

**SINUMERIK 810N
SINUMERIK 820N
Grundaufführung 2 und 3**

Teil 2: Programmieren

Benutzeranleitung

Anwender-Dokumentation

Vorbemerkung

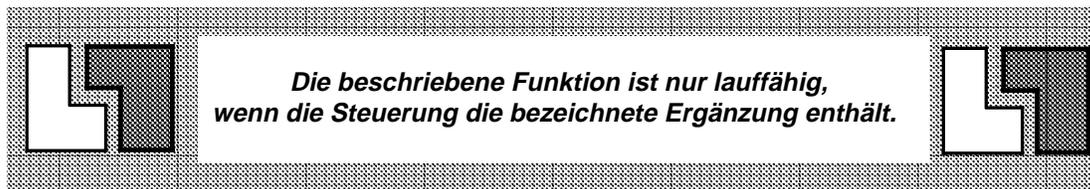
Die vorliegende Programmieranleitung beschreibt die für die Steuerung SINUMERIK 810N/820N gültigen Programmiermöglichkeiten und Standard-Funktionen.

Ein Überblick über alle, auch die mit SINUMERIK 810M/820M identischen Funktionen der SINUMERIK 810N/820N sind dem Kapitel 13 "Programmschlüssel" zu entnehmen. Die Hardware der Steuerung SINUMERIK 810N/820N ist mit jener von SINUMERIK 810M/820M identisch.

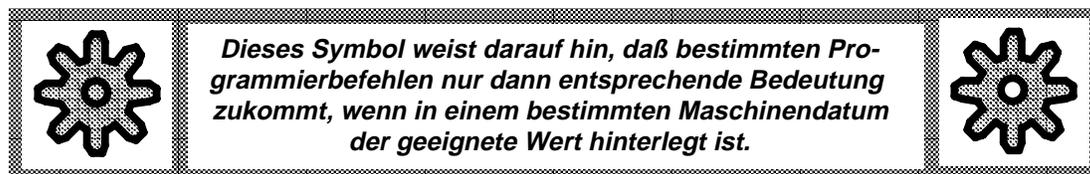
Ausnahme: Statt der Taste  ist die Taste  vorgesehen.

SINUMERIK 810N/820N wird für die Steuerung von Stanz-/Nibbel- und Laserbearbeitungsmaschinen eingesetzt. Der Standard-Anwender-Speichermodul (Standard-ASM) enthält demgemäß Stanz-, Nibbel- und Laserbearbeitungs-Zyklen.

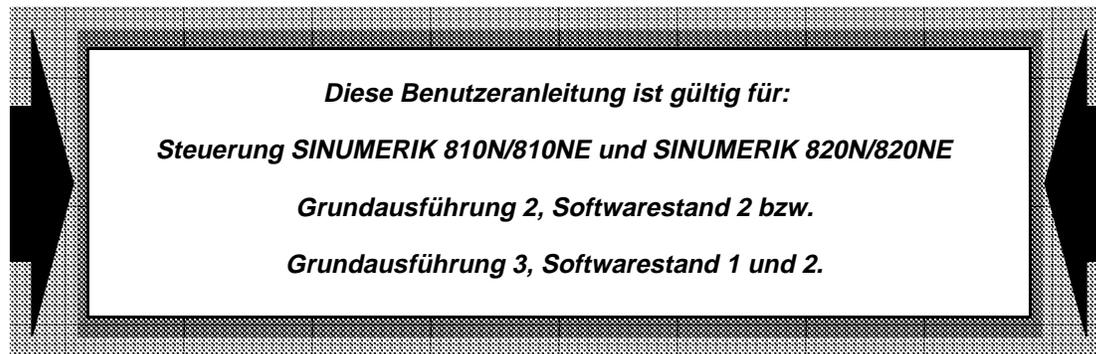
Die vorliegende Beschreibung ist ausgelegt für den maximalen Funktionsumfang der SINUMERIK 810N/820N. Allenfalls notwendige Software-Optionen sind angeführt und durch folgendes Symbol gekennzeichnet:



Einen Überblick über alle zur Verfügung stehenden Optionen bietet der Katalog NC 33. Weitere Funktionen, die in dieser Programmieranleitung nicht beschrieben sind, können in der Steuerung lauffähig sein. Es besteht jedoch kein Anspruch auf diese Funktionen bei Neulieferung oder im Service-Fall.



Die Programmierbeispiele sind im ISO-Code geschrieben. Alle Geometriewerte sind metrisch. Änderungen aufgrund technischer Weiterentwicklung behalten wir uns vor.



Grundlagen der Programmierung	1
Bewegungsrichtungen, Maßangaben	2
Programmieren von Bewegungssätzen	3
Schalt-, Hilfs-, Zusatz-, Sonderfunktionen 810N/820N	4
Unterprogramme	5
Parameter	6
Konturzug	7
Werkzeugkorrekturen	8
Fräserradiuskorrektur (FRK)	9
Zyklen	10
Programmieren von Zyklen	11
Anhang	12
Programmschlüssel SINUMERIK 810N/820N	13

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	1-5
1.5 Zeichenvorrat	1-6
1.6 Lochstreifen	1-7
1.6.1 Lochstreifenleser	1-7
1.6.2 Lochstreifen-Code	1-7
1.6.3 Vorspann	1-7
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 810N/820N	1-16
1.10 Umdrehungsvorschub Grenzdaten	1-19
1.11 Kanalstruktur	1-20
2 Bewegungsrichtungen, Maßangaben	2-1
2.1 Koordinatensystem	2-1
2.2 Weginformation, Wegbedingungen	2-2
2.3 Maßsysteme: Bezugsmaß, Kettenmaß G90/G91	2-2
2.4 Bezugspunkte	2-4
2.5 Nullpunktverschiebung	2-5
2.6 Wegberechnung	2-9
2.7 Werkstückvermessung, Eingabesystem G70/G71	2-10
2.8 Spiegeln	2-11
2.9 Programmierbare Arbeitsfeldbegrenzung G25/G26	2-15
2.10 Softwaresnocken (nur GA3 SW2)	2-16
2.11 Koordinatendrehung (KD)	2-19
2.12 Maßstabsänderung: Anwahl G51, Abwahl G50	2-21
3 Programmieren von Bewegungssätzen	3-1
3.1 Achsbefehle	3-1
3.1.1 Achsbewegung ohne Bearbeitung G00	3-3
3.1.2 Achsverdopplung (nur GA3 SW2)	3-4
3.1.2.1 Funktionsweise	3-4
3.2 Achsbewegungen mit Bearbeitung	3-7
3.2.1 Geradeninterpolation G01	3-7
3.2.2 Kreisinterpolation G02/G03	3-8
3.2.2.1 Interpolationsparameter I, J, K	3-9
3.2.2.2 Radiusprogrammierung	3-12

3.2.3	Zylinderinterpolation (nur GA3 SW2)	3-13
3.2.4	Polarkoordinaten	3-15
3.2.4.1	Polarkoordinaten G10/G11/G12/G13	3-15
3.2.4.2	Polarkoordinaten G110/G111	3-17
3.2.5	Vorschub F, G94/G95	3-18
3.2.6	Genauhalt G09/G60/G00, Bahnsteuerbetrieb G62/G64	3-18
3.2.6.1	Genauhaltgrenze fein und grob G09/G60/G00	3-18
3.2.6.2	Bahnsteuerbetrieb G62/G64	3-20
3.2.7	Verweilzeit G04	3-21
3.2.8	Weiches Anfahren und Verlassen der Kontur	3-22
3.2.9	SPLINE-Interpolation G06	3-25
3.3	Referenzpunktfahren über Teileprogramm: G74 (nur GA3 SW2)	3-26
3.3.1	Veranlassung	3-26
3.3.2	Funktionsbeschreibung	3-26
3.3.3	Starten der Funktion	3-26
4	Schaltfunktionen, Hilfsfunktionen, Zusatzfunktionen, Sonderfunktionen 810N/820N	4-1
4.1	Allgemeines	4-1
4.2	Übersicht spezieller Funktionen für SINUMERIK 810N/820N	4-3
4.3	Schnelle M-Funktionen (Ein-/Ausschaltfunktionen)	4-4
4.4	Sonderfunktionen	4-5
4.4.1	Automatische Streckenaufteilung E-/H-/H1-Funktionen	4-5
4.4.1.1	Programmierung mit dem E-Wert, Vorschub in mm/Hub bzw. inch/Hub	4-5
4.4.1.2	Programmierung mit dem H-Wert, selbsttätige Streckenaufteilung	4-9
4.4.1.3	Programmierung mit dem H1-Wert, Vervielfältigung einer programmierten Teilstrecke	4-10
4.4.2	Nibbel- und Stanzfunktionen (Spezielle M-Funktionen)	4-11
4.4.2.1	M 20 positionieren	4-11
4.4.2.2	M 21 Tangentialsteuerung AUS	4-11
4.4.2.3	M 22/M 24 Nibbeln	4-11
4.4.2.4	M 25 Stanzen	4-15
4.4.2.5	Zuordnung von E- und H-Programmierung zu den M-Funktionen	4-16
4.4.2.6	Programmierbeispiele	4-17
4.4.2.7	M 29 Pratzenpositionen aktualisieren	4-20
4.4.3	Programmierung der Pulsauswahl: schnelle T-Nr.-Ausgabe (nur GA3 SW2)	4-21
4.4.4	Programmierung der Analogausgänge (nur GA3 SW2)	4-21
4.5	Tangentialachse (Drehbare Werkzeugachse)	4-23
4.5.1	Mitschleppachse	4-24
4.5.2	Referenzpunkt, Bezugspunkt, Einbaulage	4-24
4.5.3	Drehrichtung	4-25
4.5.4	Programmierung	4-26
4.6	Tangentialsteuerung	4-26
4.6.1	Definition	4-26
4.6.1.1	Geradeninterpolation	4-27
4.6.1.2	Kreisinterpolation G02/G03	4-28
4.6.2	Aktivieren der Tangentialsteuerung	4-31
4.6.2.1	Selbsthaltende Tangentialsteuerung	4-31
4.6.3	Programmierbeispiele für Tangentialsteuerung in Verbindung mit Stanzen/Nibbeln	4-32
4.6.3.1	Programmierbeispiel: Geradeninterpolation (G00/G01)	4-32
4.6.3.2	Programmierbeispiel: Kreisinterpolation (G02/G03)	4-33
4.6.3.3	Beispiel	4-34

4.7	Analoge Tangentialwinkelausgabe	4-35
4.7.1	Grundlagen der Winkelberechnung	4-35
4.7.1.1	Geradeninterpolation G01/G00	4-35
4.7.1.2	Kreisinterpolation G02/G03	4-36
4.7.2	Programmierung	4-37
4.7.2.1	Beispiel	4-37
4.8	Mini Pattern	4-38
4.9	Laserleistungssteuerung	4-39
4.9.1	Vorbereitende Festlegung der Grenzdaten	4-39
4.9.2	Auswahl und Ausführung der Laserfunktion	4-40
4.9.3	Schnelle M-Funktionen für die Laserleistungssteuerung	4-41
4.9.4	Beschreibung der Laserfunktionen	4-42
4.9.4.1	Ausgabe von Konstantspannungen	4-42
4.9.4.2	Ausgabe der Laserspannung als Funktion der Bahngeschwindigkeit ...	4-43
4.9.4.3	Ausgabe der Laserspannung als Funktion der Strecke	4-45
4.9.4.4	Ausgabe der Laserspannung als Funktion der Zeit	4-46
4.10	Nibbelspezifische Alarmmeldungen	4-48
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	Programmbeispiele mit R-Parametern (Nibbelmaschine)	6-5
6.6	Belegung der R-Parametern	6-6
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	Programmierbeispiel Laserschneiden	7-6
8	Werkzeugkorrekturen	8-1
8.1	Werkzeugdaten	8-1
8.2	Aufbau des Werkzeugkorrekturspeichers	8-2
8.3	Schnittpunkt-Fräseradius-Bahnkorrektur G40/G41/G42	8-3
8.4	Werkzeug-Längenkorrektur, positiv oder negativ	8-7

9	Fräserradiuskorrektur (FRK)	9-1
9.1	Anwahl der FRK	9-1
9.2	FRK im Programm	9-3
9.3	Abwahl der FRK (G40)	9-6
9.4	Wechsel der Korrekturrichtung (G41, G42)	9-8
9.5	Wechsel der Korrekturnummer (D..)	9-8
9.6	Wechsel der Korrekturwerte (R1, R2)	9-9
9.7	Wiederholung der schon angewählten G-Funktion (G41, G42) mit der gleichen Korrekturnummer	9-9
9.8	M00, M01, M02 und M30 bei angewählter FRK	9-10
9.9	FRK bei Kombination von verschiedenen Satzarten und Auftreten von Konturfehlern	9-12
9.10	Sonderfälle der FRK	9-16
9.11	Wirkung bei negativen Korrekturwerten	9-20
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	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	11-22
11.9	NC-spezifische Funktionen	11-28
11.10	E/A-Funktionen (nur GA3 SW2)	11-37
11.11	@-Code-Tabelle	11-41
11.12	Pattern (ab SINUMERIK 810N/820N Grundausführung 3)	11-51
11.12.1	Aufgabenstellung	11-51
11.12.1.1	Einzelteil	11-52
11.12.2	Pattern	11-53
11.12.2.1	Definition	11-53
11.12.2.2	Werkzeugorientierte Bearbeitung	11-54
11.12.2.3	Teileorientierte Bearbeitung	11-54
11.12.3	Patternprogrammierung mit SINUMERIK 810N/820N	11-54
11.12.3.1	Numerierung der Patternteile	11-55
11.12.3.2	Gleichsinnige Patternbearbeitung	11-57
11.12.3.3	Mäanderförmige Patternbearbeitung	11-60
11.12.4	Erläuterung der @-Befehle für Pattern	11-62
11.12.4.1	@-Befehl für Initialisierung	11-63
11.12.4.2	@-Befehle für gleichsinnige Abarbeitung	11-64
11.12.4.3	@-Befehle für mäanderförmige Abarbeitung	11-64
11.12.4.4	Programmiervorschrift	11-65
11.12.5	Fertigungslösung	11-67
11.12.5.1	Wegoptimierung bei gleichsinniger Bearbeitung	11-69
11.12.5.2	Wegoptimierung bei mäanderförmiger Bearbeitung	11-71
11.12.5.3	Trennen	11-73

11.12.6	Anzeige Pattern-Teil im AUTOMATIK-Bild	11-77
11.12.7	Satzvorlauf	11-78
11.12.7.1	Satzvorlauf im AUTOMATIK-Betrieb	11-78
11.12.7.2	Beispiele zum Satzvorlauf	11-79
11.12.8	Grafische Simulation	11-81
11.12.8.1	Grafische Simulation bei Patternbearbeitung	11-81
11.12.8.2	Satzvorlauf bei grafischer Simulation	11-82
11.12.8.3	Nullpunktverschiebungen G54 ... G57 bei Simulation	11-83
12	Anhang	12-1
12.1	Exzenterbetrieb	12-1
12.1.1	Umschalten der Exzentergeschwindigkeit	12-1
12.1.1.1	Umschalten der Exzentergeschwindigkeit mit Anwahl des Nibbelmodus ..	12-2
12.1.1.2	Umschalten der Exzentergeschwindigkeit durch spezielle M-Funktion	12-2
12.1.2	E-Wertüberwachung	12-3
12.1.2.1	Maximalwerte für den E-Wert und den Drehwinkel	12-3
12.1.2.2	Überwachungsarten der maximalen E- und Drehwinkel-Werte bei M 22	12-4
12.1.3	Automatische Stanz-Nibbelumschaltung	12-4
12.1.3.1	Programmiervorschrift	12-5
12.1.3.2	Umschaltkriterium Stanzen Nibbeln beim Arbeiten mit M 25	12-5
12.1.3.3	Beispiele für die automatischen Stanz-Nibbelschaltung	12-7
12.2	Pratzenschutz	12-8
12.2.1	Eingabewerte	12-8
12.2.1.1	Pratzen auf Bedienerseite	12-9
12.2.1.2	Werkzeugspezifische Pratzenschutzbereiche	12-10
12.2.2	Pratzenpositionen aktualisieren M 29	12-11
12.2.3	Pratzen umfahren	12-12
12.2.3.1	Beispiele	12-13
12.2.4	Signal an PLC für Stanze hochfahren M 46	12-15
12.2.5	Stanzen im Pratzenschutzbereich erlaubt	12-16
12.2.6	Pratzenschutz für Laserbearbeitung	12-17
12.3	Simulation	12-17
12.3.1	Simulation mit Werkzeugdarstellung	12-17
12.3.2	Rote Doppellinie zeichnen mit M 26	12-18
12.3.3	Variabler Nullpunkt bei Simulation	12-18
13	Programmschlüssel SINUMERIK 810N/820N	13-1
13.1	Interne G-Gruppeneinteilung bei @36b	13-1
13.2	Programmschlüssel	13-2

1 Grundlagen der Programmierung

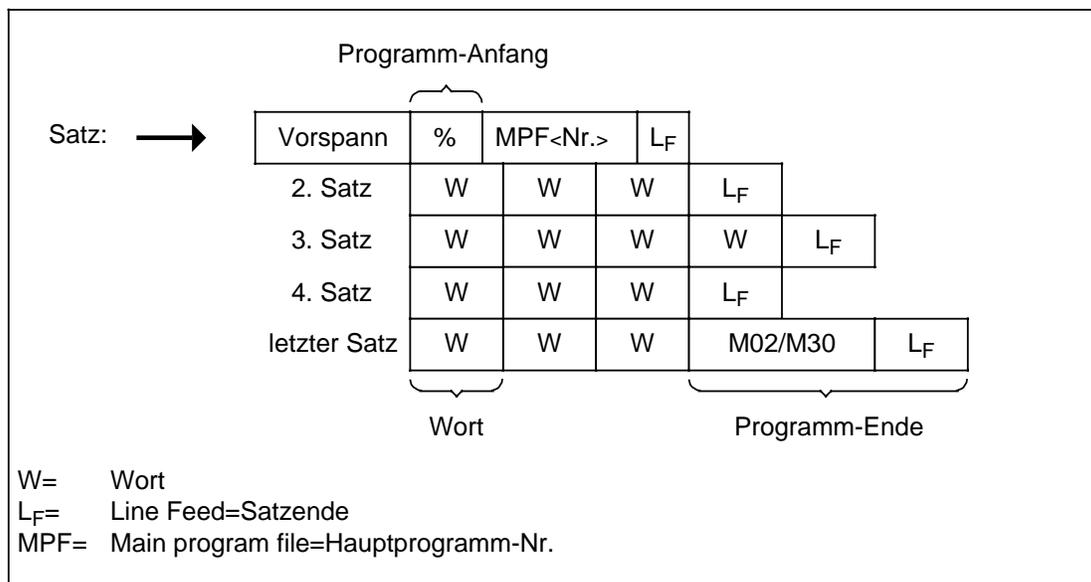
1.1 Programmaufbau

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

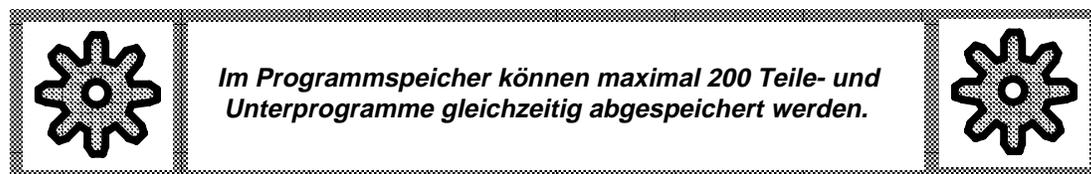
- dem Zeichen für den Programmanfang
- einer Anzahl von Sätzen und
- dem Zeichen für das Programmende.

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:

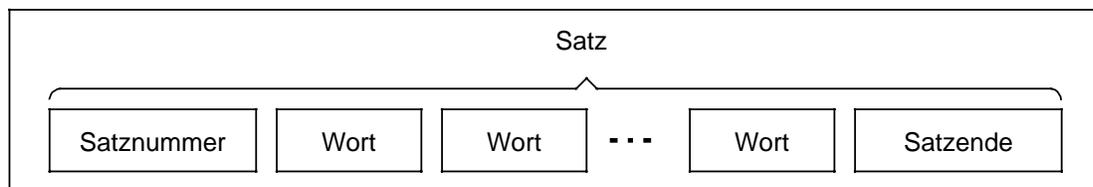
- 0 ... 9999 Bearbeitungsprogramme,
- 1 ... 9999 Unterprogramme bei GA3 und 1 ... 999 Unterprogramme bei GA2.

Erfolgt die Programmeingabe mit Bedienerunterstützung über die Bedientafel, so werden bei Betätigen des Softkey SATZNUMMER die Satznummern in Fünferschritten automatisch generiert, nachdem die erste Satznummer per Tastatur eingegeben wurde. 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:

N9235 G.. X.. Z.. F.. S.. T.. M.. H.. LF

N	Adresse der Satznummer
9235	Satznummer
G..	Wegbedingung
X.. Z..	Weginformation
F..	Vorschub
S..	Drehzahl
T..	Werkzeugnummer
M..	Hilfsfunktion
H..	Hilfsfunktion
LF	Satzende

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 Y-15 F200 S1000 M03 LF
```

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

Satzbeispiel:

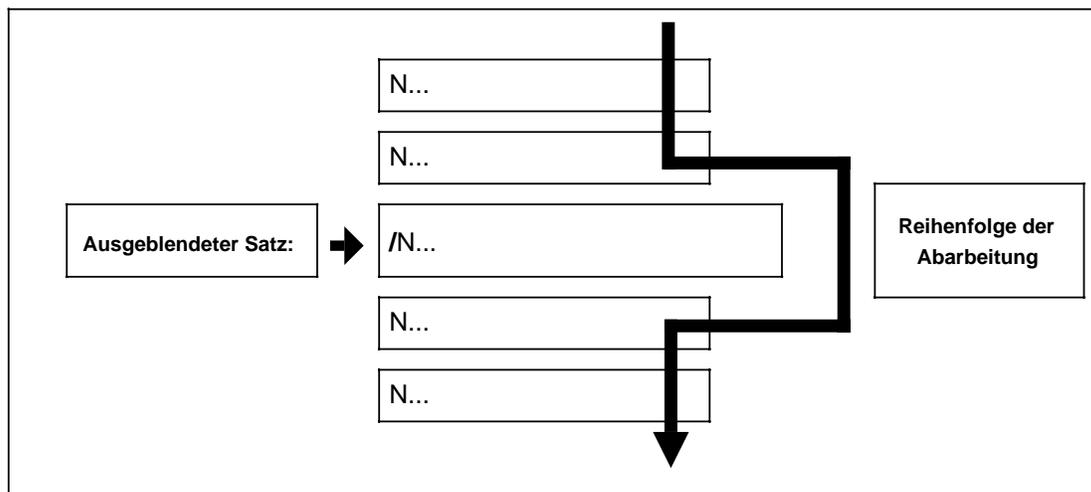
```
N15 Y20 LF
```

Ein Hauptsatz und mehrere Nebensätze bilden einen Programmabschnitt.

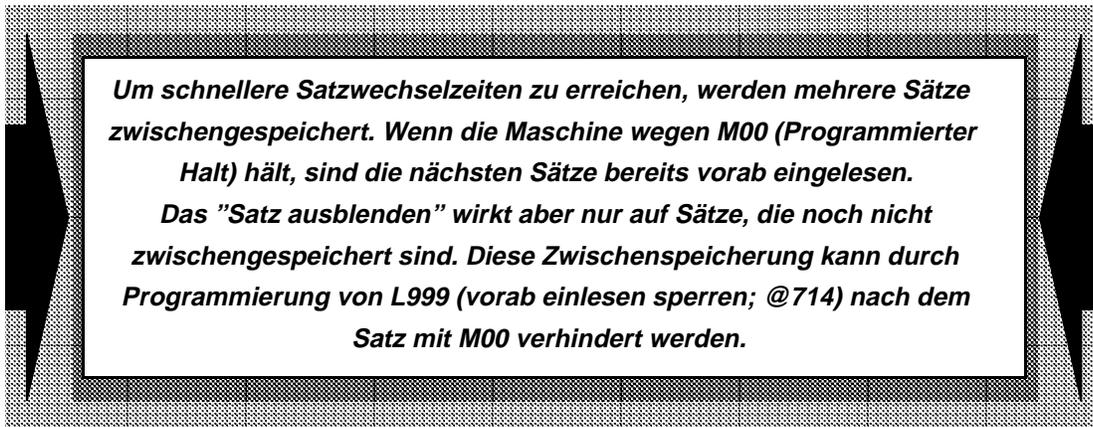
```
Beispiel:  :10
           N105
           N110
           N115 } Abschnitt
```

1.3.2 Ausblendbare Sätze

Sätze eines Programmes, die nicht bei jedem Programmablauf ausgeführt werden sollen, können durch das Zeichen Schrägstrich "/" vor dem Wort der Satznummer ausgeblendet werden. Das Satzausblenden wird über die Maschinensteuertafel oder durch die Anpaßsteuerung aktiviert. 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

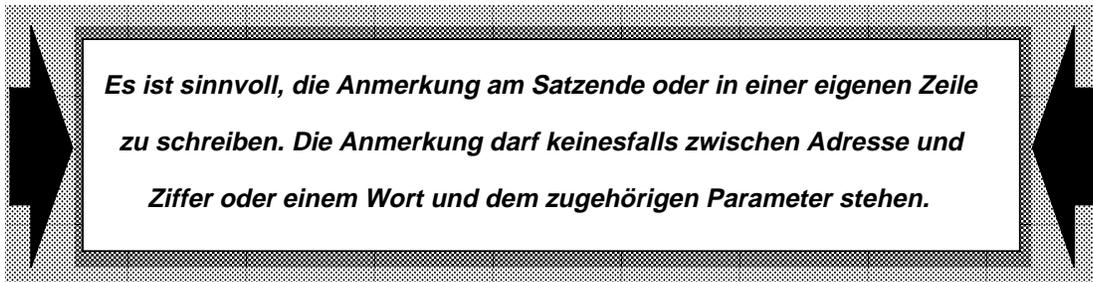


1.3.3 Anmerkungen (Kommentare)

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 %, LF, "(" und ")" stehen.

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



Richtig:

```
X100 Y200 ( Prätze ) Lf  
X100+R1 Y200 ( Prätze ) Lf
```

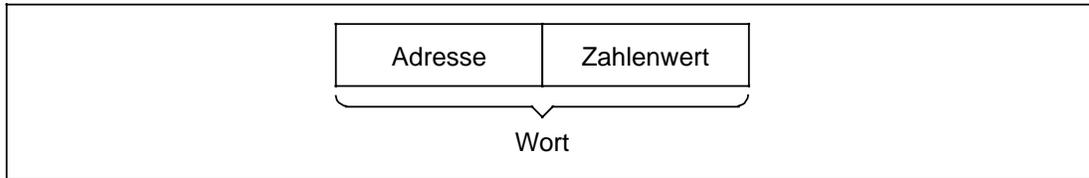
x	Adresse
100	Zahlenwert
R1	R-Parameter
(Anmerkungsbeginn, Leerzeichen
Prätze	Anmerkung, Leerzeichen
)	Anmerkungsende

Falsch:

```
X ( Prätze ) 100  
X100+ ( Prätze ) R1
```

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 Dezimalpunkt versehen sein. Das Vorzeichen steht zwischen Adreßbuchstaben und Ziffernfolge. Das positive Vorzeichen kann entfallen.



Wortschema: Aufbau eines Wortes

Der Wortaufbau ist an DIN 66025 angelehnt.

Kurzbeschreibung der Wörter:

%4 N04 G02/G03 D03 XL+053 YL+053 ZL+053 AL053 ID053 JD053 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.

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, 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

Bei SINUMERIK 810N/820N GA3 muß die 5. Achse mit erweiterter Adresse eingegeben werden, z. B. C1=5.

Hexadezimalziffern in Verbindung mit CL800 Maschinencode

a, b, c, d, e, f (siehe Projektierungsanleitung CL800)

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 (Setting-Daten)

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 (siehe Code-Tabelle).

Alle Zeichen eines Codes haben ein gemeinsames Kennzeichen:

- ISO immer gerade Lochanzahl und
- 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 an der Bedientafel 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.

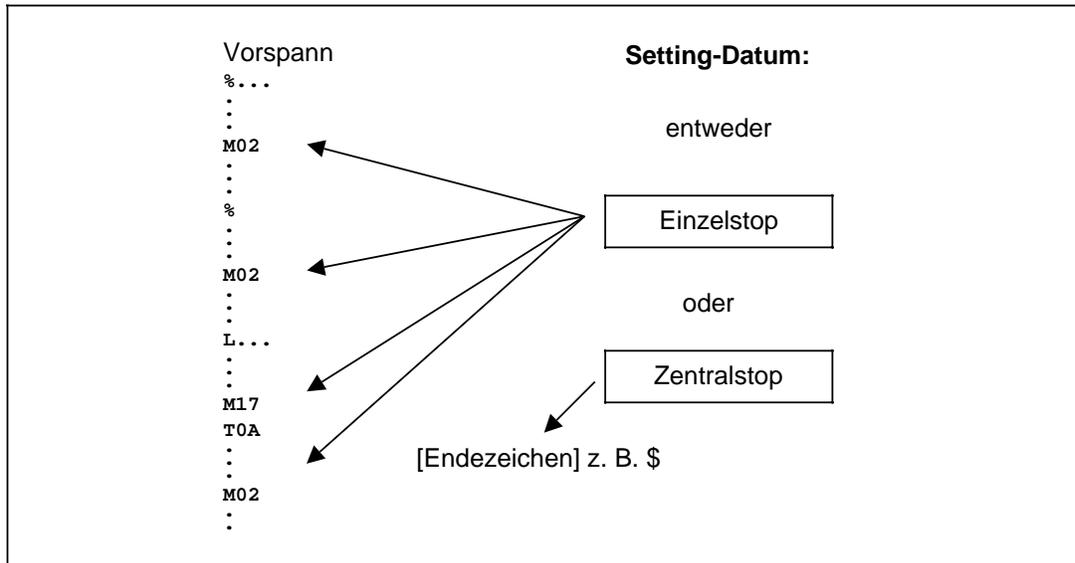
TASCHE	%	MPF	1579	L_F
--------	---	-----	------	-------

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	Vorspann
%MPF1235 L_F	Teileprogramm 1235 (MAIN PROGRAM FILE)
(Messung durchführen)	Anmerkung
N... L _F	Teileprogramm
N... L _F	
M02 L _F	Teileprogramm-Ende

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

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

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

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

%RPA0 L _F	(R-PARAMETER AKTIV) Kanal-Nr. (0=zentrale R-Parameter)
R...=... L _F R...=... L _F	Parameter-Nummern mit Wertzuweisungen
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 entspr. MD)
M02 L _F	Werkzeugkorrekturen Datenblock-Ende

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

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

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

Speicherbereiche

Die Speicherbereiche der Steuerung SINUMERIK 810N/820N werden über folgende **Kennungen** angesprochen:

Kennung	Bedeutung
MPF	Teilprogramm (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 - P arameter 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 LF	Kennung Programm löschen (CLEAR FILE) Über Setting-Daten (SD für serielle Schnittstelle) kann festgelegt werden, ob mit % CLF eine automatische Reorganisation des Teileprogrammspeichers verhindert werden soll.
MPF1234 LF	Teileprogramm % 1234 löschen
MPF 1, 1200 LF	Teileprogramm % 1 bis % 1200 löschen
MPF 0, 9999 LF	Löschen aller Teileprogramme
SPF 10 LF	Unterprogramm L10 löschen
SPF 11, 79 LF	Unterprogramm L11 bis L79 löschen
SPF 1, 999 LF	Löschen aller Unterprogramme bei GA2
SPF 1, 9999 LF	Löschen aller Unterprogramme bei GA3
M30, M02 oder M17 LF	Endekennung M30, M02 oder M17

- **Textdaten löschen**

%PCA LF
M02 oder M30 LF

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	x
F		
G		
H		.			.	.				
I	
J		
K		
L		
M		
N		
O	
P		.		.		.				x
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.

1.9 Eingabe-/Ausgabeformate 810N/820N

Eingabe-/Ausgabeformate

Eingabefeinheit:
 0,01 mm (Maschinendatum)
 0,001 inch
 0,001 Grad

bzw. Lageregelfeinheit
 0,005 mm
 0,0005 inch
 0,0005 Grad



Maschinendatum

Bedeutung		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 999999.99		0.001 bis 99999.999		–	
Nullpunktverschiebung		±0.01 bis 999999.99		±0.001 bis 99999.999		±0.001 bis 99999.999	
Linear-Vorschub (F) 		0.1 bis 446000.0	inch/ min	0.01 bis 17700.00	inch/ min	1 bis 45000	Grad/ min
Werkzeug- Korrektur	Länge	±0.01 bis 99999.99	min	±0.001 bis 999.999	inch		
	Radius	±0.01 bis 9999.99		±0.01 bis 999.999			
Verweilzeit	X	0.01 bis 99999.999	sec	0.01 bis 99999.999	sec		
	F	0.01 bis 99999.999		0.01 bis 99999.999			
	S	0.1 bis 99.9	Umdrehungen	0.1 bis 99.9	Umdrehungen		
Winkel bei Polarkoordinaten		–		–		0 bis 359.99999	Grad
R-Parameter		Dimension je nach Zugehörigkeit (intern Gleitkomma) alle Kombinationen					
Vorschub in mm/ Hub		E	0,01 bis 83000,00	mm/ Hub	0,001 bis 8300,000	inch/ Hub	
Spannungswert bei Laserleistungssteuerung		E	0,001 bis 10,000	V	0,001 bis 10,000	V	

Eingabe-/ Ausgabeformate

Eingabefinheit:
0,001 mm (Maschinendatum)
0,0001 inch
0,001 Grad

bzw. Lageregelfinheit
0,0005 mm
0,00005 inch
0,0005 Grad



Maschinendatum

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 999999.99		0.001 bis 99999.999		–	
Nullpunktverschiebung		±0.01 bis 999999.99		±0.001 bis 99999.999		±0.001 bis 99999.999	
Linear-Vorschub (F) 		0.1 bis 446000.0	inch/ min	0.01 bis 17700.00	inch/ min	1 bis 45000	Grad/ min
Werkzeug- Korrektur	Länge	±0.01 bis 99999.99	min	±0.001 bis 999.999	inch		
	Radius	±0.01 bis 9999.99		±0.01 bis 999.999			
Verweilzeit	X	0.01 bis 99999.999	sec	0.01 bis 99999.999	sec		
	F	0.01 bis 99999.999		0.01 bis 99999.999			
	S	0.1 bis 99.9	Umdrehungen	0.1 bis 99.9	Umdrehungen		
Winkel bei Polarkoordinaten		–		–			
R-Parameter		Dimension je nach Zugehörigkeit (intern Gleitkomma) alle Kombinationen					
Vorschub in mm/ Hub E		0,001 bis 8300,000	mm/ Hub	0,0001 bis 830,0000	inch/ Hub		
Spannungswert bei Laserleistungssteuerung E		0,001 bis 10,000	V	0,001 bis 10,000	V		

Eingabe-/ Ausgabeformate

Eingabefinheit:
 0,0001 mm (Maschinendatum)
 0,00001 inch
 0,001 Grad

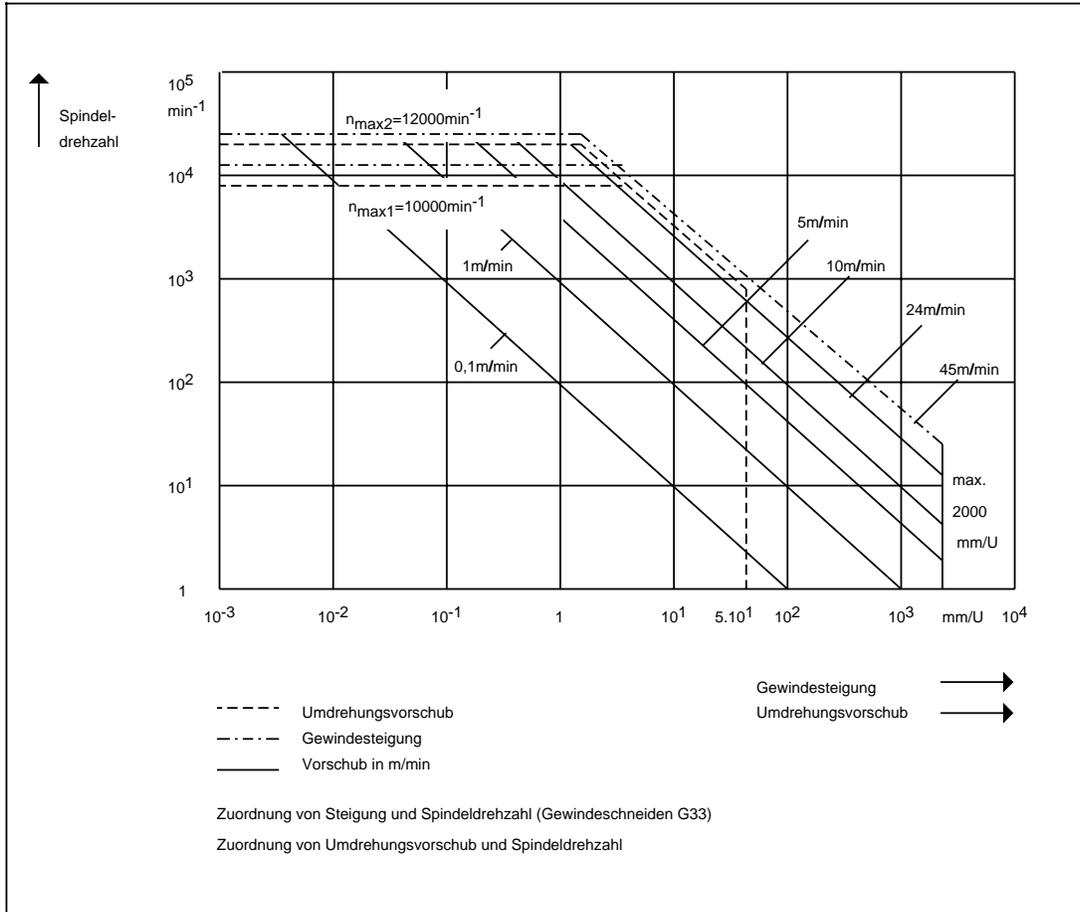
bzw. Lageregelfinheit
 0,00005 mm
 0,000005 inch
 0,0005 Grad



Maschinendatum

Bedeutung Adressen		metrisch		Zoll		Grad	
		Bereich	Einheit	Bereich	Einheit	Bereich	Einheit
Weginformationen (Linearachsen) Interpolationsparameter		±0.0001 bis 9999.9999	mm	±0.00001 bis 999.9999	inch	-	Grad
Weginformationen bei G91 (Rundachsen)		-		-		0.001 bis 99999.999	
Weginformationen bei G90 (Rundachsen)		-		-		±0.001 bis 359.999	
Fase (U-); Radius (U)		0.0001 bis 9999.9999		0.00001 bis 999.99999		-	
Nullpunktverschiebung		±0.0001 bis 9999.9999		±0.00001 bis 999.99999		±0.001 bis 99999.999	
Linear-Vorschub (F) 		0.001 bis 4460.000	mm/ min	0.0001 bis 177.0000	inch/ min	1 bis 45000	Grad/ min
Werkzeug- Korrektur	Länge	±0.0001 bis 999.9999	min	±0.00001 bis 99.99999	inch		
	Radius	±0.0001 bis 99.9999		±0.00001 bis 9.99999			
Verweilzeit	X	0.01 bis 99999.999	sec	0.01 bis 99999.999	sec		
	F	0.01 bis 99999.999		0.01 bis 99999.999			
	S	0.1 bis 99.9	Umdrehungen	0.1 bis 99.9	Umdrehungen		
Winkel bei Polarkoordinaten		-		-			
R-Parameter		Dimension je nach Zugehörigkeit (intern Gleitkomma) alle Kombinationen					
Vorschub in mm/ Hub E		0,0001 bis 830,0000	mm/ Hub	0,00001 bis 83,00000	inch/ Hub		
Spannungswert bei Laserleistungssteuerung E		0,0001 bis 10,000	v	0,001 bis 10,000	v		

1.10 Umdrehungsvorschub Grenzdaten



$n_{\text{max}1}$ erreichbar mit Drehgeber **1024** Pulse/Umdrehung
 $n_{\text{max}2}$ erreichbar mit Drehgeber **512** Pulse/Umdrehung

1.11 Kanalstruktur

Die Steuerung SINUMERIK 810N/820N ist mit 3 Kanälen, zwei Hauptkanälen und einem Hilfskanal, ausgestattet.

Jeder NC-Kanal wirkt wie eine eigene NC mit eigenem Programm, Decodierung, Satzaufbereitung und Interpolation. Der Hauptkanal ist z. B. für die Achsen X und Y sowie für das drehbare Werkzeug (C-Achse) vorgesehen. Der Hilfskanal kann dann z. B. den Werkzeugteller handhaben. (Für dieses Beispiel sind die Ergänzungen "Tangentialsteuerung" und "4. Achse" Voraussetzung.)

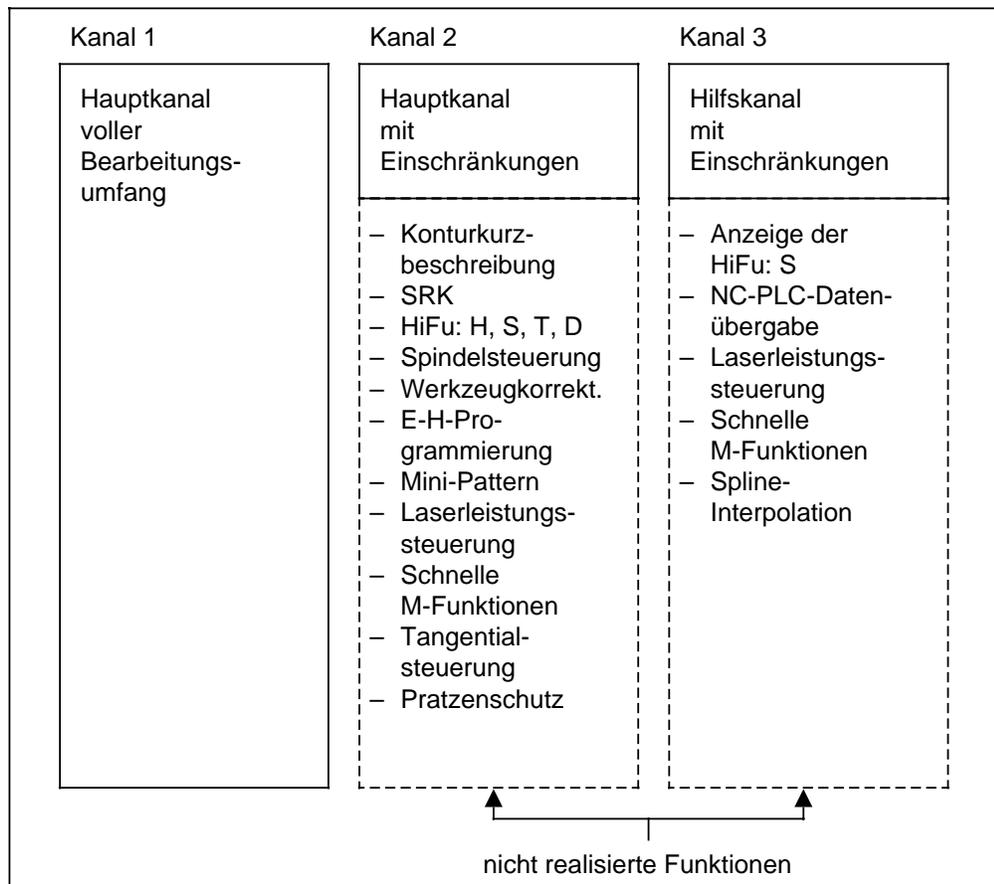
Hauptkanäle und Hilfskanal lassen sich simultan und unabhängig voneinander betreiben. Auf diese Weise können Zusatzbewegungen, z. B. von Lader und Werkzeugwechsler, unabhängig von den Hauptbewegungen ausgeführt werden. Außerdem ist z. B. die Durchführung von Rechenoperationen im Hilfskanal möglich. Synchronisiert werden die Kanäle über die PLC.

Funktionen der Programmbeeinflussung wie "Satzausblenden", "Einzelsatz", "Decodierung Einzelsatz" und "Probelaufvorschub" sind kanalspezifisch wirksam.

Im dritten Kanal ist nur die Simulation möglich.

Werkzeugkorrekturen, Konturzüge und nibbel-/laserspezifische Funktionen sind im Hilfskanal nicht wirksam. Interpolation (linear und zirkular) ist im Hilfskanal ohne Einschränkung wirksam.

Das folgende Bild zeigt den Funktionsumfang der einzelnen 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 Automatikbetrieb) 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). Jedoch muß beachtet werden, daß bei abwechselndem Verfahren von ein und derselben Achse in beiden Kanälen (z.B. Kanal1 Kanal2 Kanal1) die Istposition dieser Achse nur mit der programmierten Position übereinstimmt, wenn vor jedem Aufruf der Achse im gewechselten Kanal mit der Sequenz NC-STOP und NC-START gearbeitet wird. Danach muß die Achse mit G90 positioniert werden. Bei Rundachsen mit Modulprogrammierung muß hierbei die Funktion "Korrektur bei Rundachse" aktiviert sein.

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.

2 Bewegungsrichtungen, Maßangaben

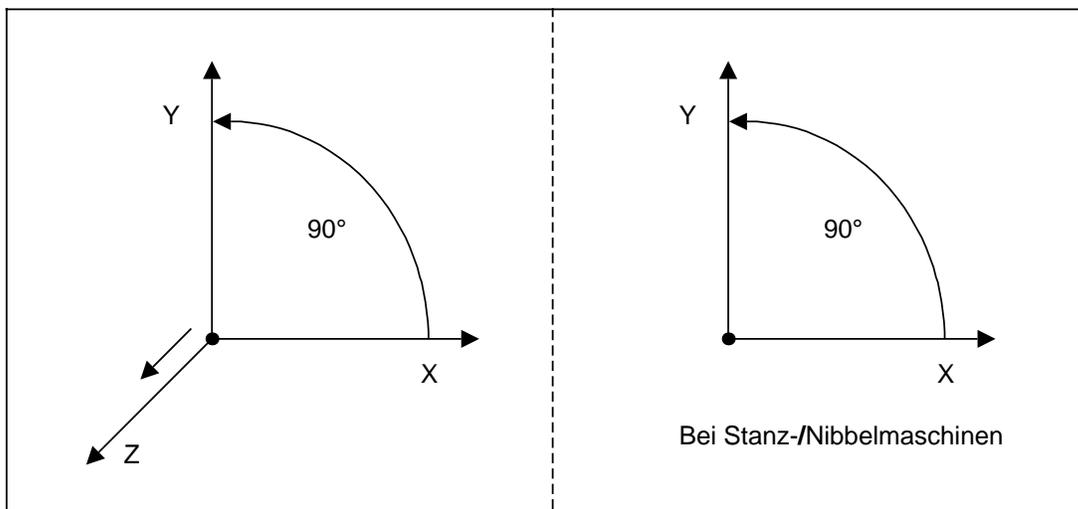
2.1 Koordinatensystem

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

Verwendet wird ein rechtsdrehendes, rechtwinkliges Koordinatensystem mit den Achsen X, Y 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.
- Dreht sich die erste Achse auf kürzestem Weg (90 Grad) in Richtung der zweiten Achse, so bewegt sich eine mit ihr verbundene Schraube mit Rechtsgewinde in Richtung der dritten Achse.

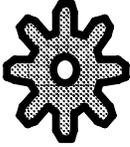
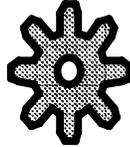


Rechtsdrehendes Koordinatensystem

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

Als Vorzugsbelegung für Stanzmaschinen gilt:

- Hauptachsen X und Y
- Drehbares Werkzeug C

	<p>Die Achsnamen sind über Maschinendaten frei projektierbar. Einschränkung: Der Achsname E ist unzulässig. Für Stanz-/Nibbelfunktionen muß die erste Achse mit X und die zweite Achse mit Y bezeichnet werden.</p>	
-------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------

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 Vermaßungsart.

Die **G-Funktionen** sind **in Gruppen eingeteilt** (siehe Programmschlüssel Kapitel 12). In einem Programmsatz darf aus jeder Gruppe jeweils nur eine Funktion stehen. Die G-Funktionen sind entweder **modal** (gespeichert) 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/G91

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

Bezugsmaßeingabe G90

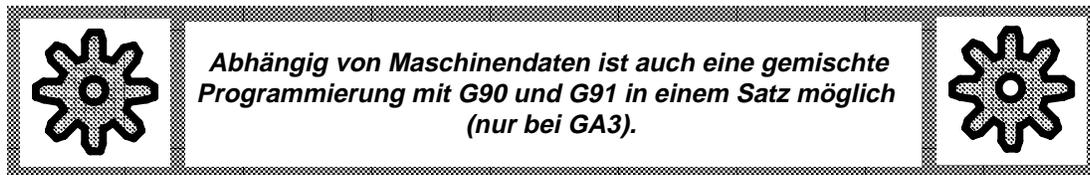
Bei der Bezugsmaßeingabe (Absolutmaßeingabe) beziehen sich alle Maßangaben auf einen festgelegten Nullpunkt, der meist der **Werkstücknullpunkt W** ist. Der Zahlenwert der zugehörigen Weginformation gibt die Zielposition im Koordinatensystem an.

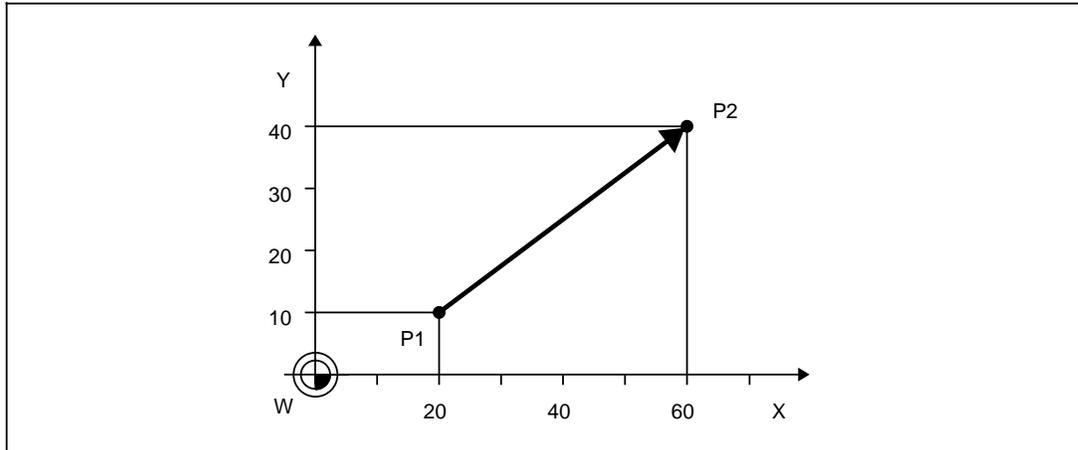
Kettenmaßeingabe G91

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

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.



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

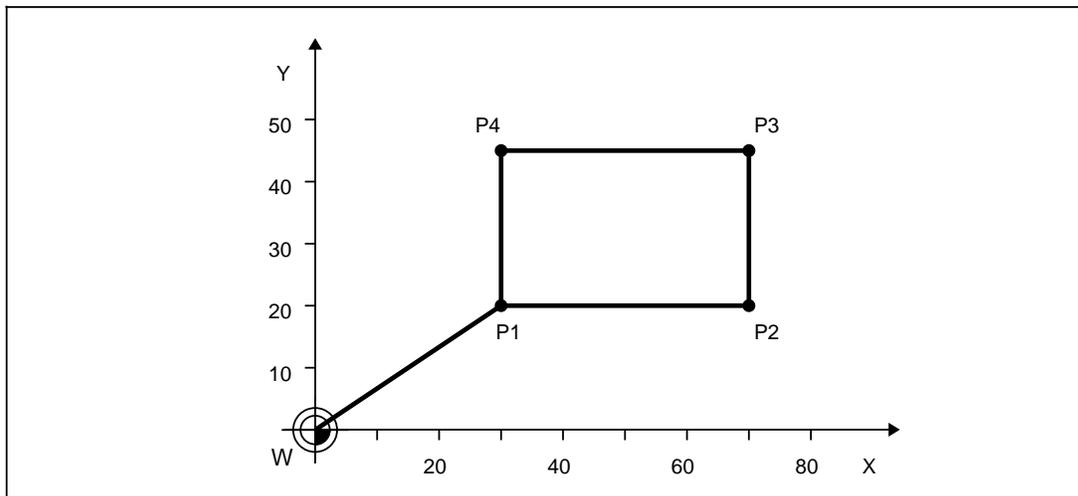
```
N.. G00 G90 X60 Y40 LF
```

Das Werkzeug fährt aus beliebiger Position nach P2.

Kettenmaßeingabe:

```
N.. G00 G91 X40 Y30 LF
```

Das Werkzeug fährt von P1 nach P2.

Beispiel: Programmierung im Bezugsmaß und im Kettenmaß

```
%10 LF
N05 G00 G90 G94 X30 Y20 LF (P1)
N10 G01 G91 X40 F100 LF (P2)
N15 Y25 LF (P3)
N20 X-40 LF (P4)
N25 Y-25 LF (P1)
N30 M30 LF
```

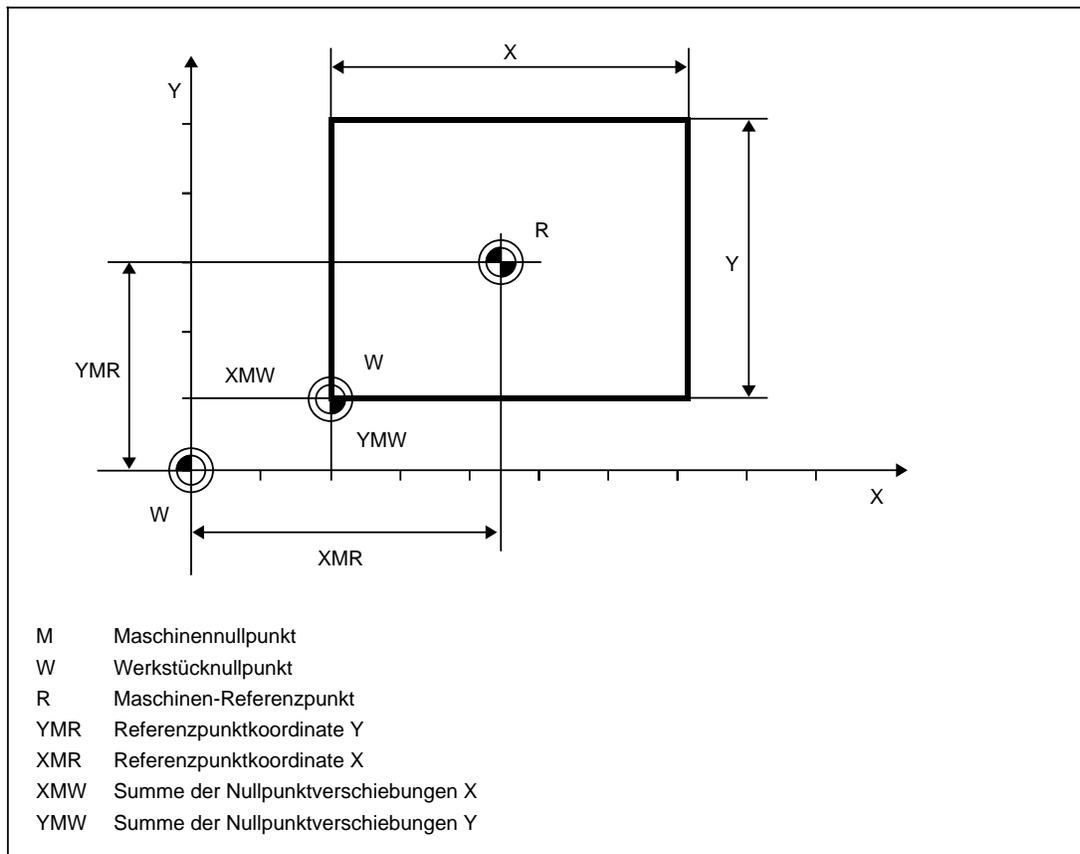
2.4 Bezugspunkte

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

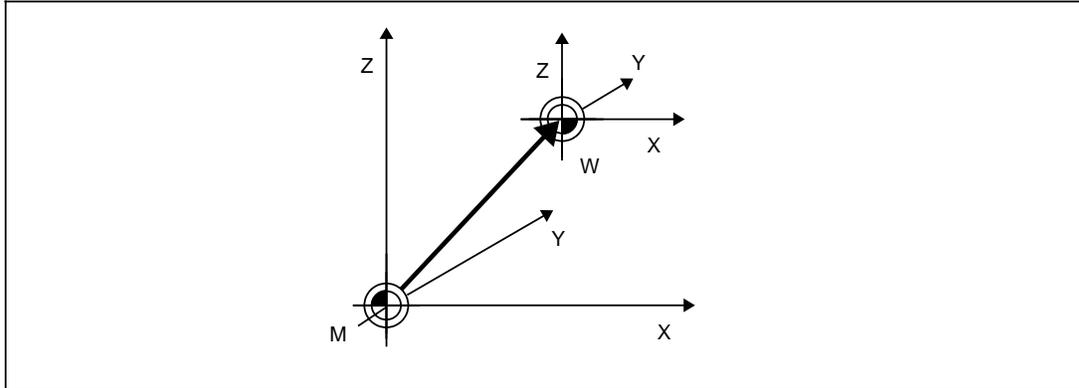
Der Referenzpunkt **R** ist ein vom Maschinenhersteller festgelegter Punkt, der nach dem Einschalten der Steuerung angefahren wird und das System synchronisiert. Das Referenzmaß ist im Maschinendatum festgelegt.



Beispiel: Stanz-/Nibbelmaschine

2.5 Nullpunktverschiebung

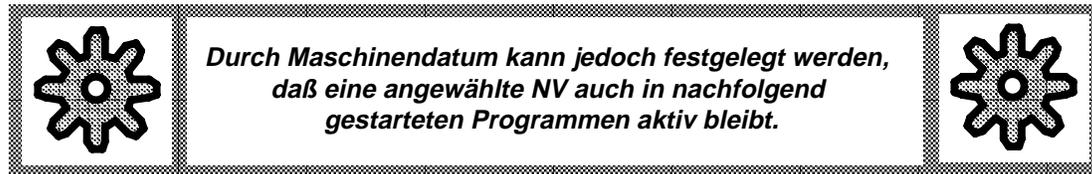
Die Nullpunktverschiebung ist der Abstand des Werkstücknullpunktes W (von dem die Vermessung ausgeht) zum Maschinennullpunkt M.

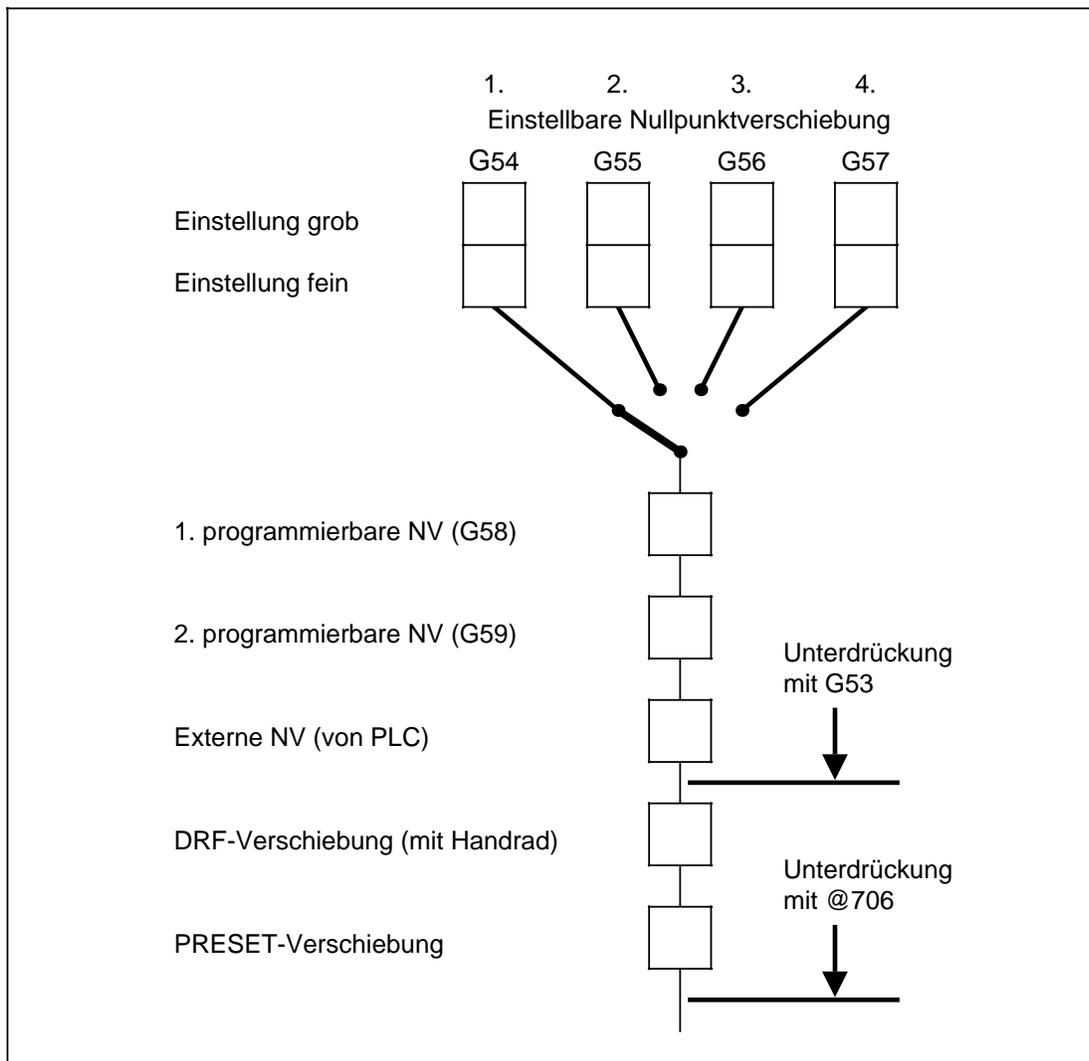


Nullpunktverschiebung in der Bearbeitungsebene

Folgende Nullpunktverschiebungen (NV) können aktiviert werden:

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





Summe der Nullpunktverschiebungen

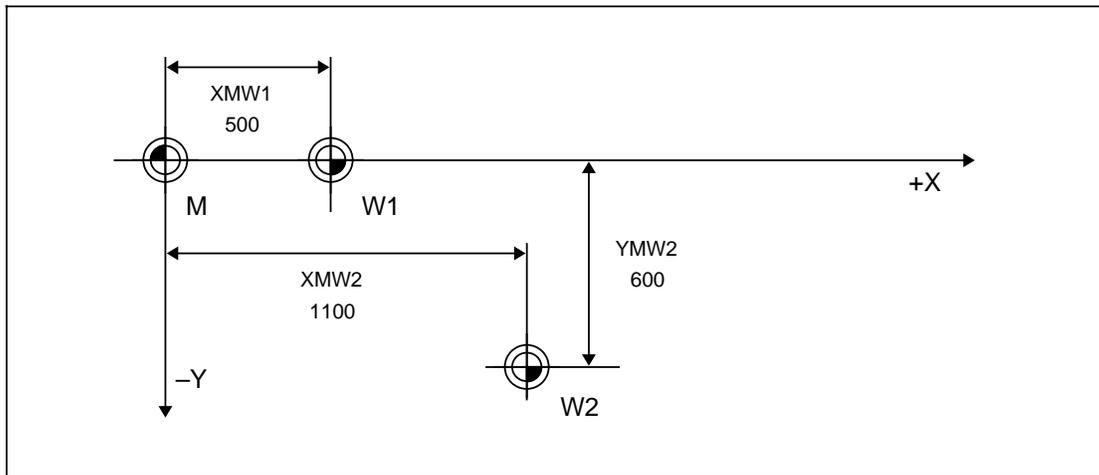
Summe der NV=(G54 bis G57)+externe NV (von PLC)+(G58, G59).

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 4 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:

```
%Z0A LF
G154 Y = 250 X = 280.1 LF
G155 Y = 220.34 X = 250.125 LF
.
.
G157 Y = 320 X = 350 LF
G254 Y = 0.1 X = 0.3 LF
.
.
G257 Y = 0.1 X = 0.5 LF
M02 LF
```

} einstellbare NV grob
 } einstellbare NV fein

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

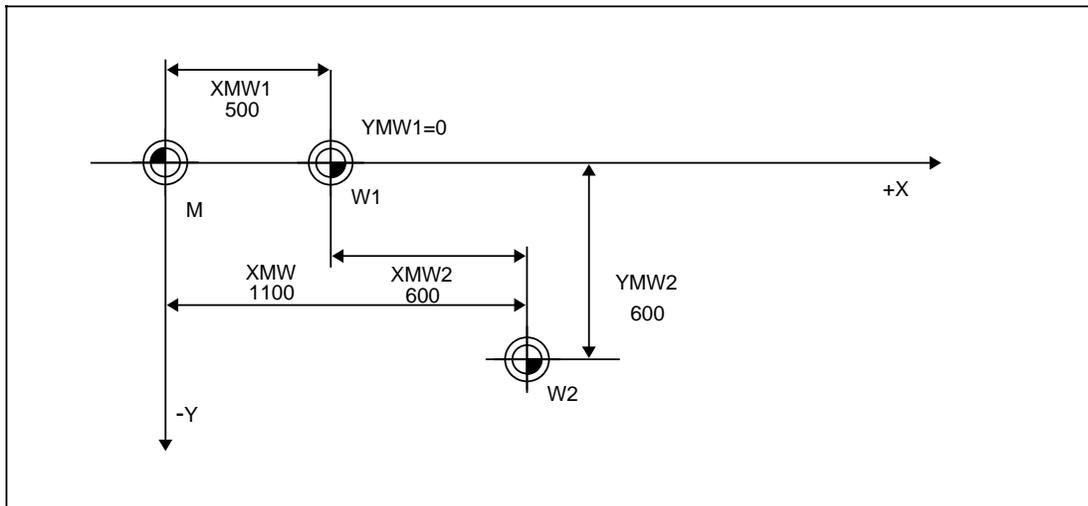
Programmierbare Nullpunktverschiebung G58/G59

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, YMW1, XMW1
- **Programmierbare** Nullpunktverschiebung
Eingabewerte, YMW2, XMW2
- Gesamte wirksame Nullpunktverschiebung
 $YMW = YMW1 + YMW2$
 $XMW = XMW1 + XMW2$

Programmierung:

```
N30
N35 G54 LF
N40 G59 Y600 X600 LF
N45
```

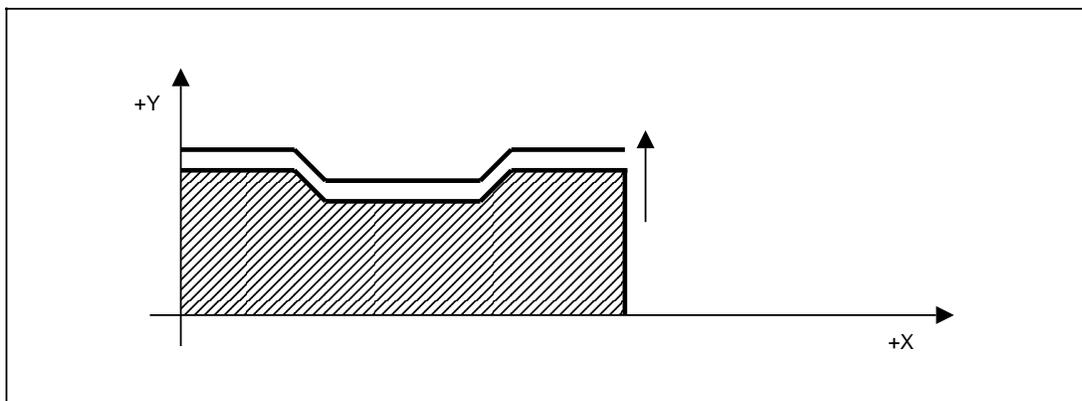


Einstellbare und programmierbare Nullpunktverschiebungen

In einem Satz mit G58 oder G59 dürfen außer den Nullpunktverschiebungen keine weiteren Funktionen geschrieben werden.

Anwendungsbeispiel mit G59:

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



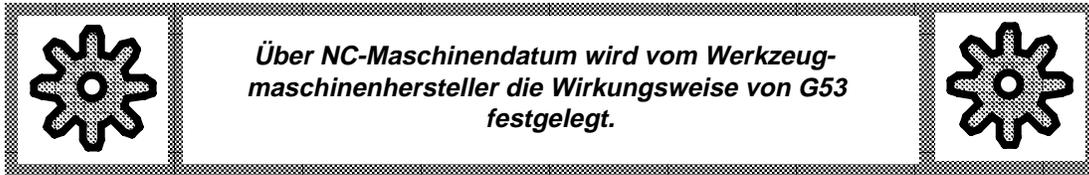
Nullpunktverschiebung mit G59

Anwahl: N.. G59 Y.. LF

Abwahl: N.. G59 Y0 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



Mit G53 erfolgt eine satzweise Unterdrückung der durch

- einstellbare Nullpunktverschiebung (G54 bis G57),
- programmierbare Nullpunktverschiebung (G58, G59) und
- externe Nullpunktverschiebung (von PLC) erreichten Koordinatenverschiebung 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.

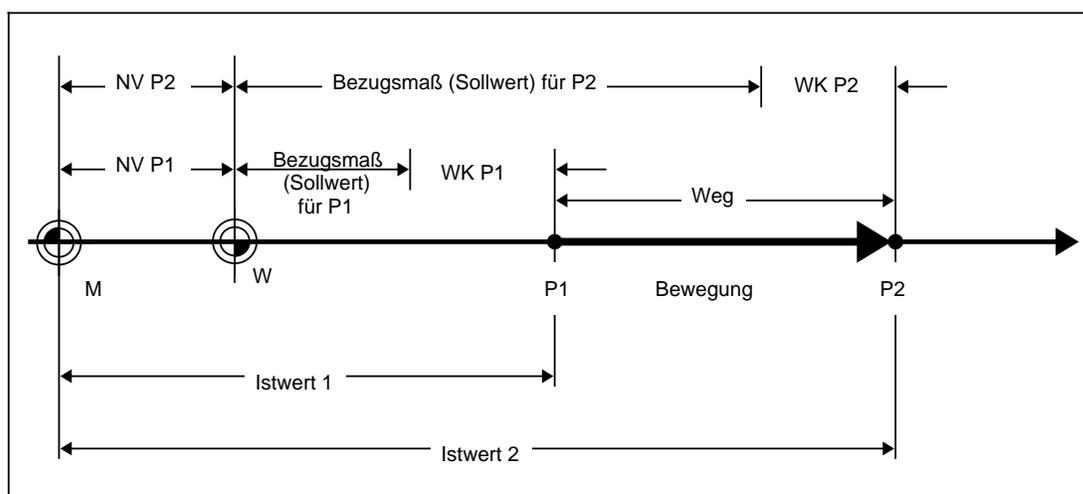
N35 G53 Y.. X.. L_F

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:

Weg=Sollwert – Istwert+Nullpunktverschiebung (NV)+Werkzeugkorrektur (WK).



Wegberechnung bei Bezugsmaßeingabe

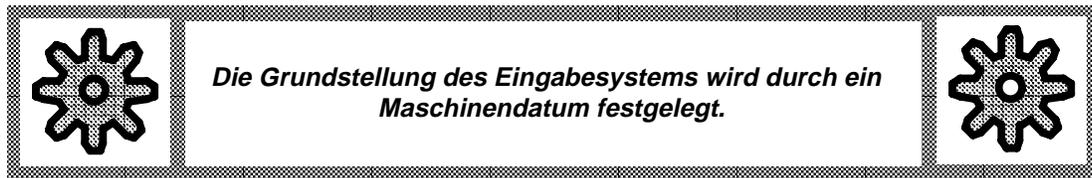
Bei **Kettenmaßeingabe** wird im ersten Satz die Nullpunktverschiebung normal eingerechnet:
 $\text{Weg} = \text{Kettenmaß} + \text{NV} + \text{WK}$.

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 Werkstückvermaßung, 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

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.

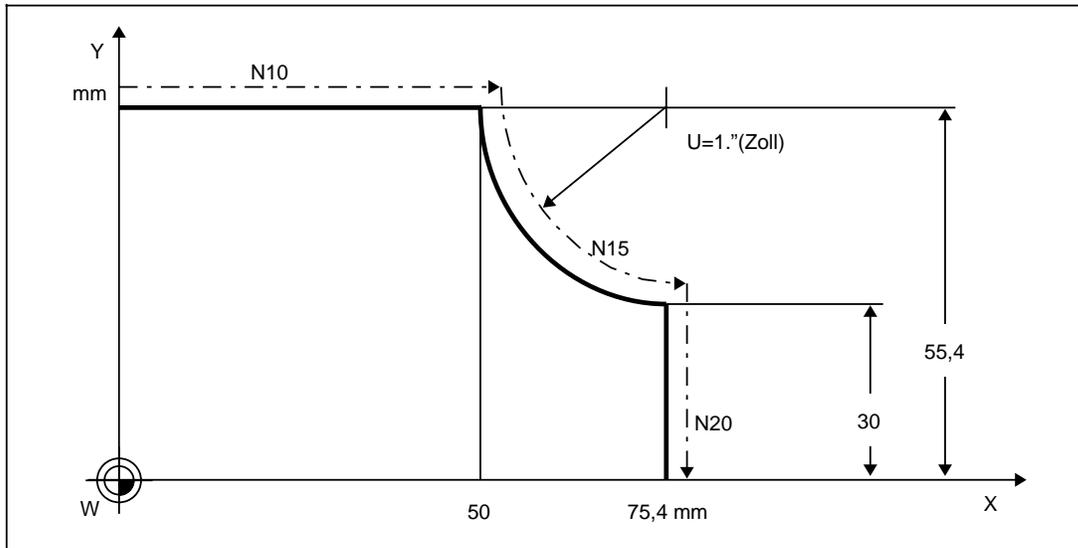
Im ersten Satz muß dann die erforderliche G-Funktion, und nach dem letzten Satz wieder die Grundstellung geschrieben werden (bei Programmende M02, M30 kommt automatisch die Grundstellung).

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, Y, Z
- Interpolationsparameter I, J, K
- Fasen/Radien U-/U
- Parameter, soweit diesen Weginformationen, Interpolationsparametern und Fasen/Radien zugeordnet sind.

Beispiel: G71 – Grundstellung (metrisch)

Eingabe in inch bei Grundstellung G71

N05	..									
N10	G91	X50	L _F							
N15	G03	G70	X1	Y-1	I1	J0	L _F			Kreisbogen in Inch programmiert
N20	G01	G71	Y-30	L _F						Gerade im metrischen Maßsystem programmiert
N25	..									

2.8 Spiegeln

Spiegeln einer Achse

Durch Spiegeln einer Koordinatenachse wird eine Kontur

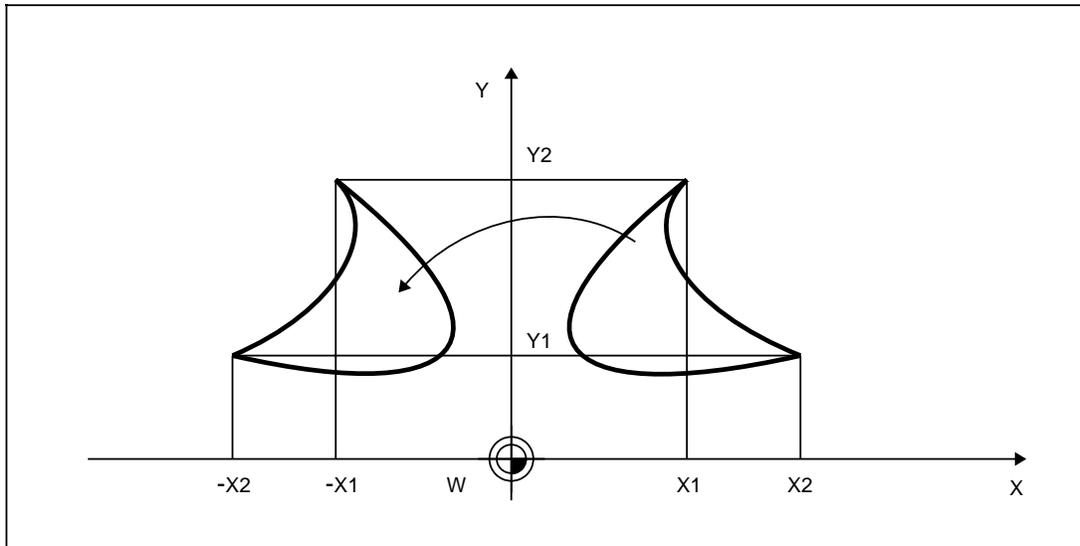
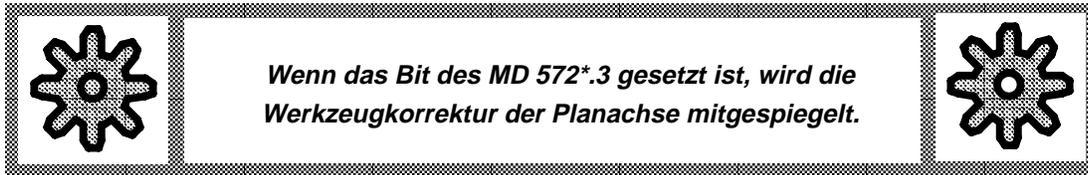
- in derselben Größe,
- im gleichen Abstand zu den anderen Achsen und
- auf der anderen Seite der Spiegelachse spiegelbildlich bearbeitet.

Beim Spiegeln einer Achse vertauscht die Steuerung

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

Keine Spiegelung von:

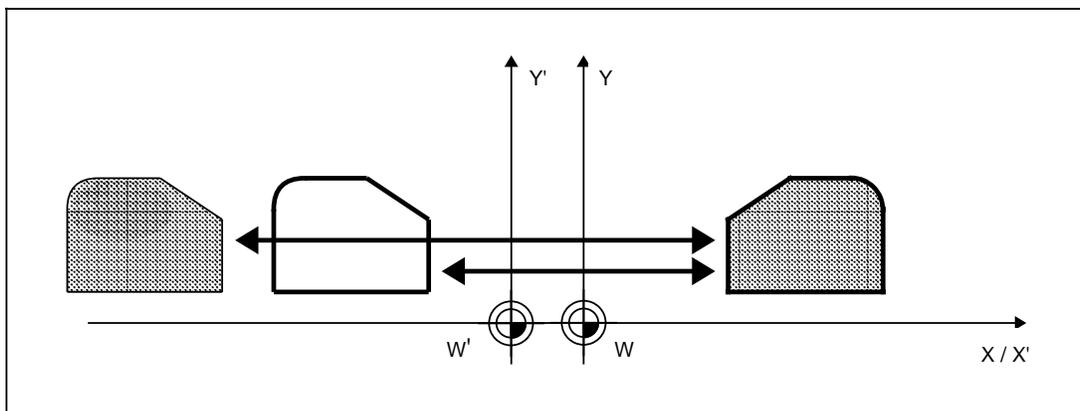
- Werkzeuglängenkorrekturen
- Nullpunktverschiebungen.



Spiegeln der X-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 dem Aufruf der Spiegelung im Programm an die richtige Stelle verschoben (W nach W').



Verschiebung des Werkstücknullpunkts

Spiegeln zweier Achsen

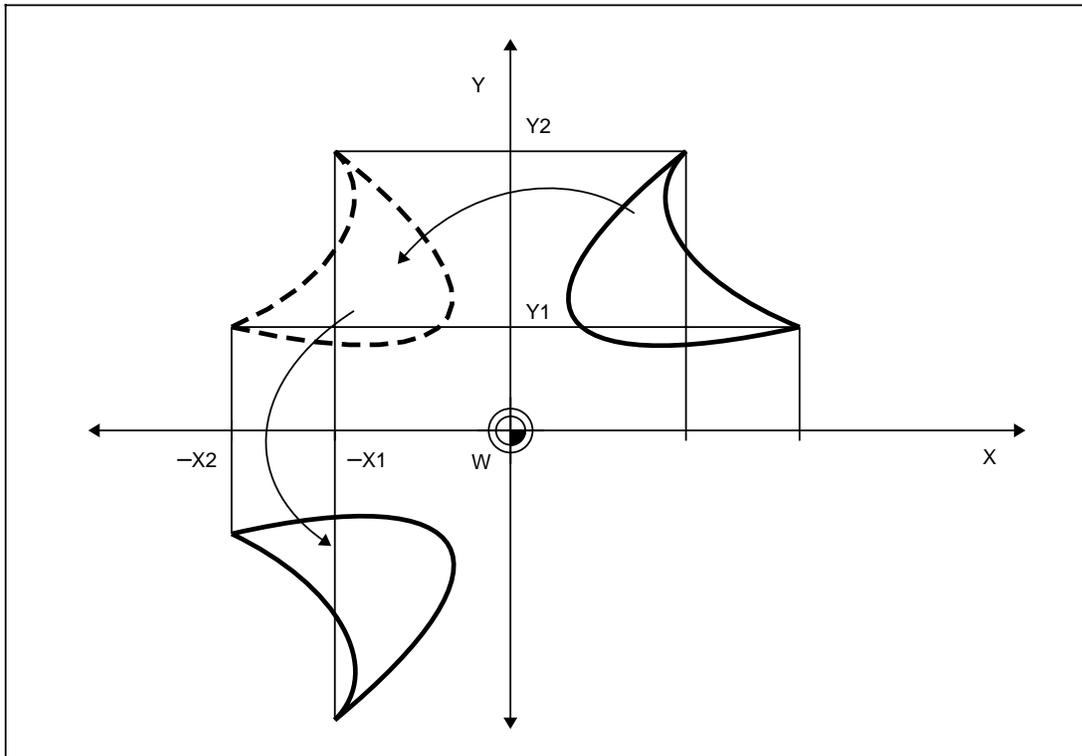
Man führt den beschriebenen Vorgang des Spiegels zweimal aus, z. B.:

- einmal Spiegeln der X-Achse und
- einmal Spiegeln der Y-Achse.

Die Steuerung vertauscht die Vorzeichen der beiden gespiegelten Koordinaten (X/Y). Bearbeitungsrichtung und Drehsinn der Kreise bleiben – da zweimal umgekehrt – gleich.

Keine Spiegelung von:

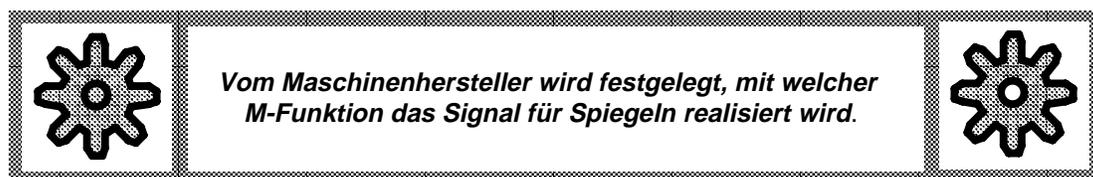
- Werkzeuglängenkorrektur
- Nullpunktverschiebungen.



Spiegeln der X- und Y-Achse

Beim Spiegeln der Hauptachsen wird stets das Werkstück gespiegelt. Programmabschnitte, die gespiegelt werden sollen, werden im Programm geschlossen als Unterprogramm aufgebaut. Dieses Unterprogramm wird dann unter Vorschalten der entsprechenden Spiegelfunktion für die jeweilige Kontur aufgerufen.

Die Funktion "Spiegeln" wird über die PLC angewählt.



Beispiel: Anwahl von Spiegeln

```
:  
N10 G90 G54 G00 X0 Y0 L_F  
N20 Z30 L_F  
N30 G1 Z0 F500 L_F  
N35 M.. L_F  
N36 G04 F.. L_F  
N37 L999 P1 L_F  
:  
N40 X50 Y50 L_F  
N45 M.. L_F  
:  
N55 G04 F.. L_F  
N56 L999 P1 L_F  
:  
N60 G0 G53 Z0 L_F  
N70 G53 X0 Y0 M30 L_F
```

X-Achse spiegeln
Verweilzeit ev. wegen PLC-Zykluszeit
Zwischenspeicher leeren

Aufheben der Achsenspiegelung (L999 L_F; @714 M17 L_F)

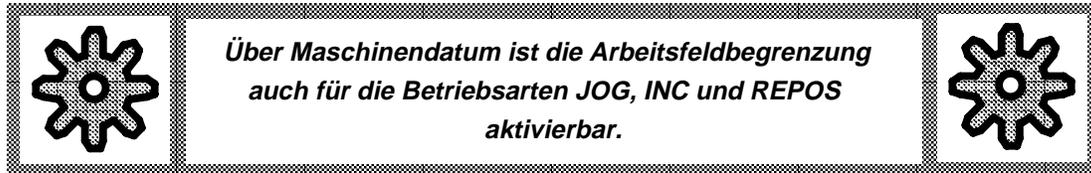
Mit der Sonderfunktion @714 (Zwischenspeicher leer) kann eine weitere Satzinkrement-Berechnung angehalten werden, bis der Zwischenspeicher leer ist.

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 Werkzeugbezugspunkt F darf sich nur im begrenzten Bereich (helles Feld) bewegen. Sobald das Werkzeug den begrenzten Bereich verläßt, sich beim Programmstart außerhalb dieses Bereichs 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 Automatikbetrieb mit den Werten in den Settingdaten wirksam .



Die programmierbare Arbeitsfeldbegrenzung wird mit G25 und G26 aufgerufen:

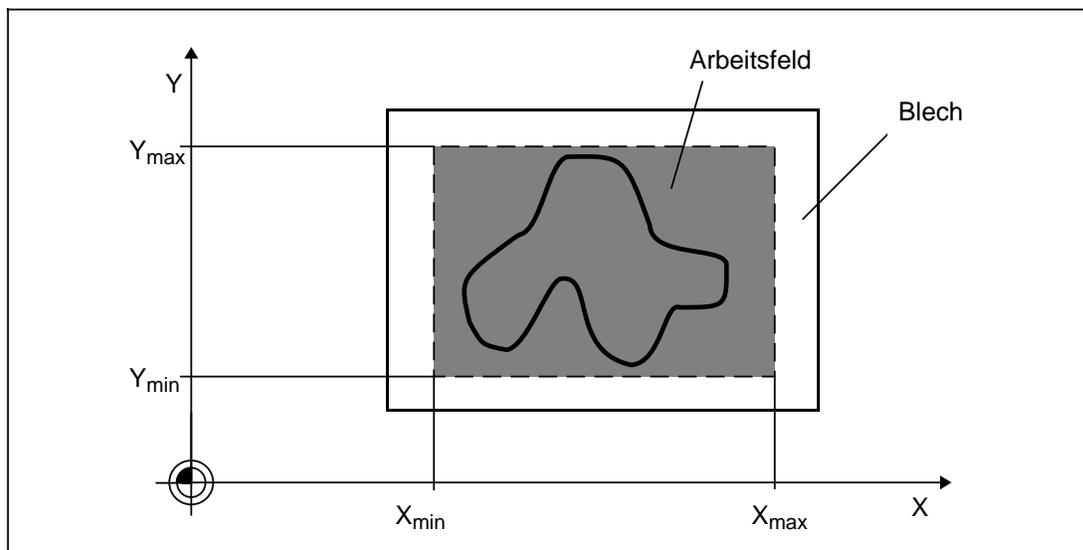
G25 minimale Arbeitsfeldbegrenzung
G26 maximale Arbeitsfeldbegrenzung

```
N10 D25 X-30 Y300 LF
N20 G26 X200 Y500 LF
```

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:

Laserschneiden



Beispiel für Laserschneiden

2.10 Softwaresocken (nur GA3 SW2)

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 "Softwaresocken" können Sie den Bearbeitungstisch in zwei Arbeitsfelder aufteilen.



Die Funktion "Softwaresocken" 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 Schaltenocken 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.

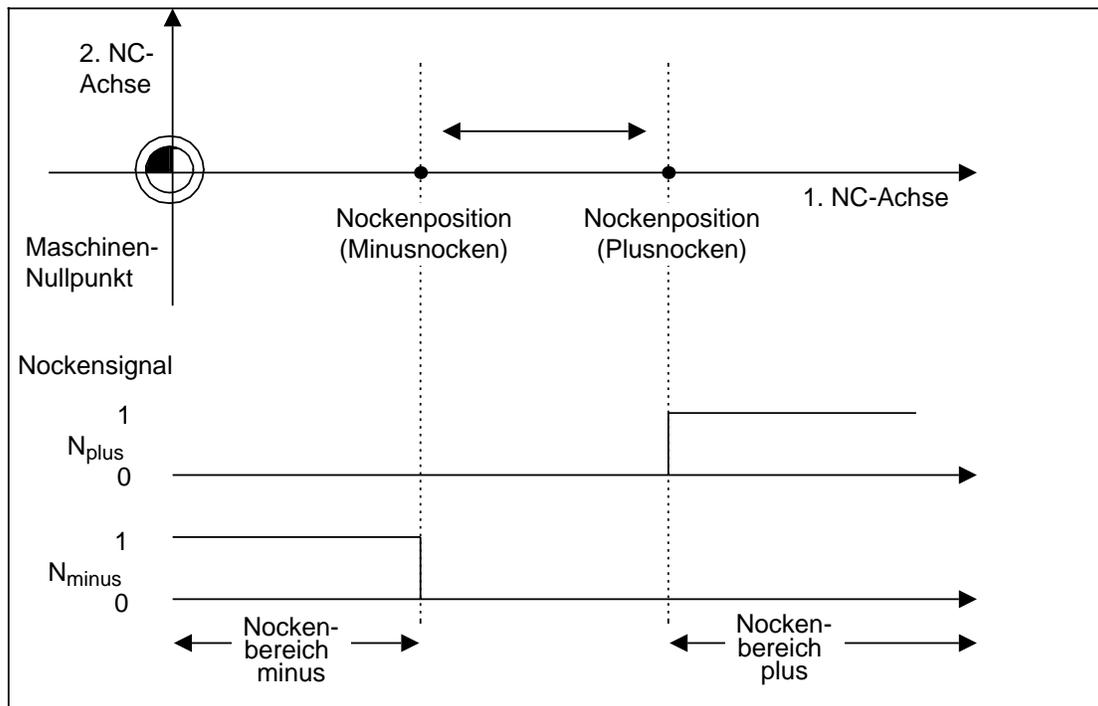
Eine Überprüfung der Nockenpositionen hinsichtlich des maximalen Verfahrbereichs erfolgt nicht.

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

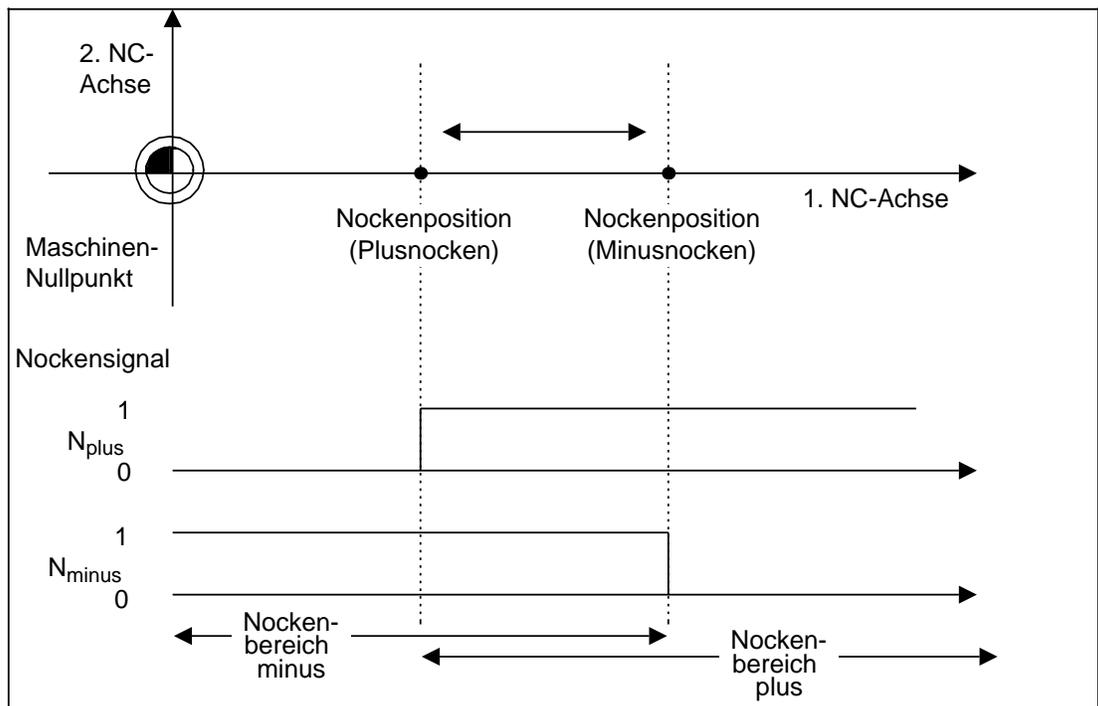
Ein Nockenpaar kann immer nur einer NC-Achse zugeordnet werden,

aber:

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



Minusnocken<Plusnocken



Plusnocken<Minusnocken

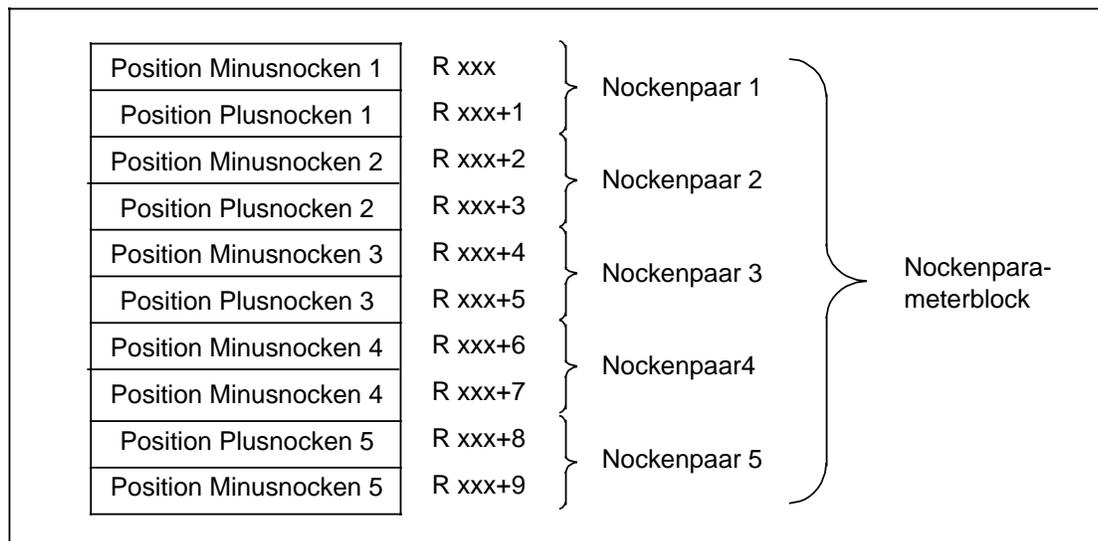
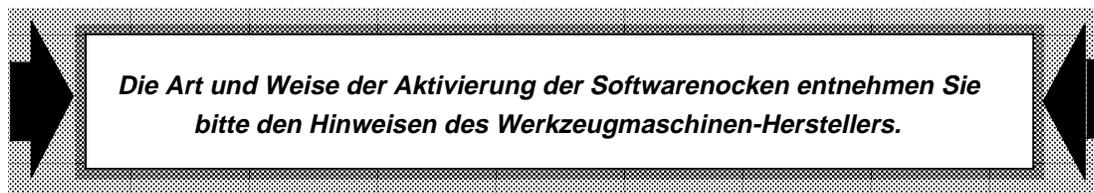
Nockenparameter:

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

In 5 aufeinanderfolgenden R-Paramaterpaaren (**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.

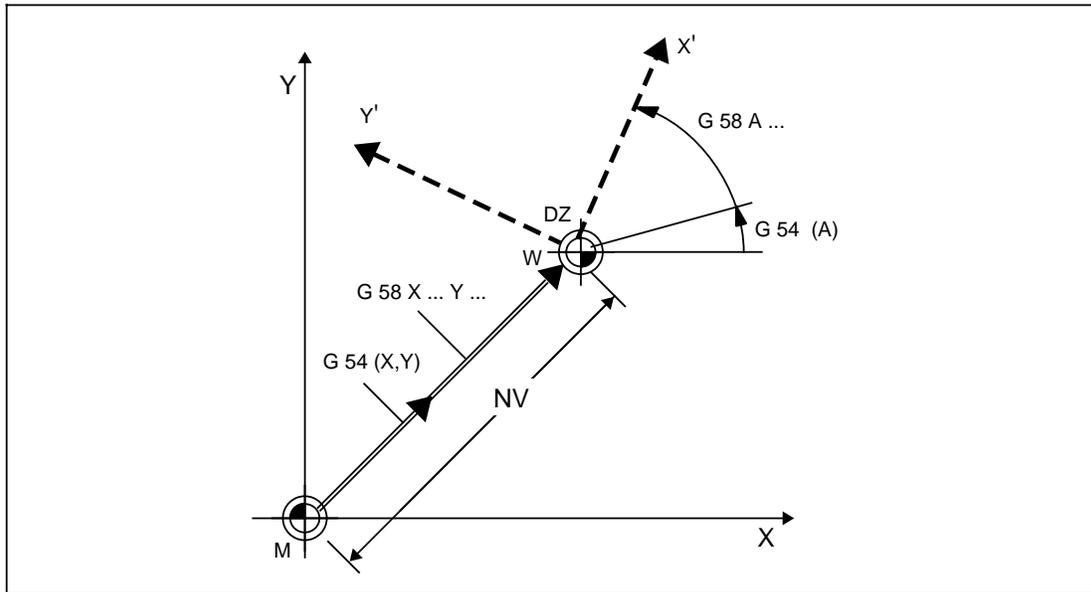


Festlegung der Nockenparameter

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 G58A ... und G59A ...

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 Setting-Daten festgelegt.
- Eingabe des Wertes für Drehwinkel "A" über Tastatur in das Eingabebild 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 Setting-Daten für G54 eingetragene NV und die im Setting-Datum eingetragene KD G54A ... aktiviert.

Beispiel: einstellbare Koordinatendrehung

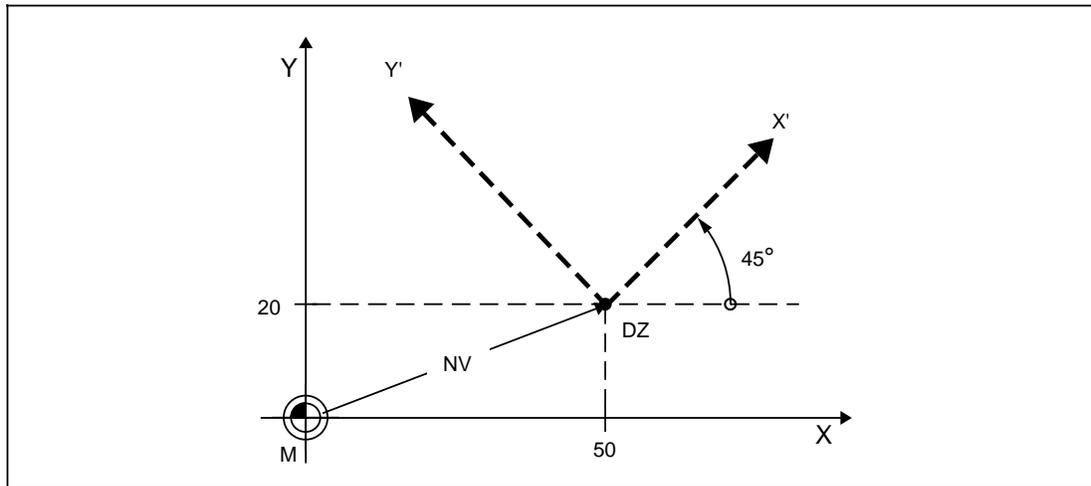
```
N..
N.. G90 G54 L_F
N..
```

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 X50 Y20 A45 L_F  
N..
```

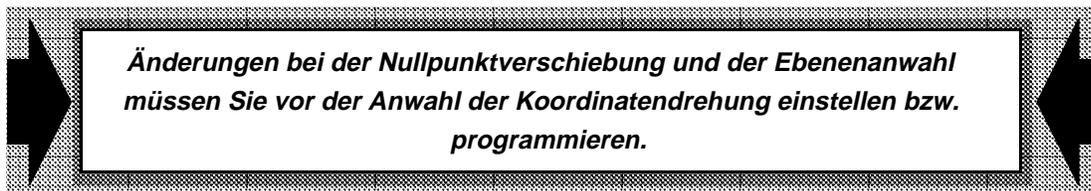


Beeinflussung durch Programmierung

- 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 (Setting-Daten).

Weitere Kennzeichen:

- Die Koordinatendrehung ist in der Betriebsart "AUTOMATIC" ablauffähig.
- Die KD wirkt **kanalspezifisch**; bei der "SIMULATION" werden die für **Kanal 3** (Simulationskanal) eingegebenen Koordinatendrehungen wirksam.
- Bei der "ISTWERTANZEIGE" wird die KD **nicht** berücksichtigt.
- Unmittelbar nach einer KD darf keine Kreisinterpolation programmiert werden.
- Bei angewählter FRK:
 - 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 Koordinatendrehung“ (@715) verbunden. Diese Funktion wird von der Satzaufbereitung intern ausgelöst, muß daher also nicht programmiert werden.



2.12 Maßstabsänderung: Anwahl G51, Abwahl G50

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

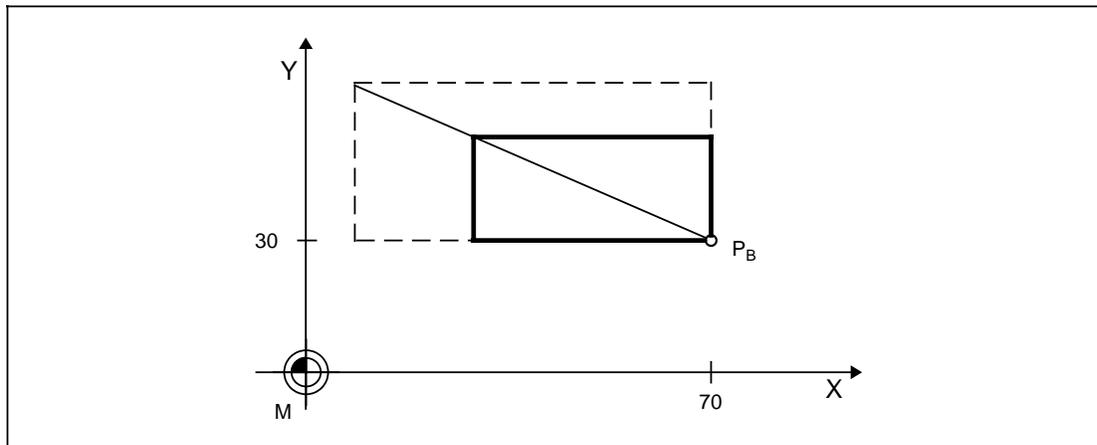
- Eine Maßstabsänderung wird mit der Programmierung von G51 wirksam.
- G51 ist selbsthaltend und wird erst durch G50 (Abwahl der Maßstabsänderung) aufgehoben.
- Die Maßstabsänderung orientiert sich an einem Bezugspunkt P_B (Maßstabszentrum). Die Koordinaten (X, Y) von P_B werden beim Programmieren angegeben. Werden X und Y nicht angegeben, so generiert die Steuerung die Werte $X=0$ und $Y=0$ (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.

Beispiele: Mögliche Schreibweisen bei der Programmierung

```
a) N..
   N.. G51 X70 Y30 P1,5 L_F
   N..
```

oder

```
b) N..
   N.. G51 P1,5 L_F
   N.. G51 X70 Y30 L_F
   N..
```



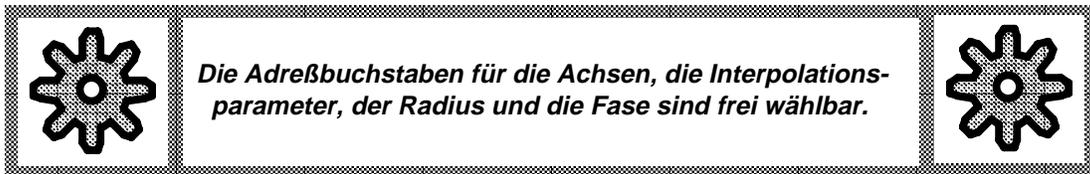
**Maßstabsfaktor (kanalspezifisch) und Bezugspunkt werden auf dem Bildschirm unter "SETTING-DATEN" angezeigt.
Die Maßstabsänderung wird für die jeweilige Achse durch das SettingDatum 560*, Bit 2 freigegeben.**

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, Y, Z.

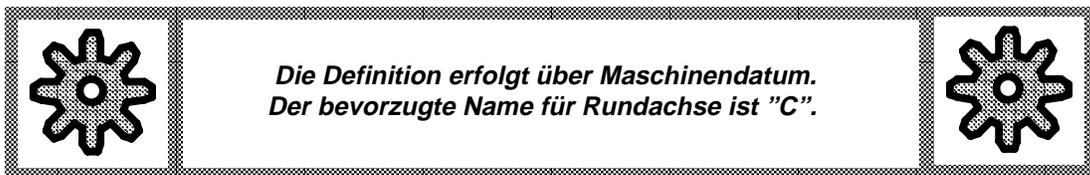
Für weitere Achsen stehen wahlweise die Adressen A, B, C, U, V und W zur Verfügung. Die erweiterte Adreßschreibweise ist auch möglich (GA 3).



Nachfolgend werden die vorzugsweise bei Stanz- und Nibbelmaschinen benutzten Adreßbuchstaben verwendet.

Rundachse

Jede Achse kann als Rundachse eingesetzt werden.



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

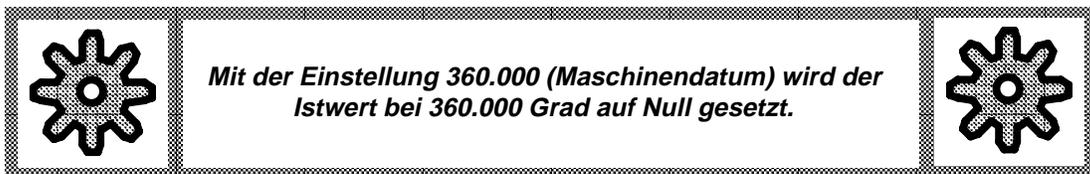
Das Vorzeichen gibt die Verfahrrichtung an, mit der auf die absolute Position innerhalb einer Umdrehung positioniert wird.

Bei Bezugsmaßprogrammierung auf kürzestem Wegt (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) beträgt der Verfahrbereich ± 99999.999 Grad.

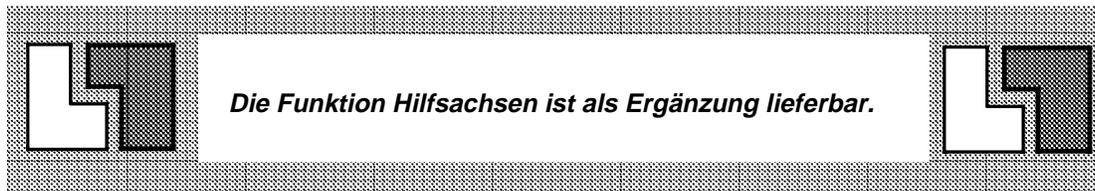
Es können **mehrere Achsen gleichzeitig** als Rundachsen erklärt werden.

Die Rundachsen können generell endlos drehen.

Programm-Ausschnitt:

```
:  
:  
N5 G91 C 99999.999 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.).

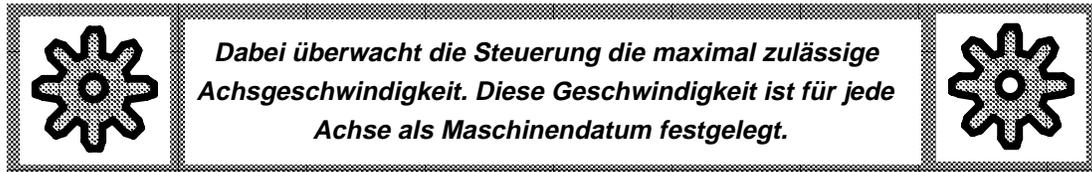
Sofern die Hilfsachse nicht als **"NC-Achse"** im Teileprogramm programmiert wird, kann eine Synchronisierung zu den NC-Hauptachsen auch über die Anpaßsteuerung realisiert werden. Wegmessung, Verfahrbereich, Eingabefeinheit, Geometrie- und Lageregelfeinheit entsprechen einer NC-Achse.

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).



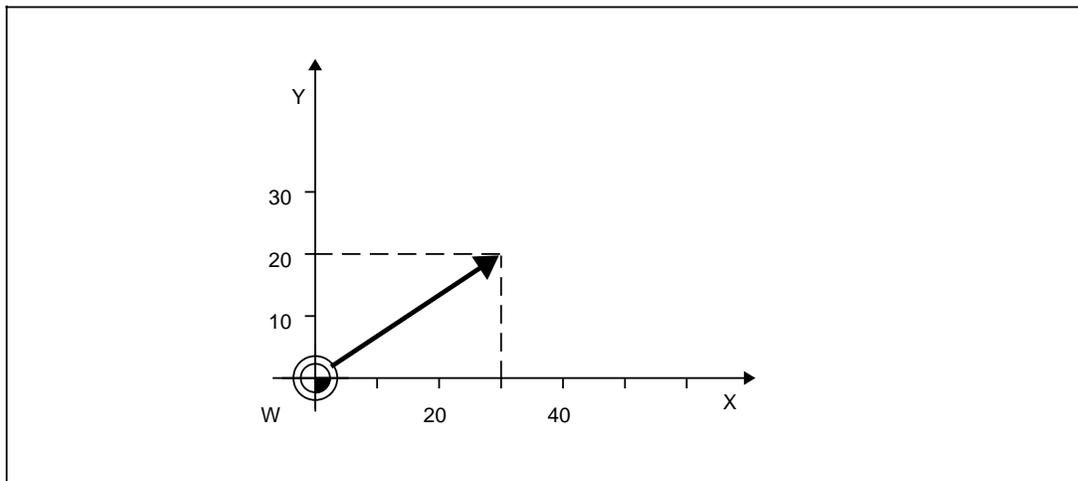
Wird die Eilgangbewegung in mehreren Achsen gleichzeitig ausgeführt, so wird die Verfahrensgeschwindigkeit durch den kleinsten Wert der als MASCHINENDATUM festgelegten Achsgeschwindigkeiten bestimmt. Die Wegbedingung G00 bewirkt automatisch Genauhalt grob.

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

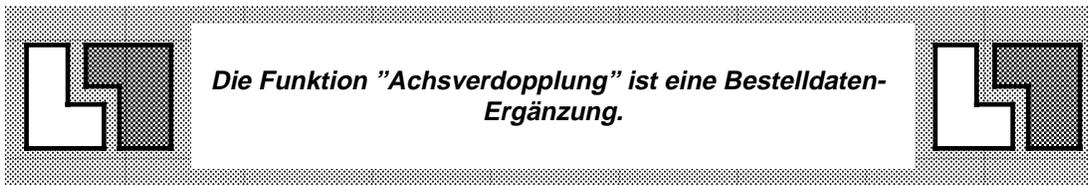
Beispiel:

N5 G00 G90 X30 Y20 L_F

Eilgang, Bezugsmaßvorgabe



3.1.2 Achsverdopplung (nur GA3 SW2)



Die Funktion "Achsverdopplung" bewirkt eine Verdopplung der programmierbaren Hauptachsen im Rahmen der Anzahl der maximal zugelassenen Achsen.

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

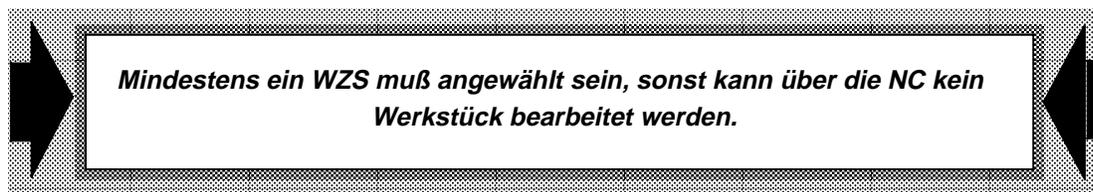
Mit der Funktion "Achsverdopplung" arbeitet die NC **ein** Teileprogramm simultan in **zwei getrennten Werkzeugsystemen (WZS)** ab, wobei im Teileprogramm nur ein WZS programmiert ist (WZS1).

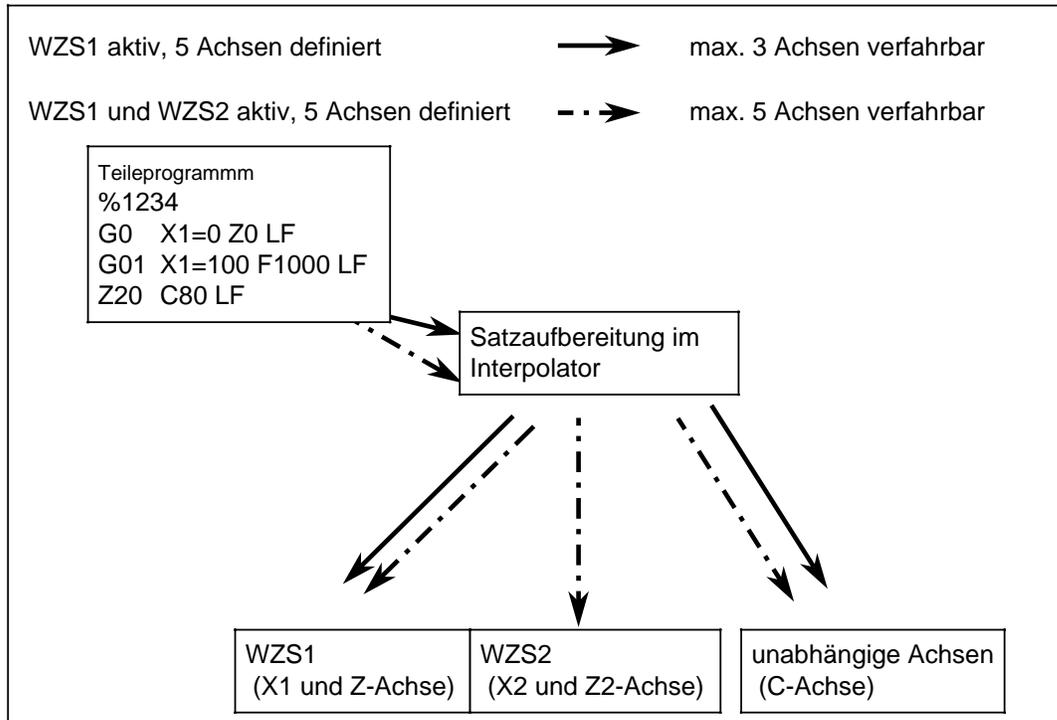
Es werden dadurch die **führenden (programmierten) Achsen**, z.B. die X1-Achse und die Z-Achse, **im WZS1** verfahren, die **verdoppelten Achsen**, z.B. die X2-Achse und die Z2-Achse, **im WZS2** verfahren.

Die Achsen des WZS2 dürfen im Teileprogramm **nicht** programmiert werden. Die Namen und Verfahrwege 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 angewählte Funktion (WZS1 oder WZS2 bzw. beide WZS aktiv) aktiviert und abgespeichert. Die angewählte Funktion ist selbsthaltend bis zum nächsten Programmende bzw. RESET und kann während der Programmbearbeitung nicht geändert werden (Alarmmeldung).



Teileprogramm ohne oder mit Achsverdopplung:

Wirkungsweise bei aktivem WZS1:

Teileprogramm

```

N0010 G0 X1=0 Z D5 LF
N0020 G01 X1=100 F2000 LF
N0030 Z30 C50 LF
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.

Wirkungsweise bei aktivem WZS1 und WZS2 (Achsverdopplung):

Teileprogramm

```

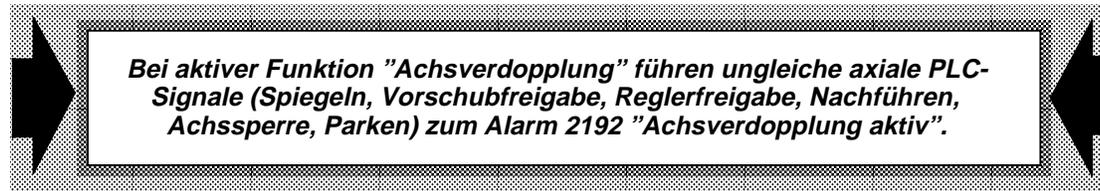
N0010 G0 X1=0 Z D5 LF
N0020 G01 X1=100 F2000 LF
N0030 Z30 C50 LF
N0050...
  
```

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.

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.
- Ein Maßstabsfaktor im WZS1 wirkt auch im WZS2.

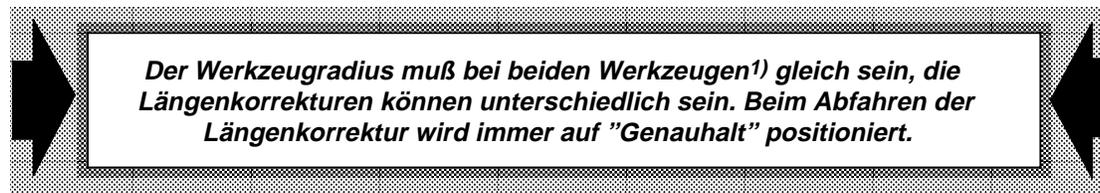


Werkzeugverwaltung, -korrektur:

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:

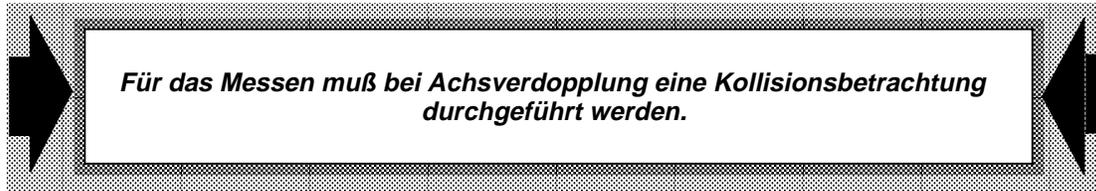
Bei aktiver Achsverdopplung und MD 5011 Bit 3= 1 bzw. Bit 4= 1 werden beim Werkzeugtyp P1=0 die Länge P2 und die Längenkorrektur P5 im Durchmesser gerechnet.

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 Verfahrenweg 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.

3.2 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 und
- mit dem programmierten Vorschub.

Es können achsparallele und unter beliebigen Winkeln verlaufende Bewegungen ausgeführt werden.

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 und
- vom Ausgangspunkt auf einer Kreisbahn zum programmierten Endpunkt.

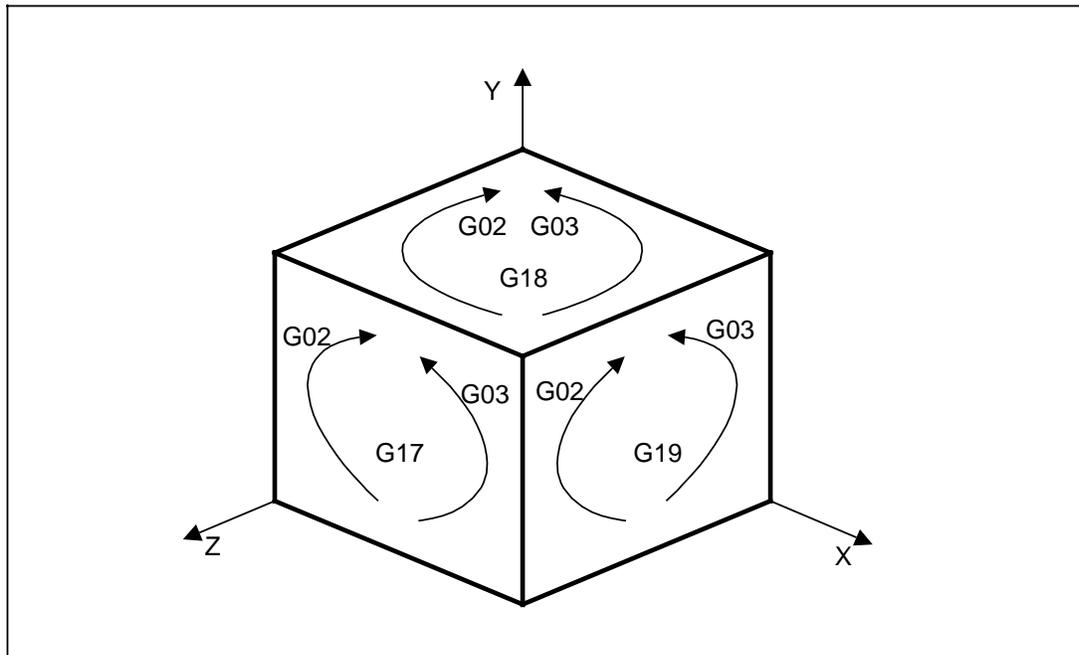
Die Wegbedingungen **G02** und **G03** sind **modal** (speichernd) wirksam.

Es kann in jeder Ebene eine Kreisbewegung ausgeführt werden (XY, ZX oder YZ).

Die Kreisebene wird durch die im Kreissatz zuerst programmierten Achsen bestimmt. Wird nur eine Achsposition programmiert (z. B. bei Halbkreis), so wird die Kreisebene durch die angewählte Ebene bestimmt.

Stimmen Ebene und die über MASCHINENDATUM festgelegten Interpolationsparameter nicht überein, wird Alarm 3006 (falsche Satzstruktur) ausgelöst.

Die Drehrichtung in den verschiedenen Ebenen ist folgendermaßen festgelegt: man blickt gegen die Richtung der Achse, die senkrecht auf der Ebene steht. Mit G02 bewegt sich das Werkzeug im Uhrzeigersinn, mit G03 gegen den Uhrzeigersinn.



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, Y und Z fest.
- Der Kreismittelpunkt "**KM**" wird festgelegt
 - entweder durch die Interpolationsparameter oder
 - direkt über den Radius (U).

3.2.2.1 Interpolationsparameter I, J, K

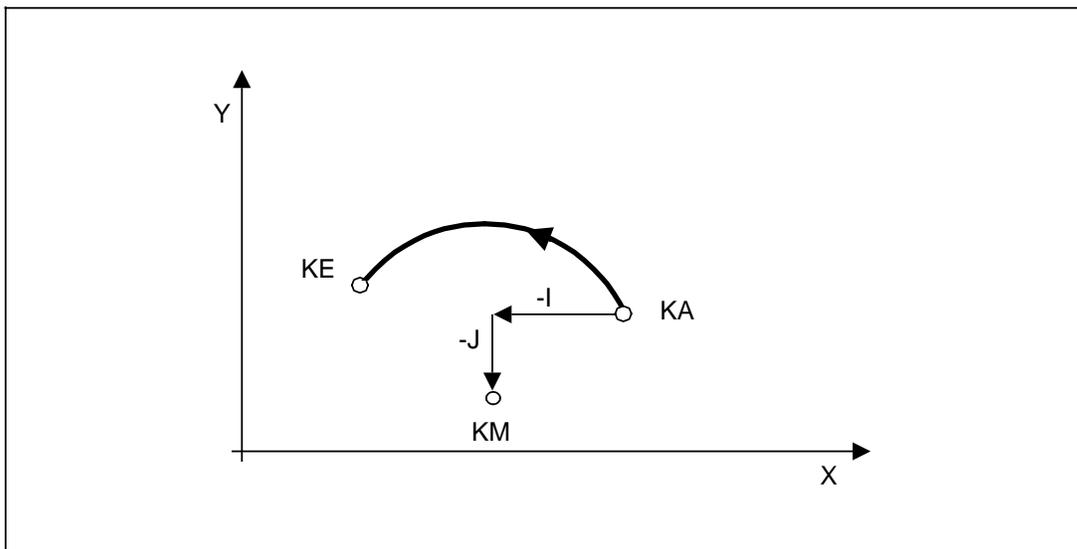
Die Interpolationsparameter sind die achsparallelen Koordinaten des Abstandsvektors vom Anfangspunkt zum Mittelpunkt des Kreises.

Nach DIN 66 025 sind den Achsen **X**, **Y** und **Z** die Interpolationsparameter **I**, **J** und **K** zugeordnet. Unabhängig davon ob X, Y und Z im Bezugsmaß oder im Kettenmaß programmiert sind, müssen die Interpolationsparameter immer im **Kettenmaß** und in der **richtigen Reihenfolge** eingegeben werden.

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

Nicht programmiert werden:

- ein Interpolationsparameter der den Wert 0 hat und
- die Endpunktkoordinaten, die sich gegenüber dem Kreisumfang nicht geändert haben. Beim Vollkreis muß mindestens eine Achse programmiert werden (X0, Y0 oder Z0).

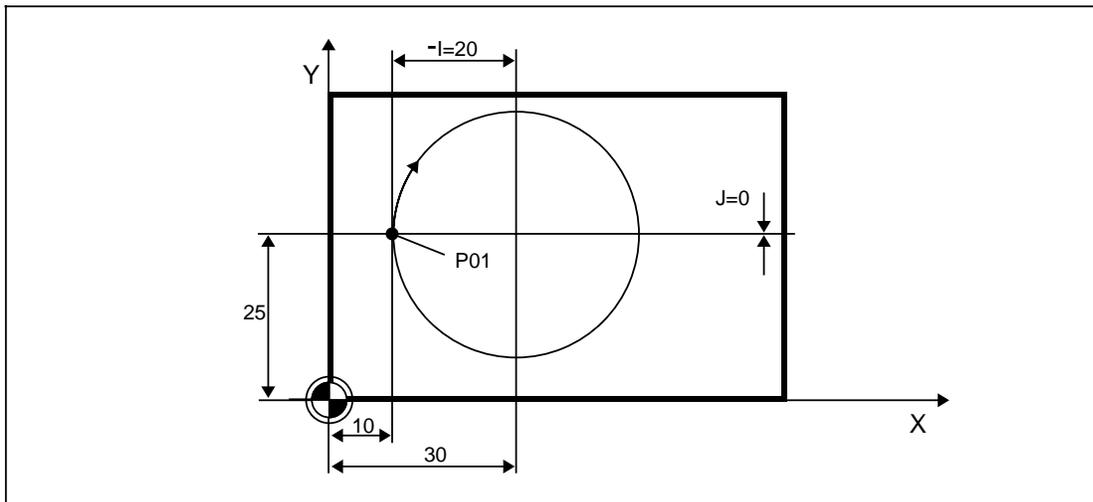


Kreisinterpolation mit Interpolationsparametern

Beispiel:

Vollkreis in der X-Y-Ebene

Im folgenden Beispiel wird ein Vollkreis programmiert. In diesem Fall muß der Startpunkt auch als Endpunkt eingegeben werden.



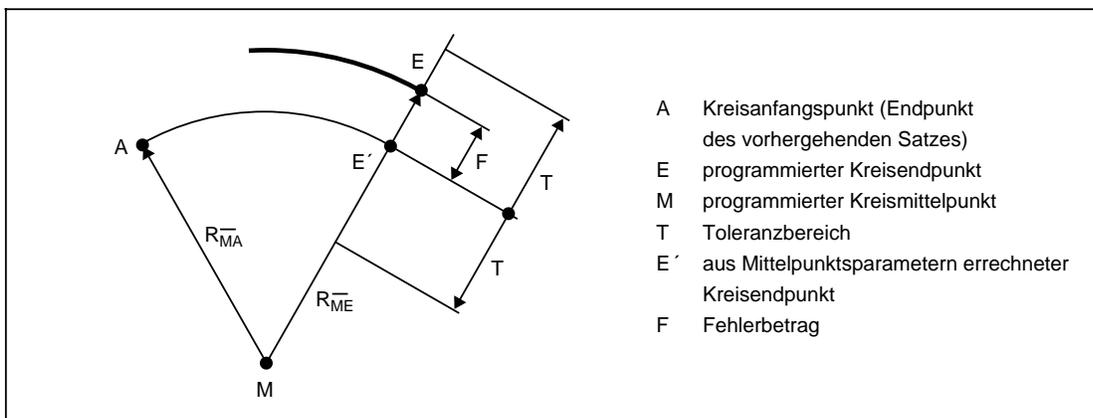
Vollkreis in der XY-Ebene

```
% 40 LF
N1 G00 X10 Y25 F125 LF
N3 G02 X10 Y25 I20 J0 LF
N5 X-20 LF
N6 M30 LF
```

- N1 Werkzeug fährt im Eilgang auf Punkt P01.
- N3 X-Y-Ebene ist automatisch angewählt (Grundstellung). Werkzeug fährt einen Vollkreis im Uhrzeigersinn (G02).
- N6 Programmende.

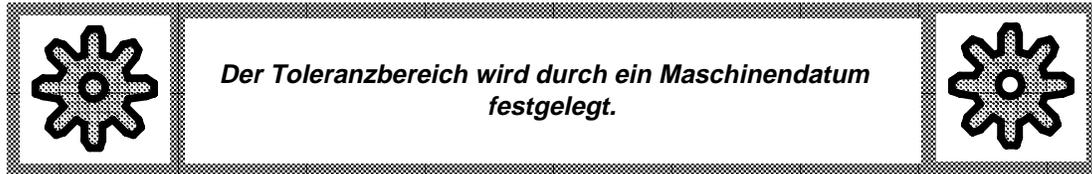
Kreisendpunktüberwachung

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.



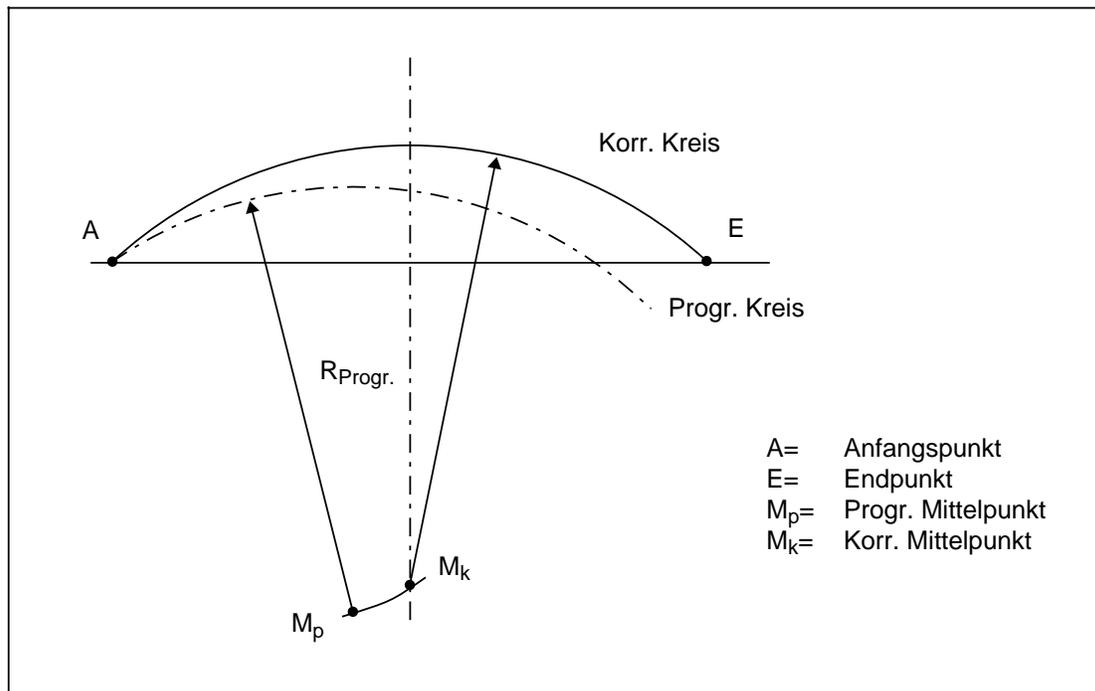
Definition des Toleranzbereichs der Kreisendpunktüberwachung

Bei ungenauer Eingabe des Radius oder der Interpolationsparameter überschreitet die Differenz (F) den Toleranzbereich (T), und der Kreissatz wird nicht bearbeitet. Es erfolgt Alarmmeldung "Kreisendpunktfehler".



Liegt die Abweichung der Radien innerhalb des Toleranzbereiches, 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 Vermaung einer Zeichnung so gewhlt, da es gnstiger ist, fr die Festlegung der Kreisbahn den Radius (z.B. B) anzugeben. Die erweiterte Adreschreibweise fr den Radius ist dabei ausgeschlossen.

Die Radiusangabe erhlt demnach folgendes Vorzeichen:

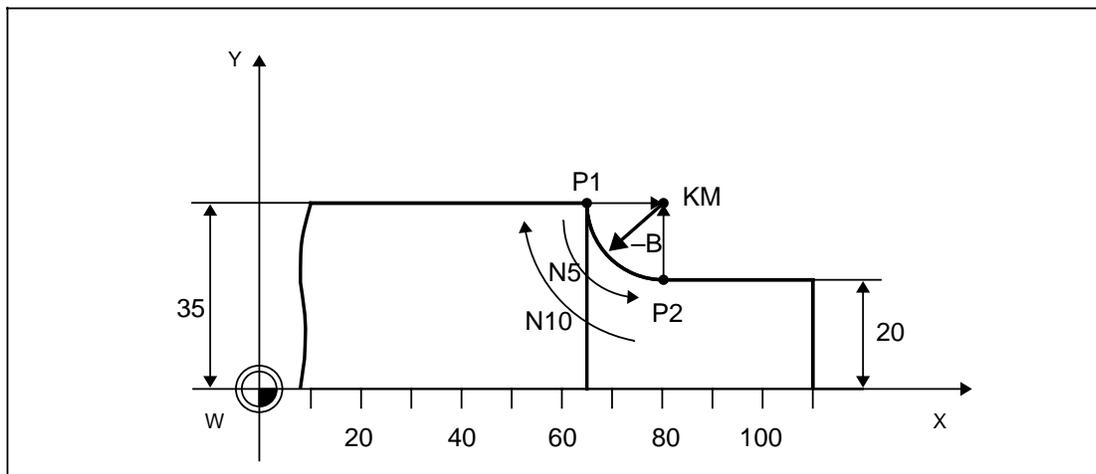
- +B: Verfahrwinkel kleiner oder gleich 180 Grad
- B: Verfahrwinkel grer 180 Grad

Bei einem Verfahrwinkel von 0 Grad oder 360 Grad ist die Radiusprogrammierung nicht zulssig. Vollkreise mssen also mit Interpolationsparametern programmiert werden.

G02 oder G03 bestimmt die Bewegungsrichtung auf dem durch Kreisendpunkt und Interpolationsparameter bzw. Radius B bestimmten Kreis.

Programmbeispiel:

Laserschneiden



Kreisinterpolation

```
N5 G03 G90 X80 Y20 I0 J15 L_F  
N10 G02 X65 Y35 I15 J0 L_F
```

N5 Werkzeug verfhrt von P1 nach P2
N10 Werkzeug verfhrt von P2 nach P1
(Interpolationsparameter)

oder

```
N5 G03 G90 X80 Y20 B+15 L_F  
N10 G02 X65 Y35 B+15 L_F
```

N5 Werkzeug verfhrt von P1 nach P2
N10 Werkzeug verfhrt von P2 nach P1
(Radiusprogrammierung)

3.2.3 Zylinderinterpolation (nur GA3 SW2)

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. Die Eingabe der Interpolationsparameter I, J und K ist nicht erlaubt.

SPLINE-Interpolation 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 mm bei Eingabefinheit 10^{-3} inch.

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

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

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.

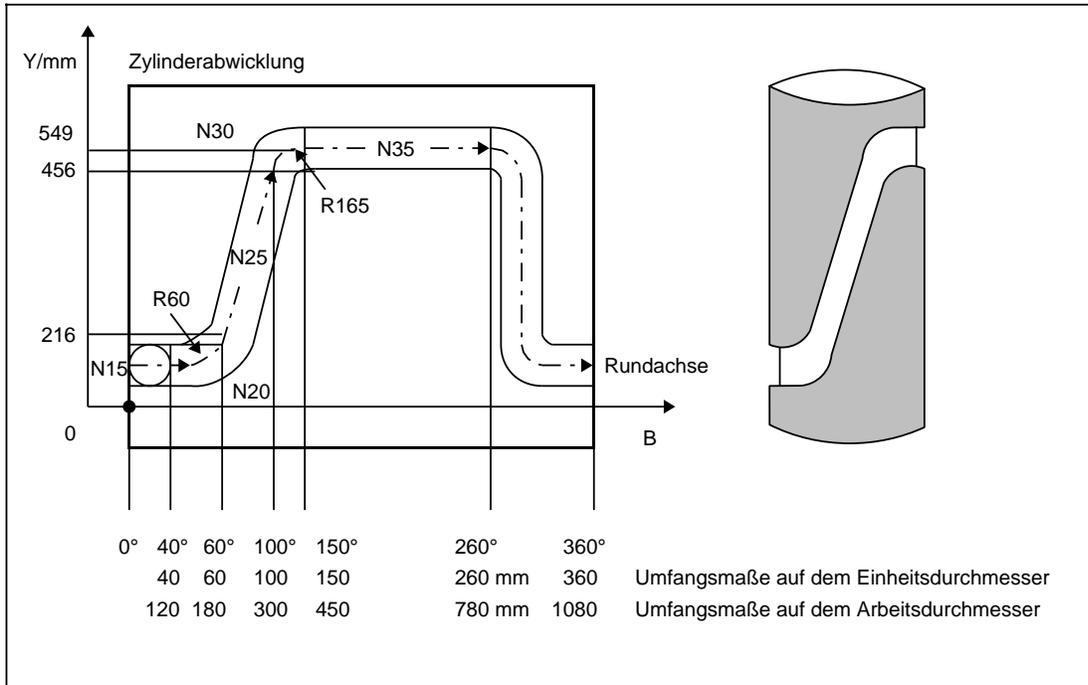
Die Eingabefinheit für P beträgt 10^{-5} und der Wert für P ist auf maximal < 100 begrenzt. Der Bearbeitungsdurchmesser ist modal wirksam, bis eine erneute Programmierung oder Rücksetzen mit M02/M30 bzw. RESET erfolgt. Die programmierte Vorschubgeschwindigkeit wird auf der Mittelpunktbahn eingehalten. 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 einsetzen. Die An-/Abwahl der Zylinderinterpolation ist nur bei abgewählter FRK möglich. Die Funktion "Weiches An-/Abfahren" dürfen Sie nicht zusammen mit der Zylinderinterpolation anwenden.

Beispiel: Zylinderinterpolation



Zylinderinterpolation

```

%10 LF
N10 G92 P3 C LF                               Anwahl der Zylinderinterpolation
N15 G01 B40 Y.. F1000 S800 M3 LF
N20 G03 B60 Y216 U+60 LF                       Radius R60 (U60)
N25 G01 B100 Y456 LF
N30 G02 B150 Y549 U+165 LF                   Radius R165 (U165)
N35 G01 B260 LF
:
N60 G92 P1 C LF                               Abwahl der Zylinderinterpolation
N70 M02 LF
    
```

3.2.4 Polarkoordinaten

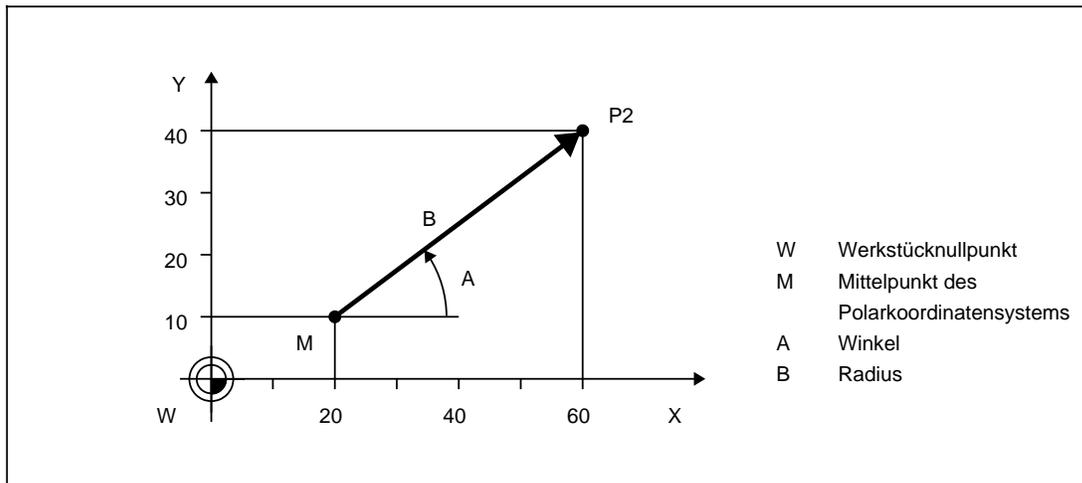
3.2.4.1 Polarkoordinaten G10/G11/G12/G13

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.

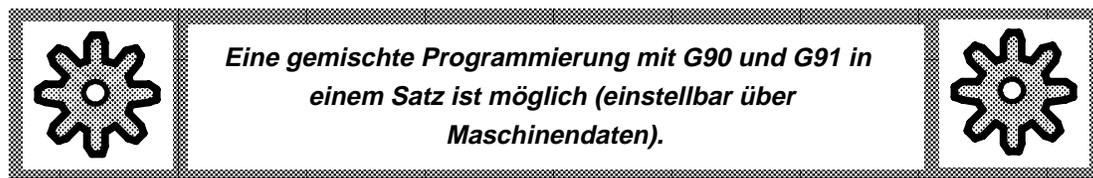


Zur Bestimmung des Verfahrenweges benötigt die Steuerung die **Angabe des Mittelpunktes**, des **Radius** und des **Winkels**. Der Mittelpunkt wird mit rechtwinkligen Koordinaten (X, Y, Z) und bei erstmaliger Programmierung im Bezugsmaß eingegeben. Eine spätere Kettenmaßeingabe (mit G91) bezieht sich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

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

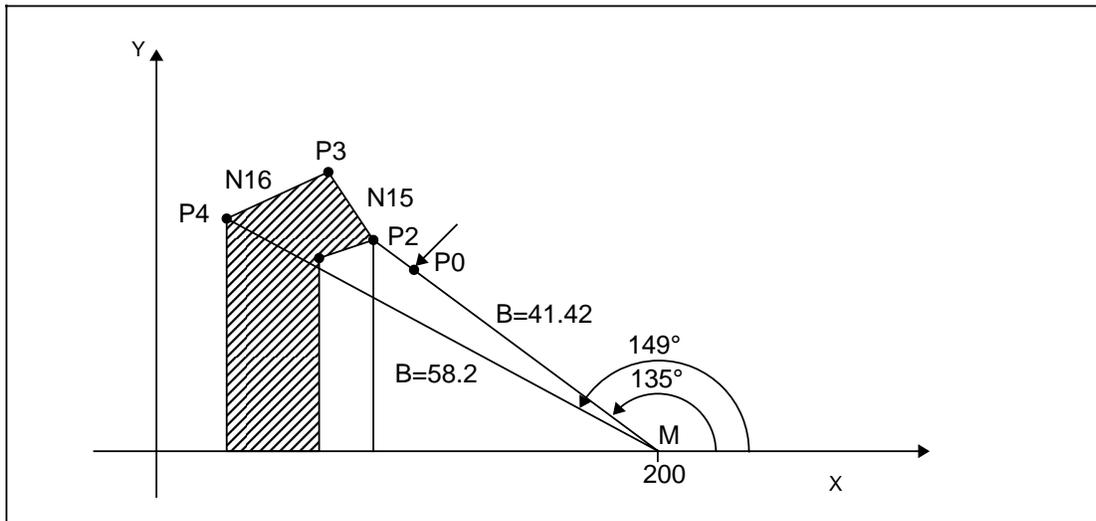
Der **Radius** wird unter der **Adresse B** ohne Vorzeichen programmiert. Der **Winkel** wird unter der **Adresse A** ohne Vorzeichen eingegeben (Eingabefeinheit 10^{-5} Grad). Er bezieht sich immer auf die zuerst programmierte positive Achse des Mittelpunkts-Koordinaten (Bezugsachse).

Die positive Richtung dieser Achse entspricht einem Winkel von 0 Grad. Die Winkelangabe ist absolut oder inkrementell und positiv.



Beispiel:

G10, G11 Laserschneidmaschine



```
N14 G90 G10 X200 Y0 B30 A+135 LF (P0)  
N15 G11 B56.56 LF (P3)  
N16 B58.2 A+149 LF (P4)
```

N14 im Eilgang Startposition P0 anfahren
N15 Außenkontur des Drehteils mit Vorschub F... bearbeiten.

3.2.4.2 Polarkoordinaten G110/G111

Die Funktionen G110 und G111 dienen zur vereinfachten Nullpunkt-Programmierung bei Polarkoordinaten.

G110: Erreichte Sollposition als neuen Mittelpunkt übernehmen.

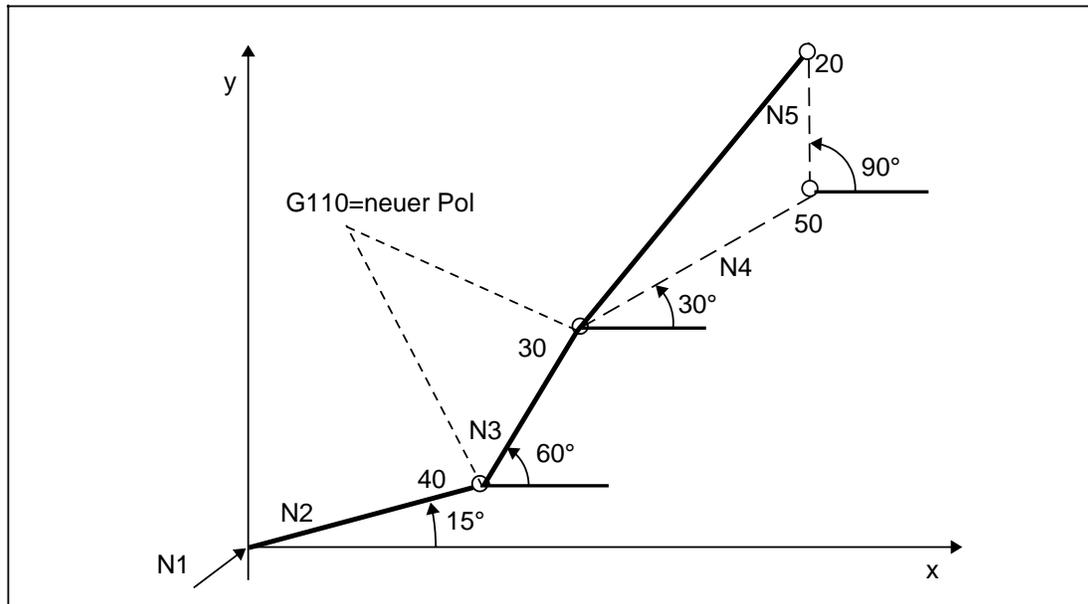
G111: Mittelpunktprogrammierung mit Winkel und Radius, ohne Achsbewegung.

Diese Funktionen sind satzweise wirksam. Sie sind nur für Geradeninterpolation gültig.

Vor einem Satz mit G110 muß ein Satz mit G10 oder G11 programmiert sein.

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

Beispiel: Polarkoordinatenprogrammierung



```

N1 G0 G90 X0 Y0 LF
N2 G11 X0 Y0 A15 B40 LF
N3 G110 A60 B30 LF
N4 G111 A30 B50 LF
N5 G110 A90 B20 LF

```

3.2.5 Vorschub F, G94/G95

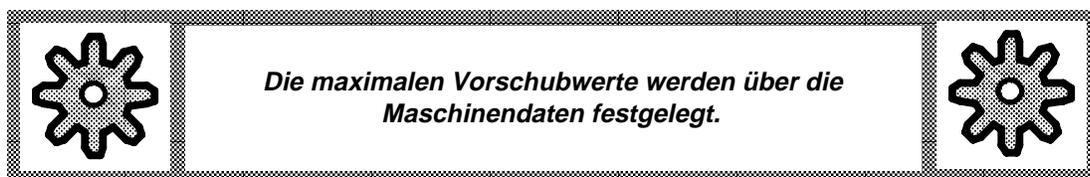
Der Vorschub F wird in mm/min (m/min) oder in mm/U programmiert:

G94 F.. Vorschub in mm/min
G95 F.. Vorschub in mm/U

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 Wechsel der G-Funktion G94/G95 muß spätestens im nächsten Satz der F-Wert neu programmiert werden.

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



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

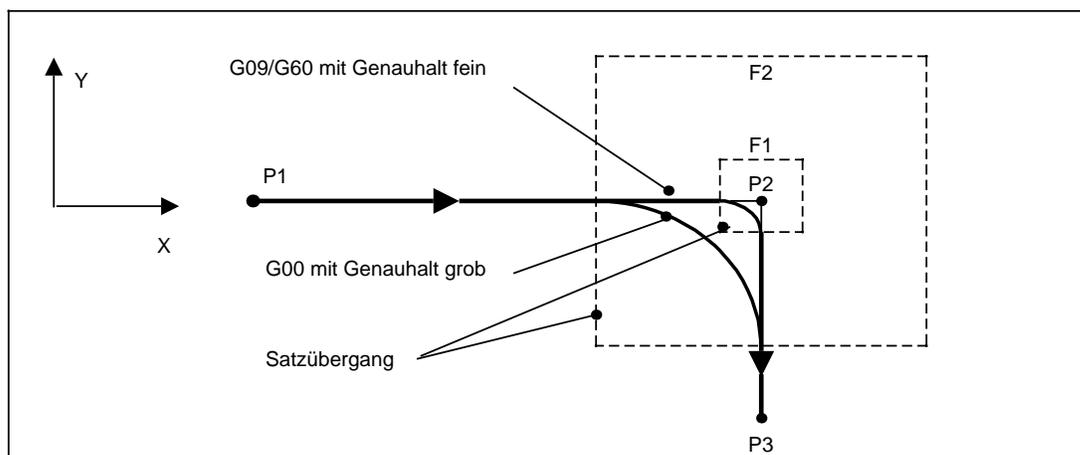
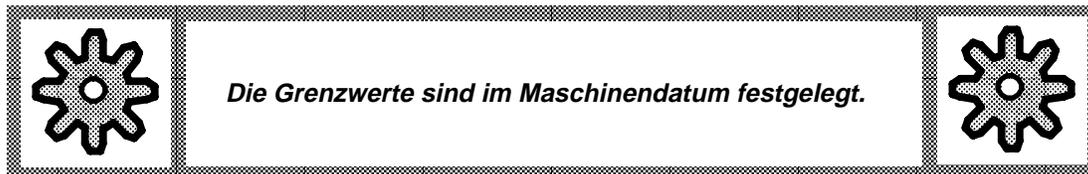
3.2.6.1 Genauhaltgrenze fein und grob G09/G60/G00

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) 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.

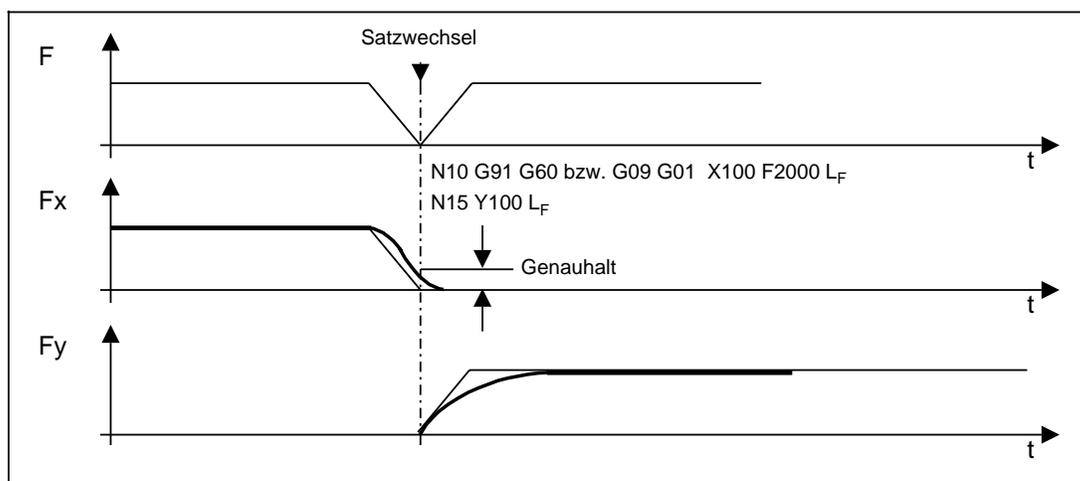
Die Funktion G09 bzw. G60 kann z. B. verwendet werden, wenn scharfe Ecken zu bearbeiten sind, beim Einstechen oder bei einer Richtungskehr.

Genauhaltgrenze fein und grobGenauhalt 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 Lageregel innerhalb der NC ergibt sich ein verschliffener Kurvenverlauf (dicke Linie).

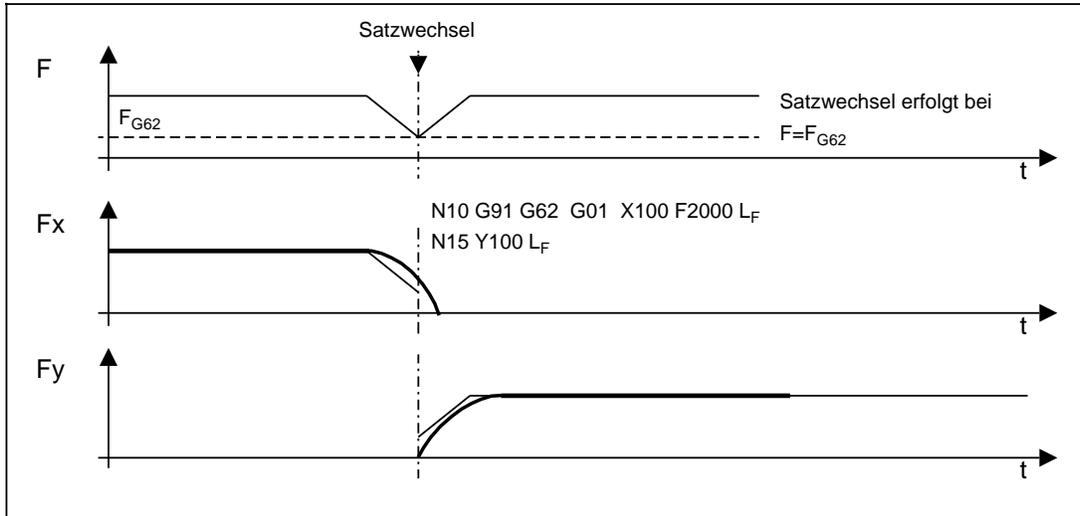


Genauhalt G60/G09

3.2.6.2 Bahnsteuerbetrieb G62/G64

Mit der Funktion G62 wird der **Vorschub F zum Satzende** hin auf eine im **NC-MD 3** festgelegte Reduziergeschwindigkeit **verringert**. G62 und G64 sind modal wirksam.

Bei G64 wird der Vorschub nicht reduziert.



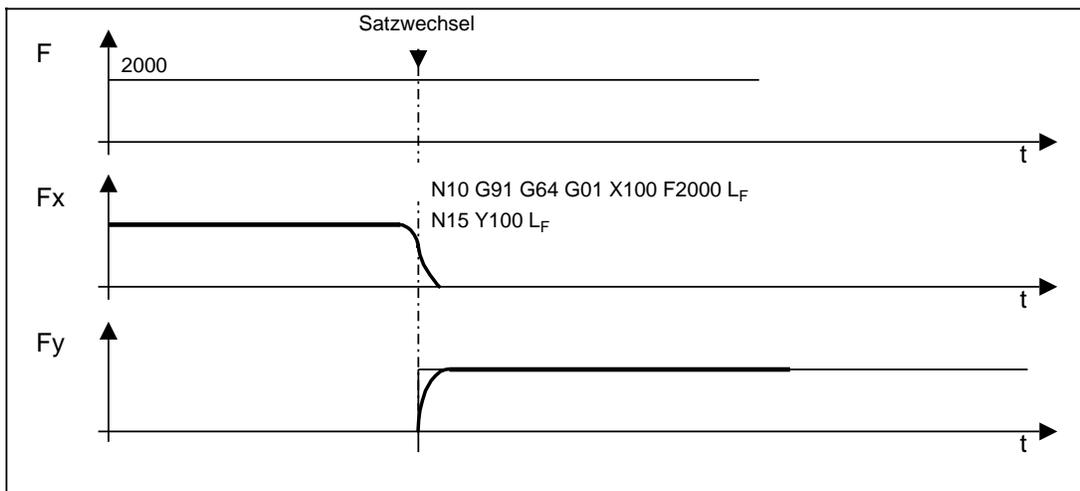
Bahnsteuerbetrieb mit G62

Anwendung:

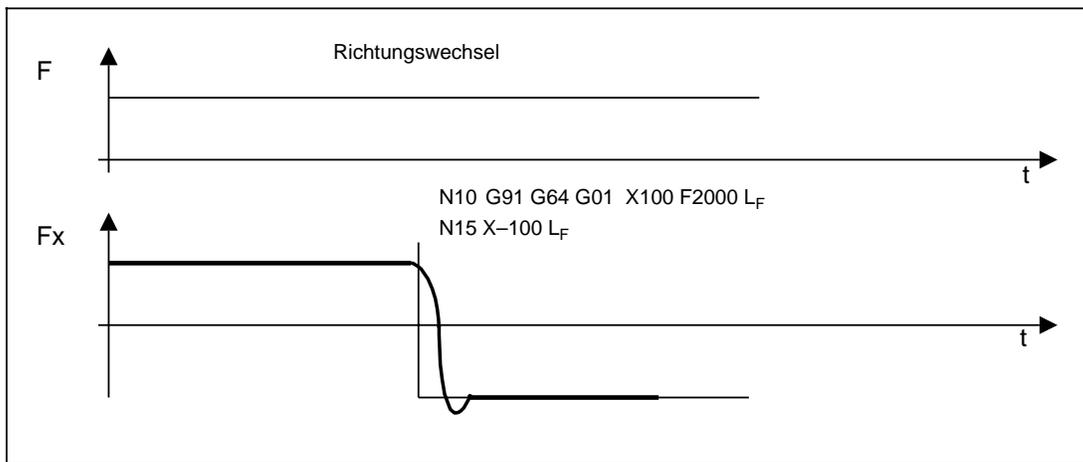
Bei der Holzverarbeitung darf beim Satzübergang der Vorschub nicht zu Null werden, da sonst Brandmarken am Werkstück entstehen würden.

Satzübergang ohne Geschwindigkeitsreduzierung G64

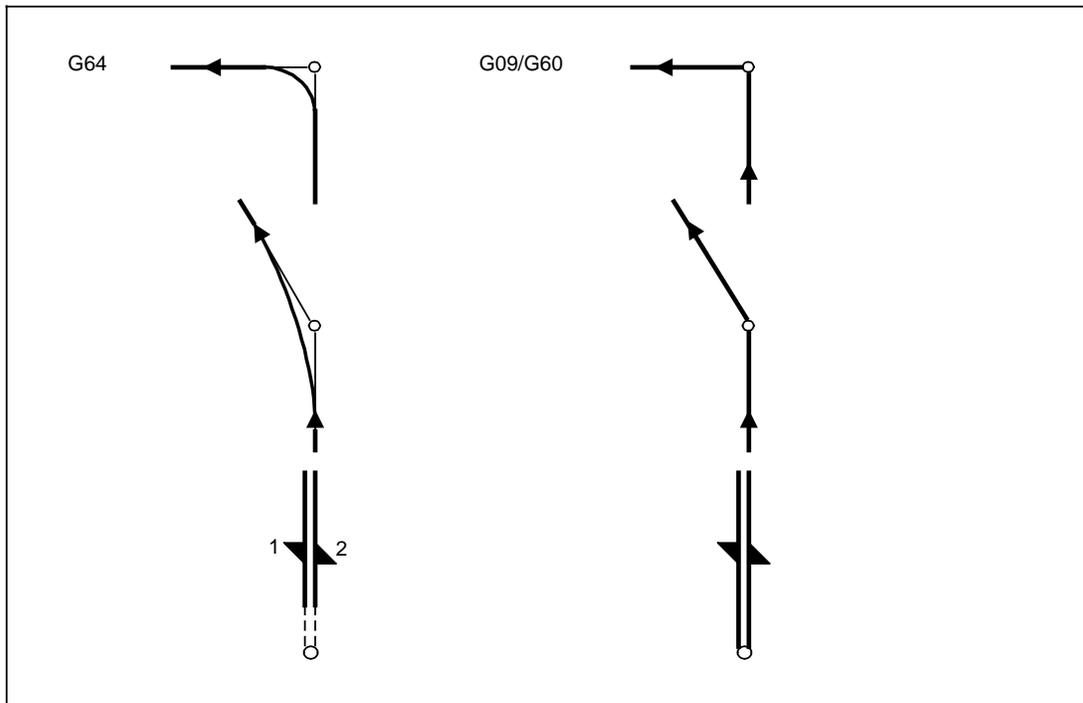
Die Wegbedingung G64 wird verwendet, wenn **kein Freischneiden bei Übergängen** von Satz zu Satz auftreten soll. Außerdem werden damit bei Änderung der Bewegungsrichtung die Übergänge verschliffen.



Bahnsteuerbetrieb mit G64 ohne Geschwindigkeitsreduzierung mit unterschiedlichen Achsen



Bahnsteuerbetrieb mit G64 und Richtungswechsel einer Achse



Richtungswechsel mit und ohne Geschwindigkeitsreduzierung

3.2.7 Verweilzeit G04

Die Verweilzeit wird unter der Adresse **X** oder **F** angegeben. 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:

N.. G04 X11.5 L_F

X11.5=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**. Verweilzeiten werden benötigt beim Freischneiden, evtl. bei Drehzahlwechsel und Maschinenschaltfunktionen (Lünette, Reitstock u.s.w.).

3.2.8 Weiches Anfahren und Verlassen der Kontur

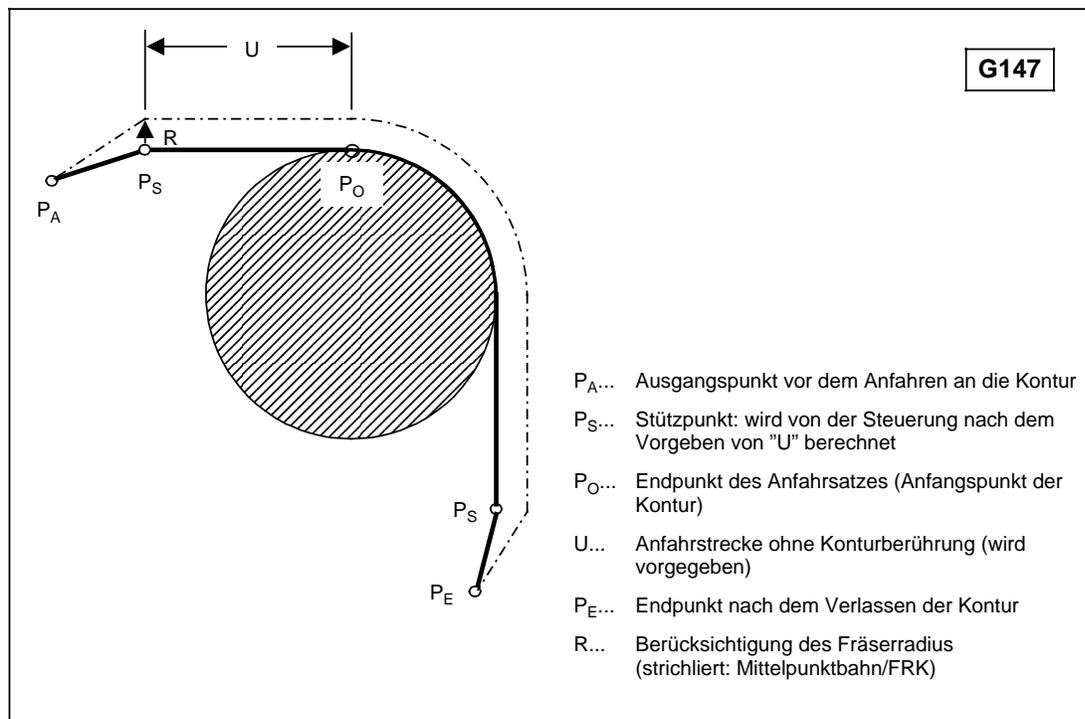
Um Schneidmarken zu vermeiden, wird eine Kontur **tangential** angefahren bzw. verlassen.

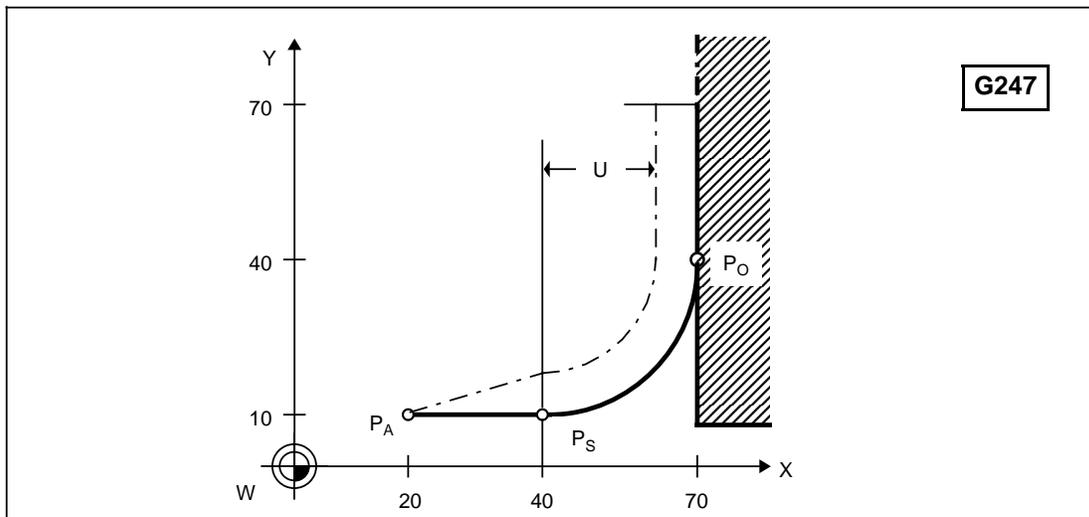
Weiches Anfahren und Verlassen einer Kontur kann **bei allen Werkzeugtypen** angewendet werden. Verrechnet werden "Radius" und "Verschleiß".

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

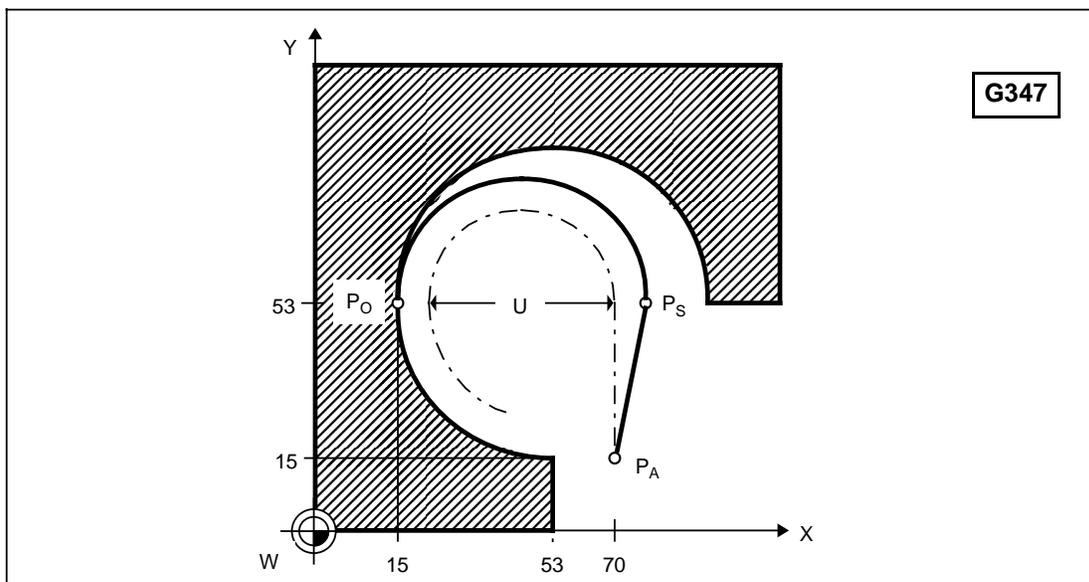
- G147** Anfahren linear
- G247** Anfahren im Viertelkreis
- G347** Anfahren im Halbkreis
- G148** Verlassen linear
- G248** Verlassen im Viertelkreis
- G348** Verlassen im Halbkreis
- G48** Verlassen der Kontur in der gleichen Weise, wie sie angefahren wurde

Beispiel: Gerade an Kreis



Beispiel: Anfahren im Viertelkreis an Geradenkontur

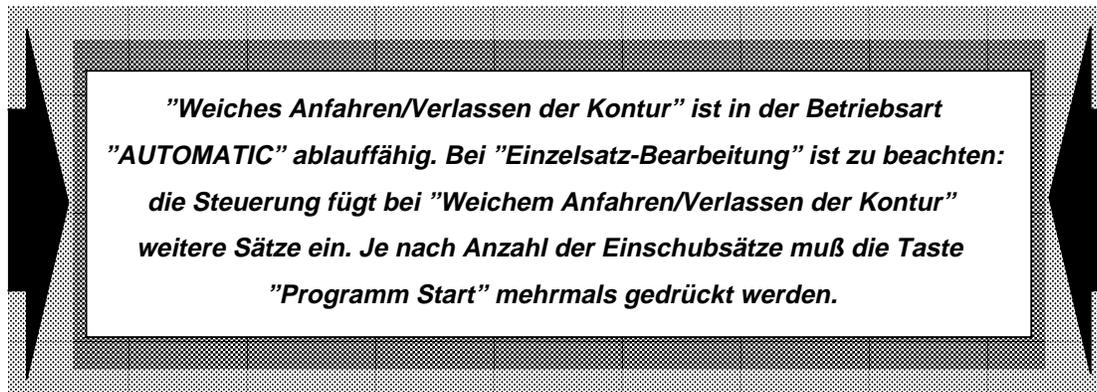
N05	G00	Y10	X20	L_F		im Eilgang auf P _A fahren
N10	G01	G41	Y10	X20	D1 F1000	Anwahl FRK links (G41), Werkzeugkorrekturanwahl (D1)
N15	G247	Y40	X70	U25	L_F	Anwahl "Anfahren im Viertelkreis" (G247)
N20	G01	Y70		L_F		Abfahren der Geradenkontur
N25	M30			L_F		Programmende

Beispiel: Anfahren im Halbkreis an Innenkreis

N05	G00	Y15	X70	L_F		im Eilgang auf P _A fahren
N10	G01	G41	Y15	X70	D2 F1000	Anwahl FRK links (G41), Werkzeugkorrekturanwahl (D2)
N15	G347	Y53	X15	U50	L_F	Anwahl "Anfahren im Halbkreis" (G347)
N20	G03	Y15	X53	I38	J0 L_F	Kontur abfahren
N25	G0	X70		L_F		im Eilgang auf P _A fahren
N30	M30			L_F		Programmende

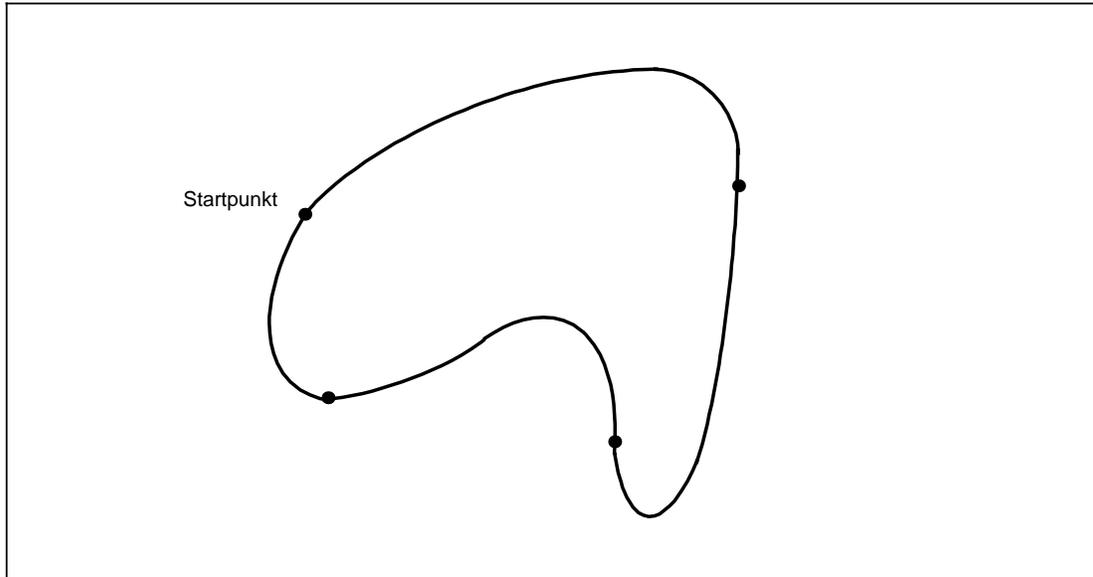
Kennzeichen der Programmierung

- Die Funktionen zum Anfahren bzw. Verlassen der Kontur sind satzweise wirksam.
- Im Anfahrtsatz sind anzugeben:
 - die Koordinaten des Anfangspunktes P_0 der Kontur und
 - der Wert von U (Anfahrstrecke ohne Konturberührung).
- Im Abfahrtsatz sind anzugeben:
 - die Koordinaten des Endpunktes P_E nach dem Verlassen der Kontur und
 - der Wert von U (Abfahrstrecke ohne Konturberührung).
- In einem Satz mit "G47" oder "G48" dürfen keine anderen Verfahrensbewegungen programmiert werden.
- Sowohl im Anfahrtsatz als auch im Abfahrtsatz können Hilfsfunktionen programmiert werden.
- Nach einem Anfahrtsatz oder vor einem Abfahrtsatz darf kein reiner Hilfsfunktionssatz programmiert werden.
- Bei der Kombination von Anfahr- und Abfahrtsätzen ist zu beachten, daß im Abfahrtsatz von der Steuerung "G40" (Abwahl der FRK) generiert wird: Vor jedem neuen Anfahrtsatz G41 programmieren!
- Bei satzweise zu erstellenden Teileprogrammen (TEACH IN/PLAYBACK) kann das fertige Teileprogramm nachträglich ergänzt werden.



3.2.9 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.

3.3 Referenzpunktfahren über Teileprogramm: G74 (nur GA3 SW2)

3.3.1 Veranlassung

Das Synchronisieren einer NC-Achse soll durch die Programmierung einer G-Funktion im Teileprogramm (PP) realisiert werden.

3.3.2 Funktionsbeschreibung

Mit der Funktion "Referenzpunktfahren über Teileprogramm" erhalten Sie 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 "Referenzpunktfahren über Teileprogramm" stellt eine Standardfunktion dar, die für Rund- und Linearachsen nutzbar ist.

3.3.3 Starten der Funktion

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

```
N <Nr> G74 <Achname> L_F
```

Erläuterungen:

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

Beispiel:

```
N1000 G74 C L_F
```

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

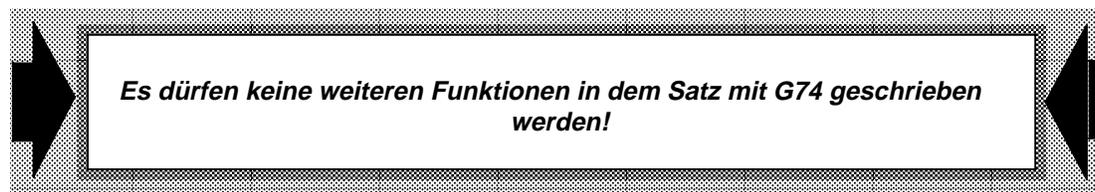
```
N <Nr> G74 Y1=L_F
```

Beispiel:

```
N1000 G74 Y1= L_F
```

Bemerkungen:

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



4 Schaltfunktionen, Hilfsfunktionen, Zusatzfunktionen, Sonderfunktionen 810N/820N

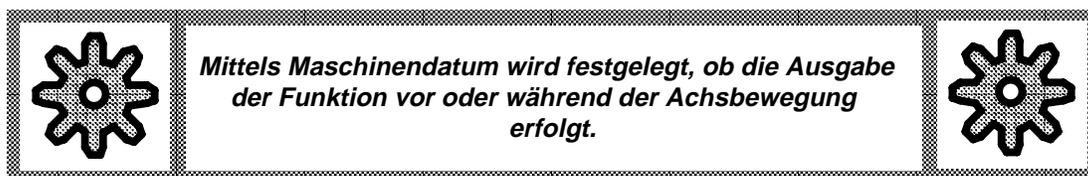
4.1 Allgemeines

M, S, T

Die Zusatzfunktionen enthalten vorwiegend technologische Angaben, soweit diese nicht in den dafür vorgesehenen Wörtern mit den Adreßbuchstaben F, S, T programmiert werden:

- Zusatzfunktion, Stanz-Nibbelfunktion M
- Spindeldrehzahl, Laserfunktionen S
- Werkzeugnummer, Laserfunktion T
- Stanz-Nibbelfunktion H

Es können maximal 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 Werkzeugmaschinenhersteller. 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.

Zusatzfunktion M

Die Zusatzfunktionen sind zum Teil in DIN 66 025 Teil 2, zum Teil vom Werkzeugmaschinenhersteller festgelegt.

M00 Programmierter Halt (unbedingt)

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

M01 Programmierter Halt (bedingt)

M01 wirkt wie M00, jedoch nur dann, wenn die Funktion "Bedingter Halt (M01) wirksam" über Softkey eingeschaltet wurde.

M02 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 Unterprogrammende

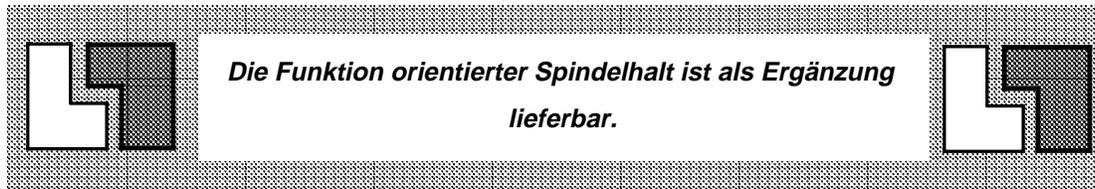
M17 wird im letzten Satz eines Unterprogramms geschrieben. Es kann in diesem Satz allein oder zusammen mit anderen Funktionen stehen. Der Aufruf eines Unterprogramms und M17 dürfen nicht im gleichen Satz stehen (Schachtelung).

M30 Programmende

M30 wirkt wie M02.

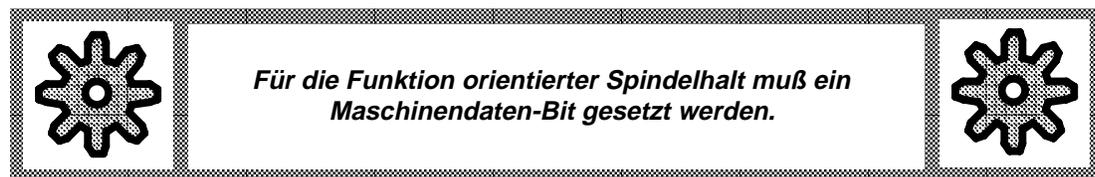
M03, M04, M05, M19 Hauptspindelsteuerung

(M19 nur mit Pulsgeber an der Hauptspindel.)



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

- M03 Spindeldrehrichtung rechts
- M04 Spindeldrehrichtung links
- M05 Spindelhalt, ohne Orientierung
- M19 Orientierter Spindelhalt



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.

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.

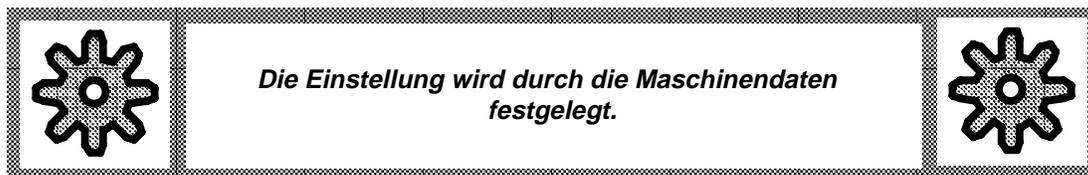
Hinweis:

Die nibbelspezifischen M-Funktionen sind ab Kapitel 4.2 beschrieben.

Spindelfunktion S

Bei Laserleistungssteuerung sind die Funktionen S0 bis S18 reserviert (siehe Kapitel 4.9.2)!
Unter der Adresse S kann wahlweise eingegeben werden:

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



Für das S-Wort gilt die erweiterte Adreßschreibweise mit Angabe der Spindelnummer, z. B.
S2=1000

S2 Angabe der Spindelnummer
1000 Spindeldrehzahl.

Werkzeugnummer T

Die Werkzeugnummer bestimmt das für einen Bearbeitungsabschnitt benötigte Werkzeug
T 1234 ...

T Adresse
1234 Werkzeugnummer (max. 4dekadig).

Bei Laserleistungssteuerung wird für die Ramp- und Impulszeit ebenfalls die Adresse T verwendet (siehe Kapitel 4.9.1).

4.2 Übersicht spezieller Funktionen für SINUMERIK 810N/820N

M 00 ... M 19	Schnelle M-Funktionen	Kapitel 4.3
M 20	Positionieren	Kapitel 4.4.2.1
M 21	Tangentialsteuerung aus	Kapitel 4.4.2.2
M 22/M 24	Nibbel	Kapitel 4.4.2.3
M 25	Stanzen	Kapitel 4.4.2.4
M 26	Rote Doppellinie zeichnen	Kapitel 12.3.2
M 28	Analoge Tangentialwinkelausg.	Kapitel 4.7
M 29	Pratzenposition aktualisieren	Kapitel 4.4.2.7
M 31 ... M 33	Laserleistungssteuerung E/F/T	Kapitel 4.9.1
M 34 ... M 36	Laserleistungssteuerung T	Kapitel 4.9.1
M 41	Zielposition im Bereich Prätze 1	Kapitel 12.2.6
M 42	Zielposition im Bereich Prätze 2	Kapitel 12.2.6
M 43	Zielposition im Bereich Prätze 3	Kapitel 12.2.6
M 44	Zielposition im Bereich Prätze 4	Kapitel 12.2.6
M 46	Signal für Stanze hochfahren	Kapitel 12.2.5

*) Drehzahl und Schnittgeschwindigkeit sind in der gleichen Eingabeform zu programmieren.
=0 ... 99, projektierbar über Maschinendatum

H	Selbsttätige Satzaufteilung	Kapitel 4.4.1.2
H1	Streckenvervielfältigung	Kapitel 4.4.1.3
E	Vorschub pro Hub	Kapitel 4.4.1.1
E	Laser Spannungswert	Kapitel 4.9.1
T	Laser Ramp-/Impulszeit	Kapitel 4.9.1
S00 ... S05	Laser Konstantenspannungen	Kapitel 4.9.2
S06 ... S09	Laser Bahnsteuerung	Kapitel 4.9.2
S10 ... S12	Laser Streckensteuerung	Kapitel 4.9.2
S13 ... S18	Laser Zeitsteuerung	Kapitel 4.9.2
X1=	Mini-Pattern	Kapitel 4.8
Y1=	Mini-Pattern	Kapitel 4.8

4.3 Schnelle M-Funktionen (Ein-/Ausschaltfunktionen)

Durch Programmieren von schnellen M-Funktionen können die hardwaremäßig vorhandenen Ausgänge direkt ein- bzw. ausgeschaltet werden.

Da beispielsweise das Öffnen des Shutters bei einer Lasersteuerung eine zeitkritische Funktion ist, muß die NC ein schnelles unverzögertes Signal ausgeben.

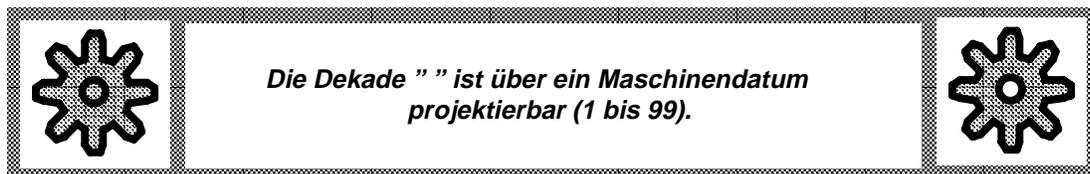
Manche Laser benötigen weitere schnelle Signale. Dafür stehen die Bits 0 ... 6 des für schnelle Ausgänge reservierten Bytes zur Verfügung. Sie werden über M-Funktionen geschaltet. Folgende M-Funktionen stehen zur Verfügung:

Einschaltfunktionen

M 00	Ausgangsbit 0	setzen
M 01	Ausgangsbit 1	setzen
:	:	:
M 06	Ausgangsbit 6	setzen
M 08	Ausgangsbit 0 ... 3	setzen
M 09	Ausgangsbit 4 ... 6	setzen

Ausschaltfunktionen

M 10	Ausgangsbit 0	löschen
M 11	Ausgangsbit 1	löschen
:	:	:
M 16	Ausgangsbit 6	löschen
M 18	Ausgangsbit 0 ... 3	löschen
M 19	Ausgangsbit 4 ... 6	löschen



Zur Anwendung dieser Funktion siehe Dokumentation des Maschinenherstellers. Eine ausführliche Beschreibung dieser Funktionen ist in der Inbetriebnahmeanleitung SINUMERIK 810N/820N enthalten.

4.4 Sonderfunktionen

4.4.1 Automatische Streckenaufteilung E-/H-/H1-Funktionen

Diese Funktionen generieren selbständig Teilsätze.

4.4.1.1 Programmierung mit dem E-Wert, Vorschub in mm/Hub bzw. inch/Hub

Die Auswertung der E-Funktion erfolgt nur in Kanal 1 und Kanal 3.

Vereinbarungen

- Die automatische Streckenaufteilung (E) teilt den programmierten X-Y-Verfahrweg entsprechend dem programmierten E-Vorschubweg in gleich große Teilstrecken auf.
- Diese Funktion wird nur in Verbindung mit Nibbel- und Stanzfunktionen programmiert (M 22/M 24/M 25).
- Ein E-Wert kann nicht gleichzeitig mit einem H- oder H1-Wert programmiert werden.

Programmierung

Programmierbereich:

- Metrisch 3.1 Stellen in mm/Hub inkrementell
- Zoll 2.2 Stellen in inch/Hub inkrementell

Diese Aufteilung ist wirksam bei:

- Geradeninterpolation
mit fester Geschwindigkeit (G00)
mit programmierter Geschwindigkeit (G01),
- Kreisinterpolation (G02/G03).

Die Teilstrecken werden von der Steuerung bei Bedarf abgerundet.

Die automatische Teilstreckenaufteilung ist

- selbsthaltend in Verbindung mit M 22, M 24,
- satzweise unterdrückt durch H-Wert in Verbindung mit M 22, M 24¹⁾ und
- satzweise wirksam in Verbindung mit M 20, M 25 bzw. jedem anderen Verfahrssatz.

Ist kein M 22, M 24, M 25 aktiv "ruckelnde" Achsbewegung ohne Stanzauslösung!

Wird ein E-Wert > der programmierten Teilstrecke eingegeben, wird zur nächsten programmierten Position gefahren. Beim Nibbeln mit Excenter wird der F-Wert, der bei der Programmierung von G01, G02, G03 notwendig ist, durch Eilgang G00 ersetzt.

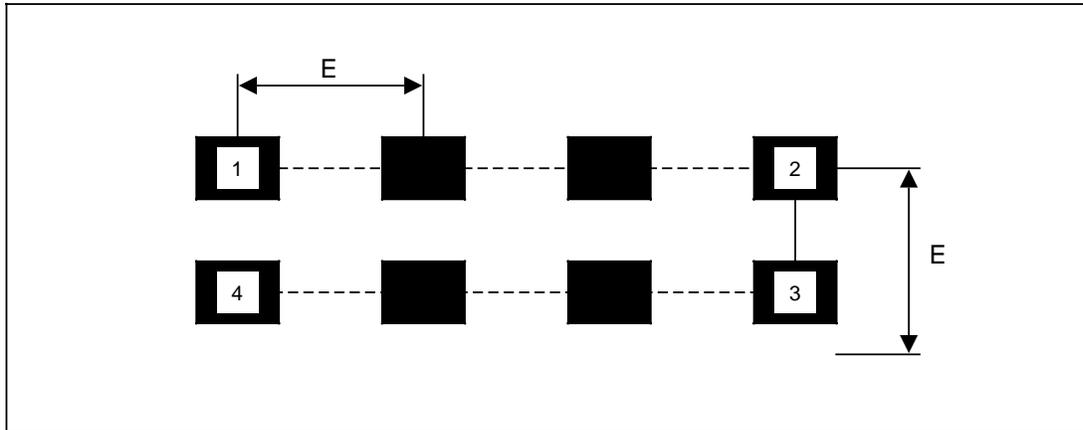
Bei Excenterbetrieb gelten folgende Einschränkungen:

Der programmierte E-Wert wird bei M 22/M 24 auf einen Maximalwert überwacht. Dieser Grenzwert ist in R-Parametern hinterlegt (siehe Kapitel 12.1.2).

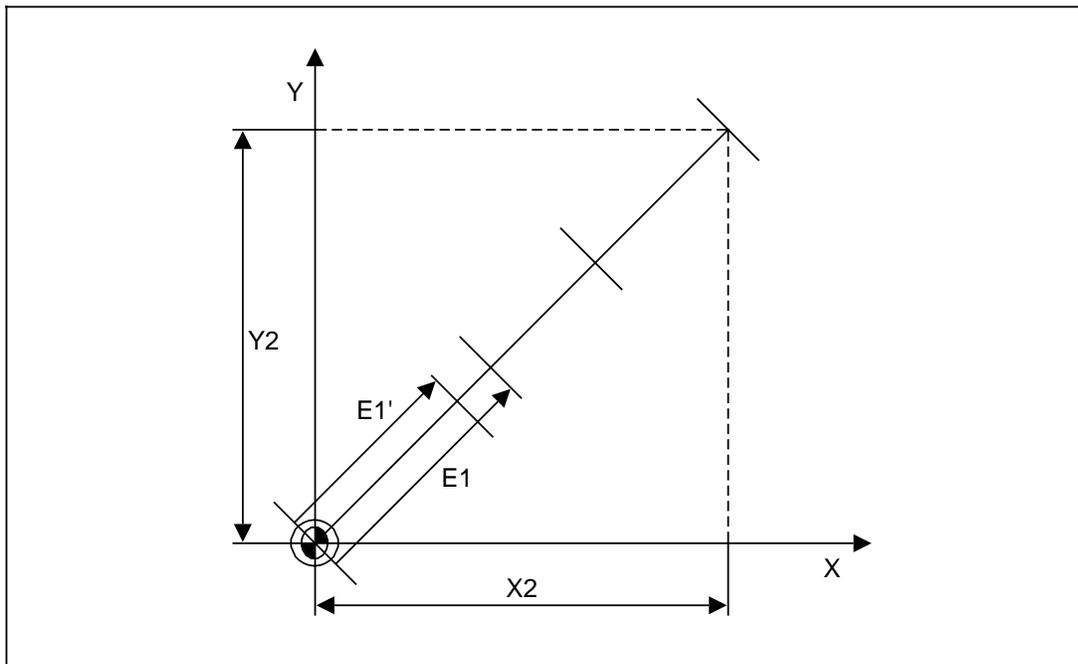
1) Ein zuvor programmierter E-Wert wird im ersten Folgesatz ohne H-Programmierung wieder aktiv.

Programmierbeispiel:

N1	G0	X	Y	M 20	Positionieren mit fester Geschwindigkeit
N2	X75	E25	M 22		Stanzn mit definierten Nibbelbeginn
N3	Y10				Nur reduzierter E-Wert, da Verfahrstrecke < E-Wert
N4	X0				Zurückfahren auf X0/Y10

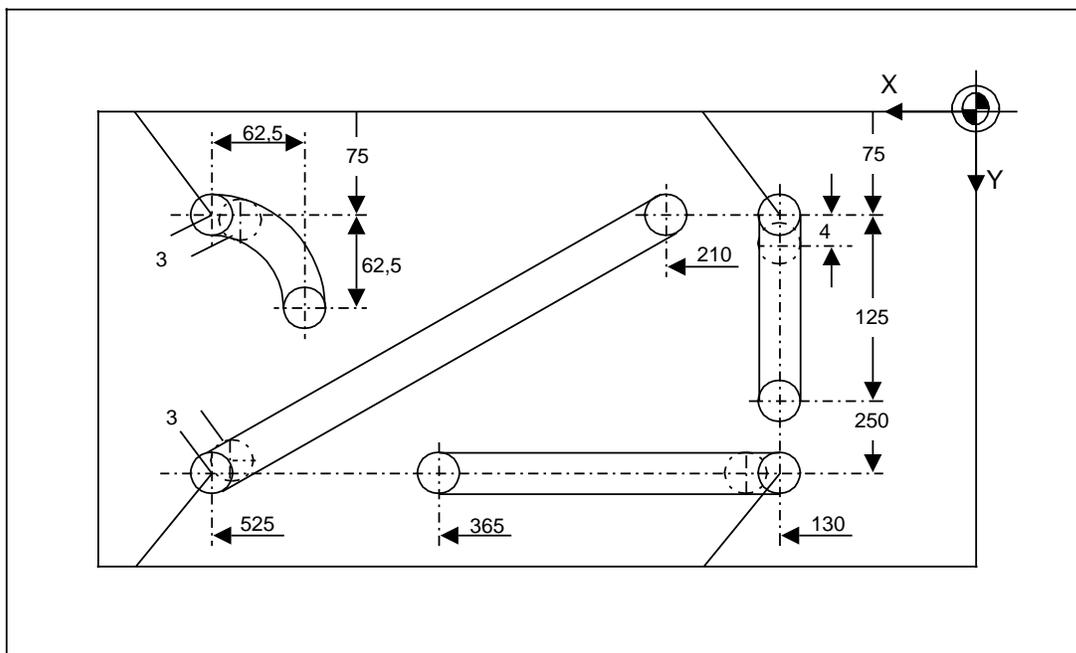


E0 bzw. Programmende (M02/M30) löscht den zuletzt programmierten Wert. Die Anzahl der Teilsätze ist begrenzt (65535).



Streckenaufteilung

- X2/Y2 Programmierter Verfahrweg (Nibbel oder Stanzsatz)
- E1 Programmierter Vorschubweg
- E1' Automatisch abgerundeter Vorschubweg

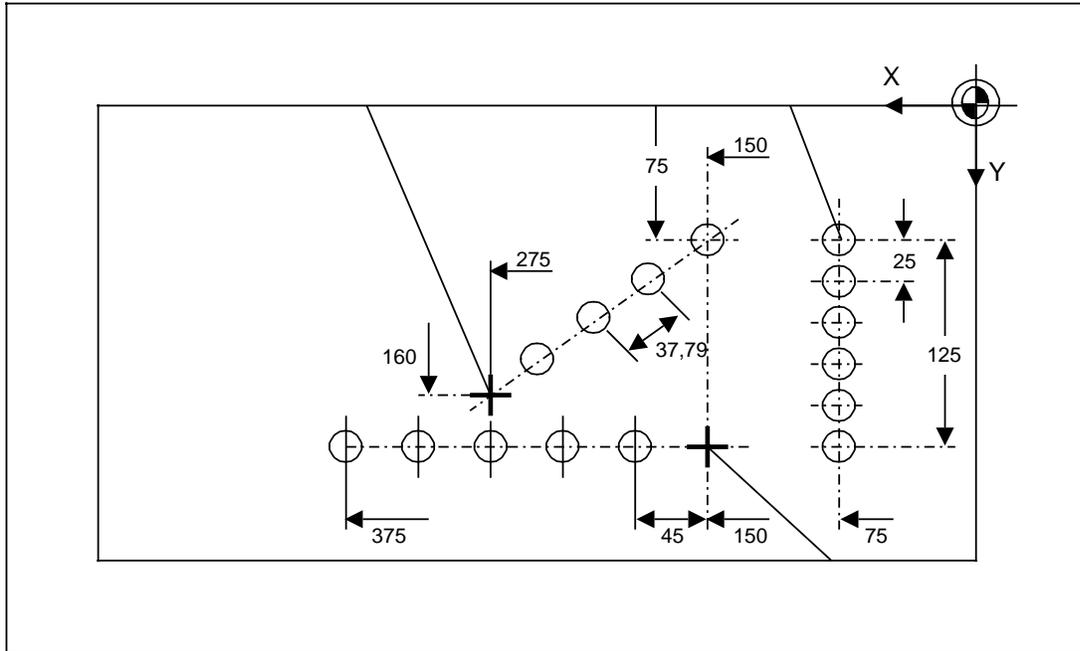
Anwendungsbeispiel: E-Wert beim Nibbeln

Werkstück

Programm-Ausschnitt

N100	G90	X130	Y75	F60	M 20	L _F	Positionieren auf Startpunkt der senkrechten Nibbelstrecken
N110	G91	Y125	E4	M 22	L _F		Endpunktkoordinaten (inkrementell), Vorschubwert: 4 mm, Nibbeln aktivieren
N120	G90	Y250	M 20	L _F			Absolute Vermaßung, Positionieren auf Startpunkt der waagerechten Nibbelstrecke
N130	X365	M 22	L _F				Endpunktkoordinaten, Vorschubwert: 4 mm, Nibbeln aktivieren
N140	X525	M 20	L _F				Positionieren auf Startpunkt der schrägen Nibbelstrecke
N150	X210	Y75	E3	M 22	L _F		Endpunktkoordinaten, Vorschubwert: 3 mm, Nibbeln aktivieren
N160	X525	M 20	L _F				Positionieren auf Startpunkt der Nibbelstrecke auf Teilkreisbahn
N170	G02	G91	X-62.5	Y62.5	I J62.5	E3	Inkrementelle Kreisinterpolation mit Interpolationsparametern, Nibbeln aktivieren
N180	G00	G90	Y300	M 20	L _F		Positionieren

Anwendungsbeispiel: E-Wert beim Stanzen



Werkstück

Programm-Ausschnitt:

N100 G90 X75 Y75 F60 M 25 L_F

Positionieren auf Startpunkt der senkrechten Lochreihe,
 Einzelloch stanzen

N110 G91 Y125 E25 M 25 L_F

Endpunktkoordinaten (inkrementell),
 Vorschubwert: 25 mm,
 Stanzen aktivieren

N120 G90 X150 M 20 L_F

Absolute Vermaßung,
 Positionieren auf Startpunkt der waagrechten Lochreihe

N130 X375 E45 M 25 L_F

Endpunktkoordinaten,
 Vorschubwert: 45 mm
 Stanzen aktivieren

N140 X275 Y160 M 20 L_F

Positionieren auf Startpunkt der schrägen Lochreihe

N150 X150 Y75 E40 M 25 L_F

Endpunktkoordinaten,
 Programmierter Vorschubwert: 40 mm,
 Berechneter Vorschubwert: 37,79 mm

N160 G00 Y300 M 20 L_F

Positionieren

4.4.1.2 Programmierung mit dem H-Wert, selbsttätige Streckenaufteilung

Die Auswertung der H-Funktion erfolgt nur in Kanal 1 und Kanal 3.

Vereinbarungen

- Die Streckenaufteilung auf H-Teilstrecken wird in Verbindung mit Nibbel- und Stanzsätzen programmiert (M 22, M 24, M 25).
- Bei Exzentersteuerung nur in Verbindung mit M 25 möglich.
- Ein H-Wert kann nicht gleichzeitig mit einem E-Wert oder H1-Wert programmiert werden.

Programmierung

Programmierbereich: 1 bis 255

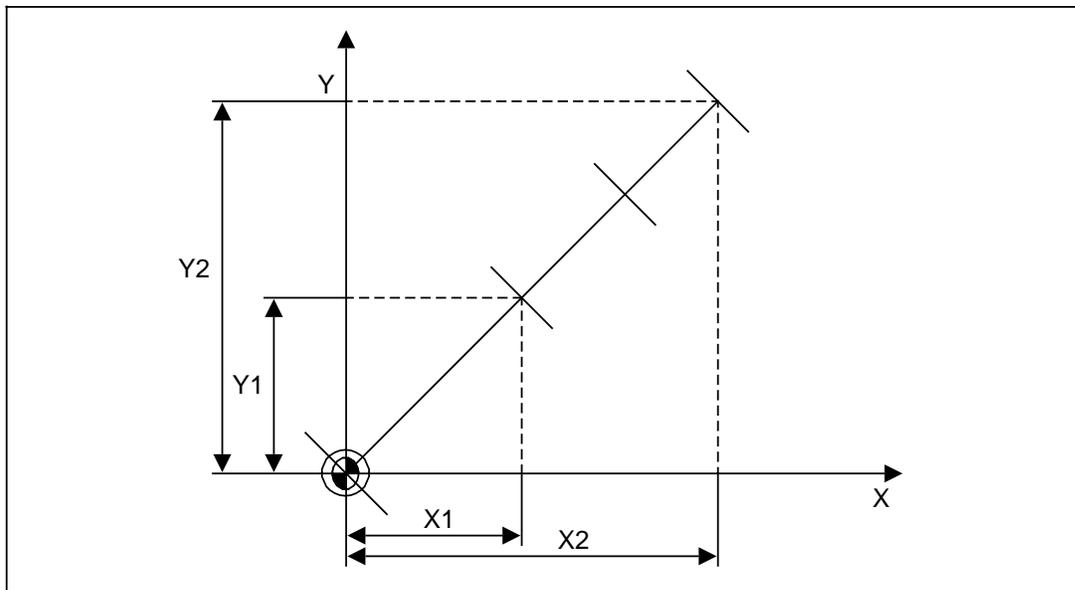
Der programmierte X-Y-Verfahrweg wird in eine unter H programmierte Anzahl gleicher Teilstrecken aufgeteilt. Diese Aufteilung ist wirksam bei:

- Geradeninterpolation
 - mit fester Geschwindigkeit (G00)
 - mit programmierter Geschwindigkeit (G01)
- Kreisinterpolation (G02/G03)

Die programmierte H-Stanzaufteilung ist satzweise wirksam. Ein zuvor programmierter E-Wert wird im ersten nachfolgenden Satz ohne H-Programmierung wieder aktiv (siehe Programmbeispiel 4.4.2.6, Beispiel (3)).

Ist kein M 22, M 24, M 25 aktiv "ruckelnde" Achsbewegung ohne Stanzauslösung!

Der F-Wert, der bei der Programmierung von G01, G02, G03 notwendig ist, wird bei Exzentermaschinen durch Eilgang (G00) ersetzt.



Streckenaufteilung (H1=3 programmiert)

X2/Y2 Programmierte Verfahrstrecke
X1 Automatisch errechnete Teilstrecke in X
Y1 Automatisch errechnete Teilstrecke in Y

4.4.1.3 Programmierung mit dem H1-Wert, Vervielfältigung einer programmierten Teilstrecke

Die Auswertung der H1-Funktion erfolgt nur in Kanal 1 und Kanal 3.

Vereinbarungen

- Die Vervielfältigung der programmierten Teilstrecke zu einer Gesamtstrecke wird in Verbindung mit Nibbel- und Stanzsätzen programmiert (M 22, M 24, M 25).
- Ein H1-Wert kann nicht gleichzeitig mit einem E-Wert oder H-Wert programmiert werden.
- Bei Exzentersteuerung nur in Verbindung mit M 25 möglich.

Programmierung

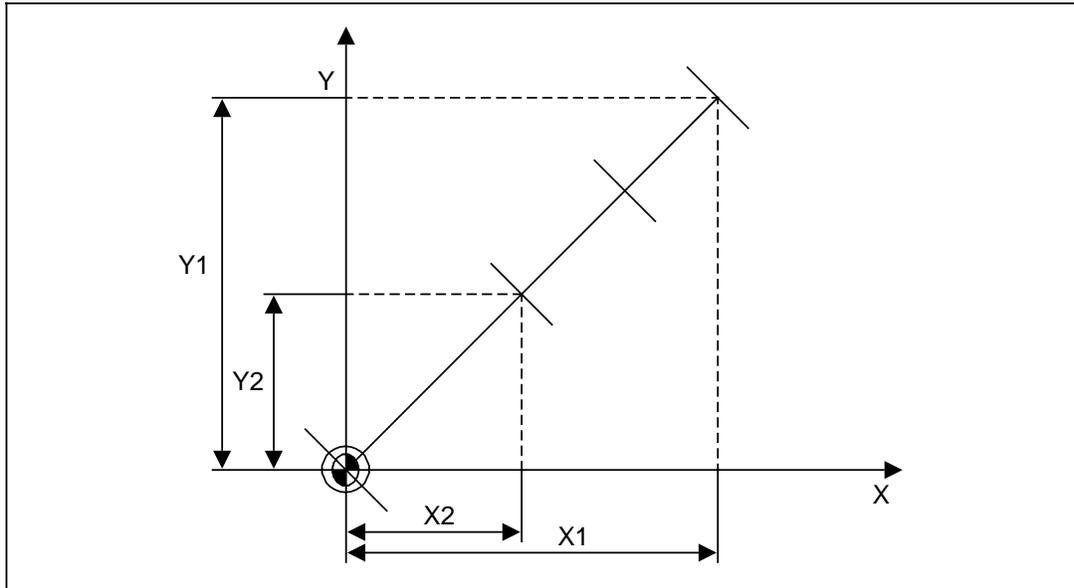
Programmierbereich: 1 bis 255

H1 darf nur im Absolutmaß programmiert werden. Andernfalls wird eine Fehlermeldung ausgegeben.

Der programmierte X-Y-Verfahrweg wird mit einer unter H1 programmierten Anzahl wiederholt. Diese Vervielfältigung ist wirksam bei:

- Geradeninterpolation mit fester Geschwindigkeit (G00),
- Geradeninterpolation mit programmierter Geschwindigkeit (G01) (wird zu G00) und
- nicht bei Kreisinterpolation (G02, G03)!

Die H1-Programmierung ist satzweise wirksam. Ist kein M 22, M 24, M 25 aktiv "ruckelnde" Achsbewegung ohne Stanzauslösung!



Streckenaufteilung (H1=3 programmiert)

- X2/Y2 Programmierte Teilstrecke
- X1 Automatisch errechnete Verfahrstrecke in X
- Y1 Automatisch errechnete Verfahrstrecke in Y

4.4.2 Nibbel- und Stanzfunktionen (Spezielle M-Funktionen)

4.4.2.1 M 20 positionieren

Zum Abschalten des Stanz- oder Nibbelbetriebes sowie der Tangentialsteuerung (außer M 21 ist wirksam) wird die Funktion M 20 verwendet.

M 20 ist Resetzustand.

Wird M 20 gemeinsam mit einem Wegbefehl im gleichen Satz programmiert, fährt die Maschine die programmierte Position an, ohne einen Stanz- oder Nibbelhub auszulösen. M 20 bewirkt Löschen der Funktionen M 22/M 24 (Nibbeln) oder M 25 (Stanzen).

Programmierbeispiel:

```

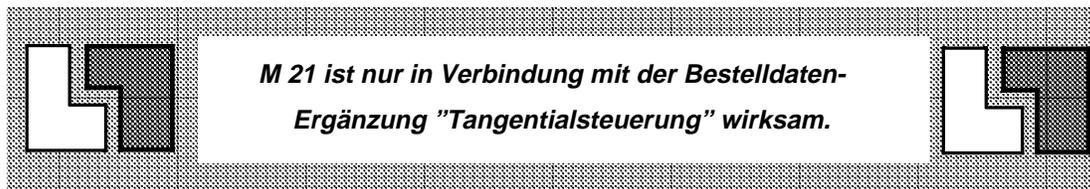
:
:
N20 G90 X100 E20 M 22 L_F           Nibbeln aktivieren mit M 22
                                     Nibbelvorschubwert: 20 mm

N25 X50 M 20 L_F                   Nibbeln abwählen/Positionieren mit M 20
:
:

```

4.4.2.2 M 21 Tangentialsteuerung AUS

Diese Funktion schaltet die selbsthaltende Tangentialsteuerung aus. Die Abschaltfunktion wird **am Ende** des Satzes wirksam, in dem M 21 programmiert ist (selbsthaltende Tangentialsteuerung siehe 4.6.2.1).



4.4.2.3 M 22/M 24 Nibbeln

Diese Funktion wird im gleichen Satz mit dem ersten Nibbelschritt programmiert. Der Nibbelschritt (Vorschub in mm/Hub) wird unter der Adresse E oder die Anzahl der Stanzlöcher unter H bzw. H1 angegeben. Die Schnittaufteilung ist dabei automatisch wirksam. Der erste Nibbelhub wird in der Anfangsposition des ersten Nibbelsatzes ausgeführt, also vor dem Verfahren des 1. Nibbelschrittes.

Ausnahme: definierter Nibbelbeginn

”**Definierter Nibbelbeginn**” bedeutet, daß grundsätzlich an der Anfangsposition des ersten Nibbelsatzes der erste Hub ausgeführt wird, außer wenn an dieser Position schon ein Loch gestanzt wurde und danach kein M 20 programmiert war (siehe Beispiel 1 und 2).

M 22/M 24 reaktiviert die E-Funktion mit dem zuletzt programmierten E-Wert.

=0 ... 99, projektierbar über Maschinendatum

Ausnahme: Im gleichen Satz ist eine H- oder eine H1-Funktion programmiert.

Ist kein E-Wert vorhanden, erscheint Alarm "Kein E-Wert vorhanden" (Nr. 2232).

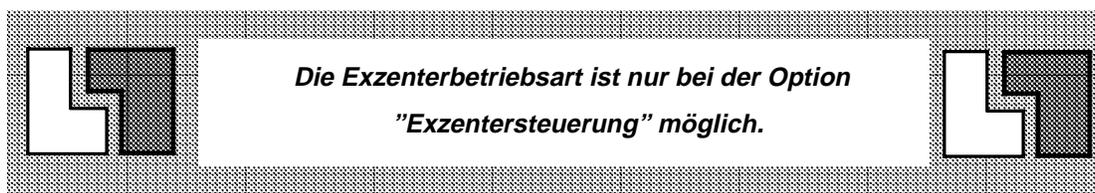
M 22/M 24 ist selbsthaltend und wird durch M 20, M 25 oder M-Funktionen der 2. M-Gruppe gelöscht.

- M 22 bedeutet bei Exzentersteuerung niedrige Exzenterdrehzahl.
- M 24 bedeutet bei Exzentersteuerung hohe Exzenterdrehzahl, siehe Anhang Kapitel 12.1.

Ohne Exzentersteuerung haben M 22/M 24 die gleiche Funktion.

Bei Exzenterbetrieb gelten folgende Einschränkungen:

- M 22/M 24 ist nicht mit H-Wert programmierbar.
- Der programmierte E-Wert wird auf einen Maximalwert überwacht. Er ist im R-Parameter R247 für die niedere Exzentergeschwindigkeit und in R248 für die höhere Exzentergeschwindigkeit hinterlegt (siehe Kapitel 12.1.2.1).



Programmierbeispiel:

- •
•
N80 X100 E5 M 22 L_F Nibbeln aktivieren mit M 22
Die Schnittaufteilung ist wirksam.
Nibbelvorschubwert: 5 mm
Die Lochabstände sind kleiner oder gleich E.
- N90 Y200 Y100 E1.5 M 24 L_F Nibbeln aktivieren mit M 24.
Die Schnittaufteilung ist wirksam.
Nibbelvorschubwert: 1,5 mm
Die Lochabstände sind kleiner oder gleich E.
- •

=0 ... 99, projektierbar über Maschinendatum

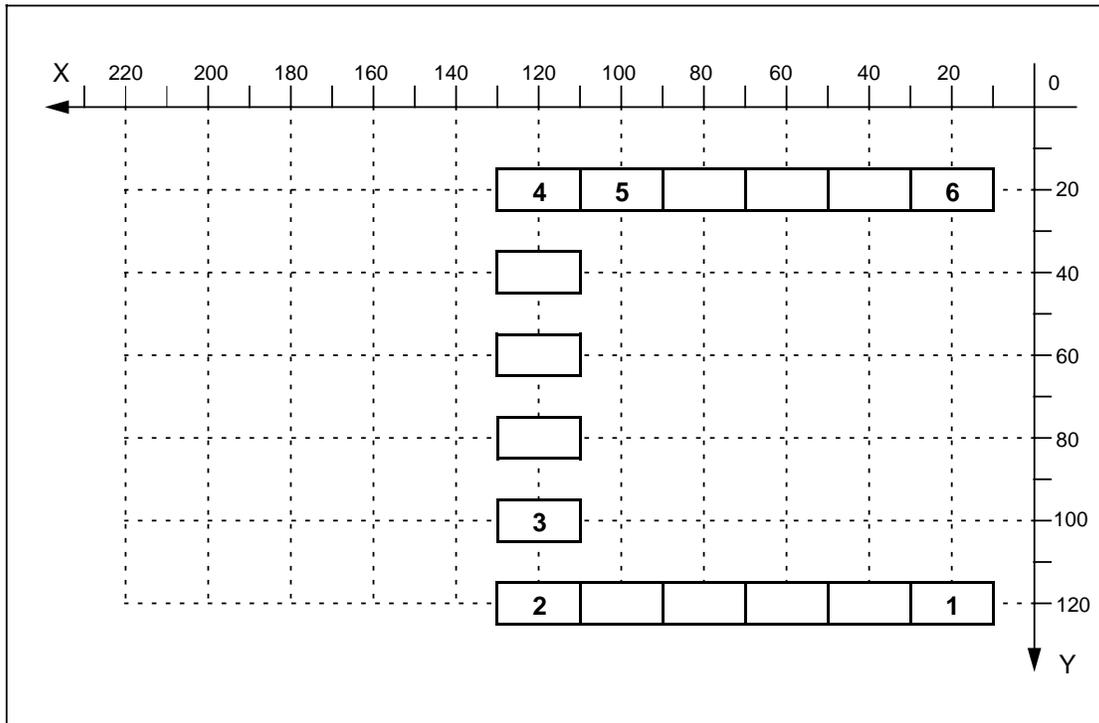
Beispiele zum definierten Nibbelbeginn**1. Beispiel:**%3 L_FN10 G0 X20 Y120 L_FN20 X120 E20 M 22 L_FN30 Y20 L_FN40 X20 L_FN50 M 20 L_FN60 M2 L_F

Position 1 wird angefahren

Definierter Nibbelbeginn, erster Hub auf "1", letzter Hub auf "2"

Definierter Nibbelbeginn, erster Hub auf "3", letzter Hub auf "4"

Definierter Nibbelbeginn, erster Hub auf "5", letzter Hub auf "6"



=0 ... 99, projektierbar über Maschinendatum

4.4.2.4 M 25 Stanzen

Ist in einem Satz M 25 programmiert, wird nach Erreichen der programmierten Position ein einzelner Stanzhub ausgelöst.

Bei E- oder H-Programmierung wird ein Stanzhub am Ende jeder Teilstrecke ausgeführt.

Am Startpunkt eines Satzes wird kein Hub ausgelöst (also kein definierter Nibbelbeginn wie bei M 22/M 24). Ausnahme: Start- und Zielpunkt sind identisch.

M 25 ist selbsthaltend und wird durch M 20/M 22/M 24 oder M-Funktionen der 2. M-Gruppe gelöscht.

Programmierbeispiel:

```

:
:
N50 X100 Y200 F60 M 25 L_F           Stanzhub in Satzendposition.
:
:
N110 X250 Y250 H5 M 25 L_F          Verfahrenstrecke wird in 5 gleiche Teilstrecken
:                                     aufgeteilt, 4 Hübe.
:
:
N200 X300 Y400 E10 M 25 L_F        Schnittaufteilung ist wirksam.
:                                     Vorschubwert: 10 mm.
:                                     Die Lochabstände sind kleiner oder gleich E.
:
:

```

Hinweis:

M 25 hat eine geänderte Bedeutung bei Exzentermaschinen mit "automatischer Stanz-/Nibbelumschaltung", siehe Anhang "Exzenter" Kapitel 12.1.3.

Bei Benutzung von Mitschleppachsen beachten Sie bitte Kapitel 4.5.1.

=0 ... 99, projektierbar über Maschinendatum

4.4.2.5 Zuordnung von E- und H-Programmierung zu den M-Funktionen

Bedeutung:

E Vorschub in mm/Hub

H Anzahl der Teilstrecken

Wenn in einem Satz	zu verfahrenende Strecke >0 mm			Achse(n) programmiert, aber zu verfahrenende Strecke =0 mm	
	E programmiert	H programmiert	nur Strecke programmiert	E programmiert	H programmiert
M 20 aktiv	Ruckeln auf die programmierte Endposition	Ruckeln auf die programmierte Endposition	Positionieren	keine Auswirkung	keine Auswirkung
M 22 oder M 24 aktiv	Stanzen von Löchern im Abstand E. Anfangsposition wird gelocht. *)	Stanzen von H+1-Löchern. Anfangsposition wird gelocht. *) **)	Stanzen von Löchern. Als E-Wert wird der zuletzt programmierte genommen.	keine Auswirkung	keine Auswirkung **)
M 25 aktiv	Stanzen von Löchern im Abstand E. Anfangsposition wird nicht gelocht	Stanzen von H Löchern. Anfangsposition wird nicht gelocht.	Stanzen von einem Loch in der programmierten Endposition.	Stanzen von einem Loch	Stanzen von H Löchern auf der Stelle.

Zuordnung von E- und H-Programmierung zu den M-Funktionen

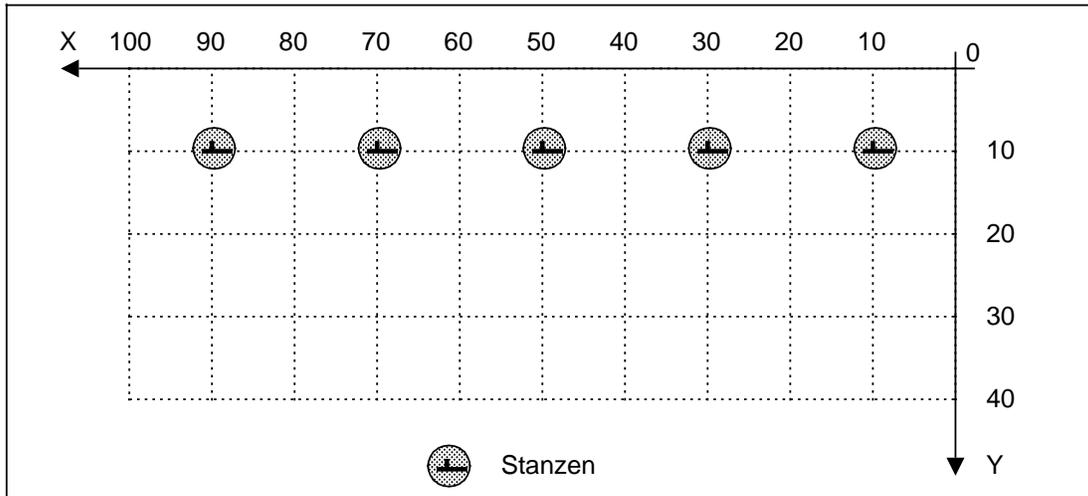
*) Die Ausgangsposition wird nicht gestanzt, wenn diese mit gleicher Werkzeugachsenstellung bereits gestanzt ist (definierter Nibbelbeginn).

**) Diese Kombination ist bei Exzentersteuerung nicht möglich.

4.4.2.6 Programmierbeispiele

(1) Beispiel für definierten Nibbelbeginn:

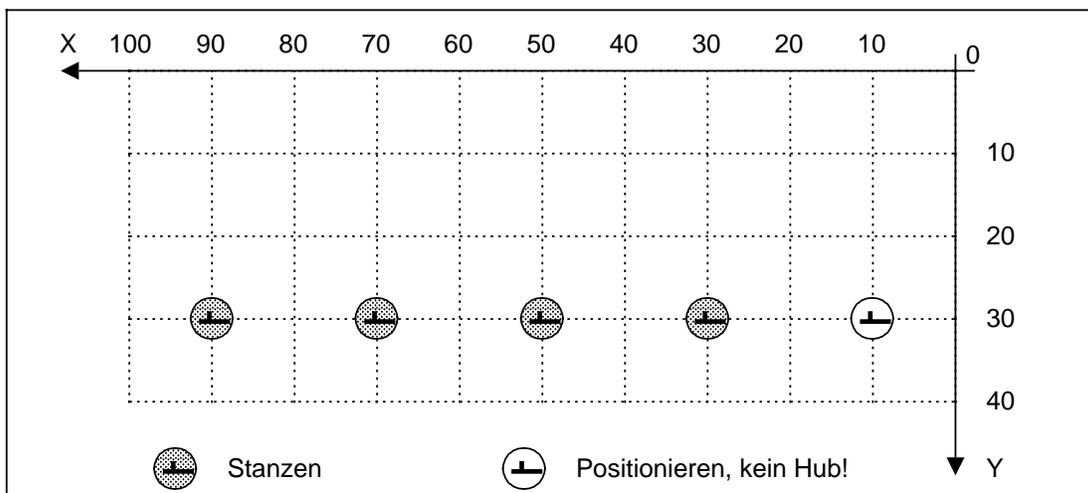
```
%5 L_F
N5 G0 X10 Y10 L_F      Positionieren
N10 X90 E20 M 22 L_F   Definierter Nibbelbeginn, 5 Stanzauslösungen
N20 M 20 L_F
N30 M2 L_F
```



Beispiel für definierten Nibbelbeginn

(2) Beispiel für ohne definierten Nibbelbeginn:

```
%6 L_F
N5 G0 X10 Y30 L_F      Positionieren
N10 X90 E20 M 25 L_F   Kein definierter Nibbelbeginn, 4 Stanzauslösungen
N20 M 20 L_F
N30 M2 L_F
```



Beispiel ohne definierten Nibbelbeginn

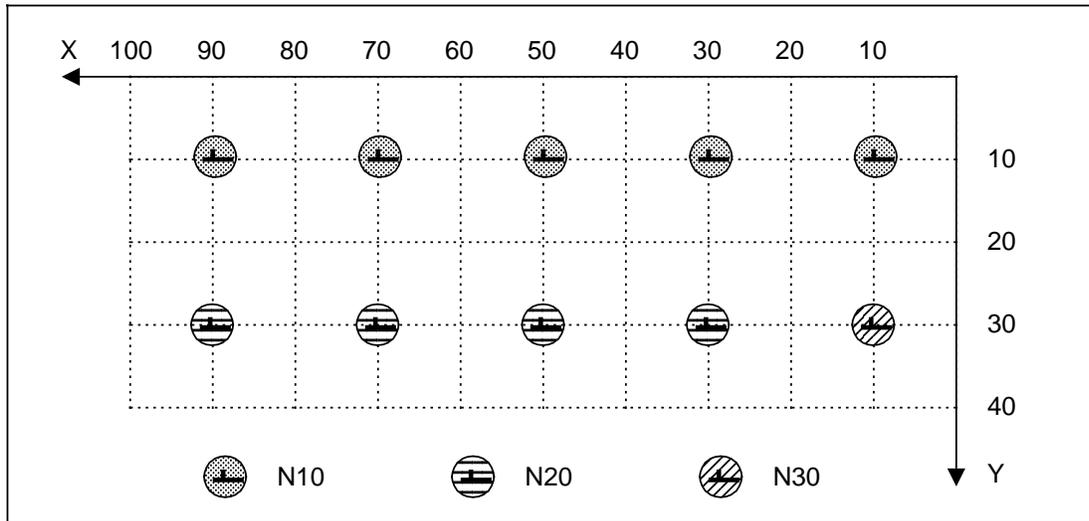
=0 ... 99, projektierbar über Maschinendatum

(3) Beispiel für **definierten Nibbelbeginn** zu nachfolgendem Bild:

%7 L _F	Dieses Beispiel gilt nicht für Exzenterstanzen!
N5 G0 X10 Y10 L _F	Positionieren
N10 X90 E20 M 22 L _F	Definierter Nibbelbeginn, 5 Stanzauslösungen.
N20 X10 Y30 H1 L _F	Am Ende der Strecke eine Stanzauslösung, M 22 ist selbsthaltend
N30 X90 L _F	4 Stanzauslösungen mit Abstand E20 und M 22
N40 M 20 L _F	(E20 aus Satz N10 wirksam.)
N50 M2 L _F	

(4) Beispiel für **definierten Nibbelbeginn** zu nachfolgendem Bild:

%8 L _F	Positionieren
N5 G0 X10 Y10 L _F	Definierter Nibbelbeginn, 5 Stanzauslösungen
N10 X90 H4 M 22 L _F	Am Ende der Strecke eine Stanzauslösung, M 22 ist selbsthaltend
N20 X10 Y30 M 25 L _F	4 Stanzauslösungen mit M 22
N30 X90 H4 M 22 L _F	
N40 M 20 L _F	
N50 M2 L _F	



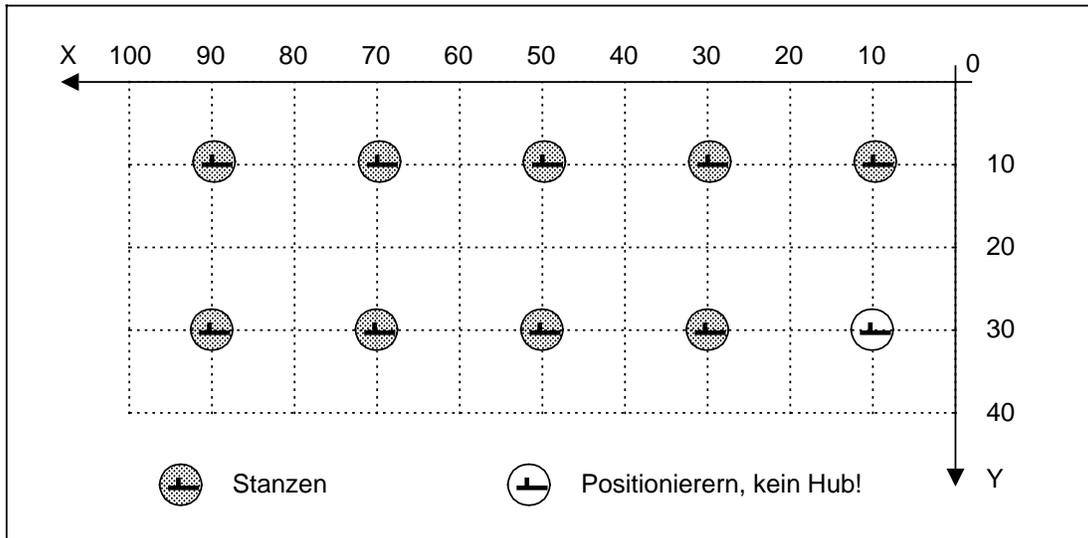
Beispiel für definierten Nibbelbeginn

(5) Beispiele für E-Programmierung **ohne definierten Nibbelbeginn** zu nachfolgendem Bild:

```
%9 L_F
N5 G0 X10 Y30 L_F      Positionieren
N10 X90 E20 M 25 L_F   Kein definierter Nibbelbeginn, 4 Stanzauslösungen
N15 Y10 L_F           Am Ende der Strecke eine Stanzauslösung
N20 X10 E20 L_F       4 Stanzauslösungen mit Abstand E20, M 25 aus N10 noch aktiv
N25 M 20 L_F
N30 M2 L_F
```

(6) Beispiele für H-Programmierung **ohne definierten Nibbelbeginn** zu nachfolgendem Bild:

```
%10 L_F
N5 G0 X10 Y30 L_F      Positionieren
N10 X90 H4 M 25 L_F   Kein definierter Nibbelbeginn, 4 Stanzauslösungen
N15 Y10 L_F           Am Ende der Strecke eine Stanzauslösung
N20 X10 H4 L_F       4 Stanzauslösungen
N25 M 20 L_F
N30 M2 L_F
```



Beispiele für E-/H-Programmierung ohne definierten Nibbelbeginn

=0 ... 99, projektiert über Maschinendatum

4.4.2.7 M 29 Pratzenpositionen aktualisieren

Die Eingabewerte der Pratzenpositionen und -maße werden das erste Mal mit NC-Start aktualisiert. Dabei werden nur die Inhalte von R241 bis R246 berücksichtigt. Wird eine Werkzeugnummer (T-Nummer) programmiert, werden die Inhalte von R241 bis R246 und der Wert aus dem Werkzeugkorrekturspeicher berücksichtigt. Während des Ablaufs eines NC-Programms können die Werte auf Anforderung erneut aktualisiert werden:

M 29 Inhalte von R241 ... R246 und L1 erneut anfordern.

Der zuletzt programmierte werkzeugspezifische Schutzbereich wird mit eingerechnet.

Bevor M 29 programmiert wird, müssen zuerst die entsprechenden R-Parameter neu belegt werden.

Einschränkung:

In dem Satz mit M 29 dürfen **keine Achsen** programmiert sein. Anwendungsmöglichkeit für M 29:

- bei verschiebbaren Pratzen.

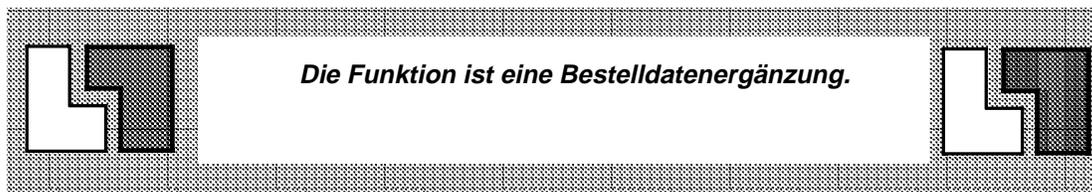
Bedeutung der Parameter

R241	Position der	1. Prätze	auf der X-Schiene
R242	"	2.	"
R243	"	3.	"
R244	"	4.	"
R245	Abmessung der Pratzen	in X	
R246	"		in Y

Die ausführliche Beschreibung dieser R-Parameter und ihrer Funktion erfolgt im Anhang, Kapitel 12.2.

=0 ... 99, projektierbar über Maschinendatum

4.4.3 Programmierung der Pulsauswahl: schnelle T-Nr.-Ausgabe (nur GA3 SW2)



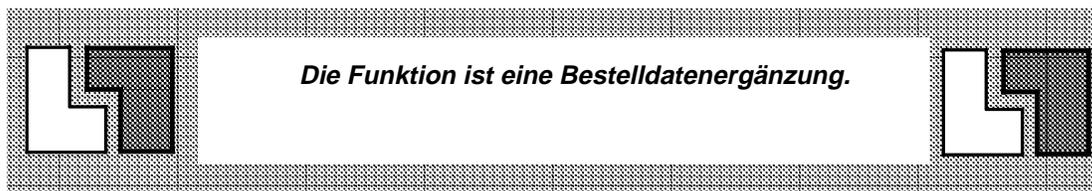
Das schnelle Ausgangsbyte (AB) zur Pulsauswahl auf der gemischten E/A-Baugruppe bzw. auf der zentralen Ausgabebaugruppe kann direkt aus dem Teileprogramm heraus programmiert werden.

Programmiert wird die Pulsauswahl durch eine T-Nr. Je nach Vorgabe (siehe Anleitung des Maschinenherstellers) wird eine T-Nr. mit oder ohne Adresserweiterung verwendet.

Wird eine richtig definierte T-Nr. in einem Bereich > 99 programmiert, wird das schnelle AB gelöscht.

Nicht definierte T-Nummern beeinflussen das AB nicht.

4.4.4 Programmierung der Analogausgänge (nur GA3 SW2)



- **Wertausgabe mit H11=...H14=**

Voraussetzung:

Die Optionen „Gemischte E/A-Baugruppe“ und „Analogausgänge belegbar“ sind aktiviert.

Die 4 Analogausgänge der gemischten E/A-Baugruppe können, falls vom Werkzeugmaschinenhersteller geplant, direkt aus einem Teileprogramm heraus programmiert werden. Dazu werden die **Sonderhilfsfunktionen H11=, H12=, H13= und H14=** verwendet.

Folgende Zuordnung gilt:

H11=	Sollwertvorgabe für Analogausgang 1
H12=	Sollwertvorgabe für Analogausgang 2
H13=	Sollwertvorgabe für Analogausgang 3
H14=	Sollwertvorgabe für Analogausgang 4

Sie sind als eine **eigene Hilfsfunktionsgruppe** zu verstehen und können somit auch zusammen mit einer normalen H-Hilfsfunktion bzw. mit einer E-Hilfsfunktion programmiert werden.

Ist die Option nicht aktiviert, werden H11= bis H14= wie normale H-Funktionen behandelt (mit den entsprechenden Programmiervorschriften).

Wird ein nicht freigegebener bzw. belegter Ausgang programmiert, wird die Vorgabe ignoriert. Es kommt zu keiner Fehlermeldung.

Falls nicht anders vom Maschinenhersteller vorgegeben, löschen 'RESET', 'M02/M30' und 'NOT AUS' die Sollwertvorgabe von der NC.

Programmierung:

Der Eintrag in die NC-Sollwertvorgabe erfolgt durch Programmierung von H11=, H12=, H13= bzw. H14=.

H11=, H12=, H13= und H14= werden als Sonderhilfsfunktionen behandelt. Sie kommen nicht zur Anzeige und werden nicht in die VDI NC-PLC Nahtstelle eingetragen. Auch können sie nicht überspeichert werden.

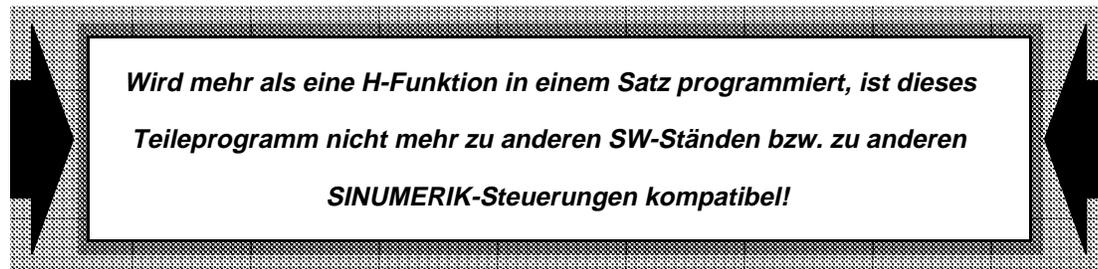
H11=, H12=, H13= und H14= können einzeln, als Paar, als Dreiergruppe oder alle zusammen in beliebiger Reihenfolge in einem Satz programmiert werden. Die Anzahl der in einem Satz programmierbaren Hilfsfunktionen bleibt jedoch weiterhin auf 6 beschränkt!

Die neuen Sollwertvorgaben aus dem Teileprogramm werden taktgleich mit dem Satzwechsel ausgegeben.

Die Sonderhilfsfunktionen werden wahlweise in [V] bzw. [mV] programmiert. Die eingestellte Vorwahl erfahren Sie aus der Anleitung des Maschinenherstellers.

Bei der Vorgabe in [V] gilt der Wertebereich 0...10,000 und bei Vorgabe in [mV] gilt der Wertebereich 0...10000.

Bei einer falschen Vorgabe wird die max. Spannung (+ bzw. -10V) ausgegeben.



- **Verwendung des 1.Analogausgangs für die Laserleistungssteuerung.**

Falls vom Werkzeugmaschinenhersteller geplant, wird der 1.Analogausgang auf der "gemischten E/A-Baugruppe" von der Laserleistungssteuerung benutzt (anstelle des Spindelausgangs). Dann darf die Sonderhilfsfunktion "H11=" nicht programmiert werden (keine Fehlermeldung).

- **Verwendung des 1.Analogausgangs für die analoge Tangentialwinkelausgabe.**

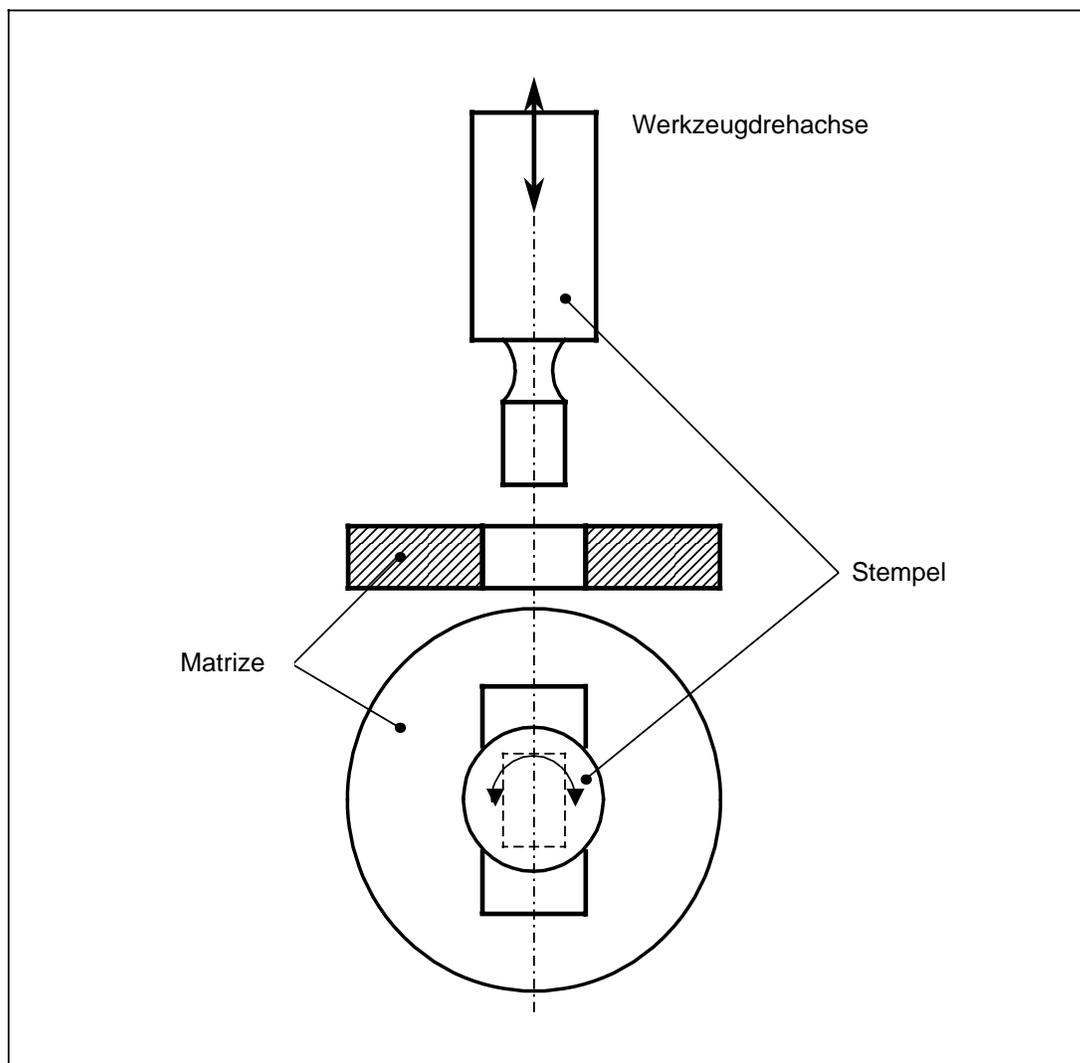
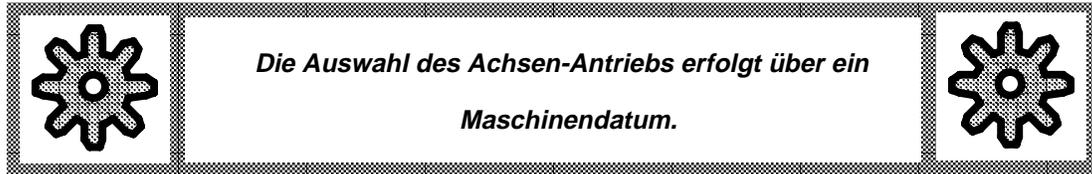
Falls vom Werkzeugmaschinenhersteller geplant, wird der 2.Analogausgang von der Funktion "analoge Tangentialwinkelausgabe" benutzt (anstelle des 6.Meßkreises). Dann darf die Sonderhilfsfunktion "H12=" nicht programmiert werden (keine Fehlermeldung).

4.5 Tangentialachse (Drehbare Werkzeugachse)

Es wird vorausgesetzt, daß die Tangentialachse eine durchdrehende Rundachse ist.

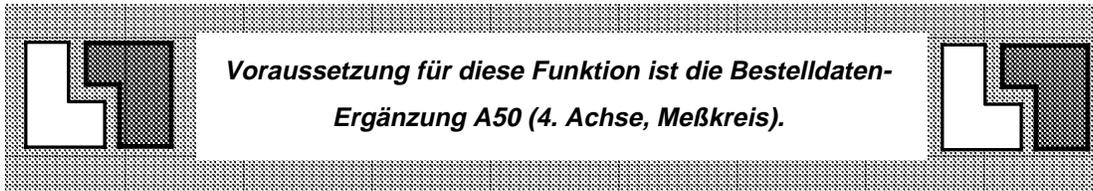
Es gibt Nibbelmaschinen, bei denen das Stanzwerkzeug und die dazugehörige Matrize drehbar sind. Dadurch ist ein Werkzeug vielseitiger verwendbar und ein Werkzeugwechsel kann häufig entfallen. Es lassen sich Konturen mit geringer Rauhgigkeit herstellen.

Die Achse für die Matrize kann durch ein Getriebe oder durch eine eigene NC-Mitschleppachse angetrieben sein.



Darstellung einer drehbaren Werkzeugachse

4.5.1 Mitschleppachse



Für drehbare Werkzeuge mit getrennten Antrieben für Ober- und Unterwerkzeug kann die Funktion Mitschleppachse verwendet werden. Dabei wird die 4. NC-Achse einer Leitachse zugeordnet. Es muß nur die Leitachse programmiert werden.

Beachte:

Hat die Mitschleppachse eine andere Position als die führende Tangentialachse, kann es zu falschen Positionierungen der Mitschleppachse kommen.

Dieses Problem tritt auf,

- nachdem die Mitschleppachse oder die führende Achse allein verfahren wurde, z. B. im JOG-Betrieb
- oder nach Referenzpunktfahren, wenn die Referenzpunktwerte nicht gleich sind und anschließend ein Programm gestartet wird, in dem die Tangentialsteuerung und gleichzeitig die X-Achse oder die Y-Achse programmiert ist.

Abhilfe:

Am Anfang eines jeden Programms wird die Tangentialachse allein im Satz ohne weitere Verfahrbewegungen und ohne Hubauslösung programmiert, z. B. C0. Dadurch werden beide Achsen in eine identische Ausgangsposition gefahren. Das nachfolgende Programm wird dann korrekt abgearbeitet.

Beispiel:

Leitachse: C
 Mitschleppachse: C1

4.5.2 Referenzpunkt, Bezugspunkt, Einbaulage

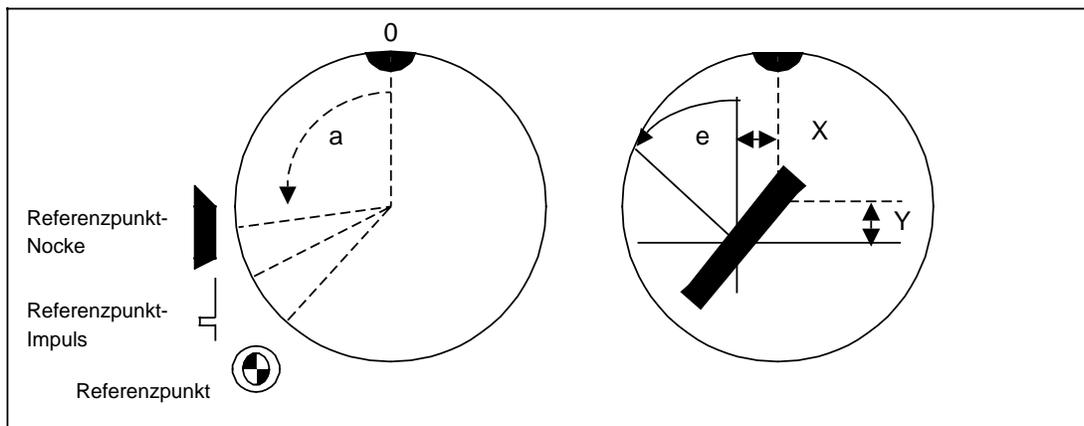
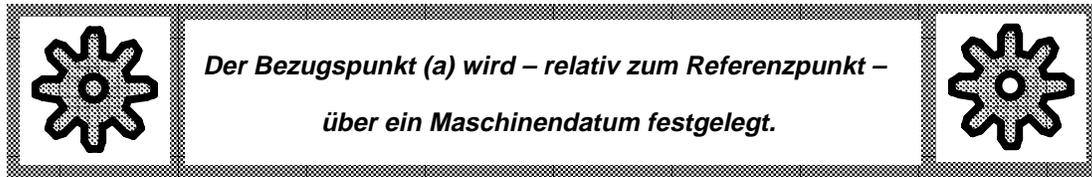


Schaubild für Referenzpunkt und Bezugspunkt mit Einbaulage

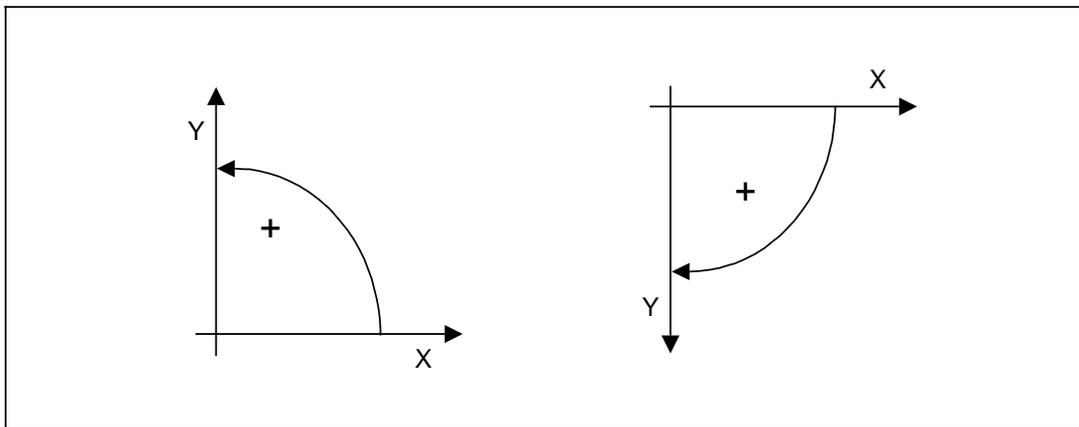
Der Referenzpunkt ist durch eine hardwaremäßig vorhandene Referenzpunkt-Nocke bestimmt.



In der Bezugslage kann das Werkzeug aus der Werkzeugaufnahme entnommen werden. Die Einbaulage ist der Winkel (e), den das Werkzeug abweichend vom Bezugspunkt hat, sowie die Lageabweichungen X und Y .

Entspricht die Einbaulage **nicht** der Bezugslage, müssen die Abweichungen programmier-technisch ausgeglichen werden (externe bzw. interne Nullpunktverschiebungen, siehe Kapitel 2.5).

4.5.3 Drehrichtung



Positive Drehrichtung für den kürzesten Weg

Positive Drehrichtung ist die Richtung, in der man die X-Achse drehen muß, um auf kürzestem Weg parallel zur Y-Achse zu liegen.

Die NC erkennt, in welcher Drehrichtung die gewünschte Position am schnellsten erreicht werden kann.

Beispiel:

C-Achse steht auf 0 und soll auf 180 drehen Achse dreht inkrementell um +180.

C-Achse steht auf 0 und soll auf 185 drehen Achse dreht inkrementell um –175.

Die Wegoptimierung wird bei Kettenmaßprogrammierung (G91) ausgeschaltet.

4.5.4 Programmierung

Verhalten bei G90/G91

G90 Absolutmaß-Angabe

Die Werkzeugachse wird in den programmierten Winkel gedreht.

G91 Kettenmaß-Angabe

Bei Kettenmaßangabe addieren sich wie auch bei den Linearachsen die Maßeinheiten zum letzten programmierten Winkel.

Werkzeugachse-Programmierung direkt

Die Werkzeugachse wird durch Nennung des Achsnamens direkt programmiert, wenn in dem Satz kein E/H aktiv ist.

Beispiel:

N10	C45	M 25	L _F	Stanzan an der aktuellen Position mit dem programmierten Winkel (45°)	
N15	M 20	L _F		M 25-Funktion ausschalten	
N20	X10	Y20	C90	L _F	In die programmierte X-Y-Position fahren und Werkzeugachse auf programmierten Winkel drehen

4.6 Tangentialsteuerung

4.6.1 Definition

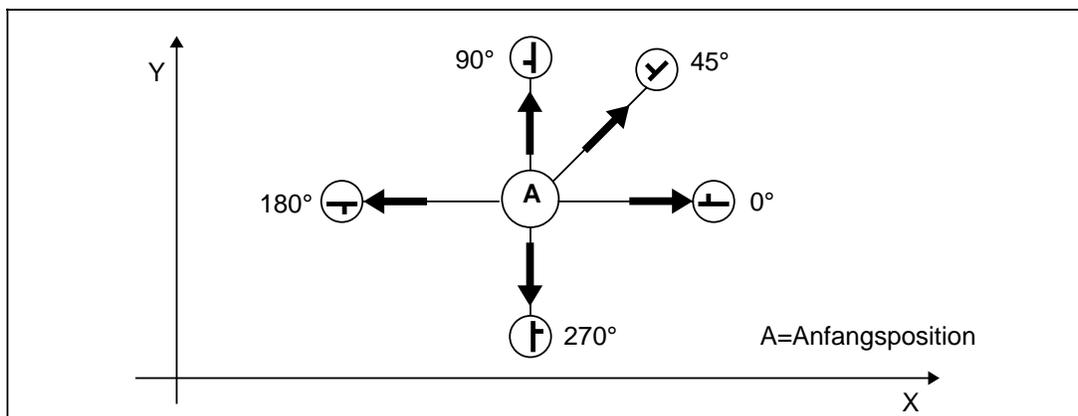
Die Tangentialsteuerung bewirkt ein automatisches Ausrichten des Stanzwerkzeuges senkrecht zum Richtungsvektor der programmierten Bahn. Vor der ersten Stanzung auf der programmierten Strecke wird das Tangentialwerkzeug positioniert. Der Tangentialwinkel ist immer auf die positive X-Achse bezogen. Ein programmierter Zusatzwinkel wird zum errechneten Winkel addiert.

Die Tangentialsteuerung wirkt bei Geraden- und Kreisinterpolation, aber nur in Sätzen, in denen die automatische Streckenaufteilung (E/H/H1) aktiv ist (siehe Kapitel 4.4.1).

Einschränkung:

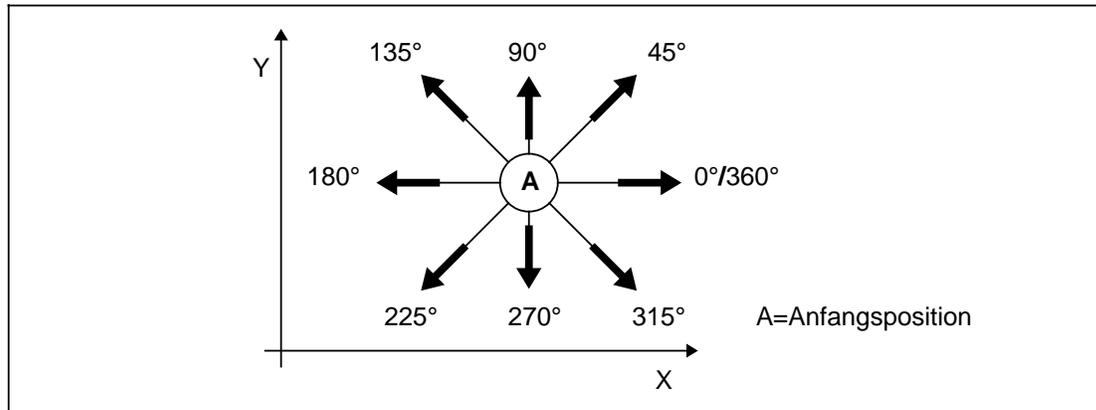
Die Option "Tangentialsteuerung" und die Option "Analoge Tangentialwinkelausgabe" schließen sich gegenseitig aus.

Bahnrichtungsvektor bei Tangentialsteuerung



4.6.1.1 Geradeninterpolation

Bei einer Geradeninterpolation ist der berechnete Bahnrichtungsvektor immer auf die positive X-Achse bezogen. Eine Bearbeitung in positiver Y-Richtung bedeutet also eine Ausrichtung der Tangentialachse auf einen Winkel von 90 Grad.



Beispiel für Geradeninterpolation

```
%11 L_F
```

```
N5 G0 X20 Y20 L_F
```

```
N10 X120 Y120 Z H5 M 22 L_F
```

```
N15 Y20 M 20 L_F
```

```
N20 X220 Y120 Z45 H5 M 22 L_F
```

```
N25 M 20 L_F
```

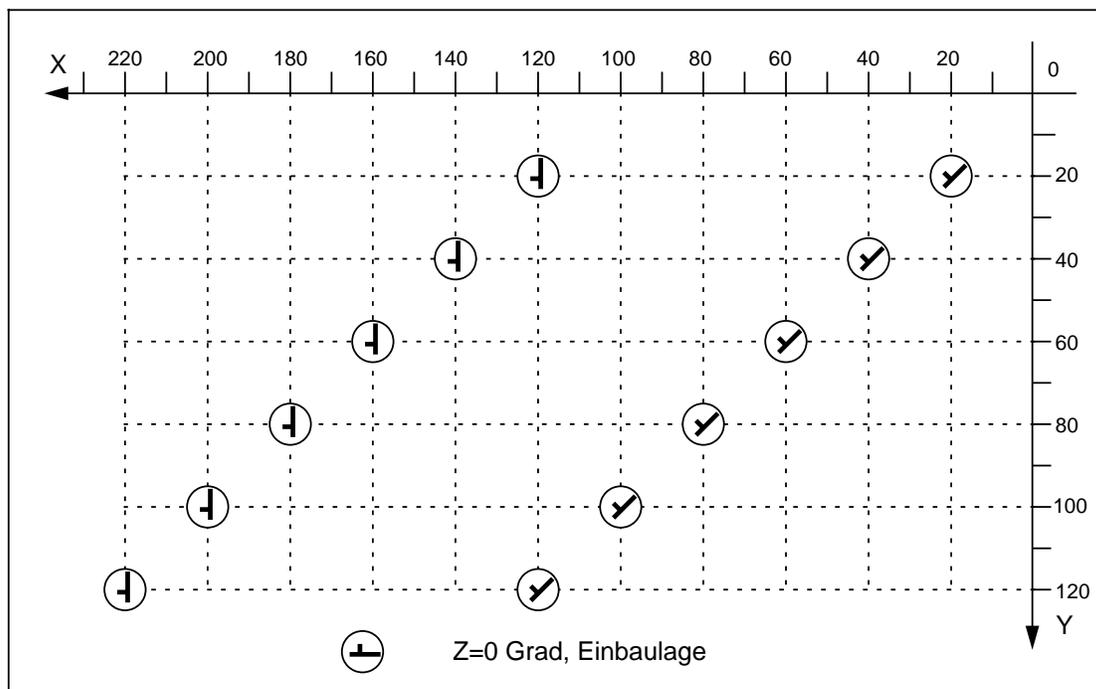
```
N30 M2 L_F
```

Positionieren

Definierter Nibbelbeginn, 5 Stanzauslösungen, Z tangential zur Bewegungsrichtung, also Z=45 Grad

Positionieren

5 Stanzauslösungen, Z tangential zur Bewegungsrichtung (45 Grad) plus Offset von 45 Grad, also Z=90 Grad



=0 ... 99, projektierbar über Maschinendatum

4.6.1.2 Kreisinterpolation G02/G03

Nicht realisiert

Funktion entfällt (Leerseite)

Funktion entfällt (Leerseite)

4.6.2 Aktivieren der Tangentialsteuerung

Die Tangentialsteuerung wird implizit für einen Satz aufgerufen, wenn der Achsname der Werkzeugdrehachse programmiert **und** die automatische Streckenaufteilung (E/H/H1) aktiv ist (siehe Kapitel 4.4.1).

Beispiel:

```
N5 G00 X0 Y0 LF
N10 X100 C45 E10 M 22 LF           Aktivierung der Tangentialsteuerung
N15 ...
:
```

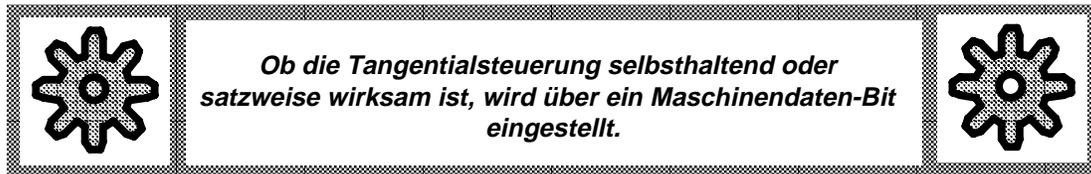
4.6.2.1 Selbsthaltende Tangentialsteuerung

Zur Erleichterung der Programmierung kann die Tangentialsteuerung auch über den aufrufenden Satz hinaus eingeschaltet bleiben.

Es muß dann nicht in jedem Satz die Tangential-Achse programmiert werden.

- Die selbsthaltende Tangentialsteuerung wird aktiviert durch die indirekte Programmierung.
- Sie wird deaktiviert durch M 20 bzw. M 21 (bei entsprechendem Maschinendatum) oder M-Funktionen der 2. M-Gruppe.

Bei der selbsthaltenden Tangentialsteuerung ist immer der zuletzt programmierte Zusatzwinkel gültig.

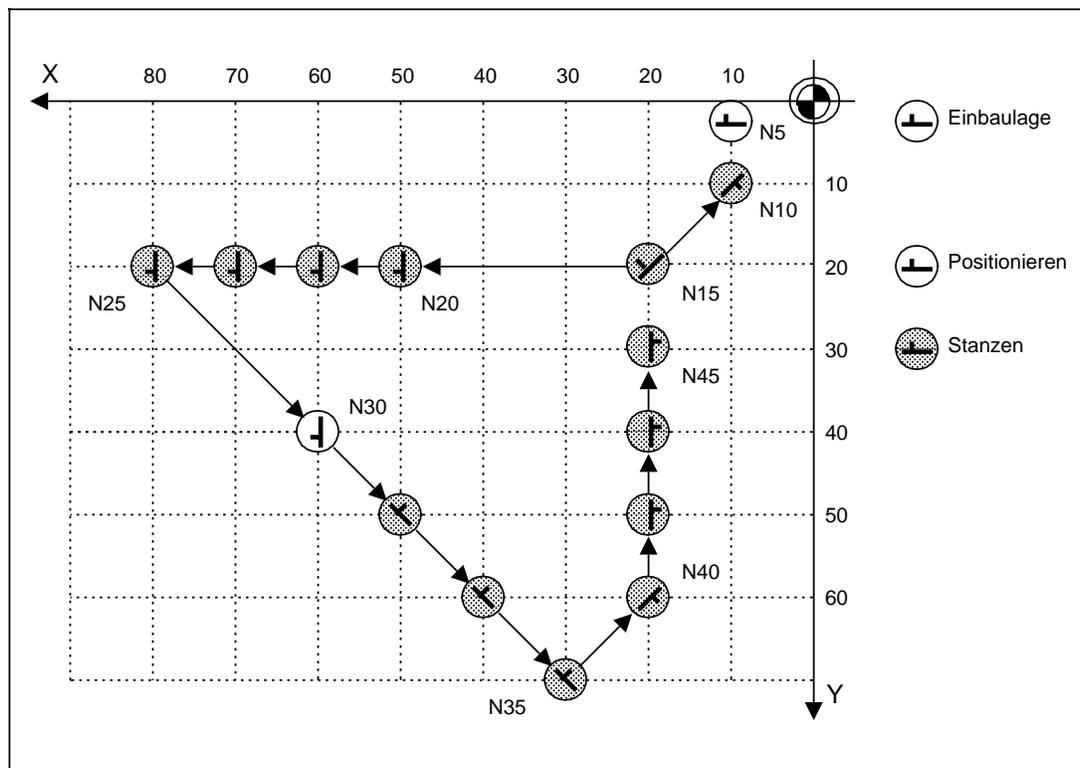


4.6.3 Programmierbeispiele für Tangentialsteuerung in Verbindung mit Stanzen/Nibbeln

4.6.3.1 Programmierbeispiel: Geradeninterpolation (G00/G01)

Voraussetzung: Selbsthaltung der Tangentialsteuerung ist aktiviert (Maschinendatum).

%13 L _F	
N5 G0 X10 Y5 L _F	Start-Position
N10 Y10 C225 M 25 L _F	X-, Y- und C-Achse positionieren gleichzeitig. Die C-Achse dreht sich inkrementell um -135° HUB.
N15 X20 Y20 C45 L _F	Die C-Achse dreht sich inkrementell um +180° auf absolut 45° HUB.
N20 X50 Y20 C90 M 20 L _F	C-Achse dreht sich auf 90°, Tangentialsteuerung ist ausgeschaltet.
N25 X80 Y20 E10 M 22 L _F	Es werden 4 Hübe ausgeführt mit einem um 90° verdrehten WZ.
N30 X60 Y40 M 20 L _F	Positionieren
N35 X30 Y70 C180 H3 M 25 L _F	Tangentialsteuerung einschalten Offsetwinkel 180° Es werden 3 Hübe gemacht.
N40 G91 C45 X-10 Y-10 L _F	C-Achse dreht auf 225°, 180° (N35)+45° HUB, Tangentialsteuerung nicht ein, da kein E oder H aktiv ist.
N45 G90 Y30 C0 H3 M 22 L _F	Tangentialsteuerung einschalten Offsetwinkel 0° Es werden 3 Hübe gemacht.
N50 M 20 L _F	
N45 M2 L _F	

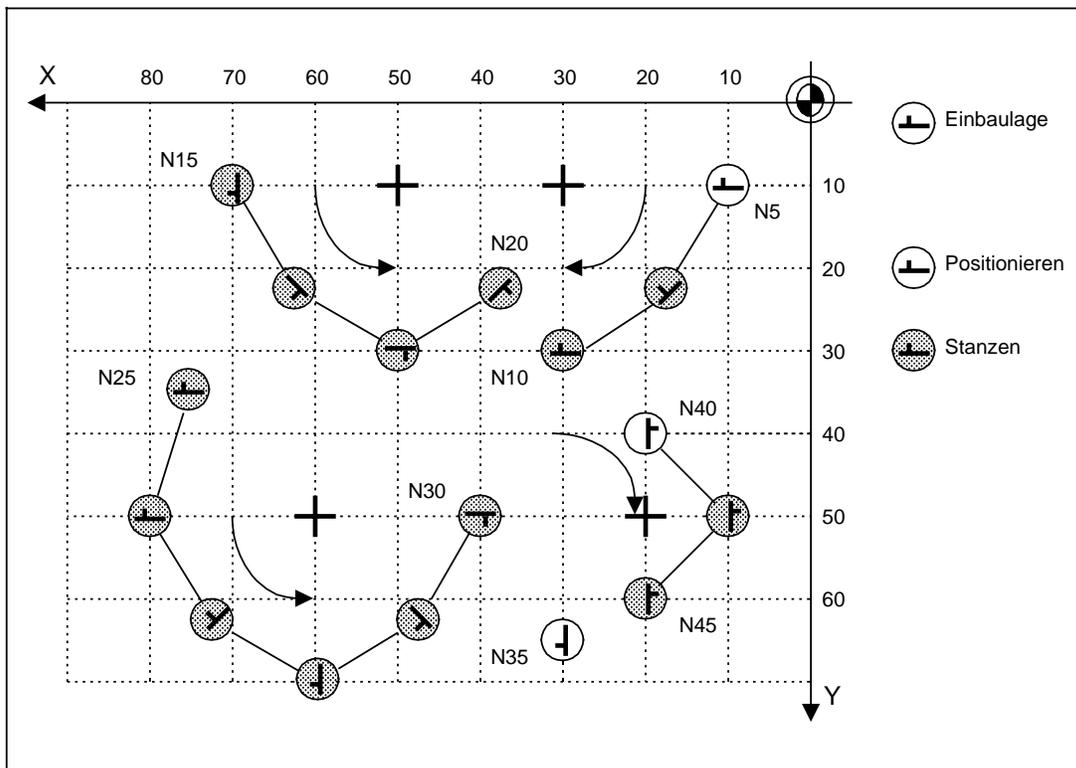


Darstellung des Programmierbeispiels in der XY-Ebene

4.6.3.2 Programmierbeispiel: Kreisinterpolation (G02/G03)

Voraussetzung: "Optimierung der Tangentialsteuerung bei tangentiellen Übergängen" ist nicht aktiv (Maschinendatum).

%14 L _F	
N05 G0 F60 X10 Y10 L _F	Start-Position
N10 G02 X30 Y30 I20 J0 C-90 H2 M25 L _F	Tangentialsteuerung einschalten, Offsetwinkel 270°
N15 G0 X70 Y10 M20 L _F	Positionieren,
N20 G03 X35,86 Y24,14 U20 C90 E16 M22 L _F	Tangentialsteuerung ausschalten Tangentialsteuerung einschalten, Offsetwinkel 90°
N25 G0 X74,14 Y35,86 C0 M25 L _F	Ein Hub
N30 G03 X40 Y50 I-14,14 J14,14 C0 H5 M22 L _F	Tangentialsteuerung einschalten, Offsetwinkel 0°, 5 Hübe,
N35 G0 X30 Y65 C90 M20 L _F	Positionieren Tangentialsteuerung ausschalten
N40 G91 X-10 Y-25 C180 L _F	Positionieren, C-Achse dreht auf 270°, 90° (N35)+180°
N45 G90 G02 Y60 I0 J10 H2 M25 L _F	Tangentialsteuerung ausgeschaltet! C-Achse bleibt auf 270°
N50 M20 L _F	
N55 M2 L _F	



Darstellung des Programmierbeispiels in der XY-Ebene

4.7 Analoge Tangentialwinkel Ausgabe

Diese Funktion ermöglicht die Ausgabe einer Analogspannung proportional zum Winkel des Bahnrichtungsvektors zweier interpolierender Achsen.

Die Winkelwerte 0 ... 360 Grad werden in Spannungswerte zwischen 0 V und 10 V umgerechnet.

Anwendungsmöglichkeit:

Zur Orientierung eines richtungsbehafteten Werkzeugs, z. B. eines Messers.

Einschränkung:

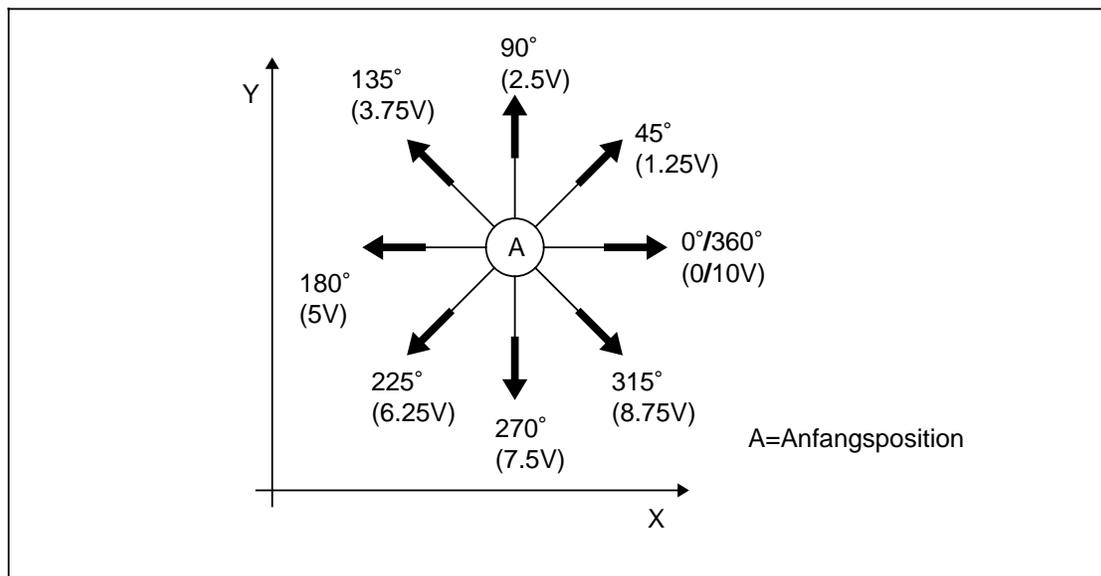
Die Option "Analoge Tangentialwinkel Ausgabe" und die Option "Tangentialsteuerung" schließen sich gegenseitig aus.

NOT-AUS löscht den zuletzt ausgegebenen Spannungswert sofort.

4.7.1 Grundlagen der Winkelberechnung

4.7.1.1 Geradeninterpolation G01/G00

Bei einer Geradeninterpolation ist der berechnete Bahnrichtungsvektor immer auf die positive X-Achse bezogen. Eine Verfahrensbewegung in positiver Y-Richtung bedeutet also einen Tangentialwinkel von 90 Grad. Die zugehörige Ausgangsspannung beträgt 2,5 V.

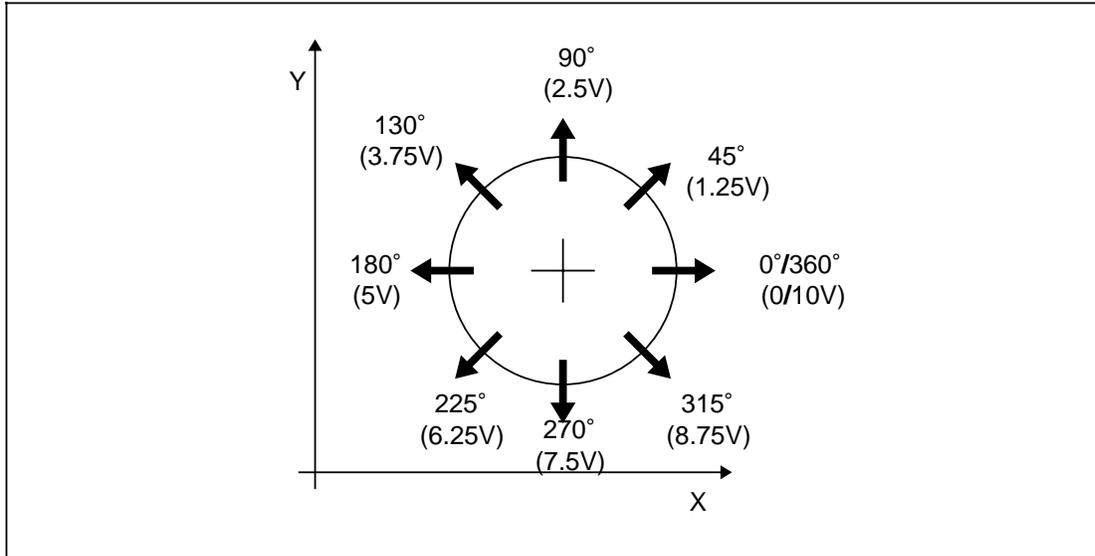


4.7.1.2 Kreisinterpolation G02/G03

Bei der Kreisinterpolation gibt es zwei Berechnungsmöglichkeiten.

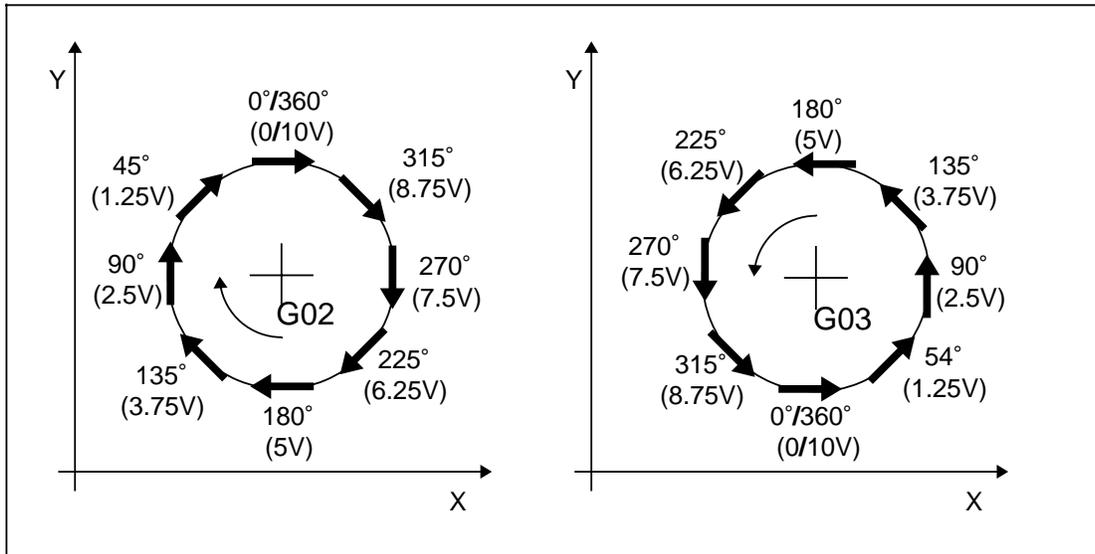
Winkelangabe mittelpunktsbezogen

Bei dieser Einstellung wird der Winkel jeweils auf den Mittelpunkt des programmierten Kreises bezogen.



Winkelangabe bezüglich der Bewegungsrichtung

Bei dieser Betriebsart wird der Winkel immer auf die Bewegungsrichtung bezogen.



Zwischen diesen beiden Winkelangaben kann per Maschinendatum ausgewählt werden.

4.7.2 Programmierung

Die analoge Tangentialwinkelausgabe muß durch eine spezielle M-Funktion aktiviert werden.

M 28 Ausgabe einschalten

M 28 ist selbsthaltend.

M 20 Ausgabe ausschalten

Grundsätzlich muß die Winkelberechnung bei Geraden- und Kreisprogrammierung durch Einschalten der automatischen Streckenaufteilung (E/H/H1) aktiviert werden (siehe Kapitel 4.4.1).

Bei Geradeninterpolation empfiehlt sich eine H1-Programmierung (siehe Kapitel 4.4.1.2). Die Strecke wird dann in einem Satz abgefahren.

Bei einem programmierten Kreis bietet sich eine Aufteilung des Kreises in Geradenteile der Länge "E" an.

Dabei ist es möglich, durch Anwahl des Bahnsteuerbetriebs (G64) eine konstante Bahngeschwindigkeit zu erreichen. Da bei der E/H-Aufteilung die programmierte Kreisbahn in viele kleine Geradenstücke unterteilt wird, muß der E-Wert mindestens so groß gewählt werden, daß von der Verfahrzeit die Satzwechselzeit nicht erreicht oder unterschritten wird. Ansonsten kommt es zu Geschwindigkeitseinbrüchen.

4.7.2.1 Beispiel

Voraussetzung:

"Optimierung der Tangentialsteuerung bei tangentialen Übergängen" ist aktiviert.

%777	L_F								
N5	G0	X	Y	F50	L_F				
N10	X100	Y100	G64	L_F					Keine Ausgabe, da M 28 nicht aktiv.
N15	X	Y	H1	M 28	L_F				Analogspannungsausgabe ist aktiv 6,25 V an Analogausgang. Die Strecke wird kontinuierlich abgefahren.
N20	G2	X	Y	J20	L_F				Keine Änderung, da kein E/H aktiv ist; Spannung 6,25 V bleibt unverändert anstehen.
N25	G2	X	Y	J20	E2	L_F			Analogspannungsausgabe ist aktiv; M 28 aus Satz N15 ist noch gültig; Spannungsänderung von ca. 4,9 V nach 0 V, Sprung auf 10 V und Reduzierung auf 5 V. Die Strecke wird kontinuierlich abgefahren.
N30	G4	F5	L_F						Verweilzeit; Spannung 5 V bleibt unverändert anstehen.
N35	M2	L_F							

=0 ... 99, projektierbar über Maschinendatum

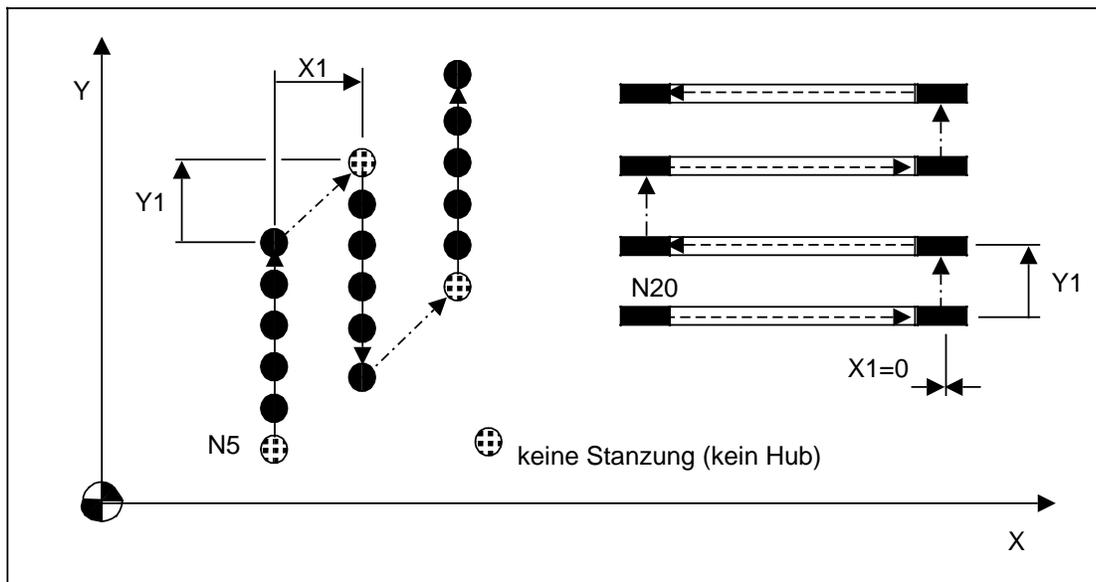
4.8 Mini Pattern

Die Funktion ermöglicht wegoptimiertes Duplizieren von linearen Konturen und Lochreihen bei minimalem Programmaufwand.

Beispiel:

```
N5 GO Y100 X1=20 Y1=15 S3=3 H5 M25 L_F  
:  
:  
N20 X150 Y1=20 S3=4 E5 M22 L_F  
:  
:  
M02 L_F
```

X1/Y1 = Versatz in X und Y
S3 = Anzahl der Sätze



Achtung:

Bei der Funktion MINI PATTERN müssen einige Besonderheiten beachtet werden.

- Der Nachfolgesatz nach einem MINI PATTERN-Satz darf nicht im Kettenmaß (G91) programmiert werden, da bei Kettenmaß-Programmierung die zuletzt programmierte Position als Istwert angenommen wird! Tatsächlich befindet sich das Werkzeug jedoch an einer anderen Stelle (siehe Beispiel).
- Bei einem MINI PATTERN-Satz darf "Koordinaten drehen" nicht aktiv sein!

**Diese Funktion ist eine Bestelldatenergänzung.
Die Namen X1 und Y1 sind dann aber nicht als
Achsnamen möglich.**

4.9 Laserleistungssteuerung

Die Funktion Laserleistungssteuerung ermöglicht eine Beeinflussung der Laserleistung, wie sie für das Laserschneiden oder das Laserschweißen benötigt wird. Neben der bahngeschwindigkeitsgesteuerten Leistungssteuerung für das Schneiden stehen zusätzliche Rampenfunktionen über der Zeit oder einer Strecke für das Schweißen zur Verfügung. Die Eckdaten der Laserleistungssteuerung sind frei programmierbar.

Die Ausgabe der Laserspannung erfolgt auf dem analogen Spindelausgang.

Die Funktion gliedert sich in 3 Gruppen:

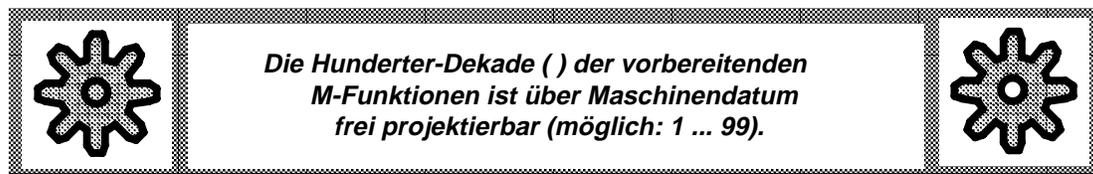
1. Vorbereitende Festlegung der Grenzdaten
2. Auswahl und Ausführung der Laserfunktion
3. Schnelle M-Funktionen für die Lasersteuerung (siehe Kapitel 4.3)

4.9.1 Vorbereitende Festlegung der Grenzdaten

Die Festlegung der Eckdaten für die Laserleistungssteuerung findet normalerweise am Anfang eines Teileprogrammes statt, sie können aber auch während der Bearbeitung geändert werden. Dazu müssen mit speziellen M-Funktionen (M 31 ... M 36) die Daten für die Spannungswerte E, die Bahngeschwindigkeiten F und die Zeiten T festgelegt werden. Diese Zuordnung geschieht in maximal sechs NC-Sätzen, die die Form wie in Tabelle 1 angegeben haben müssen. Die Reihenfolge der Programmierung ist beliebig wählbar.

programmierter NC-Satz		Zuordnung
M 31	E F T*	E U1 F V1 T T1
M 32	E F T*	E U2 F V2 T T2
M 33	E F T*	E U3 F V3 T T3
M 34	T	T T4
M 35	T	T T5
M 36	T	T T6

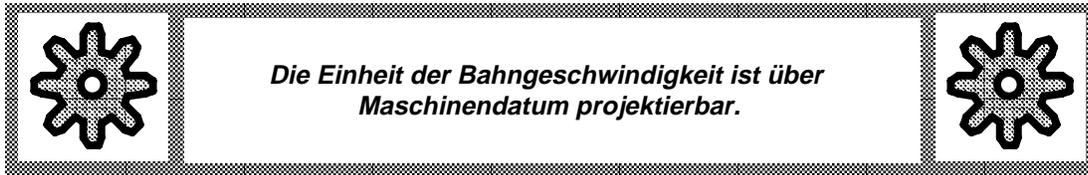
Zuordnung der Grenzdaten



Bedeutung:

- E Spannungswerte für die Laseransteuerung, Einheit: Volt (V)
 F Bahngeschwindigkeit, Einheit: mm/min bzw. m/min
 T Zeit, Einheit: ms

* In diesen Sätzen können auch einzelne Funktionen programmiert werden.



Die Adressen E/F/T haben nur in Verbindung mit M 31 ... M 36 diese spezielle Funktion. Damit besteht die Möglichkeit, eine maschinenspezifische bzw. eine an das Material angepasste Laserkennlinie zu erstellen. Die Zeiten T4 ... T6 korrespondieren mit den Spannungen U1 ... U3.

Hinweis:

Die in Verbindung mit einer vorbereitenden M-Funktion programmierten F- und T-Werte werden **nicht** an die VDI-Nahtstelle zur PLC übergeben und kommen **nicht** zur Anzeige.

Ab Grundausführung 3:

Bei den vorbereitenden M-Funktionen ist auch die Schreibweise mit Adreßerweiterung (E1=... /T1=...) zulässig.

Programmbeispiel:

```
N05 M231 E1.5 F3 T1000 LF
N10 M232 E5.8 F8 T2000 LF
N15 M234 T500 LF
```

Der NC-Satz N05 legt die Daten für den unteren Grenzwert der Kennlinie fest (Minimalspannung U1=1.5 V, minimale Geschwindigkeit V1=3 m/min und Zeit T1=1 s), im Satz N10 wird der obere Grenzwert festgelegt (Maximalspannung U2=5.8 V, maximale Geschwindigkeit V2=8 m/min und Zeit T2=2 s). Der Satz N15 gibt eine Pulszeit T5=0,5 s an. Diese ist automatisch an die Spannung U2=5.8 V gekoppelt. In diesem Beispiel wurde für die M-Gruppe " " mit 2 vorbelegt.

4.9.2 Auswahl und Ausführung der Laserfunktion

Die Beeinflussung der Laserleistung erfolgt über die Auswahl spezieller S-Funktionen. Dies geschieht direkt im Teileprogramm nach der Festlegung der Grenzdaten für die Laserleistungssteuerung. Die programmierte S-Funktion ist am Satzanfang aktiv und bleibt so lange gültig, bis eine neue S-Funktion angewählt wird. In jedem NC-Satz ist nur eine S-Funktion möglich, innerhalb des Teileprogrammes können beliebig viele programmiert werden. Die nachfolgende Tabelle gibt die verwendbaren S-Funktionen und deren Wirkung auf die Laserleistungssteuerung wieder.

S-Funktionen	Laserspannung
S 00 ... S 05	Konstantspannungen
S 06 ... S 09	Bahnsteuerung
S 10 ... S 12	Streckensteuerung
S 13 ... S 18	Zeitsteuerung

S-Funktionen der Laserleistungssteuerung

Programmbeispiel:

```

N05 M231 E1.5 F3 T1000 LF
N10 M232 E5.8 F8 2000 LF
N15 M234 T500 LF
:
:
:
N30 X200 Y150 S7 LF
:
:
:
N50 X100 Y100 S0 LF

```

Beschreibung der NC-Sätze N05 bis N15 siehe oben. Im Satz N30 wird die Laserspannung über eine Bahngeschwindigkeitssteuerung geregelt. Im Satz N50 wird die Laserspannung ausgeschaltet.

4.9.3 Schnelle M-Funktionen für die Laserleistungssteuerung

Zusätzlich zu den S-Funktionen können schnelle M-Funktionen programmiert werden. Diese setzen oder löschen beim Satzwechsel vor dem Verfahren der Achsen je nach Programmierung Ausgangsbits auf der Ausgangsbaugruppe. Sie können z. B. zum Öffnen oder Schließen des Shutters verwendet werden.

Diese schnellen M-Funktionen sind in Kapitel 4.3 beschrieben.

Programmbeispiel:

```

N05 M231 E1.5 F3 T1000 LF
N10 M232 E5.8 F8 T2000 LF
N15 M234 T500 LF
:
:
:
N30 X200 Y150 S7 M201 LF
:
:
:
N50 X100 Y100 S0 M211 LF

```

Der NC-Satz N05 legt die Daten für den unteren Grenzwert der Kennlinie fest (Minimalspannung $U_1=1.5$ V, minimale Geschwindigkeit $V_1=3$ m/min und Zeit $T_1=1$ s). Im Satz N10 wird der obere Grenzwert festgelegt (Maximalspannung $U_2=5.8$ V, maximale Geschwindigkeit $V_2=8$ m/min und Zeit $T_2=2$ s). Der Satz N15 gibt eine Pulszeit $T_5=0,5$ s an. Diese ist automatisch an die Spannung $U_2=5.8$ V gekoppelt. In diesem Beispiel wurde für die M-Gruppe " " mit 2 vorbelegt (Maschinendatum).

Im Satz N30 wird die Laserspannung über eine Bahngeschwindigkeitssteuerung geregelt. Gleichzeitig wird auf der Ausgangsbaugruppe Bit 1 gesetzt. Im Satz N50 wird die Laserspannung ausgeschaltet und Bit 1 auf der Ausgangsbaugruppe gelöscht.

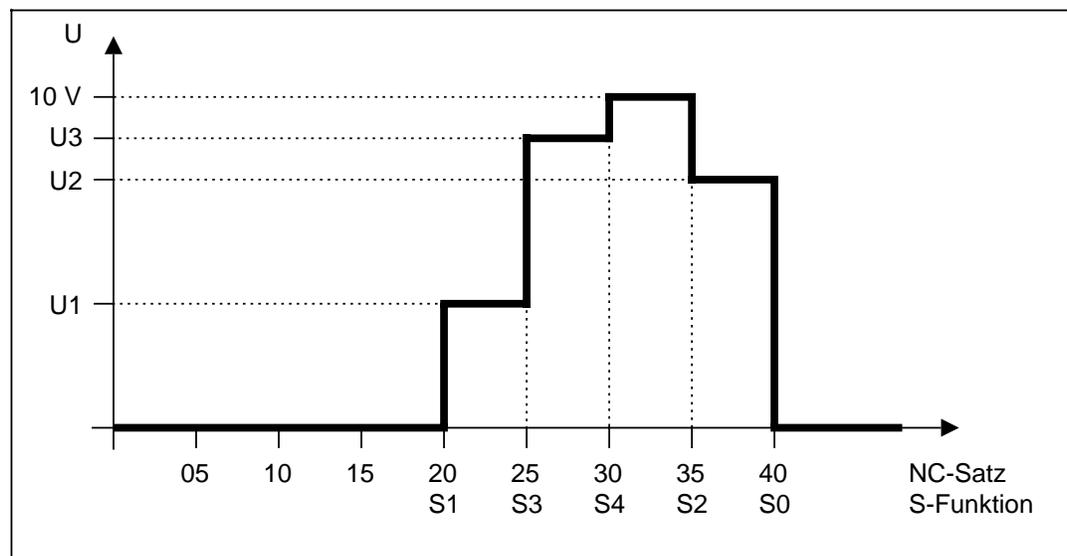
4.9.4 Beschreibung der Laserfunktionen

Im folgenden werden die S-Funktionen der Laserleistungssteuerung beschrieben.

4.9.4.1 Ausgabe von Konstanzspannungen

Die Laserspannung wird als Sprungfunktion beim Satzwechsel ausgegeben.

- S00 Laserausgangsspannung=0 V
- S01 Laserausgangsspannung=U1
- S02 Laserausgangsspannung=U2
- S03 Laserausgangsspannung=U3
- S04 Laserausgangsspannung=10 V



Beispiel für Laserkonstanzspannungen

Programmbeispiel:

```

N05 M231 E4 LF
N10 M232 E8 LF
N15 M233 E9 LF
N20 X100 S1 F10 LF
N25 X200 Y150 S3 LF
N30 X150 S4 LF
N35 X100 Y100 S2 LF
N40 X0 Y0 S0 LF
    
```

Im NC-Satz N05 bis N15 werden die Grenzwerte für die Laserleistungssteuerung festgelegt. In den Sätzen N20 bis N35 werden nacheinander die Laserspannungen U=4 V, U=9 V, U=10 V und U=8 V als Sprungfunktionen nach dem Satzwechsel durch das Programmieren der jeweiligen S-Funktion ausgegeben. Im Satz N40 wird durch S0 die Laserspannung auf Null geschaltet.

Ab SINUMERIK 810N/820N Grundausführung 3:**Ausgabe von Konstantspannungen mit variabler Spannungsvorgabe E1=...**

Mit dieser Funktion können im laufenden Teileprogramm auch Spannungswerte programmiert werden, die nicht durch die vorbereitenden M-Funktionen (M 31 ... M 33) festgelegt sind. Dadurch sind beliebig viele verschiedene Spannungen möglich.

Die variable Spannung wird mit E1=... [V] in einem Satz mit der gewählten S-Funktion programmiert. Die nach diesem Satz erreichte Ausgangsspannung bleibt bis zur Programmierung einer neuen S-Funktion erhalten.

Der den vorbereitenden M-Funktionen zugeordnete Spannungswert wird durch die variable Spannungsvorgabe nicht überschrieben.

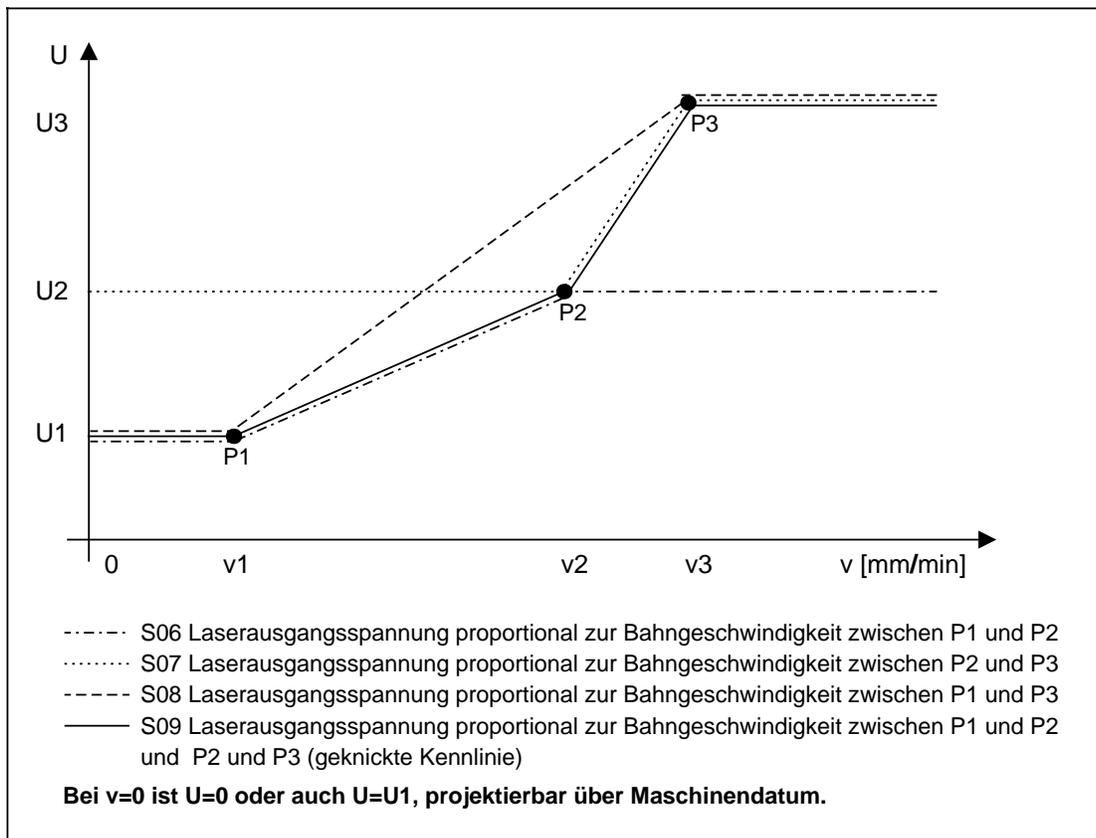
Beispiele:

N5 S00	Laserausgangsspannung= 0 V
N5 S00 E1=3.5	Laserausgangsspannung= 0 V
N5 S01	Laserausgangsspannung= U1
N5 S01 E1=3.5	Laserausgangsspannung=3.5 V
N5 S02	Laserausgangsspannung= U2
N5 S02 E1=3.5	Laserausgangsspannung=3.5 V
N5 S03	Laserausgangsspannung= U3
N5 S03 E1=3.5	Laserausgangsspannung=3.5 V
N5 S04	Laserausgangsspannung= 10 V
N5 S04 E1=3.5	Laserausgangsspannung= 10 V
N5 S05 E1=3.5	Laserausgangsspannung=3.5 V

4.9.4.2 Ausgabe der Laserspannung als Funktion der Bahngeschwindigkeit

Die Ausgabe der Laserspannung erfolgt in Abhängigkeit von der aktuellen Bahngeschwindigkeit. Dies gilt, solange die S-Funktion für bahngeschwindigkeitsabhängigen Betrieb angewählt ist.

S06	Laserausgangsspannung proportional zur Bahngeschwindigkeit zwischen $v1/U1$ und $v2/U2$
S07	Laserausgangsspannung proportional zur Bahngeschwindigkeit zwischen $v2/U2$ und $v3/U3$
S08	Laserausgangsspannung proportional zur Bahngeschwindigkeit zwischen $v1/U1$ und $v3/U3$
S09	Laserausgangsspannung proportional zur Bahngeschwindigkeit zwischen $v1/U1$ und $v2/U2$ und zwischen $v2/U2$ und $v3/U3$ (geknickte Kennlinie)



Kennlinie zur bahngeschwindigkeitsabhängigen Steuerung

Programmbeispiel:

```

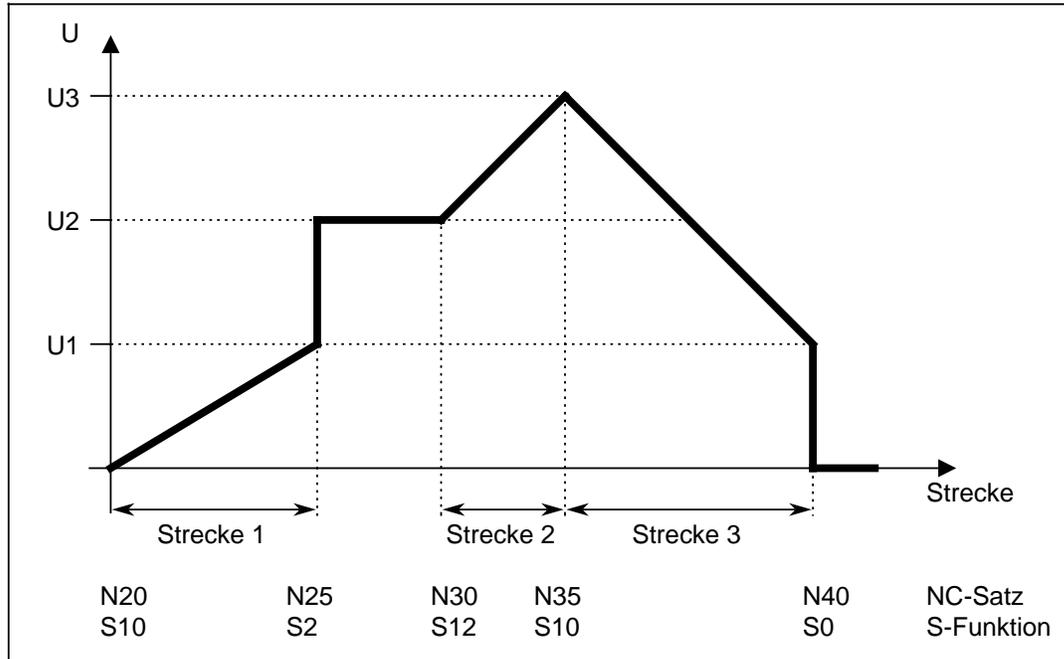
N05 M231 E4 F3 LF
N10 M232 E8 F8 LF
N15 M233 E9 F12 LF
N20 X100 S1 F7 LF
N25 X200 Y150 S6 LF
N30 X150 S8 F10 LF
N35 X100 Y100 LF
N40 X0 Y0 S0 LF
    
```

In den NC-Sätzen N05 bis N15 werden die Grenzwerte für die Laserleistungssteuerung festgelegt. Im Satz N20 wird die Laserspannung auf $U=4$ V geschaltet. Satz N25 beinhaltet eine Bahnsteuerung der Laserspannung zwischen den Eckdaten $U=4$ V/ $F=3$ m/min und $U=8$ V/ $F=8$ m/min, programmierte Geschwindigkeit $V=7$ m/min. Die Sätze N30 bis N35 regeln die Laserspannung auf der Kennlinie mit den Eckdaten zwischen $U=4$ V/ $F=3$ m/min und $U=9$ V/ $F=12$ m/min, programmierte Geschwindigkeit $F=10$ m/min. Im Satz N40 wird durch S0 die Laserspannung auf Null geschaltet.

4.9.4.3 Ausgabe der Laserspannung als Funktion der Strecke

Die Ausgabe der Laserspannung erfolgt streckengesteuert. Dabei wird die aktuelle Spannung des vorangegangenen Satzes über der im NC-Satz programmierten Bahn auf den von der S-Funktion angewählten Spannungswert verändert. Dies kann ein Auf- oder Abrampen der Spannung bedeuten.

- S10 Laserausgangsspannung als Funktion der Strecke, Ausgangsspannung U1
 S11 Laserausgangsspannung als Funktion der Strecke, Ausgangsspannung U2
 S12 Laserausgangsspannung als Funktion der Strecke, Ausgangsspannung U3



Beispiel für Streckensteuerung der Laserspannung

Ab Grundauführung 3:

Ausgabe der Laserspannung als Funktion der Strecke mit variabler Spannungsvorgabe E1=... .

Die variable Spannung wird mit E1=... [V] im Satz mit der gewählten S-Funktion programmiert.

Beispiel:

N6 S10 X10 Spannung über Strecke auf U1
 N6 S10 E1=3.5 X10 Spannung über Strecke auf 3.5 V

Gleiche Wirkung auch bei S11 und S12!

Gültigkeitsbereich Streckensteuerung (S10 ... S12)

Es gilt "FAKTOR" =
$$\frac{\text{programmierte Spannungsdifferenz [mV]} \cdot 65536}{\text{programmierte Strecke [unit]}}$$

Es muß gelten "FAKTOR" 1.

Trifft dies nicht zu, wird die programmierte Endspannung **am Anfang der programmierten Strecke als Sprungfunktion** ausgegeben. Bei Konturprogrammierung dient zur Berechnung immer die erste Teilstrecke.

Programmbeispiel:

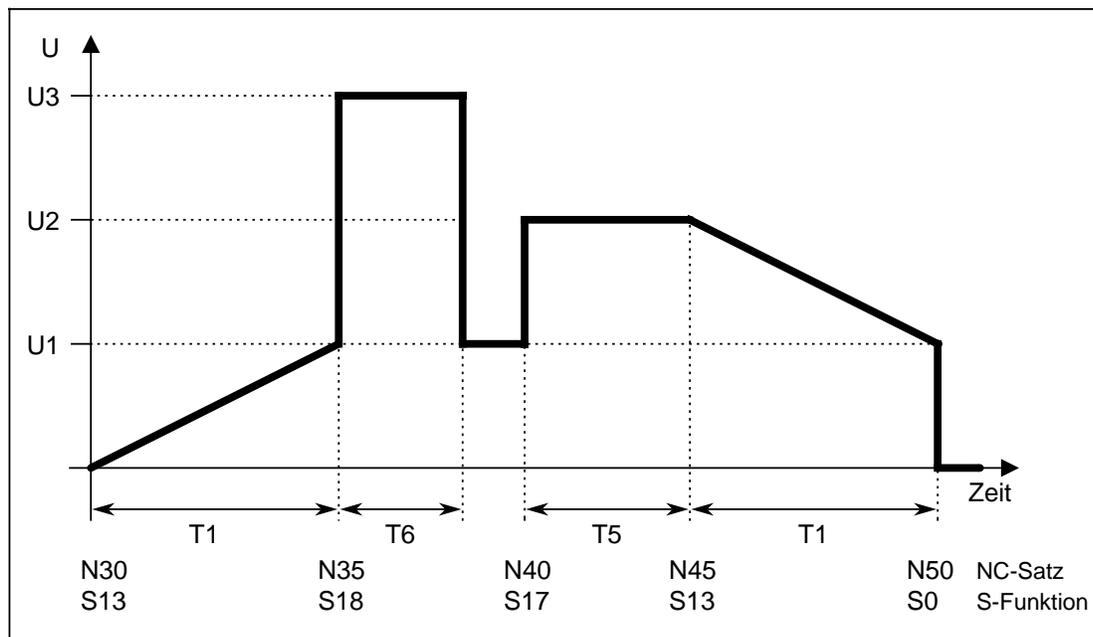
```
N05 M231 E4 LF
N10 M232 E8 LF
N15 M233 E9 LF
N20 X100 S10 F10 LF
N25 X200 Y150 S2 LF
N30 X150 Y200 S12 LF
N35 X100 Y100 S10 LF
N40 X0 Y0 S0 LF
```

In den NC-Sätzen N05 bis N15 werden die Grenzwerte für die Laserleistungssteuerung festgelegt. Im Satz N20 wird über der Strecke X=100 die Laserspannung von 0 V auf U=4 V hochgerampft. Satz N25 beinhaltet eine Sprungfunktion mit U=8 V. Im Satz N30 wird die Laserspannung von U=8 V auf U=9 V über der Bahnstrecke hochgerampft, im darauffolgenden NC-Satz erfolgt das Herunterrampen der Spannung auf den Wert U=4 V. Im Satz N40 wird durch S0 die Laserspannung auf Null geschaltet.

4.9.4.4 Ausgabe der Laserspannung als Funktion der Zeit

Die Ausgabe der Laserspannung erfolgt zeitgesteuert. Dies kann durch Rampen- oder durch Pulsfunktionen geschehen. Die Eingabe der Zeitparameter erfolgt in Millisekunden (ms).

- S13 Laserausgangsspannung als Funktion der Zeit T1, Ausgangsspannung U1 (Rampe)
- S14 Laserausgangsspannung als Funktion der Zeit T2, Ausgangsspannung U2 (Rampe)
- S15 Laserausgangsspannung als Funktion der Zeit T3, Ausgangsspannung U3 (Rampe)
- S16 Laserausgangsspannung als Funktion der Zeit T4, Ausgangsspannung U1 (Puls)
- S17 Laserausgangsspannung als Funktion der Zeit T5, Ausgangsspannung U2 (Puls)
- S18 Laserausgangsspannung als Funktion der Zeit T6, Ausgangsspannung U3 (Puls)



Beispiel für Zeitsteuerung der Laserspannung

Programmbeispiel:

```

N05 M231 E4 T100 LF
N10 M232 E7 LF
N15 M233 E9 LF
N20 M235 T75 LF
N25 M236 T50 LF
N30 X100 S13 F10 LF
N35 X200 Y150 S18 LF
N40 S17 LF
N45 X100 Y100 S13 LF
N50 X0 Y0 S0 LF

```

In den NC-Sätzen N05 bis N25 werden die Grenzwerte für die Laserleistungssteuerung festgelegt. Im Satz N30 wird die Laserspannung vom Wert U=0 V in der Zeit T=0.1 s auf den Wert U=4 V hochgerampft. Der nächste Satz pulst die Spannung 0.05 s auf den Wert U=9 V. (Ist der Verfahrssatz zeitlich länger als 0.05 s, so fällt die Spannung nach 0.05 s auf den vor dem Satz gültigen Spannungswert.) Im Satz N40 ist ein Puls von 0.075 s mit der Spannung U=7 V programmiert. Im Satz N45 wird durch S13 die Laserspannung in T=0.1 s auf U=4 V gerampft. Satz N50 schaltet die Laserspannung auf Null.

Ab Grundauführung 3:**Variable Spannungs- und Zeitwerte bei Lasersteuerung als Funktion der Zeit.**

Die variable Spannung wird mit E1=... [V], die variable Zeit wird mit T1=... [ms] in einem Satz mit der gewählten S-Funktion programmiert.

Der der vorbereitenden M-Funktion zugeordnete Spannungs-/Zeitwert wird durch die variable Spannungs-/Zeitvorgabe nicht überschrieben.

Beispiel (1):

```

N7 S13                               Spannung in T1 auf U1
N7 S13 E1=3                           Spannung in T1 auf 3 V
N7 S13 T1=1000                         Spannung in 1s auf U1
N7 S13 E1=5 T1=2000                   Spannung in 2s auf 5 V

```

Gleiche Wirkung auch bei S14 und S15!

Beispiel (2):

```

N8 S16                               Spannung in T4 auf U1
N8 S16 E1=7                           Spannung in T4 auf 7 V
N8 S16 T1=2000                         Spannung in 2s auf U1
N8 S16 E1=4 T1=1000                   Spannung in 1s auf 4 V

```

Gleiche Wirkung auch bei S17 und S18!

Gültigkeitsbereich Zeitsteuerung (S13 ... S18)

Es gilt "FAKTOR" =
$$\frac{\text{programmierte Spannungsdifferenz [mV]} \cdot 8}{\text{programmierte Zeit [ms]}}$$

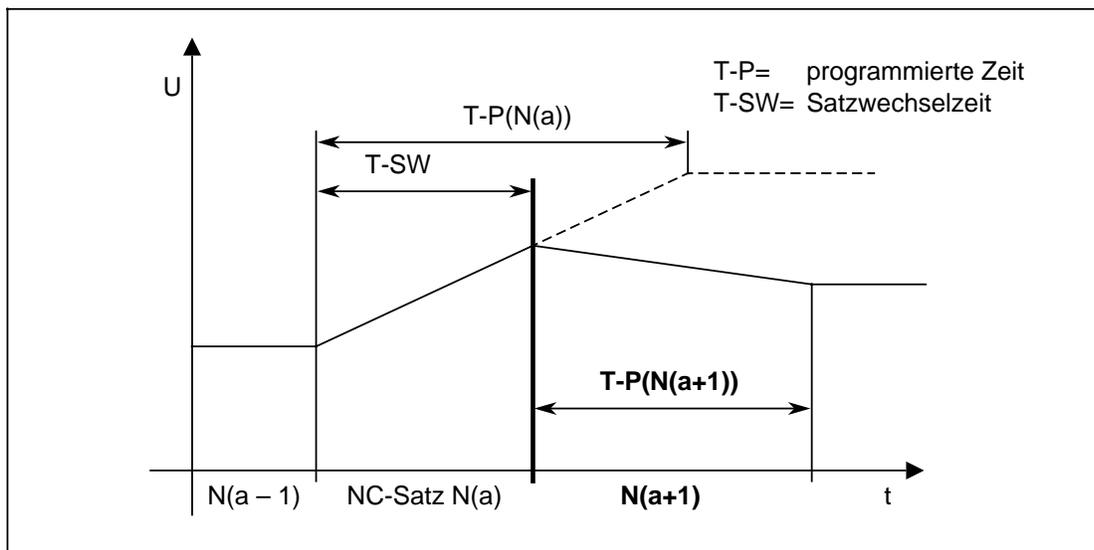
Für "FAKTOR" ist der Bereich 1 "FAKTOR" 32767 zulässig.

Trifft dies nicht zu, wird die programmierte Endspannung **nach Ablauf der Zeit als Sprungfunktion** ausgegeben.

Die Berechnungen erfolgen im IPO-Takt, also alle (4+0.5·MD 155) ms.

Hinweis zur Programmierung:

Findet ein Satzwechsel vor Ablauf der programmierten Zeit statt und ist in diesem Satz eine neue Laser-S-Funktion programmiert, so wird die noch andauernde Funktion des vorhandenen Satzes abgebrochen.



4.10 Nibbelspezifische Alarmmeldungen

Alarm Nr.	Alarm-Text	Abhilfe
2231	Schnellstop unzulässig	
2232	Kein E-Wert vorhanden	M 22 programmiert ohne E- oder H-Wert. E- oder H-Wert programmieren.
2233	Achsreihenfolge falsch	Bei Kreisprogrammierung mit E/H-Aufteilung müssen die X- und Y-Achse programmiert sein.
2234	Pratzenschutz: falsche NV	Interner Fehler NC-RESET und nochmals NC-Start drücken.
2235	E-Inkrement zu groß	Kleinerer E-Wert programmieren, siehe Programmieranleitung "Exzenter Betrieb".
2236	falscher H-Wert	H-Wert größer 65535 programmiert.
2237	Anzahl der Nibbelinkremente zu groß	Programmierter E-Wert wurde zu klein gewählt bzw. die programmierte Strecke zu groß Korrektur
2238	Falscher Radius bei Nibbeln	Programmierter E-Wert ist größer als der programmierte Radius E-Wert reduzieren.
2239	Stanzen im Pratzenschutzbereich	Programmierte Position überprüfen bzw. die Pratzenpositionen (R241 ... R246) kontrollieren.
2240	Zielposition im Pratzenschutzbereich	Programmierte Position überprüfen bzw. die Pratzenpositionen (R241 ... R246) kontrollieren.

Alarm Nr.	Alarm-Text	Abhilfe
2241	Drehwinkel bei Nibbeln zu groß	Siehe Programmieranleitung "Exzenter-Betrieb".
2242	Exzenter außerhalb O.T.	Siehe Programmieranleitung "Exzenter-Betrieb".
2243	Werkzeug noch im Blech	Siehe Programmieranleitung "Exzenter-Betrieb".
2244	Falscher H1-Wert	H1=Wert größer 65535 programmiert.
2245	Stanze noch im Blech	Eingang "Stanze aus Blech" liegt auf 0 V. Diesen auf +24 V legen. Sonst keine Achsbewegung möglich. Siehe Betriebsanleitung des Maschinenherstellers.
2246	Schleppabstand nicht abgebaut	Programmierter E-Wert zu groß oder Zeit für "Vorzeitige Stanzauslösung" (MD 169, 170) zu groß kleineren E-Wert programmieren bzw. MD 169, 170 anpassen.
2248	Falsche Projekt. MD 175-178	Die MD 175-178 passen nicht zu den bereits eingerichteten DB's für die Anwenderschnittstelle. Eingaben überprüfen
2249	HW-Schnittstelle belegt	Die im MD 179 hinterlegte HW-Schnittstellen-Nr. ist bereits durch die Universal- oder Anwender-Schnittstelle belegt. Andere Schnittstellenverteilung wählen
2250	Schnittstelle nicht vorhanden	Die HW-Schnittstellen-Nr. existiert nicht, die im MD 179 hinterlegt wurde (MD 179 <> 1; 2). MD 179 berichtigen
3035	Ind. Adressierung fehlerhaft	Die projektierte Bildbeschreibung ist fehlerhaft. Überprüfen der projektierten Bildbeschreibung

5 Unterprogramme

5.1 Anwendung

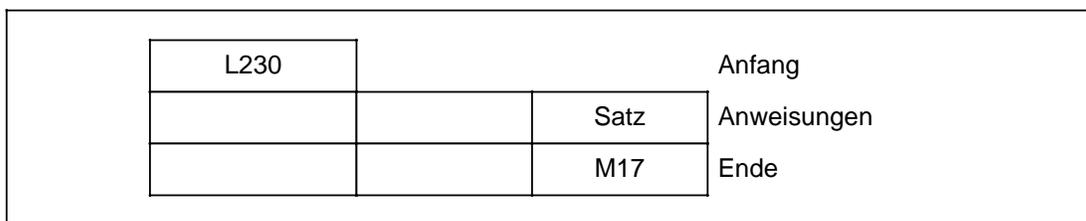
Kommt bei der Bearbeitung eines Werkstücks mehrmals der gleiche Arbeitsgang vor, so kann dieser als Unterprogramm eingegeben und beliebig oft im Teileprogramm oder über Handeingabe aufgerufen werden.

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 Be-
messung im Unterprogramm an verschiedenen Stellen des Werkstücks wiederholt werden.

5.2 Unterprogramm-Aufbau

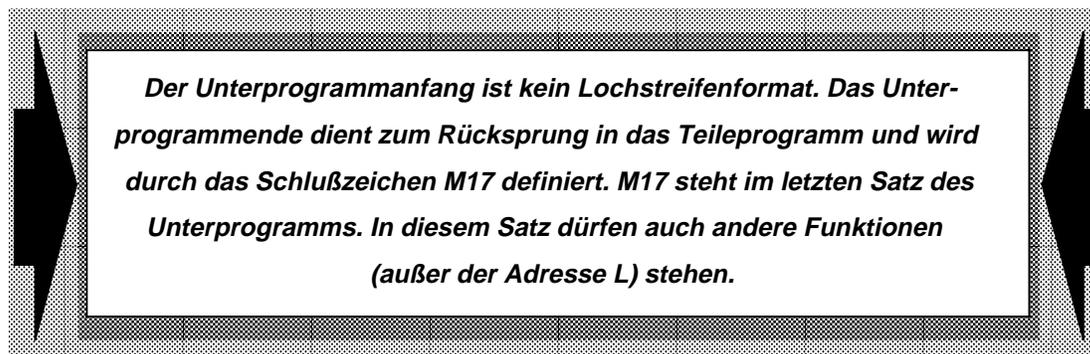
Ein Unterprogramm besteht aus

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



Unterprogramm-Aufbau

Der **Unterprogrammanfang** besteht aus der **Adresse L** und der max. 4stelligen Unterprogrammnummer (siehe Programmschlüssel).



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 UP-Nummer ohne die Adresse P programmiert, wird automatisch eine Durchlaufzeit von P1 (1 Durchlauf) angenommen.

Beispiel:

L123 P1 (1 ... 99)

L123 Unterprogramm (1 ... 9999)

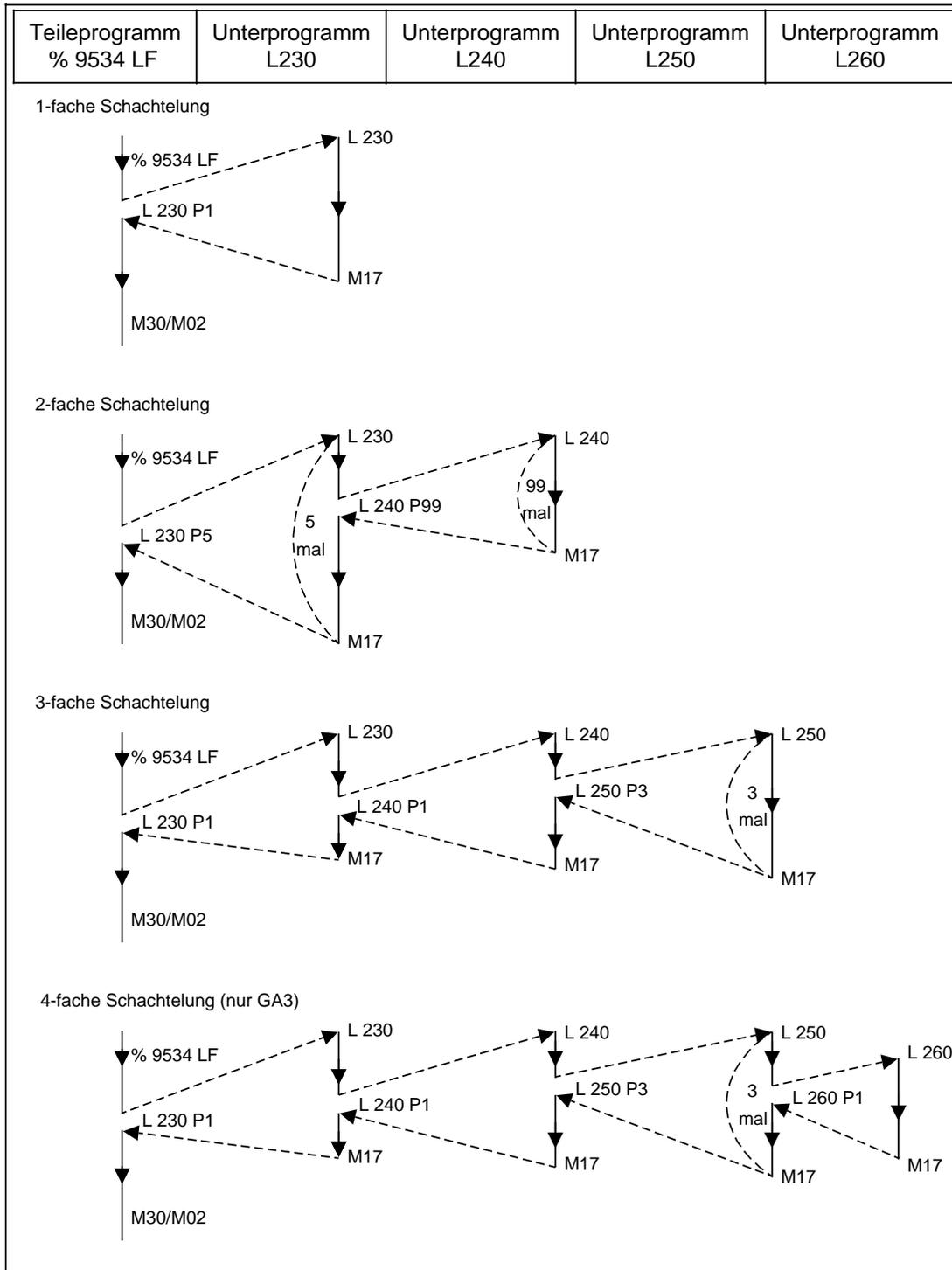
P1 ... Anzahl der Durchläufe

Bei der Programmierung ist folgendes zu beachten:

- Der Unterprogramm-Aufruf darf nicht zusammen mit M02, M17 oder M30 in einem Satz stehen.
- Erfolgt der Unterprogramm-Aufruf bei angewählter Fräserradius-Bahnkorrektur, sind die FRK-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.

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

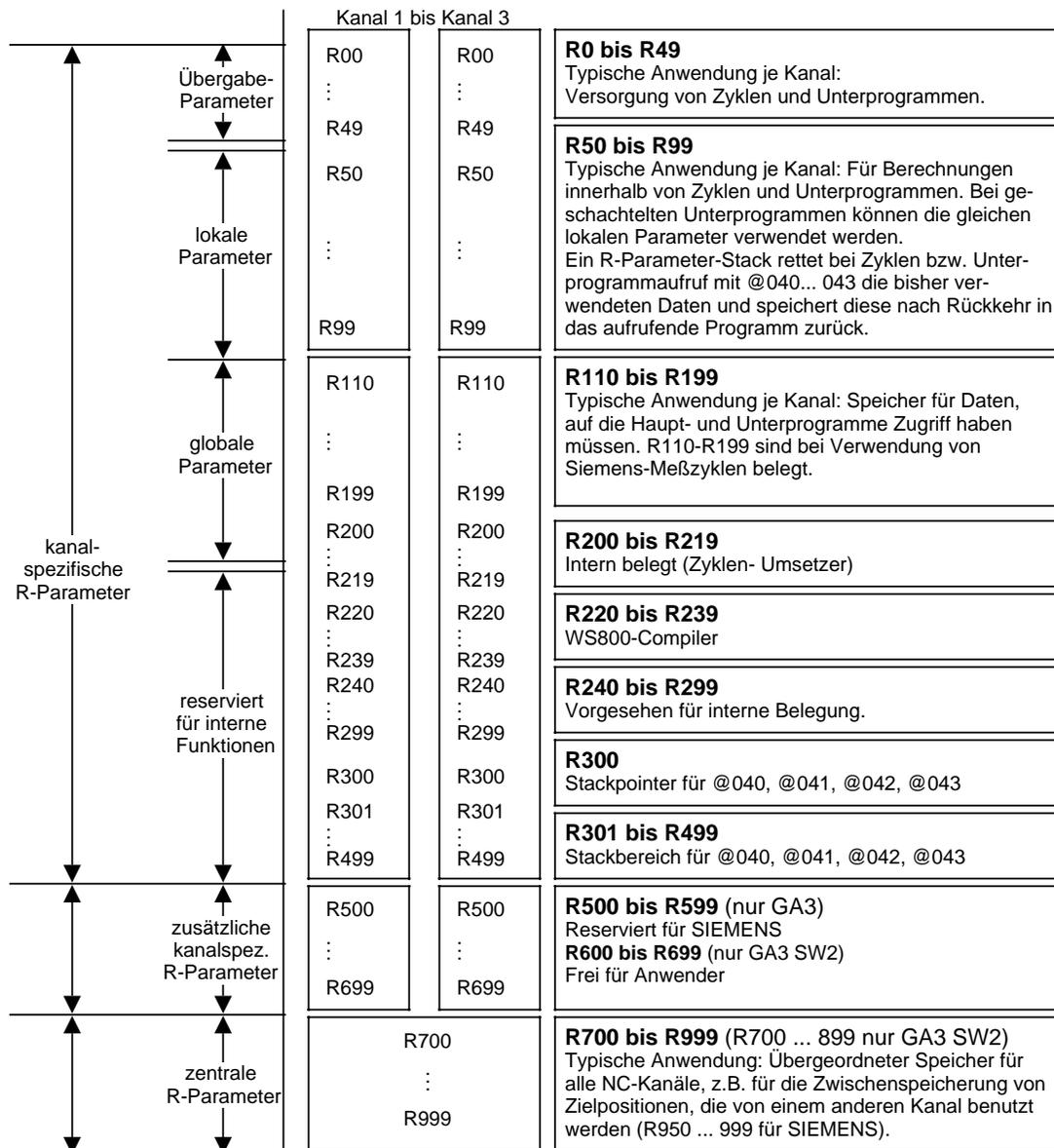
6 Parameter

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 dreistelligen Nummer**. Im Grundausbau stellt die Steuerung je Kanal **700 kanalabhängige** 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 L_F

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. Dabei ist die Reihenfolge Adresse, Zahlenwert, Parameter einzuhalten. Rechenzeichen müssen geschrieben werden. Kein Vorzeichen bedeutet positive Zahl.

Das Rechenzeichen "+" muß immer eingegeben werden.

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

Beispiel:

N35 $R1=9.7 \quad R2=-2.1 \quad \text{L}_F$

N40 $X=20.3+R1 \quad \text{L}_F$

N45 $Y=32.9-R2 \quad \text{L}_F$

N50 $Z=19.7-R1 \quad \text{L}_F$

Ergebnis: $X=30$

$Y=35$

$Z=10$

- Rechnung mit Zahlen und Parametern

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

$$R10=15+R11 L_F$$

6.4 Parameter-Reihe

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

Es sind alle 4 Grundrechenarten in beliebiger Reihenfolge erlaubt. Eine Parameter-Reihe wird durch die **Satzlänge** von **maximal 120 Zeichen** begrenzt und **maximal 10 Parameter** dürfen miteinander verknüpft werden.

Gerechnet wird in folgender Weise:

- | | | | |
|------------------|--------------------|------------------|----------------------------|
| 1. Rechenschritt | $R1 = R2$ | 1. Rechenschritt | $R1 = R2$ |
| 2. Rechenschritt | $R1 = R1 + R3$ | 2. Rechenschritt | $R1 = \underline{R2 + R3}$ |
| 3. Rechenschritt | $R1 = R1 - R4$ | 3. Rechenschritt | $\underline{R1} - R4$ |
| 4. Rechenschritt | $R1 = R1 \cdot R5$ | 4. Rechenschritt | $R1 = R1 \cdot R5$ |
| 5. Rechenschritt | $R1 = R1 / R6$ | 5. Rechenschritt | $\underline{R1 / R6}$ |
| | | | 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.

In einem Satz dürfen beliebig viele Rechenoperationen erfolgen, z. B. Multiplikation, Parameter-Reihe, Addition etc., die maximal zulässige Satzlänge von 120 Zeichen ist jedoch zu beachten. Die Berechnung der einzelnen Verknüpfungen erfolgt in der programmierten Reihenfolge.

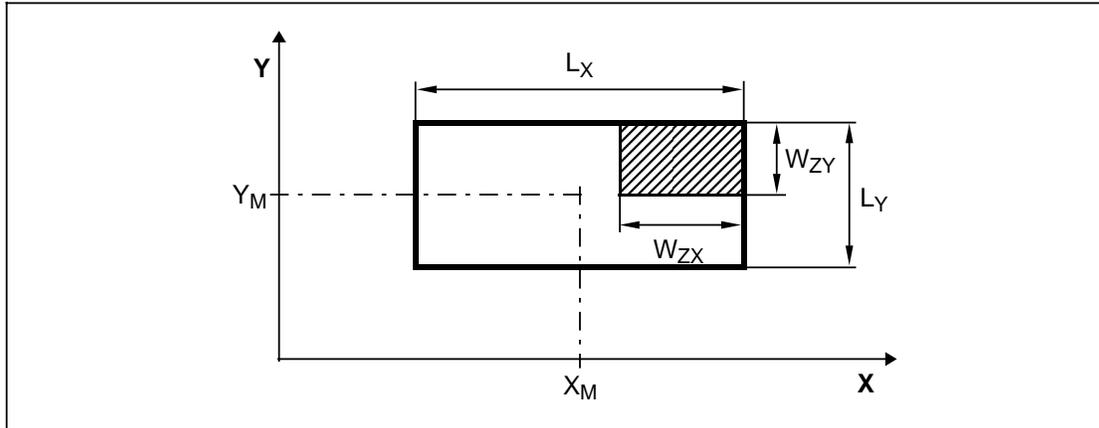
Wertebereich: kleinster Wert: $1 \cdot 10^{-8}$
 größter Wert: 99999999.
 Anzeige: Fließkomma (± 12345678) bis (± 12345678 .)

6.5 Programmbeispiele mit R-Parametern (Nibbelmaschine)

Beispiel:

Zyklus Rechteck zerspanen

Das folgende Unterprogramm gestattet es, in der X-Y-Ebene ein Rechteck mit variablem Seitenverhältnis zu zerspanen, dessen Seiten parallel zu den Maschinenachsen liegen.



$X_M =$ R01 = Mittelpunktposition in X
 $Y_M =$ R02 = Mittelpunktposition in Y
 $L_X =$ R03 = Länge in X
 $L_Y =$ R04 = Länge in Y
 $W_{ZX} =$ R07 = W_Z -Länge X
 $W_{ZY} =$ R08 = W_Z -Länge Y

6.6 Belegung der R-Parametern

Für bestimmte Funktionen sind die nachfolgend beschriebenen kanalspezifischen R-Parameter reserviert:

- R241 Position der ersten Prätze auf der X-Schiene
- R242 Position der zweiten Prätze auf der X-Schiene
- R243 Position der dritten Prätze auf der X-Schiene
- R244 Position der vierten Prätze auf der X-Schiene
- R245 Abmessung der Prätzen in X-Richtung
- R246 Abmessung der Prätzen in Y-Richtung

Die ausführliche Beschreibung dieser Funktionen erfolgt im Anhang Kapitel 12.2.

- R247 Maximaler E-Wert für niedere Exzentergeschwindigkeit
- R248 Maximaler E-Wert für hohe Exzentergeschwindigkeit
- R249 Maximaler Drehwinkel für niedere Exzentergeschwindigkeit
- R250 Maximaler Drehwinkel für hohe Exzentergeschwindigkeit

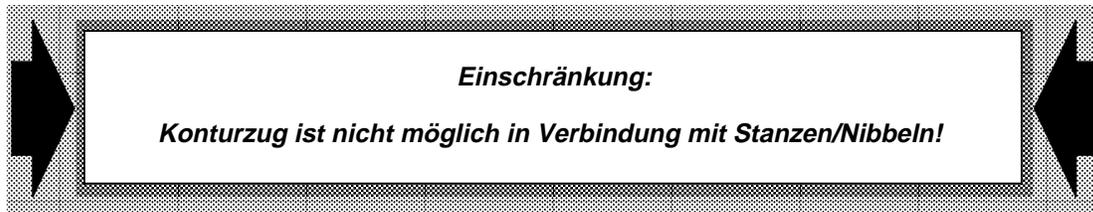
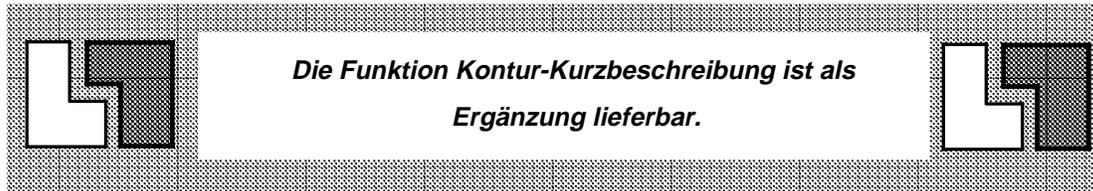
Die ausführliche Beschreibung dieser Funktionen erfolgt im Anhang Kapitel 12.1.

Pattern (Option ab GA3):

- R500 Aktuelles Patternteil in X-Richtung
- R501 Aktuelles Patternteil in Y-Richtung
- R504 Patternteil in Bearbeitung in X-Richtung
- R505 Patternteil in Bearbeitung in Y-Richtung

Die ausführliche Beschreibung dieser Funktionen erfolgt im Anhang Kapitel 11.11.6.

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.

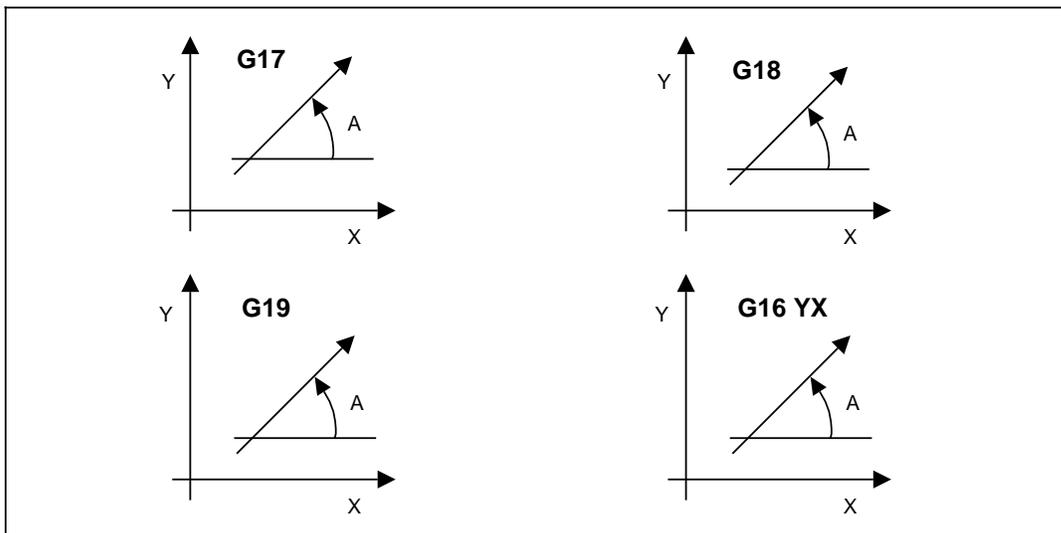
Die einzelnen Geradenzüge können direkt in Form einer Ecke ineinander übergehen, über Radien abgerundet oder angefast werden. Fasen- und Übergangsradien werden nur mit ihrer Größe angegeben. Die Geometrieberechnung übernimmt die Steuerung. Die Programmierung der Endpunktkoordinaten kann im Bezugsmaß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

Winkel (A):

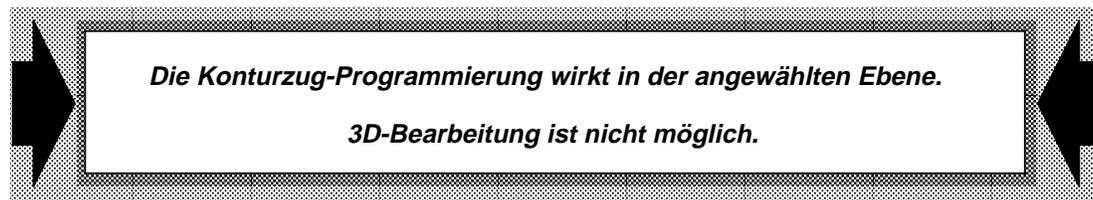
Eingabefinheit 0.00001 entspricht 10^{-5} Grad.

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

Ebenenwahl: Mit G17, G18 oder G19 wird die gewünschte Ebene angewählt.



Bei freier Ebenenanwahl (G16) wird die Ebene durch die programmierten Achsen festgelegt. Die zuerst programmierte Achse ist die Bezugsachse. Der Winkel im rechtsdrehenden Koordinatensystem bezieht sich immer auf die Bezugsachse.

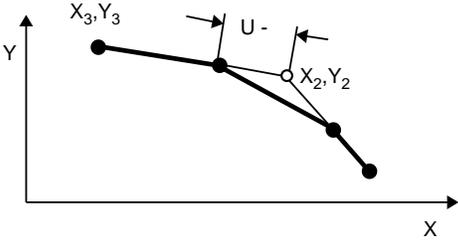
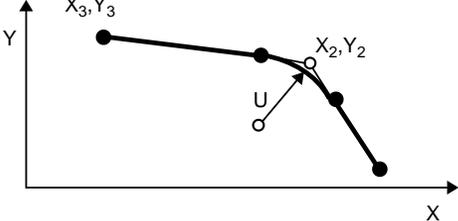
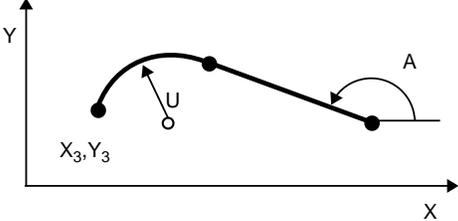
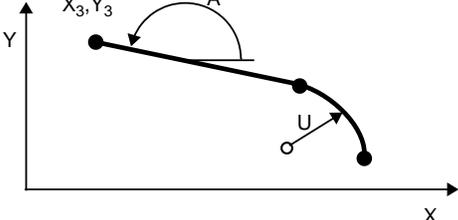
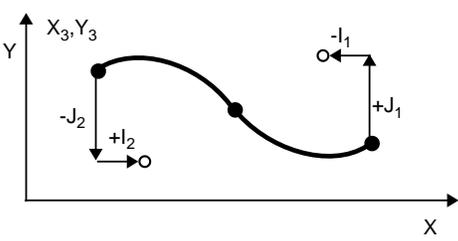


7.2 Konturzug-Programmierung

Die dargestellten Elemente gelten in der angewählten Ebene X-Y (G17).

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 U) über Maschinendaten frei wählbar. Die Adressen dürfen nicht mehrfach vergeben werden.

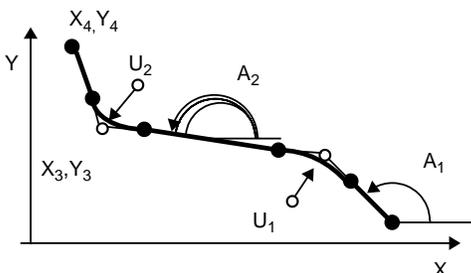
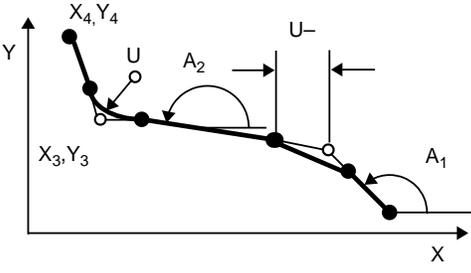
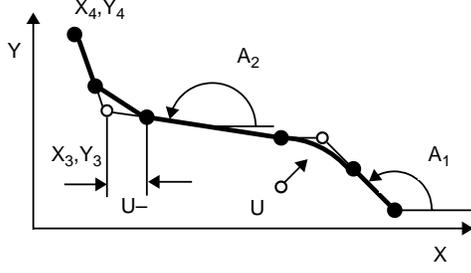
Funktion	Programmierung	Beispiel
(1) 2-Punkte- Zug	N.. A.. X ₂ .. (oder Y ₂) L _F Die zweite Endpunktkoordinate wird von der Steuerung berechnet.	
(2) Kreis- bogen	N.. G02 (oder G03) I.. J.. U.. X ₂ .. (oder Y ₂) L _F Der Kreisbogen ist auf einen Quadranten begrenzt. Die zweite Endpunktkoordinate wird von der Steuerung berechnet. Im Konturzug müssen beide Parameter I und J programmiert werden, auch wenn ein Wert Null ist.	
(3) 3-Punkte- Zug	N.. A ₁ .. A ₂ .. X ₃ .. Y ₃ .. L _F 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	$N.. X_2.. Y_2.. U-.. L_F$ $N.. X_3.. Y_3.. L_F \uparrow$ U- ... bedeutet eine Fase einfügen U ... bedeutet einen Radius einfügen. (Das Zeichen „Minus“ hat hier nicht die Bedeutung als Vorzeichen, sondern ist die Sonderkennung für U als Fase.)	
(5) Radius	$N.. X_2 .. Y_2 .. U.. L_F$ $N.. X_3 .. Y_3 .. L_F \uparrow$ Der eingefügte Radius darf nicht größer sein als die kleinere der beiden Strecken.	
(6) Gerade- Kreisbogen (tangential)	$N.. G02 (oder G03) A.. U.. X_3..$ $Y_3.. L_F$ Kreisbogen nicht über 180 Grad. Die Reihenfolge A (Winkel) und U (Radius) muß eingehalten werden.	
(7) Kreisbogen -Gerade (tangential)	$N.. G02 (oder G03) U.. A.. X_3..$ $Y_3.. L_F$ Kreisbogen nicht über 180 Grad. Die Reihenfolge U, A ist einzuhalten. Kein Radius in X_3, Y_3 einfügbar.	
(8) Kreisbogen- Kreisbogen (tangential)	$N.. G02 (oder G03) I_1.. J_1..$ $I_2.. J_2.. X_3.. Y_3.. L_F$ Kreis 1 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.	

1) Zweiter Satz kann auch ein Konturzug sein.

Funktion	Programmierung	Beispiel
(1)+(4) 2-Punkte- Zug+Fase	N.. A.. X ₂ .. (oder Y ₂ ..) U... L _F N.. X ₃ .. Y ₃ .. L _F 1)	
(1)+(5) 2-Punkte- Zug+Ra- dius	N.. A.. X ₂ .. (oder Y ₂ ..) U.. L _F N.. X ₃ .. Y ₃ .. L _F 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 .. Y ₃ .. U... L _F	
(3)+(5) 3-Punkte- Zug+Ra- dius	N.. A ₁ .. A ₂ .. X ₃ .. Y ₃ .. U... L _F	
(3)+(4) +(4) 3-Punkte- Zug+Fase +Fase	N.. A ₁ .. A ₂ .. X ₃ .. Y ₃ .. U ₁ ... U ₂ ... L _F 1) N... X ₄ .. Y ₄ .. L _F 1) Anfügen einer zweiten Fase am Endpunkt X ₃ , Y ₃ .	

1) Zweiter Satz kann auch ein Konturzug sein.

Funktion	Programmierung	Beispiel
(3)+(5) +(5) 3-Punkte- Zug+Radi- us+Radi- us	N.. A ₁ .. A ₂ .. X ₃ .. Y ₃ .. U ₁ .. U ₂ .. L _F N.. X ₄ .. Y ₄ .. L _F ¹⁾ Anfügen eines zweiten Radius am Endpunkt X ₃ , Y ₃ .	
(3)+(4) +(5) 3-Punkte- Zug+Fase +Radius	N.. A ₁ .. A ₂ .. X ₃ .. Y ₃ .. U ₁ -.. U.. L _F N.. X ₄ .. Y ₄ .. L _F ¹⁾ Anfügen eines Radius am Endpunkt X ₃ , Y ₃ . Der nächste Satz wird jeweils automatisch mit berücksichtigt.	
(3)+(5) +(4) 3-Punkte- Zug+Ra- dius+Fase	N.. A ₁ .. A ₂ .. X ₃ .. Y ₃ .. U.. U-.. L _F N.. X ₄ .. Y ₄ .. L _F ¹⁾ Anfügen einer Fase U- am Endpunkt.	

Bei Ecken, in die keine Fase oder Radius eingefügt werden soll, ist U0 zu programmieren, wenn im Konturzug noch ein Radius oder eine Fase folgt.

Bei dieser Programmierung wird von der Steuerung ein Satz mit Wegstrecke=0 generiert. Dieses muß bei Wirksamkeit der FRK beachtet werden. U-0 wird wie U0 interpretiert.

Ein Radius oder eine Fase kann nur zwischen zwei Linesätzen eingefügt werden.

Die Reihenfolge der Adressen A, X, Y, U, 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

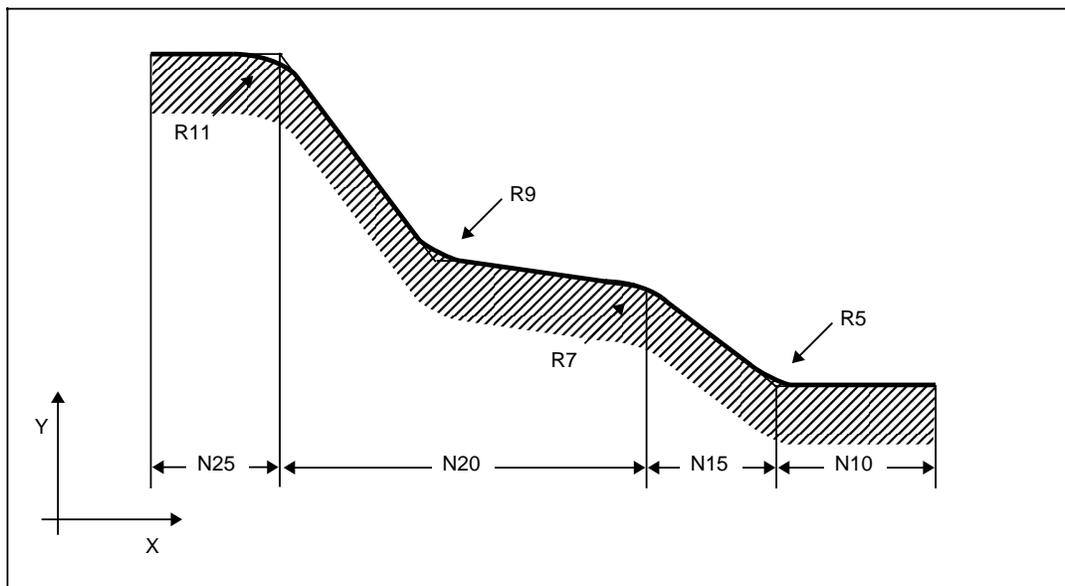
Wird in einem Konturzugsatz G09 programmiert, wirkt dies erst am Ende des Satzes, also mit Erreichen der Endposition. Innerhalb des Konturzuges wird bei Unstetigkeitsstellen (Ecken, Kanten) automatisch G09 von der Steuerung generiert.

- 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; kleinster Verrundungsradius R... Fräserradius.

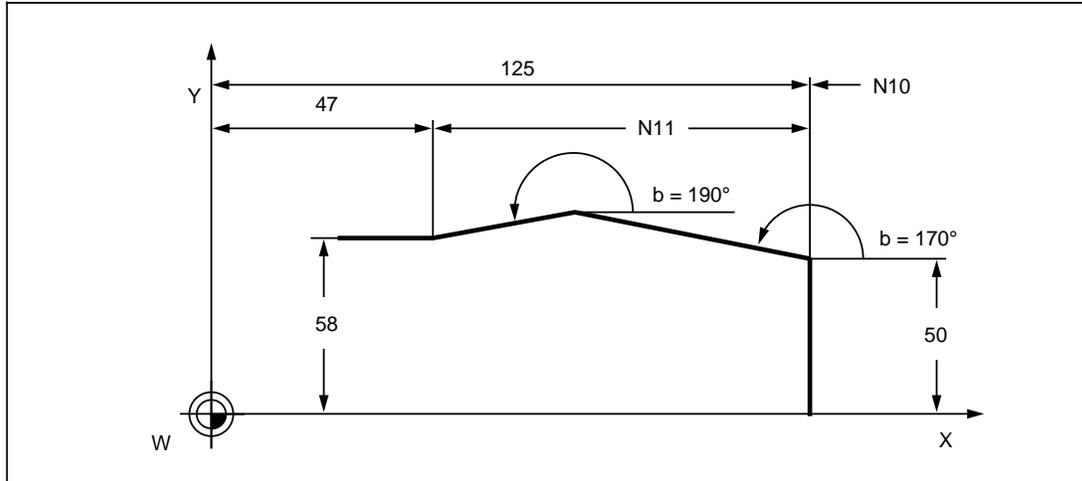
Beispiel: Kettung von Sätzen:



N10	X..	U5	L _F	Gerade mit Radius				
N15	A..	Y..	U7	L _F	Gerade mit Radius			
N20	A1	A2	X..	Y..	U9	U11	L _F	3-Punkte-Zug mit Radius an beiden Enden
N25	X..	L _F						Gerade

7.5 Programmierbeispiel Laserschneiden

Der Winkel a bezieht auf den Anfangspunkt, der Winkel b 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: Laserschneiden

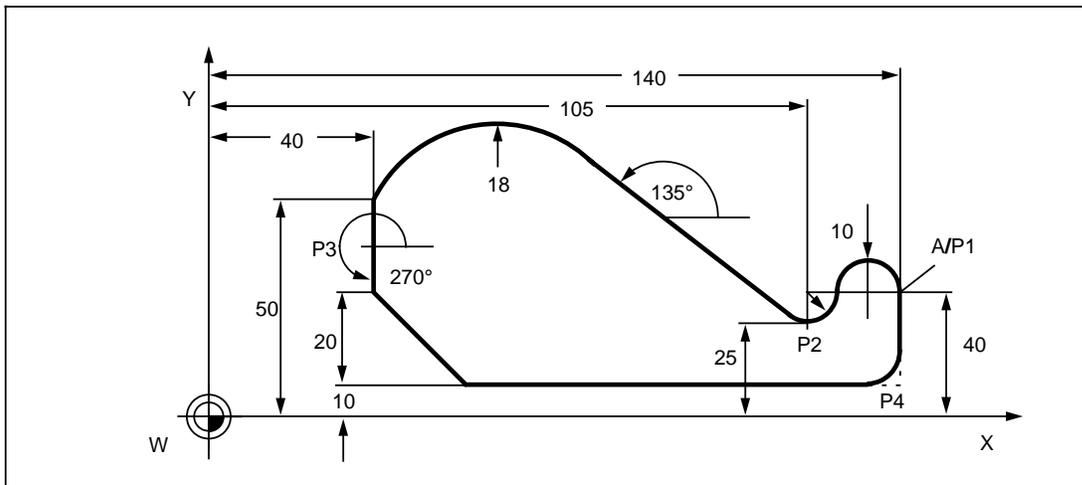
```

:
:
N10 G00 G90 X125 Y50 L_F
N11 G01 A170 A190 X47 Y58 F.. L_F
:
:

```

Beispiel: Laserschneiden mit Konturzugprogrammierung

Im folgenden Beispiel werden die Konturzüge: Kreisbogen-Kreisbogen, Gerade-Kreisbogen und ein 3 Punktezug+Fase+Radius verwendet.

Beispiel: Laserschneiden**A=Startpunkt**

```

L 168 L_F
N1 G90 G03 I-10 J0 I0 J15 X105 Y25 L_F (P2)
N2 G03 A135 U18 X30 Y50 L_F (P3)
N3 G01 A270 A0 X140 Y10 U-20 U10 L_F (P4)
N4 Y40 L_F (P1)
N5 M17 L_F

```

8 Werkzeugkorrekturen

8.1 Werkzeugdaten

Unter einer Werkzeug-Korrekturnummer D sind die geometrischen Werkzeugdaten abgelegt: Länge \pm 9999,999 mm, Radius \pm 999,999 mm; T-Nr. zwei Dekaden (Eingabefinheit 1 μ m).

Im Werkzeug-Korrekturspeicher 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:

	P0:	Werkzeugnummer
	P1:	Werkzeugtyp
	P2:	L1 Geometrie/Werkzeugabmessung in X-Richtung
	P3:	L2 Geometrie/Werkzeugabmessung in Y-Richtung
	P4:	Durchmesser/Radius
D1 ... D99	P5:	L1 Verschleiß
	P6:	L2 Verschleiß
	P7:	Durchmesser/Radius
	P8:	L1 Basis/Pratzenschutzradius
	P9:	L2 Basis

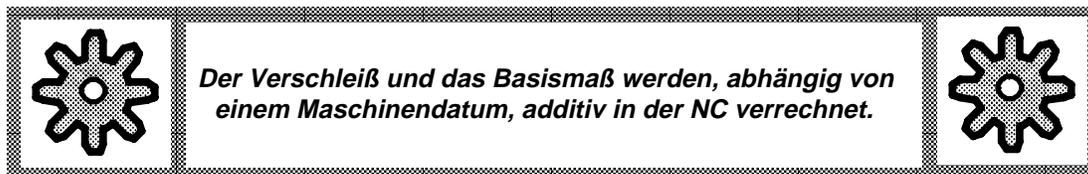
- Die Struktur des Werkzeugkorrektur-Blockes wird durch den Werkzeugtyp (P1) gekennzeichnet.

Aufschlüsselung des Werkzeugtyps P1:

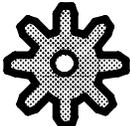
Typ 0	Werkzeug nicht definiert
Typ 1 ... 9	Drehwerkzeuge, Lage der Werkzeugschneide
Typ 10 ... 19	Werkzeuge nur mit wirksamer Längenkorrektur (z. B. Bohrer)
Typ 20 ... 29	Werkzeug mit Radius- und einer Längenkorrektur (z. B. Fräser-, Laser-, Plasmastrahl)
Typ 30 ... 39	Werkzeuge mit Radius und zwei Längenkorrekturen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

- Der Aufruf der Werkzeugkorrektur für die Fräserradiuskorrektur erfolgt max. zweidekadig mit D1 bis D99.
- Die Abwahl der Werkzeugkorrektur erfolgt mit D0.

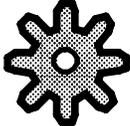
Die Korrektur wird erst ausgeführt, wenn die entsprechende Achse programmiert wird.



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).



**Über Maschinendatum kann bestimmt werden, daß
Werkzeugtyp 0 wie Werkzeugtyp 20 wirkt. Radius- und
Längenkorrektur wird dann wie bei Typ 20 zugeordnet.
Die Anzeige Typ 0 bleibt unverändert.**



8.2 Aufbau des Werkzeugkorrekturspeichers

	SIN 810N	P-Nr	SIN 810M
Dn	Werkzeugnummer	0	Werkzeugnummer <input style="width: 80px;" type="text"/>
	Werkzeugtyp	1	Werkzeugtyp <input style="width: 80px;" type="text"/>
	Wz.-Abmessung in X	2	L1 Geometrie <input style="width: 80px;" type="text"/>
	Wz.-Abmessung in Y	3	L2 Geometrie
	Durchmesser/Radius	4	Durchmesser/Radius
		5	L1 Verschleiss
		6	L2 Verschleiss
		7	Durchmesser/Radius
	Pratzenschutzradius	8	L1 Basis
		9	L2 Basis

Dn Korrekturspeichernummer

Die Werkzeugnummer T entspricht der D-Nummer des Korrekturspeichers. Somit braucht also nur die T-Nummer programmiert werden.

Wertebereich: n=1 ... 99

Beispiel:

Programmierung der T-Nummer 3 aktiviert den Korrekturspeicher D3.

Hinweis:

Bei Verwendung der Fräserradiusbahnkorrektur (FRK) muß die Anwahl des Korrekturspeichers mit D erfolgen! (siehe Kapitel 8.3)

Die FRK wird z. B. bei Laser-/Plasmaschneiden etc. zur Kompensation des Strahldurchmessers eingesetzt.

P0 Werkzeugnummer

Hier ist die Eingabe einer maximal 8stelligen Werkzeugidentnummer zur Identifikation des Werkzeugs möglich.

P1 Werkzeugtyp

Eine Auswertung des Werkzeugtyps erfolgt nur für die Fräserradiusbahnkorrektur (siehe P4 Durchmesser). Für die Korrektur eines Laser- und Plasmastrahls wird hier der Wert 20 (Fräser) eingetragen.

P2 Werkzeugabmessung in X-Richtung

Eingabe der Abmessung in X-Richtung bei rechteckigen und quadratischen Stanzwerkzeugen; bei runden Werkzeugen wird hier der Durchmesser eingegeben.

P3 Werkzeugabmessung in Y-Richtung

Eingabe der Abmessung in Y-Richtung bei rechteckigen und quadratischen Stanzwerkzeugen; bei runden Werkzeugen wird hier die Null eingetragen.

P2 und P3 sind zu belegen, wenn die Standard-Stanz-/Nibbelzyklen verwendet werden (siehe Kapitel 10 nur bei GA2). Siehe Dokumentation des Maschinenherstellers.

P4 Durchmesser/Radius

Hier erfolgt die Eingabe des Werkzeugradius des Schneidwerkzeugs beim Laser-/Plasma-schneiden für die Fräserradiusbahnkorrektur. Die Berechnung erfolgt analog wie für Fräser bei SINUMERIK 810M.

Standardmäßig ist Eingabe als Radius aktiv. Soll die Eingabe im Durchmesser erfolgen, muß dies über Maschinendatum projiziert werden.

Beispiele:

Laserstrahl	Radius 0.5 mm	Eingabe 0.5 in P4
Plasmastrahl	Radius 1.5 mm	Eingabe 1.5 in P4

P8 Pratzenschutzradius

Bei Verwendung des "Werkzeugspezifischen Pratzenschutzes" muß hier der Pratzenschutzradius (siehe Kapitel 12.2.1.2) eingetragen werden.

8.3 Schnittpunkt-Fräserradius-Bahnkorrektur G40/G41/G42

G40	Keine Schnittpunkt-Fräserradius-Bahnkorrektur
G41	Werkzeug in Bewegungsrichtung links vom Werkstück
G42	Werkzeug in Bewegungsrichtung rechts vom Werkstück

Die Korrektur des Fräserradius wirkt in der angewählten Ebene (G16 bis G19).

An- und Abwahl der Schnittpunkt-Fräserradius-Bahnkorrektur

Die Anwahl kann nur bei Wirksamkeit von G00 oder G01 erfolgen. G40, G41, G42 können in einem Satz ohne Wege programmiert werden. Sie werden jedoch erst wirksam, wenn in mindestens einer Achse eine Bewegung programmiert ist.

Beispiel für die Anwahl:

```
N10 G01 (G17)1) G41 D07 X.. Y.. LF
```

N10 Am Ende dieses Satzes ist die korrigierte Bahn in der angewählten Ebene erreicht. Es wird der Radiuskorrekturwert eingerechnet.

1) Wird als Grundstellung für die Ebenenanwahl (G16) die X-Y-Ebene projiziert, braucht die Funktion G17 nicht programmiert werden. Die Grundstellung wird über Maschinendaten festgelegt.

oder

```
N10 (G17)1 LF
N15 G41 D07 LF
N20 G01 X.. Y.. LF
```

N10 Ebenenanwahl in der X-Y-Ebene

N15 Korrekturanwahl

N20 Am Ende dieses Satzes ist die korrigierte Bahn in der angewählten Ebene erreicht. Es wird nur der Radiuskorrekturwert eingerechnet.

Mit G40 wird die Fräserkorrektur (G41 oder G42) in Linearsätzen (G00, G01) abgewählt. Damit die entsprechenden Korrekturen wirksam werden, muß mindestens eine Achse der angewählten Ebene programmiert werden. Die Längenkorrektur wird mit D0 abgewählt und wirksam, wenn die zur Längenkorrektur zugehörige Achse programmiert wird.

Beispiel für die Abwahl:

```
N30 G40 X.. LF
```

N30 Abwahl der Werkzeugkorrektur. Es wird nur der Radiuskorrekturwert wirksam.

oder

```
N30 G41 D0 X.. LF
```

N30 Abwahl aller Korrekturen durch Korrekturwerte=0. Es wird nur der Radiuskorrekturwert wirksam.

Vor Anwahl einer anderen Ebene muß die Fräserradius-Bahnkorrektur abgewählt werden.

Wechsel von G41 auf G42

```
N10 G01 (G17)1) G41 D12 X.. Y.. LF
N20 G42 X.. Y.. LF
```

N10 Einrechnung der Radiuskorrektur

N20 Radiuskorrektur geändert (z. B. Änderung der Bewegungsrichtung zum Werkstück; Mäandern)

Wechsel der Werkzeugkorrekturnummer

Die G-Funktion muß nicht neu eingegeben werden.

```
N10 G01 (G17)1) G41 D12 X.. Y.. LF
N20 D10 .. LF
N25 X.. Y.. LF
```

N20 Änderung der Werkzeugkorrekturnummer

N25 Änderung der FRK

Bei angewählter FRK darf kein G58, G59 programmiert werden.

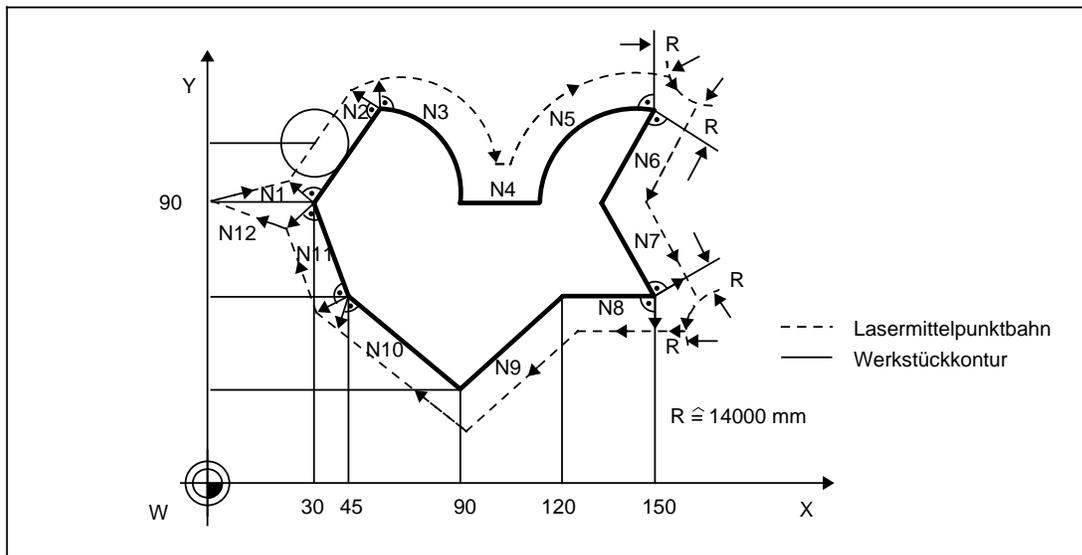
Abhilfe: Die Funktionen (G58, G59) vor Anwahl der FRK programmieren oder die FRK abwählen – G58, anwählen – und anschließend die FRK wieder anwählen.

Bei angewählter FRK, einschließlich des G40-Satzes, darf der wirksame NV-Wert nicht geändert werden.

1) Wird als Grundstellung für die Ebenenanwahl (G16) die X-Y-Ebene projiziert, braucht die Funktion G17 nicht programmiert werden. Die Grundstellung wird über Maschinendaten festgelegt.

Beispiel:

Lasern mit Fräserradiuskorrektur (nur Bewegungssätze)



```

N1 G01 G41 D1 G90 (G17)1) X30 Y90 F500 LF
N2 G91 X30 Y30 LF
N3 G02 X30 Y-30 I0 J-30 LF
N4 G01 X30 LF
N5 G02 X30 Y30 I30 J 0 LF
N6 G01 X-15 Y-30 LF
N7 X15 Y-30 LF
N8 X-30 LF
N9 X-30 Y-30 LF
N10 X-45 Y30 LF
N11 X-15 Y30 LF
N12 G40 G90 X0 Y 90 LF
N13 ..

```

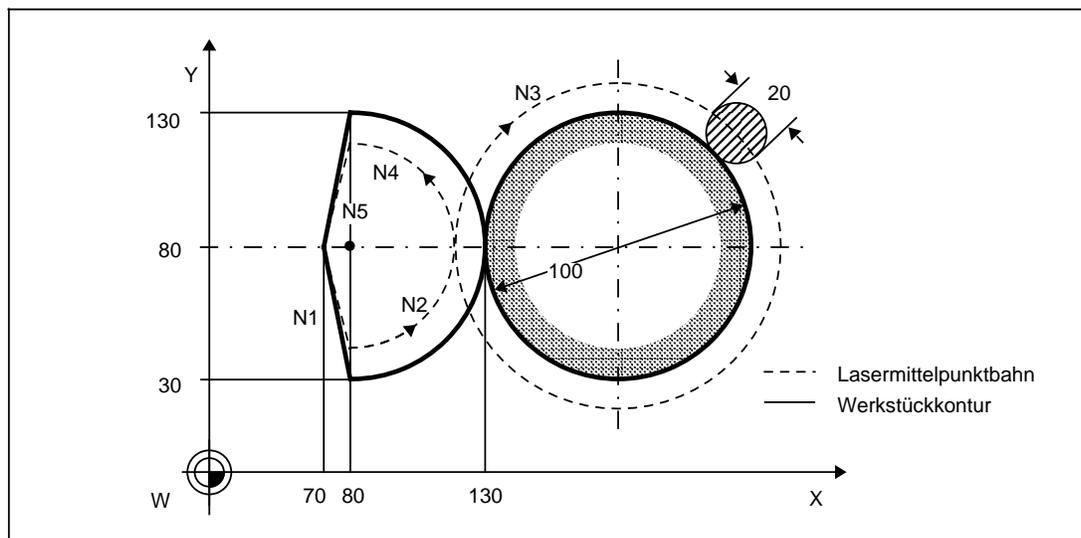
In diesem Beispiel hat der verwendete Laserstrahl zur besseren Darstellbarkeit einen Radius von 14 mm.

Unter der Werkzeugkorrekturnummer D1 muß der Laserradius bei P4 eingegeben werden.

1) Wird als Grundstellung für die Ebenenanwahl (G16) die X-Y-Ebene projiziert, braucht die Funktion G17 nicht programmiert werden. Die Grundstellung wird über Maschinendaten festgelegt.

Beispiel:

Vollkreisprogrammierung mit Fräserradiuskorrektur (nur Bewegungssätze)



```

N0 G90 G00 (G17)1) G41 D1 X70 Y80 F500 LF
N1 X80 Y30 LF
N2 G03 X130 Y80 I0 J50 LF
N3 G91 G02 X0 Y0 I50 J0 LF
N4 G90 G03 X80 Y130 I-50 J0 LF
N5 G00 G40 X70 Y80 LF

```

In diesem Beispiel hat der verwendete Laserstrahl zur besseren Darstellbarkeit einen Radius von 10 mm. Unter der Werkzeugkorrekturnummer D1 muß der Laserradius bei P4 eingegeben werden.

1) Wird als Grundstellung für die Ebenenanwahl (G16) die X-Y-Ebene projiziert, braucht die Funktion G17 nicht programmiert werden. Die Grundstellung wird über Maschinendaten festgelegt.

8.4 Werkzeug-Längenkorrektur, positiv oder negativ

Bei Anwahl der Längenkorrektur ist eine positive Korrektur wirksam. **Negative** Werkzeuglängenkorrektur ist **mit G16** programmierbar. Die WZ-Längenkorrektur ist in der senkrecht auf der FRK-Ebene stehenden Achse wirksam.

Beispiel:

G16 U V±W

U, V Ebenenanwahl (FRK-Ebene)

±W Vorzeichen sagt aus, ob die WZ-Längenkorrektur der Achse W positiv oder negativ wirksam ist.

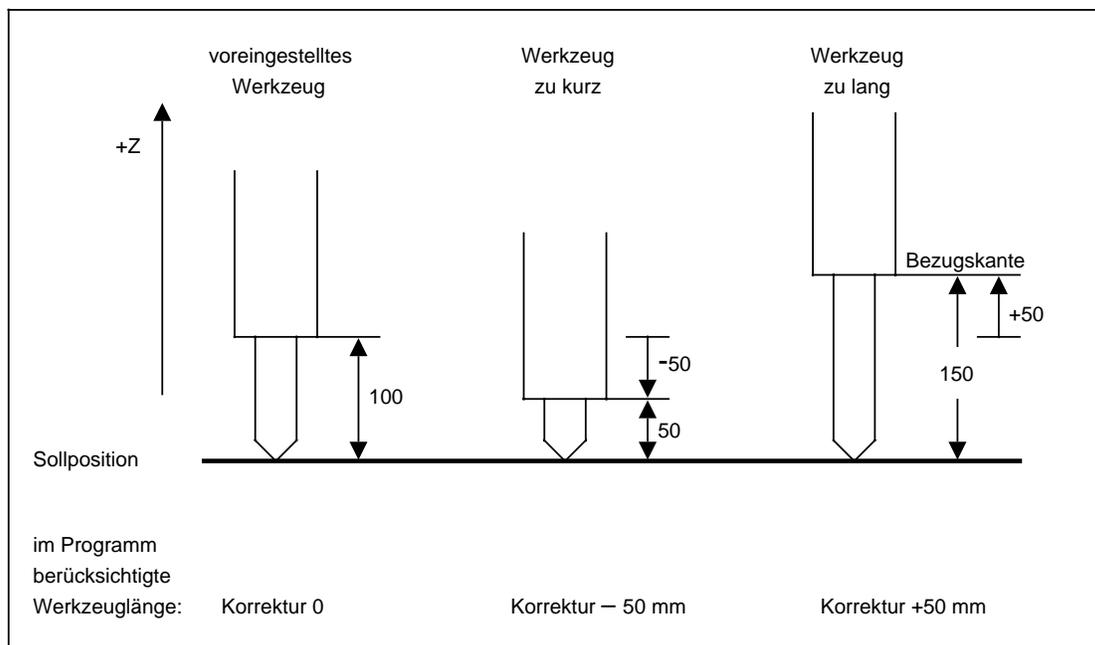
Festlegung des Vorzeichens

Als Vorzeichen wird vom Bediener ein **Pluszeichen** eingegeben, wenn der **tatsächliche Wert** des Werkzeugs **größer** ist als der vom Programmierer berücksichtigte Wert. Ein Minuszeichen wird eingegeben, wenn der tatsächliche Wert des Werkzeugs kleiner ist als der vom Programmierer berücksichtigte Wert.

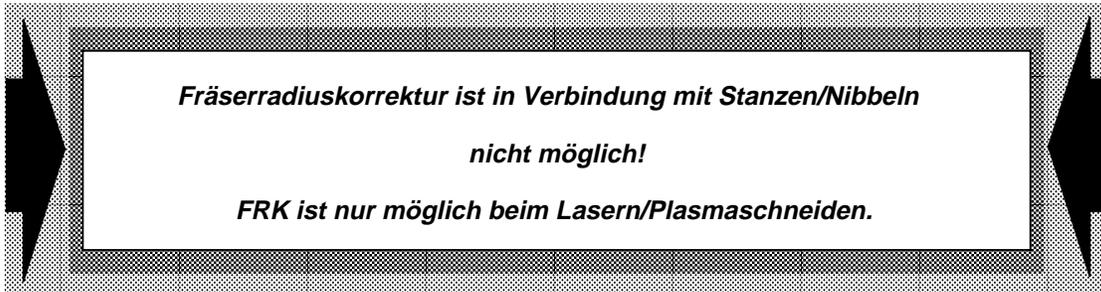
Beispiel:

Der verwendete Bohrer ist **länger** als der programmierte Bohrer: +Korrekturwert

Der verwendete Bohrer ist **kürzer** als der programmierte Bohrer: – Korrekturwert



9 Fräserradiuskorrektur (FRK)



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.

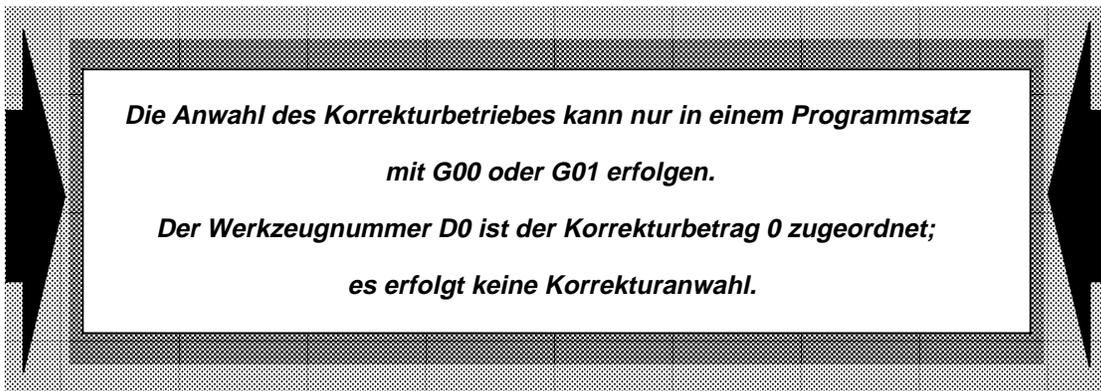
Konturelemente sind mit dicker Linie  gekennzeichnet.

9.1 Anwahl der FRK

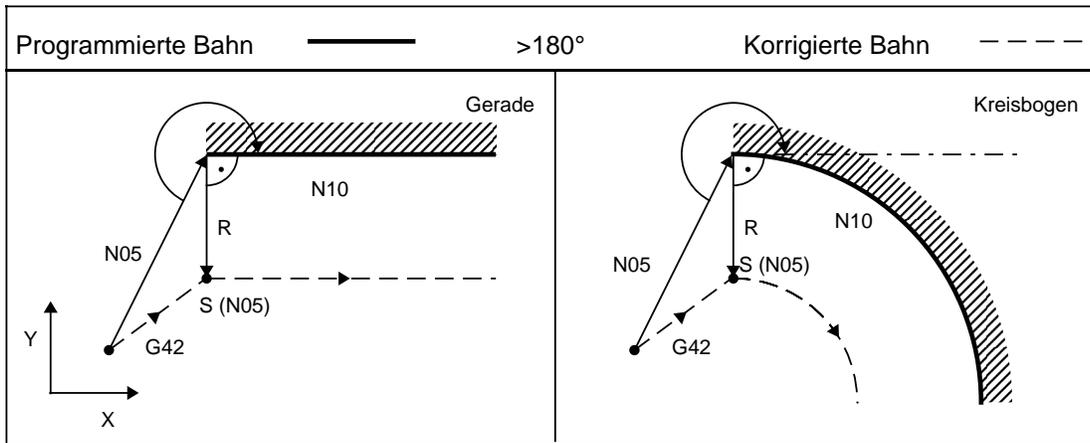
Die Anwahl des Korrekturbetriebes erfolgt in der festgelegten Ebene mit den Wegbedingungen G41/G42 und Angabe der Korrekturnummer D.

Hierbei wird die Korrektur bei **G41 links** (in Verfahrrichtung) und bei **G42 rechts** (in Verfahr-
richtung) von der Werkstückkontur durchgeführt.

Bei der Anwahl der FRK werden immer zwei bzw. drei Programmsätze zur Schnittpunkt-
berechnung eingelesen.

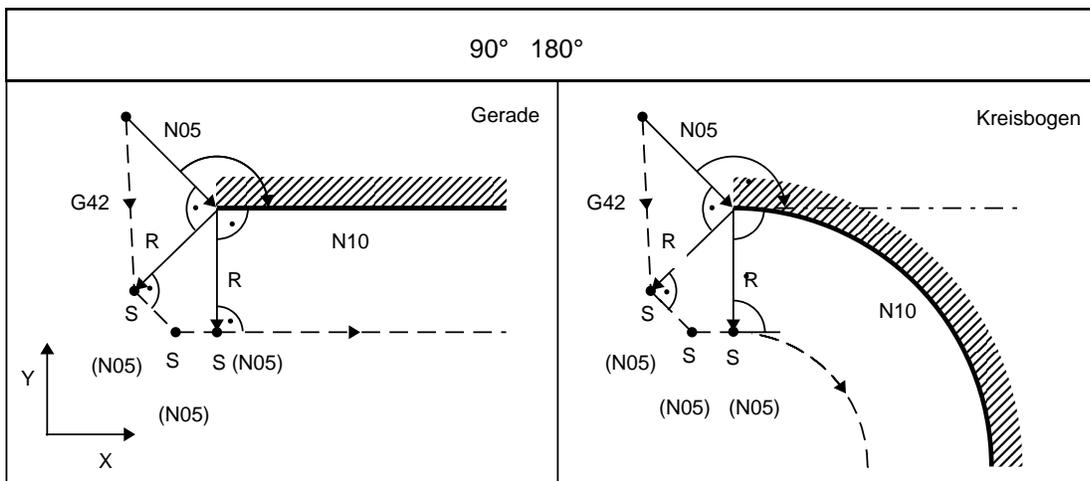


Die folgenden Bilder zeigen die Korrekturanwahl bei verschiedenen Anfahrwinkeln. Die Bilder
sind mit G42 dargestellt. Bei Programmen mit G41 gilt als Winkelübergang $\beta=360^\circ - \alpha$.

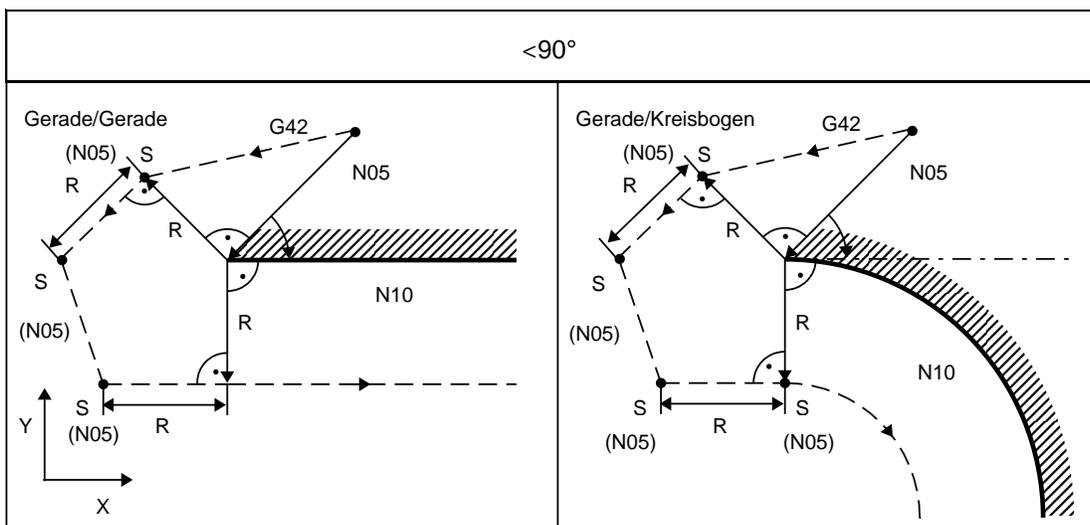


Anwahl des Korrekturbetriebes bei >180°

Anwahl des Korrekturbetriebes



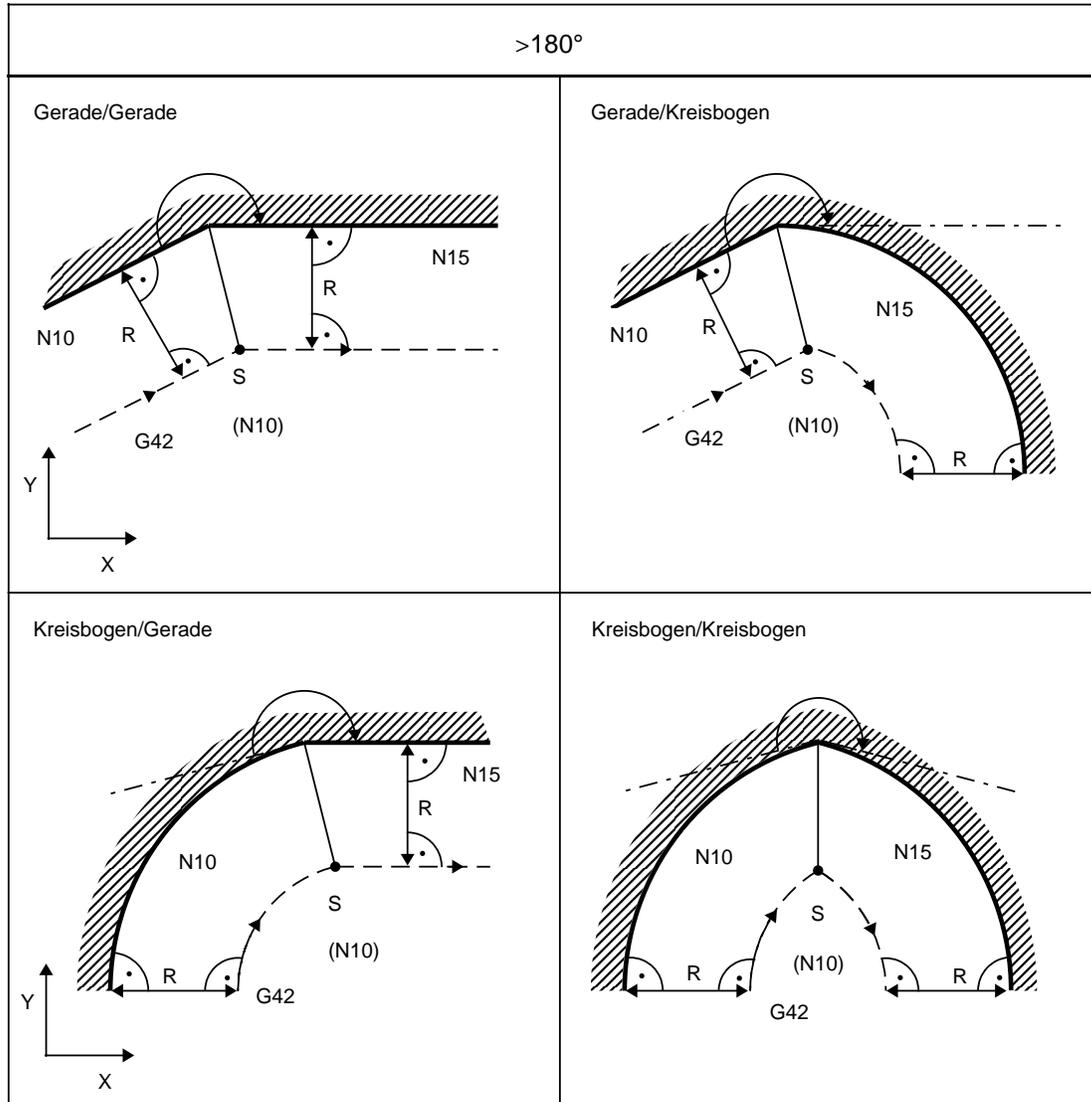
Anwahl des Korrekturbetriebes bei 90° 180°



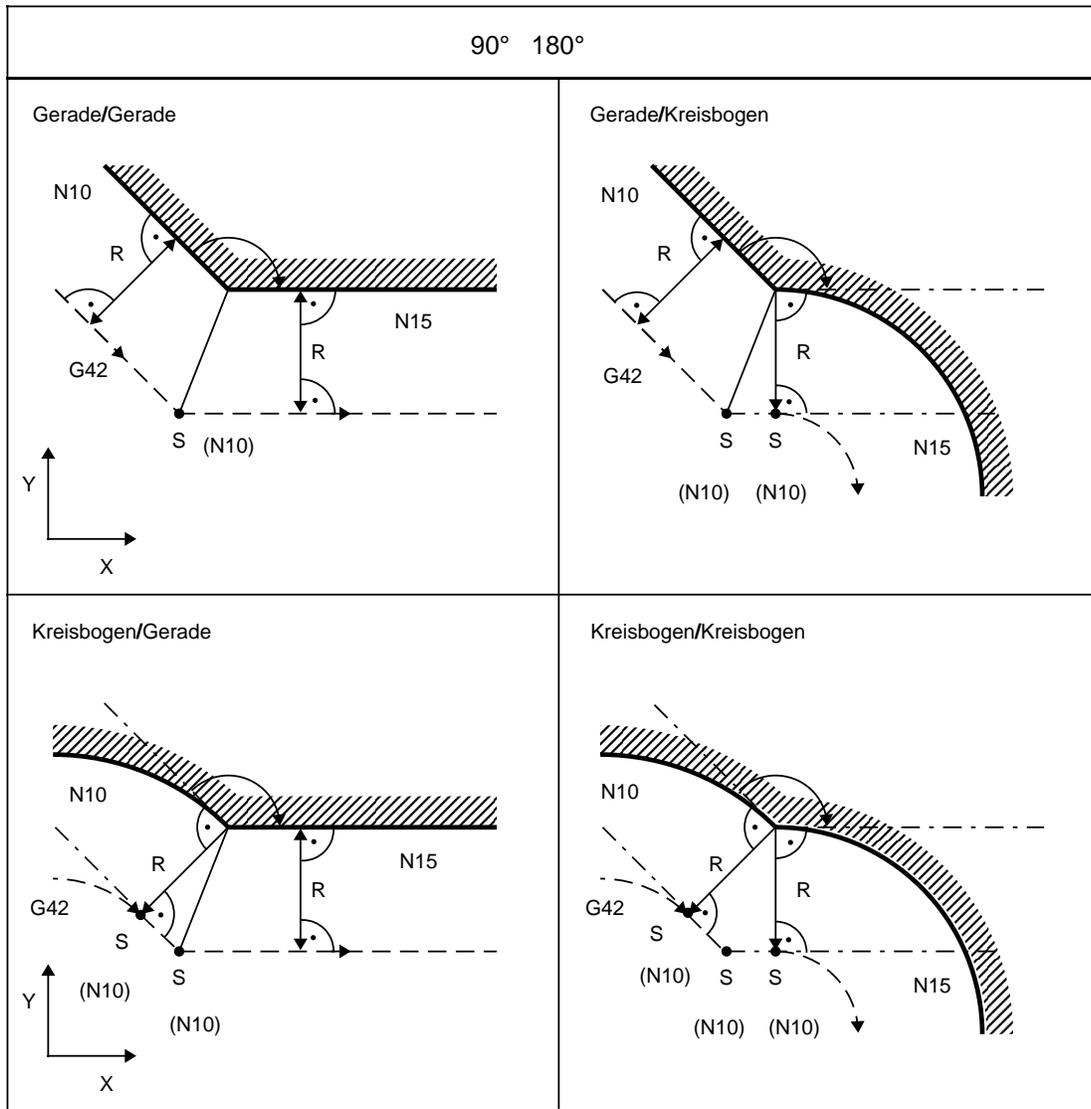
Anwahl des Korrekturbetriebes bei <90°

9.2 FRK im Programm

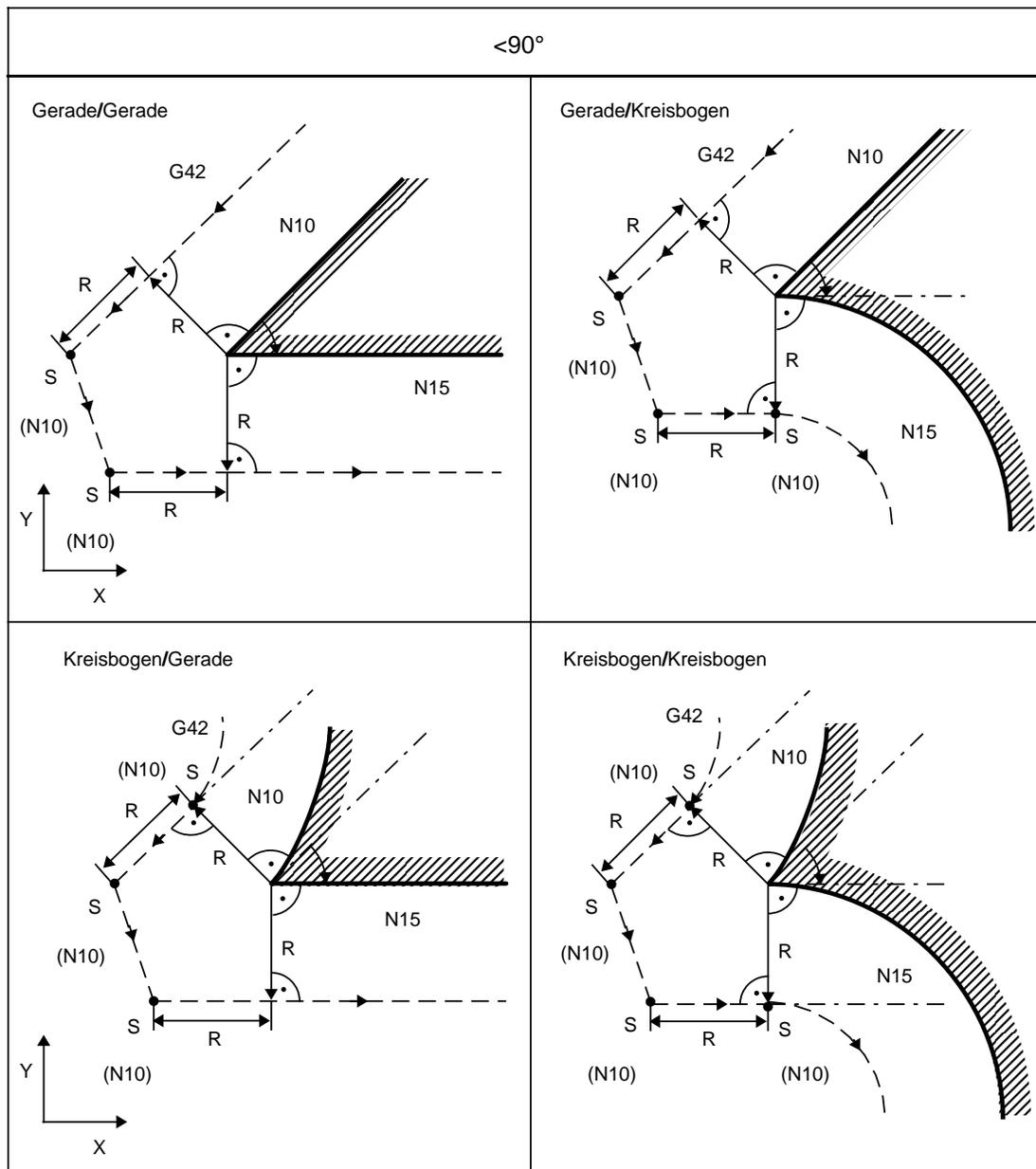
Bei der FRK werden von der Steuerung bereits während der **Abarbeitung des aktuellen Satzes** zwei weitere Sätze im voraus eingelesen und der Schnittpunkt der **korrigierten Bahnen berechnet**. Den Korrekturbetrieb bei **verschiedenen Übergängen** zeigen die folgenden Bilder.



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



FRK bei verschiedenen Übergängen bei 90° 180°

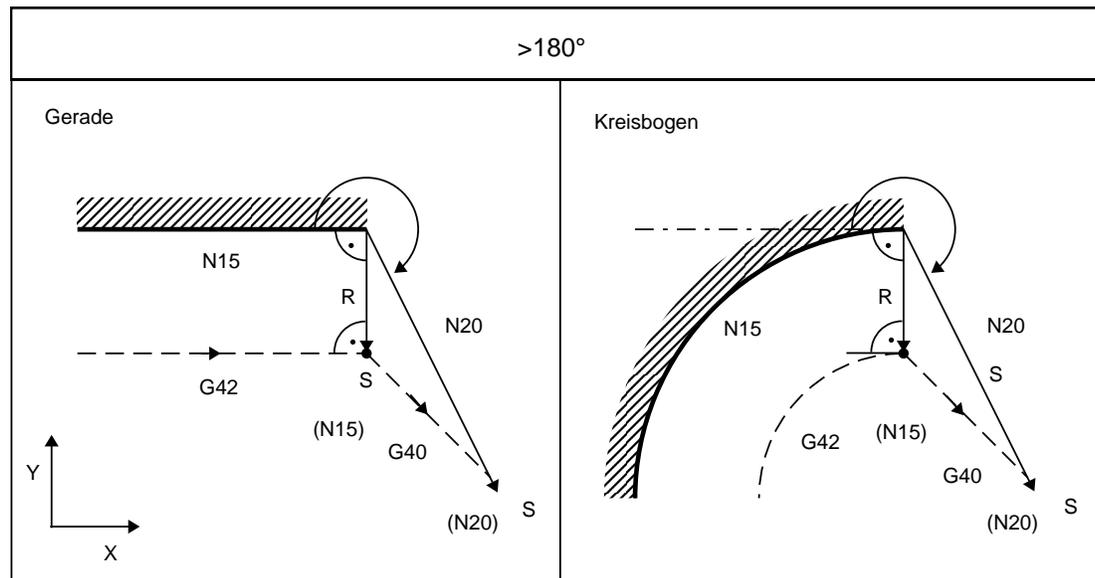
FRK bei verschiedenen Übergängen bei $<90^\circ$

9.3 Abwahl der FRK (G40)

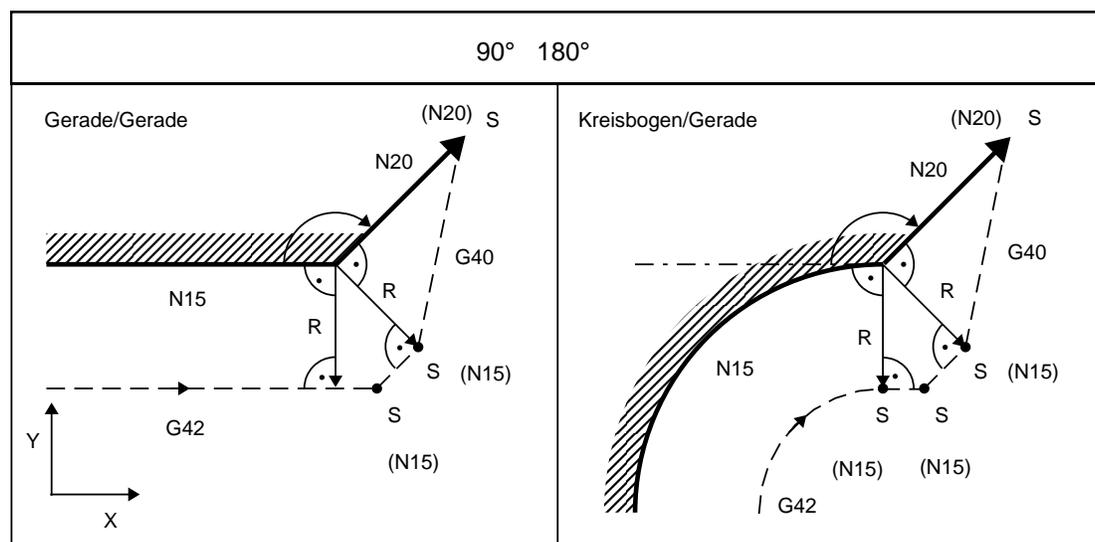
Die **Abwahl** des Korrekturbetriebes erfolgt mit der Wegbedingung **G40**.

**Die Abwahl des Korrekturbetriebes kann nur in einem Programmsatz mit G00 oder G01 erfolgen.
 Der Werkzeugnummer D0 entspricht der Korrekturbetrag 0.
 Der Korrekturbetrieb kann damit ebenfalls abgewählt werden.**

Abwahl des Korrekturbetriebes



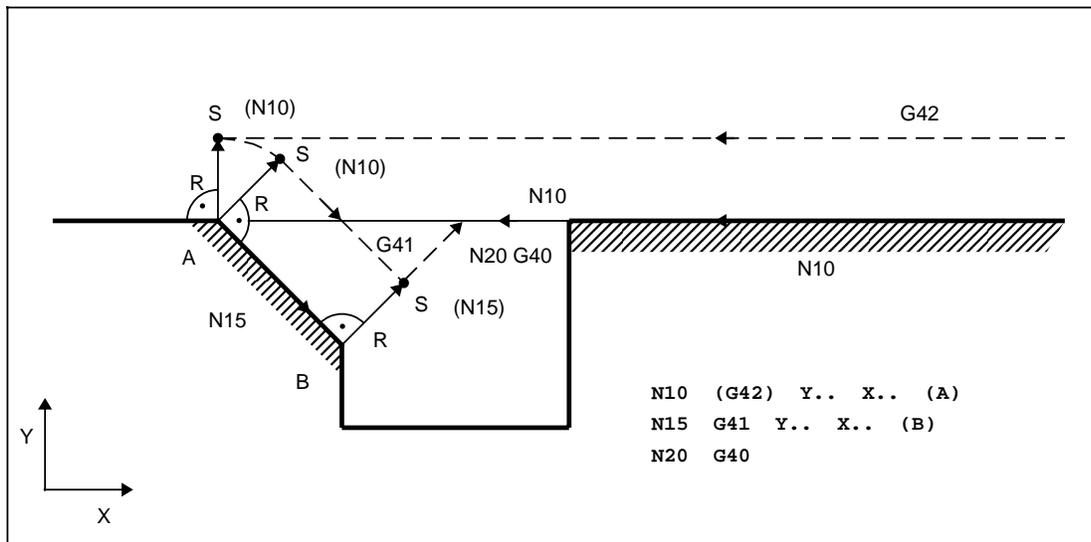
Abwahl des Korrekturbetriebes bei $>180^\circ$



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

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 z. B. G41 wird ein senkrechter **Vektor** mit der Länge **R** in der entsprechenden **Korrekturrichtung** errichtet.

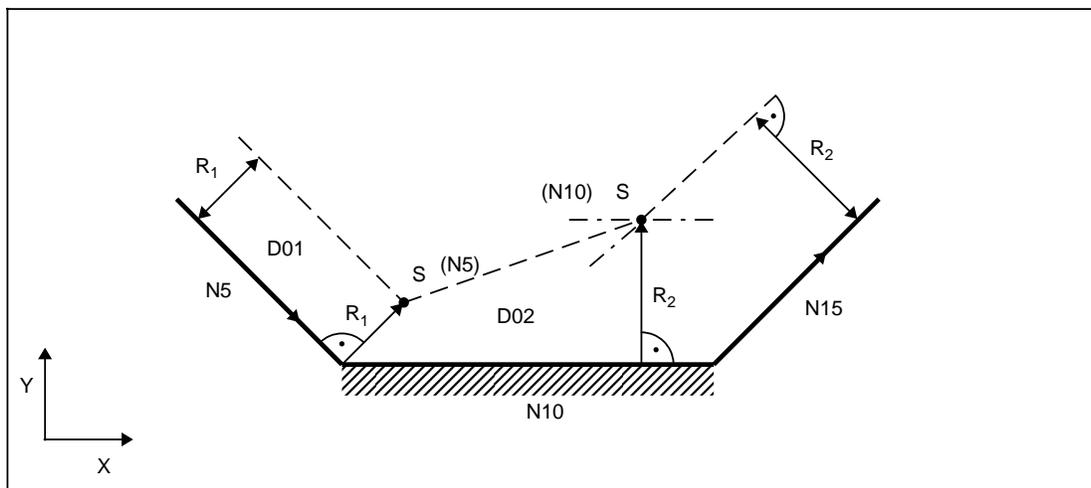


Wechsel der Korrekturrichtung

9.5 Wechsel der Korrekturnummer (D..)

Bei Wechsel der Korrekturnummer (G41 D.., G41 D..) gilt:

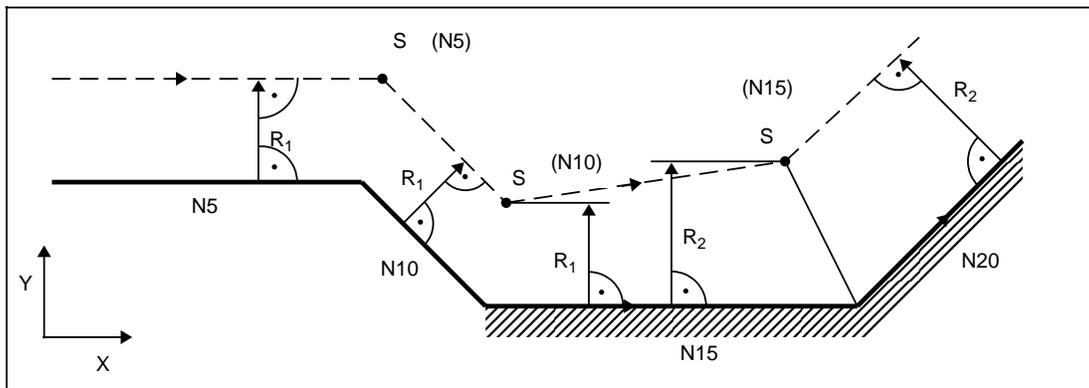
es wird **kein** Satzanfangsschnittpunkt mit der alten Korrektur errechnet; im Satzende des Satzes mit alter Korrekturnummer D01 wird ein senkrechter Vektor mit der Länge **R1** errichtet. Der **Satzendeschnittpunkt** wird mit der neuen Korrektur D02 **errechnet**.



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 N20** wirksam.



Wechsel der Korrekturwerte

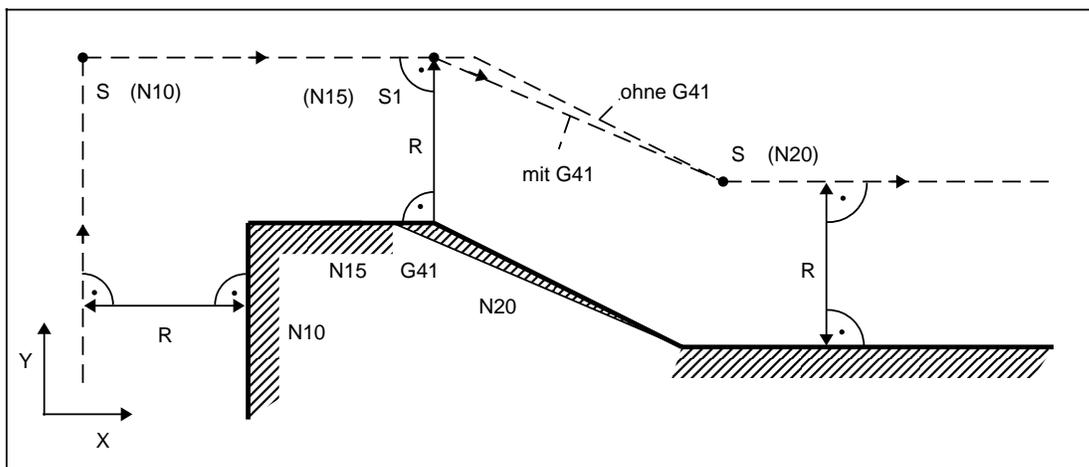
9.7 Wiederholung der schon angewählten G-Funktion (G41, G42) mit der gleichen Korrekturnummer

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

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

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

Fehler! G41 in N20 wiederholt

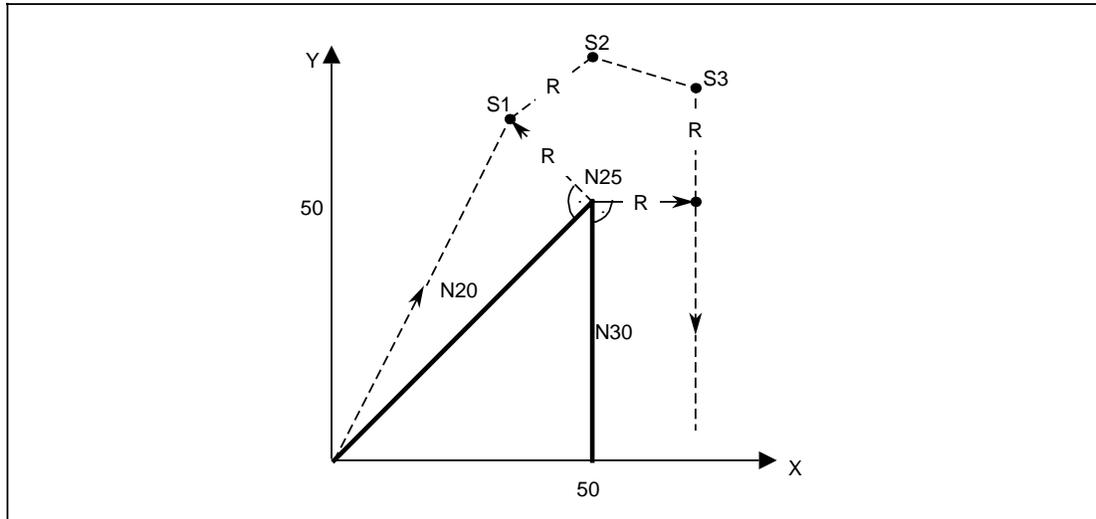


Wiederholte Anwahl

9.8 M00, M01, M02 und M30 bei angewählter FRK

M00, M01

Die NC **stoppt** am Stoppunkt **S** für **Einzelsatz**.
 (Die Punkte sind in den Bildern gekennzeichnet).



M00, M01 bei angewählter FRK.

Es sind **zwei Fälle** zu unterscheiden:

Fall 1: N10 G01 F100 X0 Y0 L_F
 N20 G41 X50 Y50 L_F
 N25 H111 M00 L_F
 N30 Y0 L_F

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

Fall 2: N10 G01 F100 X0 Y0 L_F
 N20 G41 X50 Y50 L_F
 N25 H111 M00 Z-5 L_F
 N30 Y0 L_F

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

M02, M30

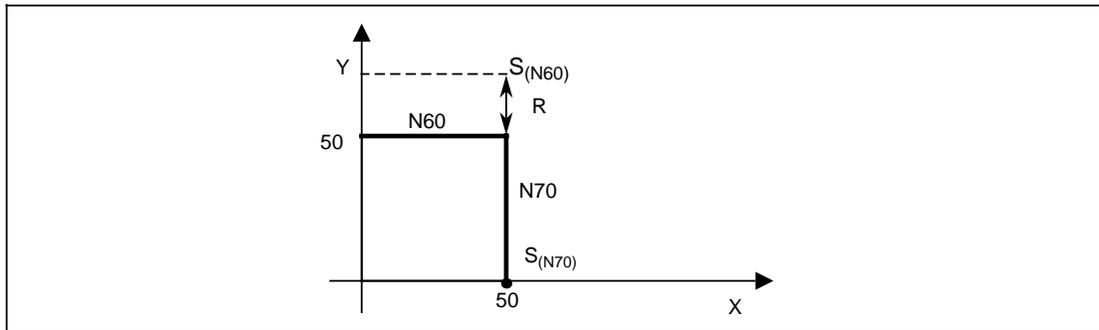
- Die **Korrektur** wird einen Satz vor dem Satz **herausgefahren**, in dem sie mit **G40** abge-
 wählt und mit mindestens **einer Achsadresse** programmiert wurde (hier N150).

```
N150 Y.. X.. LF
N155 G40 X.. M30 LF
```

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

```
N150 Y.. X.. LF
N155 G40 LF
N160 M30 LF
```

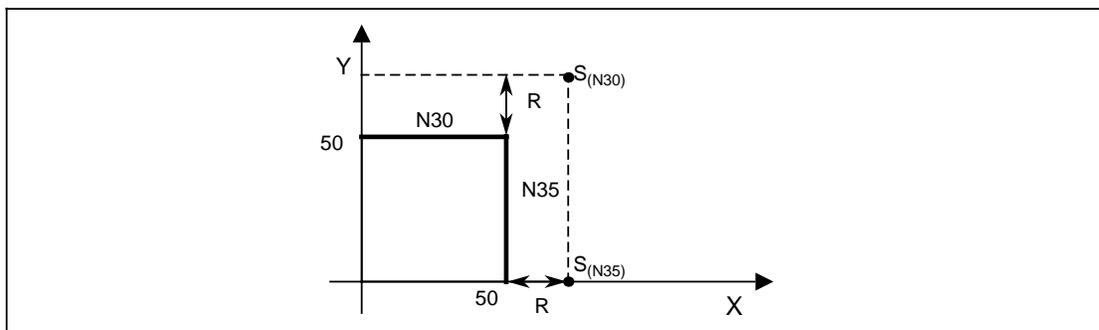
Abwahl mit G40:



N60 X50 Y50 L_F
 N70 G40 X50 Y0 M30 L_F

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

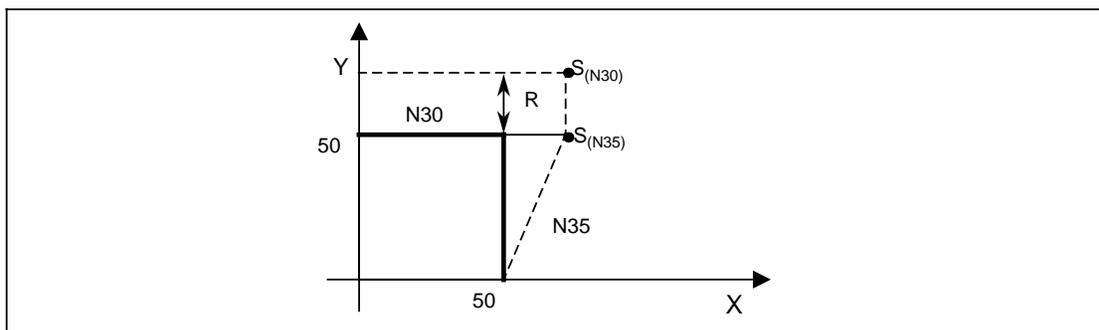
Kein Weg bei G40 oder Programmierung einer Achse im letzten Satz:



N30 X50 Y50 L_F
 N35 Y0 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:



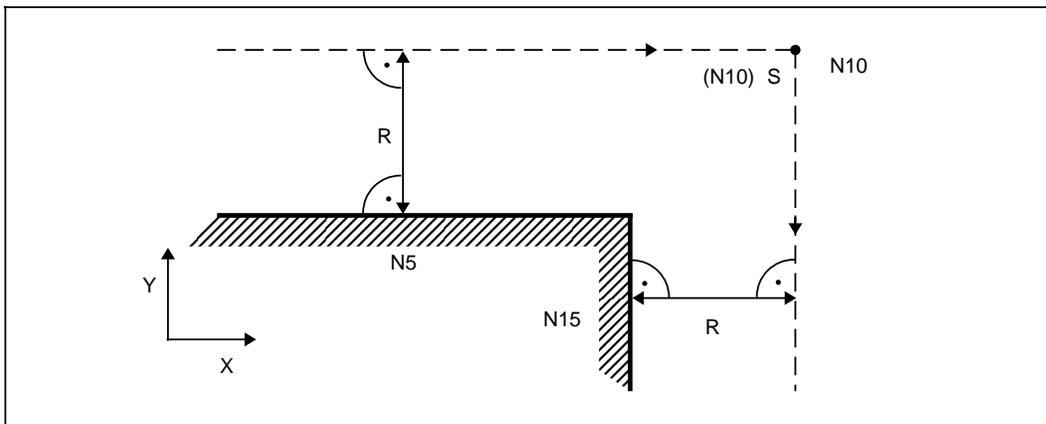
N30 X50 Y50 L_F
 N35 G40 Y0 L_F
 N40 M30 L_F

Die Korrektur wird in N35 herausgefahren.

9.9 FRK 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:

- **Wegadressen** sind programmiert, es findet aber keine Bewegung statt, weil der **Weg 0** ist.
N5 G91 X0 L_F
- 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":
Achsadressen außerhalb der Korrektorebene sind programmiert.
- **Ein** "Hilfsfunktionssatz" zwischen Strecken in der Korrektorebene.

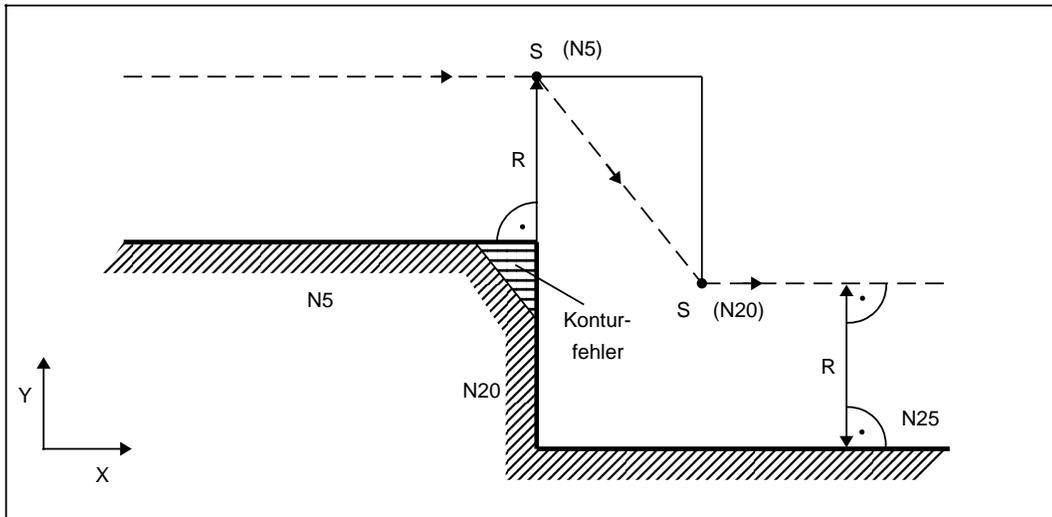


FRK: Ein „Hilfsfunktionssatz“ zwischen zwei Bewegungssätzen

```
N5 G91 X100 LF
N10 M08 LF
N15 Y-100 LF
```

Der Satz **N10** wird am Punkt **S** ausgeführt.

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



FRK: Zwei „Hilfsfunktionssätze“ zwischen zwei Bewegungssätzen

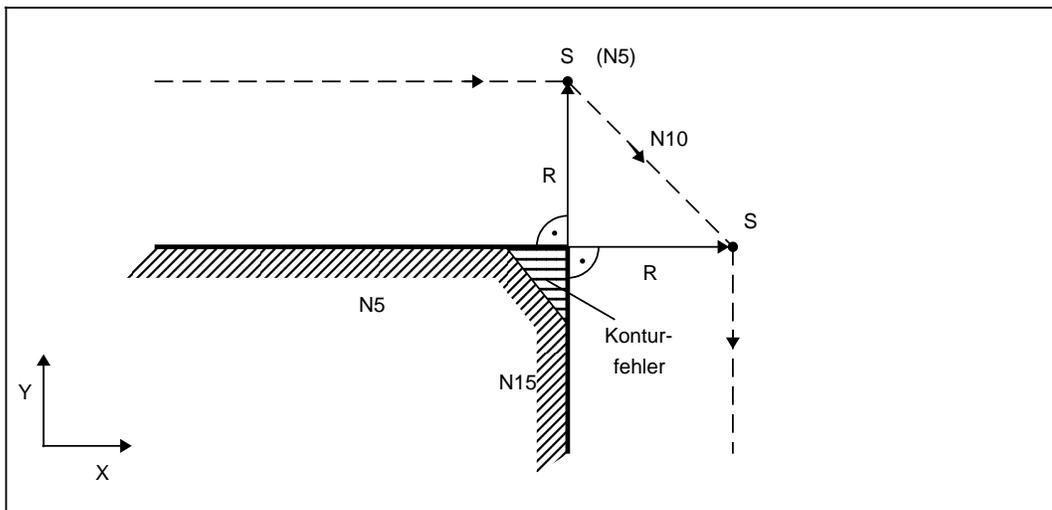
```

N5 G91 X100 LF
N10 M08 LF
N15 M09 LF
N20 Y-100 LF
N25 X100 LF

```

Die Sätze **N10** und **N15** werden am Punkt **S (N5)** ausgeführt.
Konturfehler (gestricheltes Gebiet)

- **Ein Satz** "Strecke=0" zwischen zwei Bewegungssätzen.



FRK: Ein Satz mit „Strecke 0“

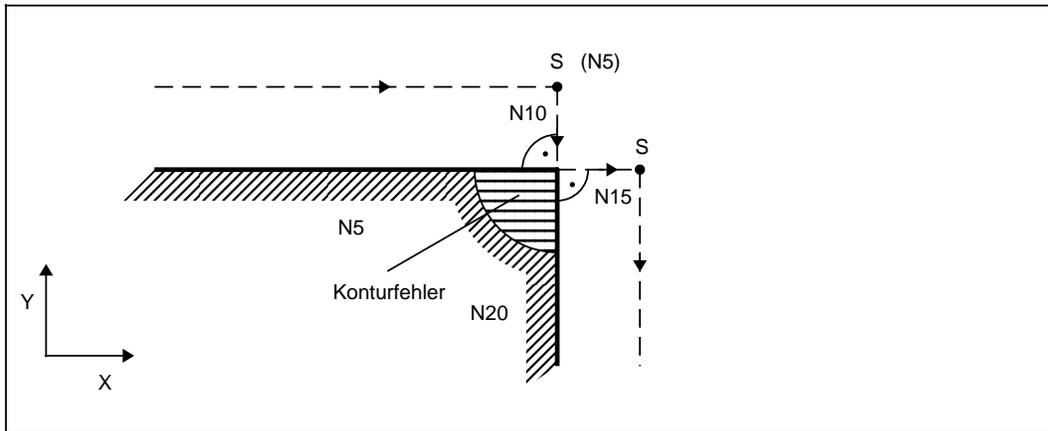
```

N5 G91 X100 LF
N10 X0 LF
N15 Y-100 LF

```

Konturfehler (gestricheltes Gebiet)

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



FRK: Zwei Sätze mit „Strecke 0“

N5 G91 X100 L_F

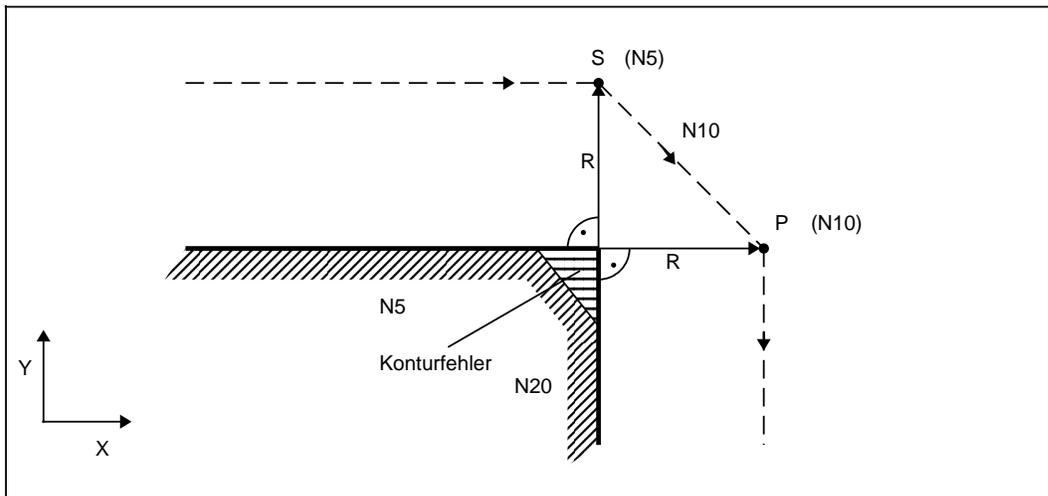
N10 X0 L_F

N15 X0 L_F

N20 Y-100 L_F

Konturfehler (gestricheltes Gebiet)

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



FRK: Ein Satz mit „Strecke 0“ und ein „Hilfsfunktionssatz“

N5 G91 X100 L_F

N10 X0 L_F

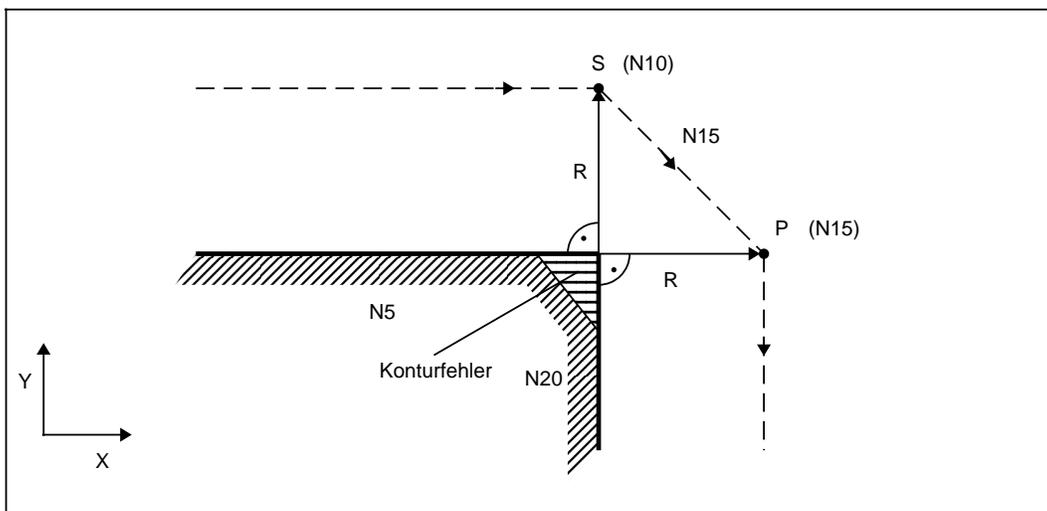
N15 M08 L_F

N20 Y-100 L_F

Der Satz **N15** wird am Punkt **P** ausgeführt.

Konturfehler (gestricheltes Gebiet)

- Ein „Hilfsfunktionssatz“ und ein Satz „Strecke=0“ zwischen Strecken in der Korrektur-ebene.



FRK: Ein „Hilfsfunktionssatz“ und ein Satz mit „Strecke 0“

N5 G91 X100 L_F

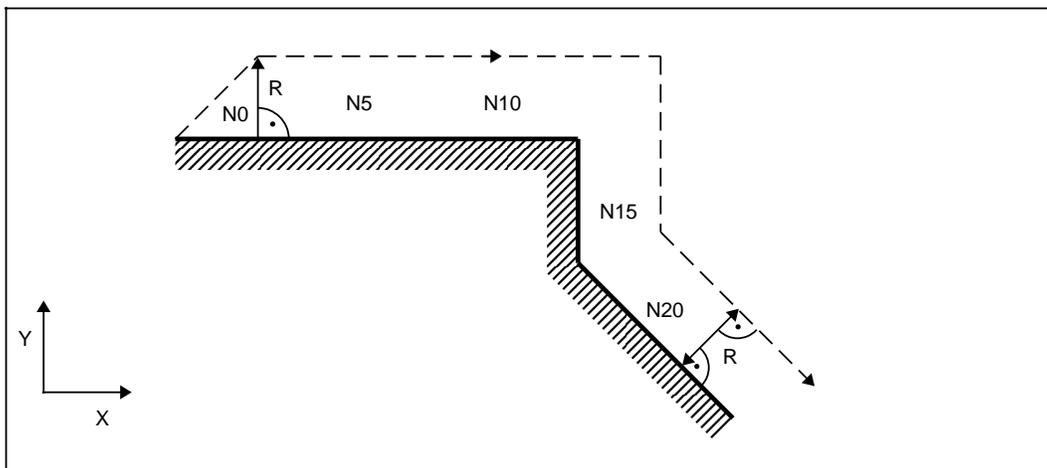
N10 M08 L_F

N15 X0 L_F

N20 Y-100 L_F

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

- Ein Satz „Nicht in der Korrektur-ebene“ zwischen Strecken in der Korrektur-ebene.



FRK: Ein Satz „Nicht in der Korrektur-ebene“

N0 G91 G41 G01 G17 X100 F100 D01 L_F

N5 X500 L_F

N10 Z500 L_F¹⁾

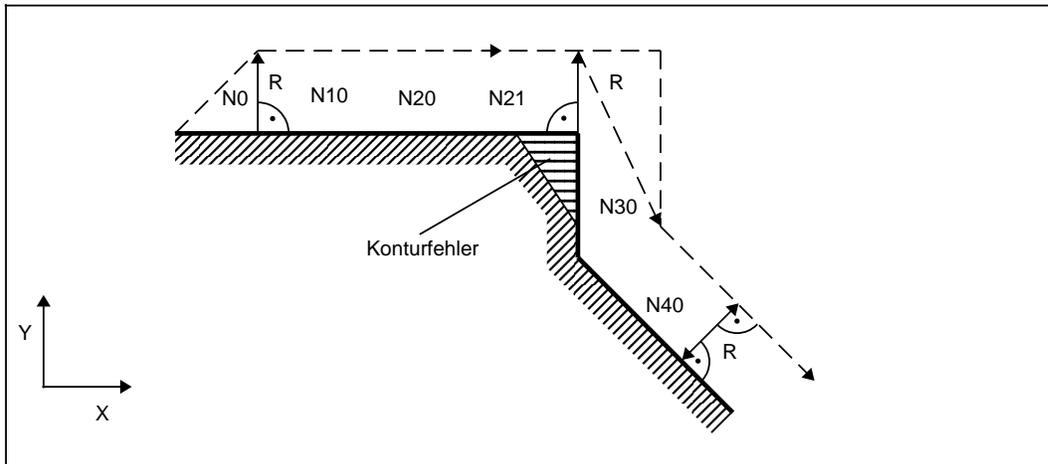
N15 Y-500 L_F

N20 X1000 Y-600 L_F

1) Satz nicht in Korrektur-ebene

keine Konturverletzung

- **Zwei Sätze „Nicht in der Korrektorebene“** zwischen Strecken in der Korrektorebene.



FRK: Zwei Sätze „Nicht in der Korrektorebene“

```

N0 G91 G41 G01 G17 X100 F100 D01 L_F
N10 X500 L_F
N20 Z500 L_F1)
N21 Z-500 L_F1)
N30 Y-500 L_F
N40 X1000 Y-600 L_F

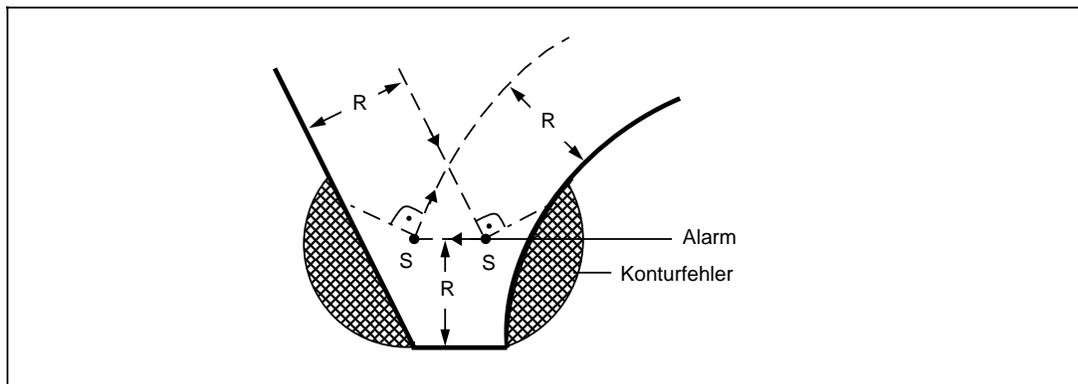
```

1) Satz nicht in der Korrektorebene
Konturfehler (gestricheltes Gebiet)

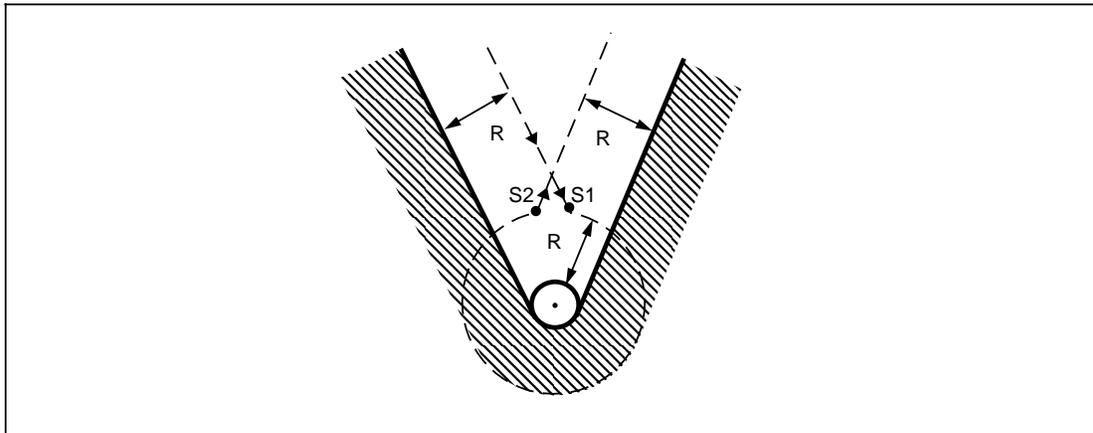
9.10 Sonderfälle der FRK

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.

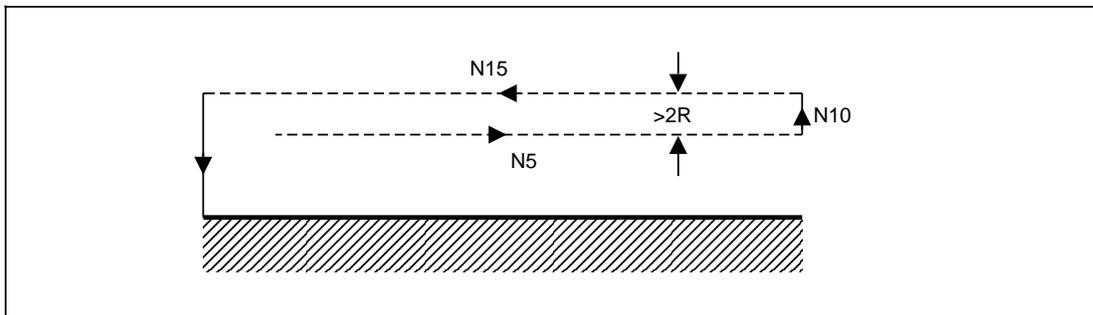
Die Bearbeitung wird nicht unterbrochen, aber es wird ein **Alarm angezeigt**.



FRK-Sonderfall: Zwischensatz < Korrekturbetrag



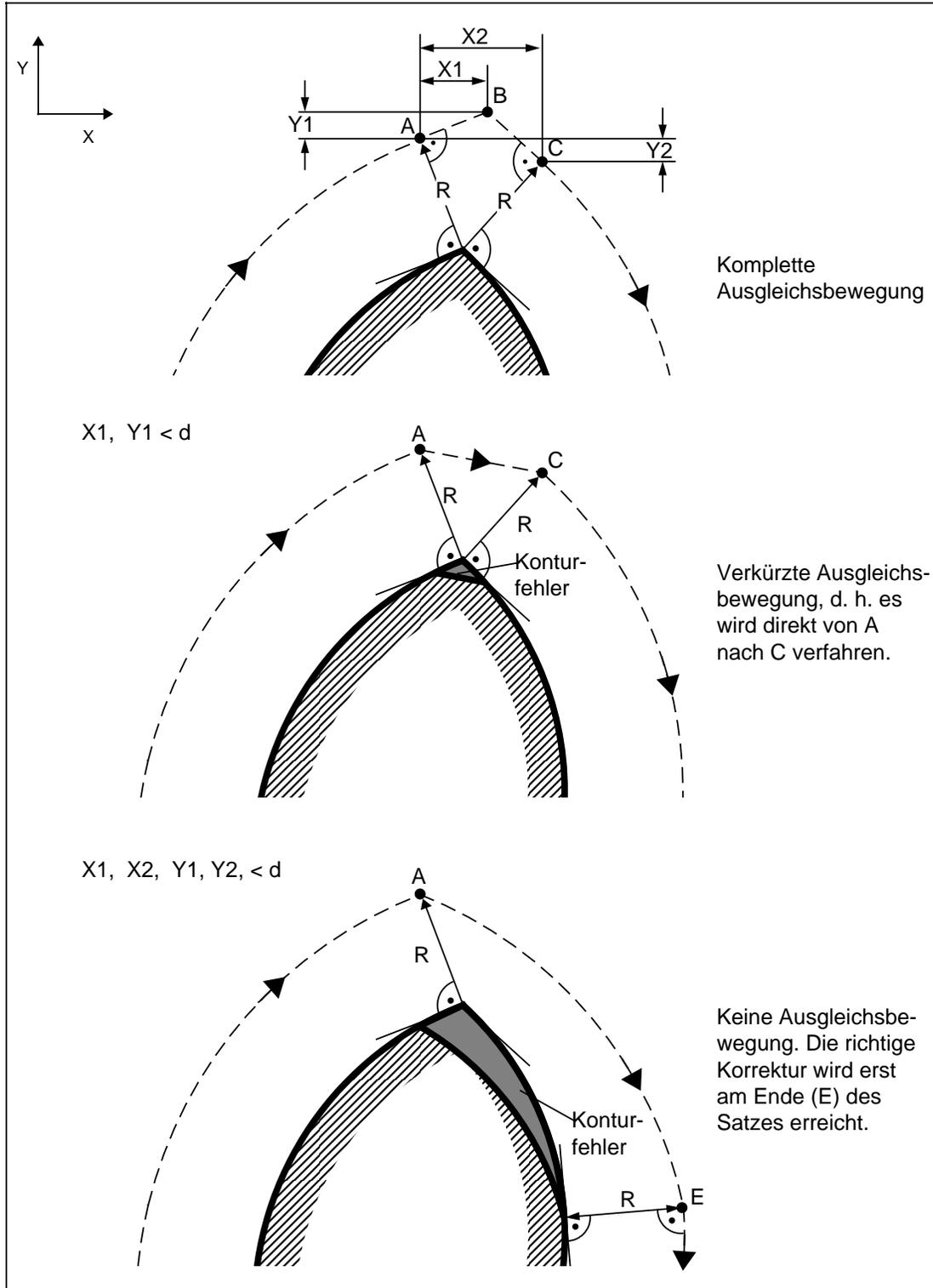
FRK-Sonderfall: Zwischensatz für Korrektur zu klein



FRK-Sonderfall: gleiche Korrekturrichtung und Verfahrrichtungswechsel

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

Für Außenkontur mit Kreisübergängen und stumpfe Winkel gilt:

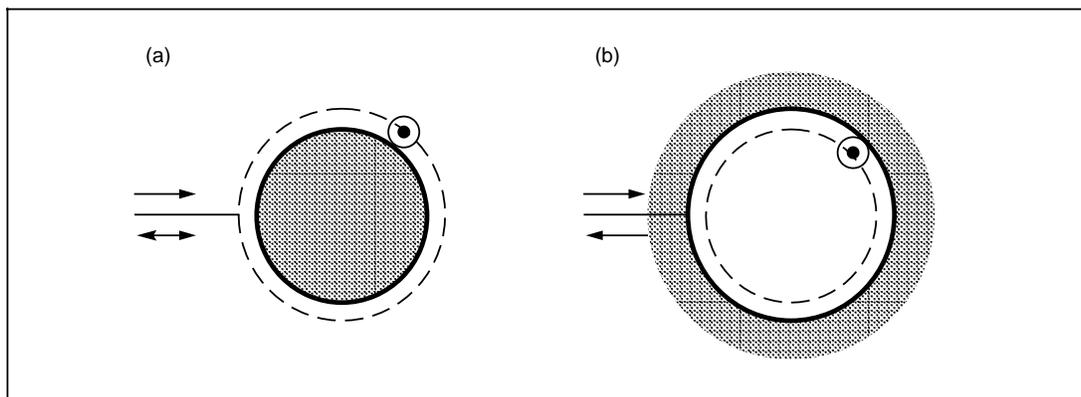


Um durch kleine Zwischensätze bedingtes Anhalten im Bahnsteuerbetrieb zu vermeiden, können die Wege AB und BC von der NC ausgelassen werden. Der genaue Bahnverlauf ist abhängig von einer bei der Inbetriebnahme festgelegten Toleranz d (max. $32000 \mu\text{m}$).

9.11 Wirkung bei negativen Korrekturwerten

Bei einem negativen Korrekturwert, (Fräserradius 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 Fräser-Mittelpunktsbahn 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 in SINUMERIK 810N/820N GA2/GA3 als fest abgespeicherte Unterprogramme im Anwenderspeichermodul (ASM) zur Verfügung:

- L801 Einzelloch stanzen
- L802 Lineare Lochfolge stanzen mit Zielangabe
- L803 Lineare Lochfolge stanzen mit Abstandsangabe
- L804 Lochfolge auf Kreislinie stanzen
- L805 Lochfolge im Rechteck stanzen
- L806 Kontur Rechteck trennen
- L807 Kontur Rechteck zerspanen
- L808 Kontur Kreis trennen
- L809 Kontur Kreis zerspanen
- L810 Kontur linearer Schlitz
- L811 Kontur zirkularer Schlitz
- L850 Lineares Anfahren/Verlassen der Kreiskontur
- L851 Anfahren/Verlassen der Kreiskontur im Viertelkreis
- L852 Rechteck schneiden
- L853 Dreieck schneiden
- L854 Laserschneiden
- L855 Laserschweißen

***Eine Beschreibung der Zyklen steht nicht zur Verfügung. Die Zyklen sind
nur als Anwendungsbeispiele zu betrachten.***

11.2.2 Operanden

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

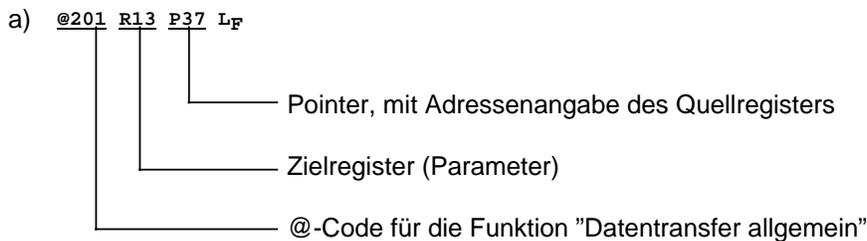
- K** Konstante
- R** R-Parameter (Register)
- P** Pointer (Zeiger)

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).

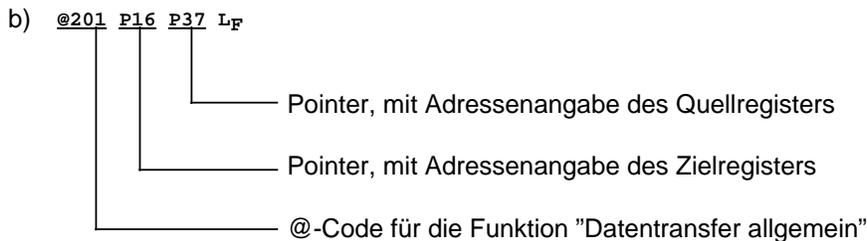
Der Pointer P weist auf einen Parameter hin, in dem die Adresse des Parameters steht, auf dessen Inhalt die Funktion angewendet werden soll (indirekte Wertangabe).

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.

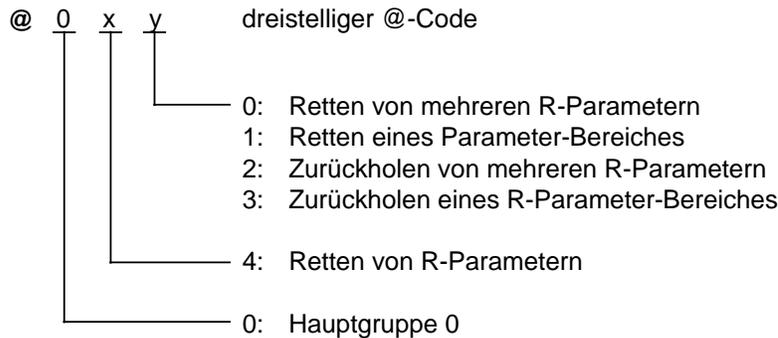
11.2.3 Notation

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 4: **Retten von R-Parametern**

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

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 gerettet.

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

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 gerettet.

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

Dieser Befehl entnimmt die geretteten 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>

Die mit @041 geretteten 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 rettet 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:

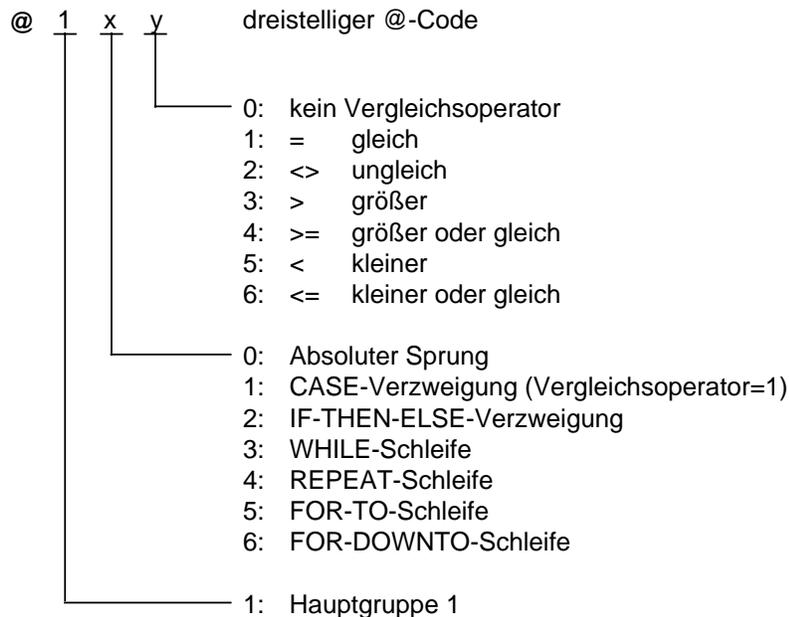
```

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

```

11.4 Programmverzweigungen

Die Hauptgruppe 1 ist folgendermaßen gegliedert:



Hauptgruppe 1/Untergruppe 0: **Absoluter Sprung**

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

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: „Satz nicht vorhanden“).

Beispiele:

```

@100  K375  LF          Absoluter Sprung auf Satz N375 in Richtung Programmende
@100  K-150 LF         Absoluter Sprung auf Satz N150 in Richtung Programmanfang

```

Hauptgruppe 1/Untergruppe 1: CASE-Verzweigung

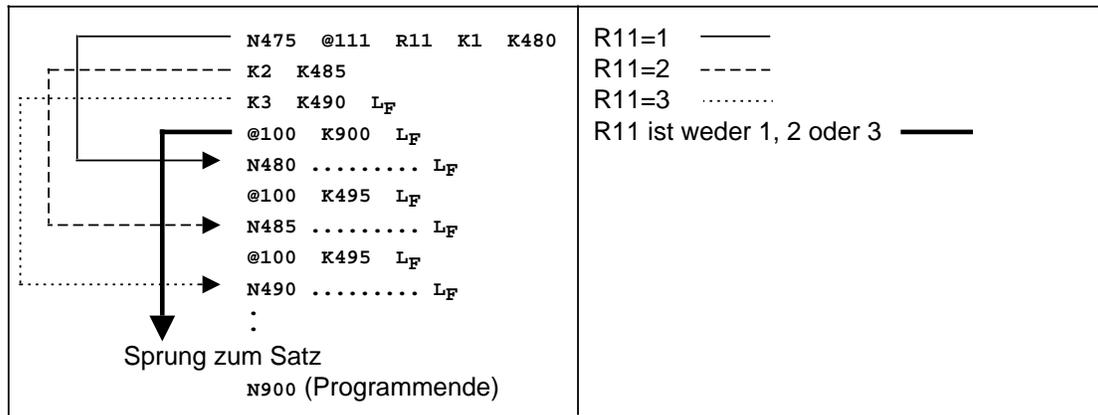
```

@111  <Var>  <Wert 1> <Const 1>
        <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 unbedingten/absoluten Sprung).

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 der Programmierer die Bearbeitungsebene durch Parametrierung von R11=1, 2 oder 3 festgelegt hat, 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 diesem Satz dessen Nummer durch die zuletzt stehende Konstante genannt wird.

Auch hier wirkt das Vorzeichen der Satznummer als Suchrichtung.

@121 <Var> <Wert> <Const>

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>

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>

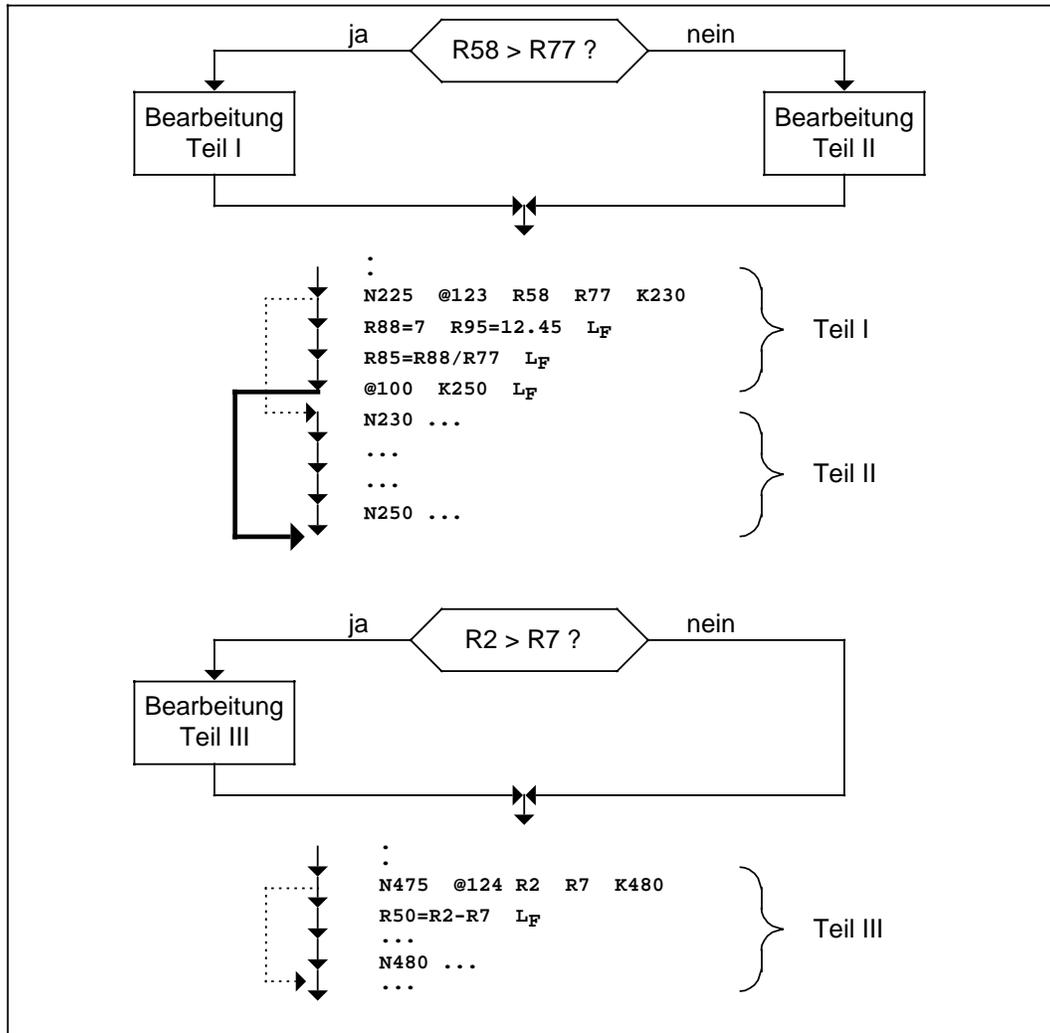
Ist (IF) der mit der Notation <Var> definierte Zahlenwert größer als der oder gleich dem mit <Wert> definierte, so (THEN) wird das Programm mit dem nächsten Satz fortgesetzt. Andererseits (ELSE) wird in den mit der Konstanten bestimmten Satz gesprungen.

@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

Programmablauf mit IF-THEN-ELSE-Verzweigung

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 Wiederholungsbedingungen 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.

Die Besetzung von y ist entsprechend den Anweisungen @121 ... @126 vorzunehmen. Ist der Vergleich nicht erfüllt, so wird an den unter <Const> definierten Satz gesprungen, der i.a. hinter dem mit dem absoluten Sprung steht.

Beispiele:

N300 @131 R13 R27 K375 L_F Fortsetzung der Schleife, solange Schleifenbedingung
R13=R27 erfüllt.

.

@100 K-300 L_F

N375 ... L_F

N300 @133 R13 R27 K375 L_F Fortsetzung der Schleife, solange Schleifenbedingung
R13>R27 erfüllt.

.

@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 Wiederholungsbedingungen 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-300 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:

R50=1 R51=5 R52=10 L _F	Wertzuweisung für R50, R51, R52
@201 R50 R51 L _F	Datentransfer von R51 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.

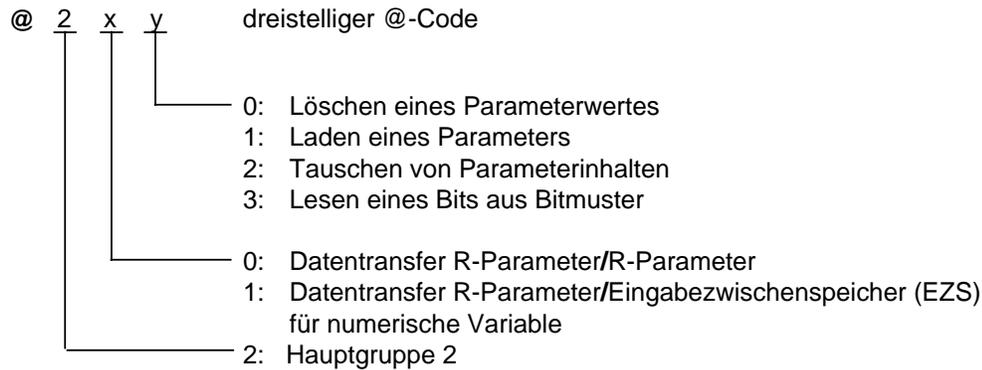
Am Ende der Schleife muß die Variable <Var> dekrementiert (@621) und mit einem absoluten Sprung zum Anfang der Schleife zurückgesprungen werden.

Beispiel:

R50=10 R51=5 R52=1 L _F	Wertzuweisung für Parameter R50, R51, R52
@201 R50 R51 L _F	Datentransfer von R51 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 Eingabezwichenspeicher

Der Eingabezwichenspeicher (EZS) wird gelöscht.
<Wert 3> Anfangsadresse
<Wert 4> Endadresse

@211 <Var > <Wert 1>

Eingabezwichenspeicher lesen

Der R-Parameter <Var> wird mit dem Inhalt der EZS-Zelle <Wert 1> geladen.

Beispiel:

@211 R50 K101 L_F

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.

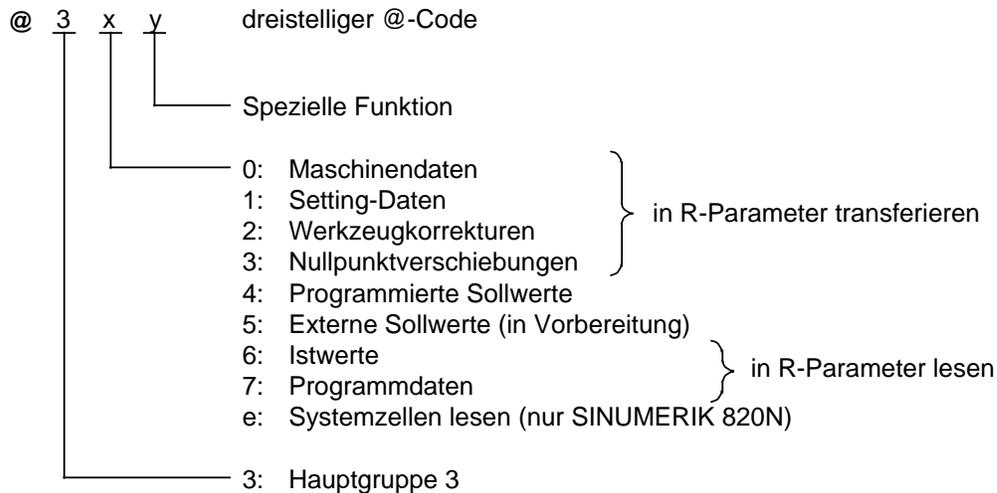
Beispiel:

@212 K102 K5

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 L_F

Im Parameter R50 steht der Wert des 1. Software-Endschalters in Plusrichtung für die 1. Achse

@301 <Var> <Wert 1>**Maschinendaten-NC-Bytes**

Unter <Wert 1> wird die Byte-Adresse eines NC-Maschinendatums definiert.
Adreßbereich: 5000 bis 6999.

@302 <Var> <Wert 1> <Wert 2> Maschinendaten-NC-Bits

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-Bytes

Unter <Wert 1> wird die Byte-Adresse eines PLC-Maschinendatums definiert.
Adreßbereich: 2000 bis 3999.

@308 <Var> <Wert 1> <Wert 2> Maschinendaten-PLC-Bits

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: **Setting-Daten 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-Bytes

Unter <Wert 1> wird die Byte-Adresse eines Setting-Datums definiert.
Adreßbereich: 5000 bis 9999.

@312 <Var> <Wert 1> <Wert 2> Settingdaten-NC-Bits

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> Werkzeugkorrektur

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_F

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- und 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 additive 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. Z.=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 additiven Koordinatendrehungen (G58=1 bis G59=2)
- <Wert 3> Nummer des Winkels (z. Z.=1)

Hauptgruppe 3/Untergruppe 4: **Programmierte Sollwerte 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.

@360 <Var> <Wert 2>

Achs-Position-Istwerkstückbezogen

Unter <Wert 2> ist die Achse zu definieren, deren werkstückbezogener Istwert in den R-Parameter zu übertragen ist.

Beispiel:

@360 R54 K2 I_F

Lesen des Positions-Istwertes der Achse Y (=2), bezogen auf den Werkstück-Nullpunkt und Eintragen in das Register 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.

**@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 Ist Drehzahl in den R-Parameter zu übertragen ist.

@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/M 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 (speziell nur 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 Sendebits für "Probelaufvorschub" im eigenen Kanal wird im Parameter R81 gelesen.

Hauptgruppe 3/Untergruppe e: **Systemzellen lesen**

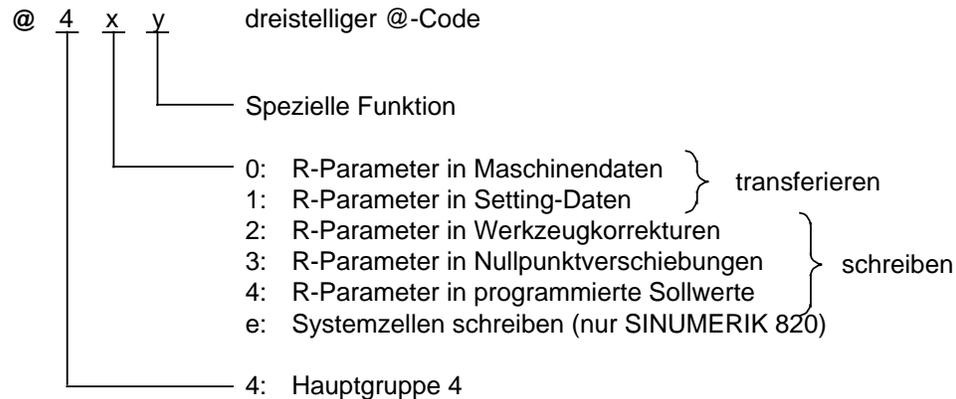
@3e4 <Var> <Wert 1>

Aktive Getriebestufe lesen (nur SINUMERIK 820N)

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 LF

Das Maschinendatum des 1. Software-Endschalters für die 2. Achse in Plusrichtung wird über Parameter R90 geladen.

@401 <Wert 1> <Wert>

Maschinendaten-NC-Bytes

Unter <Wert 1> wird die Byte-Adresse eines NC-Maschinendatenbits definiert.
 Adreßbereich: 5000 bis 6999.

@402 <Wert 1> <Wert 2> <Wert>

Maschinendaten-NC-Bits

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-Bytes

Unter <Wert 1> wird die Byte-Adresse eines PLC-Maschinendatums definiert.
 Adreßbereich: 2000 bis 3999.

@408 <Wert 1> <Wert 2> <Wert>

Maschinendaten-PLC-Bits

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 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 additiven 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. Die Funktionen G92 (Istwertsetzen im Programm) kann damit nicht realisiert werden.

Der @435 dient hauptsächlich zum gezielten **Rücksetzen** der PRESET-Verschiebung bei Programmende.

@437 <Wert 1> <Wert 2> <Wert 3> <Wert>**Einstellbare Koordinatendrehung**

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**

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 Koordinaten-
drehung**

Programmierbare Koordinatendrehung schreiben.

Es bedeuten:

- <Wert 1> Nummer des Kanals (0=eigener Kanal)
- <Wert 2> Gruppe der programmierbaren Koordinatendrehungen
(G58=1 bis G59=2)
- <Wert 3> Nummer des Winkels (z. Zt.=1)

@43a <Wert 1> <Wert 2> <Wert 3> <Wert>

**Programmierbare Koordinaten-
drehung additiv**

Es bedeuten:

- <Wert 1> Nummer des Kanals (0=eigener Kanal)
- <Wert 2> Gruppe 1 bis 2 (G58 bis G59)
- <Wert 2> 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.

Beispiel:

@440 K2 K100

Dem Verfahrweg 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 e: **Systemzellen schreiben**

@4e1 <Wert 1> <Wert 2> <Wert>**Spindelbeschleunigungszeitkonstante programmieren (nur SINUMERIK 820)**

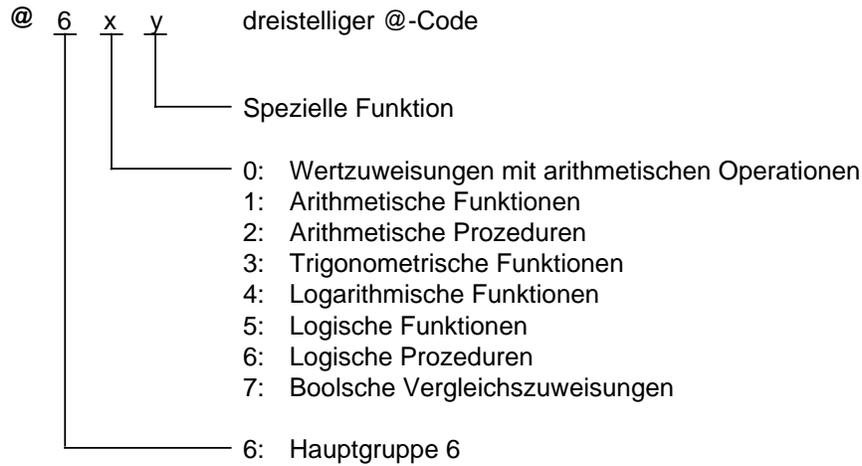
Dieser Befehl ermöglicht es, die Spindelbeschleunigungszeitkonstante zu programmieren.

Die unter <Wert> angegebene Spindelbeschleunigungszeitkonstante überschreibt nicht die in den Maschinendaten eingegebenen Werte, 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 L_F

Die Quadratwurzel wird aus der Konstanten (=64) gezogen und das Ergebnis (08) 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 den unter <Wert> definierten Winkelwert wird der Tangens gebildet und nach <Var> abgespeichert.

@634 <Var> <Wert>

Arcus Sinus

Von den unter <Wert> definierten Winkelwert 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.



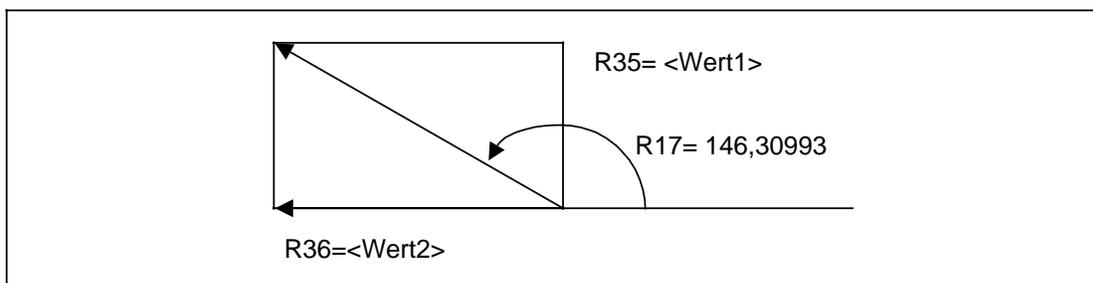
Als Operanden für <Wert 1> und <Wert 2> ist nur einmal eine Konstante <Const> zugelassen. Der andere Operand muß eine Variable <Var> (R-Parameter oder Pointer) sein.



Beispiel:

R35=20 R36=30 L_F
@637 R17 R35 R36 L_F

Der Winkel wird aus den Vektorkomponenten der 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

Von dem 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> stehenden Bitmuster (Pattern) wird logisch negiert . Das Ergebnis wird nach <Var> abgespeichert.

Beispiel:

R50=00101100 L_F
@654 R52 R50 L_F

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 L_F
R51=0 L_F
@657 R52 R50 R51 L_F

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 L_F
R51=0 L_F
@658 R52 R50 R51 L_F

Die booleschen Variablen R50 und R51 werden logisch nach UND verknüpft und 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 LF
```

```
@660 R60 K6 LF
```

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 LF
```

```
@661 R70 K2 LF
```

Das Bit Nr. 2 der Pattern-Variablen wird gesetzt. R70 hat den Inhalt 00000100.

Hauptgruppe 6/Untergruppe 7: Boolsche Vergleichszuweisungen

@671 <Var 1> <Var 2> <Wert>

Sind die unter <Var 2> und <Wert > definierten Zahlenwerte gleich, so wird die Boolsche Variable <Var 1> auf "1" gesetzt.

Beispiel:

```
R50=11001100 LF
```

```
@671 R51 R50 K11001100 LF
```

Da R50 gleich dem Bitmuster ist, wird R51 auf "1" gesetzt.

@672 <Var 1> <Var 2> <Wert>

Sind die unter <Var 2> und <Wert> definierten Zahlenwerte ungleich, so wird die boolsche 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 boolsche 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 boolsche Variable <Var 1> auf "1" gesetzt.

@675 <Var 1> <Var 2> <Wert>

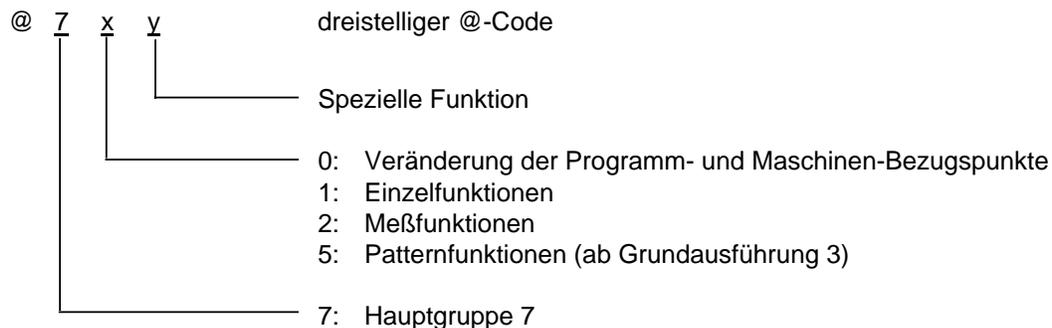
Ist der unter <Var 2> definierte Zahlenwert kleiner als der unter <Wert> stehende, so wird die boolsche Variable <Var 1> auf "1" gesetzt.

@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 boolsche 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 eines festen Maschinenpunktes 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 mit unterdrückt.

Beispiel:

@706 X1000 Y500 L_P

Die programmierten Verfahrwege in X und Y werden bezogen auf den Maschinennullpunkt angefahren

Hauptgruppe 7/Untergruppe 1 : Einzelfunktionen

@710 <Var 1> <Var 2>**Referenzaufbereitung**

Dieser Befehl wird zur Referenzaufbereitung benötigt.

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 Referenzaufbereitung benötigt insgesamt vier R-Parameter als Eingangsdaten.

Der erste dieser R-Parameter wird durch <Var 2> (R_m) festgelegt.

Der Eingangssteuerparameter (R_{m+3}) muß 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 daß bei jedem weiteren Aufruf die Werte vom nächsten 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 lädt 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	

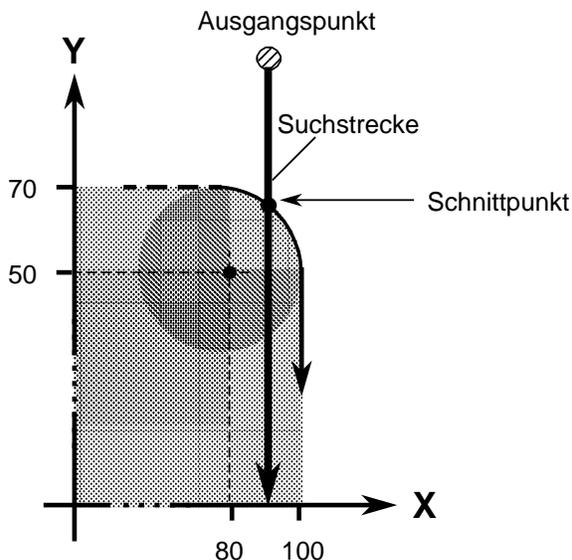
$R_{n+7} = 0$:	Satz ohne M17
$R_{n+7} = 1$:	Satz mit M17
$R_{n+7} = 2$:	M17 allein im Satz

@711 <Var 1> <Var 2> <Var 3>

Schnittpunktberechnung

Dieser Befehl wird zur Schnittpunktberechnung 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 Verfahrensbewegung programmiert. Es werden beide Achswerte zur Ermittlung des Schnittpunktes benötigt. Die Ausgangsdaten der Referenzaufbereitung werden dabei als Eingangsdaten für die Schnittstellenberechnung verwendet.



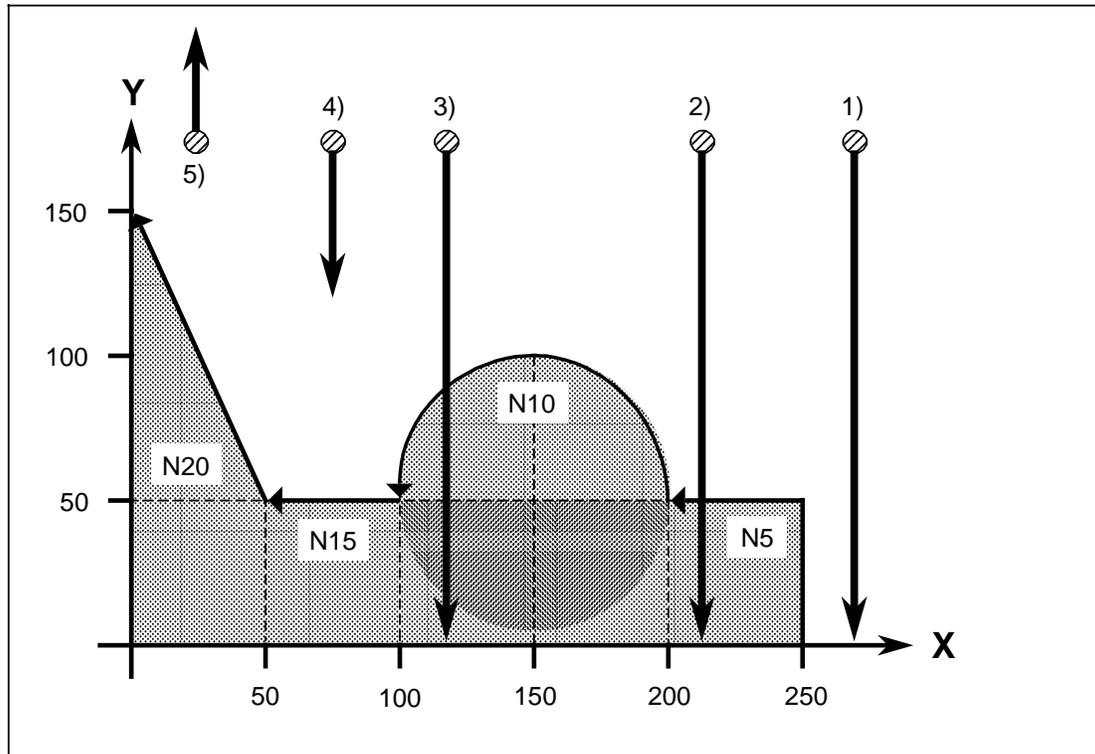
R-Parameter	Bedeutung	Registerinhalt
R _n	Satzanfang Y	70
R _n +1	Satzanfang X	80
R _n +2	Satzende Y	50
R _n +3	Satzende X	100
R _n +4	Interpolationsparameter J	-20
R _n +5	Interpolationsparameter I	0
R _n +6	G-Funktion	2

```
@711 Rm Rn Rr G01 G90 Y0 X90 LF
```

Suchrichtung, läßt in:

R _m	Ergebnis	1*)
R _m +1	Schnittpunkt Y	67,320
R _m +2	Schnittpunkt X	90,000

*) 1=gefunden, 0=nicht gefunden

Bewertung von Schnittpunktberechnungen mit @711:

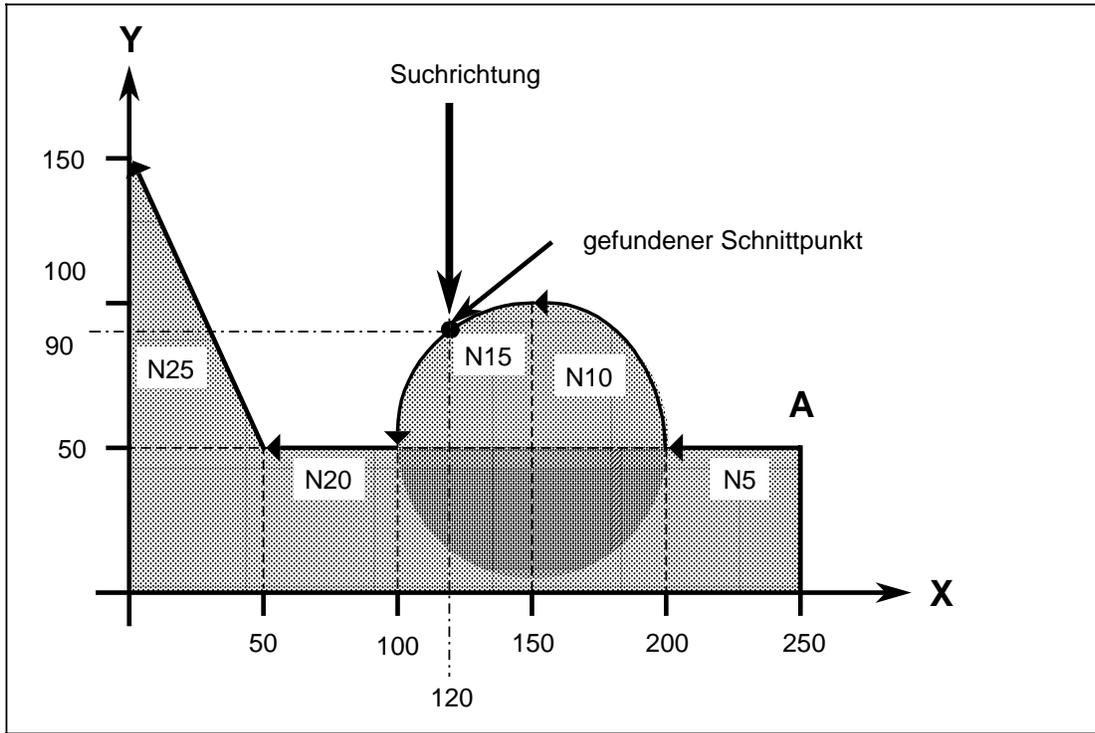
- 1) Schnittpunkt nicht gefunden
- 2) Schnittpunkt gefunden
- 3) Kreisbogen geht über die Quadrantengrenze (!):
Schnittpunkt nicht gefunden
- 4) Suchstrecke war zu kurz:
Schnittpunkt trotzdem gefunden
- 5) Falsche Suchrichtung:
Schnittpunkt trotzdem gefunden

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 Verfahrensbewegung 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 X=120.



Unterprogramm der Kontur:

```
L20 LF
N5 X200 Y50LF
N10 G03 X150 Y100 I-50 J0 LF
N15 G03 X100 Y50 I0 J-50 LF
N20 X50 Y50 LF
N25 X0 Y150 LF
N30 M17 LF
```

Hauptprogramm:

```
%30 LF
G0 X120 Y150 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 X120 Y0 LF
@100 K-300 LF
N305 G00 X=R72 Y=R71 LF
.
.
.
M30 LF
```

R50 bis R54, Eingangsdaten; R70: Flag für
 Schnittpunktberechnung
 While-Schleife, solange R 70=0
 Referenzaufarbeitung
 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 $Y=90$ (R_{71}) und $X=120$ (R_{72}) 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 $G_{70} = 0.03937$ und bei $G_{71} = 1$). 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 R_0 bis R_{98} und R_{900} bis R_{998} .

@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 L_F

@714 L_F

@123 R60 K100 K5 L_F

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 Verfahransätze 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 FRK erreicht werden. Die Dekodierung wird dabei erst wieder aktiviert, wenn alle Zwischenspeicher bis zur Koordinatendrehung geleert sind.

Beispiel:

@715

@437 K0 K1 K1 K30

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 FRK 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 „Achsverdopplung“ wirkt der Befehl @440 sowohl auf die führenden, programmierten Achsen als auch auf die verdoppelten Achsen.

Beispiel:

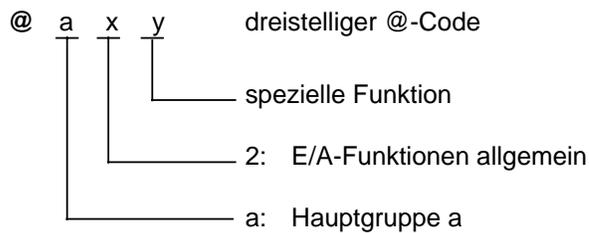
```

:
:
:
N30 @720 R93 K1 @440 K1 R70 L_F           Der Istwert der 1. Achse wird gemessen und in R93 geladen.
N40 @714 L_F                               Stop der Dekodierung nach dem Messen
:
:

```

Ist die Funktion „Achsverdopplung“ aktiv, wird der programmierte Verfahrweg auf die entsprechende Achse verdoppelt, aber nur der Istwert der führenden Achse gemessen. Beide Achsen stoppen nach dem Auslösen des Meßtasters.

11.10 E/A-Funktionen (nur GA3 SW2)



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 L_F Die vorhandenen Teileprogramme mit der Programm-Kennung
N.. @a26 K1 K1 K10 L_F %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>

Parameter Ausgabe 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:

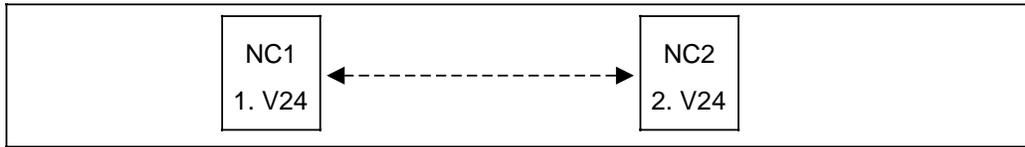
N.. @a20 K1	L _F	Anwahl der ersten V24-Schnittstelle
N.. @a28 K10	L _F	Die Steuerung erwartet einen R-Parameter in der Eingabe. Wird ein anderer Datentyp empfangen, erfolgt eine Fehlermeldung.

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. V24 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 R0 bis R100 vom Kanal 1 und anschließend das Übertragungsendezeichen ausgegeben.
N.. @a27 K1 K0 K100	L _F	
N.. @a29	L _F	

Anwendung:

- Nach einer Folge von Ausgaben ohne Übertragungsendekennung wird dem Empfänger mit @a29 das Ende des Datenpaketes mitgeteilt.

11.11 @-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 CL800-Ebene nur Const vorgebar

@-Code	CL800-Anweisung	Funktion
@040 Const R Par 1 . . . R Par n	(Push) ¹⁾	Sichern der angegebenen lokalen R-Parameter auf den Stack
@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

@-Code	CL800-Anweisung	Funktion
@12y <i>Var Wert Const</i>	IF "Bedingung" ²⁾ THEN Anweisung 1 ; [ELSE Anweisung 2 ;] END IF;	IF-THEN-ELSE-Anweisung y Vergleichsoperator Vop Var R-Parameter oder Pointer
@13y <i>Var Wert Const</i>	WHILE "Bedingung" ²⁾ DO Anweisung ;	Wiederholungs-Anweisung mit Abfrage der Wiederhol- bedingung am Anfang. y Vergleichsoperator Vop
@14y <i>Var Wert Const</i>	REPEAT Anweisung ; UNTIL "Bedingung"; ²⁾	Wiederholungs-Anweisung mit Abfrage der Wiederhol- bedingung am Ende. y Vergleichsoperator Vop
@151 <i>Var Wert 2 Const</i>	FOR Var = Wert 1 TO Wert 2 DO Anweisung ;	Wiederholungs-Anweisung mit Wiederholungen solange bis <i>Var</i> inkrementell <i>Wert 2</i> erreicht hat
@161 <i>Var Wert 2 Const</i>	FOR Var = Wert 1 DOWNTO Wert 2 DO Anweisung ;	Wiederholungs-Anweisung mit Wiederholungen solange bis <i>Var</i> dekrementell <i>Wert 2</i> erreicht hat
@200 <i>Var</i>	CLEAR(<i>Var</i>);	Lösche Variable
@201 <i>Var Wert</i>	Var = Wert	Lade Variable mit Wert
@202 <i>Var 1 Var 2</i>	XCHG (<i>Var 1</i> , <i>Var 2</i>);	Tauschen der Variablen- inhalte
@203 <i>Var 1 Var 2 Const</i>		Lesen eines Bits aus Bit- muster
@210 <i>Wert 3 Wert 4</i>	CLEAR MIB (<i>Wert 3</i> , <i>Wert 4</i>);	Lösche Eingabezwischen- speicher Wert 3: EZS-Anfangsadr. Wert 4: EZS-Endadresse
@211 <i>Var Wert 1</i>	Var = 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>	Var =MDN (<i>Wert 1</i>);	Maschinendaten NC Wert 1: Adr. 0 ... 4999
@301 <i>Var Wert 1</i>	Var =MDNBY (<i>Wert 1</i>);	Maschinendaten-NC-Bytes Wert 1: Byte-Adr. 5000 ... 6999

@-Code	CL800-Anweisung	Funktion
@302 <i>Var Wert 1 Wert 2</i>	Var =MDNBI (Wert 1 , Wert 2);	Maschinendaten-NC-Bits Wert 1: Byte-Adr. 5000 ... 6999 Wert 2: Bit-Adr. 0 ... 7
@306 <i>Var Wert 1</i>	Var =MDP (Wert 1);	Maschinendaten-PLC Wert 1: Adr. 0 ... 1999
@307 <i>Var Wert 1</i>	Var =MDPBY (Wert 1);	Maschinendaten-PLC-Bytes Wert 1: Byte-Adr. 2000 ... 3999
@308 <i>Var Wert 1 Wert 2</i>	Var =MDPBI (Wert 1 , Wert 2);	Maschinendaten-PLC-Bits 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-Bytes Wert 1: Byte-Adr. 5000 ... 9999
@312 <i>Var Wert 1 Wert 2</i>	Var =SENB (Wert 1 , Wert 2);	Settingdaten-NC-Bits 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: Achs-Nr. 1, 2 ... Wert 3: 0/1 (Grob/Fein)
@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 ...

@-Code	CL800-Anweisung	Funktion
@334 Var Wert 2	Var =ZOPS (Wert 2);	PRESET-Verschiebung Wert 2: Achs-Nr. 1, 2 ...
@336 Var Wert 2	Var =ZOS (Wert 2);	Summen-Verschiebung Wert 2: Achs-Nr. 1, 2 ...
@337 Var Wert 1 Wert 2 Wert 3	Var = ZOADW (Wert 1 , Wert 2 , Wert 3);	Einstellbare Koordinatendrehung (G54 ... G57) Wert 1: Kanal-Nr. 0 ... 3 Wert 2: Gruppe 1 ... 4 (G54 ... G57) Wert 3: =Winkel-Nr. (1)
@338 Var Wert 1 Wert 2 Wert 3	Var =ZOPRDW (Wert 1 , Wert 2 , Wert 3);	Programmierbare Koordinatendrehung (G58 ... G59) Wert 1: Kanal-Nr. 0 ... 3 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 ... 3 Wert 3: Spindel-Nr. 0 ...6
@345 Var Wert 1 Wert 2	Var =PRVC (Wert 1 , Wert 2);	Programmierte Schnittgeschwindigkeit Wert 1: Kanal-Nr. 0, 1, 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. 1 ... 6
@364 Var Wert 2	Var =ACSS (Wert 2);	Spindeldrehzahl-Istwert Wert 2: Spindel-Nr. 1 ... 6
@36a Var Wert 1	Var =ACD (Wert 1);	D-Funktion-Ist Wert 1=0
@36b Var Wert 1 Wert 3	Var =ACG (Wert 1 , Wert 3);	Lesen der G-Funktion aus dem Arbeitsspeicher des aktuellen Satzes Wert 1: Kanal-Nr. 0, 1, 2 Wert 3: interne G-Gruppe, zu der G-Funktion gehört 0 ... 15

@-Code	CL800-Anweisung	Funktion
@371 <i>Var Wert 1 Wert 3</i>	Var =SOB (Wert 1 , Wert 3);	Sonderbits Wert 1: Kanal-Nr. 0 ... 2 =Kanalab- hängig 99=Kanal- unabhängig Wert 3: Bit-Nr. 0 ... 7
@3e4 <i>Var Wert 1</i>	Var =AGS (Wert 1);	Aktive Getriebestufe lesen Wert 1: Spindel-Nr. 0 bis 6
@400 <i>Wert 1 Wert</i>	MDN (Wert 1)= Wert ;	Maschinendaten-NC Wert 1: Adr. 0 ... 4999
@401 <i>Wert 1 Wert</i>	MDNBY (Wert 1)= Wert ;	Maschinendaten-NC-Bytes Wert 1: Byte-Adr. 5000 ... 6999
@402 <i>Wert 1 Wert 2 Wert</i>	MDNBI (Wert 1 , Wert 2) = Wert ;	Maschinendaten-NC-Bits Wert 1: Byte-Adr. 5000 ... 6999 Wert 2: Bit-Adr. 0 ... 7
@406 <i>Wert 1 Wert</i>	MDP (Wert 1)= Wert ;	Maschinendaten-PLC Wert 1: Adr. 0 ... 1999
@407 <i>Wert 1 Wert</i>	MDPBY (Wert 1)= Wert ;	Maschinendaten-PLC-Bytes Wert 1: Byte-Adr. 2000 ... 3999
@408 <i>Wert 1 Wert 2 Wert</i>	MDNBI (Wert 1 , Wert 2) = Wert ;	Maschinendaten-PLC-Bits Wert 1: Byte-Adr. 2000 ... 3999 Wert 2: Bit-Adr. 0 ... 7
@410 <i>Wert 1 Wert</i>	SEN (Wert 1)= Wert ;	Settingdaten-NC Wert 1: Adr. 0 ... 4999
@411 <i>Wert 1 Wert</i>	SENBY (Wert 1)= Wert ;	Settingdaten-NC-Bytes Wert 1: Byte-Adr. 5000 ... 9999
@412 <i>Wert 1 Wert 2 Wert</i>	SENBI Wert 1 , Wert 2 = Wert ;	Settingdaten-NC-Bits Wert 1: Byte-Adr. 5000 ... 9999 Wert 2: Bit-Adr. 0 ... 7
@420 <i>Wert 1 Wert 2 Wert 3 Wert</i>	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)

@-Code	CL800-Anweisung	Funktion
@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: 0/1 (Grob/Fein)
@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: 0/1 (Grob/Fein)
@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 ... 3 Wert 2: Gruppe 1 ... 4 (G54 ... G57) Wert 3: =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 ... 3 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 ... 3 Wert 2: Gruppe 1 oder 2 (G58-G59) Wert 3: = Winkel-Nr. (1)

@-Code	CL800-Anweisung	Funktion
@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 ...3 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
@447 Wert	PANG= Wert ;	Programmierter Winkel
@448 Wert 3 Wert	PRIP (Wert 3)= Wert ;	Programmierter Interpola- tionsparameter für Kreis und Gewinde Wert 3: Achs-Nr. 1,2, ...
@4e1 Wert 1 Wert 2 Wert	SATC (Wert 1 , Wert 2)= Wert ;	Spindelbeschleunigungs- zeitkonstante schreiben Wert 1: Spindel-Nr. 0 bis 6 Wert 2: Getriebestufe 1 bis 8 Wert : Spindelbeschleu- nigungszeitkon- stante 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" mit 1
@621 Var	DEC (Var);	Dekrementieren von "Var" mit 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

@-Code	CL800-Anweisung	Funktion
@634 <i>Var Wert</i>	<i>Var</i> =ARC SIN (Wert);	Arcus Sinus
@637 <i>Var Wert 1 Wert 2</i>	<i>Var</i> =ANGLE (Wert 1 , Wert 2);	Winkel aus zwei Vektorkomponenten
@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> = Var1 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 (Var . Const);	Lösche Bit in Pattern Const=Bit-Nr. 0 ... 7
@661 <i>Var Const</i>	SET BIT (Var . 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 Maschinen- Istwert-System
@710 <i>Var 1 Var 2</i>	<i>Var 1</i> =PREP REF (<i>Var 2</i>);	Referenzaufbereitung <i>Var 1</i> : Ausg.daten ab <i>Var 1</i> <i>Var 2</i> : Eing.datum ab <i>Var 2</i>
@711 <i>Var 1 Var 2 Var 3</i>	<i>Var 1</i> =INT SEC (<i>Var 2</i> <i>Var 3</i>);	Schnittpunktberechnung <i>Var 1</i> : Ausg.daten ab <i>Var 1</i> <i>Var 2</i> : erste Kontur ab <i>Var 2</i> <i>Var 3</i> : mit 0 vorbesetzen
@713 <i>Var</i>	<i>Var</i> =PREP CYC;	Startvorbereitung f. Zyklen <i>Var</i> : Ausgangsdaten ab <i>Var</i>
@714	STOP DEC;	Stop der Decodierung, bis Zwischenspeicher leer ist.

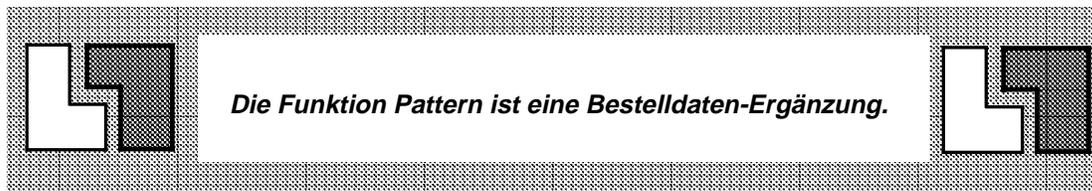
@-Code	CL800-Anweisung	Funktion
@715	STOP DEC 1;	Stop der Decodierung, bis Zwischenspeicher leer ist. (bei Koordinatendrehung)
@720 <i>Var Wert</i>	Var =MEAS M Wert	Fliegendes Messen Var: Daten hinterlegt ab Var Wert: Nr. des Meßeingangs; 1 oder 2
@750 <Var 1><Var 2> < Wert 3><Wert 4>		Pattern Initialisierung <Var 1> Verschiebung in X-Richtung zwischen den Patternteilen <Var 2> Verschiebung in Y-Richtung zwischen den Patternteilen <Wert 3> Anzahl Patternteile in X-Richtung <Wert 4> Anzahl Patternteile in Y-Richtung
@751 <Wert 1>		Pattern Schleifenanfang bei gleichsinniger Bearbeitung <Wert 1> 0/1 0= zuerst Teile in X-Richtung abarbeiten und dann in Y-Richtung 1= zuerst Teile in Y-Richtung abarbeiten und dann in X-Richtung
@752 <Wert 1>		Pattern Schleifenende bei gleichsinniger Bearbeitung <Wert 1> 0/1 0= Start der nächsten Schleife bei Teil 1/1 1= Start der nächsten Schleife am zuletzt bearbeiteten Teil
@753 <Wert 1>		Pattern Schleifenanfang bei mäanderförmiger Bearbeitung <Wert 1> 0/1 0= zuerst Teile in X-Richtung abarbeiten und dann in Y-Richtung 1= zuerst Teile in Y-Richtung abarbeiten und dann in X-Richtung

@-Code	CL800-Anweisung	Funktion
@754		Pattern Trennung Programmabschnitt A und B bei mäanderförmiger Patternbearbeitung
@755		Pattern Schleifenende bei mäander- förmiger Patternbearbeitung
@a20 <i>Wert</i>		V24-Schnittstelle anwählen
@a25 <i>Wert1</i>		Ausgabe der Nullpunktver- schiebungen über V24
@a26 <i>Wert 2 Wert 3</i> <i>Wert 4</i>		Ausgabe der Daten über V24
@a27 <i>Wert 1 Wert 3</i> <i>Wert 4</i>		Parameterausgabe über V24
@a28 <i>Wert 2</i>		Einlesen von Daten über V24
@a29		Ausgabe ETX über V24

11.12 Pattern (ab SINUMERIK 810N/820N Grundauführung 3)

In der nachfolgenden Beschreibung werden die Patternfunktionen anhand einer konkreten Aufgabenstellung dargestellt und eine Lösungsmöglichkeit aufgezeigt.

Die Kenntnis der nibbelspezifischen Funktionen der SINUMERIK 810N/820N wird vorausgesetzt.

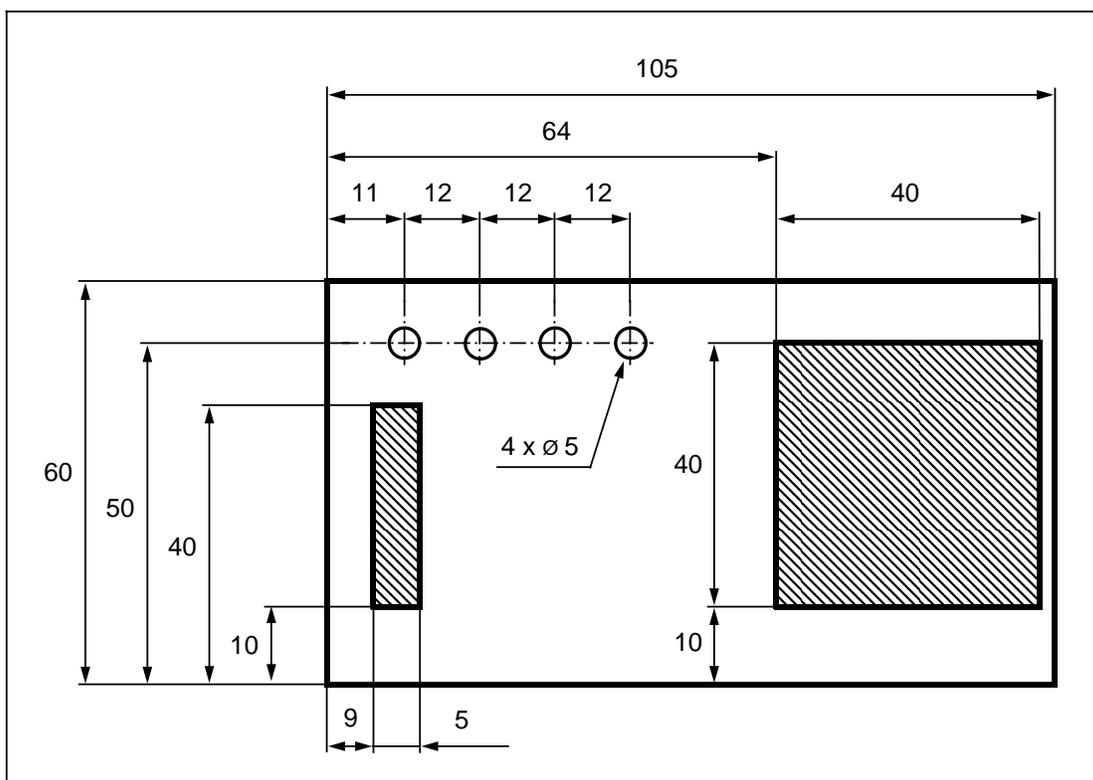


11.12.1 Aufgabenstellung

Das im nachfolgenden Bild dargestellte Teil soll mit den zur Verfügung stehenden Werkzeugen programmiert werden.

Werkzeugtabelle:

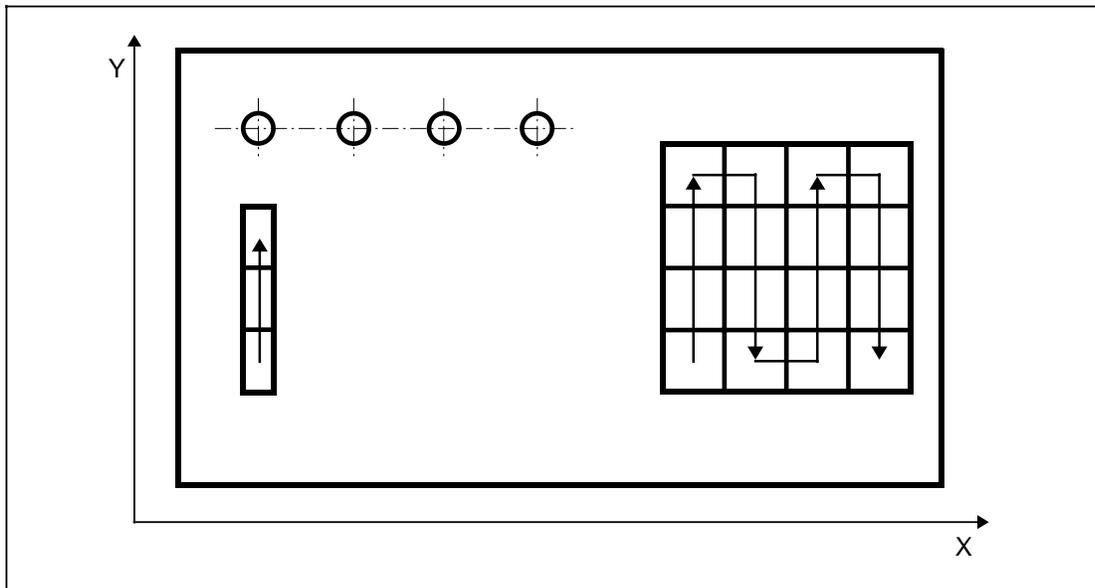
Rundwerkzeug	T1 mit Durchmesser 5 mm
Quadratwerkzeug	T2 mit 10 mm
Rechteckwerkzeug	T3 mit 5-10 mm



11.12.1.1 Einzelteil

Die Aufgabe kann durch nachfolgendes NC-Programm gelöst werden:

```
%101 L_F  
N05 T1 L_F (Rundwerkzeug)  
N10 G0 X11 Y50 M20 L_F  
N15 X47 E12 M22 L_F  
N20 M20 L_F  
N25 T2 L_F (Quadratwerkzeug)  
N30 X69 Y15 M20 L_F  
N35 G91 Y30 E10 M22 L_F  
N40 X10 L_F  
N45 Y-30 L_F  
N50 X10 L_F  
N55 Y30 L_F  
N60 X10 L_F  
N65 Y-30 L_F  
N70 G90 M20 L_F  
N75 T3 L_F (Rechteckwerkzeug)  
N80 X11.5 Y15 M20 L_F  
N85 Y35 M22 L_F  
N90 M20 L_F  
N95 M02 L_F
```



11.12.2 Pattern

11.12.2.1 Definition

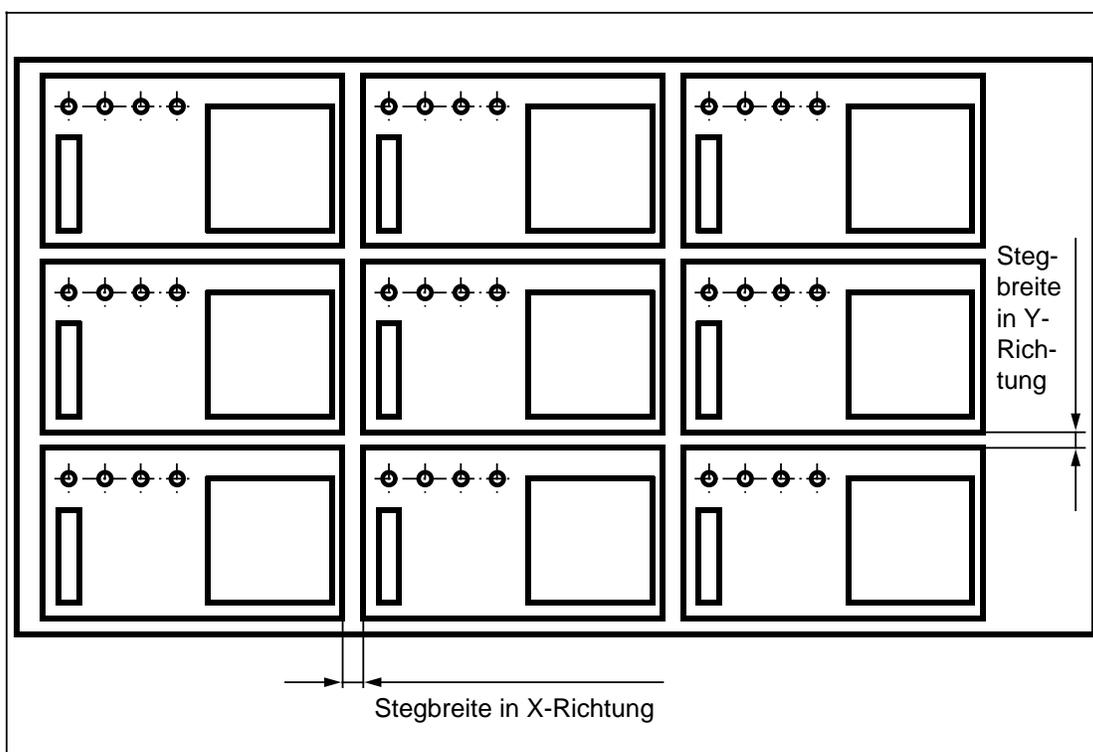
Pattern: englisch Muster, bedeutet, das gleiche Einzelteil mehrfach aus einer Blechtafel herzustellen.

Das Teil wird nur einmal programmiert und dann steuerungsimtern auf der Blechtafel "kopiert". Dieses Vorgehen wird als **Pattern** bezeichnet.

Hierzu werden weitere Angaben benötigt:

- Maße der Blechtafel
- Maße der Trennwerkzeuge
- Stegbreite in X- und Y-Richtung

Mit Hilfe dieser Angaben berechnet die SINUMERIK die Verschiebung in X- und Y-Richtung, sowie die maximal mögliche Anzahl der Patternteile.



Zusätzlich muß noch die Bearbeitungsreihenfolge festgelegt werden. Es gibt zwei Möglichkeiten:

- die werkzeugorientierte Bearbeitung und
- die teileorientierte Barbeitung.

11.12.2.2 Werkzeugorientierte Bearbeitung

Jedes Werkzeug wird nur einmal eingewechselt.

Mit dem eingewechselten Werkzeug werden zuerst alle Patternteile bearbeitet, bevor das nächste Werkzeug eingewechselt wird. Diese Art der Bearbeitung wird normalerweise eingesetzt und empfiehlt sich besonders dann, wenn viele verschiedene Werkzeuge zur Bearbeitung eines Teils nötig sind.

11.12.2.3 Teileorientierte Bearbeitung

Jedes Patternteil wird komplett bearbeitet, bevor das nächste Teil begonnen wird. Diese Art der Bearbeitung hat eventuell viele Werkzeugwechsel zur Folge und kommt daher nur dann zum Einsatz, wenn wenig verschiedene Werkzeuge nötig sind und der Werkzeugwechsel wenig Zeit beansprucht.

11.12.3 Patternprogrammierung mit SINUMERIK 810N/820N

Die SINUMERIK 810N/820N bietet spezielle Programmbefehle, sogenannte Klammeraffenbefehle an, die dem Anwender die Erstellung optimierter Patternprogramme mit minimalem Programmieraufwand ermöglichen.

Das Einzelteil muß nur einmal programmiert werden. Mit Hilfe dieser Programmbefehle ist ein mehrmaliges Durchlaufen dieses Programmteils, die sogenannte Schleifenbildung, möglich, wobei automatisch die notwendigen Verschiebungen zu den weiteren Patternteilen erfolgen.

Diese Klammeraffenbefehle (@-Befehle) haben unterschiedliche Funktionen:

Der **erste @-Befehl dient zur Initialisierung** der Patternroutine. Dabei werden die Blechtafelgröße, die Verschiebungen in X- und Y-Richtung zwischen den einzelnen Patternteilen sowie die Anzahl der Teile in der jeweiligen Richtung festgelegt.

Die **weiteren @-Befehle** dienen als Start-, Ende- bzw. Zwischen-**Sprungziele** der jeweiligen Programmschleifen.

Bei der SINUMERIK 810N/820N ist eine gleichsinnige und eine määnderförmige Patternbearbeitung möglich.

11.12.3.1 Numerierung der Patternteile

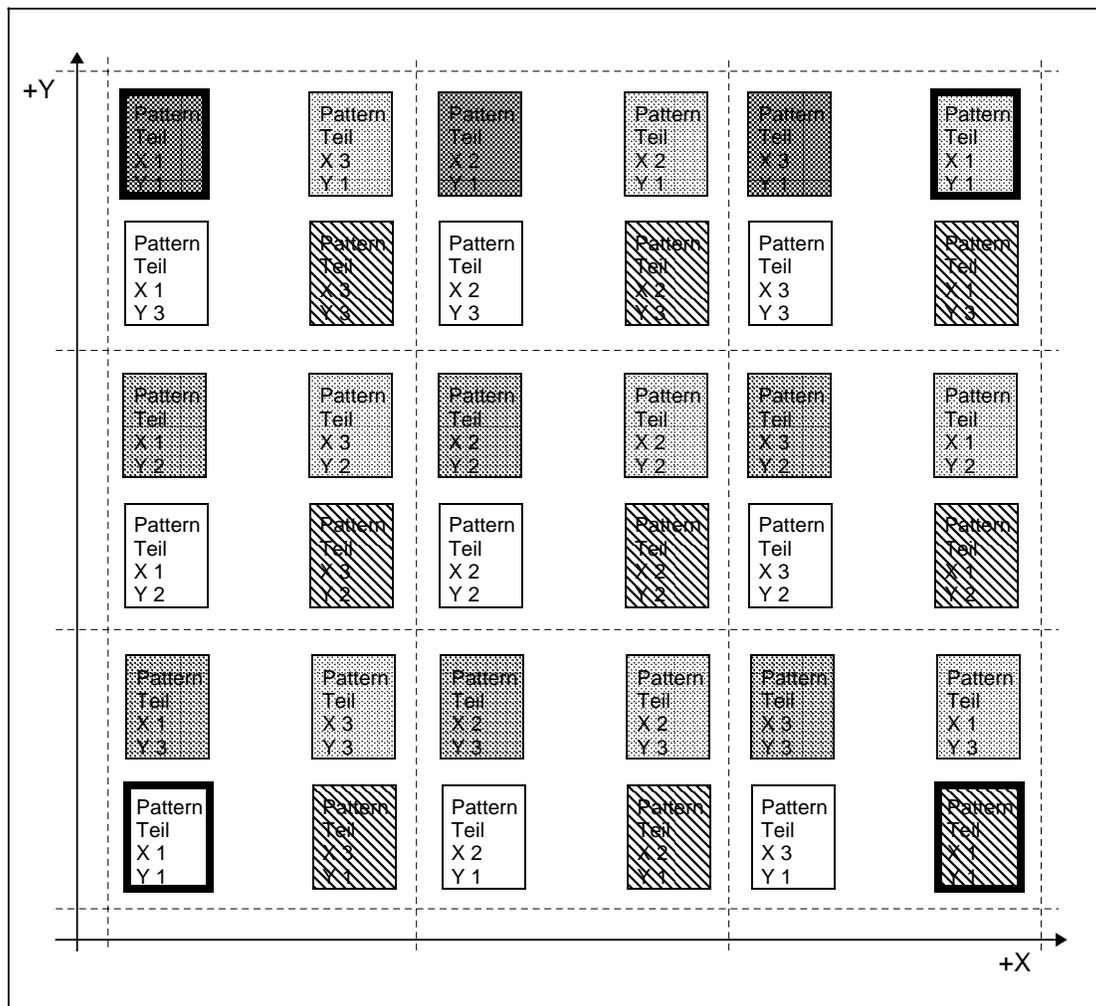
Die Numerierung der Patternteile beginnt immer am Teil 1/1, wobei die Lage dieses ersten Teils zum Koordinatensystem der X- und Y-Achse über Maschinendatum projektierbar ist.

Für die Festlegung von Teil 1 / 1 gibt es vier Möglichkeiten (siehe Bild):



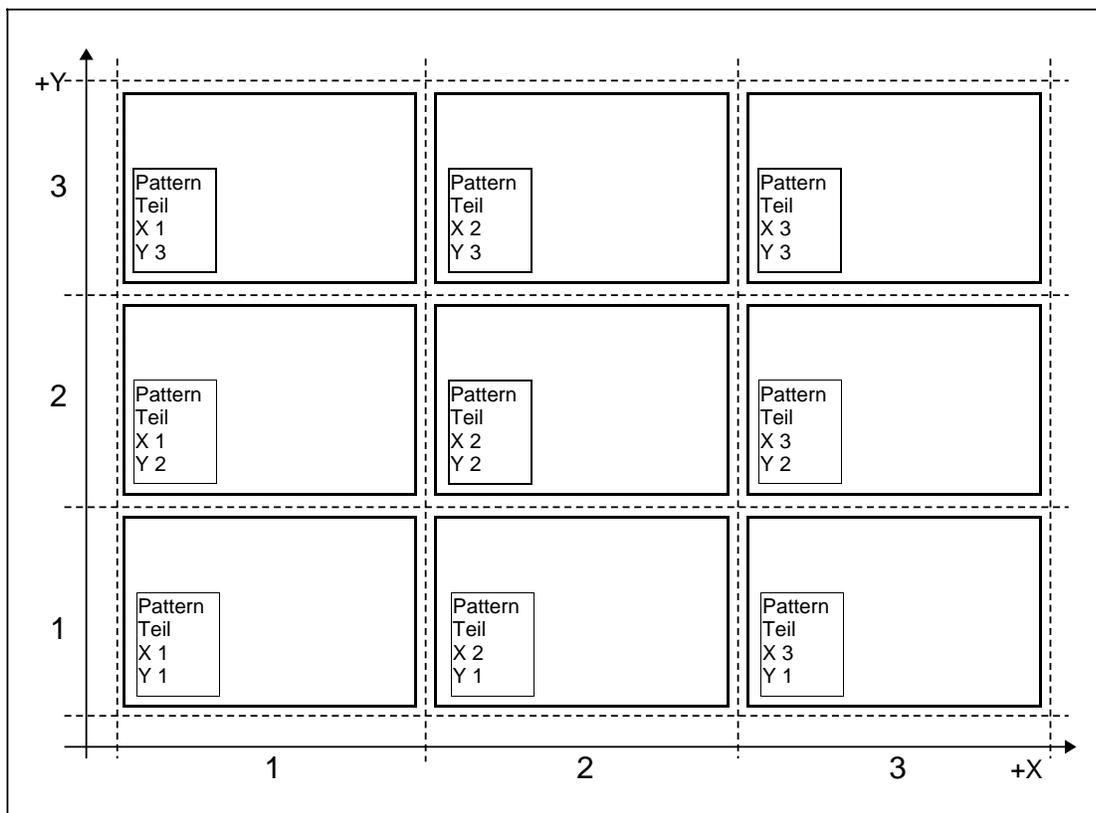
Teil 1/ ... bei X_{\min} oder X_{\max}

Teil ... /1 bei Y_{\min} oder Y_{\max}



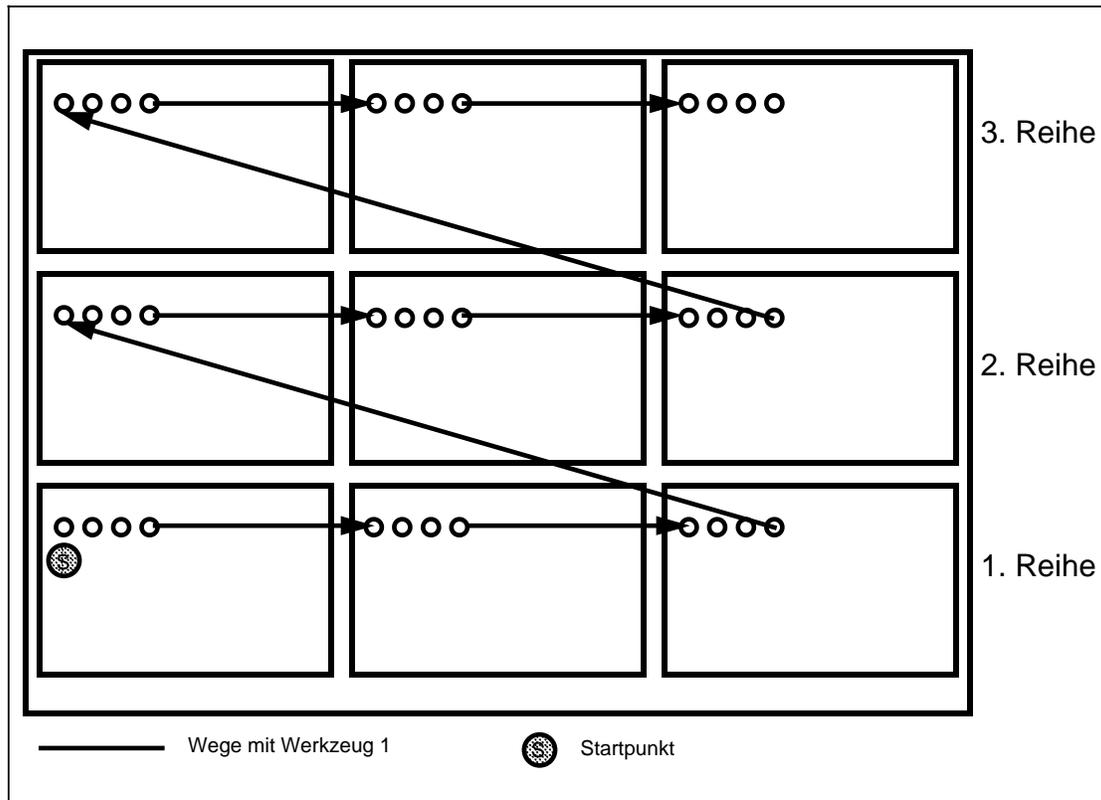
Die Bearbeitung beginnt immer am Teil 1/1. Daher wird dieses normalerweise in X-Richtung zu X_{\min} und in Y-Richtung "gegenüber" den Prätzen projiziert. Damit erfolgt die Bearbeitung zu den Prätzen hin, was der Stabilität des Blechs zugute kommt.

Als Beispiel ist hier eine Kombination gezeigt, bei der die positive Zählrichtung nach rechts (X) bzw. oben (Y) zeigt und sich die Pratten bei +Y befinden.

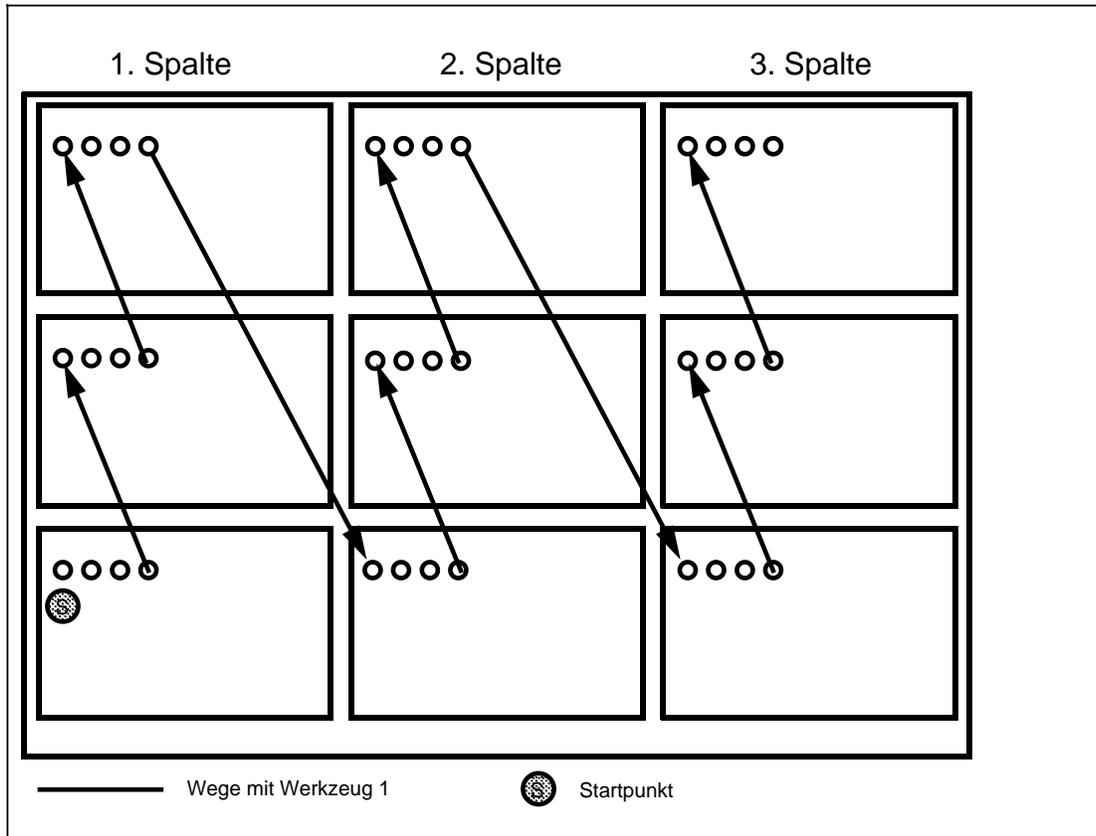


11.12.3.2 Gleichsinnige Patternbearbeitung

Bei dieser Art der Bearbeitung werden die Reihen (Spalten) in gleicher Richtung abgearbeitet.



Oder zweite Möglichkeit:



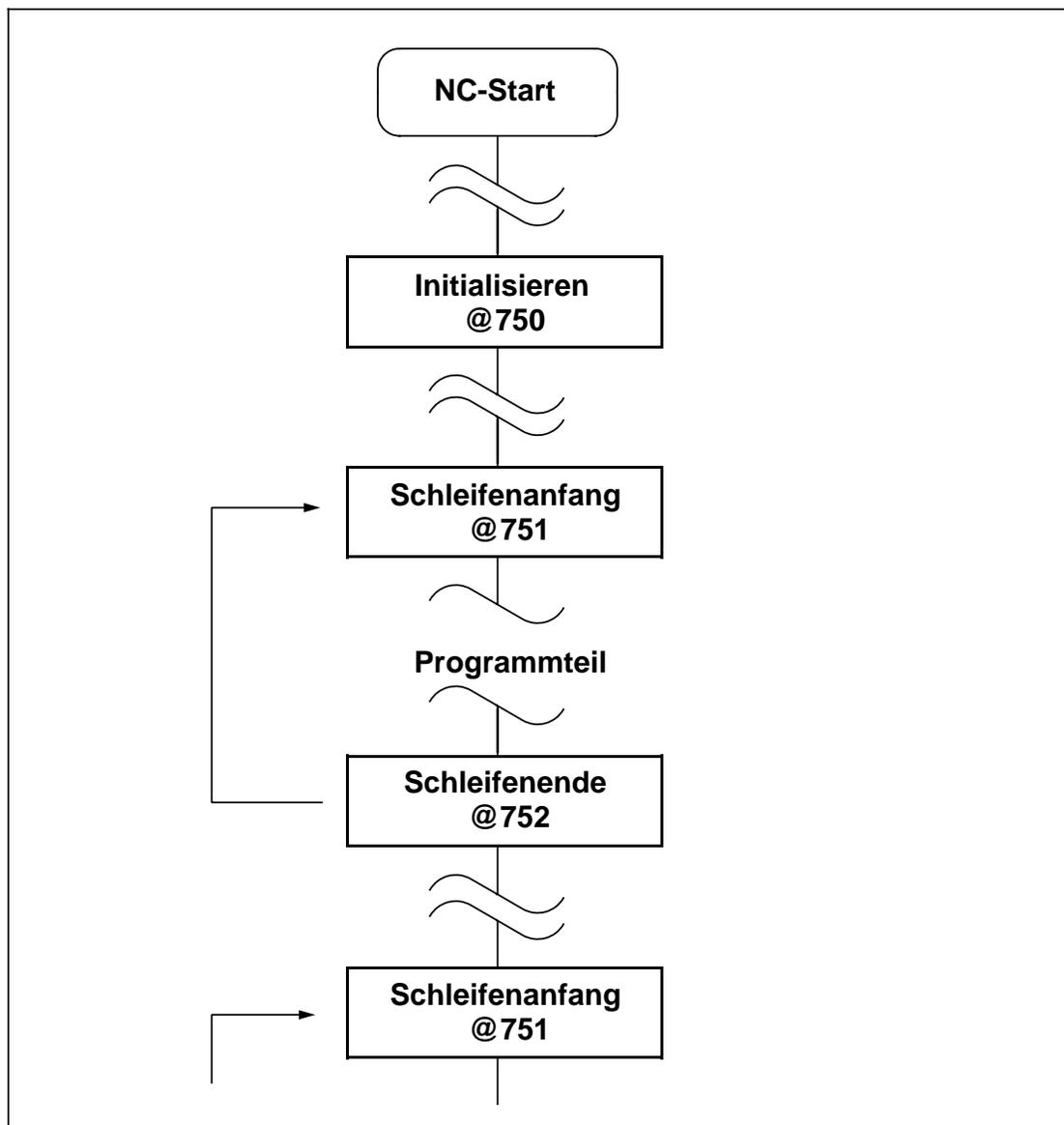
Vorteil:

Der Programmieraufwand ist minimal.

Nachteil:

Am Ende der ersten Reihe (Spalte) wird über die ganze Breite (Länge) der Blechtafel zum Anfang der zweiten Reihe (Spalte) verfahren.

Das nachfolgende Ablaufdiagramm zeigt den Schleifenbau im Teileprogramm bei gleichsinniger Patternbearbeitung.

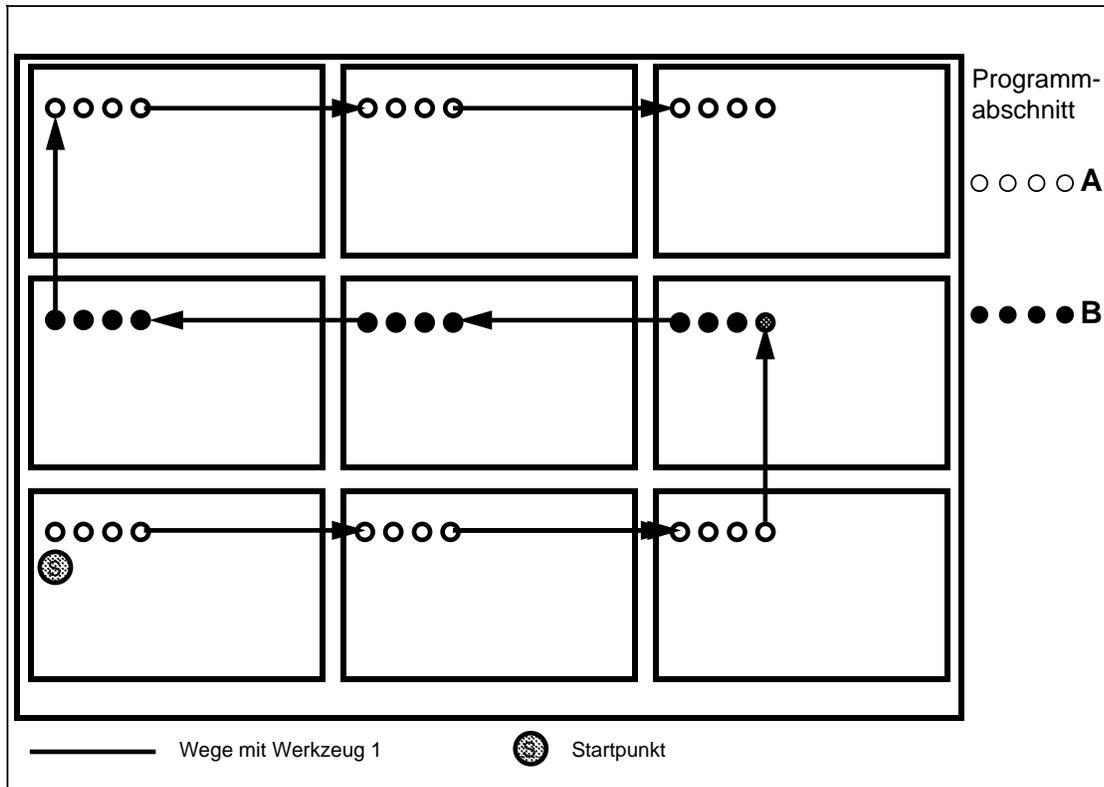


Eine ausführliche Beschreibung der einzelnen @-Befehle erfolgt in Kapitel 11.12.4.

11.12.3.3 Mäanderförmige Patternbearbeitung

Die Abarbeitung der einzelnen Reihen (Spalten) erfolgt gegensinnig.

Am Ende der ersten Reihe (Spalte) wird in die zweite Reihe (Spalte) verfahren und diese in entgegengesetzter Richtung abgearbeitet.



Vorteil:

Die Verfahrbewegung über die ganze Breite (Länge) der Blechtafel entfällt. Dadurch verkürzt sich die Bearbeitungszeit.

Nachteil:

Der Programmieraufwand ist größer als bei gleichsinniger Bearbeitung, da beide Bearbeitungsrichtungen programmiert werden müssen. Dies ist daher nur empfehlenswert bei Programmerstellung mit Programmiersystemen, bzw. wenn die zu produzierende Stückzahl so hoch ist, daß der höhere Programmieraufwand vernachlässigbar ist.

Bearbeitungsrichtung:

Richtung der Bearbeitung **in einem Patternteil**, z. B. bei einer Lochfolge.

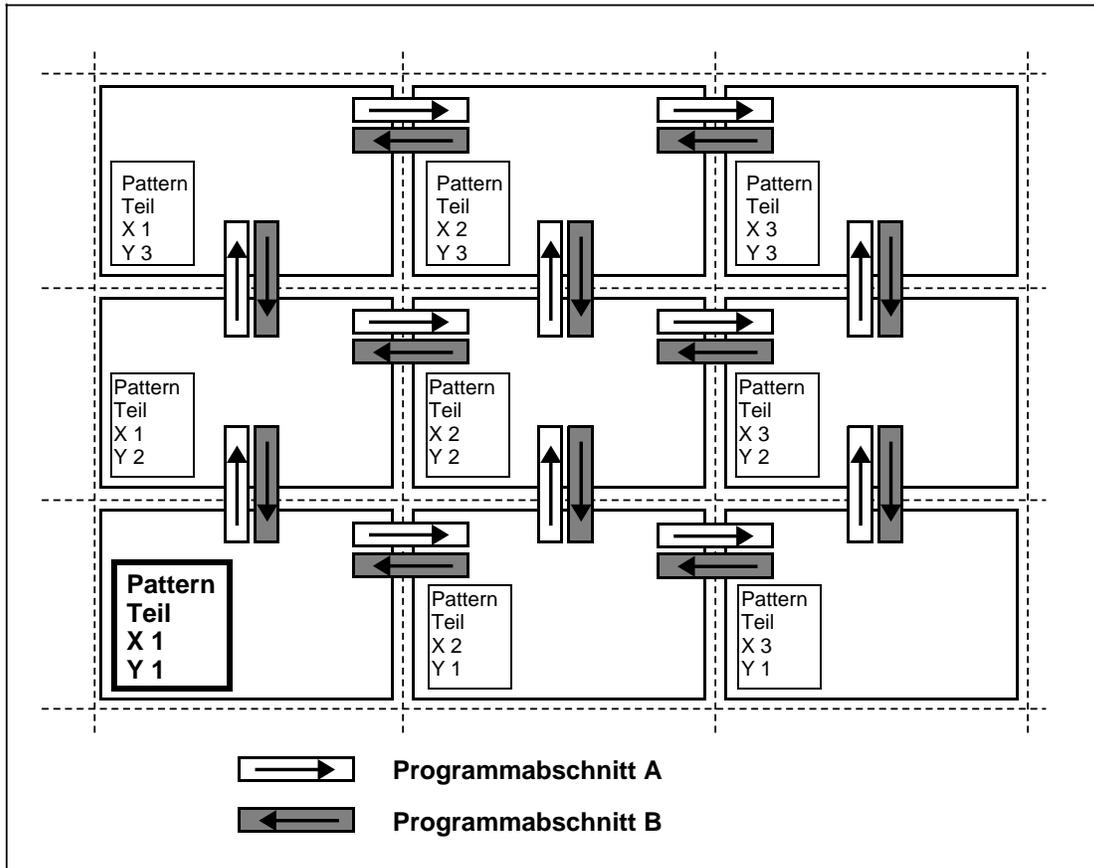
Beim Mäandern werden zur Unterscheidung die gegensinnigen Programmabschnitte mit A und B bezeichnet.

Begriffsdefinition:**Programmabschnitt A:**

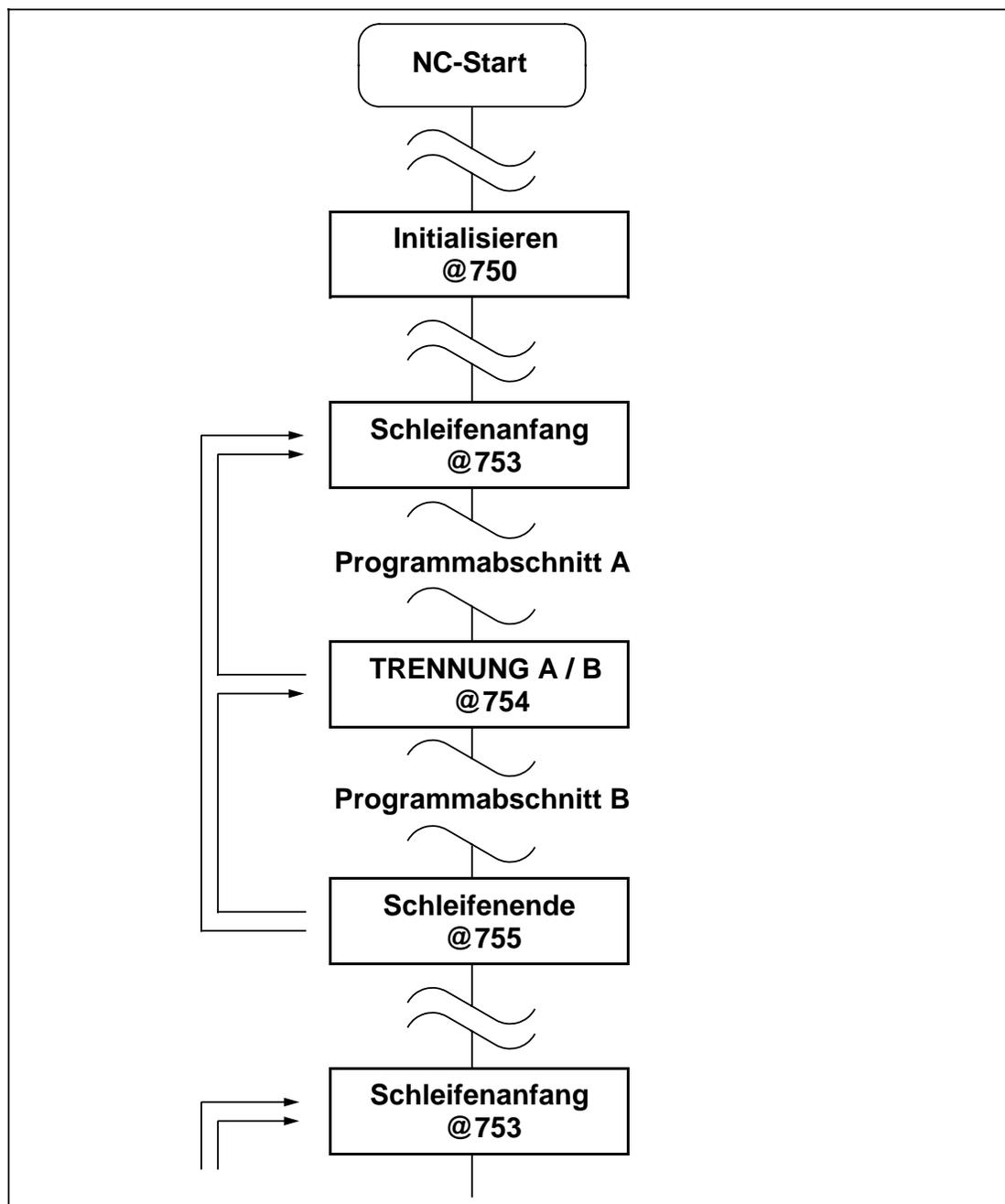
Bearbeitungsrichtung **weg vom Patternteil 1/1** (siehe Bild).
 (Numerierung der Patternteile siehe Kapitel 11.12.3.1)

Programmabschnitt B:

Bearbeitungsrichtung **zum Patternteil 1/1 hin**.



Das nachfolgende Ablaufdiagramm zeigt den Schleifenaufbau im Teileprogramm bei mäanderförmiger Patternbearbeitung.



Eine ausführliche Beschreibung der einzelnen @-Befehle erfolgt in Kapitel 11.12.4.

11.12.4 Erläuterung der @-Befehle für Pattern

Dieses Kapitel enthält eine Beschreibung der **speziellen Patternbefehle** der SINUMERIK 810N/820N.

Allgemeine Erläuterungen zu Begriffen wie z. B. Zielcode, Hauptgruppe, Operanden und Notation sind in Kapitel 11.2 dieser Programmieranleitung enthalten. Beschreibung der einzelnen Patternbefehle:

11.12.4.1 @-Befehl für Initialisierung

Initialisierung: @750 <VAR 1> <VAR 2> <WERT3> <WERT4>

Dieser Befehl dient zur Initialisierung aller nachfolgenden Patternbefehle.

<VAR 1> Verschiebung in X-Richtung zwischen den Patternteilen
 <VAR 2> Verschiebung in Y-Richtung zwischen den Patternteilen
 <WERT 3> Anzahl Patternteile in X-Richtung
 <WERT 4> Anzahl Patternteile in Y-Richtung

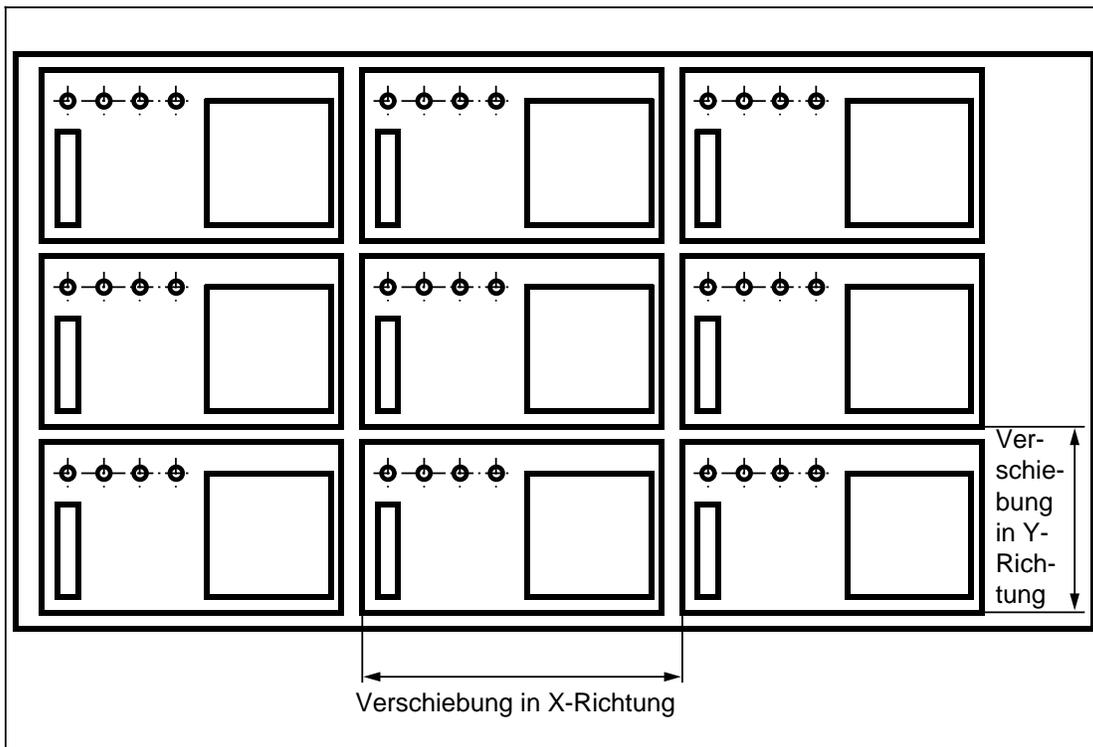
Bedeutung:

<VAR> indirekte Wertangabe (R-Parameter oder Pointer)
 <WERT> gemischte Wertangabe (Konstante, R-Parameter oder Pointer)

Wertebereich:

<VAR> und "Verschiebung insgesamt": Maximalwerte wie Weginformation Linearachse, siehe Kapitel 1.9.
 <WERT> 1 ... 9999, eventuell vorhandene Nachkommastellen werden ignoriert.

Aus <WERT 3> und <WERT 4> ergibt sich die Gesamtstückzahl der produzierten Teile.



Die Schleife, die direkt auf den @750 folgt, beginnt immer am Patternteil 1/1 (Numerierung siehe Kapitel 11.12.3.1).

- 1. Patternteil in X-Richtung
- 1. Patternteil in Y-Richtung

11.12.4.2 @-Befehle für gleichsinnige Abarbeitung

Schleifenanfang:	@751	<WERT 1>
<WERT 1>	0/1	0 = erst die Patternteile in X-Richtung abarbeiten und dann in Y-Richtung 1 = erst die Patternteile in Y-Richtung abarbeiten und dann in X-Richtung

Dieser Befehl kennzeichnet den Beginn einer Programmschleife bei gleichsinniger Patternprogrammierung.

Der @751 dient als Sprungziel für den @752.

Für <WERT 1> sind nur die Zahlenwerte 0 oder 1 zulässig. Bei anderen Zahlenwerten erscheint die Fehlermeldung "3004 CL800-Fehler".

Im Satz mit @751 muß keine Satznummer programmiert werden.

Schleifenende:	@752	<WERT 1>
<WERT 1>	0/1	0 = Start der nächsten Schleife bei Teil 1/1 1 = Start der nächsten Schleife am zuletzt bearbeiteten Teil

Dieser Befehl kennzeichnet das Ende einer Programmschleife bei gleichsinniger Patternprogrammierung.

Die Vergleichsbedingungen "Anzahl der Patternteile in X- und Y-Richtung" werden bei der Initialisierung mit @750 angegeben.

Solange die Vergleichsbedingungen in der jeweiligen Richtung nicht erfüllt sind, wird zum Satz mit @751 zurückgesprungen. Gleichzeitig werden auch die Verschiebungen in X- und Y-Richtung für das nachfolgende Patternteil berechnet und zu den **programmierten Achswerten** addiert.

Für <WERT 1> sind nur die Zahlenwerte 0 oder 1 zulässig. Bei anderen Zahlenwerten erscheint die Fehlermeldung "3004 CL800-Fehler".

Im Satz mit @752 muß keine Satznummer programmiert werden.

11.12.4.3 @-Befehle für mäanderförmige Abarbeitung

Schleifenanfang:	@753	<WERT 1>
<WERT 1>	0/1	0 = erst die Patternteile in X-Richtung abarbeiten und dann in Y-Richtung 1 = erst die Patternteile in Y-Richtung abarbeiten und dann in X-Richtung

Dieser Befehl kennzeichnet den Beginn einer Programmschleife bei mäanderförmiger Patternprogrammierung.

Der @753 dient als Sprungziel für den @754 bzw. @755.

Für <WERT 1> sind nur die Zahlenwerte 0 oder 1 zulässig. Bei anderen Zahlenwerten erscheint die Fehlermeldung "3004 CL800-Fehler".

Im Satz mit @753 muß keine Satznummer programmiert werden. Der nachfolgende Satz gehört bereits zum Programmabschnitt A.

Trennung A-/B-Teil: @754

Der @754 trennt die Programmabschnitte A und B.

Für die Bearbeitung von Programmabschnitt A hat er die Funktion "Schleifenende" (also wie @752 bei gleichsinniger Abarbeitung). Ist die vorgegebene Anzahl der Teile von Programmabschnitt A erreicht, wird in Programmabschnitt B verzweigt.

Für die Bearbeitung von Programmabschnitt B dient @754

- als Rücksprungziel von @755, vergleichbar @753 mit beim Programmabschnitt A;
- falls die vorgegebene Anzahl der Teile von Programmabschnitt B erreicht ist, wird zum "Schleifenende" (Satz mit @755) verzweigt.

Die Vergleichsbedingungen "Anzahl der Patternteile in X- und Y-Richtung" werden bei der Initialisierung durch @750 angegeben. Im Satz mit @754 muß keine Satznummer programmiert werden. Der nachfolgende Satz gehört bereits zum Programmabschnitt B.

Schleifenende: @755

Dieser Befehl kennzeichnet das Ende einer Programmschleife bei mäanderförmiger Patternprogrammierung. Die Funktion "Schleifenende" wirkt auf drei unterschiedliche Arten:

- falls die vorgegebene Anzahl der Teile für Programmabschnitt B in der jeweiligen Richtung noch nicht erreicht ist, wird zum Schleifenanfang des Programmabschnitts B (Satz mit @754) verzweigt.
- Ist die Teilezahl des Programmabschnitts B erreicht und die Gesamtstückzahl noch nicht erreicht, erfolgt eine Verzweigung zum Schleifenanfang des Programmabschnitts A (Satz mit @753).
- Ist die Gesamtstückzahl erreicht, wird die Schleife beendet und zum nachfolgenden Satz verzweigt.

Die Vergleichsbedingungen "Anzahl der Patternteile in X- und Y-Richtung" werden bei der Initialisierung durch @750 angegeben.

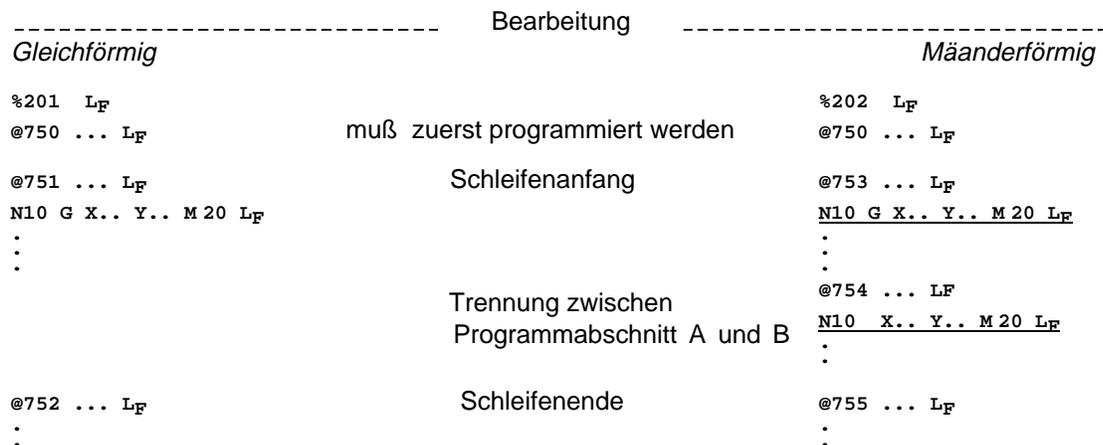
@755 dient außerdem als Sprungziel für @754. Er ist Endekennung der mäanderförmigen Bearbeitung. Im Satz mit @755 muß keine Satznummer programmiert werden.

11.12.4.4 Programmervorschrift

Eine bestimmte Reihenfolge bei der Programmierung mit @-Befehlen ist zwingend. Fehlerhafte Programmierung führt zur Fehlermeldung "CL800 Fehler".

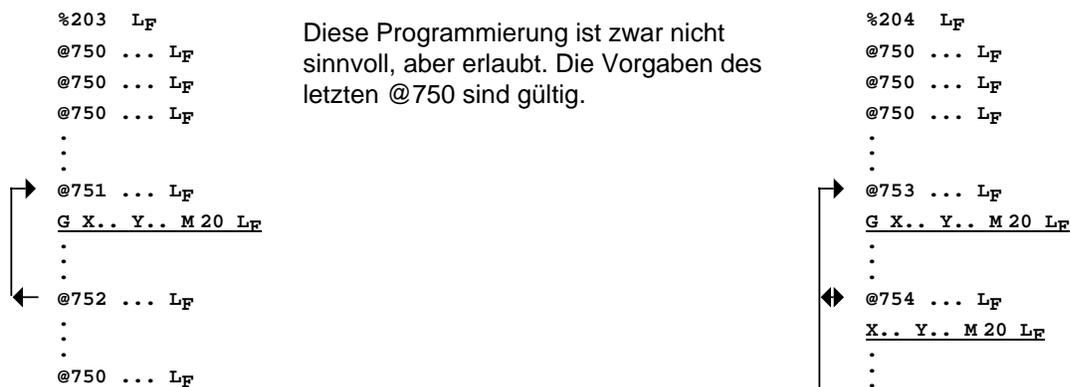
1. **Alle Schleifen müssen mit einem Positioniersatz beginnen, in dem die X- und Y-Achse im Absolutmaß (G90) programmiert ist** (unterstrichene Sätze bei den nachfolgenden Programmbeispielen).
2. **Die Klammeraffen zur Schleifenbildung (@751, @752, @753, @754, @755) müssen in einer Programmebene stehen.** Eine Aufteilung auf Haupt- und Unterprogramme bzw. auf verschiedene Unterprogramme ist nicht zulässig!
3. **Patternspezifische @-Befehle müssen allein im Satz programmiert sein.** Ausnahme: Kommentare sind zulässig, z. B. (Programmteil A).
4. **Vor @751 bzw. @753 muß zur Initialisierung @750 programmiert werden.**
5. **Während des Programmablaufes ist editieren im Teileprogramm nicht erlaubt.**

Beispiel:

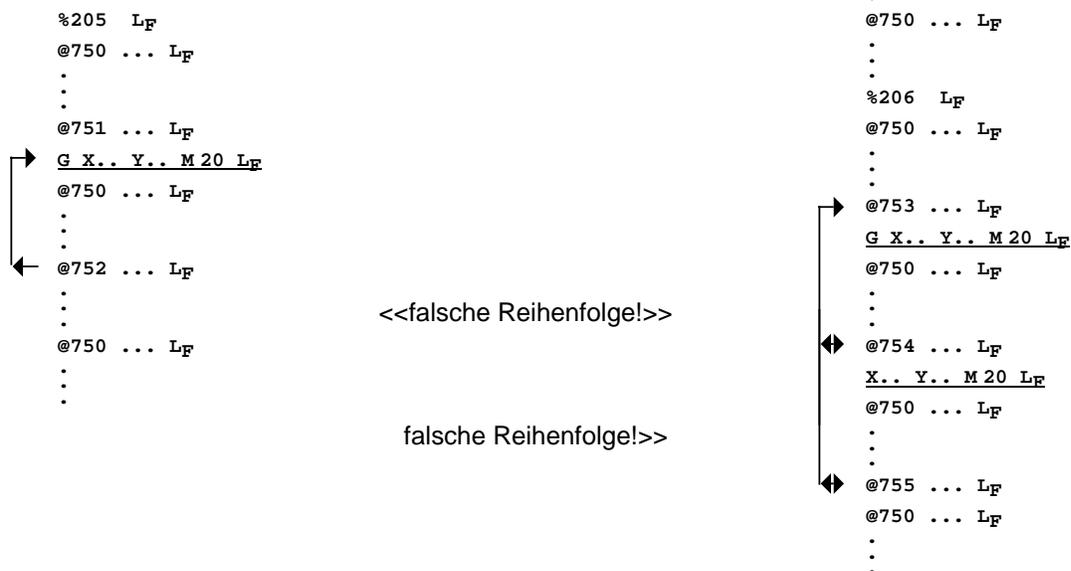


Ein @750 kann mehrmals in einem Programm programmiert werden, jedoch nicht innerhalb einer Schleife.

Beispiel:



Beispiel:



11.12.5 Fertigungslösung

Vorgaben:

1. Größe der Blechtafel = (350 x 210) mm
2. Trennwerkzeuge T10 = (5 x 20) mm
T11 = (20 x 8) mm
3. Restgitterbreite = 0 mm
4. Gefertigt wird das Teil aus Kapitel 11.12.1

"Restgitter" ist das nach Austrennen der Patternteile verbleibende Gitter. Dieses soll im Beispiel nicht entstehen.

Unter Berücksichtigung der Vorgaben können folgende Angaben für die Musteraufgabe errechnet werden:

Anzahl der Teile in X=3

Anzahl der Teile in Y=3

Gesamtverschiebung in X= 110 mm

Gesamtverschiebung in Y= 68 mm

Das Programm aus Kapitel 11.12.1.1 müßte dann wie folgt abgeändert werden (Änderungen fettgedruckt):

Hinweis: Die Angabe von **K** ohne Zahlenwert entspricht **K0** oder **K=0**.

```

%102 LF

R20=110 R21=68 LF (Errechnete Gesamtverschiebung)
@750 R20 R21 K3 K3 LF
N05 T1 LF
@751 K LF (K0 zuerst alle Teile in X-Richtung bearbeiten)
N10 G0 X11 Y50 M20 LF
N15 X47 E12 M22 LF
N20 M20 LF
@752 K LF (K0 nächste Schleife beginnt beim ersten Teil)
N25 T2 LF
@751 K LF
N30 X69 Y15 M20 LF
N35 G91 Y30 E10 M22 LF
N40 X10 LF
N45 Y-30 LF
N50 X10 LF
N55 Y30 LF
N60 X10 LF
N65 Y-30 LF
N70 G90 M20 LF
@752 K LF
N75 T3 LF
@751 K LF
N80 X11.5 Y15 M20 LF
N85 Y35 M22 LF
N90 M20 LF
@752 K LF
N95 M02 LF

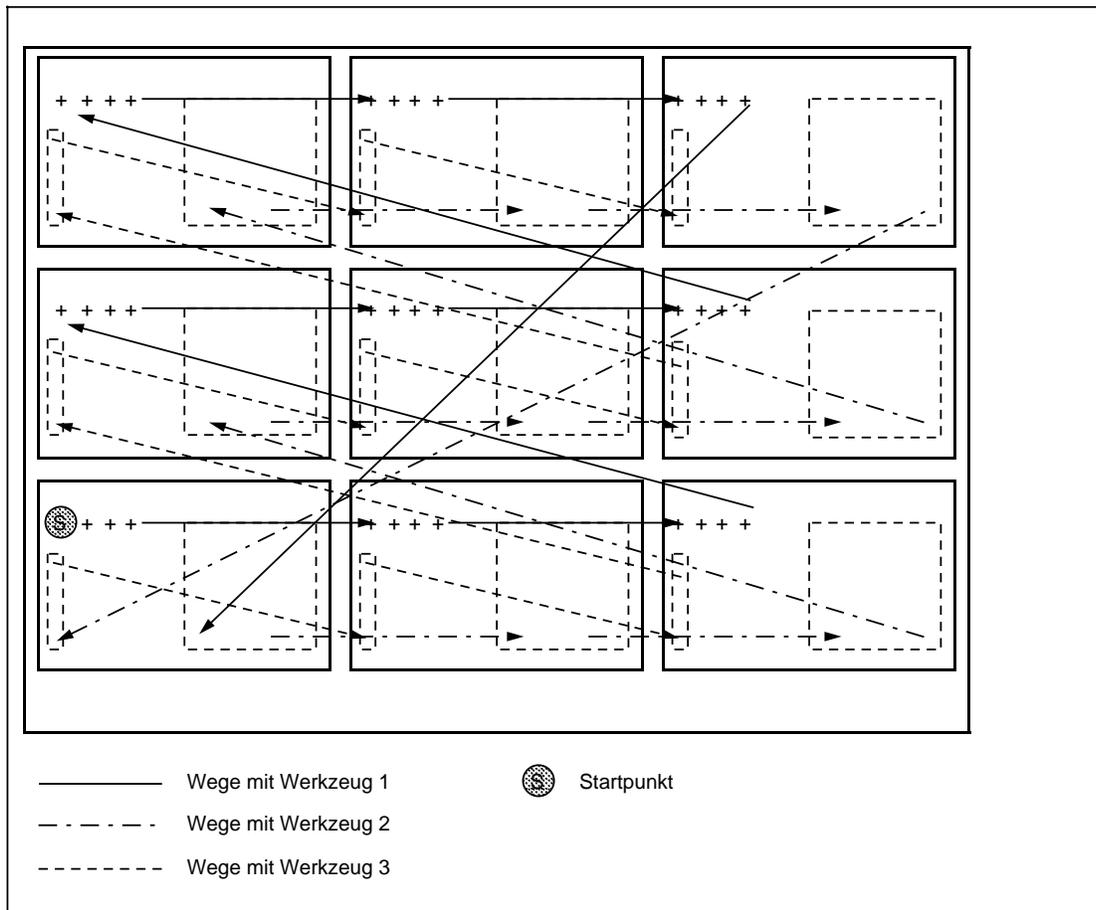
```

In diesem Beispiel wurde eine werkzeugorientierte Bearbeitung gewählt.

Hinweis:

Im Beispiel ist für die Hunderter-Dekade der Stanz-/Nibbelfunktionen der Wert 0 gewählt.

Im folgenden Bild sind die Verfahrensbewegungen dargestellt:



Bemerkung:

Um Rauigkeiten an den Blechkanten zu umgehen, sollten die Patternteile etwas in die Blechmitte verschoben werden. Die Ränder werden dann beim Trennen ebenfalls bearbeitet. Sinnvollerweise benutzt man dazu eine Nullpunktverschiebung.

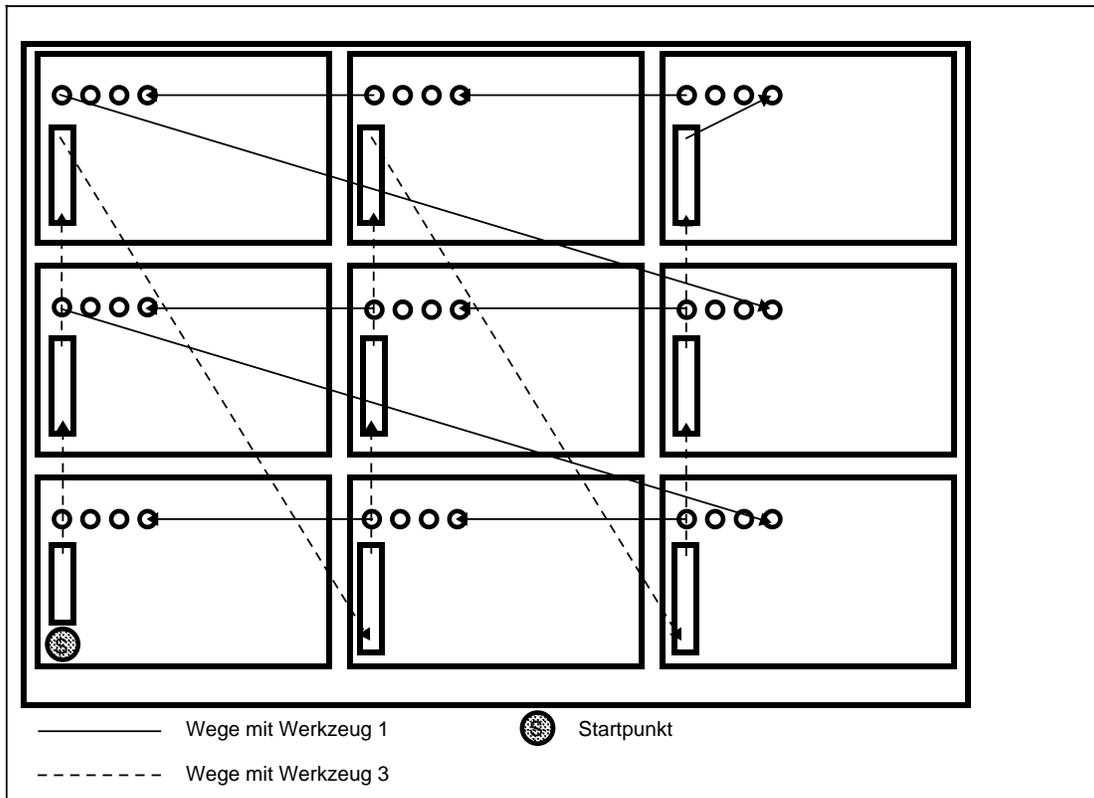
Beachte:

Für eine Simulation des Programms simultan zur Bearbeitung ist Kapitel 11.12.8.3 (**Kanal-spezifische Anwahl der Nullpunktverschiebungen**) zu beachten!

11.12.5.1 Wegoptimierung bei gleichsinniger Bearbeitung

Zusätzlich kann vom Programmierer (oder Programmiersystem) mit minimalem Aufwand eine Optimierung der Verfahrenswege vorgenommen werden. Hierzu dient der zusätzliche Operand des @752: Wird diesem Wert 1 zugewiesen, wird damit eine Positionierbewegung diagonal über die Blechtafel am Ende der ersten Bearbeitung unterbunden (siehe Bild). Die nachfolgende Bearbeitung beginnt am zuletzt bearbeiteten Teil und nicht wieder am ersten.

Außerdem kann eventuell (wie auch im vorliegenden Beispiel) eine weitere Optimierung durch Ändern der Werkzeugreihenfolge oder durch Änderung der Bearbeitungsrichtung in einer Programmschleife erreicht werden.



In obigem Bild ist eine Möglichkeit gezeigt, wie die optimierte Bearbeitung aussehen kann. Die Bewegungen mit Werkzeug 2 sind zur besseren Übersichtlichkeit weggelassen.

Das modifizierte Programm sieht dann wie folgt aus (Ergänzungen gegenüber %102 fettgedruckt):

```
%103 LF

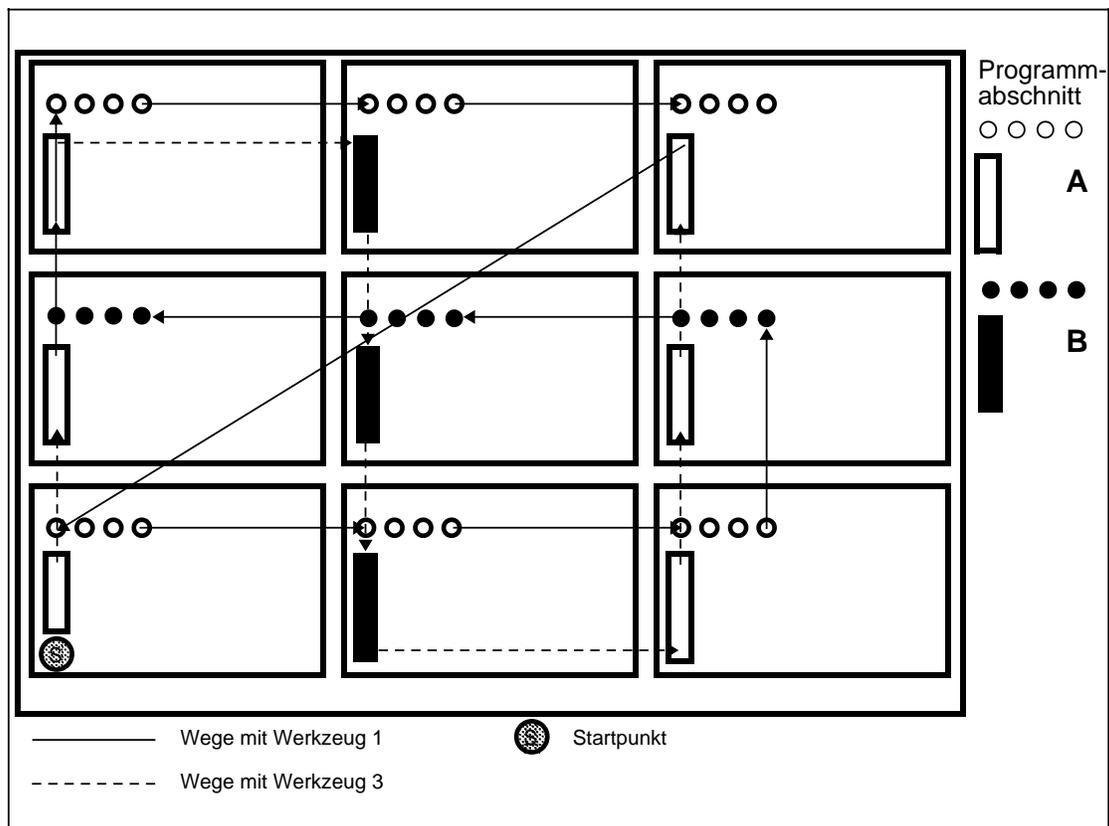
R20=110 R21=68 LF
@750 R20 R21 K3 K3 LF
N75 T3 LF
→ @751 K1 LF (K1 zuerst alle Teile in Y-Richtung bearbeiten)
N80 G0 X11.5 Y15 M20 LF
N85 Y35 E10 M22 LF
N90 M20 LF
@752 K1 LF (K1 nächste Schleife beginnt am letzten Teil)
N05 T1 LF
→ @751 K LF (K0 Bearbeitung in X-Richtung)
N10 X47 Y50 M20 LF
N15 X11 E12 M22 LF
N20 M20 LF
@752 K LF
N25 T2 LF
→ @751 K LF
N30 X69 Y15 M20 LF
N35 G91 Y30 E10 M22 LF
N40 X10 LF
N45 Y-30 LF
N50 X10 LF
N55 Y30 LF
N60 X10 LF
N65 Y-30 LF
N70 G90 M20 LF
@752 K LF
N95 M02 LF
```

Hinweis:

Im Beispiel ist für die Hunderter-Dekade der Stanz-/Nibbelfunktionen der Wert 0 gewählt worden.

11.12.5.2 Wegoptimierung bei mäanderförmiger Bearbeitung

Durch zusätzlichen Programmcode kann gegenüber der gleichsinnigen Bearbeitung eine weitere Wegoptimierung erreicht werden. Hierzu muß der Programmierer (oder das Programmiersystem) festlegen, welche Bearbeitungen in zwei Programmabschnitten aufteilbar sind, die in entgegengesetzter Richtung abgearbeitet werden können.



Die Bewegungen mit Werkzeug 2 sind zur besseren Übersichtlichkeit weggelassen.

Das modifizierte Programm sieht dann wie folgt aus (Ergänzungen gegenüber %103 fettgedruckt):

```

%104 LF
R20=110 R21=68 LF
@750 R20 R21 K3 K3 LF
N75 T3 LF
→ @753 K1 LF
( Programmabschnitt A ) LF
N80 G0 X11.5 Y15 M20 LF
N85 Y35 E10 M22 LF
N90 M20 LF
→ @754 LF
( Programmabschnitt B ) LF
N80 X11.5 Y35 M20 LF
N85 Y15 E10 M22 LF
N90 M20 LF
→ @755 LF

```

```

N05 T1 LF
→ @753 K LF
  ( Programmabschnitt A ) LF
N10 X11 Y50 M20 LF
N15 X47 E12 M22 LF
N20 M20 LF
→ @754 LF
  ( Programmabschnitt B ) LF
N10 X47 Y50 M20 LF
N15 X11 E12 M22 LF
N20 M20 LF
→ @755 LF
N25 T2 LF
→ @753 K LF
  ( Programmabschnitt A ) LF
N30 X69 Y15 M20 LF
N35 G91 Y30 E10 M22 LF
N40 X10 LF
N45 Y-30 LF
N50 X10 LF
N55 Y30 LF
N60 X10 LF
N65 Y-30 LF
N70 G90 M20 LF
→ @754 LF
  ( Programmabschnitt B ) LF
N30 X99 Y15 M20 LF
N35 G91 Y30 E10 M22 LF
N40 X-10 LF
N45 Y-30 LF
N50 X-10 LF
N55 Y30 LF
N60 X-10 LF
N65 Y-30 LF
N70 G90 M20 LF
→ @755 LF
N95 M02 LF
```

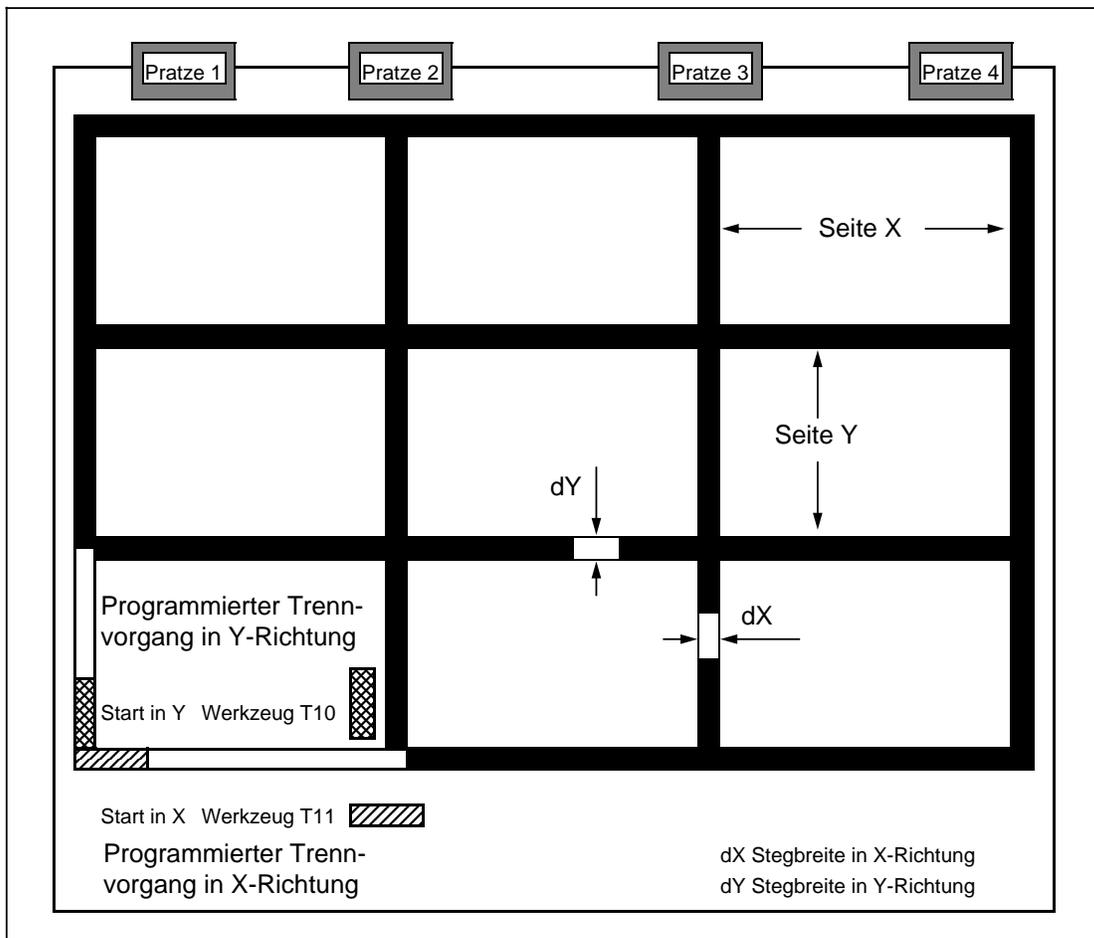
Hinweis:

Im Beispiel ist für die Hunderter-Dekade der Stanz-/Nibbelfunktionen der Wert 0 gewählt worden.

11.12.5.3 Trennen

Am Ende der Bearbeitung sind die Patternteile aus dem Blech herauszutrennen.

Dafür gibt es mehrere Möglichkeiten. Für die Beispielaufgabe soll ein Weg gezeigt werden, der die Pattern-@-Befehle anwendet.



Gestartet wird der **Trennvorgang in Y-Richtung am Teil 1/1** (im Bild links unten):

Startpunkt ist die linke Seite des ersten Patternteils.

Startpunkt in X= $-1/2 \cdot$ (Werkzeugabmessung in X von T10)

Startpunkt in Y= $-1/2 \cdot dY + 1/2 \cdot$ (Werkzeugabmessung in Y von T10)

Endpunkt in X= wie Startpunkt in X

Endpunkt in Y= $\text{SeiteY} + 1/2 \cdot dY - 1/2 \cdot$ (Werkzeugabmessung in Y von T10)

Anzahl der Schleifen in X= (Anzahl Patternteile in X)+1

Anzahl der Schleifen in Y= (Anzahl Patternteile in Y)

Als nächstes folgt der **Trennvorgang in X-Richtung**, der ebenfalls am Teil 1/1 beginnt:

Startpunkt ist die untere Kante des linken Patternteils in der ersten Reihe.

Startpunkt in X= $-1/2 \cdot dX + 1/2 \cdot$ (Werkzeugabmessung in X von T11)

Startpunkt in Y= $-1/2 \cdot$ (Werkzeugabmessung in Y von T11)

Endpunkt in X= Seite X + $1/2 \cdot dX - 1/2 \cdot$ (Werkzeugabmessung in X von T11)

Endpunkt in Y= wie Startpunkt in Y

Anzahl der Schleifen in X= (Anzahl der Patternteile in X)

Anzahl der Schleifen in Y= (Anzahl der Patternteile in Y)+1

Trennvorgang als Unterprogramm:

Um das Teileprogramm übersichtlich zu gestalten, soll der Trennvorgang als Unterprogramm programmiert werden. Im eigentlichen Teileprogramm erfolgt dann lediglich die Parametrierung und der Aufruf des Trennunterprogramms.

Als Eingabeparameter für das Unterprogramm werden die R-Parameter R20 ... R29 benutzt, als Rechenparameter die Parameter ab R50.

Vorgaben:

Es gelten die Vorgaben wie im Beispiel aus Kapitel 11.11.5.

R20 = Verschiebung in X-Richtung

R21 = Verschiebung in Y-Richtung

R22 = X-Trennwerkzeug auf Platz T10 X-Abmessung
 entspricht hier dX, da Restgitterbreite=0 mm

R23 = X-Trennwerkzeug auf Platz T10 Y-Abmessung

R24 = Y-Trennwerkzeug auf Platz T11 X-Abmessung

R25 = Y-Trennwerkzeug auf Platz T11 Y-Abmessung
 entspricht hier dY, da Restgitterbreite=0 mm

R26 = Anzahl der Patternteile in X-Richtung

R27 = Anzahl der Patternteile in Y-Richtung

R28 = Patternteil Seite X

R29 = Patternteil Seite Y

Trennunterprogramm:

L10 (Trennunterprogramm) L_F

(Berechnungen für Trennvorgang in Y-Richtung)

R50=R26+1 L_F

@750 R20 R21 R50 R27 L_F

(Startpunkt in X)

R50=-R22/2 L_F

-1/2·(Werkzeugabmessung in X T10)

(Startpunkt in Y)

R51=R23/2 L_F

R52=R25/2 L_F

R51=R51-R52 L_F

1/2·(Werkzeugabmessung in X T10) - 1/2·dY

(Endpunkt in Y)

R53=R23/2 L_F

R52=R52+R29 L_F

R52=R52-R53 L_F

1/2·dY+SeiteY

-1/2·(Werkzeugabmessung in Y T10)



(Trennen in Y-Richtung)

```

T10 LF
→ @751 K1 LF
N05 G0 X= R50 Y= R51 M20 LF
N10 X= R50 Y= R52 E= R23 M22 LF
N15 M20 LF
@752 K0 LF

```

(Berechnungen für Trennvorgang in X-Richtung)

```

R50=R27+1
@750 R20 R21 R26 R50 LF
(Startpunkt in X)
R50=R24/2
R51=R22/2
R50=R50-R51 LF
(Startpunkt in Y)
R51=R25/2 LF
(Endpunkt in X)
R52=R22/2
R52=R52+R28
R53=R24/2
R52=R52-R53 LF

```



$1/2 \cdot (\text{Werkzeugabmessung in X T11}) - 1/2 \cdot dX$
 $-1/2 \cdot (\text{Werkzeugabmessung in Y T11})$
 $1/2 \cdot dX$
 $+ \text{SeiteX}$
 $-1/2 \cdot (\text{Werkzeugabmessung in X T11})$

(Trennen in X-Richtung)

```

T11 LF
→ @751 K0 LF
N30 X= R50 Y= R51 M20 LF
N35 X= R52 Y= R51 E= R24 M22 LF
N40 M20 LF
@752 K0 LF
M17 LF

```

Hinweis:

Im Beispiel ist für die Hunderter-Dekade der Stanz-/Nibbelfunktionen der Wert 0 gewählt worden.

Das folgende Beispiel zeigt die Parametrierung und den Aufruf dieses Unterprogramms im Hauptprogramm. Das Hauptprogramm %103 aus Kapitel 11.12.5.1 wurde modifiziert, die Ergänzungen zum Aufruf des Trennunterprogramms sind fettgedruckt:

```
%105 LF  
R20= 110 R21= 68 (Verschiebung in X/Y)
```

(**Parameterbelegung für Trennunterprogramm**)

```
R22= 5 R23= 20 (Y-Trennwerkzeug T10 Abmessungen X/Y)  
R24= 20 R25= 8 (X-Trennwerkzeug T11 Abmessungen X/Y)  
R26= 3 R27= 3 (Anzahl der Patternteile in X/Y)  
R28= 105 R29= 60 LF (Patternteile Seite X/Y)  
@750 R20 R21 R26 R27 LF  
N75 T3 LF  
@751 K1 LF  
N80 G0 G56 X11.5 Y15 M20 LF  
N85 Y35 M22 LF  
N90 M20 LF  
@752 K1 LF  
N05 T1 LF  
@751 K LF  
N10 X47 Y50 M20 LF  
N15 X11 E12 M22 LF  
N20 M20 LF  
@752 K LF  
N25 T2 LF  
@751 K LF  
N30 X69 Y15 M20 LF  
N35 G91 Y30 E10 M22 LF  
N40 X10 LF  
N45 Y-30 LF  
N50 X10 LF  
N55 Y30 LF  
N60 X10 LF  
N65 Y-30 LF  
N70 G90 M20 LF  
@752 K LF
```

(**Trennunterprogramm Aufruf**)

```
N500 L10 P1 LF  
N95 M02 LF
```

Hinweis:

Im Beispiel ist für die Hunderter-Dekade der Stanz-/Nibbelfunktionen der Wert 0 gewählt worden.

G56:

Die in G56 hinterlegten Werte verschieben die Patternteile vom Blechtafelrand weg zur Blechmitte hin, um Rauigkeiten und Winkelfehler der Blechtafelkanten zu eliminieren.

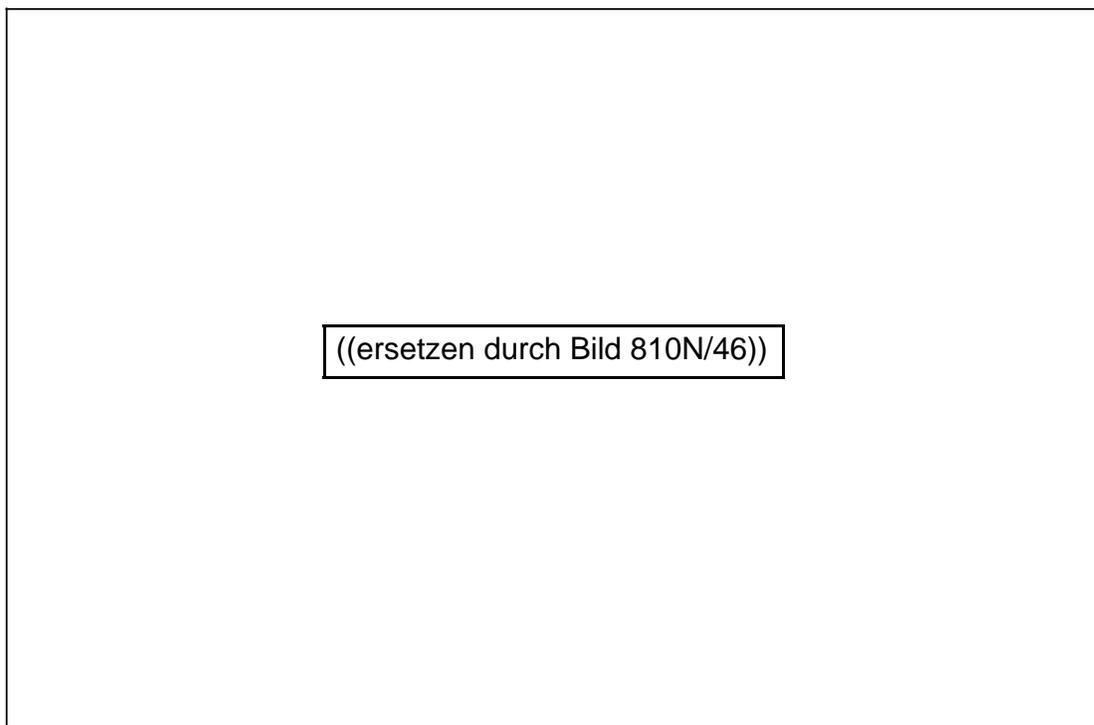
Anmerkung:

Für eine Simulation des Programms simultan zur Bearbeitung ist Kapitel 11.12.8.3 (**Kanal-spezifische Anwahl der Nullpunktverschiebungen**) zu beachten!

11.12.6 Anzeige Pattern-Teil im AUTOMATIK-Bild

Die Numerierung der Patternteile ist im Kapitel 11.12.3.1 beschrieben.

Das Bild "Aktueller Satz" wurde um die Anzeige des aktuellen Patternteils erweitert.



Um im Teileprogramm eine Auswertung zu ermöglichen, welches Patternteil momentan in Bearbeitung ist, kann die Nummer des aktuellen Teils in R-Parametern abgefragt werden:

R500 = aktuelles Patternteil in X-Richtung

R501 = aktuelles Patternteil in Y-Richtung

Da steuerungsintern mehrere Sätze im Zwischenspeicher "vorverarbeitet" werden, entspricht der Zählerstand in R500 und R501 dem bereits in der Dekodierung befindlichen Patternteil. Für Abfragen im laufenden NC-Teileprogramm sind also diese Parameter zu verwenden.

Soll das momentan tatsächlich (mechanisch) in Bearbeitung befindliche Patternteil ermittelt werden, ist dies aus den R-Parametern R504 und R505 ersichtlich:

R504 = Patternteil in Bearbeitung in X-Richtung

R505 = Patternteil in Bearbeitung in Y-Richtung

Die Anzeigen in R500 bzw. R501 sind also gegenüber R504 und R505 voraus.

11.12.7 Satzvorlauf

11.12.7.1 Satzvorlauf im AUTOMATIK-Betrieb

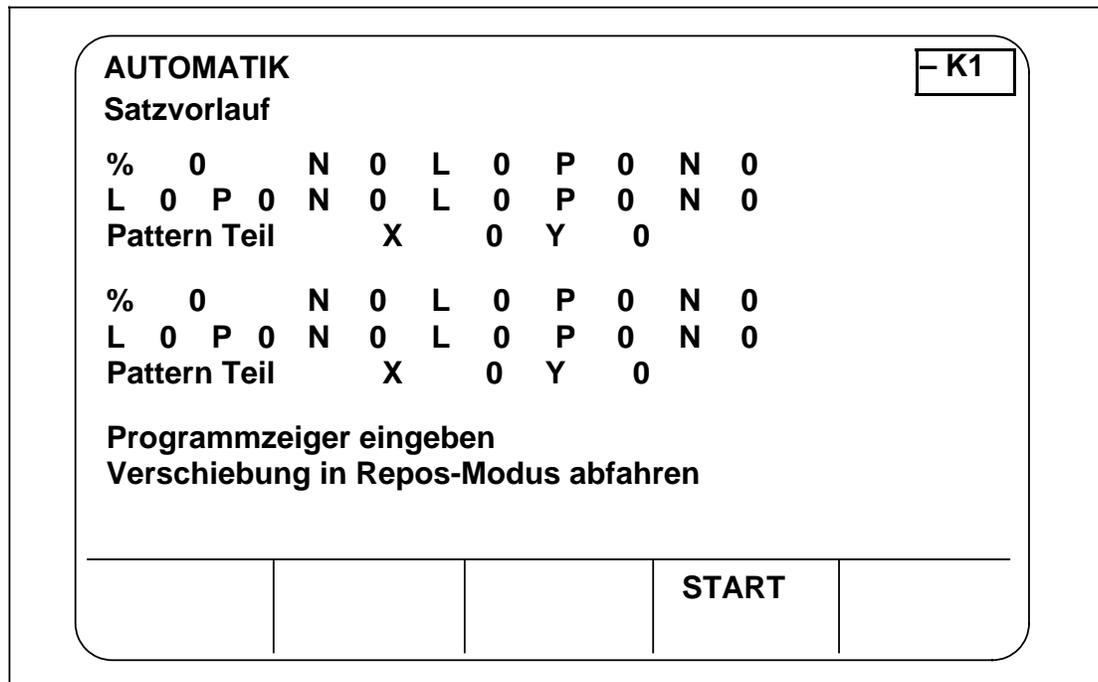
Bisher waren die Suchkriterien bei Satzvorlauf:

- Satznummer im Hauptprogramm
- Satznummern in den aufgerufenen Unterprogrammen bis zur dritten Unterprogrammebene
- Programmnummern der aufgerufenen Unterprogramme
- Durchlaufzahlen der aufgerufenen Unterprogramme

Für die Patternbearbeitung wurde ein weiteres Suchkriterium eingeführt:

Zahl der Patternteile in X- und Y-Richtung.

Durch Angabe der Zahl der Patternteile ist es sehr einfach, die Bearbeitung bei jedem beliebigen Patternteil wiederaufzunehmen.



Bei einer Programmunterbrechung durch "RESET" wird automatisch der zuletzt gültige Status in die Vorschlagsparameter als Suchkriterium übertragen.

Wichtig:

Soll der Satzvorlauf auf einen Satz in einer Pattern-Schleife (also zwischen @751 und @752 bzw. zwischen @753, @754 und @755) gemacht werden, muß die Vorgabe für das Patternteil in X- und Y-Richtung **mindestens 1** sein **und darf nicht größer als die programmierte maximale Anzahl** (<Wert3> und <Wert4> von @750) sein.

Im ersten Verfahrssatz in einer Schleife müssen unbedingt **beide Achswerte** im Absolutmaß (G90) programmiert werden, da sich in den Schleifen die achsspezifischen Verschiebungen zum programmierten Achswert addieren.

Bei der mäanderförmigen Bearbeitung erfolgt die Verzweigung automatisch in den richtigen Programmabschnitt, auch wenn im Programmabschnitt A und B identische Satznummern vergeben wurden.

Satzvorlauf bei grafischer Simulation

Der Satzvorlauf in der Simulation unterscheidet sich nicht vom hier beschriebenen Satzvorlauf (siehe Kapitel 11.8.2).

11.12.7.2 Beispiele zum Satzvorlauf

Beispiel:

Im folgenden Programm wird dreimal der Satz N10 programmiert. Dabei hat jeder Satz mindestens ein eindeutiges Suchkriterium.

```

%110 LF
@750 R1 R2 K3 K2 LF
                                κ3 ... 3 Patternteile in X-Richtung
                                κ2 ... 2 Patternteile in Y-Richtung
1)  N10 M100 LF
    @751 K0 LF
2)  N10 M101 LF
    @752 K0 LF
    :
    :
    @750 R1 R2 K1 K5 LF
                                κ1 ... 1 Patternteil in X-Richtung
                                κ5 ... 5 Patternteile in Y-Richtung

    @751 K0 LF
3)  N10 M102 LF
    @752 K0 LF
    M02 LF

```

1. Vorgabe

Satzvorlauf

```

%110          N .. 10 L .. 0 P .. 0 N ... 0
L .. 0 P.0    N ... 0 L .. 0 P .. 0 N ... 0
Pattern Teil  x  0  Y  0
              gefundener Zielsatz=1)

```

2. Vorgabe

Satzvorlauf

```

%110          N .. 10 L .. 0 P .. 0 N ... 0
L .. 0 P.0    N ... 0 L .. 0 P .. 0 N ... 0
Pattern Teil  x  2  Y  2
              gefundener Zielsatz=2)

```

3. Vorgabe

Satzvorlauf

```

%110          N .. 10 L .. 0 P .. 0 N ... 0
L .. 0 P.0    N ... 0 L .. 0 P .. 0 N ... 0
Pattern Teil  x  1  Y  3
              gefundener Zielsatz=3)

```

4. Vorgabe

Satzvorlauf

```

%110          N .. 10 L .. 0 P .. 0 N ... 0
L .. 0 P.0    N ... 0 L .. 0 P .. 0 N ... 0
Pattern Teil  x  2  Y  4

```

Dieser Satz kann nicht gefunden werden.

Der Alarm "**3012 N10 Satz im Speicher nicht vorhanden**" erscheint.

Einschränkung:

Soll ein Vorlauf auf den Satz @751 bzw. @752 erfolgen, **muß** die Satznummer in diesem Satz an **erster Stelle** stehen!

Beispiel:

```
%111 LF
@750 R1 R2 K0 K0 LF
:
.
.
@751 N50 LF
:
:
M02 LF

%112 LF
@750 R1 R2 K K LF
:
.
.
N50 @751 LF
:
:
M02 LF
```

Satzvorlauf auf 'N50'
Satz wird nicht gefunden!

Satzvorlauf auf 'N50'
Satz wird gefunden!

Beispiel für Satzvorlauf bei mäanderförmiger Programmbearbeitung:

```
%113 LF
@750 R1 R2 K3 K2 LF
k3 ... 3 Patternteile in X-Richtung
k2 ... 2 Patternteile in Y-Richtung

@753 K0 LF
( Programmabschnitt A )
1) N20 M101 LF
@754 LF
( Programmabschnitt B )
2) N20 M102 LF
@755 LF
M02 LF
```

1. Vorgabe

Satzvorlauf

```
%113          N .. 20 L .. 0 P .. 0 N ... 0
L .. 0 P.0     N ... 0 L .. 0 P .. 0 N ... 0
Pattern Teil   x 2  Y 1
gefundener Zielsatz=1)
```

2. Vorgabe

Satzvorlauf

```
%113          N .. 20 L .. 0 P .. 0 N ... 0
L .. 0 P.0     N ... 0 L .. 0 P .. 0 N ... 0
Pattern Teil   x 2  Y 2
gefundener Zielsatz=2)
```

11.12.8 Grafische Simulation

11.12.8.1 Grafische Simulation bei Patternbearbeitung

Da die Verschiebungen zwischen den einzelnen Patternteilen kanalspezifisch berücksichtigt werden, kann eine Patternbearbeitung in Kanal 1 und simultan dazu im Kanal 3 eine Simulation gestartet werden. Auch sind alle Anzeigen des aktuellen Patternteils kanalspezifisch, so daß auch hier keine Überschneidungen entstehen.

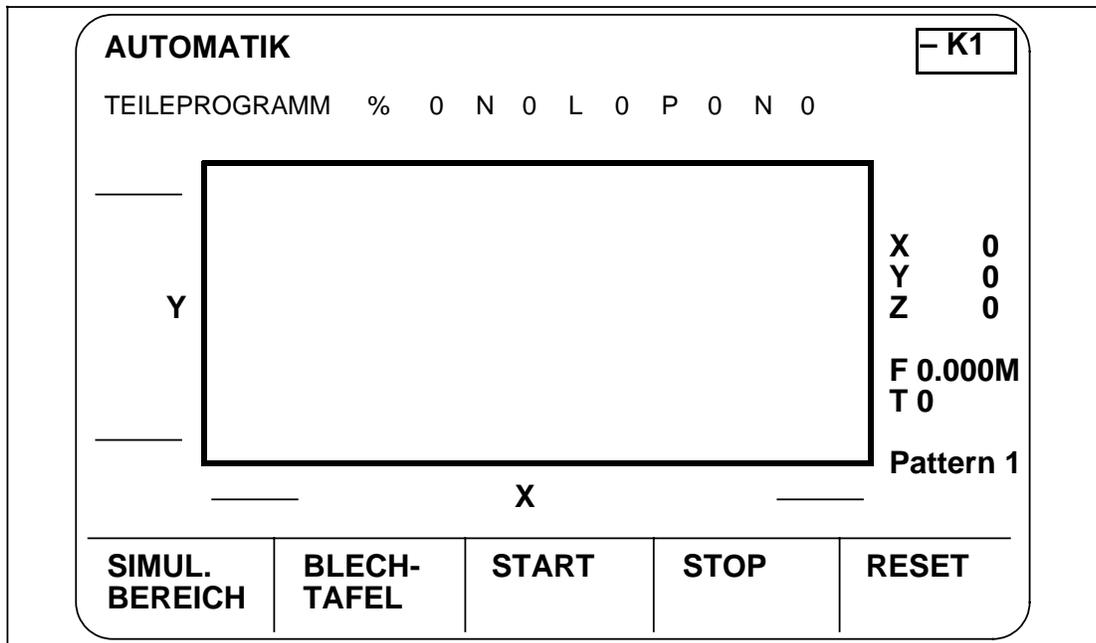
Für eine Einzelteildarstellung in der Simulation kann die Patternbearbeitung über Softkey wahlweise aus- oder eingeschaltet werden (Pattern 0/1).

Ist die Patternbearbeitung ausgeschaltet, so ist auch die Überwachung der Reihenfolge aller Pattern-@-Befehle im Programm nicht aktiv!

AUTOMATIK				- K1
TEILEPROGRAMM % 0				
SIMULATION PROGRAMMBEEINFLUSSUNG				
STATUS				
Einzelatz		NEIN		
Satz ausblenden		NEIN		
Programmierter Halt		NEIN		
Pattern		1		
EINZELS JA-NEIN	AUSBL. JA-NEIN	PR.HALT JA-NEIN	Pattern 0 / 1	SIMULA- TION

Wird 'Pattern 0' angewählt (Pattern aus), werden die @-Befehle @750 ... @752 nicht ausgeführt.

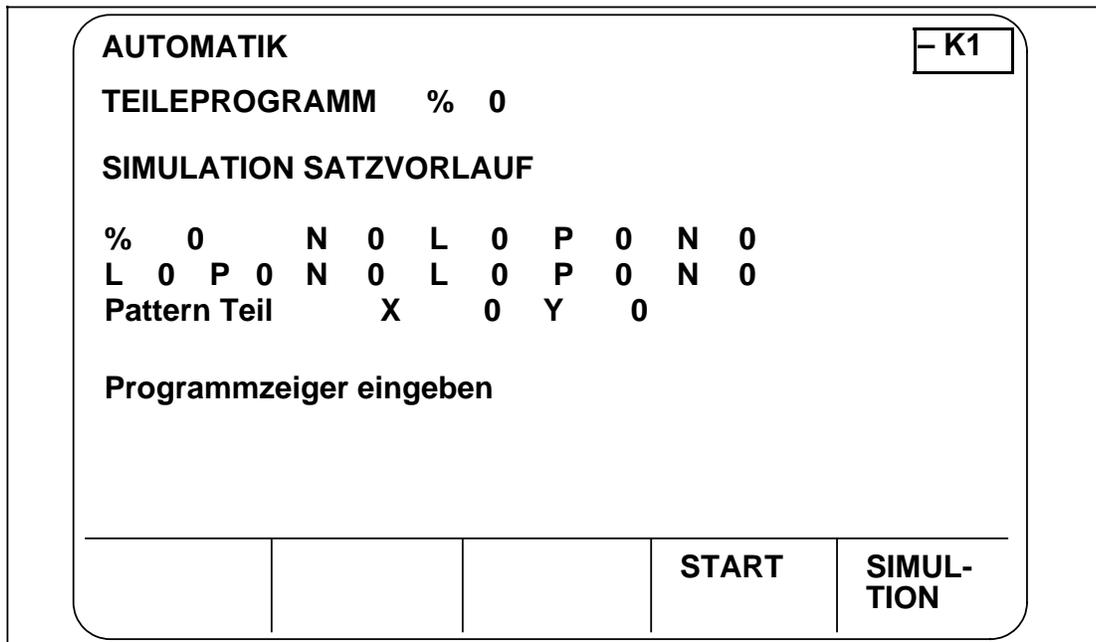
Ob Pattern ein- (1) oder ausgeschaltet (0) ist, wird auch im Simulationsbild angezeigt.



Beim Satzvorlauf in der Simulation kann/muß ebenfalls das gewünschte Patternteil angegeben werden. (Wenn Pattern abgewählt ist, wird das Suchkriterium "Pattern Reihe X/Y" nicht beachtet.)

11.12.8.2 Satzvorlauf bei grafischer Simulation

Beim Satzvorlauf in der Simulation kann ebenfalls das gewünschte Patternteil angegeben werden, siehe Bild:



Der Satzvorlauf in der grafischen Simulation unterscheidet sich nicht vom Satzvorlauf im AUTOMATIK-Betrieb, daher gelten hier ebenfalls die Aussagen und die Beispiele aus Kapitel 11.12.7.1.

11.12.8.3 Nullpunktverschiebungen G54 ... G57 bei Simulation

Über Maschinendatum ist die Berücksichtigung der Nullpunktverschiebungen G54 ... G57 (grob und fein) bei der Simulation anwählbar.

Dadurch ist auch eine Verschiebung der Patternteile vom Rand weg bei der Simulation sichtbar.

Beachte:

Um eine Überschneidung mit der Bearbeitung im Hauptkanal (Kanal 1) zu vermeiden, muß daher festgelegt werden, daß für den Simulationskanal (Kanal 3) z. B. die Nullpunktverschiebung G57 und für den Hauptkanal z. B. die Nullpunktverschiebung G56 benützt wird.

Die kanalspezifische Anwahl der Nullpunktverschiebung kann mit Hilfe eines Unterprogramms automatisch erfolgen.

Hierzu kann der Befehl @371 verwendet werden:

Sonderbits lesen @371 <VAR> <WERT1> <WERT3>

<VAR> Parameter, in welchem das Bit gespeichert wird

<WERT1> Gewünschte Kanalnummer, 0=eigener Kanal

<WERT3> Gewünschte Bit-Nummer
 Bit0=0/1 Satzvorlauf aktiv (Ja/Nein)
 Bit1=0/1 Probelaufvorschub aktiv (Ja/Nein)
 Bit2=0/1 Simulation aktiv (Ja/Nein)

Mit diesem Befehl werden Sonderbits zur Erfassung verschiedener aktiver Signale ausgelesen (siehe auch Kapitel 11.6).

Beispiel:

Unterprogramm zur automatischen Anwahl der Nullpunktverschiebungen

```
L11 L_F (Anwahl G56 in Kanal 1 und 2 bzw. G57 in Kanal 3)
R50=57 L_F
@371 R51 K0 K2 L_F (eigener Kanal Bit2 R51)
@121 R51 K0 K10 L_F (Sprung wenn Simulation aktiv)
@621 R50 L_F (Dekrementieren von R50)
N10 G= R50 L_F
M17 L_F
```

Dieser Zyklus wird dann jeweils am Anfang des Patternprogramms aufgerufen.

12 Anhang

12.1 Exzenterbetrieb

Bei dieser Betriebsart wird die Stanzbewegung durch einen "Exzenter" ausgeführt, der über eine schnellschaltende Kupplung mit einer Schwungscheibe verbunden ist. Der Antrieb der Schwungscheibe erfolgt durch einen Motor mit nahezu konstanter Drehzahl. Es sind zwei unterschiedliche Drehzahlen möglich und daraus resultierend zwei unterschiedliche Hubzahlen.

Der Exzenterbetrieb unterscheidet sich vom hydraulischen Stanzbetrieb dadurch, daß im Nibbelmodus die Stanze eingekuppelt bleibt. Die Maschine gibt also den Hubtakt vor und die Steuerung muß sich nach diesem Takt richten. Nur in dem Zeitraum, während die Stanze aus dem Blech ist, darf dieses Blech bewegt werden.

Im Nibbelbetrieb bedeutet dies eine Begrenzung der maximalen Strecke zwischen zwei Hübten.

Es wird ausgekuppelt

- im Stanzmodus: nach jedem Hub und
- im Nibbelmodus: nach jedem letzten Hub einer Nibbelstrecke, oder wenn kein nachfolgender Satz zum Auskuppelzeitpunkt vorliegt.

Die Exzentersteuerung ist auch mit **drehbarem Werkzeug** möglich.

Einschränkung:

M 22 ist nicht mit H-Wert programmierbar.

12.1.1 Umschalten der Exzentergeschwindigkeit

Die Steuerung erlaubt die Verwendung einer hohen und einer niedrigen Exzentergeschwindigkeit, entsprechend einer hohen und einer niedrigen Drehzahl des Antriebsmotors. Die Wahl der Exzentergeschwindigkeit hat Einfluß auf den maximal möglichen Verfahrweg zwischen zwei Hübten:

- **Niedrige Exzentergeschwindigkeit** bedeutet eine lange Zeit zwischen Aus- und Wiedereintritt der Stanze ins Blech. In dieser langen Zeit ist ein **großer Verfahrweg** möglich.
- **Hohe Exzentergeschwindigkeit** bedeutet eine kurze Zeit zwischen Aus- und Wiedereintritt der Stanze ins Blech, wodurch nur ein **kleiner Verfahrweg** möglich ist.

Die Grundgeschwindigkeit nach Einschalten der Steuerung läßt sich über Maschinendaten einstellen.

12.1.1.1 Umschalten der Exzentergeschwindigkeit mit Anwahl des Nibbelmodus

Die Exzentergeschwindigkeit wird mit der Anwahl des Nibbelmodus festgelegt, wobei M 22 niedrige und M 24 hohe Exzentergeschwindigkeit bedeutet.

Wird durch Programmierung einer dieser M-Funktionen Nibbeln angewählt, wird damit gleichzeitig auch eine Exzentergeschwindigkeit festgelegt. Der Exzenter hält diese Geschwindigkeit bei (auch nach NC-Reset), bis die andere Exzentergeschwindigkeit programmiert wird.

Programmbeispiel:

```

:
:
X0 Y0 G0 LF
X100 E5 M 22 LF      Umschalten auf niedrige Geschwindigkeit und nibbeln mit niedriger Exzenter-
                       geschwindigkeit
X200 E2 M 24 LF      Umschalten auf hohe Geschwindigkeit und nibbeln mit hoher Exzentergeschwindigkeit
M 25 LF              Einzelhub mit hoher Exzentergeschwindigkeit
:
:

```

12.1.1.2 Umschalten der Exzentergeschwindigkeit durch spezielle M-Funktion

Es wird eine gesonderte M-Funktion zur Geschwindigkeitsumschaltung definiert. Diese ist in einem Maschinendatum hinterlegt und wird zum Umschalten auf die niedrige Exzentergeschwindigkeit verwendet. Mit der nächsthöheren M-Funktion wird die hohe Exzentergeschwindigkeit angewählt.

Programmiervorschrift:

Die umschaltende M-Funktion darf nicht in einem Satz zusammen mit einer der nibbelspezifischen M-Funktionen (M 20 bis M 29) programmiert werden. Die Funktionen M 22 und M 24 wirken identisch und haben keinen Einfluß auf die Exzentergeschwindigkeit.

Programmbeispiel:

In diesem Beispiel wird als M-Funktion zum Umschalten auf niedrige Exzentergeschwindigkeit M140 angenommen.

Diese M-Nummer ist projektierbar über Maschinendatum.

```

:
:
X0 Y0 G0 LF
M140 LF              Umschalten auf niedrige Exzentergeschwindigkeit
X100 E5 M 22 LF      Nibbeln mit niedriger Exzentergeschwindigkeit
M141 LF              Umschalten auf hohe Exzentergeschwindigkeit
X200 E2 M 22 LF      Nibbeln mit hoher Exzentergeschwindigkeit
:
:

```

=0 ... 99, projektierbar über Maschinendatum

12.1.2 E-Wertüberwachung

12.1.2.1 Maximalwerte für den E-Wert und den Drehwinkel

Im Nibbelmodus ist die maximale Verfahrestrecke und der maximale Drehwinkel zwischen zwei Hüben beschränkt. Für die maximal zulässigen Werte sind die folgenden **kanalspezifischen** R-Parameter reserviert:

- R247 Maximaler E-Wert bei niedriger Exzentergeschwindigkeit
- R248 Maximaler E-Wert bei hoher Exzentergeschwindigkeit
(jeweils im gültigen Eingabesystem metrisch/inch)
- R249 Maximaler Drehwinkel bei niedriger Exzentergeschwindigkeit
- R250 Maximaler Drehwinkel bei hoher Exzentergeschwindigkeit
(Wert in Winkelgrad)

Die **Übernahme** der R-Parameter als neue Maximalwerte erfolgt durch **Anwahl des Nibbelmodus und Programmierung**

- eines **E-Werts** oder
- einer **Geschwindigkeitsumschaltung**.

Beispiel:

X10 Y10 E2 L _F	Erstmals E-Wert programmiert
X40 M 22 L _F	Nibbeln einschalten
X100 Y100 L _F	
X150 Y150 E5 L _F	Neuer E-Wert programmiert
X200 L _F	
M140 L _F	Umschalten der Exzentergeschwindigkeit, (M-Funktion projektierbar über MD)
X200 Y200 L _F	
X0 M 20 L _F	Nibbeln ausschalten
X150 Y150 M 22 L _F	Nibbeln wieder einschalten
:	
:	

Sätze, in denen der maximale E-Wert aus den R-Parametern gelesen wird.

Um sicherzustellen, daß ein Eintrag eines neuen maximalen E-Wertes in die R-Parameter nicht vorauseilend wirkt, muß vor dem Eintrag eines neuen E-Wertes in die R-Parameter ein Befehl @714 programmiert werden, wenn

- ein E-Wert oder
- eine Exzenter-Geschwindigkeitsumschaltung oder
- Nibbelmodus einschalten

weniger als 10 Sätze vorher programmiert war.

Beachte:

Während des Nibbelmodus darf die Nullpunktverschiebung nicht verändert werden, weil sonst die zu verfahrenende Strecke zu groß werden kann.

12.1.2.2 Überwachungsarten der maximalen E- und Drehwinkel-Werte bei M 22

Durch Maschinendaten lassen sich folgende Reaktionen bei Überschreitung der Grenzwerte projektieren:

1. Alarm und Stop der Bearbeitung

Überschreitet der programmierte E-Wert den Grenzwert in R247 bzw. R248, wird ein Reset-Alarm ausgelöst (Alarm 2235).

Überschreitet der Drehwinkel des Werkzeuges den Grenzwert in R249 bzw. R250, wird ein Reset-Alarm ausgelöst (Alarm 2241).

2. E-Wert reduzieren

Überschreitet der programmierte E-Wert den Grenzwert in R247 bzw. R248, wird der E-Wert auf den Grenzwert reduziert.

Überschreitet der Drehwinkel des Werkzeuges den Grenzwert in R249 bzw. R250, wird ein Reset-Alarm ausgelöst (Alarm 2241).

3. Automatischer Übergang in den Stanzmodus

Überschreitet der programmierte E-Wert den Grenzwert in R247 bzw. R248, erfolgt automatisch eine Umschaltung in den Stanzmodus, bis erneut ein E-Wert, der Nibbelmodus oder eine Änderung der Exzentergeschwindigkeit programmiert wird (siehe auch Kapitel 12.1.3 "Automatische Stanz-Nibbelumschaltung").

Überschreitet der Drehwinkel des Werkzeuges den Grenzwert in R249 bzw. R250, wird automatisch in den Stanzmodus umgeschaltet.

12.1.3 Automatische Stanz-Nibbelumschaltung

Diese Funktion wird durch Maschinendatum aktiviert und gilt ab GA3.

Bei Maschinen mit Exzentersteuerung wird zwischen der Betriebsart M 25 "Stanzen" und M 22 "Nibbeln" unterschieden.

Die Betriebsart Nibbeln ist nur geeignet für kurze Strecken zwischen zwei Hüben, da der Exzenter ständig eingekuppelt bleibt und daher der Verfahrweg zwischen Aus- und Wiedereintritt der Stanze aus dem Blech begrenzt ist.

Ist die automatische Umschaltung aktiviert, schaltet die Steuerung selbstständig bei M 25 zwischen Nibbeln und Stanzen um. Hierbei wird nach jedem Hub geprüft, ob die nächste Stanzposition so nahe liegt, daß sie mit eingekuppeltem Exzenter erreicht werden kann. Ist dies nicht der Fall, wird ausgekuppelt. Als Grenzwert zum Umschalten dienen die maximalen E-Werte in den R-Parametern.

Ohne diese automatische Umschaltung richtet sich die Betriebsart ausschließlich nach der Programmierung.

12.1.3.1 Programmiervorschrift

Durch diese Funktion wird die Programmiervorschrift für die Exzentersteuerung nicht geändert.

Ist Stanzen (M 25) programmiert, wird automatisch zwischen den Betriebsarten "Stanzen" und "Nibbeln" ausgewählt.

Die Bedeutung der M 22-Funktion bleibt bestehen: Es wird in der Betriebsart Nibbeln gearbeitet.

12.1.3.2 Umschaltkriterium Stanzen Nibbeln beim Arbeiten mit M 25

Es wird die tatsächliche Verfahrstrecke in X- und Y-Richtung mit den maximalen E-Werten (niedrige Drehzahl R247, hohe Drehzahl R248) verglichen, genauso der Drehwinkel mit dem maximal zulässigen Drehwinkel (R249 bzw. R250).

Der Exzenter bleibt bei M 25-Programmierung eingekuppelt, wenn

- die nächste Strecke mit Teilstreckenprogrammierung am Ende einer Teilstrecke eine Stanzung hat
- **und** die Strecke in X-Richtung kleiner gleich dem maximalen E-Wert ist
- **und** die Strecke in Y-Richtung kleiner gleich dem maximalen E-Wert ist
- **und** der Drehwinkel kleiner gleich dem maximalen Drehwinkel ist.

Dies bedeutet bei nicht achsparallelen Strecken einen größeren möglichen E-Wert im Nibbelmodus als der Maximalwert aus den R-Parametern.

Der maximale E-Wert wird aktualisiert, wenn

- der Nibbelmodus angewählt ist und
 - ein E-Wert programmiert wird oder
 - die Exzentergeschwindigkeit verändert wird
- oder M 25 programmiert ist,
- oder die Exzentergeschwindigkeit geändert wurde, während M 25 aktiv ist.

Beispiel:

```

:
X10 Y10 M 25 LF
X100 Y20 LF
M140 LF                (M140 projiziert über Maschinendatum)
X150 Y20 LF
X200 M 20 LF
X250 M 25 LF
:

```

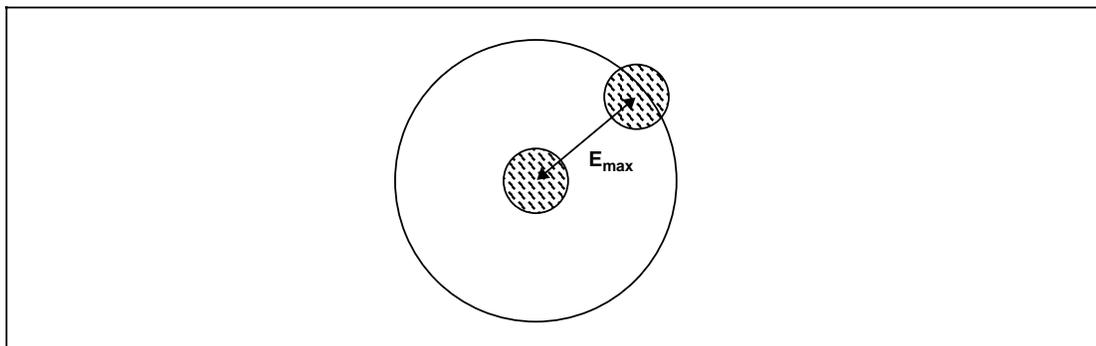
Sätze, in denen die R-Parameter gelesen werden.

Um sicherzustellen, daß der Eintrag eines neuen E-Wertes in die R-Parameter nicht voraus-eilend wirkt, muß vor dem Eintrag der Befehl @714 programmiert werden, wenn ein E-Wert oder eine Exzentergeschwindigkeitsumschaltung oder die M 25-Funktion vorher programmiert ist.

Folgerungen:

M 22- bzw. M 24-Programmierung:

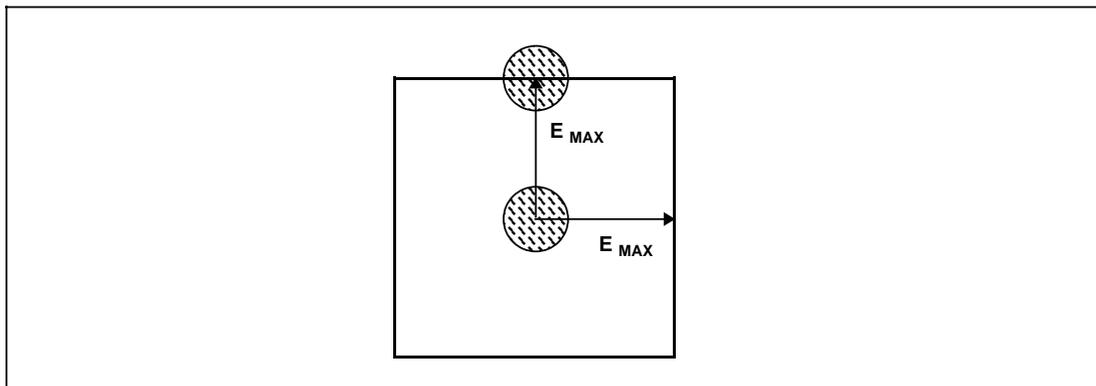
Es wird der Grenzwert in den R-Parametern mit dem programmierten E-Wert (und nicht mit dem intern errechneten, reduzierten E-Wert) verglichen.



Bereich, in dem bei M 22- bzw. M 24-Programmierung der Exzenter eingekuppelt bleibt

M 25-Programmierung:

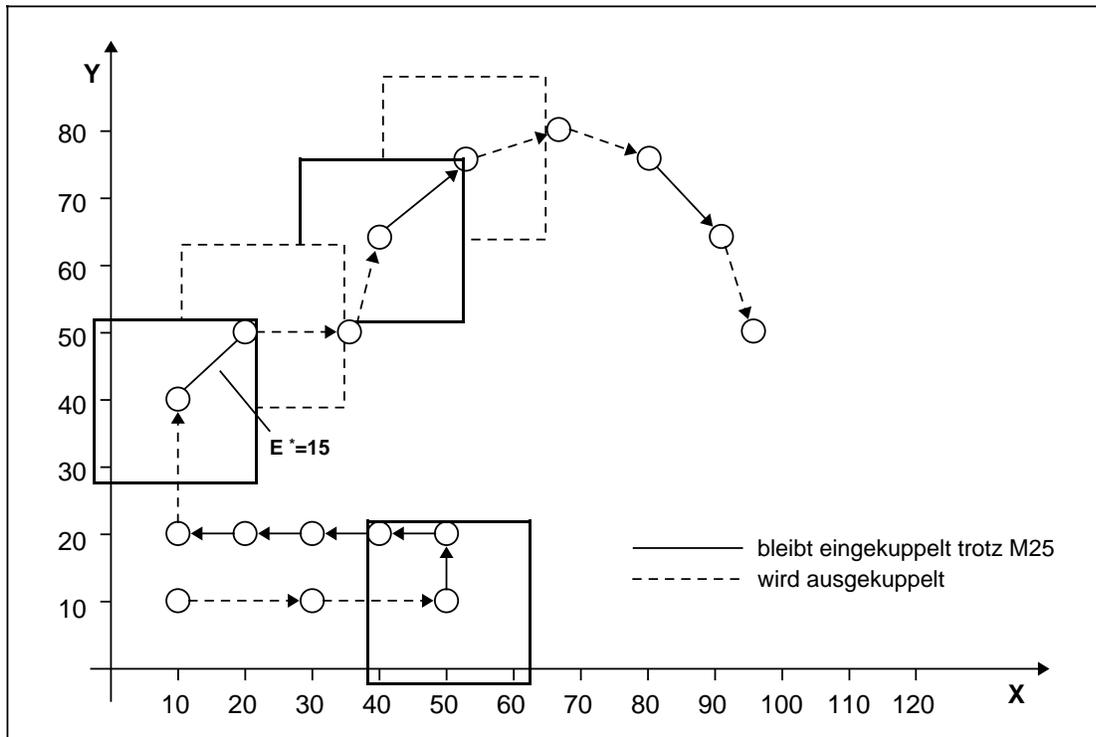
Der Grenzwert in den R-Parametern wird mit der tatsächlichen Verfahrstrecke in den einzelnen Achsrichtungen verglichen.



Bereich, in dem bei M 25-Programmierung der Exzenter eingekuppelt bleibt:

Wie vorher beschrieben, sind die Bereiche verschieden, in denen Nibbeln erlaubt ist. Dies hat folgende Auswirkungen:

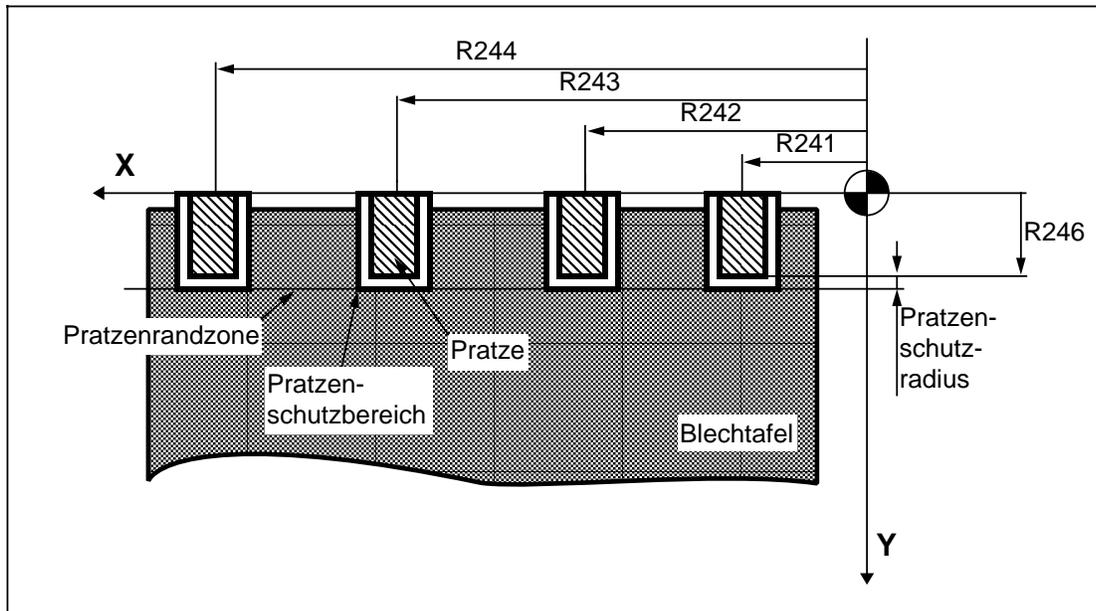
- Der maximale E-Wert ist nicht nur achsparallel zu berücksichtigen, sondern auch diagonal.
- Bei Bearbeitung eines Kreisbogens kann automatisch zwischen Stanzen und Nibbeln umgeschaltet werden:
 - Wenn der Teil des Kreisbogens bearbeitet wird, in dem beide Achsen etwa gleichweit bewegt werden, wird genibbelt;
 - im Bereich des Kreisbogens, in dem sich eine Achse nur minimal bewegt, wird gestanzt.



12.2 Pratzenschutz

Die Option Pratzenschutz ermöglicht einen softwaremäßigen Schutz der Pratzen gegen Kollision mit im Arbeitsbereich befindlichen Einheiten wie z. B. dem Stanzkopf oder einer Laserschneideeinrichtung.

12.2.1 Eingabewerte



Eingabewerte für den Pratzenschutz der SINUMERIK 810N sind die Positionen der Pratzen 1 bis 4 auf der X-Achse, sowie X- und Y-Abmessungen der Pratzen. Die einzelnen Werte sind in R-Parametern abgelegt.

R241 Position der ersten Prätze auf der X-Schiene
 R242 Position der zweiten Prätze auf der X-Schiene
 R243 Position der dritten Prätze auf der X-Schiene
 R244 Position der vierten Prätze auf der X-Schiene
 wobei $R241 < R242 < R243 < R244$ einzugeben ist!

R245 Abmessung der Pratzen in X-Richtung
 R246 Abmessung der Pratzen in Y-Richtung

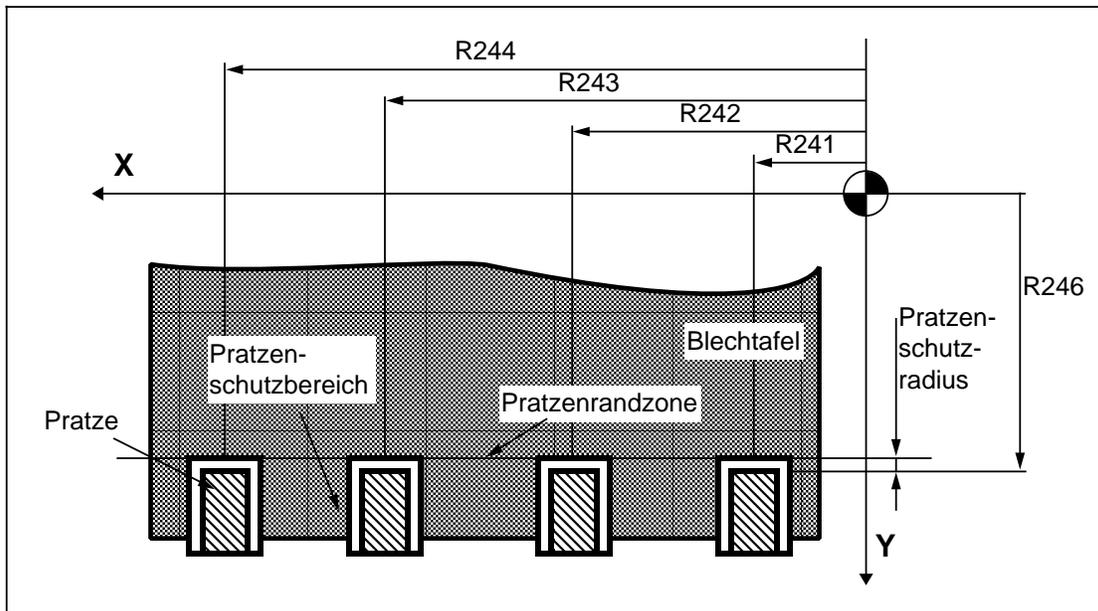
Beachte:

R246 hat eine andere Bedeutung, wenn die Pratzen auf Bedienerseite sind (siehe Kapitel 12.2.1.1).

Alle Angaben erfolgen gemäß eingestelltem Eingabesystem.

Wenn der Maschinentyp nur über 2 oder 3 Pratzen verfügt, sollte R243 bzw. R244 mit einem hohen Wert belegt werden (z. B. 99999). Die Aktualisierung der Eingabewerte erfolgt mit dem ersten NC-Start oder mit M 29 (siehe Kapitel 12.2.2).

12.2.1.1 Pratzen auf Bedienerseite



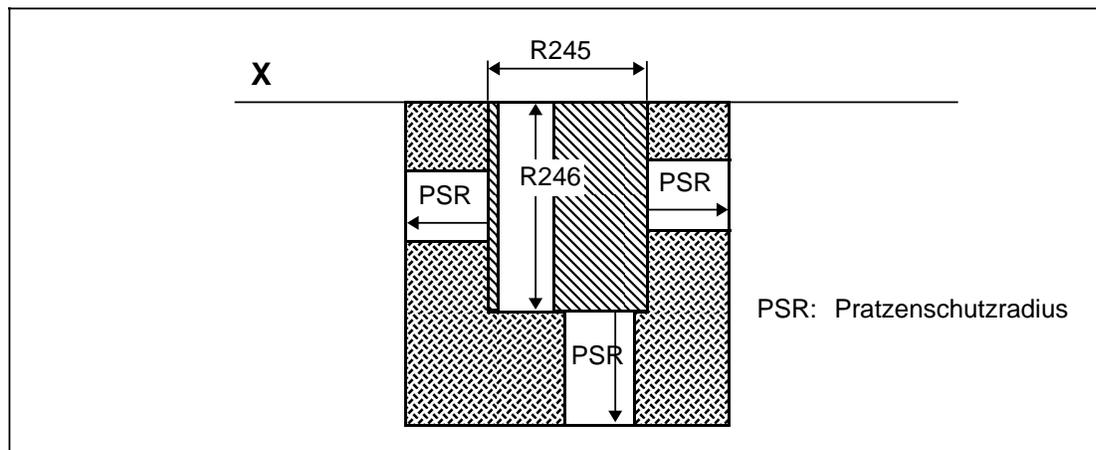
Sollten sich die Pratzen auf Bedienerseite befinden, muß dies per Maschinendatum projiziert werden.

Der R-Parameter R246 hat in diesem Fall eine andere Bedeutung, er enthält den maximalen Arbeitsbereich auf der Blechtafel (siehe Bild).

Ist der werkzeugspezifische Schutzbereich eingeschaltet, erstreckt sich der Pratzenschutzbereich Y von $(R246 - (\text{Pratzenschutzradius}))$ bis zum positiven Softwareendschalter.

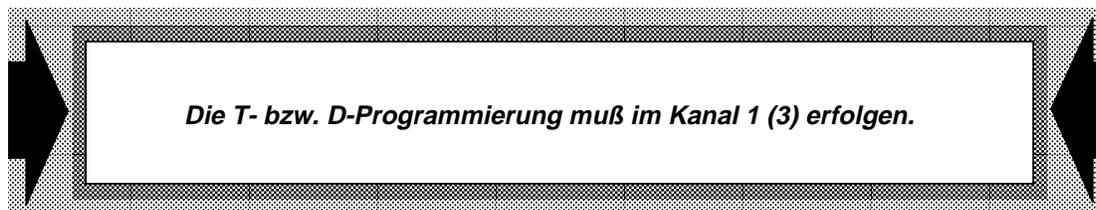
12.2.1.2 Werkzeugspezifische Pratzenschutzbereiche

Der Pratzenschutzbereich ergibt sich aus den Abmessungen der Pratzen und dem Radius der Werkzeugaufnahme, nachfolgend mit Pratzenschutzradius bezeichnet.



Der Werkzeugaufnahmedurchmesser kann sich theoretisch mit jedem Werkzeugwechsel ändern. Für diese Fälle kann man im Werkzeugkorrekturspeicher einen Wert (Pratzenschutzradius P8) eintragen, der beim Anwählen des Werkzeugs durch T-Programmierung (oder ab GA3 auch durch D-Programmierung, siehe nachfolgende Beschreibung) einberechnet wird (Übersicht Werkzeugkorrekturspeicher Kapitel 8.2).

Die D-Nummer dieses Korrekturspeichers entspricht der T-Nummer des Werkzeugs.



Der aktuelle Pratzenschutzbereich für eine Pratze berechnet sich dann wie folgt:

Beispiel für Pratze 3

Pratzenschutzbereich

- in Y-Richtung von 0 bis $(R246 + (\text{Pratzenschutzradius}))$
- in X-Richtung von $(R243 - 1/2 \cdot R245 - (\text{Pratzenschutzradius}))$
bis $(R243 + 1/2 \cdot R245 + (\text{Pratzenschutzradius}))$

(Achtung: geänderter Y-Schutzbereich, wenn die Pratzen auf Bedienerseite sind, siehe Kapitel 12.2.1.1)

Werkzeugspezifischer Pratzenschutz Aktualisierung mit T-Nummer oder ab GA3 mit D-Nummer

Bei SINUMERIK 810N/820N wird

- bis einschließlich GA2:
der Pratzenschutzradius aus dem Werkzeugkorrekturspeicher durch Programmierung einer T-Nummer einberechnet. Dabei entspricht die programmierte T-Nummer der D-Nummer des Werkzeugkorrekturspeichers.
- ab GA3:
der Pratzenschutzradius aus dem Werkzeugkorrekturspeicher wahlweise durch Programmierung einer T-Nummer oder (und) einer D-Nummer einberechnet. Dies ist über Maschinendaten projektierbar.

Festlegung (ab GA3):

Da T-Nummern bis zu 4stellig sein können, wird der durch die unteren beiden Dekaden adressierte Korrekturwert einberechnet.

Beispiel:

N5 T1234



Der Pratzenschutzradius wird aus dem Werkzeugkorrekturspeicher Nr. 34 genommen.

Sind die beiden unteren Dekaden 0, wird als Pratzenschutzradius der Wert 0 genommen. Bei Programmierung von D-Nummern sind nur Werte von D1 bis D99 zulässig.

Es ist auch die Anwahl mit T- **und** D-Nummer erlaubt. Wird dann in einem Satz eine T- **und** eine **D-Nummer** programmiert, gilt der im Satz **zuletzt** geschriebene Wert.

Beispiel:

N5 T1234 X100 Y50 D2 M6



Der Pratzenschutzradius wird aus dem Werkzeugkorrekturspeicher Nr. 2 (D02) genommen.

12.2.2 Prätzenpositionen aktualisieren M 29

Die Eingabewerte werden beim NC-Start aktualisiert. Dabei werden nur die Inhalte von R241 bis R246 berücksichtigt. Wird eine Werkzeugnummer programmiert, werden die Inhalte von R241 bis R246 und der Wert aus dem Werkzeugkorrekturspeicher berücksichtigt. Während des Ablaufs eines NC-Programms können die Werte auf Anforderung erneut aktualisiert werden:

M 29 Inhalte von R241 ... R246 und Pratzenschutzradius aktualisieren.

Der zuletzt programmierte werkzeugspezifische Schutzbereich wird mit einberechnet.

Bevor M 29 programmiert wird, müssen zuerst die entsprechenden R-Parametern neu belegt werden.

Einschränkung: Im Satz mit M 29 dürfen keine Achsen programmiert sein.

Anwendungsmöglichkeit:

- bei verschiebbaren Prätzen

12.2.3 Prätzen umfahren

Diese Funktion ist über Maschinendatum aktivierbar.

Bei aktivem Prätzenumfahren werden mittels der Ausgangs- und Zielposition eines Satzes und den Prätzenpositionen automatisch "Ausweichbewegungen" generiert oder eine Alarmmeldung ausgegeben werden. Eine Auswahl der Reaktion ist über Maschinendaten möglich.

Die Ausweichbewegungen sind interne eingeschobene Sätze und werden vor dem eigentlichen Satz mit den Zielkoordinaten und den programmierten Hilfs- und Zusatzfunktionen ausgeführt.

Die Funktion Prätzenumfahren unterscheidet zwischen 6 Standardroutinen:

1. Start- und Zielpunkt befinden sich **außerhalb** des Prätzenschutzbereiches in Y-Richtung.
keine Ausweichbewegungen

2. **Startpunkt außerhalb** und **Zielpunkt innerhalb** des Prätzenschutzbereiches in Y-Richtung.

Es wird eine Ausweichbewegung generiert.
In der Ausweichbewegung werden der Zielpunkt in X-Richtung und der Rand der Prätzenschutzzone in Y-Richtung angefahren.

3. **Startpunkt innerhalb** und **Zielpunkt außerhalb** des Prätzenschutzbereiches in Y-Richtung.

Es wird eine Ausweichbewegung generiert.
In der Ausweichbewegung wird der Rand der Prätzenschutzzone in Y-Richtung angefahren.

4. **Start- und Zielpunkt** befinden sich **innerhalb** des Prätzenschutzbereiches in Y-Richtung **und zwischen zwei Prätzen**.

Es wird keine Ausweichbewegung generiert.

Ist der Prätzenschutzbereich für Laser-/Plasmabearbeitung aktiviert (siehe Kapitel 12.2.6), **muß vor M 29** ein Satz mit **@714 programmiert** werden. Sonst arbeitet der Prätzenschutz mit falschen Werten.

5. **Start- und Zielpunkt** befinden sich **innerhalb** des Prätzenschutzbereiches **und sind durch Prätze(n) getrennt**.

Es werden zwei Ausweichbewegungen generiert.
In der ersten Ausweichbewegung wird der Rand der Prätzenschutzzone in Y-Richtung angefahren. In der zweiten Ausweichbewegung wird der Endwert in X-Richtung angefahren.

6. Der **Zielpunkt** befindet sich **in einem Prätzenschutzbereich**.

Es wird ein "RESET" Alarm ausgegeben.

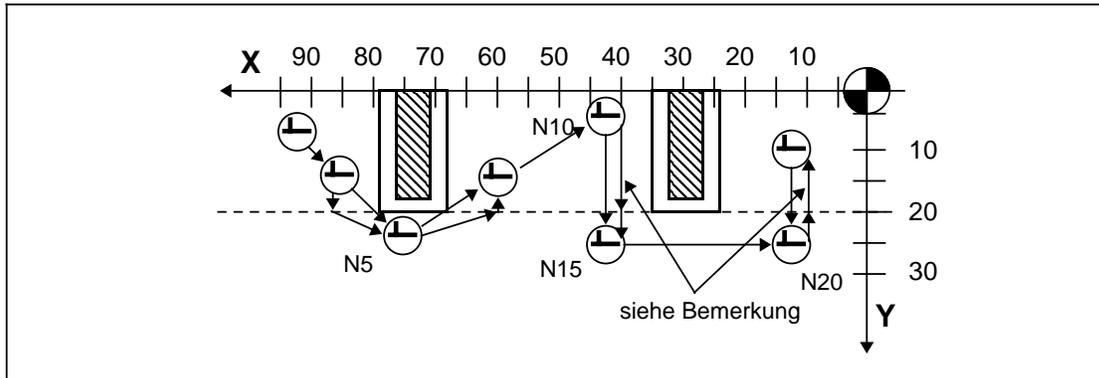
Ausnahme:

Stanzen im Prätzenschutzbereich ist erlaubt, siehe Kapitel 12.2.6.

=0 ... 99, projektierbar über Maschinendatum

12.2.3.1 Beispiele

Beispiel 1:



```

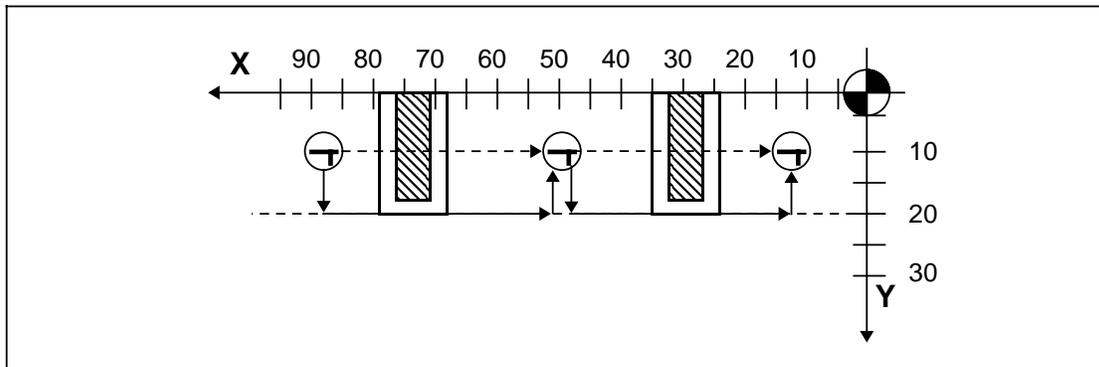
%16 LF
N10 X95 Y5 LF
N20 X75 Y25 H2 M 22 LF
N30 X40 Y5 H2 LF
N40 Y30 E25 LF
N50 X15 LF
N60 Y10 LF
N70 M 20 LF
N80 M2 LF

```

Bemerkung:

Ist die X-Achse nicht programmiert, so wird der Satzendwert ohne Ausgleichbewegung erreicht. Im obigen Beispiel ist dies in den Sätzen N15 und N25 der Fall.

Beispiel 2:

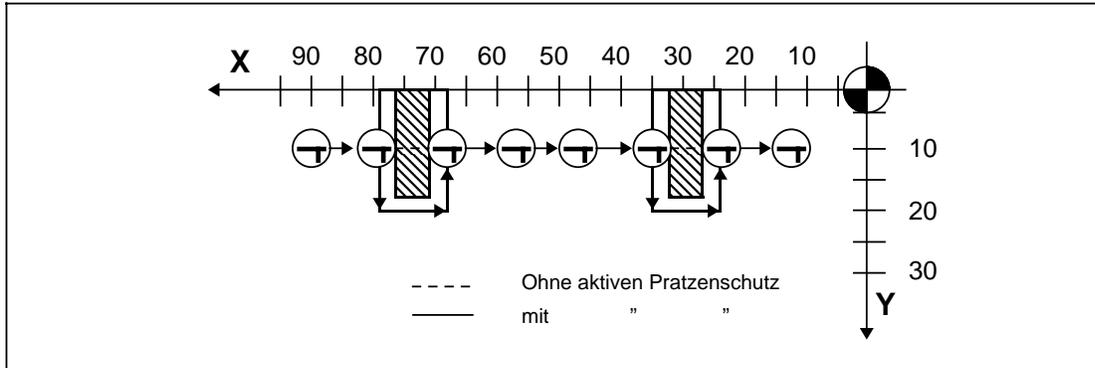


```

%17 LF
N10 X90 Y10 LF
N20 X10 M 22 H2 LF
N30 M 20 LF
N40 M2 LF

```

Beispiel 3:



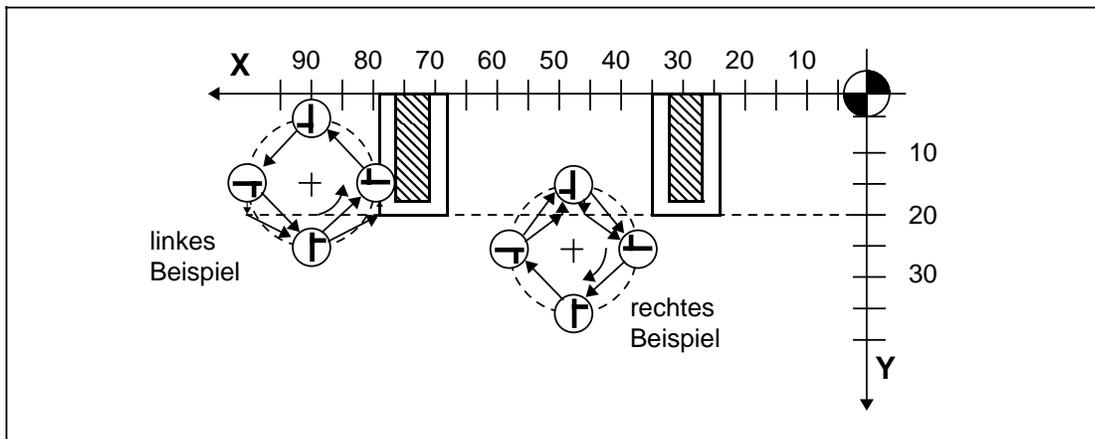
```
%18 LF
N10 X90 Y10 LF
N20 X10 M 22 H7 LF
N30 M 20 LF
N40 M2 LF
```

Alle Ausweichbewegungen werden ohne Nullpunktverschiebungen abgefahren. Mitprogrammierte Hilfsfunktionen werden erst in dem Satz ausgegeben, in dem die Zielkoordinaten erreicht werden.

Ausnahme:

Die Ausgabe mitprogrammierter schneller M-Funktionen erfolgt sofort.

Die Ausweichbewegungen werden nur bei Geradeninterpolation (G00/G01) und bei Kreisinterpolation (G02/G03) in Verbindung mit E- oder H-Programmierung generiert. Bei Kreisinterpolationen ohne Stanzfunktionen, wie z. B. beim Laserschneiden, werden lediglich die programmierten Endwerte überwacht und gegebenenfalls ein Alarm ausgegeben.



12.2.5 Stanzen im Pratzenschutzbereich erlaubt

Bei einigen Maschinen ist es möglich, auch im Pratzenschutzbereich zu stanzen. Dadurch ist die Blechausnutzung besser.

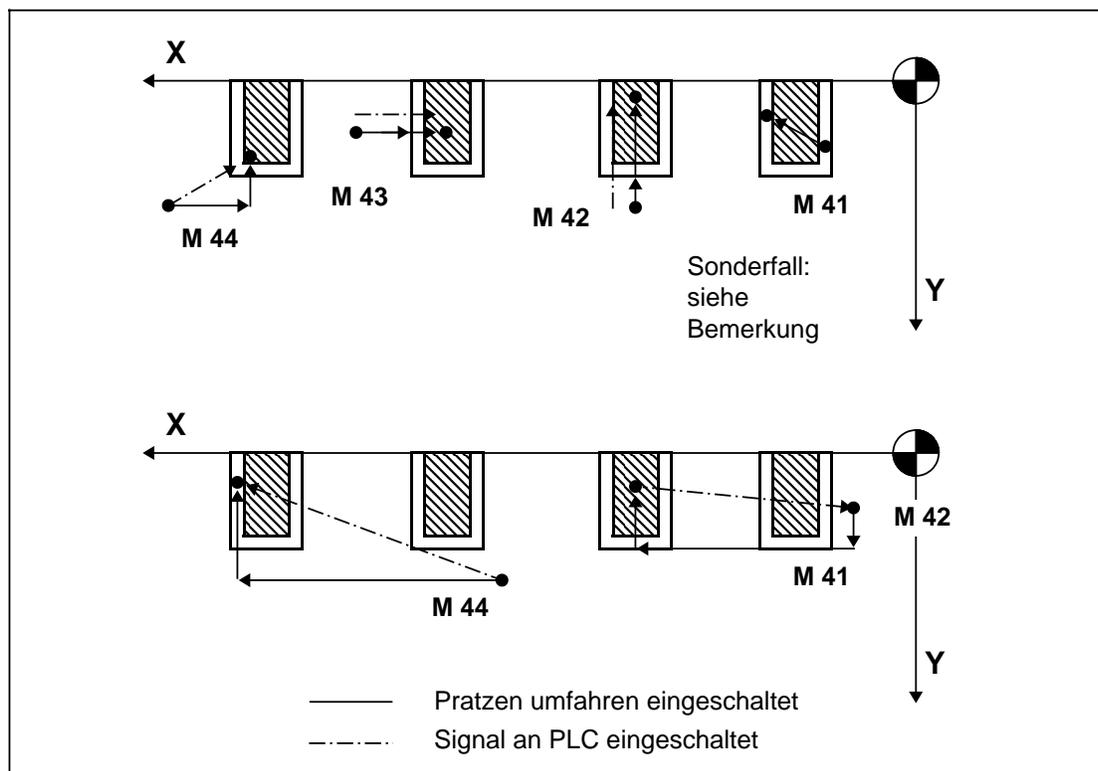
Voraussetzung hierfür ist, daß die Pratzen vor dem Stanzvorgang von der zu stanzenden Position entfernt werden können.

Für diese Maschinen kann über Maschinendatum eine Sonderroutine im Pratzenschutzbereich aktiviert werden. Anstelle der Alarmmeldung (siehe Fall 6 Kapitel 12.2.3) wird mit dem **ersten Ausweichsatz** eine spezielle M-Funktion ausgegeben. Diese M-Funktion gibt die Pratze an, die sich auf der gewünschten Zielposition befindet.

Festlegung:

- M 41 Zielposition ist im Bereich der Pratze 1
- M 42 Zielposition ist im Bereich der Pratze 2
- M 43 Zielposition ist im Bereich der Pratze 3
- M 44 Zielposition ist im Bereich der Pratze 4

Diese M-Funktion muß das Zurückziehen der jeweiligen Pratze bewirken (Auswertung in PLC).



Bemerkung:

Liegt der Startpunkt auf dem Rand des Pratzenschutzbereichs, enthält der eingeschobene Satz nur die M-Funktion.

=0 ... 99, projektierbar über Maschinendatum

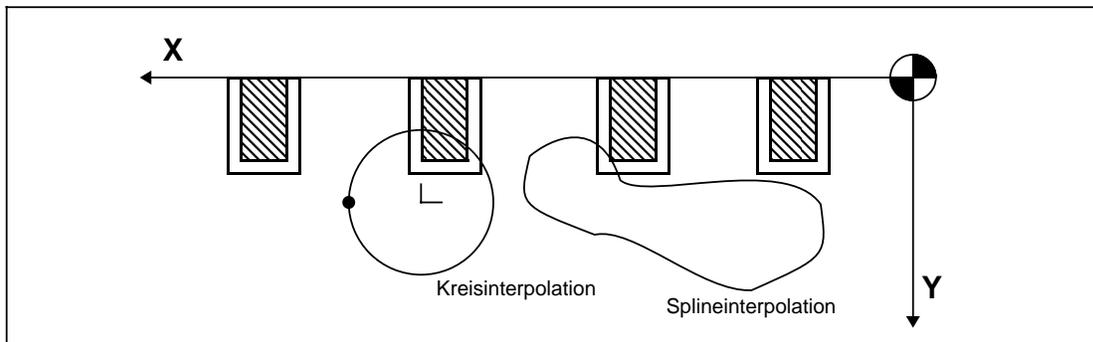
12.2.6 Pratzenschutz für Laserbearbeitung

Diese Funktion ist über Maschinendatum aktivierbar.

Ist diese Pratzenschutzart aktiv und wird bei einer Kreisbewegung (G02/G03 ohne E/H) oder bei einem Splinesatz der Pratzenschutzbereich verletzt, so spricht ein Softwareendschalter an und die Achsen werden angehalten. Die Achsen müssen dann im JOG-Betrieb aus der Gefahrenzone herausgefahren werden.

Diese Überwachung ist auch bei Geradeninterpolation aktiv, kommt jedoch nur zum Einsatz, wenn keine Ausweichbewegungen generiert werden.

Die Überwachung ist nur während AUTOMATIK-Betrieb und nur im Kanal 1 wirksam.



Achtung:

Bei einer Kollision werden die Achsen mit der maximalen Beschleunigung (siehe Maschinendatum) abgebremst. Daher kommt es zu einer Verletzung des Pratzenschutzbereichs. Dieser muß also entsprechend größer dimensioniert werden!

Hinweis zur Programmierung:

Ist dieser Pratzenschutz aktiviert, **muß vor M 29** (Pratzenposition aktualisieren) ein Satz mit **@714** programmiert werden. Sonst arbeitet der Pratzenschutz mit falschen Werten!

12.3 Simulation

12.3.1 Simulation mit Werkzeugdarstellung

Ab Softwarestand 2/21 und bei GA3 ist eine Simulationsmöglichkeit für Nibbelprogramme eingebaut. Diese ist über Maschinendatum aktivierbar. Bei der Programmsimulation am Bildschirm wird für jede Stanzauslösung ein Werkzeug gezeichnet.

Die Werkzeugdaten werden aus dem Werkzeugkorrekturspeicher genommen (Wz.-Abmessung in X und Wz.-Abmessung in Y). Jede Programmierung einer neuen Werkzeugnummer im Teileprogramm bewirkt die Aktualisierung der Werkzeugdaten aus dem Korrekturspeicher.

In der Simulation werden nur die Nibbel- und Stanzsätze mit einer strichpunktierten Linie gezeichnet. Positionierbewegungen (M 20) werden nicht gezeichnet.

Ist der **Pratzenschutz aktiv**, werden **auch die Ausweichbewegungen** gezeichnet.

Ist kein Werkzeug angewählt oder sind keine Werkzeugdaten im Korrekturspeicher vorhanden, werden lediglich die Verfahrbewegungen gezeichnet. Bei Tangentialsteuerung wird das Werkzeug in der jeweiligen Lage gekennzeichnet. Auch DIN-Programme werden simuliert.

12.3.2 Rote Doppellinie zeichnen mit M 26

Für Testzwecke kann mit M 26 eine rote Doppellinie gezeichnet werden. M 26 ist satzweise gültig und hat nur für den Simulationskanal eine sichtbare Auswirkung.

Beispiel:

%21 L_F

N10 G0 X20 Y20 L_F

N20 X100 Y100 M 26 L_F

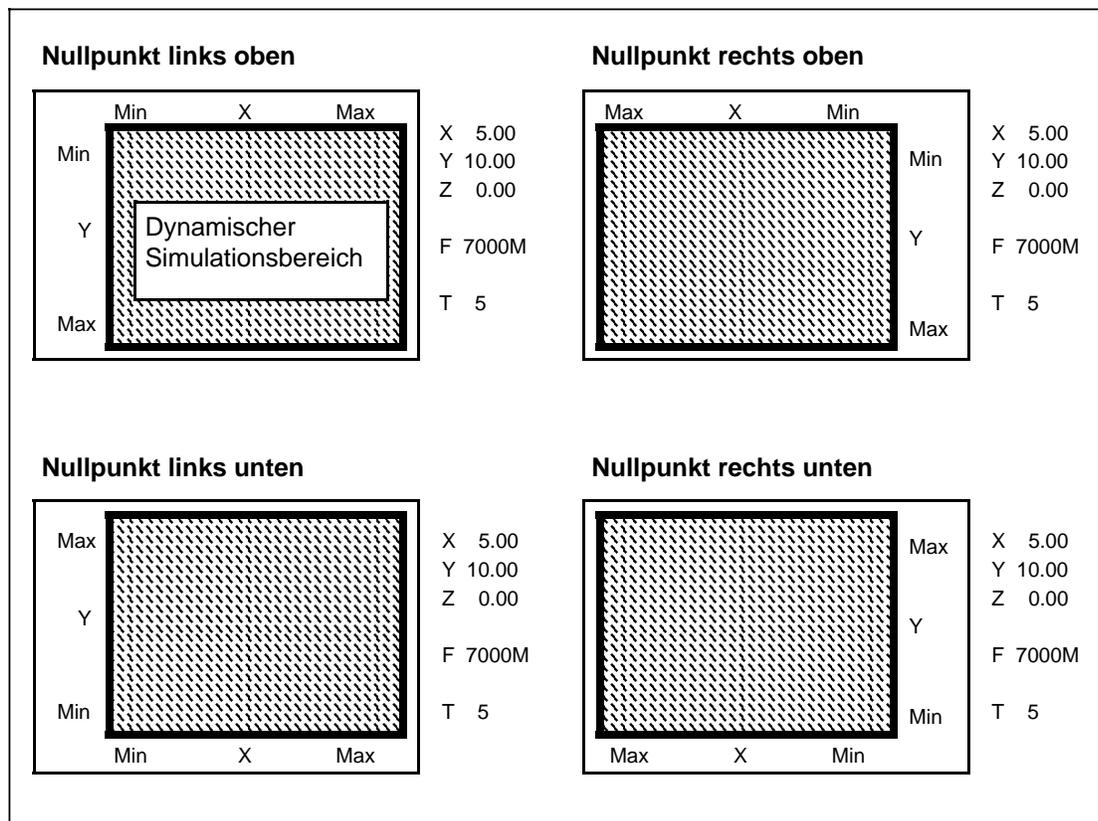
N30 X200 M 26 L_F

N40 M2 L_F

es wird eine rote Doppellinie von 20/20 nach 100/100 gezeichnet
rote Doppellinie von 100/100 nach 200/100

12.3.3 Variabler Nullpunkt bei Simulation

Bei aktivierter Simulation mit Werkzeugdarstellung kann der Nullpunkt per Maschinendatum in jede Ecke des Bildschirms projiziert werden.



13 Programmschlüssel SINUMERIK 810N/820N

13.1 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	10	11	02	03	33			06	12	13			
1	09														
2	17	18	19	16											
3	40	41	42												
4	53														
5	54	55	56	57											
6	04	25	26	58	59	92	74								
7	60	63	64	62											
8	70	71													
9	80	81	82	83	84	85	86	87	88	89					
10	90	91	68												
11	94	95	96	97											
12	147	247	347	148	248	348	48	110	111						
13	50	51													
14															
15															

13.2 Programmschlüssel

Gruppe	EIA	ISO	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	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
G1	g	G	09 #	Geschwindigkeitsabnahme, Genau-Halt fein
G2	g	G	16 * 17 18 19	Ebenenwahl mit freier Achsenwahl (Grundstellung wie G17, Maschinendatum) Ebenenwahl X-Y Ebenenwahl Z-X Ebenenwahl Y-Z
G3	g	G	40 * 41 42	Keine Fräserradiusbahn-Korrektur Fräserradiusbahn-Korrektur links Fräserradiusbahn-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 minimale Arbeitsfeldbegrenzung maximale Arbeitsfeldbegrenzung Programmierbare Nullpunktverschiebung 1 Programmierbare Nullpunktverschiebung 2 Referenzpunktfahren über Teileprogramm Spindeldrehzahl-Sollwertbegrenzung unter Adresse S

Satzweise wirksam, alle übrigen selbsthaltend

* Löschstellung (Grundstellung, nach Reset, M02/M30, nach Einschalten der Steuerung)

1) Es dürfen keine weiteren Funktionen in diesem Satz geschrieben werden.

3) Andere Achsen wählbar (A, B, C, E, U, V, W)

Gruppe	EIA	ISO	Code	Funktion und Bedeutung
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
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	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
G12	g	G	110 111 147 # 1) 247 # 1) 347 # 1) 148 # 1) 248 # 1) 348 # 1) 48 # 1)	Polarkoordinatenprogrammierung, erreichte programmierte Soll- position als neuen Mittelpunkt übernehmen Polarkoordinatenprogrammierung, Mittelpunktprogrammierung mit Winkel und Radius Weiches Anfahren der Kontur mit Gerade Weiches Anfahren der Kontur mit Viertelkreis Weiches Anfahren der Kontur mit Halbkreis Weiches Verlassen mit Gerade Weiches Verlassen mit Viertelkreis Weiches Verlassen mit Halbkreis Gleiches Verlassen wie Anfahren
G13			50 51	Abwahl der Maßstabsänderung Maßstabsänderung
	x	X	±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	Y	±0.001 bis ±99999.999 ±0.0001 bis 3999.9999 ±0.001° bis ±99999.999°	Weginformation in mm Weginformation in inch Weginformation in Grad
	z	Z	±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	Q	±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

Satzweise wirksam, alle übrigen selbsthaltend

* Löschstellung (Grundstellung, nach Reset, M02/M30, nach Einschalten der Steuerung)

1) Es dürfen keine weiteren Funktionen in diesem Satz geschrieben werden.

3) Andere Achsen wählbar (A, B, C, E, U, V, W)

Gruppe	EIA	ISO	Code	Funktion und Bedeutung
	a 3)	A 3)	0 bis 359.9999°	Winkel in Grad bei Konturzug bzw. Winkel in Grad bei Polarkoordinaten
	d	D	1 bis 99 0	Anwahl der Werkzeugkorrektur Abwahl der Werkzeugkorrektur
	e	E	0.001 bis 9999.9 0.0001 bis 399.99 0.000 bis 10.000 0 bis 10000	Vorschub pro Hub in mm Vorschub in inch/min Spannungswert für Laserleistungssteuerung in V Spannungswert für Laserleistungssteuerung in mV
	f	F	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	Vorschub in mm/min, auch als Eckpunkt für Laserleistungssteuerung Vorschub in inch/min, auch als Eckpunkt für Laserleistungssteuerung Vorschub in mm/u Vorschub in inch/u Gewinde. Steigungszu- bzw. -abnahme in mm/u Gewinde. Steigungszu- bzw. -abnahme in inch/u
	h	H	1 bis 9999	Hilfsfunktionen
	i	I	± 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	J	± 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	K	± 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
	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 12000 1 bis 12000 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, Abstand von der Nullmarke des Gebers Verweilzeit in U/min
	t	T	1 bis 9999	Werkzeugnummer
	u 3)	U 3)	± 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

Satzweise wirksam, alle übrigen selbsthaltend

* Löschstellung (Grundstellung, nach Reset, M02/M30, nach Einschalten der Steuerung)

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	
		4)	M1= bis M99 00 . 06 08 09 10 . 16 18 19	Erweiterte Adresse mit Angabe der Nummer Schnelles Ausgangsbyte Bit 0 setzen . . Bit 6 setzen Bits 0 bis 3 setzen Bits 4 bis 6 setzen Bit 0 zurücksetzen . . Bit 6 zurücksetzen Bits 0 bis 3 zurücksetzen Bits 4 bis 6 zurücksetzen	
		4)	20 21 22 24 25	Positionieren Tangentialeistungsteuerung AUS Nibbeln (bei Exzenter mit niedriger Geschwindigkeit) Nibbeln (bei Exzenter mit hoher Geschwindigkeit) Stanzen	
		4)	26	Rote Doppellinie zeichnen (Simulation)	
		4)	28	Analoge Tangentialwinkelausgabe	
		4)	29	Pratzenschutzbereiche aktualisieren	
		4)	31 . 36	Vorbereitende M-Funktion für Laserleistungssteuerung . . Vorbereitende M-Funktion für Laserleistungssteuerung	
		4)	41 42 43 44 46	Zielposition ist im Bereich der Prätze 1 – Signale an PLC Zielposition ist im Bereich der Prätze 2 – Signale an PLC Zielposition ist im Bereich der Prätze 3 – Signale an PLC Zielposition ist im Bereich der Prätze 4 – Signale an PLC Signal an PLC für Stanze hochfahren	

Der Stern steht wahlweise für die Ziffern 1 bis 99.

- Lösstellung (Grundstellung, nach Reset, M02/M30, nach Einschalten der Steuerung)
- # Satzweise wirksam, alle übrigen selbsthaltend

- 1) Es dürfen keine weiteren Funktionen in diesem Satz geschrieben werden.
- 2) Gelochte Spuren
- 3) Andere Adressen wählbar (A, B, C, Q, U, V, W)
- 4) Erweiterte M-Funktionen

An
Siemens AG

AUT V250
Postfach 4848
W-8500 Nürnberg 1

Vorschläge

Korrekturen

für Druckschrift:

SINUMERIK 810N, GA2 und GA3
SINUMERIK 820N, GA2 und GA3
Bedienen und Programmieren
Anwender-Dokumentation

Absender

Name _____

Anschrift Ihrer Firma/Dienststelle

Straße: _____

PLZ: _____ Ort: _____

Telefon: _____ / _____

Benutzeranleitung

Bestell-Nr.: 6ZB5 410-0HM01-0BA1

Ausgabe: 09.92

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 AG 1991 All Rights Reserved
Änderungen vorbehalten

Siemens Aktiengesellschaft

Bestell-Nr. 6ZB5 410-0HM01-0BA1
Printed in the Fed. Rep. of Germany

