

**SINUMERIK 810T
Grundauführung 3
Softwarestand 3**

Teil 2: Programmieren

Anwender-Dokumentation

Grundlagen der Programmierung	1
Bewegungsrichtungen, Maßangaben	2
Programmieren von Bewegungssätzen	3
Zusatz-, Schalt- und Hilfsfunktionen	4
Unterprogramme	5
Parameter	6
Konturzug	7
Werkzeugkorrekturen	8
Schneidenradiuskorrektur (SRK)	9
Zyklen	10
Programmieren von Zyklen	11
Programmschlüssel SINUMERIK 810T	12

Inhalt

	Seite
1 Grundlagen der Programmierung	1-1
1.1 Programmaufbau	1-1
1.2 Satzaufbau	1-2
1.3 Satzelemente	1-3
1.3.1 Haupt- und Nebensatz	1-3
1.3.2 Ausblendbare Sätze	1-3
1.3.3 Anmerkungen (Kommentare)	1-4
1.4 Wortaufbau	1-4
1.5 Zeichenvorrat	1-7
1.6 Lochstreifen	1-7
1.6.1 Lochstreifenleser	1-7
1.6.2 Lochstreifen-Code	1-8
1.6.3 Vorspann	1-8
1.6.4 Einlesestop	1-8
1.7 Programmformat für Ein- und Ausgabe	1-9
1.8 Code-Tabelle	1-13
1.9 Eingabe-/Ausgabeformate	1-16
1.10 Diagramme	1-20
1.10.1 Umdrehungsvorschub Grenzdaten	1-20
1.10.2 Spindeldrehzahl als Funktion des Drehradius bei $v = \text{konstant}$	1-21
1.11 Kanalstruktur	1-22
2 Bewegungsrichtungen, Maßangaben	2-1
2.1 Koordinatensystem	2-1
2.1.1 Flexible Ebenenanwahl	2-2
2.2 Weginformation, Wegbedingungen	2-4
2.3 Maßsysteme: Bezugsmaß, Kettenmaß G90/G68/G91	2-4
2.4 Bezugspunkte	2-6
2.5 Nullpunktverschiebung	2-7
2.6 Wegberechnung	2-11
2.7 Werkstückvermessung, Eingabesystem G70/G71	2-12
2.8 Spiegeln	2-13
2.9 Programmierbare Arbeitsfeldbegrenzung G25/G26	2-15
2.10 Softwaresnocken	2-17
2.11 Koordinatendrehung (KD)	2-19
2.12 Maßstabsänderung: Anwahl G51, Abwahl G50	2-22
3 Programmieren von Bewegungssätzen	3-1
3.1 Achsbefehle	3-1
3.1.1 Achsbewegung ohne Bearbeitung G00	3-3
3.1.2 Achsverdopplung	3-4
3.1.2.1 Funktionsweise	3-4
3.1.2.2 Komplettbearbeitung	3-7
3.2 Achsbewegungen mit Bearbeitung	3-8
3.2.1 Geradeninterpolation G01	3-8

3.2.2	Kreisinterpolation G02/G03	3-9
3.2.2.1	Interpolationsparameter I, K	3-10
3.2.2.2	Radiusprogrammierung	3-11
3.2.3	Zylinderinterpolation	3-14
3.2.4	Vorschub F/G94/G95/G96/G97/G98	3-16
3.2.5	Gewindeschneiden G33/G34/G35	3-18
3.2.5.1	Gewinde mit konstanter Steigung	3-19
3.2.5.2	Gewinde mit veränderlicher Steigung	3-22
3.2.5.3	Zustellmöglichkeiten	3-23
3.2.5.4	Mehrgängige Gewinde	3-25
3.2.6	Gewindebohren ohne Geber G63	3-27
3.2.7	Gewindebohren ohne Ausgleichsfutter G36	3-27
3.2.8	Genauhalt G09/G60/G00, Bahnsteuerbetrieb G62/G64	3-29
3.2.8.1	Genauhaltgrenze fein und grob G09/G60/G00	3-29
3.2.8.2	Bahnsteuerbetrieb G62/G64	3-31
3.2.9	Verweilzeit G04	3-32
3.2.10	Weiches Anfahren und Verlassen der Kontur	3-33
3.2.11	Polarkoordinaten G10/G11/G12/G13	3-35
3.2.11.1	Polarkoordinaten G110/G111	3-36
3.2.12	SPLINE - Interpolation G06	3-38
3.2.13	Koordinatentransformation TRANSMIT	3-39
3.2.13.1	Funktion TRANSMIT	3-40
3.2.13.2	Satzvorlauf bei der Funktion TRANSMIT	3-42
3.2.13.3	Prinzip der Koordinatentransformation TRANSMIT	3-43
3.2.13.4	Bearbeitungsgenauigkeit bei TRANSMIT	3-47
3.2.13.5	Geschwindigkeitsüberwachung bei TRANSMIT	3-48
3.3	Referenzpunktfahren über Teileprogramm (G74)	3-50
3.3.1	Funktionsbeschreibung	3-50
3.3.2	Starten der Funktion	3-50
3.4	Zweite Spindel	3-51
3.5	Rundachse fliegend synchronisieren	3-52
4	Zusatz-, Schalt- und Hilfsfunktionen	4-1
4.1	M, S, T, H	4-1
4.2	Zusatzfunktionen M	4-1
4.3	Spindelfunktion S	4-3
4.4	Hilfsfunktionen H	4-4
4.5	Werkzeugnummer T	4-4
5	Unterprogramme	5-1
5.1	Anwendung	5-1
5.2	Unterprogramm-Aufbau	5-1
5.3	Unterprogramm-Aufruf	5-2
5.4	Unterprogramm-Schachtelung	5-3
6	Parameter	6-1
6.1	Parameter-Programmierung	6-1
6.2	Parameter-Definition	6-2
6.3	Parameter-Rechnung	6-3
6.4	Parameter-Reihe	6-4
6.5	Programmbeispiel mit R-Parametern	6-5

7	Konturzug	7-1
7.1	Kontur-Kurzbeschreibung	7-1
7.2	Konturzug-Programmierung	7-2
7.3	Wirkungsweise der Funktion G09, F, S, T, H, M im Konturzug	7-6
7.4	Kettung von Sätzen	7-6
7.5	Programmierbeispiele Drehmaschine	7-7
7.6	Zusatzfunktionen in verketteten Sätzen	7-9
8	Werkzeugkorrekturen	8-1
8.1	Werkzeugdaten	8-1
8.2	Werkzeugkorrektur ohne Einsatz der Schneidenradiuskorrektur	8-2
8.3	Werkzeugkorrektur mit Einsatz der Schneidenradiuskorrektur	8-5
9	Schneidenradiuskorrektur (SRK)	9-1
9.1	Anwahl der SRK	9-1
9.2	SRK im Programm	9-4
9.3	Abwahl der SRK (G40)	9-7
9.4	Wechsel der Korrekturrichtung (G41, G42)	9-9
9.5	Wechsel der Korrekturnummer (G41 D., G41 D..)	9-9
9.6	Wechsel der Korrekturwerte (R1, R2)	9-10
9.7	Wiederholung der schon angewählten G-Funktion (G41, G42) mit der gleichen Korrekturnummer	9-10
9.8	M00, M01, M02 und M30 bei angewählter SRK	9-11
9.9	SRK bei Kombination von verschiedenen Satzarten und Auftreten von Konturfehlern	9-13
9.10	Sonderfälle der SRK	9-17
9.11	Wirkung bei negativen Korrekturwerten	9-21
10	Zyklen	10-1
11	Programmieren von Zyklen	11-1
11.1	Allgemeine Hinweise	11-1
11.2	Zielcode	11-1
11.2.1	Hauptgruppen	11-1
11.2.2	Operanden hinter dem Zielcode	11-2
11.2.3	Notation	11-2
11.3	Allgemeine Anweisungen für den Programmaufbau	11-3
11.4	Programmverzweigungen	11-4
11.5	Datentransfer allgemein	11-10
11.6	Datentransfer: Systemspeicher in R-Parameter	11-11
11.7	Datentransfer: R-Parameter in Systemspeicher	11-17
11.8	Mathematische Funktionen	11-23
11.9	NC-spezifische Funktionen	11-29
11.10	E/A-Funktionen	11-37
11.11	Bedienerführungsmakro (BFM)	11-42
11.12	@-Code-Tabelle	11-44
12	Programmschlüssel SINUMERIK 810T	12-1
12.1	Interne G-Gruppeneinteilung bei @36b	12-1
12.2	Programmschlüssel	12-2

1 Grundlagen der Programmierung

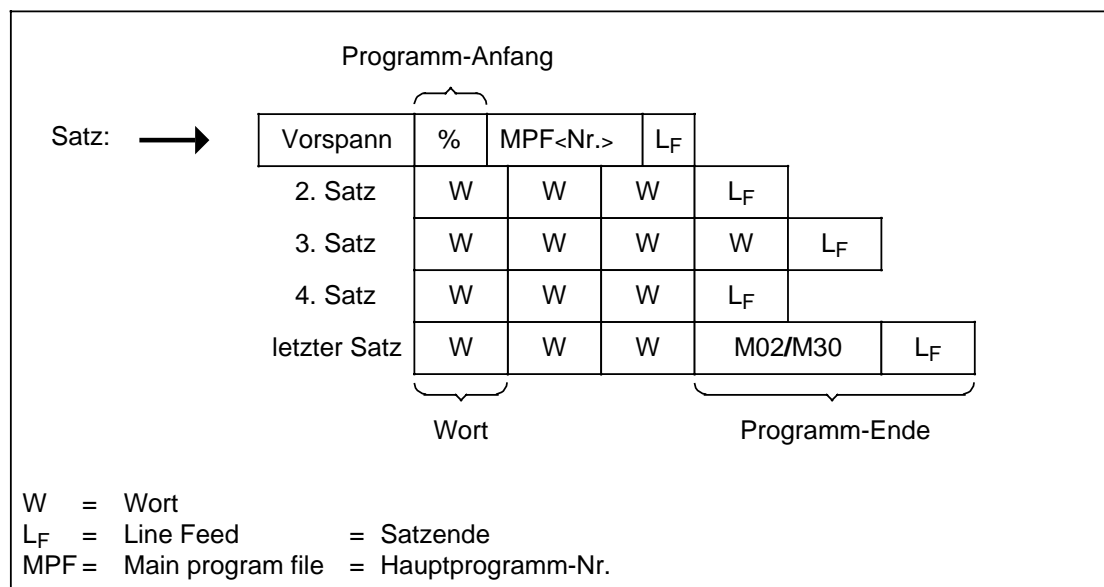
1.1 Programmaufbau

Der Programmaufbau ist an DIN 66025 angelehnt. Ein Teileprogramm besteht aus einer vollständigen Folge von Sätzen, die den Ablauf eines Bearbeitungsvorgangs auf einer numerisch gesteuerten Werkzeugmaschine beschreiben.

Ein Teileprogramm ist zusammengesetzt aus

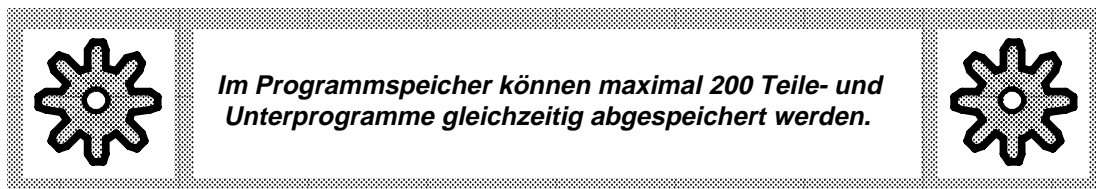
- dem Zeichen für den Programmanfang
- einer Anzahl von Sätzen und
- dem Zeichen für das Programmende.

Das Zeichen für Programmanfang geht dem ersten Satz des Teileprogramms voraus. Das Zeichen für Programmende steht im letzten Satz des Teileprogramms.



Programmschema: Teileprogramm im Ein-/Ausgabeformat

Unterprogramme und Zyklen können Bestandteile des Programms sein. Zyklen sind Unterprogramme, die vom Maschinenhersteller oder von SIEMENS erstellt wurden. Sie können gegen Mißbrauch besonders gesichert werden.



Die Eingabereihenfolge ist beliebig. Insgesamt verfügbar für Teileprogramme sind

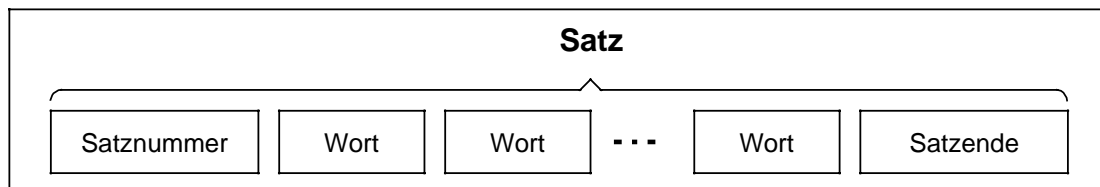
- 0 – 9999 Bearbeitungsprogramme,
- 1 – 9999 Unterprogramme.

Erfolgt die Programmeingabe mit Bedienerunterstützung über die Bedientafel, so werden bei Betätigen des Softkeys "SATZNUMMER" die Satznummern in Fünferschritten automatisch generiert. In Verbindung mit der Taste "Cancel" kann die eingegebene Satznummer gelöscht und mit "Edit" überschrieben werden.

1.2 Satzaufbau

Ein Satz enthält alle Daten zur Ausführung eines Arbeitsschrittes. Der Satz besteht aus mehreren Wörtern und dem Zeichen „LF“ für Satzende.

Die Satzlänge kann maximal 120 Zeichen betragen. Der Satz wird komplett – auf mehrere Zeilen verteilt – angezeigt.



Satzschema: Aufbau eines Satzes

Die Satznummer wird unter der Adresse N oder mit ":" eingegeben. Satznummern können frei gewählt werden. Um einen definierten Satzvorlauf und definierte Sprungfunktionen zu erhalten, darf eine Satznummer in einem Programm nur einmal verwendet werden.

Eine Programmierung ohne Satznummer ist zulässig. In diesem Fall ist jedoch kein Satzvorlauf und keine Sprungfunktion möglich.

Um den Satzaufbau übersichtlich zu gestalten, sollten die Wörter eines Satzes in der Reihenfolge des Programmschlüssels angeordnet werden.

Satzbeispiel:

```
N925 G.. X.. Z.. F.. S.. T.. M.. H.. LF
N      Adresse der Satznummer
925    Satznummer
G..    Wegbedingung
X.. Z.. Weginformation
F..    Vorschub
S..    Drehzahl
T..    Werkzeugnummer
M..    Hilfsfunktion
H..    Hilfsfunktion
LF     Satzende
```

Wird der Wert für einen Adreßbuchstaben mehrmals programmiert, so gilt der zuletzt programmierte Wert.

Jeder Satz muß am Ende mit dem Satzendezeichen "LF" abgeschlossen werden. Auf dem Bildschirm erscheint dieses Zeichen als Sonderzeichen LF. Beim Abdrucken des Programms erscheint dieses Zeichen nicht.

1.3 Satzelemente

1.3.1 Haupt- und Nebensatz

Es werden zwei Arten von Sätzen unterschieden: Hauptsatz und Nebensatz.

Im **Hauptsatz** sind alle **Wörter** anzugeben, die notwendig sind, um den Arbeitsablauf in dem dort beginnenden Programmabschnitt starten zu können. Ein Hauptsatz darf nur im Teilprogramm (Hauptprogramm) stehen. Ein Hauptsatz wird durch das **Zeichen** ":" anstelle des Adreßzeichens "N" (N=Nebensatz) gekennzeichnet. Ein Hauptsatz und ein Nebensatz dürfen in einem Programm **nicht** die gleiche Satznummer haben.

Satzbeispiel:

```
:10 G1 X10 Z-15 F200 S1000 M03 LF
```

Ein Nebensatz enthält nur die Funktionen, die sich gegenüber dem vorherigen Satz ändern.

Satzbeispiel:

```
N15 Z20 LF
```

Ein Hauptsatz und mehrere Nebensätze bilden einen Programmabschnitt.

Beispiel:

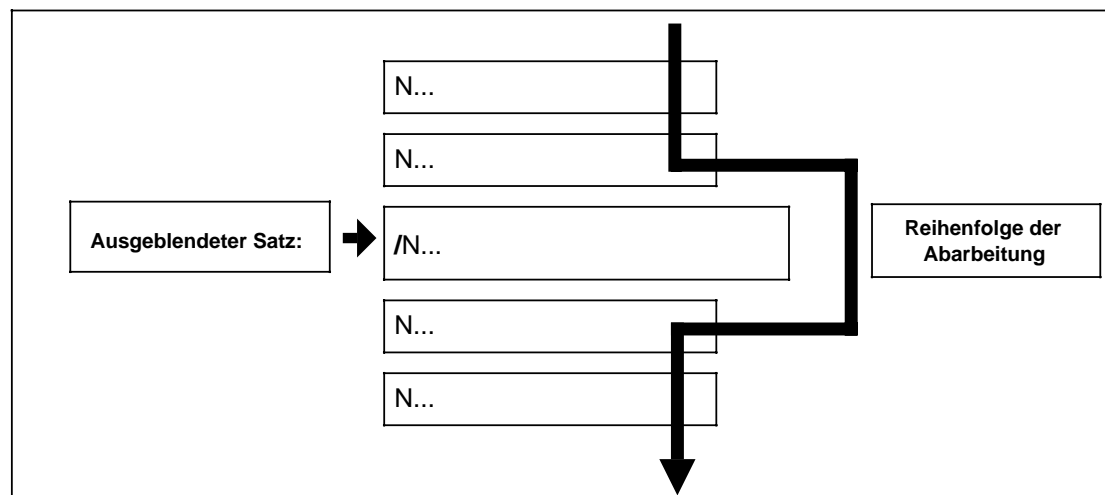
```

:10
N105
N110
N115
  
```

} Abschnitt

1.3.2 Ausblendbare Sätze

Sätze eines Programms, die nicht bei jedem Programmablauf ausgeführt werden sollen, können durch das Zeichen Schrägstrich "/" vor dem Wort der Satznummer ausgeblendet werden. Das Satzausblenden kann über den Softkey "AUSBLEND. JA-NEIN" oder durch die Anpaßsteuerung aktiviert werden. Die ausgeblendeten Sätze müssen eine Schleife bilden (gleicher Anfangs- und Endpunkt), andernfalls kann das Programm verfälscht ablaufen. Ein Abschnitt kann durch mehrere aufeinanderfolgende ausblendbare Sätze ausgeblendet werden.



Satzausblendung

Um schnellere Satzwechselzeiten zu erreichen, werden mehrere Sätze zwischengespeichert. Wenn die Maschine wegen M00 (Programmierter Halt) hält, sind die nächsten Sätze bereits vorab eingelesen. Das "Satz ausblenden" wirkt aber nur auf Sätze, die noch nicht zwischengespeichert sind. Diese Zwischenspeicherung kann durch Programmierung von L999 (Vorabeslesen sperren; @714) nach dem Satz mit M00 verhindert werden.

1.3.3 Anmerkungen (Kommentare)

Die Sätze eines Programms können durch Anmerkungen erläutert werden. Mit einer Anmerkung ist es auch möglich, Anweisungen für den Bediener auf dem Bildschirm anzuzeigen. Der Text einer Anmerkung steht zwischen den Zeichen Anmerkungsbeginn "(" und Anmerkungsende ")".

Innerhalb einer Anmerkung darf kein Prozentzeichen %, kein Satzendezeichen L_F und keine Klammerung "(",")" stehen.

Eine Anmerkung kann **bis zu 120 Zeichen lang** sein. Davon werden **höchstens 41 Zeichen** in der Kommentarzeile des **Bildschirms angezeigt**.

Es ist sinnvoll, die Anmerkung am Satzende oder in einer eigenen Zeile zu schreiben. Die Anmerkung darf keinesfalls zwischen Adresse und Ziffer oder einem Wort und dem zugehörigen Parameter stehen!

Richtig:

```
N05      G00  X100 Z200 ( Positionieren ) LF
N10      G01  X100+R1 Z200 ( Bearbeiten ) LF
N15      ....
```

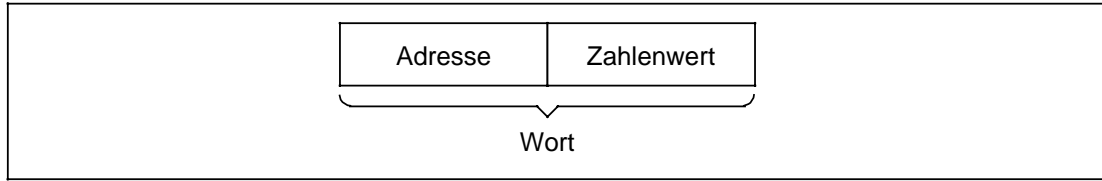
x	Adresse
100	Zahlenwert
R1	R-Parameter
(Anmerkungsbeginn, Leerzeichen
BEARBEITEN	Anmerkung, Leerzeichen
)	Anmerkungsende.

Falsch:

```
N05      X ( Positionieren ) 100 Z200 LF
N10      X100+ ( Bearbeiten ) R1 Z200 LF
```

1.4 Wortaufbau

Ein Wort ist ein Element eines Satzes. Es besteht aus einem Adreßzeichen und einer Ziffernfolge. Das Adreßzeichen ist im allgemeinen ein Buchstabe. Die Ziffernfolge kann mit Vorzeichen und Dezimalpunkten versehen sein. Das Vorzeichen steht zwischen Adreßbuchstaben und Ziffernfolge. Das positive Vorzeichen kann entfallen.



Wortschema: Aufbau eines Wortes

Beispiele:

G91 oder M30

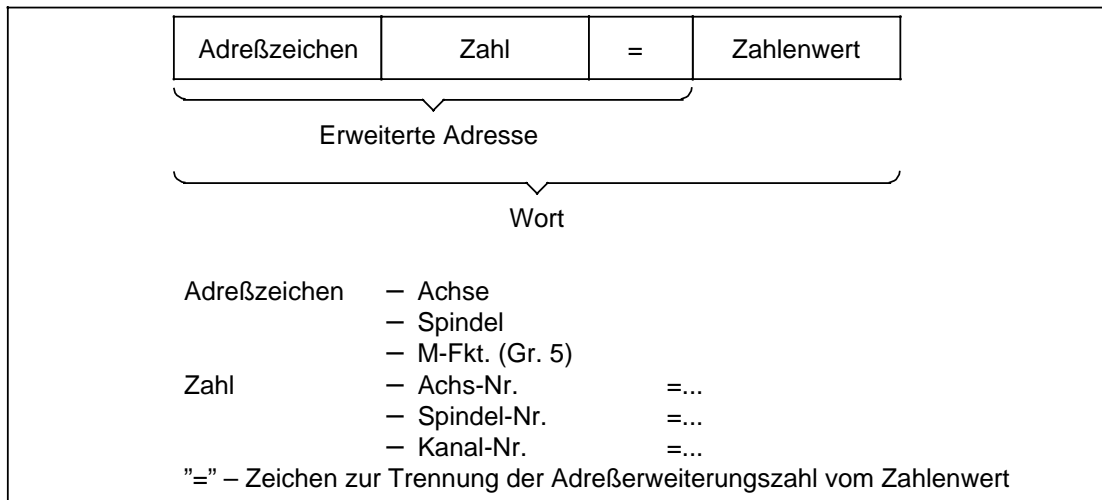
G Adresse

91 Zahlenwert, mit der Bedeutung: "Kettenmaßangabe"

M Adresse

30 Zahlenwert, mit der Bedeutung: "Programmende".

Erweiterte Adresse



Erweiterte Adresse

Beispiel:

Q1= 100

1. Hilfsachse

M1= 19

orientierter Spindelhalt der 1. Spindel

M2= 100

M-Funktion 100 wird für Kanal 2 ausgegeben

Hinweis:

G1 X1 = Z L_F bedeutet: G1 X1=0 Z0 L_F

G16 X Y1 = Z ±L_F bedeutet: G16 X0 Y1 = 0 Z0 ± L_F

Ebenenwahl mit erweiterter Adresse (Y1)

"±" gibt die Werkzeugkorrektur-Richtung an und muß immer hinter der Achse stehen.

"=" Zeichen muß geschrieben werden, um die Ziffer nach der Adresse als Adreßerweiterung zu definieren.

Der Wortaufbau ist an DIN 66025 angelehnt.

Kurzbeschreibung der Wörter:

%4 N04 G02/G03 D03 XL+053 ZL+053 QL+053 AL053 ID053 KD053
 F05 L03/L04 S05 T08 R03 RL+053 BD033 M04 H08 P02 L_F

Definition:

erster Buchstabe	Adresse
zweiter Buchstabe	L absolut/inkremental
zweiter Buchstabe	D inkremental
Zeichen	± absolute Maßangaben mit Vorzeichen positiv oder negativ
erste Ziffer	0 führende Nullen können entfallen: variable Wortlänge (G01=G1)
zweite Ziffer	Dekaden Stellen der Ziffernfolge
zweite/dritte Ziffer	Dekaden Stellen der Ziffernfolge vor und nach dem Dezimalpunkt (Koordinatenwerte X, Y, Z, I, J, K in mm)
Zeichen	L _F Satzende

Beispiel:

XL+053

x	Adresse
L	absolut/inkremental
+	Vorzeichen
0	führende Nullen können entfallen
5	Anzahl der Stellen vor dem Dezimalpunkt
3	Anzahl der Stellen nach dem Dezimalpunkt

Wortbeispiele:

x-12345.531

G9

x	Adresse	G	Adresse
-	Vorzeichen	9	Ziffer
12345	Ziffern		
.	Dezimalpunkt		
531	Ziffern		

Dezimalpunkteingabe:

Wert	Programmierter Wert mit Dezimalpunkt
0,1 μm	X.0001
1 μm	X.001
10 μm	X.01
100 μm	X.1
1000 μm	X1 oder X1.
10200 μm	X10.2

Dezimalpunkteingabe ist bei folgenden Adressen möglich:

X, Y, Z, E, A, B, C, U, V, W, Q, I, J, K, R, F, S.

Für die Adresse "R" gilt nur die Schreibweise mit erweiterter Adresse: R10 = 50.0
 (Einschränkung für S, siehe Programmschlüssel Kap. 12).

Führende und nachfolgende Nullen müssen bei der Dezimalpunktschreibweise nicht geschrieben werden.

1.5 Zeichenvorrat

Für die **Programmierung** können grundsätzlich 2 Codes verwendet werden:

- DIN 66025 (**ISO**) oder
- **EIA-RS 244-B**.

Die in der vorliegenden Anleitung aufgeführten **Beispiele** basieren auf dem **ISO-Code**. Zur Formulierung der programmtechnischen, geometrischen und technologischen Anweisungen stehen im ISO-Code folgende Zeichen zur Verfügung:

Adreßbuchstaben

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z

Kleinbuchstaben

a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z

Ziffern

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Der Name für die 5. Achse muß u.U. mit erweiterter Adresse eingegeben werden, z. B. Q1=5.

Hexadezimalziffern in Verbindung mit CL-800 Maschinencode

a, b, c, d, e, f (siehe Projektierungsanleitung CL-800)

Buchstabe

D (Eingabe der Werkzeugkorrektur [TO-TOOL OFFSET])

Abdruckbare Sonderzeichen

%, (,), +, -, /, :, ., =, *, @

Dateneingabe

Folgende Zeichen werden nicht verarbeitet und nicht abgespeichert:

HT = Tabulator (Horizontal Tabulator)
 SP = Zwischenraum (Space)
 DEL = Korrekturzeichen (Delete)
 CR = Wagenrücklauf (Carriage Return).

Weitere Steuerzeichen sind in der Code-Tabelle dargestellt.

Datenausgabe

Folgende Zeichen werden erzeugt:

- SP (nach jedem Wort)
- CR wird nach L_F zweimal oder vor L_F einmal erzeugt (Settingdaten).

1.6 Lochstreifen

1.6.1 Lochstreifenleser

Der Lochstreifenleser muß an die Steuerung angepaßt werden. Über die Setting-Daten werden Datenübertragungsrate und das Übertragungsformat (siehe Universalschnittstelle) festgelegt.

1.6.2 Lochstreifen-Code

Die Daten auf dem Lochstreifen sind nach festen Vorschriften codiert, d. h. eine Lochkombination entspricht einem ganz bestimmten Zeichen. Verwendet werden **zwei Lochstreifencodes**: ISO oder EIA (s. Code-Tabelle).

Alle Zeichen eines Codes haben ein gemeinsames Kennzeichen:

- **ISO** immer **gerade** Lochanzahl
- **EIA** immer **ungerade** Lochanzahl.

Die Steuerung erkennt mit dem ersten gelesenen % (ISO) bzw. EOR (EIA) automatisch den richtigen Code. Das Kriterium Lochanzahl gerade oder ungerade wird ab dem zweiten Zeichen des Programms zu einer Zeichen-Parity-Prüfung verwendet, womit Einfachfehler zu 100% erkannt werden.

Jeder Lochstreifen muß in einem der zugelassenen Codes geschrieben werden. Ein Wechsel des Codes auf einem Streifen, oder das Zusammenkleben von Streifen unterschiedlicher Codes ist nicht zulässig und führt zum Ansprechen der Zeichen-Parity-Prüfung.

Als weitere Prüfung wird beim wiederholten Einlesen eines bereits im Programmspeicher vorhandenen Programms ein kompletter Programmvergleich durchgeführt. Bei Fehlererkennung wird der Einlesevorgang gestoppt und der Fehler auf dem Bildschirm der Steuerung angezeigt.

1.6.3 Vorspann

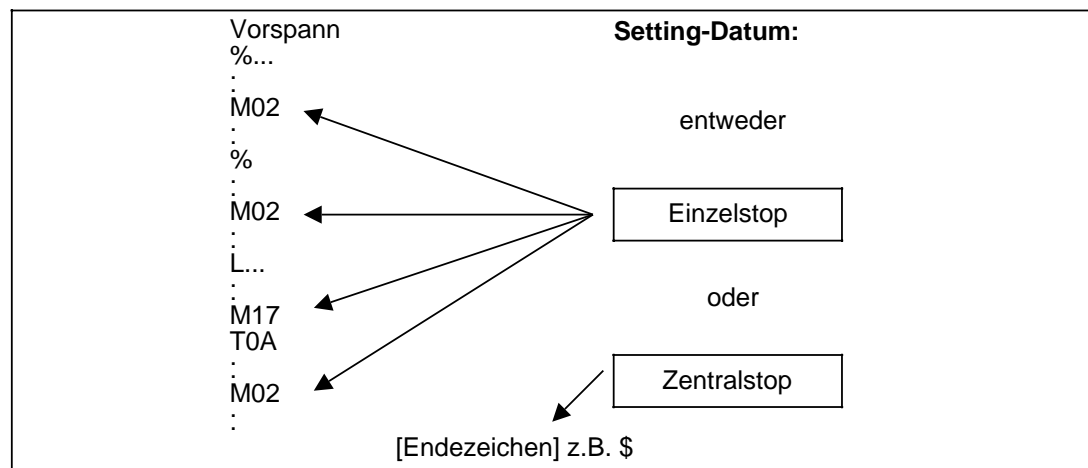
Der Vorspann wird zur Kennzeichnung der Programme verwendet. Im Lochstreifenvorspann sind alle Zeichen zugelassen, außer dem Zeichen für Programmanfang (%). Der Vorspann wird nicht abgespeichert und bei der Verarbeitung des Programms von der Steuerung nicht verarbeitet.

WELLE	%	MPF	1579	L _F
-------	---	-----	------	----------------

1.6.4 Einlesestop

Mit M02, M30, M17 wird der Einlesevorgang angehalten, wenn kein zentrales Übertragungsende-Zeichen festgelegt wurde.

Ist im Setting-Datum ein zentrales Übertragungsende-Zeichen angegeben, so führen Programm oder Datenblock-Ende (M02, M17, M30) beim Einlesen des Lochstreifens nicht zu einem Anhalten des Lesers. Der Einlesevorgang wird erst mit dem zentralen Übertragungsende Zeichen angehalten.



1.7 Programmformat für Ein- und Ausgabe

Programm für Welle	Vorspann
%MPF1234 L _F	Teileprogramm 1234 (MAIN PROGRAM FILE)
(Messung durchführen)	Anmerkung
N...L _F	Teileprogramm
N...L _F	
M02 L _F oder M30 L _F	Teileprogramm-Ende

Unterprogramme	Vorspann
%SPF234 L _F	Unterprogramm 234 (SUB PROGRAM FILE)
N5...L _F	Unterprogramm
N10...L _F	
(Bohrzyklus)...L _F	Anmerkung
M17 L _F	Unterprogramm-Ende

%ZOA L _F	Einstellbare Nullpunktverschiebung (ZERO OFFSET AKTIV)
G154 X=... Z=... L _F : G157 X=... Z=... L _F	1.-4. einstellbare Nullpunktverschiebung (grob)
G254 X=... Z=... L _F : G257 X=... Z=... L _F	
M02 L _F oder M30 L _F	Nullpunktverschiebungs-Datenblock-Ende

%TEA1 L _F	NC-Maschinendaten (TESTING DATA AKTIV 1)
N...=...L _F N...=...L _F	Maschinendaten
M02 L _F oder M30 L _F	Maschinendaten Datenblock-Ende

%TEA2 L_F	PLC-Maschinendaten (TESTING DATA AKTIV 2)
N...=...L _F N...=...L _F	Maschinendaten
M02 L _F oder M30 L _F	Maschinendaten Datenblock-Ende

%RPA0..2 L_F	(R-PARAMETER AKTIV) Kanal-Nr. (0 = zentrale R-Parameter)
R...=...L _F R...=...L _F	Parameter-Nummern mit Wertzuweisungen (Kanal 1 und 2, kanalspezifisch)
M02 L _F oder M30 L _F	R-Parameter Datenblock-Ende

%TOA L_F	Werkzeugkorrekturen (TOOL OFFSET AKTIV) je TO-Bereich (MD)
D1 P0=...P1=...P9=...L _F D2 P0=...P1=...L _F	Werkzeugkorrekturen (Anzahl der Parameter entsprechend MD)
M02 L _F oder M30 L _F	Werkzeugkorrekturen Datenblock-Ende

%SEA L_F	(SETTING DATA AKTIV) (allgemeine Settingdaten)
N...=...L _F N...=...L _F	Adressen mit Wertzuweisungen (0...9, 2000...2003, 3000...3171, 4000...4033, 5000...5771)
M02 L _F oder M30 L _F	Settingdaten Datenblock-Ende

%PCA L_F	PLC-Alarmtexte/Betriebsmeldungen (PROGRAMMABLE LOGIC CONTROL ALARM)
N6000 (Text ...) L _F : N6063 (Text ...) L _F	PLC-Alarmtexte (6000 - 6063) } PLC-Betriebsmeldungen (7000 - 7063) } Textlänge: max. 36 ASCII-Zei- chen (nicht erlaubt 'CR' und 'L _F ')
N7000 (Text ...) L _F : N7063 (Text ...) L _F	
M02 L _F oder M30 L _F	PLC-Textdatenblock-Ende

%PCP L_F	PLC-Programm (PROGRAMMABLE LOGIC CONTROL PROGRAM)
7070 8005. . .	Maschinencode
M02 L _F oder M30 L _F	PLC-Programm Datenblock-Ende

%ASM L_F	Anwender-Speicher-Modul
:Hexcode	projektierte Daten
:Hexcode	projektierte Daten
M02 L _F oder M30 L _F	ASM Datenblock-Ende

Speicherbereiche:

Die Speicherbereiche der Steuerung SINUMERIK 810T werden über folgende **Kennungen** angesprochen:

Kennung	Bedeutung
MPF	Teileprogramm (M ain P rogram F ile)
SPF	Unterprogramm (S ub P rogram F ile)
TOA	Werkzeugkorrekturen (T ool O ffset A ktiv)
ZOA	Nullpunktverschiebungen (Z ero O ffset A ktiv)
TEA1	NC-Maschinendaten (T Esting D ata A ktiv 1)
TEA2	PLC-Maschinendaten (T Esting D ata A ktiv 2)
PCA	PLC-Alarm texte
PCP	PLC-Programm-Maschinencode (P rogrammable C ontrol P rogram)
RPA	R-Parameter-Nummern mit Wertzuweisungen (R -Parameter A ktiv)
SEA	Adressen mit Wertzuweisungen (S Etting D ata A ktiv)
CLF	Löschanweisung (C Lear F ile)
ASM	Anwender-Speicher-Modul

- **Programm löschen**

Mit diesen Funktionen können über die Ein-/Ausgabe-Schnittstelle Teile- und Unterprogramme in beliebiger Reihenfolge gelöscht werden.

PROGRAMM LÖSCHEN	Vorspann
%CLF L _F	Kennung Programm löschen (CLEAR FILE) Über Settingdaten (SD für serielle Schnittstelle) kann festgelegt werden, ob mit % CLF eine automatische Reorganisation verhindert werden soll.
MPF1234 L _F	Teileprogramm % 1234 löschen
MPF 1, 1200 L _F	Teileprogramm % 1 bis % 1200 löschen
MPF 0, 9999 L _F	Löschen aller Teileprogramme
SPF 10 L _F	Unterprogramm L10 löschen
SPF 11, 79 L _F	Unterprogramm L11 bis L79 löschen
SPF 1, 9999 L _F	Löschen aller Unterprogramme
M30, M02 oder M17 L _F	Endekennung M30, M02 oder M17

- **Textdaten löschen**

%PCA L_F
 M02 oder M30 L_F

1.8 Code-Tabelle

ISO/DIN 66024 erweitert										
Zeichen	Lochkombination								Nur Vorspann u. Anmerkung	
	P	7	6	5	4	T	3	2		1
NUL						.				
SOH	.					.			.	
STX	.					.		.		
ETX						.		.	.	
EOT	.					.	.			
ENQ						.	.		.	
ACK						.	.	.		
BEL	
BS	.				.	.				
HT					.	.			.	
VT	
FF					.	.	.			
CR	
SO	
SI					
DLE	.			.		.				
DC1				.		.			.	
DC2				.		.		.		
DC3	
DC4				.		.	.			
NAK	
SYN	
ETB				
CAN				.	.	.				
EM	
SUB		
ESC				
FS			
GS				
RS				
US	
SP	.		.			.				
LF				.	.			.		
!			x
"			x
'	x
\$			x
%	
&	x
'			x
(.		.	.				
)		
*	x
+			
,	x
-			
.			
/	

Steuerzeichen werden nicht abgespeichert

ISO/DIN 66024 erweitert										
Zeichen	Lochkombination								Nur Vorspann u. Anmerkung	
	P	7	6	5	4	T	3	2		1
0			.	.		.				
1	
2		
3			
4			
5			
6			
7	
8				
9			
:				
;	x
<					x
=	
>	x
?			x
@	.	.				.				
A	
B		.				.		.		
C	
D		.				.	.			
E	
F		
G		
H		.			.	.				
I	
J		
K		
L		
M		
N		
O	
P		.		.		.				
Q	
R	
S		
T		
U		
V		
W	
X	
Y		
Z			
[.	x
\				x
]	x
^	x
-		x

EIA/ 244B										
Zeichen	Lochkombination								Nur Vorspann u. Anmerkung	
	P	7	6	5	4	T	3	2		1
kein Loch						.				x
RT				x
TAB				x
<=EOB	.					.				
LC)			
ZWR				.		.				
(.	.	.		.		
)		
EOR					
UC		
%					
&					
>			
@	
:	
.	
/			
+	
-	
0		
1						.		.	.	
2						.		.	.	
3			
4						
5			
6			
7						
8					
9			
a	
b	
c	
d	
e	
f	
g	
h	
i	
j	
k	
l	
m	
n	
o	
p	
q	
r	
s		

EIA/ 244B										
Zeichen	Lochkombination								Nur Vorspann u. Anmerkung	
	P	7	6	5	4	T	3	2		1
t			
u					
v			
w				
x			
y						
z			
IRR	

Nicht alle ISO-Zeichen können im EIA-Code dargestellt werden. Daher können beim Vergleich eines im ISO-Code erstellten Programms, das in der NC abgespeichert ist, mit seinem in den EIA-Code umgesetzten Äquivalent Unstimmigkeiten entstehen.

Beim Wiedereinlesen in die SINUMERIK-Steuerung sind folgende Funktionen nicht mehr lauffähig:

- Parameter-Rechnung,
- erweiterte Adresse,
- @-Befehle mit HEX-Ziffern (@ 36a),
- Sonderzeichen und
- Kommentare.

Der EIA-Code für "@" und ":" kann über Settingdaten vorgegeben werden (siehe Teil1, "Bedienen", Kap. 6.2 "Settingdaten zur Beschreibung der Schnittstellen").

1.9 Eingabe-/Ausgabeformate

Die Eingabe-/Ausgabeformate sind abhängig von der Einstellung der Maschinendaten des Maschinen-Herstellers.

Eingabefeinheit: 0,01 mm bzw. Lageregelfeinheit 0,005 mm
 0,001 inch 0,0005 inch
 0,001 grad 0,005 grad

Bedeutung Adressen		metrisch		Zoll		Grad	
		Bereich	Einheit	Bereich	Einheit	Bereich	Einheit
Weginformationen (Linearachsen) Interpolationsparameter		±0.01 bis 99999.99	mm	±0.001 bis 9999.999	inch	—	Grad
Weginformationen bei G91 (Rundachsen)		—		—		0.001 bis 99999.999	
Weginformationen bei G90 (Rundachsen)		—		—		±0.001 bis 359.999	
Fase (U-), Radius (U)		0.01 bis 99999.99		0.001 bis 99999.999		—	
Nullpunktverschiebung		±0.01 bis 999999.99		±0.001 bis 99999.999		±0.001 bis 99999.999	
Gewindesteigung		0.01 bis 4000.00		0.001 bis 160.000		—	
Spindeldrehzahl S (Wertigkeit über Inbetrieb- nahmeEinstellung festgelegt)		1-16000	1 min ⁻¹	1 - 16000	1 min ⁻¹		
		0.1-1600.0	0.1 min ⁻¹	0.1-1600.0	0.1 min ⁻¹		
Linear-Vorschub (F) (G94) ²⁾		0.1 bis 450000	mm/min	0.01 bis 17700	inch/min	1 bis 450000	Grad/min
Umdrehungsvorschub (F) (G95)		0.01 bis 500.00 ¹⁾	mm/U	0.001 bis 20.000 ¹⁾	inch/U		
Werkzeug- Korrektur	Länge	±0.01 bis 99999.99	mm	±0.001 bis 999.999	inch		
	Radius	±0.01 bis 9999.99		±0.01 bis 999.999			
Verweilzeit	X	0.01 bis 99999.999	s	0.01 bis 99999.999	s		
	F	0.01 bis 99999.999		0.01 bis 99999.999			
	S	0.1 bis 99.9	Umdre- hungen	0.1 bis 99.9	Umdre- hungen		
Winkel bei Konturzug (A)		—		—		0 bis 359.99999	Grad
Winkel bei orientiertem Spindelhalt (M19)						0.1 bis 359.9	Grad
R-Parameter		Dimension je nach Zugehörigkeit (intern Gleitkomma) alle Kombinationen					

1) Die max. Geschwindigkeit bei Linearvorschub (G94) darf nicht überschritten werden.
 2) Die Grenzwerte gelten für MD 155=2

Eingabefinheit: 0,001 mm bzw. Lageregelfinheit 0,0005 mm
 0,0001 inch 0,00005 inch
 0,001 grad 0,0005 grad

Bedeutung Adressen		metrisch		Zoll		Grad	
		Bereich	Einheit	Bereich	Einheit	Bereich	Einheit
Weginformationen (Linearachsen) Interpolationsparameter		±0.001 bis 99999.999	mm	±0.0001 bis 9999.9999	inch	–	Grad
Weginformationen bei G91 (Rundachsen)		–		–		0.001 bis 99999.999	
Weginformationen bei G90 (Rundachsen)		–		–		±0.001 bis 359.999	
Fase (U-), Radius (U)		0.001 bis 99999.999		0.0001 bis 9999.9999		–	
Nullpunktverschiebung		±0.001 bis 99999.999		±0.0001 bis 9999.9999		±0.001 bis 99999.999	
Gewindesteigung		0.001 bis 400.000		0.0001 bis 16.0000		–	
Spindeldrehzahl S (Wertigkeit über Inbetrieb- nahmeEinstellung festgelegt)		1 - 16000	1 min ⁻¹	1 - 16000	1 min ⁻¹		
		0.1-1600.0	0.1 min ⁻¹	0.1-1600.0	0.1 min ⁻¹		
Linear-Vorschub (F) (G94) ²⁾		0.01 bis 45000	mm/min	0.001 bis 1770	inch/min	1 bis 45000	Grad/min
Umdrehungsvorschub (F) (G95)		0.001 bis 50.000 ¹⁾	mm/U	0.0001 bis 2.0000 ¹⁾	inch/U		
Werkzeug- Korrektur	Länge	±0.001 bis 9999.999	mm	±0.0001 - 999.9999	inch		
	Radius	±0.001 bis 999.999		±0.0001 - 99.9999			
Verweilzeit	X	0.01 bis 99999.999	s	0.01 bis 99999.999	s		
	F	0.01 bis 99999.999		0.01 bis 99999.999			
	S	0.1 bis 99.9	Umdre- hungen	0.1 bis 99.9	Umdre- hungen		
Winkel bei Konturzug (A)		–		–		0 bis 359.99999	Grad
Winkel bei orientiertem Spindelhalt (M19)						0.1 bis 359.9	Grad
R-Parameter		Dimension je nach Zugehörigkeit (intern Gleitkomma) alle Kombinationen					

1) Die max. Geschwindigkeit bei Linearvorschub (G94) darf nicht überschritten werden.

2) Die Grenzwerte gelten für MD 155=2

Eingabefinheit: 0,0001 mm bzw. Lageregelfinheit 0,00001 mm
 0,0001 inch 0,000001 inch
 0,001 grad 0,00001 grad

Bedeutung Adressen		metrisch		Zoll		Grad	
		Bereich	Einheit	Bereich	Einheit	Bereich	Einheit
Weginformationen (Linearachsen) Interpolationsparameter		±0.0001 bis 99999.999	mm	±0.00001 bis 999.99999	inch	—	Grad
Weginformationen bei G91 (Rundachsen)		—		—		0.001 bis 99999.999	
Weginformationen bei G90 (Rundachsen)		—		—		±0.001 bis 359.999	
Fase (U-), Radius (U)		0.0001 bis 9999.9999		0.00001 bis 999.99999		—	
Nullpunktverschiebung		±0.0001 bis 9999.9999		±0.00001 bis 999.99999		±0.001 bis 99999.999	
Gewindesteigung		0.0001 bis 40.0000		0.0001 bis 1.6000		—	
Spindeldrehzahl S (Wertigkeit über Inbetrieb- nahmeeinstellung festgelegt)		1 - 16000	1 min ⁻¹	1 - 16000	1 min ⁻¹		
		0.1-1600.0	0.1 min ⁻¹	0.1-1600.0	0.1 min ⁻¹		
Linear-Vorschub (F) (G94) ²⁾		0.001 bis 9000.000	mm/min	0.0001 bis 350.0000	inch/min	1 bis 45000	Grad/min
Umdrehungsvorschub (F) (G95)		0.0001 bis 5.0000 ²⁾	mm/U	0.0001 bis 02.0000 ²⁾	inch/U		
Werkzeug- Korrektur	Länge	±0.0001 bis 999.9999	mm	±0.0001 bis 99.99999	inch		
	Radius	±0.0001 bis 99.9999		±0.00001 bis 9.99999			
Verweilzeit	X	0.01 bis 99999.999	s	0.01 bis 99999.999	s		
	F	0.01 bis 99999.999		0.01 bis 99999.999			
	S	0.1 bis 99.9	Umdre- hungen	0.1 bis 99.9	Umdre- hungen		
Winkel bei Konturzug (A)		—		—		0 bis 359.99999	Grad
Winkel bei orientiertem Spindelhalt (M19)						0.1 bis 359.9	Grad
R-Parameter		Dimension je nach Zugehörigkeit (intern Gleitkomma) alle Kombinationen					

1) Die max. Geschwindigkeit bei Linearvorschub (G94) darf nicht überschritten werden.
 2) Die Grenzwerte gelten für MD 155=2

Eingabefinheit: 0,0001 mm bzw. Lageregelifeinheit 0,00005 mm
 0,00001 inch 0,000005 inch
 0,001 grad 0,00005 grad

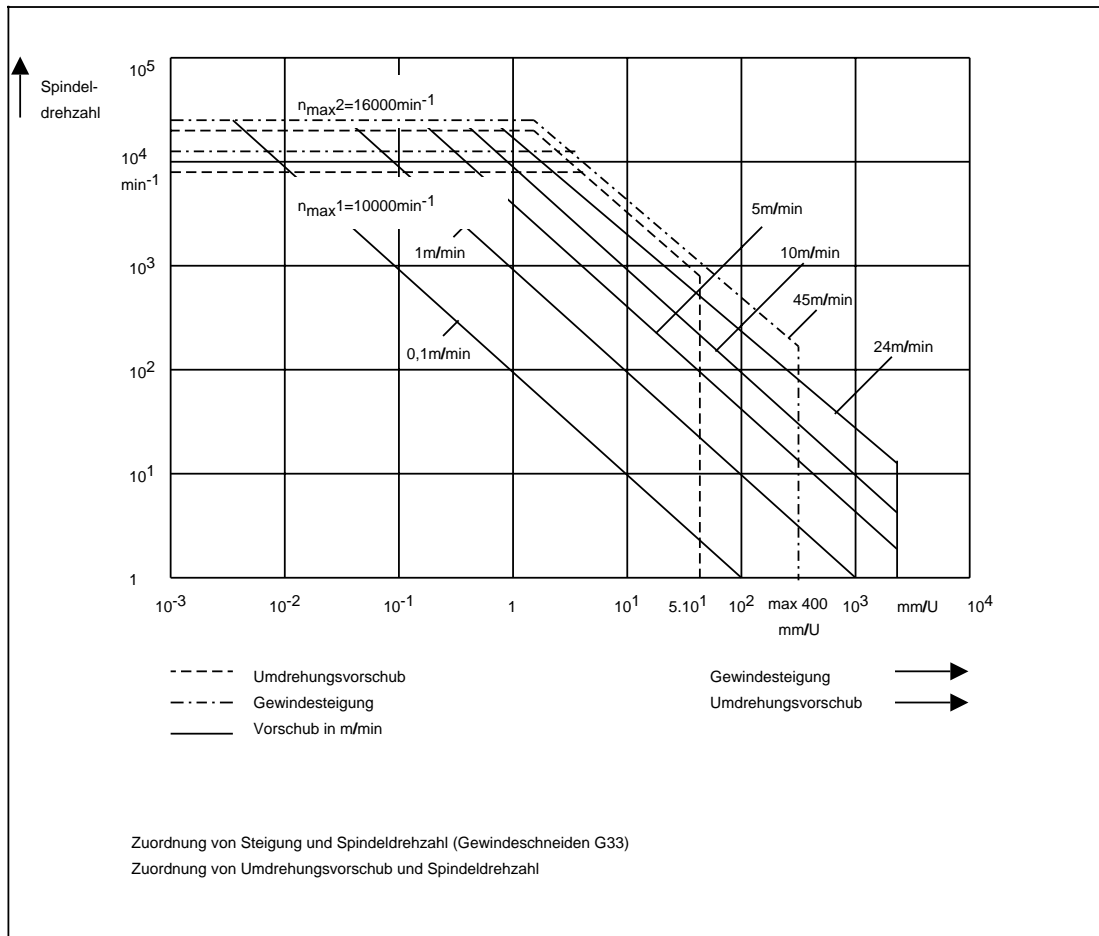
Bedeutung Adressen		metrisch		Zoll		Grad	
		Bereich	Einheit	Bereich	Einheit	Bereich	Einheit
Weginformationen (Linearachsen) Interpolationsparameter		±0.0001 bis 9999.9999	mm	±0.00001 bis 999.99999	inch	–	Grad
Weginformationen bei G91 (Rundachsen)		–		–		0.001 bis 99999.999	
Weginformationen bei G90 (Rundachsen)		–		–		±0.001 bis 359.999	
Fase (U-), Radius (U)		0.0001 bis 9999.9999		0.00001 bis 999.99999		–	
Nullpunktverschiebung		±0.0001 bis 9999.9999		±0.00001 bis 999.99999		±0.001 bis 99999.999	
Gewindesteigung		0.0001 bis 200.0000		0.00001 bis 8.00000		–	
Spindeldrehzahl S (Wertigkeit über Inbetrieb- nahmeEinstellung festgelegt)		1 - 16000	1 min ⁻¹	1 - 16000	1 min ⁻¹		
		0.1-1600.0	0.1 min ⁻¹	0.1-1600.0	0.1 min ⁻¹		
Linear-Vorschub (F) (G94) ²⁾		0.001 bis 4500.000	mm/min	0.0001 bis 69.0000	inch/min	1 bis 450	Grad/min
Umdrehungsvorschub (F) (G95)		0.0001 bis 5.0000 ¹⁾	mm/U	0.0001 bis 0.20000 ¹⁾	inch/U		
Werkzeug- Korrektur	Länge	±0.0001 bis 999.9999	mm	±0.00001 bis 99.99999	inch		
	Radius	±0.0001 bis 99.9999		±0.00001 bis 9.99999			
Verweilzeit	X	0.01 bis 99999.999	s	0.01 bis 99999.999	s		
	F	0.01 bis 99999.999		0.01 bis 99999.999			
	S	0.1 bis 99.9	Umdre- hungen	0.1 bis 99.9	Umdre- hungen		
Winkel bei Konturzug (A)		–		–		0 bis 359.99999	Grad
Winkel bei orientiertem Spindelhalt (M19)						0.1 bis 359.9	Grad
R-Parameter		Dimension je nach Zugehörigkeit (intern Gleitkomma) alle Kombinationen					

1) Die max. Geschwindigkeit bei Linearvorschub (G94) darf nicht überschritten werden.

2) Die Grenzwerte gelten für MD 155=2

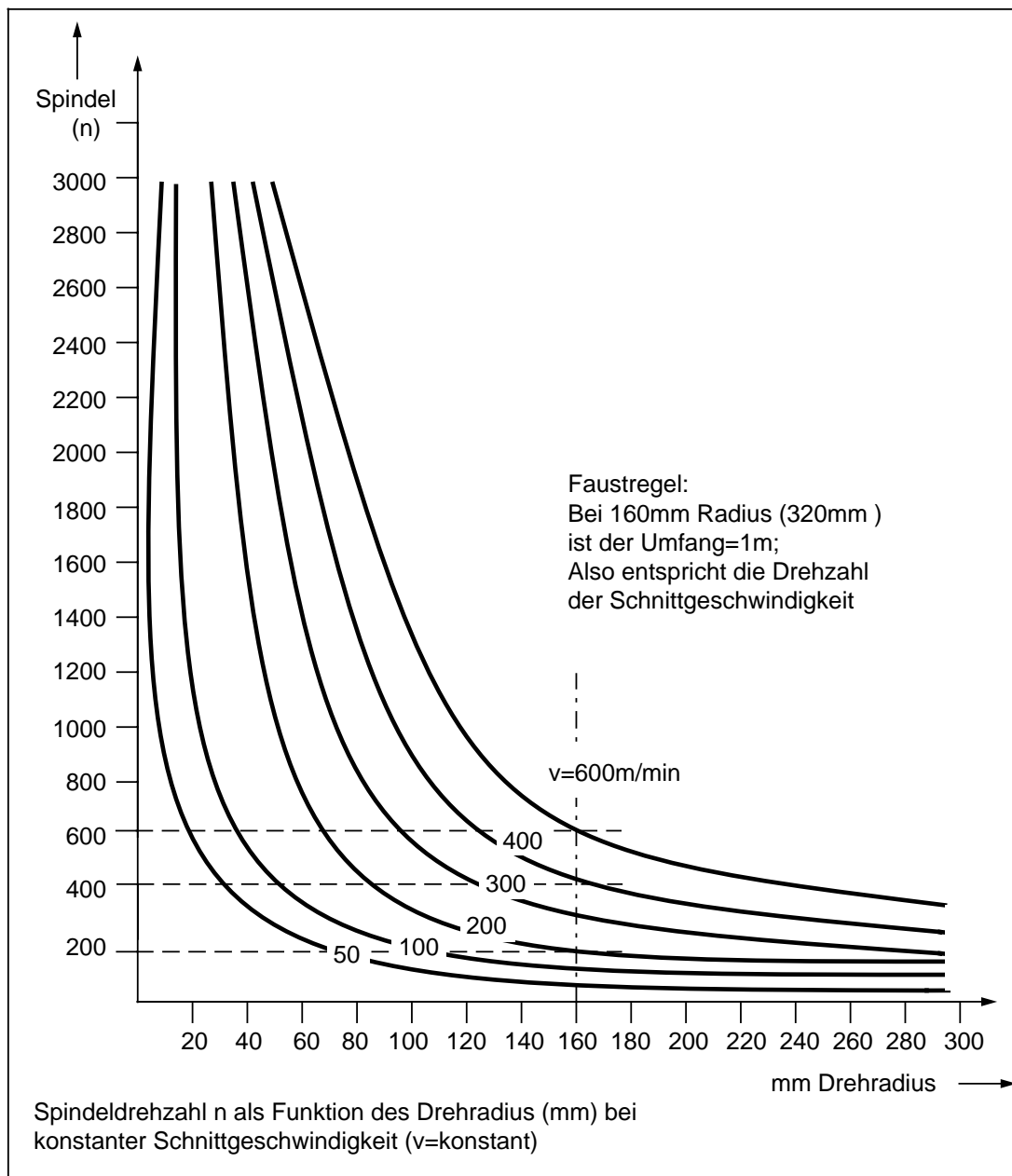
1.10 Diagramme

1.10.1 Umdrehungsvorschub Grenzdaten



n_{max1} erreichbar mit Geber 1024 Pulse/Umdrehung
 n_{max2} erreichbar mit Geber 512 Pulse/Umdrehung

1.10.2 Spindeldrehzahl als Funktion des Drehradius bei $v = \text{konstant}$



1.11 Kanalstruktur

Die **Steuerung SINUMERIK 810T** ist mit 3 Kanälen ausgestattet. Diese Kanäle ermöglichen, neben anderen strukturellen Bedienungen wie Programmeditieren und Schnittstellenbedienungen parallel zur Abarbeitung in der Betriebsart "AUTOMATIC", auch das **simultane** Abarbeiten von 2 unterschiedlichen Programmen.

Die 3 Kanäle haben folgende Bedeutung:

Kanal 1: Hauptkanal zum Abarbeiten von Programmen und Spindelprogrammierung.

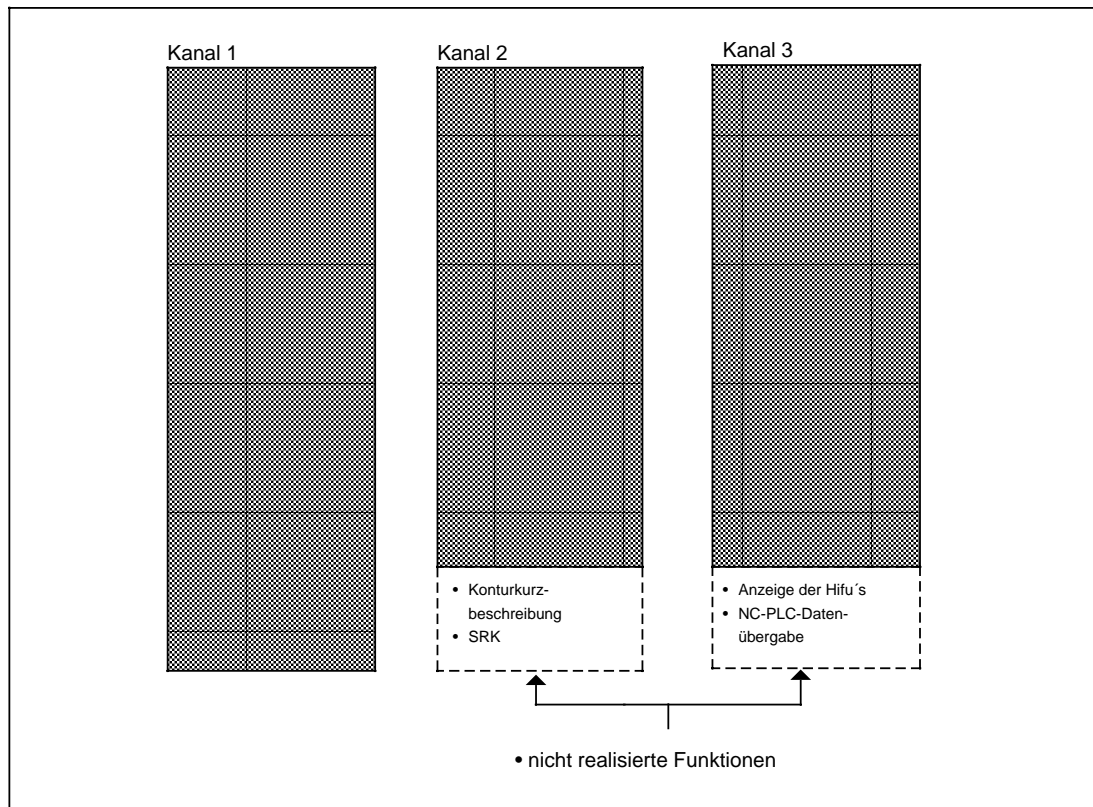
Kanal 2: Hilfskanal zum Abarbeiten von Programmen für Zusatzachsen oder für Rechenfunktionen im Hintergrund.

Kanal 3: Grafische Simulation zur Programm-Darstellung am Bildschirm.



Grundsätzlich können alle **3 Kanäle gleichzeitig** betrieben werden. Bei einigen wenigen Funktionen treten jedoch **Kollisionsprobleme** auf.

Funktionsumfang der Kanäle:



Der **Hilfskanal (Kanal 2)** ist bis auf die nicht realisierten Funktionen ein vollwertiger Kanal. Seine Hauptaufgabe ist es, Rechnungen die im Hintergrund laufen durchzuführen oder Hilfsbewegungen (z. B. Werkzeugwechsel, ...) auszuführen.

Die Zuordnung, welche Achse (im Automatik-Betrieb) in welchem Kanal verfahren wird, muß im Programm getroffen werden. Ein und dieselbe Achse kann in Kanal 1 und Kanal 2 bewegt werden, wenn ausgeschlossen wird, daß vom 1. und 2. Kanal gleichzeitig ein Fahrbefehl ausgegeben wird (... Alarm 180* , "Achse in beiden Kanälen programmiert").

Hauptsächlich dient der **Hilfskanal** jedoch dazu, simultan zum Hauptkanal von der PLC gesteuert, **Ladeachsen** zu bedienen. Mit den oben genannten Bedingungen können mit dem Hilfskanal aber auch andere Konzepte realisiert werden, so daß sich mannigfaltige Einsatzmöglichkeiten ergeben.

Da vom **Kanal 2** aber **nur M-Funktionen** an die PLC übergeben werden können, sind die Möglichkeiten des Datenaustausches mit der PLC eingeschränkt.

Der **Kanal 3** dient ausschließlich zur Grafischen Simulation eines Teileprogramms. Dabei kann gleichzeitig ein anderes Teileprogramm abgearbeitet werden

(Beachten Sie die Hinweise im Teil1, "Bedienen", Kap 3.1.13.5 "Programm verschieben" und Kap. 3.1.14 "SIMULATION").

Durch Maschinendatum kann dem 2. Kanal die Steuerung von Spindeln ermöglicht werden. Dazu stehen auch im 2. Kanal alle Funktionen bezüglich der Spindeln zur Verfügung.

2 Bewegungsrichtungen, Maßangaben

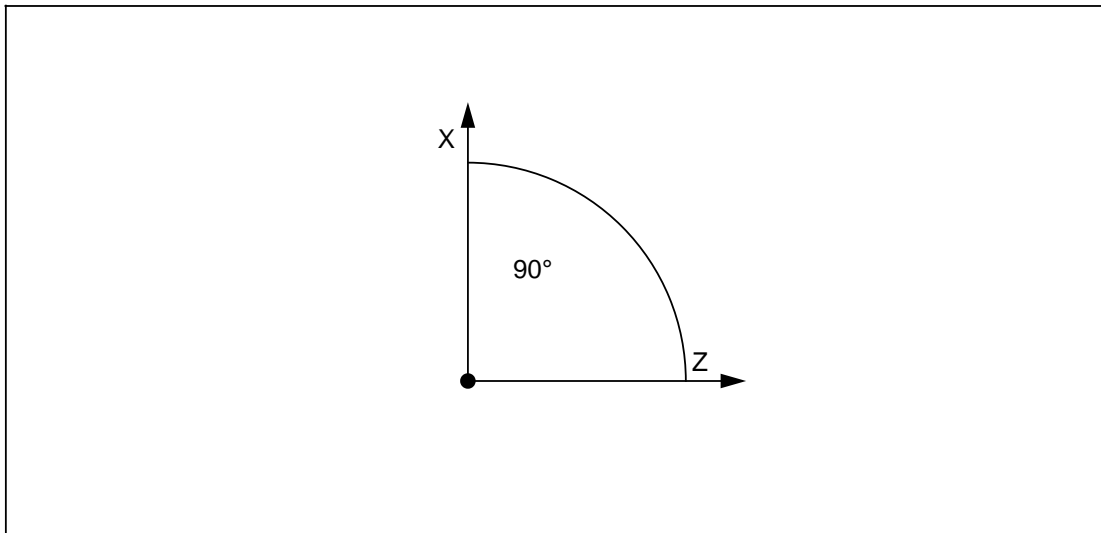
2.1 Koordinatensystem

Die Bewegungsrichtungen einer Werkzeugmaschine lassen sich auf ein Koordinatensystem zurückführen, das den Bewegungsachsen der Maschine zugeordnet ist.

Verwendet wird ein **rechtsdrehendes, rechtwinkeliges Koordinatensystem** mit den Achsen X und Z. Das System ist auf die Hauptachsen der Maschine ausgerichtet.

Das Koordinatensystem ist folgendermaßen definiert:

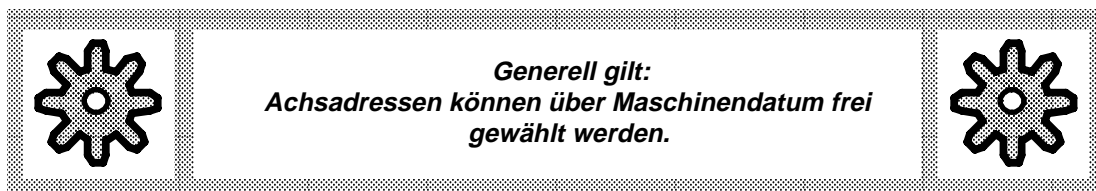
- Die zweite Achse steht senkrecht auf der ersten Achse.



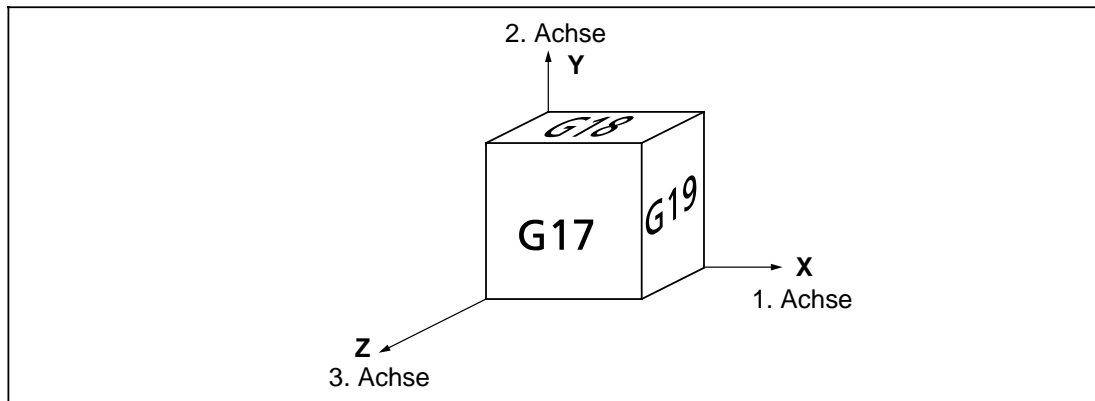
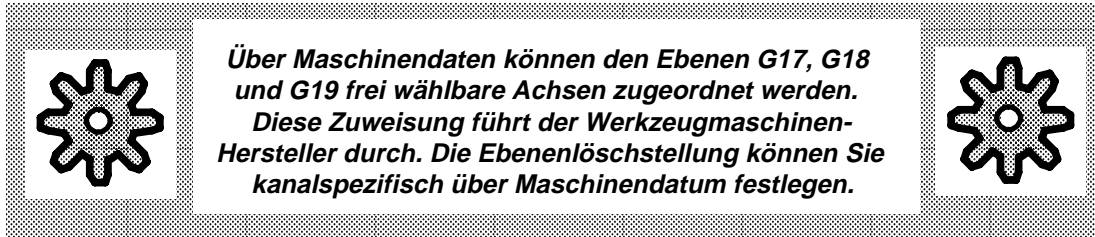
Rechtsdrehendes Koordinatensystem

Die Programmierung erfolgt unabhängig davon, ob bei der Bearbeitung das Werkstück oder das Werkzeug bewegt wird.

Als Vorzugsbelegung für **Drehmaschinen** gilt: **Hauptachsen X und Z.**



2.1.1 Flexible Ebenenanwahl



Beispiel der Achsbelegung bei flexibler Ebenenanwahl

Definition der Ebenen in der Grundstellung nach DIN:

	M-Version	T-Version
G16	Ebenenwahl mit freier Achsenwahl	
G17	Ebene X – Y (1. Achse – 2. Achse)	Ebene X – Z (1. Achse – 2. Achse)
G18	Ebene X – Z (1. Achse – 3. Achse)	Ebene X – Z (1. Achse – 3. Achse)
G19	Ebene Y – Z (2. Achse – 3. Achse)	Ebene Z – Z (2. Achse – 3. Achse)

Die drei Achsnamen für die Achsen, welche die Ebenen G17, G18 oder G19 bilden, erfahren Sie vom Werkzeugmaschinen-Hersteller.

Die Zuordnung der Achsen zu den Ebenen G17, G18 und G19 legt ebenfalls der Werkzeugmaschinen-Hersteller fest. Bei fehlerhafter Eingabe der Achsnamen erscheint nach NC-Start der Alarm 3003 "ungültige Adresse".

Die vordefinierten Ebenen G17, G18 und G19 können Sie anwählen, indem Sie die entsprechende G-Funktion im Programm aufrufen.

Getrennt für jeden Kanal, können Sie zusätzlich über Maschinendaten (NC-MD 110*) die Grundstellung bzw. Löschstellung für die Ebenenanwahl einstellen (siehe Teil 1, "Bedienen", Kap. 7.3.4 "Festlegung der Löschstellung der G-Gruppen").

Ebenenwahl über G16

G16 programmieren Sie, wenn die gewünschte Ebene nicht mehr durch die in G17, G18 oder G19 festgelegten Achsen bestimmt ist oder zusätzliche Definitionen erforderlich sind. Dies ist der Fall bei folgenden Funktionen:

- Parallelachse übernimmt die Funktion einer Hauptachse
- Zusätzliche Definition der Werkzeugkorrektur
- Angabe der Richtungen von Winkelköpfen.

Beispiel 1: Parallelachse

G16 X Y W
G17-Ebene sei definiert durch die Achsen X,Y,Z,
die W-Achse (Parallelachse) sei parallel zur Z-Achse.

Beispiel 2: Winkelkopf

G16 X Y Z Y
Die ersten beiden Achsadressen (X,Y) legen die Ebene fest, in der die Fräserradius-/Schneidenradiuskorrektur wirkt. Die 4. Achsadresse gibt die Achse mit der zusätzlichen Längenkorrektur in der zuvor festgelegten Ebene an. Zur Umschaltung der Korrekturrichtung bei FRK/SRK kann die 3. und 4. Achsadresse mit einem negativen Vorzeichen versehen werden.

2.2 Weginformation, Wegbedingungen

Eine Weginformation besteht aus einer Achsadresse und einem Zahlenwert, der den Weg auf der adressierten Achse beschreibt.

Wird ein Vorzeichen angegeben, so steht es zwischen Adresse und dem Zahlenwert.

Um den Positioniervorgang zu starten, muß die Weginformation durch die Wegbedingung (G-Funktion) und die Angaben über den Vorschub (F) ergänzt werden. Die Wegbedingungen beschreiben die Art der Maschinenbewegungen, die Interpolationsart sowie die Vermaßlungsart.

Die **G-Funktionen** sind **in Gruppen eingeteilt** (siehe Programmschlüssel Kap. 12).

In einem Programmsatz darf aus jeder Gruppe jeweils nur eine Funktion stehen.

Die G-Funktionen sind entweder **modal** (selbthaltend) oder **satzweise** wirksam:

- Modal wirksam sind G-Funktionen, die so lange wirken, bis sie durch eine neue G-Funktion aus derselben Gruppe ersetzt werden.
- Satzweise wirksam sind G-Funktionen, die nur in dem Satz wirken, in dem sie stehen.

Nach dem Einschalten der Steuerung, Reset oder Programmende sind die Grundstellungen wirksam. Sie müssen nicht programmiert werden.

2.3 Maßsysteme: Bezugsmaß, Kettenmaß G90/G68/G91

Die Verfahrbewegung auf einen bestimmten Punkt im Koordinatensystem kann durch **ABSOLUTE** oder **INKREMENTELLE** Maßeingabe beschrieben werden:

Bezugsmaßeingabe G90

Bei der Bezugsmaßeingabe (Absolutmaßeingabe) beziehen sich alle Maßangaben auf einen festgelegten Nullpunkt, der meist der **Werkstücknullpunkt W** ist. Der Zahlenwert der zugehörigen Weginformation gibt die Zielposition im Koordinatensystem an.

Bezugsmaßeingaben auf kürzestem Weg (nur bei Rundachse) G68

Das Anfahren des Satzendwertes erfolgt bei G68 auf kürzestem Weg (< 180°). Die Funktion ist selbthaltend. Die Funktion wirkt nur bei Achsen, denen die Teilfunktion "Modulo-Programmierung" zugewiesen wurde. Außerhalb der Teilfunktion ist das Verhalten von G68 äquivalent zu G90.

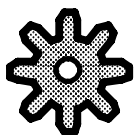
Kettenmaßeingabe G91

Bei der Kettenmaßeingabe (Inkrementalmaßeingabe) entspricht der Zahlenwert der Weginformation dem zu verfahrenen Weg. Das Vorzeichen gibt die Verfahrrichtung an.

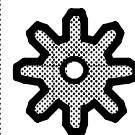
Abhängig von der momentanen Position, fährt das Werkzeug um die programmierten Werte weiter.

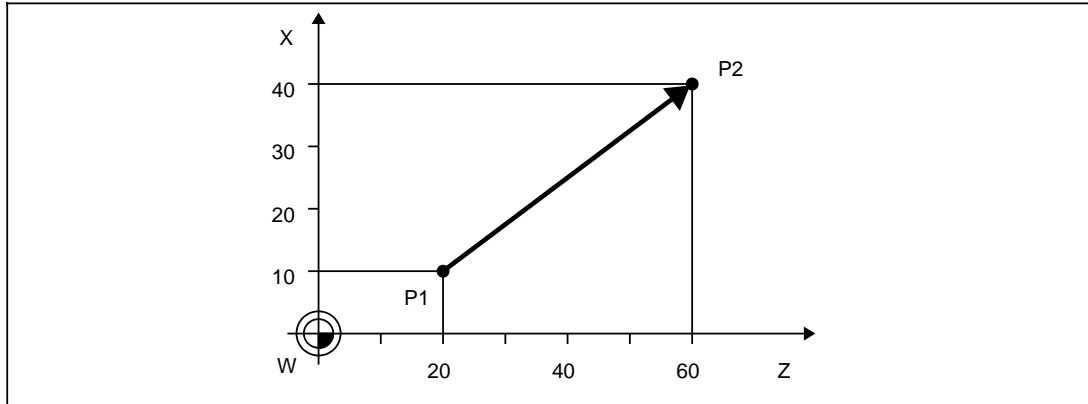
Zwischen Bezugsmaß- und Kettenmaßeingabe kann von Satz zu Satz beliebig umgeschaltet werden, da der Steuerungswert immer auf den Nullpunkt bezogen mitläuft.

Eine Nullpunktverschiebung wird sowohl bei absoluter als auch bei inkrementeller Programmierung eingerechnet.



Abhängig von Maschinendaten ist auch eine gemischte Programmierung mit G90 und G91 in einem Satz möglich.



Beispiel: Bezugsmaß- und Kettenmaßeingabe**Bezugsmaßeingabe:**

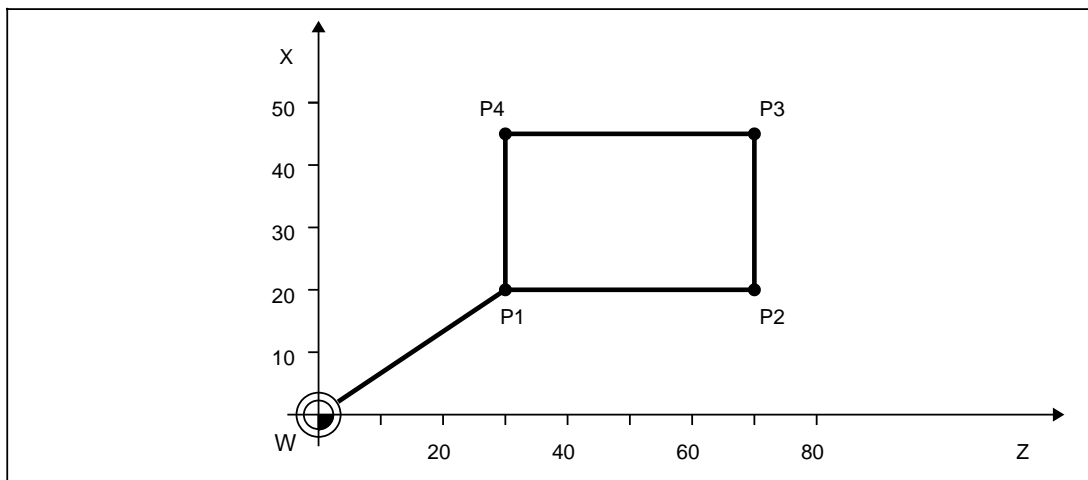
```
N...G00 G90 X40 Z60 LF
```

Das Werkzeug fährt aus beliebiger Position nach P2

Kettenmaßeingabe:

```
N...G00 G91 X30 Z40 LF
```

Das Werkzeug fährt von P1 nach P2

Beispiel: Programmierung im Bezugsmaß und im Kettenmaß**Wechsel von Bezugsmaßeingabe G90 und Kettenmaßeingabe G91:**

Zwischen Bezugsmaß- und Kettenmaßeingabe kann von Satz zu Satz beliebig umgeschaltet werden.

```
%10 LF
N5 G00 G90 G94 X20 Z30 LF (P1)
N10 G01 G91 Z40 F100 LF (P2)
N15 X25 LF (P3)
N20 Z-40 LF (P4)
N25 X-25 LF (P1)
N30 M30 LF
```

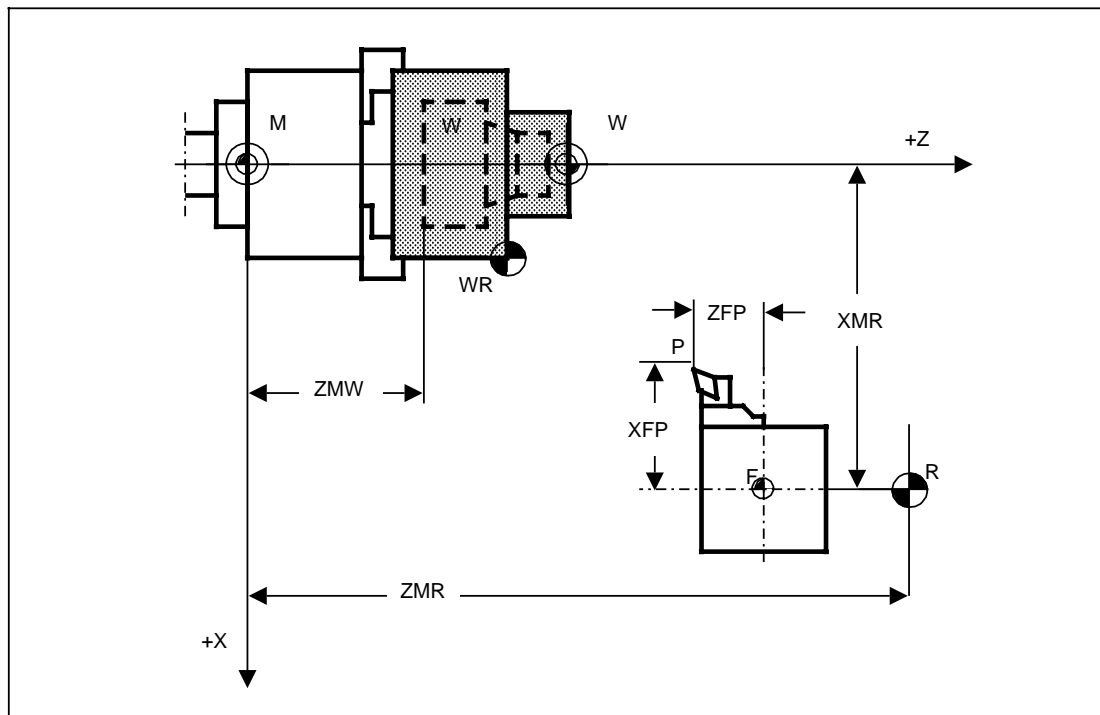
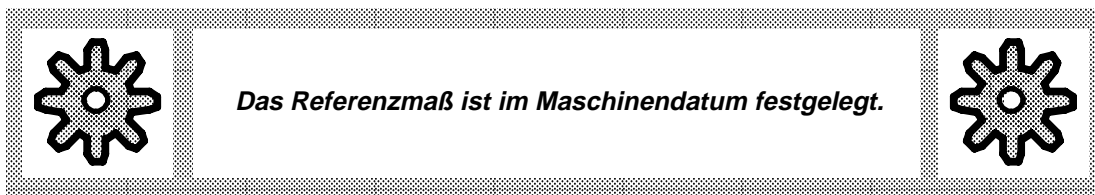
2.4 Bezugspunkte

An jeder numerisch gesteuerten Werkzeugmaschine werden die Nullpunkte und verschiedene Bezugs- oder Referenzpunkte definiert.

Der Maschinennullpunkt **M** ist der konstruktionsmäßig festgelegte Nullpunkt des Maschinenkoordinatensystems.

Der Werkstücknullpunkt **W** ist der für die Programmierung der Werkstückvermessung festgelegte Nullpunkt. Er kann vom Programmierer frei gewählt werden. Der Bezug zum Maschinennullpunkt wird durch die Nullpunktverschiebung festgelegt.

Der Referenzpunkt **R** ist ein vom Maschinenhersteller festgelegter Punkt, der nach dem Einschalten der Steuerung angefahren wird und die NC-Steuerung mit der Werkzeugmaschine synchronisiert.



Beispiel: Längsdrehen (Bearbeitung vor der Drehmitte)

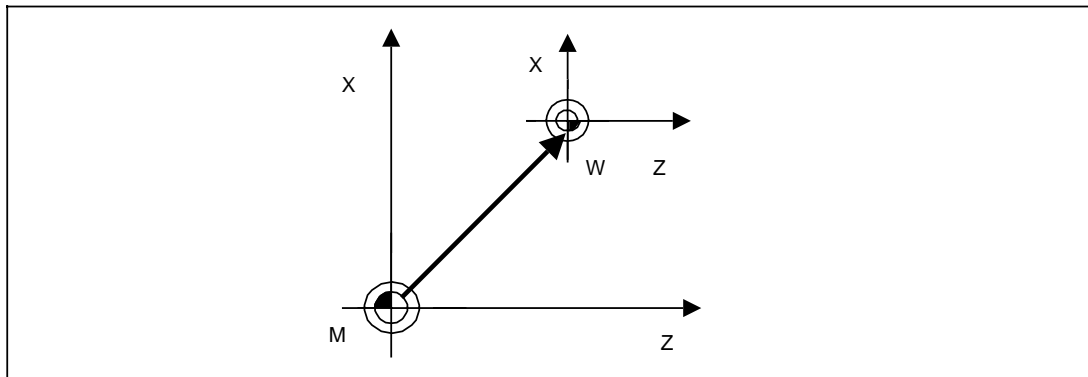
P ist der Referenzpunkt für die Werkzeugeinstellung.

Zum Beispiel "Längsdrehen (Bearbeitung vor der Drehmitte)":

P	Werkzeugeinstellpunkt
M	Maschinennullpunkt
W	Werkstücknullpunkt
R	Maschinen-Referenzpunkt
F	Schlittenbezugspunkt
WR	Werkstück-Referenzpunkt
XMR	Referenzpunktcoordinate X
ZMR	Referenzpunktcoordinate Z
XMW	Summe der Nullpunktverschiebungen X
ZMW	Summe der Nullpunktverschiebungen Z
XFP	Werkzeuggeometrie L1
ZFP	Werkzeuggeometrie L2

2.5 Nullpunktverschiebung

Die Nullpunktverschiebung ist der Abstand des Werkstücknullpunktes (W) (von dem die Vermaßung ausgeht) zum Maschinennullpunkt (M).

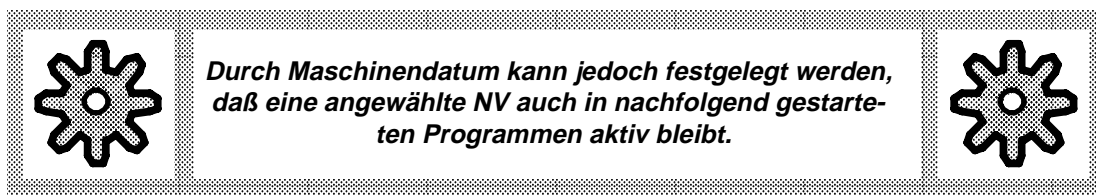


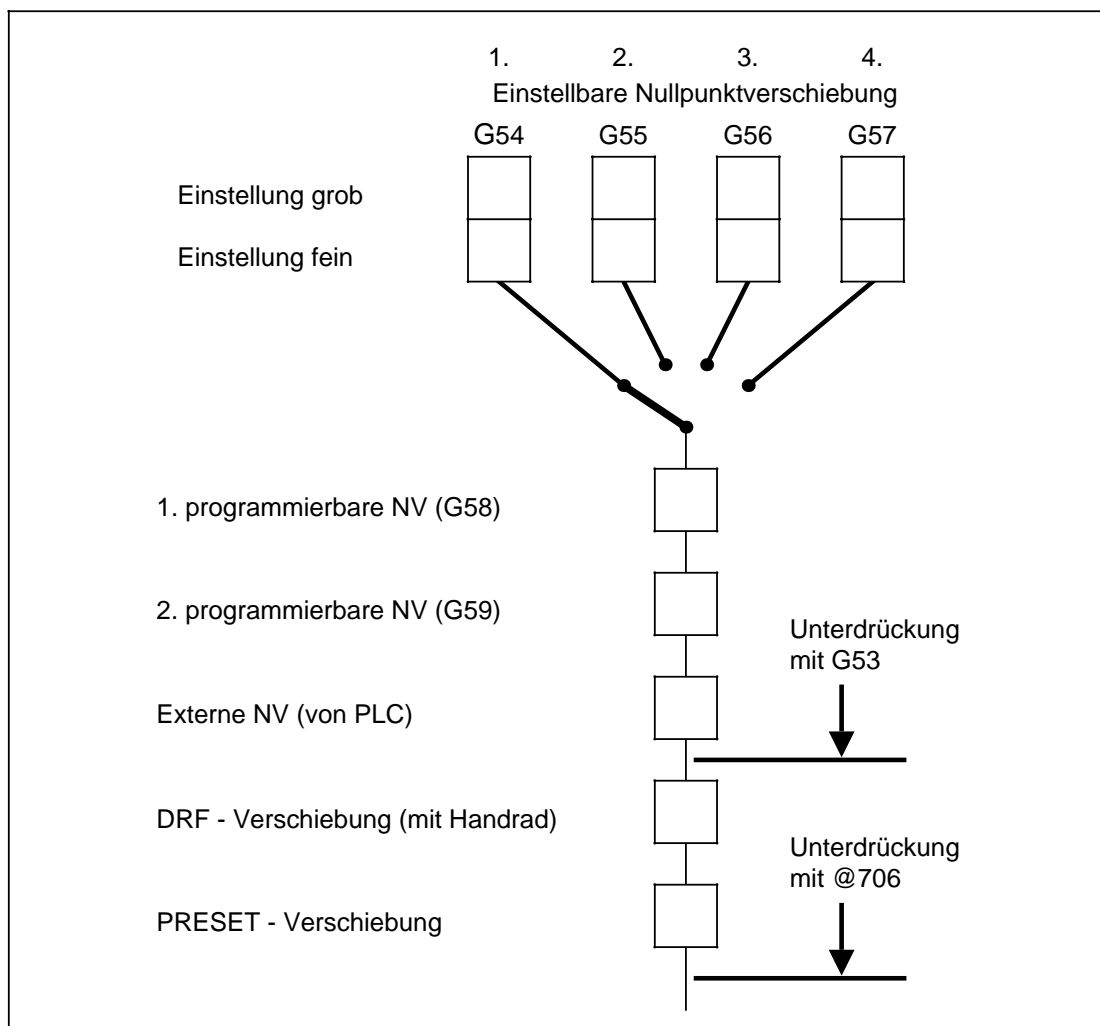
Nullpunktverschiebung

Folgende Nullpunktverschiebungen (NV) können aktiviert werden:

- einstellbare NV (G54 bis G57)
- programmierbare NV (G58, G59),
- externe NV (von PLC).

Die Nullpunktverschiebungen gelten für das aktuelle Programm.





Summe der Nullpunktverschiebung

Summe der Nullpunktverschiebungen =
 einstellbare NV (G54 bis G57) + programmierbare NV (G58, G59) + externe NV (von PLC).

Einstellbare Nullpunktverschiebung G54, G55, G56, G57

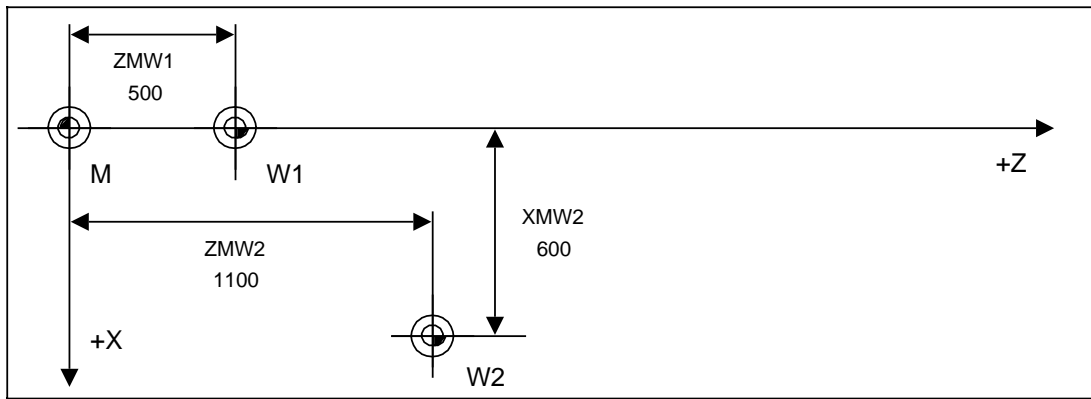
Die Werte für die einstellbare Nullpunktverschiebung je Achse können über die Bedientafel oder über die Universal-Schnittstelle in die Steuerung eingegeben werden.

Die Einrechnung erfolgt für den Satzendpunkt in Bezugsmaß- und Kettenmaßsätzen, wenn die betreffende Achse programmiert ist.

Mit G54 bis G57 kann eine der vier einstellbaren Nullpunktverschiebungen für die einzelnen Achsen angewählt werden.

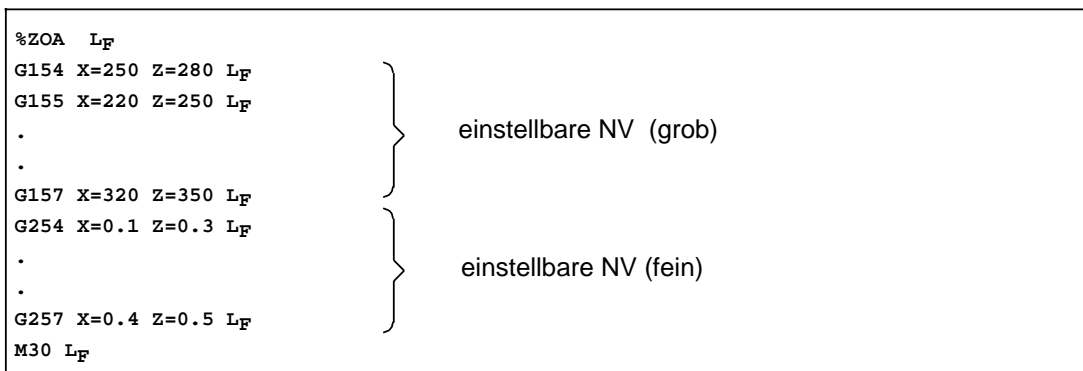
Die einzelnen einstellbaren NV gliedern sich in jeweils zwei NV (NV grob und NV fein), die additiv verrechnet werden.

Die NV fein wird als zusätzliche Feinverschiebung (Korrektur) des Nullpunkts eingesetzt.



Einstellbare Nullpunktverschiebung

Eingabe der einstellbaren Nullpunktverschiebung über die Universalschnittstelle:



Aus Kompatibilitätsgründen kann z.B. das Format **G54 X = 250 L_F** eingelesen werden, wobei dann die Werte in die einstellbare NV grob eingetragen werden.

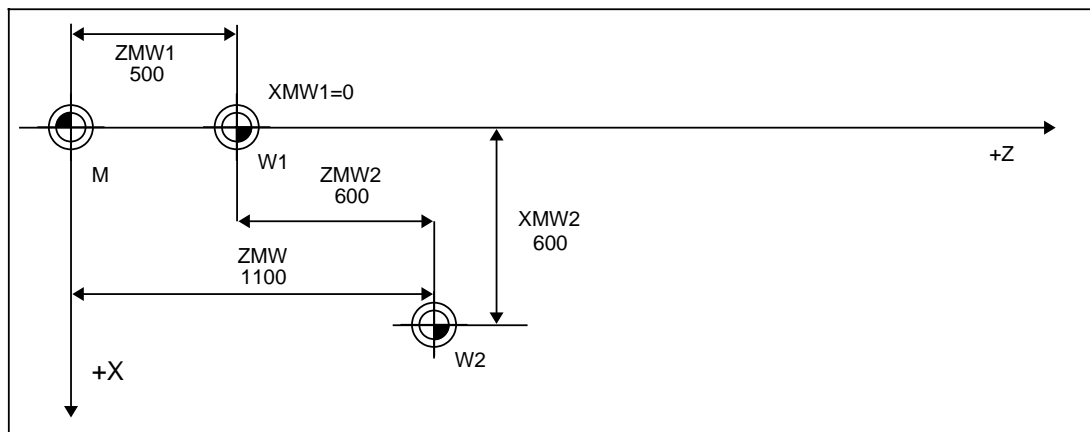
Programmierbare Nullpunktverschiebung G58/G59

Mit G58 und G59 kann unter der Achsadresse für alle vorhandenen Achsen eine zusätzliche Nullpunktverschiebung programmiert werden. Die programmierten Werte werden bei der Wegberechnung zu den Werten der einstellbaren Nullpunktverschiebung und der externen Nullpunktverschiebung addiert.

- **Einstellbare** Nullpunktverschiebung (grob und fein)
Eingabewerte: XMW1, ZMW1
- **Programmierbare** Nullpunktverschiebung
Eingabewerte: XMW2, ZMW2
- Gesamte wirksame Nullpunktverschiebung
 $XMW = XMW1 + XMW2$
 $ZMW = ZMW1 + ZMW2$

Programmierung:

```
N30 ...
N35 G54 LF
N40 G59 X600 Z600 LF
N45 ...
```

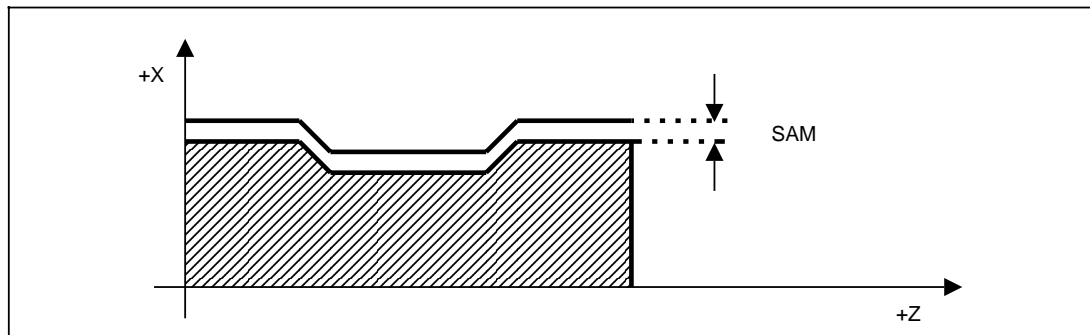


Einstellbare und programmierbare Nullpunktverschiebungen

In einem Satz mit G58 oder G59 dürfen außer den Nullpunktverschiebungen keine weiteren Funktionen geschrieben werden. Die Funktionen G58/G59 können auf max. 5 Achsen in einem Satz wirken.

Anwendungsbeispiel mit G59:

Die Kontur wurde ausschließlich im Bezugsmaß programmiert. Um ein **Schlichtaufmaß (SAM)** zu erreichen, kann die Gesamtkontur mittels einer **programmierbaren Nullpunktverschiebung** in der Koordinate X versetzt werden.



Nullpunktverschiebung mit G59

Anwahl: N..G59 X...LF

Abwahl: N..G59 X0 LF

Mit Programmende M02, M30 oder Programmabbruch werden die Werte der programmierbaren Nullpunktverschiebung, die in diesem Programm gesetzt worden sind, gelöscht. Mit RESET werden alle programmierbaren NV gelöscht.

G53 Abwahl der Nullpunktverschiebungen

Über NC-Maschinendatum wird vom Werkzeugmaschinenhersteller die Wirkungsweise von G53 festgelegt.

Mit G53 erfolgt eine satzweise Unterdrückung der durch

- einstellbare Nullpunktverschiebungen (G54 bis G57)
- programmierbare Nullpunktverschiebungen (G58, G59)
- externe Nullpunktverschiebung erreichten Koordinatenverschiebungen aus dem Maschinennullpunkt in den Werkstücknullpunkt.

Bei gesetztem Maschinendatum lassen sich mit G53 zusätzlich satzweise unterdrücken:

- DRF-Verschiebung
- PRESET-Verschiebung.

In diesem Fall wirkt G53 wie @706 (siehe Kap 11.9).

Die Werkzeugkorrektur muß getrennt abgewählt werden.

Im nächsten Satz nach G53 sind alle Nullpunktverschiebungen wieder wirksam.

Bezug auf Maschinennullpunkt:

N30 D0 L_F

Abwahl der Werkzeugkorrektur

N35 G53 x... z...L_F

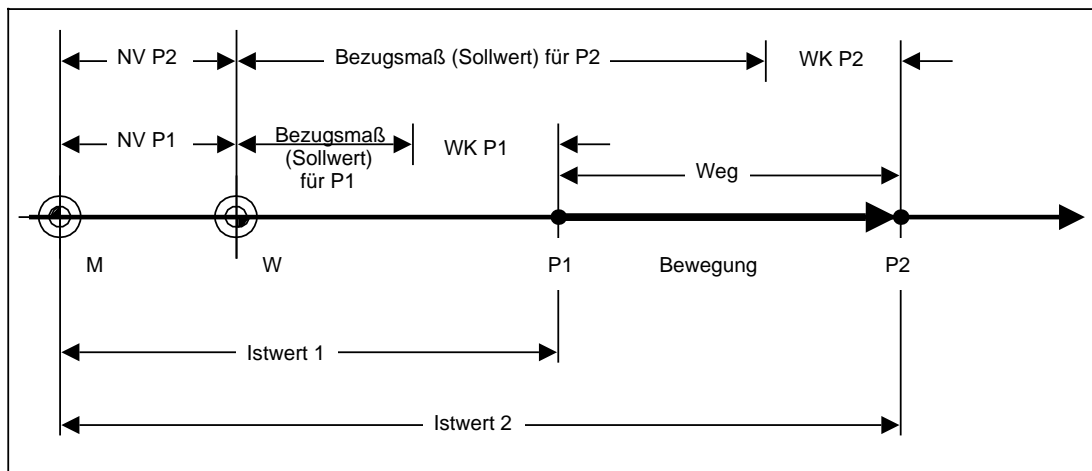
Abwahl aller NV und Fahren auf Position im Maschinensystem

2.6 Wegberechnung

Die Wegberechnung ermittelt die in einem Satz zu verfahrenende Wegstrecke unter Berücksichtigung aller Verschiebungen und Korrekturen.

Allgemein gilt:

$$\text{Weg} = \text{Sollwert} - \text{Istwert} + \text{Nullpunktverschiebung (NV)} + \text{Werkzeugkorrektur (WK)}$$



Wegberechnung bei Bezugsmaßeingabe

Bei **Kettenmaßeingabe** wird im ersten Satz die Nullpunktverschiebung normal eingerechnet:

$$\text{Weg} = \text{Kettenmaß} + \text{NV} + \text{WK}$$

Wird in einem neuen Programmsatz eine neue Nullpunktverschiebung und eine neue Werkzeugkorrektur programmiert, so gilt:

- Bei **Bezugsmaßeingabe**

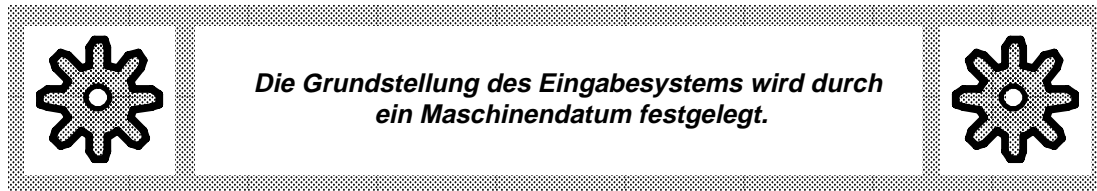
$$\text{Weg} = \text{Bezugsmaß P2} - \text{Bezugsmaß P1} + \text{NVP2} - \text{NVP1} + \text{WKP2} - \text{WKP1}$$

- Bei **Kettenmaßeingabe**

$$\text{Weg} = \text{Kettenmaß} + \text{NVP2} - \text{NVP1} + \text{WKP2} - \text{WKP1}$$

2.7 Werkstückvermessung, Eingabesystem G70/G71

Die Maßeinheiten können bei der Programmierung in mm oder in inch eingegeben werden.



Ein Wechsel des Eingabesystems muß über die Wegbedingung G70 bzw. G71 angewählt werden:

G70 Eingabesystem inch
G71 Eingabesystem metrisch (mm).

Die Steuerung rechnet den eingegebenen Wert in das Eingabesystem der Grundstellung um. Beim Abarbeiten eines solchen Satzes wird bereits der umgerechnete Wert im System der Grundstellung angezeigt.

Vor der Anwahl von Unterprogrammen oder Zyklen muß auf **gleiche Maßeinheit** geachtet werden.

Die von der Grundstellung abweichende Maßeinheit kann für einen oder mehrere Sätze oder ein ganzes Programm festgelegt werden.

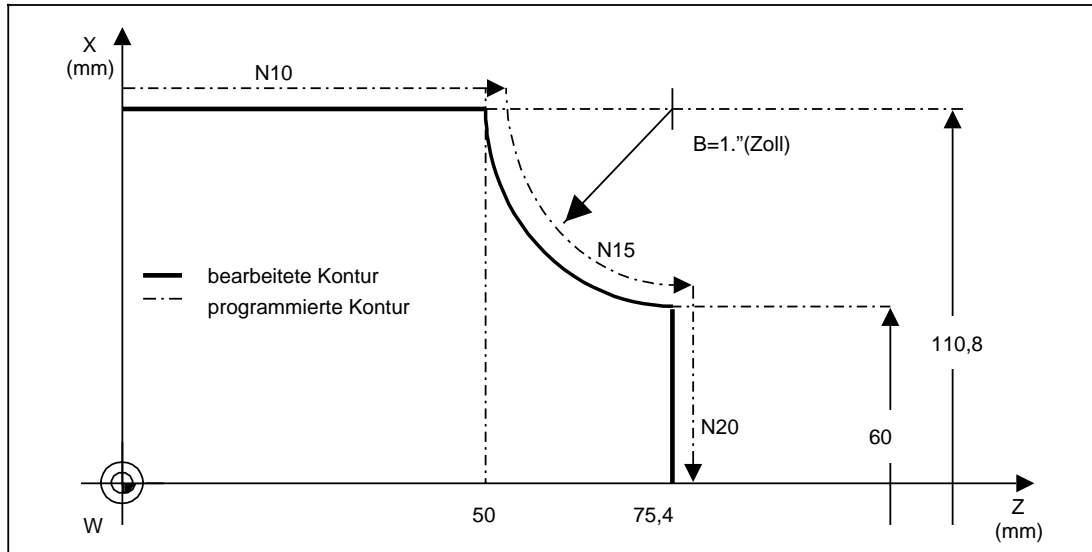
Im ersten Satz muß dann die erforderliche G-Funktion, und nach dem letzten Satz wieder die Grundstellung geschrieben werden (bei Programmende M02, M30 kommt automatisch die Grundstellung).

Von der **Grundeinstellung** des Eingabesystems sind abhängig:

- Istwertanzeige (auch Soll/Ist-Differenz)
- Nullpunktverschiebung
- Vorschubgeschwindigkeit/Schnittgeschwindigkeit G94/G95
- Werkzeugkorrektur.

Vom **programmierten G70 oder G71** sind abhängig:

- Weginformationen X, Z
- Interpolationsparameter I, K
- Fasen/Radien B-/B
- Parameter, soweit diese Weginformationen, Interpolationsparametern und Fasen/Radien zugeordnet sind.

Beispiel: G71 - Grundstellung (metrisch)

Eingabe in inch bei Grundstellung G71

N05 . . .

N10 G91 Z50 L_F

N15 G03 G70 X - 1 Z1 K1 L_F

N20 G01 G71 X - 30 L_F

N25 . . .

Kreisbogen in Inch programmiert

Gerade im metrischen Maßsystem programmiert

2.8 Spiegeln

Spiegeln einer Achse

Durch Spiegeln einer Koordinatenachse wird eine Kontur

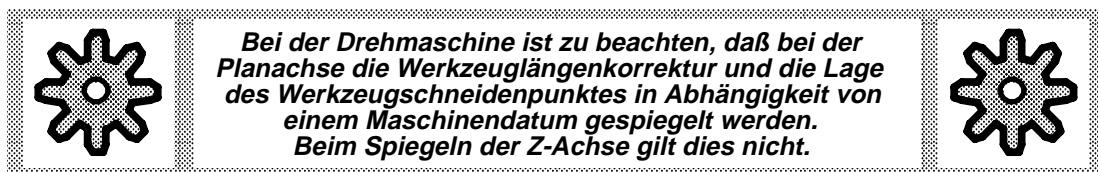
- in derselben Größe,
- im gleichen Abstand zu den anderen Achsen,
- auf der anderen Seite der Spiegelachse spiegelbildlich bearbeitet.

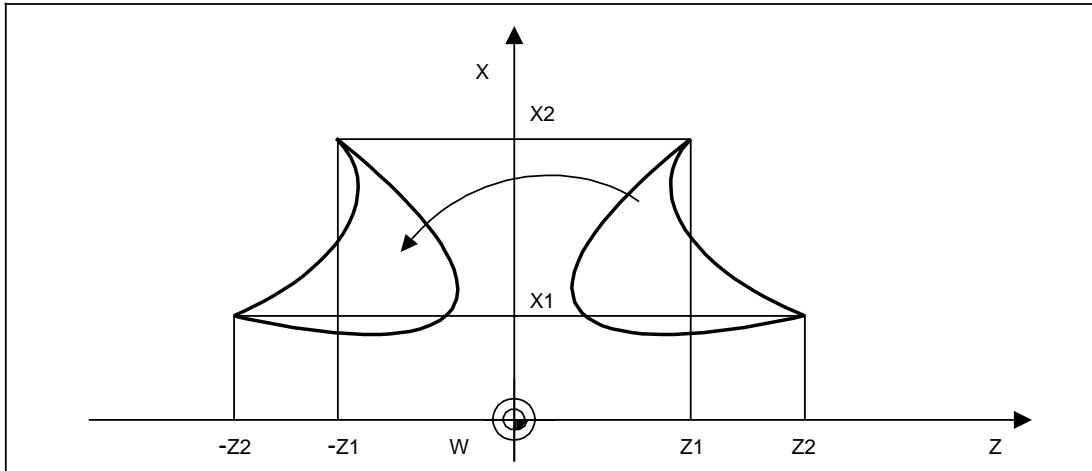
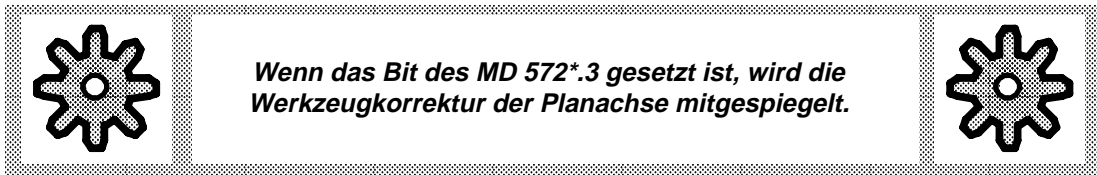
Beim Spiegeln einer Achse vertauscht die Steuerung

- die Vorzeichen der Koordinaten der gespiegelten Achse,
- den Drehsinn bei der Kreisinterpolation (G02 G03, G03 G02),
- die Bearbeitungsrichtung (G41 G42, G42 G41).

Keine Spiegelung von:

- Werkzeuglängenkorrekturen
- Nullpunktverschiebungen.

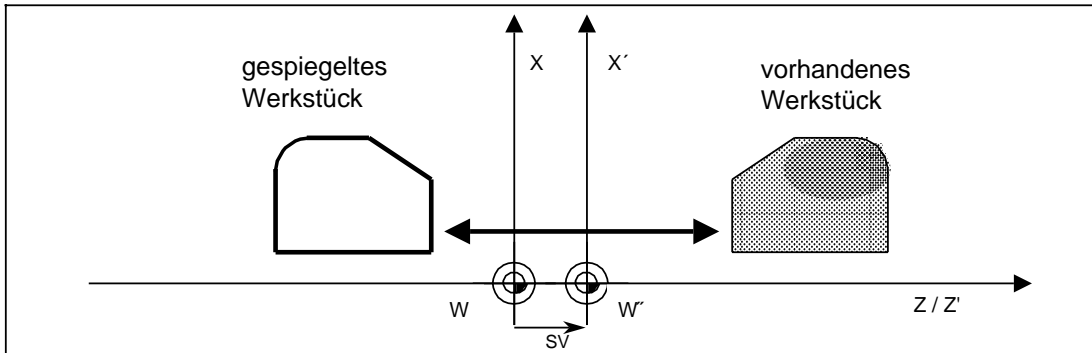




Spiegeln der Z-Achse

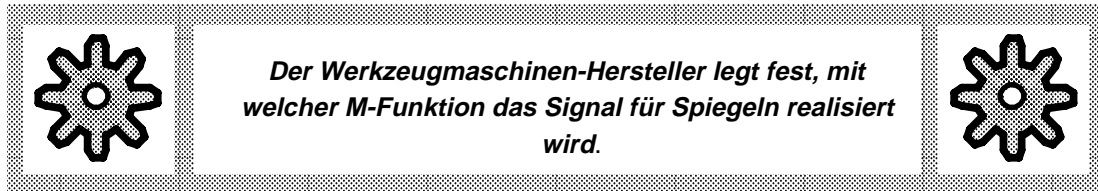
Die Spiegelung erfolgt immer um die Koordinatenachse. Damit die Konturen genau dorthin gespiegelt werden, wo sie bearbeitet werden sollen, muß beim Aufruf der Spiegelung der Programmnullpunkt so liegen, daß die Achsen des Koordinatensystems genau zwischen programmierter und gespiegelter Kontur liegen.

Falls erforderlich, wird der Nullpunkt des Koordinatensystems vor Aufruf der Spiegelung im Programm an die richtige Stelle verschoben (W nach W'). Um das Werkstück auf die richtige Position zu spiegeln, muß der Nullpunkt um den Wert "SV" verschoben werden. Dadurch wird erreicht, daß der Abstand der beiden Werkstücke zum Nullpunkt gleich groß ist. Nach der Spiegelung kann der Nullpunkt auf die ursprüngliche Position zurückgesetzt werden.



Verschiebung des Werkstücknullpunkts

Die Funktion "Spiegeln" wird über die PLC angewählt.



Beispiel: Anwahl von Spiegeln

```

:
N10 G90 G54 G00 X0 Z0 L_F
N20 Z30 L_F
N30 G1 Z0 F500 L_F
N35 M... L_F           Z-Achse spiegeln
N36 G04 F... L_F      Verweilzeit ev. wegen PLC-Zykluszeit
N37 L999 P1 L_F       Zwischenspeicher leeren
:
N40 X50 Z50 M30 L_F

```

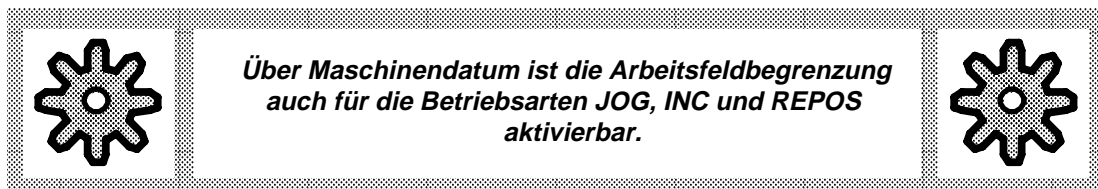
Mit der Sonderfunktion @714 (Zwischenspeicher leer) kann eine weitere Satzinkrement-Berechnung angehalten werden bis der Zwischenspeicher leer ist.

Der Einsatz von @714 bzw. L999 ist bei allen extern zu beeinflussenden Verschiebungen, z.B. Spiegeln notwendig.

2.9 Programmierbare Arbeitsfeldbegrenzung G25/G26

Die programmierbare Arbeitsfeldbegrenzung bewirkt einen **Maschinenschutz** bei Programmier- und Bedienfehlern.

Der Schlittenbezugspunkt F darf sich nur im begrenzten Bereich bewegen. Sobald das Werkzeug den begrenzten Bereich verläßt, sich beim Programmstart außerhalb dieses Bereiches befindet, oder eine Position programmiert wird, die außerhalb der Arbeitsfeldbegrenzung liegt, wird die Wegvorgabe abgebrochen oder kein Fahrbefehl angenommen (Programmstop, kein Programmstart, Alarm). Der aktuelle Schleppabstand wird noch abgebaut. Die programmierbare Arbeitsfeldbegrenzung ist im Automatik-Betrieb mit den in den Settingdaten hinterlegten Werten wirksam.



Die programmierbare Arbeitsfeldbegrenzung wird mit G25 und G26 aufgerufen:

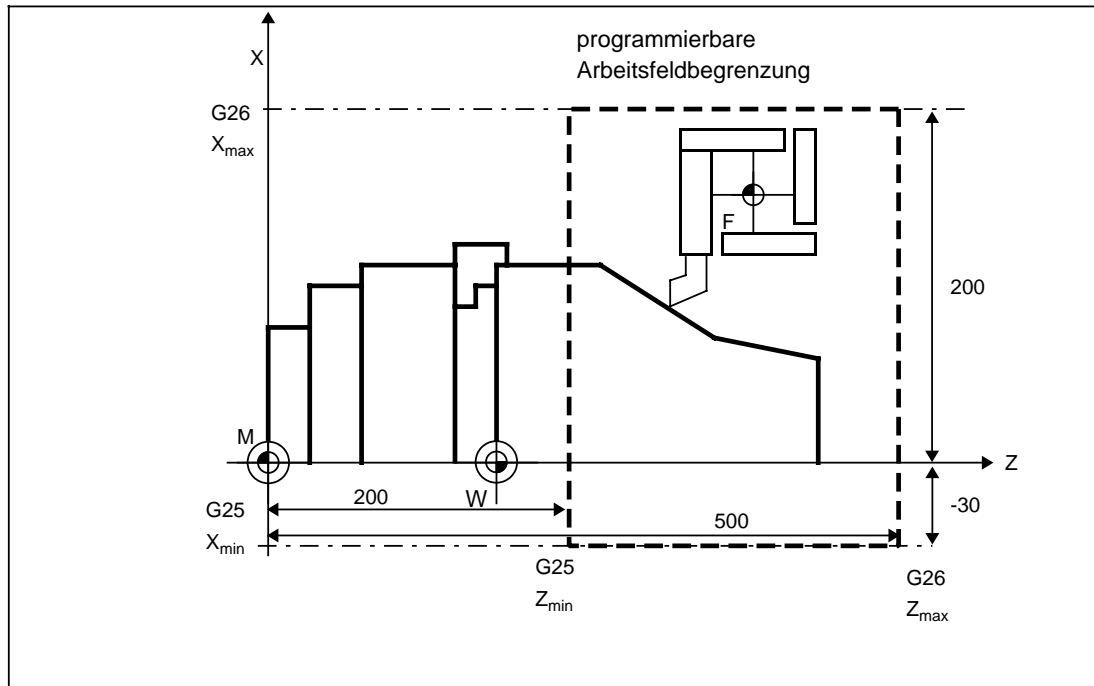
G25 minimale Arbeitsfeldbegrenzung
G26 maximale Arbeitsfeldbegrenzung

Beispiel:

N10 G25 X-30 Z200 L_F
N20 G26 X200 Z500 L_F

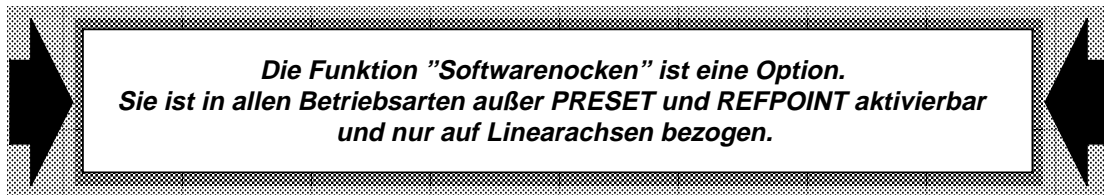
Weitere Angaben sind in diesem Satz nicht zulässig. Mit der Eingabe von -99999.999 für den Minimalwert und +99999.999 für den Maximalwert pro Achse in den Settingdaten wird die Arbeitsfeldbegrenzung unwirksam.

Beispiel: Drehmaschine



2.10 Softwaresnocken

Die Umrüstzeit bei Bearbeitungsvorgängen an einem Werkstück kann man reduzieren, indem man den Bearbeitungstisch in zwei Arbeitsfelder aufteilt. Während im ersten Feld ein Werkstück bearbeitet wird, kann im zweiten ein Werkzeug gewechselt werden. Mit Hilfe der Funktion "Softwaresnocken" können Sie den Bearbeitungstisch in zwei Arbeitsfelder aufteilen.



Die Funktion "Softwaresnocken" erzeugt **Nockensignale** und ist über R-Parameter und Maschinendaten parametrierbar. Die R-Parameter enthalten die Achspositionen der einzelnen Nocken (**Nockenpositionen**) und sind zu einem Nockenparameterblock zusammengefasst. Bei 5 realen Achsen, wie bei SINUMERIK 810T/M, 820T/M, können bis zu zehn Nocken eingerichtet werden. Jeweils zwei Nocken bilden ein **Nockenpaar**. Die Nummern der R-Parameter für die jeweiligen Nocken erfahren Sie vom Werkzeugmaschinen-Hersteller.

Nockensignale:

Nockensignale sind **Steuersignale** der NC. Sie bilden einen Schaltnocken mit unendlicher Länge nach, der in einer bestimmten Anfahrrichtung an der Nockenposition aktiviert wird. Die Nockensignale werden vom PLC-Programm ausgewertet.

- Nockensignale werden erst nach dem Referieren der Achsen ausgegeben.
Ausnahme: Achsen, bei denen keine Startsperrung vor Referenzpunktfahren gesetzt ist.
- Die Nockenpositionen beziehen sich auf das Maschinensystem.

Nockenpaar und Nockenbereich:

Ein Nockenpaar besteht aus einem **Plusnocken** und einem **Minusnocken**. Dem Plusnocken ist der Achsbereich größer seiner Nockenposition und dem Minusnocken der Achsbereich kleiner seiner Nockenposition zugeordnet.

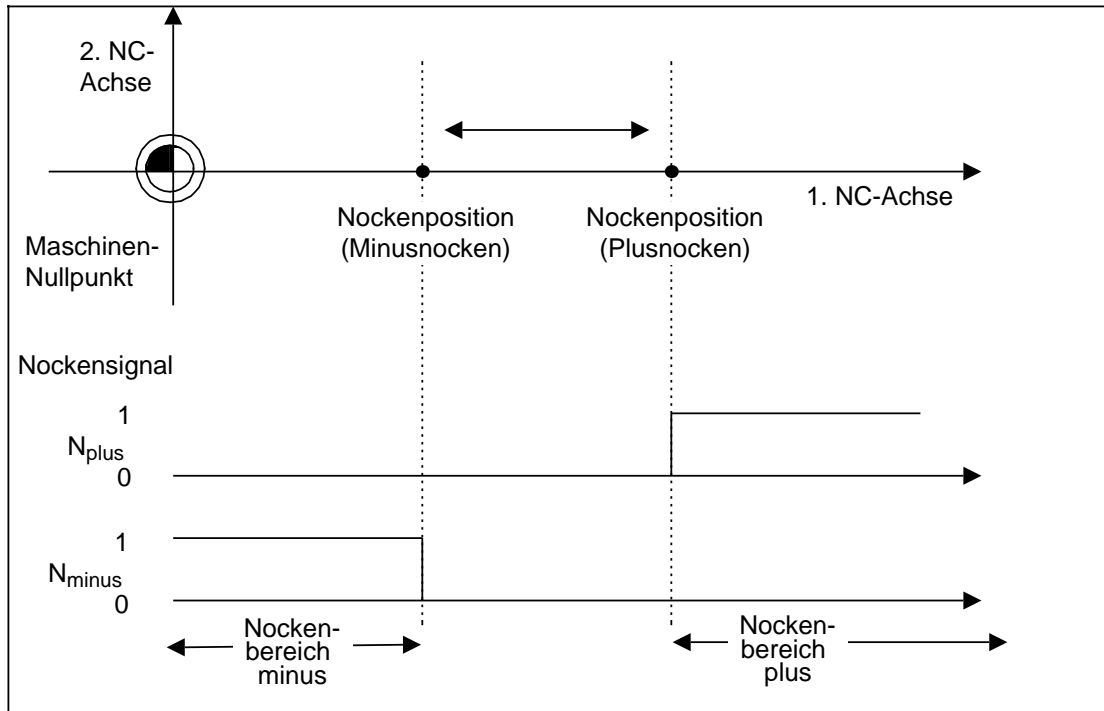
Die Nockenpositionen müssen sich auf das jeweilige Maschinensystem (metrisch oder inch) beziehen. Mit @361 kann die maschinenbezogene Achsposition gelesen werden. Eine Überprüfung der Nockenpositionen hinsichtlich des maximalen Verfahrbereichs erfolgt nicht.

Der dem Nocken zugeordnete Achsbereich wird als **Nockenbereich** bezeichnet.

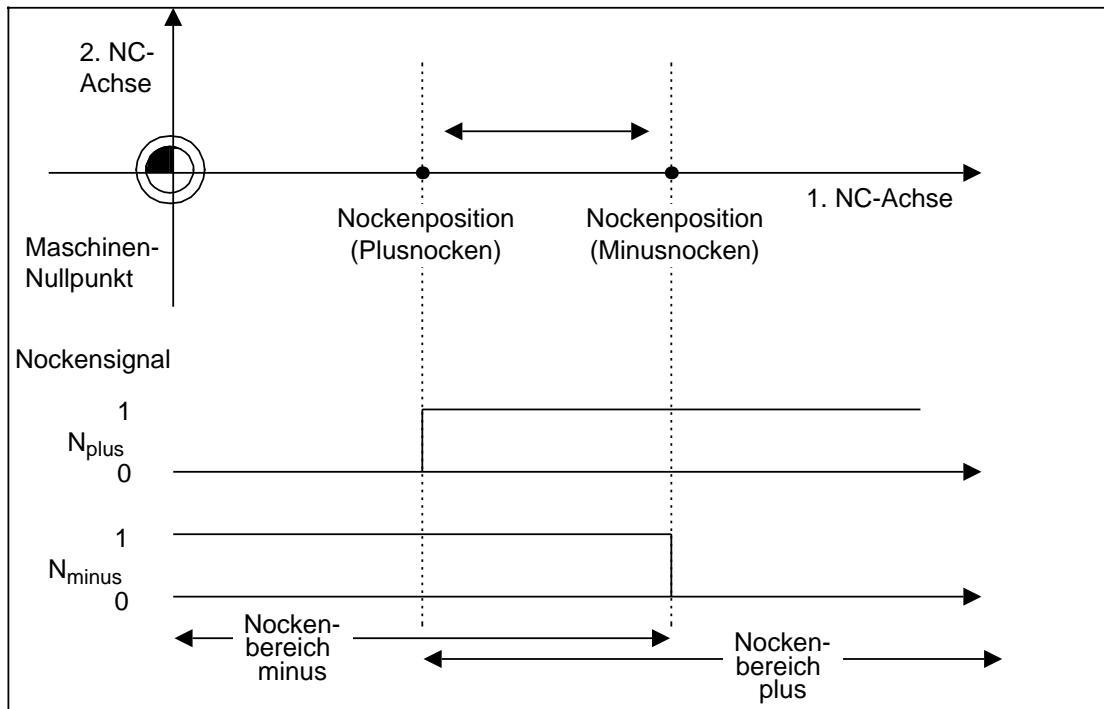
Ein Nockenpaar kann immer nur einer NC-Achse zugeordnet werden,

aber:

Für eine Achse können auch **mehrere** Nockenpaare aktiviert werden.



Minusnocken < Plusnocken



Plusnocken < Minusnocken

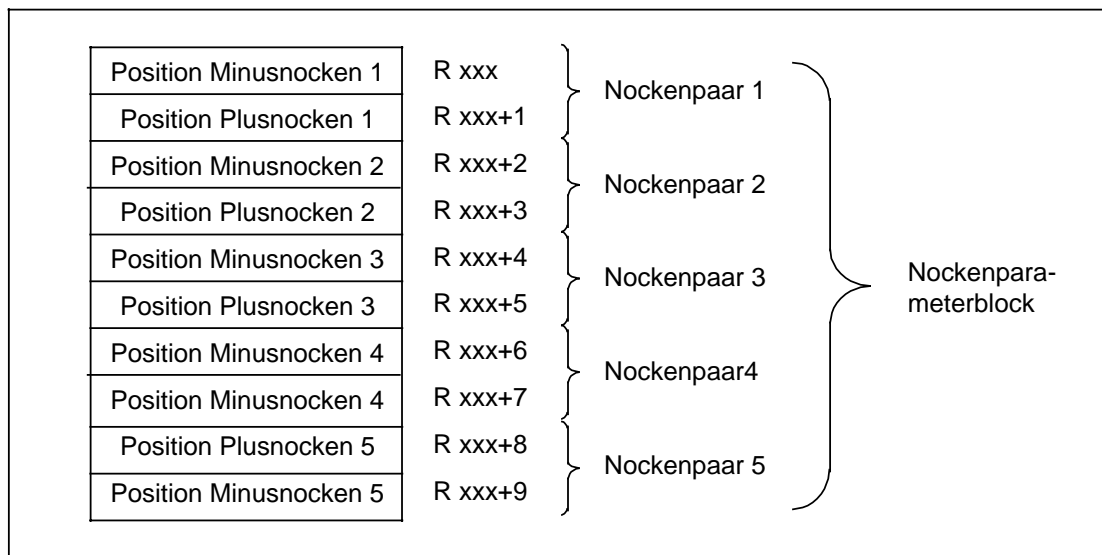
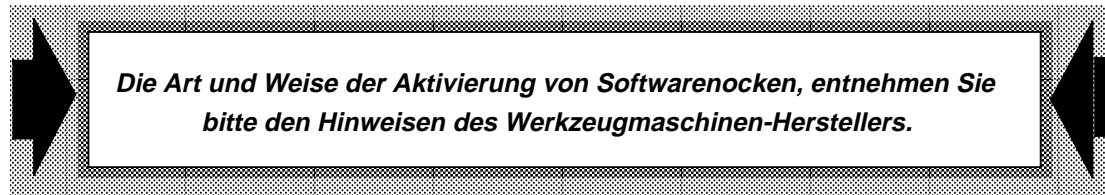
Nockenparameter:

Alle Nockenparameter sind in einem R-Parameterblock zusammengefaßt. Der R-Parameterblock wird als Nockenparameterblock bezeichnet.

In 5 aufeinanderfolgenden R-Parameterpaaren (**Nockenpaaren**) werden die Werte der Achspositionen für die 10 SW-Nocken hinterlegt. Je Nockenpaar gibt es einen R-Parameter für die negative Nockenrichtung und einen für die positive Nockenrichtung. Ein Nockenparameterblock kann aus den globalen oder den kanalabhängigen R-Parametern gebildet werden.

Der Anfang des Nockenparameterblockes (Rxxx), d.h. die Nummer xxx des ersten R-Parameters, wird vom Werkzeugmaschinen-Hersteller festgelegt.

Die Zuweisung Achse/Nockenpaar wird ebenfalls vom Werkzeugmaschinen-Hersteller durchgeführt und Ihnen von diesem mitgeteilt.

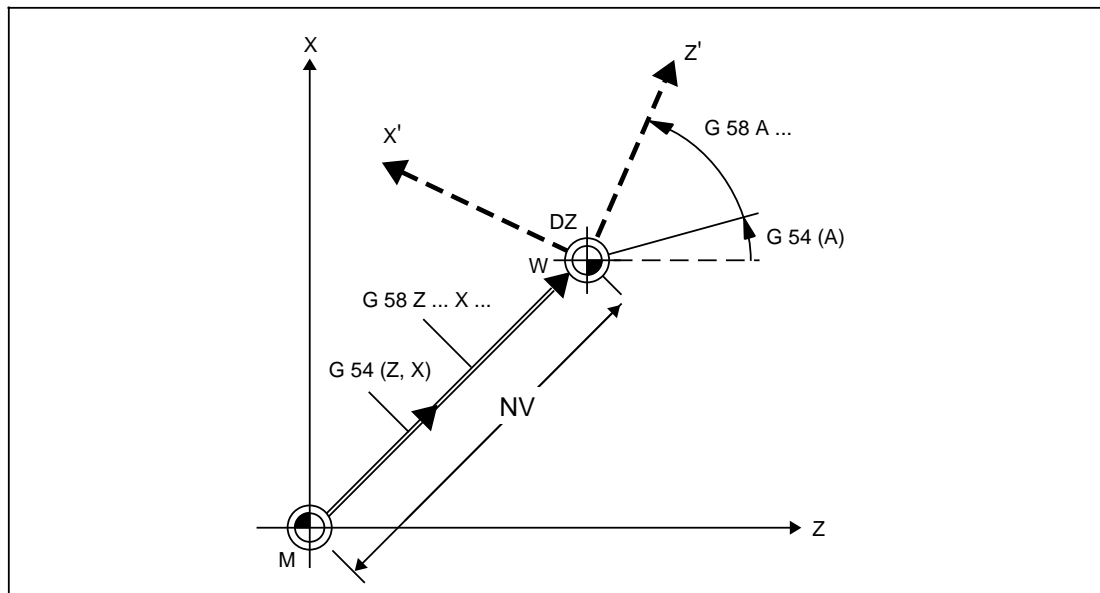


Festlegung der Nockenparameter

2.11 Koordinatendrehung (KD)

Mit der Koordinatendrehung (KD) kann das Koordinatensystem des Werkstücks an das Koordinatensystem der Maschine angepaßt werden. Der Ablauf eines Teileprogramms (die Bearbeitung) erfolgt dann im gedrehten Koordinatensystem.

Das Drehzentrum (DZ) ist der durch die Summe der Nullpunktverschiebungen (NV) bestimmte Punkt.



Folgende Koordinatendrehungen können aktiviert werden:

einstellbare Koordinatendrehung (G54 bis G57)

programmierbare Koordinatendrehung (G58 A... und G59 A...)

Die Summe von "einstellbarer Koordinatendrehung" und "programmierbarer Koordinatendrehung" ergibt die "wirksame Koordinatendrehung".

Einstellbare Koordinatendrehung:

- Es gibt keine Unterscheidung zwischen Grob- und Feineinstellungen.
- Die einstellbare Koordinatendrehung wird über Settingdaten festgelegt.
- Eingabe des Wertes für Drehwinkel "A" über Tastatur in das Eingabefeld KD.
- Über die Universalschnittstelle kann der Drehwinkel "A" für G54 ... G57 oder für G154 ... G157 eingegeben werden (die Eingabe für G254 ... G257 ist **nicht** zulässig).

Mit G54 wird die in den Settingdaten für G54 eingetragene NV und die im Settingdatum eingetragene KD G54 A... aktiviert.

Beispiel: einstellbare Koordinatendrehung

N...

N... G90 G54 L_F

N...

Bei aktiver Koordinatendrehung dürfen die betroffenen Achsen nur mit Bahnvorschub programmiert werden.

Programmierbare Koordinatendrehung:

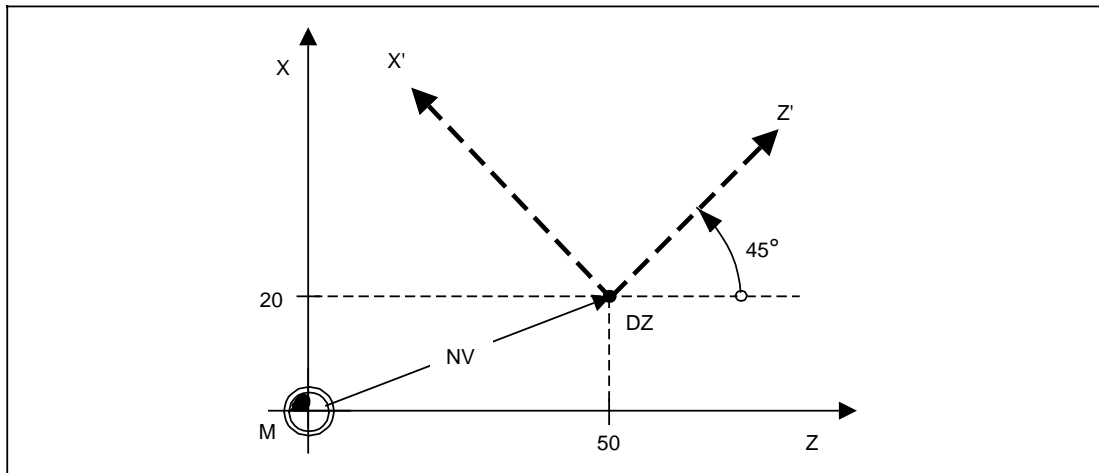
- die Koordinatendrehung kann mit G90 (Bezugsmaß) oder G91 (Kettenmaß) programmiert werden
- der Wert des Drehwinkels wird unter der Adresse "A..." programmiert.

Beispiel: programmierbare Koordinatendrehung

N...

N... G90 G58 X20 Z50 A45 L_P

N...

**Beeinflussung durch Programmierung**

- G53 bzw. @706: KD und NV werden satzweise unterdrückt.
- M02, M30: programmierbare KD und NV werden gelöscht, die einstellbare KD und NV bleiben dagegen erhalten (Settingdaten).

Weitere Kennzeichen:

- Die Koordinatendrehung ist in der Betriebsart "AUTOMATIC" ablauffähig.
- Die KD wirkt **kanalspezifisch**; bei der "SIMULATION" werden die für **Kanal 3** eingegebenen Koordinatendrehungen wirksam.
- Bei der "ISTWERTANZEIGE" wird die KD **nicht** berücksichtigt.
- Unmittelbar nach einer KD darf keine Kreisinterpolation programmiert werden.
- Bei angewählter SRK:
 - das Drehzentrum (Summe der NV) darf **nicht** verändert werden
 - der Drehwinkel "A" der einstellbaren und programmierbaren KD kann abgeändert werden.
- Die Koordinatendrehung ist mit der Funktion "Zwischenspeicher leeren bis Koordinaten" drehung (@715) verbunden. Diese Funktion wird von der Satzaufbereitung intern ausgelöst, muß daher also nicht programmiert werden.

Änderungen bei der Nullpunktverschiebung und der Ebenenwahl müssen Sie vor der Anwahl der Koordinatendrehung einstellen bzw. programmieren.

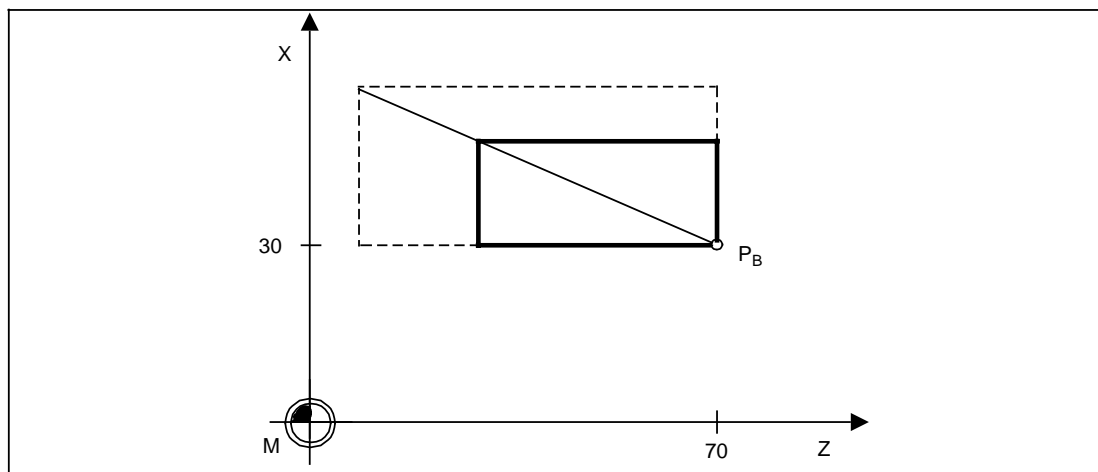
2.12 Maßstabsänderung: Anwahl G51, Abwahl G50

Mit dieser Funktion kann der Programmieraufwand für geometrisch ähnliche Teile stark reduziert werden.

- Eine Maßstabsänderung wird mit der Programmierung von G51 wirksam.
- G51 ist selbsthaltend und wird erst durch G50 (Abwahl der Maßstabsänderung) aufgehoben.
- Die Maßstabsänderung orientiert sich an einem Bezugspunkt P_B (Maßstabszentrum). Die Koordinaten (Z, X) von P_B werden beim Programmieren angegeben. Werden Z und X nicht angegeben, so generiert die Steuerung die Werte $Z = 0$ und $X = 0$ (Bezugspunkt = Werkstücknullpunkt).
- Der Wert für die Maßstabsänderung (Maßstabsfaktor) wird unter der **Adresse "P..."** im **Bereich 0.00001 bis 99.9999** angegeben.
- Durch die Maßstabsänderung werden folgende Werte umgerechnet
 - Achskoordinaten
 - Interpolationsparameter
 - Radius
 - Programmierbare NV
 - Gewindesteigung, Steigungsabnahme bzw. -zunahme.
- Eine Maßstabsänderung ist mit "Zwischenspeicher leeren" @ 714 verbunden. Diese Funktion wird von der Satzaufbereitung intern ausgelöst.

Beispiel: Mögliche Schreibweisen bei der Programmierung

- a) N...
N... G51 Z70 X30 P1.5 L_F
N...
oder
- b) N...
N... G51 P1.5 L_F
N... G51 Z70 X30 L_F
N...



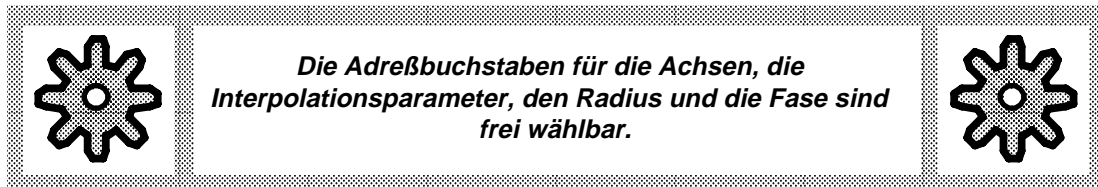
**Maßstabsfaktor (kanalspezifisch) und Bezugspunkt werden auf dem Bildschirm unter "SETTINGDATEN" angezeigt.
Die Maßstabsänderung wird für die jeweilige Achse durch das Setting-Datum 560*, Bit 2 freigegeben.**

3 Programmieren von Bewegungssätzen

3.1 Achsbefehle

Die Adresse des Achsbefehls bestimmt die Achse, in welcher der nachfolgende Zahlenwert verfahren werden soll, z.B. X, Z.

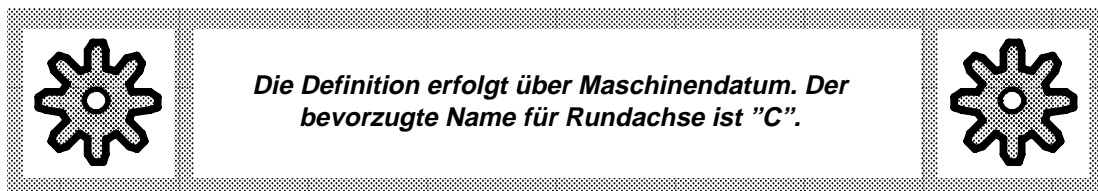
Für weitere Achsen stehen wahlweise die Adressen A, B, C, U, V, W und Q zur Verfügung. Die erweiterte Adreßschreibweise ist möglich (z.B.: C1=...).



Nachfolgend werden die vorzugsweise bei Drehmaschinen benutzten Adreßbuchstaben verwendet.

Rundachse

Jede Achse kann als Rundachse deklariert werden.



- Bei Bezugsmaßprogrammierung (G90) beträgt der Verfahrbereich ± 360.000 Grad.

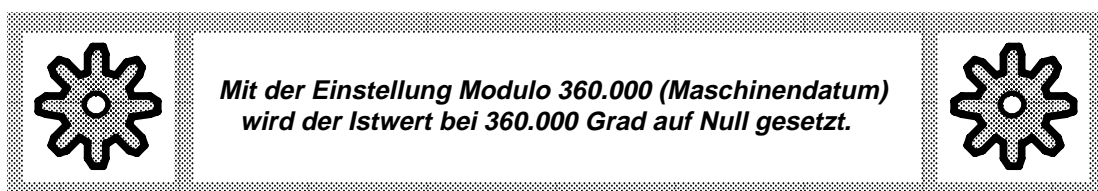
Das Vorzeichen gibt die Verfahrrichtung an, mit der auf die absolute Position innerhalb einer Umdrehung positioniert wird.

Bei Bezugsmaßprogrammierung auf kürzestem Weg (G68) beträgt der Verfahrbereich 0 bis +360.000 Grad.

Die Steuerung sucht sich aufgrund der aktuellen Position der Rundachse den kürzesten Weg zur programmierten Position und bestimmt daher automatisch die Verfahrrichtung.

Bei der erstmaligen Programmierung der Rundachse mit "G90" im Teileprogramm wird für diesen ersten Satz automatisch "G68" aktiv. Diese automatische Generierung kann mit "G91 C0 L_F" abgewählt werden.

Die oben gegebenen Hinweise für G68 gelten auch, wenn auf ein Teileprogramm erstmalig der "Satzvorlauf" angewendet wird.



- Bei Kettenmaßprogrammierung (G91) ist der Verfahrbereich ± 99999.999 Grad zu beachten. Das Vorzeichen gibt nur die Verfahrrichtung an.

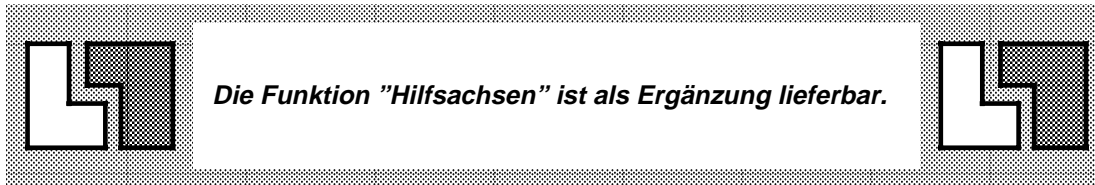
Es können **mehrere Achsen gleichzeitig** als Rundachsen erklärt werden.
Die Rundachsen können generell endlos drehen.

Programm-Ausschnitt:

·
·
·

```
N5 G91 C99999.999 L_F
```

Hilfsachsen



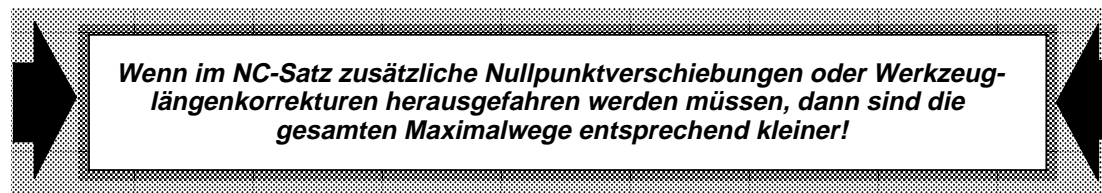
Der Begriff "**Hilfsachse**" kennzeichnet eine Achse, die nicht zur eigentlichen Werkstückbearbeitung, sondern für die Werkstück- oder Werkzeughandhabung eingesetzt wird (Lader, Revolver, Magazin usw.).

Hilfsachsen werden im Bearbeitungskanal oder durch einen eigenen NC-Kanal geführt. Sie enthalten den Funktionsumfang einer NC-Hauptachse (Linearinterpolation, Kreisinterpolation, Konturzug, Werkzeugkorrektur usw.).

Sofern die Hilfsachse nicht als "**NC-Achse**" im Teileprogramm programmiert wird, kann eine Synchronisierung zu den NC-Hauptachsen auch über die Anpaßsteuerung realisiert werden. Wegmessung, Verfahrbereich, Eingabefeinheit, Geometrie- und Lageregeleinheit entsprechen einer NC-Achse.

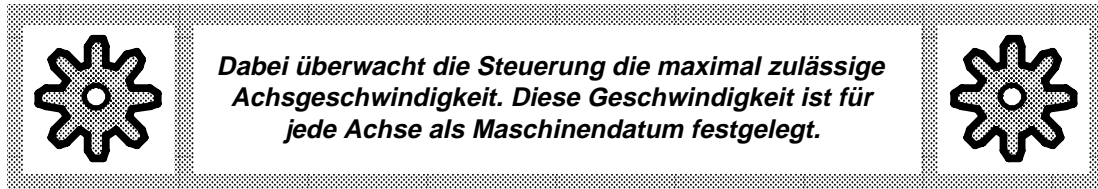
Die Achsadressen sind innerhalb der verfügbaren Adressen frei wählbar. Bei der Vorzugsadresse Q können sie mit einer erweiterten Adreßschreibweise unterschieden werden, z.B.: Q1, Q2 ...

Wird die Hilfsachse durch einen eigenen Kanal geführt, so kann kein Konturzug und keine Werkzeugkorrektur programmiert werden. Die Synchronisierung zwischen NC-Achsen und den Hilfsachsen wird dann durch die Anpaßsteuerung (Anwenderprogramm) realisiert.



3.1.1 Achsbewegung ohne Bearbeitung G00

- Eilgangbewegungen werden durch die Weginformation G00 und durch Angabe des Zielpunktes programmiert. Der Zielpunkt kann durch Eingabe im Bezugsmaß (G90) oder im Kettenmaß (G91) erreicht werden.
- Der mit G00 programmierte Weg wird mit der **größtmöglichen Geschwindigkeit**, dem Eilgang, **ohne Bearbeitung** des Werkstücks auf einer Geraden verfahren (Geradeninterpolation).



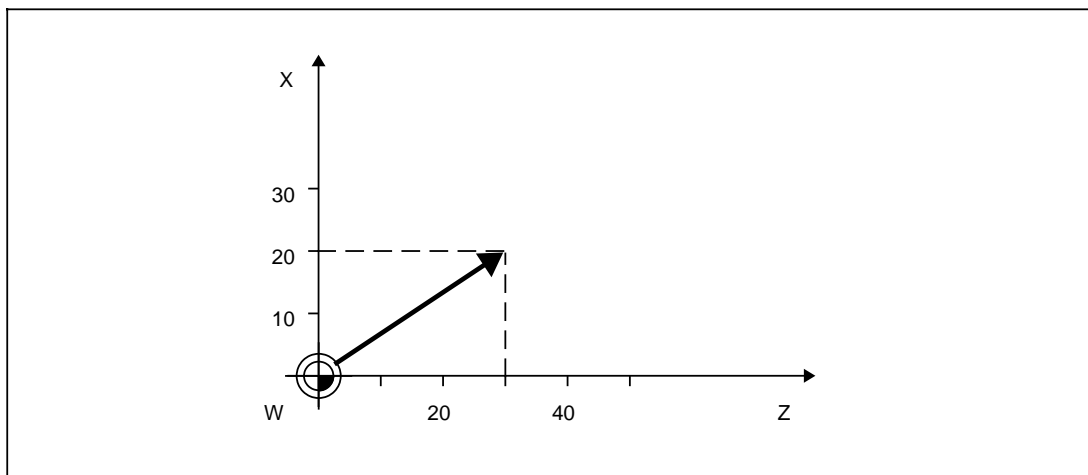
Wird die Eilgangbewegung in mehreren Achsen gleichzeitig ausgeführt, so wird die Verfahrensgeschwindigkeit durch den kleinsten Wert der als MASCHINENDATUM festgelegten Achsgeschwindigkeiten bestimmt. Die Wegbedingung G00 bewirkt automatisch "Genauhalt grob".

Bei der Programmierung von G00 bleibt der unter der Adresse F programmierte Vorschub gespeichert und wird z.B. mit G01 wieder wirksam.

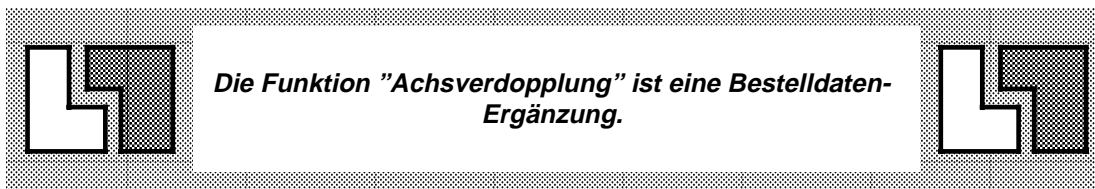
Beispiel:

N5 G00 G90 Z30 X20 L_F

Eilgang, Bezugsmaßvorgabe



3.1.2 Achsverdopplung



Die Funktion "Achsverdopplung" bewirkt eine Verdopplung der programmierbaren Hauptachsen im Rahmen der Anzahl der maximal zugelassenen Achsen (bei 810T/M,820T/M: 5 Achsen).

Damit lassen sich über ein Teileprogramm, das in einem Kanal der NC abläuft, gleichzeitig zwei gleiche Werkstücke von einer Werkzeugmaschine mit zwei Werkzeugsystemen produzieren.

Die Funktion "Achsverdopplung" ist nur in den Betriebsarten "AUTOMATIC", "MDI-AUTOMATIC" und bei "Automatic unterbrochen" verfügbar. Bei Achsverdopplung ist für jede Achse - auch für die verdoppelten Achsen - ein Meßkreis vorgesehen. Es sind somit maximal zwei Achsen verdoppelbar. Die 5. Achse steht dann noch für eine unabhängige Achse zur Verfügung.

Über die Achsbezeichnungen bei der Funktion "Achsverdopplung" informiert Sie der Werkzeugmaschinen-Hersteller.

3.1.2.1 Funktionsweise

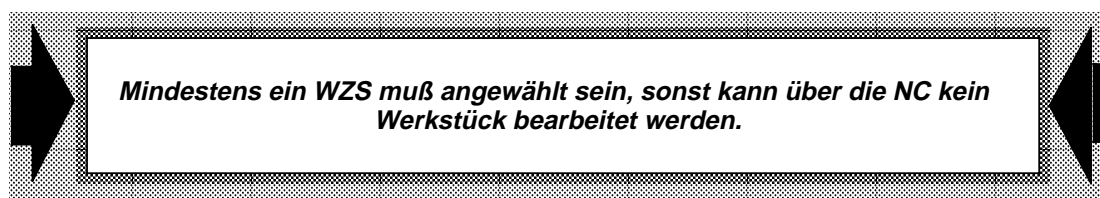
Mit der Funktion "Achsverdopplung" arbeitet die NC **ein** Teileprogramm simultan in **zwei getrennten Werkzeugsystemen (WZS)** ab, wobei im Teileprogramm nur ein WZS programmiert ist (WZS1).

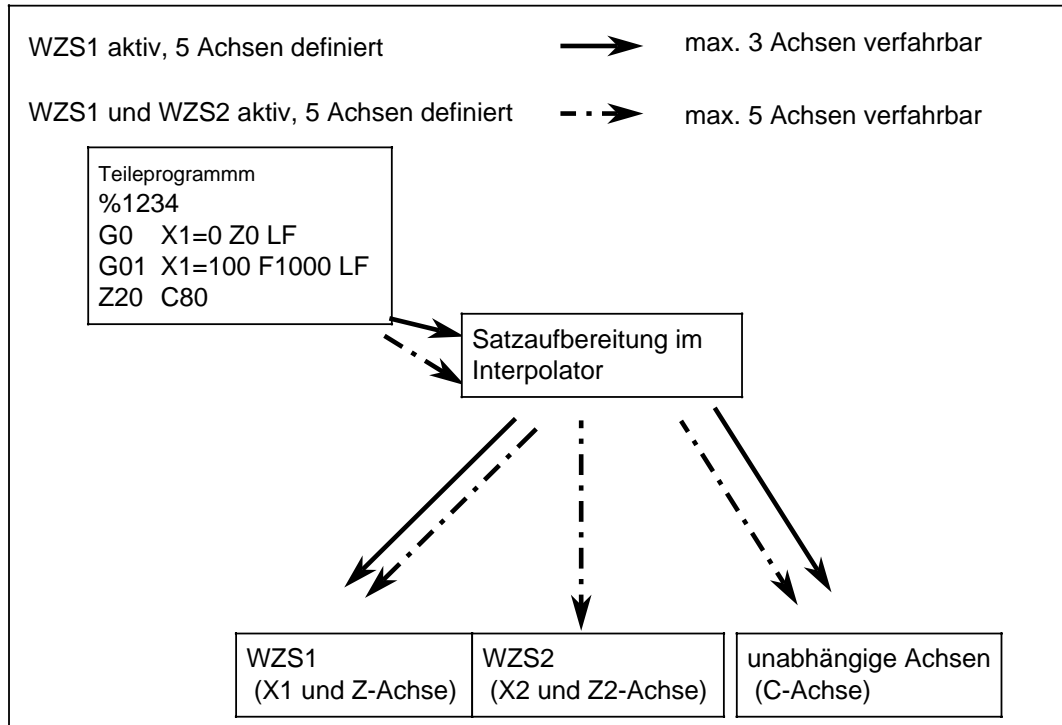
Bei einer Drehmaschine werden dadurch die **führenden (programmierten) Achsen**, z.B. die X1-Achse und die Z-Achse, **im WZS1** verfahren, die **verdoppelten Achsen**, z.B. die X2-Achse und die Z2-Achse, **im WZS2** verfahren.

Die Achsen des WZS2 dürfen im Teileprogramm **nicht** programmiert werden. Die Namen und Verfahrswege für die Achsen des WZS2 werden NC-intern vom Programm des WZS1 abgeleitet. Dies gilt auch für die Ebenenanwahl und die Geometriezuordnung der Werkzeuge.

Wie die Werkzeugsysteme bei "Achsverdopplung" angewählt werden, entnehmen Sie bitte den Hinweisen des Werkzeugmaschinen-Herstellers.

Nach "NC-Start" wird die über die PLC angewählte Funktion (WZS1 oder WZS2 bzw. beide WZS aktiv) aktiviert und abgespeichert. Die angewählte Funktion ist selbthaltend bis zum nächsten Programmende bzw. RESET und kann während der Programmbearbeitung nicht geändert werden (Alarmmeldung).



Teileprogramm ohne oder mit Achsverdopplung:

Wirkungsweise bei aktivem WZS1:

TeileprogrammN0010 G0 X1=0 Z D5 L_FN0020 G01 X1=100 F2000 L_FN0030 Z30 C50 L_FN0040 G33 X1 = 50 I2 L_F

N0050...

Wirkung

Achsen X1 und Z fahren mit Eilgang auf 0, angewähltes Werkzeug ist D5.

Achse X1 fährt mit Geschwindigkeit 2000 mm/min auf die Position 100 mm.

Achsen Z und C (Rundachse) interpolieren mit der Geschwindigkeit 2000 mm/min auf die Position 30 mm bzw. 50 Grad.

Gewindschneiden mit Achse X1.

Wirkungsweise bei aktivem WZS1 und WZS2 (Achsverdopplung):

TeileprogrammN0010 G0 X1=0 Z D5 L_FN0020 G01 X1=100 F2000 L_FN0030 Z30 C50 L_FN0040 G33 X1= 50 I2 L_F

N0050...

Wirkung

Achsen X1/X2 und Z/Z2 fahren mit Eilgang auf 0, angewähltes Werkzeug ist D5 für WZS1 und D55 für WZS2.

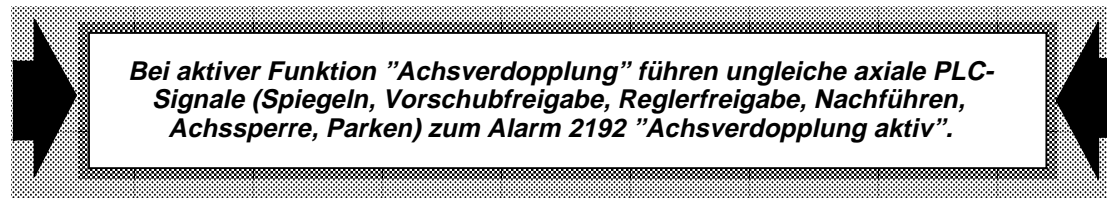
Achse X1/X2 fahren mit Geschwindigkeit 2000 mm/min auf die Position 100 mm.

Achsen Z/Z2 und C (Rundachse) interpolieren mit der Geschwindigkeit 2000 mm/min auf die Position 30 mm bzw. 50 Grad.

Gewindschneiden mit Achse X1/X2.

Besonderheiten der Funktion "Achsverdopplung":

- Die programmierte Sollgeschwindigkeit gilt für beide WZS.
- Handräder und Softwareendschalter wirken axial.
- Die Arbeitsfeldbegrenzungen G25/G26 werden verdoppelt (WZS1 und WZS2).
- Die programmierbaren Nullpunktverschiebungen G58,G59 gelten ebenfalls auch für die verdoppelten Achsen.
Im Bild "Programmierte Nullpunktverschiebung" wird der Wert der Nullpunktverschiebung immer für **beide** Achsen (programmierte und verdoppelte Achse) angezeigt.
- Mit der Funktion "Zweiter Referenzpunkt" und "Nullpunktverschiebung spiegeln" ist auch eine Komplettbearbeitung eines Werkstückes möglich (siehe nächste Seite).
- Ein Maßstabsfaktor im WZS1 wirkt auch im WZS2.

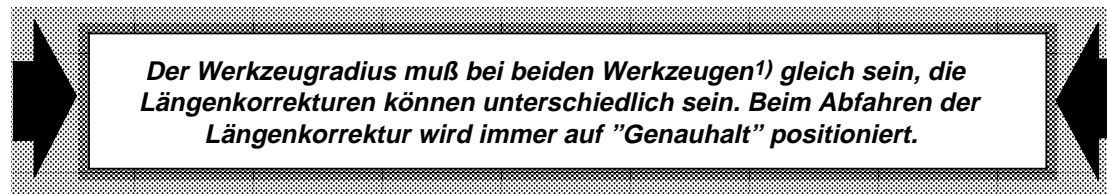


Werkzeugverwaltung, -korrektur:

Bei Betrieb in der Funktion "Achsverdopplung" wird der Werkzeugspeicher hälftig aufgeteilt, d.h. für das WZS1 gelten die Werkzeugnummern D1 bis D49 und für das WZS2 gelten die Nummern D51 bis D99. Der Offset bzw. die Verschiebung zwischen WZS1 und WZS2 beträgt damit 50.

Beispiel:

Für das WZS1 wird z.B. D3 programmiert. Die Steuerung wählt für WZS2 automatisch D53 an, wenn im Programm der Befehl "D3" bearbeitet wird.



Hinweis:

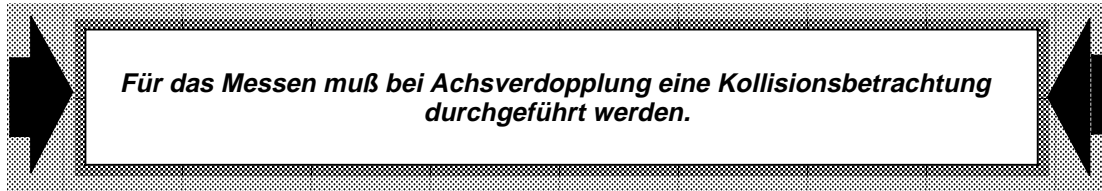
Bei aktiver Achsverdopplung und MD 5011 Bit 3= 1 bzw. Bit 4= 1 werden beim Werkzeugtyp P1=0 die Länge P2 und die Längenkorrektur P5 im Durchmesser gerechnet.

1) Werkzeugradius von WZS1 gilt für das WZS1 und das WZS2.

@-Befehle:

Außer dem @440 (programmierte Achsposition) wirken alle achsbezogenen @-Befehle wie bisher.

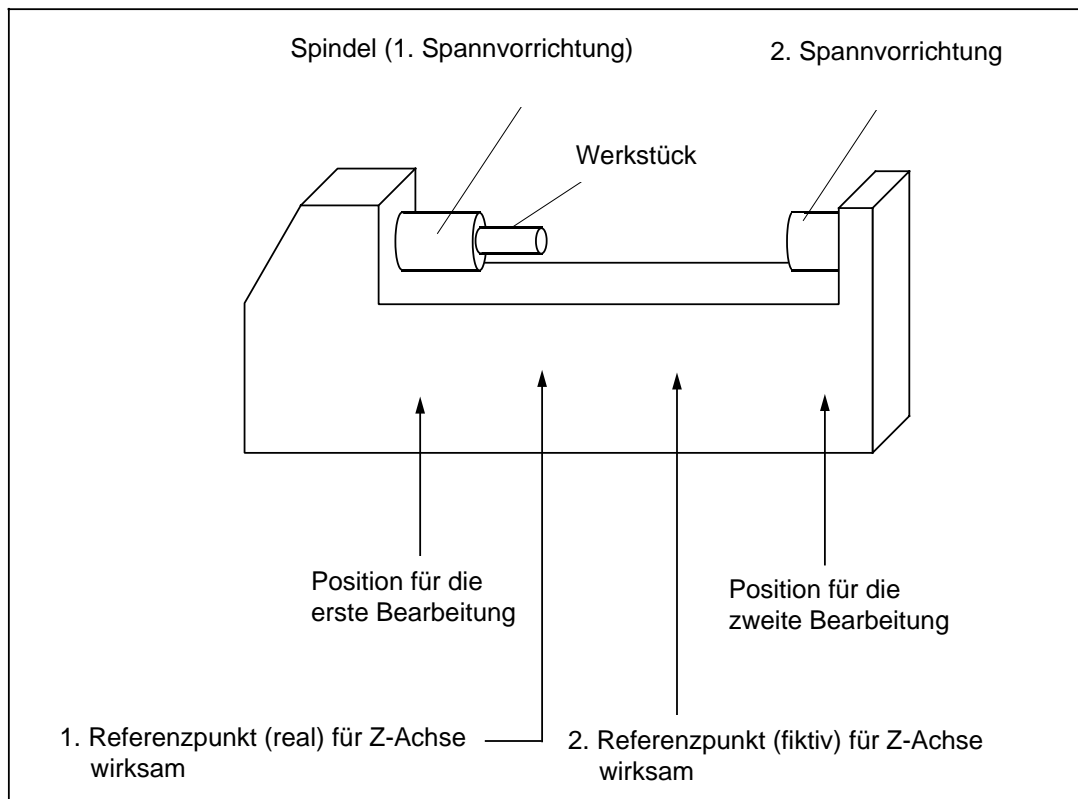
Bei @720 (fliegendes Messen) wird der programmierte Verfahrensweg auf die entsprechende Achse verdoppelt, aber nur ein Istwert gemessen. Beide Achsen stoppen nach dem Auslösen des Meßtasters.

**Settingdaten (SD):**

Als Maßstabsfaktoren wirken die SD-Optionsbits des WZS1 für das WZS1 und das WZS2. Bezüglich der Nullpunktverschiebungen gelten die unter "Besonderheiten" erläuterten Hinweise.

3.1.2.2 Komplettbearbeitung

Zur Komplettbearbeitung eines Werkstückes sind neben den Maschinendaten zum Spiegeln der Nullpunktverschiebungen Maschinendaten für einen "fiktiven, zweiten Referenzpunkt" vorgesehen. Die Differenz der Maschinendaten zwischen "Referenzpunktwert" und "fiktivem, zweiten Referenzpunktwert" wird dazu von der PLC als externe Nullpunktverschiebung vorgegeben. Der folgende Teil des NC-Programms muß sich dann auf diesen zweiten Referenzpunkt beziehen.



3.2 Achsbewegungen mit Bearbeitung

Je nach Art der Achsbewegung führt die **Steuerung** eine Geradeninterpolation oder eine Kreisinterpolation aus:

- **Geradeninterpolation:**
Linearbewegung
 - achsparallel
 - in zwei Achsen
 - in drei Achsen.
- **Kreisinterpolation:**
Kreisbewegung in 2 Achsen (Ebene).

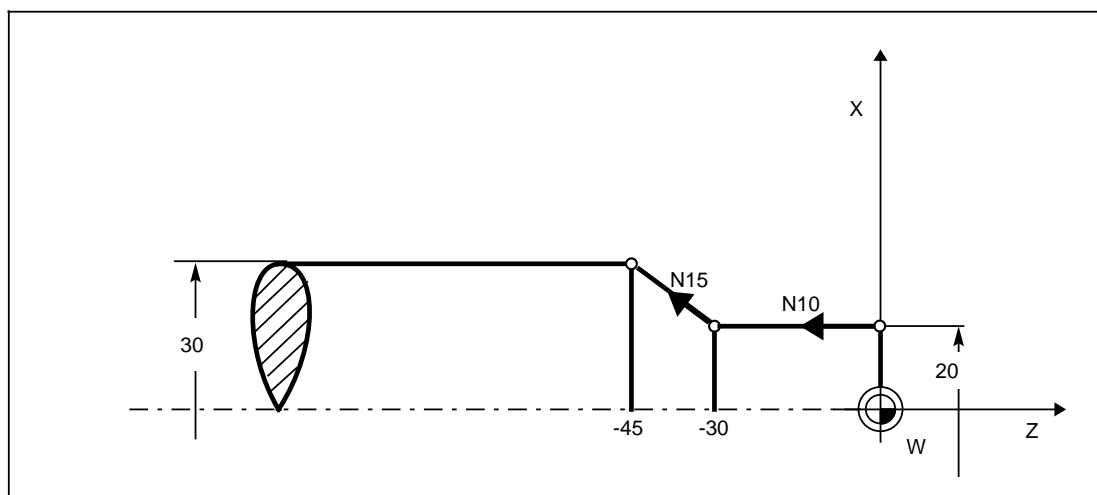
3.2.1 Geradeninterpolation G01

Das Werkzeug soll mit einem bestimmten Vorschub auf einer Geraden zum Zielpunkt fahren und dabei das Werkstück bearbeiten. Zur Berechnung des Werkzeugwegs führt die Steuerung eine Geradeninterpolation aus.

Die **Geradeninterpolation** bewirkt die Bewegung

- in einer Achsrichtung (Linearachse oder Rundachse),
- vom Ausgangspunkt auf den im Bezugs- oder Kettenmaß programmierten Zielpunkt,
- mit dem programmierten Vorschub und
- mit der programmierten Spindeldrehzahl.

Es können achsparallele und unter beliebigem Winkel verlaufende Bewegungen ausgeführt werden.



Geradeninterpolation G01

Bezugsmaßeingabe:

```
N5 ...  
N10 G01 G90 Z-30 F10 L_F  
N15 X5 Z-15 L_F
```

Kettenmaßeingabe:

```
N5 ...  
N10 G01 G91 Z-30 F10 L_F  
N15 X30 Z-45 L_F
```

3.2.2 Kreisinterpolation G02/G03

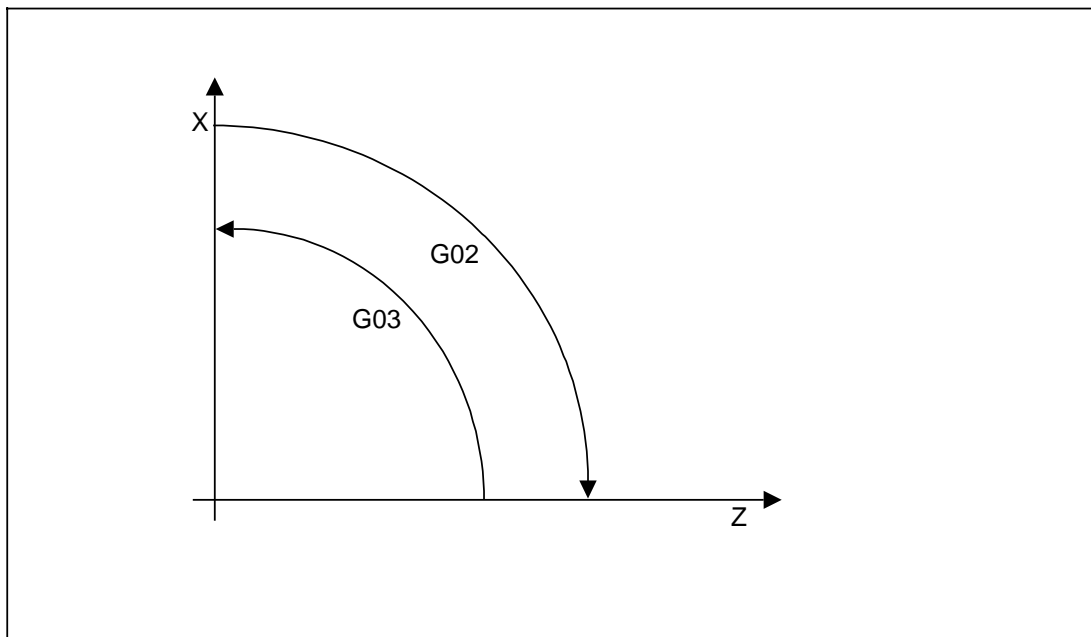
Das Werkzeug soll zwischen zwei Punkten der Kontur auf einem Kreis fahren und dabei das Werkstück bearbeiten.

Zur Berechnung des Werkzeugweges führt die Steuerung eine Kreisinterpolation aus.

Die **Kreisinterpolation** bewirkt die Bewegung des Werkzeugs

- auf einem Kreisbogen
mit **G02** im **Uhrzeigersinn**,
mit **G03** **gegen** den **Uhrzeigersinn**,
- um den programmierten Mittelpunkt des Kreises,
- vom Ausgangspunkt auf einer Kreisbahn zum programmierten Endpunkt.

Die Wegbedingungen **G02** und **G03** sind **modal (speichernd)** wirksam.



Kreisinterpolation

Die Interpolationsparameter bestimmen zusammen mit den Achsbefehlen den Kreis oder den Kreisbogen:

- Der Anfangspunkt "**KA**" des Kreises bzw. Kreisbogens wird durch den vorhergehenden Satz bestimmt.
- Den Endpunkt "**KE**" legen die Achsenwerte X und Z fest.
- Der Kreismittelpunkt "**KM**" wird festgelegt
 - entweder durch die Interpolationsparameter oder
 - direkt über den Radius (B).

3.2.2.1 Interpolationsparameter I, K

Die Interpolationsparameter sind die achsparallelen Koordinaten des Abstandsvektors vom Anfangspunkt zum Mittelpunkt des Kreises.

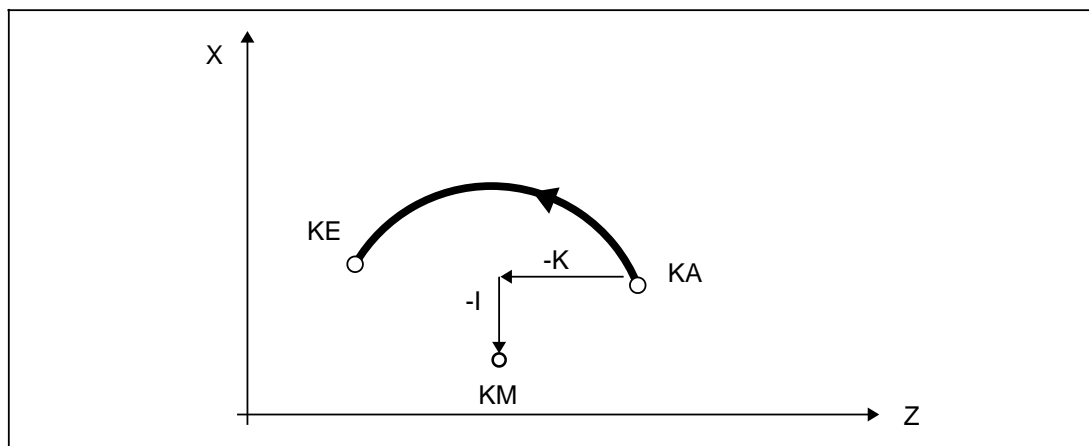
Nach DIN 66 025 sind den Achsen **X** und **Z** die Interpolationsparameter **I** und **K** zugeordnet. Abhängig davon, ob X und Z im Bezugsmaß oder im Kettenmaß programmiert sind, müssen die Interpolationsparameter **im Kettenmaß oder Bezugsmaß (über MD anwählbar)** und in der **richtigen Reihenfolge** eingegeben werden. Die Auswahl erfolgt per Maschinendatum durch den Maschinenhersteller.

Das Vorzeichen ergibt sich aus der Koordinatenrichtung vom Anfangspunkt zum Kreismittelpunkt.

Nicht programmiert wird:

- Ein Interpolationsparameter der den Wert 0 hat und
- die Endpunktkoordinaten, die sich gegenüber dem Kreisanzfang nicht geändert haben.

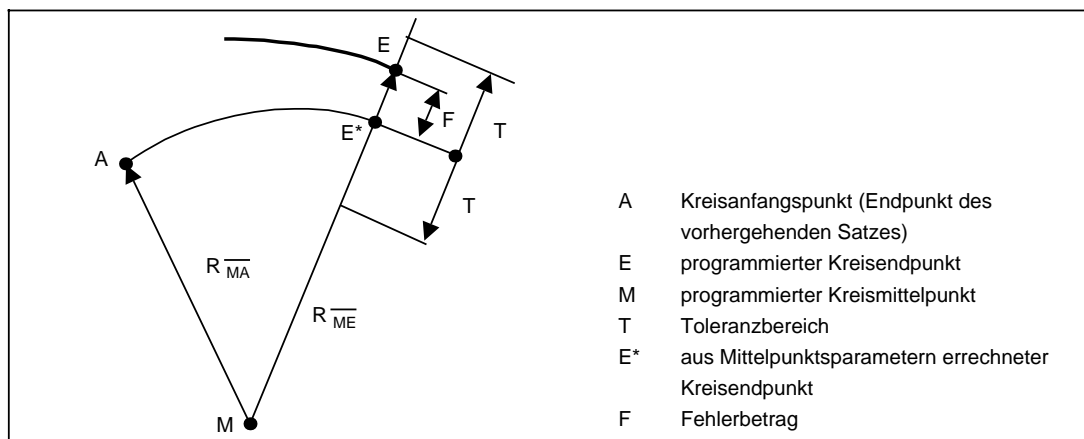
Beim Vollkreis muß mindestens eine Achse programmiert werden (X0 oder Z0).



Kreisinterpolation mit Interpolationsparametern

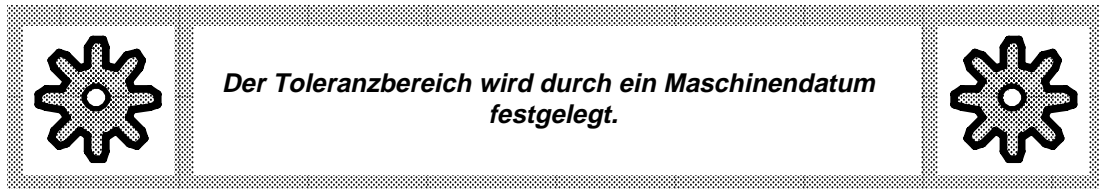
Kreisendpunktüberwachung:

Vor der Bearbeitung eines Kreissatzes prüft die NC-Steuerung die Übereinstimmung der programmierten Werte, indem die **Differenz der Radien** für den Anfangspunkt **A** und den Endpunkt **E** ermittelt wird.



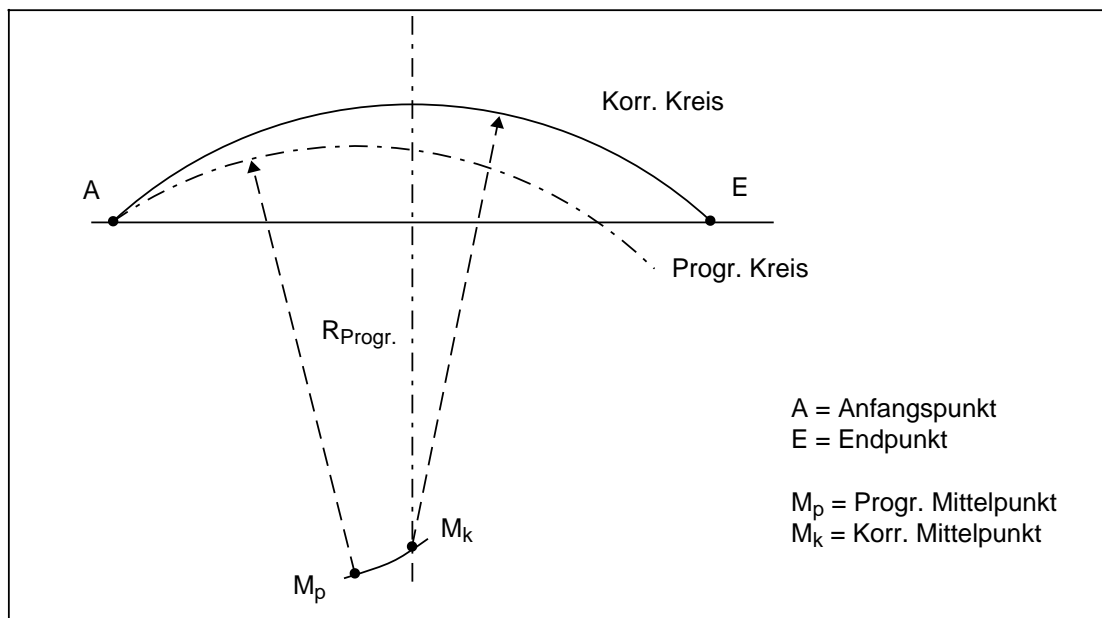
Definition des Toleranzbereiches der Kreisendpunktüberwachung

Bei ungenauer Eingabe des Radius oder der Interpolationsparameters überschreitet die Differenz (F) den Toleranzbereich (T) und der Kreissatz wird nicht bearbeitet. Die Steuerung meldet den Alarm 2048 "Kreisendpunktfehler".



Liegt die Abweichung der Radien innerhalb des Toleranzbereichs, so werden die Mittelpunktsparameter korrigiert, da davon ausgegangen wird, daß der Kreisendpunkt "richtig" programmiert wurde.

Danach wird der Kreissatz mit dem neuen, **korrigierten Mittelpunkt** verfahren.



Prinzip der Kreiskorrektur

3.2.2.2 Radiusprogrammierung

In vielen Fällen ist die Vermaßung einer Zeichnung so gewählt, daß es günstiger ist, für die Festlegung der Kreisbahn den Radius B anzugeben. Die erweiterte Adreßschreibweise für den Radius ist dabei ausgeschlossen. Für die Radiusprogrammierung werden der Endpunkt und der Radius B programmiert.

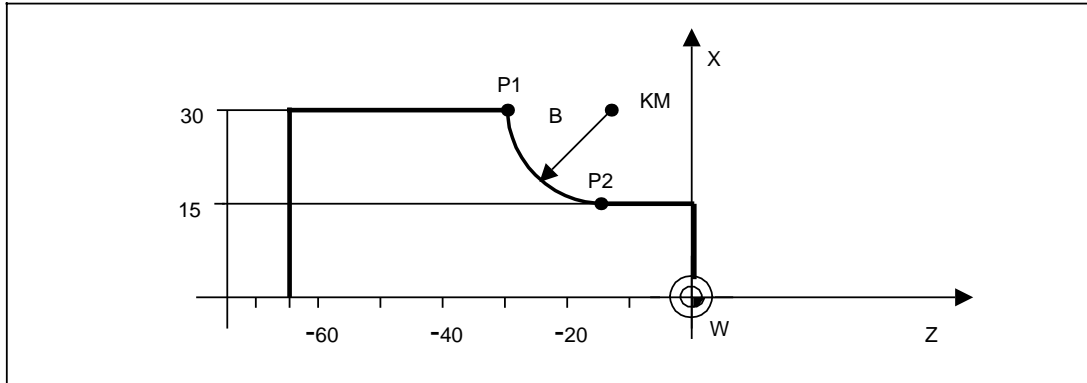
G02 oder G03 bestimmt die Bewegungsrichtung auf dem durch Kreisendpunkt und Interpolationsparameter bzw. Radius B bestimmten Kreis. Da die Radiusangabe zusammen mit G02 oder G03 nur innerhalb eines Halbkreises eine eindeutige Kreisbahn ergibt, muß noch angegeben werden, ob der Verfahrenswinkel kleiner oder größer als 180 Grad sein soll.

Die Radiusangabe erhält demnach folgendes Vorzeichen:

- + B : Verfahrenswinkel kleiner oder gleich 180 Grad.
- B : Verfahrenswinkel größer 180 Grad.

Bei einem Verfahrwinkel von 0 Grad oder 360 Grad ist die Radiusprogrammierung nicht zulässig. Vollkreise müssen mit Interpolationsparametern programmiert werden.

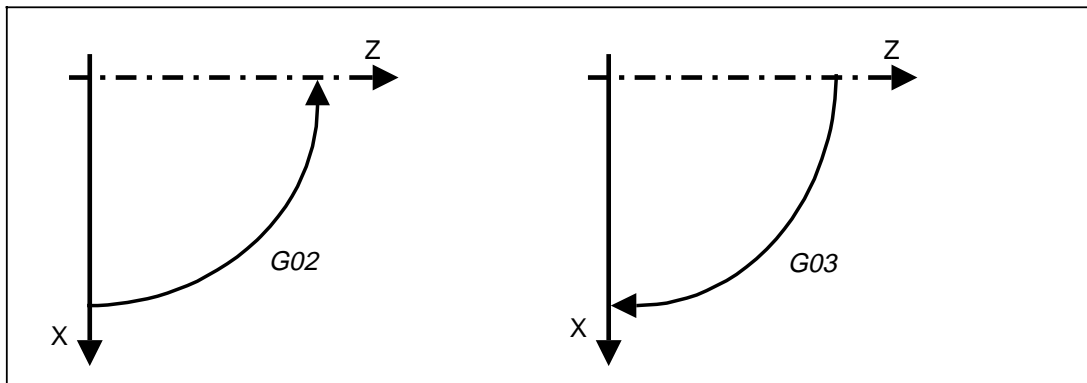
Beispiel: Radiusprogrammierung



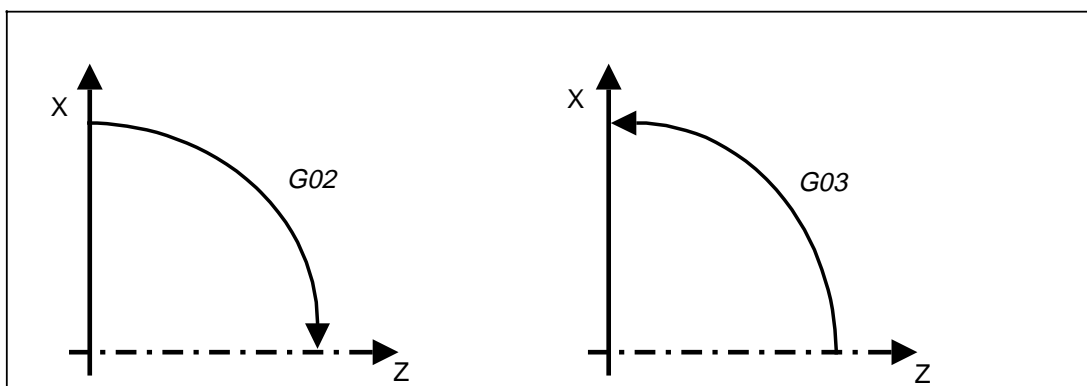
```
N5 G03 G90 X15 Z-15 B15 F500 L_F
N10 G02 G90 X30 Z-30 B15 L_F
```

Werkzeug bearbeitet von P1 nach P2
 Werkzeug bearbeitet von P2 nach P1

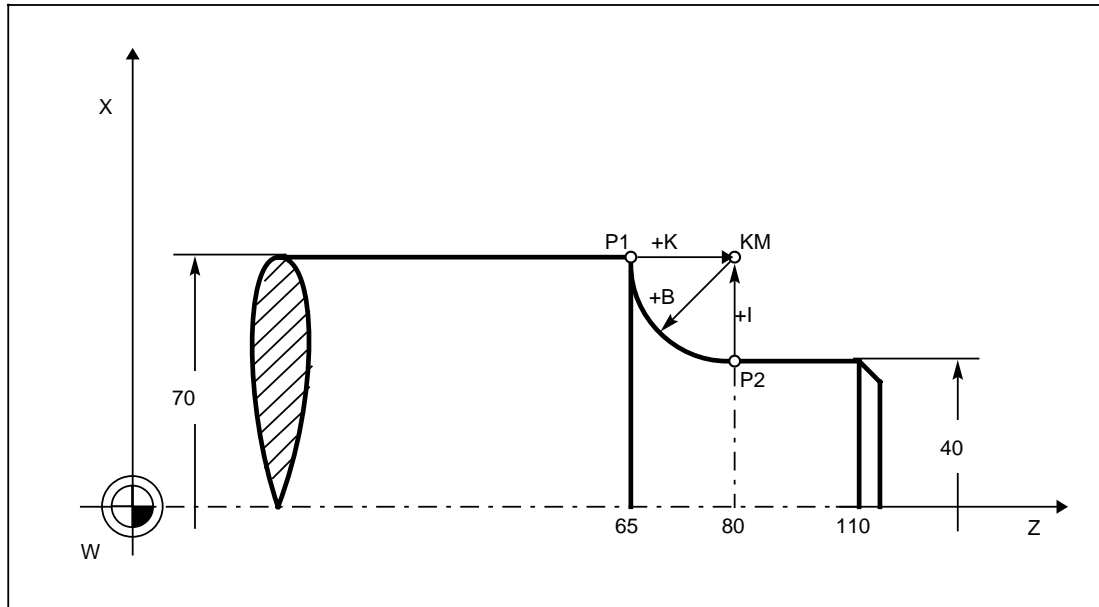
Wirkungsweise von G02 bzw. G03 vor der Drehmitte und hinter der Drehmitte:



Vor der Drehmitte



Hinter der Drehmitte

Programmierbeispiel: Drehmaschine*Kreisinterpolation*

Interpolationsparameter:

N5 G03 G90 X40 Z80 K15 I0 F500 L_F
 N10 G02 X70 Z65 K0 I15 L_F

Werkzeug bearbeitet von P1 nach P2
 Werkzeug bearbeitet von P2 nach P1

oder Radiusprogrammierung:

N5 G03 G90 X40 Z80 B+15 F500 L_F
 N10 G02 X70 Z65 B+15 L_F

Werkzeug bearbeitet von P1 nach P2
 Werkzeug bearbeitet von P2 nach P1

3.2.3 Zylinderinterpolation

Die Funktion "Zylinderinterpolation" ist eine Bestelldaten-Ergänzung.

Die Zylinderinterpolation ermöglicht die Bearbeitung von Zylinderbahnen mit einer **rotatorischen** und einer **linearen** Achse auf konstantem Rundtischdurchmesser. Es können sowohl Geraden als auch Kreiskonturen programmiert werden, Kreise jedoch nur mit Kreis-Radius-Programmierung (siehe Kap. 3.2.2.2). Die Eingabe von den Interpolationsparametern I,J undK ist nicht erlaubt. SPLINE-Interpolation (siehe Kap. 3.2.11) ist ebenfalls nicht möglich. Die Position der Rundachse wird in Gradmaßen eingegeben. Die Umrechnung auf die Umfangsmaße des Arbeitsdurchmessers erfolgt steuerungsintern. Hierzu wird das Verhältnis P unter G92 P .. programmiert.

Steuerungsintern wird der Bearbeitungsdurchmesser mit dem Einheitsdurchmesser ins Verhältnis gesetzt:

$$P = \frac{\text{Bearbeitungsdurchmesser}}{\text{Einheitsdurchmesser}}$$

Eingabesystem:

Der Einheitsdurchmesser beträgt

- 114,592 mm im metrischen System bei Eingabefinheit 10^{-3} mm
- 114,592 inch bei Eingabefinheit 10^{-3} inch.

Der Einheitsdurchmesser leitet sich aus der Beziehung $\pi \cdot d = 360$ ab.

$$\text{Einheitsdurchmesser} = \frac{360}{\pi} \text{ in mm bzw. inch.}$$

In einem Satz mit G92 P.. dürfen außer dem Achsnamen keine weiteren Zeichen geschrieben werden.

N.. G92 P.. C LF

P.. Faktor für Einheitskreis, **P = Arbeitsdurchmesser/Einheitsdurchmesser**

C Achsname für Rundachse.

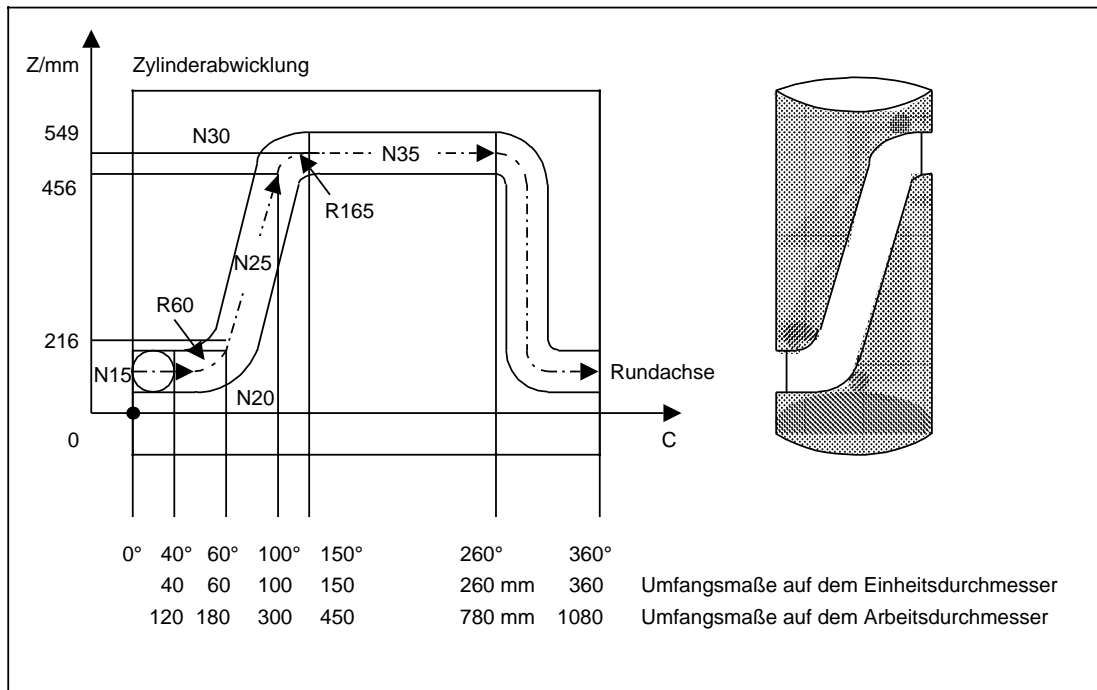
Die Eingabefinheit für P beträgt 10^{-5} und der Wert für P ist < 100 . Der Bearbeitungsdurchmesser ist modal wirksam bis eine erneute Programmierung oder Rücksetzen mit M02/M30 bzw. RESET erfolgt. Die programmierte Vorschubgeschwindigkeit wird auf der Mittelpunktbahn eingehalten. Solange der Bearbeitungsdurchmesser ungleich dem Einheitsdurchmesser ist (d.h. P ungleich 1 ist), kann diese Achse (z.B. C) nur mit einer weiteren Achse interpolieren. Für Interpolation mit mehr als zwei Achsen muß der Bearbeitungsdurchmesser gleich dem Einheitsdurchmesser sein (Faktor P =1).

Für $P < 1$ ist zu beachten, daß die Interpolationsfeinheit der Rundachse bei aktiver Zylinderinterpolation $1/p$ mal so groß ist, wie ohne Zylinderinterpolation.

Bei angewählter Zylinderinterpolation darf die Rundachse nicht mit G68 programmiert werden (nur G90,G91).

Die Zylinderinterpolation können Sie zusammen mit der FRK/SRK einsetzen. Die An-/Abwahl der Zylinderinterpolation ist nur bei abgewählter FRK/SRK möglich. Die Funktion "Weiches An-/Abfahren" dürfen Sie nicht zusammen mit der Zylinderinterpolation anwenden.

Beispiel: Zylinderinterpolation



Zylinderinterpolation

```

%10 L_F
N10 G92 P3 C L_F           Anwahl der Zylinderinterpolation
N15 G01 C40 Z.. F1000 S800 M3 L_F
N20 G03 C60 Z216 B+60 L_F   Radius R60 (B60)
N25 G01 C100 Z456 L_F      Radius R165 (B165)
N30 G02 C150 Z549 B+165 L_F
N35 G01 C260 L_F
:
N60 G92 P1 C L_F           Abwahl der Zylinderinterpolation
N70 M02 L_F

```

3.2.4 Vorschub F/G94/G95/G96/G97/G98

Der **Vorschub F** wird in mm/min (m/min) oder in mm/U programmiert:

G94 F.. Vorschub in mm/min

G95 F.. Vorschub in mm/U

G96 F.. Vorschub in mm/U

G96 F..S.. Konstante Schnittgeschwindigkeit (S in m/min) bei vorgegebenem Vorschub F in (mm/U). (Gilt nur für Leitspindel)

G98 F.. Vorschub in U/min (nur für Rundachsen)

Der Vorschub bestimmt die Bearbeitungsgeschwindigkeit (Bahngeschwindigkeit) und wird bei jeder Interpolationsart auch unter Berücksichtigung von Werkzeugkorrekturen auf der Kontur eingehalten. Der unter der Adresse F programmierte Wert bleibt in einem Programm erhalten bis ein neuer F-Wert programmiert wird. Mit Programmende oder Reset wird der F-Wert gelöscht. Deshalb muß im ersten Programmsatz ein F-Wert eingegeben werden.

Bei Änderungen der G-Funktionen G94, G95 oder G96 ist spätestens im folgenden Satz der F-Wert neu zu programmieren.

Der programmierte Vorschub F kann über einen Vorschubkorrektorschalter an der Maschinensteuertafel von 1% bis 120% verändert werden. Die 100%-Stellung entspricht dem programmierten Wert.

Zu jedem G98-aktiven Verfahrssatz muß genau eine Rundachse mit einem Verfahrweg > 0 geschrieben werden.



Konstante Schnittgeschwindigkeit G96 S..

Mit G96 kann eine konstante Schnittgeschwindigkeit unter der Adresse S eingegeben werden. Gleichzeitig kann auch ein Vorschub F programmiert werden.

G96 F.. S.. konstante Schnittgeschwindigkeit S in (m/min) bei vorgegebenem Vorschub F in (mm/U).

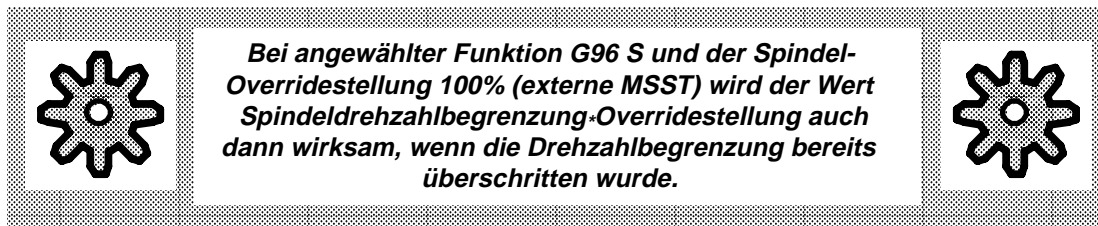
Beispiel: Drehmaschine

Abhängig von der programmierten Schnittgeschwindigkeit **ermittelt die Steuerung** die zum jeweils aktuellen Drehdurchmesser gehörende **Spindeldrehzahl**.

N5.. G96 F.. S..

Die Abstimmung von Drehdurchmesser, Spindeldrehzahl und Vorschubbewegung aufeinander, ermöglicht eine optimale Anpassung des Programms auf die Maschine, das zu bearbeitende Material und das verwendete Werkzeug.

Dabei muß der Nullpunkt der X-Achse in der Drehmitte liegen. Dies ist durch Referenzpunktverfahren sichergestellt.



Bei der Berechnung der Spindeldrehzahl für die konstante Schnittgeschwindigkeit werden folgende Werte berücksichtigt:

- Istwert
- Werkzeuglängenkorrektur
- Nullpunktverschiebung in X-Richtung
 - einstellbare NV G54, G55, G56, G57
 - programmierbare NV G58, G59
 - externe Zusatzkompensation (PRESET, DRF).

Das Werkstück darf durch eine Nullpunktverschiebung in der X-Achse nicht aus der Drehmitte verschoben werden. Die Nullpunktverschiebung in der X-Achse kann zur Verschiebung des Werkzeugträgers verwendet werden.

Bei der Ermittlung der Spindeldrehzahl wird immer auf den Werkstücknullpunkt Bezug genommen.

Bei konstanter Schnittgeschwindigkeit wird in einer Getriebestufe gearbeitet. Ein Wechsel der Getriebestufe ist nicht zulässig. Die entsprechende Getriebestufe ist vorher anzuwählen.

Spindeldrehzahlbegrenzung G92

Arbeiten Sie mit konst. Schnittgeschwindigkeit (G96), kann die Spindeldrehzahl S (U/min) bei kleinen Radialwerten unzulässig anwachsen. Dies können Sie vermeiden, indem Sie die Spindeldrehzahl S mit dem Befehl G92 S.. begrenzen. Die Spindeldrehzahlbegrenzung ist nur wirksam, wenn Sie zuvor G96 angewählt haben. Die Funktion G92 ist selbsthaltend und muß einzeln in einem Satz stehen.

Beispiel:

G92 S500 bedeutet, daß während einer konst. Schnittgeschwindigkeit (G96 S..) die Spindeldrehzahl auf 500 U/min begrenzt bleibt.

Werden in einem nachfolgenden Satz höhere Werte für die Spindeldrehzahl programmiert, so läuft die Spindel trotzdem nur mit der Grenzdrehzahl. Eine Spindeldrehzahlbegrenzung G92 können Sie auch direkt über die Maske "SETTINGDATEN SPINDEL" festlegen.

Konstante Drehzahl G97

Mit G97 wird die konstante Schnittgeschwindigkeit aufgehoben. Die zuletzt erreichte Drehzahl wird als konstante Drehzahl beibehalten.

Damit wird eine unerwünschte Drehzahländerung bei Bewegungen in X-Richtung ohne Bearbeitung vermieden.

G97 Abwahl der konstanten Schnittgeschwindigkeit und Abspeichern des letzten Drehzahl-sollwertes von G96.

Vorschubuntersetzung M37

Der programmierte Vorschub kann mit M37 um 1:100 untersetzt werden. Mit **M36** (Grundstellung) wird **M37** aufgehoben.

3.2.5 Gewindeschneiden G33/G34/G35

Die Gewinde können je nach Ausführung unterschieden werden in

- Gewinde mit konstanter Steigung,
- Gewinde mit veränderlicher Steigung,
- eingängige oder mehrgängige Gewinde,
- Gewinde auf zylindrischen oder kegeligen Körpern,
- Außen- oder Innengewinde,
- Plangewinde.

Für die **Bearbeitung von Gewinden** stehen folgende Wegbedingungen zur Verfügung:

G33 Gewindeschneiden mit konstanter Steigung.

G34 Gewindeschneiden mit linear zunehmender Steigung.

G35 Gewindeschneiden mit linear abnehmender Steigung.

Die G-Funktionen G33, G34 und G35 sind der Leitspindel zugeordnet.

Die **Gewindelänge** wird unter der entsprechenden Wegadresse eingegeben, wobei An- und Auslaufstrecken zu berücksichtigen sind, auf denen die Vorschubgeschwindigkeit hochgefahren bzw. reduziert wird.

Die Werte können im Bezugs- oder im Kettenmaß eingegeben werden.

Die **Gewindesteigung** wird unter den Adressen I und K eingegeben. Bei Längsgewinden wird die Steigung unter der Adresse K, bei Plangewinden unter der Adresse I und bei Kegelgewinden unter I und K eingegeben. I und K sind stets im Kettenmaß ohne Vorzeichen einzugeben.

Die Standard-Eingabefeinheit der Gewindesteigung beträgt 0.001 mm/Umdrehung.

Die Gewindesteigung ist von 0.001 mm bis 400.000 bzw. 2000.000 mm programmierbar.

Ist als Eingabefeinheit der Gewindesteigung 1 mm programmiert, so kann mit M37 als Feinheit 0,01 mm/Umdrehung erreicht werden.

Rechts- oder Linksgewinde werden durch Angabe der Spindeldrehrichtung **M03 bzw. M04** programmiert.

Die Spindeldrehrichtung und die Drehzahl sind in dem Satz vor der eigentlichen Gewindeschneideoperation zu programmieren, damit die Spindel auf ihre Soll Drehzahl hochlaufen kann.

Beispiel:

N10 S500 M03 L_F

Spindeldrehzahl S=500 U/min, Rechtslauf

N15 G33 Z... K... L_F

Gewindeschneiden mit konstanter Steigung

Um Gewinde in mehreren Schnitten fertigen zu können, beginnt der Vorschubstart erst mit Hilfe einer Nullmarke am Pulsgeber. Damit ist gewährleistet, daß das Werkzeug immer an der gleichen Stelle des Werkstückumfangs in das Werkstück einfährt. Die Schnitte sollen mit gleicher Geschwindigkeit (Spindeldrehzahl) ausgeführt werden, um unterschiedlichen Schleppabstand zu vermeiden.

Der Vorschubkorrektorschalter, die Taste "VORSCHUB AUS", der Spindeldrehzahlkorrektorschalter und die Taste "EINZELSATZ" sind beim Gewindeschneiden unwirksam.

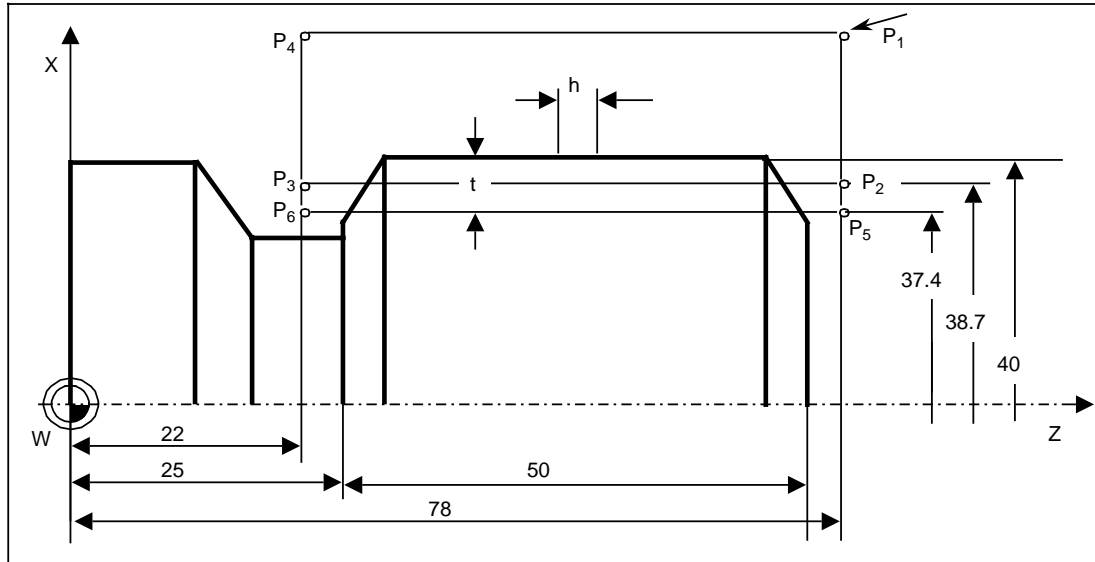
Der unter F programmierte Vorschub bleibt jedoch erhalten und wird bei der nächsten Programmierung von z.B. G01 wieder wirksam.

Die Steigung der Kegel, auf denen das Gewinde geschnitten wird, kann in Stufen geändert werden. Dadurch erreicht man bei Längsgewinden ein sanftes Auslaufen des Gewindes.

3.2.5.1 Gewinde mit konstanter Steigung

Die Vorschubprogrammierung F entfällt hier, da über einen Pulsgeber die Vorschubgeschwindigkeit direkt mit der Spindeldrehzahl verknüpft wird.

Beispiel: Gewinde auf einem zylindrischen Körper (Längsgewinde)



Steigung $h = 2 \text{ mm}$
 Gewindetiefe $t = 1,3 \text{ mm}$
 Radiale Zustellrichtung

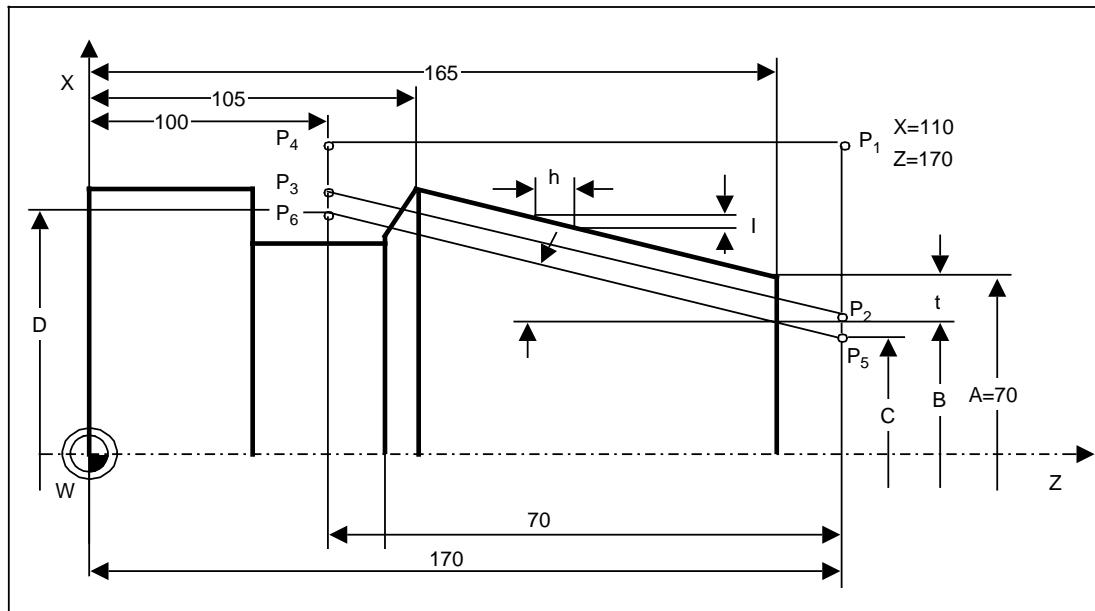
Bezugsmaßeingabe:

```
%387 (G33, G90 GEWINDESCHNEIDEN, LAENGS) L_F
N5 G90 S400 M3 L_F Spindeldrehzahl
N10 G00 X46 Z78 L_F (P1) Fahre im Eilgang auf P1
N15 X38.7 L_F (P2) Zustellung auf 1. Schnittiefe
N20 G33 Z22 K2 L_F (P3) Gewindeschneiden 1. Durchlauf von P2 auf P3
N25 G00 X46 L_F (P4) Ausfahren auf P4 im Eilgang
N30 Z78 L_F (P1) Fahre im Eilgang auf P1
N35 X37.4 L_F (P5) Zustellung auf 2. Schnittiefe
N40 G33 Z22 K2 L_F (P6) Gewindeschneiden 2. Durchlauf von P5 auf P6
N45 G00 X46 L_F (P4) Ausfahren auf P4 im Eilgang
N50 M30 L_F
```

Kettenmaßeingabe:

```
%388 (G33, G90 GEWINDESCHNEIDEN, LAENGS) L_F
N5 G91 S400 M3 L_F Spindeldrehzahl
N10 G00 G90 X46 Z78 L_F (P1) Fahre im Eilgang auf P1
N15 G91 X-3.65 L_F (P2) Zustellung auf 1. Schnittiefe
N20 G33 Z-56 K2 L_F (P3) Gewindeschneiden 1. Durchlauf von P2 auf P3
N25 G00 X3.65 L_F (P4) Ausfahren auf P4 im Eilgang
N30 Z56 L_F (P1) Fahre im Eilgang auf P1
N35 X-4.3 L_F (P5) Zustellung auf 2. Schnittiefe
N40 G33 Z-56 K2 L_F (P6) Gewindeschneiden 2. Durchlauf von P5 auf P6
N45 G00 X4.3 L_F (P4) Ausfahren auf P4 im Eilgang
N50 M30 L_F
```

Beispiel: Gewinde auf einem kegeligen Körper (Kegelgewinde)



Steigung $h = 5 \text{ mm}$
 Gewindetiefe $t = 1,73 \text{ mm}$; $\alpha = 15 \text{ Grad}$
 Radiale Zustellrichtung

Es müssen beide Endpunktkoordinaten geschrieben werden. Die Steigung h wird unter K eingegeben.

Berechnung der Gewindeanfangs- und Endpunktkoordinaten:

(A, B, C, usw. sind Durchmesserwerte)

1. Schnitt P2 . . . P3, $t = 1 \text{ mm}$
2. Schnitt P5 . . . P6, $t = 1.73 \text{ mm}$

$$\begin{aligned}
 A &= 70 \\
 B &= A - 2 \cdot t = 66.54 \\
 C &= B - 2 \cdot (5 \cdot \tan 15^\circ) = 63.86 \\
 D &= C + 2 \cdot (70 \cdot \tan 15^\circ) = 101.366 \\
 K &= h = 5 \\
 I &= h \cdot \tan 15^\circ = 1.34 \text{ mm}
 \end{aligned}$$

Berechnung der Punkte P2 und P3:

$$\begin{aligned}
 X(P2) &= C + 2 \text{ mm} = 65.86 \text{ mm} \\
 X(P3) &= D + 2 \text{ mm} = 103.366 \text{ mm}
 \end{aligned}$$

Bezugsmaßeingabe:

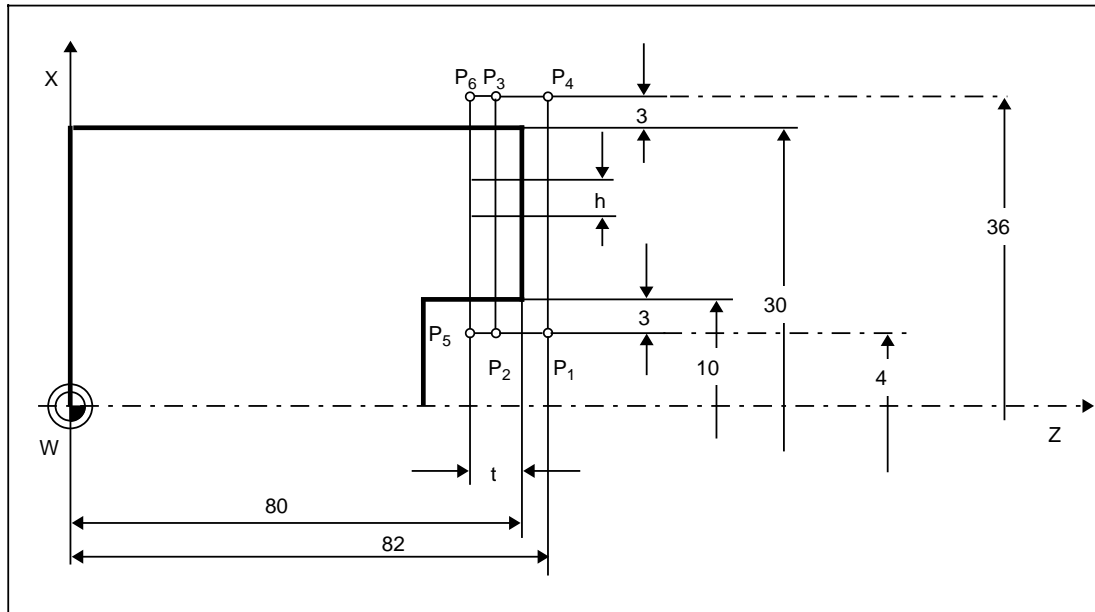
```

%389 (G33 GEWINDESCHNEIDEN, KEGEL) L_F
N30 G90 S200 M3 L_F
N35 G00 X110 Z170 L_F
N40 X65.86 L_F
N45 G33 X103.366 Z100 K5 I1.34 L_F
N50 G00 X110 L_F
N55 Z170 L_F
N60 X63.86 L_F
N65 G33 X101.366 Z100 K5 I1.34 L_F
N70 G00 X110 L_F
N75 M30 L_F
    
```

Bezugsmaßvorgabe
 (P1) Fahre im Eilgang auf P1
 (P2) Zustellung auf 1. Schnittiefe
 (P3) Gewindeschneiden 1. Durchlauf von P2 auf P3
 (P4) Ausfahren auf P4 im Eilgang
 (P1) Ausfahren auf P1 im Eilgang
 (P5) Zustellung auf 2. Schnittiefe
 (P6) Gewindeschneiden 2. Durchlauf von P5 auf P6
 (P4) Ausfahren auf P4 im Eilgang

Falls $> 45^\circ$ muß statt der Gewindesteigung in Z-Richtung die Gewindesteigung in X-Richtung (mit dem Parameter I) programmiert werden.

Beispiel: Plangewinde



Steigung $h = 2 \text{ mm}$
 Gewindetiefe $t = 1.3 \text{ mm}$
 Zustellung rechtwinklig zur Schnitttrichtung

Bezugsmaßeingabe:

<code>%390 (G33 GEWINDESCHNEIDEN, PLAN) L_F</code>		
<code>N40 G90 S100 M3 L_F</code>		Bezugsmaßvorgabe, Spindeldrehzahl
<code>N45 G00 X4 Z82 L_F</code>	(P1)	Fahre im Eilgang auf P1
<code>N50 Z79.35 L_F</code>	(P2)	Zustellung auf 1. Schnittiefe
<code>N55 G33 X36 I2 L_F</code>	(P3)	Gewindeschneiden 1. Durchlauf von P2 auf P3
<code>N60 G00 Z82 L_F</code>	(P4)	Fahre im Eilgang auf P4
<code>N65 X4 L_F</code>	(P1)	Fahre im Eilgang auf P1
<code>N70 Z78.7 L_F</code>	(P5)	Zustellung auf 2. Schnittiefe
<code>N75 G33 X36 I2 L_F</code>	(P6)	Gewindeschneiden 2. Durchlauf von P5 auf P6
<code>N80 G00 Z82 L_F</code>	(P4)	Ausfahren im Eilgang auf P4
<code>M85 M30 L_F</code>		

3.2.5.2 Gewinde mit veränderlicher Steigung

Die **Gewindesteigung pro Gang** wird um den unter **Adresse F** programmierten Wert verändert, bis der maximal bzw. minimal mögliche Wert erreicht ist.

G34 Steigung zunehmend:

N25 G34 G90 Z217 K2 F0.1 L_F

K2 Anfangssteigung 2 mm

F0.1 Steigungsänderung + 0,1 mm pro Gang,
d.h. nach 5 Gängen beträgt die Gewindesteigung 2,5 mm

G35 Steigung abnehmend:

N45 G35 G90 Z417 K10 F0.5 L_F

K10 Anfangssteigung 10 mm

F0.5 Steigungsänderung 0,5 mm pro Gang,
d.h. nach 10 Umdrehungen beträgt die Gewindesteigung 5 mm.

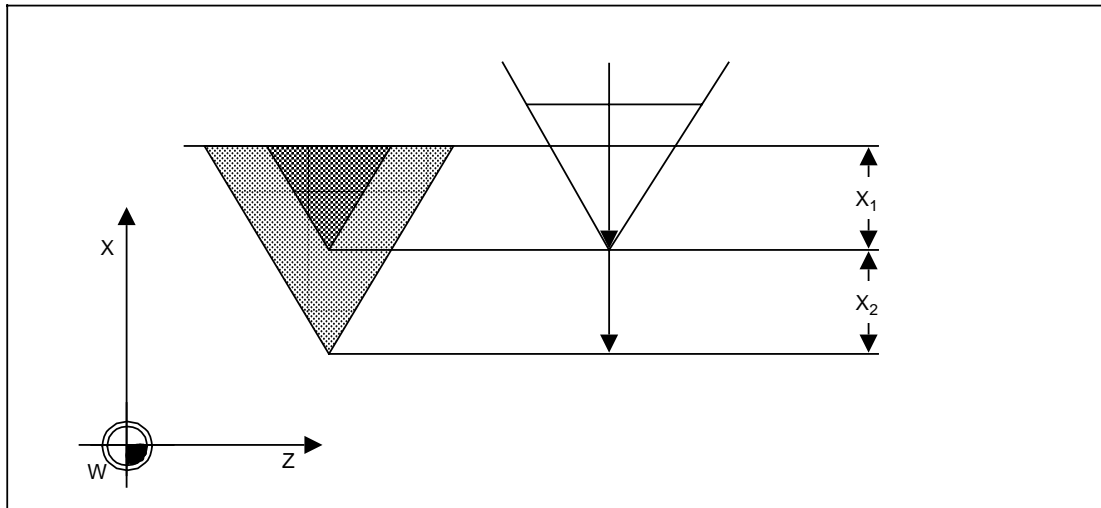
Der Wert **F errechnet** sich aus Anfangs- und Endsteigung:

$$F = \frac{\text{Anfangssteigung}^2 - \text{Endsteigung}^2}{2 \cdot \text{Gewindelänge}}$$

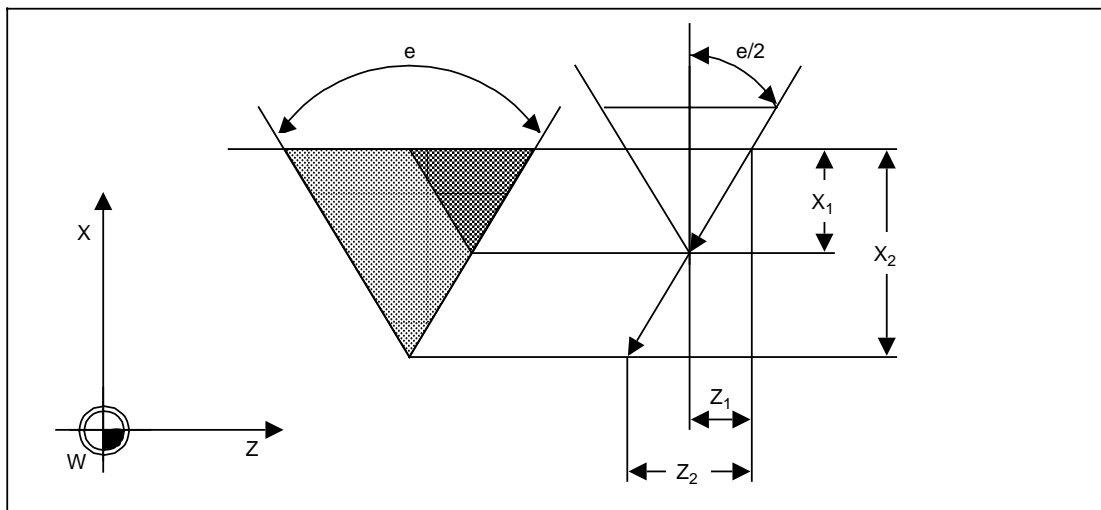
Der Wert ist ohne Vorzeichen einzusetzen.

3.2.5.3 Zustellmöglichkeiten

Es besteht die Möglichkeit, das Werkzeug rechtwinklig zur Schnittrichtung oder entlang der Flanke zuzustellen.

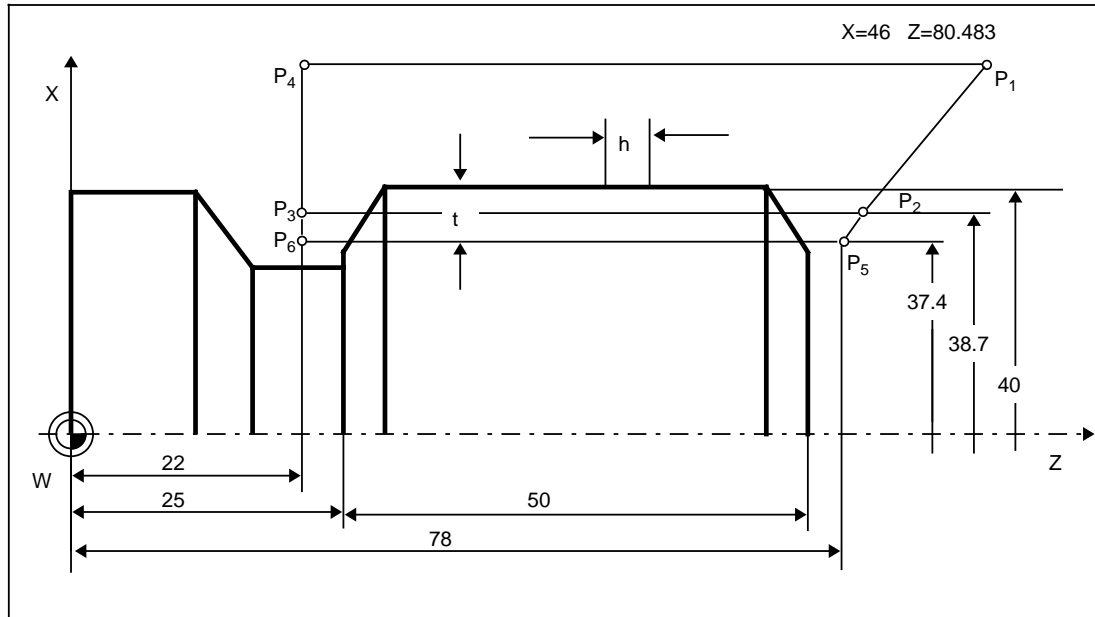


Zustellmöglichkeit "rechtwinklig zur Schnittrichtung"



"Flankenzustellung" $Z = X \cdot \tan e/2$ - mit einer Werkzeugseite wird geschnitten

Beispiel: Längsgewinde mit konstanter Steigung, Flankenzustellung



Steigung $h = 2 \text{ mm}$
 Gewindetiefe $t = 1.3 \text{ mm}$
 Zustellwinkel $e = 60 \text{ Grad}$

$$X_1 = (46 - 38.7) / 2 = 3.65$$

$$Z_1 = X_1 \cdot \tan e / 2 = 2.136$$

$$X_2 = 3.65 + 1,3 / 2 = 4.3$$

$$Z_2 = X_2 \cdot \tan e / 2 = 2.483$$

Bezugsmaßeingabe:

```
%391 (G33 GESCHWINDESCHNEIDEN) L_F
N05 G90 S50 M3 L_F
N10 G00 X46 Z80.483 L_F (P1)
N15 X38.7 Z78.347 L_F (P2)

N20 G33 Z22 K2 L_F (P3)
N25 G00 X46 L_F (P4)
N30 Z80.483 L_F (P1)
N40 X37.4 Z78 L_F (P5)

N45 G33 Z22 K2 L_F (P6)
N50 G00 X46 L_F (P4)
N55 M30 L_F
```

Bezugsmaßvorgabe, Spindeldrehzahl
 Fahre im Eilgang auf P1
 Zustellung auf 1. Schnitttiefe
 (Z-Wert = Gewindeschneiden Anfangspunkt)
 Gewindeschneiden 1. Durchlauf
 Fahre im Eilgang auf P4
 Fahre im Eilgang auf P1
 Zustellung auf 2. Schnitttiefe
 (Z-Wert = Z-Wert von P2 - (Z₂ - Z₁))
 Flankenzustellung
 Gewindeschneiden 2. Durchlauf
 Ausfahren im Eilgang auf P4

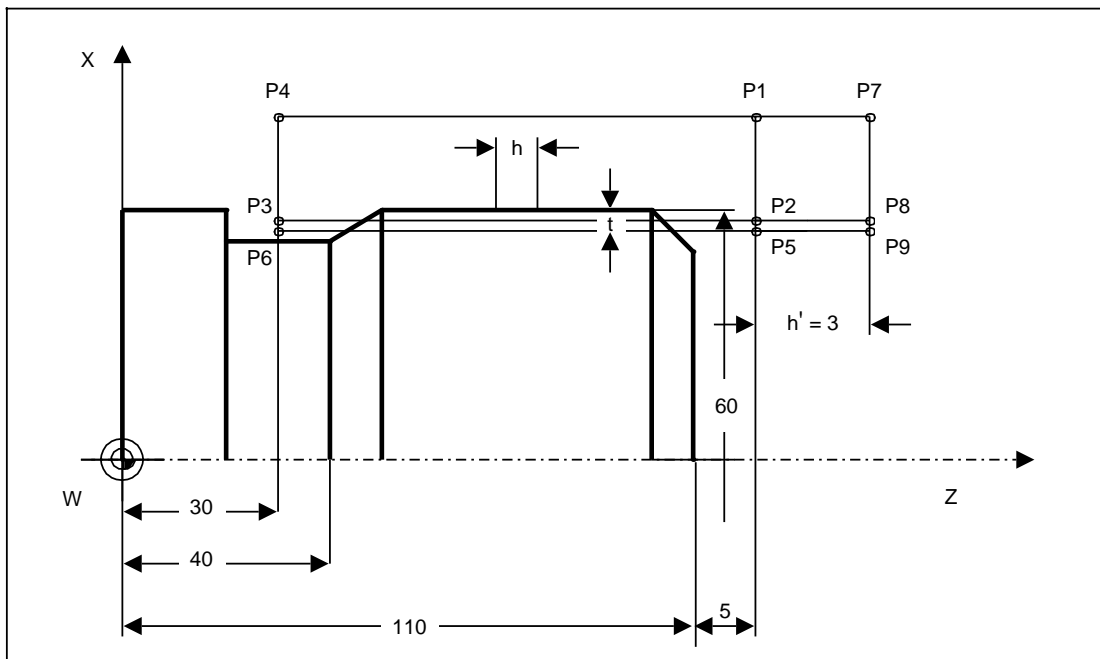
3.2.5.4 Mehrgängige Gewinde

Das Gewindeschneiden beginnt stets beim Synchronisierungspunkt der Nullmarke des Pulsgebers. Erst wenn dieses Signal vom digitalen Drehgeber kommt, wird der Vorschub freigegeben. Über die Programmierung kann man die Lage des Startpunktes für das Gewindeschneiden versetzt angeben. Dadurch ist es möglich, mehrgängige Gewinde zu schneiden. Ein Gang eines mehrgängigen Gewindes wird in der gleichen Weise wie ein **eingängiges Gewinde** programmiert. Nach der kompletten Bearbeitung des ersten Ganges wird der **Startpunkt um h' versetzt** und der nächste Gang bearbeitet.

$$h' = \frac{\text{Gewindesteigung}}{\text{Gangzahl}}$$

Die einzelnen Gänge müssen mit **gleicher Spindeldrehzahl** ausgeführt werden, um unterschiedliche Schleppabstände zu vermeiden.

Beispiel: Mehrgängiges Gewinde mit konstanter Steigung



Steigung $h = 6 \text{ mm}$
 Gewindetiefe $t = 3.9 \text{ mm}$
 Radiale Zustellrichtung
 zweigängig

Im Beispiel wird jeder Gang in zwei Schritten bearbeitet. Nachdem der erste Gang fertig bearbeitet ist, wird der zweite Gang bearbeitet, in dem der Startpunkt um h' versetzt liegt.

$$h' = \text{Gewindesteigung} / \text{Gangzahl}$$

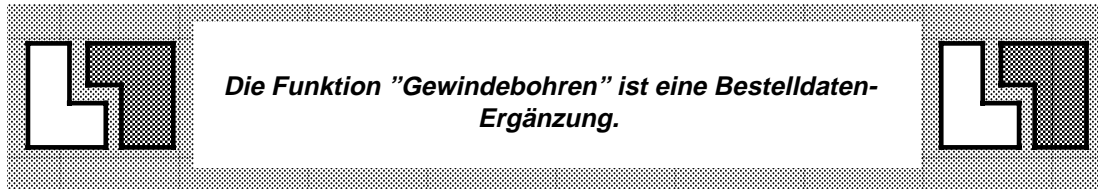
$$h' = 6 / 2 = 3 \text{ mm.}$$

Die Z-Koordinate liegt deshalb beim 2. Gang statt auf Z115 auf Z118.

```
%392 (G33, MEHRGÄNGIGES GEWINDE) L_F
N05 G90 S50 M3 L_F
N10 G00 X66 Z115 L_F
N15 X56.1 L_F
N20 G33 Z30 K6 L_F
N25 G00 X66 L_F
N30 Z115 L_F
N35 X52.2 L_F
N40 G33 Z30 K6 L_F
N45 G00 X66 L_F
N50 Z118 L_F
N55 X56.1 L_F
N60 G33 Z30 K6 L_F
N65 G00 X66 L_F
N70 Z118 L_F
N75 X52.2 L_F
N80 G33 Z30 K6 L_F
N85 G00 X66 L_F
N90 M30 L_F
```

(P1)
(P2)
(P3)
(P4)
(P1)
(P5)
(P6)
(P4)
(P7) } 1. Gang
(P8)
(P3)
(P4)
(P7)
(P9) } 2. Gang
(P6)
(P4)

3.2.6 Gewindebohren ohne Geber G63



Die Wegbedingung G63 wird zum Bohren von Gewinden mit einem **Gewindebohrer im Ausgleichsfutter** programmiert. Ein Funktionszusammenhang zwischen Spindeldrehzahl und Vorschub ist **nicht** gegeben.

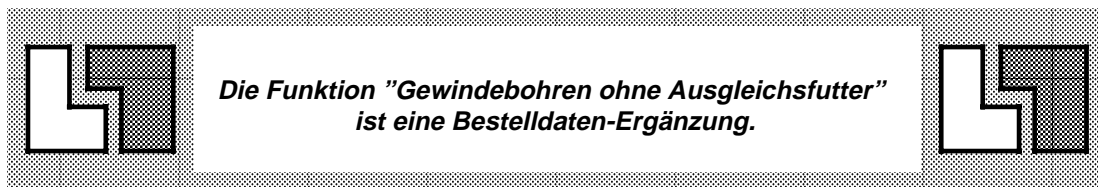
Unter der Adresse **S** wird die Spindeldrehzahl und unter der Adresse **F** ein dazu passender Vorschub programmiert.

Das Längenausgleichsfutter muß die Toleranzen zwischen Vorschub und Drehzahl sowie den Spindelauslauf nach Erreichen der Position aufnehmen können.

Bei G63 ist der **Vorschubkorrektorschalter auf 100%** festgelegt. Abhängig von der Auslegung der Anpaßsteuerung wird bei "Vorschub Halt" auch die Spindel stillgesetzt. Der Spindel-drehzahlkorrektorschalter ist wirksam.

G63 kann nur in Sätzen **mit** Geradeninterpolation **G01** angewendet werden.

3.2.7 Gewindebohren ohne Ausgleichsfutter G36



Voraussetzung für diese Funktion ist der Rundachsbetrieb der Spindel.

Die Wegbedingung G36 wird zum Bohren (bzw. auch Schneiden bei Drehmaschinen) von Gewinden ohne Ausgleichsfutter programmiert. Dabei interpolieren die Spindel in Rundachsbetrieb und die Zustellachse(n). Damit ist der Einsatz eines Ausgleichsfutters für Gewindebohren bzw. -schneiden (bei Drehmaschinen) nicht nötig.

Mit G36 können folgende Gewindearten realisiert werden:

- Zylindrische Gewinde (Spindel im Rundachsbetrieb und **eine** Linearachse)
- Kegelgewinde (Spindel im Rundachsbetrieb mit **zwei** Linearachsen)

Davon abweichende Programmierungen mit G36 sind nicht erlaubt!

Zunehmende und abnehmende Gewindesteigungen (wie bei G34, G35) sind mit G36 nicht realisierbar.

Beim Schreiben von G36 wird automatisch der Vorschub für die Rundachse in U/min (G98) wirksam. Ein anderer Vorschubtyp für die Rundachse darf in einem G36-aktiven Programmteil nicht angewählt werden.

Die Funktion G36 bleibt bis zur Abwahl durch eine andere G-Funktion der G-Gruppe 0 (z. B. G0) wirksam. Mit Abwahl von G36 wird auch dazugehörige G98 abgewählt. Die vorhergehende Vorschubbedingung wird wieder wirksam.

Programmierbedingungen:

- Der Rundachsbetrieb der Spindel muß angewählt sein.
- In jedem G36-aktiven Satz darf nur und muß die Rundachse mit den zugehörigen Linearachsen geschrieben werden.
Ein Zahlenwert hinter der Rundachse wird ignoriert. Der Vorschubwert F in U/min für die Rundachse ist im ersten G36-aktiven Satz zu schreiben.
- Die Gewindesteigung wird mit der Adresse "I", "J" oder "K" angegeben (Vorgabe des Maschinenherstellers beachten). Das Vorzeichen des Wertes bei "I", "J" oder "K" gibt die Drehrichtung der Rundachse an. Plus bedeutet rechtsdrehend, minus bedeutet linksdrehend.
- Es müssen genau **eine** Rundachse **und eine** Linearachse für zylindrische Gewinde bzw. **eine** Rundachse mit **zwei** Linearachsen für Kegелgewinde programmiert werden. Abweichende Achsprogrammierungen erzeugen den Alarm 3006.
Beim Kegелgewinde wird der Steigungsparameter "I", "J" oder "K" der Linearachse mit dem größeren Verfahrweg zugeordnet. Es sind somit Kegелgewinde bis max. 45° Neigungswinkel möglich.

Beispiel 1: Zylindrisches Gewinde mit Zuordnung von "K" für Zustellachse Z

N10 M1=70 L _F	Anwahl Rundachsbetrieb für die Spindel
N15 G0 C30 Z2 L _F	Rundachse C auf 30 Grad und Zustellachse Z auf Gewindeeinsatzpunkt positionieren
N20 G36 C Z-30 K5 F10 L _F	Gewindebohrung starten Gewindesteigung 5 mm/U rechtsdrehend und Vorschub der C-Achse 10 U/min
N25 C Z2 K-5 L _F	Gewindebohrung Gewindesteigung 5 mm/U linksdrehend und Vorschub der C-Achse 10 U/min (zurück zum Gewindeeinsatzpunkt)

Beispiel 2: Kegелgewinde mit Zuordnung von "K" für Zustellachse

N10 M1=70 L _F	Anwahl Rundachsbetrieb für die Spindel
N15 G0 C30 X10 Z2 L _F	Positionieren von C-Achse auf 30 Grad und Zustellachsen X, Z auf Gewindeeinsatzpunkt
N20 G36 C Z-30 X20 K5 F10 L _F	Vorschub der C-Achse 10 U/min rechtsdrehend, "K" wirkt auf Z, weil der Weg bei Achse Z größer ist als der Weg bei Achse X

Reset-Verhalten:

Mit M02/M30 bzw. Bedientafelreset wird die in den Maschinendaten hinterlegte Löschestellung aktiv (bezüglich G36 bzw. G98).

Anzeige:

- G36 und G98 werden im Anzeigefeld "G-Funktionen" angezeigt.
- Bei aktivem G98 erfolgt die Anzeige des Vorschubs, bezogen auf die Rundachse in U/min (Kennung "U").
- Der Vorschubtyp "U/min" wird in der Vorschubanzeige mit dem Kürzel "U" bezeichnet.

Verhalten nach Satzvorlauf

Bei NC-Start eines G36-Satzes nach Satzvorlauf muß sich die Spindel im Rundachsbetrieb befinden. Andernfalls wird der achsspezifische Reset-Alarm 196* "Nachf./Parken f. Achse" gemeldet. Im Nichtfehlerfall wird als Vorschub der programmierte G98-Wert (U/min) verwendet.

Aus technologischer Sicht sollte auf einen dem Gewindeschneiden voranstehenden Positioniersatz vorgelaufen werden bzw. in der Betriebsart "REPOS" alle aufgesammelten Wege und Verschiebungen abgefahren werden.

Hinweis:

G36 erzwingt Satzendgeschwindigkeit 0. Damit ist eine Verweilzeit technologisch nicht mehr nötig.

3.2.8 Genauhalt G09/G60/G00, Bahnsteuerbetrieb G62/G64

3.2.8.1 Genauhaltgrenze fein und grob G09/G60/G00

G09/G60 Genauhaltgrenze fein.
G00 Genauhaltgrenze grob.

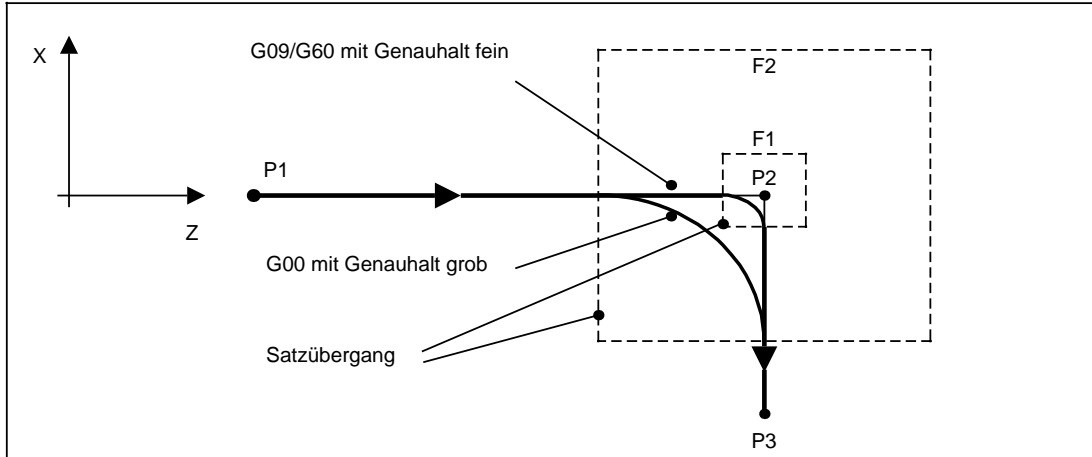
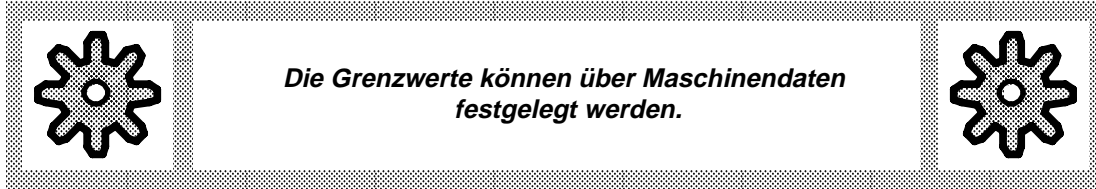
Mit G09, G60 bzw. G00 ist es möglich, eine Zielposition innerhalb einer vorgegebenen **Genauhaltgrenze** anzufahren. Bei Erreichen des **Genauhaltfensters** wird die Vorschubgeschwindigkeit der verfahrenen Achse (von P1 nach P2, siehe nächste Seite) auf 0 verringert. Der Schleppabstand wird abgebaut. Gleichzeitig wird der Satzwechsel eingeleitet und die im nächsten Satz programmierte Achsbewegung (von P2 nach P3) beginnt.

G09 ist satzweise, G60 modal wirksam.

Die Funktion G09 bzw. G60 kann z.B. verwendet werden, wenn scharfe Ecken zu bearbeiten sind, beim Einstechen oder bei einer Richtungsumkehr.

Genauhaltgrenze fein und grob

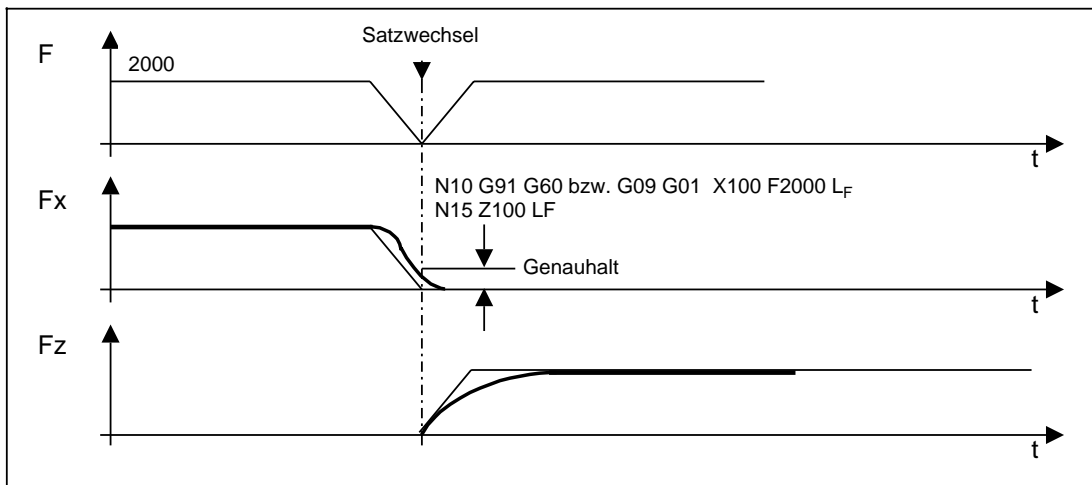
Genauhalt fein: 10 μm (Fenster F1).
 Genauhalt grob: 250 μm (Fenster F2).



Genauhaltfenster

Falls beide Genauhalt-Grenzen gleich groß sind (Fenster F1 = Fenster F2), verhält sich G00 wie G09/G60. In der Regel hat die **Eilgangbewegung ein größeres Genauhaltfenster**. Damit wird bei Eilgangbewegungen eine Zeitersparnis erreicht (früherer Satzwechsel).

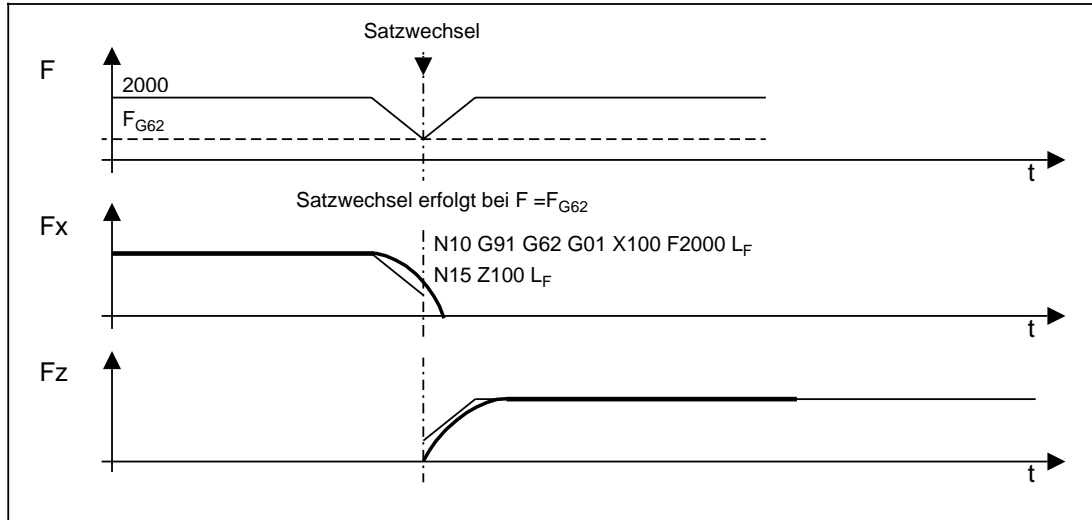
Mit der dünnen Linie ist die Geschwindigkeitsführung der Steuerung dargestellt. Durch den Lagereger innerhalb der NC ergibt sich ein verschliffener Kurvenverlauf (dicke Linie).



Genauhalt G60/G09

3.2.8.2 Bahnsteuerbetrieb G62/G64

Mit der Funktion G62 wird der **Vorschub F zum Satzende** hin auf eine im **MC-MD 3** festgelegte Reduziergeschwindigkeit **verringert**. Bei G64 wird der Vorschub nicht reduziert (Grundstellung bei Bahnsteuerbetrieb). G62 und G64 sind modal wirksam.



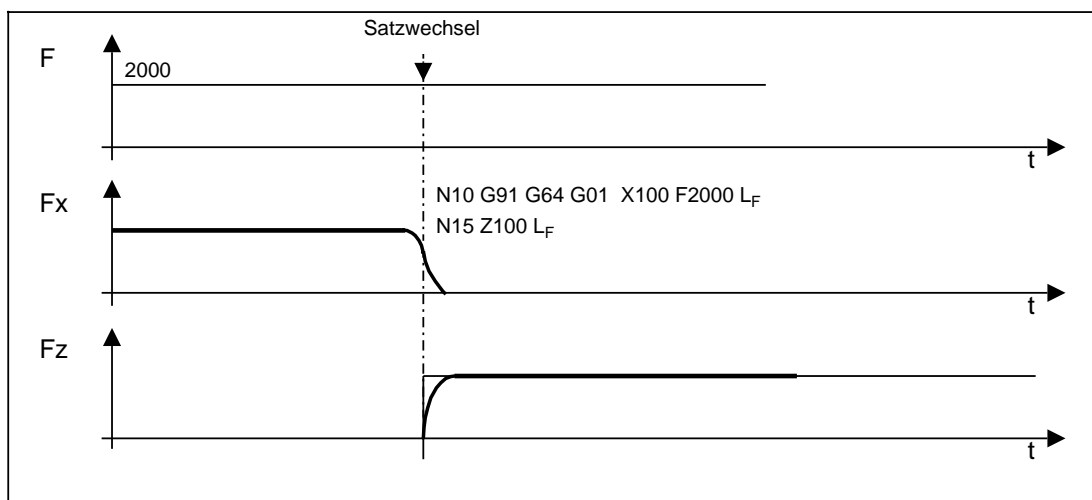
Bahnsteuerbetrieb mit G62

Anwendung:

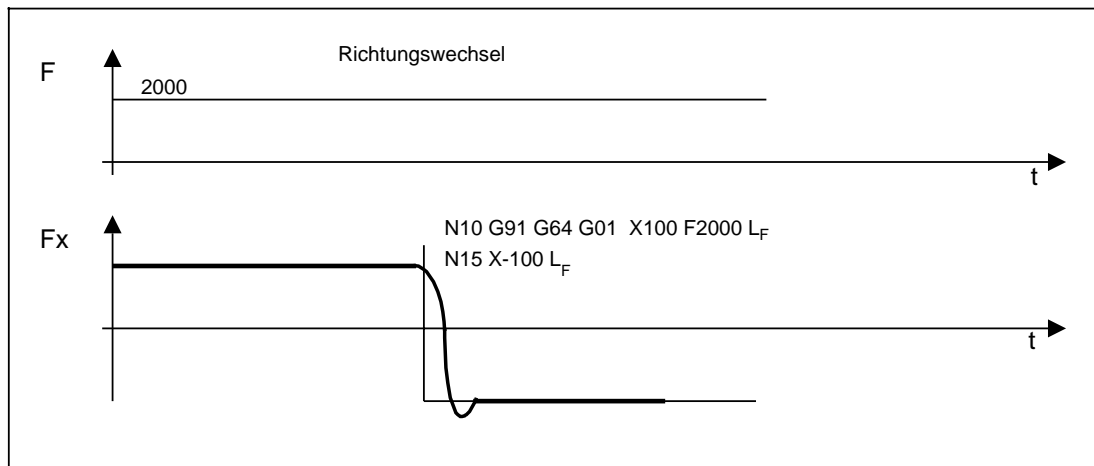
Bei der Holzverarbeitung darf beim Satzübergang der Vorschub nicht zu Null werden, da sonst Brandmarken am Werkstück entstehen würden.

Satzübergang ohne Geschwindigkeitsreduzierung G64

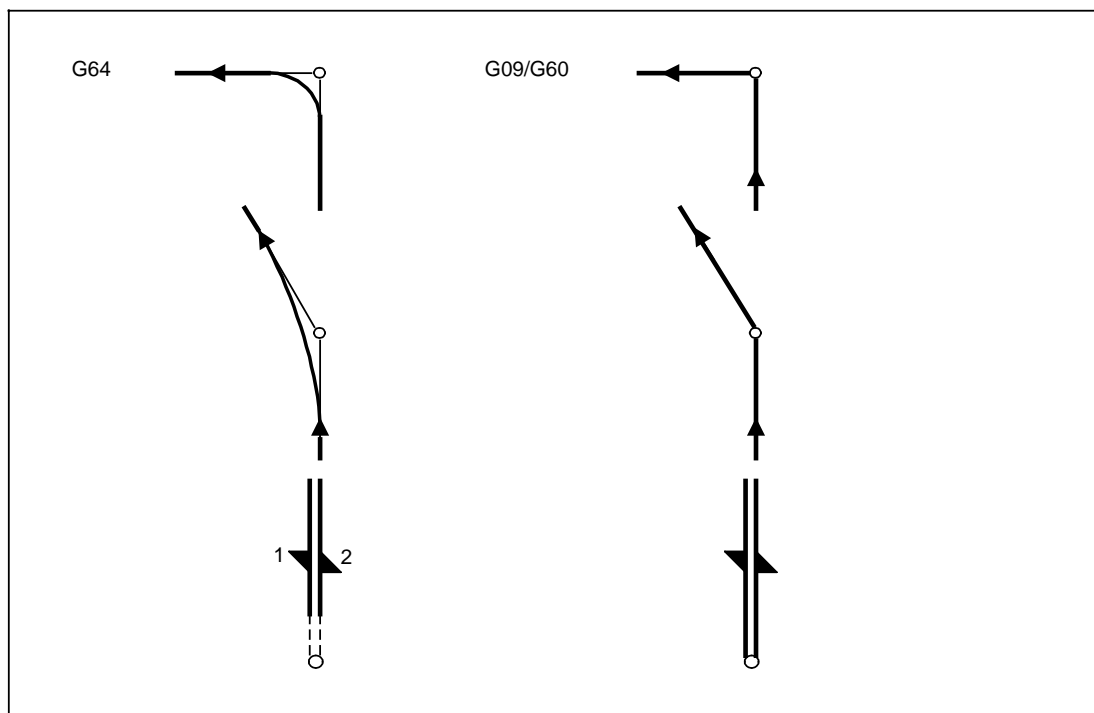
Die Wegbedingung G64 wird verwendet, wenn **kein Freischneiden bei Übergängen** von Satz zu Satz auftreten soll. Außerdem werden damit bei Änderung der Bewegungsrichtung die Übergänge verschliffen.



Bahnsteuerbetrieb mit G64 ohne Geschwindigkeitsreduzierung für unterschiedliche Achsen



Bahnsteuerbetrieb mit G64 und Richtungswechsel einer Achse



Richtungswechsel mit (G09/G60) und ohne (G64) Geschwindigkeitsreduzierung

3.2.9 Verweilzeit G04

Verweilzeiten werden benötigt beim Freischneiden, evtl. bei Drehzahlwechsel und Maschinenschaltfunktionen (Lünette, Reitstock u.s.w.).

Die Verweilzeit wird immer ohne Vorzeichen eingegeben.

Die Verweilzeit geben Sie unter der Adresse **X** oder **F** an. Der **Zeitbereich** liegt zwischen:

- 0.001 bis 99999.999 s bei X,
- 0.001 bis 99.999 s bei F und
- 0.1 bis 99.9 Spindelumdrehungen.

G04 ist **satzweise** wirksam. In einem Satz mit Verweilzeit dürfen **keine weiteren Funktionen** geschrieben werden.

Beispiel:

```
N10 G04 X11.5 LF
```

Verweilzeit 11,5 s immer ohne Vorzeichen

Eine weitere Möglichkeit besteht darin, die Verweilzeit in Umdrehungen der Spindel zu programmieren. Die Programmierung erfolgt unter **S** im Bereich von **0,1 bis 99,9 Umdrehungen**.

Beispiel:

```
N10 G04 S5 LF
```

Verweilzeit 5 Spindelumdrehungen

3.2.10 Weiches Anfahren und Verlassen der Kontur

Um Schneidmarken zu vermeiden, wird eine Kontur **tangential** angefahren bzw. verlassen.

Weiches Anfahren und Verlassen einer Kontur kann **bei allen Werkzeugtypen** angewendet werden. Verrechnet werden "Radius" und "Verschleiß".

Das **Anfahren** an die Kontur bzw. das **Verlassen** der Kontur kann mit folgenden **Funktionen** programmiert werden:

G147 Anfahren linear.

G247 Anfahren im Viertelkreis.

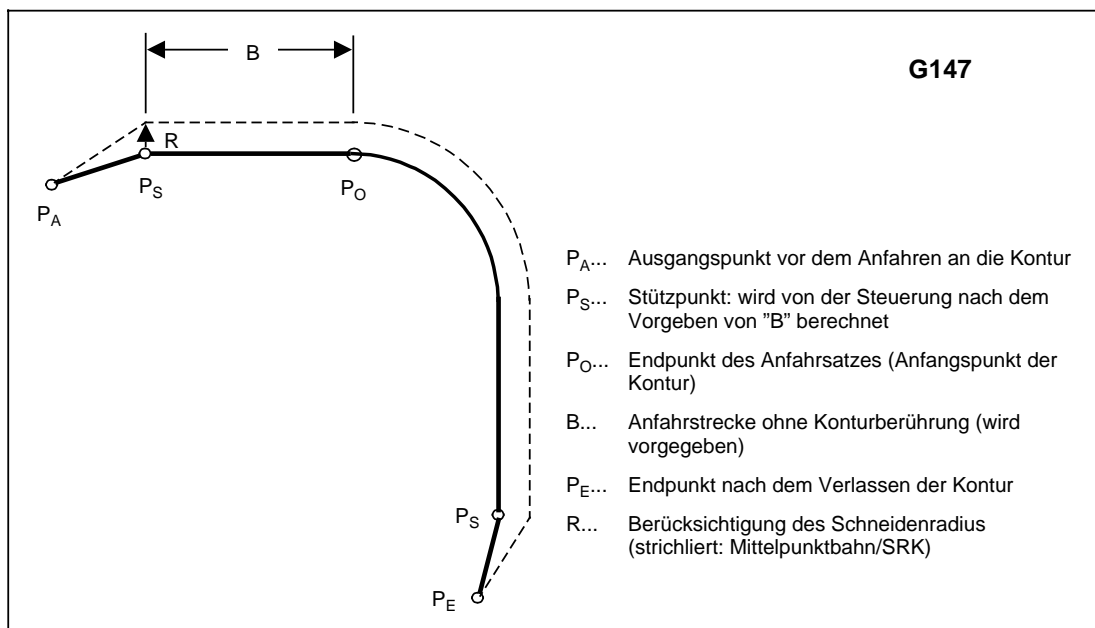
G347 Anfahren im Halbkreis.

G148 Verlassen linear.

G248 Verlassen im Viertelkreis.

G348 Verlassen im Halbkreis.

G48 Verlassen der Kontur in der gleichen Weise, wie sie angefahren wurde.

Beispiel: Gerade an Kreis

3.2.11 Polarkoordinaten G10/G11/G12/G13

Mit **Winkel und Radius** bemaßte Zeichnungen können mit Hilfe von Polarkoordinaten **direkt** ins Programm eingegeben werden.

Für die **Programmierung mit Polarkoordinaten** stehen folgende Wegbedingungen zur Verfügung:

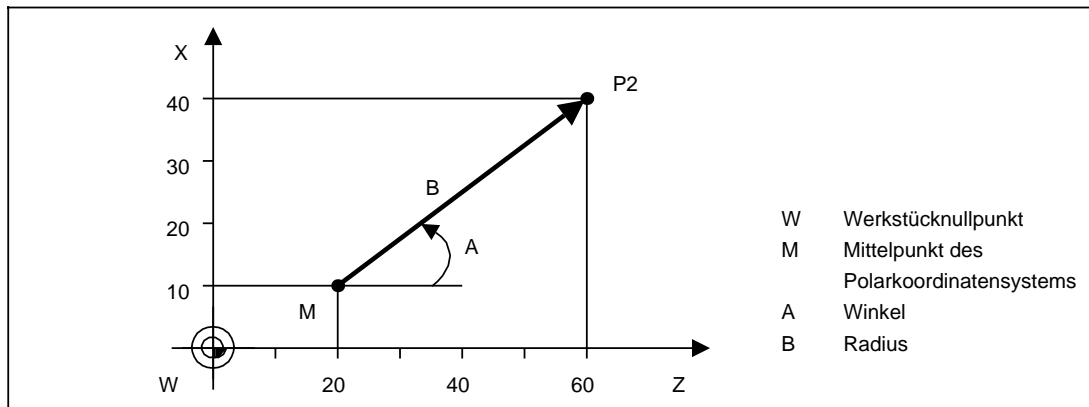
G10 Geradeninterpolation Eilgang.

G11 Geradeninterpolation Vorschub (F).

G12 Kreisinterpolation im Uhrzeigersinn.

G13 Kreisinterpolation im Gegenuhrzeigersinn.

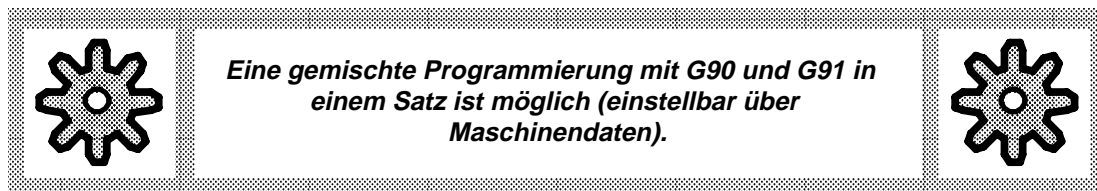
Die Wegbedingungen sind **modal** wirksam.



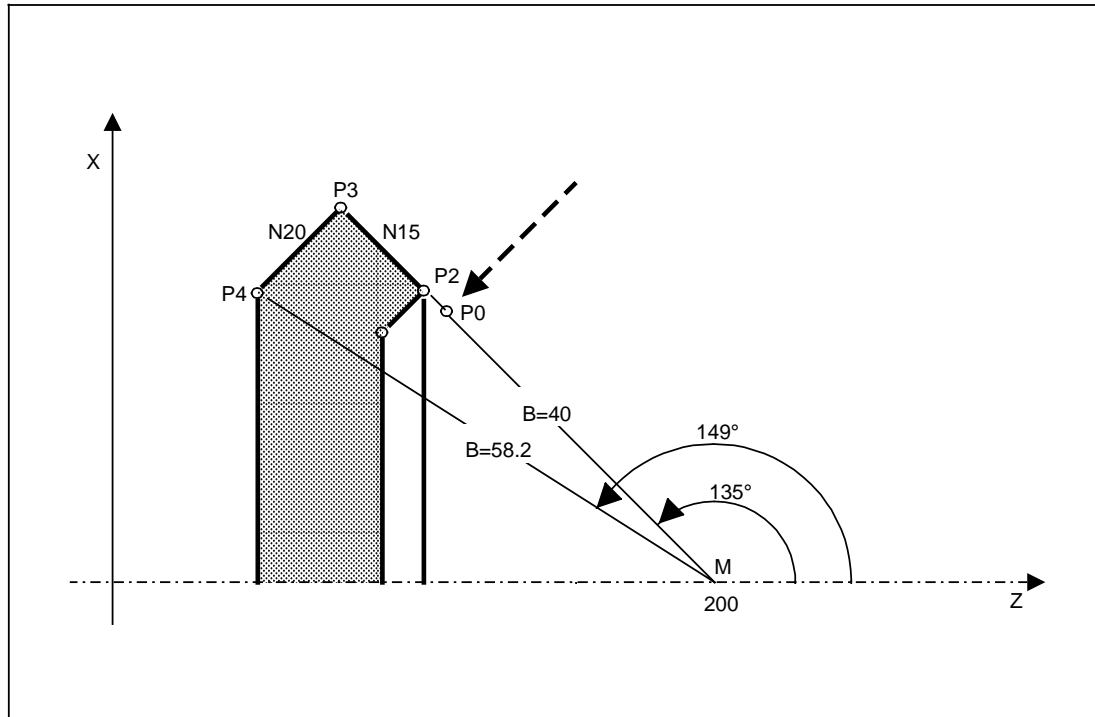
Zur Bestimmung des Verfahrenswegs benötigt die Steuerung die **Angabe** des **Mittelpunktes**, des **Radius** und des **Winkels**. Der Mittelpunkt wird mit rechtwinkligen Koordinaten (X, Z) und bei erstmaliger Programmierung im Bezugsmaß eingegeben. Eine spätere Kettenmaßeingabe (mit G91) bezieht sich immer auf den zuletzt programmierten Mittelpunkt.

Die Angabe des Mittelpunktes ist modal wirksam und wird mit M02/M30 gelöscht.

- Der **Radius** wird unter der Adresse **B** ohne Vorzeichen programmiert.
- Der **Winkel** wird unter der Adresse **A** ohne Vorzeichen eingegeben (Eingabefeinheit 10^{-5} Grad). Er bezieht sich immer auf die zuerst programmierte positive Achse der Mittelpunkt-Koordinaten (Bezugsachse).
 - Die positive Richtung dieser Achse entspricht einem Winkel von 0 Grad.
 - Die Winkelangabe ist absolut oder inkrementell und positiv.



Beispiel: G10, G11 Drehmaschine



```
N10 G90 G10 Z200 X0 B40 A+135 L_F      (P0)   im Eilgang Startposition P0 anfahren
N15 G11 B56.56 L_F                    (P3)   Außenkontur des Drehteils mit Vorschub F. . . bearbeiten.
N20 B58.2 A+149 L_F                    (P4)
```

3.2.11.1 Polarkoordinaten G110/G111

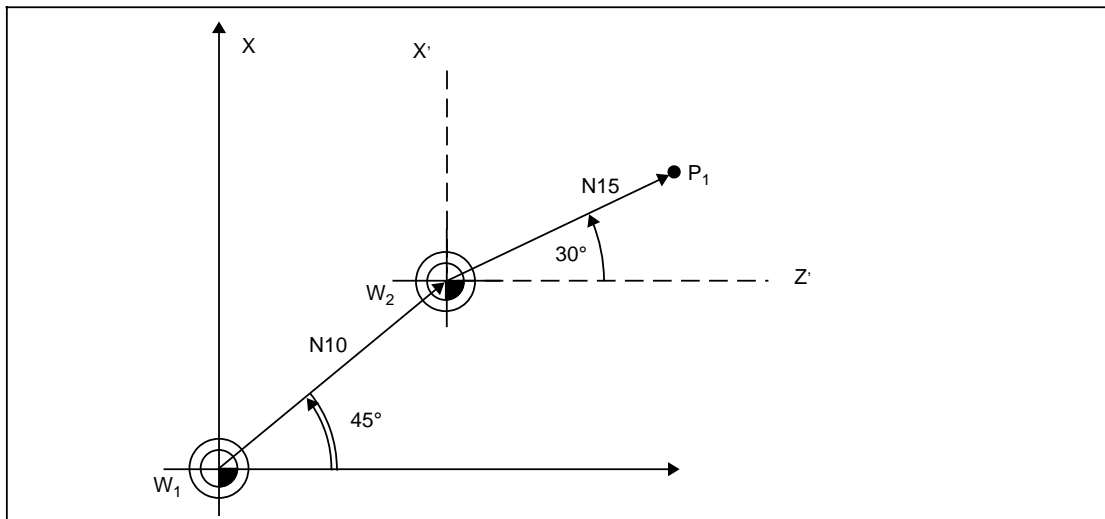
Die Funktionen G110 und G111 werden zur Übernahme eines neuen **Mittel- bzw. Nullpunktes** bei Programmierung mit Polarkoordinaten benützt.

Von diesem neuen Mittelpunkt aus, werden die Winkel wieder von der Waagerechten aus gezählt und der Radius wird von dem neuen Mittelpunkt aus gerechnet. Es bedeuten:

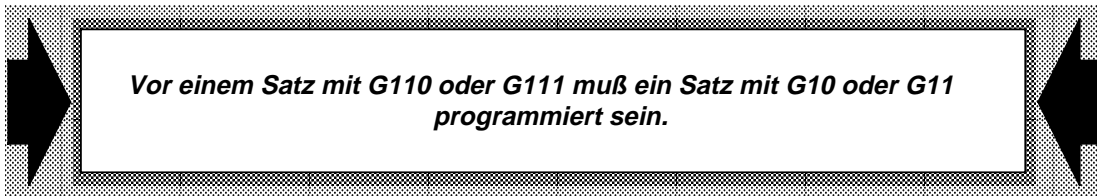
- G110** Erreichte Sollposition als neuen Mittelpunkt übernehmen
- G111** Mittelpunktprogrammierung mit Winkel und Radius, ohne Achsbewegung (Beispiel: Kreismittelpunkt eines Lochkreises setzen; die nachfolgende Verfahrbewegung muß mit G110 programmiert werden)

Beispiel: Polarkoordinaten G110

```
%385
(G110 Polarkoordinaten) L_F
N5 G90 G10 Z0 X0 B0 A0 F1000 L_F
N10 G11 B30 A45 L_F
N15 G110 B20 A30 L_F
N20 M30 L_F
```



Polarkoordinaten G110



Die Funktionen G110/G111 sind **satzweise** wirksam. Sie sind nur für **Geradeninterpolation** gültig.

Als Vorschub wirkt der zuletzt programmierte F-Wert (G11) oder Eilgang (G10).

Beispiel: Polarkoordinaten G111

%386

(G111 Polarkoordinaten) L_F

N5 G11 Z40 X40 A B F1000 L_F

N10 G111 A225 B28 L_F

N15 G110 A90 B20 L_F

N20 A180 L_F

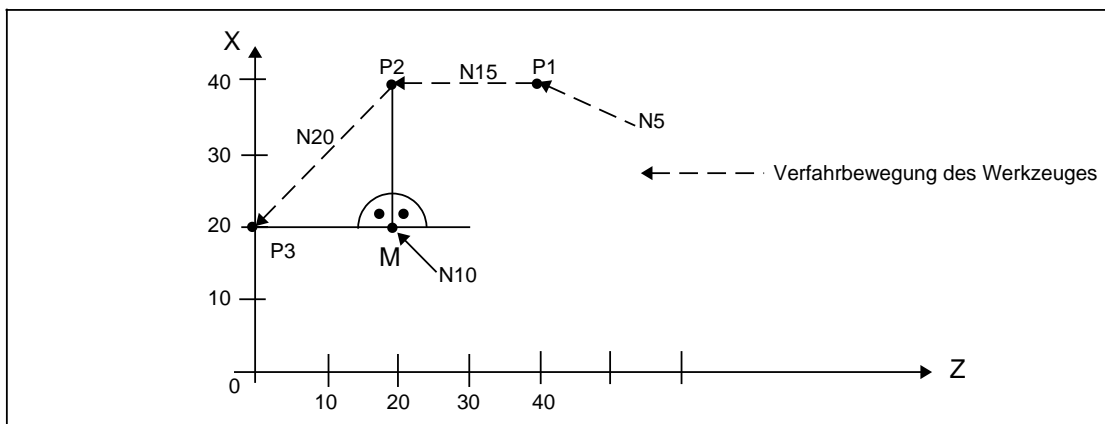
N25 M30 L_F

P₁

Mittelpunkt setzen ohne Verfahren

P₂

P₃

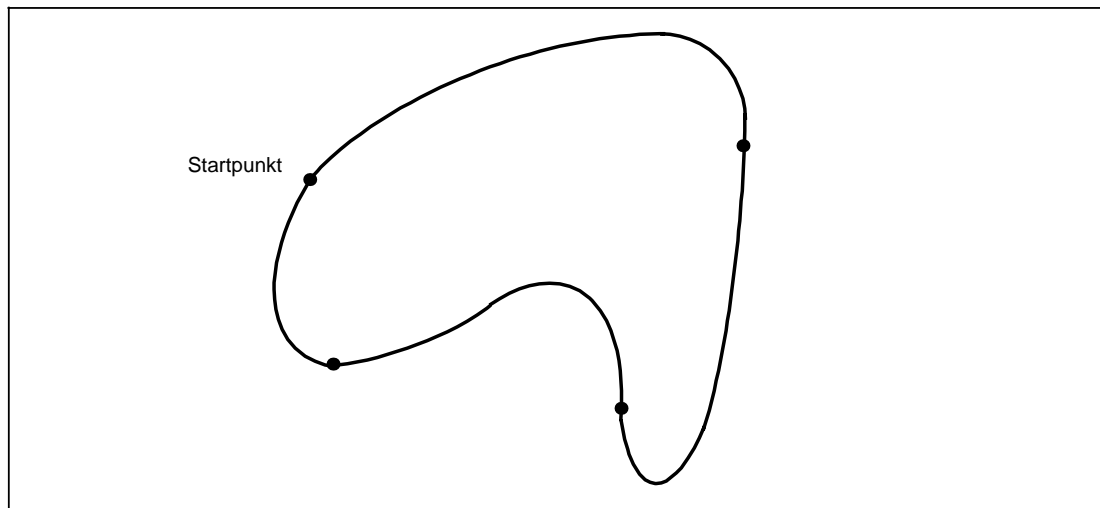


Polarkoordinaten G111

3.2.12 SPLINE - Interpolation G06



Ein Spline ist eine **Verkettung von Kurvenzügen**, die an ihren Verbindungspunkten gleichen Funktionswert, gleiche Steigung und gleiche Krümmung aufweisen.



SPLINE - Interpolation G06

Die SPLINE - Interpolation verringert den Programmieraufwand bei der Bearbeitung von **komplexen** Werkstückkonturen und ermöglicht die Bearbeitung von Formen, die durch Standardgeometrien nicht zu beschreiben sind. Die Programmierung der SPLINE - Interpolation ist in einer gesonderten **Druckschrift** "Programmierung der SPLINE - Interpolation" (Best.Nr. 6ZB5 410-7BA01-0BA1) beschrieben. G06 ist modal wirksam und wird durch eine andere Wegbedingung der 0. G-Gruppe abgewählt.

3.2.13 Koordinatentransformation TRANSMIT



Definition: TRANSMIT = **TRANS**formation **M**illing Into **T**urning,
Transformation einer Fräsbearbeitung in eine Drehbearbeitung.

Die Koordinatentransformation TRANSMIT ermöglicht Ihnen die stirnseitige Fräsbearbeitung von Drehteilen an Drehmaschinen (810/820T). Das gewünschte Programm erstellen Sie in einem **fiktiven** (kartesischen) Koordinatensystem. Maschinenbewegungen werden in einem **realen** Maschinenkoordinatensystem ausgeführt. Für das fiktive Koordinatensystem werden die Achsen vom Werkzeugmaschinen-Hersteller über Maschinendaten definiert.

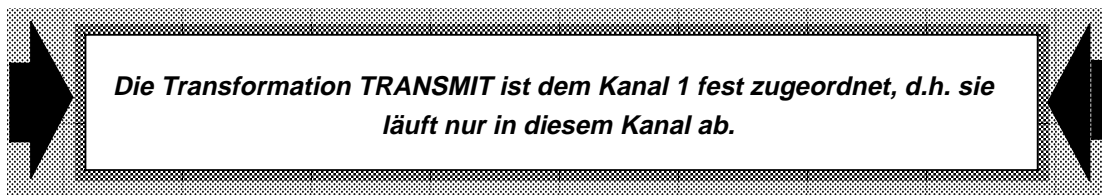
Eine fiktive Achse kann nur bei angewählter Transformation verfahren werden. Gleichzeitig mit der Regelung der fiktiven Achsen werden von der NC die zum Transformationsverband zugehörigen realen Achsen verfahren.

Zu den maximal fünf realen Achsen kommen zusätzlich zwei fiktive Achsen. Die fiktiven Achsen werden von der NC gleichwertig zu realen Achsen behandelt. Die einzige Ausnahme bildet die Lageregelung. Hier werden nur die fünf realen, an die Meßkreise angeschlossenen Achsen bearbeitet.

Die Funktion TRANSMIT ist definiert durch einen Transformationsdatensatz. Dieser umfaßt folgende Informationen:

- G-Funktion für TRANSMIT: G131
- die Namen der an der Transformation beteiligten fiktiven Achsen (programmierte Achsen)
- die Namen der von der Transformation generierten und zu verfahrenen Achsen (reale Achsen)
- den Namen der Zustellachse.

Dieser Transformationsdatensatz wird vom Werkzeugmaschinen-Hersteller festgelegt und muß Ihnen vor der Programmierung bekannt sein.

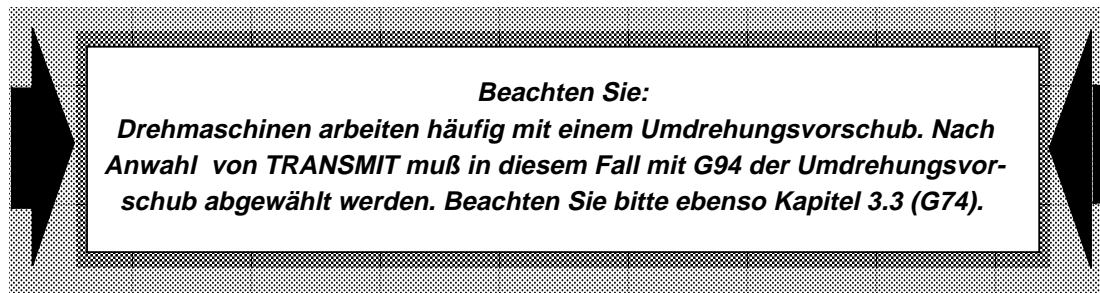


3.2.13.1 Funktion TRANSMIT

Die in den kanalspezifischen Maschinendaten festgelegte Ebenendefinition gilt für das reale System. Bei TRANSMIT ist G17 als Grundebene definiert (fiktive Ebene), d.h. G17 wird nach Transformationsanwahl automatisch eingestellt. Nach der Transformationsabwahl schaltet die Steuerung ebenfalls **automatisch** auf die vor der Transformationsanwahl wirksame Ebene zurück.

Die **fiktive Ebene** ist in den Transformationsdaten definiert durch die Zuordnung der fiktiven Achsen. Abweichungen von den Grundebenen können Sie über G16 programmieren. Die Achsbezeichnung für fiktive und reale Achsen muß eindeutig sein, d.h. die mehrfache Verwendung eines Achsnamens ist nicht zulässig.

Die Funktion TRANSMIT wählen Sie mit der G-Funktion G131 an. Diese muß einzeln in einem Satz stehen, d.h. es dürfen in diesem Satz keine Verfahrbewegungen oder weitere Funktionen programmiert sein. Ansonsten meldet die Steuerung den Alarm 2043 "Programmierfehler bei Transformation".



TRANSMIT können Sie durch den Abwahlsatz G130 wieder abwählen. **In Löschstellung** (G130) dürfen keine fiktiven Achsen programmiert werden. Die an der Transformation beteiligten realen Achsen hingegen kann man nur in der Löschstellung (G130) programmieren. Beachten Sie diese Bedingungen nicht, erscheint jedesmal der Alarm 2043.

Bei angewählter Transformation TRANSMIT

Die realen Achsen des Transformationsverbandes sind verriegelt. Die Bearbeitungsebene ist jetzt durch das fiktive System definiert. Linear- und Zirkularbewegungen führen zu einem definierten Bahnverlauf.

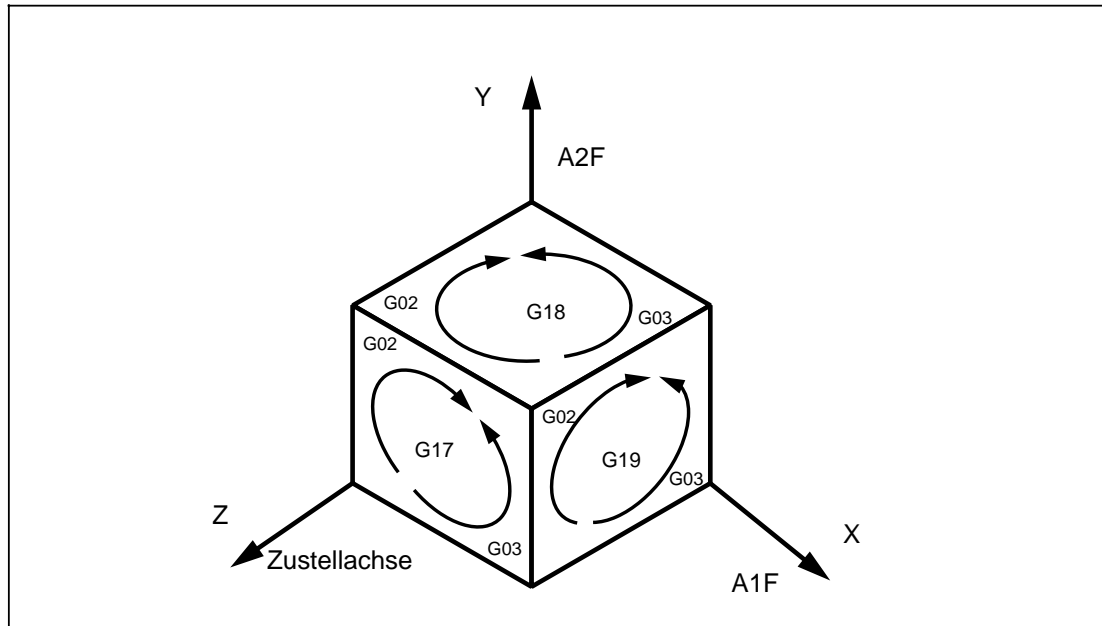
Die Arbeitsfeldbegrenzungen (G25/G26) können nur im fiktiven KS (kartesisches KS) programmiert werden.

Werkzeugkorrekturen und Nullpunktverschiebungen beziehen sich ebenfalls nur auf das fiktive KS.

Verwenden Sie bei TRANSMIT Werkzeuge mit der Längenkorrektur L1, so müssen Sie die Achse vorgeben, in welcher die Korrektur wirksam sein soll. Diese können Sie entweder über die freie Ebenanwahl G16 einrichten

- Befehl: G16<1. fiktive Achse><2. fiktive Achse> <Zustellachse>
oder
- direkt über das MD 5064 (Name der realen Zustellachse) angeben.

In beiden Fällen steht die Längenkorrektur L1 in einer realen Achse.

Beispiel: Ebenendefinition

N05 G17	Bearbeitungsebene X-Y
N10 G00 X Y	Ebene X-Y
N15 G16 Q1 B	Ebenanwahl Q1-B
N20 G01 Q1=20 B=30	Ebene Q1-B
N25 G131	Anwahlsatz der Transformation TRANSMIT
.	Bearbeitungsebene A1F,A2F
N40 G130	Abwahlsatz der Transformation TRANSMIT
.	Bearbeitungsebene Q1-B
.	
N60 G17	Bearbeitungsebene X-Y

Erläuterung:

A1F ... A2F:

1. ... 2. fiktive Achse (A1F und A2F ist durch sinnvolle Achsnamen zu ersetzen)

Jede An-/Abwahl von TRANSMIT ist mit der Funktion „Zwischenspeicher leeren“ (@714) verbunden. Diese Funktion wird von der Satzaufbereitung intern ausgelöst, muß also nicht programmiert werden. Die Fräser-/Schneidenradiuskorrektur (FRK/SRK) müssen Sie vor Anwahl von TRANSMIT abwählen (bedingt durch @714). Ansonsten erscheint der Alarm 2081"SRK nicht erlaubt".

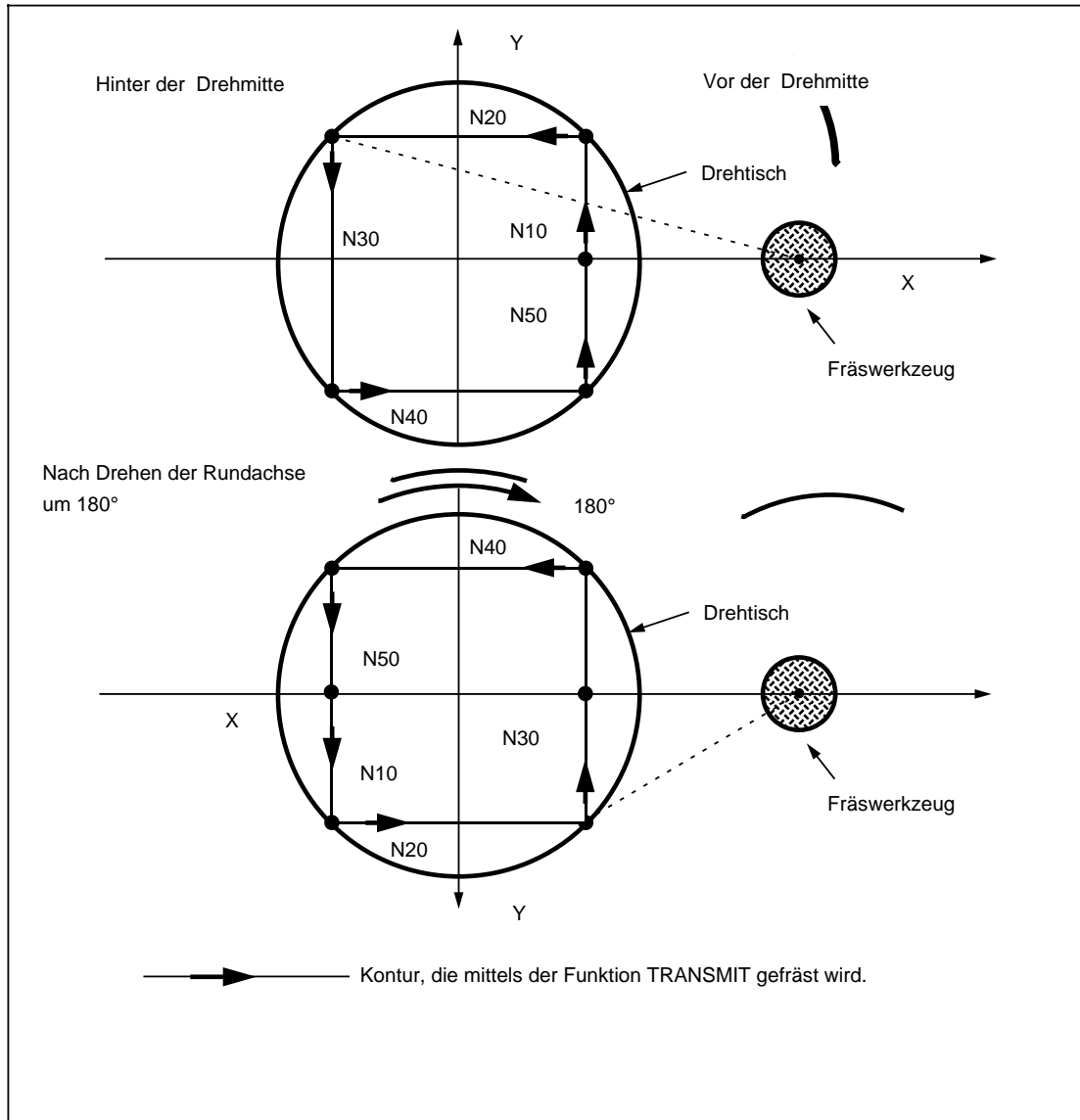
Die FRK/SRK müssen Sie vor Abwahl von TRANSMIT wieder abwählen, da bei TRANSMIT-Abwahl die Bearbeitungsebene gewechselt wird. Nach Abwahl von TRANSMIT stellt die Steuerung jene Bearbeitungsebene ein, die vor der Transformationsanwahl aktiv war. Erst nach Abwahl bzw. Anwahl der Funktion TRANSMIT können Sie die FRK/SRK wieder anwählen.

Innerhalb einer Kontursatzfolge darf TRANSMIT nicht an- oder abgewählt werden. Satzvorlauf auf einen Programmteil, in dem TRANSMIT aktiviert wird, ist zulässig (siehe Kap. 3.2.12.2).

3.2.13.2 Satzvorlauf bei der Funktion TRANSMIT

Die Funktion "Satzvorlauf" ist bei aktiver Funktion TRANSMIT uneingeschränkt einsetzbar. Die Steuerung generiert nach "Satzvorlauf" und "NC-Start" einen Linearsatz von der augenblicklichen Position auf die Zielposition (**Vorlaufsatz**).

Liegt die Zielposition vom Werkzeug aus betrachtet hinter der Drehmitte, kann es während der Bearbeitung des Vorlaufsatzes zu einer Kollision zwischen Werkzeug und Werkstück kommen. In diesem Fall muß die Rundachse im REPOS-Betrieb solange verfahren werden, bis die Zielposition vor der Drehmitte liegt. Steht das Werkzeug in der Drehmitte, so erscheint bei Anwahl von TRANSMIT der Alarm 2191 "Transformation im Nullpunkt". Dann muß im JOG-Betrieb das Werkzeug aus der Drehmitte gefahren werden.



Manuelle Annäherung an die Zielposition

3.2.13.3 Prinzip der Koordinatentransformation TRANSMIT

In CNC-Steuerungen verwendet man 2 Arten von Koordinatensystemen:

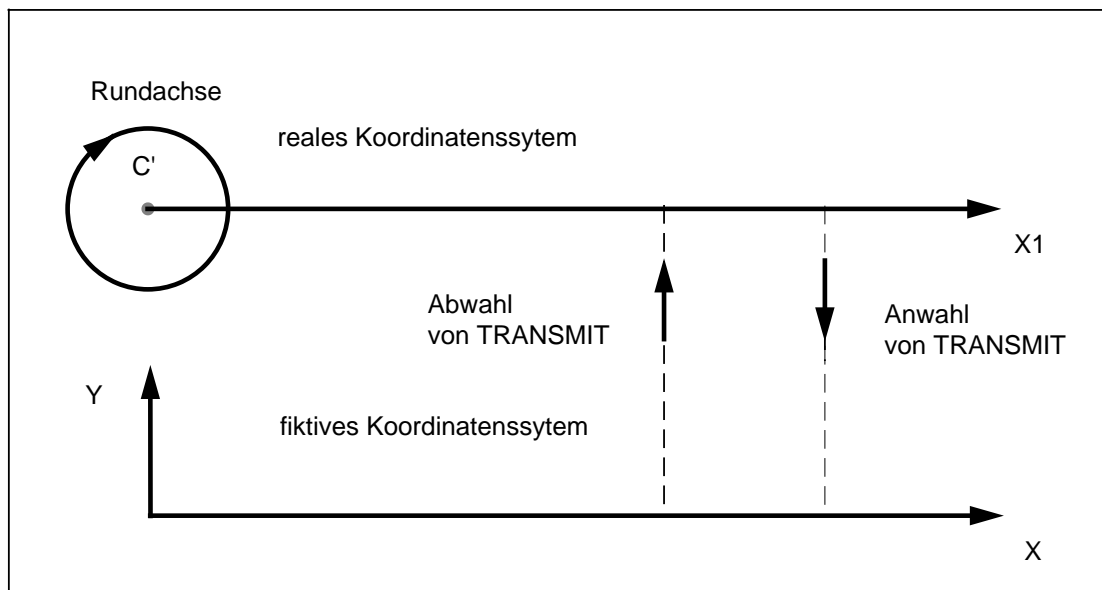
- a) **Fiktives Koordinatensystem** (fiktives KS):
Kartesisches Koordinatensystem (KS) zum Erstellen von z.B. Teileprogrammen für das Fräsen auf Drehmaschinen.
(Werkstücknahes-Koordinatensystem, Steuerungs-KS, Interpolations-KS).
- b) **Reales Koordinatensystem** (reales KS):
Koordinatensystem, welches die reale Anordnung der Maschinenachsen berücksichtigt (Maschinen-KS).

Eine Koordinatentransformation beschreibt den mathematischen Zusammenhang zwischen zwei verschiedenen Koordinatensystemen durch ein System von Transformationsgleichungen. Dieses Gleichungssystem muß für beide Transformationsrichtungen (Hin- und Rücktransformation) gelten. Die Koordinaten gibt man im kartesischen Koordinatensystem (fiktives KS) ein, in dem auch die Interpolation abläuft. Die Sollwerte werden an den Lageregler im Maschinenkoordinatensystem (reales KS) ausgegeben. Zwischen diesen beiden Koordinatensystemen wirkt die Transformation.

Durch die Funktion TRANSMIT lassen sich Fräsbewegungen auf Drehmaschinen ausführen. Bewegungen, die im realen Koordinatensystem nur schwer zu programmieren sind, können durch relativ einfache Bewegungssätze im fiktiven Koordinatensystem realisiert werden.

Initialisierung der Koordinatensysteme:

Die Nullpunkte von fiktivem und realem KS stimmen überein.



An- und Abwahl von TRANSMIT

Anwahl/Abwahl von TRANSMIT im "AUTOMATIC"-Betrieb oder "MDI-AUTOMATIC"-Betrieb:

AUTOMATIC-Betrieb:

Anwahl von TRANSMIT (G131)

$$X = X1$$

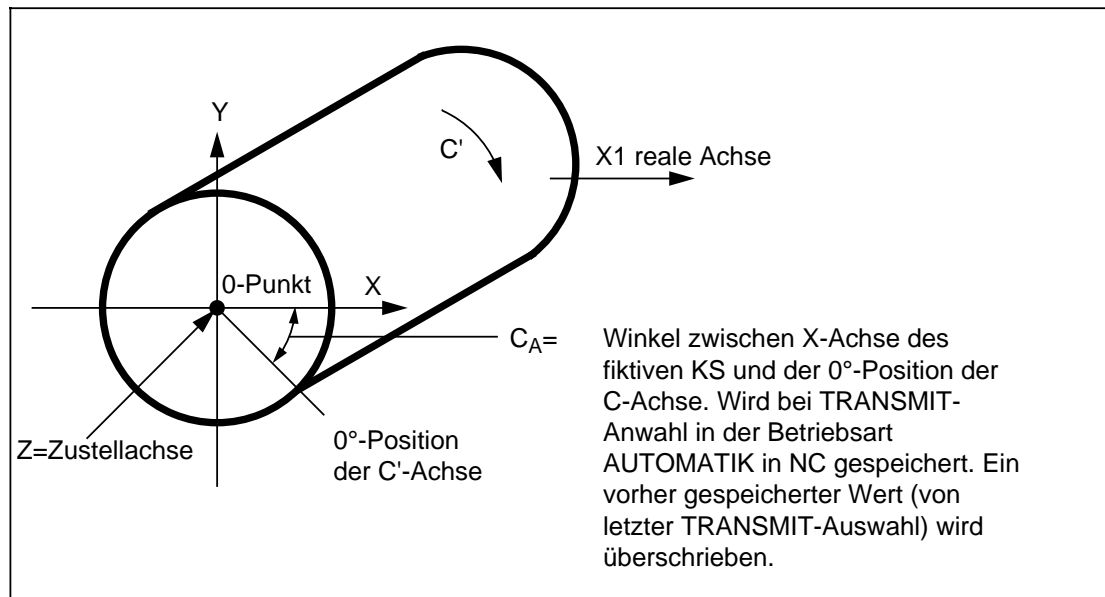
$$Y = 0$$

Abwahl von TRANSMIT (G130)

$$X = 0$$

$$Y = 0$$

Wird in einem Teileprogramm TRANSMIT mehrfach an- und abgewählt, so muß bei Bearbeitung des gleichen Werkstückteils die Rundachse zwischendurch referiert und geeignet positioniert werden.



MDI-AUTOMATIC-Betrieb:

Anwahl von TRANSMIT (G131)

$$X = X1 \cos (C_M - C_A)$$

$$Y = X1 \sin (C_M - C_A)$$

Abwahl von TRANSMIT (G130)

$X = 0$ C_M : momentaner Winkel der Rundachse
 $Y = 0$ C_A : Winkel der Rundachse bei der letzten Anwahl der Transformation im Automatik-Betrieb.

Bei Anwahl von TRANSMIT im "MDI-AUTOMATIC"-Betrieb wird der Winkel C_A der Rundachse von der letzten TRANSMIT- Bearbeitung im "AUTOMATIC"- Betrieb verwendet.

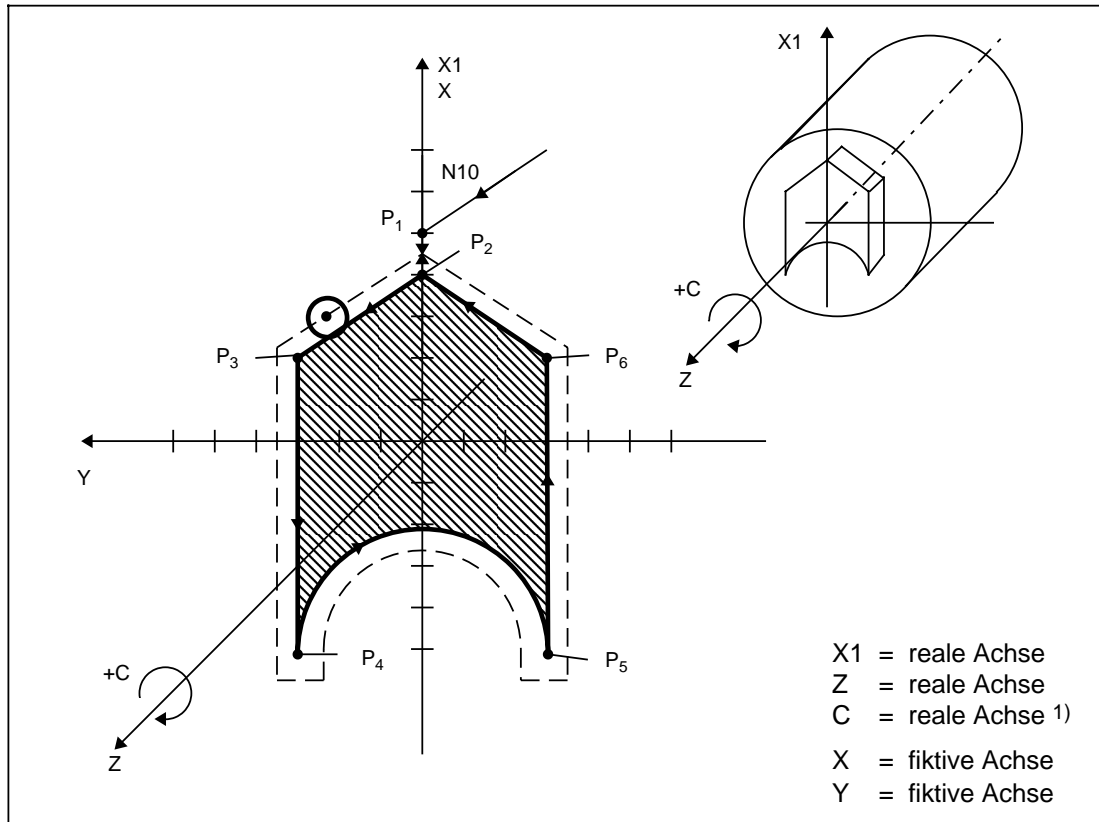
Beispiel: Bearbeitungskombination beim Fräsen

Fiktives KS: X,Y
 Reales KS: X1,C',Z

Progr. Achse	Gesteuerte Achse	
	nach Anwahl von TRANSMIT	nach Abwahl
1-dimensional X Y X1 C' Z	X1 + C' X1 + C' - - Z	- - X1 C' Z
2-dimensional X,Y X,Z Y,Z C',X1 C',Z X1,Z	X1 + C' X1 + C',Z X1 + C',Z - - -	- - - C',X1 C',Z X1,Z
3-dimensional X,Y,Z X1,C',Z	X1 + C',Z -	- X1,C',Z

Die mit "-" gekennzeichneten Kombinationen führen zu Bearbeitungsstop und gleichzeitig zu einer Alarmmeldung (Alarm 2043 "Programmierfehler bei Transformation").

Programm-Beispiel: Stirnflächenbearbeitung mit TRANSMIT



Stirnflächenbearbeitung mit TRANSMIT

```

%100 LF
N15 G01 F5000 X1=50 Z0 LF (P1) Anfahren des Startpunktes im realen System
N20 G131 LF (P2) Anwahl der Transformation TRANSMIT
N25 G01 G94 F1000 X40 Y0 G42 D1 LF (P2) Anwahl der FRK/SRK
N30 X20 Y30 LF (P3)
N35 X-50 LF (P4)
N40 G02 X-50 Y-30 J-30 I0 LF (P5)
N45 G01 X20 Y30 LF (P6)
N50 X40 Y0 LF (P2)
N55 G40 X50 LF (P1) Abwahl der FRK/SRK
N60 G130 LF (P2) Abwahl der Transformation TRANSMIT
N65 G00 X1=60 LF (P1) Reales System freifahren
N70 M02 LF
    
```

1) Da C nicht eingegeben werden kann, wird C benutzt.

3.2.13.4 Bearbeitungsgenauigkeit bei TRANSMIT

Die Bearbeitungsgenauigkeit bei TRANSMIT ist bestimmt durch

- den für die fiktiven Achsen programmierten Vorschub
- die Geschwindigkeit der realen Achsen, d.h. die Bearbeitungsgeschwindigkeit
- die eingesetzten Meßsysteme (Geber sw.).

Programmierter Vorschub, Bearbeitungsgeschwindigkeit:

Die Größe des Vorschubs ist gleichbedeutend mit der Bahngeschwindigkeit der programmierten Bewegung in der fiktiven Ebene X,Y. Die Geschwindigkeit der realen Achsen, d.h. die Bearbeitungsgeschwindigkeit, ändert sich dagegen ständig (siehe Kap. 3.2.12.5). Es wirken Beschleunigungen, so daß dadurch nicht die Formtreue bzw. Bearbeitungsgenauigkeit einer reinen Fräsmaschine garantiert werden kann.

Eingesetztes Meßsystem:

Die Bearbeitungsgenauigkeit in der translatorischen Achse (X1-Achse) ist abhängig von der Auflösung des entsprechenden Gebers. Der Einfluß der Rundachse (C-Achse) auf die Bearbeitungsgenauigkeit ist abhängig von der Auflösung des Winkelgebers und wird mit einer "Faustformel" angegeben:

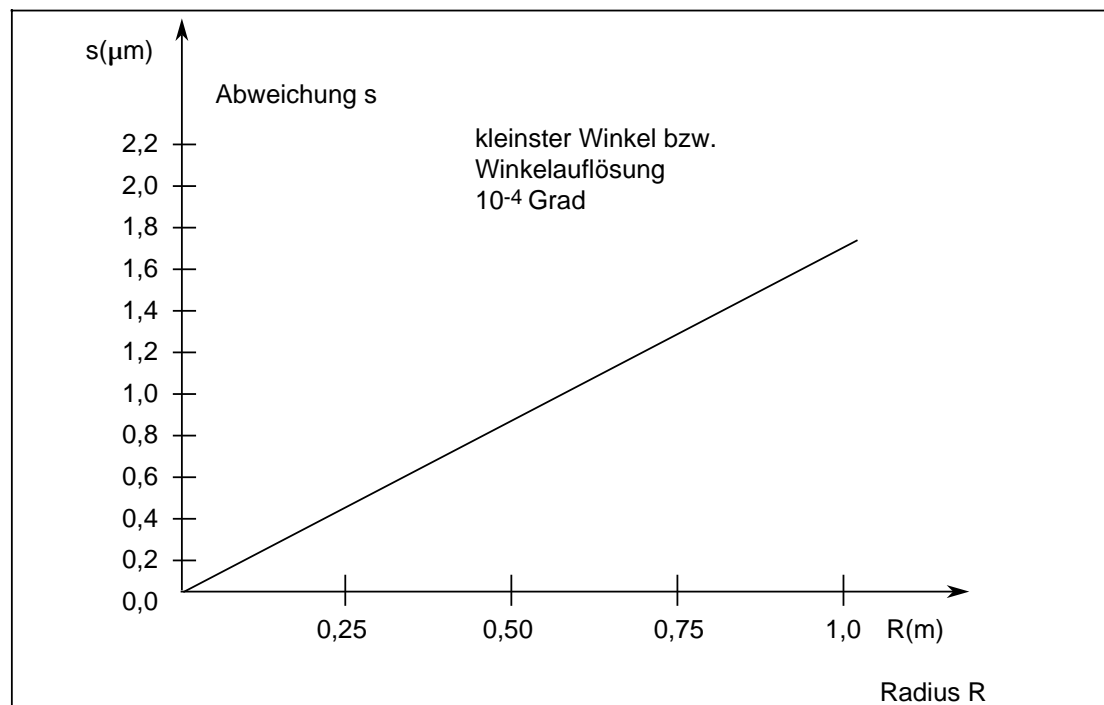
Die lineare, reale Abweichung s beträgt danach höchstens:

$$s = \cdot \cdot R / 180$$

mit $R = \sqrt{X^2 + Y^2}$: Abstand der programmierten Position (Radius) vom fiktiven KS-Nullpunkt

: Kleinster, einstellbarer Winkelwert in Grad, d.h. die Auflösung des Winkelgebers.

Der Einfluß der Winkelauflösung auf die zu erwartenden Abweichungen s bei verschiedenen Radien R zeigen die folgende Abbildung und die nachfolgende Tabelle. Die realen Abweichungen sind jedoch meistens kleiner als die berechneten.



Abweichung s an der Oberfläche einer Rundachse bei der Winkelauflösung von 10^{-4} Grad

Abweichung /10 ⁻⁴ mm	Radiuswert R /10 ⁻³ m	Abweichung /10 ⁻⁴ mm	Radiuswert R /10 ⁻³ m
1	57.29747	10	572.9747
2	114.5949	11	630.2721
3	171.8924	12	687.5695
4	229.1899	13	744.8670
5	286.4873	14	802.1645
6	343.7848	15	859.4618
7	401.0827	16	916.7592
8	458.3798	17	974.0567
9	515.6772	18	1031.354

Abweichung *s* bei einer Winkelauflösung von 10⁻⁴ Grad

3.2.13.5 Geschwindigkeitsüberwachung bei TRANSMIT

In der Betriebsart "AUTOMATIC" wird die programmierte Geschwindigkeit, d.h. die Geschwindigkeit im fiktiven KS, überwacht.

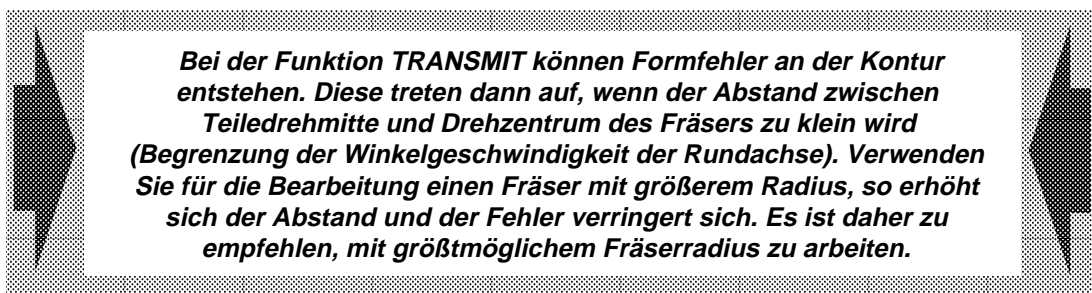
Damit die Rundachse nicht schneller dreht als im MD 280* definiert, wird der Vorschub gegebenenfalls satzweise reduziert. In diesem Fall wird der satzbezogene Alarm 3083 "Vorschub-begr. fiktive Achse" gemeldet. Die Geschwindigkeit der anderen realen Achse ist ebenfalls eine Funktion der programmierten Kontur und deshalb nicht konstant.

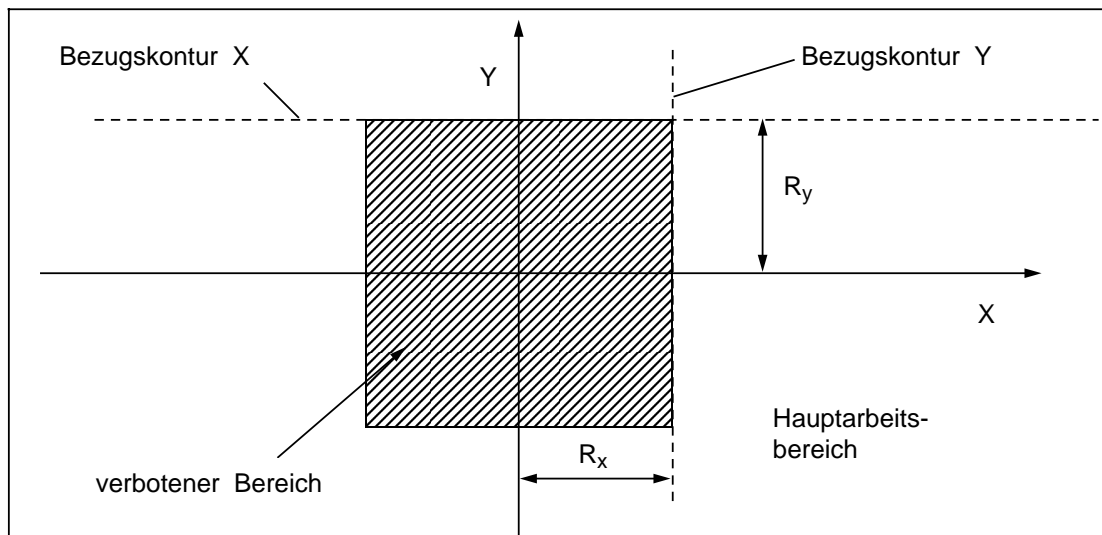
Arbeitsbereich für das fiktive Koordinatensystem:

Das fiktive Koordinatensystem ist unterteilt in einen **Hauptarbeitsbereich** und einen **verbotenen Bereich**. Gelangen die fiktiven Achsen in den **verbotenen Bereich**, so spricht die Überwachungseinrichtung der realen Achsen an. Kontrollierte Bewegungen der fiktiven Achsen im verbotenen Bereich sind jedoch über den Override-Schalter möglich. Mit dem Override-Schalter läßt sich auch das Ansprechen der Überwachung beeinflussen.

Die Winkelgeschwindigkeit der Rundachse wird überwacht und bei Überschreitung des Maximalwertes (Maschinendatum) auf diesen begrenzt. Gleichzeitig erscheint der Alarm 2035 "Vorschubbegrenzung" und der Alarm 3083 "Vorschubbegr. fiktive Achse" (erscheint nicht bei JOG, INC, REPOS). Diese Alarmer müssen über RESET (Alarm 2035) bzw. die Quittierungstaste (Alarm 3083) quittiert werden. Ein NC-Stop wird nicht ausgelöst.

Durch die Begrenzung kommt es bei Auftreten des Alarms 2035 zu Formfehlern an der Kontur.





Unterteilung des Koordinatensystems

Die Grenze zwischen **Hauptarbeitsbereich** und **verbotenem Bereich** bildet die Bezugskontur.

Die Daten für die Bereiche erfahren Sie vom Werkzeugmaschinen-Hersteller.

Die maximal möglichen Achsgeschwindigkeiten entnehmen Sie bitte ebenfalls den Hinweisen des Werkzeugmaschinen-Herstellers.

Eine Bahn, die durch die Drehmitte verläuft, kann nicht abgefahren werden. Es würden dabei Winkelgeschwindigkeiten um bis zu 180° pro Abtastintervall auftreten und damit die max. Rundachsengeschwindigkeit überschritten werden.

Die Rundachse müßte in der Drehmitte um 180° gedreht werden. Um dies zu vermeiden müssen derartige Konturen in mindestens drei Sätze (2. bis 4. Satz) zerlegt werden:

- | | |
|---------|---------------------------------------|
| 1.Satz: | Kontur bis zur Drehmitte. |
| 2.Satz: | Freifahren des Fräasers. |
| 3.Satz: | Drehen der Rundachse um 180° . |
| 4.Satz: | Zustellen des Fräasers. |
| 5.Satz: | Kontur nach der Drehmitte. |

3.3 Referenzpunktfahren über Teileprogramm (G74)

3.3.1 Funktionsbeschreibung

Mit der Funktion "Referenzpunkt-Synchronisieren per Teileprogramm" erhält der Anwender die Möglichkeit, aus einem Teileprogramm den Referenzpunkt einer realen, programmierten NC-Achse über eine G-Funktion anzufahren. Dabei werden die Maschinenbeeinflussung (Regler- und Fahrverhalten) sowie die Kommunikations-Oberfläche der CNC (Istwert-, Service-Anzeige) unverändert genutzt. Die Funktion "REF per PP" stellt eine Standardfunktion dar, die für Rund- und Linearachsen nutzbar ist.

3.3.2 Starten der Funktion

Aktivierungsmöglichkeiten für die Funktion "REF per PP" bestehen in den NC-Automatik-Betriebsarten "AUTOMATIC" und "MDI-AUTOMATIC". Syntax der Funktion im Teileprogramm:

N <Nr> G74 <Achse> LF

Erläuterungen:

<Nr> = Satznummer
<Achse> = Adresse einer generierten NC-Achse

Beispiel:

N1000 G74 C LF

Bei erweitertem Adressnamen einer NC-Achse, z.B. Y1 gilt:

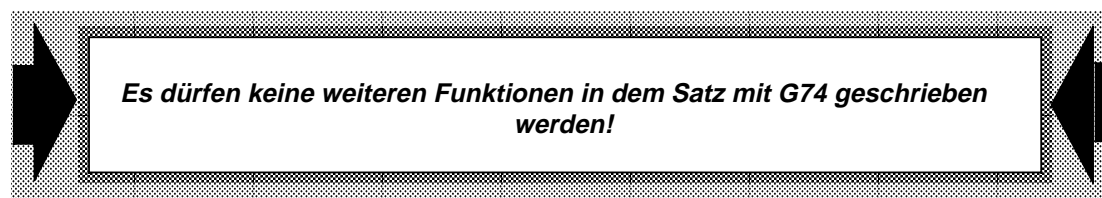
N <Nr> G74 Y1=LF

Beispiel:

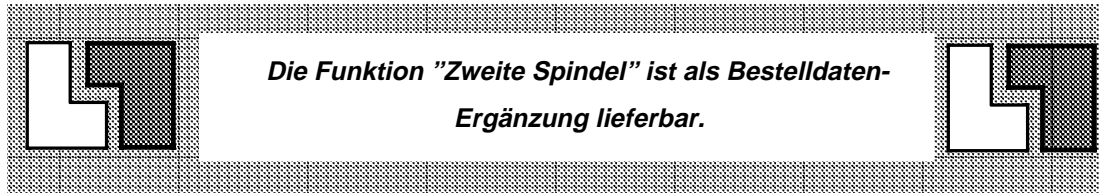
N1000 G74 Y1=LF

Bemerkungen:

1. Für die Steuerung 810/820 T/M gilt die Beschränkung auf eine NC-Achse pro G74-Satz.
2. Es können Rund- bzw. Linear-Achsen angegeben werden.



3.4 Zweite Spindel



Die zweite Spindel kann als Haupt- oder Hilfsspindel eingesetzt werden, z. B. als angetriebenes Werkzeug für die Funktion TRANSMIT.

Bei zwei Spindeln muß für jeden Kanal (1 bzw. 2) eine Spindel als Leitspindel definiert werden. Die entsprechende Festlegung erfahren Sie vom Maschinenhersteller. Im Betriebsarten-Grundbild werden Soll- und Istwerte für die jeweils gültige Leitspindel angezeigt.

Folgende G-Funktionen sind der Leitspindel zugeordnet:

- G33, G34, G35 Gewindebohren, Gewindeschneiden
- G04 Sn= Verweilzeit auf Spindelumdrehungen bezogen, wobei unter "n" die Leitspindelnummer stehen muß
- G95 Umdrehungsvorschub
- G96 Sn= Konstante Schnittgeschwindigkeit, wobei unter "n" die Leitspindelnummer stehen muß
- G97 Erweiterte Spindeldrehzahl von G96 beibehalten
- G63 Gewinde ohne Geber

M- und S-Funktionen können mit und ohne Adreßerweiterung programmiert werden (Ausnahme bei G04, G96). Bei Programmierung ohne Adreßerweiterung beziehen sich die programmierten M- und S-Funktionen auf die Leitspindel. Die Adreßerweiterung wird direkt hinter dem Adreßbuchstaben S oder M, dahinter "=" und anschließend der Hilfsfunktionswert programmiert. In einem Teileprogrammsatz dürfen nur für eine Spindel Spindelfunktionen programmiert werden.

Programmieren von Settingdaten für die Spindel

SD 4010	(Spindel 1)	Spindeldrehzahlbegrenzung für G92
SD 4011	(Spindel 2)	Spindeldrehzahlbegrenzung für G92
SD 4020	(Spindel 1)	Orientierter Spindelhalt (M19)
SD 4021	(Spindel 2)	Orientierter Spindelhalt (M19)
SD 4030	(Spindel 1)	Spindrehzahlbegrenzung
SD 4031	(Spindel 2)	Spindrehzahlbegrenzung
SD 4040	(Spindel 1)	Glättungskonstante bei Gewinde
SD 4041	(Spindel 2)	Glättungskonstante bei Gewinde

Mit G26 S1=500 wird z. B. die Spindeldrehzahlbegrenzung in das Settingdatum 4030 eingetragen.

Mit G92 S1=500 wird z.B. die Spindeldrehzahlbegrenzung für G92 in das Settingdatum 4010 eingetragen.

Mit M1=19 S1=180 wird zusätzlich zum orientierten Spindelhalt, z. B. die Soll-Position der Spindel in Grad, in das Settingdatum 4020 eingetragen.

Die Settingdaten 4040 und 4041 ersetzen das Settingdatum SD1.

3.5 Rundachse fliegend synchronisieren

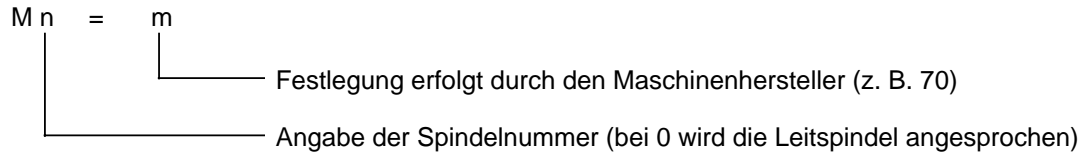
Die Funktionen

- Transmit für Drehmaschinen
- Gewindebohren ohne Ausgleichsfutter (interpolierend)
- Gewindeschneiden ohne Ausgleichsfutter (interpolierend)
- Zylinderinterpolation für Drehmaschinen

bedingen ein Umschalten vom Spindelbetrieb zum Rundachsenbetrieb und umgekehrt.

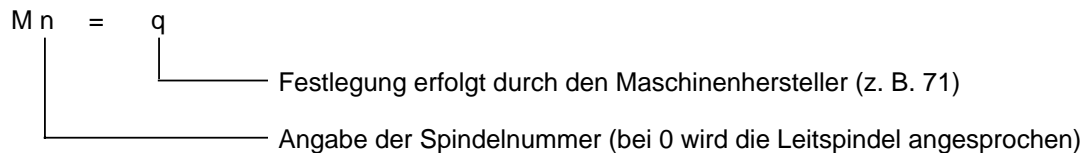
Mit der Funktion "Rundachse fliegend synchronisieren" wird die Synchronisation der Rundachse bereits beim Abbremsen der Spindel durchgeführt.

Das Umschalten vom Spindelbetrieb in den Rundachsbetrieb erfolgt über die Hilfsfunktion



Beispiel: M1=70

Das Umschalten vom Rundachsbetrieb in den Spindelbetrieb erfolgt über die Hilfsfunktion



Beispiel: M1=70

Hierbei wird intern auch der Befehl @714 ausgeführt.

Die Befehle zum Umschalten müssen in einem eigenen NC-Satz programmiert sein. Die Umschaltung kann im NC-Reset-Zustand auch durch Überspeichern erfolgen. Vor dem Satzvorlauf ist durch Überspeichern der Rundachsbetrieb an- bzw. abzuwählen.

Hinweise:

- Spindelbetrieb
Die Positionsanzeige für die Rundachse ist "0". Eine wiederholte Abwahl des Rundachsenbetriebs darf erfolgen.
Die Rundachse darf weder programmiert noch per Hand verfahren werden (sonst Alarm 1961).
- Rundachsbetrieb
Die Spindeldrehzahl bleibt "0". Eine wiederholte Anwahl des Rundachsenbetriebs darf programmiert werden.
Die Spindel darf nicht programmiert werden.
- Überspeichern des Umschaltens in einem unterbrochenen Programm ist nicht möglich.
- Je nach Vorgabe des Maschinenherstellers bleibt der Rundachsbetrieb erhalten oder wird mit RESET abgewählt.
- Das An- und Abwählen des Rundachsbetriebs muß im gleichen Kanal erfolgen.

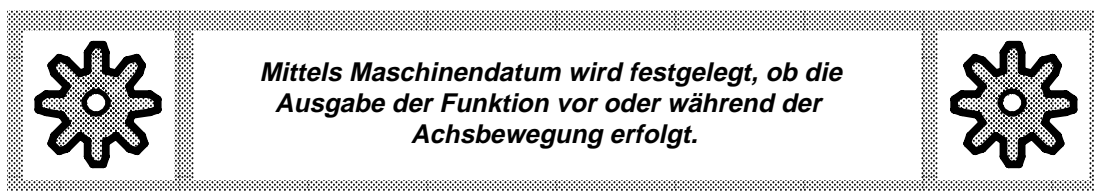
4 Zusatz-, Schalt- und Hilfsfunktionen

4.1 M, S, T, H

Die **Zusatz-, Schalt- und Hilfsfunktionen** enthalten vorwiegend technologische Angaben, soweit diese nicht in den dafür vorgesehenen Wörtern mit den Adreßbuchstaben F, S, T programmiert werden:

- Zusatzfunktion **M** (2 stellig)
- Spindeldrehzahl **S** (5 stellig)
- Werkzeugnummer **T** (4 stellig)
- Hilfsfunktion **H** (4 stellig).

Es können max. drei M-, eine S-, eine T- und eine H-Funktion in einem Satz stehen. Die Ausgabe an die Anpaßsteuerung geschieht in der **Reihenfolge**: M, S, T, H.



Angaben hierzu erfolgen vom Maschinenhersteller.

Bei Ausgabe der Funktionen während der Achsbewegung gilt:

Soll ein neuer Wert bereits vor dem Fahren der Achsen wirksam sein, so muß die neue Funktion im vorhergehenden Satz geschrieben werden.

4.2 Zusatzfunktionen M

Die Zusatzfunktionen sind zum Teil in DIN 66 025 Teil 2, zum Teil vom Werkzeugmaschinen-Hersteller festgelegt.

M-Funktionen können mit oder ohne erweiterte Adressierung programmiert werden, z. B. M** oder M* =*** steht für eine Zahl im Bereich von 0 bis 9

Daraus ergeben sich Zusatzfunktionen von M0 bis M99 und von M1=0 bis M9=99. M-Funktionen mit der erweiterten Adresse 0 werden wie Funktionen ohne erweiterter Adresse behandelt und standardmäßig ausdecodiert.

M00 oder M*=0 Programmierter Halt (unbedingt)

M00 ermöglicht es, das Programm zu unterbrechen, z. B. um eine Messung durchzuführen. Nach Abschluß der Messung kann das Programm durch Betätigen der Taste "NC-Start" wieder fortgesetzt werden. Die eingegebenen Informationen bleiben erhalten. Die Zusatzfunktion M00 wird in allen Automatikbetriebsarten wirksam. Ob auch der Spindelantrieb stillgesetzt wird, ist der speziellen Programmieranleitung der jeweiligen Maschine zu entnehmen. M00 ist auch in einem Satz ohne Weginformation wirksam.

M01 oder M*=1 Programmierter Halt (bedingt)

M01 wirkt wie M00, jedoch nur dann, wenn die Funktion "Bedingter Halt (M01) wirksam" über Softkey oder an der Nahtstelle eingeschaltet wurde. Die Anzeige von invertiertem Text ist bei M01 nicht möglich.

M02 oder M*=2 Programmende

M02 signalisiert Programmende mit Rücksetzen des Programms auf Programmanfang. Es wird im letzten Satz des Programms geschrieben. Die Steuerung wird in Grundstellung gebracht. M02 kann mit anderen Funktionen oder allein in einem Satz stehen. Der Einlesevorgang kann mit M02 gestoppt werden (Setting-Datum).

M17 oder M*=17 Unterprogrammende

M17 wird im letzten Satz eines Unterprogramms geschrieben. Es kann in diesem Satz allein oder zusammen mit anderen Funktionen stehen. Der Aufruf eines Unterprogramms und M17 dürfen **nicht** im gleichen Satz stehen (Schachtelung), da sonst das Unterprogramm nicht mehr ausgeführt wird.

M30 oder M*=30 Programmende

M30 wirkt wie M02.

M36, M37 oder M*=36, M*=37 Vorschubuntersetzung 1:100

M36 Der programmierte Vorschub ist wieder gültig.
 Abwahl von M37

M37 Der programmierte Vorschub wird um 1:100 untersetzt. Wirksam bei G94, G95 und G96. Gilt für Bahnvorschub und alle Simulationsvorschübe.

Beispiel:

N10	G94	X30	F1000	L _F	Vorschub 1000 mm/min wirkt
:					
N30	M37	L _F			Vorschub 10 mm/min wirkt
:					
N50	M36	L _F			Vorschub 1000 mm/min wirkt

M1=3, M2=3, M1=4, M2=4, M1=5, M2=5, M1=19, M2=19 Hauptspindelsteuerung

(M19 nur mit Pulsgeber an der Spindel)

Die erweiterte Adreßschreibweise stellt den Bezug zur Spindel 1 oder 2 her.



Bei Ausführung mit analoger Spindeldrehzahlausgabe sind folgende M-Wörter zur Spindelsteuerung festgelegt:

M1=3, M2=3	Spindeldrehrichtung rechts	Spindel 1, Spindel 2
M1=4, M2=4	Spindeldrehrichtung links	Spindel 1, Spindel 2
M1=5, M2=5	Spindelhalt, ohne Orientierung	Spindel 1, Spindel 2
M1=19, M2=19	Orientierter Spindelhalt	Spindel 1, Spindel 2
M0=...	gilt für die Leitspindel	

Mit M19 S.. ist es möglich, die Hauptspindel orientiert anzuhalten. Die Programmierung des entsprechenden Winkels erfolgt unter S in Grad. Der Winkel wird von der Nullmarke in Drehrichtung rechts gemessen. M19 S.. muß in einem eigenen Programmsatz stehen.

Der unter der Adresse S programmierte Winkel ist modal wirksam. Wird M19 ohne S programmiert, so ist der unter S gespeicherte Wert für den Winkel wirksam, d.h. für wiederholtes Stillsetzen genügt die Programmierung von M19.

Der Winkelwert kann auch über die Bedientafel unter "Settingdaten Spindel" eingegeben werden. Über ein MASCHINENDATUM kann festgelegt werden, ob die Spindel zum Stillstand gekommen sein muß, bevor die im nächsten Satz programmierte Achsbewegung gestartet wird, oder ob der nächste Satz noch während der Spindelpositionierung freigegeben wird.

M19 wählt M03 bzw. M04 nicht ab.

Freie Zusatzfunktionen

Alle Zusatzfunktionen **außer** M00, M01, M02, M03, M04, M05, M17, M19, M30, M36 und M37 sind frei verfügbar.

Für alle freien Zusatzfunktionen gilt die erweiterte Adreßschreibweise mit Angabe der Kanalnummer:

Beispiel:

M3=124

3. . . Angabe der NC-Kanalnummer

124. . . M-Funktion 124.

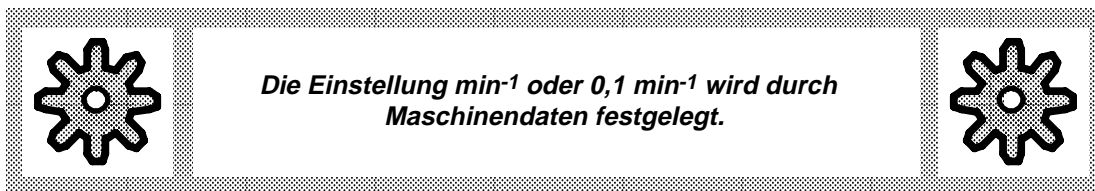
Genauere Auskunft über die Verwendung der einzelnen Funktionen gibt der spezielle **Programmschlüssel** der Maschine (siehe Kap.12). Ein Teil der Bedeutung dieser Funktion ist in DIN 66025 festgelegt.

4.3 Spindelfunktion S

Unter der Adresse S kann wahlweise eingegeben werden:

- Spindeldrehzahl in min^{-1} oder $0,1 \text{ min}^{-1}$ *)
- Schnittgeschwindigkeit in m/min oder $0,1\text{m}/\text{min}$ *)
- Spindeldrehzahlbegrenzung in min^{-1} oder $0,1 \text{ min}^{-1}$ *)
- Spindelhalt in Grad (M19)
- Verweilzeit in Spindelumdrehungen (siehe G04).

*) Drehzahl und Schnittgeschwindigkeit sind in der gleichen Eingabeeinheit zu programmieren.



Für das S-Wort gilt die **erweiterte Adreßschreibweise** mit Angabe der Spindelnummer:

Beispiel:

S2=1000

2... Angabe der Spindelnummer

1000... Spindeldrehzahl (U/min) bzw. (1/min).

Für nicht vorhandene Spindeln kann das S-Wort auch als Hilfsfunktion genutzt werden.

S3=... S9=12345 (max. 5 Stellen für den Wert)



Bei einer Spindel: 2, 3, ..., 9

Bei zwei Spindeln: 3, 4, ..., 9

4.4 Hilfsfunktionen H

Für Schaltfunktionen an der Maschine oder Bewegungen, die nicht der numerischen Kontrolle unterliegen, steht mit der Adresse H eine Hilfsfunktion je Satz zur Verfügung. H kann mit 4 Dekaden programmiert werden. Die Bedeutung der Funktionen ist aus der Programmieranleitung des Werkzeugmaschinen-Herstellers zu ersehen.

Für das H-Wort gilt die erweiterte Adreßschreibweise:

H..=...

4.5 Werkzeugnummer T

Die Werkzeugnummer bestimmt das für einen Bearbeitungsabschnitt benötigte Werkzeug.

Beispiel:

T1234

T Adresse

1234 Werkzeugnummer (max. 4-dekadig).

Für das T-Wort gilt die **erweiterte Adreßschreibweise** mit Angabe der Kanal-Nr.:

T1=1234 Anwahl der Werkzeug-Nr. 1234 für die Bearbeitung im Kanal 1.

5 Unterprogramme

5.1 Anwendung

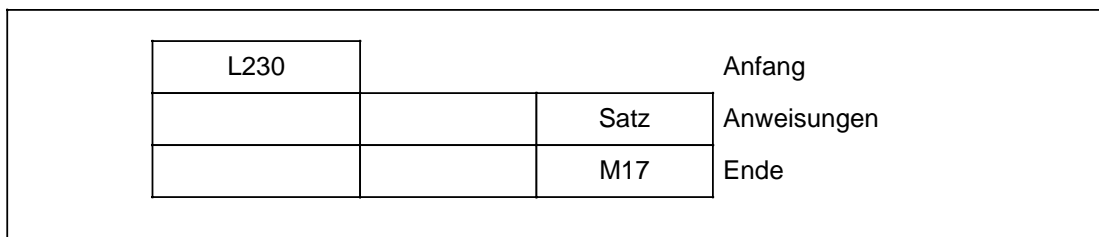
Kommt bei der Bearbeitung eines Werkstücks mehrmals der gleiche Arbeitsgang vor, so kann dieser als Unterprogramm eingegeben und beliebig oft im Teileprogramm oder über Handeingabe aufgerufen werden.

Im **Programmspeicher** können standardmäßig **200** Teile- und Unterprogramme gleichzeitig abgelegt werden. Unterprogramme werden vorzugsweise im **Kettenmaß** programmiert. Im Teileprogramm wird vor dem Aufruf des Unterprogramms das Werkzeug auf den Ausgangspunkt gestellt. Der Bearbeitungsablauf am Werkstück kann dann ohne Änderung der Bemaßung im Unterprogramm an verschiedenen Stellen des Werkstücks wiederholt werden.

5.2 Unterprogramm-Aufbau

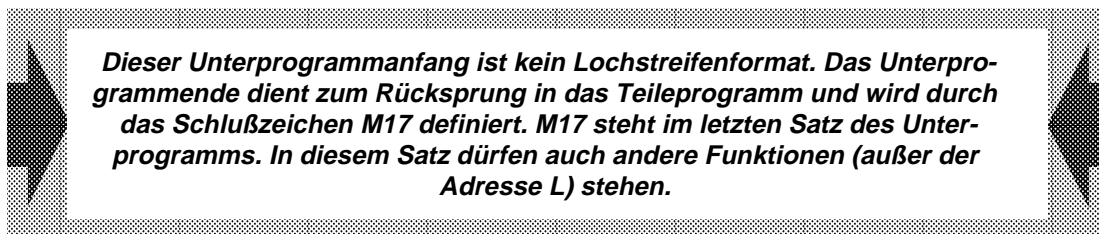
Ein Unterprogramm besteht aus

- dem Unterprogrammanfang
- den Sätzen des Unterprogramms und
- dem Unterprogrammende.



Unterprogramm-Aufbau

Den **Unterprogrammanfang** bildet die **Adresse L** und die max. 4-stellige Unterprogrammnummer (siehe Programmschlüssel, Kap.12).



5.3 Unterprogramm-Aufruf

Der Unterprogramm-Aufruf erfolgt in einem Teileprogramm über die Adresse **L** mit der Unterprogrammnummer und der **Durchlaufzahl** mit der Adresse **P**. Wird eine Unterprogrammnummer ohne die Adresse **P** programmiert, wird automatisch eine Durchlaufzahl von **P1** (1 Durchlauf) angenommen.

Beispiel:

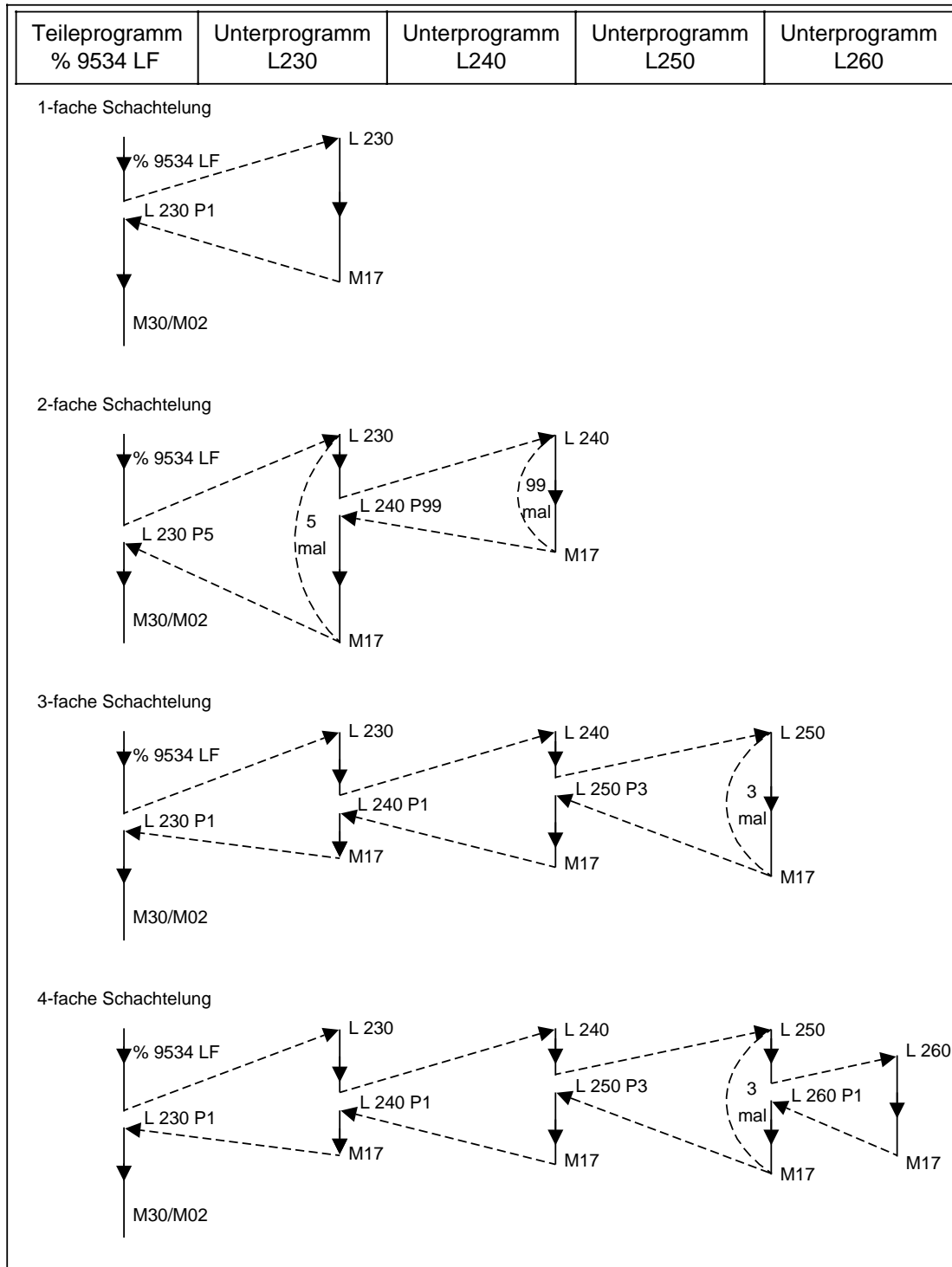
L123 P1 L_P	
L123	Unterprogrammnummer (1...9999)
P1	Anzahl der Durchläufe (1...99)

Bei der Programmierung ist folgendes zu beachten:

- Der Unterprogramm-Aufruf darf nicht zusammen mit **M02**, **M30** oder **M17** in einem Satz stehen, da sonst das Unterprogramm nicht ausgeführt wird.
- Erfolgt der Unterprogramm-Aufruf bei angewählter Schneidenradius-Bahnkorrektur, sind die **SRK-Sonderfälle** "Sätze ohne Wegadressen" zu beachten.
- Steht der Unterprogramm-Aufruf zusammen mit anderen Funktionen in einem Satz, so wird das Unterprogramm am Ende des Satzes aufgerufen.
- Die Sonderfunktionen **L81** bis **L89** können über die **G-Funktionen** **G81** bis **G89** aufgerufen werden. **G80** hebt **G81** bis **G89** auf (Löschstellung).
G81 bis **G89** und **L81** bis **L89** in einem Satz löst den Alarm 3006 "Falsche Satzstruktur" aus.
- **G81** bis **G89** sind modal wirksam.

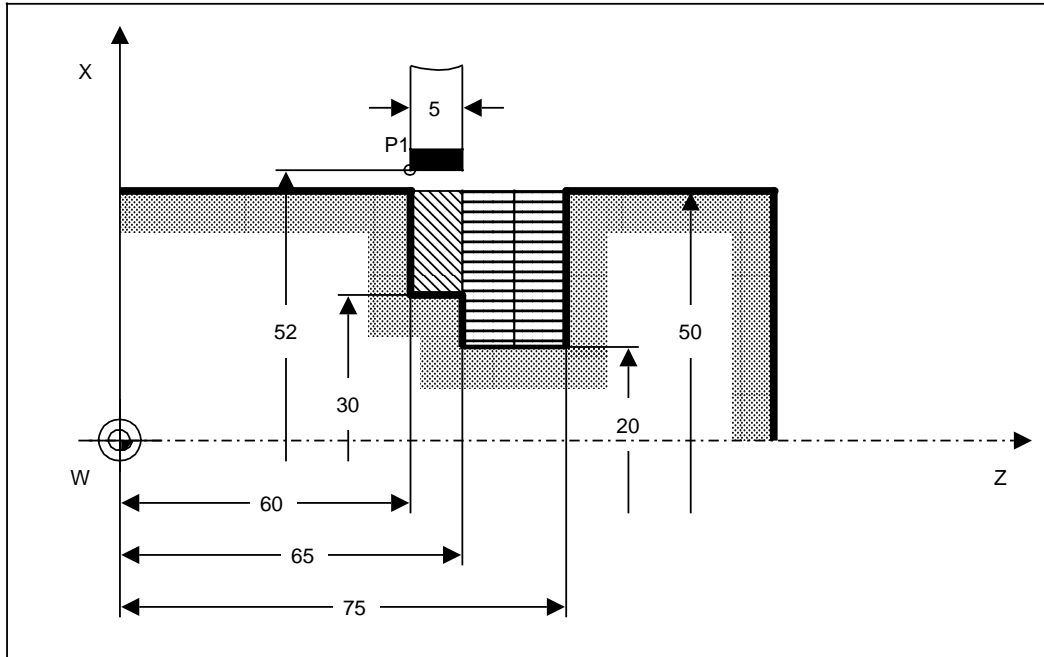
5.4 Unterprogramm-Schachtelung

Unterprogramme können nicht nur in einem Teileprogramm, sondern auch in anderen Unterprogrammen aufgerufen werden. Hier spricht man von Unterprogramm-Schachtelung. Es ist maximal eine **4-fache** Unterprogramm-Schachtelung möglich.



Unterprogramm-Aufruf und Unterprogramm-Schachtelung

Beispiel: 2-fache Unterprogramm-Schachtelung




Unterprogramm-Schachtelung

Programmausführung:


```
% 4011 L_F
N5 G90 G94 F500 S2000 M3 T3 L_F
N10 G00 X52 Z60 L_F
N15 L230 P1 L_F
.
.
N90 M30 L_F
```

Programmkopf % 4011 (Hauptprogramm)
 Anwahl Bezugsmaß, Vorschub mm/min., S-, D-, T-, M-Funktion
 Fahre im Eilgang auf P1
 Aufruf Unterprogramm L230 mit 1 Durchlauf
 Programmende

```
L230
N5 G91 G01 X-11 L_F
N10 G09 X11 L_F
N15 L240 P2 L_F
N20 M17 L_F
```

Programmkopf L230 (Unterprogramm)
 Programmteil zum Abdrehen des Segments 
 Programmteil zum Abdrehen des Segments
 Aufruf Unterprogramm L240 mit 2 Durchläufen
 Unterprogrammende vom L230

```
L240
N5 G91 G00 Z5 L_F
N10 G01 G09 X-16 L_F
N15 G00 X16 L_F
N20 M17 L_F
```

Programmkopf L240 (Unterprogramm)
 Programmteil zum Abdrehen des Segments 
 Programmteil zum Abdrehen des Segments
 Programmteil zum Abdrehen des Segments
 Unterprogrammende vom L240

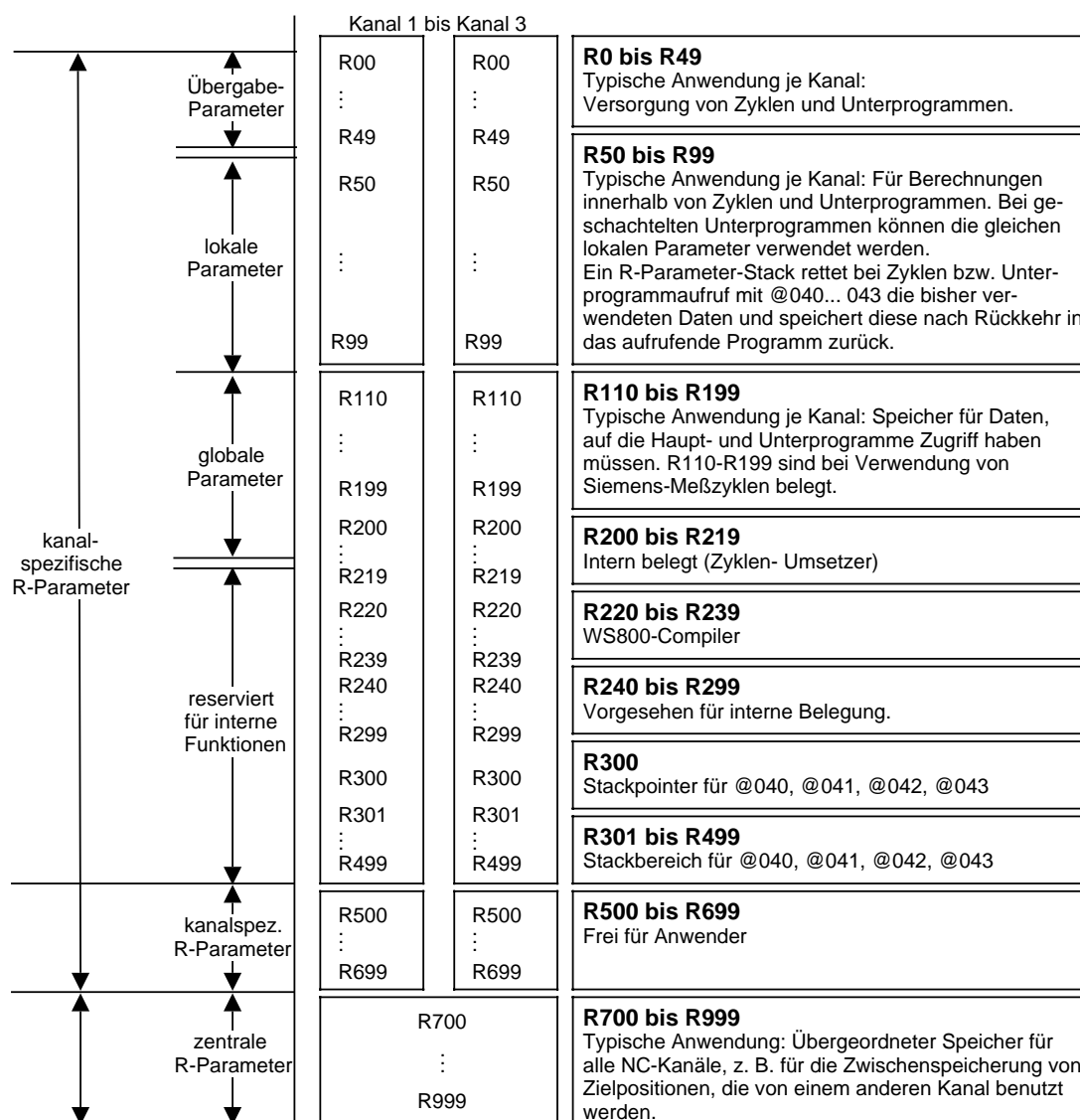
6 Parameter

6.1 Parameter-Programmierung

Parameter stehen in einem Programm stellvertretend für einen Zahlenwert einer Adresse.

Man weist ihnen innerhalb des Programms Werte zu und kann damit ein Programm für mehrere ähnliche Einsatzzwecke passend machen (z. B. unterschiedlicher Vorschub, für verschiedene Werkstoffe, Arbeitszyklen).

Parameter bestehen aus der Adresse **R** und einer **maximal 3-stelligen Nummer**. Im Grundausbau stellt die Steuerung je Kanal **700 kanalspezifische** und insgesamt **300 zentrale Parameter** zur Verfügung, die in die Bereiche Übergabeparameter, Rechenparameter, kanalabhängig und kanalunabhängig deklarierte Parameter und zentrale Parameter unterteilt sind (siehe Sprachbeschreibung CL800).



Strukturierung und Anwendung von R-Parametern

Allen Adressen außer N kann anstelle eines Wertes ein Parameter zugeordnet werden.

N5 Z = R5 L_F

Dabei ist der Wortaufbau der einzelnen Adressen zu beachten (nicht bei allen Adressen gilt Dezimalschreibweise oder Vorzeicheneingabe)

Falsch:

R1= 51120, 98
H = - R1

Dieser Zahlenwert ist bei der Adresse H nicht möglich Alarm 3000 "Allgemeiner Programmierfehler".

6.2 Parameter-Definition

Mit der Parameter-Definition werden den einzelnen Parametern bestimmte **vorzeichenbehaftete** Zahlenwerte zugeordnet.

Die Definition der Parameter kann sowohl in **Teileprogrammen** als auch in **Unterprogrammen** erfolgen.

R1 = 10 L_F

Parameter-Definition, Unterprogramm-Aufruf und Schaltfunktionen dürfen in einem Satz stehen. Der für einen Parameter definierte Wert wird der Adresse direkt zugeordnet.

Beispiel:

% 5772 L_F

N5...

.

.

N35 R1=10 R29=-20.05 R5=50 L_F

Parameter-Definition

N40 L51 P2 L_F

Aufruf des Unterprogramms L51, zweimaliger Durchlauf

N45 M02 L_F

L51

N5 Z=-R5 B=-R1 L_F

N10 X=-R29 L_F

.

.

N50 M17 L_F

6.3 Parameter-Rechnung

Verknüpfung von Parametern

Mit Parametern kann in allen **vier Grundrechenarten** gerechnet werden. Allerdings ist hier die **Reihenfolge** der Verknüpfung für das Rechenergebnis maßgebend. Die Punkt-vor-Strich-Regel gilt hier nicht.

Rechenarten	Programmierte Rechenoperation
Definition	$R1 = 100$
Zuweisung	$R1 = R2$
Negation	$R1 = - R2$
Addieren	$R1 = R2+R3$
Subtrahieren	$R1 = R2 - R3$
Multiplizieren	$R1 = R2 \cdot R3$
Dividieren	$R1 = R2 / R3$

Das Ergebnis einer Rechenoperation steht im erstgenannten Parameter einer Verknüpfung, sein Ausgangswert wird also bei der Verknüpfung überschrieben und geht verloren. Der Wert des zweiten und/oder dritten Parameters bleibt erhalten.

Wertzuweisung unter Parametern

Soll ein Parameter den Wert eines anderen Parameters erhalten, so gilt:

$$R1 = R3 \text{ L}_F$$

Rechnung zwischen Zahlen und Parametern

- Addition und Subtraktion von Zahlen und Parametern in Verbindung mit Adressen.

Man kann einen Parameter zum Wert einer Adresse dazuzählen oder ihn davon abziehen. Rechenzeichen müssen dabei geschrieben werden. Kein Vorzeichen bedeutet positive Zahl.

Das Rechenzeichen "+" muß immer eingegeben werden.

$$X=10+R100 \text{ L}_F$$

oder

$$X=R100 + 10 \text{ L}_F$$

Beispiel:

N05 R1=9.7 R2 = - 2.1 L_F

N10 X=20.3 + R1 L_F

N15 Z=32.9 - R2 L_F

Ergebnis: x=30
z=35.

- Rechnung mit Zahlen und Parametern

Die Multiplikation, Division, Addition und Subtraktion von absoluten Zahlen und R-Parametern ist möglich.

$$R10=15+ R11 L_F$$

- Es können mehrere unabhängige Rechnungen in einem Satz programmiert werden.

$$R1 = R2+ 23 \quad R50 = R37\cdot3 \quad R99 = R27 / R13 L_F$$

6.4 Parameter-Reihe

$$R1 = R2+ R3 - R4\cdot R5 / R6..... L_F$$

Es sind alle 4 Grundrechenarten in beliebiger Reihenfolge erlaubt. Eine Parameter-Reihe wird durch die **Satzlänge** von **maximal 120 Zeichen** begrenzt.

Gerechnet wird in folgender Weise:

1. Rechenschritt R1 = R2
2. Rechenschritt R1 = R1 + R3
3. Rechenschritt R1 = R1 - R4
4. Rechenschritt R1 = R1 · R5
5. Rechenschritt R1 = R1 / R6

1. Rechenschritt R1 = R2
2. Rechenschritt R1 = R2 + R3
3. Rechenschritt R1 - R4
4. Rechenschritt R1·R5
5. Rechenschritt R1 / R6

R1

In der Parameter-Reihe sind anstelle eines Verknüpfungs-R-Parameters (nicht Ergebnis-Parameter) auch Konstanten und Pointer (Zeiger auf R-Parameter) mit der Adresse P erlaubt.

Beispiel:

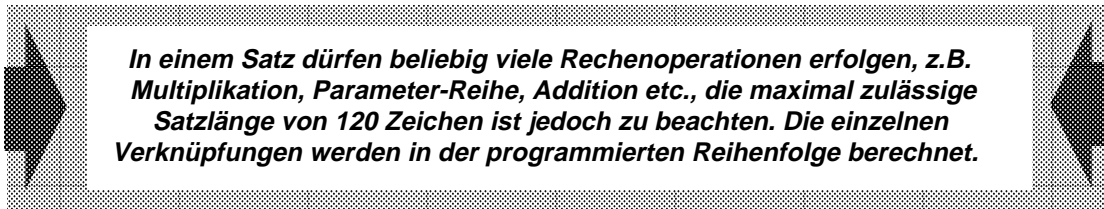
$$R1=R2+10.5 - P3 \cdot R5/R6\dots$$

Der Ergebnisparameter muß ein R-Parameter sein.

P3:

P Adresse des Pointers

3 Zeiger auf R-Parameter R3, d.h. der Inhalt von R3 ist die Adresse eines R-Parameters, dessen Wert in die Parameter-Reihe eingerechnet wird.



Wertebereich: kleinster Wert: $1 \cdot 10^{-8}$
 größter Wert: 99999999.
 Anzeige: Fließkomma (± 1.2345678) bis ($\pm 12345678.$)

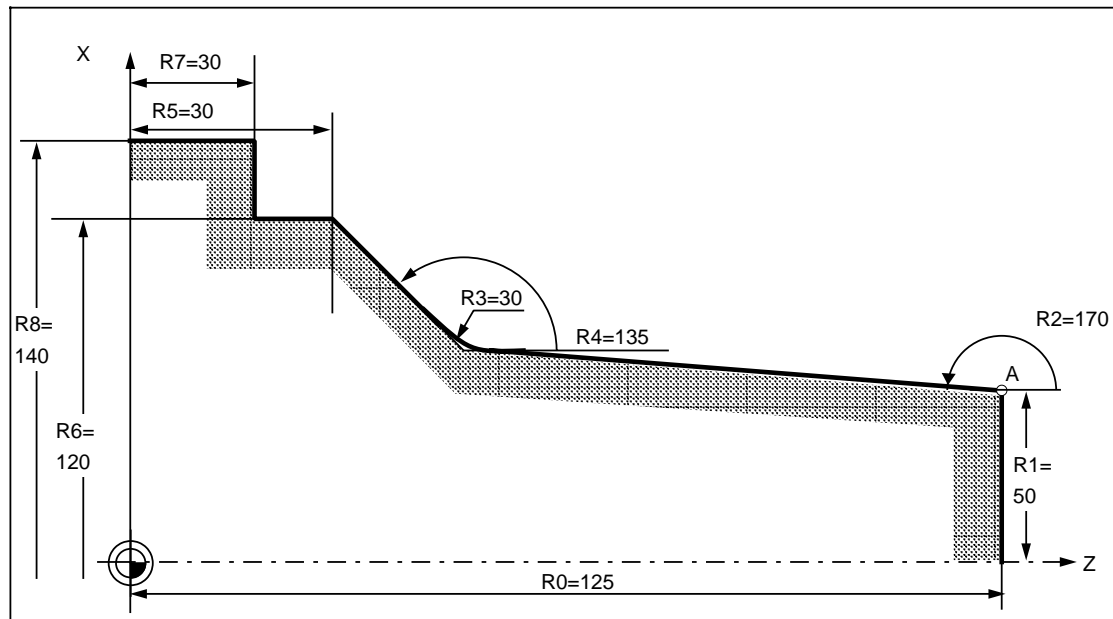
6.5 Programmbeispiel mit R-Parametern

Parameter für wertfreie Unterprogramme

Bei wertfreien Unterprogrammen erfolgt die Übernahme der aktuellen Daten mittels der Parameter **R00 bis R99**.

Diese Parameter verwendet man im Unterprogramm anstelle von Zahlenangaben. Je Satz dürfen **maximal 10 Parameter** programmiert werden.

Beim Aufruf des Unterprogrammes muß sichergestellt sein, daß die verwendeten Parameter die richtigen Werte besitzen. Die Wertzuweisung an die Parameter erfolgt im Hauptprogramm.

Beispiel:

Die abzuspandende Kontur wird in einem Unterprogramm abgelegt.

```
L47 LF
N5 G90 Z=R0 LF
N10 X=R1 LF
N15 A=R2 A=R4 X=R6 Z=R5 B=R3 LF
N20 Z=R7 LF
N25 X=R8 LF
N30 Z=R0 M17 LF
```

Unterprogramm-Aufruf:

```
N.. X.. Z.. LF
N.. L47 P1 R0=125 R1=50 R2=170 R3=30
R4=135 R5=30 R6=120 R7=20 R8=140 LF
```

7 Konturzug



7.1 Kontur-Kurzbeschreibung

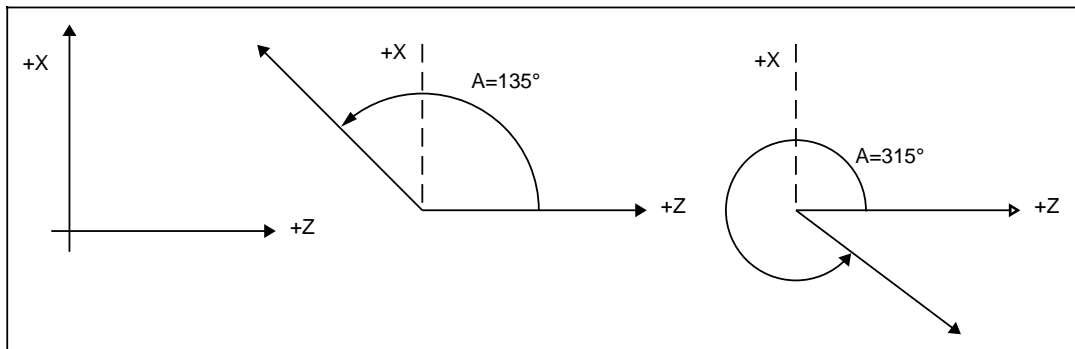
Zur **Konturbeschreibung** sind Mehrpunktezüge zur direkten **Programmierung** nach der Werkstückzeichnung vorgesehen. Schnittpunkte der Geradenzüge werden als Koordinatenwerte oder über Winkel angegeben.

Die einzelnen Geradenzüge können direkt in Form einer Ecke ineinander übergehen, über Radien abgerundet oder angefast werden. Fasen- und Übergangsradien werden nur mit ihrer Größe angegeben. Die **Geometrieberechnung** übernimmt die Steuerung. Die Programmierung der Endpunktkoordinaten kann im Bezugsmaß oder im Kettenmaß erfolgen.

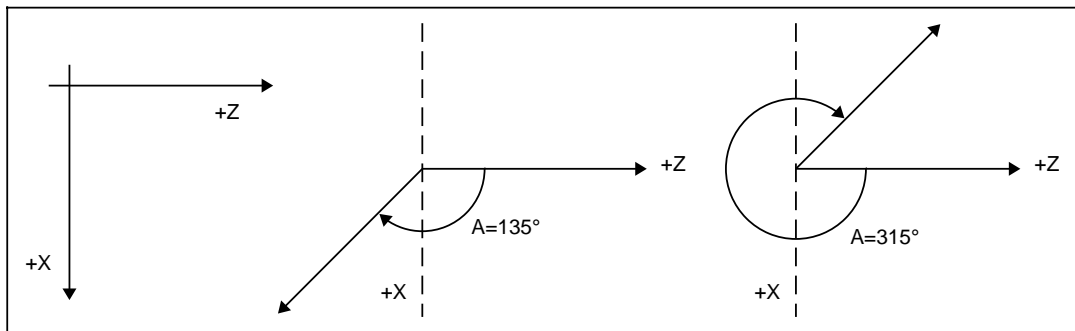
Winkel (A): Eingabefinheit 0.00001 entspricht 10^{-5} Grad.

Im rechtsdrehenden Koordinatensystem bezieht sich die Winkelangabe (max. 359.99999 Grad) immer von der horizontalen Achsrichtung auf die vertikale Achsrichtung.

Im Beispiel wurden die Achsadressen mit X, Z definiert.



Rechtssystem und Arbeitsbereich hinter der Drehmitte

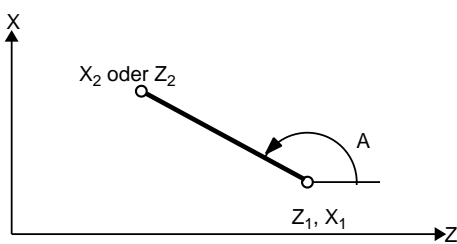
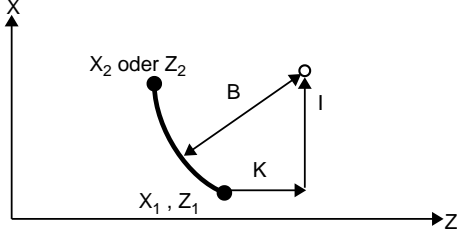
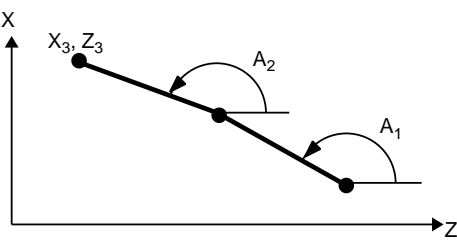


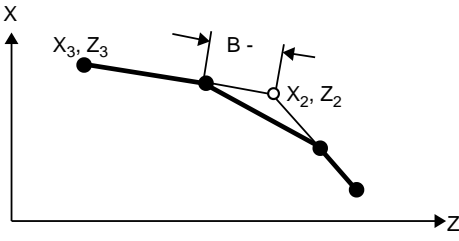
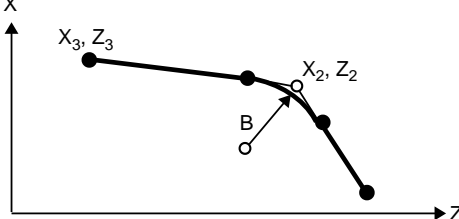
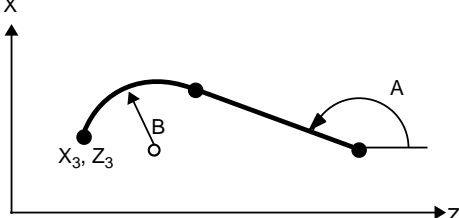
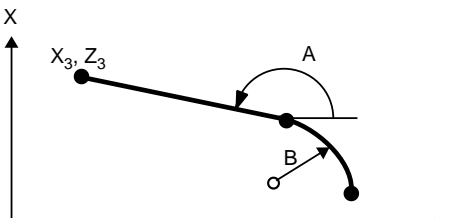
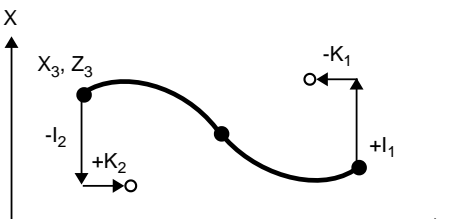
Rechtssystem und Arbeitsbereich vor der Drehmitte

7.2 Konturzug-Programmierung

Die dargestellten Elemente gelten für die Z-X-Ebene (G18) und eine Drehmaschine mit Arbeitsbereich hinter der Drehmitte.

Die Beispiele **1 bis 8** stellen die **Grundelemente** der Konturzugprogrammierung dar. Diese Grundelemente lassen sich vielfältig **kombinieren**. In der Steuerung sind die Adressen für Winkel (hier A) und Radius (hier B) frei wählbar. Die Adressen dürfen nicht mehrfach vergeben werden. Erweiterte Adressen für Winkel (A) und Radius/Fase (B,B-) sind nicht möglich.

Funktion	Programmierung	Beispiel
(1) 2-Punkte- Zug	N... A... X ₂ ... (oder Z ₂) LF Die zweite Endpunktkoordinate wird von der Steuerung berechnet.	
(2) Kreis- bogen	N...G02 (oder G03) I.. K.. B.. X ₂ .. (oder Z ₂) LF Der Kreisbogen ist auf einen Quadranten begrenzt. Die zweite Endpunktkoordinate wird von der Steuerung berechnet. Im Konturzug müssen beide Parameter I und K programmiert werden, auch wenn ein Wert Null ist.	
(3) 3-Punkte- Zug	N.. A ₁ .. A ₂ .. X ₃ .. Z ₃ .. LF Die Steuerung berechnet die Koordinaten des Stützpunktes und generiert 2 Sätze. Der Winkel A ₂ bezieht sich auf die zweite Gerade.	

Funktion	Programmierung	Beispiel
(4) Fase	<p>N... X₂... Z₂... B-... LF N... X₃... Z₃... LF 1)</p> <p>B- ... bedeutet eine Fase einfügen. B ... bedeutet einen Radius einfügen. (Das Zeichen „Minus“ hat hier nicht die Bedeutung als Vorzeichen, sondern ist die Sonderkennung für B als Fase.)</p>	
(5) Radius	<p>N... X₂ ... Z₂ ... B... N... X₃ ... Z₃ ... LF 1)</p> <p>Der eingefügte Radius darf nicht größer sein als die kleinere der beiden Strecken.</p>	
(6) Gerade- Kreisbogen (tangential)	<p>N.. G02 (oder G03) A.. B.. X₃.. Z₃.. LF</p> <p>Kreisbogen nicht über 180 Grad. Die Reihenfolge A (Winkel) und B (Radius) muß eingehalten werden.</p>	
(7) Kreisbogen -Gerade (tangential)	<p>N.. G02 (oder G03) B.. A.. X₃.. Z₃.. LF</p> <p>Kreisbogen nicht über 180 Grad. Die Reihenfolge B, A ist einzuhalten. Kein Radius in X₃, Z₃ einfügbar.</p>	
(8) Kreisbogen- Kreisbogen (tangential)	<p>N.. G02 (oder G03) I₁.. K₁.. I₂.. K₂.. X₃.. Z₃.. LF</p> <p>Die Wegbedingung wird für den ersten Kreisbogen programmiert. Die zweite Wegbedingung ist immer entgegengesetzt und wird nicht programmiert. Die Interpolationsparameter des zweiten Kreises sind auf den Endpunkt dieses Kreises bezogen. Es müssen beide Interpolationsparameter programmiert werden, auch wenn ein Wert Null ist.</p>	

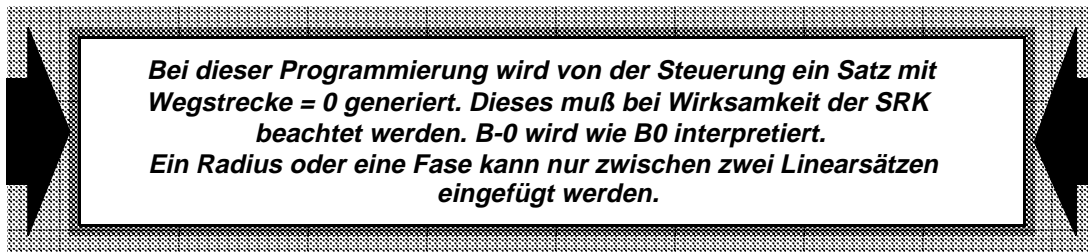
1) Zweiter Satz kann auch ein Konturzug sein

Funktion	Programmierung	Beispiel
(1) + (4) 2-Punkte- Zug + Fase	N... A.. X ₂ ... (oder Z ₂ ..) B... LF N... X ₃ .. Z ₃ ... LF 1)	
(1) + (5) 2-Punkte- Zug + Radius	N... A.. X ₂ ... (oder Z ₂ ...) B... LF N... X ₃ .. Z ₃ ... LF 1) Der eingefügte Radius darf nicht größer sein als die kleinere der beiden Strecken.	
(3) + (4) 3-Punkte- Zug + Fase	N... A ₁ .. A ₂ .. X ₃ .. Z ₃ .. B... LF	
(3) + (5) 3-Punkte- Zug + Radius	N... A ₁ .. A ₂ .. X ₃ .. Z ₃ .. B... LF	
(3) + (4) + (4) 3-Punkte- Zug + Fase + Fase	N... A ₁ .. A ₂ .. X ₃ ... Z ₃ .. B ₁ ... B ₂ ... LF 1) N... X ₄ .. Z ₄ .. LF 1) Anfügen einer zweiten Fase am Endpunkt X ₃ , Z ₃ .	

1) Zweiter Satz kann auch ein Konturzug sein

Funktion	Programmierung	Beispiel
(3) + (5) + (5) 3-Punkte- Zug + Radius + Radius	N... A ₁ .. A ₂ .. X ₃ .. Z ₃ .. B ₁ ... B ₂ .. LF N... X ₄ .. Z ₄ .. LF 1)	
(3) + (4) + (5) 3-Punkte- Zug + Fase + Radius	N... A ₁ .. A ₂ .. X ₃ .. Z ₃ .. B-... B.. LF N... X ₄ .. Z ₄ .. LF 1)	
(3) + (5) + (4) 3-Punkte- Zug + Radius + Fase	N... A ₁ .. A ₂ .. X ₃ .. Z ₃ .. B... B-.. LF N... X ₄ .. Z ₄ .. LF 1)	

Bei Ecken, in die **keine Fase oder Radius** eingefügt werden soll, ist **B0** zu programmieren, wenn im Konturzug noch ein **Radius oder eine Fase** folgt.



Die Reihenfolge der Adressen A, X, Z, B, F usw. ist frei wählbar, jedoch müssen Winkel und Radien in der zuvor beschriebenen **Reihenfolge** stehen (erster Winkel vor zweitem Winkel, erster Radius vor zweitem Radius **in Bearbeitungsrichtung**).

1) Zweiter Satz kann auch ein Konturzug sein

7.3 Wirkungsweise der Funktion G09, F, S, T, H, M im Konturzug

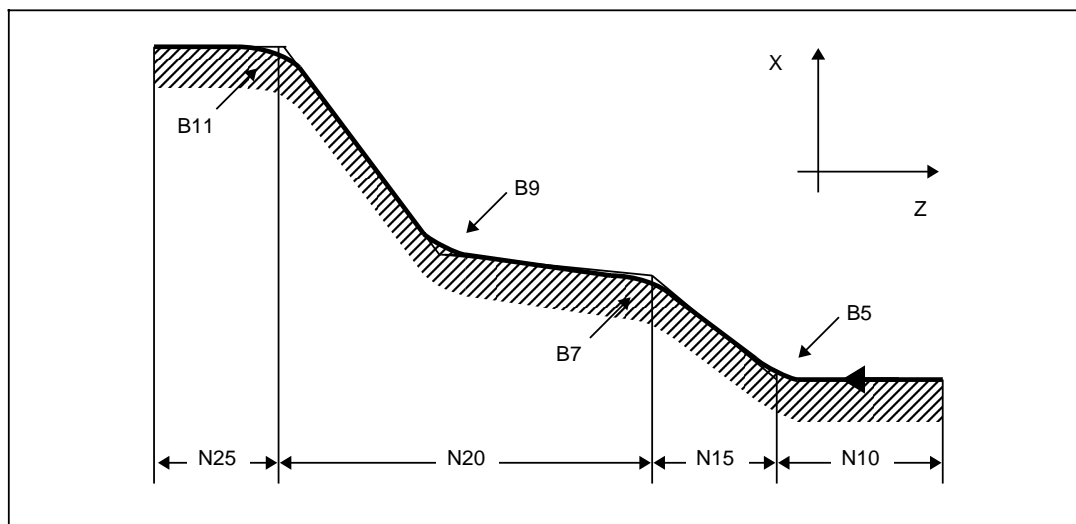
Wird in einem Konturzugsatz **G09** programmiert, wirkt dies erst am **Ende des Satzes**, also mit Erreichen der **Endposition**. Innerhalb des Konturzuges wird bei Unstetigkeitsstellen (Ecken, Kanten) automatisch G09 von der Steuerung generiert.

- Werden in einem Konturzugsatz **F, S, T, H, M** programmiert, wirken diese am **Satzanfang**,
- **M00, M01, M02, M17, M30** wirken am **Ende des Satzes**.

7.4 Kettung von Sätzen

Die Kettung von Sätzen mit und ohne Winkeleingaben mit eingefügten Radien oder Fasen ist in beliebiger Reihenfolge möglich.

Beispiel: Kettung von Sätzen



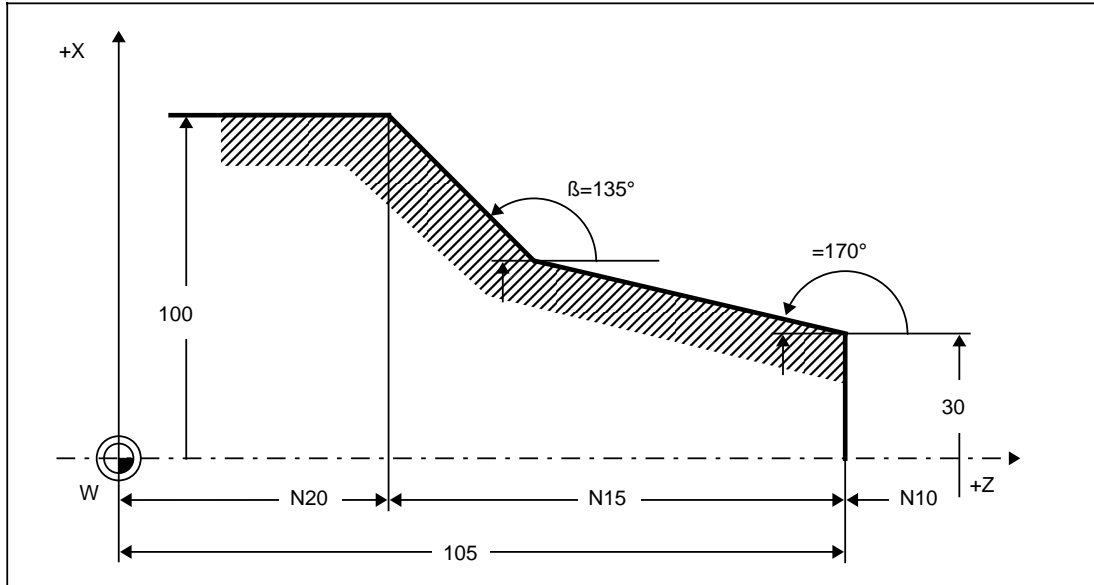
Kettung von Sätzen

N10	Z... B5 L _F	Gerade mit Radius
N15	A... X... B7 L _F	Gerade mit Radius
N20	A1... A2... X... Z... B9 B11 L _F	3 Punkte Zug mit Radius an beiden Enden
N25	Z... L _F	Gerade

7.5 Programmierbeispiele Drehmaschine

Der Winkel β bezieht sich auf den Anfangspunkt, der Winkel β auf den fehlenden Stützpunkt. Der Endpunkt kann im Bezugsmaß G90 oder im Kettenmaß G91 programmiert werden. Es sind beide Endpunktkoordinaten anzugeben. Die **Steuerung** ermittelt aus dem bekannten Anfangspunkt, den beiden Winkeln und dem Endpunkt den **Stützpunkt**.

Beispiel: Außenbearbeitung

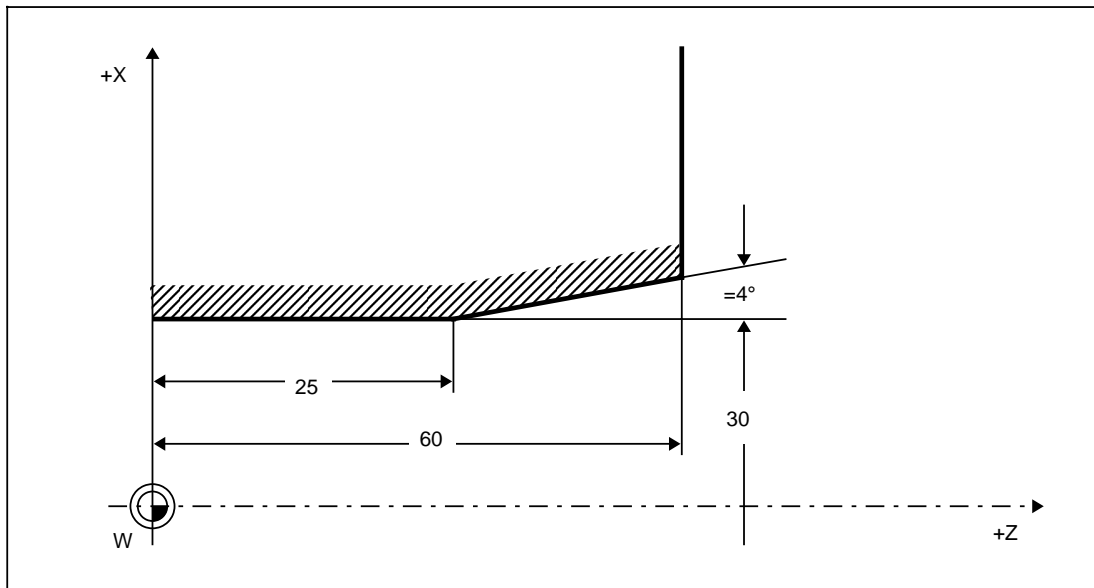


N10 G00 G90 X30 Z105 L_F

N15 G01 A170 A135 X100 Z20 F... L_F

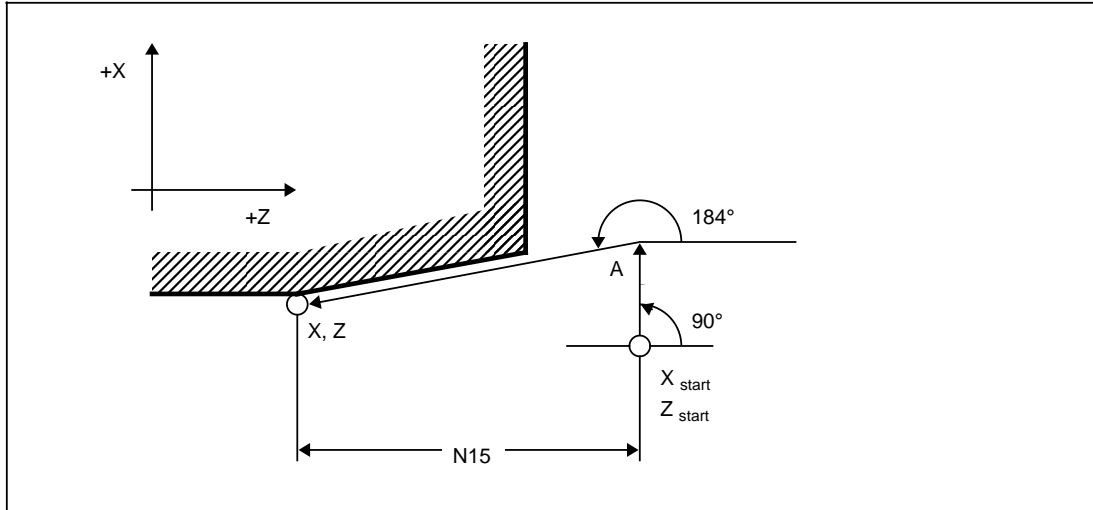
3-Punkte-Zug

Beispiel: Innenbearbeitung



Zeichnungsvermessung

Der Startpunkt wird beliebig außerhalb des Innenkegels festgelegt.



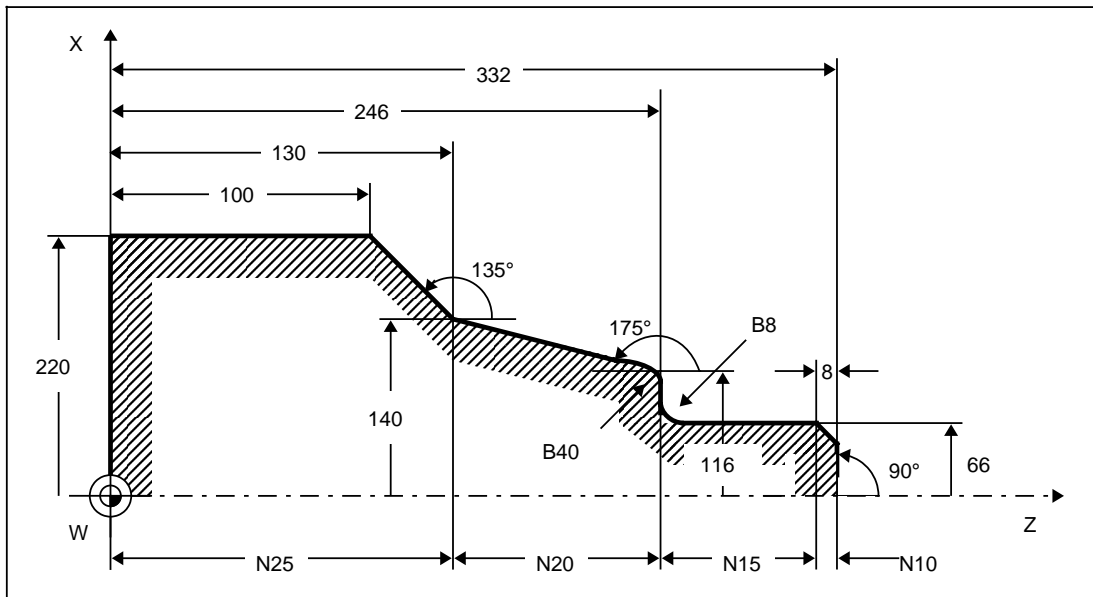
Die Senkrechte durch den Startpunkt und die Verlängerung des Innenkegels ergibt den Schnittpunkt A.

Das Programm lautet dann wie folgt:

```

.
N10 G00 X_start Z_start L_F
N15 G01 A90 A184 X... Z... F... L_F
.
    
```

Beispiel: Konturzugprogrammierung für Außenbearbeitung



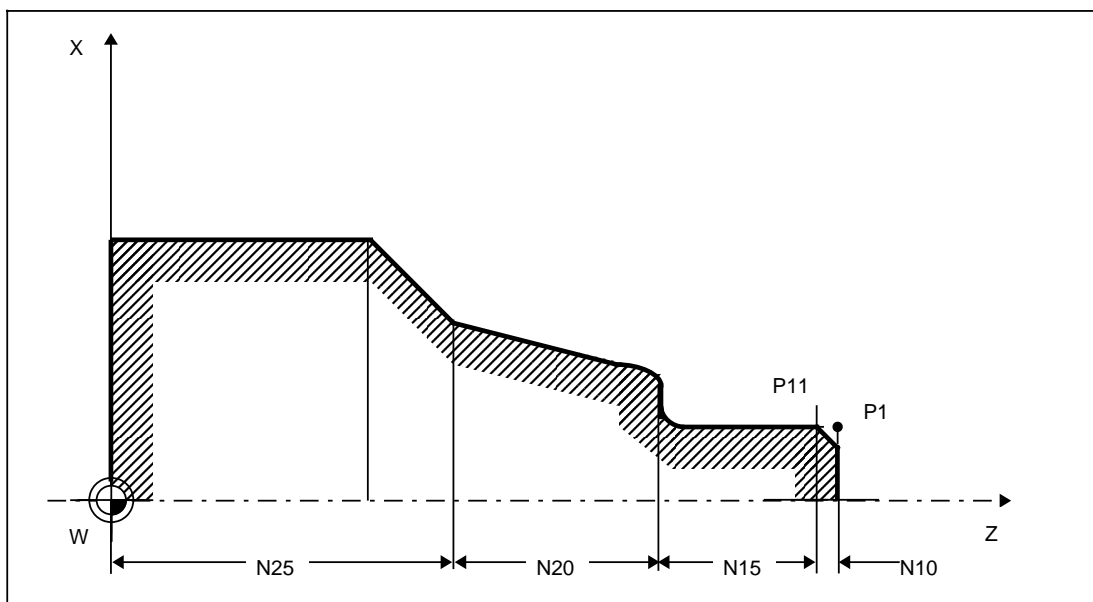
Programmierung mit Kontur-Kurzbeschreibung:

% 495 LF	
N5 G00 G90 X0 Z332 LF	Bezugsmaßvorgabe, Startpunktvorgabe
N10 G01 G09 A90 X66 B-8 F0.2 LF	2-Punkte-Zug + Fase (B-8) (Verkettung durch B-)
N15 A180 A90 X116 Z246 B8 LF	3-Punkte-Zug + Radius (B8)
N20 G03 B40 A175 X140 Z130 LF	Kreisbogen-Gerade
N25 G01 A135 A180 X220 Z0 LF	3-Punkte-Zug
N30 M02 LF	Unterprogrammende

7.6 Zusatzfunktionen in verketteten Sätzen

Verkettungen liegen immer dann vor, wenn Sätze durch Radien oder Fasen ineinander übergehen.

Beispiel: Zusatzfunktionen in verketteten Sätzen



Zwischen verketteten Sätzen kann ein Satz mit Zusatzfunktionen stehen:

```

•
N10 G01 G09 A90 X66 B-8 F0.2 LF (P1)
N101 M... H... ... LF
N15 A180 A90 X116 Z246 B8 LF
•

```

Zusatzfunktionen werden im Punkt P11 wirksam (siehe oben).

Ein **Freischneiden** erfolgt somit im Punkt P11. Der im Satz N10 programmierte F-Wert wird am Anfang von Satz N10 wirksam.

8 Werkzeugkorrekturen

8.1 Werkzeugdaten

Unter einer **Werkzeug-Korrekturnummer D** sind die **geometrischen** Werkzeugdaten abgelegt:

Länge ± 9999.999 mm, Radius ± 999.999 mm (Eingabefinheit 1 μm); T-Nr. acht Dekaden.

Im Werkzeugkorrekturspeicher der NC werden T-Nummer, Werkzeugtyp, Geometrie, Basismaß und Verschleiß aller aktiven Werkzeuge gespeichert.

- Der Speicher ist unterteilt in 99 Werkzeugkorrektur-Blöcke (D1 bis D99).
- Ein Block hat jeweils 10 Spalten bzw. **10 Werkzeug-Parameter (P0 bis P9)** und ist folgendermaßen aufgebaut:

D1 ... D99	P0 :	Werkzeugnummer	
	P1 :	Werkzeugtyp	
	P2 :	L1 Geometrie	
	P3 :	L2 Geometrie	
	P4 :	Durchmesser/Radius	
	P5 :	L1 Verschleiß	}
	P6 :	L2 Verschleiß	
	P7 :	Durchmesser/ Radius	
	P8 :	L1 Basis	
	P9 :	L2 Basis	

Verschleißdaten

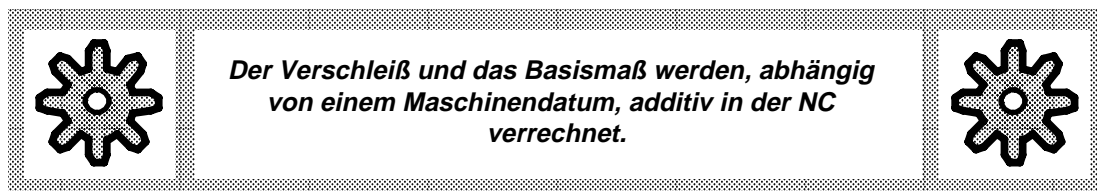
- Die Struktur des Werkzeugkorrektur-Blockes wird durch den **Werkzeugtyp (P1)** gekennzeichnet.

Aufschlüsselung des Werkzeugtyps P1:

Typ 0	Werkzeug nicht definiert
Typ 1 . . . 9	Drehwerkzeuge, Lage der Werkzeugschneide
Typ 10 . . . 19	Werkzeuge nur mit wirksamer Längenkorrektur (z. B. Bohrer)
Typ 20 . . . 29	Werkzeuge mit Radius- und einer Längenkorrektur (z. B. Fräser)
Typ 30 . . . 39	Werkzeuge mit Radius- und zwei Längenkorrekturen (z. B. Winkelkopfräser)

- Der **Aufruf** der Werkzeugkorrektur erfolgt max. dreidekadig mit **D1 bis D99**.
- Die **Abwahl** der Werkzeugkorrektur erfolgt mit **D0**.

Die Korrektur wird erst ausgeführt wenn die entsprechende **Achse programmiert** wird. Geometrie und Verschleiß werden z.B. durch Meßzyklen in der NC aktualisiert.

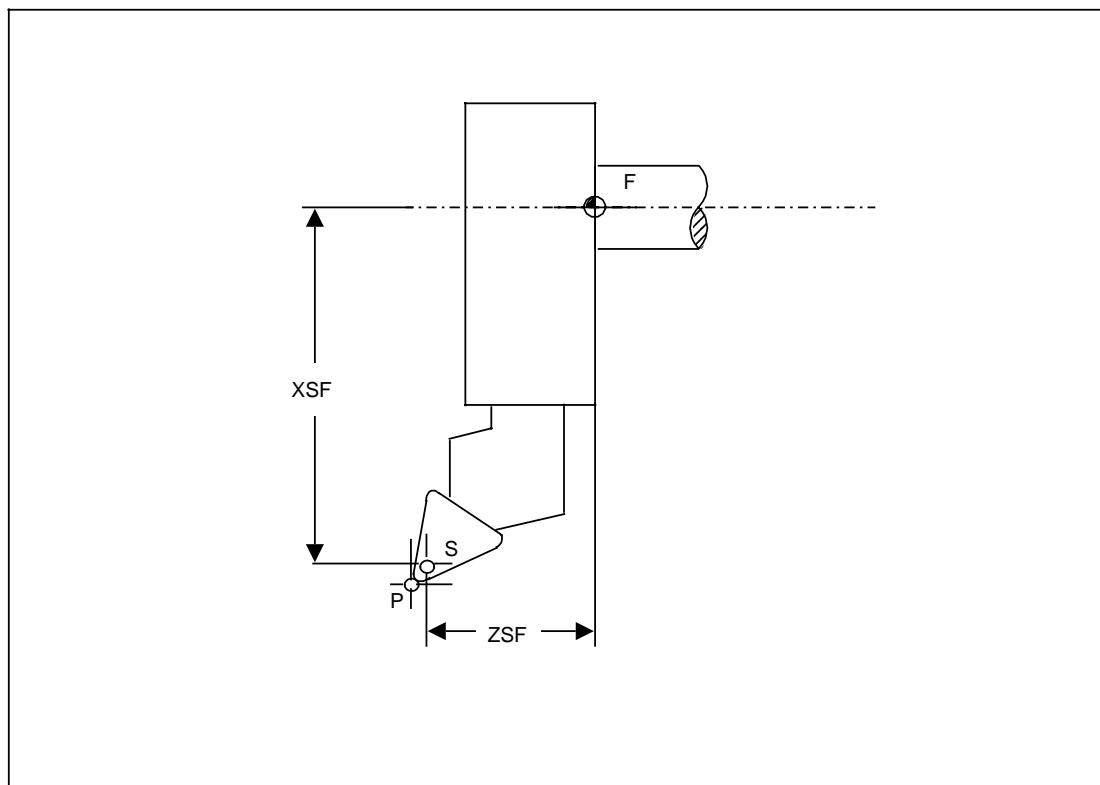


Die Werte der Werkzeugkorrektur können nicht nur über die Bedientafel, sondern auch über die **Dateneingabe-Schnittstelle** eingegeben und ausgelesen werden. Hierbei dürfen **keine** Satznummern programmiert werden (siehe Lochstreifenformate).

Vom Maschinenhersteller kann die maximale Verschleißangabe von $\pm 9,999$ mm auf $\pm 0,999$ mm beschränkt werden.

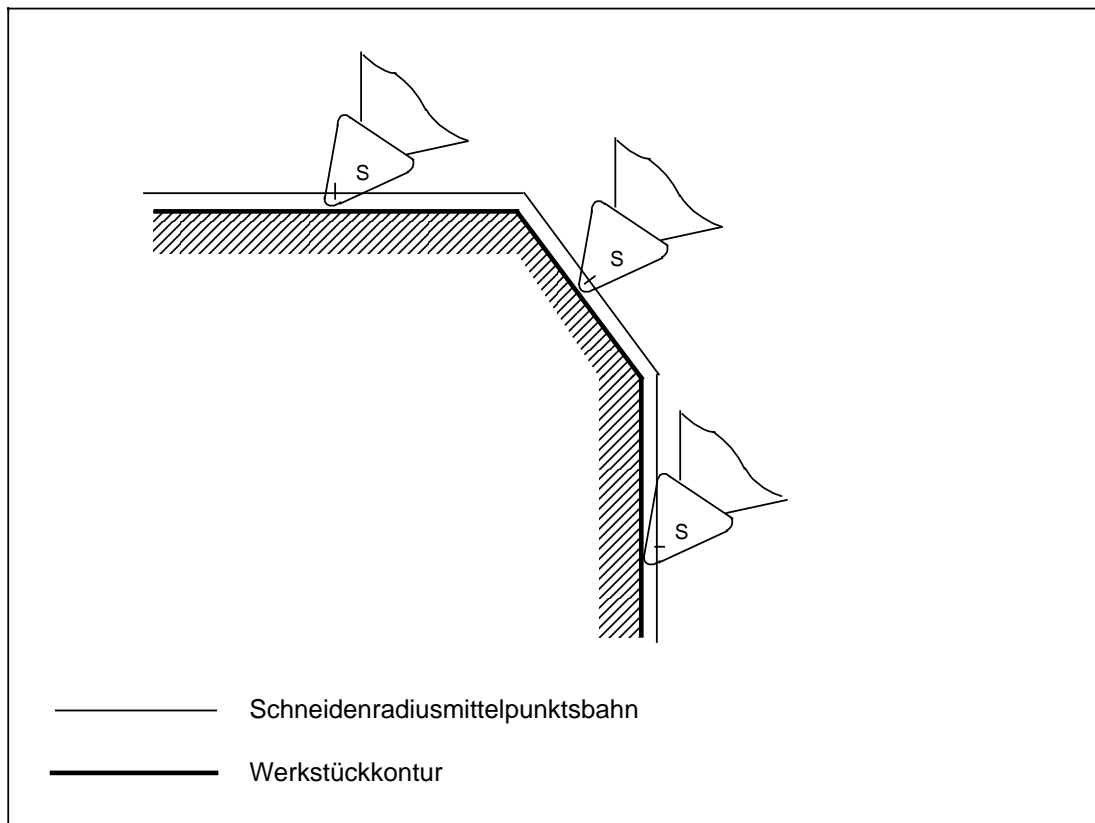
8.2 Werkzeugkorrektur ohne Einsatz der Schneidenradiuskorrektur

Die **wirksame Werkzeugkorrektur** errechnet sich aus der Summe der Werkzeuglängenkorrekturen und einer evtl. anstehenden externen additiven Werkzeuglängenkorrektur. Die Summe entspricht dem Maß **XSF bzw. ZSF**.



- P** theoretische Stahlspitze
- S** Schneidenradiusmittelpunkt
- F** Schlittenbezugspunkt
- XSF** Werkzeuglängenkorrektur in X-Richtung
- ZSF** Werkzeuglängenkorrektur in Z-Richtung.

Programmiert wird die **Bahn des Schneidenradiusmittelpunktes S**. Die Längenkorrektur ist auf den Schneidenradiusmittelpunkt bezogen.

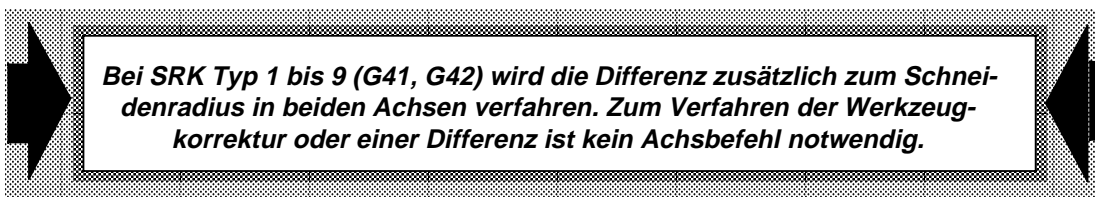


Korrektureinrechnung

Beim **Wechsel der Werkzeugkorrekturnummer** wird die Differenz zwischen dem neuen und dem alten Korrekturwert gebildet.

Bei der **Inbetriebnahme** wird festgelegt ob die Differenz

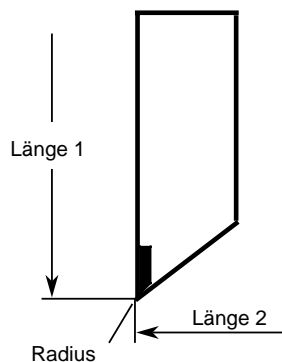
- direkt nach dem Wechsel verfahren wird,
- oder erst beim programmierten Verfahren der entsprechenden Achse berücksichtigt wird.



Aufbau des Werkzeugkorrekturspeichers

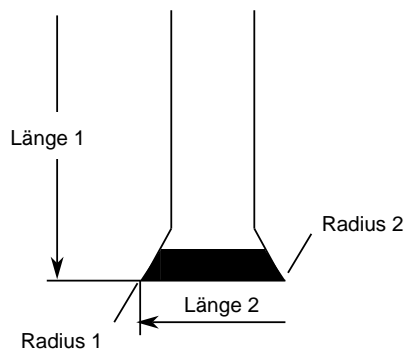
	Struktur der Korrekturspeicher										Programmierung
	T-Nr.	Typ	Geometrie			Verschleiß			Basis (Zus.-WK)		Werkzeug-Aufruf
	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	
Dn	123...8	1...9	Länge 1	Länge 2	Radius	Länge 1	Länge 2	Radius	Länge 1	Länge 2	T100 . G16 ZX . . D50 Länge 1 in X-Achse (Planachse) Länge 2 in Z-Achse

1. Drehmeißel



	Struktur der Korrekturspeicher										Programmierung
	T-Nr.	Typ	Geometrie			Verschleiß			Basis (Zus.-WK)		Werkzeug-Aufruf
	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	
Dn	123...8	1...9	Länge 1	Länge 2	Rad. 1	Länge 1	Länge 2	Radius	Länge 1	Länge 2	T100
Dm	123...8	1...9	Länge 1	Länge 2	Rad. 2	Länge 1	Länge 2	Radius	Länge 1	Länge 2	. D50 (z.B. für linke Schneide)
m=											. D51 (z.B. für rechte Schneide)
n+1											

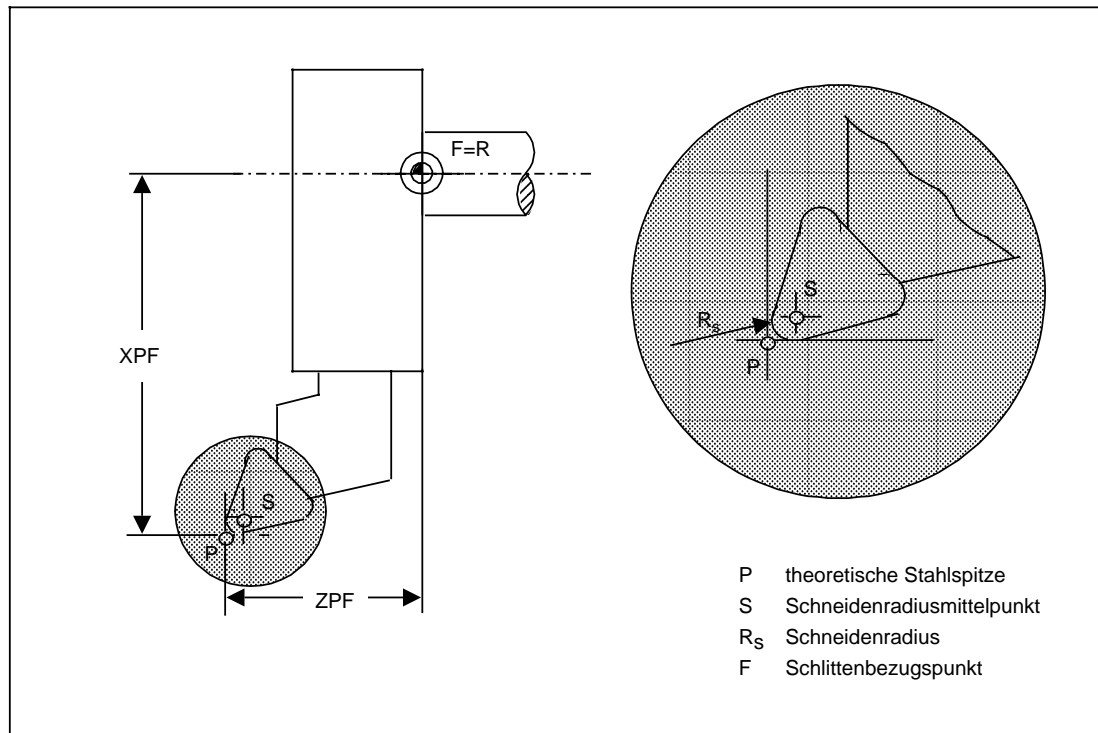
2. Einstechstahl



8.3 Werkzeugkorrektur mit Einsatz der Schneidenradiuskorrektur

Bei Einsatz der Schneidenradiuskorrektur (SRK) kann die Werkstückkontur programmiert werden. Die einzugebende Längskorrektur wird auf den Schneidenpunkt "P" bezogen. Zusätzlich muß der Schneidenradius R_S und die Lage des Schneidenpunktes P eingegeben werden.

Die Steuerung errechnet die dann zu verfahrenende Bahn. Es entstehen keine Konturfehler. Die Schneidenradiuskorrektur ist im Endpunkt des Satzes wirksam in dem sie **aufgerufen** wurde (G41, G42), d.h. der nächste Satz wird richtig abgefahren.



Zur Berechnung der Schneidenradiuskorrektur benötigt die Steuerung den Schneidenradius (**P4**) und eine Angabe darüber, wie der Drehmeißel im Werkzeugträger eingespannt ist. Dazu müssen Sie in die Werkzeugkorrekturspalte P1 (Werkzeugtyp) eine der Kennungen (**P1=1 bis P1=9**) eingeben.

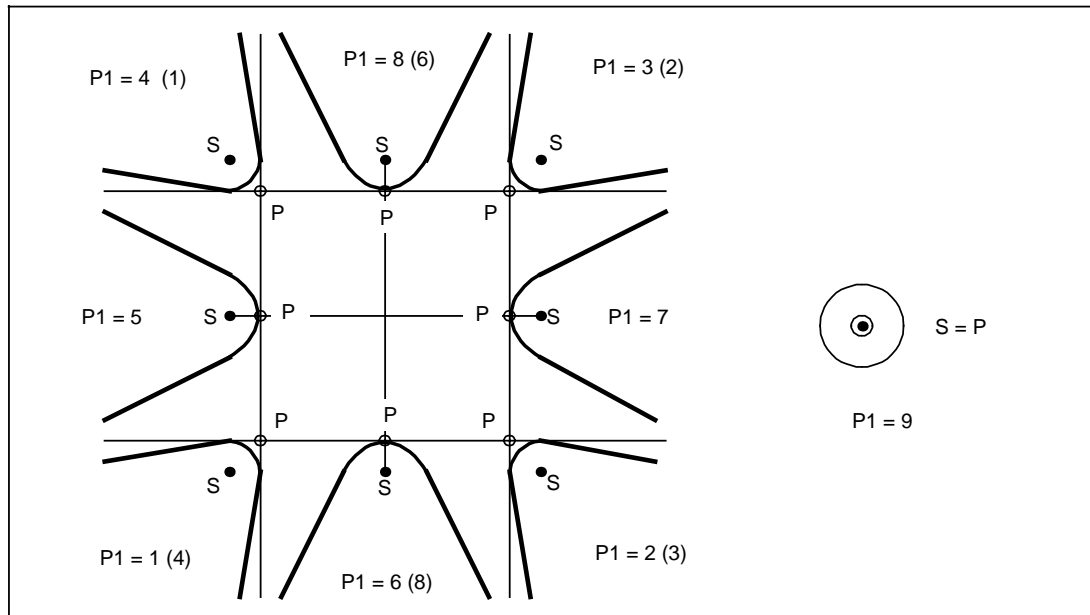
Damit ist die relative Lage des Punktes P zum Schneidenmittelpunkt S festgelegt.

Wird als Bezugspunkt für die Ermittlung der Werkzeug-Längskorrektur nicht der Punkt P sondern der Schneidenmittelpunkt **S benutzt**, so verwenden Sie die Kennung **P1=9**. Die Kennung P1=0 ist nicht zugelassen.

Im folgenden Bild sind die **möglichen Lagen der Schneidenradien** mit den zugehörigen **Kennungen P1=1 bis P1=9** dargestellt.

Die in **Klammern** angegebenen Werte gelten für die Bearbeitung **vor der Drehmitte**.

Lage der theoretischen Schneidenkante P bezogen auf den Schneidenradiusmittelpunkt S (Blickrichtung ist immer S zu P).



Bearbeitung hinter der Drehmitte (vor der Drehmitte).

9 Schneidenradiuskorrektur (SRK)

Alle **Stoppunkte bei Einzelsatz** sind mit **S** gekennzeichnet.

In Klammern steht die zugehörige Satznummer.

Im Satz nach dem Anwahlsatz wird ein Satzanfangsvektor (**Länge R**) **senkrecht** auf der programmierten Bahn errichtet.

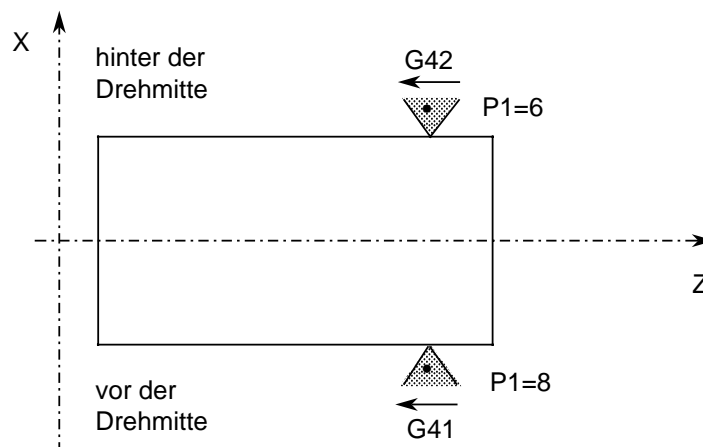
Konturelemente sind mit dicker Linie (**—**) gekennzeichnet.

9.1 Anwahl der SRK

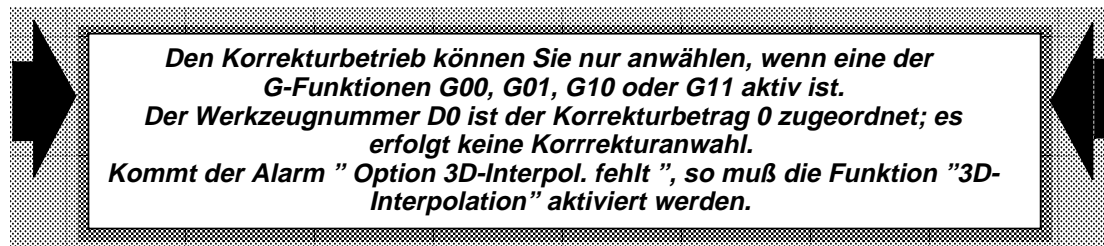
Die **Anwahl des Korrekturbetriebes** erfolgt in der festgelegten Ebene mit den Wegbedingungen **G41/G42** und Angabe der Korrekturnummer **D**.

Hierbei wird die Korrektur bei **G41 links** (in Fahrrichtung) und bei **G42 rechts** (in Fahrrichtung) von der Werkstückkontur durchgeführt.

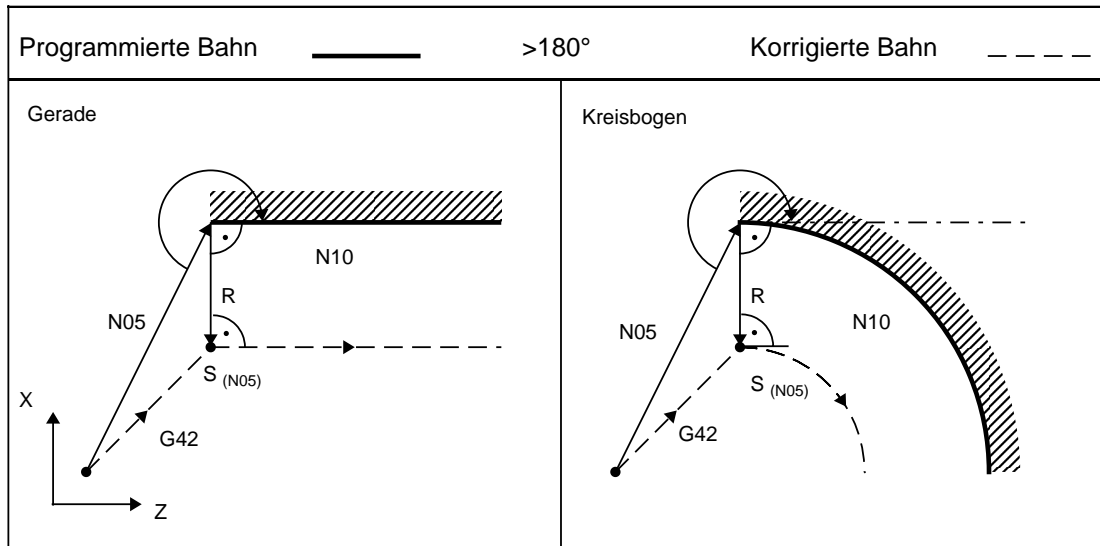
Beispiel für ein Drehteil:



Bei der Anwahl der SRK werden immer zwei bzw. drei Programmsätze zur Schnittpunktberechnung eingelesen.

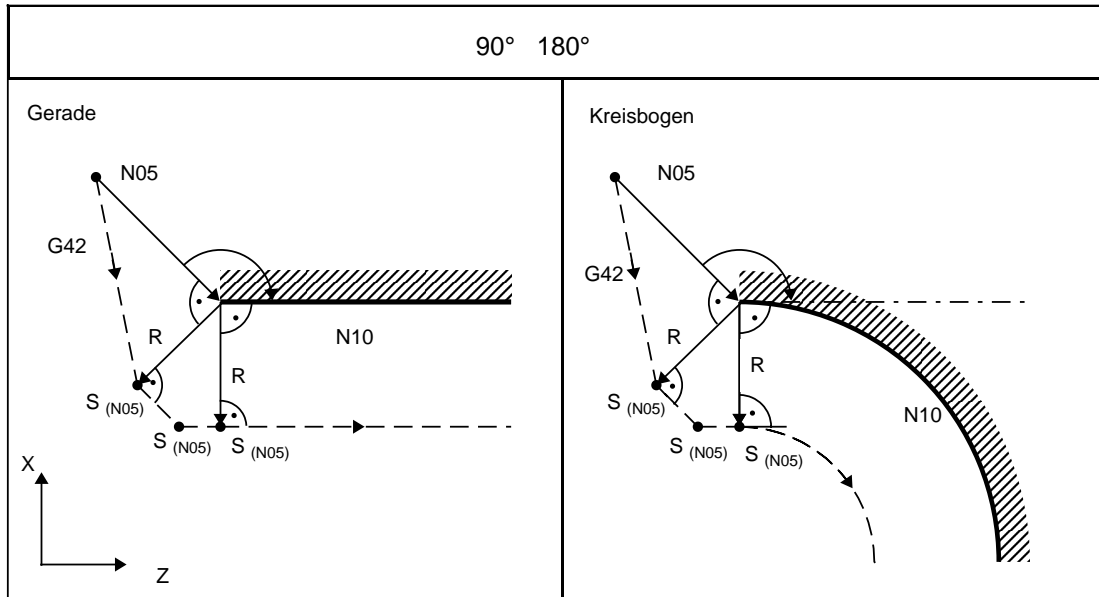


Die folgenden Bilder zeigen die **Korrekturanwahl bei verschiedenen Anfahrwinkeln**. Die Bilder sind mit G42 dargestellt. Bei Programmen mit G41 gilt als Winkelübergang $\beta=360^\circ$.

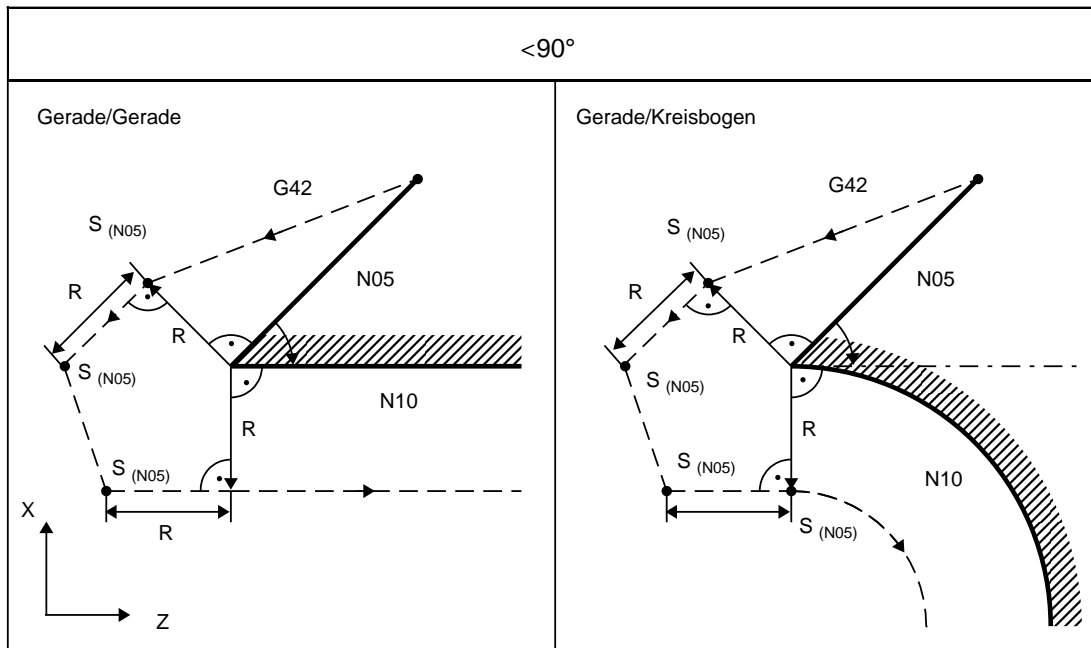


Anwahl des Korrekturbetriebes bei >180°

Anwahl des Korrekturbetriebes



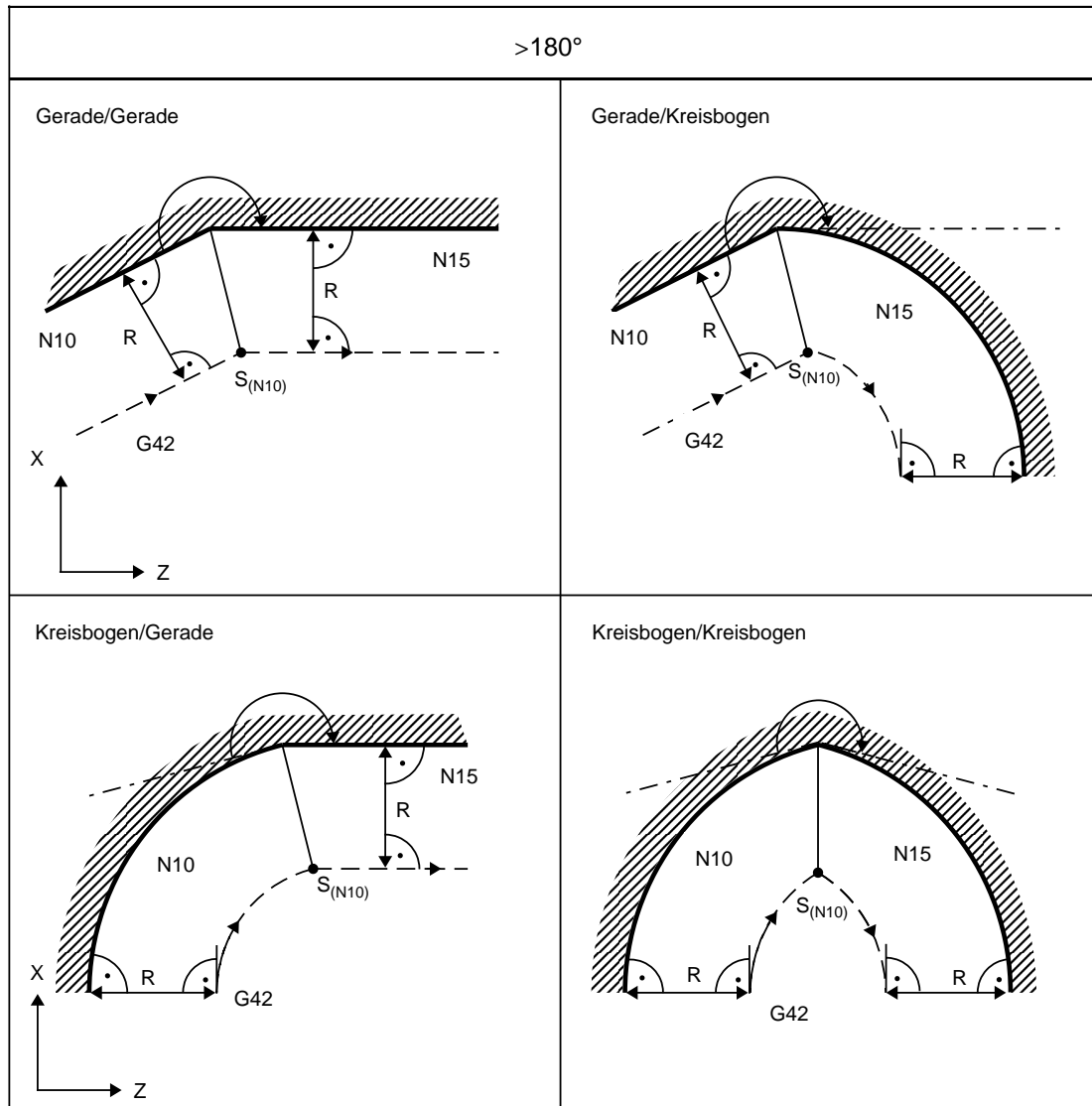
Anwahl des Korrekturbetriebes bei $90^\circ \quad 180^\circ$



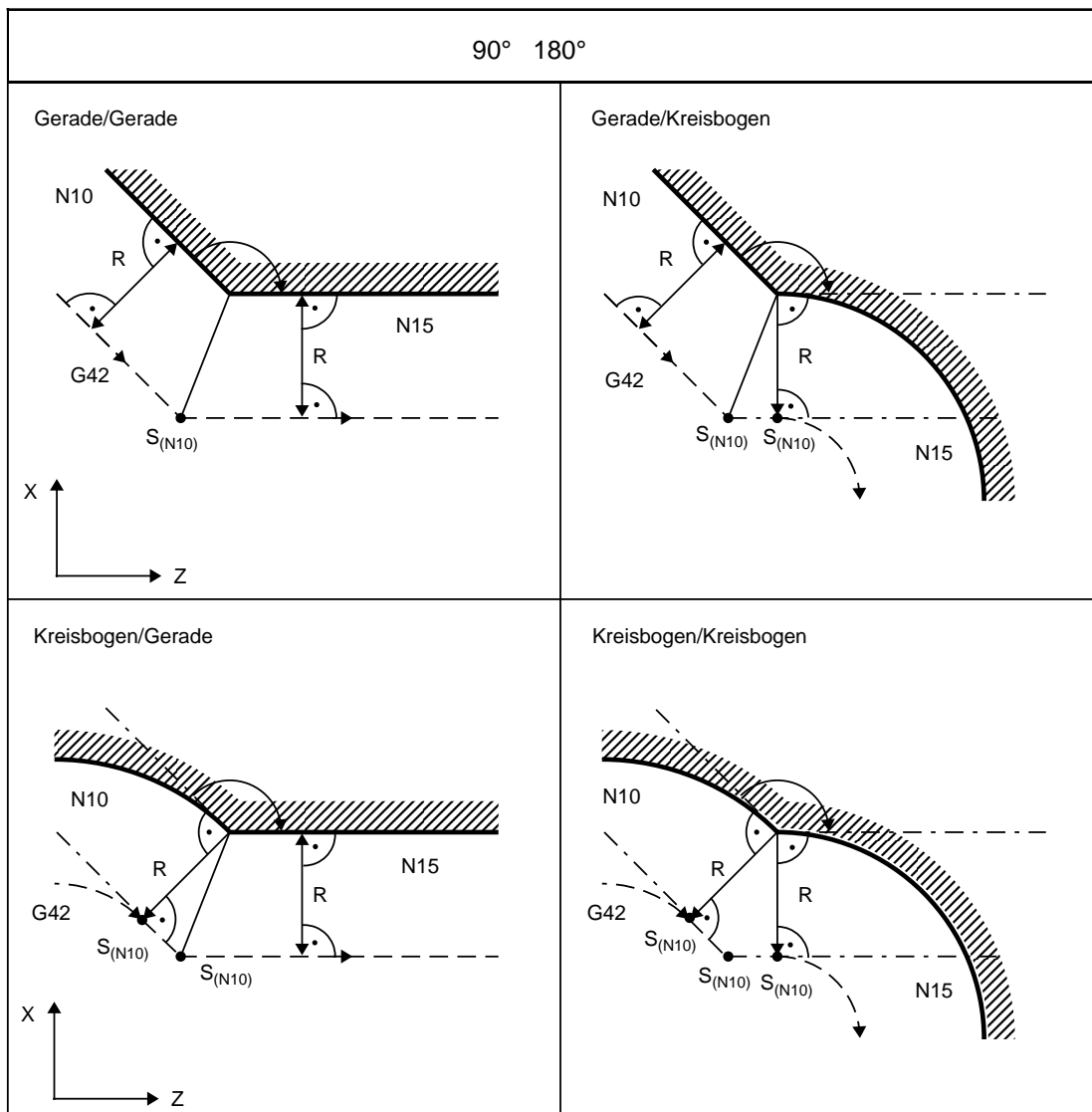
Anwahl des Korrekturbetriebes bei $<90^\circ$

9.2 SRK im Programm

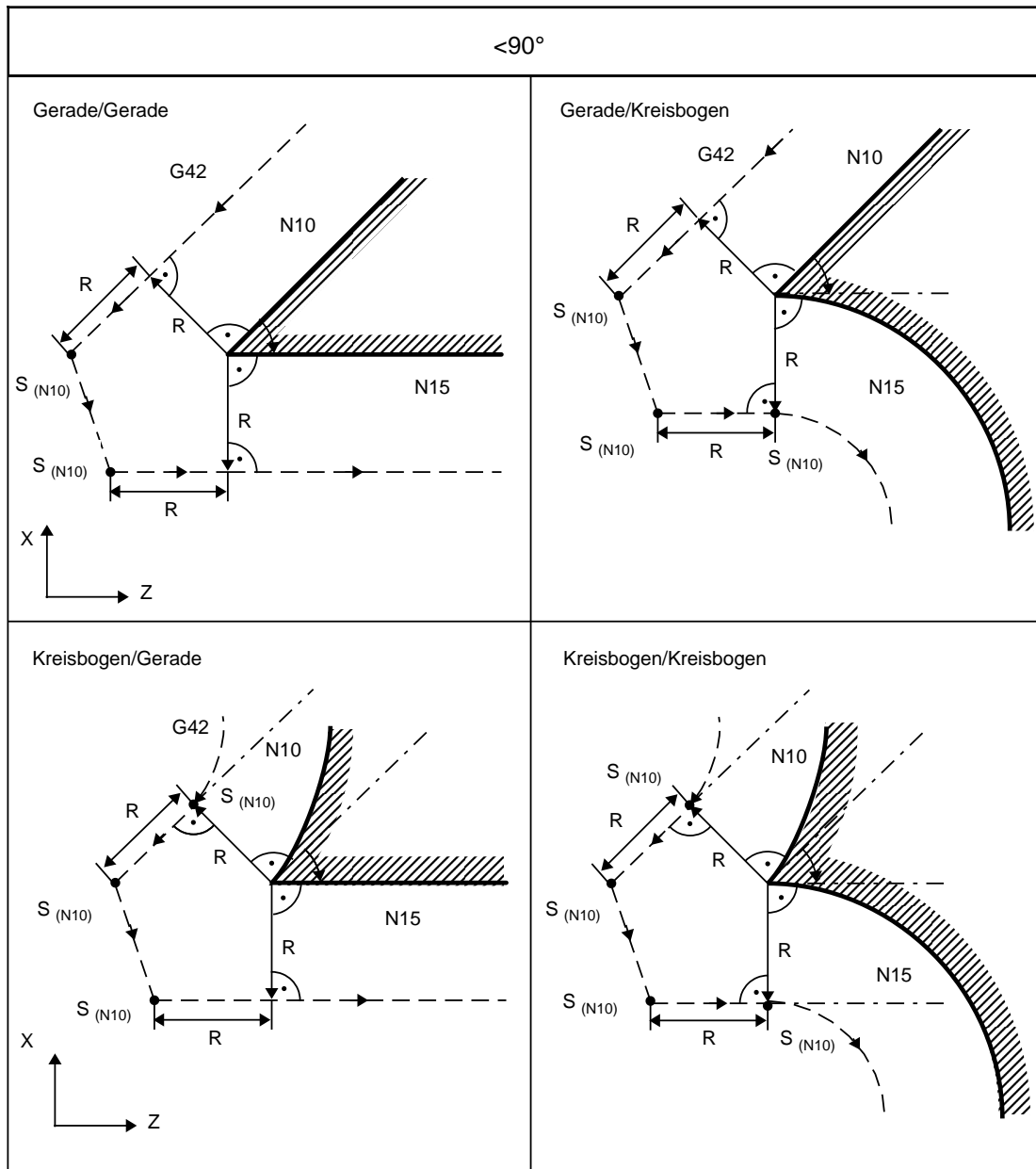
Bei der SRK werden von der Steuerung bereits während der **Abarbeitung des aktuellen Satzes** zwei weitere Sätze im voraus eingelesen und der **Schnittpunkt der korrigierten Bahnen berechnet**. Den Korrekturbetrieb mit G42 bei verschiedenen Übergängen zeigen die folgenden Bilder.



SRK bei verschiedenen Übergängen bei $>180^\circ$



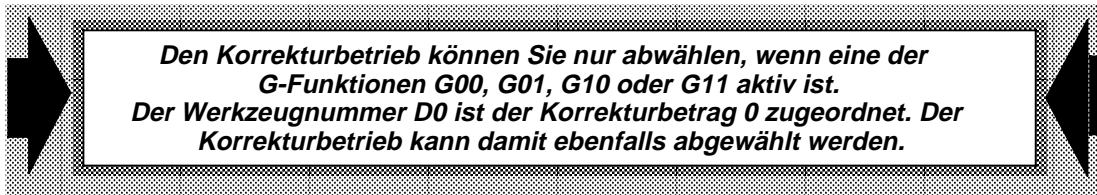
SRK bei verschiedenen Übergängen bei 90° 180°



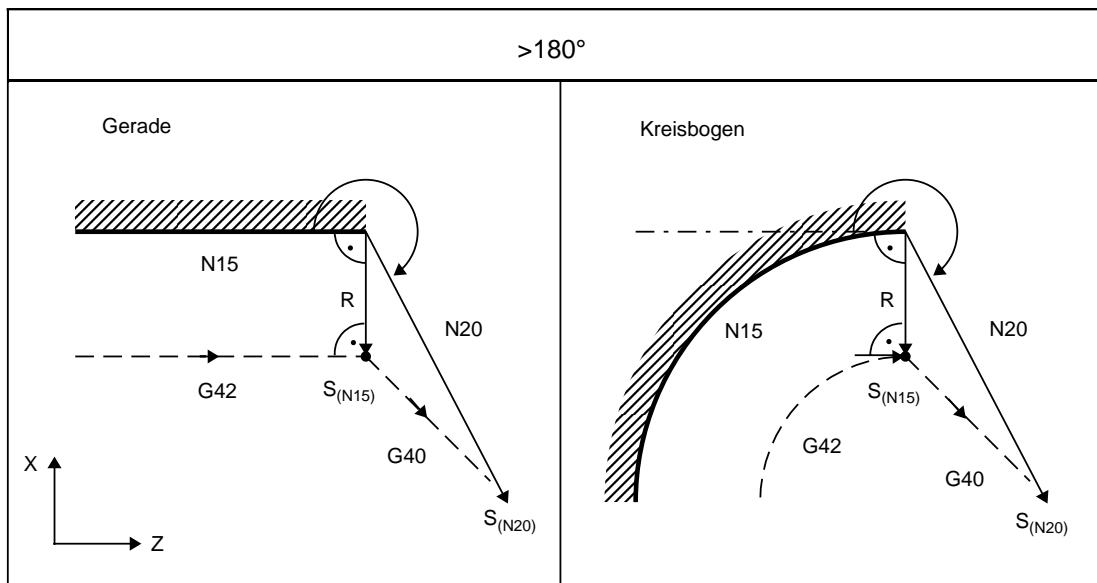
SRK bei verschiedenen Übergängen bei $<90^\circ$

9.3 Abwahl der SRK (G40)

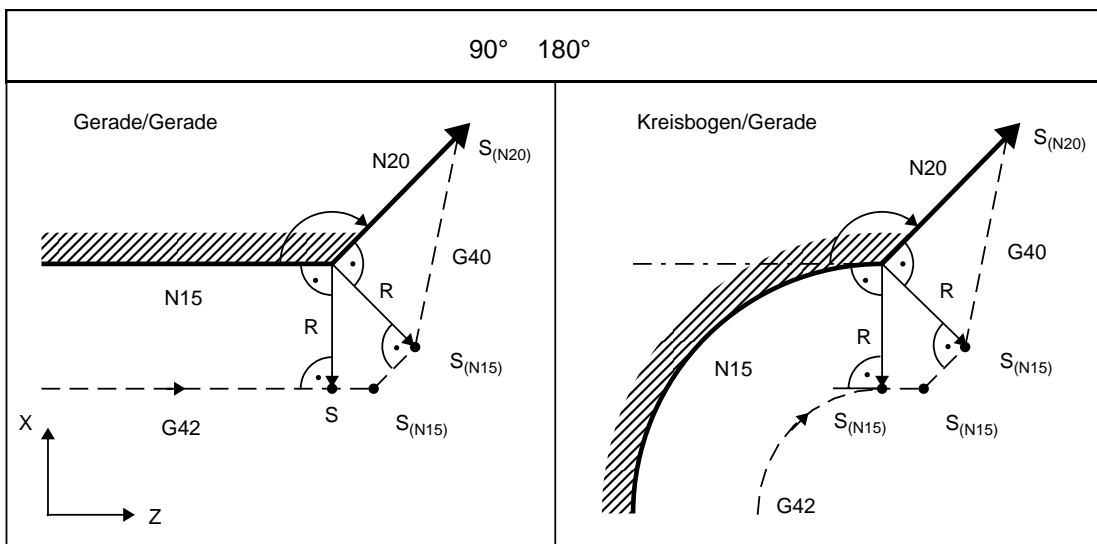
Die **Abwahl** des Korrekturbetriebs erfolgt mit der Wegbedingung **G40**.



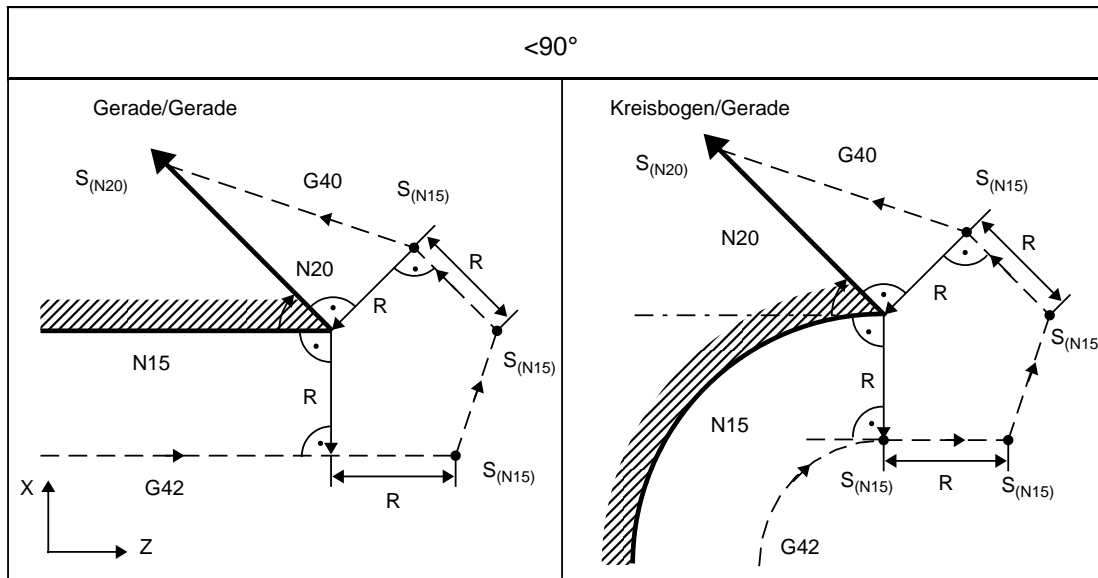
Abwahl des Korrekturbetriebes



Abwahl des Korrekturbetriebes bei $> 180^\circ$

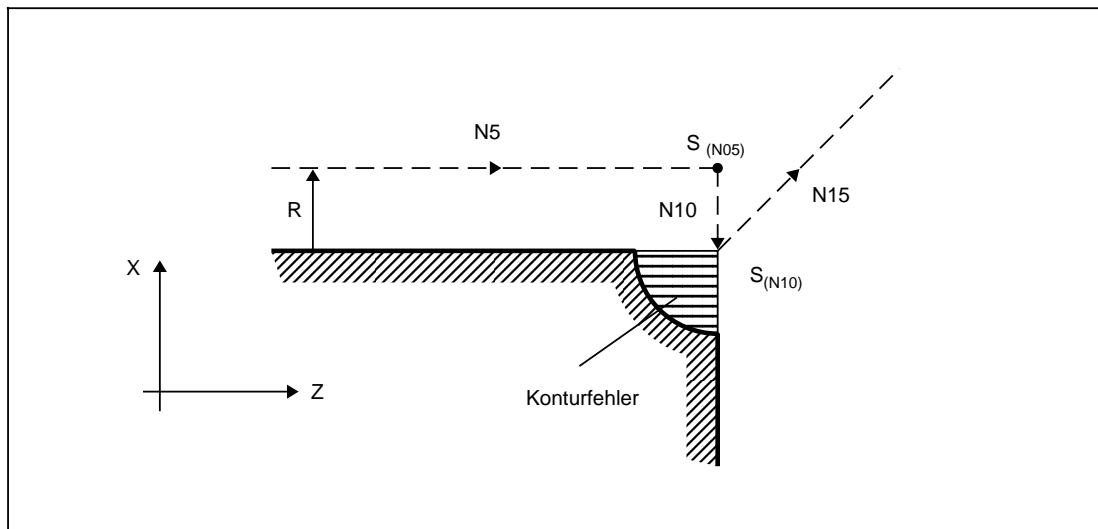


Abwahl des Korrekturbetriebes bei 90° 180°



Abwahl des Korrekturbetriebes bei $<90^\circ$

Abwahl des Korrekturbetriebes (Sonderfall)



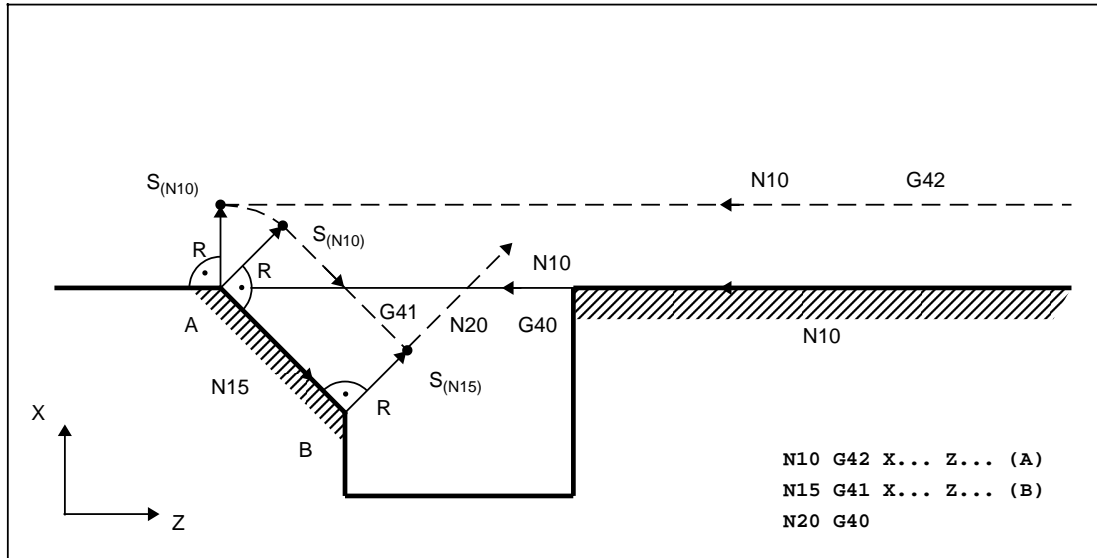
Abwahl der SRK in einem Satz "Strecke = 0"

N5 G91 Z100 L_F
 N10 G40 Z0 L_F
 N15 Z100 X+100 L_F

Konturfehler (gestricheltes Gebiet).

9.4 Wechsel der Korrekturrichtung (G41, G42)

Im Endpunkt des Satzes mit der **alten G-Funktion**, z.B. G42 und im Anfangspunkt (A) des Satzes mit der **neuen G-Funktion** z.B. G41 wird ein senkrechter **Vektor** mit der Länge **R** in der entsprechenden **Korrekturrichtung** errichtet.

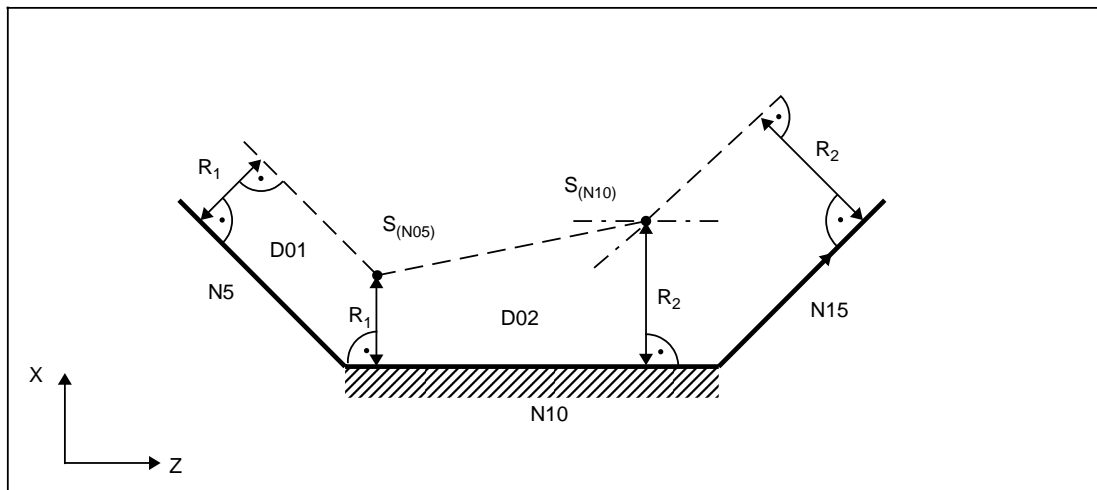


Wechsel der Korrekturrichtung

9.5 Wechsel der Korrekturnummer (G41 D.. , G41 D..)

Bei Wechsel der Korrekturnummer von z.B. G41 D01 auf G41 D02 gilt:

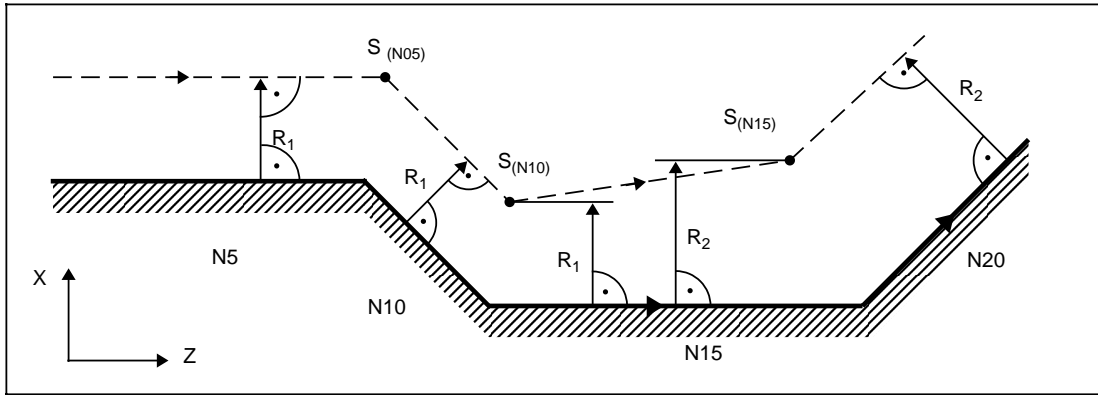
Es wird **kein** Satzanfangsschnittpunkt mit der alten Korrektur errechnet; im Satzende des Satzes mit der alten Korrekturnummer D01 wird ein Vektor mit der Länge **R1** senkrecht auf der Kontur errichtet; der **Satzendeschnittpunkt** wird mit der neuen Korrektur D02 **errechnet**.



Wechsel der Korrekturnummer

9.6 Wechsel der Korrekturwerte (R1, R2)

Die **Korrekturwerte** können über die Bedientafel, über Lochstreifeneingabe, über die externe Werkzeugkorrektur oder im **Teileprogramm geändert** werden. Der **neue** Korrekturwert wird im **nächsten Satz** (im Beispiel N20) wirksam.



Wechsel der Korrekturwerte

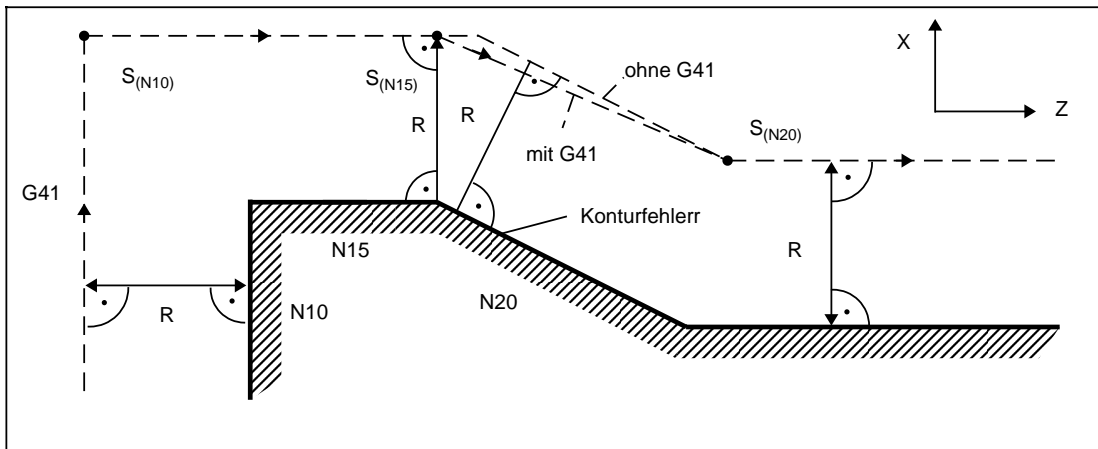
9.7 Wiederholung der schon angewählten G-Funktion (G41, G42) mit der gleichen Korrekturnummer

Wird ein bereits programmiertes **G41, G42 wiederholt**, dann wird im **vorhergehenden Satz** im **Satzendpunkt** ein **Vektor** mit der Länge **R** senkrecht auf der programmierten Bahn errichtet.

Für den folgenden Satz wird der Satzanfangsschnittpunkt S1 **berechnet**:

```
N5 G91 D10 G41 X... Z... LF
N10 X... LF
N15 Z... LF
N20 G41 X... Z... LF
N25 Z... LF
```

Fehler! G41 in N20 wiederholt. Dabei entsteht ein Konturfehler.

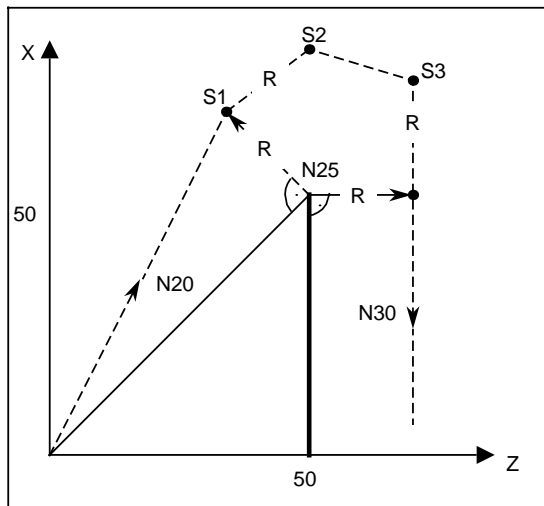


Wiederholte Anwahl

9.8 M00, M01, M02 und M30 bei angewählter SRK

M00, M01

Die NC **stoppt** am Stoppunkt **S1**, **S2** und **S3** für **Einzelsatz**. (Die Punkte sind in den Bildern gekennzeichnet).



M00, M01 bei angewählter SRK.

Es sind **zwei Fälle** zu unterscheiden:

Fall 1:
 N10 G01 F100 Z0 X0 L_F
 N20 G41 Z50 X50 L_F
 N25 H111 M00 L_F
 N30 X0 L_F

H- und M-Funktionen werden am Punkt S3 ausgeführt.

Fall 2:
 N10 G01 F100 Z0 X0 L_F
 N20 G41 Z50 X50 L_F
 N25 H111 M00 Q-5 L_F
 N30 X0 L_F

H-/M-Funktionen und Q-Bewegung werden am Punkt S1 ausgeführt.

M02, M30

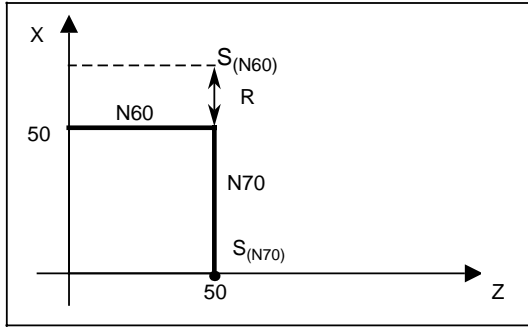
- Die **Korrektur** wird einen Satz vor dem Satz **herausgefahren**, in dem sie mit **G40** abgewählt und mit mindestens **einer Achsadresse** programmiert wurde (hier N150).

```
N150 X.. Z.. LF
N200 G40 X.. M30 LF
```

- Die Korrektur wird **nicht** herausgefahren, wenn bei G40 **kein Weg** programmiert ist und danach **M30** steht:

```
N150 X.. Z.. LF
N200 G40 LF
N250 M30 LF
```

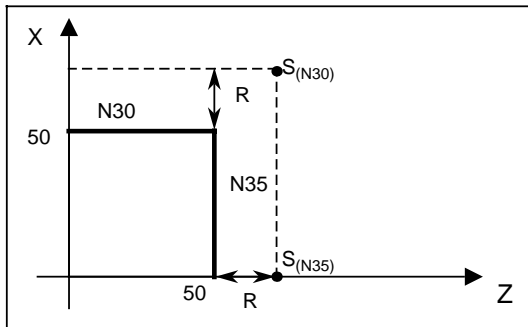
Abwahl mit G40:



N60 Z50 X50 L_F
 N70 G40 Z50 X0 M30 L_F

Die Korrektur wird einen Satz vor dem Satz herausgefahren, indem sie mit G40 abgewählt wurde (hier: N60)

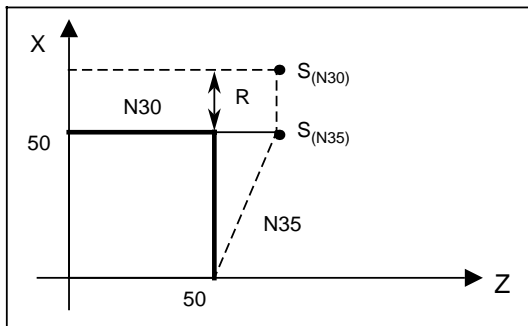
Kein Weg bei G40 oder Programmierung einer Achse im letzten Satz:



N30 Z50 X50 L_F
 N35 X0 L_F
 N40 G40 L_F
 N45 M30 L_F

Die Korrektur wird nicht herausgefahren.

G40 im vorletzten Satz oder Programmierung von 2 Achsen im letzten Satz:



N30 Z50 X50 L_F
 N35 G40 X0 L_F
 N40 M30 L_F

Die Korrektur wird in N35 herausgefahren.

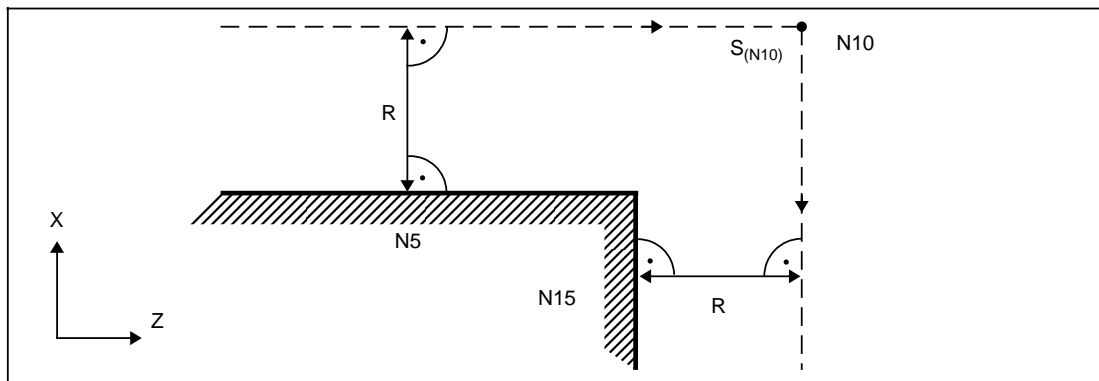
9.9 SRK bei Kombination von verschiedenen Satzarten und Auftreten von Konturfehlern

Im **Korrekturbetrieb** sind bei der Programmierung im Hinblick auf **Konturfehler** besonders die Sätze **ohne Werkzeugbewegung** zu beachten. Dabei sind 3 Satztypen zu unterscheiden:

- Sätze mit Strecke 0:
Wegadressen sind programmiert, es findet aber keine Bewegung statt, weil der **Weg 0** ist.
`N5 G91 Z0 L_F`
- Sätze "ohne Weg":
 In der Korrektorebene sind **keine** Wegadressen programmiert, sondern Hilfsfunktionen oder Verweilzeiten.
`N5 M05 L_F`
`N10 G04 X100 L_F`
- Sätze "Nicht in der Korrektorebene":
 Es sind Achsadressen außerhalb der Korrektorebene programmiert.

Beispiele:

- Ein "Hilfsfunktionssatz" zwischen Strecken in der Korrektorebene.

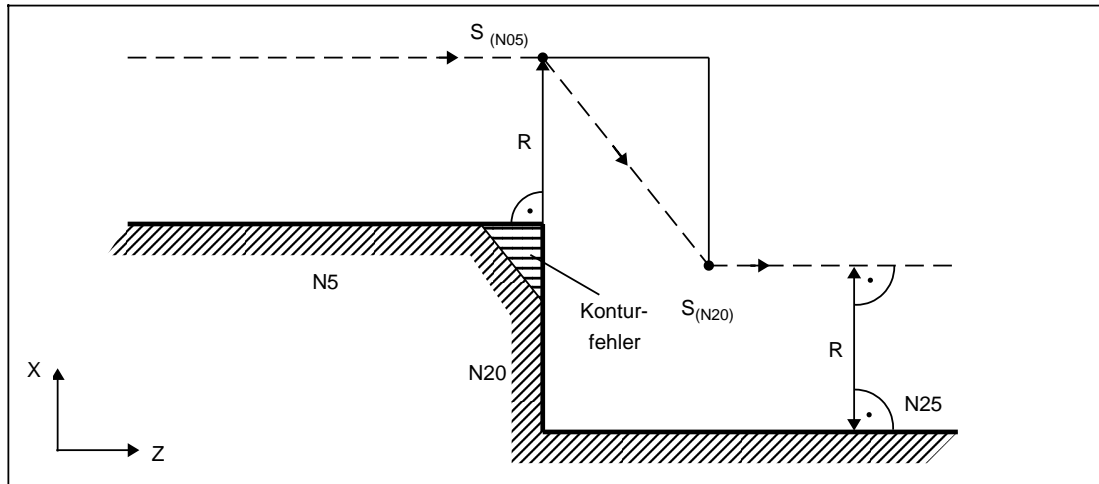


SRK: Ein "Hilfsfunktionssatz" zwischen zwei Bewegungssätzen

```
N5 G91 Z100 L_F
N10 M08 L_F
N15 X-100 L_F
```

Der Satz **N10** wird am Punkt **S_(N10)** ausgeführt.

- **Zwei** "Hilfsfunktionssätze" zwischen Strecken in der Korrekturebene.



SRK: Zwei "Hilfsfunktionssätze" zwischen zwei Bewegungssätzen

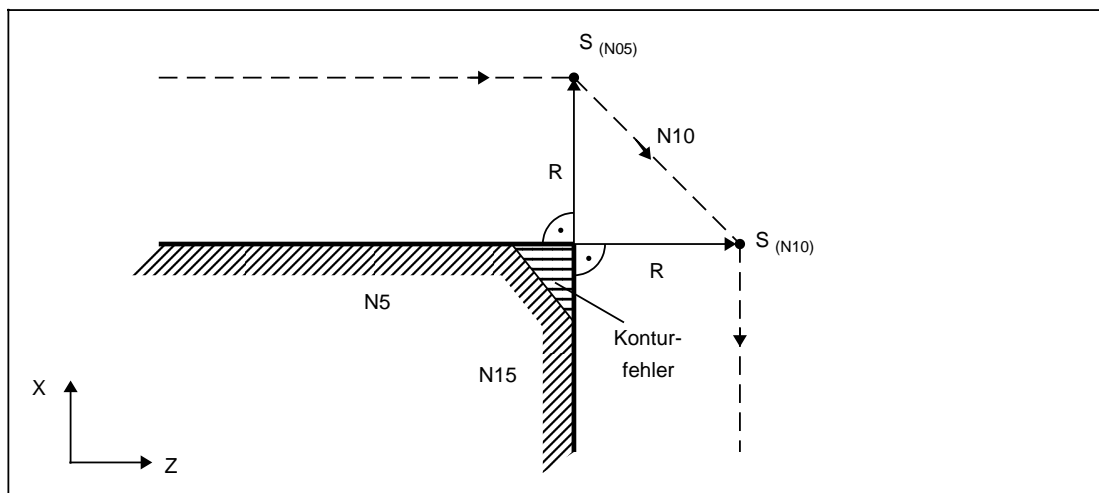
```

N5  G91  Z100  L_F
N10 M08  L_F
N15 M09  L_F
N20 X-100 L_F
N25 Z100  L_F

```

Die Sätze **N10** und **N15** werden am Punkt **S (N05)** ausgeführt.
Konturfehler (gestricheltes Gebiet)

- **Ein Satz** "Strecke =0" zwischen Strecken in der Korrekturebene



SRK: Ein Satz mit "Strecke 0"

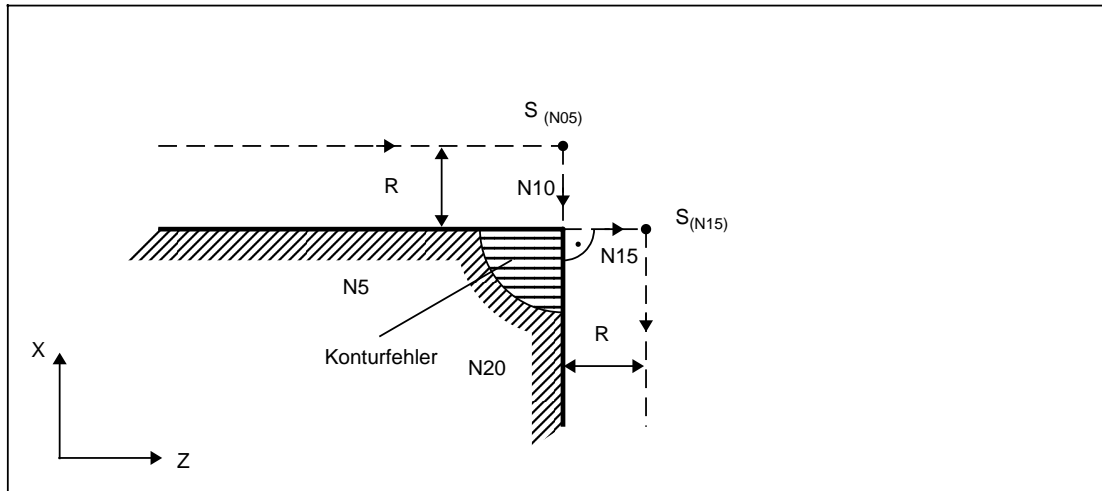
```

N5  G91  Z100  L_F
N10 Z0    L_F
N15 X-100 L_F

```

Konturfehler (gestricheltes Gebiet)

- **Zwei Sätze** "Strecke 0" zwischen Strecken in der Korrektorebene.

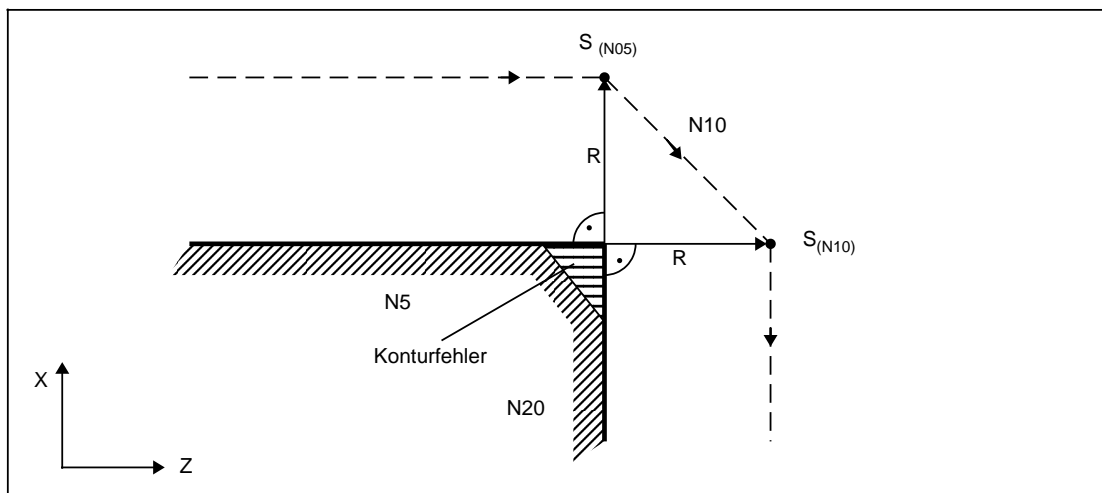


SRK: Zwei Sätze mit "Strecke 0"

N5 G91 Z100 L_F
 N10 Z0 L_F
 N15 Z0 L_F
 N20 X-100 L_F

Konturfehler (gestricheltes Gebiet)

- **Ein Satz** "Strecke = 0" und ein "Hilfsfunktionssatz" zwischen Strecken in der Korrektorebene.

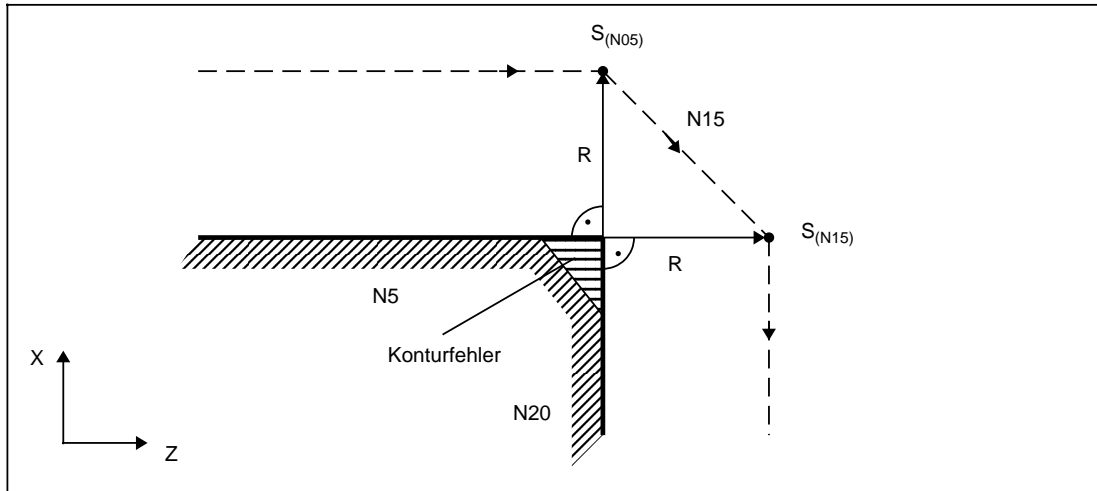


SRK: Ein Satz mit "Strecke 0" und ein "Hilfsfunktionssatz"

N5 G91 Z100 L_F
 N10 Z0 L_F
 N15 M08 L_F
 N20 X-100 L_F

Der Satz **N15** wird am Punkt **S(N10)** ausgeführt.
Konturfehler (gestricheltes Gebiet)

- Ein "Hilfsfunktionssatz" und ein Satz "Strecke=0" zwischen Strecken in der Korrektorebene.

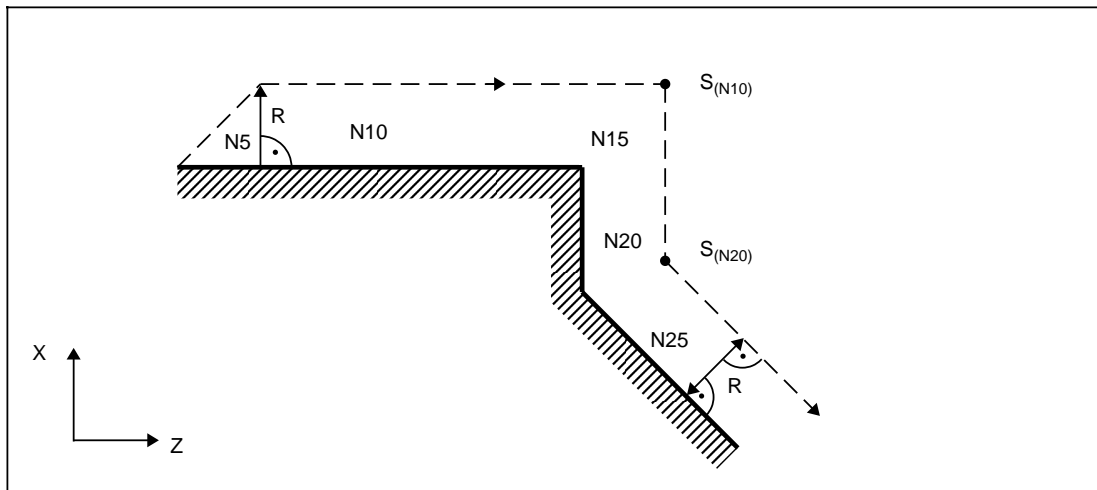


SRK: Ein "Hilfsfunktionssatz" und ein Satz mit "Strecke 0"

```
N5 G91 Z100 L_F
N10 M08 L_F
N15 Z0 L_F
N20 X-100 L_F
```

Der Satz **N10** wird am Punkt **S(N05)** ausgeführt. **Konturfehler** (gestricheltes Gebiet)

- Ein Satz "Nicht in der Korrektorebene" zwischen Strecken in der Korrektorebene.



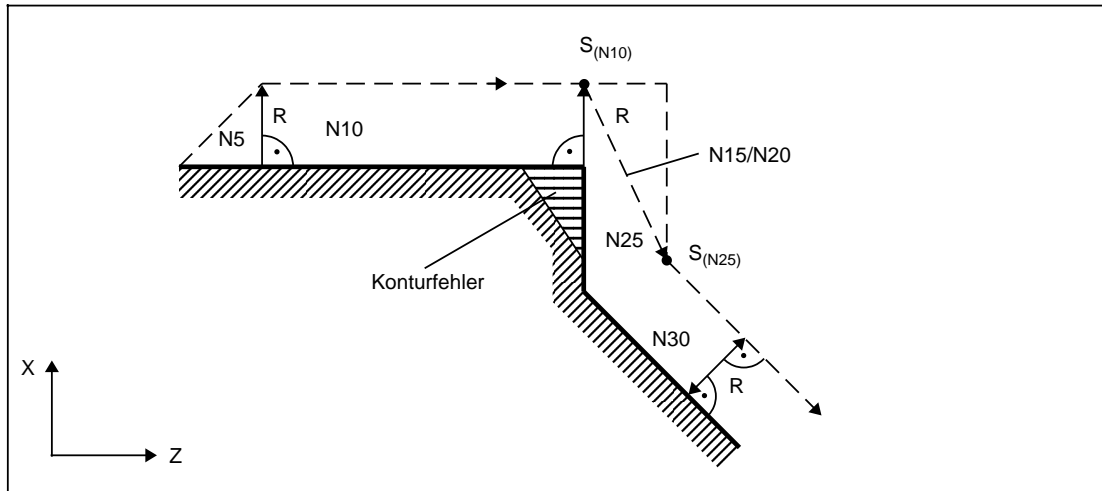
SRK: Ein Satz "Nicht in der Korrektorebene"

```
N5 G91 G41 G01 G17 Z100 F100 D01 L_F
N10 Z500 L_F
N15 Q500 L_F1)
N20 X-500 L_F
N25 Z1000 X-600 L_F
```

1) Satz nicht in Korrektorebene

keine Konturverletzung

- **Zwei Sätze "Nicht in der Korrektorebene"** zwischen Strecken in der Korrektorebene.



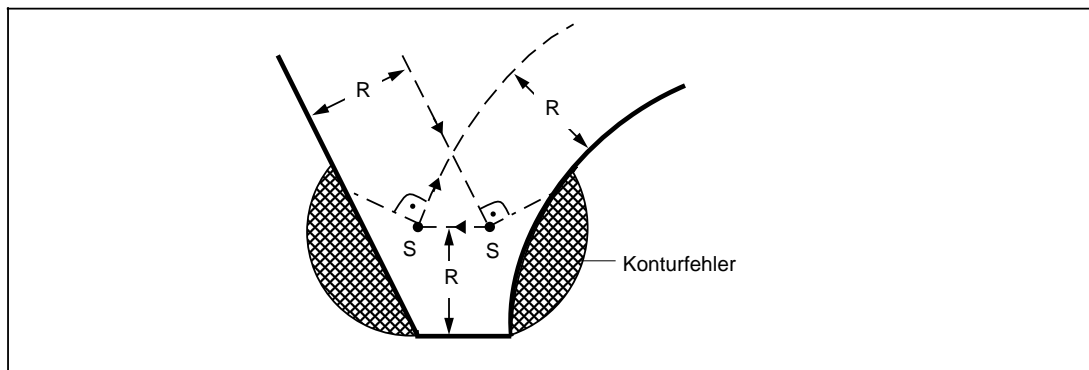
SRK: Zwei Sätze "Nicht in der Korrektorebene"

N5	G91	G41	G01	G17	Z100	F100	D01	L _F		
N10	Z500		L _F							
N15	Q500		L _F						Satz nicht in der Korrektorebene	} Konturverletzung (gestricheltes Gebiet)
N20	Q-500		L _F						Satz nicht in der Korrektorebene	
N25	X-500		L _F							
N30	Z1000	X-600	L _F							
Konturfehler (gestricheltes Gebiet)										

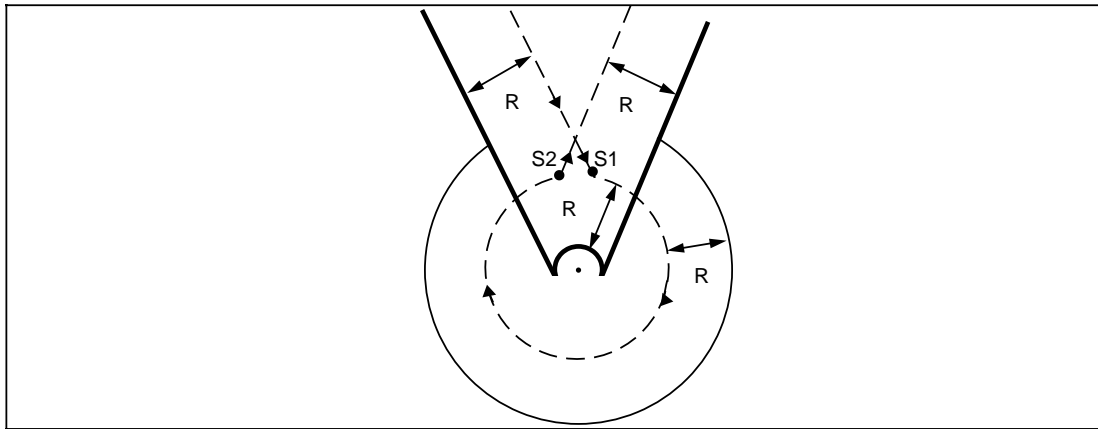
9.10 Sonderfälle der SRK

Zur **Schnittpunktberechnung der korrigierten Bahnen** benutzt die Steuerung immer den **nächsten** Satz. Falls im nächsten Satz keine Achsen der Korrektorebene programmiert sind, benutzt die Steuerung den **übernächsten** Satz. **Konturfehler** können auftreten, wenn der **Zwischensatz kleiner** ist als der angewählte Korrekturbetrag.

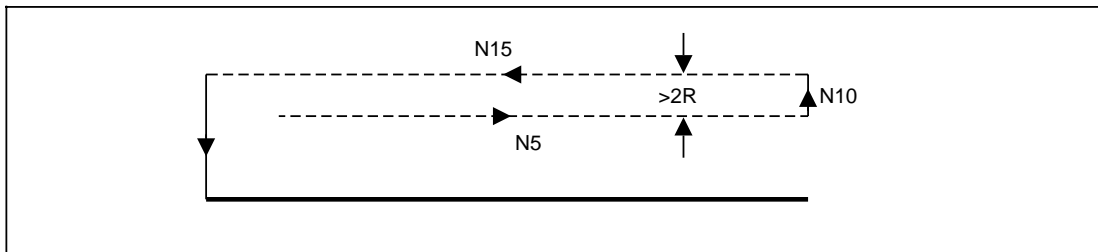
Die Bearbeitung wird nicht unterbrochen, aber es wird ein **Alarm angezeigt**.



SRK-Sonderfall: Zwischensatz < Korrekturbetrag



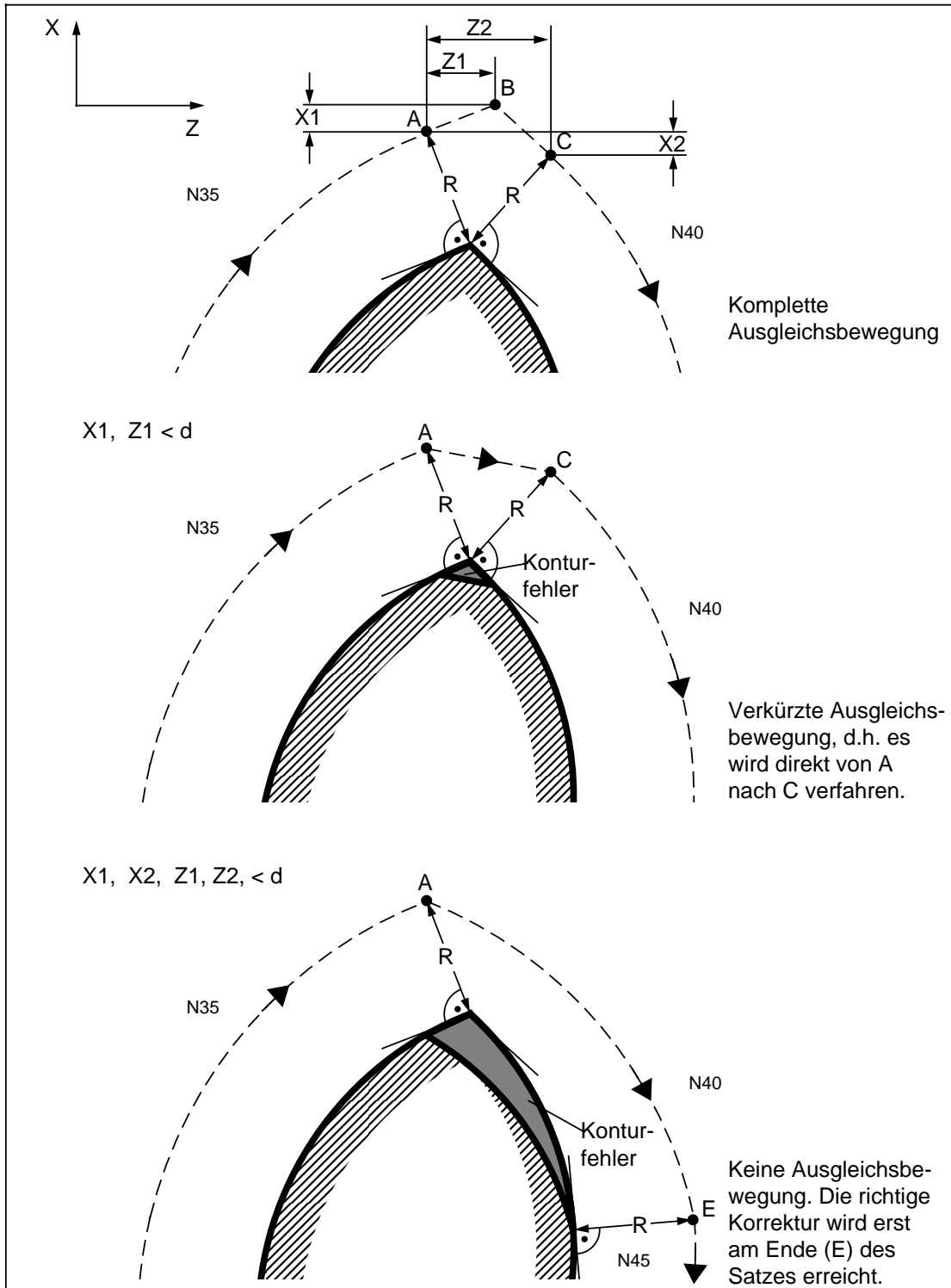
SRK-Sonderfall: Zwischensatz für Korrektur zu klein



SRK-Sonderfall: gleiche Korrekturrichtung und Verfahrrichtungswechsel

Die **Korrekturrichtung** der SRK bleibt beibehalten und die **Verfahrrichtung** wird umgekehrt. Der Rückzugsweg in N10 muß **größer** als der **doppelte** Schneidradius sein, andernfalls würde das **Werkzeug** eine Bewegung in die **falsche Richtung** ausführen.

Für eine Außenkontur mit Kreisübergängen und stumpfe Winkel gilt:

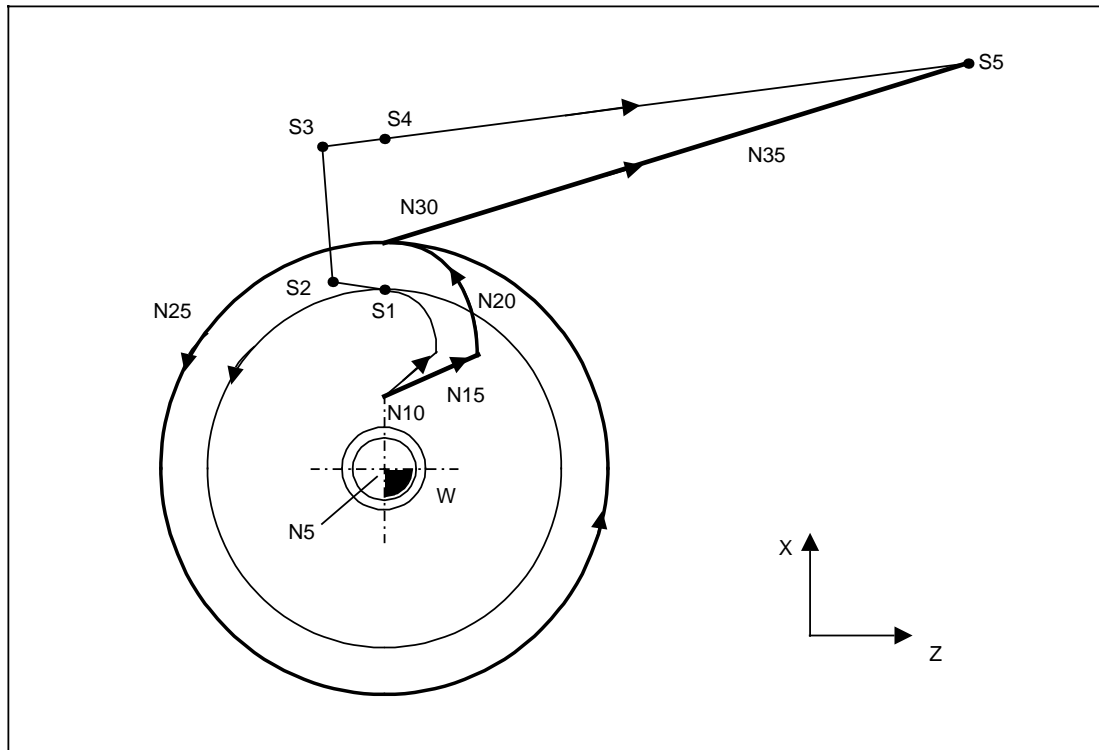


Um durch kleine Zwischensätze bedingtes Anhalten im Bahnsteuerbetrieb zu vermeiden, können die Wege AB und BC von der NC ausgelassen werden.

Wie die Bahn in diesem Fall verläuft ist abhängig von einer bei der Inbetriebnahme festgelegten Toleranz d (max. 32000 μm).

Die Satznummern werden vertauscht, wenn die SRK Zwischensätze generiert (auch bei An- und Abwahl) und dazwischen eine Achsbewegung außerhalb der Korrekturebene programmiert ist.

Die Steuerung generiert drei SRK-Zwischensätze zwischen den Punkten S1-S2-S3-S4; erst nach dem Heben der Schneide (S1) werden diese abgefahren.



```

N5   G00  Q100  L_F
N10  Z0   X10   L_F
N15  G41  D01   Z20  X20  L_F
N20  G03  Z0   X40  K-20  I0  L_F
N25  Z0   X40  K0-  I-40  L_F
N30  G01  Q50   L_F
N35  G40  Z80  X60  L_F
N40  M30  L_F
    
```

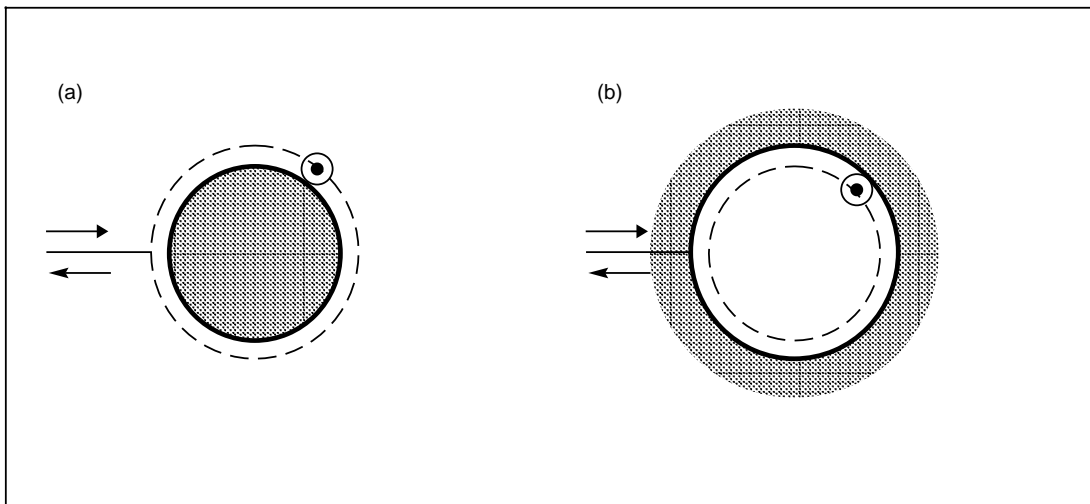
Die Punkte S1, S2, S3, S4 gehören logisch zum Satz N25. Die Bearbeitungsreihenfolge (im Einzelsatz sichtbar) ist:

..., N20, N25 (S1), N30 (Herausziehen des Werkzeuges aus dem Werkstück, Freifahren), N25 (S2), N25 (S3), N25 (S4), N35... . Das Verfahren gilt auch, wenn N25 ein Linearsatz ist.

9.11 Wirkung bei negativen Korrekturwerten

Bei einem negativen Korrekturwert, Schneidradius P4, wird mit G41 eine korrigierte Bahn ausgeführt, die G42 mit positivem Korrekturwert entspricht, d.h. statt der programmierten Außenkontur wird eine analoge Innenkontur gefahren und umgekehrt.

Bei der im nachstehenden Bild (a) dargestellten Schneiden-Mittelpunktsbahn ($- \bullet -$) wurde ein positiver Korrekturbetrag eingegeben.
Ein negativer Korrekturbetrag bei gleichem Bearbeitungsprogramm bewirkt die in Bild (b) gezeigte Bearbeitung.



Wird das Programm nach Bild (b) mit einem positiven Korrekturbetrag erstellt, dann ergibt sich bei negativem Korrekturbetrag eine Bearbeitung nach Bild (a). Die Ausführung wird durch die positive/negative Korrekturbetragseingabe unterschieden.

10 Zyklen

Die folgenden Bearbeitungszyklen stehen als fest abgespeicherte Unterprogramme im Anwenderspeicher-Modul (ASM) zur Verfügung:

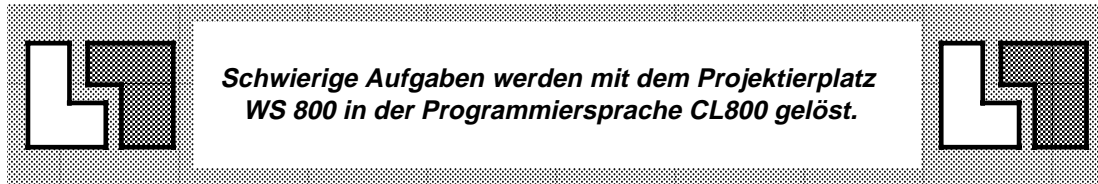
- L91/92 Rückzugszyklen für Werkzeugwechsel
- L93 Einstechzyklus
- L95 Abspannzyklus (Schruppen achsparallel)
- L97 Gewindeschneidzyklus
- L98 Tiefbohrzyklus
- L99 Ketten von Gewinden (Vier-Punkte-Gewindeschneidzyklus)

Diese Zyklen sind in einer getrennten Druckschrift "Programmieranleitung, SINUMERIK System 800, Zyklen" beschrieben.

11 Programmieren von Zyklen

11.1 Allgemeine Hinweise

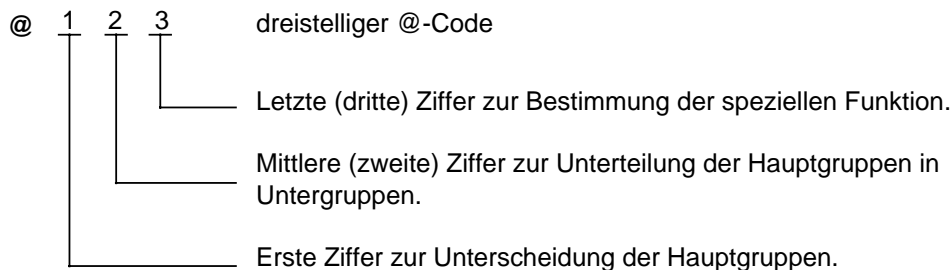
Mit Hilfe des @-Codes können Sie direkt an der Steuerung z. B. **Zyklen** erstellen.



Die folgenden Abschnitte sollen Ihnen eine Einführung in die **Programmierung** mit dem **@-Code** geben.

11.2 Zielcode

Der **dreistellige** @-Code ist ein Zielcode mit folgendem Aufbau:



11.2.1 Hauptgruppen

Den **acht Gruppen** des @-Codes sind folgende Funktionen zugeordnet:

- @0 . . . Allgemeine Anweisungen für den Programmaufbau
- @1 . . . Programmverzweigungen
- @2 . . . Datentransfer allgemein
- @3 . . . Datentransfer: Systemspeicher in R-Parameter
- @4 . . . Datentransfer: R-Parameter in Systemspeicher
- @6 . . . Mathematische und logische Funktionen
- @7 . . . NC-spezifische Funktionen
- @a . . . E/A-Funktionen

11.2.2 Operanden hinter dem Zielcode

Der @-Code benötigt ergänzende Angaben (Operanden) für seine Funktion.
Diese Operanden werden durch folgende **Buchstaben** definiert:

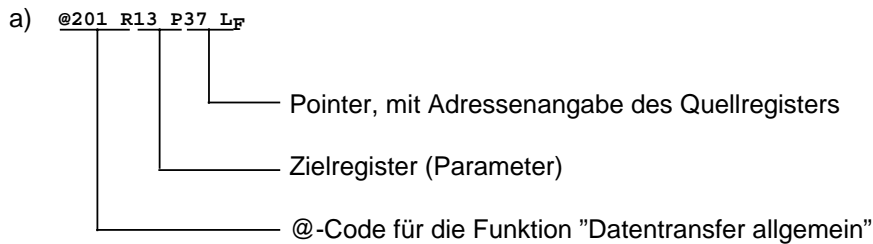
- K** Konstante
- R** R-Parameter (Register)
- P** Pointer (Zeiger).

Der durch die Konstante K definierte Wert liegt im Programm fest und kann nicht verändert werden (direkte Wertangabe).

Der in einem R-Parameter stehende Wert kann durch das Programm verändert werden (indirekte Wertangabe).

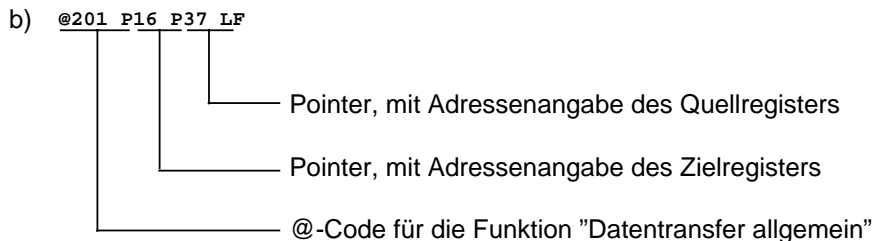
Der Pointer P weist auf einen Parameter hin, in dem die Adresse des Parameters steht, auf dessen Inhalt die Funktion angewendet werden soll (indirekte Wertangabe).

Beispiele für @-Code mit Operanden:



Erläuterung der Funktion zu Beispiel a):

Lade den Inhalt des Quellregisters, dessen Adresse im Register R37 steht, in das Zielregister R13.



Erläuterung der Funktion zu Beispiel b):

Lade den Inhalt des Quellregisters, dessen Adresse im Register R37 steht, in das Zielregister, dessen Adresse im Register R16 zu finden ist.

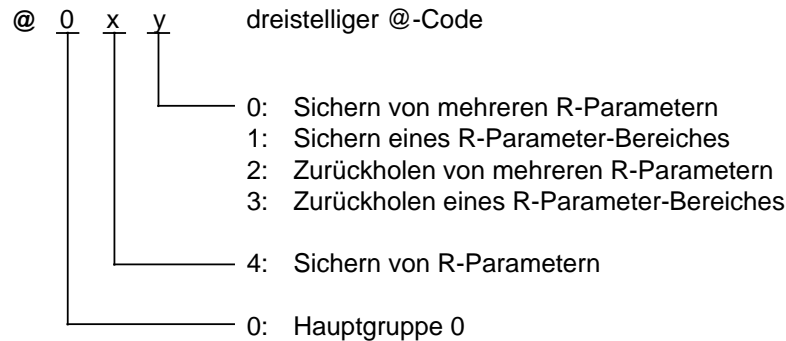
11.2.3 Notation

Der @-Code erfordert eine strenge Notation. In der Auflistung der einzelnen Befehle auf den nächsten Seiten folgen dem dreistelligen @-Code eine Reihe von Notationsangaben jeweils in "spitzen" Klammern. Die einzelnen Notationen haben folgende Bedeutung:

- <Const>** direkte Wertangabe (Konstante K)
- <R-Par>** indirekte Wertangabe (R-Parameter)
- <Var>** indirekte Wertangabe (R-Parameter oder Pointer)
- <Wert>** gemischte Wertangabe (Konstante, R-Parameter oder Pointer).

11.3 Allgemeine Anweisungen für den Programmaufbau

Die Hauptgruppe 0 ist folgendermaßen gegliedert:



Hauptgruppe 0/Untergruppe 0: Programm-Rahmenanweisungen

@00f

(Freigabe für Softkeystart)

Ist dieser Befehl alleine im ersten Satz eines Unterprogramms programmiert, so ist dieses für den Start mit Softkey freigegeben. Um Suchzeiten zu sparen, sollte der Befehl am Programmangfang stehen. Vom Werkzeugmaschinenhersteller erfahren Sie, wo der Softkey projektiert ist.

Hauptgruppe 0/Untergruppe 4: Sichern von R-Parametern

@040 <Const> <R-Par 1> . . . <R-Par n>

Sichern der R-Parameter auf Stack

Mit der Konstanten <Const> wird die Anzahl der folgenden R-Parameter angegeben, die zu dieser Funktion gehören. Die Inhalte der R-Parameter werden durch Übertragen in ein Stack-Register ab R300 gesichert.

@041 <R-Par 1> . . . <R-Par 2>

Sichern einer Gruppe von R-Parametern auf Stack

Es werden die Inhalte der R-Parameter im Bereich von <R-Par 1> bis <R-Par 2> durch Übertragen in ein Stack-Register ab R300 gesichert.

@042 <Const> <R-Par n> . . . <R-Par 1>

Gesicherte R-Parameter von Stack holen

Dieser Befehl entnimmt die gesicherten Werte aus dem Stack-Register und lädt sie in die angegebenen R-Parameter. Die R-Parameter sind in umgekehrter Reihenfolge gegenüber @040 aufzuführen.

@043 <R-Par 1> . . . <R-Par 2>

Gruppe der gesicherten R-Parameter von Stack holen

Die mit @041 gesicherten Werte werden in die R-Parameter zurückgeladen.

Diese Befehle der Hauptgruppe 0/Untergruppe 4 werden angewendet, wenn man in einem Unterprogramm mit R-Parametern arbeitet, die möglicherweise in einer höheren Ebene verwendet worden sind.

Man schreibt am Anfang eines Unterprogramms einen Push-Befehl (@040 oder @041), mit dem man die Werte sichert und die genannten R-Parameter mit dem Wert Null besetzt.

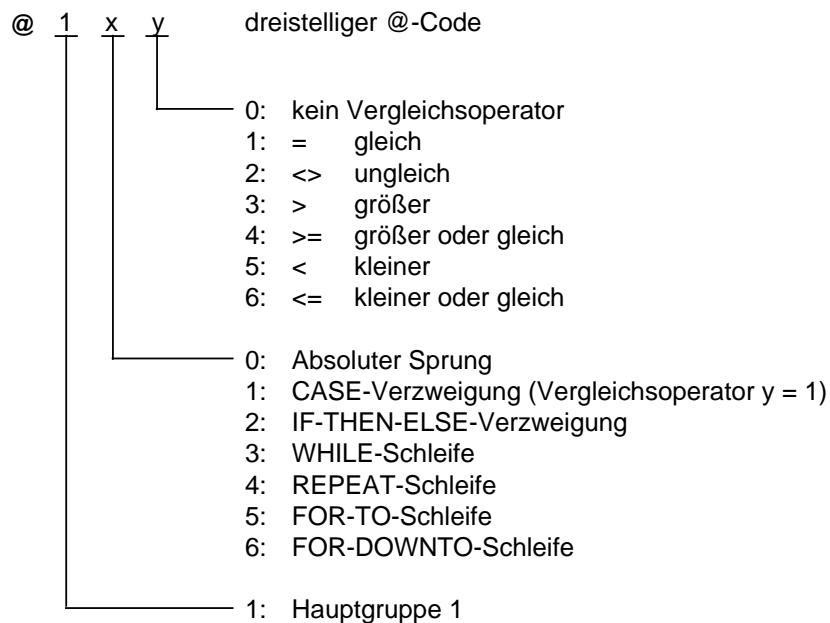
Am Ende des Unterprogramms wird mit einem Pop-Befehl (@042 oder @043) der ursprüngliche Zustand wiederhergestellt.

Beispiel für Anweisungsform im Programm:

L100 L _F	Aufruf Unterprogramm
@041 R61 R69 L _F	Die Inhalte der R-Parameter von R61 bis R69 werden in das Stack-Register übertragen und mit "0" vorbesetzt.
:	
:	
@043 R61 R69 L _F	Die gesicherten Werte werden in die Parameter R61 bis R69 zurückgeladen.
:	
:	
M17 L _F	Unterprogramm-Ende

11.4 Programmverzweigungen

Die Hauptgruppe 1 ist folgendermaßen gegliedert:



Hauptgruppe 1/Untergruppe 0: Absoluter Sprung

@100 <Const> oder @100 <R-Par>

Absoluter Sprung

Mit der Konstanten oder R-Parameter (keine Pointer erlaubt) wird das Sprungziel (Satznummer) und die Sprungrichtung angegeben. Eine positive Satznummer bedeutet, daß der anzuspringende Satz in Richtung Programmende liegt, bei einer negativen Satznummer ist er in Richtung Programmanfang zu suchen.

Weist das Vorzeichen in die falsche Richtung, so findet die Steuerung den Satz nicht, auch wenn er im Programm vorhanden ist (Alarm 3012: "Satz im Speicher nicht vorhanden").

Beispiele:

@100 K375 L_F

Absoluter Sprung auf Satz N375 in Richtung Programmende.

@100 K-150 L_F

Absoluter Sprung auf Satz N150 in Richtung Programmstart.

Hauptgruppe 1/Untergruppe 1: CASE-Verzweigung

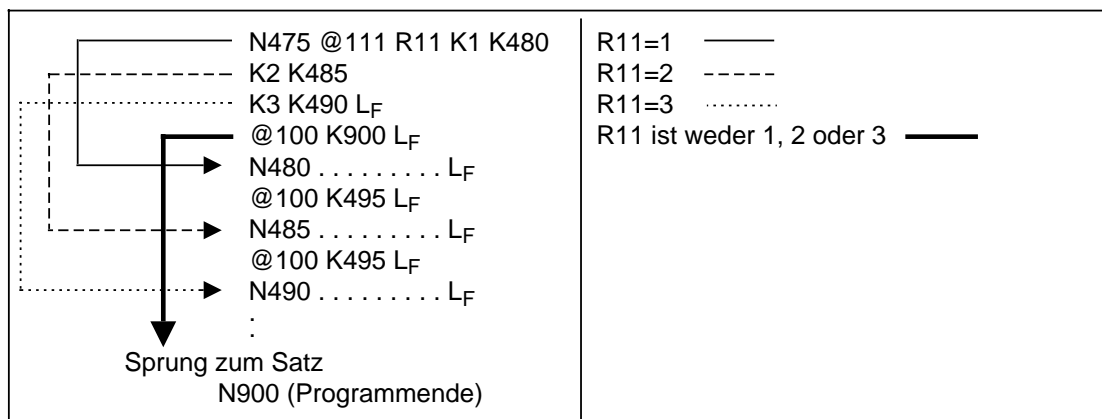
```
@111 <Var> <Wert 1> <Const 1>          CASE-Verzweigung
      <Wert 2> <Const 2>
      ⋮
      <Wert n> <Const n>
```

Die Notation <Var> wird nacheinander fortlaufend mit der Notation <Wert...> verglichen.

Ist die Vergleichsfunktion erfüllt, verzweigt das Programm zu der unter der Notation <Const ...> programmierten Satznummer.

Ist der Vergleich nicht erfüllt, so wird im Programm mit dem nächsten Satz fortgefahren, bei dem die Satznummer für die Programm-Fortsetzung programmiert ist (z.B. durch einen absoluten Sprung mit @100).

Beispiel für CASE-Verzweigung:



Einsatz einer CASE-Verzweigung bei einem Zyklus mit umschaltbaren Achsen

Erläuterung zum Beispiel:

Das Bild zeigt den Einsatz einer CASE-Verzweigung bei einem Zyklus mit umschaltbaren Achsen. Je nachdem, ob die Bearbeitungsebene durch Parametrierung von R11=1, 2 oder 3 festgelegt wurde, verzweigt das Programm in die Sätze N480 (R11=1), N485 (R11=2) oder N490 (R11=3). Steht in R11 keiner der drei Werte, so liegt offensichtlich ein Parametrierfehler vor.

Das Programm verzweigt zum Satz N900 (=Programmende).

Hauptgruppe 1/Untergruppe 2: IF-THEN-ELSE-Verzweigung

Eine IF-THEN-ELSE-Verzweigung sagt aus:

Wenn (IF) die Bedingung (dritte Stelle des @-Codes) erfüllt ist, dann (THEN) führe die in den nächsten Sätzen stehenden Anweisungen aus, sonst (ELSE) verzweige zu dem Satz, dessen Nummer durch die zuletzt stehende Konstante genannt wird.

Auch hier wirkt das Vorzeichen der Satznummer als Suchrichtung.

@ 121 <Var> <Wert> <Const> **Gleich**

Ist (IF) der mit der Notation <Var> definierte Zahlenwert gleich dem mit <Wert> definierten, so (THEN) wird das Programm mit dem nächsten Satz fortgesetzt. Andererseits (ELSE) wird in den mit der Konstanten bestimmten Satz gesprungen.

Beispiel:

@121 R13 R27 K375 L_F Fortsetzung des Programmes, wenn R13=R27, sonst bedingter Sprung auf Satz N375 in Richtung Programmende.

@ 122 <Var> <Wert> <Const> **Ungleich**

Ist (IF) der mit der Notation <Var> definierte Zahlenwert ungleich dem mit <Wert> definierten, so (THEN) wird das Programm mit dem nächsten Satz fortgesetzt. Andererseits (ELSE) wird in den mit der Konstanten bestimmten Satz gesprungen.

@ 123 <Var> <Wert> <Const> **">"**

Ist (IF) der mit der Notation <Var> definierte Zahlenwert größer als der mit <Wert> definierte, so (THEN) wird das Programm mit dem nächsten Satz fortgesetzt. Andererseits (ELSE) wird in den mit der Konstanten bestimmten Satz gesprungen.

Beispiel:

@123 R13 R27 K-150 L_F Fortsetzung des Programmes, wenn R13>R27, sonst bedingter Sprung auf Satz N150 in Richtung Programmanfang.

@ 124 <Var> <Wert> <Const> **">="**

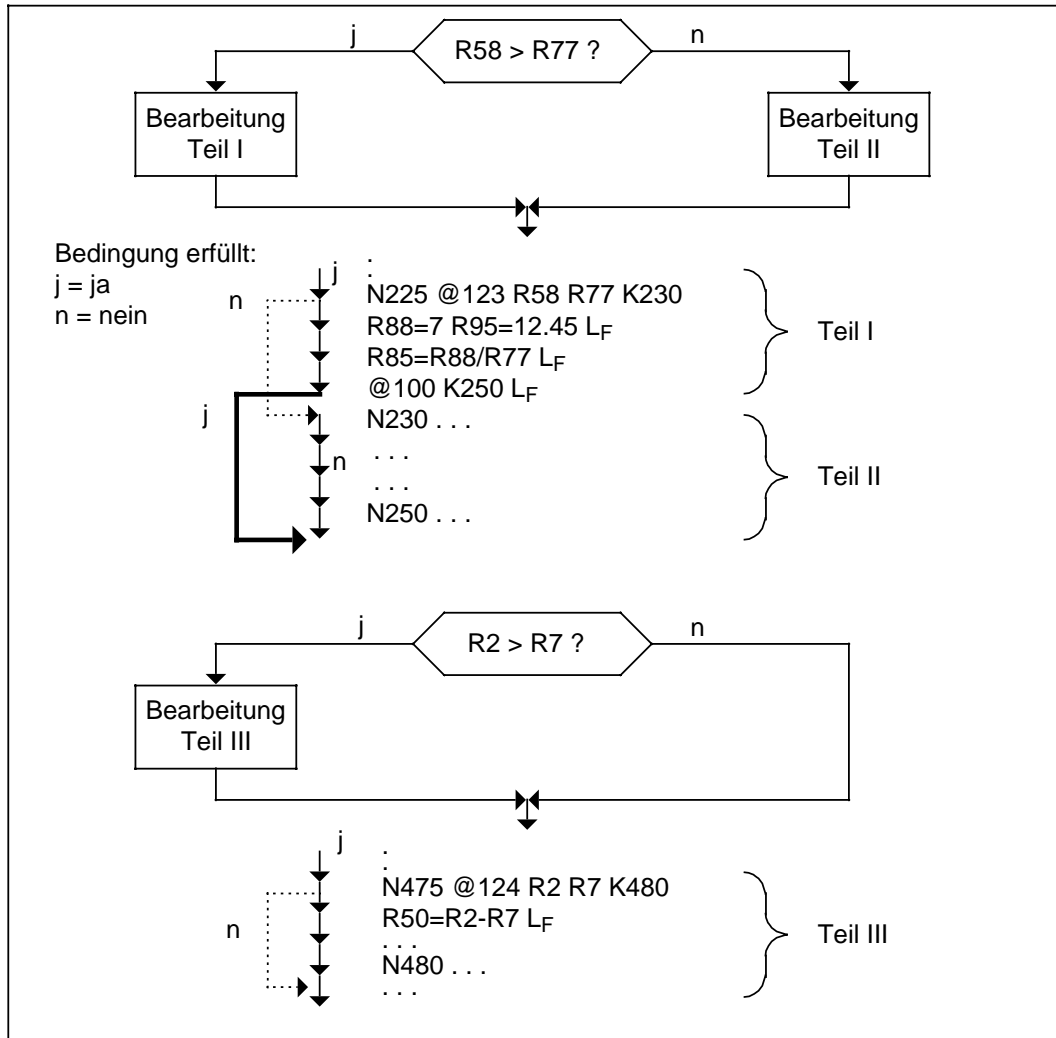
Ist (IF) der mit der Notation <Var> definierte Zahlenwert größer als der oder gleich dem mit <Wert> definierte, so (THEN) wird das Programm mit dem nächsten Satz fortgesetzt. Andererseits (ELSE) wird in den mit der Konstanten bestimmten Satz gesprungen.

@ 125 <Var> <Wert> <Const> **"<"**

Ist (IF) der mit der Notation <Var> definierte Zahlenwert kleiner als der mit <Wert> definierte, so (THEN) wird das Programm mit dem nächsten Satz fortgesetzt. Andererseits (ELSE) wird in den mit der Konstanten bestimmten Satz gesprungen.

@ 126 <Var> <Wert> <Const> **"<="**

Ist (IF) der mit der Notation <Var> definierte Zahlenwert kleiner als der oder gleich dem mit <Wert> definierte, so (THEN) wird das Programm mit dem nächsten Satz fortgesetzt. Andererseits (ELSE) wird in den mit der Konstanten bestimmten Satz gesprungen.

Beispiel: Programmablauf mit IF-THEN-ELSE-Verzweigungen**Erläuterung zum Beispiel:**

Der im Beispiel dargestellte Programmabschnitt aus einem Zyklus zeigt, wie man mit IF-THEN-ELSE-Verzweigungen Programmverzweigungen aufbauen kann. Ist im Satz N225 der Inhalt des Registers R58 größer als der Inhalt des Registers R77, dann werden die Anweisungen der nächsten Zeile ausgeführt.

Das Register R88 wird mit 7 geladen und das Register R95 mit 12.45. Ist jedoch R58 kleiner oder gleich R77, so verzweigt das Programm zum Satz N230. In der Zeile vor dem Satz N230 steht jedoch ein absoluter Sprung zum Satz N250.

Mit der IF-THEN-ELSE-Verzweigung im Satz N225 wird auf diese Weise bewirkt, daß entweder der Satz N225 oder der Programmabschnitt vom Satz N230 bis N250 bearbeitet wird.

Im unteren Beispiel fehlt der absolute Sprung, so daß entweder die im Satz N475 stehenden Anweisungen befolgt werden oder nicht. Dieser Programmteil kann also mit der IF-THEN-ELSE-Verzweigung @124 übersprungen werden.

Hauptgruppe 1/Untergruppe 3: WHILE-Schleife

@13y <Var> <Wert> <Const>

Die WHILE-Schleife ist eine Wiederholungsanweisung mit Abfrage der Wiederholungsbedingung am Anfang der Schleife. Die Vergleichsoperatoren entsprechen denen der IF-THEN-ELSE-Verzweigung. Solange der Vergleich erfüllt ist, wird der folgende Block bearbeitet. Am Ende des Blocks muß ein absoluter Sprung mit @100 <Const> programmiert sein, der zur Abfrage zurückführt.

Ist der Vergleich nicht erfüllt, so wird an den unter <Const> definierten Satz gesprungen, der i.a. hinter dem mit dem absoluten Sprung steht.

Die Besetzung von y ist entsprechend den Anweisungen @ 121 ...@ 126 vorzunehmen

Beispiele:

N300 @131 R13 R27 K375 L_F Fortsetzung der Schleife, solange Schleifenbedingung R13=R27 erfüllt ist.

:

@100 K-300 L_F

N375 . . L_F

N300 @133 R13 R27 K375 L_F Fortsetzung der Schleife, solange Schleifenbedingung R13>R27 erfüllt ist.

:

:

@100 K-300 L_F

N375 . . L_F

:

:

Hauptgruppe 1/Untergruppe 4: REPEAT-Schleife

@14y <Var> <Wert> <Const>

Die REPEAT-Schleife ist eine Wiederholungsanweisung mit Abfrage der Wiederholungsbedingung am Ende der Schleife. Die Vergleichsoperatoren entsprechen denen der IF-THEN-ELSE-Verzweigung. Solange der Vergleich nicht erfüllt ist, wird zu dem unter <Const> definierten Satz zurückgesprungen. Ist die Bedingung erfüllt, wird die Schleife verlassen und das Programm fortgesetzt.

Die Besetzung von y ist entsprechend den Anweisungen @ 121 ...@ 126 vorzunehmen.

Beispiele:

N400 . . L_F Wiederhole die folgenden Anweisungen, solange bis Bedingung R13=R27 erfüllt ist.

.

.

@141 R13 R27 K-400 L_F

N400 . . L_F Wiederhole die folgenden Anweisungen, solange bis Bedingung R13>R27 erfüllt ist.

.

.

@143 R13 R27 K-400 L_F

:

:

Hauptgruppe 1/Untergruppe 5: FOR-TO-Schleife

@151 <Var> <Wert 2> <Const>

Die FOR-TO-Schleife ist eine Zählschleife, in der der Inhalt des unter <Var> definierten R-Parameters bei jedem Durchlauf inkrementiert wird. Die Abfrage auf "gleich" erfolgt am Anfang der Schleife. Solange eine Ungleichheit besteht, wird die Schleife bearbeitet, ansonsten wird in den unter <Const> definierten Satz gesprungen. Am Ende der Schleife muß die Variable <Var> inkrementiert (@620) und mit einem absoluten Sprung zum Anfang der Schleife zurückgesprungen werden.

Beispiel:

R5=1 R51=5 R52=10 L _F	Wertzuweisung für R5, R51, R52
@201 R50 P51 L _F	Datentransfer von R5 nach R50
N500 @151 R50 R52 K505 L _F	Anfang der FOR-TO-Schleife
:	
:	
@620 R50 L _F	
@100 K-500 L _F	
N505 ...	

Hauptgruppe 1/Untergruppe 6: FOR-DOWNTO-Schleife

@161 <Var> <Wert 2> <Const>

Die FOR-DOWNTO-Schleife ist eine Zählschleife, in der der Inhalt des unter <Var> definierten R-Parameters bei jedem Durchlauf dekrementiert wird. Die Abfrage auf "gleich" erfolgt am Anfang der Schleife. Solange eine Ungleichheit besteht, wird die Schleife bearbeitet, ansonsten wird in den unter <Const> definierten Satz gesprungen.

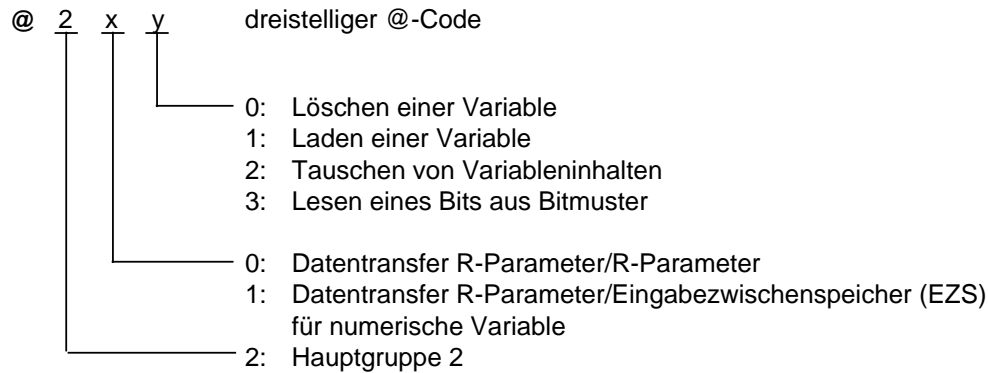
Am Ende der Schleife muß die Variable <Var> dekrementiert (@621) und mit einem absoluten Sprung zum Anfang der Schleife zurückgesprungen werden.

Beispiel:

R5=1 R51=5 R52=1 L _F	Wertzuweisung für Parameter R5, R51, R52
@201 R50 P51 L _F	Datentransfer von R5 nach R50
N600 @161 R50 R52 K605 L _F	Anfang der FOR-DOWNTO-Schleife
:	
:	
@621 R50 L _F	
@100 K-600 L _F	
N605	

11.5 Datentransfer allgemein

Die Hauptgruppe 2 ist folgendermaßen gegliedert:



Hauptgruppe 2/Untergruppe 0: Datentransfer R-Parameter/R-Parameter

@200 <Var>

Lösche Variable

Der Wert des mit der Notation <Var> definierten R-Parameter wird gelöscht.

@201 <Var> <Wert>

Lade Variable mit Wert

Der unter <Wert> definierte Zahlenwert wird in den unter <Var> definierten R-Parameter geladen.

@202 <Var 1> <Var 2>

Tauschen der Variableninhalte

Die Inhalte der beiden unter <Var1> und <Var 2> definierten R-Parameter werden getauscht.

@203 <Var 1> <Var 2> <Const>

Lesen eines Bit aus Bitmuster

Das durch <Const> definierte Bit des in <Var 2> stehenden Bitmusters wird in <Var 1> eingelesen.

Hauptgruppe 2/Untergruppe 1: Datentransfer R-Parameter/EZS für numerische Variable

@210 <Wert 3> <Wert 4>

Lösche Eingabezwischenspeicher

Der Eingabezwischenspeicher (EZS) wird gelöscht.
<Wert 3> Anfangsadresse
<Wert 4> Endadresse.

@211 <Var > <Wert 1>

Eingabezwischenspeicher lesen

Der R-Parameter <Var> wird mit dem Inhalt der EZS-Zelle <Wert 1> geladen.

Beispiel:

@211 R50 K101 LF

Der Inhalt der EZS-Zelle 101 wird in den Parameter R50 gelesen.

@212 <Wert 1 > <Wert>**Eingabezwischenspeicher
schreiben**

Die EZS-Zelle <Wert 1> wird mit der numerischen Größe <Wert> geladen.

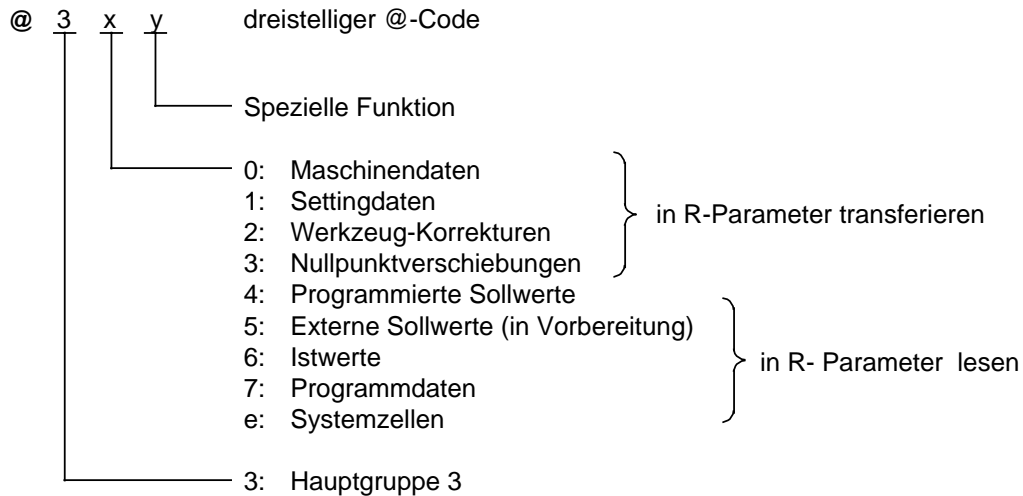
Beispiel:

@212 K102 K5 LF

In die EZS-Zelle 102 wird der Wert 5 geschrieben.

11.6 Datentransfer: Systemspeicher in R-Parameter

Die Hauptgruppe 3 ist folgendermaßen gegliedert:



Alle @-Befehle der Hauptgruppe 3 haben als erste Notation <Var>. Damit wird direkt oder über einen Pointer ein R-Parameter definiert, in den der Inhalt der angesprochenen Systemzelle zu laden ist.

Hauptgruppe 3/Untergruppe 0: Maschinendaten in R-Parameter transferieren

@300 <Var> <Wert 1>**Maschinendaten-NC**

Unter <Wert 1> wird die Adresse eines NC-Maschinendatums definiert.
Adreßbereich: 0 bis 4999.

Beispiel:

@300 R50 K2240 LF

Im Parameter R50 steht der Wert des 1. Software-Endschalters in Plusrichtung für die 1. Achse

@301 <Var> <Wert 1>**Maschinendaten-NC-Byte**

Unter <Wert 1> wird die Byte-Adresse eines NC-Maschinendatums definiert.
Adreßbereich: 5000 bis 6999.

@302 <Var> <Wert 1> <Wert 2> Maschinendaten-NC-Bit

Unter <Wert 1> wird die Byte-Adresse eines NC-Maschinendatenbits definiert.
Adreßbereich: 5000 bis 6999. Die Bit-Adresse (0 bis 7) steht unter <Wert 2>.

@306 <Var> <Wert 1> Maschinendaten-PLC

Unter <Wert 1> wird die Adresse eines PLC-Maschinendatums definiert.
Adreßbereich: 0 bis 1999.

@307 <Var> <Wert 1> Maschinendaten-PLC-Byte

Unter <Wert 1> wird die Byte-Adresse eines PLC-Maschinendatums definiert.
Adreßbereich: 2000 bis 3999.

@308 <Var> <Wert 1> <Wert 2> Maschinendaten-PLC-Bit

Unter <Wert 1> wird die Byte-Adresse eines PLC-Maschinendatenbits definiert.
Adreßbereich: 2000 bis 3999. Die Bit-Adresse (0 bis 7) steht unter <Wert 2>.

Hauptgruppe 3/Untergruppe 1: Settingdaten in R-Parameter transferieren

@310 <Var> <Wert 1> Settingdaten-NC

Unter <Wert 1> wird die Adresse eines Setting-Datums definiert.
Adreßbereich: 0 bis 4999.

@311 <Var> <Wert 1> Settingdaten-NC-Byte

Unter <Wert 1> wird die Byte-Adresse eines Setting-Datums definiert.
Adreßbereich: 5000 bis 9999.

@312 <Var> <Wert 1> <Wert 2> Settingdaten-NC-Bit

Unter <Wert 1> wird die Byte-Adresse eines Setting-Datenbits definiert.
Adreßbereich: 5000 bis 9999. Die Bit-Adresse (0 bis 7) steht unter <Wert 2>.

Hauptgruppe 3/Untergruppe 2: Werkzeug-Korrekturen in R-Parameter transferieren

@320 <Var> <Wert 1> <Wert 2> <Wert 3> Werkzeug-Korrektur

Mit diesem Befehl können die einzelnen Korrekturwerte aus dem Werkzeugkorrektur-Speicher in den Parameter unter der Notation <Var> gelesen werden.

Die Notationen <Wert 1> bis <Wert 3> sind wie folgt zu besetzen:

<Wert 1> TO-Bereich
 Bereich: 0.

<Wert 2> Werkzeug-Korrekturnummer (D-Nummer)
 Bereich: 1 bis 99.

<Wert 3> Nummer des Werkzeug-Korrekturspeichers (P-Nummer)
 Bereich: 0 bis 9.

Beispiel:

@320 R67 K0 K14 K2 L_P

Lesen des Korrekturwertes P2 (Geometrie, Länge 1) der Werkzeugkorrekturnummer D14 für den TO-Bereich 0 in den Parameter R67.

Hauptgruppe 3/Untergruppe 3: Nullpunktverschiebungen in R-Parameter transferieren

@330 <Var> <Wert 1> <Wert 2> <Wert 3>

Einstellbare Nullpunktverschiebung

Die Notationen <Wert 1> bis <Wert 3> sind wie folgt zu besetzen:

- <Wert 1> Gruppe der einstellbaren Nullpunktverschiebungen (G54 = 1 bis G57 = 4).
- <Wert 2> Nummer der Achse.
- <Wert 3> Grob- oder Feinwert (0 oder 1).

Beispiel:

@330 R81 K1 K2 K0 L_F

In den Parameter R81 wird der grobe Wert der 1. einstellbaren Nullpunktverschiebung (G54) der 2. Achse gelesen.

@331 <Var> <Wert 1> <Wert 2>

Programmierbare Nullpunktverschiebung

Es bedeuten:

- <Wert 1> Gruppe der programmierbaren additiven Nullpunktverschiebungen (G58 = 1 und G59 = 2).
- <Wert 2> Nummer der Achse.

@332 <Var> <Wert 2>

Externe Nullpunktverschiebung von der PLC

Unter <Wert 2> wird die Nummer der Achse der von der PLC eingegebenen externen Nullpunktverschiebung definiert.

@333 <Var> <Wert 2>

DRF-Verschiebung

Unter <Wert 2> wird die Nummer der Achse der DRF-Verschiebung definiert.

@334 <Var> <Wert 2>

PRESET-Verschiebung

Unter <Wert 2> wird die Nummer der Achse der PRESET-Verschiebung definiert.

@336 <Var> <Wert 2>

Summen-Verschiebung

Unter <Wert 2> wird die Nummer der Achse der Summen-Verschiebung definiert. Die Summenverschiebung beinhaltet:

- die angewählte einstellbare Nullpunktverschiebung
- die programmierbare Nullpunktverschiebung
- die externe Nullpunktverschiebung
- die angewählte Werkzeug-Korrektur.

PRESET- und DRF-Verschiebungen werden nicht berücksichtigt.

@337 <Var> <Wert 1> <Wert 2> <Wert 3>

Einstellbare Koordinatendrehung

In den Parameter <Var> kann der Drehwinkel der einstellbaren Koordinatendrehung gelesen werden.

Die Notationen <Wert 1> bis <Wert 3> sind wie folgt zu besetzen:

- <Wert 1> Nummer des Kanals (0 = eigener Kanal).
- <Wert 2> Gruppe der einstellbaren Koordinatendrehungen (G54 = 1 bis G57 = 4).
- <Wert 3> Nummer des Winkels (z.Zt. = 1).

@338 <Var> <Wert 1> <Wert 2> <Wert 3>

Programmierbare Koordinatendrehung

In den Parameter <Var> kann der Drehwinkel der programmierbaren Koordinatendrehung gelesen werden.

Es bedeuten:

- <Wert 1> Nummer des Kanals (0 = eigener Kanal).
- <Wert 2> Gruppe der programmierbaren Koordinatendrehungen (G58 = 1 und G59 = 2).
- <Wert 3> Nummer des Winkels (z.Zt. = 1).

Hauptgruppe 3/Untergruppe 4: Programmierte Sollwerte in R-Parameter lesen

@342 <Var> <Wert 1> <Wert 3>

Programmierte Spindeldrehzahl

In den Parameter <Var> wird die programmierte Spindeldrehzahl aus einem Kanal gelesen.

Die Notationen sind wie folgt zu besetzen:

- <Wert 1> Nummer des Kanals (0= eigener Kanal).
- <Wert 3> Spindelnummer (0= Nummer der Leitspindel).

@345 <Var> <Wert 1> <Wert 2>

Programmierte Schnittgeschwindigkeit

Mit diesem Befehl kann man die unter G96 programmierte Schnittgeschwindigkeit in einen R-Parameter übertragen.

Unter <Wert 1> wird die Kanalnummer eingetragen (eigener Kanal=„0“).

Der <Wert 2> wird mit „0“ besetzt.

Hauptgruppe 3/Untergruppe 6: Istwerte in R-Parameter lesen

Vor den Befehlen @360 bis @36b muß ein eigener Satz mit @714 programmiert werden. Um Werte aus einem anderen als dem eigenen Kanal lesen zu können, muß auch in diesem NC-Start gegeben werden.

@360 <Var> <Wert 2>

Achs-Position-Ist werkstückbezogen

Unter <Wert 2> ist die Achse zu definieren, deren werkstückbezogener Istwert in den R-Parameter zu übertragen ist.

Beispiel:

@360 R54 K2 L_F

Lesen des Positions-Istwertes der 2. Achse , bezogen auf den Werkstück-Nullpunkt und Eintragen in den Parameter R54.

@361 <Var> <Wert 2>

Achs-Position-Ist maschinenbezogen

Unter <Wert 2> ist die Achse zu definieren, deren maschinenbezogener Istwert in den R-Parameter zu übertragen ist.

In der Simulation liefern @360 und @361 dieselben Werte.

@363 <Var> <Wert 2>**Spindel-Position-Ist**

Unter <Wert 2> ist die Spindel zu definieren, deren Istposition in den R-Parameter zu übertragen ist.

@364 <Var> <Wert 2>**Spindel-Drehzahl-Ist**

Unter <Wert 2> ist die Spindel zu definieren, deren Istdrehzahl in den R-Parameter zu betragen ist.

@367 <Var> <Wert 1>**Achsnummer der aktuellen Ebene/ Leitspindelnummer**

Unter <Var> ist der R-Parameter zu definieren, ab dem die Achsnummern der mit G16 angewählten Ebene (4 Werte) eingetragen werden. In dem fünften R-Parameter wird die Nummer der Leitspindel abgelegt. Unter <Wert 1> ist die Kanal-Nummer festzulegen (0= eigener Kanal).

Folgende R-Parameter werden geladen:

Rn	Nummer der waagrechten Achse
Rn+1	Nummer der senkrechten Achse
Rn+2	Nummer der Achse senkrecht zur Ebene
Rn+3	Nummer der Achse in der Länge 2 wirkt (Werkzeugtyp 30, bei M-Version)
Rn+4	Nummer der Leitspindel.

Wird mit G16 XYZ- eine Werkzeuglängenkorrektur in negativer Richtung programmiert, so wird mit dem @367 bei der Z-Achse der Wert 128 addiert, um das Minuszeichen zu erkennen.

Beispiel:

@367 R50 K1 L_F

Die Daten der aktuellen Ebene und Spindelnummer werden im Kanal 1 gelesen und ab R50 hinterlegt.

@36a <Var> <Wert 1>**D-Funktion-Ist**

Mit diesem Befehl wird die Nummer der angewählten Werkzeug-Korrektur (D-Nummer) in den R-Parameter übertragen. Unter <Wert 1> ist dabei der Kanal anzugeben (eigener Kanal = 0). Bei der SINUMERIK 810T ist "0" einzugeben.

Anmerkung: "a" im @-Code steht für Sedezimal "A" (=10).

@36b <Var> <Wert 1> <Wert 3>**G-Funktion-Ist**

In den Parameter <Var> wird aus dem Arbeitsspeicher die G-Funktion des gerade sich in Bearbeitung befindlichen Teileprogrammsatzes gelesen. Durch <Wert 1> wird die Kanal-Nr. angegeben. Bei Vorgabe einer 0 wird im eigenen Kanal gelesen. <Wert 3> definiert die **interne** G-Gruppe, zu der die aktuelle G-Funktion gehört. Eine Tabelle mit der **internen** G-Gruppeneinteilung (Übersicht für @36b) befindet sich im Kapitel 12.

Beispiel:

@36b R50 K0 K0 L_F

Es wird im eigenen Kanal die aktuelle G-Funktion der ersten internen G-Gruppe (Gruppe 0) in den Parameter R50 gelesen.

Hauptgruppe 3/Untergruppe 7: Programmdaten in R-Parameter lesen

@371 <Var> <Wert 1> <Wert 3>

Sonderbits

Mit diesem Befehl werden Sonderbits zur Erfassung verschiedener aktiver Signale ausgelesen.

Kanalabhängige Bits: Bit 0 = Satzvorlauf aktiv
Bit 1 = Probelaufvorschub aktiv
Bit 2 = Simulation aktiv.

Im <Wert 1> ist die Kanalnummer (eigener Kanal = 0) einzutragen. <Wert 3> enthält die Bit-Nummer.

Kanalunabhängige Bits: Bit 0 = Meßeingang 1 aktiv
Bit 1 = Meßeingang 2 aktiv.

In <Wert 1> ist 99 einzugeben. <Wert 3> enthält die Bit-Nummer.

Beispiel:

@371 R81 K0 K1 L_F

Der Zustand des Sonderbits für "Probelaufvorschub" im eigenen Kanal wird in den Parameter R81 gelesen.

Hauptgruppe 3 / Untergruppe 8: PLC-Signale Bits in R-Parameter lesen

@380 <Var> <Wert 1> <Wert 2> Wert 3>

Der Zustand eines Eingangsbits in der PLC wird in den mit <Var> definierten Parameter gelesen. Die PLC-Nummer wird durch den <Wert 1>, die Byteadresse durch <Wert 2> und die Bitadresse durch <Wert 3> definiert. Die PLC-Nummer muß mit K1 versehen werden.

Beispiel:

@380 R50 K1 K2 K0 L_F

Der Zustand des definierten Eingangsbits wird in R50 gelesen.

@381 <Var> <Wert 1> <Wert 2> <Wert 3>

Der Zustand eines Ausgangsbits in der PLC wird in den mit <Var> definierten Parameter gelesen. Die PLC-Nummer wird durch den <Wert 1>, die Byteadresse durch <Wert 2> und die Bitadresse durch <Wert 3> definiert. Die PLC-Nummer muß mit K1 versehen werden.

@382 <Var> <Wert 1> <Wert 2> <Wert 3>

Der Zustand eines Merkerbits in der PLC wird in den mit <Var> definierten Parameter gelesen. Die PLC-Nummer wird durch den <Wert 1>, die Byteadresse durch <Wert 2> und die Bitadresse durch <Wert 3> definiert. Die PLC-Nummer muß mit K1 versehen werden.

@ 383 <Var> <Wert 1> <Wert 2> <Wert 3> <Wert 4>

Der Zustand eines PLC-Datenwortbits in der PLC wird in den mit <Var> definierten Parameter gelesen. Die PLC-Nummer wird durch den <Wert 1>, die Nummer des DB oder DX durch <Wert 2>, die Datenwortnummer durch <Wert 3> und die Bitadresse durch <Wert 4> definiert. Die PLC-Nummer muß mit K1 versehen werden.

Beispiel:

@383 R51 K1 K2 K4 K2

Der Zustand des definierten PLC-Datenwortbits wird in R51 gelesen.

Hinweis:

Im 3. Kanal (Simulation) werden @380 bis @383 überlesen. Dies kann zu einem veränderten Programmablauf führen.

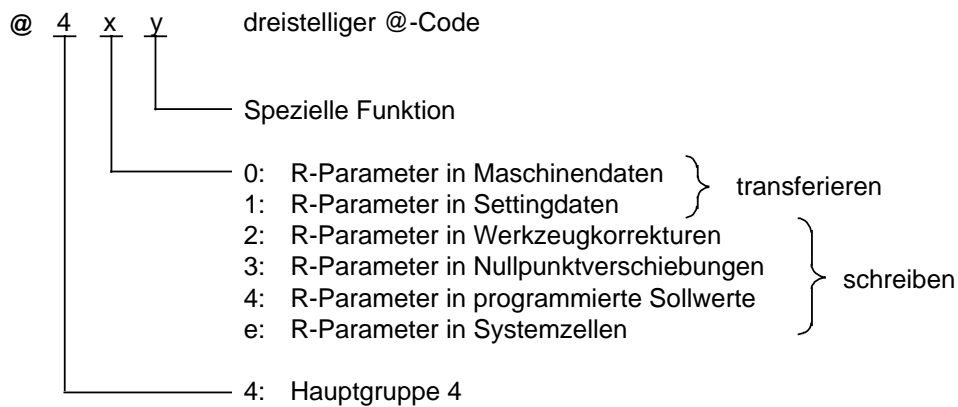
Hauptgruppe 3 / Untergruppe e: Systemzellen in R-Parameter lesen

@ 3e4 <Var> <Wert 1>**Aktive Getriebestufe lesen**

In den Parameter <Var> wird abhängig von der Spindelnummer <Wert 1> die aktive Getriebestufe gelesen. Ist die Spindelnummer 0, so wird als Spindelnummer die Nummer der Leitspindel verwendet.

11.7 Datentransfer: R-Parameter in Systemspeicher

Die Hauptgruppe 4 ist folgendermaßen gegliedert:



Alle @-Befehle dieser Hauptgruppe haben als letzte Notation <Wert>. Damit wird direkt mit einer Konstanten oder indirekt über einen R-Parameter oder einen Pointer der zu transferierende Zahlenwert definiert.

Hauptgruppe 4/Untergruppe 0: R-Parameter in Maschinendaten transferieren

@400 <Wert 1> <Wert>

Maschinendaten-NC

Unter <Wert 1> wird die Adresse eines NC-Maschinendatums definiert.
Adreßbereich: 0 bis 4999.

Beispiel:

@400 K2241 R90 L_F

Das Maschinendatum des 1. Software-Endschalters für die 2. Achse in Plusrichtung wird über den Parameter R90 geladen.

@401 <Wert 1> <Wert>

Maschinendaten-NC-Byte

Unter <Wert 1> wird die Byte-Adresse eines NC-Maschinendatenbits definiert.
Adreßbereich: 5000 bis 6999.

@402 <Wert 1> <Wert 2> <Wert>

Maschinendaten-NC-Bit

Unter <Wert 1> wird die Byte-Adresse eines NC-Maschinendatenbits definiert.
Adreßbereich: 5000 bis 6999. Die Bit-Adresse (0 bis 7) steht unter <Wert 2>.

@406 <Wert 1> <Wert>

Maschinendaten-PLC

Unter <Wert 1> wird die Adresse eines PLC-Maschinendatums definiert.
Adreßbereich: 0 bis 1999.

@407 <Wert 1> <Wert>

Maschinendaten-PLC-Byte

Unter <Wert 1> wird die Byte-Adresse eines PLC-Maschinendatums definiert.
Adreßbereich: 2000 bis 3999.

@408 <Wert 1> <Wert 2> <Wert>

Maschinendaten-PLC-Bit

Unter <Wert 1> wird die Byte-Adresse eines PLC-Maschinendatenbits definiert.
Adreßbereich: 2000 bis 3999. Die Bit-Adresse (0 bis 7) steht unter <Wert 2>.

Hauptgruppe 4/Untergruppe 1: R-Parameter in Settingdaten transferieren

@410 <Wert 1> <Wert>

Settingdaten-NC

Unter <Wert 1> wird die Adresse eines Setting-Datums definiert.
Adreßbereich: 0 bis 4999.

@411 <Wert 1> <Wert>

Settingdaten-NC-Byte

Unter <Wert 1> wird die Byte-Adresse eines Setting-Datenbits definiert.
Adreßbereich: 5000 bis 9999.

@412 <Wert 1> <Wert 2> <Wert>

Settingdaten-NC-Bit

Unter <Wert 1> wird die Byte-Adresse eines Setting-Datenbits definiert.
Adreßbereich: 5000 bis 9999. Die Bit-Adresse (0 bis 7) steht unter <Wert 2>.

Hauptgruppe 4/Untergruppe 2: R-Parameter in Werkzeug-Korrekturen schreiben

@420 <Wert 1> <Wert 2> <Wert 3> <Wert> **Werkzeug-Korrektur**

Der Zahlenwert wird in den Werkzeug-Korrekturspeicher eingetragen. Der vorhandene Inhalt des Speichers wird dabei überschrieben.

Die Notationen <Wert 1> bis <Wert 3> sind wie folgt zu besetzen:

- <Wert 1> TO-Bereich
Bereich: 0.
- <Wert 2> Werkzeug-Korrekturnummer (D-Nummer)
Bereich: 1 bis 99.
- <Wert 3> Nummer des Werkzeug-Korrekturwertes (P-Nummer)
Bereich: 0 bis 9.

Beispiel:

@420 K0 K2 K3 R80 L_F

Im TO-Bereich 0 wird der Korrektur-Speicher D2 unter Werkzeug-Korrekturwert P3 mit dem Inhalt von R80 geladen.

@423 <Wert 1> <Wert 2> <Wert 3> <Wert> **Werkzeug-Korrektur additiv**

Der Zahlenwert wird zu dem im Werkzeug-Korrekturspeicher stehenden Wert addiert.

Die Notationen <Wert 1> bis <Wert 3> sind wie folgt zu besetzen:

- <Wert 1> TO-Bereich
Bereich: 0.
- <Wert 2> Werkzeug-Korrekturnummer (D-Nummer)
Bereich: 1 bis 99.
- <Wert 3> Nummer des Werkzeug-Korrekturwertes (P-Nummer)
Bereich: 0 bis 9.

Hauptgruppe 4/Untergruppe 3: R-Parameter in Nullpunktverschiebungen schreiben

@430 <Wert 1> <Wert 2> <Wert 3> <Wert> **Einstellbare Nullpunktverschiebung**

Der Zahlenwert wird in den Nullpunktverschiebungs-Speicher eingetragen. Der vorhandene Inhalt wird dabei überschrieben.

Die Notationen <Wert 1> bis <Wert 3> sind wie folgt zu besetzen:

- <Wert 1> Gruppe der einstellbaren Nullpunktverschiebungen (G54=1 bis G 57=4).
- <Wert 2> Nummer der Achse.
- <Wert 3> Grob- oder Feinwert (0 oder 1).

Beispiel:

@430 K1 K2 K0 K500 L_F

Der grobe Wert der 1. einstellbaren Nullpunktverschiebung wird in der 2. Achse mit der Konstanten 500 geladen.

@431 <Wert 1> <Wert 2> <Wert 3> <Wert>

**Einstellbare
Nullpunktverschiebung additiv**

Der Zahlenwert wird zu dem im Nullpunktverschiebungs-Speicher stehenden Wert addiert.

Die Notationen <Wert 1> bis <Wert 3> sind wie folgt zu besetzen:

- <Wert 1> Gruppe der einstellbaren Nullpunktverschiebungen (G 54=1 bis G 57=4).
- <Wert 2> Nummer der Achse.
- <Wert 3> Grob- oder Feinwert (0 oder 1).

@432 <Wert 1> <Wert 2> <Wert>

**Programmierbare
Nullpunktverschiebung**

Es bedeuten:

- <Wert 1> Gruppe der programmierbaren Nullpunktverschiebungen (G58=1 und G 59=2).
- <Wert 2> Nummer der Achse.

@434 <Wert 2> <Wert>

DRF-Verschiebung

Unter <Wert 2> wird die Nummer der Achse der DRF-Verschiebung definiert.

@435 <Wert 2> <Wert>

PRESET-Verschiebung

Unter <Wert 2> wird die Nummer der Achse der PRESET-Verschiebung definiert.

Wird die PRESET-Verschiebung in der Betriebsart AUTOMATIC oder MDA mit dem @435 beschrieben, wirkt die Verschiebung erst nach M2/M30/PRESET.

Bei aktiver Funktion "TRANSMIT" wirkt der Befehl @435 auch auf fiktive Achsen.

Der @435 dient hauptsächlich zum gezielten **Rücksetzen** der PRESET-Verschiebung bei Programmende.

@437 <Wert 1> <Wert 2> <Wert 3> <Wert>

**Einstellbare
Koordinatendrehung**

Einstellbare Koordinatendrehung schreiben.

Die Notationen <Wert 1> bis <Wert 3> sind wie folgt zu besetzen:

- <Wert 1> Nummer des Kanals (0 = eigener Kanal).
- <Wert 2> Gruppe der einstellbaren Koordinatendrehungen (G54=1 bis G57=4).
- <Wert 3> Nummer des Winkels (z.Zt. = 1).

@438 <Wert 1> <Wert 2> <Wert 3> <Wert>

**Einstellbare
Koordinatendrehung additiv**

Einstellbare Koordinatendrehung (G54 bis G57) additiv schreiben.

Es bedeuten:

- <Wert 1> Nummer des Kanals (0 = eigener Kanal).
- <Wert 2> Gruppe 1 bis 4 (G54 bis G57).
- <Wert 3> Nummer des Winkels (z.Zt. = 1).

@439 <Wert 1> <Wert 2> <Wert 3> <Wert>**Programmierbare
Koordinatendrehung**

Programmierbare Koordinatendrehung schreiben.

Es bedeuten:

- <Wert 1> Nummer des Kanals (0 = eigener Kanal).
- <Wert 2> Gruppe der programmierbaren Koordinatendrehungen
(G58 = 1 und G59 = 2).
- <Wert 3> Nummer des Winkels (z.Zt. = 1).

@43a <Wert 1> <Wert 2> <Wert 3> <Wert>**Programmierbare
Koordinatendrehung additiv**

Programmierbare Koordinatendrehung (G58 und G59) additiv schreiben.

Es bedeuten:

- <Wert 1> Nummer des Kanals (0 = eigener Kanal).
- <Wert 2> Gruppe 1 bis 2 (G58 und G59).
- <Wert 3> Nummer des Winkels (z.Zt. = 1).

Hauptgruppe 4/Untergruppe 4: R-Parameter in programmierte Sollwerte schreiben**@440 <Wert 3> <Wert>****Programmierte Achs-Position**

Dieser Befehl ermöglicht es, Achsen unabhängig von Achsnamen zu programmieren.

Unter <Wert 3> wird die Nummer der zu verfahrenen Achse angegeben und unter <Wert> die anzufahrende Position bzw. der zu verfahrenende Weg vorgegeben.

Bei aktiver Funktion "Achsverdopplung" wird der Wert der programmierten Achsposition sowohl der führenden als auch der verdoppelten Achse zugewiesen.

Beispiel:

@440 K2 K100 L_F

Dem Verfahrenweg der 2. Achse wird über eine Konstante der Wert 100 zugewiesen.

@442 <Wert 3> <Wert>**Programmierte Spindeldrehzahl**

Dieser Befehl ermöglicht es, die Spindeldrehzahl zu programmieren.

Unter <Wert 3> wird die Nummer der Spindel und unter <Wert> die Spindeldrehzahl angegeben.

@446 <Wert>**Programmierter Radius**

Dieser Befehl ermöglicht es, den Radius unabhängig von der im Maschinendatum festgelegten Adresse zu programmieren. Unter <Wert> wird der Zahlenwert vorgegeben.

@447 <Wert>**Programmierter Winkel**

Dieser Befehl ermöglicht es, den Winkel unabhängig von der im Maschinendatum festgelegten Adresse zu programmieren. Unter <Wert> wird der Zahlenwert vorgegeben.

@448 <Wert 3> <Wert>

Programmierter Interpolationsparameter

Dieser Befehl ermöglicht es, die Interpolationsparameter für Kreis und Gewinde zu programmieren.

Unter <Wert 3> wird die Nummer der Achse und unter <Wert> der Interpolationsparameter angegeben.

Hauptgruppe 4/Untergruppe 8: R-Parameter in PLC-Signalbits schreiben

@482 <Wert 1> <Wert 2> <Wert 3> <Wert>

Der Zustand eines Merkerbits in der PLC wird über einen Parameter, einen Pointer oder eine Konstante geladen.

Die PLC-Nummer wird durch den <Wert 1>, die Byteadresse durch <Wert 2> und die Bitadresse durch <Wert 3> angegeben. Die PLC-Nummer muß mit K1 versehen werden.

Beispiel:

@482 K1 K2 K0 K0

Der Zustand des definierten PLC-Merkerbits wird mit 0 geladen.

@483 <Wert 1> <Wert 2> <Wert 3> <Wert 4> <Wert>

Der Zustand eines Datenwortbits in der PLC wird über einen Parameter, einen Pointer oder eine Konstante geladen.

Die PLC-Nummer wird durch den <Wert 1>, die Nummer des DB durch <Wert 2>, die Datenwortnummer durch <Wert 3> und die Bitadresse durch <Wert 4> definiert. Die PLC-Nummer muß mit K1 versehen werden.

Beispiel:

R60=1 R62=2 R63=4

@483 R60 R62 R63 K2 K1

Der Zustand des definierten PLC-Datenwortbits wird mit 1 geladen.

Hinweis:

Im 3. Kanal (Simulation) werden @482 und @483 überlesen. Dies kann zu einem veränderten Programmablauf führen.

Hauptgruppe 4/Untergruppe e: R-Parameter in Systemzellen schreiben

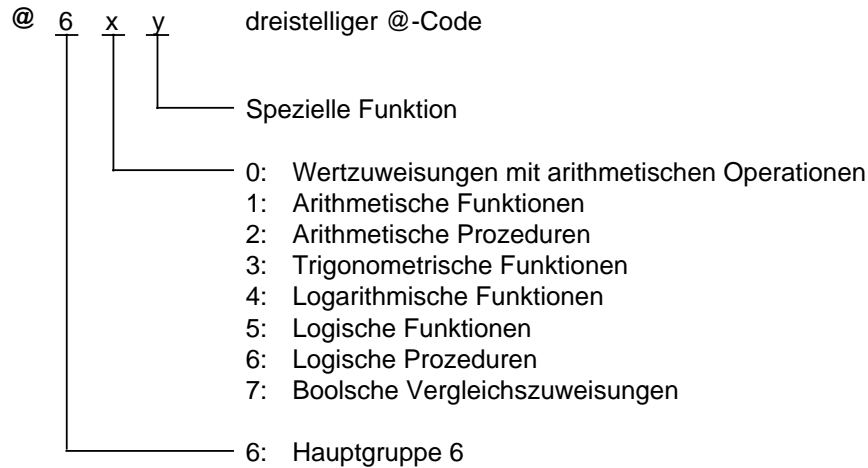
@4e1 <Wert 1> <Wert 2> <Wert>

Spindelbeschleunigungszeitkonstante programmieren

Dieser Befehl ermöglicht es, die Spindelbeschleunigungszeitkonstante zu programmieren. Die unter <Wert> angegebene Spindelbeschleunigungszeitkonstante überschreibt nicht den in den Maschinendaten eingegebenen Wert, sondern eine interne Datenzelle. Die Maschinendaten werden wieder aktiv nach dem Abwählen von Gewindebearbeitung (z.B. G00...), nach Maschinendatenänderung, nach RESET und nach POWER ON. Die Spindelnummer wird durch <Wert 1> und die Getriebestufe durch <Wert 2> definiert.

11.8 Mathematische Funktionen

Die Hauptgruppe 6 ist folgendermaßen gegliedert:



Hauptgruppe 6/Untergruppe 0: Wertzuweisungen mit arithmetischen Operationen

In dieser Untergruppe wird kein @ benötigt. Eine Kettenrechnung mit mehreren Notationen auf der rechten Seite der Gleichung ist erlaubt.

<Var>	=	<Wert 1>	+	<Wert 2>	Addition
<Var>	=	<Wert 1>	-	<Wert 2>	Subtraktion
<Var>	=	<Wert 1>	.	<Wert 2>	Multiplikation
<Var>	=	<Wert 1>	/	<Wert 2>	Division.

Hauptgruppe 6/Untergruppe 1: Arithmetische Funktionen

@610 <Var> <Wert>

Betragsbildung

Von dem unter <Wert> definierten Zahlenwert wird der Betragsanteil nach <Var> abgespeichert.

Beispiel:

```
R12=-34 LF
```

```
@610 R76 R12 LF
```

```
:
```

Im R-Parameter R76 steht der Betrag (=34) aus R12.

@613 <Var> <Wert>

Quadratwurzel

Von dem unter <Wert> definierten Zahlenwert wird die Quadratwurzel gebildet und nach <Var> abgespeichert.

Beispiel:

```
@613 R13 K64 LF
```

Die Quadratwurzel wird aus der Konstanten (=64) gezogen und das Ergebnis (=8) in R13 eingetragen/gespeichert.

@614 <Var> <Wert 1> <Wert 2>

Wurzel aus Quadratsumme

Von dem unter <Wert 1> und <Wert 2> definierten Zahlenwert wird die Quadratsumme gebildet, dann die Quadratwurzel gezogen und nach <Var> abgespeichert.

Beispiel:

R25=15 R26=20 L_F

@614 R77 R25 R26 L_F

Die Quadratwurzel wird aus der Summe der Quadrate der R-Parameter R25 (=225) und R26 (=400) gebildet. Das Ergebnis (=25) wird in R77 eingetragen/ gespeichert.

Hauptgruppe 6/Untergruppe 2: Arithmetische Prozeduren

@620 <Var>

Inkrementieren

Der Inhalt des unter <Var> definierten R-Parameters wird inkrementiert.

Beispiel:

R70=1 L_F

@620 R70 L_F

Der Inhalt des Parameters R70 wird inkrementiert; der neue Inhalt ist 2.

@621 <Var>

Dekrementieren

Der Inhalt des unter <Var> definierten R-Parameters wird dekrementiert.

Beispiel:

R70=1 L_F

@621 R70 L_F

Der Inhalt des Parameters R70 wird dekrementiert; der neue Inhalt ist 0.

@622 <Var>

Ganzzahliger Anteil

Von dem durch einen R-Parameter oder einen Pointer definierten Zahlenwert wird der ganzzahlige Anteil gebildet. Das Ergebnis steht danach im gleichen R-Parameter oder Pointer.

Beispiel:

R60=2.9 L_F

@622 R60 L_F

Es wird der ganzzahlige Inhalt von R60 gebildet; der neue Inhalt ist 2.

Hauptgruppe 6/Untergruppe 3: Trigonometrische Funktionen

@630 <Var> <Wert>

Sinus

Von dem unter <Wert> definierten Winkelwert wird der Sinus gebildet und nach <Var> abgespeichert.

Beispiel:

R27=30 L_F

@630 R15 R27 L_F

Der Sinus wird aus dem Inhalt von R27 (=0.5) in R15 eingetragen/gespeichert.

@631 <Var> <Wert>

Cosinus

Von dem unter <Wert> definierten Winkelwert wird der Cosinus gebildet und nach <Var> abgespeichert.

@632 <Var> <Wert>

Tangens

Von dem unter <Wert> definierten Winkelwert wird der Tangens gebildet und nach <Var> abgespeichert.

@634 <Var> <Wert>

ArcusSinus

Von dem unter <Wert> definierten Wert wird der ArcusSinus gebildet und als Winkelwert nach <Var> abgespeichert.

Beispiel:

R35=0.70710678 L_F

@634 R17 R35 L_F

Der ArcusSinus wird aus dem Inhalt von R35 gebildet und das Ergebnis (=45) in R17 eingetragen/gespeichert.

@637 <Var> <Wert 1> <Wert 2>

Winkel aus 2 Vektorkomponenten

Die unter <Wert 1> und <Wert 2> definierten Zahlenwerte werden als Vektoren betrachtet. Das Ergebnis ist der Winkel zwischen der unter <Wert 2> stehenden Komponente und dem Summenvektor.



Als Operanden für <Wert 1> und <Wert 2> ist nur einmal eine Konstante <Const> zugelassen. Der andere Operand muß eine Variable <Var> (R-Parameter oder Pointer) sein.

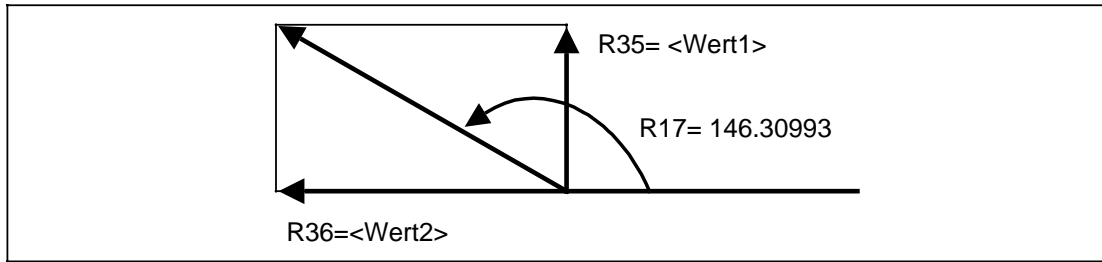


Beispiel:

R35=20 R36=30 L_F

@637 R17 R35 R36 L_F

Der Winkel wird aus den Vektorkomponenten (Inhalte der R-Parameter R35 und R36) gebildet und das Ergebnis (=146.30993) in R17 eingetragen/gespeichert.



Beispiel zu @637

Hauptgruppe 6/Untergruppe 4: Logarithmische Funktionen

@640 <Var> <Wert>

Natürlicher Logarithmus

Von dem unter <Wert> definierten Zahlenwert wird der natürliche Logarithmus gebildet und nach <Var> abgespeichert.

Beispiel:

@640 R80 K10 L_F

Es wird der natürliche Logarithmus von der Konstanten 10 gebildet. Das Ergebnis (=2.3025846) wird in R80 eingetragen/gespeichert.

@641 <Var> <Wert>

Exponentialfunktion

Von dem unter <Wert> definierten Zahlenwert wird die Exponentialfunktion e^x gebildet und nach <Var> abgespeichert.

Beispiel:

@641 R80 K2.5 L_F

Die Exponentialfunktion für den durch die Konstante angegebenen Exponenten wird berechnet. Das Ergebnis (=12.182496) wird in R80 eingetragen/gespeichert.

Hauptgruppe 6/Untergruppe 5: Logische Funktionen

@650 <Var> <Var 1> <Wert>

ODER

Die unter <Var 1> und <Wert> stehenden Bitmuster (Pattern) werden logisch nach ODER verknüpft. Das Ergebnis wird nach <Var> abgespeichert.

Beispiel:

R50=00101100 L_F

R51=10110011 L_F

@650 R52 R50 R51 L_F

Die Pattern-Variablen R50 und R51 werden logisch nach ODER verknüpft und das Ergebnis in R52 hinterlegt. R52 hat den Inhalt 10111111.

@651 <Var> <Var 1> <Wert>**EXKLUSIVE-ODER**

Die unter <Var 1> und <Wert> stehenden Bitmuster (Pattern) werden logisch nach EXODER verknüpft. Das Ergebnis wird nach <Var> abgespeichert.

@652 <Var> <Var 1> <Wert>**AND**

Die unter <Var 1> und <Wert> stehenden Bitmuster (Pattern) werden logisch nach UND verknüpft. Das Ergebnis wird nach <Var> abgespeichert.

@653 <Var> <Var 1> <Wert>**NAND**

Die unter <Var 1> und <Wert> stehenden Bitmuster (Pattern) werden logisch nach UND verknüpft. Das Ergebnis wird negiert und nach <Var> abgespeichert.

@654 <Var> <Wert>**NOT**

Das unter <Wert> stehende Bitmuster (Pattern) wird logisch negiert. Das Ergebnis wird nach <Var> abgespeichert.

Beispiel:

```
R50=00101100 LF
```

```
@654 R52 R50 LF
```

Der Inhalt der Pattern-Variablen R50 wird negiert und das Ergebnis in R52 hinterlegt. R52 hat den Inhalt 11010011.

@655 <Var> <Var 1> <Wert>**ODER-Bit**

Die unter <Var 1> und <Wert> stehenden Bits werden logisch nach ODER verknüpft. Das Ergebnis wird nach <Var> abgespeichert.

@656 <Var> <Var 1> <Wert>**EXKLUSIVE-ODER-Bit**

Die unter <Var 1> und <Wert> stehenden Bits werden logisch nach EXODER verknüpft. Das Ergebnis wird nach <Var> abgespeichert.

@657 <Var> <Var 1> <Wert>**AND-Bit**

Die unter <Var 1> und <Wert> stehenden Bits werden logisch nach UND verknüpft. Das Ergebnis wird nach <Var> abgespeichert.

Beispiel:

```
R50=1 LF
```

```
R51=0 LF
```

```
@657 R52 R50 R51 LF
```

Die Booleschen Variablen R50 und R51 werden logisch nach UND verknüpft und das Ergebnis in R52 hinterlegt. R52 hat den Inhalt 0.

@658 <Var> <Var 1> <Wert>**NAND-Bit**

Die unter <Var 1> und <Wert> stehenden Bits werden logisch nach UND verknüpft. Das Ergebnis wird negiert und nach <Var> abgespeichert.

Beispiel:

```
R50=1 LF
```

```
R51=0 LF
```

```
@658 R52 R50 R51 LF
```

Die Booleschen Variablen R50 und R51 werden logisch nach UND verknüpft, das Ergebnis negiert und in R52 hinterlegt. R52 hat den Inhalt 1.

@659 <Var> <Wert>**NOT-Bit**

Das unter <Wert> stehende Bit wird logisch negiert und nach <Var> abgespeichert.

Hauptgruppe 6/Untergruppe 6: Logische Prozeduren

@660 <Var> <Const> Lösche Bit in PATTERN
Durch die Konstante <Const> wird ein Bit (0 bis 7) definiert, das in dem durch <Var> festgelegten Bitmuster (Pattern) gelöscht werden soll.

Beispiel:

R60=01100111 L_F
@660 R60 K6 L_F

Das Bit Nr. 6 der Pattern-Variablen wird gelöscht. R60 hat den Inhalt 00100111.

@661 <Var> <Const> Setze Bit
Durch die Konstante <Const> wird ein Bit (0 bis 7) definiert, das in dem durch <Var> festgelegten Bitmuster (Pattern) auf "1" gesetzt werden soll.

Beispiel:

R70=00000000 L_F
@661 R70 K2 L_F

Das Bit Nr. 2 der Pattern-Variablen wird gesetzt. R70 hat den Inhalt 00000100.

Hauptgruppe 6/Untergruppe 7: Boolesche Vergleichszuweisungen

@671 <Var 1> <Var 2> <Wert> Gleich
Sind die unter <Var 2> und <Wert> definierten Zahlenwerte gleich, so wird die Boolesche Variable <Var 1> auf "1" gesetzt.

Beispiel:

R50=11001100 L_F
@671 R51 R50 K11001100 L_F

Da R50 gleich dem Bitmuster der Konstanten K ist, wird R51 auf "1" gesetzt.

@672 <Var 1> <Var 2> <Wert> Ungleich
Sind die unter <Var 2> und <Wert> definierten Zahlenwerte ungleich, so wird die Boolesche Variable <Var 1> auf "1" gesetzt.

@673 <Var 1> <Var 2> <Wert> ">"
Ist der unter <Var 2> definierte Zahlenwert größer als der unter <Wert> stehende, so wird die Boolesche Variable <Var 1> auf "1" gesetzt.

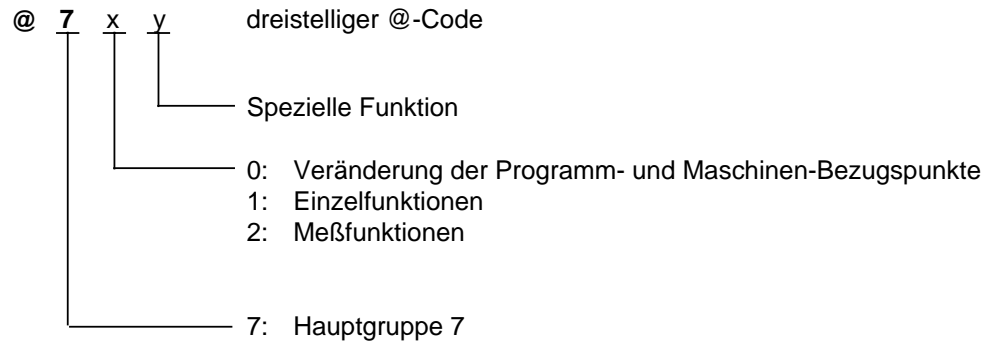
@674 <Var 1> <Var 2> <Wert> ">="
Ist der unter <Var 2> definierte Zahlenwert größer als der oder gleich dem unter <Wert> stehenden, so wird die Boolesche Variable <Var 1> auf "1" gesetzt.

@675 <Var 1> <Var 2> <Wert> "<"
Ist der unter <Var 2> definierte Zahlenwert kleiner als der unter <Wert> stehende, so wird die Boolesche Variable <Var 1> auf "1" gesetzt.

@676 <Var 1> <Var 2> <Wert> "<="
Ist der unter <Var 2> definierte Zahlenwert kleiner als der oder gleich dem unter <Wert> stehenden, so wird die Boolesche Variable <Var 1> auf "1" gesetzt.

11.9 NC-spezifische Funktionen

Die Hauptgruppe 7 ist folgendermaßen gegliedert:



Hauptgruppe 7/Untergruppe 0: Veränderung der Programm- und Maschinen-Bezugspunkte

@706

Position im Satz bezogen auf das Maschinen Istwert-System

Mit dem Befehl @706 wird - bezogen auf den Maschinennullpunkt - eine Position vorgegeben und von den angegebenen Achsen angefahren. Der Befehl wirkt nur satzweise.

Es können so viele Achsen vorgegeben werden, wie die NC gleichzeitig verfahren kann. Die anzufahrenden Positionen der Achsen werden entweder über DIN-Code oder über den Befehl @440 ... programmiert.

Der Befehl @706 unterdrückt alle Nullpunktverschiebungen (einstellbar, einstellbar additiv, programmierbar und extern) sowie die PRESET- und die DRF-Verschiebungen.

Zum Anfahren einer Position im Maschinen-Istwert-System müssen außerdem die Werkzeug-Korrekturen abgewählt werden.

Ist das Maschinendatum 5007.1 gesetzt, so wirkt G53 wie @706.

Ist das Maschinendatum nicht gesetzt, so werden mit G53 die PRESET- und die DRF-Verschiebungen nicht unterdrückt.

Beispiel:

@706 X1000 Z500 L_F

Die programmierten Verfahrenswege in X und Z werden bezogen auf den Maschinennullpunkt angefahren.

Hauptgruppe 7/Untergruppe 1 : Einzelfunktionen

@710 <Var 1> <Var 2>

Referenzauflbereitung

Dieser Befehl wird als Referenzauflbereitung im Drehzyklus L95 "Abspannen" ben6tigt. Mit diesem Befehl wird eine in einem Unterprogramm programmierte Kontur in einzelne Sätze zerlegt, wobei die Daten in R-Parametern hinterlegt werden.

Das Konturelement wird ab dem R-Parameter <Var 1> (R_n) in insgesamt acht R-Parametern abgelegt. Die Referenzauflbereitung ben6tigt insgesamt vier R-Parameter als Eingangsdaten.

Der erste dieser R-Parameter wird durch <Var 2> (R_m) festgelegt. Der Eingangssteuerparameter (R_m+3) mu0df vor dem 1. Aufruf von @710 auf "1" gesetzt werden. Das 1. Konturelement des Unterprogramms wird in R-Parametern abgelegt, beginnend vom Anfangspunkt der Kontur (R_m+1, R_m+2).

Im Unterprogramm wird dieser Punkt nicht programmiert. Der Befehl @710 setzt den Steuerparameter auf "0", so da0f bei jedem weiteren Aufruf die Werte vom nlichsten Konturelement geladen werden. Erkennt der Befehl "Unterprogrammende" (M17), dann wird der Ausgangssteuerparameter (R_n+7) automatisch auf "1" bzw. "2" gesetzt.

Voraussetzung

1. Kontur im Unterprogramm
2. Anfangspunkte
3. Steuerparameter (R_m + 3) = 1



Parameter

- | | | | |
|-------------------|---------------------|---|-----------|
| R _m | Unterprogrammnummer | ← | < Var 2 > |
| R _m +1 | Anfangspunkt Y | | |
| R _m +2 | Anfangspunkt X | | |
| R _m +3 | Steuerparameter | | |

@710 l6dt in:



- | | | | |
|-------------------|---------------------------|---|-----------|
| R _n | Anfangspunkt Y | ← | < Var 1 > |
| R _n +1 | Anfangspunkt X | | |
| R _n +2 | Endpunkt Y | | |
| R _n +3 | Endpunkt X | | |
| R _n +4 | Interpolationsparameter J | | |
| R _n +5 | Interpolationsparameter I | | |
| R _n +6 | G-Funktion | | |
| R _n +7 | Steuerungsparameter | | |



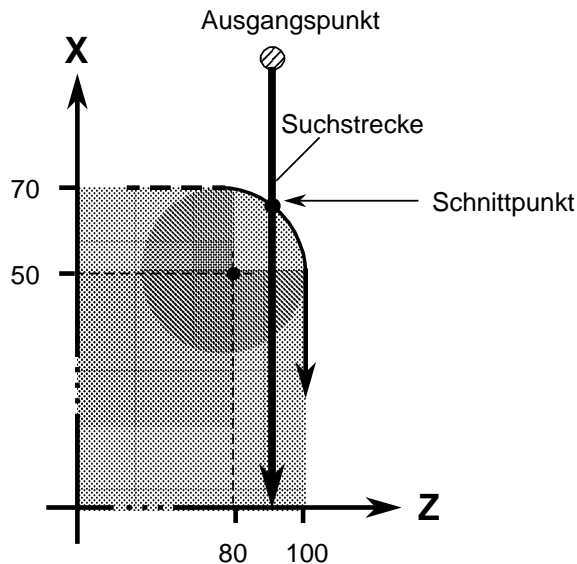
- | | |
|------------------------|--------------------|
| R _n +7 = 0: | Satz ohne M17 |
| R _n +7 = 1: | Satz mit M17 |
| R _n +7 = 2: | M17 allein im Satz |

@711 <Var 1> <Var 2> <Var 3>**Schnittpunktberechnung**

Dieser Befehl wird als Schnittpunktberechnung im Drehzyklus L95 "Abspannen" benötigt. Mit @711 wird der Schnittpunkt einer Suchstrecke mit einem Konturelement ermittelt. Die Schnittpunktberechnung benötigt insgesamt acht Parameter als Eingangsdaten für das erste Konturelement.

Die erste R-Parameter-Nummer wird durch < Var 2 > (Rn) festgelegt. Die zweite Kontur ab < Var 3 > (Rr) ist zur Zeit nicht realisiert, d.h. diese Notation wird z. Z. nicht benötigt; es muß aber irgendein R-Parameter beim Programmieren angegeben werden.

Die Ausgangsdaten der Schnittpunktberechnung werden ab dem R-Parameter < Var 1 > (Rm) in insgesamt drei R-Parametern abgelegt. Die Suchrichtung wird hinter dem Befehl als normale Verfahrbewegung programmiert. Es werden beide Achswerte zur Ermittlung des Schnittpunktes benötigt. Die Ausgangsdaten der Referenzaufbereitung werden dabei als Eingangsdaten für die Schnittpunktberechnung verwendet.



R-Parameter	Bedeutung	Registerinhalt
R _n	Satzanfang X	70
R _n +1	Satzanfang Z	80
R _n +2	Satzende X	50
R _n +3	Satzende Z	100
R _n +4	Interpolationsparameter I	- 20
R _n +5	Interpolationsparameter K	0
R _n +6	G-Funktion	2

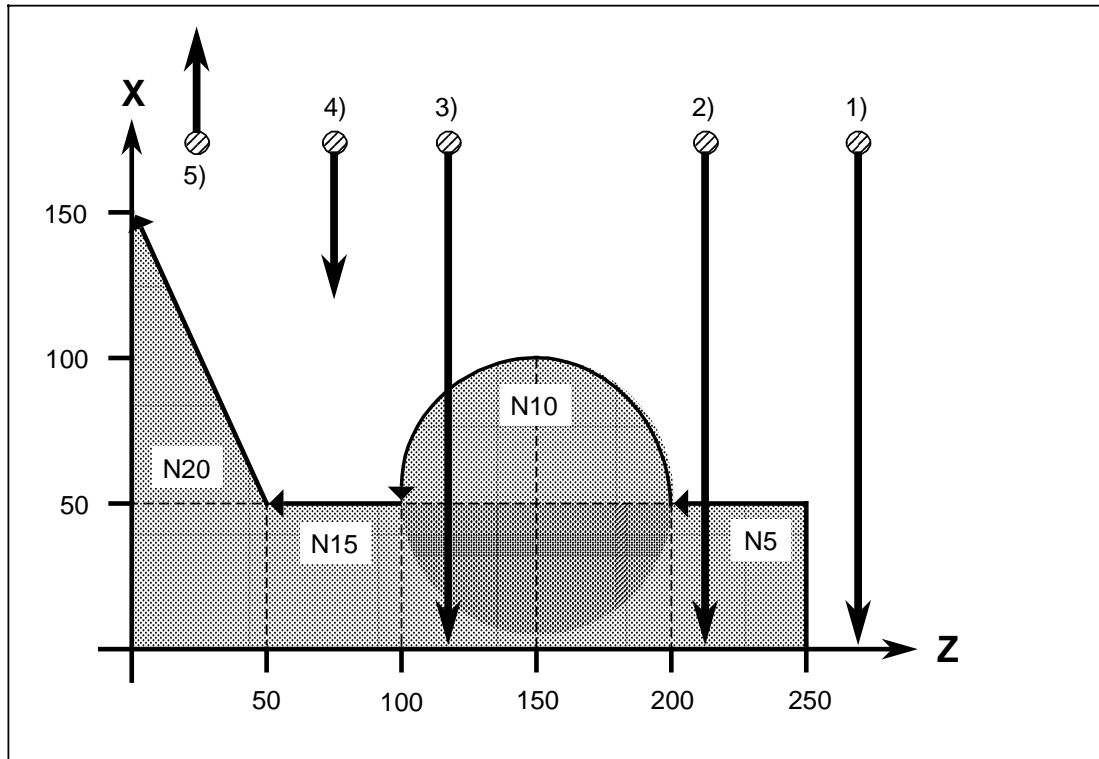
@711 R_m R_n R_r G01 G90 X0 Z90 L_F

**Suchrichtung,
lädt in:**

R _m	Ergebnis	1*)
R _m +1	Schnittpunkt X	67,320
R _m +2	Schnittpunkt Z	90,000

*) 1=gefunden, 0=nicht gefunden

Bewertung von Schnittpunktberechnungen mit @711:



- 1) Schnittpunkt nicht gefunden
- 2) Schnittpunkt gefunden
- 3) Kreisbogen geht über die Quadrantengrenze (!):
Schnittpunkt nicht gefunden
- 4) Suchstrecke war zu kurz:
Schnittpunkt trotzdem gefunden
- 5) Falsche Suchrichtung:
Schnittpunkt trotzdem gefunden

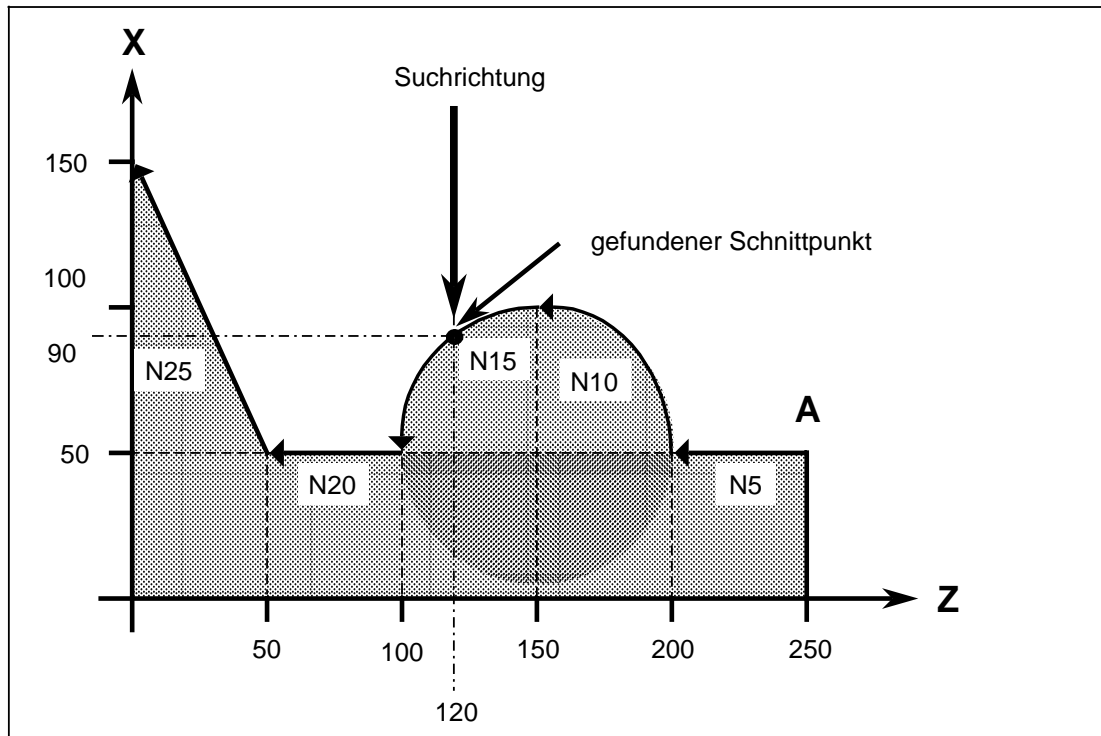
Mit der Suchrichtung wird eine Strecke festgelegt, längs der der Schnittpunkt erwartet wird. Ein Schnittpunkt wird immer dann gefunden, wenn diese Strecke oder ihre Verlängerung die in R_n bis (R_n+6) hinterlegte Kontur trifft. Es wird also auch dann ein Schnittpunkt gefunden, wenn die hinter @711 programmierte Wegstrecke in die falsche Richtung weist oder zu kurz ist.

Die Funktion @711 trägt nun in das Register R_m den Wert 0 ein, wenn kein Schnittpunkt gefunden wurde, oder den Wert 1, wenn die Suche erfolgreich war. Wurde ein Schnittpunkt gefunden, so enthalten dann die Parameter $(R_m + 1)$ und $(R_m + 2)$ die Schnittpunktkoordinaten, im anderen Fall die Werte 0.

Wenn nun mit den Werten aus (R_m+1) und (R_m+2) eine Verfahrbewegung programmiert werden soll, muß vorher in einer Schleife abgefragt werden, ob ein Schnittpunkt gefunden wurde oder nicht.

Beispiel: Schnittpunktberechnung mit @711

Gesucht wird der Schnittpunkt mit einer programmierten Kontur bei Z=120.



Unterprogramm der Kontur:

```
L20 LF
N5 X50 Z200 LF
N10 G03 X100 Z150 I0 K-50 LF
N15 G03 X50 Z100 I-50 K0 LF
N20 X50 Z50 LF
N25 X150 Z0 LF
N30 M17 LF
```

Hauptprogramm:

```
%30 LF
G0 X150 Z120 LF
R50=20 R51=0 R52=250 R53=1 R70=0 LF

N300 @131 R70 K0 K305 LF
@710 R54 R50 LF
@711 R70 R54 R62 G01 G90 X0 Z120 LF
@100 K-300 LF
N305 G00 X=R71 Z=R72 LF
:
:
M30 LF
```

R50 bis R54, Eingangsdaten; R70: Flag für
Schnittpunktberechnung

While-Schleife, solange R 70=0

Referenzauflösung

Schnittpunktberechnung

Schleifenende

Anfahren des Schnittpunktes im Eilgang

Die Programmschleife wird solange durchlaufen, bis ein Schnittpunkt gefunden wird.

	R _n	R _{n+1}	R _{n+2}	R _{n+3}	R _{n+4}	R _{n+5}	R _{n+6}	R _{n+7}
1. Aufruf @710	50	250	50	200	0	0	1	0
@711	SCHNITTPUNKT GEFUNDEN? Nein: R _m = 0							
2. Aufruf @710	50	200	100	150	0	-50	3	0
@711	SCHNITTPUNKT GEFUNDEN? Nein: R _m = 0							
3. Aufruf @710	100	150	50	100	-50	0	3	0
@711	SCHNITTPUNKT GEFUNDEN? Ja: R _m = 1							

Nach dem 3. Aufruf wird in diesem Beispiel der Schnittpunkt bei X=90 (R71) und Z=120 (R72) gefunden.

@713 <Var > Startvorbereitung für Zyklen

Mit diesem Befehl wird in den mit <Var> definierten R-Parameter der Zahlenwert geladen, der dem Sicherheitsabstand von 1 mm im aktuellen Eingabeformat entspricht (bei G70 = 0.03937 und bei G71 = 1).

In den unmittelbar folgenden R-Parameter wird bei Radiusprogrammierung der Zahlenwert "1" und bei Durchmesserprogrammierung der Zahlenwert "2" geladen. Der zulässige Parameterbereich für <Var> ist R0 bis R98 und R900 bis R998.

@714 Stop Dekodierung bis Zwischenspeicher leer

Mit diesem Befehl "STOP-DEC" wird die Satzaufbereitung (Dekodierung) solange unterbunden, bis der Zwischenspeicher leer ist.

Bei der Programmbearbeitung werden in der Steuerung mehrere Programmsätze im voraus dekodiert und in den Zwischenspeicher der NC geladen. Dies bewirkt eine schnellere Programmbearbeitung, kann jedoch in Zusammenhang mit bestimmten

NC-Befehlen (Istwertlesen, Messen, Datenübergabe NC-PLC) zu einem fehlerhaften Programmablauf führen.

Durch den STOP-DEC-Befehl (Stop der Dekodierung) wird die Vorausdekodierung der NC-Sätze, die nach diesem Befehl stehen, solange gestoppt, bis der Satz (@714) mit dem STOP-DEC- Befehl bearbeitet wurde. Dadurch wird erreicht, daß der Zwischenspeicher leer ist und Informationen, die in den nächsten NC-Sätzen benötigt werden, zur Verfügung stehen.

Für die folgenden Informationen aus der Anpaßsteuerung muß die STOP-DEC-Anweisung programmiert werden, sofern sie in den nächsten NC-Sätzen benötigt werden:

- Maschinendaten
- Settingdaten
- Werkzeug-Korrekturen
- Nullpunktverschiebungen
- R-Parameter
- Signal"Spiegeln".

Der Befehl STOP-DEC ist vor jedem Istwertlesen im eigenen Kanal und nach jedem Messen zu programmieren.

Ebenso ist beim Schreiben von Nullpunktverschiebungen und Werkzeugkorrekturen dieser Befehl vorher zu programmieren, wenn die neuen Werte erst ab diesem Satz wirken sollen.



Der STOP-DEC-Befehl @714 muß grundsätzlich in einem eigenen Satz stehen.

Beispiel:

M94 LF

@714 LF

@123 R60 K100 K5 LF

Die Teilenummer wird von der PLC in R60 übertragen.

Stop der Dekodierung, damit die aktuelle Teilenummer im nächsten NC-Satz ausgewertet werden kann.

Verzweigung entsprechend der Teilenummer.

@715**Stop Dekodierung bis Zwischenspeicher leer bei Koordinatendrehung**

Wird mit Hilfe der CL800-Sprache der Drehwinkel der einstellbaren oder programmierbaren Koordinatendrehung geladen, so wird dieser Winkelwert sofort zur Berechnung herangezogen. Dies kann zur Folge haben, daß dieser Winkel bereits in frühere Verfahrssätze eingerechnet wird.

Soll der Winkel erst beim nächsten Satz gültig sein, so müssen alle vorherigen Sätze abgearbeitet werden (Zwischenspeicher leeren).

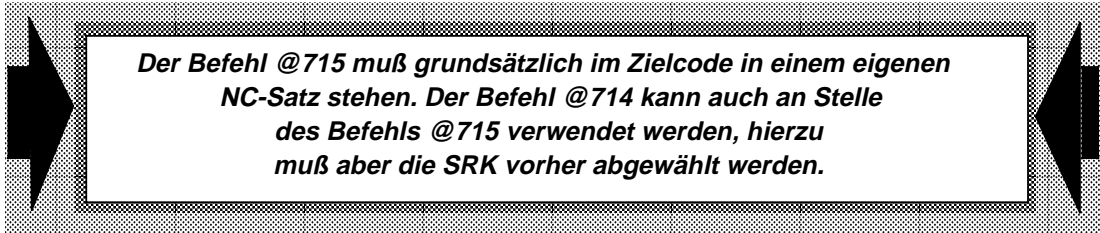
Dies kann durch programmieren von STOP DEC 1 auch bei angewählter SRK erreicht werden. Die Dekodierung wird dabei erst wieder aktiviert, wenn der Zwischenspeicher bis zur Koordinatendrehung geleert ist.

Beispiel:

@715 LF

@437 K0 K1 K1 K30 LF

Koordinatendrehung G54 mit Drehwinkel 30° im Kanal 0 (eigener Kanal)



Der Befehl @715 muß grundsätzlich im Zielcode in einem eigenen NC-Satz stehen. Der Befehl @714 kann auch an Stelle des Befehls @715 verwendet werden, hierzu muß aber die SRK vorher abgewählt werden.

Hauptgruppe 7/Untergruppe 2: Meßfunktionen

@720 <Var> <Wert>

Fliegendes Messen bezogen auf den Maschinennullpunkt

Dieser Wert wird in Meßzyklen verwendet.

Mit der Meßfunktion werden die Istwerte der fahrenden Achsen zum Zeitpunkt eines Eingangssignals vom Meßtaster ermittelt. Die Istwerte werden direkt vom Meßkreis der Steuerung bei Erkennen der Schaltflanke des Meßtasters erfaßt.

Die Istwerte werden ab dem Parameter <Var> mit aufsteigender Achsnummer hinterlegt und sind auf den Maschinennullpunkt bezogen. Anschließend generiert die Steuerung ein "Restweg löschen", d.h. die Restwege (Soll-Istwert-Differenzen) aller Achsen werden gelöscht.

Von der Steuerung wird der Sollwert 0 als Sprungfunktion vorgegeben. Die Bremswege der Achsen werden noch verfahren, d.h. die Schleppabstände werden abgebaut. Dadurch müssen die nächsten Verfahrsätze im Absolutmaß (G90) programmiert werden.

Die erfaßten Istwerte der zum Zeitpunkt des Messens verfahrenen Achsen werden ab dem unter <Var> definierten Parameter mit aufsteigenden Achsnummern hinterlegt.

Die Nummer des Meßeingangs (1 oder 2) wird mit <Wert> vorgegeben.

Die Verfahrwege der Achsen (Sollpositionen) werden über DIN-Code oder den Befehl @440 im selben NC-Satz programmiert. Die Sollpositionen der Achsen sind auf den Werkstücknullpunkt bezogen. Bei aktiver Funktion "Achverdopplung" wirkt der Befehl

@440 sowohl auf die führenden, programmierten Achsen als auch auf die verdoppelten Achsen.

Beispiel:

·
·

@720 R93 K1
@440 K1 R70 L_F

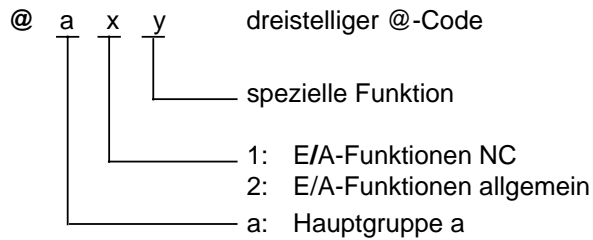
Der Istwert der 1. Achse wird gemessen und in R93 geladen.
Dem Verfahrweg der 1. Achse wird über eine Konstante der Wert aus R70 zugewiesen.

·
·

Ist die Funktion "Achsverdopplung" aktiv, wird der programmierte Verfahrweg auf die entsprechenden Achse verdoppelt, aber nur der Istwert der führenden Achse gemessen. Beide Achsen stoppen nach dem Auslösen des Meßtasters.

11.10 E/A-Funktionen

Die Hauptgruppe a (E/A-Funktionen) ist folgendermaßen gegliedert:



Hauptgruppe a/Untergruppe 1: E/A-Funktionen NC

@a15 <Wert 4> <Wert 5> (Menüanwahl vom NC-Programm)

Mit diesem Befehl können Menüs (Bilder und Menütexe) aus dem Anwenderbereich und dem Standardbereich vom NC-Programm aus aufgerufen werden.

<Wert 4> = 0: Anwenderbereich
 1: Standardbereich
<Wert 5> = 1 bis 254: Menünummer

Beispiel:

```

.
N.. @a15 K1 K20 LF
.

```

Es wird das Standardbild für die Daten-Ein-/Ausgabe aufgerufen.

Anwendung:

- Aufruf von projektierten Bildern, die dem Bediener Informationen über den aktuellen Bearbeitungszyklus geben.
- Vorgeben von Bedienungsmasken, die der Bediener nur quittieren muß.

@a1b (Rücksprung in das Ausgangsmemü)

Nachdem mit @a15 ein beliebiges Menübild aufgerufen wurde, kann mit diesem Befehl wieder in das Bild zurückgesprungen werden, das vor dem ersten @a15-Aufruf ausgewählt war.

Beispiel:

Es ist z. B. momentan das PLC-Statusbild aufgerufen.

```

.
N.. @a15 K1 K20           Aufruf des E/A-Bildes
N.. @a1b                 Rückkehr in das PLC-Statusbild
.

```

Anwendung:

Ein Anzeigebild, das vom Bediener ausgewählt war und überschrieben wurde, wird wieder aufgerufen.

Hauptgruppe a/Untergruppe 2: E/A-Funktionen allgemein

@a20 <Wert>

V24-Schnittstelle anwählen

Mit diesem Befehl wählen Sie die Schnittstelle an, über die die Daten-Ein/-Ausgabe abgewickelt werden soll.

<Wert> = Nummer der Schnittstelle (1 oder 2).

Beispiel:

.
N.. @a20 K2 L_F Die zweite Schnittstelle wird angewählt.

Hinweis:

Die über Bedienung angewählte Schnittstelle (1 oder 2) wird durch diesen Befehl nicht beeinflusst.

Bevor mit den Befehlen @a25, @a26, @a27, @a28 und @a29 Daten ein- oder ausgegeben werden, muß im Programm mit @a20 die Schnittstelle angewählt worden sein.

@a25 <Wert 1>

Ausgabe Nullpunktverschiebungen über V24

Mit diesem Befehl können Sie Nullpunktverschiebungen und kanalspezifische Drehwinkel über V24 ausgeben.

<Wert 1> = 0 Nullpunktverschiebungen G54-G57
1 Drehwinkel für Koordinatendrehung Kanal 1
2 " Kanal 2
3 " Kanal 3.

Beispiel:

N.. @a20 K2 L_F Die einstellbaren Nullpunktverschiebungen
N.. @a25 K0 L_F G54-G57 werden über die zweite V24-Schnittstelle ausgegeben.

Bemerkung:

- Die Anwahl der Schnittstelle mit @a20 muß vor dem Aufruf von @a25 programmiert werden.
- Durch ein Maschinendatenbit kann eingestellt werden, ob die Datenausgabe simultan zum weiteren Programmablauf erfolgt, oder ob der Satzwechsel für die Zeit der Übertragung gesperrt wird.

@a26 <Wert 2> <Wert 3> <Wert 4>**Ausgabe Daten über V24**

Mit diesem Befehl können Sie bestimmte Daten über V24 ausgeben. Um welche Daten es sich handelt, wird mit <Wert 2> parametrisiert.

<Wert 2> = 1: Hauptprogramme
2: Unterprogramme
5: NC-Maschinendaten
6: Werkzeugkorrekturen
8: Settingdaten
9: PLC-Maschinendaten.

Mit <Wert 3> wird die Anfangsadresse, mit <Wert 4> die Endadresse des Datensatzes definiert.

Beispiel

```
N.. @a20 K1 LF           Die vorhandenen Teileprogramme mit der Programm-Kennung
N.. @a26 K1 K1 K10 LF    %1 bis %10 werden über die erste V24-Schnittstelle ausgegeben.
```

Bemerkung:

- Vor der Datenausgabe muß die Schnittstelle mit dem Befehl @a20 angewählt werden.
- Durch ein Maschinendatenbit kann eingestellt werden, ob die Datenausgabe simultan zum weiteren Programmablauf erfolgt, oder ob der Satzwechsel für die Zeit der Übertragung gesperrt wird.
- Wenn die auszugebenden Daten im anschließenden Programmteil verändert werden, muß nach dem Befehl @a26 der Befehl @714 (STOP DEC) programmiert werden.

@a27 <Wert 1> <Wert 3> <Wert 4>**Parameterausgabe über V24**

Mit diesem Befehl können Sie gezielt einzelne R-Parameterblöcke über die V24 ausgeben. Die Kanalnummer wird durch <Wert 1> definiert. Bei Vorgabe einer 0 werden die R-Parameter aus dem eigenen Kanal ausgegeben. Bei zentralen Variablen müssen Sie immer 0 vorgeben. Mit <Wert 3> wird die Anfangsadresse, mit <Wert 4> die Endadresse des R-Parameterblockes definiert. Durch das Maschinendatum 5147.1="1" kann der Satzwechsel während der simultanen Ausgabe der R-Parameter im laufenden Programm gesperrt werden.

<Wert 1> = Kanal-Nummer
0: zentrale R-Parameter oder eigener Kanal
1: R-Parameter Kanal 1
2: R-Parameter Kanal 2
3: R-Parameter Kanal 3.

<Wert 3> = Anfangsadresse
000 bis 699 für globale Parameter
700 bis 999 für zentrale Parameter.

<Wert 4> = Endadresse
000 bis 699 für globale Parameter
700 bis 999 für zentrale Parameter.

Beispiel:

N.. @a20 K1 L _F	Erste V24-Schnittstelle anwählen
N.. @a27 K1 K0 K699 L _F	Alle globalen R-Parameter von Kanal 1 werden ausgegeben
N.. @a27 K0 K700 K999 L _F	Alle zentralen R-Parameter werden ausgegeben
N.. R1=20 R11=40 L _F	
N.. @a27 K1 R1 R11 L _F	Die globalen R-Parameter 20 bis 40 vom Kanal 1 werden ausgegeben.

Anwendung:

- Ausgabe von Meßwerten an einen Drucker
- Ausgabe von Parametern an einen gekoppelten Rechner
- Übergabe von Daten an eine andere NC-Steuerung.

Bemerkung:

Die Anwahl der Schnittstelle mit @a20 müssen Sie vor dem Aufruf von @a27 programmieren. Wenn die auszugebenden R-Parameter im anschließenden Programmteil verändert werden, muß nach dem Befehl @a27 der Befehl @714 (STOP DEC) programmiert werden.

Durch ein Maschinendatenbit kann eingestellt werden, ob die Ausgabe simultan zum weiteren Programmablauf erfolgt, oder ob der Satzwechsel für die Zeit der Übertragung gesperrt wird.

@a28 <Wert 2>

Einlesen von Daten über V24

Mit diesem Befehl können Sie das Einlesen von Daten über die V24-Schnittstelle starten. <Wert 2> gibt an, welcher Datentyp eingelesen werden darf.

- <Wert 2> =
- 0: keine Prüfung des Datentyps
 - 1: Hauptprogramme
 - 2: Unterprogramme
 - 3: Clearprogramme
 - 5: NC-Maschinendaten
 - 6: Werkzeugkorrekturen
 - 7: Nullpunktverschiebungen
 - 8: Settingdaten
 - 9: PLC-Maschinendaten
 - 10: R-Parameter.

Beispiel:

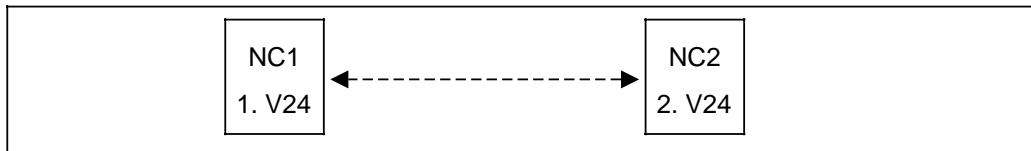
.	
N.. @a20 K1 L _F	Anwahl der ersten V24-Schnittstelle
N.. @a28 K10 L _F	Die Steuerung erwartet einen R-Parameter in der Eingabe. Wird ein anderer Datentyp empfangen, erfolgt eine Fehlermeldung.

Bemerkung:

Die Anwahl der Schnittstelle mit @a20 müssen Sie vor dem Aufruf von @a28 programmieren.

Durch ein Maschinendatenbit kann eingestellt werden, ob das Einlesen der Daten simultan zum weiteren Programmablauf erfolgt, oder ob der Satzwechsel für die Zeit der Dateneingabe gesperrt wird.

Soll im anschließenden Programmteil mit dem eingelesenen R-Parametern gearbeitet werden, muß nach der Übertragung der Befehl @714 (STOP DEC) programmiert werden. Mit den Befehlen @a27 und @a28 können Sie auf einfache Weise ein Datenaustausch zwischen zwei SINUMERIK- Steuerungen realisieren.



Voraussetzungen für die NC-NC-Kopplung:

Schnittstellenspezifische Settingdaten:

- kein Vor- und Nachspann beim Auslesen.

Soll dieselbe Schnittstelle auch zur Kopplung mit anderen Geräten genutzt werden, können die entsprechenden Settingdaten vor der Übertragung umgesetzt und anschließend wieder zurückgesetzt werden.

Beispiel:

N.. @311 R700 K5016 L _F	SD 5016 wird in R700 gesichert
N.. @411 K5016 K00000010 L _F	SD 5016 umsetzen
N.. @a20 K1 L _F	1. V.24 anwählen
N.. @a27 K1 K50 K50 L _F	R50 wird ausgegeben
N.. @714 L _F	STOP DEC
N.. @411 K5016 R700 L _F	SD 5016 zurücksetzen.

@a29

Ausgabe ETX über V24

Mit diesem Befehl kann das im Settingdatum definierte Übertragungsendezeichen über die V24-Schnittstelle ausgegeben werden.

Beispiel:

N.. @a20 K1 L _F	Über die erste V24-Schnittstelle werden die R-Parameter
N.. @a27 K1 K0 K100 L _F	R0 bis R100 vom Kanal 1 und anschließend das Über-
N.. @a29 L _F	tragungsendezeichen ausgegeben.

Anwendung:

- Nach einer Folge von Ausgaben ohne Übertragungsendekennung wird dem Empfänger mit @a29 das Ende des Datenpaketes mitgeteilt.

11.11 Bedienerführungsmakro (BFM)

Hauptgruppe c/Untergruppe c: Rückübersetzung

@ccc

(Kennung für den Anfang des BFM-Datensatzes)

Die Versorgung von Bearbeitungszyklen kann mit Hilfe von projektierten Eingabebildern graphisch unterstützt werden. Wird ein solches Eingabebild vom Bediener mit neuen Bearbeitungsparametern versorgt, werden diese Werte zunächst in Eingabezwischenspeichern (EZS) abgelegt. Zur weiteren Verarbeitung der Eingabewerte wird durch Betätigung eines Softkeys ein Bedienerführungsmakro (BFM) gestartet.

Dieser BFM hat die Aufgabe, die Werte aus den EZS bestimmten Adressen zuzuordnen. Adressen können sein G, N, R, X, Y, Z, L, U, I, K usw. Durch das Zusammenführen von Adressen und EZS in einem BFM kann z. B. folgender Datensatz erstellt werden:

```
G01 G90 R10 = 50 R11 = 20 R12 = 15 L901 LF
```

Dieser Datensatz wird vom BFM in einem Teileprogramm als Programmsatz eingetragen. Das Programm muß zuvor im Editiermodus angewählt und die Einfügeposition mit dem Cursor markiert sein.

Die Funktion der Rückübersetzung bedeutet, daß ein mit BFM erzeugter Programmsatz in sein Eingabebild zurückgeführt wird. Es eröffnet sich dadurch die Möglichkeit des grafisch unterstützten Editierens.

Ein für die Rückübersetzung vorbereiteter Datensatz ist an zwei zusätzlichen Elementen zu erkennen:

- 1: @ccc (x:z:y) kennzeichnet den Anfang des Datensatzes. Die Achsnamen geben die Ebene an, für die der Datensatz programmiert wurde.
- 2: (BFM:EINSTECHEN) kennzeichnet das Ende des Datensatzes. "EINSTECHEN" ist der Name des BFM.

Ist ein solcher Satz im Teileprogramm (Editiermodus) mit dem Cursor markiert, wird nach Betätigung des Softkeys "EINGABE BILD" das zugehörige Eingabebild aufgerufen. Die im Datensatz enthaltene Information wird in die EZS kopiert und im Bild angezeigt. Der komplette Datensatz hat z. B. folgendes Aussehen:

```
@ccc (x:z:y) G01 G90 R10 = 50 R11 = 20 R12 = 15 L901 (BFM:EINSTECHEN1) LF
```

Wird die Reihenfolge der Parameter verändert, ist eine Rückübersetzung nicht mehr möglich. Meldung: "BFM nicht rückübersetzbar".

Projektierung der Rückübersetzbarkeit

Jeder BFM, der einen rückübersetzbaren Datensatz erzeugen soll, enthält im ersten Satz einen eindeutigen Namen. Der Name wird wie ein Kommentar programmiert und muß folgendes Format haben:

(BFM:EINSTECHEN1). "EINSTECHEN1" ist der Name.

Dieser Name wird in eine Zuordnungsliste eingetragen. Die Funktion der Zuordnungsliste erfüllt der BFM 999. In ihm wird dem Namen die BFM-Nummer und die Menüblocknummer (Nummer des projektierten Eingabebildes) zugeordnet.

Der Name darf 12 Zeichen lang sein. Blanks sind nicht erlaubt. Bei mehr als 12 Zeichen oder bei fehlendem Eintrag in der Zuordnungsliste (BFM 999) wird "BFM nicht gefunden" gemeldet.

Beispiel: Aufbau der Zuordnungsliste im BFM 999

```
% BFM 999
EINSTECHEN1 = 7, 11
EINSTECHEN2 = 9, 12
BEARBEITEN = 12, 13
KONUS = 4, 25
```

Die Anfangskennung @ccc wird erzeugt, wenn nach der BFM-Nr. der BFM-Name in Form von: (BFM:<Name>) projiziert wird. Ein solcher Datensatz kann später mit der Funktion "Rückübersetzung" graphisch editiert werden.

Beispiel: Aufbau des BFM 7

Projektierter Datensatz

```
% BFM 7
(BFM: EINSTECHEN 1)
N_150 G_100 G_101 R=_103 R12=_104 R13=_105 L_F
R14=_106 R15=_107 L901 L_F
```

Dieser BFM 7 würde an der mit dem Cursor im Teileprogramm markierten Stelle z. B. folgenden Datensatz einfügen:

```
@ccc (x:z:y) N2 G01 G90 R10=50 R11=20 R12=15 R13=25 L_F
R14=27 R15=39 L901 (BFM:EINSTECHEN1) L_F
```

Hinweis:

- Die max. Satzlänge von 120 Zeichen und die max. BFM-Namenslänge von 12 Zeichen muß eingehalten werden.
- Die erste Anweisung sollte eine Satznummer ("N_ ..."), um zum späteren Editieren das Auffinden des Datensatzes zu erleichtern.
- Das BFM darf jedoch im Teileprogramm am Satzanfang keine Satznummer und keinen Kommentar erzeugen.
- Zum Löschen des Datensatzes wird der Cursor in den zu löschen Datensatz positioniert, "N+" eingegeben und die Taste "CANCEL" betätigt bzw. nach dem Positionieren des Cursors mit dem projizierten Softkey "Block löschen" der Vorgang gestartet.

11.12 @-Code-Tabelle

Zeichenerklärung:

- y Vergleichsoperator vop>
 0: keine Bedingung
 1:= gleich
 2: ungleich
 3: größer
 4: = größer gleich
 5: kleiner
 6: = kleiner gleich
 7: wahr (true)
 8: nicht (not)

- 1) nicht auf CL800-Ebene
 2) "Bedingung":
 a) Var =Boolsche Variable
 b) Var . Const =Bit aus Pattern
 c) Var "Vop" Wert
 d) Erweiterte Bedingung
 3) Keine Pointer möglich,
 auf CL 800-Ebene nur Const vorgebar

@-Code	CL-800-Anweisung	Funktion
@00f		Freigabe für Softkeystart
@041 R Par 1 R Par 2	(Push Block) ¹⁾	Sichern einer Gruppe lokaler R-Parameter auf den Stack
@042 Const R Par n . . . R Par 1	(Pop) ¹⁾	Gesicherte R-Parameter vom Stack holen
@043 R Par 1 R Par 2	(Pop Block) ¹⁾	Gruppe der gesicherten R-Parameter vom Stack holen
@100 Const @100 R Par ³⁾	GOTO Label ;	Absoluter Sprung zu NC-Satz
@111 Var Wert 1 Const 1 Wert 2 Const 2 . . Wert n Const n	CASE Var = Wert 1 : Anweisung1 ; . . = Wert n : Anweisung n ;	CASE-Verzweigung
@12y Var Wert Const	IF "Bedingung" ²⁾ THEN Anweisung 1 ; [ELSE Anweisung 2 ;] END IF;	IF-THEN-ELSE-Anweisung y Vergleichsoperator Vop Var R-Parameter oder Pointer
@13y Var Wert Const	WHILE "Bedingung" ²⁾ DO Anweisung ;	Wiederholungs-Anweisung mit Abfrage der Wiederhol- bedingung am Anfang. y Vergleichsoperator Vop
@14y Var Wert Const	REPEAT Anweisung ; UNTIL "Bedingung" ²⁾	Wiederholungs-Anweisung mit Abfrage der Wiederhol- bedingung am Ende. y Vergleichsoperator Vop

@-Code	CL-800-Anweisung	Funktion
@151 <i>Var Wert 2 Const</i>	FOR <i>Var</i> = Wert 1 TO Wert 2 DO Anweisung ;	Wiederholungs-Anweisung mit Wiederholungen, solange bis <i>Var</i> inkrementell Wert 2 erreicht hat.
@161 <i>Var Wert 2 Const</i>	FOR <i>Var</i> = Wert 1 DOWNTO Wert 2 DO Anweisung ;	Wiederholungs-Anweisung mit Wiederholungen, solange bis <i>Var</i> dekrementell Wert 2 erreicht hat.
@200 <i>Var</i>	CLEAR(<i>Var</i>);	Lösche Variable
@201 <i>Var Wert</i>	<i>Var</i> = Wert	Lade Variable mit Wert
@202 <i>Var 1 Var 2</i>	XCHG (<i>Var 1</i> , <i>Var 2</i>);	Tauschen der Variableninhalte
@203 <i>Var 1 Var 2 Const</i>		Lesen eines Bits aus Bitmuster
@210 <i>Wert 3 Wert 4</i>	CLEAR MIB (<i>Wert 3</i> , <i>Wert 4</i>);	Lösche Eingabezwischenspeicher Wert 3: EZS-Anfangsadr. 0 ... 499 Wert 4: EZS-Endadresse 0 ... 499
@211 <i>Var Wert 1</i>	<i>Var</i> = MIB (<i>Wert</i>);	Lade numerische Variable <i>Var</i> mit dem Inhalt der EZS-Zelle <i>Wert 1</i> Wert 1: EZS-Nr. 0 ... 499
@212 <i>Wert 1 Wert</i>	MIB (<i>Wert 1</i>) = <i>Wert</i> ;	Lade EZS-Zelle <i>Wert 1</i> mit der num. Zahl <i>Wert</i> Wert 1: EZS-Nr. 0 ... 499
@300 <i>Var Wert 1</i>	<i>Var</i> =MDN (<i>Wert 1</i>);	Maschinendaten NC Wert 1: Adr. 0 ... 4999
@301 <i>Var Wert 1</i>	<i>Var</i> =MDNBY (<i>Wert 1</i>);	Maschinendaten-NC-Byte Wert 1: Byte-Adr. 5000 ... 6999
@302 <i>Var Wert 1 Wert 2</i>	<i>Var</i> =MDNBI (<i>Wert 1</i> , <i>Wert 2</i>);	Maschinendaten-NC-Bit Wert 1: Byte-Adr. 5000 ... 6999 Wert 2: Bit-Adr. 0 ... 7
@306 <i>Var Wert 1</i>	<i>Var</i> =MDP (<i>Wert 1</i>);	Maschinendaten-PLC Wert 1: Adr. 0 ... 1999
@307 <i>Var Wert 1</i>	<i>Var</i> =MDPBY (<i>Wert 1</i>);	Maschinendaten- PLC-Byte Wert 1: Byte-Adr. 2000 ... 3999

@-Code	CL-800-Anweisung	Funktion
@308 <i>Var Wert 1 Wert 2</i>	Var =MDPBI (Wert 1 , Wert 2);	Maschinendaten-PLC-Bit Wert 1: Byte-Adr. 2000 . . . 3999 Wert 2: Bit-Adr. 0 . . . 7
@310 <i>Var Wert 1</i>	Var =SEN (Wert 1);	Settingdaten-NC Wert 1: Adr. 0 . . . 4999
@311 <i>Var Wert 1</i>	Var =SENB (Wert 1);	Settingdaten-NC-Byte Wert 1: Byte-Adr. 5000 . . . 9999
@312 <i>Var Wert 1 Wert 2</i>	Var =SENBI (Wert 1 , Wert 2);	Settingdaten-NC-Bit Wert 1: Byte-Adr. 5000 . . . 9999 Wert 2: Bit-Adr. 0 . . . 7
@320 <i>Var Wert 1 Wert 2 Wert 3</i>	Var =TOS (Wert 1 , Wert 2 , Wert 3);	Werkzeugkorrektur Wert 1: 0 Wert 2: D-Nr. 1 . . 99 Wert 3: P-Nr. 0 . . (9)
@330 <i>Var Wert 1 Wert 2 Wert 3</i>	Var =ZOA (Wert 1 , Wert 2 , Wert 3);	Einstellbare Nullpunktver- schiebung (G54-G57) Wert 1: Gruppe 1 . . . 4 (G54-G57) Wert 2: Ach-Nr. 1, 2 . . . Wert 3: Grob/Fein (0/1)
@331 <i>Var Wert 1 Wert 2</i>	Var =ZOPR (Wert 1 , Wert 2);	Programmierbare Nullpunkt- verschiebung (G58,G59) Wert 1: Gruppe 1 oder 2 (G58 oder G59) Wert 2: Achs-Nr. 1, 2 . . .
@332 <i>Var Wert 2</i>	Var =ZOE (Wert 2);	externe Nullpunktverschie- bung von der PLC Wert 2: Achs-Nr. 1, 2 . . .
@333 <i>Var Wert 2</i>	Var =ZOD (Wert 2);	DRF-Verschiebung Wert 2: Achs-Nr. 1, 2 . . .
@334 <i>Var Wert 2</i>	Var =ZOPS (Wert 2);	PRESET-Verschiebung Wert 2: Achs-Nr. 1, 2 . . .
@336 <i>Var Wert 2</i>	Var =ZOS (Wert 2);	Summen-Verschiebung Wert 2: Achs-Nr. 1, 2 . . .
@337 <i>Var Wert 1 Wert 2 Wert 3</i>	Var =ZOADW (Wert 1 , Wert 2 , Wert 3);	Einstellbare Koordinaten- drehung (G54-G57) Wert 1: Kanal-Nr. 0 . . . 2 Wert 2: Gruppe 1 . . . 4 (G54- G57) Wert 3: Winkel-Nr. (=1)

@-Code	CL-800-Anweisung	Funktion
@338 Var Wert 1 Wert 2 Wert 3	Var =ZOPRDW (Wert 1 , Wert 2 , Wert 3);	Programmierbare Koordi- natendrehung (G58-G59) Wert 1: Kanal-Nr. 0 . . . 2 Wert 2: Gruppe 1 oder 2 (G58 oder G59) Wert 3: Winkel-Nr. (=1)
@342 Var Wert 1 Wert 3	Var =PRSS (Wert 1 , Wert 3);	Programmierte Spindel- drehzahl lesen Wert 1: Kanal-Nr. 0 . . . 2 Wert 3: Spindel-Nr. 0 . . 6
@345 Var Wert 1 Wert 2	Var =PRVC (Wert 1 , Wert 2);	Programmierte Schnittge- schwindigkeit Wert 1: Kanal-Nr. 0 . . . 2 Wert 2: 0=G 96
@360 Var Wert 2	Var =ACPW (Wert 2);	Achs-Position-Ist werkstück- bezogen Wert 2: Achs-Nr. 1, 2 . . .
@361 Var Wert 2	Var =ACPM (Wert 2);	Achs-Position-Ist maschinen- bezogen Wert 2: Achs-Nr. 1, 2 . . .
@363 Var Wert 2	Var =ACSP (Wert 2);	Spindelpositions-Istwert Wert 2: Spindel-Nr. 0 . . 6
@364 Var Wert 2	Var =ACSS (Wert 2);	Spindeldrehzahl-Istwert Wert 2: Spindel-Nr. 0 . . 6
@367 Var Wert 1	Var =ACAS (Wert 1);	Achsnummer der aktuellen Ebene/Leitspindelnummer in R-Parameter Var lesen: Var+0: Nr. der waag- rechten Achse Var+1: Nr. der senkrechten Achse Var+2: Nr. der Achse senkrecht zur Ebene Var+3: Nr. der Achse, in der Länge 2 wirkt (Werkzeugtyp 30) Var+4: Nr. der Leitspindel Wert 1: Kanal-Nr. 0 . . . 2
@36a Var Wert 1	Var =ACD (Wert 1);	D-Funktion-Ist Wert 1=0

@-Code	CL-800-Anweisung	Funktion
@36b <i>Var Wert 1 Wert 3</i>	Var =ACG (Wert 1 , Wert 3);	Lesen der G-Funktion aus dem Arbeitsspeicher des aktuellen Satzes Wert 1: Kanal-Nr. 0 . . . 2 Wert 3: interne G-Gruppe, zu der G-Funktion gehört 0 . . . 15
@371 <i>Var Wert 1 Wert 3</i>	Var =SOB (Wert 1 , Wert 3);	Sonderbits Wert 1: Kanal-Nr. 0 . . . 2 =Kanalabhängig 99=Kanalunabhängig Wert 3: Bit-Nr. 0 . . . 7
@380 <i>Var Wert 1 Wert 2 Wert 3</i>	Var =PLCI (Wert 1 , Wert 2 , Wert 3);	PLC-Eingangsbit Wert 1: PLC-Nr. 1 Wert 2: Byteadr. 0...255 Wert 3: Bitnr. 0...7
@381 <i>Var Wert 1 Wert 2 Wert 3</i>	Var =PLCQ (Wert 1 , Wert 2 , Wert 3);	PLC-Ausgangsbit Wert 1: PLC-Nr. 1 Wert 2: Byteadr. 0...255 Wert 3: Bitnr. 0...7
@382 <i>Var Wert 1 Wert 2 Wert 3</i>	Var =PLCF (Wert 1 , Wert 2 , Wert 3);	PLC-Merkerbit Wert 1: PLC-Nr. 1 Wert 2: Byteadr. 0...255 Wert 3: Bitnr. 0...7
@383 <i>Var Wert 1 Wert 2 Wert 3</i>	Var =PLCF (Wert 1 , Wert 2 , Wert 3 , Wert 4);	PLC-Datenwortbit Wert 1: PLC-Nr. 1 Wert 2: DB-Nr. 0...255 Wert 3: DW-Nr. 0...255 Wert 4: Bitnr. 0...15
@3e4 <i>Var Wert 1</i>	Var =AGS (Wert 1);	Aktive Getriebestufe lesen Wert 1: Spindel-Nr. 0 bis 6

@-Code	CL-800-Anweisung	Funktion
@400 Wert 1 Wert	MDN (Wert 1)= Wert ;	Maschinendaten-NC Wert 1: Adr. 0 .. 4999
@401 Wert 1 Wert	MDNBY (Wert 1)= Wert ;	Maschinendaten-NC-Byte Wert 1: Byte-Adr. 5000 ... 6999
@402 Wert 1 Wert 2 Wert	MDNBI (Wert 1 , Wert 2) = Wert ;	Maschinendaten-NC-Bit Wert 1: Byte-Adr. 5000 ... 6999 Wert 2: Bit-Adr. 0 ... 7
@406 Wert 1 Wert	MDP (Wert 1)= Wert ;	Maschinendaten-PLC Wert 1: Adr. 0 .. 1999
@407 Wert 1 Wert	MDPBY (Wert 1)= Wert ;	Maschinendaten-PLC-Byte Wert 1: Byte-Adr. 2000 ... 3999
@408 Wert 1 Wert 2 Wert	MDNBI (Wert 1 , Wert 2) = Wert ;	Maschinendaten-PLC-Bit Wert 1: Byte-Adr. 2000 ... 3999 Wert 2: Bit-Adr. 0 ... 7
@410 Wert 1 Wert	SEN (Wert 1)= Wert ;	Settingdaten-NC Wert 1: Adr. 0 .. 4999
@411 Wert 1 Wert	SENBY (Wert 1)= Wert ;	Settingdaten-NC-Bytes Wert 1: Byte-Adr. 5000 ... 9999
@412 Wert 1 Wert 2 Wert	SENBI (Wert 1 , Wert 2) = Wert ;	Settingdaten-NC-Bits Wert 1: Byte-Adr. 5000 ... 9999 Wert 2: Bit-Adr. 0 ... 7
@420 Wert 1 Wert 2 Wert 3 Wert	TOS (Wert 1 , Wert 2 , Wert 3)= Wert ;	Werkzeugkorrektur Wert 1: 0 Wert 2: D-Nr. 1 .. 99 Wert 3: P-Nr. 0 .. 7 (9)
@423 Wert 1 Wert 2 Wert 3 Wert	TOAD (Wert 1 , Wert 2 , Wert 3)= Wert ;	Werkzeugkorrektur additiv Wert 1: 0 Wert 2: D-Nr. 1 .. 99 Wert 3: P-Nr. 0 .. 7 (9)
@430 Wert 1 Wert 2 Wert 3 Wert	ZOA (Wert 1 , Wert 2 , Wert 3)= Wert ;	Einstellbare Nullpunktver- schiebung (G54-G57) Wert 1: Gruppe 1 ... 4 (G54-G57) Wert 2: Achs-Nr. 1, 2 ... Wert 3: Grob/Fein (0/1)

@-Code	CL-800-Anweisung	Funktion
@431 Wert 1 Wert 2 Wert 3 Wert	ZOFA (Wert 1 , Wert 2 , Wert 3)= Wert ;	Einstellbare Nullpunktver- schiebung additiv Wert 1: Gruppe 1 . . . 4 (G54-G57) Wert 2: Achs-Nr. 1, 2 . . . Wert 3: Grob/Fein (0/1)
@432 Wert 1 Wert 2 Wert	ZOPR (Wert 1 , Wert 2)= Wert ;	Programmierbare Nullpunkt- verschiebung (G58,G59) Wert 1: Gruppe1 oder 2 (G58 oder G59) Wert 2: Achs-Nr. 1, 2 . . .
@434 Wert 2 Wert	ZOD (Wert 2)= Wert ;	DRF-Verschiebung Wert 2: Achs-Nr. 1, 2 . . .
@435 Wert 2 Wert	ZOPS (Wert 2)= Wert ;	PRESET-Verschiebung Wert 2: Achs-Nr. 1, 2 . . .
@437 Wert 1 Wert 2 Wert 3 Wert	ZOADW (Wert 1 , Wert 2 , Wert 3)= Wert ;	Einstellbare Koordinaten- drehung absolut Wert 1: Kanal-Nr. 0 . . . 2 Wert 2: Gruppe 1 . . . 4 (G54-G57) Wert 3: Winkel-Nr. (=1)
@438 Wert 1 Wert 2 Wert 3 Wert	ZOFADW (Wert 1 , Wert 2 , Wert 3) = Wert ;	Einstellbare Koordinaten- drehung additiv Wert 1: Kanal-Nr. 0 . . . 2 Wert 2: Gruppe 1 . . . 4 Wert 3: Winkel-Nr. (=1)
@439 Wert 1 Wert 2 Wert 3 Wert	ZOPRDW (Wert 1 , Wert 2 , Wert 3)= Wert ;	Programmierbare Koordina- tendrehung Wert 1: Kanal-Nr. 0 . . . 2 Wert 2: Gruppe 1 oder 2 (G58 oder G59) Wert 3: Winkel-Nr. (=1)
@43a Wert 1 Wert 2 Wert 3 Wert	ZOFPROW (Wert 1 , Wert 2 , Wert 3)= Wert ;	Programmierbare Koordina- tendrehung additiv Wert 1: Kanal-Nr. 0 . . . 2 Wert 2: Gruppe 1 oder 2 (G58 oder G59) Wert 3: Winkel-Nr. (=1)
@440 Wert 3 Wert	PRAP (Wert 3)= Wert ;	Programmierte Achsposition Wert 3: Achs-Nr. 1, 2 . . .
@442 Wert 3 Wert	PRSS (Wert 3)= Wert ;	Programmierte Spindel- drehzahl Wert 3: Spindel-Nr. 0 . . . 6
@446 Wert	PRAD= Wert ;	Programmierter Radius

@-Code	CL-800-Anweisung	Funktion
@447 Wert	PANG= Wert ;	Programmierter Winkel
@448 Wert 3 Wert	PRIP (Wert 3)= Wert ;	Programmierter Interpolationsparameter für Kreis und Gewinde Wert 3: Achs-Nr. 1,2,...
@482 Wert 1 Wert 2 Wert 3 Wert	PLCF (Wert 1 , Wert 2 , Wert 3) = Wert ;	PLC-Merkerbit Wert 1: PLC-Nr. 1 Wert 2: Byteadr. 100...115, 136...199, 224...255 Wert 3: Bitnr. 0...7
@483 Wert 1 Wert 2 Wert 3 Wert 4 Wert	PLCW (Wert 1 , Wert 2 , Wert 3) = Wert ;	PLC-Datenwortbit Wert 1: PLC-Nr. 1 Wert 2: DB-Nr. 1...255 Wert 3: DW-Nr. 0...255 Wert 4: Bitnr. 0...15
@4e1 Wert 1 Wert 2 Wert	SATC (Wert 1 , Wert 2)= Wert ;	Spindelbeschleunigungszeitkonstante Wert 1: Spindel-Nr. 0 bis 6 Wert 2: Getriebestufe 1 bis 8 Wert : Spindelbeschleunigungszeitkonstante 0 bis 16000
Var = Wert 1 + Wert 2 ; Var = Wert 1 - Wert 2 ; Var = Wert 1 · Wert 2 ; Var = Wert 1 / Wert 2 ;	Var = Wert 1 + Wert 2 ; Var = Wert 1 - Wert 2 ; Var = Wert 1 · Wert 2 ; Var = Wert 1 / Wert 2 ;	Addition Subtraktion Multiplikation Division
@610 Var Wert	Var =ABS (Wert);	Betragsbildung
@613 Var Wert	Var =SQRT (Wert);	Quadratwurzel
@614 Var Wert 1 Wert 2	Var =SQRTS (Wert 1 , Wert 2);	Wurzel aus Quadratsumme
@620 Var	INC (Var);	Inkrementieren von "Var" um 1
@621 Var	DEC (Var);	Dekrementieren von "Var" um 1
@622 Var	TRUNC (Var);	Ganzzahliger Anteil
@630 Var Wert	Var =SIN (Wert);	Sinus
@631 Var Wert	Var =COS (Wert);	Cosinus
@632 Var Wert	Var =TAN (Wert);	Tangens
@634 Var Wert	Var =ARCSIN (Wert);	ArcusSinus
@637 Var Wert 1 Wert 2	Var =ANGLE (Wert 1 , Wert 2);	Winkel aus zwei Vektorkomponenten

@-Code	CL-800-Anweisung	Funktion
@640 <i>Var Wert</i>	<i>Var</i> =LN (Wert);	nat. Logarithmus
@641 <i>Var Wert</i>	<i>Var</i> =INV LN (Wert);	e ^x Exponentialfunktion
@650 <i>Var Var 1 Wert</i>	<i>Var</i> = Var 1 OR Wert ;	ODER
@651 <i>Var Var 1 Wert</i>	<i>Var</i> = Var 1 XOR Wert ;	EXKLUSIV-ODER
@652 <i>Var Var 1 Wert</i>	<i>Var</i> = Var 1 AND Wert ;	AND
@653 <i>Var Var 1 Wert</i>	<i>Var</i> = Var 1 NAND Wert ;	NAND
@654 <i>Var Wert</i>	<i>Var</i> =NOT Wert ;	NOT
@655 <i>Var Var 1 Wert</i>	<i>Var</i> = Var 1 ORB Wert ;	ODER-Bit
@656 <i>Var Var 1 Wert</i>	<i>Var</i> = Var 1 XORB Wert ;	EXKLUSIV-ODER-Bit
@657 <i>Var Var 1 Wert</i>	<i>Var</i> = Var1 ANDB Wert ;	AND-Bit
@658 <i>Var Var 1 Wert</i>	<i>Var</i> = Var 1 NANDB Wert ;	NAND-Bit
@659 <i>Var Wert</i>	<i>Var</i> =NOTB Wert	NOT-Bit
@660 <i>Var Const</i>	CLEAR BIT (<i>Var</i> . Const);	Lösche Bit in Pattern Const=Bit-Nr. 0 . . . 7
@661 <i>Var Const</i>	SET BIT (<i>Var</i> . Const);	Setze Bit ;Const=Bit-Nr.0..7
@67y <i>Var 1 Var 2 Wert</i>		Ist der Vergleich von Var 2 und Wert erfüllt, wird die boolsche Variable Var 1 auf „1“ gesetzt.
@706	POS MSYS;	Vorgabe einer Position be- zogen auf das Maschinent- Istwert-System
@710 <i>Var 1 Var 2</i>	<i>Var 1</i> =PREP REF (<i>Var 2</i>);	Referenzauflbereitung Var 1: Ausg.daten ab Var 1 Var 2: Eing.datum ab Var 2
@711 <i>Var 1 Var 2 Var 3</i>	<i>Var 1</i> =INT SEC (<i>Var 2</i> <i>Var 3</i>);	Schnittpunktberechnung Var 1: Ausg.daten ab Var 1 Var 2: erste Kontur ab Var 2 Var 3: mit 0 vorbesetzen
@713 <i>Var</i>	<i>Var</i> =PREP CYC;	Startvorbereitung f. Zyklen Var: Ausgangsdaten ab Var
@714	STOP DEC;	Stop der Decodierung, bis Zwischenspeicher leer ist.
@715	STOP DEC1;	Stop der Decodierung, bis Zwischenspeicher leer ist. (bei Koordinatendrehung)

@-Code	CL-800-Anweisung	Funktion
@720 <i>Var Wert</i>	Var =MEAS M Wert	Fliegendes Messen Var: Daten hinterlegt ab Var Wert: Nr. des Meßeingangs: 1 oder 2
@a15 <i>Wert 4 Wert 5</i>	WRT PIC (Wert 1 , Wert 2 , Wert 3);	Anwahl von Bildern und Menütexen Wert4: 0=Anwenderber. 1=Standardber. Wert 5: 1...254 Menünr.
@a1b	RECALL PIC	Rücksprung in das Ausgangsmenü.
@a20 <i>Wert</i>	PORT Wert	Anwahl der V24-Schnittstelle Wert: Nummer der Schnittstelle 1 oder 2
@a25 <i>Wert 1</i>	OUTP ZOA (Wert 1);	Ausgabe Nullpunktverschiebungen über V24 Wert 1: Kanal-Nr. 0 . . . 3 0= NV G54 - G57 1= Drehwinkel Kanal 1 2= Drehwinkel Kanal 2 3= Drehwinkel Kanal 3
@a26 <i>Wert 2 Wert 3 Wert 4</i>	OUTP DATA (Wert 2 Wert 3 Wert 4);	Ausgabe Daten über V24 Wert 2: Kennung der Datenart 1= Hauptprogramm 2= Unterprogramm 5= NC-Maschinendaten 6= Werkzeugkorrekturen 8= Settingdaten 9= PLC-Maschinendaten Wert 3: Anfangsadresse Wert 4: Endadresse

@-Code	CL-800-Anweisung	Funktion
@a27 Wert 1 Wert 3 Wert 4	OUTP PARA (Wert 1 Wert 3 Wert 4);	Parameterausgabe über V 24 Wert 1: Kanalnummer Wert 2: Anfangsadresse Wert 3: Endadresse
@a28 Wert 2	INP (Wert 2);	Einlesen über V24 Wert 2: 0=keine Daten- typprüfung 10=R-Parameter
@a29	OUTP ETX	Ausgabe des Übertragungs- endezeichens über V24
@ccc		Rückübersetzung in die Eingabemaske. Kennung für den Anfang des "BFM-Datensatzes"

12 Programmschlüssel SINUMERIK 810T

12.1 Interne G-Gruppeneinteilung bei @36b

Interne G- Gruppe	G-Funktionen														
	Hex-Code														
	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	01 ^T	10	11	02	03	33	34	35	06	12	13	36		
1	09														
2	17	18 ^T	19	16											
3	40 ^T	41	42												
4	53														
5	54 ^T	55	56	57											
6	04	25	26	58	59	92	74								
7	60	63	64 ^T	62											
8	70	71													
9	80 ^T	81	82	83	84	85	86	87	88	89					
10	90 ^T	91	68												
11	94	95 ^T	96	97	98										
12	147	247	347	148	248	348	48	110	111						
13	50 ^T	51													

T: Grundstellung (Löschstellung) der G-Gruppe

12.2 Programmschlüssel

Gruppe	EIA	ISO	Wertebereich/Code	Funktion und Bedeutung
	EOR	%	0 bis 9999	Programmnummer
	mpf ... spf EOB	MPF ... SPFLF	0 bis 9999 1 bis 9999	Hauptprogramm Unterprogramm
	o n /o /n	: N /: /N	1 bis 9999 . . .	Hauptsatz Nebensatz Ausblendbarer Hauptsatz Ausblendbarer Nebensatz
G0	g	G	00 01 * 02 03 06 10 11 12 13 33 34 35 36	Eilgang, Genau-Halt grob Geradeninterpolation Kreisinterpolation im Uhrzeigersinn Kreisinterpolation im Gegenuhrzeigersinn SPLINE-Interpolation Polarkoordinatenprogrammierung, Eilgang Polarkoordinatenprogrammierung, Geradeninterpolation Polarkoordinatenprogrammierung, Kreisinterpolation im Uhrzeigersinn Polarkoordinatenprogrammierung, Kreisinterpolation im Gegenuhrzeigersinn Gewindeschneiden, Steigung konstant Gewindeschneiden, Steigung linear ansteigend Gewindeschneiden, Steigung linear abnehmend Gewindebohren ohne Ausgleichsfutter
G1	g	G	09 #	Geschwindigkeitsabnahme, Genau-Halt fein
G2	g	G	16 17 18 * 19	Ebenenwahl mit freier Achsenwahl Ebenenwahl X - Y (abhängig von Maschinendatum) Ebenenwahl Z - X (abhängig von Maschinendatum) Ebenenwahl Y - Z (abhängig von Maschinendatum)
G3	g	G	40 * 41 42	Keine Schneidenradiusbahn-Korrektur Schneidenradiusbahn-Korrektur links Schneidenradiusbahn-Korrektur rechts
G4	g	G	53 #	Unterdrückung der Nullpunktverschiebung
G5	g	G	54 * 55 56 57	Nullpunktverschiebung 1 Nullpunktverschiebung 2 Nullpunktverschiebung 3 Nullpunktverschiebung 4
G6	g	G	04 # 1) 25 # 1) 26 # 1) 58 # 1) 59 # 1) 74 # 1) 92 1)	Verweilzeit, zeitlich vorbestimmt unter Adresse X oder F in Sekunden und Adresse S in Umdrehungen. Der Zeitbereich liegt zwischen: 0.001 bis 99999.999 s bei X 0.001 bis 99.999 s bei F 0.1 bis 99.9 bei S minimale Arbeitsfeldbegrenzung maximale Arbeitsfeldbegrenzung Programmierbare Nullpunktverschiebung 1 Programmierbare Nullpunktverschiebung 2 Referenzpunktfahren über Teileprogramm (PP) Spindeldrehzahl-Sollwertbegrenzung unter Adresse S
G7	g	G	60 62 63 64 *	Geschwindigkeitsabnahme, Genau-Halt fein Bahnsteuerbetrieb, Satzübergang mit Geschwindigkeitsreduzierung Gewindebohren ohne Geber, Vorschubkorrektur 100% Bahnsteuerbetrieb, Satzübergang ohne Geschwindigkeitsreduzierung
G8	g	G	70 71	Eingabesystem Zoll Löschstellung über Maschinendaten Eingabesystem metrisch Löschstellung über Maschinendaten

Gruppe	EIA	ISO	Code/Wertebereich	Funktion und Bedeutung
G9	g	G	80 * 81 82 83 84 85 86 87 88 89	Löschen G81 bis G89 Aufruf Zyklus L81 - Bohren, Zentrieren achsumschaltb. Aufruf Zyklus L82 - Bohren, Plansenken achsumschaltb. Aufruf Zyklus L83 - Tieflochbohren achsumschaltb. Aufruf Zyklus L84 - Gewindebohren m. Geber achsumschaltb. Aufruf Zyklus L85 - Ausbohren 1 achsumschaltb. Aufruf Zyklus L86 - Ausbohren 2 achsumschaltb. Aufruf Zyklus L87 - Ausbohren 3 achsumschaltb. Aufruf Zyklus L88 - Ausbohren 4 achsumschaltb. Aufruf Zyklus L89 - Ausbohren 5 achsumschaltb.
G10	g	G	90 * 91 68	Bezugmaßangabe Kettenmaßangabe Bezugsmaßangabe auf kürzestem Weg (nur bei Rundachse)
G11	g	G	94 * 95 96 97 98	Vorschub unter Adresse F in mm/min oder inch/min Vorschub unter Adresse F in mm/U oder inch/U Vorschub unter Adresse F in mm/U oder inch/U und konstante Schnittgeschwindigkeit unter Adresse S in m/min oder ft/min Aufheben G96, letzten Drehzahlsollwert von G96 abspeichern Vorschub für rotatorische Achse in U/min
G12	g	G	110 111 147 # 1) 247 # 1) 347 # 1) 148 # 1) 248 # 1) 348 # 1) 48 # 1)	Polarkoordinatenprogrammierung, erreichte programmierte Sollposition als neuen Mittelpunkt übernehmen Polarkoordinatenprogrammierung, Mittelpunktprogrammierung mit Winkel und Radius ohne Achsbewegung Weiches Anfahren der Kontur linear Weiches Anfahren der Kontur im Viertelkreis Weiches Anfahren der Kontur im Halbkreis Weiches Verlassen linear Weiches Verlassen im Viertelkreis Weiches Verlassen im Halbkreis Gleiches Verlassen wie Anfahren
G13			50 * 51	Abwahl der Maßstabsänderung Maßstabsänderung
	x 4)	X 4)	± 0.001 bis± 99999.999 ± 0.0001 bis± 3999.9999 0.001 bis 99999.999 ± 0.001° bis± 99999.999°	Weginformation in mm Weginformation in inch Verweilzeit in s Weginformation in Grad
	y 4)	Y 4)	± 0.001 bis± 99999.999 ± 0.0001 bis± 3999.9999 0.001 bis 99999.999 ± 0.001° bis± 99999.999°	Weginformation in mm Weginformation in inch Verweilzeit in s Weginformation in Grad
	z 4)	Z 4)	± 0.001 bis± 99999.999 ± 0.0001 bis± 3999.9999 ± 0.001° bis± 99999.999°	Weginformation in mm Weginformation in inch Weginformation in Grad
	q 4)	Q 4)	± 0.001 bis± 99999.999 ± 0.0001 bis± 3999.9999 ± 0.001° bis± 99999.999°	Hilfsachsen Weginformation in mm Hilfsachsen Weginformation in inch Weginformation in Grad

- 1) Es dürfen keine weiteren Funktionen in diesem Satz geschrieben werden
3) Andere Achsen wählbar (A, B, C, E, U, V, W)
Satzweise wirksam, alle übrigen selbsthaltend
* Löschstellung (Grundstellung, nach Reset, M02/M30, nach Einschalten der Steuerung)
4) Durch Maschinendatum wird der Wertebereich eingestellt. Vom Maschinenhersteller erfahren Sie den gültigen Wertebereich.

12.2 Programmschlüssel

Gruppe	EIA	ISO	Wertebereich	Funktion und Bedeutung
	a 3)	A 3)	0 bis 359.99999°	Winkel in Grad bei Konturzug
	b 3) 4)	B 3) 4)	± 0.001 bis + 99999.999 ± 0.0001 bis + 3999.9999 + 0 bzw. - 0 - 0.001 bis - 99999.999 - 0.0001 bis - 3999.9999 + 0.001 bis + 99999.999 + 0.0001 bis + 3999.9999	Radius bei Kreisinterpolation in mm Radius bei Kreisinterpolation in inch Ecke bei Konturzug Fase bei Konturzug in mm Fase bei Konturzug in inch Radius bei Konturzug in mm Radius bei Konturzug in inch
	c 4)	C 4)	± 0.001 bis± 99999.999 ± 0.0001 bis± 3999.9999 ± 0.001° bis± 99999.999°	Hilfs-/Rundachsen Weginformation in mm Hilfs-/Rundachsen Weginformation in inch Weginformation in Grad
	d	D	1 bis 99 0	Anwahl der Werkzeugkorrektur Abwahl der Werkzeugkorrektur
	f 4)	F 4)	0.01 bis 45000 0.1 bis 1770.000 0.001 bis 50.000 0.0001 bis 2.0000 0.001 bis 16.000 0.0001 bis 6.0000 0.001 bis 99999.999	Vorschub in mm/min Vorschub in inch/min Vorschub in mm/U Vorschub in inch/U Gewinde. Steigungszu- bzw. -abnahme in mm/U Gewinde. Steigungszu- bzw. -abnahme in inch/U Verweilzeit in s
	h	H	1 bis 9999	Hilfsfunktionen
	i 4)	I 4)	± 0.001 bis ± 99999.999 ± 0.0001 bis ± 3999.9999 0.001 bis 400.000 0.0001 bis 16.000	Interpolationsparameter für X-Achse in mm Interpolationsparameter für X-Achse in inch Gewindesteigung in mm Gewindesteigung in inch
	j 4)	J 4)	± 0.001 bis ± 99999.999 ± 0.0001 bis ± 3999.9999 0.001 bis 400.000 0.0001 bis 16.000	Interpolationsparameter für Y-Achse in mm Interpolationsparameter für Y-Achse in inch Gewindesteigung in mm Gewindesteigung in inch
	k 4)	K 4)	± 0.001 bis ± 99999.999 ± 0.0001 bis ± 3999.9999 0.001 bis 400.000 0.0001 bis 16.000	Interpolationsparameter für Z-Achse in mm Interpolationsparameter für Z-Achse in inch Gewindesteigung in mm Gewindesteigung in inch
	l	L	1 bis 9999	Unterprogramm-Nummer
	p	P	1 bis 99	Anzahl der Durchläufe des Unterprogramms Verhältnis P = Arbeitsdurchmesser/Einheitsdurchmesser bei Zylinderinterpolation
	r	R	0 bis 49 50 bis 99 110 bis 699 700 bis 999	Übergabeparameter Rechenparameter Kanalabhängig deklarierte Parameter Zentrale Parameter
	s	S	1 bis 16000 1 bis 16000 0.5 bis 359.5 0.1 bis 99.9	Spindeldrehzahl in min ⁻¹ oder in 0.1 min ⁻¹ und konstante Schnittgeschwindigkeit in m/min oder 0.1 m/min oder ft/min oder 0.1 ft/min Spindeldrehzahlbegrenzung in min ⁻¹ oder 0.1 min ⁻¹ Spindelhalt in Grad (M19), Abstand von der Nullmarke des Gebers Verweilzeit in U/min
	t	T	1 bis 9999	Werkzeugnummer

1) Es dürfen keine weiteren Funktionen in diesem Satz geschrieben werden

3) Andere Achsen wählbar (A, B, C, E, U, V, W)

Satzweise wirksam, alle übrigen selbsthaltend

* Löschstellung (Grundstellung, nach Reset, M02/M30, nach Einschalten der Steuerung)

4) Durch Maschinendatum wird der Wertebereich eingestellt. Vom Maschinenhersteller erfahren Sie den gültigen Wertebereich.

Gruppe	EIA	ISO	Code	Funktion und Bedeutung	
M1	m	M	00 # 01	Programmierter Halt, unbedingt Programmierter Halt, bedingt	
M2	m	M	02 17 30	Programmende, steht im letzten Satz des Programms Unterprogrammende, steht im letzten Satz des Unterprogramms, ohne Halt bei Wiederholddurchläufen Programmende, steht im letzten Satz des Programms	
M3	m	M	03 04 05 * 19	Spindeldrehrichtung rechts Spindeldrehrichtung links Spindelhalt, ohne Orientierung Orientierter Spindelhalt, Winkel unter Adresse S in Grad	
M4	m	M	36 37	Vorschub wie unter F programmiert Vorschub in mm/min oder mm/U 1:100 untersetzt	wirkt auch bei G33
M5	m	M	0 bis 99	Zusatzfunktionen, frei belegbar außer Gruppen M1 bis M4	
	I	L	70...	Zyklus prozeßnahes Messen (fliegend)	
	I	L	91/92	Rückzugszyklen für Werkzeugwechsel	
	I	L	93	Einstechzyklus	
	I	L	95	Abspannzyklus gegen die Kontur, achsparallel	
	I	L	97	Gewindeschneidzyklus mit Flankenzustellung	
	I	L	98	Tieflochbohrzyklus	
	I	L	99	Ketten von Gewinden (Vier-Punkte-Gewindeschneidzyklus)	
	I	L	999	Zwischenspeicher leeren	
	@	@			
	=	=		Trennzeichen, zwingend notwendig bei Adreßerweiterungen z.B. R35 = 123.5	
	+	+		Addition bei Parametern	
	-	-		Subtraktion bei Parametern	
	*	*		Multiplikation bei Parametern	
	/	/		Division bei Parametern	
	5-4-2 2) 7-4-2 2)	()		Anmerkungsbeginn Anmerkungsende	
	EOB	LF		Satzende	

2) *Gelochte Spuren*

Satzweise wirksam, alle übrigen selbsthaltend

* *Löschstellung (Grundstellung, nach Reset, M02/M30, nach Einschalten der Steuerung)*

An
Siemens AG

AUT V250
Postfach 4848
W-8500 Nürnberg 1

Vorschläge

Korrekturen

für Druckschrift:

SINUMERIK 810T, GA3, SW3
Bedienen und Programmieren
Anwender-Dokumentation

Absender

Name _____

Anschrift Ihrer Firma/Dienststelle

Straße: _____

PLZ: _____ Ort: _____

Telefon: _____ / _____

Benutzeranleitung

Bestell-Nr.: 6ZB5 410-0EP01-0BA2

Ausgabe: 01.93

Sollten Sie beim Lesen dieser Unterlage auf Druckfehler gestoßen sein, bitten wir Sie, uns diese mit diesem Vordruck mitzuteilen. Ebenso dankbar sind wir für Anregungen und Verbesserungsvorschläge.

Vorschläge und/oder Korrekturen

Herausgegeben von Siemens AG
Bereich Automatisierungstechnik
Geschäftsgebiet Automatisierungssysteme
für Werkzeugmaschinen, Roboter
und Sondermaschinen
Postfach 4848, W-8500 Nürnberg 1

Siemens Aktiengesellschaft

© Siemens AG 1990 All Rights Reserved
Änderungen vorbehalten

Bestell-Nr. 6ZB5 410-0EP01-0BA2
Printed in the Fed. Rep. of Germany

