorsteuerung aktivieren von Programm
32650	AX_INERTIA	Trägheit für Drehmomentvorsteuerung
32700	ENC_COMP_ENABLE	Interpolatorische Kompensation
32710	CEC_ENABLE	Freigabe der Durchhangkompensation
32711	CEC_SCALING_SYSTEM_METRIC	Maßsystem der Durchhangkompensation
32720	CEC_MAX_SUM	Maximaler Kompensationswert bei Durchhangkompensation
32730	CEC_MAX_VELO	Geschwindigkeitsänderung bei Durchhangkompensation
32750	TEMP_COMP_TYPE	Temperaturkompensationstyp
32760	COMP_ADD_VELO_FACTOR	Geschwindigkeitsüberhöhung durch Kompensation
32711	CEC_SCALING_SYSTEM_METRIC	Maßsystem der Durchhangkompensation
32800	EQUIV_CURRCTRL_TIME	Ersatzzeitkonstante Stromregelkreis für Vorsteuerung
32810	EQUIV_SPEEDCTRL_TIME	Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung
32910	DYN_MATCH_TIME	Zeitkonstante der Dynamikanpassung
36500	ENC_CHANGE_TOL	Maximale Toleranz bei Lageistwertumschaltung
38000	MM_ENC_COMP_MAX_POINTS	Anzahl der Stützpunkte bei interpolatorischer Kompensation
38010	MM_QEC_MAX_POINTS	Anzahl der Werte für Quadrantenfehler-Kompensation mit neuronalem Netz

5.9.2 Settingdaten

5.9.2.1 Allgemeine Settingdaten

Nummer	Bezeichner: $SN_	Beschreibung
41300	CEC_TABLE_ENABLE[t]	Auswertung der Durchhangkompensations-Tabelle freigeben
41310	CEC_TABLE_WEIGHT[t]	Wichtungsfaktor für Durchhangkompensations-Tabelle

5.9.2.2 Achs-/Spindel-spezifische Settingdaten

Nummer	Bezeichner: $SA_	Beschreibung
43900	TEMP_COMP_ABS_VALUE	Positionsunabhängiger Temperaturkompensationswert
43910	TEMP_COMP_SLOPE	Steigungswinkel für positionsabhängige Temperaturkompensation
43920	TEMP_COMP_REF_POSITION	Bezugsposition für positionsabhängige Temperaturkompensation

5.9.3 Signale

5.9.3.1 Signale von NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
NC-Ready	DB10.DBX108.7	DB2700.DBX2.7

5.9.3.2 Signale von BAG

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
BAG Betriebsbereit	DB11.DBX6.3	DB3100.DBX0.3

5.9.3.3 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Kanal betriebsbereit	DB21, ...DBX36.5	DB3300.DBX4.5

5.9.3.4 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Referiert/Synchronisiert 1	DB31, ...DBX60.4	DB390x.DBX0.4
Referiert/Synchronisiert 2	DB31, ...DBX60.5	DB390x.DBX0.5
Achse betriebsbereit	DB31, ...DBX61.2	DB390x.DBX1.2

K5: BAGs, Kanäle, Achstausch

6

6.1 Kurzbeschreibung

Betriebsartengruppe

Die Betriebsartengruppe ist die Zusammenfassung von Maschinenachsen, Spindeln und Kanälen zu einer Einheit. Prinzipiell kann jede Betriebsartengruppe mit einer eigenständigen NC-Steuerung (mit mehreren Kanälen) verglichen werden. Eine Betriebsartengruppe enthält die Kanäle, die vom Betriebsablauf her immer gleichzeitig in derselben Betriebsart arbeiten müssen.

Hinweis

Im Standardfall ist eine BAG vorhanden.

Literatur:
Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb (K1)

Hinweis

Bei SINUMERIK 828D ist nur 1 BAG verfügbar.

Kanäle

Jeder Kanal besitzt eine eigene Programmdekodierung, Satzaufbereitung und Interpolation. Innerhalb eines Kanals kann ein eigenes Teileprogramm abgearbeitet werden.

Hinweis

Im Standardfall ist ein Kanal vorhanden.

Literatur:
Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb (K1)

Die Abläufe in mehreren Kanälen einer BAG können in den Teileprogrammen synchronisiert werden.

Hinweis

Bei SINUMERIK 828D ist nur 1 Kanal verfügbar.

Achs-/Spindeltausch

Eine Achse/Spindel ist nach dem Einschalten der Steuerung einem bestimmten Kanal zugeordnet und kann dann nur in diesem Kanal benutzt werden.

Mit der Funktion "Achs-/Spindeltausch" ist es möglich, eine Achse/Spindel freizugeben und einem anderen Kanal zuzuordnen, d. h. die Achse/Spindel zu tauschen.

Der Achs-/Spindeltausch kann sowohl durch das Teileprogramm als auch durch das PLC-Programm und aus Bewegungssynchronaktionen aktiviert werden.

Der Achs-/Spindeltausch ist möglich über:

- Programmierung im Teileprogramm GET/GETD.
- Automatisch durch Programmierung des Achsnamens.
- Ohne Vorlaufstopp und gegebener Synchronisation zwischen Vorlauf und Hauptlauf.
- Durch PLC über die VDI–Nahtstelle zur NCK.

Achstauscherweiterungen

- Achstauschverhalten veränderbar einstellen.
- Achstausch bei einer Achs–Containerdrehung mit impliziten GET/GETD
- Achstausch ohne Vorlaufstopp der nicht an der Kontur beteiligten Achsen
- Geometrieachse mit gedrehten WKS (ROT) und Achstausch in der Betriebsart JOG.
- Achstausch über Synchronaktionen GET(Achse), AXTOCHAN.

Hinweis

Bei SINUMERIK 828D ist ein Achs-/Spindeltausch zwischen Kanälen nicht möglich.

6.2 Betriebsartengruppen (BAGs) - nur 840D sl

Betriebsartengruppen

Eine Betriebsartengruppe fasst NC-Kanäle mit Achsen und Spindeln zu einer Bearbeitungseinheit zusammen.

Eine Betriebsartengruppe enthält die Kanäle, die vom Bearbeitungsablauf her immer gleichzeitig in der gleichen Betriebsart laufen müssen.

Innerhalb der Betriebsartengruppe kann jede Achse in jedem Kanal programmiert werden. Eine Betriebsartengruppe kann also als eine eigenständige, mehrkanalige NC angesehen werden.

Beispiel

Bei größeren Werkzeugmaschinen (Bearbeitungszentren) besteht die Notwendigkeit, dass in einem Teil der Maschine ein Teileprogramm abgearbeitet werden soll, während in einem anderen Teil neue zu bearbeitende Werkstücke aufgespannt und eingerichtet werden sollen. Solche Aufgaben erfordern normalerweise zwei eigenständige NC-Steuerungen.

Durch die Funktion Betriebsartengruppen können jedoch beide Aufgaben mit einer NC-Steuerung mit zwei BAGs realisiert werden, da für jede Betriebsartengruppe eine andere Betriebsart eingestellt werden kann (AUTOMATIK für die Programmbearbeitung, JOG für das Einrichten eines Werkstückes).

Zuordnung der Betriebsartengruppe

Mit der Konfiguration einer Betriebsartengruppe wird festgelegt, welche Kanäle, Geometrieachsen, Maschinenachsen und Spindeln eine BAG bilden.

Eine Betriebsartengruppe besteht aus einem oder mehreren Kanälen, die nicht zusätzlich noch einer anderen BAG zugeordnet sein dürfen. Den Kanälen sind wiederum Maschinenachsen, Geometrieachsen und Zusatzachsen zugeordnet. Eine Maschinenachse darf nur den Kanälen einer BAG zugeordnet sein und kann nur innerhalb dieser verfahren werden.

Die Konfiguration einer Betriebsartengruppe erfolgt über folgende Daten:

- Kanalspezifisches Maschinendatum:

  MD10010 $MN_ASSIGN_CHAN_TO_MODE_GROUP (Kanal gültig in BAG)

- Konfigurationsdaten der Kanäle

Hinweis

Informationen für die erste BAG sind zu entnehmen der:

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb (K1)

6.3 Kanäle - nur 840D sl

Hinweis

Der Begriff Kanal, die Kanalkonfiguration, Kanalzustände, Auswirkungen von Kommandos/Signalen usw. ist für den ersten Kanal beschrieben in:
Literatur:
Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb (K1)

Für alle weiteren Kanäle können diese Informationen sinngemäß übertragen werden.

6.3.1 Kanalsynchronisation (Programmkoordination)

Allgemeines

z. B. bei Doppelschlittenbearbeitung oder Echtzeitaktionen muss die Möglichkeit zur Synchronisation der Bearbeitung zwischen Kanälen vorhanden sein. Die beteiligten Kanäle sollen damit bestimmte Bearbeitungsvorgänge zeitlich abgestimmt ablaufen lassen. Damit dies möglich ist, müssen die beteiligten Kanäle zu einer Synchronisationsgruppe (BAG) zusammengefasst werden.

Die Kanalsynchronisation wird ausschließlich über die NC-Sprache programmiert.

Voraussetzung

Die beteiligten Kanäle müssen zu der **gleichen Betriebsartengruppe** gehören.

Programmkoordinierung

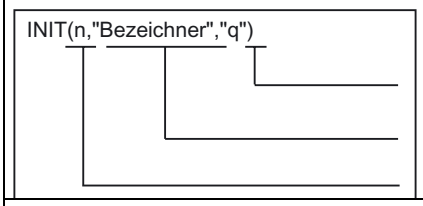
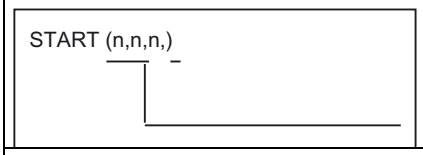
Sind mehrere Kanäle an der Fertigung eines Werkstückes beteiligt, so kann eine Synchronisation der Programmabläufe erforderlich werden.

Für diese Programmkoordinierung gibt es besondere Anweisungen (Kommandos). Sie stehen jeweils alleine in einem Satz.

Literatur:

/PG/ Programmierhandbuch Arbeitsvorbereitung; Flexible NC-Programmierung

Tabelle 6- 1 Anweisungen zur Programmkoordinierung

Anweisung	Bedeutung
	Anwahl eines Programms zur Abarbeitung in einem bestimmten Kanal Quittungsmodus: **n** (ohne) bzw. **s** (synchrone) Name des Programms mit Angabe des Pfad Nummer des Kanals: Wert 1 bis 4 möglich
CLEAR (Bezeichner)	Löschen eines Programms unter Angabe des Programmbezeichners
	Starten der angewählten Programme in den anderen Kanälen Aufzählung der Kanalnummern: Wert 1 bis 4 möglich
WAITM (Mnr, n, n, n, n)	Warten auf Markennummer Mnr zur Synchronisation in den angegebenen Kanälen n (eigener Kanal kann, muss nicht angeben werden). Die Markennummer muss in allen Kanälen gleich sein. Es sind die Nummern 0 bis 9 möglich.
WAITE (n,n,n)	Warten auf das Programmende der angegebenen Kanäle (eigenen Kanal nicht angeben)
SETM(Mnr1, Mnr2, ...Mnri)	Setzen der Wartemarken Mnr1, Mnr2, ...Mnri für bedingtes Warten mit WAITMC() für den Kanal, in dem SETM() abgegeben wird. Der Kanal erklärt damit sein Wartemerkmal für die Partnerkanäle als erfüllt. Der Befehl kann in Synchronaktionen aktiviert werden. Mit einem Befehl können bis zu 10 Marken (0 - 9) gesetzt werden.
CLEARM(Mnr1, Mnr2, ...Mnri)	Löschen der Wartemarken Mnr1, Mnr2, ...Mnri für bedingtes Warten mit WAITMC() für den Kanal, in dem CLEARM() abgegeben wird. Der Kanal erklärt damit sein Wartemerkmal für die Partnerkanäle als nicht/ nicht mehr erfüllt. Der Befehl kann in Synchronaktionen aktiviert werden. Mit einem Befehl können bis zu 10 Marken (0 - 9) gelöscht werden.
WAITMC(Mnr, n1, n2, ...)	Bedingtes Warten im Bahnsteuerbetrieb auf die angegebene Wartemarke Mnr aus den genannten Kanälen n1, n2, ... nk. Der eigene Kanal kann, muss aber nicht mit angegeben werden. Nach Fortsetzung bei eingetroffener Wartemarke aus den betreffenden Kanälen werden die Wartemarken dieser Kanäle gelöscht.

Die Anzahl der Marker ist abhängig von der eingebauten CPU

CPU 572 --> 2 Kanäle --> = 20

CPU 573 --> 10 Kanäle --> =100

Verhalten bis SW-Stand 3

Beim Erreichen eines WAITM()-Aufrufes werden die Achsen des aktuellen Kanals gebremst, und es wird auf das Eintreffen der im Aufruf angegebenen Markennummer aus den anderen zu synchronisierenden Kanälen gewartet. Wenn diese beim Erreichen ihres WAITM()-Befehls ebenfalls gebremst sind, herrscht Synchronität. Die synchronisierten Kanäle werden fortgesetzt.

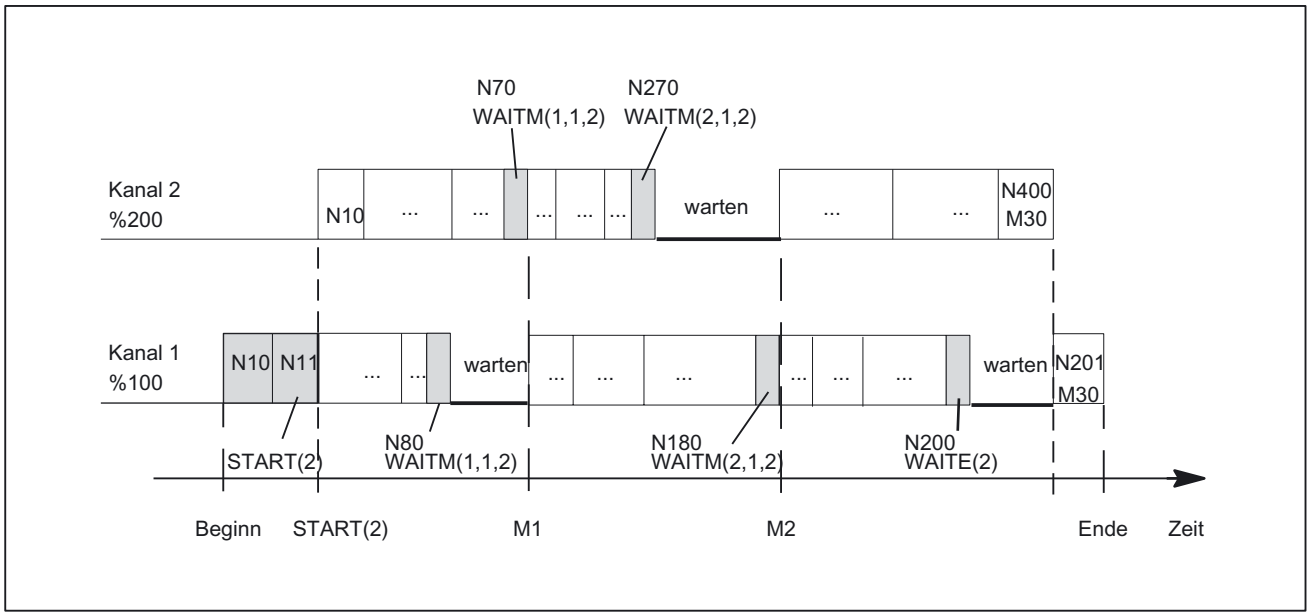


Bild 6-1 Programmabläufe am Beispiel, Koordinierung mit WAITM(), unbedingtes Warten

Beispiel für Programmkoordinierung

Kanal 1:

```
%100
N10 INIT(2,"_N_200_MPF","n")
N11 START(2)
.                                   ; Bearbeiten im Kanal 1

N80 WAITM(1,1,2)                    ; Warten auf WAIT-Marke 1 im Kanal 1 und im
                                    Kanal 2

.                                   ; weiteres Bearbeiten in Kanal 1

N180 WAITM(2,1,2)                   ; Warten auf WAIT-Marke 2 im Kanal 1 und im
                                    Kanal 2

.                                   ; weiteres Bearbeiten im Kanal 1
N200 WAITE(2)                       ; Warten auf Programmende des Kanals 2
N201 M30                            ; Programmende Kanal 1, Gesamtende
.
.
```

Kanal 2:

```
%200
.                                   ; Bearbeiten im Kanal 2

N70 WAITM(1,1,2)                    ; Warten auf WAIT-Marke 1 im Kanal 1 und im
                                    Kanal 2

.                                   ; weiteres Bearbeiten im Kanal 2

N270 WAITM(2,1,2)                   ; Warten auf WAIT-Marke 2 im Kanal 1 und im
                                    Kanal 2

.                                   ; weiteres Bearbeiten im Kanal 2

N400 M30                            ; Programmende des Kanals 2
.
.
```

6.3.2 Bedingtes Warten im Bahnsteuerbetrieb WAITMC

Zielsetzung

Es soll nur dann gebremst und gewartet werden, wenn noch nicht alle zu koordinierenden Kanäle ihre Markennummern für eine Synchronisation gesetzt haben. Bedingtes Warten.

Die Zeitpunkte für die Erzeugung von Wartemarken und die bedingten Warteaufrufe sind entkoppelt.

Marken können zur Verständigung zwischen Kanälen auch dann gesetzt werden, wenn Warten und Bremsen überhaupt nicht beabsichtigt ist. Kein WAITMC()-Befehl. In diesem Fall behalten die Marken der Kanäle über RESET und NC-Start hinweg ihre Werte.

Voraussetzungen für bedingtes Warten

Um das bedingte Warten mit WAITMC() mit reduzierten Wartezeiten nutzen zu können, muss:

- Bahnsteuerbetrieb G64 eingestellt sein
- die Look Ahead-Funktion aktiv sein
- Genauhalt (G60, G09) **nicht** eingestellt sein.

  Ist Genauhalt angewählt, entspricht das Warten mit WAITMC() dem Warten mit WAITM() aus SW-Stufe 3.

Verhalten

A) Beginnend mit dem Bewegungssatz vor dem Aufruf WAITMC() werden die Wartemarken der anderen zu synchronisierenden Kanäle geprüft. Liegen diese bereits alle vor, so wird ungebremst weiter mit Bahngeschwindigkeit gearbeitet. Kein Warten. Die Bahngeschwindigkeit wird aufrechterhalten.

B) Fehlt wenigstens eine Wartemarke eines zu synchronisierenden Kanals, so wird von Bahngeschwindigkeit nach Genauhaltgeschwindigkeit zu bremsen begonnen. In jedem Interpolationstakt wird nun geprüft, ob die noch fehlenden Wartemarken der zu koordinierenden Kanäle inzwischen eingetroffen sind. Ist dies der Fall, so wird wieder auf Bahngeschwindigkeit beschleunigt und weitergearbeitet.

C) Sind bis zum Erreichen der Genauhaltgeschwindigkeit die erwarteten Marken der zu synchronisierenden Kanäle nicht eingetroffen, kommt die Bearbeitung bis zum Eintreffen der fehlenden Marken zum Stillstand. Aus dem Stillstand wird beim Eintreffen der letzten erwarteten Marke wieder auf Bahngeschwindigkeit beschleunigt.

Die folgende Tabelle zeigt die Abläufe für die Fälle A) - C):

Bremsverhalten bei bedingtem Warten mit WAITMC()		
bei WAITMC	**Verhalten**	**Geschwindigkeitsverlauf**
A) Wartemarken aller Kanäle bereits vorhanden	ungebremste Weiterarbeit	
B) alle Wartemarken kommen während des Bremsens von Bahngeschwindigkeit auf Genauhaltgeschwindigkeit	Das Bremsen wird in dem Moment abgebrochen, wenn die letzte Wartemarke da ist. Es wird wieder auf Bahngeschwindigkeit beschleunigt.	
C) die letzte Wartemarke kommt nach dem Erreichen der Genauhaltgeschwindigkeit	Bremsen bis zu Genauhaltgeschwindigkeit. Nach Eintreffen der letzten Wartemarke wird aus der Genauhaltgeschwindigkeit auf die Bahngeschwindigkeit beschleunigt	

Erweitertes Verhalten und Satzwechsel beim Eintreffen WAITMC

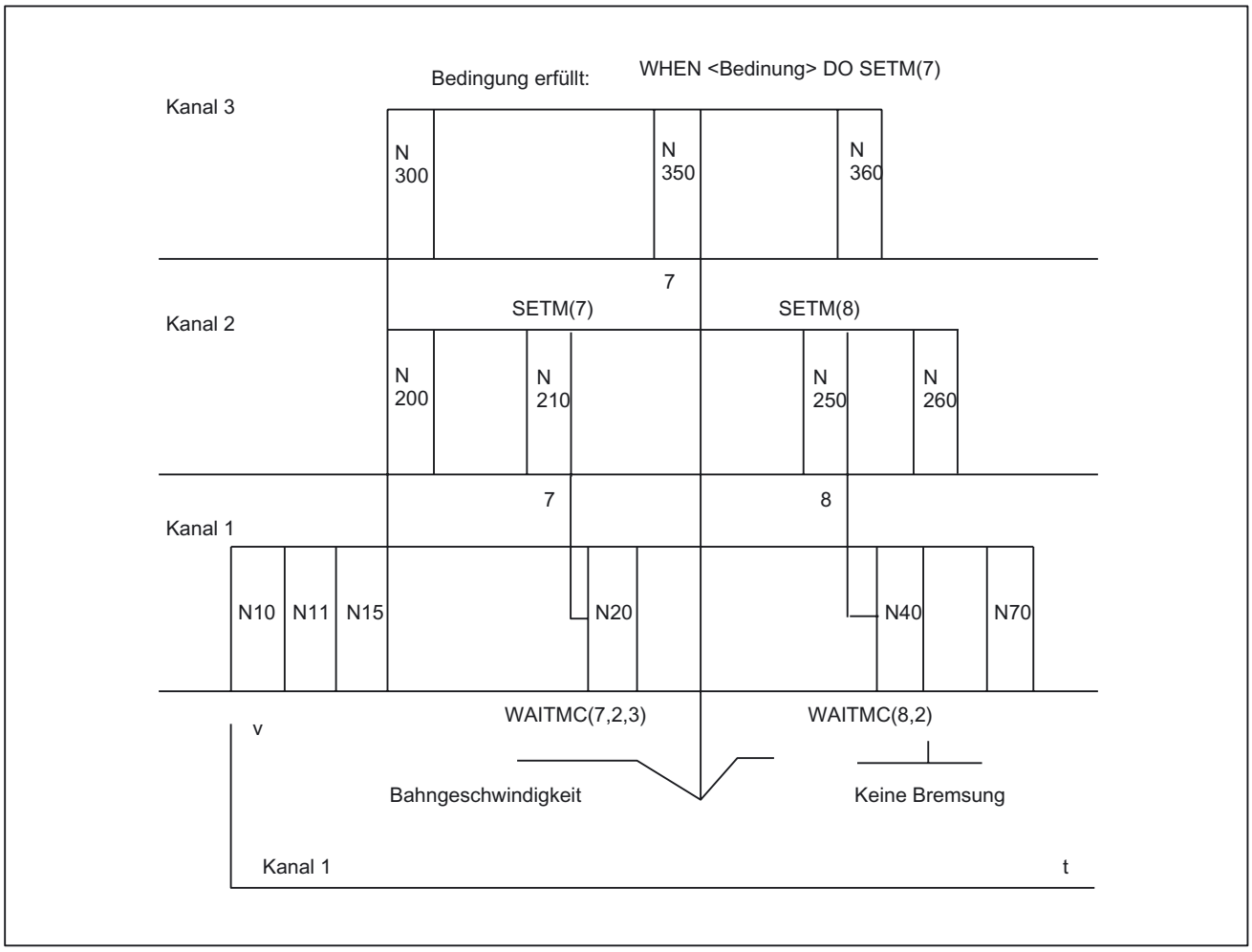
Es können WAITMC und SETM (in Synact) synchronisiert werden.

Hinweis

Ein Satz WAITMC(1,2,3) erzeugt bei aktiven 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.

Bei dem Satzwechselkriterium IPOBRKA wird bei Eintreffen der Wait-Marke augenblicklich der nächste Satz eingewechselt und die Achsen durchgestartet, falls nicht ein anderes Satzendekriterium den Satzwechsel verhindert. Es wird nur abgebremst, wenn die Marke noch nicht erreicht ist oder ein anderes Satzendekriterium den Satzwechsel verhindert.

Beispiel für bedingtes Warten im Bahnsteuerbetrieb



Bedingtes Warten mit drei beteiligten Kanälen (schematisch)

Das Beispiel ist schematisch und zeigt nur die synchronisationsbedeutsamen Befehle.

Kanal 1:

```
%100
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
...                                      ; 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:

```
%200
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:

```
%300
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
```

Beispiel für WAITMC und Einlesesperre

M555 wird in Kanal 3 während des Fahrens ausgegeben und erzeugt eine Einlesesperre (ELSP). Da das WAITMC dem SATZ N312 hinzufügt wird, ist die Waitmarke gesetzt und der Kanal 2 fährt weiter.

Kanal 2:

```
N112 G18 G64 X200 Z200 F567             ; Bearbeitung in Kanal 2
N120 WAITMC(1,2,3)                      ; Kanal 2 setzt Waitmarke 1, 2 und 3
...                                     ; 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:

```
                                        ; während des Fahrens Einlesesperre M555
N300                                    ; Bearbeitung in Kanal 3
N312 G18 G64 D1 X180 Z300 M555
N320 WAITMC(1,2,3)                      ; warten wegen ELSP
```

Waitmarke 1 ist in Kanal 2 und Kanal 3 gesetzt

Der Kanal 2 führt die weitere Bearbeitung fort und die Programmbearbeitung im Kanal 3 bleibt wegen der Einlesesperre stehen.

Dieses Verhalten kann auf alle verfügbaren Kanäle übertragen werden.

6.4 Achs-/Spindeltausch

6.4.1 Einführung

Allgemeines

Eine Achse/Spindel wird über Maschinendatum einem bestimmten Kanal fest zugeordnet. Die Achse/Spindel kann dann nur in diesem Kanal benutzt werden.

Definition

Mit der Funktion "Achs- bzw. Spindeltausch" ist es möglich, eine Achse bzw. Spindel freizugeben und einem anderen Kanal zuzuordnen, d. h. die Achse/Spindel zu tauschen.

Da die Spindelfunktion der Achsfunktion untergeordnet ist, wird im Folgenden nur noch der Begriff "Achstausch" verwendet.

Achstypen

Es wird kanalspezifisch zwischen vier verschiedenen Achstypen unterschieden. Die Reaktionen bei Achswechsel hängen vom folgenden Maschinendatum ab:

MD30552 $MA_AUTO_GET_TYPE

Kanal-Achse

Eine Kanal-Achse kann im Teileprogramm programmiert und in allen Betriebsarten verfahren werden.

PLC-Achse

Eine PLC-Achse kann nur durch die PLC positioniert werden.

Wird eine PLC-Achse im Teileprogramm programmiert, wird:

- wenn MD AUTO_GET_TYPE = 0 ist, ein Alarm ausgegeben.
- wenn MD AUTO_GET_TYPE = 1 ist, ein automatisches GET erzeugt.
- wenn MD AUTO_GET_TYPE = 2 ist, ein automatisches GETD erzeugt.

Neutrale Achse

Wird eine neutrale Achse im Teileprogramm programmiert, wird:

- wenn MD AUTO_GET_TYPE = 0 ist, ein Alarm ausgegeben.
- wenn MD AUTO_GET_TYPE = 1 ist, ein automatisches GET erzeugt.
- wenn MD AUTO_GET_TYPE = 2 ist, ein automatisches GETD erzeugt.

Achse in anderem Kanal

Dies ist eigentlich kein richtiger Achstyp. Es ist der interne Zustand einer tauschbaren Achse, wenn sie gerade in einem anderen Kanal aktiv ist (als Kanal-, PLC- oder neutrale Achse).

Wird eine Achse im anderen Kanal im Teileprogramm programmiert, wird:

- wenn MD AUTO_GET_TYPE = 0 ist, ein Alarm ausgegeben.
- wenn MD AUTO_GET_TYPE = 1 ist, ein automatisches GET erzeugt.
- wenn MD AUTO_GET_TYPE = 2 ist, ein automatisches GETD erzeugt.

Hinweis

Die beiden Maschinendaten:

MD20110 $MC_RESET_MODE_MASK

MD20112 $MC_START_MODE_MASK

steuern das Verhalten der Achszuordnungen im RESET, Hochlauf und Teileprogrammstart. Die Einstellungen für Kanäle, zwischen denen Achstausch beabsichtigt ist, müssen so gewählt werden, dass in Verbindung mit folgendem Maschinendatum keine unverträglichen Konstellationen (Alarme) entstehen:

MD30552 $MA_AUTO_GET_TYPE

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; " ... Werkstücknahes Istwertsystem" (K2)

Voraussetzungen

Damit ein Achstausch durchgeführt werden kann, muss der für die Maschinenachsnummer gültige Kanal festgelegt werden über das **kanalspezifische** Maschinendatum

MD20070 $MC_AXCONF_MACHAX_USED (Maschinenachsnummer gültig im Kanal)

und die Löschstellung des Kanals für den Achstausch über das **achsspezifische** MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN (Kanal für Achswechsel).

Daraus ergeben sich folgende Festlegungen:

1. In welchem Kanal kann die Achse benutzt und getauscht werden?
2. Welchem Kanal soll die Achse bei POWER ON zugeordnet werden?

Beispiel eines Achstausches zwischen den Kanälen

Bei 6 Achsen und 2 Kanälen soll die 1., 2., 3. und 4. Achse im Kanal 1 und die 5. und 6. Achse im Kanal 2 benutzt werden. Die 1. Achse soll getauscht werden können und nach POWER ON dem Kanal 2 zugeordnet sein.

Das **kanalspezifische** Maschinendatum muss belegt werden mit:

CHANDATA(1)
MD20070 $MC_AXCONF_MACHAX_USED=(**1**, 2, 3, 4, 0, 0, 0, 0)

CHANDATA(2)
MD20070 $MC_AXCONF_MACHAX_USED=(5, 6, **1**, 0, 0, 0, 0, 0)

Das **achsspezifische** Maschinendatum muss belegt werden mit:

MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN[AX1]=2

Anzeige

Der aktuelle Achstyp und der aktuell zuständige Kanal für diese Achse werden in einem axialen PLC-Nahtstellenbyte angezeigt. Siehe Abschnitt "Achstausch durch die PLC".

Hinweis

Wenn eine Achse in dem angewählten Kanal nicht gültig ist, wird dies durch Invertierung des Achsnamens an der Bedientafelfront von HMI angezeigt.

6.4.2 Beispiel eines Achstausches

Annahmen

Bei 6 Achsen und 2 Kanälen soll die 1., 2., 3. und 4. Achse im Kanal 1 und die 5. und 6. Achse im Kanal 2 benutzt werden. Die 2. Achse soll zwischen den Kanälen getauscht werden können und nach POWER ON dem Kanal 1 zugeordnet sein.

Aufgabe

Die Aufgabe teilt sich in folgende Bereiche:

- Maschinendaten belegen, so dass die Voraussetzungen für einen Achstausch vorhanden sind.
- Programmieren eines Achstausches zwischen Kanal 1 und Kanal 2.

Realisierung der Voraussetzungen

Belegung des kanalspezifischen Maschinendatums MD20070

$MC_AXCONF_MACHAX_USED[1]=(1, **2**, 3, 4, 0, 0, 0, 0)

$MC_AXCONF_MACHAX_USED[2]=(5, 6, **2**, 0, 0, 0, 0, 0)

Belegung des achsspezifischen Maschinendatums:

MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN[AX2]=1

```
Programm im Kanal 1
...
RELEASE (AX2)
        ; Freigeben der Achse AX2
INIT (2, "_N_MPF_DIR\_N_TAUSH2_MPF", "S")
        ; Anwahl des Programms TAUSH2 im Kanal 2
START (2)
        ; Starten des Programms im Kanal 2
WAITM (1,1,2)
        ; Warten auf Wait–Marke 1 im Kanal 1 und 2
...     ;
...     ; Weiterer Ablauf nach dem Achstausch
...     ;
...     ;
M30
```

```
Programm TAUSH2 im Kanal 2
...
WAITM (1,1,2)
        ; Warten auf Wait–Marke 1 im Kanal 1 und 2
GET (AX2)
...     ; Übernehmen der Achse AX2
...     ;
...     ; Weiterer Ablauf nach dem Achstausch
...     ;
RELEASE (AX2)
...     ; Freigeben für weiteren Achstausch
...     ;
M30
```

6.4.3 Achstausch Möglichkeiten

Eine Achse-/Spindel oder auch mehrere können für einen Tausch zwischen den Kanälen sowohl vom Teileprogramm als auch über Bewegungssynchronaktionen aktiviert werden. Es kann auch ein Achs-/Spindeltausch vom PLC aus über die VDI–Nahtstelle angefordert und freigegeben werden. Die Achse-/Spindel muss im aktuellen Kanal freigegeben sein und wird bei einer Anforderung mit GET in dem anderen Kanal übernommen und mit RELEASE freigegeben.

Bei Erfüllung der angegebenen Voraussetzungen wird ein Achs-/Spindeltausch eingeleitet durch:

- Programmierung im Teileprogramm GET/GETD

  Eine Achse bzw. Spindel aus einem anderen Kanal mit GET übernehmen oder direkt mit GETD aus einem anderen Kanal holen. Ein passendes RELEASE ist bei GETD nicht erforderlich.

- Automatisch durch Programmierung des Achsnamens, wenn die Voraussetzung hierfür mit MD30552 $MA_AUTO_GET_TYPE > 0 erfüllt ist.

- Ohne Vorlaufstopp und gegebener Synchronisation zwischen Vorlauf und Hauptlauf.

- Durch PLC über die VDI–Nahtstelle zum NCK.

Bei der Übernahme einer PLC kontrollierten Achse kann das vom NC-Programmablauf getriggerte Kanalverhalten über ein Nahtstellensignal entkoppelt werden. Dies ermöglicht z.B. einzelne PLC-Achsen unabhängig vom NC-Programm zu interpolieren.

Literaratur:

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)
Kapitel "Autarke Einzelachsvorgänge".

Achstauscherweiterungen

- Achstauschverhalten über das Maschinendatum MD10722 $MN_AXCHANGE_MASK veränderbar einstellen.
- Achstausch bei einer Achs–Containerdrehung mit impliziten GET/GETD
- Achstausch ohne Vorlaufstopp der nicht an der Kontur beteiligten Achsen
- Geometrieachse mit gedrehten WKS und Achstausch in der Betriebsart JOG über das Maschinendatum MD32074 $MA_FRAME_OR_CORRPOS_NOTALLOWED aktivierbar.
- Achstausch über Synchronaktionen GET(Achse), RELEASE(Achse), AXTOCHAN, $AA_AXCHANGE_TYP(Achse).

6.4.4 Achstauschverhalten NC-Programm

Mögliche Übergänge

Das folgende Bild zeigt, welche Achstauschmöglichkeiten bestehen.

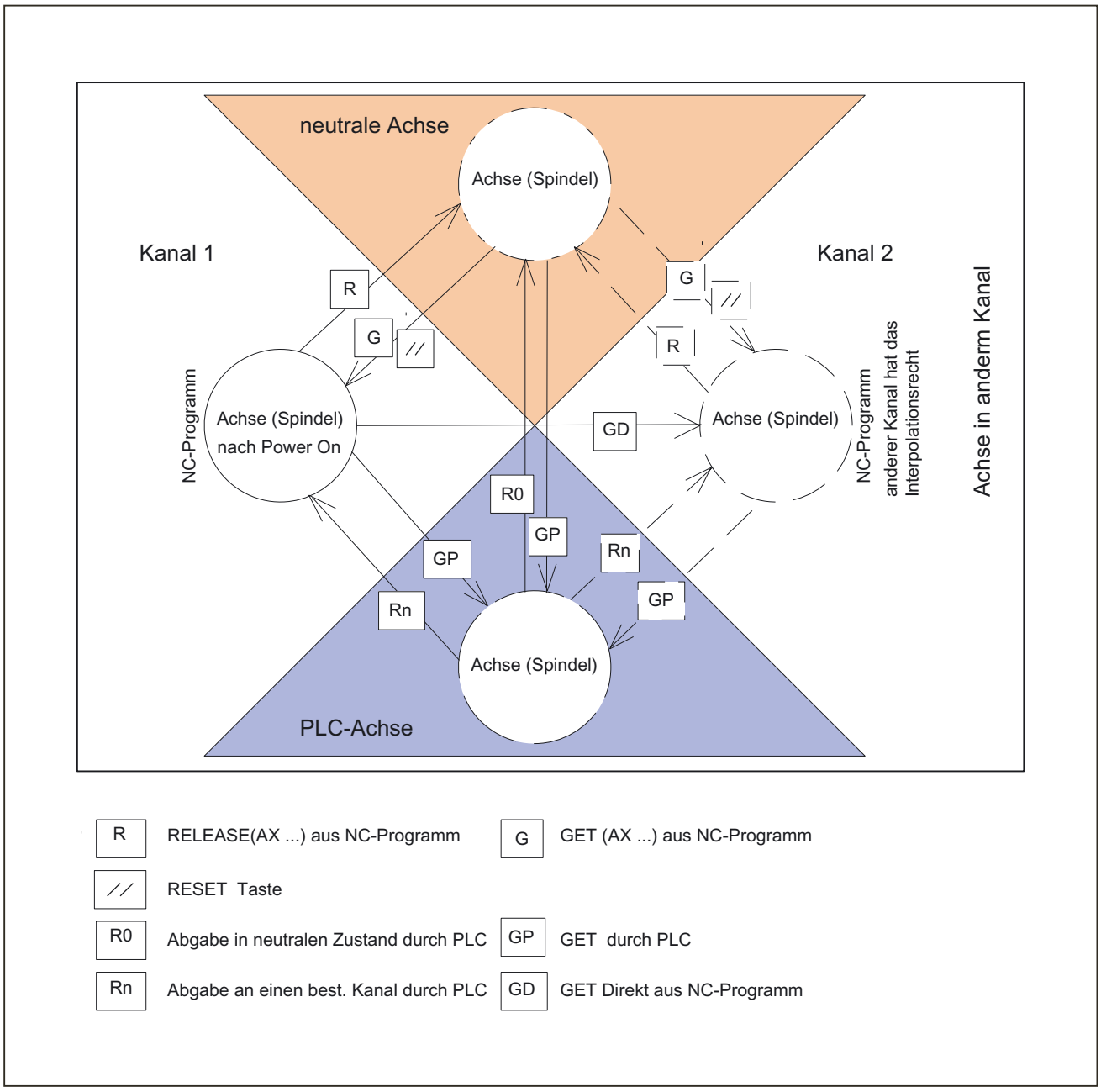


Bild 6-2 Übergänge möglicher Achszustände bei einem Achstausch

6.4.5 Achse in den neutralen Zustand überführen (abgeben)

RELEASE

Notation im Teileprogramm:

RELEASE(Achsname, Achsname, SPI (Spindelnr.), ...)

Hinweis

Der Achsname entspricht den Achszuordnungen im System und ist entweder

- AX1, AX2, AX3, ... **oder**
- der über folgendes Maschinendatum zugeordnete Name:
  MD10000 $MN_AXCONF_MACHAX_NAME_TAB

Mit RELEASE (Achsname, ...) wird immer ein eigener NC-Satz erzeugt.

Ausnahme: Die Achse befindet sich schon im neutralen Zustand.

Der Befehl RELEASE wird abgebrochen, wenn

- die Voraussetzungen zum Achstausch nicht erfüllt sind
  (MD20070 $MC_AXCONF_MACHAX_USED und
  MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN)
- die Achse an einer Transformation beteiligt ist
- die Achse sich in einem Achsverbund befindet.

Hinweis

Wenn der Befehl RELEASE auf eine Gantry-Masterachse angewendet wird, dann werden auch alle Folgeachsen freigegeben.

Wenn ...	und ...	dann ...
die Achse freigegeben, aber noch nicht mit GET übernommen ist ...	... es erfolgt ein RESET über die Bedientafelfront, ...	... wird die Achse wieder dem zuletzt verantwortlichen Kanal zugeordnet.

6.4.6 Achse oder Spindel im Teileprogramm übernehmen

Möglichkeiten

Der Abgabezeitpunkt und das Verhalten eines Achs– oder Spindeltauschs wird im Teileprogramm wie folgt beeinflusst:

– Programmierung mit dem Befehl GET im selben Kanal.

– Direkt aus einem anderen Kanal durch Programmierung mit GETD.

Literatur:
/PGA/ Programmierhandbuch Arbeitsvorbereitung; Kapitel "Achstausch, Spindeltausch (RELEASE, GET, GETD)"

Mit Befehl GET

GET (Achsname, Achsname, SPI (Spindelnr.), ...)

Die Übernahme einer Achse wird verzögert, wenn

- die Achse gerade das Messsystem wechselt.
- für die Achse die Reglerfreigabe bearbeitet wird (Übergang von Regeln in Nachführen/Halten und umgekehrt).
- die Achs-/Spindelsperre ansteht.
- die Achse von einem anderen Kanal aus mit RELEASE noch nicht freigegeben ist.
- für die Achse die Interpolation noch nicht abgeschlossen ist (außer bei drehzahlgeregelter Spindel).

Mit GET (Achsname, ...) wird immer ein eigener NC-Satz mit Vorlaufstopp erzeugt.

Ausnahmen:

- Wenn die Achse schon eine Kanalachse ist, wird kein Satz erzeugt.
- Wenn die Achse noch synchron ist, (d. h. sie wurde zwischenzeitlich nicht in einen anderen Kanal getauscht, bzw. von der PLC beaufschlagt) wird ebenfalls kein extra Satz erzeugt.

Mit Befehl GETD

Mit **GETD** (GET Directly) wird eine Achse direkt aus einem anderen Kanal geholt. Das bedeutet, dass zu diesem GETD kein passendes RELEASE in einem anderen Kanal programmiert sein muss. Das bedeutet aber auch, dass jetzt eine andere Kanalkommunikation aufgebaut werden muss (z. B. Waitmarken), da mit GETD der abgebende Kanal unterbrochen wird! Falls die Achse eine PLC-Achse ist, wird der Tausch so lange verzögert, bis die PLC die Achse freigegeben hat.

⚠ VORSICHT
Mit diesem Programmierbefehl wird der Programmablauf in dem Kanal, in welchem sich die gewünschte Achse zurzeit befindet, unterbrochen! (REORG). Ausnahme: - die Achse befindet sich zu dem Zeitpunkt im neutralen Zustand.

Hinweis

Wenn der Befehl GET oder GETD programmiert wurde, die Übernahme verzögert wird und es erfolgt ein RESET in dem Kanal, dann versucht der Kanal nicht länger, die Achse zu übernehmen.

Eine mit GET übernommene Achse bleibt auch nach einem Tasten-RESET oder Programm-RESET diesem Kanal zugeordnet. Die Achse kann durch erneutes Programmieren von RELEASE und GET getauscht werden oder wird bei POWER ON dem im folgenden Maschinendatum definierten Kanal zugeordnet:

MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN

6.4.7 Automatischer Achstausch

Automatisch durch Programmieren des Achsnamens

Abhängig vom Maschinendatum MD30552 $MA_AUTO_GET_TYPE wird automatisch ein GET oder GETD erzeugt, wenn eine neutrale Achse erneut programmiert wird bzw. die Achse einem anderen Kanal zugeordnet ist.

Voraussetzung für den automatischen Achstausch

MD30552 $MA_AUTO_GET_TYPE > 0 muss für den automatische Achstausch erfüllt sein.

Automatisches GETD

Hinweis

Falls ein automatisches GETD eingestellt ist, muss folgendes beachtet werden.

- Die Kanäle könnten sich gegenseitig beeinflussen.
  (REORG, wenn Achse weggenommen wird.)
- Bei gleichzeitigem Zugriff mehrerer Kanäle auf eine Achse ist nicht vorhersehbar, welcher Kanal die Achse am Ende behält.

Beispiel 1

```
N1 M3 S1000
N2 RELEASE (SPI(1))                    ; => Übergang in neutralen Zustand
N3 S3000                               ; Neue Drehzahl für abgegebene Spindel
                                       ; MD AUTO_GET_TYPE =
                                       ; 0 =>Alarm "Falscher Achstyp" wird
                                       ausgegeben
                                       ; 1 => GET (SPI(1)) wird erzeugt.
                                       ; 2 => GETD (SPI(1)) wird erzeugt.
```

Beispiel 2

```
                                       ; (Achse 1 = X)
N1 RELEASE (AX1)                       ; => Übergang in neutralen Zustand
N2 G04 F2
N3 G0 X100 Y100:                       ; Bewegung der abgegebenen Achse
                                       ; MD AUTO_GET_TYPE =
                                       ; 0 =>Alarm "Falscher Achstyp" wird
                                       ausgegeben
                                       ; 1 => GET (AX1) wird erzeugt.
                                       ; 2 => GETD (AX1) wird erzeugt.
```

Beispiel 3

```
                                       ; (Achse 1 = X)
N1 RELEASE (AX1)                       ; => Übergang in neutralen Zustand
N2 G04 F2
N3 POS (X) = 100:                      ; Positionierung der abgegebenen Achse:
                                       ; MD AUTO_GET_TYPE =
                                       ; 0 =>Alarm "Falscher Achstyp" wird
                                       ausgegeben
                                       ; 1 => GET (AX1) wird erzeugt. *)
                                       ; 2 => GETD (AX1) wird erzeugt. *)
```

*) Sofern die Achse noch synchronisiert ist, wird kein eigener Satz erzeugt.

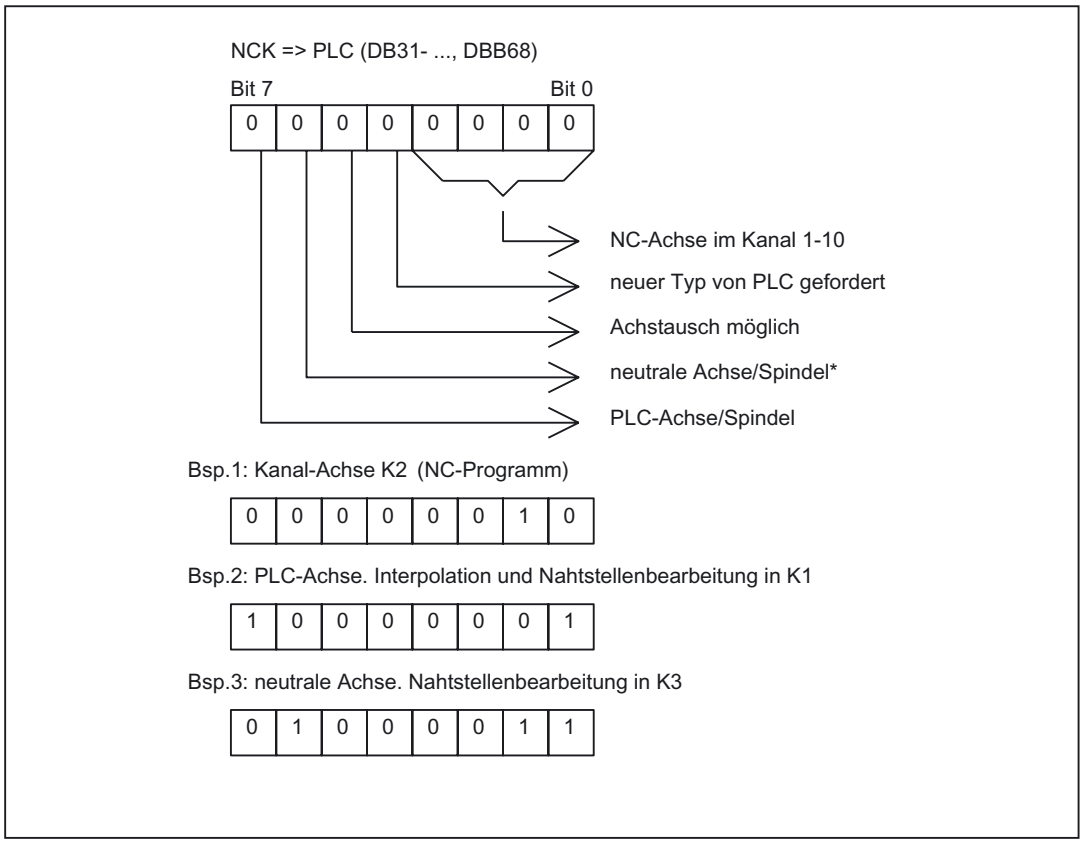
6.4.8 Achstausch durch PLC

Die PLC kann zu jeder Zeit und in jeder Betriebsart eine Achse anfordern und verfahren.

Die PLC kann eine Achse von einem Kanal in den anderen wechseln (nur bei 840D sl).

TYP-Anzeige

Der Typ einer Achse kann über ein Nahtstellenbyte jederzeit bestimmt werden (PLC-Achse, Kanal-Achse, neutrale Achse).



*** neutrale Achse/Spindel** beinhaltet auch die **Kommando-/Pendel-Achse**

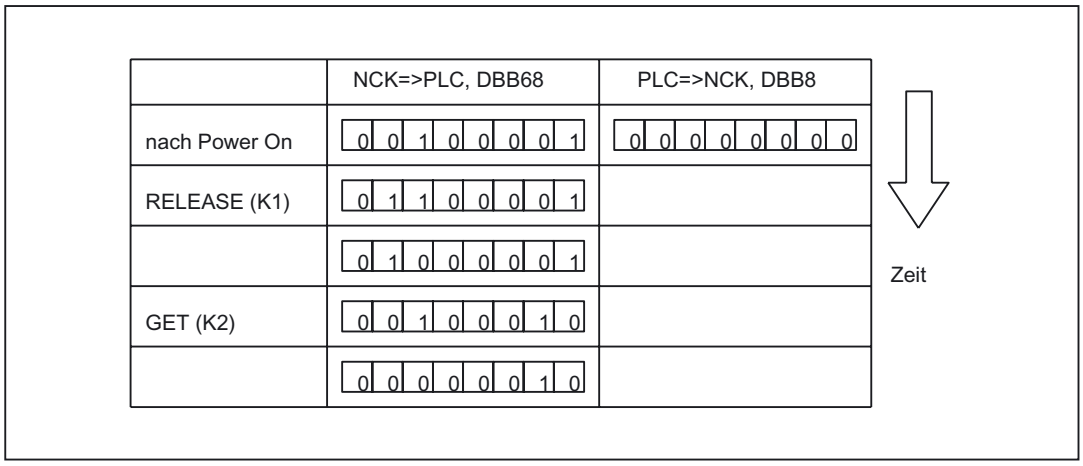
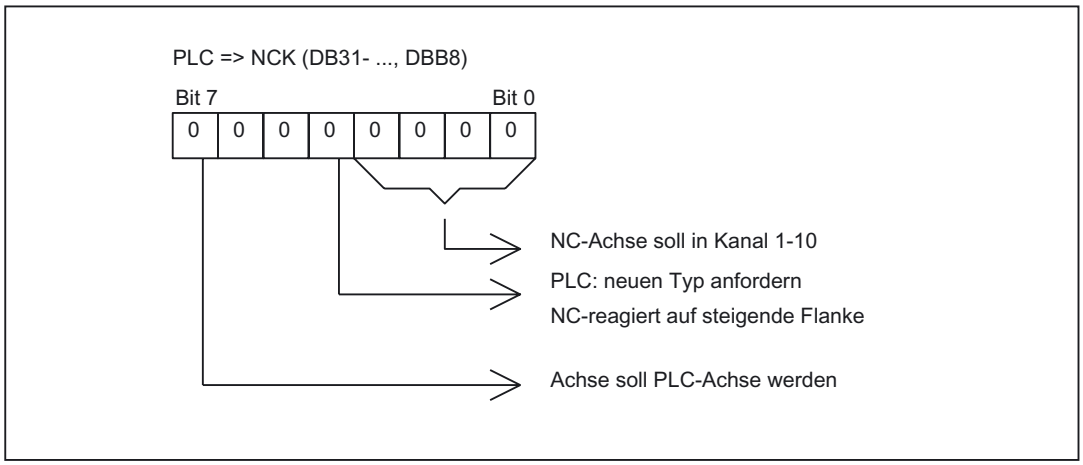


Bild 6-3 Tausch einer Achse aus K1 nach K2 durch Teileprogramm

TYP-Vorgabe



Das Signal "neuen Typ anfordern" muss die PLC prinzipiell setzen. Es ist nach dem Wechsel wieder gelöscht. Auch bei einem Kanaltausch mit GET und RELEASE (nur 840D sl).

Steuerung von PLC-Achsen/-Spindeln bei 840D sl

PLC-Achsen und PLC-Spindeln werden über den Funktionsbaustein FC18 im PLC-Grundprogramm verfahren

FC18: SpinCtrl Spindelsteuerung

Beispiele

Die folgenden Abbildungen zeigen die Folgen von NC/PLC-Nahtstellensignalen für den Wechsel einer NC-Achse zur PLC-Achse und die Überführung einer NC-Achse in eine neutrale Achse durch die PLC.

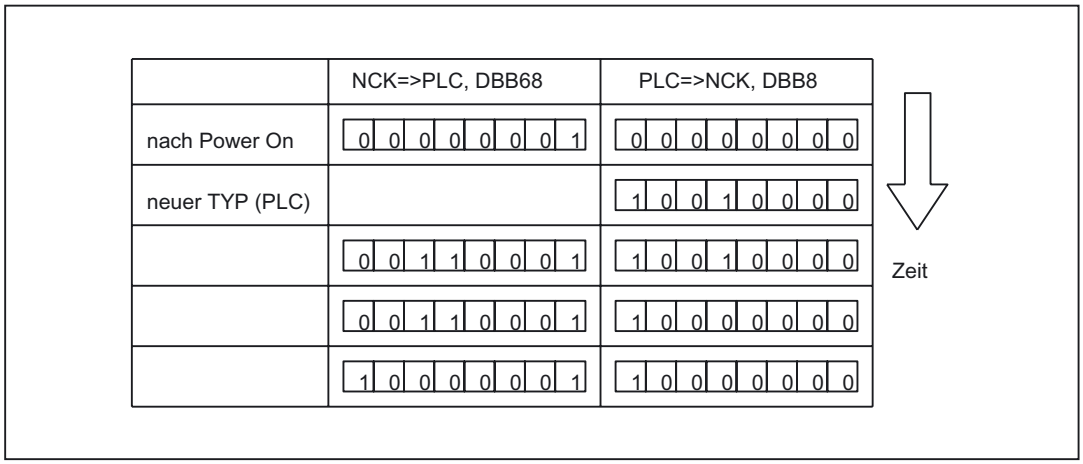


Bild 6-4 Wechsel einer NC-Achse in eine PLC-Achse

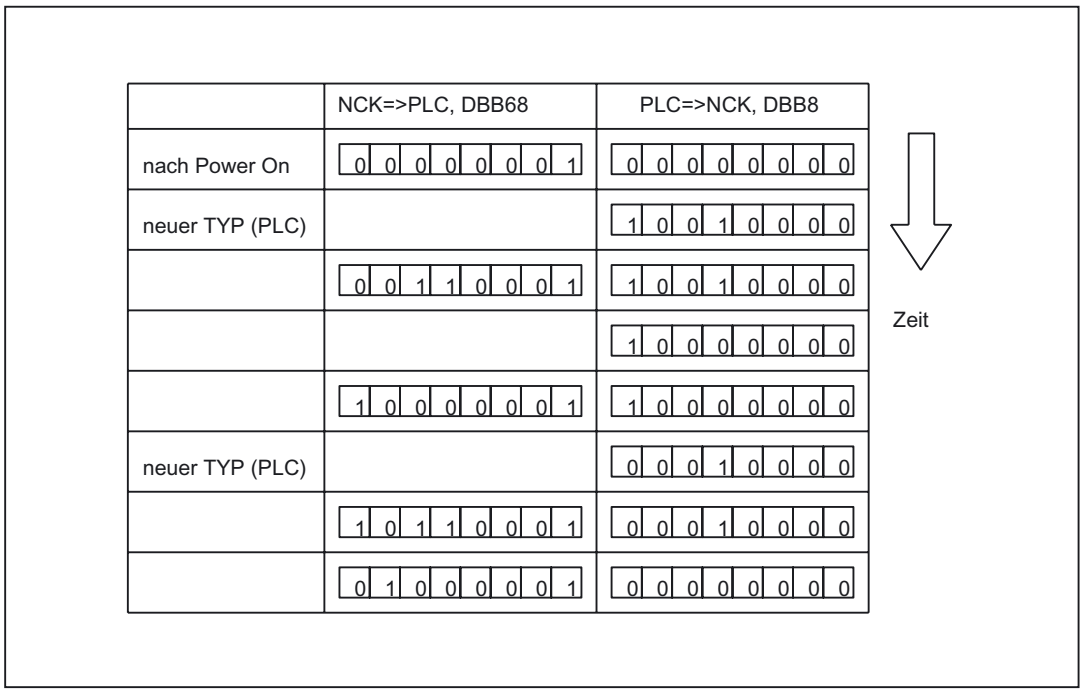


Bild 6-5 Wechsel einer NC-Achse in eine neutrale Achse durch die PLC

6.4.9 Achstauschverhalten veränderbar einstellen

Die Achse wird im aktuellen Kanal getauscht und das Achstauschverhalten kann abhängig vom entsprechenden Achsentyp über das Maschinendatum MD10722 $MN_AXCHANGE_MASK beeinflusst werden:

Tabelle 6- 2 Abgabezeitpunkt Achsen bzw. Spindeln beim Achstausch

MD10722	Achstauschverhalten
Bit 0 = 1	**Automatischer Achstausch** findet zwischen zwei Kanälen auch dann statt, wenn die Achse durch WAITP in einen neutralen Zustand gebracht wurde.
Bit 1 = 1	Freigabe über **Achs-Containerdrehung** mittels implizit erzeugtem GET/GETD Bei der Anforderung einer Achs-Containerdrehung werden alle dem ausführenden Kanal zuordenbaren Achsen des Achs-Containers mittels impliziten GET bzw. GETD in den Kanal geholt. Ein anschließender Achstausch ist erst nach dem Abschluss der Achs-Containerdrehung wieder erlaubt.
Bit 2 = 1	**Achstausch ohne Vorlaufstopp** und möglicher Zwangsreorganisierung der nicht an der Kontur beteiligten Achsen. Nach einem eingeschobenen Zwischensatz im Hauptlauf wird geprüft, ob ein Reorganisieren erforderlich ist, oder nicht. Nur wenn die Achszustände dieses Satzes mit den aktuellen Achszuständen **nicht** übereinstimmen, ist ein Reorganisieren erforderlich.
Bit 3 = 0	Achstausch vom PLC für fest **zugeordnete PLC-Achse** Für jede Achse ein Achstausch vom PLC angefordert werden kann. Die fest zugeordnete PLC-Achse nur von neutraler Achse zu PLC Achse und umgekehrt.
Bit 3 = 1	Achstausch Anforderung über VDI–Nahtstelle Eine Achstausch Anforderung über die VDI–Nahtstelle wird nur ausgeführt für eine: **ausschließlich von der PLC kontrollierte Achse** bei MD30460 $MA_BASE_FUNCTION_MASK mit Bit4=1. **fest zugeordnete** konkurrierende Positionierachse (PLC-Achse) bei MD30460 $MA_BASE_FUNCTION_MASK mit Bit5=1. Für solche Achsen ist das Nahtstellensignal NCK → PLC DB31, ... DBX68.5 (Achstausch möglich) immer auf Eins gesetzt. Bei allen anderen Achsen ist dieses Bit auf Null gesetzt. Für die fest zugeordnete PLC-Achse ist nur ein Achstausch von neutraler Achse zur PLC-Achse und umgekehrt möglich.

6.4.10 Achstausch über Achs-Containerdrehung

Achs-Containerdrehung freigeben

Bei der Freigabe einer Achs-Containerdrehung werden alle dem Kanal zuordenbaren Achs-Container-Achsen mittels implizit erzeugtem GET oder GETD dem Kanal zugeordnet. Eine Abgabe der Achsen wird erst nach der Achs-Container-Rotation wieder erlaubt.

Hinweis

Dieses Verhalten kann **nicht** angewendet werden, wenn eine Achse im Zustand Hauptlauf-Achse (z. B. PLC-Achse) an einer Achs-Containerdrehung teilnehmen soll, da diese Achse diesen Zustand zur Achs-Containerdrehung verlassen müsste.

Literatur:

Weitere Erläuterungen zum Achstausch von Kanalachsen mit Achscontainer entnehmen Sie bitte:

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafelfronten und NCUs (B3)

Beispiel Achs-Container-Drehung mit implizitem GET oder GETD

Aktion Kanal 1	Aktion Kanal 2
	SPOS = 180 positioniert
axctswe(CT 1)	; holt Spindel in Kanal 1
	; und erlaubt Achs-Container-Drehung

Annahme:

Die Spindel wird in beiden Kanälen verwendet und ist auch Achse im Achs-Container CT 1.

Aktivierung

Der Achstausch mittels Achs-Containerdrehung und implizitem GET/GETD wird aktiviert mit Maschinendatum MD10722 $MN_AXCHANGE_MASK, Bit 1=1.

6.4.11 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

```
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.

6.4.12 Ausschließlich PLC-kontrollierte Achse und fest zugeordnete PLC-Achse

Ausschließlich PLC-kontrollierte Achse

Aktivierung

Die ausschließlich PLC-kontrollierte Achse wird aktiviert über:

MD30460 $MA_BASE_FUNCTION_MASK, Bit 4 = 1

Hinweis

Wird versucht, eine ausschließlich PLC-kontrollierte Achse dem NC-Programm zuzuordnen oder die Achse für das NC–Programm anzufordern, so wird dies mit dem Alarm 26075 abgelehnt. Analog wird für eine fest zugeordnete PLC-Achse der Alarm 26076 gemeldet.

Für eine ausschließlich PLC-kontrollierte Achse, die **nicht** eine fest zugeordnete PLC-Achse ist, sind folgende Funktionen erlaubt:

- JOG Bewegungen der Achse über Verfahrtasten bzw. Handrad
- Referenzieren der Achse
- Verfahren der Achse über den Funktionsbaustein **FC18** (bei 840D sl)
- Verfahren als fest zugeordnete PLC-Achse
- Verfahren als Pendelachse (Voraussetzung: Option "Pendeln" ist verfügbar.)

  Literatur:
  Funktionshandbuch Erweiterungsfunktionen; "Pendeln (P5)"

Achskontrolle

Im Zustand neutrale Achse kann die Achse durch die im Kanal wirksamen Signale AXRESET, AXSTOP und AXRESUME kontrolliert werden.

Fest zugeordnete PLC-Achse

Die fest zugeordnete PLC-Achse wird aktiviert über:

MD30460 $MA_BASE_FUNCTION_MASK, Bit 5 = 1

Die Achse wird beim Hochlauf zur neutralen Achse. Eine Verfahranforderung über die NC/PLC-Nahtstelle wandelt eine neutrale Achse ohne einen vorhergehenden Achstausch automatisch zur konkurrierenden Positionierachse (PLC-Achse).

Nach Abschluss der Verfahrbewegung, wenn z. B. der programmierte Endpunkt erreicht wird oder bei abgebrochener Verfahrbewegung, wird die Achse wieder zur neutralen Achse. In diesem Zustand kann die Achse über JOG mit den Verfahrtasten bzw. Handrad bewegt oder referenziert werden.

Hinweis

Ein Achstausch einer fest zugeordneten PLC-Achse ist nicht möglich.

Für eine fest zugeordnete PLC-Achse sind folgende Funktionen **nicht** möglich:

- Achstausch zwischen den Kanälen
- Achstausch zur Kommandoachse
- Achstausch zur Pendelachse
- Zuordnung der Achse zum NC–Programm

Ihre Eigenschaft als fest zugeordnete PLC-Achse meldet die Achse über das folgende NC/PLC-Nahtstellensignal zur PLC:

DB31, ... DBX71.7 (fest zugeordnete PLC-Achse)

Achskontrolle

Ist die Achse nicht PLC kontrolliert, so kann die Achse durch die im Kanal wirksamen Signale RESET, STOP und RESUME kontrolliert werden.

Wird im Hochlauf erkannt, dass eine als Geometrieachse definierte Achse gleichzeitig als fest zugeordnete PLC-Achse definiert wurde, wird die Achsfunktion mit Alarm 4320 "Achse %1 Funktion MA_BASE_FUNCTION_MASK Bit 5 und MC_AXCONF_GEOAX_ASSIGN_TAB nicht zugelassen" gemeldet.

Wenn ein Wechsel zur PLC-Achse nicht möglich ist

Bei einer Verfahranforderung aus dem NC/PLC-Nahtstellenbereich wird ein nicht möglicher Wechsel zur PLC-Achse mit dem Startsignal "Konkurrierende Positionierachse aktiv" an der NC/PLC-Nahtstelle quittiert.

Hinweis

Konkurrierende Positionierachsen bekommen ihre Satzendpunkte ausschließlich von der PLC und können zu jedem beliebigen Zeitpunkt gestartet werden. Positionierachsen werden nur an Satzgrenzen im laufenden Teileprogramm in ihrer Bearbeitung beeinflusst.

6.4.13 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 eine Kommandoachse über statische Synchronaktionen verfahren werden. Dazu muss im Maschinendatum MD32074 $MA_FRAME_OR_CORRPOS_NOTALLOWED das Bit 10=1 gesetzt sein. Das Repositionierverhalten dieser Achse wird über das Bit 11 beeinflusst.

Hinweis

Vor einen Betriebsartenwechsel in der Betriebsart JOG

Vor einem Wechsel in der 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 gemeldet. Dieser Alarm wird auch dann ausgelöst, 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 erlaubt.

Voraussetzung beim Wechsel von JOG nach AUTOMATIK

Beim Wechsel von der Betriebsart JOG nach AUTOMATIK wird im Zustand Programm unterbrochen der Endpunkt dieser Geometrieachsbewegungen nur übernommen, wenn im MD 32074: FRAME_OR_CORRPOS_NOTALLOWED das Bit 11=1 ist. 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
- 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, so werden alle anderen Achsen dieses Verbundes zu neutralen Achsen.

Randbedingungen

Ist im MD32074 $MA_FRAME_OR_CORRPOS_NOTALLOWED das Bit 10=0 gesetzt und im NC-Programm **ROT Z45** programmiert, dann ist für die X-Achse 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-Achse und Y-Achse an der VDI–Schnittstelle die Signale

NST DB31, ...DBX68.5 (”Achstausch möglich”) = 1
NST DB32, ...DBX68.5 (”Achstausch möglich”) = 1

aber es werden beide Signale auf Null zurückgesetzt.

Nur wenn im MD32074 $MA_FRAME_OR_CORRPOS_NOTALLOWED das Bit 10=1 gesetzt ist und kein Satz mit dieser Programmierung aktuell interpoliert, ist in der Betriebsart JOG ein Achstausch für solche Achsen möglich.

6.4.14 Achstausch aus Synchronaktionen

Funktion

Als Aktion einer Synchronaktion kann mit GET(Achse) eine Achse angefordert werden und mit RELEASE(Achse) für den Achstausch freigegeben werden.

Hinweis

Die Achse muss dem Kanal über Maschinendaten als Kanalachse zugeordnet sein.

Mit dem NC-Sprachbefehl AXTOCHAN kann über Synchronaktionen oder im Teileprogramm eine Achse zwischen den Kanälen direkt an einem 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 aus der Systemvariable $AA_AXCHANGE_TYP[Achse] gelesen werden:

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.

Die fest zugeordnete PLC Achse

im Zustand neutrale Achse $AA_AXCHANGE_TYP = 9 und
von der PLC kontrolliert, im Zustand neutrale Achse $AA_AXCHANGE_TYP = 10

wird **unabhängig von GET und RELEASE** fest dem PLC zugeordnet.

Ob die Achse auch getauscht werden kann, wird über die Systemvariable $AA_AXCHANGE_STAT[Achse] angezeigt.

Zustandsübergänge GET, RELEASE aus Synchronaktionen und wenn GET erfüllt ist

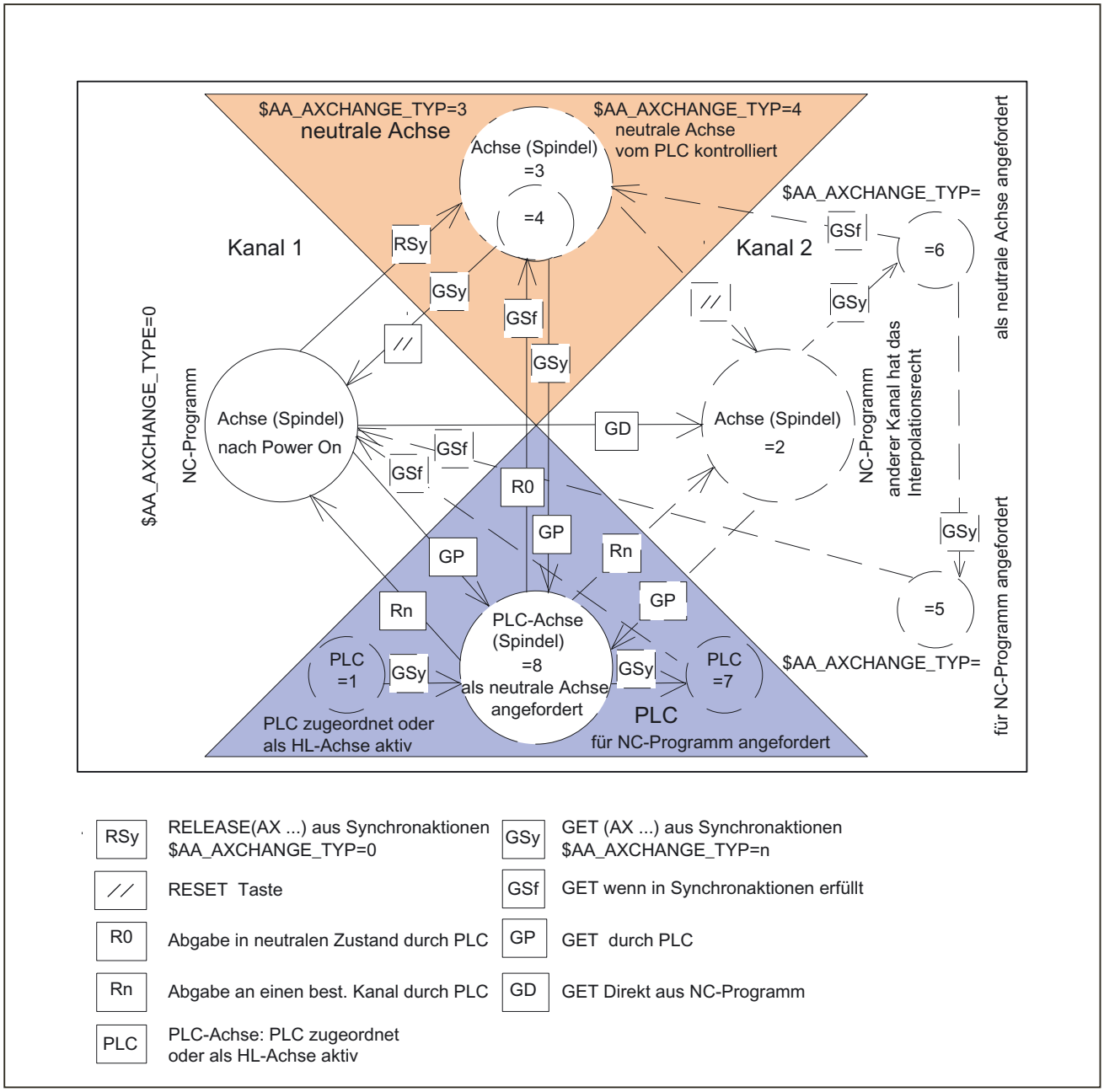


Bild 6-6 Übergänge aus Synchronaktionen

Weitere Informationen entnehmen Sie bitte:

Literatur:

/FBSY/ Funktionshandbuch Synchronaktionen; "Aktionen in Synchronaktionen"
/PGA/ Programmierhandbuch Arbeitsvorbereitung; "Bewegungssynchronaktionen"

6.5 Randbedingungen

Betriebsartengruppe

Bei SINUMERIK 840D sl stehen bis zu 10 BAGs zur Verfügung.

Bei SINUMERIK 828D ist nur 1 BAG verfügbar.

Kanäle

Bei SINUMERIK 840D sl stehen bis zu 10 Kanäle zur Verfügung.

Bei SINUMERIK 828D ist nur 1 Kanal verfügbar.

Achs-/Spindeltausch

Bei SINUMERIK 828D ist ein Achs-/Spindeltausch zwischen Kanälen nicht möglich.

Wechsel zur Kanal-Achse

Beim Wechsel einer Achse von PLC-Achse, neutrale Achse bzw. Achse im anderen Kanal in den Achstyp Kanal-Achse muss eine Synchronisierung stattfinden.

Bei dieser Synchronisation werden

- die aktuellen Positionen übernommen
- bei Spindeln die aktuelle Drehzahl und die aktuelle Getriebestufe übernommen.

Deshalb muss zwingend immer ein Vorlaufstopp durchgeführt werden, wodurch eine aktive Bahnbewegung unterbrochen wird.

Wird die Achse durch GET übernommen, ist dieser Übergang vom Teileprogramm her klar definiert.

Wird die Achse jedoch von der PLC zugeteilt, ist der Programmabschnitt, während dessen dieser Wechsel stattfindet, nicht eindeutig vorhersehbar.

(Außer durch eine eigene anwenderspezifische NC <-> PLC Logik)

Aus diesem Grund wird der Wechsel zur Kanal-Achse in folgenden Zuständen verzögert:

- Bahnbetrieb ist aktiv (G64+Achsen programmiert)
- Gewindeschneiden/-bohren ist aktiv (G33/G331/G332)

Wechsel von einer Kanal-Achse

Auch der Wechsel einer Kanal-Achse zu einer neutralen Achse bzw. PLC-Achsen kann nicht während eines aktiven Bahnbetriebs erfolgen.

Bei der Abgabe durch RELEASE ist dies dadurch bedingt, dass RELEASE in einem eigenen NC-Satz stehen muss.

Wechselt die PLC den Achstyp, wird intern ein REORG ausgelöst werden. Deshalb wird der Wechsel bei den genannten Programmzuständen verzögert.

Satzsuchlauf

Bei Satzsuchlauf mit Berechnung werden alle GET, GETD bzw. RELEASE Sätze gespeichert und, nach erneutem NC-Start, ausgegeben.

Ausnahme:

Sätze, die sich gegenseitig ausschließen, werden gelöscht.

Beispiel:

N10	RELEASE (AX1)	**Sätze werden gelöscht**
N40	GET (AX1)	"
N70	Ziel	

6.6 Datenlisten

6.6.1 Maschinendaten

6.6.1.1 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

6.6.1.2 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
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

Nummer	Bezeichner: $MC_	Beschreibung
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 Korrekturschalter-Eckwerte bei Lookahead
20440	LOOKAH_OVR_POINTS[n]	Korrekturschalter-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
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

Nummer	Bezeichner: $MC_	Beschreibung
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-Funktionen 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	Speichergröße 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	Speichergröße 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 Präperationtask (DRAM)
28510	MM_IPO_TASK_STACK_SIZE	Stackgröße der Ipo-Task (DRAM)

6.6.1.3 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
32074	FRAME_OR_CORRPOS_NOTALLOWED	Frame oder HL-Korrektur sind unzulässig
33100	COMPRESS_POS_TOL	Maximale Abweichung bei Kompression

6.6.2 Settingdaten

6.6.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: $SC_	Beschreibung
42000	THREAD_START_ANGLE	Startwinkel bei Gewinde
42100	DRY_RUN_FEED	Probelaufvorschub

6.6.3 Signale

6.6.3.1 Signale an/von BAG

Die BAG-Signale von PLC an NCK und von NCK an PLC sind im Datenbaustein 11 enthalten.

Die Signale werden beschrieben in:
Literatur:
Funktionshandbuch Grundfunktionen; NC/PLC-Nahtstellensignale (Z1),
Kapitel "BAG, Kanal, Programmbetrieb (K1)"

6.6.3.2 Signale an/von Kanal

Die Kanal-Signale von PLC an NCK und von NCK an PLC sind im Datenbaustein 21, 22, ... für den ersten, zweiten, ... Kanal enthalten.

Die Signale werden beschrieben in:
Literatur:
Funktionshandbuch Grundfunktionen; NC/PLC-Nahtstellensignale (Z1),
Kapitel "BAG, Kanal, Programmbetrieb (K1)"

M1: Kinematische Transformation 7

7.1 Kurzbeschreibung

7.1.1 TRANSMIT (Option)

Hinweis

Für die Funktion "TRANSMIT" ist die lizenzpflichtige Option "TRANSMIT und Mantelflächentransformation" erforderlich.

Die Funktion "TRANSMIT" ermöglicht folgende Leistungen:

- Stirnseitige Bearbeitung an Drehteilen in der Drehaufspannung
  - Bohrungen
  - Konturen
- Für die Programmierung dieser Bearbeitungen kann ein kartesisches Koordinatensystem benutzt werden.
- Die Steuerung transformiert die programmierten Verfahrbewegungen des kartesischen Koordinatensystems auf die Verfahrbewegungen der realen Maschinenachsen (Standardfall):
  - Rundachse (1)
  - Zustellachse senkrecht zur Drehachse (2)
  - Längsachse parallel zur Drehachse (3)

    Die Linearachsen (2) und (3) stehen senkrecht aufeinander.
- Werkzeugmittenversatz relativ zur Drehmitte ist zulässig.
- Die Geschwindigkeitsführung berücksichtigt die für die Drehbewegungen definierten Begrenzungen.
- Eine Bahn im kartesischen Koordinatensystem darf den Drehmittelpunkt nicht durchfahren. (Diese Einschränkung gilt für Softwarestände 2 und 3).

Weitere Möglichkeiten

- Die Werkzeugmittelpunktsbahn kann den Drehmittelpunkt der Rundachse durchfahren.
- Die Rundachse muss keine Moduloachse sein.

7.1.2 TRACYL (Option)

Hinweis

Für die Funktion "Zylindermantelkurventransformation (TRACYL)" ist die lizenzpflichtige Option "TRANSMIT und Mantelflächentransformation" erforderlich.

Die Funktion "Zylindermantelkurventransformation (TRACYL)" ermöglicht folgende Leistungen:

Bearbeitung von

- Längsnuten an zylindrischen Körpern,
- Quernuten an zylindrischen Körpern,
- beliebig verlaufenden Nuten an zylindrischen Körpern.

Der Verlauf der Nuten wird bezogen auf die abgewickelte, ebene Zylindermantelfläche programmiert.

Für die Bearbeitung werden Drehmaschinen mit

- X-C-Z-Kinematik und
- X-Y-Z-C-Kinematik

unterstützt.

- Die Steuerung transformiert die programmierten Verfahrbewegungen des Zylinder-Koordinatensystems auf die Verfahrbewegungen der realen Maschinenachsen (Standardfall X-C-Z-Kinematik TRAFO_TYPE_n = 512):
  - Rundachse (1)
  - Zustellachse senkrecht zur Drehachse (2)
  - Längsachse parallel zur Drehachse (3)

  Hinweis

  Die Linearachsen (2) und (3) stehen senkrecht aufeinander. Die Zustellachse (2) schneidet die Rundachse. Diese Anordnung ermöglicht keine Nutwandkorrektur.

- Für Nutwandkorrektur ist X-Y-Z-C Kinematik mit folgenden Achsen erforderlich (TRAFO_TYPE_n = 513):
  - Rundachse (1)
  - Zustellachse senkrecht zur Drehachse (2)
  - Längsachse parallel zur Drehachse (3)
  - Längsachse (4), die (2) und (3) zum rechthändigen kartesischen Koordinatensystem ergänzt.

  Hinweis

  Die Linearachsen (2), (3) und (4) stehen senkrecht aufeinander. Diese Konstellation ermöglicht Nutwandkorrektur.

- Die Geschwindigkeitsführung berücksichtigt die für die Drehbewegungen definierten Begrenzungen.

`TRACYL`-Transformation, ohne Nutwandkorrektur, mit zusätzlicher Längsachse (Zylindermantelkurventransformation ohne Nutwandkorrektur TRAFO_TYPE_n= 514)

- Für die Transformation ohne Nutwandkorrektur reicht eine Rundachse und eine Linearachse, die senkrecht zur Rundachse angeordnet ist völlig aus.
- Verfügt eine Maschine über eine weitere Linearachse, die senkrecht zur Rundachse und der ersten Linearachse ist, so kann diese Redundanz zur verbesserten Werkzeugkorrektur genutzt werden.

7.1.3 TRAANG (Option)

Hinweis

Für die Funktion "Schräge Achse (TRAANG)" ist die lizenzpflichtige Option "Schräge Achse" erforderlich.

Die Funktion "Schräge Achse (TRAANG)" ist für die Technologie Schleifen gedacht. Sie ermöglicht folgende Leistungen:

- Bearbeitung mit schräger Zustellachse.
- Für die Programmierung kann ein kartesisches Koordinatensystem verwendet werden.
- Die Steuerung transformiert die programmierten Verfahrbewegungen des kartesischen Koordinatensystems auf die Verfahrbewegungen der realen Maschinenachsen (Standardfall): schräge Zustellachse.

7.1.4 Verkettete Transformationen

Einführung

Es können jeweils zwei Transformationen hintereinander geschaltet (verkettet) werden, so dass die Bewegungsanteile für die Achsen aus der ersten Transformation Eingangsdaten für die verkettete zweite Transformation sind. Die Bewegungsanteile aus der zweiten Transformation wirken auf die Maschinenachsen.

Verkettungsmöglichkeiten

- Die Kette darf **zwei** Transformationen umfassen.
- Die **zweite** Transformation muss **"Schräge Achse"** (TRAANG) sein.
- Als erste Transformation sind möglich:
  - Orientierungstransformationen (TRAORI), inkl. Kardanischer Fräskopf
  - TRANSMIT
  - TRACYL
  - TRAANG

Informationen zu den übrigen Transformationen finden Sie in:
Literatur:
/FB3/ Funktionshandbuch Sonderfunktion; 3- bis 5-Achs-Transformationen (F2)

7.1.5 Transformations-MD über Teileprogramm/Softkey wirksam setzen

Die für die kinematischen Transformationen relevanten Maschinendaten sind bisher größtenteils POWER-ON-wirksam.

Es können Transformations-Maschinendaten auch über Teileprogramm/Softkey wirksam gesetzt werden, wodurch das Booten der Steuerung entfallen kann.

7.2 TRANSMIT (Option)

Hinweis

Für die im Folgenden beschriebene Transformation TRANSMIT müssen die während aktiver Transformation vergebenen Maschinenachsnamen, Kanalachsnamen und Geometrieachsnamen unterschiedlich sein.

Vergl.

MD10000 $MN_AXCONF_MACHAX_NAME_TAB,

MD20080 $MC_AXCONF_CHANAX_NAME_TAB,

MD20060 $MC_AXCONF_GEOAX_NAME_TAB.

Sonst sind keine eindeutigen Zuordnungen gegeben.

Aufgabenstellung

Komplettbearbeitung siehe Bild:

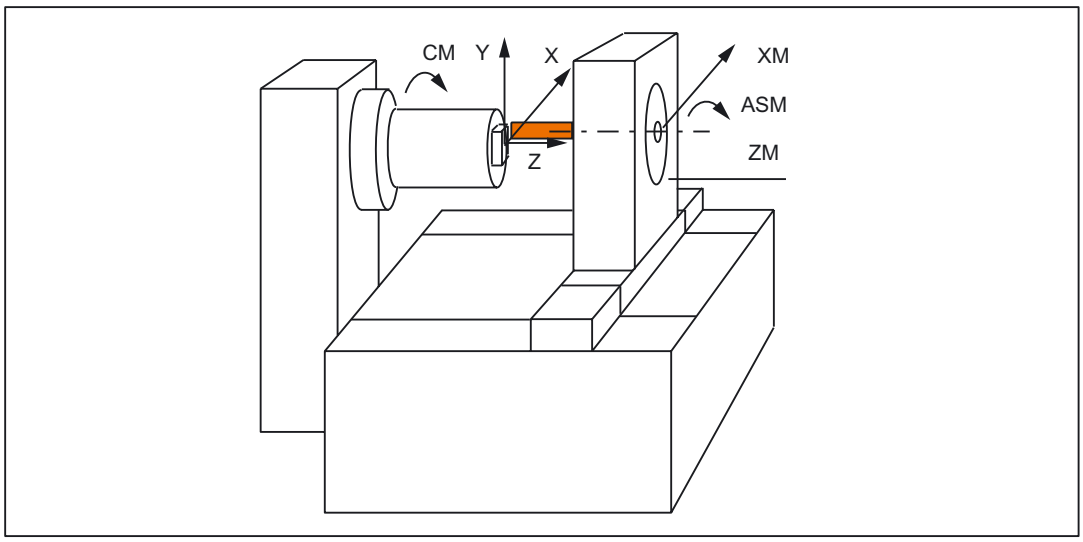


Bild 7-1 Stirnseitige Drehteilbearbeitung

Legende:

CM: Rundachse (Hauptspindel)

ASM: Arbeitsspindel (Fräser, Bohrer)

X, Y, Z: Kartesisches Koordinatensystem für die Programmierung der Stirnseitenbearbeitung (Ursprung im Drehmittelpunkt der Stirnfläche)

ZM: Maschinenachse (linear)

XM: Maschinenachse (linear)

7.2.1 Voraussetzungen für TRANSMIT

Achskonfiguration

Um im kartesischen Koordinatensystem (laut Bild X, Y, Z) programmieren zu können, muss der Steuerung der Zusammenhang zwischen diesem Koordinatensystem und den tatsächlich existierenden Maschinenachsen (CM, XM, ZM, ASM) mitgeteilt werden:

- Benennung der Geometrieachsen
- Zuordnung der Geometrieachsen zu Kanalachsen
  - allgemeiner Fall (`TRANSMIT` nicht aktiv)
  - `TRANSMIT` aktiv
- Zuordnung der Kanalachsen zu den Maschinenachsen-Nummern
- Kennzeichnung der Spindeln
- Zuweisung von Maschinenachs-Namen

Das Vorgehen entspricht mit Ausnahme des Punktes "`TRANSMIT` aktiv" dem Vorgehen bei der normalen Achskonfiguration. Wem die allgemeinen Schritte bereits bekannt sind, der möge von den folgenden Schritten nur den Schritt "Zuordnung der Geometrieachsen zu Kanalachsen" nachlesen.

Literatur:

/FB1/ Funktionsbeschreibung Grundmaschine; "Koordinatensysteme, Achstypen, Achskonfigurationen, Werkstücknahes Istwertsystem, Externe Nullpunktverschiebung" (K2)

Anzahl Transformationen

Im System können bis zu zehn Transformationsdatensätze je Kanal definiert werden. Die Maschinendaten-Namen dieser Transformationen beginnen mit "$MC_TRAFO …" und enden mit "... _n", wobei n für eine Ziffer von 1 bis 10 steht. Die folgenden Abschnitte beschreiben unter anderem diese Daten:

MD24100 $MC_TRAFO_TYPE_n

MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_n

MD24110 $MC_TRAFO_AXES_IN_n.

Anzahl TRANSMIT

Zwei der 10 zulässigen Datenstrukturen für Transformationen im Kanal dürfen für `TRANSMIT` belegt werden. Sie zeichnen sich dadurch aus, dass der mit folgendem Maschinendatum zugewiesene Wert 256 oder 257 ist:

MD24100 $MC_TRAFO_TYPE_n

Für diese maximal 2 TRANSMIT-Transformationen müssen die folgenden Maschinendaten definiert gesetzt werden:

MD24950 $MC_TRANSMIT_ROT_AX_OFFSET_t

MD24910 $MC_TRANSMIT_ROT_SIGN_IS_PLUS_t

MD24920 $MC_TRANSMIT_BASE_TOOL_t

MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_t

Dabei gibt t die Nummer der vereinbarten TRANSMIT Transformation an (maximal 2).

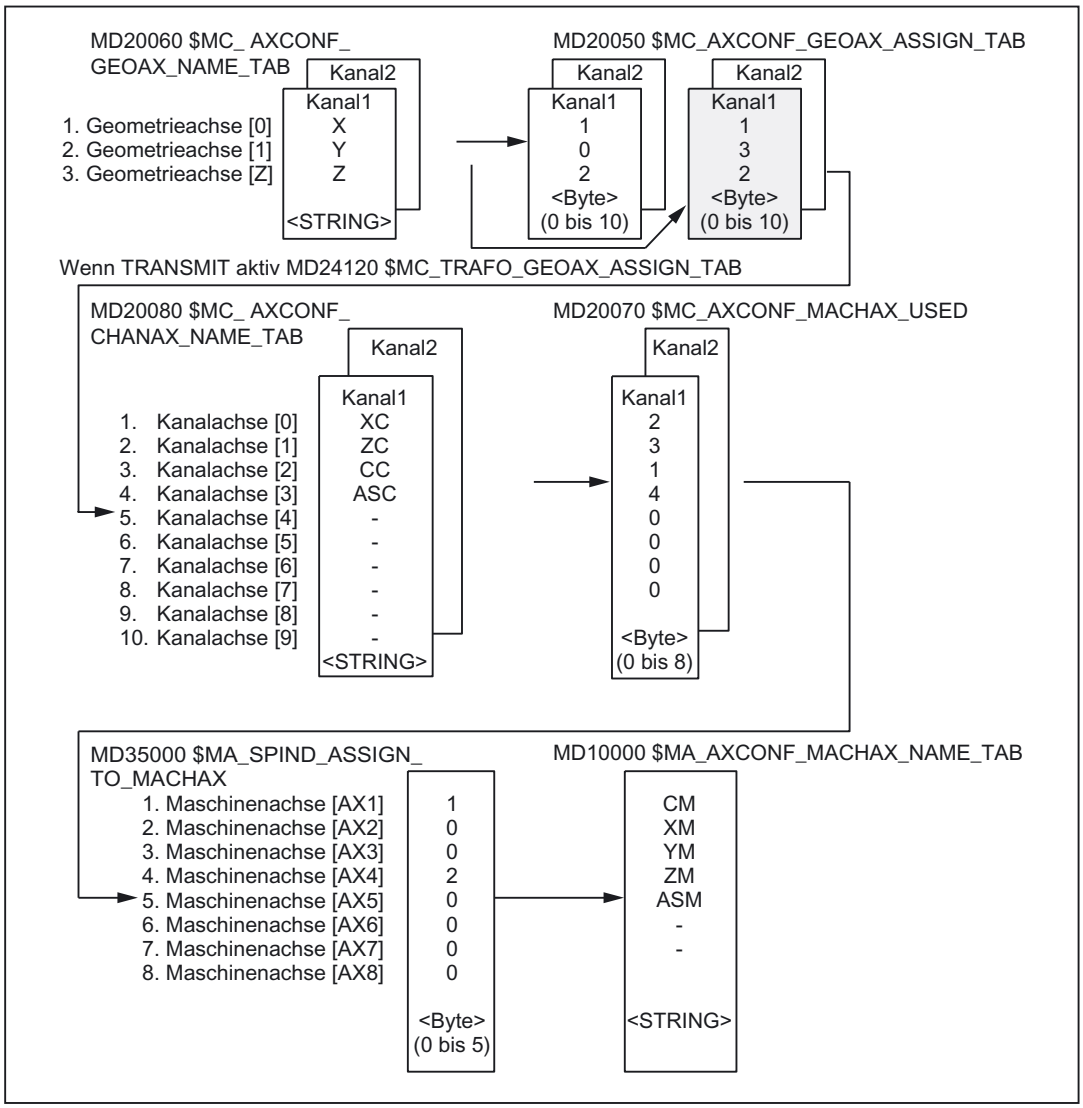


Bild 7-2 Achskonfiguration für das Beispiel im Bild "Stirnseitige Drehteilbearbeitung" (TRANSMIT)

Die im obigen Bild hervorgehobenen Anordnungen gelten bei TRANSMIT aktiv.

Benennung der Geometrieachsen

Gemäß vorstehender Übersicht über die Achskonfiguration sind die während TRANSMIT gewünschten Geometrieachsen folgendermaßen zu definieren:

MD20060 $MC_AXCONF_GEOAX_NAME_TAB[0]="X"

MD20060 $MC_AXCONF_GEOAX_NAME_TAB[1]="Y"

MD20060 $MC_AXCONF_GEOAX_NAME_TAB[2]="Z"

(Namenswahl laut obigen Bild entspricht auch der Vorbesetzung).

Zuordnung der Geometrieachsen zu Kanalachsen

Es sind die Fälle zu unterscheiden, ob `TRANSMIT` aktiv ist oder nicht:

- TRANSMIT nicht aktiv

  Eine Y-Achse steht nicht zur Verfügung.

  MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[0]=1

  MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB_TAB[1]=0

  MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB_TAB[2]=2

- TRANSMIT aktiv

  Die Y-Achse kann vom Teileprogramm angesprochen werden.

  MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1

  MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=3

  MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=2

  Die Y-Achse wird zum dritten Eintrag bei den Kanalachsen.

Eintrag der Kanalachsen

Es werden die Achsen zugefügt, die nicht zum kartesischen Koordinatensystem gehören.

MD20080 $MC_AXCONF_CHANAX_NAME_TAB[0]="XC"

MD20080 $MC_AXCONF_CHANAX_NAME_TAB[1]="ZC"

MD20080 $MC_AXCONF_CHANAX_NAME_TAB[2]="CC"

MD20080 $MC_AXCONF_CHANAX_NAME_TAB[3]="ASC"

Zuordnung der Kanalachsen zu Maschinenachsen

Mit Bezug auf den cd der Kanalachsen wird der Steuerung mitgeteilt, welcher Maschinenachsen-Nummer die Kanalachsen zugewiesen werden.

MD20070 $MC_AXCONF_MACHAX_USED[0]=2

MD20070 $MC_AXCONF_MACHAX_USED[1]=3

MD20070 $MC_AXCONF_MACHAX_USED[2]=1

MD20070 $MC_AXCONF_MACHAX_USED[3]=4

MD20070 $MC_AXCONF_MACHAX_USED[4]=0

(Einträge entsprechend obigem Bild)

Kennzeichnung der Spindeln

Je Maschinenachse wird festgelegt, ob eine Spindel vorliegt (Wert > 0: Spindelnummer) oder Bahnachse (Wert 0).

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[0]=1

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[1]=0

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[2]=0

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[3]=2

Zuweisung von Namen an Maschinenachsen

Mit Bezug auf den cd der Maschinenachsen wird der Steuerung ein Maschinenachs-Name mitgeteilt

MD10000 $MN_AXCONF_MACHAX_NAME_TAB[0]="CM"

MD10000 $MN_AXCONF_MACHAX_NAME_TAB[1]="XM"

MD10000 $MN_AXCONF_MACHAX_NAME_TAB[2]="ZM"

MD10000 $MN_AXCONF_MACHAX_NAME_TAB[3]="ASM"

7.2.2 TRANSMIT-spezifische Einstellungen

Art der Transformation

Der folgende Absatz beschreibt die Vorgabe des Transformationstyps.

TRAFO_TYPE_n

Bei den Transformationsdatensätzen (maximal n = 10) muss der Anwender den Typ der Transformation angeben. Für `TRANSMIT` ist der WERT 256 zu setzen oder bei einer Rundachse mit zusätzlicher Linearachse gilt der WERT 257.

Beispiel für WERT 256: MD24100 $MC_TRAFO_TYPE_1=256

Die Einstellung muss getroffen sein, bevor `TRANSMIT` oder `TRANSMIT`(t) aufgerufen wird. t ist die Nummer der vereinbarten `TRANSMIT`-Transformation.

Für die `TRANSMIT`-Transformation reicht eine Rundachse und eine Linearachse aus, die senkrecht zur Rundachse angeordnet ist. Mit den Transformationstyp 257 wird eine reale Y-Achse dazu genutzt, um z. B. einen Werkzeugversatz kompensieren zu können.

Transformations-Typ 257

Polartransformation mit einer Rundachse TRAFO_TYPE_n = 25710.04

Transformation mit zusätzlicher Linearachse

Verfügt die Maschine über eine weitere Linearachse, die senkrecht zur Rundachse und zur ersten Linearachse ist, so kann der Transformationstyp 257 für Werkzeugkorrekturen zum Herausfahren mit der realen Y-Achse eingesetzt werden. Dabei wird davon ausgegangen, dass der Arbeitsraum der zweiten Linearachse klein ist und nicht für das Abfahren des Teileprogramms genutzt werden soll.

Für MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_n gelten die bisherigen Einstellungen.

Achsabbildung

Der folgende Absatz beschreibt die Vorgabe der Transformations-Achsenabbildung.

TRAFO_AXES_IN_n

Für den Transformationsdatensatz n sind drei Kanalachsnummern anzugeben:

MD24110 $MC_TRAFO_AXES_IN_1[0]=Kanalachsnummer der Achse senkrecht zur Rundachse.

MD24110 $MC_TRAFO_AXES_IN_1[1]=Kanalachsnummer der Rundachse

MD24110 $MC_TRAFO_AXES_IN_1[2]=Kanalachsnummer der Achse parallel zur Rundachse

Beispiel für die Konfiguration laut Bild "Stirnseitige Drehteilbearbeitung" (TRANSMIT):

MD24110 $MC_TRAFO_AXES_IN_1[0]=1

MD24110 $MC_TRAFO_AXES_IN_1[1]=3

MD24110 $MC_TRAFO_AXES_IN_1[2]=2

Die Einstellung muss getroffen sein, bevor TRANSMIT oder TRANSMIT(t) aufgerufen wird. Die Achsnummern müssen sich auf die mit

MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_n definierten Kanalachsfolgen beziehen.

Für den Transformations-Typ 257 gelten für das

MD24110 $MC_TRAFO_AXES_IN_n[ ]

folgende Zuordnungen der Indizes.

Bedeutung der Indizes bezüglich des Basis-Koordinatensystems (BKS):

- [0]: kartesische Achse senkrecht zur Rundachse (in Maschinennullstellung ist diese Achse parallel zur Linearachse, die senkrecht zur Rundachse steht)
- [1]: kartesische Achse senkrecht zur Rundachse
- [2]: kartesische Achse parallel zur Rundachse (falls vorhanden)
- [3]: Linearachse parallel zur Index [2] in der Maschinengrundstellung

Bedeutung der Indizes bezüglich des Maschinen-Koordinatensystems (MKS):

- [0]: Linearachse senkrecht zur Rundachse
- [1]: Rundachse
- [2]: Linearachse parallel zur Rundachse (falls vorhanden)
- [3]: Linearachse senkrecht zu den Achsen von Index [0] und [1]

Drehlage

Die Drehlage des kartesischen Koordinatensystems wird, wie im folgenden Absatz beschrieben, durch Maschinendatum angegeben.

TRANSMIT_ROT_AX_OFFSET_t

Die Drehlage der x-y-Ebene des kartesischen Koordinatensystems gegenüber der definierten Nullstellung der Rundachse wird angegeben mit:

MD24900 $MC_TRANSMIT_ROT_AX_OFFSET_t= ... °

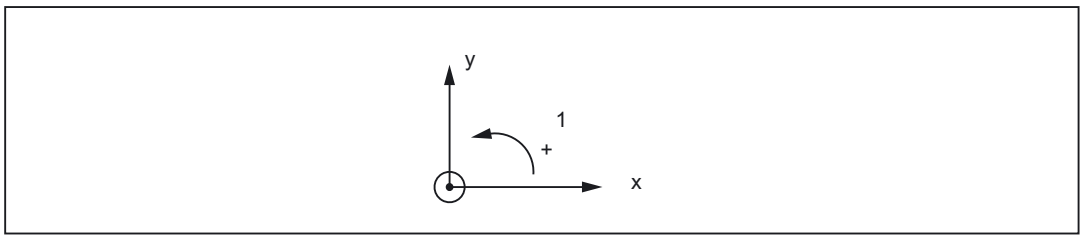
Dabei wird t ersetzt durch die Nummer der in den Transformationsdatensätzen vereinbarten TRANSMIT-Transformationen (t darf maximal 2 sein).

Drehsinn

Der Drehsinn der Rundachse wird, wie im folgenden Absatz beschrieben, durch Maschinendatum angegeben.

TRANSMIT_ROT_SIGN_IS_PLUS_t

Ist der Drehsinn der Rundachse in der x-y-Ebene bei Betrachtung gegen die z-Achse gegen den Uhrzeigersinn, so ist das Maschinendatum auf 1 zu setzen, andernfalls auf 0.



MD24910 $MC_TRANSMIT_ROT_SIGN_IS_PLUS_t=1

Dabei wird t ersetzt durch die Nummer der in den Transformationsdatensätzen vereinbarten TRANSMIT-Transformationen (t darf maximal 2 sein).

Lage des Werkzeugnullpunktes

Die Lage des Werkzeugnullpunktes wird, wie im folgenden Absatz beschrieben, durch Maschinendatum angegeben.

TRANSMIT_BASE_TOOL_t

Mit dem Maschinendatum:

MD24920 $MC_TRANSMIT_BASE_TOOL_t

wird der Steuerung mitgeteilt, in welcher Lage der Werkzeugnullpunkt bezogen auf den Ursprung des bei TRANSMIT vereinbarten Koordinatensystems liegt. Das Maschinendatum hat drei Komponenten für die drei Achsen des kartesischen Koordinatensystems.

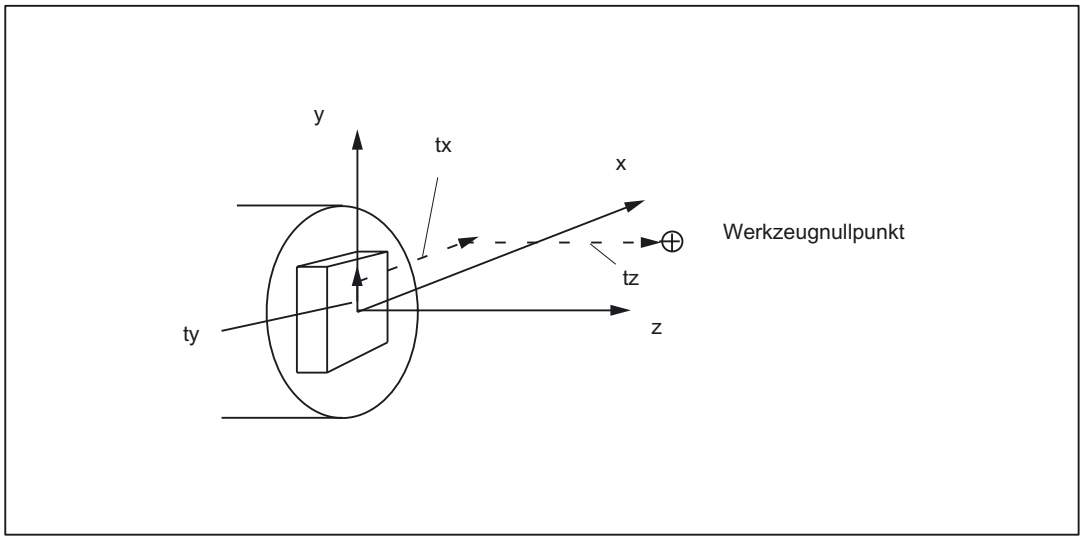


Bild 7-3 Lage des Werkzeugnullpunktes zum Ursprung des kartesischen Koordinatensystems

MD24920 $MC_TRANSMIT_BASE_TOOL_t[0]=tx

MD24920 $MC_TRANSMIT_BASE_TOOL_t [1]=ty

MD24920 $MC_TRANSMIT_BASE_TOOL_t [2]=tz

Dabei wird t vor der Index-Angabe [ ] ersetzt durch die Nummer der in den Transformationsdatensätzen vereinbarten TRANSMIT-Transformationen (t darf maximal 2 sein).

Umschaltbare Geometrieachsen

Ein Umschalten der Geometrieachsen mit GEOAX( ) wird dem PLC mitgeteilt, in dem optional ein über MD einstellbarer M-Code ausgegeben wird.

- MD22534 $MC_TRAFO_CHANGE_M_CODE

  Nummer des M-Codes, der bei einer Transformationsumschaltung am VDI-Interface ausgegeben wird.

Hinweis

Hat dieses Maschinendatum einen der Werte 0 bis 6, 17, 30, so wird kein M-Code ausgegeben.

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; K2, "Koordinatensysteme, Achstypen, Achskonfigurationen, Werkstücknahes Istwertsystem, Externe Nullpunktverschiebung"

7.2.3 Aktivieren von TRANSMIT

TRANSMIT

Nachdem die in den vorausgehenden Abschnitten beschriebenen Einstellungen getroffen wurden, kann die TRANSMIT-Funktion aktiviert werden:

TRANSMIT oder

TRANSMIT(t)

Mit TRANSMIT wird die erste vereinbarte TRANSMIT-Funktion aktiviert. TRANSMIT(t) aktiviert die t. vereinbarte TRANSMIT-Funktion. t darf maximal 2 sein.

Mit der Aktivierung werden ab Softwarestand 4 auch die speziellen Verfahren für Poldurchquerung usw. gemäß "Bearbeitungsmöglichkeiten für TRANSMIT" verfügbar.

Zwischen dem Aktivieren und dem nachfolgend beschriebenen Ausschalten können die Verfahrbewegungen für die Achsen des kartesischen Koordinatensystems programmiert werden.

7.2.4 Ausschalten der TRANSMIT-Funktion

TRAFOOF

Das Schlüsselwort TRAFOOF schaltet eine aktive Transformation aus. Bei ausgeschalteter Transformation ist das Basis-Koordinatensystem wieder mit dem Maschinen-Koordinatensystem identisch.

Eine aktive Transformation TRANSMIT wird ebenfalls ausgeschaltet, wenn im jeweiligen Kanal eine der übrigen Transformationen aktiviert wird.

(z. B. TRACYL, TRAANG, TRAORI).

Literatur:

/FB3/ Funktionshandbuch Sonderfunktion; "3-5-Achs-Transformation"(F2).

7.2.5 Besondere Reaktionen bei TRANSMIT

An- und Abwahl der Transformation sind über Teileprogramm bzw. MDA möglich.

Bei Anwahl beachten

- Ein Bewegungszwischensatz wird nicht eingefügt (Phasen/Radien).
- Eine Splinesatzfolge muss abgeschlossen sein.
- Werkzeugradiuskorrektur muss abgewählt sein.
- Eine aktivierte Werkzeuglängenkorrektur wird von Steuerung in die Transformation in die Geometrie-Achse übernommen.
- Der vor TRANSMIT wirksame Frame wird von der Steuerung abgewählt. (Entspricht programmiertem Frame rücksetzen G500).

- Eine aktive Arbeitsfeldbegrenzung wird für die von der Transformation betroffenen Achsen von der Steuerung abgewählt (entspricht programmiertem WALIMOF).
- Bahnsteuerbetrieb und Überschleifen werden unterbrochen.
- DRF-Verschiebungen in transformierten Achsen müssen vom Bediener gelöscht worden sein.

Bei Abwahl beachten

- Ein Bewegungszwischensatz wird nicht eingefügt (Phasen/Radien).
- Eine Splinesatzfolge muss abgeschlossen sein.
- Werkzeugradiuskorrektur muss abgewählt sein.
- Der vor TRANSMIT wirksame Frame wird von der Steuerung abgewählt. (Entspricht programmierten Frame rücksetzen G500).
- Bahnsteuerbetrieb und Überschleifen werden unterbrochen.
- DRF-Verschiebungen in transformierten Achsen müssen vom Bediener gelöscht worden sein.
- Eine Werkzeuglängenkorrektur in der virtuellen Achse (im Bild die Y-Achse) wird nicht mehr durchgeführt.

Einschränkungen bei aktivem TRANSMIT

Die im Folgenden genannten Einschränkungen sind bei aktiviertem TRANSMIT zu beachten.

Werkzeugwechsel

Ein Werkzeugwechsel ist nur bei abgewählter Werkzeugradiuskorrektur zulässig.

Frame

Alle Anweisungen, die sich nur auf das Basis-Koordinatensystem beziehen sind erlaubt (FRAME, Werkzeugradiuskorrektur). Ein Framewechsel bei G91 (Kettenmaß) wird aber - anders als bei inaktiver Transformation - nicht gesondert behandelt. Das zu fahrende Inkrement wird im Werkstück-Koordinatensystem des neuen Frames ausgewertet - unabhängig davon, welches Frame im Vorgängersatz wirkte.

Rundachse

Die Rundachse kann nicht programmiert werden, da sie von einer Geometrie-Achse belegt wird und somit als Kanalachse nicht direkt programmierbar ist.

Erweiterungen

Eine Verschiebung der Rundachse CM kann z. B. durch eine Kompensation der Schräglage eines Werkstückes in einem Frame innerhalb der Framekette eingetragen werden und führt dann zu entsprechenden x- und y-Werten gemäß des unteren Bildes.

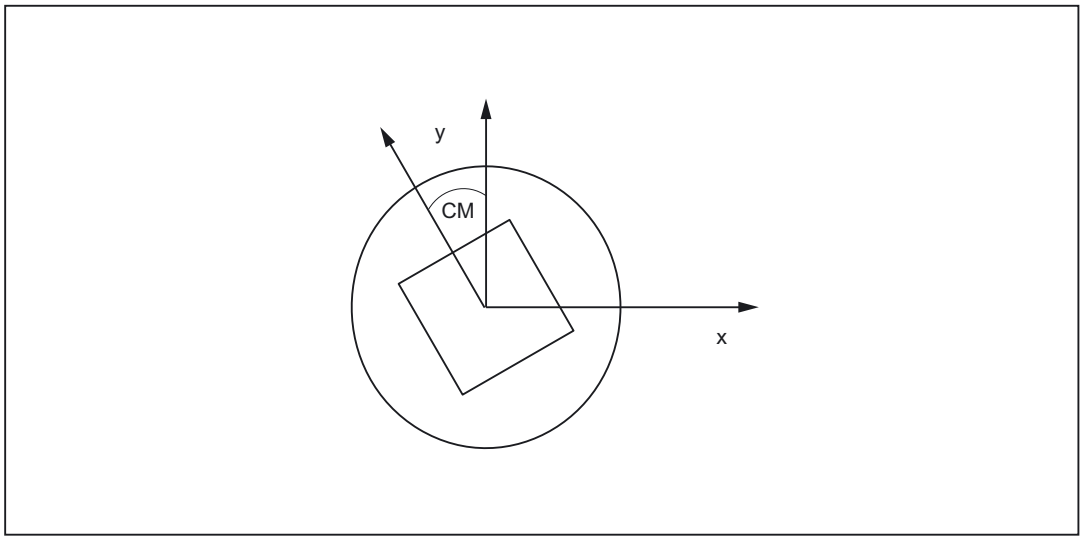


Bild 7-4 Rundachsverschiebung bei TRANSMIT

Diese Verschiebung kann auch in der Transformation als Offset der Rundachse berücksichtigt werden. Damit das axiale Gesamtframe der Transmit-Rundachse, d. h. die Translation, die Feinverschiebung, die Spiegelung und die Skalierung in der Transformation übernimmt, sind folgende Einstellungen erforderlich:

MD24905 $MC_TRANSMIT_ROT_AX_FRAME_1 = 1

MD24955 $MC_TRANSMIT_ROT_AX_FRAME_2 = 1

Hinweis

Änderungen der Achszuordnungen werden jeweils bei der An- und Abwahl der Transformation umgesetzt. Weitere Informationen zur axialen Verschiebung der Rundachsen bis zum ENS entnehmen Sie bitte:

Literatur:

/FB1/ Funktionshandbuch Grundfunktion; "Koordinatensysteme, Frames"(K2).

Pol

Ein Durchfahren durch den Pol (Ursprung des kartesischen Koordinatensystems) wird verhindert. Eine durch den Pol führende Bewegung bleibt im Pol stehen und es wird ein Alarm ausgegeben. Bei Fräsermittenversatz bleibt die Bewegung entsprechend am Rand des nicht anfahrbaren Bereiches stehen.

Die Möglichkeiten der Poldurchfahrung und Bearbeitungen in Polnähe sind in den Abschnitten "Bearbeitungsmöglichkeiten von TRANSMIT" beschrieben.

Ausschlüsse

Von der Transformation betroffene Achsen können nicht verwendet werden:

- als Preset-Achse (Alarm)
- für das Fixpunkt anfahren (Alarm)
- zum Referieren (Alarm)

Geschwindigkeitsführung

Die Geschwindigkeitsüberwachung bei `TRANSMIT` wird standardgemäß im Vorlauf durchgeführt. Die Überwachung und Begrenzung im Hauptlauf wird aktiviert:

- im AUTOMATIK-Betrieb, wenn eine Positionier- oder Pendelachse programmiert wurde, die in die Transformation über Maschinendatum $MC_TRAFO_AXES_IN_n Index 0 bzw. 1 eingeht.
- beim Wechsel in den JOG-Betrieb.

Die Überwachung wird wieder vom Hauptlauf in den Vorlauf übernommen, wenn die transformationsbedeutsamen Achsen als Bahnachsen verfahren werden.

Die Geschwindigkeitsüberwachung im Vorlauf nutzt die Maschine besser aus als die Überwachung im Hauptlauf. Die Überwachung im Hauptlauf deaktiviert ferner den Look Ahead.

Unterbrechung des Teileprogramms

Wird die Teileprogrammabarbeitung für JOG unterbrochen, so ist das Folgende zu beachten:

JOG

Mit Umschalten in JOG wird die herkömmliche Online-Geschwindigkeitskontrolle statt der optimierten Geschwindigkeitskontrolle aktiviert.

Von Automatik nach Jog

Wird eine Teileprogrammbearbeitung bei aktiver Transformation unterbrochen und mit Jog verfahren, so ist bei Wiederanwahl von Automatik zu beachten:

- die Transformation ist auch im Anfahrsatz von der aktuellen Position zur Unterbrechungsstelle aktiv. Eine Kollisionsüberwachung wird nicht durchgeführt.

⚠ WARNUNG
Der Bediener ist dafür verantwortlich, dass das Werkzeug problemlos rückpositioniert werden kann.

Im AUTOMATIK-Betrieb

Solange die transformationssignifikanten Achsen als Bahnachsen synchron zueinander verfahren werden, ist die geschwindigkeitsoptimierte Geschwindigkeitsplanung aktiv. Wird eine transformationsbedeutsame Achse als Positionierachse verfahren, so wird bis zum Ausschalten der Transformation, bzw. bis alle transformationssignifikanten Achsen wieder Bahnachsen sind, die Online-Geschwindigkeitskontrolle aktiviert. Die Rückkehr in den geschwindigkeitsoptimierten Betrieb löst automatisch `STOPRE` und Synchronisation der azyklischen Satzvorverarbeitung mit der Interpolation aus.

Von Start nach Reset

Wird eine Teileprogrammbearbeitung mit `RESET` abgebrochen und mit `START` neu begonnen, so ist zu beachten:

- Nur, wenn zu Teileprogrammbeginn alle Achsen über einen Linearsatz (G0 bzw. G1) auf eine definierte Position gefahren werden, wird das restliche Teileprogramm reproduzierbar abgefahren. Ein bei `RESET` aktives Werkzeug wird ggf. von der Steuerung nicht mehr beachtet (festlegbar über Maschinendaten).

Power On RESET

Für das Verhalten nach Power On sind die in den folgenden Maschinendaten hinterlegten Vorgaben bedeutsam:

MD20110 $MC_RESET_MODE_MASK und

MD20140 $MC_RAFO_RESET_VALUE

Literatur:

/FB1/ Funktionshandbuch Grundfunktion; "Werkstücknahes Istwertsystem"(K2).

Referenzpunktfahren

Referieren ist mit aktiver Transformation nicht möglich. Eine aktive Transformation wird beim Referenzpunktfahren durch die Steuerung abgewählt.

7.2.6 Bearbeitungsmöglichkeiten für TRANSMIT

Einführung

Die Transformation `TRANSMIT` weist einen Pol im Nullpunkt der `TRANSMIT`-Ebene auf (im Beispiel Bild: 2–1, x = 0, Y = 0). Der Pol liegt im Schnittpunkt der radialen Linearachse mit der Rundachse (im Beispiel X und CM). In Polnähe erzeugen kleine Positionsänderungen der Geometrieachsen in der Regel große Positionsänderungen der Maschinenrundachse. Davon ausgenommen sind nur Linearbewegungen in bzw. durch den Pol.

Eine Werkzeugmittelpunktsbahn, die durch den Pol führt, verursacht keinen Teileprogrammabbruch. Es gibt keine Einschränkung hinsichtlich der programmierbaren Wegbefehle oder bezüglich einer aktiven Werkzeugradiuskorrektur. Eine Werkstückbearbeitung in Polnähe ist aber unabhängig davon nicht empfehlenswert, da ggf. starke Vorschubreduzierungen erforderlich sind, um die Rundachse nicht zu überlasten.

Neue Möglichkeiten

Definition:

Ein Pol liegt vor, wenn die Linie, die der Werkzeugmittelpunkt beschreibt, die Drehmitte der Rundachse schneidet.

Folgende Fälle werden behandelt:

- Unter welchen Bedingungen und wie durch den Pol gefahren werden kann
- Das Verhalten in Polnähe
- Das Verhalten bezüglich Arbeitsraumbegrenzungen
- Kontrolle bei Rundachsendrehungen über 360°.

Poldurchfahrung

Für die Poldurchfahrung gibt es zwei Möglichkeiten:

- Verfahren der Linearachse allein
- Verfahren in den Pol mit Drehung der Rundachse im Pol

Linearachse allein

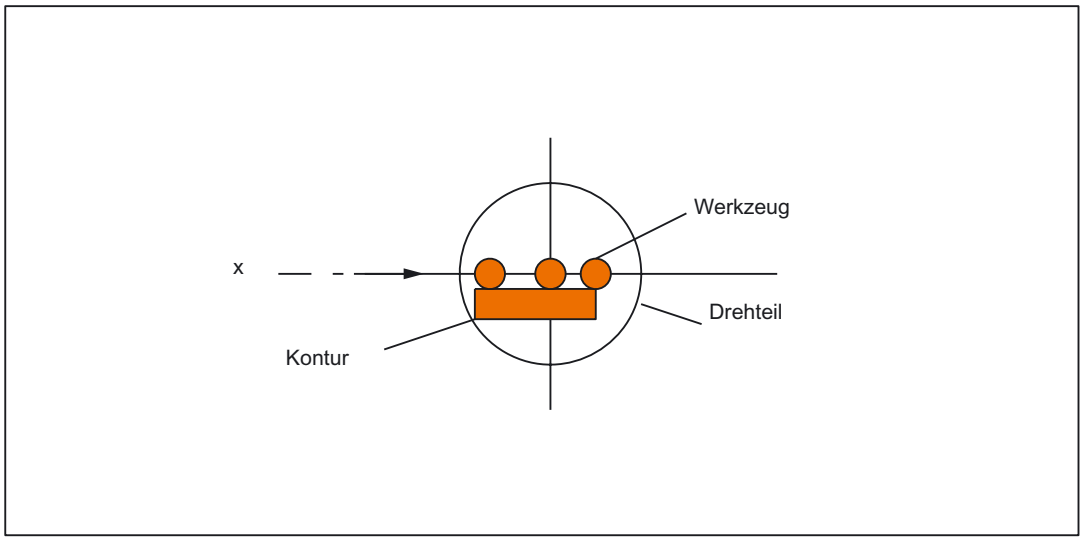


Bild 7-5 Fahren der X-Achse durch den Pol

Drehung im Pol

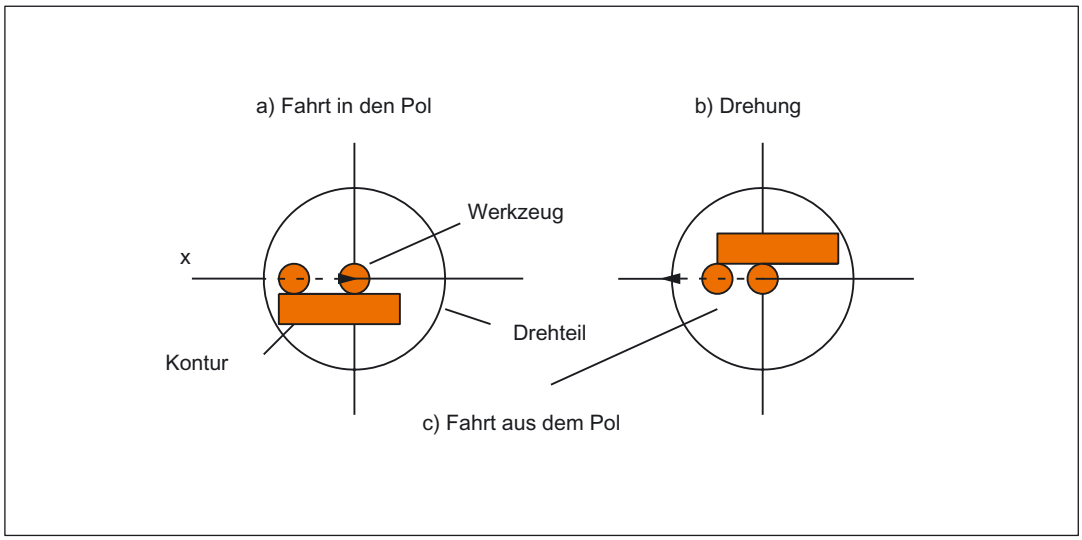


Bild 7-6 Fahren der X-Achse in den Pol (a), Drehung (b), Fahren aus dem Pol (c)

Verfahrensauswahl

Die Auswahl des Verfahrens muss die Möglichkeiten der Maschine und die Erfordernisse des herzustellenden Teils berücksichtigen. Die Auswahl erfolgt durch die Maschinendaten:

MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1

MD24951 $MC_TRANSMIT_POLE_SIDE_FIX_2

Das erste MD gilt für die erste TRANSMIT-Transformation im Kanal, das zweite entsprechend für die zweite TRANSMIT-Transformation im Kanal.

WERT	Bedeutung
0	Poldurchfahrung Die Werkzeugmittelpunktsbahn (Linearachse) soll stetig durch den Pol führen.
1	Drehung um den Pol. Die Werkzeugmittelpunktsbahn soll sich ausschließlich im positiven Verfahrbereich der Linearachse befinden. (Vor der Drehmitte).
2	Drehung um den Pol. Die Werkzeugmittelpunktsbahn soll sich ausschließlich im negativen Verfahrbereich der Linearachse befinden. (Hinter der Drehmitte).

Besonderheiten bei Poldurchfahren

Die Durchfahrung des Pols mit der Linearachse allein ist in den Betriebsarten AUTOMATIK und JOG zulässig.

Verhalten:

Tabelle 7- 1 Durchfahren des Pols mit der Linearachse allein

Betriebsart	Zustand	Reaktion
AUTOMATIK	Alle an der Transformation beteiligten Achsen werden synchron bewegt. TRANSMIT aktiv.	Zügige Poldurchfahrung
	Nicht alle an der Transformation beteiligten Achsen werden synchron bewegt. (z. B. Positionsachse). TRANSMIT nicht aktiv.	Schleichfahrt durch den Pol
	Eine eingefahrene DRF (externe Nullpunktsverschiebung) stört nicht. Beim Einfahren können in Polnähe ggf. Servo-Fehler hervorgerufen werden.	Abbruch der Bearbeitung, Alarm
JOG	-	Schleichfahrt durch den Pol

Besonderheiten bei Drehung im Pol

Voraussetzung: Das Verfahren wirkt nur in Betriebsart AUTOMATIK.

MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1 = 1 oder 2

MD24951 $MC_TRANSMIT_POLE_SIDE_FIX_2 = 1 oder 2

Wert: 1 Linearachse bleibt im positiven Verstellbereich

Wert: 2 Linearachse bleibt im negativen Verstellbereich

Bei einer Kontur, welche die Durchquerung des Pols mit der Werkzeugmittelpunktsbahn erfordern würde, wird die Bewegung der Linearachse in Bereiche jenseits der Drehmitte vermieden durch folgende drei Schritte:

Schritt	Aktion
1	Fahrt der Linearachse in den Pol
2	Drehung der Rundachse um 180°, dabei bleiben die übrigen an der Transformation beteiligten Achsen stehen.
3	Abarbeiten des restlichen Satzes. Dabei bewegt sich die Linearachse wieder weg vom Pol.

Bei Betriebsart JOG bleibt die Bewegung im Pol stehen. In Betriebsart JOG darf aus dem Pol heraus nur auf der Tangente der Bahn gefahren werden, auf der in den Pol gefahren wurde. Alle anderen Bewegungsvorgaben erforderten einen Positionssprung der Rundachse bzw. eine große Maschinenbewegung bei minimalen Bewegungsvorgaben. Sie werden mit Alarm 21619 abgewiesen.

Fahren in Polnähe

Führt eine Werkzeugmittelpunktsbahn am Pol vorbei, verringert die Steuerung automatisch den Vorschub und die Bahnbeschleunigung so, dass die Kennwerte der Maschinenachsen (MD32000 $MA_MAX_AX_VELO[AX*] und MD32300 $MA_ MAX_AX_ACCEL[AX*]) nicht überschritten werden. Je näher die Bahn am Pol vorbei führt, desto größer ist die Zurücknahme des Vorschubs.

WZ-Mittelpunktsbahn mit Ecke im Pol

Weist die Werkzeugmittelpunktsbahn eine Ecke im Pol auf, so bedeutet dies nicht nur einen Sprung in den Achsgeschwindigkeiten, sondern auch einen Positionssprung in der Rundachse. Dieser kann nicht durch Abbremsen verringert werden.

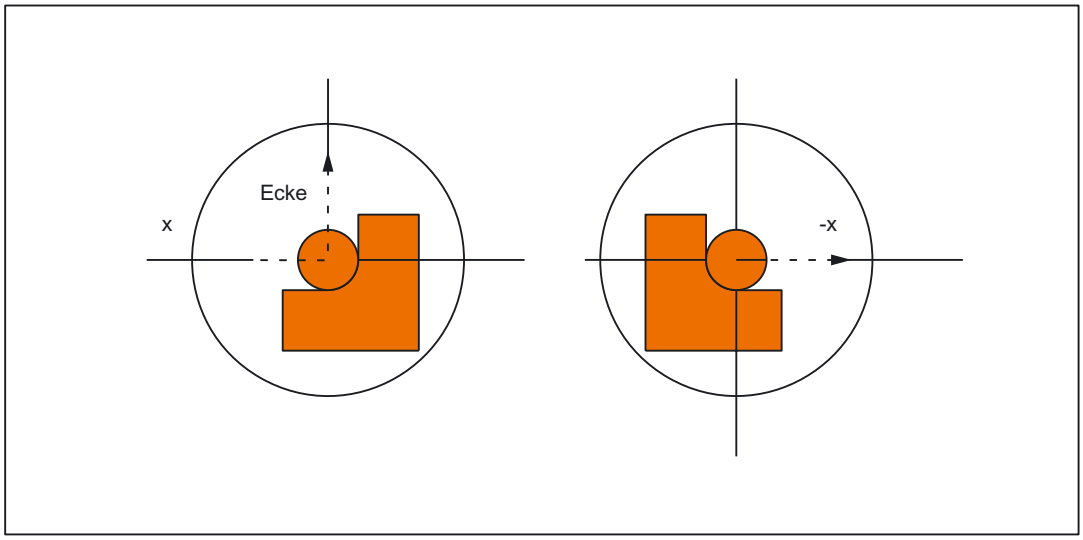


Bild 7-7 Poldurchquerung

Voraussetzungen:

Betriebsart AUTOMATIK,

MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1 = 0

oder

MD24951 $MC_TRANSMIT_POLE_SIDE_FIX_2 = 0

Die Steuerung fügt an der Sprungstelle einen Verfahrsatz ein, der die **kleinstmögliche Drehung** erzeugt, um die Kontur weiter zu bearbeiten.

Ecke ohne Poldurchquerung

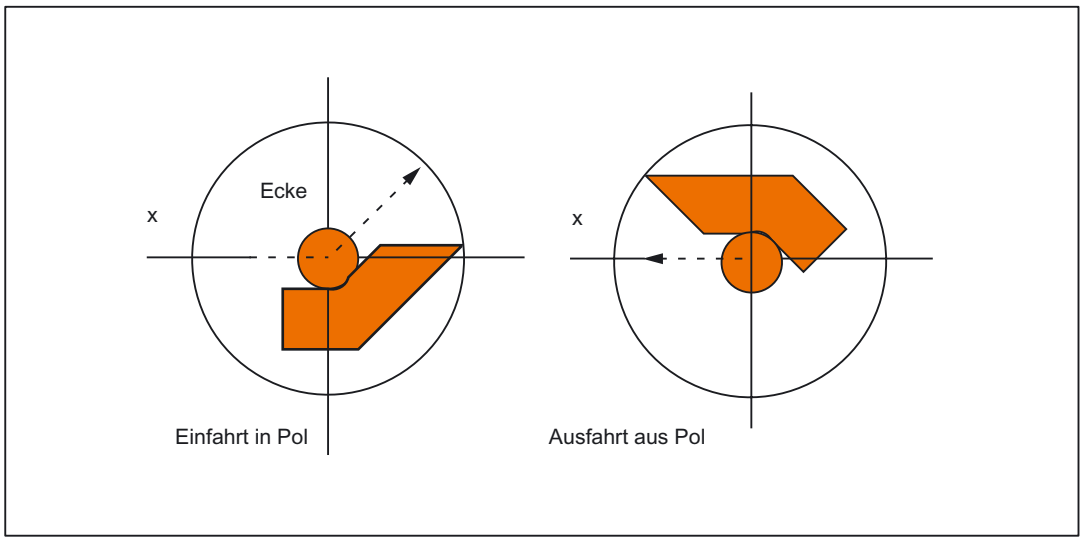


Bild 7-8 Bearbeitung auf einer Polseite

Voraussetzungen:

Betriebsart AUTOMATIK,

MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1 = 1 oder 2

oder

MD24951 $MC_TRANSMIT_POLE_SIDE_FIX_2 = 1 oder 2

Die Steuerung fügt an der Sprungstelle einen Verfahrsatz ein, der die **erforderliche Drehung** erzeugt, um die Kontur weiter auf der gleichen Seite des Pols zu bearbeiten.

Transformationsanwahl im Pol

Soll aus einer Stellung der Werkzeugmittenbahn heraus, die dem Pol der eingeschalteten Transformation entspricht, weitergearbeitet werden, so liegt für die neue Transformation ein Fahren aus dem Pol vor.

Ist

MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1= 0

oder

MD24951 $MC_TRANSMIT_POLE_SIDE_FIX_2= 0

gesetzt (Poldurchquerung), so wird zu Beginn des Satzes, der aus dem Pol führt, eine **möglichst kleine** Drehung erzeugt. Entsprechend dieser Drehung wird im weiteren vor bzw. hinter der Drehmitte gefahren.

Für

MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1= 1

oder

MD24951 $MC_TRANSMIT_POLE_SIDE_FIX_2= 1

wird **vor** der Drehmitte gearbeitet (Linearachse im positiven Stellbereich), für

MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1= 2

oder

MD24951 $MC_TRANSMIT_POLE_SIDE_FIX_2= 2

hinter der Drehmitte (Linearachse im negativen Stellbereich).

Transformationsanwahl außerhalb des Pols

Die Steuerung verfährt die an der Transformation beteiligten Achsen, ohne das Maschinendatum MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_t auszuwerten. Dabei kennzeichnet t = 1 die erste und t = 2 die zweite TRANSMIT-Transformation im Kanal.

7.2.7 Arbeitsraumbegrenzungen

Ausgangssituatiion

Bei `TRANSMIT` erhält man anstelle des Pols eine Arbeitsraumbegrenzung, wenn der Werkzeugmittelpunkt nicht in die Drehmitte der in die Transformation eingehenden Rundachse positioniert werden kann. Dies tritt auf, wenn die zur Rundachse senkrechte Achse (unter Berücksichtigung der Werkzeugkorrektur) nicht radial zur Rundachse liegt, bzw. beide Achsen zueinander windschief stehen. Der Abstand beider Achsen definiert einen zylinderförmigen Raum im BCS, in den nicht positioniert werden kann.

Durch die Software-Endschalter-Überwachung kann der verbotene Bereich nicht berücksichtigt werden, da der Verfahrbereich der Maschinenachsen nicht betroffen ist.

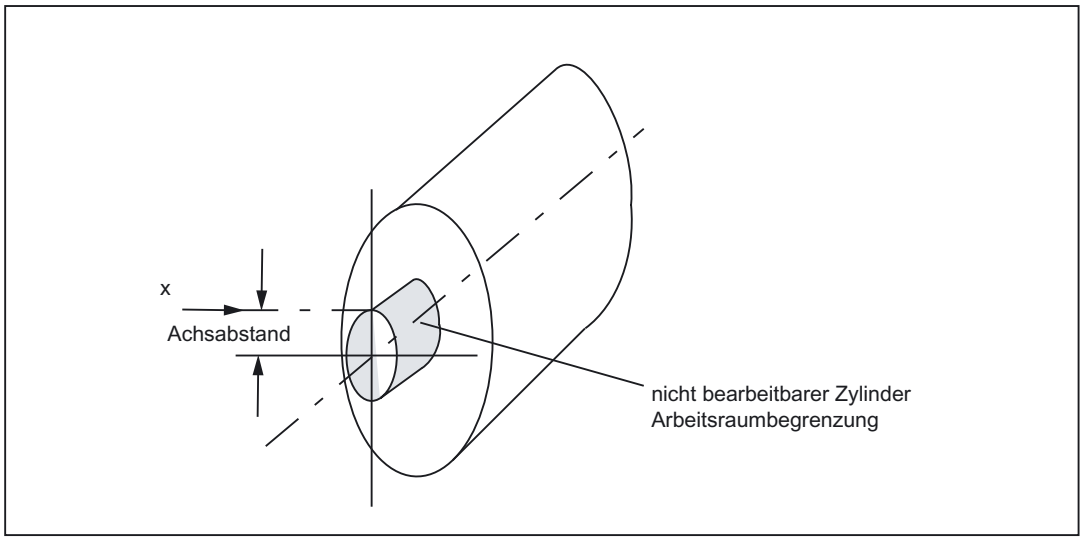


Bild 7-9 Arbeitsraumbegrenzung durch versetzte Linearachse

Fahren in die Arbeitsraumbegrenzung

Eine Bewegung, die in die Arbeitsraumbegrenzung führt, wird mit Alarm 21619 abgewiesen. Ein entsprechender Teileprogrammsatz wird nicht abgearbeitet. Die Steuerung bleibt am Ende des vorhergehenden Satzes stehen.

Kann die Bewegung nicht ausreichend vorhergesehen werden (JOG-Betriebsarten, Positionierachsen), so bleibt die Steuerung am Rand der Arbeitsraumbegrenzung stehen.

Verhalten nahe der Arbeitsraumbegrenzung

Führt eine Werkzeugmittelpunktsbahn am verbotenen Bereich vorbei, verringert die Steuerung automatisch den Vorschub und die Bahnbeschleunigung so, dass die Kennwerte der Maschinenachsen (MD32000 $MA_MAX_AX_VELO[AX*] und MD32300 $MA_MAX_AX_ACCEL[AX*]) nicht überschritten werden. Je näher die Bahn an der Arbeitsraumbegrenzung vorbeiführt, desto größer ist ggf. die Zurücknahme des Vorschubs.

7.2.8 Überlagerte Bewegungen bei TRANSMIT

Überlagerte Bewegungsverläufe sind durch die Steuerung nicht voraussehbar. Sie stören jedoch nicht, wenn sie - bezogen auf den aktuellen Abstand zum Pol (bzw. zur Arbeitsraumbegrenzung) - sehr klein sind (z. B. Werkzeugfeinkorrektur). Für die transformationsbedeutsamen Achsen überwacht die Transformation die überlagerte Bewegung und meldet eine kritische Größenordnung durch den Alarm 21618 . Der Alarm zeigt an, dass die satzbezogene Geschwindigkeitsplanung nicht mehr gut genug den tatsächlichen Verhältnissen an der Maschine entspricht. Mit dem Alarm wird deshalb die herkömmliche, nicht optimierte Online-Geschwindigkeitskontrolle aktiviert. Durch steuerungsintern erzeugten `REORG` wird der Vorlauf mit dem Hauptlauf wieder synchronisiert.

Der Alarm sollte durch den Anwender vermieden werden, da er einen Zustand signalisiert, der zu Achsüberlastung und damit Abbruch der Teileprogrammbearbeitung führen kann.

7.2.9 Kontrolle bei Rundachsdrehungen über 360 Grad

Mehrdeutigkeit der Rundachspositionen

Die Positionen der Rundachse sind mehrdeutig bezüglich der Anzahl der Umdrehungen. Die Steuerung zerlegt Sätze mit mehreren Umschlingungen des Pols in Teilsätze.

Diese Unterteilung ist bei parallelen Aktionen (z. B. Ausgabe von Hilfsfunktionen, satzsynchronisierten Positionierachsbewegungen) zu beachten, da für die Synchronisation nicht mehr das programmierte Satzende sondern das Ende des ersten Teilsatzes maßgeblich ist. Siehe:

Literatur:

/FB1/ Funktionshandbuch Grundfunktion; "Hilfsfunktionsausgabe an PLC"(H2)

/FB2/ Funktionshandbuch Grundfunktion; "Synchronaktionen"(S5).

Im Einzelsatzbetrieb bearbeitet die Steuerung die einzelnen Sätze explizit. Ansonsten werden die Teilsätze per Look Ahead so abgefahren wie ein einziger. Eine Begrenzung des Stellbereichs der Rundachse wird über die Software-Endschalterkontrolle überwacht.

7.2.10 Randbedingungen

Vorausschau

Alle Funktionen, die eine Vorausschau benötigen (Fahren durch den Pol, Look Ahead), arbeiten nur zufrieden stellend, wenn die relevanten Achsbewegungen exakt vorausgerechnet werden können. Bei `TRANSMIT` betrifft dies die Rundachse und die dazu senkrechte Linearachse. Ist eine dieser Achsen Positionierachse, so wird mit der Ausgabe des Alarms 10912 Look Ahead abgeschaltet und die herkömmliche Online-Geschwindigkeitskontrolle aktiviert.

Verfahrenswahl

Für die optimale Wahl "Fahren durch den Pol" oder "Drehen um den Pol" ist der **Anwender verantwortlich**.

Mehrere Poldurchquerungen

Ein Satz kann beliebig oft durch den Pol führen (z. B. durch Programmieren einer Schraubenlinie [Helix] mit mehreren Windungen). Der Teileprogrammsatz wird entsprechend in kleinere Teilsätze zerlegt. Analog dazu werden Sätze, die den Pol mehrfach umschlingen, ebenfalls in Teilsätze aufgeteilt.

Rundachse als Modulo-Achse

Die Rundachse kann eine Modulorundachse sein. Im Gegensatz zu Softwarestand 2 und 3 ist dies aber keine Voraussetzung. Die diesbezüglichen Beschränkungen von Softwarestand 2 und 3 entfallen.

Rundachse als Spindel

Wird die Rundachse ohne Transformation als Spindel verwendet, so muss sie vor Anwahl der Transformation mit `SPOS` in den lagegeregelten Betrieb geschaltet werden.

TRANSMIT mit zusätzlicher Linearachse

Im Teileprogramm muss bei aktiven TRANSMIT der Kanalbezeichner von posBCS[ax[3]] einen anderen Namen haben, wie die Geometrieachsen. Wird posBCS[ax[3]] nur außerhalb TRANSMIT geschrieben, dann gilt diese Einschränkung nicht, wenn die Achse einer Geometrieachse zugeordnet war. Bei aktivem TRANSMIT wird keine Konturinformation über ax[3] verarbeitet.

REPOS

Auf die Teilsätze, die durch das für Softwarestand 4 erweiterte TRANSMIT-Verfahren entstehen, kann repositioniert werden. Die Steuerung verwendet dabei den ersten Teilsatz, der im BCS am nächsten zur zu repositionierenden Position ist.

Satzsuchlauf

Bei Satzsuchlauf mit Berechnung wird auf den Satzendpunkt (des letzten Teilsatzes) gefahren, wenn im Zuge der Erweiterungen im Softwarestand 4 Zwischensätze erzeugt wurden.

7.3 TRACYL (Option)

Hinweis

Für die im folgenden beschriebene Transformation TRACYL müssen die während aktiver Transformation vergebenen Maschinenachsnamen, Kanalachsnamen und Geometrieachsnamen unterschiedlich sein. Vergl.

MD10000 $MN_AXCONF_MACHAX_NAME_TAB,

MD20080 $MC_AXCONF_CHANAX_NAME_TAB,

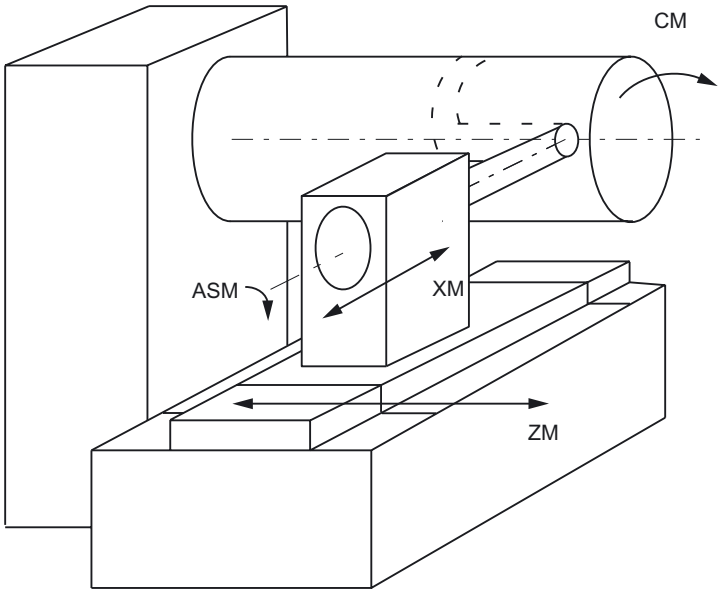
MD20060 $MC_AXCONF_GEOAX_NAME_TAB.

Sonst sind keine eindeutigen Zuordnungen gegeben.

Aufgabenstellung

Nutbearbeitung siehe Bild.

Achskonfiguration 1



Legende:
XM Zustellachse, senkrecht zur Rundachse
ZM Achse parallel zur Rundachse
CM Rundachse
ASM Arbeitsspindel

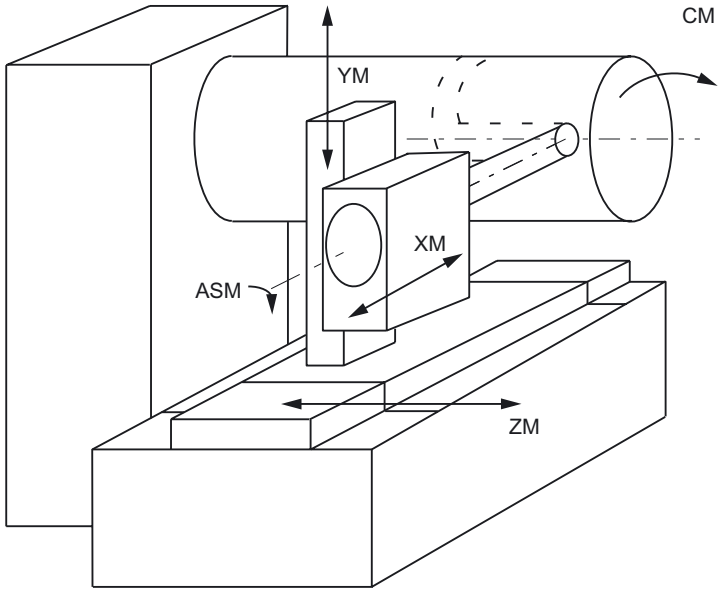
Bild 7-10 Nutbearbeitung am Zylindermantel mit X-C-Z-Kinematik

Achskonfiguration 1

Die Zylindermantelkurven-Transformation gestattet die Vorgabe des Verfahrwunsches bezogen auf die Mantelfläche eines Zylinderkoordinatensystems. Die Maschinenkinematik muss dem Zylinderkoordinatensystem entsprechen. Sie muss ein bzw. zwei Linearachsen und eine Rundachse umfassen. Die beiden Linearachsen müssen senkrecht zueinander stehen. Die Rundachse muss zu einer der Linearachsen parallel ausgerichtet sein und die zweite Linearachse schneiden. Die Rundachse muss ferner kollinear zum Zylinderkoordinatensystem der Programmierung sein.

Bei nur einer Linearachse (X) können nur Nuten parallel zum Umfang des Zylinders erzeugt werden. Bei zwei linearen Achsen (X,Z) können die Nuten eine beliebige Form auf dem Zylinder annehmen.

Achskonfiguration 2



Legende:

XM	Zustellachse, senkrecht zur Rundachse
YM	Ergänzungsachse
ZM	Achse parallel zur Rundachse
CM	Rundachse
ASM	Arbeitsspindel

Bild 7-11 Nutbearbeitung am Zylindermantel mit X-Y-Z-C-Kinematik

Steht eine dritte Linearachse zur Verfügung, die die beiden anderen Linearachsen (Achskonfiguration 1) zu einem rechtshändigen kartesischen Koordinatensystem ergänzt, so wird diese ausgenützt, um mit Hilfe der Werkzeugradiuskorrektur das Werkzeug **parallel zur programmierten Bahn** zu versetzen. Damit können Nuten mit rechteckigem Querschnitt erzeugt werden.

Funktionalität

Während der Transformation (beide Achskonfigurationen) ist die volle Funktionalität der Steuerung sowohl bei der Abarbeitung aus dem NC-Programm als auch bei JOG verfügbar.

Nutquerschnitt

Bei Achskonfiguration 1 sind Nuten längs zur Rundachse nur dann parallel begrenzt, wenn die Nutbreite genau dem Werkzeugradius entspricht.

Nuten parallel zum Umfang (Quernuten) sind an Anfang und Ende nicht parallel.

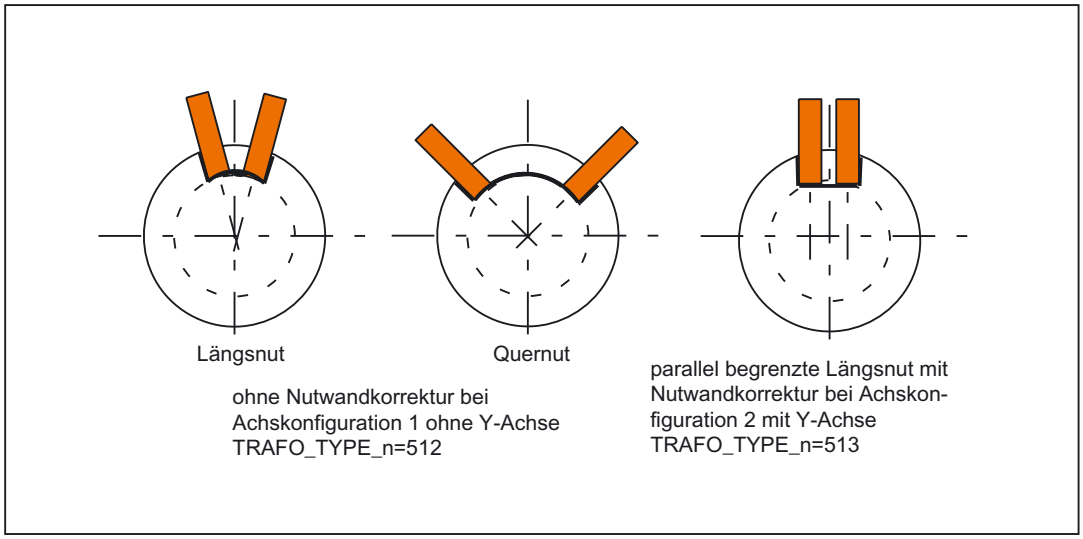


Bild 7-12 Nuten ohne und mit Nutwandkorrektur

7.3.1 Voraussetzungen für TRACYL

Anzahl Transformationen

Im System können bis zu 10 Transformationsdatensätze je Kanal definiert werden. Die Maschinendaten-Namen dieser Transformationen beginnen mit $MC_TRAFO .. und enden mit ... _n, wobei n für eine Ziffer von 1 bis 10 steht. Das erste Maschinendatum hat die gleiche Bedeutung wie bei TRANSMIT beschrieben:

MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_n

MD24100 $MC_TRAFO_TYPE_n

MD24110 $MC_TRAFO_AXES_IN_n.

Für MD24100 $MC_TRAFO_TYPE_n und MD24110 $MC_TRAFO_AXES_IN_n gelten für Zylindermantelflächen-Transformation (TRACYL) besondere Einstellungen, die im folgenden beschrieben werden.

Anzahl TRACYL

Drei der 10 zulässigen Datenstrukturen für Transformationen dürfen für TRACYL belegt werden. Sie zeichnen sich dadurch aus, dass der mit MD24100 $MC_TRAFO_TYPE_n zugewiesene Wert 512 oder 513 oder 514 ist.

Für diese maximal 3 TRACYL-Transformationen müssen die folgenden Maschinendaten definiert gesetzt werden:

MD24800 $MC_TRACYL_ROT_AX_OFFSET_t

MD24810 $MC_TRACYL_ROT_SIGN_IS_PLUS_t

MD24820 $MC_TRACYL_BASE_TOOL_t

Dabei gibt t die Nummer der vereinbarten TRACYL-Transformation an (maximal 3).

Achskonfiguration

Die folgende Übersicht zeigt den Zusammenhang zwischen den Achsen einer Maschine und den zugehörigen Achsdaten.

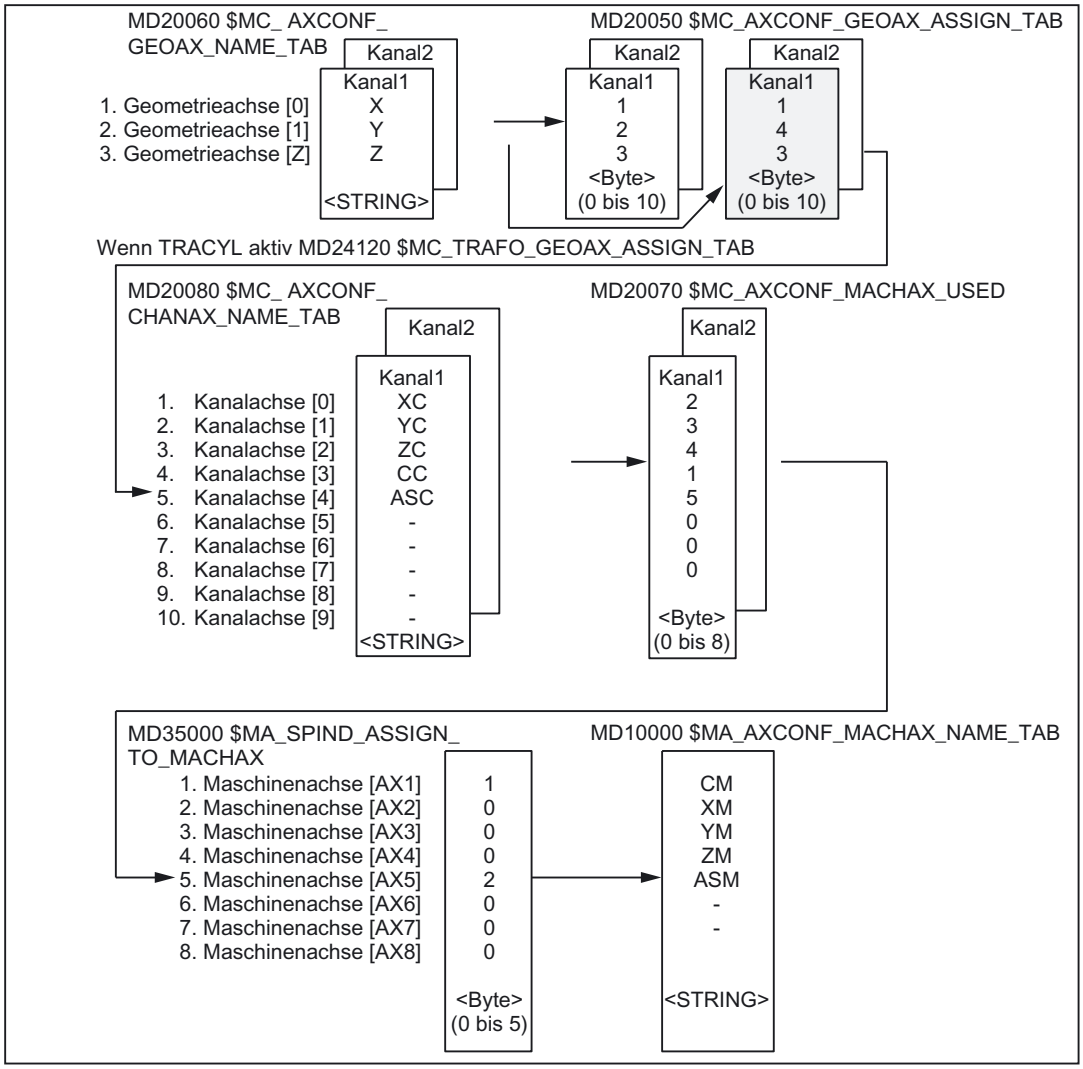


Bild 7-13 Achskonfiguration für das Beispiel im Bild "Nutzbearbeitung am Zylindermantel mit X-Y-Z-C-Kinematik

Die im obigen Bild hervorgehobenen Anordnungen gelten bei `TRACYL` aktiv.

Benennung der Geometrieachsen

Gemäß vorstehender Übersicht über die Achskonfiguration sind die während TRACYL gewünschten Geometrieachsen z. B. mit:

MD20050 $MC_AXCONF_GEOAX_NAME_TAB[0]="X"

MD20050 $MC_AXCONF_GEOAX_NAME_TAB[1]="Y"

MD20050 $MC_AXCONF_GEOAX_NAME_TAB[2]="Z"

zu definieren (Namenswahl laut entspricht auch der Vorbesetzung).

Zuordnung der Geometrieachsen zu Kanalachsen

Es sind die Fälle zu unterscheiden, ob `TRACYL` aktiv ist oder nicht:

- TRACYL nicht aktiv

  Eine Y-Achse wird normal verfahren.

  MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[0]=1

  MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[1]=2

  MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[2]=3

- TRACYL aktiv

  Die Y-Achse wird zur Achse in Umfangsrichtung des Zylinderkoordinatensystems.

  MD20050 $MC_TRAFO_GEOAX_ASSIGN_TAB[0]=1

  MD20050 $MC_TRAFO_GEOAX_ASSIGN_TAB[1]=4

  MD20050 $MC_TRAFO_GEOAX_ASSIGN_TAB[2]=3

Eintrag der Kanalachsen

Es werden die Achsen zugefügt, die nicht zum kartesischen Koordinatensystem gehören.

MD20080 $MC_AXCONF_CHANAX_NAME_TAB[0]="XC"

MD20080 $MC_AXCONF_CHANAX_NAME_TAB[1]="YC"

MD20080 $MC_AXCONF_CHANAX_NAME_TAB[2]="ZC"

MD20080 $MC_AXCONF_CHANAX_NAME_TAB[3]="CC"

MD20080 $MC_AXCONF_CHANAX_NAME_TAB[4]="ASC"

Zuordnung der Kanalachsen zu Maschinenachsen

Mit Bezug auf den cd der Kanalachsen wird der Steuerung mitgeteilt, welcher Maschinenachsen-Nummer die Kanalachsen zugewiesen werden.

MD20070 $MC_AXCONF_MACHAX_USED[0]=2

MD20070 $MC_AXCONF_MACHAX_USED[1]=3

MD20070 $MC_AXCONF_MACHAX_USED[2]=4

MD20070 $MC_AXCONF_MACHAX_USED[3]=1

MD20070 $MC_AXCONF_MACHAX_USED[4]=5

(Einträge entsprechend Bild "Nutbearbeitung am Zylindermantel mit X-Y-Z-C-Kinematik")

Kennzeichnung der Spindeln

Je Maschinenachse wird festgelegt, ob eine Spindel vorliegt (Wert > 0: Spindelnummer) oder Bahnachse (Wert 0).

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[0]=1

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[1]=0

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[2]=0

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[3]=0

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[4]=2

Zuweisung von Namen an Maschinenachsen

Mit Bezug auf den cd der Maschinenachsen wird der Steuerung ein Maschinenachs-Name mitgeteilt:

MD10000 $MN_AXCONF_MACHAX_NAME_TAB[0]="CM"

MD10000 $MN_AXCONF_MACHAX_NAME_TAB[1]="XM"

MD10000 $MN_AXCONF_MACHAX_NAME_TAB[2]="YM"

MD10000 $MN_AXCONF_MACHAX_NAME_TAB[3]="ZM"

MD10000 $MN_AXCONF_MACHAX_NAME_TAB[4]="ASM"

7.3.2 TRACYL-spezifische Einstellungen

Art der Transformation

Der folgende Absatz beschreibt die Vorgabe des Transformationstyps.

TRAFO_TYPE_n

Bei den Transformationsdatensätzen (maximal n = 10) muss der Anwender den Typ der Transformation angeben. Für `TRACYL` ist der WERT 512 zu setzen bei Achskonfiguration1 und 513 für Achskonfiguration 2 oder 514 für ohne Nutwandkorrektur mit zusätzlicher Linearachse. Über einen zusätzlichen Parameter kann der Tansformations-Typ 514 auch mit Nutwandkorrektur aktiviert werden. Siehe hierzu Kapitel "Aktivierung".

Beispiel für WERT 512: MD24100 $MC_TRAFO_TYPE_1=512

Die Einstellung muss getroffen sein, bevor TRACYL(d,t) aufgerufen wird. t ist die Nummer der vereinbarten TRACYL-Transformation.

Für die `TRACYL`-Transformation reichen eine Rundachse und eine Linearachse aus, die senkrecht zur Rundachse angeordnet ist. Mit den Transformationstyp 514 wird eine reale Y-Achse dazu genutzt, um z. B. einen Werkzeugversatz kompensieren zu können.

Transformations-Typ 514 ohne Nutwandkorrektur

Zylindermantelkurventransformation TRAFO_TYPE_n = 514

Verfügt die Maschine über eine weitere Linearachse, die senkrecht zur Rundachse und zur ersten Linearachse ist, so kann der Transformationstyp 514 für Werkzeugkorrekturen zum Herausfahren mit der realen Y-Achse eingesetzt werden. Dabei wird davon ausgegangen, dass der Arbeitsspeicher der zweiten Linearachse klein ist und nicht für das Abfahren des Teileprogramms genutzt werden soll.

Für MD10000 $MC_TRAFO_GEOAX_ASSSIGN_TAB_n gelten die bisherigen Einstellungen.

Nuten mit Nutwandkorrektur

Für die TRACYL-Transformation mit Nutwandkorrektur wird die gewünschte Einrechnung der Werkzeugkorrektur bereit berücksichtigt.

Achsabbildung

Der folgende Absatz beschreibt die Vorgabe der Transformations-Achsenabbildung.

TRAFO_AXES_IN_n

Für den Transformationsdatensatz n sind bei TRACYL drei (bzw. 4) Kanalachsnummern anzugeben:

MD24110 $MC_TRAFO_AXES_IN_1[0]=Kanalachsnummer der Achse radial zur Rundachse

MD24110 $MC_TRAFO_AXES_IN_1[1]=Kanalachsnummer der Rundachse

MD24110 $MC_TRAFO_AXES_IN_1[2]=Kanalachsnummer der Achse parallel zur Rundachse

MD24110 $MC_TRAFO_AXES_IN_1[3]=Kanalachsnummer der Zusatzachse, parallel zur Zylindermantelfläche und senkrecht zur Rundachse (falls Achskonfiguration 2 vorliegt)

Beispiel gemäß Bild "Nutbearbeitung am Zylindermantel mit X-Y-Z-C-Kinematik":

MD24110 $MC_TRAFO_AXES_IN_1[0]=1

MD24110 $MC_TRAFO_AXES_IN_1[1]=4

MD24110 $MC_TRAFO_AXES_IN_1[2]=3

MD24110 $MC_TRAFO_AXES_IN_1[3]=2

Die Einstellung muss getroffen sein, bevor TRACYL(d) oder TRACYL(d,t) aufgerufen wird. Die Achsnummern müssen sich auf die mit folgendem Maschinendatum definierten Kanalachsfolgen beziehen:

MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_n

Nuten ohne Nutwandkorrektur

Für den Transformations-Typ 514 gelten für das $MC_TRAFO_AXES_IN_n[ ] folgende Zuordnungen der Indizes.

Bedeutung der Indizes bezüglich des Basis-Koordinatensystems (BKS):

- [0]: kartesische Achse radial zur Rundachse (falls vorhanden)
- [1]: Achse in Zylindermantelfläche senkrecht zur Rundachse
- [2]: kartesische Achse parallel zur Rundachse
- [3]: Linearachse parallel zur Index 2 in der Maschinengrundstellung

Bedeutung der Indizes bezüglich des Maschinen-Koordinatensystems (MKS):

- [0]: Linearachse radial zur Rundachse (falls vorhanden)
- [1]: Rundachse
- [2]: Linearachse parallel zur Rundachse
- [3]: Linearachse senkrecht zu den Achsen von Index [0] und [1]

Drehlage

Die Drehlage der Achse in der Zylindermantelfläche senkrecht zur Rundachse ist wie folgt zu definieren:

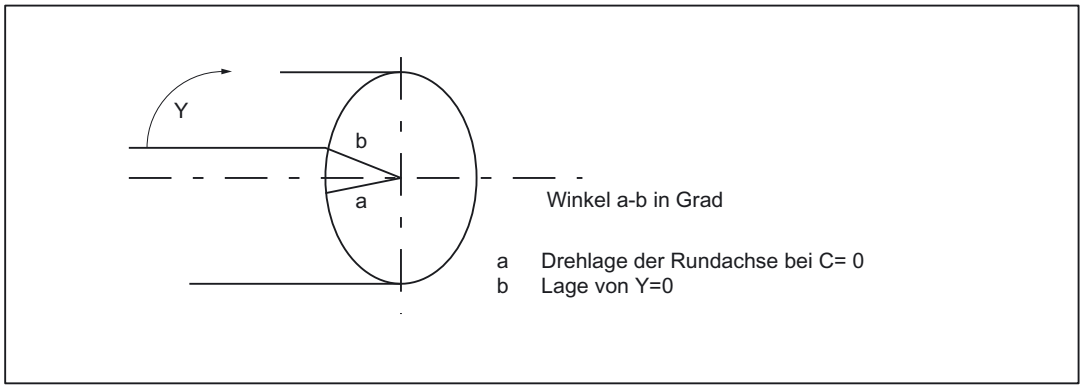


Bild 7-14 Drehlage der Achse in der Zylindermantelfläche

TRACYL_ROT_AX_OFFSET_t

Die Drehlage Mantelfläche gegenüber der definierten Nullstellung der Rundachse wird angegeben mit:

MD24800 $MC_TRACYL_ROT_AX_OFFSET_t=...°

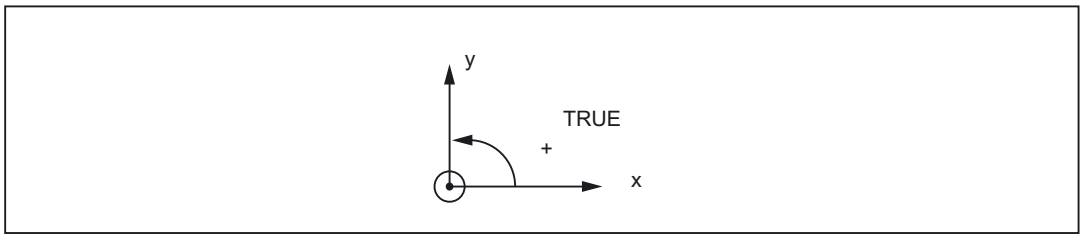
Dabei wird t ersetzt durch die Nummer der in den Transformationsdatensätzen vereinbarten TRACYL-Transformationen (t darf maximal 2 sein).

Drehsinn

Der Drehsinn der Rundachse wird, wie im folgenden Absatz beschrieben, durch Maschinendatum angegeben.

TRACYL_ROT_SIGN_IS_PLUS_t

Ist der Drehsinn der Rundachse in der x-y-Ebene bei Betrachtung gegen die z-Achse im Gegenuhrzeigersinn, so ist das Maschinendatum auf TRUE zu setzen, andernfalls auf FALSE.



MD24810 $MC_TRACYL_ROT_SIGN_IS_PLUS_t=TRUE

Dabei wird t ersetzt durch die Nummer der in den Transformationsdatensätzen vereinbarten TRACYL-Transformationen (t darf maximal 2 sein).

Umschaltbare Geometrieachsen

Ein Umschalten der Geometrieachsen mit `GEOAX( )` wird dem PLC mitgeteilt, indem optional ein über MD einstellbarer M-Code ausgegeben wird.

- MD22534 $MC_TRAFO_CHANGE_M_CODE

  Nummer des M-Codes, der bei einer Transformationsumschaltung am VDI-Interface ausgegeben wird.

Hinweis

Hat dieses Maschinendatum einen der Werte 0 bis 6, 17, 30, so wird kein M-Code ausgegeben.

Literatur:

/FB1/ Funktionshandbuch Grundfunktion; Koordinatensysteme, Achstypen, Achskonfigurationen, Werkstücknahes Istwertsystem, Externe Nullpunktverschiebung (K2)

Lage des Werkzeugnullpunktes

Die Lage des Werkzeugnullpunktes zum Ursprung des kartesischen Koordinatensystems wird, wie im folgenden Absatz beschrieben, durch Maschinendatum angegeben.

TRACYL_BASE_TOOL_t

MD24820 $MC_TRACYL_BASE_TOOL_t

Mit dem obigen Maschinendatum wird der Steuerung mitgeteilt, in welcher Lage der Werkzeugnullpunkt bezogen auf den Ursprung des bei `TRACYL` vereinbarten Zylinder-Koordinatensystems liegt. Das Maschinendatum hat drei Komponenten für die drei Achsen X, Y, Z des Maschinen-Koordinatensystems.

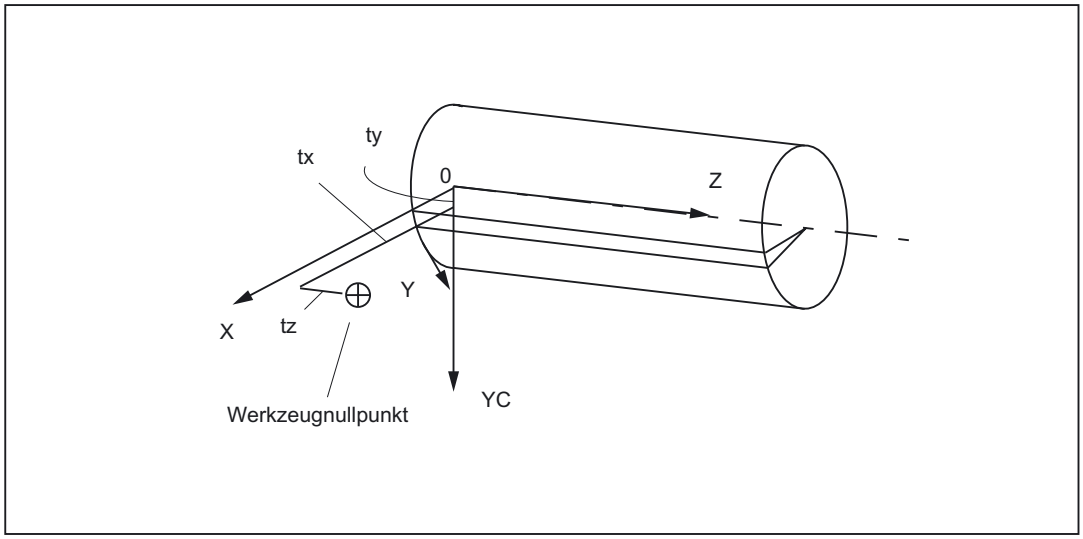


Bild 7-15 Lage des Werkzeugnullpunktes zum Nullpunkt der Maschine

Beispiel:

MD24820 $MC_TRACYL_BASE_TOOL_t[0]=tx

MD24820 $MC_TRACYL_BASE_TOOL_t[1]=ty

MD24820 $MC_TRACYL_BASE_TOOL_t[2]=tz

Dabei wird t ersetzt durch die Nummer der in den Transformationsdatensätzen vereinbarten TRACYL-Transformationen (t darf maximal 2 sein).

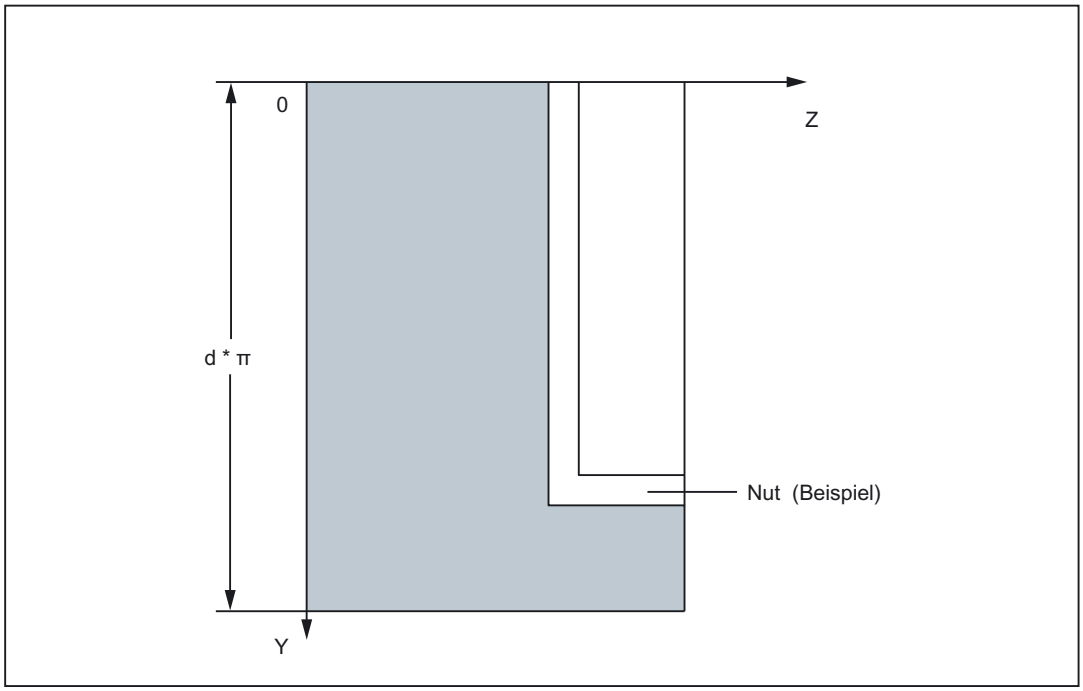


Bild 7-16 Zylinderkoordinatensystem

7.3.3 Aktivieren von TRACYL

TRACYL

Nachdem die in den vorausgehenden Abschnitten beschriebenen Einstellungen getroffen wurden, kann die TRACYL-Funktion aktiviert werden:

TRACYL(d)

oder

TRACYL(d,t) TRACYL(Bezugsdurchmesser, Tracyl-Datensatz)

Mit TRACYL(d) wird die erste vereinbarte TRACYL-Funktion aktiviert. TRACYL(d,t) aktiviert die t. vereinbarte TRACYL-Funktion. t darf maximal 2 sein. Der Wert d steht für den aktuellen Durchmesser des zu bearbeitenden Zylinders.

Zwischen dem Aktivieren und dem nachfolgend beschriebenen Ausschalten können die Verfahrbewegungen für die Achsen des Zylinder-Koordinatensystems programmiert werden.

Transformations-Typ 514 mit Nutwandkorrektur

Der Transformationstyp 514 erhält einen zusätzlichen Aufrufparameter indem mit einem dritten Parameter die TRACYL-Transformation mit Nutwandkorrektur angewählt werden kann.

TRACYL(Bezugsdurchmesser, Tracyl-Datensatz, Nutwandkorrektur).

- Bezugsdurchmesser: Pflichtparameter (muss immer angeben werden)

  Wertebereich: >0
- Tracyl-Datensatz: Optionaler Parameter, vorangewählt ist 1

  Wertebereich: 1,2
- Nutwandkorrektur: Optionaler Parameter, vorangewählt ist der in Maschinendaten angegebene Wert gemäß

  MD24808 $MC_TRACYL_DEFAULT_MODE_1) bzw.

  MD24858 $MC_TRACYL_DEFAULT_MODE_2)

  Wertebereich: 0,1

7.3.4 Ausschalten der TRACYL-Funktion

TRAFOOF

Das Schlüsselwort TRAFOOF schaltet eine aktive Transformation wieder aus. Bei ausgeschalteter Transformation ist das Basis-Koordinatensystem wieder mit dem Maschinen-Koordinatensystem identisch.

Eine aktive Transformation TRACYL wird ebenfalls ausgeschaltet, wenn im jeweiligen Kanal eine der übrigen Transformationen aktiviert wird (z. B. TRANSMIT, TRAANG, TRAORI).

Literatur:

/FB3/ Funktionshandbuch Sonderfuntion; 5-Achs-Transformation (F2)

7.3.5 Besondere Reaktionen bei TRACYL

An- und Abwahl der Transformation sind über Teileprogramm bzw. MDA möglich.

Bei Anwahl beachten

- Ein Bewegungszwischensatz wird nicht eingefügt (Phasen/Radien).
- Eine Splinesatzfolge muss abgeschlossen sein.
- Werkzeugradiuskorrektur muss abgewählt sein.
- Der vor TRACYL wirksame Frame wird von Steuerung abgewählt

  (Entspricht programmiertem Frame rücksetzen G500).

- Eine aktive Arbeitsfeldbegrenzung wird für die von der Transformation betroffenen Achsen von der Steuerung abgewählt

  (Entspricht programmiertem `WALIMOF`).
- Bahnsteuerbetrieb und Überschleifen werden unterbrochen.
- DRF-Verschiebungen müssen vom Bediener gelöscht worden sein.
- Bei Zylindermantelkurventransformation mit Nutwandkorrektur (Achskonfiguration 2, TRAFO_TYPE_n=513) sollte die für die Korrektur verwendete Achse (TRAFO_AXES_IN_n[3]) auf Null (y=0) stehen, damit die Nut mittig zur programmierten Nutmittellinie gefertigt wird.

Bei Abwahl beachten

Bei der Abwahl sind dieselben Punkte zu beachten wie bei der Anwahl.

Einschränkungen bei aktivem TRACYL

Die folgenden Einschränkungen sind bei aktivem TRACYL zu beachten:

Werkzeugwechsel

Ein Werkzeugwechsel ist nur bei abgewählter Werkzeugradiuskorrektur zulässig.

Randbedingungen TRACYL ohne Nutwandkorrektur

Im Teileprogramm muss bei aktiven `TRANSMIT` der Kanalbezeichner von posBCS[ax[3]] einen anderen Namen haben, wie die Geometrieachsen. Wird posBCS[ax[3]] nur außerhalb `TRACYL` geschrieben, dann gilt diese Einschränkung nicht, wenn die Achse einer Geometrieachse zugeordnet war. Bei aktivem `TRACYL` wird keine Konturinformation über ax[3] verarbeitet.

Frame

Alle Anweisungen, die sich nur auf das Basis-Koordinatensystem beziehen, sind erlaubt (`FRAME`, Werkzeugradiuskorrektur). Ein Framewechsel bei G91 (Kettenmaß) wird aber - anders als bei inaktiver Transformation - nicht gesondert behandelt. Das zu fahrende Inkrement wird im Werkstück-Koordinatensystem des neuen Frames ausgewertet, unabhängig davon, welches Frame im Vorgängersatz wirkte.

Rundachse

Die Rundachse kann nicht programmiert werden, da sie von einer Geometrie-Achse belegt wird und somit als Kanalachse nicht direkt programmierbar ist.

Erweiterungen

Eine Verschiebung der Rundachse CM kann z. B. durch eine Kompensation der Schräglage eines Werkstückes in einem Frame innerhalb der Framekette eingetragen werden und führt dann zu entsprechenden x- und y-Werten gemäß des folgenden Bildes.

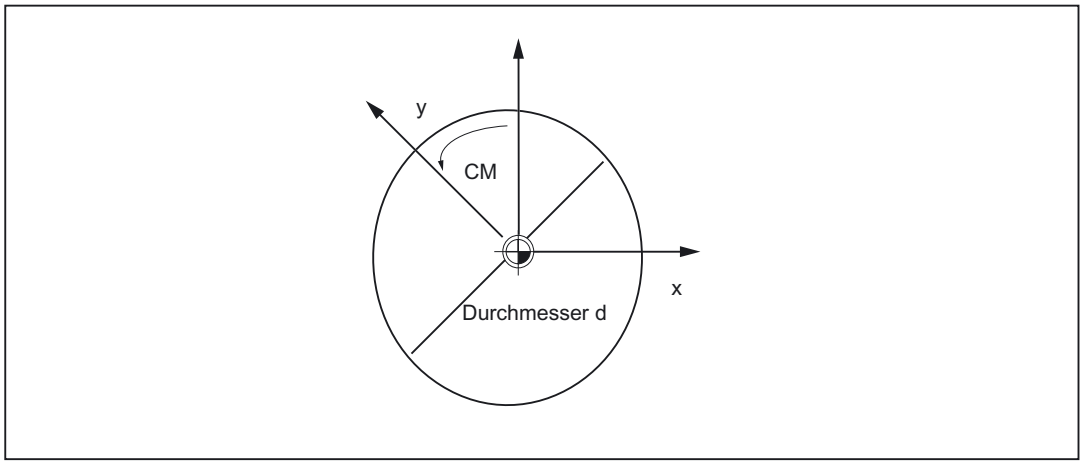


Bild 7-17 Rundachsverschiebung bei TRACYL

Diese Verschiebung kann auch in der Transformation als Offset der Rundachse oder als y-Verschiebung berücksichtigt werden. Damit das axiale Gesamtframe der Tracyl-Rundachse, d. h. die Translation, die Feinverschiebung, die Spiegelung und die Skalierung in der Transformation übernimmt, sind folgende Einstellungen erforderlich:

MD24805 $MC_TRACYL_ROT_AX_FRAME_1 = 1

MD24855 $MC_TRACYL_ROT_AX_FRAME_2 = 1

Hinweis

Änderungen der Achszuordnungen werden jeweils bei der An- und Abwahl der Transformation umgesetzt. Weitere Informationen zur axialen Verschiebung der Rundachse bis zum ENS als Verschiebung auf der Manteloberfläche siehe:

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; Koordinatensysteme, Frames (K2)

Achsnutzung

Die Achsen:

- in Zylindermantelfläche senkrecht zur Rundachse (Y) und
- Zusatzachse (YC)

dürfen nicht als Positionierachse bzw. Pendelachse verwendet werden.

Ausschlüsse

Von der Transformation betroffene Achsen können nicht verwendet werden:

- als Preset-Achse (Alarm)
- für das Fixpunkt anfahren (Alarm)
- zum Referenzieren (Alarm)

Unterbrechung des Teileprogramms

Die folgenden Punkte sind bei Unterbrechung der Teileprogrammbearbeitung im Zusammenhang mit TRACYL zu beachten:

Automatik nach Jog

Wird eine Teileprogrammbearbeitung bei aktiver Transformation unterbrochen und mit Jog verfahren, so ist bei Wiederanwahl von Automatik zu beachten:

- Die Transformation ist auch im Anfahrsatz von der aktuellen Position zur Unterbrechungsstelle aktiv. Eine Kollisionsüberwachung wird nicht durchgeführt.

⚠ WARNUNG
Der Bediener ist dafür verantwortlich, dass das Werkzeug problemlos rückpositioniert werden kann.

START nach RESET

Wird eine Teileprogrammbearbeitung mit `RESET` abgebrochen und mit `START` neu begonnen, so ist zu beachten:

- Nur wenn zu Teileprogrammbeginn alle Achsen über einen Linearsatz (G0 bzw. G1) auf eine definierte Position gefahren werden, wird das restliche Teileprogramm reproduzierbar abgefahren. Ein bei `RESET` aktives Werkzeug wird ggf. von der Steuerung nicht mehr beachtet (festlegbar über Maschinendaten).

7.3.6 Jog

Besonderheiten bei Jog

Bei Zylindermanteltransformation mit Nutwandkorrektur ($MC_TRAFO_TYPE = 513) und Jog ist zu beachten, das die Achsen abhängig vom vorhergehenden Zustand in `AUTOMATIK` verfahren werden. Bei aktiver Nutwandkorrektur bewegen sich die Achsen demnach anders als bei abgewählter Korrektur. Damit kann nach einer Teileprogrammunterbrechung das Teileprogramm wieder fortgesetzt werden (`REPOS`).

7.4 TRAANG (Option)

Hinweis

Für die im Folgenden beschriebene Transformation `TRAANG` müssen die während aktiver Transformation vergebenen Maschinenachsnamen, Kanalachsnamen und Geometrieachsnamen unterschiedlich sein. Vergl.

MD10000 $MN_AXCONF_MACHAX_NAME_TAB,

MD20080 $MC_AXCONF_CHANAX_NAME_TAB,

MD20060 $MC_AXCONF_GEOAX_NAME_TAB.

Sonst sind keine eindeutigen Zuordnungen gegeben.

Aufgabenstellung

Schleifbearbeitung

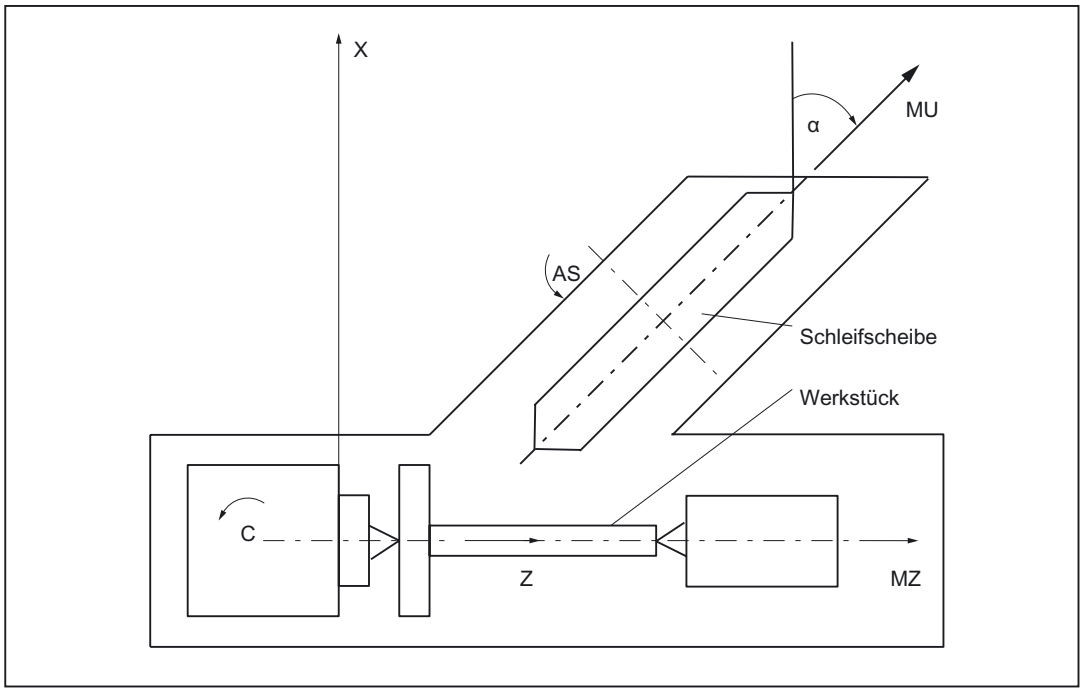


Bild 7-18 Maschine mit schrägstehender Zustellachse

Legende:

X, Z: Kartesisches Koordinatensystem für die Programmierung

C: Rundachse

AS: Arbeitsspindel

MZ: Maschinenachse (linear)

MU: Schräge Achse

Folgende Bearbeitungsvielfalt ist möglich:

- Längsschleifen
- Planschleifen
- Schleifen einer bestimmten Kontur
- Schrägeinstechschleifen

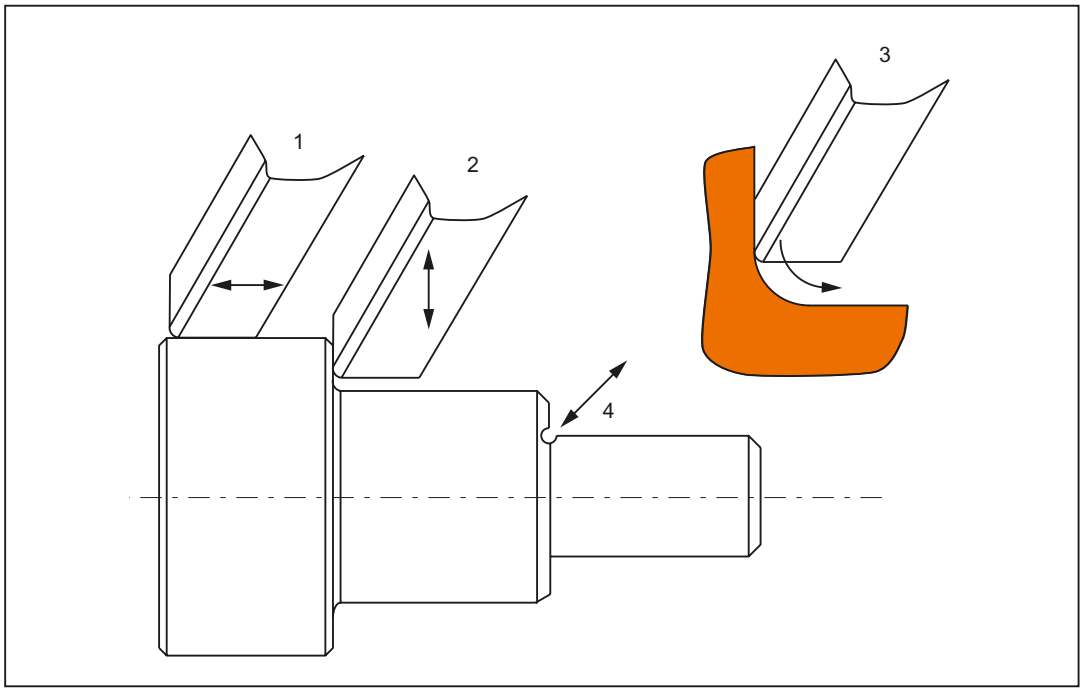


Bild 7-19 Mögliche Schleifbearbeitungen

7.4.1 Voraussetzungen für TRAANG (Schräge Achse)

Achskonfiguration

Um im kartesischen Koordinatensystem (siehe Bild "Maschine mit schrägstehender Zustellachse": X-Y-Z) programmieren zu können, muss der Steuerung der Zusammenhang zwischen diesem Koordinatensystem und den tatsächlich existierenden Maschinenachsen (MU, MZ) mitgeteilt werden:

- Benennung der Geometrieachsen
- Zuordnung der Geometrieachsen zu Kanalachsen
  - allgemeiner Fall (Schräge Achse nicht aktiv)
  - Schräge Achse aktiv
- Zuordnung der Kanalachsen zu den Maschinenachsnummern
- Kennzeichnung der Spindeln
- Zuweisung von Maschinenachsnamen

Das Vorgehen entspricht mit Ausnahme des Punktes "Schräge Achse aktiv" dem Vorgehen bei der normalen Achskonfiguration.

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; Koordinatensysteme, Achstypen, Achskonfigurationen, Werkstücknahes Istwertsystem, Externe Nullpunktverschiebung (K2).

Anzahl Transformationen

Im System können bis zu 10 Transformationsdatensätze je Kanal definiert werden. Die Maschinendaten-Namen dieser Transformationen beginnen mit $MC_TRAFO .. und enden mit ... _n, wobei n für eine Ziffer von 1 bis 10 steht. Die folgenden Abschnitte beschreiben unter anderem diese Daten:

MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_n

MD24100 $MC_TRAFO_TYPE_n

MD24110 $MC_TRAFO_AXES_IN_n

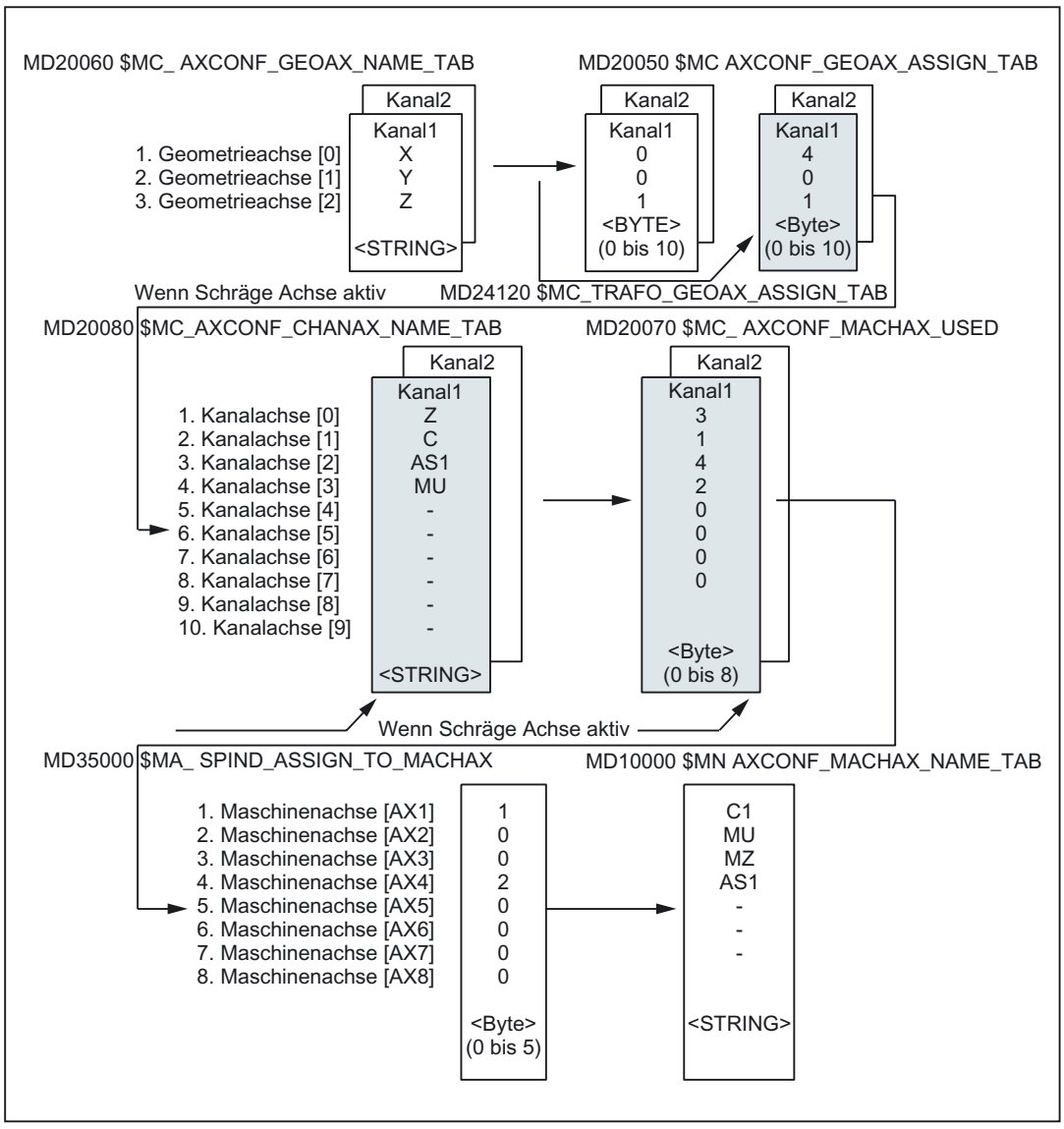
Anzahl Schräge Achse

Zwei der 10 zulässigen Datenstrukturen für Transformationen dürfen für Schräge Achse belegt werden. Sie zeichnen sich dadurch aus, dass der mit

MD24100 $MC_TRAFO_TYPE_n zugewiesene Wert 1024 ist.

Achskonfiguration

Die im Bild dargestellten Achsen der Schleifmaschine sind folgendermaßen in den Maschinendaten eingetragen:



Achskonfiguration für das Beispiel im Bild "Maschine mit schrägstehender Zustellachse"

Die im obigen Bild hervorgehobenen Anordnungen gelten bei `TRAANG` aktiv.

7.4.2 TRAANG-spezifische Einstellungen

Art der Transformation

TRAFO_TYPE_n

Bei den Transformationsdatensätzen (maximal n = 10) muss der Anwender den Typ der Transformation im folgenden Maschinendatum angeben:

MD24100 $MC_TRAFO_TYPE_n

Für Schräge Achse ist der Wert 1024 vorgesehen:

MD24100 $MC_TRAFO_TYPE_1=1024

Achsabbildung

TRAFO_AXES_IN_n

Für den Transformationsdatensatz n sind zwei Kanalachsnummern anzugeben:

MD24110 $MC_TRAFO_AXES_IN_1[0]=4 ; Kanalachsnummer der schrägen Achse

MD24110 $MC_TRAFO_AXES_IN_1[1]=1 ; Kanalachsnummer der parallelen Achse zu Z

MD24110 $MC_TRAFO_AXES_IN_1[2]=0 ; Kanalachsnummer nicht aktiv

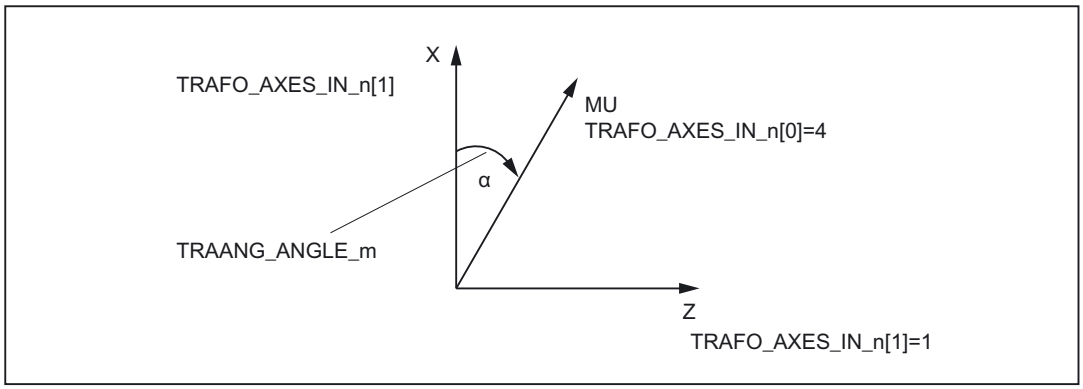


Bild 7-20 Parameter TRAANG_ANGLE_m

Zuordnung der Geometrieachsen zu Kanalachsen

Beispiel:

MD24430 $MC_TRAFO_TYPE_5 = 8192 Verkettung

MD24110 $MC_TRAFO_AXIS_IN_1[0..x]

MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[0] =1 Definition Geoachszuordnung von Trafo 1

MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[1] =6 Definition Geoachszuordnung von Trafo 1

MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[2] =3 Definition Geoachszuordnung von Trafo 1

MD24996 $MC_TRACON_CHAIN_2[0] = 2 Eingangsgrößen in TRACON

MD24996 $MC_TRACON_CHAIN_2[1] = 3 Eingangsgrößen in TRACON

MD24996 $MC_TRACON_CHAIN_2[2] = 0 Eingangsgrößen in TRACON

MD24996 $MC_TRACON_CHAIN_2[3] = 0 Eingangsgrößen in TRACON

Winkel der Schrägen Achse

TRAANG_ANGLE_m

Mit dem folgenden Maschinendatum wird der Steuerung mitgeteilt, welcher Winkel zwischen einer Maschinenachse und der schrägen Achse in Grad besteht:

MD24700 $MC_TRAANG_ANGLE_m

MD24700 $MC_TRAANG_ANGLE_m = Winkel zwischen einer kartesischen Achse und der zugeordneten schrägstehenden Maschinenachse in Grad. Der Winkel wird positiv im Uhrzeigersinn gezählt (siehe Bild "Maschine mit schrägstehender Zustellachse", Winkel α).

Dabei wird m ersetzt durch die Nummer der in den Transformationsdatensätzen vereinbarten TRAANG-Transformation. m darf maximal 2 sein.

Zulässiger Winkelbereich

Der zulässige Winkelbereich beträgt:

-90° < TRAANG_ANGLE_m < 0°

0° < TRAANG_ANGLE_m < 90°

Für 0° ist keine Transformation erforderlich.

Bei +/- 90° verläuft die Schräge Achse parallel zur zweiten Linearachse.

Lage des Werkzeugnullpunktes

TRAANG_BASE_TOOL_m

Mit dem folgenden Maschinendatum kann der Steuerung mitgeteilt werden, in welcher Lage der Werkzeugnullpunkt bezogen auf den Ursprung des bei der Funktion Schräge Achse vereinbarten Koordinatensystems liegt:

MD24710 $MC_TRAANG_BASE_TOOL_m

Das Maschinendatum hat drei Komponenten für die 3 Achsen des kartesischen Koordinatensystems.

Standardmäßig wird Null vorgegeben.

Die Korrekturen werden bei Änderung des Winkels nicht umgerechnet.

Optimierung der Geschwindigkeitsführung

Zur Optimierung der Geschwindigkeitsführung in Jog und im Positionier- bzw. Pendelbetrieb dienen folgende Maschinendaten:

TRAANG_PARALLEL_VELO_RES_m

Mit dem Maschinendatum MD24720 $MC_TRAANG_PARALLEL_VELO_RES_m wird die Geschwindigkeitsreserve eingestellt, die auf der parallelen Achse für die Ausgleichsbewegung bereitgehalten wird (siehe folgendes Maschinendatum:)

MD24110 $MC_TRAFO_AXES_IN_n[1]

Wertebereich: 0 ... 1

0: Mit dem Wert 0 legt die Steuerung die Reserve selbsttätig fest: die Achsen werden gleichgewichtig begrenzt (= Standardwert).

>0: Mit Werten >0 wird die Reserve fest auf das folgende Maschinendatum als zulässiger Maschinenachsgeschwindigkeitswert der parallelen Achse eingestellt:

MD24720 $MC_TRAANG_PARALLEL_VELO_RES_m

Das Geschwindigkeitsverhalten der senkrechten Achse wird von der Steuerung anhand der Reserve bestimmt.

TRAANG_PARALLEL_ACCEL_RES_m

Mit folgendem Maschinendatum wird die Achsbeschleunigungsreserve eingestellt, die auf der parallelen Achse (siehe MD24110 $MC_TRAFO_AXES_IN_n[1]) für die Ausgleichsbewegung bereitgehalten wird:

MD24721 $MC_TRAANG_PARALLEL_ACCEL_RES_m

Wertebereich: 0 ... 1

0: Mit dem Wert 0 legt die Steuerung die Reserve selbsttätig fest: die Achsen werden gleichgewichtig beschleunigt. (= Standardwert)

>0: Mit Werten >0 wird die Beschleunigung fest auf das folgende Maschinendatum als zulässiger Maschinenachsbeschleunigungswert der parallelen Achse eingestellt

MD24721 $MC_TRAANG_PARALLEL_ACCEL_RES_m

Das Geschwindigkeitsverhalten der senkrechten Achse wird von der Steuerung anhand der Reserve bestimmt.

Umschaltbare Geometrieachsen

Ein Umschalten der Geometrieachsen mit GEOAX( ) wird dem PLC mitgeteilt, in dem optional ein über MD einstellbarer M-Code ausgegeben wird.

- MD22534 $MC_TRAFO_CHANGE_M_CODE

  Nummer des M-Codes, der bei einer Transformationsumschaltung am VDI-Interface ausgegeben wird.

  Hinweis

  Hat dieses Maschinendatum einen der Werte 0 bis 6, 17, 30, wird kein M-Code ausgegeben.

Literatur:

/FB1/ Funktionshandbuch Grundfunktion; Koordinatensysteme, Achstypen, Achskonfigurationen, Werkstücknahes Istwertsystem, Externe Nullpunktverschiebung (K2).

7.4.3 Aktivieren von TRAANG

TRAANG(a)

Nachdem die in den vorausgehenden Abschnitten beschriebenen Einstellungen getroffen wurden, kann die TRAANG-Funktion aktiviert werden:

TRAANG(a)

oder

TRAANG(a,n)

Mit TRAANG(a) wird die erste vereinbarte Transformation Schräge Achse aktiviert.

Mit a kann der Winkel der schrägstehenden Achse angegeben werden.

- Wird a weggelassen oder Null eingetragen, wird die Transformation mit der Parametrierung der vorhergehenden Anwahl aktiviert.

  Bei der ersten Anwahl gilt die Vorbelegung gemäß den Maschinendaten.
- Wird a (Winkel) weggelassen (z. B. TRAANG(), TRAANG(,n)), wird die Transformation mit der Parametrierung der vorhergehenden Anwahl aktiviert. Bei der ersten Anwahl gilt die Vorbelegung gemäß den Maschinendaten. Ein Winkel a = 0 (z. B. TRAANG(0), TRAANG(0,n)) ist eine gültige Parametrierung und entspricht nicht mehr dem Weglassen des Parameters bei älteren Versionen. Zulässiger Wertebereich ist für a: -90 Grad < a < + 90 Grad.

  TRAANG(a,n) aktiviert die n. vereinbarte Transformation Schräge Achse.

  Diese Form wird nur benötigt, wenn im Kanal mehrere Transformationen aktiviert sind. n darf maximal 2 sein.

Programmiervarianten

TRAANG(a,1) == TRAANG(a,0) == TRAANG(a,) == TRAANG(a)

Zwischen dem Aktivieren und dem nachfolgend beschriebenen Ausschalten müssen die Verfahrbewegungen für die Achsen des kartesischen Koordinatensystems programmiert werden.

7.4.4 Ausschalten von TRAANG

TRAFOOF

Das Schlüsselwort TRAFOOF schaltet eine aktive Transformation wieder aus. Bei ausgeschalteter Transformation ist das Basis-Koordinatensystem wieder mit dem Maschinen-Koordinatensystem identisch.

Eine aktive Transformation TRAANG wird ebenfalls ausgeschaltet, wenn im jeweiligen Kanal eine der übrigen Transformationen aktiviert wird (z. B. TRACYL, TRANSMIT, TRAORI).

Literatur:

/FB3/ Funktionshandbuch Sonderfunktion; 5-Achs-Transformation(F2).

7.4.5 Besondere Reaktionen bei TRAANG

An- und Abwahl der Transformation sind über Teileprogramm bzw. MDA möglich.

Anwahl und Abwahl

- Ein Bewegungszwischensatz wird nicht eingefügt (Phasen/Radien).
- Eine Spline-Satzfolge muss abgeschlossen sein.
- Werkzeugradiuskorrektur muss abgewählt sein.
- Der aktuelle Frame wird von der Steuerung abgewählt

  (entspricht programmiertem G500).
- Eine aktive Arbeitsfeldbegrenzung wird für die von der Transformation betroffenen Achsen von der Steuerung abgewählt (entspricht programmiertem `WALIMOF`).
- Eine aktivierte Werkzeuglängenkorrektur wird von der Steuerung in die Transformation übernommen.
- Bahnsteuerbetrieb und Überschleifen werden unterbrochen.
- DRF-Verschiebungen müssen vom Bediener gelöscht worden sein.
- Alle in dem Maschinendatum MD24110 $MC_TRAFO_AXES_IN_n angegebene Achsen müssen satzbezogen synchronisiert sein (z. B. keine Verfahranweisung mit POSA...).

Einschränkungen

Werkzeugwechsel

Ein Werkzeugwechsel ist nur bei abgewählter Werkzeugradiuskorrektur zulässig.

Frame

Alle Anweisungen, die sich auf das Werkstück-Koordinatensystem beziehen, sind erlaubt (`FRAME`, Werkzeugradiuskorrektur). Ein Frame-Wechsel bei G91 (Kettenmaß) wird aber - anders als bei inaktiver Transformation - nicht gesondert behandelt. Das zu fahrende Inkrement wird im Werkstück-Koordinatensystem des neuen Frames ausgewertet - unabhängig davon, welcher Frame im Vorgängersatz wirkte.

Erweiterungen

Bei An- und Abwahl von `TRAANG` kann sich die Geometrie-Achs- Zuordnung zu den Kanalachsen ändern. Diese geometrischen Konturanteile können vom Anwender zum axialen Frame als Translation, Drehung, Skalierung und Spiegelung zur x-und z-Ebene bezüglich der schrägstehenden Zustellachse beaufschlagt werden.

Weitere Informationen zu diesen Framekorrekturen bei Transformationen siehe:

Literatur:

/FB1/ Funktionshandbuch Grundfunktion; Achsen, Koordinatensysteme, Frames (K2)

Ausschlüsse

Von der Transformation betroffene Achsen können nicht verwendet werden:

- als Preset-Achse (Alarm),
- für das Fixpunkt anfahren (Alarm),
- zum Referieren (Alarm).

Geschwindigkeitsführung

Die Geschwindigkeitsüberwachung bei `TRAANG` wird standardgemäß im Vorlauf durchgeführt.

Die Überwachung und Begrenzung im Hauptlauf wird aktiviert:

- im `AUTOMATIK`-Betrieb, wenn eine Positionier- oder Pendelachse programmiert wurde, die in die Transformation eingeht.
- beim Wechsel in den JOG-Betrieb.

Die Überwachung wird wieder vom Hauptlauf in den Vorlauf übernommen, wenn der Vorlauf neu auf den Hauptlauf synchronisiert wird (derzeit z. B. beim Wechsel von JOG nach AUTOMATIK).

Die Geschwindigkeitsüberwachung im Vorlauf nützt die dynamischen Begrenzungen der Maschine besser aus als die entsprechende im Hauptlauf.

Ebenso an Maschinen, bei denen, bei einer Schrägbearbeitung,

7.4.6 Schräge Achse programmieren (G05, G07)

Funktion

Folgende Funktionen stehen zur Verfügung:

- Positionen im kartesischen Koordinatensystem programmieren und anzeigen
- Werkzeugkorrektur und Nullpunktverschiebung kartesisch einrechnen
- Winkel für die Schräge Achse im NC-Programm programmieren
- die Startposition zum Schrägeinstechen (G07) anfahren
- Schrägeinstechen (G05)
- Im JOG-Betrieb die Schleifscheibe wahlweise kartesisch oder in Richtung der Schrägen Achse bewegen (Anzeige bleibt kartesisch).

  Die Anwahl erfolgt über den DB21-28 DBX29.4 "PTP-Fahren". Bei aktiviertem PTP-Fahren bewegt sich nur die reale U-Achse, die Anzeige der Z-Achse wird aktualisiert.

Programmierung

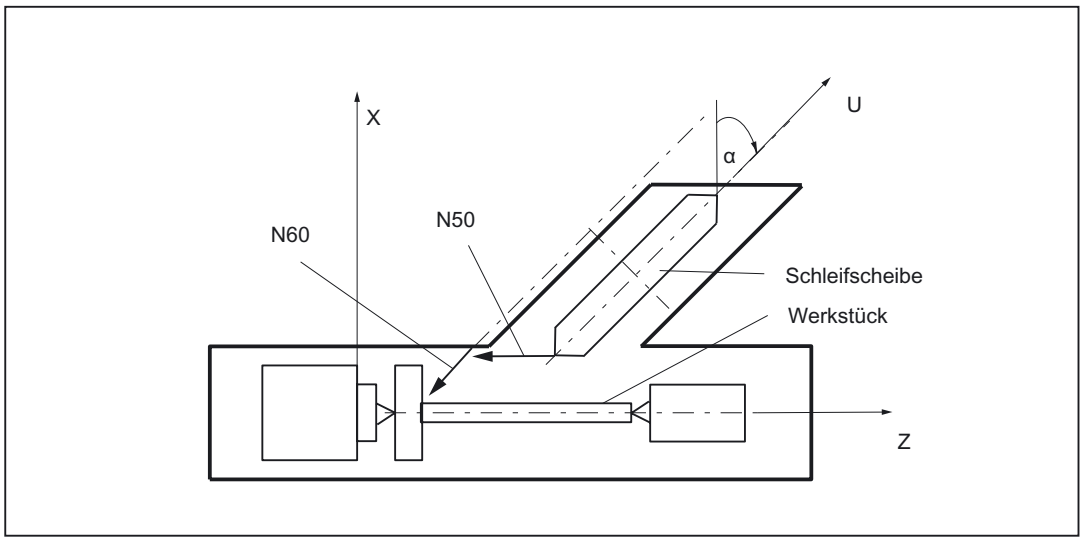


Bild 7-21 Maschine mit schrägstehender Zustellachse

Beispiel:

```
N...                                    Winkel für die schräge Achse progr
N50 G07 X70 Z40 F4000                   Startposition anfahren
N60 G05 X70 F100                        Schrägeinstechen
N...
```

Randbedingungen

- Die Anwahl der Funktion "Kartesisches PTP-Fahren" im JOG-Betrieb (Bewegung entsprechend G05) ist nur bei aktiver Transformation (TRAANG) sinnvoll. Die Einstellung im MD20140 $MC_TRAFO_RESET_VALUE ist zu beachten.
- REPOS-Verschiebungen müssen im JOG-Betrieb kartesisch zurückgefahren werden, während "PTP-Fahren" nicht aktiv ist.
- Das Überfahren der kartesischen Arbeitsfeldbegrenzung wird im Jog-Betrieb bei aktivem "PTP-Fahren" überwacht, die entsprechende Achse wird vorher gebremst. Ist "PTP-Fahren" nicht aktiv, kann die Achse exakt bis zur Arbeitsfeldbegrenzung gefahren werden.

Siehe auch

Kartesisches PTP-Fahren (Seite 561)

7.5 Verkettete Transformationen

Einführung

Es ist möglich, die hier beschriebenen kinematischen Transformationen mit einer weiteren Transformation des Typs "Schräge Achse" zu verketten:

- TRANSMIT
- TRACYL
- TRAANG (Schräge Achse)

sowie die in

Literatur:

/FB3/ Funktionshandbuch Sonderfunktion; 3- bis 5-Achs-Transformationen (F2)

beschriebenen

- Orientierungstransformationen
- Kardanischer Fräskopf

Anwendungen

Aus der Fülle der denkbaren verketteten Transformationen hier ein Ausschnitt:

- Schleifen von Konturen, die als Mantellinie einer Zylinderabwicklung programmiert wurden (TRACYL) mit einer schräg stehenden Schleifscheibe, z. B. Werkzeugschleifen.
- Feinbearbeitung einer mit TRANSMIT erzeugten, nicht runden Kontur mit schräg stehender Schleifscheibe.

Hinweis

Für die im folgenden beschriebene Transformationen müssen die während aktiver Transformation vergebenen Maschinenachsnamen, Kanalachsnamen und Geometrieachsnamen unterschiedlich sein. Vgl. folgende Maschinendaten:

MD10000: AXCONF_MACHAX_NAME_TAB

MD20080: AXCONF_CHANAX_NAME_TAB

MD20060: AXCONF_GEOAX_NAME_TAB

Sonst sind keine eindeutigen Zuordnungen gegeben.

Achskonfiguration

Für die verkettete Transformation sind folgende Konfigurationsmaßnamen nötig:

- Benennung der Geometrieachsen
- Benennung der Kanalachsen
- Zuordnung der Geometrieachsen zu den Kanalachsen
  - allgemeiner Fall (keine Transformation aktiv)
- Zuordnung der Kanalachsen zu den Maschinenachsen-Nummern

- Kennzeichnung von Spindel, Rotation, Modulo für Achsen
- Zuweisung von Maschinenachsnamen
- Transformationsspezifische Angaben (je einzelneTransformation **und** je **verkettete** Transformation)
  - Transformationstyp
  - in die Transformation eingehende Achsen
  - Zuordnung der Geometrieachsen zu den Kanalachsen bei aktiver Transformation
  - je nach Transformation noch Drehlage des Koordinatensystems, Drehsinn, Werkzeugnullpunkt bez. ursprüngliches Koordinatensystem, Winkel der schrägen Achse usw.

Anzahl Transformationen

Im System können bis zu zehn Transformationsdatensätze je Kanal definiert werden. Die Maschinendaten-Namen dieser Transformationen beginnen mit $MC_TRAFO .. und enden mit ... _n, wobei n für eine Ziffer von 1 bis 10 steht.

Anzahl verketteter Transformationen

Innerhalb der maximal 10 Transformationen eines Kanals dürfen maximal **zwei verkettete** Transformationen definiert werden.

Reihenfolge der Transformationen

Bei der Projektierung der Maschinendaten müssen die Angaben zu den einzelnen Transformationen (die ggf. Bestandteil von verketteten Transformationen werden sollen) vor den Angaben zu den verketteten Transformationen abgesetzt werden.

Kettungsreihenfolge

Bei verketteten Transformationen muss die zweite Transformation "Schräge Achse" (`TRAANG`) sein.

Kettungsrichtung

Die erste der verketteten Transformationen hat das BKS als Eingang, die zweite der verketteten Transformationen hat das MKS als Ausgang.

Randbedingungen

Die in den Einzelbeschreibungen für die Transformationen angegebenen Randbedingungen und Sonderfälle sind auch bei der Benutzung innerhalb einer Verkettung zu beachten.

7.5.1 Aktivierung verketteter Transformationen

TRACON

Eine verkettete Transformation wird aktiviert mit:

TRACON(trf, par)

- trf:

  Nummer der verketteten Transformation:0 oder 1 für erste/einzige verkettete Transformation. Ist an dieser Stelle nichts programmiert, ist das gleichbedeutend mit der Angabe des Wertes 0 oder 1, d. h. es wird die erste/einzige Transformation aktiviert.2 für die zweite verkettete Transformation. (Werte ungleich 0 - 2 erzeugen einen Fehleralarm).

- par

  Ein oder mehrere durch Komma getrennte Parameter für die Transformationen in der Verkettung, die Parameter erwarten, z. B. Winkel der schrägen Achse. Bei nicht gesetzten Parametern werden die Voreinstellungen oder die zuletzt benutzten Parameter wirksam. Durch Kommasetzung muss dafür gesorgt werden, dass die angegebenen Parameter in der Reihenfolge ausgewertet werden, in der sie erwartet werden, wenn für vorher stehende Parameter Voreinstellungen wirken sollen. Insbesondere muss bei Angabe mindestens eines Parameters vor diesem ein Komma stehen, auch wenn die Angabe von trf nicht notwendig ist, also beispielsweise `RACON( , 3.7)`.

Eine zuvor aktivierte andere Transformation wird durch `TRACON()` implizit ausgeschaltet.

7.5.2 Ausschalten der verketteten Transformation

TRAFOOF

Eine verkettete Transformation wird mit `TRAFOOF` wie jede andere Transformation ausgeschaltet.

7.5.3 Besonderheiten für verkettete Transformationen

Werkzeugdaten

Ein Werkzeug wird immer der ersten Transformation einer Kette zugeordnet. Die nachfolgende Transformation verhält sich dann so, als ob die aktive Werkzeuglänge Null wäre. Es sind nur die über Maschinendaten eingestellten Basislängen eines Werkzeuges (_BASE_TOOL_) für die **erste** Transformation der Kette wirksam.

Beispiel

In Kapitel "Verkettete Transformationen" finden Sie Konfigurationsbeispiele für einzelne Transformationen und daraus zusammengefügte verkettete Transformationen.

Siehe auch

Verkettete Transformationen (Seite 550)

7.5.4 Persistente Transformation

Funktion

Eine persistente Transformation ist immer aktiv und wirkt relativ zu den anderen explizit angewählten Transformationen. Zusätzlich angewählte Transformationen werden als erste verkettete Transformation mit der persistenten Transformation verrechnet.

Relativ zur persistenten Transformation anzuwählende Transformationen wie z. B. `TRANSMIT` müssen über `TRACON` mit der persistenten Transformation verkettet parametriert werden. Im Teileprogramm wird nicht die `TRACON` Transformation, sondern die erste verkettete Transformation programmiert.

Weitere Hinweise zur Programmierung siehe

Literatur:

/PGA/ Programmierhandbuch Arbeitsvorbereitung; Transformationen "verkettete Transformation"

An- und Abwahl

Die persistente Transformation wird angewählt über folgende Maschinendaten:

MD20144 $MC_TRAFO_MODE_MASK, Bit 0 = 1

MD20144 $MC_TRAFO_RESET_VALUE definiert persistente Transformation.

MD20140 $MC_TRAFO_RESET_VALUE=Nummer des Transformationsdatensatzes der persistenten Transformation

Ferner müssen gesetzt werden (d. h. beachtet wird):

MD20110 $MC_RESET_MODE_MASK

Bit 0 = 1 (Bit 7 wird ausgewertet)

Bit 7 =0 (MD20140 $MC_TRAFO_RESET_VALUE bestimmt den Transformationsdatensatz)

MD20112 $MC_START_MODE_MASK (MD20140 $MC_TRAFO_RESET_VALUE)

MD20118 $MC_GEOAX_CHANGE_RESET= TRUE (d. h. Geometrieachsen werden zurückgesetzt).

Werden diese zusätzlichen Einstellungen nicht passend gesetzt, wird der

Alarm 14404 ausgegeben.

Mit `TRAFOOF` wird die aktive `TRACON` abgewählt und automatisch die persistente Transformation angewählt.

Auswirkungen auf die Bedienung HMI

Da bei der persistenten Transformation immer eine Transformation aktiv ist, wird die HMI Bedienoberfläche gegenüber den bisherigen Verhalten hinsichtlich der An- und Abwahl von Transformationen angepasst:

`TRACON` auf HMI

Auf der Bedienoberfläche HMI wird entsprechend dazu **nicht**`TRACON`, sondern die 1. verkettete Transformation von `TRACON` z. B. `TRANSMIT` angezeigt. Entsprechend dazu wird der Transformationstyp der 1. verketteten Transformation durch die entsprechende Systemvariable $P_TRAFO sowie $AC_TRAFO geliefert. Zyklen, die in `TRANSMIT` geschrieben sind, können dann direkt verwendet werden.

`TRAFOOF` auf HMI

Auf der Bedienoberfläche HMI wird entsprechend zur Programmieranweisung `TRAFOOF` **keine** Transformation in der G-Code Liste angezeigt. Die Systemvariable $P_TRAFO sowie $AC_TRAFO liefern entsprechend dazu den Wert 0. Es wirkt die persistente Transformation und die Koordinatensysteme BKS und MKS unterscheiden sich. Die angezeigte MKS-Position bezieht sich immer auf die tatsächlichen Maschinenachsen.

Systemvariable

Neue Systemvariable liefern die Transformationstypen der aktiv verketteten Transformationen.

Beschreibung	NCK-Variable
keine Transformation aktiv: 0 eine Transformation aktiv: Typ der 1. verketteten Transformation bei TRACON bzw. Typ der aktiven Transformation, wen kein TRACON	$P_TRAFO_CHAIN[0]
keine Transformation aktiv: 0 eine Transformation aktiv: Typ der 2. verketteten Transformation bei TRACON	$P_TRAFO_CHAIN[1] $AC_TRAFO_CHAIN[1]
werden erstgenutzt, wenn mehr als 2 Transformationen verkettet sind. Diese Variablen liefern zur Zeit immer 0	$P_TRAFO_CHAIN[2] $AC_TRAFO_CHAIN[2] und $P_TRAFO_CHAIN[3] $AC_TRAFO_CHAIN[3]

Persistente Transformation auch anzeigen:

$P_TRAFO_CHAIN[0], $AC_TRAFO_CHAI[0]

Im Teileprogramm oder Zyklen kann hiermit eine aktive Transformation zuverlässig angezeigt werden.

Unterscheidung zwischen einer TRACON und den anderen Transformationen:

$P_TRAFO, $AC_TRAFO wenn keine Transformation aktiv ist oder man fragt $P_TRAFO_CHAIN[1], $AC_TRAFO_CHAI[1] auf ungleich Null ab.

Frames

Frameanpassungen bei An- und Abwahl der TRACON werden so durchgeführt, als gäbe es nur die 1. verkettete Transformation. Transformationen der virtuellen Achse werden bei TRAANG-Anwahl nicht beibehalten.

JOG

Beim Verfahren mit JOG bleibt die persisistene Transformation wirksam.

Randbedingungen

Die persistente Transformation ändert nicht die prinzipiellen Abläufe im NCK. Alle Einschränkungen, die bei aktiver Transformation vorhanden sind, gelten weiterhin.

Bei `RESET` wird eine vorhandene Transformation weiterhin komplett abgewählt und dafür die persistente Transformation neu angewählt. Im Fehlerzustand wird die Wiederanwahl nicht durchgeführt. Ein entsprechender Alarm weist dann auf die Fehlerkonstellation hin.

Bei `TRAANG` als persistenter Transformation kann der Alarm 14401 oder 14404 auftreten. Mit aktiver persistenter Transformation sind hiervon abhängig, im Fehlerfall auch weitere Trafo-Alarme möglich.

Beim Referieren wird die Transformation implizit abgewählt und erfordert anschließend ein `RESET` oder `START`, um die persistente Transformation wieder anzuwählen.

Beispiel

Bei einer Drehmaschine mit einer schrägen zusätzlichen Y-Achse soll die Transformation der schrägen Achse Bestandteil der Maschinenprojektierung sein und deshalb vom Programmierer nicht weiter beachtet werden. Mit `TRACYL` oder `TRANSMIT` werden Transformationen angewählt, die dann das `TRAANG` beinhalten müssen. Beim Ausschalten der programmierten Transformationen wird automatisch wieder `TRAANG` aktiv, In der `HMI` Bedienoberfläche wird entsprechend `TRACYL` oder `TRANSMIT` angezeigt.

Maschinendaten für eine Drehmaschine mit schräger Y1-Achse, welche zu X1 aber senkrecht zu Z1 steht.

CANDATA (1)

; Kinematik ohne Transformationen

MD20080 $MC_AXCONF_CHANAX_NAME_TAB[1] = "Y2"

MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1

MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0

MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3

; Daten für TRAANG

MD24100 $MC_TRAFO_TYP_1 = 1024; TRAANG Y1-Achse schräg zu X1, senkrecht zu Z1

MD24110 $MC_TRAFO_AXES_IN_1[0] = 2

MD24110 $MC_TRAFO_AXES_IN_1[1] = 1

MD24110 $MC_TRAFO_AXES_IN_1[2] = 3

MD24110 $MC_TRAFO_AXES_IN_1[3] = 0

```
MD24110 $MC_TRAFO_AXES_IN_1[4] = 0
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[0] =1
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[1] =2
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[2] =3
MD24700 $MC_TRAANG_ANGLE_1 = 60
MD24720 $MC_TRAANG_PARALLEL_VELO_RES_1 = 0,2
MD24721 $MC_TRAANG_PARALLEL_ACCEL_RES_1 = 0,2
 ; Definition der persistenten Transformation
MD20144 $MC_TRAFO_MODE_MASK = 1
MD20140 $MC_TRAFO_RESET_VALVUE= 1
MD20110 $MC_RESET_MODE_MASK = 'H01'
MD20112 $MC_START_MODE_MASK = 'H80'
MD20140 $MC_TRAFO_RESET_VALUE
MD20118 $MC_GEOAX_CHANGE_RESET= TRUE
 ; Daten für TRANSMIT, TRACYL
MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1 = 1 ; auch 2, bewirkt Alarm 21617
MD24200 $MC_TRAFO_TYP_2 = 257
MD24210 $MC_TRAFO_AXES_IN_2[0] = 1
MD24210 $MC_TRAFO_AXES_IN_2[1] = 4
MD24210 $MC_TRAFO_AXES_IN_2[2] = 3
MD24210 $MC_TRAFO_AXES_IN_2[3] = 0
MD24210 $MC_TRAFO_AXES_IN_2[4] = 0
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[0] =1
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[1] =4
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[2] =3
MD24300 $MC_TRAFO_TYP_3 = 514
MD24310 $MC_TRAFO_AXES_IN_3[0] = 1
MD24310 $MC_TRAFO_AXES_IN_3[1] = 4
MD24310 $MC_TRAFO_AXES_IN_3[2] = 3
MD24310 $MC_TRAFO_AXES_IN_3[3] = 0
MD24310 $MC_TRAFO_AXES_IN_3[4] = 0
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[0] =1
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[1] =4
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[2] =3
 ; Daten für TRACON
 ; TRACON Verkettung TRANSMIT 514/TRAANG (Y1-Achse zu X1)
```

MD24400 $MC_TRAFO_TYP_4 = 8192

MD24995 $MC_TRACON_CHAIN_1[0] = 3

MD24995 $MC_TRACON_CHAIN_1[1] = 1

MD24995 $MC_TRACON_CHAIN_1[2] = 0

MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[0] =1

MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[1] =4

MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[2] =3

;TRACON Verkettung TRANSMIT 257/TRAANG(Y1-Achse schräg zu X1)

MD24430 $MC_TRAFO_TYP_5 = 8192

MD24996 $MC-TRACON_CHAIN_2[0] = 2

MD24996 $MC-TRACON_CHAIN_2[1] = 1

MD24996 $MC_TRACON_CHAIN_2[2] = 0

MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[0] =1

MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[1] =4

MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[2] =3

M17

; passendes **Teileprogramm** dazu:

$TC_DP1[1,1]=120 ; Werkzeugtyp

$TC_DP2[1,1]=0

$TC_DP3[1,1]=3 ; Längenkorrekturvektor

$TC_DP4[1,1]=25

$TC_DP5[1,1]=5

$TC_DP6[1,1]=2 ; Radius; Werkzeugradius

;Transformationswechsel:

N1000 G0 X0 Y=0 Z0 A80 G603 SOFT G64

N1010 N1020 X10 Y20 Z30 ; TRAANG(,1) nicht nötig, da automatisch angewählt

N1110 TRANSMIT(1) N1120 X10 Y20 Z30N1130 Y2=0 ; TRACON(2) nicht nötig, da automatisch umgesetzt

N1210 TRAFOOF ; TRAANG(,1) nicht nötig, da automatisch umgesetzt

N1220 X10 Y20 Z30

M30

7.5.5 Achspositionen in der Transformationskette

Funktion

Für Maschinen mit System- oder OEM-Transformationen, insbesondere für verkettete Transformationen (TRACON) werden Systemvariablen mit folgenden Inhalten zur Verfügung gestellt:

Typ	Systemvariable	Bedeutung
REAL	$AA_ITR[ax,n]	Aktueller Sollwert am Ausgang der n-ten Transformation
REAL	$AA_IBC[ax]	Aktueller Sollwert einer kartesischen Achse
REAL	$VA_ITR[ax,n]	Aktueller Istwert am Ausgang der n-ten Transformation
REAL	$VA_IBC[ax]	Aktuelle kartesische BKS-Encoder-Position einer Achse
REAL	$VA_IW[ax]	Aktueller WKS-Istwert einer Achse
REAL	$VA_IB[ax]	Aktuelle BKS-Encoder-Position einer Achse

In Bezug auf das Steuerungsverhalten gibt es Folgendes zu beachten:

- POWER ON

  Bei nicht referierten Achsen hat die Encoder-Position den Wert 0. Die Encoder-Istwerte werden bei den $VA-Variablen entsprechend rücktransformiert.

- RESET

  Bei RESET kann sich eine aktive Transformation ändern, was unmittelbar Einfluss auf die Werte der Systemvariablen hat. Eine aktive Transformation, die nach RESET auch wieder aktiv ist, wird bei RESET kurzzeitig ausgeschaltet und wieder eingeschaltet. Dies hat unmittelbaren Einfluß auf die Positionsvariablen. Die Werte der Variablen können springen.

  Über die Variable:

  $AC_STAT == 0

  kann dieser Zustand in Synchronaktionen abgefragt werden.

$AA_ITR[<Achse>, <Trafo-Layer>]

Die Variable $AA_ITR[ax,n] ermittelt die Sollposition einer Achse am Ausgang der n-ten verketteten Transformation.

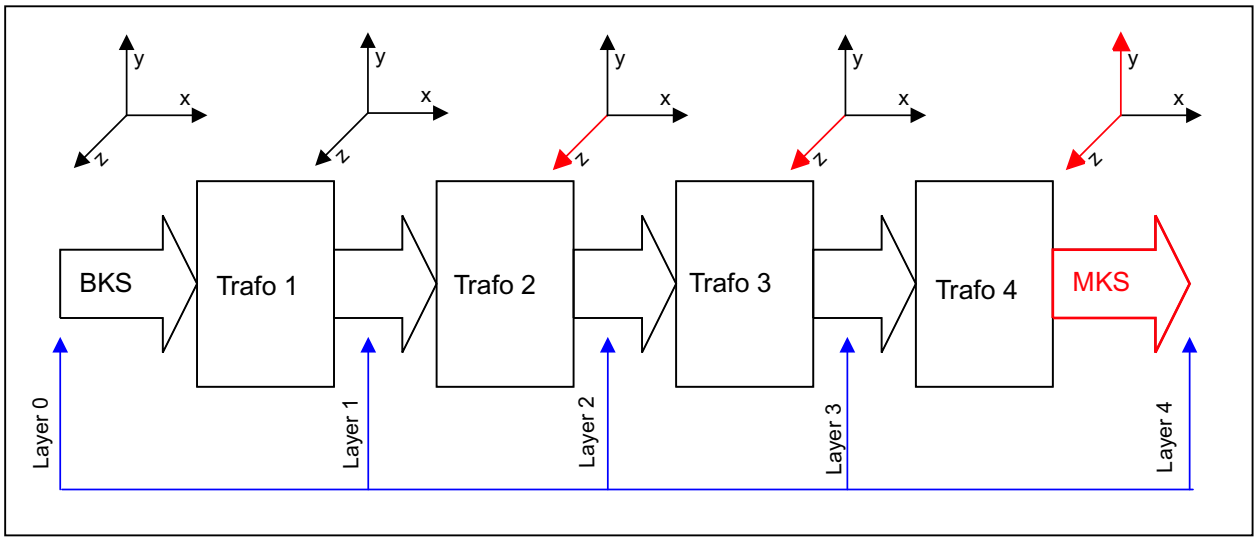


Bild 7-22 Trafo-Layer

Trafo-Layer

Der 2. Index der Variable entspricht dem Trafo-Layer, bei der die Positionen abgegriffen werden:

- Trafo-Layer 0: Die Positionen entsprechen den BKS-Positionen, d. h.:

  $AA_ITR[x,0] == $AA_IB[x]
- Trafo-Layer 1: Sollpositionen am Ausgang der 1. Transformation
- Trafo-Layer 2: Sollpositionen am Ausgang der 2. Transformation
- Trafo-Layer 3: Sollpositionen am Ausgang der 3. Transformation
- Trafo-Layer 4: Sollpositionen am Ausgang der 4. Transformation, d. h.:

  $AA_ITR[x,4] == $AA_IM[x]

Sollte eine oder mehrere Transformationen der Trafokette fehlen, so liefern die höchsten Ebenen die gleichen Werte. Wenn z. B. Trafo 3 und Trafo 4 fehlen, so entspricht: $AA_ITR[x,2] = $AA_ITR[x,3] = $AA_ITR[x,4] = $AA_IM[x]

Beim Ausschalten von Transformationen über `TRAFOOF` oder im RESET verschmelzen die Layer 0 bis 4 und die Variable liefert in diesem Fall immer den BKS-Wert (Layer 0).

Achse

Als 1. Index der Variable ist entweder ein Geometrie-, ein Kanal-, oder ein Maschinen-Achsbezeichner zulässig. Bei Programmierung von Geometrieachs-Bezeichner wirkt in jedem Trafo-Layer die Zuordnung der Kanalachsen zu den Geometrieachsen entsprechend der 0. Ebene. D. h. die BKS-seitige Geometrieachszuordnung ist in allen Ebenen wirksam. Geometrie-Achsbezeichner zu verwenden macht nur dann Sinn, wenn die Geometrieachsen nicht umgeschalten werden. Ansonsten sollten besser Kanalachsbezeichner verwendet werden.

$AA_IBC[<Achse>]

Die Variable $AA_IBC[ax] ermittelt die zwischen BKS und MKS liegende Sollpositon einer kartesischen Achse. Wenn eine Achse am Ausgang der n-ten Transformation kartesisch ist, so wird dieser Ausgangswert geliefert. Ist die entsprechende Achse am Ausgang aller Transformationen nicht kartesisch, so wird der BKS-Wert inklusive aller BKS-Korrekturen der Achse ermittelt.

Verhält sich TRACON kartesisch zu einer Achse, so wird deren MKS-Wert geliefert. Der verwendete Achsbezeichner kann ein Geometrie-, Kanal- oder ein Maschinen-Achsbezeichner sein.

$VA_ITR[<Achse>, <Trafo-Layer>]

Die Variable $VA_ITR[ax,n] ermittelt die Encoder-Position einer Achse am Ausgang der n-ten verketteten Transformation.

$VA_IBC[<Achse>]

Die Variable $VA_IBC[ax] ermittelt die zwischen BKS und MKS liegende Encoder-Positon einer kartesischen Achse. Der verwendete Achsbezeichner kann ein Geometrie-, Kanal- oder ein Maschinen-Achsbezeichner sein.

Ist eine Achse am Ausgang der n-ten Transformation kartesisch, so wird dieser Ausgangswert geliefert. Ist die entsprechende Achse am Ausgang aller Transformationen nicht kartesisch, so wird der BKS-Wert der Achse ermittelt.

$VA_IW[<Achse>]

Die Variable $VA_IW[ax] ermittelt die ins WKS rücktransformierte Encoder-Position einer Achse. Der WKS-Wert enthält alle achsialen Überlagerungsanteile (DRF, AA_OFF, ext. Nullpunktverschiebung, etc.) und Korrekturwerte (CEC, etc.).

$VA_IB[<Achse>]

Die Variable $VA_IB[ax] ermittelt die ins BKS rücktransformierte Encoder-Position einer Achse. Der BKS-Wert enthält alle achsialen Überlagerungsanteile (DRF, AA_OFF, ext. Nullpunktverschiebung, etc.) und Korrekturwerte (CEC, etc.).

Hinweis

VA_ITR, VA_IBC, $VA_IW, $VA_IB

Beim Lesen der Variable innerhalb eines IPO-Taktes ändert sich der Wert der Variablen nicht, obwohl sich der Istwert verändert haben könnte.

Bei aktiven Transformationen ist zu berücksichtigen, dass die Transformation der Istwerte ins BKS im IPO-Takt sehr zeitaufwändig sein kann. In diesem Fall ist ein ausreichender IPO-Takt einzustellen.

7.6 Kartesisches PTP-Fahren

Funktion

Mit dieser Funktion ist es möglich, eine kartesische Position mit einer Synchronachsbewegung anzufahren.

Dies ist immer dann sinnvoll, wenn beispielsweise die Gelenkstellung gewechselt wird und dabei die Bewegung durch eine Singularität führen würde.

Beim Durchfahren der Singularität würde es dabei entweder zu einer Herabsetzung der Vorschubgeschwindigkeit oder zu einer Achsüberlastung kommen.

Hinweis

Im MD24100 $MC_TRAFO_TYPE_1 muss der in TE4 beschriebene Transformationstyp eingetragen werden.

Die Funktion ist nur in Verbindung mit einer aktiven Transformation sinnvoll. Weiterhin ist die Funktion "Kartesisches PTP-Fahren" nur mit den Befehlen G0 und G1 zulässig. Anderenfalls wird der Alarm 14144 "PTP-Fahren nicht möglich" ausgegeben.

Bei aktivem PTP können Achsen der Transformation, die z. B. über POS verfahren werden, nicht gleichzeitig Positionierachsen sein. Der Alarm 17610 verhindert diesen unerlaubten Fall.

Aktivierung

Die Aktivierung der Funktion erfolgt mit Programmierung des Befehls PTP.

Mit dem Befehl CP wird die Funktion wieder ausgeschaltet. Die beiden Befehle sind in der G-Gruppe 49 enthalten.

- Befehl PTP: Der programmierte kartesische Punkt wird mit einer Synchronachsbewegung angefahren (PTP=point to point)
- Befehl CP: Der programmierte kartesische Punkt wird mit einer Bahnbewegung angefahren (Standardeinstellung), (CP=continuous path)
- Befehl PTPG0: Die programmierte kartesische PTP-Bewegung erfolgt bei jedem G0-Satz automatisch. Danach wird wieder auf CP gesetzt.

Power On

Nach `Power On` wird standardmäßig die Verfahrart CP für das Verfahren mit Transformation eingestellt. Über das MD20152 $MC_GCODE_RESET_VALUES[48] lässt sich die Voreinstellung auf kartesische PTP-Fahren umschalten.

Reset

Mit MD20152 $MC_GCODE_RESET_MODE[48] (Gruppe 49) wird festgelegt, welche Einstellung nach Reset/Teileprogrammende vorgenommen wird.

- MD=0: Einstellung wird entsprechend Maschinendatum

  MD20150 $MC_GCODE_RESET_VALUES[48] vorgenommen
- MD=1: Aktuelle Einstellung bleibt erhalten

Anwahl

Bei der Einstellung MD20152 $MC_GCODE_RESET_MODE[48] =0, kann mit MD20150 $MC_GCODE_RESET_VALUES[48] folgendes aktiviert werden:

- MD=2:

  Kartesisches PTP-Fahren wie bisher oder
- MD=3:

  PTPG0, nur G0-Sätze mit PTP automatisch abfahren und danach wieder auf CP umschalten

Randbedingungen

Bezüglich der Werkzeugbewegung und Kollision ist folgendes zu beachten:

- Da sich bei PTP deutlich andere Werkzeugbewegungen als bei CP ergeben können, müssen vor allem bei PTPG0 bestehende Unterprogramme, die unabhängig von der aktiven Transformation geschrieben wurden, nun auch die Kollisionsgefahren mit aktivem TRANSMIT berücksichtigen.
- Bei TRANSMIT und PTP verfahren Maschinenachsen grundsätzlich auf dem kürzesten Weg. Kleine Verschiebungen des Satzendpunktes können dazu führen, dass die Rundachse anstatt um + 179,99° um -179,99° gedreht wird, obwohl sich der Satzendpunkt fast nicht geändert hat.

Folgende Kombinationen mit weiteren NC-Funktionen werden nicht erlaubt:

- Mit PTP darf keine Werkzeugradiuskorrektur (WRK) aktiv sein.

  Grundsätzlich schließen sich G0 und G41 nicht gegeneinander aus. Ein aktives PTP erzeugt aber andere Konturen, als von der WRK eingerechnet wird und liefert eine Alarmmeldung der WRK.
- Mit PTPG0 wird bei aktiver Werkzeugradiuskorrektur per CP gefahren.

  Da sich G0 und G41 nicht gegeneinander ausschließen, wird bei aktiver Werkzeugradiuskorrektur automatisch auf CP umgeschaltet. Damit arbeitet die Radiuskorrektur auf aussagefähigen Konturen.
- Mit PTP ist Weiches An- und Abfahren (WAB) nicht möglich.

  WAB benötigt eine Kontur, um die An- bzw. Abfahrbewegung zu konstruieren. Mit PTP gibt es diese Information nicht.
- Mit PTPG0 wird bei Weichem An- und Abfahren (WAB) per CP gefahren

  WAB benötigt eine Kontur, um die An- bzw. Abfahrbewegung zu konstruieren und tangentiell aufsetzen bzw. abheben können. Die dafür erforderlichen Sätze werden

deshalb mit CP gefahren. Die G0-Sätze bis zur eigentlichen Anfahrkontur erfolgen aber mit PTP und damit schnell. Entsprechendes gilt auch für das Abfahren.

- Mit PTP sind Abspanzyklen wie CONTPRON, CONTDCON nicht möglich

  Die Abspanzyklen benötigen eine Kontur, um die Schnittaufteilung zu konstruieren. Mit PTP gibt es diese Information nicht. Es wird ein Alarm 10931 "Fehlerhafte Abspankorrektur" ausgegeben.

- Mit PTPG0 wird in Abspanzyklen wie CONTPRON, CONTDCON per CP gefahren. Die Abspanzyklen benötigen eine Kontur, um die Schnittaufteilung zu konstruieren. Die dafür erforderlichen Sätze werden mit CP gefahren.
- Fase und Rundung werden ignoriert.
- Eine Achsüberlagerung in der Interpolation darf sich während des PTP-Abschnitts nicht ändern. Hiervon betroffen sind z. B. Schnellabheben LIFTFAST, Werkzeugfeinkorrektur, Mitschleppen TRAILON sowie Tangentiales Nachführen TANGON.

In den PTP-Sätzen wird

- Kompressor automatisch abgewählt, da er nicht mit PTP verträglich ist.
- bei G643 automatisch nach G642 umgeschaltet.
- Achsen der Transformation nicht gleichzeitig Positionierachsen sein.

Besonderheiten

Folgende Besonderheiten sind zum Bezugskoordinatensystem zu beachten:

- Überschleifen G642 wird immer im Maschinenkoordinatensystem interpretiert und nicht wie üblich im kartesischen Basiskoordinatensystem.
- G641bestimmt Überschleifen abhängig von dem fiktiven Bahnweg, der aus den Maschinenachskoordinaten berechnet wurde.
- Eine F-Wert-Vorgabe bezieht sich bei G1 auf den fiktiven Bahnweg, der aus den Maschinenachskoordinaten berechnet wurde.

Satzsuchlauf

TRANSMIT bei Satzsuchlauf können unterschiedliche Maschinenachspositionen bei gleicher kartesischer Position ergeben, wenn ein Programmabschnitt mit Satzsuchlauf abgefahren wird.

Alarme

Eine unerlaubte Aktion, die zu einem Konflikt führen kann, wird mit folgenden Alarmen abgelehnt:

Alarm 14144: Wenn in PTP eine WRK angewählt oder aktiviert wird. Ebenso bei PTP mit Weichen An- und Abfahren (WAB) oder PTP ohne der erforderlichen G0- und G1-Sätze.

Alarm 10753: Mit PTPG0 wird bei aktiver WRK intern auf CP geschaltet, damit die Werkzeugradiuskorrektur korrekt durchlaufen wird.

Alarm 10754: Im Konfliktfall dennoch möglich.

Alarm 10778: Im Konfliktfall dennoch möglich.

Alarm 10744: Mit PTPG0 wird bei aktiver WAB per CP gefahren, damit Weichen An- und Abfahren korrekt abgearbeitet wird.

Alarm 10746: Im Konfliktfall dennoch möglich.

Alarm 17610: Achse der Transformation dürfen nicht gleichzeitig Positionierachsen sein, die über POS verfahren werden.

Hinweis

Weitere Informationen zur Programmierung mit Beispielen entnehmen Sie bitte:

Literatur:

/PGA/ Programmierhandbuch Arbeitsvorbereitung; Kapitel Transformationen, "Kartesisches PTP-Fahren"

7.6.1 Programmierung der Stellung

Eine Maschinenstellung ist im Allgemeinen allein durch die Positionsangabe mit kartesischen Koordinaten und die Orientierung des Werkzeugs nicht eindeutig bestimmt. Je nachdem, um welche Kinematik es sich handelt, existieren bis zu 8 unterschiedliche Gelenkstellungen. Diese unterschiedlichen Gelenkstellungen sind transformationsspezifisch.

Adresse STAT

Eine kartesische Position muss eindeutig in die Achswinkel umgerechnet werden können. Deshalb muss die Stellung der Gelenke unter der Adresse `STAT` eingegeben werden.

Die Adresse `STAT` enthält als Binärwert für jede der möglichen Stellungen ein Bit. Die Bedeutung der Bits wird von der jeweiligen Transformation festgelegt.

Für die Transformationen, welche in der Druckschrift "Transformationspaket Handling (TE4)", enthalten sind, sind die Bits wie in obigem Bild den unterschiedlichen Stellungen zugeordnet.

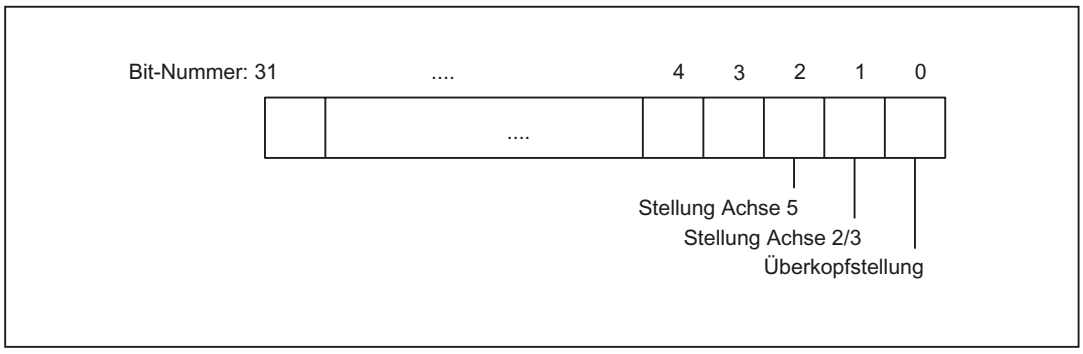


Bild 7-23 Stellungsbits für Transformationspaket Handling

Hinweis

Die Programmierung der Adresse `STAT` ist nur beim "Kartesischen PTP-Fahren" sinnvoll, da beim Verfahren mit aktiver Transformation ein Stellungswechsel normalerweise nicht möglich ist. Wird mit dem Befehl CP verfahren, wird die Stellung für den Zielpunkt vom Startpunkt übernommen.

7.6.2 Überlappbereiche der Achswinkel

Adresse TU

Um auch Achswinkel eindeutig anfahren zu können, welche größer ±180° sind, muss diese Information unter der Adresse TU (Turn) programmiert werden. Die Adresse TU stellt damit das Vorzeichen der Achswinkel dar. Damit kann ein Achswinkel von $|\theta| < 360°$ eindeutig angefahren werden.

Die Variable TU enthält für jede Achse, die in die Transformation eingeht, ein Bit, das die Verfahrrichtung anzeigt.

- TU-Bit=0: $0° \leq \theta < 360°$
- TU-Bit=1: $360° < \theta < 0°$

Bei Linearachsen ist das TU-Bit auf 0 gesetzt.

Bei Achsen mit einem Verfahrbereich >±360°wird immer auf kürzestem Weg verfahren, da die Achsstellung durch die TU-Information nicht eindeutig bestimmbar ist.

Wird bei einer Position kein TU programmiert, wird immer auf kürzestem Weg verfahren.

7.6.3 Beispiele für Mehrdeutigkeiten in der Stellung

Als Beispiel für die Mehrdeutigkeit durch die unterschiedlichen Gelenkstellungen soll eine 6-Achs Gelenkkinematik dienen.

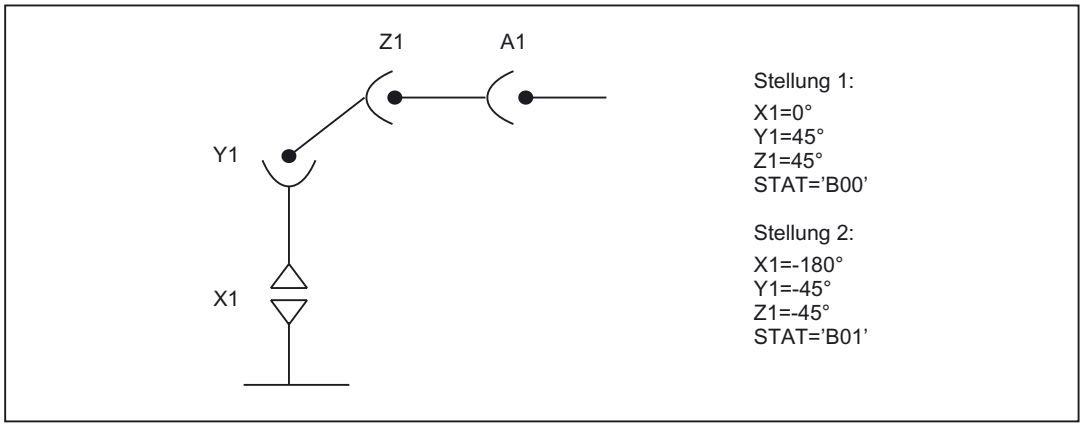


Bild 7-24 Mehrdeutigkeit Überkopfbereich

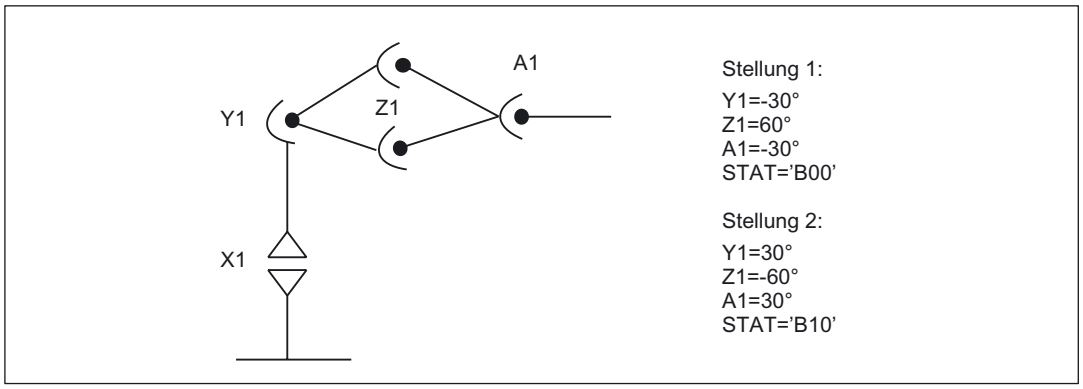


Bild 7-25 Mehrdeutigkeit Ellbogen oben oder unten

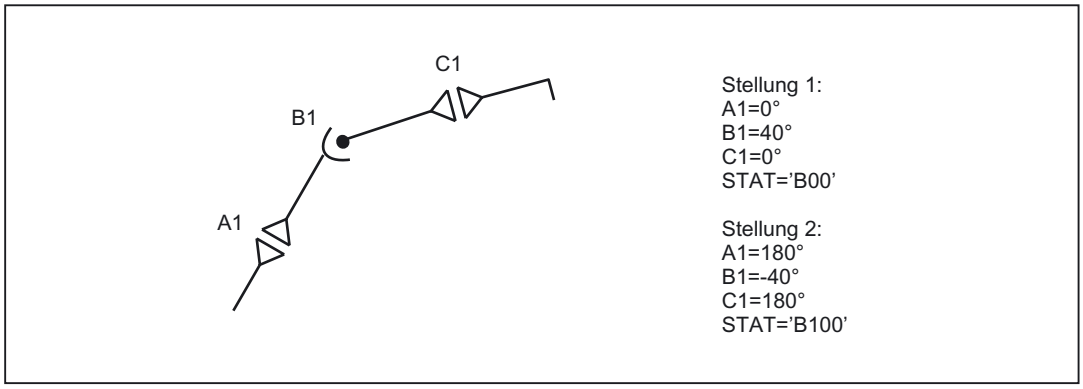


Bild 7-26 Mehrdeutigkeit Achse B1

7.6.4 Beispiel für Mehrdeutigkeit in der Rundachsposition

Die im folgenden Bild angegebene Rundachsposition kann in negativer oder in positiver Richtung angefahren werden. Unter der Adresse A1 wird die Richtung programmiert.

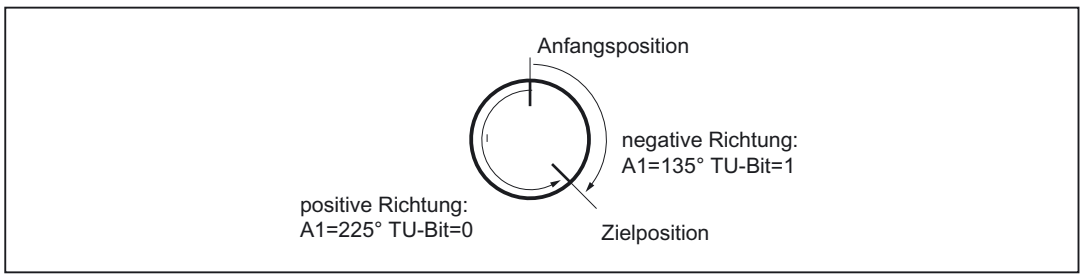


Bild 7-27 Mehrdeutigkeit in der Rundachsposition

7.6.5 PTP/CP-Umschaltung in der Betriebsart JOG

In der Betriebsart JOG kann über ein PLC-Steuersignal das Ein- und Ausschalten der Transformation durchgeführt werden. Dieses Steuersignal ist nur in der Betriebsart JOG wirksam und, wenn eine Transformation über das Programm aktiviert wurde.

Beim Zurückschalten in die Betriebsart AUTO wird der Zustand, welcher vor dem Umschalten aktiv war, wieder hergestellt.

Mit dem Signal "PTP-Fahren aktiv" DBX317.6 wird angezeigt welche Verfahrart aktiv ist. Mit dem Signal "PTP-Fahren aktivieren" DBX29.4 kann die Verfahrart umgeschaltet werden

Betriebsartenwechsel

Die Funktion "Kartesisches PTP-Fahren" ist nur in den Betriebsarten AUTO und MDA sinnvoll. Beim Wechsel der Betriebsart nach JOG wird mit der Einstellung CP verfahren. Beim Wechsel zurück nach AUTO oder MDA wird der in diesen Betriebsarten zuletzt eingestellte Modus wieder hergestellt.

REPOS

Während des Rückpositionierens wird die Einstellung für "Kartesisches PTP-Fahren" nicht geändert. War im Unterbrechungssatz PTP eingestellt, wird auch mit PTP rückpositioniert. Bei Schräger Achse "TRAANG" ist nur CP-Fahren in der Betriebsart `REPOS` aktiv.

7.7 Kartesisches Handverfahren (Option)

Hinweis

Für die Funktion "Kartesisches Handverfahren" ist die Option "Transformationspaket Handling" erforderlich.

Funktion

Die Funktion "Kartesisches Handverfahren" ermöglicht, als Bezugssystem für die Betriebsart JOG Achsen in folgenden kartesischen Koordinatensystemen unabhängig voneinander einzustellen:

- Basiskoordinatensystem (BKS)
- Werkstückkoordinatensystem (WKS)
- Werkzeugkoordinatensystem (TKS)

Einstellung und Aktivierung erfolgt über das Maschinendatum:

MD21106 $MC_CART_JOG_SYSTEM (Koordinatensysteme beim kartesischen JOG)

Bit	Bedeutung
0	Basiskoordinatensystem
1	Werkstückkoordinatensystem
2	Werkzeugkoordinatensystem

Hinweis

Das Werkstückkoordinatensystem ist gegenüber dem Basiskoordinatensystem über Frames verschoben und verdreht.

Literatur:
Funktionshandbuch Grundfunktion; Achsen, Koordinatensysteme, Frames (K2)

Darstellung der Bezugssysteme im Koordinatensystem:

	**WKS** = Werkstücknullpunkt		**TKS** = Werkzeugbezugspunkt

Bezugssysteme auswählen

Für die JOG-Bewegung kann eines der drei Bezugssysteme sowohl für die

Translation (Grobverschiebung) bei Geometrieachsen, als auch für die

Orientierung bei Orientierungsachsen über das

SD42650 $SC_CART_JOG_MODE getrennt vorgegeben werden.

Wenn für das Translations- oder für das Orientierunsbezugssystem mehr als ein Bit gesetzt ist, oder wenn versucht wird, ein nicht über das MD21106 $MC_CART_JOG_SYSTEM freigegebene Bezugssystem einzustellen, dann erfolgt der Alarm 14148 "Bezugsystem für Kartesisches Handverfahren unzulässig".

Translation

Mit einer translatorischen Bewegung kann die Werkzeugspitze (TCP) 3-dimensional parallel zu den Achsen des Bezugssystems bewegt werden. Die Verfahrbewegung erfolgt hierbei über die VDI-Signale der Geometrieachsen.

Über das Maschinendatum MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_x[n] werden die Geometrieachsen zugeordnet. Durch gleichzeitiges Verfahren in mehr als einer Richtung lassen sich auch Bewegungen, die diagonal zu den Richtungen des Bezugssystems liegen, ausführen.

Translation im BKS

Das Basiskoordinatensystem (BKS) beschreibt den kartesischen Nullpunkt der Maschine.

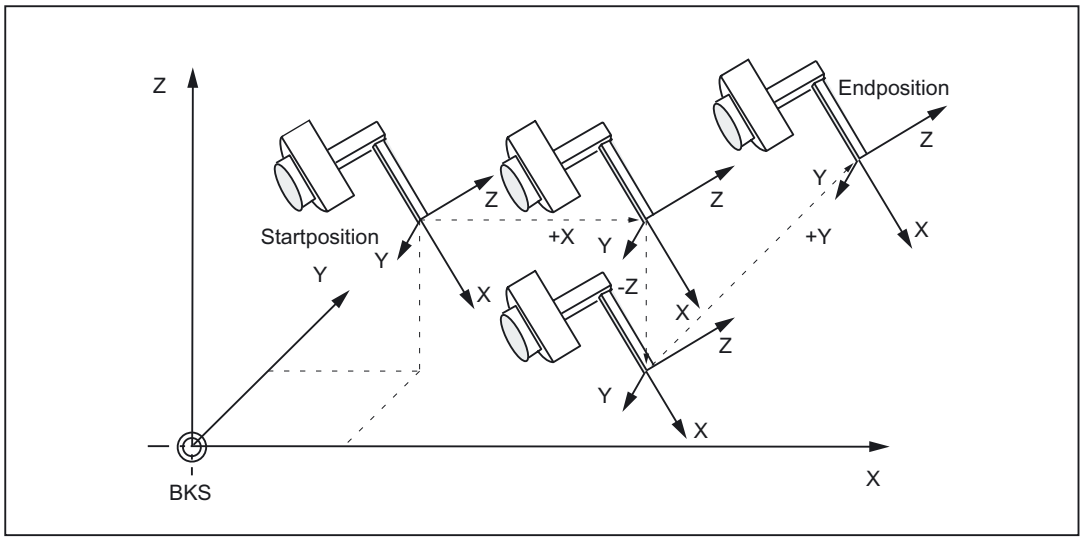


Bild 7-28 Kartesisches Handverfahren im Basiskoordinatensystem (Translation)

Translation im WKS

Das Werkstückkoordinatensystem (WKS) liegt im Nullpunkt des Werkstücks. Über Frames kann das Werkstückkordinatensystem bezüglich dem Basiskordinatensystem verschoben und verdreht. sein. Solange keine Framedrehung aktiv ist, entsprechen die Verfahrbewegungen für die Translation den Bewegungen im Basiskoordinatensystem.

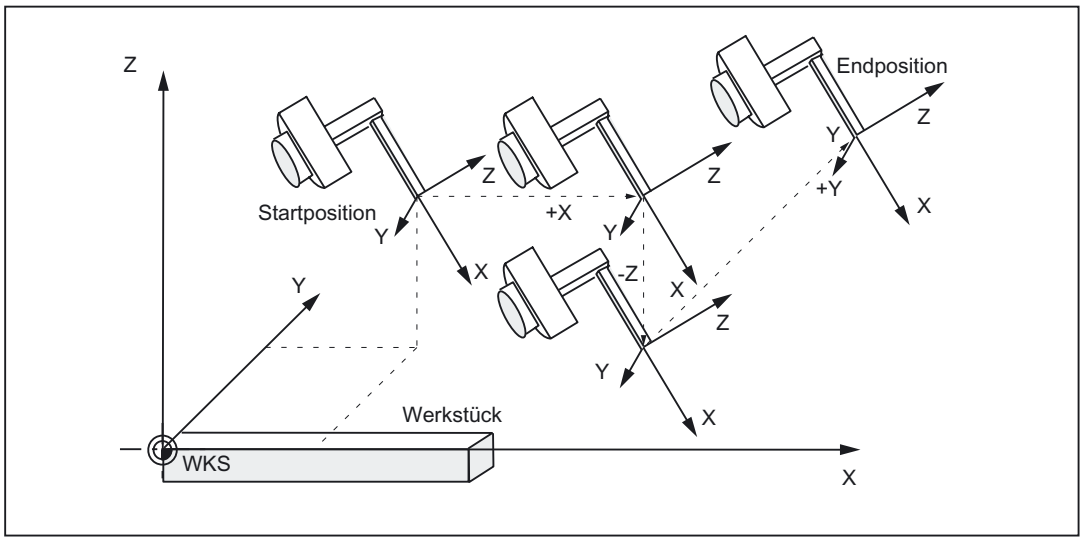


Bild 7-29 Kartesisches Handverfahren im Werkstückkoordinatensystem (Translation)

Translation im TKS

Das Werkzeugkoordinatensystem (TKS) liegt in der Werkzeugspitze. Seine Richtung ist von der aktuellen Stellung der Maschine abhängig, denn das Werkzeugkoordinatensystem bewegt sich während der Bewegung mit.

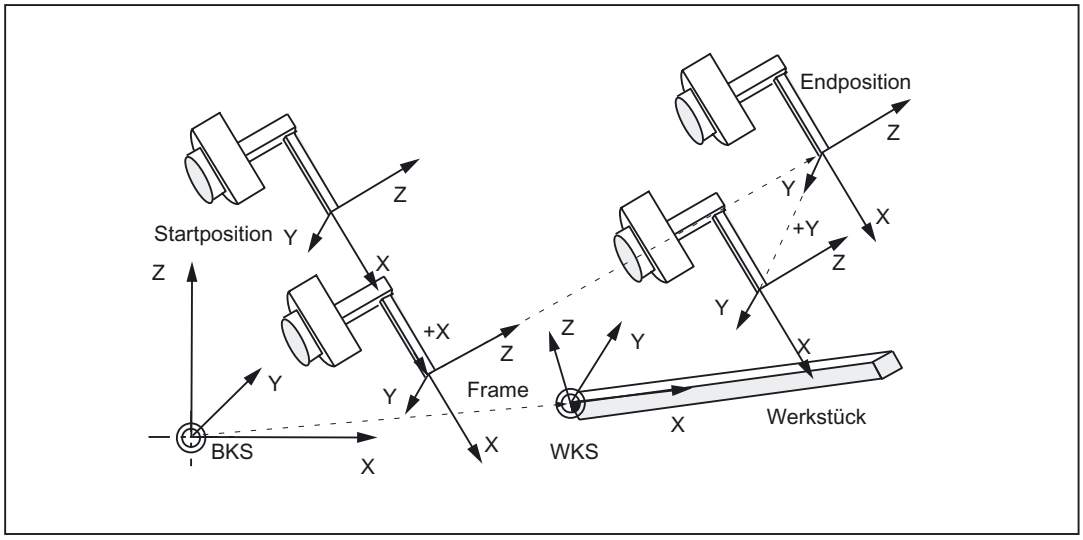


Bild 7-30 Kartesisches Handverfahren im Werkzeugkoordinatensystem (Translation)

Translation und Orientierung im TKS gleichzeitig

Werden gleichzeitig Translations- und Orientierungsbewegungen ausgeführt, wird die Translation immer zur aktuellen Orientierung des Werkzeugs verfahren. Damit lassen sich Zustellbewegungen, die direkt in Werkzeugrichtung, oder Bewegungen, die senkrecht zur Werkzeugrichtung verlaufen, ausführen.

Orientierung

Über eine Orientierungsbewegung kann das Werkzeug zur Bauteiloberfäche ausgerichtet werden. Die Orientierungsbewegung bekommt die Steuerung vom PLC über die VDI-Signale der Orientierungsachsen (DB21, ... DBB321).

Es können gleichzeitig mehrere Orientierungsachsen verfahren werden. Die virtuellen Orientierungsachsen führen hierbei Drehungen um die feststehenden Achsen des jeweiligen Bezugssystems aus.

Die **Drehungen** werden nach den RPY-Winkeln bezeichnet.

- A-Winkel : Drehung um Z-Achse
- B-Winkel : Drehung um Y-Achse
- C-Winkel : Drehung um X-Achse

Programmierung der Drehungen:

Wie die Drehungen ausgeführt werden sollen, kann der Anwender mit den aktuellen G-Codes der Gruppe 50 für die Orientierungsdefinition

`ORIEULER`, `ORIRPY`, `ORIVIRT1` und `ORIVIRT2` festlegen.

Bei ORIVIRT1 werden die Drehungen laut MD21120 $MC_ORIAX_TURN_TAB_1 ausgeführt. Die Zuordnung der Orientierungsachsen zu den Kanalachsen erfolgt über das Maschinendatum: MD24585 $MC_TRAFO5_ORIAX_ASSIGN_TAB_1.

Die **Drehrichtung** ergibt sich nach der "Rechte Hand Regel". Hierbei zeigt der Daumen in Richtung der Drehachse. Die Finger geben die positive Drehrichtung vor.

Orientierung im WKS

Die Drehungen erfolgen um die feststehenden Richtungen des Werkstückkoordinatensystems. Wenn keine Framedrehung aktiv ist, entsprechen die Bewegungen den Drehungen im Basiskoordinatensystem.

Orientierung im BKS

Die Drehungen erfolgen um die feststehenden Richtungen des Basiskoordinatensystems.

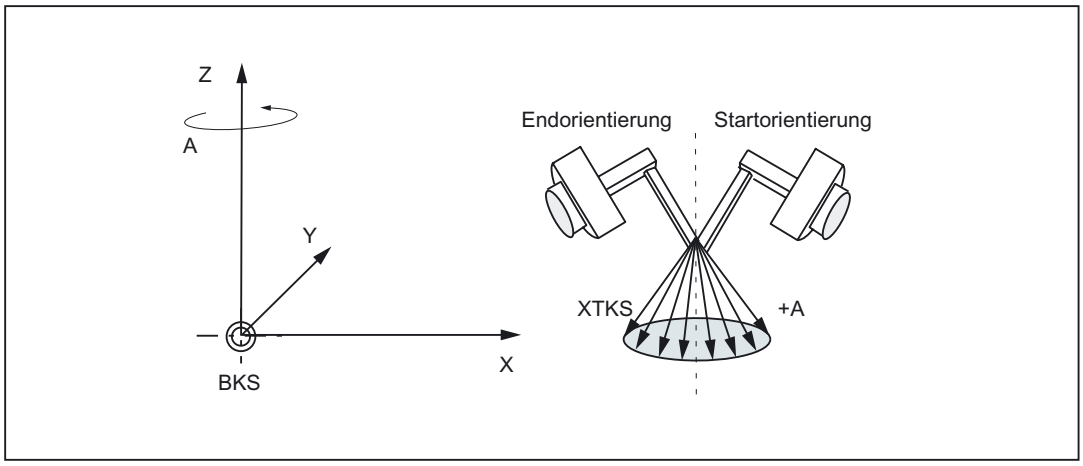


Bild 7-31 Kartesisches Handverfahren im Basiskoordinatensyst. Orientierungswinkel A

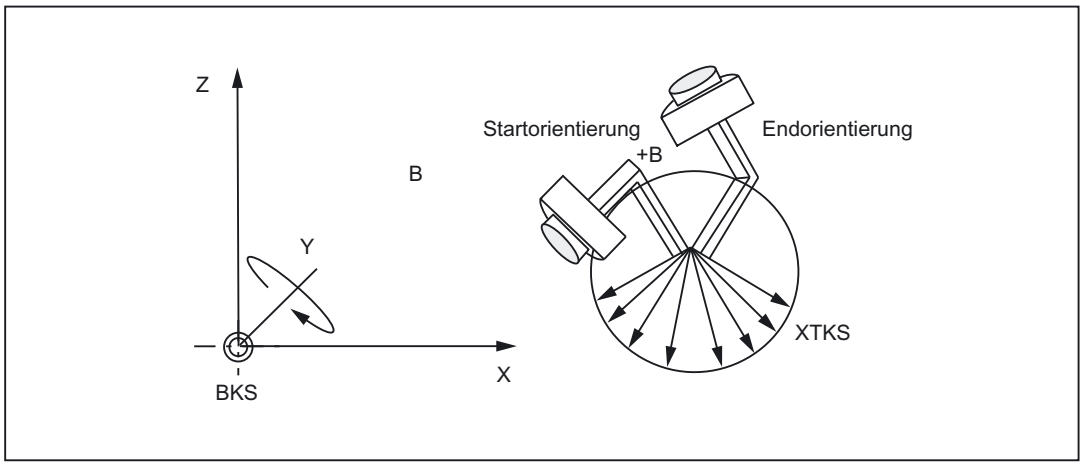


Bild 7-32 Kartesisches Handverfahren im Basiskoordinatensyst. Orientierungswinkel B

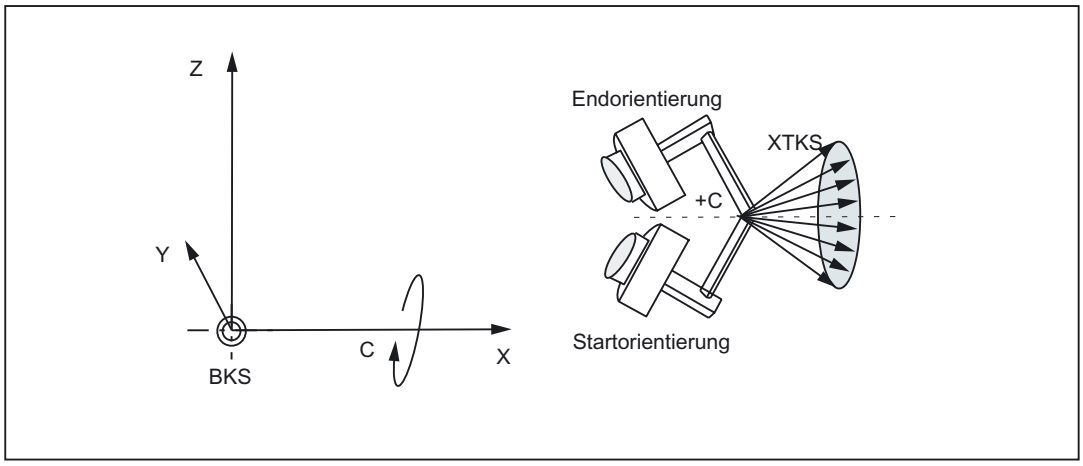


Bild 7-33 Kartesisches Handverfahren im Basiskoordinatensyst. Orientierungswinkel C

Orientierung im TKS

Die Drehungen erfolgen um die sich bewegenden Richtungen im Werkzeugkoordinatensystem. Die aktuellen Bezugsrichtungen des Werkzeugs werden hierbei immer als Drehachsen verwendet.

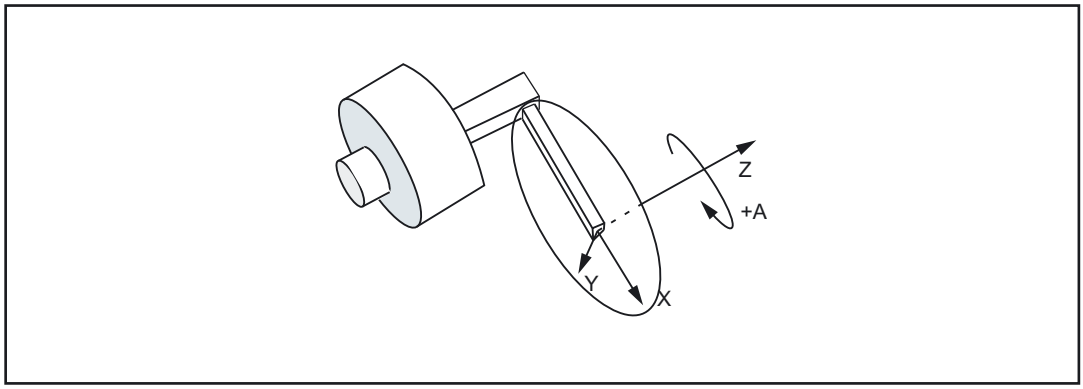


Bild 7-34 Kart. Handverfahren im Werkzeugkoordinatensystem Orientierungswinkel A

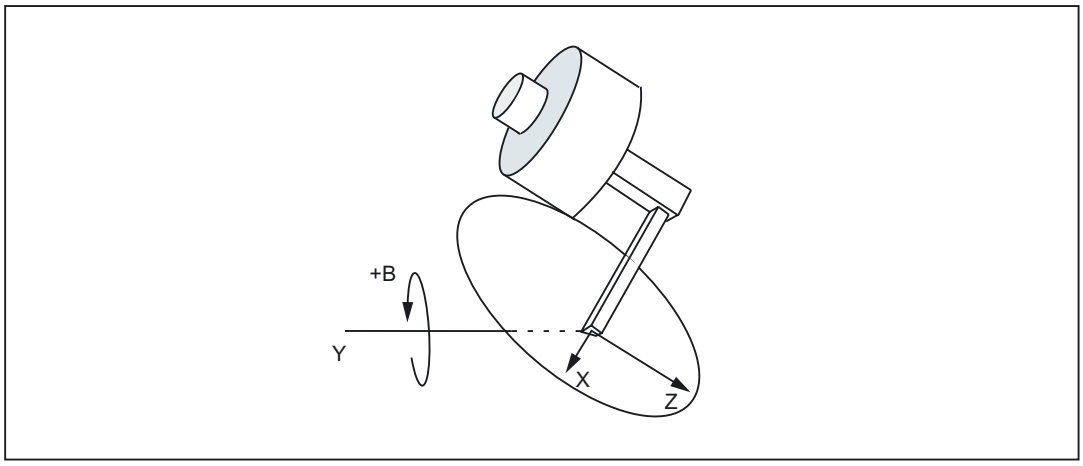


Bild 7-35 Kart. Handverfahren im Werkzeugkoordinatensystem Orientierungswinkel B

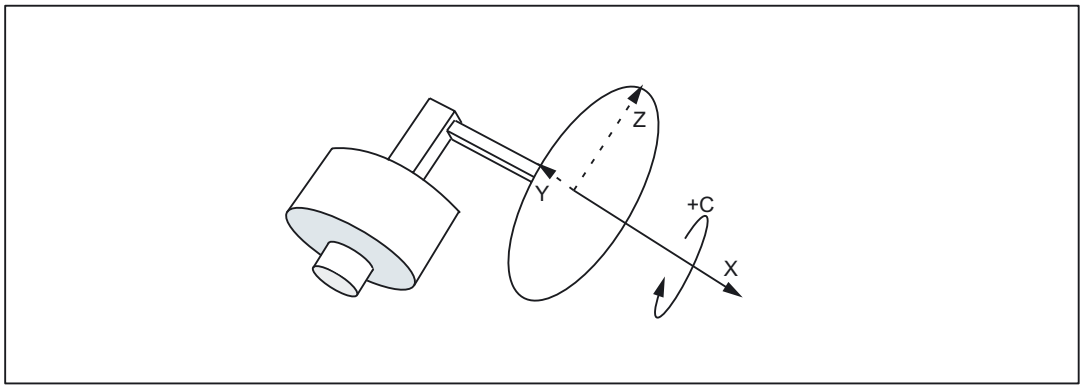


Bild 7-36 Kart. Handverfahren im Werkzeugkoordinatensystem Orientierungswinkel C

Randbedingungen

Nur wenn das NST DB31, ... DBX33.6 ("Transformation aktiv") auf 1 steht, ist es möglich die Funktion Kartesische Handverfahren auszuführen. Dabei gelten folgende Randbedingungen:

- Für SINUMERIK 840D ist die Option "Transformationspaket Handling" mit 5- oder 6-Achs-Transformation erforderlich.
- Virtuelle Orientierungsachsen müssen über folgendes Maschinendatum definiert sein:

  MD24585 $MC_TRAFO5_ORIAX_ASSIGN_TAB_1[n]
- Das NST DB31, ... DBX29.4 ("PTP/CP Fahren aktivieren") muss 0 sein.
- Das Maschinendatum MD21106 $MC_CART_JOG_SYSTEM muss > 0 sein.

Tabelle 7- 2 Bedingungen für Kartesisches Handverfahren

Transformation im Programm aktiv (TRAORI..)	G-Codes PTP/CP	NST "PTP/CP Fahren aktivieren"	NST "Transformation aktiv"
FALSE	nicht wirksam	nicht wirksam	DB31, ... DBX33.6 = 0
TRUE	CP	DB31, ... DBX29.4 = 0	DB31, ... DBX33.6 = 1
TRUE	CP	DB31, ... DBX29.4 = 1	DB31, ... DBX33.6 = 0
TRUE	PTP	DB31, ... DBX29.4 = 0	DB31, ... DBX33.6 = 1
TRUE	PTP	DB31, ... DBX29.4 = 1	DB31, ... DBX33.6 = 0

Der aktuell im Programm wirksame G-Code PTP/CP beeinflusst das Kartesische Handverfahren nicht. Die VDI-Nahtstellensignale werden im Kanal-DB für Geometrie- und Orientierungsachsen ausgewertet.

Aktivierung

Das Bezugsystem für das Kartesisches Handverfahren wird wie folgt einstellt:

- Die Funktion Kartesisches Handverfahren wird mit folgendem Maschinendatum aktiviert:

  MD21106 $MC_CART_JOG_SYSTEM > 0

  Die Freischaltung der Bezugssysteme BKS, WKS oder TKS erfolgt durch Setzen von Bits im MD21106 $MC_CART_JOG_SYSTEM.

- Die JOG-Verfahrbewegung über das SD42650 SC_CART_JOG_MODE

  Standard-Verhalten wie bisher: Bits 0 bis 2 = 0, Bits 8 bis 10 = 0.

  Bezugssystem für die Translation über die Bits 0 - 2 und das Bezugssystem für die Orientierung über die Bits 8 - 10.

  Nur wenn nicht alle Bits = 0 eingestellt ist, erfolgt ein Verfahren über die neue Funktionalität. Die Bezugssysteme für die Translation und die Orientierung können unabhängig voneinander eingestellt werden.

In nachfolgender Tabelle wird die Bedeutung der Bits erläutert.

Tabelle 7- 3 Bitzugsordnung für SD42650 $SC_CART_JOG_MODE (nur ein Bit darf gesetzt werden)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserviert					Translation im TKS	Translation im WKS	Translation im BKS
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Reserviert					Orientierung im TKS	Orientierung im WKS	Orientierung im BKS

Bezugssysteme kombinieren

In der nachfolgenden Tabelle sind alle Kombinationsmöglichkeiten der Bezugssysteme dargestellt.

Tabelle 7- 4 Kombinationsmöglichkeiten der Bezugssysteme

SD42650 $SC_CART_JOG_MODE						Bezugssystem für	
Bit 10	Bit 9	Bit 8	Bit 2	Bit 1	Bit 0	Orientierung	Translation
0	0	0	don't care	don't care	don't care	Standard	Standard
Standard	Standard	Standard	0	0	0	Standard	Standard
0	0	1	0	0	1	BKS	BKS
0	0	1	0	1	0	BKS	WKS
0	0	1	1	0	0	BKS	TKS
0	1	0	0	0	1	WKS	BKS
0	1	0	0	1	0	WKS	WKS
0	1	0	1	0	0	WKS	TKS
1	0	0	0	0	1	TKS	BKS
1	0	0	0	1	0	TKS	WKS
1	0	0	1	0	0	TKS	TKS

7.8 Transformations-MD über Teileprogramm/Softkey wirksam setzen

7.8.1 Funktionalität

Transformations-MD können über Programmbefehl/Softkey wirksam gesetzt werden, d. h. diese können z. B. vom Teileprogramm aus beschrieben und die Transformationskonfiguration deshalb vollständig verändert werden.

Es können in der Steuerung zehn verschiedene Transformationen eingestellt werden. Der Transformationstyp wird dabei über die folgenden Maschinendaten eingestellt:

MD24100 $MC_TRAFO_TYPE_1

bis

MD24460 $MC_TRAFO_TYPE_10.

Eigenschaften

Die Transformations-MD sind NEWCONFIG-wirksam.

Die Schutzstufe ist 7/7 (KEYSWITCH_0), so dass eine Änderung vom NC-Programm aus ohne besondere Rechte zulässig ist.

Ist beim Wirksamwerden des NEWCONFIG-Kommandos (gleichgültig ob über den NC-Programmbefehl `NEWCONF`, über HMI oder implizit bei Reset bzw. Programmende) keine Transformation angewählt (aktiviert), können die oben aufgeführten Maschinendaten ohne Einschränkung verändert und wirksam gemacht werden.

Insbesondere können neue Transformationen konfiguriert werden oder vorhandene Transformationen durch solche eines anderen Typs ersetzt bzw. gelöscht werden, da die Änderungsmöglichkeiten sich nicht auf die Neuparametrierung vorhandener Transformationen beschränken.

7.8.2 Randbedingungen

Maschinendaten ändern

Die Maschinendaten, die eine aktive Transformation betreffen, dürfen nicht geändert werden; ansonsten wird Alarm gegeben.

Dies sind in der Regel alle Maschinendaten, die einer Transformation über die zugehörige Transformationsdatengruppe zugeordnet sind. Maschinendaten, die zwar in der Gruppe einer aktiven Transformation enthalten sind, aber nicht verwendet werden, können verändert werden (obgleich das kaum eine sinnvolle Operation sein wird). So dürfte zum Beispiel bei einer aktiven Transformation mit MD24100 $MC_TRAFO_TYPE = 16 (5-Achstransformation mit drehbarem Werkzeug und zwei rechtwinklig angeordneten Rundachsen A und B) das Maschinendatum MD24564 $MC_TRAFO5_NUTATOR_AX_ANGLE_n verändert werden, da es in diese Transformation nicht eingeht.

Zusätzlich darf bei einer aktiven Orientierungstransformation das Maschinendatum MD21110 $MC_X_AXIS_IN_OLD_X_Z_PLANE nicht verändert werden.

Hinweis

Bei einer Programmunterbrechung (Repos, Restweglöschen, ASUPs usw.) werden zum Wiederanfahren steuerungsintern mitunter mehrere Sätze benötigt, die bereits abgefahren wurden. Das Verbot, Maschinendaten einer aktiven Transformation zu verändern, bezieht sich auch auf diese Sätze.

Beispiel:

Es sind zwei Orientierungstransformationen über Maschinendatum eingestellt, z. B. MD24100 $MC_TRAFO_TYPE_1 = 16, MD24200 $MC_TRAFO_TYPE_2 = 18.

Beim Ausführen des NEWCONFIG-Kommandos sei die zweite Transformation aktiv. Es dürfen dann alle Maschinendaten geändert werden, die nur die erste Transformation betreffen, z. B.:

MD24500 $MC_TRAFO5_PART_OFFSET_1

nicht aber z. B.:

MD24650 $MC_TRAFO5_BASE_TOOL_2

oder

MD21110 $MC_X_AXIS_IN_OLD_X_Z_PLANE

Weiterhin kann z. B. mit MD24300 $MC_TRAFO_TYPE_3 = 256 eine weitere Transformation (Transmit) eingestellt und mittels weiterer Maschinendaten auch parametriert werden.

Geometrieachsen definieren

Geometrieachsen müssen **vor** dem Hochlauf der Steuerung über folgende Maschinendaten definiert werden:

MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_X[n]

oder

MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[n]

Zuordnung ändern

Die Zuordnung eines Transformationsdatensatzes zu einer Transformation ergibt sich aus der Reihenfolge der Einträge in MD24100 $MC_TRAFO_TYPE_X. Dem ersten Eintrag in der Tabelle wird der erste Transformationsdatensatz zugeordnet, dem zweiten entsprechend der zweite. Diese Zuordnung darf (und kann) für eine aktive Transformation nicht verändert werden.

Beispiel:

Es sind drei Transformationen eingestellt: zwei Orientierungstransformationen und eine Transmittransformation, z. B.

MD24100 $MC_TRAFO_TYPE_1 = 16

; Orientierungstransformation, 1. Orientierungs-Trafo-Datensatz

MD24200 $MC_TRAFO_TYPE_2 = 256 : Transmit-Transformation

MD24300 $MC_TRAFO_TYPE_3 = 18

; Orientierungstransformation, 2. Orientierungs-Trafo-Datensatz

Der ersten Transformation (gleichzeitig erste Orientierungstransformation) wird der erste Datensatz für Orientierungstransformationen zugeordnet, der dritten Transformation (gleichzeitig zweite Orientierungstransformation) entsprechend der zweite Transformationsdatensatz.

Ist nun beim Ausführen des `NEWCONFIG`-Kommandos die dritte Transformation aktiv, so ist es nicht zulässig, die erste Transformation in eine Transformation einer anderen Gruppe (z. B. TRACYL) zu verwandeln, da in diesem Falle die dritte Transformation nicht mehr die zweite, sondern die erste Orientierungstransformation werden würde.

In diesem Beispiel ist es jedoch zulässig, für die erste Transformation eine andere Orientierungstransformation einzustellen (z. B. mit MD24100 $MC_TRAFO_TYPE_1 = 32) oder als erste Transformation die einer anderen Gruppe einzustellen (z. B. mit MD24100 $MC_TRAFO_TYPE_1 = 1024, TRAANG), wenn gleichzeitig die zweite Transformation zu einer Orientierungstransformation wird, z. B. mit MD24200 $MC_TRAFO_TYPE_2 = 48.

7.8.3 Steuerungsverhalten bei Power On, Betriebsartenwechsel, Reset, Satzsuchlauf, REPOS

Mit Hilfe folgender Maschinendaten ist es möglich, eine Transformation automatisch bei `RESET` (d. h. auch bei Programmende) und/oder bei Programmstart anzuwählen:

MD20110 $MC_RESET_MODE_MASK

MD20112 $MC_START_MODE_MASK

und

MD20140 $MC_TRAFO_RESET_VALUE

Das kann dazu führen, dass z. B. bei Programmstart oder bei Programmende ein Alarm auftritt, wenn das Maschinendatum einer aktiven Transformation verändert wurde.

Um dieses Problem beim Umkonfigurieren von Transformationen mit einem NC-Programm zu vermeiden, wird deshalb vorgeschlagen, ein NC-Programm wie folgt aufzubauen:

```
N10 TRAFOOF()                                ; Eine evtl. noch aktive Trafo abwählen
N20$MC_TRAFO5_BASE_TOOL_1[0]=0               ; Maschinendatum beschreiben
N30$MC_TRAFO5_BASE_TOOL_1[0]=3               ;
N40$MC_TRAFO5_BASE_TOOL_1[0]=200             ;
N130 NEWCONF                                 ; Neu beschriebene Maschinendaten
                                             ; übernehmen
N140 M30
```

7.8.4 Liste der betroffenen Maschinendaten

Die Maschinendaten, die NEWCONFIG-fähig gemacht werden können, sind im folgenden aufgelistet.

Alle Transformationen

Maschinendaten, die für alle Transformationen relevant sind:

- MD24100 $MC_TRAFO_TYPE_1 bis MD24480 $MC_TRAFO_TYPE_10
- MD24110 $MC_TRAFO_AXES_IN_1 bis MD24482 $MC_TRAFO_AXES_IN_10
- MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 bis
  MD24484 $MC_TRAFO_GEOAX_ASSIGN_TAB_10

Orientierungstransformationen

Maschinendaten, die für Orientierungstransformationen relevant sind:

- MD24550 $MC_TRAFO5_BASE_TOOL_1 und
  MD24650 $MC_TRAFO5_BASE_TOOL_2
- MD24558 $MC_TRAFO5_JOINT_OFFSET_1 und
  MD24658 $MC_TRAFO5_JOINT_OFFSET_2
- MD24500 $MC_TRAFO5_PART_OFFSET_1 und
  MD24600 $MC_TRAFO5_PART_OFFSET_2
- MD24510 $MC_TRAFO5_ROT_AX_OFFSET_1 und
  MD24610 $MC_TRAFO5_ROT_AX_OFFSET_2
- MD24520 $MC_TRAFO5_ROT_SIGN_IS_PLUS_1 und
  MD24620 $MC_TRAFO5_ROT_SIGN_IS_PLUS_2
- MD 24530: TRAFO5_NON_POLE_LIMIT_1 und
  MD24630 $MC_TRAFO5_NON_POLE_LIMIT_2
- MD24540 $MC_TRAFO5_POLE_LIMIT_1 und
  MD24640 $MC_TRAFO5_POLE_LIMIT_2
- MD24570 $MC_TRAFO5_AXIS1_1 und
  MD24670 $MC_TRAFO5_AXIS1_2

- MD24572 $MC_RAFO5_AXIS2_1 und
  MD24672 $MC_TRAFO5_AXIS2_2
- MD24574 $MC_TRAFO5_BASE_ORIENT_1 und
  MD24674 $MC_TRAFO5_BASE_ORIENT_2
- MD24562 $MC_TRAFO5_TOOL_ROT_AX_OFFSET_1 und
  MD24662 $MC_TRAFO5_TOOL_ROT_AX_OFFSET_2
- MD24564 $MC_TRAFO5_NUTATOR_AX_ANGLE_1 und
  MD24664 $MC_TRAFO5_NUTATOR_AX_ANGLE_2
- MD24566 $MC_TRAFO5_NUTATOR_VIRT_ORIAX_1 und
  MD24666 $MC_TRAFO5_NUTATOR_VIRT_ORIAX_2

Transmit-Transformationen

Maschinendaten, die für Transmit-Transformationen relevant sind:

- MD24920 $MC_TRANSMIT_BASE_TOOL_1 und
  MD24970 $MC_TRANSMIT_BASE_TOOL_2
- MD24900 $MC_TRANSMIT_ROT_AX_OFFSET_1 und
  MD24950 $MC_TRANSMIT_ROT_AX_OFFSET_2
- MD24910 $MC_TRANSMIT_ROT_SIGN_IS_PLUS_1 und
  MD24960 $MC_TRANSMIT_ROT_SIGN_IS_PLUS_2
- MD24911 MC_RANSMIT_POLE_SIDE_FIX_1 und
  MD24961 $MC_TRANSMIT_POLE_SIDE_FIX_2

Tracyl-Transformationen

Maschinendaten, die für Tracyl-Transformationen relevant sind:

- MD24820 $MC_TRACYL_BASE_TOOL_1 und
  MD24870 $MC_TRACYL_BASE_TOOL_2
- MD24800 $MC_TRACYL_ROT_AX_OFFSET_1 und
  MD24850 $MC_TRACYL_ROT_AX_OFFSET_2
- MD24810 $MC_TRACYL_ROT_SIGN_IS_PLUS_1 und
  MD24870 $MC_TRACYL_ROT_SIGN_IS_PLUS_2
- MD24808 $MC_TRACYL_DEFAULT_MODE_1 und
  MD24858 $MC_TRACYL_DEFAULT_MODE_2

Schräge-Achse-Transformationen

Maschinendaten, die für Schräge-Achse-Transformationen relevant sind:

- MD24710 $MC_TRAANG_BASE_TOOL_1 und
  MD24760 $MC_TRAANG_BASE_TOOL_2
- MD24700 $MC_TRAANG_ANGLE_1 und
  MD24750 $MC_TRAANG_ANGLE_2
- MD24720 $MC_TRAANG_PARALLEL_VELO_RES_1 und
  MD24770 $MC_TRAANG_PARALLEL_VELO_RES_2
- MD24721 $MC_TRAANG_PARALLEL_ACCEL_RES_1 und
  MD24771 $MC_TRAANG_PARALLEL_ACCEL_RES_2

Verkettete Transformationen

Maschinendaten, die für verkettete Transformationen relevant sind:

- MD24995 $MC_TRACON_CHAIN_1 und
  MD24996 $MC_TRACON_CHAIN_2
- MD24997 $MC_TRACON_CHAIN_3 und
  MD24998 $MC_TRACON_CHAIN_4

Persistente Transformation

Maschinendaten, die für persistente Transformationen relevant sind:

- MD20144 $MC_TRAFO_MODE_MASK
- MD20140 $MC_TRAFO_RESET_VALUE
- MD20110 $MC_RESET_MODE_MASK und
  MD20112 $MC_START_MODE_MASK

Nicht transformationsspezifisch

Maschinendaten, die nicht transformationsspezifisch sind. Sie sind nicht eindeutig einem bestimmten Transformationsdatensatz zugeordnet bzw. haben Bedeutung auch außerhalb einer aktiven Transformation:

- MD21110 $MC_X_AXIS_IN_OLD_X_Z_PLANE
- MD21090 $MC_ MAX_LEAD_ANGLE
- MD21092 $MC_ MAX_TILT_ANGLE
- MD21100 $MC_ORIENTATION_IS_EULER

7.9 Randbedingungen

7.9.1 Verkettete Transformationen

Es können **zwei** Transformationen verkettet werden.

Es können jedoch nicht beliebige Transformationen miteinander verkettet werden.

Es gelten folgende Einschränkungen:

- Die **erste** Transformation in der Kette muss eine der folgenden Transformationen sein:
  - Orientierungstransformation (3-, 4-, 5-Achs-Transformation, kardanischer Fräskopf)
  - Transmit
  - Mantellinientransformation
  - Schräge Achse
- Die **zweite** Transformation muss eine **schräge** Achsentransformation sein.
- Es dürfen nur zwei Transformationen verkettet werden.

Es ist zulässig (z. B. zu Testzwecken), in die Verkettungsliste nur eine einzelne Transformation einzutragen.

7.10 Beispiele

7.10.1 TRANSMIT

Für die im folgenden Bild skizzierte Konfiguration wird ein Beispiel im Zusammenhang notiert, das die wesentlichen Schritte für Konfiguration der Achsen bis zur Aktivierung von TRANSMIT enthält.

```
; Allgemeine Achskonfiguration für Drehen
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[0] = "X"        : Geometrieachse
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[1] = "Y"        : Geometrieachse
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"        : Geometrieachse
MD20060 $MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1        : X als Kanalachse 1
MD20060 $MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0        : Y keine Kanalachse
MD20060 $MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 2        : Z als Kanalachse 2
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[0] = "XC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[1] = "ZC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[2] = "CC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[3] = "ASC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[4] = " "
MD20070 $MC_AXCONF_MACHAX_USED[0] = 2             : XC als Maschinenachse 2
```

```
MD20070 $MC_AXCONF_MACHAX_USED[1] = 3          : ZC als Maschinenachse 3
MD20070 $MC_AXCONF_MACHAX_USED[2] = 1          : CC als Maschinenachse 1
MD20070 $MC_AXCONF_MACHAX_USED[3] = 4          : ASC als Maschinenachse
                                                 4
MD20070 $MC_AXCONF_MACHAX_USED[3] = 0          : leer
MD20070 $MA_SPIND_ASSIGN_TO_MACHAX[AX1]= 1     : C ist Spindel 1
MD20070 $MA_SPIND_ASSIGN_TO_MACHAX[AX2]= 0     : X ist keine Spindel
MD20070 $MA_SPIND_ASSIGN_TO_MACHAX[AX3]= 0     : Z ist keine Spindel
MD20070 $MA_SPIND_ASSIGN_TO_MACHAX[AX4]= 2     : AS ist Spindel 2
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[0]= "CM"    : 1. Maschinenachse
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[1]= "XM"    : 2. Maschinenachse
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[2]= "ZM"    : 3. Maschinenachse
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[3]= "ASM"   : 4. Maschinenachse
```

```
; Vorbereiten für TRANSMIT (als erste und einzige Transformation)
$MA_ROT_IS_MODULO[3] = TRUE                    ; c als Moduloachse
MD24100 $MC_TRAFO_TYPE_1 = 256                 ; Transformation TRANSMIT
MD24110 $MC_TRAFO_AXES_IN_1[0] = 1 ;           ; Kanalachse senkrecht zur
                                                 Rundachse
MD24110 $MC_TRAFO_AXES_IN_1[1] = 3             ; Kanalachse Rundachse
MD24110 $MC_TRAFO_AXES_IN_1[2] = 2             ; Kanalachse parallel zur Rundachse
MD24120$MC_TRAFO_GEOAX_ASSIGN_TAB_1[           ; 1. Kanalachse wird GEOAX X
0]=1
MD24120                                        ; 2. Kanalachse wird GEOAX Y
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=3
MD24120                                        ; 3. Kanalachse wird GEOAX Z
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=2
MD24900                                        ; Drehlage X-Y-Ebene gegen
$MC_TRANSMIT_ROT_AX_OFFSET_1=0                 Nullstell. der Rundachse
MD24910                                        ; Rundachse dreht
$MC_TRANSMIT_ROT_SIGN_IS_PLUS_1=FALSE
MD24920 $MC_TRANSMIT_BASE_TOOL_1               ; WZ-Abstand in X
[0]=0.0
MD24920 $MC_TRANSMIT_BASE_TOOL_1               ; WZ-Abstand in Y
[1]=0.0
MD24920 $MC_TRANSMIT_BASE_TOOL_1               ; WZ-Abstand in Z
[2]=0.0
; Aktivierung TRANSMIT
; Programmierung in X,Y, Z
; Rückkehr zum Drehbetrieb
TRAFOOF
```

7.10.2 TRACYL

Für die im folgenden Bild skizzierte Konfiguration wird ein Beispiel im Zusammenhang notiert, das die wesentlichen Schritte für Konfiguration der Achsen bis zur Aktivierung von TRACYL enthält.

```
; Allgemeine Achskonfiguration für Drehen
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[0] = "X"          ; Geometrieachse
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[1] = "Y"          ; Geometrieachse
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"          ; Geometrieachse
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1          ; X als Kanalachse 1
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 2          ; Y keine Kanalachse
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3          ; Z als Kanalachse 2
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[0] = "XC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[1] = "YC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[2] = "ZC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[3] = "CC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[4] = "ASC"
MD20070 $MC_AXCONF_MACHAX_USED[0] = 2               ; X als Maschinenachse 2
MD20070 $MC_AXCONF_MACHAX_USED[1] = 3               ; Y als Maschinenachse 3
MD20070 $MC_AXCONF_MACHAX_USED[2] = 4               ; Z als Maschinenachse 4
MD20070 $MC_AXCONF_MACHAX_USED[3] = 1               ; C als Maschinenachse 1
MD20070 $MC_AXCONF_MACHAX_USED[4] = 5               ; AS als Maschinenachse
                                                      5
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX1]= 1          ; C ist Spindel 1
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX2]= 0          ; X ist keine Spindel
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX3]= 0          ; Y ist keine Spindel
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX4]= 0          ; Z ist keine Spindel
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX5]= 2          ; AS ist Spindel 2
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[0]= "CM"         ; 1. Maschinenachse
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[1]= "XM"         ; 2. Maschinenachse
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[2]= "YM"         ; 3. Maschinenachse
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[3]= "ZM"         ; 4. Maschinenachse
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[4]= "ASM"        ; 5. Maschinenachse
```

```
; Vorbereiten für TRACYL (erste und einzige Transformation)
MD24100 $MC_TRAFO_TYPE_1 = 513 ;    ; Transformation TRACYL mit
                                      Nutwandkorrektur
MD24110 $MC_TRAFO_AXES_IN_1[0] = 1  ; Kanalachse radial zur Rundachse
MD24110 $MC_TRAFO_AXES_IN_1[1] = 4  ; Kanalachse in Zylindermantelfläche
                                      senkrecht zur Rundachse
MD24110 $MC_TRAFO_AXES_IN_1[2] = 3  ; Kanalachse parallel zur Rundachs
```

```
MD24110 $MC_TRAFO_AXES_IN_1[3] = 2     ; Kanalachse Zusatzachse zu Index [0]
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0] = 1     ; 1. Kanalachse wird GEOAX X
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1] = 4     ; 2. Kanalachse wird GEOAX Y
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2] = 3     ; 3. Kanalachse wird GEOAX Z
MD24800 $MC_TRACYL_ROT_AX_OFFSET_1 = 0     ; Drehlage X-Y-Ebene gegen Nullstell. der Rundachse
MD24810 $MC_TRACYL_ROT_SIGN_IS_PLUS_1 = FALSE     ; Rundachse dreht
MD24820 $MC_TRACYL_BASE_TOOL_1 [0] = 0.0     ; WZ-Abstand in X
MD24820 $MC_TRACYL_BASE_TOOL_1 [1] = 0.0     ; WZ-Abstand in Y
MD24820 $MC_TRACYL_BASE_TOOL_1 [2] = 0.0     ; WZ-Abstand in Z
; Aktivierung TRACYL(40.0)
; Programmierung in Y und Z siehe unten
; Rückkehr zum Drehbetrieb
TRAFOOF
```

Programmierung mit Nutwandkorrektur

(TRAFO_TYPE_n=513)

Kontur

Eine Nut, die breiter als das Werkzeug ist, erzeugt man, indem man relativ zur programmierten Bezugskontur die Korrekturrichtung (G41, G42) und den Abstand der Nutseitenwand zur Bezugskontur über die Adresse OFFN programmiert (siehe Bild).

Werkzeugradius

Der Werkzeugradius wird bezüglich der Nutseitenwand automatisch berücksichtigt (siehe Bild). Die volle Funktionalität der ebenen Werkzeugradiuskorrektur steht zur Verfügung (stetiger Übergang an Außen- und Innenecken sowie Lösung von Flaschenhalsproblemen).

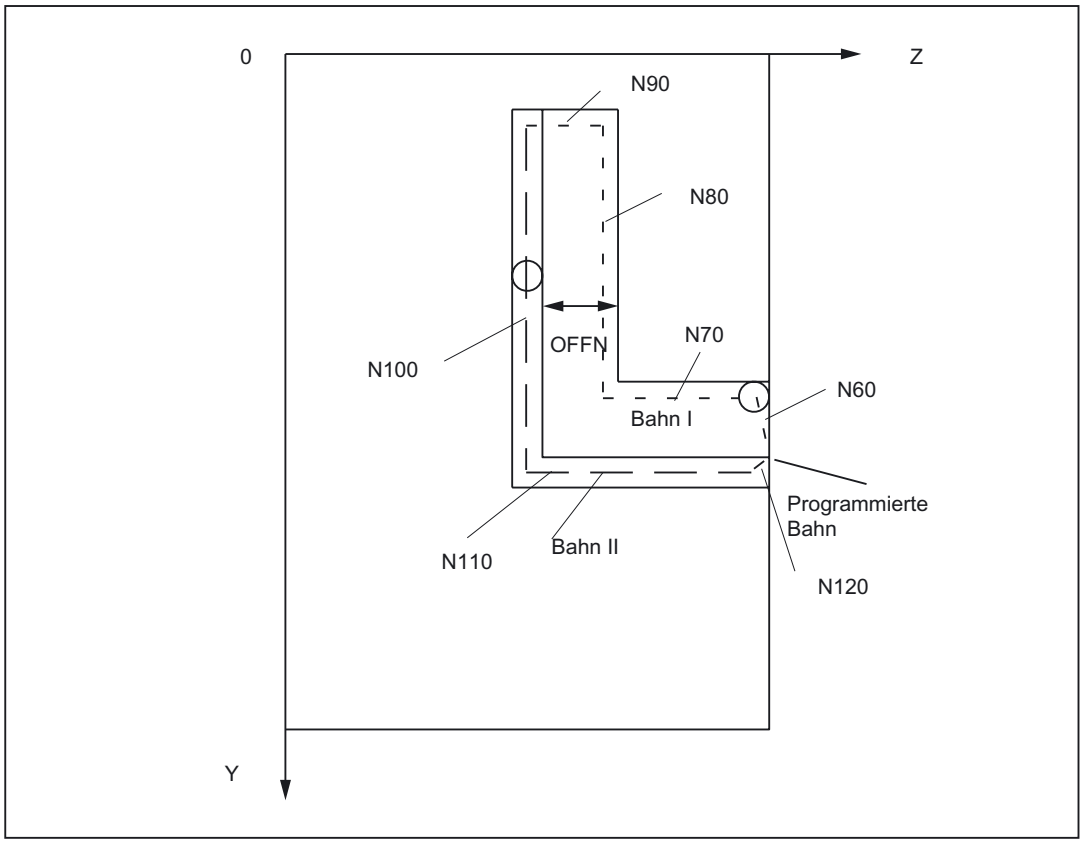


Bild 7-37 Nut mit Nutwandkorrektur, Zylinderkoordinaten (Skizze vereinfacht).

; Beispielprogramm, das nach der Transformationsanwahl das Werkzeug

; auf der Bahn I über die Bahn II zurück zum Ausgangspunkt führt

; (Maschinendaten siehe "Datenbeschreibungen", Beispiel X-Y-Z-C-Kinematik):

```
N1 SPOS=0;                              ; Übernahme der Spindel in den
                                        Rundachsbetrieb
N5 G0 X25 Y0 Z105 CC=200 F5000 G64      ; Positionieren der Maschine über der
                                        Nutmitte
N10 TRACYL(40.)                         ; Transformationsanwahl mit
                                        Bezugsdurchmesser
                                        :40 mm
N20 G19 G90                             ; Bearbeitungsebene ist
                                        Zylindermantelfläche
N30 T1 D1                               ; Werkzeuganwahl, kann auch vor TRACY
                                        (..) stehen
N40 G1 X20                              ; Werkzeug auf Nutgrund zustellen
```

```
N50 OFFN=12                                    ; Nutwandabstand festlegen, muss nicht
                                                in
                                               ; einer eigenen Zeile stehen
```

```
; Anfahren der Nutwand
N60 G1 Z100 G42                                ; WRK-Anwahl zum Anfahren der
                                                Nutwand
Fertigen des Nutabschnittes Bahn I
N70 G1 Z50                                     ; Nutteil parallel zur
                                                Zylinderebene
N80 G1 Y10                                     ; Nutteil parallel zum Umfang
; Anfahren der Nutwand für Bahn II
N90 OFFN=4 G42                                 ; Nutwandabstand festlegen und
                                                WRK-Anwahl
                                               ; zum Anfahren der Nutwand
; Fertigen des Nutabschnittes Bahn II
N100 G1 Y70                                    ; entspricht CC=200 Grad
N110 G1 Z100                                   ; zum Ausgangswert zurück
; Abfahren von der Nutwand
N120 G1 Z105 G40                               ; WRK-Abwahl zum Wegfahren von
                                                der Nutwand
N130 G0 X25                                    ; Abheben von der Nut
N140 TRAFOOF N150 G0 X25 Y0 Z105 CC=200 D0     ; zum Ausgangspunkt zurück und
                                                Abwahl
                                               ; der Werkzeugkorrektur
```

Programmierung ohne Nutwandkorrektur

TRACYL ohne Nutwandkorrektur mitzusätzlicher Linearachse (TRAFO_TYPE_n = 514)

```
;Für das folgende Teileprogramm wird folgende Maschinendateneinstellung
vorausgesetzt:
MD20070 $MC_AXCONF_MACHAX_USED[0] = 1            ; X als Maschinenachse 1
MD20070 $MC_AXCONF_MACHAX_USED[1] = 2            ; Y als Maschinenachse 2
MD20070 $MC_AXCONF_MACHAX_USED[2] = 3            ; Z als Maschinenachse 3
MD20070 $MC_AXCONF_MACHAX_USED[3] = 4            ; C als Maschinenachse 4
MD20070 $MC_AXCONF_CHANAX_NAME_TAB[1] = "Y2"
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0] = 1;    ; X als Kanalachse 1
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1] = 2;    ; Y keine Kanalachse
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2] = 3;    ; Z als Kanalachse 2
MD24100 $MC_TRAFO_TYPE_1 = 514                   ; TRACYL ohne
                                                  Nutwandkorrektur mit
                                                  erweiterter
                                                  Werkzeuglängenkorrektur
MD24110 $MC_TRAFO_AXES_IN_1[0] = 1               ; Kanalachse radial zur
                                                  Rundachse
```

```
MD24110 $MC_TRAFO_AXES_IN_1[1] = 4              ; Kanalachse in
                                                Zylindermantelfläche
                                                ; senkrecht zur Rundachse
MD24110 $MC_TRAFO_AXES_IN_1[2] = 3              ; Kanalachse parallel zur
                                                Rundachse
MD24110 $MC_TRAFO_AXES_IN_1[3] = 2              ; Kanalachse Zusatzachse
                                                zu Index [0]
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0] = 1    ; 1. Kanalachse wird
                                                GEOAX X
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1] = 4    ; 2. Kanalachse wird
                                                GEOAX Y
MD24110 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2] = 3    ; 3. Kanalachse wird
                                                GEOAX Z
MD24808 $MC_TRACYL DEFAULT MODE_1 =0            ; oder gar nicht gesetzt
```

Werkzeugdaten:

```
$TC_DP1[1,1]=120                ; Werkzeugtyp Schaftfräser
$TC_DP2[1,1]=0
$TC_DP3[1,1]=0                  ; Längenkorrekturvektor
$TC_DP4[1,1]=25
$TC_DP5[1,1]=5
$TC_DP6[1,1]=4                  ; Radius, Werkzeugradius
```

Teileprogramm:

```
N1001 T1 D1 G54 G19 G90 F5000 G64    ; Anwahl der 1. TRACYL ohne
N1005 G0 X25 Y0 Z105 A=200           Nutwandkorrektur
N1010 TRACYL(40.)                    ; Transformationsanwahl
N1040 G1 X20
N1060 G1 Z100
N1070 G1 Z50
N1080 G1 Y10
N1140 TROFOOF
N1150 G0 X25 Y0 Z105 A=200           ; Anwahl der 1. TRACYL mit
                                     Nutwandkorrektur
N2010 G0 TRACYL(40.,1,1)             ; möglich wäre auch TRCYL(40., ,1)
N2040 G1 X20
N2060 G1 Z100
N2070 G1 Z50
N2080 G1 Y10
N2140 TROFOOF
```

7.10.3 TRAANG

Für die im Bild "Nut mit Nutwandkorrektur, Zylinderkoordinaten" skizzierte Konfiguration wird ein Beispiel im Zusammenhang notiert, das die wesentlichen Schritte für Konfiguration der Achsen bis zur Aktivierung von TRAANG enthält.

```
; Allgemeine Achskonfiguration für Schleifen
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[0] = "X"          ; Geometrieachse
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[1] = " "          ; Geometrieachse
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"          ; Geometrieachse
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 0          ; X keine Kanalachse
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0          ; Y keine Kanalachse
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 1          ; Z als Kanalachse 1
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[0] = "Z"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[1] = "C"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[2] = "AS1"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[3] = "MU"
MD20070 $MC_AXCONF_MACHAX_USED[0] = 3               ; Z als Maschinenachse 3
MD20070 $MC_AXCONF_MACHAX_USED[1] = 1               ; C als Maschinenachse 1
MD20070 $MC_AXCONF_MACHAX_USED[2] = 4               ; AS als Maschinenachse
                                                    4
MD20070 $MC_AXCONF_MACHAX_USED[3] = 2               ; MU als Maschinenachse
                                                    2
MD20070 $MC_AXCONF_MACHAX_USED[3] = 0               ; leer
MD20070 $MC_AXCONF_MACHAX_USED[3] = 0               ; leer
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX1]= 1          ; C ist Spindel 1
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX2]= 0          ; X ist keine Spindel
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX3]= 0          ; Z ist keine Spindel
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX4]= 2          ; AS ist Spindel 2
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[0]= "C1"         ; 1. Maschinenachse
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[1]= "MU"         ; 2. Maschinenachse
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[2]= "MZ"         ; 3. Maschinenachse
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[3]="AS1"         ; 4. Maschinenachse
```

```
; Vorbereiten für TRAANG (erste und einzige Transformation)
MD24100 $MC_TRAFO_TYPE_1 = 1024                     ; Transformation TRAANG
MD24110 $MC_TRAFO_AXES_IN_1[0] = 4                  ; Kanalachse Schräge Achse
MD24110 $MC_TRAFO_AXES_IN_1[1] = 1                  ; Kanalachse parallel zur Achse Z
MD24110 $MC_TRAFO_AXES_IN_1[2] = 0                  ; Kanalachse nicht aktiv
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0]            ; X 1. Kanalachse
= 4
```

```
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1]    ; Y 2. Kanalachse
= 0
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2]    ; Z 3. Kanalachse
= 1
MD24700 $MC_TRAANG_ANGLE_1 = 30.            ; Winkel der Schrägen Achse
MD24710 $MC_TRAANG_BASE_TOOL_1 [0] = 0      ; WZ-Abstand in X
MD24710 $MC_TRAANG_BASE_TOOL_1 [1] = 0      ; WZ-Abstand in Y
MD24710 $MC_TRAANG_BASE_TOOL_1 [2] = 0      ; WZ-Abstand in Z
TRAANG                                      ; Aktivierung
                                            ; Programmierung in X,Y, Z
TRAFOOF                                     ; Rückkehr zum Drehbetrieb
```

7.10.4 Verkettete Transformationen

Beispiele

Im folgenden Kapitel wird festgelegt:

- die allgemeine Kanalkonfiguration
- einzelne Transformationen
- verkettete Transformationen aus zuvor definierten Einzeltransformationen
- die Aktivierung von Einzeltransformationen
- die Aktivierung von verketteten Transformationen

Die Beispiele umfassen folgende Transformationen:

- 5-Achs-Transformation mit drehbarem Werkzeug und Achsfolge AB (Trafotyp 16)
- Transmit (Trafotyp 256)
- Schräge Achse (Trafotyp 1024)
- Verkettung der 1. und der 3. Transformation (Trafotyp 8192)
- Verkettung der 2. und der 3. Transformation (Trafotyp 8192)

Allgemeine Kanalkonfiguration

```
CHANDATA (1)                                ; Kanaldaten im Kanal 1
MD20070 $MC_AXCONF_MACHAX_USED[0] = 1
MD20070 $MC_AXCONF_MACHAX_USED[1] = 2
MD20070 $MC_AXCONF_MACHAX_USED[2] = 3
MD20070 $MC_AXCONF_MACHAX_USED[3] = 4
MD20070 $MC_AXCONF_MACHAX_USED[4] = 5
MD20070 $MC_AXCONF_MACHAX_USED[5] = 6
MD20070 $MC_AXCONF_MACHAX_USED[6] = 7
```

```
MD20070 $MC_AXCONF_MACHAX_USED[7] = 0
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[3]="A"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[4]="B"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[5]="C"
MD36902 $MA_IS_ROT_AX[ AX4 ] = TRUE
MD36902 $MA_IS_ROT_AX[ AX5 ] = TRUE
MD36902 $MA_IS_ROT_AX[ AX6 ] = TRUE
MD36902 $MA_IS_ROT_AX[ AX7 ] = TRUE
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[ AX5 ] = 0
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX7] = 1
MD35000 $MA_ROT_IS_MODULO[AX7] = TRUE
```

Einzeltransformationen

```
; 1. TRAORI
MD24470 $MC_TRAFO_TYPE_1= 16          ; TRAORI: A-B-Kinematik
MD24410 $MC_TRAFO_AXES_IN_1[0]=1
MD24410 $MC_TRAFO_AXES_IN_1[1]=2
MD24410 $MC_TRAFO_AXES_IN_1[2]=3
MD24410 $MC_TRAFO_AXES_IN_1[3]=4
MD24410 $MC_TRAFO_AXES_IN_1[4]=5
MD24410 $MC_TRAFO_AXES_IN_1[5]=0
MD24120$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1
MD24120$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=2
MD24120$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3
MD24550$MC_TRAFO5_BASE_TOOL_1[0]=0
MD24550$MC_TRAFO5_BASE_TOOL_1[1]=0
MD24550$MC_TRAFO5_BASE_TOOL_1[2]=0


; 2. TRANSMIT
MD24200 $MC_TRAFO_TYPE_2 = 256        ;TRANSMIT
MD24210 $MC_TRAFO_AXES_IN_2[0]=1
MD24210 $MC_TRAFO_AXES_IN_2[1]=6
MD24210 $MC_TRAFO_AXES_IN_2[2]=3
MD24210 $MC_TRAFO_AXES_IN_2[3]=0
MD24210 $MC_TRAFO_AXES_IN_2[4]=0
MD24210 $MC_TRAFO_AXES_IN_2[5]=0
MD24210 $MC_TRAFO_AXES_IN_2[6]=0
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[0]=1
```

```
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[1]=6
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[2]=3
; 3. TRAANG
MD24300 $MC_TRAFO_TYPE_3 = 1024        ;TRAANG
MD24310 $MC_TRAFO_AXES_IN_3[0] = 1
MD24310 $MC_TRAFO_AXES_IN_3[1] = 3
MD24310 $MC_TRAFO_AXES_IN_3[2] = 2
MD24310 $MC_TRAFO_AXES_IN_3[3] = 0
MD24310 $MC_TRAFO_AXES_IN_3[4] = 0
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[0] = 1
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[1] = 3
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[2] = 2
MD24700 $MC_TRAANG_ANGLE_1 = 45.
MD24720 $MC_TRAANG_PARALLEL_VELO_RES_1 = 0.2
MD24721 $MC_TRAANG_PARALLEL_ACCEL_RES_1 = 0.2
MD24710 $MC_TRAANG_BASE_TOOL_1[0] = 0.0
MD24710 $MC_TRAANG_BASE_TOOL_1[1] = 0.0
MD24710 $MC_TRAANG_BASE_TOOL_1[2] = 0.0
```

Verkettete Transformationen

```
; 4. TRACON (Verkettung TRAORI/TRAANG)
MD24400 $MC_TRAFO_TYPE_4 = 8192
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[0] = 2
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[1] = 1
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[2] = 3
MD24995 $MC_TRACON_CHAIN_1[0] = 1
MD24995 $MC_TRACON_CHAIN_1[1] = 3
MD24995 $MC_TRACON_CHAIN_1[2] = 0
; 5. TRACON (Verkettung TRANSMIT/TRAANG)
MD24430 $MC_TRAFO_TYPE_5 = 8192
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[0] = 1
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[1] = 6
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[2] = 3
MD24996 $MC_TRACON_CHAIN_2[0] = 2
MD24996 $MC_TRACON_CHAIN_2[1] = 3
MD24996 $MC_TRACON_CHAIN_2[2] = 0
```

Teileprogramm (Auszüge)

Beispiel für ein NC-Programm, das die eingestellten Transformationen verwendet:

```
; Einzeltransformationen aufrufen
                      ; Werkzeugfestlegung
$TC_DP1[1,1]=120      ; Werkzeugtyp
$TC_DP3[1,1]= 10      ; Werkzeuglänge
n2 x0 y0 z0 a0 b0 f20000 t1 d1n4 x20
n30 TRANSMIT          ; Transmit einschalten
n40 x0 y20
n50 x-20 y0
n60 x0 y-20
n70 x20 y0
n80 TRAFOOF           ; Transmit ausschalten
n130 TRAANG(45.)      ; Schräge-Achse-Transformation einschalten, Parameter: Winkel
                      45°
n140 x0 y0 z20
n150 x-20 z0
n160 x0 z-20
n170 x20 z0
```

Hinweis

Für die obigen Beispiele wird davon ausgegangen, dass der Winkel der "Schrägen Achse" an der Maschine einstellbar ist und dass er bei Aktivierung der Einzeltransformationen auf 0° eingestellt ist.

```
; 1. Verkettete Transformationen aufrufen
; TRAORI + TRAANG
n230 TRACON(1, 45.)     ; 1. der 2 verketteten Transformationen einschalten
                        ; Die vorher aktive Transformation TRAANG wird automatisch
                        abgewählt
                        ; Der Parameter für die schräge Achse ist 45°
n240 x10 y0 z0 a3=-1 C3 =1 oriwks
n250 x10 y20 b3 = 1 c3 = 1
```

```
; 2. Verkettete Transformationen aufrufen
; TRANSMIT + TRAANG
n330 TRACON(2, 40.)     ; 2. verkettete Transformation einschalten
                        ; Der Parameter für die schräge Achse ist 40°
n335 x20 y0 z0
```

```
n340 x0 y20 z10
n350 x-20 y0 z0
n360 x0 y-20 z0
n370 x20 y0 z0
n380 TRAFOOF            ; 2. verkettete Transformation ausschalten
n1000 M30
```

7.10.5 Wirksamsetzen von Transformations-MD über Teileprogramm

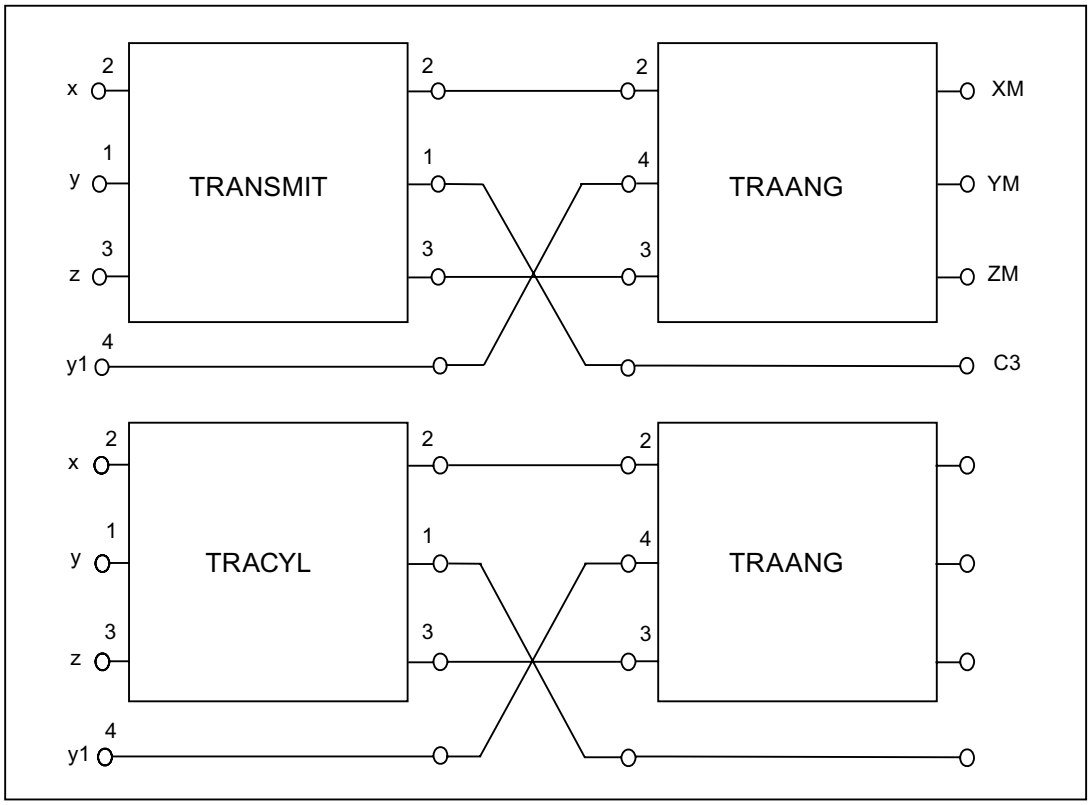
Im folgenden Beispiel wäre es zulässig, im Satz N90 auch ein Maschinendatum zu beschreiben, das die zweite Transformation betrifft (z. B. MD24650 $MC_TRAFO5_BASE_TOOL_2[2]), da das Maschinendatum durch Beschreiben allein nicht wirksam wird. Dies würde jedoch bei sonst unverändertem Programm im Satz N130 zu einem Alarm führen, da dann versucht würde, eine aktive Transformation zu modifizieren.

Beispielprogramm:

```
N40 TRAORI(2)                              ; Anwahl 2. Orientierungstransformation
N50 X0 Y0 Z0 F20000 T1 T1
N60 A50 B50
N70 A0 B0
N80 X10
N90 $MC_TRAFO5_BASE_TOOL_1[2] = 50         ; Überschreiben eines MD der
                                           ; 1. Orientierungstransformation
N100 A20
N110 X20
N120 X0
N130 NEWCONF                               ; Neu beschriebenes MD übernehmen
N140 TRAORI(1)                             ; Anwahl 1. Orientierungstransformation-
                                           MD
                                           ; wird wirksam
N150 G19 X0 Y0 Z0
N160 A50 B50
N170 A0 B0
N180 TRAFOOF
N190 M30
```

7.10.6 Achspositionen in der Transformationskette

Im folgenden Beispiel werden zwei verkettete Transformationen projektiert und im Teileprogramm werden die Systemvariablen zum Bestimmen der Achspositionen in der Synchronaktion zyklisch gelesen.



Maschinendaten

```
CHANDATA(1)

MD24100 $MC_TRAFO_TYPE_1=256                    ; TRANSMIT
MD24110 $MC_TRAFO_AXES_IN_1[0]=2
MD24110 $MC_TRAFO_AXES_IN_1[1]=1
MD24110 $MC_TRAFO_AXES_IN_1[2]=3
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=2
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=1
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3

MD24200 $MC_TRAFO_TYPE_2=512                    ; TRACYL
MD24210 $MC_TRAFO_AXES_IN_2[0]=2
MD24210 $MC_TRAFO_AXES_IN_2[1]=1
MD24210 $MC_TRAFO_AXES_IN_2[2]=3
```

```
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[0]=2
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[1]=1
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[2]=3

MD24300 $MC_TRAFO_TYPE_3=1024                    ; TRAANG
MD24310 $MC_TRAFO_AXES_IN_3[0]=2
MD24310 $MC_TRAFO_AXES_IN_3[1]=4
MD24310 $MC_TRAFO_AXES_IN_3[2]=3
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[0]=2
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[1]=4
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[2]=3
MD24700 $MC_TRAANG_ANGLE_1 = 45.
MD24720 $MC_TRAANG_PARALLEL_VELO_RES_1 = 0.2
MD24721 $MC_TRAANG_PARALLEL_ACCEL_RES_1 = 0.2
MD24710 $MC_TRAANG_BASE_TOOL_1[0] = 0.0
MD24710 $MC_TRAANG_BASE_TOOL_1[1] = 0.0
MD24710 $MC_TRAANG_BASE_TOOL_1[2] = 0.0
```

1. Verkettung TRANSMIT / TRAANG

```
MD24400 $MC_TRAFO_TYPE_4=8192                    ; TRACON (1)
MD24995 $MC_TRACON_CHAIN_1[0]=1
MD24995 $MC_TRACON_CHAIN_1[1]=3
MD24995 $MC_TRACON_CHAIN_1[2]=0
MD24995 $MC_TRACON_CHAIN_1[3]=0
MD24410 $MC_TRAFO_AXES_IN_4[0]=1
MD24410 $MC_TRAFO_AXES_IN_4[1]=2
MD24410 $MC_TRAFO_AXES_IN_4[2]=3
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[0]=2
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[1]=1
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[2]=3
```

2. Verkettung TRACYL / TRAANG

```
MD24430 $MC_TRAFO_TYPE_5=8192                    ; TRACON (2)
MD24996 $MC_TRACON_CHAIN_2[0]=2
MD24996 $MC_TRACON_CHAIN_2[1]=3
MD24996 $MC_TRACON_CHAIN_2[2]=0
MD24996 $MC_TRACON_CHAIN_2[3]=0
MD24432 $MC_TRAFO_AXES_IN_5[0]=1
MD24432 $MC_TRAFO_AXES_IN_5[1]=2
MD24432 $MC_TRAFO_AXES_IN_5[2]=3
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[0]=2
```

MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[1]=1
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[2]=3

M17

Teileprogramm

```
Programmcode                                  Kommentar
N10 $TC_DP1[1,1]=120
N20 $TC_DP3[1,1]=20
N30 $TC_DP4[1,1]=0
N40 $TC_DP5[1,1]=0
N50
N60 X0 Y0 Z0 F20000 T1 D1
N70
N80                                           ; zyklisches Lesen der Variablen in der Synchronaktion
N90 ID=1 WHENEVER TRUE DO $R0=$AA_ITR[X,0] $R1=$AA_ITR[X,1] $R2=$AA_ITR[X,2]
N100 ID=2 WHENEVER TRUE DO $R3=$AA_IBC[X] $R4=$AA_IBC[Y] $R5=$AA_IBC[Z]
N110 ID=3 WHENEVER TRUE DO $R6=$VA_IW[X]-$AA_IW[X]
N120 ID=4 WHENEVER TRUE DO $R7=$VA_IB[X]-$AA_IB[X]
N130 ID=5 WHENEVER TRUE DO $R8=$VA_IBC[X]-$AA_IBC[X]
N140 ID=6 WHENEVER TRUE DO $R9=$VA_ITR[X,1]-$AA_ITR[X,1]
N150
N160                                          ; 1. Verkettung TRANSMIT / TRAANG
N170 TRACON(1,)
N180 X20 Y0 Z0
N190 X0 Y20 Z10
N200 X-20 Y0 Z0
N210 X0 Y-20 Z0
N220 X20 Y0 Z0
N230 TRAFOOF
N240
N250                                          ; 2. Verkettung TRACYL / TRAANG
N260 TRACON (2, 40.)
N270 X20 Y0 Z0
N280 X0 Y20 Z10
N290 X-20 Y0 Z0
N300 X0 Y-20 Z0
N310 X20 Y0 Z0
N320 TRAFOOF
N330
N340 M30
```

7.11 Datenlisten

7.11.1 Maschinendaten

7.11.1.1 TRANSMIT

Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
20110	RESET_MODE_MASK	Festlegung der Steuerungsgrundstellung nach Hochlauf und RESET/Teileprogrammende
20140	TRAFO_RESET_VALUE	Grundstellung Transformation
22534	TRAFO_CHANGE_M_CODE	M-Code bei Transformationsumschaltung der Geometrieachsen
24100	TRAFO_TYPE_1	Definition der 1.Transformation im Kanal
24110	TRAFO_AXES_IN_1	Achszuordnung für die 1. Transformation
24120	TRAFO_GEOAX_ASSIGN_TAB_1	Zuordnung Geo.-Achsen bei 1. Transf.
24200	TRAFO_TYPE_2	Definition der 2. Transformation im Kanal
24210	TRAFO_AXES_IN_2	Achszuordnung für die 2.Transformation
24220	TRAFO_GEOAX_ASSIGN_TAB_2	Zuordnung Geo.-Achsen bei 2. Transf.
24300	TRAFO_TYPE_3	Definition der 3.Transformation im Kanal
24310	TRAFO_AXES_IN_3	Achszuordnung für die 3.Transformation
24320	TRAFO_GEOAX_ASSIGN_TAB_3	Zuordnung Geo.-Achsen bei 3. Transf.
24400	TRAFO_TYPE_4	Definition der 4.Transformation im Kanal
24410	TRAFO_AXES_IN_4	Achszuordnung für die 4.Transformation
24420	TRAFO_GEOAX_ASSIGN_TAB_4	Zuordnung Geo.-Achsen bei 4. Transf.
24430	TRAFO_TYPE_5	Definition der 5.Transformation im Kanal
24432	TRAFO_AXES_IN_5	Achszuordnung für die 5.Transformation
24434	TRAFO_GEOAX_ASSIGN_TAB_5	Zuordnung Geo.-Achsen bei 5. Transf.
24440	TRAFO_TYPE_6	Definition der 6.Transformation im Kanal
24442	TRAFO_AXES_IN_6	Achszuordnung für die 6.Transformation
24444	TRAFO_GEOAX_ASSIGN_TAB_6	Zuordnung Geo.-Achsen bei 6. Transf.
24450	TRAFO_TYPE_7	Definition der 7.Transformation im Kanal
24452	TRAFO_AXES_IN_7	Achszuordnung für die 7.Transformation
24454	TRAFO_GEOAX_ASSIGN_TAB_7	Zuordnung Geo.-Achsen bei 7. Transf.
24460	TRAFO_TYPE_8	Definition der 8.Transformation im Kanal
24462	TRAFO_AXES_IN_8	Achszuordnung für die 8.Transformation
24464	TRAFO_GEOAX_ASSIGN_TAB_8	Zuordnung Geo.-Achsen bei 8. Transf.
24900	TRANSMIT_ROT_AX_OFFSET_1	Abweichung der Rundachse von Nullstellung in Grad (1. TRANSMIT)

Nummer	Bezeichner: $MC_	Beschreibung
24910	TRANSMIT_ROT_SIGN_IS_PLUS_1	Vorzeichen der Rundachse bei TRANSMIT (1. TRANSMIT)
24911	TRANSMIT_POLE_SIDE_FIX_1	Einschränkung des Arbeitsbereiches vor/hinter dem Pol, 1. Transformation
24920	TRANSMIT_BASE_TOOL_1	Abstand des Werkzeugnullpunktes vom Ursprung d. Geo. Achsen (1. TRANSMIT)
24950	TRANSMIT_ROT_AX_OFFSET_2	Abweichung der Rundachse von Nullstellung in Grad (2. TRANSMIT)
24960	TRANSMIT_ROT_SIGN_IS_PLUS_2	Vorzeichen der Rundachse bei TRANSMIT (2. TRANSMIT)
24961	TRANSMIT_POLE_SIDE_FIX_2	Einschränkung des Arbeitsbereiches vor/hinter dem Pol, 2. Transformation
24970	TRANSMIT_BASE_TOOL_2	Abstand des Werkzeugnullpunktes vom Ursprung d. Geo. Achsen (2. TRANSMIT)

7.11.1.2 TRACYL

Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
20110	RESET_MODE_MASK	Festlegung der Steuerungsgrundstellung nach Hochlauf und RESET/Teileprogrammende
20140	TRAFO_RESET_VALUE	Grundstellung Transformation
20144	TRAFO_MODE_MASK	Funktionsanwahl der kinematischen Transformation
24100	TRAFO_TYPE_1	Definition der 1.Transformation im Kanal
24110	TRAFO_AXES_IN_1	Achszuordnung für die 1. Transformation
24120	TRAFO_GEOAX_ASSIGN_TAB_1	Zuordnung Geo.-Achsen bei 1. Transf.
24130	TRAFO_INCLUDES_TOOL_1	Werkzeugbehandlung bei aktiver Transformation 1.
24200	TRAFO_TYPE_2	Definition der 2. Transformation im Kanal
24210	TRAFO_AXES_IN_2	Achszuordnung für die 2.Transformation
24220	TRAFO_GEOAX_ASSIGN_TAB_2	Zuordnung Geo.-Achsen bei 2. Transf.
24230	TRAFO_INCLUDES_TOOL_2	Werkzeugbehandlung bei aktiver Transformation 2.
24300	TRAFO_TYPE_3	Definition der 3.Transformation im Kanal
24310	TRAFO_AXES_IN_3	Achszuordnung für die 3.Transformation
24320	TRAFO_GEOAX_ASSIGN_TAB_3	Zuordnung Geo.-Achsen bei 3. Transf.
24330	TRAFO_INCLUDES_TOOL_3	Werkzeugbehandlung bei aktiver Transformation 3.
24400	TRAFO_TYPE_4	Definition der 4.Transformation im Kanal
24410	TRAFO_AXES_IN_4	Achszuordnung für die 4.Transformation
24420	TRAFO_GEOAX_ASSIGN_TAB_4	Zuordnung Geo.-Achsen bei 4. Transf.
24426	TRAFO_INCLUDES_TOOL_4	Werkzeugbehandlung bei aktiver Transformation 4.
24430	TRAFO_TYPE_5	Definition der 5.Transformation im Kanal
24432	TRAFO_AXES_IN_5	Achszuordnung für die 5.Transformation
24434	TRAFO_GEOAX_ASSIGN_TAB_5	Zuordnung Geo.-Achsen bei 5. Transf.

Nummer	Bezeichner: $MC_	Beschreibung
24436	TRAFO_INCLUDES_TOOL_5	Werkzeugbehandlung bei aktiver Transformation 5.
24440	TRAFO_TYPE_6	Definition der 6.Transformation im Kanal
24442	TRAFO_AXES_IN_6	Achszuordnung für die 6.Transformation
24444	TRAFO_GEOAX_ASSIGN_TAB_6	Zuordnung Geo.-Achsen bei 6. Transf.
24446	TRAFO_INCLUDES_TOOL_6	Werkzeugbehandlung bei aktiver Transformation 6.
24450	TRAFO_TYPE_7	Definition der 7.Transformation im Kanal
24452	TRAFO_AXES_IN_7	Achszuordnung für die 7.Transformation
24454	TRAFO_GEOAX_ASSIGN_TAB_7	Zuordnung Geo.-Achsen bei 7. Transf.
24456	TRAFO_INCLUDES_TOOL_7	Werkzeugbehandlung bei aktiver Transformation 7.
24460	TRAFO_TYPE_8	Definition der 8.Transformation im Kanal
24462	TRAFO_AXES_IN_8	Achszuordnung für die 8.Transformation
24464	TRAFO_GEOAX_ASSIGN_TAB_8	Zuordnung Geo.-Achsen bei 8. Transf.
24466	TRAFO_INCLUDES_TOOL_8	Werkzeugbehandlung bei aktiver Transformation 8.
24470	TRAFO_TYPE_9	Definition der 9.Transformation im Kanal
24472	TRAFO_AXES_IN_9	Achszuordnung für die 9.Transformation
24474	TRAFO_GEOAX_ASSIGN_TAB_9	Zuordnung Geo.-Achsen bei 9. Transf.
24476	TRAFO_INCLUDES_TOOL_9	Werkzeugbehandlung bei aktiver Transformation 9.
24480	TRAFO_TYPE_10	Definition d. 10.Transformation im Kanal
24482	TRAFO_AXES_IN_10	Achszuordnung für d. 10.Transformation
24484	TRAFO_GEOAX_ASSIGN_TAB_10	Zuordnung Geo.-Achsen bei 10. Transf.
24486	TRAFO_INCLUDES_TOOL_10	Werkzeugbehandlung bei aktiver Transformation 10.
24800	TRACYL_ROT_AX_OFFSET_1	Abweichung der Rundachse von Nullstellung in Grad (1.TRACYL)
24808	TRACYL_DEFAULT_MODE_1	Auswahl des TRACYL-Modes (1.TRACYL)
24810	TRACYL_ROT_SIGN_IS_PLUS_1	Vorzeichen der Rundachse bei TRACYL (1.TRACYL)
24820	TRACYL_BASE_TOOL_1	Abstand des Werkzeugnullpunktes vom Ursprung d. Geo. Achsen (1. TRACYL)
24850	TRACYL_ROT_AX_OFFSET_2	Abweichung der Rundachse von Nullstellung in Grad (2. TRACYL)
24858	TRACYL_DEFAULT_MODE_2	Auswahl des TRACYL-Modes (2.TRACYL)
24860	TRACYL_ROT_SIGN_IS_PLUS_2	Vorzeichen der Rundachse bei TRACYL (2.TRACYL)
24870	TRACYL_BASE_TOOL_2	Abstand des Werkzeugnullpunktes vom Ursprung d. Geo. Achsen (2. TRACYL)
22534	TRAFO_CHANGE_M_CODE	M-Code bei Transformationsumschaltung der Geometrieachsen

7.11.1.3 TRAANG

Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
20110	RESET_MODE_MASK	Festlegung der Steuerungsgrundstellung nach Hochlauf und RESET/Teileprogrammende
20140	TRAFO_RESET_VALUE	Grundstellung Transformation
20144	RAFO_MODE_MASK	Funktionsanwahl der kinematischen Transformation
20534	TRAFO_CHANGE_M_CODE	M-Code bei Transformationsumschaltung der Geometrieachsen
24100	TRAFO_TYPE_1	Definition der 1.Transformation im Kanal
24110	TRAFO_AXES_IN_1	Achszuordnung für die 1. Transformation
24120	TRAFO_GEOAX_ASSIGN_TAB_1	Zuordnung Geo.-Achsen bei 1. Transf.
24200	TRAFO_TYPE_2	Definition der 2. Transformation im Kanal
24210	TRAFO_AXES_IN_2	Achszuordnung für die 2.Transformation
24220	TRAFO_GEOAX_ASSIGN_TAB_2	Zuordnung Geo.-Achsen bei 2. Transf.
24300	TRAFO_TYPE_3	Definition der 3.Transformation im Kanal
24310	TRAFO_AXES_IN_3	Achszuordnung für die 3.Transformation
24320	TRAFO_GEOAX_ASSIGN_TAB_3	Zuordnung Geo.-Achsen bei 3. Transf.
24400	TRAFO_TYPE_4	Definition der 4.Transformation im Kanal
24410	TRAFO_AXES_IN_4	Achszuordnung für die 4.Transformation
24420	TRAFO_GEOAX_ASSIGN_TAB_4	Zuordnung Geo.-Achsen bei 4. Transf.
24430	TRAFO_TYPE_5	Definition der 5.Transformation im Kanal
24432	TRAFO_AXES_IN_5	Achszuordnung für die 5.Transformation
24434	TRAFO_GEOAX_ASSIGN_TAB_5	Zuordnung Geo.-Achsen bei 5. Transf.
24440	TRAFO_TYPE_6	Definition der 6.Transformation im Kanal
24442	TRAFO_AXES_IN_6	Achszuordnung für die 6.Transformation
24444	TRAFO_GEOAX_ASSIGN_TAB_6	Zuordnung Geo.-Achsen bei 6. Transf.
24450	TRAFO_TYPE_7	Definition der 7.Transformation im Kanal
24452	TRAFO_AXES_IN_7	Achszuordnung für die 7.Transformation
24454	TRAFO_GEOAX_ASSIGN_TAB_7	Zuordnung Geo.-Achsen bei 7. Transf.
24460	TRAFO_TYPE_8	Definition der 8.Transformation im Kanal
24462	TRAFO_AXES_IN_8	Achszuordnung für die 8.Transformation
24464	TRAFO_GEOAX_ASSIGN_TAB_8	Zuordnung Geo.-Achsen bei 8. Transf.
24700	TRAANG_ANGLE_1	Winkel der Schrägen Achse in Grad (1. TRAANG)
24710	TRAANG_BASE_TOOL_1	Abstand des Werkzeugnullpunktes vom Ursprung der Geometrieachsen (1. TRAANG)
24720	TRAANG_PARALLEL_VELO_RES_1	Geschwindigkeitsreserve der parallelen Achse für die Ausgleichsbewegung (1. TRAANG)
24721	TRAANG_PARALLEL_VELO_RES_2	Geschwindigkeitsreserve der parallelen Achse für die Ausgleichsbewegung (2. TRAANG)
24750	TRAANG_ANGLE_2	Winkel der Schrägen Achse in Grad (2. TRAANG)

Nummer	Bezeichner: $MC_	Beschreibung
24760	TRAANG_BASE_TOOL_2	Abstand des Werkzeugnullpunktes vom Ursprung der Geometrieachsen (2. TRAANG)
24770	TRAANG_PARALLEL_ACCEL_RES_1	Achsbeschleunigungsreserve der parallelen Achse für die Ausgleichsbewegung (1. TRAANG)
24771	TRAANG_PARALLEL_ACCEL_RES_2	Achsbeschleunigungsreserve der parallelen Achse für die Ausgleichsbewegung (2. TRAANG)

7.11.1.4 Verkettete Transformationen

Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
24995	TRACON_CHAIN_1	Transformationskette der ersten verketteten Transformation
24996	TRACON_CHAIN_2	Transformationskette der zweiten verketteten Transformation
24997	TRACON_CHAIN_3	Transformationskette der dritten verketteten Transformation
24998	TRACON_CHAIN_4	Transformationskette der vierten verketteten Transformation

7.11.1.5 Nicht-transformationsspezifische Maschinendaten

Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
21110	X_AXIS_IN_OLD_X_Z_PLANE	Koordinatensystem bei automatischer Framedefinition
21090	MAX_LEAD_ANGLE	Maximal zulässiger Voreilwinkel bei Orientierungsprogrammierung
21092	MAX_TILT_ANGLE	Maximal zulässiger Seitwärtswinkel bei Orientierungsprogrammierung
21100	ORIENTATION_IS_EULER	Winkeldefinition bei Orientierungsprogrammierung

7.11.2 Signale

7.11.2.1 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Transformation aktiv	DB21, ...DBX33.6	DB3300.DBX1.6

M5: Messen

8

8.1 Kurzbeschreibung

Kanalspezifisches Messen

Beim kanalspezifischen Messen wird einem Teileprogrammsatz ein Triggerereignis programmiert, das den Messvorgang auslöst und ein Messmodus festgelegt, in dem die Messung erfolgt. Die Anweisungen gelten für alle in diesem Satz programmierten Achsen.

Istwertsetzen und Ankratzen

Das **Istwertsetzen** erfolgt über die HMI-Bedienung. Das berechnete Frame kann in den Systemframe $P_SETFRAME geschrieben werden. Beim Istwertsetzen kann die Sollposition einer Achse im WKS geändert werden.

Die Berechnung erfolgt in der NC durch Aktivierung eines PI-Dienstes über die

- HMI-Bedienung oder über einen
- Teileprogrammbefehl aus den Messzyklen.

Unter dem Begriff **Ankratzen** werden die Werkstückvermessung **und** die Werkzeugvermessung verstanden. Die Messungen können über die

- HMI-Bedienung oder über
- Messzyklen erfolgen.

Zur Kommunikation mit der NC dienen vordefinierte Systemvariablen.

Werkstück-/Werkzeugvermessung

Bei der Werkstückvermessung kann die Lage des Werkstückes bezüglich einer Kante, einer Ecke oder einer Bohrung vermessen werden.

Zur Festlegung der Nulllage des Werkstückes (Werkstücknullpunkt W) oder einer Bohrung können die gemessenen Positionen mit Sollpositionen im WKS beaufschlagt werden. Die resultierenden Verschiebungen können dabei in einen ausgewählten Frame eingetragen werden.

Bei der Werkzeugvermessung berechnet die Steuerung aus der einzugebenen Werkzeuglänge den Abstand der Werkzeugspitze vom Werkzeugträgerbezugspunkt T.

Messzyklen

Eine Beschreibung zur Handhabung der Messzyklen ist zu finden in:
Literatur:
/BNM/ Programmierhandbuch Messzyklen

8.2 Hardwarevoraussetzungen

8.2.1 Verwendbare Messtaster

Allgemeines

Zur Erfassung von Werkzeug- und Werkstückabmessungen wird ein schaltender Messtaster benötigt, der bei Auslenkung ein konstantes Signal (keinen Impuls) liefert.

Der Messtaster muss nahezu prellfrei schalten. Dies ist im Allgemeinen durch eine mechanische Justierung des Tasters möglich.

Auf dem Markt werden von verschiedenen Herstellern unterschiedliche Ausführungen von Messtastern angeboten. Die Messtaster werden daher nach der Anzahl der Richtungen, in die ein Messtaster ausgelenkt werden kann, in drei Gruppen unterteilt (siehe folgendes Bild).

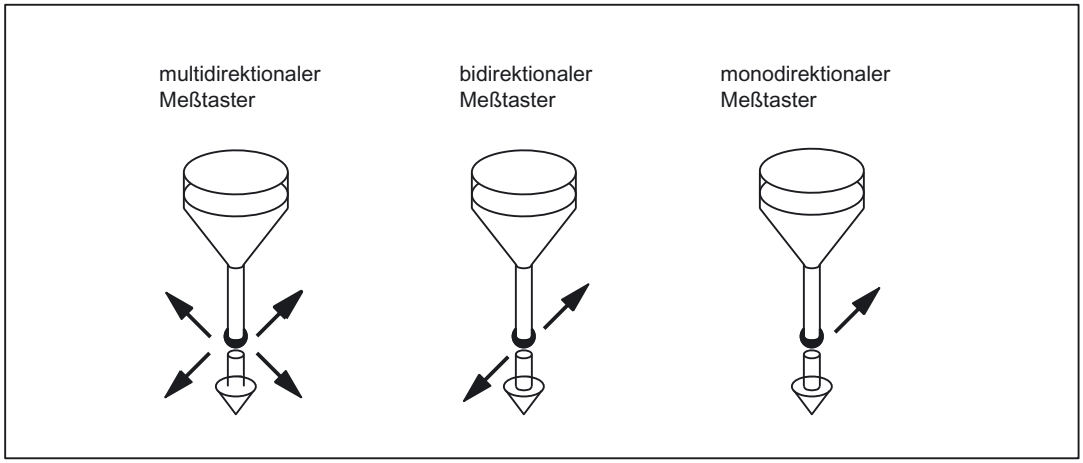


Bild 8-1 Messtastertypen

Tabelle 8- 1 Messtastertypzuordnung

Messtastertyp	Drehmaschinen		Fräs- und Bearbeitungszentren
	Werkzeugmessung	Werkstückmessung	Werkstückmessung
multidirektionaler	X	X	X
bidirektionaler	_	X	X
monodirektionaler	_	_	X

Während bei Drehmaschinen ein bidirektionaler Messtaster einsetzbar ist, kann bei Fräs- und Bearbeitungszentren auch ein Monotaster für die Werkstückmessung verwendet werden.

Multidirektionaler Messtaster (3D)

Dieser Typ kann zur Werkzeug- und Werkstückmessung ohne Einschränkung benutzt werden.

Bidirektionaler Messtaster

Bei der Werkstückmessung in Fräs- und Bearbeitungszentren wird dieser Typ wie ein Monotaster behandelt. Bei Drehmaschinen kann dieser Typ für die Werkstückmessung verwendet werden.

Monodirektionaler Messtaster

An Fräs- und Bearbeitungszentren kann dieser Typ zur Werkstückmessung mit geringen Einschränkungen benutzt werden.

Spindelposition beim Monotaster

Um diesen Typ bei Fräs- und Bearbeitungszentren einsetzen zu können, muss die Spindel mit der NC-Funktion `SPOS` positionierbar und das Schaltsignal des Tasters über 360° an die Empfangsstation (am Ständer der Maschine) übertragbar sein.

In der Spindel muss der Messtaster so mechanisch ausgerichtet werden, dass bei der Spindelposition von 0 Grad in folgende Richtungen gemessen werden kann.

Tabelle 8- 2 Spindelpositionen für Ausrichtung des Messtasters

	Messung bei 0 Grad Spindelposition
X-Y-Ebene G17	positive X-Richtung
Z-X-Ebene G18	positive Z-Richtung
Y-Z-Ebene G19	positive Y-Richtung

Mit Monotaster dauert die Messung länger, weil die Spindel mehrmals im Messzyklus mit `SPOS` positioniert werden muss.

8.2.2 Messtasteranschluss

Anschluss an SINUMERIK 840D

Der Anschluss des Messtasters an die SINUMERIK 840D erfolgt über die Peripherieschnittstelle X121, die sich auf der Frontplatte des NCU-Moduls befindet.

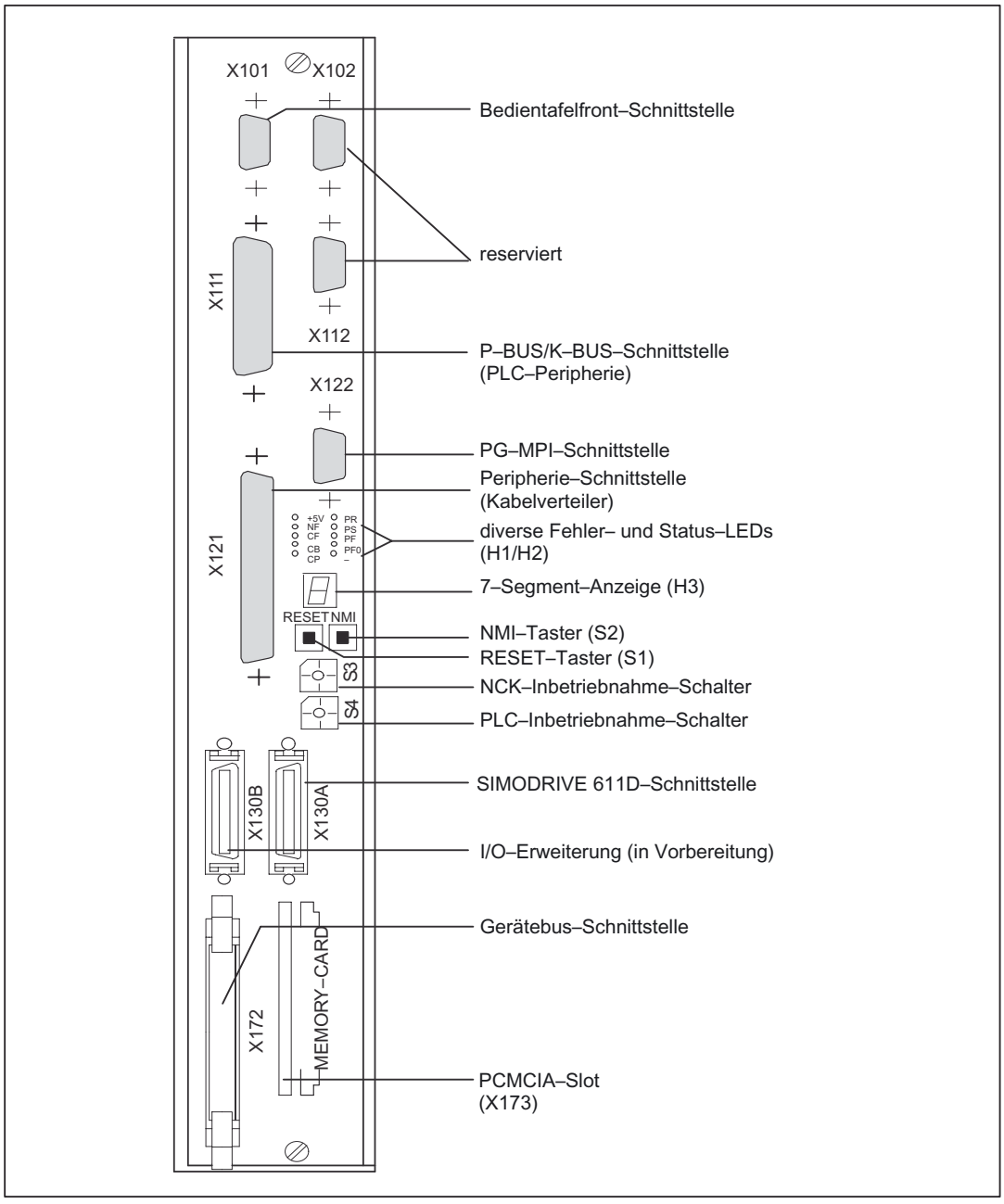


Bild 8-2 Schnittstellen, Bedien- und Anzeigeelemente des NCU-Moduls

Anschluss an SINUMERIK 840D sl

Der Anschluss des Messtasters an die SINUMERIK 840D sl erfolgt über die Peripherieschnittstelle X121, die sich auf der oberen Front des NCU–Moduls befindet.

Für die digitalen Ein–und /Ausgänge dieser Schnittstelle sind verschiedene herstellerspezifische Telegrammtypen parametrierbar.

SINUMERIK 840D sl verwendet vorzugsweise folgende Telegrammtypen für

- ohne Messtaster
- bis zu zwei Messtaster
- bis zu sechs Messtaster

Es bestehen zwei Varianten zur Klemmenbelegung X122, die jeweils für zwei Messtaster vorbelegt sind und bei beiden Varianten eine identische Pinbelegung aufweist. Für beide Messtastersignale der konfigurierbaren Achsen ist ein Konfigurationsmakro vorgeschaltet. Damit ist eine zusätzliche Parametrierung der Messtastersignale vom Anwender nicht mehr notwendig.

Literatur

Inbetriebnahmehandbuch CNC Teil1 (NCK, PLC, Antrieb); Kapitel "Grundlagen".
/PHD/ Gerätehandbuch Projektierung NCU

Anschluss an SINUMERIK 840Di

Der Anschluss des Messtasters an die SINUMERIK 840Di erfolgt über die Peripherieschnittstelle X121 der MCI-Board-Extension Baugruppe (Option).

Literatur:
/HBi/ Handbuch Projektierung SINUMERIK 840Di

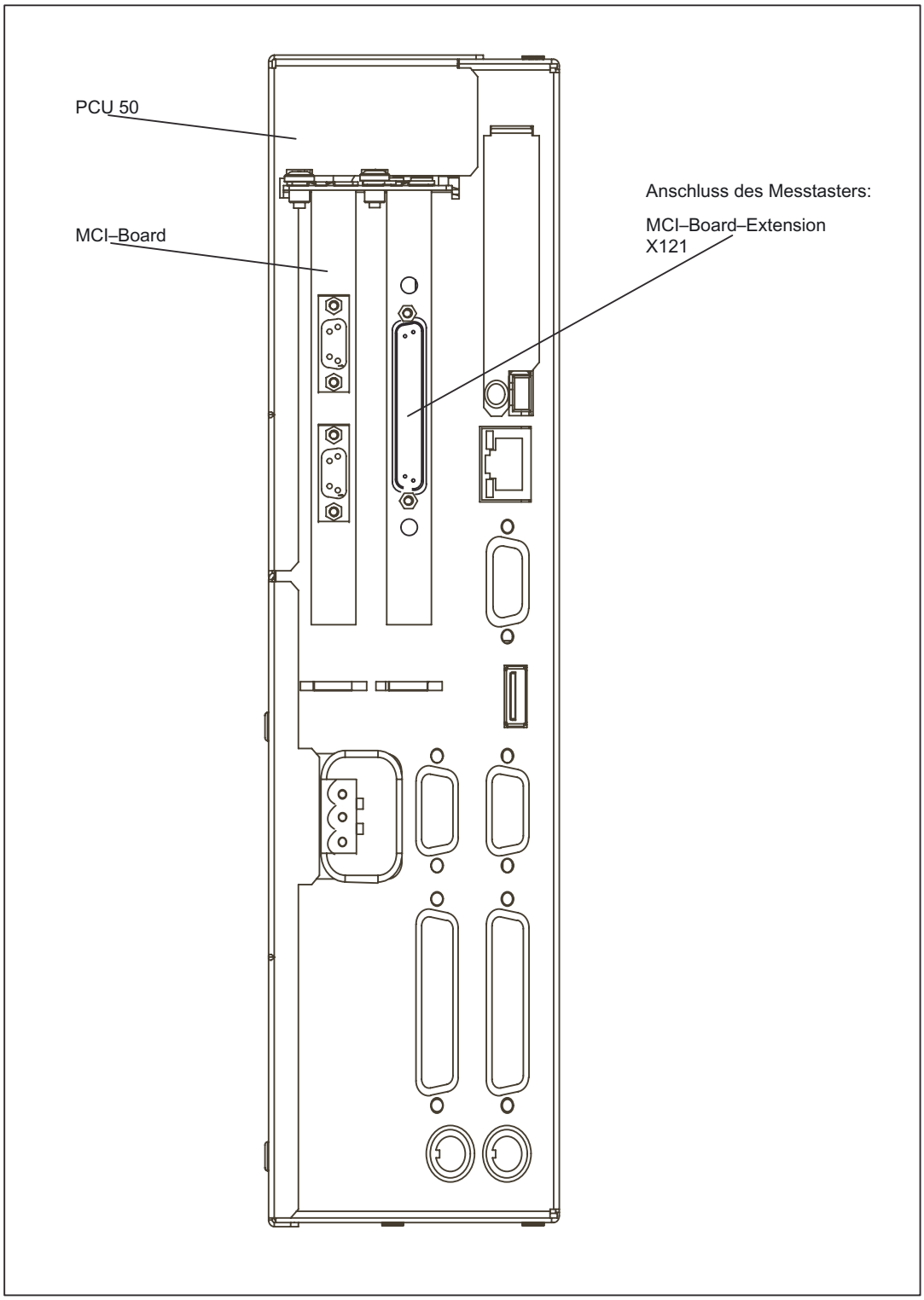


Bild 8-3 Schnittstellen der SINUMERIK 840Di (PCU 50, MCI-Board und MCI-Board-Extension)

Peripherieschnittstelle X121

Die Anschaltung eines Messtasters erfolgt über die

- **Peripherie-Schnittstelle**

37-polige D-Sub Stiftleiste (X121), es können **maximal 2** Messtaster angeschlossen werden;

Der Anschluss der 24V-Laststromversorgung befindet sich ebenfalls auf diesem Stecker.

Tabelle 8- 3 Auszug aus der PIN-Belegungstabelle für Frontstecker X121

	PIN		Bezeichnung
X121			Externe Stromversorgung
	1	M24EXT	Masse extern
	2	M24EXT	Masse extern
	...	...	...
			Anschluss Messtaster 1
	9	MEPUS 0	Messpuls-Signal Input
	10	MEPUC 0	Messpuls-Commen Input
	...	...	...
			Externe Stromversorgung
	20	P24EXT	P 24 V extern
	21	P24EXT	P 24 V extern
	...	...	...
			Anschluss Messtaster 2
	28	MEPUS 1	Messpuls-Signal Input
	29	MEPUC 1	Messpuls-Commen Input
	...	...	...

Die Schnittstellen mit Pinbelegung sind ausführlich dargestellt und beschrieben:
Literatur:
/PHC/ Gerätehandbuch Projektierung CCU

PROFIBUS-DP Antriebe

Es besteht für die SINUMERK 840D mit einer NCU 573.2/3/4 die Möglichkeit einen Messtaster dezentral direkt am PROFIBUS-DP Antrieb zu betreiben. Diese Methode ist genauer als über zentrale Messtaster zyklische Positionswerte mit der NC interpolatorisch zu ermitteln.

Die Art der Messfunktion für PROFIBUS-DP Antriebe z. B. mit SIMODRIVE 611 universal wird mit dem folgenden Maschinendatum festgelegt:

MD13210 $MN_MEAS_TYPE

Wert = 0: ein zentral an der NC angeschlossener Messtaster oder

Wert = 1: ein dezentral verdrahteter Messtaster für **alle** Antriebe.

SIMODRIVE 611 universal Antriebe unterstützen die Messfunktionalität dezentraler Messtaster, indem der tatsächliche Geber- Istwert zum Zeitpunkt der Messflanke in der Hardware abgespeichert wird. Für PROFIBUS-DP-Antriebe ist die genauere Messmethode eines dezentralen Messtasters vorzuziehen.

Die Antriebe SIMODRIVE 611 digital werden unverändert mit zentralem Messtaster über die Steckleiste X121 an der SINUMERIK 840D/840Di betrieben.

Literatur:
/BHA/ Benutzerhandbuch Absolutwertgeber mit PROFIBUS-DP
/FBU/ Funktionshandbuch SIMODRIVE 611 universal

8.3 Kanalspezifisches Messen

8.3.1 Messmodus

Messbefehle MEAS und MEAW

Die Aktivierung des Messvorganges erfolgt aus dem Teileprogramm. Es werden ein Triggerereignis und ein Messmodus programmiert.

Es wird zwischen zwei Messmodi unterschieden:

- MEAS: Messen mit Restweglöschen

  Beispiel:

  N10 G01 F300 X300 Z200 MEAS=-2

  Triggerereignis ist die fallende Flanke (-) des zweiten Messtasters (2).

- MEAW: Messen ohne Restweglöschen

  Beispiel:

  N20 G01 F300 X300 Y100 MEAW=1

  Triggerereignis ist die steigende Flanke des ersten Messtasters (1).

Der Messauftrag wird mit `RESET` bzw. beim Einwechseln eines neuen Satzes abgebrochen.

Hinweis

Ist in einem Messsatz eine GEO-Achse programmiert, werden die Messwerte für alle aktuellen GEO-Achsen abgelegt.

Ist in einem Messsatz eine an einer Transformation beteiligte Achse programmiert, werden die Messwerte aller an dieser Transformation beteiligten Achsen abgelegt.

Messtasterstatus

Es kann der Messtasterstatus direkt im Teileprogramm und in Synchronaktionen abgefragt werden.

$A_PROBE[n] mit n=Messtaster

$A_PROBE[n]==1: Messtaster ausgelenkt

$A_PROBE[n]==0: Messtaster nicht ausgelenkt

8.3.2 Messergebnisse

Messergebnisse lesen im TP

Die Ergebnisse des Messbefehls werden in Systemdaten des NCK hinterlegt und sind im Teileprogramm über Systemvariablen lesbar.

- **Systemvariable $AC_MEA[Nr]**

  Statussignal des Messauftrages abfragen.

  [Nr] steht für Messtaster (1 oder 2)

  Die Variable wird zu Beginn einer Messung gelöscht. Sobald der Messtaster das Auslösekriterium erreicht (steigende oder fallende Flanke), wird die Variable gesetzt. Damit kann im Teileprogramm die Durchführung des Messauftrages kontrolliert werden.

- **Systemvariable $AA_MM[Achse]**

  Zugriff auf das Messergebnis im Maschinenkoordinatensystem.

  Lesen im Teileprogramm und in den Synchronaktionen.

  [Achse] steht für den Name der Messachse (X, Y, ...).

- **Systemvariable $AA_MW[Achse]**

  Zugriff auf das Messergebnis im Werkstückkoordinatensystem.

  Lesen im Teileprogramm und in den Synchronaktionen.

  [Achse] steht für den Namen der Messachse (X, Y, ...).

Literatur:
/PGZ/ Programmierhandbuch Zyklen

PLC-Service-Anzeige

Die Funktionsprüfung des Messtasters erfolgt über ein NC-Programm.

Über das Diagnose-Menü "PLC-Status" kann das Messsignal nach Programmende kontrolliert werden.

Tabelle 8- 4 Statusanzeige für Messsignal

	Statusanzeige
Messfühler 1 ausgelenkt	DB10 DB B107.0
Messfühler 2 ausgelenkt	DB10 DB B107.1

Mit dem Nahtstellensignal DB31, ... DBX62.3 wird der augenblickliche Messstatus der Achse angezeigt:

Bit 3=1: Messen aktiv
Bit 3=0: Messen nicht aktiv

Diese Signal kann bei allen Messfunktionen angezeigt und auch in Synchronaktionen mit

- **Systemvariable $AA_MEAACT[Achse]** gelesen werden.

Literatur:

FBSY/ Funktionshandbuch Synchronaktionen

8.4 Setzen von Nullpunkten, Werkstückvermessung und Werkzeugvermessung

8.4.1 Istwertsetzen und Ankratzen

Istwertsetzen

Das **Istwertsetzen** erfolgt über die HMI-Bedienung oder über Messzyklen. Der berechnete Frame kann in den Systemframe $P_SETFRAME geschrieben werden. Beim Istwertsetzen kann die Sollposition einer Achse im WKS geändert werden.

Die Berechnung erfolgt in der NC durch Aktivierung eines PI-Dienstes über die

- HMI-Bedienung oder über einen
- Teileprogrammbefehl aus den Messzyklen.

Als Grundlage für die Berechnung kann ein Werkzeug und eine Ebene ausgewählt werden. Der berechnete Frame wird in den Ergebnis-Frame eingetragen.

Ankratzen

Unter dem Begriff **Ankratzen** werden die Werkstück- **und** die Werkzeugvermessung verstanden. 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

- HMI-Bedienung oder über
- Messzyklen erfolgen.

Zur Kommunikation mit der NC dienen vordefinierte Systemvariablen. Die Berechnung erfolgt im NCK durch Aktivierung eines PI-Dienstes über:

- die HMI-Bedienung
- oder über einen Teileprogrammbefehl 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.

Weitere Erläuterungen zu den kanalspezifischen Systemframes entnehmen Sie bitte:

/PGA1/ Listen der Systemvariablen; Kapitel "Frames".

/FB1/ Funktionshandbuch Grundfunktionen; Achsen, Koordinatensysteme, Frames (K2), Kapitel "Frames der Framekette"

Weitere Literatur:

/BAD/BEM/BEMsl Bedienhandbücher HMI-Advanced/Embedded; Kapitel "Ankratzen".

/PGZ/ Programmierhandbuch Zyklen; Kapitel "Schwenken - CYCLE800".

8.4.2 Werkstückvermessung

Vermessung des Werkstückes

Für die Werkstückvermessung wird ein Messtaster wie ein Werkzeug an das aufgespannte Werkstück herangefahren. Durch eine vielfältige Auswahl an unterschiedlichen Messtypen können die gebräuchlichsten Messaufgaben an einer Dreh- oder Fräsmaschine möglichst einfach gelöst werden.

Bei der Werkstückvermessung kann die Lage des Werkstückes bezüglich einer Kante, einer Ecke oder einer Bohrung vermessen werden.

Zur Festlegung der Nulllage des Werkstückes (Werkstücknullpunkt W) oder einer Bohrung können die gemessenen Positionen mit Sollpositionen im WKS beaufschlagt werden. Die resultierenden Verschiebungen können dabei in ein ausgewähltes Frame eingetragen werden.

Variablenschnittstelle

Die Variablenschnittstelle besteht aus mehreren Systemvariablen.

Diese werden unterschieden zwischen:

- Eingangswerten
- Ausgangswerten

Literatur:
/PGA1/ Listenhandbuch Systemvariablen

Die Eingangswerte müssen von HMI oder von den Zyklen beschrieben werden. Die Ausgangswerte sind die Rechenergebnisse.

Literatur:
/BNM/ Programmierhandbuch Messzyklen

8.4.2.1 Eingangswerte

Gültigkeitsbits der Messtypen

Zur Festlegung, welche Systemvariablen für die aktuelle Messung gültig sind, sollte jeder Messvorgang zuerst alle Variablen als ungültig deklarieren. Dies erfolgt mit:
$AC_MEAS_VALID = 0.

Jede Eingangs-Variable setzt implizit beim Beschreiben das entsprechende Bit in $AC_MEAS_VALID. Werden die Gültigkeitsbits nicht zurückgesetzt, so bleiben die Werte auch für die nächste Berechnung gültig.

Hinweis

Bei Maschinensteuertafel-Reset oder nach M30 (Reset bei Programmende) wird die Schnittstelle nicht zurückgesetzt.

Tabelle 8- 5 Gültigkeitsbits für die Eingangswerte der Variablen $AC_MEAS_VALID

Bit	Eingangswert	Bedeutung
0	$AA_MEAS_POINT1[Achse]	1. Messpunkt für alle Kanalachsen
1	$AA_MEAS_POINT2[Achse]	2. Messpunkt für alle Kanalachsen
2	$AA_MEAS_POINT3[Achse]	3. Messpunkt für alle Kanalachsen
3	$AA_MEAS_POINT4[Achse]	4. Messpunkt für alle Kanalachsen
4	$AA_MEAS_SETPOINT[Achse]	Sollposition der Kante, Ecke, Bohrung
5	$AC_MEAS_WP_SETANGLE	Soll-Werkstücklage-Winkel α; -90 < φ < 180
6	$AC_MEAS_CORNER_SETANGLE	Soll-Schnittwinkel φ der Ecke 0 < φ < 180
7	$AC_MEAS_T_NUMBER	ausgewähltes Werkzeug
7	$AC_MEAS_D_NUMBER	ausgewählte Schneide
9	$AC_MEAS_DIR_APPROCH	Anfahrrichtung nur bei Kanten-, Nut,- Steg- und Werkzeugvermessung
10	$AC_MEAS_ACT_PLANE	Arbeitsebene und Zustellrichtung einstellen
11	$AC_MEAS_FRAME_SELECT	Berechneter Frame in den spezifizierten Frame
12	$AC_MEAS_TYPE	Messtypen der Werkstückvermessung
13	$AC_MEAS_FINE_TRANS	Translatorische Verschiebungen eintragen
14	$AA_MEAS_SETANGEL[Achse]	Soll-Winkel einer Achse
15	$AA_MEAS_SCALEUNIT	Maßeinheit für Ein- und Ausgangswerte
16	$AA_MEAS_TOOL_MASK	Werkzeugeinstellungen
17	$AA_MEAS_P1_COORD	Koordinatensystem des 1. Messpunktes
18	$AA_MEAS_P2_COORD	Koordinatensystem des 2. Messpunktes
19	$AA_MEAS_P3_COORD	Koordinatensystem des 3. Messpunktes
20	$AA_MEAS_P4_COORD	Koordinatensystem des 4. Messpunktes
21	$AA_MEAS_SET_COORD	Koordinatensystem des Sollpunktes
22	$AA_MEAS_CHSFR	Systemframemaske
23	$AA_MEAS_NCBFR	Maske für globale Basisframes
24	$AA_MEAS_CHBFR	Maske für Kanal-Basisframes
25	$AA_MEAS_UIFR	Einstellbares Frame aus der Datenhaltung
26	$AA_MEAS_PFRAME	Programmierbare Frames nicht verrechnen
27	$AC_MEAS_INPUT[n]	Mess-Eingangsparameter mit der Länge n

Hinweis

Alle Achs-Istwerte des entsprechenden Messpunktes werden ungültig durch:

$AC_MEAS_LATCH = 0

Messpunkte

Für die Messungen stehen max. vier Messpunkte für alle Kanalachsen zur Verfügung:

Typ	Eingangs-Variable	Bedeutung
REAL	$AA_MEAS_POINT1[Achse]	1. Messpunkt für alle Kanalachsen
REAL	$AA_MEAS_POINT2[Achse]	2. Messpunkt für alle Kanalachsen
REAL	$AA_MEAS_POINT3[Achse]	3. Messpunkt für alle Kanalachsen
REAL	$AA_MEAS_POINT4[Achse]	4. Messpunkt für alle Kanalachsen

Die gemessenen Punkte liegen im Normalfall als Istwerte (= Sollwerte) im WKS vor. Ein Messpunkt wird als gültig gekennzeichnet, sobald ein Achswert davon beschrieben wurde. Jeder der einzelnen Messpunkte kann beschrieben oder abgelatched werden.

Einige Messtypen unterstützen auch Messpunkte, die in einem anderen Koordinatensystem (BKS, MKS) vorliegen. Die Eingabe, in welchem Koordinatensystem der entsprechende Messpunkt gemessen wurde, kann über folgende Variablen erfolgen:

Typ	Eingangs-Variable	Bedeutung	Werte
INT	$AA_MEAS_P1_COORD	Koordinatensystem des 1. Messpunktes	0: WKS ist Standardeinstellung 1: BKS 2: MKS 3: ENS 4: WKS_REL 5: ENS_REL
INT	$AA_MEAS_P2_COORD	Koordinatensystem des 2. Messpunktes	
INT	$AA_MEAS_P3_COORD	Koordinatensystem des 3. Messpunktes	
INT	$AA_MEAS_P4_COORD	Koordinatensystem des 4. Messpunktes	
INT	$AA_MEAS_SET_COORD	Koordinatensystem des Sollpunktes	

Istwerte

Die Messpunkte können für alle Achsen mit den aktuellen Achs-Istwerten beschrieben werden. Die Positionen werden bezüglich dem ausgewähltem Koordinatensystem abgelatched. Wird kein Koordinatensystem vorgegeben, so werden die Positionen im WKS gelatched. Dazu dient die folgende Variable:
$AC_MEAS_LATCH[0..3]

Der Index variert von 0 bis 3, entsprechend dem 1. bis 4. Messpunkt. Die Zuweisung des Wertes 0 an die Variable bewirkt, dass alle Achsistwerte des entsprechenden Messpunktes ungültig werden. Bei Zuweisung des Wertes 1, werden alle Achsistwerte im entsprechenden Messpunkt abgelatched. Die Variable kann nur geschrieben werden.

Einzelne Achsistwerte eines Messpunktes können mit folgenden Variablen beschrieben werden:

Typ	Systemvariable	Bedeutung	Werte
REAL	$AA_MEAS_P1_VALID[ax]	1. Messpunkt einer Achse ablatchen	0: Messpunkt der Achse ist ungültig 1: Messpunkt der Achse wird ermittelt
REAL	$AA_MEAS_P2_VALID[ax]	2. Messpunkt einer Achse ablatchen	
REAL	$AA_MEAS_P3_VALID[ax]	3. Messpunkt einer Achse ablatchen	
REAL	$AA_MEAS_P4_VALID[ax]	4. Messpunkt einer Achse ablatchen	

Die Variablen $AC_MEAS_LATCH[0..3] und $AA_MEAS_P[1..4]_VALID können interaktiv angewendet werden. Bei Durchmesserprogrammierung wird die Planachse entsprechend berücksichtigt.

Sollwerte

Die Berechnung des resultierenden Frames erfolgt so, dass die vom Anwender vorgegebenen Sollwerte eingehalten werden.

Tabelle 8- 6 Eingangswerte für die Anwender-Sollwerte

Typ	Systemvariable	Bedeutung
REAL	$AA_MEAS_SETPOINT[ax]	Sollposition einer Achse
REAL	$AA_MEAS_SETANGLE[ax]	Sollwinkel einer Achse
INT	$AA_MEAS_SP_VALID[ax]	1: Sollposition der Achse ist gültig/0: ungültig
REAL	$AC_MEAS_WP_SETANGLE	Soll-Werkstücklage-Winkel α: -90 < α < 180
REAL	$AC_MEAS_CORNER_SETANGLE	Soll-Schnittwinkel φ der Ecke: 0 < φ < 180
INT	$AC_MEAS_DIR_APPROACH *)	Anfahrrichtung: 0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z

*) Die Anfahrrichtung wird nur bei der Kanten-, Nut-, Steg- und bei der Werkzeugmessung benötigt.

Folgende Messpunkte sind irrelevant und werden nicht ausgewertet:

- Bei Eingabe des Soll-Werkstücklage-Winkel α: der 2. Messpunkt.
- Bei Eingabe des Soll-Schnittwinkel φ: der 4. Messpunkt.

Ebeneneinstellung

Ebeneneinstellung für die Festlegung der Werkzeugorientierung. Wird keine Ebene vorgegeben, so wird mit der aktiven Ebene gerechnet.

Typ	Systemvariable	Werte
INT	$AC_MEAS_ACT_PLANE	0: G17 Arbeitsebene x/y Zustellrichtung z 1: G18 Arbeitsebene z/x Zustellrichtung y 2: G19 Arbeitsebene y/z Zustellrichtung x

Translatorische Verschiebungen

Beim Vermessen von Werkstücken können translatorische Verschiebungen in den Feinverschiebungsanteil des ausgewählten Frames eingetragen werden. Hierzu dient die Variable $AC_MEAS_FINE_TRANS.

Typ	Systemvariable	Werte
INT	$AC_MEAS_FINE_TRANS	0: Translatorische Korrektur wird in die Grobverschiebung eingetragen 1: Translatorische Korrektur wird in die Feinverschiebung eingetragen

Wird die Variable $AC_MEAS_FINE_TRANS nicht beschrieben, gilt:

- Der Korrekturwert wird in die Grobverschiebung eingetragen und in das Zielframe transformiert. Es kann sich durch die Transformation auch ein Feinanteil in der Translation ergeben.
- Wenn das folgende Maschinendatum nicht mit 1 vorbesetzt ist:

  MD18600 $MN_MM_FRAME_FINE_TRANS

  Es wird immer die Korrektur in die Grobverschiebung eingetragen.

Berechneter Frame

Im Falle der Werkstückvermessung wird der berechnete Frame in den spezifizierten Frame eingetragen.

Typ	Systemvariable	Bedeutung
INT	$AC_MEAS_FRAME_SELECT	Frameauswahl bei der Werkstückvermessung

Die Variable $AC_MEAS_FRAME_SELECT kann folgende Werte annehmen

Wert		Bedeutung
0	$P_SETFRAME	aktiver Systemframe
1	$P_PARTFRAME	aktiver Systemframe
2	$P_EXTFRAME	aktiver Systemframe
10..25	$P_CHBFRAME[0..15]	aktive kanalspezifische Basisframes
50..65	$P_NCBFRAME[0..15]	aktive NCU-globale Basisframes
100..199	$P_IFRAME	Berechnung erfolgt mit aktivem einstellbaren Frame, wenn das entsprechende Frame angewählt ist. Ist das ausgewählte Frame nicht aktiv, so wird das entsprechende Datenhaltungsframe in die Berechnung mit einbezogen.
500	$P_TOOLFRAME	aktiver Systemframe
501	$P_WPFRAME	aktiver Systemframe
502	$P_TRAFRAME	aktiver Systemframe
503	$P_PFRAME	aktiver aktueller programmierbarer Frame
504	$P_CYCFRAME	aktiver Systemframe
505	$P_RELFRAME (PCS)	aktiver Systemframe
506	$P_RELFRAME (ACS)	aktiver Systemframe
1010..1025	$P_CHBFRAME[0..15]	aktive kanalspez. Basisframes mit aktiven G500
1050..1065	$P_NCBFRAME[0..15]	aktive NCU-glob. Basisframes mit aktiven G500
2000	$P_SETFR	Systemframe in der Datenhaltung
2001	$P_PARTFR	Systemframe in der Datenhaltung
2002	$P_EXTFR	Systemframe in der Datenhaltung
2010..2025	$P_CHBFR[0..15]	kanalspezifische Basisframes in der Datenhaltung
2050..2065	$P_NCBFR[0..15]	NCU-globale Basisframes in der Datenhaltung
2100..2199	$P_UIFR[0..99]	einstellbare Frames in der Datenhaltung
2500	$P_TOOLFR	Systemframe in der Datenhaltung

Wert		Bedeutung
2501	$P_WPFR	Systemframe in der Datenhaltung
2502	$P_TRAFR	Systemframe in der Datenhaltung
2504	$P_CYCFR	Systemframe in der Datenhaltung
2505	$P_RELFR (PCS)	Systemframe in der Datenhaltung
2506	$P_RELFR (ACS)	Systemframe in der Datenhaltung
3010..3025	$P_CHBFR[0..15]	kanalspez. Basisframes mit aktiven G500 in der Datenhaltung
3050..3065	$P_NCBFR[0..15]	NCU-globale Basisframes mit aktiven G500 in der Datenhaltung

Die Funktion MEASURE( ) berechnet das Frame $AC_MEAS_FRAME entsprechend dem spezifizierten Frame.

Bei den Werten

von **0 bis 1065** wird die Berechnung mit Hilfe des aktiven Frames durchgeführt.

von **2000 bis 3065** wird die Berechnung bezüglich des ausgewählten Frames in der Datenhaltung durchgeführt. Die Auswahl eines Frames in der Datenhaltung wird nicht bei den Messtypen 14 und 15 unterstützt. Bei der Auswahl eines Frames in der Datenhaltung muss dieser Frame nicht aktiv sein. In diesem Fall wird die Berechnung so durchgeführt, als ob dieser aktiv in der Kette wäre.

Der Messpunkt wird in das ausgewählte System transformiert und der selektierte Frame wird mit Hilfe des Gesamtframes incl. des ausgewählten Frames bestimmt. Nach Korrektur und Aktivierung des Frames wird das Istwertsetzen erst wirksam.

Bei den Werten

mit aktiven **G500** (1010..1025, 1050..1065, 3010..3025, 3050..3065) wird der Zielframe so berechnet, dass nach Anwahl dieses Frame auch G500 aktiv sein muss, um die Sollposition zu erreichen.

Umrechnung in ein anderes Koordinatensystem

Wenn eine Position in eine Position eines anderen Koordinatensystems umgerechnet werden soll, kann die Zusammensetzung der gewünschten Framekette über folgende Variablen vorgegeben werden:

Typ	Systemvariable	Bedeutung	Werte
INT	$AC_MEAS_CHSFR	Auswahl Systemframes	Bitmaske entsprechend MD28082 $MC_MM_SYSTEM_FRAME_MASK
INT	$AC_MEAS_NCBFR	Auswahl globale Basisframes	Bitmaske (0 ... FFFF)
INT	$AC_MEAS_CHBFR	Auswahl Kanal-Basisframes	Bitmaske (0 ... FFFF)
INT	$AC_MEAS_UIFR	Auswahl einstellbarer Frames	0 ... 99
INT	$AC_MEAS_PFRAME	programmierbares Frame	0: wird eingerechnet 1: wird nicht eingerechnet

Für die entsprechenden Werte in den Variablen werden die Datenhaltungsframes gelesen und eine neue Framekette aufgebaut.

Hinweis

Werden Variablen nicht gesetzt, so werden die aktiven Frames beibehalten.

Es sind nur die Variablen zu beschreiben, deren Datenhaltungsframes auch in die neue Framekette einbezogen werden sollen. Bei den Basisframes können **nur alle** ausgetauscht werden und nicht nur ein spezieller. Aktive Änderungen über $P_NCBFRMASK und $P_CHBFRMASK werden nicht berücksichtigt.

Feldvariable für die Werkstück- und Werkzeugvermessung

Für weitere Eingangsparameter, die in den verschiedenen Messtypen verwendet werden, dient die folgende Feldvariable der Länge n

Typ	Systemvariable	Bedeutung	Werte
REAL	$AC_MEAS_INPUT[n]	Messeingangsparameter	n = 0 ... 9

Die Steuerungswirkung der Messeingangsparameter ist in den Messvarianten beschrieben.

Auswahl Werkzeug oder Schneide

Die Werkzeug- und Schneidennummer des aktiven Werkzeuges muss mit dem ausgewählten Werkzeug übereinstimmen. Bei Auswahl von T0, D0 wird das aktive Werkzeug eingerechnet. Ist kein Werkzeug aktiv, so wird das durch T, D angewählte Werkzeug eingerechnet. Es darf aber kein anderes Werkzeug, als das ausgewählte aktiv sein.

Typ	Systemvariable	Bedeutung
INT	$AC_MEAS_T_NUMBER	ausgewähltes Werkzeug
INT	$AC_MEAS_D_NUMBER	ausgewählte Schneide

Messungen mit 3D-Messtaster

Bei Messungen mit dem 3D-Messtaster ist der Radius des Werkzeuges bereits über den Messpunkt kompensiert, so dass der Radius in die Berechnung der verschiedenen Messvorgänge nicht mehr einbezogen werden darf. Diese Eigenschaft kann durch folgende Variable vorgegeben werden:

Typ	Systemvariable	Bedeutung
INT	$AC_MEAS_TOOL_MASK	Werkzeuglage

Die Variable $AC_MEAS_TOOL_MASK kann folgende Werte annehmen:

Wert	Bedeutung
0x0	alle Werkzeuglängen werden berücksichtigt (Standard)
0x1	Radius WZ geht nicht in Berechnung ein
0x2	Werkzeuglage ist in x-Richtung (G19)
0x4	Werkzeuglage ist in y-Richtung (G18)
0x8	Werkzeuglage ist in y-Richtung (G17)
0x10	Länge WZ geht nicht in Berechnung ein
0x20	Länge des aktiven Werkzeuges geht bei der Koordinatentransformation einer Position mit ein
0x40	Werzeuglage ist in -x-Richtung
0x80	Werzeuglage ist in -y-Richtung
0x100	Werzeuglage ist in -z-Richtung
0x200	Werkzeuglängen-Differenzwerte werden neg. eingerechnet

Aus der Werkzeuglage und der Anfahrrichtung kann erkannt werden, ob der Radius eines Fräsers mit in die Berechnung eingeht. Wird die Anfahrrichtung nicht explizit vorgegeben, so ergibt sie sich aus der ausgewählten Ebene:

Arbeitsebene	Anfahrrichtung
G17	-z-Richtung
G18	-y-Richtung
G19	-x-Richtung

8.4.2.2 Auswahl der Messung

Die Auswahl der Messung wird mit folgender Variable festgelegt:

Typ	Systemvariable	Bedeutung
INT	$AC_MEAS_TYPE	Auswahl Messtyp

Die Variable $AC_MEAS_TYPE kann folgende Werte annehmen:

Wert		Bedeutung
0		Vorbesetzung
1	Edge_x	Messung der x-Kante
2	Edge_y	Messung der y-Kante
3	Edge_z	Messung der z-Kante
4	Corner_1	Messung der Ecke 1
5	Corner_2	Messung der Ecke 2
6	Corner_3	Messung der Ecke 3
7	Corner_4	Messung der Ecke 4
8	Hole	Messung einer Bohrung
9	Stud	Messung einer Welle

Wert		Bedeutung
10 *	ToolLength	Messung der Werkzeuglänge
11 *	ToolDiameter	Messung des Werkzeugdurchmessers
12	Slot	Messung einer Nut
13	Plate	Messung eines Steges
14	Set_Pos	Istwertsetzen für Geo- und Zusatzachsen
15	Set_AuxPos	Istwertsetzen nur für Zusatzachsen
16	Edge_2P	Messung einer schrägen Kante
17	Plane_Angles	Winkel einer Ebene
18	Plane_Normal	Winkel einer Ebene mit Sollwertvorgabe
19	Dimension_1	1-dimensionale Sollwertvorgabe
20	Dimension_2	2-dimensionale Sollwertvorgabe
21	Dimension_3	3-dimensionale Sollwertvorgabe
22 *	ToolMagnifier	ShopTurn: Messen von Werkzeuglängen mit Lupe
23 *	ToolMarkedPos	ShopTurn: Messen einer Werkzeuglänge mit gemerkter Position
24	Koordinatentransformation	Koordinatentransformation einer Position
25	Rectangle	Messen eines Rechtecks
26	Save	Sichern von Datenhaltungsframes
27	Restore	Datenhaltungsframes wiederherstellen
28	Kegeldrehen	Additive Drehung der Ebene

* Messtypen der Werkzeugvermessung

Die einzelnen Messmethoden werden unter Kapitel "Messtypen der Werkstückvermessung" oder "Messtypen der Werkzeugvermessung" genauer dargestellt und mittels eines geeigneten Programmierbeispiels näher erläutert.

8.4.2.3 Ausgangswerte

Ergebnisse der Berechnung

Wurde eine Sollposition angegeben, so wird der resultierende Frame in das Ergebnis-Frame $AC_MEAS_FRAME eingetragen. Dieses Frame kann im Teileprogramm gelesen und geschrieben werden. Das Ergebnis-Frame wird entsprechend dem ausgewählten Frame berechnet.

Wurde kein Frame ausgewählt, so ergibt das Ergebnis-Frame die resultierende Translation und Drehung im WKS. Mit dem PI-Dienst _N_SETUDT und Parameter Typ Nr. 7 kann dieser Frame in den selektierten Frame eingetragen werden. Nach dem Eintragen des Frames wird der Ergebnisframe gelöscht.

Tabelle 8- 7 Ausgangswerte der Berechnungsergebnisse

Typ	Systemvariable	Bedeutung
FRAME	$AC_MEAS_FRAME	Ergebnisframe
REAL	$AC_MEAS_WP_ANGLE	Berechneter Werkstücklage-Winkel α
REAL	$AC_MEAS_CORNER_ANGLE	Berechneter Schnittwinkel ϕ

Typ	Systemvariable	Bedeutung
REAL	$AC_MEAS_DIAMETER	Berechneter Durchmesser
REAL	$AC_MEAS_TOOL_LENGTH	Berechnete Werkzeuglänge
REAL	$AC_MEAS_RESULTS[10]	Berechnungsergebnisse (abhängig von $AC_MEAS_TYPE)

8.4.2.4 Berechnungsmethode

Aktivierung der Berechnung

Die Aktivierung der Berechnung über die HMI-Bedienung erfolgt mit den PI-Dienst _N_SETUDT. Dieser PI-Dienst kann einen der folgenden Parameter-Typen annehmen:

Typ	Bedeutung
1	aktive Werkzeugkorrektur
2	aktive Basisframes
3	aktiver einstellbarer Frame
4	globale Basisframes
5	globale einstellbare Frames
6	Werkstück-Nullpunkt oder Werkzeuglängen berechnen
7	Werkstück-Nullpunkt aktivieren (Ankratzen schreiben)
8	externe Nullpunktverschiebung aktivieren
9	aktiven Tool-Carrier, TCOABS und PAROT aktivieren

Im Reset-Zustand wird die Änderung sofort ersichtlich, im Stopp-Zustand wird der Frame erst mit dem nächsten Start herausgefahren.

Hinweis

Der PI-Dienst kann nur im Reset- und Stopp-Zustand ausgeführt werden. Der berechnete Frame im Falle der Werkstückvermessung wird mit Typ Nr. 7 sofort aktiviert. Bei der Vermessung von Werkzeugen darf der PI mit Typ Nr. 7 nicht abgeschickt werden, da kein Nullpunkt aktiviert werden muss.

Aktivierung im Stopp-Zustand

Die neuen WKS-Positionen werden im Stopp-Zustand aktualisiert. Mit dem Fortsetzstart auf das Teileprogramm wird der Restweg des unterbrochenen Satzes gelöscht und es wird von der momentanen Position auf den Endpunkt des nächsten Satzes angefahren.

Damit kann auch im Stopp-Zustand eine Spindel in der Betriebsart MDA oder im Teileprogramm starten und mit M0 ein Istwertsetzen und Ankratzen oder eine andere Messung durchgeführt werden.

Messzyklen

Die Berechnung in den Messzyklen erfolgt über die vordefinierte Funktion:

INT MEASURE( )

MEASURE() liefert ein Ergebnis-Frame, das über $AC_MEAS_FRAME gelesen werden kann:

- Das Ergebnis ist die Translation und Rotation aus den Sollwerten, umgerechnet auf das selektierte Frame.
- Der Ergebnis-Frame berechnet sich wie folgt:

  Der verkettete Summenframe ergibt gleich die Verkettung von Gesamtframe (vor der Messung) mit der berechneten Translation und Rotation.

Hinweis

Ist kein Frame ausgewählt, so wird der berechnete Frame nicht transformiert. D. h. die Translation und Rotation ergibt sich anhand der vorgegebenen Sollwerte und der berechneten Position der Kante, Ecke, Nut, etc. Eine mehrfache Anwendung der Funktion wirkt immer additiv zum Ergebnisframe.

Es ist darauf zu achten, dass das Ergebnisframe evtl. vorher gelöscht werden muss.

ACHTUNG
MEASURE() löst keinen impliziten Vorlaufstopp aus. Da MEASURE() mit den Frames des Vorlaufsatzes arbeitet, muss Sie selbst entscheiden, ob ein Vorlaufstopp vor der Berechnung notwendig ist.

Semaphor-Variable

Pro Kanal gibt es die Messvariablen nur einmal. Der Messvorgang kann über die Bedienung im Stopp- und im Reset-Zustand erfolgen. Im Stopp-Zustand kann der Bedienvorgang sich mit den Messzyklen überschneiden.Zum Schutz des gegenseitigen Überschreibens dient die Variable

$AC_MEAS_SEMA (Semaphore zum Messinterface)

Die Semaphor-Variable $AC_MEAS_SEMA wird vom Zyklus

- am Anfang mit 1 beschrieben und
- am Ende des Zyklus wieder auf 0 zurückgesetzt.

HMI benutzt das Messinterface nicht, wenn die Variable den Wert 1 hat.

Fehlermeldungen

Wenn der Client sich nicht anmeldet, wird immer die Sammelfehlernummer 0xD003 erzeugt. Erfolgt eine Anmeldung durch DIAGN:errCodeSetNrGent bzw. DIAGN:errCodeSetNrPi, so stellt PI_SETUDT Fehlercodes zur Verfügung, entsprechend folgender Syntax:

EX_ERR_PI_REJ_<Rückgabewert>, z. B.: EX_ERR_PI_REJ_ MEASNOTYPE

Folgende Rückgabewerte werden über die vordefinierte Funktion MEASURE() ausgegeben:

Tabelle 8- 8 Vordefinierte Fehlermeldungen

Nr.	Rückgabewerte	Bedeutung
0	MEAS_OK	Korrekte Berechnung
1	MEAS_NO_TYPE	Type nicht spezifiziert
2	MEAS_TOOL_ERROR	Fehler bei der Werkzeug-Ermittlung
3	MEAS_NO_POINT1	Messpunkt 1 nicht vorhanden
4	MEAS_NO_POINT2	Messpunkt 2 nicht vorhanden
5	MEAS_NO_POINT3	Messpunkt 3 nicht vorhanden
6	MEAS_NO_POINT4	Messpunkt 4 nicht vorhanden
7	MEAS_NO_SPECPOINT	Kein Referenzpunkt vorhanden
8	MEAS_NO_DIR	Keine Anfahrrichtung
9	MEAS_EQUAL_POINTS	Messpunkte sind identisch
10	MEAS_WRONG_ALPHA	Alpha α ist falsch
11	MEAS_WRONG_PHI	Phi ϕ ist falsch
12	MEAS_WRONG_DIR	Falsche Anfahrrichtung
13	MEAS_NO_CROSSING	Geraden schneiden sich nicht
14	MEAS_NO_PLANE	Ebenen nicht vorhanden
15	MEAS_WRONG_FRAME	Kein oder falsches Frame selektiert
16	MEAS_NO_MEMORY	Nicht genügend Speicher vorhanden
17	MEAS_INTERNAL_ERROR	Interner Fehler

Werkzeug-Ermittlungsfehler

Im Falle des Errorcodes MEAS_TOOL_ERROR oder EX_ERR_PI_REJ_MEASTOOLERROR wird vom System in die Ausgabevariable $AC_MEAS_TOOL_LENGTH eine genauere Spezifikation des Fehlers mit den folgenden Werten abgelegt:

Tabelle 8- 9 Vordefinierte Fehlermeldungen für MEAS_TOOL_ERROR

Nr.	Rückgabewerte	Bedeutung
1	TOOL_NO_BLOCK	Kein Satz für die Tool-Berechnung verfügbar
2	TOOL_WRONG_T_NUMBER	Falsche T-Nummer
3	TOOL_WRONG_D_NUMBER	Falsche D-Nummer
4	TOOL_EVAL_WRONG_TYPE	Das Werkzeug existiert nicht
5	TOOL_NO_TOOLCORR_BODY	Speicher Problem
6	TOOL_DATA_READ_ERROR	Fehler beim Lesen der Werkzeugdaten
7	TOOL_NO_TOOL_WITH_TRAFO	Bei aktiver Transformation ist kein Tool angewählt

8.4.2.5 Maßeinheiten und Meßgrößen für die Berechnung

Maßeinheiten INCH oder METRISCH

Folgende Ein- und Ausgabevariablen werden mit den Maßeinheiten Inch oder Metrisch bewertet:

$AA_MEAS_POINT1[Achse]	Eingangsvariable für 1. Messpunkt
$AA_MEAS_POINT2[Achse]	Eingangsvariable für 2. Messpunkt
$AA_MEAS_POINT3[Achse]	Eingangsvariable für 3. Messpunkt
$AA_MEAS_POINT4[Achse]	Eingangsvariable für 4. Messpunkt
$AA_MEAS_SETPOINT[Achse]	Eingangsvariable für Sollposition
$AC_MEAS_DIAMETER	Ausgangsvariable berechneter Durchmesser
$AC_MEAS_TOOL_LENGTH	Ausgangsvariable berechnete WZ-Länge
$AC_MEAS_RESULTS[n]	Ausgangsvariable Berechnungsergebnisse

Das Maßsystem in dem die Ein- und Ausgangswerte gelesen oder geschrieben werden, kann eingestellt werden über die Eingangsvariable

INT $AC_MEAS_SCALEUNIT	Maßeinheit für Ein- und Ausgangsvariable
0: Maßeinheit ist bezüglich des	aktiven G-Codes G70/G700 in INCH aktiven G-Codes G71/G701 in METRISCH
1: Maßeinheit ist entsprechend	der Projektierung, das Maßsystem kann über BTSS eingestellt werden (Standardeinstellung)

Wird die Variable nicht geschrieben, so gilt immer der Wert 1 als Standardeinstellung.

Beispiel:

Grundsystem ist Metrisch:

```
G70
$AC_MEAS_POINT1[x] = $AA_IW[x]            ; $AA_IW[x] liefert Grundsystem
$AC_MEAS_POINT1[x] = 10                   ; 10 mm
G71
$AC_MEAS_POINT1[x] = $AA_IW[x]            ; $AA_IW[x] liefert Grundsystem
$AC_MEAS_POINT1[x] = 10                   ; 10 mm
G700
$AC_MEAS_POINT1[x] = $AA_IW[x]            ; $AA_IW[x] liefert Inch-Wert
$AC_MEAS_POINT1[x] = 10                   ; 10 Inch
G710
$AC_MEAS_POINT1[x] = $AA_IW[x]            ; $AA_IW[x] liefert metrischen Wert
$AC_MEAS_POINT1[x] = 10                   ; 10 mm
```

Durchmesserprogrammierung

Die Durchmesserprogrammierung wird eingestellt durch die Maschinendaten:

```
MD20100 $MC_DIAMETER_AX_DEF = "X"          ; Planachse ist x
MD20150 $MC_GCODE_RESET_VALUES[28] = 2     ; DIAMON
MD20360 $MC_TOOL_PARAMETER_DEF_MASK        ; WZL, Frames und
= 'B1001010'                               ; Istwerte im Durchmesser
```

Folgendes ist zu beachten:

- Achspositionen im MKS werden nicht als Durchmesserwert eingerechnet.
- Die berechneten Werkzeuglängen und Framekomponenten sind nicht abhängig vom aktiven G-Code DIAMON oder DIAMOF.
- Die Mess- und Sollpositionen werden abhängig von DIAMON gelesen und beschrieben.
- Die Translationen in den Frames werden als Durchmesserwert in der Planachse eingerechnet.

Rechen- und Anzeigegenauigkeit

Positionswerte in mm, Inch oder Grad werden auf 6 Nachkommastellen genau berechnet und angezeigt.

8.4.2.6 Diagnose

Für das Messinterface gibt es folgende Diagnosemöglichkeit:

- Bei vorhandenem File /_N_MPF_DIR/_N_MEAS_DUMP_MPF wird ein Protokoll in das File geschrieben, das eine Reproduzierung des Problems ermöglichen soll.
- Die Protokollierung startet man, indem man ein leeres File mit dem Namen _N_MEAS_DUMP_MPF im Directory /_N_MPF_DIR anlegt.
- Der Inhalt des Files bleibt solange erhalten, bis es mit $AC_MEAS_VALID = 0 gelöscht wird.

Aus Laufzeitgründen sollte man den Trace nur einschalten, wenn ein Problem erkannt wurde.

8.4.3 Messtypen der Werkstückvermessung

8.4.3.1 Messen einer Kante (Mess Type 1, 2, 3)

Messen einer x-Kante ($AC_MEAS_TYPE = 1)

Die Kante eines eingespannten Werkstückes wird durch Anfahren an diese Kante mit einem bekannten Werkzeug vermessen.

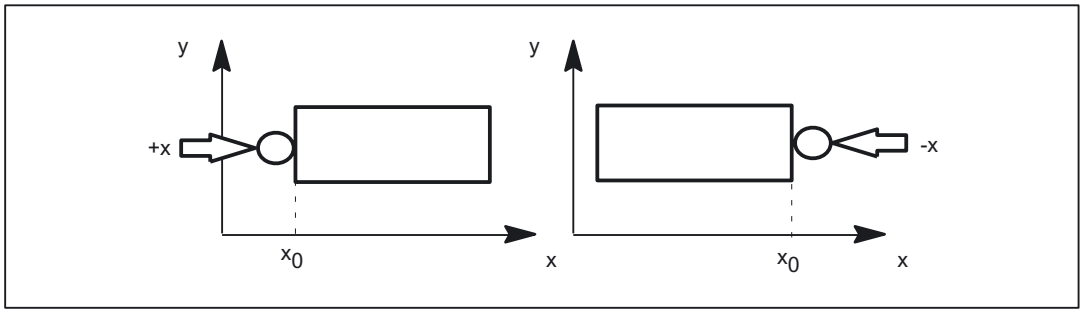


Bild 8-4 x-Kante

Für den Messtyp 1 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für alle Kanalachsen
$AA_MEAS_SETPOINT[Achse]	Sollposition der x-Kante *
$AC_MEAS_DIR_APPROACH	0: +x, 1: -x
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet, Radius des Werkzeugs geht nur bei G17 und G18 ein *
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_TYPE	1

* optional

Für den Messtyp 1 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Translation
$AC_MEAS_RESULTS[0]	Position der gemessenen Kante

Beispiel

x-Kantenmessung

```
DEF INT RETVAL
DEF FRAME TMP

$TC_DP1[1,1]=120                    ; Typ
$TC_DP2[1,1]=20                     ; 0
$TC_DP3[1,1]= 10                    ; (z) Längenkorrekturvektor
$TC_DP4[1,1]= 0                     ; (y)
$TC_DP5[1,1]= 0                     ; (x)
$TC_DP6[1,1]= 2                     ; Radius

T1 D1
g0 x0 y0 z0 f10000
G54

                                    ; x-Kante vermessen
$AC_MEAS_VALID = 0                  ; Alle Eingangswerte ungültig setzen

g1 x-1 y-3                          ; 1. Messpunkt anfahren

$AA_MEAS_POINT1[x] = $AA_IW[x]
$AA_MEAS_POINT1[y] = $AA_IW[y]
$AA_MEAS_POINT1[z] = $AA_IW[z]

$AC_MEAS_DIR_APPROACH = 0           ; Anfahrrichtung +x setzen

$AA_MEAS_SETPOINT[x] = 0            ; Sollposition der Kante setzen
$AA_MEAS_SETPOINT[y] = 0
$AA_MEAS_SETPOINT[z] = 0

$AC_MEAS_ACT_PLANE = 0              ; Ebene für die Messung ist G17

$AC_MEAS_FRAME_SELECT = 101         ; Frame auswählen - IFRAME

$AC_MEAS_T_NUMBER = 1               ; Werkzeug auswählen
$AC_MEAS_D_NUMBER = 1

$AC_MEAS_TYPE = 1                   ; Messtyp x-Kante setzen

RETVAL = MEASURE()                  ; Messvorgang starten

if RETVAL <> 0
```

```
setal(61000 + RETVAL)
endif

$P_IFRAME = $AC_MEAS_FRAME

$P_UIFR[1] = $P_IFRAME                ; Systemframe in der Datenhaltung beschreiben

g1 x0 y0                              ; Fahre die Kante an

m30
```

Messen einer y-Kante ($AC_MEAS_TYPE = 2)

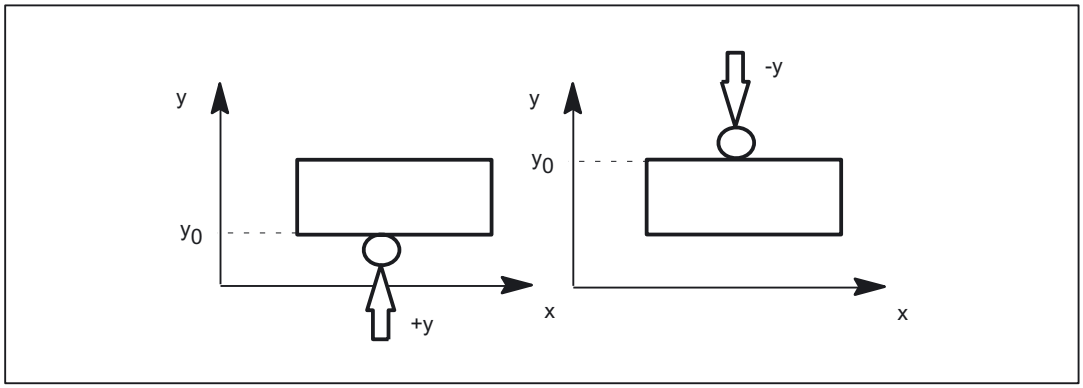


Bild 8-5 y-Kante

Für den Messtyp 2 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für alle Kanalachsen
$AA_MEAS_SETPOINT[Achse]	Sollposition der y-Kante *
$AC_MEAS_DIR_APPROACH	2: +y, 3: -y
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet, Radius des Werkzeugs geht nur bei G17 und G19 ein *
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additiver Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_TYPE	2

* optional

Für den Messtyp 2 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Translation
$AC_MEAS_RESULTS[0]	Position der gemessenen Kante

Messen einer z-Kante ($AC_MEAS_TYPE = 3)

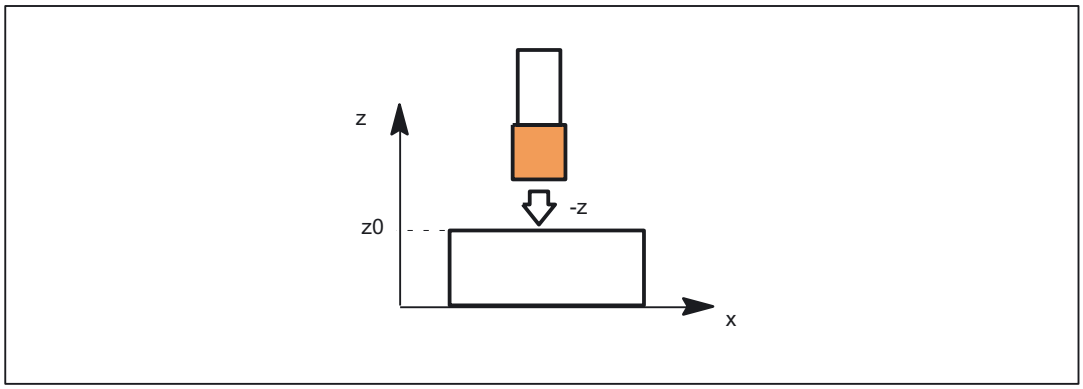


Bild 8-6 z-Kante

Für den Messtyp 3 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für alle Kanalachsen
$AA_MEAS_SETPOINT[Achse]	Sollposition der z-Kante *
$AC_MEAS_DIR_APPROACH	4: +z, 5: -z
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet, Radius des Werkzeugs geht nur bei G18 und G19 ein *
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additiver Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_TYPE	3

* optional

Für den Messtyp 3 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Translation
$AC_MEAS_RESULTS[0]	Position der gemessenen Kante

8.4.3.2 Messung eines Winkels (Mess Type 4, 5, 6, 7)

Messen einer Ecke C1 - C4 ($AC_MEAS_TYPE = 4, 5, 6, 7)

Eine Ecke ist durch Anfahren von 4 Messpunkten P1 bis P4 eindeutig definiert. Bei bekannten Schnittwinkel ϕ können 3 Messpunkte ausreichen.

Ist der Schnittwinkel ϕ und der Werkstücklage-Winkel α bekannt, so sind 2 Messpunkte P1 und P3 ausreichend.

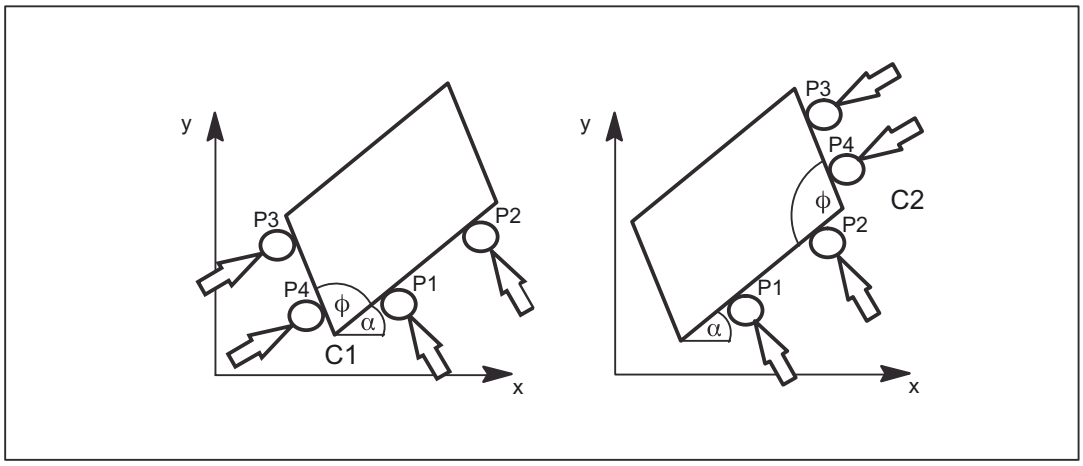


Bild 8-7 Ecke C1, Ecke C2

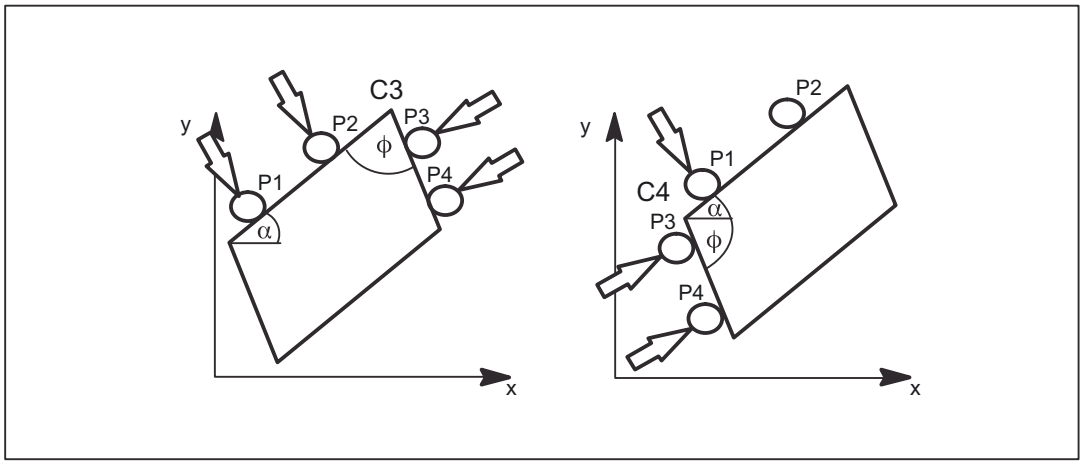


Bild 8-8 Ecke C3, Ecke C4

Für Messtypen 4 bis 7 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1
$AA_MEAS_POINT2[Achse]	Messpunkt 2 nicht relevant bei $AC_MEAS_WP_SETANGLE
$AA_MEAS_POINT3[Achse]	Messpunkt 3

Eingangs-Variable	Bedeutung
$AA_MEAS_POINT4[Achse]	Messpunkt 4 nicht relevant bei $AC_MEAS_CORNER_SETANGLE
$AA_MEAS_WP_SETANGLE	Soll-Werkstücklage-Winkel *
$AA_MEAS_CORNER_SETANGLE	Soll-Schnittwinkel *
$AA_MEAS_SETPOINT[Achse]	Soll-Position der Ecke *
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additiver Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_INPUT[0]	ohne Angabe für die Außenecke* =0: Messung für Außenecke =1: Messung für Innenecke
$AC_MEAS_TYPE	4, 5, 6, 7

* optional

Für die Messtypen 4 bis 7 werden folgende Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Translation und Drehung
$AC_MEAS_WP_ANGLE	Berechneter Werkstücklage-Winkel
$AC_MEAS_CORNER_ANGLE	Berechneter Schnittwinkel
$AC_MEAS_RESULTS[0]	Abszisse des berechneten Eckpunktes
$AC_MEAS_RESULTS[1]	Ordinate des berechneten Eckpunktes
$AC_MEAS_RESULTS[2]	Applikate des berechneten Eckpunktes

Beispiel

Eckenmessung C1: Ecke mit 3 Messpunkten (P1, P3 und P4), mit bekanntem Schnittwinkel ϕ (90°) und unbekannten Werkstücklage-Winkel α.

```
DEF INT RETVAL
DEF FRAME TMP

$TC_DP1[1,1]=120                        ; Typ
$TC_DP2[1,1]=20                         ; 0
$TC_DP3[1,1]= 10                        ; (z) Längenkorrekturvektor
$TC_DP4[1,1]= 0                         ; (y)
$TC_DP5[1,1]= 0                         ; (x)
$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

                                        ; Ecke mit 3 Messpunkten vermessen
$AC_MEAS_VALID = 0                      ; Alle Eingangswerte ungültig setzen

g1 x-1 y-3                              ; 1. Messpunkt anfahren
$AC_MEAS_LATCH[0] = 1                   ; Messpunkt P1 ablatchen

g1 x-4 y4                               ; 3. Messpunkt anfahren
$AC_MEAS_LATCH[2] = 1                   ; Messpunkt P3 ablatchen

g1 x-4 y1                               ; 4. Messpunkt anfahren
$AC_MEAS_LATCH[3] = 1                   ; Messpunkt P4 ablatchen

$AA_MEAS_SETPOINT[x] = 0                ; Sollposition der Ecke auf (0, 0, 0)
                                          setzen
$AA_MEAS_SETPOINT[y] = 0
$AA_MEAS_SETPOINT[z] = 0

$AC_MEAS_CORNER_SETANGLE = 90           ; Sollschnittwinkel φ vorgeben
$AC_MEAS_ACT_PLANE = 0                  ; Ebene für die Messung ist G17
$AC_MEAS_FRAME_SELECT = 0               ; Frame auswählen - SETFRAME

$AC_MEAS_T_NUMBER = 1                   ; Werkzeug auswählen
$AC_MEAS_D_NUMBER = 1

$AC_MEAS_TYPE = 4                       ; Messtyp auf Ecke 1 setzen

RETVAL = MEASURE()                      ; Messvorgang starten

if RETVAL <> 0
setal(61000 + RETVAL)
endif

if $AC_MEAS_CORNER_ANGLE <> 90          ; bekannten Sollschnittwinkel φ abfragen
setal(61000 + $AC_MEAS_CORNER_ANGLE)
endif

$P_SETFRAME = $AC_MEAS_FRAME

$P_SETFR = $P_SETFRAME                  ; Systemframe in der Datenhaltung
```

```
                                                beschreiben

g1 x0 y0                                      ; Fahre die Ecke an

g1 x10                                        ; Rechteck abfahren
y10
x0
y0

m30
```

8.4.3.3 Messen einer Bohrung (Mess Type 8)

Messpunkte zur Bestimmung einer Bohrung ($AC_MEAS_TYPE = 8)

Zur Bestimmung von Mittelpunkt und Durchmesser sind 3 Messpunkte erforderlich. Die drei Punkte müssen verschieden voneinander sein. Bei Angabe von 4 Punkten wird der Kreis nach der kleinsten Fehlerquadratmethode angepasst. Der Kreis wird so ermittelt, dass die Summe der Abstandsquadrate der Punkte vom resultierenden Kreis minimal wird. Die Güte der Anpassung kann gelesen werden.

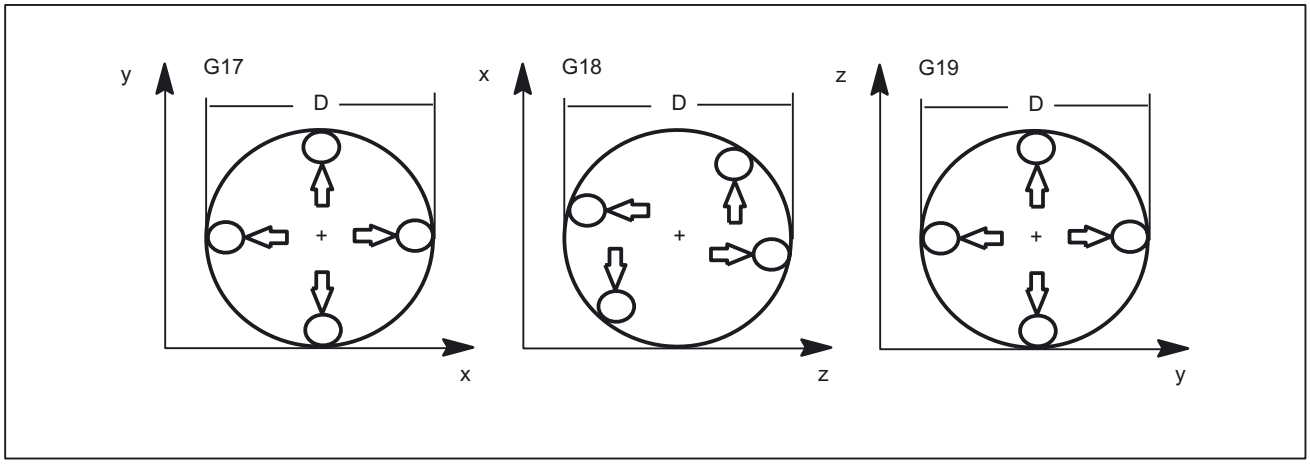


Bild 8-9 Bohrung

Für den Messtyp 8 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1
$AA_MEAS_POINT2[Achse]	Messpunkt 2
$AA_MEAS_POINT3[Achse]	Messpunkt 3
$AA_MEAS_POINT4[Achse]	bei Angabe wird die Mitte aus 4 Punkten bestimmt *
$AA_MEAS_SETPOINT[Achse]	Soll-Position der Bohrungsmitte *
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *

Eingangs-Variable	Bedeutung
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_TYPE	8

* optional

Für den Messtyp 8 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Translation
$AC_MEAS_DIAMETER	Durchmesser der Bohrung
$AC_MEAS_RESULTS[0]	Abszisse des berechneten Mittelpunktes
$AC_MEAS_RESULTS[1]	Ordinate des berechneten Mittelpunktes
$AC_MEAS_RESULTS[2]	Applikate des berechneten Mittelpunktes
$AC_MEAS_RESULTS[3]	Gütemaß für die Kreisanpassung: Summe der Abstandsquadrate

Beispiel

Messen einer Bohrung

```
DEF INT RETVAL
DEF FRAME TMP

$TC_DP1[1,1]=120                    ;  Typ
$TC_DP2[1,1]=20                     ;  0
$TC_DP3[1,1]= 10                    ;  (z) Längenkorrekturvektor
$TC_DP4[1,1]= 0                     ;  (y)
$TC_DP5[1,1]= 0                     ;  (x)
$TC_DP6[1,1]= 2                     ;  Radius

T1 D1
g0 x0 y0 z0 f10000
G54

                                    :  Bohrung vermessen
$AC_MEAS_VALID = 0                  ;  alle Eingangswerte ungültig setzen

g1 x-3 y0                           ;  1. Messpunkt anfahren
$AA_MEAS_POINT1[x] = $AA_IW[x]
$AA_MEAS_POINT1[y] = $AA_IW[y]
$AA_MEAS_POINT1[z] = $AA_IW[z]
```

```
g1 x0 y3                              ;  2. Messpunkt anfahren
$AA_MEAS_POINT2[x] = $AA_IW[x]
$AA_MEAS_POINT2[y] = $AA_IW[y]
$AA_MEAS_POINT2[z] = $AA_IW[z]

g1 x3 y0                              ;  3. Messpunkt anfahren
$AA_MEAS_POINT3[x] = $AA_IW[x]
$AA_MEAS_POINT3[y] = $AA_IW[y]
$AA_MEAS_POINT3[z] = $AA_IW[z]

$AA_MEAS_SETPOINT[x] = 0              ;  Sollposition der Mitte setzen
$AA_MEAS_SETPOINT[y] = 0
$AA_MEAS_SETPOINT[z] = 0

$AC_MEAS_ACT_PLANE = 0                ;  Ebene für die Messung ist G17
$AC_MEAS_FRAME_SELECT = 0             ;  Frame auswählen - SETFRAME

$AC_MEAS_T_NUMBER = 1                 ;  Werkzeug auswählen
$AC_MEAS_D_NUMBER = 1

$AC_MEAS_TYPE = 8                     ;  Messtyp auf Bohrung setzen

RETVAL = MEASURE()                    ;  Messvorgang starten
if RETVAL <> 0
setal(61000 + RETVAL)
endif

if $AC_MEAS_DIAMETER <> 10            ;  bekannten Durchmesser abfragen
setal(61000 + $AC_MEAS_WP_ANGLE)
endif

$P_SETFRAME = $AC_MEAS_FRAME

$P_SETFR = $P_SETFRAME                ;  Systemframe in der Datenhaltung beschreiben

g1 x-3 y0                             ;  Fahre P1 an

g2 I = $AC_MEAS_DIAMETER / 2          ;  Bohrung bezüglich Kreismittelpunkt abfahren

m30
```

8.4.3.4 Messen einer Welle (Mess Type 9)

Messpunkte zur Bestimmung einer Welle ($AC_MEAS_TYPE = 9)

Zur Bestimmung von Mittelpunkt und Durchmesser sind 3 Messpunkte erforderlich. Die drei Punkte müssen verschieden voneinander sein. Bei Angabe von 4 Punkten wird der Kreis nach der kleinsten Fehlerquadratmethode angepasst. Der Kreis wird so ermittelt, dass die Summe der Abstandsquadrate der Punkte vom resultierenden Kreis minimal wird. Die Güte der Anpassung kann gelesen werden.

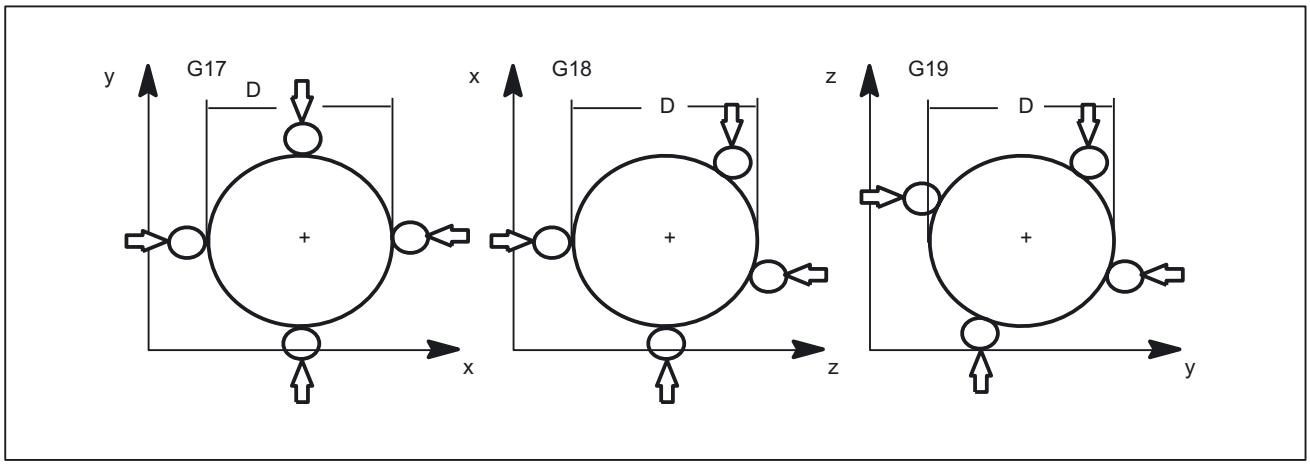


Bild 8-10 Welle

Für den Messtyp 9 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1
$AA_MEAS_POINT2[Achse]	Messpunkt 2
$AA_MEAS_POINT3[Achse]	Messpunkt 3
$AA_MEAS_POINT4[Achse]	bei Angabe wird die Mitte aus 4 Punkten bestimmt *
$AA_MEAS_SETPOINT[Achse]	Soll-Position des Wellen-Mittelpunktes *
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_TYPE	9

* optional

Für den Messtyp 9 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Translation
$AC_MEAS_DIAMETER	Durchmesser der Welle
$AC_MEAS_RESULTS[0]	Abszisse des berechneten Mittelpunktes
$AC_MEAS_RESULTS[1]	Ordinate des berechneten Mittelpunktes
$AC_MEAS_RESULTS[2]	Applikate des berechneten Mittelpunktes
$AC_MEAS_RESULTS[3]	Gütemaß für die Kreisanpassung: Summe der Abstandsquadrate

8.4.3.5 Messen einer Nut (Mess Type 12)

Messpunkte zur Bestimmung der Lage einer Nut ($AC_MEAS_TYPE = 12)

Eine Nut wird durch Anfahren der beiden Außenkanten oder Innenkanten vermessen. Die Nut-Mitte kann auf eine Sollposition gesetzt werden. Die Komponente der Anfahrrichtung legt die Nut-Lage fest.

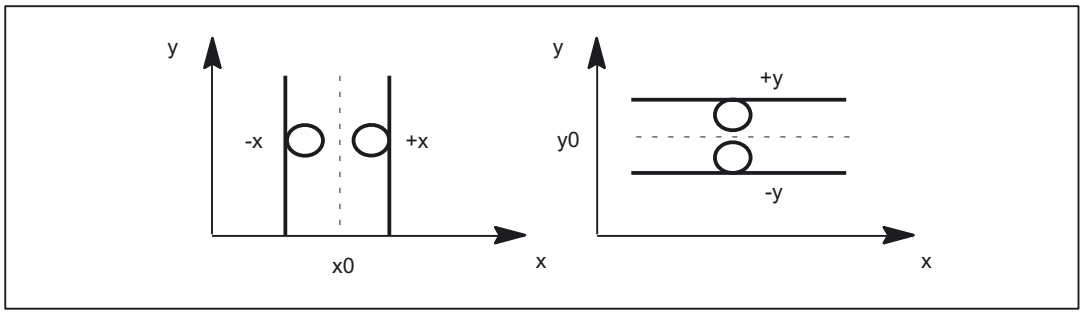


Bild 8-11 Nut

Für den Messtyp 12 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1
$AA_MEAS_POINT2[Achse]	Messpunkt 2
$AA_MEAS_SETPOINT[Achse]	Soll-Position der Nutmitte *
$AC_MEAS_DIR_APPROACH	0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additiver Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *

Eingangs-Variable	Bedeutung
$AC_MEAS_INPUT[0]	Anfahrrichtung für den 2. Messpunkt bei Vermessung eines Absatzes. Muss die gleiche Koordinate, wie die Anfahrrichtung des 1. Punktes haben. * 0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
$AC_MEAS_TYPE	12

* optional

Für den Messtyp 12 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Translation
$AC_MEAS_RESULTS[0]	Lage der berechneten Nut-Mitte (x0, y0 oder z0)
$AC_MEAS_RESULTS[1]	Nutbreite in Anfahrrichtung

Beispiel

Nutmessung mit Anfahrrichtung in x

```
DEF INT RETVAL
DEF FRAME TMP

$TC_DP1[1,1]=120                    ; Typ
$TC_DP2[1,1]=20                     ; 0
$TC_DP3[1,1]= 10                    ; (z) Längenkorrekturvektor
$TC_DP4[1,1]= 0                     ; (y)
$TC_DP5[1,1]= 0                     ; (x)
$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,rt] = -45

                                    ; Nut vermessen
$AC_MEAS_VALID = 0                  ; alle Eingangswerte ungültig setzen

g1 x-2                              ; 1. Messpunkt anfahren
$AA_MEAS_POINT1[x] = $AA_IW[x]
$AA_MEAS_POINT1[y] = $AA_IW[y]
$AA_MEAS_POINT1[z] = $AA_IW[z]
```

```
g1 x4                                ; 2. Messpunkt anfahren
$AA_MEAS_POINT2[x] = $AA_IW[x]
$AA_MEAS_POINT2[y] = $AA_IW[y]
$AA_MEAS_POINT2[z] = $AA_IW[z]

$AA_MEAS_SETPOINT[x] = 0             ; Sollposition der Nut-Mitte setzen
$AA_MEAS_SETPOINT[y] = 0
$AA_MEAS_SETPOINT[z] = 0

$AC_MEAS_DIR_APPROACH = 0            ; Anfahrrichtung +x setzen
$AC_MEAS_ACT_PLANE = 0               ; Ebene für die Messung ist G17
$AC_MEAS_FRAME_SELECT = 0            ; Frame auswählen - SETFRAME

$AC_MEAS_T_NUMBER = 1                ; Werkzeug auswählen
$AC_MEAS_D_NUMBER = 1

$AC_MEAS_TYPE = 12                   ; Messtyp auf Nut setzen

RETVAL = MEASURE()                   ; Messvorgang starten

if RETVAL <> 0 setal(61000 + RETVAL)
endif

$P_SETFRAME = $AC_MEAS_FRAME
$P_SETFR = $P_SETFRAME               ; Systemframe in der Datenhaltung beschreiben

g1 x0 y0                             ; Fahre die Nut-Mitte an

m30
```

8.4.3.6 Messen eines Steg (Mess Type 13)

Messpunkte zur Bestimmung der Lage eines Stegs ($AC_MEAS_TYPE = 13)

Ein Steg wird durch Anfahren der beiden Außenkanten oder Innenkanten vermessen. Die Steg-Mitte kann auf eine Sollposition gesetzt werden. Die Komponente der Anfahrrichtung legt die Steg-Lage fest.

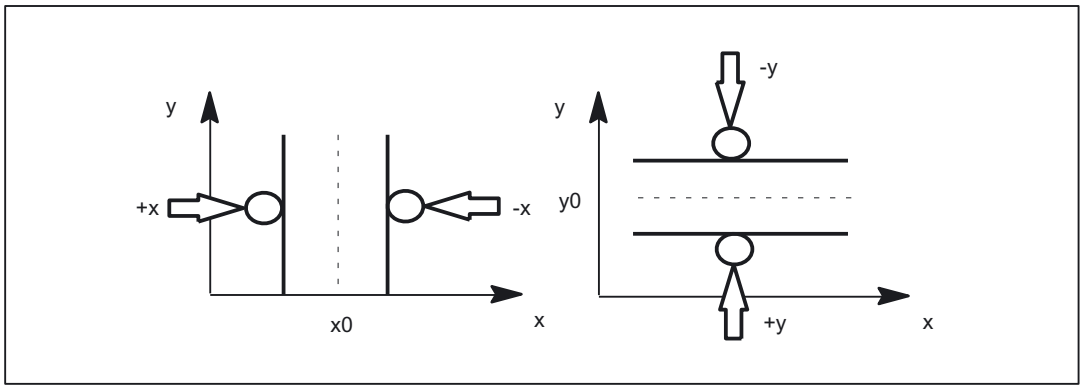


Bild 8-12 Steg

Für den Messtyp 13 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1
$AA_MEAS_POINT2[Achse]	Messpunkt 2
$AA_MEAS_SETPOINT[Achse]	Soll-Position der Stegmitte *
$AC_MEAS_DIR_APPROACH	0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additiver Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_INPUT[0]	Anfahrrichtung für den 2. Messpunkt bei Vermessung eines Absatzes. Muss die gleiche Koordinate, wie die Anfahrrichtung des 1. Punktes haben. * 0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
$AC_MEAS_TYPE	13

* optional

Für den Messtyp 13 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Translation
$AC_MEAS_RESULTS[0]	Lage der berechneten Steg-Mitte (x0, y0 oder z0)
$AC_MEAS_RESULTS[1]	Stegbreite in Anfahrrichtung

8.4.3.7 Messen von Geo- und Zusatzachsen (Mess Type 14, 15)

Istwertsetzen für Geo- und Zusatzachsen ($AC MEAS TYPE = 14)

Dieser Messtyp wird über die HMI-Bedienoberfläche genutzt.

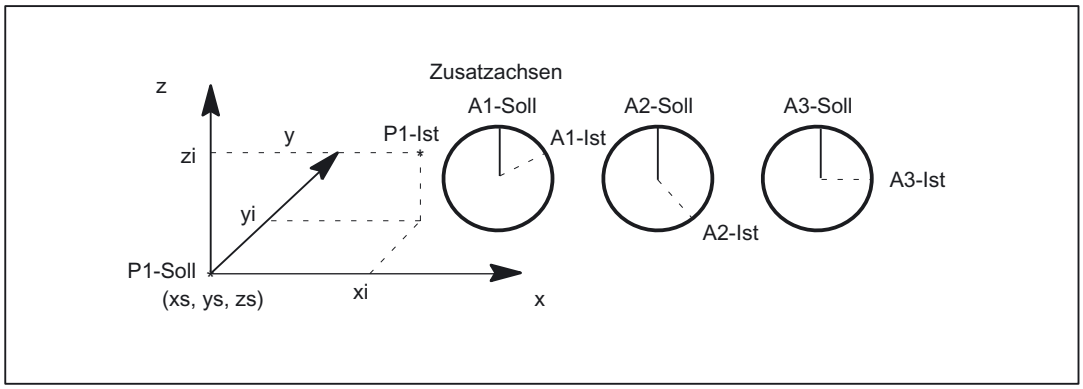


Bild 8-13 Istwertsetzen

Für den Messtyp 14 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Istwerte der Achsen
$AA_MEAS_SETPOINT[Achse]	Soll-Position der einzelnen Achsen *
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_TYPE	14

* optional

Für den Messtyp 14 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Translationen

Beispiel:

Referenzpunktsetzen in relativen Koordinatensystemen.

```
DEF INT RETVAL

T1 D1                                   ; Messtaster aktivieren
G54                                     ; alle Frames und G54 aktivieren

TRANS x=10                              ; Verschiebung zwischen WKS und ENS
G0 x0 f10000                            ; WKS(x) = 0; ENS(x) = 10

$AC_MEAS_VALID = 0                      ; alle Eingangsvariablen ungültig setzen
$AC_MEAS_TYPE = 14                      ; Messtyp für Istwertsetzen
$AC_MEAS_ACT_PLANE = 0                  ; Ebene fuer die Messung ist G17

$AC_MEAS_P1_COORD = 5                   ; ENS_REL für 1. Messpunkt
$AC_MEAS_LATCH[0] = 1                   ; Alle Achspositionen ablatchen

$AC_MEAS_SET_COORD = 5                  ; Sollposition ist relativ zu ENS
$AA_MEAS_SETPOINT[x] = 0                ; Sollposition im rel. ENS-
                                          Koordinatensystem
$AC_MEAS_FRAME_SELECT = 2505            ; $P_RELFR

RETVAL = MEASURE()                      ; Berechnung von $P_RELFR; PI SETUDT(6)
IF RETVAL <> 0 GOTOF ERROR
ENDIF $ P_RELFR = $AC_MEAS_FRAME        ; Aktivierung; PI SETUDT(7)
```

Istwertsetzen nur für Zusatzachsen ($AC MEAS TYPE = 15)

Dieser Messtyp wird über die HMI-Bedienoberfläche genutzt.

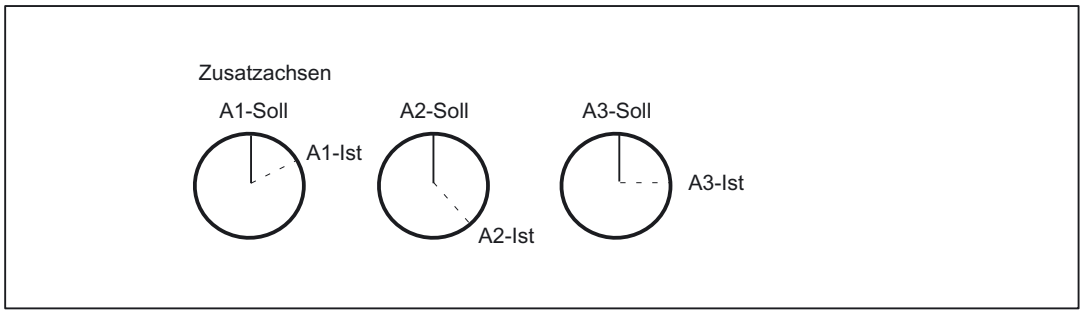


Bild 8-14 Istwertsetzen nur für Zusatzachsen

Für den Messtyp 15 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Istwerte der Achsen
$AA_MEAS_SETPOINT[Achse]	Soll-Position der einzelnen Achsen *
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additiver Frame berechnet *
$AC_MEAS_TYPE	15

* optional

Für den Messtyp 15 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Translationen

8.4.3.8 Messen einer schrägen Kante (Mess Type 16)

Messen einer schrägen Kante ($AC_MEAS_TYPE = 16)

Mit dieser Messung wird der Werkstücklagewinkel bestimmt und in den Frame eingetragen. Es kann ein Sollwinkel im Bereich +/- 90 Grad vorgegeben werden, der als resultierende Verdrehung des Werkstückes nach Aktivierung des Ergebnisframe zum aktiven WKS interpretiert werden kann.

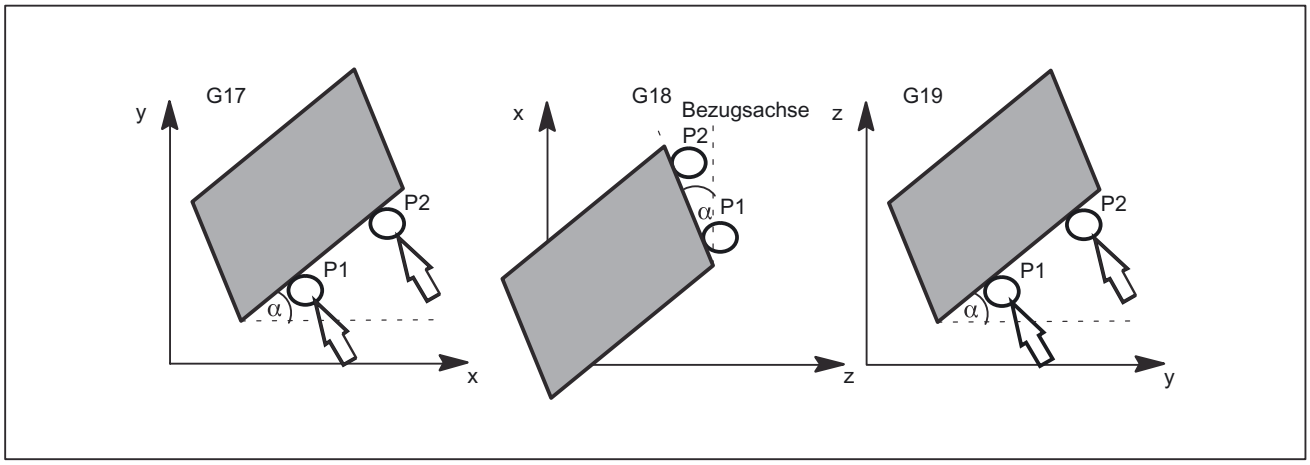


Bild 8-15 Schräge Kante in der Ebene G17, G18 und G19

Für den Messtyp 16 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1

Eingangs-Variable	Bedeutung
$AA_MEAS_POINT2[Achse]	Messpunkt 2
$AA_MEAS_SETANGLE	Soll-Winkel *
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additiver Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_INPUT[0]	ohne Angabe ist die Bezugskoordinate, auf welche das Werkstück ausgerichtet werden soll immer die Abszisse der ausgewählten Ebene. * =0: Bezugskoordinate ist die Abszisse =1: Bezugskoordinate ist die Ordinate
$AC_MEAS_INPUT[1]	ohne Angabe wird der Werkstücklage-Winkel als Drehung in das Frame eingetragen. Ansonsten kann ein Kanalachsindex für eine Rundachse angegeben werden, deren Translation auf die aktuelle Rundachsposition plus der berechneten Verdrehung gesetzt wird. Das Werkstück ist dann bei der Rundachsposition = 0 ausgerichtet. Der aktuelle Rundachswert muss in $AA_MEAS_POINT[Achse] vorliegen. *
$AC_MEAS_TYPE	16

* optional

Für den Messtyp 16 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Drehung
$AC_MEAS_WP_ANGLE	Berechneter Werkstücklage-Winkel

8.4.3.9 Messen eines schrägen Winkels in einer Ebene (Mess Type 17)

Messen eines Winkels in einer Ebene ($AC_MEAS_TYPE = 17)

Die schräge Ebene wird über drei Messpunkte P1, P2 und P3 bestimmt.

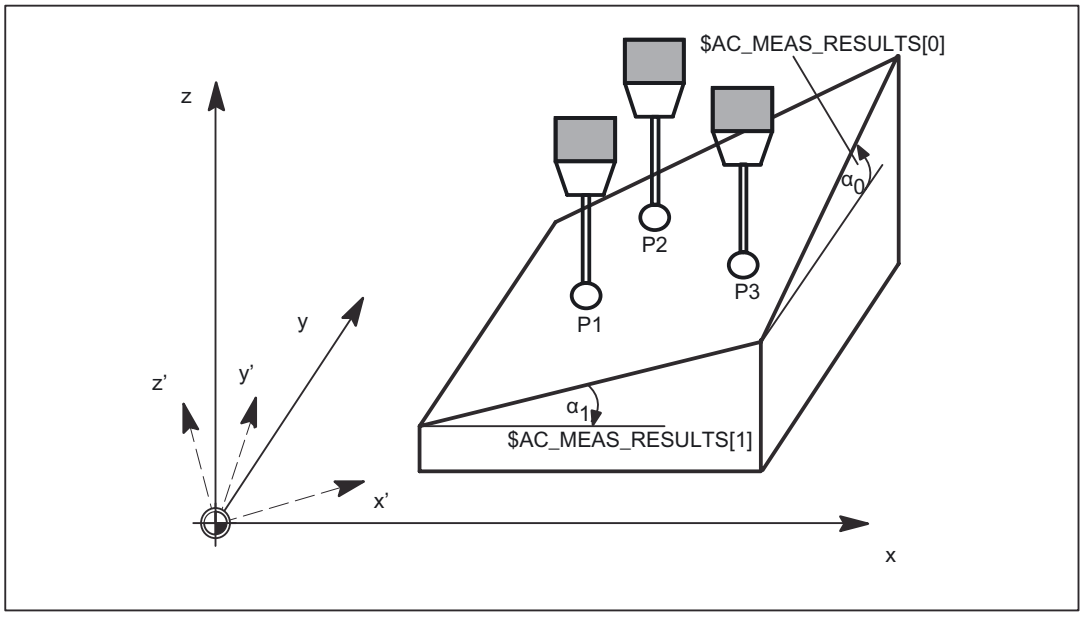


Bild 8-16 Schräge Ebene in G17

Mit $AC_MEAS_TYPE = 17 werden zwei resultierende Winkel α_0 und α_1 für die Schieflage der Ebene bestimmt und in $AC_MEAS_RESULTS[0..1] eingetragen:

- $AC_MEAS_RESULTS[0] → Drehung um die Abszisse
- $AC_MEAS_RESULTS[1] → Drehung um die Ordinate

Diese Winkel berechnen sich mit Hilfe der drei Messpunkte P1, P2 und P3. Bei diesem Messtyp wird der Winkel für die Applikate ($AC_MEAS_RESULTS[2]) immer mit 0 vorbesetzt.

Für die Abszisse und/oder für die Ordinate kann eine Soll-Drehung vorgegeben werden, die in das Ergebnisframe eingetragen werden. Wird nur ein Winkel mit einem Soll-Wert vorgegeben, so wird der zweite Winkel so berechnet, dass die drei Messpunkte auf einer schrägen Fläche mit dem Sollwinkel liegen. Im Ergebnisframe werden nur Drehungen eingetragen, der WKS-Bezugspunkt bleibt erhalten. Das WKS wird so gedreht, dass z' senkrecht auf der schrägen Ebene steht.

Für den Messtyp 17 werden folgende Ebenen-Einstellungen festgelegt:

Achsbezeichnung	G17	G18	G19
Abszisse	x-Achse	z-Achse	y-Achse
Ordinate	y-Achse	x-Achse	z-Achse
Applikate (Zustellachse)	z-Achse	y-Achse	x-Achse

Für den Messtyp 17 werden die Werte folgender Variablen ausgewertet:

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1
$AA_MEAS_POINT2[Achse]	Messpunkt 2
$AA_MEAS_POINT3[Achse]	Messpunkt 3
$AA_MEAS_SETANGLE[Achse]	Soll-Drehungen um Abszisse und Ordinate *
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_INPUT[0]	ohne Angabe werden die Punkte nicht in eine Ebene projiziert * 0: Punkte werden nicht auf eine Ebene projiziert 1: Punkte werden in die aktive Ebene, oder in die ausgewählte Ebene projiziert
$AC_MEAS_TYPE	17

* optional

Für den Messtyp 17 werden folgende Ausgangs-Variablen geschrieben:

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe
$AC_MEAS_RESULTS[0]	Winkel um Abszisse, aus den drei Messpunkten berechnet
$AC_MEAS_RESULTS[1]	Winkel um Ordinate, aus den drei Messpunkten berechnet
$AC_MEAS_RESULTS[2]	Winkel um Applikate, aus den drei Messpunkten berechnet
$AC_MEAS_RESULTS[3]	Winkel um Abszisse, der im Ergebnisframe eingetragen wird
$AC_MEAS_RESULTS[4]	Winkel um Ordinate, der im Ergebnisframe eingetragen wird
$AC_MEAS_RESULTS[5]	Winkel um Applikate, der im Ergebnisframe eingetragen wird

Beispiel

Winkel einer Ebene messen

```
DEF INT RETVAL
DEF AXIS _XX, _YY, _ZZ

T1 D1                                   ; Messtaster aktivieren
G54                                     ; alle Frames und G54 aktivieren
```

```
$AC_MEAS_VALID = 0                      ; alle Eingangswerte ungültig setzen

$AC_MEAS_TYPE = 17                      ; Messtyp für schräge Ebene setzen
$AC_MEAS_ACT_PLANE = 0                  ; Ebene für die Messung ist G17

_XX=$P_AXN1                             ; Achsen entsprechend der Ebene festlegen
_YY=$P_AXN2
_ZZ=$P_AXN3

G17 G1 _XX=10 _YY=10 F1000              ; 1. Messpunkt anfahren
MEAS = 1 _ZZ=...
$AA_MEAS_POINT1[_xx] = $AA_MW[_xx]      ; Messwert Abszisse zuweisen
$AA_MEAS_POINT1[_yy] = $AA_MW[_yy]      ; Messwert Ordinate zuweisen
$AA_MEAS_POINT1[_zz] = $AA_MW[_zz]      ; Messwert Applikate zuweisen

G1 _XX=20 _YY=10 F1000                  ; 2. Messpunkt anfahren
MEAS = 1 _ZZ=...
$AA_MEAS_POINT2[_xx] = $AA_MW[_xx]      ; Messwert Abszisse zuweisen
$AA_MEAS_POINT2[_yy] = $AA_MW[_yy]      ; Messwert Ordinate zuweisen
$AA_MEAS_POINT2[_zz] = $AA_MW[_zz]      ; Messwert Applikate zuweisen

G1 _XX=20 _YY=20 F1000                  ; 3. Messpunkt anfahren
MEAS = 1 _ZZ=...
$AA_MEAS_POINT3[_xx] = $AA_MW[_xx]      ; Messwert Abszisse zuweisen
$AA_MEAS_POINT3[_yy] = $AA_MW[_yy]      ; Messwert Ordinate zuweisen
$AA_MEAS_POINT3[_zz] = $AA_MW[_zz]      ; Messwert Applikate zuweisen

                                        ; Sollwerte für die Winkel vorgeben
$AA_MEAS_SETANGLE[_xx] = 12             ; Drehung um die Abszisse
$AA_MEAS_SETANGLE[_yy] = 4              ; Drehung um die Ordinate

$AC_MEAS_FRAME_SELECT = 102             ; Ziel-Frame auswählen - G55

$AC_MEAS_T_NUMBER = 1                   ; Werkzeug auswählen
$AC_MEAS_D_NUMBER = 1

RETVAL = MEASURE()                      ; Messberechnung starten

if RETVAL <> 0
setal(61000 + RETVAL)
endif

if $AC_MEAS_RESULTS[0] <> 12
setal(61000 + $AC_MEAS_RESULTS[0])
```

```
endif

if $AC_MEAS_RESULTS[1] <> 4
setal(61000 + $AC_MEAS_RESULTS[1])
endif

$P_UIFR[2] = $AC_MEAS_FRAME             ; Messframe in die Datenhaltung schreiben
                                          (G55)

G55 G0 AX[_xx]=10 AX[_yy]=10            ; Frame aktivieren und verfahren
m30
```

8.4.3.10 Messung um ein WKS-Bezugssystem neu definieren (Mess Type 18)

WKS'-Koordinatensystem neu definieren ($AC_MEAS_TYPE = 18)

Der Nullpunkt des neuen WKS'-Koordinatensystem wird mit dem Messpunkt P1 flächennormal auf der schrägen Ebene festgelegt.

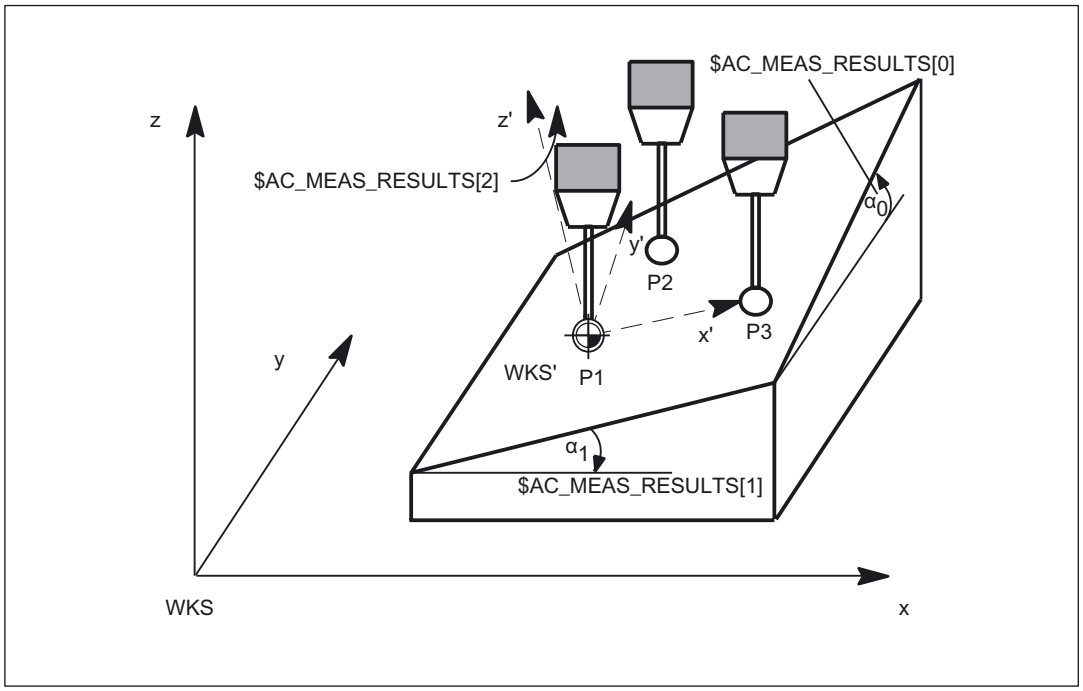


Bild 8-17 Schräge Ebene in G17

Vermessung der Ebene

Die Ebenenvermessung wird in einem Messzyklus durchgeführt. Der Zyklus nimmt die drei Messpunkte auf und versorgt das Variablen-Interface mit den notwendigen Werten.

Die Funktion MEASURE() berechnet anhand der Eingangswerte die Raumwinkel und die translatorische Verschiebung des neuen WKS'.

Transformation des Messframe

Die Ergebnisse der Berechnung, also die Raumwinkel und die Translation werden in das Messframe $AC_MEAS_FRAME eingetragen. Das Messframe wird entsprechend dem ausgewählten Frame in der Framekette transformiert. Die Raumwinkel werden zusätzlich in den Ausgangswerten $AC_MEAS_RESULTS[0..2] abgelegt. In

- $AC_MEAS_RESULTS[0] steht der Winkel um die Abszisse des alten WKS,
- $AC_MEAS_RESULTS[1] steht der Winkel um die Ordinate und in
- $AC_MEAS_RESULTS[2] steht der Winkel um die Applikate.

Neuen WKS' Nullpunkt festlegen

Der Messzyklus kann nach der Berechnung das ausgewählte Frame der Framekette mit dem Messframe beschreiben und aktivieren. Nach der Aktivierung liegt das neue WKS'-Koordinatensystem flächennormal auf der schrägen Ebene mit dem Messpunkt P1 als Nullpunkt des neuen WKS'.

Die programmierten Positionen beziehen sich dann relativ bezüglich der schrägen Ebene.

Anwendung

CAD Systeme definieren schräg im Raum liegende Flächen sehr häufig durch Angabe von drei Punkten P1, P2 und P3 auf dieser Fläche. Dabei wird der

- 1. Messpunkt P1 als neuer WKS'-Bezugspunkt verwendet, der
- 2. Messpunkt P2 gibt die Richtung der Abszisse x' des neuen gedrehten WKS'-Koordinatensystems vor und der
- 3. Messpunkt P3 wird zur Bestimmung der Raumwinkel benötigt.

Für den Messtyp 18 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1
$AA_MEAS_POINT2[Achse]	Messpunkt 2
$AA_MEAS_POINT3[Achse]	Messpunkt 3
$AA_MEAS_SETPOINT[Achse]	Soll-Position von P1 *
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_INPUT[0]	ohne Angabe werden die Punkte nicht in eine Ebene projiziert * 0: Punkte werden nicht auf eine Ebene projiziert 1: Punkte werden in die aktive Ebene, oder in die ausgewählte Ebene projiziert
$AC_MEAS_TYPE	18

* optional

Für den Messtyp 18 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Drehungen und Transformation
$AC_MEAS_RESULTS[0]	Berechneter Winkel um die Abszisse
$AC_MEAS_RESULTS[1]	Berechneter Winkel um die Ordinate
$AC_MEAS_RESULTS[2]	Berechneter Winkel um die Applikate

Beispiel

Werkstück-Koordinatensystem auf der schrägen Ebene

```
DEF INT RETVAL
DEF AXIS _XX, _YY, _ZZ

T1 D1                                   ; Messtaster aktivieren
G54                                     ; alle Frames und G54 aktivieren

$AC_MEAS_VALID = 0                      ; alle Eingangswerte ungültig setzen

$AC_MEAS_TYPE = 18                      ; Messtyp für schräge Ebene setzen
$AC_MEAS_ACT_PLANE = 0                  ; Ebene für die Messung ist G17

_XX=$P_AXN1                             ; Achsen entsprechend der Ebene festlegen
_YY=$P_AXN2
_ZZ=$P_AXN3

G17 G1 _XX=10 _YY=10 F1000              ; 1. Messpunkt anfahren
MEAS = 1 _ZZ=...

$AA_MEAS_POINT1[_xx] = $AA_MW[_xx]      ; Messwert Abszisse zuweisen
$AA_MEAS_POINT1[_yy] = $AA_MW[_yy]      ; Messwert Ordinate zuweisen
$AA_MEAS_POINT1[_zz] = $AA_MW[_zz]      ; Messwert Applikate zuweisen

G1 _XX=20 _YY=10 F1000                  ; 2. Messpunkt anfahren
MEAS = 1 _ZZ=...

$AA_MEAS_POINT2[_xx] = $AA_MW[_xx]      ; Messwert Abszisse zuweisen
$AA_MEAS_POINT2[_yy] = $AA_MW[_yy]      ; Messwert Ordinate zuweisen
$AA_MEAS_POINT2[_zz] = $AA_MW[_zz]      ; Messwert Applikate zuweisen

G1 _XX=20 _YY=20 F1000                  ; 3. Messpunkt anfahren
MEAS = 1 _ZZ=...

$AA_MEAS_POINT3[_xx] = $AA_MW[_xx]      ; Messwert Abszisse zuweisen
```

```
$AA_MEAS_POINT3[_yy] = $AA_MW[_yy]      ; Messwert Ordinate zuweisen
$AA_MEAS_POINT3[_zz] = $AA_MW[_zz]      ; Messwert Applikate zuweisen

$AA_MEAS_SETPOINT[_xx] = 10             ; Sollwerte für P1 vorgeben
$AA_MEAS_SETPOINT[_yy] = 10
$AA_MEAS_SETPOINT[_zz] = 10

$AC_MEAS_FRAME_SELECT = 102             ; Ziel-Frame auswählen - G55

$AC_MEAS_T_NUMBER = 1                   ; Werkzeug auswählen
$AC_MEAS_D_NUMBER = 1

RETVAL = MEASURE()                      ; Messberechnung starten

if RETVAL <> 0
setal(61000 + RETVAL)
endif

                                        ; Berechnungsergebnisse der Raumwinkel
                                        ; Winkel um die
R0 = $AC_MEAS_RESULTS[0]                ; Abszisse des alten WKS
R1 = $AC_MEAS_RESULTS[1]                ; Ordinate
R2 = $AC_MEAS_RESULTS[2]                ; Applikate

$P_UIFR[2] = $AC_MEAS_FRAME             ; Messframe in die Datenhaltung schreiben
                                          (G55)

G55 G0 AX[_xx]=10 AX[_yy]=10            ; Frame aktivieren und verfahren
m30
```

8.4.3.11 Messen einer 1-, 2- und 3-dimensionalen Sollwertvorgabe (Mess Type 19, 20, 21)

1-dimensionale Sollwertvorgabe ($AC_MEAS_TYPE = 19)

Bei dieser Messmethode kann genau ein Sollwert für die Abszisse, die Ordinate oder die Applikate vorgegeben werden. Werden zwei oder drei Sollwerte vorgegeben, so wird nur der erste nach der Reihenfolge Abszisse, Ordinate und Applikate genommen. Das Werkzeug bleibt dabei unberücksichtigt.

Es ist ein reines Istwertsetzen für die Abszisse, die Ordinate oder die Applikate.

Für den Messtyp 19 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für die Abszisse

Eingangs-Variable	Bedeutung
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für die Ordinate
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für die Applikate
$AA_MEAS_SETPOINT[Achse]	Sollposition der Abszisse, oder Ordinate oder Applikate
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet *
$AC_MEAS_FINE_TRANS	ohne Angabe wird in die Grob-Translation geschrieben *
$AC_MEAS_TYPE	19

* optional

Für den Messtyp 19 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Drehungen und Translation

Beispiel

1-dimensionale Sollwertvorgabe

```
DEF INT RETVAL
DEF REAL _CORMW_XX,
_CORMW_YY,
_CORMW_ZZ
DEF AXIS _XX, _YY, _ZZ
T1 D1                                   ; Messtaster aktivieren
G54                                     ; alle Frames und G54 aktivieren
$AC_MEAS_VALID = 0                      ; Alle Eingangswerte ungültig setzen
$AC_MEAS_TYPE = 19                      ; Messtyp 1-dimensionale Sollwertvorgabe
                                          setzen
$AC_MEAS_ACT_PLANE = 0                  ; Ebene für die Messung ist G17
_XX=$P_AXN1                             ; Achsen entsprechend der Ebene festlegen
_YY=$P_AXN2
_ZZ=$P_AXN3
                                        ; Messwerte zuweisen
$AA_MEAS_POINT1[_xx] = $AA_MW[_xx]      ; Messwert Abszisse zuweisen
$AA_MEAS_POINT1[_yy] = $AA_MW[_yy]      ; Messwert Ordinate zuweisen
$AA_MEAS_POINT1[_zz] = $AA_MW[_zz]      ; Messwert Applikate zuweisen
$AA_MEAS_SETPOINT[_xx] = 10             ; Sollwert für die Abszisse vorgeben
$AC_MEAS_FRAME_SELECT = 102             ; Ziel-Frame auswählen - G55
RETVAL = MEASURE()                      ; Messberechnung starten
if RETVAL <> 0
setal(61000 + RETVAL)
endif
$P_UIFR[2] = $AC_MEAS_FRAME             ; Messframe in die Datenhaltung schreiben
                                          (G55)
```

```
G55 G0 AX[_xx]=10 AX[_yy]=10          ; Frame aktivieren und verfahren
m30
```

2-dimensionale Sollwertvorgabe ($AC_MEAS_TYPE = 20)

Bei dieser Messmethode können Sollwerte für zwei Dimensionen vorgegeben werden. Es ist jede Kombination von 2 aus 3 Achsen möglich. Werden drei Sollwerte angegeben, so werden nur die Werte für die Abszisse und die Ordinate genommen. Das Werkzeug bleibt dabei unberücksichtigt.

Es ist ein reines Istwertsetzen.

Für den Messtyp 20 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für die Abszisse
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für die Ordinate
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für die Applikate
$AA_MEAS_SETPOINT[Achse]	Sollposition für die 1. Dimension
$AA_MEAS_SETPOINT[Achse]	Sollposition für die 2. Dimension
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet *
$AC_MEAS_FINE_TRANS	ohne Angabe wird in die Grob-Translation geschrieben *
$AC_MEAS_TYPE	20

* optional

Für den Messtyp 20 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Drehungen und Translation

Beispiel

2-dimensionale Sollwertvorgabe

```
DEF INT RETVAL
DEF REAL _CORMW_XX,
_CORMW_YY,
_CORMW_ZZ
DEF AXIS _XX, _YY, _ZZ
T1 D1                                 ; Messtaster aktivieren
G54                                   ; alle Frames und G54 aktivieren
$AC_MEAS_VALID = 0                    ; Alle Eingangswerte ungültig setzen
$AC_MEAS_TYPE = 20                    ; Messtyp 2-dimensionale Sollwertvorgabe
                                        setzen
$AC_MEAS_ACT_PLANE = 0                ; Ebene für die Messung ist G17
```

```
_XX=$P_AXN1                              ; Achsen entsprechend der Ebene festlegen
_YY=$P_AXN2
_ZZ=$P_AXN3
                                         ; Messwerte zuweisen
$AA_MEAS_POINT1[_xx] = $AA_MW[_xx]       ; Messwert Abszisse zuweisen
$AA_MEAS_POINT1[_yy] = $AA_MW[_yy]       ; Messwert Ordinate zuweisen
$AA_MEAS_POINT1[_zz] = $AA_MW[_zz]       ; Messwert Applikate zuweisen
$AA_MEAS_SETPOINT[_xx] = 10              ; Sollwert für die Abszisse und Ordinate
                                           vorgeben
$AA_MEAS_SETPOINT[_yy] = 10
$AC_MEAS_FRAME_SELECT = 102              ; Ziel-Frame auswählen - G55
RETVAL = MEASURE()                       ; Messberechnung starten
if RETVAL <> 0
setal(61000 + RETVAL)
endif
$P_UIFR[2] = $AC_MEAS_FRAME              ; Messframe in die Datenhaltung schreiben
                                           (G55)
G55 G0 AX[_xx]=10 AX[_yy]=10             ; Frame aktivieren und verfahren
m30
```

3-dimensionale Sollwertvorgabe ($AC_MEAS_TYPE = 21)

Bei dieser Messmethode kann ein Sollwert für die Abszisse, Ordinate und die Applikate vorgegeben werden. Das Werkzeug bleibt dabei unberücksichtigt.

Es ist ein reines Istwertsetzen für die Abszisse, Ordinate und die Applikate.

Für den Messtyp 21 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für Abszisse
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für Ordinate
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für Applikate
$AA_MEAS_SETPOINT[Achse]	Sollposition für die Abszisse
$AA_MEAS_SETPOINT[Achse]	Sollposition für die Ordinate
$AA_MEAS_SETPOINT[Achse]	Sollposition für die Applikate
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet *
$AC_MEAS_FINE_TRANS	ohne Angabe wird in die Grob-Translation geschrieben *
$AC_MEAS_TYPE	21

* optional

Für den Messtyp 21 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Drehungen und Translation

Beispiel

3-dimensionale Sollwertvorgabe

```
DEF INT RETVAL
DEF REAL _CORMW_XX,
_CORMW_YY,
_CORMW_ZZ
DEF AXIS _XX, _YY, _ZZ
T1 D1                                   ; Messtaster aktivieren
G54                                     ; alle Frames und G54 aktivieren
$AC_MEAS_VALID = 0                      ; Alle Eingangswerte ungültig setzen
$AC_MEAS_TYPE = 21                      ; Messtyp 3-dimensionale Sollwertvorgabe
                                          setzen
$AC_MEAS_ACT_PLANE = 0                  ; Ebene für die Messung ist G17
_XX=$P_AXN1                             ; Achsen entsprechend der Ebene festlegen
_YY=$P_AXN2
_ZZ=$P_AXN3
                                        ; Messwerte zuweisen
$AA_MEAS_POINT1[_xx] = $AA_MW[_xx]      ; Messwert Abszisse zuweisen
$AA_MEAS_POINT1[_yy] = $AA_MW[_yy]      ; Messwert Ordinate zuweisen
$AA_MEAS_POINT1[_zz] = $AA_MW[_zz]      ; Messwert Applikate zuweisen
$AA_MEAS_SETPOINT[_xx] = 10             ; Sollwert für die Abszisse, Ordinate u.
                                          Applikate
$AA_MEAS_SETPOINT[_yy] = 10             ; vorgeben
$AA_MEAS_SETPOINT[_zz] = 10
$AC_MEAS_FRAME_SELECT = 102             ; Ziel-Frame auswählen - G55
$AA_MEAS_SETPOINT[_yy] = 10
RETVAL = MEASURE()                      ; Messberechnung starten
if RETVAL <> 0
setal(61000 + RETVAL)
endif
$P_UIFR[2] = $AC_MEAS_FRAME             ; Messframe in die Datenhaltung schreiben
                                          (G55)
G55 G0 AX[_xx]=10 AX[_yy]=10            ; Frame aktivieren und verfahren
m30
```

8.4.3.12 Messen eines schrägen Winkel (Mess Type 24)

Messmethode zur Umrechnung eines Messpunktes in ein beliebiges Koordinatensystem

Koordinatentransformation einer Position ($AC_MEAS_TYPE = 24)

Bei dieser Messmethode kann ein Messpunkt, der in einem beliebigen Koordinatensystem (WKS, BKS, MKS) vorliegt durch eine Koordinatentransformation bezüglich eines neuen Koordinatensystems umgerechnet werden.

Das neue Koordinatensystem wird durch die Angabe einer gewünschten Framekette gebildet.

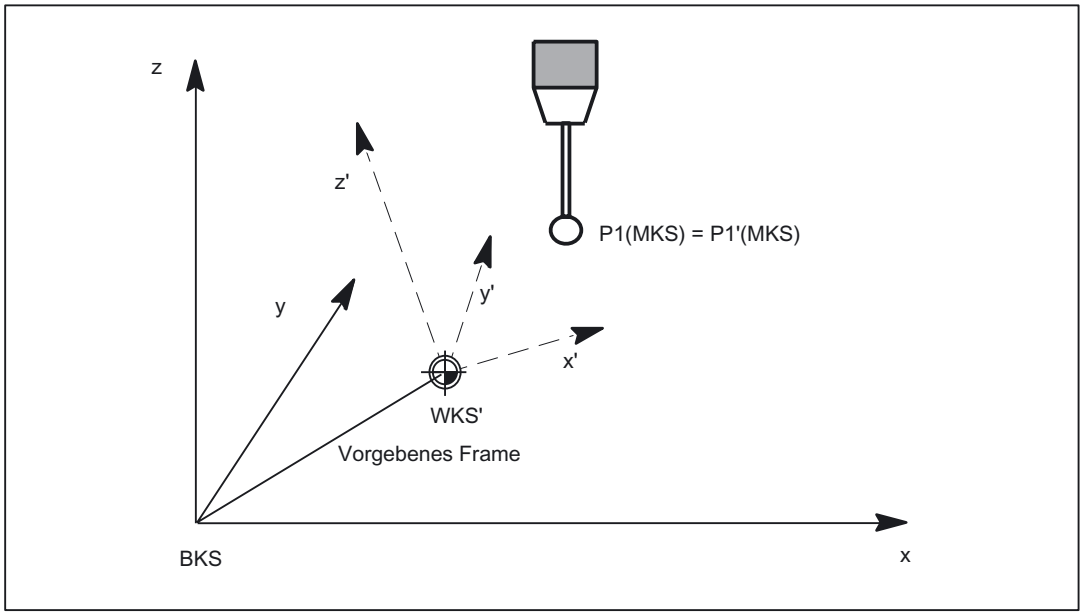


Bild 8-18 Koordinatentransformation einer Position

Für den Messtyp 24 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Position. die transformiert werden soll
$AC_MEAS_P1:COORD	Standard ist 0: WKS, 1: BKS, 2: MKS *
$AC_MEAS_P2_COORD	Ziel-Koordinatensystem *
$AC_MEAS_TOOL_MASK	0x20; Länge des aktiven Werkzeuges geht bei der Koordinatentransformation einer Position mit ein *
$AC_MEAS_CHSFR	Systemframes aus der Datenhaltung *
$AC_MEAS_NCBFR	globale Basisframes aus der Datenhaltung *
$AC_MEAS_CHBFR	Kanal-Basisframes aus der Datenhaltung *
$AC_MEAS_UIFR	einstellbarer Frame aus der Datenhaltung *
$AC_MEAS_PFRAME	1: programmierbarer Frame wird nicht verrechnet *
$AC_MEAS_TYPE	24

* optional

Für den Messtyp 24 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_POINT2[Achse]	Umgerechnete Achspositionen

Beispiel

Koordinatentransformation WKS einer gemessenen Position

```
DEF INT RETVAL
DEF INT LAUF
DEF REAL _CORMW_xx, _CORMW_yy, _CORMW_zz
DEF AXIS _XX, _YY, _ZZ

$TC_DP1[1,1]=120                        ; Werkzeugtyp Schaftfräser
$TC_DP2[1,1]=20
$TC_DP3[1,1]=0                          ; (z)Längenkorrekturvektor
$TC_DP4[1,1]=0                          ; (y)Längenkorrekturvektor
$TC_DP5[1,1]=0                          ; (x)Längenkorrekturvektor
$TC_DP6[1,1]=2                          ; Radius

T1 D1                                   ; Messtaster aktivieren

G17                                     ; schräge Ebene G17

_xx=$P_AXN1 _yy=$P_AXN2 _zz=$P_AXN3     ; Achsen entsprechend der Ebene festlegen

                                        ; Gesamtframe ergibt sich zu
                                          CTRANS(_xx,10,_yy,-1,_zz,5,A,6,B,7)
$P_CHBFR[0]=CTRANS(_zz,5,A,6) : CROT(_zz,45)
$P_UIFR[1]=CTRANS( )
$P_UIFR[1,_xx,TR]=-SIN(45)
$P_UIFR[1,_yy,TR]=-SIN(45)
$P_UIFR[2]=CTRANS( )
$P_PFRAME=CROT(_zz,-45)
$P_CYCFR=CTRANS(_xx,10,B,7)

G54                                     ; alle Frames und G54 aktivieren

G0 X0 Y0 Z0 A0 B0 F1000

$AC_MEAS_VALID = 0                      ; alle Eingangswerte ungültig setzen

$AC_MEAS_TYPE = 24                      ; Messtyp für Koordinatentransformation
                                          setzen

$AC_MEAS_ACT_PLANE = 0                  ; Ebene für die Messung ist G17

                                        ; Messwerte zuweisen
$AA_MEAS_POINT1[_xx] = $AA_IW[_xx]      ; Messwert Abszisse zuweisen
$AA_MEAS_POINT1[_yy] = $AA_IW[_yy]      ; Messwert Ordinate zuweisen
```

```
$AA_MEAS_POINT1[_zz] = $AA_IW[_zz]      ; Messwert Applikate zuweisen
$AA_MEAS_POINT1[A] = $AA_IW[A]
$AA_MEAS_POINT1[B] = $AA_IW[B]

$AC_MEAS_P1_COORD=0                     ; Umrechnung einer Position vom WKS in WKS'
$AC_MEAS_P2_COORD=0
                                        ; WKS einstellen
                                        ; Gesamtframe ergibt sich aus
                                          CTRANS(_xx,0,_yy,0,_zz,5,A,6,B,0)

                                        ; Zyklenframe ausschalten
$AC_MEAS_CHSER=$MC_MM_SYSTEM_FRAME_MASK B_AND 'B1011111'

$AC_MEAS__NCBFR='B0'                    ; globale Basisframe ausschalten
$AC_MEAS__CHBFR='B1'                    ; Kanal-Basisframe 1 aus Datenhaltung
$AC_MEAS__UIFR=2                        ; einstellbares Frame G55 aus Datenhaltung

$AA_MEAS_PFRAME=1                       ; Pogrammierbares Frame nicht einrechnen

RETVAL = MEASURE()                      ; Messberechnung starten

if RETVAL <> 0
setal(61000 + RETVAL)
endif

if $AA_MEAS_PIONT2[_xx] <> 10
setal(61000)
M0
stopre
endif
if $AA_MEAS_PIONT2[_yy] <> -1
setal(61000)
M0
stopre
if $AA_MEAS_PIONT2[_zz] <> 0
setal(61000)
M0
stopre
if $AA_MEAS_PIONT2[A] <> 0
setal(61000)
M0
stopre
if $AA_MEAS_PIONT2[B] <> 7
setal(61000)
M0
```

```
stopre

m30
```

8.4.3.13 Messen eines Rechtecks (Mess Type 25)

Messmethode zur Ermittlung eines Rechtecks ($AC_MEAS_TYPE = 25)

Zur Ermittlung eines Rechtecks Werkzeugabmessungen in den Arbeitsebenen

- G17 Arbeitsebene x/y Zustellrichtung z
- G18 Arbeitsebene z/x Zustellrichtung y
- G19 Arbeitsebene y/z Zustellrichtung x

sind pro Rechteck 4 Messpunkte erforderlich.

Die Messpunkte können in beliebiger Reihenfolge angegeben werden. Die Messpunkte mit dem größten Ordinatenabstand entsprechen den Punkten P3 und P4.

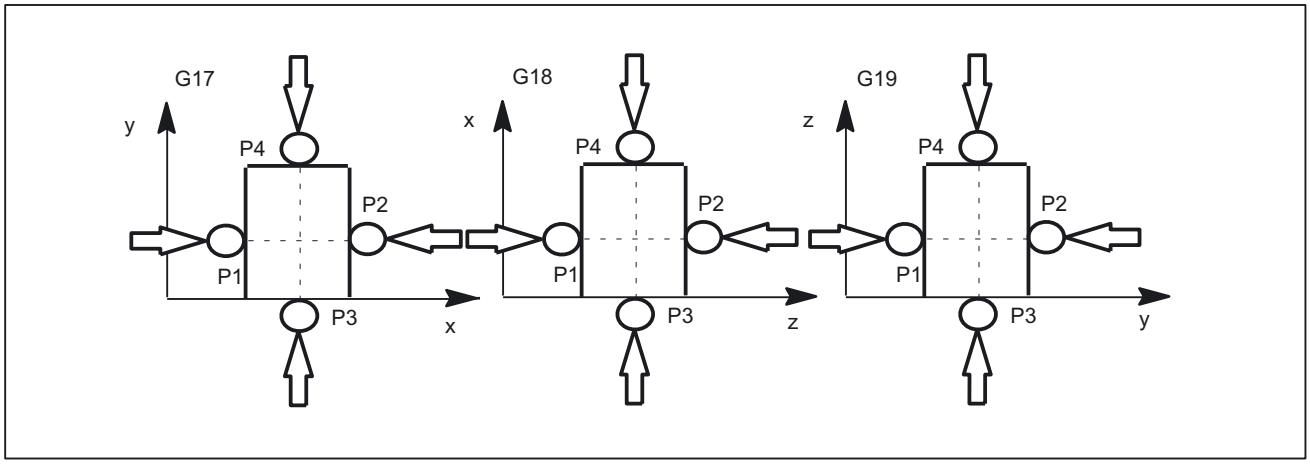


Bild 8-19 Rechteckermittlung mit Zustellung in der Arbeitsebene G17, G18 und G19

Für den Messtyp 25 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1
$AA_MEAS_POINT2[Achse]	Messpunkt 2
$AA_MEAS_POINT3[Achse]	Messpunkt 3
$AA_MEAS_POINT4[Achse]	Messpunkt 4
$AA_MEAS_SETPOINT[Achse]	Soll-Position der Stegmitte *
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FINE_TRANS	0: Grobverschiebung, 1: Feinverschiebung *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *

Eingangs-Variable	Bedeutung
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_INPUT[0]	ohne Angabe für die Außenecke * =0: Messung für Außenecke =1: Messung für Innenecke
$AC_MEAS_TYPE	25

* optional

Für den Messtyp 25 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnisframe mit Translation
$AC_MEAS_RESULTS[0]	Abszisse des berechneten Mittelpunktes
$AC_MEAS_RESULTS[1]	Ordinate des berechneten Mittelpunktes
$AC_MEAS_RESULTS[2]	Applikate des berechneten Mittelpunktes
$AC_MEAS_RESULTS[3]	Breite des Rechteckes P1/P2
$AC_MEAS_RESULTS[4]	Länge des Rechteckes P3/P4

8.4.3.14 Messung zum Sichern von Datenhaltungsframes (Mess Type 26)

Save von Datenhaltungsframes ($AC_MEAS_TYPE = 26)

Mit diesem Messtyp besteht die Möglichkeit alle oder auch ausgewählte Datenhaltungsframes mit der aktuellen Wertebelegung in ein File zu sichern. Es wird dabei unterschieden, ob dies über die Kommandobearbeitung oder über das Teileprogramm erfolgte. Die Funktion kann auch aus verschiedenen Kanälen aktiviert werden. Die Files werden im Directory _N_SYF_DIR angelegt.

Ein Restore löscht die Datensicherung und ein erneutes Sichern überschreibt die vorherige Sicherung. Die zuletzt gesicherten Daten können dann durch ein zweites Save mit

- $AC_MEAS_CHSFR = 0 Systemframes;
- $AC_MEAS_NCBFR = 0 globale Basisframes;
- $AC_MEAS_CHBFR = 0 Kanal-Basisframes;
- $AC_MEAS_UIFR = 0 Anzahl einstellbare Frames

aus der Datenhaltung gelöscht werden.

Hinweis

Beim Sichern von allen Datenhaltungsframes ist zu beachten, dass pro Frame ca. 1 kByte Speicher benötigt wird. Ist nicht genügend Speicher vorhanden, so wird der Vorgang mit der Fehlermeldung MEAS_NO_MEMORY abgebrochen. Über das folgende Maschinendatum kann die erforderliche Größe im DRAM verändert werden:

MD18351 $MM_DRAM_FILE_MEM_SIZE

Für den Messtyp 26 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AC_MEAS_CHSFR	Bitmaske Systemframes aus der Datenhaltung. * Wird die Variable nicht geschrieben, dann werden alle Systemframes gesichert.
$AC_MEAS_NCBFR	Bitmaske globale Basisframes aus der Datenhaltung. * Wird die Variable nicht geschrieben, dann werden alle globalen Basisframes gesichert.
$AC_MEAS_CHBFR	Bitmaske Kanal-Basisframes aus der Datenhaltung. * Wird die Variable nicht geschrieben, dann werden alle Kanal-Basisframes gesichert.
$AC_MEAS_UIFR	Anzahl einstellbarer Frames aus der Datenhaltung. * 0..100: 1: G500 2: G500, G54. Wird die Variable nicht geschrieben, dann werden alle einstellbaren Frames gesichert.
$AC_MEAS_TYPE	26

8.4.3.15 Messung zum Zurückschreiben gesicherter Datenhaltungsframes (Mess Type 27)

Restore von zuletzt gesicherten Datenhaltungsframes ($AC_MEAS_TYPE = 27)

Mit diesem Messtyp besteht die Möglichkeit die durch den Messtyp 26 gesicherten Datenhaltungsframes wieder ins SRAM zurückzuschreiben.

Es können alle zuletzt gesicherten Frames oder auch nur ausgewählte Frames zurückgeschrieben werden. Wird ein Frame ausgewählt, der nicht gesichert wurde, dann wird dies ignoriert und führt zu keinen Abbruch.

Für den Messtyp 27 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AC_MEAS_CHSFR	Bitmaske Systemframes aus der Datenhaltung. * Wird die Variable nicht geschrieben, dann werden alle Systemframes restored.
$AC_MEAS_NCBFR	Bitmaske globale Basisframes aus der Datenhaltung. * Wird die Variable nicht geschrieben, dann werden alle globalen Basisframes restored.
$AC_MEAS_CHBFR	Bitmaske Kanal-Basisframes aus der Datenhaltung. * Wird die Variable nicht geschrieben, dann werden alle Kanal-Basisframes restored.
$AC_MEAS_UIFR	Anzahl einstellbarer Frames aus der Datenhaltung. * Bereich von 1: G54 bis G99: G599. Wird die Variable nicht geschrieben, dann werden alle einstellbaren Frames restored.
$AC_MEAS_TYPE	27

* optional

8.4.3.16 Messung zur Vorgabe einer additiven Drehung für Kegeldrehen (Mess Type 28)

Kegeldrehen

Additive Drehung der Ebene ($AC_MEAS_TYPE = 28)

Der Messtyp 28 wird über die Bedienoberfläche ManualTurn-Advanced für die Anwendung Kegeldrehen genutzt. Es kann damit eine additive Drehung der aktiven oder einer bestimmten Ebene um einen Winkel im Bereich von α = +/- 90° vorgegeben werden. Die Drehung erfolgt um die zur Ebene senkrecht stehenden Koordinatenachse.

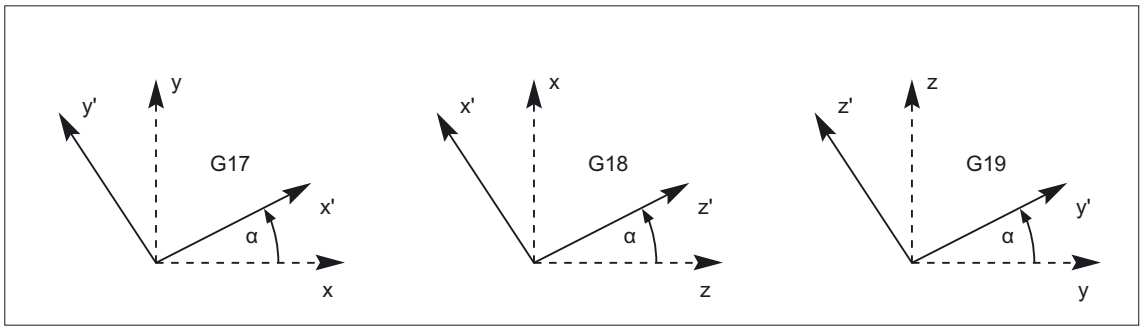


Bild 8-20 Drehung der Ebene G17, G18 und G19 um Winkel α = +30°

Anwendung

Beim Kegeldrehen wird die aktive Ebene um den Kegelwinkel gedreht, wobei die Drehung in das aktive Zyklenframe geschrieben wird. Bei RESET wird das Zyklenframe gelöscht. Eine erneute Aktivierung ist eventuell notwendig. Die Auswahl des Zyklenframe erfolgt bedingt durch die ENS-Positionsanzeige. Wird nach Aktivierung der Drehung z. B. bei aktiver Ebene G18 in Richtung z' verfahren, so verändern sich **gleichzeitig** die Istpositionen der entsprechenden Achsen für **x und z**

Drehungen bei aktiver Ebene G17 und G18 verhalten sich anlog und sind im oberen Bild dargestellt. Für den Messtyp 28 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AC_MEAS_WP_SETANGLE	Soll-Winkel
$AC_MEAS_ACT_PLANE	ohne Angabe wird um die aktive Ebene gedreht. *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet. *
$AC_MEAS_INPUT[0]	1: Kegeldrehen ist aktiv. *
$AC_MEAS_TYPE	28

* optional

Für den Messtyp 28 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_FRAME	Ergebnis mit Drehung

8.4.4 Werkzeugvermessung

Aus der einzugebenen Werkzeuglänge berechnet die Steuerung den Abstand der Werkzeugspitze vom Werkzeugträgerbezugspunkt T.

Zur Werkzeugvermessung eines eingewechselten Werkzeuges an einer Dreh- oder Fräsmaschine sind folgende Messtypen verfügbar:

Messtypen	Werkzeugvermessung
$AC_MEAS_TYPE = 10	Werkzeuglänge an einem bereits vermessenen Referenzteil
$AC_MEAS_TYPE = 11	Werkzeugdurchmesser an einem bereits vermessenen Referenzteil
$AC_MEAS_TYPE = 22	Werkzeugdurchmesser an Maschinen mit einer Lupe (ShopTurn)
$AC_MEAS_TYPE = 23	Werkzeuglängen mit gemerkter oder aktueller Positionen (ShopTurn) Messen einer Werkzeuglänge zweier Werkzeuge mit Orientierung: Zwei Drehwerkzeuge mit eigenem Referenzpunkt bei Werkzeugorientierung in Anfahrrichtung einem Referenzpunkt bei gegenläufiger Werkzeuglage zur Anfahrrichtung und Werkzeugorientierung. Zwei Fräser mit eigenem Referenzpunkt bei Werkzeugorientierung in -y mit einem Referenzpunkt bei Werkzeugorientierung in -y und gegenläufiger Werkzeuglage zur Anfahrrichtung. Zwei um 90 Grad gedrehte Fräser mit eigenem Referenzpunkt bei Werkzeugorientierung in Anfahrrichtung einem Referenzpunkt bei gegenläufiger Werkzeuglage zur Anfahrrichtung und Werkzeugorientierung.

8.4.5 Messtypen der Werkzeugvermessung

8.4.5.1 Messen der Werkzeuglänge (Mess Type 10)

Werkzeuglängenmessung an einem vermessen Referenzteil ($AC_MEAS_TYPE = 10)

Die Werkzeuglänge kann bei der Werkzeugvermessung an einem bereits vermessenen Referenzteiles erfolgen. Je nach Lage des Werkzeugs ist eine Ebenenanwahl bei G17 für Werkzeuglage in z-Richtung, bei G18 für Werkzeuglage in y-Richtung, bei G19 für Werkzeuglage in x-Richtung möglich.

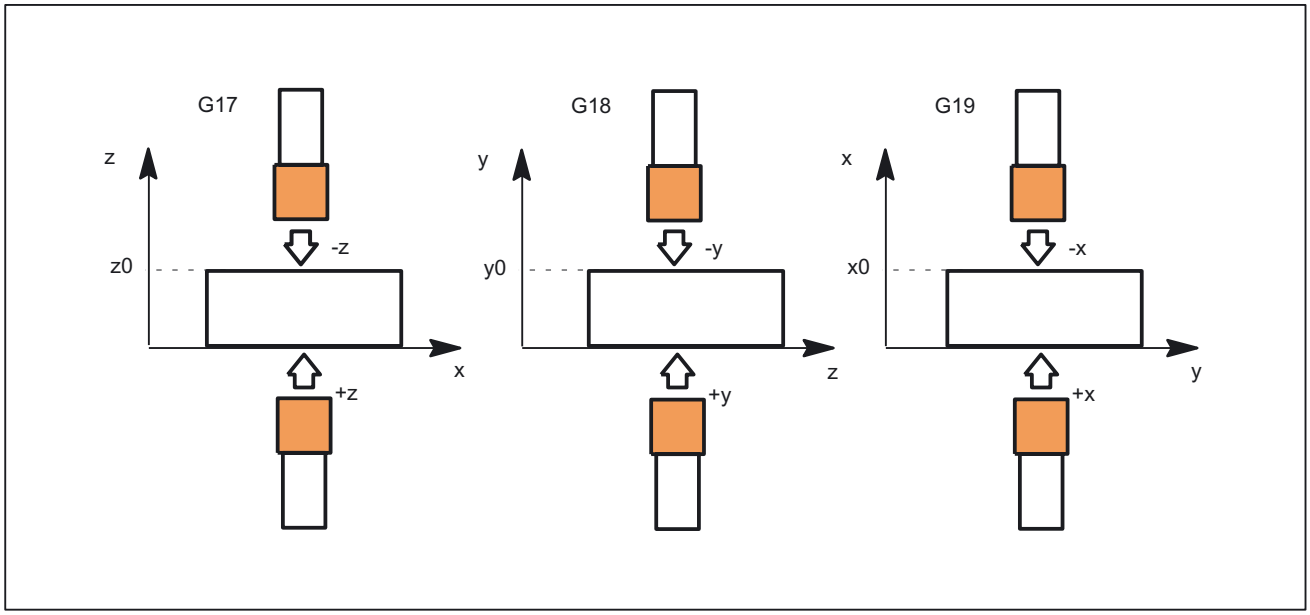


Bild 8-21 Werkzeuglängenvermessung für die ausgewählte Ebene G17, G18 und G19

Für den Messtyp 10 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1
$AC_MEAS_P1_COORD	Koordinatensystem des Messpunktes *
$AA_MEAS_SETPOINT[Achse]	Soll-Position z0
$AC_MEAS_SET_COORD	Koordinatensystem des Sollpunktes *
$AC_MEAS_DIR_APPROACH	0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_TYPE	10

* optional

Für den Messtyp 10 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_TOOL_LENGTH	Werkzeuglänge
$AC_MEAS_RESULTS[0]	Werkzeuglänge in x
$AC_MEAS_RESULTS[1]	Werkzeuglänge in y
$AC_MEAS_RESULTS[2]	Werkzeuglänge in z
$AC_MEAS_RESULTS[3]	Werkzeuglänge L1
$AC_MEAS_RESULTS[4]	Werkzeuglänge L2
$AC_MEAS_RESULTS[5]	Werkzeuglänge L3

Beispiel

Messung der Werkzeuglänge

```
DEF INT RETVAL
T0 D0
g0 x0 y0 z0 f10000
                                        ; Werkzeuglänge vermessen
$AC_MEAS_VALID = 0                      ; Alle Eingangswerte ungültig setzen
g1 z10                                  ; Werkzeug auf Referenzteil fahren

$AC_MEAS_LATCH[0] = 1                   ; Messpunkt 1 ablatchen
$AC_MEAS_DIR_APPROACH = 5               ; Anfahrrichtung -z setzen
$AA_MEAS_SETPOINT[x] = 0                ; Referenzposition setzen
$AA_MEAS_SETPOINT[y] = 0
$AA_MEAS_SETPOINT[z] = 0
$AC_MEAS_ACT_PLANE = 0                  ; Ebene für die Messung ist G17
$AC_MEAS_T_NUMBER = 0                   ; Es ist kein Werkzeug ausgewählt
$AC_MEAS_D_NUMBER = 0
$AC_MEAS_TYPE = 10                      ; Messtyp auf Werkzeuglänge setzen

RETVAL = MEASURE()                      , Messvorgang starten
if RETVAL <> 0 setal(61000 + RETVAL)
endif

if $AC_MEAS_TOOL_LENGTH <> 10           ; bekannte Werkzeuglänge abfragen
setal(61000 + $AC_MEAS_TOOL_LENGTH)
endif
m30
```

8.4.5.2 Messen des Werkzeugdurchmessers (Mess Type 11)

Werkzeugdurchmessermessung an einem Referenzteil ($AC_MEAS_TYPE = 11)

Der Werkzeugdurchmesser kann bei der Werkzeugvermessung an einem bereits vermessenen Referenzteiles erfolgen. Je nach Lage des Werkzeugs ist eine Ebenenanwahl bei G17 für Werkzeuglage in z-Richtung, bei G18 für Werkzeuglage in y-Richtung und bei G19 für Werkzeuglage in x-Richtung möglich.

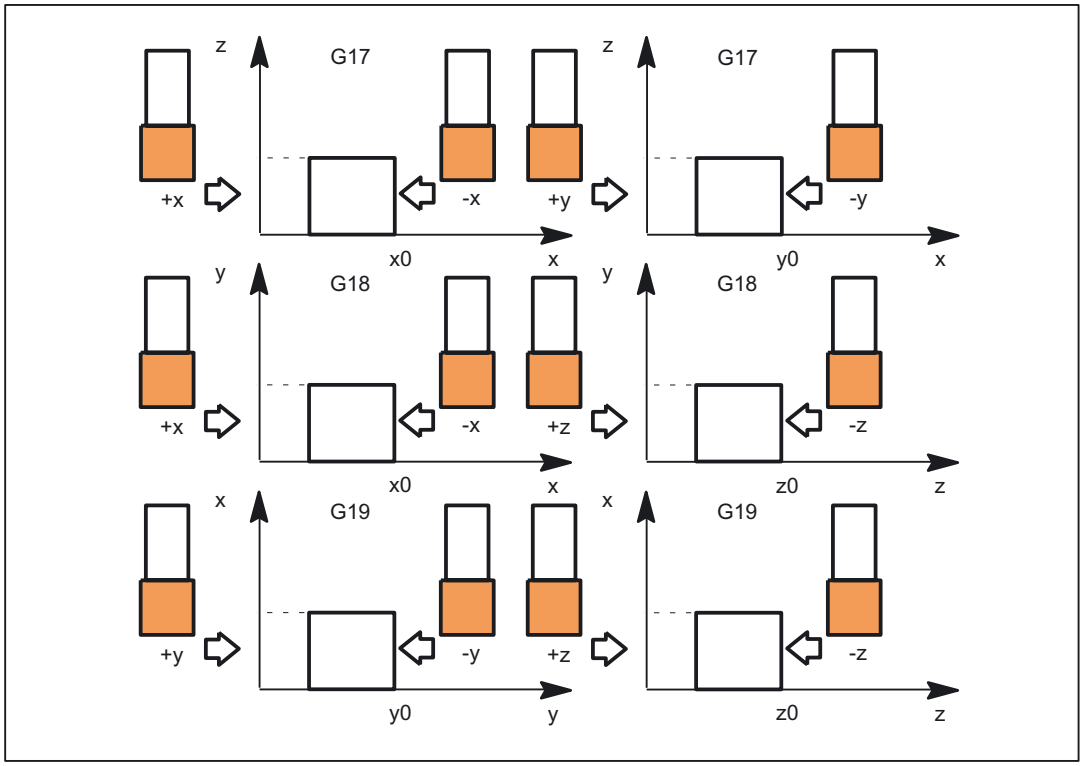


Bild 8-22 Werkzeugdurchmesser für die ausgewählte Ebene G17, G18 und G19

Für den Messtyp 11 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1
$AA_MEAS_SETPOINT[Achse]	Soll-Position x0
$AC_MEAS_DIR_APPROACH	0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_TYPE	11

* optional

Für den Messtyp 11 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_TOOL_DIAMETER	Werkzeugdurchmesser

8.4.5.3 Messen von Werkzeuglängen mit Lupe (Mess Type 22)

Werkzeuglänge mit Lupe

Werkzeuglängenmessung mit Lupe ($AC_MEAS_TYPE = 22)

Zur Ermittlung der Werkzeugabmessungen, kann falls an der Maschine vorhanden, auch eine Lupe verwendet werden.

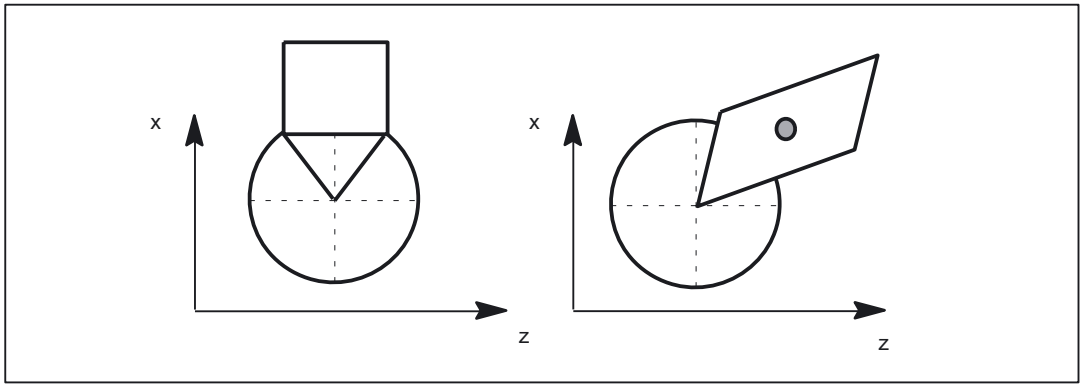


Bild 8-23 Messen von Werkzeuglängen mit Lupe

Für den Messtyp 22 werden die Werte folgender Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Messpunkt 1 für alle Kanalachsen
$AC_MEAS_P1_COORD	Koordinatensystem des Messpunktes *
$AA_MEAS_SETPOINT[Achse]	Lupenposition x und z müssen vorgegeben werden
$AC_MEAS_SET_COORD	Koordinatensystem des Sollpunktes *
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_FRAME_SELECT	ohne Angabe wird additives Frame berechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_TYPE	22

* optional

Für den Messtyp 22 werden folgende Ausgangs-Variablen geschrieben

Ausgangs-Variable	Bedeutung
$AC_MEAS_RESULT[0]	Werkzeuglänge in x
$AC_MEAS_RESULT[1]	Werkzeuglänge in y
$AC_MEAS_RESULT[2]	Werkzeuglänge in z
$AC_MEAS_RESULT[3]	Werkzeuglänge L1
$AC_MEAS_RESULT[4]	Werkzeuglänge L2
$AC_MEAS_RESULT[5]	Werkzeuglänge L3

8.4.5.4 Messen einer Werkzeuglänge mit gemerkter oder aktueller Position (Mess Type 23)

Werkzeuglängen mit gemerkter/aktueller Position

Werkzeuglängenmessung mit gemerkter oder aktueller Position ($AC_MEAS_TYPE = 23)

Beim manuellen Messen können die Werkzeugabmessungen in X- und Z-Richtung ermittelt werden. Aus der bekannten Position des

- Werkzeugträgerbezugspunkts und den
- Werkstückabmessungen

berechnet ShopTurn die Werkzeugkorrekturdaten.

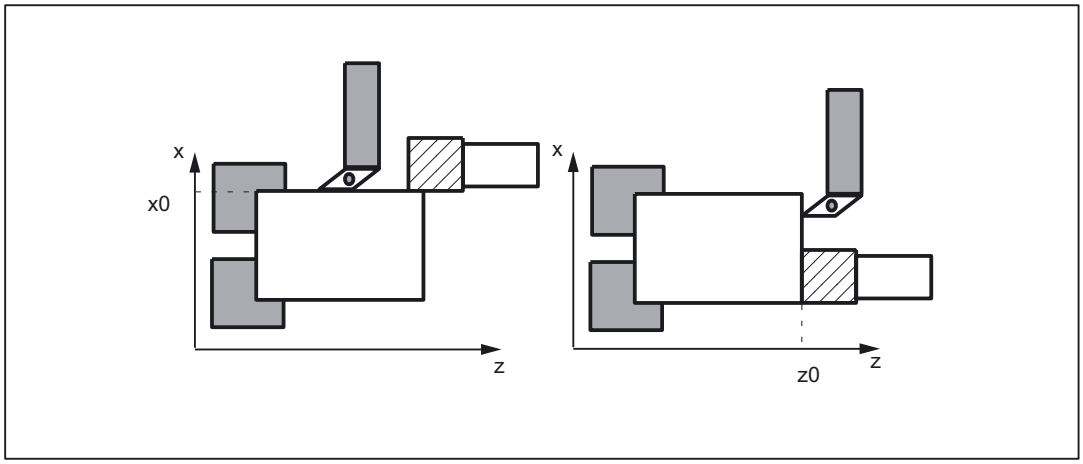


Bild 8-24 Messen einer Werkzeuglänge mit gemerkter oder aktueller Position

Für den Messtyp 23 werden die Werte folgender Eingangs-Variablen ausgewertet

Eingangs-Variable	Bedeutung
$AC_MEAS_VALID	Gültigkeitsbits für die Eingangsvariablen
$AA_MEAS_POINT1[Achse]	Aktuelle oder gemerkte Position
$AC_MEAS_P1_COORD	Koordinatensystem des Messpunktes *
$AA_MEAS_SETPOINT[Achse]	Sollposition (mindestens eine Geo-Achse muss vorgegeben werden)

Eingangs-Variable	Bedeutung
$AC_MEAS_SET_COORD	Koordinatensystem des Sollpunktes *
$AC_MEAS_ACT_PLANE	ohne Angabe wird mit aktiver Ebene gerechnet *
$AC_MEAS_T_NUMBER	ohne Angabe wird mit aktivem T gerechnet (T0) *
$AC_MEAS_D_NUMBER	ohne Angabe wird mit aktivem D gerechnet (D0) *
$AC_MEAS_TOOL_MASK	Werkzeuglage, Radius *
$AC_MEAS_DIR_APPROACH	Anfahrrichtung *
$AC_MEAS_INPUT[0] = 1	die berechneten Werkzeuglängen werden in die Datenhaltung geschrieben *
$AC_MEAS_TYPE	23

* optional

Für den Messtyp 23 werden folgende Ausgangs-Variablen geschrieben

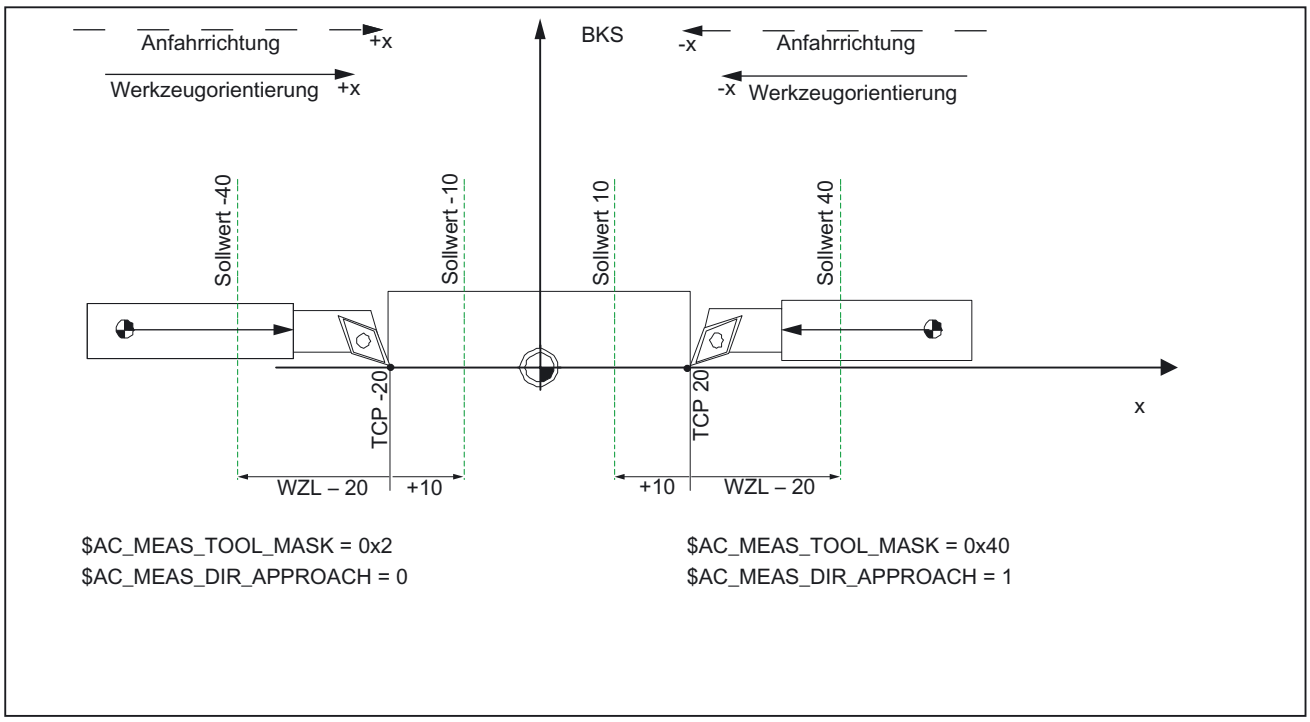
Ausgangs-Variable	Bedeutung
$AC_MEAS_RESULT[0]	Werkzeuglänge in x
$AC_MEAS_RESULT[1]	Werkzeuglänge in y
$AC_MEAS_RESULT[2]	Werkzeuglänge in z
$AC_MEAS_RESULT[3]	Werkzeuglänge L1
$AC_MEAS_RESULT[4]	Werkzeuglänge L2
$AC_MEAS_RESULT[5]	Werkzeuglänge L3

8.4.5.5 Messen einer Werkzeuglänge zweier Werkzeuge mit Orientierung

Werkzeugorientierung

Werkzeuge, deren Orientierung zur Werkzeugaufnahme zeigt, müssen durch $AC_MEAS_TOOL_MASK = 0x200 gekennzeichnet werden. Die berechneten Werkzeuglängen werden dann negativ eingerechnet.

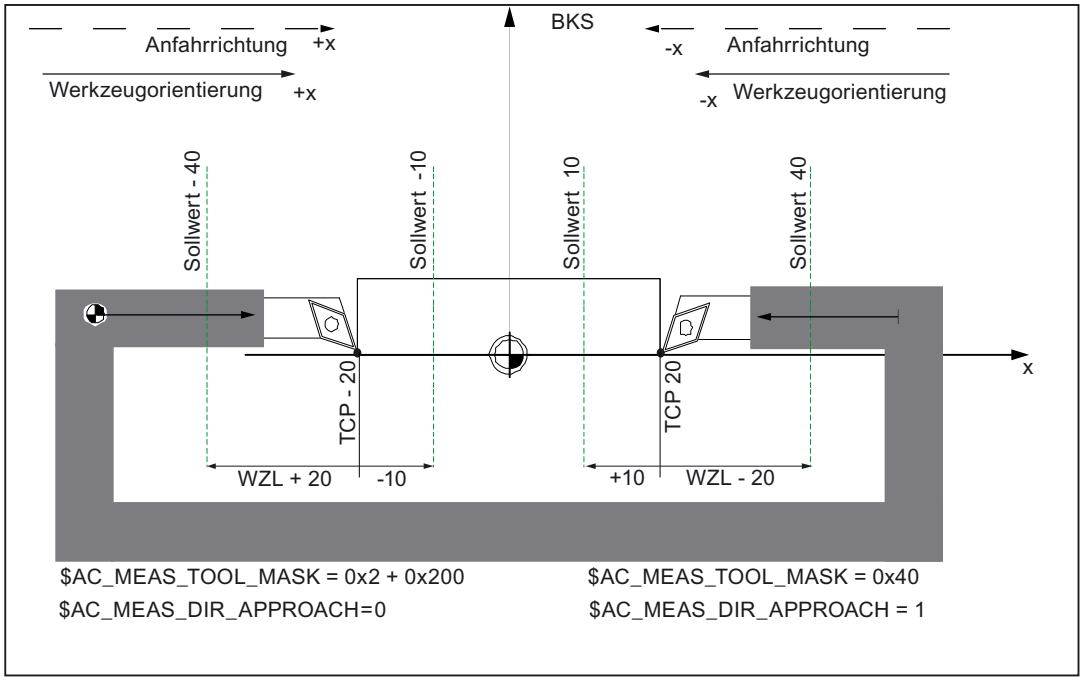
Zwei Drehwerkzeuge mit eigenem Referenzpunkt bei Werkzeugorientierung in Anfahrrichtung



Bei Werkzeuglage zweier Drehwerkzeuge mit eigenem Referenzpunkt werden zusätzlich zu den Eingangsvariablen von Messtype 23 die Werte folgender Eingangs-Variablen ausgewertet

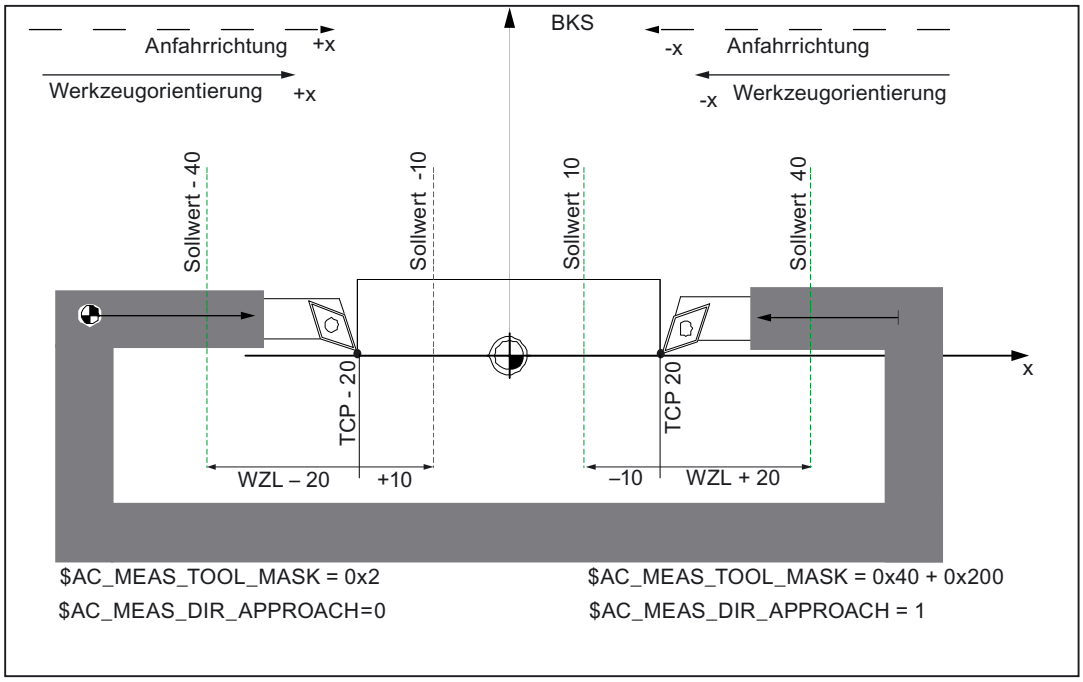
Anfahrrichtung und Werkzeugorientierung +x	Anfahrrichtung und Werkzeugorientierung -x
$AC_MEAS_TOOL_MASK = 0x2	$AC_MEAS_TOOL_MASK = 0x40
$AC_MEAS_DIR_APPROACH = 0	$AC_MEAS_DIR_APPROACH = 1

Zwei Drehwerkzeuge mit einem Referenzpunkt bei gegenläufiger Werkzeuglage zur Orientierung



Bei Werkzeuglage zweier Drehwerkzeuge mit einem Referenzpunkt werden zusätzlich zu den Eingangsvariablen von Messtype 23 die Werte folgender Eingangs-Variablen ausgewertet

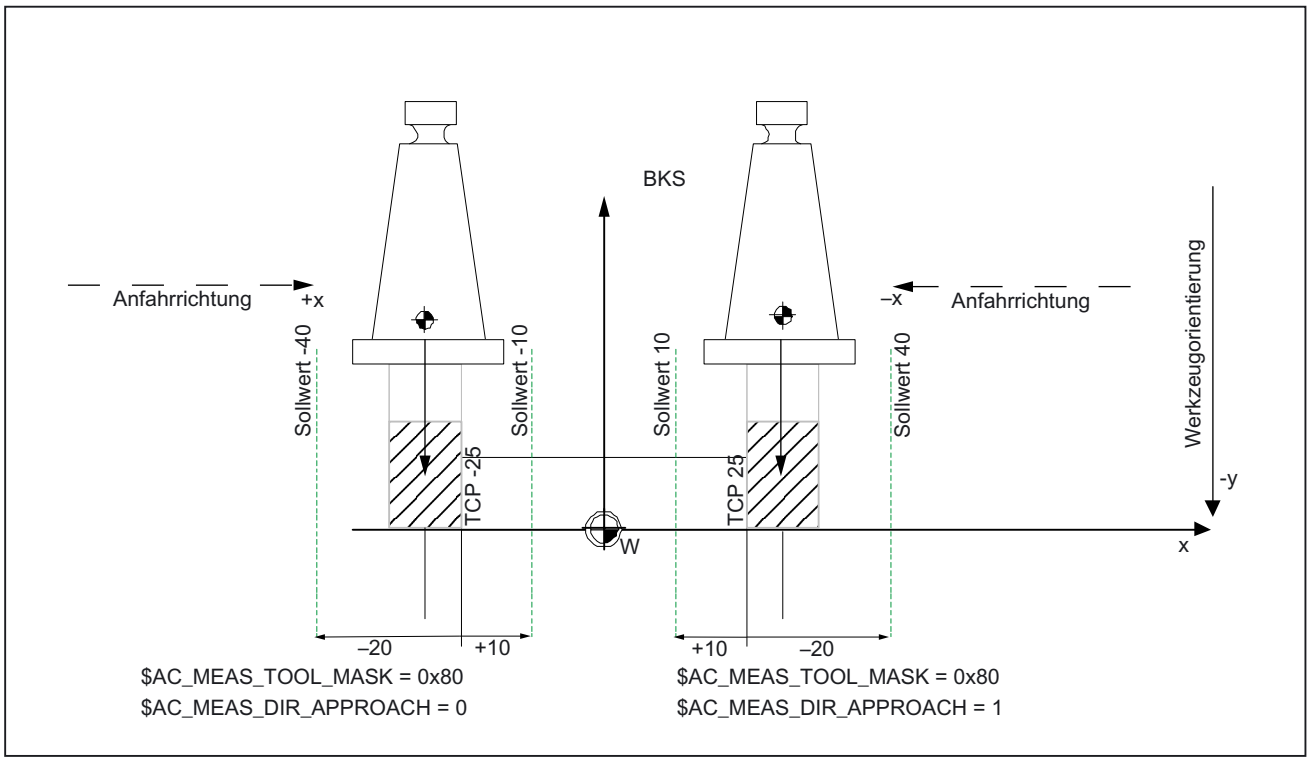
Anfahrrichtung und Werkzeugorientierung +x	Anfahrrichtung und Werkzeugorientierung -x
$AC_MEAS_TOOL_MASK = 0x2 + 0x200	$AC_MEAS_TOOL_MASK = 0x40
$AC_MEAS_DIR_APPROACH = 0	$AC_MEAS_DIR_APPROACH = 1



Bei Werkzeuglage zweier Drehwerkzeuge mit einem Referenzpunkt werden zusätzlich zu den Eingangsvariablen von Messtype 23 die Werte folgender Eingangs-Variablen ausgewertet

Anfahrrichtung und Werkzeugorientierung +x	Anfahrrichtung und Werkzeugorientierung -x
$AC_MEAS_TOOL_MASK = 0x2	$AC_MEAS_TOOL_MASK = 0x40 + 0x200
$AC_MEAS_DIR_APPROACH = 0	$AC_MEAS_DIR_APPROACH = 1

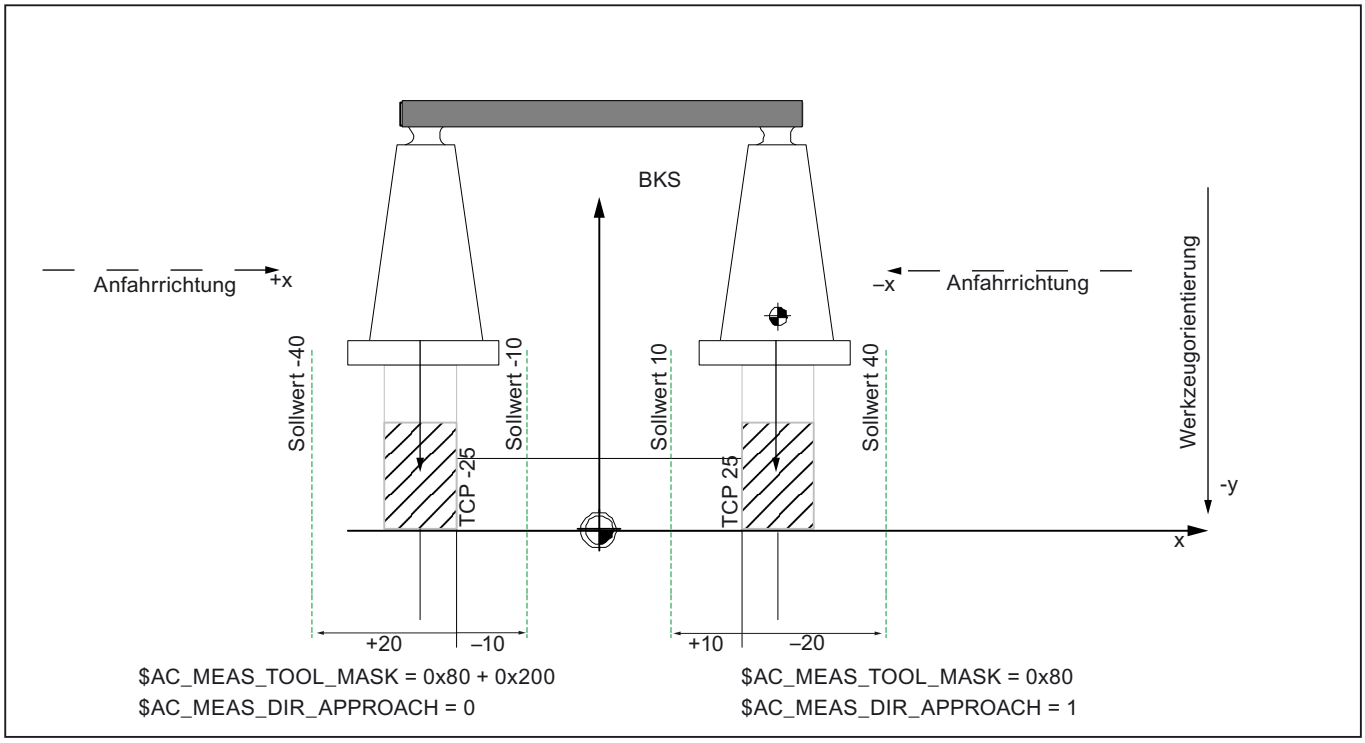
Zwei Fräser mit eigenem Referenzpunkt bei Werkzeugorientierung in -y-Richtung



Bei Werkzeuglage zweier Fräser mit eigenem Referenzpunkt werden zusätzlich zu den Eingangsvariablen von Messtype 23 die Werte folgender Eingangs-Variablen ausgewertet

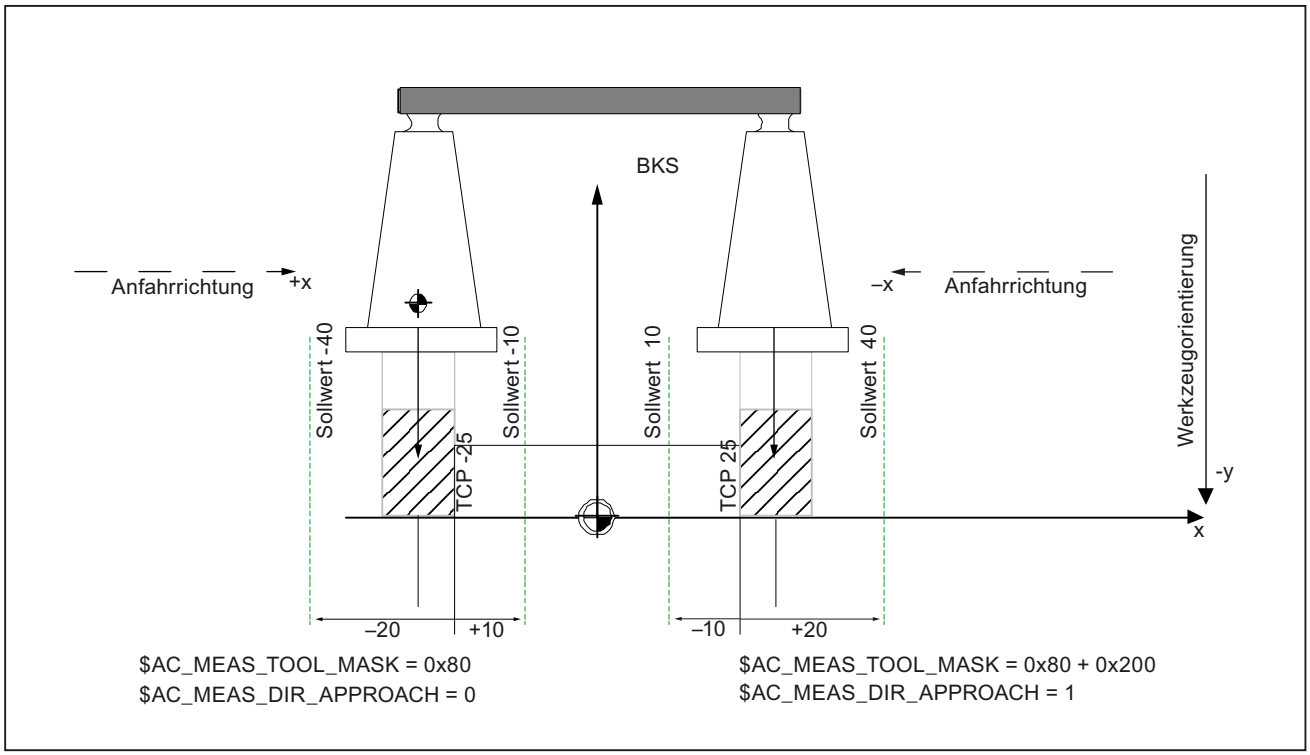
Anfahrrichtung +x, Werkzeugorientierung -y	Anfahrrichtung -x, Werkzeugorientierung -y
$AC_MEAS_TOOL_MASK = 0x80	$AC_MEAS_TOOL_MASK = 0x80
$AC_MEAS_DIR_APPROACH = 0	$AC_MEAS_DIR_APPROACH = 1

Zwei Fräser mit einem Referenzpunkt bei Werkzeugorientierung in -y



Bei Werkzeuglage zweier Fräser mit einem Referenzpunkt werden zusätzlich zu den Eingangsvariablen von Messtype 23 die Werte folgender Eingangs-Variablen ausgewertet

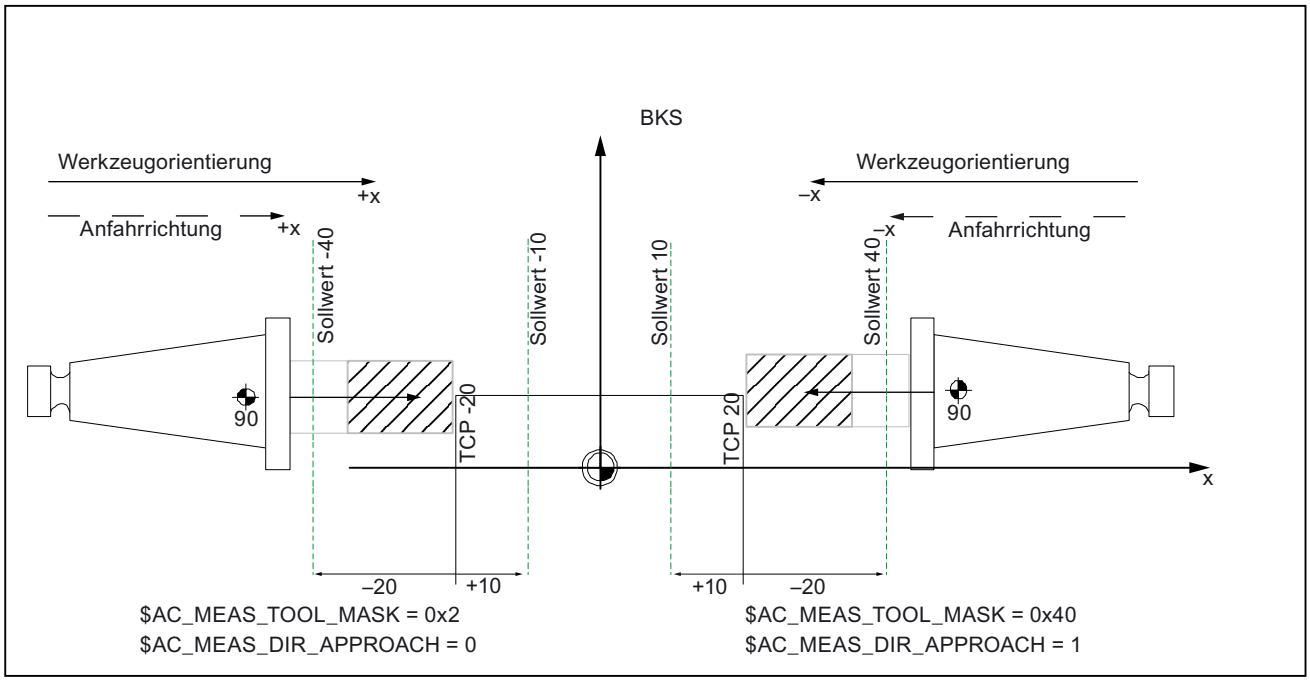
Anfahrrichtung +x, Werkzeugorientierung -y	Anfahrrichtung -x, Werkzeugorientierung -y
$AC_MEAS_TOOL_MASK = 0x80 + 0x200	$AC_MEAS_TOOL_MASK = 0x80
$AC_MEAS_DIR_APPROACH = 0	$AC_MEAS_DIR_APPROACH = 1



Bei Werkzeuglage zweier Fräser mit einem Referenzpunkt werden zusätzlich zu den Eingangsvariablen von Messtype 23 die Werte folgender Eingangs-Variablen ausgewertet

Anfahrrichtung +x, Werkzeugorientierung -y	Anfahrrichtung -x, Werkzeugorientierung -y
$AC_MEAS_TOOL_MASK = 0x80	$AC_MEAS_TOOL_MASK = 0x80 + 0x200
$AC_MEAS_DIR_APPROACH = 0	$AC_MEAS_DIR_APPROACH = 1

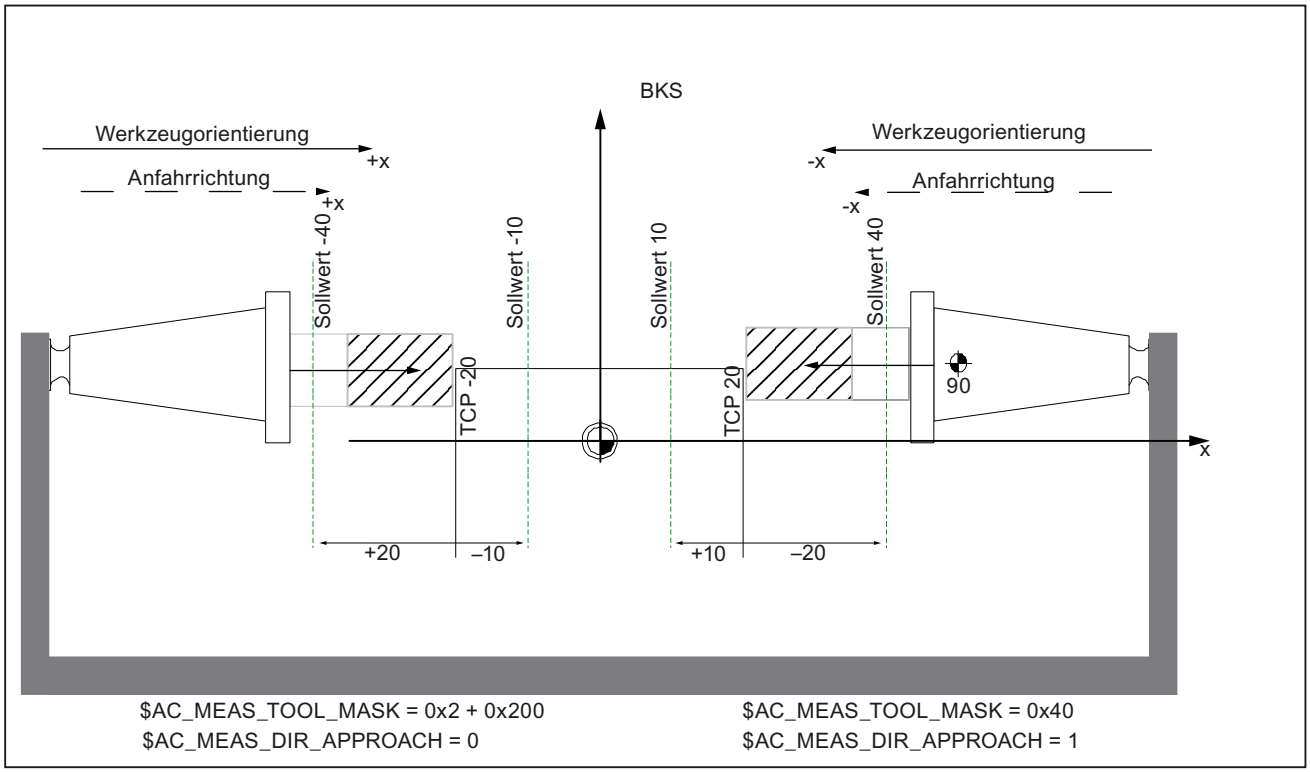
Zwei Fräser mit eigenem Referenzpunkt bei Werkzeugorientierung in Anfahrrichtung



Bei Werkzeuglage zweier Fräser mit eigenem Referenzpunkt werden zusätzlich zu den Eingangsvariablen von Messtype 23 die Werte folgender Eingangs-Variablen ausgewertet

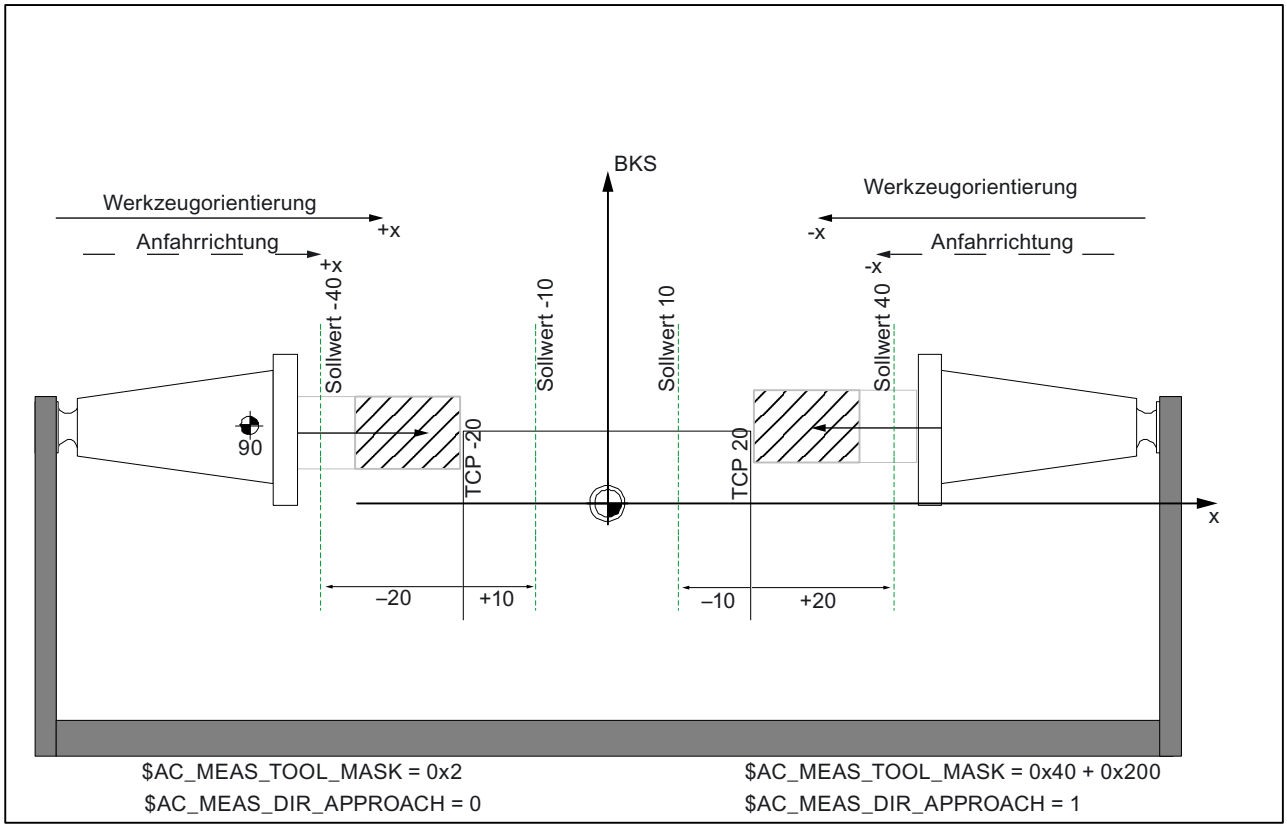
Anfahrrichtung und Werkzeugorientierung +x	Anfahrrichtung und Werkzeugorientierung -x
$AC_MEAS_TOOL_MASK = 0x2	$AC_MEAS_TOOL_MASK = 0x40
$AC_MEAS_DIR_APPROACH = 0	$AC_MEAS_DIR_APPROACH = 1

Zwei Fräser mit einem Referenzpunkt bei gegenläufiger Werkzeuglage zur Orientierung



Bei Werkzeuglage zweier Fräser mit einem Referenzpunkt werden zusätzlich zu den Eingangsvariablen von Messtype 23 die Werte folgender Eingangs-Variablen ausgewertet

Anfahrrichtung und Werkzeugorientierung +x	Anfahrrichtung und Werkzeugorientierung -x
$AC_MEAS_TOOL_MASK = 0x2 +200	$AC_MEAS_TOOL_MASK = 0x40
$AC_MEAS_DIR_APPROACH = 0	$AC_MEAS_DIR_APPROACH = 1



Bei Werkzeuglage zweier Fräser mit einem Referenzpunkt werden zusätzlich zu den Eingangsvariablen von Messtype 23 die Werte folgender Eingangs-Variablen ausgewertet

Anfahrrichtung und Werkzeugorientierung +x	Anfahrrichtung und Werkzeugorientierung -x
$AC_MEAS_TOOL_MASK = 0x2	$AC_MEAS_TOOL_MASK = 0x40 +200
$AC_MEAS_DIR_APPROACH = 0	$AC_MEAS_DIR_APPROACH = 1

Beliebig orientierte Werkzeuge

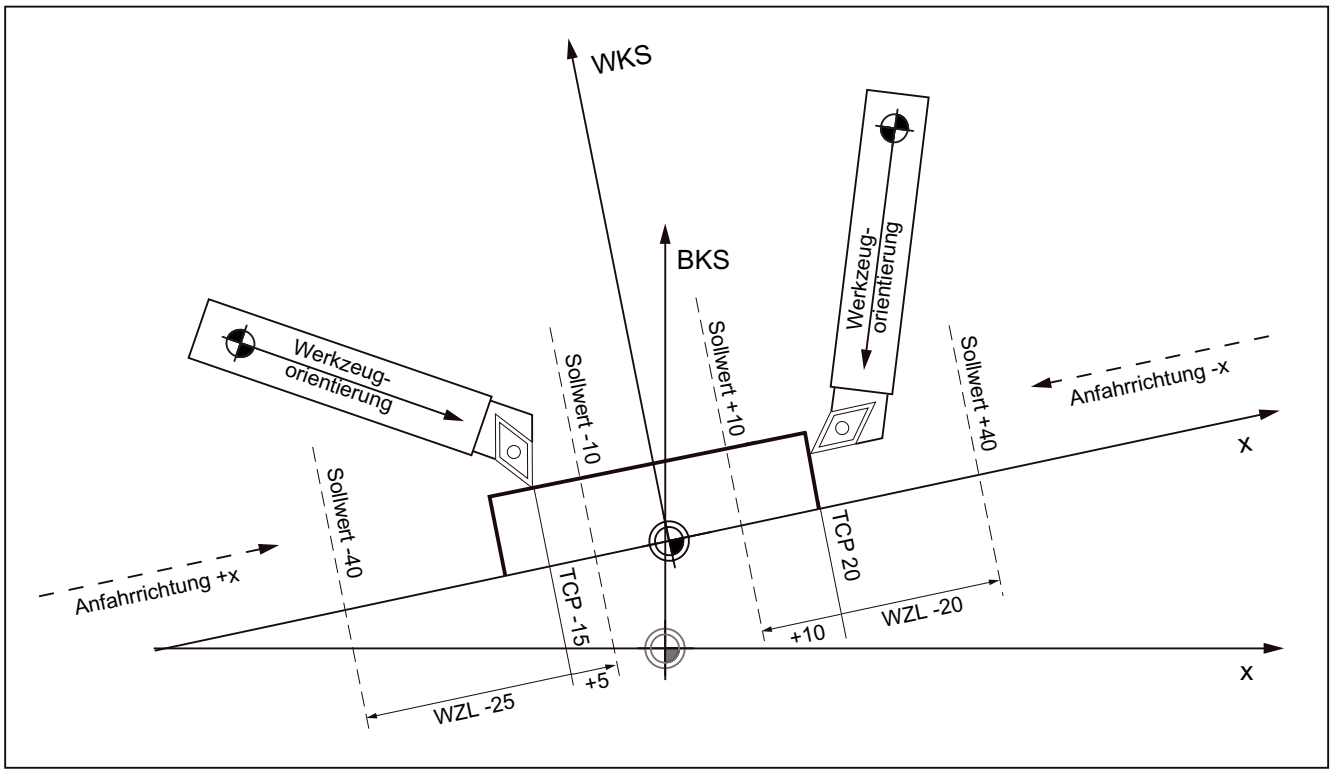


Bild 8-25 Zwei Drehwerkzeuge mit eigenem Referenzpunkt

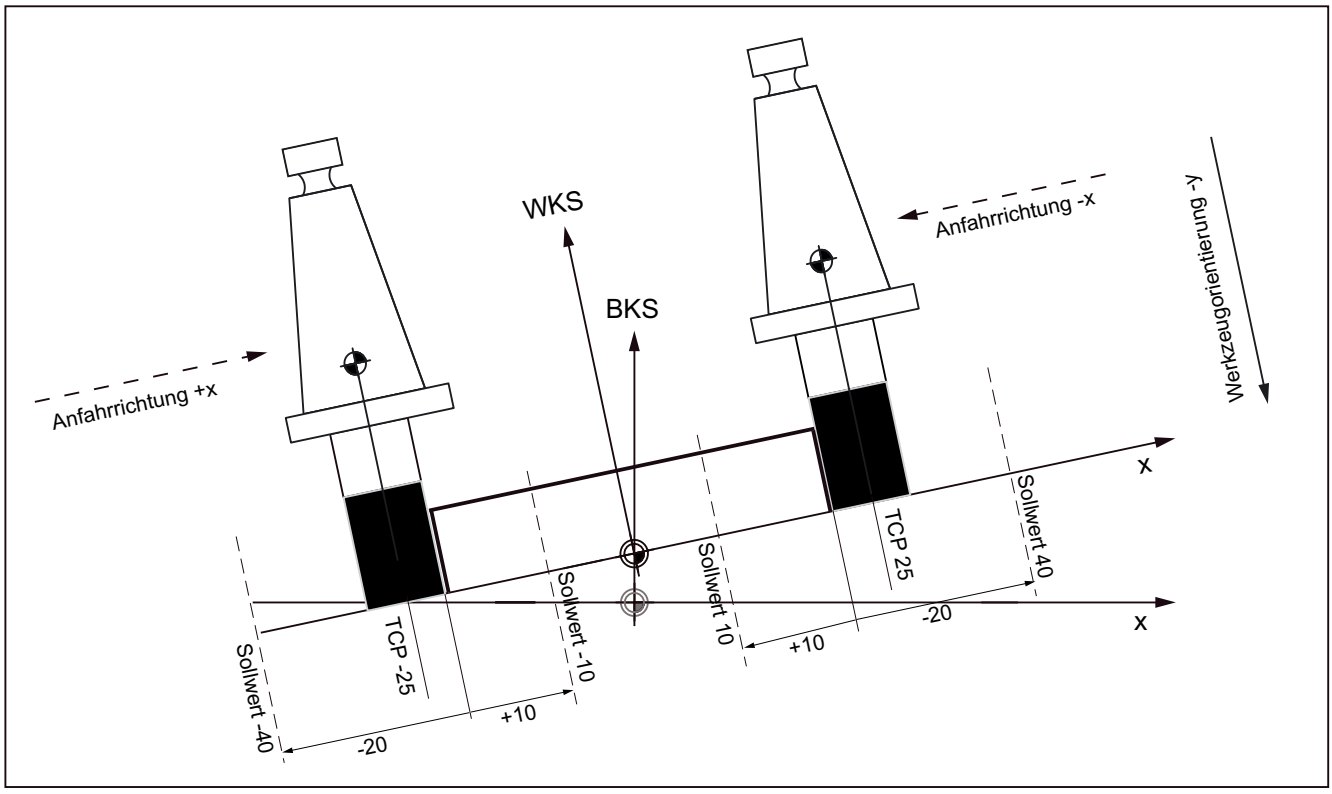


Bild 8-26 Zwei Fräser mit eigenem Referenzpunkt

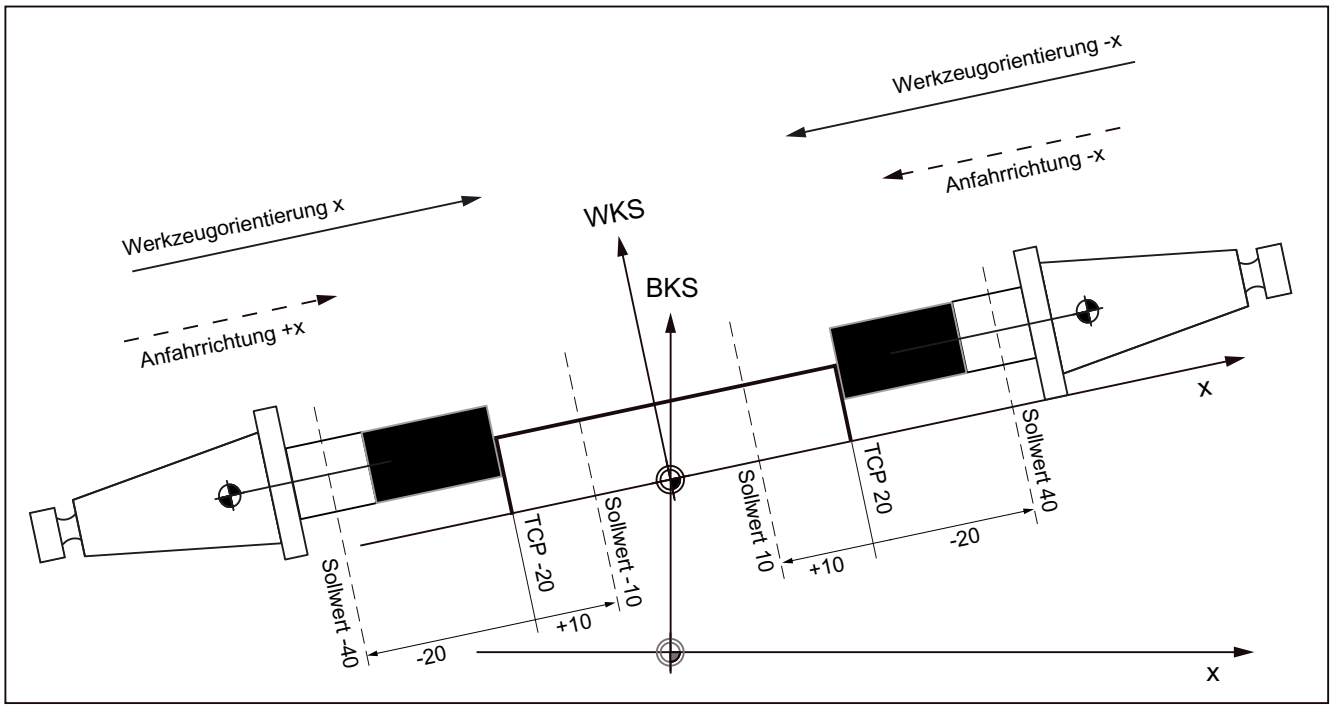


Bild 8-27 Zwei um 90 Grad gedrehte Fräser mit eigenem Referenzpunkt

8.5 Messgenauigkeit und Prüfung

8.5.1 Messgenauigkeit

Genauigkeit

Die Laufzeit des Messsignals ist durch die eingesetzte Hardware vorgegeben. Die Verzögerungszeiten beim Einsatz von SIMODRIVE 611D liegen zwischen 3,625µ ... 9,625µ plus der Reaktionszeit des Messtasters.

Die Messunsicherheit ergibt sich zu:

Messunsicherheit = Laufzeit Messsignal x Verfahrgeschwindigkeit

Die zulässigen Verfahrgeschwindigkeiten hängen von der Anzahl der programmierten Messflanken und dem Verhältnis von IPO-Takt/Lagereglertakt ab.

Korrekte Ergebnisse werden nur bei Verfahrgeschwindigkeiten eingehalten, bei denen nicht mehr als 1 gleiches und nicht mehr als 4 verschieden Triggersignale pro Lagereglertakt eintreffen.

8.5.2 Messtaster-Funktionsprüfung

Beispiel Funktionsprüfung

```
%_N_PRUEF_MESSTASTER_MPF
;$PATH=/_N_MPF_DIR
;Prüfprogramm Messtasteranschaltung
N05 DEF INT MTSIGNAL                              ; Merker für Ansteuerungszustand
N10 DEF INT ME_NR=1                               ; Messeingang-Nummer
N20 DEF REAL MESSWERT_IN_X
N30 G17 T1 D1                                     ; Werkzeugkorrektur für
                                                  ; Messtaster vorwählen
N40 _ANF: G0 G90 X0 F150                          ; Startposition und
                                                  ; Messgeschwindigkeit
N50 MEAS=ME_NR G1 X100                            ; Messung am Messeingang 1
                                                  ; in der X-Achse
N60 STOPRE
N70 MTSIGNAL=$AC_MEA[1]                           ; softwaremäßiges Schaltsignal
                                                  ; am 1. Messeingang lesen
N80 IF MTSIGNAL == 0 GOTOF _FEHL1                 ; Auswertung des Signals
N90 MESSWERT_IN_X=$AA_MW[X]                       ; Messwert in Werkstück-
                                                  ; koordinaten einlesen
N95 M0
N100 M02
N110 _FEHL1: MSG ("Messtaster schaltet nicht!")
N120 M0
N130 M02
```

8.6 Beispiele - nur 840D sl

8.6.1 Messmodus 1

Messung mit 1 Geber

- einmalige Messung
- 1 Messtaster
- Triggersignale sind die steigende und fallende Flanke
- Istwert vom aktuellen Geber

```
N2    MEASA[X] = (1, 1, -1) G01 X100 F100
N3    STOPRE
N4    IF $AC_MEA[1]==FALSE gotof ENDE
N5    R10=$AA_MM1[X]
N6    R11=$AA_MM2[X]
N7    ENDE:
```

Messung mit 2 Gebern

- einmalige Messung
- 1 Messtaster
- Triggersignale sind die steigenden und fallenden Flanken
- Istwerte von 2 Gebern

```
N2    MEASA[X]=(31, 1, -1) G01 X100 F100
N3    STOPRE
N4    IF $AC_MEA[1]==FALSE gotof ENDE
N5    R10=$AA_MM1[X]
N6    R11=$AA_MM2[X]
N7    R12=$AA_MM3[X]
N8    R13=$AA_MM4[X]
N9    ENDE:
```

8.6.2 Messmodus 2

- 2 Messtaster
- Triggersignale sind die steigenden und fallenden Flanken
- Istwert vom aktuellen Geber

```
N2     MEASA[X] = (2, 1, -1, 2, -2) G01 X100 F100
N3     STOPRE
N4     IF $AC_MEA[1]==FALSE gotof MESSTASTER2
N5     R10=$AA_MM1[X]
N6     R11=$AA_MM2[X]
N7     MESSTASTER2
N8     IF $AC_MEA[2]==FALSE gotof ENDE
N9     R12=$AA_MM3[X]
N10    R13=$AA_MM4[X]
N11    ENDE:
```

8.6.3 Kontinuierliches Messen

8.6.3.1 Kontinuierlilches Messen nach Abschluss der programmierten Verfahrbewegung

- Die Messung erfolgt im Messmodus 1
- Messung mit 100 Werten
- 1 Messtaster
- Triggersignal ist die fallende Flanke
- Istwert vom aktuellen Geber

```
N1     DEF REAL MESSWERT[100]
N2     DEF INT INDEX=0
N3     MEAC[x]=(1, 1, -1) G01 X1000 F100
N4     MEAC[X]=(0)                          ; Abbruch
N5     R1=$AC_FIFO1[4]                      ; Anzahl Messwerte
N6     FOR INDEX=0 TO R1
N7     MESSWERT[INDEX]=$AC_FIFO1[0]         ; Messwerte auslesen
N8     ENDFOR:
```

8.6.3.2 Kontinuierliches Messen mit Restweglöschen

- Restweg nach letzter Messung löschen
- Die Messung erfolgt im Messmodus 1
- Messung mit 100 Werten
- 1 Messtaster
- Triggersignal ist die fallende Flanke
- Istwert vom aktuellen Geber

```
N1    DEF INT ANZAHL=100
N2    DEF REAL MESSWERT[ANZAHL]
N3    DEF INT INDEX=0
N4    WHEN $AC_FIFO1[4]==ANZAHL DO DELDTG (X) MEAC[X] =(0)
N5    MEAC[X]=(1, 1, -1) G01 X1000 F100                    ; Messen Start
N6    R1=$AC_FIFO1[4]                                      ; Anzahl Messwerte
N7    FOR INDEX=0 TO R1
N8    MESSWERT[INDEX]=$AC_FIFO1[0]                         ; Messwerte auslesen
N9    ENDFOR:
```

8.6.3.3 Kontinuierliches Messen modal über mehrere Sätze

- Die Messung erfolgt im Messmodus 1
- Messung mit 100 Werten
- 1 Messtaster
- Triggersignal ist die fallende Flanke
- Istwert vom aktuellen Geber

```
N1    DEF INT ANZAHL=100
N2    DEF REAL MESSWERT[ANZAHL]
N3    DEF INT INDEX=0
N4    ID=1 MEAC[X]=(1, 1, -1)                              ; Messen Start
N5    ID=2 WHEN $AC_FIFO1[4]==ANZAHL DO MEAC[X]=(0) CANCEL(2)
N6    G01 X1000 Y100
N7    X100 Y100
N8    R1=$AC_FIFO1[4]                                      ; Anzahl Messwerte
N9    FOR INDEX=0 TO R1
N10   MESSWERT[INDEX]=$AC_FIFO1[0]                         ; Messwerte
                                                           auslesen
N11   ENDFOR:
```

8.6.4 Funktionsprüfung und Wiederholgenauigkeit

Funktionsprüfung

```
%_N_PRUEF_MESSTASTER_MPF
;$PATH=/_N_MPF_DIR
;Prüfprogramm Messtasteranschaltung
N05 DEF INT MTSIGNAL                          ; Merker für Ansteuerungszustand
N10 DEF INT ME_NR=1                           ; Messeingang-Nummer
N20 DEF REAL MESSWERT_IN_X
N30 G17 T1 D1                                 ; Werkzeugkorrektur für
                                              ; Messtaster vorwählen
N40 _ANF: G0 G90 X0 F150                      ; Startposition und
                                              ; Messgeschwindigkeit
N50 MEAS=ME_NR G1 X100                        ; Messung am Messeingang 1
                                              ; in der X-Achse

N60 STOPRE
N70 MTSIGNAL=$AC_MEA[1]                       ; softwaremäßiges Schaltsignal
                                              ; am 1. Messeingang lesen
N80 IF MTSIGNAL == 0 GOTOF _FEHL1             ; Auswertung des Signals
N90 MESSWERT_IN_X=$AA_MW[X]                   ; Messwert in Werkstück-
                                              ; koordinaten einlesen
N95 M0
N100 M02
N110 _FEHL1: MSG ("Messtaster schaltet
nicht!")
N120 M0
N130 M02
```

Wiederholgenauigkeit

Mit dem Programm kann die Messstreuung (Wiederholgenauigkeit) des gesamten Messsystems (Maschine-Messtaster-Signalübertragung zur NC) ermittelt werden.

Im Beispiel wird in der X-Achse 10 mal gemessen und der Messwert in Werkstückkoordinaten aufgenommen.

Es können also die sogenannten zufallsbedingten Maßabweichungen festgestellt werden, die keinem Trend unterliegen.

```
%_N_PRUEF_GENAU_MPF;
$PATH=/_N_MPF_DIR
N05 DEF INT SIGNAL, II                    ; Variablendefinition
N10 DEF REAL MESSWERT_IN_X[10]
N15 G17 T1 D1                             ; Anfangsbedingungen,
```

```
                                        ; Werkzeugkorrektur
                                        ; für Messtaster vorwählen
N20 _ANF: G0 X0 F150 ←                  ; Vorpositionieren in der Messachse
N25 MEAS=+1 G1 X100 ←                   ; Messung am 1. Messeingang bei
                                        ; Schaltsignal nicht ausgelenkt,
                                        ; ausgelenkt in der X-Achse
N30 STOPRE ←                            ; Stop Dekodierung zur nach
                                        ; nachfolgenden Auswertung des
                                        ; Ergebnisses
N35 SIGNAL= $AC_MEA[1]                  ; softwaremäßiges Schaltsignal am
                                        ; 1. Messeingang lesen
N37 IF SIGNAL == 0 GOTOF_FEHL1          ; Schaltsignal prüfen
N40 MESSWERT_IN_X[II]=$AA_MW[X]         ; Messwert in Werkstückkoordinaten lesen
N50 II=II+1
N60 IF II<10 GOTOB_ANF                  ; Wiederholung 10 mal
N65 M0
N70 M02
N80 _FEHL1: MSG ("Messtaster schaltet
nicht")
N90 M0
N95 M02
```

Nach Anwahl der Parameteranzeige (anwenderdefinierte Variablen) können im Feld MESSWERT_IN_X[10] die Messergebnisse abgelesen werden, solange die Programmabarbeitung aktiv ist.

8.7 Datenlisten

8.7.1 Maschinendaten

8.7.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
13200	MEAS_PROBE_LOW_ACTIVE	Schaltverhalten des Messtasters
13201	MEAS_PROBE_SOURCE	Messpulsimulation über digitalen Ausgang
13210	MEAS_TYPE	Art des Messens bei PROFIBUS_DP Antrieben

8.7.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
20360	TOOL_PARAMETER_DEF_MASK	Definition der WZ-Parameter
28264	MM_LEN_AC_FIFO	Länge der FIFO-Variablen $AC_FIFO ...

8.7.2 Systemvariablen

Tabelle aller Eingangswerte

Typ	Systemvariablen-Name	Beschreibung
INT	$AC_MEAS_SEMA	Belegung des Interfaces
INT	$AC_MEAS_VALID	Gültigkeitsbits für die Eingangswerte
REAL	$AA_MEAS_POINT1[ax]	1. Messpunkt für alle Kanalachsen
REAL	$AA_MEAS_POINT2[ax]	2. Messpunkt für alle Kanalachsen
REAL	$AA_MEAS_POINT3[ax]	3. Messpunkt für alle Kanalachsen
REAL	$AA_MEAS_POINT4[ax]	4. Messpunkt für alle Kanalachsen
REAL	$AA_MEAS_SETPOINT[ax]	Sollposition für alle Kanalachsen
REAL	$AA_MEAS_SETANGLE[ax]	Soll-Winkel für alle Kanalachsen
INT	$AC_MEAS_P1_COORD	Koordinatensystem für den 1. Messpunkt
INT	$AC_MEAS_P2_COORD	Koordinatensystem für den 2. Messpunkt
INT	$AC_MEAS_P3_COORD	Koordinatensystem für den 3. Messpunkt
INT	$AC_MEAS_P4_COORD	Koordinatensystem für den 4. Messpunkt
INT	$AC_MEAS_SET_COORD	Koordinatensystem des Sollpunktes
INT	$AC_MEAS_LATCH[0..3]	Messpunkte im WKS ablatchen
INT	$AA_MEAS_P1_VALID[ax]	1. Messpunkt im WKS ablatchen

Typ	Systemvariablen-Name	Beschreibung
INT	$AA_MEAS_P2_VALID[ax]	2. Messpunkt im WKS ablatchen
INT	$AA_MEAS_P3_VALID[ax]	3. Messpunkt im WKS ablatchen
INT	$AA_MEAS_P4_VALID[ax]	4. Messpunkt im WKS ablatchen
INT	$AA_MEAS_SP_VALID[ax]	Sollposition der Achse gültig setzen
REAL	$AC_MEAS_WP_SETANGLE	Werkstücklage-Sollwinkel
REAL	$AC_MEAS_CORNER_SETANGLE	Soll-Schnittwinkel der Ecke
INT	$AC_MEAS_DIR_APPROACH	Anfahrrichtung
INT	$AC_MEAS_ACT_PLANE	Arbeitsebene zur Werkstückes
INT	$AC_MEAS_SCALEUNIT	Maßeinheit INCH / METRISCH
INT	$AC_MEAS_FINE_TRANS	Korrekturen in die Feinverschiebung
INT	$AC_MEAS_FRAME_SELECT	Frameauswahl bei der Werkstückvermessung
INT	$AC_MEAS_CHSFR	Einstellung Framekette: Systemframes
INT	$AC_MEAS_NCBFR	Einstellung Framekette: Globale Basisframes
INT	$AC_MEAS_CHBFR	Einstellung Framekette: Kanal-Basisframes
INT	$AC_MEAS_UIFR	Einstellung Framekette: Einstellbare Frames
INT	$AC_MEAS_PFRAME	Einstellung Framekette: Prog. Frame
INT	$AC_MEAS_T_NUMBER	Werkzeugauswahl
INT	$AC_MEAS_D_NUMBER	Schneidenauswahl
INT	$AC_MEAS_TOOL_MASK	Werkzeugeinstellungen
INT	$AC_MEAS_TYPE	Messtyp
REAL	$AC_MEAS_INPUT[10]	Mess-Eingangsparameter

Tabelle aller Ausgangswerte

Typ	Systemvariablen-Name	Beschreibung
FRAME	$AC_MEAS_FRAME	Ergebnisframe
REAL	$AC_MEAS_WP_ANGLE	erechneter Werkstücklage-Winkel
REAL	$AC_MEAS_CORNER_ANGLE	Berechneter Schnittwinkel
REAL	$AC_MEAS_DIAMETER	Berechneter Durchmesser
REAL	$AC_MEAS_TOOL_LENGTH	Berechnete Werkzeuglänge
REAL	$AC_MEAS_RESULTS[n]	Messergebnisse (abhängig vom Messtyp)

N3: Softwarenocken, Wegschaltsignale - nur 840D sl

9

9.1 Kurzbeschreibung

Funktion

Die Funktion "Softwarenocken" erzeugt positionsabhängige Schaltsignale für Achsen, die einen Lageistwert liefern (Maschinenachsen), und für simulierte Achsen.
Diese Nockensignale können an die PLC und zusätzlich an die NCK-Peripherie ausgegeben werden.

Die Nockenpositionen, bei denen die Signalausgänge gesetzt werden, können über Settingdaten definiert und verändert werden. Die Settingdaten können über HMI, PLC und Teileprogramm gelesen und geschrieben werden.

Aktivierung

Die Funktion "Softwarenocken" ist in allen Betriebsarten funktionsfähig und aktivierbar. Sie bleibt auch bei `RESET` oder NOT-AUS aktiv.

Anwendungsbereich

Nockensignale können z.B. verwendet werden:

- zum Aktivieren von Schutzbereichen
- zum positionsabhängigen Einleiten zusätzlicher Bewegungen
- als Umsteuersignale für hydraulisch gesteuerte Pendelachsen

Achstypen

Softwarenocken können bei Linearachsen und Modulo-Rundachsen, die als Maschinenachsen definiert sind, eingesetzt werden.

Nockenbereich / Nockenpaar

Die Nocken werden immer als Nockenpaar (bestehend aus einem Plus- und einem Minusnocken) einer Achse zugeordnet. Es stehen 32 Nockenpaare zur Verfügung.

Der Plus- bzw. Minusnocken bildet jeweils einen mechanischen Schaltnocken nach, der in einer bestimmten Anfahrrichtung an definierter Stelle (Nockenposition) betätigt wird, wenn die Achse die Nockenposition erreicht.

Den Plus- und Minusnocken sind Nockenbereiche wie folgt zugeordnet:

Nockenbereich plus:	alle Positionen ≥ Plusnocken
Nockenbereich minus:	alle Positionen ≤ Minusnocken

9.2 Nockensignale und Nockenpositionen

9.2.1 Erzeugung der Nockensignale für getrennte Ausgabe

Allgemeines

Beide Nockensignale können an die PLC und an die NCK-Peripherie ausgegeben werden. Die getrennte Ausgabe der Plus- und Minus-Nockensignale ermöglicht es, auf einfache Weise zu erkennen, ob die Achse im oder außerhalb vom Nockenbereich plus bzw. minus steht.

Linearachsen

Die Schaltflanken der Nockensignale werden abhängig von der Verfahrrichtung der Achse erzeugt:

- Das Minus-Nockensignal schaltet von 1 auf 0, wenn die Achse den Minusnocken in positiver Achsrichtung überfährt.
- Das Plus-Nockensignal schaltet von 0 auf 1, wenn die Achse den Plusnocken in positiver Richtung überfährt.

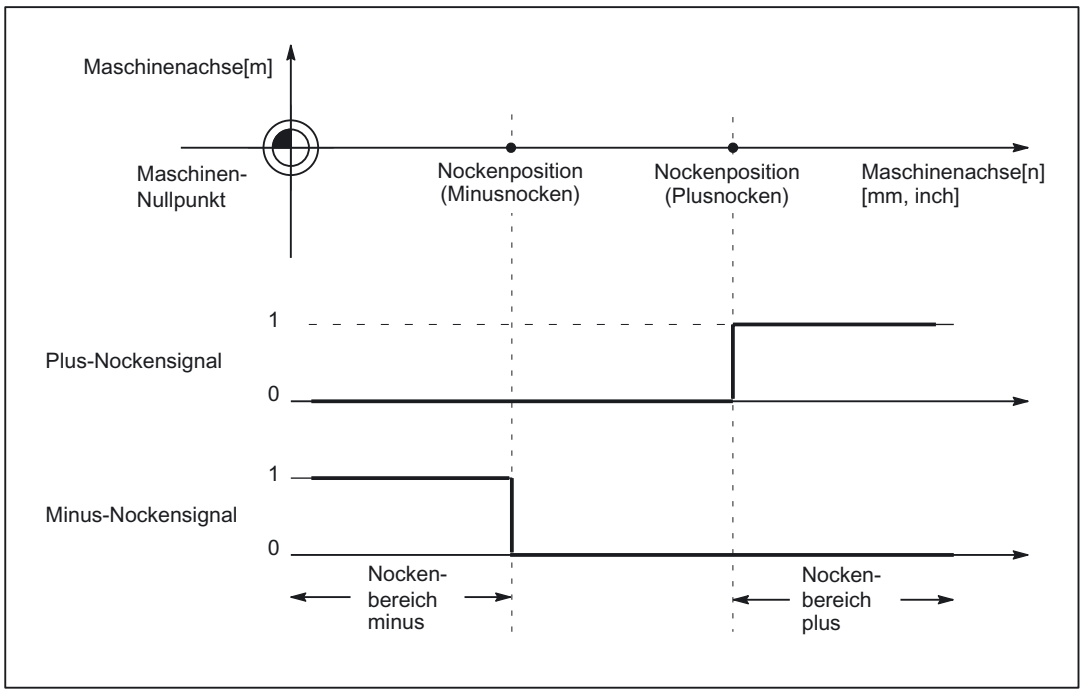


Bild 9-1 Softwarenocken für Linearachse (Minusnocken < Plusnocken)

Hinweis

Wegschaltsignale:

Steht die Achse genau auf Nockenposition (plus oder minus) flackert der definierte Ausgang. Wird ein Inkrement weitergefahren, ist der Ausgang eindeutig null oder eins.

Das Flackern der Signale ergibt sich durch Flackern der Istposition.
Die Istposition wird ausgewertet.

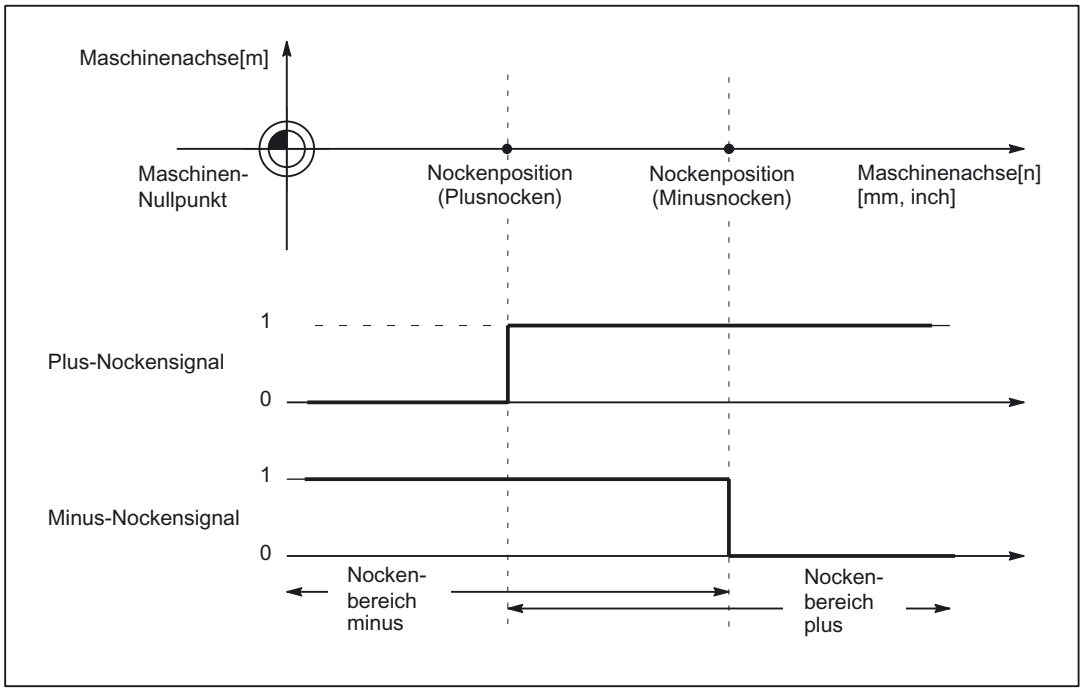


Bild 9-2 Softwarenocken für Linearachse (Plusnocken < Minusnocken)

Modulo-Rundachsen

Die Schaltflanken der Nockensignale werden abhängig von der Verfahrrichtung der Rundachse erzeugt:

- Das Plus-Nockensignal schaltet bei Überfahren des Minusnockens in positiver Achsrichtung von 0 auf 1 und bei Überfahren des Plusnockens von 1 auf 0 zurück.
- Das Minus-Nockensignal wechselt den Pegel bei jeder positiven Flanke des Plus-Nockensignals.

Hinweis

Das beschriebene Verhalten des Plus-Nockens gilt unter der **Bedingung**:

Plusnocken - Minusnocken < 180 Grad

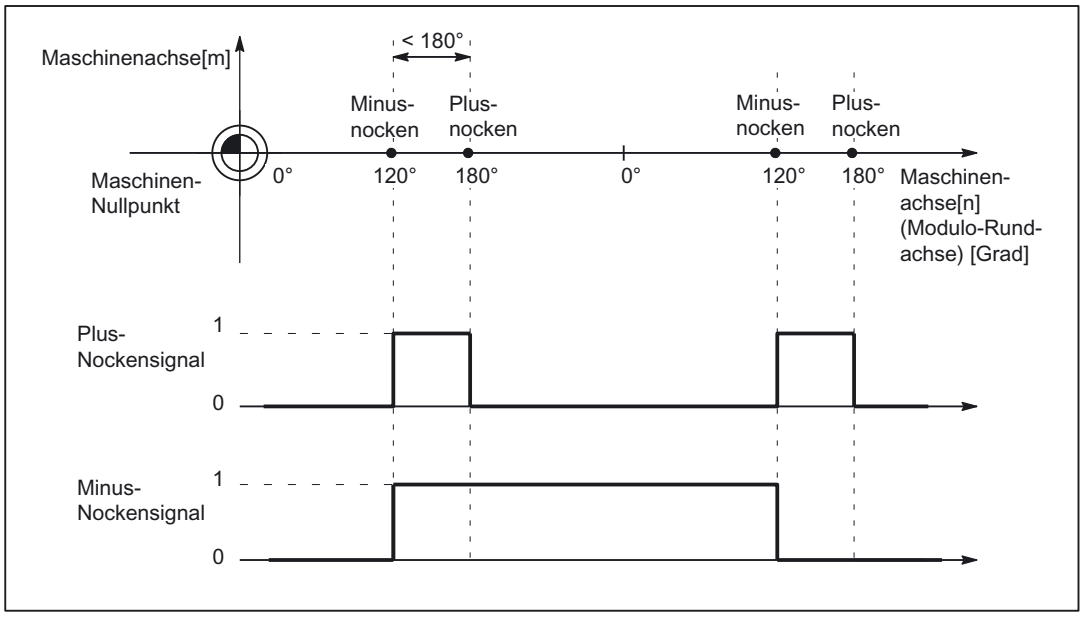


Bild 9-3 Softwarenocken für Modulo-Rundachse (Plusnocken - Minusnocken < 180°)

Am Signalwechsel des Minusnockens ist das Überfahren auch dann zu erkennen, wenn der Nockenbereich so kurz eingestellt ist, dass die PLC ihn nicht sicher erfassen kann.

Beide Nockensignale können an die PLC und an die NCK-Peripherie ausgegeben werden. Die getrennte Ausgabe der Plus- und Minus-Nockensignale ermöglicht es, auf einfache Weise zu erkennen, ob die Achse im oder außerhalb vom Nockenbereich plus bzw. minus steht.

Ist diese Bedingung (Plusnocken - Minusnocken < 180°) nicht erfüllt oder wird der Minusnocken größer als der Plusnocken gewählt, so invertiert sich das Verhalten des Plus-Nockensignals. Das Verhalten des Minus-Nockensignals bleibt gleich.

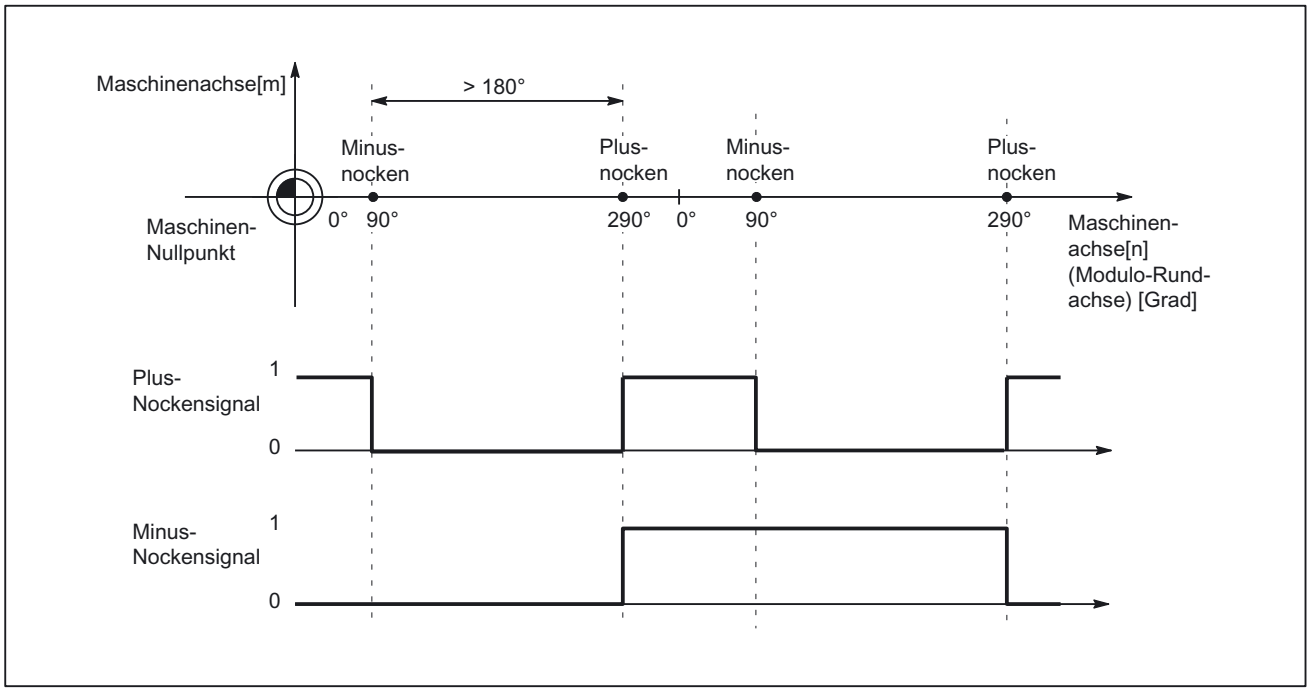


Bild 9-4 Softwarenocken für Modulo-Rundachse (Plusnocken - Minusnocken > 180 Grad)

9.2.2 Erzeugung der Nockensignale bei verknüpfter Ausgabe

Allgemeines

Eine verknüpfte Ausgabe von Plus- und Minus-Nockensignale erfolgt für:

- Timergesteuerte Nockensignalausgabe auf die 4 On-Board-Ausgänge der NCU
- Ausgabe auf die NCK-Peripherie, falls das 2. Byte in den folgenden Maschinendaten nicht spezifiziert wurde (= "0"):
  MD10470 SW_CAM_ASSIGN_FASTOUT_2
  bis
  MD10473 SW_CAM_ASSIGN_FASTOUT_4

Linearachsen

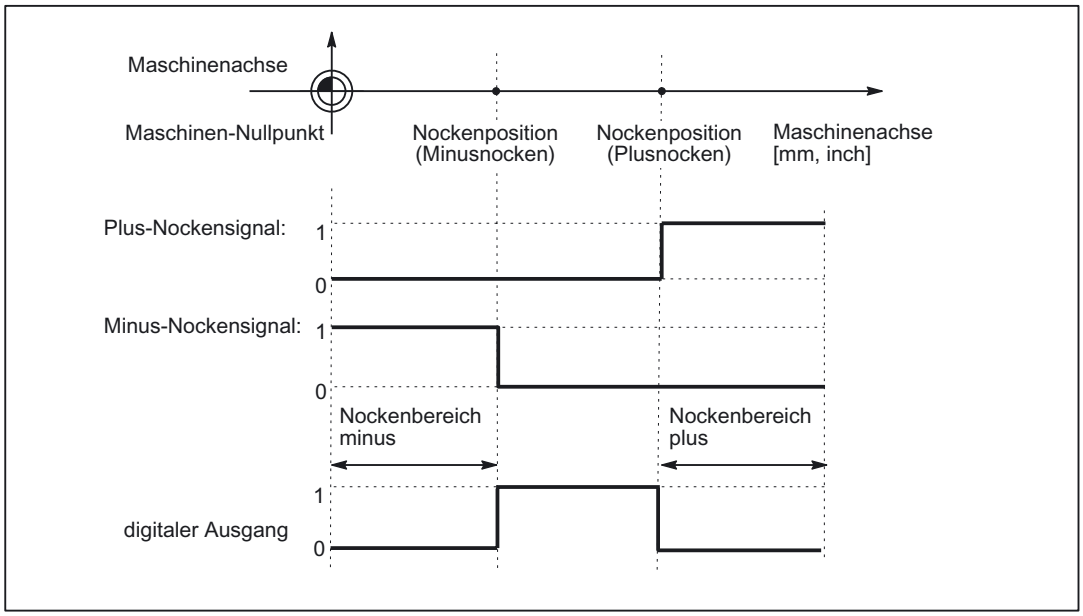


Bild 9-5 Wegschaltsignale für Linearachse (Minusnocken < Plusnocken)

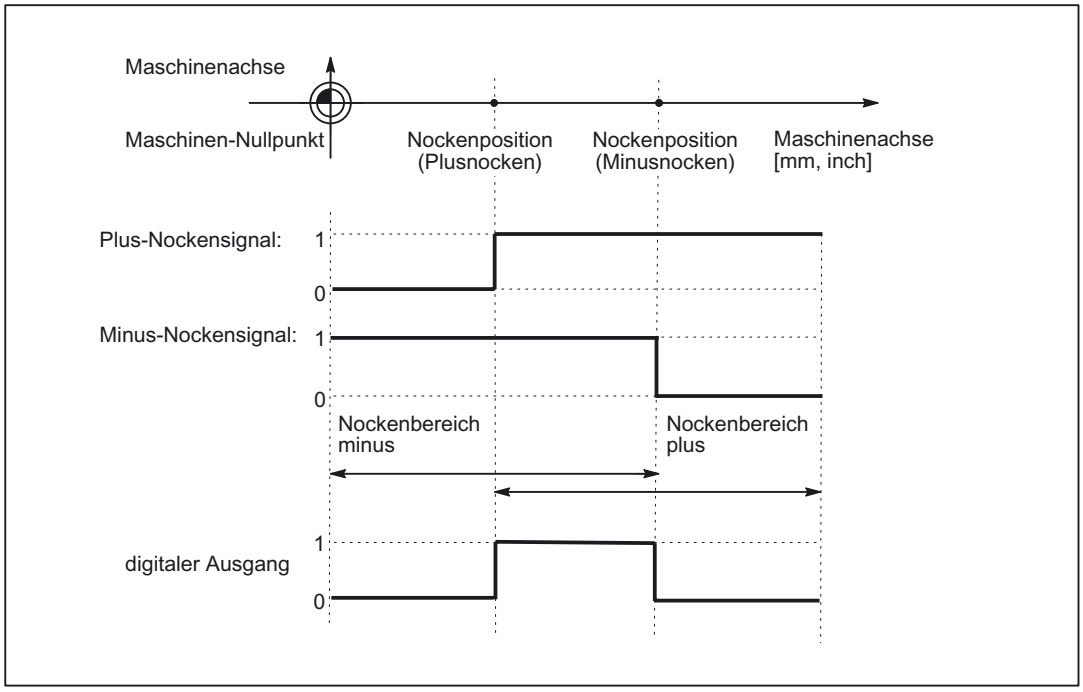


Bild 9-6 Wegschaltsignale für Linearachse (Plusnocken < Minusnocken)

Modulo-Rundachse

Bei Modulo-Rundachsen ist das Signalverhalten standardmäßig abhängig von der Nockenbreite:

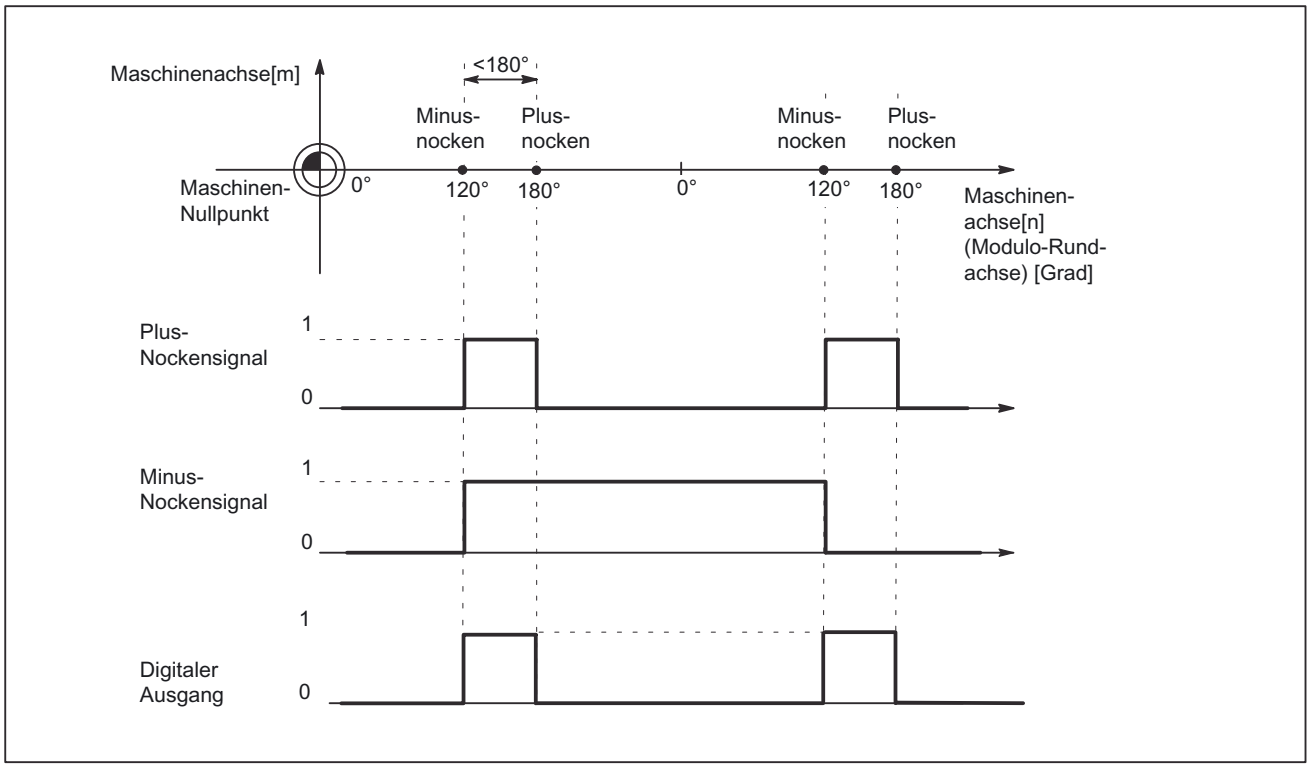


Bild 9-7 Softwarenocken für Modulo-Rundachse (Plusnocken - Minusnocken < 180 Grad)

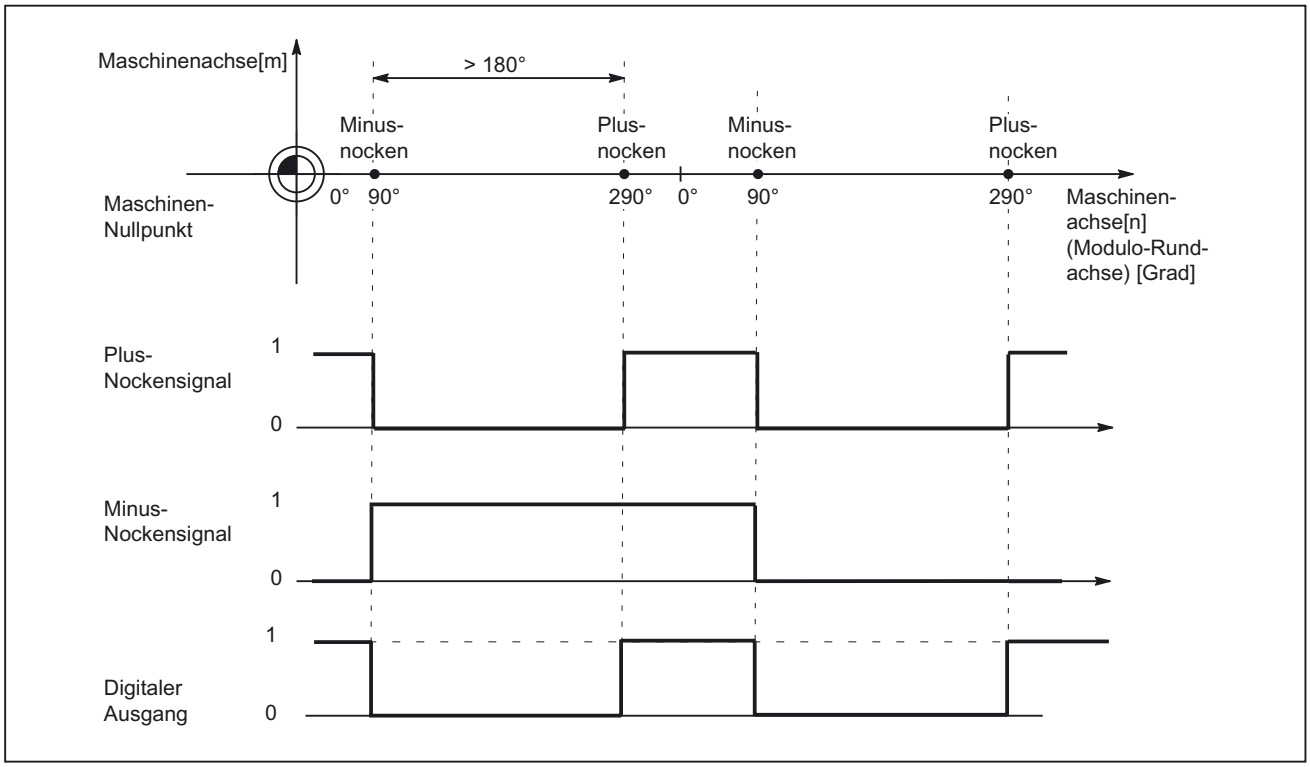


Bild 9-8 Softwarenocken für Modulo-Rundachse (Plusnocken - Minusnocken > 180°)

Unterdrückung der Signalinvertierung

Mit der folgenden Maschinendatum-Einstellung kann eine Unterdrückung der Signalinvertierung für
Plusnocke - Minusnocke > 180°
angewählt werden :
MD10485 SW_CAM_MODE Bit 1=1

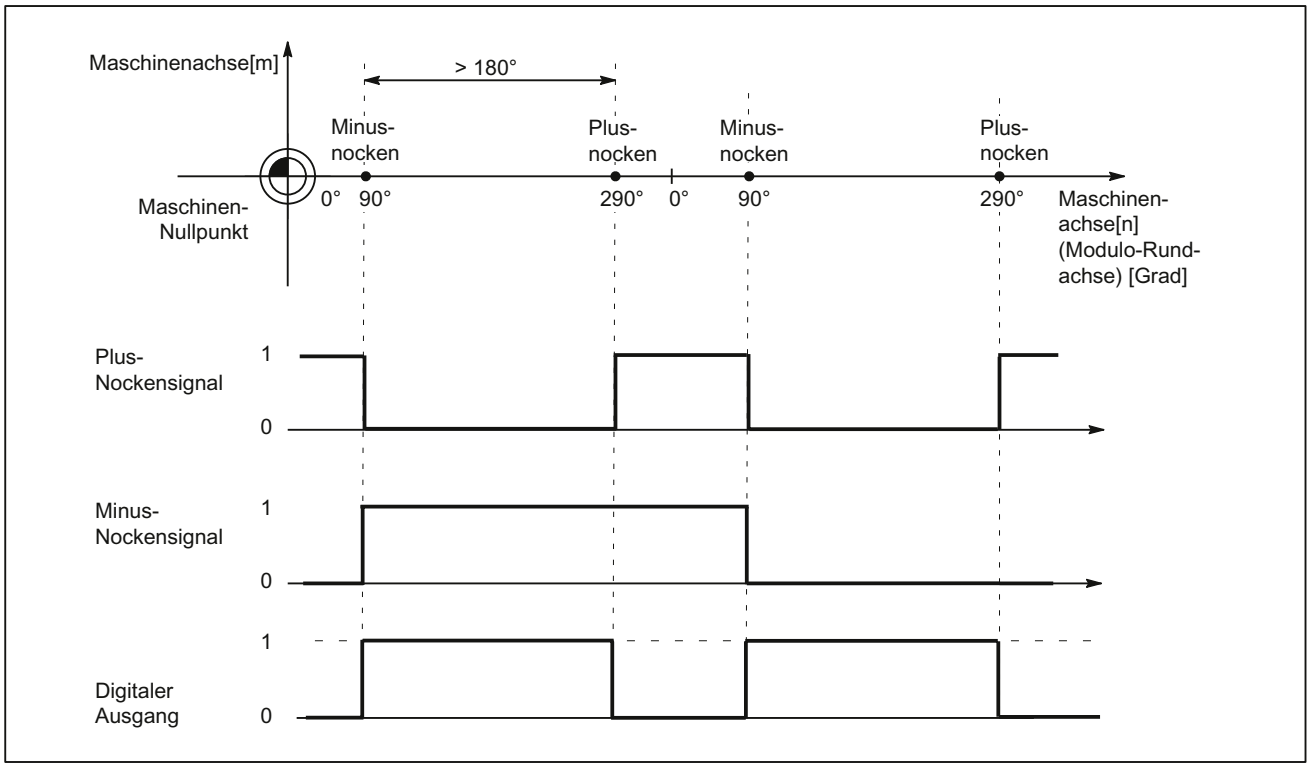


Bild 9-9 Softwarenocken für Modulo-Rundachse (Plusnocken - Minusnocken > 180°) und Unterdrückung der Signalinvertierung

9.2.3 Nockenpositionen

Setzen der Nockenpositionen

Die Nockenpositionen der Plus- und Minusnocken werden über die folgenden allgemeinen Settingdaten definiert:

SD41500 SW_CAM_MINUS_POS_TAB_1[n]	Position Minusnocken 1 - 8
SD41501 SW_CAM_PLUS_POS_TAB_1[n]	Position Plusnocken 1 – 8
SD41502 SW_CAM_MINUS_POS_TAB_2[n]	Position Minusnocken 9 - 16
SD41503 SW_CAM_PLUS_POS_TAB_2[n]	Position Plusnocken 9 - 16
SD41504 SW_CAM_MINUS_POS_TAB_3[n]	Position Minusnocken 17 - 24
SD41505 SW_CAM_PLUS_POS_TAB_3[n]	Position Plusnocken 17 - 24

SD41506 SW_CAM_MINUS_POS_TAB_4[n]	Position Minusnocken 25 - 32
SD41507 SW_CAM_PLUS_POS_TAB_4[n]	Position Plusnocken 25 - 32

Hinweis

Durch die Aufteilung in Gruppen für je acht Nockenpaare können unterschiedliche Zugriffsschutzstufen vergeben werden (z. B. für maschinenbezogene und werkstückbezogene Nockenpositionen). Die Positionen werden im Maschinenkoordinatensystem eingegeben. Eine Überprüfung hinsichtlich des maximalen Verfahrbereiches erfolgt nicht.

Maßsystem metrisch/inch

MD10260 CONVERT_SCALING_SYSTEM=1
Bei der obigen Maschinendatum-Einstellung beziehen sich die Nockenpositionen nicht mehr auf das eingestellte Grundsystem, sondern auf das im folgenden Maschinendatum projektierte Maßsystem: :
MD10270 POS_TAB_SCALING_SYSTEM

MD10270 POS_TAB_SCALING_SYSTEM=0:	metrisch
MD10270 POS_TAB_SCALING_SYSTEM=1:	inch

MD10270 POS_TAB_SCALING_SYSTEM
Das obige Maschinendatum legt damit das Maßsystem für Positionsangaben aus den folgenden Settingdaten fest:
SD41500 bis SD41507

Eine Umschaltung mit `G70`/`G71` bzw. `G700`/`G710` hat keine Auswirkung.

Erfassung der Nockenpositionen

Für das Setzen der Nockensignale wird die Istposition der Achsen mit der Nockenposition verglichen.

Schreiben/Lesen der Nockenpositionen

Auf die Settingdaten kann über HMI, PLC und Teileprogramm lesend und schreibend zugegriffen werden.

Zugriffe vom Teileprogramm aus erfolgen nicht bearbeitungssynchron.
Eine Synchronisation kann nur durch einen programmierten Satzvorlauf-Stopp (`STOPRE`-Befehl) erreicht werden.

Im PLC-Anwenderprogramm besteht die Möglichkeit, die Nockenpositionen mit dem FB2 und FB3 zu lesen und zu schreiben.

Achs-Nocken-Zuordnung

Die Zuordnung eines Nockenpaares zu einer Maschinenachse erfolgt mit dem allgemeinen Maschinendatum:
MD10450 SW_CAM_ASSIGN_TAB[n]
(Zuordnung Softwarenocken zu Maschinenachsen).

Hinweis

Eine geänderte Achszuordnung wird mit dem nächsten NCK-Hochlauf wirksam.

Nockenpaare, denen keine Achse zugeordnet wurde, sind nicht aktiv.

Ein Nockenpaar kann immer nur einer Maschinenachse zugeordnet werden.

Für eine Maschinenachse können mehrere Nockenpaare definiert werden.

9.2.4 Vorhalte-/Verzögerungszeit (dynamischer Nocken)

Laufzeiten

Zur Kompensation von Verzögerungszeiten können jedem Minus- und Plusnocken zwei additiv wirkende Vorhalte- bzw. Verzögerungszeiten für die Nockensignal-Ausgabe zugeordnet werden.

Die zwei Vorhalte- bzw. Verzögerungszeiten werden in je ein Maschinen- und Settingdatum eingetragen.

Hinweis

Die Eingabe negativer Zeitwerte bewirkt eine zeitliche Verzögerung der Nockensignal-Ausgabe.

Eingabe in Maschinendaten

Die **erste** Vorhalte- bzw. Verzögerungszeit wird in die folgenden allgemeinen Maschinendaten eingetragen:

MD10460 SW_CAM_MINUS_LEAD_TIME[n]
(Vorhalte- bzw. Verzögerungszeit an den Minusnocken)

MD10461 SW_CAM_PLUS_LEAD_TIME[n]
(Vorhalte- bzw. Verzögerungszeit an den Plusnocken)

In diese Maschinendaten können z.B. eingetragen werden:

- konstante interne Verzögerungszeiten zwischen der Istwerterfassung und der Nockensignalausgabe (z.B. mit einem Oszilloskop ermittelt)
- konstante externe Verzögerungszeiten

Eingabe in Settingdaten

Die zweite Vorhalte- bzw. Verzögerungszeit wird in die folgenden allgemeinen Settingdaten eingetragen:

SD41520 SW_CAM_MINUS_TIME_TAB_1[n]	Vorhalte- bzw. Verzögerungszeit an den Minusnocken 1 - 8
SD41521 SW_CAM_PLUS_TIME_TAB_1[n]	Vorhalte- bzw. Verzögerungszeit an den Plusnocken 1 - 8
SD41522 SW_CAM_MINUS_TIME_TAB_2[n]	Vorhalte- bzw. Verzögerungszeit an den Minusnocken 9 - 16
SD41523 SW_CAM_PLUS_TIME_TAB_2[n]	Vorhalte- bzw. Verzögerungszeit an den Plusnocken 9 - 16
SD41524 SW_CAM_MINUS_TIME_TAB_3[n]	Vorhalte- bzw. Verzögerungszeit an den Minusnocken 17 - 24
SD41525 SW_CAM_PLUS_TIME_TAB_3[n]	Vorhalte- bzw. Verzögerungszeit an den Plusnocken 17 - 24
SD41526 SW_CAM_MINUS_TIME_TAB_4[n]	Vorhalte- bzw. Verzögerungszeit an den Minusnocken 25 - 32
SD41527 SW_CAM_PLUS_TIME_TAB_4[n]	Vorhalte- bzw. Verzögerungszeit an den Plusnocken 25 - 32

In diese Settingdaten sind z. B. Verzögerungszeiten einzutragen, die sich während der Bearbeitung ändern können.

9.3 Ausgabe der Nockensignale

9.3.1 Aktivierung

Der Status der Nocken (Nockensignale) kann an die PLCund zusätzlich an die NCK-Peripherieausgegeben werden.

Aktivierung der Nockensignalausgabe

Die Ausgabe der Nockensignale einer Achse wird über das achsspezifische NC/PLC-Nahtstellensignal:
DB31, ... DBX2.0 (Nocken-Aktivierung)
aktiviert.

Rückmeldung an die PLC

Die erfolgreiche Aktivierung aller Nocken einer Achse wird über das achsspezifische NC/PLC-Nahtstellensignal:
DB31, ... DBX62.0 (Nocken aktiv)
an die PLC zurückgemeldet.

Hinweis

Die Aktivierung kann durch den PLC-Anwender mit anderen Bedingungen verknüpft werden (z.B. Achse referiert, Reset wirksam).

9.3.2 Ausgabe der Nockensignale an die PLC

Ausgabe an die PLC

Für alle Maschinenachsen, deren Softwarenocken aktiviert sind, wird der Status der Nockensignale an die PLC ausgegeben.

Die Ausgabe erfolgt im IPO-Takt und wird von der PLC asynchron übernommen.

Minus-Nockensignale

Der Status der Minus-Nockensignale wird in die allgemeinen NC/PLC-Nahtstellensignale:
DB10 DBX110.0 bis 113.7 (Minus-Nockensignal 1 bis 32)
eingetragen.

Plus-Nockensignale

Der Status der Plus-Nockensignale wird in die allgemeinen NC/PLC-Nahtstellensignale:
DB10 DBX114.0 bis 117.7 (Plus-Nockensignal 1 bis 32)
eingetragen.

Wenn kein Messsystem angewählt oder das NC/PLC-Nahtstellensignal:
DB31, ... DBX2.0 (Nocken-Aktivierung)
auf 0 gesetzt ist,
dann werden die folgenden NC/PLC-Nahtstellensignale ebenfalls auf 0 gesetzt:

DB10 DBX110.0-113.7 (Minus-Nockensignal 1-32)

DB10 DBX114.0-117.7 (Plus-Nockensignal 1-32)

DB31, ... DBX62.0 (Nocken aktiv)

9.3.3 Ausgabe der Nockensignale im Lageregeltakt an die NCK-Peripherie

Signalausgabe im Lageregeltakt

Für die über die Maschinendaten:
MD10470 SW_CAM_ASSIGN_FASTOUT_1
bis
MD10473 SW_CAM_ASSIGN_FASTOUT_4
einem HW-Byte zugeordneten Nocken erfolgt die Signalausgabe im Lageregeltakt.

Als digitale Ausgänge der NCK-Peripherie stehen die 4 On-Board-Ausgänge auf der NCU und optional max. 32 externe NCK-Ausgänge zur Verfügung.

Literatur:
/FB2/ Funktionshandbuch Erweiterungsfunktionen; Digitale und analoge NCK–Peripherie (A4)

HW-Zuordnung

Die Zuordnung zu den verwendeten HW-Bytes erfolgt für je 8 Nockenpaare in den folgenden allgemeinen Maschinendaten:

MD10470 SW_CAM_ASSIGN_FASTOUT_1	Hardwarezuordnung für die Ausgabe der Nocken 1 - 8 an die NCK-Peripherie
MD10471 SW_CAM_ASSIGN_FASTOUT_2	Hardwarezuordnung für die Ausgabe der Nocken 9 - 16 an die NCK-Peripherie
MD10472 SW_CAM_ASSIGN_FASTOUT_3	Hardwarezuordnung für die Ausgabe der Nocken 17 - 24 an die NCK-Peripherie
MD10473 SW_CAM_ASSIGN_FASTOUT_4	Hardwarezuordnung für die Ausgabe der Nocken 25 - 32 an die NCK-Peripherie

Hinweis

Je Maschinendatum kann ein HW-Byte für die Ausgabe von 8 Minus-Nockensignalen und ein HW-Byte für 8 Plus-Nockensignale definiert werden.

Weiterhin kann mit den beiden Maschinendaten die Ausgabe der Nockensignale invertiert werden.

Wird das 2. Byte nicht spezifiziert (= "0"), so erfolgt die Ausgabe der 8 Nocken als Verknüpfung der Minus- und Plusnockensignale unter Verwendung der 1. Invertiermaske über das 1. HW-Byte.

Statusabfrage im Teileprogramm

Der Status der HW-Ausgänge kann im Teileprogramm mit der Hauplaufvariablen **$A_OUT[n]** gelesen werden (n = Nr. des Ausgangsbits).

Schaltgenauigkeit

Die Ausgabe an die NCK-Peripherie bzw On-Board-Ausgänge erfolgt im Lageregeltakt. Durch die zeitliche Rasterung des Lageregeltaktes wird die Schaltgenauigkeit der Nockensignale geschwindigkeitsabhängig begrenzt.

Es gilt:

Delta-Pos = V_{act} * Lageregeltakt

Delta-Pos:	Schaltgenauigkeit (bedingt durch Lageregeltakt)
V_{act}:	Aktuelle Achsgeschwindigkeit

Beispiel:

V_{act} = 20 m/min, LR-Takt = 4 ms	Delta-Pos = 1.33 mm
V_{act} = 2000 U/min, LR-Takt = 2 ms	Delta-Pos = 24 Grad

9.3.4 Timergesteuerte Nockensignalausgabe

Timergesteuerte Ausgabe

Eine deutlich höhere Präzision wird durch die taktunabhängige Ausgabe des Nockensignals über einen Timer-Interrupt erreicht.

Es kann mit dem folgenden, allgemeinen Maschinendatum für 4 Nockenpaare die timergesteuerte Ausgabe auf die 4 On-Board-Ausgänge der NCU angewählt werden:
MD10480 SW_CAM_TIMER_FASTOUT_MASK
(Maske für die Ausgabe von Nockensignalen über Timer-Interrupts auf NCU)

Dabei werden die Minus- und Plus-Signale eines Nockenpaares verknüpft als ein Signal ausgegeben.

Signalbildung

Mit dem folgenden Maschinendatum muss zuvor festgelegt sein, auf welche Weise die zu verknüpfenden Signale gebildet werden sollen:
MD10485 SW_CAM_MODE Bit 1

Bit	Signalbildung
nicht gesetzt	Invertierung des Signalverhaltens des Plusnockens bei: Plusnocken - Minusnocken ≥ 180 Grad
gesetzt	**Keine** Invertierung des Signalverhaltens des Plusnockens bei: Plusnocken - Minusnocken ≥ 180 Grad

Hinweis

Diese Funktion arbeitet unabhängig von der getroffenen Zuordnung in Maschinendatum:
MD10470 SW_CAM_ASSIGN_FASTOUT_1 bzw.
MD10471 SW_CAM_ASSIGN_FASTOUT_2 bzw.
MD10472 SW_CAM_ASSIGN_FASTOUT_3 bzw.
MD10473 SW_CAM_ASSIGN_FASTOUT_4.

Das On-Board-Byte darf nicht mehrfach verwendet werden.

Einschränkung

Für die gegenseitige Lage der Nockenpositionen gilt:

Pro IPO-Takt erfolgt nur **eine** timergesteuerte Ausgabe.

Stehen in einem IPO-Takt Signalwechsel für mehr als ein Nockenpaar an, so erfolgt die Ausgabe priorisiert:

Das Nockenpaar mit der niedrigsten Nummer (1...32) bestimmt den Ausgabezeitpunkt für **alle** anstehenden Signale, d.h. die Signalwechsel der anderen Nockenpaare erfolgen zum selben Zeitpunkt.

PLC-Nahtstelle

Am PLC-Interface ist das NCK-Abbild der On-Board-Ausgänge und der Zustand von Plus- und Minusnocken sichtbar.

Diese Signale sind bei **timergesteuerter** Nockenausgabe, wie sie in den beiden folgenden Absätzen beschrieben wird, jedoch nicht relevant bzw. entsprechend ungenau. Die Signale für Plus- und Minusnocken werden im Interpolationstakt synchron (einmalig) gebildet und gemeinsam zur PLC übertragen.

Impulse kürzer als ein Interpolationstakt sind damit in der PLC nicht sichtbar.
Die On-Board-Ausgänge werden asynchron zum Interpolationstakt per Interrupt gesetzt und

rückgesetzt. Der Zustand der On-Board-Ausgänge wird synchron zum Aktualisierungszeitpunkt der PLC-Nahtstelle erfasst und zur PLC übertragen.

Impulse kürzer als ein Interpolationstakt können abhängig vom momentanen Zustand zum Aktualisierungszeitpunkt der PLC-Nahtstelle nicht oder um einen bzw. mehrere IPO-Takte verlängert in der PLC sichtbar sein.

Weitere Einstellungen

Soll das hier beschriebene Verhalten wirksam werden,
muss im folgenden Maschinendatum Bit 0 = 0 gesetzt werden:
MD10485 SW_CAM_MODE

9.3.5 Unabhängige, timergesteuerte Ausgabe der Nockensignale

Unabhängige, timergesteuerte Nockenausgabe

Durch die (Interpolationstakt-) unabhängige, timergesteuerte Nockenausgabe wird jede Schaltflanke getrennt per Interrupt ausgegeben.

Eine gegenseitige Beeinflussung der Nockensignale durch:

- einmalige Ausgabe je Interpolationstakt
- Ausgabezeitpunkt bestimmt durch höchstpriores Nockenpaar (niedrigste Nockenpaar-Nummer)

entfällt.

Für die 4 On-Board-Ausgänge sind **pro Interpolationstakt maximal 8 timergesteuerte Nockenausgaben** zum Setzen/Rücksetzen der On-Board-Ausgänge möglich. Auch für diese Nocken werden die Signalzustände der Plus- und Minusnocken standardmäßig an der PLC-Nahtstelle bereitgestellt. Diese Signale sind jedoch bei timergesteuerter Ausgabe nicht relevant bzw. entsprechend ungenau.

Signalbildung

MD10485 SW_CAM_MODE Bit 1
Mit dem obigen Maschinendatum muss zuvor festgelegt sein, auf welche Weise die zu verknüpfenden Signale gebildet werden sollen:

Bit	Signalbildung
nicht gesetzt	Invertierung des Signalverhaltens des Plusnockens bei: Plusnocken - Minusnocken ≥ 180 Grad
gesetzt	**Keine** Invertierung des Signalverhaltens des Plusnockens bei: Plusnocken - Minusnocken ≥ 180 Grad

Einstellungen

Die Zuordnung der Nockenpaare zu den On-Board-Ausgängen
erfolgt durch das Maschinendatum:
MD10480 SW_CAM_TIMER_FASTOUT_MASK
(Maske für die Ausgabe von Nockensignalen über Timer-Interrupts auf NCU).

Zusätzlich muss diese Bearbeitungsweise über das unten stehende, allgemeine Maschinendatum explizit angewählt werden:
MD10485 SW_CAM_MODE Bit0=1

Hinweis

Diese Funktion arbeitet unabhängig von der getroffenen Zuordnung in Maschinendatum:
MD10470 SW_CAM_ASSIGN_FASTOUT_1 bzw.
MD10471 SW_CAM_ASSIGN_FASTOUT_2 bzw.
MD10472 SW_CAM_ASSIGN_FASTOUT_3 bzw.
MD10473 SW_CAM_ASSIGN_FASTOUT_4.

Das On-Board-Byte darf nicht mehrfach verwendet werden.

9.4 Weg-Zeit-Nocken

Weg-Zeit-Nocken

Unter einem Weg-Zeit-Nocken wird ein Softwarenockenpaar verstanden, das an einer definierten Achsposition einen Impuls einer bestimmten Dauer bereitstellen kann.

Lösung

Die Position wird durch ein Softwarenockenpaar festgelegt.
Die Impulsdauer wird über die Vorhalte-/Verzögerungszeit des Plusnockens definiert.

Durch Maschinendatum wird festgelegt, dass Nockenpaare mit:
Minusnockenposition = Plusnockenposition
als Weg-Zeit-Nocken bearbeitet werden sollen.

Eigenschaften von Weg-Zeit-Nocken

- Die Impulsdauer ist unabhängig von der Achsgeschwindigkeit und der Verfahrrichtungsumkehr.
- Die Impulsdauer ist unabhängig von Veränderung der Achsposition (Preset).
- Die Aktivierung (Einschaltflanke) erfolgt nur beim Überfahren der Nockenposition.

  Eine Verschiebung der Achsposition (z.B. `PRESET`) führt nicht zur Aktivierung.
- Eine Vorhalte-/Verzögerungszeit für den Minus-Nocken ist wirksam und führt zur zeitlichen Verschiebung des Impulses.
- Die Aktiverung (Einschaltflanke) und Impulsdauer sind unabhängig von der Verfahrrichtung.
- Ein nochmaliges Überfahren der Nockenposition bei aktiven Nocken (Richtungsumkehr) führt nicht zur Deaktivierung.
- Die Nockenzeit (Impulsbreite) wird durch ein erneutes Überfahren der Nockenposition nicht unterbrochen sowie die Nockenzeit nicht erneut gestartet.

  Das Verhalten ist besonders bei Modulo-Achsen zu beachten, falls die Nockenzeit größer der Durchfahrzeit des Modulo-Bereichs ist, wird nicht in jeder Umdrehung geschaltet.

Einstellungen

Um einen Weg-Zeit-Nocken festzulegen, müssen Sie folgende Einstellungen treffen:

- **Position**

  Die Position ist durch ein Nockenpaar zu definieren, für das die Minusnockenposition gleich der Plusnockenposition ist.

  Die Festlegung erfolgt durch die Settingdaten:
  SD41500 SW_CAM_MINUS_POS_TAB_1
  bis
  SD41507 SW_CAM_PLUS_POS_TAB_4.

- **Impulsdauer**

  Die Impulsdauer setzt sich additiv zusammen aus den zusammengehörigen Einträgen für das Nockenpaar in:

  MD10461 SW_CAM_PLUS_LEAD_TIME[n]

  SD41521 SW_CAM_PLUS_TIME_TAB_1[n]...

  SD41527 SW_CAM_PLUS_TIME_TAB_4[n]

- **Verschiebung**

  Die zeitliche Verschiebung des Weg-Zeit-Nockens setzt sich additiv zusammen aus den zusammengehörigen Einträgen für das Nockenpaar in:

  MD10460 SW_CAM_MINUS_LEAD_TIME[n]

  SD41520 SW_CAM_MINUS_TIME_TAB_1[n]...

  SD41526 SW_CAM_MINUS_TIME_TAB_4[n]

- **Modus**

  MD10485 SW_CAM_MODE
  Im Maschinendatum muss Bit 2 = 1 gesetzt werden, damit alle Nockenpaare mit gleichen Werten der Minusnockenposition und der Plusnockenposition als Weg-Zeit-Nocken behandelt werden.

9.5 Randbedingungen

Verfügbarkeit der Funktion "Softwarenocken, Wegschaltsignale"

Die Funktion ist eine Option und verfügbar bei:

- SINUMERIK 840D mit NCU 572/573, ab SW2

Erweiterungen

- Die Erweiterung:
  32 statt 16 Nockenpaare
  ist verfügbar ab SW-Stand 4.1.
- Die Erweiterungen:
  Unabhängige, timergesteuerte Ausgabe
  Weg-Zeit-Nocken
  Unterdrückung der Signalinvertierung bei timergesteuerten Ausgaben
  sind verfügbar ab SW-Stand 6.3.

9.6 Datenlisten

9.6.1 Maschinendaten

9.6.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
10260	CONVERT_SCALING_SYSTEM	Grundsystem Umschaltung aktiv
10270	POS_TAB_SCALING_SYSTEM	Maßsystem der Positionstabellen
10450	SW_CAM_ASSIGN_TAB[n]	Zuordnung Softwarenocken zu Maschinenachsen
10460	SW_CAM_MINUS_LEAD_TIME[n]	Vorhalte- bzw. Verzögerungszeit an den Minusnocken 1-16
10461	SW_CAM_PLUS_LEAD_TIME[n]	Vorhalte- bzw. Verzögerungszeit an den Plusnocken 1-16
10470	SW_CAM_ASSIGN_FASTOUT_1	Hardwarezuordnung für die Ausgabe der Nocken 1-8 an die NCK-Peripherie
10471	SW_CAM_ASSIGN_FASTOUT_2	Hardwarezuordnung für die Ausgabe der Nocken 9-16 an die NCK-Peripherie
10472	SW_CAM_ASSIGN_FASTOUT_3	Hardwarezuordnung für die Ausgabe der Nocken 17-24 an die NCK-Peripherie
10473	SW_CAM_ASSIGN_FASTOUT_4	Hardwarezuordnung für die Ausgabe der Nocken 25-32 an die NCK-Peripherie
10480	SW_CAM_TIMER_FASTOUT_MASK	Maske für die Ausgabe von Nockensignalen über Timer-Interrupts auf NCU
10485	SW_CAM_MODE	Verhalten der SW-Nocken

9.6.2 Settingdaten

9.6.2.1 Allgemeine Settingdaten

Nummer	Bezeichner: $SN_	Beschreibung
41500	SW_CAM_MINUS_POS_TAB_1[n]	Position Minusnocken 1-8
41501	SW_CAM_PLUS_POS_TAB_1[n]	Position Plusnocken 1-8
41502	SW_CAM_MINUS_POS_TAB_2[n]	Position Minusnocken 9-16
41503	SW_CAM_PLUS_POS_TAB_2[n]	Position Plusnocken 9-16
41504	SW_CAM_MINUS_POS_TAB_3[n]	Position Minusnocken 17-24
41505	SW_CAM_PLUS_POS_TAB_3[n]	Position Plusnocken 17-24
41506	SW_CAM_MINUS_POS_TAB_4[n]	Position Minusnocken 25-32
41507	SW_CAM_PLUS_POS_TAB_4[n]	Position Plusnocken 25-32

Nummer	Bezeichner: $SN_	Beschreibung
41520	SW_CAM_MINUS_TIME_TAB_1[n]	Vorhalte- bzw. Verzögerungszeit an den Minusnocken 1-8
41521	SW_CAM_PLUS_TIME_TAB_1[n]	Vorhalte- bzw. Verzögerungszeit an den Plusnocken 1-8
41522	SW_CAM_MINUS_TIME_TAB_2[n]	Vorhalte- bzw. Verzögerungszeit an den Minusnocken 9-16
41523	SW_CAM_PLUS_TIME_TAB_2[n]	Vorhalte- bzw. Verzögerungszeit an den Plusnocken 9-16
41524	SW_CAM_MINUS_TIME_TAB_3[n]	Vorhalte- bzw. Verzögerungszeit an den Minusnocken 17-24
41525	SW_CAM_PLUS_TIME_TAB_3[n]	Vorhalte- bzw. Verzögerungszeit an den Plusnocken 17-24
41526	SW_CAM_MINUS_TIME_TAB_4[n]	Vorhalte- bzw. Verzögerungszeit an den Minusnocken 25-32
41527	SW_CAM_PLUS_TIME_TAB_4[n]	Vorhalte- bzw. Verzögerungszeit an den Plusnocken 25-32

9.6.3 Signale

9.6.3.1 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Nocken-Aktivierung	DB31, ...DBX2.0	-

9.6.3.2 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Nocken aktiv	DB31, ...DBX62.0	-

N4: Stanzen und Nibbeln - nur 840D sl

10

10.1 Kurzbeschreibung

Unterfunktionen

Die stanz- und nibbelspezifischen Funktionen setzen sich zusammen aus:

- Hubsteuerung
- Automatische Wegaufteilung
- Drehbares Unter- und Oberwerkzeug
- Pratzenschutz

Die Aktivierung und Deaktivierung erfolgt über Sprachbefehle.

10.2 Hubsteuerung

10.2.1 Allgemeines

Funktionalität

Die Hubsteuerung dient der eigentlichen Bearbeitung des Werkstücks. Nach Erreichen der Position wird über ein Ausgangssignal der NC die Stanze aktiviert. Die Stanzeinheit quittiert ihre Stößelbewegung mit einem Eingangssignal an der NC. In dieser Zeit darf keine Achsbewegung stattfinden. Nach Beendigung des Stanzvorgangs wird neu positioniert.

Schnelle Signale

Zur direkten Kommunikation zwischen NC und Stanzeinheit werden "schnelle Signale" verwendet. Sie dienen in Verbindung mit der Stanze dazu, eine hohe Anzahl von Löchern pro Minute zu erzielen, da die Positionierungen als Totzeiten für die Bearbeitung angesehen werden.

PLC-Signale

Zeitlich unkritische Signale wie Überwachungs- oder Freigabe-Signale sind über PLC-Nahtstellensignale realisiert.

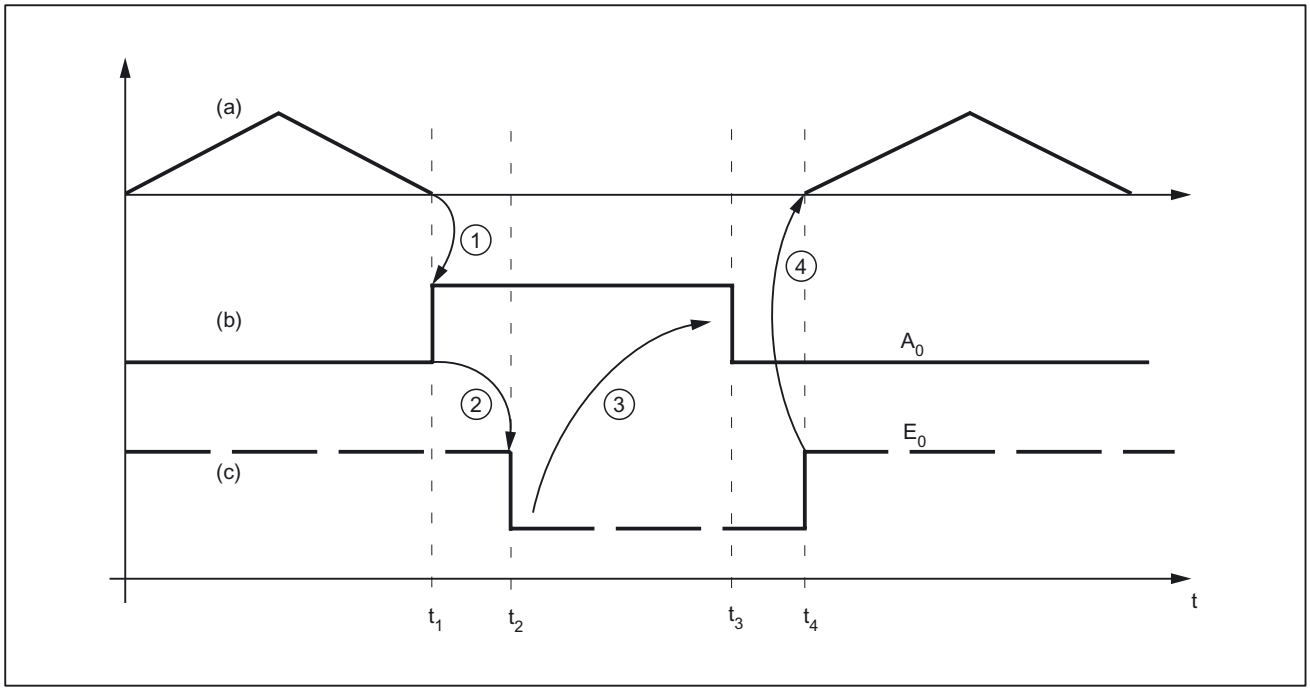
10.2.2 Schnelle Signale

Funktionalität

Die schnellen Signale dienen zur Synchronisation zwischen der NC und der Stanzeinheit. Sie sorgen zum einen dafür, dass über einen schnellen Ausgang der Stanzhub erst dann ausgelöst wird, wenn das Blech zum Stillstand gekommen ist. Zum anderen wird über einen schnellen Eingang das Blech so lange nicht bewegt, wie das Stanzwerkzeug sich im Eingriff zum Blech befindet.

Für die Steuerung der Stanzeinheit werden die schnellen digitalen Ein- und Ausgänge der Steuerung verwendet.

Nachfolgendes Bild stellt die Signalfolge schematisch dar.



(a) Achsbewegung der Maschine als Funktion v(t)

(b) Signal "Hubauslösung"

(c) Signal "Hub aktiv"

Bild 10-1 Signalverlauf

Hinweis

Das Signal "Hub aktiv" ist aus Gründen der Drahtbruchüberwachung high-aktiv.

Die zeitliche Abfolge beim Stanzen und Nibbeln wird durch die beiden Signale A_0 und E_0 gesteuert:

A_0	Wird vom NCK gesetzt und ist identisch mit der Hubauslösung.
E_0	Beschreibt den Zustand der Stanzeinheit und ist identisch mit dem Signal "Hub aktiv".

Durch die Signalzustände werden die Zeiten t_1 bis t_4 wie folgt charakterisiert und definiert:

t_1	Zum Zeitpunkt t_1 ist die Relativbewegung des Werkstücks (Blech) zum Stanzwerkzeug beendet. Je nach definiertem Kriterium für die Hubauslösung (siehe Thema "Kriterien für die Hubauslösung") wird der schnelle Ausgang A_0 für die Stanzauslösung gesetzt ①.
t_2	Zum Zeitpunkt t_2 meldet die Stanzeinheit über den schnellen Eingang E_0 die Stößelbewegung. Dies ist durch das Signal A_0 getriggert ②. Das Signal E_0 ist aus Sicherheitsgründen high-aktiv (bei Drahtbruch ist immer "Hub aktiv" gesetzt und es findet keine Achsbewegung statt). Das Signal "Hub aktiv" wird erst dann wieder zurückgesetzt, wenn das Werkzeug das Blech verlassen hat (t_4).
t_3	Auf das Signal "Hub aktiv" reagiert die NC zum Zeitpunkt t_3 mit der Wegnahme des Signals "Hubauslösung" ③. Ab diesem Zeitpunkt ist die NC in Wartestellung. Sie wartet nur noch auf die Wegnahme des "Hub aktiv"-Signals, um die nächste Achsbewegung zu starten. Erst nach Löschen des Signals A_0 ist die nächste Hubauslösung möglich.
t_4	Zum Zeitpunkt t_4 ist der Stanzvorgang beendet, d. h. der Stößel hat das Blech wieder verlassen. Die NC reagiert bei Flankenwechsel des Signals E_0 mit dem Start der Achsbewegung ④. Die Reaktion der NC auf den Flankenwechsel ④ ist unter "Achsstart nach Stanzen" beschrieben.

Hinweis

Die Hubdauer ist durch die Zeit $\Delta t_h = t_4 - t_1$ bestimmt.

Additiv ergeben sich Reaktionszeiten im Zeitpunkt t_4 zwischen Flankenwechsel von E_0 und dem Start der Achsbewegung.

10.2.3 Kriterien für die Hubauslösung

Hub auslösen

Die Hubauslösung darf frühestens zu dem Zeitpunkt gesetzt werden, zu dem sichergestellt werden kann, dass die Achsen zum Stillstand gekommen sind. Dadurch wird sichergestellt, dass zum Zeitpunkt des Stanzens keinerlei Relativbewegung zwischen dem Stanzwerkzeug und dem Blech in der Bearbeitungsebene stattfindet.

Im folgenden Bild wird auf die unterschiedlichen Kriterien der Hubauslösung eingegangen.

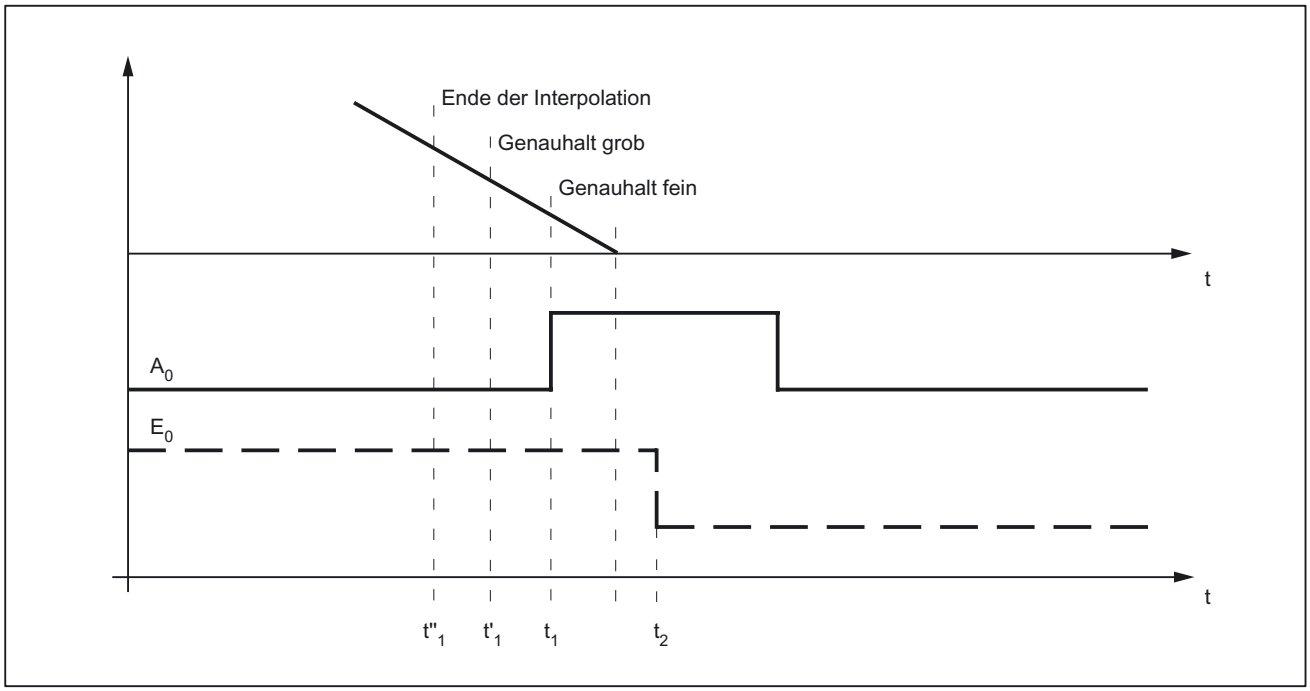


Bild 10-2 Signalverlauf: Kriterien für die Hubauslösung

Die Zeitspanne zwischen t_1 und t_2 ist durch die Reaktion der Stanzeinheit auf das Setzen des Ausgangs A_0 gegeben. Diese ist nicht beeinflussbar, sehr wohl aber als Vorlaufzeit zur Minimierung von Totzeiten nutzbar.

Im Bild ist die Standardeinstellung dargestellt, bei der nach Erreichen des "Genauhalt-fein-Fensters" der Ausgang gesetzt wird (`G601`; Standardeinstellung der G-Gruppe 12). Die Zeiten für die Stanzauslösung t"$_1$ und t'$_1$ werden durch Verwendung von `G602` und `G603` erzielt (siehe folgende Tabelle).

Programmierung	Wirkung	Beschreibung
`G603`	Stopp der Interpolation	Die Interpolation erreicht das Satzende. In diesem Falle bewegen sich die Achsen noch solange, bis der Nachlauf abgebaut ist. Das heißt, das Signal wird deutlich vor dem Achsstillstand ausgegeben (siehe t"$_1$).
`G602`	Erreichen des groben Inpositionsfensters	Das Signal wird ausgegeben, wenn die Achsen das grobe Inpositionsfenster erreicht haben. Wählt man dieses Kriterium für die Ausgabe der Hubauslösung, so kann der Zeitpunkt der Hubauslösung über die Größe des Inpositionsfensters variiert werden (siehe t'$_1$).
`G601`	Erreichen des feinen Inpositionsfensters	In diesem Falle ist bei guter Einstellung der Achsdaten in jedem Fall sichergestellt, dass die Maschine zum Zeitpunkt des Stanzens zum Stillstand gekommen ist. Dieser Fall führt auch zu einer maximalen Totzeit (siehe t_1).

Hinweis

Die Löschstellung der G-Gruppe mit `G601`, `G602` und `G603` (G-Gruppe 12) wird festgelegt über das Maschinendatum:

MD20150 $MC_GCODE_RESET_VALUES[11]

Standardmäßig ist `G601` eingestellt.

G603

Je nach Geschwindigkeit und Maschinendynamik verstreichen nach Interpolationsende ca. 3 - 5 Interpolationstakte bis die Achsen zum Stillstand kommen.

MD26018 $MC_NIBBLE_PRE_START_TIME
In Verbindung mit dem obigen Maschinendatum lässt sich der Zeitpunkt zwischen Erreichen des Interpolationsendes und dem Setzen des schnellen Ausgangs für "Hub ein" verzögern und damit optimieren.

Neben MD26018 existiert das Settingdatum:

SD42402 $SC_NIBPUNCH_PRE_START_TIME

SD42402 kann aus dem Teileprogramm heraus geändert und somit je nach Fortgang der Teileprogrammbearbeitung dem Stanzprozess angepasst werden.

Für die Verzögerungszeit gilt folgendes:

MD26018 = 0	→ SD42402 wirkt
MD26018 ≠ 0	→ MD26018 wirkt

Ist die Funktion "Stanzen mit Verweilzeit, `PDELAYON`" aktiv, so wirkt die in Verbindung mit dieser Funktion eingestellte Verweilzeit. Sowohl MD26018 als auch SD42402 bleiben wirkungslos.

10.2.4 Achsstart nach Stanzen

Eingangssignal "Hub ein"

Der Start der Achsbewegung nach einer Hubauslösung wird über das Eingangssignal "Hub ein" gesteuert.

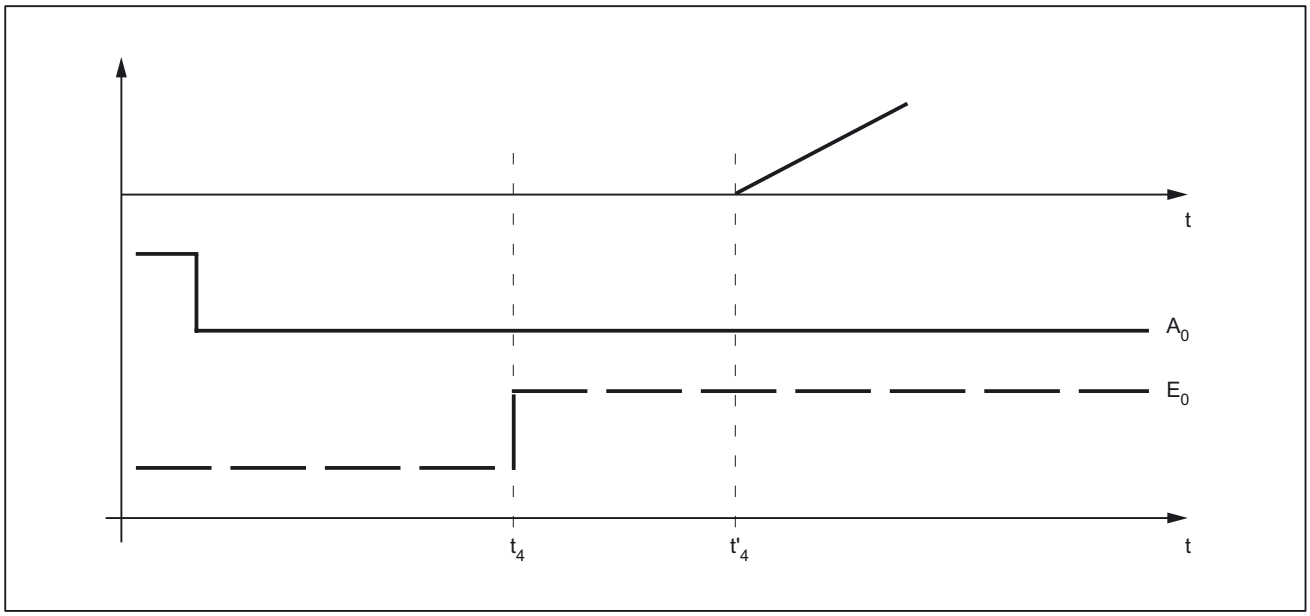


Bild 10-3 Signalverlauf: Achsstart nach Stanzen

Dabei ergibt sich die Zeitspanne zwischen t_4 und t'_4 als schaltzeitbedingte Reaktionszeit. Sie wird bestimmt durch die Interpolationsabtastzeit und durch die programmierte Stanz-/Nibbelart.

PON/SON

Bei Steuerung der Stanzeinheit über PON/SON errechnet sich die Verzögerungszeit maximal zu:

$|t'_4 - t_4|$ = 3 x Interpolationstakt

PONS/SONS

Wird die Stanze über PONS/SONS gesteuert, so wird die Verzögerungszeit bestimmt durch:

$|t'_4 - t_4| \leq$ 3 x Lagereglertakt

Voraussetzung: Hubdauer ($t_4 - t_2$) > 4 Interpolationstakte

10.2.5 Stanz- und nibbelspezifische PLC-Signale

Funktion

Zusätzlich zu den Signalen zur unmittelbaren Hubsteuerung existieren Kanal-spezifische PLC-Nahtstellensignale. Diese dienen zum einen zur Beeinflussung des Stanzprozesses, andererseits zur Statusanzeige.

Signale

Signal	Wirkung
DB21, ... DBX3.0 (Keine Hubfreigabe)	Unterbindet jede Stanzauslösung der NC. Die NC wartet, bis die Freigabe vorhanden ist, um das Teileprogramm fortzusetzen.
DB21, ... DBX3.2 (Hubunterdrückung)	Ermöglicht eine Abarbeitung des Teileprogramms ohne Stanzauslösung (Trockenlauf). Bei aktiver Wegaufteilung verfahren die Achsen in "Stop and Go"-Betrieb.
DB21, ... DBX3.4 (Verzögerter Hub)	Aktiviert verzögerte Hubausgabe, wie sie durch `PDELAYON` möglich ist.
DB21, ... DBX3.1 (Manuelle Hubauslösung)	Ermöglicht eine Stanzauslösung über den Bediener (über PLC gesteuert) auch ohne Teileprogrammbearbeitung. Die manuelle Hubauslösung wird quittiert mit dem Signal: DB21, .. DBX38.1 (Quittung manuelle Hubauslösung)

10.2.6 Stanz- und nibbelspezifische Reaktionen auf Standard-PLC-Signale

DB21, ... DBX12.3 (Vorschub Halt)

Beim Nahtstellensignal:
DB21, ... DBX12.3 (Vorschub Halt)
reagiert die NC bezüglich der Hubsteuerung wie folgt:

Signal wird vor dem Zeitpunkt t_1 erkannt:	Die Hubauslösung wird unterdrückt. Der nächste Hub wird erst nach dem nächsten Start bzw. nach Wegnahme von "Vorschub Halt" ausgelöst. Die Fortsetzung der Bearbeitung erfolgt dann wie im Falle ohne Unterbrechung.
Signal wird an dem Zeitpunkt t_1 erkannt:	Der aktuelle Hub wird noch zu Ende geführt. Die NC verharrt anschließend in dem durch t_4 charakterisierten Zustand. Um dieses Verhalten problemlos zu ermöglichen, wird auf eine Zeitüberwachung der Signale "Hub aktiv" und "Hubauslösung" verzichtet.

10.2.7 Signalüberwachung

Oszillierendes Signal

Durch das Altern der Stanzhydraulik kann es vorkommen, dass das "Hub aktiv"-Signal nach dem Hub-Ende durch das Überschwingen des Stößels oszilliert.

In diesem Fall kann abhängig vom Maschinendatum:
MD26020 $MC_NIBBLE_SIGNAL_CHECK
ein Alarm erzeugt werden (Alarm 22054 "unsauberes Stanzsignal").

Reset-Verhalten

Bei NCK-Reset wird das Nahtstellensignal:
DB21, ... DBX38.0 (Hubauslösung aktiv)
sofort weggenommen und nicht auf die Quittung durch den schnellen Eingang gewartet.

Ein aktuell aktivierter Hub kann nicht unterdrückt werden.

10.3 Aktivierung und Deaktivierung

10.3.1 Sprachbefehle

Die Aktivierung und Deaktivierung der Stanz- und Nibbelfunktionen erfolgt über projektierbare Sprachbefehle. Die in früheren Systemen gebräuchlichen speziellen M-Funktionen sind abgelöst.

Literatur:
/PGA/ Programmierhandbuch Arbeitsvorbereitung

Gruppen

Die Sprachbefehle sind in folgende Gruppen eingeteilt:

Gruppe 35	
Über die folgenden Sprachbefehle werden die eigentlichen stanz- und nibbelspezifischen Funktionen aktiviert bzw. deaktiviert:	
PON	= Stanzen ein
SON	= Nibbeln ein
PONS	= Stanzen ein, Aktivierung im Lageregler
SONS	= Nibbeln ein, Aktivierung im Lageregler
SPOF	= Stanzen/Nibbeln aus

Gruppe 36	
Diese Gruppe beinhaltet die Befehle, die nur vorbereitenden Charakter haben und die konkrete Ausprägung der Stanzfunktion bestimmen:	
PDELAYON	= Stanzen mit Verzögerung ein
PDELAYOF	= Stanzen mit Verzögerung aus
Da für diese vorbereitenden Funktionen im Normalfall von der PLC einige Vorarbeiten zu leisten sind, werden sie vor den aktivierenden Befehlen programmiert.	

Gruppe 38	
Diese Gruppe beinhaltet die Befehle für das Umschalten auf ein zweites Stanz-Interface. Dieses kann zum Beispiel für eine zweite Stanzeinheit oder eine Schlagschere benutzt werden. Über Maschinendaten wird ein zweites I/O-Paar definiert, das für die Stanzfunktionalität verwendet werden kann.	
SPIF1	= Erstes Interface ist aktiv
SPIF2	= Zweites Interface ist aktiv

Hinweis

Innerhalb einer G-Code-Gruppe kann jeweils nur eine Funktion aktiv sein (ähnlich zum Beispiel den unterschiedlichen Interpolationsarten G0, G1, G2, G3 usw., die sich auch gegenseitig ausschließen).

SPOF

Stanzen und Nibbeln aus

Die Funktion SPOF beendet alle Stanz- und Nibbelfunktionen. In diesem Zustand reagiert der NCK weder auf das Signal "Hub aktiv" noch auf die stanz-/nibbel-spezifischen PLC-Signale.

Wird SPOF gemeinsam mit einem Wegbefehl in einem Satz programmiert (und in allen nachfolgenden, falls nicht mit SON oder PON Stanzen/Nibbeln aktiviert wird) fährt die Maschine die programmierte Position ohne Stanzauslösung an. SPOF wählt SON, SONS, PON und PONS ab und entspricht dem Reset-Zustand.

Programmierbeispiel:

```
:
:
N20 G90 X100 SON              ;  Nibbeln aktivieren
N25 X50 SPOF                  ;  Nibbeln deaktivieren,
                              ;  Positionieren ohne Hubauslösung
:
:
```

SON

Nibbeln ein

SON schaltet die Nibbelfunktion ein und wählt die anderen Funktionen der G-Gruppe 35 ab (z. B. PON).

Im Gegensatz zum Stanzen erfolgt der erste Hub bereits im Startpunkt des aktivierenden Satzes, d. h. vor der ersten Bewegung der Maschine.

SON wirkt modal, d. h. es bleibt so lange aktiv, bis entweder SPOF oder PON programmiert wird oder das Programmende erreicht ist.

In Sätzen ohne Verfahrinformation bezüglich der als Stanz- oder Nibbelachsen gekennzeichneten Achsen (typischerweise diejenigen der aktiven Ebene) wird die Hubauslösung unterdrückt. Soll trotzdem ein Hub ausgelöst werden, so muss explizit eine der Stanz-/Nibbelachsen mit dem Verfahrweg 0 programmiert werden. Ist der erste Satz mit SON ein Satz ohne Verfahrinformation im genannten Sinne, erfolgt in diesem Satz nur ein einziger Hub, da Start- und Endpunkt zusammenfallen.

Programmierbeispiel:

```
:
:
```

```
N70 X50 SPOF                ;  Positionieren ohne Stanzauslösung
N80 X100 SON                ;  Nibbeln aktivieren, Auslösung eines Hubs vor der
                            ;  Bewegung (X=50) und am Ende der programmierten
                            ;  Bewegung (X=100)
:
:
```

SONS

Nibbeln ein (im Lageregeltakt)

SONS verhält sich wie SON. Die Auslösung erfolgt im Lageregeltakt. Damit kann eine zeitliche Optimierung der Hubauslösung und eine Erhöhung der Stanzrate pro Minute erreicht werden.

PON

Stanzen ein

PON aktiviert die Stanzfunktion und deaktiviert SON.

PON wirkt ebenso wie SON modal.

Im Unterschied zu SON wird aber erst im Satzendpunkt, bzw. bei automatischer Wegaufteilung am Ende jedes Teilstückes ein Hub ausgeführt. Bei Sätzen ohne Verfahrinformation ist das Verhalten identisch wie im Falle SON.

Programmierbeispiel:

```
:
:
N100 Y30 SPOF               ;  Positionieren ohne Stanzauslösung
N110 Y100 PON               ;  Stanzen aktivieren, Stanzauslösung am Ende des
                            ;  Positioniervorgangs (Y=100)
:
:
```

PONS

Stanzen ein (im Lageregeltakt)

PONS verhält sich wie PON. Erläuterung siehe SONS.

PDELAYON

Stanzen mit Verzögerung

PDELAYON ist eine vorbereitende Funktion. Dies bedeutet, dass PDELAYON in der Regel vor PON programmiert wird. Nach Erreichen der programmierten Endposition wird der Stanzhub verzögert ausgegeben.

Die Verzögerungszeit in Sekunden wird eingestellt im Settingdatum:

SD42400 $SC_PUNCH_DWELLTIME

Ist der definierte Wert nicht ganzzahlig durch den Interpolationstakt teilbar, so wird auf den nächstgelegenen ganzzahlig teilbaren Wert gerundet.

Die Funktion wirkt modal.

PDELAYOF

Stanzen mit Verzögerung aus

PDELAYOF schaltet das Stanzen mit Verzögerung ab, d. h. es wird normal weitergestanzt. PDELAYON und PDELAYOF bilden eine G-Code-Gruppe.

Programmierbeispiel:

SPIF2 aktiviert das zweite Stanzinterface. D. h. die Hubsteuerung erfolgt über das zweite Paar der schnellen I/O (siehe Maschinendaten MD26004 und MD26006).

```
:
:
N170 PDELAYON X100 SPOF           ;  Positionieren ohne Stanzauslösung, Aktivierung
                                  ;  der verzögerten Stanzauslösung
:
:
N180 X800 PON                     ;  Stanzen aktivieren. Nach Erreichen der
                                  ;  Endposition wird Stanzhub verzögert ausgegeben
:
:
N190 PDELAYOF X700                ;  Stanzen mit Verzögerung deaktivieren, normale
                                  ;  Stanzauslösung an. Ende der programmierten
                                  ;  Bewegung
:
:
:
```

SPIF1

Aktivierung des ersten Stanz-Interfaces

SPIF1 aktiviert das erste Stanz-Interface. D. h. die Hubsteuerung erfolgt über das erste Paar der schnellen I/O (siehe Maschinendaten MD26004 und MD26006).

Nach Reset oder Steuerungshochlauf ist immer das erste Stanz-Interface aktiv. Wird nur ein Interface benutzt, so muss dieses nicht programmiert werden.

SPIF2

Aktivierung des zweiten Stanzinterfaces

SPIF2 aktiviert das zweite Stanzinterface. D. h. die Hubsteuerung erfolgt über das zweite Paar der schnellen I/O (siehe Maschinendaten MD26004 und MD26006).

Programmierbeispiel:

```
:
:
N170 SPIF1 X100 PON          ;  Am Ende des Satzes erfolgt eine Hubauslösung auf
                             ;  dem ersten schnellen Ausgang. Das Signal "Hub
                             ;  aktiv" wird auf dem ersten Eingang überwacht.
:
:
:
:
N180 X800 SPIF2              ;  Die zweite Hubauslösung erfolgt auf dem zweiten
                             ;  schnellen Ausgang. Das Signal "Hub aktiv" wird auf
                             ;  dem zweiten Eingang überwacht.
:
:
:
N190 SPIF1 X700              ;  Die Hubsteuerung für alle weiteren Hübe erfolgt mit
                             ;  dem ersten Interface.
:
```

10.3.2 Funktionserweiterungen

Umschaltbares Interface

Maschinen, die abwechselnd eine zweite Stanzeinheit oder ein vergleichbares Medium nutzen sollen, können auf ein zweites I/O-Paar umgeschaltet werden.

Das zweite I/O-Paar ist über folgende Maschinendaten definierbar:

MD26004 $MC_NIBBLE_PUNCH_OUTMASK

MD26006 $MC_NIBBLE_PUNCH_INMASK

Die Umschaltung erfolgt über die Befehle `SPIF1` oder `SPIF2`.

Für beide Interfaces ist die volle Stanz-/Nibbel-Funktionalität nutzbar.

Beispiel:

Hardware-Zuordnung bei Hubsteuerung

Jeweils das schnelle Byte auf der CPU als schnelles Stanz-Interface definieren:

MD26000 $MC_PUNCHNIB_ASSIGN_FASTIN = 'H00030001' → Byte 1

MD26002 $MC_PUNCHNIB_ASSIGN_FASTOUT = 'H00000001'

Anmerkung:

Das erste und zweite Bit sind invertiert.

Maske für schnelle Ausgabe- und Eingabebits:

MD26004 $MC_NIBBLE_PUNCH_OUTMASK[0]	= 1	Erstes Interface Outputbit → Bit 1 SPIF1
MD26004 $MC_NIBBLE_PUNCH_OUTMASK[1]	= 2	Zweites Interface Outputbit → Bit 2 SPIF2
MD26006 $MC_NIBBLE_PUNCH_INMASK[0]	= 1	Erstes Interface Inputbit → Bit 1 SPIF1
MD26006 $MC_NIBBLE_PUNCH_INMASK[1]	= 2	Zweites Interface Inputbit → Bit 2 SPIF2

Automatisch aktivierbare Vorauslösezeit

Totzeiten aufgrund der Reaktionszeit der Stanzeinheit lassen sich minimieren, wenn der Hub schon vor Erreichen des Interpolationsfensters der Achsen ausgelöst werden kann. Der Referenzpunkt dafür ist das Interpolationsende. Die Hubauslösung wird automatisch mit `G603` aktiv und verzögert sich um den eingestellten Wert gegenüber dem Zeitpunkt des Erreichens des Interpolationsendes.

Die Verzögerungszeit für die Hubauslösung ist einstellbar über das Maschinendatum:

MD26018 $MC_NIBBLE_PRE_START_TIME

Beispiel:

Bei einem IPO-Takt von 5 ms soll 2 Takte nach Erreichen des Interpolationsendes ein Hub ausgelöst werden:

⇒ MD26018 $MC_NIBBLE_PRE_START_TIME = 0,01 [s]

Eine Vorauslösezeit ist auch einstellbar im Settingdatum:

SD42402 $SC_NIBPUNCH_PRE_START_TIME

Der Wert wird aber nur wirksam, wenn MD26018 = 0 gesetzt wurde.

Überwachung des Eingangssignals

Wenn das Signal "Hub aktiv" beispielsweise durch Stößelüberschwingungen zwischen den Hüben schwankt, kann zusätzlich zum Stopp der Interpolation die Meldung "Unsauberes Stanzsignal" ausgegeben werden.

Die Ausgabe der Meldung ist abhängig von der Einstellung im Maschinendatum:

MD26020 $MC_NIBBLE_SIGNAL_CHECK

MD26020 = 0	Kein Alarm
MD26020 = 1	Alarm

Mindestzeit zwischen zwei Hüben

Ein zeitlicher Mindestabstand zwischen zwei aufeinander folgenden Hüben kann eingestellt werden über das Settingdatum:

SD42404 $SC_MINTIME_BETWEEN_STROKES

Beispiel:

Zwischen zwei Hubauslösungen sollen unabhängig von der räumlichen Entfernung mindestens 1,3 Sekunden liegen:

⇒ SD42404 $SC_MINTIME_BETWEEN_STROKES = 1,3 [s]

Ist zusätzlich eine Stanzverweilzeit (`PDELAYON`) programmiert, so wirken beide Zeiten additiv.

Eine eventuell eingestellte Vorauslösezeit bei `G603` ist nur wirksam, wenn das Interpolationsende erst nach Ablauf von SD42404 erreicht wird.

Die programmierte Zeit wird sofort aktiv. Je nach Größe des Satzpuffers lassen sich zuvor programmierte Hübe mit diesem Mindestabstand ausführen. Dies lässt sich durch folgende Programmierung (Beispiel) verhindern:

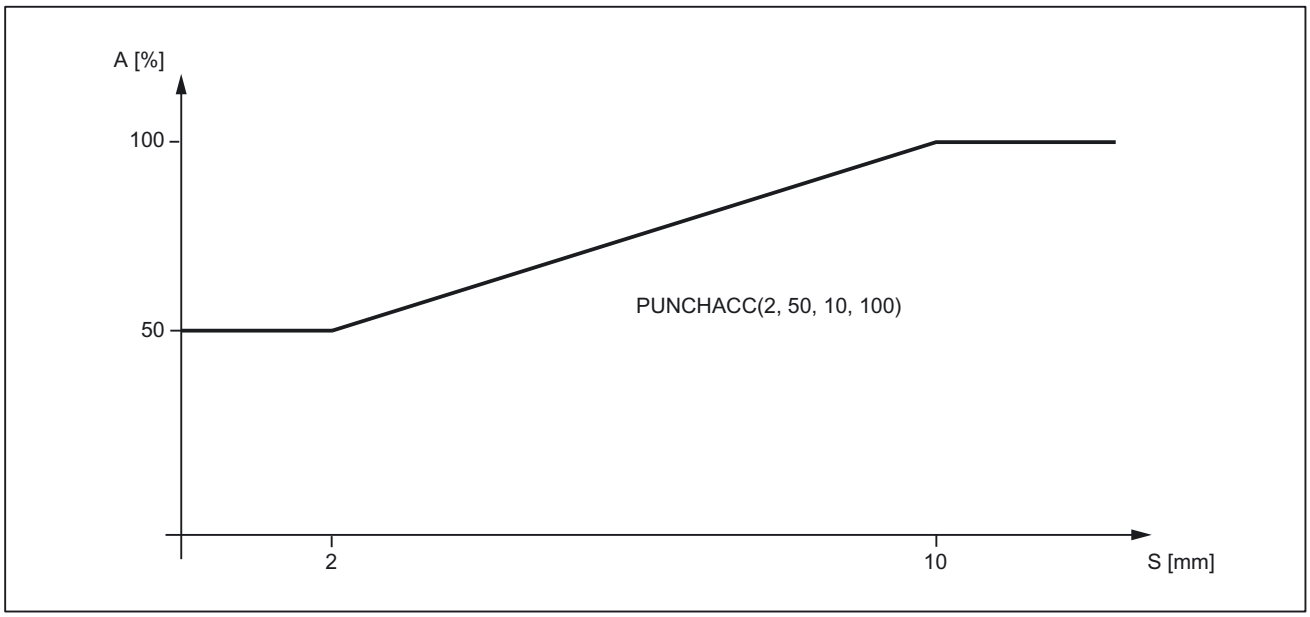
```
N...
N100 STOPRE
N110 $SC_ MINTIME_BETWEEN_STROKES = 1,3
```

Für SD42404 = 0 ist die Funktion nicht aktiv.

Wegabhängige Beschleunigung

Über den Sprachbefehl `PUNCHACC (Smin, Amin, Smax, Amax)` kann eine Beschleunigungskennlinie festgelegt werden. Damit ist es möglich, je nach Lochabstand unterschiedliche Beschleunigungen zu definieren.

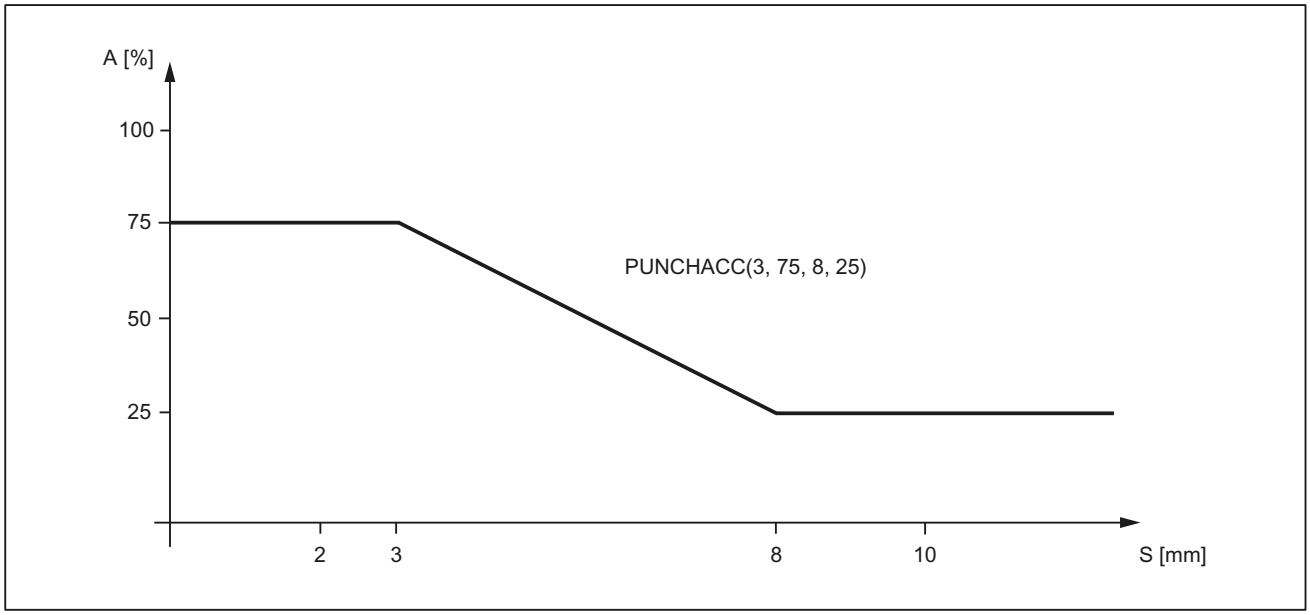
Beispiel 1:



Die Kennlinie legt folgende Beschleunigungen fest:

Lochabstand	Beschleunigung
< 2 mm	Es wird mit einer Beschleunigung von 50 % der Maximalbeschleunigung verfahren.
2 - 10 mm	Die Beschleunigung wird proportional zum Abstand auf 100 % gesteigert.
> 10 mm	Es wird mit einer Beschleunigung von 100 % verfahren.

Beispiel 2:



Die Kennlinie legt folgende Beschleunigungen fest:

Lochabstand	Beschleunigung
< 3 mm	Es wird mit einer Beschleunigung von 75 % der Maximalbeschleunigung verfahren.
3 - 8 mm	Die Beschleunigung wird proportional zum Abstand auf 25 % reduziert.
> 10 mm	Es wird mit einer Beschleunigung von 25 % verfahren.

Wenn vorher über `ACC` eine reduzierte Beschleunigung programmiert wurde, so beziehen sich die über `PUNCHACC` definierten Grenzbeschleunigungen auf die reduzierte Beschleunigung.

Die Abwahl der Funktion erfolgt über:

$S_{min} = S_{max} = 0$

Die vorherige Beschleunigungsprogrammierung über `ACC` bleibt weiter wirksam.

Satzsuchlauf

Beim Satzsuchlauf auf einen Satz mit Nibbelfunktion kann eingestellt werden, ob der Stanzhub am Anfang des Satzes ausgeführt oder unterdrückt wird.

Die Einstellung erfolgt im Maschinendatum:

MD11450 $MN_SEARCH_RUN_MODE

Bit	Wert	Bedeutung
5	0	Stanzhub am Anfang des Satzes wird unterdrückt.
	1	Stanzhub am Anfang des Satzes wird ausgeführt.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1), Kapitel "Satzsuchlauf"

10.3.3 Kompatibilität zu älteren Systemen

Verwendung von M-Funktionen

Mit Hilfe der Makrotechnik ist es nach wie vor möglich, spezielle M-Funktionen statt der Sprachbefehle zu benutzen (Kompatibilität).

Dabei gelten die folgenden Entsprechungen zu älteren Systemen:

```
M20,M23   ≙   SPOF
M22       ≙   SON
M25       ≙   PON
M26       ≙   PDELAYON
```

Hinweis

Die M-Funktionen sind über Maschinendaten projektierbar.

Bei der Zuordnung der M-Funktionen zu den Sprachbefehlen ist die Einteilung der M-Funktionen in Hilfsfunktionsgruppen zu beachten.

Beispiele

```
DEFINE M20 AS SPOF
```
Stanzen/Nibbeln aus

oder

```
DEFINE M20 AS SPOF M=20
```
Stanzen mit Hilfsfunktionsausgabe

`DEFINE M20 AS SPOF PDELAYOF`	Stanzen/Nibbeln aus und Stanzen mit Verzögerung aus
`DEFINE M22 AS SON`	Nibbeln ein
oder	
`DEFINE M22 AS SON M=22`	Nibbeln ein mit Hilfsfunktionsausgabe
`DEFINE M25 AS PON`	Stanzen ein
oder	
`DEFINE M25 AS PON M=25`	Stanzen ein mit Hilfsfunktionsausgabe
`DEFINE M26 AS PDELAYON`	Stanzen mit Verzögerung
oder	
`DEFINE M26 AS PDELAYON M=26`	Stanzen und Hilfsfunktionsausgabe

Programmierbeispiel:

```
:
:
N100 X100 M20              ;  Positionieren ohne Stanzauslösung
N110 X120 M22              ;  Nibbeln aktivieren, vor und nach Bewegung
                              Hubauslösung
:
N120 X150 Y150 M25         ;  Stanzen aktivieren, Hubauslösung am Ende der
                              Bewegung
:
:
```

10.4 Automatische Wegaufteilung

10.4.1 Allgemeines

Funktion

Bei der automatischen Wegaufteilung einer programmierten Verfahrstrecke wird unterschieden zwischen:

- Streckenaufteilung mit maximal programmiertem Teilweg über den Sprachbefehl SPP
- Streckenaufteilung mit programmierter Anzahl von Teilstrecken über den Sprachbefehl SPN

Beide Funktionen generieren selbständig Teilsätze.

In älteren Systemen korrespondieren:

- SPP<Zahl> mit E<Zahl>
- SPN<Zahl> mit H<Zahl>

Da die Adressen E und H jetzt Hilfsfunktionen darstellen, werden zur Vermeidung von Konflikten die Sprachbefehle SPP und SPN verwendet. Eine Kompatibilität zu älteren Systemen ist damit nicht mehr gegeben. Beide Sprachbefehle (SPP und SPN) sind projektierbar.

Hinweis

Die mit SPP programmierten Werte sind entsprechend der Grundeinstellung entweder Angaben in mm oder in inch (entsprechend wie bei den Achsen).

Die automatische Wegaufteilung sorgt bei Linear- und Kreisinterpolation für eine äquidistante Streckenaufteilung.

Bei Programmunterbrechung mit aktiver automatischer Wegaufteilung (SPP/SPN) ist der Wiedereintritt an die Kontur nur am Anfang des aufgeteilten Satzes möglich. Der erste Stanzhub erfolgt am Ende dieses Teilsatzes.

SPP und SPN sind nur aktivierbar wenn Geoachsen projektiert sind.

SPP

Die automatische Streckenaufteilung SPP teilt den programmierten Verfahrweg entsprechend den vorgegebenen Teilstrecken in gleich große Strecken auf.

Es gelten folgende Bedingungen:

- Die Wegaufteilung ist nur wirksam bei aktivem SON oder PON.

  (Ausnahme: MD26014 $MC_PUNCH_PATH_SPLITTING = 1)
- SPP wirkt modal, d. h. die programmierte Teilstrecke wirkt bis sie neu programmiert wird, kann aber satzweise durch SPN unterdrückt werden.
- Die Teilstrecken werden von der Steuerung bei Bedarf abgerundet, so dass eine programmierte Gesamtstrecke in eine ganzzahlige Anzahl von Teilstrecken zerlegt wird.

- Die Teilstrecke wirkt in mm/Hub oder inch/Hub (entsprechend den Achsvorgaben).
- Ist der programmierte SPP-Wert größer als die Verfahrstrecke, dann wird zur programmierten Endposition ohne Wegaufteilung positioniert.
- SPP = 0, Reset bzw. Programmende löschen den programmierten SPP-Wert. Deaktivieren von Stanzen/Nibbeln löscht nicht den programmierten SPP-Wert.

SPN

Die automatische Streckenaufteilung SPN teilt den Verfahrweg in die programmierte Anzahl von Teilstrecken.

Es gelten folgende Bedingungen:

- Die Wegaufteilung ist nur wirksam bei aktivem SON oder PON.

  (Ausnahme: MD26014 $MC_PUNCH_PATH_SPLITTING = 1)
- SPN wirkt satzweise.
- Ein vorher programmierter SPP-Wert wird für diesen Satz unterdrückt, ist aber in nachfolgenden Sätzen wieder aktiv.

Randbedingungen

- Die Wegaufteilung ist bei Geraden- und Kreisinterpolationen wirksam.

  Die Interpolationsart bleibt erhalten, d. h. bei Kreisinterpolationen werden die Kreise abgefahren.
- Wurde in einem Satz sowohl SPN (Hubzahl) als auch SPP (Hubstrecke) programmiert, so wird im aktuellen Satz die Hubanzahl wirksam, während in den darauf folgenden Sätzen die Hubstrecke aktiv wird.
- Die Wegaufteilung ist nur bei aktivem Stanzen oder Nibbeln aktiv.

  (Ausnahme: MD26014 $MC_PUNCH_PATH_SPLITTING = 1).
- Evtl. programmierte Hilfsfunktionen werden vor, während dem ersten oder nach dem letzten Teilsatz ausgegeben.
- Bei Sätzen ohne Verfahrinformationen gelten auch bei SPP- und SPN-Programmierung die gleichen Regeln wie sie für SON und PON festgelegt wurden. D. h., es wird nur dann ein Hub ausgelöst, wenn eine Achsbewegung programmiert wurde.

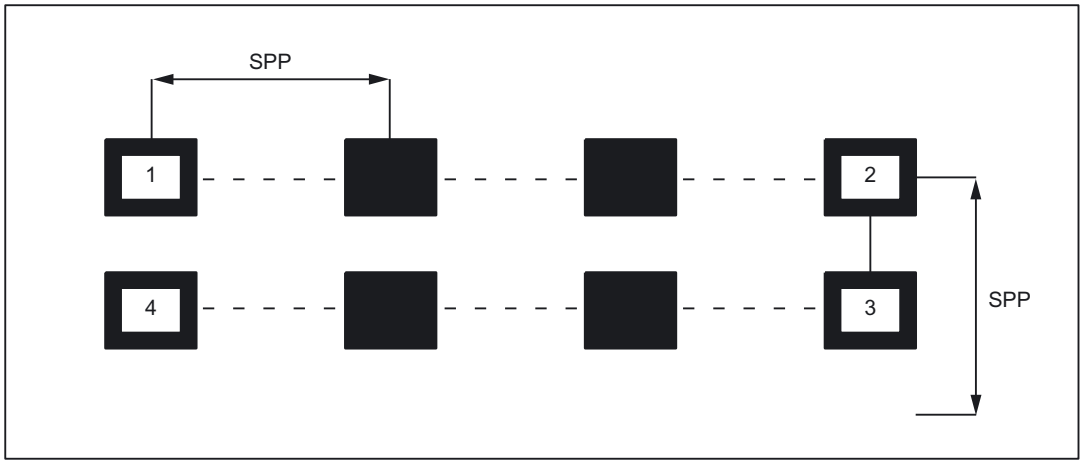
10.4.2 Verhalten bei Bahnachsen

MD26010

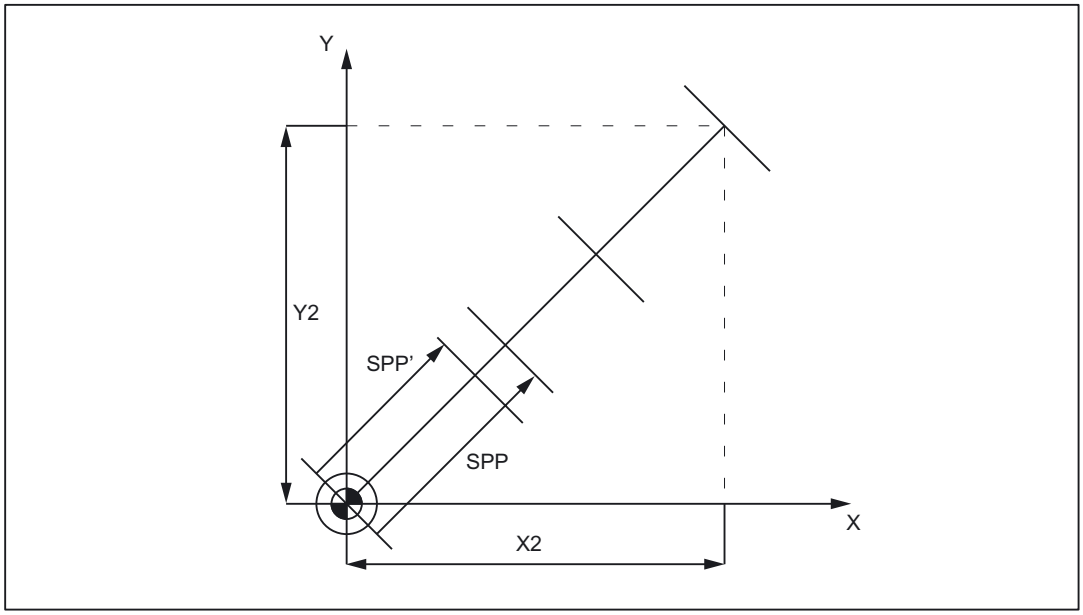
Alle über das Maschinendatum:
MD26010 $MC_PUNCHNIB_AXIS_MASK
definierten und programmierten Achsen werden mit SPP und SPN bis zum Erreichen des programmierten Endpunkts mit gleich großen Teilstrecken bewegt. Dies gilt auch für eine evtl. vorhandene drehbare Werkzeugachse. Das Verhalten kann für Einzelachsen angepasst werden.

Beispiel für SPP

```
N1 G01 X0 Y0 SPOF          ;  Positionierung ohne Stanzauslösung
N2 X75 SPP=25 SON          ;  Nibbeln mit Vorschubwert 25 mm; Stanzauslösung vor
                           ;  der ersten Bewegung und nach jeder Teilstrecke.
:
:
N3 Y10                     ;  Positionierung mit reduziertem SPP-Wert, da
                           ;  Verfahrstrecke < SPP-Wert und Stanzauslösung nach
                              Bewegung.
:
:
:
N4 X0                      ;  Rückpositionieren mit Stanzauslösung nach jeder
                              Teilstrecke.
:
```



Ist die programmierbare Wegaufteilung kein ganzzahliges Vielfaches der Gesamtstrecke, so wird der Vorschubweg reduziert:



X2/Y2: Programmierter Verfahrweg
SPP: Programmierter SPP-Wert
SPP': Automatisch abgerundeter Verschiebweg

Bild 10-4 Streckenaufteilung

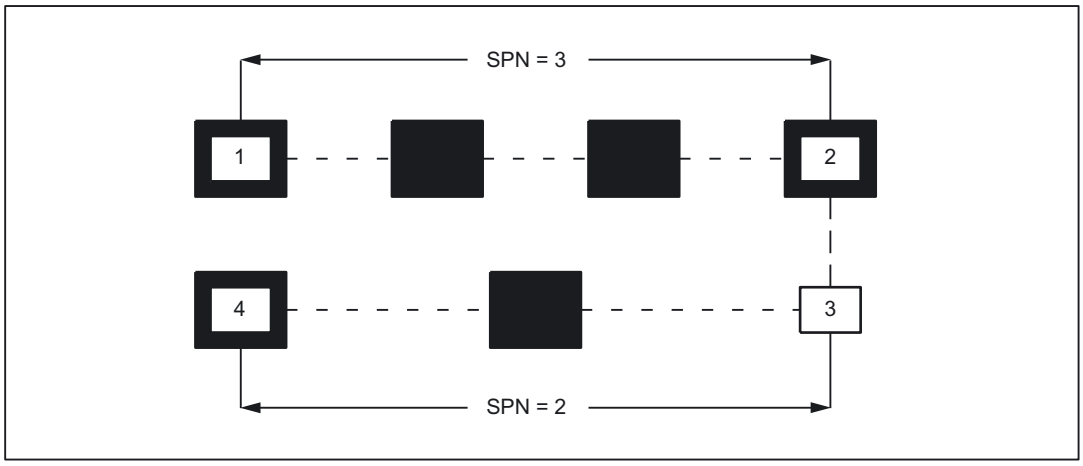
Beispiel für SPN

Über `SPN` wird die Anzahl der Teilstrecken pro Satz programmiert.

Ein über `SPN` programmierter Wert wirkt sowohl beim Stanzen als auch beim Nibbeln satzweise. Der einzige Unterschied zwischen den beiden Betriebsarten betrifft auch hier den ersten Hub. Dieser wird beim Nibbeln normalerweise am Anfang des ersten Teilschrittes, beim Stanzen an dessen Ende ausgeführt. Das bedeutet, wenn n Teilstrecken programmiert sind, so werden beim Stanzen n Hübe, beim Nibbeln jedoch n+1 Hübe ausgeführt. Außerdem erfolgt in Fällen ohne Verfahrinformation nur ein einziger Hub, auch wenn mehrere programmiert sind. Sollte es nötig sein, an einer bestimmten Stelle mehrere Hübe zu erzeugen, so müssen dafür entsprechend viele Sätze ohne Verfahrinformation programmiert werden.

```
N1 G01 X0 Y0 SPOF          ;  Positionieren ohne Stanzauslösung
N2 X75 SPN=3 SON           ;  Nibbeln aktivieren. Die Gesamtstrecke wird in 3
                           ;  Teilstrecken aufgeteilt. Vor der ersten Bewegung und
                           ;  am Ende jeder Teilstrecke wird ein Hub ausgelöst.
:
:
:
:
N3 Y10 SPOF                ;  Positionierung ohne Stanzauslösung
N4 X0 SPN=2 PON            ;  Stanzen aktivieren. Die Gesamtstrecke wird in 2
```

```
                          ;  Teilstrecken aufgeteilt. Da Stanzen aktiviert wurde,
                          ;  wird der erste Hub am Ende der ersten Teilstrecke
                             ausgelöst.
:
:
:
:
```



Beispiel

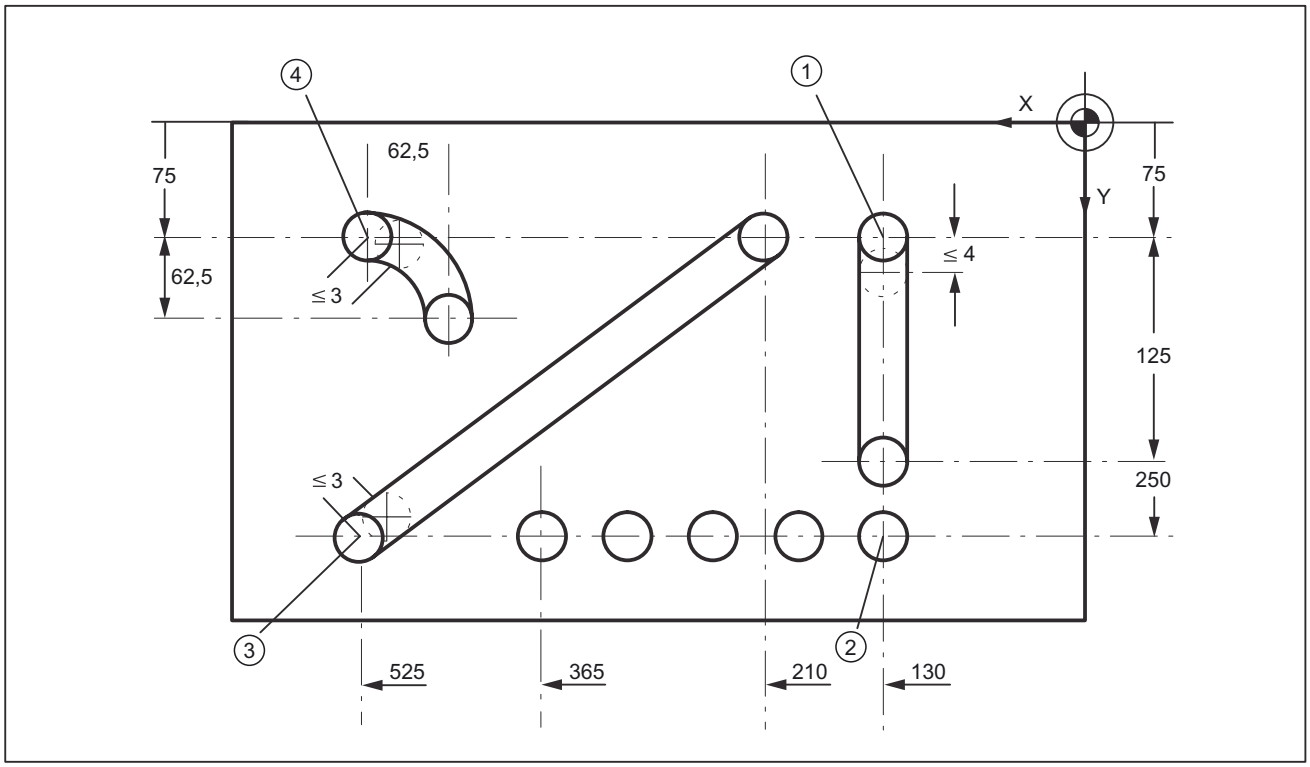


Bild 10-5 Werkstück

Programm-Ausschnitt

```
N100 G90 X130 Y75 F60 SPOF                ; Positionieren auf Startpunkt ① der
                                          ; senkrechten Nibbelstrecken
N110 G91 Y125 SPP=4 SON                   ; Endpunktkoordinaten (inkrementell);
                                          ; Teilstrecke: 4 mm, Nibbeln aktivieren
N120 G90 Y250 SPOF                        ; Absolute Vermaßung, Positionieren auf
                                          ; Startpunkt ② der waagrechten
                                            Nibbelstrecke
N130 X365 SPN=4 SON                       ; Endpunktkoordinaten, 4 Teilstrecken,
                                          ; Nibbeln aktivieren
N140 X525 SPOF                            ; Positionieren auf Startpunkt ③ der
                                          ; schrägen Nibbelstrecke
N150 X210 Y75 SPP=3 SON                   ; Endpunktkoordinaten Teilstrecke: 3 mm,
                                          ; Nibbeln aktivieren
N160 X525 SPOF                            ; Positionieren auf Startpunkt ④ der
                                          ; Nibbelstrecke auf Teilkreisbahn
N170 G02 G91 X-62.5 Y62.5 I0 J62.5 SON    ; Inkrementelle Kreisinterpolation mit
                                          ; Interpolationsparametern, Nibbeln
                                            aktivieren
N180 G00 G90 Y300 SPOF                    ; Positionieren
```

10.4.3 Verhalten im Zusammenhang mit Einzelachsen

MD26016

Standardmäßig wird der Weg der Einzelachsen, die neben den Bahnachsen programmiert sind, gleichmäßig auf die erzeugten Zwischensätze verteilt.

Im nachfolgenden Beispiel ist die zusätzliche Rundachse C als Synchronachse definiert.

Ist die Achse zusätzlich als "Stanz-Nibbel-Achse" definiert:
MD26010 $MC_PUNCHNIB_AXIS_MASK = 1,
lassen sich in Abhängigkeit des Maschinendatums:
MD26016 $MC_PUNCH_PARTITION_TYPE
unterschiedliche Verhalten der Synchronachse erzeugen.

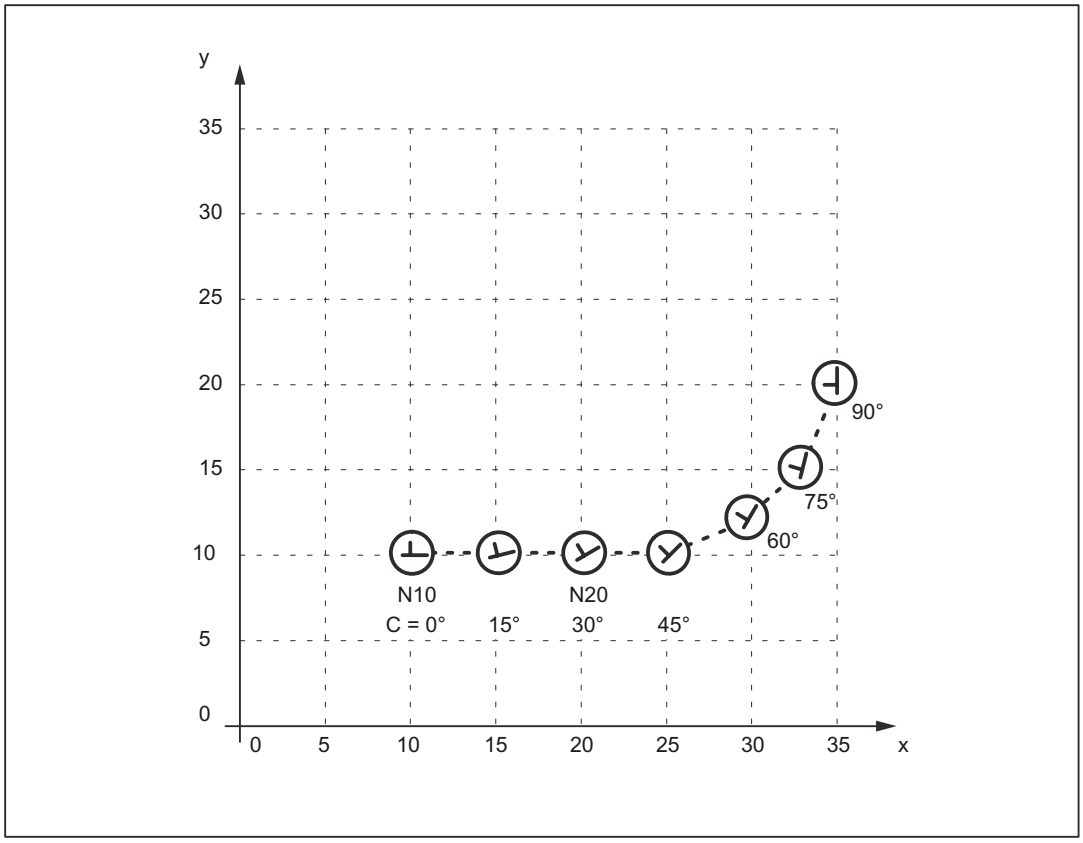
Programmierbeispiel:

```
N10 G90 G1 PON X10 Y10 C0 F10000
N20 SPP=5 X25 C45
N30 G3 SPN=3 X35 Y20 I0 J10 C90
```

MD26016 $MC_PUNCH_PARTITION_TYPE=0 (Standardeinstellung)

In dieser Einstellung verhalten sich die Achsen standardmäßig, d. h. in allen Interpolationsarten werden die programmierten Zusatzachsbewegungen auf die erzeugten Zwischensätze der aktiven Wegaufteilung aufgeteilt.

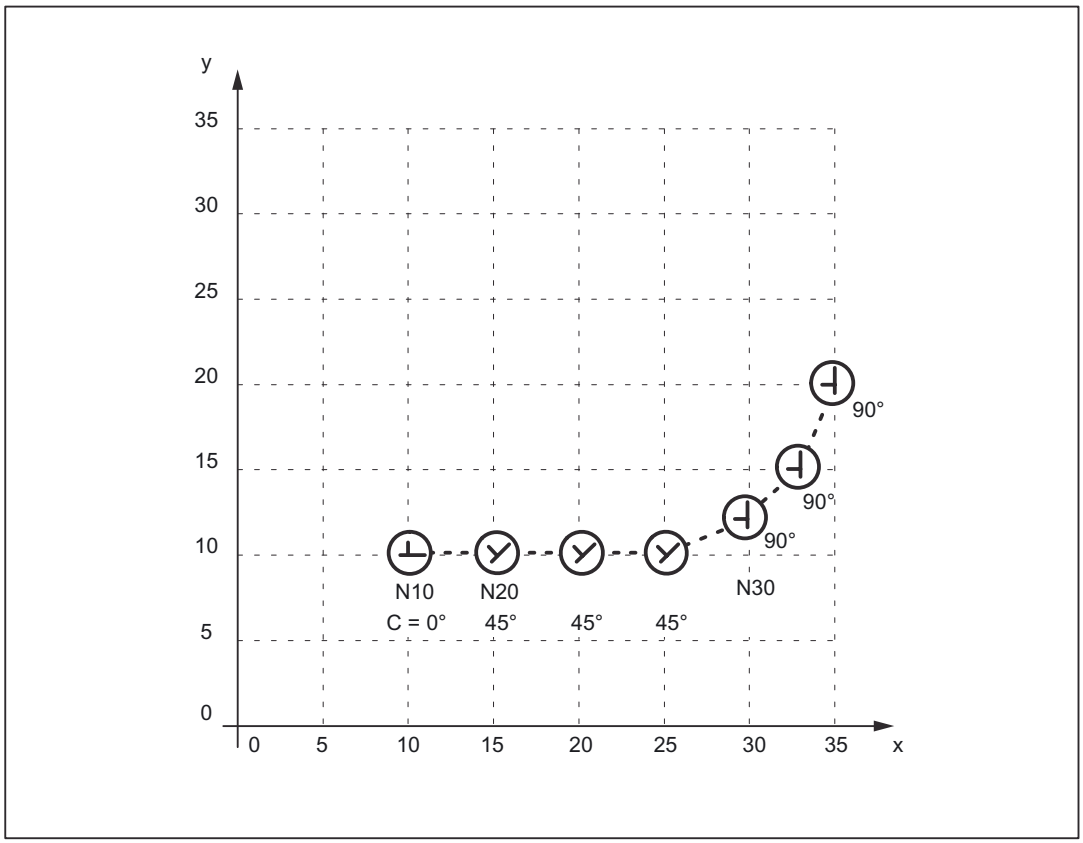
Im Satz `N20` wird in jedem der drei Zwischensätze die C-Achse um 15° gedreht. Das gleiche Verhalten ergibt sich im Satz `N30` bei Kreisinterpolation (drei Teilsätze mit jeweils 15° Achsen-Drehung).



MD26016 $MC_PUNCH_PARTITION_TYPE=1

Im Gegensatz zum ersten beschriebenen Verhalten dreht in diesem Fall die Synchronachse die gesamte programmierte Drehinformation im ersten Teilsatz der angewählten Wegaufteilung.

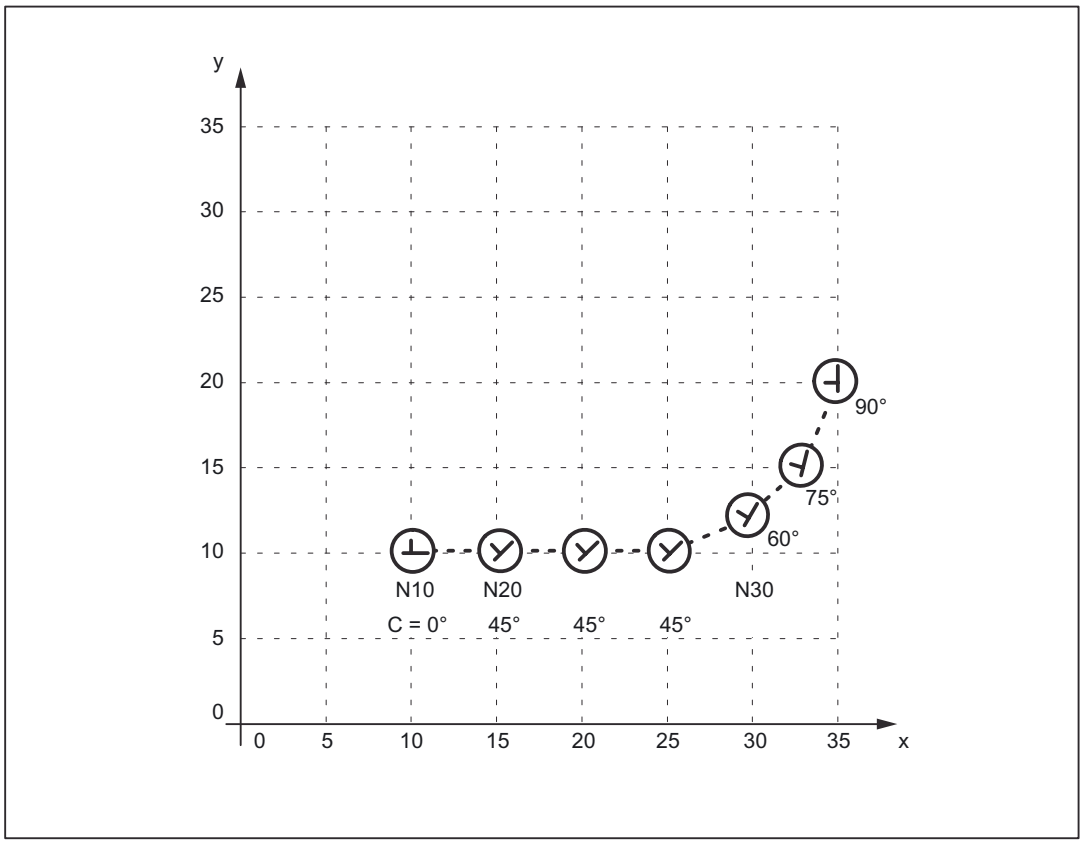
Angewendet auf das Beispiel erreicht die C-Achse bereits an der X-Position X=15 ihre im Satz programmierte Endposition `C=45`. Das gleiche Verhalten zeigt sich im nachfolgenden Kreisinterpolationssatz.



MD26016 $MC_PUNCH_PARTITION_TYPE=2

Soll nur das Verhalten bei aktiver Wegaufteilung für Linearinterpolation wie im letzten Fall, für die Kreisinterpolation aber standardmäßig sein (siehe 1. Fall), so wird MD26016=2 gesetzt.

Für das Beispiel ergibt sich dann: Im Satz `N20` wird die C-Achse im ersten Teilsatz auf C=45° gedreht. Der nachfolgende Kreisinterpolationssatz dreht in jedem Teilsatz die C-Achse um 15°.



Dieses eben beschriebene Verhalten kann speziell für die Achse des drehbaren Werkzeugs gewünscht werden, wenn diese dazu dient, das Werkzeug in eine definierte Richtung (zum Beispiel tangential) zur Kontur zu bringen und dabei **nicht** die Tangentialsteuerung verwendet werden soll. Es ersetzt aber nicht die Funktion Tangentialsteuerung, da immer Start- und Endposition der Drehachse programmiert werden muss.

Hinweis

Zusätzliche Offsetbewegung der Zusatzachsen (hier Rundachse C) erfolgt über Nullpunktverschiebung.

Randbedingungen

- Ist die C-Achse nicht als "Stanz-Nibbel-Achse" definiert, so ergeben sich im obigen Beispiel für Satz `N30` keine Wegaufteilung der C-Achsbewegung und auch keine Hubauslösung am Satzende.
- Soll die beschriebene Funktionalität für nicht-nibbeltechnologische Ausprägung aber mit Ausrichtung der Zusatzachse erfolgen, so kann die Hubauslösung durch folgendes PLC-Nahtstellensignal unterdrückt werden:

  DB 21, 22 DBX3.2 (Hubunterdrückung)

  (Anwendung: z. B. Ausrichtung Elektronenstrahl beim Schweißen)

  Ein ähnliches Verhalten ist durch folgende Maschinendatum-Einstellung möglich:

  MD26014 $MC_PUNCH_PATH_SPLITTING=1

  Damit wird die Wegaufteilung unabhängig von Stanz-/Nibbel-Funktionen vorgenommen.

10.5 Drehbares Werkzeug

10.5.1 Allgemeines

Funktionsübersicht

Für Nibbel-/Stanz-Maschinen mit drehbarem Stanzwerkzeug und dazugehörigem Unterwerkzeug (Matrize) stehen die beiden folgenden Funktionen zur Verfügung:

- Mitschleppen

  Für die gleichartige Drehung von Ober- und Unterwerkzeug

- Tangentialsteuerung

  Für die Drehachsenausrichtung der Stanzwerkzeuge normal zur Bearbeitung

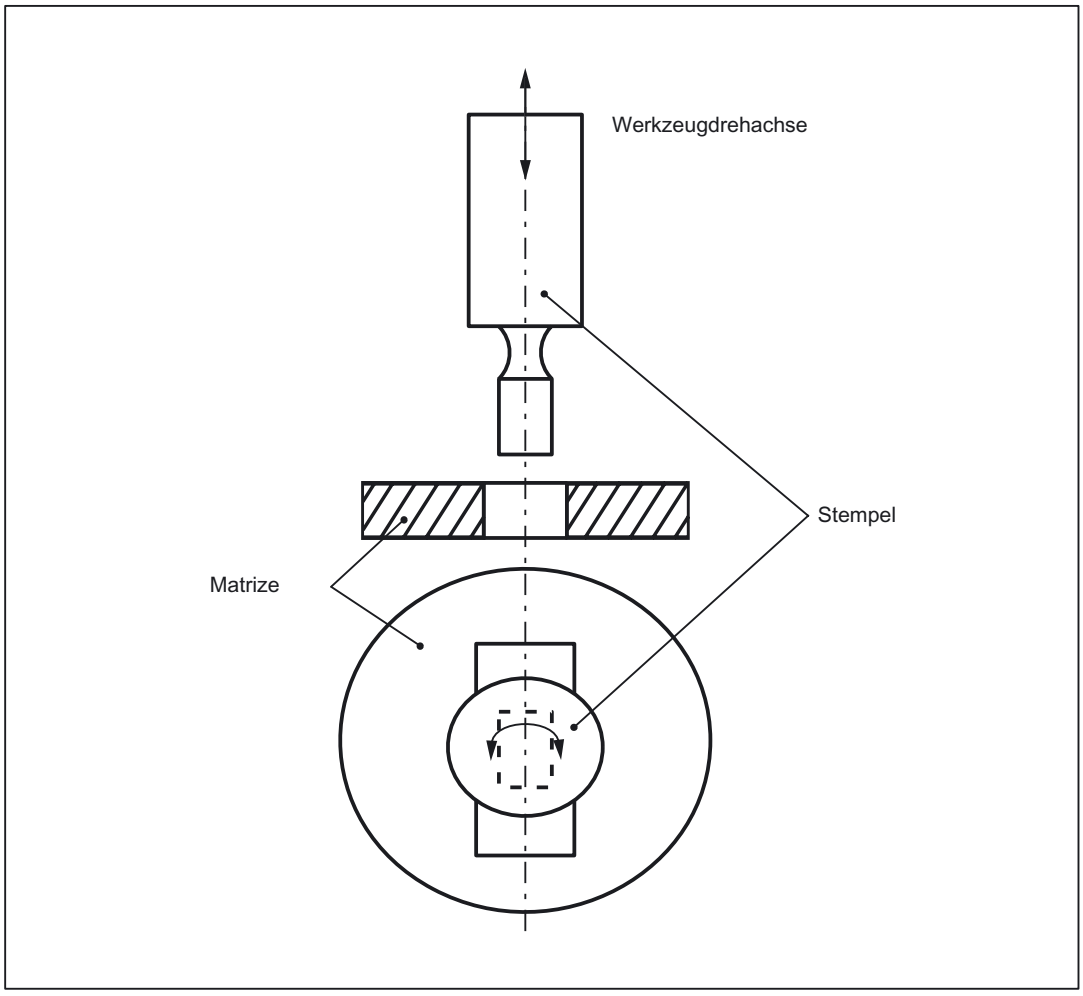


Bild 10-6 Darstellung einer drehbaren Werkzeugachse

10.5.2 Mitschleppen Ober- und Unterwerkzeug

Funktion

Durch Verwendung der Standardfunktion "Mitschleppen" kann der drehbaren Werkzeugachse des Oberwerkzeugs die Unterwerkzeugachse als Mitschleppachse zugeordnet werden.

Aktivierung

Die Funktion Mitschleppen wird mit den Sprachbefehlen `TRAILON` bzw. `TRAILOF` aktiviert bzw. deaktiviert.

Literatur:
/FB3/ Funktionshandbuch Sonderfunktionen; Achskopplungen und ESR (M3)

Beispiel

Beispiel für eine typische Nibbelmaschine bei drehbaren Stanzwerkzeugen mit C als Ober- und C1 als Unterwerkzeugachse:

```
:
:
TRAILON (C1, C, 1)            ;  Einschalten des Mitschleppverbands
G01 X100 Y100 C0 PON          ;  Hubauslösung mit C-Achs-/C1-Achs-Position C=0=C1
X150 C45                      ;  Hubauslösung mit C-Achs-/C1-Achs Position C=45=C1
:
:
M30
```

Grundstellung

Nach dem Hochlauf sind keine Mitschleppverbände aktiv. Nach Referenzpunktfahren der beiden Werkzeugachsen wird der Mitschleppverband im Normalfall nicht mehr aufgelöst.

Dies kann erreicht werden durch:

- einmalige Aktivierung des Mitschleppverbands (siehe obiges Beispiel)
- Setzen der MD-Einstellung:

  MD20110 $MC_RESET_MODE_MASK, Bit 8=1

Damit wird der Mitschleppverband über Reset/Teileprogrammstart bzw. -ende aktiv gehalten.

10.5.3 Tangentialsteuerung

Funktion

Die Ausrichtung der drehbaren Werkzeugachsen bei Stanz-/Nibbel-Maschinen tangential zur programmierten Bahn der Leitachsen wird durch die Funktion "Tangentialsteuerung" erzielt.

Aktivierung

Die Funktion "Tangentialsteuerung" wird mit den Sprachbefehlen `TANGON` bzw. `TANGOF` aktiviert bzw. deaktiviert.

Literatur:
/PGA/ Programmierhandbuch Arbeitsvorbereitung

Funktionsweise

Die Tangentialachse ist an die Interpolation der Führungsachsen gekoppelt. Eine geschwindigkeitsunabhängige Positionierung tangential zur Bahn auf die jeweilige Stanzposition ist damit nicht möglich. Bei ungünstiger Auslegung der Achsdynamik der Drehachse im Vergleich zu den Führungsachsen kann dies zu einer Verringerung der Bearbeitungsgeschwindigkeit führen. Zusätzliche Offsetwinkel sind direkt über den Sprachbefehl `TANGON` möglich.

Hinweis

Wird das Werkzeug (Stempel und Matrize) über 2 getrennte Antriebe positioniert, so kann dies über die Funktionen "Tangentialsteuerung" und "Mitschleppen" realisiert werden.

Achtung: Es muss zuerst "Tangentialsteuerung" und dann "Mitschleppen" eingeschaltet werden.

Die Tangentialsteuerung bewirkt ein automatisches Ausrichten des Stanzwerkzeuges senkrecht zum Richtungsvektor der programmierten Bahn. Vor der ersten Stanzung auf der programmierten Strecke wird das Tangentialwerkzeug positioniert. Der Tangentialwinkel ist immer auf die positive X-Achse bezogen. Ein programmierter Zusatzwinkel wird zum errechneten Winkel addiert.

Die Tangentialsteuerung wirkt bei Geraden- und Kreisinterpolation.

Beispiel: Geradeninterpolation

Die Stanz-/Nibbel-Maschine hat drehbares Unter- und Oberwerkzeug mit getrennten Antrieben.

Programmierbeispiel:

```
:
:
N2 TANG (C, X, Y, 1, "B")          ; Definition von Leit- und Folgeachsen, C ist
                                     Folgeachse zu X und Y im Basiskoordinatensystem
N5 G0 X10 Y5                       ; Start-Position
N8 TRAILON (C1, C, 1)              ; Einschalten Mitschleppen der drehbaren
                                     Werkzeugachsen C/C1
N10 Y10 C225 PON F60               ; C/C1-Achse dreht auf 225° → Hub
N15 X20 Y20 C45                    ; C/C1-Achse dreht auf 45° → Hub
N20 X50 Y20 C90 SPOF               ; C/C1-Achse dreht auf 90°, keine Hubauslösung
N25 X80 Y20 SPP=10 SON             ; Wegaufteilung; 4 Hübe werden ausgeführt mit auf
                                     90° gedrehtem WZ
N30 X60 Y40 SPOF                   ; Positionieren
N32 TANGON (C, 180)                ; Tangentialsteuerung einschalten, Offset-Winkel
                                     der drehbaren WZ-Achsen 180°
N35 X30 Y70 SPN=3 PON              ; Wegaufteilung, 3 Hübe bei aktiver
                                     Tangentialsteuerung und einem Offset-Winkel von
                                     180°
N40 G91 C45 X-10 Y-10              ; C/C1 dreht auf 225° (180° + 45° INC)
                                     Tangentialsteuerung inaktiv, da keine
                                     Wegaufteilung → Hub
N42 TANGON (C, 0)                  ; Tangentialsteuerung ohne Offset
N45 G90 Y30 SPN=3 SON              ; Wegaufteilung, 3 Hübe bei aktiver
                                     Tangentialsteuerung ohne Offset-Winkel
N50 SPOF TANGOF                    ; Ausschalten Hubauslösung + Tangentialsteuerung
N55 TRAILOF (C1, C)                ; Ausschalten Mitschleppen der drehbaren
                                     Werkzeugachsen C/C1
N60 M2
```

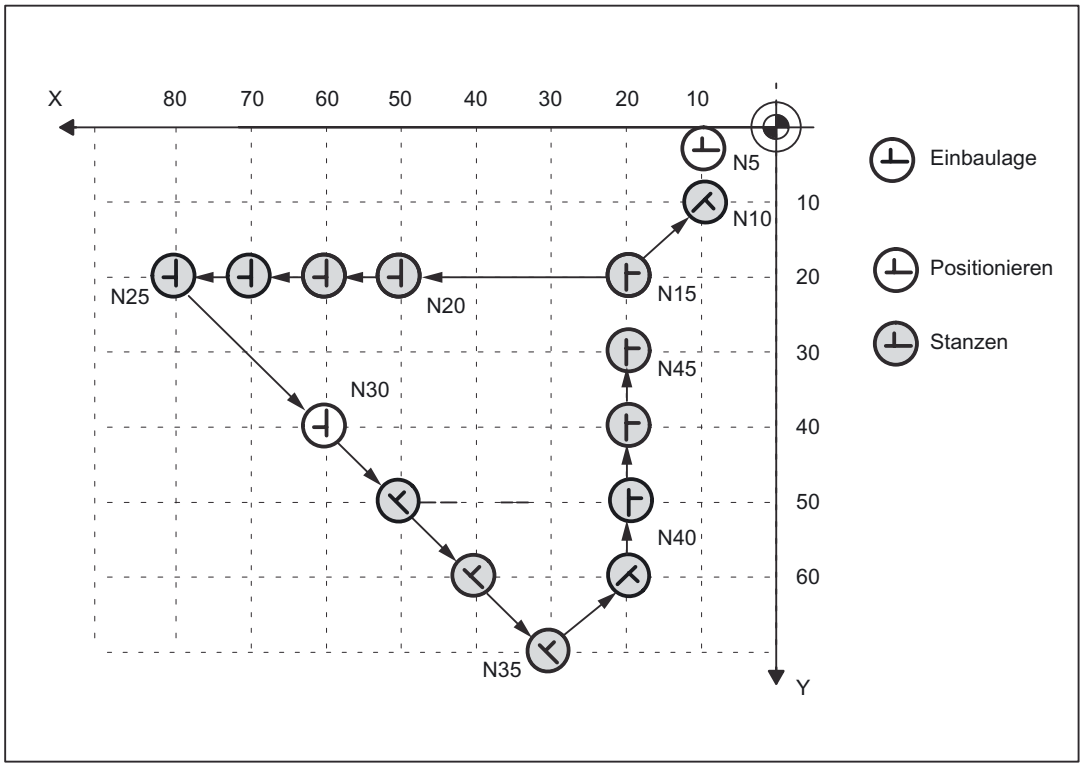


Bild 10-7 Darstellung des Programmierbeispiels in der XY-Ebene

Beispiel: Kreisinterpolation

Bei Kreisinterpolation und insbesondere bei aktiver Wegaufteilung drehen sich die WZ-Achsen in jedem Teilsatz auf eine tangentiale Ausrichtung zu den programmierten Bahnachsen.

Programmierbeispiel:

```
:
:
N2 TANG (C, X, Y, 1, "B")                    ; Definition von Leit- und
                                               Folgeachsen, C ist Folgeachse zu
                                               X und Y im Basiskoordinatensystem
N5 G0 F60 X10 Y10                            ; Startposition
N8 TRAILON (C1, C, 1)                        ; Einschalten Mitschleppen der
                                               drehbaren Werkzeugachsen C/C1 für
                                               Unter- und Oberwerkzeug
N9 TANGON (C, -90)                           ; Einschalten Tangentialsteuerung
                                               mit Offset 270°
N10 G02 X30 Y30 I20 J0 SPN=2 PON             ; Kreisinterpolation mit
                                               Wegaufteilung, 2 Hübe werden
                                               ausgeführt mit 270° Offset-Winkel
                                               und tangentialem Ausrichten auf
                                               der Kreisbahn
N15 G0 X70 Y10 SPOF                          ; Positionieren
```

```
N17 TANGON (C, 90)                          ; Einschalten Tangentialsteuerung
                                              mit Offset 90°
N20 G03 X35,86 Y24,14 CR=20 SPP=16 SON      ; Kreisinterpolation,
                                              Wegaufteilung, 4 Hübe werden
                                              ausgeführt mit 90° Offset-Winkel
                                              und tangentialem Ausrichten auf
                                              der Kreisbahn
N25 G0 X74,14 Y35,86 C0 PON                 ; Drehen der Werkzeugachsen auf 0°,
                                              Hub
N27 TANGON (C, 0)                           ; Einschalten Tangentialsteuerung
                                              mit Offset 0°
N30 G03 X40 Y50 I-14,14 J14,14 SPN=5 SON    ; Kreisinterpolation,
                                              Wegaufteilung, 5 Hübe mit Offset-
                                              Winkel 0° und tangentialem
                                              Ausrichten auf der Kreisbahn
N35 G0 X30 Y65 C90 SPOF                     ; Positionieren ohne aktive
                                              Tangentialsteuerung
N40 G91 X-10 Y-25 C180                      ; Positionierung, C-Achse dreht auf
                                              270°
N43 TANGOF                                  ; Ausschalten Tangentialsteuerung
N45 G90 G02 Y60 I0 J10 SPP=2 PON            ; Kreisinterpolation,
                                              Wegaufteilung, 2 Hübe ohne
                                              Tangentialsteuerung mit C=270°
N50 SPOF                                    ; Stanzen aus
N55 TRAILOF (C1, C)                         ; Ausschalten Mitschleppen der
                                              drehbaren Werkzeugachsen C/C1
N60 M2
```

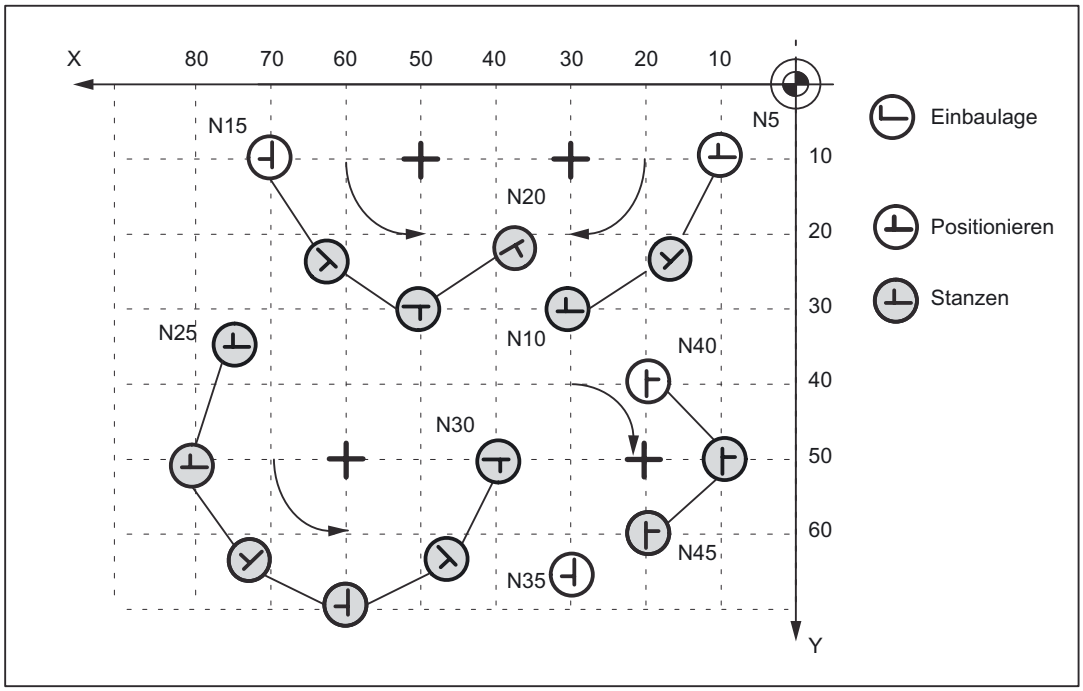


Bild 10-8 Darstellung des Programmierbeispiels in der XY-Ebene

10.6 Schutzbereiche

Pratzentotbereich

Die Funktion "Pratzentotbereich" ist als Untermenge in der Funktion "Schutzbereiche" enthalten. Überwacht wird dabei lediglich, ob Pratzen und Werkzeug gegenseitig in Konflikt kommen.

Hinweis

Bei Verletzung des Pratzenschutzbereiches kommen **keine** Umfahrungsstrategien zur Anwendung.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Achsüberwachungen, Schutzbereiche (A3)

10.7 Randbedingungen

Verfügbarkeit der Funktion "Stanzen und Nibbeln"

Die Funktion ist eine Option und verfügbar bei:

- SINUMERIK 840D mit NCU 572 und 573

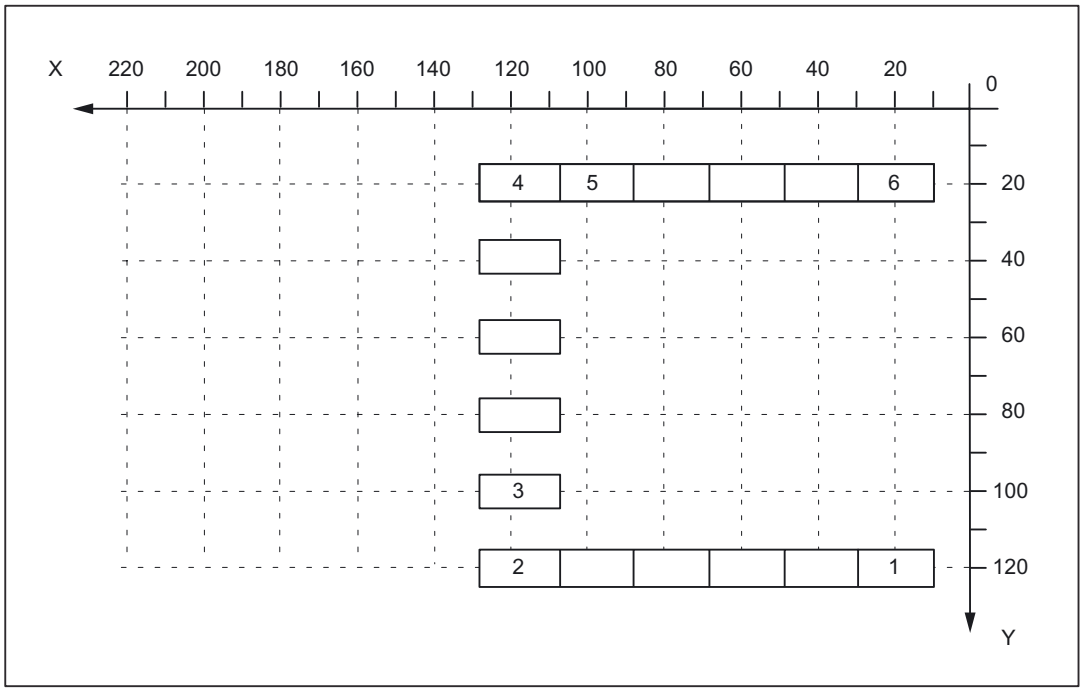
10.8 Beispiele

10.8.1 Beispiele zum definierten Nibbelbeginn

Beispiel 1

Beispiel zum definierten Nibbelbeginn.

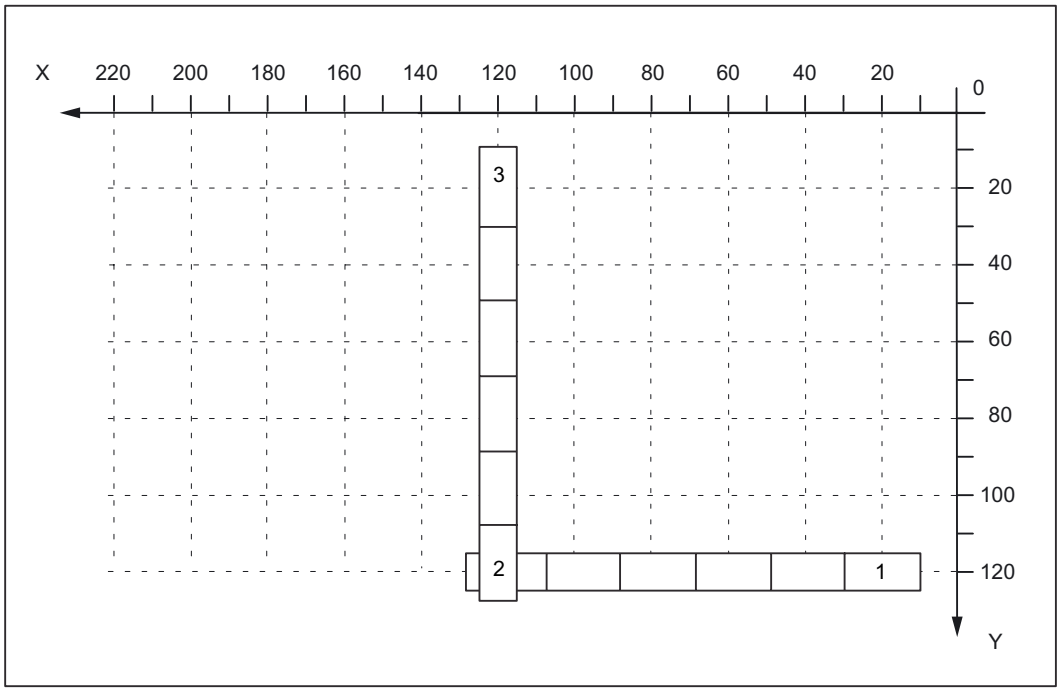
```
:
:
N10 G0 X20 Y120 SPP= 20          ;  Position 1 wird angefahren
N20 X120 SON                     ;  Definierter Nibbelbeginn, erster Hub auf "1",
                                 ;  letzter Hub auf "2"
N30 Y20                          ;  Definierter Nibbelbeginn, erster Hub auf "3",
                                 ;  letzter Hub auf "4"
N40 X20                          ;  Definierter Nibbelbeginn, erster Hub auf "5",
                                 ;  letzter Hub auf "6"
N50 SPOF
N60 M2
```



Beispiel 2

Dieses Beispiel verwendet die Funktion "Tangentialsteuerung". Als Achsname für die Tangentialachse wurde Z gewählt.

```
:
:
N5 TANG (Z, X, Y, 1, "B")        ;  Definition Tangentialachse
N8 TANGON (Z, 0)                 ;  Anwahl Tangentialsteuerung
N10 G0 X20 Y120                  ;  Position 1 wird angefahren
N20 X120 SPP=20 SON              ;  Definierter Nibbelbeginn, Tangentialsteuerung
                                    angewählt, erster Hub auf "1", letzter Hub auf
                                    "2"
N30 SPOF TANGOF                  ;  Abwahl Nibbelbetrieb und Abwahl
                                    Tangentialsteuerung
N38 TANGON (Z, 90)               ;  Anwahl Tangentialsteuerung
N40 Y20 SON                      ;  Definierter Nibbelbeginn, Tangentialsteuerung
                                    angewählt, erster Hub auf "2" um 90 Grad gedreht
                                    zu Satz N20, letzter Hub auf "3"
N50 SPOF TANGOF                  ;  Abwahl Nibbelbetrieb und Abwahl
                                    Tangentialsteuerung
N60 M2
```



Beispiele 3 und 4 für definierten Nibbelbeginn

Beispiel 3: SPP-Programmierung

```
:
:
N5 G0 X10 Y10                ;  Positionieren
N10 X90 SPP=20 SON           ;  Definierter Nibbelbeginn, 5 Stanzauslösungen
N20 X10 Y30 SPP=0            ;  Am Ende der Strecke eine Stanzauslösung
N30 X90 SPP=20               ;  4 Stanzauslösungen mit Abstand 20 mm
N40 SPOF
N50 M2
```

Beispiel 4: SPN-Programmierung

```
:
:
N5 G0 X10 Y10                ;  Positionieren
N10 X90 SPN=4 SON            ;  Definierter Nibbelbeginn, 5 Stanzauslösungen
N20 X10 Y30 PON              ;  Am Ende der Strecke eine Stanzauslösung
N30 X90 SPN=4                ;  4 Stanzauslösungen
N40 SPOF
```

```
N50 M2
```

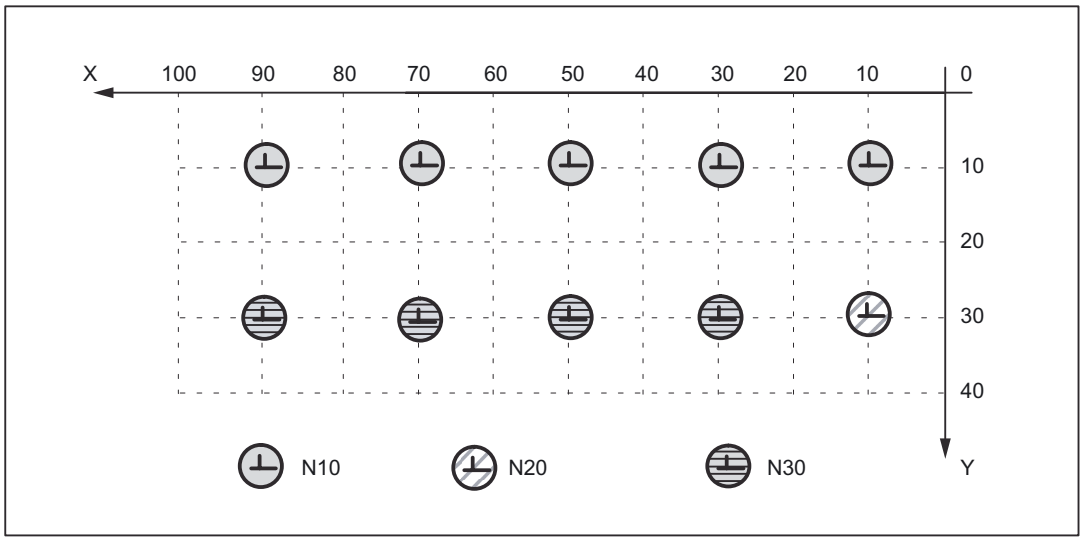


Bild 10-9 Beispiele 3 und 4 für definierten Nibbelbeginn

Beispiele 5 und 6 ohne definierten Nibbelbeginn

Beispiel 5: SPP-Programmierung

```
:
:
N5 G0 X10 Y30                    Positionieren
N10 X90 SPP=20 PON               Kein definierter Nibbelbeginn, 4 Stanzauslösungen
N15 Y10                          Am Ende der Strecke eine Stanzauslösung
N20 X10 SPP=20                   4 Stanzauslösungen mit Abstand 20 mm
N25 SPOF
N30 M2
```

Beispiel 6: SPN-Programmierung

```
:
:
N5 G0 X10 Y30                ;  Positionieren
N10 X90 SPN=4 PON            ;  Kein definierter Nibbelbeginn, 4 Stanzauslösungen
N15 Y10                      ;  Am Ende der Strecke eine Stanzauslösung
N20 X10 SPN=4                ;  4 Stanzauslösungen
N25 SPOF
N30 M2
```

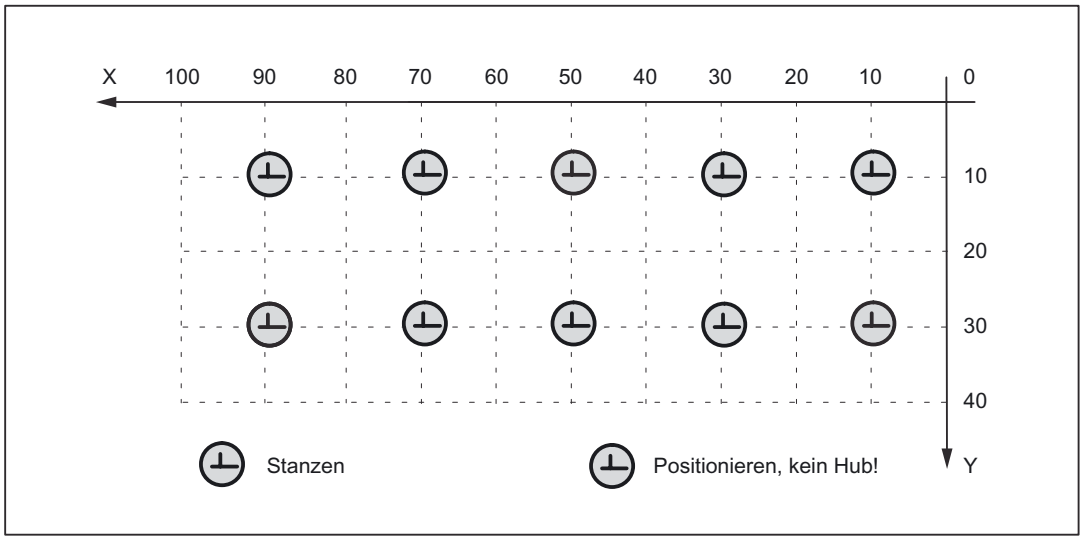


Bild 10-10 Beispiele 5 und 6 ohne definierten Nibbelbeginn

Beispiel 7: Anwendungsbeispiel für SPP-Programmierung

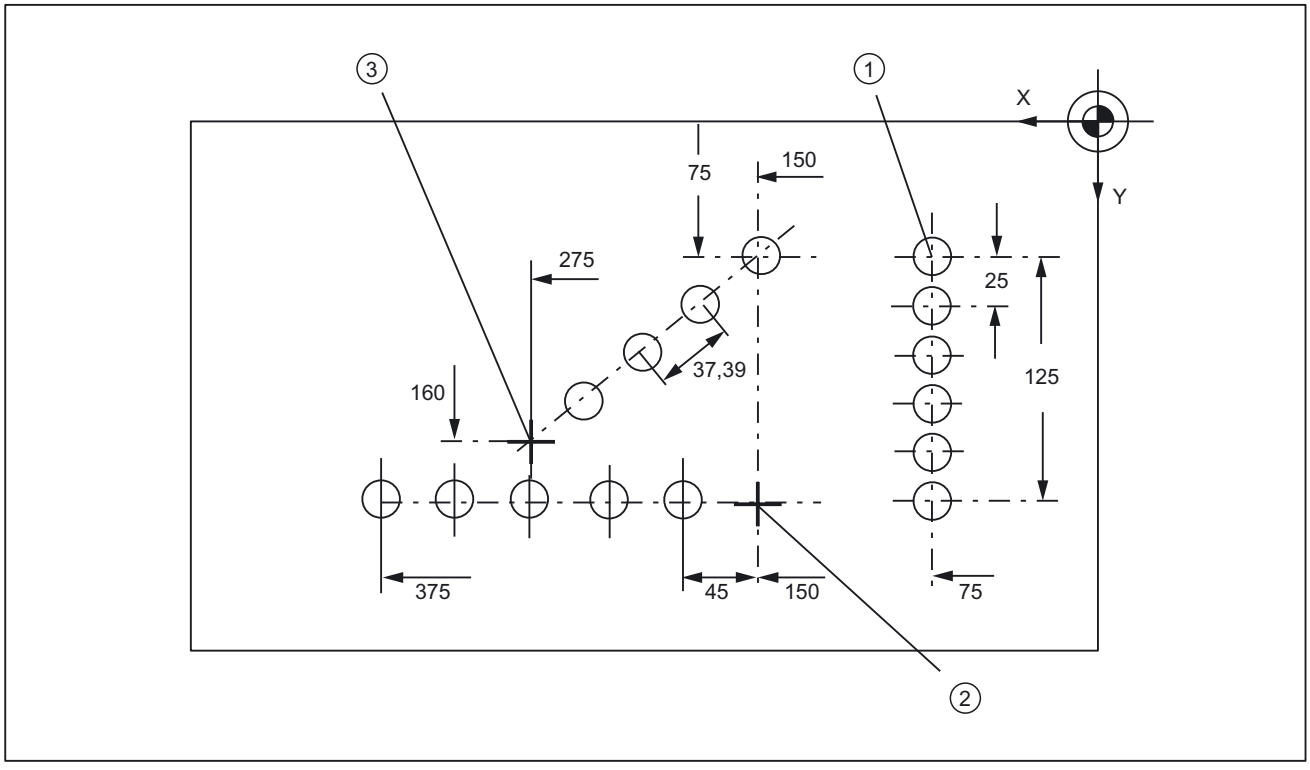


Bild 10-11 Werkstück

Programm-Ausschnitt:

```
N100 G90 X75 Y75 F60 PON          ;  Positionieren auf Startpunkt 1 der senkrechten
                                     Lochreihe, Einzelloch stanzen
N110 G91 Y125 SPP=25 PON          ;  Endpunktkoordinaten (inkrementell), Teilstrecke:
                                     25 mm, Stanzen aktivieren
N120 G90 X150 SPOF                ;  Absolute Vermaßung, Positionieren auf Startpunkt
                                     2 der waagrechten Lochreihe
N130 X375 SPP=45 PON              ;  Endpunktkoordinaten, Teilstrecke: 45 mm
N140 X275 Y160 SPOF               ;  Positionieren auf Startpunkt 3 der schrägen
                                     Lochreihe
N150 X150 Y75 SPP=40 PON          ;  Endpunktkoordinaten, programmierte Teilstrecke:
                                     40 mm, berechnete Teilstrecke: 37,79 mm
N160 G00 Y300 SPOF                ;  Positionieren
```

10.9 Datenlisten

10.9.1 Maschinendaten

10.9.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
11450	SEARCH_RUN_MODE	Suchlauf Parametrierung

10.9.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
20150	GCODE_RESET_VALUES[n]	Löschstellung der G-Gruppen
26000	PUNCHNIB_ASSIGN_FASTIN	Hardware-Zuordnung für Eingangs-Byte bei Hubsteuerung
26002	PUNCHNIB_ASSIGN_FASTOUT	Hardware-Zuordnung für Ausgangs-Byte bei Hubsteuerung
26004	NIBBLE_PUNCH_OUTMASK[n]	Maske für schnelle Ausgabe-Bits
26006	NIBBLE_PUNCH_INMASK[n]	Maske für schnelle Eingabe-Bits
26008	NIBBLE_PUNCH_CODE[n]	Festlegung der M-Funktionen
26010	PUNCHNIB_AXIS_MASK	Festlegung der Stanz- und Nibbelachsen
26012	PUNCHNIB_ACTIVATION	Aktivierung der Stanz- und Nibbelfunktionen
26014	PUNCH_PATH_SPLITTING	Aktivierung der automatischen Wegaufteilung
26016	PUNCH_PARTITION_TYPE	Verhalten von Einzelachsen bei automatischer Wegaufteilung
26018	NIBBLE_PRE_START_TIME	Automatisch aktivierbare Vorauslösezeit
26020	NIBBLE_SIGNAL_CHECK	Überwachung des Eingangssignals

10.9.2 Settingdaten

10.9.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: $SC_	Beschreibung
42400	PUNCH_DWELL_TIME	Verweilzeit
42402	NIBPUNCH_PRE_START_TIME	Vorauslösezeit
42404	MINTIME_BETWEEN_STROKES	Zeitlicher Mindestabstand zwischen zwei aufeinander folgenden Hüben

10.9.3 Signale

10.9.3.1 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Keine Hubfreigabe	DB21, ...DBX3.0	-
Manuelle Hubauslösung	DB21, ...DBX3.1	-
Hubunterdrückung	DB21, ...DBX3.2	-
Hub läuft nicht	DB21, ...DBX3.3	-
Verzögerter Hub	DB21, ...DBX3.4	-
Manuelle Hubauslösung	DB21, ...DBX3.5	-

10.9.3.2 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Hubauslösung aktiv	DB21, ...DBX38.0	DB3300.DBX6.0
Quittung manuelle Hubauslösung	DB21, ...DBX38.1	DB3300.DBX6.1

10.9.4 Sprachbefehle

G-Gruppe	Sprachbefehl	Bedeutung	
35	SPOF	Stroke / Punch Off	Stanzen und Nibbeln aus
35	SON	Stroke On	Nibbeln ein
35	SONS	Stroke On	Nibbeln ein (Lageregler)
35	PON	Punch On	Stanzen ein
35	PONS	Punch On	Stanzen ein (Lageregler)
36	PDELAYON	Punch with Delay On	Stanzen mit Verzögerung ein
36	PDELAYOF	Punch with Delay Off	Stanzen mit Verzögerung aus
Wegaufteilung			
	SPP		Weg pro Hub, modal wirksam
	SPN		Anzahl Hübe pro Satz, satzweise wirksam

P2: Positionierachsen

11

11.1 Kurzbeschreibung

Achsen für Hilfsbewegungen

Moderne Werkzeugmaschinen können außer den Achsen für die Bearbeitung eines Werkstücks auch Achsen für Hilfsbewegungen haben, wie z. B.:

- Achse für Werkzeugmagazin
- Achse für Werkzeugrevolver
- Achse für Werkstücktransport
- Achse für Palettentransport
- Achse für Lader (auch mehrachsig)
- Achse für Werkzeugwechsler
- Achse für Pinole / Lünette

Die Achsen für die Werkstückbearbeitung werden Bahnachsen genannt. Sie werden innerhalb eines Kanals vom Interpolator so geführt, dass sie gleichzeitig starten, beschleunigen, gemeinsam den Endpunkt erreichen und stoppen.

Achsen für Hilfsbewegungen werden unabhängig von den Bahnachsen mit ihrem eigenen achsspezifischen Vorschub verfahren. Viele dieser Achsen wurden früher hydraulisch bewegt und vom Teileprogramm mit einer Hilfsfunktion angestoßen. Mit der Regelung der Achse in der NC kann die Achse im Teileprogramm mit Namen angesprochen, sowie die Istposition am Bildschirm angezeigt werden.

Hinweis

Option "Positionierachse/Hilfsspindel"

Achsen für Hilfsbewegungen müssen keine interpolierenden ("vollwertigen") NC-Achsen sein. Die Hilfsbewegungen können auch mit Zusatzachsen realisiert werden, die über die Option "Positionierachse/Hilfsspindel" erhältlich sind.

Funktionseinschränkungen

Optionale Positionierachsen/Hilfsspindeln besitzen einen reduzierten Funktionsumfang. Folgende Funktionen sind **nicht** möglich:

- Verwendung als Bahnachse, Geometrieachse oder Bahnzusatzachse
- Aufnahme in Geometriachsverbund (GEOAX)
- Gewindeschneiden und Gewindebohren ohne Ausgleichsfutter

Inbetriebnahme

Achsen sind standardmäßig als interpolierende Achsen definiert:

MD30460 $MA_BASE_FUNCTION_MASK Bit 8 = 0

Soll eine Achse als Positionierachse/Hilfsspindel mit reduzierter Funktionalität betrieben werden, dann muss der Wert von Bit 8 auf "1" gesetzt werden:

MD30460 $MA_BASE_FUNCTION_MASK Bit 8 = **1**

Funktion

Mit der Funktion "Positionierachsen" können Achsen für Hilfsbewegungen einfacher in die Steuerung integriert werden:

- bei der Programmierung:

  Die Programmierung erfolgt gemeinsam mit den Achsen für die Werkstückbearbeitung im selben Teileprogramm, ohne dass dabei wertvolle Hauptzeit für die Bearbeitung verloren geht.

  Es existieren spezielle Verfahranweisungen (`POS`, `POSA`).

- beim Programmtest / Programmeinfahren:

  Der Programmtest und das Programmeinfahren erfolgt für alle Achsen gemeinsam.

- bei der Bedienung:

  Der Start der Bearbeitung sowie die Beobachtung des Bearbeitungsablaufs erfolgt für alle Achsen gemeinsam.

- bei der PLC-Projektierung / Inbetriebnahme:

  Berücksichtigung von Synchronisationsmechanismen zwischen Achsen für die Bearbeitung und Achsen für Hilfsbewegungen in der PLC oder externen Rechnern (PC) ist nicht nötig.

- bei der Systemprojektierung:

  Es ist dafür kein zweiter Kanal nötig.

Bewegungen und Interpolationen

Jeder Kanal besitzt einen Bahninterpolator und mindestens einen Achsinterpolator mit folgenden Interpolationsmöglichkeiten:

- bei einen Bahninterpolator:

  Geradeninterpolation (G1), Kreisinterpolation (G2 / G3), Spline-Interpolation, etc.

- bei einem Achsinterpolator:

  Jeder Kanal besitzt einen Achsinterpolator. Wird eine Positionierachse programmiert, startet in der Steuerung ein Achsinterpolator (mit Geradeninterpolation G1).

- Bewegungsendekriterium:

  Die programmierte Endposition einer Positionierachse ist dann erreicht, wenn das Bewegungsendekriterium FINEA, COARSA, IPOENDA erfüllt ist.

- Bahnachsen bei Eilgangbewegung:

  Bei Eilgangbewegung (G0) können Bahnachsen wahlweise mit linearer oder nicht-linearer Interpolation bewegt werden.

- Autarke Einzelachsvorgänge:

  Einzelne PLC-Achsen, Kommandoachsen gestartet über statische Synchronaktionen oder asynchrone Pendelachsen können unabhängig von der NCK interpoliert werden.

  Eine vom Hauptlauf interpolierte Achse / Spindel reagiert dann unabhängig vom NC-Programm. Das vom Programmablauf getriggerte Kanalverhalten wird entkoppelt, um dem PLC die Kontrolle einer bestimmten Achse / Spindel zu übertragen.

- Beeinflussung durch die PLC:

  Alle kanalspezifischen Signale wirken i d. R. gleichermaßen auf Bahnachsen und Positionierachsen.

  Positionierachsen können über zusätzliche achsspezifische Signale beeinflusst werden.

  PLC-Achsen werden von der PLC über spezielle Funktionsbausteine im Grundprogramm verfahren und können sich asynchron zu allen übrigen Achsen bewegen. Die Fahrbewegungen laufen losgelöst von Bahn- und Synchronaktionen ab.

11.2 Eigener Kanal, Positionierachse oder konkurrierende Positionierachse

Wenn an einer Werkzeugmaschine Achsen für Hilfsbewegungen vorhanden sind, muss entschieden werden, ob es sich aufgrund der geforderten Eigenschaften um eine Achse handelt, die:

- in einem eigenen Teileprogramm programmiert wird (→ siehe "Eigener Kanal - nur 840D sl (Seite 766) ").
- im selben Teileprogramm wie die Bearbeitung programmiert wird (→ siehe "Positionierachse (Seite 767) ").
- ausschließlich von der PLC während der Bearbeitung angestoßen wird (→ siehe "Konkurrierende Positionierachse (Seite 769) ").

11.2.1 Eigener Kanal - nur 840D sl

Ein Kanal stellt eine eigene NC dar, mit der mit Hilfe eines Teileprogramms, unabhängig von anderen Kanälen, die Bewegungen von Achsen und Spindeln sowie Maschinenfunktionen gesteuert werden können.

Unabhängigkeit zwischen Kanälen

Die Unabhängigkeit zwischen Kanälen wird wie folgt sichergestellt:

- ein aktives Teileprogramm je Kanal
- kanalspezifische Nahtstellensignale wie:
  - DB21, ... DBX7.1 (NC-Start)
  - DB21, ... DBX7.3 (NC-Stop)
  - DB21, ... DBX7.7 (Reset)
- eine Vorschubkorrektur je Kanal
- eine Eilgangkorrektur je Kanal
- kanalspezifische Auswertung und Anzeige von Alarmen
- kanalspezifische Anzeige z. B. für:
  - Achsistpositionen
  - aktive G-Funktionen
  - aktive Hilfsfunktionen
  - aktueller Programmsatz
- kanalspezifischer Test und kanalspezifische Beeinflussung von Programmen:
  - Einzelsatz
  - Probelaufvorschub (DRY RUN)
  - Satzsuchlauf
  - Programmtest

Literatur

Zur weiteren Beschreibung der Kanalfunktionalität siehe:
Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)

11.2.2 Positionierachse

Positionierachsen werden gemeinsam mit Bahnachsen, also mit den Achsen, die für die Werkstückbearbeitung verantwortlich sind, in einem Teileprogramm programmiert.

Dabei können Anweisungen für Positionierachsen und Bahnachsen gemeinsam in einem NC-Satz stehen. Trotz der gemeinsamen Programmierung in einem NC-Satz werden Bahnachsen und Positionierachsen nicht miteinander interpoliert und erreichen nicht gleichzeitig ihre Satzendpunkte (keine zeitliche Kopplung, siehe auch Kapitel "Bewegungsverhalten und Interpolationsvorgänge").

Positionierachstypen und Satzwechsel

Der Satzwechselzeitpunkt ist abhängig vom programmierten Positionierachstyp (siehe auch Kapitel "Satzwechsel"):

Typ	Beschreibung
1	Satzwechsel, wenn alle Bahnachsen und alle Positionierachsen ihre programmierten Endpunkte erreicht haben.
2	Satzwechsel, wenn alle Bahnachsen ihre programmierten Endpunkte erreicht haben. Bei Positionierachsen vom Typ 2 ist es möglich, die programmierte Endposition über mehrere Satzgrenzen hinweg anzufahren
3	Satzwechsel einstellbar bereits innerhalb der Bremsrampe der Einzelachsinterpolation, wenn das Bewegungsendekriterium und das Satzwechselkriterium für die Bahninterpolation erfüllt sind.

Bewegungssynchronisation

Mit Positionierachsen ist es möglich, Bewegungen aus demselben Bearbeitungsprogramm anzustoßen und diese Bewegungen an Satzgrenzen (Typ 1) oder an expliziten Stellen mit einem `WAITP`-Befehl (Typ 2) zu synchronisieren.

Bewegungsendekriterium für den Satzwechsel in der Bremsrampe

Für Einzelachsinterpolation kann ein weiteres Bewegungsendekriterium für den Satzwechsel bereits in der Bremsrampe eingestellt werden.

Bahnachsen bei G0 als Positionierachsen verfahren

Bei Eilgangbewegung (G0) kann jede Bahnachse als Positionierachse gefahren werden. Alle Achsen fahren somit unabhängig zu ihrem Endpunkt.

Auf diese Weise werden zwei aufeinander folgend programmierte Achsen X und Z bei G0 wie Positionierachsen behandelt. Der Satzwechsel nach Achse Z kann abhängig vom eingestellten Zeitpunkt der Bremsrampe (100-0%) von der Achse X eingeleitet werden. Während die Achse X noch fährt, startet bereits die Achse Z. Beide Achsen fahren unabhängig voneinander zu ihrem Endpunkt.

Achstypen

Positionierachsen können sowohl Linearachsen als auch Rundachsen sein.

Positionierachsen können auch Teilungsachsen sein.

Unabhängigkeit von Positionierachsen und Bahnachsen

Die Unabhängigkeit von Positionierachsen und Bahnachsen wird wie folgt sichergestellt:

- keine gemeinsame Interpolation
- jede Positionierachse hat einen eigenen Achsinterpolator
- eigene Vorschubkorrektur für jede Positionierachse
- eigener programmierbarer Vorschub
- eigenes Nahtstellensignal "Restweg löschen achsspezifisch".

Abhängigkeiten

Positionierachsen sind in folgenden Punkten abhängig:

- ein gemeinsames Teileprogramm
- Start von Positionierachsen nur an Satzgrenzen im Teileprogramm
- Bei Eilgangbewegung G0 fahren Bahnachsen als Positionierachsen wahlweise in zwei verschiedenen Modi.
- keine Eilgangkorrektur
- Die folgenden Nahtstellensignale wirken auf den gesamten Kanal und somit auch auf die Positionierachsen:
  - DB21, ... DBX7.1 (NC-Start)
  - DB21, ... DBX7.3 (NC-Stopp)
  - DB21, ... DBX7.7 (Reset)
  - DB21, ... DBX6.1 (Einlesesperre)
- Programmspezifische und kanalspezifische Alarme setzen auch die Positionierachsen still.
- Programmbeeinflussung (Probelaufvorschub, Programmtest, DRF, ... etc.) wirkt auch auf die Positionierachsen
- Satzsuchlauf und Einzelsatz wirkt auch auf die Positionierachsen.

- Der letzte im Suchlauf behandelte Satz mit einem programmierten Bewegungsendekriterium dient als Behälter zur Einstellung aller Achsen.
- Die Gruppe1 (modal wirkende Bewegungsbefehle) der G-Funktionen (das sind `G0`, `G1`, `G2`, ...) gilt nicht für Positionierachsen.

Literatur:
Programmierhandbuch Grundlagen

Anwendungsfälle

Typische Anwendungen für Positionierachsen sind:

- einachsige Lader
- mehrachsige Lader ohne Interpolation (PTP → Punkt-zu-Punkt-Fahren)
- Werkstückzuführung und Werkstücktransport

Weitere Anwendungen sind möglich:

- Werkstückzuführung und Werkstücktransport können bei `G0` unabhängig voneinander zu ihrem Endpunkt fahren.
- Bei Maschinen mit mehreren aufeinander folgenden Bearbeitungsgängen: Einzelne Bearbeitungsschritte können aufgrund des Satzwechsels bereits innerhalb der Bremsrampe der Einzelachsinterpolation erheblich reduziert werden.

Hinweis

Nicht geeignet sind Positionierachsen für mehrachsige Lader, die eine Interpolation zwischen den Achsen erfordern (Bahninterpolator).

11.2.3 Konkurrierende Positionierachse

Konkurrierende Positionierachsen sind Positionierachsen mit folgenden Eigenschaften:

- Der Anstoß von der PLC muss nicht an Satzgrenzen erfolgen, sondern ist zu jedem beliebigen Zeitpunkt in jeder Betriebsart möglich (auch während im Kanal schon ein Teileprogramm läuft).
- Um eine konkurrierende Positionierachse sofort nach Power On vom Teilprogramm aus zu bewegen, ist der Programmbefehl `WAITP` erforderlich.
- Das Teileprogramm läuft unbeeinflusst weiter, auch wenn die konkurrierende Positionierachse ihre von der PLC vorgegebene Position noch nicht erreicht hat.
- Abhängig von der Einstellung im Maschinendatum MD30552 $MA_AUTO_GET_TYPE ist ein automatischer Achstausch möglich.
- Mit den Programmierbefehlen:
  - `GET(<Achse>)` oder `WAITP(<Achse>)` wird eine konkurrierende Positionierachse wieder zur Kanalachse.
  - `RELEASE(<Achse>)` oder `WAITP(<Achse>)` ist eine Kanalachse als konkurrierende Positionierachse über die PLC beeinflussbar.

Anstoß von der PLC

Von der PLC wird die konkurrierende Positionierachse bei SINUMERIK 840D sl über den FC18 angestoßen.

- Vorschub

  Bei Vorschub=0 wird der Vorschub aus folgendem Maschinendatum genommen:

  MD32060 $MA_POS_AX_VELO (Löschstellung für Positionierachsgeschwindigkeit)

- Absolutmaß (`G90`), Kettenmaß (`G91`)

  Absolutmaß auf kürzestem Weg für Rundachsen (`<Rundachsname>=DC(<Wert>)`)

Folgende Funktionen sind fest vorgegeben:

- Linearinterpolation (`G1`)
- Vorschub in mm/min bzw. grad/min (`G94`)
- Genauhalt (`G9`)
- aktuell angewählte einstellbare Nullpunktverschiebungen sind gültig

Anwendungsfälle

Typische Anwendungen für konkurrierende Positionierachsen sind:

- Werkzeugmagazin mit manuellem Be- und Entladen während der Bearbeitung
- Werkzeugmagazine mit Werkzeugbereitstellung während der Bearbeitung

11.3 Bewegungsverhalten und Interpolationsvorgänge

11.3.1 Bahninterpolator und Achsinterpolator

Bahninterpolator

Jeder Kanal besitzt einen Bahninterpolator für die unterschiedlichsten Interpolationsarten wie Geradeninterpolation (G1), Kreisinterpolation (G2/G3), Spline-Interpolation, etc.

Achsinterpolator

Zusätzlich zum Bahninterpolator besitzt jeder Kanal Achsinterpolatoren. Die maximale Anzahl entspricht der maximalen Anzahl vorhandener Kanalachsen.

Wird eine Positionierachse programmiert, startet in der Steuerung ein Achsinterpolator mit Geradeninterpolation G1. Dieser Achsinterpolator läuft unabhängig von Bahninterpolator so lange, bis die programmierte Endposition der Positionierachse erreicht ist.

Eine zeitliche Kopplung zwischen Bahninterpolator und Achsinterpolator sowie zwischen den Achsinterpolatoren existiert nicht.

Bahnsteuerbetrieb (G64) ist bei Positionierachsen nicht möglich.

Die programmierte Endposition einer Positionierachse ist dann erreicht, wenn das Bewegungsendekriterium FINEA, COARSA oder IPOENDA erfüllt ist.

11.3.2 Interpolationsverhalten der Bahnachsen bei G0

Bei Eilgangbewegung (G0) können Bahnachsen wahlweise mit linearer oder nicht-linearer Interpolation bewegt werden.

Lineare Interpolation

Eigenschaften:

- Die Bahnachsen werden gemeinsam interpoliert.
- Die mit G0 programmierte Werkzeugbewegung wird mit der größtmöglichen Verfahrgeschwindigkeit (Eilgang) ausgeführt.
- Die Eilganggeschwindigkeit ist im folgenden Maschinendatum für jede Achse getrennt festgelegt:

  MD32000 $MA_MAX_AX_VELO
- 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-Code-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 angewählten 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 im folgenden Maschinendatum angegebenen Eilganggeschwindigkeit:

  MD32000 $MA_MAX_AX_VELO
- Den kanalspezifischen Restweg löschen über PLC und über Synchronaktion wirkt auf alle Positionierachsen, die als Bahnachsen programmiert wurden.

Bei nicht-linearer Interpolation gilt bezüglich des axialen Rucks:

- die Einstellung für die jeweilige Positionierachse BRISKA, SOFTA, DRIVEA

oder

- die Einstellung in den Maschinendaten:

  MD32420 $MA_JOG_AND_POS_JERK_ENABLE

  und

  MD32430 $MA_JOG_AND_POS_MAX_JERK

Die bereits existierenden Systemvariablen, die sich auf den Bahnrestweg beziehen ($AC_PATH, $AC_PLTBB und $AC_PLTEB) werden unterstützt.

⚠ VORSICHT

Da bei der nicht-linearen Interpolation eine andere Kontur gefahren werden kann, werden Synchronaktionen, die sich auf Koordinaten der ursprünglichen Bahn beziehen, ggf. nicht aktiv.

Auswahl der Interpolationsart

Die Interpolationsart, die bei G0 wirksam sein soll, wird über das folgende Maschinendatum voreingestellt:

MD20730 $MC_G0_LINEAR_MODE (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.

Unabhängig von der Voreinstellung kann das gewünschte Interpolationsverhalten bei G0 auch über die beiden folgenden Teileprogrammbefehle programmiert werden:

RTLIOF Ausschalten der linearen Interpolation.
⇒ Bei Eilgang (G0) ist die **nicht-lineare** Interpolation aktiv.

RTLION Einschalten der linearen Interpolation.
⇒ Bei Eilgang (G0) ist die **lineare** Interpolation aktiv.

Das aktuell eingestellte Interpolationsverhalten der Bahnachsen bei G0 kann mit der Systemvariablen $AA_G0MODE abgefragt werden.

Hinweis

Bei beiden Interpolationsarten wirkt als Override die Eilgangkorrektur kanalspezifisch.

11.3.3 Autarke Einzelachsvorgänge

Funktionalität

Einzelne PLC-Achsen, Kommandoachsen gestartet über statische Synchronaktionen oder asynchrone Pendelachsen können unabhängig von der NCK interpoliert werden. Eine vom Hauptlauf interpolierte Achse/Spindel reagiert dann unabhängig vom NC-Programm bezüglich:

- NC-STOP
- Alarmbehandlung
- Programmbeeinflussungen
- Programmende
- RESET

Randbedingungen

Aktuell vom NC-Programm beaufschlagte Achsen/Spindeln werden von der PLC nicht kontrolliert.

Für PLC-kontrollierte Achsen/Spindeln können über satzbezogene und modale Synchronaktion **keine** Kommandoachsbewegungen gestartet werden. Es wird der Alarm 20143 gemeldet.

Ablaufkoordinator

Der Ablauf der autarken Einzelachsfunktionen mit den jeweiligen Übergaben wird in einer so genannten **"Use-Case"** Übersicht dargestellt:

NCK kontrolliert: PLC will die Kontrolle der Achse/Spindel übernehmen

Use-Case 1: Ablauf von Achse/Spindel abbrechen

Use-Case 2: Achse/Spindel stoppen

Use-Case 3: Achse/Spindel weiterfahren

Use-Case 4: Achse/Spindel zurücksetzen

PLC kontrolliert: PLC will die Kontrolle der Achse/Spindel abgeben

Achskontrolle durch PLC

Kontrolle der Achse/Spindel übernehmen

Das vom NC-Programmablauf getriggerte Kanalverhalten wird entkoppelt. Die Übernahme der Kontrolle über die Achse/Spindel erfolgt über die axiale VDI-Nahtstelle mit dem NC/PLC-Nahtstellensignal:

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

Ablaufbeschreibung:

1. PLC an NCK mit NST:

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

   → Kontrolle über eine Achse übernehmen.
2. NCK prüft: Ist die Achse eine Hauptlaufachse oder neutrale Achse
3. NCK prüft, ob eine weitere Achse von der PLC kontrolliert werden kann.
4. NCK bestätigt die Übernahme und überträgt den Zustand der Achse mit NST:

   DB31, ... DBX63.1 (PLC kontrolliert Achse) == 1 zum PLC

   Systemvariable $AA_SNGLAX_STAT als abfragbarer aktueller Achsstatus.
5. PLC hat die Kontrolle der Achse/Spindel übernommen.

Alternativen

Der Kanalzustand steht auf "unterbrochen", da ein Kanalstopp ansteht. Die Achse wird analog der Ablaufbeschreibung behandelt. Abhängig vom Zustand der zu kontrollierenden Achse bestehen die zwei folgenden Alternativen:

- Die von der PLC zu kontrollierende Achse ist **nicht** aktiv.

  Zusätzlich wird der Stoppzustand aufgehoben. Ein anschließender Start der Achse führt direkt zu einer Achsbewegung.

- Die von der PLC zu kontrollierende Achse ist aktiv.

  Der Stoppzustand wird nicht aufgehoben. Den Achszustand nach Use-Case 2 "Achse stoppen" erzeugen. Fortsetzen der Achsbewegung nach Use-Case 3 "Achse weiterfahren".

- Der Kanal führt einen NC-`RESET` aus.

  Dieser Vorgang ist asynchron zur Übernahme der Kontrolle durch die PLC. Die beiden vorher genannten Alternativen können auftreten oder die Achse ist noch dem Kanal zugeordnet und wird zurückgesetzt.

Randbedingungen

Die NCK muss hochgefahren sein.

Es können keine Achsen/Spindeln, die aktuell vom NC-Programm beeinflusst werden, von der PLC kontrolliert werden, d. h. die Achse muss über statische Synchronaktionen oder Pendelachse als neutrale Achse, konkurrierende Positionierachse oder Kommandoachse gestartet sein.

Die NCK bestätigt die Übernahme einer Programm-Achse nicht und gibt den Alarm 26072 "Kanal %1 axis %2 kann nicht vom PLC kontrolliert werden" aus.

Wird der Wert aus dem in einem Optionsdatum eingestellten Wert überschritten, so ist eine Übernahme an die PLC nicht möglich.

Es wird der Alarm 26070 "Kanal %1 axis %2 kann nicht vom PLC kontrolliert werden, max. Anzahl überschritten" ausgegeben.

Achskontrolle vom PLC abgeben

Kontrolle der Achse/Spindel abgeben

Die Übernahme der Kontrolle über die Achse/Spindel erfolgt über die axiale VDI-Nahtstelle mit dem NC/PLC-Nahtstellensignal:

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

Ablaufbeschreibung:

1. PLC an NCK mit NST:

   DB31, ... DBX28.7 (PLC kontrolliert Achse) == 0

   → Kontrolle über eine Achse abgeben.

2. NCK prüft, ob ein axialer Alarm ansteht.

3. NCK prüft, ob eine Bewegung aktiviert wurde, die noch nicht beendet ist und hält diese Bewegung mit einem axialen Stopp gemäß Use-Case 2 "Achse/Spindel Stoppen" an.

4. NCK führt einen axialen RESET entsprechend Use-Case 4 "Achse/Spindel zurücksetzen" aus, indem für eine einzelne Achse die erforderlichen Maschinendaten für RESET eingelesen und aktiviert werden.
5. NCK bestätigt die Übernahme und überträgt den Zustand der Achse zum PLC über die axiale VDI-Nahtstelle mit den NC/PLC-Nahtstellensignalen:

   DB31, ... DBX63.1 (PLC kontrolliert Achse) == 0

   DB31, ... DBX63.2 (Achsstop aktiv) == 0

   DB31, ... DBX63.0 (AXRESET DONE) == 0

   Achsstatus mit Systemvariable $AA_SNGLAX_STAT == 0 aktiv.
6. Der Kanal hat die Kontrolle über die Achse/Spindel.

Alternativen

Die NCK bestätigt die Übergabe und setzt intern für die Achse/Spindel den Kanalzustand "gestoppt", wenn von der NCK festgestellt wird, dass:

- der Kanal zur Achskontrolle aufgrund eines Stopps im Zustand "unterbrochen" ist.
- ein Stoppalarm für den Kanal ansteht.
- ein Stoppalarm für die BAG ansteht.

Randbedingungen

Die Achse/Spindel muss von der PLC kontrolliert sein.

Die NCK bestätigt die Übernahme einer Achse/Spindel nur, wenn kein axialer Alarm ansteht. Anderenfalls wird der Alarm 26074 "Kanal %1 ausschalten der PLC-Kontrolle von Achse %2 im aktuellen Zustand nicht erlaubt" wird gemeldet.

Anwendungsfälle

Eine vom Hauptlauf interpolierte Achse/Spindel kann vom PLC über folgende VDI-Nahtstellensignale unabhängig vom NC-Programm beeinflusst werden:

- Ablauf abbrechen:

  DB21, ... DBX6.2 (Restweg löschen)
- Achse/Spindel stoppen:

  DB31, ... DBX28.6 (AxStop, Halt)

- Achse/Spindel weiterfahren:

  DB31, ... DBX28.2 (AXRESUME)

- Achse/Spindel zurücksetzen:

  DB31, ... DBX28.1 (AXRESET)

Hinweis

Die Achse/Spindel muss von der PLC kontrolliert sein. Diese Randbedingung gilt grundsätzlich für alle Anwendungsfälle: Use-Case 1 bis 4.

Der Signalaustausch an der VDI-Nahtstelle bei autarken Einzelvorgängen ist anhand einer Maschinenachse 1 in einer Gegenüberstellung von Aktionen des PLC als Reaktion des NCK im Kapitel "Beeinflussung durch die PLC" beschrieben.

Weitere Informationen zum kanalspezifischen VDI-Signal entnehmen Sie bitte:

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen, Diverse Nahtstellensignale (A2)

Alarmbehandlung bei Einzelachsen

Alarme mit einem Achsparameter werden nur angezeigt und nicht als Einzelachsalarme behandelt. Es ist ausreichend, diese so genannten "Showalarme" mit `CANCEL` zu quittieren.

Hinweis

Nur Alarme, die mit AXRESET wieder gelöscht werden müssen, wirken sich auf Einzelachsen aus.

Use-Case 1

Ablauf von Achse/Spindel abbrechen

Die Funktion Achse/Spindel abbrechen entspricht den bisherigen Verhalten des kanalspezifischen NC/PLC-Nahtstellensignals:

DB21, ... DBX6.2 (Restweg löschen)

Use-Case 2

Achse/Spindel stoppen

Alle vom Hauptlauf gesteuerten Achsbewegungen werden gestoppt. Damit werden folgende Bewegungen unterbrochen

- Achse/Spindel:

  PLC-Achse, asynchrone Pendelachse oder per statische Synchronaktion gestartete Kommandoachse. Folgeachsbewegung der Achse/Spindel wird nicht gestoppt.

- Überlagerte Bewegung wie:

  $AA_OFF, DRF-Handradverfahren oder Online-Werkzeugkorrektur und externe Nullpunktverschiebung.

Ablaufbeschreibung:

- PLC fordert vom NCK mit NST:
  DB31, ... DBX28.6 ("AxStop, Halt") == 1
  die entsprechende Achse anzuhalten.
- NCK bremst die Achse über eine Rampe ab.
- NCK führt die Achse in den Zustand gestoppt über und teilt dem PLC den Zustandswechsel über die VDI-Nahtstelle (NCK→PLC) wie folgt mit:

  DB31, ... DBX63.2 (Achsstop aktiv) == 0,

  DB31, ... DBX64.7 (Fahrbefehl plus) == 0 bzw.

  DB31, ... DBX64.6 (Fahrbefehl minus) == 0 und

  DB31, ... DBX60.7 (Genauhalt fein) == 1 bzw.

  DB31, ... DBX60.6 (Genauhalt grob) == 1

  Achsstatus mit Systemvariable $AA_SNGLAX_STAT == 3 unterbrochen.
- NCK beendet diesen Vorgang.

Alternativen

Folgende Fälle können auftreten, um die Achse/Spindel zu stoppen:

- Gleichzeitig die Achse/Spindel übernehmen und Stoppen.

  Wenn PLC der NCK mitteilt: PLC will die Achse/Spindel übernehmen und gleichzeitig festgestellt wird, dass die Achse/Spindel anzuhalten ist, dann: NCK markiert die Achse/Spindel als übernommen und fährt an der Stelle fort, wo unterbrochen wurde.
- Die Achse/Spindel wird durch mehrere Abläufe beaufschlagt:

  Alle Abläufe außer den aus Folgeachs-Bewegungen resultierenden Bewegungen stoppen. Eine Folgeachs-Bewegung kann nur durch einen Stopp der Leitachse gestoppt werden!
- PLC fordert NCK auf, die Achse/Spindel anhalten und gleichzeitig tritt ein axialer Stoppalarm für diese Achse auf:

  NCK bremst die Achse über eine Rampe ab und bestätigt den Bremsvorgang über BTSS.

  Gleichzeitig wird der Alarm der PLC mit:
  DB31, ... DBX61.1 (Axialer Alarm) == 1 der Alarm gemeldet und
  der Zustand Systemvariable $AA_SNGLAX_STAT == 5 gesetzt.

  NCK führt die Achse in den Zustand gestoppt über und teilt dem PLC den Zustandswechsel mit.
- PLC fordert NCK auf, die Achse/Spindel anzuhalten und ein gleichzeitiger Stoppalarm tritt auf, dann wird das NC-Programm aktiv. Der Stoppalarm wird ignoriert und hat keinen Einfluss.
- Eine Rückzugsbewegung des "Erweiterten Stillsetzen und Rückziehens" kann nicht gestoppt werden. Diese Rückzugsbewegung kann mit einem NST:
  DB31, ... DBX28.6 (AxStop, Halt)
  der PLC-Schnittstelle nicht angehalten werden.

Randbedingungen

Die PLC muss vorher die Achse/Spindel tatsächlich übernommen haben. Anderenfalls wird das folgende NC/PLC-Nahtstellensignal ignoriert:

NST DB31, ... DBX28.6 (AxStop, Halt)

Use-Case 3

Achse/Spindel weiterfahren

Die unterbrochenen, vom Hauptlauf gesteuerten Achsbewegungen einer Achse/Spindel gemäß Use Case 2 "Achse Stoppen" wird fortgesetzt.

Ablaufbeschreibung:

- PLC fordert NCK mit NST:
  DB31, ... DBX28.2 (AXRESUME) == 1
  auf, die entsprechende Achse fortzusetzen.
- NCK prüft, ob ein axialer Alarm die Achse/Spindel mit den Löschkriterium CANCELCLEAR oder NCSTARTCLEAR ansteht und löscht diesen.
- NCK prüft, ob die Achse fortgesetzt werden kann - keine Verriegelung aufgrund eines Alarms und führt Achse in den Zustand aktiv über.
- Die Achsbewegung wird fortgesetzt und der PLC wird der Zustandswechsel über die VDI-Nahtstelle (NCK→PLC) wie folgt mitgeteilt:

  DB31, ... DBX63.2 (Achsstop aktiv) == 0,

  DB31, ... DBX64.7 (Fahrbefehl plus) == 1 bzw.

  DB31, ... DBX64.6 (Fahrbefehl minus) == 1 und

  DB31, ... DBX60.7 (Genauhalt fein) == 0 bzw.

  DB31, ... DBX60.6 (Genauhalt grob) == 0

  Achsstatus mit Systemvariable $AA_SNGLAX_STAT == 4 aktiv.
- NCK beendet diesen Vorgang.

Alternativen

Die Achse/Spindel durch mehrere Abläufe beaufschlagen:

- Alle Abläufe fortsetzen.
- Folgebewegungen sind von der Bewegung der Leitachse abhängig.

Randbedingungen

Folgende Fehlerfälle werden vom NCK festgestellt:

- Wenn PLC auffordert Achsen/Spindeln weiterzufahren, und diese Achse/Spindel nicht vom PLC übernommen wurde, wird das NST DB31, ... DBX28.2 (AXRESUME) ignoriert.
- Die Achse/Spindel sich nicht im gestoppten Zustand befindet, wird das NST DB31, ... DBX28.2 (AXRESUME) ignoriert.
- Die Achse/Spindel aufgrund eines anstehenden Alarms nicht fortgesetzt werden darf, wird das NST DB31, ... DBX28.2 (AXRESUME) ignoriert.

Use-Case 4

Achse/Spindel zurücksetzen

Eine Achse/Spindel wird in den Grundzustand zurückgesetzt.

Ablaufbeschreibung:

- PLC fordert NCK mit NST:
  DB31, ... DBX28.1 (AXRESET) == 1
  auf, die entsprechende Achse zurückzusetzen.
- NCK stellt fest, dass die Achse/Spindel aktiv ist und überführt diese in den Zustand gestoppt.
- Die gestoppten Abläufe werden abgebrochen und dem PLC wie beim "Restweg löschen" mitgeteilt.
- Die NCK-internen Zustände für die Achse/Spindel werden zurückgesetzt.
- Die axialen, bei RESET wirksamen Maschinendaten werden wirksam.

Hinweis

Bei einem Kanal-Reset werden keine axialen Maschinendaten für eine von der PLC kontrollierten Achse wirksam.

- Ein axialer RESET durchgeführt wurde, wird an der Nahtstelle gemeldet:

  DB31, ... DBX63.0 ("AXRESET DONE") == 1

  DB31, ... DBX63.2 ("Achsstop aktiv") == 0

  und

  Systemvariable $AA_SNGLAX_STAT == 1 Einzelachse in RESET aktiv.
- NCK beendet diesen Vorgang.

Alternativen

Die NCK stellt fest, dass die entsprechende Achse/Spindel:

- im Zustand "gestoppt" ist:

  → Ablaufbeschreibung ab Punkt 3. fortsetzen.
- im Zustand "StopByAlarm" ist:

  → Ablaufbeschreibung ab Punkt 3. fortsetzen.
- nicht aktiv ist:

  → Ablaufbeschreibung ab Punkt 4. fortsetzen.

Randbedingungen

Die PLC muss vorher die Achse/Spindel tatsächlich übernommen haben. Anderenfalls wird das NST DB31, ... DBX28.1 (AXRESET) ignoriert.

11.3.4 Autarke Einzelachsvorgänge mit NC-geführtem ESR

Erweitertes Stillsetzen NC-geführt

Das NC-geführte Erweiterte Stillsetzen und Rückziehen ist auch für Einzelachsen möglich und ist projektierbar mit den axialen Maschinendaten:

Verzögerungszeit für ESR-Einzelachse mit

MD37510 $MA_AX_ESR_DELAY_TIME1

ESR-Zeit für interpolatorische Bremsen der Einzelachse mit

MD37511 $MA_AX_ESR_DELAY_TIME2

Die Werte dieser axialen Maschinendaten wirken jedoch nur dann, wenn die Achse/Spindel eine Einzelachse ist.

Ausgelöst wird das NC-geführtes Erweiterte Stillsetzen und Rückziehen mit dem axialen Trigger $AA_ESR_TRIGGER[Achse]. Er funktioniert analog zu $AC_ESR_TRIGGER und wirkt ausschließlich auf Einzelachsen.

Literatur:

/FB3/ Funktionshandbuch Sonderfunktionen; Achskopplungen und ESR (M3)

Erweitertes Rückziehen NC-geführt

Beim Einzelachsenrückziehen muss der Wert per POLFA(Achse, Typ, Wert) programmiert worden sein, wobei folgende Bedingungen erfüllt sein müssen:

- Die Achse muss zum Triggerzeitpunkt Einzelachse sein
- $AA_ESR_ENABLE[Achse]=1
- POLFA(Achse, Typ, Wert) nur bei Typ=1 oder Typ=2

  POLFA(Achse, , Wert, Achse, Typ, ,Achse, Typ).

Hinweis

NC-geführtes Erweitertes Stillsetzen für Einzelachsen:

Der Trigger wirkt nur, wenn die Achse zum Triggerzeitpunkt eine Einzelachse ist, andernfalls wird der Trigger ignoriert und damit das axiale Stillsetzen für diese Achse **nicht** ausgeführt.

NC-geführtes Erweitertes Rückziehen für Einzelachsen:

Das kanalspezifische NC-geführte Erweiterte Rückziehen wirkt **nicht** auf Einzelachsen. Alle Achsen, die zum Triggerzeitpunkt $AC_ESR_TRIGGER Einzelachsen sind, werden beim kanalspezifischen Rückziehen ignoriert.

Das gilt auch wenn alle Parameter für das Rückziehen gesetzt sind, wie:

MD37500 $MA_ESR_REACTION

$AA_ESR_ENABLE der Achse etc.

Beispiele

Erweitertes **Stillsetzen** einer Einzelachse:

MD37500 $MA_ESR_REACTION[AX1]=22

MD37510 $MA_AX_ESR_DELAY_TIME1[AX1]=0.3

MD37511 $MA_AX_ESR_DELAY_TIME2[AX1]=0.06

...

$AA_ESR_ENABLE[AX1] = 1

$AA_ESR_TRIGGER[AX1]=1 ; ab hier läuft das Stillsetzen an

Erweitertes **Rückziehen** einer Einzelachse:

MD37500 $MA_ESR_REACTION[AX1]=21

...

$AA_ESR_ENABLE[AX1] = 1

POLFA(AX1, 1, 20.0); AX1 wird die axiale Rückzugsposition 20.0 ; (absolut) zugeteilt

$AA_ESR_TRIGGER[AX1]=1 ; ab hier beginnt der Rückzug der AX1

POLFA(Achse, Typ) ; Erlaubte Kurzform bei der Programmierung

POLFA(Achse, 0/1/2) ; Schnelle Deaktivierung/Aktivierung

⚠WARNUNG

Wird die verkürzte Schreibweise benutzt und nur der Typ geändert, ist in der Anwendung dafür Sorge zu tragen, dass die Rückzugsposition bzw. der Rückzugsweg einen sinnvollen Wert enthält!

Es sollte nur in Ausnahmefällen die verkürzte Schreibweise genutzt werden.

Insbesondere nach:

Power On ist der Rückzugsweg bzw. die Rückzugsposition neu zu setzen.

POLFA(Achse, 1, $AA_POLFA[Achse]) ; bewirkt einen Vorlaufstop

POLFA(Achse, 1); bewirkt **keinen** Vorlaufstop

11.4 Geschwindigkeit

Für Positionierachsen gelten die achsspezifischen Geschwindigkeitsgrenzwerte und Beschleunigungsgrenzwerte.

Vorschubkorrektur

Es wirkt eine getrennte Vorschubkorrektur für Bahn- und Positionierachsen. Jede Positionierachse ist durch eine eigene achsspezifische Vorschubkorrektur beeinflussbar.

Eilgangkorrektur

Die Eilgangkorrektur wirkt nur auf die Bahnachsen. Positionierachsen haben keine Eilganginterpolation (nur axiale Linearinterpolation G01) und daher auch keine Eilgangkorrektur.

Vorschub

Die Positionierachsen verfahren mit dem für sie festgelegten achsspezifischen Vorschub. Wie Kap. "Bewegungsverhalten und Interpolationsvorgänge" zeigt, wird dieser Vorschub nicht durch Bahnachsen beeinflusst.

Der Vorschub wird als achsspezifische Geschwindigkeit in den Einheiten mm/min, inch/min oder Grad/min programmiert.

Der achsspezifische Vorschub ist durch den Achsnamen immer fest einer Positionierachse zugeordnet.

Wird eine Positionierachse ohne Vorschub programmiert, holt die Steuerung automatisch den Vorschub aus dem achsspezifischen Maschinendatum:

MD32060 $MA_POS_AX_VELO (Löschstellung für Positionierachsgeschwindigkeit).

Der programmierte achsspezifische Vorschub ist selbsthaltend bis Programmende.

Umdrehungsvorschub

In der Betriebsart JOG ist das Verhalten der Achse/Spindel auch von der Einstellung von SD41100 $SN_JOG_REV_IS_ACTIVE (Umdrehungsvorschub bei JOG aktiv) abhängig.

- Ist dieses Settingdatum aktiv, so wird eine Achse/Spindel immer mit Umdrehungsvorschub MD32050 $MA_JOG_REV_VELO (Umdrehungsvorschub bei JOG) bzw. MD32040 $MA_JOG_REV_VELO_RAPID (Umdrehungsvorschub bei JOG mit Eilgangsüberlagerung) in Abhängigkeit von der Masterspindel verfahren.
- Ist das Settingdatum nicht aktiv, so ist das Verhalten der Achse/Spindel abhängig vom SD43300 $SA_ASSIG_FEED_PER_REV_SOURCE (Umdrehungsvorschub für Positionsachsen/-spindeln)
- Ist das Settingdatum nicht aktiv, so ist das Verhalten einer Geometrieachse auf die ein Frame mit Rotation wirkt abhängig vom kanalspezifischen Settingdatum SD42600 $SC_JOG_FEED_PER_REV_SOURCE. (In der Betriebsart JOG Umdrehungsvorschub für Geometrieachsen auf die ein Frame mit Rotation wirkt)

11.5 Programmierung

11.5.1 Allgemein

Hinweis

Für die Programmierung von Positionierachsen bitte folgende Dokumentation beachten:
Literatur:
Programmierhandbuch Grundlagen; Kapitel: "Vorschubregelung" und "Spindelbewegung"

Hinweis

Die maximale Anzahl an Positionierachsen, die in einem Satz programmiert werden können, ist auf die maximale Anzahl vorhandener Kanalachsen begrenzt.

Definition

Positionierachsen werden durch folgende Angaben bestimmt:

- Achstyp: Positionierachse Typ 1, Typ 2 oder Typ 3
- Endpunktkoordinaten (im Absolutmaß oder Kettenmaß)
- Vorschub bei Linearachsen in [mm/min], bei Rundachsen in [Grad/min]

Beispiel: Positionierachse Typ 1

Programmcode	Kommentar
`POS[Q1]=200 FA[Q1]=1000`	`; Achse Q1 mit Vorschub 1000mm/min auf Position 200.`

Beispiel: Positionierachse Typ 2

Programmcode	Kommentar
`POSA[Q2]=300 FA[Q2]=1500`	`; Achse Q2 mit Vorschub 1500mm/min auf Position 300.`

Hinweis

Innerhalb eines Teileprogramms kann prinzipiell jede Achse Bahnachse oder Positionierachse sein. Innerhalb eines Bewegungssatzes ist aber jede Achse genau einem Achstyp zugeordnet.

Programmierung in Synchronaktion

Achsen können auch vollkommen asynchron zum Teileprogramm aus Synchronaktionen positioniert werden.

Beispiel:

Programmcode	Kommentar
`ID=1 WHENEVER $R==1 DO POS[Q4]=10 FA[Q3]=990`	`; Der axiale Vorschub wird fest vorgegeben.`

Literatur:
Programmierhandbuch Arbeitsvorbereitung; Kapitel "Bewegungssynchronaktionen"

Satzwechsel

Bei Positionierachse Typ 1 und Typ 2 ist der Satzwechsel einstellbar mit:

`FINEA=<Achsbezeichner>` bzw.
`FINEA[<Achsbezeichner>]`

`COARSEA=<Achsbezeichner>` bzw.
`COARSEA[<Achsbezeichner>]`

`IPOENDA=<Achsbezeichner>` bzw.
`IPOENDA[<Achsbezeichner>]`

Bei Positionierachse Typ 3 ist zusätzlich der Satzwechsel innerhalb der Bremsrampe der Einzelinterpolation einstellbar mit:

`IPOBRKA=<Achsbezeichner>` bzw.
`IPOBRKA(<Achsbezeichner>[,<Zeitpunkt*>])`

* Zeitpunkt des Satzwechsels, bezogen auf die Bremsrampe in %

Absolutmaß / Kettenmaß

Die Programmierung der Endpunktkoordinaten erfolgt im Absolutmaß (`G90`) oder im Kettenmaß (`G91`).

Beispiel	Bedeutung
	Programmierung der Endpunktkoordinaten:
`G90 POS[Q1]=200`	im Absolutmaß
`G91 POS[Q1]=AC(200)`	im Absolutmaß
`G91 POS[Q1]=200`	im Kettenmaß
`G90 POS[Q1]=IC(200)`	im Kettenmaß

Positionierachsen vom Typ 2 erneut programmieren

Bei Positionierachsen vom Typ 2 (Bewegung über Satzgrenzen) ist es nötig, im Teileprogramm feststellen zu können, ob die Positionierachse ihre Endposition erreicht hat. Erst dann ist es möglich, diese Positionierachse erneut zu programmieren (sonst wird ein Alarm gesetzt).

Wird `POSA` und anschließend wieder `POSA` mit `IPOBRKA` (Satzwechsel in der Bremsrampe) programmiert, dann wird kein Alarm gemeldet. Weitere Hinweise sind im Kapitel "Satzwechselzeitpunkt einstellbar" unter dem Sprachbefehl `IPOBKA` beschrieben.

Koordinierung (WAITP)

Mit dem Koordinierungsbefehl `WAITP` wird im Teileprogramm die Stelle gekennzeichnet, an der so lange gewartet werden soll, bis eine in einem früheren NC-Satz unter `POSA` programmierte Achse ihren Endpunkt erreicht hat.

`WAITP` steht in einem eigenen Satz.

Jede Achse, auf die gewartet werden soll, muss explizit angegeben werden.

Beispiel:

```
Programmcode                                    Kommentar
N10 G01 G90 X200 F1000 POSA[Q1]=200 FA[Q1]=500
N15 X400
N20 WAITP(Q1)                                   ; Die Programmbearbeitung
                                                  wird automatisch
                                                  gestoppt, bis Q1 auf
                                                  Position.
N25 X600 POS[Q1]=300                            ; Q1 ist Positionierachse
                                                  Typ 1 (Vorschub FA[Q1]
                                                  aus Satz N10).
N30 X800 Q1=500                                 ; Q1 ist Bahnachse
                                                  (Bahnvorschub F1000 aus
                                                  Satz N10).
```

Werkzeugkorrektur

Für Positionierachsen ist eine Werkzeuglängenkorrektur über die axiale Nullpunktverschiebung möglich. Damit kann z. B. der Positionierweg eines Laders verändert werden. Eine Anwendung für die axiale Nullpunktverschiebung als Ersatz für die Werkzeuglängenkorrektur ist beispielsweise dann gegeben, wenn der Lader mit Werkzeugen verschiedener Abmessungen ein Hindernis überfahren muss.

Programmende

Das Programmende (Programmzustand angewählt) wird so lange verzögert, bis alle Achsen (Bahnachsen + Positionierachsen) ihre programmierten Endpunkte erreicht haben.

11.5.2 Umdrehungsvorschub bei externer Programmierung

Über die beiden folgenden Settingdaten kann angegeben werden, dass der Umdrehungsvorschub für eine Positionierachse von einer anderen Rundachse/Spindel abgeleitet werden soll:

SD43300 ASSIGN_FEED_PER_REV_SOURCE (Umdrehungsvorschub für Positionierachsen/Spindel)

SD42600 JOG_FEED_PER_REV_SOURCE (Steuerung Umdrehungsvorschub in JOG)

Folgende Einstellungen sind möglich:

Wert	Bedeutung
0	Kein Umdrehungsvorschub angewählt.
>0	Der Umdrehungsvorschub wird abgeleitet von der Rundachse/Spindel mit dem hier angegebenen Maschinenachsindex.
-1	Der Umdrehungsvorschub wird abgeleitet von der Masterspindel des Kanals, in dem die Achse/Spindel aktiv ist.
-2	Der Umdrehungsvorschub wird abgeleitet von der Rundachse/Spindel mit dem Maschinenachsindex 0.
-3	Der Umdrehungsvorschub wird abgeleitet von der Masterspindel des Kanals, in dem die Achse/Spindel aktiv ist. Bei stehender Masterspindel ist kein Umdrehungsvorschub angewählt.

11.6 Satzwechsel

Positionierachsen können alleine oder zusammen mit Bahnachsen im NC-Satz programmiert werden.

Bahnachsen und Positionierachsen werden grundsätzlich getrennt interpoliert (Bahninterpolator und Achsinterpolatoren) und erreichen dadurch zu unterschiedlichen Zeitpunkten ihre programmierten Endpositionen.

Man kennt dabei drei Arten von Positionierachsen, die sich in ihrem Satzwechselverhalten unterscheiden:

- Typ 1

  Satzwechsel, wenn alle Bahnachsen und alle Positionierachsen ihre programmierten Endpunkte erreicht haben.

- Typ 2

  Satzwechsel, wenn alle Bahnachsen ihre programmierten Endpunkte entsprechend G601, G602, G603 erreicht haben.

- Typ 3

  Satzwechsel ist bereits innerhalb der Bremsrampe der Einzelachsinterpolation einstellbar, wenn das Bewegungsendkriterium und das Satzwechselkriterium für die Bahninterpolation erfüllt sind.

Positionierachse Typ 1

Satzwechsel zum programmierten Endpunkt aller Bahnachsen und Positionierachsen

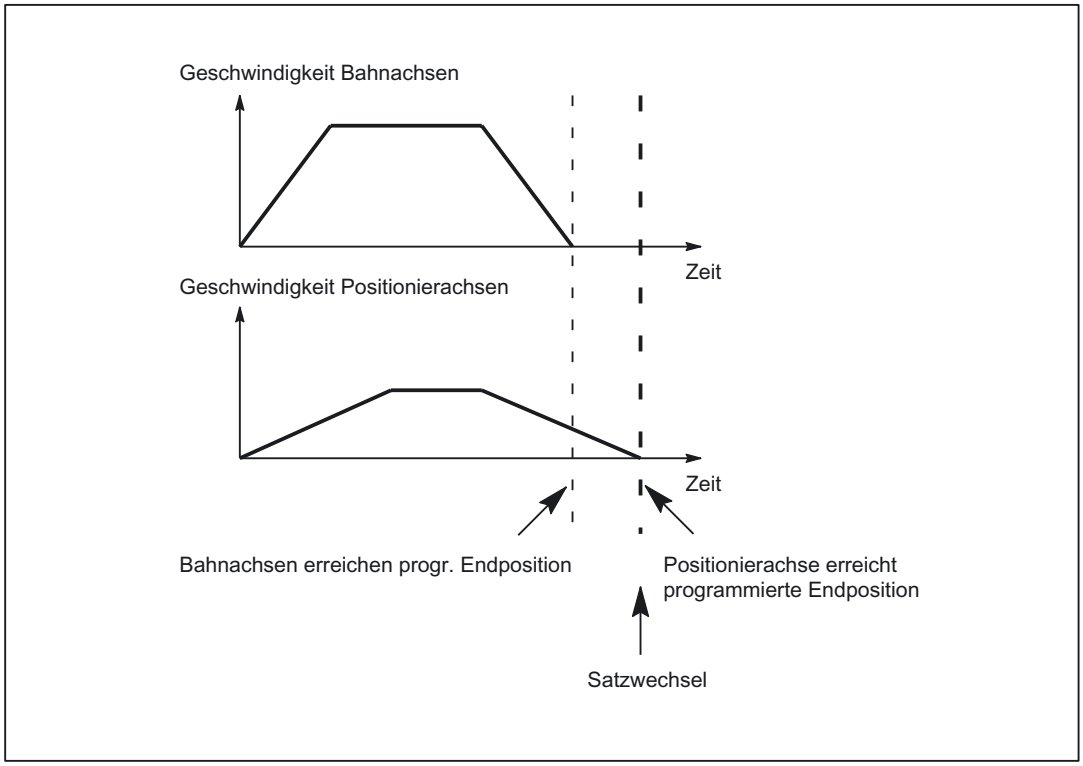


Bild 11-1 Satzwechsel bei Positionierachse Typ 1, Ablaufbeispiel

Eigenschaften der Positionierachse Typ 1

Bis SW 5 weisen Positionierachsen vom Typ 1 folgendes Verhalten auf:

- Der Satzwechsel erfolgt (NC-Satz beendet), wenn alle Bahnachsen und Positionierachsen das jeweilige Bewegungsendekriterium erreicht haben.
- Ein Bahnsteuerbetrieb (G64) für Bahnachsen ist dabei nur dann möglich, wenn die Positionierachsen vor den Bahnachsen ihr Bewegungsendekriterium erfüllen (Dies ist im Beispiel im obigen Bild nicht der Fall).
- Programmierung mit

  POS[Name] = Endpunkt FA[Name] = Vorschub

  oder verkürzt mit

  POSA[Name] = Endpunkt

  wobei dann der Vorschub aus dem MD32060 $MA_POS_AX_VELO genommen wird.
- Die programmierte Anweisung ist satzweise wirksam. Geometrie- und Synchronachsen werden mit den Anweisungen aus dem Bahnachsverbund herausgetrennt und verfahren mit einer achsspezifischen Geschwindigkeit.

Ab SW 5 haben Positionierachsen vom Typ 1 zu bisherigen Softwareständen zusätzlich folgendes Verhalten:

- Bei Bahnsteuerbetrieb (G601, G602, G603) verfahren Positionierachsen bzw. Spindeln zum Positionierende, wenn das Bewegungsendekriterium FINEA, COARSA oder IPOENDA erfüllt ist.

Positionierachse Typ 2

Satzwechsel zum programmierten Endpunkt aller Bahnachsen

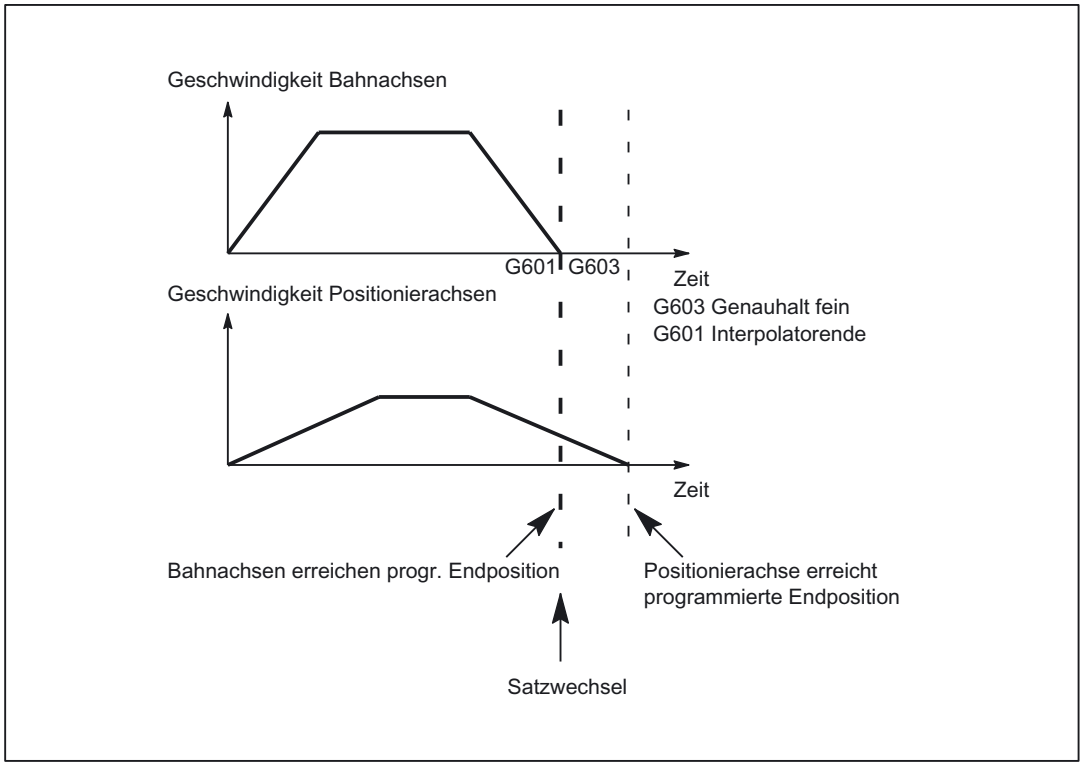


Bild 11-2 Satzwechsel bei Positionierachse Typ 2, Ablaufbeispiel

Eigenschaften der Positionierachse Typ 2

Bis SW 5 weisen Positionierachsen vom Typ 2 folgendes Verhalten auf:

- Der Satzwechsel erfolgt (NC-Satz beendet), wenn die Bahnachsen ihre programmierten Endpositionen bezüglich G601, G602, G603 erreicht haben.
- Die Positionierachsen können dabei über Satzgrenzen hinweg auf ihre programmierten Endpositionen fahren.
- Da es keinen zeitlichen Zusammenhang zwischen "NC-Satz beendet" und dem Erreichen der programmierten Endpunkte von Positionierachsen Typ 2 gibt, wird eine Synchronisation der Positionierachsen mit den Koordinierungsbefehl WAITP ermöglicht (siehe Kapitel "Programmierung")
- Wird eine Positionierachse erneut programmiert, ohne das diese die vorhergehende Position einschließlich dessen Bewegungsendekriterium erreicht hat, so wird der Alarm "Achse kann nicht repositioniert werden" gesetzt.

- Programmierung mit

  POSA[Name] = Endpunkt FA[Name] = Vorschub

  oder verkürzt mit

  POSA[Name] = Endpunkt

  wobei dann der Vorschub aus dem MD32060 $MA_POS_AX_VELO genommen wird.
- Die programmierte Anweisung ist satzweise wirksam. Geometrie- und Synchronachsen werden mit den Anweisungen aus dem Bahnachsverbund herausgetrennt und verfahren mit einer achsspezifischen Geschwindigkeit.

11.6.1 Satzwechselzeitpunkt einstellbar

Positionierachse Typ 3

Satzwechselzeitpunkt einstellbar für Einzelachsinterpolation

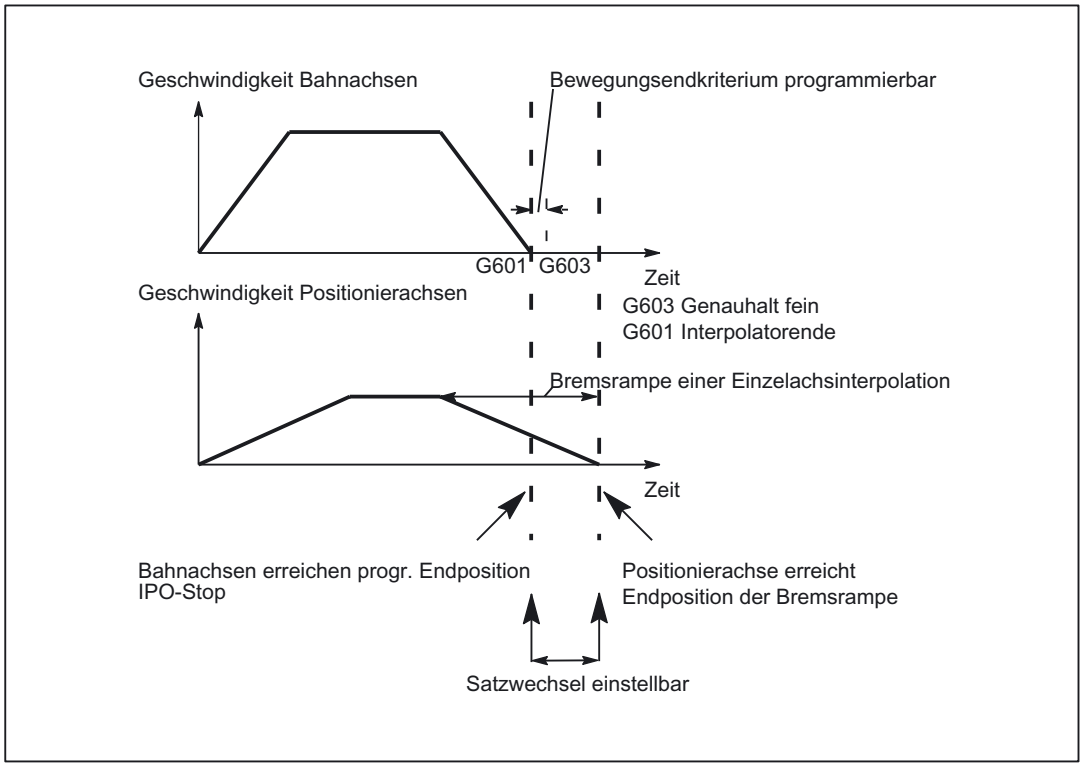


Bild 11-3 Einstellbarer Satzwechsel bei Positionierachse Typ 3, Ablaufbeispiel

Eigenschaften der Positionierachse Typ 3

Bei Positionierachsen vom Typ 3 ist das Bewegungsendekriterium mit FINEA, COARSEA oder IPOENDA programmierbar. Das Satzwechselkriterium ist innerhalb der Bremsrampe der Einzelachsinterpolation einstellbar.

- Der Satzwechsel erfolgt (NC-Satz beendet), wenn alle Bahnachsen oder Spindeln die programmierten Bewegungsendkriterien erfüllt haben. Zusätzlich muss die Positionierachse bzw. Spindel das eingestellte Satzwechselkriterium innerhalb der Bremsrampe der Einzelachsinterpolation erreicht haben. Nur wenn beide Bedingungen zutreffen, erfolgt der Satzwechsel.
- Das Positionsende der Positionierachsen kann im programmierbaren Bewegungsendekriterium mit einen der folgenden Sprachbefehlen

  FINEA[Achse]: Bewegungsende von "Genauhalt fein" oder

  COARSEA[Achse]: Bewegungsende von "Genauhalt grob" oder

  IPOENDA[Achse]: Bewegungsende beim Erreichen von "IPO-Stop"

  festgelegt werden.
- Für das Positionsende von Positionierspindeln gilt folgende Syntax:

  FINEA[Sn]: Bewegungsende von "Genauhalt fein" oder

  COARSEA[Sn] Bewegungsende von "Genauhalt grob" oder

  IPOENDA[Sn]: Bewegungsende beim Erreichen von "IPO-Stop"

  Sn: Spindelnummer, 0... max. Spindelnummer oder

  Achse: Achsbezeichner, X, Y, Z.

Kommandoachsen

Kommandoachsen reagieren sofort auf eine erneute Programmierung weshalb keine Änderungen vorgenommen werden sollten. Jede einzelne Achse hat ihren eigenen Achsinterpolator und ihren eigenen Vorschub. Eine Achse kann nicht gleichzeitig aus dem Teileprogramm und aus Synchronaktionen bewegt werden.

Die Bewegungsendekriterien FINEA, COARSEA, IPOENDA gelten für den Übergang zu einer neutralen Achse. Anschließend kann die Achse wieder über NC-Programm, als PLC-Achse usw. verwendet werden.

Pendelachsen

Pendelachsen bremsen immer auf ihre jeweilige Umkehrposition und fahren dann in Gegenrichtung. Damit ist bei Pendelachsen keine Erweiterung nötig.

Hinweis

Das Verhalten von PLC-Achsen bei Satzwechsel ist im Kapitel "Beeinflussung durch PLC" beschrieben.

Weitere Informationen zum Satzwechsel beim programmierbaren Bewegungsendekriterium `FINEA`, `COARESA` und `IPOENDA` entnehmen Sie bitte:

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; Spindeln (S1), Kapitel Spindelbetriebsarten

/FB1/ Funktionshandbuch Grundfunktionen; Vorschübe (V1), Kapitel Programmierbare Einzelachsdynamik

---

IPOBRKA

Mit den Sprachbefehl `IPOBRKA` wird das Bewegungsendekriterium in der Bremsrampe entweder

- im Teileprogramm: Satzwechsel im Teileprogramm oder
- in Synchronaktion: Technologiezyklen

aktiviert.

Ab SW 6.2 haben Positionierachsen vom **Typ 2** zu bisherigen Softwareständen folgendes abweichende Verhalten:

Wird `POSA` und anschließend wieder `POSA` mit `IPOBRKA` (Satzwechsel in der Bremsrampe) programmiert, dann wird kein Alarm gemeldet. Es wird mit der erneuten `POSA`-Bewegung gewartet, bis die vorhergehende `POSA` Bewegung den Satzwechselpunkt in der Bremsrampe erreicht hat.

Bedingung für Satzwechsel

Sind die Bewegungsendekriterien für alle im Satz zu bearbeitenden Achsen bzw. Spindeln und zusätzlich das Satzwechselkriterium erfüllt, so erfolgt der Satzwechsel. Dies gilt sowohl für Teileprogrammsätze als auch für Technologiezyklussätze.

Satzwechselkriterium Bremsrampe in SD43600

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

Über obige das Settingdatum "Satzwechselkriterium Bremsrampe" kann bei Einzelachsinterpolation der Zeitpunkt des Satzwechsels vom Einsatzzeitpunkt der Bremsrampe mit 100% bis zum IPO-Stop mit 0% eingestellt werden.

Bei 100% ist das Satzwechselkriterium der Positionierachse bereits zum Einsatzzeitpunkt der Bremsrampe erfüllt.

Bei 0% ist das Satzwechselkriterium identisch mit `IPOENDA`.

Vorteile der Prozentangabe

Vorteile dieser Prozentangabe in SD43600 sind:

- Das Satzwechselkriterium ist positionsunabhängig und damit abhängig vom eingestellten Override.
- Bei maximalen Override wird die größte Überschleif-Abweichung gefahren.
- Mit kleinerem Override wird die Abweichung immer geringer.

Das Satzwechselkriterium Bremsrampe kann wie die bisherigen axialen Bewegungsendekriterien mit $AA_MOTEND[Achse] = 4 abgefragt werden.

Bremsrampe mit Toleranzfenster

Zusätzliches Satzwechselkriterium in der Bremsrampe

Zum bereits bestehenden Satzwechselkriterium in der Bremsrampe kann ein zusätzliches Toleranzfenster angewählt werden. Die Freigabe erfolgt erst, wenn die Achse

- wie bisher den vorgegebenen %-Wert ihrer Bremsrampe erreicht hat **und**
- ihre aktuelle Istposition oder Sollposition nicht weiter als eine einstellbare Toleranz von der Endposition der Achse im Satz entfernt ist.

Spätestens wenn der Interpolator die Endposition erreicht wird die Achse freigegeben. Das Toleranzfenster ist modal wirksam.

Für das Satzwechselkriterium Bremsrampe **mit Toleranzfenster** gilt:

Die Sollposition wird gemeldet mit $AA_MOTEND[Achse] = 5

Die Istposition wird gemeldet mit $AA_MOTEND[Achse] = 6

Randbedingungen

Satzwechsel und Wechsel des Achszustandes

Ein vorzeitiger Satzwechsel ist nicht möglich:

- Beim Pendeln mit Teilzustellung

  muss die satzbezogene Pendelbewegung aktiv bleiben, bis die Achse mit Teilzustellung ihre Endposition erreicht hat.

- Bei einer Handradvorgabe

  wirkt das zuletzt eingestellte Bewegungsendekriterium.

Im nachfolgenden Satz darf die Achse, für die ein Satzwechsel innerhalb der Bremsrampe erfolgt ist, nur im selben Achszustand erneut programmiert werden.

Beim Wechsel des Achszustandes z. B. auf POS folgt SPOS wirkt das zuletzt programmierte Bewegungsendekriterium FINEA, COARSEA, IPOENDA.

Dies gilt auch:

- wenn aus einer Positionierachse eine Bahnachse wird oder
- wenn eigens auf das Ende der Positionierbewegung gewartet wird

  (WAITP, M30, Technologiezyklusende, Vorlaufstop).

- wenn Geschwindigkeitsüberlagerung aktiviert oder deaktiviert wird.

Aktivierung und Deaktivierung

Bewegungsendekriterium IPOBRKA und genauer Einsatzzeitpunkt.

Für Teileprogrammbearbeitung:

- Das Bewegungsendekriterium Bremsrampe kann über den NC-Befehl IPOBRKA aktiviert werden.
- Der genaue Einsatzzeitpunkt wird über das Settingdatum

  SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE bestimmt.
- Mit der nächsten Programmierung eines axialen Bewegungsendekriteriums `FINEA`, `COARSEA` oder `IPOENDA` für die entsprechende Achse wird `IPOBRKA` deaktiviert.
- Über den NC-Befehl ADISPOSA kann die Größe des Toleranzfenster zum Bewegungsendekriterium Bremsrampe programmiert werden und wird über das Settingdatum SD43610 $SA_ADISPOSA_VALUE[Achse] festgelegt.

Für Technologiezyklen:

- Das Bewegungsendekriterium Bremsrampe kann über die Synchronaktion `IPOBRKA` aktiviert werden.
- Mit der nächsten Programmierung des axialen Bewegungsendekriteriums `FINEA`, `COARSEA` oder `IPOENDA` für die entsprechende Achse in einer Synchronaktion wird `IPOBRKA` deaktiviert.
- Die Größe des Toleranzfensters zum Bewegungsendekriterium Bremsrampe kann über die Synchronaktion ADISPOSA programmiert werden.

Hinweis

Weitere Informationen zur Programmierung von Positionierachsen siehe:

Literatur:

/PG/ Programmierhandbuch Grundlagen, Kapitel "Vorschubregelung und Spindelbewegung"

/PGA/ Programmierhandbuch Arbeitsvorbereitung, Kapitel "Spezielle Wegbefehle"

Beispiele

Für Satzwechselkriterium "Bremsrampe" im **Teileprogramm**:

```
                        ; Defaulteinstellung wirksam
N10 POS[X] = 100        ; Satzwechsel erfolgt, wenn die X-Achse die
                        ; Position 100 und Genauhalt fein erreicht hat.
N20 IPOBRKA(X,100)      ; Satzwechselkriterium Bremsrampe
N30 POS[X] = 200        ; Satzwechsel erfolgt, sobald die X-Achse
                        ; zu bremsen beginnt
N40 POS[X] = 250        ; X-Achse bremst nicht auf Position 200,
                        ; sondern fährt weiter
                        ; auf Position 250, sobald die X-Achse
                        ; zu bremsen beginnt, erfolgt der Satzwechsel.
```

```
N50 POS[X] = 0          ; die X-Achse bremst und fährt auf
                        ; Position 0 zurück, der Satzwechsel
                        ; erfolgt bei Position 0 Genauhalt fein
N60 X10 F100
N70 M30
```

Bei einer Richtungsumkehr (N50) muss erst auf die Zielposition gebremst werden, bevor in die Gegenrichtung gefahren werden kann.

Alternativ könnte im Satz N20 auch IPOBRKA(X) geschrieben werden, wenn im SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE[AX1] bereits der Wert 100 eingetragen ist.

Für Satzwechselkriterium "Bremsrampe" in **Synchronaktion**:

Im Technologie-Zyklus:

```
FINEA                   ; Bewegungsendekriterium Genauhalt fein
POS[X] = 100            ; Technologie-Zyklus-Satzwechsel erfolgt,
                        ; wenn die X-Achse die Position 100
                        ; und Genauhalt fein erreicht hat.
IPOBRKA(X,100)          ; Satzwechselkriterium Bremsrampe aktivieren
POS[X] = 200            ; Technologie-Zyklus-Satzwechsel erfolgt,
                        ; sobald die X-Achse zu bremsen beginnt.
POS[X] = 250            ; X-Achse bremst nicht auf Position 200,
                        ; sondern fährt weiter auf Position 250
N40 POS[X] = 250        ; sobald die X-Achse zu bremsen beginnt,
                        ; erfolgt der Satzwechsel im Technologie-Zyklus.
POS[X] = 0              ; X-Achse bremst und fährt auf Position 0 zurück
M17                     ; der Satzwechsel erfolgt bei Position 0 und Genauhalt
                        fein.
```

Mit Toleranzfenster

Für Satzwechselkriterium "Bremsrampe" im **Teileprogramm**:

```
                          ; Defaulteinstellung wirksam
N10 POS[X] = 100          ; Satzwechsel erfolgt, wenn die X-Achse die
                          ; Position 100 und Genauhalt fein erreicht hat.
N20 IPOBRKA(X,100)        ; Satzwechselkriterium Bremsrampe
N21 ADISPOSA(X,1,0.5)     , Toleranzfenster Sollposition für X-Achse
                          ; mit 0,5 festlegen
N30 POS[X] = 200          ; Satzwechsel erfolgt, sobald die X-Achse
                          ; zu bremsen beginnt und die
                          ; Sollposition ≥ 199.5 ist.
N40 POS[X] = 250          ; X-Achse bremst nicht auf Position 200,
                          ; sondern fährt weiter auf Position 250,
                          ; sobald die X-Achse zu bremsen beginnt
                          ; und die Position ≥ 249.5 ist,
```

```
                          ; erfolgt der Satzwechsel.
N50 POS[X] = 0            ; die X-Achse bremst und fährt auf
                          ; Position 0 zurück
                          ; der Satzwechsel erfolgt bei
                          ; Position 0 Genauhalt fein
N60 X10 F100
N70 M30
```

Bei einer Richtungsumkehr (N50) muss erst auf die Zielposition gebremst werden, bevor in die Gegenrichtung gefahren werden kann.

Für Satzwechselkriterium "Bremsrampe" in **Synchronaktion**:

Im Technologie-Zyklus:

```
FINEA                     ; Bewegungsendekriterium Genauhalt fein
POS[X] = 100              ; Technologie-Zyklus-Satzwechsel erfolgt,
                          ; wenn die X-Achse die Position 100
                          ; und Genauhalt fein erreicht hat.
ADISPOSA(X,2,0.3)         ; Toleranzfenster Istposition für
                          ; X-Achse mit 0,3 festlegen
IPOBRKA(X,100)            ; Satzwechselkriterium Bremsrampe aktivieren
POS[X] = 200              ; Technologie-Zyklus-Satzwechsel erfolgt,
                          ; sobald die X-Achse zu bremsen beginnt und
                          ; die Istposition der X-Achse ≥ 199.7 ist.
POS[X] = 250              ; X-Achse bremst nicht auf Position 200,
                          ; sondern fährt weiter
N40 POS[X] = 250          ; auf Position 250, sobald die X-Achse
                          ; zu bremsen beginnt und die Position
                          ; ≥ 249.7 ist, erfolgt der Satzwechsel im
                          ; Technologie-Zyklus.
POS[X] = 0                ; die X-Achse bremst und fährt auf
                          ; Position 0 zurück
M17                       ; der Satzwechsel erfolgt bei Position 0
                          ; und Genauhalt fein.
```

11.6.2 Bewegungsendekriterien bei Satzsuchlauf

Letzter Satz dient als Behälter

Jeweils das zuletzt programmierte Bewegungsendekriterium einer Achse wird aufgesammelt und in einem Aktionssatz ausgegeben. Der letzte im Suchlauf behandelte Satz mit einem programmierten Bewegungsendekriterium dient als Behälter in dessen alle programmierten Bewegungsendkriterien für alle Achsen eingestellt werden.

Beispiel

Für zwei Aktionssätze mit Bewegungsendekriterien für drei Achsen:

```
N01 G01 POS[X]=20 POS[Y]=30
 IPOENDA[X]                      ; zuletzt programmierte
                                 ; Bewegungsendekriterium IPOENDA
N02 IPOBRKA(Y, 50)               ; zweiter Aktionssatz für
                                 ; die Y-Achse IPOENDA
N03 POS[Z]=55 FINEA[Z]           ; zweiter Aktionssatz für die Z-Achse FINEA
N04 $A_OUT[1]=1                  ; erster Aktionssatz für Ausgabe
                                 ; als digitaler Ausgang
N05 POS[X]=100                   ;
N06 IPOBRKA(X, 100)              ; zweiter Aktionssatz für
                                 ; die X-Achse IPOBRKA Behälter
...                              ; für alle programmierten
                                 ; Bewegungsendekriterien
ZIEL:                            ; Satzsuchlaufziel
```

Im ersten Aktionssatz wird der digitale Ausgang:

$A_OUT[1]=1 ausgegeben.

Im zweiten Aktionssatz werden die Bewegungsendekriterien:

für X-Achse IPOBRKA / $SA_IPOBRAKE_BLOCK_EXCHANGE[AX1]=100

für Y-Achse IPOBRKA / $SA_IPOBRAKE_BLOCK_EXCHANGE[AX2]=50

für Z-Achse FINEA eingestellt. Außerdem wird für die X-Achse das zuletzt programmierte Bewegungsendkriterium IPOENDA gemerkt.

11.7 Beeinflussung durch die PLC

PLC-Achsen

PLC-Achsen werden von der PLC verfahren und können sich asynchron zu allen übrigen Achsen bewegen. Die Fahrbewegungen laufen losgelöst von Bahn- und Synchronaktionen ab.

Literatur:
Funktionshandbuch Grundfunktionen; "PLC-Grundprogramm für SINUMERIK 840D sl" (P3) bzw. "PLC für SINUMERIK 828D" (P4)

Konkurrierende Positionierachsen

Mit dem Funktionsbaustein FC18 können bei SINUMERIK 840D sl konkurrierende Positionierachsen von der PLC gestartet werden.

Kanalspezifische Signale

Alle kanalspezifischen Signale wirken gleichermaßen auf Bahnachsen und Positionierachsen.

Eine Ausnahme bilden lediglich folgende Signale:

- NST DB21, ... DBB4 ("Vorschubkorrektur")
- NST DB21, ... DBX6.2 ("Restweg löschen")

Achsspezifische Signale

Positionierachsen haben folgende zusätzliche Signale:

- NST DB31, ... DBX76.5 ("Positionierachse")
- Vorschub für Positionierachsen/Spindeln (FA)
- NST DB31, ... DBB0 ("Vorschubkorrektur") achsspezifisch
- NST DB31, ... DBX2.2 ("Restweg/Spindel-Reset") Restweg löschen achsspezifisch

Einzelachsfunktionen PLC-kontrollierter Achsen

Das Verhalten einzelner PLC-Achsen kann mit dem Maschinendatum:
MD30460 $MA_BASE_FUNCTION_MASK
wie folgt verändert werden:

- Bit 4 = 1

  Die Achse ist ausschließlich PLC-kontrolliert.

- Bit 5 = 1

  Die Achse ist eine fest zugeordnete PLC-Achse.

- Bit 6 = 1

  Das kanalspezifische NC/PLC-Nahtstellensignal:
  DB21, ... DBX6.0 ("Vorschubsperre")
  wirkt **nicht** auf die Achse, wenn diese eine PLC-kontrollierte Achse ist.

- Bit 7= 1

  Das kanalspezifische NC/PLC-Nahtstellensignal:
  DB21, ... DBX36.3 ("alle Achsen stehen")
  wird **unabhängig** von der Achse gesetzt, wenn diese eine PLC-kontrollierte Achse ist.

Für eine PLC-kontrollierte Achse:

- wirkt das kanalspezifische NC/PLC-Nahtstellensignal DB21, ... DBX6.0 ("Vorschubsperre"), wenn im Maschinendatum MD30460 $MA_BASE_FUNCTION_MASK das Bit 6 = 0 ist.
- wird das kanalspezifische NC/PLC-Nahtstellensignal DB21, ... DBX36.3 ("alle Achsen stehen") nur gesetzt, wenn im Maschinendatum MD30460 $MA_BASE_FUNCTION_MASK das Bit 7 = 0 ist.

Wird versucht, eine **ausschließlich PLC-kontrollierte Achse** dem NC-Programm zuzuordnen oder die Achse für das NC–Programm anzufordern, so wird dies mit dem Alarm 26075 abgelehnt. Analog wird für eine fest zugeordnete PLC Achse der Alarm 26076 gemeldet.

Eine **fest zugeordnete PLC-Achse** wird beim Hochlauf "Neutrale Achse". Bei einer Verfahranforderung über die NC/PLC-Nahtstelle wird eine konkurrierende Positionierachse ohne einen vorhergehenden Achstausch automatisch zur PLC-Achse.

Achstausch durch PLC

Über das axiale Nahtstellenbyte NCK→PLC NST DB31, ... DBB68 wird der Typ einer Achse für den Achstausch dem PLC mitgeteilt:

- NST DB31, ... DBX68.0-68.3 ("NC-Achse/Spindel im Kanal") Kanal 1 bis 10
- NST DB31, ... DBX68.4 ("neuer Typ von PLC angefordert")
- NST DB31, ... DBX68.5 ("Achstausch möglich")
- NST DB31, ... DBX68.6 ("neutrale Achse/Spindel")
- NST DB31, ... DBX68.7 ("PLC-Achse/Spindel")

Ist das NST DB31, ... DBX68.5 ("Achstausch möglich") == 1 so ist eine Achstausch-Anforderung vom PLC aus möglich.

Literatur:
Funktionshandbuch Erweiterungsfunktionen; "BAGs, Kanäle, Achstausch" (K5)

11.7.1 Konkurrierende Positionierachsen von der PLC starten

Anstoß von der PLC

Werden konkurrierende Positionierachsen von der PLC angestoßen, so wird dazu der FC 18 aufgerufen und mit den folgenden Parametern versorgt:

- Achsname bzw. Achsnummer
- Endposition
- Vorschub (bei Vorschub=0 wird der Vorschub aus dem Maschinendatum MD32060 $MA_POS_AX_VELO genommen).
- Bezugsmaß (G90),
  Kettenmaß (G91),
  Bezugsmaß auf kürzestem Weg für Rundachsen (Rundachsname = DC(Wert))

Folgende Funktionen sind fest vorgegeben:

- Linearinterpolation (G01)
- Vorschub in mm/min bzw. grad/min (G94)
- Genauhalt (G09)
- aktuell angewählte einstellbare Nullpunktverschiebungen sind gültig

Da jede Achsen genau einem Kanal zugeordnet ist, kann die Steuerung aus dem Achsnahmen bzw. der Achsnummer den richtigen Kanal auswählen und die konkurrierende Positionierachse in diesem Kanal starten.

11.7.2 PLC kontrollierte Achsen

Aktionen der PLC

Für eine Maschinenachse 1 werden folgende Aktionen der PLC als Reaktion des NCK gegenübergestellt:

- Maschinenachse als PLC-Achse über den FC 18 starten
- NC-Start oder NC-Sop auslösen
- Axialen `STOP`, `RESUME` oder `RESET` auslösen
- `NC-RESET` auslösen
- Reglerfreigabe für die Maschinenachse wegnehmen oder setzen
- Kontrolle der Maschinenachse an NC abgeben

Beispiele für Reaktionen des NCK

In folgender Gegenüberstellung sind Aktionen der PLC als Reaktion des NCK dargestellt.

Aktionen der PLC	Reaktion des NCK
Maschinenachse 1, im 1. Kanal beheimatet, als PLC–Achse über den FC 18 starten	
NST DB21, ... DBX7.4 (”NC–Stop Achsen plus Spindel”) auslösen	Maschinenachse 1 wird angehalten.
NST DB21, ... DBX7.1 (”NC–Start”) auslösen	Maschinenachse 1 fährt weiter.
PLC will Maschinenachse 1 kontrollieren, NST DB31, ... DBX28.7 ==1 (”PLC kontrolliert Achse”)	Kontrolle der Maschinenachse 1 wird an PLC abgegeben. NST DB31, ... DBX63.1==1 (”PLC kontrolliert Achse”)
NST DB21, ... DBX7.4 (”NC–Stop Achsen plus Spindel”)auslösen	Maschinenachse 1 wird nicht angehalten.
Axialen Achsstop auslösen NST DB31, ... DBX28.6 (”AxStop, Halt”)	Maschinenachse 1 wird angehalten NST DB31, ... DBX63.2==1 (”Achsstop aktiv”)
Axialen RESUME auslösen NST DB31, ... DBX28.2 (”AXRESUME")	Maschinenachse 1 fährt wieder NST DB31, ... DBX63.2==0 (”Achsstop aktiv”)
NC–Reset auslösen NST DB21, ... DBX7.7 (”Reset”) auslösen	keine Auswirkung auf Maschinenachse 1
Axialen RESET auslösen NST DB31, ... DBX28.1 (”AXRESET”)	Maschinenachse 1 wird gestoppt und die Verfahrbewegung wird abgebrochen. NST DB31, ... DBX63.2==1 (”Achsstop aktiv”) auf 0 zurückgesetzt, ihre axialen Maschinendaten werden eingelesen, NST DB31, ... DBX63.0 (”AXRESET DONE”) wird auf 1 gesetzt und NST DB31, ... DBX63.2 (”Achsstop aktiv”) auf 0 zurückgesetzt.
Maschinenachse 1 als PLC–Achse über den FC 18 starten	NST DB31, ... DBX63.0==0 (”AXRESET DONE”)
Reglerfreigabe für die Maschinenachse 1 NST DB31, ... DBX2.1==0 (”Reglerfreigabe”) wegnehmen	es wird der axiale Alarm 21612 ”Achse %1 Mess–Systemwechsel” gemeldet.
Axialen RESUME auslösen NST DB31, ... DBX28.2 (”AXRESUME”)	axialer Alarm 21612 (”Achse %1 Mess–Systemwechsel”) wird gelöscht und Fahrbefahl NST DB21, ... DBX40.7 (”Fahrbefehl plus”) wird ausgegeben. Die Maschinenachse 1 fährt wegen fehlender Reglerfreigabe jedoch nicht.
Reglerfreigabe für die Maschinenachse 1 NST DB31, ... DBX2.1==1 (”Reglerfreigabe”) setzen	Die Maschinenachse 1 fährt auf den programmierten Endpunkt.
Axialen RESET auslösen NST DB31, ... DBX28.1 (”AXRESET”)	Maschinenachse 1 wird gestoppt und Verfahren wird abgebrochen, ihre axialen Maschinendaten werden eingelesen und das Ende des axialen RESET NST DB31, ... DBX63.0==0 (”AXRESET DONE”) wird gemeldet.
PLC gibt die Kontrolle der Maschinenachse 1 an NCK ab. NST DB31, ... DBX28.7==0 (”PLC kontrolliert Achse”)	NCK übernimmt Kontrolle der Maschinenachse NST DB31, ... DBX63.1==0 (”PLC kontrolliert Achse”) und setzt das Endesignal des axialen RESET NST DB31, ... DBX63.0 (”AXRESET DONE”) von 1 auf 0 zurück.

11.7.3 Steuerungsverhalten PLC-kontrollierter Achsen

Verhalten bei Kanal-Reset, NEWCONFIG, Satzsuchlauf und MD30460

Steuerungsverhalten bei	PLC kontrollierte Achse
Betriebsartenwechsel und NC–Programmbeeinflussungen	sind hierfür unabhängig.
Kanal–RESET	es werden keine axialen Maschinendaten wirksam und eine Verfahrbewegung wird nicht abgebrochen.
NEWCONFIG	es werden keine axialen Maschinendaten wirksam.
Satzsuchlauf SERUPRO vom Typ 5	werden während SERUPRO wie im normalen Ablauf simuliert bearbeitet, z. B. PLC übernimmt oder übergibt die Kontrolle dieser Achse, die auch über PLC verfährt.
allen Satzsuchlaufvarianten vom Typ 1, 2, 4 und 5	PLC übernimmt die Kontrolle der Achse **vor** dem Anfahrsatz und ist für die Positionierung dieser Achse verantwortlich.
NC–geführten Rückzug, der mit $AC_ESR_TRIGGER aktiviert wird.	hat keine Auswirkung und wirkt nur auf die spezifizierte PLC-kontrollierte Achse.
Maschinendatum: MD30460 $MA_BASE_FUNKTION_MASK	**die nicht ausschließlich von PLC kontrolliert wird**
Bit 4 = 0	kann mittels Achstausch GET(Achse) bzw. AXTOCHAN(Achse, Kanal) **nicht** unmittelbar zu einer vom NC–Programm kontrollierte Achse geändert werden, siehe *** Hinweis Achstausch**.
Bit 4 = 1	kann nicht für das NC–Programm angefordert werden. GET bzw. AXTOCHAN vom NC–Programm oder einer Synchronaktion sowie die Programmierung der Achse im NC–Programm wird mit Alarm 26075 abgelehnt.
MD30460 $MA_BASE_FUNKTION_MASK	**Für die PLC-kontrollierte Achse**
Bit 6 = 1	wirkt das kanalspezifische NST DB 21, ... DBX6.0 ("Vorschubsperre") nicht. Diese Achse wird bei aktivierter Vorschub-Vorschubsperre nicht angehalten, sondern sie verfährt weiter.
Bit 7 = 1	wird die Achse bei der Bildung des NST DB 21, ... DBX36.3 ("alle Achsen stehen") nicht berücksichtigt. Dieses Signal wird auch dann mit 1 ausgegeben, wenn alle anderen Achsen des Kanals stehen und nur die PLC-kontrollierte Achse aktiv ist.

*** Hinweis zum Achstausch**

Dieser Achstausch von "Neutrale Achse" in "NC–Programm-Achse" findet erst statt, wenn die PLC entsprechend Use Case "Kontrolle der Achse abgeben" auch wirklich abgegeben hat. Das Warten auf diesen Achstausch wird über die Bedientafelfront HMI angezeigt.

11.8 Verhalten bei Sonderfunktionen

11.8.1 Probelaufvorschub (DRY RUN)

Der Probelaufvorschub wirkt auch für Positionierachsen, außer der programmierte Vorschub ist größer als der Probelaufvorschub.

Die Wirksamkeit des in SD42100 $SA_DRY_RUN_FEED eingetragenen Probelaufvorschubs kann mit SD42101 $SA_DRY_RUN_FEED_MODE gesteuert werden. Siehe

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; Vorschübe (V1)

11.8.2 Einzelsatz

Positionierachse Typ 1

Der Einzelsatz wirkt auch bei Positionierachsen Typ 1.

Positionierachse Typ 2

Positionierachsen vom Typ 2 laufen auch bei Einzelsatzbetrieb über Satzgrenzen hinweg weiter.

Positionierachse Typ 3

Positionierachsen vom Typ 3 laufen auch bei Einzelsatzbetrieb über Satzgrenzen hinweg weiter.

11.9 Beispiele

11.9.1 Bewegungsverhalten und Interpolationsvorgänge

In diesem Beispiel stellen die beiden Positionierachsen Q1 und Q2 zwei getrennte Bewegungseinheiten dar. Sie stehen in keinem interpolatorischen Zusammenhang. Im Beispiel werden die Positionierachsen als Typ 1 (z. B. in N20) und als Typ 2 (z. B. in N40) programmiert.

Programmbeispiel

```
N10 G90 G01 G40 T0 D0 M3 S1000
N20 X100 F1000 POS[Q1]=200 POS[Q2]=50 FA[Q1]=500
FA[Q2]=2000
N30 POS[Q2]=80
N40 X200 POSA[Q1] = 300 POSA[Q2]=200] FA[Q1]=1500
N45 WAITP[Q2]
N50 X300 POSA[Q2]=300
N55 WAITP[Q1]
N60 POS[Q1]=350
N70 X400
N75 WAITP[Q1, Q2]
N80 G91 X100 POS[Q1]=150 POS[Q2]=80
N90 M30
```

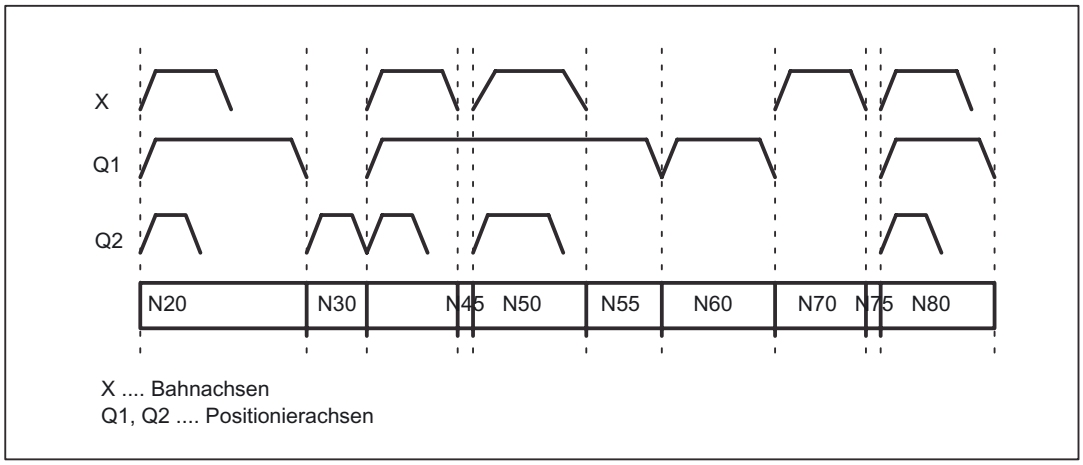


Bild 11-4 Zeitliche Abläufe von Bahnachsen und Positionierachsen

11.9.1.1 Bahnachsen fahren bei G0 nicht interpolierend

Beispiel bei G0 für Positionierachsen

Bahnachsen fahren bei Eilgangsbewegung (G0) nicht interpolierend als Positionierachsen:

```
                                ; Aktivierung von nicht-linearer
                                ; Interpolation
                                ; MD20730 $MC_GO_LINEAR_MODE == FALSE
                                ; ist eingestellt
G0 X0 Y10                       ; wird nicht interpolierend verfahren
G0 G43 X20 Y20                  ; es wird im Bahnbetrieb (interpolierend)
                                ; verfahren
G0 G64 X30 Y30                  ; es wird im Bahnbetrieb (interpolierend)
                                ; verfahren
G0 G95 X100 Z100 m3 s100        ; es wird nicht interpolierend verfahren
                                ; kein Umdrehungsvorschub aktiv
```

11.10 Datenlisten

11.10.1 Maschinendaten

11.10.1.1 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
20730	G0_LINEAR_MODE	Interpolationsverhalten bei G0
20732	EXTERN_G0_LINEAR_MODE	Interpolationsverhalten bei G00
22240	AUXFU_F_SYNC_TYPE	Ausgabezeitpunkt der F-Funktionen

11.10.1.2 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: $MA_	Beschreibung
30450	IS_CONCURRENT_POS_AX	Konkurrierende Positionierachse
30460	BASE_FUNCTION_MASK	Achsfunktionen
32060	POS_AX_VELO	Vorschub für Positionierachse
37510	AX_ESR_DELAY_TIME1	Verzögerungszeit ESR-Einzelachse
37511	AX_ESR_DELAY_TIME2	ESR-Zeit für interpolatorisches Bremsen Einzelachse

11.10.2 Settingdaten

11.10.2.1 Achs-/Spindel-spezifische Settingdaten

Nummer	Bezeichner: $SA_	Beschreibung
43600	IPOBRAKE_BLOCK_EXCHANGE	Satzwechselkriterium Bremsrampe
43610	ADISPOSA_VALUE	Toleranzfenster Bremsrampe

11.10.3 Signale

11.10.3.1 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Vorschubsperre	DB21, ...DBX6.0	DB3200.DBX6.0
NC-Start	DB21, ...DBX7.1	DB3200.DBX7.1

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
NC-Stop Achsen plus Spindel	DB21, ...DBX7.4	DB3200.DBX7.4
Reset	DB21, ...DBX7.7	-

11.10.3.2 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Alle Achsen stehen	DB21, ...DBX36.3	DB3300.DBX4.3
Fahrbefehl minus	DB21, ...DBX40.6	DB3300.DBX1000.6
Fahrbefehl plus	DB21, ...DBX40.7	DB3300.DBX1000.7

11.10.3.3 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Vorschubkorrektur achsspezifisch	DB31, ...DBB0	DB380x.DBB0
Reglerfreigabe	DB31, ...DBX2.1	DB380x.DBX2.1
Restweg löschen Spindel-Reset achsspezifisch	DB31, ...DBX2.2	DB380x.DBX2.2
AXRESET	DB31, ...DBX28.1	-
AXRESUME	DB31, ...DBX28.2	-
AxStop, Halt	DB31, ...DBX28.6	-
PLC kontrolliert Achse	DB31, ...DBX28.7	-

11.10.3.4 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Genauhalt grob	DB31, ...DBX60.6	DB390x.DBX0.6
Genauhalt fein	DB31, ...DBX60.7	DB390x.DBX0.7
Axialer Alarm	DB31, ...DBX61.1	DB390x.DBX1.1
Achse betriebsbereit (AX_IS_READY)	DB31, ...DBX61.2	DB390x.DBX1.2
Achscontainer Rotation aktiv	DB31, ...DBX62.7	-
AXRESET DONE	DB31, ...DBX63.0	-
PLC kontrolliert Achse	DB31, ...DBX63.1	-
Achsstop aktiv	DB31, ...DBX63.2	DB390x.DBX3.2
Fahrbefehl minus	DB31, ...DBX64.6	DB390x.DBX4.6
Fahrbefehl plus	DB31, ...DBX64.7	DB390x.DBX4.7
Positionierachse	DB31, ...DBX76.5	DB390x.DBX1002.5
F-Funktion (Vorschub) für Positionierachse	DB31, ...DBB78-81	-
Notrückzug aktiv	DB31, ...DBX98.7	DB390x.DBX5002.7

P5: Pendeln - nur 840D sl

12

12.1 Kurzbeschreibung

Definition

Bei der Funktion Pendeln pendelt eine Pendelachse mit dem programmierten Vorschub oder einem abgeleiteten Vorschub (Umdrehungsvorschub) zwischen zwei Umkehrpunkten hin und her. Es können mehrere Pendelachsen aktiv sein.

Ausprägungsformen des Pendelns

Die Pendelfunktionen können nach dem Verhalten an den Umkehrpunkten und bezüglich der Zustellung klassifiziert werden:

- Asynchrones Pendeln über Satzgrenzen hinweg.

  Während der Pendelbewegung können andere Achsen beliebig interpolieren. Die Pendelachse kann Eingangsachse für die dynamische Transformation oder Führungsachse bei Gantry- und Mitschleppachsen sein. Das Pendeln ist nicht automatisch an die Betriebsart Automatik geknüpft.

- Pendeln mit kontinuierlicher Zustellung.

  Die Zustellung ist in mehreren Achsen gleichzeitig möglich. Es besteht jedoch kein interpolatorischer Zusammenhang zwischen Zustellbewegung und Pendelbewegung.

- Pendeln mit Zustellung in beiden Umkehrpunkten oder nur im linken oder rechten Umkehrpunkt. Die Zustellung kann eine programmierbare Distanz vor dem Umkehrpunkt eingeleitet werden.
- Ausfeuerhübe im Anschluss an Pendeln.
- Beginn und Ende des Pendelns an definierten Positionen.

Verhalten an den Umkehrpunkten

Der Richtungswechsel wird eingeleitet:

- ohne Erreichen der Genauhaltgrenze (Genauhalt fein oder grob)
- nach Erreichen der programmierten Position oder
- nach Erreichen der programmierten Position und Ablauf einer Verweilzeit.
- durch externes Signal (von der PLC).

Beeinflussungen

Pendelbewegungen können auf mehrfache Art beeinflusst werden:

- Die Pendelbewegung und/oder Zustellung kann durch Restweglöschen unterbrochen werden.
- Über NC-Programm, PLC, HMI, Handrad oder Richtungstasten können die Umkehrpunkte verändert werden.
- Die Vorschubgeschwindigkeit der Pendelachse kann über Werteingabe im NC-Programm, PLC, HMI oder über Override verändert werden. Der Vorschub kann programmiert von einer Masterspindel, Rundachse oder Spindel abhängig gemacht werden (Umdrehungsvorschub).

  Literatur:

  /FB1/ Funktionshandbuch Grundfunktionen; Vorschübe (V1)
- Die Pendelbewegung kann vollständig durch PLC gesteuert werden.

Arten der Steuerung des Pendelns

Beim Pendeln wird unterschieden zwischen:

1. Asynchronem Pendeln:

   Ist über Satzgrenzen hinweg aktiv und kann auch von PLC/HMI aus gestartet werden.
2. Pendeln über Bewegungssynchronaktionen:

   Dabei werden das asynchrone Pendeln und eine Zustellbewegung über Synchronaktionen miteinander gekoppelt. So kann ein Pendeln mit Zustellung in den Umkehrpunkten programmiert werden, das satzweise aktiv ist.

12.2 Asynchrones Pendeln

Eigenschaften

Asynchrones Pendeln hat die folgenden Eigenschaften:

- Die Pendelachse fährt zwischen den Umkehrpunkten 1 und 2 mit dem gewählten Vorschub hin und her bis die Pendelbewegung abgeschaltet wird oder bis einer Randbedingung entsprechend reagiert wird. Steht die Pendelachse beim Starten nicht auf dem Umkehrpunkt 1, so wird zuerst dieser angefahren.
- Für die Pendelachse ist Linearinterpolation G01 aktiv unabhängig vom im Programm aktuell gültigen G-Code. Alternativ kann Umdrehungsvorschub G95 aktiviert werden.
- Das asynchrone Pendeln ist achsspezifisch über Satzgrenzen hinweg wirksam.
- Es können gleichzeitig mehrere Pendelachsen aktiv sein (maximal Anzahl der Positionierachsen).
- Während der Pendelbewegung können andere Achsen als die Pendelachse beliebig interpoliert werden. Über eine Bahnbewegung oder mit einer Positionierachse kann eine kontinuierliche Zustellung erreicht werden. Dabei besteht jedoch kein interpolatorischer Zusammenhang zwischen Pendel- und Zustellbewegung.
- Falls die PLC nicht die Kontrolle über die Achse ausübt, wird die Achse beim asynchronen Pendeln wie eine normale Positionierachse behandelt. Bei PLC-Steuerung hat das PLC-Programm über die entsprechenden Stop-Bits der VDI-Nahtstelle dafür zu sorgen, dass auf die VDI-Signale wie erwünscht reagiert wird. Zu diesen VDI-Signalen zählt auch Programmende, Betriebsartenwechsel und Einzelsatz.
- Die Pendelachse kann Eingangsachse für Transformationen sein (z. B. schräge Achse).

  Literatur:

  /FB2/ Funktionshandbuch Erweiterungsfunktionen; Transmit/Mantelflächentransformation (M1)
- Die Pendelachse kann Führungsachse bei Gantry- und Mitschleppachsen sein.

  Literatur:

  /FB3/ Funktionshandbuch Sonderfunktionen; Gantry-Achse" (G1)
- Es kann mit Ruckbegrenzung (SOFT) und/oder mit geknickter Beschleunigungskennlinie verfahren werden (wie bei Positionierachsen).
- Über das Teileprogramm ist ein satzsynchrones Einschalten der Pendelbewegung weiterhin gegeben.
- Die Pendelbewegung kann ebenfalls von PLC/HMI aus gestartet, beeinflusst und gestoppt werden.
- Ein interpolatorisches Pendeln ist nicht möglich (z. B. Schräges Pendeln).

12.2.1 Einflüsse auf das asynchrone Pendeln

Settingdaten

Für das Pendeln erforderliche Settingdaten können mit speziellen Sprachbefehlen per Teileprogramm im NCK, über HMI und/oder über PLC eingestellt werden.

Vorschubgeschwindigkeit

Die Vorschubgeschwindigkeit für die Pendelachse wird wie folgt gewählt bzw. vorgegeben:

- Als Vorschubgeschwindigkeit wird die Geschwindigkeit verwendet, die für die Achse als Positionierachse definiert ist. Diese ist über FA[Achse] programmierbar und modal wirksam. Ist keine Geschwindigkeit programmiert, so wird der in Maschinendatum POS_AX_VELO hinterlegte Wert verwendet (siehe Positionierachsen).
- Bei einer laufenden Pendelbewegung kann die Vorschubgeschwindigkeit der Pendelachse über Settingdatum geändert werden. Dabei ist über Teileprogramm und über Settingdatum einstellbar, ob die Änderung sofort oder erst ab dem nächsten Umkehrpunkt wirken soll.
- Die Vorschubgeschwindigkeit ist über den Override (axiales VDI-Signal und programmierbar) beeinflussbar.
- Wenn Probelauf aktiv ist, wird mit dem Probelaufvorschub gearbeitet, wenn dieser größer als die momentan programmierte Geschwindigkeit ist.

  Die Wirksamkeit des in SD42100 $SC_DRY_RUN_FEED eingetragenen Probelaufvorschubs kann mit SD42101 $SC_DRY_RUN_FEED_MODE gesteuert werden. Siehe

  Literatur:

  /FB1/ Funktionshandbuch Grundfunktionen; Vorschübe (V1)
- Geschwindigkeitsüberlagerung/Wegüberlagerung können durch Handrad beeinflusst werden.

  Siehe auch Tabelle "Haltezeitauswirkung".

  Literatur:

  /FB2/ Funktionshandbuch Erweiterungsfunktionen; Handfahren und Handradfahren (H1)
- Die Pendelachse kann mit Umdrehungsvorschub verfahren werden.

Umdrehungsvorschub

Der Umdrehungsvorschub kann auch für Pendelachsen genutzt werden.

Umkehrpunkte

Die Positionen der Umkehrpunkte können vor oder während einer bereits aktiven Pendelbewegung über Settingdaten eingegeben werden.

- Während der Pendelbewegung kann die Position eines Umkehrpunkts über Handverfahren (Handrad, JOG-Tasten) verändert werden. Dies ist unabhängig davon, ob die Pendelbewegung unterbrochen wurde oder nicht.

Für die Veränderung der Position eines Umkehrpunkts gilt: Bei einer bereits aktiven Pendelbewegung wird eine geänderte Position eines Umkehrpunkts erst wirksam, wenn dieser Umkehrpunkt erneut angefahren wird. Wird der Umkehrpunkt bereits angefahren, so wirkt die Korrektur im nächsten Pendelhub.

Hinweis

Ist das VDI-Nahtstellensignal *DRF aktivieren* aktiv und soll gleichzeitig ein Umkehrpunkt geändert werden, so wirken die Signale des Handrads sowohl für die DRF-Verschiebung als auch für die Verschiebung des Umkehrpunkts, d. h. der Umkehrpunkt wird absolut um die doppelte Strecke verschoben.

Haltezeiten

Pro Umkehrpunkt kann eine Haltezeit über Settingdatum programmiert werden.

In nachfolgenden Sätzen des NC-Programms kann die Einstellung verändert werden. Sie ist dann satzsynchron ab dem nächsten entsprechenden Umkehrpunkt wirksam.

Über Settingdatum kann die Haltezeit asynchron geändert werden. Sie ist dann ab dem nächsten Durchfahren des entsprechenden Umkehrpunktes wirksam.

Die folgende Tabelle erläutert das Bewegungsverhalten im Genauhaltbereich bzw. am Umkehrpunkt abhängig von einer Haltezeitvorgabe.

Tabelle 12- 1 Haltezeitauswirkung

Wert Haltezeit	Verhalten
-2	Interpolation wird ohne Warten auf Genauhalt fortgesetzt
-1	Im Umkehrpunkt wird auf Genauhalt grob gewartet
0	Im Umkehrpunkt wird auf Genauhalt fein gewartet
>0	Im Umkehrpunkt wird auf Genauhalt fein gewartet und anschließend wird die Haltezeit abgewartet.

Pendeln ausschalten

Beim Ausschalten des Pendelns ist für das Beenden der Pendelbewegung eine der folgenden Möglichkeiten einstellbar:

- die Pendelbewegung wird im nächsten Umkehrpunkt beendet
- die Pendelbewegung wird im Umkehrpunkt 1 beendet
- die Pendelbewegung wird im Umkehrpunkt 2 beendet

Im Anschluss daran werden ggf. Ausfeuerungshübe abgearbeitet und ggf. eine Endposition angefahren.

Beim Wechseln von der asynchronen Pendelbewegung zum Ausfeuern und während des Ausfeuerns entspricht das Verhalten im Umkehrpunkt bezüglich Genauhalt dem durch die Haltezeit des entsprechenden Umkehrpunkts vorgegebenen Verhalten. Ein Ausfeuerungshub ist die Bewegung zum anderen Umkehrpunkt und zurück. Siehe Tabelle:

Hinweis

Pendeln mit Bewegungssynchronaktionen und Haltezeiten "OST1/OST2"

Nach Ablauf der eingestellten Haltezeiten findet beim Pendeln der interne Satzwechsel statt (sichtbar an den neuen Restwegen der Achsen). Beim Satzwechsel wird die Ausschaltfunktion überprüft. Dabei wird nach der eingestellten Steuereinstellung für den Bewegungsablauf "OSCTRL" die Ausschaltfunktion festgelegt.

Dieses Zeitverhalten ist durch den Vorschuboverride beeinflussbar.

Unter Umständen wird danach noch ein Pendelhub ausgeführt, bevor die Ausfeuerungshübe gestartet oder die Endposition angefahren wird.

Es entsteht dabei der Eindruck, es verändert sich das Ausschaltverhalten. Dem ist jedoch nicht so.

Tabelle 12- 2 Abläufe bei Pendeln ausschalten

Funktion	Vorgaben	Erläuterung
Abschalten im definierten Umkehrpunkt	Anzahl der Ausfeuerungshübe ist gleich 0 keine Endposition aktiv	Die Pendelbewegung wird im entsprechenden Umkehrpunkt gestoppt.
Abschalten mit spezifizierter Anzahl von Ausfeuerungshüben	Anzahl der Ausfeuerungshübe ist ungleich 0 keine Endposition aktiv	Vom entsprechenden Umkehrpunkt an wird noch die im Befehl angegebene Anzahl von Ausfeuerungshüben abgearbeitet.
Abschalten mit Ausfeuerungshüben und definierter Endposition (beliebig)	Anzahl der Ausfeuerungshübe ist ungleich 0 Endposition aktiv	Vom entsprechenden Umkehrpunkt an wird noch die im Befehl angegebene Anzahl von Ausfeuerungshüben abgearbeitet und anschließend wird die angegebene Endposition angefahren.
Abschalten ohne Ausfeuerungshübe aber mit definierter Endposition (beliebig)	Anzahl der Ausfeuerungshübe ist gleich 0 Endposition aktiv	Vom entsprechenden Umkehrpunkt an wird die angegebene Endposition angefahren.

NC-Sprache

Die NC-Programmiersprache erlaubt es, das asynchrone Pendeln vom Teileprogramm her zu beeinflussen. Über die folgenden Funktionen ist ein der Abarbeitung des NC-Programms entsprechendes Einschalten und Beeinflussen des asynchronen Pendelns möglich.

Hinweis

Werden im Teileprogramm die Settingdaten direkt beschrieben, so wird die Änderung bezüglich des Abarbeitens des Teileprogramms zu früh (zum Vorlaufzeitpunkt) wirksam. Über einen Vorlaufstop (`STOPRE`) kann wieder ein synchrones Verhalten erreicht werden.

Literatur:

/PA/ Programmieranleitung

1) Pendeln einschalten, ausschalten:

- OS[Pendelachse] = 1; Für Pendelachse Pendeln einschalten
- OS[Pendelachse] = 0; Für Pendelachse Pendeln ausschalten

Hinweis

Jede Achse kann als Pendelachse benutzt werden.

2) Pendeln fertig:

- WAITP(Pendelachse)

  Positionierachsbefehl - hält Satz bis Pendelachse in Feinstop und synchronisiert den Vor- und Hauptlauf. Die Pendelachse wird wieder als Positionierachse eingetragen und kann anschließend wieder normal verwendet werden.

  Soll mit einer Achse gependelt werden, so ist diese mit einem WAITP(Achse) Aufruf vorher für das Pendeln freizugeben.

  Dies gilt auch, wenn das Pendeln von PLC/HMI aus angestoßen werden soll. Dann ist der WAITP(Achse) Aufruf auch erforderlich, wenn die Achse vorher über NC-Programm programmiert wurde. Mit Softwarestand 3.2 ist über Maschinendatum $MA_AUTO_GET_TYPE wählbar, ob WAITP() weiterhin programmiert oder automatisiert durchgeführt werden soll.

Hinweis

Bei WAITP wird solange gewartet bis die Pendelbewegung beendet ist. Dies kann z. B. durch ein vorher programmiertes Ausschalten im NC-Programm angestoßen werden oder über die PLC bzw. HMI oder durch Restweglöschen.

3) Umkehrpunkte setzen:

- OSP1[Pendelachse] = Position des Umkehrpunkts 1
- OSP2[Pendelachse] = Position des Umkehrpunkts 2

Eine Position wird satzsynchron im Hauptlauf in das entsprechende Settingdatum eingetragen und ist damit bis zur nächsten Änderung des Settingdatums wirksam.

Falls ein inkrementelles Verfahren aktiv ist, so wird die Position inkrementell zur letzten im NC-Programm programmierten entsprechenden Umkehrposition berechnet.

4) Haltezeiten in den Umkehrpunkten:

- OST1[Pendelachse] = Haltezeit im Umkehrpunkt 1 in [s]
- OST2[Pendelachse] = Haltezeit im Umkehrpunkt 2 in [s]

Eine Haltezeit wird satzsynchron im Hauptlauf in das entsprechende Settingdatum eingetragen und ist damit bis zur nächsten Änderung des Settingdatums wirksam.

Die Einheit für die Haltezeit ist identisch der Einheit für die mit G04 programmierte Haltezeit.

5) Vorschub einstellen:

- FA[Achse] = FWert

  Vorschub Positionierachse.

Der Vorschub wird im Hauptlauf satzsynchron in das entsprechende Settingdatum übertragen. Wird die Pendelachse mit Umdrehungsvorschub verfahren, sind die entsprechenden Abhängigkeiten, wie in Funktionsbeschreibung V1 beschrieben, anzugeben.

6) Steuereinstellungen für den Bewegungsablauf setzen:

- OSCTRL[Pendelachse] = (Setz-Optionen, Rücksetz-Optionen)

  Die Setzoptionen sind wie folgt definiert (die Rücksetzoptionen wählen die Einstellungen ab):

Tabelle 12- 3 Setz-/Rücksetzoptionen

Optionswert	Bedeutung
0	beim Abschalten der Pendelbewegung im nächsten Umkehrpunkt stoppen (Default). Kann nur durch Rücksetzen der Optionswerte 1 und 2 erreicht werden.
1	beim Abschalten der Pendelbewegung im Umkehrpunkt 1 stoppen
2	beim Abschalten der Pendelbewegung im Umkehrpunkt 2 stoppen
3	beim Abschalten der Pendelbewegung keinen Umkehrpunkt anfahren, falls keine Ausfeuerungshübe programmiert sind
4	nach dem Ausfeuern soll eine Endposition angefahren werden
8	wird die Pendelbewegung durch Restweglöschen abgebrochen, so sollen anschließend die Ausfeuerungshübe abgearbeitet werden und ggf. die Endposition angefahren werden
16	wird die Pendelbewegung durch Restweglöschen abgebrochen, so soll wie beim Abschalten der Pendelbewegung die entsprechende Umkehrposition angefahren werden
32	geänderter Vorschub ist erst ab nächsten Umkehrpunkt wirksam
64	falls der Vorschub 0 ist, ist Wegüberlagerung aktiv, andernfalls ist Geschwindigkeitsüberlagerung aktiv
128	bei Rundachse DC (kürzester Weg)
256	Ausfeuerungshub als Einfachhub
512	Zuerst Startposition anfahren

Hinweis

Die Optionswerte 0 - 3 verschlüsseln das Verhalten an den Umkehrpunkten beim Ausschalten. Es kann eine der Alternativen 0 - 3 ausgewählt werden. Die übrigen Einstellungen sind nach Bedarf kombinierbar mit der gewählten Alternative. Mehrer Optionen werden durch Zeichen + aneinander gefügt.

Beispiel: Die Pendelbewegung für die Achse Z soll beim Abschalten im Umkehrpunkt 1 stoppen, es soll eine Endposition angefahren werden, ein geänderter Vorschub soll nun sofort wirksam werden und nach Restweglöschen soll die Achse sofort stoppen.

OSCTRL[Z] = (1+4, 16+32+64)

Die Setz- und Rücksetzoptionen werden satzsynchron im Hauptlauf in das entsprechende Settingdatum übertragen und sind damit bis zur nächsten Änderung des Settingdatums wirksam.

Hinweis

Die Steuerung wertet zuerst die Rücksetzoptionen aus, darauf die Setzoptionen.

7) Ausfeuerungshübe:

- OSNSC[Pendelachse] = Anzahl der Ausfeuerungshübe

Die Anzahl der Ausfeuerungshübe wird satzsynchron im Hauptlauf in das entsprechende Settingdatum eingetragen und ist damit bis zur nächsten Änderung des Settingdatums wirksam.

8) Endposition, die nach Ausschalten des Pendelns angefahren werden soll:

- OSE[Pendelachse] = Endposition der Pendelachse

Die Endposition wird im Hauptlauf satzsynchron in das entsprechende Settingdatum eingetragen und ist bis zu dessen nächster Änderung wirksam. Implizit wird Optionswert 4 gemäß Tabelle 7–3 gesetzt, damit die gesetzte Endposition angefahren wird.

9) Anfangsposition, die vor Einschalten des Pendelns angefahren werden soll:

- OSB [Pendelachse] = Anfangsposition der Pendelachse

Die Startposition wird im Hauptlauf satzsynchron in das entsprechende Settingdatum SD43790 $SA_OSCILL_START_POS eingetragen und ist bis zu dessen nächster Änderung wirksam. Damit die Startposition angefahren wird, muss im Settingdatum SD43770 $SA_OSCILL_CTRL_MASK Bit 9 gesetzt sein. Die Startposition wird **vor Umkehrpunkt 1** angefahren. Stimmt die Startposition mit der Umkehrposition 1 überein, so wird als nächstes die Umkehrposition 2 angefahren.

Alternativ zum Programmierbefehl OSB kann die Startposition auch direkt in das Settingdatum SD43790 $SA_OSCILL_START_POS eingetragen werden.

Alle Positionsangaben in den Settingdaten und Systemvariablen sind Angaben im Basiskoordinatensystem (BKS). Die Positionsangabe bei OSB, OSE erfolgt im Werkstück-Koordinatensystem (WKS).

Beim Erreichen der Startposition wirkt keine Haltezeit, auch wenn die Startposition mit der Umkehrposition 1 übereinstimmt, stattdessen wird auf Genauhalt fein gewartet. Eine eingestellte Genauhaltbedingung wird eingehalten.

Soll beim satzbezogenen Pendeln keine Zustellbewegung stattfinden, wenn die Startposition mit der Umkehrposition 1 übereinstimmt, so kann dies mit einer weiteren Synchronaktion bestimmt werden, siehe Beispiele im Kap. "Satzbezogenes Pendeln (Startposition = Umkehrpunkt 1)".

Programmierbeispiel

Im Kap. "Beispiele" befindet sich ein Beispiel, das alle für asynchrones Pendeln wesentlichen Elemente im Zusammenhang enthält.

12.2.2 Asynchrones Pendeln unter PLC-Regie

Aktivierung

Die Anwahl der Funktion ist von der PLC aus über das Settingdatum OSCILL_IS_ACTIVE in allen Betriebsarten außer MDA-Ref und JOG-Ref möglich.

Einstellungen

Das Ein- und Ausschalten der Pendelbewegung, die Positionen der Umkehrpunkte, die Haltezeiten in den Umkehrpunkten, die Vorschubgeschwindigkeit, die Optionen in den Umkehrpunkten, die Anzahl Ausfeuerungshübe und die Endposition nach dem Ausschalten sind von der PLC über Settingdaten steuerbar. Diese Werte können jedoch auch vorab als Settingdatum über die HMI-Bedienoberfläche direkt oder über ein NC-Programm gesetzt werden. Diese Einstellungen bleiben über Power On hinweg erhalten und die PLC kann eine so eingestellte Pendelbewegung auch direkt über das Settingdatum OSCILL_IS_ACTIVE starten (über Variablendienst).

Randbedingungen

Eine Spindel, die als Achse eine über PLC gestartete Pendelbewegung ausführen soll, muss die Voraussetzungen erfüllen, die notwendig sind, um auch als Positionierachse verfahren zu können, d. h. die Spindel muss z. B. vorher in eine lagegeregelte Spindel (`SPOS`) überführt werden.

Auf die beiden Halt-Bits der VDI-Nahtstelle NST DB31, ... DBX28.6 ("Halt") und NST DB31, ... DBX28.5 ("Halt im nächsten Umkehrpunkt") reagieren die Achsen immer unabhängig davon, ob die Achse von der PLC kontrolliert wird oder nicht.

12.2.3 Besondere Reaktionen beim asynchronen Pendel

Mit PLC-Steuerung

Das PLC-Programm kann über Signale der VDI-Nahtstelle die Kontrolle über eine Pendelachse übernehmen. Zu den VDI-Signalen zählen auch Programmende, Betriebsartenwechsel und Einzelsatz.

Bis SW 6.2 werden folgende VDI-Nahtstellensignale ignoriert: Vorschub-/Spindel-Halt und NC-STOP; beim Restweglöschen wird die daraus resultierende Bremsanforderung unterdrückt.

Ab SW 6.3 reagiert eine vom Hauptlauf interpolierende asynchrone Pendelachse unabhängig vom NC-Programm bezüglich NC-STOP, Alarmbehandlung, Programmende, Programmbeeinflussungen und RESET.

Die Kontrolle über die Achse/Spindel erfolgt über die axiale VDI-Nahtstelle (PLC→NCK) mit den NST DB31, ... DBX28.7 ("PLC kontrolliert Achse") == 1

Weitere Informationen zu Achsen mit PLC-Steuerung entnehmen Sie bitte:

Literatur:

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)

Ohne PLC-Steuerung

Falls die PLC nicht die Kontrolle über die Achse ausübt, wird die Achse beim asynchronen Pendeln wie eine normale Positionierachse (POSA) behandelt.

Restweg löschen

Kanalspezifisches Restweg löschen wird ignoriert. Axiales Restweg löschen:

Ohne PLC-Steuerung

Falls die Pendelachse nicht von der PLC kontrolliert wird, wird mit Bremsrampe gestoppt.

Mit PLC-Steuerung

wird das Bremsen unterdrückt, dies muss von der PLC ausgelöst werden.

Für **beide** Fälle gilt: Nachdem die Achse gestoppt wurde, wird ggf. der entsprechende Umkehrpunkt angefahren (siehe OSCILL_CONTROL_MASK, Kapitel 4) und der Restweg gelöscht. Anschließend werden die Ausfeuerungshübe ausgeführt und die Endposition angefahren, falls dies in OSCILL_CONTROL_MASK so eingestellt wurde.

Die Pendelbewegung ist somit beendet.

Hinweis

Beim Schleifen kann die Messzange über das axiale Restweg löschen eingreifen.

NOTAUS

Bei NOTAUS wird die Achse vom Servo abgebremst (über Wegnahme der Reglerfreigabe und Nachführen). Die Pendelbewegung ist beendet und muss neu gestartet werden.

Reset

Die Pendelbewegung wird mit Bremsrampe abgebrochen und abgewählt. Nachfolgend gewählte Optionen werden nicht abgearbeitet (Ausfeuerhübe, Endpunkt anfahren).

Arbeitsfeldbegrenzung, Endschalter

Wird bei der Vorbearbeitung festgestellt, dass die Pendelbewegung eine aktive Begrenzung überschreitet, so wird ein Alarm gemeldet und das Pendeln wird nicht gestartet. Überfährt die Pendelachse eine zwischenzeitlich aktivierte Begrenzung (z. B. 2. Softwareendschalter), so wird mit Bremsrampe abgebremst und ein Alarm gemeldet.

⚠ VORSICHT
Schutzbereiche wirken nicht!

Nachführbetrieb

Kein Unterschied zu Positionierachsen.

Programmende

Wird die Achse nicht von der PLC kontrolliert, so wird das Programmende erst erreicht, wenn die Pendelbewegung beendet ist (Verhalten wie POSA:

Positionieren über Satzgrenzen hinweg).

Wird die Achse von der PLC kontrolliert, so pendelt sie über Programmende hinweg weiter.

Betriebsartenwechsel

Die folgende Tabelle zeigt, bei welchen Betriebsarten Pendeln möglich ist. Die Pendelbewegung wird bei Betriebsartenwechsel in eine Betriebsart, die Pendeln erlaubt, nicht beeinflusst. Unerlaubte Betriebsartenwechsel werden mit Alarm abgelehnt. Das gleichzeitige Verfahren als Pendelachse und über NC-Programm oder über Bedienung (JOG) ist über einen Alarm abgesichert. Es gilt: die zuerst gestartete Bewegungsart gewinnt.

Tabelle 12- 4 Betriebsarten, die Pendeln erlauben

Betriebsart	Pendeln möglich
AUTO	ja
MDA	ja
MDA Repos	ja
MDA Teachin	ja
MDA Ref	nein
JOG	ja
JOG Ref	nein
JOG Repos	ja

Einzelsatzbearbeitung

Wird die Achse nicht von der PLC kontrolliert, so verhält sie sich bei Einzelsatz wie eine Positionierachse (POSA), d. h. sie setzt die Bewegung fort.

Override

Der Override ist vorgegeben durch:

VDI Nahtstelle

Auf die Pendelachse wirkt der axiale Override.

Programmierung

Der Override ist wie bei Positionierachsen wirksam.

Satzsuchlauf

Bei Satzsuchlauf wird die zuletzt gültige Pendelfunktion registriert und einem Maschinendatum OSCILL_MODE_MASK entsprechend sofort nach NC-Start (bei Anfahren der Anfahrposition nach Satzsuchlauf) oder nach Erreichen der Anfahrposition nach Satzsuchlauf aktiviert (Vorbesetzung).

OSCILL_MODE_MASK Bit 0:

0: Nach Erreichen der Anfahrposition beginnt das Pendeln.

1: Nach NC-Start beginnt das Pendeln sofort.

REORG

Es wird immer zuerst der Umkehrpunkt 1 angefahren und dann wieder gependelt.

ASUP

Während eines ASUP's (asynchronen Unterprogramms) wird die Pendelbewegung fortgesetzt.

12.3 Pendeln über Synchronaktionen gesteuert

Prinzip

Eine asynchrone Pendelbewegung wird über Synchronaktionen mit einer Zustellbewegung gekoppelt und so entsprechend beeinflusst.

Literatur:

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Synchronaktionen (S5)

Im Folgenden wird nur der Aspekt der Bewegungssynchronen Aktionen betrachtet, der mit Pendeln im Zusammenhang steht.

Funktionen

Mit den im Folgenden detailliert beschriebenen Sprachmitteln können die folgenden Funktionskomplexe realisiert werden:

1. Zustellung im Umkehrpunkt (s. Kap. "Zustellung im Umkehrpunkt 1 oder 2").
2. Zustellung im Umkehrbereich (s. Kap. "Zustellung im Umkehrbereich").
3. Zustellung in beiden Umkehrpunkten (s. Kap. "Zustellung in beiden Umkehrpunkten").
4. Anhalten der Pendelbewegung im Umkehrpunkt bis die Zustellung erfolgt ist (s. Kap. "Anhalten der Pendelbewegung im Umkehrpunkt").
5. Pendelbewegung freigeben (s. Kap. "Pendelbewegung wieder starten").
6. Teilzustellung nicht zu früh starten (s. Kap. "Teilzustellung nicht zu früh starten").

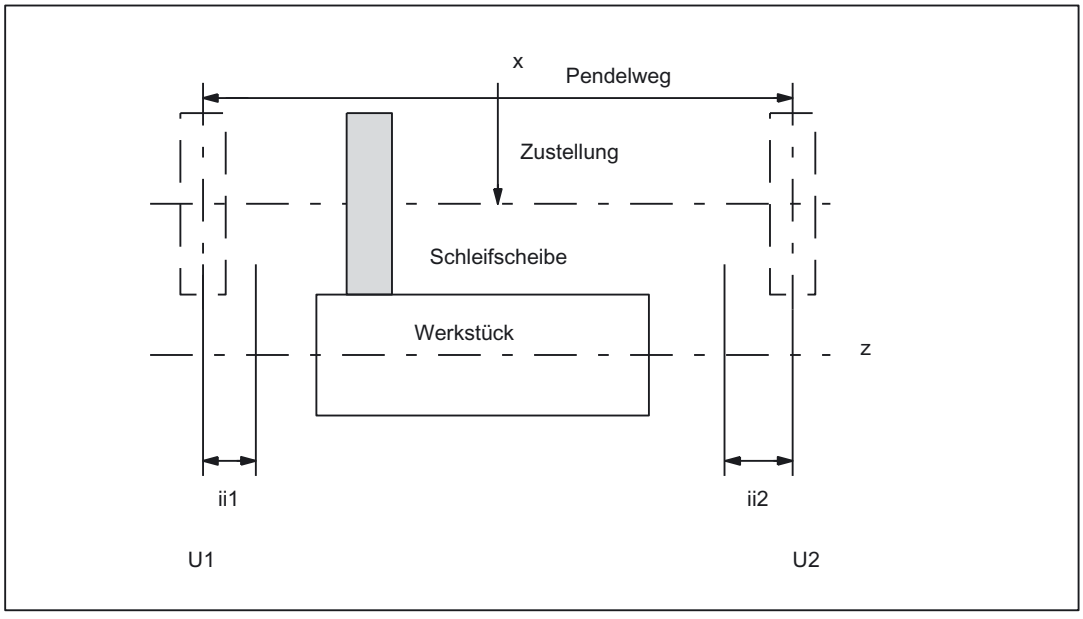


Bild 12-1 Anordnung Pendelachse, Zustellachse und Begriffe

Legende:

U1: Umkehrpunkt 1

U2: Umkehrpunkt 2

ii1: Umkehrbereich 1

ii2: Umkehrbereich 2

Programmierung

Vor dem Bewegungssatz, der die Zuordnung von Zustellachse und Pendelachse (s. Kap. "Zuordnung von Pendel- und Zustellachse OSCILL"), die Festlegung der Zustellung (POSP) und die Bewegungssynchronaktionen enthält, sind zunächst die Parameter für das Pendeln (s. Kap. "Einflüsse auf das asynchrone Pendeln") festzulegen:

Über ein WAITP [Pendelachse] (s. MD30552 $MA_AUTO_GET_TYPE) wird die Achse für das Pendeln freigegeben und damit kann gleichzeitig eine Übernahme der Pendelparameter in das System, d. h. in die Settingdaten erreicht werden. Dann können für die Programmierung der Bewegungssynchronaktionen die symbolischen Namen, z. B. SA43700 $SA_REVERSE_POS1 verwendet werden.

Hinweis

Für Bewegungssynchronaktionen mit $SA_REVERSE_POS Werten gelten die Vergleichswerte zum **Interpretationszeitpunkt**. Werden Settingdaten nachträglich verändert, so hat das keinen Einfluss.

Für Bewegungssynchronaktionen mit $$AA_REVERSE_POS Werten gelten die Vergleichswerte innerhalb der **Interpolation**. Somit ist ein Reagieren auf geänderte Umkehrpositionen gewährleistet.

- **Bewegungssynchronbedingungen WHEN, WHENEVER**
- Aktivierung durch Bewegungssatz
  - Pendelachse und Zustellachsen einander zuordnen OSCILL
  - Zustellverhalten angeben POSP.

In den folgenden Abschnitten werden die noch nicht vorgestellten Elemente näher erläutert.

Einige Beispiele sind im Kapitel "Beispiele" beschrieben.

Hinweis

Ist die Bedingungen, die mit der Bewegungssynchronaktion (WHEN und WHENEVER) festgelegt wurde, nicht mehr gültig, so wird als Ergebnis, falls der OVERRIDE auf 0% gesetzt war, für diese Bedingung der OVERRIDE **automatisch** auf 100% gesetzt.

Hauptlauf-Auswertung

Es besteht die Möglichkeit, dass die Synchronisationsbedingungen im Interpolationstakt im Hauptlauf mit aktuellen Istwerten verglichen werden ($$-Variable auf der rechten Seite von Vergleichsbedingungen). Beim normalen Systemvariablenvergleich werden die Ausdrücke im Vorlauf ausgewertet. Die vollständigen, erweiterten Möglichkeiten für Synchronaktionen finden Sie in:

Literatur:

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Bewegungssynchronaktionen (S5)

Beispiel 1

Pendeln, Umkehrposition fest über Settingdatum eingestellt:

```
$SA_OSCILL_REVERSE_POS1[Z]=-10
$SA_OSCILL_REVERSE_POS2[Z]=10
G0 X0 Z0
WAITP(Z)
ID=1 WHENEVER $AA_IM[Z] <
$SA_OSCILL_REVERSE_POS1[Z] DO
$AA_OVR[X]=0
ID=2 WHENEVER $AA_IM[Z] >
$SA_OSCILL_REVERSE_POS2[Z] DO
$AA_OVR[X]=0
                                    ; Wenn der Istwert der Pendelachse den
                                    ; Umkehrpunkt überschritten hat,
                                    ; wird Zustellachse angehalten.
OS[Z]=1 FA[X]=1000 POS[X]=40        ; Pendeln einschalten
OS[Z]=0                             ; Pendeln ausschalten
M30
```

Beispiel 2

Pendel mit Online-Änderung der Umkehr-Position, d. h. Änderungen der Umkehrposition 1 über die Bedienoberfläche werden bei aktiver Pendelbewegung sofort berücksichtigt:

```
$SA_OSCILL_REVERSE_POS1[Z]=-10
$SA_OSCILL_REVERSE_POS2[Z]=10
G0 X0 Z0
WAITP(Z)
ID=1 WHENEVER $AA_IM[Z] <
$$SA_OSCILL_REVERSE_POS1[Z] DO
$AA_OVR[X]=0
ID=2 WHENEVER $AA_IM[Z] >
$$SA_OSCILL_REVERSE_POS2[Z] DO
$AA_OVR[X]=0
                                    ; Wenn der Istwert der Pendelachse den
                                    ; Umkehrpunkt überschritten hat, wird
                                    ; Zustellachse angehalten.
OS[Z]=1 FA[X]=1000 POS[X]=40        ; Pendeln einschalten
OS[Z]=0                             ; Pendeln ausschalten
M30
```

12.3.1 Zustellung im Umkehrpunkt 1 oder 2

Funktion

Solange die Pendelachse den Umkehrpunkt nicht erreicht hat, findet keine Bewegung der Zustellachse statt.

Anwendung

Direktes Zustellen im Umkehrpunkt

Programmierung

Für Umkehrpunkt 1:

WHENEVER $AA_IM[Z] <> $SA_OSCILL_REVERSE_POS1[Z]

DO $AA_OVR[X] = 0 $AA_OVR[Z] = 100

Für Umkehrpunkt 2:

WHENEVER $AA_IM[Z] <> $SA_OSCILL_REVERSE_POS2[Z]

DO $AA_OVR[X] = 0 $AA_OVR[Z] = 100

Erklärung der Systemvariablen:

$AA_IM[Z]: Ist-Position der Pendelachse Z im Maschinenkoordinatensystem

$SA_OSCILL_REVERSE_POS1[Z]: Position des Umkehrpunktes1 der Pendelachse

$AA_OVR[X]: axialer Override der Zustellachse

$AA_OVR[Z]: axialer Override der Pendelachse

Erklärung der Schlüsselwörter:

WHENEVER ... DO ... Immer wenn Bedingung erfüllt, dann...

Zustellung

Der Betrag der Zustellung wird durch die Anweisung POSP definiert.

Siehe Kap. "Festlegung der Zustellungen POSP".

Zuordnung

Der Zusammenhang zwischen Pendelachse und Zustellachse wird durch die Anweisung OSCILL definiert.

Siehe Kap. "Zuordnung von Pendel- und Zustellachse OSCILL".

12.3.2 Zustellung im Umkehrbereich

Funktion

Umkehrbereich 1:

Solange die Pendelachse den Umkehrbereich (Position im Umkehrpunkt 1 plus des Inhalts der Variablen ii1) nicht erreicht hat, findet keine Zustellung statt. Dies gilt unter der Voraussetzung, dass die Umkehrposition 1 kleiner als die Umkehrposition 2 ist. Ist dies nicht der Fall, so ist die Bedingung entsprechend zu ändern.

Anwendung

Umkehrbereich 1:

Diese Synchronaktion wird dazu benutzt, die Zustellbewegung zu verhindern, bis die Pendelbewegung den Umkehrbereich 1 erreicht.

Siehe Bild im Kap. "Pendeln über Synchronaktionen gesteuert".

Programmierung

Umkehrbereich 1:

WHENEVER $AA_IM[Z] > $SA_OSCILL_REVERSE_POS1[Z] + ii1

DO $AA_OVR[X] = 0

Erklärung der Systemvariablen:

$AA_IM[Z]: Ist-Position der Pendelachse Z

$SA_OSCILL_REVERSE_POS1[Z]: Position des Umkehrpunktes 1 der Pendelachse

$AA_OVR[X]: axialer Override der Zustellachse

ii1: Größe des Umkehrbereiches (Anw. Variable)

Erklärung der Schlüsselwörter:

WHENEVER ... DO ... Immer wenn Bedingung erfüllt, dann...

Funktion

Umkehrbereich 2:

Bis die aktuelle Position der Pendelachse kleiner der Position im Umkehrpunkt 2 minus dem Inhalt der Variablen ii2 ist, dann stoppe die Zustellachse. Dies gilt unter der Voraussetzung, dass die Umkehrposition 2 größer als die Umkehrposition 1 ist. Ist dies nicht der Fall, so ist die Bedingung entsprechend zu ändern.

Anwendung

Umkehrbereich 2:

Diese Synchronaktion wird dazu benutzt, die Zustellbewegung erst zu starten, wenn die Pendelbewegung den Umkehrbereich 2 erreicht.

Siehe Bild im Kap. "Pendeln über Synchronaktionen gesteuert".

Programmierung

Umkehrbereich 2:

WHENEVER $AA_IM[Z] < $SA_OSCILL_REVERSE_POS2[Z] - ii2

DO $AA_OVR[X] = 0

Erklärung:

$AA_IM[Z]: Ist-Position der Pendelachse Z

$SA_OSCILL_REVERSE_POS2[Z]: Position des Umkehrpunktes 2 der Pendelachse

$AA_OVR[X]: axialer Override der Zustellachse

ii2: Größe des Umkehrbereiches 2 (Anwender Variable)

Zustellung

Der Betrag der Zustellung wird durch die Anweisung POSP definiert.

Siehe Kapitel "Festlegung der Zustellungen POSP".

Zuordnung

Der Zusammenhang zwischen Pendelachse und Zustellachse wird durch die Anweisung OSCILL definiert.

Siehe Kapitel "Zuordnung von Pendel- und Zustellachse OSCILL".

12.3.3 Zustellung in beiden Umkehrpunkten

Prinzip

Die in den beiden vorhergehenden Abschnitten beschriebenen Funktionen für Zustellung im Umkehrpunkt und im Umkehrbereich können beliebig kombiniert werden.

Kombinationen

Zustellung:

an U1 - an U2

an U1 - Bereich U2

Bereich U1 - an U2

Bereich U1 - Bereich U2

Zustellung einseitig

an U1

an U2

Bereich U1

Bereich U2

Diese Möglichkeiten sind in Kapitel "Zustellung im Umkehrpunkt 1 oder 2" und Kapitel "Zustellung im Umkehrbereich" beschrieben.

12.3.4 Anhalten der Pendelbewegung im Umkehrpunkt

Funktion

Umkehrpunkt 1:

Immer, wenn die Pendelachse die Umkehrposition 1 erreicht, soll sie über den Override angehalten werden und die Zustellung soll gestartet werden.

Anwendung

Diese Synchronaktion wird dazu benutzt, die Pendelachse so lange anzuhalten bis die Teilzustellung erfolgt ist. Falls die Pendelachse im Umkehrpunkt 1 nicht auf das Beenden der Teilzustellung warten soll, entfällt diese Synchronaktion. Gleichzeitig kann diese Synchronaktion dazu benutzt werden, die Zustellbewegung zu starten, falls diese durch eine vorhergehende Synchronaktion, die noch wirksam ist, gestoppt wurde.

Programmierung

WHENEVER $AA_IM[Pendelachse] == $SA_OSCILL_REVERSE_POS1[Pendelachse]

DO $AA_OVR[Pendelachse] = 0 $AA_OVR[Zustellachse] = 100

Erklärung der Systemvariablen:

$AA_IM[Pendelachse]: Aktuelle Position der Pendelachse

$SA_OSCILL_REVERSE_POS1[Pendelachse]: Umkehrpunkt 1 der Pendelachse

$AA_OVR[Pendelachse]: axialer Override der Pendelachse

$AA_OVR[Zustellachse]: axialer Override der Zustellachse

Funktion

Umkehrpunkt 2:

Immer, wenn die Pendelachse die Umkehrposition 2 erreicht, soll sie über den Override 0 angehalten werden und die Zustellung soll gestartet werden.

Anwendung

Diese Synchronaktion wird dazu benutzt, die Pendelachse solange anzuhalten, bis die Teilzustellung erfolgt ist. Falls die Pendelachse im Umkehrpunkt 2 nicht auf das Beenden der Teilzustellung warten soll, entfällt diese Synchronaktion. Gleichzeitig kann diese Synchronaktion dazu benutzt werden, die Zustellbewegung zu starten, falls diese durch eine vorhergehende Synchronaktion, die noch wirksam ist, gestoppt wurde.

Programmierung

WHENEVER $AA_IM[Pendelachse] == $SA_OSCILL_REVERSE_POS2[Pendelachse]

DO $AA_OVR[Pendelachse] = 0 $AA_OVR[Zustellachse] = 100

Erklärung:

$AA_IM[Pendelachse]: Aktuelle Position der Pendelachse

$SA_OSCILL_REVERSE_POS2[Pendelachse]: Umkehrpunkt 2 der Pendelachse

$AA_OVR[Pendelachse]: axialer Override der Pendelachse

$AA_OVR[Zustellachse]: axialer Override der Zustellachse

12.3.5 Pendelbewegung wieder starten

Funktion

Immer, wenn der Restweg für das aktuell zurückgelegte Wegstück der Zustellachse = 0 ist, also die Teilzustellung damit erfolgt ist, startet die Pendelachse über den Override.

Anwendung

Diese Synchronaktion wird dazu benutzt, die Bewegung der Pendelachse fortzusetzen, wenn die Teilzustellbewegung abgeschlossen ist. Falls die Pendelachse nicht auf das Ende der Teilzustellung warten soll, so ist die jeweilige Bewegungssynchronaktion, mit der die Pendelachse im Umkehrpunkt gestoppt wird, wegzulassen.

Programmierung

WHENEVER $AA_DTEPW[Zustellachse] == 0

DO $AA_OVR[Pendelachse] =100

Erklärung der Systemvariablen:

$AA_DTEPW[Zustellachse]: axialer Restweg für Zustellachse im Werkstückkoordinatensystem: Wegstück der Zustellachse

$AA_OVR[Pendelachse]: axialer Override für Pendelachse

Erklärung der Schlüsselwörter:

WHENEVER ... DO ... Immer wenn Bedingung erfüllt, dann...

12.3.6 Teilzustellung nicht zu früh starten

Funktion

Die in den vorhergehenden Abschnitten beschriebenen Funktionen verhindern eine Zustellbewegung außerhalb des Umkehrpunktes bzw. Umkehrbereiches. Nach erfolgter Zustellung muss jedoch ein erneutes Starten der nächsten Teilzustellung verhindert werden.

Anwendung

Dazu wird ein kanalspezifischer Merker verwendet, der am Ende der Teilzustellung (Teilrestweg == 0) gesetzt wird und beim Verlassen des Umkehrbereichs gelöscht wird. Dann wird mit einer Synchronaktion die nächste Zustellbewegung verhindert.

Programmierung

WHENEVER $AA_DTEPW[Zustellachse] == 0

DO $AC_MARKER[Index]=1

und z. B. für Umkehrpunkt1:

WHENEVER $AA_IM[Z]<> $SA_OSCILL_REVERSE_POS1[Z]

DO $AC_MARKER[Index]=0

WHENEVER $AC_MARKER[Index]==1

DO $AA_OVR[Zustellachse]=0

Erklärung der Systemvariablen:

$AA_DTEPW[Zustellachse]: axialer Restweg für Zustellachse im Werkstückkoordinatensystem: Wegstück der Zustellachse

$AC_MARKER[Index]: kanalspezifischer Merker mit Index

$AA_IM[Pendelachse]: Aktuelle Position der Pendelachse

$SA_OSCILL_REVERSE_POS1[Pendelachse]: Umkehrpunkt 1 der Pendelachse

$AA_OVR[Zustellachse]: axialer Override für Zustellachse

Erklärung der Schlüsselwörter:

WHENEVER ... DO ... Immer wenn Bedingung erfüllt, dann...

12.3.7 Zuordnung von Pendel- und Zustellachse OSCILL

Funktion

Mit dem Befehl OSCILL werden der Pendelachse eine oder mehrere Zustellachsen zugeordnet. Die Pendelbewegung beginnt.

Die zugeordneten Achsen werden über die VDI-Nahtstelle der PLC mitgeteilt. Falls die PLC die Pendelachse kontrolliert, muss sie nun auch die Zustellachsen überwachen und aus den Signalen für die Zustellachsen die Rückwirkungen auf die Pendelachse über 2 Halt-Bits der Nahtstelle erzeugen.

Anwendung

Die Achsen, deren Verhalten durch Synchronbedingungen zuvor festgelegt wurde, werden für die Aktivierung des Pendelns einander zugeordnet. Das Pendeln startet.

Programmierung

OSCILL[Pendelachse] = (Zustellachse1, Zustellachse2, Zustellachse3)

Die in Klammer gesetzten Zustellachse2 und Zustellachse3 entfallen einschließlich der Trennzeichen, wenn sie nicht benötigt werden.

12.3.8 Festlegung der Zustellungen POSP

Funktion

Der Steuerung wird für die Zustellachse mitgeteilt:

- Gesamtzustellung
- Teilzustellung am Umkehrpunkt/Umkehrbereich
- Teilzustellverhalten am Ende

Anwendung

Die Anweisung muss nach einer Pendelaktivierung mit `OSCILL` gegeben werden, um der Steuerung die in den Umkehrpunkten/Umkehrbereichen nötigen Zustellbeträge mitzuteilen.

Programmierung

POSP[Zustellachse] = (Endposition, Teilstück, Modus)

Endposition: Endposition für die Zustellachse, nachdem alle Teilzustellungen abgefahren sind.

Teilstück: Teilzustellung am Umkehrpunkt/Umkehrbereich

Modus 0: Für die beiden letzten Teilschritte erfolgt eine Aufteilung des verbleibenden Weges bis zum Zielpunkt auf zwei gleich große Restschritte (Vorbesetzung).

Modus 1: Die Teillänge wird so angepasst, dass die Summe aller errechneten Teillängen genau den Weg zum Zielpunkt ergibt.

12.3.9 Pendelumkehr von Extern

Zum Beispiel über die Umsetzung von Tasten durch die PLC kann erreicht werden, dass der Pendelbereich sich ändert bzw. die Pendelrichtung sich sofort umkehrt.

Über das flankengetriggerte PLC-Eingangssignal **Pendelumkehr** (DB31 DBB28 Bit0) wird die aktuelle Pendelbewegung abgebremst und dann weiter in Gegenrichtung gefahren. Die Bremsphase wird über das PLC-Ausgangssignal **Pendelumkehr aktiv** (DB31 DBB100 Bit2) zurückgemeldet.

Soll die Position, auf die abgebremst wurde, als **neue Umkehrposition** übernommen werden, so kann dies mit dem PLC-Signal **Umkehrpunkt ändern** (DB31 DBB28 Bit4) veranlasst werden.

Das PLC-Eingangssignal **Umkehrpunkt selektieren** (DB31 DBB28 Bit 3) wird nicht beachtet, sondern die Änderung wirkt bzgl. der letzten ausgelösten *Pendelumkehr von Extern*.

Es darf keine Änderung der Umkehrpunkte über Handrad bzw. JOG-Tasten für die entsprechende Achse aktiv sein. In diesem Fall wird der Anzeigealarm 20081 (Bremsposition kann nicht als Umkehrposition übernommen werden - Handrad aktiv) gemeldet. Nach Auflösung des Konflikts wird der Alarm selbstständig zurückgesetzt.

Haltezeit

Bei einer Richtungsänderung aufgrund einer *Pendelumkehr von Extern* wirkt keine Haltezeit. Es wird auf Genauhalt fein gewartet. Eine eingestellte Genauhaltbedingung wird eingehalten.

Zustellbewegung

Bei satzbezogenem Pendeln wird bei einer Richtungsänderung aufgrund einer *Pendelumkehr von Extern* keine Zustellbewegung ausgeführt, da die Umkehrposition nicht erreicht wird und damit die entsprechende Synchronaktion nicht erfüllt wird.

Systemvariablen

Die Position, auf die abgebremst wird, kann über die Systemvariable $AA_OSCILL_BREAK_POS1, beim Abbrechens des Anfahrens der Umkehrposition 1, bzw. über

$AA_OSCILL_BREAK_POS2, beim Abbrechens des Anfahrens der Umkehrposition 2, abgefragt werden.

Wird der entsprechende Umkehrpunkt erneut angefahren, so wird in $AA_OSCILL_BREAK_POS1 bzw. in $AA_OSCILL_BREAK_POS2 die Position des entsprechende Umkehrpunkt gemeldet.

D. h. nur nach einer *Pendelumkehr von Extern* unterscheiden sich die Werte in $AA_OSCILL_BREAK_POS1 und $AA_OSCILL_REVERSE_POS1 bzw. die Werte in $AA_OSCILL_BREAK_POS2 und $AA_OSCILL_REVERSE_POS2.

Damit kann die *Pendelumkehr von Extern* mit einer Synchronaktion **erkannt** werden, siehe Beispiele.

Sonderfälle

Wird das PLC-Eingangssignal Pendelumkehr bereits während des Anfahrens der Startposition aktiviert, so wird das Anfahren der Startposition abgebrochen und es wird mit dem Anfahren der Unterbrechungsposition 1 fortgesetzt.

Wird das PLC-Eingangssignal Pendelumkehr gesetzt, während eine Haltezeit abgewartet wird, so wird diese Haltezeit abgebrochen; falls Genauhalt fein noch nicht erreicht ist, wird darauf gewartet und dann die Bewegung fortgesetzt.

Wenn das PLC-Eingangssignal Pendelumkehr während des Anfahrens der Endposition aktiviert wird, so wird das Anfahren der Endposition abgebrochen und das Pendeln ist beendet.

Ein Beispiel zu Pendelumkehr von Extern finden Sie unter Kap. "Umkehrposition mit 'Pendelumkehr von Extern' über Synchronaktion ändern".

12.4 Randbedingungen

Verfügbarkeit der Funktion "Pendeln"

Pendeln ist eine Option mit Bestellnummer 6FC5 251-0AB04-0AA0.

Asynchrones und satzübergreifendes Pendeln ist für NCU570, 571, 572, 573 verfügbar.

Pendeln mit Bewegungssynchronaktionen steht mit NCU 572, 573 zur Verfügung.

12.5 Beispiele

Voraussetzungen

Die im Folgenden angegebenen Beispiele benötigen Anteile der NC-Sprache, die in den Abschnitten:

- Asynchrones Pendeln

und

- Pendeln über Bewegungssynchronaktionen gesteuert

angegeben wurden.

12.5.1 Beispiel Asynchrones Pendeln

Aufgabe

Die Pendelachse Z soll zwischen -10 und 10 pendeln. Umkehrpunkt 1 mit Genauhalt grob, Umkehrpunkt 2 ohne Genauhalt anfahren. Es soll mit Vorschub 5000 für die Pendelachse gearbeitet werden. Am Ende der Bearbeitung sollen 3 Ausfeuerungshübe erfolgen und die Endposition 30 mit der Pendelachse angesteuert werden. Der Vorschub für die Zustellachse ist 1000, Ende der Zustellung in X-Richtung ist bei 15.

Programmausschnitt

```
OSP1[Z]=-10              ; Umkehrpunkt 1
OSP2[Z]=10               ; Umkehrpunkt 2
OST1[Z]=-1               ; Haltezeit am Umkehrpunkt 1: Genauhalt grob
OST2[Z]=-2               ; Haltezeit am Umkehrpunkt 2: ohne Genauhalt
FA[Z]=5000               ; Vorschub Pendelachse
OSNSC[Z]=3               ; drei Ausfeuerungshübe
OSE[Z]=-3                ; Endposition
OS1 F500 X15             ; Pendeln starten, Zustellung X Achse
                         ; mit Vorschub 500, Zustellziel 15
```

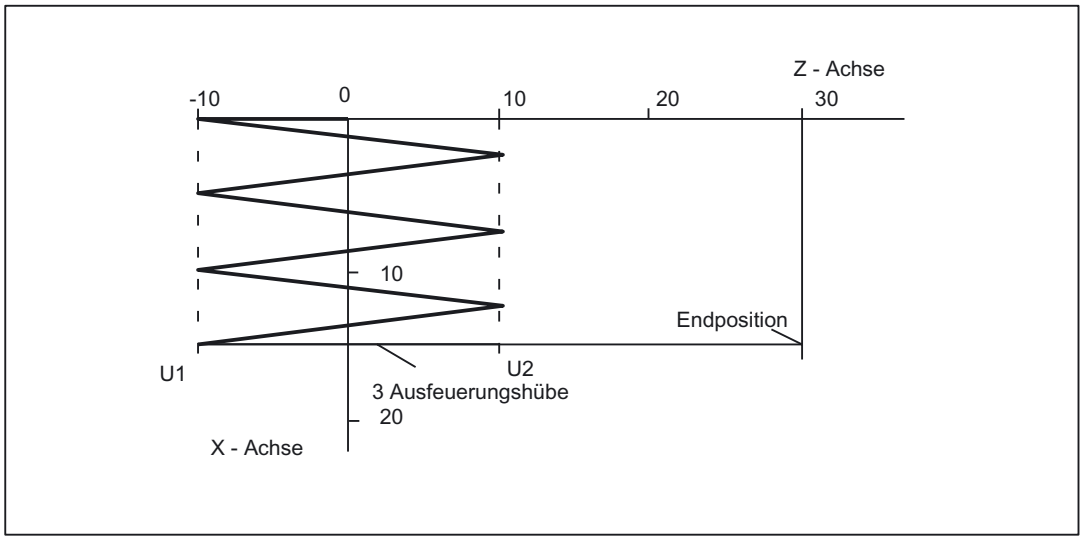


Bild 12-2 Abläufe von Pendelbewegungen und Zustellung, Beispiel 1

12.5.2 Beispiel 1 Pendeln mit Synchronaktionen

Aufgabe

Im Umkehrpunkt 1 soll die Zustellung direkt erfolgen und dabei auf das Beenden der Teilzustellung gewartet werden bevor die Pendelachse weiterfährt. Beim Umkehrpunkt 2 soll die Zustellung bereits im Abstand -6 vor dem Umkehrpunkt 2 erfolgen und die Pendelachse im Umkehrpunkt nicht auf das Beenden der Teilzustellung warten. Die Achse Z ist Pendelachse und die Achse X Zustellachse. (S. Kap. "Pendeln über Synchronaktionen gesteuert").

Hinweis

Die Settingdaten OSCILL_REVERSE_POS_1/2 sind Werte im Maschinen-Koordinatensystem, ein Vergleichen ist also nur mit $AA_IM[n] sinnvoll.

Programmausschnitt

```
; Beispiel1: Pendeln mit Synchronaktionen
OSP1[Z]=10 OSP2[Z]=60      ; Umkehrpunkt 1 und 2 erklären
OST1[Z]=-2 OST2[Z]=0       ; Umkehrpunkt 1: ohne Genauhalt
                           ; Umkehrpunkt 2: Genauhalt fein
FA[Z]=5000 FA[X]=250       ; Vorschub Pendelachse, Vorschub, Zustellachse
OSCTRL[Z]=(1+8+16,0)       ; Pendelbewegung abschalten im Umkehrpunkt 1
                           ; nach RWL Ausfeuern und Endposition anfahren
                           ; nach RWL entsprechende Umkehrposition
                           ; anfahren
OSNSC[Z]=3                 ; 3 Ausfeuerungshübe
```

```
OSE[Z]=0                    ; Endposition = 0;
WAITP(Z)                    ; erlaube Pendeln für die Z Achse


; Bewegungssynchronaktionen:
;
; Immer, wenn               die aktuelle Position der Pendelachse im
;                           Maschinenkoordinatensystem
; ungleich                  der Umkehrposition 1 ist
; dann                      setze den Merker mit dem Index 1 auf den Wert 0
;                           (Rücksetzen Merker 1)
WHENEVER $AA_IM[Z]<>$SA_OSCILL_REVERSE_POS1[Z] DO $AC_MARKER[1]=0
;
; Immer, wenn               die aktuelle Position der Pendelachse im
;                           Maschinenkoordinatensystem
; kleiner                   als der Beginn des Umkehrbereichs 2 (hier: Umkehrpunkt 2
;                           -6) ist,
; dann                      setze den axiale Override der Zustellachse auf 0%
; und                       setze den Merker mit dem Index 2 auf den Wert 0
;                           (rücksetzen Merker 2).
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0 $AC_MARKER[2]=0
;
; Immer, wenn               die aktuelle Position der Pendelachse im
;                           Maschinenkoordinatensystem
; gleich                    der Umkehrposition 1 ist,
; dann                      setze den axialen Override der Pendelachse auf 0%
;
; und                       setze den axialen Override der Zustellachse auf 100%
;                           (damit wird die vorhergehende Synchronaktion aufgehoben!)
;
WHENEVER $AA_IM[Z]==$SA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[Z]=0 $AA_OVR[X]=100
;
; Immer, wenn               der Restweg der Teilzustellung
; gleich                    0 ist,
; dann                      setze den Merker mit dem Index 2 auf den Wert 1
; und                       setze den Merker mit dem Index 1 auf den Wert 1
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[2]=1 $AC_MARKER[1]=1
;
; Immer, wenn               der Merker mit dem Index 2
; gleich                    1 ist,
; dann                      setze den axialen Override der Zustellachse auf 0%, damit
;                           wird eine zu frühe Zustellung (Pendelachse hat die
;                           Umkehrposition 1 noch nicht wieder verlassen) verhindert.
;
;
WHENEVER $AC_MARKER[2]==1 DO $AA_OVR[X]=0
;
; Immer, wenn               der Merker mit dem Index 1
```

```
; gleich                  1 ist,
; dann                    setze den axialen Override der Zustellachse auf 0%, damit
;                         wird eine zu frühe Zustellung (Pendelachse hat den
;                         Umkehrbereich 2 noch nicht wieder verlassen) verhindert
;
; und                     setze den axialen Override der Pendelachse auf 100%
;                         ('starte' Pendeln)
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100
;
; Wenn die aktuelle Position der Pendelachse im Maschinenkoordinatensystem
; gleich                  der Umkehrposition 1 ist,
; dann                    setze den axialen Override der Pendelachse auf 100%
;
; und                     setze den axialen Override der Zustellachse auf 0% (damit
;                         wird die zweite Synchronaktion einmalig aufgehoben!)
;
WHEN $AA_IM[Z]==$SA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[Z]=100 $AA_OVR[X]=0
;
;---------------------------------
---------
OSCILL[Z]=(X) POSP[X]=(5,1,1)       ; ordne der Pendelachse Z die Achse X als
;                                   Zustellachse zu, diese soll bis Endposition 5
;                                   zustellen in Teilschritten von 1 und die Summe
;                                   aller Teillängen soll genau die Endposition
                                    ergeben
;
M30                                 ; Programmende
```

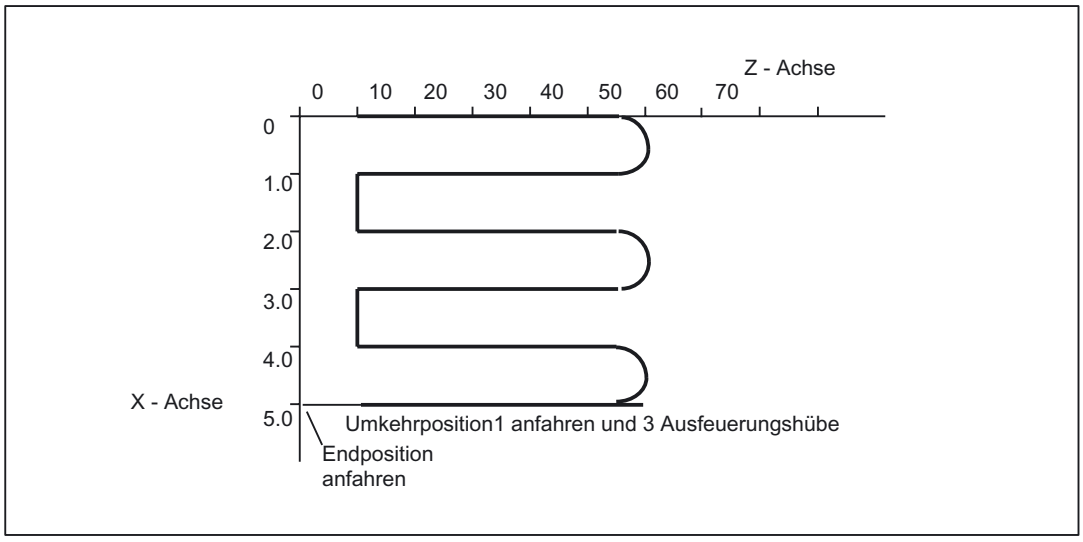


Bild 12-3 Abläufe von Pendelbewegungen und Zustellung, Beispiel 1

12.5.3 Beispiel 2 Pendeln mit Synchronaktionen

Aufgabe

Im Umkehrpunkt 1 soll keine Zustellung erfolgen. Beim Umkehrpunkt 2 soll die Zustellung bereits im Abstand ii2 vor dem Umkehrpunkt 2 erfolgen und die Pendelachse im Umkehrpunkt auf das Beenden der Teilzustellung warten. Die Achse Z ist Pendelachse und die Achse X Zustellachse.

Programmausschnitt

Beispiel 2: Pendeln mit Synchronaktionen

```
DEF INT ii2                ; Variable für Umkehrbereich 2 definieren
;
OSP1[Z]=10 OSP2[Z]=60      ; Umkehrpunkt 1 und 2 erklären
OST1[Z]=0 OST2[Z]=0        ; Umkehrpunkt 1: Genauhalt fein
                           ; Umkehrpunkt 2: Genauhalt fein
FA[Z]=5000 FA[X]=100       ; Vorschub Pendelachse, Vorschub Zustellachse
OSCTRL[Z]=(2+8+16,1)       ; Pendelbewegung abschalten im Umkehrpunkt 2
                           ; nach Restweg löschen Ausfeuern und Endposition
                           ; anfahren nach Restweg löschen entsprechende
                           ; Umkehrposition anfahren
OSNSC[Z]=3                 ; 3 Ausfeuerungshübe
OSE[Z]=70                  ; Endposition = 70
ii2=2                      ; Umkehrbereich einstellen
WAITP(Z)                   ; erlaube Pendeln für Z-Achse

; Bewegungssynchronaktionen:
; Immer, wenn              die aktuelle Position der Pendelachse im
;                          Maschinenkoordinatensystem
; kleiner                  dem Beginn des Umkehrbereichs 2 ist,
; dann                     setze den axialen Override der Zustellachse
;                          auf 0%
; und                      setze den Merker mit dem Index 0 auf den Wert 0
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z]-ii2 DO $AA_OVR[X]=0 $AC_MARKER[0]=0
;
; Immer, wenn              die aktuelle Position der Pendelachse im
;                          Maschinenkoordinatensystem
; größer gleich            der Umkehrposition 2 ist
; dann                     setze den axialen Override der Pendelachse
;                          auf 0%
WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0
;
; Immer wenn               der Restweg der Teilzustellung
; gleich                   0 ist,
; dann                     setze den Merker mit dem Index 0 auf den Wert 1
```

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[0]=1
;
; Immer, wenn              der Merker mit dem Index 0
; gleich                   1 ist,
; dann                     setze den axialen Override der Zustellachse auf 0%, damit
;                          wird eine zu frühe erneute Zustellung verhindert
;                          (Pendelachse hat den Umkehrbereich 2 noch nicht wieder
;                          verlassen, die Zustellachse ist aber bereit für eine
;                          erneute Zustellung)
; und                      setze den axialen Override der Pendelachse auf 100%
;                          (damit wird die 2. Synchronaktion aufgehoben)
;
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100
;
OSCILL[Z]=(X) POSP[X]=(5,1,1)       ; starten der Achsen
                                    ; der Pendelachse Z wird die Achse X als
                                    ; Zustellachse zugewiesen
                                    ; die Achse X soll bis Endposition 5 in
                                    ; Schritten von 1 fahren
;
M30
```

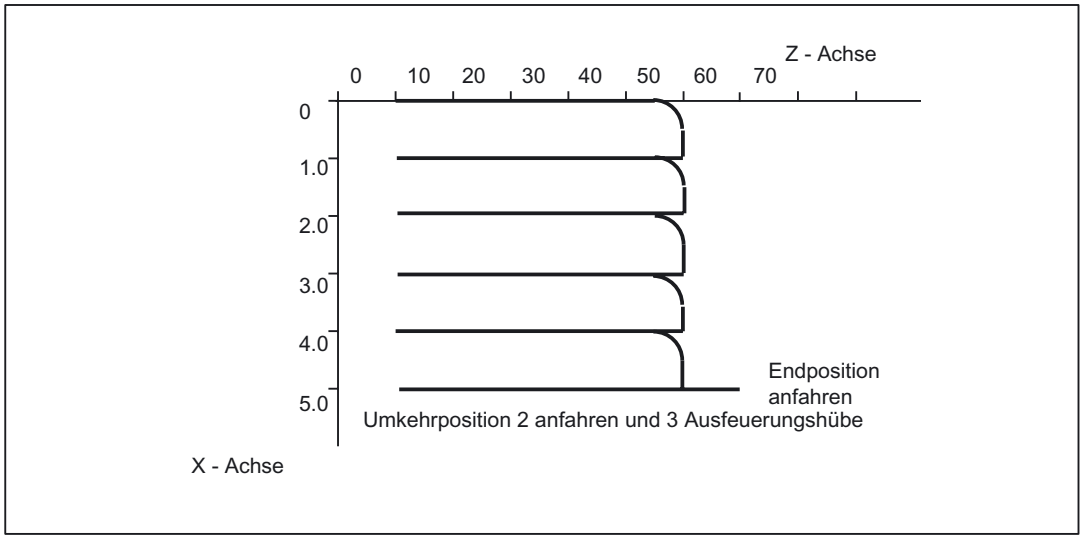


Bild 12-4 Abläufe von Pendelbewegungen und Zustellung, Beispiel 2

12.5.4 Beispiele Startposition

12.5.4.1 Startposition über Sprachbefehl definieren

```
WAITP(Z)                    ; erlaube Pendeln für die Z Achse
OSP1[Z]=10 OSP2[Z]=60       ; Umkehrpunkt 1 und 2 erklären
OST1[Z]=-2 OST2[Z]=0        ; Umkehrpunkt 1: ohne Genauhalt
                            ; Umkehrpunkt 2: Genauhalt fein
FA[Z]=5000 FA[X]=2000       ; Vorschub Pendelachse,
                            ; Vorschub Zustellachse
OSCTRL[Z]=(1+8+16,0)        ; Pendelbewegung abschalten im Umkehrpunkt1
                            ; nach RWL Ausfeuern und Endposition anfahren,
                            ; nach RWL entsprechende Umkehrposition
                            ; anfahren
OSNSC[Z]=3                  ; 3 Ausfeuerungshübe
OSE[Z]=0                    ; Endposition = 0
OSB[Z]=0                    ; Startposition = 0
OS[Z]=1 X15 F500            ; Start Pendeln, kontinuierliche Zustellung
OS[Z]=0                     ; Pendeln ausschalten
WAITP(Z)                    ; auf das Ende der Pendelbewegung warten
M30
```

Erläuterung

Zu Beginn der Pendelbewegung der Z-Achse wird zunächst die Startposition, im Beispiel Position = 0, angefahren und dann die Pendelbewegung zwischen den Umkehrpositionen 10 und 60 aufgenommen. Nachdem die X-Achse ihre Endposition 15 erreicht hat, endet das Pendeln mit 3 Ausfeuerhüben und dem Anfahren der Endposition 0.

12.5.4.2 Pendeln über Settingdaten starten

```
WAITP(Z)
STOPRE
$SA_OSCILL_REVERSE_POS1[ Z ] = -10        ; Umkehrposition 1 = -10
$SA_OSCILL_REVERSE_POS2[ Z ] = 30         ; Umkehrposition 2 = 30
$SA_OSCILL_START_POS[Z] = -50             ; Startposition = -50
$SA_OSCILL_CTRL_MASK[Z] = 512             ; Startposition anfahren,
                                          ; beim Abschalten im nächsten
                                          ; Umkehrpunkt stoppen
                                          ; keine Endposition anfahren
                                          ; bei RWL keine Ausfeuerhübe
$SA_OSCILL_VELO[ Z ] = 5000               ; Vorschub Pendelachse
$SA_OSCILL_IS_ACTIVE[ Z ] = 1             ; starten
$SA_OSCILL_DWELL_TIME1[ Z ] = -2          ; ohne Warten auf Genauhalt
$SA_OSCILL_DWELL_TIME2[ Z ] = 0           ; warten auf Genauhalt fein
```

```
STOPRE
X30 F100
$SA_OSCILL_IS_ACTIVE[ Z ] = 0              ; stoppen
WAITP(Z)
M30
```

Erläuterungen

Zu Beginn der Pendelbewegung der Z-Achse wird zunächst die Startposition, im Beispiel Position = -50, angefahren und dann die Pendelbewegung zwischen den Umkehrpositionen -10 und 30 aufgenommen. Nachdem die X-Achse ihre Endposition 30 erreicht hat, endet das Pendeln im nächsten angefahrenen Umkehrpunkt.

12.5.4.3 Satzbezogenes Pendeln (Startposition = Umkehrpunkt 1)

Pendeln mit Synchronaktionen

```
N701                              ; Pendeln mit Synchronaktionen,
                                  ; Startposition == Umkehrpunkt 1
;
N702 OSP1[Z]=10 OSP2[Z]=60        ; Umkehrpunkt 1 und 2 erklären
N703 OST1[Z]=0 OST2[Z]=0          ; Umkehrpunkt 1: Genauhalt grob
                                  ; Umkehrpunkt 2: Genauhalt fein
N704 FA[Z]=5000 FA[X]=2000        ; Vorschub Pendelachse,
                                  ; Vorschub Zustellachse
N705 OSCTRL[Z]=(1+8+16,0)         ; Pendelbewegung abschalten im
                                  : Umkehrpunkt 1 nach RWL Ausfeuern
                                  ; und Endposition anfahren
                                  ; nach RWL entsprechende Umkehrposition
                                  ; anfahren
N706 OSNSC[Z]=3                   ; 3 Ausfeuerungshübe
N707 OSE[Z]=0                     ; Endposition = 0
N708 OSB[Z]=10                    ; Startposition = 10
N709 WAITP(Z)                     ; erlaube Pendeln für die Z Achse
;
; Bewegungssynchronaktionen:
; Setze Merker mit Index 2 auf 1 (Initialisierung)
WHEN TRUE DO $AC_MARKER[2]=1
;
Immer, wenn                       der Merker mit Index 2 gleich 0 ist und die
;                                 aktuelle Position der Pendelachse
;                                 ungleich der Umkehrposition 1 ist
Dann                              setze den Merker mit Index 1 auf 0.
;
WHENEVER ($AC_MARKER[2] == 0) AND $AA_IW[Z]>$SA_OSCILL_REVERSE_POS1[Z])
DO $AC_MARKER[1]=0
```

```
; Immer, wenn                    die aktuelle Position der Pendelachse kleiner als
;                                der Beginn des Umkehrbereichs 2 ist,
;
; dann                           setze den axiale Override der Zustellachse auf 0
;                                und setze den Merker mit Index 0 auf 0
;
WHENEVER $AA_IW[Z]<$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0 $AC_MARKER[0]=0
;
; Immer, wenn                    die aktuelle Position der Pendelachse gleich der
;                                Umkehrposition 1 ist,
; dann                           setze den axialen Override der Pendelachse auf 0
;                                und setze den axialen Override der Zustellachse
;                                auf 100% (damit wird die vorhergehende
;                                Synchronaktion aufgehoben!)
WHENEVER $AA_IW[Z]==$SA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[Z]=0 $AA_OVR[X]=100
;
; Immer, wenn                    der Restweg der Teilzustellung gleich 0 ist,
;
; dann                           setze den Merker mit dem Index 0 auf 1 und setze
;                                den Merker mit dem Index 1 auf 1
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[0]=1 $AC_MARKER[1]=1
;
; Immer, wenn                    der Merker mit Index 0 gleich 1 ist,
; dann                           setze den axialen Override der Zustellachse auf
;                                0, damit wird eine zu frühe erneute Zustellung
;                                verhindert!
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0
;
; Immer wenn                     der Merker mit Index 1 gleich 1 ist,
; dann                           setze den axialen Override der Zustellachse auf
;                                0, (damit wird eine zu frühe erneute Zustellung
;                                verhindert!) und setze den axialen Override der
;                                Pendelachse auf 100% (damit wird die
;                                vorhergehende Synchronaktion aufgehoben!)
;
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100
;
; Wenn                           die aktuelle Position der Pendelachse gleich der
;                                Umkehrposition 1 ist,
; dann                           setze den Merker mit Index 2 zurück, schalte 1.
;                                Synchronaktion frei (keine Zustellung bei
;                                Erreichen der Startposition == Umkehrposition 1)
WHEN $AA_IW[Z]==$SA_OSCILL_REVERSE_POS1[Z] DO $AC_MARKER[2]=0
;
;---------------------------------
N750 OSCILL[Z]=(X) POSP[X]=(5,1,1)
; Ordne der Pendelachse Z die Achse X als Zustellachse zu,
; diese soll bis Endposition 5 zustellen
; in Teilschritten von 1 und die Summe aller Teillängen
```

```
; soll genau die Endposition ergeben.
;
N780 WAITP(Z)                    ; Z - Achse wieder freigeben
;
N790 X0 Z0
N799 M30                         ; Programmende
```

Erläuterungen

Die Startposition stimmt mit der Umkehrposition 1 überein. Mit den Synchronaktionen WHEN .... (siehe oben) wird eine Zustellung bei Erreichen der Startposition verhindert.

12.5.5 Beispiel Pendelumkehr von Extern

12.5.5.1 Umkehrposition mit "Pendelumkehr von Extern" über Synchronaktion ändern

```
DEFINE BREAKPZ AS $AA_OSCILL_BREAK_POS1[Z]
DEFINE REVPZ AS $SA_OSCILL_REVERSE_POS1[Z]

WAITP(Z)                       ; erlaube Pendeln für die Z Achse
OSP1[Z]=10 OSP2[Z]=60          ; Umkehrpunkt 1 und 2 erklären
OSE[Z]=0                       ; Endposition = 0
OSB[Z]=0                       ; Startposition = 0

                               ; Bei Pendelumkehr von extern für die
                               ; Pendelumkehrposition 1, dies anpassen
WHENEVER BREAKPZ <> REVPZ DO $$SA_OSCILL_REVERSE_POS1 = BREAKPZ
OS[Z]=1 X150 F500              ; Start Pendeln, kontinuierliche Zustellung
OS[Z]=0                        ; Pendeln ausschalten
WAITP(Z)                       ; auf das Ende der Pendelbewegung warten
M30
```

12.6 Datenlisten

12.6.1 Maschinendaten

12.6.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
10710	PROG_SD_RESET_SAVE_TAB	Zu sichernde Pendeleinstellungen aus SD
11460	OSCILL_MODE_MASK	Steuermaske für asynchrones Pendeln

12.6.2 Settingdaten

12.6.2.1 Achs-/Spindel-spezifische Settingdaten

Nummer	Bezeichner: $SA_	Beschreibung
43700	OSCILL_REVERSE_POS1	Position im Umkehrpunkt 1
43710	OSCILL_REVERSE_POS2	Position im Umkehrpunkt 2
43720	OSCILL_DWELL_TIME1	Haltezeit im Umkehrpunkt 1
43730	OSCILL_DWELL_TIME2	Haltezeit im Umkehrpunkt 2
43740	OSCILL_VELO	Vorschubgeschwindigkeit der Pendelachse
43750	OSCILL_NUM_SPARK_CYCLES	Anzahl der Ausfeuerungshübe
43760	OSCILL_END_POS	Position nach den Ausfeuerungshüben am Ende des Pendelns
43770	OSCILL_CTRL_MASK	Steuermaske für Pendeln
43780	OSCILL_IS_ACTIVE	Pendelbewegung ein/aus
43790	OSCILL_START_POS	Position, die nach Pendelstart vor Umkehrpunkt 1 angefahren wird, wenn in SD43770 aktiviert.

12.6.3 Signale

12.6.3.1 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Pendelumkehr von Extern	DB31, ...DBX28.0	-
Umkehrpunkt setzen	DB31, ...DBX28.3	-
Umkehrpunkt ändern	DB31, ...DBX28.4	-

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Halt im nächsten Umkehrpunkt	DB31, ...DBX28.5	-
Halt mit Bremsrampe	DB31, ...DBX28.6	-
PLC kontrolliert Achse	DB31, ...DBX28.7	-

12.6.3.2 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Pendelumkehr ist aktiv	DB31, ...DBX100.2	-
Pendeln nicht startbar	DB31, ...DBX100.3	-
Fehler während Pendelbewegung	DB31, ...DBX100.4	-
Ausfeuern aktiv	DB31, ...DBX100.5	-
Pendelbewegung aktiv	DB31, ...DBX100.6	-
Pendeln aktiv	DB31, ...DBX100.7	-

12.6.4 Systemvariablen

12.6.4.1 Hauptlaufvariablen für Bewegungssynchronaktionen

Hauptlaufvariable_lesen

Für Hauptlaufvariable_lesen sind folgende Variablen vorgesehen:

$A_IN[<arith. Ausdruck>**]**	digitaler Eingang (Boolean)
$A_OUT[<arith. Ausdruck>**]**	digitaler Ausgang (Boolean)
$A_INA[<arith. Ausdruck>**]**	analoger Eingang (Boolean)
$A_OUTA[<arith. Ausdruck>**]**	analoger Ausgang (Boolean)
$A_INCO[<arith. Ausdruck>**]**	Comparator Eingänge (Boolean)
$AA_IW[<axialer Ausdruck>**]**	Istposition Achse PCS (Real)
$AA_IB[<axialer Ausdruck>**]**	Istposition Achse BCS (Real)
$AA_IM[<axialer Ausdruck>**]**	Istposition Achse MCS (IPO-Sollwerte) (Real) Mit $AA_IM[S1] können Istwerte für Spindeln ausgewertet werden. Für Spindeln und Rundachsen erfolgt in Abhängigkeit vom Maschinendatum $MA_ROT_IS_MODULO und $MA_DISPLAY_IS_MODULO Modulo-Rechnung.
$AA_OSCILL_BREAK_POS1	Bremsposition nach Pendelumkehr von Extern beim Anfahren von Umkehrpunkt 1
$AA_OSCILL_BREAK_POS2	Bremsposition nach Pendelumkehr von Extern beim Anfahren von Umkehrpunkt 2

$AC_TIME	Zeit vom Satzanfang (Real) in Sekunden (einschließlich Zeiten für intern generierte Zwischensätze)
$AC_TIMES	Zeit vom Satzanfang (REAL) in Sekunden (ohne Zeiten für intern generierte Zwischensätze)
$AC_TIMEC	Zeit vom Satzanfang (Real) in IPO-Takten (einschließlich Takten für intern generierte Zwischensätze)
$AC_TIMESC	Zeit vom Satzanfang (Real) in IPO-Takten (ohne Takte für intern generierte Zwischensätze)
$AC_DTBB	Entfernung vom Satzanfang im BCS (Distance to begin, baseCoor) (Real)
$AC_DTBW	Entfernung vom Satzanfang im PCS (Distance to begin, workpieceCoor) (Real)
$AA_DTBB[<axialer Ausdruck>**]**	axialer Weg vom Satzanfang im BCS (Distance to begin, baseCoor) (Real)
$AA_DTBW[<axialer Ausdruck>**]**	axialer Weg vom Satzanfang im PCS (Distance to begin, workpieceCoor) (Real)
$AC_DTEB	Entfernung zum Satzende im BCS (Distance to end) (Distance to end, baseCoor) (Real)
$AC_DTEW	Entfernung zum Satzende im PCS (Distance to end, workpieceCoor) (Real)
$AA_DTEB[<axialer Ausdruck>**]**	axialer Weg bis Bewegungsende im BCS (Distance to begin, baseCoor) (Real)
$AA_DTEW[<axialer Ausdruck>**]**	axialer Weg bis Bewegungsende im PCS (Distance to end, workpieceCoor) (Real)
$AC_PLTBB	Bahnweg vom Satzanfang im BCS (Path Length from begin, baseCoor) (Real)
$AC_PLTEB	Bahnweg zum Satzende im BCS (Distance to end) (Path Length to end, baseCoor) (Real)
$AC_VACTB	Bahngeschwindigkeit im BCS (Velocity actual, baseCoor) (Real)
$AC_VACTW	Bahngeschwindigkeit im PCS (Velocity actual, workPieceCoor) (Real)
$AA_VACTB[<axialer Ausdruck>**]**	Achsgeschwindigkeit im BCS (Velocity actual, baseCoor) (Real)
$AA_VACTW[<axialer Ausdruck>**]**	Achsgeschwindigkeit im PCS (Velocity actual, workPieceCoor) (Real)
$AA_DTEPB[<axialer Ausdruck>**]**	axialer Restweg für Zustellung Pendeln im BCS (Distance to end, pendulum,baseCoor) (Real)
$AA_DTEPW[<axialer Ausdruck>**]**	axialer Restweg für Zustellung Pendeln im PCS (Distance to end, pendulum,workpieceCoor) (Real)
$AC_DTEPB	Bahnrestweg für Zustellung Pendeln im BCS (nicht P2) (Distance to end, pendulum,baseCoor) (Real)
$AC_DTEPW	Bahnrestweg für Zustellung Pendeln im PCS (nicht P2) (Distance to end, pendulum, workpieceCoor) (Real)

$AC_PATHN	(Path-Parameter normalized) (Real) Normierter Bahnparameter: 0 für Satzanfang bis 1 für Satzende
$AA_LOAD[<axialer Ausdruck>**]**	Antriebsauslastung (nur bei 611D)
$AA_POWER[<axialer Ausdruck>**]**	Antriebswirkleistung in W (nur bei 611D)
$AA_TORQUE[<axialer Ausdruck>**]**	Antriebsmomentensollwert in Nm (nur bei 611D)
$AA_CURR[<axialer Ausdruck>**]**	Stromistwert der Achse (nur bei 611D)
$AC_MARKER[<arithmetischer_Ausdruck>**]** (int)	Merkervariable: kann in Synchronaktionen zum Aufbau komplexer Bedingungen verwendet werden: Es stehen 8 Merker (Index 0 - 7) zur Verfügung. Bei Reset werden die Merker zu 0 gesetzt. Bsp.: WHEN .....DO $AC_MARKER[0]=2 WHEN .....DO $AC_MARKER[0]=3 WHEN $AC_MARKER[0]==3 DO $AC_OVR=50 Können auch unabhängig von Synchronaktionen im Teileprogramm gelesen und beschrieben werden: IF $AC_MARKER == 4 GOTOF SPRUNG
$AC_PARAM[<arithmetischer_Ausdruck>**]** (Real)	Gleitkomma-Parameter für Synchronaktionen. Dient zum Zwischenspeichern und Auswerten in Synchronaktionen. Es stehen 50 Parameter (Index 0 - 49) zur Verfügung.
$AA_OSCILL_REVERSE_POS1 [<axialer Ausdruck>**]** (Real)	
$AA_OSCILL_REVERSE_POS2 [<axialer Ausdruck>**]** (Real)	aktuelle Umkehrpositionen 1 und 2 für Pendeln: Es wird jeweils der aktuelle Settingdatenwert aus $SA_OSCILL_REVERSE_POS1 bzw. $SA_OSCILL_REVERSE_POS2 gelesen. Damit werden Änderungen an den Umkehrpositionen in den Settingdaten bei aktivem Pendeln, d. h. während einer aktiven Synchronaktion wirksam.

Bedingungen

Bedingungen für bewegungssynchrone Aktionen werden formuliert:

Hauptlaufvariable Vergleichsoperator Ausdruck

Details siehe:
Literatur:
/FBSY/ Funktionshandbuch Synchronaktionen

R2: Rundachsen

13

13.1 Kurzbeschreibung

Rundachsen in Werkzeugmaschinen

Heute werden bei vielen Werkzeugmaschinen Rundachsen verwendet. Sie werden zur Werkzeug- bzw. Werkstückorientierung, für Hilfsbewegungen oder aus technologischen bzw. kinematischen Gründen benötigt.

Typische Anwendungsbeispiele für den Einsatz von Rundachsen sind die 5-Achsen-Fräsmaschinen. Nur mit Hilfe von Rundachsen kann bei diesen Maschinen die Spitze des Werkzeuges an jeden beliebigen Punkt des Werkstückes positioniert werden.

Abhängig vom jeweiligen Maschinentyp werden an eine Rundachse viele unterschiedliche Anforderungen gestellt. Damit die Steuerung an die verschiedenen Maschinentypen angepasst werden kann, sind die einzelnen Rundachsfunktionen mit Hilfe von Maschinendaten oder durch besondere Programmierung aktivierbar.

Rundachsen werden grundsätzlich in Grad programmiert. Sie zeichnen sich in der Regel dadurch aus, dass sie nach genau 1 Umdrehung die gleiche Position wieder einnehmen (Modulo 360°). Abhängig vom Anwendungsfall kann der Verfahrbereich der Rundachse auf kleiner 360° beschränkt (z. B. bei Schwenkachsen für Werkzeughalter) oder endlos (z. B. bei Drehbewegungen des Werkzeugs oder Werkstücks) sein.

In vielen Punkten ist das Verhalten sowie die Merkmale der Rundachsen identisch mit denen der Linearachsen. In der vorliegenden Funktionsbeschreibung werden lediglich die speziellen Merkmale der Rundachse sowie die Unterschiede gegenüber einer Linearachse beschrieben.

Definition der Rundachse

Eine Achse kann mit Hilfe des folgenden achsspezifischen Maschinendatums als Rundachse deklariert werden:
MD30300 $MA_IS_ROT_AX
.

Geometrieachsen sind als Linearachsen definiert. Der Versuch sie zu Rundachsen zu erklären, wird mit Alarm 4200 (Geometrieachse darf nicht als Rundachse definiert werden) abgelehnt.

Nur wenn eine Achse als Rundachse deklariert ist, können die nachfolgend beschriebenen Funktionen (z. B. unendlicher Verfahrbereich, Moduloanzeige der Achsposition) ausgeführt und verwendet werden.

Es können gleichzeitig mehrere Achsen als Rundachsen deklariert werden.

Arten von Rundachsen

Je nach Anwendungsfall ist der Arbeitsbereich der Rundachse unendlich, d. h. endlos drehend in beiden Drehrichtungen (MD30310 $MA_ROT_IS_MODULO = 1) oder durch Software-Endschalter begrenzt (z. B. Arbeitsbereich zwischen 0 ... 60°) bzw. auf eine entsprechende Anzahl von Umdrehungen (z. B. 1000°) beschränkt.

Nachfolgend werden einige typische Anwendungen von Rundachsen aufgezeigt.

Typische Anwendungsfälle

- 5-Achsbearbeitung (Arbeitsbereich endlos oder begrenzt)
- Rundachse für Unrundbearbeitung (Arbeitsbereich endlos)
- Rundachse für Rund- und Formschleifen (Arbeitsbereich endlos)
- C-Achse bei `TRANSMIT` (Arbeitsbereich endlos)
- Rundachse bei Wickelmaschinen (Arbeitsbereich endlos)
- Werkstückdrehachse (C) bei Wälzfräsmaschinen (Arbeitsbereich endlos)
- Werkzeugrundmagazine und Werkzeugrevolver (Arbeitsbereich endlos)
- Rundachse für Mantelflächentransformation (Arbeitsbereich begrenzt)
- Schwenkachsen zum Greifen (Arbeitsbereich 360°)
- Rundachsen zum Schwenken (Arbeitsbereich < 360°; z. B. 60°)
- Fräserschwenkachse (A) bei Wälzfräsmaschinen (Arbeitsbereich z. B. 90°)

Achsadressen

Die Bezeichnung der Koordinatenachsen und Bewegungsrichtungen numerisch gesteuerter Werkzeugmaschinen ist nach DIN festgelegt.

Als Achsadressen von Rund- oder Schwenkachsen sind laut DIN66025 festgelegt:

- A, B und C mit X, Y und Z als Mittelachse

D. h. A dreht um X, B dreht um Y und C dreht um Z (siehe Bild).

- Die positive Drehrichtung der Rundachsen entspricht der Rechtsdrehung bei Blickrichtung in die positive Achsrichtung der jeweiligen Mittelachse (siehe Bild).

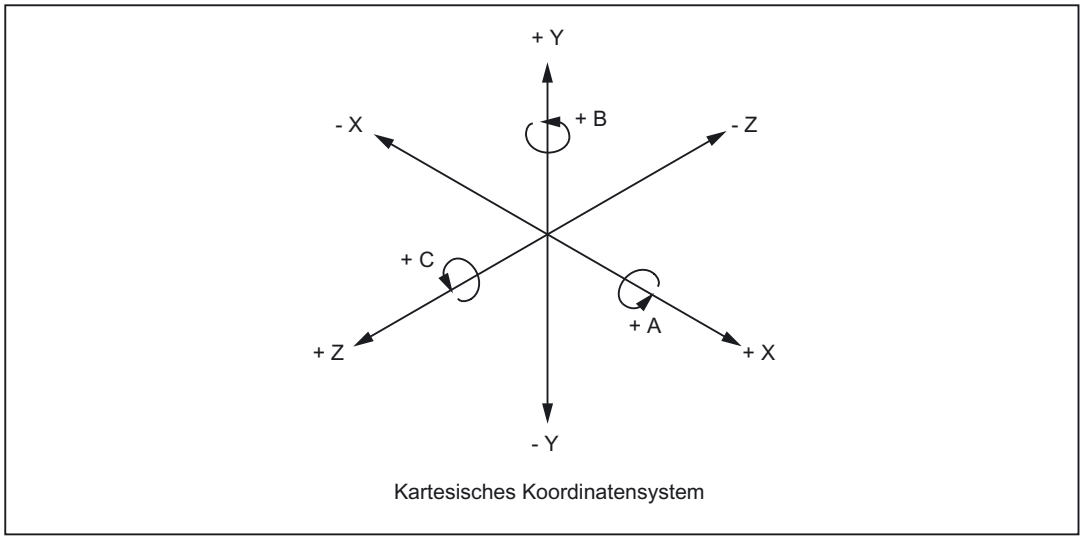


Bild 13-1 Achsbezeichnungen und Bewegungsrichtungen von Rundachsen

Für zusätzliche Rundachsen kann die erweiterte Adressierung (z. B. C2=) oder auch freie über Projektierung festgelegte Achsadressen verwendet werden.

Hinweis

Das Maschinendatum:
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB
(Zuordnung Geometrieachse zu Kanalachse)
muss für die entsprechende Achse angepasst werden.

Maßeinheiten

Standardmäßig gelten bei den Rundachsen für die Ein- und Ausgabe folgende Maßeinheiten:

Maßeinheiten für Rundachsen	
Physikalische Größe	Einheit
Winkel-Position	Grad
Programmierte Winkel-Geschwindigkeit	Grad/Minute
MD für Winkel-Geschwindigkeit	1) Umdr./Minute
MD für Winkel-Beschleunigung	1) Umdr./Sekunde^2
MD für Winkel-Ruckbegrenzung	1)Umdr./Sekunde^3

1) Diese Einheiten werden von der Steuerung bei den achsspezifischen Maschinendaten interpretiert, sobald die Achse als Rundachse deklariert wird. Der Anwender hat die Möglichkeit, über Maschinendaten andere Einheiten für die Daten-Ein-/Ausgabe zu definieren.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (G2)

Arbeitsbereich

Der Arbeitsbereich kann mit Hilfe von achsspezifischen Maschinen- und Settingdaten (Software-Endschalter und Arbeitsfeldbegrenzungen) festgelegt werden. Sobald für die Rundachse die Modulowandlung aktiviert ist (MD30310 $MA ROT_IS_MODULO = 1), ist der Arbeitsbereich unendlich und die Software-Endschalter sowie die Arbeitsfeldbegrenzungen sind unwirksam.

Über das Nahtstellensignal:
DB31, ... DBX12.4 (Modulo Limit Enabled)
können die SW-Endschalter/Arbeitsfeldbegrenzung auch für Modulo-Rundachsen dynamisch durch die PLC (ggf. ausgelöst vom Teileprogramm über M-/H-Funktionen) aktiviert werden.
Die Rückmeldung der NC erfolgt über das Nahtstellensignal:
DB31, ... DBX74.4 (Modulo Limit Enabled aktiv).

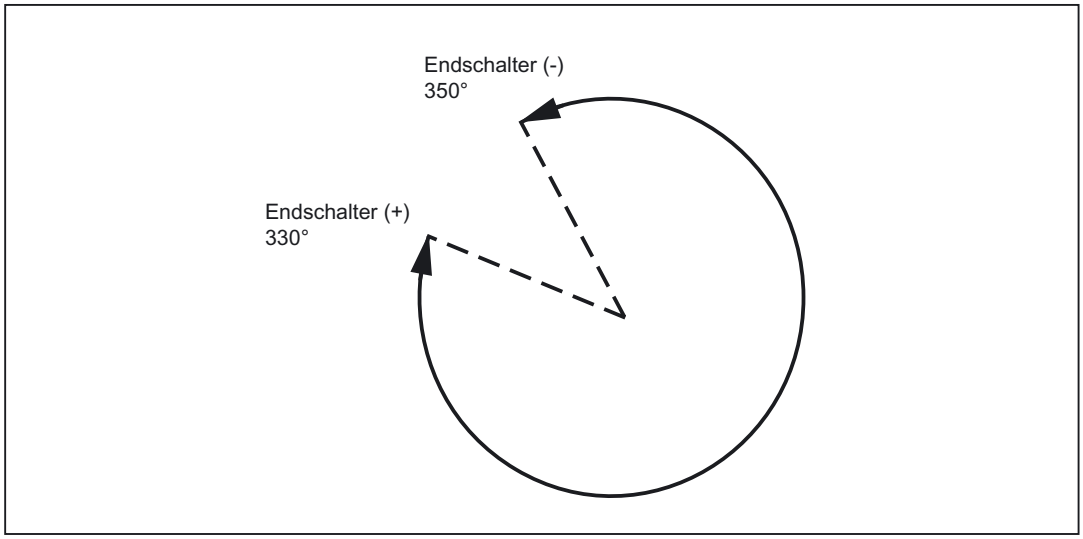


Bild 13-2 Begrenzter Arbeitsbereich einer Modulo-Rundachse

Positionsanzeige

Der Wertebereich für die Positionsanzeige kann auf die bei Rundachsen häufig gewählte Modulo 360°-Darstellung eingestellt werden:
MD30320 $MA_DISPLAY_IS_MODULO = 1

Vorschub

Der programmierte Vorschub F entspricht bei Rundachsen einer Winkelgeschwindigkeit [Grad/min].

Verfahren Rundachsen und Linearachsen bei `G94` oder `G95` gemeinsam eine Bahn, so ist der Vorschub in der Maßeinheit der Linearachsen zu interpretieren [z. B. mm/min, inch/min].

Die Tangentialgeschwindigkeit der Rundachse bezieht sich dabei auf den Durchmesser D_E (Einheitsdurchmesser $D_E=360/\pi$). Bei dem Einheitsdurchmesser $D=D_E$ sind die programmierte Winkelgeschwindigkeit in Grad/min und die Tangentialgeschwindigkeit in mm/min (bzw. inch/min) zahlenmäßig gleich.

Für die Tangentialgeschwindigkeit gilt allgemein:

$F = F_W * D / D_E$	F	=Tangentialgeschwindigkeit [mm/min]
	F_W	= Winkelgeschwindigkeit [Grad/min]
	D	= Durchmesser, an dem F wirkt [mm]
mit $D_E = 360 / \pi$	D_E=	Einheitsdurchmesser [mm]
	π	= Kreiskonstante Pi

Umdrehungsvorschub

In der Betriebsart JOG ist das Verhalten der Achse/Spindel auch von der Einstellung des Settingdatums abhängig:
SD41100 $SN_JOG_REV_IS_ACTIVE (Umdrehungsvorschub bei JOG aktiv)

SD41100 $SN_JOG_REV_IS_ACTIVE	
aktiv	Eine Achse/Spindel wird immer mit Umdrehungsvorschub: MD32050 $MA_JOG_REV_VELO (Umdrehungsvorschub bei JOG) bzw. MD32040 $MA_JOG_REV_VELO_RAPID (Umdrehungsvorschub bei JOG mit Eilgangsüberlagerung) in Abhängigkeit von der Masterspindel verfahren.
nicht aktiv	Verhalten der Achse/Spindel ist abhängig vom Settingdatum: SD43300 $SA_ASSIGN_FEED_PER_REV_SOURCE (Umdrehungsvorschub für Positionsachsen/-spindeln)
	Verhalten einer Geometrieachse, auf die ein Frame mit Rotation wirkt, ist abhängig vom kanalspezifischen Settingdatum: SD42600 $SC_JOG_FEED_PER_REV_SOURCE (in der Betriebsart JOG Umdrehungsvorschub für Geometrieachsen auf die ein Frame mit Rotation wirkt)

13.2 Modulo 360 Grad

Begriff Modulo 360°

Bei Rundachsen erfolgt häufig die Programmierung in der 360°-Darstellung. Voraussetzung für die Modulo-Eigenschaft ist die Definition der Achse als Rundachse.

Unter dem Begriff "Modulo" bei einer Rundachse versteht man eine steuerungsinterne Abbildung der Position der Rundachse im Bereich von 0° bis 359,999°. Bei Wegvorgaben > 360° (z. B. bei Kettenmaßprogrammierung `G91`) wird die Position durch steuerungsinterne Umrechnung in den Wertebereich zwischen 0° bis 360° abgebildet. Die Abbildung findet sowohl in JOG als auch in AUTOMATIK statt. Ausnahme: Service-Anzeige.

Im folgenden Bild ist die Absolutposition der Rundachse in positiver Drehrichtung als Spirale dargestellt. Mit Hilfe eines Zeigers auf dieser Spirale wird die tatsächliche Absolutposition markiert (Beispiel: Punkt C' = 420°). Durch Rückführung des Zeigers auf den Kreis (Position 0° von Spirale und Kreis sind identisch) kann zu jeder Absolutposition eine zugehörige Moduloposition innerhalb 360° zugeordnet werden. Im Beispiel wird die absolute Position Punkt C' = 420° durch Modulo-Wandlung in den Punkt C = 60° abgebildet.

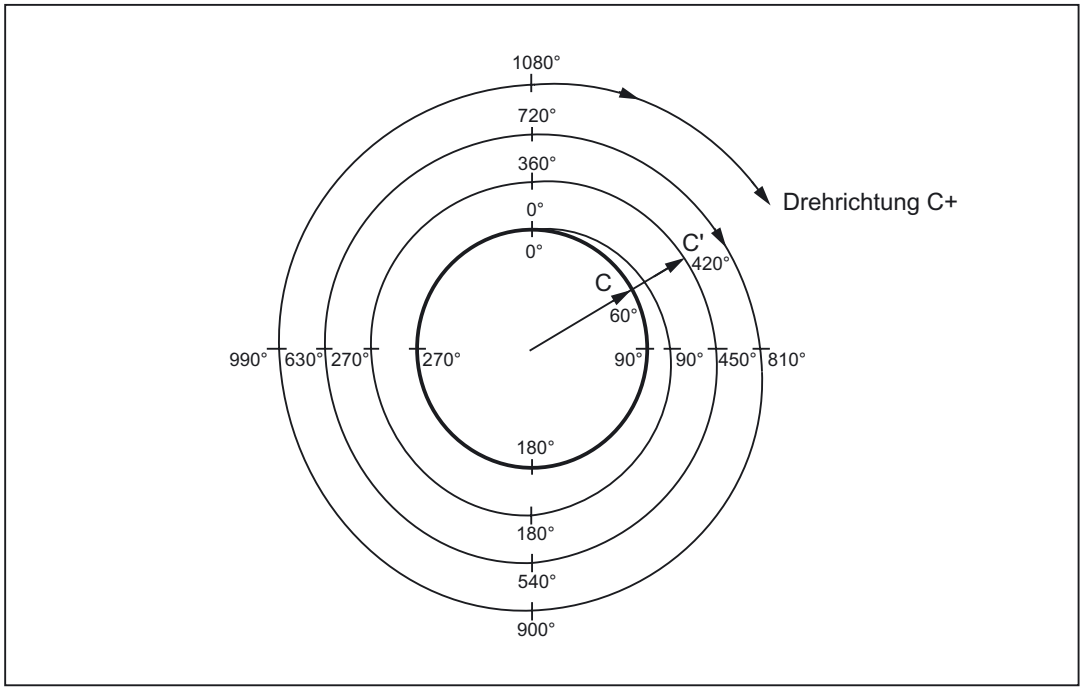


Bild 13-3 Abbildung Modulo 360°

Maschinendateneinstellungen

Mit Hilfe von Maschinendaten kann je nach Maschinenanforderung für jede Rundachse die Programmierung und Positionierung (MD30310 $MA_ROT_IS_MODULO) sowie die Positionsanzeige (MD30320 $MA_DISPLAY_IS_MODULO) in Modulo 360° einzeln festgelegt werden.

Achse ist Modulo

MD30310 $MA_ROT_IS_MODULO = 1:

Bei Aktivierung des Maschinendatums kommt das spezielle Rundachsverhalten zum Tragen. Damit wird das Positionierverhalten der Rundachse bei der Programmierung (G90, AC, ACP, ACN oder DC) festgelegt. Dabei wird steuerungsintern nach der Einberechnung der aktuellen Nullpunktverschiebungen eine Modulo 360°-Darstellung durchgeführt. Anschließend wird die so ermittelte **Zielposition innerhalb einer Umdrehung** angefahren.

Die Software-Endschalter und die Arbeitsfeldbegrenzungen sind unwirksam und der Arbeitsbereich ist somit **unendlich** (endlosdrehende Rundachse).

Positionsanzeige Modulo

MD30320 $MA_DISPLAY_IS_MODULO	
= 1	Bei Rundachsen wird die Positionsanzeige häufig mit "Modulo 360°"(1 Umdrehung) benötigt; d. h. die Anzeige wird bei positiver Drehrichtung steuerungsintern nach 359,999° periodisch wieder auf 0,000° zurückgesetzt; bei negativer Drehrichtung werden die Positionen ebenfalls im Bereich von 0°...359,999° angezeigt.
= 0	Im Gegensatz zur Anzeige Modulo 360° werden bei der Absolutpositionsanzeige, z. B. bei positiver Drehrichtung nach 1 Umdrehung +360°, nach 2 Umdrehungen +720° usw., angezeigt. Hier ist der Anzeigebereich entsprechend den Linearachsen von der Steuerung begrenzt.

MD30320 $MA_DISPLAY_IS_MODULO = 1

Hinweis

Für eine Modulo-Achse (MD30310 $MA_ROT_IS_MODULO = 1) sollte auch immer die Positionsanzeige Modulo 360° angewählt sein.

Startposition für die Modulo-Rundachse

Es kann eine Startposition für den Modulo-Bereich ungleich 0 im Maschinendatum festgelegt werden:
MD30340 $MA_MODULO_RANGE_START
Damit ist z. B. ein Modulo-Bereich von -180° bis +180° bei Vorgabe von -180 im MD30340 möglich.

Die Vorbesetzung mit 0 (Grad) gilt für einen Modulo-Bereich von 0° - 360°.

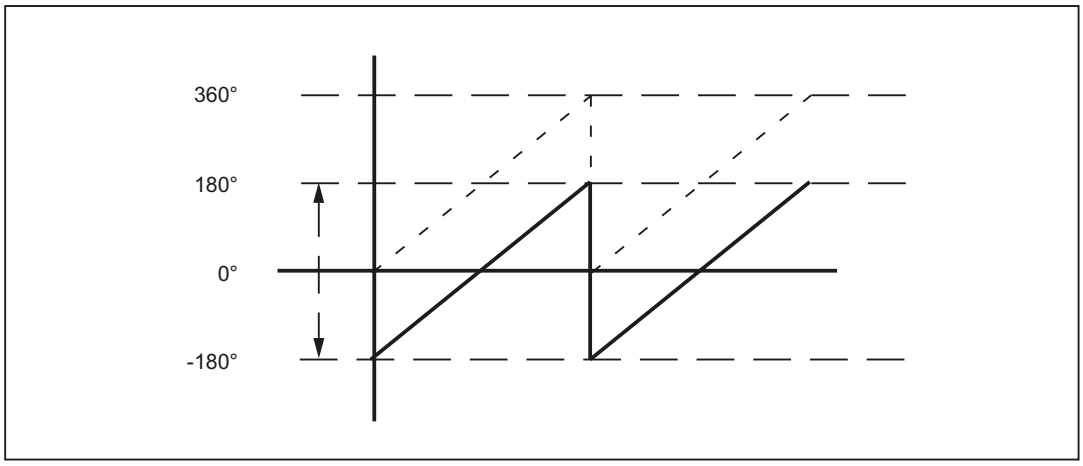


Bild 13-4 Startposition -180° verlegt den Modulo-Bereich nach -180° bis + 180°

Anwendung

Durch die Angleichung der folgenden beiden Maschinendaten können Teilungspositionen von Modulo-Teilungsachsen analog zum Modulo-Bereich realisiert werden:
MD30503 $MA_INDEX_AX_OFFSET
MD30340 $MA_MODULO_RANGE_START

Literatur:
/FB2/ Funktionshandbuch Erweiterungsfunktionen; Teilungsachsen (T1)

13.3 Programmierung von Rundachsen

13.3.1 Allgemeines

Hinweis

Allgemeine Informationen zur Programmierung entnehmen Sie bitte:
Literatur:
/PAG/ Programmieranleitung Grundlagen

MD30310

Mit Hilfe des achsspezifischen Maschinendatums:
MD30310 ROT_IS_MODULO (Modulowandlung für Rundachse)
wird festgelegt, ob sich die Rundachse bei der Programmierung und Positionierung entsprechend wie eine Linearachse verhält oder ob die speziellen Merkmale der Rundachse berücksichtigt werden.

Nachfolgend werden diese Merkmale sowie die Unterschiede (im Wesentlichen bei der Absolutmaßprogrammierung) erläutert.

13.3.2 Rundachse bei aktiver Modulo-Wandlung (endlos drehende Rundachse)

Modulo-Wandlung aktivieren

→ MD30310 $MA_ROT_IS_MODULO = 1 setzen

Hinweis

Bei Modulo-Achsen ist es sinnvoll, auch die Positionsanzeige in Modulo 360º anzuzeigen (MD30320 $MA_DISPLAY_IS_MODULO = 1 setzen).

Absolutmaßprogrammierung (AC, ACP, ACN, G90)

Beispiel für Positionierachse: **POS[Achsname] = ACP(Wert)**

- Der Wert bezeichnet die Zielposition der Rundachse in einem Bereich von 0° bis 359,999°.

  Es sind auch negative Werte möglich, wenn mit den folgenden Maschinendaten eine Bereichsverschiebung vorgenommen wurde.:
  MD30340 $MA_ MODULO_RANGE_START

  und

  MD30330 MA_MODULO_RANGE

- Mit `ACP` (positiv) und `ACN` (negativ) wird die Verfahrrichtung der Rundachse eindeutig festgelegt (unabhängig von der Istposition).
- Bei Programmierung von nur `AC` bzw. mit `G90` ist die Verfahrrichtung abhängig von der Istposition der Rundachse. Ist die Zielposition größer als die Istposition, so fährt die Achse in positiver Drehrichtung, ansonsten in negativer Drehrichtung.

  MD30455 $MA_MISC_FUNCTION_MASK Bit 2
  Das Positionierverhalten ist über das obige Maschinendatum projektierbar:

  Bit 2 = 0: Moduloachse positioniert bei `G90` standardmäßig mit `AC`

  Bit 2 = 1: Moduloachse positioniert bei `G90` standardmäßig mit `DC` (kürzester Weg)
- Anwendung von ACP und ACN: Bei unsymmetrischen Werkstücken muss die Verfahrrichtung definiert vorgegeben werden können, um Kollisionen bei der Drehbewegung auszuschließen.

Beispiel:

Anfangsposition von C ist 0° (siehe folgendes Bild).

①	`POS[C] = ACP(100)`	Rundachse C fährt in positiver Drehrichtung auf Position 100°
②	`POS[C] = ACN(300)`	C fährt in negativer Drehrichtung auf Position 300°
③	`POS[C] = ACP(240)`	C fährt in positiver Drehrichtung auf Position 240°
④	`POS[C] = AC (0)`	C fährt in negativer Drehrichtung auf Position 0°

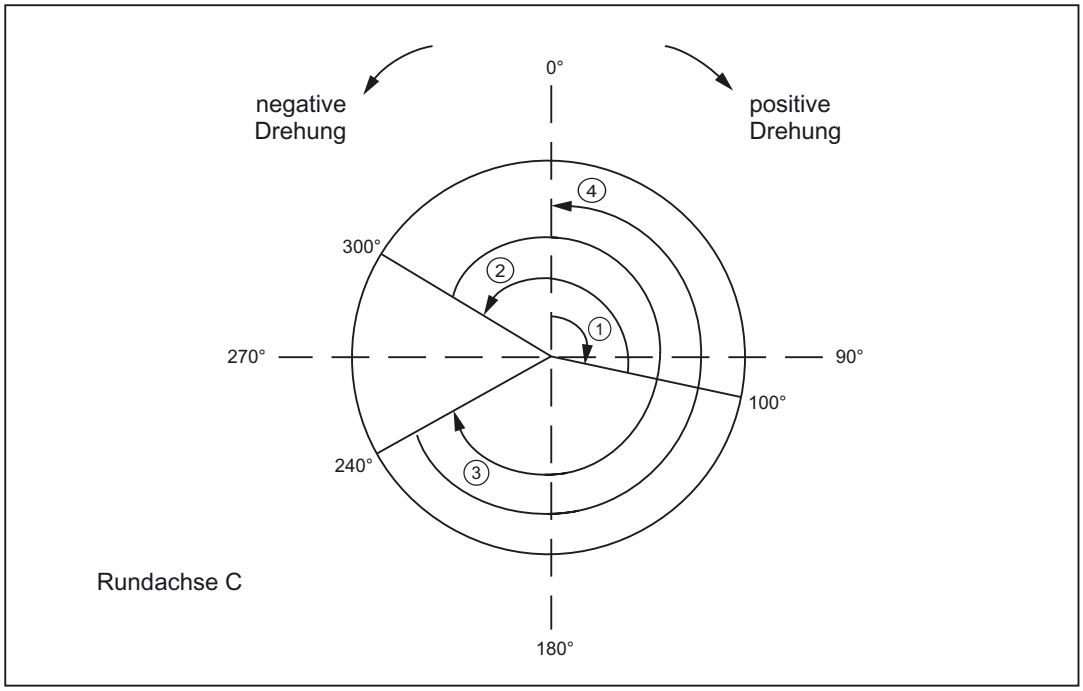


Bild 13-5 Beispiele für Absolutmaßprogrammierung bei Modulo-Achsen

Absolutmaßprogrammierung auf kürzestem Weg (DC)

POS[Achsname] = DC(Wert)

- Der Wert bezeichnet die Zielposition der Rundachse in einem Bereich von 0° bis 359,999°. Bei Werten mit negativem Vorzeichen oder ≥ 360° wird der Alarm 16830 "falsche Modulo-Position programmiert" gemeldet.
- Mit DC (Direct Control) fährt die Rundachse auf **kürzestem Weg** die programmierte absolute Position innerhalb einer Umdrehung an (Verfahrbewegung max. ∓ 180°).
- Abhängig von der aktuellen Istposition ermittelt die Steuerung die Drehrichtung und den Verfahrweg. Ist der abzufahrende Weg in beiden Richtungen gleich (180°), erhält die positive Drehrichtung den Vorzug.
- Anwendungsbeispiel von DC: Der Rundtisch soll in kürzester Zeit (damit kürzesten Weg) die Wechselposition anfahren.
- Wird bei einer Linearachse DC programmiert, so erfolgt die Alarmmeldung 16800 "Verfahranweisung DC kann nicht angewendet werden".

Beispiel:

Anfangsposition von C ist 0º (siehe folgendes Bild)

①	`POS[C] = DC(100)`	C-Achse fährt auf kürzestem Weg auf Position 100°
②	`POS[C] = DC(300)`	C-Achse fährt auf kürzestem Weg auf Position 300°
③	`POS[C] = DC(240)`	C-Achse fährt auf kürzestem Weg auf Position 240°
④	`POS[C] = DC (60)`	C-Achse fährt auf kürzestem Weg auf Position 60°. Da hier der Weg in beiden Richtungen gleich 180° ist, wird die positive Drehrichtung bevorzugt.

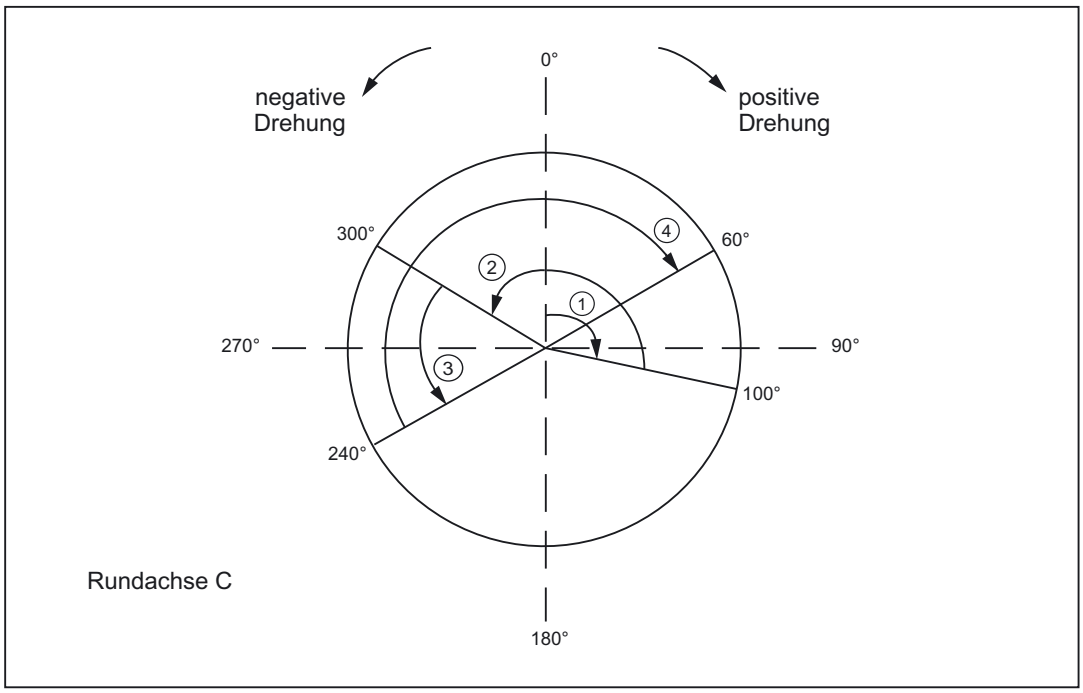


Bild 13-6 Beispiele für DC-Programmierung

Verhalten bei Satzsuchlauf

Nach einem Satzsuchlauf mit Berechnung kann über die Systemvariable:
$AC_RETPOINT
die aufgesammelte Suchlaufposition der Modulo-Wandlung abgefragt werden.

Diese Systemvariable liefert die Position Modulo-gewandelt.

Randbedingungen für ASUP nach Satzsuchlauf mit Berechnung:

In diesem Fall und auch bei den kanalübergreifenden Satzsuchlauf SERUPRO muss die im Satzsuchlauf simulierte Modulo-Wandlung im Teileprogramm durchgeführt werden.

Modulo Rundachse mit/ohne Arbeitsfeldbegrenzung

Es kann durch Setzen des Nahtstellensignals:
DB31, ... DBX12.4
durch die PLC für eine Modulo Rundachse Arbeitsfeldbegrenzung/SW-Endschalter dynamisch ein-/ausgeschaltet werden (analog zu Rundachsen).

Der aktuelle Zustand der Verfahrbereichsbegrenzung wird von der NC mit dem Nahtstellensignal:
DB31, ... DBX74.4
zurückgemeldet.

Die Überwachung wird aktiviert, wenn das Nahtstellensignal:
DB31, ... DBX12.4
durch die PLC gesetzt wurde.
Dem M-/H-Befehl, der die PLC zum Setzen des Nahtstellensignals veranlasst, muss ein `STOPRE` folgen, um durch Synchronisation sicherzustellen, dass nur die Sätze nach der Umschaltung überwacht werden.

Randbedingungen:

Das Ein-/Ausschalten von Software-Endschalter-Überwachung über die PLC-Nahtstelle ist nur für Modulo-Achsen möglich.

Die Verfahrbereichsüberwachung bei Modulo-Achsen setzt immer eine referenzierte Achse und ein aktives Begrenzungspaar voraus.

Bei den Software-Endschaltern ist das immer der Fall, da diese nur paarweise ein-/ausgeschaltet werden. Für die korrekte Überwachung der Arbeitsfeldbegrenzungen müssen **beide** Begrenzungen aktiviert worden sein, entweder über `G26`/`G25` oder die Settingdaten:
SD43400 $SA_ WORKAREA_PLUS_ENABLE

und

SD43410 $SA_WORKAREA_MINUS_ENABLE.

Beispiel einer Umschaltung der Verfahrbereichsbegrenzungen

Auf einer Modulo-Rundachse soll nacheinander eine Palette mit mehreren aufgespannten Werkstücken bearbeitet werden. Dann wird die Palette gegen eine Palette mit einer Aufbauachse eingewechselt, deren Arbeitsbereich überwacht werden muss, um Versorgungsleitungen nicht zu zerstören.

Konfiguration:

MD30300 $MA_IS_ROT_AX[AX4] = 1

MD30310 $MA_ROT_IS_MODULO[AX4] = 1

MD36110 $MA_POS_LIMIT_PLUS[AX4] = 340

MD36100 $MA_POS_LIMIT_MINUS[AX4] = 350

Teileprogrammausschnitt:

```
M123                                              ;  Palette mit Vierfachaufspannung
                                                     mechanisch einlegen
                                                     Software-Endschalter in der B-
                                                     Achse von der PLC aus
                                                     deaktivieren
                                                     DB35, DBX12.4=0
STOPRE                                            ;  Vorlaufstopp auslösen
S1000 M3
G4 F2
G1 X0 Y300 Z500 B0 F5000
CYCLE84(500,400,0,350,0,1,4,10,,0,500,1000) ; Bohrzyklus
Z500
B90
CYCLE84(500,400,0,350,0,1,4,10,,0,500,1000) ; Bohrzyklus
Z500
B180
CYCLE84(500,400,0,350,0,1,4,10,,0,500,1000) ; Bohrzyklus
Z500
B270
CYCLE84(500,400,0,350,0,1,4,10,,0,500,1000) ; Bohrzyklus
Z500
G0 Z540 B0
M124                                              ;  Palette mit Aufbauachse
                                                     mechanisch einlegen
                                                     Software-Endschalter in der B-
                                                     Achse von der PLC aus aktivieren
                                                     DB35, DBX12.4=1
                                                  ;
STOPRE                                            ;  Vorlaufstopp auslösen
B270
```

Kettenmaß-Programmierung (IC, G91)

Beispiel für Positionierachse: **POS[Achsname] = IC(+/-Wert)**

- Der Wert bezeichnet die Verfahrdistanz der Rundachse. Der Wert kann negativ und auch ≥ +/-360° sein.
- Das **Vorzeichen** des Wertes gibt die **Verfahrrichtung** der Rundachse zwingend vor.
- Anwendungsbeispiel: Fräsen einer Spiralnut über mehrere Umdrehungen

Beispiel:

Programmierung	Wirkung
`POS[C] = IC(720)`	C-Achse verfährt inkrementell in positiver Richtung um 720° (2 Umdrehungen)
`POS[C] = IC(-180)`	C-Achse verfährt inkrementell in negativer Richtung um 180°

Endloser Verfahrbereich

Sobald die Modulofunktion aktiv ist, wird der Verfahrbereich nicht begrenzt (Software-Endschalter sind nicht aktiv). Durch entsprechende Programmierung kann die Rundachse endlos verfahren werden.

Beispiel:

```
LOOP:
POS[C] = IC(720)
GOTOB LOOP
```

13.3.3 Rundachse ohne Modulo-Wandlung

Modulo-Wandlung ausschalten

→ MD30310 $MA_ROT_IS_MODULO = 0 setzen

Absolutmaßprogrammierung (AC, G90)

Beispiel für Positionierachse: **POS[Achsname] = AC (+/-Wert)**

- Der Wert und dessen Vorzeichen bezeichnen eindeutig die Zielposition der Rundachse. Der Wert kann auch ≥ +/-360° sein. Der Positionswert wird durch die Software-Endschalter-Positionen begrenzt.
- Die Verfahrrichtung wird von der Steuerung abhängig von der vorzeichenbehafteten Istposition der Rundachse ermittelt.
- Bei Programmierung von `ACP` bzw. `ACN` werden die Alarme 16810 "Verfahranweisung ACP kann nicht angewendet werden" bzw. 16820 "Verfahranweisung ACN kann nicht angewendet werden" gemeldet.
- Anwendungsbeispiel:

  An die Rundachse sind Linearbewegungen gekoppelt (Kurvengetriebe). Daher dürfen bestimmte Endpositionen nicht überfahren werden.

Beispiel:

Programmierung	Wirkung
`POS[C] = AC (-100)`	Rundachse C fährt auf Position -100°; Verfahrrichtung abhängig von der Anfangsposition
`POS[C] = AC (1500)`	Rundachse C fährt auf Position 1500°

Absolutmaßprogrammierung auf kürzestem Weg (DC)

POS[Achsname] = DC(Wert)

Auch wenn die Rundachse nicht als Modulo-Achse definiert ist, kann die Achse mit `DC` (Direct Control) positioniert werden. Dabei ist das Verhalten entsprechend wie als Modulo-Achse.

- Der Wert bezeichnet die Zielposition der Rundachse **in einem Bereich von 0° bis 359,999° (Modulo 360°)**. Bei Werten mit negativem Vorzeichen oder ≥ 360º wird der Alarm 16830 "falsche Modulo-Position programmiert" gemeldet.
- Mit DC (Direct Control) fährt die Rundachse auf **kürzestem Weg** die programmierte absolute Position innerhalb einer Umdrehung an (Verfahrbewegung max. ±180°).
- Abhängig von der aktuellen Istposition (bezogen auf Modulo 360°) ermittelt die Steuerung die Drehrichtung und den Verfahrweg. Ist der abzufahrende Weg in beiden Richtungen gleich (180°), erhält die positive Drehrichtung den Vorzug.
- Anwendungsbeispiel von `DC`: Der Rundtisch soll in kürzester Zeit (damit auf kürzestem Weg) die Wechselposition anfahren.
- Wird bei einer Linearachse `DC` programmiert, so erfolgt die Alarmmeldung 16800 "Verfahranweisung DC kann nicht angewendet werden".

Beispiel:

Programmierung	Wirkung
`POS[C] = AC (7200)`	Rundachse C verfährt auf Position 7200°; Verfahrrichtung abhängig von der Anfangsposition
`POS[C] = DC (300)`	Rundachse C verfährt auf kürzestem Weg die "Modulo"-Position 300° an. Somit verfährt C um 60° mit negativer Drehrichtung und steht auf der absoluten Position 7140°.
`POS[C] = AC (7000)`	Rundachse C verfährt absolut auf Position7000°; hierbei verfährt C um 140° mit negativer Drehrichtung

Hinweis

Bei diesem Beispiel sollte sinnvollerweise auch die Anzeige Modulo 360º (MD30320 $MA_DISPLAY_IS_MODULO = 1) aktiviert sein!

Kettenmaß-Programmierung (IC, G91)

Beispiel für Positionierachse: **POS[Achsname] = IC(+/-Wert)**

Bei Programmierung in Kettenmaß verfährt die Rundachse die identische Strecke wie bei der Moduloachse ab. Der Verfahrbereich ist hier allerdings durch die SW-Endschalter begrenzt.

- Der Wert bezeichnet die Verfahrdistanz der Rundachse.

  Der Wert kann negativ und auch ≥ +/-360° sein.

- Das **Vorzeichen** des Wertes gibt die **Verfahrrichtung** der Rundachse zwingend vor.

Verfahrbereich begrenzt

Der Verfahrbereich ist entsprechend wie bei den Linearachsen begrenzt. Die Bereichsgrenzen werden durch die Software-Endschalter "Plus" und "Minus" festgelegt.

13.3.4 Sonstige Programmiereigenschaften bei Rundachsen

Verschiebungen

`TRANS` (absolut) und `ATRANS` (additiv) sind bei Rundachsen möglich.

Skalierungen

`SCALE` bzw. `ASCALE` sind bei Rundachsen nicht sinnvoll, da die Steuerung immer von einem 360º-Vollkreis die Modulo-Berechnung vornimmt.

Istwertsetzen

`PRESETON` ist möglich.

Teilungsachsen

Literatur:
/FB2/ Funktionshandbuch Erweiterunsfunktionen; Teilungsachsen (T1)

13.4 Inbetriebnahme von Rundachsen

Vorgehen

Die Inbetriebnahme von Rundachsen kann bis auf wenige Ausnahmen entsprechend den Linearachsen erfolgen. Zu beachten ist dabei, dass sobald die Achse als Rundachse definiert ist (MD30300 $MA_IS_ROT_AX = 1), die Einheiten der achsspezifischen Maschinen- und Settingdaten von der Steuerung wie folgt interpretiert werden:

Positionen	in "Grad"
Geschwindigkeiten	in "Umdr./Minute"
Beschleunigungen	in "Umdr./Sekunden²"
Ruckbegrenzung	in "Umdr./Sekunden³"

Spezielle Maschinendaten

Zusätzlich sind je nach Anwendungsfall spezielle Maschinendaten der Rundachse einzugeben:

MD30310 $MA_ROT_IS_MODULO	Modulowandlung für Positionierung und Programmierung
MD30320 $MA_DISPLAY_IS_MODULO	Modulowandlung für Positionsanzeige
MD10210 $MN_INT_INCR_PER_DEG	Rechenfeinheit für Winkelpositionen

In der nachfolgenden Übersicht sind die möglichen Kombinationen dieser Maschinendaten bei einer Rundachse dargestellt:

Kombinationsmöglichkeiten der Maschinendaten der Rundachse				
MD30300	**MD30310**	**MD30320**	**Anwendung zulässig**	**Bemerkung**
0	0	0	ja	Achse ist Linearachse (Standardfall)
1	0	0	ja	Achse ist Rundachse; für die Positionierung erfolgt keine Modulowandlung, d. h. die SW-Endschalter sind aktiv; die Positionsanzeige erfolgt absolut
1	0	1	ja	Achse ist Rundachse; für die Positionierung erfolgt keine Modulowandlung, d. h. die SW-Endschalter sind aktiv; die Positionsanzeige erfolgt Modulo; Anwendung: z. B. für Achsen mit Arbeitsbereich von +/-1000°

Kombinationsmöglichkeiten der Maschinendaten der Rundachse				
1	1	1	ja	Achse ist Rundachse; die Positionierung erfolgt mit Modulowandlung, d. h. die SW-Endschalter sind nicht aktiv, der Arbeitsbereich ist endlos; die Positionsanzeige erfolgt Modulo (häufigste Einstellung bei Rundachsen); die Achse mit/ohne Arbeitsfeldbegrenzung kann benutzt werden
1	1	0	ja	Achse ist Rundachse; die Positionierung erfolgt mit Modulowandlung, d. h. die SW-Endschalter sind nicht aktiv, der Arbeitsbereich ist endlos; die Positionsanzeige erfolgt absolut; die Achse mit/ohne Arbeitsfeldbegrenzung kann benutzt werden
0	0 oder 1	0 oder 1	nicht sinnvoll	Achse ist keine Rundachse; die anderen MD werden daher nicht ausgewertet

JOG-Geschwindigkeit bei Rundachsen

SD41130 $SN_JOG_ROT_AX_SET_VELO (JOG-Geschwindigkeit bei Rundachsen)

Mit dem oben stehenden Settingdatum kann eine für alle Rundachsen gültige JOG-Geschwindigkeit festgelegt werden.

Wird in das Settingdatum der Wert = 0 eingetragen, so wirkt als JOG-Geschwindigkeit für die Rundachse das axiale Maschinendatum:
MD21150 $MC_JOG_VELO (Konventionelle Achsgeschwindigkeit).

Literatur:
/FB2/ Funktionshandbuch Erweiterungsfunktion; Handfahren und Handradfahren (H1)

13.5 Besonderheiten von Rundachsen

Software-Endschalter

Die Software-Endschalter und Arbeitsfeldbegrenzungen sind wirksam und werden bei Schwenkachsen mit einem begrenzten Arbeitsbereich benötigt. Dagegen werden bei endlosdrehenden Rundachsen mit (MD30310 $MA_ROT_IS_MODULO=1) die Software-Endschalter und Arbeitsfeldbegrenzungen achsspezifisch inaktiv geschaltet.

Es kann eine Modulo-Rundachse mit/ohne Arbeitsfeldbegrenzung benutzt werden.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Achsüberwachungen, Schutzbereiche (A3)

Spiegeln von Rundachsen

Mit Programmierung von `MIRROR(C)` bzw. `AMIRROR(C)` ist ein Spiegeln bei Rundachsen möglich.

Referenzpunktfahren

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Referenzpunktfahren (R1)

Spindeln als Rundachsen

Hinweise über die Verwendung von Spindeln als Rundachsen (sog. C-Achsbetrieb) entnehmen Sie bitte:

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Spindeln (S1)

13.6 Beispiele

Gabelkopf, Schrägachsenkopf

Bei 5-Achsen-Fräsmaschinen werden sehr häufig Rundachsen zur Schwenkbewegung der Werkzeugachse oder zur Drehbewegung des Werkstücks verwendet. Diese Maschinen können die Spitze eines Werkzeuges an jeden beliebigen Punkt des Werkstückes positionieren und dabei eine beliebige Lage der Werkzeugachse zulassen. Je nach Anwendungsfall werden dazu unterschiedliche Fräsköpfe benötigt. Im Bild ist der Gabelkopf und der Schrägachsenkopf als Beispiele für Anordnungen von Rundachsen dargestellt.

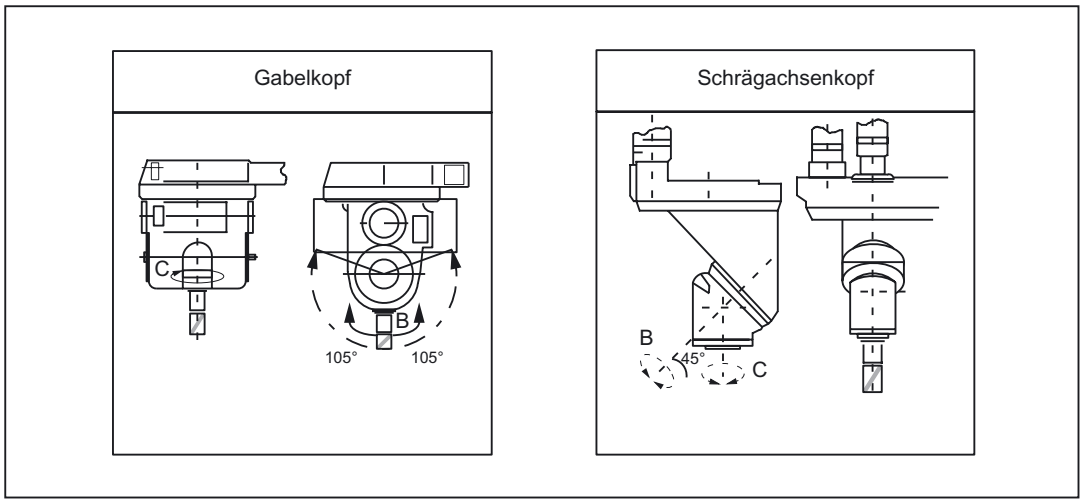


Bild 13-7 Gabelkopf, Schrägachsenkopf

13.7 Datenlisten

13.7.1 Maschinendaten

13.7.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
10210	INT_INCR_PER_DEG	Rechenfeinheit für Winkelpositionen

13.7.1.2 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: $MA_	Beschreibung
30300	IS_ROT_AX	Achse ist Rundachse
30310	ROT_IS_MODULO	Modulo-Wandlung für Rundachse
30320	DISPLAY_IS_MODULO	Istwertanzeige Modulo
30330	MODULO_RANGE	Größe des Modulo-Bereichs
30340	MODULO_RANGE_START	Anfangsposition für den Modulo-Bereich
30455	MISC_FUNCTION_MASK	Achsfunktionen
36100	POS_LIMIT_MINUS	Softwareendschalter minus
36110	POS_LIMIT_PLUS	Softwareendschalter plus

13.7.2 Settingdaten

13.7.2.1 Allgemeine Settingdaten

Nummer	Bezeichner: $SN_	Beschreibung
41130	JOG_ROT_AX_SET_VELO	JOG-Geschwindigkeit bei Rundachsen

13.7.2.2 Achs-/Spindel-spezifische Settingdaten

Nummer	Bezeichner: $SA_	Beschreibung
43420	WORKAREA_LIMIT_PLUS	Arbeitsfeldbegrenzung plus
43430	WORKAREA_LIMIT_MINUS	Arbeitsfeldbegrenzung minus

13.7.3 Signale

13.7.3.1 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Verfahrbereichsbegrenzung bei Modulo-Achse	DB31, ...DBX12.4	DB380x.DBX1000.4

13.7.3.2 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Zustand der SW-Endschalterüberwachung bei Modulo-Achse	DB31, ...DBX74.4	DB390x.DBX1000.4

S3: Synchronspindel

14

14.1 Kurzbeschreibung

14.1.1 Funktion

Durch die Funktion "Synchronspindel" können 2 Spindeln lage- oder drehzahlsynchron gekoppelt werden. Eine Spindel ist dabei als Leitspindel (LS) zu definieren, die zweite Spindel ist dann Folgespindel (FS).

Drehzahl-Synchronität:	$n_{FS} = k_Ü * n_{LS}$, mit $k_Ü$ = „1, „2, „3, ...
Lagesynchronität:	$\varphi_{FS} = \varphi_{LS} + \Delta\varphi$, mit 0^o x $\Delta\varphi$ t=360^o

Anwendungsmöglichkeiten

Rückseitenbearbeitung

Eine Anwendungsmöglichkeit ist z. B. die Rückseitenbearbeitung innerhalb einer Doppelspindel-Drehmaschine durch fliegende Übergabe des Werkstücks von lagesynchroner LS auf FL (oder umgekehrt), ohne diese dazu bis zum Stillstand abbremsen zu müssen.

Mehrkantbearbeitung (Polygondrehen)

Die Funktion "Synchronspindel" bietet durch Vorgabe eines ganzzahligen Übersetzungsverhältnisses $k_Ü$ zwischen LS und FS die Voraussetzung zur Mehrkantbearbeitung (Polygondrehen).

FS Anzahl

Die Anzahl von FS, die synchron zu einer LS betrieben werden können, ist nur durch die Leistungsfähigkeit der verwendeten NC beschränkt. Prinzipiell können beliebig viele FS in beliebigen Kanälen der NC gleichzeitig auf eine LS gekoppelt werden.

Pro NC-Kanal können gleichzeitig bis zu 2 Synchronspindelpaare aktiv sein.

Definition

Die Zuordnung von FS zu LS Synchronspindelpaar kann über Maschinendaten kanalspezifisch parametriert oder über Teileprogrammbefehle flexibel definiert werden.

An-/Abwahl

Die An-/Abwahl des Synchronbetriebes eines Synchronspindelpaares erfolgt über Teileprogrammbefehle.

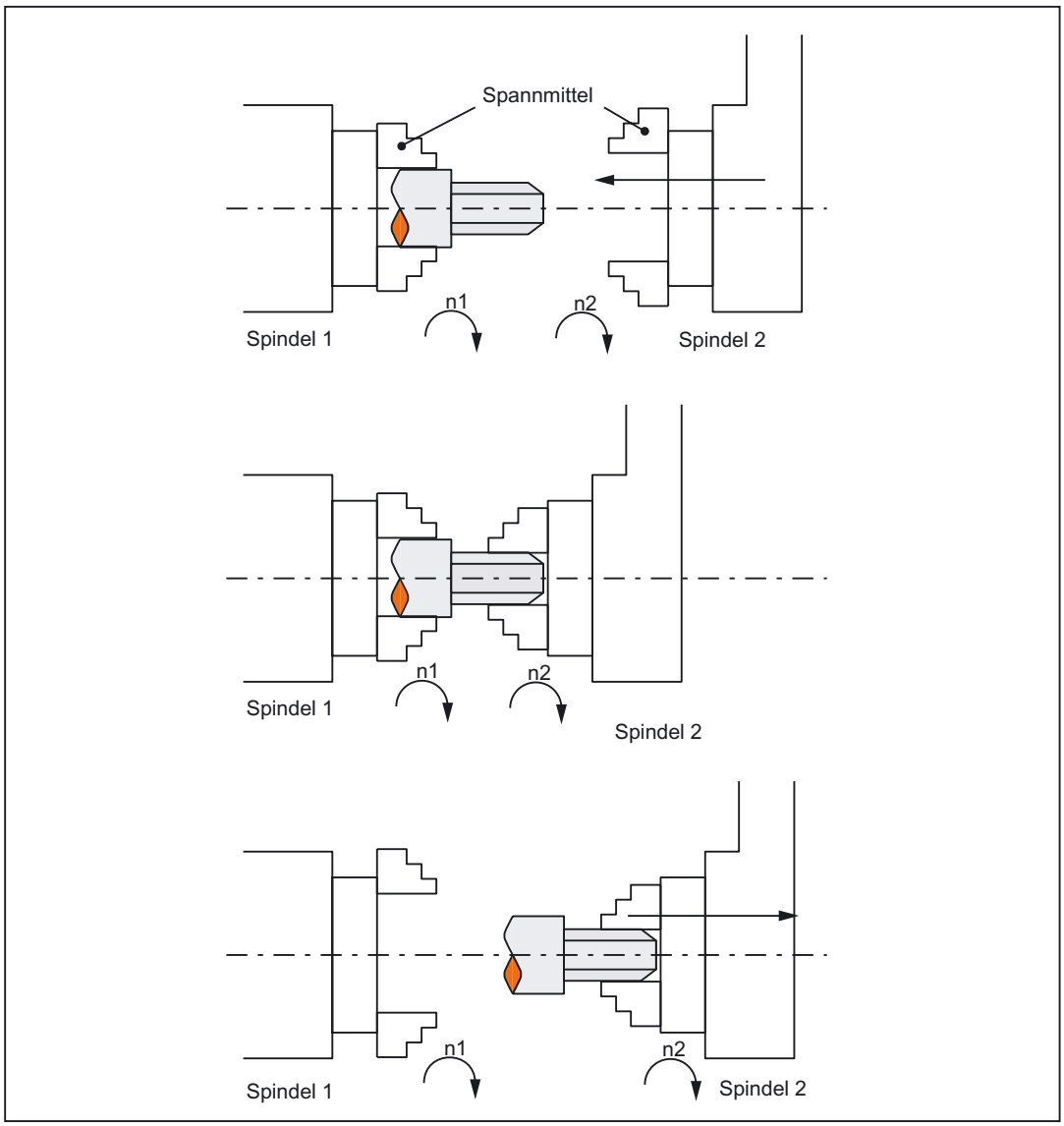


Bild 14-1 Synchronbetrieb: Fliegende Werkstückübergabe von Spindel 1 auf Spindel 2

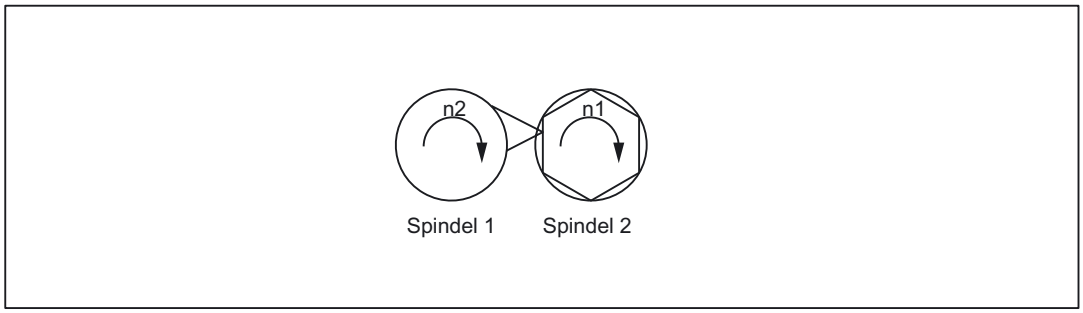


Bild 14-2 Synchronbetrieb: Mehrkantdrehen

14.1.2 Voraussetzungen

Für die Nutzung der Funktion wird die Option "Synchronspindel/Mehrkantdrehen" oder die entsprechende optionale Ausführung der Generischen Kopplung benötigt.

Informationen zu den verschiedenen Ausführungen der Generischen Kopplung finden sich in:
Literatur:
Funktionshandbuch Sonderfunktionen; Achskopplungen (M3)

14.1.3 Synchronbetrieb

Erläuterungen

<axialer Ausdruck>:	kann sein: - Achsbezeichner - Spindelbezeichner
<Achsbezeichner>:	C (Wenn die Spindel im Achsbetrieb den Bezeichner "C" hat.)
<Spindelbezeichner>:	Sn, SPI(n) mit n = Spindelnummer
<Spindelnummer>:	1, 2, ... entsprechend der im MD35000 $MA_SPIND_ASSIGN_TO_MACHAX festgelegten Spindelnummer
(FS, LS, Offset):	LS = Leitspindel, FS = Folgespindel, Offset = programmierbarer Offset der Folgespindel mit Systemvariablen lesen
$P_COUP_OFFS[Sn]	Programmierter Positionsoffset der Synchronspindel

Synchronspindelpaar

Im Synchronbetrieb gibt es eine Folgespindel (FS) und eine Leitspindel (LS), das sog. **Synchronspindelpaar**. Die Folgespindel folgt bei aktiver Kopplung (Synchronbetrieb) den Bewegungen der Leitspindel entsprechend dem festgelegten Funktionszusammenhang.

Synchronbetrieb

Der Synchronbetrieb (auch Synchronspindelbetrieb genannt) ist eine weitere Spindelbetriebsart. Vor Aktivierung des Synchronbetriebs muss die Folgespindel in Lageregelung überführt sein. Mit Einschalten der Kopplung wird bei der Folgespindel der Synchronbetrieb aktiv. Nach Aufhebung der Kopplung wird bei der Folgespindel der Steuerbetrieb wirksam.

Sobald für die Folgespindel der Synchronbetrieb aktiv ist, wird folgendes Nahtstellensignal an die PLC gemeldet:

NST "Synchronbetrieb" (DB31, ... DBX84.4) = 1.

Synchronspindel-Anzahl

Es besteht die Möglichkeit, mehrere Folgespindeln an ein und dieselbe Leitspindel anzukoppeln. Die Anzahl der Folgespindeln an dieser Leitspindel hängt von den jeweiligen Ausprägungen der entsprechenden Softwarestände ab.

Es können beliebig viele Folgespindeln in beliebigen Kanälen einer NCU oder einer anderen NCU an diese Leitspindel angekoppelt werden.

Dabei ist zu beachten, dass eine Spindel immer der Master ist und die Anzahl der Kopplungen sich aus der Anzahl der Achsen minus den Master ergibt.

Möglichkeiten des Synchronbetriebes

Folgende Funktionsmöglichkeiten sind für den Synchronbetrieb gegeben:

- FS und LS drehen mit gleicher Drehzahl

  ($n_{FS} = n_{LS}$; Übersetzungsverhältnis $k_Ü = 1$)

- Gleichsinniger bzw. gegensinniger Lauf zwischen FS und LS

  (über Übersetzungsverhältnis $k_Ü$ positiv bzw. negativ festlegbar)

- Folge- und Leitspindel drehen mit unterschiedlichen Drehzahlen

  ($n_{FS} = k_Ü \cdot n_{LS}$; Übersetzungsverhältnis $k_Ü \neq 1$)

  Anwendung: Mehrkantdrehen

- Einstellbare Winkellage zwischen FS und LS ($\varphi_{FS} = \varphi_{LS} + \Delta\varphi$)

  Die Spindeln laufen drehzahlsynchron mit definiertem Winkelversatz zwischen FS und LS (positionssynchrone Kopplung).

  Anwendung: profilierte Werkstücke

- Einschalten des Synchronbetriebs zwischen FS und LS kann jeweils in der Bewegung oder im Stillstand erfolgen.
- Für die Leitspindel steht die gesamte Funktionalität von Steuer- und Positionierbetrieb zur Verfügung.
- Bei nicht aktivem Synchronbetrieb können FS und LS in allen Spindelbetriebsarten verfahren werden.

- Das Übersetzungsverhältnis kann auch während aktivem Synchronbetrieb in der Bewegung verändert werden.
- Bei eingeschalteter Synchronspindelkopplung kann der Offset der FS zur LS (überlagerte Bewegung) verändert werden.

Möglichkeiten der Kopplung

Die Festlegung der Synchronspindelkopplungen kann sowohl

- fest projektiert über kanalspezifische Maschinendaten

  (nachfolgend als **"fest projektierte Kopplung"** bezeichnet) als auch
- frei definiert über Sprachanweisungen (COUP...) im Teileprogramm

  (nachfolgend als **"Anwenderdefinierte Kopplung"** bezeichnet)

erfolgen. Dabei sind folgende Varianten möglich:

1. Eine Kopplung ist fest über Maschinendaten projektiert. Eine zweite Kopplung kann zusätzlich durch das Teileprogramm frei definiert werden.
2. Es ist keine Kopplung über Maschinendaten projektiert. Damit können durch das Teileprogramm die Kopplungen anwenderdefiniert und parametriert werden.

Eigener Folgespindelinterpolator

Der eigene **Folgespindelinterpolator** ermöglicht mehrere Folgespindeln aus verschiedenen Kanälen oder einer anderen NCU, an eine einzige Leitspindel anwenderdefiniert zu koppeln. Der Folgespindelinterpolator wird durch

- `COUPON` oder `COUPONC` aktiviert und
- `COUPOF` oder `COUPOFS` deaktiviert

und befindet sich immer in dem Kanal, in dem `COUPON`, `COUPONC` für die Folgespindel programmiert wird. Falls die einzuschaltenende Folgespindel vorher in einem anderen Kanal programmiert wurde, erzeugt `COUPON`/`COUPONC` einen Achstausch und holt die Spindel in diesen Kanal.

Durch kopplungsspezifische axiale VDI-Nahtstellensignale lassen sich vom PLC bestimmte Synchronspindelfunktionen beeinflussen. Diese Signale wirken ausschließlich auf die Folgespindeln und sind für die Leitspindel wirkungslos. Weitere Informationen hierzu siehe Kapitel "Synchronspindelkopplung durch PLC beeinflussen".

Festlegung der Synchronspindeln

Vor dem Aktivieren des Synchronbetriebs müssen die zu koppelnden Spindeln (FS, LS) festgelegt werden.

Dies kann je nach Anwendungsfall auf zwei unterschiedliche Arten erfolgen:

1. Fest projektierte Kopplung:

   Maschinenachsen, die als Folgespindel (FS) und als Leitspindel (LS) wirken sollen, werden mit dem kanalspezifischen MD21300 $MC_COUPLE_AXIS_1[n] festgelegt.

   Bei dieser projektierten Kopplung können die festgelegten Maschinenachsen für FS und LS vom NC-Teileprogramm nicht verändert werden.

   Die Kopplungsparameter können ggf. durch das NC-Teileprogramm geändert werden.

2. Anwenderdefinierte Kopplung:

   Mit der Sprachanweisung "`COUPDEF`(FS, LS, **...**)" können Kopplungen im NC-Teileprogramm neu angelegt und verändert werden. Wenn eine neue Kopplungsbeziehung definiert werden soll, muss ggf. zuvor eine bestehende anwenderdefinierte Kopplung gelöscht werden (mit Sprachanweisung `COUPDEL`(FS, LS)).

   Mit FS und LS sind die Spindelbezeichner (Sn, SPI(n)) für die Folge- und Leitspindel bei jeder Sprachanweisung COUP**...** zu programmieren. Damit ist die Synchronspindelkopplung eindeutig bestimmt.

   Die gültige Spindelnummer muss in folgendem achsspezifischen Maschinendatum einer Maschinenachse zugeordnet sein:

   MD35000 $MA_SPIND_ASSIGN_TO_MACHAX.

An die PLC wird mit den NST "FS aktiv" (DB31, ... DBX99.1) und NST "LS aktiv" (DB31, ... DBX99.0) für die jeweilige Maschinenachse gemeldet, ob diese als Folge- oder Leitspindel wirksam ist.

Die LS kann wahlweise über Teileprogramm, PLC und auch über Synchronaktionen programmiert werden.

Übersetzungsverhältnis

Das Übersetzungsverhältnis wird mit getrennten Zahlenwerten für Zähler und Nenner (Übersetzungsparameter) vorgegeben. Dies ermöglicht eine sehr genaue Vorgabe für das Übersetzungsverhältnis auch bei rationalen Zahlen.

Allgemein gilt:

$k_Ü$ = Übersetzungsparameter Zähler : Übersetzungsparameter Nenner= $Ü_{Zähler} : Ü_{Nenner}$

Der Wertebereich des Übersetzungsparameter ($Ü_{Zähler}$, $Ü_{Nenner}$) ist steuerungsintern nahezu unbegrenzt.

Für die über Maschinendaten projektierte Kopplung können die Übersetzungsparameter mit Hilfe von kanalspezifischen SD 42300: COUPLE_RATIO_1[n] festgelegt werden. Darüber hinaus kann das Übersetzungsverhältnis mit der Sprachanweisung `COUPDEF`(FS, LS, $Ü_{Zähler}$, $Ü_{Nenner}$,**...**) verändert werden. Die im Settingdatum eingetragenen Werte werden dabei nicht überschrieben (Default-Werte).

Für die über das NC-Teileprogramm definierte Kopplung ist das Übersetzungsverhältnis nur mit der Sprachanweisung `COUPDEF` (**...**) vorgebbar.

Die neuen Übersetzungsparameter werden sofort nach Bearbeitung der `COUPDEF`-Anweisung wirksam.

Weitere Programmieranweisungen für Synchronspindelkopplungen entnehmen Sie bitte dem Kapitel "Programmierung von Synchronspindelkopplungen".

Kopplungseigenschaften

Für jede Synchronspindelkopplung können folgende Eigenschaften festgelegt werden:

- **Satzwechselverhalten**

  Beim Einschalten des Synchronbetriebs oder beim Ändern des Übersetzungsverhältnisses bzw. des definierten Winkelversatzes bei aktiver Kopplung kann festgelegt werden, wann das Einwechseln des nächsten Satzes erfolgen soll:

  – Satzwechsel erfolgt sofort
  – Satzwechsel bei "Synchronlauf fein"
  – Satzwechsel bei "Synchronlauf grob"
  – Satzwechsel bei IPOSTOP (d. h. nach sollwertseitigem Synchronlauf)
  – Prüfung der Synchronlaufbedingungen zum beliebigen Zeitpunkt mit `WAITC`.

- **Kopplungsart** zwischen FS und LS

  Als Führungsgröße für die Folgespindel kann entweder der Lagesollwert oder der Lageistwert der Leitspindel verwendet werden. Daher sind folgende Kopplungsarten wählbar:

  – Sollwertkopplung (DV)

    Anwendung im lagegeregelten Betrieb. Dabei sollte die Regeldynamik beider Spindeln nahezu übereinstimmen. Die Sollwertkopplung sollte vorzugsweise Anwendung finden.

  – Istwertkopplung (AV)

    Anwendung wenn keine Lageregelung der LS möglich ist oder bei starker Abweichung der Regeleigenschaften zwischen FS und LS. Die Sollwerte für die FS werden von den Istwerten der LS abgeleitet. Der Synchronlauf besitzt bei sich ändernder Drehzahl der LS eine schlechtere Güte als bei der Sollwertkopplung.

  – Geschwindigkeitskopplung (VV)

    Die Geschwindigkeitskopplung ist intern eine Sollwertkopplung. Es werden geringere Anforderungen an die FS und LS gestellt. Lageregelung und Messsysteme sind für FS und LS nicht erforderlich.

    Der Positionsoffset zwischen FS und LS ist undefiniert.

Die Auswahl der jeweiligen Kopplungseigenschaft erfolgt bei der **projektierten Kopplung** über Maschinendaten, siehe Kapitel "Projektierung eines Synchronspindelpaares über Maschinendaten" und bei der **anwenderdefinierten Kopplung** über die Sprachanweisung `COUPDEF`, siehe Kapitel "vorbereitende Programmieranweisungen".

Darüber hinaus können die Kopplungseigenschaften Kopplungsart und Satzwechselverhalten bei der fest projektierten Kopplung über die Sprachanweisung `COUPDEF` verändert werden.

Literatur:

/PGA/ Programmierhandbuch Arbeitsvorbereitung ("Synchronspindel").

Änderungsschutz für Kopplungseigenschaften

Mit dem kanalspezifischen MD21340 $MC_COUPLE_IS_WRITE_PROT_1 wird festgelegt, ob die fest projektierten Kopplungsparameter Übersetzungsverhältnis, Kopplungsart und Satzwechselverhalten vom NC-Teileprogramm beeinflussbar sind:

0: Kopplungsparameter sind vom NC-Teileprogramm durch `COUPDEF` veränderbar

1: Kopplungsparameter sind vom NC-Teileprogramm nicht veränderbar Änderungsversuche werden mit einer Alarmmeldung abgewiesen.

Überlagernde Bewegung

Bei aktivem Synchronbetrieb folgt die Synchronspindel der Bewegung der Leitspindel entsprechend dem vorgegebenen Übersetzungsverhältnis.

Gleichzeitig kann die Synchronspindel zusätzlich überlagert verfahren werden, um die definierte Winkellage der FS zur LS ändern zu können.

Die überlagernde Verfahrbewegung der FS kann auf unterschiedliche Weise ausgelöst werden:

- Programmierbare Lageverschiebung der FS bei AUTOMATIK und MDA:
  - Mit den Sprachanweisungen `COUPON` und `SPOS` kann bei aktivem Synchronbetrieb der Lagebezug zwischen FS und LS verändert werden, siehe Kapitel "Anwahl des Synchronbetriebes vom Teileprogramm."
- Manuelle Lageverschiebung der FS:
  - in der Betriebsart JOG (JOG-kontinuierlich oder JOG-inkrementell)

    Überlagerung der FS mit Handrad oder mit Verfahrtasten plus bzw. minus bei aktivem Synchronbetrieb.
  - in der Betriebsart AUTOMATIK und MDA

    Überlagerung der FS mit Handrad über DRF-Verschiebung

Sobald die FS die überlagernde Verfahrbewegung ausführt, wird das NST "überlagerte Bewegung" (DB31, ... DBX98.4) auf 1-Signal gesetzt.

Die überlagernde Verfahrbewegung erfolgt bei `COUPON` zeitoptimal mit der noch maximal möglichen Geschwindigkeit der FS. Bei der Offsetänderung durch SPOS kann die Positioniergeschwindigkeit mit FA[Sn] vorgegeben und mit Override (Anwahlmöglichkeit durch das NST "Vorschubkorrektur für Spindel gültig "DB31, ... DBX17.0) beeinflusst werden.

Hinweis

Weitere Informationen über das Vorgeben der Positioniergeschwindigkeit mit FA[Sn] in:

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; Spindeln (S1), Kapitel Spindelbetriebsart Positionierbetrieb

---

Sollwertkorrektur

Die Sollwertkorrektur der Systemvariable $AA_COUP_CORR[Sn] wirkt sich auf alle nachfolgenden Programmierungen der Folgespindel wie eine Positionsverschiebung aus und entspricht einer DRF-Verschiebung im MKS.

Beispiel Korrekturwert ermitteln

Wurde mittels COUPON(....,77) ein Kopplungsoffset von 7° programmiert und ist durch das Schließen des Werkstückaufnahmevorrichtung ein mechanischer Kopplungsoffset von 81° entstanden, so wird ein Korrekturwert von 4° ermittelt:

Die Systemvariablen liefern für die Folgespindel folgende Werte:

$P_COUP_OFFS[S2] ; programmierter Positionsoffset = 77°

$AA_COUP_OFFS[S2] ; sollwertseitiger Positionsoffset = 77°

$VA_COUP_OFFS[S2] ; istwertseitiger Positionsoffset ca. 77°

$AA_COUP_CORR[S2] ; Korrekturwert = 4°

14.1.4 Voraussetzungen für Synchronbetrieb

Voraussetzungen bei Anwahl des Synchronbetriebs

Bevor die Synchronspindelkopplung eingeschaltet wird, müssen folgende Voraussetzungen vorliegen, andernfalls werden Alarmmeldungen generiert.

- Die Synchronspindelkopplung muss zuvor festgelegt sein (entweder über Maschinendaten fest projektiert oder über das Teileprogramm anwenderdefiniert mittels `COUPDEF`).
- Die zu koppelnden Spindeln müssen in dem NC-Kanal, in dem die Kopplung eingeschaltet wird, bekannt sein.

  kanalspez. MD20070 $MC_AXCONF_MACHAX_USED

  achsspez. MD35000 $MA_SPIND_ASSIGN_TO_MACHAX
- Die Folgespindel muss dem NC-Kanal, in dem die Kopplung eingeschaltet wird, zugeordnet sein.

  Defaulteinstellung mit achsspez. MD30550 AXCONF_ASSIGN_MASTER_CHAN
- Bei Soll- und Istwerkopplung (DV, AV) gilt:

  FS und LS müssen zur Positionserfassung mindestens ein Lagemesssystem besitzen und die Lageregelungen muss in Betrieb genommen sein.

Hinweis

Die maximale Solldrehzahl der LS wird bei eingeschalteter Lageregelung automatisch auf 90% (Regelreserve) der Maximaldrehzahl begrenzt. Die Begrenzung wird über das NST "Solldrehzahl begrenzt" (DB31, ... DBX83.1) gemeldet.

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; Spindeln (S1)

- Bei Sollwerkopplung DV gilt:

  Die LS sollte sich zur Erzielung eines besseren Sychronisationsverhaltens vor dem Einschalten der Kopplung in Lageregelung befinden (Sprachanweisung SPCON).

- Vor Anwahl des Synchronbetriebs muss die erforderliche Getriebestufe für FS und LS angewählt sein. Im Synchronbetrieb ist keine Getriebestufenumschaltung und daher auch kein Pendelbetrieb für FS und LS möglich. Bei entsprechender Anforderung wird eine Alarmmeldung generiert.

Kanalübergreifende Kopplung

Die LS kann sich in einem beliebigen Kanal befinden.

- Die LS kann mittels "Achstausch" zwischen Kanälen getauscht werden.
- Bei mehreren Folgespindeln an einer Leitspindel wird die Dynamik eines Koppelverbandes koppelfaktorabhängig von der schwächsten Dynamik bestimmt. Das Beschleunigungsvermögen und die maximale Drehzahl werden für die Leitspindel soweit reduziert, dass keine der angekoppelten Folgespindeln überlastet werden kann.
- Die Folgespindel wird immer in dem Kanal behandelt, in dem die Kopplung mit COUPON oder COUPONC eingeschaltet wurde.

14.1.5 Anwahl des Synchronbetriebes vom Teileprogramm

Kopplung einschalten COUPON, COUPONC

Mit der Sprachanweisung COUPON wird im Teileprogramm die Kopplung zwischen den angegebenen Spindeln mit den zuletzt gültigen Parametern eingeschaltet und damit der Synchronbetrieb aktiviert. Diese Kopplung kann fest projektiert oder anwenderdefiniert sein. Die Leitspindel und/oder die Folgespindel können sich dabei im Stillstand oder in der Bewegung befinden.

Vor der Aktivierung des Synchronbetriebs müssen bestimmte Voraussetzungen vorliegen, siehe Kapitel "Voraussetzungen für Synchronbetrieb".

Mit der Anweisung COUPONC wird für die Folge- und Leitspindel die im Teileprogramm vorhergehende programmierte Spindeldrehrichtung und Spindeldrehzahl übernommen. Die Angabe eines Winkelversatzes ist nicht möglich.

COUPON Einschaltvarianten

Für die Aktivierung des Synchronbetriebs sind zwei Varianten wählbar:

1. Schnellstmögliches Einschalten der Kopplung mit **beliebigemWinkelbezug** zwischen Leit- und Folgespindel.

   COUPON(FS, LS)

2. Einschalten der Kopplung mit einem **definierten Winkelversatz** POS_{FS} zwischen Leit- und Folgespindel. Bei dieser Variante ist der Winkelversatz bei der Anwahl mit zu programmieren.

   COUPON(FS, LS, POS_{FS})

Hinweis

Befindet sich die LS und/oder FS vor dem Einschalten der Synchronkopplung im Achsbetrieb, dann wird mit der Verwendung des Spindelbezeichners der Achsbetrieb verlassen und Spindelbetrieb aktiviert.

Erfolgt das Einschalten der Spindel unter Verwendung des Achsbezeichners, dann findet keine Umschaltung statt.

Satzwechselverhalten

Vor Anwahl des Synchronbetriebs ist das Verhalten festzulegen, unter welchen Kriterien der Satzwechsel beim Einschaltvorgang erfolgen soll, siehe Kapitel "Vorbereitende Programmieranweisungen".

Aktuellen Kopplungszustand feststellen

Mit der axialen Systemvariablen $AA_COUP_ACT[<axialer Ausdruck >], siehe Kapitel "Axiale Systemvariablen für Synchronspindeln" kann im NC-Teileprogramm für die angegebene Achse/Spindel der aktuelle Kopplungszustand festgestellt werden. Sobald für die Folgespindel die Synchronspindelkopplung aktiv ist, wird für das Bit 2 "1" gelesen.

Definierten Winkelversatz ändern

Mit den Sprachanweisungen `COUPON` und `SPOS` kann bei aktivem Synchronbetrieb der definierte Winkelversatz verändert werden. Die Folgespindel positioniert als überlagernde Bewegung auf den mit POS_{FS} programmierten Winkelversatz. Während dieses Zeitraumes wird das NST "überlagerte Bewegung" (DB31, ... DBX98.4) gesetzt.

Winkelversatz POS_{FS}

Der definierte Winkelversatz POS_{FS} ist als Absolutposition bezogen auf die Null-Grad-Position der Leitspindel in positiver Drehrichtung anzugeben.

Die "0°-Position" einer lagegeregelten Spindel ergibt sich folgendermaßen:

- aus dem Nullmarken- bzw. Bero-Signal des Messsystems und
- aus den über achsspezifischen Maschinendaten hinterlegten Bezugswerten:

  MD34100 $MA_REFP_SET_POS, Referenzpunktwert,

  bei abstandskodiertem System bedeutungslos.

  MD34080 $MA_REFP_MOVE_DIST Referenzpunktabstand/Zielpunkt

  bei abstandskodiertem System,

  MD34090 $MA_REFP_MOVE_DIST_CORR
  Referenzpunktverschiebung/Absolutverschiebung abstandscodiert.

Bereich von POS_{FS}: 0 ... 359,999°.

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; Referenzpunktfahren (R1)

Aktuellen Winkelversatz lesen

Mit Hilfe von axialen Systemvariablen kann im NC-Teileprogramm der aktuelle Positionsoffset der FS zur LS gelesen werden. Dabei unterscheidet man:

- aktueller Positionsoffset des Sollwertes der FS zur LS

  $AA_COUP_OFFS [<Achsbezeichner für FS>]

- aktueller Positionsoffset des Istwertes der FS zur LS

  $VA_COUP_OFFS [<Achsbezeichner für FS>]

(Erläuterung zu <Achsbezeichner>, siehe Kapitel "Synchronbetrieb")

Aktivierung nach POWER ON

Der Synchronbetrieb kann auch bei nicht referenzierter/synchronisierter FS oder LS (NST "Referenziert/Synchronisiert 1 bzw.2" DB31, ... DBX60.4 bzw. DBX60.5 = 0) aktiviert werden. In diesem Fall wird eine Warnung gemeldet.

Beispiel:

FS und LS sind nach Power ON bereits mit einem Werkstück kraftschlüssig miteinander gekoppelt.

14.1.6 Abwahl des Synchronbetriebes vom Teileprogramm

Kopplung ausschalten COUPOF, COUPOFS

Der Synchronbetrieb zwischen den angegebenen Spindeln wird mit der Teileprogrammanweisung `COUPOF` aufgehoben. Es sind drei Varianten möglich.

Wird mit `COUPOF` der Synchronbetrieb zwischen den angegebenen Spindeln aufgehoben, dann ist es unerheblich, ob diese Kopplung fest projektiert oder anwenderdefiniert ist. Die Folgespindel und die Leitspindel können sich dabei im Stillstand oder in der Bewegung befinden.

Beim Ausschalten des Synchronbetriebs mit `COUPOF` wird die Folgespindel in den **Steuerbetrieb** geschaltet. Das ursprünglich programmierte S-Wort ist für die FS nicht mehr gültig, die Folgespindel kann wieder wie eine normale Spindel betrieben werden.

Beim Ausschalten der Kopplung mit `COUPOF` wird steuerungsintern ein Satzvorlauf-Stop analog `STOPRE` ausgelöst.

Mit der Anweisung `COUPOFS` kann eine Kopplung entweder schnellstmöglich mit Stop ohne Positionsangabe oder mit Stop auf der programmierten Position ausgeschaltet werden.

COUPOF Varianten

Für die Abwahl mit `COUPOF` des Synchronbetriebs sind drei Varianten wählbar:

1. Schnellstmögliches Ausschalten der Kopplung.

   Der Satzwechsel wird sofort freigegeben.

   COUPOF(FS, LS)

2. Abwahl der Kopplung erfolgt erst, nach dem die Folgespindel die programmierte Ausschaltposition POS_{FS} überfahren hat.

   Anschließend wird der Satzwechsel freigegeben.

   COUPOF(FS, LS, POS_{FS})

3. Abwahl der Kopplung erfolgt erst, nach dem die Folgespindel und die Leitspindel die programmierten Ausschaltpositionen POS_{FS} und POS_{LS} überfahren haben.

   Anschließend wird der Satzwechsel freigegeben.

   COUPOF(FS, LS, POS_{FS} , POS_{LS})

Hinweis

Befindet sich die LS und/oder FS vor dem Ausschalten der Synchronkopplung im Achsbetrieb, dann wird mit der Verwendung des Spindelbezeichners der Achsbetrieb verlassen und im Spindelbetrieb der Drehzahlsteuerbetrieb aktiviert.

Erfolgt das Ausschalten der Spindel unter Verwendung des Achsbezeichners, dann findet keine Umschaltung statt. Die LS muss sich dann vor dem Ausschalten im sollwertseitigen Stillstand befinden.

POS_{FS}, POS_{LS}

Die Ausschaltpositionen POS_{FS} bzw. POS_{LS} entsprechen den Istpositionen von FS bzw. LS bezogen auf den festgelegten Referenzpunktwert.

Bereich von POS_{FS}, POS_{LS}: 0 ... 359,999°.

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; Referenzpunktfahren (R1)

COUPOF während der Bewegung

Wird der Synchronbetrieb während der Bewegung mit `COUPOF` abgewählt, dreht die Folgespindel mit der aktuellen Drehzahl (n_{FS}) weiter. Die Drehzahl kann im NC-Teileprogramm mit der Systemvariablen $AA_S gelesen werden.

Die FS kann nach der Abwahl aus dem Teileprogramm mit M05, `SPOS`, `SPOSA` oder mit dem entsprechenden Nahtstellensignal vom PLC stillgesetzt werden.

COUPOFS mit Stop der Folgespindel

Das Ausschalten einer Synchronspindelkopplung wird mit Stop der Folgespindel erweitert:

- Schnellstmögliches Ausschalten einer Kopplung und Stop ohne Positionsangabe.

  Anschließend wird der Satzwechsel freigegeben.

  COUPOFS(FS, LS)

- Ausschalten der Kopplung mit Stop der Folgespindel auf der programmierten Position. Anschließend wird der Satzwechsel freigegeben.

Randbedingung:

COUPOFS(FS, LS) und COUPOFS(FS, LS, POS_{FS}) haben keine Bedeutung, wenn keine Kopplung aktiv war.

14.1.7 Synchronspindelkopplung durch PLC beeinflussen

Folgespindel vom PLC beeinflussen

Mit den kopplungsspezifischen axialen VDI-Nahtstellensignalen können vom PLC-Programm aus, Synchronisationsbewegungen für die Folgespindel beeinflusst werden. Dabei besteht die Möglichkeit eine durch Offset-Programmierung vorgegebene Synchronisationsbewegung für die Folgespindel vom PLC zu sperren bzw. zu unterdrücken oder nachzuholen.

Für die Leitspindel sind diese Signale wirkungslos. Es steht folgendes kopplungsspezifisches VDI-Signal (PLC → NCK) zur Verfügung:

NST **"Synchronisation sperren" (DB31, ... DBX31.5**

"Synchronisation sperren"

Die Synchronisationsbewegung für die Folgespindel wird mit den axialen Signal NST "Synchronisation sperren" (DB31, ... DBX31.5) unterdrückt.

Beim Einwechseln eines Satzes mit der Teileprogrammanweisung `COUPON` (FS, LS, Offset) in den Hauptlauf, wird für die Folgespindel folgendes Nahtstellensignal ausgewertet:

NST "Synchronisation sperren" (DB31, ... DBX31.5).

- Bei NST "Synchronisation sperren" (DB31, ... DBX31.5) = 0 wird der Positionsoffset wie bisher herausgefahren.
- Bei NST "Synchronisation sperren" (DB31, ... DBX31.5) = 1 wird nur der geschwindigkeitsstetige Synchronlauf hergestellt. Es erfolgt keine zusätzliche Bewegung der Folgespindel.

  Die Kopplung verhält sich dann analog zur Programmierung `COUPON(<FS>,<LS>)`.

Besonderheiten

Mit dem NST "Synchronisation sperren" (DB31, ... DBX31.5) können Offsetbewegungen der Folgespindel nicht beeinflusst werden, die folgendermaßen erzeugt wurden:

- SPOS, POS
- Synchronaktionen
- FC18 (bei 840D sl)
- JOG

Diese Funktionen werden mit den VDI-Signal NST "Vorschub halt/Spindel halt" (DB31, ... DBX4.3) beeinflusst.

Erreichen des Synchronlaufes

Immer bei Erreichen des Synchronlaufes werden unabhängig davon, ob die Synchronisation gesperrt wurde oder nicht, folgende beide VDI-Signale gesetzt:

NST "Synchronlauf grob" (DB31, ... DBX98.1) und

NST "Synchronlauf fein" (DB31, ... DBX98.0)

Der weitere Satzwechsel nach COUPON wird durch eine unterdrückte Synchronisation nicht verhindert.

Beispiel

Verhalten eines Satzwechsels nach `COUPON`

```
                                        ; NST "Synchronisation sperren"
                                        ; (DB31, ... DBX31.5) = 1 für S2 setzen
N51 SPOS=10 SPOS[2]=10                  ; Positionen entsprechen einem Offset
                                        von 0°
N52 COUPDEF(S2,S1,1,1,"FINE","DV")
N53 COUPON(S2,S1,77)                    ; Ist-Offset von 0 Grad bleibt bestehen
                                        ; keine Bewegung der Folgespindel,
```

```
                                                    ; VDI-Signale
                                                    ; NST "Synchronlauf grob"
                                                    ; (DB31, ... DBX98.1) und
                                                    ; NST "Synchronlauf fein"
                                                    ; (DB31, ... DBX98.0)
                                                    ; werden gesetzt u. der Satzwechsel
                                                    ; freigegeben.
N54 M0
N57 COUPOF(S2,S1)
N99 M30
```

Rücksetzen und Nachholen

Das Rücksetzen des NST "Synchronisation sperren" (DB31, ... DBX31.5) hat keine Auswirkungen auf den Folgespindeloffset. Wurde die Offsetbewegung der Folgespindel durch das VDI-Nahstellensignal unterdrückt, dann wird der Offset durch das Rücksetzen dieses Signals nicht automatisch herausgefahren.

Ein Nachholen der Synchronisation wird folgendermaßen erreicht:

- Durch Wiederholen der Teileprogrammanweisung `COUPON` (FS, LS, Offset) bei NST "Synchronisation sperren" (DB31, ... DBX31.5) = 0.

  `COUPON` (FS, LS, Offset) kann z. B. in einem ASUP geschrieben werden.
- Durch Setzen des NST "Neu Synchronisieren" (DB31, ... DBX31.4) = 1

Offset lesen

Mit folgenden Systemvariablen können drei verschiedene Positionsoffsetwerte der Folgespindel aus dem Teileprogramm und aus Synchronaktionen gelesen werden. Die Variable $P_COUP_OFFS[Sn] ist nur im Teileprogramm verfügbar.

Beschreibung	NCK-Variable
Programmierter Positionsoffset der Synchronspindel	$P_COUP_OFFS[Sn]
Positionsoffset der Synchronspindel sollwertseitig	$AA_COUP_OFFS[Sn]
Positionsoffset der Synchronspindel istwertseitig	$VA_COUP_OFFS[Sn]

"Vorschub halt/Spindel halt"

Mit Projektierung von Bit 4 im MD30455 MISC_FUNCTION_MASK wird das Verhalten des axialen NST **"Vorschub halt/Spindel halt" (DB31, ... DBX4.3)** für die Folgespindel festgelegt.

Bit 4 = 0 Kompatibilitätsmethode:

Die Wegnahme der Vorschubfreigabe für die Folgespindel bremst den Kopplungsverband ab.

Bit 4 = 1:

Die Vorschubfreigabe bezieht sich nur auf den Interpolationsanteil (SPOS,...) und hat keinen Einfluss auf die Kopplung.

Hinweis

Weitere Projektierungsmöglichkeiten von Achsfunktionen über MD30455 $MA_MISC_FUNCTION_MASK:

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; Rundachsen (R2), Kapitel Programmierung von Rundachsen

14.1.8 Überwachungen des Synchronbetriebs

Synchronlauf fein/grob

Neben den herkömmlichen Spindelüberwachungen wird im Synchronbetrieb zusätzlich der Synchronlauf der FS zur LS überwacht.

Dabei wird über die NST "Synchronlauf fein" (DB31, ... DBX98.0) bzw. NST "Synchronlauf grob" (DB31, ... DBX98.1) an die PLC gemeldet, ob die aktuelle Istposition (AV, DV) bzw. Istgeschwindigkeit (VV) der Folgespindel innerhalb des vorgegebenen Toleranzfensters liegt.

Beim Einschalten der Kopplung werden die Signale "Synchronlauf grob" und "Synchronlauf fein" mit dem Erreichen des sollwertseitigen Synchronlaufes aktualisiert.

Die Größe der Toleranzfenster wird durch Maschinendaten der FS eingestellt. Das Erreichen des Synchronlaufs wird durch folgende Faktoren beeinflusst:

- AV, DV: Positionsabweichung zwischen FS und LS
- VV: Geschwindigkeitsunterschied zwischen FS und LS

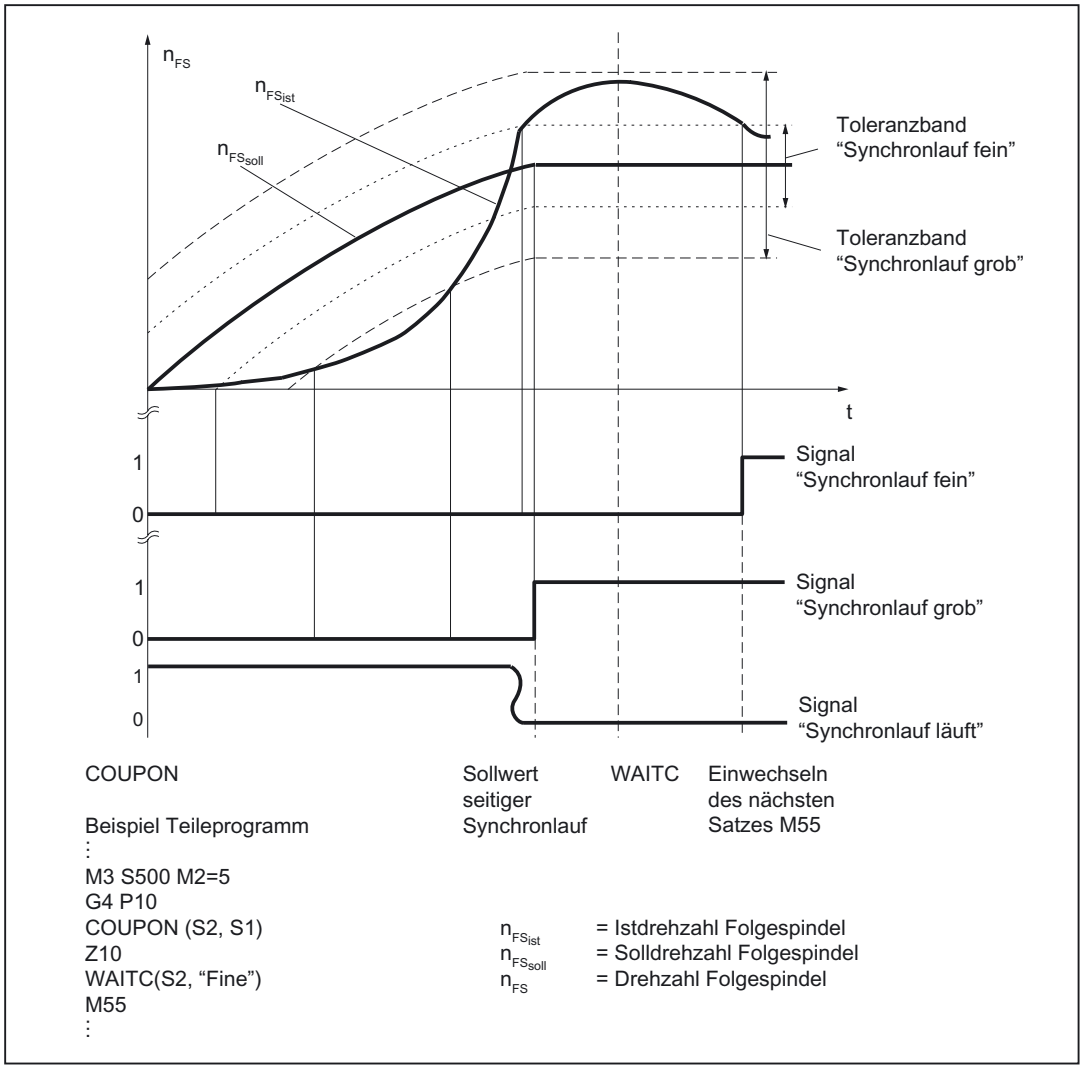


Bild 14-3 Synchronlaufüberwachung bei `COUPON` und Synchronlauftestmarke `WAITC` bei der Synchronisation auf eine drehende Leitspindel

Schwellwerte

Für die Folgespindel ist der jeweilige Positions- bzw. Geschwindigkeitstoleranzbereich der FS zur LS in Grad bzw. 1/min anzugeben.

- Schwellwert für "Synchronlauf grob"

  achsspez. MD37200: AV, DV: COUPLE_POS_TOL_COARSE

  MD37220: VV: COUPLE_VELO_TOL_COARSE

- Schwellwert für "Synchronlauf fein"

  achsspez. MD37210: AV,DV: COUPLE_POS_TOL_FINE

  MD37230: VV: COUPLE_VELO_TOL_FINE

Drehzahl-/Beschleunigungsgrenzen

Im Synchronbetrieb werden die Drehzahl- und Beschleunigungsgrenzwerte der Leitspindel steuerungsintern in der Weise angepasst, dass die Folgespindel unter Berücksichtigung der aktuellen Getriebestufe und des wirksamen Übersetzungsverhältnisses den Bewegungen der Leitspindel folgen kann ohne dabei ihre Grenzwerte zu überschreiten.

Beispielsweise wird die LS automatisch abgebremst, damit die FS die maximale Geschwindigkeit nicht überschreiten muss, um die Synchronität aufrecht zu erhalten.

14.2 Programmierung von Synchronspindelkopplungen

Tabelle 14- 1 Übersicht

Programmierte Kopplung	Projektierte Kopplung(en)	Bemerkung
Definieren einer Kopplung: **COUPDEF(FS, ...)**	Ändern von projekt. Daten: **COUPDEF(FS, ...)**	Einstellen der Kopplungsparameter
Einschalten einer Kopplung: **COUPON(FS, LS, POS$_{FS}$)** Einschalten und Übernahme einer Bewegung zur Differenzdrehzahl der Kopplung: **COUPONC(FS, LS)**		Ein- und Ausschalten
Ausschalten einer Kopplung: **COUPOF(FS, LS, POS$_{FS}$, POS$_{LS}$)** mit Stop der Folgespindel: **COUPOFS(FS, LS, POS$_{FS}$)**		
Löschen der Kopplungsdaten: **COUPDEL(FS, LS)**	Reaktivierung der projektierten Daten: **COUPRES(FS, LS)**	Aufräumen, Wiederherstellen

Verkürzte Angabe ohne Leitspindel

Ohne Angabe einer Leitachse sind folgende Sprachanweisungen zugelassen:

`COUPOF(FS)`, `COUPOFS(FS)`, `COUPDEL(FS)`, `COUPRES(FS)`.

Hinweis

Bei jeder `COUPDEF`-, `COUPON`- und `COUPONC`-Anweisung müssen FS und LS programmiert werden, damit keine Alarmmeldungen abgesetzt werden.

Literatur:

/PGA/ Programmierhandbuch Arbeitsvorbereitung, "Synchronspindel"

14.2.1 Vorbereitende Programmieranweisungen

Programmierbare Kopplungen

Die Anzahl der Kopplungen kann abhängig von den verfügbaren Achsen beliebig oft programmiert werden. Sie ergibt sich aus Anzahl der Achsen/Spindeln minus eine für den Master. Außerdem ist wie bisher eine Kopplung über Maschinendaten projektierbar.

Fest projektierte Kopplung

Für eine fest projektierte Synchronspindelkopplung können die Kopplungseigenschaften (sofern dafür kein Schreibschutz besteht) und die Übersetzungsverhältnisse durch das NC-Teileprogramm geändert werden. Die Maschinenachsen für FS und LS können nicht geändert werden.

Neue Kopplungen definieren

Mit der Sprachanweisung "COUPDEF" können Synchronspindelkopplungen neu angelegt (anwenderdefiniert) und für bestehende Kopplungen die Parameter geändert werden.

Bei vollständiger Angabe der Kopplungsparameter gilt:

COUPDEF (FS, LS, $Ü_{Zähler}$, $Ü_{Nenner}$, Satzwechselverhalten, Kopplungsart)

Mit FS und LS wird die Synchronspindelkopplung eindeutig bestimmt.

Die anderen Kopplungsparameter müssen nur dann programmiert werden, wenn sie geändert werden sollen. Für die nicht angegebenen Parameter bleibt der letztgültige Zustand erhalten.

Nachfolgend werden die einzelnen Kopplungsparameter erläutert:

- **FS, LS**: Spindelbezeichner für Folgespindel und Leitspindel

  z. B.: S1, SPI(1), S2, SPI(2)

  Die gültige Spindelnummer muss in dem achsspezifischen MD35000 $MA_SPIND_ASSIGN_TO_MACHAX einer Maschinenachse zugeordnet sein.

- **$Ü_{Zähler}$, $Ü_{Nenner}$**: Übersetzungsparameter für Zähler und Nenner

  Das Übersetzungsverhältnis wird mit Wertangaben für Zähler und Nenner vorgegeben, siehe Kapitel "Synchronbetrieb".

  Der Zähler muss mindestens programmiert werden. Falls kein Nenner angegeben ist, so wird dieser immer mit "1,0" angenommen.

- **Satzwechselverhalten**

  Damit kann bei Anwahl des Synchronbetriebes ausgewählt werden, wann der Satzwechsel erfolgen soll:

NOC: Satzwechsel wird sofort freigegeben

FINE: Satzwechsel bei "Synchronlauf fein"

COARSE: Satzwechsel bei "Synchronlauf grob"

IPOSTOP: Satzwechsel bei IPOSTOP (d. h. nach sollwertseitigem Synchronlauf)

Das Satzwechselverhalten wird als Zeichenkette (d. h. mit Anführungszeichen) vorgegeben.

Für die Angabe des Satzwechselverhaltens ist das Schreiben der fettgedruckten Buchstaben ausreichend. Die restlichen Buchstaben können zur besseren Lesbarkeit des Teileprogramms angegeben werden, sie sind jedoch nicht signifikant.

Wird das Satzwechselverhalten nicht angegeben, so bleibt das zu diesem Zeitpunkt wirksame Verhalten erhalten.

Mit den programmierbaren Synchronlauftestmarken **WAITC** wird das Einwechseln neuer Sätze bis zum Erreichen des angegebenen Synchronlaufs verzögert.

- **Kopplungsart**

DV (Desired Values): Sollwertkopplung zwischen FS und LS

AV (Actual Values): Istwertkopplung zwischen FS und LS

VV (Velocity Values): Geschwindigkeitskopplung zwischen FS und LS

Wird keine Kopplungsart angegeben, so bleibt die zu diesem Zeitpunkt wirksame erhalten.

Hinweis

Die Kopplungsart darf nur bei ausgeschaltetem Synchronbetrieb verändert werden!

Beispiele

COUPDEF (SPI(2), SPI(1), 1.0, 1.0, "FINE", "DV")

COUPDEF (S2, S1, 1.0, 4.0)

COUPDEF (S2, SPI(1), 1.0)

Defaulteinstellungen

Für die anwenderdefinierten Kopplungen gelten folgende Defaulteinstellungen:

- $Ü_{Zähler}$ =1,0
- $Ü_{Nenner}$ =1,0
- Satzwechselverhalten = **IPOSTOP** (Satzwechsel wird mit sollwertseitigem Synchronlauf freigegeben)
- Kopplungsart = **DV** (Sollwertkopplung)

Kopplungen löschen

Mit der Sprachanweisung "COUPDEL" werden anwenderdefinierte Kopplungen gelöscht.

COUPDEL (FS, LS)

Hinweis

COUPDEL wirkt auf eine aktive Kopplung, schaltet diese aus und löscht damit die Kopplungsdaten. Der Alarm 16797 ist damit bedeutungslos.

Die Folgespindel übernimmt die letzte Drehzahl. Dies entspricht dem Verhalten von COUPOF(FS, LS).

Ursprüngliche Kopplungsparameter aktivieren

Mit der Sprachanweisung "COUPRES" können die projektierten Kopplungsparameter wieder aktiviert werden.

COUPRES (FS, LS)

Dabei gehen die mit COUPDEF geänderten Parameter (einschließlich dem Übersetzungsverhältnis) verloren.

Mit "COUPRES" werden die in den Maschinen- und Settingdaten hinterlegten Parameter aktiviert (projektierte Kopplung) und die Defaulteinstellungen aktiviert (anwenderdefinierte Kopplung).

Programmierbarer Satzwechsel

Mit der Sprachanweisung "WAITC" kann im NC-Programm eine Stelle gekennzeichnet werden, an der ein Warten auf Synchronlaufbedingungen für die angegebene FS erfolgt und das Einwechseln neuer Sätze bis zum Erreichen des angegebenen Synchronlaufs verzögert (siehe Bild).

WAITC (FS)

Vorteil: Die Zeitspanne vom Einschalten der Synchronkopplung bis zum Erreichen des Synchronlaufs kann technologisch sinnvoll genutzt werden.

Hinweis

WAITC kann prinzipiell immer geschrieben werden. Ist eine angegebene Spindel nicht als FS aktiv bleibt die Anweisung für diese Spindel wirkungslos.

Wenn keine Synchronlaufbedingung angegeben ist, dann wird immer auf die jeweilige Kopplung programmierte/projektierte Synchronlaufbedingung, mindestens jedoch auf sollwertseitigen Synchronlauf, getestet.

Beispiele:

```
WAITC(S2),
:
WAITC(S2, "Fine"),
:
WAITC(S2, ,S4, "Fine")
```

Halten und Satzwechsel

Wurde für den Zeitraum der Wegnahme der Achsfreigaben für die Leit- bzw. Folgespindel "Halten" aktiviert, dann werden die **letzten** Soll-Positionen mit dem Geben der Achsfreigaben vom Servoantrieb wieder angefahren.

Mit den Programmanweisungen COUPON und WAITC kann der Satzwechsel beeinflusst werden. Das Satzwechselkriterium wird dabei mit COUPDEF oder über das MD21320 $MC_COUPLE_BLOCK_CHANGE_CTRL_1 festgelegt.

14.2.2 Programmieranweisungen für Ein- und Ausschalten der Kopplung

Synchronbetrieb einschalten

Mit der Sprachanweisung COUPON wird die Kopplung eingeschaltet und der Synchronbetrieb aktiviert.

Für das Einschalten des Synchronbetriebs sind zwei Varianten wählbar:

1. **COUPON(FS, LS)**

   Schnellstmögliches Einschalten des Synchronbetriebs mit beliebigem Winkelbezug zwischen Leit- und Folgespindel.

2. **COUPON(FS, LS, POS_{FS})**

   Einschalten des Synchronbetriebs mit einem definierten Winkelversatz POS_{FS} zwischen FS und LS. Dieser bezieht sich auf die Null-Grad-Position der Leitspindel in positiver Drehrichtung. Der Satzwechsel wird entsprechend der festgelegten Einstellung freigegeben. Bereich von POS_{FS}: 0 ... 359,999 Grad.

3. **COUPONC(FS, LS)**

   Beim Einschalten mit `COUPONC` wird die vorhergehende Programmierung von M3 S... oder M4 S... übernommen. Eine Differenzdrehzahl wird sofort übernommen. Die Programmierung einer Offsetposition ist nicht möglich.

Mit Programmierung von `COUPON`(FS, LS, POS_{FS}) oder SPOS bei bereits aktivem Synchronbetrieb kann der Winkelversatz zwischen FS und LS verändert werden.

Synchronbetrieb ausschalten

Für das Ausschalten des Synchronbetriebs sind drei Varianten wählbar:

1. **COUPOF(FS, LS)**

   Schnellstmögliches Ausschalten des Synchronbetriebs. Der Satzwechsel wird sofort freigegeben.

2. **COUPOF(FS, LS, POS_{FS})**

   Abwahl des Synchronbetriebes nach Überfahren der Ausschaltposition POS_{FS} . Der Satzwechsel wird erst nach Überfahren der Ausschaltposition freigegeben.

3. **COUPOF(FS, LS, POS_{FS} , POS_{LS})**

   Abwahl des Synchronbetriebes nach Überfahren der beiden Ausschaltposition POS_{FS} und POS_{LS} . Der Satzwechsel wird erst nach Überfahren der **beiden** programmierten Positionen freigegeben.

   Bereich von POS_{FS}, POS_{LS}: 0 ... 359,999°.

Falls Bahnsteuerbetrieb (G64) programmiert ist, wird steuerungsintern ein satzweiser Stop erzeugt.

Beispiele:

```
COUPDEF (S2, S1, 1.0, 1.0, "FINE, "DV")
:
COUPON (S2, S1, 150)
:
COUPOF (S2, S1, 0)
:
COUPDEL (S2, S1)
```

1. **COUPOFS(FS, LS)**

   Ausschalten einer Kopplung mit Stopp der Folgespindel. Der Satzwechsel erfolgt schnellstmöglich mit sofortigem Satzwechsel)

2. **COUPOFS(FS, LS, POS$_{FS}$)**

   Nach Überfahren der programmierten Folgeachs-Ausschaltposition, die sich auf das Maschinenkoordinatensystem bezieht, wird der Satzwechsel erst nach Überfahren der Ausschaltpositionen POSFS freigegeben.

   Wertebereich 0 ... 359,999°.

14.2.3 Axiale Systemvariablen für Synchronspindel

Aktuellen Kopplungszustand feststellen

Für die Folgespindel kann der aktuelle Kopplungszustand im NC-Teileprogramm mit folgender axialen Systemvariable gelesen werden:

$AA_COUP_ACT [<axialer Ausdruck>]

Erläuterung zu <axialer Ausdruck> siehe " Synchronbetrieb (Seite 875) ".

Beispiel:

$AA_COUP_ACT[S2]

Der gelesene Wert hat für die Folgespindel folgende Bedeutung:

Byte = 0:	keine Kopplung aktiv
Bit 2 = 1:	Synchronspindelkopplung aktiv
Bit 2 = 0:	Synchronspindelkopplung ist nicht aktiv

Aktuellen Winkelversatz lesen

Der aktuelle Positionsoffset der FS zur LS kann im NC-Teileprogramm mit folgenden axialen Systemvariablen gelesen werden:

- sollwertseitiger Positionsoffset der FS zur LS:

  $AA_COUP_OFFS[<axialer Ausdruck>]

- istwertseitiger Positionsoffset der FS zur LS:

  $VA_COUP_OFFS[<axialer Ausdruck>]

Beispiel:

$AA_COUP_OFFS[S2]

Wird mit `COUPON` ein Winkelversatz programmiert, dann stimmt dieser nach Erreichen des sollwertseitigen Synchronlaufs mit dem gelesenen Wert überein.

Programmierten Winkelversatz lesen

Der zuletzt programmierte Positionsoffset der FS zur LS im NC-Teileprogramm kann mit der folgenden axialen Systemvariablen gelesen werden:

$P_COUP_OFFS[<axialer Ausdruck>]

Hinweis

Nach Wegnahme der Reglerfreigabe und Nachführbetrieb bei aktivem Synchronbetrieb stellt sich nach erneuter Erteilung der Reglerfreigabe ein anderer Positionsoffset ein als der ursprünglich programmierte Wert.

$P_COUP_OFFS gibt nur den ursprünglich programmierten Wert zurück. Den aktuellen Wert geben $AA_COUP_OFFS und auch $VA_COUP_OFFS zurück. Mit NST DB31, ... DBX31.4 (Neu synchronisieren) kann der programmierte Offset wieder hergestellt werden.

14.2.4 Automatische An- und Abwahl der Lageregelung

Verhalten im Drehzahlsteuerbetrieb

In der Kopplungsart DV wird mit den Programmanweisungen `COUPON`, `COUPONC` und `COUPOF`, `COUPOFS` die Lageregelung für die Leitspindel im Bedarfsfall ein- bzw. ausgeschaltet. Sind dabei mehrere Folgespindeln an der Leitspindel, dann schaltet im Drehzahlsteuerbetrieb die **erste** DV-Kopplung die Lageregelung für die Leitspindel **ein** und die **letzte** DV-Kopplung die Lageregelung für die Leitspindel **aus**, wenn nicht SPCON programmiert ist.

Die Leitspindel muss sich nicht im gleichen Kanal wie die Folgespindel befinden.

Automatische Anwahl bei COUPON und COUPONC

Abhängig von der Kopplungsart wirkt sich `COUPON` und `COUPONC` auf die Lageregelung für den Synchronbetrieb wie folgt aus:

Kopplungsart	DV	AV	VV
Folgespindel FS	Lageregelung Ein	Lageregelung Ein	keine Aktion
Leitspindel LS	Lageregelung Ein [1]	keine Aktion	keine Aktion

[1] Die Lageregelung wird bei `COUPON` und `COUPONC` eingeschaltet, wenn **mindestens eine** Folgespindel mit der Kopplungsart DV an die Leitspindel angekoppelt wurde.

Automatische Abwahl bei COUPOF und COUPOFS

Abhängig von der Kopplungsart wirkt sich COUPOF und COUPOFS auf die Lageregelung wie folgt aus:

Kopplungsart	DV	AV	VV
Folgespindel FS	Lageregelung Aus [2]	Lageregelung Aus [2]	keine Aktion [2]
Leitspindel LS	Lageregelung Aus [3]	keine Aktion	keine Aktion

[2] COUPOF und COUPOFS ohne Positionsangabe

Für die Folgespindel wird Drehzahlsteuerbetrieb aktiviert. Bei COUPFS mit Stopp-Position wird Positionierbetrieb aktiviert. Die Lageregelung wird **nicht ausgeschaltet**, wenn sich die Folgespindel im lagegeregelten Spindelbetrieb mit SPCON befand oder COUPFS mit Position programmiert wurde.

[3] Die Lageregelung wird bei COUPOF **ausgeschaltet**, wennkeine weitere Kopplung mit der Kopplungsart DV zu dieser Leitspindel besteht. Die Lageregelung **bleibt weiterhin bestehen**, falls sich die Leitspindel im Positionierbetrieb oder Achsbetrieb befindet **oder** im lagegegerelten Spindelbetrieb mit SPCON befand.

14.3 Projektierung eines Synchronspindelpaares über Maschinendaten

Kopplungsparameter

Je NC-Kanal kann **eine** Synchronspindelkopplung fest über kanalspezifische Maschinendaten projektiert werden.

Dabei sind die Maschinenachsen (Spindeln) festzulegen, zwischen welchen die Kopplung bestehen soll und welche Eigenschaften die Kopplung besitzt.

Folgende Parameter sind für die Synchronspindelkopplung fest projektierbar:

- **Synchronspindelpaar** (kanalspez. MD21300 $MC_COUPLE_AXIS_1[n])

  Mit diesen Maschinendaten werden die beiden Maschinenachsen festgelegt, die das Synchronspindelpaar bilden (Folgespindel(n=0), Leitspindel (n=1)).

  Der Eintrag der Achsnummern 0 hat zur Folge, dass keine Kopplung über Maschinendaten fest projektiert ist. Die Maschinendaten für die Kopplungseigenschaften sind dann irrelevant.

  Für die projektierte Kopplung sind die Maschinenachsnummern für FS und LS vom NC-Teileprogramm nicht veränderbar.

- **Übersetzungsverhältnis**

  Dies wird über Settingdaten mit Hilfe von zwei Übersetzungsparametern (kanalspez. SD42300 $SC_COUPLE_RATIO_1[n]) als Zähler und Nenner vorgegeben.

  $k_Ü$ = Übersetzungsparameter Zähler : Übersetzungsparameter Nenner = $SC_COUPLE_RATIO[0] : $SC_COUPLE_RATIO[1]

  Mit der Sprachanweisung `COUPDEF` kann das Übersetzungsverhältnis vom NC-Teileprogramm verändert werden, sofern kein Änderungsschutz besteht.

- **Satzwechselverhalten**

  (kanalspezifisches MD21320 $MC_COUPLE_BLOCK_CHANGE_CTRL_1)

  Damit kann zwischen folgenden Möglichkeiten festgelegt werden, wann der Satzwechsel erfolgen soll:

  0: Satzwechsel erfolgt sofort

  1: Satzwechsel bei "Synchronlauf fein"

  2: Satzwechsel bei "Synchronlauf grob"

  3: Satzwechsel bei `IPOSTOP` (d. h. nach sollwertseitigem Synchronlauf)

- **Kopplungsart** zwischen FS und LS:

  (kanalspezifisches MD21310 $MC_COUPLING_MODE_1)

  0: Istwertkopplung (AV)

  1: Sollwertkopplung (DV)

  2: Geschwindigkeitskopplung (VV)

- **Abbruch der Kopplung mit NC-Start**:

  kanalspezifisches MD21330 $MC_COUPLE_RESET_MODE_1

- **Änderungsschutz der Kopplungsparameter:**

  (kanalspezifisches MD21340 $MC_COUPLE_IS_WRITE_PROT_1)

  Mit diesem Maschinendatum kann festgelegt werden, ob die projektierten Kopplungsparameter Übersetzungsverhältnis, Kopplungsart und Satzwechselverhalten vom NC-Teileprogramm beeinflusst werden dürfen.

  0: Kopplungsparameter sind vom NC-Teileprogramm veränderbar

  1: Kopplungsparameter sind vom NC-Teileprogramm nicht veränderbar
  Änderungsversuche werden mit Alarmmeldung abgewiesen.

14.3.1 Projektierung des Verhaltens bei NC-Start

Das Verhalten bei NC-Bearbeitungsprogramm-Start wird durch kanalspezifische Maschinendaten festgelegt.

Tabelle 14- 2 Synchronkopplungsverhalten bei NC-Start

	Projektierte Kopplung	Programmierte Kopplung *
	MD COUPLE_RESET_MODE	MD START_MODE_MASK
Kopplung bleibt bestehen	Bit 0 = 0	Bit 10 = 0
Kopplung abwählen	Bit 0 = 1	Bit 10 = 1
Projektierte Daten aktivieren	Bit 5 = 1	-
Kopplung einschalten	Bit 9 = 1	-

* siehe Kapitel "Projektierung eines Synchronspindelpaares über Maschinendaten"

14.3.2 Projektierung des Verhaltens bei Reset

Bei Reset und NC-Bearbeitungsprogramm-Ende kann mit kanalspezifischen Maschinendaten folgendes Verhalten eingestellt werden:

Tabelle 14- 3 Synchronkopplungsverhalten bei NC-Bearbeitungsprogramm-Ende und nach Reset

	Projektierte Kopplung	Programmierte Kopplung *
Kopplung bleibt bestehen	MD COUPLE_RESET_MODE Bit 1 = 0	MD RESET_MODE_MASK Bit 10 = 1
Kopplung abwählen	MD COUPLE_RESET_MODE Bit 1 = 1 MD RESET_MODE_MASK Bit 0 = 1 (Erzeugen eines Satzes bei RESET)	MD RESET_MODE_MASK Bit 10 = 0 Bit 0 = 1
Projektierte Daten aktivieren	MD COUPLE_RESET_MODE Bit 6 = 1 MD RESET_MODE_MASK Bit 0 = 1	-

* siehe Kapitel "Projektierung eines Synchronspindelpaares über Maschinendaten"

14.4 Besonderheiten des Synchronbetriebs

14.4.1 Allgemeine Besonderheiten des Synchronbetriebs

Regeldynamik

Bei Verwendung der Sollwertkopplung sind die Lageregler-Parameter von FS und LS (z. B. K_V-Faktor) aufeinander abzugleichen. Gegebenenfalls sind unterschiedliche Parametersätze für den Drehzahlregel- und Synchronbetrieb zu aktivieren. Die Regel-Parameter der Folgespindel können abweichend von Lageregelung, Vorsteuerung und Parametersatz auch wie im ungekoppelten Fall über dem MD30455 $MA_MISC_FUNCTION_MASK eingestellt werden, siehe Kapitel "Besonderheiten bei der Inbetriebnahme der Synchronspindelkopplung".

Vorsteuerung

Die Vorsteuerung ist aufgrund der besseren Regeldynamik für die Leit- und Folgespindel im Spindelbetrieb **immer aktiv.**

Eine Abwahl der Vorsteuerung für FS und LS ist jedoch mit dem achsspezifischen MD32620 $MA_FFW_MODE möglich. Wenn MD32620 $MA_FFW_MODE auf Null gesetzt wird, gibt es Funktionseinschränkungen. Die Lageregelung kann nicht mehr in Bewegung mit SPCON eingeschaltet werden. SPOS, M19 oder SPOSA sind damit nicht möglich. Für FS und LS kann die Vorsteuerung vom NC-Teileprogramm nicht mit FFWOF ausgeschaltet werden.

Die Vorsteuerungsart (Drehzahl- oder Momenten-Vorsteuerung) wird mit dem achsspezifischen. MD32620 $MA_FFW_MODE festgelegt.

Literatur:

/FB2/ Funktionsbeschreibung Erweiterungsfunktionen, Kompensationen (K3)

Drehzahl und Beschleunigungsgrenzen

Die Drehzahl- und Beschleunigungsgrenzen der zum Synchronbetrieb gehörenden Spindeln werden von der "schwächsten Spindel" des Synchronspindelpaares bestimmt. Dabei werden die aktuellen Getriebestufen, die programmierte Beschleunigung und der wirksame Lageregelzustand (Ein/Aus) berücksichtigt.

Steuerungsintern wird die maximale Spindeldrehzahl und Beschleunigung der LS unter Berücksichtigung des Übersetzungsverhältnisses und den Spindelbegrenzungen der Folgespindel ermittelt.

Mehrfach-Kopplungen

Wird beim Einschalten des Synchronbetriebs festgestellt, dass bereits für FS oder LS eine Kopplungsbeziehung aktiv ist, so wird der Einschaltvorgang ignoriert und eine Alarmmeldung generiert.

Beispiel für Mehrfach-Kopplungen:

- eine Spindel wird als FS für mehrere LS verwendet
- Kopplungskaskade

Anzahl der projektierbaren Spindeln pro Kanal:

- Jede im Kanal vorhandene Achse kann als Spindel projektiert werden. Die Anzahl der Achsen pro Kanal ist abhängig von der Steuerungsausprägung.

Kanalübergreifende Sollwertkopplung und beliebige Anzahl der Folgespindeln in beliebigen Kanälen einer NCU:

- Es ist möglich, kanalübergreifende Synchronspindelkopplungen ohne zusätzliche Einschränkungen für DV, AV und VV vorzunehmen.
- Es können beliebig viele Folgespindeln, entsprechend Anzahl aller Spindeln minus eine Spindel für den Master, in beliebigen Kanälen einer NCU an eine Leitspindel angekoppelt werden. Der Realtime-Rechenzeitverbrauch kann sich gegebenenfalls auch auf die Anzahl der Folgespindeln auswirken.

Synchronbetrieb von ASUP starten

Mit Hilfe von ASUP (Start von asynchronen Unterprogrammen) durch die PLC kann in der Betriebsart AUTOMATIK bzw. MDA der Synchronbetrieb zu einem beliebigen Zeitpunkt eingeschaltet oder beendet werden!

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen, BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)

Verhalten bei Alarmen

Bei auftretenden Alarmen (z. B. Servoalarme) **während des Synchronbetriebs**, bei denen steuerungsintern die Reglerfreigabe weggenommen wird und Nachführen aktiv ist, erfolgt das anschließende Verhalten entsprechend, als ob das NST "Reglerfreigabe" (DB31, ... DBX2.1) von der PLC weggenommen wurde (und NST "Nachführbetrieb" (DB31, ... DBX1.4) ist gesetzt), siehe Kapitel "Einfluss über PLC-Nahtstelle auf den Synchronbetrieb".

Mit NST "Neu synchronisieren" (DB31, ... DBX31.4) wird der programmierte Offset wieder hergestellt, siehe Kapitel "Synchronität der Folgespindel wieder herstellen".

14.4.2 Synchronität der Folgespindel wieder herstellen

Ursachen für einen Positionsversatz

Beim Wiederaufsetzen der Kopplung nach der Wegnahme von Antriebsfreigaben kann bei aktiviertem Nachführbetrieb ein Positionsversatz zwischen Leit- und Folgespindel auftreten. Ursachen für einen Positionsversatz können sein:

- Ein Teil wurde eingespannt oder beide Spindeln wurden manuell verdreht (Arbeitsraum ist geöffnet, Antriebe sind stromlos).
- Nach der Wegnahme der Spindelfreigaben trudeln die beiden Spindeln unterschiedlich aus, falls keine mechanische Verbindung besteht.
- Ein Antriebsalarm ist aufgetreten (interner Nachführbetrieb):

  DB31, ... DBX61.3 (Nachführbetrieb aktiv) = 1

  Beim Löschen des Alarms darf von der NC keine Synchronisationsbewegung ausgelöst werden.
- Eine Synchronisation wurde nicht ausgeführt wegen einer Synchronisationssperre der Folgespindel:

  DB31, ... DBX29.5 (Synchronisation sperren)

Prinzipieller Ablauf

Eine verloren gegangene oder eine nicht ausgeführte Synchronität zwischen Folge- und Leitspindel kann nach folgendem Ablauf wieder hergestellt werden:

1. Die Achsfreigaben setzen und Synchronisationssperre aufheben, falls diese gesetzt wurde.
2. Folgespindel-Neusynchronisation starten mit dem NC/PLC-Nahtstellensignal:

   DB31, ... DBX31.4 (Neu Synchronisieren)

   Erst nachdem die Neusynchronisation ausgeführt worden ist, kann der sollwertseitige Synchronlauf wieder vollständig hergestellt werden.
3. Auf Synchronlauf der gekoppelten Spindel warten.

Neusynchronisation freigeben

Mit dem Setzen der Freigaben wird die Kopplung auf den aktuellen Istpositionen geschlossen. Die beiden folgenden NC/PLC-Nahtstellensignale werden gesetzt:

DB31, ... DBX98.1 (Synchronlauf grob)

DB31, ... DBX98.0 (Synchronlauf fein)

Folgende **Voraussetzungen** müssen für eine Neusynchronisation erfüllt sein:

- Es ist eine gesetzte Achsfreigabe für die Folgespindel erforderlich.
- Für die Folgespindel darf von der PLC keine Synchronisationssperre gesetzt sein:

  DB31, ... DBX31.5 (Synchronisation sperren)

Folgespindel neu synchronisieren

Eine Neusynchronisation wird für die betreffende Folgespindel gestartet und mit dem Erkennen der Low-High-Flanke des folgenden NC/PLC-Nahtstellensignals vorgenommen:

DB31, ... DBX31.4 (Neu synchronisieren)

Die NC quittiert das Erkennen der Flanke durch Ausgabe des NC/PLC-Nahtstellensignals:

DB31, ... DBX99.4 (Synchronisation läuft)

Das Nahtstellensignal "Synchronisation läuft" wird zurückgesetzt, wenn:

- die Synchronisation der Folgespindel bis zur Herstellung der sollwertseitigen Synchronität abgeschlossen ist.
- das NST DB31, ... DBX31.4 (Neu synchronisieren) zurückgesetzt wurde.

Verhalten der Synchronlaufsignale bei Zusatzbewegungen für die Folgespindel

Es wird der überlagerte Anteil zur Ermittlung der Synchronlaufsignale herausgerechnet.

Beispiel

```
Programmcode                                Kommentar
N51 SPOS=0 SPOS[2]=90
N52 OUPDEF(S2,S1,1,1,"FINE","DV")
N53 COUPON(S2,S1,77)
N54 M0                                      ; Offset=77°, Synchronlaufsignale "grob",
                                              "fein" stehen an.
N55 SPOS[2]=0 FA[S2]=3600                   ; Offsetänderung, Synchronlaufsignale
                                              "grob", "fein" werden gemeldet
N56 M0                                      ; (Toleranzen beachten, s. o.)
                                            ; Offset=0°, Synchronlaufsignale "grob"
                                              ,"fein" stehen an.
N60 M2=3 S2=500                             ; Differenzdrehzahl, Synchronlaufsignale
                                              "grob", "fein" werden gemeldet.
                                            ; Offset unbestimmt Synchronlaufsignale
                                              "grob" "fein" werden gemeldet.
N65 M0                                      ; (Toleranzen beachten, s. o.)
```

Hinweis

Mit der Wegnahme der Achsfreigaben kann eine der Folgespindel überlagerte Bewegung (z. B. SPOS) unterbrochen werden. Dieser Bewegungsanteil wird nicht durch das NC/PLC-Nahtstellensignal DB31, ... DBX31.4 (Neu Synchronisieren) beeinflusst, sondern wird über den REPOS-Vorgang restauriert.

Randbedingung

NST DB31, ... DBX31.4 (Neu synchronisieren) wirkt nur, wenn es eine **definierte Offsetposition** der Folgespindel zur Leitspindel gibt.

Dies ist der Fall nach `COUPON` mit Offsetposition, wie z. B. `COUPON(...,77)` oder `SPOS`, `SPOSA`, `M19` für die Folgespindel bei geschlossener Kopplung.

14.4.3 Einfluss über PLC-Nahtstelle auf den Synchronbetrieb

PLC-Nahtstellensignale

Bei Synchronbetrieb muss der von der PLC durch Setzen von Nahtstellensignalen von LS bzw. FS resultierende Einfluss auf die Synchronspindelkopplung beachtet werden.

Die Wirkung der wesentlichen PLC-Nahtstellensignale auf die Synchronspindelkopplung wird nachfolgend beschrieben.

Spindelkorrektur (DB31, ... DBB19)

Im Synchronbetrieb ist nur für die Leitspindel der von der PLC vorgegebene Spindelkorrekturwert wirksam.

Achsen-/Spindelsperre (DB31, ... DBX1.3)

Das Verhalten der beteiligten Achsen kann aus folgender Tabelle entnommen werden:

gesetzt: 1, nicht gesetzt: 0

Nr.	LS/LA	FS/FA	Kopplung	Verhalten
1	0	0	Aus	Sollwerte der Achsen werden ausgegeben
2	0	1	Aus	keine Sollwertausg. für FS/FA
3	1	0	Aus	keine Sollwertausg. für LS/LA
4	1	1	Aus	keine Sollwertausg. für LS/LA und FS/FA
5	0	0	Ein	Sollwerte der Achsen werden ausgegeben
6	0	1	Ein	Sperre wirkt nicht für FS/FA
7	1	0	Ein	Sperre wirkt auch für FS/FA keine Sollwertausgabe
8	1	1	Ein	keine Sollwertausg. für LS/LA und FS/FA

- Das Signal wirkt bei eingeschalteter Kopplung für FS/FA nicht mehr. →Nr. 6
- Wird das Signal für die LS/LA gesetzt, dann wirkt dies auch auf die FS/FA(en), →Nr. 7
- Ein eventuell zwischen zwei Spindeln eingespanntes Werkstück (Werkstückübergabe von Vorder- nach Rückseitenbearbeitung) kann nicht zerstört werden.

Reglerfreigabe (DB31, ... DBX2.1)

Wegnahme der "Reglerfreigabe" bei LS (entweder über PLC-Nahtstelle oder steuerungsintern bei Störungen):

Falls im Synchronbetrieb die Reglerfreigabe der LS auf "0" gesetzt wird und Sollwertkopplung aktiv ist, erfolgt steuerungsintern eine Umschaltung auf Istwertkopplung. Falls sich die LS dabei in Bewegung befindet, wird diese stillgesetzt und eine Alarmmeldung erzeugt. Der Synchronbetrieb bleibt erhalten.

Wegnahme der "Reglerfreigabe" bei FS im Synchronbetrieb (entweder über PLC-Nahtstelle oder steuerungsintern bei Störungen).

Falls vor Anwahl des Synchronbetriebes die "Reglerfreigabe" für beide Spindeln nicht ansteht wird beim Einschalten der Kopplung dennoch der Synchronbetrieb aktiviert. FS und LS bleiben jedoch solange im Stillstand, bis die Reglerfreigabe für beide erteilt wird.

Setzen der "Reglerfreigabe" bei FS und LS:

Mit der Signalflanke von NST "Reglerfreigabe" = 1 wird entsprechend dem NST "Nachführbetrieb" entweder die alte Position (Position bei Wegnahme der Reglerfreigabe) wieder eingenommen (Signalzustand = 0: Halten aktiv) oder es werden die aktuellen Positionen (Positionsoffset) verwendet (Signalzustand = 1: Nachführen aktiv).

Hinweis

Wird für die FS nach Spindel-Halt die "Reglerfreigabe" weggenommen, ohne vorher die Kopplung auszuschalten, so wird ein durch äußere Einwirkung hervorgerufener Synchronlauffehler (z. B. Verdrehen von Hand) bei Wiedereinschalten der "Reglerfreigabe" nicht kompensiert.

Dadurch kann für spezielle Anwendungen der definierte Winkelbezug zwischen FS und LS verloren gehen. Der programmierte Offset kann mit folgendem Nahtstellensignal wieder hergestellt werden:

NST "Neu synchronisieren" (DB31, ... DBX31.4).

Nachführbetrieb (DB31, ... DBX1.4)

Das NST "Nachführbetrieb" ist nur dann relevant, wenn die Reglerfreigabe des Antriebs weggenommen ist. Bei Setzen der Reglerfreigabe für die FS und LS wird abhängig von dem NST "Nachführbetrieb" entweder die Position bei Wegnahme der Reglerfreigabe wieder eingenommen (Signalzustand = 0: Halten aktiv) oder es werden die aktuellen Positionen verwendet (Signalzustand = 1: Nachführen aktiv).

Lagemesssystem 1/2 (DB31, ... DBX1.5 u. 1.6)

Während des Synchronbetriebes ist eine Umschaltung des Lagemesssystems für FS und LS nicht verriegelt. Die Kopplung wird aufrechterhalten. Es wird jedoch empfohlen, die Umschaltung nur bei abgewähltem Synchronbetrieb vorzunehmen.

Wird im Synchronbetrieb für die FS oder LS der Parkzustand angewählt, so ist das Verhalten entsprechend wie bei Wegnahme der "Reglerfreigabe".

Restweg löschen/Spindel-Reset (DB31, ... DBX2.2)

Mit Setzen von Spindel-Reset für die LS im Synchronbetrieb wird diese mit der eingestellten Beschleunigung auf Stillstand abgebremst. Der Synchronbetrieb zwischen FS und LS bleibt dabei erhalten. Die überlagerte Bewegung (außer bei COUP...) wird schnellstmöglich beendet.

Spindel-Halt (Vorschub-Halt) (DB31, ... DBX4.3)

Mit Aktivierung von "Spindel-Halt" für FS oder LS werden beide gekoppelten Spindeln über eine Rampe bis zum Stillstand abgebremst; der Synchronbetrieb bleibt weiterhin wirksam.

Sobald für keine der an der Kopplung beteiligten Spindeln das NST "Spindel-Halt" mehr ansteht, wird auf den vorherigen Drehzahlsollwert beschleunigt.

Anwendung

Mit "Spindel-Halt" kann das Synchronspindelpaar ohne Versatz stillgesetzt werden, da die Lageregelung erhalten bleibt.

Beispiel

Beim Öffnen der Schutztür während aktiver Synchronspindelkopplung müssen FS und LS unter Einhaltung der Kopplungsbeziehung stillgesetzt werden. Dabei ist mit NST "Spindel-Halt" FS und LS bis zum Stillstand abzubremsen (NST "Achse/Spindel steht" (DB31, ... DBX61.4) = 1). Anschließend kann für beide Spindeln die "Reglerfreigabe" weggenommen werden.

S-Wert löschen (DB31, ... DBX16.7)

Der programmierte S-Wert für die LS wird gelöscht und die LS wird über eine Rampe bis zum Stillstand abgebremst. Der Synchronbetrieb zwischen FS und LS bleibt dabei erhalten.

Dagegen ist für die FS im Synchronbetrieb das NST "S-Wert löschen" ohne Wirkung.

Spindel neu synchronisieren 1/2 (DB31, ... DBX16.4 u. 16.5)

Auch im Synchronbetrieb kann ein Synchronisieren der Spindel mit dem Lagemesssystem bei LS vorgenommen werden. Es wird jedoch empfohlen, die Neusynchronisation der Leitspindel nur bei abgewähltem Synchronbetrieb vorzunehmen.

Neu synchronisieren (DB31, ... DBX31.4)

Bei einen verloren gegangen oder bei einer nicht ausgeführten Synchronität zwischen Folge und Leitspindel kann der programmierte Offset mit folgendem Nahtstellensignal wieder hergestellt werden:

NST "Neu synchronisieren" (DB31, ... DBX31.4).

Mit dem Erkennen der Low-High-Flanke des VDI-Signals wird für die betreffende Folgespindel die Neusynchronisation gestartet und von der NC mit folgendem Nahtstellensignal für die Folgespindel quittiert:

NST "Synchronisation läuft" (DB31, ... DBX99.4) für die Folgespindel quittiert.

Verfahrtasten bei JOG (DB31, ... DBX4.6 u. 4.7)

Im Synchronbetrieb werden für die FS die "Verfahrtasten plus und minus" bei JOG steuerungsintern **nicht verriegelt**, so dass bei einer Betätigung eine überlagernde Verfahrbewegung der FS erzeugt wird.

Hinweis

Falls eine überlagernde Verfahrbewegung verhindert werden soll, muss dies vom PLC-Anwenderprogramm verriegelt werden.

NC-Stop Achsen plus Spindeln (DB21, ... DBX7.4)

Mit "NC-Stop Achsen plus Spindeln" im Synchronbetrieb werden die gekoppelten Spindeln unter Beachtung der eingestellten Dynamik abgebremst. Der Synchronbetrieb bleibt dabei erhalten.

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

siehe Kapitel "Projektierung des Verhaltens bei NC-Start".

Hinweis

NC-Start nach NC-Stop wählt den Synchronbetrieb nicht ab.

14.4.4 Differenzdrehzahl zwischen Leit- und Folgespindel

Wann entsteht eine Differenzdrehzahl?

Eine Differenzdrehzahl entsteht z. B. bei Drehmaschinenanwendungen, wenn sich zwei Spindeln gegenüber stehen. Durch vorzeichenbehaftete Addition zweier Drehzahlquellen wird dabei ein Drehzahlanteil über den Koppelfaktor von der Leitspindel abgeleitet. Additiv dazu kann für die Folgespindel Folgendes programmiert werden:

- Drehzahl mit S... und
- Drehrichtung mit M3, M4, oder M5

Um einen Gleichlauf zu erhalten, wird in der Regel ein Koppelfaktor mit den Wert '-1' erreicht. Durch diese Vorzeichenumkehr ergibt sich dann für die Folgespindel gegenüber einer zusätzlichen Drehzahlprogrammierung eine Drehzahldifferenz. Dieses typische Verhalten bezogen auf die NC wird im folgenden Bild verdeutlicht.

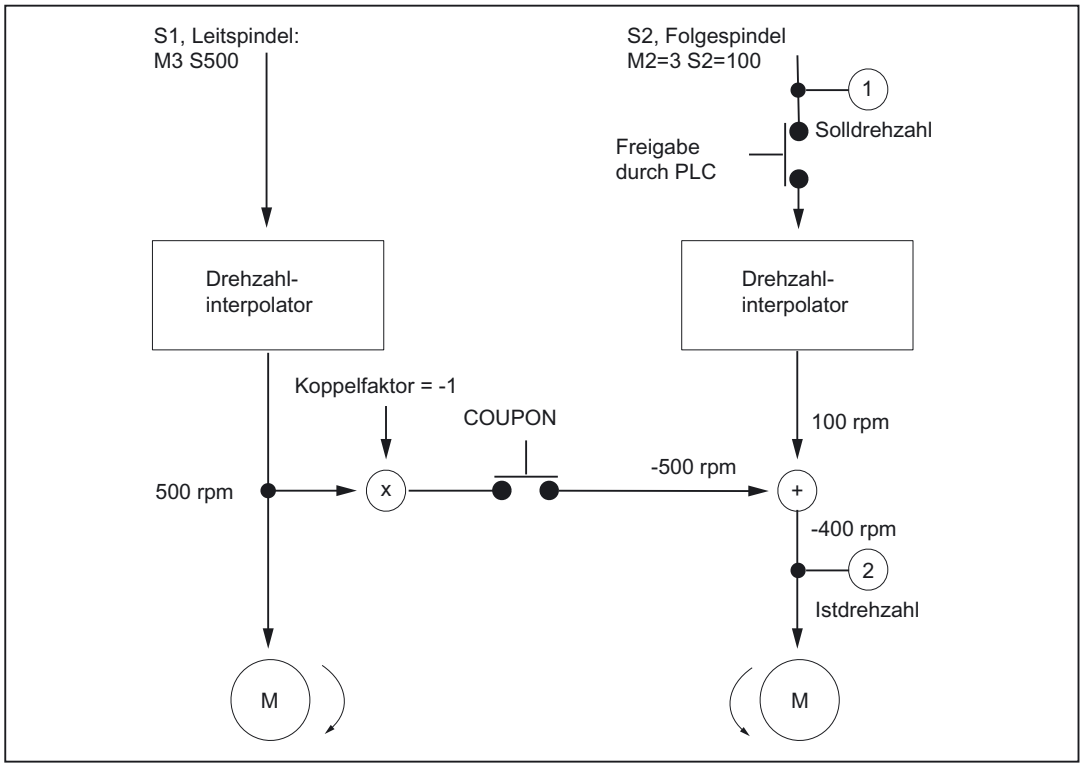


Bild 14-4 Schematische Darstellung zur Differenzdrehzahlerzeugung

Beispiel

```
N01 M3 S500                ; S1 dreht 500U/min positiv
                           ; Masterspindel ist die Spindel 1
N02 M2=3 S2=300            ; S2 dreht 300U/min positiv
N05 G4 F1;
N10 COUPDEF(S2,S1,-1)      ; Koppelfaktor -1:1
N11 COUPON(S2,S1)          ; Kopplung aktivieren, die Drehzahl der Folgespindel S2
                           ; ergibt sich aus der Drehzahl der Leitspindel S1 und
                           ; dem Koppelfaktor
N26 M2=3 S2=100            ; Programmierung der Differenzdrehzahl,
                           ; S2 ist Folgespindel
```

Anwendung

Fertigungsverfahren bei positionierter Leitspindel mit angetriebenen Werkzeugen erfordern einen exakten Synchronlauf der Gegenspindel, die dann wie eine Folgespindel wirkt. Ein sich um die Folgespindel drehender Revolver ermöglicht die Bearbeitung mit verschiedenen Werkzeugtypen. Folgendes Bild zeigt eine Anwendung, bei der das Werkzeug parallel zur Hauptspindel steht.

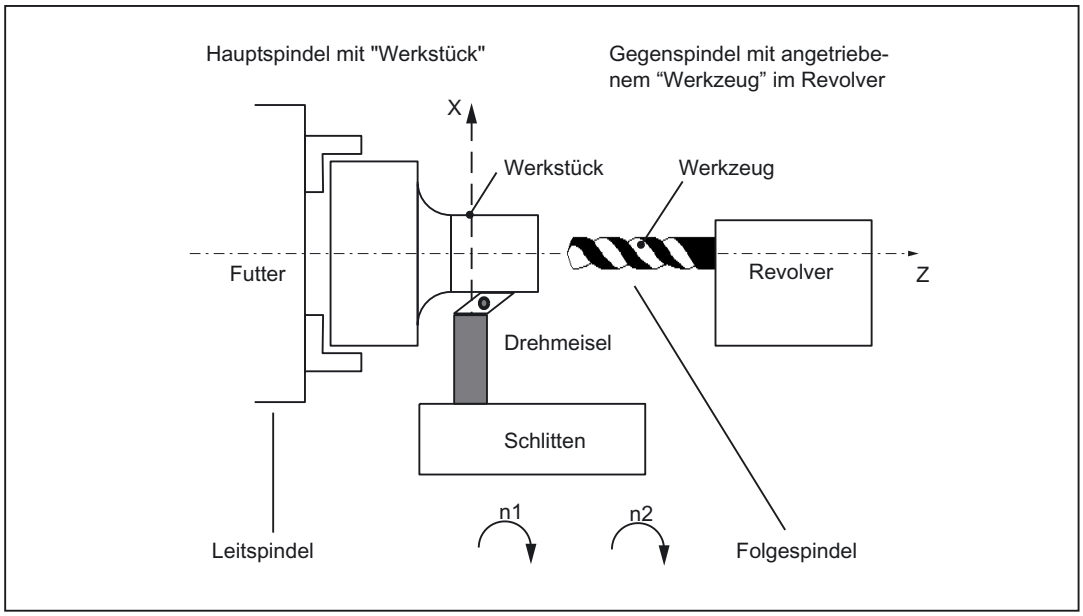


Bild 14-5 Applikation an einer Einschlittendrehmaschine mit Revolver um Z-Achse

Voraussetzungen

Grundsätzliche Voraussetzungen für eine Differenzdrehzahlprogrammierung:

- Es ist die Funktionalität der Synchronspindel erforderlich.
- Die Dynamik der Folgespindel sollte mindestens genauso groß sein wie die der Leitspindel. Anderenfalls kann es z. B. beim Gewindebohren ohne Ausgleichsfutter G331/G332 zu Qualitätseinbußen kommen.
- Die Differenzdrehzahl muss in dem Kanal programmiert werden, in dem sich die Folgespindel befindet. Die Leitspindel kann in einem anderen Kanal programmiert werden.
- Die Differenzdrehzahl muss von der PLC für die Folgespindel durch Setzen des NST "Freigabe Überlagerung" (DB31, ... DBX26.4) freigegeben werden. Ist die Freigabe nicht vorhanden, dann wird der Alarm 16771 "Kanal% Folge-Achse% überlagerte Bewegung nicht freigegeben" gemeldet. Dieser Alarm wird mit dem Setzen des NST "Freigabe Überlagerung" (DB31, ... DBX26.4) oder mit der Beendigung der Kopplung gelöscht.

Hinweis

Die Differenzdrehzahl wirkt sich nicht auf das Einkoppelverhalten aus.

Ein Getriebestufenwechsel ist während aktiver Kopplung für die Folge- und Leitspindel nicht möglich.

Kopplung mit COUPONC einschalten

Beim Einschalten der Kopplung wird die Folgespindel wie bisher unter Berücksichtigung des Koppelfaktors auf die Drehzahl der Leitspindel beschleunigt. Befand sich die Folgespindel vor dem Einkoppeln in Drehung (M3, M4) so wird diese Bewegung von der Kopplung übernommen.

Kopplung ausschalten

Wird die Kopplung ausgeschaltet, so dreht die Folgespindel mit der Drehzahlsumme aus beiden Drehzahlanteilen weiter. Die Spindel verhält sich so, als wäre sie mit der übernommenen Drehzahl und Drehrichtung programmiert. Beim Ausschalten gibt es keine Unterschiede zum bisherigen Verhalten.

Differenzdrehzahl

Eine Differenzdrehzahl entsteht durch **erneute** Programmierung der Folgespindel (im Beispiel S2=...) oder M2=3, M2=4 im Drehzahlsteuerbetrieb **während** einer aktiver Synchronspindelkopplung oder durch Übernahme der Drehzahl der Folgespindel bei COUONC.

Randbedingung:

Mit der Drehrichtung M3 bzw. M4 muss auch die Drehzahl S... neu programmiert werden. Anderenfalls erfolgt der Alarm 16111 " Kanal% Satz% Spindel% keine Drehzahl programmiert".

Offsets der Folgespindel lesen

Bei der Differenzdrehzahlprogrammierung ändert sich immer der aktuelle Offset. Der aktuelle Offset kann mit $AA_COUP_OFFS[Sn] sollwertseitig und mit $VA_COUP_OFFS[Sn] istwertseitig gelesen werden.

Den zuletzt programmierten Offset liefert die Variable $P_COUP_OFFS[Sn].

Differenzdrehzahl anzeigen

Als Drehzahlsollwert wird bei der programmierten Differenzdrehzahl (entspricht im Beispiel 100 rpm) der programmierte Differenzanteil angezeigt.

Der Drehzahlistwert bezieht sich auf die Motordrehzahl. Im Beispiel ergibt sich die Istdrehzahl zu 500 rpm * (-1) + 100 rpm = -400 rpm.

NST NCK zum PLC

Folgespindel im Drehzahlsteuerbetrieb

Das NST "Spindel im Sollbereich" (DB31, ... DBX83.5) wird für die Folgespindel vom NCK gesetzt, wenn die programmierte Differenzdrehzahl (siehe vorheriges Beispiel N26 mit M2=3 S2=100) erreicht wird. Ist eine Differenzdrehzahl programmiert und nicht von der PLC freigegeben, so wird dieses VDI-Nahtstellensignal auch nicht gesetzt.

Die Folgespindel bleibt auch nach einer Differenzdrehzahlprogrammierung in Lageregelung, wenn diese durch die bestehende Kopplung benötigt wird.

Hinweis

Es wird das axiale VDI-Nahtstellensignal NCK → PLC NST "Überlagerte Bewegung" (DB31, ... DBX98.4) gesetzt, wenn durch Differenzdrehzahlprogrammierung zusätzliche Sollwerte zu den Kopplungssollwerten erzeugt werden.

Istdrehrichtung rechts (DB31, ... DBX83.7)

Das NST "Istdrehrichtung rechts" (DB31, ... DBX83.7) bezieht sich auf die resultierende Motordrehrichtung.

NST PLC zum NCK

Einfluss über PLC-Nahtstelle auf die Folgespindel

Nachfolgend wird die Wirkung der axialen VDI-Nahtstellensignale auf die Folgespindel mit einer Differenzdrehzahl im Drehzahlsteuerbetrieb beschrieben:

Restweg löschen/Spindel-Reset (DB31, ... DBX2.2)

Die programmierte Differenzdrehzahl und Drehrichtung kann durch das NST "Restweg löschen/Spindel Reset" (DB31, ... DBX2.2) beendet werden. Soll nur die programmierte Drehzahl gelöscht werden, kann dies durch Setzen von NST "S-Wert löschen" (DB31, ... DBX16.7) erreicht werden.

Spindel neu synchronisieren 1/2 (DB31, ... DBX16.4 u. 16.5)

Die NST "Spindel neu synchronisieren 1/2 " (DB31, ... DBX16.4/16.5) sind **nicht** verriegelt. Ein sich ergebener Positionsversatz wird nicht automatisch durch die Kopplung ausgeglichen.

M3/M4 invertieren (DB31, ... DBX17.6)

Das NST "M3/M4 invertieren" (DB31, ... DBX17.6) invertiert nur den für die Folgespindel zusätzlich programmierten Drehzahlanteil.

Der durch die Sychronspindelkopplung erzeugte Bewegungsanteil bleibt davonunbeeinflusst.

Spindelkorrektur (DB31, ... DBB19)

Die "Spindelkorrektur" VDI-Nahtstelle (DB31, ... DBB19) wirkt nur auf den für die Folgespindel zusätzlich programmierten Drehzahlanteil. Wenn der Spindelkorrekturschalter auf alle axialen Eingänge übertragen wird, dann wirkt sich eine Änderung des Spindelkorrekturwertes für die Folgespindel **doppelt** aus:

- einmal indirekt durch die Drehzahländerung der Leitspindel und
- einmal im programmierten Anteil der Folgespindel.

Im PLC-Programm kann der Korrekturwert entsprechend angepasst werden.

Kupplungsabwahl

Wird die Kopplung ausgeschaltet, so dreht die Folgespindel mit der Summe aus beiden Drehzahlanteilen weiter. Der Bewegungsübergang bei Kopplungsabwahl ist drehzahlstetig.

Bei COUPOF verhält sich die Spindel so als wäre sie mit der übernommenen Drehzahl und Drehrichtung programmiert worden. Im Beispiel entspricht dies M4 S400.

Bei COUPOFS wird die Folgespindel aus der aktuellen Drehzahl stillgesetzt.

Zusätzliche Funktionen aktivieren

Die Folgespindel kann auch Masterspindel sein. In diesem Fall sind zusätzliche Funktionen möglich.

- Umdrehungsvorschub bei G95, G96, G97. Mit G96 S2=... kann die Funktion "konstante Schnittgeschwindigkeit" für die Folgespindel aktiviert werden.

  Die von der Position der Planachse abhängige Drehzahl ist die Solldrehzahl für den Drehzahlinterpolator der Spindel 2. und geht damit additiv in die Gesamtdrehzahl von S2 ein.
- Gewindebohren ohne Ausgleichsfutter mit G331, G332.

14.4.5 Verhalten der Synchronlaufsignale bei Synchronlaufkorrektur

Wirkung der Synchronlaufkorrektur

Durch Vergleich der Istwerte mit den korrigierten Sollwerten werden die neuen Synchronlaufsignale gebildet. Nach einem erfolgten Korrekturvorgang sollten die Synchronlaufsignale wieder anstehen.

14.4.6 Synchronlaufkorrektur ablöschen und NC-Reset

Für verschiedene Situationen, in denen der Synchronlaufkorrekturwert abgelöscht wird, liefert Variable $AA_COUP_CORR[Sn] den Wert Null zurück:

- Mit dem Einschalten einer Synchronspindelkopplung für die betreffende Folgespindel mit COUPON(..)/COUPONC(..) wird eine bestehende Synchronlaufkorrektur in die Sollposition übernommen.
- Eine bestehende Synchronlaufkorrektur wird bei NC-Reset jedoch nicht beim Teileprogramm-Ende in die Sollposition übernommen. Die Synchronlaufsignale werden davon nicht beeinflusst.
- Bei M30 bleibt eine bestehende Synchronlaufkorrektur bestehen.
- Anwenderseitig kann der Korrekturwert auch durch Beschreiben der Variablen $AA_COUP_CORR mit dem **Wert Null** zu einem früheren Zeitpunkt gelöscht werden. Die Synchronlaufkorrektur wird sofort und bei größeren Werten über eine Rampe mit reduziertem Beschleunigungsvermögen abgebaut.

14.4.7 Besonderheiten bei der Inbetriebnahme der Synchronspindelkopplung

Spindel-Inbetriebnahme

Zunächst sind die Leit- und Folgespindel wie eine normale Spindel in Betrieb zu nehmen. Das entsprechende Vorgehen ist beschrieben in:
Literatur:
/IADC/ Inbetriebnahmehandbuch SINUMERIK 840D
/FB1/ Funktionshandbuch Grundfunktionen, Spindeln (S1)

Festlegungen

Anschließend sind für das Synchronspindelpaar folgende Festlegungen zu treffen:

- Maschinenachsnummern für die Leit- und Folgespindel

  (bei der fest projektierten Kopplung mit dem kanalspezifischen Maschinendatum MD21300 MC_COUPLE_AXIS_1[n])

- benötigte Kopplungsart (Sollwert-, Istwert- oder Geschwindigkeitskopplung)

  (bei der fest projektierten Kopplung mit dem kanalspezifischen Maschinendatum MD21310 $MC_COUPLING_MODE_1[n])

- Getriebestufe (n) von FS und LS für den Synchronbetrieb einlegen
- bei der fest projektierten Synchronspindelkopplung noch folgende Kopplungseigenschaften:
  - Satzwechselverhalten im Synchronspindelbetrieb:

    MD21320 $MC_COUPLE_BLOCK_CHANGE_CTRL_1
  - Kopplungs-Abbruch-Verhalten:

    MD21330 $MC_COUPLE_RESET_MODE_1
  - Änderungsschutz der Kopplungsparameter:

    MD21340 $MC_COUPLE_IS_WRITE_PROT_1
  - Übersetzungsparameter für Synchronspindelkopplung:

    SD42300 $SC_COUPLE_RATIO_1[n]

Führungsverhalten der FS und LS bei Sollwertkopplung

Um bei Anwendung der **Sollwertkopplung** den bestmöglichen Synchronlauf zu erzielen, müssen die FS und LS für das Führungsverhalten die gleiche Dynamik aufweisen. Die axialen Regelkreise (Lage-, Drehzahl- und Stromregler) sollten jeweils **optimal** eingestellt werden, um Abweichungen möglichst schnell und effizient ausregeln zu können.

Zur Anpassung von unterschiedlichen Achsdynamiken ohne Verlust von Regelgüte dient die **Dynamikanpassung** im Sollwertzweig. Folgende Regelungsparametern sind für die FS und LS jeweils axial optimal einzustellen:

- K_V-Faktor (MD32200 $MA_POSCTRL_GAIN)
- Vorsteuerungsparameter

  MD32620 $MA_FFW_MODE

  MD32610 $MA_VELO_FFW_WEIGHT

  MD32650 $MA_AX_INERTIA

  MD32800 $MA_EQUIV_CURRCTRL_TIME

  MD32810 $MA_EQUIV_SPEEDCTRL_TIME

Literatur:
/FB2/ Funktionshandbuch Erweiterungsfunktionen, Kompensationen (K3)

Verhalten bei Verlust des Synchronlaufs:

- achsspezifisches MD32620 $MA_FFW_MODE

Es wird empfohlen, die **Vorsteuerungsart** der Folgeachse auf Drehzahlvorsteuerung mit Tt-Symmetrierung MD32620 = 3 zu setzen.

Dieser Vorsteuermodus kann für einen sicheren Symmetriervorgang durch Veränderung achsspezifischer Maschinendaten weiter optimiert werden:

Maschinendatum	Bedeutung
MD32810 EQUIV_SPEEDCTRL_TIME	Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung
MD37200 COUPLE_POS_TOL_COURSE	Schwellwert für 'Synchronlauf grob'
MD37210 COUPLE_POS_TOL_FINE	Schwellwert für 'Synchronlauf fein'
MD37220 COUPLE_VELO_TOL_COURSE	Geschwindigkeitstoleranz 'grob'
MD37220 COUPLE_VELO_TOL_FINE	Geschwindigkeitstoleranz 'fein'

Hierbei führen höhere Schwellwerte für die Synchronlaufsignale und größere Positions-bzw. Geschwindigkeitstoleranzen zu stabileren Ergebnissen.

Dynamikanpassung

Die FS und die gekoppelte LS müssen für das Führungsverhalten die gleiche Dynamik aufweisen. Gleiche Dynamik heißt, die Schleppabstände sind bei gleicher Drehzahl gleich groß.

Mit der Dynamikanpassung im Sollwertzweig lässt sich eine sehr gute Angleichung des Führungsverhaltens von dynamisch unterschiedlichen Achsen (Regelkreise) erzielen. Als Zeitkonstante der Dynamikanpassung ist die Differenz der Ersatzzeitkonstanten zwischen der dynamisch "schwächsten" Spindel zu der jeweiligen anderen Spindel einzugeben.

Beispiel

Bei aktiver Drehzahlvorsteuerung wird die Dynamik im Wesentlichen durch die Ersatzzeitkonstanten des "langsamsten" Drehzahlregelkreises bestimmt.

Leitspindel:

MD32810 $MA_EQUIV_SPEEDCTRL_TIME [n] = 5 ms

Folgespindel:

MD32810 $MA_EQUIV_SPEEDCTRL_TIME [n] = 3 ms

Zeitkonstante der Dynamikanpassung für Folgespindel:

MD32910 $MA_DYN_MATCH_TIME [n] = 5 ms - 3 ms = 2 ms

Die Dynamikanpassung muss axial mit dem MD32900 $MA_DYN_MATCH_ENABLE aktiviert werden.

Die vorgenommene Dynamikanpassung kann durch Vergleich der Schleppabstände von FS und LS kontrolliert werden (im Bedienbereich Diagnose; Bild Service-Achsen). Die Schleppabstände müssen bei gleicher Drehzahl gleich groß sein!

Zur Feinabstimmung kann es notwendig sein, K_V-Faktoren oder Vorsteuerparameter geringfügig anzugleichen, um ein optimales Ergebnis zu erzielen.

Regelungs-Parametersätze

Bei Spindeln wird jeder Getriebestufe ein eigener Parametersatz für die Lageregelung zugeordnet.

Mit Hilfe der Parametersätze kann beispielsweise die Anpassung der Dynamik für die Leitspindel an die Folgespindel im Synchronbetrieb erfolgen.

Bei ausgeschalteter Kopplung (Drehzahl- oder Positionierbetrieb) können somit für die FS und LS andere Lagereglerparameter angewählt werden. Um diese Möglichkeit zu nutzen, ist für den Synchronbetrieb eine eigene Getriebestufe zu verwenden, die vor Anwahl des Synchronbetriebes zu aktivieren ist.

Die Kopplungsparameter der Folgespindel können mit diesem Maschinendatum wie folgt eingestellt werden:

MD30455 $MA_MISC_FUNCTION_MASK

Bit 5=0: Synchronspindelkopplung, Folgespindel:

Lageregelung, Vorsteuerung und Parametersatz werden wie bei der Folgespindel eingestellt.

Bit 5=1: Synchronspindelkopplung:

Die Regel-Parameter der Folgespindel wie im ungekoppelten Fall eingestellt.

Folgende Regelungsparametern müssen für die FS und LS gleich eingestellt werden:

- Feininterpolatortyp (MD33000 $MA_FIPO_TYPE)
- Axiale Ruckbegrenzung

  MD32400 $MA_AX_JERK_ENABLE

  MD32410 $MA_AX_JERK_TIME

  MD32420 $MA_JOG_AND_POS_JERK_ENABLE

  MD32430 $MA_JOG_AND_POS_MAX_JERK

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen, Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (G2)

Getrennte Dynamik für Spindel- und Achsbetrieb

Im Spindel- und Achsbetrieb kann Dynamikprogrammierung FA, OVRA, ACC und VELOLIMA mit folgendem MD getrennt voneinander eingestellt werden:

MD30455 $MA_MISK_FUNCTION_MASK Bit 6=0

Die Zuordnung erfolgt durch den programmierten Achs- oder Spindelbezeichner. So reduziert z. B. VELOLIMA[S1]=50 nur im Spindelbetrieb die Maximaldrehzahl auf 50% und VELOLIMA[C]=50 nur im Achsbetrieb die Maximalgeschwindigkeit auf 50%.

Soll z. B. VELOLIMA[S1]=50 und VELOLIMA[C]=50 die gleiche Wirkung haben, so wirkt die Programmierung von FA, OVRA, ACC und VELOLIM wie bisher mit diesem Maschinendatum unabhängig vom programmierten Bezeichner gemeinsam:

MD30455 $MA_MISK_FUNCTION_MASK Bit 6=1

Geknickte Beschleunigungskennlinie

Für die Leitspindel wird der Einfluss einer geknickten Beschleunigungskennlinie auf die Folgespindel durch folgende achsspezifische Maschinendaten gekennzeichnet:

MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT (Drehzahl reduz. Beschl.) und

MD35230 $MA_ACCEL_REDUCTION_FACTOR (Reduzierte Beschleunigung).

Sofern das MD35242 $MA_ACCEL_REDUCTION_TYPE vorhanden ist, wird es zur Projektierung der Art der Beschleunigungsreduktion auch hinzugezogen. Andernfalls wird ein hyperbolischer Beschleunigungsabfall angenommen.

Ist unter Berücksichtigung des Kopplungsfaktors die Dynamik einer Folgespindel kleiner als die der Leitspindel, wird die Dynamik der Leitspindel für die Dauer der Kopplung auf das erforderliche Maß reduziert.

Für die Folgespindel sollte die Beschleunigung über den gesamten Drehzahlbereich **konstant** sein. Falls dennoch auch für die Folgespindel eine geknickte Beschleunigungskennlinie in den o. g. Maschinendaten hinterlegt ist, so wird diese nur beim Einkoppelvorgang berücksichtigt. Die Sollwerte der Folgespindel werden für die vorgegebene geknickte Beschleunigungskennlinie eingehalten.

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen, Beschleunigung (B2), Kapitel Geknickte Beschleunigungskennlinie

Istwertkopplung

Bei Anwendung der Istwertkopplung (AV) sollte der Antrieb der FS wesentlich dynamischer sein als der Leitspindel-Antrieb. Die einzelnen Antriebe werden auch hier gemäß ihrer Antriebsdynamik optimal eingestellt.

Die Istwertkopplung sollte nur in Sonderfällen angewendet werden.

Geschwindigkeitskopplung

Die Geschwindigkeitskopplung (VV) ist intern eine Sollwertkopplung (DV) mit geringeren dynamischen Anforderungen an die FS und LS. Lageregelung ist für FS und /oder LS nicht notwendig. Messsysteme sind nicht erforderlich.

Schwellwerte für Synchronlauf grob/fein

Nach Optimierung der Regler und Einstellung der Vorsteuerung sind für die Folgespindel die Schwellwerte für Synchronlauf grob und fein einzutragen.

- Schwellwert für "Synchronlauf grob"

  achsspez. MD7200: AV, DV: COUPLE_POS_TOL_COARSE

  MD37220: VV: COUPLE_VELO_TOL_COARSE

- Schwellwert für "Synchronlauf fein"

  achsspez. MD37210: AV, DV: COUPLE_POS_TOL_FINE

  MD37230: VV: COUPLE_VELO_TOL_FINE

Die Werte der FS sind je nach geforderter Genauigkeit des Maschinenherstellers zu ermitteln und die PLC-Nahtstelle über die Service-Anzeige zu kontrollieren.

Winkelversatz LS/FS

Falls ein definierter Winkelversatz zwischen FS und LS z. B. beim Einschalten des Synchronbetriebs notwendig ist, müssen die "Null-Grad-Positionen" von FS und LS aneinander angepasst werden. Dies ist mit folgenden Maschinendaten möglich:

- MD34100 $MA_REFP_SET_POS
- MD34080 $MA_REFP_MOVE_DIST
- MD34090 $MA_REFP_ MOVE_DIST_CORR

Literatur:

/FB1/ Funktionshandbuch Grundfunktionen, Referenzpunktfahren (R1)

Service-Anzeige für FS

Im Bedienbereich "Diagnose" im Bild "Servicewerte Achsen" werden für die Inbetriebnahme im Synchronbetrieb bei der Folgespindel folgende Werte angezeigt:

- tatsächliche Abweichung der Sollwerte von FS und LS

  Anzeigewert: Positionsoffset zur Leitspindel (Sollwert)

  (Wert entspricht dem mit der Achsvariablen $AA_COUP_OFFS im Teileprogramm lesbaren Winkelversatz zwischen FS und LS)

- tatsächliche Abweichung der Istwerte von FS und LS

  Anzeigewert: Positionsoffset zur Leitspindel (Istwert)

Literatur:

/BAD, BEM/ HMI Advanced/Embedded, Service-Anzeige Kapitel "Service Antrieb"

14.5 Beispiele

Programmierbeispiel

```
                                    ; Leitspindel = Masterspindel = Spindel 1
                                    ; Folgespindel = Spindel 2
N05 M3 S3000 M2=4 S2=500            ; Leitspindel dreht mit 3000/min
                                    ; FS: 500/min.
N10 COUPDEF (S2, S1, 1, 1, "No",    ; Def. der Kopplung kann auch
"Dv")                               ; projektiert werden
N70 SPCON                           ; Leitspindel in Lageregelung nehmen
                                    ; (Sollwertkopplung).
N75 SPCON(2)                        ; Folgespindel in Lageregelung nehmen
N80 COUPON (S2, S1, 45)             ; Fliegend auf Offsetposition = 45°
                                    einkoppeln
N200 FA [S2] = 100                  ; Positioniergeschwindigkeit = 100 grd/min
N205 SPOS[2] = IC(-90)              ; 90° überlagert in negative Richtung fahren
N210 WAITC(S2, "Fine")              ; Warten auf Synchronlauf "fein"
N212 G1 X.., Y.. F...               ; Bearbeitung
N215 SPOS[2] = IC(180)              ; 180° überlagert in positive Richtung fahren
N220 G4 S50                         ; Verweilzeit = 50 Umdrehungen
                                    ; der Masterspindel
N225 FA [S2] = 0                    ; Projektierte Geschw. (MD) aktivieren
N230 SPOS[2] = IC (-7200)           ; 20 Umdr. mit projek. Geschwindigkeit
                                    ; in neg. Richtung
N350 COUPOF (S2, S1)                ; Fliegend auskoppeln, S = S2 = 3000
N355 SPOSA[2] = 0                   ; FS bei Null Grad stoppen
N360 G0 X0 Y0
N365 WAITS(2)                       ; Warten auf Spindel 2
N370 M5                             ; FS stoppen
N375 M30
```

14.6 Datenlisten

14.6.1 Maschinendaten

14.6.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
10000	AXCONF_MACHAX_NAME_TAB	Maschinenachsname

14.6.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
20070	AXCONF_MACHAX_USED	Maschinenachsnummer gültig im Kanal
21300	COUPLE_AXIS_1	Synchronspindelpaar-Festlegung
21310	COUPLING_MODE_1	Kopplungsart im Synchronspindelbetrieb
21320	COUPLE_BLOCK_CHANGE_CTRL_1	Satzwechselverhalten im Synchronspindelbetrieb
21330	COUPLE_RESET_MODE_1	Kopplungs-Abbruch-Verhalten
21340	COUPLE_IS_WRITE_PROT_1	Änderungen der Kopplungsparameter nicht möglich

14.6.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: $MA_	Beschreibung
30455	MISK_FUNKTION_MASK	Achsfunktionen
30550	AXCONF_ASSIGN_MASTER_CHAN	Löschstellung des Kanals für Achswechsel
32200	POSCTRL_GAIN	KV-Faktor
32400	AX_JERK_ENABLE	Axiale Ruckbegrenzung
32410	AX_JERK_TIME	Zeitkonstante für den axialen Ruckfilter
32420	JOG_AND_POS_JERK_ENABLE	Grundeinstellung der axialen Ruckbegrenzung
32430	JOG_AND_POS_MAX_JERK	Axialer Ruck
32610	VELO_FFW_WEIGHT	Vorsteuerfaktor für Drehzahlvorsteuerung
32620	FFW_MODE	Vorsteuerungsart
32650	AX_INERTIA	Trägheitsmoment für Drehmomentvorsteuerung
32800	EQUIV_CURRCTRL_TIME	Ersatzzeitkonstante Stromregelkreis für Vorsteuerung
32810	EQUIV_SPEEDCTRL_TIME	Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung
34080	REFP_MOVE_DIST	Referenzpunktabstand
34090	REFP_MOVE_DIST_CORR	Referenzpunktverschiebung
34100	REFP_SET_POS	Referenzpunktwert
35000	SPIND_ASSIGN_TO_MACHAX	Zuordnung Spindel zu Maschinenachse

Nummer	Bezeichner: $MA_	Beschreibung
37200	COUPLE_POS_TOL_COARSE	Schwellwert für "Synchronlauf grob"
37210	COUPLE_POS_TOL_FINE	Schwellwert für "Synchronlauf fein"
37220	COUPLE_VELO_TOL_COARSE	Geschwindigkeitstoleranz "grob" zwischen Leit- und Folgespindel
37230	COUPLE_VELO_TOL_FINE	Geschwindigkeitstoleranz "fein" zwischen Leit- und Folgespindel

14.6.2 Settingdaten

14.6.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: $SC_	Beschreibung
42300	COUPLE_RATIO_1	Übersetzungsparameter für Synchronspindelbetrieb

14.6.3 Signale

14.6.3.1 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
NC-Start	DB21, ...DBX7.1	DB3200.DBX7.1
NC-Stopp Achsen plus Spindel	DB21, ...DBX7.4	DB3200.DBX7.4

14.6.3.2 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Probelaufvorschub angewählt	DB21, ...DBX24.6	DB1700.DBX0.6
Vorschubkorrektur für Eilgang angewählt	DB21, ...DBX25.3	DB1700.DBX1.3

14.6.3.3 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Achsen-/Spindelsperre	DB31, ...DBX1.3	DB380x.DBX1.3
Nachführbetrieb	DB31, ...DBX1.4	DB380x.DBX1.4
Lagemesssystem 1, Lagemesssystem 2	DB31, ...DBX1.5/6	DB380x.DBX1.5/6
Reglerfreigabe	DB31, ...DBX2.1	DB380x.DBX2.1
Restweg/Spindel-RESET	DB31, ...DBX2.2	DB380x.DBX2.2

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Spindel-Halt/Vorschub-Halt	DB31, ...DBX4.3	DB380x.DBX4.3
Verfahrtasten bei JOG	DB31, ...DBX4.6/7	DB380x.DBX4.6/7
Spindel neu synchronisieren 1, Spindel neu synchronisieren 2	DB31, ...DBX16.4/5	DB380x.DBX2000.4/5
S-Wert löschen	DB31, ...DBX16.7	DB380x.DBX2000.7
Vorschubkorrektur gültig	DB31, ...DBX17.0	DB380x.DBX2001.0
M3/M4 invertieren	DB31, ...DBX17.6	DB380x.DBX2001.6
Spindelkorrektur	DB31, ...DBX19	DB380x.DBB2003
Neu synchronisieren	DB31, ...DBX31.4	-
Synchronisation sperren	DB31, ...DBX31.5	-

14.6.3.4 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Referiert/Synchronisiert 1, Referiert/Synchronisiert 2	DB31, ...DBX60.4/5	DB390x.DBX0.4/5
Synchronbetrieb	DB31, ...DBX84.4	DB390x.DBX2002.4
Synchronlauf fein	DB31, ...DBX98.0	-
Synchronlauf grob	DB31, ...DBX98.1	-
Istwertkopplung	DB31, ...DBX98.2	-
Überlagerte Bewegung	DB31, ...DBX98.4	DB390x.DBX5002.4
Leitspindel LS/LA aktiv	DB31, ...DBX99.0	-
Folgespindel FS/FA aktiv	DB31, ...DBX99.1	-

14.6.4 Systemvariablen

14.6.4.1 Systemvariablen

Systemvariable	Beschreibung
$P_COUP_OFFS[Folgespindel]	Programmierter Offset der Synchronspindel
$AA_COUP_OFFS[Folgespindel]	Positions-Offset für Synchronspindel sollwertseitig
$VA_COUP_OFFS[Folgespindel]	Positions-Offset für Synchronspindel istwertseitig

Ausführliche Erläuterungen zu den Systemvariablen finden sich in:
Literatur:
/PGA1/ Listenhandbuch Systemvariablen

S7: Speicherkonfiguration 15

15.1 Kurzbeschreibung

Speicherarten

Zur Datenhaltung und Datenverwaltung benötigt die NC einen statischen und einen dynamischen Speicher:

- **Statischer NC-Speicher**

  Im statischen NC-Speicher werden die Programmdaten (Teileprogramme, Zyklen, ...) und die aktuellen System- und Anwenderdaten (Werkzeugverwaltung, Globale User-Daten, ...) **persistent** gespeichert.

- **Dynamischer NC-Speicher**

  Im dynamischen NC-Speicher werden Daten **nicht-persistent** gespeichert. Dabei handelt es sich um Daten, die von der NC erzeugt und nur für eine bestimmte Zeit benötigt werden (z. B. Makros, Lokale User-Daten, Interpolationspuffer, ...).

Speicherorganisation

Die Speicherbereiche der einzelnen Datengruppen im statischen und dynamischen NC-Speicher besitzen definierte Größen, die im Verlauf der Speicherkonfiguration fest eingestellt werden.

Diese Form der Speicherorganisation ermöglicht ein **deterministisches** Verhalten der Steuerung: zu jedem Zeitpunkt der Teileprogrammabarbeitung ist der reservierte Speicherbereich garantiert.

Speicherkonfiguration

Beim ersten Systemhochlauf werden die speicherkonfigurierenden Maschinendaten mit Standardwerten besetzt. Im Normalfall ist diese Konfiguration ausreichend. Eine Änderung durch den Maschinenhersteller/Anwender ist aber jederzeit möglich (**Neukonfiguration**).

15.2 Speicherorganisation

15.2.1 Aktives und passives Filesystem

Im statischen NC-Speicher befinden sich das aktive und passive Filesystem.

Aktives Filesystem

Das aktive Filesystem umfasst die Systemdaten zur Parametrierung des NCK:

- Maschinendaten
- Settingdaten
- Optionsdaten
- Globale Anwenderdaten (GUD)
- Werkzeugkorrektur-/Magazin-Daten
- Schutzbereiche
- R-Parameter
- Nullpunktverschiebungen/FRAME
- Durchhangkompensationen
- Quadrantenfehlerkompensation
- Spindelsteigungsfehler-Kompensation

Diese Daten repräsentieren die aktuellen Arbeitsdaten des NCK.

Die Anwendersicht auf das aktive Filesystem ist Daten-orientiert.

Passives Filesystem

Das passive Filesystem umfasst alle im NCK geladenen Dateien:

- Hauptprogramme
- Unterprogramme
- Definitionsdateien (*.DEF) der globalen Anwenderdaten und Makros
- Standard-Zyklen
- Anwender-Zyklen
- Werkstücke
- Kommentare

Die Anwendersicht auf das passive Filesystem ist File-orientiert.

15.2.2 Neukonfiguration

Neukonfiguration

Folgende Handlungen führen zu einer Neukonfiguration des statischen und/oder dynamischen NC-Speichers:

- Ändern der Einstellungen der speicherkonfigurierenden Maschinendaten:

  MD... $..._**MM**_...
- Ändern der Kanalanzahl

Schutz vor Datenverlust

ACHTUNG
Eine Neukonfiguration des **statischen** NC-Speichers hat den Verlust der Daten des aktiven und passiven Filesystems zur Folge. Vor der Aktivierung der geänderten Speicherkonfiguration müssen die Daten deshalb durch Erstellen einer **Serieninbetriebnahme-Datei** gesichert werden.

15.3 Konfiguration des statischen Anwenderspeichers

15.3.1 Aufteilung des statischen NC-Speichers

Das folgende Bild zeigt die Aufteilung des statischen NC-Speichers bei SINUMERIK 840D sl:

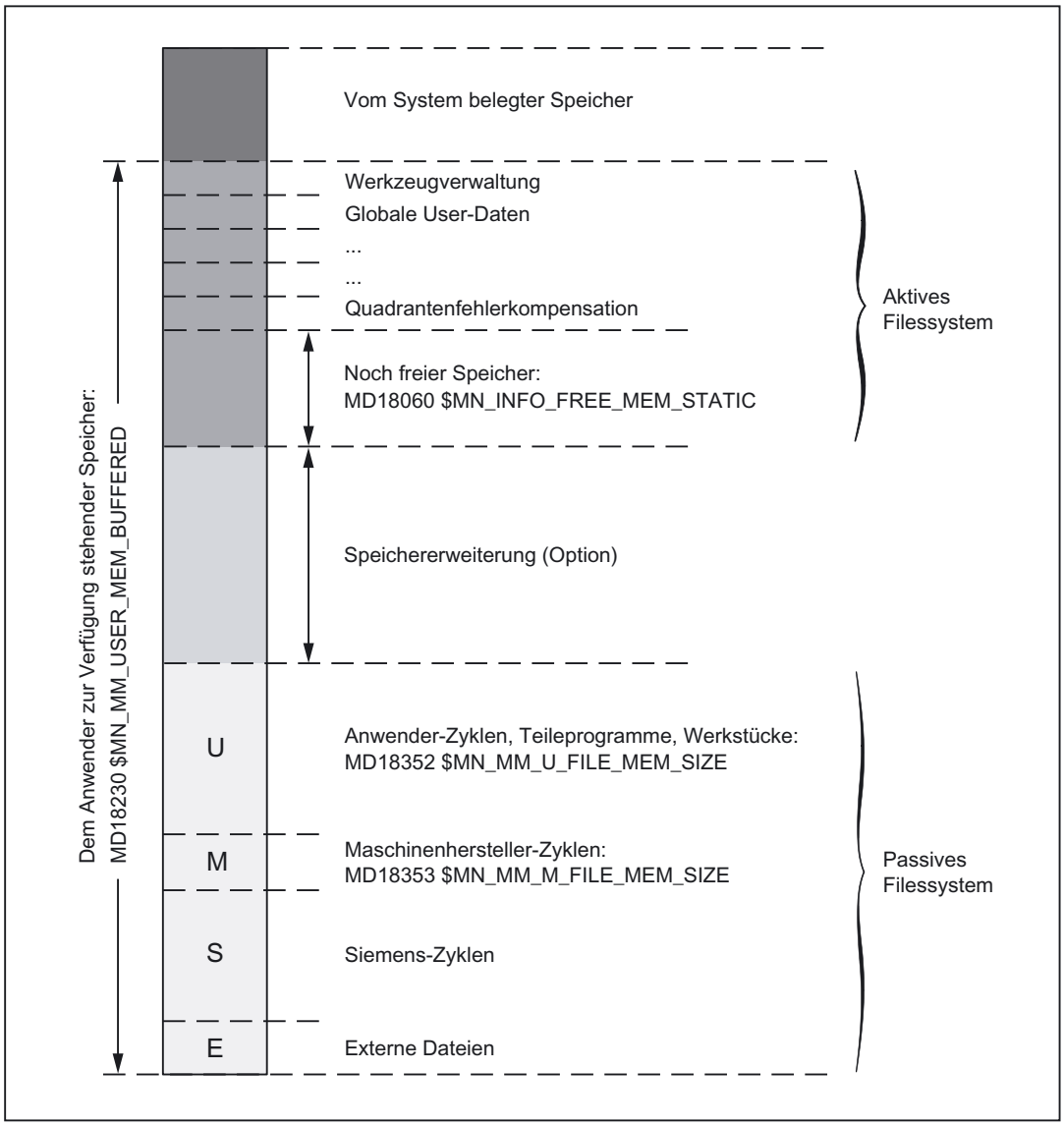


Bild 15-1 Statischer NC-Speicher bei SINUMERIK 840D sl

Statischer Anwenderspeicher

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.

Größe des statischen Anwenderspeichers

Die Größe des statischen Anwenderspeichers wird eingestellt im Maschinendatum:

MD18230 $MN_MM_USER_MEM_BUFFERED

Speicherplatz für passives Filesystem

Der Speicherplatz für das passive Filesystem hat eine definierte Größe und ist in folgende Partitionen unterteilt:

Partition	Speicherung von:
S (**S**iemens = Steuerungshersteller)	Dateien aus dem Verzeichnis _N_CST_DIR (Siemens-Zyklen)
M (**M**anufacturer = Maschinenhersteller)	Dateien aus dem Verzeichnis _N_CMA_DIR (Maschinenhersteller-Zyklen)
U (**U**ser = Endkunde)	Dateien aus dem Verzeichnis _N_CUS_DIR (Anwender-Zyklen, Teileprogramme, Werkstücke)
E (**E**XT = extern)	Externen Dateien (z. B. Teileprogrammen von USB-FlashDrive oder CD). **Hinweis:** Um Namenskonflikte mit internen Teileprogrammen zu vermeiden, werden die von externen Datenträgern geladenen Teileprogramme in einem separaten Verzeichnis (_N_EXT_DIR) abgelegt.

Vorteil der Unterteilung:

Durch die Unterteilung in verschiedene Partitionen wird sichergestellt, dass eine Serieninbetriebnahme-Datei auch nach einer Hochrüstung der NCK-Software oder eines Zyklenpakets (bei der sich der Bereich der Siemens-Zyklen vergrößert hat) wieder in den NCK geladen werden kann.

Größe der Partitionen:

Die Größen der Partitionen **S** und **E** sind voreingestellt und nicht veränderbar.

Der restliche für das passive Filesystem zur Verfügung stehende Speicher kann beliebig auf die Partitionen **U** und **M** aufgeteilt werden. Die Einstellungen erfolgen mit den Maschinendaten:

MD18352 $MN_MM_U_FILE_MEM_SIZE
(Endanwenderspeicher für Teileprogramme/Zyklen/Dateien)

MD18353 $MN_MM_M_FILE_MEM_SIZE
(Speichergröße für Zyklen/Dateien des Maschinenherstellers)

Die maximal einstellbaren Werte sind abhängig:

- vom System und dem damit zur Verfügung stehenden Speicherplatz (inklusive optionaler Speichererweiterung)
- von den definierten Maximalwerten (siehe "Ausführliche MD-Beschreibung")

Speicherplatz für aktives Filesystem

Der Speicherplatz für das aktive Filesystem ist in verschiedene Datenbereiche unterteilt (Werkzeugverwaltung, Globale User-Daten, ...), die einzeln über Maschinendaten eingestellt werden können.

Der noch freie Speicher wird angezeigt im Maschinendatum:

MD18060 $MN_INFO_FREE_MEM_STATIC
(Anzeigedatum des freien statischen Speichers)

Die Größen der einzelnen Speicherbereiche für das aktive Filesystem können über die entsprechenden Maschinendaten soweit vergrößert werden, bis der verfügbare Speicher aufgebraucht ist.

Hinweis

In 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)

Der Maschinenhersteller kann zusätzlichen statischen Anwenderspeicher erwerben (Option "Erweiterung des gepufferten CNC-Anwenderspeichers").

Der zusätzliche Speicherplatz kann je nach Bedarf zur Vergrößerung der Partitionen U und M oder zur Vergrößerung des Speicherbereichs für das aktive Filesystem verwendet werden.

15.3.2 Inbetriebnahme

Vorgehensweise

1. Standardmaschinendaten laden.
2. Bei erworbener Option "Erweiterung des gepufferten CNC-Anwenderspeichers":

   Option freischalten.
3. Maschinendatum:
   MD18230 $MN_MM_USER_MEM_BUFFERED
   mit einem hohen Wert (> standardmäßig vorhandener Speicher + optionaler Zusatzspeicher) vorbesetzen.
4. NCK-Reset ausführen.

   Der Alarm 6030 "Anwenderspeicherlimit wurde angepasst" wird ausgegeben und im MD18230 wird der maximal dem Anwender zur Verfügung stehende Speicher eingetragen (inklusive der optionalen Speichererweiterung). Alle weiteren speicherkonfigurierenden Maschinendaten werden mit ihren Standardwerten besetzt.

5. Größe der Partitionen U und M einstellen in den Maschinendaten:

   MD18352 $MN_MM_U_FILE_MEM_SIZE
   (Endanwenderspeicher für Teileprogramme/Zyklen/Dateien)

   MD18353 $MN_MM_M_FILE_MEM_SIZE
   (Speichergröße für Zyklen/Dateien des Maschinenherstellers)

6. Anzahl der benötigten Kanäle und Achsen aktivieren.

   Literatur:

   Inbetriebnahmeanleitung SINUMERIK 840D
   Handbuch SINUMERIK 840Di

7. Die standardmäßige Speicheraufteilung kann durch Vergrößern/Verkleinern einzelner Speicherbereiche des aktiven Filesystems (Werkzeugverwaltung, Globale User-Daten, ...) anwendungsspezifisch angepasst werden:

   – Freien statischen Anwenderspeicher ablesen im Maschinendatum:

     MD18060 $MN_INFO_FREE_MEM_STATIC
     (Anzeigedatum des freien statischen Speichers)

   – Maschinendaten einstellen (→ Datenlisten).

     Literatur:
     Ausführliche Maschinendaten-Beschreibung

8. NCK-Reset ausführen.

   Der Speicher wird neu eingerichtet.

15.4 Konfiguration des dynamischen Anwenderspeichers

15.4.1 Aufteilung des dynamischen NC-Speichers

Das folgende Bild zeigt die Aufteilung des dynamischen NC-Speichers:

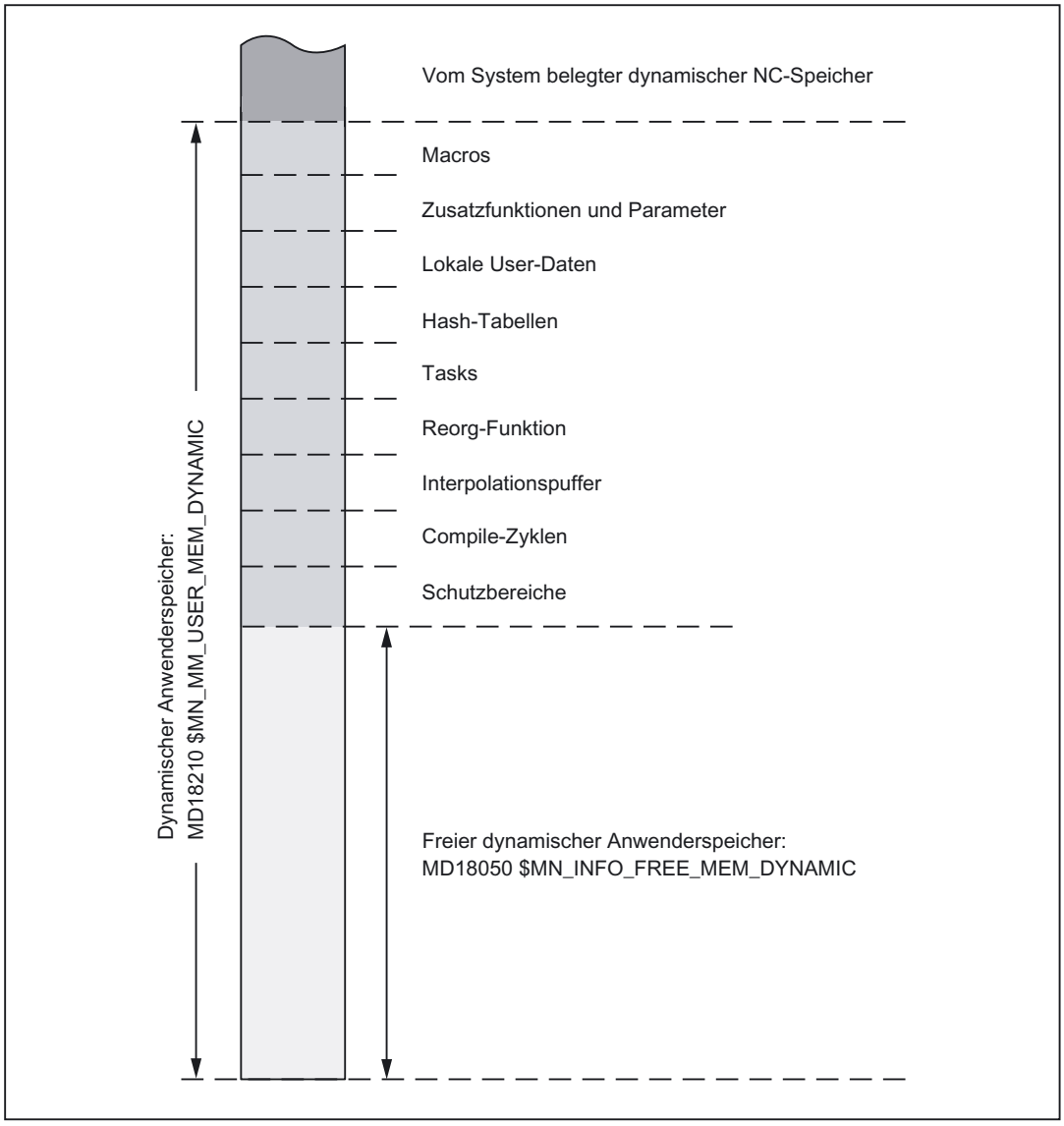


Bild 15-2 Dynamischer NC-Speicher

Dynamischer Anwenderspeicher

Der dynamische NC-Speicher wird vom System und vom Anwender gemeinsam genutzt. Der dem Anwender zur Verfügung stehende Bereich wird als dynamischer Anwenderspeicher bezeichnet.

Größe des dynamischen Anwenderspeichers

Die Größe des dynamischen Anwenderspeichers wird festgelegt mit dem Maschinendatum:

MD18210 $MN_MM_USER_MEM_DYNAMIC

Da bei einer Neukonfiguration automatisch ein passender Wert eingestellt wird, ist eine Änderung in der Regel nicht erforderlich.

Freier dynamischer Anwenderspeicher

Der freie dynamische Anwenderspeicher wird angezeigt im Maschinendatum:

MD18050 $MN_INFO_FREE_MEM_DYNAMIC
(Anzeigedatum des freien dynamischen Speichers)

Der Inhalt dieses Maschinendatums gibt an, wie viel Speicher für die Vergrößerung der Anwenderdatenbereiche (Lokale User-Daten, IPO-Puffer, ...) pro Kanal zur Verfügung steht.

15.4.2 Inbetriebnahme

Die standardmäßige Speicheraufteilung kann durch Vergrößern/Verkleinern einzelner Speicherbereiche anwendungsspezifisch angepasst werden.

Vorgehensweise

1. Freien dynamischen Anwenderspeicher ablesen im Maschinendatum:

   MD18050 $MN_INFO_FREE_MEM_DYNAMIC
   (Anzeigedatum des freien dynamischen Speichers)

2. Maschinendaten einstellen (→ Datenlisten).

   Literatur:
   Ausführliche Maschinendaten-Beschreibung

3. NCK-Reset ausführen.

   Der Speicher wird neu eingerichtet.

15.5 Datenlisten

15.5.1 Maschinendaten

15.5.1.1 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-Codes
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 DUAL-PORT-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 NCK verwalteten Werkzeuge
18084	MM_NUM_MAGAZINE	Anzahl der von der NCK 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 NCK
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
18140	MM_NUM_GUD_NAMES_AXIS	Anzahl der achsspezifischen 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 NCK
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 in DRAM
18220	MM_USER_MEM_DPR	Anwenderspeicher im Dualport-RAM
18230	MM_USER_MEM_BUFFERED	Anwenderspeicher im SRAM
18231	MM_USER_MEM_BUFFERED_TYPEOF	Technologie für die Datenpufferung
18232	MM_ACTFILESYS_LOG_FILE_MEM	System: Logdateigröße
18238	MM_CC_MD_MEM_SIZE	Compilezyklen-Maschinendaten im SRAM
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

Nummer	Bezeichner: $MN_	Beschreibung
18350	MM_USER_FILE_MEM_MINIMUM	Minimaler Teileprogrammspeicher
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 (DRAM)
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 (SRAM)
18402	MM_NUM_CURVE_SEGMENTS	Anzahl der Kurvensegmente (SRAM)
18403	MM_NUM_CURVE_SEG_LIN	Anzahl der linearen Kurvensegmente (SRAM)
18404	MM_NUM_CURVE_POLYNOMS	Anzahl der Kurventabellenpolynome (SRAM)
18406	MM_NUM_CURVE_TABS_DRAM	Anzahl der Kurventabellen (DRAM)
18408	MM_NUM_CURVE_SEGMENTS_DRAM	Anzahl der Kurvensegmente (DRAM)
18409	MM_NUM_CURVE_SEG_LIN_DRAM	Anzahl der linearen Kurvensegmente (DRAM)
18410	MM_NUM_CURVE_POLYNOMS_DRAM	Anzahl der Kurventabellenpolynome (DRAM)
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 (DRAM)
18502	MM_COM_TASK_STACK_SIZE	Stackgröße in kByte für Kommunikationstask (DRAM)
18510	MM_SERVO_TASK_STACK_SIZE	Stackgröße der Servotask (DRAM)
18512	MM_IPO_TASK_STACK_SIZE	Stackgröße der Ipo-Task (DRAM)
18520	MM_DRIVE_TASK_STACK_SIZE	Stackgröße der Antriebstask (DRAM)
18540	MM_PLC_TASK_STACK_SIZE	Stackgröße der PLC-Task (DRAM)
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)
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

15.5.1.2 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

Nummer	Bezeichner: $MC_	Beschreibung
28081	MM_NUM_BASE_FRAMES	Anzahl Basisframes (SRAM)
28082	MM_SYSTEM_FRAME_MASK	Systemframes (SRAM)
28083	MM_SYSTEM_DATAFRAME_MASK	Systemframes (SRAM)
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 (DRAM)
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 (DRAM)
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[ ] wird im SRAM gespeichert.
28256	MM_NUM_AC_MARKER	Dimension von $AC_MARKER
28257	MM_BUFFERED_AC_MARKER	$AC_MARKER[ ] wird im SRAM gespeichert.
28258	MM_NUM_AC_TIMER	Anzahl Zeitvariablen $AC_TIMER (DRAM)
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 (DRAM)
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

Nummer	Bezeichner: $MC_	Beschreibung
28560	MM_SEARCH_RUN_RESTORE_MODE	Restore von Daten nach einer Simulation
28580	MM_ORIPATH_CONFIG	Einstellung für bahnrelative Orientierung ORIPATH

15.5.1.3 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

T1: Teilungsachsen

16

16.1 Kurzbeschreibung

Teilungsachsen in Werkzeugmaschinen

Bei einigen Anwendungsfällen soll eine Achse immer nur ganz bestimmte Rasterpunkte (z. B. Platznummern) anfahren. Das Anfahren der festgelegten Rasterpositionen, den Teilungen, soll dabei sowohl in der Betriebsart AUTOMATIK als auch in den Einrichte-Betriebsarten erfolgen.

Die betreffenden Achsen werden als "Teilungsachsen" bezeichnet. Die festgelegten Positionen der Teilungsachsen werden als "codierte Positionen" bzw. als "Teilungspositionen" bezeichnet.

Anwendungen

Die häufigsten Anwendungen von Teilungsachsen findet man bei Werkzeugmagazinen in Form von Werkzeug-Revolvern, Werkzeug-Kettenmagazinen oder Werkzeug-Kassettenmagazinen. Hierbei sind die codierten Positionen die einzelnen Magazinplätze der Werkzeuge. Bei einem Werkzeugwechsel wird das Magazin auf den Platz positioniert, auf dem sich das einzuwechselnde Werkzeug befindet.

Teilung anzeigen

Über Systemvariablen können folgende Daten abgefragt werden:

- die Nummer der aktuellen Teilungsposition:
  - beim Erreichen des "Genauhalt fein"-Fensters der Teilungsposition
  - beim Überschreiten des halben Weges zur nächsten Teilungsposition
- die programmierte Teilungsposition

16.2 Verfahren von Teilungsachsen

16.2.1 Allgemeines

Teilungsachsen können verfahren werden:

- von Hand in den Einrichtebetriebsarten JOG und INC
- aus einem Teileprogramm heraus mit speziellen Anweisungen für "codierte Positionen"
- von PLC

Bei Erreichen der Teilungsposition wird an die PLC folgendes Nahtstellensignal ausgegeben:

DB31, ... DBX76.6 (Teilungsachse in Position)

Hinweis

Vor dem Referenzpunktfahren können Hirth-Teilungsachsen nicht in JOG verfahren werden.

16.2.2 Verfahren von Teilungsachsen in der Betriebsart JOG

Referenzpunktfahren

Das Referenzpunktfahren einer Teilungsachse erfolgt analog wie bei anderen Achsen. Eine Übereinstimmung des Referenzpunkts mit einer Teilungsposition ist nicht erforderlich.

Erst nach Erreichen des Referenzpunkts:
DB31, ... DBX60.4 bzw. 5 (Referiert/Synchronisiert 1 bzw. 2) = 1
fährt die Teilungsachse in der Betriebsart JOG beim konventionellen und beim inkrementellen Verfahren nur noch Teilungspositionen an.

Ausnahme: Bei Verfahren über Handrad werden keine Teilungspositionen angefahren.

Ist die Achse nicht referiert (DB31, ... DBX60.4 bzw. 5 = 0), werden die Teilungspositionen beim Handfahren in JOG nicht berücksichtigt!

Kontinuierliches Verfahren bei JOG

- Tippbetrieb aktiv:

  SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD = 1

  Nach Betätigung einer Verfahrtaste "+" oder "-" wird die Teilungsachse entsprechend wie beim normalen Fahren im JOG-Betrieb bewegt. Nach Loslassen der Fahrtaste fährt die Teilungsachse auf die nächste erreichbare Teilungsposition in Fahrtrichtung.

- Dauerbetrieb aktiv:

  SD41040 $SN_JOG_CONT_MODE_LEVELTRIGGRD = 0

  Mit kurzer Betätigung der Verfahrtaste (erste steigende Flanke) startet die Teilungsachse die Verfahrbewegung in die gewünschte Richtung. Sie wird auch nach Loslassen der

Verfahrtaste fortgesetzt. Nach erneuter Betätigung der Verfahrtaste (zweite steigende Flanke) fährt die Teilungsachse auf die nächste erreichbare Teilungsposition in Fahrtrichtung.

Teilungsachsen werden in der Regel im Tippbetrieb (entspricht Grundstellung) verfahren. Der Dauerbetrieb hat hier eine untergeordnete Rolle.

Wird vor Erreichen der Teilungsposition vom Bediener ein Richtungswechsel vorgenommen, so positioniert die Teilungsachse auf die nächste in Fahrtrichtung liegende Teilungsposition. Anschließend ist die Verfahrbewegung in Gegenrichtung erneut zu starten.

Weitere Informationen bzgl. "kontinuierliches Verfahren im Tipp- bzw. Dauerbetrieb" siehe:
Literatur:
/FB2/ Funktionshandbuch Erweiterungsfunktionen; Handfahren und Handradfahren (H1)

Inkrementelles Verfahren bei JOG (INC)

Unabhängig vom aktuell eingestellten Inkrementwert (INC1, ... ,INCvar) fährt die Teilungsachse nach Betätigung einer Verfahrtaste "+" oder "-" inkrementell **immer um 1 Teilungsposition** in die angewählte Richtung.

Im Tippbetrieb wird die Verfahrbewegung nach Loslassen der Verfahrtaste unterbrochen. Die Teilungsposition kann durch erneutes Drücken der Verfahrtaste angefahren werden.

Im Dauerbetrieb dagegen wird die Verfahrbewegung durch nochmalige Betätigung der Verfahrtaste abgebrochen. Die Teilungsachse steht damit nicht auf einer Teilungsposition.

Zwischen-Teilungsposition

Steht eine Teilungsachse zwischen 2 Teilungspositionen, so wird in der Betriebsart JOG-INC mit Betätigung einer Verfahrtaste "+" die nächsthöhere Teilungsposition angefahren. Analog wird mit Betätigung der Verfahrtaste "-" die nächstniedrigere Teilungsposition angefahren.

Verfahren mit Handrad

Beim Verfahren der Teilungsachse mit dem Handrad in JOG werden die **Teilungspositionen nicht berücksichtigt**. Mit Verdrehen des Handrades wird die Teilungsachse entsprechend dem eingestellten Grundsystem in mm, inch oder Grad auf jede beliebige Position verfahren.

Falls ein Verfahren der Teilungsachse mit dem Handrad verriegelt werden soll, kann dies durch das PLC-Anwenderprogramm erfolgen.

Meldung von PLC "Teilungsachse in Position"

Bei Verfahrbewegungen der Teilungsachse in der Betriebsart JOG wird das Erreichen der Teilungsposition mit folgendem NC/PLC-Nahtstellensignal angezeigt:

DB31, ... DBX76.6 (Teilungsachse in Position)

Voraussetzung: Teilungsachse ist referiert (DB31, ... DBX60.4 bzw. 5 = 1)

Alarme bei JOG

Wird beim Verfahren der Teilungsachse in JOG der mit Hilfe der Teilungspositionstabelle definierte gültige Verfahrbereich verlassen, so wird der Alarm 20054 "Falscher Index für Teilungsachse für JOG" gemeldet.

Umdrehungsvorschub

In der Betriebsart JOG ist das Verhalten der Achse / Spindel auch abhängig von der Einstellung im Settingdatum:

SD41100 $SN_JOG_REV_IS_ACTIVE (Umdrehungsvorschub bei JOG aktiv)

SD41100	Bedeutung
= 1 (aktiv)	Die Achse / Spindel wird immer mit Umdrehungsvorschub in Abhängigkeit von der Masterspindel verfahren: MD32050 $MA_JOG_REV_VELO (Umdrehungsvorschub bei JOG) bzw. MD32040 $MA_JOG_REV_VELO_RAPID (Umdrehungsvorschub bei JOG mit Eilgangsüberlagerung)
= 0 (nicht aktiv)	Das Verhalten der Achse / Spindel ist abhängig vom Settingdatum: SD43300 $SA_ASSIG_FEED_PER_REV_SOURCE (Umdrehungsvorschub für Positionsachsen / -spindeln) Das Verhalten einer Geometrieachse, auf die ein Frame mit Rotation wirkt, ist abhängig vom Settingdatum: SD42600 $SC_JOG_FEED_PER_REV_SOURCE

16.2.3 Verfahren von Teilungsachsen in der Betriebsart AUTOMATIK

Verfahren auf beliebige Positionen

Mit einer als Teilungsachse definierten Achse kann mit Hilfe des NC-Teileprogramms in der Betriebsart AUTOMATIK **jede beliebige Position** angefahren werden. Hierbei sind auch Positionen zwischen den festgelegten Teilungspositionen anfahrbar.

Die Positionen werden wie üblich in der für die Achse gültigen Maßeinheit (mm, inch oder Grad) programmiert. Die dafür allgemein gültigen Programmieranweisungen (`G90`, `G91`, `AC`, `IC`, ...) können den Programmierhandbüchern entnommen werden.

Verfahren auf "codierte Positionen"

Zusätzlich können Teilungsachsen mit speziellen Anweisungen im Teileprogramm auf die "codierten Positionen" verfahren werden:

Anweisung	Wirkung
`CAC`	Codierte Position absolut anfahren
`CACP`	Codierte Position absolut anfahren in positiver Richtung
`CACN`	Codierte Position absolut anfahren in negativer Richtung
`CIC`	Codierte Position inkrementell anfahren
`CDC`	Codierte Position auf direktem (kürzestem) Weg anfahren

Dabei ist bei Absolutpositionierung die anzufahrende Teilungsposition bzw. bei Bezugsmaßpositionierung die Anzahl der zu verfahrenden Teilungen in "+"- oder "-"-Richtung zu programmieren.

Bei Rundachsen besteht die Möglichkeit, die Teilungsposition direkt auf kürzestem Weg (CDC) oder mit eindeutiger Drehrichtung (CACP, CACN) anzufahren.

Erreichen der Teilungsposition

Wenn das Fenster "Genauhalt fein" erreicht wird und die Teilungsachse auf einer Teilungsposition steht, wird das folgende NC/PLC-Nahtstellensignal gesetzt, unabhängig davon, wie die Teilungsposition erreicht wurde:

DB31, ... DBX76.6 (Teilungsachse in Position)

16.2.4 Verfahren von Teilungsachsen von PLC

Teilungsachsen können auch vom PLC-Anwenderprogramm verfahren werden.

Dafür gibt es folgende Möglichkeiten:

- Konkurrierende Positionierachsen

  Hier kann die anzufahrende Teilungsposition von der PLC vorgegeben werden.

  Literatur:
  /FB2/ Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)

- Asynchrone Unterprogramme (ASUP)

  Literatur:
  /FB1/ Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)

16.3 Parametrierung der Teilungsachsen

Definition der Teilungsachse

Eine Achse (Linear- oder Rundachse) kann als Teilungsachse definiert werden mit dem axialen Maschinendatum:

MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB

Wert	Bedeutung
0	Die Achse ist nicht als Teilungsachse deklariert.
1	Die Achse ist Teilungsachse. Die zugehörigen Teilungspositionen sind in der Teilungspositionstabelle 1 hinterlegt.
2	Die Achse ist Teilungsachse. Die zugehörigen Teilungspositionen sind in der Teilungspositionstabelle 2 hinterlegt.

Teilungspositionstabellen

Für jede Teilungsachse müssen die zu den jeweiligen Teilungen gehörigen Achspositionen (in mm oder Grad) in Form einer Tabelle als Maschinendaten hinterlegt sein.

2 Teilungspositionstabellen sind möglich:

MD10910 $MN_INDEX_AX_POS_TAB_1 [n] (Teilungspositionstabelle 1)

MD10930 $MN_INDEX_AX_POS_TAB_2 [n] (Teilungspositionstabelle 2)

Hinweis

Einer Teilungspositionstabelle können mehrere Achsen zugeordnet werden. Voraussetzung ist, dass diese Teilungsachsen vom gleichen Typ sind (Linearachse, Rundachse, Modulo-360°-Funktion).

Anzahl der Teilungspositionen

In jede Teilungspositionstabelle können bis zu 60 Positionen eingetragen werden:

[n = 0 ... 59]

Die tatsächlich verwendete Anzahl von Einträgen wird festgelegt mit den Maschinendaten:

MD10900 $MN_INDEX_AX_LENGTH_POS_TAB_1 (Anzahl der Positionen für Teilungspositionstabelle 1)

MD10920 $MN_INDEX_AX_LENGTH_POS_TAB_2 (Anzahl der Positionen für Teilungspositionstabelle 2)

Hinweis

Einträge der Teilungspositionstabelle, die die parametrierte Anzahl an Teilungspositionen (MD10900 bzw. MD10920) überschreiten, werden nicht berücksichtigt.

Gültiges Maßsystem

Die mit MD10900 und MD10920 definierten Teilungspositionen beziehen sich auf das für Positionstabellen projektierte Maßsystem:

MD10270 $MN_POS_TAB_SCALING_SYSTEM

Wert	Maßsystem
0	metrisch
1	inch

Hinweis

MD10270 wirkt sich auch auf folgende Settingdaten aus:

SD41500 $SN_SW_CAM_MINUS_POS_TAB_1 (Schaltpunkte bei fallender Nockenflanke 1-8)

bis

SD41507 $SN_SW_CAM_PLUS_POS_TAB_4 (Schaltpunkte bei steigender Nockenflanke 25-32)

Angabe der Teilungspositionen

Es gelten folgende Regeln:

- Die Teilungspositionen sind in aufsteigender Reihenfolge (beginnend vom negativen zum positiven Verfahrbereich) ohne Lücken in die Tabelle einzutragen.
- Aufeinanderfolgende Positionswerte dürfen nicht identisch sein.
- Die Achspositionen sind im Basiskoordinatensystem einzutragen.

Modulo-Rundachse als Teilungsachse

Die Teilungsachse ist als Rundachse mit Modulo 360° definiert:

MD30300 $MA_IS_ROT_AX = 1

und

MD30310 $MA_ROT_IS_MODULO = 1

In diesem Fall sind für die Angabe der Teilungspositionen folgende Punkte zusätzlich zu beachten:

- Zulässiger Bereich: 0° ≤ Pos < 360°
- Da die Teilungsachse als endlos drehende Rundachse definiert ist, wird beispielsweise nach Erreichen der höchstgültigen Teilungsposition der Tabelle und anschließendem Weiterfahren mit INC in positive Drehrichtung wieder die Teilungsposition 1 angefahren. Analog wird bei negativer Drehrichtung mit INC von Teilungsposition 1 auf die höchstgültige Teilungsposition verfahren.

16.4 Programmierung von Teilungsachsen

Codierte Position

Für das Positionieren der Teilungsachsen vom NC-Teileprogramm gibt es spezielle Anweisungen, bei denen anstelle von Achspositionen in mm oder Grad die Teilungsnummern (z. B. Platznummern) programmiert werden. Die Verfügbarkeit einer speziellen Anweisung ist abhängig vom Achstyp (Linear- oder Rundachse):

Anweisung	Wirkung	Verfügbarkeit
CAC(i)	**C**odierte Position **a**bsolut anfahren	Linearachse, Rundachse
CACP(i)	**C**odierte Position **a**bsolut anfahren in **p**ositiver Richtung	Rundachse
CACN(i)	**C**odierte Position **a**bsolut anfahren in **n**egativer Richtung	Rundachse
CDC(i)	**C**odierte Position auf **d**irektem (kürzestem) Weg anfahren	Rundachse
CIC(i)	**C**odierte Position **i**nkrementell anfahren	Linearachse, Rundachse

i: Codierte Position (Teilungsposition)

Wertebereich von i: 0 ... 59; ganzzahlig (bei CIC sind positive und negative Werte möglich)

Beispiele

Programmierung	Kommentar
POS[B]=CAC(20)	; Die Teilungsachse B fährt die codierte Position (Teilung) 20 **absolut** an. Die Verfahrrichtung ist abhängig von der aktuellen Istposition.

Programmierung	Kommentar
POS[B]=CACP(10)	; Die Teilungsachse B fährt die codierte Position (Teilungsposition) 10 **absolut mit positiver Drehrichtung** an (nur bei Rundachsen möglich).

Programmierung	Kommentar
POS[B]=CACN(10)	; Die Teilungsachse B fährt die codierte Position (Teilungsposition) 10 **absolut mit negativer Drehrichtung** an (nur bei Rundachsen möglich).

Programmierung	Kommentar
POS[B]=CDC(50)	; Die Teilungsachse B fährt die Teilungsposition 50 **direkt auf kürzestem Weg** an (nur bei Rundachsen möglich).

Programmierung	Kommentar
POS[B]=CIC(-4)	; Die Teilungsachse B fährt von der momentanen Teilungsposition **inkrementell** um 4 Teilungspositionen **in negativer Richtung.**

Programmierung	Kommentar
POS[B]=CIC(35)	; Die Teilungsachse B fährt von der momentanen Teilungsposition **inkrementell** um 35 Teilungspositionen **in positiver Richtung.**

Besonderheiten

- Modulo-Rundachse als Teilungsachse

  Bei Modulo-Rundachsen werden die Teilungspositionen direkt angefahren und Vielfache von 360° abgeschnitten.

- Teilungsachse steht zwischen zwei Teilungspositionen

  Die angegebenen Positionsanweisungen haben in der Betriebsart AUTOMATIK folgende Wirkung:

POS[B]=CIC(1)	Die **nächsthöhere** Teilungsposition wird angefahren.
POS[B]=CIC(-1)	Die **nächstniedrigere** Teilungsposition wird angefahren.
POS[B]=CIC(0)	Die Teilungsachse wird **nicht** verfahren.

Anzeige der Teilungsposition

Die Nummer der zuletzt programmierten Teilungsposition kann gelesen werden mit der Systemvariablen:

$AA_PROG_INDEX_AX_POS_NO

Die Nummer der zuletzt überfahrenen Teilungsposition kann angezeigt werden mit der Systemvariablen:

$AA_ACT_INDEX_AX_POS_NO

Die Anzeige ist abhängig von der Einstellung im Maschinendatum:

MD10940 $MN_INDEX_AX_MODE (Einstellungen für Teilungsposition)

Bit	Wert	Bedeutung
0	0	Die Teilungsposition ändert sich beim Erreichen der Teilungsposition ("Genauhalt fein"-Fenster) und bleibt bis zum Erreichen der nächsten Teilungsposition unverändert. Der Teilungsbereich beginnt also an einer Teilungsposition und endet vor der nächsten Teilungsposition.
	1	Die Teilungsposition ändert sich beim Erreichen der halben Teilungsposition. Damit wird ein quasi symmetrischer Teilungsbereich um die Teilungsposition gelegt (symmetrisch nur bei Linearachsen mit äquidistanter Aufteilung bzw. Modulo-Rundachsen, bei denen der Teilungsbereich ein ganzzahliges Vielfaches des Modulo-Bereiches ist (MD30330 $MA_MODULO_RANGE), ansonsten proportional zu den Abständen der Teilungspositionen). Bei **Modulo-Rundachsen** wird der Bereich zwischen letzter Teilungsposition und erster Teilungsposition entsprechend der Längen von erstem Teilungsbereich und letztem Teilungsbereich **proportional aufgeteilt**.

Die folgenden Grafiken verdeutlichen den Unterschied zwischen Bit 0 = 0 und Bit 0 = 1:

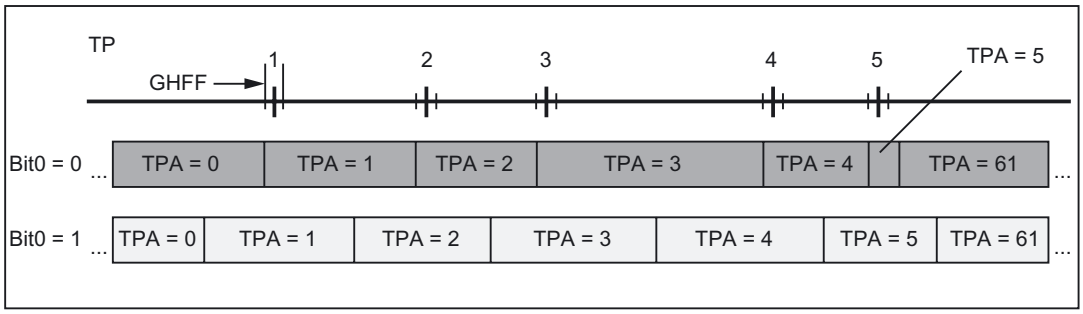
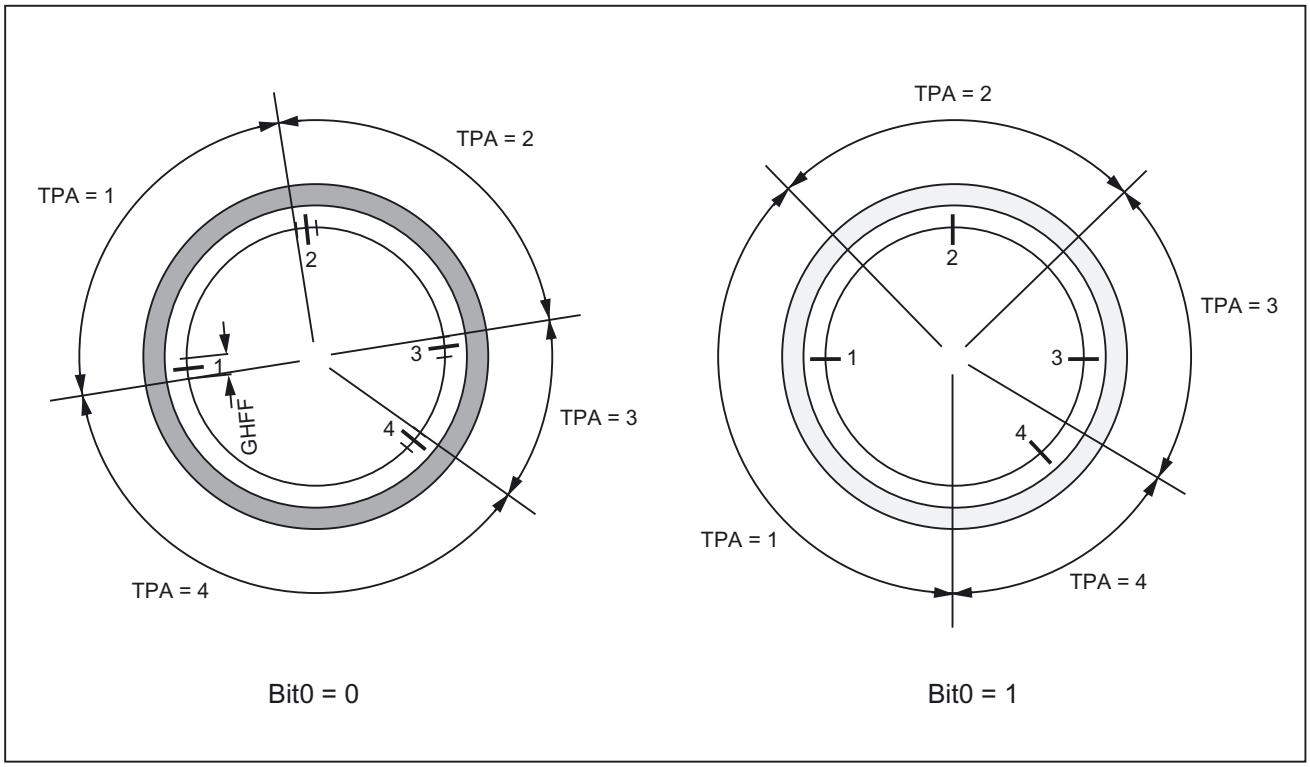


Bild 16-1 Teilungspositionsanzeigen: Linearachse



TP Programmierte Teilungsposition
TPA Angezeigte Teilungsposition
GHFF "Genauhalt fein"-Fenster

Bild 16-2 Teilungspositionsanzeigen: Modulo-Rundachse

Wertebereich von $AA_ACT_INDEX_AX_POS_NO

Zu erwartende Wertebereiche der Systemvariablen $AA_ACT_INDEX_AX_POS_NO:

Teilungspositionen aus Tabelle			
Modulo-Rundachse	1 ... n	keine 0 n = maximal 60	
Linearachse	0*, 1, 2, 3, ... 59, 60, 61*	0*:	**unterhalb** des gesamten Teilungsbereichs
		61*:	**oberhalb** des gesamten Teilungsbereichs

Äquidistante Teilungspositionen		
Modulo-Rundachse	1 ... m	keine 0 m = Denominator (Zähler)
Linearachse	... -3, -2, -1, 0, 1, 2, 3, ...	

Fahren auf nächste Teilungsposition

Das Verhalten beim Kommando"Fahre auf nächste Teilungsposition" ist abhänig von der Einstellung im Maschinendatum:

MD10940 $MN_INDEX_AX_MODE (Einstellungen für Teilungsposition)

Bit	Wert	Bedeutung
0	0	Nächste Teilungsposition wird angefahren.
	1	Es wird immer die am nächsten liegende Teilungsposition in Fahrtrichtung angefahren.

Zur Erläuterung dient folgendes Beispiel:

Bit 0 = 1 und die Achse steht unterhalb der Teilungsposition (aber außerhalb des "Genauhalt fein" -Fensters).

Obwohl die Systemvariable $AA_ACT_INDEX_AX_POS_NO Teilungsposition 2 anzeigt, wird mit dem Kommando "Fahre auf nächste Position" **nicht** auf Teilungsposition 3 gefahren, sondern erst exakt auf Teilungsposition 2. Erst wenn die Achse exakt auf (Genauhalt fein) oder oberhalb der Teilungsposition steht, wird mit dem Kommando "Fahre auf nächste Position" die nächste Teilungsposition (hier im Beispiel Teilungsposition 3) angefahren.

Es wird immer zuerst die Teilungsposition angefahren, die in Fahrtrichtung am nächsten liegt! Unter Umständen muss also das Kommando "Fahre auf nächste Position" zweimal gegeben werden, damit von der aktuell angezeigten Teilungsposition zur nächsten Teilungspositionsnummer verfahren wird (z. B. von 2 auf 3).

FRAMES

Da die Steuerung die in der Teilungspositionstabelle abgelegten Positionen wie programmierte Positionen in mm, inch oder Grad interpretiert, sind FRAMES bei Teilungsachsen nicht verriegelt.

Entsprechend dem Anwendungsbereich sind FRAMES bei Teilungsachsen in der Regel nicht notwendig. Daher sind FRAMES und Nullpunktverschiebungen bei Teilungsachsen im Teileprogramm zu unterdrücken.

16.5 Äquidistante Teilungsintervalle

16.5.1 Funktion

Allgemeines

Es gibt:

- beliebig viele, gleich große (äquidistante) Teilungsintervalle
- geänderte Wirksamkeit von MD für Teilungsachsen

Äquidistante Teilungsintervalle sind möglich für:

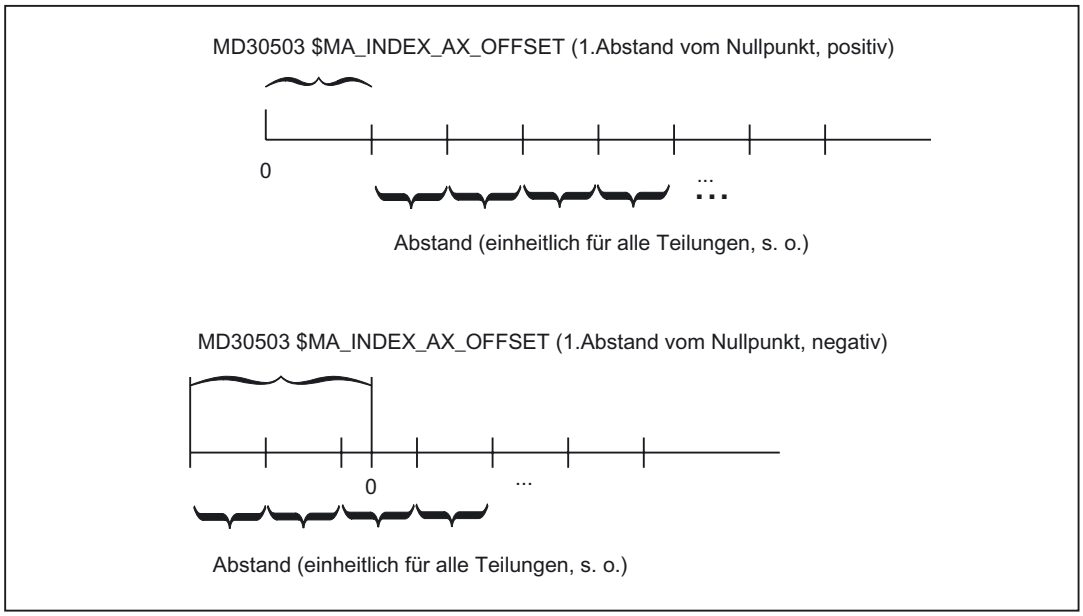
- Linearachsen
- Modulo-Rundachsen
- Rundachsen

Teilungsabstand

Für äquidistante Teilungsintervalle bei Linearachsen wird der Teilungsabstand nach folgender Formel bestimmt:

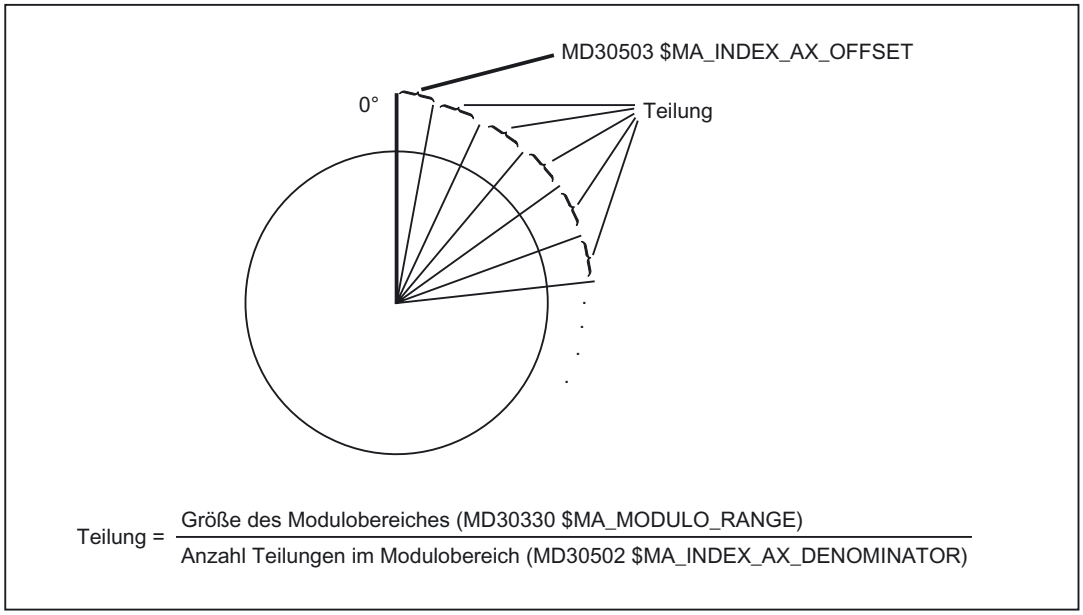
$$\text{Abstand} = \frac{\text{Zähler (MD30501 \$MA_INDEX_AX_NUMERATOR)}}{\text{Nenner (MD30502 \$MA_INDEX_AX_DENOMINATOR)}}$$

Lineare Achse



Modulo-Rundachse

$$\text{Teilung} = \frac{\text{Zähler (MD30330 \$MA_MODULO_RANGE)}}{\text{Nenner (MD30502 \$MA_INDEX_AX_DENOMINATOR)}}$$



Aktivierung

Die Funktionen mit äquidistanter Teilung für eine Achse (Linearachse, Modulo-Rundachse oder Rundachse) werden durch folgende Einstellung aktiviert:

MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB[Achse] = 3

16.5.2 Hirth-Verzahnung

Funktion

Mit Hirth-Verzahnungen werden i. d. R. Drehpositionen einer Rundachse verriegelt, indem jeweils an der Teilungsposition ein Riegel oder ein weiteres Zahnrad über eine lineare Achse in Eingriff gebracht wird. Die Verriegelung darf nur erfolgen, wenn eine Teilungsposition genau erreicht ist. Die Teilungspositionen haben über den ganzen Umfang gleichen Abstand (äquidistant).

Voraussetzungen

Die Rundachse muss Teilungsachse sein. Die Achse muss referiert sein.

Literatur:
/FB1/ Funktionshandbuch Grundfunktionen; Referenzpunktfahren (R1)

Aktivierung

Das folgende Maschinendatummuss auf 1 gesetzt sein:
MD30505 $MA_HIRTH_IS_ACTIVE

Das Maschinendatum muss auf 3 gesetzt sein (äquidistante Teilungen):
MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB

Wirkung

- Die Rundachse kann in allen Betriebsarten und Betriebszuständen nur Teilungspositionen anfahren.
- In Betriebsart JOG kann die Achse konventionell oder inkrementell verfahren werden.

  Voraussetzung: Achse ist referiert.
- Handradfahren ist nicht möglich.

  Literatur:
  /FB2/ Funktionshandbuch Erweiterungsfunktionen; Handfahren und Handradfahren (H1)
- In AUTO, MDA oder über ASUP können nur "Codierte Positionen" angefahren werden.
- Die PLC kann die Achse nur auf Teilungspositionen verfahren.

16.5.3 Verhalten der Hirth-Achsen in besonderen Situationen

STOP/RESET

Bei `NC-STOP` und `RESET` während einer Fahrbewegung wird noch die nächste Teilungsposition angefahren.

NOT AUS

Nach `NOT AUS` muss die PLC oder der Bediener mit JOG die Teilungsachse wieder auf eine Teilungsposition ausrichten, bevor die Längsachse wieder eingerückt/abgesenkt werden kann.

Override = 0 oder Signal "Achse anhalten"

Ist die vorhergehende Teilungsposition bereits verlassen, wenn diese Ereignisse eintreten, dann wird die nächstmögliche Teilungsposition noch durch die Steuerung angefahren.

Restweglöschen

Nach dem Fahren zur nächstmöglichen Teilungsposition wird auf dieser die Fahrbewegung abgebrochen.

Kommandoachsen

Wird für eine bewegte Kommandoachse ein MOV = 0 gegeben, so fährt die Achse noch zur nächstmöglichen Teilungsposition.

Literatur:
/FBSY/ Funktionsbeschreibung Synchronaktionen

Befehl MOV

`MOV = 1`	Geht bei Teilungsachsen mit und ohne Hirthverzahnung.
`MOV = 0`	Funktioniert bei beiden gleich, es wird die nächste Position angefahren.

Befehl DELDTG

Bei Teilungsachsen ohne Hirthverzahnung:	Achse steht sofort.
Bei Teilungsachsen mit Hirthverzahnung:	Achse fährt nächste Position an.

16.5.4 Einschränkungen

Transformationen

Die Achse, für die Hirth-Verzahnung definiert ist, kann nicht an kinematischen Transformationen beteiligt werden.

PRESET

Die Achse, für die Hirth-Verzahnung definiert ist, kann nicht mit `PRESET` auf einen neuen Wert eingestellt werden.

Umdrehungsvorschub

Die Achse, für die Hirth-Verzahnung definiert ist, darf nicht mit Umdrehungsvorschub verfahren werden.

Weg-/Geschwindigkeitsüberlagerung

Die Achse, für die Hirth-Verzahnung definiert ist, kann nicht mit Weg- oder Geschwindigkeitsüberlagerung benutzt werden.

Frames, ext. NPV, DRF

Die Achse, für die Hirth-Verzahnung definiert ist, erlaubt keine Frames, interpolatorische Korrekturen wie externe Nullpunktverschiebung, DRF usw.

Kopplungen

Eine Hirth-Verzahnungs-Achse kann niemals gleichzeitig einer der folgenden Achsentypen sein:

- Leitwertgekoppelte Folgeachse
- Mitgeschleppte Achse
- Gantry-Folgeachse

Literatur:
/FB3/ Funktionshandbuch Sonderfunktionen; Achskopplungen und ESR (M3)

16.5.5 Geänderte Wirksamkeit von Maschinendaten

RESET

Für die folgenden Maschinendaten ist nach Belegung mit neuen Werten `RESET` erforderlich, um sie wirksam zu setzen.

MD10900 $MN_INDEX_AX_LENGTH_POS_TAB_1

MD10920 $MN_INDEX_AX_LENGTH_POS_TAB_2

MD10910 $MN_INDEX_AX_POS_TAB_1

MD10930 $MN_INDEX_AX_POS_TAB_2

MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB

16.6 Inbetriebnahme von Teilungsachsen

Vorgehen

Die Inbetriebnahme von Teilungsachsen erfolgt entsprechend wie bei normalen NC-Achsen (Linear- bzw. Rundachsen).

Rundachse

Falls die Teilungsachse als Rundachse (MD30300 $MA_IS_ROT_AX = "1") mit Modulo 360°-Wandlung (MD30310 $MA_ROT_IS_MODULO = "1") definiert ist, werden auch die Teilungspositionen mit Modulo 360° verfahren. In der Teilungspositionstabelle dürfen dann nur Positionen im Bereich 0° bis 359,999° eingetragen werden. Ansonsten wird beim Hochlauf der Alarm 4080 "fehlerhafte Konfiguration für Teilungsachse in MD [Name]" gemeldet.

Die Positionsanzeige kann mit dem Maschinendatum:
MD30320 $MA_DISPLAY_IS_MODULO = 1
auf Modulo 360° eingestellt werden.

Spezielle Maschinendaten

Zusätzlich sind die folgenden Maschinendaten einzustellen:

Allgemeine Maschinendaten	
MD10900 $MN_INDEX_AX_LENGTH_POS_TAB_1	Anzahl der Positionen für Teilungsachstabelle 1
MD10920 $MN_INDEX_AX_LENGTH_POS_TAB_2	Anzahl der Positionen für Teilungsachstabelle 2
MD10910 $MN_INDEX_AX_POS_TAB_1 [n]	Teilungspositionstabelle 1
MD10930 $MN_INDEX_AX_POS_TAB_2 [n]	Teilungspositionstabelle 2

Axiale Maschinendaten	
MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB	Achse ist Teilungsachse (Zuordnung der Teilungspositionstabelle 1 oder 2, oder 3 für äquidistante Teilung)
MD30505 $MA_HIRTH_IS_ACTIVE	Achse hat Eigenschaft "Hirthverzahnung"
MD30501 INDEX_AX_NUMERATOR	Zähler für äquidistante Teilung
MD30502 INDEX_AX_DENOMINATOR	Nenner für äquidistante Teilung
MD30503 INDEX_AX_OFFSET	Abstand der 1. Teilungsposition von Null

Maschinendatenbeispiele

Nachfolgend wird anhand von zwei Beispielen die Belegung der o. g. Maschinendaten erläutert.

Beispiel für Teilungsachse als Rundachse

Werkzeug-Revolver mit 8 Revolverplätzen. Der Werkzeug-Revolver wird als endlos drehende Rundachse definiert. Die Abstände zwischen den 8 Revolverplätzen sind konstant. Der erste Revolverplatz liegt bei Position 0°:

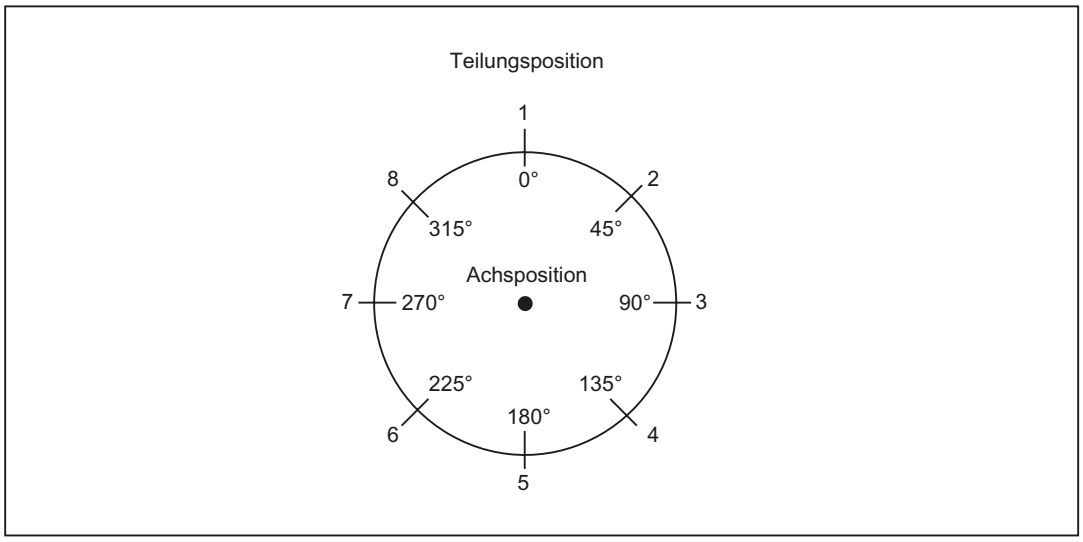


Bild 16-3 Beispiel: Werkzeug-Revolver mit 8 Plätzen

Die Teilungspositionen für den Werkzeug-Revolver werden in Teilungspositionstabelle 1 eingetragen:

```
MD10910 $MN_INDEX_AX_POS_TAB_1[0] = 0      ;  1. Teilungsposition bei 0°
MD10910 $MN_INDEX_AX_POS_TAB_1[1] = 45     ;  2. Teilungsposition bei 45°
MD10910 $MN_INDEX_AX_POS_TAB_1[2] = 90     ;  3. Teilungsposition bei 90°
MD10910 $MN_INDEX_AX_POS_TAB_1[3] = 135    ;  4. Teilungsposition bei 135°
MD10910 $MN_INDEX_AX_POS_TAB_1[4] = 180    ;  5. Teilungsposition bei 180°
MD10910 $MN_INDEX_AX_POS_TAB_1[5] = 225    ;  6. Teilungsposition bei 225°
MD10910 $MN_INDEX_AX_POS_TAB_1[6] = 270    ;  7. Teilungsposition bei 270°
MD10910 $MN_INDEX_AX_POS_TAB_1[7] = 315    ;  8. Teilungsposition bei 315°
```

Weitere Maschinendaten:

```
MD10900 $MN_INDEX_AX_LENGTH_POS_TAB_1= 8   ; 8 Teilungspositionen in Tabelle
                                             1
MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB [AX5]  ; Achse 5 ist als Teilungsachse
= 1                                        ; def.
                                             Teilungspositionen in Tabelle 1
MD30300 $MA_IS_ROT_AX [AX5] = 1            ; Achse 5 ist Rundachse
MD30310 $MA_ ROT_IS_MODULO [AX5] = 1       ; Modulowandlung ist aktiviert
```

Beispiel für Teilungsachse als Linearachse

Werkstückpalette mit 10 Plätzen.

Die Abstände zwischen den 10 Plätzen sind unterschiedlich. Der erste Palettenplatz liegt bei Position -100 mm.

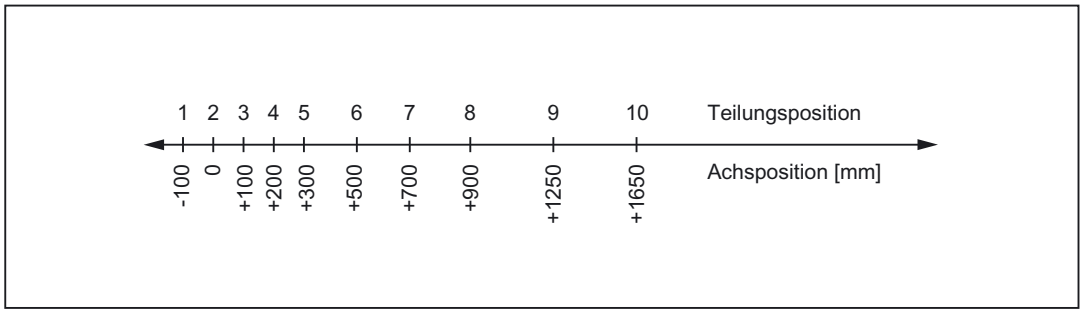


Bild 16-4 Beispiel: Werkstückpalette als Teilungsachse

Die Teilungspositionen für die Werkstückpalette werden in Teilungspositionstabelle 2 eingetragen:

```
MD10930 $MN_INDEX_AX_POS_TAB_2[0] = -100    ;  1. Teilungsposition bei -100
MD10930 $MN_INDEX_AX_POS_TAB_2[1] = 0       ;  2. Teilungsposition bei 0
MD10930 $MN_INDEX_AX_POS_TAB_2[2] = 100     ;  3. Teilungsposition bei 100
MD10930 $MN_INDEX_AX_POS_TAB_2[3] = 200     ;  4. Teilungsposition bei 200
MD10930 $MN_INDEX_AX_POS_TAB_2[4] = 300     ;  5. Teilungsposition bei 300
MD10930 $MN_INDEX_AX_POS_TAB_2[5] = 500     ;  6. Teilungsposition bei 500
MD10930 $MN_INDEX_AX_POS_TAB_2[6] = 700     ;  7. Teilungsposition bei 700
MD10930 $MN_INDEX_AX_POS_TAB_2[7] = 900     ;  8. Teilungsposition bei 900
MD10930 $MN_INDEX_AX_POS_TAB_2[8] = 1250    ;  9. Teilungsposition bei 1250
MD10930 $MN_INDEX_AX_POS_TAB_2[9] = 1650    ;  10. Teilungsposition bei 1650
```

Weitere Maschinendaten

```
MD10920
$MN_INDEX_AX_LENGTH_POS_TAB_2=10            ; 10 Teilungspositionen in Tabelle
                                              2
MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB [AX6]   ; Achse 6 ist als Teilungsachse
= 2                                         ; def.
                                              Teilungspositionen in Tabelle2
```

16.7 Besonderheiten von Teilungsachsen

DRF

Auch bei Teilungsachsen kann mit Hilfe der Funktion DRF im Automatik-Betrieb mit dem Handrad eine zusätzliche inkrementelle Nullpunktverschiebung generiert werden.

Software-Endschalter

Nachdem die Teilungsachse referiert ist, sind bei Verfahrbewegungen auch die Software-Endschalter wirksam.

Beim Handfahren mit JOG-kontinuierlich oder JOG-inkrementell bleibt die Teilungsachse an der Teilungsposition stehen, die vor dem Software-Endschalter liegt.

Referenzpunktfahren

Erst **nach Erreichen des Referenzpunktes**:
DB31, ... DBX60.4 bzw. 60.5 (Referiert/Synchronisiert 1 bzw. 2) = 1
fährt die Teilungsachse in der Betriebsart JOG kontinuierlich oder inkrementell nur noch Teilungspositionen an.

Ist die Achse nicht referiert:
DB31, ... DBX60.4 bzw. 60.5 (Referiert/Synchronisiert 1 bzw. 2) = 0,
so werden die Teilungspositionen beim Handfahren nicht berücksichtigt!

Da die in den Teilungspositionstabellen hinterlegten Achspositionen nur bei einer referierten Achse mit den zugehörigen Maschinenpositionen übereinstimmen, muss ein NC-Start verriegelt werden, solange die Teilungsachse nicht referiert ist.

Positionsanzeige

Die Positionsanzeigen von Teilungsachsen erfolgen in den achstypischen Maßeinheiten (mm, inch oder Grad).

Abbruch durch RESET

Mit RESET wird die Verfahrbewegung einer Teilungsachse abgebrochen und die Achse stillgesetzt. Die Teilungsachse positioniert dabei nicht mehr auf eine Teilungsposition.

16.8 Beispiele

16.8.1 Beispiele mit äquidistanten Teilungen

Modulo-Rundachse

MD30502 $MA_INDEX_AX_DENOMINATOR[AX4] =18

MD30503 $MA_INDEX_AX_OFFSET[AX4]=5

MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3

MD30300 $MA_IS_ROT_AX[AX4] = TRUE

MD30310 $MA_ROT_IS_MODULO[AX4] = TRUE

Mit den obigen Maschinendaten wird die Achse 4 als eine Modulo-Rundachse und Teilungsachse mit äquidistanten Positionen alle 20° beginnend auf 5° definiert.
Damit ergeben sich die folgenden Teilungspositionen:
5, 25, 45, 65, 85, 105, 125, 145, 165, 185, 205, 225, 245, 265, 285, 305, 325 und 245 Grad.

Hinweis

Die Angabe:
MD30502 $MA_INDEX_AX_DENOMINATOR[AX4] =18
führt zu einer 20°-Teilung, da das Maschinendatum:
MD30330 $MA_MODULO_RANGE
360° als Standardwert enthält.

Rundachse

MD30501 $MA_INDEX_AX_NUMERATOR[AX4] = 360

MD30502 $MA_INDEX_AX_DENOMINATOR[AX4] =18

MD30503 $MA_INDEX_AX_OFFSET[AX4]=100

MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3

MD30300 $MA_IS_ROT_AX[AX4] = TRUE

MD36100 $MA_POS_LIMIT_MINUS[AX1]=100

MD36110 $MA_POS_LIMIT_PLUS[AX1]=260

Mit den obigen Maschinendaten wird die Achse 4 als eine Rundachse und Teilungsachse mit äquidistanten Positionen alle 20° beginnend auf 100° definiert.
Damit ergeben sich die folgenden Teilungspositionen:
100°, 120°, 140° usw.
Positionen kleiner als 100° können nicht als Teilungspositionen angefahren werden.
Es empfiehlt sich hier den unteren Software-Endschalter zu platzieren. Die Teilungspositionen setzen sich bis zum Erreichen des Software-Endschalters (hier 260°) fort.
Damit kann die Rundachse nur zwischen 100° und 260° verfahren werden.

Linearachse

MD30501 $MA_INDEX_AX_NUMERATOR[AX1] = 10

MD30502 $MA_INDEX_AX_DENOMINATOR[AX1] =1

MD30503 $MA_INDEX_AX_OFFSET[AX1]=-200

MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB[AX1] = 3

MD30300 $MA_IS_ROT_AX[AX1] = FALSE

MD36100 $MA_POS_LIMIT_MINUS[AX1]=-200

MD36110 $MA_POS_LIMIT_PLUS[AX1]=200

Mit den obigen Maschinendaten wird die Achse 4 als eine Linearachse und Teilungsachse mit äquidistanten Positionen alle 10 mm beginnend auf -200 mm definiert.
Damit ergeben sich die folgenden Teilungspositionen:
-200, -190, -180 mm usw.
Diese Teilungspositionen setzen sich bis zum Erreichen des Software-Endschalters (hier 200 mm) fort.

Hirth-Verzahnung

MD30502 $MA_INDEX_AX_DENOMINATOR[AX4] =360

MD30503 $MA_INDEX_AX_OFFSET[AX4]=0

MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3

MD30300 $MA_IS_ROT_AX[AX4] = TRUE

MD30310 $MA_ROT_IS_MODULO[AX5] = TRUE

MD30505 $MA_HIRTH_IS_ACTIVE[AX4] = TRUE

Mit den obigen Maschinendaten wird die Achse 4 als Modulo-Rundachse und Teilungsachse mit Hirth-Verzahnung und äquidistanten Positionen alle 1° beginnend auf 0° definiert.

16.9 Datenlisten

16.9.1 Maschinendaten

16.9.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
10260	CONVERT_SCALING_SYSTEM	Grundsystem Umschaltung aktiv
10270	POS_TAB_SCALING_SYSTEM	Maßsystem der Positionstabellen
10900	INDEX_AX_LENGTH_POS_TAB_1	Anzahl der Positionen für Teilungsachstabelle 1
10910	INDEX_AX_POS_TAB_1[n]	Teilungspositionstabelle 1
10920	INDEX_AX_LENGTH_POS_TAB_2	Anzahl der Positionen für Teilungsachstabelle 2
10930	INDEX_AX_POS_TAB_2[n]	Teilungspositionstabelle 2
10940	INDEX_AX_MODE	Optionen für Teilungspositionen

16.9.1.2 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: $MA_	Beschreibung
30300	IS_ROT_AX	Rundachse
30310	ROT_IS_MODULO	Modulowandlung für Rundachse
30320	DISPLAY_IS_MODULO	Positionsanzeige ist Modulo 360°
30500	INDEX_AX_ASSIGN_POS_TAB	Achse ist Teilungsachse
30501	INDEX_AX_NUMERATOR	Zähler für Teilungsachsen mit äquidistanten Positionen
30502	INDEX_AX_DENOMINATOR	Nenner für Teilungsachsen mit äquidistanten Positionen
30503	INDEX_AX_OFFSET	Erste Teilungsposition für Teilungsachsen mit äquidistanten Positionen
30505	HIRTH_IS_ACTIVE	Hirth-Verzahnung ist aktiv

16.9.2 Settingdaten

16.9.2.1 Allgemeine Settingdaten

Nummer	Bezeichner: $SN_	Beschreibung
41050	JOG_CONT_MODE_LEVELTRIGGRD	JOG-kontinuierlich im Tippbetrieb

16.9.3 Signale

16.9.3.1 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Referiert/Synchronisiert 1, Referiert/Synchronisiert 2	DB31, ...DBX60.4/5	DB390x.DBX0.4/5
Teilungsachse in Position	DB31, ...DBX76.6	DB390x.DBX1002.6

16.9.4 Systemvariablen

16.9.4.1 Systemvariablen

Bezeichner	Beschreibung
$AA_ACT_INDEX_AX_POS_NO[Achse]	Nr. der zuletzt erreichten bzw. überfahrenen Teilungsposition
$AA_PROG_INDEX_AX_POS_NO[Achse]	Nr. der programmierten Teilungsposition

W3: Werkzeugwechsel 17

17.1 Kurzbeschreibung

Werkzeugwechsel

Für die Komplettbearbeitung von Werkstücken sind CNC-gesteuerte Werkzeugmaschinen mit Werkzeugmagazinen und automatischem Werkzeugwechsel ausgestattet.

Ablauf

Der Ablauf des Werkzeugwechsels von Span zu Span läuft in drei Schritten ab:

1. Bewegung des Werkzeugträgers von der Bearbeitungsstelle zur Werkzeugwechselposition
2. Werkzeugwechsel
3. Bewegung des Werkzeugträgers von der Werkzeugwechselposition an die neue Bearbeitungsstelle.

Forderungen

An den Werkzeugwechsel werden folgende Forderungen gestellt:

- kurze Nebenzeiten
- zeitsparendes Suchen, Bereitstellen und Zurückgeben des Werkzeugs während der Hauptzeit
- einfaches Programmieren des Werkzeugwechselzyklus
- automatischer Ablauf der notwendigen Achsen- und Greiferbewegungen
- problemloses Beheben von Störungen

17.2 Werkzeugmagazine und Wechseleinrichtungen

Werkzeugmagazine und Wechseleinrichtungen sind an den Maschinentyp angepasst:

Maschinentyp	Werkzeugmagazin	Wechseleinrichtungen
Drehmaschinen	Revolver (Scheiben-, Flach-, Schräg-)	Keine gesonderte Wechseleinrichtung. Der Wechsel erfolgt durch Drehen des Revolvers.
Fräsmaschinen	Magazine (Ketten-, Teller-, Scheiben-, Kassetten-)	Greifer / Doppelgreifer als Wechseleinrichtung.

Da der Wechselvorgang die Bearbeitung unterbricht, entstehen hier Nebenzeiten, die zu minimieren sind.

17.3 Werkzeugwechselzeiten

Werkzeugwechselzeiten hängen sehr stark von der konstruktiven Auslegung der Werkzeugmaschine ab.

Typische Werkzeugwechselzeiten	
0,1 bis 0,2 s	für Weiterschalten eines Revolvers
0,3 bis 2 s	für Werkzeugwechsel mit Greifer für ein bereitgestelltes Werkzeug

17.4 Span-zu-Span-Zeit

Span-zu-Span-Zeit ist die Zeit, die vergeht, um von der Unterbrechungsstelle an der Kontur (vom Span) bei drehender Spindel zum Zweck des Werkzeugwechsels wegzufahren bis zum Zurückkehren an die Unterbrechungsstelle (zum Span) bei drehender Spindel mit dem neuen Werkzeug.

Damit ergeben sich folgende typische Span-zu-Span-Zeiten:

Typische Span-zu-Span-Zeiten	
0,3 bis 1 s	für Drehmaschine mit Revolver
0,5 bis 5 s	für Fräsmaschine mit Werkzeugwechsler

17.5 Ansteuerung des Werkzeugwechsels

Varianten

Die Ansteuerung des Werkzeugwechsels kann erfolgen über:

- T-Funktion
- M-Befehl (vorzugsweise `M06`)

Parametrierung

Welche der Ansteuerungsvarianten wirksam sein soll, wird festgelegt mit dem Maschinendatum:

MD22550 $MC_TOOL_CHANGE_MODE

Wert	Bedeutung	Typische Anwendung
0	Neues Werkzeug wird mit T-Funktion sofort eingewechselt.	Drehmaschine mit Werkzeugrevolver
1	Neues Werkzeug wird mit T-Funktion zum Wechsel vorbereitet und parallel zur Hauptzeit in die Werkzeugwechselposition gebracht. Mit dem M-Befehl zum Werkzeugwechsel wird dann das bisherige Werkzeug aus der Spindel entfernt und das neue eingewechselt.	Fräsmaschine mit Werkzeugmagazin

Der M-Befehl zum Werkzeugwechsel wird festgelegt im Maschinendatum:

MD22560 $MC_TOOL_CHANGE_M_CODE

Grundeinstellung ist 6 (entsprechend DIN 66025).

Hinweis

Wenn die Werkzeugkorrekturnummer von der PLC oder einer HMI-Werkzeugverwaltung kommt, dann muss an der geeigneten Stelle ein Vorlaufstopp `STOPRE` eingefügt werden. `STOPRE` ist allerdings bei eingeschalteter Werkzeugradiuskorrektur (`G41` / `G42`) bzw. SPLINE-Interpolation zu vermeiden, da hier für die Bahnberechnung mehrere Sätze im Voraus benötigt werden.

Literatur

Weitere Informationen über M-Funktionen, die auch für Werkzeugwechsel `M06` gelten (z. B. Erweiterte Adresse, Ausgabezeitpunkt an die PLC, Hilfsfunktionsgruppen, Verhalten bei Satzsuchlauf, Verhalten bei Überspeichern), finden Sie in:

/FBSY/ Funktionshandbuch Synchronaktionen

17.6 Werkzeugwechselpunkt

Werkzeugwechselpunkt

Die Wahl des Werkzeugwechselpunkts hat großen Einfluss auf die Span-zu-Span-Zeit (Seite 968). Seine Festlegung richtet sich nach dem Werkzeugmaschinenkonzept und evtl. der aktuellen Bearbeitungsaufgabe.

Festpunktfahren

Mit der Funktion "Festpunktfahren" können feste Positionen einer Maschinenachse angefahren werden, die in Maschinendaten abgelegt sind. Dies lässt sich für die Definition und Ansteuerung eines oder mehrerer Werkzeugwechselpunkte verwenden.

Festpunktfahren ist auf zwei Arten möglich:

- Festpunktfahren in JOG

  Der Maschinenbediener startet das Festpunktfahren in der Betriebsart JOG mit den Verfahrtasten oder dem Handrad.

  Literatur:
  Funktionshandbuch Erweiterungsfunktionen; Handfahren und Handradfahren (H1), Kapitel: "Festpunktfahren in JOG"

- Festpunktfahren mit `G75`

  Das Festpunktfahren wird über den Befehl `G75` aus dem Teileprogramm heraus aufgerufen.

  Literatur:
  Programmierhandbuch Grundlagen; Kapitel: "Wegbefehle" > "Festpunkt anfahren (G75)"

17.7 Randbedingungen

Der Werkzeugwechsel setzt voraus, dass u. U. eine geeignete Werkzeugverwaltung dafür sorgt, dass das zum Einwechseln vorgesehene Werkzeug an der Werkzeugwechselposition bereit steht.

17.8 Beispiele

Fräsmaschine

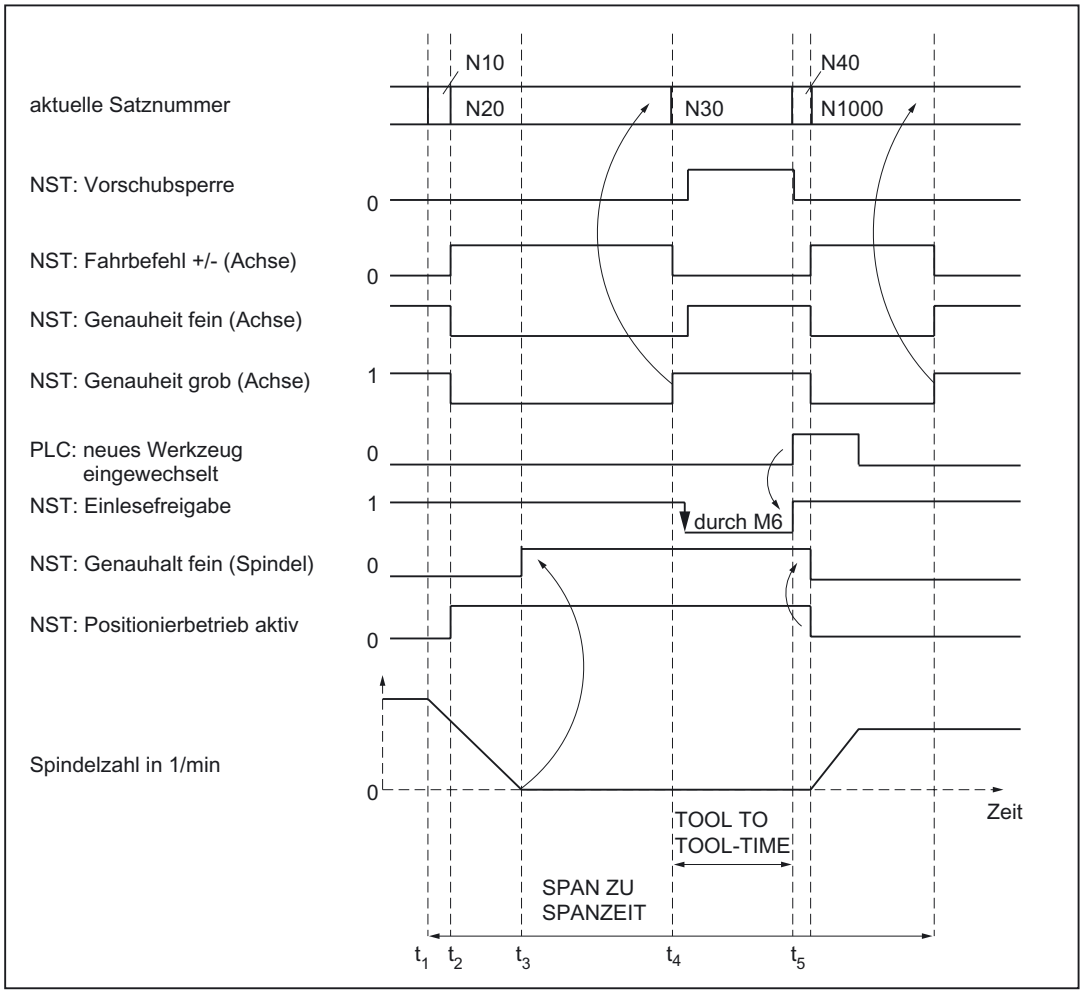
Im folgenden Beispiel wird ein typischer zeitlicher Ablauf der Span-zu-Span-Zeit mit einem Werkzeugwechsler und festem absolutem Werkzeugwechselpunkt an einer Fräsmaschine dargestellt.

Bearbeitungsprogramm:

```
N970 G0 X= Y= Z= LF                          ;  Freifahren von der Kontur
N980 T1 LF                                   ;  Werkzeugvorwahl
N990 W_WECHSEL LF                            ;  Unterprogrammaufruf ohne Parameter
N1000 G90 G0 X= Y= Z= M3 S1000 LF            ;  Weiterbearbeitung
```

Unterprogramm zum Werkzeugwechsel:

```
PROC W_WECHSEL LF
N10 SPOSA= S0 LF                             ;  Spindelpositionierung
N20 G75 FP=2 X1=0 Y1=0 Z1=0                  ;  Werkzeugwechselpunkt anfahren
N30 M06 LF                                   ;  Werkzeug wechseln
N40 M17 LF
```



t_1: Achsen stehen.
Spindel dreht.
Start des Werkzeugwechselzyklus in `N10`.

t_2: Achsen fahren mit `G75` in `N20` auf Werkzeugwechselpunkt.

t_3: Spindel erreicht programmierte Position aus Satz `N10`.

t_4: Achsen erreichen Genauhalt grob aus `N20`; damit beginnt `N30`:
Mit `M06` wird das bisherige Werkzeug aus der Spindel entfernt, das neue Werkzeug eingesetzt und gespannt.

t_5: Werkzeugwechsler ist in Ausgangsposition zurückgeschwenkt.

Bild 17-1 Zeitablauf des Werkzeugwechsels

Damit kann in `N1000` des aufrufenden Hauptprogramms:

- die neue Werkzeugkorrektur angewählt werden.
- die Achsen wieder an die Kontur zurückgeführt werden.
- die Spindel beschleunigt werden.

17.9 Datenlisten

17.9.1 Maschinendaten

17.9.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
18082	MM_NUM_TOOL	Anzahl der Werkzeuge

17.9.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
22200	AUXFU_M_SYNC_TYPE	Ausgabezeitpunkt der M-Funktionen
22220	AUXFU_T_SYNC_TYPE	Ausgabezeitpunkt der T-Funktionen
22550	TOOL_CHANGE_MODE	Neue Werkzeugkorrektur bei M-Funktion
22560	TOOL_CHANGE_M_CODE	M-Funktion für Werkzeugwechsel

17.9.1.3 Achs/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: $MA_	Beschreibung
30600	FIX_POINT_POS[n]	Festwertpositionen der Maschinenachsen bei G75

17.9.2 Signale

17.9.2.1 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
M-Funktion M06	DB21, ...DBX194.6	DB2500.DBB1000.6

W4: Schleifspezifische Werkzeugkorrektur und Überwachungen - nur 840D sl

18

18.1 Kurzbeschreibung

Inhalt

Themen dieser Funktionsbeschreibung sind:

- Schleifspezifische Werkzeugkorrektur
- Online-Werkzeugkorrektur (Continous Dressing)
- Schleifspezifische Werkzeugüberwachung
- Konstante Scheibenumfangsgeschwindigkeit (SUG)

Literatur

Grundlagen siehe:
/FB1/ Funktionshandbuch Grundfunktionen; Werkzeugkorrektur (W1)

Programmierung, Wirkungsweise und Handhabung siehe:
/PG/ Programmierhandbuch Grundlagen

18.2 Schleifspezifische Werkzeugkorrektur

18.2.1 Struktur der Werkzeugdaten

Schleifwerkzeuge

Schleifwerkzeuge sind Werkzeuge mit Typ 400 bis 499.

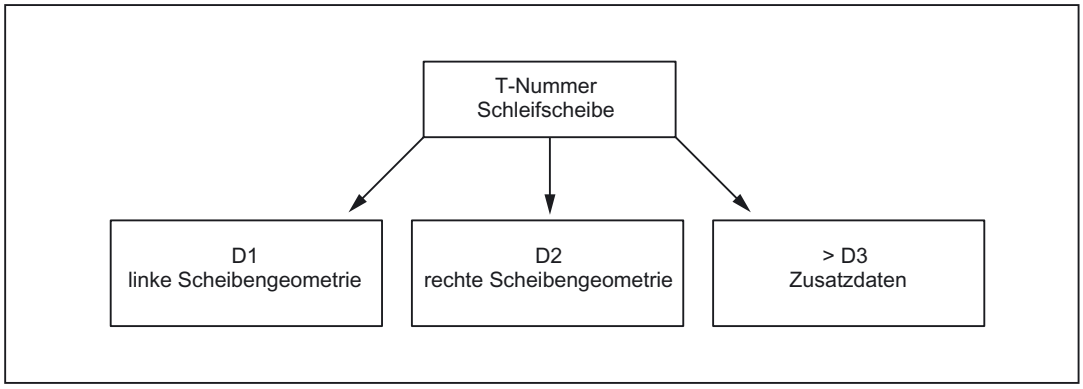
Werkzeugkorrektur für Schleifwerkzeuge

Schleifwerkzeuge besitzen in der Regel neben schneidenspezifischen auch werkzeug- und abrichterspezifische Daten.

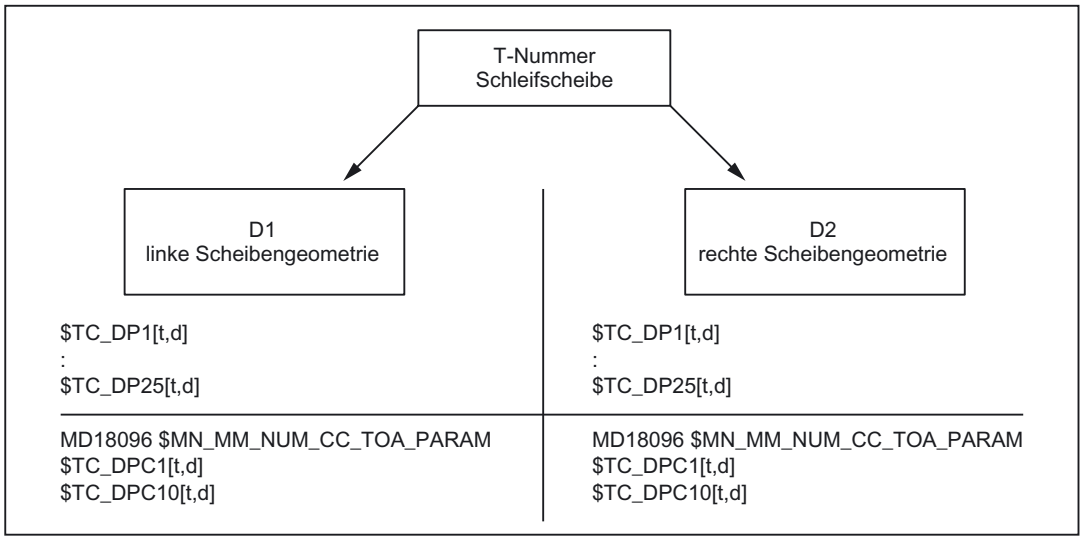
Es können unter einer T-Nummer die schleifscheibenspezifischen Daten für die linke und rechte Scheibengeometrie in `D1` bzw. `D2` abgelegt werden.

Werden Daten für die Abrichtergeometrie benötigt, können sie z. B. beginnend ab `D3` einer T-Nummer oder in zusätzliche schneidenspezifische Daten hinterlegt werden (MD18096 $MN_MM_NUM_CC_TOA_PARAM).

Beispiel 1:



Beispiel 2:



Unter einer T-Nummer können alle zu einer Schleifscheibe und Abrichter gehörenden Korrekturen in den WZ-Schneiden `D1`, `D2` für die Schleifscheibe und z. B. `D3`, `D4` für den Abrichter wie folgt zusammengefasst werden:

- `D1`: Schleifscheibengeometrie links
- `D2`: Schleifscheibengeometrie rechts
- `D3`: Abrichtergeometrie links
- `D4`: Abrichtergeometrie rechts

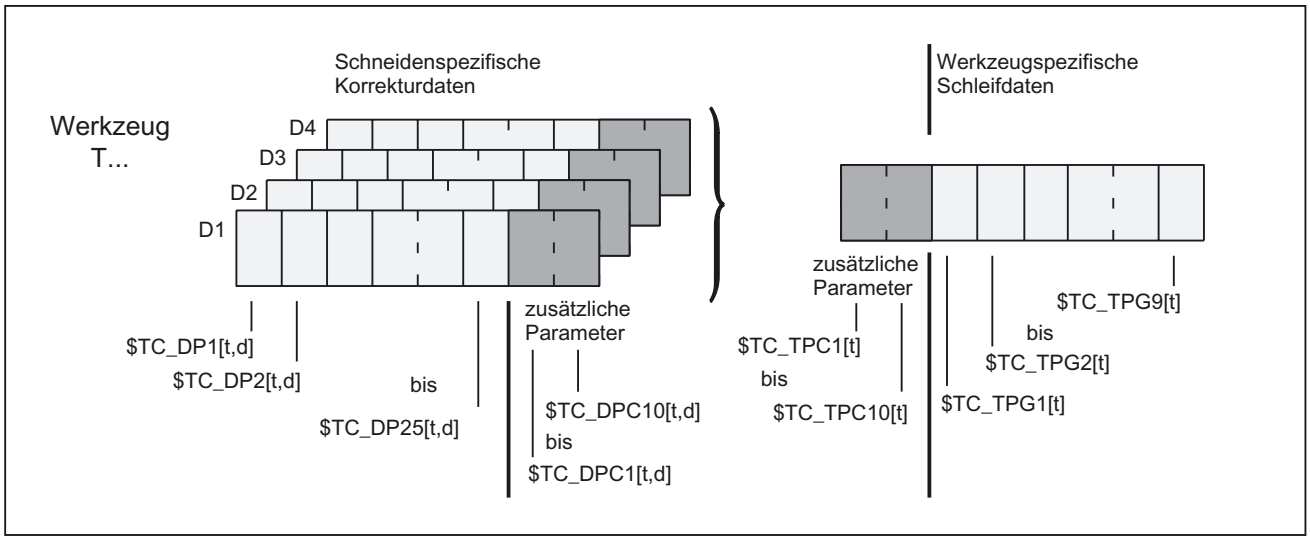


Bild 18-1 Struktur der Werkzeugkorrekturdaten bei Schleifwerkzeugen

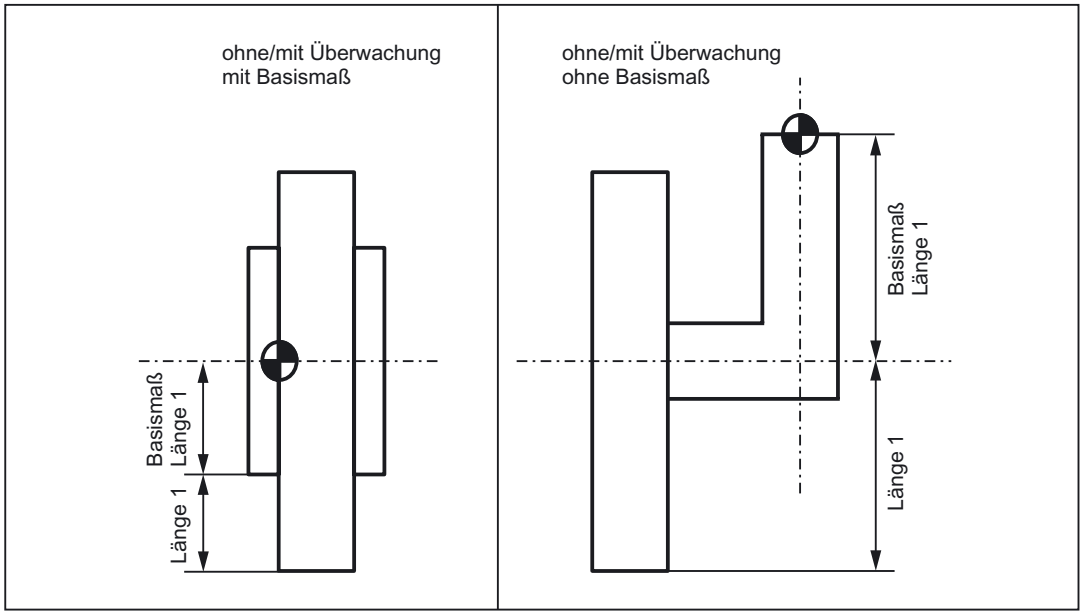
18.2.2 Schneidenspezifische Korrekturdaten

WZ-Parameter

Die Werkzeugparameter für Schleifwerkzeuge haben die gleiche Bedeutung wie für Dreh- und Fräswerkzeuge.

WZ-Parameter	Bedeutung	Bemerkung	
1	Werkzeugtyp		
2	WZ-Schneidenlage	nur für Dreh- und Schleifwerkzeuge	
Geometrie - Werkzeuglängenkorrektur			
3	Länge 1		
4	Länge 2		
5	Länge 3		
Geometrie - Werkzeugradiuskorrektur			
6	Radius 1		
7			reserviert[1)]
8			reserviert[1)]
9			reserviert[1)]
10			reserviert[1)]
11			reserviert[1)]
Verschleiß - Werkzeuglängenkorrektur			
12	Länge 1		
13	Länge 2		
14	Länge 3		
Verschleiß - Werkzeugradiuskorrektur			
15	Radius 1		
16			reserviert[1)]
17			reserviert[1)]
18			reserviert[1)]
19			reserviert[1)]
20			reserviert[1)]
Basismaß/Adaptermaß - Werkzeuglängenkorrektur			
21	Basis-Länge 1		
22	Basis-Länge 2		
23	Basis-Länge 3		
Technologie			
24	Freischneidwinkel	nur für Drehwerkzeuge	
25			reserviert[1)]

[1)] "Reserviert" bedeutet, dass dieser WZ-Parameter der 840D nicht benutzt wird (reserviert für Erweiterungen).



Hinweis

Die Schneidendaten für `D1` und `D2` eines gewählten Schleifwerkzeuges können verkettet werden, d. h., wird ein Parameter in `D1` bzw. `D2` geändert, wird automatisch der gleiche Parameter in `D1` bzw. `D2` mit dem neuen Wert überschrieben (siehe werkzeugspezifisches Datum $TC_TPG2).

Definition zusätzlicher Parameter $TC_DPC1...10

Für anwenderbezogene Schneidendaten können die zusätzlichen Parameter:
$TC_DPC1 bis 10
unabhängig vom WZ-Typ eingerichtet werden über das allgemeine Maschinendatum:
MD18096 $MN_MM_NUM_CC_TOA_PARAM

⚠ VORSICHT

Eine Änderung des MDs ist nach POWER ON wirksam und hat eine Speicherformatierung zur Folge (eventuell Datensicherung vorher durchführen!).

Ein automatischer Wechsel zwischen Scheibenkorrektur links und rechts findet beim Konturschleifen **nicht** statt. Er muss programmiert werden.

Werkzeugtypen für Schleifwerkzeuge

Die Werkzeugtypen für Schleifwerkzeuge sind wie folgt aufgebaut:

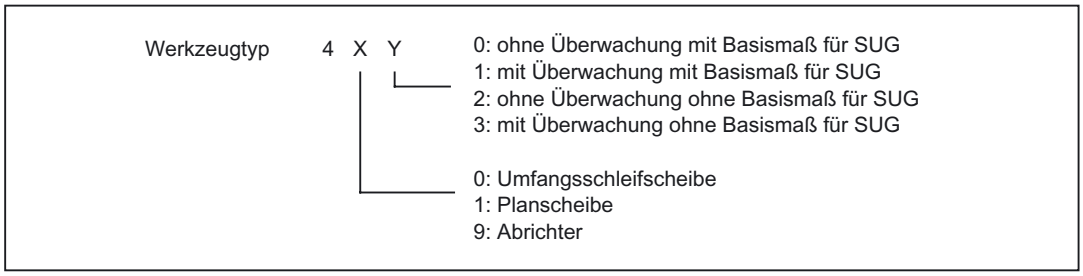


Bild 18-2 Aufbau des Werkzeugtyps bei Schleifwerkzeugen

Hinweis

MD20350 $MC_TOOL_GRIND_AUTO_TMON

Über dieses kanalspezifische Maschinendatum kann festgelegt werden, ob für Schleifwerkzeuge **mit Überwachung** (d. h. ungerade Werkzeugtypen) die Überwachung bei Anwahl dieses Werkzeugs bereits aktiv ist oder nicht.

Beispiele:

Es können daraus folgende WZ-Typen konstruiert werden:

Typ	Beschreibung
400	Umfangsschleifscheibe
401	Umfangsschleifscheibe mit Überwachung mit Basismaß für SUG
403	Umfangsschleifscheibe mit Überwachung ohne Basismaß für SUG
410	Planscheibe
411	Planscheibe mit Überwachung mit Basismaß für SUG
413	Planscheibe mit Überwachung ohne Basismaß für SUG
490	Abrichter

18.2.3 Werkzeugspezifische Schleifdaten

Werkzeugspezifische Schleifdaten

Die werkzeugspezifischen Schleifdaten sind für jede T-Nummer (Typ 400 - 499) einmal vorhanden. Sie werden automatisch mit jedem neuen Schleifwerkzeug (Typ 400 - 499) angelegt.

Hinweis

Die werkzeugspezifischen Schleifdaten verhalten sich wie eine Schneide.

Bei der Angabe der Anzahl Schneiden ist dies gegebenenfalls zu berücksichtigen:

MD18100 $MN_MM_NUM_CUTTING_EDGES_IN_TOA

Mit dem Löschen aller Schneiden eines Werkzeugs werden auch automatisch die vorhandenen werkzeugspezifischen Schleifdaten gelöscht.

Parameter

Die Parameter sind wie folgt belegt:

Parameter	Bedeutung	Datentyp
$TC_TPG1	Spindelnummer	Integer
$TC_TPG2	Verkettungsvorschrift	Integer
$TC_TPG3	Minimaler Scheibenradius	Real
$TC_TPG4	Minimale Scheibenbreite	Real
$TC_TPG5	Aktuelle Scheibenbreite	Real
$TC_TPG6	Maximale Drehzahl	Real
$TC_TPG7	Maximale Umfangsgeschwindigkeit	Real
$TC_TPG8	Winkel der schrägen Scheibe	Real
$TC_TPG9	Parameter-Nummer für Radiusberechnung	Integer
Zusätzliche Parameter (Anwenderspezifische Schneidendaten)		
$TC_TPC1 bis $TC_TPC10		Real

Definition zusätzlicher Parameter $TC_DPC1...10

Für anwenderbezogene Schneidendaten können die zusätzlichen Parameter $TC_DPC1 bis $TC_DPC10 unabhängig vom WZ-Typ eingerichtet werden. Dies erfolgt über das allgemeine Maschinendatum:

MD18096 $MN_MM_NUM_CC_TDA_PARAM

> ⚠ VORSICHT
>
> Eine Änderung des MDs ist nach POWER ON wirksam und hat eine Speicherformatierung zur Folge (eventuell Datensicherung vorher durchführen!).

Spindelnummer $TC_TPG1

Nummer der zu überwachenden (z. B. Scheibenradius und Scheibenbreite) und programmierten (z. B. Scheibenumfangsgeschwindigkeit) Spindel.

Verkettungsvorschrift $TC_TPG2

Mit diesem Parameter wird festgelegt, welche WZ-Parameter von Schneide 2 (D2) und Schneide1 (D1) miteinander verkettet sein sollen. Eine Änderung des Wertes einer der verketteten Parameter wird dann automatisch beim Verketten des Parametes der anderen Schneide übernommen.

WZ-Parameter	Bedeutung	Bit in $TC_TPG2	Hex	Dez
$TC_DP1	Werkzeugtyp	0	0001	1
$TC_DP2	Schneidenlage	1	0002	2
Geometrie - Werkzeuglängenkorrektur				
$TC_DP3	Länge 1	2	0004	8
$TC_DP4	Länge 2	3	0008	16
$TC_DP5	Länge 3	4	0010	32
$TC_DP6	Radius	5	0020	64
$TC_DP7	reserviert	6	0040	128
$TC_DP8		7	0080	256
$TC_DP9		8	0100	512
$TC_DP10		9	0200	1024
$TC_DP11	reserviert	10	0400	2048
Verschleiß - Werkzeuglängenkorrektur				
$TC_DP12	Länge 1	11	0800	4096
$TC_DP13	Länge 2	12	1000	8192
$TC_DP14	Länge 3	13	2000	16384
$TC_DP15	Radius	14	4000	32768
$TC_DP16	reserviert	15	8000	65536
$TC_DP17		16	10000	131072
$TC_DP18		17	20000	262144
$TC_DP19		18	40000	524288
$TC_DP20	reserviert	19	80000	1048576
Basismaß/Adaptermaß - Werkzeuglängenkorrektur				
$TC_DP21	Basis-Länge 1	20	100000	2097152
$TC_DP22	Basis-Länge 2	21	200000	4194304

WZ-Parameter	Bedeutung	Bit in $TC_TPG2	Hex	Dez
$TC_DP23	Basis-Länge 3	22	400000	8388608
Technologie				
$TC_DP24	reserviert	23	800000	16777216
$TC_DP25	reserviert	24	1000000	33554432

Beispiel für eine Verkettung:

Bei einem Schleifwerkzeug (im Beispiel T1) sollen die Längen 1, 2 und 3 der Geometrie, der Verschleiß der Längen und die Basis-/Adaptermaße der Längen 1, 2 und 3 automatisch übernommen werden.

Desweiteren gilt der gleiche Werkzeugtyp für die Schneiden 1 und 2.

Werkzeugtyp	$TC_DP1	Bit 0
Länge 1	$TC_DP3	Bit 2
Länge 2	$TC_DP4	Bit 3
Länge 3	$TC_DP5	Bit 4
Verschleiß		
Länge 1	$TC_DP12	Bit 11
Länge 2	$TC_DP13	Bit 12
Länge 3	$TC_DP14	Bit 13
Basis-/Adaptermaß		
Länge 1	$TC_DP21	Bit 20
Länge 2	$TC_DP22	Bit 21
Länge 3	$TC_DP23	Bit 22

Somit muss der Parameter $TC_TPG2 wie folgt belegt werden:

binär:	$TC_TPG2[1]= 'B111 0000 0011 1000 0001 1101' (Bit 22 ... Bit 0)
hexadezimal:	$TC_TPG2[1]= 'H70381D'
dezimal:	$TC_TPG2[1]='D7354397'

Hinweis

Wird die Verkettungsvorschrift nachträglich geändert, so werden die Werte der beiden Schneiden nicht automatisch abgeglichen, sondern erst nach einer Änderung eines Parameters.

Minimaler Scheibenradius und -breite $TC_TPG3 $TC_TPG4

In diese Parameter sind die Grenzwerte für Radius und Breite der Schleifscheibe einzugeben. Für die Geometrieüberwachung der Schleifscheibe werden diese Parameterwerte verwendet.

Hinweis

Bei schräger Scheibe ist zu beachten, dass der minimale Scheibenradius in kartesischen Koordinaten anzugeben ist. Werden minimale Breite bzw. Radius unterschritten, erfolgt eine Reaktion an der PLC-Nahtstelle. Der Anwender kann mit diesen Signalen seine Fehlerstrategie festlegen.

Aktuelle Breite $TC_TPG5

Die Breite der Schleifscheibe, die sich z. B. nach dem Abrichten ergibt, wird hier eingetragen.

Maximale Drehzahl und Umfangsgeschwindigkeit $TC_TPG6 $TC_TPG7

In diese Parameter sind die oberen Grenzwerte für Drehzahl und Umfangsgeschwindigkeit der Schleifscheibe einzugeben.

Voraussetzung: Eine Spindel ist deklariert.

Winkel der schrägen Scheibe $TC_TPG8

Dieser Parameter gibt die Neigung der schrägen Scheibe in der aktuellen Ebene an. Er wird für SUG ausgewertet.

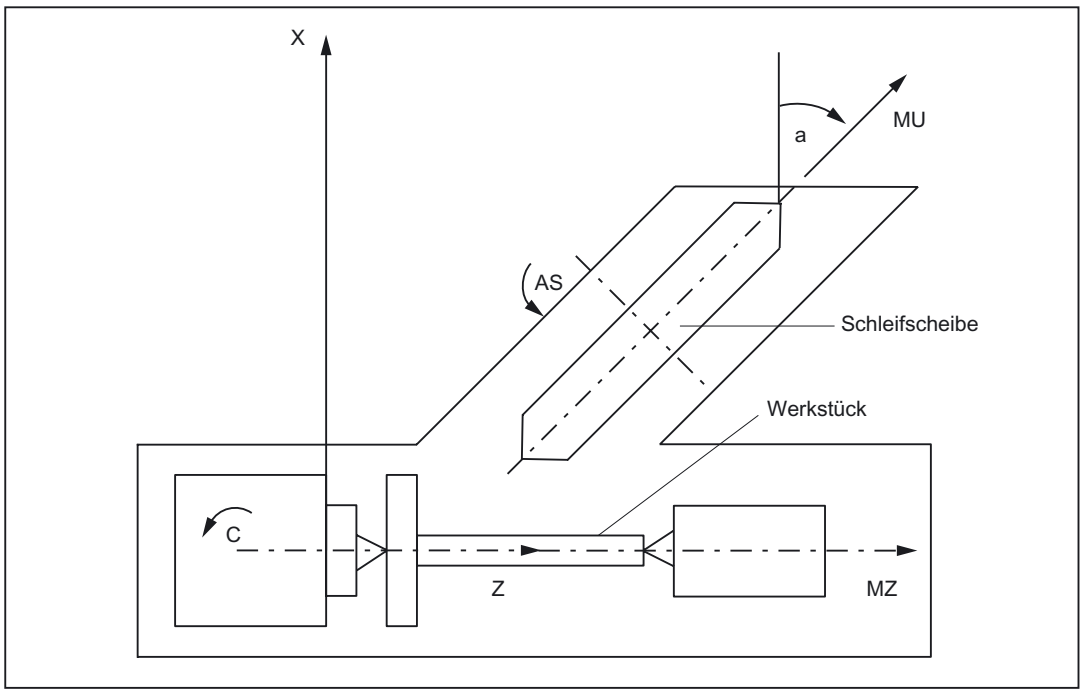


Bild 18-3 Maschine mit schräg stehender Zustellachse

Hinweis

Eine automatische Korrektur der Werkzeuglängen bei einer Änderung des Winkels erfolgt nicht.

Der Winkel muss im Bereich -90° ≤ $TC_TPG8 < +90° liegen.

Bei Schrägachsmaschinen muss der gleiche Winkel für die schräge Achse und für die schräge Scheibe verwendet werden.

Parameter-Nr. für Radiusberechnung $TC_TPG9

Mit diesem Parameter wird festgelegt, welche Korrekturwerte für die SUG-Berechnung und die WZ-Überwachung des minimalen Scheibenradius ($TC_TPG3) genutzt werden sollen.

$TC_TPG9 = 3	Länge 1 (Geometrie + Verschleiß + Basis, abhängig vom WZ-Typ)
$TC_TPG9 = 4	Länge 2 (Geometrie + Verschleiß + Basis, abhängig vom WZ-Typ)
$TC_TPG9 = 5	Länge 3 (Geometrie + Verschleiß + Basis, abhängig vom WZ-Typ)
$TC_TPG9 = 6	Radius

Zugriff aus dem Teileprogramm

Vom Teileprogramm aus können Parameter gelesen und geschrieben werden.

Beispiel	Programmierung
Lesen der aktuellen Breite von Werkzeug 2 und ablegen in R10	`R10 = $TC_TPG5 [2]`
Schreiben der maximalen Drehzahl von Werkzeug 3 mit dem Wert 2000	`$TC_TPG6 [3] = 2000`

$P_ATPG[m] für aktuelles Werkzeug

Über diese Systemvariable kann auf die werkzeugspezifischen Schleifdaten für das **aktuelle** Werkzeug zugegriffen werden.

m: Parameter-Nummer (Datentyp: Real)

Beispiel:

Parameter 3 ($TPG3[<T-Nr.>])

$P_ATPG[3]=R10

Hinweis

Die Überwachungsdaten gelten sowohl für die linke als auch die rechte Schneide der Schleifscheibe.

Wirksam werden die werkzeugspezifischen Schleifdaten bei Programmierung von `GWPSON` (Anwahl der konstanten Scheibenumfangsgeschwindigkeit) und `TMON` (Anwahl der Werkzeugüberwachung). Soll ein geändertes Datum wirksam werden, so muss `GWPSON` bzw. `TMON` erneut programmiert werden.

Die Längenkorrekturen geben in jedem Fall die Abstände zwischen Werkzeugträgerbezugspunkt und Werkzeugspitze in den kartesischen Koordinaten an (bei schräger Scheibe ist dies zu beachten).

18.2.4 Schleifwerkzeuge - Beispiele

Zuordnung der Längenkorrekturen

Die Zuordnung der Längenkorrekturen für die Geometrieachsen bzw. die Radiuskorrektur in der Ebene richtet sich nach der aktuellen Ebene.

Ebenen

Folgende Ebenen und Achszuordnungen sind möglich (Abszisse, Ordinate, Applikate zu 1., 2. und 3. Geometrieachse):

Befehl	Ebene (Abszisse / Ordinate)	Senkrechte Achse darauf (Applikate)
G17	X / Y	Z
G18	Z / X	Y
G19	Y / Z	X

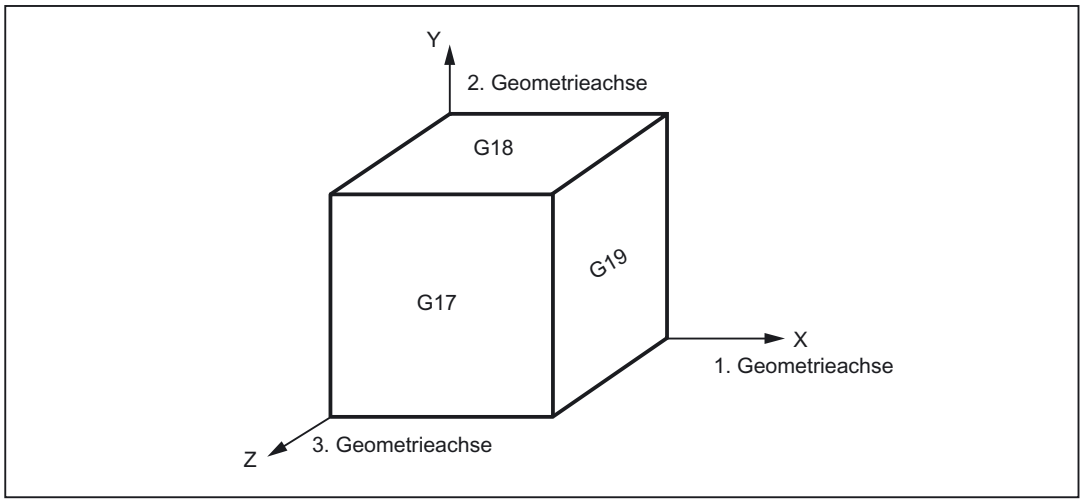


Bild 18-4 Ebenen und Achszuordnung

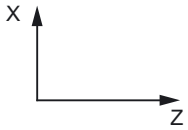
Umfangsschleifscheibe

Einträge in Werkzeugparameter

$TC_DP1	400
$TC_DP3	Länge 1
$TC_DP4	Länge 2
$TC_DP6	Radius

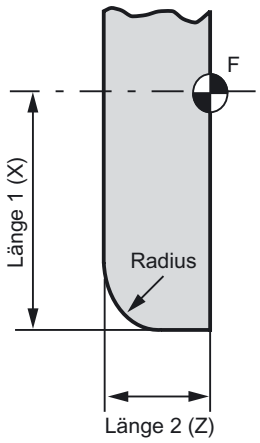
Die nicht verwendeten Parameter sind auf 0 zu setzen

z. B.
G18: Z / X-Ebene



Wirkung

G17	Länge 1 in Y Länge 2 in X Radius in X / Y
G18	Länge 1 in X Länge 2 in Z Radius in Z / X
G19	Länge 1 in Z Länge 2 in Y Radius in Y / Z



F: Werkzeugträgerbezugspunkt

Bild 18-5 Erforderliche Korrekturwerte einer Umfangsschleifscheibe

Schräge Scheibe

Ohne Basismaß für SUG

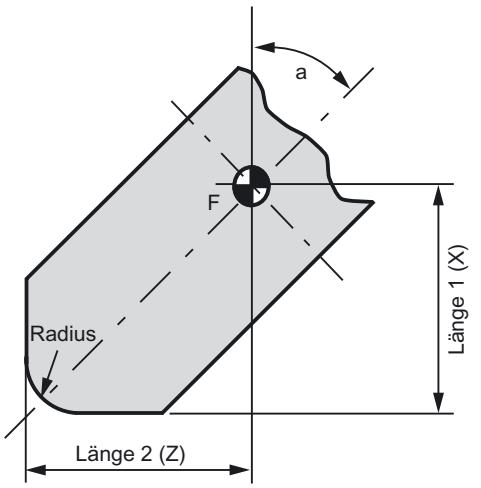
Einträge in Werkzeugparameter

$TC_DP1	403
$TC_DP3	Länge 1
$TC_DP4	Länge 2
$TC_DP6	Radius

Die nicht verwendeten Parameter sind auf 0 zu setzen

Wirkung

G17	Länge 1 in Y Länge 2 in X Radius in X / Y
G18	Länge 1 in X Länge 2 in Z Radius in Z / X
G19	Länge 1 in Z Länge 2 in Y Radius in Y / Z



F: Werkzeugträgerbezugspunkt

Bild 18-6 Erforderliche Korrekturwerte bei Schräger Scheibe mit impliziter Überwachungsanwahl

Schräge Scheibe

Mit Basismaß für SUG

Einträge in Werkzeugparameter	
$TC_DP1	401
$TC_DP3	Länge 1
$TC_DP4	Länge 2
$TC_DP6	Radius
$TC_DP21	L1 Basis
$TC_DP22	L2 Basis

Verschleißwerte entsprechend Erfordernis

Die nicht verwendeten Parameter sind auf 0 zu setzen

Wirkung	
G17	Länge 1 in Y Länge 2 in X Radius in X / Y
G18	Länge 1 in X Länge 2 in Z Radius in Z / X
G19	Länge 1 in Z Länge 2 in Y Radius in Y / Z

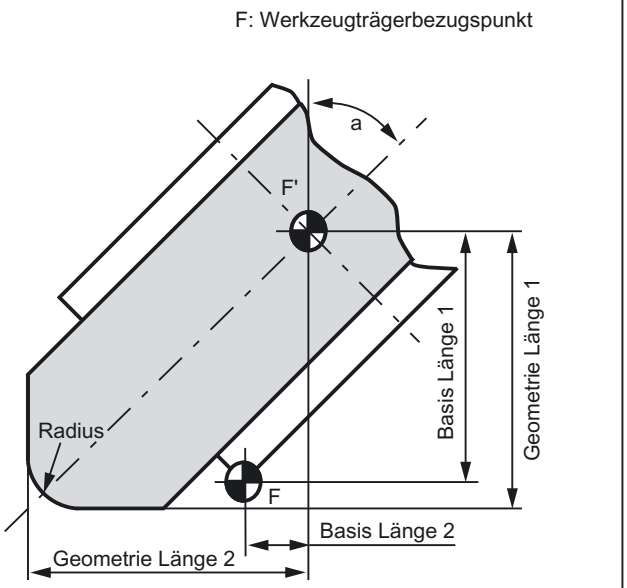


Bild 18-7 Erforderliche Korrekturwerte am Beispiel Schräge Scheibe mit impliziter Überwachungsanwahl und mit Basismaß für SUG-Berechnung

Umfangsschleifscheibe

Einträge in Werkzeugparameter

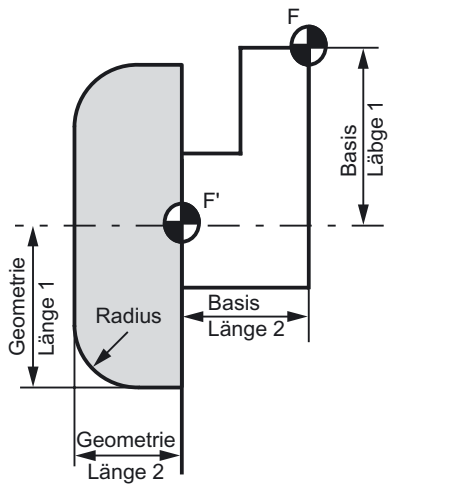
$TC_DP1	403
$TC_DP3	Länge 1
$TC_DP4	Länge 2
$TC_DP6	Radius
$TC_DP21	L1 Basis
$TC_DP22	L2 Basis

Verschleißwerte entsprechend Erfordernis

Die nicht verwendeten Parameter sind auf 0 zu setzen

Wirkung

G17	Länge 1 in Y Länge 2 in X Radius in X / Y
G18	Länge 1 in X Länge 2 in Z Radius in Z / X
G19	Länge 1 in Z Länge 2 in Y Radius in Y / Z



F: Werkzeugträgerbezugspunkt

Bild 18-8 Erforderliche Korrekturwerte einer Umfangsschleifscheibe ohne Basismaß für SUG

Planscheibe

Einträge in Werkzeugparameter

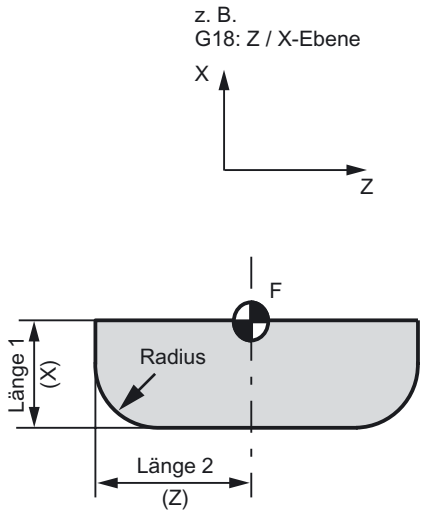
$TC_DP1	411
$TC_DP3	Länge 1
$TC_DP4	Länge 2
$TC_DP6	Radius

Verschleißwerte entsprechend Erfordernis

Die nicht verwendeten Parameter sind auf 0 zu setzen

Wirkung

G17	Länge 1 in Y Länge 2 in X Radius in X / Y
G18	Länge 1 in X Länge 2 in Z Radius in Z / X
G19	Länge 1 in Z Länge 2 in Y Radius in Y / Z



F: Werkzeugträgerbezugspunkt

Bild 18-9 Erforderliche Korrekturwerte einer Planscheibe mit Überwachungsparametern

18.3 Online-Werkzeugkorrektur

18.3.1 Allgemeines

Anwendung

Schleifen bedeutet zum einen das Bearbeiten eines Werkstücks und zum anderen das Abrichten der Schleifscheibe. Dies kann sowohl in einem Kanal als auch in getrennten Kanälen erfolgen.

Damit das Abrichten während der Schleifbearbeitung möglich wird, muss die Maschine eine Funktion bieten, die die Verkleinerung der Schleifscheibe durch das Abrichten am Werkstück kompensiert. Das kann mit der Funktion "Online-Werkzeugkorrektur" (Continuous Dressing) realisiert werden.

Abrichten während der Bearbeitung

Damit eine Bearbeitung während dem Abrichten möglich ist, muss die durch das Abrichten der Schleifscheibe entstehende Verkleinerung der Schleifscheibe als Werkzeugkorrektur dem aktuellen Werkzeug im Bearbeitungskanal übergeben und sofort wirksam werden.

Das kann mit der Funktion "Continuous Dressing, (paralleles Abrichten), Online-Werkzeugkorrektur" realisiert werden.

Hinweis

Die Online-Werkzeugkorrektur kann nur bei Schleifwerkzeugen angewendet werden.

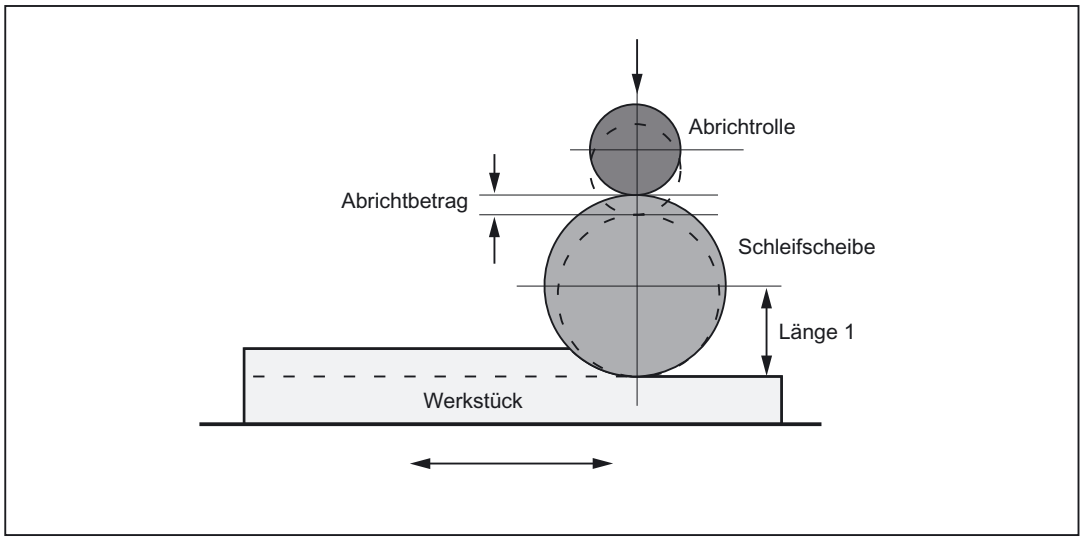


Bild 18-10 Abrichten während der Bearbeitung mit einer Abrichtrolle

Allgemeines

Die Online-Werkzeugkorrektur kann für jedes Schleifwerkzeug in jedem Kanal aktiviert werden.

Im Allgemeinen wirkt die Online-Werkzeugkorrektur auf den Längenverschleiß. Die Zuordnung der Längen zu den Geometrieachsen erfolgt anhand der aktuellen Ebene entsprechend des Werkzeugtyps wie bei Geometrie und Verschleiß.

Die Überwachung der Schleifspindel wirkt auch bei Online-Werkzeugkorrektur.

Hinweis

Korrigiert wird immer der Verschleißparameter der gewählten Länge. Ist die Längenkorrektur für mehrere Schneiden identisch, so muss über Verkettungsvorschrift dafür gesorgt werden, dass die Werte automatisch auch für die 2. Schneide korrigiert werden.

Ist im Bearbeitungskanal Online-Korrekturen aktiv, dürfen die Verschleißwerte für das aktive Werkzeug in diesem Kanal nicht aus dem Bearbeitungsprogramm oder über Bedienung geändert werden.

Eine Änderung des Verschleißes für den Radius (P15) wird erst bei Neuanwahl des Werkzeuges berücksichtigt.

Die Online-Korrektur wirkt auch für die konstante Scheibenumfangsgeschwindigkeit (SUG), d. h. die Spindeldrehzahl wird um den entsprechenden Wert korrigiert.

Befehle

Für die Online-Werkzeugkorrektur gibt es folgende Befehle:

Befehl	Bedeutung
FCTDEF <Polynom-Nr>, <unterer Grenzwert>, <oberer Grenzwert>, <Koeffizient 0>, <Koeffizient 1>, <Koeffizient 2>, <Koeffizient 3>)	Funktion parametrieren (Polynom bis 3. Grad) (Fine Tool Offset Definition)
PUTFTOCF (<Polynom-Nr>, <Bezugswert>, <Länge1_2_3>, <Kanalnr>, <Spindelnr>)	Online-Werkzeugkorrektur schreiben kontinuierlich (Put Fine Tool Offset Compensation)
PUTFTOC (<Wert>, <Länge1_2_3>, <Kanalnr>, <Spindelnr>)	Online-Werkzeugkorrektur schreiben diskret (Put Fine Tool Offset Compensation)
FTOCON	Einschalten der Online-Werkzeugkorrektur (Fine Tool Offset Compensation On)
FTOCOF	Ausschalten der Online-Werkzeugkorrektur (Fine Tool Offset Compensation Off)

Hinweis

Änderungen der Korrekturwerte im TOA-Speicher werden nur bei erneuter T- oder D-Programmierung wirksam.

Literatur:
/PGA/ Programmierhandbuch Arbeitsvorbereitung

18.3.2 Online-Werkzeugkorrektur schreiben: kontinuierlich

FCTDEF

Bestimmte Abrichtstrategien (z. B. Abrichtrolle) zeichnen sich dadurch aus, dass die Schleifscheibe kontinuierlich (linear) mit Zustellung der Abrichtrolle am Radius abnimmt. Hierfür benötigt man eine lineare Funktion zwischen Zustellung der Abrichtrolle und dem Schreiben des Verschleißwertes der jeweiligen Länge.

Die Funktion `FCTDEF` erlaubt es, 3 unabhängige Funktionen nach folgender Syntax zu definieren.

Funktion parametrieren

Die Parametrierung der Funktion erfolgt in einem eigenen Satz und hat folgende Syntax:

FCTDEF(<Polynom-Nr>, <unterer Grenzwert>, <oberer Grenzwert>, <Koeffizient a0>,<Koeffizient a1>, <Koeffizient a2>, <Koeffizient a3>)

`FCTDEF`	Function Definition
Polynom-Nr.:	Nummer der Funktion (z. B. 1, 2 oder 3)
unterer / oberer Grenzwert:	bestimmt den Wertebereich der Funktion; (Grenzwerte in Eingabefeinheiten)
Koeffizienten a_0, a_1, a_2:	Koeffizienten des Polynoms

Ein Polynom 3. Grades wird allgemein wie folgt definiert:

$y = a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3$

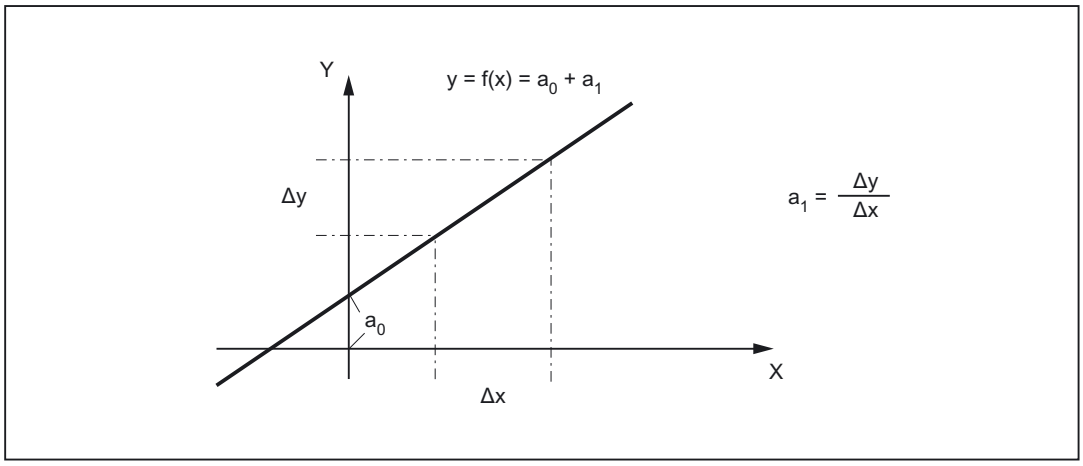


Bild 18-11 Geradengleichung

Hinweis

`FCTDEF` muss separat in einem eigenen NC-Satz programmiert werden.

Beispiel:

Gegeben: Steigung: $a_1 = +1$
$a_2 = 0$
$a_3 = 0$

Zum Definitionszeitpunkt soll der Funktionswert y = 0 sein und von der Maschinenachse XA (z. B. Abrichterachse) abgeleitet werden.

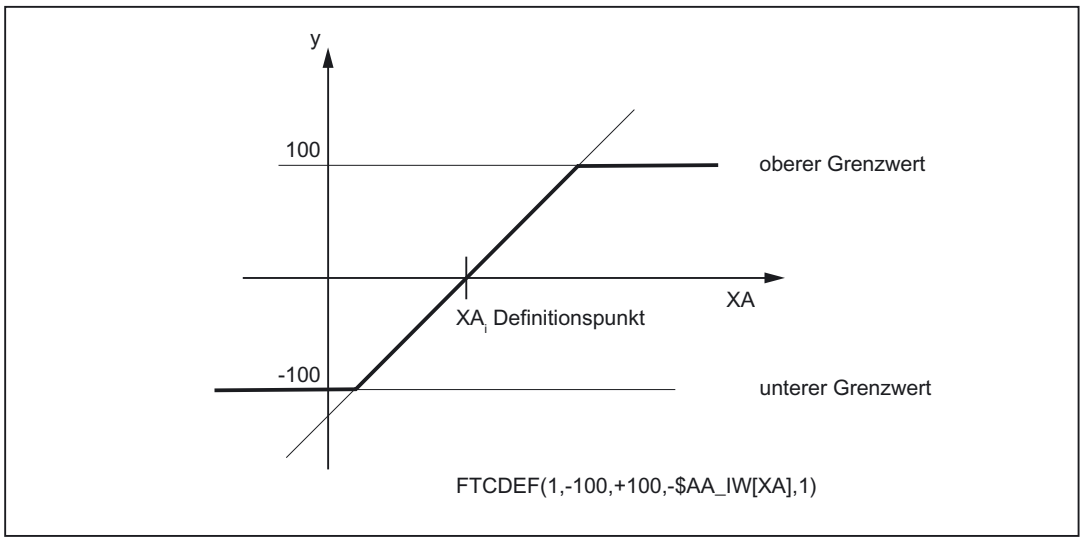


Bild 18-12 Gerade mit der Steigung 1

Online-WZK schreiben kontinuierlich

PUTFTOCF(<Polynom-Nr>, <Bezugswert>, <Länge1_2_3>, <Kanalnr>, <Spindelnr>)

PUFTOCF	
Polynom-Nr.:	Nummer der Funktion (1, 2, 3)
Bezugswert:	Bezugswert der Funktion
Länge 1_2_3:	Verschleißparameter in dem der Korrekturwert addiert wird
Kanalnr.:	Kanal in dem die Korrektur wirksam werden soll
Spindelnr.:	Spindel für die die Online-Korrektur wirken soll

Die Aktivierung der Online-Korrektur erfolgt vor dem Bewegungssatz der Abrichterachse.

Beispiel:

```
FCTDEF (1, -100, 100, -$AA_IW[X], 1)          ; Funktion definieren
PUTFTOCF (1, $AA_IW[X], 1, 2, 1)              ; Online-WZK schreiben
                                                kontinuierlich
```

Die Länge 1 des Werkzeuges für Spindel 1 im Kanal 2 wird abhängig von der Bewegung der X-Achse verändert.

Hinweis

Mit Angabe der Spindelnummer kann auch die Online-Werkzeugkorrektur für ein (geometrisch) nicht aktives Schleifwerkzeug aktiviert werden.

Wenn die Kanalnummer entfällt, wird die Online-Korrektur im eigenen Kanal wirksam.

Wenn die Spindelnummer entfällt, wird die Online-Korrektur für das aktuelle Werkzeug wirksam.

Der Aufruf der Online Werkzeugkorrektur kann auch als Synchronaktion erfolgen.

Literatur:
/FBSY/ Funktionshandbuch Synchronaktionen

18.3.3 Online-Werkzeugkorrektur ein-/ausschalten

Ein-/Ausschalten der Online-WZK

Folgende Befehle aktivieren/deaktivieren die Online-Werkzeugkorrektur im Bearbeitungskanal (Schleif-, Zielkanal):

`FTOCON` Einschalten der Online-WZK

Der Bearbeitungskanal kann nur Online-Werkzeugkorrekturen verarbeiten (`PUTFTOC`), wenn die Online-Werkzeugkorrektur (`FTOCON`) aktiv ist. Sonst wird der Alarm 20204 "PUTFTOC-Kommando nicht erlaubt" ausgegeben.

`FTOCOF` Ausschalten der Online-WZK

Mit `FTOCOF` wird die Online-Werkzeugkorrektur ausgeschaltet. In dem jeweiligen Längenverschleiß bleiben die geschriebenen Werte erhalten.

Die Online-Korrektur wird im Basis-Koordinatensystem verfahren, d. h. auch bei gedrehtem Werkstückkoordinatensystem wirken die Längenkorrekturen immer parallel zu den Koordinaten des ungedrehten Systems.
Die Korrektur wird herausgefahren unabhängig davon, ob die zu korrigierende Achse im aktuellen Satz verfahren wird oder nicht.

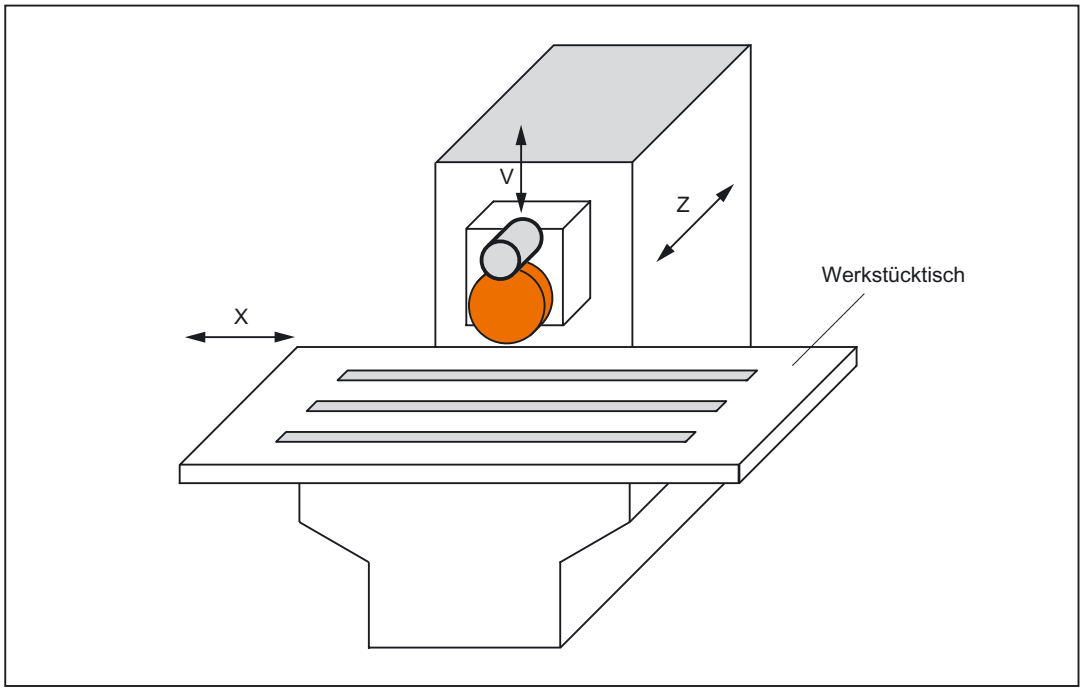
Hinweis

Der Befehl `FTOCON` muss in dem Kanal geschrieben werden, in dem die Korrektur wirksam werden soll (Bearbeitungskanal beim Schleifen).

`FTOCOF` ist immer Reset-Stellung. `PUTFTOC`-Befehle wirken nur bei aktivem Teileprogramm und wenn `FTOCON` aktiv ist.

18.3.4 Beispiel für Online-Werkzeugkorrektur schreiben kontinuierlich

Flachschleifmaschine



Y: Zustellachse für die Schleifscheibe
V: Zustellachse für die Abrichtrolle
X: Pendelachse, links - rechts

Ebene für die Werkzeugkorrektur: `G19` (Y/Z-Ebene)

Länge 1 wirkt in Z, Länge 2 wirkt in Y, WZ-Typ = 401

Bearbeitung:	Kanal 1 mit den Achsen Y, X
Abrichten:	Kanal 2 mit der Achse V

Aufgabe

Nach dem Beginn der Schleifbebearbeitung bei Y100 soll die Schleifscheibe um den Betrag 0,05 (in V-Richtung) abgerichtet werden. Der Abrichtbetrag soll mit Online-Werkzeugkorrektur kontinuierlich kompensiert werden.

Bearbeitungs-Hauptprogramm im Kanal 1

```
...
G1 G19 F10 G90                                  ;  Grundstellung
T1 D1                                           ;  aktuelles Werkzeug anwählen
S100 M3 Y100                                    ;  Spindel ein, Fahren auf
                                                   Ausgangsposition
FTOCON                                          ;  Online-Korrektur einschalten
INIT (2, "/_N_MPF_DIR/_N_ABRICHT_MPF", "S")     ;  Anwahl des Programms im
                                                   Kanal 2
START (2)                                       ;  Starten des Programms im
                                                   Kanal 2


Y200                                            ;  Fahren auf Zielposition
...
M30
```

Abrichtprogramm im Kanal 2_N_ABRICHT_MPF

```
...
FCTDEF (1, -1000, 1000, -$AA_IW[V], 1)          ;  Funktion definieren
PUTFTOCF (1, $AA_IW[V], 2, 1)                   ;  Online-WZK schreiben
                                                ;  kontinuierlich
U-0.05 G1 F0.01 G91                             ;  Zustellbewegung zum Abrichten
...
M30
```

Hinweis

Die Achse V arbeitet (richtet ab) parallel zu Y, d. h. Länge 2 wirkt in Y und muss somit kompensiert werden.

18.3.5 Online-Werkzeugkorrektur schreiben diskret

PUTFTOC

Mit diesem Befehl kann ein Korrekturwert per Programmbefehl geschrieben werden.

PUTFTOC(<Wert>, <Länge1_2_3>, <Kanalnr>, <Spindelnr>)

Put Fine Tool-Offset-Compensation

Der Verschleiß der angegebenen Länge 1, 2 oder 3 wird um den programmierten Wert online geändert.

Hinweis

Mit Angabe der Spindelnummer kann auch die Online-Werkzeugkorrektur für ein (geometrisch) nicht aktives Schleifwerkzeug aktiviert werden.

Wenn die Kanalnummer entfällt, wird die Online-Korrektur im eigenen Kanal wirksam.

Wenn die Spindelnummer entfällt, wird die Online-Korrektur für das aktuelle Werkzeug wirksam.

18.3.6 Hinweise zur Online-Korrektur

Verhalten bei Werkzeugwechsel

- Sofern seit dem letzten Werkzeug- bzw. Schneidenwechsel `FTOCON` aktiv war, wird bei Werkzeugwechsel steuerungsintern Vorlaufstopp mit Neusynchronisation ausgelöst.
- Schneidenwechsel ist ohne Vorlaufstopp möglich.

Hinweis

Der Werkzeugwechsel in Verbindung mit der Online-Werkzeugkorrektur kann durch Anwahl über T-Nummern durchgeführt werden.

Werkzeugwechsel mit `M6` ist in Verbindung mit der Online-Werkzeugkorrektur nicht möglich.

Bearbeitungsebene und Transformation

- `FTOCON` ist nur mit der Transformation "Schräge Achse" möglich.
- Ebenenwechsel (z. B. `G17` nach `G18`) und Transformationswechsel ist bei `FTOCON` nicht möglich, außer im Zustand `FTOCOF`.

Reset und Betriebsartenwechsel

- `NC-STOP` und Programmende mit `M2`/`M30` wird bei anstehender Online-Korrektur solange verzögert, bis die Korrektur herausgefahren ist.
- Bei `NC-RESET` wird die Online-Werkzeugkorrektur sofort abgewählt.
- Online-Werkzeugkorrekturen sind in der Betriebsart "Automatik" und bei aktivem Programm möglich.

Randbedingungen

- Die Online-Werkzeugkorrektur wird der programmierten Bewegung der Achse unter Berücksichtigung der vorgegebenen Grenzwerte überlagert (z. B. Geschwindigkeit).

  Stehen für eine Achse gleichzeitig DRF-Verschiebung und Online-Korrektur an, so wird zuerst die DRF-Verschiebung berücksichtigt.

- Die anstehende Korrektur wird mit der JOG-Geschwindigkeit unter Berücksichtigung der maximalen Beschleunigung herausgefahren.

  Bei `FTOCON` wird folgendes kanalspezifisches Maschinendatum berücksichtigt:

  MD20610 $MC_ADD_MOVE_ACCEL_RESERVE

  Damit kann eine Beschleunigungsreserve für die Bewegung reserviert werden, so dass die überlagerte Bewegung sofort ausgeführt werden kann.

- Bei Referenzpunktfahren mit `G74` wird die anstehende Online-Korrektur gelöscht.
- Bei einem Werkzeugwechsel mit `M6` wird die Feinkorrektur nicht abgewählt.

18.4 Online-Werkzeugradiuskorrektur

Allgemeines

Wenn die Werkzeuglängsachse und die Kontur senkrecht aufeinander stehen, dann kann die Korrekturgröße als Längenkorrektur auf eine der drei Geometrieachsen wirken (Online-Werkzeuglängenkorrektur).

Ist diese Bedingung nicht erfüllt, dann kann die Korrekturgröße als echte Radiuskorrektur eingegeben werden (Online-Werkzeugradiuskorrektur).

Freigabe der Funktion

Die Online-Werkzeugradiuskorrektur wird aktiviert über das Maschinendatum:

MD20254 $MC_ONLINE_CUTCOM_ENABLE (Freigabe Online-Werkzeugradiuskorrektur)

Ein-/Ausschalten

Die Online-Werkzeugradiuskorrektur wird mit den Befehlen `FTOCON` und `FTOCOF` ein-/ausgeschaltet (wie die Online-Werkzeuglängenkorrektur).

Parametrierung

Die Parametrierung der Online-Werkzeugkorrektur erfolgt mit den Befehlen `PUTFTOCF` und `PUTFTOC`. Der Parameter "LÄNGE 1_2_3" muss für die Online-Werkzeugradiuskorrektur wie folgt versorgt werden:

Parameter <Länge 1_2_3> = 4
Verschleißparameter in dem der Korrekturwert addiert wird.

Randbedingungen

- Die Werkzeugradiuskorrektur und damit auch die Online-Werkzeugradiuskorrektur kann nur dann aktiviert werden, wenn das angewählte Werkzeug einen Radius ungleich Null hat. Damit kann eine Bearbeitung ausschließlich mit der Online-Werkzeugradiuskorrektur nicht realisiert werden.
- Die Online-Korrekturwerte sollten klein gegenüber dem Originalradius sein, damit die Dynamikreserven bei den überlagerten Bewegungen nicht überschritten werden.
- Berücksichtigung der Online-Werkzeugradiuskorrektur bei Schleif- und Drehwerkzeugen (Typ 400 - 599) wird der Korrekturwert entsprechend der Schneidenlage berücksichtigt, d. h. bei aktiver Werkzeugradiuskorrektur wirkt er als Radiuskorrektur, bei ausgeschalteter Werkzeugradiuskorrektur als Längenkorrektur in den durch die Schneidenlage spezifizierten Achsen.

  Bei allen anderen Werkzeugtypen wird der Korrekturwert nur dann berücksichtigt, wenn die Werkzeugradiuskorrektur mit `G41` bzw. `G42` aktiviert wurde. Beim Ausschalten der Werkzeugradiuskorrektur mit `G40` wird der Korrekturwert herausgefahren.

18.5 Schleifspezifische Werkzeugüberwachung

18.5.1 Allgemeines

Aktivierung

Die Werkzeugüberwachung setzt sich aus Geometrie- und Drehzahlüberwachung zusammen und kann für jedes Schleifwerkzeug (WZ-Typ: 400 bis 499) aktiviert werden.

Anwahl

Die Anwahl erfolgt:

- durch Programmierung (`TMON`) im Teileprogramm

  oder

- automatisch durch die Anwahl der Werkzeuglängenkorrektur eines Schleifwerkzeuges mit ungerader Werkzeugtypnummer

Hinweis

Die automatische Anwahl der Überwachung muss eingestellt werden über das kanalspezifische Maschinendatum:

MD20350 $MC_TOOL_GRIND_AUTO_TMON.

Überwachung aktiv

Die Überwachung eines Schleifwerkzeuges bleibt solange aktiv, bis sie durch den Teileprogrammbefehl `TMOF` abgewählt wird.

Hinweis

Die Überwachung eines Werkzeuges wird durch die Anwahl der Überwachung eines anderen Werkzeuges nicht abgewählt, wenn sich die Werkzeuge auf unterschiedliche Spindeln beziehen.

Zu jedem Zeitpunkt kann für jede Spindel ein Werkzeug und somit **eine** Überwachung aktiv sein.

Aktive Überwachungen bleiben über RESET wirksam.

18.5.2 Geometrieüberwachung

Funktion

Überwacht werden kann:

- der aktuelle Scheibenradius

  und

- die aktuelle Scheibenbreite

Der aktuelle Scheibenradius wird mit dem im Parameter $TC_TPG3 stehenden Wert verglichen.
Der aktuelle Radius wird aus der im Parameter $TC_TPG9 vereinbarten Parameternummer der ersten Schneide (`D1`) eines Schleifwerkzeuges verglichen.

Die aktuelle Scheibenbreite wird in der Regel durch den Abrichtzyklus ermittelt und kann im Parameter $TC_TPG5 eines Schleifwerkzeuges eingetragen werden. Dieser eingetragene Wert wird bei aktiver Überwachung mit dem im Parameter $TC_TPG4 stehenden Wert verglichen.

Wann wird überwacht?

Die Überwachung für den Schleifscheibenradius wirkt auch bei aktiver Online-Werkzeugkorrektur:

- bei der Aktivierung der Überwachung
- beim Ändern des aktuellen Radius (Online-Werkzeugkorrektur, Verschleißparameter) bzw. der aktuellen Breite ($TC_TPG5)

Überwachungsreaktion

Unterschreitet zum einen der aktuelle Schleifscheibenradius den im Parameter $TC_TPG3 bzw. die aktuelle Schleifscheibenbreite ($TC_TPG5) den im Parameter $TC_TPG4 festgelegten Wert, wird an der PLC-Nahtstelle das achs-/spindelspezifische Bit DBX83.3 im DB31, ... auf "1" gesetzt.
Ansonsten ist der Wert "0".

DB31, ... DBX83.3 = 1 ⇒ Geometrieüberwachung hat angesprochen

DB31, ... DBX83.3 = 0 ⇒ Geometrieüberwachung hat nicht angesprochen

Hinweis

Eine steuerungsinterne Fehlerreaktion erfolgt nicht.

18.5.3 Drehzahlüberwachung

Funktion

Überwacht werden die maximale Scheibenumfangsgeschwindigkeit (Parameter $TC_TPG7) bzw. die maximale Spindeldrehzahl (Parameter $TC_TPG6).

Die Maßeinheit ist:

- Scheibenumfangsgeschwindigkeit $m * s^{-1}$
- Spindeldrehzahl min^{-1}

Die Überwachung erfolgt zyklisch. Dabei wird stets auf den zuerst erreichten Grenzwert begrenzt.

Wann wird überwacht?

Die Überwachung des Drehzahlsollwertes auf den Drehzahlgrenzwert erfolgt zyklisch unter Berücksichtigung des Spindel-Overrides.

Wann wird der Drehzahlgrenzwert neu bestimmt?

Der Drehzahlgrenzwert wird neu bestimmt:

- bei der Anwahl der Überwachung
- beim Ändern der Online-Korrekturwerte (Verschleißparameter)

Überwachungsreaktion

Beim Ansprechen der Drehzahlüberwachung wird:

- auf den Drehzahlgrenzwert begrenzt

  und

- das Nahtstellensignal:
  DB31, ... DBX83.6 (Drehzahlüberwachung)
  ausgegeben.

DB31, ... DBX83.6 = 1 ⇒ Grenzwert der Drehzahlüberwachung erreicht

DB31, ... DBX83.6 = 0 ⇒ Grenzwert der Drehzahlüberwachung nicht erreicht

Hinweis

Eine steuerungsinterne Fehlerreaktion erfolgt nicht.

18.5.4 An-/Abwahl der WZ-Überwachung

Teileprogrammbefehle

Zur An- und Abwahl der schleifspezifischen Werkzeugüberwachung eines aktiven und eines nicht aktiven Werkzeuges stehen folgende Teileprogrammbefehle zur Verfügung:

Befehl	Bedeutung
TMON tool monitoring on	Anwahl der WZ-Überwachung für das aktive Werkzeug im Kanal.
TMOF tool monitoring off	Abwahl der WZ-Überwachung für das aktive Werkzeug im Kanal.
TMON (T-Nummer) tool monitoring on (t-nr)	Anwahl der WZ-Überwachung für ein nicht aktives Werkzeug mit der T-Nummer.
TMOF (T-Nummer) tool monitoring off (t-nr)	Abwahl der WZ-Überwachung für ein nicht aktives Werkzeug mit der T-Nummer.
TMOF (0) tool monitoring off (0)	Abwahl der WZ-Überwachung für alle Werkzeuge.

18.6 Konstante Scheibenumfangsgeschwindigkeit (SUG)

18.6.1 Allgemeines

Was ist SUG?

Für Schleifscheiben wird in der Regel keine Spindeldrehzahl, sondern die Scheibenumfangsgeschwindigkeit programmiert. Diese ist eine Größe, die vom technologischen Prozess bestimmt wird (z. B. Schleifscheibenkenngröße, Materialpaarung). Die Drehzahl wird dann aus dem programmierten Wert und dem aktuellen Scheibenradius ermittelt.

Hinweis

Die SUG kann für Schleifwerkzeuge (Typ 400 - 499) angewählt werden.

Drehzahlberechnung

Die Formel für die Umrechnung lautet:

$$n[min^{-1}] = \frac{SUG\,[m*s^{-1}] * 60}{2\pi * R[m]}$$

Hinweis

Die Scheibenumfangsgeschwindigkeit kann für die Schleifwerkzeugtypen (400 - 499) programmiert und angewählt werden.

Für die Radiusermittlung (Parameter $TC_TPG9) wird der Verschleiß berücksichtigt.

Diese Funktion gilt auch für Schräge Scheiben/Achsen.

Zu dem durch $TC_TPG9 ausgewähltem Parameter wird der jeweilige Verschleiß und in Abhängigkeit vom Werkzeugtyp das Basismaß addiert.

Die Summe wird bei positivem Wert des Parameters $TC_TPG8 (Winkel der schrägen Scheibe) durch "cos" ($TC_TPG8) und bei negativem Wert durch "sin" ($TC_TPG8) dividiert.

Wann wird die Drehzahl neu berechnet?

Die Drehzahl wird beim Auftreten folgender Ereignisse neu berechnet:

- SUG-Programmierung
- Ändern der Online-Korrekturwerte (Verschleißparameter)

18.6.2 An-/Abwahl und Programmierung der SUG, Systemvariable

Teileprogrammbefehle

Die SUG wird mit den folgenden Teileprogrammbefehlen an-/abgewählt:

Befehl	Bedeutung
GWPSON grinding wheel peripheral speed on	Anwahl der SUG für das aktive Werkzeug im Kanal.
GWPSOF grinding wheel peripheral speed off	Abwahl der SUG für das aktive Werkzeug im Kanal.
GWPSON(T-Nummer) grinding wheel peripheral speed on (t-nr)	Anwahl der SUG für ein nicht aktives Werkzeug mit der T-Nummer.
GWPSOF(T-Nummer) grinding wheel peripheral speed off (t-nr)	Abwahl der SUG für ein nicht aktives Werkzeug mit der T-Nummer.
S[Spindel-Nummer] = Wert	Programmierung der konstanten Scheibenumfangsgeschwindigkeit. Einheit der Wertangabe ist abhängig vom Grundsystem (m/s oder ft/s).

Literatur:
/PG/ Programmierhandbuch Grundlagen

Hinweis

Durch den Parameter $TC_TPG1 wird dem Werkzeug eine Spindel zugeordnet. Jeder nachfolgende S-Wert für diese Spindel wird bei aktiver SUG (siehe oben) als Scheibenumfangsgeschwindigkeit interpretiert.

Soll für eine Spindel, für die SUG bereits aktiv ist, SUG mit einem neuen Werkzeug angewählt werden, muss zuerst die aktive SUG mit GWPSOF abgewählt werden (sonst erfolgt ein Alarm).

SUG kann für mehrere Spindeln eines Kanals mit jeweils unterschiedlichen Schleifwerkzeugen gleichzeitig aktiv sein.

Die Anwahl der SUG mit GWPSON führt nicht zur automatischen Aktivierung der Werkzeuglängenkorrektur oder der Geometrie- und Drehzahlüberwachung. Bei der Abwahl der SUG wird die zuletzt ermittelte Drehzahl als Sollwert beibehalten.

$P_GWPS[Spindel-Nummer]

Mit dieser Systemvariablen kann vom Teileprogramm ermittelt werden, ob die SUG für eine bestimmte Spindel aktiv ist.

TRUE:	SUG-Programmierung der Spindel aktiv
FALSE:	SUG-Programmierung der Spindel nicht aktiv

Literatur:
/PG/ Programmierhandbuch Grundlagen

18.6.3 SUG in allen Betriebsarten

Allgemeines

Mit dieser Funktion ist es möglich, die konstante Scheibenumfangsgeschwindigkeit (SUG) für eine Spindel sofort nach `POWER ON` anzuwählen und über Betriebsartenwechsel, `RESET` oder Teileprogrammende angewählt zu lassen.

Aktiviert wird die Funktion über das Maschinendatum:
MD35032 $MA_SPIND_FUNC_RESET_MODE
(Parametrierung der Funktion SUG).

SUG nach POWER ON

Ein schleifspezifisches Werkzeug wird über folgende Maschinendaten definiert:

MD20110 $MC_RESET_MODE_MASK

MD20120 $MC_TOOL_RESET_VALUE

MD20130 $MC_CUTTING_EDGE_RESET_VALUE

Hinweis

MD35032 $MA_SPIND_FUNC_RESET_MODE
Ist das obige Maschinendatum gesetzt und wenn es sich um ein schleifspezifisches Werkzeug (Werkzeug-Typ 400 bis 499, MD20110, MD20120, MD20130) mit Verweis auf eine gültige Spindel (Parameter $TC_TPG1) handelt, dann wird für diese Spindel die SUG aktiviert.

Für alle anderen Spindeln dieses Kanals ist die SUG abgewählt.

SUG nach RESET/Teileprogrammende

Nach RESET/Teileprogrammende bleibt für alle Spindeln, für die SUG angewählt war, die SUG auch weiterhin angewählt.

Hinweis

MD35032 $MA_SPIND_FUNC_RESET_MODE
Ist das oben stehende Maschinendatum gesetzt und ist die SUG aktiv bei RESET bzw. Teileprogrammende, dann bleibt die SUG für diese Spindel aktiv.

Wenn das Maschinendatum:
MD35032 $MA_SPIND_FUNC_RESET_MODE
nicht gesetzt ist und die SUG aktiv ist bei RESET bzw. Teileprogrammende, dann wird die SUG für diese Spindel deaktiviert.

Für alle anderen Spindeln dieses Kanals ist die SUG abgewählt.

Über das Maschinendatum:
MD35040 $MA_SPIND_ACTIVE_AFTER_RESET
kann bestimmt werden, ob die Spindel nach RESET mit der aktuellen Drehzahl weiterdreht.

Programmierung

Die Spindeldrehzahl kann durch Vorgabe einer Scheibenumfangsgeschwindigkeit verändert werden.

Dies ist möglich durch:

- Programmierung im Teileprogramm/Überspeichern
- Programmieren der Scheibenumfangsgeschwindigkeit durch Zuweisung an die Adresse "S" in MDA
- Spindelsteuerung über PLC (FC18)

DB31, ... DBX84.0 (SUG aktiv)

Über das Nahtstellensignal:
DB31, ... DBX84.0 (SUG aktiv)
kann festgestellt werden, ob die SUG aktiv ist oder nicht.

18.6.4 Programmierbeispiel für SUG

Daten von Werkzeug T1 (Umfangsschleifscheibe)

```
$TC_DP1[1,1] = 403              ;Werkzeugtyp
$TC_DP3[1,1] = 300              ;Länge1
$TC_DP4[1,1] = 50               ;Länge2
$TC_DP12[1,1] = 0               ;Verschleiß Länge 1
$TC_DP13[1,1] = 0               ;Verschleiß Länge 2
```

```
$TC_DP21[1,1] = 300              ;Basis Länge 1
$TC_DP22[1,1] = 400              ;Basis Länge 2
$TC_TPG1[1] = 1                  ;Spindelnummer
$TC_TPG8[1] = 0                  ;Winkel der schrägen Scheibe
$TC_TPG9[1] = 3                  ;Parameter-Nr. für Radiusberechnung
```

Daten von Werkzeug T5 (schräge Scheibe)

```
$TC_DP1[5,1] = 401               ;Werkzeugtyp
$TC_DP3[5,1] = 120               ;Länge1
$TC_DP4[5,1] = 30                ;Länge2
$TC_DP12[5,1] = 0                ;Verschleiß Länge 1
$TC_DP13[5,1] = 0                ;Verschleiß Länge 2
$TC_DP21[5,1] = 100              ;Basis Länge 1
$TC_DP22[5,1] = 150              ;Basis Länge 2
$TC_TPG1[5] = 2                  ;Spindelnummer
$TC_TPG8[5] = 45                 ;Winkel der schrägen Scheibe
$TC_TPG9[5] = 3                  ;Parameter-Nr. für Radiusberechnung
```

Programmierung

```
N20     T1 D1                        ;  T1 und D1 anwählen
N25     S1=1000 M1=3                 ;  1000Umdr./min für Spindel 1
N30     S2=1500 M2=3                 ;  1500Umdr./min für Spindel 2
...
N40     GWPSON                       ;  SUG-Anwahl für aktives Werkzeug T1
N45     S[$P_AGT[1]] = 60            ;  SUG für aktives Werkzeug auf 60 m/s setzen
                                     ;  n=1909,85 min-1
...
N50     GWPSON(5)                    ;  SUG-Anwahl für Werkzeug 5 (2. Spindel)
N55     S[$TC_TPG1[5] ] = 40         ;  SUG für Spindel 2 auf 40 m/s setzen
                                     ;  n=1909,85 min-1
...
N60     GWPSOF                       ;  SUG für aktives Werkzeug ausschalten
N65     GWPSOF(5)                    ;  SUG für Werkzeug 5 (Spindel 2) ausschalten
...
```

Ergänzende Literatur

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Pendeln (P5)

/FB1/ Funktionshandbuch Grundfunktionen; Vorschübe (V1)

/FBSY/ Funktionshandbuch Synchronaktionen

18.7 Randbedingungen

18.7.1 Werkzeugwechsel mit Online-Werkzeugkorrektur

Werkzeugwechsel

Ein Werkzeugwechsel mit `M6` in Verbindung mit der Online-Werkzeugkorrektur ist nicht möglich.

18.8 Datenlisten

18.8.1 Maschinendaten

18.8.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: $MN_	Beschreibung
18094	MM_NUM_CC_TDA_PARAM	Anzahl der TDA
18096	MM_NUM_CC_TOA_PARAM	Anzahl der TOA
18100	MM_NUM_CUTTING_EDGES_IN_TOA	Werkzeugkorrekturen pro TOA

18.8.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: $MC_	Beschreibung
20254	ONLINE_CUTCOM_ENABLE	Freigabe Online-Werkzeugradiuskorrektur
20350	TOOL_GRIND_AUTO_TMON	WZ-Überwachung automatisch
20610	ADD_MOVE_ACCEL_RESERVE	Beschleunigungsreserve für überlagerte Bewegungen

18.8.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: $MA_	Beschreibung
32020	JOG_VELO	Konventionelle Achsgeschwindigkeit
35032	SPIND_FUNC_RESET_MODE	Parametrierung der Funktion SUG

18.8.2 Signale

18.8.2.1 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Geometrieüberwachung	DB31, ...DBX83.3	DB390x.DBX2001.3
Drehzahlüberwachung	DB31, ...DBX83.6	DB390x.DBX2001.6
SUG aktiv	DB31, ...DBX84.1	DB390x.DBX2002.1

Z2: NC/PLC-Nahtstellensignale 19

19.1 Digitale und analoge NCK-Peripherie

19.1.1 Signale an NC (DB10)

Übersicht der Signale von PLC an NC

DB10	Signale an NC Nahtstelle PLC → NC							
DBB	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Sperre der digitalen NCK-Eingänge							
	digitale Eingänge ohne Hardware *)				On-Board-Eingänge **)			
	Eingang 8	Eingang 7	Eingang 6	Eingang 5	Eingang 4	Eingang 3	Eingang 2	Eingang 1
1	Setzen von PLC der digitalen NCK-Eingänge							
	digitale Eingänge ohne Hardware *)				On-Board-Eingänge **)			
	Eingang 8	Eingang 7	Eingang 6	Eingang 5	Eingang 4	Eingang 3	Eingang 2	Eingang 1
4	Sperre der digitalen NCK-Ausgänge							
	digitale Ausgänge ohne Hardware *)				On-Board-Ausgänge **)			
	Ausgang 8	Ausgang 7	Ausgang 6	Ausgang 5	Ausgang 4	Ausgang 3	Ausgang 2	Ausgang 1
5	Überschreibmaske der digitalen NCK-Ausgänge							
	digitale Ausgänge ohne Hardware *)				On-Board-Ausgänge **)			
	Ausgang 8	Ausgang 7	Ausgang 6	Ausgang 5	Ausgang 4	Ausgang 3	Ausgang 2	Ausgang 1
6	Setzwert von PLC der digitalen NCK-Ausgänge							
	digitale Ausgänge ohne Hardware *)				On-Board-Ausgänge **)			
	Ausgang 8	Ausgang 7	Ausgang 6	Ausgang 5	Ausgang 4	Ausgang 3	Ausgang 2	Ausgang 1
7	Vorgabemaske der digitalen NCK-Ausgänge							
	digitale Ausgänge ohne Hardware *)				On-Board-Ausgänge **)			
	Ausgang 8	Ausgang 7	Ausgang 6	Ausgang 5	Ausgang 4	Ausgang 3	Ausgang 2	Ausgang 1
Anmerkungen: *) Bit 4 bis 7 der digitalen NCK-Ausgänge können von der PLC bearbeitet werden, obwohl hierfür keine Hardware-Peripherie vorhanden ist. Damit können diese Bits zusätzlich zum Informationsaustausch zwischen NCK und PLC genutzt werden. **) Bei 840D sind die digitalen Ein- und Ausgänge 1 bis 4 der NCK hardwaremäßig On-Board. Diese können entsprechend *) von der PLC bearbeitet werden.								
DB10	Signale an NC Nahtstelle PLC ! NC							
DBB	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
122	Sperre der digitalen NCK-Eingänge							
	Eingang 16	Eingang 15	Eingang 14	Eingang 13	Eingang 12	Eingang 11	Eingang 10	Eingang 9
123	Setzen von PLC der digitalen NCK-Eingänge							

	Eingang 16	Eingang 15	Eingang 14	Eingang 13	Eingang 12	Eingang 11	Eingang 10	Eingang 9
124	Sperre der digitalen NCK-Eingänge							
	Eingang 24	Eingang 23	Eingang 22	Eingang 21	Eingang 20	Eingang 19	Eingang 18	Eingang 17
125	Setzen von PLC der digitalen NCK-Eingänge							
	Eingang 24	Eingang 23	Eingang 22	Eingang 21	Eingang 20	Eingang 19	Eingang 18	Eingang 17
126	Sperre der digitalen NCK-Eingänge							
	Eingang 32	Eingang 31	Eingang 30	Eingang 29	Eingang 28	Eingang 27	Eingang 26	Eingang 25
127	Setzen von PLC der digitalen NCK-Eingänge							
	Eingang 32	Eingang 31	Eingang 30	Eingang 29	Eingang 28	Eingang 27	Eingang 26	Eingang 25
128	Sperre der digitalen NCK-Eingänge							
	Eingang 40	Eingang 39	Eingang 38	Eingang 37	Eingang 36	Eingang 35	Eingang 34	Eingang 33
129	Setzen von PLC der digitalen NCK-Eingänge							
	Eingang 40	Eingang 39	Eingang 38	Eingang 37	Eingang 36	Eingang 35	Eingang 34	Eingang 33
130	Sperre der digitalen NCK-Ausgänge							
	Ausgang 16	Ausgang 15	Ausgang 14	Ausgang 13	Ausgang 12	Ausgang 11	Ausgang 10	Ausgang 9
131	Überschreibmaske der digitalen NCK-Ausgänge							
	Ausgang 16	Ausgang 15	Ausgang 14	Ausgang 13	Ausgang 12	Ausgang 11	Ausgang 10	Ausgang 9
132	Setzwert von PLC der digitalen NCK-Ausgänge							
	Ausgang 16	Ausgang 15	Ausgang 14	Ausgang 13	Ausgang 12	Ausgang 11	Ausgang 10	Ausgang 9
133	Vorgabemaske der digitalen NCK-Ausgänge							
	Ausgang 16	Ausgang 15	Ausgang 14	Ausgang 13	Ausgang 12	Ausgang 11	Ausgang 10	Ausgang 9
134	Sperre der digitalen NCK-Ausgänge							
	Ausgang 24	Ausgang 23	Ausgang 22	Ausgang 21	Ausgang 20	Ausgang 19	Ausgang 18	Ausgang 17
135	Überschreibmaske der digitalen NCK-Ausgänge							
	Ausgang 24	Ausgang 23	Ausgang 22	Ausgang 21	Ausgang 20	Ausgang 19	Ausgang 18	Ausgang 17
136	Setzwert von PLC der digitalen NCK-Ausgänge							
	Ausgang 24	Ausgang 23	Ausgang 22	Ausgang 21	Ausgang 20	Ausgang 19	Ausgang 18	Ausgang 17
137	Vorgabemaske der digitalen NCK-Ausgänge							
	Ausgang 24	Ausgang 23	Ausgang 22	Ausgang 21	Ausgang 20	Ausgang 19	Ausgang 18	Ausgang 17
138	Sperre der digitalen NCK-Ausgänge							
	Ausgang 32	Ausgang 31	Ausgang 30	Ausgang 29	Ausgang 28	Ausgang 27	Ausgang 26	Ausgang 25
139	Überschreibmaske der digitalen NCK-Ausgänge							
	Ausgang 32	Ausgang 31	Ausgang 30	Ausgang 29	Ausgang 28	Ausgang 27	Ausgang 26	Ausgang 25
140	Setzwert von PLC der digitalen NCK-Ausgänge							
	Ausgang 32	Ausgang 31	Ausgang 30	Ausgang 29	Ausgang 28	Ausgang 27	Ausgang 26	Ausgang 25

141	Vorgabemaske der digitalen NCK-Ausgänge							
	Ausgang 32	Ausgang 31	Ausgang 30	Ausgang 29	Ausgang 28	Ausgang 27	Ausgang 26	Ausgang 25
142	Sperre der digitalen NCK-Ausgänge							
	Ausgang 40	Ausgang 39	Ausgang 38	Ausgang 37	Ausgang 36	Ausgang 35	Ausgang 34	Ausgang 33
143	Überschreibmaske der digitalen NCK-Ausgänge							
	Ausgang 40	Ausgang 39	Ausgang 38	Ausgang 37	Ausgang 36	Ausgang 35	Ausgang 34	Ausgang 33
144	Setzwert von PLC der digitalen NCK-Ausgänge							
	Ausgang 40	Ausgang 39	Ausgang 38	Ausgang 37	Ausgang 36	Ausgang 35	Ausgang 34	Ausgang 33
145	Vorgabemaske der digitalen NCK-Ausgänge							
	Ausgang 40	Ausgang 39	Ausgang 38	Ausgang 37	Ausgang 36	Ausgang 35	Ausgang 34	Ausgang 33
146	Sperre der analogen NCK-Eingänge							
	Eingang 8	Eingang 7	Eingang 6	Eingang 5	Eingang 4	Eingang 3	Eingang 2	Eingang 1
147	Vorgabemaske der analogen NCK-Eingänge							
	Eingang 8	Eingang 7	Eingang 6	Eingang 5	Eingang 4	Eingang 3	Eingang 2	Eingang 1
148, 149	Setzwert von PLC für analogen Eingang 1 der NCK							
150, 151	Setzwert von PLC für analogen Eingang 2 der NCK							
152, 153	Setzwert von PLC für analogen Eingang 3 der NCK							
154, 155	Setzwert von PLC für analogen Eingang 4 der NCK							
156, 157	Setzwert von PLC für analogen Eingang 5 der NCK							
158, 159	Setzwert von PLC für analogen Eingang 6 der NCK							
160, 161	Setzwert von PLC für analogen Eingang 7 der NCK							
162, 163	Setzwert von PLC für analogen Eingang 8 der NCK							
166	Überschreibmaske der analogen NCK-Ausgänge							
	Ausgang 8	Ausgang 7	Ausgang 6	Ausgang 5	Ausgang 4	Ausgang 3	Ausgang 2	Ausgang 1
167	Vorgabemaske der analogen NCK-Ausgänge							
	Ausgang 8	Ausgang 7	Ausgang 6	Ausgang 5	Ausgang 4	Ausgang 3	Ausgang 2	Ausgang 1
168	Sperre der analogen NCK-Ausgänge							
	Ausgang 8	Ausgang 7	Ausgang 6	Ausgang 5	Ausgang 4	Ausgang 3	Ausgang 2	Ausgang 1
170, 171	Setzwert von PLC für analogen Ausgang 1 der NCK							
172, 173	Setzwert von PLC für analogen Ausgang 2 der NCK							
174, 175	Setzwert von PLC für analogen Ausgang 3 der NCK							
176, 177	Setzwert von PLC für analogen Ausgang 4 der NCK							
178, 179	Setzwert von PLC für analogen Ausgang 5 der NCK							
180, 181	Setzwert von PLC für analogen Ausgang 6 der NCK							
182, 183	Setzwert von PLC für analogen Ausgang 7 der NCK							
184, 185	Setzwert von PLC für analogen Ausgang 8 der NCK							

Beschreibung der Signale von PLC an NC

DB10 DBB0, 122, 124, 126, 128	**Sperre der digitalen NCK-Eingänge**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der digitale Eingang der NCK wird von der PLC gesperrt. Er wird damit steuerungsintern definiert auf "0" gesetzt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der digitale Eingang der NCK ist freigegeben. Damit kann der am Eingang anliegende Signalzustand im NC-Teileproramm direkt gelesen werden.
Korrespondierend mit ...	DB10 DBB1 (Setzen von PLC der digitalen NCK-Eingänge) DB10 DBB60 (Istwert der digitalen NCK-Eingänge) MD10350 $MN_FASTIO_DIG_NUM_INPUTS

DB10 DBB1, 123, 125, 127, 129	**Setzen von PLC der digitalen NCK-Eingänge**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der digitale NCK-Eingang wird von der PLC definiert auf "1" gesetzt. Damit wird der am Hardware-Eingang anstehende Signalzustand sowie das Sperren des Eingangs (NST "Sperre der digitalen NCK-Eingänge") wirkungslos.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der am NCK-Eingang anliegende Signalzustand ist für Lesezugriffe vom NC-Teileprogramm freigegeben. Voraussetzung ist jedoch, dass der NCK-Eingang nicht von der PLC gesperrt ist (NST "Sperre der digitalen NCK-Eingänge" = 0).
Korrespondierend mit ...	DB10 DBB0 (Sperre der digitalen NCK-Eingänge) DB10 DBB60 (Istwert der digitalen NCK-Eingänge) MD10350 $MN_FASTIO_DIG_NUM_INPUTS

DB10 DBB4, 130, 134, 138, 142	**Sperre der digitalen NCK-Ausgänge**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der digitale Ausgang der NCK ist gesperrt. Am Hardware-Ausgang wird damit definiert "0V" ausgegeben.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der digitale Ausgang der NCK ist freigegeben. Damit wird der vom NC-Teileprogramm bzw. von der PLC vorgegebene Wert an den Hardware-Ausgang ausgegeben.
Korrespondierend mit ...	DB10 DBB5 (Überschreibmaske der digitalen NCK-Ausgänge) DB10 DBB7 (Vorgabemaske der digitalen NCK-Ausgänge) DB10 DBB6 (Setzen von PLC der digitalen NCK-Ausgänge) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS

DB10 DBB5, 131, 135, 139, 143	**Überschreibmaske der digitalen NCK-Ausgänge**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Bei Flankenwechsel von 0 → 1 wird der bisherige NCK-Wert mit dem Setzwert (NST "Setzwert von PLC der digitalen NCK-Ausgänge") überschrieben. Der bisherige NCK-Wert, der beispielsweise vom Teileprogramm direkt gesetzt war, geht dadurch verloren. Der mit dem Setzwert vorgegebene Signalstatus bildet den neuen NCK-Wert.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Da von NCK das Nahtstellensignal nur bei Flankenwechsel von 0 →1 ausgewertet wird, ist es vom PLC-Anwenderprogramm im nächsten PLC-Zyklus wieder auf "0" zurückzusetzen.
Sonderfälle, Fehler, ...	**Hinweis:** Die PLC-Nahtstelle für den Setzwert (DB10 DBB6) wird gemeinsam von der Überschreibmaske (bei Flankenwechsel 0 → 1) und der Vorgabemaske (bei Signalzustand 1) verwendet. Ein zeitgleiches Aktivieren der beiden Masken ist über PLC-Anwenderprogramm zu vermeiden.
Korrespondierend mit ...	DB10 DBB4 (Sperre der digitalen NCK-Ausgänge) DB10 DBB7 (Vorgabemaske der digitalen NCK-Ausgänge) DB10 DBB6 (Setzwert von PLC der digitalen NCK-Ausgänge) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS

DB10 DBB6, 132, 136, 140, 144	**Setzwert von PLC der digitalen NCK-Ausgänge**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit dem Setzwert kann der Signalzustand für den digitalen Hardware-Ausgang durch die PLC verändert werden. Dazu gibt es zwei Möglichkeiten: • Mit der "Überschreibmaske": Mit Flankenwechsel von 0 → 1 bei der "Überschreibmaske" überschreibt die PLC den bisherigen "NCK-Wert" mit dem "Setzwert". Dies ist der neue 'NCK-Wert'. • Mit der 'Vorgabemaske': Mit Signalzustand 1 bei der "Vorgabemaske" wird auf den "PLC-Wert" umgeschaltet. Die Vorgabe erfolgt mit dem "Setzwert". Bei "Setzwert" "1" wird am Hardware-Ausgang 1-Pegel ausgegeben. Bei "0" wird 0-Pegel ausgegeben. Die zugehörigen Spannungswerte sind zu entnehmen: **Literatur:** /PHD/ SINUMERIK 840D Handbuch Projektierung (HW)
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Da von NCK das Nahtstellensignal nur bei Flankenwechsel von 0 → 1 ausgewertet wird, ist es vom PLC-Anwenderprogramm im nächsten PLC-Zyklus wieder auf "0" zurückzusetzen.
Sonderfälle, Fehler, ...	**Hinweis:** Die PLC-Nahtstelle für den Setzwert (DB10 DBB6) wird gemeinsam von der Überschreibmaske (bei Flankenwechsel 0 → 1) und der Vorgabemaske (bei Signalzustand 1) verwendet. Ein zeitgleiches Aktivieren der beiden Masken ist über PLC-Anwenderprogramm zu vermeiden.
Korrespondierend mit ...	DB10 DBB4 (Sperre der digitalen NCK-Ausgänge) DB10 DBB5 (Überschreibmaske der digitalen NCK-Ausgänge) DB10 DBB7 (Vorgabemaske der digitalen NCK-Ausgänge) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS

DB10 DBB7, 133, 137, 141, 145	**Vorgabemaske der digitalen NCK-Ausgänge**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Anstelle des NCK-Wertes wird an den digitalen Hardware-Ausgang der PLC-Wert ausgegeben. Der PLC-Wert ist zuvor im NST "Setzwert von PLC der digitalen NCK-Ausgänge" zu hinterlegen. Der momentane NCK-Wert geht dadurch nicht verloren.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	An den digitalen Hardware-Ausgang wird der NCK-Wert ausgegeben.
Sonderfälle, Fehler, ...	**Hinweis:** Die PLC-Nahtstelle für den Setzwert (DB10 DBB6) wird gemeinsam von der Überschreibmaske (bei Flankenwechsel 0 → 1) und der Vorgabemaske (bei Signalzustand 1) verwendet. Ein zeitgleiches Aktivieren der beiden Masken ist über PLC-Anwenderprogramm zu vermeiden.
Korrespondierend mit ...	DB10 DBB4 (Sperre der digitalen NCK-Ausgänge) DB10 DBB5 (Überschreibmaske der digitalen NCK-Ausgänge) DB10 DBB6 (Setzwert von PLC der digitalen NCK-Ausgänge) MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS

DB10 DBB146	**Sperre der analogen NCK-Eingänge**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der analoge Eingang der NCK wird von der PLC gesperrt. Er wird damit steuerungsintern definiert auf "0" gesetzt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der analoge Eingang der NCK ist freigegeben. Damit kann der am Eingang anliegende Analogwert im NC-Teileprogramm direkt gelesen werden, sofern für diesen NCK-Eingang die Vorgabemaske von PLC auf 0-Signal gesetzt ist.
Korrespondierend mit ...	DB10 DBB147 (Vorgabemaske der analogen NCK-Eingänge) DB10 DBB148 (Setzwert von PLC der analogen NCK-Eingänge) DB10 DBB199 ... (Istwert der analogen NCK-Eingänge) MD10300 $MN_FASTIO_ANA_NUM_INPUTS

DB10 DBB147	**Vorgabemaske der analogen NCK-Eingänge**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Damit wirkt als freigegebener Analogwert der von der PLC vorgegebene Setzwert (NST "Setzwert von PLC der analogen NCK-Eingänge").
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der am NCK-Eingang anliegende Analogwert ist für Lesezugriffe vom NC-Teileprogramm freigegeben. Voraussetzung ist jedoch, dass der NCK-Eingang nicht von der PLC gesperrt ist (NST "Sperre der analogen NCK-Eingänge" = 0).
Korrespondierend mit ...	DB10 DBB146 (Sperre der analogen NCK-Eingänge) DB10 DBB148 bis 163 (Setzwert von PLC der analogen NCK-Eingänge) DB10 DBB199-209 (Istwert der analogen NCK-Eingänge) MD10300 $MN_FASTIO_ANA_NUM_INPUTS

DB10 DBB148 - 163	Setzwert von PLC der analogen NCK-Eingänge
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit dem Setzwert kann ein definierter Analogwert von der PLC vorgegeben werden. Die Auswahl, ob der am HW-Eingang anliegende Analogwert oder der Setzwert von der PLC als freigegebener Analogwert verwendet wird, erfolgt von der PLC mit dem NST "Vorgabemaske der analogen NCK-Eingänge". Sobald das NST "Vorgabemaske" auf 1-Signale gesetzt wird, wirkt der Setzwert von der PLC. Der Setzwert von der PLC wird als Festpunktzahl (16 Bitwert einschließlich Vorzeichen) im 2er-Komplement vorgegeben.
Korrespondierend mit ...	DB10 DBB146 (Sperre der analogen NCK-Eingänge) DB10 DBB147 (Vorgabemaske der analogen NCK-Eingänge) DB10 DBB199-209 (Istwert der analogen NCK-Eingänge) MD10300 $MN_FASTIO_ANA_NUM_INPUTS

DB10 DBB166	Überschreibmaske der analogen NCK-Ausgänge
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Bei Flankenwechsel von 0 → 1 wird der bisherige NCK-Wert mit dem Setzwert (NST "Setzwert von PLC der analogen NCK-Ausgänge") überschrieben. Der bisherige NCK-Wert, der beispielsweise vom Teileprogramm direkt vorgegeben war, geht dadurch verloren. Der mit dem Setzwert von der PLC vorgegebene Analogwert bildet den neuen NCK-Wert.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Da von NCK das Nahtstellensignal nur bei Flankenwechsel von 0 → 1 ausgewertet wird, ist es vom PLC-Anwenderprogramm im nächsten PLC-Zyklus wieder auf "0" zurückzusetzen.
Sonderfälle, Fehler, ...	**Hinweis:** Die PLC-Nahtstelle für den Setzwert wird gemeinsam von der Überschreibmaske (bei Flankenwechsel 0 → 1) und der Vorgabemaske (bei Signalzustand 1) verwendet. Ein zeitgleiches Aktivieren der beiden Masken ist über das PLC-Anwenderprogramm zu vermeiden.
Korrespondierend mit ...	DB10 DBB168 (Sperre der analogen NCK-Ausgänge) DB10 DBB167 (Vorgabemaske der analogen NCK-Ausgänge) DB10 DBB170-185 (Setzwert von PLC der analogen NCK-Ausgänge) MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS

DB10 DBB167	Vorgabemaske der analogen NCK-Ausgänge
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Anstelle des NCK-Wertes wird an den analogen Hardware-Ausgang der PLC-Wert ausgegeben. Der PLC-Wert ist zuvor im NST "Setzwert von PLC der analogen NCK-Ausgänge" zu hinterlegen. Der momentane NCK-Wert geht dadurch nicht verloren.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	An den analogen Hardware-Ausgang wird der NCK-Wert ausgegeben.

DB10 DBB167	**Vorgabemaske der analogen NCK-Ausgänge**
Sonderfälle, Fehler, ...	**Hinweis:** Die PLC-Nahtstelle für den Setzwert wird gemeinsam von der Überschreibmaske (bei Flankenwechsel 0 → 1) und der Vorgabemaske (bei Signalzustand 1) verwendet. Ein zeitgleiches Aktivieren der beiden Masken ist über das PLC-Anwenderprogramm zu vermeiden.
Korrespondierend mit ...	DB10 DBB168 (Sperre der analogen NCK-Ausgänge) DB10 DBB166 (Überschreibmaske der analogen NCK-Ausgänge) DB10 DBB170–185 (Setzwert von PLC der analogen NCK-Ausgänge) MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS

DB10 DBB168	**Sperre der analogen NCK-Ausgänge**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der analoge Ausgang der NCK ist gesperrt. Am Hardware-Ausgang wird damit definiert "0V" ausgegeben.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der analoge Ausgang der NCK ist freigegeben. Damit wird der vom NC-Teileprogramm bzw. von der PLC vorgegebene Wert an den Hardware-Ausgang ausgegeben.
Korrespondierend mit ...	DB10 DBB166 (Überschreibmaske der analogen NCK-Ausgänge) DB10 DBB167 (Vorgabemaske der analogen NCK-Ausgänge) DB10 DBB170-185 (Setzwert von PLC der analogen NCK-Ausgänge) MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS

DB10 DBB170 - 185	**Setzwert von PLC der analogen NCK-Ausgänge**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit dem Setzwert kann der Wert für den analogen Hardwareausgang durch die PLC verändert werden. Dazu gibt es zwei Möglichkeiten: • Mit der "Überschreibmaske": Mit Flankenwechsel von 0 → 1 bei der "Überschreibmaske" überschreibt die PLC den bisherigen "NCK-Wert" mit dem "Setzwert". Dies ist der neue "NCK-Wert". • Mit der "Vorgabemaske": Mit Signalzustand 1 bei der "Vorgabemaske" wird auf den "PLC-Wert" umgeschaltet. Die Vorgabe erfolgt mit dem "Setzwert". Der Setzwert von der PLC wird als Festpunktzahl (16 Bitwert einschließlich Vorzeichen) im 2er-Komplement vorgegeben.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Da von NCK das Nahtstellensignal nur bei Flankenwechsel von 0 → 1 ausgewertet wird, ist es vom PLC-Anwenderprogramm im nächsten PLC-Zyklus wieder auf "0" zurückzusetzen.

DB10 DBB170 - 185	**Setzwert von PLC der analogen NCK-Ausgänge**
Sonderfälle, Fehler, ...	**Hinweis:** Die PLC-Nahtstelle für den Setzwert wird gemeinsam von der Überschreibmaske (bei Flankenwechsel 0 → 1) und der Vorgabemaske (bei Signalzustand 1) verwendet. Ein zeitgleiches Aktivieren der beiden Masken ist über das PLC-Anwenderprogramm zu vermeiden.
Korrespondierend mit ...	DB10 DBB168 (Sperre der analogen NCK-Ausgänge) DB10 DBB166 (Überschreibmaske der analogen NCK-Ausgänge) DB10 DBB167 (Vorgabemaske der analogen NCK-Ausgänge) MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS

19.1.2 Signale von NC (DB10)

Übersicht der Signale von NC an PLC

DB10	**Signale von NC Nahtstelle NC → PLC**							
DBB	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
60	Istwert der digitalen NCK-Eingänge							
					On-Board-Eingänge **)			
					Eingang 4	Eingang 3	Eingang 2	Eingang 1
64	Sollwert der digitalen NCK-Ausgänge							
	digitale Eingänge ohne Hardware *)				On-Board-Ausgänge **)			
	Ausgang 8	Ausgang 7	Ausgang 6	Ausgang 5	Ausgang 4	Ausgang 3	Ausgang 2	Ausgang 1
186	Istwert der digitalen NCK-Eingänge							
	Eingang 16	Eingang 15	Eingang 14	Eingang 13	Eingang 12	Eingang 11	Eingang 10	Eingang 9
187	Istwert der digitalen NCK-Eingänge							
	Eingang 24	Eingang 23	Eingang 22	Eingang 21	Eingang 20	Eingang 19	Eingang 18	Eingang 17
188	Istwert der digitalen NCK-Eingänge							
	Eingang 32	Eingang 31	Eingang 30	Eingang 29	Eingang 28	Eingang 27	Eingang 26	Eingang 25
189	Istwert der digitalen NCK-Eingänge							
	Eingang 40	Eingang 39	Eingang 38	Eingang 37	Eingang 36	Eingang 35	Eingang 34	Eingang 33
190	Sollwert der digitalen NCK-Ausgänge							
	Ausgang 16	Ausgang 15	Ausgang 14	Ausgang 13	Ausgang 12	Ausgang 11	Ausgang 10	Ausgang 9
191	Sollwert der digitalen NCK-Ausgänge							
	Ausgang 24	Ausgang 23	Ausgang 22	Ausgang 21	Ausgang 20	Ausgang 19	Ausgang 18	Ausgang 17
192	Sollwert der digitalen NCK-Ausgänge							

	Ausgang 32	Ausgang 31	Ausgang 30	Ausgang 29	Ausgang 28	Ausgang 27	Ausgang 26	Ausgang 25
193	Sollwert der digitalen NCK-Ausgänge							
	Ausgang 40	Ausgang 39	Ausgang 38	Ausgang 37	Ausgang 36	Ausgang 35	Ausgang 34	Ausgang 33
Anmerkungen: *) Bit 4 bis 7 der digitalen Ein- und NCK-Ausgänge können von der PLC bearbeitet werden, obwohl hierfür keine Hardware-Peripherie vorhanden ist. Damit können diese Bits zusätzlich zum Informationsaustausch zwischen NCK und PLC genutzt werden. **) Bei 840D sind die digitalen Ein- und Ausgänge 1 bis 4 der NCK hardwaremäßig On-Board. Diese können entsprechend *) von der PLC bearbeitet werden.								
DB10	Signale von NC Nahtstelle NC **!** PLC							
DBB	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
194, 195	Istwert des analogen Eingangs 1 der NCK							
196, 197	Istwert des analogen Eingangs 2 der NCK							
198, 199	Istwert des analogen Eingangs 3 der NCK							
200, 201	Istwert des analogen Eingangs 4 der NCK							
202, 203	Istwert des analogen Eingangs 5 der NCK							
204, 205	Istwert des analogen Eingangs 6 der NCK							
206, 207	Istwert des analogen Eingangs 7 der NCK							
208, 209	Istwert des analogen Eingangs 8 der NCK							
210, 211	Sollwert des analogen Ausgangs 1 der NCK							
212, 213	Sollwert des analogen Ausgangs 2 der NCK							
214, 215	Sollwert des analogen Ausgangs 3 der NCK							
216, 217	Sollwert des analogen Ausgangs 4 der NCK							
218, 219	Sollwert des analogen Ausgangs 5 der NCK							
220, 221	Sollwert des analogen Ausgangs 6 der NCK							
222, 223	Sollwert des analogen Ausgangs 7 der NCK							
224, 225	Sollwert des analogen Ausgangs 8 der NCK							

Beschreibung der Signale von NC an PLC

DB10 DBB60, 186 - 189	Istwert der digitalen NCK-Eingänge
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Am digitalen Hardwareeingang der NCK liegt "1-Pegel" an.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Am digitalen Hardwareeingang der NCK liegt "0-Pegel" an.
Sonderfälle, Fehler, ...	Beim Istwert wird der Einfluss des Nahtstellensignals: DB10 DBB0 (Sperre der digitalen NCK-Eingänge) nicht berücksichtigt.
Korrespondierend mit ...	DB10 DBB0 (Sperre der digitalen NCK-Eingänge) MD10350 $MN_FASTIO_DIG_NUM_INPUTS

DB10 DBB64, 190 - 193	Sollwert der digitalen NCK-Ausgänge
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der momentan vorgegebene NCK-Wert für den digitalen Ausgang (Sollwert) ist "1".
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der momentan vorgegebene NCK-Wert für den digitalen Ausgang (Sollwert) ist "0".
Signal irrelevant bei ...	Dieser "Sollwert" wird nur bei folgenden Bedingungen an den Hardware-Ausgang ausgegeben: • Ausgang ist nicht gesperrt (NST "Sperre der digitalen NCK-Ausgänge") • Von der PLC ist auf den NCK-Wert umgeschaltet (NST "Vorgabemaske der digitalen NCK-Ausgänge") Sobald diese Bedingungen erfüllt sind, entspricht der "Sollwert" des digitalen Ausgangs dem "Istwert".
Korrespondierend mit ...	DB10 DBB4 (Sperre der digitalen NCK-Ausgänge) DB10 DBB5 (Überschreibmaske der digitalen NCK-Ausgänge) DB10 DBB6 (Setzwert von PLC der digitalen NCK-Ausgänge) DB10 DBB7 (Vorgabemaske der digitalen NCK-Ausgänge) MD10310 $MN_FASTIO_DIG_NUM_OUTPUTS

DB10 DBB194 - 209	Istwert der analogen NCK-Eingänge
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der am analogen NCK-Eingang anliegende Analogwert wird zur PLC gemeldet. Der Istwert wird von NCK als Festpunktzahl (16 Bitwert einschließlich Vorzeichen) im 2er-Komplement vorgegeben.

DB10 DBB194 - 209	**Istwert der analogen NCK-Eingänge**
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Beim Istwert wird der Einfluss von der PLC auf den Analogwert (z. B. mit NST "Sperre der analogen NCK-Eingänge") nicht berücksichtigt.
Korrespondierend mit ...	DB10 DBB146 (Sperre der analogen NCK-Eingänge) DB10 DBB147 (Vorgabemaske der analogen NCK-Eingänge) DB10 DBB148-163 (Setzwert von PLC der analogen NCK-Eingänge) MD10300 $MN_FASTIO_ANA_NUM_INPUTS

DB10 DBB210 - 225	**Sollwert der analogen NCK-Ausgänge**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der momentan vorgegebene NCK-Wert für den analogen Ausgang (Sollwert) wird zur PLC gemeldet. Der Sollwert wird von NCK als Festpunktzahl (16 Bitwert einschließlich Vorzeichen) im 2er-Komplement vorgegeben.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Dieser "Sollwert" wird nur bei folgenden Bedingungen an den Hardware-Ausgang ausgegeben: • Der Ausgang ist nicht gesperrt (NST "Sperre der analogen NCK-Ausgänge") • Von der PLC ist auf den NCK-Wert umgeschaltet (NST "Vorgabemaske der analogen NCK-Ausgänge")
Korrespondierend mit	DB10 DBB168 (Sperre der analogen NCK-Ausgänge) DB10 DBB166 (Überschreibmaske der analogen NCK-Ausgänge) DB10 DBB170-185 (Setzwert von PLC der analogen NCK-Ausgänge) DB10 DBB167 (Vorgabemaske der analogen NCK-Ausgänge) MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS

19.2 Mehrere Bedientafeln an mehreren NCUs, Dezentrale Systeme

19.2.1 Festgelegte logische Funktionen/Defines

BUSTYP

Name	Wert	Nahtstelle DB19	Bedeutung
MPI	1	DBW100,102,104, 120, 130 Bit 8-15	Bedieneinheit an MPI, 187,5 KBaud
BTSS	2	"	Bedieneinheit an MPI, 1,5 MBaud

STATUS

Name	Wert	Nahtstelle DB19	Bedeutung
OFFL_REQ_PLC	1	Online Schnittstelle 1. : DBB124 2. : DBB134	PLC an Bedieneinheit: PLC möchte Bedieneinheit mit Offline-Request verdrängen.
OFFL_CONF_PLC	2	Online Schnittstelle 1. : DBB124 2. : DBB134	Bedieneinheit an PLC: Quittung auf OFFL_REQ_PLC Die Bedeutung des Signals ist abhängig von Z_INFO DBB125 oder DBB135
OFFL_REQ_OP	3	Online Schnittstelle 1. : DBB124 2. : DBB134	Bedieneinheit an PLC: Bedieneinheit möchte von dieser NCU Offline gehen, stellt den Offline-Request
OFFL_CONF_OP	4	Online Schnittstelle 1. : DBB124 2. : DBB134	PLC an Bedieneinheit: Quittung auf OFFL_REQ_OP Die Bedeutung des Signals ist abhängig von Z_INFO DBB125 oder DBB135
ONL_PERM	5	Anklopf Schnittstelle DBB108	PLC an Bedieneinheit: PLC gibt Bedieneinheit die Mitteilung, ob sie Online gehen kann oder nicht. Die Bedeutung des Signals ist abhängig von Z_INFO: DBB109
S_ACT	6	Online Schnittstelle 1. : DBB124 2. : DBB134	Bedieneinheit an PLC: Bedieneinheit geht Online oder wechselt Bedienfocus. Die Bedeutung des Signals ist abhängig von Z_INFO DBB125 oder DBB135
OFFL_REQ_FOC	7	Online Schnittstelle 1. : DBB124 2. : DBB134	Bedieneinheit an PLC: Bedieneinheit möchte Bedienfocus von dieser NCU wegnehmen
OFFL_CONF_FOC	8	Online Schnittstelle 1. : DBB124 2. : DBB134	PLC an Bedieneinheit: Quittung auf OFFL_REQ_FOC Die Bedeutung des Signals ist abhängig von Z_INFO DBB125 oder DBB135

Name	Wert	Nahtstelle DB19	Bedeutung
ONL_REQ_FOC	9	Online Schnittstelle 1. : DBB124 2. : DBB134	Bedieneinheit an PLC: Bedieneinheit möchte Bedienfocus auf diese NCU setzen
ONL_PERM_FOC	10	Online Schnittstelle 1. : DBB124 2. : DBB134	PLC an Bedieneinheit: Quittung auf ONL_REQ_FOC Die Bedeutung des Signals ist abhängig von Z_INFO DBB125 oder DBB135

Z_INFO

Name	Wert	Nahtstelle DB19	Bedeutung
DISC_FOC	9	DBB125 DBB135	Bedieneinheit wechselt den Bedienfocus auf eine andere NCU.
OK	10	DBB109 Bit 0-3 DBB125 DBB135	positive Quittung
CONNECT	11	DBB125 DBB135	Bedieneinheit ist an dieser NCU Online gegangen.
MMC_LOCKED	13	DBB109 Bit 0-3 DBB125 DBB135	HMI hat Umschaltsperre gesetzt. Auf dieser Bedieneinheit laufen Prozesse, die nicht durch eine Umschaltung unterbrochen werden dürfen.
PLC_LOCKED	14	DBB109 Bit 0-3 DBB125 DBB135	In der HMI-PLC-SS ist die HMI-Umschaltsperre gesetzt. Bedieneinheit kann von dieser NCU nicht Offline gehen oder den Bedienfocus wechseln.
PRIO_H	15	DBB109 Bit 0-3 DBB125 DBB135	An dieser NCU sind Bedieneinheiten mit höherer Priorität Online. Anklopfende Bedieneinheit kann an diese NCU nicht Online gehen.

Die folgenden Kombinationen von STATUS und Z_INFO sind möglich

Name: Status	Z_INFO	Bedeutung
OFFL_REQ_PLC	OK	PLC möchte mit Offline-Request Online-Bedieneinheit verdrängen.
OFFL_CONF_PLC	OK	Bedieneinheit gibt positive Quittung auf Offline-Request von PLC. Bedieneinheit geht anschließend Offline.
OFFL_CONF_PLC	MMC_LOCKED	Bedieneinheit gibt negative Quittung auf Offline-Request. Bedieneinheit geht nicht Offline, weil nicht unterbrechbare Prozesse laufen.
OFFL_REQ_OP	OK	Bedieneinheit möchte von der Online NCU offline gehen, stellt den Offline-Request.
OFFL_CONF_OP	OK	PLC gibt positive Quittung auf Offline-Request. Bedieneinheit geht anschließend offline von dieser NCU.

Name: Status	Z_INFO	Bedeutung
OFFL_CONF_OP	PLC_LOCKED	PLC gibt negative Quittung auf Offline-Request von Bedieneinheit. Anwender hat die HMI-Umschaltsperre gesetzt, Bedieneinheit kann nicht Offline gehen, MMCx_SHIFT_LOCK = TRUE, x=1 oder 2, 1. oder 2. HMI-PLC-Schnittstelle.
ONL_PERM	Nr. der HMI-PLC-Online-Schnittstelle, OK	PLC gibt der anklopfenden Bedieneinheit die Freigabe Online zu gehen. Bedieneinheit kann anschließend an diese NCU Online gehen. Inhalt von Z_INFO: Bit 0 ..3: OK Bit 4 ... 7: Nr. der HMI-PLC-Online-Schnittstelle mit der sich die Bedieneinheit verbinden soll: 1 erste HMI-PLC-Online-Schnittstelle 2 zweite HMI-PLC-Online-Schnittstelle
ONL_PERM	MMC_LOCKED	Die anklopfende Bedieneinheit kann nicht Online gehen. An dieser NCU sind zwei Bedieneinheiten online, auf denen nicht unterbrechbare Prozesse laufen. Die PLC kann keinen der beiden Bedieneinheiten verdrängen
ONL_PERM	PLC_LOCKED	Die anklopfende Bedieneinheit kann nicht online gehen. Anwender hat die HMI-Umschaltsperre gesetzt, MMCx_SHIFT_LOCK = TRUE, x=1 oder 2, 1. oder 2.HMI-Online-Schnittstelle.
ONL_PERM	PRIO_H	Die anklopfende Bedieneinheit kann nicht Online gehen. An dieser NCU sind zwei Bedieneinheiten online, die höherprior als die anklopfende Bedieneinheit sind. Die PLC kann keine der beiden Bedieneinheiten verdrängen.
S_ACT	CONNECT	Die anklopfende MMC ist Online gegangen. Die PLC schaltet jetzt die HMI-Lebenszeichenüberwachung ein.
S_ACT	DISC_FOCUS	Server-HMI hat den Bedienfocus von dieser NCU gelöst.
OFFL_REQ_FOC	OK	Server-HMI möchte den Bedienfocus von dieser NCU lösen, stellt den Offline-Request für den Focus.
OFFL_CONF_FOC	OK	PLC gibt positive Quittung auf Offline-Request für den Focus. Server-HMI kann Bedienfocus lösen.
OFFL_CONF_FOC	PLC_LOCKED	PLC gibt neg. Quittung auf Offline-Request für den Focus. Anwender hat die HMI-Umschaltsperre gesetzt, Server-HMI kann den Bedienfocus nicht lösen, MMCx_SHIFT_LOCK = TRUE, x=1 oder 2, 1. oder 2. HMI-PLC-Schnittstelle.
ONL_REQ_FOC	OK	Server-HMI möchte den Bedienfocus auf diese NCU setzen, stellt den Online-Request für den Focus.
ONL_PERM_FOC	OK	PLC gibt pos. Quittung auf Online-Request für den Focus. Server-HMI schaltet anschließend den Bedienfocus auf diese NCU.
ONL_PERM_FOC	PLC_LOCKED	PLC gibt negative Quittung auf Online-Request für den Focus. Anwender hat die HMI-Umschaltsperre gesetzt, Server-HMI kann den Bedienfocus nicht setzen, MMCx_SHIFT_LOCK = TRUE, x=1 oder 2, 1. oder 2. HMI-PLC-Schnittstelle.

19.2.2 Schnittstellen in DB19 für M:N

Die HMI-PLC Schnittstelle in DB19 gliedert sich in 3 Bereiche.

Anklopf-Schnittstelle

Auf diese Schnittstelle wird die Anklopf-Sequenz ausgeführt, falls eine Bedieneinheit online gehen möchte.

HMI schreibt seine Client-Ident nach ONL_REQUEST und wartet auf die Rückgabe von Client-Ident in

ONL_CONFIRM.

Nach der positiven Quittung von der PLC übergibt die Bedieneinheit ihre Parameter und wartet auf die Online-Permission (in PAR_STATUS, PAR_Z_INFO).

HMI-Parameterübergabe:

Client-Ident -> PAR_CLIENT_IDENT

HMI-Typ -> PAR_MMC_TYP

MSTT-Adresse -> PAR_MSTT_ADR

Mit der positiven Online-Permission wird der Bedieneinheit von der PLC die Nummer seiner HMI-PLC-Online-Schnittstelle DBB109.4-7 übergeben.

Anschließend geht er online und belegt seine ihm zugewiesene Online-Schnittstelle.

Online-Schnittstellen

An einer NCU können zur gleichen Zeit zwei Bedieneinheiten online sein.

Die Online-Schnittstelle ist für jeden der beiden Online-Bedieneinheiten getrennt vorhanden.

Nach erfolgreicher Anklopf-Sequenz wird der Bedieneinheit die Nummer seiner Online-Schnittstelle von der PLC mitgeteilt.

Die HMI-Parameter werden anschließend von der PLC in die entsprechende Online-Schnittstelle übertragen.

Die Bedieneinheit geht online und belegt seine eigene Online-Schnittstelle, über die dann der Datenverkehr zwischen HMI und PLC stattfindet.

HMI-Datenschnittstellen

Hier sind Nutzdaten von/an HMI definiert:

- DBB 0-49 Schnittstelle Bedieneinheit 1
- DBB 50-99 Schnittstelle Bedieneinheit 2

Diese Daten und Signale sind grundsätzlich für den Betrieb von Bedieneinheiten erforderlich.

M:N-Lebenszeichenüberwachung

Es handelt sich um eine zusätzliche Überwachung, die nicht zu verwechseln ist mit der HMI-Lebenszeichenüberwachung. Weitere Hinweise finden Sie bei den entsprechenden Signalen.

Bedieneinheiten mit aktivierter M:N-Umschaltung (parametrierbar in der NETNAMES.INI) müssen in bestimmten Betriebszuständen anhand eines PLC-Datums erkennen können, ob vor der Verbindungsaufnahme angeklopft werden muss oder nicht.

Beispiel:

Bedieneinheit mit aktivierter Bedieneinheiten-Umschaltung muss eine NCU in Betrieb nehmen können ohne vorher anklopfen zu müssen.

Bedieneinheit muss aus Servicegründen online gehen.

Die Koordinierung geschieht in der Anklopf-Schnittstelle über das Datum DBW110: **M_TO_N_ALIVE**

Das M:N-Lebenszeichen ist ein Umlaufzähler, der von der PLC zyklisch erhöht, bzw. bei Überlauf auf den Wert 1 gesetzt wird.

Bevor eine Bedieneinheit anklopft muss sie anhand des Lebenszeichens prüfen, ob die M:N-Umschaltung in der PLC aktiviert ist.

Vorgehensweise:

HMI liest das Lebenszeichen zum Zeitpunkt T0 und T0 + 1.

Fall 1: negative Quittung für den Lesevorgang, DB19 existiert nicht. Bedieneinheit geht ohne Anklopfvorgang online.

Fall 2: m_to_n_alive = 0, Bedieneinheiten-Umschaltung nicht aktiviert. Bedieneinheit geht ohne Anklopfvorgang online.

Fall 3: m_to_n_alive (T0) = m_to_n_alive (T0+1), Bedieneinheiten-Umschaltung nicht aktiviert. Bedieneinheit geht ohne Anklopfvorgang online.

Fall 4: m_to_n_alive (T0) <> m_to_n_alive (T0+1), Bedieneinheiten-Umschaltung aktiviert

Fall 1 ... Fall 3 gilt nur für Sonderfälle und nicht im Normalbetrieb.

Anklopf-Schnittstelle

DB19 DBW100	**ONL_REQUEST**
Client_Ident	Bedieneinheit möchte online gehen und möchte die Anklopf-Schnittstelle belegen. Als Anforderung schreibt HMI zuerst seine Client_Ident. Bit 8 .. 15: Bustyp: MPI 1 oder BTSS 2 Bit 0 .. 7: HMI-Busadresse
DB19 DBW102	**ONL_CONFIRM**
Client_Ident	Falls die Anklopf-Schnittstelle nicht von einer anderen Bedieneinheit belegt ist, gibt die PLC als positive Quittung die Client-Ident zurück. Bit 8 .. 15: Bustyp: MPI 1 oder BTSS 2 Bit 0 .. 7: HMI-Busadresse
DB19 DBW104	**PAR_CLIENT_IDENT** HMI-Parameterübergabe an PLC
Client_Ident	Bit 8 .. 15: Bustyp: MPI 1 oder BTSS 2 Bit 0 .. 7: HMI-Busadresse

DB19 DBB106	**PAR_MMC_TYP** HMI-Parameterübergabe an PLC
HMI-Typ aus NETNAMES.INI	Typeigenschaften der Bedieneinheit, die in der Datei NETNAMES.INI projektiert wurden. Wird von der PLC bei Verdrängung ausgewertet (Server, Haupt-/Nebenbedienfeld, ...), siehe Beschreibung der Datei NETNAMES.INI
DB19 DBB107	**PAR_MSTT_ADR** HMI-Parameterübergabe an PLC
MSTT-Adresse aus NETNAMES.INI	Adresse der MSTT, die zusammen mit der Bedieneinheit umgeschaltet, bzw. de-/aktiviert werden soll. Parameter aus NETNAMES.INI
255	Bedieneinheit ist keine MSTT zugeordnet, es wird keine MSTT de-/aktiviert
DB19 DB108	**PAR_STATUS** PLC gibt HMI die pos./neg. Online-Permission
ONL_PERM (5)	PLC gibt HMI die Mitteilung, ob Bedieneinheit Online gehen kann oder nicht. Die Bedeutung des Signals ist abhängig von PAR_Z_INFO
DB19 DBB109	**PAR_Z_INFO** PLC gibt HMI die pos./neg. Online-Permission
Nr. der HMI-PLC-Online-Schnittstelle, OK (10)	PLC gibt der anklopfenden Bedieneinheit die Freigabe Online zu gehen. Bedieneinheit kann anschließend an diese NCU Online gehen. Bit 0 ..3: OK Bit 4 .. 7: Nr. der HMI-PLC-Online-Schnittstelle mit der sich die Bedieneinheit verbinden soll: 1 erste HMI-PLC-Online-Schnittstelle 2 zweite HMI-PLC-Online-Schnittsteelle
MMC_LOCKED (13)	Die anklopfende Bedieneinheit kann nicht Online gehen. An dieser NCU sind zwei Bedieneinheiten online, auf denen nicht unterbrechbare Prozesse laufen. Die PLC kann keinen der beiden Bedieneinheiten verdrängen.
PLC_LOCKED (14)	In der HMI-PLC-Schnittstelle ist die Bedieneinheiten-Umschaltsperre gesetzt.
PRIO_H (15)	Die anklopfende Bedieneinheit kann nicht Online gehen. An dieser NCU sind zwei Bedieneinheiten online, die höherprior als die anklopfende Bedieneinheit sind. Die PLC kann keine der beiden Bedieneinheiten verdrängen.

Lebenszeichen der M:N-Umschaltung

DB19 DBW110	**M_TO_N_ALIVE**
1 ... 65535	Umlaufzähler, der von der PLC zyklisch erhöht wird. Erkennungsmerkmal für HMI, dass die M:N-Umschaltung aktiv und betriebsbereit ist.

1. HMI-PLC-Online-Schnittstelle

DB19 DBW120	**MMC1_CLIENT_IDENT**
	siehe PAR_CLIENT_IDENT nach positiver Online-Permission überträgt die PLC die HMI-Parameter in die Online-Schnittstelle PAR_CLIENT_IDENT -> MMC1_CLIENT_IDENT
DB19 DBB122	**MMC1_TYP**
	siehe PAR_MMC_TYP Nach positiver Online-Permission überträgt die PLC die HMI-Parameter in die Online-Schnittstelle PAR_MMC_TYP -> MMC1_TYP
DB19 DBB123	**MMC1_MSTT_ADR**

	siehe PAR_ MSTT_ADR Nach positiver Online-Permission überträgt die PLC die HMI-Parameter in die Online-Schnittstelle PAR_ MSTT_ADR -> MMC1_MSTT_ADR
DB19 DBB124	**MMC1_STATUS**
	Anforderungen von Online-HMI an PLC oder umgekehrt. Die Bedeutung des Signals ist abhängig von MMC1_Z_INFO
OFFL_REQ_PLC (1)	PLC an HMI: PLC möchte Bedieneinheit mit Offline-Request verdrängen.
OFFL_CONF_PLC (2)	HMI an PLC: Quittung auf OFFL_REQ_PLC
OFFL_REQ_OP (3)	HMI an PLC: Bedieneinheit möchte von dieser NCU Offline gehen, stellt den Offline-Request
OFFL_CONF_OP (4)	PLC an HMI: Quittung auf OFFL_REQ_OP
S_ACT (6)	HMI an PLC: Bedieneinheit geht Online oder wechselt Bedienfocus
OFFL_REQ_FOC (7)	HMI an PLC: Bedieneinheit möchte Bedienfocus von dieser NCU wegnehmen
OFFL_CONF_FOC (8)	PLC an HMI: Quittung auf OFFL_REQ_FOC
ONL_REQ_FOC (9)	HMI an PLC: Bedieneinheit möchte Bedienfocus auf diese NCU setzen
ONL_PERM_FOC (10)	PLC an HMI: Quittung auf ONL_REQ_FOC
DB19 DBB125	**MMC1_Z_INFO**
	Anforderungen von Online-HMI an PLC oder umgekehrt Die Bedeutung des Signals ist abhängig von MMC1_STATUS
DISC_FOC (9)	Bedieneinheit wechselt den Bedienfocus auf eine andere NCU
OK (10)	positive Quittung
CONNECT (11)	Bedieneinheit ist an diese NCU Online gegangen
PLC_LOCKED (14)	In der HMI-PLC-Schnittstelle ist die Bedieneinheiten-Umschaltsperre gesetzt. Bedieneinheit kann von dieser NCU nicht Offline gehen oder den Bedienfocus wechseln
PRIO_H (15)	An dieser NCU sind Bedieneinheiten mit höherer Prio Online. Anklopfende Bedieneinheit kann an diese NCU nicht Online gehen

Bitsignale

DB19 DBX 126.0 Datenbaustein	**MMC1_SHIFT_LOCK Bedieneinheiten-Umschaltung sperren/freigeben**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Bedieneinheiten-Umschaltung oder Wechsel des Bedienfocus ist gesperrt. Der aktuelle Bedieneinheit-NCU-Verbindungszustand bleibt erhalten.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Bedieneinheiten-Umschaltung oder Wechsel des Bedienfocus ist freigegeben

DB19 DBX 126.1 Datenbaustein	**MMC1_MSTT_SHIFT_LOCK MSTT-Umschaltung sperren/freigeben**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	MSTT-Umschaltung ist gesperrt. Die aktuelle MSTT-NCU-Konstellation bleibt erhalten.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	MSTT-Umschaltung ist freigegeben

DB19 DBX 126.2 Datenbaustein	**MMC1_ACTIVE_REQ Bedieneinheit 1 fordert aktiven Bedienmodus**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	HMI an PLC: passive Bedieneinheit 1 fordert aktiven Bedienmodus
Signalzustand 0 bzw. Flankenwechsel 1 → 0	PLC an HMI: Anforderung erhalten

DB19 DBX 126.3 Datenbaustein	**MMC1_ACTIVE_PERM Aktiver/passiver Bedienmodus**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	PLC an HMI: passive Bedieneinheit 1 kann in aktiven Bedienmodus wechseln
Signalzustand 0 bzw. Flankenwechsel 1 → 0	PLC an MMC: aktive Bedieneinheit soll in passiven Bedienmodus wechseln

DB19 DBX 126.4 Datenbaustein	**MMC1_ACTIVE_CHANGED Aktiver/passiver Bedienmodus von MMC**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	HMI an PLC: Bedieneinheit hat Wechsel von Passiv- nach Aktiv-Modus vollzogen
Signalzustand 0 bzw. Flankenwechsel 1 → 0	HMI an PLC: Bedieneinheit hat Wechsel von Aktiv- nach Passiv-Modus vollzogen

DB19 DBX126.5 Datenbaustein	MMC1_CHANGE_DENIED Bedienmoduswechsel abgelehnt
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	HMI an PLC oder PLC an HMI, je nach Zustand der Schnittstelle: Bedienmoduswechsel nicht möglich, nicht unterbrechbare Prozesse auf aktiver Bedieneinheit
Signalzustand 0 bzw. Flankenwechsel 1 → 0	HMI an PLC oder PLC an HMI, je nach Zustand der Schnittstelle: Quittung auf MMC1_CHANGE_DENIED (FALSE → TRUE)

2. HMI-PLC-Online-Schnittstelle

Die Signale der 2. HMI-PLC-Online-Schnittstelle sind in ihrer Bedeutung analog zu den Signalen der 1. HMI-PLC-Online-Schnittstelle (MMC2_ ... tritt anstelle MMC1_...).

HMI-Lebenszeichenüberwachung

Nachdem eine Bedieneinheit an einer NCU online gegangen ist, wird in der Nahtstelle das HMI-Lebenszeichen gesetzt. (E_BTSSReady, E_MMCMPI_Ready, E_MMC2Ready)

Die Signale werden von HMI automatisch gesetzt, wenn die Bedieneinheit online geht und bleiben solange gesetzt, wie sie online ist.

Sie sind für jede HMI-PLC-Schnittstelle getrennt vorhanden und werden von der PLC zur HMI-Lebenszeichenüberwachung benützt.

1. HMI-PLC-Online-Schnittstelle

Hier wird unterschieden, ob die Bedieneinheit über BTSS (1.5 MBaud) oder MPI (187.5 kBaud) online ist.

Es wird das dem Bustyp entsprechende Signal gesetzt, solange die Bedieneinheit online ist.

DB**10** DBX104.0	MSTT1 bereit
FALSE	MSTT1 ist nicht bereit
TRUE	MSTT1 ist bereit
DB**10** DBX104.1	MSTT2 bereit
FALSE	MSTT2 ist nicht bereit
TRUE	MSTT2 ist bereit
DB**10** DBX104.2	BHG bereit
FALSE	BHG ist nicht bereit
TRUE	BHG ist bereit
DB**10** DBX108.3	E_MMCBTSSReady
FALSE	keine Bedieneinheit online an BTSS
TRUE	Bedieneinheit online an BTSS
DB**10** DBX108.2	E_MMCMPIReady
FALSE	keine Bedieneinheit online an MPI
TRUE	Bedieneinheit online an MPI

2. HMI-PLC-Online-Schnittstelle

Für diese Schnittstelle existiert ein Sammelsignal für beide Bustypen. Es wird nicht zwischen BTSS oder MPI unterschieden.

DB**10** DBX108.1	E_MMC2Ready
FALSE	keine Bedieneinheit online an BTSS oder MPI
TRUE	Bedieneinheit online an BTSS oder MPI

Die Lebenszeichenüberwachung wird von der PLC eingeschaltet, sobald eine Bedieneinheit an seine Schnittstelle online gegangen ist und wieder eingestellt, sobald sie offline geht.

Lebenszeichenüberwachung wird **eingeschaltet:**
sobald Bedieneinheit, bzw. HMI sich mit S_ACT/CONNECT in seiner HMI-PLC-Schnittstelle online meldet.

Lebenszeichenüberwachung wird **abgeschaltet:**
sobald Bedieneinheit offline geht.

1. HMI will umschalten und meldet sich bei der PLC mit

   OFFL_REQ_OP/OK ab
2. PLC quittiert der HMI mit OFFL_CONF_OP/OK
3. Bedieneinheit, bzw. HMI wird durch die PLC mit OFFL_REQ_PLC/OK verdrängt

   HMI quittiert der PLC mit OFFL_CONF_PLC/OK

In beiden Fällen erkennt die PLC, dass eine Bedieneinheit offline geht und wartet auf die TRUE-FALSE Flanke seines HMI-Lebenszeichens.

Anschließend wird dieses Lebenszeichen nicht mehr überwacht.

19.2.3 Signale von NC (DB10)

DB10 DBX107.6	**NCU-Link aktiv**
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	NCU-Link-Kommunikation ist aktiv.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Keine NCU-Link-Kommunikation ist aktiv.
Signal irrelevant bei ...	Systemen ohne NCU-Linkmodule.
Weiterführende Literatur	/PHD/ Handbuch Projektierung NCU

19.2.4 Signale von Achse/Spindel (DB31, ...)

DB31, ... DBX60.1	NCU-Link Achse aktiv
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Achse ist als NCU-Link-Achse aktiv.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Achse ist als lokale Achse benutzt.
Signal irrelevant bei ...	Systemen ohne NCU-Linkmodule.
weiterführende Literatur	/PHD/ Handbuch Projektierung NCU

DB31, ... DBX61.1	Achscontainer-Drehung aktiv
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Für die Achse ist eine Achscontainer-Drehung aktiv.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Für die Achse ist keine Achscontainer-Drehung aktiv.

DB31, ... DBX61.2	Achse betriebsbereit
Flankenauswertung:	Signal(e) aktualisiert:
Bedeutung	Das Signal wird auf der Heimat-NCU im NCU-Linkverband geführt. Heimat-NCU ist die NCU, auf der die Achse physikalisch angeschlossen ist.
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Achse ist betriebsbereit.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Achse ist nicht betriebsbereit. Dieser Zustand wird eingestellt, wenn der Kanal, die Betriebsartengruppe oder der NCK den Alarm "Nicht bereit" erzeugt haben.

19.3 Bedienung über PG/PC (B4)

Keine Signalbeschreibungen erforderlich.

19.4 Handfahren und Handradfahren

19.4.1 Signale von NC (DB10)

<table>
<tr><td>DB10
DBB97, 98, 99</td><td colspan="9">Kanalnummer Geometrieachse für Handrad 1, 2, 3</td></tr>
<tr><td>Flankenauswertung: nein</td><td colspan="9">Signal(e) aktualisiert: zyklisch</td></tr>
<tr><td>Signalbedeutung</td><td colspan="9">Der Bediener kann direkt an der Bedientafelfront dem Handrad (1, 2, 3) eine Achse zuordnen. Falls diese Achse eine Geometrieachse ist (NST "Maschinenachse" = 0), wird für das jeweilige Handrad die zugeordnete Kanalnummer zur PLC übertragen.
Damit wird vom PLC-Grundprogramm für die angewählte Geometrieachse das NST "Handrad aktivieren" entsprechend den vom Bediener vorgegebenen Zustand (NST "Handrad angewählt") gesetzt.
Folgende Codierung gilt für die Kanalnummer:
<table>
<tr><th colspan="8">Bit</th><th rowspan="2">Kanalnummer</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
</table>
Bei Maschinenachsen (NST "Maschinenachse" = 1) ist das NST "Kanalnummer Geometrieachse für Handrad 1, 2, 3" ohne Bedeutung.
Weitere Informationen siehe NST "Achsnummer für Handrad 1, 2, 3".</td></tr>
<tr><td>Korrespondierend mit ...</td><td colspan="9">DB10 DBB100 ff (Achsnummer für Handrad 1, 2, 3)
DB10 DBX100.6 ff (Handrad angewählt)
DB10 DBX100.7 ff (Maschinenachse)
DB21, ... DBX12.0 - 12.2 ff (Handrad aktivieren)</td></tr>
<tr><td>Anwendungs-beispiel(e)</td><td colspan="9">Wenn DB10 DBB97 = 2, dann ist das Handrad 1 dem Kanal 2 zugeordnet.</td></tr>
</table>

<table>
<tr><td>DB10
DBB100, 101, 102
Bit 0 - 4</td><td colspan="2">Achsnummer für Handrad 1, 2 oder 3</td></tr>
<tr><td>Flankenauswertung: nein</td><td colspan="2">Signal(e) aktualisiert: zyklisch</td></tr>
<tr><td>Signalbedeutung</td><td colspan="2">Der Bediener kann direkt an der Bedientafelfront jedem Handrad eine Achse zuordnen. Dazu gibt er die gewünschte Achse (z. B. X) vor. Vom PLC–Grundprogramm wird die der Achse zugehörige Achsnummer zzgl. der Information "Maschinen- oder Geometrieachse" (NST "Maschinenachse") als HMI-Nahtstellensignale zur Verfügung gestellt. Damit wird vom PLC-Grundprogramm für die vorgegebene Achse das Nahtstellensignal "Handrad aktivieren" gesetzt. Abhängig von dem HMI-Nahtstellensignal "Maschinenachse" wird dabei die Nahtstelle zur Geometrieachse bzw. zur Maschinenachse verwendet.
Bei der Zuordnung der Achsbezeichnung zur Achsnummer gilt folgendes:
NST "Maschinenachse" = 1; d. h. Maschinenachse:
Die Zuordnung erfolgt über das Maschinendatum:
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[n] (Maschinenachsname).
NST "Maschinenachse" = 0; d. h. Geometrieachse:
Die Zuordnung erfolgt über das Maschinendatum:
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[n] (Geometrieachsname im Kanal)
Mit dem NST "Kanalnummer Geometrieachse Handrad n" wird die dem Handrad zugeordnete Kanalnummer vorgegeben.
Folgende Codierung gilt für die Achsnummer:</td></tr>
<tr><td></td><td colspan="2">
<table>
<tr><th colspan="5">Bit</th><th>Achsnummer</th></tr>
<tr><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>–</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>8</td></tr>
</table>
</td></tr>
<tr><td>Korrespondierend mit ...</td><td colspan="2">DB10 DBX97 ff (Kanalnummer Geometrie-Achse Handrad n)
DB10 DBX100.6 ff (Handrad angewählt)
DB10 DBX100.7 ff (Maschinenachse)
DB21, ... DBX12.0 bis DBX12.2 ff (Handrad aktivieren)
DB31, ... DBX4.0 bis DBX4.2 (Handrad aktivieren)
MD10000 $MN_AXCONF_MACHAX_NAME_TAB [n] (Maschinenachsname)
MD20060 $MC_AXCONF_GEOAX_NAME_TAB [n] (Geometrieachsname im Kanal)</td></tr>
</table>

DB10 DBX100.6, 101.6, 102.6	**Handrad angewählt (für Handrad 1, 2 oder 3)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der Bediener hat an der Bedientafelfront das Handrad für die vorgegebene Achse angewählt (d. h. aktiv gesetzt). Diese Information wird vom PLC-Grundprogramm an der HMI-Nahtstelle zur Verfügung gestellt. Damit wird vom PLC-Grundprogramm für die vorgegebene Achse das Nahtstellensignal: DB21, ... DBX12.0-12.2 ff (Handrad aktivieren) auf "1" gesetzt. Die zugehörige Achse wird ebenfalls an der HMI-Nahtstelle angezeigt: DB10 DBX100.7 ff (Maschinenachse) und DB10 DBB100 ff (Achsnummer für Handrad 1). Sobald das Handrad aktiv ist, kann die Achse bei JOG mit dem Handrad verfahren werden (DB21, ... DBX40.0-40.2 ff (Handrad aktiv).
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der Bediener hat an der Bedientafelfront das Handrad für die vorgegebene Achse inaktiv gesetzt. Diese Information wird vom PLC-Grundprogramm an der HMI-Nahtstelle zur Verfügung gestellt. Damit kann vom PLC-Grundprogramm für die vorgegebene Achse das Nahtstellensignal: DB21, ... DBX12.0-12.2 ff (Handrad aktivieren) auf "0" gesetzt werden.
Korrespondierend mit ...	DB10 DBB100 ff (Achsnummer) DB10 DBX100.7 ff (Maschinenachse) DB21, ... DBX12.0-12.2 ff (Handrad aktivieren) DB21, ... DBX40.0 - DBX40.2 ff (Handrad aktiv) DB31, ... DBX4.0 - DBX4.2 (Handrad aktivieren) DB10 DBB97 ff (Kanalnummer Geometrieachse für Handrad 1, 2 oder 3)

DB10 DBX100.7, 101.7, 102.7	**Maschinenachse (für Handrad 1, 2 oder 3)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der Bediener hat direkt an der Bedientafelfront dem Handrad (1, 2, 3) eine Achse zugeordnet. Diese Achse ist eine Maschinenachse. Weitere Informationen siehe NST "Achsnummer".
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der Bediener hat direkt an der Bedientafelfront dem Handrad (1, 2, 3) eine Achse zugeordnet. Diese Achse ist eine Geometrieachse. Weitere Informationen siehe NST "Achsnummer".
Korrespondierend mit	DB10 DBB100 ff (Achsnummer) DB10 DBX100.6 ff (Handrad angewählt) DB10 DBB97 ff (Kanalnummer Geometrieachse für Handrad 1, 2 oder 3)

19.4.2 Signale an Kanal (DB21, ...)

Übersicht der Signale an Kanal zum NCK

DB21, ...	Signale an Kanal							
DBB	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0		Probelauf-vorschub aktivieren	M01 aktivieren	Einzelsatz aktivieren	DRF aktivieren	Vorwärts-fahren aktivieren	Rückwärtsf ahren aktivieren	Ab-arbeiten von extern
12	Geometrieachse 1							
	Verfahrtasten		Eilgang-über-lagerung	Verfahr-tasten-sperre	Vorschub-Halt	Handrad aktivieren		
	+	-				3	2	1
13	Geometrieachse 1 Maschinenfunktion							
		kontinuier-lich	variabel INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
16	Geometrieachse 2							
	Verfahrtasten		Eilgang-über-lagerung	Verfahr-tasten-sperre	Vorschub-Halt	Handrad aktivieren		
	+	-				3	2	1
17	Geometrieachse 2 Maschinenfunktion							
		kontinuier-lich	variabel INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
20	Geometrieachse 3							
	Verfahrtasten		Eilgang-über-lagerung	Verfahr-tasten-sperre	Vorschub-Halt	Handrad aktivieren		
	+	-				3	2	1
21	Geometrieachse 3 Maschinenfunktion							
		kontinuier-lich	variabel INC	10000 INC	1000 INC	100 INC	10 INC	1 INC

Beschreibung der Signale an Kanal zum NCK

DB21, ... DBX0.3	**DRF aktivieren**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Funktion DRF wird angewählt. Die Anwahl kann beispielsweise direkt vom PLC-Anwenderprogramm erfolgen oder von der Bedientafelfront über das HMI-Nahtstellensignal: DB21, ... DBX24.3 (DRF angewählt) Dieses HMI-Nahtstellensignal wird entweder vom PLC-Grundprogramm oder vom PLC-Anwenderprogramm auf das Nahtstellensignal: DB21, … DBX0.3 (DRF aktivieren) umgesetzt. Sobald die Funktion DRF aktiv ist, kann mit dem Handrad in der Betriebsart AUTOMATIK bzw. MDA die DRF-Verschiebung verändert werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Funktion DRF wird nicht angewählt.
Signal irrelevant bei ...	Betriebsart JOG
Anwendungs- beispiel(e)	Mit dem Nahtstellensignal: DB21, … DBX0.3 (DRF aktivieren) kann vom PLC-Anwenderprogramm gezielt die DRF-Funktion freigegeben werden.
Korrespondierend mit ...	DB21, ... DBX24.3 (DRF angewählt)

DB21, ... DBB12, 16, 20 Bit 0-2	**Handrad aktivieren (1 bis 3) für Geometrieachse (1, 2, 3)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit diesen PLC-Nahtstellensignalen wird festgelegt, ob diese Geometrieachse dem Handrad 1, 2 oder 3 bzw. keinem Handrad zugeordnet ist. Zu einem Zeitpunkt kann einer Achse jeweils nur ein Handrad zugeordnet werden. Sind mehrere Nahtstellensignale: DB21, … DBX12.1-3, DBX16.1-3, DBX20.1-3 (Handrad aktivieren) gesetzt, so gilt die Priorität "Handrad 1" vor "Handrad 2" vor "Handrad 3". **Hinweis:** Über Handrad 1 bis 3 können 3 Geometrieachsen simultan verfahren werden!
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Dieser Achse ist Handrad 1, 2 oder 3 nicht zugeordnet.
Anwendungs- beispiel(e)	Mit dem Nahtstellensignal kann vom PLC-Anwenderprogramm die Beeinflussung der Geometrieachse durch Verdrehung eines Handrades verriegelt werden.
Korrespondierend mit ...	DB21, ... DBX40.7 bzw. DBX40.6 ff (Handrad aktiv für Geometrieachse)

DB21, ... DBX12.4, 16.4, 20.4	**Verfahrtastensperre für Geometrieachse (1, 2, 3)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Verfahrtasten plus und minus sind für die entsprechende Geometrieachse ohne Wirkung. Somit ist beispielsweise ein Verfahren der Geometrieachse in JOG über die Verfahrtasten der MSTT nicht möglich. Wird die Verfahrtastensperre während einer Verfahrbewegung aktiviert, so wird die Geometrieachse stillgesetzt.
Signalzustand 0	Die Verfahrtasten plus und minus sind freigegeben.
Anwendungs- beispiel(e)	Damit kann vom PLC-Anwenderprogramm, abhängig vom Betriebszustand, ein Verfahren der Geometrieachse in JOG über die Verfahrtasten verriegelt werden.
Korrespondierend mit ...	DB21, ... DBX12.7 bzw DBX12.6 ff (Verfahrtaste plus bzw. Verfahrtaste minus für Geometrieachse)

DB21, ... DBX12.5, 16.5, 20.5	**Eingangüberlagerung für Geometrieachse (1, 2, 3)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Wird zusammen mit Nahtstellensignal: DB21, ... DBX12.7 bzw DBX12.6 ff (Verfahrtaste Plus bzw. Verfahrtaste Minus) das PLC-Nahtstellensignal: DB21, … DBB12.5, 16.5 bzw. 20.5 (Eilgangüberlagerung an Geometrieachsen) gegeben, so verfährt die angesprochene Geometrieachse mit Eilgang. Die Eilganggeschwindigkeit ist mit dem Maschinendatum: MD32010 $MA_JOG_VELO_RAPID (Konventioneller Eilgang) festgelegt. Die Eilgangüberlagerung ist bei folgenden Varianten in der Betriebsart JOG wirksam: • Kontinuierliches Verfahren • Inkrementelles Verfahren Bei wirksamer Eilgangüberlagerung ist die Geschwindigkeit mit dem Eilgang-Korrekturschalter beeinflussbar.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Geometrieachse verfährt mit der vorgegebenen JOG-Geschwindigkeit: SD41110 $SN_JOG_SET_VELO (Achsgeschwindigkeit bei JOG) oder MD32020 $MA_JOG_VELO (Konventionelle Achsgeschwindigkeit).
Signal irrelevant bei ...	• Betriebsart AUTOMATIK und MDA • Referenzpunktfahren (Betriebsart JOG)
Korrespondierend mit ...	DB21, ... DBX12.7 bzw. DBX12.6 ff (Verfahrtaste plus und Verfahrtaste minus für Geometrieachse)
Weiterführende Literatur	/FB1/ Funktionshandbuch Grundfunktion; Vorschübe (V1)

DB21, ... DBB12, 16, 20 Bit 7, 6	**Verfahrtasten plus und minus für Geometrieachse (1, 2, 3)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	In der Betriebsart JOG kann mit den Verfahrtasten plus und minus die angewählte Geometrieachse in beiden Richtungen verfahren werden. Abhängig von der aktiven Maschinenfunktion sowie von der Einstellung "Tipp- oder Dauerbetrieb": SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD (Tipp-/ Dauerbetr. bei JOG kontinuierlich) und MD11300 $MN_JOG_INC_MODE_LEVELTRIGGRD (INC und REF im Tippbetrieb) werden bei Signalwechsel unterschiedliche Reaktionen ausgelöst. **Fall 1: Kontinuierliches Verfahren mit Tippbetrieb** Die Geometrieachse verfährt solange in die jeweilige Richtung wie das Nahtstellensignal 1-Signalzustand hat (sofern die Achsposition keine gültige Begrenzung erreicht hat). **Fall 2: Kontinuierliches Verfahren mit Dauerbetrieb** Beim ersten Flankenwechsel 0 → 1 startet die Geometriechse die Verfahrbewegung in die entsprechende Richtung. Diese Verfahrbewegung wird auch bei einem Flankenwechsel 1 → 0 weiter fortgesetzt. Mit einem erneuten Flankenwechsel 0 → 1 (gleiche Verfahrrichtung!) wird die Verfahrbewegung wieder beendet. **Fall 3: Inkrementelles Verfahren mit Tippbetrieb** Mit Signalzustand 1 beginnt die Geometrieachse das eingestellte Inkrement zu verfahren. Wechselt das Signal auf Zustand 0 bevor das Inkrement abgefahren wurde, so wird die Verfahrbewegung unterbrochen. Mit erneutem Signalzustand 1 wird die Verfahrbewegung wieder fortgesetzt. Bis das Inkrement vollständig abgefahren ist, kann die Verfahrbewegung der Geometrieachse mehrfach wie oben beschrieben gestoppt und fortgesetzt werden. **Fall 4: Inkrementelles Verfahren mit Dauerbetrieb** Beim ersten Flankenwechsel 0 → 1 beginnt die Geometrieachse das eingestellte Inkrement zu verfahren. Erfolgt bei dem gleichen Verfahrsignal ein erneuter Flankenwechsel 0 → 1, bevor die Geometrieachse das Inkrement abgefahren hat, so wird die Verfahrbewegung abgebrochen. Das Inkrement wird somit nicht mehr zu Ende gefahren. Werden beide Verfahrsignale (plus und minus) gleichzeitig gesetzt, so erfolgt keine Verfahrbewegung bzw. wird die Verfahrbewegung abgebrochen! Mit dem PLC-Nahtstellensignal "Verfahrtastensperre" kann einzeln für jede Geometrieachse die Wirkung der Verfahrtasten gesperrt werden. **Achtung!** Im Gegensatz zu Maschinenachsen kann bei Geometrieachsen über Verfahrtasten simultan jeweils nur eine Geometrieachse verfahren werden. Beim Versuch, mehr als eine Geometrieachse über Verfahrtasten zu verfahren, wird der Alarm 20062 gemeldet.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Siehe oben Fall 1 bis 4.
Signal irrelevant bei ...	Betriebsart AUTOMATIK und MDA
Sonderfälle, Fehler, ...	Die Geometrieachse kann im JOG nicht verfahren werden: • falls sie bereits über die axiale PLC-Nahtstelle (als Maschinenachse) verfahren wird. • falls eine andere Geometrieachse bereits über Verfahrtasten verfahren wird. Es wird der Alarm 20062 "Achse ist bereits aktiv" gemeldet.
Korrespondierend mit ...	DB31, ... DBX8.7 bzw. DBX8.6 (Verfahrtasten plus und minus für Maschinenachsen) DB21, ... DBX12.4 ff (Verfahrtastensperre für Geometrieachsen)

DB21, ... DBB13, 17, 21 Bit 0-5	**Maschinenfunktion für Geometrieachse (1, 2, 3) INC1, INC10, INC100, INC1000, INC10000, INCvar**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit diesen Nahtstellensignalen wird festgelegt, wie viel Inkremente bei Betätigung der Verfahrtaste oder bei Verdrehung des Handrades je Rasterstellung die Geometrieachse verfährt. Dabei muss die Betriebsart JOG aktiv sein (Ausnahme bei DRF). Die den einzelnen Nahtstellensignalen zugehörige Inkrementgröße wird wie folgt festgelegt: • Bei INC1 bis INC10000 Mit dem allgemeinen Maschinendatum: MD11330 $MN_JOG_INCR_SIZE_TAB (Inkrementgröße bei INC/Handrad) • Bei INCvar Mit dem allgemeinen Settingdatum: SD41010 $SN_JOG_VAR_INCR_SIZE (Größe des variablen Inkrements bei JOG) Sobald die angewählte Maschinenfunktion wirksam ist, wird dies an die PLC-Nahtstelle gemeldet: DB21, ... DBB41 ff (aktive Maschinenfunktion INC1; ...) Werden an der Nahtstelle gleichzeitig mehrere Maschinenfunktionen-Signale (INC1, INC... oder Nahtstellensignal "Kontinuierliches Verfahren" angewählt, so wird steuerungsintern keine Maschinenfunktion aktiv gesetzt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Entsprechende Maschinenfunktion ist nicht angewählt. Fährt gerade eine Achse ein Schrittmaß ab, so wird mit Abwahl oder Umschaltung der Maschinenfunktion auch die Bewegung abgebrochen.
Korrespondierend mit ...	DB21, ... DBB41 ff (aktive Maschinenfunktion INC1, ...) für Geometrieachsen DB21, ... DBX13 ff (Maschinenfunktion kontinuierlich) für Geometrieachsen

DB21, ... DBX13.6, 17.6, 21.6	**Maschinenfunktion kontinuierlich für Geometrieachse (1, 2, 3)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenfunktion "Kontinuierliches Verfahren" ist angewählt. In der Betriebsart JOG kann die zugehörige Geometrieachse mit den Verfahrtasten Plus und Minus verfahren werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenfunktion "Kontinuierliches Verfahren" ist nicht angewählt.
Korrespondierend mit ...	DB21, ... DBB41 ff (aktive Maschinenfunktion INC 1, ..., kontinuierlich) DB21, ... DBB13 ff (Maschinenfunktion INC1, ..., INC10000)

19.4.3 Signale von Kanal (DB21, ...)

Übersicht der Signale von Kanal zum PLC

DB21, ...	Signale von Kanal							
DBB	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
24 (HMI → PLC)		Probelauf-vorschub angewählt	M01 angewählt	Einzelsatz angewählt	DRF angewählt			
33 (HMI → PLC)					Handrad-über-lagerung aktiv	Handrad aktivieren		
37	Stoppen am Blockende bei SBL unterdrückt	Einlese-freigabe wird ignoriert	CLC gestoppt Ober-grenze /TE1/	CLC gestoppt Unter-grenze /TE1/	CLC aktiv /TE1/	Konturhandrad aktiv		
						Handrad 1	Handrad 2	Handrad 3
	Geometrieachse 1							
40	Fahrbefehl		Fahranforderung			Handrad aktiv		
	plus	minus	plus	minus		3	2	1
41		Geometrieachse 1 Aktive Maschinenfunktion						
		kontinuier-lich	variabel INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
	Geometrieachse 2							
46	Fahrbefehl		Fahranforderung			Handrad aktiv		
	plus	minus	plus	minus		3	2	1
47		Geometrieachse 2 Aktive Maschinenfunktion						
		kontinuier-lich	variabel INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
	Geometrieachse 3							
52	Fahrbefehl		Fahranforderung			Handrad aktiv		
	plus	minus	plus	minus		3	2	1
53		Geometrieachse 3 Aktive Maschinenfunktion						
		kontinuier-lich	variabel INC	10000 INC	1000 INC	100 INC	10 INC	1 INC

Beschreibung der Signale von Kanal zum PLC

DB21, ... DBX24.3	DRF angewählt	
Flankenauswertung: nein		Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der Bediener hat DRF an der Bedientafelfront angewählt. Das PLC-Programm (PLC-Grundprogramm oder Anwenderprogramm) überträgt dieses HMI-Nahtstellensignal nach entsprechender Verknüpfung zum Nahtstellensignal: DB21, … DBB0.3 (DRF aktivieren). Sobald DRF aktiv ist, kann die DRF-Verschiebung während der Betriebsart AUTOMATIK oder MDA über das der Achse zugeordnete Handrad verändert werden.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der Bediener hat DRF an der Bedientafelfront nicht angewählt.	
Signal irrelevant bei ...	Betriebsart JOG	
Korrespondierend mit ...	DB21, ... DBX0.3 (DRF aktivieren)	

DB21, ... DBX33.3	Handradüberlagerung aktiv	
Flankenauswertung: nein		Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Funktion "Handradüberlagerung in Automatik" ist für die programmierten Bahnachsen aktiv. Handradimpulse der 1.Geometrieachse wirken als Geschwindigkeitsüberlagerung auf den programmierten Bahnvorschub.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Funktion "Handradüberlagerung in Automatik" ist für die programmierten Bahnachsen nicht aktiv. Eine aktive Handradüberlagerung wird unwirksam, wenn: • die Bahnachsen die programmierte Zielposition erreicht haben. • der Restweg durch das kanalspezifische Nahtstellensignal: DB21, ... DBX6.2 (Restweg löschen) gelöscht wird. • RESET betätigt wird.	

DB21, ... DBB37 Bit 0-2	Konturhandrad aktiv (1 bis 3)
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit diesen PLC-Nahtstellensignalen wird mitgeteilt, ob diese Geometrieachse dem Konturhandrad 1, 2 oder 3 bzw. keinem Konturhandrad zugeordnet ist. Zu einem Zeitpunkt kann einer Achse jeweils nur ein Konturhandrad zugeordnet werden. Sind mehrere Nahtstellensignale: DB21, … DBB30.0-3 (Konturhandrad aktivieren) gesetzt, gilt die Priorität "Konturhandrad 1" vor "Konturhandrad 2" vor "Konturhandrad 3". Ist die Zuordnung aktiv, kann die Geometrieachse mit dem Konturhandrad in der Betriebsart JOG verfahren werden oder in der Betriebsart AUTOMATIK bzw. MDA eine DRF-Verschiebung erzeugt werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Dieser Geometrieachse ist Konturhandrad 1, 2 oder 3 nicht zugeordnet.

DB21, ... DBB40, 46, 52 Bit 0-2	Handrad aktiv (1 bis 3) für Geometrieachse
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit diesen PLC-Nahtstellensignalen wird mitgeteilt, ob diese Geometrieachse dem Handrad 1, 2 oder 3 bzw. keinem Handrad zugeordnet ist. Zu einem Zeitpunkt kann einer Achse jeweils nur ein Handrad zugeordnet werden. Sind mehrere Nahtstellensignale: DB21, … DBX12.1-3; 16.1-3; 20.1-3 (Handrad aktivieren) gesetzt, so gilt die Priorität "Handrad 1" vor "Handrad 2" vor "Handrad 3". Ist die Zuordnung aktiv, so kann die Geometrieachse mit dem Handrad in der Betriebsart JOG verfahren werden oder in der Betriebsart AUTOMATIK bzw. MDA eine DRF-Verschiebung erzeugt werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Dieser Geometrieachse ist Handrad 1, 2 oder 3 nicht zugeordnet.
Korrespondierend mit ...	DB21, ... DBX12.0-12.2 ff (Handrad aktivieren)

DB21, ... DBB40, 46, 52 Bit 5, 4	Fahranforderung plus und minus (für Geometrieachse)
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Das Signal entspricht dem bisherigen Signal Fahrbefehl.	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	• Betriebsart JOG: Mit Verfahrtaste plus bzw. minus. • Betriebsart REF: Mit Verfahrtaste, die zum Referenzpunkt hinführt. Betriebsart AUT/MDA: Ein Programmsatz, der einen Koordinatenwert für die betreffende Achse enthält, wird ausgeführt.

DB21, ... DBB40, 46, 52 Bit 5, 4	**Fahranforderung plus und minus (für Geometrieachse)**
Signalzustand 0 bzw. Flankenwechsel 1 → 0	In der betreffenden Achsrichtung steht momentan keine Fahranforderung an bzw. ist eine erfolgte Verfahrbewegung beendet. • Betriebsart JOG: Der Fahrbefehl wird in Abhängigkeit von der Einstellung "Tipp- oder Dauerbetrieb" zurückgesetzt (DB21, … DBB12.6-7; 16.6-7; 20.6-7). Beim Verfahren mit Handrad. • Betriebsart REF: Mit Erreichen des Referenzpunktes. Betriebsart AUT/MDA: Der Programmsatz ist abgearbeitet (und der nachfolgende Programmsatz enthält keinen Koordinatenwert für die betreffende Achse). Abbruch durch "RESET", etc. NST DB21, … DBX25.7 (Achsensperre) steht an.
Korrespondierend mit ...	DB21, ... DBX40.7 bzw. DBX40.6 DB21, ... DBX46.7 bzw. DBX46.6 DB21, ... DBX52.7 bzw. DBX52.6 (Fahrbefehl plus" und "Fahrbefehl minus)

DB21, ... DBB40, 46, 52 Bit 7, 6	**Fahrbefehl plus und minus (für Geometrieachse)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Das Signal wirkt wie beschrieben, wenn das Bit 0 im Maschinendatum: MD17900 $MN_VDI_FUNCTION_MASK (Einstellung zu VDI-Signalen) gleich 0 gesetzt ist. Ist das Bit 0 im MD auf 1 gesetzt, so zeigt das Signal nur dann 1, wenn die Achse tatsächlich fährt. Das immer ausgegebene Nahtstellensignal: DB21, ... DBX 40, 46, 52 Bit 5, 4 (Fahranforderung plus/minus) verhält sich wie Signal Fahrbefehl plus/minus bei MD17900 Bit 0 = 0.	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	In der betreffenden Achsrichtung soll eine Fahrbewegung erfolgen. Der Fahrbefehl wird entsprechend der Betriebsart auf unterschiedliche Weise ausgelöst. • Betriebsart JOG: Mit Verfahrtaste plus bzw. minus. • Betriebsart REF: Mit Verfahrtaste, die zum Referenzpunkt hinführt. • Betriebsart AUT/MDA: Ein Programmsatz, der einen Koordinatenwert für die betreffende Achse enthält, wird ausgeführt.

DB21, ... DBB40, 46, 52 Bit 7, 6	**Fahrbefehl plus und minus (für Geometrieachse)**
Signalzustand 0 bzw. Flankenwechsel 1 → 0	In der betreffenden Achsrichtung steht momentan keine Fahranforderung an bzw. ist eine erfolgte Verfahrbewegung beendet. • Betriebsart JOG: Der Fahrbefehl wird in Abhängigkeit von der Einstellung "Tipp- oder Dauerbetrieb" zurückgesetzt (siehe DB21, ... DBX12.7 bzw. DBX12.6 ff). Beim Verfahren mit Handrad. • Betriebsart REF: Mit Erreichen des Referenzpunktes. • Betriebsart AUT/MDA: Der Programmsatz ist abgearbeitet (und der nachfolgende Programmsatz enthält keinen Koordinatenwert für die betreffende Achse). Abbruch durch "RESET", etc. NST DB21, … DBX25.7 (Achsensperre) steht an.
Anwendungs- beispiel(e)	Lösen der Klemmung bei Achsen mit Klemmung (z. B. bei Rundtischen). **Hinweis:** Wird die Klemmung erst mit dem Fahrbefehl gelöst, so ist bei diesen Achsen kein Bahnbetrieb möglich!
Korrespondierend mit ...	DB21, ... DBX12.7 bzw. DBX12.6 ff (Verfahrtaste plus und Verfahrtaste minus für Geometrieachse) DB21, ... DBX 40, 46, 52 Bit 5 (Fahranforderung plus/minus)

DB21, ... DBB41, 47, 53 Bit 0-6	**Aktive Maschinenfunktion für Geometrieachse (1, 2, 3) INC1, ..., kontinuierlich**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	An die PLC-Nahtstelle wird zurückgemeldet, welche Maschinenfunktion in der Betriebsart JOG für die Geometrieachsen wirksam ist. Abhängig von der aktiven Maschinenfunktion ist die Reaktion bei Betätigung der Verfahrtaste oder bei Verdrehung des Handrades unterschiedlich.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Entsprechende Maschinenfunktion ist nicht aktiv.
Korrespondierend mit ...	DB21, ... DBB13 ff (Maschinenfunktion INC1, ..., kontinuierlich für Geometrieachse)

DB21, ... DBB332, 336, 340 Bit 5, 4	**Fahranforderung plus und minus (für Orientierungsachse)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Das Signal entspricht dem bisherigen Signal Fahrbefehl.	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	• Betriebsart JOG: Mit Verfahrtaste plus bzw. minus. • Betriebsart REF: Mit Verfahrtaste, die zum Referenzpunkt hinführt. • Betriebsart AUT/MDA: Ein Programmsatz, der einen Koordinatenwert für die betreffende Achse enthält, wird ausgeführt.

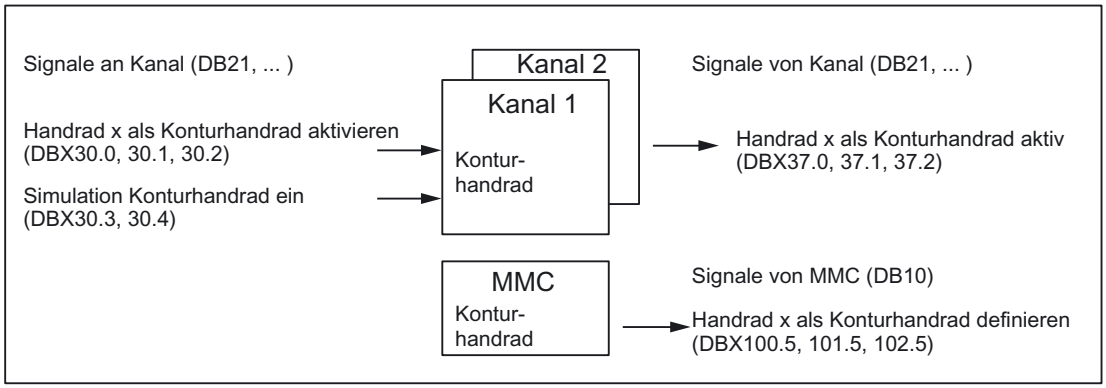
DB21, ... DBB332, 336, 340 Bit 5, 4	**Fahranforderung plus und minus (für Orientierungsachse)**
Signalzustand 0 bzw. Flankenwechsel 1 → 0	In der betreffenden Achsrichtung steht momentan keine Fahranforderung an bzw. ist eine erfolgte Verfahrbewegung beendet. • Betriebsart JOG: Der Fahrbefehl wird in Abhängigkeit von der Einstellung "Tipp- oder Dauerbetrieb" zurückgesetzt (siehe DBX12.7 bzw. DBX12.6 ff. Beim Verfahren mit Handrad. • Betriebsart REF: Mit Erreichen des Referenzpunktes. • Betriebsart AUT/MDA: Der Programmsatz ist abgearbeitet (und der nachfolgende Programmsatz enthält keinen Koordinatenwert für die betreffende Achse). Abbruch durch "RESET", etc. Nahtstellensignal DB21, … DBX25.7 (Achsensperre) steht an.
Korrespondierend mit ...	DB31, ... DBX332.7 bzw. DBX332.6 DB31, ... DBX336.7 bzw. DBX336.6 DB31, ... DBX340.7 bzw. DBX340.6 (Verfahrbefehl plus und Verfahrbefehl minus)

DB21, ... DBB332, 336, 340 Bit 7, 6	**Fahrbefehl plus und minus (für Orientierungsachse)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Das Signal wirkt wie beschrieben, wenn das Bit 0 im Maschinendatum: MD17900 $MN_VDI_FUNCTION_MASK (Einstellung zu VDI-Signalen) gleich 0 gesetzt ist. Ist das Bit 0 im MD auf 1 gesetzt, so zeigt das Signal nur dann 1, wenn die Achse tatsächlich fährt. Das immer ausgegebene Nahtstellensignal: DB21, ... DBX 332, 336, 340 Bit 5, 4 (Fahranforderung plus/minus) verhält sich wie Signal Fahrbefehl plus/minus bei MD17900 Bit 0 = 0.	
Signalzustand 1 bzw. Flankenwechsel 0 → 1	In der betreffenden Achsrichtung soll eine Fahrbewegung erfolgen. Der Fahrbefehl wird entsprechend der Betriebsart auf unterschiedliche Weise ausgelöst: • Betriebsart JOG: Mit Verfahrtaste plus bzw. minus. • Betriebsart REF: Mit Verfahrtaste, die zum Referenzpunkt hinführt. • Betriebsart AUT/MDA: Ein Programmsatz, der einen Koordinatenwert für die betreffende Achse enthält, wird ausgeführt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	In der betreffenden Achsrichtung steht momentan keine Fahranforderung an bzw. ist eine erfolgte Verfahrbewegung beendet. • Betriebsart JOG: Der Fahrbefehl wird in Abhängigkeit von der Einstellung "Tipp- oder Dauerbetrieb" zurückgesetzt (siehe DB21, ... DBX12.7 bzw. DBX12.6 ff). Beim Verfahren mit Handrad. • Betriebsart REF: Mit Erreichen des Referenzpunktes. • Betriebsart AUT/MDA: Der Programmsatz ist abgearbeitet (und der nachfolgende Programmsatz enthält keinen Koordinatenwert für die betreffende Achse). Abbruch durch "RESET", etc. Nahtstellensignal DB21, … DBX25.7 (Achsensperre) steht an.

DB21, ... DBB332, 336, 340 Bit 7, 6	Fahrbefehl plus und minus (für Orientierungsachse)
Anwendungs-beispiel(e)	Lösen der Klemmung bei Achsen mit Klemmung (z. B. bei Rundtischen). Hinweis: Wird die Klemmung erst mit dem Fahrbefehl gelöst, so ist bei diesen Achsen kein Bahnbetrieb möglich!
Korrespondierend mit ...	DB21, ... DBX12.7 bzw. DBX12.6 ff (Verfahrtaste plus und Verfahrtaste minus für Geometrieachse) DB21, ... DBX 332, 336, 340 Bit 5, 4 (Fahranforderung plus/minus)

19.4.4 Signale bei Konturhandrad

Übersicht der Nahtstellensignale bei Konturhandrad



Beschreibung der Nahtstellensignale bei Konturhandrad

DB21, ... DBX30.0 DBX30.1 DBX30.2	Handrad 1 als Konturhandrad aktivieren Handrad 2 als Konturhandrad aktivieren Handrad 3 als Konturhandrad aktivieren	
Flankenauswertung: nein		Signal(e) aktualisiert: zyklisch
Beschreibung	Über diese Signale kann eines der drei Handräder als Konturhandrad an-/abgewählt werden:	
	Signal = 1	Das Handrad x ist als Konturhandrad angewählt
	Signal = 0	Das Handrad x ist als Konturhandrad abgewählt

DB21, ... DBX30.0 DBX30.1 DBX30.2	**Handrad 1 als Konturhandrad aktivieren Handrad 2 als Konturhandrad aktivieren Handrad 3 als Konturhandrad aktivieren**
	Das Ein-/Ausschalten des Konturhandrads kann mitten in einem Satz erfolgen. Beim Einschalten wird die Bewegung zunächst abgebremst und dann entsprechend des Konturhandrads verfahren. Beim Ausschalten wird die Bewegung abgebremst und das NC-Programm unmittelbar fortgesetzt. Soll das NC-Programm erst nach erneutem NC-START fortgesetzt werden, so muss das Ausschalten des Konturhandrads im PLC-Anwenderprogramm mit einem NC-STOP verknüpft werden.
Sonderfälle, Fehler, ...	Das Signal bleibt über NC-RESET hinweg erhalten.
Korrespondierend mit ...	DB21, ... DBX37.0, 37.1, 37.2 (Handrad x als Konturhandrad aktiv)

DB21, ... DBX30.3 DBX30.4	**Simulation Konturhandrad ein Negative Richtung Simulation Konturhandrad**		
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch		
Beschreibung	Zum Ein-/Ausschalten der Simulation des Konturhandrads und zur Vorgabe der Verfahrrichtung sind diese Signale wie folgt zu setzen:		
	Bit 3	**Bit 4**	**Bedeutung**
	0	0	Simulation aus
	0	1	Simulation aus
	1	0	Simulation ein, Richtung wie programmiert
	1	1	Simulation ein, Richtung entgegen der Programmierung
	Bei der Simulation wird der Vorschub nicht mehr vom Konturhandrad vorgegeben, sondern es wird mit dem programmierten Vorschub auf der Kontur verfahren. Wird die Funktion abgewählt, so wird die laufende Bewegung mit Bremsrampe abgebremst. Bei einem Umschalten der Verfahrrichtung wird die laufende Bewegung mit Bremsrampe abgebremst und dann in Gegenrichtung verfahren.		
Sonderfälle, Fehler, ...	Die Simulation wirkt nur in der Betriebsart AUTOMATIK und kann nur eingeschaltet werden, wenn das Konturhandrad aktiviert ist.		

DB21, ... DBX37.0 DBX37.1 DBX37.2	**Handrad 1 als Konturhandrad aktiv Handrad 2 als Konturhandrad aktiv Handrad 3 als Konturhandrad aktiv**	
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch	
Beschreibung	Über diese Signale wird angezeigt, welches Handrad als Konturhandrad angewählt ist:	
	Signal = 1	Das Handrad x ist als Konturhandrad angewählt.
	Signal = 0	Das Handrad x ist als Konturhandrad abgewählt.
Sonderfälle, Fehler, ...	Das Signal bleibt über NC-RESET hinweg erhalten.	
Korrespondierend mit ...	DB21, ... DBX30.0, 30.1, 30.2 (Handrad x als Konturhandrad aktivieren)	

DB10 DBX100.5 DBX101.5 DBX102.5	**Handrad 1 als Konturhandrad definieren Handrad 2 als Konturhandrad definieren Handrad 3 als Konturhandrad definieren**	
Flankenauswertung: nein		Signal(e) aktualisiert: zyklisch
Beschreibung	Über diese Signale wird angezeigt, welches Handrad über MMC als Konturhandrad definiert ist:	
	Signal = 1	Das Handrad x ist über MMC als Konturhandrad definiert.
	Signal = 0	Das Handrad x ist nicht als Konturhandrad definiert.
	Damit das über MMC definierte Handrad auch als Konturhandrad wirksam wird, muss das entsprechende Signal auf Nahtstellensignal: DB21, ... DBX30.0, 30.1, 30.2 (Handrad x als Konturhandrad aktivieren) verknüpft werden.	
Sonderfälle, Fehler, ...	Ahhängig vom Parameter HWheelMMC im FB1 des PLC-Grundprogramms werden diese Signale vom Grundprogramm versorgt oder müssen vom PLC-Anwenderprogramm versorgt werden.	
Korrespondierend mit ...	DB21 ... DBX30.0, 30.1, 30.2 (Handrad x als Konturhandrad aktivieren) FB1 Parameter HWheelMMC	

19.4.5 Signale an Achse/Spindel (DB31, ...)

Übersicht der Signale an Achse/Spindel

DB31, ...	**Signale an Achse/Spindel**							
DBB	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Verfahrtasten		Eilgang-überlagerung	Verfahrtasten-sperre	Vorschub-Halt Spindel-Halt	Handrad aktivieren		
	plus	minus				3	2	1
5	Maschinenfunktion							
		kontinuierlich	variabel INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
13						JOG - Festpunkt anfahren		

Beschreibung der Signale an Achse/Spindel

DB31, ... DBB4 Bit 0-2	**Handrad aktivieren (1 bis 3)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit diesen PLC-Nahtstellensignalen wird festgelegt, ob diese Maschinenachse dem Handrad 1, 2 oder 3 bzw. keinem Handrad zugeordnet ist. Zu einem Zeitpunkt kann einer Achse jeweils nur ein Handrad zugeordnet werden. Sind mehrere Nahtstellensignale: DB31, ... DBX4.0, 4.1, 4.2 (Handrad aktivieren) gesetzt, so gilt die Priorität "Handrad 1" vor "Handrad 2"' vor "Handrad 3". Ist die Zuordnung aktiv, so kann die Maschinenachse mit dem Handrad in der Betriebsart JOG verfahren werden oder in der Betriebsart AUTOMATIK bzw. MDA eine DRF-Verschiebung erzeugt werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Dieser Maschinenachse ist Handrad 1, 2 oder 3 nicht zugeordnet.
Anwendungs-beispiel(e)	Mit dem Nahtstellensignal kann vom PLC-Anwenderprogramm die Beeinflussung der Achse durch Verdrehung eines Handrades verriegelt werden.
Korrespondierend mit ...	DB31, ... DBX64.0 bis DBX64.2 (Handrad aktiv)

DB31, ... DBX4.4	**Verfahrtastensperre**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Verfahrtasten plus und minus sind für die entsprechende Maschinenachse ohne Wirkung. Somit ist beispielsweise ein Verfahren der Achse in JOG über die Verfahrtasten der MSTT nicht möglich. Wird die Verfahrtastensperre während einer Verfahrbewegung aktiviert, so wird die Maschinenachse stillgesetzt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Verfahrtasten plus und minus sind freigegeben.
Anwendungs-beispiel(e)	Damit kann vom PLC-Anwenderprogramm abhängig vom Betriebszustand ein Verfahren der Maschinenachse in JOG über die Verfahrtasten verriegelt werden.
Korrespondierend mit ...	DB31, ... DBX4.7 bzw. DBX4.6 (Verfahrtaste plus und Verfahrtaste minus)

DB31, ... DBX4.5	**Eilgangüberlagerung**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Wird zusammen mit dem Nahtstellensignal: DB31, ... DBX4.0, 4.1, 4.2 (Verfahrtaste Plus bzw. Verfahrtaste Minus) das PLC-Nahtstellensignal DB31, ... DBX4.5 (Eilgangüberlagerung) gegeben, so verfährt die angesprochene Maschinenachse mit Eilgang. Die Eilganggeschwindigkeit ist mit dem Maschinendatum: MD32010 $MA_JOG_VELO_RAPID (Konventioneller Eilgang) festgelegt. Die Eilgangüberlagerung ist bei folgenden Varianten in der Betriebsart JOG wirksam: • Kontinuierlichen Verfahren • Inkrementellen Verfahren Bei wirksamer Eilgangüberlagerung ist die Geschwindigkeit mit dem axialen Vorschubkorrekturschalter beeinflussbar.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenachse verfährt mit der vorgegebenen JOG-Geschwindigkeit: SD41110 $SN_JOG_SET_VELO (Achsgeschwindigkeit bei JOG) oder MD32020 $MA_JOG_VELO (Konventionelle Achsgeschwindigkeit).
Signal irrelevant bei ...	Betriebsart AUTOMATIK und MDA Referenzpunktfahren (Betriebsart JOG)
Korrespondierend mit ...	DB31, ... DBX4.7 bzw. DBX4.6 (Verfahrtaste plus und Verfahrtaste minus) DB31, ... DBB0 (axiale Vorschub-/Spindelkorrektur)

DB31, ... DBB4 Bit 7, 6	**Verfahrtasten plus und minus**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	In der Betriebsart JOG kann mit den Verfahrtasten plus und minus die angewählte Maschinenachse in beiden Richtungen verfahren werden. Abhängig von der aktiven Maschinenfunktion sowie von der Einstellung von Settingdatum: SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD (Tipp-/ Dauerbetrieb bei JOG kontinuierlich) bei JOG-kontinuierlich bzw. In Maschinendatum: MD11300 $MN_JOG_INC_MODE_LEVELTRIGGRD bei JOG-INCR (INC und REF im Tippbetrieb), werden bei Signalwechsel unterschiedliche Reaktionen ausgelöst. **Fall 1: Kontinuierliches Verfahren mit Tippbetrieb** Die Maschinenachse verfährt solange in die jeweilige Richtung wie das Nahtstellensignal 1-Signalzustand hat (sofern die Achsposition keine gültige Begrenzung erreicht hat). **Fall 2:Kontinuierliches Verfahren mit Dauerbetrieb** Beim ersten Flankenwechsel 0 → 1 startet die Maschinenachse die Verfahrbewegung in die entsprechende Richtung. Diese Verfahrbewegung wird auch bei einem Flankenwechsel 1 → 0 weiter fortgesetzt. Mit einem erneuten Flankenwechsel 0 → 1 (gleiche Verfahrrichtung!) wird die Verfahrbewegung wieder beendet. **Fall 3:Inkrementelles Verfahren mit Tippbetrieb** Mit Signalzustand 1 beginnt die Maschinenachse das eingestellte Inkrement zu verfahren. Wechselt das Signal auf Zustand 0 bevor das Inkrement abgefahren wurde, so wird die Verfahrbewegung unterbrochen. Mit erneutem Signalzustand 1 wird die Verfahrbewegung wieder fortgesetzt. Bis das Inkrement vollständig abgefahren ist, kann die Verfahrbewegung der Achse mehrfach wie oben beschrieben gestoppt und fortgesetzt werden. **Fall 4: Inkrementelles Verfahren mit Dauerbetrieb** Beim ersten Flankenwechsel 0 → 1 beginnt die Maschinenachse das eingestellte Inkrement zu verfahren. Erfolgt bei dem gleichen Verfahrsignal ein erneuter Flankenwechsel 0 → 1, bevor die Achse das Inkrement abgefahren hat, so wird die Verfahrbewegung abgebrochen. Das Inkrement wird somit nicht mehr zu Ende gefahren. Werden beide Verfahrsignale (plus und minus) gleichzeitig gesetzt, so erfolgt keine Verfahrbewegung bzw. wird die Verfahrbewegung abgebrochen. Mit dem PLC-Nahtstellensignal: DB31, ... DBX4.4 (Verfahrtastensperre) kann axial die Wirkung der Verfahrtasten gesperrt werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Siehe oben Fall 1 bis 4.
Signal irrelevant bei ...	Betriebsart AUTOMATIK und MDA.
Anwendungs-beispiel(e)	Die Maschinenachse kann im JOG nicht verfahren werden, falls sie bereits über die kanalspezifische PLC-Nahtstelle (als Geometrieachse) verfahren wird. Es wird der Alarm 20062 gemeldet.
Sonderfälle, ...	Teilungsachsen
Korrespondierend mit ...	DB21, ... DBX12.7, DBX12.6 ff (Verfahrtasten plus und minus für Geometrieachsen) DB31, ... DBX4.4 (Verfahrtastensperre)

DB31, ... DBB5 Bit 0-5	Maschinenfunktion INC1, INC10, INC100, INC1000, INC10000, INCvar
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit diesen Nahtstellensignalen wird festgelegt, wie viel Inkremente bei Betätigung der Verfahrtaste oder bei Verdrehung des Handrades je Rasterstellung die Maschinenachse verfährt. Dabei muss die Betriebsart JOG aktiv sein (Ausnahme bei DRF). Die den einzelnen Nahtstellensignalen zugehörige Inkrementgröße wird wie folgt festgelegt: • Bei INC1 bis INC10000: Mit dem allgemeinen Maschinendatum: MD11330 $MN_JOG_INCR_SIZE_TAB (Inkrementgröße bei INC/Handrad) • Bei INCvar: Mit dem allgemeinem Settingdatum: SD41010 $SN_JOG_VAR_INCR_SIZE (Größe des variablen Inkrements bei JOG) Sobald die angewählte Maschinenfunktion wirksam ist, wird dies an die PLC-Nahtstelle gemeldet: DB31, ... DBB65 (aktive Maschinenfunktion INC1; ...) Werden an der Nahtstelle gleichzeitig mehrere Maschinenfunktionen-Signale (INC1, INC... oder DB31, ... DBX5.6) angewählt, so wird steuerungsintern keine Maschinenfunktion aktiv gesetzt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Entsprechende Maschinenfunktion ist nicht angewählt. Fährt gerade eine Achse ein Schrittmaß ab, so wird mit Abwahl oder Umschaltung der Maschinenfunktion auch die Bewegung abgebrochen.
Korrespondierend mit ...	DB31, ... DBB65 (aktive Maschinenfunktion INC1, ...) DB31, ... DBX5.6 (Maschinenfunktion kontinuierlich)

DB31, ... DBX5.6	Maschinenfunktion kontinuierlich
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenfunktion: DB31, ... DBX5.6 (kontinuierliches Verfahren) ist angewählt. In der Betriebsart JOG kann die zugehörige Maschinenachse mit den Verfahrtasten Plus und Minus verfahren werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenfunktion: DB31, ... DBX5.6 (kontinuierliches Verfahren) ist nicht angewählt.
Korrespondierend mit ...	DB31, ... DBB65 (aktive Maschinenfunktion INC 1, ..., kontinuierlich) DB31, ... DBB5 (Maschinenfunktion INC1, ..., INC10000)

DB31, ... DBB13 Bit 0-2	**JOG - Festpunkt anfahren 0/1/2**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Aktiviert die Funktion "Festpunkt anfahren in JOG". Die Nummer des anzufahrenden Festpunkts wird durch Bit 0-2 binärcodiert ausgegeben. Sobald die Funktion aktiv ist (siehe DB31, ... DBX75.0-2), kann die angewählte Maschinenachse mit den Verfahrtasten oder dem Handrad auf den entsprechenden Festpunkt gefahren werden. Die zur Verfügung stehenden Festpunkte (max. 4 für jede Achse) werden über das folgende Maschinendatum festgelegt: MD30600 $MA_FIX_POINT_POS[n]
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Deaktiviert die Funktion "Festpunkt anfahren in JOG".
Korrespondierend mit ...	DB31, ... DBX75.0-2 (Festpunkt anfahren in JOG aktiv) DB31, ... DBX75.3-5 (Festpunkt anfahren erreicht) MD30600 $MA_FIX_POINT_POS[n] (Festwertpositionen der Achse)

19.4.6 Signale von Achse/Spindel (DB31, ...)

Übersicht der Signale von Achse/Spindel

DB31, ...	**Signale von Achse/Spindel**							
DBB	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
64	Fahrbefehl		Fahranforderung			Handrad aktiv		
	plus	minus	plus	minus		3	2	1
65		Aktive Maschinenfunktion						
		kontinuier-lich	variabel INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
75			JOG - Festpunkt anfahren erreicht			JOG - Festpunkt anfahren aktiv		

Beschreibung der Signale von Achse/Spindel

DB31, ... DBX62.1	**Handradüberlagerung aktiv**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Funktion "Handradüberlagerung in Automatik" ist für die programmierte Positionierachse (FDA[AXi]) aktiv. Handradimpulse für diese Achse wirken entweder als Wegvorgabe (bei FDA=0) oder als Geschwindigkeitsüberlagerung (bei FDA > 0) auf den programmierten Achsvorschub. Das Nahtstellensignal wird auch gesetzt, wenn "Handradüberlagerung in Automatik" bei einer konkurrierenden Positionierachse mit FC18 (bei 840D sl) aktiv ist.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Funktion "Handradüberlagerung in Automatik" ist für die programmierte Positionierachse (oder konkurrierende Positionierachse) nicht aktiv. Eine aktive Handradüberlagerung wird unwirksam, wenn: • die Positionierachse die programmierte Zielposition erreicht hat. • der Restweg durch das achsspezifische Nahtstellensignal: DB31, ... DBX2.2 (Restweg löschen) gelöscht wird. • RESET betätigt wird.

DB31, ... DBB64 Bit 0-2	**Handrad aktiv (1 bis 3)**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit diesen PLC-Nahtstellensignalen wird zurückgemeldet, ob diese Maschinenachse dem Handrad 1, 2 oder 3 bzw. keinem Handrad zugeordnet ist. Zu einem Zeitpunkt kann einer Achse jeweils nur ein Handrad zugeordnet werden. Sind mehrere Nahtstellensignale: DB31, ... DBX4.0 bis DBX4.2 (Handrad aktivieren) gesetzt, so gilt die Priorität "Handrad 1" vor "Handrad 2" vor "Handrad 3". Ist die Zuordnung aktiv, so kann die Maschinenachse mit dem Handrad in der Betriebsart JOG verfahren werden oder in der Betriebsart AUTOMATIK bzw. MDA eine DRF-Verschiebung erzeugt werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Dieser Maschinenachse ist Handrad 1, 2 oder 3 nicht zugeordnet.
Korrespondierend mit ...	DB31, ... DBX4.0 bis DBX4.2 (Handrad aktivieren) DB10 DBB100.6 ff (Handrad angewählt)

DB31, ... DBB64 Bit 5, 4	**Fahranforderung plus und minus**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	In der betreffenden Achsrichtung soll eine Fahrbewegung erfolgen. Der Fahrbefehl wird entsprechend der Betriebsart auf unterschiedliche Weise ausgelöst: • Betriebsart JOG: Mit Verfahrtaste plus bzw. minus. • Betriebsart REF: Mit Verfahrtaste, die zum Referenzpunkt hinführt. • Betriebsart AUT/MDA: Ein Programmsatz, der einen Koordinatenwert für die betreffende Achse enthält, wird ausgeführt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	In der betreffenden Achsrichtung steht momentan keine Fahranforderung an bzw. ist eine erfolgte Verfahrbewegung beendet. • Betriebsart JOG: Der Fahrbefehl wird in Abhängigkeit von der Einstellung "Tipp- oder Dauerbetrieb" zurückgesetzt (siehe Nahtstellensignal DB31, ... DBX4.7 bzw. DBX4.6). Beim Verfahren mit Handrad. • Betriebsart REF: Mit Erreichen des Referenzpunktes. • Betriebsart AUT/MDA: Der Programmsatz ist abgearbeitet (und der nachfolgende Programmsatz enthält keinen Koordinatenwert für die betreffende Achse). Abbruch durch "RESET", etc. DB21 … DBX25.7 (Achsensperre) steht an.
Anwendungs- beispiel(e)	Lösen der Klemmung bei Achsen mit Klemmung (z. B. bei Rundtischen). **Hinweis:** Wird die Klemmung erst mit dem Fahrbefehl gelöst, so ist bei diesen Achsen kein Bahnbetrieb möglich!
Korrespondierend mit ...	DB31, ... DBX4.7 bzw. DBX4.6 (Verfahrtaste plus und Verfahrtaste minus) DB31, ... DBX64.7 bzw. DBX64.6 (Fahrbefehl plus und minus)

DB31, ... DBB64 Bit 7, 6	**Fahrbefehl plus und minus**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Das Signal wirkt wie beschrieben, wenn das Bit 0 im Maschinendatum: MD17900 $MN_VDI_FUNCTION_MASK (Einstellung zu VDI-Signalen) gleich 0 gesetzt ist. Ist das Bit 0 im MD auf 1 gesetzt, so zeigt das Signal nur dann 1, wenn die Achse tatsächlich fährt. Das immer ausgegebene Nahtstellensignal: DB31, ... DBX64 Bit 5, 4 (Fahranforderung plus bzw. Fahranforderung minus) verhält sich wie Signal Fahrbefehl plus/minus bei MD17900 Bit 0 = 0.	

DB31, ... DBB64 Bit 7, 6	**Fahrbefehl plus und minus**
Signalzustand 1 bzw. Flankenwechsel 0 → 1	In der betreffenden Achsrichtung soll eine Fahrbewegung erfolgen. Der Fahrbefehl wird entsprechend der Betriebsart auf unterschiedliche Weise ausgelöst: • Betriebsart JOG: Mit Verfahrtaste plus bzw. minus. • Betriebsart REF: Mit Verfahrtaste, die zum Referenzpunkt hinführt. • Betriebsart AUT/MDA: Ein Programmsatz, der einen Koordinatenwert für die betreffende Achse enthält, wird ausgeführt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	In der betreffenden Achsrichtung steht momentan keine Fahranforderung an bzw. ist eine erfolgte Verfahrbewegung beendet. • Betriebsart JOG: Der Fahrbefehl wird in Abhängigkeit von der Einstellung "Tipp- oder Dauerbetrieb" zurückgesetzt (DB31, ... DBX4.7 bzw. DBX4.6). Beim Verfahren mit Handrad. • Betriebsart REF: Mit Erreichen des Referenzpunktes. • Betriebsart AUT/MDA: Der Programmsatz ist abgearbeitet (und der nachfolgende Programmsatz enthält keinen Koordinatenwert für die betreffende Achse). Abbruch durch "RESET", etc. DB21 … DBX25.7 (Achsensperre) steht an.
Anwendungs- beispiel(e)	Lösen der Klemmung bei Achsen mit Klemmung (z. B. bei Rundtischen). **Hinweis:** Wird die Klemmung erst mit dem Fahrbefehl gelöst, so ist bei diesen Achsen kein Bahnbetrieb möglich!
Korrespondierend mit ...	DB31, ... DBX4.7 bzw. DBX4.6 (Verfahrtaste plus und Verfahrtaste minus) DB31, ... DBX64.5 bzw. DBX.4 (Fahranforderung plus und minus)

DB31, ... DBB65 Bit 0-6	**Aktive Maschinenfunktion INC1, ..., kontinuierlich**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	An die PLC-Nahtstelle wird zurückgemeldet, welche Maschinenfunktion in der Betriebsart JOG für Maschinenachsen wirksam ist. Abhängig von der aktiven Maschinenfunktion ist die Reaktion bei Betätigung der Verfahrtaste oder bei Verdrehung des Handrades unterschiedlich.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Entsprechende Maschinenfunktion ist nicht aktiv.
Korrespondierend mit ...	DB31, ... DBB5 (Maschinenfunktion INC1, ..., kontinuierlich)

DB31, ... DBB75 Bit 0-2	JOG - Festpunkt anfahren aktiv
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Meldung an die PLC, dass die Funktion "Festpunkt anfahren in JOG" wirksam ist. Mit den Verfahrtasten oder dem Handrad kann die angewählte Maschinenachse auf den durch Bit 0-2 binärcodiert angegebenen Festpunkt gefahren werden.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	"Festpunkt anfahren in JOG" ist nicht aktiv.
Korrespondierend mit ...	DB31, ... DBX13.0-2 (JOG - Festpunkt anfahren) DB31, ... DBX75.3-5 (JOG - Festpunkt anfahren erreicht)

DB31, ... DBB75 Bit 3-5	JOG - Festpunkt anfahren erreicht
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Meldung an die PLC, dass die angewählte Achse durch die Verfahrbewegung in JOG den anzufahrenden Festpunkt mit "Genauhalt fein" erreicht hat. Dieses Anzeigesignal wird auch gemeldet, wenn die Achse über andere Methoden, wie z. B. NC-Programm, FC18 (bei 840D sl) oder Synchronaktion, sollwertseitig die Festpunktposition im Maschinenkoordinatensystem erreicht und istwertseitig innerhalb des Toleranzfensters "Genauhalt fein" (MD36010 $MA_STOP_LIMIT_FINE) zum Stehen kommt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Achse hat den anzufahrenden Festpunkt noch nicht erreicht.
Korrespondierend mit ...	DB31, ... DBX13.0-2 (JOG - Festpunkt anfahren) DB31, ... DBX75.0-2 (JOG - Festpunkt anfahren aktiv)

19.5 Kompensationen (K3)

Keine Signalbeschreibungen erforderlich.

19.6 BAGs, Kanäle, Achstausch

19.6.1 Signale an Achse/Spindel (DB31, ...)

DB31, ... DBB8	**Achs-/Spindeltausch**	
Flankenauswertung: ja		Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Es muss der aktuelle Achstyp und der aktuelle zuständige Kanal für diese Achse angegeben werden. Beim Achstausch durch PLC bedeuten von Signal an Achse/Spindel DB31, ... DBB8:	
	Bit 0:	A NC-Achse/Spindel Kanal zuordnen
	Bit 1:	B ...
	Bit 2:	C ...
	Bit 3:	D NC-Achse/Spindel Kanal zuordnen
	Bit 4:	Aktivierung, Zuordnung durch positive Flanke
	Bit 5:	-
	Bit 6:	-
	Bit 7:	PLC-Achse/Spindel anfordern
Signalzustand 0 bzw. Flankenwechsel 1 → 0		
Korrespondierend mit ...	DB31, ... DBB68 (Achs-/Spindeltausch) MD20070 $MC_AXCONF_ASSIGN_MASTER_USED (Maschinenachsnummer gültig im Kanal) MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN (Löschstellung des Kanals für Achstausch)	
Sonderfälle, Fehler, ...		

19.6.2 Signale von Achse/Spindel (DB31, ...)

DB31, ... DBB68	**Achs-/Spindeltausch**	
Flankenauswertung: ja		Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Es wird der aktuelle Achstyp und der aktuelle zuständige Kanal für diese Achse angezeigt. Beim Achstausch durch PLC bedeuten von Signal von Achse/Spindel DB31, ... DBB68:	
	Bit 0:	A NC-Achse/Spindel in Kanal
	Bit 1:	B ...
	Bit 2:	C ...
	Bit 3:	D NC-Achse/Spindel in Kanal
	Bit 4:	neuer Typ von PLC gefordert
	Bit 5:	Achstausch möglich

DB31, ... DBB68	Achs-/Spindeltausch	
	Bit 6:	neutrale Achse/Spindel sowie Kommando-/Pendelachsen
	Bit 7:	PLC-Achse/Spindel
Signalzustand 0 bzw. Flankenwechsel 1 → 0		
Korrespondierend mit ...	DB31, ... DBB8 (Achs-/Spindeltausch) MD20070 $MC_AXCONF_ASSIGN_MASTER_USED (Maschinenachsnummer gültig im Kanal) MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN (Löschstellung des Kanals für Achstausch)	
Sonderfälle, Fehler, ...		

19.7 Kinematische Transformation

19.7.1 Signale von Kanal (DB21, ...)

DB21, ... DBX33.6	Transformation aktiv
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Im Teileprogramm ist der NC-Befehl `TRANSMIT`, `TRACYL`, `TRAANG` oder `TRAORI` programmiert. Der entsprechende Satz wurde von der NC abgearbeitet und eine Transformation ist nun aktiviert.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Keine Transformation ist aktiv.
Weiterführende Literatur	/PGA/ Programmieranleitung Arbeitsvorbereitung /FB3/ Funktionsbeschreibung Sonderfunktionen; 3- bis 5-Achs-Transformation (F2)

19.8 Messen

19.8.1 Signale von NC (DB10)

DB10 DBX107.0 und DBX107.1	Messtaster betätigt	
Flankenauswertung: nein		Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der Messtaster 1 bzw. 2 ist betätigt.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der Messtaster 1 bzw. 2 ist nicht betätigt.	
Weiterführende Literatur	/PHD/ Gerätehandbuch NCU	

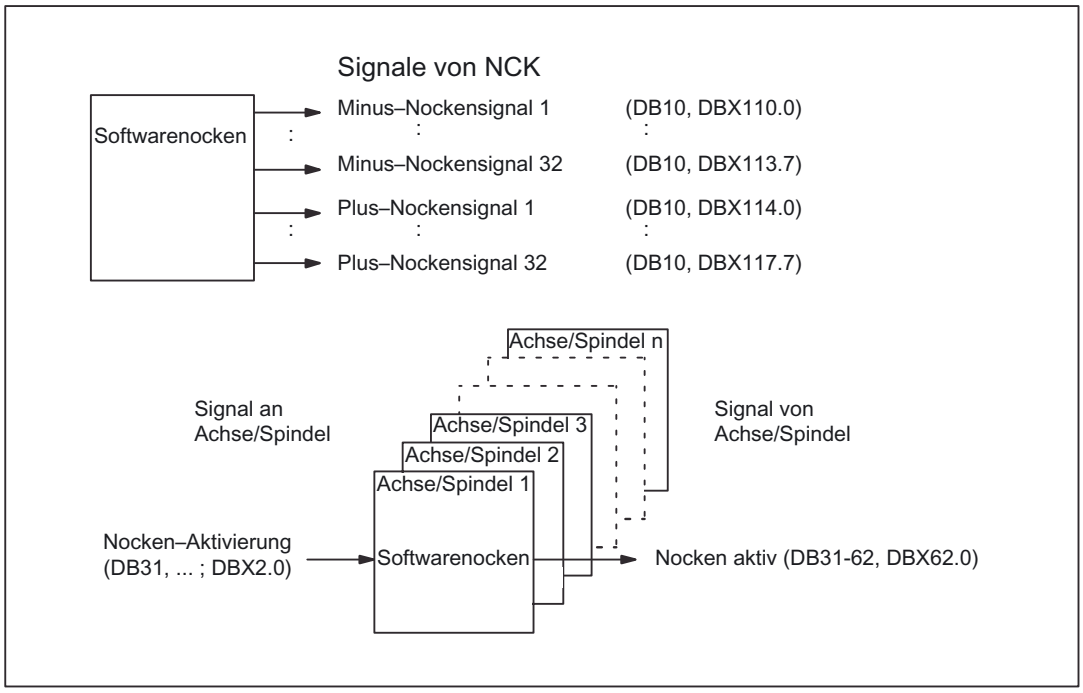
19.8.2 Signale von Achse/Spindel (DB31, ...)

DB31, ... DBX62.3	Messstatus	
Flankenauswertung: nein		Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Funktion "Messen" ist aktiv. Dieses Signal wird beim Messen verwendet und zeigt den augenblicklichen Messstatus der Achse an.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Funktion "Messen" ist nicht aktiv.	

19.9 Softwarenocken, Wegschaltsignale

19.9.1 Signalübersicht

PLC-Nahtstellensignale für "Softwarenocken, Wegschaltsignale"



19.9.2 Signale von NC (DB10)

DB10 DBX110.0-113.7	**Minus-Nockensignal 1-32**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Schaltflanke des Minus-Nockensignals 1-32 wird abhängig von der Verfahrrichtung der (Rund-) Achse erzeugt und im IPO-Takt an die PLC-Nahtstelle übertragen. **Linearachse:** Das Minus-Nockensignal schaltet von 0 auf 1, wenn die Achse den Minusnocken in negativer Achsrichtung überfährt. **Modulo-Rundachse:** Das Minus-Nockensignal wechselt den Pegel bei jeder positiven Flanke des Plus-Nockensignals.

DB10 DBX110.0-113.7	Minus-Nockensignal 1-32
Signalzustand 0 bzw. Flankenwechsel 1 → 0	**Linearachse:** Das Minus-Nockensignal schaltet von 1 auf 0, wenn die Achse den Minusnocken in positiver Achsrichtung überfährt. **Modulo-Rundachse:** Das Minus-Nockensignal wechselt den Pegel bei jeder positiven Flanke des Plus-Nockensignals.

DB10 DBX114.0-117.7	Plus-Nockensignal 1-32
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Schaltflanke des Plus-Nockensignals 1-32 wird abhängig von der Verfahrrichtung der (Rund-) Achse erzeugt und im IPO-Takt an die PLC-Nahtstelle übertragen. **Linearachse:** Das Plus-Nockensignal schaltet von 0 auf 1, wenn die Achse den Plusnocken in positiver Richtung überfährt. **Modulo-Rundachse:** Das Plus-Nockensignal schaltet bei Überfahren des Minusnockens in positiver Achsrichtung von 0 auf 1. Das beschriebene Verhalten des Plusnockens gilt unter der Bedingung: Plusnocken - Minusnocken < 180 Grad Ist diese Bedingung nicht erfüllt oder wird der Minusnocken größer als der Plusnocken gewählt, so invertiert sich das Verhalten des Plus-Nockensignals. Das Verhalten des Minus-Nockensignals bleibt gleich.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	**Linearachse:** Das Plus-Nockensignal schaltet von 1 auf 0, wenn die Achse den Plusnocken in negativer Richtung überfährt. **Modulo-Rundachse:** Das Plus-Nockensignal schaltet bei Überfahren des Plusnockens in positiver Achsrichtung von 1 auf 0 zurück. Das beschriebene Verhalten des Plusnockens gilt unter der Bedingung: Plusnocken - Minusnocken < 180 Grad Ist diese Bedingung nicht erfüllt oder wird der Minusnocken größer als der Plusnocken gewählt, so invertiert sich das Verhalten des Plus-Nockensignals. Das Verhalten des Minus-Nockensignals bleibt gleich.

19.9.3 Signale an Achse/Spindel (DB31, ...)

DB31, ... DBX2.0	Nocken-Aktivierung
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Ausgabe der Minus- und Plus-Nockensignale einer Achse an die allgemeine PLC-Nahtstelle wird aktiviert. Nach Bearbeitung des NST "Nocken-Aktivierung" ist die Aktivierung sofort wirksam.

DB31, ... DBX2.0	Nocken-Aktivierung
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Minus- und Plus-Nockensignale einer Achse werden **nicht** an die allgemeine PLC-Nahtstelle ausgegeben.
Korrespondierend mit ...	DB10 DBX110.0 - 113.7 (Minus-Nockensignal 1-32) DB10 DBX114.0 - 117.7 (Plus-Nockensignal 1-32)

19.9.4 Signale von Achse/Spindel (DB31, ...)

DB31, ... DBX62.0	Nocken aktiv	
Flankenauswertung: nein		Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Alle Nocken der über das NC/PLC-Nahtstellensignals: DB31, ... DBX2.0 (Nocken-Aktivierung) angewählten Achse wurden erfolgreich aktiviert.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Nocken der Achse sind **nicht** aktiviert.	
Korrespondierend mit ...	DB31, ... DBX2.0 (Nocken-Aktivierung) DB10 DBX110.0 - 113.7 (Minus-Nockensignal 1-32) DB10 DBX114.0 - 117.7 (Plus-Nockensignal 1-32)	

19.10 Stanzen und Nibbeln

19.10.1 Signalübersicht

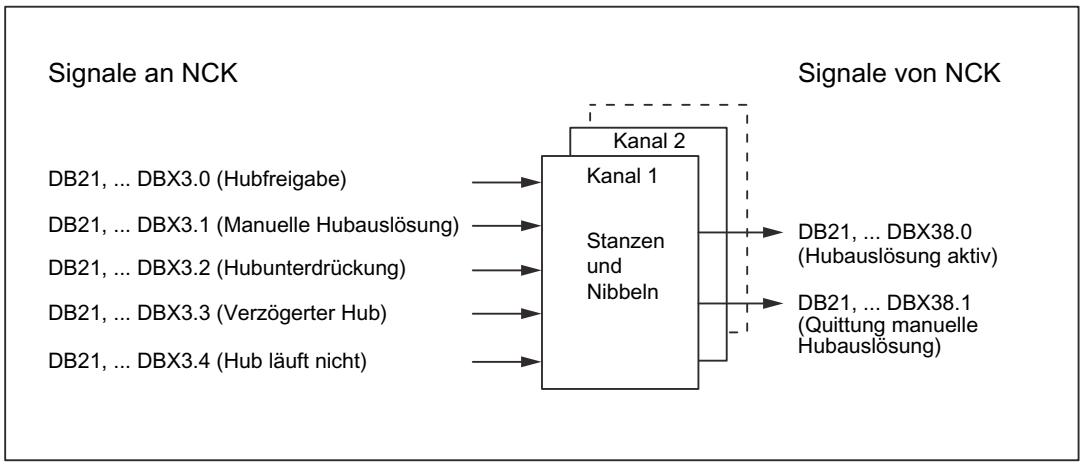


Bild 19-1 PLC-Nahtstellensignale für "Stanzen und Nibbeln"

19.10.2 Signale an Kanal (DB21, ...)

DB21, ... DBX3.0	**Keine Hubfreigabe**
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Über dieses Signal erfolgt die Freigabe der Stanzhübe über die PLC. 1-Signal: Der Hub ist gesperrt, die NC darf keinen Stanzhub auslösen.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	0-Signal: Die Hubfreigabe ist vorhanden. Solange die Freigabe nicht gesetzt ist, kann die NC einen Stanzhub ausführen

DB21, ... DBX3.1	**Manuelle Hubauslösung**
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Dieses Signal ermöglicht es, im Handbetrieb einen einzelnen Hub auszulösen. 1-Signal: Manueller Hub wird ausgeführt.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	0-Signal: Keine Wirkung.

DB21, ... DBX3.2	**Hubunterdrückung**
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Das Signal verhindert lediglich den Hub. Die Maschine fährt trotzdem. Falls die automatische Wegaufteilung aktiv sein sollte, bleibt diese auch aktiv. Es wird lediglich das Signal "Hubauslösung" unterdrückt. Die Maschine verfährt im "Stop and Go"-Betrieb. Die Schrittlänge ist durch die Wegaufteilung definiert. 1-Signal: Die Hubunterdrückung ist aktiv.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	0-Signal: Die Hubunterdrückung ist nicht aktiv.

DB21, ... DBX3.3	**Verzögerter Hub**
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Über dieses Signal kann ein "Verzögerter Hub" aktiviert werden. Dies entspricht funktional der Programmierung von `PDELAYON`. Weitere, nicht dem Standard entsprechende PLC-Signale werden seitens des NCK nicht ausgewertet. Die Auswertung der Signale ist mit Ausnahme der manuellen Hubauslösung auf `PON` aktiv beschränkt. 1-Signal: Verzögerter Hub ist aktiv.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	0-Signal: Verzögerter Hub ist nicht aktiv.

DB21, ... DBX3.4	**Hub läuft nicht**
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die NC reagiert auf dieses Signal mit sofortigem Bewegungsstopp. Falls eine Bewegung oder eine andere Aktion aufgrund dieses Signals unterbrochen werden sollte, erfolgt ein Alarm. Physikalisch ist das Signal für die CNC identisch mit dem Signal "Hub aktiv", d. h. das System ist so beschaltet, dass die beiden Signale über eine UND-Verknüpfung auf denselben NC-Eingang geführt werden. 1-Signal: Hub läuft nicht (entspricht dem Signal "Hubfreigabe").
Signalzustand 0 bzw. Flankenwechsel 1 → 0	0-Signal: Hub läuft (entspricht dem Signal "Hubfreigabe").

DB21, ... DBX3.5	**Manuelle Hubaulösung**
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Über das Signal "Manuelle Hubauslösung" ist eine Stanzauslösung über den Bediener auch ohne Teileprogrammbearbeitung möglich. Damit wird Stanzauslösung vom PLC aus gesteuert. Eine erfolgte Hubauslösung wird dem PLC mit dem NC/PLC-Nahtstellensignal: DB21, ... DBX38.1 (Quittierung manuelle Hubauslösung) angezeigt. 1-Signal: Manuelle Hubauslösung ist aktiv.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	0-Signal: Manuelle Hubauslösung ist nicht aktiv.

19.10.3 Signale von Kanal (DB21, ...)

DB21, ... DBX38.0	**Hubauslösung aktiv**
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit diesem Signal wird angezeigt, ob die Hubauslösung aktiv ist. 1-Signal: Die Hubauslösung ist aktiv.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	0-Signal: Die Hubauslösung ist nicht aktiv.

DB21, ... DBX38.1	**Quittung manuelle Hubauslösung**
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Mit diesem Signal wird angezeigt, ob ein manueller Hub ausgelöst wurde. 1-Signal: Manueller Hub wurde ausgelöst.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	0-Signal: Manueller Hub wurde nicht ausgelöst.

19.11 Positionierachsen

Folgende Signale bzw. Kommandos an der Nahtstelle NCK-HMI-PLC haben ausschließlich für die Positionierachse eine Bedeutung:

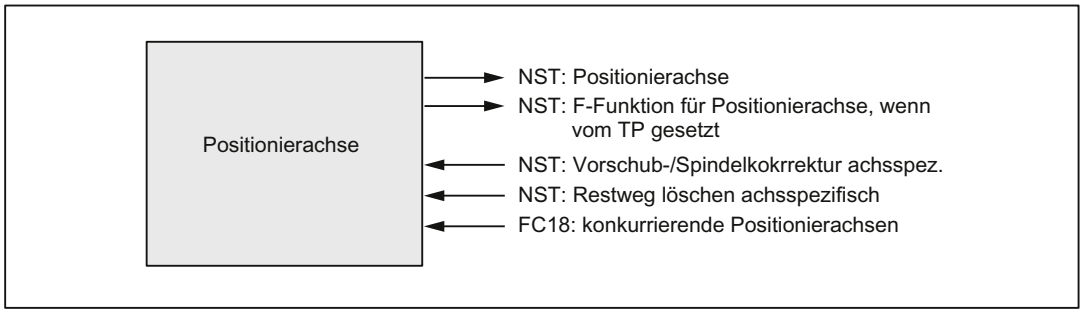


Bild 19-2 Signalbeeinflussung durch die PLC

19.11.1 Signale an Achse/Spindel (DB31, ...)

DB31, ... DBB0	**Vorschubkorrektur/Spindelkorrektur achspezifisch**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Positionierachsen haben eine eigene achsspez. Vorschubkorrektur. Die Bewertung der achsspez. Vorschubkorrektur erfolgt wie bei der kanalspez. Vorschubkorrektur.
Signal irrelevant bei ...	NST DB31, ... DBX74.5 ("Positionierachse") = NULL
Weiterführende Literatur	Bewertung siehe: DB21, ... DBB4 (Vorschubkorrektur); kanalspezifisch

DB31, ... DBX2.2	**Restweg löschen achspezifisch**
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der achsspezifische Restweg der Positionierachse wird abgebrochen. Die Positionierachse wird mit ihrer Beschleunigung abgebremst und der Schleppabstand abgebaut. Die programmierte Endposition gilt als erreicht. Mit dem NST "Restweg löschen" achsspezifisch werden die Bahnachsen nicht beeinflusst. Dazu dient das NST "Restweg löschen" (kanalspez).
Sonderfälle, Fehler, ...	Wird das NST "Restweg löschen" achsspezifisch gesetzt, obwohl in diesem Satz keine Positionierachsen programmiert wurden, reagiert die NCK nicht.
Korrespondierend mit ...	DB21, ... DBX6.2 (Restweg löschen); kanalspezifisch für Bahnachsen

DB31, ... DBX28.1	**AxReset Achse/Spindel zurücksetzen**
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Eine vom PLC kontrollierte Achse/Spindel soll wieder zurückgesetzt werden. Die aktive Achse/Spindel wird vom NCK mit den NST DB31 ... DBX63.2 ("AXSTOP aktiv") in den gestoppten Zustand gebracht.
Sonderfälle, Fehler, ...	Randbedingung: Die Achse/Spindel muss die PLC tatsächlich übernommen haben und vom PLC kontrolliert werden.
Korrespondierend mit ...	DB31, ... DBX63.2 (AXRSTOP); Achse/Spindel stoppen

DB31, ... DBX28.2	**AxResume Achse/Spindel weiterfahren**
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Eine vom Hauptlauf interpolierende Achse/Spindel fährt unabhängig vom NC-Programm weiter und wird vom PLC beeinflusst. Vom NCK wird geprüft, ob ein axialer Alarm die Achse/Spindel mit den Löschkriterium CANCELCLEAR oder NCSTARTCLEAR ansteht und löscht diesen. Der achsspezifische RESUME (DB31 ... DBX28.2) kann vom NCK mit den NST DB31 … DBX28.2 ("AXSTOP aktiv") abgebrochen werden.
Sonderfälle, Fehler, ...	Randbedingung: Die Achse/Spindel muss vom PLC kontrolliert werden. Das NST DB31 … DBX28.2 ("AXRESUME") wird bei folgenden vom NCK festgestellten Fehlerfällen ignoriert: Wenn PLC auffordert Achsen/Spindeln weiterzufahren, und diese Achse/Spindel nicht vom PLC übernommen wurde. Die Achse/Spindel sich nicht im gestoppten Zustand befindet. Die Achse/Spindel wegen eines anstehenden Alarms nicht fortgesetzt werden darf.
Korrespondierend mit ...	DB31, … DBX28.1 (AXRESET); Achse/Spindel zurücksetzen

DB31, ... DBX61.1	**AxAlarm Axialer Stoppalarm für diese Achse**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Achse/Spindel wird vom NCK über eine Rampe abgebremst und bestätigt den Bremsvorgang über BTSS. Gleichzeitig wird der Alarm der PLC mit NST DB31 ... DBX61.1 ("Axialer Alarm") == 1 gemeldet und der Zustand Systemvariable $AA_SNGLAX_STAT == 5 gesetzt.
Korrespondierend mit ...	Systemvariable $AA_SNGLAX_STAT == 5

DB31, ... DBX63.0	**AxReset done Axialer Reset wurde durchgeführt**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Nachdem die bei Reset wirksamen axialen Maschinendaten wirksam wurden, wird das NST DB31 ... DBX63.0 ("AXRESET DONE") == 1, NST DB31 ... DBX63.2 ("Axstop aktiv") == 0 und Systemvariable $AA_SNGLAX_STAT == 1 Einzelachse in Reset aktiv gemeldet.
Korrespondierend mit ...	DB31 ... DBX63.2 (" Axstop aktiv") == 0 Systemvariable $AA_SNGLAX_STAT == 1

DB31, ... DBX63.1	**PLC kontrolliert Achse Achskontrolle an dem PLC übergeben**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die vom NCK bestätigte Achs/Spindel Übernahme wird mit den Zustand der Achse/Spindel mit den NST DB31 ... DBX63.1 ("PLC kontrolliert Achse") == 1 zum PLC übertragen. Mit Systemvariable $AA_SNGLAX_STAT kann der aktuelle Achsstatus abgefragt werden.
Korrespondierend mit ...	DB31 … DBX28.7 (PLC kontrolliert Achse); Achse wird vom PLC kontrolliert. Systemvariable $AA_SNGLAX_STAT aktueller Achsstatus

DB31, … DBX63.2	**AxStop aktiv Quittierung des gestoppten Zustands**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	NCK führt die Achse/Spindel mit den NST DB31 … DBX63.2 ("AXSTOP ACTIVE") == 0 in den Zustand gestoppt. Alle vom Hauptlauf gesteuerten Achsbewegungen werden gestoppt. Achsstatus mit Systemvariable $AA_SNGLAX_STAT == 3 unterbrochen.
Korrespondierend mit ...	DB31 … DBX28.6 (AXSTOP, Halt); Achse/Spindel stoppen Systemvariable $AA_SNGLAX_STAT == 3

DB31, ... DBX76.5	**Positionierachse Achse wird vom NCK als Positionierachse behandelt**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Achse wird von der NCK als Positionierachse behandelt. Damit hat sie: • einen eigenen Achsinterpolator (Linearinterpolator) • einen eigenen Vorschub (F-Wert) • eine eigene Vorschubkorrektur • Genauhalt (G09) an der progr. Endposition

DB31, ... DBB78-81	F-Funktion (Vorschub) für Positionierachse Vorschub durch den programmierten Achsnamen einer Positionierachse zuordnen
Flankenauswertung: nein	Signal(e) aktualisiert: bei Änderung
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der programmierte axiale Vorschub wird durch den programmierten Achsnamen einer Positionierachse zugeordnet und für diese Achse an die PLC ausgegeben. Es erfolgt keine Ausgabe des durch FC18 (bei 840D sl) vorgegebenen Werts.
Signal irrelevant bei ...	NST: Positionierachse = NULL
Sonderfälle, Fehler, ...	Wird die Positionierachse mit dem Vorschub aus folgendem Maschinendatum verfahren, gibt die NC keine F-Funktion (Vorschub) an die PLC aus: MD32060 $MA_POS_AX_VELO (Löschstellung für Positionierachsgeschwindigkeit)
Korrespondierend mit ...	DB31, ... DBX74.5 (Positionierachse) MD22240 $MC_AUXFU_F_SYNC_TYPE (Ausgabezeitpunkt der F-Funktionen)

19.11.2 Function Call - nur 840D sl

FC18

Mit dem FC18 (Function Call 18) der PLC können bei SINUMERIK 840D sl konkurrierende Positionierachsen von der PLC gestartet werden. Der FC wird dabei mit folgenden Parametern versorgt:

- Achsname/Achsnummer
- Endposition
- Vorschub

  (bei Vorschub=0 wird der Vorschub aus MD32060 $MA_POS_AX_VELO genommen)

  Der F-Wert des FC18 wird **nicht** in die achsspezifische NST DB31, ...DBB78-81 ("F-Funktion (Vorschub) für Positionierachse") übertragen.
- Bezugsmaß (G90), Kettenmaß (G91), Bezugsmaß auf kürzestem Weg für Rundachsen (Rundachsname = DC(Wert))

Da jede Achse genau einem Kanal zugeordnet ist, kann die Steuerung aus dem Achsnamen/der Achsnummer den richtigen Kanal auswählen und die konkurrierende Positionierachse in diesem Kanal starten.

Literatur:
Funktionshandbuch Grundfunktionen; PLC-Grundprogramm für SINUMERIK 840D sl (P3)

19.12 Pendeln

19.12.1 Signale Achse/Spindel (DB31, ...)

VDI Input Signale

Mit den folgenden Signalen kann das PLC-Anwenderprogramm den Pendelvorgang steuern.

DB31, ... DBX28.0	Pendelumkehr von Extern
Flankenauswertung: ja	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Pendelbewegung abbremsen und in Gegenrichtung mit der Pendelachse weiterfahren.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Ohne Unterbrechung weiter pendeln

DB31, ... DBX28.3	Umkehrpunkt setzen
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Umkehrpunkt2
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Umkehrpunkt1

DB31, … DBX28.4	Umkehrpunkt ändern
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Der gewählte Umkehrpunkt kann über Handfahren verändert werden. In Verbindung mit DB31, …DBX28.0: Die Position, auf die nach Pendelumkehr von Extern abgebremst wurde, soll neuer Umkehrpunkt sein.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der gewählte Umkehrpunkt kann über Handfahren nicht verändert werden. In Verbindung mit DB31, …DBX28.0: Keine Umkehrpunktänderung
Korrespondierend mit ...	DBX28.3

DB31, ... DBX28.5	Halt im nächsten Umkehrpunkt
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Pendelbewegung wird im nächsten Umkehrpunkt unterbrochen.

DB31, ... DBX28.5	Halt im nächsten Umkehrpunkt
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Pendelbewegung geht nach dem nächsten Umkehrpunkt weiter.
Korrespondierend mit ...	DBX28.6, DBX28.7

DB31, ... DBX28.6	Halt mit Bremsrampe
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Achse wird mit Rampe abgebremst, die Pendelbewegung ist unterbrochen.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Pendelbewegung geht ununterbrochen weiter.
Korrespondierend mit ...	DBX28.5, DBX28.7

DB31, ... DBX28.7	PLC kontrolliert Achse
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Achse wird von PLC kontrolliert. Die Reaktion auf Nahtstellensignale wird von der PLC über die 2 Halt-Bits gesteuert, einige andere bremsend wirkenden Signale werden ignoriert.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Achse wird nicht von PLC kontrolliert.
Korrespondierend mit ...	DBX28.5, DBX28.6

19.12.2 Signale von Achse/Spindel (DB31, ...)

VDI Output Signale

Die folgenden Signale werden von NCK dem PLC-Anwenderprogramm zur Verfügung gestellt.

DB31, ... DBX100.2	Pendelumkehr aktiv
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Bremsphase nach Pendelumkehr von Extern (DB31, ...DBX28.0) ist aktiv
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Kein Bremsen nach Pendelumkehr von Extern ist aktiv.

DB31, ... DBX100.3	**Pendeln nicht startbar**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Pendelachse kann nicht gestartet werden, fehlerhaft Programmierung. Zustand kann auch auftreten, wenn bereits verfahren wurde.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Der Start der Pendelbewegung ist möglich.

DB31, ... DBX100.4	**Fehler während Pendelbewegung**
Flankenauswertung:	Signal(e) aktualisiert:
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Pendelbewegung wurde abgebrochen.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Pendelbewegung verläuft fehlerfrei.

DB31, … DBX100.5	**Ausfeuern aktiv**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Achse führt Ausfeuerhübe aus.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Achse führt z. Zeit keine Ausfeuerhübe aus.
Korrespondierend mit ...	DBX100.7

DB31, ... DBX100.6	**Pendelbewegung aktiv**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Achse führt eine Pendelbewegung zwischen 2 Umkehrpunkten aus.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Achse pendelt zurzeit nicht.
Signal irrelevant bei	DBX100.7 = 0
Korrespondierend mit ...	DBX100.7

DB31, ... DBX100.7	**Pendeln aktiv**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Achse wird momentan als Pendelachse verfahren.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Achse ist Positionierachse
Korrespondierend mit ...	DBX100.5, DBX100.6

DB31, ... DBX104.0 - 7	Aktive Zustellachsen
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Achse, von der das Signal kommt ist derzeit Pendelachse und meldet in diesem Feld ihre aktiven Zustellachsen (104.0 Achse 1 ist Zustellachse, 104.1 Achse 2 ist Zustellachse, usw.)
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die zugehörige Achse ist keine Zustellachse.
Korrespondierend mit ...	DBX100.7

19.13 Rundachsen

19.13.1 Signale an Achse/Spindel (DB31, ...)

DB31, ... DBX12.4	Verfahrbereichsbegrenzung bei Modulo-Rundachsen
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Verfahrbereichsbegrenzung bei Modulo-Rundachse aktivieren (Softwareendschalter, Arbeitsfeldbegrenzungen).
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Verfahrbereichsbegrenzung bei Modulo-Rundachse deaktivieren.
Signal irrelevant bei ...	Linearachsen / Rundachsen ohne Modulo-Funktionalität.
Anwendungs- beispiel(e)	Aufbaurundachse mit Überwachung

19.13.2 Signale von Achse/Spindel (DB31, ...)

DB31, ... DBX74.4	Überwachungszustand bei Modulo-Rundachsen
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Verfahrbereichsbegrenzung bei Modulo-Rundachse aktiv (Softwareendschalter, Arbeitsfeldbegrenzungen).
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Verfahrbereichsbegrenzung bei Modulo-Rundachse nicht aktiv.

DB31, ... DBX74.4	Überwachungszustand bei Modulo-Rundachsen
Signal irrelevant bei ...	Linearachsen / Rundachsen ohne Modulo-Funktionalität.
Anwendungs-beispiel(e)	Aufbaurundachse mit Überwachung

19.14 Synchronspindel

19.14.1 Signale an Achse/Spindel (DB31, ...)

DB31, ... DBX31.5	Synchronisation sperren
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Synchronisationsbewegung für die Folgespindel wird vom PLC aus nicht gesperrt. Der Positions-Offset wird nicht unterdrückt und wie bisher herausgefahren.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Synchronisationsbewegung für die Folgespindel wird vom PLC aus gesperrt Eine durch Offset Programmierung vorgegebene Synchronisationsbewegungen wird für die Folgespindel unterdrückt. Es erfolgt keine zusätzliche Bewegung der Folgespindel.
korrespondierend mit ...	DB31, ... DBX98.1 (Synchronlauf grob) DB31, ... DBX98.0 (Synchronlauf fein)

19.14.2 Signale von Achse/Spindel (DB31, ...)

DB31, ... DBX84.4	Synchronbetrieb
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Spindel befindet sich in der Spindelbetriebsart "Synchronbetrieb". Damit folgt die Folgespindel den Bewegungen der Leitspindel entsprechend dem Übersetzungsverhältnis. Im Synchronbetrieb werden die Überwachungen auf Synchronlauf grob und fein durchgeführt. Hinweis: Das Signal wird nur für die Maschinenachse gesetzt, die als Folgespindel aktiv ist (NST "FS aktiv" = 1)
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Spindel wird nicht als Folgespindel im "Synchronbetrieb" betrieben. Beim Ausschalten der Kopplung (Abwahl des Synchronbetriebes) wird die Folgespindel in den "Steuerbetrieb" geschaltet.
korrespondierend mit ...	DB31, ... DBX98.0 (Synchronlauf fein) DB31, ... DBX98.1 (Synchronlauf grob) DB31, ... DBX99.1 (FS aktiv)

DB31, ... DBX98.0	**Synchronlauf fein**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Lagedifferenz bzw. Geschwindigkeitsdifferenz zwischen der Folgespindel und ihrer Leitspindel liegt innerhalb des Toleranzbandes "Synchronlauf fein".
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Lagedifferenz bzw. Geschwindigkeitsdifferenz zwischen der Folgespindel und ihrer Leitspindel liegt nicht innerhalb des Toleranzbandes "Synchronlauf fein". Hinweis: Das Signal ist nur für die Folgespindel im Synchronbetrieb von Bedeutung.
Anwendungsbeispiel	Einspannen des Werkstücks in die Folgespindel bei der Übernahme von der Leitspindel: Das Spannen des Werkstücks wird vom PLC-Anwenderprogramm erst dann ausgelöst, wenn die Spindeln ausreichend synchron laufen.
korrespondierend mit ...	DB31, ... DBX84.4 (Synchronbetrieb) MD37210 $MA_COUPLE_POS_TOL_FINE (Schwellwert für "Synchronlauf fein") MD37230 $MA_COUPLE_VELO_TOL_FINE (Geschwindigkeitstoleranz "fein")

DB31, ... DBX98.1	**Synchronlauf grob**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Lagedifferenz bzw. Geschwindigkeitsdifferenz zwischen der Folgespindel und ihrer Leitspindel liegt innerhalb des Toleranzbandes "Synchronlauf grob". Hinweis: Das Signal ist nur für die Folgespindel im Synchronbetrieb von Bedeutung.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Lagedifferenz bzw. Geschwindigkeitsdifferenz zwischen der Folgespindel und ihrer Leitspindel liegt nicht innerhalb des Toleranzbandes "Synchronlauf grob".
Anwendungsbeispiel	Einspannen des Werkstücks in die Folgespindel bei der Übernahme von der Leitspindel. Das Spannen des Werkstücks wird vom PLC-Anwenderprogramm erst dann ausgelöst, wenn die Spindeln ausreichend synchron laufen.
korrespondierend mit ...	DB31, ... DBX84.4 (Synchronbetrieb) MD37200 $MA_COUPLE_POS_TOL_COARSE (Schwellwert für "Synchronlauf grob") MD37220 $MA_COUPLE_VELO_TOL_COARSE (Geschwindigkeitstoleranz "grob")

DB31, ... DBX98.2	**Istwertkopplung**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Zwischen der Leitspindel und der Folgespindel ist als Kopplungsart die Istwertkopplung aktiv (siehe MD21310). Hinweis: Das Signal ist nur für die aktive Folgespindel im Synchronbetrieb von Bedeutung.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Zwischen der Leitspindel und der Folgespindel ist als Kopplungsart eine Sollwertkopplung aktiv (siehe MD21310).
Sonderfälle, Fehler, ...	Bei Störungen an der Folgespindel, die eine Wegnahme der "Reglerfreigabe" für die FS bewirken, wird unter bestimmten Voraussetzungen steuerungsintern die Kopplungsbeziehung von FS und LS auf Istwertkopplung umgeschaltet.
korrespondierend mit ...	DB31, ... DBX84.4 (Synchronbetrieb) MD21310 $MC_COUPLING_MODE_1 (Kopplungsart im Synchronspindelbetrieb)

DB31, ... DBX98.4	**Überlagerte Bewegung**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Von der Folgespindel wird eine zusätzliche Bewegungskomponente abgefahren, die der Bewegung aus der Kopplung mit der Leitspindel überlagert ist. Beispiele für überlagerte Bewegung der FS: - Einschalten des Synchronbetriebs mit definiertem Winkelversatz zwischen FS und LS - Einschalten des Synchronbetriebs bei rotierender LS - Änderung des Übersetzungsverhältnisses bei aktiven Synchronbetrieb - Vorgabe eines neuen definierten Winkelversatzes bei aktiven Synchronbetrieb - Verfahren der FS mit Verfahrtasten plus oder minus oder Handrad in JOG bei aktiven Synchronbetrieb Sobald die FS eine überlagerte Bewegung ausführt, kann das NST "Synchronlauf fein" bzw. das NST "Synchronlauf grob" (abhängig vom Schwellwert) weggenommen werden. Hinweis: Das Signal ist nur für die Folgespindel im Synchronbetrieb von Bedeutung.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Von der Folgespindel wird keine zusätzliche Bewegungskomponente abgefahren bzw. diese ist beendet.
korrespondierend mit ...	DB31, ... DBX84.4 (Synchronbetrieb)

DB31, ... DBX99.0	**LS (Leitspindel) aktiv**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenachse ist momentan als Leitspindel aktiv. Hinweis: Das Signal ist nur im Synchronbetrieb von Bedeutung.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenachse ist momentan nicht als Leitspindel aktiv.
Sonderfälle, Fehler, ...	Bei Störungen an der Folgespindel, die eine Wegnahme der "Reglerfreigabe" für die FS bewirken, wird unter bestimmten Voraussetzungen steuerungsintern die Kopplungsbeziehung von FS und LS getauscht und auf Istwertkopplung umgeschaltet. In diesen Fall wird die bisherige Leitspindel zur neuen aktiven Folgespindel (NST "FS aktiv").
korrespondierend mit ...	DB31, ... DBX84.4 (Synchronbetrieb) DB31, ... DBX99.1 (FS aktiv)

DB31, ... DBX99.1	**FS (Folgespindel) aktiv**
Flankenauswertung: nein	Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die Maschinenachse wird momentan als Folgespindel betrieben. Damit folgt die Folgespindel im Synchronbetrieb den Bewegungen der Leitspindel entsprechend dem Übersetzungsverhältnis. Hinweis: Das Signal ist nur im Synchronbetrieb von Bedeutung.
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die Maschinenachse wird momentan nicht als Folgespindel betrieben.

DB31, ... DBX99.1	FS (Folgespindel) aktiv
Sonderfälle, Fehler, ...	Bei Störungen an der Folgespindel, die eine Wegnahme der "Reglerfreigabe" für die FS bewirken, wird unter bestimmten Voraussetzungen steuerungsintern die Kopplungsbeziehung von FS und LS getauscht und auf Istwertkopplung umgeschaltet.
korrespondierend mit ...	DB31, ... DBX84.4 (Synchronbetrieb) DB31, ... DBX99.0 (LS aktiv)

19.15 Speicherkonfiguration (S7)

Keine Signalbeschreibungen erforderlich.

19.16 Teilungsachsen

19.16.1 Signale von Achse/Spindel (DB31, ...)

DB31, ... DBX76.6	Teilungsachse in Position	
Flankenauswertung: nein		Signal(e) aktualisiert: zyklisch
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Das Signal ist abhängig von "Genauhalt fein": Wenn "Genauhalt fein" erreicht ist, dann wird das Signal gesetzt. Beim Verlassen von "Genauhalt fein" wird das Signal wieder zurückgesetzt. • Die Teilungsachse steht auf einer Teilungsposition. Die Teilungsachse wurde mit Anweisungen für "Codierte Position" positioniert. **Hinweis:** Wenn das Fenster "Genauhalt fein" erreicht wird und die Teilungsachse auf einer Teilungsposition steht, wird das Signal gesetzt, unabhängig davon, wie die Teilungsposition erreicht wurde.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	• Die Achse ist nicht als Teilungsachse definiert. • Die Teilungsachse fährt: DB31, ... DBX64.7/64.6 (Fahrbefehl +/-) steht an. • Die Teilungsachse steht auf einer Position, die nicht einer Teilungsposition entspricht. Beispiele: – bei JOG nach Abbruch der Fahrbewegung z. B. mit RESET – bei Automatik: mit Teilungsachse wurde beispielsweise mit AC- oder DC-Anweisung auf eine beliebige Position angefahren • Die Teilungsachse wurde in der Betriebsart Automatik nicht mit Anweisungen für "Codierte Position" (CAC, CACP, CACN, CDC, CIC) positioniert. • Die "Reglerfreigabe" der Teilungsachse ist weggenommen: DB31, ... DBX2.1 (Reglerfreigabe)	
Signal irrelevant bei ...	... Achsen, die nicht als Teilungsachsen definiert sind: MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB = 0	

DB31, ... DBX76.6	**Teilungsachse in Position**
Anwendungs-beispiel(e)	Werkzeugmagazin: Die Aktivierung des Greifers für die Entnahme des Werkzeugs aus dem Magazin wird dann ausgelöst, sobald die Teilungsachse in Position ist: DB31, ... DBX76.6 (Teilungsachse in Position) = 1. Dies ist vom PLC-Anwenderprogramm sicherzustellen.
Sonderfälle, Fehler, ...	**Hinweise:** Die in der Teilungspositionstabelle für die einzelnen Teilungen eingetragenen Achspositionen können durch Nullpunktverschiebungen (u. a. auch `DRF`) verändert werden. Das Nahtstellensignal: DB31, ... DBX76.6 (Teilungsachse in Position) wird dann auf 1 gesetzt, wenn die Istposition der Teilungsachse den in der Teilungstabelle eingetragenen Positionswert zuzüglich den Korrekturen einnimmt. Wird bei einer Teilungsachse in AUTOMATIK eine `DRF`-Verschiebung bewerkstelligt, so bleibt das Nahtstellensignal "Teilungsachse in Position" weiterhin anstehen, obwohl die Achse nicht mehr auf einer Teilungsposition steht.
Korrespondierend mit ...	MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB (Achse ist Teilungsachse)

19.17 Werkzeugwechsel (W3)

Keine Signalbeschreibungen erforderlich.

19.18 Schleifspezifische Werkzeugkorrektur und Überwachungen

19.18.1 Signale von Achse/Spindel (DB31, ...)

DB31, ... DBX83.3	**Geometrieüberwachung**	
Flankenauswertung: nein		Signal(e) aktualisiert: -
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Fehler Schleifscheibengeometrie. **Hinweis:** Es erfolgt keine weitere Reaktion auf das Ansprechen dieser Überwachung. Erforderliche Reaktionen sind vom PLC-Anwender zu programmieren.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Kein Fehler Schleifscheibengeometrie.	
Anwendungs-beispiel(e)	Schleifspezifische Werkzeugüberwachung.	

DB31, ... DBX83.6	Drehzahlüberwachung	
Flankenauswertung: nein		Signal(e) aktualisiert: -
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Fehler Schleifscheibendrehzahl. **Hinweis:** Es erfolgt keine weitere Reaktion auf diesen Signalzustand. Erforderliche Reaktionen sind vom PLC-Anwender zu programmieren.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Kein Fehler Schleifscheibendrehzahl.	
Anwendungs- beispiel(e)	Schleifspezifische Werkzeugüberwachung.	

DB31, ... DBX84.1	SUG aktiv	
Flankenauswertung: nein		Signal(e) aktualisiert: -
Signalzustand 1 bzw. Flankenwechsel 0 → 1	Die konstante Scheibenumfangsgeschwindigkeit (SUG) ist aktiv. Wenn die SUG aktiv ist, dann werden alle S-Wert-Vorgaben von der PLC als Scheibenumfangsgeschwindigkeit interpretiert.	
Signalzustand 0 bzw. Flankenwechsel 1 → 0	Die konstante Scheibenumfangsgeschwindigkeit (SUG) ist nicht aktiv.	
Anwendungs- beispiel(e)	SUG in allen Betriebsarten.	

Anhang

A

A.1 Liste der Abkürzungen

A	
A	Ausgang
ADI4	Analog Drive Interface for 4 Axis
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
ASUP	Asynchrones Unterprogramm
AUXFU	Auxiliary Function: Hilfsfunktion
AWP	Anwenderprogramm

B	
BA	Betriebsart
BAG	Betriebsartengruppe
BCD	Binary Coded Decimals: Im Binärcode verschlüsselte Dezimalzahlen
BERO	Berührungsloser Endschalter
BI	Binector Input
BICO	Binector Connector
BIN	Binary Files: Binärdateien
BHG	Bedienhandgerät
BKS	Basiskoordinatensystem
BO	Binector Output
BOT	Boot-Datei (SIMODRIVE 611D)
BTSS	Bedientafelschnittstelle

C	
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CC	Compile Cycle: Compile-Zyklen
CI	Connector Input
CF-Card	Compact Flash-Card
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung

CO	Connector Output
CoL	Certificate of License
COM	Communication
CPA	Compiler Projecting Data: Projektierdaten des Compilers
CU	Control Unit
CP	Communication Processor
CPU	Central Processing Unit: Zentrale Rechnereinheit
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)
DIN	Deutsche Industrie Norm
DIO	Data Input/Output: Datenübertragungs-Anzeige
DIR	Directory: Verzeichnis
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
ENC	Encoder: Istwertgeber
EFP	Einfach Peripheriemodul (PLC–E/A–Baugruppe)
EGB	Elektronisch gefährdete Baugruppen/Bauelemente
EMV	Elektromagnetische Verträglichkeit
EN	Europäische Norm

EnDat	Geberschnittstelle
EPROM	Erasable Programmable Read Only Memory (Löschbarer, elektrisch programmierbarer nur Lesespeicher)
EQN	Typbezeichnung eines Absolutwertgebers mit 2048 Sinussignalen/Umdrehung
E/R	Ein–/Rückspeiseeinheit des SIMODRIVE 611(D)
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
FRAME	Koordinatentransformation
FRK	Fräsradiuskorrektur
FST	Feed Stop: Vorschub Halt
FW	Firmware

G	
GC	Global Control (PROFIBUS: Broadcast-Telegramm)
GEO	Geometrie, z.B. Geometrieachse
GIA	Gear Interpolation Data: Getriebeinterpolationsdaten
GND	Signal Ground
GP	Grundprogramm (PLC)
GS	Getriebestufe
GSD	Gerätestammdatei zur Beschreibung eines PROFIBUS Slaves
GSDML	Generic Station Description Markup Language: XML-basierte Beschreibungs-sprache zur Erstellung einer GSD-Datei
GUD	Global User Data: Globale Anwenderdaten

H	
HEX	Kurzbezeichnung für hexadezimale Zahl
HiFu	Hilfsfunktion
HMI	Human Machine Interface, SINUMERIK Bedienoberfläche
HSA	Hauptspindelantrieb
HW	Hardware

HW–Konfig	SIMATIC S7–Tool zum Konfigurieren und Parametrieren von Hardware-Komponenten innerhalb eines S7–Projektes
HW–Endschalter	Hardware–Endschalter

I	
IBN	Inbetriebnahme
IKA	Interpolatorische Kompensation
INC	Increment: Schrittmaß
IPO	Interpolator

J	
JOG	Jogging: Einrichtbetrieb

K	
K_V	Verstärkungsfaktor des Regelkreises
K_P	Proportionalverstärkung
$K_Ü$	Übersetzungsverhältnis
KOP	Kontaktplan

L	
LAN	Local Area Network
LED	Light Emitting Diode: Leuchtdiode
LF	Line Feed
LMS	Lagemesssystem
LR	Lageregler
LSB	Least significant Bit, niederwertigstes Bit
LUD	Local User Data: Anwenderdaten (lokal)

M	
MAC	Media Access Control
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
MELDW	Meldungswort

MKS	Maschinenkoordinatensystem
MLFB	Maschinenlesbare Fabrikatbezeichnung
MM	Motor Module
MMC	Man Machine Communication
MPF	Main Program File: NC-Teileprogramm (Hauptprogramm)
MPI	Multi Port Interface: Mehrpunktfähige Schnittstelle
MSTT	Maschinensteuertafel

N	
NC	Numerical Control: Numerische Steuerung
NCK	Numerical Control Kernel
NCU	Numerical Control Unit
NRK	Bezeichnung des Betriebssystems des NCK
NST	Nahtstellensignal
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

P	
PAA	Prozessabbild der Ausgänge
PAE	Prozessabbild der Eingänge
PC	Personal Computer
PCMCIA	Personal Computer Memory Card International Association (Speichersteckkarten Normierung)
PCU	PC Unit
PG	Programmiergerät
PKE	Parameterkennung: Teil eines PKW
PKW	Parameter Kennung: Wert: Parametrierteil eines PPO
PLC	Programmable Logic Control: Anpass-Steuerung
PN	PROFINET
PNO	PROFIBUS-Nutzerorganisation
PO	POWER ON

POE	Programmorganisationseinheit
POS	Positionieren: z.B. POS-Achse = Positionierachse = Kanalachse die nicht interpolatorische, d.h. unabhängig von anderen Kanalachsen auf ihre programmierte Position verfahren wird.
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 Gleichstromeinspeisung
POSMO SI	Positioning Motor Servo Integrated: Positioniermotor; Gleichstromeinspeisung
PPO	Parameter Prozessdaten Objekt ; Zyklisches Datentelegramm bei der Übertragung mit PROFIBUS–DP und Profil "Drehzahlveränderbare Antriebe"
PROFIBUS	Process Field Bus: Serieller Datenbus
PRT	Programmtest
PSW	Programmsteuerwort
PTP	Point to Point: Punkt zu Punkt
PUD	Program Global User Data: Programmglobale Variable
PZD	Prozessdaten: Prozessdatenteil eines PPO

Q	
QFK	Quadrantenfehler Kompensation

R	
RAM	Random Access Memory: Schreib-/Lese-Speicher
REF	Funktion Referenzpunkt anfahren
REPOS	Funktion Repositionieren
RISC	Reduced Instruction Set Computer: Prozessortyp mit kleinem Befehlssatz und schnellem Befehlsdurchsatz
ROV	Rapid Override: Eingangskorrektur
RP	R-Parameter, Rechenparameter, vordefinierte Anwendervariable
RPY	Roll Pitch Yaw: Drehungsart eines Koordinatensystems
RTLI	Rapid Traverse Linear Interpolation: Lineare Interpolation bei Eilgangbewegung
RTCP	Real Time Control Protocol

S	
SBC	Safe Break Control: Sichere Bremsenansteuerung
SBL	Single Block: Einzelsatz
SD	Settingdatum bzw. Settingdaten
SEA	Setting Data Active: Kennzeichnung (Dateityp) für Settingdaten
SERUPRO	Search–Run by Program Test: Suchlauf via Programmtest

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
SPF	Sub Program File: Unterprogramm
SPS	Speicherprogrammierbare Steuerung = PLC
SRAM	Statischer Speicher (gepuffert)
SRK	Schneidenradiuskorrektur
SRM	Synchron rotatorischer Motor
SSFK	Spindelsteigungsfehlerkompensation
SSI	Synchron Serielles Interface (Schnittstellen–Typ)
SSL	Satzsuchlauf
STW	Steuerwort
SUG	Scheibenumfangsgeschwindigkeit
SW	Software
SW–Endschalter	Software–Endschalter
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)

U	
USB	Universal Serial Bus
USV	Unterbrechungsfreie Stromversorgung

V	
VDI	Interne Kommunikationsschnittstelle zwischen NCK und PLC
VDI	Verein Deutscher Ingenieure
VDE	Verband Deutscher Elektrotechniker
VI	Voltage Input
VO	Voltage Output
VSA	Vorschubantrieb

W	
WKS	Werkstückkoordinatensystem
WKZ	Werkzeug
WLK	Werkzeuglängenkorrektur
WOP	Werkstatt-orientierte Programmierung
WPD	Work Piece Directory: Werkstückverzeichnis
WRK	Werkzeug-Radius-Korrektur
WZ	Werkzeug
WZK	Werkzeugkorrektur
WZV	Werkzeugverwaltung
WZW	Werkzeugwechsel

X	
XML	Extensible Markup Language

Z	
ZOA	Zero Offset Active: Kennung für Nullpunktverschiebungen
ZSW	Zustandswort (des Antriebs)

A.2 Feedback zur Dokumentation

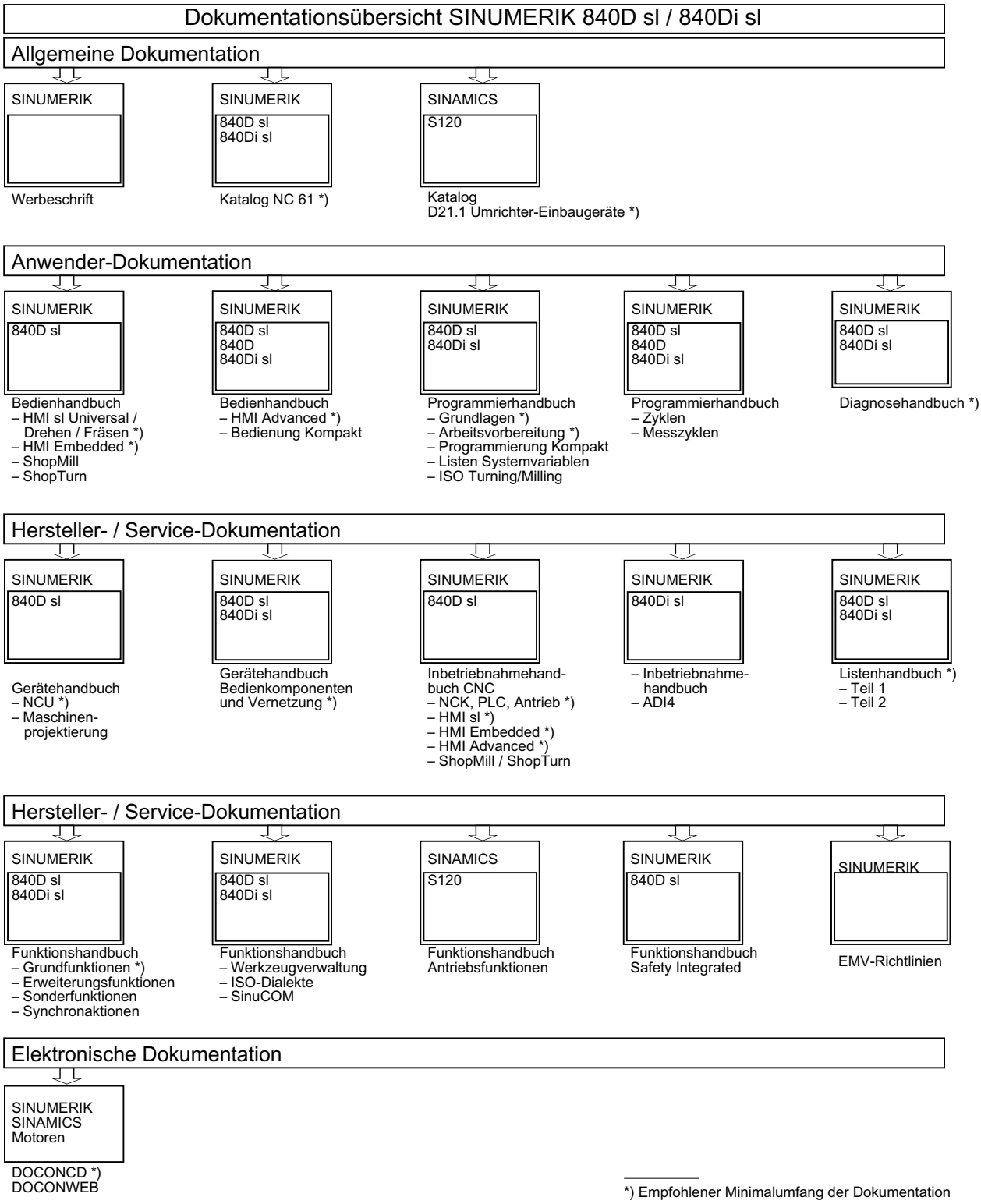
Das vorliegende Dokument wird bezüglich seiner Qualität und Benutzerfreundlichkeit ständig weiterentwickelt. Bitte helfen Sie uns dabei, indem Sie Ihre Anmerkungen und Verbesserungsvorschläge per E-Mail oder Fax senden an:

E-Mail:	mailto:docu.motioncontrol@siemens.com
Fax:	+49 9131 - 98 2176 Bitte verwenden Sie die Faxvorlage auf der Blattrückseite.

An	Absender
SIEMENS AG I DT MC MS1 Postfach 3180 D-91050 Erlangen	Name:
	Anschrift Ihrer Firma / Dienststelle
	Straße:
	PLZ: Ort:
	Telefon: /
Fax.: +49 9131 - 98 2176 (Dokumentation)	Telefax: /

Vorschläge und / oder Korrekturen

A.3 Dokumentationsübersicht



Glossar

Absolutmaß

Angabe des Bewegungsziels einer Achsbewegung durch ein Maß, das sich auf den Nullpunkt des momentan gültigen Koordinatensystems bezieht. Siehe → Kettenmaß.

Achsadresse

Siehe → Achsbezeichner

Achsbezeichner

Achsen werden nach DIN 66217 für ein rechtsdrehendes, rechtwinkliges → Koordinatensystem bezeichnet mit X, Y, Z.

Um X, Y, Z drehende → Rundachsen erhalten die Bezeichner A, B, C. Zusätzliche Achsen, parallel zu den angegebenen, können mit weiteren Adressbuchstaben gekennzeichnet werden.

Achsen

Die CNC-Achsen werden entsprechend ihres Funktionsumfangs abgestuft in:

- Achsen: interpolierende Bahnachsen
- Hilfsachsen: nicht interpolierende Zustell- und Positionierachsen mit achsspezifischem Vorschub. Hilfsachsen sind an der eigentlichen Bearbeitung nicht beteiligt, z. B. Werkzeugzubringer, Werkzeugmagazin.

Achsname

Siehe → Achsbezeichner

Adresse

Eine Adresse ist die Kennzeichnung für einen bestimmten Operanden oder Operandenbereich, z. B. Eingang, Ausgang usw.

Alarme

Alle → Meldungen und Alarme werden auf der Bedientafel im Klartext mit Datum und Uhrzeit und dem entsprechenden Symbol für das Löschkriterium angezeigt. Die Anzeige erfolgt getrennt nach Alarmen und Meldungen.

1. Alarme und Meldungen im Teileprogramm

   Alarme und Meldungen können direkt aus dem Teileprogramm im Klartext zur Anzeige gebracht werden.

2. Alarme und Meldungen von PLC

   Alarme- und Meldungen der Maschine können aus dem PLC-Programm im Klartext zur Anzeige gebracht werden. Dazu sind keine zusätzlichen Funktionsbaustein-Pakete notwendig.

Antrieb

Der Antrieb ist diejenige Einheit der CNC, welche die Drehzahl- und Momentenregelung aufgrund der Vorgaben der NC ausführt.

Anwenderdefinierte Variable

Anwender können für beliebige Nutzung im → Teileprogramm oder Datenbaustein (globale Anwenderdaten) anwenderdefinierte Variablen vereinbaren. Eine Definition enthält eine Datentypangabe und den Variablennamen. Siehe → Systemvariable.

Anwenderprogramm

Anwenderprogramme für Automatisierungssysteme S7-300 werden mit der Programmiersprache STEP 7 erstellt. Das Anwenderprogramm ist modular aufgebaut und besteht aus einzelnen Bausteinen.

Die grundlegenden Bausteintypen sind:

- Code-Bausteine

  Diese Bausteine enthalten die STEP 7-Befehle.

- Datenbausteine

  Diese Bausteine enthalten Konstanten und Variablen für das STEP 7-Programm.

Anwenderspeicher

Alle Programme und Daten wie Teileprogramme, Unterprogramme, Kommentare, Werkzeugkorrekturen, Nullpunktverschiebungen/Frames sowie Kanal- und Programmanwenderdaten können in den gemeinsamen CNC-Anwenderspeicher abgelegt werden.

Arbeitsfeldbegrenzung

Mit der Arbeitsfeldbegrenzung kann der Verfahrbereich der Achsen zusätzlich zu den Endschaltern eingeschränkt werden. Je Achse ist ein Wertepaar zur Beschreibung des geschützten Arbeitsraumes möglich.

Arbeitsraum

Dreidimensionaler Raum, in den die Werkzeugspitze aufgrund der Konstruktion der Werkzeugmaschine hineinfahren kann. Siehe → Schutzraum.

Arbeitsspeicher

Der Arbeitsspeicher ist ein RAM-Speicher in der → CPU, auf den der Prozessor während der Programmbearbeitung auf das Anwenderprogramm zugreift.

Archivieren

Auslesen von Dateien und/oder Verzeichnissen auf ein **externes** Speichergerät.

Asynchrones Unterprogramm

Teileprogramm, das asynchron (unabhängig) zum aktuellen Programmzustand durch ein Interruptsignal (z. B. Signal "schneller NC-Eingang") gestartet werden kann.

Automatik

Betriebsart der Steuerung (Satzfolgebetrieb nach DIN): Betriebsart bei NC-Systemen, in der ein → Teileprogramm angewählt und kontinuierlich abgearbeitet wird.

Bahnachse

Bahnachsen sind alle Bearbeitungsachsen des → Kanals, die vom → Interpolator so geführt werden, dass sie gleichzeitig starten, beschleunigen, stoppen und den Endpunkt erreichen.

Bahngeschwindigkeit

Die maximal programmierbare Bahngeschwindigkeit ist abhängig von der Eingabefeinheit. Bei einer Auflösung von beispielsweise 0,1 mm beträgt die maximal programmierbare Bahngeschwindigkeit 1000 m/min.

Bahnsteuerbetrieb

Ziel des Bahnsteuerbetriebes ist es, ein größeres Abbremsen der → Bahnachsen an den Teileprogramm-Satzgrenzen zu vermeiden und mit möglichst gleicher Bahngeschwindigkeit in den nächsten Satz zu wechseln.

Bahnvorschub

Bahnvorschub wirkt auf → Bahnachsen. Er stellt die geometrische Summe der Vorschübe der beteiligten → Geometrieachsen dar.

Basisachse

Achse, deren Soll- oder Istwert für die Berechnung eines Kompensationswertes herangezogen wird.

Basiskoordinatensystem

Kartesisches Koordinatensystem, wird durch Transformation auf das Maschinenkoordinatensystem abgebildet.

Im → Teileprogramm verwendet der Programmierer Achsnamen des Basiskoordinatensystems. Es besteht, wenn keine → Transformation aktiv ist, parallel zum → Maschinenkoordinatensystem. Der Unterschied zu diesem liegt in den → Achsbezeichnern.

Baudrate

Geschwindigkeit bei der Datenübertragung (Bit/s).

Baustein

Als Bausteine werden alle Dateien bezeichnet, die für die Programmerstellung und Programmverarbeitung benötigt werden.

Bearbeitungskanal

Über eine Kanalstruktur können durch parallele Bewegungsabläufe Nebenzeiten verkürzt werden, z. B. Verfahren eines Ladeportals simultan zur Bearbeitung. Ein CNC-Kanal ist dabei als eigene CNC-Steuerung mit Dekodierung, Satzaufbereitung und Interpolation anzusehen.

Bedienoberfläche

Die Bedienoberfläche (BOF) ist das Anzeigemedium einer CNC-Steuerung in Gestalt eines Bildschirms. Sie ist mit horizontalen und vertikalen Softkeys gestaltet.

Beschleunigung mit Ruckbegrenzung

Zur Erzielung eines optimalen Beschleunigungsverhaltens an der Maschine bei gleichzeitiger Schonung der Mechanik kann im Bearbeitungsprogramm zwischen sprunghafter Beschleunigung und stetiger (ruckfreier) Beschleunigung umgeschaltet werden.

Betriebsart

Ablaufkonzept für den Betrieb einer SINUMERIK-Steuerung. Es sind die Betriebsarten → Jog, → MDA, → Automatik definiert.

Betriebsartengruppe

Technologisch zusammengehörige Achsen und Spindeln können zu einer Betriebsartengruppe (BAG) zusammengefasst werden. Achsen/Spindeln einer BAG können von einem oder mehreren → Kanälen gesteuert werden. Den Kanälen der BAG ist immer die gleiche → Betriebsart zugeordnet.

Bezeichner

Die Wörter nach DIN 66025 werden durch Bezeichner (Namen) für Variable (Rechenvariable, Systemvariable, Anwendervariable), für Unterprogramme, für Schlüsselwörter und Wörter mit mehreren Adressbuchstaben ergänzt. Diese Ergänzungen kommen in der Bedeutung den Wörtern beim Satzaufbau gleich. Bezeichner müssen eindeutig sein. Derselbe Bezeichner darf nicht für verschiedene Objekte verwendet werden.

Booten

Laden des Systemprogramms nach Power On.

C-Achse

Achse, um die eine gesteuerte Drehbewegung und Positionierung mit der Werkstückspindel erfolgt.

CNC

Siehe → NC

COM

Komponente der NC-Steuerung zur Durchführung und Koordination von Kommunikation.

CPU

Central Processing Unit, siehe → Speicherprogrammierbare Steuerung

C-Spline

Der C-Spline ist der bekannteste und am meisten verwendete Spline. Die Übergänge an den Stützpunkten sind tangenten- und krümmungsstetig. Es werden Polynome 3. Grades verwendet.

Datenbaustein

1. Dateneinheit der → PLC, auf die → HIGHSTEP-Programme zugreifen können.
2. Dateneinheit der → NC: Datenbausteine enthalten Datendefinitionen für globale Anwenderdaten. Die Daten können bei der Definition direkt initialisiert werden.

Datenübertragungsprogramm PCIN

PCIN ist ein Hilfsprogramm zum Senden und Empfangen von CNC-Anwenderdaten über die serielle Schnittstelle, wie z. B. Teileprogramme, Werkzeugkorrekturen etc. Das PCIN-Programm ist unter MS-DOS auf Standard-Industrie-PCs lauffähig.

Datenwort

Zwei Byte große Dateneinheit innerhalb eines → Datenbausteins.

Diagnose

1. Bedienbereich der Steuerung
2. Die Steuerung besitzt sowohl ein Selbstdiagnose-Programm als auch Testhilfen für den Service: Status-, Alarm- und Serviceanzeigen

DRF

Differential Resolver Function: NC-Funktion, die in Verbindung mit einem elektronischen Handrad eine inkrementale Nullpunktverschiebung im Automatik-Betrieb erzeugt.

Editor

Der Editor ermöglicht das Erstellen, Ändern, Ergänzen, Zusammenschieben und Einfügen von Programmen/Texten/Programmsätzen.

Eilgang

Schnellste Verfahrgeschwindigkeit einer Achse. Sie wird z. B. verwendet, wenn das Werkzeug aus einer Ruhestellung an die → Werkstückkontur herangefahren oder von der Werkstückkontur zurückgezogen wird. Die Eilganggeschwindigkeit wird maschinenspezifisch über Maschinendatum eingestellt.

Externe Nullpunktverschiebung

Von der → PLC vorgegebene Nullpunktverschiebung.

Fertigteilkontur

Kontur des fertig bearbeiteten Werkstücks. Siehe → Rohteil.

Festpunkt-Anfahren

Werkzeugmaschinen können feste Punkte wie Werkzeugwechselpunkt, Beladepunkt, Palettenwechselpunkt etc. definiert anfahren. Die Koordinaten dieser Punkte sind in der Steuerung hinterlegt. Die Steuerung verfährt die betroffenen Achsen, wenn möglich, im → Eilgang.

Frame

Ein Frame stellt eine Rechenvorschrift dar, die ein kartesisches Koordinatensystem in ein anderes kartesisches Koordinatensystem überführt. Ein Frame enthält die Komponenten → Nullpunktverschiebung, → Rotation, → Skalierung, → Spiegelung.

Führungsachse

Die Führungsachse ist die → Gantry–Achse, die aus Sicht des Bedieners und des Programmierers vorhanden und damit entsprechend wie eine normale NC-Achse beeinflussbar ist.

Genauhalt

Bei programmierter Genauhalt-Anweisung wird die in einem Satz angegebene Position genau und ggf. sehr langsam angefahren. Zur Reduktion der Annäherungszeit werden für Eilgang und Vorschub → Genauhaltsgrenzen definiert.

Genauhaltgrenze

Erreichen alle Bahnachsen ihre Genauhaltgrenze, so verhält sich die Steuerung als habe sie einen Zielpunkt exakt erreicht. Es erfolgt Satzweiterschaltung des → Teileprogramms.

Geometrie

Beschreibung eines → Werkstücks im → Werkstückkoordinatensystem.

Geometrieachse

Geometrieachsen dienen der Beschreibung eines 2- oder 3-dimensionalen Bereichs im Werkstückkoordinatensystem.

Geradeninterpolation

Das Werkzeug wird auf einer Geraden zum Zielpunkt verfahren und dabei das Werkstück bearbeitet.

Geschwindigkeitsführung

Um bei Verfahrbewegungen um sehr kleine Beträge je Satz eine akzeptable Verfahrgeschwindigkeit erreichen zu können, kann vorausschauende Auswertung über mehrere Sätze (→ Look Ahead) eingestellt werden.

Gewindebohren ohne Ausgleichsfutter

Mit dieser Funktion können Gewinde ohne Ausgleichsfutter gebohrt werden. Durch das interpolierende Verfahren der Spindel als Rundachse und der Bohrachse werden Gewinde exakt auf Endbohrtiefe geschnitten, z. B. Sacklochgewinde (Voraussetzung: Achsbetrieb der Spindel).

Gleichlaufachse

Die Gleichlaufachse ist die → Gantry-Achse, deren Sollposition stets von der Verfahrbewegung der → Führungsachse abgeleitet und damit synchron verfahren wird. Aus Sicht des Bedieners und des Programmierers ist die Gleichlaufachse "nicht vorhanden".

Grenzdrehzahl

Maximale/minimale (Spindel-)Drehzahl: Durch Vorgaben von Maschinendaten, der → PLC oder → Settingdaten kann die maximale Drehzahl einer Spindel begrenzt sein.

Hauptprogramm

Mit Nummer oder Bezeichner gekennzeichnetes → Teileprogramm, in dem weitere Hauptprogramme, Unterprogramme oder → Zyklen aufgerufen werden können.

Hauptsatz

Durch ":" eingeleiteter Satz, der alle Angaben enthält, um den Arbeitsablauf in einem → Teileprogramm starten zu können.

HIGHSTEP

Zusammenfassung der Programmiermöglichkeiten für die → PLC des Systems AS300/AS400.

Hilfsfunktionen

Mit Hilfsfunktionen können in → Teileprogrammen → Parameter an die → PLC übergeben werden, die dort vom Maschinenhersteller definierte Reaktionen auslösen.

Hochsprache CNC

Die Hochsprache bietet: → Anwenderdefinierte Variable, → Systemvariable, → Makrotechnik.

Interpolator

Logische Einheit des → NCK, die nach Angaben von Zielpositionen im Teileprogramm Zwischenwerte für die in den einzelnen Achsen zu fahrenden Bewegungen bestimmt.

Interpolatorische Kompensation

Mit Hilfe der interpolatorischen Kompensation können fertigungsbedingte **S**pindel**s**teigungs**f**ehler und **M**ess**s**ystem**f**ehler **k**ompensiert werden (SSFK, MSFK).

Interruptroutine

Interruptroutinen sind spezielle → Unterprogramme, die durch Ereignisse (externe Signale) vom Bearbeitungsprozess gestartet werden können. Ein in Abarbeitung befindlicher

Teileprogrammsatz wird abgebrochen, die Unterbrechungsposition der Achsen wird automatisch gespeichert.

JOG

Betriebsart der Steuerung (Einrichtebetrieb): In der Betriebsart JOG kann die Maschine eingerichtet werden. Einzelne Achsen und Spindeln können über die Richtungstasten im Tippbetrieb verfahren werden. Weitere Funktionen in der Betriebsart JOG sind das → Referenzpunktfahren, → Repos sowie → Preset (Istwert setzen).

Kanal

Ein Kanal ist dadurch gekennzeichnet, dass er unabhängig von anderen Kanälen ein → Teileprogramm abarbeiten kann. Ein Kanal steuert exklusiv die ihm zugeordneten Achsen und Spindeln. Teileprogrammabläufe verschiedener Kanäle können durch → Synchronisation koordiniert werden.

Kettenmaß

Auch Inkrementmaß: Angabe eines Bewegungsziels einer Achse durch eine zu verfahrende Wegstrecke und Richtung bezogen auf einen bereits erreichten Punkt. Siehe → Absolutmaß.

Kompensationsachse

Achse, deren Soll- oder Istwert durch den Kompensationswert modifiziert wird.

Kompensationstabelle

Tabelle von Stützpunkten. Sie liefert für ausgewählte Positionen der Basisachse die Kompensationswerte der Kompensationsachse.

Kompensationswert

Differenz zwischen der durch den Messgeber gemessenen Achsposition und der gewünschten, programmierten Achsposition.

Kontur

Umriss des → Werkstücks

Konturüberwachung

Als Maß für die Konturtreue wird der Schleppfehler innerhalb eines definierbaren Toleranzbandes überwacht. Ein unzulässig hoher Schleppfehler kann sich z. B. durch Überlastung des Antriebs ergeben. In diesem Fall kommt es zu einem Alarm und die Achsen werden stillgesetzt.

Koordinatensystem

Siehe → Maschinenkoordinatensystem, → Werkstückkoordinatensystem

Korrekturspeicher

Datenbereich in der Steuerung, in dem Werkzeugkorrekturdaten hinterlegt sind.

Kreisinterpolation

Das → Werkzeug soll zwischen festgelegten Punkten der Kontur mit einem gegebenen Vorschub auf einem Kreis fahren und dabei das Werkstück bearbeiten.

Krümmung

Die Krümmung k einer Kontur ist das Inverse des Radius r des anschmiegenden Kreises in einem Konturpunkt (k = 1/r).

KÜ

Übersetzungsverhältnis

KV

Kreisverstärkungsfaktor, regelungstechnische Größe eines Regelkreises

Ladespeicher

Der Ladespeicher ist bei der CPU 314 der → SPS gleich dem → Arbeitsspeicher.

Linearachse

Die Linearachse ist eine Achse, welche im Gegensatz zur Rundachse eine Gerade beschreibt.

Look Ahead

Mit der Funktion **Look Ahead** wird durch das "Vorausschauen" über eine parametrierbare Anzahl von Verfahrsätzen ein Optimum an Bearbeitungsgeschwindigkeit erzielt.

Losekompensation

Ausgleich einer mechanischen Maschinenlose, z. B. Umkehrlose bei Kugelrollspindeln. Für jede Achse kann die Losekompensation getrennt eingegeben werden.

Makrotechnik

Zusammenfassung einer Menge von Anweisungen unter einem Bezeichner. Der Bezeichner repräsentiert im Programm die Menge der zusammengefassten Anweisungen.

Maschinenachsen

In der Werkzeugmaschine physikalisch existierende Achsen.

Maschinenfestpunkt

Durch die Werkzeugmaschine eindeutig definierter Punkt, z. B. Maschinen-Referenzpunkt.

Maschinenkoordinatensystem

Koordinatensystem, das auf die Achsen der Werkzeugmaschine bezogen ist.

Maschinennullpunkt

Fester Punkt der Werkzeugmaschine, auf den sich alle (abgeleiteten) Messsysteme zurückführen lassen.

Maschinensteuertafel

Bedientafel der Werkzeugmaschine mit den Bedienelementen Tasten, Drehschalter usw. und einfachen Anzeigeelementen wie LEDs. Sie dient der unmittelbaren Beeinflussung der Werkzeugmaschine über die PLC.

Maßangabe metrisch und inch

Im Bearbeitungsprogramm können Positions- und Steigungswerte in inch programmiert werden. Unabhängig von der programmierbaren Maßangabe (`G70`/`G71`) wird die Steuerung auf ein Grundsystem eingestellt.

Masse

Als Masse gilt die Gesamtheit aller untereinander verbundenen inaktiven Teile eines Betriebsmittels, die auch im Fehlerfall keine gefährliche Berührungsspannung annehmen können.

MDA

Betriebsart der Steuerung: Manual Data Automatic. In der Betriebsart MDA können einzelne Programmsätze oder Satzfolgen ohne Bezug auf ein Haupt- oder Unterprogramm eingegeben und anschließend über die Taste NC-Start sofort ausgeführt werden.

Meldungen

Alle im Teileprogramm programmierten Meldungen und vom System erkannte → Alarme werden auf der Bedientafel im Klartext mit Datum und Uhrzeit und dem entsprechenden Symbol für das Löschkriterium angezeigt. Die Anzeige erfolgt getrennt nach Alarmen und Meldungen.

Metrisches Messsystem

Genormtes System von Einheiten: für Längen z. B. mm (Millimeter), m (Meter).

NC

Numerical Control: NC-Steuerung umfasst alle Komponenten der Werkzeugmaschinensteuerung: → NCK, → PLC, HMI, → COM.

Hinweis

Für die Steuerungen SINUMERIK 840D wäre CNC-Steuerung korrekter: Computerized Numerical Control.

NCK

Numerical Control Kernel: Komponente der NC-Steuerung, die → Teileprogramme abarbeitet und im Wesentlichen die Bewegungsvorgänge für die Werkzeugmaschine koordiniert.

Nebensatz

Durch "N" eingeleiteter Satz mit Informationen für einen Arbeitsschritt, z. B. eine Positionsangabe.

Netz

Ein Netz ist die Verbindung von mehreren S7-300 und weiteren Endgeräten, z. B. einem PG, über → Verbindungskabel. Über das Netz erfolgt ein Datenaustausch zwischen den angeschlossenen Geräten.

NRK

Numeric Robotic Kernel (Betriebssystem des → NCK)

Nullpunktverschiebung

Vorgabe eines neuen Bezugspunktes für ein Koordinatensystem durch Bezug auf einen bestehenden Nullpunkt und ein → Frame.

1. Einstellbar

   SINUMERIK 840D: Es steht eine projektierbare Anzahl von einstellbaren Nullpunktverschiebungen für jede CNC-Achse zur Verfügung. Die über G-Funktionen anwählbaren Verschiebungen sind alternativ wirksam.

2. Extern

   Zusätzlich zu allen Verschiebungen, die die Lage des Werkstücknullpunktes festlegen, kann eine externe Nullpunktverschiebung durch Handrad (DRF-Verschiebung) oder von der PLC überlagert werden.

3. Programmierbar

   Mit der Anweisung `TRANS` sind für alle Bahn- und Positionierachsen Nullpunktverschiebungen programmierbar.

NURBS

Die steuerungsinterne Bewegungsführung und Bahninterpolation wird auf Basis von NURBS (**N**on **U**niform **R**ational **B**-**S**plines) durchgeführt. Damit steht bei SINUMERIK 840D steuerungsintern für alle Interpolationen ein einheitliches Verfahren zur Verfügung.

OEM

Für Maschinenhersteller, die ihre eigene Bedienoberfläche erstellen oder technologiespezifische Funktionen in die Steuerung einbringen wollen, sind Freiräume für individuelle Lösungen (OEM-Applikationen) für SINUMERIK 840D vorgesehen.

Orientierter Spindelhalt

Halt der Werkstückspindel in vorgegebener Winkellage, z. B. um an bestimmter Stelle eine Zusatzbearbeitung vorzunehmen.

Orientierter Werkzeugrückzug

`RETTOOL`: Bei Bearbeitungsunterbrechungen (z. B. bei Werkzeugbruch) kann das Werkzeug per Programmbefehl mit vorgebbarer Orientierung um einen definierten Weg zurückgezogen werden.

Override

Manuelle bzw. programmierbare Eingriffsmöglichkeit, die es dem Bediener gestattet, programmierte Vorschübe oder Drehzahlen zu überlagern, um sie einem bestimmten Werkstück oder Werkstoff anzupassen.

Peripheriebaugruppe

Peripheriebaugruppen stellen die Verbindung zwischen CPU und Prozess her.

Peripheriebaugruppen sind:

- → Digital-Ein-/Ausgabebaugruppen
- → Analog-Ein-/Ausgabebaugruppen
- → Simulatorbaugruppen

PLC

Programmable **L**ogic **C**ontrol: → Speicherprogrammierbare Steuerung. Komponente der → NC: Anpass-Steuerung zur Bearbeitung der Kontroll-Logik der Werkzeugmaschine.

PLC-Programmierung

Die PLC wird mit der Software **STEP 7** programmiert. Die Programmiersoftware STEP 7 basiert auf dem Standardbetriebssystem **WINDOWS** und enthält die Funktionen der STEP 5 -Programmierung mit innovativen Weiterentwicklungen.

PLC-Programmspeicher

SINUMERIK 840D: Im PLC-Anwenderspeicher werden das PLC-Anwenderprogramm und die Anwenderdaten gemeinsam mit dem PLC-Grundprogramm abgelegt.

Polarkoordinaten

Koordinatensystem, das die Lage eines Punktes in einer Ebene durch seinen Abstand vom Nullpunkt und den Winkel festlegt, den der Radiusvektor mit einer festgelegten Achse bildet.

Polynom-Interpolation

Mit der Polynom-Interpolation können die unterschiedlichsten Kurvenverläufe erzeugt werden, wie **Gerade-, Parabel-, Potenzfunktionen** (SINUMERIK 840D).

Positionierachse

Achse, die eine Hilfsbewegung an einer Werkzeugmaschine ausführt. (z. B. Werkzeugmagazin, Palettentransport). Positionierachsen sind Achsen, die nicht mit den → Bahnachsen interpolieren.

Programmbaustein

Programmbausteine enthalten die Haupt- und Unterprogramme der → Teileprogramme.

Programmierbare Arbeitsfeldbegrenzung

Begrenzung des Bewegungsraumes des Werkzeugs auf einen durch programmierte Begrenzungen definierten Raum.

Programmierbare Frames

Mit programmierbaren → Frames können dynamisch im Zuge der Teileprogramm-Abarbeitung neue Koordinatensystem-Ausgangspunkte definiert werden. Es wird unterschieden nach absoluter Festlegung anhand eines neuen Frames und additiver Festlegung unter Bezug auf einen bestehenden Ausgangspunkt.

Programmierschlüssel

Zeichen und Zeichenfolgen, die in der Programmiersprache für → Teileprogramme eine festgelegte Bedeutung haben.

Pufferbatterie

Die Pufferbatterie gewährleistet, dass das → Anwenderprogramm in der → CPU netzausfallsicher hinterlegt ist und festgelegte Datenbereiche und Merker, Zeiten und Zähler remanent gehalten werden.

Quadrantenfehlerkompensation

Konturfehler an Quadrantenübergängen, die durch wechselnde Reibverhältnisse an Führungsbahnen entstehen, sind mit der Quadrantenfehlerkompensation weitgehend eliminierbar. Die Parametrierung der Quadrantenfehlerkompensation erfolgt durch einen Kreisformtest.

Referenzpunkt

Punkt der Werkzeugmaschine, auf den sich das Messsystem der → Maschinenachsen bezieht.

Rohteil

Teil, mit dem die Bearbeitung eines Werkstücks begonnen wird.

Rotation

Komponente eines → Frames, die eine Drehung des Koordinatensystems um einen bestimmten Winkel definiert.

R-Parameter

Rechenparameter, kann vom Programmierer des → Teileprogramms für beliebige Zwecke im Programm gesetzt oder abgefragt werden.

Rundachse

Rundachsen bewirken eine Werkstück- oder Werkzeugdrehung in eine vorgegebene Winkellage.

Rundungsachse

Rundungsachsen bewirken eine Werkstück- oder Werkzeugdrehung in eine einem Teilungsraster entsprechende Winkellage. Beim Erreichen eines Rasters ist die Rundungsachse "in Position".

Satzsuchlauf

Zum Austesten von Teileprogrammen oder nach einem Abbruch der Bearbeitung kann über die Funktion "Satzsuchlauf" eine beliebige Stelle im Teileprogramm angewählt werden, an der die Bearbeitung gestartet oder fortgesetzt werden soll.

Schlüsselschalter

Der Schlüsselschalter auf der → Maschinensteuertafel besitzt 4 Stellungen, die vom Betriebssystem der Steuerung mit Funktionen belegt sind. Zum Schlüsselschalter gehören drei verschiedenfarbige Schlüssel, die in den angegebenen Stellungen abgezogen werden können.

Schlüsselwörter

Wörter mit festgelegter Schreibweise, die in der Programmiersprache für → Teileprogramme eine definierte Bedeutung haben.

Schneidenradiuskorrektur

Bei der Programmierung einer Kontur wird von einem spitzen Werkzeug ausgegangen. Da dies in der Praxis nicht realisierbar ist, wird der Krümmungsradius des eingesetzten Werkzeugs der Steuerung angegeben und von dieser berücksichtigt. Dabei wird der Krümmungsmittelpunkt um den Krümmungsradius verschoben äquidistant um die Kontur geführt.

Schnellabheben von der Kontur

Beim Eintreffen eines Interrupts kann über das CNC-Bearbeitungsprogramm eine Bewegung eingeleitet werden, die ein schnelles Abheben des Werkzeugs von der gerade bearbeiteten Werkstückkontur ermöglicht. Zusätzlich kann der Rückzugwinkel und der Betrag des Weges parametriert werden. Nach dem Schnellabheben kann zusätzlich eine Interruptroutine ausgeführt werden (SINUMERIK 840D).

Schnelle digitale Ein-/Ausgänge

Über die digitalen Eingänge können z. B. schnelle CNC-Programmroutinen (Interruptroutinen) gestartet werden. Über die digitalen CNC-Ausgänge können schnelle, programmgesteuerte Schaltfunktionen ausgelöst werden (SINUMERIK 840D).

Schrägenbearbeitung

Bohr- und Fräsbearbeitungen an Werkstückflächen, die nicht in den Koordinatenebenen der Maschine liegen, können mit Unterstützung der Funktion "Schrägenbearbeitung" komfortabel ausgeführt werden.

Schraubenlinien-Interpolation

Die Schraubenlinien-Interpolation eignet sich besonders zum einfachen Herstellen von Innen- oder Außengewinden mit Formfräsern und zum Fräsen von Schmiernuten.

Dabei setzt sich die Schraubenlinie aus zwei Bewegungen zusammen:

- Kreisbewegung in einer Ebene
- Linearbewegung senkrecht zu dieser Ebene

Schrittmaß

Verfahrweglängenangabe über Inkrementanzahl (Schrittmaß). Inkrementanzahl kann als → Settingdatum hinterlegt sein bzw. durch entsprechend beschriftete Tasten 10, 100, 1000, 10000 gewählt werden.

Schutzraum

Dreidimensionaler Raum innerhalb des → Arbeitsraumes, in den die Werkzeugspitze nicht hineinreichen darf.

Serielle Schnittstelle V.24

Für die Dateneingabe/-ausgabe ist auf der PCU 20 eine serielle V.24-Schnittstelle (RS232), auf der PCU 50/70 sind zwei V.24-Schnittstellen vorhanden. Über diese Schnittstellen können Bearbeitungsprogramme sowie Hersteller- und Anwenderdaten geladen und gesichert werden.

Settingdaten

Daten, die Eigenschaften der Werkzeugmaschine auf durch die Systemsoftware definierte Weise der NC-Steuerung mitteilen.

Sicherheitsfunktionen

Die Steuerung enthält ständig aktive Überwachungen, die Störungen in der → CNC, der Anpass-Steuerung (→ PLC) und der Maschine so frühzeitig erkennen, dass Schäden an Werkstück, Werkzeug oder Maschine weitgehend ausgeschlossen werden. Im Störungsfall wird der Bearbeitungsablauf unterbrochen und die Antriebe werden stillgesetzt, die Störungsursache gespeichert und als Alarm angezeigt. Gleichzeitig wird der PLC mitgeteilt, dass ein CNC-Alarm ansteht.

Skalierung

Komponente eines → Frames, die achsspezifische Maßstabsveränderungen bewirkt.

Softkey

Taste, deren Beschriftung durch ein Feld im Bildschirm repräsentiert wird, das sich dynamisch der aktuellen Bediensituation anpasst. Die frei belegbaren Funktionstasten (Softkeys) werden softwaremäßig definierten Funktionen zugeordnet.

Software-Endschalter

Software-Endschalter begrenzen den Verfahrbereich einer Achse und verhindern ein Auffahren des Schlittens auf die Hardware-Endschalter. Je Achse sind 2 Wertepaare vorgebbar, die getrennt über die → PLC aktiviert werden können.

Speicherprogrammierbare Steuerung

Speicherprogrammierbare Steuerungen (SPS) sind elektronische Steuerungen, deren Funktion als Programm im Steuerungsgerät gespeichert ist. Aufbau und Verdrahtung des Gerätes hängen also nicht von der Funktion der Steuerung ab. Die speicherprogrammierbare Steuerung hat die Struktur eines Rechners; sie besteht aus CPU (Zentralbaugruppe) mit Speicher, Ein-/Ausgabebaugruppen und internem Bus-System. Die Peripherie und die Programmiersprache sind auf die Belange der Steuerungstechnik ausgerichtet.

Spiegelung

Bei Spiegelung werden die Vorzeichen der Koordinatenwerte einer Kontur bezüglich einer Achse vertauscht. Es kann bezüglich mehrerer Achsen zugleich gespiegelt werden.

Spindelsteigungsfehler-Kompensation

Ausgleich mechanischer Ungenauigkeiten einer am Vorschub beteiligten Kugelrollspindel durch die Steuerung anhand von hinterlegten Messwerten der Abweichungen.

Spline-Interpolation

Mit der Spline-Interpolation kann die Steuerung aus nur wenigen vorgegebenen Stützpunkten einer Sollkontur einen glatten Kurvenverlauf erzeugen.

Standardzyklen

Für häufig wiederkehrende Bearbeitungsaufgaben stehen Standardzyklen zur Verfügung:

- für die Technologie Bohren/Fräsen
- für die Technologie Drehen

Im Bedienbereich "Programm" werden unter dem Menü "Zyklenunterstützung" die zur Verfügung stehenden Zyklen aufgelistet. Nach Anwahl des gewünschten Bearbeitungszyklus werden die notwendigen Parameter für die Wertzuweisung im Klartext angezeigt.

Synchronachsen

Synchronachsen benötigen für ihren Weg die gleiche Zeit wie die Geometrieachsen für ihren Bahnweg.

Synchronaktionen

1. Hilfsfunktionsausgabe

   Während der Werkstückbearbeitung können aus dem CNC-Programm heraus technologische Funktionen (→ Hilfsfunktionen) an die PLC ausgegeben werden. Über diese Hilfsfunktionen werden beispielsweise Zusatzeinrichtungen der Werkzeugmaschine gesteuert, wie Pinole, Greifer, Spannfutter etc.

2. Schnelle Hilfsfunktionsausgabe

   Für zeitkritische Schaltfunktionen können die Quittierungszeiten für die → Hilfsfunktionen minimiert und unnötige Haltepunkte im Bearbeitungsprozess vermieden werden.

Synchronisation

Anweisungen in → Teileprogrammen zur Koordination der Abläufe in verschiedenen → Kanälen an bestimmten Bearbeitungsstellen.

Systemspeicher

Der Systemspeicher ist ein Speicher in der CPU, in der folgende Daten abgelegt werden:

- Daten, die das Betriebssystem benötigt
- die Operanden Zeiten, Zähler, Merker

Systemvariable

Ohne Zutun des Programmierers eines → Teileprogramms existierende Variable. Sie ist definiert durch einen Datentyp und dem Variablennamen, der durch das Zeichen **$** eingeleitet wird. Siehe → Anwenderdefinierte Variable.

Teileprogramm

Folge von Anweisungen an die NC-Steuerung, die insgesamt die Erzeugung eines bestimmten → Werkstücks bewirken. Ebenso Vornahme einer bestimmten Bearbeitung an einem gegebenen → Rohteil.

Teileprogrammsatz

Teil eines → Teileprogramms, durch Line Feed abgegrenzt. Es werden → Hauptsätze und → Nebensätze unterschieden.

Teileprogrammverwaltung

Die Teileprogrammverwaltung kann nach → Werkstücken organisiert werden. Die Größe des Anwenderspeichers bestimmt die Anzahl der zu verwaltenden Programme und Daten. Jede Datei (Programme und Daten) kann mit einem Namen von maximal 24 alphanumerischen Zeichen versehen werden.

Text-Editor

Siehe → Editor

TOA–Bereich

Der TOA–Bereich umfasst alle Werkzeug- und Magazindaten. Standardmäßig fällt der Bereich bzgl. der Reichweite der Daten mit dem Bereich → Kanal zusammen. Über Maschinendaten kann jedoch festgelegt werden, dass sich mehrere Kanäle eine → TOA-Einheit teilen, so dass diesen Kanälen dann gemeinsame WZV-Daten zur Verfügung stehen.

TOA–Einheit

Jeder → TOA-Bereich kann mehrere TOA-Einheiten enthalten. Die Anzahl der möglichen TOA-Einheiten wird über die maximale Anzahl aktiver → Kanäle begrenzt. Eine TOA-Einheit umfasst genau einen WZ-Daten-Baustein und einen Magazindaten-Baustein. Zusätzlich kann noch ein WZ-Trägerdaten-Baustein enthalten sein (optional).

Transformation

Additive oder absolute Nullpunktsverschiebung einer Achse.

Unterprogramm

Folge von Anweisungen eines → Teileprogramms, die mit unterschiedlichen Versorgungsparametern wiederholt aufgerufen werden kann. Der Aufruf des Unterprogramms erfolgt aus einem Hauptprogramm. Jedes Unterprogramm kann gegen nicht autorisiertes Auslesen und Anzeigen gesperrt werden. → Zyklen sind eine Form von Unterprogrammen.

Urlöschen

Beim Urlöschen werden folgende Speicher der → CPU gelöscht:

- → Arbeitsspeicher
- Schreib-/Lesebereich des → Ladespeichers
- → Systemspeicher
- → Backup-Speicher

Variablendefinition

Eine Variablendefinition umfasst die Festlegung eines Datentyps und eines Variablennamens. Mit dem Variablennamen kann der Wert der Variablen angesprochen werden.

Verbindungskabel

Verbindungskabel sind vorgefertigte bzw. vom Anwender selbst anzufertigende 2-Draht-Leitungen mit 2 Anschlussstecker. Diese Verbindungskabel verbinden die → CPU über die → Mehrpunkt-Schnittstelle (MPI) mit einem → PG bzw. mit anderen CPUs.

Verfahrbereich

Der maximal zulässige Verfahrbereich bei Linearachsen beträgt ± 9 Dekaden. Der absolute Wert ist abhängig von der gewählten Eingabe- und Lageregelfeinheit und dem Einheitensystem (inch oder metrisch).

Vorkoinzidenz

Satzwechsel bereits, wenn Bahnweg um ein vorgegebenes Delta der Endposition nahe gekommen ist.

Vorschub-Override

Der programmierten Geschwindigkeit wird die aktuelle Geschwindigkeitseinstellung über → Maschinensteuertafel oder von der → PLC überlagert (0-200%). Die Vorschubgeschwindigkeit kann zusätzlich im Bearbeitungsprogramm durch einen programmierbaren Prozentfaktor (1-200%) korrigiert werden.

Vorsteuerung, dynamisch

Ungenauigkeiten der → Kontur, bedingt durch Schleppfehler, lassen sich durch die dynamische, beschleunigungsabhängige Vorsteuerung nahezu eliminieren. Dadurch ergibt sich auch bei hohen → Bahngeschwindigkeiten eine hervorragende Bearbeitungsgenauigkeit. Die Vorsteuerung kann achsspezifisch über das → Teileprogramm an- und abgewählt werden.

Weg-Zeit-Nocken

Unter einem Weg-Zeit-Nocken wird ein Softwarenockenpaar verstanden, das an einer definierten Achsposition einen Impuls einer bestimmten Dauer bereitstellen kann.

Werkstück

Von der Werkzeugmaschine zu erstellendes/zu bearbeitendes Teil.

Werkstückkontur

Sollkontur des zu erstellenden/bearbeitenden → Werkstücks.

Werkstückkoordinatensystem

Das Werkstückkoordinatensystem hat seinen Ausgangspunkt im → Werkstücknullpunkt. Bei Programmierung im Werkstückkoordinatensystem beziehen sich Maße und Richtungen auf dieses System.

Werkstücknullpunkt

Der Werkstücknullpunkt bildet den Ausgangspunkt für das → Werkstückkoordinatensystem. Er ist durch Abstände zum → Maschinennullpunkt definiert.

Werkzeug

An der Werkzeugmaschine wirksames Teil, das die Bearbeitung bewirkt (z. B. Drehmeißel, Fräser, Bohrer, LASER-Strahl ...).

Werkzeugkorrektur

Berücksichtigung der Werkzeug-Abmessungen bei der Berechnung der Bahn.

Werkzeugradiuskorrektur

Um eine gewünschte → Werkstückkontur direkt programmieren zu können, muss die Steuerung unter Berücksichtigung des Radius des eingesetzten Werkzeugs eine äquidistante Bahn zur programmierten Kontur verfahren (G41/G42).

WinSCP

WinSCP ist ein frei verfügbares Open Source-Programm für Windows zum Transferieren von Dateien.

Zeitreziproker Vorschub

Bei SINUMERIK 840D kann anstelle der Vorschubgeschwindigkeit für die Achsbewegung die Zeit programmiert werden, die der Bahnweg eines Satzes benötigen soll (G93).

Zoll-Maßsystem

Maßsystem, das Entfernungen in "inch" und Bruchteilen davon definiert.

Zwischensätze

Verfahrbewegungen mit angewählter → Werkzeugkorrektur (G41/G42) dürfen durch eine begrenzte Anzahl Zwischensätze (Sätze ohne Achsbewegungen in der Korrekturebene) unterbrochen werden, wobei die Werkzeugkorrektur noch korrekt verrechnet werden kann. Die zulässige Anzahl Zwischensätze, die die Steuerung vorausliest, ist über Systemparameter einstellbar.

Zyklen

Geschützte Unterprogramme zur Ausführung von wiederholt auftretenden Bearbeitungsvorgängen am → Werkstück.

Index

$

$A_DPx_IN, 61
$A_DPx_OUT, 61
$A_OUT[n], 708
$AA_ACT_INDEX_AX_POS_NO, 949
$AA_COUP_ACT, 897
$AA_COUP_OFFS, 897
$AA_ENC_COMP, 361
$AA_ENC_COMP_IS_MODULO, 362
$AA_ENC_COMP_MAX, 362
$AA_ENC_COMP_MIN, 361
$AA_ENC_COMP_STEP, 361
$AA_FIX_POINT_ACT, 335
$AA_FIX_POINT_SELECTED, 335
$AA_G0MODE, 773
$AA_PROG_INDEX_AX_POS_NO, 949
$AC_AXCTSWA[ ], 156
$AN_AXCTAS[ ], 156
$AN_AXCTSWA[ ], 156
$AN_CEC, 369
$AN_CEC_DIRECTION, 370
$AN_CEC_INPUT_AXIS, 369
$AN_CEC_IS_MODULO, 371
$AN_CEC_MAX, 370
$AN_CEC_MIN, 369
$AN_CEC_MULT_BY_TABLE, 371
$AN_CEC_OUTPUT_AXIS, 369
$AN_CEC_STEP, 369
$AN_REBOOT_DELAY_TIME, 454
$P_COUP_OFFS, 898
$VA_COUP_OFFS, 897

1

1-dimensionale
    Sollwertvorgabe ($AC_MEAS_TYPE = 19), 655

2

2-dimensionale
    Sollwertvorgabe ($AC_MEAS_TYPE = 20), 657

3

3-dimensionale
    Sollwertvorgabe ($AC_MEAS_TYPE = 21), 658
3D-Messtaster, 622

A

Abrichten während der Bearbeitung, 991
Achs
    -interpolator, 771
Achs-/Spindeltausch, 460, 470, 1062
Achsabbildung, 543
Achscontainer, 101, 102, 155, 391
Achscontainer-Drehung, 158
Achscontainer-Drehung-Aktiv, 1035
Achscontainer-Namen, 156
Achse
    Basis-, 355
    Kompensations-, 355
    lokale, 101, 102
    physikalische, 101
    werkstückhaltende, 101
    werkstücktragende, 102
Achse wird vom NCK als Positionierachse behandelt, 1074
Achse/Spindel weiterfahren, 1073
Achse/Spindel zurücksetzen, 1073
Achse-Betriebsbereit, 1035
Achsen
    für Hilfsbewegungen, 763
Achskonfiguration, 525, 528, 540, 542, 550
Achskontrolle an dem PLC übergeben, 1074
Achstausch, 470
    Abgabezeitpunkt, 483
    Achs-Containerdrehung freigeben, 484
    Achse in anderem Kanal, 471
    Achstausch über Synchronaktionen, 489
    Achstypen, 470
    automatisch erzeugtes GET/GETD, 478
    Geometrieachse im gedrehten WKS, 488
    ohne Vorlaufstopp, 485
    Voraussetzungen, 471
Achstausch durch PLC, 800
Achstypen
    bei Positionierachsen, 768

ACN, 858
ACP, 858
Aktive Zustellachsen, 1079
Aktiver/passiver Bedienmodus, 1032
Aktiver/passiver Bedienmodus von Bedieneinheit, 1032
Aktives Filesystem, 926
Aktivierung, 561
Alarm-
Textdatei, 92
Texthaltung, 92
Alarme, 94
Alarme,
Meldungen, 92
Alle Transformationen, 580
Allgemeine Kanalkonfiguration, 591
Allgemeines, 606
AMIRROR(C), 868
An- und Abwahl, 553
Analoge Peripherie|840Di, 29
Analogwertdarstellung, 52
Ankratzen, 615
Anschluss an 840D, 608
Anschluss an SINUMERIK 840D sl, 609
Anschluss an SINUMERIK 840Di, 609
Antriebssteuerung, 101
Anwahl, 562
Anwahl und Abwahl, 547
Anwender-
Alarm, 92
Anwenderdefinierte Kopplung, 877
Anwenderkommunikation, 81
Anwendungen, 550, 941
Anzahl Schräge Achse, 541
Anzahl Transformation, 503
Anzahl Transformationen, 526, 551
Anzahl verketteter Transformationen, 551
Arbeitsfeldbegrenzung, 328, 868
Art der Transformation, 543
Art derTransformation, 530
ASCALE, 865
ATRANS, 865
Aufgabenstellung, 539
Aufgabenstellung, 524
Aufspannachse/Spindel, 102
Ausfeuerhübe, 809
Ausfeuern aktiv, 1078
Ausgangswerte, 616
Ausschlüsse, 548
Auswahl Werkzeug oder Schneide, 622
Auswirkungen auf die Bedienung HMI, 554
Autarke Einzelachsvorgänge
Aktionen der PLC, 801
Reaktionen des NCK, 802
Automatisch aktivierbare Vorauslösezeit, 730
Automatischer Achstausch mit GETD, 478
autonome Maschine, 97
AxAlarm, 1073
Axialer Reset wurde durchgeführt, 1074
Axialer Stoppalarm für diese Achse, 1073
AxReset, 1073
AXRESET DONE, 1074
AxResume, 1073
AxStop aktiv, 1074
AXTOCHAN, 489

B

Bahn
-interpolator, 771
Bahnachsen
Bahninterpolator, 788
Bearbeitung
stirnseitig, 497
Bedien-
Bereich, 91, 96
Bild, 95, 96
einheit, 79
Einheit, 86, 89, 94
Bedienbereich Verbinden, 95
Bedieneinheit fordert aktiven Bedienmodus, 1032
Bedieneinheiten-Umschaltsperre, 1031
Bedienmodus-Umschaltung, 124
Bedienmoduswechsel abgelehnt, 1033
Bedientafel, 79
Bedientafelfront, 86
Bedientafel-Schnittstelle (BTSS), 86
Bedingtes Warten
aus den Stillstand auf Bahngeschwindigkeit beschleunigen, 466
im Bahnsteuerbetrieb, 465
in der Bremsrampe abhängig vom IPO-Takt, 466
kein Warten, 466
Befehle MEAS, MEAW, 613
Beispiel, 555
eine Bedientafel:drei NCUs, 221
Kont. Messen nach Abschluss der prog. Verfahrbewegung, 687
Kontinuierliches Messen mit Restweglöschen, 688
Kontinuierliches Messen modal über mehrere Sätze, 688
Messmodus 1, 686
Messmodus 2, 687
TRAANG, 590
TRACYL, 585
TRANSMIT, 583

zwei Bedientafeln:eine NCU, 221
Beispiel für die Funktionsprüfung, 689
Benennung der Geometrieachsen, 504, 528
Berechneter Frame, 620
Berechnungsmethode, 625
Beschleunigung, 286, 298, 328
Beschleunigungskennlinie, 731
Besonderheiten, 563
Besonderheiten bei Jog, 538
Betriebsart JOG, 284
Betriebsarten, 330
Betriebsartengruppe, 459, 461
Bewegungsendekriterium
bei Satzsuchlauf, 798
Bremsrampe IPOBRKA, 793
FINEA, COARSEA, IPOENDA, 792
Bewegungsführung, 101
Bewegungsverhalten, 805
Bezugssysteme auswählen, 569
BHG, 90
Bidirektionaler Messtaster, 607
BTSS, 86, 90
BTSS, Netzwerkregeln, 97
Bus-
System, 85, 86
Teilnehmer, 85, 87, 90

C

CC-Bindings, 64
CLEAR, 463
CLEARM, 463
COARSA, 792
Codierte Positionen, 944
conn_1, 91
Containerachsen, 155
Container-Link-Achsen, 155
Continuous Dressing, 991
COROS-OP, 87

D

Daten-
Austausch, 90
Sicherung über V.24, 91
Datenkonsistenz, 61
Datensicherung
über V.24, 95
Datenzugriff, 59
lesender Zugriff, 59
schreibender Zugriff, 59
Dauerbetrieb, 942
DB10
DBB0, 35, 1016
DBB1, 35, 36, 1016
DBB100, 1037
DBB100 ff, 296
DBB101, 1037
DBB102, 1037
DBB122, 35, 1016
DBB123, 35, 36, 1016
DBB124, 1016
DBB125, 1016
DBB126, 1016
DBB127, 1016
DBB128, 1016
DBB129, 1016
DBB130, 38, 1016
DBB131, 38, 1017
DBB132, 38, 1017
DBB133, 38, 1018
DBB134, 1016
DBB135, 1017
DBB136, 1017
DBB137, 1018
DBB138, 1016
DBB139, 1017
DBB140, 1017
DBB141, 1018
DBB142, 1016
DBB143, 1017
DBB144, 1017
DBB145, 1018
DBB146, 43, 1018
DBB147, 43, 1018
DBB148-163, 43, 1019
DBB166, 46, 1019
DBB167, 46, 1019
DBB168, 46, 1020
DBB170-185, 46, 1020
DBB186, 39
DBB186-189, 1023
DBB190-193, 1023
DBB194-209, 43, 1023
DBB210-225, 47, 1024
DBB4, 38, 1016
DBB5, 38, 1017
DBB6, 38, 1017
DBB60, 1023
DBB64, 39, 1023
DBB7, 38, 1018
DBB97, 1036
DBB98, 1036
DBB99, 1036
DBX100.6, 1038

DBX100.6 ff, 296
DBX100.7, 1038
DBX100.7 ff, 296
DBX101.6, 1038
DBX101.7, 1038
DBX102.6, 1038
DBX102.7, 1038
DBX107.0, 1064
DBX107.1, 1064
DBX107.6, 1034
DBX108.7, 453
DBX110.0 - 113.7, 706, 707
DBX110.0-113.7, 1065
DBX114.0 - 117.7, 707
DBX114.0-117.7, 1066
DBX97 ff, 296
DB11
DBX 6.3, 453
DBX4.2, 284
DB21, ...
DBB12/16/20, 309
DBB12-23, 326
DBX 12.0 - 12.2 ff, 296
DBX0.3, 311, 316, 1040
DBX0.6, 311
DBX100.5, 1052
DBX101.5, 1052
DBX102.5, 1052
DBX12.0-2, 1040
DBX12.0-5, 1043
DBX12.3, 723
DBX12.4, 1041
DBX12.5, 1041
DBX12.6, 1042
DBX12.7, 1042
DBX12.7 bzw. DBX12.6ff, 1041
DBX13.6, 1043
DBX16.0-2, 1040
DBX16.0-5, 1043
DBX16.4, 1041
DBX16.5, 1041
DBX16.6, 1042
DBX16.7, 1042
DBX17.6, 1043
DBX20.0-2, 1040
DBX20.0-5, 1043
DBX20.4, 1041
DBX20.5, 1041
DBX20.6, 1042
DBX20.7, 1042
DBX21.6, 1043
DBX24.3, 316, 1045
DBX3.0, 723, 1068
DBX3.1, 723, 1069
DBX3.2, 723, 1069
DBX3.3, 1069
DBX3.4, 723, 1070
DBX3.5, 1070
DBX30.0, 1050
DBX30.0, 30.1, 30.2, 313
DBX30.1, 1050
DBX30.2, 1050
DBX30.3, 314, 1051
DBX30.4, 314, 1051
DBX33.3, 308, 1045
DBX33.6, 1063
DBX332.4, 1048
DBX332.5, 1048
DBX332.6, 1049
DBX332.7, 1049
DBX336.4, 1048
DBX336.5, 1048
DBX336.6, 1049
DBX336.7, 1049
DBX340.4, 1048
DBX340.5, 1048
DBX340.6, 1049
DBX340.7, 1049
DBX37.0, 1051
DBX37.0-2, 1046
DBX37.1, 1051
DBX37.2, 1051
DBX38.0, 724, 1070
DBX38.1, 723, 1071
DBX40.0-2, 1046
DBX40.4, 1046
DBX40.5, 1046
DBX40.6, 1047
DBX40.7, 289, 295, 1047
DBX41.0-6, 1048
DBX46.0-2, 1046
DBX46.4, 1046
DBX46.5, 1046
DBX46.6, 1047
DBX46.7, 1047
DBX47.0-6, 1048
DBX52.0-2, 1046
DBX52.4, 1046
DBX52.5, 1046
DBX52.6, 1047
DBX52.7, 1047
DBX53.0-6, 1048
DB21, …
DBX 36.5, 453
DBX13.6,ff, 289
DBX40.6, 289, 295

DBX41.6,ff, 289
DB31
DBX2.0, 706
DBX62.0, 706
DB31, ...
DBB0, 1072
DBB5 ff., 292
DBB68, 1062
DBB78-81, 1075
DBB8, 1062
DBX1.7, 285
DBX100.2, 1077
DBX100.3, 1078
DBX100.4, 1078
DBX100.5, 1078
DBX100.6, 1078
DBX100.7, 1078
DBX104.0 - 7, 1079
DBX12.4, 852, 861, 1079
DBX13.0-2, 332, 333, 1057
DBX16.7, 328
DBX17.6, 328
DBX18.5, 328
DBX18.7 bzw. DBX18.6, 328
DBX2.0, 707, 1066
DBX2.2, 293, 297, 306, 307, 1072
DBX28.0, 1076
DBX28.1, 777, 1073
DBX28.2, 777, 1073
DBX28.3, 1076
DBX28.4, 1076
DBX28.5, 1076
DBX28.6, 776, 1077
DBX28.7, 774, 1077
DBX29.5, 905
DBX31.4, 906
DBX31.5, 1080
DBX4.0 - DBX4.2, 296
DBX4.0 bis DBX 4.4, 307
DBX4.0-2, 1053
DBX4.4, 1053
DBX4.5, 1054
DBX4.6, 1055
DBX4.7, 1055
DBX5.0-5, 1056
DBX5.6, 288, 1056
DBX6.2, 776, 777
DBX60.1, 1035
DBX60.4, 942, 943
DBX60.4/5, 347
DBX60.5, 942, 943
DBX60.6, 778
DBX60.7, 778
DBX60.7 bzw. DBX60.6, 328
DBX61.1, 1035, 1073
DBX61.2, 1035
DBX61.3, 905
DBX62.0, 707, 1067
DBX62.1, 308, 311, 1058
DBX62.3, 1064
DBX63.0, 776, 780, 1074
DBX63.1, 776, 1074
DBX63.2, 776, 778, 780, 1074
DBX64.0-2, 1058
DBX64.4, 1059
DBX64.5, 1059
DBX64.6, 778, 1059
DBX64.6 und 64.7, 308
DBX64.7, 778, 1059
DBX65.0-6, 1060
DBX65.6, 288
DBX74.4, 852, 861, 1079
DBX75.0-2, 332, 1061
DBX75.3-5, 332, 1061
DBX76.5, 1074
DBX76.6, 942, 943, 945, 1083
DBX83.3, 1084
DBX83.5, 328
DBX83.6, 1085
DBX83.7, 328
DBX84.1, 1085
DBX84.4, 1080
DBX98.0, 1081
DBX98.1, 1081
DBX98.2, 1081
DBX98.4, 1082
DBX99.0, 1082
DBX99.1, 1082
DBX99.4, 906
DB31, …
DBX 61.2, 453
Determinismus, 925
Differenzdrehzahl, 913
Digitale Eingänge des NCK, 35
Digitale Peripherie|840Di, 29
Direktverbindung, 92
Doppeladressierung, 90
Drehlage, 532
Drehlage, 507
Drehsinn, 508, 532
Drehzahlüberwachung, 1004, 1085
DRF, 283, 315, 962
DRF aktivieren, 1040
Durchhangkompensation, 389
Dynamik
-anpassung, 405

Dynamikprogrammierung im Spindel-/Achsbetrieb, 919
dynamische
HMI-Eigenschaft, 89
Umschaltung, 81
Dynamischer Anwenderspeicher, 932
Dynamischer NC-Speicher, 925

E

E/A-Bereich, 58
Konfiguration, 59
konfigurierte Länge, 58
Startadresse, 58
Systemvariable, 62
Ebeneneinstellung, 619
Ecke C1 - C4 ($AC_MEAS_TYPE = 4, 5, 6, 7), 634
Eckmessung C1, 635
Eigener Folgespindelinterpolator, 877
Eigenschaften, 577
Eilgang
Interpolationsarten, 771
Eilgangkorrektur, 783
Eilgangüberlagerung, 285
Eingangswerte, 616
Messtypen, 617
Sollwerte, 619
Eingangüberlagerung für Geometrieachse (1, 2, 3), 1041
Einschalten der Kopplung, 882
Einschalt-Varianten, 883
Eintrag der Kanalachsen, 505, 529
Einzelachsen
Achskontrolle durch PLC, 774
Alarmbehandlung, 777
Anwendungsfälle, 776
Erweitertes Rückziehen NC-geführt, 781
Erweitertes Stillsetzen NC-geführt, 781
Einzelsatz
Positionierachse Typ 1, 804
Positionierachse Typ 2, 804
Positionierachse Typ 3, 804
Einzeltransformationen, 592
Erreichen des Synchronlaufes, 887
Erweiterungen, 547

F

FC18, 799
Fehler
Durchhang-, 364
-kurven zur Temperaturkompensation, 345
Messsystem-, 358
Spindelsteigungs-, 358
Winkligkeits-, 364
Fehler während Pendelbewegung, 1078
fest projektierte Kopplung, 877
Festpunkt anfahren
in JOG, 331
mit G75, 331
Festpunktanfahren, 970
Festpunktpositionen, 334
FINEA, 792
Folgespindel
Neusynchronisation, 906
Folgespindelinterpolator, 877
Frames, 555
Fräs-, / Bohreinheit, 99
FS (Folgespindel) aktiv, 1082
Führungsverhalten, 916
Funktion Link, 144

G

G75, 331
G751, 331
Gantry-Masterachse freigeben, 476
Geänderte Wirksamkeit von Maschinendaten, 958
Generische Kopplung, 875
Geometrieachsen, 286
Geometrieachsen beim Handfahren, 326
Geometrieachsen definieren, 578
Geometrieachs-Verbund, 488
Geometrieüberwachung, 1084
Geschwindigkeit, 285, 297, 327
Geschwindigkeit und Beschleunigung, 329
Geschwindigkeitsführung, 513, 548
GET, 477
GETD, 477
Gitterstruktur, 374
Globale Daten, 90
Großserie, 99

H

Halt im nächsten Umkehrpunkt, 1076
Halt mit Bremsrampe, 1077
Handfahren, 326
Handfahren der Spindel, 327
Handfahren im JOG, 284
Handrad
Anschluss, 295
Anwahl von HMI, 296
Fahren im JOG, 295
Geschwindigkeitsvorgabe, 312

Wegvorgabe, 312
Zuordnung, 296
Handrad 1 als Konturhandrad aktiv, 1051
Handrad 1 als Konturhandrad aktivieren, 1050
Handrad 1 als Konturhandrad definieren, 1052
Handrad 2 als Konturhandrad aktiv, 1051
Handrad 2 als Konturhandrad aktivieren, 1050
Handrad 2 als Konturhandrad definieren, 1052
Handrad 3 als Konturhandrad aktiv, 1051
Handrad 3 als Konturhandrad aktivieren, 1050
Handrad 3 als Konturhandrad definieren, 1052
Handrad aktivieren (1 bis 3) für Geometrieachse (1, 2, 3), 1040
Handrad angewählt (für Handrad 1, 2 oder 3), 1038
Handrad-Anschluss
Ethernet, 322
Handradüberlagerung in AUTOMATIK
Geschwindigkeitsüberlagerung, 306
Programierung und Aktivierung, 309
Wegvorgabe, 306
Hardware-Endschalter, 328
Haupt-
/Nebenbedienfeld, 85
Bedienfeld, 81
Hirth-Verzahnung, 955
HMI-
Eigenschaft statische/dynamische, 88
Umschaltung, 118
Zustand, 85
Hub läuft nicht, 1070
Hubauslösung aktiv, 1070
Hubunterdrückung, 1069
HW_ASSIGN|840Di, 31

I

Inbetriebnahme der neuronalen QFK, 434
INC, 942, 943
Inkrementelles Verfahren, 292
Inkrementelles Verfahren (INC), 943
Dauerbetrieb, 292
Tippbetrieb, 292
Interpolation, 101
bei G0, 771
lineare, 771
nicht-lineare, 772
Interpolationstakt, 103
Interpolationsvorgänge, 805
Interpolator
Achs-, 771
Bahn-, 771
IPOBRKA, 793, 795
IPOENDA, 792
Istwert der analogen NCK-Eingänge, 1023
Istwert der digitalen NCK-Eingänge, 1023
Istwertkopplung, 1081
Istwertsetzen, 615
für Geo- und Zusatzachsen ($AC MEAS TYPE = 14), 645
für Zusatzachsen ($AC MEAS TYPE = 15), 646

J

JOG, 330, 555, 942
Festpunkt anfahren, 331

K

Kanal, 459, 766
Kanal-
Menü, 93
Name, 92
Kanal-Achsbezeichner, 390
Kanal-Achse, 470
Kanalnummer Geometrieachse für Handrad 1, 2, 3, 1036
Kanalsynchronisation, 462
Kartesisches Handradfahren, 568
Kartesisches PTP-Fahren, 561
Adresse STAT, 564
Adresse TU, 565
Keine Hubfreigabe, 1068
Kennzeichnung
Bedientafeln, 132
Kennzeichnung der Spindeln, 505
Kettungsrichtung, 551
Klemme X143, 318
Kommandoachsen, 792
Achsinterpolator, 792
Komparator-Eingänge, 29, 54
Kompensation
Durchhang-, 364
Interpolatorische, 355
Messsystemfehler-, 359
Schleppfehler-, 399
Spindelsteigungsfehler-, 359
Winkligkeitsfehler-, 364
Komplettbearbeitung, 502
Konfigurationsdatei, 114, 135
NETNAMES.INI, 221
Konkurrierende Positionierachsen
von der PLC starten, 801
Kontinuierliches Verfahren, 289, 942
Dauerbetrieb, 289

Tippbetrieb, 289
Konturhandrad, 312
Koordinatensysteme, 286
Kopplung
fest projektiert, 892
neue definieren, 893
Kreisformtest, 410, 412, 446
Anzeige, 448
Darstellung, 447
Messen, 447
Parametrierung, 446

L

Lage des Werkzeugnullpunktes, 508, 533
Lageregelung, 101
Längsnuten, 498
Laufzeit, 89
Lernen des neuronalen Netzes, 429
Lernen ein/aus, 431
Link-
Achse, 101, 102
Link-Achsen
Typische Linktakte, 144
Linkmodul, 102, 103
Linkvariable, 81
globale, 103
lokale NCU, 96
Lose
-kompensation, 352
Mechanische, 352
LS (Leitspindel) aktiv, 1082

M

Manuelle Hubaulösung, 1070
Manuelle Hubauslösung, 1069
Maschinen-
Steuertafel, 86
Zustand, globaler, 103
Maschinen-Achsbezeichner, 390
Maschinenachse (für Handrad 1, 2 oder 3), 1038
Maschinenfunktion für Geometrieachse (1, 2, 3), 1043
Maschinenfunktion kontinuierlich für Geometrieachse (1, 2, 3), 1043
Maschinensteuertafel, 287
Maßeinheiten INCH oder METRISCH, 628
Maßsystem, 947
Master-, Slave-Kommunikation, 80
MD10000, 476
MD10002, 146, 147, 148, 156, 161
MD10010, 461
MD10065, 192, 193, 194
MD10087, 245
MD10088, 454
MD10134, 91
MD10200, 420
MD10210, 420, 866
MD10260, 196, 363, 371, 703
MD10270, 703, 947
MD10300, 30, 32, 43, 44
MD10310, 30, 32, 45
MD10320, 33, 44
MD10330, 33, 47
MD10350, 30, 32, 35, 41
MD10360, 30, 32, 37, 39, 41
MD10361, 42
MD10362, 30
MD10364, 30
MD10366, 30
MD10368, 30
MD10382, 34
MD10384, 34, 44
MD10394, 50
MD10395, 50
MD10396, 50
MD10397, 50
MD10398, 50, 69
MD10399, 50, 51
MD10450, 704
MD10460, 704, 713
MD10461, 704, 713
MD10470, 698, 707, 709, 711
MD10471, 707, 709, 711
MD10472, 707, 709, 711
MD10473, 698, 707
MD1048, 710
MD10480, 708, 711
MD10485, 702, 709, 710, 711
MD10500, 59
MD10501, 59
MD10502, 60
MD10510, 59
MD10512, 60, 64
MD10530, 54, 55
MD10531, 55
MD10540, 55
MD10722, 483, 484, 485, 800
MD10735, 334
MD10900, 946, 958, 960
MD10910, 946, 958, 959, 960
MD10920, 946, 958, 959, 961
MD10930, 946, 958, 961
MD10940, 949, 952
MD11300, 293

MD11310, 298
MD11322, 312
MD11330, 292, 307
MD11346, 300, 301, 302, 312, 333
MD11350, 319, 320
MD11351, 319, 320
MD11352, 319, 320
MD11353, 320
MD11410, 453
MD11450, 60, 733
MD12510, 174, 175, 176, 183
MD12520, 176
MD12530, 175
MD12540, 175
MD12550, 175
MD12701, 156, 165, 255
MD12702, 165, 255
MD12703, 165, 255
MD12704, 165, 255
MD12705, 165, 255
MD12706, 165, 255
MD12707, 165, 255
MD12708, 165, 255
MD12709, 165, 255
MD12710, 165, 255
MD12711, 165, 255
MD12712, 165, 255
MD12713, 165, 255
MD12714, 165, 255
MD12715, 165, 255
MD12716, 165, 255
MD12717, 165
MD12750, 156, 160
MD13210, 612
MD1500, 443
MD18050, 933
MD18060, 930, 931
MD18096, 976, 979, 981
MD18100, 981
MD18210, 933
MD18230, 929
MD18342, 357, 368
MD18351, 664
MD18352, 929, 931
MD18353, 929, 931
MD18402, 183
MD18404, 183
MD18600, 620
MD18720, 183, 191
MD18780, 142, 168, 175, 183, 195
MD18781, 177, 178
MD18782, 183
MD20000, 92
MD20050, 851
MD20070, 146, 156, 160, 471, 472, 476
MD20100, 330, 629
MD20110, 196, 471, 747, 1008
MD20112, 471
MD20120, 1008
MD20130, 1008
MD20150, 629, 721
MD20254, 1001
MD20350, 980, 1002
MD20360, 629
MD20390, 348
MD20610, 1000
MD20620, 297
MD20624, 300, 303
MD20730, 772
MD20750, 772
MD21106, 568
MD21150, 867
MD21220, 33, 56
MD22550, 969
MD22560, 969
MD26000, 729
MD26002, 729
MD26004, 729
MD26006, 729
MD26010, 736, 741
MD26014, 735, 736, 745
MD26016, 741
MD26018, 721, 730
MD26020, 724, 730
MD30300, 849, 861, 866, 947, 959
MD30310, 850, 852, 854, 855, 857, 861, 866, 947, 959
MD30320, 852, 854, 857, 866
MD30330, 858, 950
MD30340, 855, 856, 858
MD30460, 799, 803
MD30500, 294, 946, 955, 956, 958, 961
MD30503, 856
MD30505, 955
MD30550, 160, 161, 471, 472, 476, 478
MD30552, 470, 471, 478, 769
MD30554, 176, 183
MD30560, 176
MD30600, 334
MD30610, 334
MD31090, 292, 295, 307, 313
MD32000, 308, 771, 772
MD32010, 285
MD32020, 285, 297, 327
MD32040, 285, 299, 330, 783, 944
MD32050, 285, 299, 330, 783, 944
MD32060, 311, 770, 783, 789, 791, 801

MD32074, 488
MD32080, 297, 298
MD32084, 300, 303
MD32090, 315
MD32300, 329
MD32420, 298, 772
MD32430, 772
MD32450, 353
MD32452, 354
MD32490, 409, 410, 423, 434, 443
MD32500, 413, 423, 435, 443
MD32510, 410, 413
MD32520, 411, 413
MD32530, 411
MD32540, 413, 425
MD32550, 411, 419, 420
MD32560, 412, 420
MD32570, 412, 420
MD32580, 420, 423, 442, 443
MD32610, 402, 443
MD32620, 399, 433
MD32620, 443
MD32630, 400, 443
MD32650, 404
MD32700, 356, 359
MD32710, 356, 366
MD32711, 372
MD32720, 372
MD32730, 372
MD32750, 348, 350
MD32760, 349
MD32800, 404
MD32810, 400, 401, 443
MD32900, 406
MD32910, 405
MD32990, 193, 194
MD35032, 1008, 1009
MD35040, 1009
MD36100, 861
MD36110, 861
MD36500, 354
MD36610, 453
MD36620, 453
MD37500, 781, 782
MD37510, 781, 782
MD37511, 781, 782
MD38000, 357, 360
MD38010, 423, 434, 438, 443, 444
Mechanische Lose, 352
Mehrdeutigkeit in der Rundachsposition
  Beispiel, 567
Mehrdeutigkeit in der Stellung
  Beispiele, 566
Mehrere Bedientafeln
  Alarme/Meldungen, 138
  Bedienkomponenten, 133
  Bedienoberfläche, 138
  Bedienoberflächen, 268
  Busse, 133
  Kompatibilität, 115
  Konfigurationen, 131
  Merkmale im Betrieb, 94, 280
  NCU-Komponenten, 134
  Realisierung, 93
  Umschalten der Verbindung, 135
  Verbindungen, 132
  Vorbesetzungen, 115, 135
Mehrere NCUs, 94, 268
Mehrseitenbearbeitung, 99
Mehrspindelmaschine, 99
Menü
  Verbindungen/Service, 92
Messeingangsparameter, 622
Messen
  einer Bohrung ($AC_MEAS_TYPE = 8), 637
  einer Nut ($AC_MEAS_TYPE = 12), 641
  einer schrägen Kante ($AC_MEAS_TYPE = 16), 647
  einer Welle ($AC_MEAS_TYPE = 9), 640
  eines Stegs ($AC_MEAS_TYPE = 13), 644
  eines Winkels in einer Ebene ($AC_MEAS_TYPE = 17), 649
Messergebnisse lesen im TP, 614
Messgenauigkeit, 685
Messinterface
  Ausgangswerte, 624
  Diagnose, 629
  Eingangswerte, 617
Messmethode
  um Datenhaltungsframes mit den aktuellen Wertebelegungen in ein File zu sichern ($AC_MEAS_TYPE = 26), 664
  um eine additive Drehung der aktiven oder ausgewählten Ebene vorzugeben ($AC_MEAS_TYPE = 28), 666
  um Wertebelegungen von Datenhaltungsframes zurückzuschreiben ($AC_MEAS_TYPE = 27), 665
  zur Ermittlung eines Dreiecks ($AC_MEAS_TYPE = 25), 663
  zur Koordinatentransformation einer Position ($AC_MEAS_TYPE = 24), 659
Messstatus, 1064
Messtaster betätigt, 1064
Messtasteranschluss, 608
Messtastertypen, 606
Messtastertypzuordnung, 606

Messzyklen, 626
Mindestabstand zwischen zwei aufeinanderfolgenden Hüben, 730
Minus-Nockensignal 1-32, 1065
MIRROR(C), 868
Mischung verschiedener Bussysteme, 88
Modulares Maschinenkonzept, 79
Modulo 360, 854
Modulo Rundachse
  Arbeitsfeldbegrenzung, 861
Modulo-Rundachse, 947
Möglichkeiten der Kopplung, 877
Monodirektionaler Messtaster, 607
MPI, 86, 90
MPI, Netzwerkregeln, 97
MSTT-Umschaltsperre, 1032
MSTT-Umschaltung, 124
Multi Point Interface (MPI), 86
Multidirektionaler Messtaster (3D), 606
Multiplikation
  Tabellen-, 366

N

Nachlernen, 435
Nahtstelle, 287
Nahtstellensignale
  Aktive Maschinenfunktion für Geometrieachse (1, 2, 3), 1048
  Aktive Maschinenfunktion INC1, ..., kontinuierlich, 1060
  DRF angewählt, 1045
  Eilgangüberlagerung, 1054
  Fahranforderung plus und minus, 1059
  Fahranforderung plus und minus (für Geometrieachse), 1046
  Fahranforderung plus und minus (für Orientierungsachse), 1048
  Fahrbefehl plus und minus, 1059
  Fahrbefehl plus und minus (für Geometrieachse), 1047
  Fahrbefehl plus und minus (für Orientierungsachse), 1049
  Handrad aktiv (1 bis 3), 1058
  Handrad aktiv (1 bis 3) für Geometrieachse, 1046
  Handrad aktivieren (1 bis 3), 1053
  Handradüberlagerung aktiv, 1045, 1058
  JOG - Festpunkt anfahren 0/1/2, 1057
  JOG - Festpunkt anfahren aktiv, 1061
  JOG - Festpunkt anfahren erreicht, 1061
  Konturhandrad aktiv (1 bis 3), 1046
  Maschinenfunktion INC1, INC10, INC100, INC1000, INC10000, INCvar, 1056
  Maschinenfunktion kontinuierlich, 1056
  Verfahrtasten plus und minus, 1055
  Verfahrtastensperre, 1053
NC-
  Adresse, 87, 90
NCK-Peripherie, 28
NCK-Peripherie über PROFIBUS
  Aktivierung, 59
NCU-
  Link, 85, 98
  Tausch, 92
NCU-Link, 389
NCU-Link aktiv, 1034
NCU-Link-Achse aktiv, 1035
NCU-NCU-Kommunikation, 98
Neben-
  Bedienfeld, 81
Nebenbedienfeld, 85
Negative Richtung Simulation Konturhandrad, 1051
NETNAMES.INI, 89, 91
  Syntax, 132
Netzwerkregeln, 97
Neukonfiguration, 925, 927
Neulernen, 434
Neuronale Quadrantenfehlerkompensation, 421
  Inbetriebnahme, 434
  optimieren, 437
  Parametrierung, 423
Neustart, 91
Neusynchronisation, 905
Neutrale Achse, 470
Nicht transformationsspezifisch, 582
Nocken aktiv, 1067
Nocken-Aktivierung, 1066
Nockenbereich/Nockenpaar, 693
Nockenpositionen
  Achs-/Nockenzuordnung, 704
  Setzen der Nockenpositionen, 702
Nockensignalausgabe
  unabhängig, timergesteuert, 710
Nockensignale
  Aktivierung der Ausgabe, 706
  Ausgabe an die NCK-Peripherie, 708
  Ausgabe an die PLC, 706
  für Linearachsen, 695, 699
  für Modulo-Rundachsen, 697
  HW-Zuordnung, 707
  Minus-, 706
  Plus-, 707
  Vorhalte-/Verzögerungszeit, 704
NST Vorschubhalt/Spindelhalt (DB31, ... DBX4.3), 888

Nutbearbeitung, 524
Nutquerschnitt, 526
Nutwandkorrektur, 499
Nutzungseigenschaft, 88

O

OEM-Lösung, 95, 96
Offline-
  Anforderung, 118
Offset lesen, 888
Online-Werkzeugkorrektur schreiben diskret, 998
OP030, 90
Optimierung der Geschwindigkeitsführung, 544
Orientierung, 571
Orientierung im TKS, 573
Orientierungstransformationen, 580
OS, 815
OSB, 817
OSCILL, 823
OSCTRL, 816, 817
OSE, 817
OSNSC, 817
OSP, 815
OST, 816

P

Passives Filesystem, 926
Pendelachse, 809
Pendelachsen, 793
Pendelbewegung aktiv, 1078
Pendeln, 809
  asynchrones, 809
  kontinuierliche Zustellung, 809
  mit Synchronaktionen, 835
Pendeln aktiv, 1078
Pendeln nicht startbar, 1078
Pendelumkehr aktiv, 1077
Pendelumkehr von Extern, 1076
Peripherie-Schnittstelle, 611
PG-Diagnose, 87
Planachsen, 330
PLC-
  Adresse, 87, 90
  lokale Peripherie, 80
  PLC-Kommunikation, 80, 82
PLC kontrolliert Achse, 1074, 1077
PLC kontrollierte Achse
  Steuerungverhalten bei MD30460 Bit 6 und Bit 7, 803
PLC-Achse, 470, 481
  ausschließlich PLC kontrollierte Achsen, 799
  ausschließlich PLC-kontrolliert, 486
  fest zugeordnet, 487
  fest zugeordnete PLC-Achse, 799
  über FC18 starten, 802
PLC-Achsen, 799
PLC-Service-Anzeige, 614
Plus-Nockensignal 1-32, 1066
Positieonierachsen
  Achsinterpolator, 788
Positionierachsen, 764
  achsspezifische Signale, 799
  Achstypen, 768
  Bahninterpolator, 793
  Beschleunigungswerte, 783
  Geschwindigkeitswerte, 783
  kanalspezifische Signale, 799
  Konkurrierende, 769, 799
  maximale Anzahl, 784
  Positionierachse Typ 1, 788, 789
  Positionierachse Typ 2, 788, 790
  Positionierachse Typ 3, 788, 792
  Probelaufvorschub, 804
  Satzwechsel, 788
  Typen, 767
  Werkzeugkorrektur, 786
Positionsversatz
  bei Synchronspindeln, 905
POSP, 823
Power On, 561
Pratzentotbereich, 753
Profibus, 90, 102
  -Schnittstelle, 98
Programmierung der Gelenkstellung
  Adresse STAT, 564
Programmierung mit Nutwandkorrektur, 586
Programmierung ohne Nutwandkorrektur, 588
Programmiervarianten, 546
Programmkoordination, 462
Programmkoordinierung
  Beispiel, 464
Protokollebene, 84
PTP/CP-Umschaltung
  Betriebsartenwechsel in JOG, 567
Pufferbatterie, 92
PUNCHACC, 731

Q

Quadrantenfehlerkompensation, 409, 421
Quantisierung der Kennlinie, 427
Quernuten, 498

Quittierung des gestoppten Zustands, 1074
Quittung manuelle Hubauslösung, 1071

R

Randbedingungen, 555, 562
RangeIndex, 63
RangeOffset, 63
Rechenleistung, 98
Referenzpunktfahren, 317, 942
Referenzpunktsetzen in relativen Koordinatensystemen, 645
Reibkompensation
  konventionelle, 410
Reibkompensation, 409
Reibkompensation (Quadrantenfehlerkompensation)
  Amplituden-Adaption, 410
  Inbetriebnahme, 412
  Kennlinienparameter, 411
  Quadrantenfehler, 409
RELEASE, 476
Reset, 562
Restweg löschen achspezifisch, 1072
RTLIOF, 773
RTLION, 773
Rund-
  Achse, 99
Rundachse, 849
Rundachsen, 849
  Absolutmaßprogrammierung, 858
  Absolutmaß-Programmierung, 863
  Achsadressen, 850
  Arbeitsbereich, 852
  Inbetriebnahme, 866
  Kettenmaß-Programmierung, 862, 865
  Maßeinheiten, 851
  Modulo 360, 854
  Modulowandlung, 857, 863
  Positionieranzeige, 852
  Software-Endschalter, 868
  Spiegeln, 868
  Vorschub, 852
Rundtaktmaschine, 99

S

Satzsuchlauf, 733
Satzwechsel
  Positionierachse Typ 1, 788
  Positionierachse Typ 2, 790
  Positionierachse Typ 3, 791
Satzwechselkiterium IPOBRKA
  Eintreffen WAITMC, 466
Satzwechselkriterium
  Bremsrampe, 793
  Bremsrampe mit Toleranzfenster, 794
Satzwechselzeitpunkt einstellbar, 792
Schleifbearbeitung, 539
Schleifwerkzeuge, 976
Schleppfehler, 399
Schlitten, 99
Schnellinbetriebnahme, 443
Schnittstelle, 611
Schräge Achse
  TRAANG, 539
Schräge-Achse-Transformationen, 582
Schutzstufe, 94
Schutzstufe Service, 92
SD41010, 292, 297, 298, 307
SD41040, 942
SD41050, 290, 942
SD41100, 285, 299, 330, 783, 853, 944
SD41110, 285, 297, 327
SD41120, 285, 330
SD41130, 285, 297, 867
SD41200, 327
SD41300, 366
SD41500, 702
SD41500 - 41507, 947
SD41501, 702
SD41502, 702
SD41503, 702
SD41504, 702
SD41505, 702
SD41506, 703
SD41507, 703
SD41520, 705, 713
SD41521, 705, 713
SD41522, 705
SD41523, 705
SD41524, 705
SD41525, 705
SD41526, 705, 713
SD41527, 705, 713
SD41600, 56
SD41700, 158, 159, 160
SD42100, 311, 804, 812
SD42101, 804, 812
SD42400, 727
SD42402, 721, 730
SD42404, 730
SD42600, 299, 783, 787, 944
SD43300, 299, 783, 787, 944
SD43400, 861
SD43410, 861

SD43600, 793, 795, 796
SD43610, 795
SD43770, 817
SD43790, 817
SD43900, 348
SD43910, 348
SD43920, 348, 350
Serieninbetriebnahme-Datei, 927
Service-/Inbetriebnahme, 87
Servicefall, 92
SETM, 463
Setzen von PLC der digitalen NCK-Eingänge, 1016
Setzwert von PLC der analogen NCK-Ausgänge, 1020
Setzwert von PLC der analogen NCK-Eingänge, 1019
Setzwert von PLC der digitalen NCK-Ausgänge, 1017
Simulation Konturhandrad ein, 1051
Softkey, 91, 96
Software-Endschalter, 328, 868, 962
Softwarenocken, 693
Sollwert der analogen NCK-Ausgänge, 1024
Sollwert der digitalen NCK-Ausgänge, 1023
Sollwerte, 619
Span-zu-Span-Zeit, 968
Speichererweiterung, 930
Speicherorganisation, 925
Sperre der analogen NCK-Ausgänge, 1020
Sperre der analogen NCK-Eingänge, 1018
Sperre der digitalen NCK-Ausgänge, 1016
Sperre der digitalen NCK-Eingänge, 1016
Spindelnummer, 982
Spindelposition beim Monotaster, 607
Spindeltausch, 470
Sprachbefehl
  SPN, 736, 738
  SPP, 735, 737
Sprachbefehle, 725
Standardalarmtexte, 92
Start-Bedienbereich, 91
Stations-/Lagewechsel, 102
statische HMI-Eigenschaft, 88
Statischer Anwenderspeicher, 928
Statischer NC-Speicher, 925
Statusabfrage
  HW-Ausgänge, 708
Steuerung der Handfahrfunktionen, 287
Störungen, 91
Stützpunkt, 355
SUG, 1006
  in allen Betriebsarten, 1008
SUG aktiv, 1085
Symbolname, 89
Synchronaktion, 103
Synchronbetrieb, 876, 1080
  ausschalten, 896
  geknickte Beschleunigungskennlinie, 919
Synchronisation sperren, 1080
Synchronlauf fein, 1081
Synchronlauf grob, 1081
Synchronspindel
  Positionsversatz, 905
Systemvariable, 103, 554, 614
  globale, 103

T

Tabelle
  Kompensations-, 356
  Korrektur-, 355
Teileprogramm, 102, 103, 594
Teilungsachse
  Parametrierung, 946
Teilungsachse in Position, 1083
Teilungsachsen
  Codierte Position, 948
  FRAMES, 952
  Handrad, 943
  Inbetriebnahme, 959
  Programmierung, 948
  Referenzpunktfahren, 962
Teilungspositionen
  Anzahl, 946
Teilungspositionstabelle, 946
Temperatur
  -einfluss, 345
  -kompensation, 345
Temperaturkompensation
  Koeffizient tanß(T), 350
temporäre Zuordnung, 81
Testprogramm zur Prüfung der
Wiederholgenauigkeit, 689
Tippbetrieb, 942
TRAANG
  Aktivierung, 546
  Anzahl, 541
  Ausschalten, 546
  Einschränkungen, 547
  Kurzbeschreibung, 500
  spezifische Einstellungen, 543
TRAANG
  Schräge Achse, 539
TRAANG_Angel_m, 544
TRAANG_BASE_TOOL_m, 544
TRAANG_PARALLEL_ACCEL_RES_m, 545
TRACYL, 498
  Achsabbildung, 531

Anzahl, 527
Einschränkungen, 536
TRACYL_BAE_TOOL_t, 533
TRACYL_ROT_AX_OFFSET_t, 532
TRACYL_Rot_Sign_IS_PLUS_t, 532
Tracyl-Transformationen, 581
TRAFO_AXES_IN_n, 507
TRAFO_TYPE, 530
TRAFO_TYPE_n, 506, 543
TRANS, 865
Transformation
Kettungsreihenfolge, 551
Transformation aktiv, 1063
Transformationskette, Sollpositionen, 558
Transformations-Typ 257, 506
Translation, 569
Translation im BKS, 569
Translation im TKS, 570
Translation im WKS, 570
Translation und Orientierung im TKS gleichzeitig, 571
Translatorische Verschiebungen, 619
TRANSMIT, 497, 502
Achsabbildung, 506
Aktivierung, 510
Anzahl, 503
Ausschalten, 510
Einschränkungen, 511
spezifische Einstellungen, 506
TRANSMIT_ROT_AX_OFFSET_t, 507
Transmit-Transformationen, 581
Trommel, 101
Trommel-/Rundschaltachse, 99

U

Übergänge möglicher Achszustände beim Achstausch, 475
Überlagerte Bewegung, 1082
Überlappungsbereiche der Achswinkel
Adresse TU, 565
Überschreibmaske der analogen NCK-Ausgänge, 1019
Überschreibmaske der digitalen NCK-Ausgänge, 1017
Überwachung des Eingangssignals, 730
Überwachungen, 328
Überwachungszustand bei Modulo-Rundachsen, 1079
Umkehrlose, 353
Umkehrpunkt ändern, 1076
Umkehrpunkt setzen, 1076
Umkehrpunkte, 809
Umrechnung in ein anderes Koordinatensystem, 621
Umschalt-
Bedingungen, 120
Verhalten OP030, 91
Versuch, 95
Zeitpunkt, 91
Umschaltbare Geometrieachse, 509
Umschaltbare Geometrieachsen, 533, 545
Umschaltbares Interface, 729
Unterbrechung des Teileprogramms, 513, 538
Urlöschen, 92

V

Variablenschnittstelle, 616
Verdrängungs-
Algorithmus, 89, 118
Mechanismus, 85
Regeln, 119
Strategie, 89, 119
Verfahrbereichsbegrenzung bei Modulo-Rundachsen, 1079
Verfahren von PLC, 945
Verfahrtasten plus und minus für Geometrieachse (1, 2, 3), 1042
Verfahrtastensperre für Geometrieachse (1, 2, 3), 1041
Verformung
durch Temperatureinfluss, 345
Verkabelung M
N, 88
Verkettete Transformation
Beispiel, 552
Verkettete Transformationen, 550, 593
Aktivierung, 552
Anzahl, 551
Ausschalten, 552
Beispiel, 591
Besonderheiten, 552
Persistente Transformation, 553
Verkettungsvorschrift, 982
vernetzte NCUs, 98
Verwendbare Messtaster, 606
Verzögerter Hub, 1069
Voreinstellung BTSS, 87
Vorgabemaske der analogen NCK-Ausgänge, 1019
Vorgabemaske der analogen NCK-Eingänge, 1018
Vorgabemaske der digitalen NCK-Ausgänge, 1018
Vorlaufstopp, 61
Vorschub, 783
Vorschub durch den programmierten Achsnamen einer Positionierachse zuordnen, 1075
Vorschub-/Eilgangkorrektur, 326
Vorschubkorrektur, 783, 794, 798
Vorschub-Korrektur, 285

Vorschubkorrektur/Spindelkorrektur
achspezifisch, 1072
Vorsteuerung
Drehzahl-, 401
Momenten-, 403
Vorsteuerung, 399

W

WAITE, 463
WAITM, 463
WAITMC, 463, 465
und Einlesesperre, 468
und SETM, 466
WAITP, 786
Pendelachse, 823
Wegaufteilung, 735
Wegschaltsignale, 693
Wegvorgabe mit Handrad, 312
Weg-Zeit-Nocken, 712
Eigenschaften, 712
Werkstückaufspannung, 99, 103
Werkstückvermessung, 616
Werkzeugdurchmesser messen ($AC_MEAS_TYPE = 11), 670
Werkzeugkorrektur für Schleifwerkzeuge, 976
Werkzeuglänge
($AC_MEAS_TYPE = 10), 668
mit gemerkter oder aktueller Position messen ($AC_MEAS_TYPE = 23), 672
mit Lupe messen ($AC_MEAS_TYPE = 22), 671
Werkzeugtypen für Schleifwerkzeuge, 980
Werkzeugvermessung, 667
zwei Drehwerkzeuge mit eigenem Referenzpunkt, 673
zweier Fräser mit eigenem Referenzpunkt, 680
zweier Fräser mit einem Referenzpunkt, 681
Werkzeugvermessung zweier Drehwerkzeuge
mit eigenem Referenzpunkt, 674
mit einem Referenzpunkt, 675
Werkzeugvermessung zweier Fräser
mit eigenem Referenzpunkt, 677
mit einem Referenzpunkt, 678
Werkzeugwechsel
Ablauf, 967
Festpunkte, 970
Werkzeugwechselpunkt, 970
Werkzeugwechselzeiten, 968
Winkel
schräge Achse, 544
Winkelversatz POSFS, 883
WKS auf der schrägen Ebene neu definieren ($AC_MEAS_TYPE = 18), 652

X

X143, 318
x-Kante ($AC_MEAS_TYPE = 1), 630
x-Kantenmessung, 631

Y

y-Kante ($AC_MEAS_TYPE = 2), 632

Z

Zahl der Busteilnehmer, 90
Zeitkonstante
der Dynamikanpassung, 405
z-Kante ($AC_MEAS_TYPE = 3), 633
Zulässiger Winkelbereich, 544
Zuordnung
achsgruppenweise, 101
Busteilnehmer - Bussystem, 88
HMI - PCU, 88
Zuordnung ändern, 579
Zuordnung der Geometrieachsen zu Kanalachsen, 529, 543
Zuordnung der Geometrieachsen zu Kanalachsen, 505
Zuordnung der Kanalachsen, 505
Zuordnung der Kanalachsen zu Maschinenachsen, 529
Zustellung, 809
Zuweisung von Namen an, 506
Zweite Bedientafel, 93
Zylinderkoordinatensystem, 525
Zylindermantelfläche, 498