

Anwendungsbeschreibung

G-Code Runtime App

G-Code Interpreter für ctrlX OS 02VRS

Schutzvermerk

© Bosch Rexroth AG 2024

Alle Rechte vorbehalten, auch bezüglich jeder Verfügung, Verwertung, Reproduktion, Bearbeitung, Weitergabe sowie für den Fall von Schutzrechtsanmeldungen.

Haftungsausschluss

Die angegebenen Daten dienen allein der Produktbeschreibung. Aufgrund stetiger Weiterentwicklung unserer Produkte kann eine Aussage über eine bestimmte Beschaffenheit oder eine Eignung für einen bestimmten Einsatzzweck aus unseren Angaben nicht abgeleitet werden. Die Angaben entbinden den Verwender nicht von eigenen Beurteilungen und Prüfungen. Es ist zu beachten, dass unsere Produkte einem natürlichen Verschleiß- und Alterungsprozess unterliegen.

DOK-XCORE*-GCO***V02**-AP03-DE-P

DC-AE/EPI5 (TaDo)

Inhaltsverzeichnis

1	Über diese Dokumentation	5
2	Wichtige Gebrauchshinweise	5
2.1	Bestimmungsgemäßer Gebrauch	5
2.1.1	Einführung	5
2.1.2	Einsatz- und Anwendungsbereiche	6
2.2	Nicht bestimmungsgemäßer Gebrauch	7
3	Sicherheitshinweise	7
4	Einführung und Übersicht	7
5	Voraussetzungen	7
6	Grundlagen der NC-Programmierung	8
6.1	Einführung und Übersicht	8
6.2	Ausführung des NC-Programms	8
6.2.1	Speichern des NC-Programms	9
6.2.2	Starten des NC-Programms	9
6.2.3	Status der Ausführung des NC-Programms	9
6.3	Grundlegende Komponenten des NC-Programms	10
6.3.1	Programmsatz	10
6.3.2	Befehle	11
6.3.3	Zusatzbedingungen	11
6.4	Programmwörter	12
6.4.1	Einführung und Übersicht	12
6.4.2	Programmwörter der NC-Funktionen	12
6.4.3	Programmwörter als Parameter (Adresswörter)	13
6.4.4	Verwendung von Trennzeichen zwischen zwei Teilwörtern	14
6.5	Wirkung der Programmwörter	14
6.5.1	Modal	14
6.5.2	Nicht-modal	15
6.6	Spezielle Elemente für die Programmierung	15
6.6.1	Leerzeilen im Programmcode	15
6.6.2	Kommentare in einem Teileprogramm	15
6.7	Unterprogramme	16
6.7.1	Aufruf des Unterprogramms	16
6.7.2	Suchpfad des Unterprogramms	17
7	NC-Funktionen mit Syntax nach DIN 66025 (inkl. Erweiterungen)	17
7.1	Einführung und Übersicht	17
7.2	G-Codes	17
7.2.1	Geradeninterpolation im Eilgang "G0"	17
7.2.2	Geradeninterpolation im Vorschub "G1"	18
7.2.3	Kreisinterpolation im Rechts-/Linkslauf "G2"/"G3"	18
7.2.4	Bahnslope "G8", "G9"	20
7.2.5	Ebenenumschaltung "G16"/"G17-19"	20
7.2.6	Werkzeuglängenkorrektur "G47", "G48"	21
7.2.7	Nullpunktverschiebungstabelle auswählen "G53", "G54-G59"	21
7.2.8	Absolutmaß-Programmierung "G90", Relativmaß-Programmierung "G91"	22

7.2.9	Vorschubprogrammierung "G94"	22
7.2.10	Produktkoordinatensystem "G152", "G153"	23
8	NC-Funktionen mit Hochsprachen-Syntax	24
8.1	Einführung und Übersicht	24
8.2	Globale Signale	24
8.2.1	"SetSignal"	24
8.2.2	"ResetSignal"	24
8.2.3	"WaitForSignal"	25
8.3	M-Code	25
8.4	"Contour"	26
8.5	"WAIT"	27
8.6	"PolyTrans"	27
8.7	"PathDynLim"	28
8.8	"AxsDynLim"	29
9	Erweiterte Programmierung mit High-Level-Syntax	30
9.1	Einführung und Übersicht	30
9.2	Typ und Werte	31
9.3	Variablen (global)	31
9.4	Wertzuweisung	31
9.5	Operatoren	32
9.5.1	Mathematische Operatoren	32
9.5.2	Relative Operatoren	32
9.5.3	Logische Operatoren	32
9.5.4	Priorität	33
9.5.5	Zugriff auf Data Layer-Knoten	34
10	Anhang	35
10.1	Übersicht der NC-Funktionen	35
10.1.1	NC-Funktionen mit Syntax nach DIN 66025	35
10.1.2	NC-Funktionen mit Hochsprachen-Syntax	37
11	Weiterführende Dokumentationen	38
11.1	Übersicht	38
11.2	ctrlX AUTOMATION	38
11.3	ctrlX WORKS	39
11.4	ctrlX OS	39
11.5	ctrlX OS Apps	40
12	Service und Support	44
13	Index	45

1 Über diese Dokumentation

Ausgaben dieser Dokumentation

Ausgabe	Stand	Bemerkung
01	2024-01	Erstausgabe Version GCO-V-0202
02	2024-02	Ausgabe zur Version GCO-V-0204 Neu: <ul style="list-style-type: none">➔ Kapitel 7.2.3 Kreisinterpolation im Rechts-/Linkslauf "G2"/"G3" auf Seite 18➔ Kapitel 7.2.5 Ebenenumschaltung "G16"/"G17-19" auf Seite 20➔ Kapitel 8.3 M-Code auf Seite 25
03	2024-06	Ausgabe zur Version GCO-V-0206 Neu: <ul style="list-style-type: none">➔ Kapitel 6.7 Unterprogramme auf Seite 16➔ Erweiterte Programmierung mit High-Level- Syntax

2 Wichtige Gebrauchshinweise

2.1 Bestimmungsgemäßer Gebrauch

2.1.1 Einführung

Produkte von Rexroth werden nach dem jeweiligen Stand der Technik entwickelt und gefertigt.
Vor ihrer Auslieferung werden die Produkte auf ihren betriebssicheren Zustand hin überprüft.

⚠️ **WARNUNG**

Personen- und Sachschäden durch falschen Gebrauch der Produkte!
Die Produkte dürfen nur bestimmungsgemäß eingesetzt werden.
Wenn die Produkte nicht bestimmungsgemäß eingesetzt werden, dann können Situationen entstehen, die Sach- und Personenbeschädigung nach sich ziehen.

Wichtige Gebrauchshinweise

ACHTUNG
Schäden bei nicht bestimmungsgemäßigem Gebrauch

Für Schäden bei nicht bestimmungsgemäßigem Gebrauch der Produkte leistet Rexroth als Hersteller keinerlei Gewährleistung, Haftung oder Schadensersatz. Die Risiken bei nicht bestimmungsgemäßigem Gebrauch der Produkte liegen allein beim Anwender.

Bevor Sie die Produkte der Firma Rexroth einsetzen, müssen die folgenden Voraussetzungen erfüllt sein, um einen bestimmungsgemäßen Gebrauch der Produkte zu gewährleisten:

- Jeder, der in irgendeiner Weise mit Rexroth Produkten umgeht, muss die entsprechenden Sicherheitsvorschriften und den bestimmungsgemäßen Gebrauch lesen und verstehen
- Sofern es sich bei den Produkten um Hardware handelt, müssen die Produkte in ihrem Originalzustand belassen werden; d. h. es dürfen keine baulichen Veränderungen an den Produkten vorgenommen werden. Softwareprodukte dürfen nicht dekompiert werden und ihre Quellcodes dürfen nicht verändert werden
- Beschädigte oder fehlerhafte Produkte dürfen nicht eingebaut oder in Betrieb genommen werden
- Es muss gewährleistet sein, dass die Produkte entsprechend den in der Dokumentation genannten Vorschriften installiert sind

2.1.2 Einsatz- und Anwendungsbereiche

Produkte der ctrlX Baureihe sind für Motion-/Logic-Anwendungen geeignet.

ACHTUNG

Produkte der ctrlX Baureihe dürfen nur mit den in dieser Dokumentation angegebenen Zubehör- und Anbauteilen benutzt werden. Nicht ausdrücklich genannte Komponenten dürfen weder angebaut noch angeschlossen werden. Gleiches gilt für Kabel und Leitungen.

Der Betrieb darf nur in den ausdrücklich angegebenen Konfigurationen und Kombinationen der Hardware-Komponenten und mit der in den jeweiligen Dokumentationen und den Funktionsbeschreibungen angegebenen und spezifizierten Soft- und Firmware erfolgen.

Produkte der ctrlX Baureihe sind für den Einsatz in ein- und mehrachsigen Antriebs- und Steuerungsaufgaben geeignet. Für den applikationsspezifischen Einsatz des Systems stehen Gerätetypen mit unterschiedlicher Ausstattung und unterschiedlichen Schnittstellen zur Verfügung.

Typische Anwendungsbereiche:

- Gebäudeautomatisierung
- IoT und Security Gateway bzw. Device
- Handling & Robotic

Steuerungen der ctrlX CORE Baureihe dürfen nur unter den in den weiterführenden Dokumentationen angegebenen Montage- und Installationsbedingungen, in der angegebenen Gebrauchslage und unter den angegebenen Umweltbedingungen (Temperatur, Schutzart, Feuchte, EMV u. a.) betrieben werden.

2.2 Nicht bestimmungsgemäßer Gebrauch

Die Verwendung von ctrlX-Produkten außerhalb der vorgenannten Anwendungsgebiete oder unter anderen als den in der Dokumentation beschriebenen Betriebsbedingungen und angegebenen technischen Daten gilt als "nicht bestimmungsgemäß".

ctrlX-Produkte dürfen nicht eingesetzt werden, wenn sie den folgenden Bedingungen ausgesetzt sind:

- Betriebsbedingungen, die die vorgeschriebenen Umgebungsbedingungen nicht erfüllen. Untersagt sind z. B. der Betrieb unter Wasser, unter extremen Temperaturschwankungen oder extremen Maximaltemperaturen
- Bei Anwendungen, die von Rexroth nicht ausdrücklich freigegeben sind




3 Sicherheitshinweise

Die Sicherheitshinweise, soweit in der vorliegenden Anwendungsdokumentation vorhanden, beinhalten bestimmte Signalwörter ("Gefahr", "Warnung", "Vorsicht", "Hinweis") und ggf. eine Signalgrafik (nach ANSI Z535.6-2006).

Das Signalwort soll die Aufmerksamkeit auf den Sicherheitshinweis lenken und bezeichnet die Schwere der Gefährdung.

Die Signalgrafik (Warndreieck mit Ausrufezeichen), welche den Signalwörtern "Gefahr", "Warnung" und "Vorsicht" vorangestellt wird, weist auf Gefährdungen für Personen hin.

Die Sicherheitshinweise in dieser Dokumentation werden wie folgt dargestellt:

 GEFAHR	Bei Nichtbeachtung dieses Sicherheitshinweises werden Tod oder schwere Körperverletzung eintreten.
 WARNUNG	Bei Nichtbeachtung dieses Sicherheitshinweises können Tod oder schwere Körperverletzung eintreten.
 VORSICHT	Bei Nichtbeachtung dieses Sicherheitshinweises können mittelschwere oder leichte Körperverletzung eintreten.
ACHTUNG	Bei Nichtbeachtung dieses Sicherheitshinweises können Sachschäden eintreten.

4 Einführung und Übersicht

Die G-Code Runtime App dient zum Übersetzen von G-Code (DIN 66025-1/ISO 6983-1) in Motion-Befehle (Motion App).

5 Voraussetzungen

Lizenz

Folgende Lizenzen sind erforderlich, um die G-Code Runtime App zu nutzen:

Lizenz	Typschlüssel	Materialnummer
G-Code Runtime App	SWL-XC*-GCO-GCORUN-TIME*-BANN	R911420238
Motion App (für 4 Achsen)	SWL-XC*-MOT-STD-MOTION****-NNNN	R911400503
Motion "In App"-Option: Cartesian Kinematics	SWL-XC*-MOT-CARTESIAN****-NNNN	R911400509

Installation

ctrlX OS Seitennavigation „Einstellungen → Apps“

Weiterführende Informationen zur Installation finden Sie in den folgenden Links:

ctrlX OS Runtime, Anwendungsbeschreibung: [↗ Lizenzen – Übersicht](#)

ctrlX OS Runtime, Anwendungsbeschreibung: [↗ Apps Grundlagen](#)

ctrlX OS Runtime, Anwendungsbeschreibung: [↗ Lizenzrechtliche Hinweise](#)

[↗ Motion App, Motion Runtime Environment für ctrlX OS, Anwendungsbeschreibung](#)

6 Grundlagen der NC-Programmierung

6.1 Einführung und Übersicht

Eine NC-Steuerung erhält alle Informationen, die für die Bearbeitung eines Werkstücks auf einer Werkzeugmaschine erforderlich sind, über ein NC-Programm (Teileprogramm).

Der Aufbau eines solchen NC-Programms ist variabel, so dass nahezu beliebige Werkstücke mit einer Vielzahl von Technologien bearbeitet werden können. Mit ctrlX Motion CNC werden aktuell die Technologien Additive Fertigung, Dispensen und Biegen unterstützt. Das Teileprogramm enthält nicht nur Informationen über Bewegungen, die den Weg des Werkzeugs relativ zum Werkstück beschreiben, sondern auch Informationen über Technologien.

Die Motion-Informationen werden in einzelne elementare Konturelemente (Geraden, Kreise und Spline Cornering) unterteilt.

Die Steuerung kann dann die Bewegungen für jedes dieser geometrisch einfachen Konturelemente in einem Bearbeitungsschritt ausführen, sofern alle Bearbeitungsschritte in der richtigen Reihenfolge und mit allen erforderlichen Randbedingungen im NC-Programm angegeben sind. Die erforderlichen Randbedingungen bestehen unter anderem aus Technologiefunktionen (Geschwindigkeit, Drehzahl, usw.) und Maschinenhilfsfunktionen (z. B. für Kühlmittel und Achsklemmung).



- Grundlegende Richtlinien zum Aufbau eines NC-Programms sind in DIN 66025 beschrieben.
- Der Inhalt von DIN 66025 "Programmstruktur für numerisch gesteuerte Werkzeugmaschinen" (Teil 1 und 2) entspricht der internationalen Norm: ISO/DIS 6983 und ISO/DP 6983 "Numerische Steuerung von Maschinen".
- Der in der G-Code Runtime App (app.gcode) enthaltene G-Code-Interpreter übersetzt das NC-Programm in Motion-Befehle. Die Bewegung wird von der [↗ Motion App](#) (app.motion) gesteuert.

6.2 Ausführung des NC-Programms

Voraussetzung für die Ausführung eines Skripts, ist die erfolgreiche Installation der G-Code Runtime App.

Die detaillierten Informationen des G-Code Interpreters finden Sie unter folgendem Data Layer-Pfad:

script/state/languages/gcode

Beispiel

```
{
  "language": "gcode",
  "version": "4.2.0",
  "executable": "/var/snap/rexroth-automationcore/956/script/runner/rexroth-gcode/
script.gcode",

  "fileEnding": [
    "npg"
  ],
  "license": "SWL-XCx-GCO-GCORUNTIMEex-BANN",
  "licenseVersion": "1.0"
}
```

Nachdem die G-Code App installiert wurde, stellen Sie bitte sicher, dass die Lizenz für die G-Code App aktiviert ist, bevor Sie eine G-Code-Instanz erstellen.

6.2.1 Speichern des NC-Programms

ctrlX OS verwaltet NC-Programme im Dateisystem des Geräts.

Nähere Informationen zum Dateisystem und den Zugriffsrechten, sowie Informationen über das Erstellen von Teilprogrammen finden Sie in der [ctrlX OS Runtime, Anwendungsbeschreibung](#).

Alle Skripte sind Teil der Konfiguration und sollten in "Solution" gespeichert werden, um in ctrlX OS weiterverarbeitet zu werden.

Der Standardpfad des NC-Skripts lautet:

\$SNAP_COMMON/solutions/activeConfiguration/scripts/gcode

6.2.2 Starten des NC-Programms

Beispiel für das Starten einer G-Code-Skriptinstanz über den Befehl im Data Layer

1. Erstellen Sie im Pfad *script/instances* eine Skriptinstanz mit Payload.

Der Name der G-Code-Instanz muss mit dem Namen der anzuhängenden Kinematik übereinstimmen.

```
{
  "name": "Kinematics_01"
  "language": "gcode"
}
```

2. Befehle zum Ausführen des Skripts im Data Layer:

- *script/instances/[name]/cmd/abort*
- *script/instances/[name]/cmd/file*
- *script/instances/[name]/cmd/reset*
- *script/instances/[name]/cmd/string*

Beispiel

```
{
  "name": "activeConfiguration/script/gcode/main_script.npg"
  "param": []
}
```

6.2.3 Status der Ausführung des NC-Programms

Status und Diagnoseinformationen können aus folgenden Data Layer-Pfaden ausgelesen werden:

- *script/instances//[name]/diag*
- *script/instances//[name]/state*

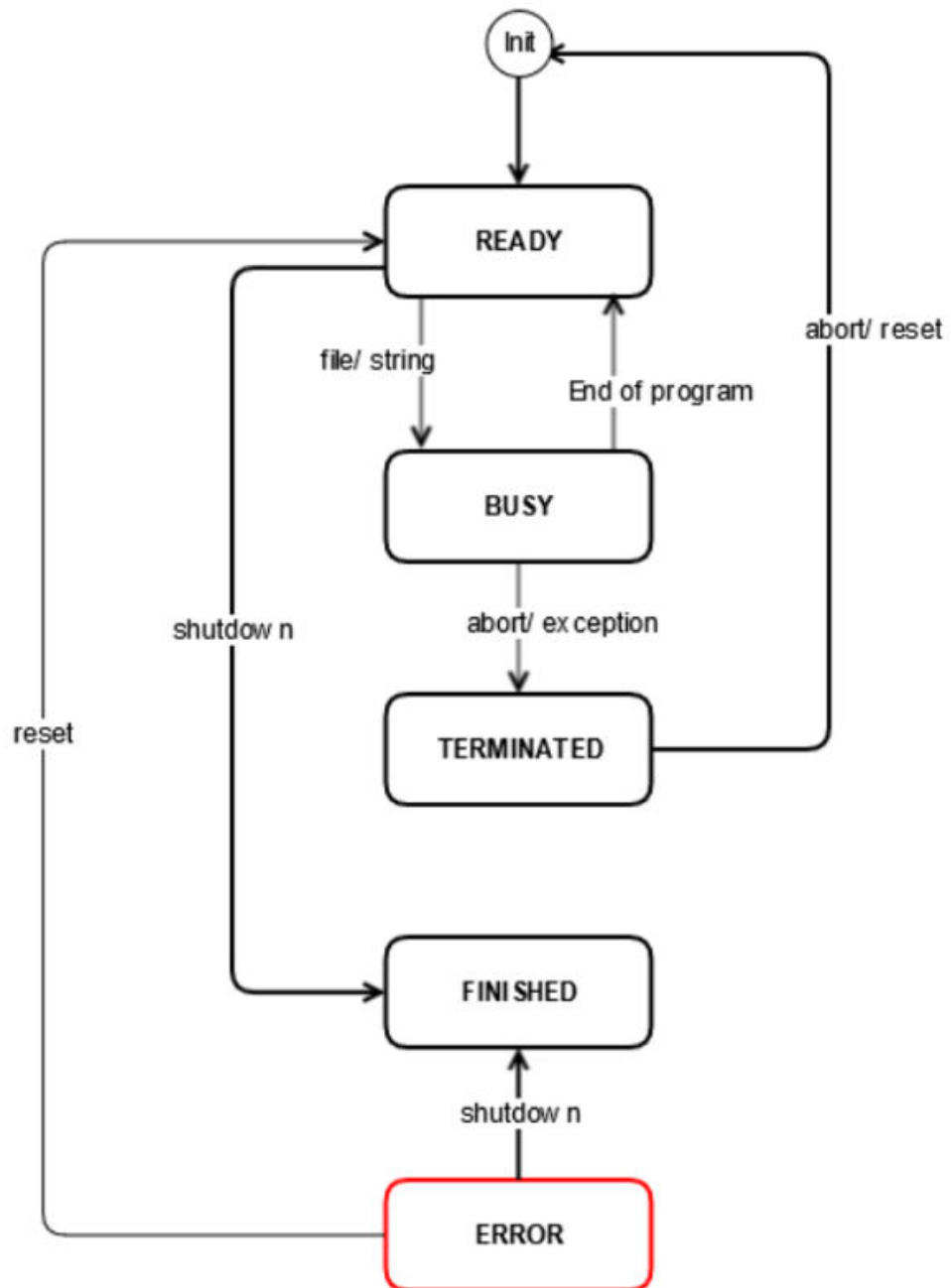


Abb. 1: Zustandsmaschine der G-Code-Instanz

Weiterführende Informationen zum Skript-Interpreter:

- ctrlX OS Runtime Anwendungsbeschreibung, Kapitel [↗ "Skript-Parser/Interpreter\(Python\)"](#)

6.3 Grundlegende Komponenten des NC-Programms

6.3.1 Programmsatz

Ein NC-Programm besteht aus mindestens einem Programmsatz.

Für einen Programmsatz gilt Folgendes:

- ### 6.3.2 Befehle

Bahnfunktionen beschreiben, auf welchem Weg eine Position angefahren werden soll (z. B. Gerade, Kreis, Anfahrbewegung im Vorschub oder Eilgang, usw.).

G0, G1, G2

Ebenso lösen Positions- oder Bahnvorgaben, die in einem Programmsatz ohne Bahnfunktion programmiert sind, immer Verfahrensbewegungen aus, da immer eine aktive Bahnfunktion vorhanden ist.

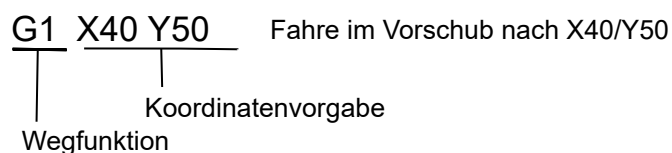


Abb. 2: Aufbau eines Befehls

- `F<number>`
Beeinflusst den Vorschub der Slave-Achsen

G1 G71 X40 Y50 F250 S500 T05 M03

Zusatzbedingungen

Fahre im Vorschub nach X40/Y50;
mit programmiertem F-Wert
(Vorschub = 250 mm/min) und
S-Wert (Drehzahl = 500 mm/min);
mit Spindel-Rechtslauf und
stelle Werkzeug T05 im
Werkzeugmagazin bereit

Abb. 3: Zusatzbedingungen

6.4 Programmwörter

6.4.1 Einführung und Übersicht

Ein Programmwort ist eine NC-Funktion oder ein Parameter mit einem Wert oder eine Parameterliste.

Ein Parameter wird im Teileprogramm über seine Syntax (Adresse) angesprochen.

Jedes Programmwort besteht immer aus einem oder zwei Teilwörtern, die wie folgt kombiniert werden können:

Tab. 1: Kombinationen von Teilwörtern

Programmwort	Teilzeichenfolge 1	Teilwort 2	Beispiel
NC-Funktion	Funktionssyntax	-	G0 G54.1 WAIT
NC-Funktion mit High-Level-Sprach-Syntax	Funktionssyntax	Parameterliste	SetSignal(1) Contour(1)
Adresswort	Parametersyntax	Wert	X-22.3 F4000

Beschreibung

G- und M-Codes können mit oder ohne Teilwort 2 vorkommen:

- Bei G- und M-Codes mit festen internen Funktionen ist der Zahlenwert Bestandteil der Syntax, z. B. G0, G17, G54, M0, M3, M19, ...
- Bei benutzerdefinierten Funktionen (Unterprogrammaufrufe und Hilfsfunktionen) ist der Zahlenwert nicht Bestandteil der Syntax, sondern stellt den Wert für die Funktion dar.

Diese Funktionen werden im Data Layer der Motion App konfiguriert.

6.4.2 Programmwörter der NC-Funktionen

Für Programmwörter der NC-Funktionen gilt Folgendes:

- Die Syntax von NC-Funktionen kann sowohl aus G- und M-Codes, als auch aus allgemeinen Sprachelementen bestehen.
Beispiele:
"G0", "G41", "G141", "G52.0", "M30", "SetSignal(...)
Alle verfügbaren NC-Funktionen mit den entsprechenden Syntaxregeln finden Sie in den Kapiteln [NC-Funktionen mit Syntax nach DIN 66025](#) und [NC-Funktionen mit Hochsprachen-Syntax](#).
- NC-Funktionen können zusätzliche Parameter enthalten, mit denen die Arbeitsweise der NC-Funktion beeinflusst werden kann.

Es werden zwei Fälle unterschieden:

- Adresswort:
Parameter, die als eigenständiges Programmwort im NC-Satz programmiert werden, z.B. "G2 X10 I1.3 J2.5 G94 F1000". Diese Parameter werden als Adresswort bezeichnet und sind hauptsächlich Parameter, die in DIN 66025 definiert sind.
- Lokale Parameter:
Parameter, die innerhalb einer Parameterliste mit bestimmten Syntaxelementen programmiert werden, die in Klammern eingeschlossen sind, z. B. "SetSignal(signalId= 10)", "ResetSignal(10)".
- Die Parameter der NC-Funktionen können optional sein.
Werden diese Parameter nicht programmiert, werden in der Regel Standardeinstellungen verwendet, die fest codiert oder im Data Layer der Motion App gespeichert sind.

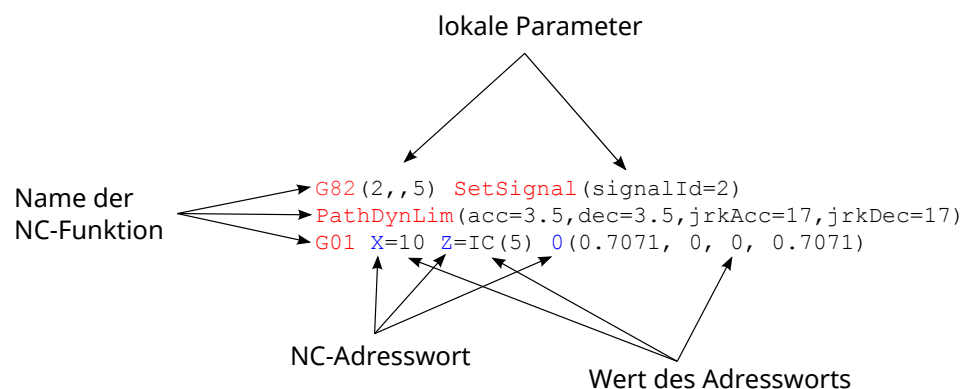


Abb. 4: Programmwörter der NC-Funktionen

6.4.3 Programmwörter als Parameter (Adresswörter)

Für Programmwörter als Parameter gilt Folgendes:

- Eine Adresse beginnt immer mit einem Buchstaben und kann aus mehreren Zeichen bestehen.
- Programmwörter mit einer Adresse und Zahlen werden z.B. zum Programmieren verwendet:
 - Achsen- und Koordinatennamen (z. B. X..., Y..., Z..., B...)
 - Radian (R...) und Interpolationsparameter (I..., J..., K...)
 - Vorschub-/Zeitwerte (F...)
 - Spindeldrehzahlen und Schnittgeschwindigkeiten (S..., Si=...)
 - Hilfsfunktionen (M..., T...)
- Führende Nullen müssen nicht programmiert werden.
- Nicht ganzzahlige Werte werden mit einem Dezimalpunkt geschrieben; nachgestellte Nullen können weggelassen werden (z. B. "X100.500" entspricht "X100.5")
- Wenn ein positives Vorzeichen (oder kein Vorzeichen) programmiert wird, wird der folgende Wert als positiv interpretiert. Ein negatives Vorzeichen deklariert einen negativen Wert.

Beispiel

N20 G0 X0 Y0	Geradlinige Interpolation im Eilgang zur Position X0/Y0 G0 wirkt modal
N30 Z100	Geradlinige Interpolation im Eilgang zur Position Z100
N40 G1 X10 Y10	Lineare Interpolation mit Vorschub F1000 auf Position X10/Y10 G1 hebt die Wirkung von G0 auf
N50 X20 F100	Lineare Interpolation auf X20 mit F100 F100 wirkt modal

6.5.2 Nicht-modal

"Nicht-modal" bedeutet, dass ein Programmwort nur in dem Programmsatz wirksam ist, in dem es programmiert wurde.

Beispiel

N10 G1 F1000	Die aktive lineare Interpolation G1 F1000 wirkt modal
N20 G75.1 X100 Y100	G75.1 Fahrt mit dem ersten Mess-taster. G75.1 wirkt nicht-modal
N20 Z100	Lineare Interpolation bei F1000 bis Z100 G1 bleibt modal aktiv

6.6 Spezielle Elemente für die Programmierung

6.6.1 Leerzeilen im Programmcode

Leerzeilen können zur Strukturierung des Programms verwendet werden. Sie erhöhen somit die Lesbarkeit des Programms. Leerzeilen werden von der Steuerung übersprungen.

6.6.2 Kommentare in einem Teileprogramm

Kommentare werden von der Steuerung bei der Ausführung des Programms übersprungen.

Sie können Kommentare nutzen, um den Programmcode zu dokumentieren oder Erklärungen einzufügen.

Se können Kommentare für ganze Programmzeilen oder für einzelne Teile nutzen.

Kommentare im Text:

- "//":
Der Text nach "//" ist ein Kommentar.
- ";":
Der Text nach ";" ist ein Kommentar.

Beispiele

```
N10 X10 ; Dies ist ein Kommentar.
N10 X10 // Dies ist ein Kommentar.
```

6.7 Unterprogramme

Unterprogramme (SR) sind Programme, die mit Hilfe eines Unterprogrammaufrufs aufgerufen werden. Sobald die Ausführung eines Unterprogramms abgeschlossen ist, läuft das aufrufende Programm an der Stelle weiter, an der das Unterprogramm aufgerufen wurde.

Das Hauptprogramm (MP) ist das Programm, von dem aus die Ausführung in die erste Unterprogrammstufe (SR-Stufe) springt.

Es wird formell nicht zwischen Hauptprogrammen und Unterprogrammen unterschieden.

Es gilt:

- Unterprogramme können Standard-NC-Sätze und CPL-Sätze enthalten.
- Jedes Teilprogramm kann von anderen Programmen als Unterprogramm aufgerufen werden. Ein Programm ist jedoch nicht in der Lage, sich selbst als Unterprogramm aufzurufen (**ein rekursiver Aufruf ist nicht möglich**).
- Das aufrufende Programm kann keine Parameter an ein Unterprogramm übergeben.
- Die maximale Schachtelungstiefe beträgt 24, d.h. die Steuerung kann maximal 24 Unterprogrammebenen gleichzeitig offen halten.
- Bei den Namen der Unterprogramme wird zwischen Groß- und Kleinschreibung unterschieden.

Die maximal zulässige Unterprogrammverschachtelung wird beispielsweise schematisch dargestellt. N1 ist der erste Satz des Programms. Das nächste Unterprogramm wird im Satz N9 aufgerufen. Das Erreichen des Dateiendes schließt das jeweilige Unterprogramm ab.

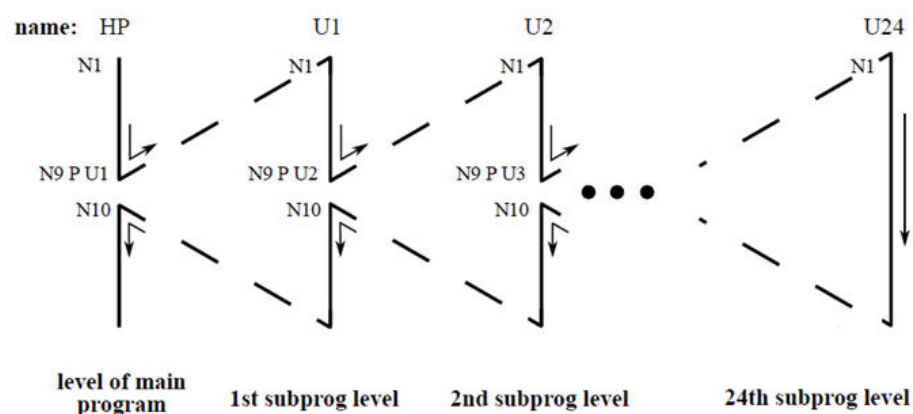


Abb. 6: Verschachtelung Unterprogramm

Auswirkungen von Unterprogrammen:

- Der Unterprogrammaufruf ist lokal wirksam (nicht modal).

6.7.1 Aufruf des Unterprogramms

Syntax

- Aufruf mit P-Adresse und Unterprogrammnamen aus einem Standard-NC-Satz (optional mit relativer **Pfadangabe**).
- Aufruf Unterprogramm ohne P-Adresse aus einem Standard-NC-Satz

Beispiel

```
P Subroutine
Subroutine2
P folder/Sub.npg
folder/sub.npg
```

6.7.2 Suchpfad des Unterprogramms

Der Ordner, in dem sich das Hauptunterprogramm befindet, wird mit höchster Priorität durchsucht.

Außerdem ist es möglich, den Suchpfad unter dem Data Layer-Pfad *g-code/cfg-instances/[Instance]/cfg/subroutine/path* hinzuzufügen, wobei die Variable [Instance] mit der G-Code-Skriptinstanz (sowie dem Kinematik-Namen) übereinstimmen muss.

7 NC-Funktionen mit Syntax nach DIN 66025 (inkl. Erweiterungen)

7.1 Einführung und Übersicht

In diesem Kapitel werden die NC-Funktionen mit der Syntax nach DIN 66025 vorgestellt.

Die freigegebenen Funktionen sind in [Kapitel 10.1.1 NC-Funktionen mit Syntax nach DIN 66025 auf Seite 35](#) aufgeführt.

7.2 G-Codes

7.2.1 Geradeninterpolation im Eilgang "G0"

Die NC-Funktion G0 fährt eine programmierte Position mit Interpolation auf einer Geraden im Eilgang an.

Mindestens eine Achse verfährt mit maximaler Geschwindigkeit oder Beschleunigung. Die Geschwindigkeit der Achsen wird durch die maximalen Dynamikgrenzen der konfigurierten Kinematik begrenzt und so gesteuert, dass alle Achsen gleichzeitig den Zielpunkt erreichen.

Die Funktion G0 ist modal und löscht die anderen modalen Funktionen, z.B. G1.

Syntax

Name der Funktion	G0, G00, G000
Modale Gruppe	Geo

Beispiel

```
N10 G0 X0 Y0 Z0 ; Anfahren der Ausgangsposition
N20 X10 Y10 Z0 ; Zielposition anfahren mit max. konfigurierten dynamischen Grenzen
```

Konfiguration

Bei Verwendung von G0 wird die dynamische Grenze der Kinematik in folgendem Data Layer-Pfad konfiguriert:

motion/kin//cfg/lim/**

Zugehöriger Motion-Befehl

moveABS

7.2.2 Geradeninterpolation im Vorschub"G1"

Die NC-Funktion G1 fährt eine programmierte Position durch Interpolation auf einer Geraden mit effektivem Vorschub an.

Die Bewegung wird so koordiniert, dass alle beteiligten Achsen gleichzeitig den programmierten Endpunkt erreichen. Der programmierte Vorschubwert (F) wirkt als Bahnvorschub. Beim Verfahren mehrerer Achsen sollte der Vorschub jeder Achse kleiner sein als F. Es wird ein Fehler gemeldet, wenn $F<number>$ nicht explizit programmiert ist.

Die Funktion G1 ist modal und löscht die anderen modalen Funktionen z.B.: G0.

Eigenschaften

Der programmierte Vorschub bleibt wirksam, bis er durch einen neuen Vorschubwert überschrieben wird.

Die programmierte Bahngeschwindigkeit kann durch den Data Layer begrenzt werden.

Die dynamische Grenze (Beschleunigung/Verzögerung/Ruck) kann durch die Funktion PathDynLim geändert werden.

Syntax

Name des Programms	G1, G01, G001
Modale Gruppe	Geo

Beispiel

```
N10 G0 X0 Y0 Z0      ; Anfahren der Ausgangsposition
N20 G1 X10 Y10 Z10 F10 ; Lineares Anfahren des Ziels mit einem Vorschub von F10
```

Konfiguration

Wenn PathDynLim ausgeschaltet ist, verwendet die Motion die standardmäßige dynamische Grenze der Kinematik, die in folgendem Data Layer-Pfad konfiguriert werden kann:

motion/kin//cfg/lim/**

Zugehöriger Motion-Befehl

moveABS

7.2.3 Kreisinterpolation im Rechts-/Linkslauf "G2"/"G3"

Die NC-Funktion G2/G3 fährt eine programmierte Position in der aktiven Bearbeitungsebene interpolierend auf einer Kreisbahn im Rechts-/Linkslauf mit dem aktuellen Vorschub (programmierbar über die F-Adresse) an. Zusätzliche Achsen außerhalb der Bearbeitungsebene können mit linear interpolierten Bewegungen in die Programmierung einbezogen werden. Die auf dem Kreisbogen verfahrenen Achsen werden durch die gewählte Bearbeitungsebene (G17, G18, G19, G20) definiert. Der programmierte Vorschubwert (F) wirkt als Bahnvorschub. Beim Verfahren mehrerer Achsen sollte der Vorschub jeder Achse kleiner als F sein. Ein Fehler wird gemeldet, wenn kein Wert $F<Zahl>$ explizit programmiert ist.

Die Funktion ist modal und löscht die Funktionen G0, G2, G3, G5, G6, G12, G13, G33.

Syntax

Name der Funktion	G2, G02, G002 G3, G03, G003
Modale Gruppe	Geo

Die Kreisbahn kann mit Hilfe der Radiusprogrammierung oder der Mittelpunktprogrammierung programmiert werden.

- Radiusprogrammierung:

Bedeutung	Wert
<EP>	Koordinaten des Endpunkts
F	Vorschubwert
R	<ul style="list-style-type: none"> • Radius der Kreisbahn • Der Radius muss mindestens die Hälfte der Entfernung zwischen dem Start- und dem Endpunkt betragen. • Positiver Radiuswert bedeutet, dass entlang des kleineren Kreisbogens (Arc ≤ 180 Grad) verfahren wird. Negativer Radiuswert bedeutet, dass entlang des größeren Kreisbogens (Arc > 180 Grad) verfahren wird. • $Radius = Abstand(Endpunkt, Startpunkt)/2$ ergibt einen Halbkreis, wobei jedes Vorzeichen (positiv/negativ) verwendet werden kann.

- Mittelpunktprogrammierung

Bedeutung	Wert				
<EP>	Koordinaten des Endpunkts Sind Anfangs- und Endpunkt innerhalb der Kreisebene identisch, wird automatisch ein Vollkreis generiert.				
F	Vorschubwert				
<IP>	<table border="1"> <tr> <td>I(X-Koordinate)</td><td rowspan="3"> Interpolationsparameter <ul style="list-style-type: none"> • I, J und K definieren den Abstand zwischen Mittelpunkt und Startpunkt entlang der jeweiligen Achse • Ein ungenauer Mittelpunkt kann automatisch entsprechend der eingestellten Toleranz korrigiert werden. • Wenn die Interpolationsparameter für die ausgewählte Position nicht geeignet sind, werden durch die Bewegung Fehler erzeugt. </td></tr> <tr> <td>J(Y-Koordinate)</td></tr> <tr> <td>K(Z-Koordinate)</td></tr> </table>	I(X-Koordinate)	Interpolationsparameter <ul style="list-style-type: none"> • I, J und K definieren den Abstand zwischen Mittelpunkt und Startpunkt entlang der jeweiligen Achse • Ein ungenauer Mittelpunkt kann automatisch entsprechend der eingestellten Toleranz korrigiert werden. • Wenn die Interpolationsparameter für die ausgewählte Position nicht geeignet sind, werden durch die Bewegung Fehler erzeugt. 	J(Y-Koordinate)	K(Z-Koordinate)
I(X-Koordinate)	Interpolationsparameter <ul style="list-style-type: none"> • I, J und K definieren den Abstand zwischen Mittelpunkt und Startpunkt entlang der jeweiligen Achse • Ein ungenauer Mittelpunkt kann automatisch entsprechend der eingestellten Toleranz korrigiert werden. • Wenn die Interpolationsparameter für die ausgewählte Position nicht geeignet sind, werden durch die Bewegung Fehler erzeugt. 				
J(Y-Koordinate)					
K(Z-Koordinate)					

Beispiel

```

N10 G0 X0 Y0 Z0      ; Anfahren der Ausgangsposition
N20 G17              ; Aktivieren der XY-Ebene
N30 G2 X20 Y0 Z10 F10 ; Kreisinterpolation im Uhrzeigersinn mit Radiusparameter in der
XY-Ebene, lineare Interpolation entlang der Z-Achse
N50 G3 X40 Y0 I10 J0  ; Kreisinterpolation im Gegenuhrzeigersinn mit dem Parameter
Mittelpunkt

```

Konfiguration

Wenn PathDynLim ausgeschaltet ist, verwendet die Motion die voreingestellte dynamische Grenze der Kinematik, die in folgendem Data Layer-Pfad konfiguriert werden kann:

motion/kin//cfg/lim/**

Zugehöriger Motion-Befehl*moveCircle2D***7.2.4 Bahnslope "G8", "G9"**

Die NC-Funktion G8 aktiviert und die NC-Funktion G9 deaktiviert Bahnslope. Ohne Bahnslope wird am Anfang eines NC-Satzes auf die programmierte Geschwindigkeit beschleunigt und am Ende des NC-Satzes wieder auf $v=0$ abgebremst. Dies reduziert zwar die Konturabweichung am Satzübergang, erfordert aber eine längere Bearbeitungszeit. Bei aktivem Bahnslope versucht die Steuerung, eine möglichst konstante Geschwindigkeit zu erzeugen, die so hoch wie der programmierte Vorschub ist, was die Bearbeitungszeit reduziert und die Kontur an den Ecken glätten kann.

Syntax

Name der Funktion G8, G9
Modale Gruppe Bahnslope

Beispiel

```
N10 G8                                      ; Aktivieren von Bahnslope
CONTOUR(3)
N20 G90 X0 Y0 Z0 F400   ; Bahnslope wird bei den folgenden Bewegungen wirksam
N30 X100 Y0 Z0
N40 X100 Y100 Z0
CONTOUR()
N30 G9                                      ; Deaktivieren von Bahnslope
```

Zugehöriger Motion-Befehl*KinContMotion***7.2.5 Ebenenumschaltung "G16"/"G17-19"**

Die NC-Funktion G16 deaktiviert die aktuelle aktive Ebene. Die Funktionen G17-G19 aktivieren die ausgewählten Ebenen. Die aktive Ebene wird durch eine Haupt- und eine Nebenkoordinate definiert, eine Zustellachse steht senkrecht zur Ebene. Nach Auswahl der aktiven Ebene werden die Interpolationsparameter I,J,K für die Kreisbewegung ausgewertet.

	Hauptkoordinate	Nebenkoordinate	Zustellkoordinate	Aktive Ebene
G17	X	Y	Z	XY
G18	Z	X	Y	ZX
G19	Y	Z	X	YZ

Die Funktionen G16, G17, G18, G19 und G20 bilden eine modale Gruppe und heben sich daher gegenseitig auf.

Syntax

Name der Funktion G16, G016
 G17, G017
 G18, G018
 G19, G019
Modale Gruppe Vor_Geo

Beispiel

```
N10 G17                                      ; Aktivieren der XY-Ebene
N20 G2 X38 Y20 R15   ; Kreisinterpolation im Uhrzeigersinn mit Radiusparameter in der XY-
Ebene
N30 G16                                      ; Deaktivieren der XY-Ebene
```

Zugehöriger Motion-Befehl*kinActPlane***7.2.6 Werkzeuglängenkorrektur "G47", "G48"**

Die NC-Funktion G47/G48 aktiviert/deaktiviert die Werkzeuglängenkorrektur.

- Die Klammern und der Inhalt von G47 sind optional.
- Wenn G47 ohne Klammern und Parameter programmiert ist, ruft die Funktion eine PCS-Satzgruppe mit dem Namen "G47" auf.
Bevor die Werkzeuglängenkorrektur aktiviert wird, sollten PCS-Sätze mit dem Namen "G47" vorhanden sein.

G47, G48 wirken modal und heben sich gegenseitig auf.

Syntax

Name der Funktion G47{(SET=<setName>)}

G48

Modale Gruppe Werkzeuglängenkorrektur

Aktivieren G47{(SET=<setName>)}

G47

Deaktivieren G48

Parameter	Schlüsselwort	Positionsindex	Bedeutung
	SET	1	Name des ausgewählten PCS-Satzes

Beispiel

```

N10 G1 G47 ; Aktivieren des PCS-Werkzeugs "G47"
N20 G90 X10 Y10 Z10 F100 ; Anfahren von X10 Y10 Z10 mit aktivierter
Werkzeuglängenkorrektur
N30 G48 ; Deaktivieren der Werkzeuglängenkorrektur
N30 G47(SET="set1") ; Aktivieren des PCS-Werkzeugsatzes "set1"

```

Konfiguration

Die Werkzeuglängenkorrekturwerte der Kinematik werden unter folgendem Data Layer-Pfad konfiguriert:

*motion/cfg/coord-systems/pcs/[set]/**

Fügen Sie "PCS sets group" mit dem Namen "G47" manuell mit den Nutzdaten der PCS-Satz-(Gruppen)-Namen hinzu.

Zugehöriger Motion-Befehl*KinPCSTool***7.2.7 Nullpunktverschiebungstabelle auswählen "G53", "G54-G59"**

Die NC-Funktionen G54 bis G59 wählen und aktivieren die Nullpunktverschiebungstabellen (z.B. Zero Offset G54), um das Maschinenkoordinatensystem im Raum zu verschieben. Die NC-Funktion G53 deaktiviert alle Nullpunktverschiebungstabellen. Die Aktivierung der Nullpunktverschiebung basiert auf der Motion-Befehlsoption *opt-pcs* (Befehlsoptionen für Kinematik-Produktkoordinatensystem). Die ZO-Tabelle muss vorhanden und unter PCS-Sets gespeichert sein, bevor sie aktiviert wird.

G53 bis G59 wirken modal und heben sich gegenseitig auf.

Syntax

Name der Funktion	G53-G59
Modale Gruppe	Nullpunktverschiebungstabelle auswählen

Beispiel

```

N10 G1 G54          ; Aktivieren von PCS mit der ZO-Tabelle von G54
N20 G90 X10 Y10 Z10 F100 ; Anfahren von X10 Y10 Z10 auf Grundlage von PCS
N30 G53             ; Alle ZO-Tabellen deaktivieren

```

Konfiguration

Die Offset-Abstände für die Maschinenkoordinaten in der Kinematik können in folgendem Data Layer-Pfad konfiguriert werden:

*motion/cfg/coord-systems/pcs/[set]/**

Fügen Sie "PCS sets group" mit dem Namen "G54"... "G59" manuell mit den Nutzdaten der PCS-Satznamen hinzu.

Zugehöriger Motion-Befehl

KinPCSP

7.2.8 Absolutmaß-Programmierung "G90", Relativmaß-Programmierung "G91"

Die NC-Funktionen G90 und G91 legen fest, ob die Steuerung die Maßangaben für die Achsen und Koordinaten als absolute oder relative (inkrementelle) Werte interpretieren soll. Die absolute Bewegung G90 bezieht sich auf den aktuellen Nullpunkt im Programmkoordinatensystem (PCS).

G90 und G91 wirken modal und heben sich gegenseitig auf.

Syntax

Name der Funktion	G90/G91
Modale Gruppe	Programmiermodus

Beispiel

```

N10 G1 G90 F100      ; Programmierung der absoluten Abmessungen EIN
N20 X100 Y100        ; Anfahren der Koordinaten X100, Y100
N30 G91 X100 Y100    ; Programmierung der relativen Abmessungen EIN, Anfahren von X200, Y200

```

Zugehöriger Motion-Befehl

moveABS/moveREL

7.2.9 Vorschubprogrammierung "G94"

Die NC-Funktion G94 interpretiert die F-Wörter als Vorschub für eine programmierte Kontur. Das F-Wort gibt die Geschwindigkeit in den dynamischen Grenzen eines bestimmten Motion-Befehls an.

G94 ist der standardmäßig aktive "Vorschubmodus".

Syntax

Name der Funktion	G94
Modale Gruppe	Vorschubmodus

Beispiel

```

N10 G94 G90 F10      ; Einstellen des Vorschubmodus mit Vorschubgeschwindigkeit 10
N20 X100 Y100        ; Anfahren der Koordinaten X100, Y100

```

Konfiguration

Die Einheit der Geschwindigkeit von G94, die standardmäßig auf "mm/min" eingestellt ist, kann in folgendem Data Layer-Pfad konfiguriert werden:

motion/kin//cfg/units/velocity*

Zugehöriger Motion-Befehl

moveABS

7.2.10 Produktkoordinatensystem "G152", "G153"

Die NC-Funktion G152/G153 aktiviert/deaktiviert die ausgewählten PCS-Sätze. PCS-Sätze können verwendet werden, um das Maschinenkoordinatensystem im Raum zu verschieben.

- Die Aktivierung von PCS-Offsets basiert auf der Motion-Befehlsoption *opt-pcs* (Befehlsoptionen für Kinematik-Produktkoordinatensystem).
- Die Offset-Sätze für die Maschinenkoordinaten in der Kinematik werden im Verzeichnis "motion/cfg/coord-systems/pcs/[set]/*" gespeichert.
- Fügen Sie die PCS-Satzgruppe mit dem eingegebenen Satznamen manuell mit der Nutzlast der PCS-Satz(gruppen)namen hinzu.
- Vor dem Aktivieren des PCS-Satzes sollte der erwartete Offset-Abstand editierbar sein.
- Wenn G152 mit einem bestimmten PCS-Satz programmiert wird, erzeugt der G-Code Interpreter eine permanente kinematische Motion-Befehlsoption: *opt-pcs("set")*
Wenn das ausgewählte PCS-Set nicht existiert, wird ein Fehler von *motion.core* erzeugt.
- Der neue G152-Befehl mit dem Namen des ausgewählten PCS-Satzes löscht den vorherigen aktiven PCS-Satz.

Syntax

Name der G152

Funktion G153

Aktivieren G152(SET = <setName>)

G152

Deakti- G153
vieren

Parameter	Schlüsselwort	Positionsindex	Bedeutung
	SET	1	Name des ausgewählten PCS-Satzes

Beispiel

```
G152("SetA")          ; Aktivieren des PCS mit der Offset-Tabelle "SetA"  
N10 G1 G90 X10 Y10 Z10 F100 ; Anfahren der Zielposition im ausgewählten PCS  
G153                  ; Deaktivieren aller PCS-Tabellen  
G152("SetB")          ; Aktivieren des PCS mit der Offset-Tabelle "SetB"
```

Zugehöriger Motion-Befehl

motion/kin/[name]/cmd/opt-pcs

8 NC-Funktionen mit Hochsprachen-Syntax

8.1 Einführung und Übersicht

In diesem Kapitel werden die NC-Funktionen mit Hochsprachen-Syntax und einige spezielle Motion-Befehle vorgestellt.

Die freigegebenen Funktionen sind in [Kapitel 10.1.2 NC-Funktionen mit Hochsprachen-Syntax auf Seite 37](#) aufgeführt.

8.2 Globale Signale

8.2.1 "SetSignal"

Die Funktion "SetSignal" löst zur Laufzeit ein globales Signal aus. Damit wird der Pegel des programmierten Ereignisses auf 1 gesetzt (unabhängig davon, ob der Wert 0 oder 1 war). Es können mehrere "SetSignal"-Funktionen in einem Satz programmiert werden. Das Signal wird durch die programmierte Reihenfolge ausgelöst.

Syntax

Name der Funktion	SetSignal SETSIGNAL
Kurzform	SSG
Parameter	ID = <number> * Der Index der Signal-ID wird ausgelöst ** Der Wertebereich für Signal-ID liegt zwischen 0 - 99 für das Gesamtsystem

Beispiel

```
N10 SetSignal(1) ; Signal 1 wird ausgelöst
N20 SSG(2) ; Signal 2 wird ausgelöst
N30 SetSignal(ID =3) ; Signal 3 wird ausgelöst
N40 SetSignal(4) Delete SetSig(5) SSG(6) ; Signale 4,5,6 werden ausgelöst
```

Zugehöriger Motion-Befehl

motion/kin//cmd/set-signal*

8.2.2 "ResetSignal"

Die Funktion "ResetSignal" setzt ein Signal zur Laufzeit zurück. Es können mehrere "ResetSignal"-Funktionen in einem Satz programmiert werden. Das Signal wird durch die programmierte Reihenfolge ausgelöst.

Syntax

Name der Funktion	ResetSignal RESETSIGNAL
Kurzform	RSG
Parameter	ID = <number> * Der Index der Signal-ID wird zurückgesetzt ** Der Wertebereich für Signal-ID liegt zwischen 0 - 99 für das Gesamtsystem

Beispiel

```
N10 ResetSignal(1) ; Zurücksetzen Signal 1
N20 RSG(10)       ; Zurücksetzen Signal 10
```

Zugehöriger Motion-Befehl

motion/kin//cmd/reset-signal*

8.2.3 "WaitForSignal"

Mit der Funktion "WaitForSignal" wird eine Motion-Befehlssequenz generiert, die zur Laufzeit abwartet, bis ein Signal ausgelöst wird.

Es können mehrere "WaitForSignal"-Funktionen in einem Satz programmiert werden. Der Ausgabezeitpunkt des Signals wird durch die programmierte Reihenfolge festgelegt.

Syntax

Name des Programms	WaitForSignal WAITFORSIGNAL
Kurzform	WSG
Parameter	ID = <number> * Warten auf ein Signal ** Der Wertebereich für Signal-ID liegt zwischen 0 - 99 für das Gesamtsystem

Beispiel

```
N10 WaitForSignal(1) ; Warten auf Signal 1
N20 WSG(10)         ; Warten auf Signal 10
```

Zugehöriger Motion-Befehl

motion/kin//cmd/wait-for-signal*

8.3 M-Code

M-Codes werden für die Synchronisierung von Signalen zwischen G-Code Interpreter und SPS-Anwendung verwendet.

Merkmale

- Insgesamt 100 M-Codes werden von G-Code für alle Kinematiken unterstützt (M0-M99)
- M0-M99 sind quittierungspflichtig
- In einem NC-Programm sind alle M-Codes verfügbar
- In einem NC-Satz können alle M-Codes programmiert werden
- Es werden nur globale M-Codes unterstützt (keine kinematikspezifischen)

Beispiel

Das Verhalten der M-Codes ist identisch mit dem der MTX.

Fall	NC-Satz	Beschreibung
Einzelner M-Code in einem Satz	M20	M0-M99 sind quittierungspflichtig.
Gleiche M-Codes in einem Satz	M17 M17 M38 M17	Gleiche M-Codes werden automatisch zusammengeführt und nur einmal ausgeführt.

Fall	NC-Satz	Beschreibung
M-Codes in verschiedenen Sätzen	N100 M20 N110 M21	
M-Codes in einem Satz	N100 M20 M21	Signale, die den M-Codes entsprechen, werden durch die Programmiersequenz ausgelöst, der NC-Satz ist abgeschlossen, bis alle Signale zurückgesetzt wurden.
M-Codes und Positions-befehle im gleichen Satz	N10 M210 M169 X10 M184	Alle Signale werden vor der Bewegung ausgelöst, der NC-Satz ist abgeschlossen, bis die Zielposition erreicht ist und alle Signale zurückgesetzt wurden.

Lesen/Schreiben von M-Codes im SPS-Projekt

Verwendung folgender Funktionsbausteine in der Bibliothek CXA_Motion:

- ML_GetSignal: Ermittelt den Status des Signals
- ML_SetSignal: Setzt ein Signal (auf TRUE)
- ML_ResetSignal: Setzt ein Signal zurück (auf FALSE)

Verwendung folgender Funktionsbausteine in der Bibliothek CXA_PLCOpen:

- MB_GetSignal: Ermittelt den Status des Signals
- MB_SetSignal: Setzt ein Signal (auf TRUE)
- MB_ResetSignal: Setzt ein Signal zurück (auf FALSE)

Data Layer-Pfad zur Überwachung von Signalen:

- *motion/state/functions/somo/signals*

Zugehörige Motion-Befehle

- *motion/kin/*/cmd/set-signal*
- *motion/kin/*/cmd/wait-for-signal*

8.4 "Contour"

Mit dieser Funktion lässt sich der Bereich einer Kontur aus kinematischen Bewegungen definieren. Für jeden Parameter wird angegeben, ob es sich um den Anfang oder das Ende der Kontur handelt.

Durch die Definition des Konturbereichs kann die ctrlX Motion sicherstellen, dass die Konturbewegungen in der gewünschten Weise abgearbeitet werden. Zusätzlich kann über einen Parameter gesteuert werden, dass am Anfang der Kontur gewartet wird, bis eine bestimmte Anzahl von Folgebefehlen vollständig vorbereitet ist.

Die Funktionen für den Start der Kontur und dem Ende der Kontur müssen paarweise programmiert werden, andernfalls wird von der ctrlX Motion ein Fehler gemeldet.

Syntax

Name des Programms	Contour CONTOUR
Start der Kontur	Contour(PREPCMDS = <prepCmds>)

Ende der Kontur	Contour()
Parameter	PREPCMDS = <prepCmds> * <prepCmds> ist ein vorzeichenloser Integer-Parameter für den Start einer Kontur. Der Parameter gibt an, wie viele Befehle vollständig vorbereitet werden sollen. Er muss >=0 sein, und der Standardwert ist 0. ** Der leere Parameter bedeutet das Ende der Kontur.

Beispiel

```
N10 Contour(prepCmds=4) ; Start einer Kontur, nach Erreichen der Anzahl der
                        ; eingestellten Befehle beginnt die Kinematik sich zu bewegen.
N20 X10 Y10 Z10        ; Befehl wird dem Befehlspuffer hinzugefügt, aber keine
                        ; Bewegungen bis zum Ende der Kontur.
...
N50 X30 Y10 Z10
N60 Contour()          ; Ende der Kontur, Kinematik beginnt sich zu bewegen.
```

Zugehöriger Motion-Befehl

motion/kin//cmd/contour*

8.5 "WAIT"

Die Funktion "WAIT" kann verwendet werden, um die Vorbereitung vorheriger Fahrbefehle zu stoppen und die Ausführung des G-Code-Skripts anzuhalten, bis alle vorherigen Fahrbefehle ausgeführt wurden.

Syntax

Name der Funktion	WAIT
Parameter	Ohne Parameter

Beispiel

```
N10 WAIT              ; Stoppen der Ausführung dieses Skripts.
```

Zugehöriger Motion-Befehl

motion/kin//cmd/pre-wait*

8.6 "PolyTrans"

Mit dieser Befehlsfunktion wird das Blending dauerhaft aktiviert, wobei zwischen aufeinanderfolgenden Fahrbefehlen eine Verrundung mittels Spline-Funktion eingefügt wird, bis das Blending wieder explizit deaktiviert wird.

Syntax

Name des Programms	PolyTrans POLYTRANS
Abkürzung	PTR
Aktivieren	PolyTrans(D1=<number>, D2 =<number>) PolyTrans(EPS=<number>)

Deaktivieren	PolyTrans()
Parameter	<p>* Die Parameter D1 und D2 können nicht gleichzeitig mit dem Parameter EPS verwendet werden.</p> <p>** Ein leerer Parameter bedeutet Ausschalten.</p>

Beispiel

```

N10 PolyTrans(D1=1, D2=1) ; Polytrans dauerhaft einschalten, und definieren
                           ; der Ecke mit den Parametern D1 und D2.

N20 X10 Y10 Z10
N30 X30 Y10 Z10
...
N100 PolyTrans()          ; Polytrans dauerhaft ausschalten.

```

Zugehöriger Motion-Befehl

motion/kin//cmd/pt-poly-trans*

8.7 "PathDynLim"

Diese Befehlsfunktion reduziert im Teileprogramm die Obergrenzen für die Geometrie-NC-Funktion G1, G2, G3, G5, G6, G12, G13, G33.

- Bahnbeschleunigung
- Bahnverzögerung
- Bahn-Ruck-Beschleunigung
- Bahn-Ruckverzögerung

Wenn PythDynLim nicht aktiviert ist, bewegt sich die Kinematik unter Verwendung der konfigurierten dynamischen Grenzen im Pfad *motion/kin/*/cfg/lim*.

Wenn PathDynLim aktiviert ist, bewegt sich die Kinematik anhand der programmierten dynamischen Grenzwerte.

Wenn PathDynLim ausgeschaltet ist, bewegt sich die Kinematik wieder mit den konfigurierten dynamischen Grenzwerten.

Syntax

Name des Programms	PathDynLim PATHDYNLIM		
Abkürzung	PDL		
Aktivieren	PathDynLim(ACC=<value>, DEC=<value>, JRKACC=<value>, JRKDEC=<value>) PathDynLim(ACC=<value>, DEC=<value>) PathDynLim(<value>, <value>, <value>, <value>) PathDynLim(, , , <value>) PathDynLim()		
Deaktivieren	PathDynLim()		
Parameter	Schlüsselwort	Positions-index	Bedeutung
	ACC	1	Double, neue Bahnbeschleunigung. Muss > 0 sein. Optional. Wenn nicht programmiert, wird die Bahnbeschleunigung nicht geändert.
	DEC	2	Double, neue Bahnverzögerung. Muss > 0 sein. Optional. Wenn nicht programmiert, wird die Bahnverzögerung nicht geändert.

JRKACC	3	Double, neue Bahn-Ruck-Beschleunigung. Muss > 0 sein. Optional. Wenn nicht programmiert, wird die Ruck-Beschleunigung nicht geändert.
JRKDEC	4	Double, neue Bahn-Ruckverzögerung. Muss > 0 sein. Optional. Wenn nicht programmiert, wird die Ruck-Beschleunigung nicht geändert.

Beispiel

```
N10 G1 X10 Y10 Z10 ; Anfahren der Zielposition mit dynamischen  
Standardgrenzwerten  
N20 PathDynLim(acc=1,dec=2) ; Anfahren der Zielposition mit dynamischen Standardgrenzwerten  
N30 G1 X30 Y10 Z10 ; Anfahren der Zielposition mit neuen dynamischen Grenzwerten  
N20 PathDynLim() ; Zurücksetzen der dynamischen Grenzwerte auf Standardwerte
```

Zugehöriger Motion-Befehl

Konfiguration:

Die dynamischen Grenzwerte für den Standardpfad werden in folgendem Data Layer-Pfad konfiguriert:

- *motion/kin/*/cfg/lim/acc*
- *motion/kin/*/cfg/lim/dec*
- *motion/kin/*/cfg/lim/jrkAcc*
- *motion/kin/*/cfg/lim/jrkDec*

Die Einheiten der dynamischen Grenzwerte können mit folgendem Data Layer-Pfad gelesen werden:

- *motion/kin/*/cfg/lim*

Die Einheiten der dynamischen Grenzwerte können mit folgendem Data Layer-Pfad geändert werden:

- Ruck: *motion/kin/*/cfg/units/jerk*
- Beschleunigung: *motion/kin/*/cfg/units/acceleration*

8.8 "AxsDynLim"

Die Funktion reduziert permanent die maximalen dynamischen Grenzen einer kinematischen Achse für alle folgenden Fahrbefehle.

Syntax

Name des Programms	AxsDynLim AXSDYNLIM		
Abkürzung	ADL		
Aktivieren	<pre>AxsDynLim(AXS=<name>{ {, ACC=<value>} {, DEC=<value>} {, JRKACC=<value>} {, JRKDEC=<value>} }) AxsDynLim(<name>, <value>, <value>, <value>, <value>, <value>)</pre>		
Deaktivieren	<pre>AxsDynLim(<name>) AxsDynLim(<name>, , , , ,) AxsDynLim(<name>, <0>, <0>, <0>, <0>, <0>) AxsDynLim(AXS=<name>, <0>, <0>, <0>, <0>, <0>)</pre>		
Parameter	Schlüsselwort	Positionsindex	Bedeutung
	AXS	1	String, Name der Achse

VEL	2	Double, neuer Geschwindigkeitsgrenzwert. Optional. Der zuletzt programmierte Beschleunigungsgrenzwert wird verwendet, wenn dieser Parameter nicht programmiert ist.
ACC	3	Double, neuer Beschleunigungsgrenzwert. Optional. Der zuletzt programmierte Verzögerungsgrenzwert wird verwendet, wenn dieser Parameter nicht programmiert ist.
DEC	4	Double, neuer Verzögerungsgrenzwert. Optional. Der zuletzt programmierte Beschleunigungsgrenzwert wird verwendet, wenn dieser Parameter nicht programmiert ist.
JRKACC	5	Double, neuer Ruck-Beschleunigungsgrenzwert. Optional. Der zuletzt programmierte Beschleunigungsgrenzwert wird verwendet, wenn dieser Parameter nicht programmiert ist.
JRKDEC	6	Double, neuer Ruck-Verzögerungsgrenzwert. Optional. Der zuletzt programmierte Beschleunigungsgrenzwert wird verwendet, wenn dieser Parameter nicht programmiert ist.

Beispiel

```

N10 G1 G90 X10 Y10 Z10 F100 ; Anfahren der Zielposition mit dynamischen
Standardgrenzwerten
N30 AxsDynLim(AXS="X", VEL=10, ACC=1, DEC=1, JRKACC=5, JRKDEC=5) ; Die dynamischen Grenzen
der "X"-Achse weiter reduzieren
N40 X20 Y20 Z20 ; Anfahren der linearen Zielposition mit neuen dynamischen
Grenzwerten
N50 AxsDynLim(AXS="X") ; Zurücksetzen der dynamischen Grenzwerte von "X" auf die
Standardkonfiguration

```

Zugehöriger Motion-Befehl

Konfiguration:

Die dynamischen Grenzwerte der Achsen können mit folgendem Data Layer-Pfad gelesen werden:

- *motion/axs/*/cfg/lim*

Die Einheiten der dynamischen Grenzwerte können mit folgendem Data Layer-Pfad geändert werden:

- Ruck: *motion/axs/*/cfg/units/jerk*
- Beschleunigung: *motion/axs/*/cfg/units/acceleration*

Folgender Data Layer-Pfad hat die gleichen Auswirkungen wie die Funktion AxsDynLim:

- *motion/kin/*/cmd/opt-axs-dyn-lim*

9 Erweiterte Programmierung mit High-Level-Syntax

9.1 Einführung und Übersicht

Die Verwendung der High-Level-Programmiersyntax macht ein Programm flexibler und führt zu einer besseren Programmierstruktur.

Die erweiterte Programmierung umfasst:

- Variablen und Berechnungen
- Flusskontrolle: Entscheidungen und Schleifen
- Sprunganweisungen

9.2 Typ und Werte

ctrlX G-Code ist eine dynamisch typisierte Sprache.

Dynamisch typisierte Variablen bedeutet:

- Es gibt keine Typdefinitionen in der Sprache, sondern sie werden zur Laufzeit festgelegt.

Grundtypen in ctrlX G-Code:

Typ	Beschreibung	Beispiel
NIL	Stellt das Fehlen eines nützlichen Wertes dar	NIL
BOOLEAN	Hat zwei Werte: TRUE oder FALSE	TRUE
INT	64-Bit-Ganzzahl (positiv oder negativ)	123
FLOAT	Doppeltgenaue (64-Bit-)Gleitkommazahlen	-123.456
STRING	Stellt unveränderliche Byte-Folgen dar	"Hello, ctrlX"

9.3 Variablen (global)

Variablen können Werte speichern.

In der aktuellen Version sind nur globale Variablen verfügbar.

- Definition der globalen Variablen:
 - Globale Variablen müssen nicht explizit mit einem Schlüsselwort deklariert werden.
 - Globale Variablen werden bei der ersten Verwendung erstellt.
- Umfang der globalen Variablen:
 - Sobald eine globale Variable einen Wert enthält, kann sie in allen Skriptbereichen (Hauptprogramm und Unterprogramm) aufgerufen werden.

9.4 Wertzuweisung

Globale Variablen können mit Werten verknüpft werden.

Dies wird durch die Verwendung des "="-Zeichens (Gleichheitszeichen) erreicht.

Beispiel

Var_1 = 100	Zuweisung eines Int-Wertes zur globalen Variablen "Var_1"
Var_1 = 2029.876	Zuweisung eines Float-Wertes zur globalen Variablen "Var_1"
str = "Hello, ctrlX"	Zuweisung eines String-Wertes zur globalen Variablen "str"
N100 Var_2 = Var_1 + 3	Verwenden Sie den Wert der globalen Variable "Var_1" und weisen Sie den Ausdruck der globalen Variable "Var_2" zu

9.5 Operatoren

9.5.1 Mathematische Operatoren

Arithmetische Funktionen, die sich auf Variablen, Konstanten oder Ausdrücke höherer Ebene auswirken, können immer noch aufgerufen werden.

In der Regel gilt "Multiplikation und Division haben Vorrang vor Addition und Subtraktion", d.h. Multiplikation und Division werden zuerst ausgeführt, gefolgt von Addition und Subtraktion.

Für weitere Informationen siehe [Kapitel 9.5.4 Priorität auf Seite 33](#).

G-Code unterstützt folgende arithmetische Operatoren:

- **+**: Addition
- **-**: Subtraktion
- *****: Multiplikation
- **/**: Float-Division
- **%**: Modulo
- **-**: einfaches Minus

Das Ergebnis der arithmetischen Operatoren funktioniert wie folgt:

- Wenn beide Operanden Ganzzahlen sind, wird die Operation über Ganzzahlen durchgeführt und das Ergebnis ist eine Ganzzahl. (neben der Division von Fließkommazahlen).
- Handelt es sich bei einem der Operanden um eine Fließkommazahl, wird die Operation nach den Regeln der Fließkommaarithmetik durchgeführt und das Ergebnis ist eine Fließkommazahl.

Beispiel

Ausdruck	Ergebnis	Ergebnis-Datentyp
1+1-3	-1	INT
1.0 + 1	2.0	FLOAT
5/2	2.5	FLOAT
5/1	5.0	FLOAT

9.5.2 Relative Operatoren

G-Code unterstützt folgende relative Operatoren:

- **==**: Gleichheit
- **!=**: Ungleichheit
- **<**: kleiner als
- **>**: größer als
- **<=**: kleiner oder gleich
- **>=**: größer oder gleich

Diese Operatoren ergeben immer **TRUE** oder **FALSE**.

Beispiel

Ausdruck	Ergebnis	Ergebnis-Datentyp
1<2	TRUE	BOOLEAN
1 > =2	FALSE	BOOLEAN
a != (b-1)		BOOLEAN

9.5.3 Logische Operatoren

Die logischen Operatoren in G-Code sind:

- AND
- OR
- NOT

Alle logischen Operatoren betrachten **NIL** und **FALSE** sowie **0** als falsch und alles andere als wahr.

Das Ergebnis der logischen Operatoren:

- Der Negationsoperator **NOT** gibt immer **FALSE** oder **TRUE** zurück.
- Der Konjunktionsoperator **AND** gibt sein erstes Argument zurück, wenn der Wert logisch falsch ist (FALSE, NIL, 0), andernfalls gibt **AND** sein zweites Argument zurück.
- Der Disjunktionsoperator **OR** gibt sein erstes Argument zurück, wenn dieser Wert logisch wahr ist (im Unterschied zu FALSE, NIL, 0), andernfalls gibt **OR** sein zweites Argument zurück.

Operator	Logischer Ausdruck	Beschreibung	Beispiel
AND	a AND b	Wenn "a" falsch ist, wird "a" oder "b" zurückgegeben	(10 AND 20) ergibt 20 (FALSE AND 10) ergibt FALSE
OR	a OR b	Wenn "a" wahr ist, wird "a" oder "b" zurückgegeben	(10 OR 20) ergibt 10 (FALSE OR 20) ergibt 20
NOT	NOT a	Wenn "a" wahr ist, wird FALSE zurückgegeben Wenn "a" falsch ist, wird TRUE zurückgegeben	NOT (10 AND 20) ergibt FALSE

Beispiel

Ausdruck	Ergebnis	Bemerkung
10 AND 20	10	
NIL AND 10	NIL	
FALSE AND error()	FALSE	error() ist der Aufruf einer Fehlerfunktion
FALSE AND NIL	FALSE	
FALSE OR NIL	NIL	
10 OR 20	10	
10 OR error()	10	error() ist der Aufruf einer Fehlerfunktion
NIL OR "str"	"str"	

9.5.4 Priorität

Die Priorität der Operatoren in ctrlX G-Code folgt der nachfolgenden Tabelle, von höherer zu niedrigerer Priorität:

	Gleiche Priorität					
hoch	NOT	-(unär)				
↓	*	/	%			
↓	+	-				
↓	<	>	<=	>=	!=	==
niedrig	AND					
	OR					

Durch die Verwendung von Klammern in einem Ausdruck kann die Reihenfolge der Operationen explizit angegeben werden.

Beispiel

Ausdruck	Priorität mit Klammern
$b+3 < 3+1*2$	$(b+3) < (3+1*2)$
$2 \text{ AND } 3 \text{ AND } 3+1*2 \text{ OR } b \text{ AND } d/2$	$((2 \text{ AND } 3) \text{ AND } 3+1*2) \text{ OR } (b \text{ AND } (d/2))$
$1 < 2 \text{ AND } 3 \leq 4$	$(1 < 2) \text{ AND } (3 \leq 4)$
$2 == 2 \text{ AND } 3 \geq 4 \text{ OR } 5*2 - 1$	$((2 == 2) \text{ AND } (3 \geq 4)) \text{ OR } (5*2 - 1)$
$\text{NOT } 2 \text{ AND NIL OR } 4 \text{ AND } 4/1 \text{ AND } 3$	$((\text{NOT } 2) \text{ AND NIL}) \text{ OR } (4 \text{ AND } 4/1 \text{ AND } 3)$

9.5.5 Zugriff auf Data Layer-Knoten

ctrlX G-Code kann Data Layer-Knoten lesen oder schreiben, um mit anderen Systemen zu interagieren (z.B. SPS-Variablen).

Ein Data Layer-Knoten wird durch eine Zeichenkette dargestellt (z. B. *plc/app/Application/sym/Machine/Status*).

Im ctrlX G-Code-Skript kann der Knoten als "DL.plc.app.Application.sym.Machine.Status" dargestellt werden.

Syntax des Data Layer-Knotens

- Vorangestelltes Schlüsselwort "DL."
- Gefolgt von einem oder mehreren Zeichen, die mit "." beginnen und eine Kombination aus Kleinbuchstaben, Großbuchstaben, Ziffern und Unterstrichen enthalten.
- Zeichen "/" im Data Layer-Pfad muss durch "." ersetzt werden

Beispiel

- Lesen des Knotenwerts und Zuweisen zu einer Variablen:
 - `varFoo = DL.plc.app.Application.sym.Machine.Status`
- Schreiben des Knotenwerts:
 - `DL.plc.app.Application.sym.Machine.Status = 10 + 1`

Konvertierung des Datentyps zwischen G-Code und Data Layer-System

Der Data Layer-Knoten hat mehr Datentypen als G-Code. Die Konvertierung des Datentyps (mit Bereichsprüfung) wird automatisch beendet, wenn der Data Layer-Knoten gelesen oder geschrieben wird.

Folgende Tabelle zeigt die Zuordnung zwischen Data Layer und G-Code:

Datentyp Data Layer	Datentyp Data Layer (detailliert)	Wert in G-Code	Beispiel für G-Code-Werte
Einfache Datentypen	BOOL8	BOOLEAN	TRUE FALSE
	INT8 UINT8 INT16 UINT16 INT32 UINT32 INT64 UINT64	INT: 64-Bit-Ganzzahl	123
	FLOAT32 FLOAT64	FLOAT: Doppelgenaue (64-Bit-)Gleitkommazahlen	123.456



Datentyp Data Layer	Datentyp Data Layer (detailliert)	Wert in G-Code	Beispiel für G-Code-Werte
	STRING	STRING	"rexroth"
Array-Typen	-	-	-
Komplexe Datentypen (Flatbuffer)	-	-	-

"-" bedeutet nicht unterstützte Konvertierung. Es wird ein Fehler gemeldet.

Einschränkungen

- ctrlX G-Code akzeptiert nur Data Layer-Pfade, die in ASCII kodiert sind, d.h. wenn eine Zeichenkette im UTF-8-Format in einem Data Layer-Knoten verwendet wird, wird ein Syntaxfehler gemeldet.
- Wenn der Achsname zum Beispiel in deutschen oder chinesischen Zeichen geschrieben ist, wird er nicht unterstützt.
- Es ist nicht möglich, Data Layer-Knoten mit Array- oder Flatbuffer-Typ zu lesen oder zu schreiben.

10 Anhang

10.1 Übersicht der NC-Funktionen

10.1.1 NC-Funktionen mit Syntax nach DIN 66025

Name der NC-Funktion			Gruppe	Beschreibung
Name	Kurzform	Alias		
↪ G0	-	G00 G000	Interpolation	Geradeninterpolation (Eilgang)
↪ G1	-	G01 G001	Interpolation	Geradeninterpolation (Vorschub)
↪ G2	-	G02 G002	Geometrie	Bewegung auf einer Kreisbahn im Uhrzeigersinn
↪ G3	-	G03 G003	Geometrie	Bewegung auf einer Kreisbahn im Gegenuhrzeigersinn
↪ G8	-	-	Bahnslope	Bahnslope EIN Basierend auf Motion-Befehl: <i>opt-cont-motion</i> mit Nutzdaten { "permType": "PermOn" }
↪ G9	-	-	Bahnslope	Bahnslope AUS Basierend auf Motion-Befehl: <i>opt-cont-motion</i> mit Nutzdaten { "permType": "PermOff" }
↪ G16	-	-	Aktive Ebene	Aktive Ebene aus
↪ G17 (G17-19)	-	-	Aktive Ebene	Aktive Ebene auswählen XY, ZX oder YZ

Name der NC-Funktion			Gruppe	Beschreibung
Name	Kurzform	Alias		
➡ G47	-	-	Werkzeuglängenkorrektur	Werkzeuglängenkorrektur einschalten Basierend auf Motion-Befehl: <i>opt-pcs-tool</i> mit Nutzdaten {"permType": "PermOn"}
➡ G48	-	-	Werkzeuglängenkorrektur	Werkzeuglängenkorrektur ausschalten Basierend auf Motion-Befehl: <i>opt-pcs-tool</i> mit Nutzdaten {"permType": "PermOff"}
➡ G53	-	-	Nullpunktverschiebung	Alle Nullpunktverschiebungen AUS
➡ G54	-	-	Nullpunktverschiebung	Fahrbefehlsoption PCS-Set (Gruppe) G54 EIN
➡ G55	-	-	Nullpunktverschiebung	Fahrbefehlsoption PCS-Set (Gruppe) G55 EIN
➡ G56	-	-	Nullpunktverschiebung	Fahrbefehlsoption PCS-Set (Gruppe) G56 EIN
➡ G57	-	-	Nullpunktverschiebung	Fahrbefehlsoption PCS-Set (Gruppe) G57 EIN
➡ G58	-	-	Nullpunktverschiebung	Fahrbefehlsoption PCS-Set (Gruppe) G58 EIN
➡ G59	-	-	Nullpunktverschiebung	Fahrbefehlsoption PCS-Set (Gruppe) G59 EIN
➡ G90	-	-	Programmiermodus	Absolutmaß-Programmierung
➡ G91	-	-	Programmiermodus	Relativmaß-Programmierung
➡ G94	-	-	Vorschubmodus	Programmierung (mm/min)
➡ G152				G152(SET = <setName>) Aktivieren des Produktkoordinatensystems mit Satz(gruppen-)namen
➡ G153				Produktkoordinatensystem ausschalten

10.1.2 NC-Funktionen mit Hochsprachen-Syntax

Name der NC-Funktion			Gruppe	Beschreibung
Name	Kurzform	Alias		
➔ PathDynLim	PDL	PATHDYNLIM	-	<p>Programmierung der dynamischen Pfadgrenzen:</p> <p>Einschalten:</p> <pre>PDL ({ACC=<number>, } {DEC=<number> , } {JRKACC=<number>, } {JRKDEC=<number>})</pre> <p>Ausschalten:</p> <pre>PDL ()</pre> <p>Programmierung der dynamischen Pfadgrenzen, entweder getrennt für <i>ACC/DEC/JRKACC/JRKDEC</i> oder kombiniert.</p>
➔ WAIT	-	-	-	Vorbereitung des Skripts und des Motion-Befehlspuffers stoppen
➔ Contour	-	CONTOUR	-	<p>Start der Kontur:</p> <pre>Contour (PREPCMDS = <INT>)</pre> <p>Ende der Kontur:</p> <pre>Contour ()</pre> <p>Basierend auf dem Motion-Befehl: specify-contour-area</p>
➔ PolyTrans	-	POLYTRANS	-	<p>Dauerhaft einschalten:</p> <pre>PTR</pre> <p>Ausschalten:</p> <pre>PolyTrans (D1=<num>, D2=<num>) PolyTrans (EPS=<num>)</pre> <p>Basierend auf der Motion-Befehls- option: path-rounding-with-polynomial</p>
➔ SetSignal	SSG	SETSIGNAL	-	<p>Auslösen eines globalen Signals:</p> <pre>SSG (ID=1)</pre> <p>SetSignal(signalId = <INT>)</p> <p>Signalzustand:</p> <pre>motion/state/functions/somo/signals/[sigId]</pre>
➔ WaitForSignal	WSG	WAITFORSIGNAL	-	<p>Warten auf ein globales Signal:</p> <pre>WSG (ID=1)</pre>
➔ ResetSignal	RSG	RESETSIGNAL	-	<pre>RSG (ID=1)</pre>

Name der NC-Funktion			Gruppe	Beschreibung
Name	Kurzform	Alias		
→ AxsDynLim	ADL	AXSDYNLIM	-	Verkleinern der dynamischen Grenzen der kinematischen Achse <ul style="list-style-type: none"> Aktivieren: AxsDynLim(AXS="X", VEL=10, ACC=2, DEC=2, JRKACC=17, JRKDEC=17) Deaktivieren: AxsDynLim(AXS="X")

11 Weiterführende Dokumentationen

11.1 Übersicht

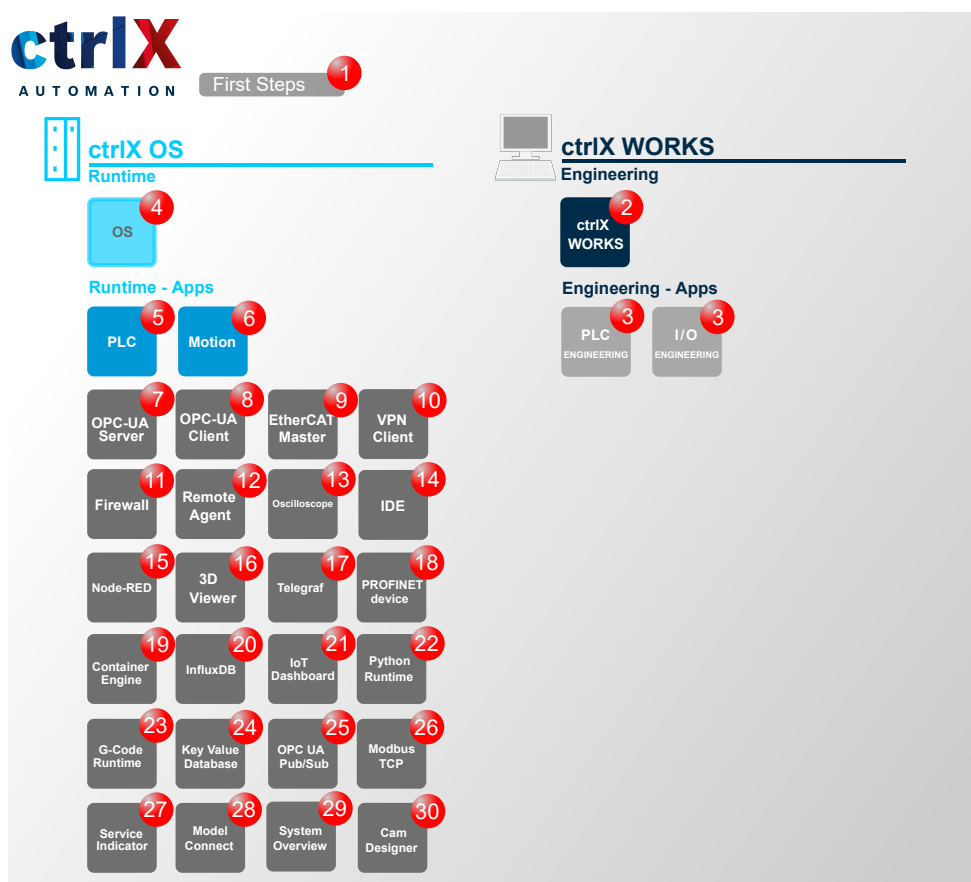


Abb. 7: Übersicht der weiterführenden Dokumentationen

11.2 ctrlX AUTOMATION

Nr.	Dokumentation
1	ctrlX WORKS - Erste Schritte 02VRS Quick Start Guide → Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> DOK-XWORKS-F*STEP**V02-QURS-DE-P R911421573

11.3 ctrlX WORKS

Nr.	Dokumentation
2	ctrlX WORKS - Basissystem 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XWORKS-WRK***V02**-APRS-DE-P• R911421575
3	ctrlX PLC Engineering - SPS-Programmiersystem 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XPLC**-ENG*****V02-APRS-DE-P• R911421577
3	ctrlX PLC Engineering - SPS-Bibliotheken 02VRS Referenz ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XPLC**-LIB***V02**-RERS-DE-P• R911421579

11.4 ctrlX OS

Nr.	Dokumentation
4	ctrlX OS - Runtime 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-XCR***V02**-APRS-DE-P• R911421589
	ctrlX OS - Knoten des Data Layer 02VRS Referenz ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-DL****V02**-RERS-DE-P• R911421591
	ctrlX OS - Diagnosen 02VRS Referenz ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-DIAG**V02**-RERS-DE-P• R911421593

11.5 ctrlX OS Apps

Nr.	Dokumentation
5	PLC App - SPS-Laufzeitumgebung für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-PLC***V02**-APRS-DE-P • R911421585
6	Motion App - Motion-Laufzeitumgebung für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-MOT***V02**-APRS-DE-P • R911421609
7	OPC UA Server App - OPC UA Server für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-UAS***V02**-APRS-DE-P • R911421597
8	OPC UA Client App - OPC UA Client für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-UAC***V02**-APRS-DE-P • R911421599
9	EtherCAT Master App - EtherCAT Master für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-ECM***V02**-APRS-DE-P • R911421603
10	VPN Client App - Fernwartungssoftware für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-VPN***V02**-APRS-DE-P • R911421595
11	Firewall App - Security Funktionen für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-FRW***V02**-APRS-DE-P • R911421605

Nr.	Dokumentation
12	Remote Agent App - ctrlX Device Portal-Anbindung für ctrlX OS Geräte 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-RMA***V02**-APRS-DE-P• R911421607
13	Oscilloscope App - Oszilloskopfunktion für ctrlX OS Geräte 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-OSC***V02**-APRS-DE-P• R911421588
14	IDE App - Integrated Development Environment 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-IDE***V02**-APRS-DE-P• R911421611
15	Node-RED App - Grafische Programmierung für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-NODERED*V02-APRS-DE-P• R911421583
16	3D Viewer App - Browserbasierte 3D-Kinematik-Simulation für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-3DV***V02**-APRS-DE-P• R911421614
17	Telegraf App - Server-Agent zum Sammeln von Daten im Data Layer 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-TSA***V02**-APRS-DE-P• R911421622
18	PROFINET Device App - PROFINET Device für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-PROFINETV02-APRS-DE-P• R911421616

Nr.	Dokumentation
19	Container Engine App - Verwendung von Docker® Images auf ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-DOE***V02**-APRS-DE-P • R911421618
20	InfluxDB App - Influx-Datenbankanbindung für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-IDB***V02**-APRS-DE-P • R911421624
21	IoT Dashboard App - Datenvisualisierung in dynamischen, interaktiven Dashboards 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-GDB***V02**-APRS-DE-P • R911421632
22	Python Runtime App - Python-Laufzeitumgebung für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-PYR***V02**-APRS-DE-P • R911421628
23	G-Code Runtime App - G-Code Interpreter für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-GCO***V02**-APRS-DE-P • R911421630
24	Key Value Database App - Verwaltung von Daten im Data Layer 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-KVD*****V02-APRS-DE-P • R911421634
25	OPC UA Pub/Sub App - OPC UA Pub/Sub für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none"> • DOK-XCORE*-UAP***V02**-APRS-DE-P • R911421601

Nr.	Dokumentation
26	Modbus TCP App - Modbus TCP-Kommunikation für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-MOD*TCP*V02-APRS-DE-P• R911421620
27	Service Indicator App -Service Indicator für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-SIN*****V02-APRS-DE-P• R911421626
28	Model Connect App - Target für modellbasierte Entwicklung und Simulation für ctrlX OS 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-MOC***V02**-APRS-DE-P• R911421630
29	System Overview App - Systemtopologie und Systeminformationen 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XCORE*-SOV***V02**-APRS-DE-P• R911424408
30	Cam Designer - Kurvenscheiben der ctrlX MOTION konfigurieren 02VRS Anwendungsbeschreibung ↗ Link zur Web-Dokumentation Bestellinformationen: <ul style="list-style-type: none">• DOK-XWORKS-CAM***V02**-AP01-DE-P• R911424389

12 Service und Support

Für Ihre schnelle und optimale Unterstützung verfügen wir über ein dichtes weltweites Servicenetz. Unsere Experten stehen Ihnen mit Rat und Tat zur Seite. Sie erreichen uns täglich **rund um die Uhr – auch an Wochenenden und Feiertagen**.

Service Deutschland

Unser technologieorientiertes Competence Center in Lohr deckt alle Belange rund um den Service für elektrische Antriebe und Steuerungen ab.

Sie erreichen unsere **Service-Hotline** und unseren **Service-Helpdesk** unter:

Telefon: **+49 9352 40 5060**

Fax: **+49 9352 18 4941**

E-Mail: ➔ service.svc@boschrexroth.de

Internet: ➔ <http://www.boschrexroth.com>

Auf unseren Internetseiten finden Sie ergänzende Hinweise zu Service, Reparatur (z. B. Anlieferadressen) und Training.

Service weltweit

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit Ihrem Ansprechpartner auf. Die Hotline-Rufnummern entnehmen Sie bitte den Vertriebsadressen im Internet.

Vorbereitung der Informationen

Wir können Ihnen schnell und effizient helfen, wenn Sie folgende Informationen bereithalten:

- Eine detaillierte Beschreibung der Störung und der Umstände
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern
- Ihre Kontaktdaten (Telefon-, Faxnummer und E-Mail-Adresse)

13 Index

B

Bestimmungsgemäßer Gebrauch

Anwendungsbereiche.	6
Einleitung.	5
Einsatzfälle.	6

C

ctrlX AUTOMATION

Weiterführende Dokumentationen.	38
--------------------------------------	----

E

Einführung. 7

Erweiterte Programmierung

Logische Operatoren.	32
Mathematische Operatoren.	32
Operatoren.	32
Priorität.	33
Relative Operatoren.	32
Typ und Werte.	31
Variablen (global).	31
Wertzuweisung.	31
Zugriff auf Data Layer-Knoten.	34

Erweiterte Programmierung mit High-Level-Syntax. 30

G

G-Code Runtime

Voraussetzungen.	7
-----------------------	---

G-Code Runtime

Einführung.	7
------------------	---

G-Codes

16.	20
17.	20
18.	20
19.	20
Absolutmaß-Programmierung.	22
Bahnslope.	20
Ebenenumschaltung.	20
G0.	17
G1.	18
G2.	18
G3.	18
G8.	20
G9.	20
G47.	21
G48.	21
G53.	21
G54-G59.	21
G90.	22
G91.	22
G94.	22
G152.	23
G153.	23
Geradeninterpolation im Eilgang.	17
Geradeninterpolation im Vorschub.	18
Kreisinterpolation im Rechts-/Linkslauf. . .	18
Nullpunktverschiebungstabelle auswählen	21

Relativmaß-Programmierung.	22
Vorschubprogrammierung.	22
Werkzeuflängenkorrektur.	21, 23
Globale Signale.	24
AxsDynLim.	29
Contour.	26, 27
M-Code.	25
PathDynLim.	28
PolyTrans.	27
ResetSignal.	24
SetSignal.	24
WaitForSignal.	25

H

Helpdesk. 44

Hotline. 44

N

NC-Funktionen. 17, 24

NC-Funktionen mit Hochsprachen-Syntax

Übersicht.	37
-----------------	----

NC-Funktionen mit höherer

Programmiersprachen-Syntax. 24

NC-Funktionen mit Syntax nach DIN 66025. . 17

Übersicht.	35
-----------------	----

NC-Programm

Ausführung.	8
Komponenten.	10
Programmwörter.	12
Speichern.	9
Status.	9

NC-Programmierung. 8

Grundlagen.	8
------------------	---

Nicht bestimmungsgemäßer Gebrauch. 7

Folgen, Haftungsausschluss.	6
----------------------------------	---

P

Programmwörter. 12

Auswirkung.	14
Modal.	14
Nicht-modal.	15

S

Service-Hotline. 44

Sicherheitshinweise. 7

Support. 44

U

Unterprogramm

Aufruf.	16
Suchpfad.	17

V

Voraussetzungen. 7

Bosch Rexroth AG
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr a.Main
Germany
Tel. +49 9352 18 0
Fax +49 9352 18 8400
www.boschrexroth.com/electrics



R911421630