

MELFA

Industrieroboter

Bedienungs- und
Programmieranleitung

**Steuergeräte
CR1/CR2**

Bedienungs- und Programmieranleitung
Steuergeräte CR1/CR2
Artikel-Nr.: 140015-A

Version			Änderungen / Ergänzungen / Korrekturen
A	08/2001	pdp	—

Zu diesem Handbuch

Die in diesem Handbuch vorliegenden Texte, Abbildungen, Diagramme und Beispiele dienen ausschließlich der Erläuterung zur Installation, Bedienung und zum Betrieb der in diesem Handbuch beschriebenen Steuergeräte.

Sollten sich Fragen bezüglich Installation und Betrieb der in diesem Handbuch beschriebenen Geräte ergeben, zögern Sie nicht, Ihr zuständiges Verkaufsbüro oder einen Ihrer Vertriebspartner (siehe Umschlagseite) zu kontaktieren.

Aktuelle Informationen sowie Antworten auf häufig gestellte Fragen erhalten Sie über das Fax-Abrufsystem MEL-FAX (Fax-Abruf: (0 21 02) 486 485 oder (0 21 02) 486 790) oder über das Internet: <http://www.mitsubishi-automation.de>.

Die MITSUBISHI ELECTRIC EUROPE B.V. behält sich vor, jederzeit technische Änderungen dieses Handbuchs ohne besondere Hinweise vorzunehmen.

© 08/2001

Sicherheitshinweise

Zielgruppe

Dieses Handbuch richtet sich ausschließlich an anerkannt ausgebildete Elektrofachkräfte, die mit den Sicherheitsstandards der Automatisierungstechnik vertraut sind. Projektierung, Installation, Inbetriebnahme, Wartung und Prüfung der Roboter nebst Zubehör dürfen nur von einer anerkannt ausgebildeten Elektrofachkraft, die mit den Sicherheitsstandards der Automatisierungstechnik vertraut ist, durchgeführt werden.

Bestimmungsgemäßer Gebrauch

Die Steuergeräte CR1 und CR2 sind nur für die Einsatzbereiche vorgesehen, die in diesem Handbuch beschrieben sind. Achten Sie auf die Einhaltung aller im Handbuch angegebenen Kenndaten.

Jede andere darüber hinausgehende Verwendung oder Benutzung gilt als nicht bestimmungsgemäß.

Sicherheitsrelevante Vorschriften

Bei der Projektierung, Installation, Inbetriebnahme, Wartung und Prüfung der Geräte müssen die für den spezifischen Einsatzfall gültigen Sicherheits- und Unfallverhütungsvorschriften beachtet werden.



ACHTUNG:

Im Lieferumfang des Roboters ist ein Sicherheitstechnisches Handbuch enthalten. Dieses Handbuch behandelt alle sicherheitsrelevanten Details zu Aufstellung, Inbetriebnahme und Wartung. Vor einer Aufstellung, Inbetriebnahme oder der Durchführung anderer Arbeiten mit oder am Roboter ist dieses Handbuch unbedingt durchzuarbeiten. Alle darin aufgeführten Angaben sind zwingend zu beachten! Sollte dieses Handbuch nicht im Lieferumfang enthalten sein, wenden Sie sich bitte umgehend an Ihren Mitsubishi-Vertriebspartner.

Darüber hinaus müssen folgende Vorschriften (ohne Anspruch auf Vollständigkeit) beachtet werden:

- VDE-Vorschriften
- Brandverhütungsvorschriften
- Unfallverhütungsvorschriften

Erläuterung zu den Gefahrenhinweisen

In diesem Handbuch befinden sich Hinweise, die wichtig für den sachgerechten sicheren Umgang mit dem Roboter sind.

Die einzelnen Hinweise haben folgende Bedeutung:



GEFAHR:

Bedeutet, dass eine Gefahr für das Leben und die Gesundheit des Anwenders, z. B. durch elektrische Spannung, besteht, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



ACHTUNG:

Bedeutet eine Warnung vor möglichen Beschädigungen des Roboters, seiner Peripherie oder anderen Sachwerten, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Inhaltsverzeichnis

1	Einführung	
1.1	Grundlegende Sicherheitshinweise	1-1
1.2	Die ersten Schritte	1-3
2	Funktionen	
2.1	Steuergerät	2-1
2.1.1	Bedien- und Signalelemente des Steuergerätes	2-1
2.2	Teaching Box	2-3
2.2.1	Display-Anzeigen und Funktionen	2-3
2.2.2	Bedienelemente der Teaching Box	2-5
2.3	Betriebsrechte	2-8
3	MELFA-BASIC-IV-Programmierung	
3.1	Programmaufbau	3-1
3.1.1	Anweisung	3-1
3.1.2	Variable	3-2
3.2	Steuerung der Roboterbewegung	3-4
3.2.1	Gelenk-Interpolation	3-4
3.2.2	Linear-Interpolation	3-6
3.2.3	Kreis-Interpolation	3-9
3.2.4	Kontinuierliche Bewegung	3-11
3.2.5	Beschleunigungs-/Bremszeit und Geschwindigkeit	3-13
3.2.6	Feinpositionierung	3-16
3.2.7	Hand- und Werkzeugsteuerung	3-18
3.3	Palettierung	3-20
3.4	Programmsteuerung	3-24
3.4.1	Verzweigungen und Wartezeit	3-24
3.4.2	Programmschleife	3-26
3.4.3	Interrupt	3-27
3.4.4	Unterprogramm	3-28
3.4.5	Timer	3-29
3.4.6	Stopp	3-30

3.5	Ein- und Ausgabe externer Signale	3-31
3.5.1	Eingangssignale	3-31
3.5.2	Ausgangssignale	3-32
3.6	Kommunikation	3-33
3.7	Ausdrücke und Operationen	3-35
3.8	Angehängte Anweisung	3-37
4	Multitask-Funktion	
4.1	Beschreibung	4-1
4.2	Betriebszustand eines Programmplatzes	4-2
4.3	Ausführung eines Multitasks	4-3
4.3.1	Programmplatzparameter	4-3
4.4	Erstellung eines Multitask-Programms	4-5
4.5	Anwendung des Multitaskings	4-7
4.6	Beispiel zur Anwendung der Multitask-Funktion	4-8
5	Parameter	
5.1	Allgemeines	5-1
5.1.1	Übersicht der Parameter	5-1
6	Externe Ein-/Ausgänge	
6.1	Einteilung	6-1
6.2	Parallele Ein-/Ausgangsschnittstelle	6-1
6.2.1	Ein-/Ausgangsbelegung der parallelen Ein-/Ausgangsschnittstelle	6-6
6.2.2	Programmsteuerung durch externe Signale	6-13
6.3	NOT-HALT-Eingang	6-17
6.3.1	Steuergerät CR1	6-17
6.3.2	Steuergerät CR2	6-18
7	Bedienung und Programmierung	
7.1	Bedienung der Teaching Box	7-1
7.1.1	Menübaum	7-1
7.1.2	Menüpunkt auswählen	7-2
7.2	Programmierung	7-4
7.2.1	Roboterprogramm erstellen	7-4
7.2.2	Roboterprogramm editieren	7-7
7.2.3	Roboterprogramm testen	7-15

7.3	Servospannung ein-/ausschalten	7-17
7.4	Automatikbetrieb	7-19
7.4.1	Geschwindigkeit einstellen	7-19
7.4.2	Auswahl der Programmnummer	7-20
7.4.3	Starten des Automatikbetriebs	7-21
7.4.4	Stoppen des Automatikbetriebs	7-23
7.4.5	Fortsetzung des Automatikbetriebs aus dem Stoppzustand	7-23
7.4.6	Programm zurücksetzen	7-23
7.5	Programmverwaltungsfunktionen	7-25
7.5.1	Programmverzeichnis anzeigen	7-25
7.5.2	Programm schützen	7-26
7.5.3	Programm kopieren	7-28
7.5.4	Programmnamen ändern	7-29
7.5.5	Programm löschen	7-30
7.6	Monitor-Funktionen	7-31
7.6.1	Monitor-Funktion für Eingangssignale	7-31
7.6.2	Monitor-Funktion für Ausgangssignale	7-32
7.6.3	Monitor-Funktion für Variable	7-33
7.6.4	Liste der aufgetretenen Fehlermeldungen	7-34
7.7	Zusatzfunktionen	7-35
7.7.1	Parameter einstellen	7-35
7.7.2	Alle gespeicherten Programme löschen	7-36
7.7.3	Batteriezüehler zurücksetzen	7-37
7.7.4	Gelenkbremsen lösen	7-38
7.7.5	Batterie und Einschaltzeit anzeigen	7-39
7.7.6	Uhrzeit und Datum einstellen	7-40

8 MELFA-BASIC IV

8.1 Begriffserklärung 8-1

8.1.1 Anweisung 8-1

8.1.2 Angehängte Anweisung 8-1

8.1.3 Zeilen 8-2

8.1.4 Zeilennummern und Marken 8-2

8.1.5 Zeichentypen 8-3

8.1.6 Zeichen mit besonderer Bedeutung 8-4

8.1.7 Datentypen 8-5

8.1.8 Konstanten 8-6

8.1.9 Variablen 8-10

8.1.10 Feldvariablen 8-14

8.1.11 Externe Variablen 8-15

8.1.12 Logische Werte 8-19

8.1.13 Funktionen 8-20

8.1.14 Konvertierte Datentypen 8-23

8.1.15 Rangfolge von Operationen 8-25

8.1.16 Programmebenen 8-25

8.1.17 Reservierte Wörter 8-25

9 MELFA-BASIC-IV-Befehle

9.1 Allgemeine Hinweise 9-1

9.1.1 Beschreibung des verwendeten Formats 9-1

9.2 Übersicht der MELFA-BASIC-IV-Befehle 9-2

9.3 Detaillierte Befehlsbeschreibung 9-4

9.3.1 ACCEL (Accelerate) 9-4

9.3.2 ACT (Act) 9-5

9.3.3 BASE (Base) 9-7

9.3.4 CALLP (Call P) 9-8

9.3.5 CLOSE (Close) 9-9

9.3.6 CLR (Clear) 9-10

9.3.7 CMP POS (Compliance Posture) 9-11

9.3.8 CMP TOOL (Compliance Tool) 9-12

9.3.9 CMP OFF (Compliance OFF) 9-13

9.3.10 CMPG (Compliance Gain) 9-14

9.3.11 CNT (Control) 9-15

9.3.12 COM OFF (Communication OFF) 9-17

9.3.13 COM ON (Communication ON) 9-18

9.3.14 COM STOP (Communication STOP) 9-19

9.3.15 DEF ACT (Define act) 9-20

9.3.16 DEF FN (Define function) 9-22

9.3.17	DEF PLT (Define pallet)	9-23
9.3.18	DEF INTE/FLOAT/DOUBLE (Define Integer/Float/Double)	9-25
9.3.19	DEF IO (Define IO)	9-26
9.3.20	DEF JNT (Define Joint)	9-28
9.3.21	DEF POS (Define Position)	9-29
9.3.22	DEF CHAR (Define Character)	9-30
9.3.23	DIM (Dim)	9-31
9.3.24	DLY (Delay)	9-32
9.3.25	ERROR (Error)	9-33
9.3.26	END (End)	9-34
9.3.27	FINE (Fine)	9-35
9.3.28	FOR-NEXT (For-Next)	9-36
9.3.29	FPRM (FPRM)	9-38
9.3.30	GETM (Get Mechanism)	9-39
9.3.31	GOSUB (Go Subroutine)	9-40
9.3.32	GOTO (Go To)	9-41
9.3.33	HLT (Halt)	9-42
9.3.34	HOPEN/HCLOSE (Hand Open/Hand Close)	9-43
9.3.35	IF ... THEN ... ELSE (If Then Else)	9-44
9.3.36	INPUT # (Input)	9-45
9.3.37	JOVRD (J Override)	9-46
9.3.38	JRC (Joint Roll Change)	9-47
9.3.39	LABEL (Label)	9-49
9.3.40	LOADSET (Load Set)	9-50
9.3.41	MOV (Move)	9-52
9.3.42	MVC (Move C)	9-53
9.3.43	MVR (Move R)	9-54
9.3.44	MVR2 (Move R2)	9-56
9.3.45	MVR3 (Move R3)	9-58
9.3.46	MVS (Move S)	9-60
9.3.47	OADL (Optimum Acceleration/Deceleration)	9-62
9.3.48	ON COM GOSUB (ON Communication Go Subroutine)	9-63
9.3.49	ON GOSUB (ON GOSUB)	9-64
9.3.50	ON ... GOTO (On Go To)	9-65
9.3.51	OPEN (Open)	9-66
9.3.52	OVRD (Override)	9-68
9.3.53	PLT (Pallet)	9-69
9.3.54	PRINT (Print)	9-70
9.3.55	RELM (Release Mechanism)	9-72
9.3.56	REM (Remarks)	9-73
9.3.57	RETURN (Return)	9-74
9.3.58	SELECT CASE	9-75

9.3.59	SERVO (Servo)	9-77
9.3.60	SKIP (Skip)	9-78
9.3.61	SPD (Speed)	9-79
9.3.62	TOOL (Tool)	9-80
9.3.63	TORQ (Torque)	9-81
9.3.64	WAIT (Wait)	9-82
9.3.65	WHILE ~ WEND (While End)	9-83
9.3.66	WTH (With)	9-84
9.3.67	WTHIF (With If)	9-85
9.3.68	XLOAD (X Load)	9-86
9.3.69	XRUN (X Run)	9-87
9.3.70	XSTP (X Stop)	9-88
9.3.71	XRST (X Reset)	9-89
9.3.72	SUBSTITUTE (Substitute)	9-90

A Anhang

A.1	Fehlerdiagnose	A-1
A.1.1	Übersicht der Fehlercodes	A-2

1 Einführung

1.1 Grundlegende Sicherheitshinweise

Der MELFA-Roboter ist nach dem neuesten Stand der Technik gebaut und betriebssicher ausgeführt. Ungeachtet dessen können von dem Roboter Gefahren ausgehen, wenn er nicht von geschultem oder zumindest eingewiesenem Personal betrieben wird oder unsachgemäß bzw. zu nicht bestimmungsgemäßem Gebrauch eingesetzt wird.

Dies betrifft insbesondere:

- **Gefahren für Leib und Leben des Benutzers oder Dritter**
- **Beeinträchtigungen des Roboters, anderer Maschinen und weiterer Sachwerte des Anwenders**



ACHTUNG:

Jede Person, die im Betrieb des Anwenders mit der Aufstellung, Inbetriebnahme, Bedienung, Wartung und Reparatur des Roboters beauftragt ist, muss neben der zum Roboter gehörenden Technischen Dokumentation besonders das mitgelieferte

SICHERHEITSTECHNISCHE HANDBUCH

gelesen und verstanden haben.



ACHTUNG:

Achten Sie strikt auf die Einhaltung aller Sicherheitsrichtlinien. Im Rahmen dieser einführenden Sicherheitshinweise werden folgende weitere Instruktionen gegeben:

Der Roboter darf nur von ausgebildetem und autorisiertem Bedienungspersonal betrieben und bedient werden.

Die Zuständigkeiten für die unterschiedlichen Tätigkeiten im Rahmen des Betriebes des Roboters müssen klar festgelegt und eingehalten werden, damit unter dem Aspekt der Sicherheit keine unklaren Kompetenzen auftreten.

Bei allen Arbeiten, die die Aufstellung, die Inbetriebnahme, das Rüsten, den Betrieb, Änderungen der Einsatzbedingungen und Betriebsweisen, Wartung, Inspektion und Reparatur betreffen, sind die in der Betriebsanleitung angegebenen Ausschaltprozeduren zu beachten.

Die Lage der NOT-AUS-Taster muss bekannt sein und die NOT-AUS-Taster müssen jederzeit zugänglich sein.

Es ist jede Arbeitsweise zu unterlassen, die die Sicherheit an der Maschine beeinträchtigt.

Der Bediener hat dafür zu sorgen, dass keine Personen an dem Roboter arbeiten, die nicht dazu autorisiert sind (z. B. auch durch Betätigung von Einrichtungen gegen unbefugtes Benutzen).



Das verwendende Unternehmen hat dafür zu sorgen, dass der Roboter immer nur in einwandfreiem Zustand betrieben wird.

Der Verwenderbetrieb sollte das zuständige Bedienungspersonal besonders schulen und dazu verpflichten, alle Wartungs- und Inspektionsarbeiten ausschließlich bei abgeschaltetem Roboter und ausgeschalteter Peripherie durchzuführen.



GEFAHR:

Das Steuergerät darf ausschließlich über einen Leistungsschalter an die Netzspannung angeschlossen werden. Bei Nichtbeachtung besteht die Gefahr eines elektrischen Schlages.

Eine detaillierte Beschreibung des Netzanschlusses finden Sie im Technischen Handbuch des Roboters.

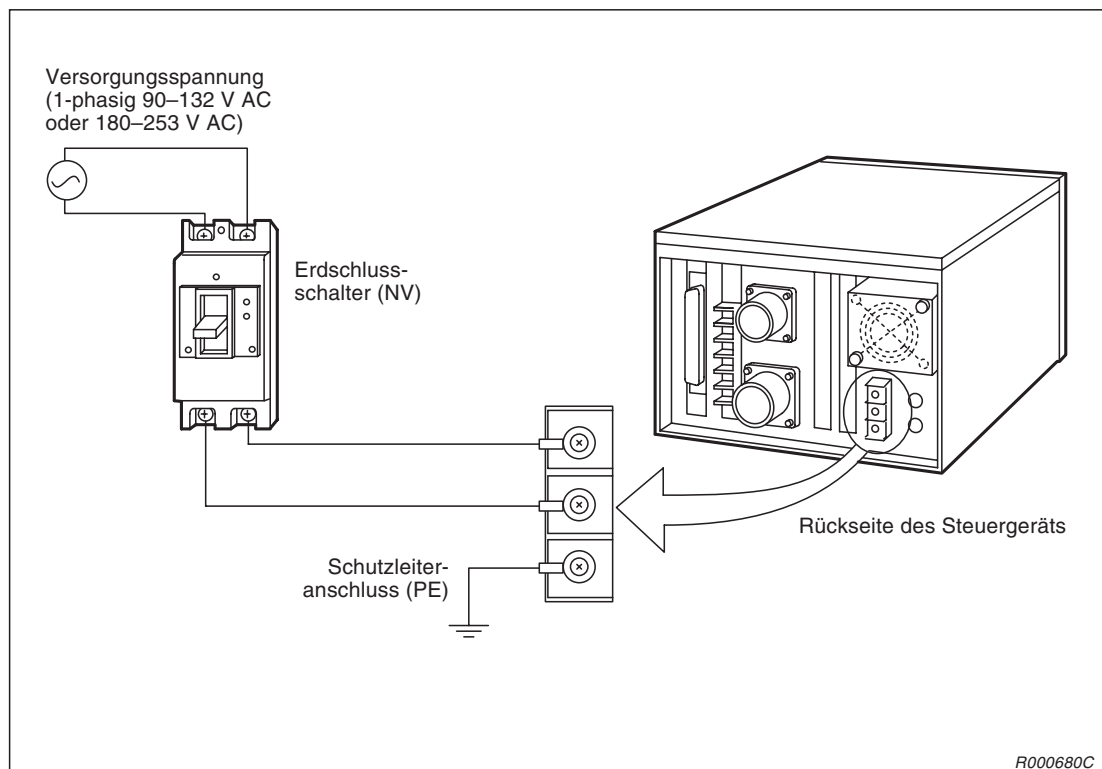


Abb. 1-1: Netzanschluss des CR1-Steuergerätes

1.2 Die ersten Schritte

Nachfolgend erhalten Sie eine Darstellung der ersten Schritte mit Ihrem MELFA-Roboter:

- ① **Roboter und Steuergerät auspacken**
- ② **Sicherheitstechnisches Handbuch lesen**
Vor der ersten Inbetriebnahme des Robotersystems lesen Sie das Sicherheitstechnische Handbuch.
- ③ **Batterien anklemmen**
Klemmen Sie zunächst die Batterien des Roboterarms an. Nähere Einzelheiten dazu entnehmen Sie dem Technischen Handbuch.
- ④ **Kabel anschließen**
Verbinden Sie alle Kabel, wie im Technischen Handbuch beschrieben, und schließen Sie die Teaching Box an.
- ⑤ **Netzspannungsversorgung einschalten**
Schalten Sie die Netzspannungsversorgung für das Steuergeräte über den POWER-Schalter ein.
- ⑥ **Selbsttest des Steuergerätes**
Das Steuergerät startet einen Selbsttest mit einer Dauer von ca. 5 Sekunden.

Sollte nach dem Selbsttest eine Fehlermeldung erscheinen, versuchen Sie den Fehler mit Hilfe der Fehlerbeschreibung im Anhang dieses Handbuches zu beheben.
- ⑦ **Teaching Box einschalten (Dreistufenschalter drücken !!!)**
Drücken Sie den Dreistufenschalter auf der Rückseite der Teaching Box in die Mittelstellung und schalten Sie anschließend die Teaching Box ein (ENBL/DISABLE-Schalter auf ENBL).
 - System einstellen (DATA-Methode)
Zur Abgleichung des Systems muss der Origin-Punkt des Roboterarms eingestellt werden. Die genaue Vorgehensweise entnehmen Sie dem Technischen Handbuch. Anschließend schalten Sie die Netzspannung des Steuergerätes kurzzeitig aus und wieder ein, um eine Übernahme der eingegebenen Werte zu gewährleisten.
- ⑧ **Teach-Modus auswählen**
Wählen Sie den Menüpunkt „1. TEACH“ aus, indem Sie lediglich die vorgegebene Auswahl mit der [INP/EXE]-Taste bestätigen.

Es wird der Teach-Modus für Positionsdaten aufgerufen.
- ⑨ **Programmnummer eingeben**
Sie werden jetzt nach der Programmnummer gefragt, unter der Sie die Positionsdaten definieren möchten.

Geben Sie z. B. „1“ für die Programmnummer 1 ein und betätigen Sie anschließend die [INP/EXE]-Taste. Dann halten Sie die Taste [POS/CHAR] gedrückt und betätigen einmal die Taste [ADD].
- ⑩ **Roboter bewegen**
Betätigen Sie die [STEP/MOVE]-Taste und halten Sie diese gedrückt. Wenn Sie nun eine der mittleren Tasten für die Achsenbewegung betätigen, wird sich der Roboter in der entsprechenden Achse bewegen.

⑪ Position definieren

Halten Sie zum Definieren (Speichern) einer Position die [STEP/MOVE]-Taste gedrückt und betätigen Sie zweimal die [ADD]-Taste. Die momentane Position des Roboters wird unter der in der Anzeige der Teaching Box dargestellten Positionsnummer gespeichert.

Verfahren Sie den Roboter zu einer weiteren Position. Betätigen Sie einmal die [+ /FORWD]-Taste auf der Teaching Box, um eine um „1“ höhere Positionsnummer für das Speichern der neuen Position zu wählen.

Statt der Taste [+ /FORWD]-Taste können Sie auch die Positionsnummer direkt über die Tasten der Teaching Box eingeben. Halten Sie jetzt wie zuvor die [STEP/MOVE]-Taste gedrückt und betätigen Sie erneut zweimal die [ADD]-Taste. Die derzeitige Position wird unter der gewählten Positionsnummer gespeichert.

⑫ Definierte Positionen zum Testen anfahren

Geben Sie mit den Tasten der Teaching Box die Positionsnummer ein, die Sie zum Testen anfahren wollen oder betätigen Sie so oft die [+ /FORWD]-Taste bzw. [- /BACKWD]-Taste, bis die gewünschte Positionsnummer in der Anzeige der Teaching Box erscheint.

Wenn Sie die [STEP/MOVE]-Taste gedrückt halten und zusätzlich die [INP/EXE]-Taste betätigen, bewegt sich der Roboter zu der gewählten Position.

2 Funktionen

2.1 Steuergerät

2.1.1 Bedien- und Signalelemente des Steuergerätes

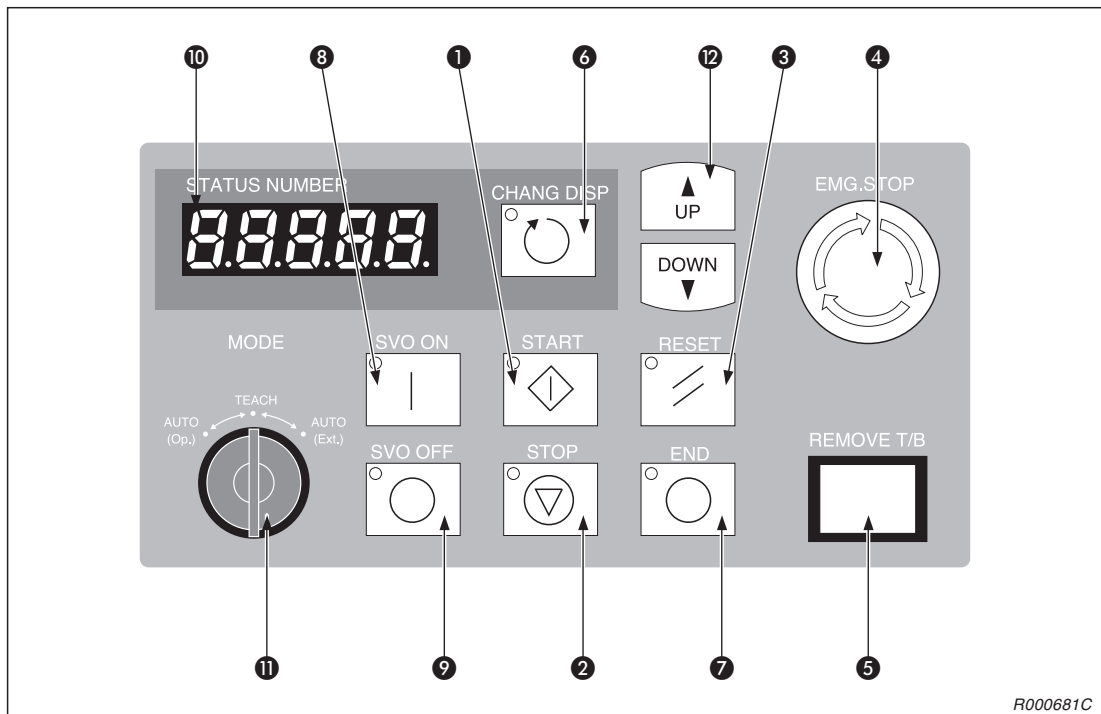


Abb. 2-1: Frontansicht des Steuergerätes

Nr.	Bezeichnung	Funktion
1	START-Taster	Starten eines Programms und Betrieb des Roboters, kontinuierliche Abarbeitung des Programms Die grüne LED leuchtet während des Betriebs.
2	STOP-Taster	Stoppen des Roboters Die Servoversorgungsspannung wird nicht abgeschaltet. Die rote LED leuchtet während eines Stopps.
3	RESET-Taster	Zurücksetzen eines haltenden Programms und Setzen auf den Anfang, Quittierung eines Fehlercodes Die rote LED leuchtet bei anstehendem Fehler.
4	EMG.STOP-Schalter	Der Rastschalter dient dem NOT-HALT des Robotersystems. Wird der Schalter gedrückt, erfolgt die unmittelbare Abschaltung der Servoversorgungsspannung und der sich bewegende Roboter hält sofort an. Durch Rechtsdrehen wird der Schalter entriegelt und springt wieder heraus.
5	REMOVE T/B-Tastschalter	Betätigen Sie den Schalter, wenn Sie die ausgeschaltete (disable) Teaching Box bei eingeschalteter Versorgungsspannung des Steuergerätes anschließen bzw. den Anschluss lösen möchten.

Tab. 2-1: Beschreibung der Bedien- und Signalelemente auf der Frontseite des Steuergerätes (1)

Nr.	Bezeichnung	Funktion	
⑥	CHANG.DISP-Taster	Anzeigenwechsel auf dem Display des Steuergerätes in der Reihenfolge: Programmnummer → Zeilennummer → Übersteuerung Bei aufgetretenem Fehler erscheint: Programmnummer → Zeilennummer → Übersteuerung nur bei betätigtem Taster. Bei nicht betätigtem Taster erscheint die Fehlernummer.	
⑦	END-Taster	Stoppen des laufenden Programms in der letzten Zeile oder bei der END-Anweisung Die rote LED leuchtet bei zyklischem Betrieb.	
⑧	SVO.ON-Taster	Einschalten der Servoversorgungsspannung Die grüne LED leuchtet bei eingeschalteter Servoversorgungsspannung.	
⑨	SVO.OFF-Taster	Abschalten der Servoversorgungsspannung Die rote LED leuchtet bei ausgeschalteter Servoversorgungsspannung.	
⑩	STATUS.NUMBER-Anzeige	Anzeige von Alarm-, Fehlernummer, Übersteuerungswert (%) usw.	
⑪	MODE-Umschalter	AUTO (Op.)	Ein Betrieb ist ausschließlich über das Steuergerät möglich. Der Betrieb über externe Signale oder die Teaching Box ist gesperrt.
		TEACH	Bei aktivierter Teaching Box ist ausschließlich ein Betrieb über die Teaching Box möglich. Der Betrieb über externe Signale oder das Steuergerät ist gesperrt.
		AUTO (Ext.)	Ein Betrieb ist ausschließlich über externe Signale möglich. Der Betrieb über die Teaching Box oder das Steuergerät ist gesperrt.
⑫	UP/DOWN-Taster	Scrollen der Anzeige	

Tab. 2-1: Beschreibung der Bedien- und Signalelemente auf der Frontseite des Steuergerätes (2)

2.2 Teaching Box

2.2.1 Display-Anzeigen und Funktionen

Display	Funktion	Referenz
Eröffnungsbildschirm <pre>CRn-5xxVer.A1 RP-1AH Copyright (C)1999 PRESS KEY</pre>	Anzeige des Robotertyps und der Software-Version	Abs. 7.1.1 „Bedienung der Teaching Box“
Hauptmenü <pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT 6 . SET</pre>	Auswahl der folgenden Menüs	
Programmauswahl <pre><TEACH> (1) SELECT PROGRAM</pre>	Programmnummer auswählen oder ändern	Abs. 7.2.1 „Roboterprogramm erstellen“
Programmstart <pre><RUN> 1 . SERVO</pre>	Servospannung EIN/AUS	Abs. 7.3 „Servospannung ein-/ausschalten“
Dateifunktionen <pre><FILE> 1 . DIR 2 . COPY 3 . RENAME 4 . DELETE</pre>	Programmverzeichnis anzeigen	Abs. 7.5.1 „Programmverzeichnis anzeigen“
	Programm schützen	Abs. 7.5.2 „Programm schützen“
	Programm kopieren	Abs. 7.5.3 „Programm kopieren“
	Programmnamen ändern	Abs. 7.5.4 „Programmnamen ändern“
	Programm löschen	Abs. 7.5.5 „Programm löschen“
Monitorfunktionen <pre><MONI > 1 . INPUT 2 . OUTPUT 3 . VAR 4 . ERROR 5 . REGISTER</pre>	Eingangssignale anzeigen	Abs. 7.6.1 „Monitor-Funktion für Eingangssignale“
	Ausgangssignale anzeigen/einstellen	Abs. 7.6.2 „Monitor-Funktion für Ausgangssignale“
	Variablen anzeigen	Abs. 7.6.3 „Monitor-Funktion für Variable“
	Alarmliste anzeigen	Abs. 7.6.4 „Liste der aufgetretenen Fehlermeldungen“
	Register anzeigen	Anwendung bei CC-Link-Optionen Benutzerhandbuch CC-Link-Schnittstelle

Tab. 2-2: Display-Anzeigen und Funktionen (1)

Display	Funktion	Referenz
Wartungsfunktionen <MAINT> 1 . PARAM 2 . INIT 3 . BRAKE 4 . ORIGIN 5 . POWER	Parameter anzeigen/einstellen	Abs. 7.7.1 „Parameter einstellen“
	Speicher löschen	Abs. 7.7.2 „Alle gespeicherten Programme löschen“
	Batteriezüher zurücksetzen	Abs. 7.7.3 „Batteriezüher zurücksetzen“
	Gelenkbremse lösen	Abs. 7.7.4 „Gelenkbremsen lösen“
	Grundposition einstellen	Technisches Handbuch
	Batterie- und Einschaltzeit anzeigen	Abs. 7.7.5 „Batterie und Einschaltzeit anzeigen“
Uhrzeit/Datum <SET> 1 . CLOCK	Uhrzeit und Datum anzeigen/einstellen	Abs. 7.7.6 „Uhrzeit und Datum einstellen“

Tab. 2-2: Display-Anzeigen und Funktionen (2)

2.2.2 Bedienelemente der Teaching Box

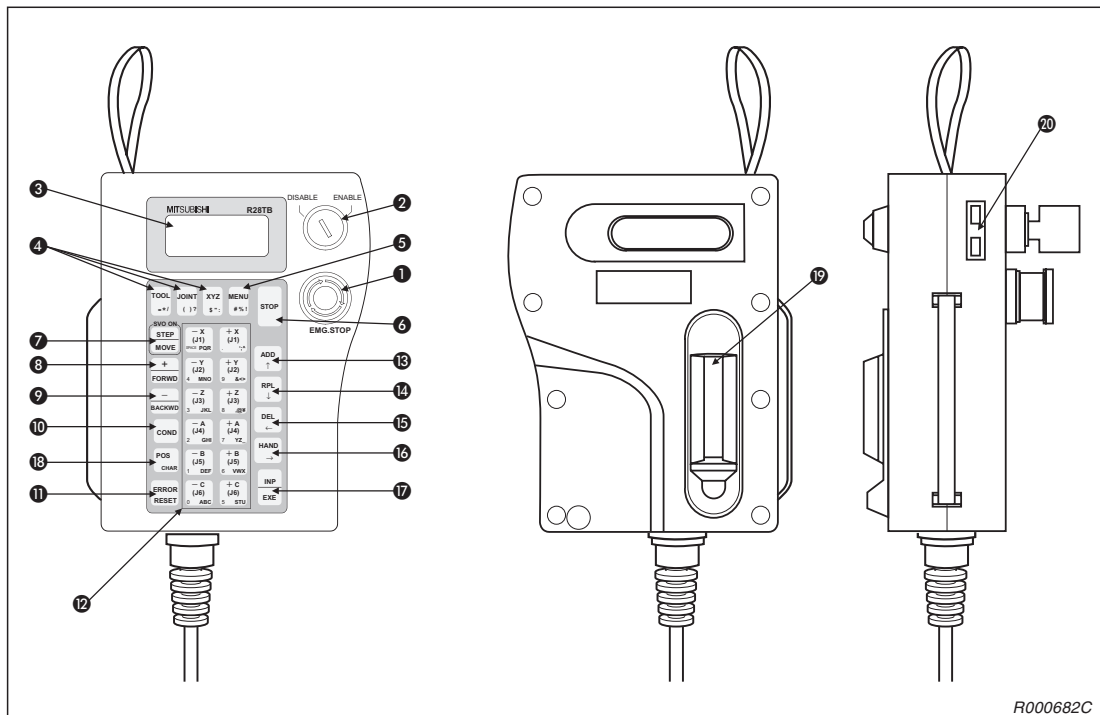








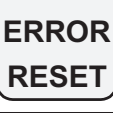

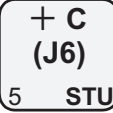








Abb. 2-2: Bedienelemente der Teaching Box

Nr.	Schalter / Taste	Beschreibung
1		Drucktaster mit Verriegelungsfunktion für NOT-HALT Nach Betätigung wird der Roboter unabhängig vom jeweiligen Betriebszustand sofort gestoppt. Durch Drehen der Drucktasterfläche wird der Taster entriegelt.
2		Freigabe der Steuerung über die Teaching Box Bringen Sie den Schalter in die Stellung „ENABLE“, um die Steuerung über die Teaching Box zu übernehmen. Wenn die Teaching Box aktiv ist, kann weder über das Bedienfeld des Steuergerätes noch von extern in die Steuerung eingegriffen werden.
3	LCD-Anzeige	Auf der LDC-Anzeige (4 Zeilen x 16 Zeichen) wird das aktuell ausgewählte Programm oder der Betriebszustand des Roboters angezeigt.
4		Auswahl des Werkzeug-Jog-Betriebs
		Auswahl des Gelenk-Jog-Betriebs
		Auswahl des XYZ-Jog- oder Kreis-Jog-Betriebs

Tab. 2-3: Bedienelemente der Teaching Box (1)

Nr.	Schalter / Taste	Beschreibung
5		Rücksprung ins Hauptmenü
6		Programmablauf und Roboterbewegung stoppen Die Taste hat die gleiche Funktion wie die STOP-Taste auf der Frontseite des Steuergerätes. Die Tastenfunktion ist unabhängig von der Stellung des [ENBL/DISABLE]-Schalters immer verfügbar.
7	  	Ausführen des Jog-Betriebs in Verbindung mit den Jog-Tasten 12, Ausführen von Anweisungsschritten in Verbindung mit der [INP/EXE]-Taste, Einschalten der Servoversorgungsspannung
8		Ausführen von Vorwärtsschritten in Verbindung mit der [INP/EXE]-Taste, Anzeige der nächsten Programmzeile im Editiermodus, Zunahme der Übersteuerung in Verbindung mit der [STEP/MOVE]-Taste
9		Ausführen von Rückwärtsschritten in Verbindung mit der [INP/EXE]-Taste, Anzeige der vorherigen Programmzeile im Editiermodus, Abnahme der Übersteuerung in Verbindung mit der [STEP/MOVE]-Taste
10		Editierung des Programms
11		Rücksetzen eines Alarms, Rücksetzen des Programms in Verbindung mit der [INP/EXE]-Taste
12	JOG-Tasten  bis 	Funktionstasten für Jog-Betrieb Im Gelenk-Jog-Betrieb können alle Gelenke einzeln bewegt werden. Im XYZ-Jog-Betrieb kann der Roboterarm an jeder der Koordinatenachsen entlang bewegt werden. Mit den Tasten erfolgt auch die Eingabe von Menüauswahlnummern oder Schrittnummern.
13		Zur Eingabe von Positionen oder Cursor nach oben bewegen
14		Zur Änderung von Positionen oder Cursor nach unten bewegen

Tab. 2-3: Bedienelemente der Teaching Box (2)

Nr.	Schalter / Taste	Beschreibung
15		Zum Löschen von Positionen oder Cursor nach links bewegen
16		In Verbindung mit der [+C/(J6)]- oder [-C/(J6)]-Taste zum Bewegen der ersten Greifhand, in Verbindung mit der [+B/(J5)]- oder [-B/(J5)]-Taste zum Bewegen der zweiten Greifhand, in Verbindung mit der [+A/(J4)]- oder [-A/(J4)]-Taste zum Bewegen der dritten Greifhand, in Verbindung mit der [+Z/(J3)]- oder [-Z/(J3)]-Taste zum Bewegen der vierten Greifhand oder den Cursor nach rechts zu bewegen
17		Zur Dateneingabe oder Schrittweilerschaltung
18		Zum Wechsel zwischen Zahlen und Buchstaben beim Editieren von Positionsdaten usw.
19	Totmannschalter	Bei eingeschalteter Teaching Box wird der Servoantrieb bei nicht betätigtem oder durchgedrücktem Dreistufen-Totmannschalter ausgeschaltet. Für ein Einschalten des Servoantriebes muss der Totmannschalter bis zur Mittelstellung betätigt sein. Ist der Servoantrieb während eines NOT-AUS oder einer Befehlsausführung ausgeschaltet, kann er durch den Totmannschalter nicht eingeschaltet werden. Schalten Sie in diesem Fall den Servoantrieb über das Hauptmenü „2.RUN“ ein.
20	LCD-Kontrasteinstellung	Zur Helligkeitseinstellung der LCD-Anzeige

Tab. 2-3: Bedienelemente der Teaching Box (3)

2.3 Betriebsrechte

Beim Anschluss mehrerer Geräte, z. B. Teaching Box und Personalcomputer, an das Steuergerät verfügt nur ein Gerät über die Betriebsrechte.

Zur Ausführung von Vorgängen, die den Roboter starten, z. B. ein Programmstart, benötigt ein Gerät die Betriebsrechte. Im Gegensatz dazu können alle Vorgänge, die den Roboter stoppen, z. B. ein Stopp-Befehl oder ein Ausschalten der Servoversorgung, aus Sicherheitsgründen auch ohne Betriebsrechte ausgeführt werden.

Schalter	T/B [ENABLE/DISABLE]	DISABLE			ENABLE		
	Steuergerät [MODE]	AUTO (Op.)	AUTO (Ext.)	TEACH	AUTO (Op.)	AUTO (Ext.)	TEACH
Betriebsrechte	T/B	—	—	—	— ^②	— ^②	●
	Steuergerät	●	—	—	— ^②	— ^②	—
	PC	—	● ^①	—	— ^②	— ^②	—
	Externes Signal	—	● ^①	—	— ^②	— ^②	—

Tab. 2-4: Einstellung der Betriebsrechte

- ① Erfolgt die Eingabe des Signals IOENA (Eingabe Betriebsrechte) über ein externes Gerät, besitzt das externe Signal die Betriebsrechte und die Betriebsrechte des PCs sind deaktiviert.
- ② Ist die Teaching Box auf „ENABLE“ gesetzt, erfolgt bei einer Einstellung der [MODE]-Taste auf die Stellung „AUTO“ die Fehlermeldung „5000“.

In folgender Tabelle sind die Vorgänge aufgeführt, die ein Betriebsrecht erfordern:

Vorgang	Betriebsrecht erforderlich	Funktion
Operation	●	Servo EIN
	—	Servo AUS
	●	Programmstart
	—	Programmstopp/Zyklusstopp
	●	Anwendungsinitialisierung (Programm zurücksetzen)
	—	Alarm zurücksetzen
	●	Geschwindigkeitsübersteuerung ändern (ist über die T/B immer möglich)
	—	Einlesen der Geschwindigkeitsübersteuerung
	●	Programmnummer ändern
	—	Programm-/Zeilennummer lesen
Ein-/Ausgangssignalfunktion	—	Eingangs-/Ausgangssignal lesen
	—	Ausgangssignal schreiben
	●	Spezielle Eingänge: Start, Reset, Servo EIN, Bremse EIN/AUS, manueller Moduswechsel, allgemeinen Ausgang zurücksetzen, Programmnummer festlegen, Zeilennummer festlegen, Geschwindigkeitsübersteuerung festlegen
	—	Spezielle Eingänge: Stopp, Servo AUS, kontinuierlicher Betrieb, Eingangssignal Betriebsrechte, Ausgabeanforderung Programmnummer, Ausgabeanforderung Zeilennummer, Ausgabeanforderung Geschwindigkeitsübersteuerung, Ausgabeanforderung Fehlernummer/numerische Eingabe
	—	Handsensor-/Handsteuersignal lesen
	●	Handsteuersignal schreiben
Programm- editierung ^①	—	Zeilennummer eingeben, lesen, aufrufen; Position hinzufügen, korrigieren, lesen; Variable schreiben, lesen
	●	Schrittweitschaltung, Ausführung
	—	Vorwärts-/Rückwärtsschritt
	●	Sprung, direkte Ausführung, JOG-Betrieb
Dateifunktion	—	Programmverzeichnis lesen, Programm schützen/kopieren/löschen/umbenennen/zurücksetzen
Wartungs- funktion	—	Parameter ändern/lesen, Uhrzeit einstellen/lesen, Betriebszeit lesen, Alarmliste lesen
	●	Grundposition einstellen

Tab. 2-5: Vorgänge und Betriebsrechte

^① Wird über ein Gerät eine Editierung online ausgeführt, ist keine Editierung über ein anderes Gerät möglich.

3 MELFA-BASIC-IV-Programmierung

In diesem Kapitel finden Sie eine Einführung in die Programmiersprache MELFA-BASIC IV. Eine detaillierte Beschreibung der einzelnen Befehle finden Sie in Kapitel 9 „MELFA-BASIC-IV-Befehle“.

3.1 Programmaufbau

In diesem Abschnitt werden die Elemente zum Aufbau einer Anweisung erläutert.

3.1.1 Anweisung

<u>10</u>	<u>MOV P1</u>	<u>WTH M_OUT(17) = 1</u>
①	② ③	④

- ① Zeilennummer
Für die einwandfreie Funktion eines Programms müssen die Zeilennummern in aufsteigender Reihenfolge angeordnet sein. Das Programm wird in dieser Reihenfolge abgearbeitet.
- ② Befehl
Der Befehl legt die Aktion des Roboters fest.
- ③ Befehlsparameter
Der Befehlsparameter kann z. B. eine Variable oder ein Wert sein.
- ④ Angehängte Anweisung
Bei Interpolationsbefehlen ist es möglich, eine Verknüpfung an die Anweisung anzuhängen. Durch Anhängen einer Verknüpfung können bestimmte Befehle parallel zum Interpolationsbefehl ausgeführt werden.

3.1.2 Variable

Folgende Variablen können in einem Programm verwendet werden:

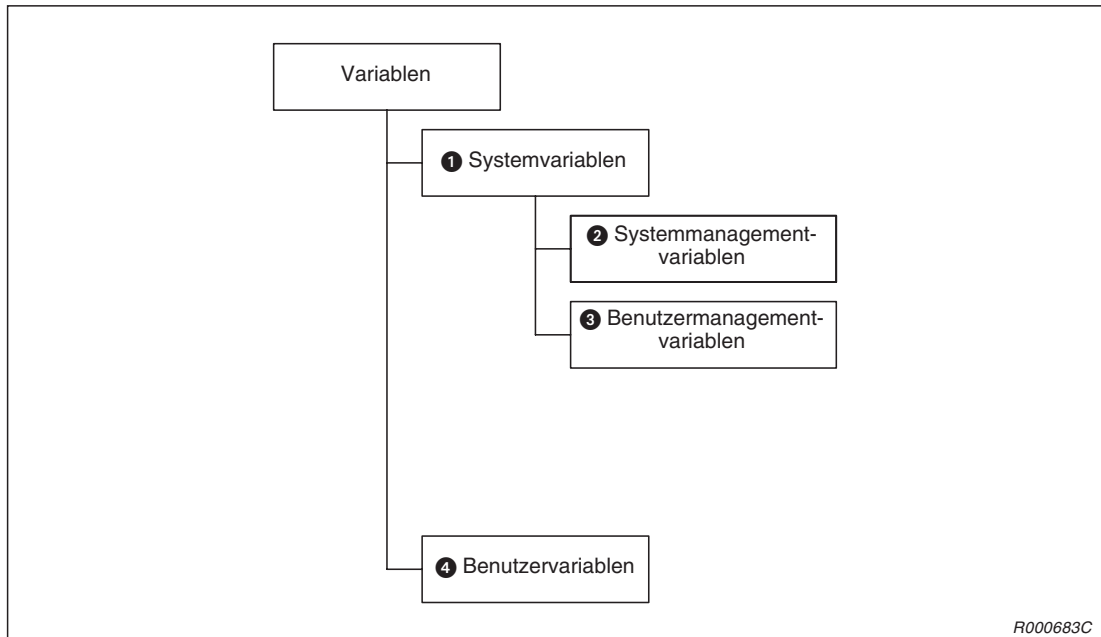


Abb. 3-1: Einteilung der Variablen

- ① Systemvariablen sind durch einen Variablennamen und einen gespeicherten Wert definiert.
- ② Systemmanagementvariablen können nur gelesen werden.
Beispiel: P_CURR
In dieser Variablen wird die aktuelle Position des Roboters ständig gespeichert.
- ③ Benutzermanagementvariablen können gelesen und geschrieben werden. Eingangssignale können nur gelesen werden.
Beispiel: M_OUT(17) = 1: Ausgangsbit 17 einschalten
M1 = M_IN(20): Schreibe den Wert des Eingangsbits 20 in die arithmetische Variable M1
- ④ Benutzervariablen sind durch einen Variablennamen und den Verwendungszweck definiert.

Jede der oben aufgeführten Variablentypen ist in die folgenden Gruppen eingeteilt:

- **Positionsvariablen**
Eine Positionsvariable enthält die kartesischen Koordinaten des Roboters. Der Variablennamen beginnt mit „P“.
Beispiel: MOV P1 Der Roboter fährt die Position an, die in der Variablen P1 abgespeichert ist.

- **Gelenkvariablen**
Eine Gelenkvariable enthält die Winkelwerte der Robotergelenke. Der Variablenname beginnt mit „J“.
Beispiel: MOV J1 Der Roboter fährt die Position an, die in der Variablen J1 abgespeichert ist.

- **Arithmetische Variablen**
Eine arithmetische Variable enthält einen numerischen Wert (Integer, Reelle Zahl, usw.). Der Variablennamen beginnt mit „M“.
Beispiel: M1 = 1 Der Wert „1“ wird in die Variable M1 geschrieben.

- **Zeichenkettenvariablen**
Eine Zeichenkettenvariable enthält eine Zeichenkette. Dem Variablennamen folgt das Zeichen „\$“.
Beispiel: C1\$ = "ERROR" Die Zeichenkette „ERROR“ wird in die Variable C1\$ geschrieben.

3.2 Steuerung der Roboterbewegung

3.2.1 Gelenk-Interpolation

Die Handspitze wird mittels Gelenk-Interpolation zu einer festgelegten Position bewegt.

Erläuterung

Befehl	Beschreibung
MOV	Bewegt die Handspitze mittels Gelenk-Interpolation zu einer festgelegten Position Mit Hilfe der Verknüpfungen WTH oder WTHIF kann eine Anweisung angehängt werden.

Anweisungsbeispiele

MOV P1	Position 1 anfahren
MOV P1 + P2	Position anfahren, die sich aus der Addition der Koordinaten der Positionen 1 und 2 ergibt
MOV P1 * P2	Position anfahren, die sich aus der relativen Konvertierung von P1 zu P2 ergibt
MOV P1, -50	Position anfahren, die 50 mm in Werkzeuglängsrichtung von der Position 1 entfernt ist (siehe Achtungshinweis)
MOV P1 WTH M_OUT(17) = 1	Position 1 anfahren und Ausgangsbit 17 auf 1 setzen
MOV P1 WTHIF M_IN(20) = 1 SKIP	Wird beim Anfahren der Position 1 das Eingangsbit 20 auf 1 gesetzt, wird die Verfahrbewegung unterbrochen und das Programm bis zum nächsten Stopp fortgesetzt.



ACHTUNG:

Die Richtung des Verfahrwegs im Werkzeugkoordinatensystem hängt vom Werkzeugkoordinatensystem des Roboters ab. Detaillierte Informationen zum Werkzeugkoordinatensystem finden Sie im Technischen Handbuch des Roboters.

Programmbeispiel

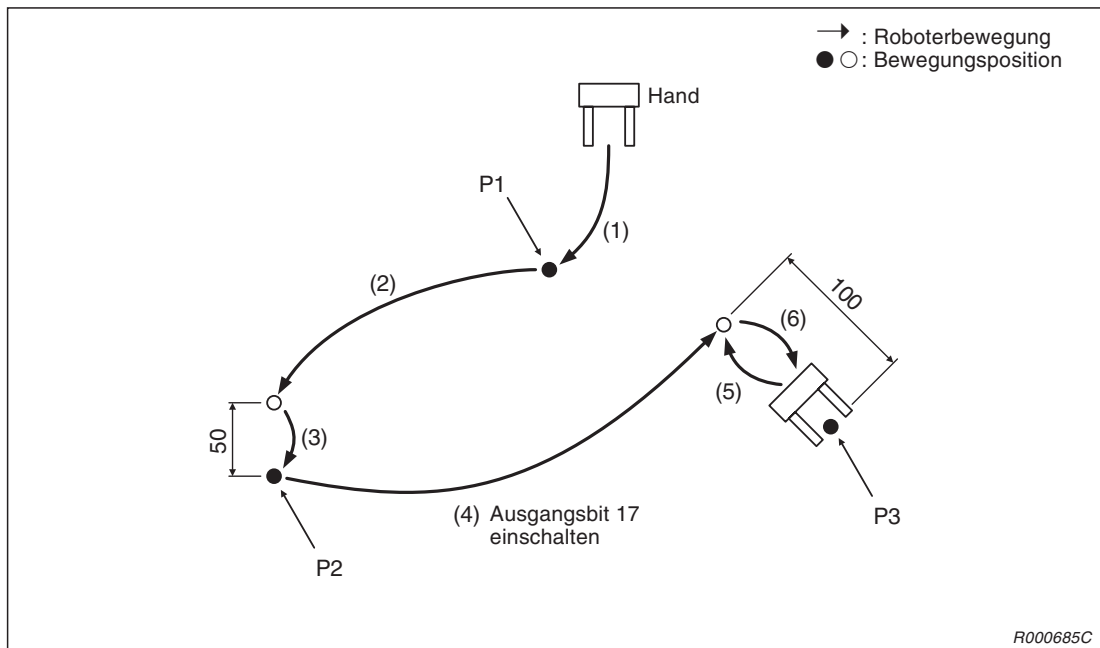


Abb. 3-2: Verlauf des Verfahrenweges bei Gelenk-Interpolation

<p>10 MOV P1 20 MOV P2, -50 30 MOV P2 40 MOV P3, -100 WTH M_OUT(17) = 1 50 MOV P3 60 MOV P3, -100 70 END</p>	<p>Position 1 anfahren Position anfahren, die 50 mm in Werkzeuglängsrichtung von der Position 2 entfernt ist (siehe Achtungshinweis) Position 2 anfahren Position anfahren, die 100 mm in Werkzeuglängsrichtung von Position 3 entfernt ist und Ausgangsbit 17 auf 1 setzen Position 3 anfahren Position anfahren, die 100 mm in Werkzeuglängsrichtung von Position 3 entfernt ist Programmende</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



ACHTUNG:
Die Richtung des Verfahrenweges im Werkzeugkoordinatensystem hängt vom Werkzeugkoordinatensystem des Roboters ab. Detaillierte Informationen zum Werkzeugkoordinatensystem finden Sie im Technischen Handbuch des Roboters.

Befehl steht in Beziehung zu folgenden Funktionen:

Festlegung der Verfahrensgeschwindigkeit	⇒	Abs. 3.2.5
Festlegung der Beschleunigungs-/Bremszeit	⇒	Abs. 3.2.5
Feinpositionierung	⇒	Abs. 3.2.6
Kontinuierliche Bewegung	⇒	Abs. 3.2.4
Linear-Interpolation	⇒	Abs. 3.2.2
Kreis-Interpolation	⇒	Abs. 3.2.3
Angehängte Anweisung	⇒	Abs. 3.8

3.2.2 Linear-Interpolation

Die Handspitze wird mittels Linear-Interpolation zu einer festgelegten Position bewegt.

Erläuterung

Befehl	Beschreibung
MVS	Bewegt die Handspitze mittels Linear-Interpolation zu einer festgelegten Position Mit Hilfe der Verknüpfungen WTH oder WTHIF kann eine Anweisung angehängt werden.

Anweisungsbeispiele

MVS P1	Position 1 anfahren
MVS P1 + P2	Position anfahren, die sich aus der Addition der Koordinaten der Positionen 1 und 2 ergibt
MVS P1 * P2	Position anfahren, die sich aus der relativen Konvertierung von P1 zu P2 ergibt
MVS P1, -50	Position anfahren, die 50 mm in Werkzeuglängsrichtung von der Position 1 entfernt ist (siehe Achtungshinweis)
MVS, -50	Position anfahren, die 50 mm in Werkzeuglängsrichtung von der aktuellen Position entfernt ist (siehe Achtungshinweis)
MVS P1 WTH M_OUT(17) = 1	Position 1 anfahren und Ausgangsbit 17 auf 1 setzen
MVS P1 WTHIF M_IN(20) = 1 SKIP	Wird beim Anfahren der Position 1 das Eingangsbit 20 auf 1 gesetzt, wird die Verfahrbewegung unterbrochen und das Programm bis zum nächsten Stopp fortgesetzt.
MVS P1, TYPE 0, 2	Position 1 mittels ABS-Interpolation anfahren
MVS P1, TYPE 0, 1	Position 1 mittels orthogonaler Interpolation anfahren



ACHTUNG:

Die Richtung des Verfahrwegs im Werkzeugkoordinatensystem hängt vom Werkzeugkoordinatensystem des Roboters ab. Detaillierte Informationen zum Werkzeugkoordinatensystem finden Sie im Technischen Handbuch des Roboters.

Programmbeispiel

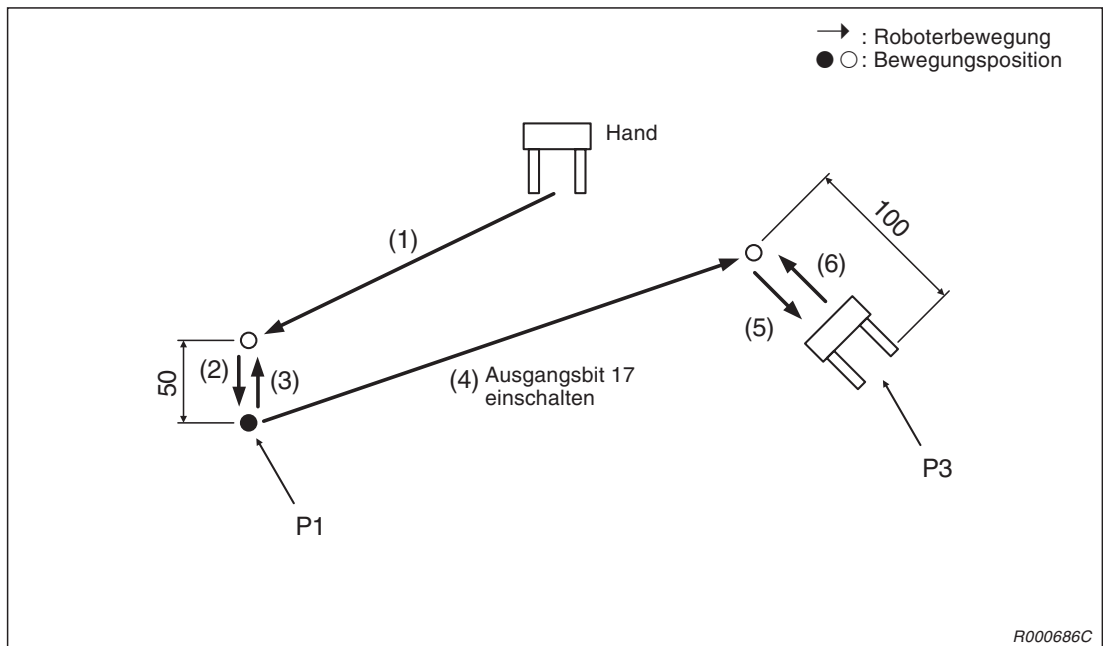


Abb. 3-3: Verlauf des Verfahrenweges bei Linear-Interpolation

<p>10 MVS P1, -50</p> <p>20 MVS P1</p> <p>30 MVS, -50</p> <p>40 MVS P2, -100 WTH M_OUT(17) = 1</p> <p>50 MVS P2</p> <p>60 MVS, -50</p> <p>70 END</p>	<p>Position mittels Linear-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der Position 1 entfernt ist (siehe Achtungshinweis)</p> <p>Position 1 mittels Linear-Interpolation anfahren</p> <p>Position mittels Linear-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der aktuellen Position (P1) entfernt ist (siehe Achtungshinweis)</p> <p>Position mittels Linear-Interpolation anfahren, die 100 mm in Werkzeuglängsrichtung von Position 2 entfernt ist und Ausgangsbit 17 auf 1 setzen</p> <p>Position 2 mittels Linear-Interpolation anfahren</p> <p>Position mittels Linear-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der aktuellen Position (P2) entfernt ist (siehe Achtungshinweis)</p> <p>Programmende</p>
------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



ACHTUNG:
 Die Richtung des Verfahrenweges im Werkzeugkoordinatensystem hängt vom Werkzeugkoordinatensystem des Roboters ab. Detaillierte Informationen zum Werkzeugkoordinatensystem finden Sie im Technischen Handbuch des Roboters.

Befehl steht in Beziehung zu folgenden Funktionen:

Festlegung der Verfahrgeschwindigkeit	⇒	Abs. 3.2.5
Festlegung der Beschleunigungs-/Bremszeit	⇒	Abs. 3.2.5
Feinpositionierung	⇒	Abs. 3.2.6
Kontinuierliche Bewegung	⇒	Abs. 3.2.4
Gelenk-Interpolation	⇒	Abs. 3.2.1
Kreis-Interpolation	⇒	Abs. 3.2.3
Angehängte Anweisung	⇒	Abs. 3.8

3.2.3 Kreis-Interpolation

Die Handspitze wird mittels 3D-Kreis-Interpolation entlang eines durch 3 Punkte festgelegten Kreises zu einer festgelegten Position bewegt. Entspricht die aktuelle Position nicht der Startposition, wird die Startposition mittels Linear-Interpolation angefahren.

Erläuterung

Befehl	Beschreibung
MVR	Bewegt die Handspitze mittels 3D-Kreis-Interpolation entlang eines durch die Startposition, Zwischenposition und Endposition festgelegten Kreisbogens Mit Hilfe der Verknüpfungen WTH oder WTHIF kann eine Anweisung angehängt werden.
MVR 2	Bewegt die Handspitze mittels 3D-Kreis-Interpolation von der Startposition zur Endposition Der Kreisbogen wird durch die Startposition, die Referenzposition und die Endposition festgelegt. Die Roboterbewegung geht dabei nicht durch den Referenzpunkt. Mit Hilfe der Verknüpfungen WTH oder WTHIF kann eine Anweisung angehängt werden.
MVR 3	Bewegt die Handspitze mittels 3D-Kreis-Interpolation von der Startposition zur Endposition Der Kreisbogen wird durch die Startposition, den Mittelpunkt und die Endposition festgelegt. Der Zentriwinkel zwischen Start- und Endposition liegt dabei zwischen 0° und 180°. Mit Hilfe der Verknüpfungen WTH oder WTHIF kann eine Anweisung angehängt werden.
MVC	Bewegt die Handspitze mittels 3D-Kreis-Interpolation entlang eines durch Startposition (Endposition), Zwischenposition 1, Zwischenposition 2 und Endposition festgelegten Kreisbogens Mit Hilfe der Verknüpfungen WTH oder WTHIF kann eine Anweisung angehängt werden.

Anweisungsbeispiele

MVR P1, P2, P3

Bewegung entlang des Kreisbogens
P1 → P2 → P3

MVR P1, P2, P3 WTH M_OUT(17) = 1

Bewegung entlang des Kreisbogens
P1 → P2 → P3 und Ausgangsbit 17 auf 1 setzen

MVR P1, P2, P3 WTHIF M_IN(20) = 1, SKIP

Fährt entlang des Kreisbogens
P1 → P2 → P3 und unterbricht die Bewegung, wenn Eingangsbit 20 auf 1 gesetzt wird
Die Programmsteuerung springt in die nächste Zeile.

MVR P1, P2, P3 TYPE 0, 1

Bewegung entlang des Kreisbogens
P1 → P2 → P3

MVR2 P1, P3, P11

Bewegung entlang des Kreisbogens von P1 nach P3, ohne die Referenzposition P11 zu durchlaufen

MVR3 P1, P3, P10

Bewegung entlang des Kreisbogens von P1 nach P3 in Richtung des kleineren Zentriwinkels
P10 ist der Mittelpunkt.

MVC P1, P2, P3

Bewegung entlang des Kreisbogens
P1 → P2 → P3 → P1

Programmbeispiel

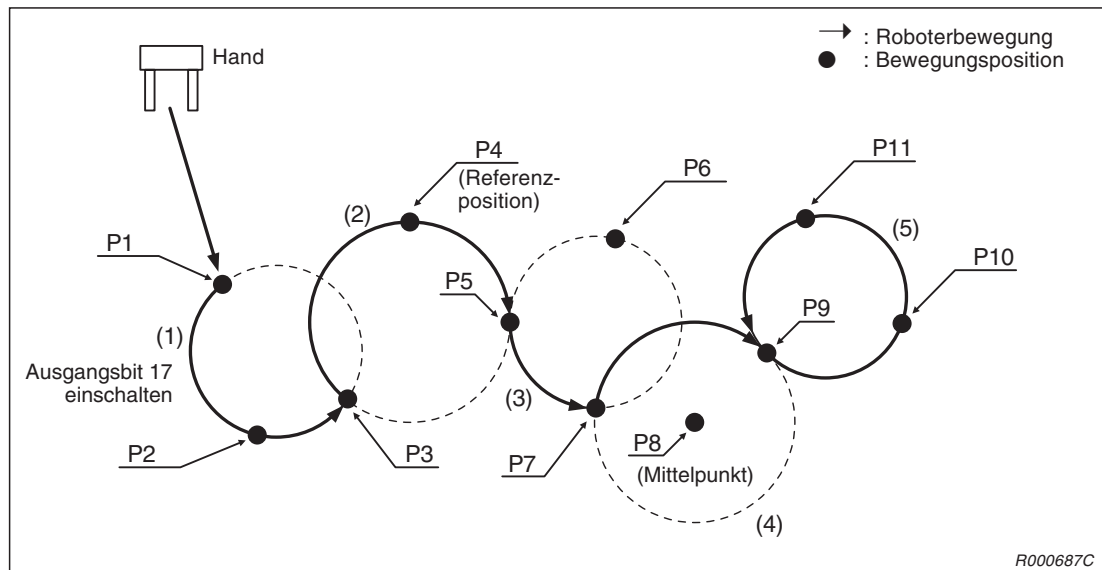


Abb. 3-4 Verlauf des Verfahrensweges bei Kreis-Interpolation

- | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>10 MVR P1, P2, P3 WTH M_OUT(18) = 1</p> <p>20 MVR P3, P4, P5</p> <p>30 MVR2 P5, P7, P6</p> <p>40 MVR3 P7, P9, P8</p> <p>50 MVC P9, P10, P11</p> <p>60 END</p> | <p>Bewegung entlang des Kreisbogens
P1 → P2 → P3
Die aktuelle Position entspricht nicht der Startposition. Der Roboter bewegt sich also zuerst mittels Linear-Interpolation zur Startposition (P1).
Mit Beginn der Kreis-Interpolation wird das Ausgangsbit 18 auf 1 gesetzt.</p> <p>Bewegung entlang des Kreisbogens
P3 → P4 → P5</p> <p>Bewegung entlang des Kreisbogens von P5 nach P7, ohne die Referenzposition P6 zu durchlaufen</p> <p>Bewegung entlang des Kreisbogens von P7 nach P9 in Richtung des kleineren Zentriwinkels
P8 ist der Mittelpunkt.</p> <p>Bewegung entlang des Kreisbogens
P9 → P10 → P11 → P9
Entspricht die aktuelle Position nicht der Startposition, wird die Startposition mittels Linear-Interpolation angefahren (1 Zyklus).
Programmende</p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Befehl steht in Beziehung zu folgenden Funktionen:

- | | | |
|-------------------------------------------|---|------------|
| Festlegung der Verfahrensgeschwindigkeit | ⇒ | Abs. 3.2.5 |
| Festlegung der Beschleunigungs-/Bremszeit | ⇒ | Abs. 3.2.5 |
| Feinpositionierung | ⇒ | Abs. 3.2.6 |
| Kontinuierliche Bewegung | ⇒ | Abs. 3.2.4 |
| Gelenk-Interpolation | ⇒ | Abs. 3.2.1 |
| Linear-Interpolation | ⇒ | Abs. 3.2.2 |
| Angehängte Anweisung | ⇒ | Abs. 3.8 |

3.2.4 Kontinuierliche Bewegung

Bei freigegebener CNT-Einstellung fährt der Roboter die festgelegten Positionen ohne zu stoppen an. Der CNT-Befehl definiert den Start- bzw. Endpunkt der kontinuierlichen Bewegung. Die Geschwindigkeit kann während der kontinuierlichen Bewegung verändert werden.

Befehl	Beschreibung
CNT	Legt den Start- und Endpunkt für die kontinuierliche Bewegung fest

Anweisungsbeispiele

CNT 1	Freigeben der CNT-Einstellung
CNT 1, 100, 200	Freigeben der CNT-Einstellung Der Anfangspunktabstand der kontinuierlichen Bewegung beträgt 100 mm und der Endpunktabstand der kontinuierlichen Bewegung beträgt 200 mm.
CNT 0	Sperren der CNT-Einstellung

Programmbeispiel

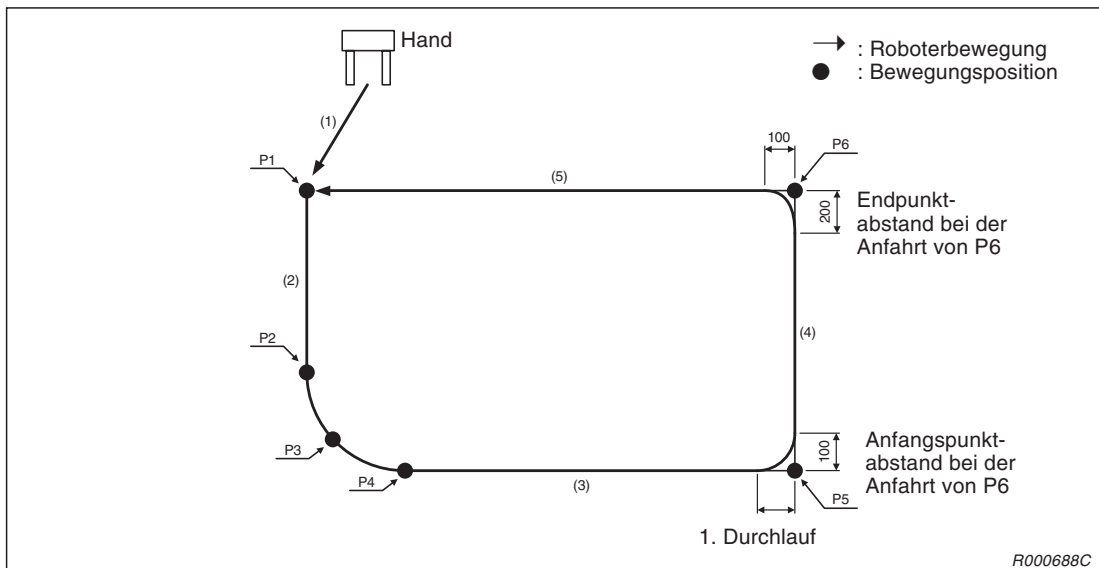


Abb. 3-5: Verlauf des Verfahrenswegs bei kontinuierlicher Bewegung

10	MOV P1	Position 1 mittels Gelenk-Interpolation anfahren
20	CNT 1	CNT-Einstellung freigeben (alle folgenden Bewegungen sind kontinuierlich)
30	MVR P2, P3, P4	Position 2 mittels Linear-Interpolation und Position 4 kontinuierlich mittels Kreis-Interpolation anfahren
40	MVS P5	Position 5 mittels Linear-Interpolation anfahren
50	CNT 1, 100, 200	Anfangspunktabstand der kontinuierlichen Bewegung auf 100 mm und Endpunktabstand der kontinuierlichen Bewegung auf 200 mm festlegen
60	MVS P6	Nach Erreichen von Position 5, Position 6 mittels Linear-Interpolation anfahren
70	MVS P1	Position 1 kontinuierlich mittels Linear-Interpolation anfahren
80	CNT 0	CNT-Einstellung sperren
70	END	Programmende

Befehl steht in Beziehung zu folgenden Funktionen:

Festlegung der Verfahrgeschwindigkeit	⇒	Abs. 3.2.5
Festlegung der Beschleunigungs-/Bremszeit	⇒	Abs. 3.2.5
Feinpositionierung	⇒	Abs. 3.2.6
Gelenk-Interpolation	⇒	Abs. 3.2.1
Linear-Interpolation	⇒	Abs. 3.2.2
Angehängte Anweisung	⇒	Abs. 3.8

3.2.5 Beschleunigungs-/Bremszeit und Geschwindigkeit

Die Beschleunigung/Abbremsung kann bezogen auf den Maximalwert eingestellt werden. Die Geschwindigkeit kann eingestellt werden.

Erläuterung

Befehl	Beschreibung
ACCEL	Die Beschleunigung/Abbremsung kann bezogen auf den Maximalwert (%) eingestellt werden.
OVRD	Die Geschwindigkeit für das gesamte Programm kann bezogen auf den Maximalwert (%) eingestellt werden.
JOVRD	Die Geschwindigkeit für die Gelenk-Interpolation kann bezogen auf den Maximalwert (%) eingestellt werden.
SPD	Legt die Geschwindigkeit (mm/s) für Linear- und Kreis-Interpolation fest

Anweisungsbeispiele

ACCEL	Setzt die Beschleunigung und die Abbremsung auf 100 %
ACCEL 60, 80	Setzt die Beschleunigung auf 60 % und die Abbremsung auf 80 % des Maximalwertes (bei einer maximalen Beschleunigungs-/Bremszeit von 0,2 s ergibt sich eine Beschleunigungszeit von 0,33 s und eine Bremszeit von 0,25 s).
OVRD 50	Legt die Geschwindigkeit für Gelenk-, Linear- und Kreis-Interpolation auf 50 % der maximalen Geschwindigkeit fest
JOVRD 70	Legt die Geschwindigkeit für Gelenk-Interpolation auf 70 % der maximalen Geschwindigkeit fest
SPD 30	Legt die Geschwindigkeit für Linear- und Kreis-Interpolation auf 30 mm/s fest

Die aktuelle Arbeitsgeschwindigkeit ergibt sich:

Gelenk-Interpolation	=	Einstellung über T/B bzw. Steuergerät	x	Einstellwert des OVRD-Befehls	x	Einstellwert des JOVRD-Befehls
Linear-Interpolation	=	Einstellung über T/B bzw. Steuergerät	x	Einstellwert des OVRD-Befehls	x	Einstellwert des SPD-Befehls

Programmbeispiel

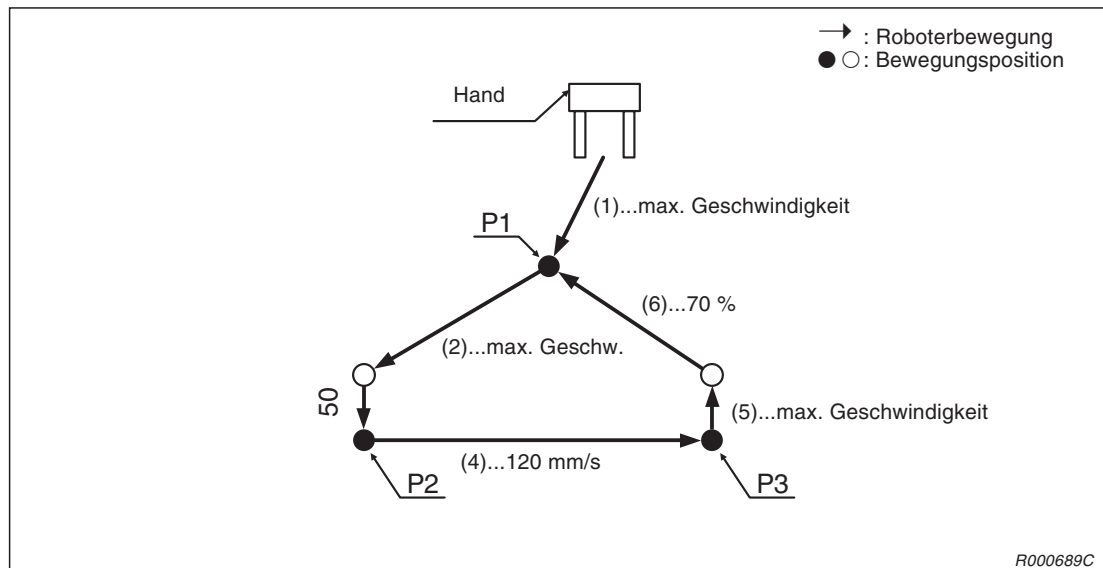


Abb. 3-6: Verfahrenweg und Geschwindigkeiten

10	OVRD 100	Legt die Geschwindigkeit für das gesamte Programm auf den Maximalwert fest
20	MOV P1	Position 1 mit Maximalgeschwindigkeit anfahren
30	MOV P2, -50	Position mit Maximalgeschwindigkeit anfahren, die 50 mm in Werkzeuglängsrichtung von der Position 2 entfernt ist (siehe Achtungshinweis)
40	OVRD 50	Legt die Geschwindigkeit für das gesamte Programm auf den halben Maximalwert fest
50	MVS P2	Position 2 mittels Linear-Interpolation und halber Maximalgeschwindigkeit anfahren
60	SPD 120	Legt die Endgeschwindigkeit auf 120 mm/s fest
70	OVRD 100	Legt die Geschwindigkeit auf 100 % fest, so dass eine Endgeschwindigkeit von 120 mm/s erreicht wird
80	ACCEL 70, 70	Die Beschleunigung/Abbremsung wird auf 70 % des Maximalwerts gesetzt
90	MVS P3	Position 3 mittels Linear-Interpolation und mit einer Endgeschwindigkeit von 120 mm/s anfahren
100	SPD M_NSPD	Setzt die Geschwindigkeit auf den Standardwert zurück
110	JOVRD 70	Legt die Geschwindigkeit für Gelenk-Interpolation auf 70 % fest
120	ACCEL	Legt die Beschleunigung/Abbremsung auf 100 % fest
130	MVS, -50	Position mittels Linear-Interpolation und Standardgeschwindigkeit anfahren, die 50 mm in Werkzeuglängsrichtung von der aktuellen Position (P3) entfernt ist (siehe Achtungshinweis)
140	MOV P1	Position 1 mit 70 % der Maximalgeschwindigkeit anfahren
140	END	Programmende

**ACHTUNG:**

Die Richtung des Verfahrenwegs im Werkzeugkoordinatensystem hängt vom Werkzeugkoordinatensystem des Roboters ab. Detaillierte Informationen zum Werkzeugkoordinatensystem finden Sie im Technischen Handbuch des Roboters.

Befehl steht in Beziehung zu folgenden Funktionen:

Gelenk-Interpolation	⇒	Abs. 3.2.1
Linear-Interpolation	⇒	Abs. 3.2.2
Kreis-Interpolation	⇒	Abs. 3.2.3
Kontinuierliche Bewegung	⇒	Abs. 3.2.4

3.2.6 Feinpositionierung

Der Abschluss eines Positioniervorgangs wird durch eine Anzahl von Impulsen festgelegt. Die Einstellung ist bei Ausführung kontinuierlicher Bewegungen deaktiviert.

Erläuterung

Befehl	Beschreibung
FINE	Legt den Abschluss eines Positioniervorgangs durch eine Anzahl von Impulsen fest

Anweisungsbeispiel

FINE 100 Legt die Anzahl der Impulse zur Feinpositionierung auf 100 fest

Programmbeispiel

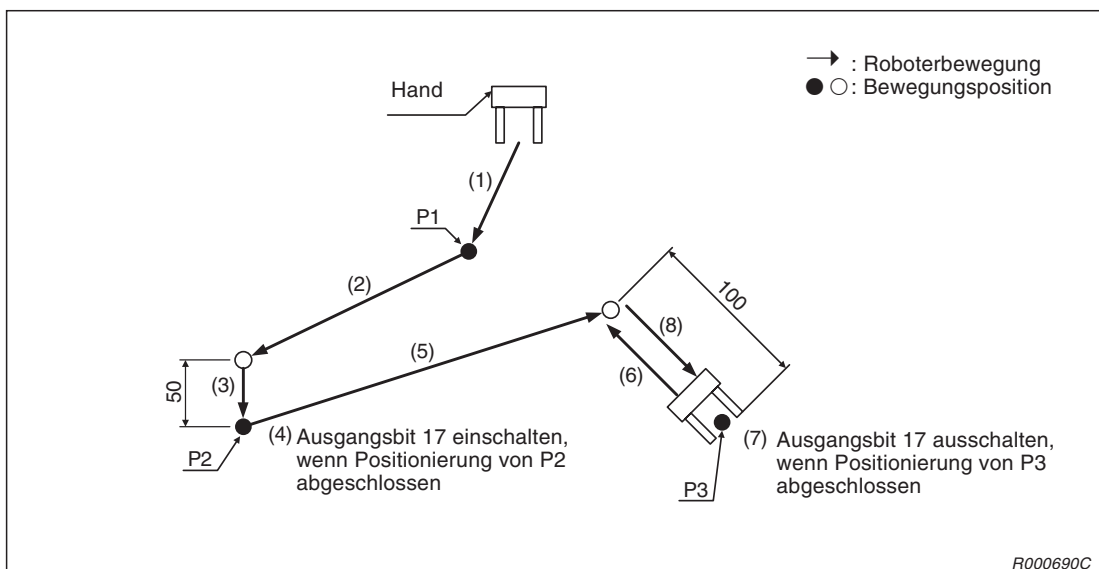


Abb. 3-7: Verfahrenweg und Feinpositionierung

10	MOV P1	Position 1 mittels Gelenk-Interpolation anfahren
20	MOV P2, -50	Position mittels Gelenk-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der Position 2 entfernt ist (siehe Achtungshinweis)
30	FINE 50	Legt die Anzahl der Impulse zur Feinpositionierung auf 50 fest
40	MVS P2	Position 2 mittels Linear-Interpolation anfahren
50	M_OUT(17) = 1	Ausgangsbit 17 wird auf 1 gesetzt, wenn die Anzahl der Impulse 50 erreicht
60	FINE 1000	Legt die Anzahl der Impulse zur Feinpositionierung auf 1000 fest
70	MVS P3, -100	Position mittels Linear-Interpolation anfahren, die 100 mm in Werkzeuglängsrichtung von der Position 3 entfernt ist (siehe Achtungshinweis)
80	FINE 50	Legt die Anzahl der Impulse zur Feinpositionierung auf 50 fest
90	MVS P3	Position 3 mittels Linear-Interpolation anfahren
100	M_OUT(17) = 0	Ausgangsbit 17 wird auf 0 gesetzt, wenn die Anzahl der Impulse 50 erreicht
110	FINE 1000	Legt die Anzahl der Impulse zur Feinpositionierung auf 1000 fest
120	MVS, -100	Position mittels Linear-Interpolation anfahren, die 100 mm in Werkzeuglängsrichtung von der aktuellen Position (P3) entfernt ist (siehe Achtungshinweis)
130	END	Programmende

**ACHTUNG:**

Die Richtung des Verfahrs im Werkzeugkoordinatensystem hängt vom Werkzeugkoordinatensystem des Roboters ab. Detaillierte Informationen zum Werkzeugkoordinatensystem finden Sie im Technischen Handbuch des Roboters.

Befehl steht in Beziehung zu folgenden Funktionen:

Gelenk-Interpolation	⇒	Abs. 3.2.1
Linear-Interpolation	⇒	Abs. 3.2.2
Kreis-Interpolation	⇒	Abs. 3.2.3
Kontinuierliche Bewegung	⇒	Abs. 3.2.4

3.2.7 Hand- und Werkzeugsteuerung

Der Handgreiferzustand (offen/geschlossen) und die Werkzeugdaten können festgelegt werden.

Erläuterung

Befehl	Beschreibung
HOPEN	Die festgelegte Hand wird geöffnet.
HCLOSE	Die festgelegte Hand wird geschlossen.
TOOL	Die Werkzeugdaten und der Überwachungspunkt können eingestellt werden.

Anweisungsbeispiele

HOPEN 1 Öffnet Hand 1
 HOPEN 2 Öffnet Hand 2
 HCLOSE 1 Schließt Hand 1
 HCLOSE 2 Schließt Hand 2
 TOOL P90 Schreibt die Werkzeugdaten in P90

Programmbeispiel

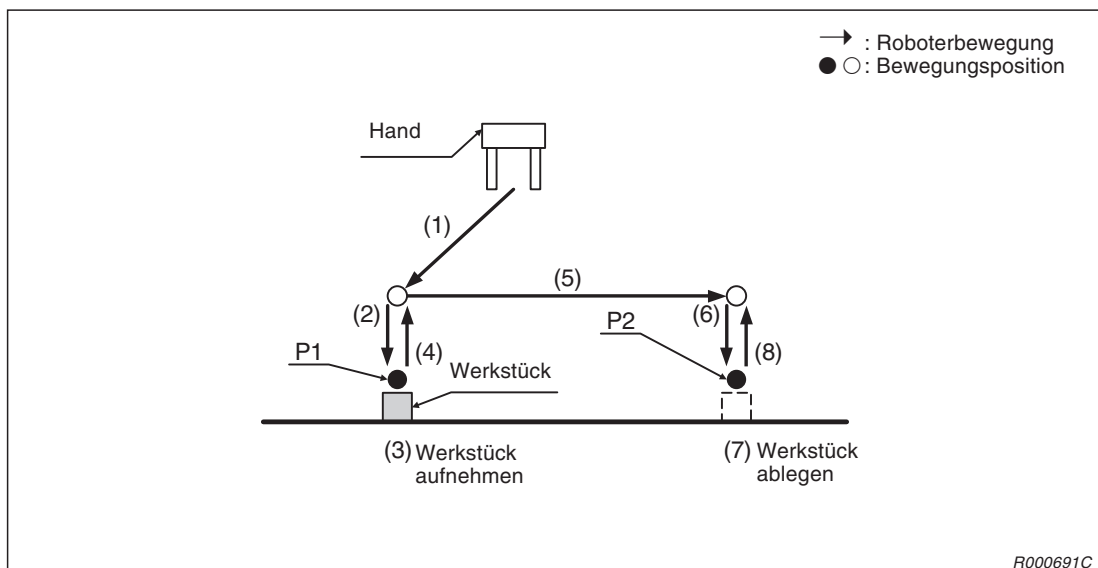


Abb. 3-8: Verfahrensbewegung und Handsteuerung

10	MOV P1, -50	Position mittels Gelenk-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der Position 1 entfernt ist (siehe Achtungshinweis)
20	OVRD 50	Legt die Geschwindigkeit auf den halben Maximalwert fest
30	MVS P1	Position 1 mittels Linear-Interpolation anfahren (Anfahren der Position zur Werkstückaufnahme)
40	HCLOSE 1	Schließt Hand 1 (Werkstück aufnehmen)
50	DLY 0.5	Wartezeit von 0,5 s
60	OVRD 100	Legt die Geschwindigkeit auf den Maximalwert fest
70	MVS, -50	Position mittels Linear-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der aktuellen Position (P1) entfernt ist (Anheben des Werkstücks) (siehe Achtungshinweis)
80	MOV P2, -50	Position mittels Gelenk-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der Position 2 entfernt ist (siehe Achtungshinweis)
90	OVRD 50	Legt die Geschwindigkeit auf den halben Maximalwert fest
100	MVS P2	Position 2 mittels Linear-Interpolation anfahren (Anfahren der Position zur Werkstückablage)
110	HOPEN 1	Öffnet Hand 1 (Werkstück ablegen)
120	DLY 0.5	Wartezeit von 0,5 s
130	OVRD 100	Legt die Geschwindigkeit auf den Maximalwert fest
140	MVS, -50	Position mittels Linear-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der aktuellen Position (P2) entfernt ist (Entfernen vom Werkstück) (siehe Achtungshinweis)
150	END	Programmende

**ACHTUNG:**

Die Richtung des Verfahrenswegs im Werkzeugkoordinatensystem hängt vom Werkzeugkoordinatensystem des Roboters ab. Detaillierte Informationen zum Werkzeugkoordinatensystem finden Sie im Technischen Handbuch des Roboters.

Befehl steht in Beziehung zu folgenden Funktionen:

Angehängte Anweisung ⇒ Abs. 3.8

3.3 Palettierung

Mit Hilfe der Palettierungsfunktion können Werkstücke geordnet abgelegt oder geordnete Werkstücke aufgenommen werden. Dabei reicht ein Teachen der Position des Referenz-Werkstücks aus. Alle anderen Positionen werden daraus berechnet.

Erläuterung

Befehl	Beschreibung
DEF PLT	Definiert eine Palette
PLT	Berechnet die Koordinaten eines Gitterpunktes der festgelegten Palette und weist die berechneten Koordinaten der festgelegten Position zu

Anweisungsbeispiele

DEF PLT 1, P1, P2, P3, P4, 4, 3, 1

Definiert Palette Nummer 1 mit Bezugsposition = P1, Spaltenendpunkt A = P2, Zeilenendpunkt B = P3, Paletteneckpunkt, der gegenüber der Bezugsposition liegt = P4, Anzahl der Gitterpunkte: 12 (Spalten A = 4, Zeilen B = 3) und einer Bewegungsrichtung = 1

DEF PLT 2, P1, P2, P3, 8, 5, 2

Definiert Palette Nummer 2 mit Bezugsposition = P1, Spaltenendpunkt A = P2, Zeilenendpunkt B = P3, Anzahl der Gitterpunkte: 40 (Spalten A = 8, Zeilen B = 5) und einer Bewegungsrichtung = 2

DEF PLT 3, P1, P2, P3, 8, 5, 3

Definiert kreisförmige Palette Nummer 3 mit 5 Positionen auf einem Kreisbogen über Startposition = P1, Zwischenposition = P2 und Endposition = P3 (insgesamt 3 Punkte)

(PLT 1, 5)

Berechnet die 5te Position der Palette Nummer 1

(PLT 1, M1)

Berechnet die in der numerischen Variablen M1 festgelegte Position der Palette Nummer 1

Palettendefinition und Bewegungsrichtung

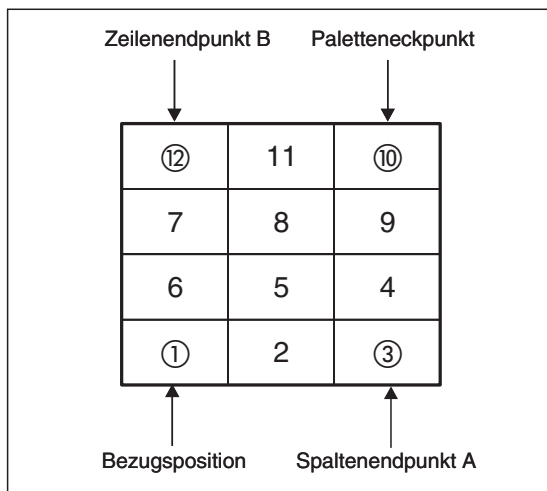


Abb. 3-9:

Palettendefinition mit Bewegungsrichtung = 1 (zickzack)

R000693C

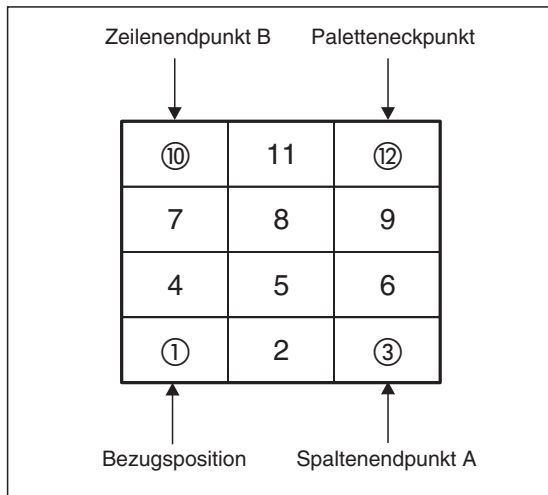


Abb. 3-12:
Palettendefinition mit
Bewegungsrichtung = 2
(Richtung beibehalten)

R000694C

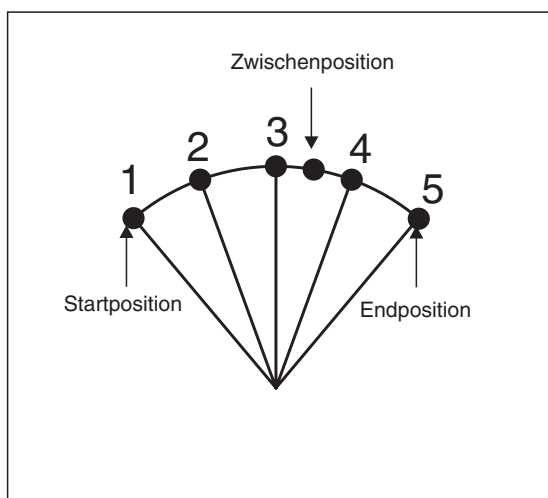
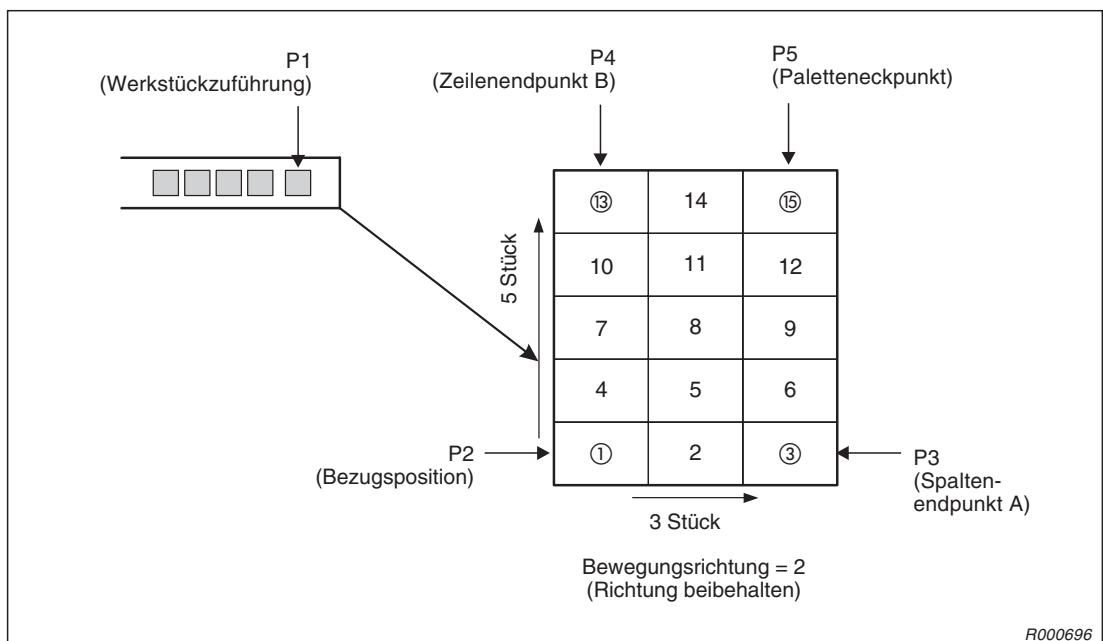


Abb. 3-10:
Palettendefinition mit
Bewegungsrichtung = 3
(kreisförmig)

R000695C

Programmbeispiel



R000696

Abb. 3-11: Palettierung

10	DEF PLT 1, P2, P3, P4, P5, 3, 5, 2	Definiert Palette Nummer 1 mit Bezugsposition = P2, Spaltenendpunkt A = P3, Zeilenendpunkt B = P4, Paletteneckpunkt, der gegenüber der Bezugsposition liegt = P5, Anzahl der Gitterpunkte: 15 (Spalten A = 3, Zeilen B = 5) und einer Bewegungsrichtung = 2
20	M1 = 1	Setzt M1 auf 1 (M1 dient als Zähler)
30	*LOOP	Definiert Marke LOOP
40	MOV P1, -50	Position mittels Gelenk-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der Position 1 entfernt ist (siehe Achtungshinweis)
50	OVRD 50	Legt die Geschwindigkeit auf den halben Maximalwert fest
60	MVS P1	Position 1 mittels Linear-Interpolation anfahren (anfahren der Position zur Werkstückaufnahme)
70	HCLOSE 1	Schließt Hand 1 (Werkstück aufnehmen)
80	DLY 0.5	Wartezeit von 0,5 s
90	OVRD 100	Legt die Geschwindigkeit auf den Maximalwert fest
100	MVS, -50	Position mittels Linear-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der aktuellen Position (P1) entfernt ist (Anheben des Werkstücks) (siehe Achtungshinweis)
110	P10 = (PLT 1, M1)	Berechnet die in der numerischen Variablen M1 festgelegte Position der Palette Nummer 1 und schreibt den Wert in P10
120	MOV P10, -50	Position mittels Gelenk-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der Position 10 entfernt ist (siehe Achtungshinweis)
130	OVRD 50	Legt die Geschwindigkeit auf den halben Maximalwert fest
140	MVS P10	Position 10 mittels Linear-Interpolation anfahren (Anfahren der Position zur Werkstückablage)
150	HOPEN 1	Öffnet Hand 1 (Werkstück ablegen)
160	DLY 0.5	Wartezeit von 0,5 s
170	OVRD 100	Legt die Geschwindigkeit auf den Maximalwert fest
180	MVS, -50	Position mittels Linear-Interpolation anfahren, die 50 mm in Werkzeuglängsrichtung von der aktuellen Position (P10) entfernt ist (Entfernen vom Werkstück) (siehe Achtungshinweis)
190	M1 = M1 + 1	Numerische Variable M1 um 1 erhöhen (Palettenzähler erhöhen)
200	IF M1 <= 15 THEN *LOOP	Ist der Wert der numerischen Variablen M1 kleiner als 15, springe zur Marke LOOP, sonst gehe in die nächste Zeile.
210	END	Programmende

**ACHTUNG:**

Die Richtung des Fahrwegs im Werkzeugkoordinatensystem hängt vom Werkzeugkoordinatensystem des Roboters ab. Detaillierte Informationen zum Werkzeugkoordinatensystem finden Sie im Technischen Handbuch des Roboters.

Befehl steht in Beziehung zu folgenden Funktionen:

Ausdrücke und Operationen ⇒ Abs. 3.7
Verzweigung und Wartezeit ⇒ Abs. 3.4.1

3.4 Programmsteuerung

Der Programmfluss kann über Verzweigungen, Interrupts, Unterprogrammaufrufe, Stoppbefehle usw. gesteuert werden.

3.4.1 Verzweigungen und Wartezeit

Ein Sprung in eine bestimmte Programmzeile kann durch eine unbedingte oder durch eine bedingte Verzweigung erfolgen.

Erläuterung

Befehl	Beschreibung
GOTO	Bewirkt einen unbedingten Sprung in eine festgelegte Zeile
ON GOTO	Bewirkt einen Sprung in Abhängigkeit vom Wert einer Variablen Die Reihenfolge der Sprungziele entspricht der der Integer-Zahlenreihe (0, 1, 2, 3, 4 ...).
IF THEN ELSE	Bewirkt einen Sprung in Abhängigkeit vom Wert einer Variablen und den festgelegten Bedingungen für diesen Wert Die Bedingungen für die Werte können frei gewählt werden. Es darf nur eine Verzweigungsart pro Anweisung verwendet werden.
SELECT CASE END SELECT	Bewirkt einen Sprung in Abhängigkeit vom Wert einer Variablen und den festgelegten Bedingungen für diesen Wert Die Bedingungen für die Werte können frei gewählt werden. Es dürfen mehrere Verzweigungsarten pro Anweisung verwendet werden.
BREAK	Bewirkt einen Sprung in die Zeile nach der END SELECT-Anweisung
WAIT	Bewirkt eine Wartezeit, bis eine Variable den festgelegten Wert erreicht hat

Anweisungsbeispiele

GOTO 200	Unbedingter Sprung in Zeile 200
GOTO *FIN	Unbedingter Sprung zur Marke *FIN
ON M1 GOTO 100, 200, 300	Sprung in Zeile 100, falls der Wert der Variablen M1 = 1 ist, Sprung in Zeile 200, falls M1 = 2 und Sprung in Zeile 300, falls M1 = 3 Entspricht M1 keinem dieser Werte, wird der nächste Programmschritt ausgeführt.
IF M1 = 1 THEN 100	Sprung in Zeile 100, falls M1 = 1, sonst wird das Programm in der nächsten Zeile fortgesetzt
IF M1 = 1 THEN 100 ELSE 200	Sprung in Zeile 100, falls M1 = 1, sonst Zeile 200
SELECT M1 CASE 10 : BREAK	Sprung zur CASE-Anweisung in Abhängigkeit von M1 Ist M1 = 10, wird das Programm nur zwischen CASE 10 und CASE IS 11 ausgeführt. Sprung in die Zeile nach der END SELECT-Anweisung
CASE IS 11 : BREAK	Ist M1 = 11, wird das Programm nur zwischen den CASE IS 11 und CASE IS < 5 ausgeführt. Sprung in die Zeile nach der END SELECT-Anweisung
CASE IS < 5 : BREAK	Ist M1 < 5, wird das Programm nur zwischen CASE IS < 5 und CASE 6 TO 9 ausgeführt. Sprung in die Zeile nach der END SELECT-Anweisung

CASE 6 TO 9	Ist $6 < M1 < 9$, wird das Programm nur zwischen
:	CASE 6 TO 9 und DEFAULT ausgeführt.
BREAK	Sprung in die Zeile nach der END SELECT-Anweisung
DEFAULT	Entspricht M1 keinem der Werte, wird das Programm
:	nur zwischen DEFAULT und END SELECT ausgeführt.
END SELECT	
WAIT M_IN(1) = 1	Wartezeit, bis Eingangsbit 1 eingeschaltet wird

Befehl steht in Beziehung zu folgenden Funktionen:

Programmschleife	⇒	Abs. 3.4.2
Interrupt	⇒	Abs. 3.4.3
Unterprogramm	⇒	Abs. 3.4.4
Eingangssignale	⇒	Abs. 3.5.1

3.4.2 Programmschleife

Bestimmte Programmteile können in Abhängigkeit einer Bedingung wiederholt werden.

Erläuterung

Befehl	Beschreibung
FOR NEXT	Bewirkt eine Wiederholung des Programmteils zwischen der FOR- und NEXT-Anweisung, bis die Abbruchbedingung erfüllt ist
WHILE WEND	Bewirkt eine Wiederholung des Programmteils zwischen der WHILE- und WEND-Anweisung, solange die Ausführungsbedingung erfüllt ist

Anweisungsbeispiele

```
FOR M1 = 1 TO 10
  :
  :
NEXT
```

10-malige Wiederholung des Programmteils zwischen der FOR- und NEXT-Anweisung
Der Startwert der Variablen M1 ist 1. Er wird bei jeder Wiederholung um 1 erhöht.

```
FOR M1 = 0 TO 10 STEP 2
  :
  :
NEXT
```

6-malige Wiederholung des Programmteils zwischen der FOR- und NEXT-Anweisung
Der Startwert der Variablen M1 ist 0. Er wird bei jeder Wiederholung um 2 erhöht.

```
WHILE (M1 >= 1) AND (M1 <= 10)
  :
  :
WEND
```

Wiederholung des Programmteils zwischen der WHILE- und WEND-Anweisung, solange der Wert der numerischen Variablen M1 größer als 1 und kleiner als 10 ist

Befehl steht in Beziehung zu folgenden Funktionen:

```
Verzweigung      => Abs. 3.4.1
Interrupt         => Abs. 3.4.3
Eingangssignale  => Abs. 3.5.1
```

3.4.3 Interrupt

Die Ausführung eines Programms kann mittels eines Interrupts unterbrochen und verzweigt werden.

Erläuterung

Befehl	Beschreibung
DEF ACT	Festlegung des Status und der Ausführung des Interrupts
ACT	Freigeben oder sperren eines Interrupts
RETURN	Bewirkt den Rücksprung aus einer Interrupt-Routine in die Zeile, in der der Interrupt aufgerufen wurde

Anweisungsbeispiele

DEF ACT 1, M_IN(10) = 1 GOSUB 100	Definiert einen Unterprogrammsprung zu Zeile 100 mit der Priorität 1, falls das Eingangssignalbit 10 auf 1 gesetzt wird
DEF ACT 2, M_IN(11) = 1 GOSUB 200, L	Definiert einen Unterprogrammsprung zu Zeile 200 mit der Priorität 2, falls das Eingangssignalbit 11 auf 1 gesetzt wird und die Ausführung der aktuellen Anweisung beendet ist
DEF ACT 3, M_IN(12) = 1 GOSUB 300	Definiert einen Unterprogrammsprung zu Zeile 300 mit der Priorität 3, falls das Eingangssignalbit 12 auf 1 gesetzt wird
ACT 1 = 1	Interrupt 1 freigeben
ACT 2 = 0	Interrupt 2 sperren
RETURN 0	Rücksprung in die Zeile, in der der Interrupt aufgerufen wurde
RETURN 1	Rücksprung in die Zeile, die der Zeile mit dem Interrupt-Aufruf folgt

Befehl steht in Beziehung zu folgenden Funktionen:

Verzweigung	⇒	Abs. 3.4.1
Unterprogramm	⇒	Abs. 3.4.4
Kommunikation	⇒	Abs. 3.6

3.4.4 Unterprogramm

Mit Hilfe von Unterprogrammen und Routinen kann die Anzahl der Schritte im Hauptprogramm reduziert werden. Ein hierarchischer Aufbau und eine bessere Verständlichkeit des Programms sind somit möglich.

Erläuterung

Befehl	Beschreibung
GOSUB	Bewirkt einen Sprung zu einem Unterprogramm, das durch eine festgelegte Zeilennummer oder eine Marke definiert ist
ON GOSUB	Bewirkt einen Sprung zu einem Unterprogramm in Abhängigkeit vom Wert einer Variablen. Die Reihenfolge der Sprungziele entspricht der Integer-Zahlenreihe (0, 1, 2, 3, 4 ...).
RETURN	Bewirkt den Rücksprung aus einer Interrupt-Routine in die Zeile, die der Zeile folgt, aus der der Unterprogrammaufruf mit dem GOSUB-Befehl erfolgte
CALLP	Bewirkt den Aufruf eines Programms. Wird im aufgerufenen Programm die END-Anweisung ausgeführt, erfolgt der Rücksprung in die Zeile des aufrufenden Programms, die der Zeile folgt, aus der der Programmaufruf mit dem CALLP-Befehl erfolgte. Beim Programmaufruf können Daten übergeben werden.
FPRM	Legt die Daten fest, die beim Aufruf eines Programms mit dem CALLP-Befehl übergeben werden

Anweisungsbeispiele

GOSUB 100	Springt zum Unterprogramm in Zeile 100
GOSUB *GET	Springt zum Unterprogramm mit der Marke GET
ON M1 GOSUB 100, 200, 300	Springt zum Unterprogramm in Zeile 100, falls M1 = 1 ist, springt zum Unterprogramm in Zeile 200, falls M1 = 2 ist und springt zum Unterprogramm in Zeile 300, falls M3 = 3 ist Entspricht M3 keinem der Werte, wird der nächste Programmschritt ausgeführt.
RETURN	Rücksprung in die Zeile, die dem Unterprogrammaufruf mit dem GOSUB-Befehl folgt
CALLP "10"	Aufruf des Programms Nummer 10
CALLP "20", M1, P1	Aufruf des Programms Nummer 20 und Übergabe der numerischen Variablen M1 und der Positionsvariablen P1
FPRM M10, P10	Festlegung der numerischen Variablen M10 und der Positionsvariablen P10, die bei Aufruf des Unterprogramms mit CALLP übernommen werden

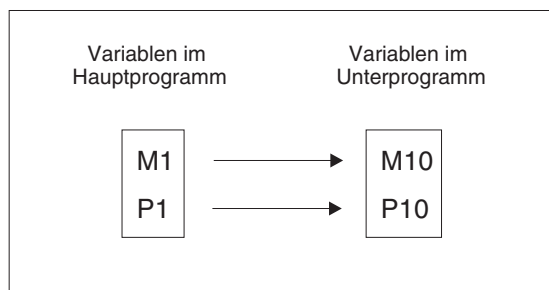


Abb. 3-13:
Übergabe der numerischen Variablen und der Positionsvariablen

Befehl steht in Beziehung zu folgenden Funktionen:

Interrupt ⇒ Abs. 3.4.3
Kommunikation ⇒ Abs. 3.6
Verzweigung ⇒ Abs. 3.4.1

3.4.5 Timer

Die Funktion ermöglicht die Festlegung eines Wartestatus. Bei Verwendung des DLY-Befehls mit einem Impulsausgang kann die Impulsdauer eingestellt werden.

Erläuterung

Befehl	Beschreibung
DLY	Festlegung von Wartezeit und Impulsdauer

Anweisungsbeispiele

DLY 0.05 Wartezeit von 0,05 s
M_OUT(10) = 1 DLY 0.5 Ausgangsbitt 10 für 0,5 s einschalten

Befehl steht in Beziehung zu folgenden Funktionen:

Impulsausgang ⇒ Abs. 3.5.2

3.4.6 Stopp

Durch den HLT-Befehl wird der Programmablauf unterbrochen und der Roboter bis zum Stillstand abgebremst.

Erläuterung

Befehl	Beschreibung
HLT	Bewirkt eine Unterbrechung des Programmablaufs und ein Stoppen des Roboters

Anweisungsbeispiele

HLT	Programmablauf und Roboterbewegung stoppen
IF M_IN(20) = 1 THEN HLT	Programmablauf und Roboterbewegung stoppen, wenn Eingangsbit 20 gleich 1 ist
MOV P1 WTHIF M_IN(20) = 1, HLT	Programmablauf und Roboterbewegung werden gestoppt, wenn bei Anfahrt der Position P1 das Eingangsbit 18 gleich 1 ist.

Befehl steht in Beziehung zu folgenden Funktionen:

Angehängte Anweisung ⇒ Abs. 3.8

3.5 Ein- und Ausgabe externer Signale

3.5.1 Eingangssignale

Die Steuergeräte CR1/CR2 können Signale von externen Geräten, z. B. einer SPS, empfangen und verarbeiten. Ein Einlesen der Eingangssignale erfolgt über die Roboterstatusvariablen M_IN() usw. (siehe auch Abs. 8.1.11).

Erläuterung

Befehl	Beschreibung
WAIT	Bewirkt eine Programmunterbrechung, bis die festgelegte Eingangsbedingung erfüllt ist

Anweisungsbeispiele

WAIT M_IN(1) = 1
M1 = M_INB(20)

Wartet, bis das Eingangsbit 1 eingeschaltet wird
Schreibt die Eingangssignalbits 20 bis 27 als
8-Bit-Wort in die numerische Variable M1

M1 = M_INW(5)

Schreibt die Eingangssignalbits 5 bis 20 als
16-Bit-Wort in die numerische Variable M1

Befehl steht in Beziehung zu folgenden Funktionen:

Ausgangssignale ⇒ Abs. 3.5.2
Verzweigung ⇒ Abs. 3.4.1
Interrupt ⇒ Abs. 3.4.3

3.5.2 Ausgangssignale

Die Steuergeräte CR1/CR2 können Signale an externe Geräte, z. B. eine SPS, ausgeben. Eine Ausgabe der Ausgangssignale erfolgt über die Roboterstatusvariablen M_OUT() usw. (siehe auch Abs. 8.1.11).

Erläuterung

Befehl	Beschreibung
CLR	Bewirkt ein Zurücksetzen der allgemeinen Ausgangssignale auf das über die Parameter ORST0 bis ORST224 vorgegebene Bitmuster

Anweisungsbeispiele

CLR 1	Zurücksetzen der Ausgänge auf das vorgegebene Bitmuster
M_OUT(1) = 1	Ausgangsbit 1 einschalten
M_OUTB(8) = 0	8 Bits, von Ausgangsbit 8 bis 15, ausschalten
M_OUTW(20) = 0	16 Bits, von Ausgangsbit 20 bis 35, ausschalten
M_OUT(1) = 1 DLY 0.5	Ausgangsbit 1 für 0,5 s einschalten (Impulsausgang)
M_OUTB(10) = &H0F	4 Bits, von Ausgangsbit 10 bis 13 einschalten und 4 Bits von Ausgangsbit 14 bis 17 ausschalten

Befehl steht in Beziehung zu folgenden Funktionen:

Eingangssignale	⇒	Abs. 3.5.1
Timer	⇒	Abs. 3.4.5

3.6 Kommunikation

Die Kommunikationsfunktionen ermöglichen einen Datenaustausch zwischen dem Steuergerät und externen Geräten (z. B. Personalcomputer).

Erläuterung

Befehl	Beschreibung
OPEN	Öffnet eine Kommunikationsleitung
CLOSE	Schließt eine geöffnete Kommunikationsleitung
PRINT #	Ausgabe von Daten im ASCII-Format Nach jeder PRINT-Anweisung wird ein „Carriage Return“ ausgeführt.
INPUT #	Eingabe von Daten im ASCII-Format Nach jeder INPUT-Anweisung wird ein „Carriage Return“ ausgeführt.
ON COM GOSUB	Legt den Sprung in ein Unterprogramm fest, wenn ein Interrupt von einer Kommunikationsleitung anliegt Ein Interrupt erfolgt durch Eingabe von Daten über ein externes Gerät.
COM ON	Gibt die Interrupts von Kommunikationsleitungen frei
COM OFF	Sperrt die Interrupts von Kommunikationsleitungen Ein generierter Interrupt bleibt wirkungslos.
COM STOP	Der Interrupt-Prozess von Kommunikationsleitungen wird unterbrochen. Ein generierter Interrupt wird gespeichert und nach Freigabe der Kommunikationsleitung abgearbeitet.

Anweisungsbeispiele

OPEN "COM1:" AS#1	Öffnet die RS232C-Schnittstelle als Datei Nr. 1
CLOSE #1	Schließt die Datei Nr. 1
CLOSE	Schließt alle geöffneten Dateien
PRINT #1, "TEST"	Zeichenkette „TEST“ an Schnittstelle ausgeben
PRINT #2, "M ="; M1	Zeichenkette „M =“ an Schnittstelle und M1 an Datei Nr. 2 ausgeben
	Ausgabe: „M1 = 1“ + CR + LF (wenn M1 = 1)
PRINT #3, P1	Positionsvariable P1 an Datei Nr. 3 ausgeben
	Ausgabe: „(123.7, 238.9, 33.1, 19.3, 0, 0)(1, 0)“ + CR + LF (wenn X = 123.7, Y = 238.9, Z = 33.1, A = 19.3, B = 0, C = 0, FL1 = 1, FL2 = 0)
PRINT #1, M5, P5	Numerische Variable M5 und Positionsvariable P5 an Datei Nr. 1 ausgeben
	M5 und P5 werden durch ein Komma getrennt (hexadezimal, 2C).
	Ausgabe: „8, (123.7, 238.9, 33.1, 19.3, 0, 0)(1, 0)“ + CR + LF (wenn M5 = 8, P5: X = 123.7, Y = 238.9, Z = 33.1, A = 19.3, B = 0, C = 0, FL1 = 1, FL2 = 0)
INPUT #1, M3	Wandelt die Eingangsdaten in einen Wert um und schreibt diesen in die numerische Variable M3
	Eingabe: „8“ + CR + LF (bei Eingabe von 8)
INPUT #1, P10	Wandelt die Eingangsdaten in Werte um und schreibt diese in die Positionsvariable P10
	Eingabe: „(123.7, 238.9, 33.1, 19.3, 0, 0)(1, 0)“ + CR + LF (wenn P5: X = 123.7, Y = 238.9, Z = 33.1, A = 19.3, B = 0, C = 0, FL1 = 1, FL2 = 0)

INPUT #1, M8, P6	Schreibt die ersten eingegeben Daten in die numerische Variable M8 Die folgenden Daten werden in die Positionsvariable P6 geschrieben. M8 und P6 werden durch ein Komma getrennt (hexadezimal, 2C). Eingabe: „7, (123.7, 238.9, 33.1, 19.3, 0, 0)(1, 0)“ + CR + LF (wenn M8 = 7, P6: X = 123.7, Y = 238.9, Z = 33.1, A = 19.3, B = 0, C = 0, FL1 = 1, FL2 = 0)
ON COM(1) GOSUB 300	Springt zum Unterprogramm in Zeile 300, falls über COM1 eine Dateneingabe erfolgt
ON COM(2) GOSUB *RECV	Springt zur Marke *RECV, falls über COM2 eine Dateneingabe erfolgt
COM(1) ON	Freigabe des Interrupts von COM1
COM(2) OFF	Sperren des Interrupts von COM2
COM(1) STOP	Stoppt den Interrupt von COM1

Befehl steht in Beziehung zu folgenden Funktionen:

Unterprogramm	⇒	Abs. 3.4.4
Interrupt	⇒	Abs. 3.4.3

3.7 Ausdrücke und Operationen

Folgende Tabelle zeigt die in MELFA-BASIC IV möglichen Operationen, deren Verwendung und Anwendungsbeispiele:

Operation	Operator	Bedeutung	Beispiel	
Substitution	=	Der rechte Operand wird in den linken geschrieben	P1 = P2 P5 = P_CURR P10.Z = 100.0 M1 = 1 STS\$ = "OK"	Schreibe P2 in P1. Schreibe die aktuelle Position in P5. Setze die Z-Koordinate von P10 auf 100.0. Schreibe den Wert 1 in die numerische Variable M1. Schreibe die Zeichenkette „OK“ in die Zeichenkettenvariable STS.
Arithmetische Operation	+	Addition	P10 = P1 + P2 MOV P8 + P9 M1 = M1 + 1 STS\$ = "ERR" + "001"	Schreibe das Ergebnis der Addition von P1 und P5 in P10. Fahre die Position an, die sich aus der Summe von P8 und P9 ergibt. Addiere 1 zu der numerischen Variablen M1. Addiere die Zeichenketten „ERR“ und „001“ und schreibe das Ergebnis in die Zeichenkettenvariable STS.
	-	Subtraktion	P10 = P1 - P2 MOV P8 - P9 M1 = M1 - 1	Schreibe das Ergebnis der Subtraktion von P1 minus P2 in P10. Fahre die Position an, die sich aus der Differenz von P8 und P9 ergibt. Subtrahiere 1 von der numerischen Variablen M1.
	*	Multiplikation	P1 = P10 * P3 M1 = M1 * 5	Schreibe das Ergebnis der relativen Konvertierung von P10 und P3 in P1. Multipliziere die numerische Variable M1 mit 5.
	/	Division	P1 = P10 / P3 M1 = M1 / 2	Schreibe das Ergebnis der umgekehrten relativen Konvertierung von P10 und P3 in P1. Dividiere die numerische Variable M1 durch 2.
	^	Exponential-Funktion	M1 = M1 ^ 2	Quadriere die numerische Variable M1.
	¥	Integer-Division	M1 = M1 ¥ 3	Dividiere die numerische Variable durch 3 und runde das Ergebnis ab.
	MOD	Modulo-Arithmetik	M1 = M1 MOD 3	Dividiere die numerische Variable M1 durch 3 und schreibe den Rest in M1.
	-	Vorzeichenumkehr	P1 = - P1 M1 = - M1	Kehre das Vorzeichen jeder Koordinate der Positionsvariablen P1 um. Kehre das Vorzeichen der numerischen Variablen M1 um.

Tab. 3-1: Ausdrücke und Operationen (1)

Operation	Operator	Bedeutung	Beispiel	
Vergleichs- operation	=	Gleich	IF M1 = 1 THEN 200 IF STS\$ = "OK" THEN 100	Springe zu Zeile 200, falls die numerische Variable M1 gleich 1 ist. Springe zu Zeile 100, falls die Zeichenkettenvariable STS\$ „OK“ ist.
	<> oder ><	Ungleich	IF M1 <> 2 THEN 300 IF STS\$ <> "OK" THEN 900	Springe zu Zeile 300, falls die numerische Variable M1 ungleich 2 ist. Springe zu Zeile 900, falls die Zeichenkettenvariable STS\$ ungleich „OK“ ist.
	<	Kleiner als	IF M1 < 10 THEN 300 IF STS\$ < 3 THEN 100	Springe zu Zeile 300, falls die numerische Variable M1 kleiner als 10 ist. Springe zu Zeile 100, falls die Anzahl der Zeichen der Zeichenkette STS\$ kleiner als 3 ist.
	>	Größer als	IF M1 > 9 THEN 200 IF STS\$ > 2 THEN 300	Springe zu Zeile 200, falls die numerische Variable M1 größer als 9 ist. Springe zu Zeile 300, falls die Anzahl der Zeichen der Zeichenkette STS\$ größer als 2 ist.
	=< oder <=	Kleiner oder gleich	IF M1 <= 10 THEN 200 IF STS\$ <= 5 THEN 300	Springe zu Zeile 200, falls die numerische Variable M1 kleiner oder gleich 10 ist. Springe zu Zeile 300, falls die Anzahl der Zeichen der Zeichenkette STS\$ kleiner oder gleich 5 ist.
	=> oder >=	Größer oder gleich	IF M1 >= 11 THEN 200 IF STS\$ >= 6 THEN 300	Springe zu Zeile 200, falls die numerische Variable M1 größer oder gleich 11 ist. Springe zu Zeile 300, falls die Anzahl der Zeichen der Zeichenkette STS\$ größer oder gleich 6 ist.
Logische Operation	AND	Logisches UND	M1 = M_INB(1) AND &H0F	Konvertiere die Eingangsbits 1 bis 4 und schreibe das Ergebnis in die numerische Variable M1 (Eingangsbits 5 bis 8 bleiben AUS).
	OR	Logisches ODER	M_OUTB(20) = M1 OR &H80	Ausgabe der numerischen Variablen M1 an Ausgangsbits 20 bis 27 (Ausgangsbit 27 ist dabei immer EIN)
	NOT	Negation	M1 = NOT M_INW(1)	Negiere die Eingangsbits 1 bis 16 und schreibe den Wert in die numerische Variable M1.
	XOR	Exklusives ODER	M2 = M1 XOR M_INW(1)	Schreibe das Ergebnis der exklusiven ODER-Verknüpfung von M1 und den Eingangsbits 1 bis 16 in die numerische Variable M2.
	<<	Logische Linksverschiebung	M1 = M1 << 2	Verschiebe die numerische Variable M1 2 Bits nach links.
	>>	Logische Rechtsverschiebung	M1 = M1 >> 1	Verschiebe die numerische Variable M1 1 Bit nach rechts.

Tab. 3-1: Ausdrücke und Operationen (2)

3.8 Angehängte Anweisung

Bei Interpolationsbefehlen ist es möglich, eine Verknüpfung an die Anweisung anzuhängen. Durch Anhängen einer Verknüpfung können bestimmte Befehle parallel zum Interpolationsbefehl ausgeführt werden.

Erläuterung

Befehl	Beschreibung
WTH	Während einer Interpolationsbewegung wird eine unbedingte, zusätzliche Anweisung ausgeführt.
WTHIF	Während einer Interpolationsbewegung wird eine bedingte, zusätzliche Anweisung ausgeführt.

Anweisungsbeispiele

MOV P1 WTH M_OUT(20) = 1	Position 1 anfahren und Ausgangsbit 20 auf 1 setzen
MOV P1 WTH HOPEN 1	Position 1 anfahren und Hand 1 öffnen
MOV P1 WTHIF M_IN(20) = 1, HLT	Stoppt, falls während der Anfahrt von P1 Eingangsbit 20 auf 1 gesetzt wird
MOV P1 WTHIF M_IN(19) = 1, SKIP	Stoppt, falls während der Anfahrt von P1 Eingangsbit 19 auf 1 gesetzt wird und springt in die nächste Zeile

Befehl steht in Beziehung zu folgenden Funktionen:

Gelenk-Interpolation	⇒	Abs. 3.2.1
Linear-Interpolation	⇒	Abs. 3.2.2
Kreis-Interpolation	⇒	Abs. 3.2.3
Stopp	⇒	Abs. 3.4.6

4 Multitask-Funktion

4.1 Beschreibung

Die Multitask-Funktion ermöglicht die parallele Ausführung mehrerer Programme zur Verkürzung der Taktzeiten. Der Roboter kann neben seiner Bewegung weitere Funktionen ausführen und mit der Peripherie kommunizieren, z. B. um Signale weiterzugeben.

Beim Multitasking wird jedem Programm ein Programmplatz (Slot/Task) zugeordnet. Die Definition der Programmplätze erfolgt über die Zuweisung von Programmnamen, Format, Startbedingungen und Priorität eines Programms in den Programmplatzparametern SLT1 bis SLT32.

Die Ausführung des Multitaskings kann über ein Startsignal, einen speziellen Eingang oder über einen auf das Multitasking bezogenen Befehl erfolgen. Insgesamt ist die parallele Ausführung von 32 Programmen möglich.

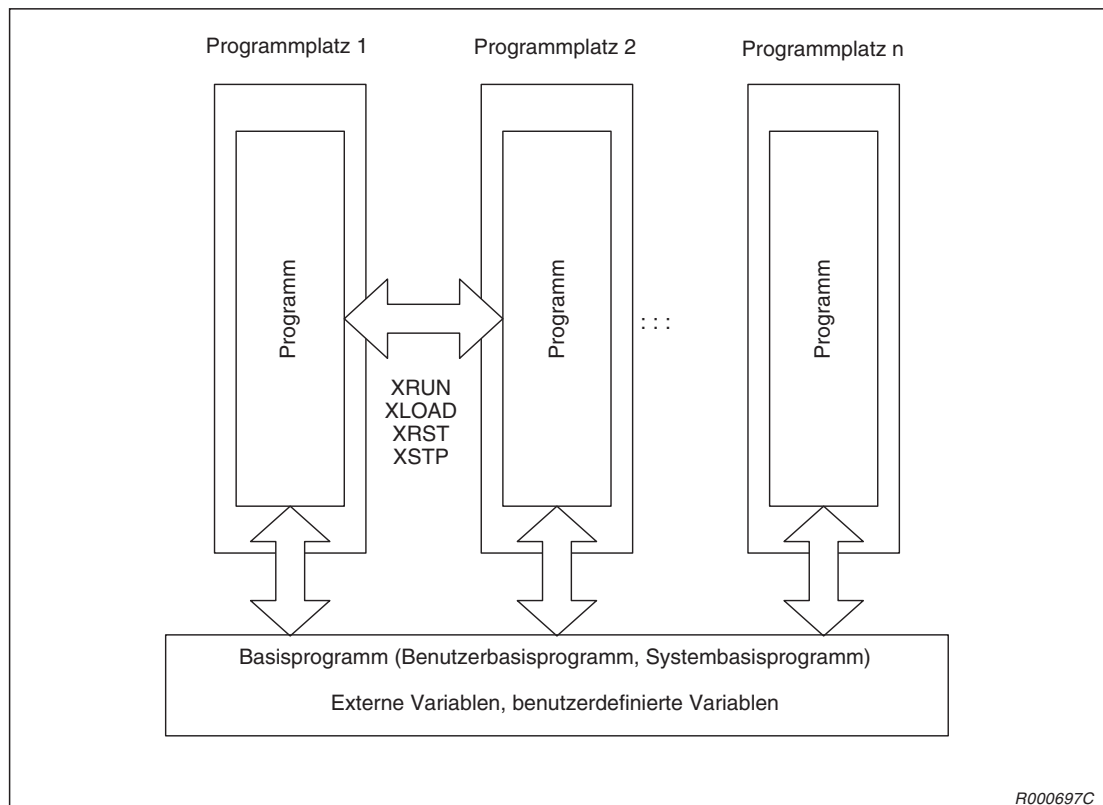


Abb. 4-1: Multitasking

HINWEIS

Bei Ausführung eines Programmes wird das Programm einem Programmplatz (Slot/Task) zugeordnet und gestartet. Bei Aufruf eines Programmes über das Bedienfeld des Steuergerätes wird das Programm vom Steuergerät automatisch dem Programmplatz 1 zugeordnet.

4.2 Betriebszustand eines Programmplatzes

Der Betriebszustand eines Programmplatzes ist von den ausgeführten Operationen und Befehlen abhängig. Jeder Zustand kann über eine Roboterstatusvariable oder ein externes Ausgangssignal angezeigt werden.

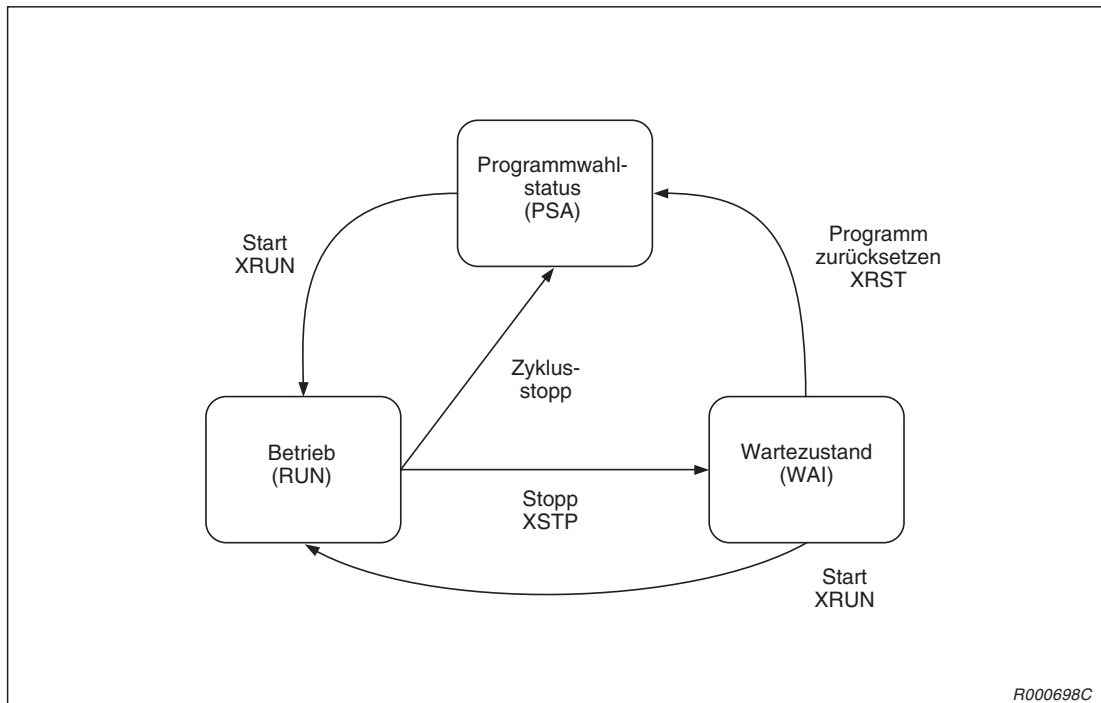


Abb. 4-2: Betriebszustand eines Programmplatzes

4.3 Ausführung eines Multitasks

Zur Ausführung eines Multitasks stehen zwei Möglichkeiten zur Verfügung:

- **Ausführung aus einem Programm**
Bei dieser Methode wird die parallele Ausführung von Programmen aus einer nicht festgelegten Position heraus über einen MELFA-BASIC-IV-Befehl gestartet. Dabei können die Programme, die parallel ausgeführt oder die Programme, die bei einer parallelen Verarbeitung gestoppt werden sollen, gewählt werden.
Diese Methode ist sinnvoll, wenn die Auswahl der Programme aus dem Programmfluss heraus erfolgen soll. Bei dieser Methode werden die Befehle XLOAD, XRUN, XSTP und XRST verwendet. Eine detaillierte Beschreibung der Befehle finden Sie in Abs. 9.3.
- **Ausführung über das Steuergerät oder ein externes Ein-/Ausgangssignal**
Bei dieser Methode wird die parallele Ausführung von Programmen im Startbetrieb, als durchgehender paralleler Betrieb oder als paralleler Betrieb bei Auftreten einer Fehlermeldung in Abhängigkeit der Programmplatzparameter gestartet.
Diese Methode ist vom Programmfluss unabhängig und sinnvoll für eine gleichzeitige Ausführung mit voreingestelltem Format oder eine sequentielle Ausführung.

4.3.1 Programmplatzparameter

Mit Hilfe der Programmplatzparameter SLT1 bis SLT 32 können der Programmname, das Format, die Startbedingung und die Priorität für 32 Programmplätze festgelegt werden.

Die Festlegung erfolgt in der Reihenfolge:

Parametername = 1. Programmname, 2. Ausführungsformat, 3. Startbedingung, 4. Priorität

Programmplatzparameter	Funktion	Einstellung
Programmname	Festlegung des Programmnamens der einem Programmplatz zugewiesen wird Das Programm dieser Anwendung wird nach Einschalten der Versorgungsspannung oder bei Initialisierung des Programmplatzes gültig.	Einstellwert = Programmname
Ausführungsformat	Festlegung eines kontinuierlichen oder zyklischen Betriebs	Einstellwert = REP: kontinuierlicher Betrieb Bei Ausführung der END-Anweisung springt das Programm an den Anfang zurück und wird erneut ausgeführt.
		Einstellwert = CYC: zyklischer Betrieb Bei Ausführung der END-Anweisung wird das Programm beendet.
Startbedingung	Festlegung, ob das Programm bei einer Startanforderung, ständig oder bei Auftreten eines Fehlers gestartet werden soll	Einstellwert = START ^① : Startanforderung Die Programmausführung wird über das Bedienfeld des Steuergerätes oder ein externes Ein-/Ausgangssignal gestartet.
		Einstellwert = ALWAYS ^② : ständig Die Programmausführung wird automatisch mit dem Einschalten der Versorgungsspannung gestartet.
		Einstellwert = ERROR ^③ : Fehler Die Programmausführung wird bei Auftreten eines Fehlers gestartet.
Priorität	Festlegung der Priorität der Ausführung	Einstellwert = 1 bis 31 Je größer der Wert, desto höher die Priorität.

Tab. 4-1: Programmplatzparameter

- ① Beim Start über das Steuergerät oder über das spezielle Eingangssignal START werden die Programme aller Programmplätze, für die ein Starten über Startanforderung definiert wurde, gestartet.
Ein unabhängiger Programmstart erfolgt über die Startsignale S1START bis S32START. In diesem Fall ist die Zeilennummer dem Ein-/Ausgangsparameter zugewiesen. Eine detaillierte Beschreibung der speziellen Parameter für die Ein-/Ausgänge finden Sie in Abs. 6.2.1.
- ② Bei einem kontinuierlichen Betrieb über die Einstellung „ALWAYS“ können keine Befehle, die den Roboter steuern, z. B. der MOV-Befehl, ausgeführt werden.
Ein Programm, das mit der Einstellung „ALWAYS“ ausgeführt wird, kann über den Befehl XSTP gestoppt werden. Ein Stoppen des Programms über das Bedienfeld des Steuergerätes, ein externes Eingangssignal oder über den NOT-HALT-Schalter ist nicht möglich.
- ③ Ist ein Betriebsstart nach Auftreten einer Fehlermeldung über die Einstellung „ERROR“ gewählt, können keine Befehle, die den Roboter steuern, z. B. der MOV-Befehl, ausgeführt werden.
Das Ausführungsformat wird ohne Berücksichtigung der Einstellung auf „zyklisch“ gesetzt.

Beispiel ▾

Dieses Beispiel zeigt die Parametereinstellungen zur Festlegung der folgenden Bedingungen für Programmplatz 2:

Programmnamen:	5
Ausführungsformat:	kontinuierlicher Betrieb
Startbedingung :	ständig
Priorität:	10

Einstellung: SLT 2 = 5, REP, ALWAYS, 10



4.4 Erstellung eines Multitask-Programms

Anzahl der Programme, die parallel ausgeführt werden sollen

Die Einstellung der Anzahl der Programme, die parallel ausgeführt werden sollen, erfolgt über den Parameter TASKMAX (Werkseinstellung: 8). Sollen mehr als 8 Programme parallel ausgeführt werden, muss der Parameterwert geändert werden.

Datenaustausch zwischen Programmen über externe Variablen

Über die externen Variablen wie M_00 oder P_00 kann ein Datenaustausch zwischen den im Multitasking betriebenen Programmen stattfinden. Eine detaillierte Beschreibung der externen Variablen finden Sie in Abs. 8.1.11.

Beispiel ▾

Dieses Beispiel zeigt eine Steuerung des EIN/AUS-Zustandes des Eingangsbits 8 über Programmplatz 2. Der EIN-Zustand wird über die externe Variable M_00 an Programmplatz 1 übertragen.

Programmplatz 1 (Slot 1)

10	M_00 = 0	Setzt die Variable M_00 auf 0
20	IF M_00 = 0 THEN 20	Wartestatus, bis M_00 ungleich 0
30	M_00 = 0	Setzt die Variable M_00 auf 0
40	MOV P1	Fortsetzung des normalen Betriebs
50	MOV P2	
	:	
100	GOTO 20	Wiederhole ab Zeile 20

Programmplatz 2 (Slot 2)

10	IF M_IN(8) <> 1 THEN 30	Sprung in Zeile 30, falls Eingangsbit 8 nicht EIN ist
30	M_00 = 1	Setzt die Variable M_00 auf 1
40	MOV P1	Fortsetzung des normalen Betriebs
	:	

△

Überwachung des Programmstatus über Roboterstatusvariablen

Über die Roboterstatusvariablen (M_RUN, M_WAI und M_ERR) kann der Programmstatus einer im Multitasking betriebenen Anwendung von jedem Programmplatz aus überwacht werden.

Beispiel ▾

M1 = M_RUN(2) Schreibt Programmstatus des Programmplatzes 2 in M1

Eine detaillierte Beschreibung der Roboterstatusvariablen finden Sie im Abs. 8.1.11.

△

Externe Signalein- und -ausgänge

Die externen Signalein- und -ausgänge können von jedem Programmplatz verwendet werden.

Programmplatz 1

Das Hauptsteuerprogramm, in dem die Bewegungsbefehle des Roboters (MOV-Befehle usw.) festgelegt sind, wird in der Regel Programmplatz 1 zugeordnet. Eine andere Zuordnung muss über die Befehle GETM und RELM erfolgen. Eine detaillierte Beschreibung der Befehle finden Sie in Abs. 9.3.

HINWEIS

In der Grundeinstellung wird der Roboter 1 (Roboter bei Standardsystem) dem Programmplatz 1 zugeordnet. Bewegungsbefehle können somit ohne Zuordnung über den GETM-Befehl im Programmplatz 1 definiert werden. Sollen Bewegungsbefehle in einer Anwendung ausgeführt werden, so ist zuerst die Zuordnung von Programmplatz und Roboter über den Befehl RELM aufzuheben und anschließend die neu gewünschte Zuordnung über den Befehl GETM zu definieren.

4.5 Anwendung des Multitaskings

Multitasking starten

Bei Ausführung des Startvorgangs über das Bedienfeld der Steuereinheit oder den speziellen Eingang START starten alle Anwendungen, für die in den Programmplatzparametern die Startbedingung „Startanforderung“ festgelegt wurde, gleichzeitig. Ein separates Starten der Programme ist über die Starteingänge S1START bis S32START möglich. In diesem Fall ist die Zeilennummer dem Ein-/Ausgangsparameter zugewiesen. Eine detaillierte Beschreibung der speziellen Parameter für die Ein-/Ausgänge finden Sie in Abs. 6.2.1.

Anzeige des Betriebszustandes

Die LEDs der Taster START und STOP auf dem Bedienfeld der Steuereinheit und die speziellen Ein-/Ausgangssignale START und STOP zeigen den Betriebszustand aller Programme an. Bei Ausführung eines Programmes leuchtet die LED des START-Tasters und das Ausgangssignal START wird eingeschaltet.

Die speziellen Ausgangssignale S1START bis S32START und S1STOP bis S32STOP zeigen den Betriebszustand jedes einzelnen Programmplatzes an. Zur unabhängigen Anzeige der Betriebszustände muss die entsprechende Zeilennummer dem Ein-/Ausgangsparameter zugewiesen werden. Eine detaillierte Beschreibung der speziellen Parameter für die Ein-/Ausgänge finden Sie in Abs. 6.2.1.

4.6 Beispiel zur Anwendung der Multitask-Funktion

Detaillierte Beschreibung des Arbeitsablaufs

Der Arbeitsablauf wird in zwei Programme aufgeteilt:

- Programm mit Bewegungsbefehlen
Dem Programm, das die Bewegungsbefehle enthält, ist der Programmplatz 1 zugewiesen.
- Programm zum Einlesen von Positionsdaten
Dem Programm zum Einlesen der Positionsdaten ist der Programmplatz 2 zugewiesen. Wird während der Roboterbewegung ein Startsignal an einen Sensor ausgegeben, erfolgt über das Programm zum Einlesen von Positionsdaten eine Datenabfrage des Personalcomputers. Der Personalcomputer überträgt die Positionsdaten über Programmplatz 2 zum Roboter.

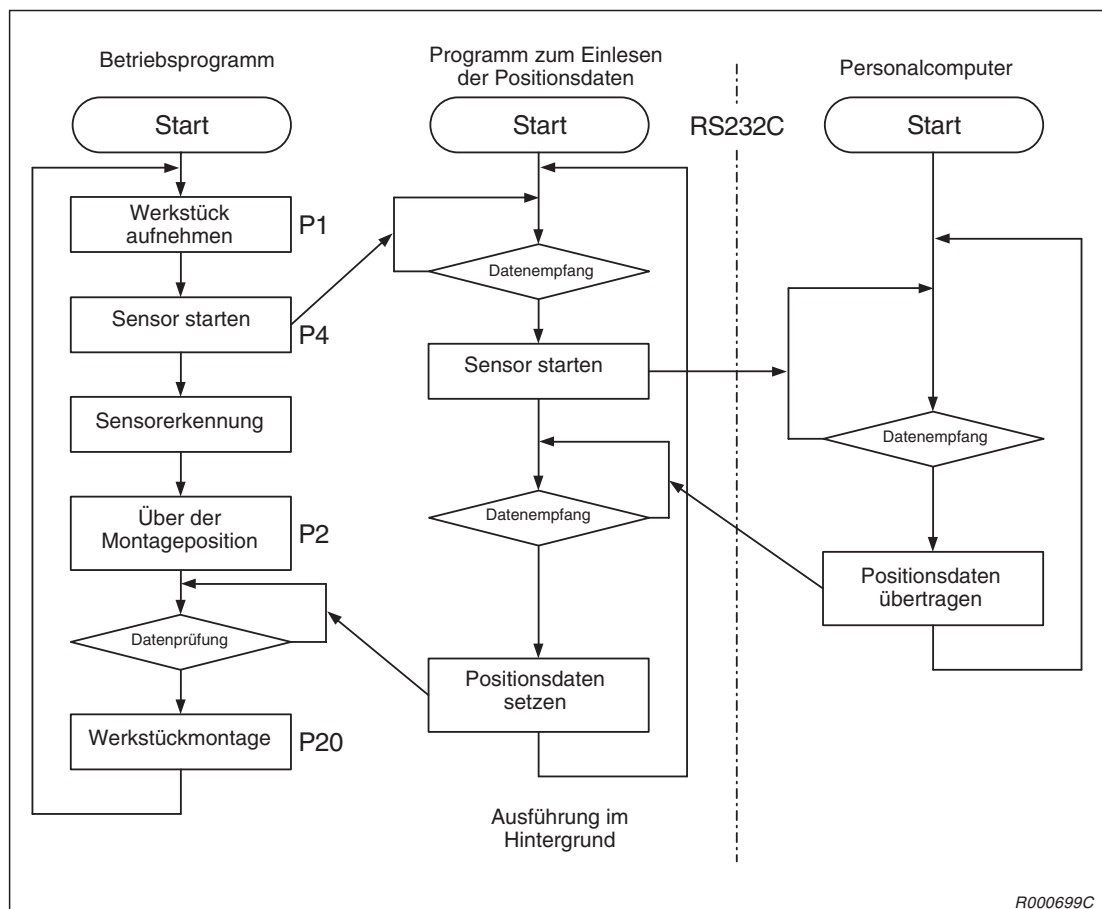


Abb. 4-3: Flussdiagramm

Positionen

P1: Aufnahme des Werkstücks (Wartezeit Vakuumgreifer DLY 0.05)

P2: Ablage des Werkstücks (Wartezeit DLY 0.05)

P3: Position vor der Überwachung (Position ohne Stopp durchlaufen CNT)

P4: Position nach der Überwachung (Position ohne Stopp durchlaufen CNT)

P_01: Kompensationsdaten der Überwachung

P20: Relative Konvertierung der Kompensationsdaten der Überwachung P_01 und P2

Bewegungsablauf

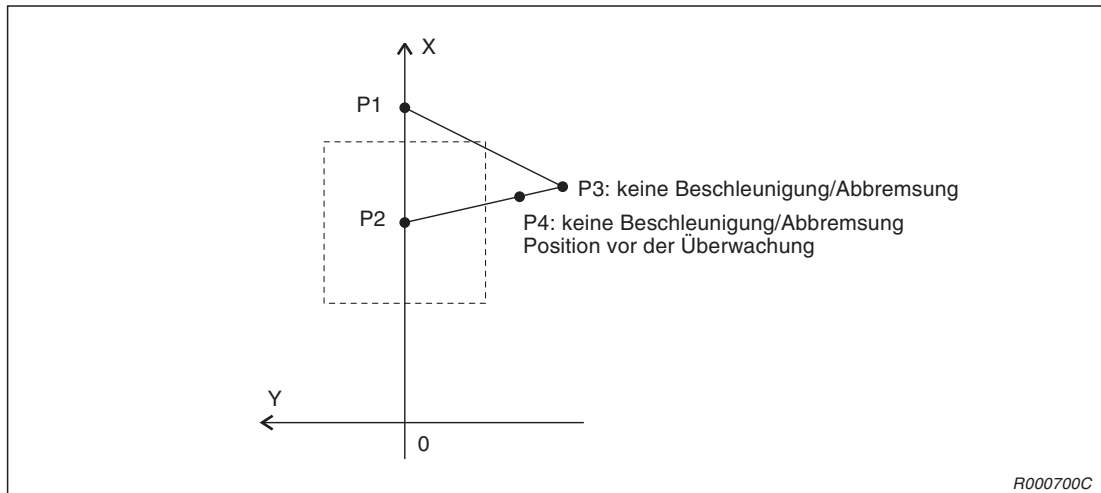


Abb. 4-4: Verfahrbewegung und Positionen

Programm mit Bewegungsbefehlen (Programmplatz 1)

100 CNT 1	'Überschleiffunktion freigeben
110 MOV P2, 10	'Position 10 mm über P2 anfahren
120 MOV P1, 10	'Position 10 mm über P1 anfahren
130 MOV P1	'Aufnahmeposition P1 anfahren
135 M_OUT(10) = 0	'Werkstück aufnehmen
140 DLY 0.05	'Wartezeit 0,05 s
150 MOV P1, 10	'Position 10 mm über P1 anfahren
160 MOV P3	'Position P3 anfahren
165 SPD 500	'Geschwindigkeit auf 500 mm/s setzen
170 MVS P4	'Mit Überfahren von P4 Überwachung starten
180 M_02# = 0	'Hintergrundprozess zum Einlesen der Daten mit Sperrvariablen (M_01 = 1/M_02 = 0) starten
190 M_01# = 1	'Einlesen der Daten im Hintergrund starten
200 MVS P2, 10	'Position 10 mm über P2 anfahren
210 IF M_02# = 0 THEN GOTO 210	'Warten, bis Sperrvariable M_02 gleich 1
220 P20 = P2 * P_01	'Addiere P_01 zu P20
230 MOV P20, 10	'Position 10 mm über P20 anfahren
240 MOV P20	'Ablageposition P20 anfahren
245 M_OUT(10) = 1	'Werkstück ablegen
250 DLY 0.05	'Wartezeit 0,05 s
260 MOV P20, 10	'Position 10 mm über P20 anfahren
270 CNT 0	'Überschleiffunktion sperren
280 END	'Zyklusende

Programm zum Einlesen von Positionsdaten (Programmplatz 2)

100 IF M_01# = 0 THEN GOTO 100	'Warten, bis Sperrvariable M_01 gleich 1
105 OPEN "COM1:" AS #1	'Öffnet die RS232C-Kommunikationsschnittstelle
110 DLY M_03#	'Hypothetische Wartezeit (0,05 s)
115 PRINT #1, "SENS"	'Zeichenkette „SENS“ über RS232C-Schnittstelle ausgeben (Anzeige)
117 INPUT #1, M1, M2, M3	'Wartet auf Einlesen der Kompensationsdaten (relative Daten)
120 P_01.X = M1	'Überschreiben der ΔX -Koordinate
130 P_01.Y = M2	'Überschreiben der ΔY -Koordinate
140 P_01.Z = 0.0	'
150 P_01.A = 0.0	'
160 P_01.B = 0.0	'
170 P_01.C = RAD(M3)	'Überschreiben der ΔC -Koordinate
175 CLOSE	'Kommunikationsschnittstelle schließen
180 M_01# = 0	'Setzen der Sperrvariablen M_01 = 0
190 M_02# = 1	'Setzen der Sperrvariablen M_02 = 1
200 END	'Programm beenden

Einstellung der Programmplatzparameter

SLT 1 = 1, REP, START, 1
 SLT 2 = 2, REP, START, 2

Zur Aktivierung der Programmplatzparameter muss die Versorgungsspannung aus und wieder eingeschaltet werden.

Start

Der Start von Programm 1 und 2 erfolgt über das Bedienfeld des Steuergerätes.

5 Parameter

5.1 Allgemeines

Folgende Tabelle zeigt eine Übersicht der Parameter der Steuergeräte CR1 und CR2 und deren Funktionen.

Eine genaue Beschreibung der Vorgehensweise zur Einstellung von Parametern finden Sie in Abs. 7.7.1.

5.1.1 Übersicht der Parameter

Parameter		Anzahl der Felder/ Zeichen	Beschreibung	Werkseinstellung
Standardwerkzeugkoordinaten	MEXTL	Reelle Zahl 6	Legt den Werkzeugmittelpunkt TCP fest (X, Y, Z, A, B, C) Einheit: mm oder Grad	0.0,0.0,0.0,0.0,0.0,0.0
Standardbasiskoordinaten	MEXBS	Reelle Zahl 6	Legt das Roboterkoordinatensystem in Beziehung zum Weltkoordinatensystem fest (X, Y, Z, A, B, C) Einheit: mm oder Grad	0.0,0.0,0.0,0.0,0.0,0.0
Verfahrweggrenzen für XYZ-Bewegungen	MEPAR	Reelle Zahl 6	Legt die Verfahrweggrenzen für das XYZ-Koordinatensystem fest (-X, +X, -Y, +Y, -Z, +Z) Einheit: mm	-10000, 10000, -10000, 10000, -10000, 10000
Verfahrweggrenzen für Gelenkbewegungen	MEJAR	Reelle Zahl 16	Legt die Verfahrweggrenzen für jedes einzelne Gelenk fest (-J1, +J1, -J2, +J2, ... -J8, +J8) Einheit: mm	Abhängig vom Roboter
Verfahrwegbegrenzungsebene			Die Verfahrwegsgrenzen werden über eine Ebene definiert. Die Ebene wird über die Koordinaten X1, Y1, Z1 bis X3, Y3, Z3 festgelegt. Bei Überschreitung dieser Bereichsgrenzen erfolgt eine Fehlermeldung. Folgende 3 Parametertypen können verwendet werden:	
	SFC1P : SFC8P	Reelle Zahl 9	Über SFC1P bis SFC8P können 8 Begrenzungsebenen definiert werden. Setzen Sie die dazu nötigen 9 Elemente in folgender Reihenfolge: X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3. Einheit: mm	0.0,0.0,0.0,0.0,0.0,0.0
	SFC1ME : SFC8ME	Ganze Zahl 1	Zuweisung der Roboter 1 bis 8 an die Begrenzungsebenen	0
	SFC1AT : SFC8AT	Ganze Zahl 1	Freigabe der 8 Begrenzungsebenen: freigegeben/gesperrt = 1/0	0

Tab. 5-1: Übersicht der Parameter (1)

Parameter	Anzahl der Felder/ Zeichen	Beschreibung	Werkseinstellung
Benutzerdefinierte Verfahrwegsgrenze		Über zwei Punkte wird ein kubischer Bereich festgelegt. Ein Eindringen in diesen Bereich wird als Verfahrweg-überschreitung definiert und ein korrespondierendes Signal kann geschaltet werden. Es können 8 Bereiche definiert werden.	
AREA1P1 : AREA8P1	Reelle Zahl 8	Festlegung des 1. Bereichspunktes Setzen Sie die 8 Elemente in folgender Reihenfolge: X, Y, Z, A, B, C, L1, L2 (L1 und L2 definieren die Zusatzachsen).	0,0,0,0,0,0,0,0,0,0,0,0
AREA1P2 : AREA8P2	Reelle Zahl 8	Festlegung des 2. Bereichspunktes Setzen Sie die 8 Elemente in folgender Reihenfolge: X, Y, Z, A, B, C, L1, L2 (L1 und L2 definieren die Zusatzachsen).	0,0,0,0,0,0,0,0,0,0,0,0
AREA1ME : AREA8ME	Ganze Zahl 1	Zuweisung der Roboter 1 bis 8 an die Begrenzungsbereiche	0
AREA1AT : AREA8AT	Ganze Zahl 1	Festlegung der Bereichsprüfmethode: Gesperrt/Zone/Überschreitung = 0/1/2 Zone: Das Signal USRAREA wird eingeschaltet. Überschreitung: Es erfolgt eine Fehlermeldung.	0
Automatische Rückkehr nach einem Interrupt	RETPATH 1	Bewirkt den Neustart des Programms nach Auftreten eines Interrupts von der Interruptposition aus (Funktion aktiviert/deaktiviert = 1/0)	1
Programmwahl	SLOTON 1	Auswahl des Programmes, das dem Programmplatz bei Initialisierung zugewiesen wurde (Dektiert/Aktiviert = 0/1) Der Status „Keine Auswahl“ wird gesetzt, wenn keine Angabe erfolgt.	1
Summer EIN/AUS	Buzzer ON/OFF 1	Schaltet den Summer EIN/AUS (AUS/EIN = 0/1)	1
Betriebsgeschwindigkeit für Automatikbetrieb	SPI	Legt die Grundgeschwindigkeit für den Automatikbetrieb fest	
Übersteuerungswert für Automatikbetrieb	EOV	Legt den Übersteuerungswert für den Automatikbetrieb fest (externe Übersteuerung, Programmübersteuerung)	
JOG-Einstellung	JOGJSP 3	Festlegung der Geschwindigkeit für den Gelenk-JOG- und den Schrittbetrieb (Einstellung der Werte H/L, maximaler Übersteuerungswert)	Abhängig vom Roboter
	JOGPSP 3	Festlegung der Geschwindigkeit für den Linear-JOG- und den Schrittbetrieb (Einstellung der Werte H/L, maximaler Übersteuerungswert)	Abhängig vom Roboter

Tab. 5-1: Übersicht der Parameter (2)

Parameter		Anzahl der Felder/ Zeichen	Beschreibung	Werkseinstellung
Geschwindigkeitsbegrenzung für den JOG-Betrieb	JOGSPMX	Reelle Zahl 1	Geschwindigkeitsbegrenzung im TEACH-Modus (max. 250 mm/s)	250.0
Handausführung	HANDTYPE	Zeichenkette 8	Festlegung der Handausführung (Einfach-/Doppelmagnetspule = S/D) und Signalnummer Geben Sie erst den Handtyp, dann die Signalnummer an: z. B. D900. Bei einer Einstellung von D900 werden die Signale Nr. 900 und 901 ausgegeben.	D900, D902, D904, D906 ...
Initialisierungsstatus der HAND	HANDINIT	Ganze Zahl 8	Festlegung der Ausgänge der Schnittstellenkarte für die pneumatische Greifhand nach Einschalten der Spannungsversorgung	1,0,1,0,1,0,1,0
Stopp-Eingangssignalarbeitung	INB	Ganze Zahl 1	Definition des Stopp-Eingangs als Standard oder Drahtbrucherkennung (Standard/Drahtbrucherkennung = 0/1)	0
Benutzerdefinierter Nullpunkt	USERORG	Reelle Zahl 8	Festlegung des benutzerdefinierten Nullpunkts (J1, J2, J3, J4, J5, J6, J7, J8) Einheit: Grad	0,0,0,0,0,0,0,0,0,0,0,0,0,0
Programmplatzliste	SLT : SLT32	Zeichenkette 4	Festlegung der Einstellungen (Programmname, Ausführungsformat, Startbedingung und Priorität) jedes Programmplatzes bei der Initialisierung Programmname: Einstellwert Ausführungsformat: kontinuierlich/zyklisch = REP/CYC Startbedingung: Normal/Fehler/Ständig = START/ ERROR/ALWAYS Priorität: 1 bis 32	1, REP, START, 1
Benutzerdefiniertes Programm	PRGUSR	Zeichenkette 1	Festlegung des benutzerdefinierten Programms (Basisprogramm)	—
Anzahl der Programmplätze	TASKMAX	Ganze Zahl 1	Festlegung der maximalen Anzahl der Programmplätze für eine parallele Ausführung (Multitasking)	8
Kommunikationseinstellungen	CBAU232	Ganze Zahl 1	Festlegung der Übertragungsrates	9600
	CLEN232	Ganze Zahl 1	Festlegung der Datenlänge	0
	CPRTY232	Ganze Zahl 1	Festlegung der Parität (0: keine, 1: ungerade, 2: gerade)	2
	CSTOP232	Ganze Zahl 1	Festlegung des Stoppbits (1, 2)	2
	CTERM232	Ganze Zahl 1	Festlegung des Endezeichens (0. CR, 1: CR + LF)	0
	CPRC232	Ganze Zahl 1	Kommunikationsart (0: kein Protokoll, 1: Protokoll, 2: Datenverbindung)	0

Tab. 5-1: Übersicht der Parameter (3)

Parameter	Anzahl der Felder/ Zeichen	Beschreibung	Werkseinstellung
Ausgangsbitmuster beim Rücksetzen		Ausgangsbitmuster beim Rücksetzen über den CLR-Befehl oder den Eingang OUTRESET Wird in Einheiten von 32 Bit mit den folgenden Parametern gesetzt (AUS/EIN/HALTEN = 0/1/*)	
	ORST0	Zeichenkette 4 Setzen der Ausgangsbits 0–31	00000000,00000000,00000000
	ORST32 : ORST224	Zeichenkette 4 : Setzen der Ausgangsbits 224–255	00000000,00000000,00000000
Ausgang beim RESET rücksetzen	SLRSTIO	Ganze Zahl 1 Zurücksetzen der Ausgänge, wenn das Programm zurückgesetzt wird (gesperrt/freigegeben = 0/1)	0
Betriebsrechte zum Rücksetzen	PRSTNA	Ganze Zahl 1 Freigabe der Betriebsrechte zum Rücksetzen des Programms (gesperrt/freigegeben = 0/1) Sind die Betriebsrechte zum Rücksetzen freigegeben, kann das Programm über ein beliebiges externes Gerät zurückgesetzt werden.	0
Betriebsrechte zur Programmwahl	OPPSL	Ganze Zahl 1 Festlegung der Betriebsrechte zur Programmwahl für den Automatikbetrieb (OP) (extern/OP = 0/1)	1
	RMTPSL	Ganze Zahl 1 Festlegung der Betriebsrechte zur Programmwahl für den Automatikbetrieb (Ext.) (extern/OP = 0/1)	0
Maximale Beschleunigung/Abbremsung		Einstellung der Hand- und Arbeitsbedingungen wenn über OADL die optimale Beschleunigung gewählt wurde Es können bis zu 8 Bedingungen definiert werden. Die Auswahl von Kombinationen der Bedingungen erfolgt über den Befehl LOADSET.	
	HNDDAT1 : HNDDAT8	Reelle Zahl 7 Einstellung der Handbedingungen (Festlegung im Werkzeugkoordinatensystem) (Gewicht, Größe X, Größe Y, Größe Z, Schwerpunkt X, Schwerpunkt Y, Schwerpunkt Z) Einheit: kg, mm	Standardlast, 0,0,0,0,0,0,0,0,0,0,0,0
	WRKDAT1 : WRKDAT8	Reelle Zahl 7 Einstellung der Arbeitsbedingungen (Festlegung im Werkzeugkoordinatensystem) (Gewicht, Größe X, Größe Y, Größe Z, Schwerpunkt X, Schwerpunkt Y, Schwerpunkt Z) Einheit: kg, mm	0,0,0,0,0,0,0,0,0,0,0,0,0
	HNDHOLD1 : HNDHOLD8	Ganze Zahl 2 Festlegung, ob die Hand beim Befehl HOPEN (HCLOSE) geschlossen wird oder nicht (Einstellung für Öffnen, Einstellung für Schließen) (nicht schließen/schließen = 0/1)	0, 1

Tab. 5-1: Übersicht der Parameter (4)

Parameter		Anzahl der Felder/ Zeichen	Beschreibung	Werkseinstellung
Programm fortsetzen	CTN	Ganze Zahl 1	<p>Für den Programmplatz 1 wird nach dem Ausschalten der Spannungsversorgung die aktuelle Position innerhalb der Anwendung gespeichert. Nach dem nächsten Einschalten der Spannungsversorgung startet die Anwendung von dieser gespeicherten Position. Gespeichert wird die Übersteuerung, die Programmzeile, Programmvariable und der Zustand des Ausgangssignals.</p> <p>(freigegeben/gesperrt = 1/0)</p> <ul style="list-style-type: none"> Die Funktion CTN greift auf den Programmspeicherbereich zurück. Bei CTN = 1 kann die Position über die Software gespeichert und die Parameter verändert werden. Nach dem Aus- und Wiedereinschalten der Spannungsversorgung werden die geänderten Parameter über die Software eingelesen und gespeichert. Ist CTN = 1 gesetzt, wird die Speicherkapazität herabgesetzt (mit den Standardeinstellungen um 56 %). Wird der Programmspeicher in eine Anzahl von Adressen konvertiert, wird die Kapazität um 1100 Adressen herabgesetzt. Bei einer Konvertierung in Schritte, wird die Kapazität um 2200 Schritte herabgesetzt. Adressenanzahl = 2500 → 1400 Anzahl der Schritte = 5000 → 2800 Nur das Programm im Programmplatz 1 kann aus dem „Standby-Modus“ gestartet werden. Die Programme aus den anderen Programmplätzen werden aus dem „Reset-Modus“ gestartet. 	
JRC-Befehl	JRCEXE	Ganze Zahl 1	Freigabe des JRC-Befehls (freigegeben/gesperrt = 1/0)	0
	JRCQTT	Reelle Zahl 8	Im Parameter JRCQTT werden die Werte festgelegt, um die benutzerdefinierte Achsen verschoben werden. Die Achsen werden in der Reihenfolge J1, J2, J3 bis J8 angegeben. J7 ist eine zusätzliche Roboterachse und J8 eine zusätzliche mechanische Achse. Die Einheit der Werte aus JRCQTT ist im Parameter AXUNT festgelegt.	JRC freigegeben: 0,0,0,0,0,360,0,0 JRC gesperrt: 0,0,0,0,0,0,0,0
	JRCORG	Reelle Zahl 8	Festlegung der Grundposition für JRC = 0. Diese Einstellung ist nur bei benutzerdefinierten Achsen möglich. Die Einheit der Werte aus JRCORG ist im Parameter AXUNT festgelegt.	0,0,0,0,0,0,0,0
Einstellung zusätzlicher Achsen	AXUNT	Reelle Zahl 16	Festlegung der Einheit. (Winkel (°)/Länge (mm) = 0/1)	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

Tab. 5-1: Übersicht der Parameter (4)

6 Externe Ein-/Ausgänge

6.1 Einteilung

Die externen Ein-/Ausgänge sind in 3 Gruppen aufgeteilt:

- **Spezielle Ein-/Ausgänge**
Die Ein-/Ausgänge dienen zur Steuerung und Statusanzeige des Roboters.
- **Allgemeine Ein-/Ausgänge**
Die Ein-/Ausgänge dienen zur Steuerung von Peripheriegeräten und können frei programmiert werden.
- **Ein-/Ausgänge für Greifhand**
Die Ein-/Ausgänge können zur Unterstützung von Handfunktionen programmiert werden. Dazu benötigen Sie das optionale Steuermodul für die pneumatische/elektrische Greifhand.



ACHTUNG:

Sie können die Spezial-Eingänge während der Programmausführung in allgemeine Eingänge umdefinieren. Das ist aus Sicherheitsgründen nur für die numerischen Dateneingänge zu empfehlen. Dagegen können Sie die Spezialausgänge nicht als allgemeine Ausgänge im Programm benutzen. Bei einem Versuch löst der Roboter Alarm aus.

6.2 Parallele Ein-/Ausgangsschnittstelle

Standardmäßig verfügt das Steuergerät über eine parallele Ein-/Ausgangsschnittstelle. Die Ein-/Ausgangskapazität kann durch Anschluss von weiteren sieben externen parallelen Ein-/Ausgangs-Schnittstellenmodulen auf 256 Ein- und Ausgänge (inkl. Standardschnittstellenmodul) erweitert werden. Die parallele Ein-/Ausgangsschnittstelle (Standard) ist mit einem 50-poligen Centronics-Steckeranschluss ausgerüstet. Wenn Sie externe Geräteeinheiten an einen Roboter anschließen möchten, benötigen Sie ein spezielles Ein-/Ausgangskabel RV-E-E/A (Option).

Die 24-V-DC-Spannungsversorgung für die externen Ein-/Ausgangsschnittstellen und die 12 bis 24-V-DC-Spannungsversorgung für die Ein- und Ausgangskreise müssen vom Anwender bereitgestellt werden.

Eine detaillierte Beschreibung der Ein- und Ausgangsschnittstellen und eine Auflistung der technischen Daten der Ein- und Ausgangskreise finden Sie im Technischen Handbuch des jeweiligen Roboters.

Übersicht der Pinbelegung für Anschluss CN100 (Kabel RV-E-E/A)

Pin-Nr.	Aderfarbe	Funktion	
		Allgemeine Verwendung	Spezial-Versorgungsspannung / Bezugspunkt
1	Weiß		FG
2	Braun		0 V für Pins 4–7
3	Grün		+12 V/+24 V für Pins 4–7
4	Gelb	Ausgang 0	Betrieb
5	Grau	Ausgang 1	Servo EIN
6	Rosa	Ausgang 2	Fehler
7	Blau	Ausgang 3	Betriebsrechte
8	Rot		0 V für Pins 10–13
9	Schwarz		+12 V/+24 V für Pins 10–13
10	Violett	Ausgang 8	
11	Grau-rosa	Ausgang 9	
12	Rot-blau	Ausgang 10	
13	Weiß-grün	Ausgang 11	
14	Braun-grün		COM0: Bezugspunkt für Pins 15–22
15	Weiß-gelb	Eingang 0	Stopp (für alle Anwendungen) ^①
16	Gelb-braun	Eingang 1	Servo AUS
17	Weiß-grau	Eingang 2	Fehler quittieren
18	Grau-braun	Eingang 3	Start
19	Weiß-rosa	Eingang 4	Servo EIN
20	Rosa-braun	Eingang 5	Betriebsrechte
21	Weiß-blau	Eingang 6	
22	Braun-blau	Eingang 7	
23	Weiß-rot		
24	Braun-rot		
25	Weiß-schwarz		
26	Braun-schwarz		FG
27	Grau-grün		0 V für Pins 29–32
28	Gelb-grau		+12 V/+24 V für Pins 29–32
29	Rosa-grün	Ausgang 4	
30	Gelb-rosa	Ausgang 5	
31	Grün-blau	Ausgang 6	
32	Gelb-blau	Ausgang 7	
33	Grün-rot		0 V für Pins 35–38
34	Gelb-rot		+12 V/+24 V für Pins 35–38
35	Grün-schwarz	Ausgang 12	
36	Gelb-schwarz	Ausgang 13	
37	Grau-blau	Ausgang 14	
38	Rosa-blau	Ausgang 15	
39	Grau-rot		COM1: Bezugspunkt für Pins 40–47
40	Rosa-rot	Eingang 8	

Tab. 6-1: Übersicht der Pinbelegung des Standard-Ein/Ausgangsmoduls CN100 (1)

Pin-Nr.	Aderfarbe	Funktion	
		Allgemeine Verwendung	Spezial-Versorgungsspannung / Bezugspunkt
41	Grau-schwarz	Eingang 9	
42	Rosa-schwarz	Eingang 10	
43	Blau-schwarz	Eingang 11	
44	Rot-schwarz	Eingang 12	
45	Weiß-braun-schwarz	Eingang 13	
46	Gelb-grün-schwarz	Eingang 14	
47	Grau-rosa-schwarz	Eingang 15	
48	Blau-rot-schwarz		
49	Weiß-grün-schwarz		
50	Grün-braun-schwarz		

Tab. 6-1: Übersicht der Pinbelegung des Standard-Ein/Ausgangsmoduls CN100 (2)

- ① Der Stopp-Eingang ist fest mit dem Eingangsbit „0“ verbunden und kann nicht gelöscht und einem anderen Eingangsbit zugewiesen werden.

HINWEIS

Bei Verwendung des CR1-Steuergerätes müssen die Pins 2, 8, 27 und 33 mit 0 V verbunden werden sowie die Pins 3, 9, 28 und 34 mit der +12 V/24 V Spannungsversorgung.

Übersicht der Pinbelegung für Anschluss CN300 (Kabel RV-E-E/A)

Pin-Nr.	Aderfarbe	Funktion	
		Allgemeine Verwendung	Spezial-Versorgungsspannung / Bezugspunkt
1	Weiß		FG
2	Braun		0 V für Pins 4–7
3	Grün		+12 V/+24 V für Pins 4–7
4	Gelb	Ausgang 16	
5	Grau	Ausgang 17	
6	Rosa	Ausgang 18	
7	Blau	Ausgang 19	
8	Rot		0 V für Pins 10–13
9	Schwarz		+12 V/+24 V für Pins 10–13
10	Violett	Ausgang 24	
11	Grau-rosa	Ausgang 25	
12	Rot-blau	Ausgang 26	
13	Weiß-grün	Ausgang 27	
14	Braun-grün		COM0: Bezugspunkt für Pins 15–22
15	Weiß-gelb	Eingang 16	
16	Gelb-braun	Eingang 17	
17	Weiß-grau	Eingang 18	
18	Grau-braun	Eingang 19	
19	Weiß-rosa	Eingang 20	
20	Rosa-braun	Eingang 21	
21	Weiß-blau	Eingang 22	
22	Braun-blau	Eingang 23	
23	Weiß-rot		
24	Braun-rot		
25	Weiß-schwarz		
26	Braun-schwarz		FG
27	Grau-grün		0 V für Pins 29–32
28	Gelb-grau		+12 V/+24 V für Pins 29–32
29	Rosa-grün	Ausgang 20	
30	Gelb-rosa	Ausgang 21	
31	Grün-blau	Ausgang 22	
32	Gelb-blau	Ausgang 23	
33	Grün-rot		0 V für Pins 35–38
34	Gelb-rot		+12 V/+24 V für Pins 35–38
35	Grün-schwarz	Ausgang 28	
36	Gelb-schwarz	Ausgang 29	
37	Grau-blau	Ausgang 30	
38	Rosa-blau	Ausgang 31	
39	Grau-rot		COM1: Bezugspunkt für Pins 40–47
40	Rosa-rot	Eingang 24	

Tab. 6-2: Übersicht der Pinbelegung des Standard-Ein/Ausgangsmoduls CN300 (1)

Pin-Nr.	Aderfarbe	Funktion	
		Allgemeine Verwendung	Spezial-Versorgungsspannung / Bezugspunkt
41	Grau-schwarz	Eingang 25	
42	Rosa-schwarz	Eingang 26	
43	Blau-schwarz	Eingang 27	
44	Rot-schwarz	Eingang 28	
45	Weiß-braun-schwarz	Eingang 29	
46	Gelb-grün-schwarz	Eingang 30	
47	Grau-rosa-schwarz	Eingang 31	
48	Blau-rot-schwarz		
49	Weiß-grün-schwarz		
50	Grün-braun-schwarz		

Tab. 6-2: Übersicht der Pinbelegung des Standard-Ein/Ausgangsmoduls CN300 (2)

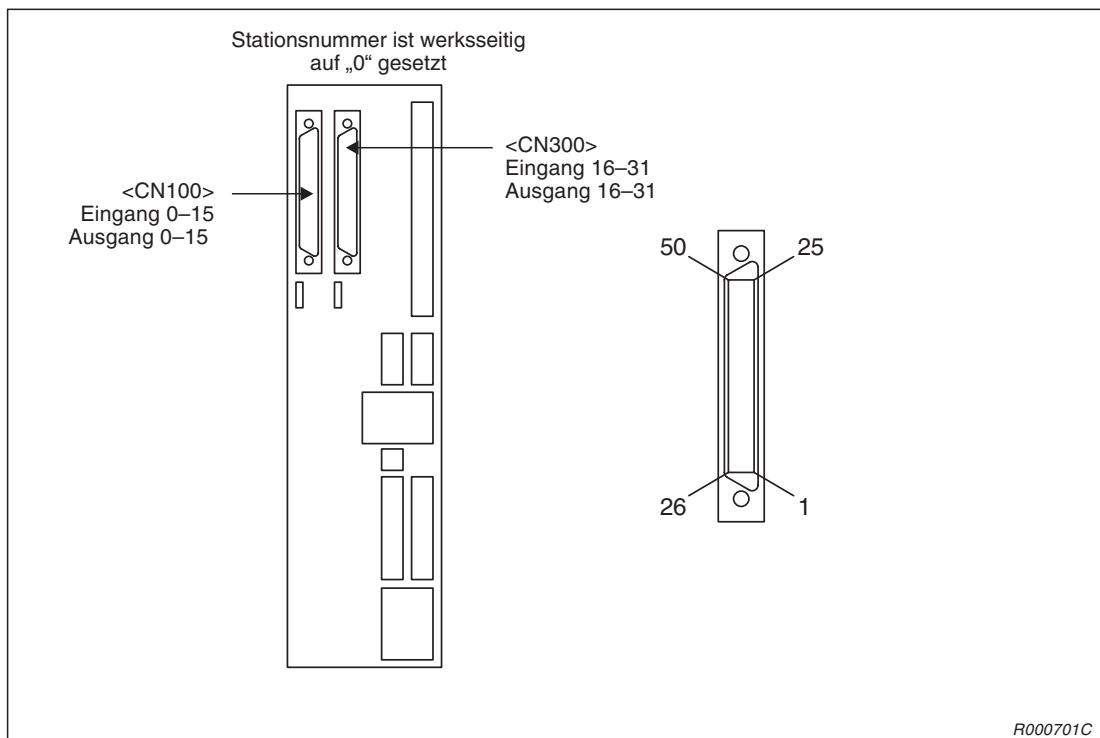


Abb. 6-1: Anschlussbelegung der parallelen Ein-/Ausgangsschnittstelle



ACHTUNG:

Werksseitig ist die Stationsnummer auf „0“ gesetzt. Stellen Sie keine Nummer zwischen 8-F ein, da dieses zu undefinierten Aktivitäten führen kann.

6.2.1 Ein-/Ausgangsbelegung der parallelen Ein-/Ausgangsschnittstelle

In folgender Tabelle sind die Funktionen aufgelistet, die den Ein-/Ausgängen zugewiesen werden können. Die Parameter werden den Signalnummern in der Reihenfolge Eingangssignalnummer/Ausgangssignalnummer zugewiesen. Die genaue Vorgehensweise zur Einstellung von Parametern finden Sie im Abs. 7.7.1. Die Anzahl der verfügbaren Ein-/Ausgangssignale kann durch die optionalen parallelen Ein-/Ausgangsschnittstellen vergrößert werden.

Parameter	Zuordnung	Bezeichnung	Beschreibung	Signal-pegel ^①	Werks-einstellung ^②
TEACHMD	Eingang	—	—		-1, -1
	Ausgang	Ausgangssignal Teach-Modus	Zeigt den Teach-Betrieb an		
ATTOPMD	Eingang	—	—		-1, -1
	Ausgang	Ausgangssignal Automatikbetrieb	Zeigt den Automatikbetrieb an		
ATEXTMD	Eingang	—	—		-1, -1
	Ausgang	Ausgangssignal externer Betrieb	Zeigt den externen Betrieb an		
AUTOENA	Eingang	Freigabe Automatikbetrieb	EIN: Automatikbetrieb freigeben, AUS: Automatikbetrieb gesperrt	H	-1, -1
	Ausgang	Ausgangssignal Automatikbetrieb freigeben	Zeigt an, dass der Automatikbetrieb freigeben ist		
START	Eingang	Startsignal	Startet die Programme im Multitasking-Betrieb	↑	3, 0
	Ausgang	Ausgangssignal Programm aktiv	Zeigt ein aktives Programm an		
STOP	Eingang	Stoppsignal	Stoppt alle Programme Die Eingangssignalnummer ist auf „0“ festgelegt. HINWEIS: Verwenden Sie für alle sicherheitsrelevanten Stopps den NOT-AUS-Eingang.	H	0, -1
	Ausgang	Wartestatus aktiv	Zeigt an, dass die Abarbeitung des entsprechenden Programms vorübergehend unterbrochen worden ist		
STOPSTS	Eingang	—	—		-1, -1
	Ausgang	Eingabe des Stoppsignals	Zeigt an, dass das Stoppsignal eingegeben wurde (logische Addition aller Geräte)		
SLOTINIT	Eingang	Programme zurücksetzen	Setzt den Wartestatus der Programme und die Programme selbst zurück	↑	-1, -1
	Ausgang	Ausgangssignal Programmwahl freigeben	Zeigt an, dass die Programmwahl freigeben ist		
ERRRESET	Eingang	Fehler quittieren	Quittiert den aktuellen Fehler	↑	2, 2
	Ausgang	Ausgangssignal Fehler	Zeigt an, dass ein Fehler aufgetreten ist		

Tab. 6-3: Spezielle Parameter für Ein-/Ausgänge (1)

Parameter	Zuordnung	Bezeichnung	Beschreibung	Signal- pegel ^①	Werks- einstellung ^②
CYCLE	Eingang	Zyklischen Betrieb stoppen	Stoppt den zyklischen Betrieb	H	-1, -1
	Ausgang	Ausgangssignal zyklischer Betrieb gestoppt	Zeigt an, dass der zyklische Betrieb gestoppt ist		
SRVOFF	Eingang	Servoversorgung abschalten	Schaltet die Servoversorgung ab, das Einschalten der Servos wird gesperrt	H	1, -1
	Ausgang	Servos einschalten gesperrt	Zeigt an, dass das Einschalten der Servos gesperrt ist (Rückmeldung)		
SRVON	Eingang	Servoversorgung einschalten	Schaltet die Servoversorgung für alle Servos ein		4, 1
	Ausgang	Servoversorgung eingeschaltet	Zeigt an, dass die Servoversorgung eingeschaltet ist		
IOENA	Eingang	Eingangssignal Betriebsrechte	Anforderung der Betriebsrechte für eine externe Steuerung	↑	5, 3
	Ausgang	Ausgangssignal Betriebsrechte	Zeigt an, dass der Betrieb über externe Signale freigegeben ist		
MELOCK	Eingang	Verriegelungssignal	Ein- bzw. Auschalten des Verriegelungszustandes		-1, -1
	Ausgang	Ausgangssignal Verriegelung aktiv	Zeigt an, dass der Roboter im verriegelten Zustand ist		
SAFEPOS	Eingang	Eingangssignal Ersatzposition anfahren	Anfahren der Ersatzposition	↑	-1, -1
	Ausgang	Fährt die Ersatzposition an	Zeigt an, dass die Ersatzposition angefahren wird		
RCREADY	Eingang	—	—		-1, -1
	Ausgang	Spannungsversorgung des Steuergerätes eingeschaltet	Zeigt an, dass die Spannungsversorgung des Steuergerätes eingeschaltet ist und externe Signale empfangen werden können		
BATERR	Eingang	—	—		-1, -1
	Ausgang	Batteriespannung niedrig	Zeigt an, dass die Batteriespannung abgesunken ist		
OUTRESET	Eingang	Allgemeine Ausgangssignale zurücksetzen	Zurücksetzen der allgemeinen Ausgangssignale	↑	-1, -1
	Ausgang	—	—		
HLVLERR	Eingang	—	—		-1, -1
	Ausgang	Schwerer Fehler	Zeigt an, dass ein schwerer Fehler aufgetreten ist		
LLVLERR	Eingang	—	—		-1, -1
	Ausgang	Leichter Fehler	Zeigt an, dass ein leichter Fehler aufgetreten ist		
CLVLERR	Eingang	—	—		-1, -1
	Ausgang	Warnung	Zeigt eine Warnung an		

Tab. 6-3: Spezielle Parameter für Ein-/Ausgänge (2)

Parameter	Zuordnung	Bezeichnung	Beschreibung	Signal- pegel ^①	Werks- einstellung ^②
EMGERR	Eingang	—	—		-1, -1
	Ausgang	Ausgangssignal NOT-HALT	Zeigt den NOT-HALT-Status an		
S1START ⋮ S32START	Eingang	Startheingang	Startet das entsprechende Programm	↑	-1, -1 ⋮ -1, -1
	Ausgang	Programm aktiv	Zeigt den aktiven Status jedes Programms an		
S1STOP ⋮ S32STOP	Eingang	Stoppeingang	Stoppt das entsprechende Programm	H	-1, -1 ⋮ -1, -1
	Ausgang	Programm gestoppt	Zeigt an, dass das Programm vorübergehend gestoppt wurde		
M1SRVOFF ⋮ M5SRVOFF	Eingang	Eingangssignal Servo AUS für Roboter 1 ⋮ Eingangssignal Servo AUS für Roboter 5	Schaltet die Servoversorgung jedes Roboters aus und sperrt das Einschalten	H	-1, -1 ⋮ -1, -1
	Ausgang	Ausgangssignal Servo EIN bei Roboter 1 gesperrt ⋮ Ausgangssignal Servo EIN bei Roboter 5 gesperrt	Zeigt an, dass das Einschalten der Servoversorgung gesperrt ist (Rückmeldung)		
M1SRVON ⋮ M5SRVON	Eingang	Eingangssignal Servo EIN für Roboter 1 ⋮ Eingangssignal Servo EIN für Roboter 5	Schaltet die Servoversorgung jedes Roboters ein	↑	-1, -1 ⋮ -1, -1
	Ausgang	Ausgangssignal Servo EIN bei Roboter 1 ⋮ Ausgangssignal Servo EIN bei Roboter 5	Zeigt an, dass die Servoversor- gung eingeschaltet ist		
M1MELOCK ⋮ M5MELOCK	Eingang	Eingangssignal Roboter 1 verriegeln ⋮ Eingangssignal Roboter 5 verriegeln	Schaltet die Verriegelung jedes Roboters aus oder ein	↑	-1, -1 ⋮ -1, -1
	Ausgang	Ausgangssignal Roboter 1 verriegelt ⋮ Ausgangssignal Roboter 5 verriegelt	Zeigt an, dass der Roboter im Verriegelungszustand ist		
PRGSEL	Eingang	Programmwahlsignal	Einlesen der numerischen Eingabe zur Programmwahl	↑	-1, -1
	Ausgang	—	—		

Tab. 6-3: Spezielle Parameter für Ein-/Ausgänge (3)

Parameter	Zuordnung	Bezeichnung	Beschreibung	Signal-pegel ^①	Werks-einstellung ^②
OVRDSEL	Eingang	Geschwindigkeitsübersteuerung auswählen	Einlesen der numerischen Geschwindigkeitsübersteuerung	↑	-1, -1
	Ausgang	—	—		
IODATA ^③	Eingang	Eingang für numerische Eingabe (Start-Nr., End-Nr.)	Eingabe der Programmnummer, Geschwindigkeitsübersteuerung, Zuordnungsnummer	H	-1, -1 -1, -1
	Ausgang	Ausgang für numerische Ausgabe (Start-Nr., End-Nr.)	Ausgabe der Programmnummer, Geschwindigkeitsübersteuerung, Zuordnungsnummer		
PRGOUT	Eingang	Ausgabeanforderung Programmnummer	Anforderung zur Ausgabe der Programmnummer	↑	-1, -1
	Ausgang	Ausgabe der Programmnummer	Zeigt an, dass die Programmnummer über den numerischen Ausgang ausgegeben wird		
LINEOUT	Eingang	Ausgabeanforderung Zeilennummer	Anforderung zur Ausgabe der Zeilennummer	↑	-1, -1
	Ausgang	Ausgabe der Zeilennummer	Zeigt an, dass die Zeilennummer über den numerischen Ausgang ausgegeben wird		
OVRDOUT	Eingang	Ausgabeanforderung Geschwindigkeitsübersteuerung	Anforderung zur Ausgabe der Geschwindigkeitsübersteuerung	↑	-1, -1
	Ausgang	Ausgabe der Geschwindigkeitsbeeinflussung	Zeigt an, dass die Geschwindigkeitsübersteuerung über den numerischen Ausgang ausgegeben wird		
ERROUT	Eingang	Ausgabeanforderung Fehlernummer	Anforderung zur Ausgabe der Fehlernummer	↑	-1, -1
	Ausgang	Ausgabe der Fehlernummer	Zeigt an, dass die Fehlernummer über den numerischen Ausgang ausgegeben wird		
JOGENA	Eingang	Freigabe Jog-Betrieb	Freigabe des Jog-Betriebs über externe Signale	↑	-1, -1
	Ausgang	Freigabe Jog-Betrieb	Zeigt an, dass der Jog-Betrieb über externe Signale freigegeben ist		
JOGM ^④	Eingang	2-Bit-Eingabe des Jog-Betriebs (Start-Nr., End-Nr.)	Festlegung des Jog-Betriebs 0/1/2/3/4 = Gelenk-, Linear-, Kreis-, 3-Achsen-Linear-, Werkzeug-JOG-Betrieb	H	-1, -1 -1, -1
	Ausgang	2-Bit-Ausgabe des Jog-Betriebs (Start-Nr., End-Nr.)	Ausgabe des aktuellen Jog-Betriebs		
JOG+ ^⑤	Eingang	Jog-Vorschub in positiver Richtung für 8 Achsen (Start-Nr., End-Nr.)	Festlegung des Jog-Betriebs in positiver Richtung	H	-1, -1
	Ausgang	—	—		
JOG- ^⑤	Eingang	Jog-Vorschub in negativer Richtung für 8 Achsen (Start-Nr., End-Nr.)	Festlegung des Jog-Betriebs in negativer Richtung	H	-1, -1
	Ausgang	—	—		

Tab. 6-3: Spezielle Parameter für Ein-/Ausgänge (4)

Parameter	Zuordnung	Bezeichnung	Beschreibung	Signal-pegel ^①	Werks-einstellung ^②
HNDCNTL1 : HNDCNTL5	Eingang	—	—		-1, -1 : -1, -1
	Ausgang	Handsteuersignal für Hand Roboter 1 : Handsteuersignal für Hand Roboter 5	Ausgabe der Signalzustände der Ausgänge 900 bis 907 : Ausgabe der Signalzustände der Ausgänge 940 bis 947		
HNDSTS1 : HNDSTS5	Eingang	—	—		-1, -1 : -1, -1
	Ausgang	Handsensorsignal für Hand Roboter 1 : Handsensorsignal für Hand Roboter 5	Ausgabe der Signalzustände der Eingänge 900 bis 907 : Ausgabe der Signalzustände der Eingänge 940 bis 947		
HNDERR1 : HNDERR5	Eingang	Eingangssignal Fehler Hand 1 : Eingangssignal Fehler Hand 5	Abfrage auf Handfehler	H	-1, -1 : -1, -1
	Ausgang	Ausgangssignal Fehler Hand 1 : Ausgangssignal Fehler Hand 5	Zeigt an, dass ein Handfehler aufgetreten ist		
AIREERR1 : AIREERR5	Eingang	Luftdruck im Pneumatiksystem 1 fehlerhaft : Luftdruck im Pneumatiksystem 5 fehlerhaft	Abfrage auf Pneumatikfehler	H	-1, -1 : -1, -1
	Ausgang	Ausgabe Pneumatikfehler im System 1 : Ausgabe Pneumatikfehler im System 5	Zeigt an, dass ein Fehler im Pneumatiksystem aufgetreten ist		
USRAREA ^⑤	Eingang	—	—		-1, -1
	Ausgang	Über 8 Punkte festgelegter Arbeitsbereich (Start-Nr., End-Nr.)	Zeigt an, dass der Roboter sich im Arbeitsbereich befindet		

Tab. 6-3: Spezielle Parameter für Ein-/Ausgänge (5)

- ① Eingang: „H“ bedeutet, die Funktion ist aktiv, wenn das externe Signal eingeschaltet ist und inaktiv, wenn das externe Signal ausgeschaltet ist.
Eingang: „↑“ bedeutet, die Funktion ist aktiv, wenn das externe Signal vom AUS- in den EIN-Zustand wechselt. Die aktivierte Funktion bleibt auch nach einem Wechsel des externen Signals in den AUS-Zustand erhalten. Erst bei einem erneuten Wechsel des externen Signals vom AUS- in den EIN-Zustand wird die Funktion zurückgesetzt.
- ② Die Werkseinstellung „-1“ bedeutet, dass die Funktion nicht aktiviert ist.
- ③ Die Eingabe erfolgt in der Reihenfolge: Eingangsstartnummer, Eingangsendnummer, Ausgangsstartnummer, Ausgangsendnummer. Geben Sie bei einer Ein-/Ausgabe eines aktuellen Wertes die Start- und Endnummer als binären Wert an. Dabei entspricht die Startnummer dem niederwertigen, die Endnummer dem höherwertigen Bit. Setzen Sie nur die zur Einstellung notwendigen Werte. Stehen z. B. bei einer Programmwahl nur die Programme 1 bis 6 zur Auswahl, reichen zur Darstellung 3 Bits. Es können bis zu 16 Bits gesetzt werden.

Beispiele ▾

Die Zuweisung des Starteingangssignals an Eingang 16 und des Ausgangssignals „Programm aktiv“ an Ausgang 25 erfolgt über:

Parameter START = [16, 25]

Die Zuweisung der numerische Eingabe an die Eingänge 6 bis 9 und der numerischen Ausgabe an die Ausgänge 6 bis 9 erfolgt über:

Parameter IODATA = [6, 9, 6, 9]

- △
- ④ Die Eingabe erfolgt in der Reihenfolge: Eingangsstartnummer, Eingangsendnummer, Ausgangsstartnummer, Ausgangsendnummer. Geben Sie bei Aktivierung des aktuellen JOG-Modus die Start- und Endnummer als binären Wert an. Dabei entspricht die Startnummer dem niederwertigen, die Endnummer dem höherwertigen Bit. Setzen Sie nur die zur Einstellung notwendigen Werte.
 - ⑤ Die Eingabe erfolgt in der Reihenfolge: Eingangsstartnummer, Eingangsendnummer, Ausgangsstartnummer, Ausgangsendnummer. Über die Startnummer wird die Achse J1/X festgelegt und über die Endnummer können Achsen bis zu J8/L2 festgelegt werden.
 - ⑥ Die Eingabe erfolgt in der Reihenfolge: Ausgangsstartnummer, Ausgangsendnummer.

Beispiele ▾

Die Festlegung zweier Benutzerbereiche erfolgt über zwei Bits. Die Einstellung beider Benutzerbereiche auf den Ausgang 10 erfolgt über:

Parameter USRAREA = [10, 10]

Die Festlegung eines Benutzerbereichs auf die Ausgänge 10–11 erfolgt über:

Parameter USRAREA = [10, 11]

Dabei müssen die Ausgänge in numerischer Reihenfolge angegeben werden. Ein Benutzerbereich kann nicht Ausgang 10 und 13 beinhalten.

△

Freigabe der zugewiesenen Eingangssignale

Die Gültigkeit eines anliegenden und zugewiesenen Eingangssignals hängt vom Betriebszustand des Roboters ab.

Parameter	Bezeichnung	Gültigkeit
SLOTINIT	Programme zurücksetzen	Keine Funktion während des Betriebs (bei Ausgabe des START-Signals)
SAFEPOS	Eingangssignal Ersatzposition anfahren	
OUTRESET	Allgemeine Ausgangssignale zurücksetzen	
PRGSEL	Programmwahlsignal	
START	Startsignal	Keine Funktion bei Freigabe des externen Betriebs (bei Ausgabe des IOENA-Signals)
SLOTINIT	Programme zurücksetzen	
SRVON	Servoversorgung einschalten	
MELOCK	Verriegelungssignal	
SAFEPOS	Eingangssignal Ersatzposition anfahren	
PRGSEL	Programmwahlsignal	
OVRDSEL	Geschwindigkeitsübersteuerung auswählen	
JOGENA	Freigabe Jog-Betrieb	
SLOTINIT	Programme zurücksetzen	Keine Funktion bei Eingabe des Stoppsignals (bei Ausgabe des STOPSTS-Signals)
SAFEPOS	Eingangssignal Ersatzposition anfahren	
JOGENA	Freigabe Jog-Betrieb	
SRVON	Servoversorgung einschalten	Keine Funktion bei eingeschaltetem SRVOFF-Signal
MELOCK	Verriegelungssignal	Funktion nur im Programmauswahlmodus (bei Ausgabe des SLOTINIT-Signals)

Tab. 6-4: Gültigkeit der Eingangssignale

6.2.2 Programmsteuerung durch externe Signale

Zeitablaufdiagramme bei externer Steuerung

Folgende Abbildung zeigt das Zeitablaufdiagramm für die Steuerung der Funktionen „Programmwahl“, „Start“, „Stopp“ und „Neustart“ durch externe Signale:

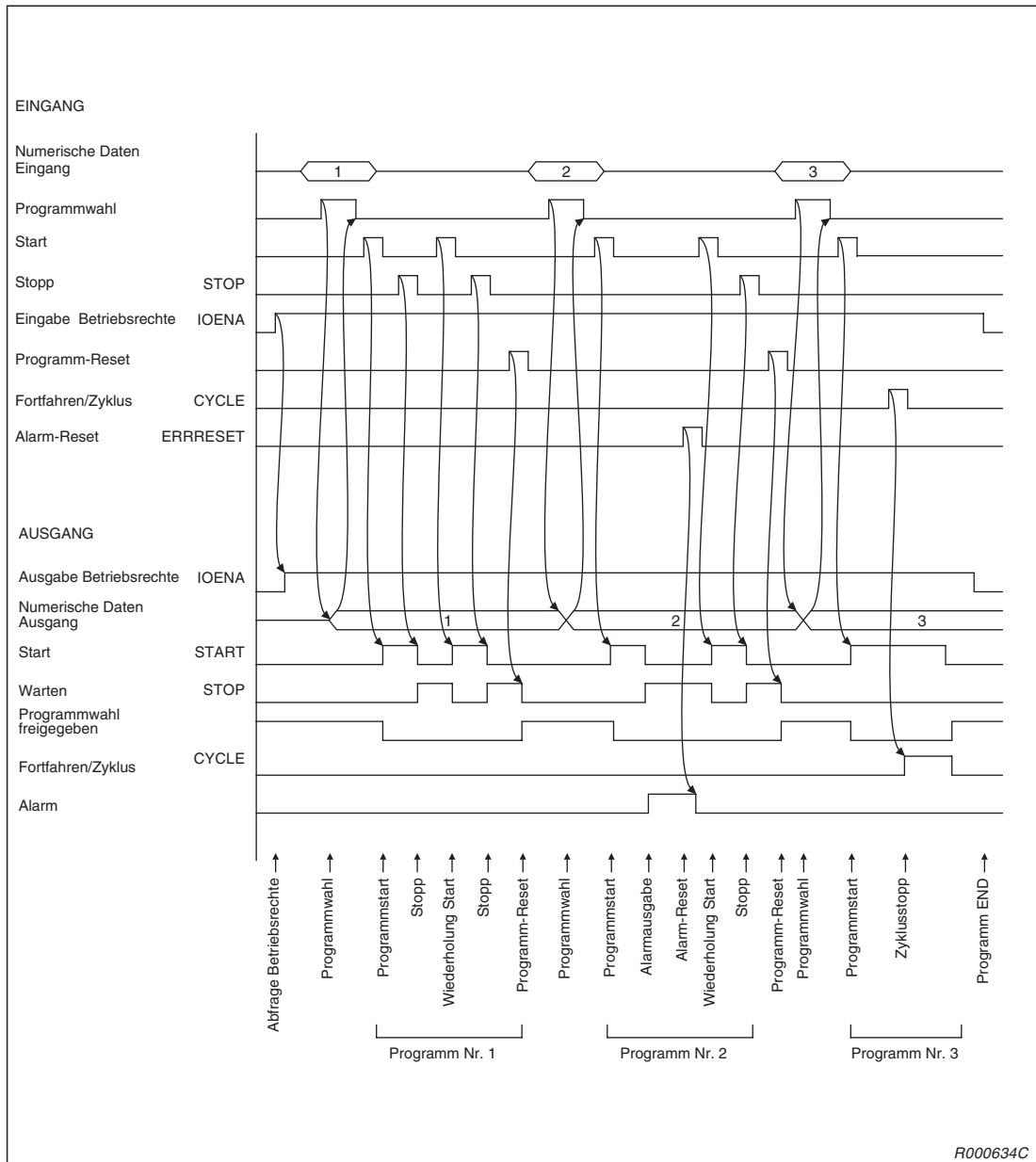


Abb. 6-2: Zeitablaufdiagramm 1 bei externer Steuerung

Folgende Abbildung zeigt das Zeitablaufdiagramm für die Steuerung der Funktionen „Servo EIN/AUS“, „Programmwahl“, „Auswahl des Geschwindigkeitsübersteuerungswertes“, „Start“, „Ausgabe der Zeilennummer“ usw. durch externe Signale:

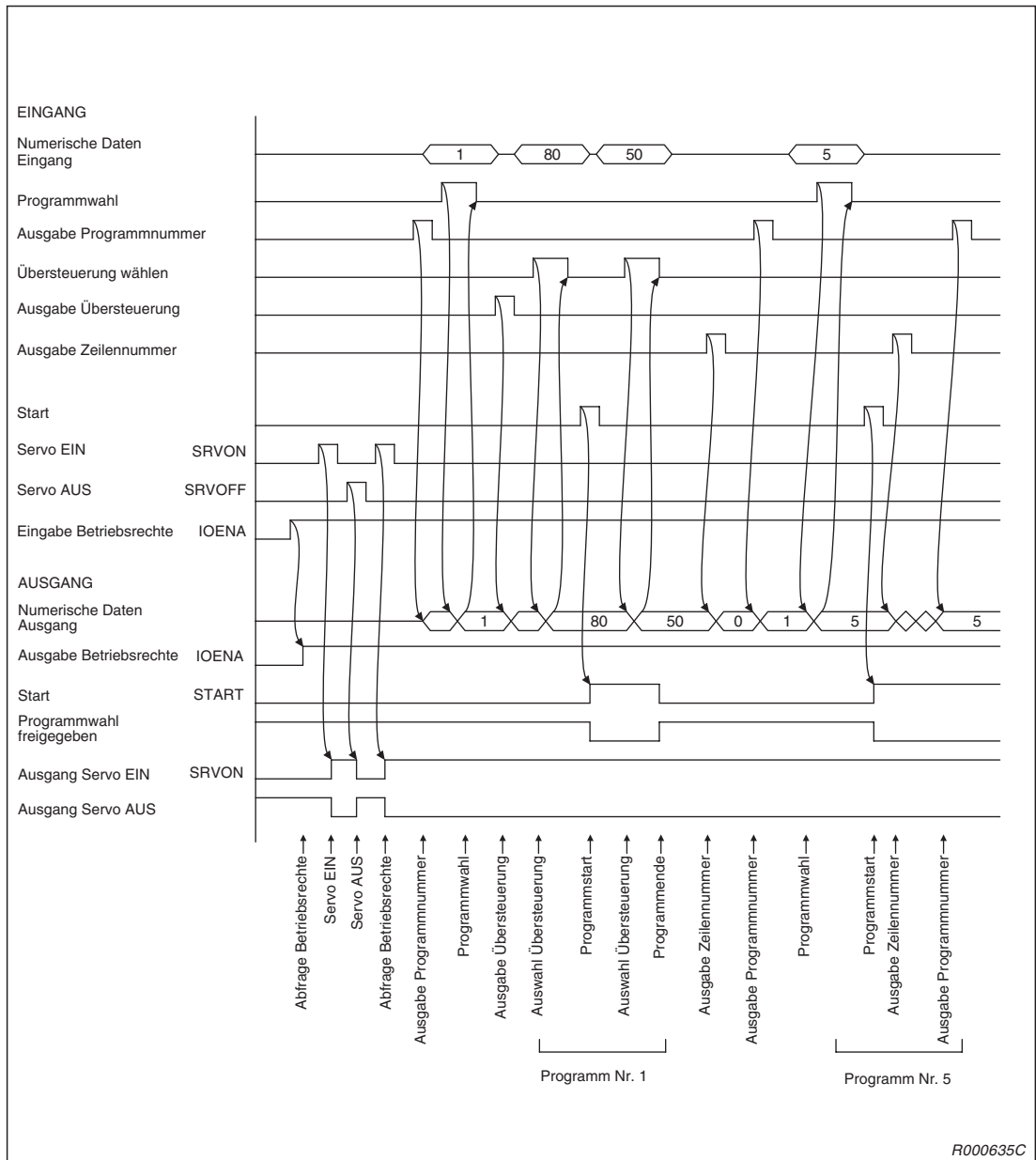


Abb. 6-3: Zeitablaufdiagramm 2 bei externer Steuerung

Folgende Abbildung zeigt das Zeitablaufdiagramm für die Steuerung der Funktionen „Fehler zurücksetzen“, „Allgemeinen Ausgang zurücksetzen“, „Programm zurücksetzen“ usw. durch externe Signale:

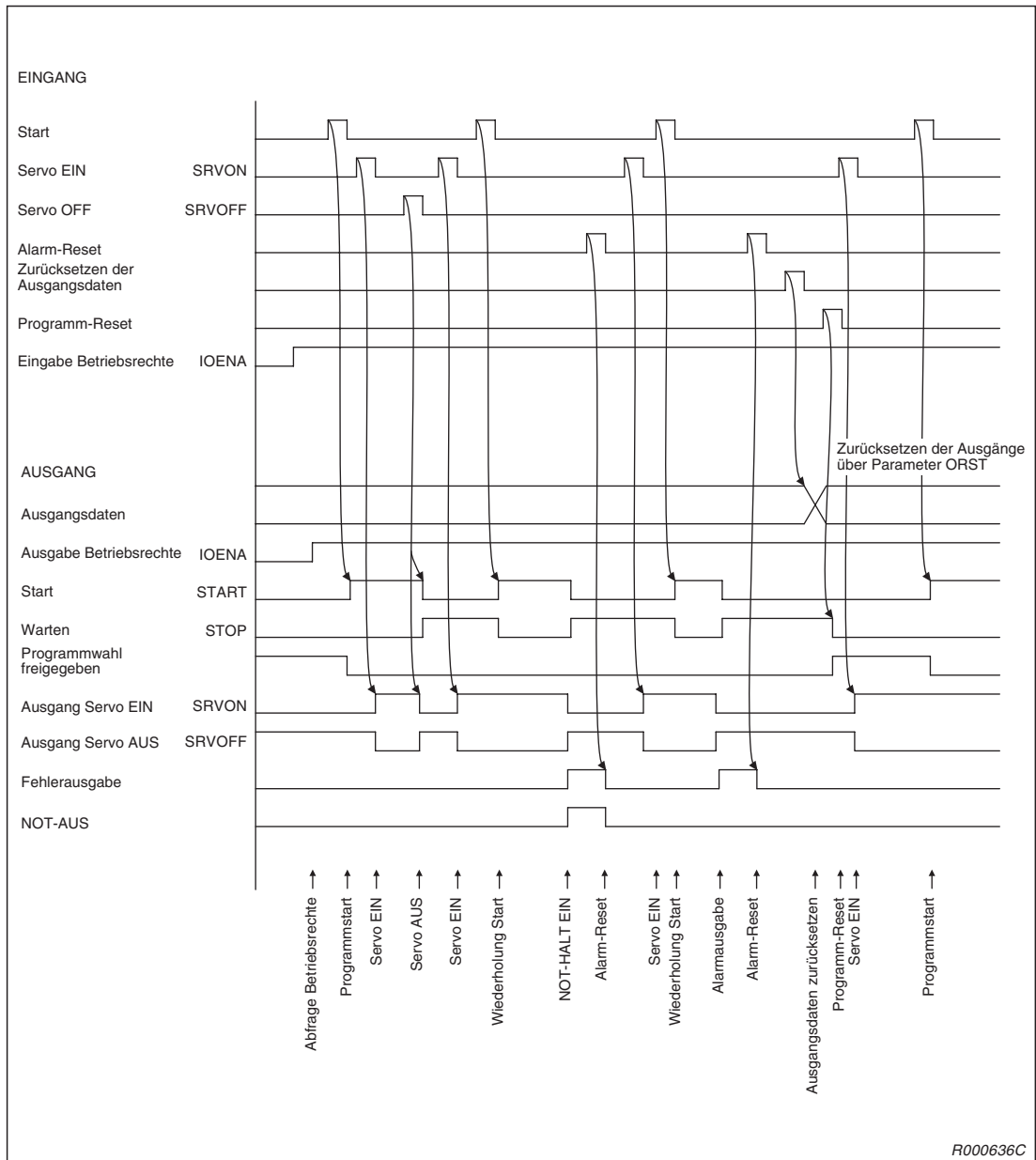


Abb. 6-4: Zeitablaufdiagramm 3 bei externer Steuerung

Folgende Abbildung zeigt das Zeitablaufdiagramm für die Steuerung der Funktionen „JOG-Betrieb“, „Anfahren der Ersatzposition“, „Programm zurücksetzen“ usw. durch externe Signale:

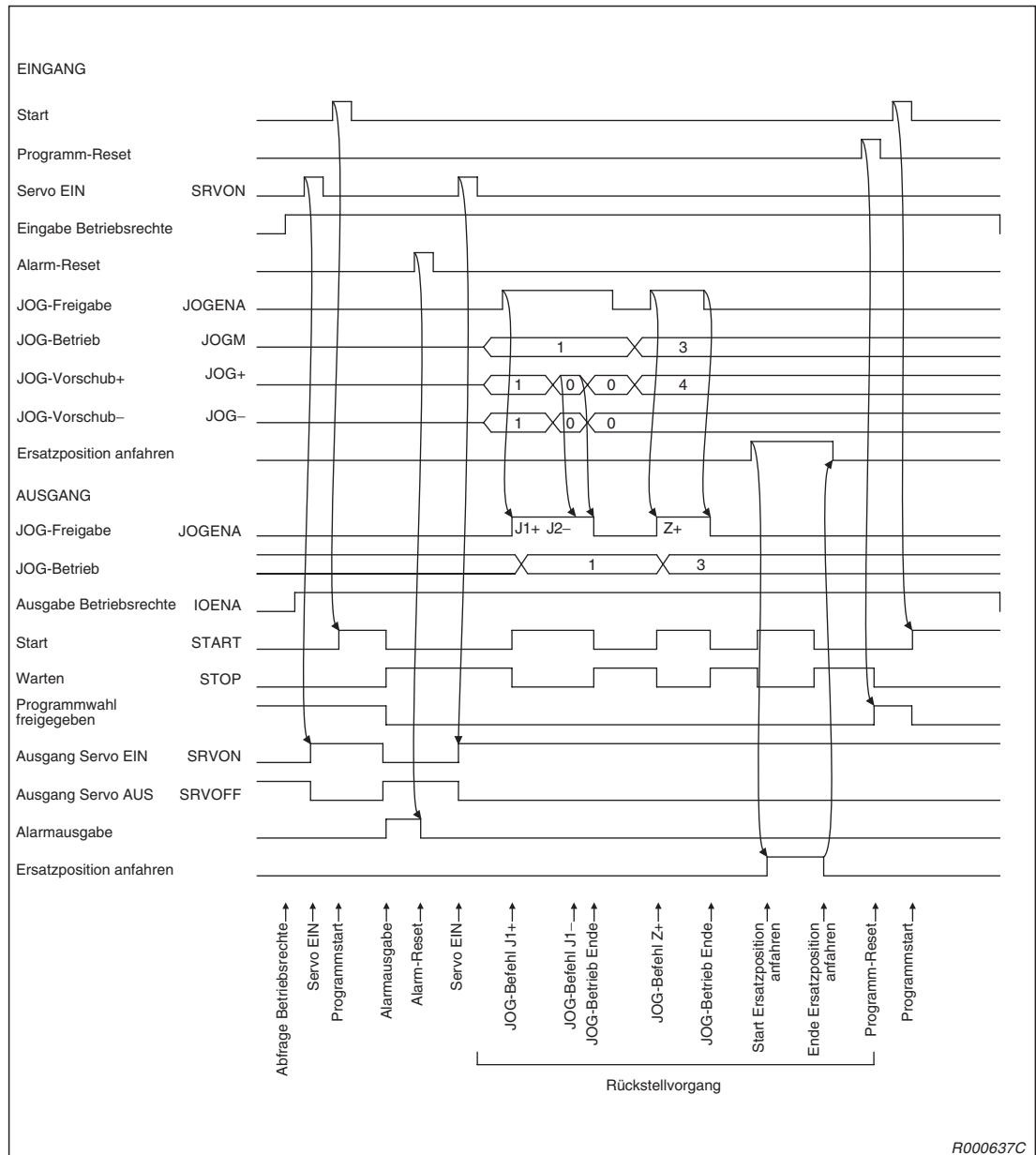


Abb. 6-5: Zeitablaufdiagramm 4 bei externer Steuerung

6.3 NOT-HALT-Eingang

6.3.1 Steuergerät CR1

Auf der Rückseite des Steuergerätes befindet sich der NOT-HALT-Stecker. Auf diesem Stecker sind 6 Anschlussklemmen, je zwei um einen externen NOT-HALT-Schalter, einen Tür-Schließkontakt und eine Signallampe in den Schaltkreis des Roboters zu integrieren. Standardmäßig sind die Anschlussklemmen für den NOT-HALT-Schalter und den Tür-Schließkontakt mit jeweils einer Drahtbrücke kurzgeschlossen. Der Roboter kann über den NOT-HALT-Schalter an der Vorderseite des Steuergerätes gestoppt werden.

Um einen externen NOT-HALT-Schalter oder Tür-Schließkontakt in den Roboterschaltkreis zu integrieren gehen Sie wie folgt vor:

- ① Lösen Sie die Schrauben der entsprechenden Anschlussklemmen und entfernen die Drahtbrücke.
- ② Nehmen Sie die Anschlussleitung des externen Schalters, z. B. NOT-HALT-Schalter, und entfernen Sie 5 bis 7 mm der Leitungsisolierung.
- ③ Legen Sie das abisolierte Leitungsende unter die Schraubenklemme.
- ④ Drehen Sie die Schrauben fest an.

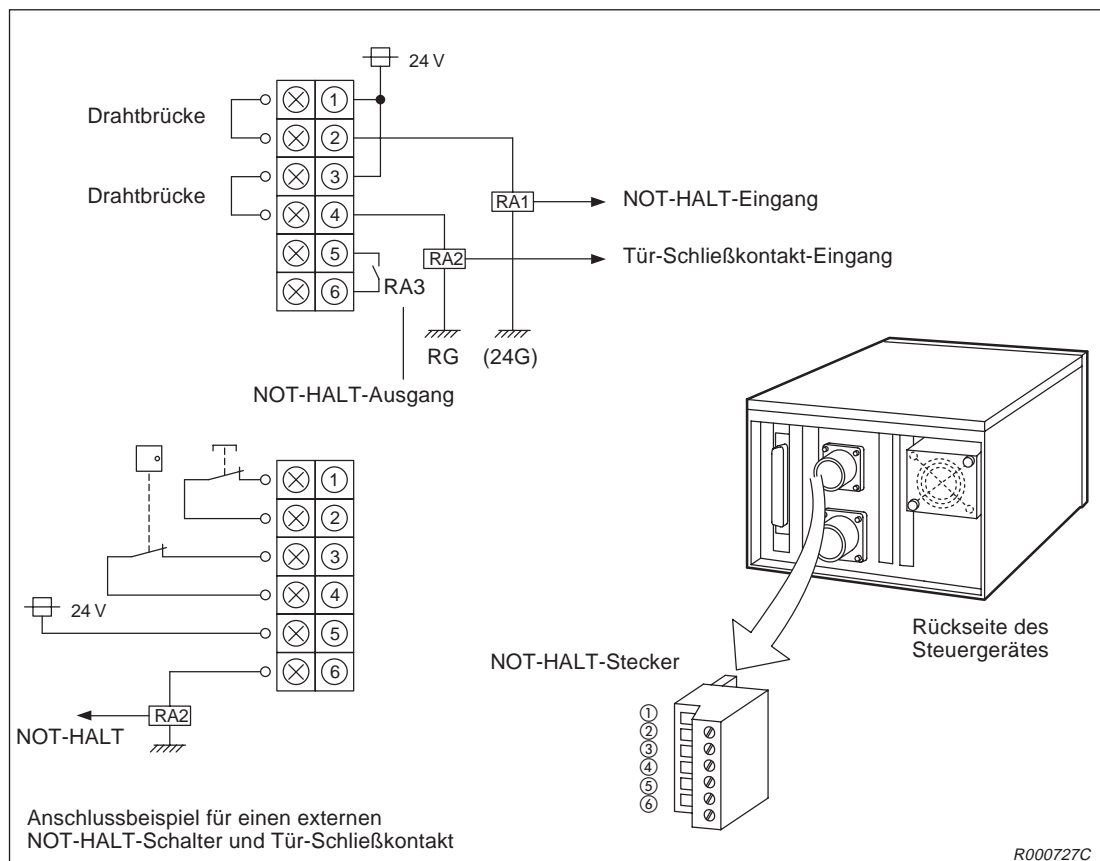


Abb. 6-6: Anschluss eines externen NOT-HALT-Schalters (Steuergerät CR1)

6.3.2 Steuergerät CR2

Der NOT-HALT-Stecker des Steuergerätes CR2 besitzt 5 Anschlussklemmen. An diese Anschlussklemmen kann ein externer NOT-HALT-Schalter angeschlossen werden. Der NOT-HALT-Schaltkreis des Steuergerätes ist redundant, so dass Sie für den externen NOT-HALT-Schalter 4 Anschlussklemmen verwenden müssen.

So integrieren Sie einen externen NOT-HALT-Schalter in den Roboterschaltkreis:

- ① Entfernen Sie die Kunststoffabdeckung des NOT-HALT-Steckers.
- ② Lösen Sie die Schrauben der entsprechenden Anschlussklemmen und entfernen die Drahtbrücken.
- ③ Nehmen Sie die Anschlussleitung des externen Schalters und entfernen 5 bis 7 mm der Leitungsisolierung.
- ④ Legen Sie die abisolierten Leitungsenden unter die Schraubenklemme. Der Anschluss des NOT-HALT-Schalters erfolgt wie in Abb. 6-7 über die Klemmen 1–2 und 3–4.
- ⑤ Drehen Sie die M3,5-Schrauben fest an.
- ⑥ Befestigen Sie die Kunststoffabdeckung des NOT-HALT-Steckers.

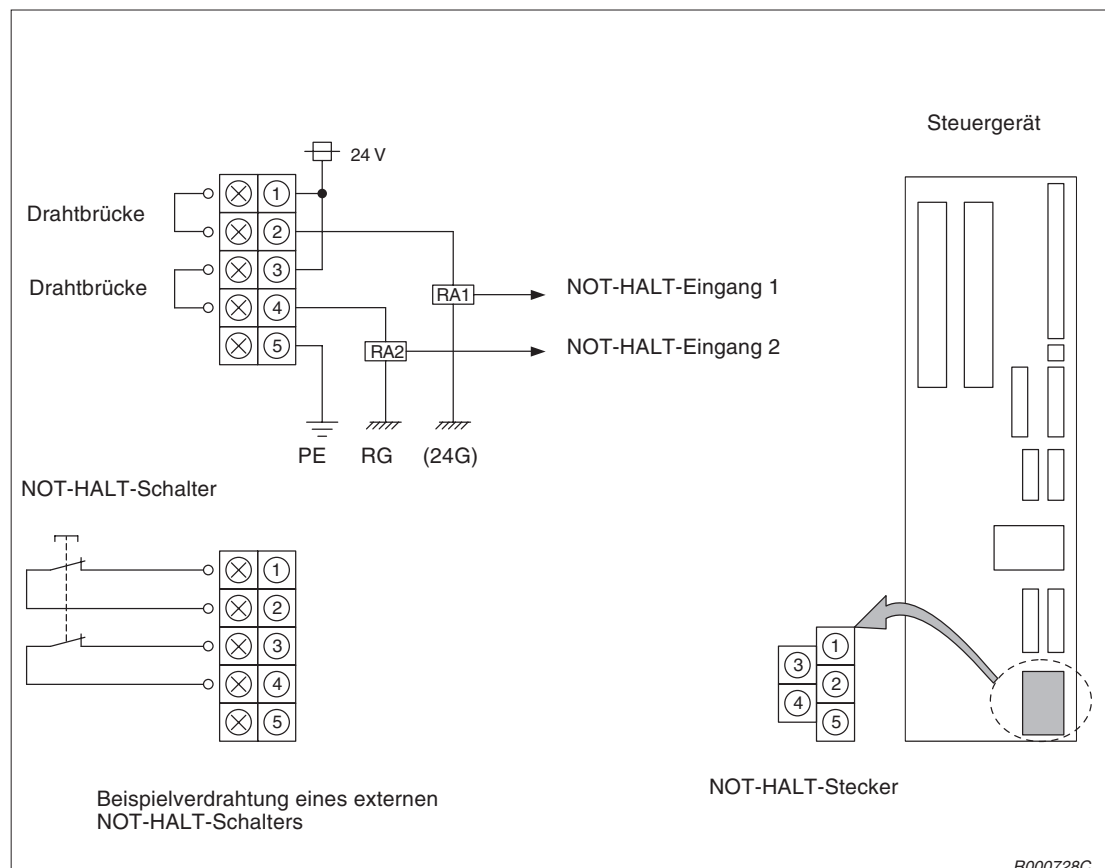


Abb. 6-7: Anschluss eines externen NOT-HALT-Schalters (Steuergerät CR2)

7 Bedienung und Programmierung

7.1 Bedienung der Teaching Box

In diesem Abschnitt wird die Bedienung der Teaching Box und die Funktionen der einzelnen Menüs beschrieben.

7.1.1 Menübaum

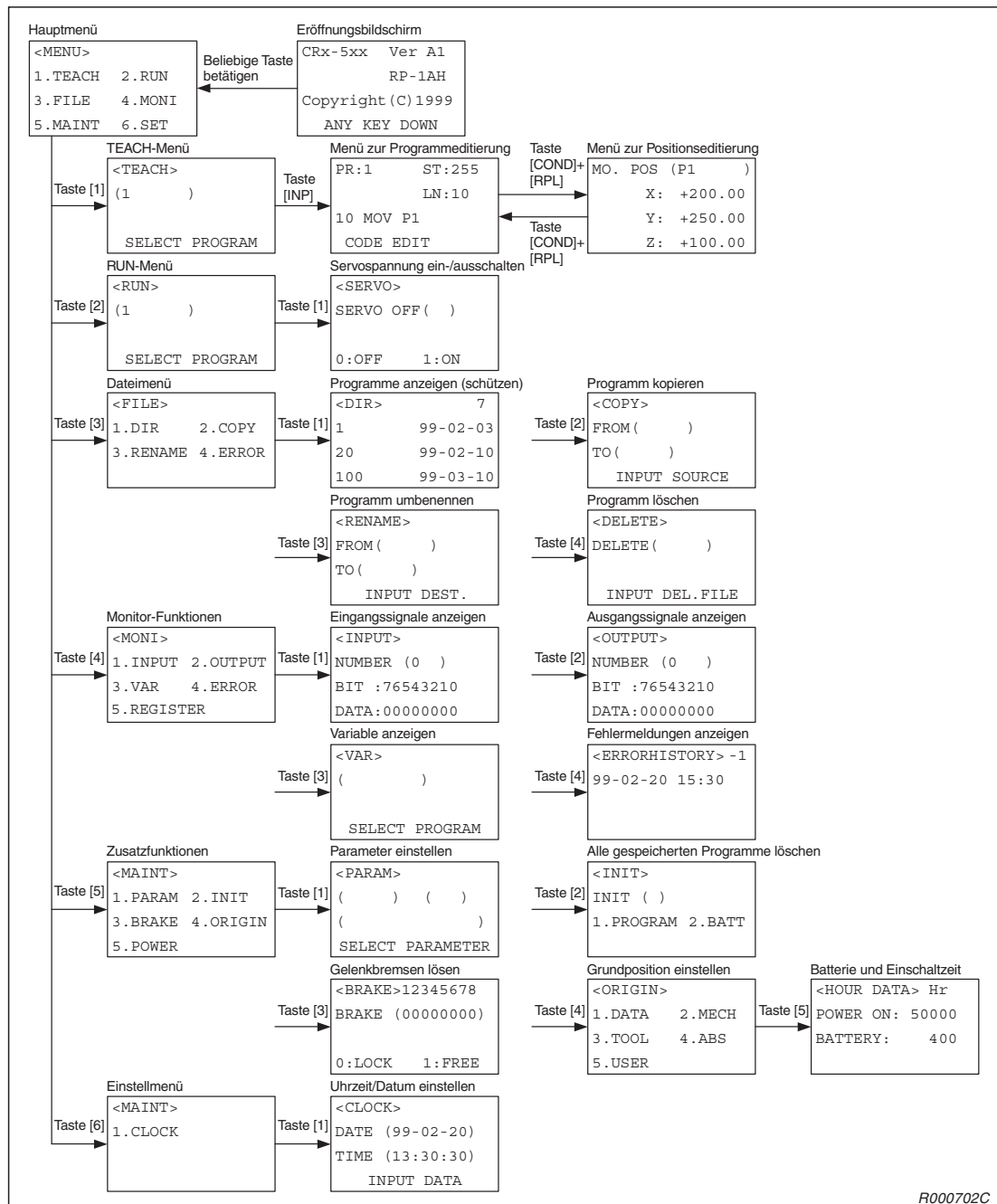


Abb. 7-1: Menübaum

7.1.2 Menüpunkt auswählen

Funktion


Zur Auswahl eines Menüpunkts stehen zwei Möglichkeiten zur Verfügung.

Ausführung

In Tab. 7-2 und Tab. 7-3 werden die beiden Möglichkeiten beispielhaft an der Auswahl des Menüpunkts „1. TEACH“ gezeigt.


Stellen Sie den [MODE]-Schalter des Steuergerätes auf die Stellung „TEACH“. Aktivieren Sie die Teaching Box, indem Sie den [ENABLE/DISABLE]-Schalter der Teaching Box auf „ENABLE“ stellen.

Nach dem Einschalten erscheint der Eröffnungsbildschirm.

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre>CRn-5xx Ver.A1 RP-1AH COPYRIGHT (C) 1999 ANY KEY DOWN</pre>		Betätigen Sie nach Erscheinen des Eröffnungsbildschirms eine beliebige Taste (z. B. [MENU]), um das Hauptmenü aufzurufen.
②	<pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT 6 . SET</pre>		Das Hauptmenü wird angezeigt.



Tab. 7-1: Aufruf des Hauptmenüs

1.) Menüauswahl über Eingabe der Nummer

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT 6 . SET</pre>		Das Menü „TEACH“ wird durch Eingabe der Ziffer „1“ ausgewählt.
②	<pre><TEACH> (1) SELECT PROGRAM</pre>		Das Menü „TEACH“ wird angezeigt.

Tab. 7-2: Beispiel zur Menüauswahl über Eingabe der Nummer

2.) Menüauswahl über Cursor

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 .TEACH 2 .RUN 3 .FILE 4 .MONI 5 .MAINT 6 .SET</pre>		Der Cursor wird über die Tasten [ADD ↑], [RPL ↓], [DEL ←] oder [HAND →] zum gewünschten Menüpunkt bewegt.
②	<pre><MENU> 1 .TEACH 2 .RUN 3 .FILE 4 .MONI 5 .MAINT 6 .SET</pre>		Die Auswahl wird bestätigt.
③	<pre><TEACH> (1) SELECT PROGRAM</pre>		Das Menü „TEACH“ wird angezeigt.

Tab. 7-3: Beispiel zur Menüauswahl über Cursor

HINWEISE

Solange der [MODE]-Schalter des Steuergerätes nicht auf „TEACH“ gestellt ist, sind über die ausgeschaltete Teaching Box nur bestimmte Funktionen (z. B. Anzeige der aktuellen Position im JOG-Betrieb, Änderung der Geschwindigkeitsübersteuerung, Anzeige der Ein- und Ausgangssignalzustände, Fehlerliste usw.) ausführbar.

Die Eingabe von Ziffern erfolgt über die Tasten mit einer Ziffer in der unteren linken Ecke. Die Eingabe eines Leerzeichens erfolgt über die [SPACE]-Taste.

Das Löschen von Zeichen erfolgt über die gleichzeitige Betätigung der Tasten [CHAR] und [DEL ←]. Zum Einfügen von Zeichen wird der Cursor über die [DEL ←]-Taste an die Stelle bewegt, an der das Zeichen eingefügt werden soll. Anschließend kann die Eingabe des gewünschten Zeichens erfolgen.

7.2 Programmierung

Die Programmiersprache MELFA-BASIC IV ermöglicht die Erstellung komplexer Programme mit umfangreichen Funktionen. In diesem Abschnitt wird die Vorgehensweise bei der Programmierung über die Teaching Box erläutert.

Eine detaillierte Beschreibung der MELFA-BASIC-IV-Befehle finden Sie in Abs. 9.3.



Die folgende Tabelle zeigt die Möglichkeiten der Weiterverarbeitung der Daten einer Zeile.

Eingabeformat	Verarbeitung
Zeilennummer und Befehl (Beispiel: 10 MOV P1)	Eingabe wird als Zeile des Roboterprogramms verarbeitet.
Nur Zeilennummer (Beispiel: 10)	Löscht die angegebene Zeile aus dem Programm (durch Überschreiben)
Nur Befehl (Beispiel: MOV P1)	Führt den Befehl sofort aus (Direkt-Modus)

Tab. 7-4: Weiterverarbeitung der übertragenen Daten einer Zeile

7.2.1 Roboterprogramm erstellen

Aufruf des Menüs zur Programmeditierung

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 .TEACH 2 .RUN 3 .FILE 4 .MONI 5 .MAINT 6 .SET</pre>		Das Menü „TEACH“ wird ausgewählt.
②	<pre><TEACH> (1)) SELECT PROGRAM</pre>		Die Programmnummer „1“ wird ausgewählt.
③	<pre>PR:1 ST:1 LN:0 --NO DATA--</pre>		Nach der Tastenbetätigung wird das Menü zur Programmeditierung aufgerufen.

Tab. 7-5: Aufruf des Menüs zur Programmeditierung















HINWEISE

Solange der [MODE]-Schalter des Steuergerätes nicht auf „TEACH“ gestellt ist, sind über die ausgeschaltete Teaching Box nur bestimmte Funktionen (z. B. Anzeige der aktuellen Position im JOG-Betrieb, Änderung der Geschwindigkeitsübersteuerung, Anzeige der Ein- und Ausgangssignalzustände, Fehlerliste usw.) ausführbar.

Die Eingabe von Ziffern erfolgt über die Tasten mit einer Ziffer in der unteren linken Ecke. Die Eingabe eines Leerzeichens erfolgt über die [SPACE]-Taste.

Das Löschen von Zeichen erfolgt über die gleichzeitige Betätigung der Tasten [CHAR] und [DEL ←]. Zum Einfügen von Zeichen wird der Cursor über die [DEL ←]-Taste an die Stelle bewegt, an der das Zeichen eingefügt werden soll. Anschließend kann die Eingabe des gewünschten Zeichens erfolgen.

Erstellung eines neuen Programms

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre>PR:1 ST:1 LN:0 --NO DATA--</pre>	3 x 	Der Cursor wird zum Eingabefeld für die Befehle bewegt.
②	<pre>PR:1 ST:1 LN:0 CODE EDIT</pre>	 →  → 	Die Zeilennummer „10“ wird eingegeben.
③	<pre>PR:1 ST:1 LN:0 10 █ CODE EDIT</pre>	 → 	Das Zeichen „M“ wird eingegeben.
④	<pre>PR:1 ST:1 LN:0 10 M █ CODE EDIT</pre>	 [POS/CHAR]-Taste gedrückt halten	Betätigen Sie die Taste [POS/CHAR] und halten Sie sie gedrückt. Es werden die 4 Befehle angezeigt, die mit „M“ beginnen.
⑤	<pre>1.MOV 2.MVS 3.MVC 4.MVR 10 M █ CODE EDIT</pre>	 → 	Der Befehl „MOV“ wird eingegeben.
⑥	<pre>1.MOV 2.MVS 3.MVC 4.MVR 10 MOV █ CODE EDIT</pre>	 → 	Das Zeichen „P“ wird eingegeben.
⑦	<pre>1.MOV 2.MVS 3.MVC 4.MVR 10 MOV P █ CODE EDIT</pre>		Die Ziffer „1“ wird eingegeben.
⑧	<pre>1.MOV 2.MVS 3.MVC 4.MVR 10 MOV P1 █ CODE EDIT</pre>	 	Jetzt ist die Programmzeile „10 MOV P1“ wirksam.
⑨	<pre>PR:1 ST:2 LN:0 CODE EDIT</pre>		Das Fenster zur Editierung des 2ten Programmschrittes wird angezeigt. Die Eingabe weiterer Programmschritte erfolgt in derselben Weise.


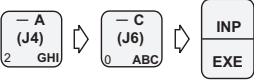
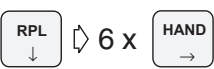




Tab. 7-6: Erstellung eines neuen Programms

Beschreibung

- Eingabe von Zeichen
Betätigen Sie die [POS/CHAR]-Taste und halten Sie diese gedrückt. Betätigen Sie dann zur Zeicheneingabe die zugehörige Taste. Jede Taste zur Zeicheneingabe ist dreifach belegt. Nach jeder Tastenbetätigung wird ein anderes Zeichen auf dem Display angezeigt. Lösen Sie die [POS/CHAR]-Taste, wenn das gewünschte Zeichen angezeigt wird.
- Eingabe von Befehlen
Befehle können zeichenweise (Beispiel „M“ → „O“ → „V“ für den MOV-Befehl) oder durch den Aufruf aus einer Liste eingegeben werden.
Durch Eingabe des Anfangsbuchstaben wird eine Liste der Befehle angezeigt, die mit dem gleichen Buchstaben beginnen. Dazu ist nach Eingabe des Anfangsbuchstaben die [POS/CHAR]-Taste zu betätigen. Die Befehlsliste erscheint. Die Eingabe des Befehls erfolgt durch gleichzeitige Betätigung der zugeordneten Nummerntaste und der [POS/CHAR]-Taste. Wird der gewünschte Befehl nicht in der Liste angezeigt, ist die [POS/CHAR]-Taste erneut zu betätigen.
- Anzeige der vorhergehenden/nächsten Programmzeile
Die vorhergehende Programmzeile wird durch Betätigung der [-/BACKWD]-Taste, die nächste Programmzeile durch Betätigung der [+ /FORWD]-Taste aufgerufen.
- Anzeige einer bestimmten Programmzeile
Bewegen Sie den Cursor über die Taste [ADD ↑] im Menü zur Programmeditierung zum Eingabefeld für die Zeilennummer „LN:“. Geben Sie die Zeilennummer ein und betätigen Sie anschließend die [INP/EXE]-Taste. Die aufgerufene Zeilennummer erscheint.

7.2.2 Roboterprogramm editieren

Rufen Sie das Menü zur Programmeditierung auf (siehe Abs. 7.2.1). Es wird automatisch das Programmverzeichnis angezeigt, wenn Sie keine Programmauswahl vornehmen.

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre>PR:1 ST:1 LN:10 10 MOV P1 CODE EDIT</pre>		Der Cursor wird zum Eingabefeld für die Zeilennummer bewegt.
②	<pre>PR:1 ST:1 LN:(10) CODE EDIT</pre>		Die Zeilennummer „20“ wird eingegeben.
③	<pre>PR:1 ST:2 LN:(20)</pre> <pre>20 MOV P2,-50 CODE EDIT</pre>		Der Cursor wird nach unten und rechts bewegt und eine Stelle hinter „V“ platziert.
④	<pre>PR:1 ST:2 LN:20 20 MOV P2,-50 CODE EDIT</pre>		Die Zeichen „OV“ werden gelöscht. Das Zeichen „M“ bleibt stehen.
⑤	<pre>PR:1 ST:2 LN:20 20 M P2,-50 CODE EDIT</pre>	 [POS/CHAR]-Taste gedrückt halten	Betätigen Sie die Taste [POS/CHAR] und halten Sie sie gedrückt. Es werden die 4 Befehle angezeigt, die mit „M“ beginnen.
⑥	<pre>1.MOV 2.MVS 3.MVC 4.MVR 20 M P2,-50 CODE EDIT</pre>		Der Befehl „MVS“ wird eingegeben.
⑦	<pre>1.MOV 2.MVS 3.MVC 4.MVR 20 MVS P2,-50 CODE EDIT</pre>		Jetzt ist die Programmzeile „20 MVS P2, -50“ wirksam.
⑧	<pre>PR:1 ST:3 LN:30 30 MVS P3 CODE EDIT</pre>		Die nächste Programmzeile wird angezeigt.

Tab. 7-7: Beispiel zum Editieren eines Programms

Beschreibung

- **Steuerung des Cursors**
Befehlszeilen können mit Hilfe des Cursors bearbeitet werden. Eine Steuerung des Cursors erfolgt dabei über die Tasten [ADD ↑], [RPL ↓], [DEL ←] oder [HAND →].
- **Anzeige einer bestimmten Programmzeile**
Bewegen Sie den Cursor im Menü zur Programmeditierung zum Eingabefeld für die Zeilennummer „LN:“. Geben Sie die Zeilennummer ein und betätigen Sie anschließend die [INP/EXE]-Taste. Die aufgerufene Zeilennummer erscheint. Mit den Tasten [+ /FORWD] (vorwärts) und [- /BACKWD] (rückwärts) können Sie im Programm „blättern“.
- **Editieren eines Zeichens**
Betätigen Sie zum Löschen eines Zeichens die Taste [POS/CHAR] zusammen mit der Taste [DEL ←]. Wenn erst [DEL ←] und dann zusätzlich [POS/CHAR] gedrückt wird, wird ein anderes als das gewünschte Zeichen gelöscht! Daher: erst [POS/CHAR] und dann zusätzlich [DEL ←] drücken.
- **Prüfen des editierten Roboterprogramms**
Überprüfen Sie nach erfolgter Editierung des Roboterprogramms die ausgeführten Korrekturen im Schrittbetrieb.

Eingabe der aktuellen Positionsdaten

Im folgenden Beispiel wird die aktuelle Position als Position Nr. 1 definiert:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre>PR:1 ST:13 LN:130 130 END CODE EDIT</pre>		Nach der Tastenbetätigung wird das Menü zur Editierung der Positionsdaten angezeigt.
②	<pre>MO.POS () X: +0.00 Y: +0.00 Z: +0.00</pre>		Die Positionsnummer „1“ wird eingegeben. Danach wird die Eingabe bestätigt und es erscheint die Anzeige der aktuell gespeicherten Position.
③	<pre>MO.POS (P1) X: +0.00 Y: +0.00 Z: +0.00</pre>		Es ertönt ein Summton und eine Bestätigungsabfrage wird angezeigt.
④	<pre>MO.POS (P1) X: +0.00 Y: +0.00 ADDITION?</pre>		Die Definition der neuen Position wird bestätigt. Nach der Anzeige von „ADDING“ und einem weiteren Summton ist die neue Position wirksam.
⑤	<pre>MO.POS (P1) X: +132.30 Y: +254.10 Z: +32.00</pre>		Die Koordinatenwerte der neuen Position sind nun registriert.

Tab. 7-8: Beispiel zur Eingabe der aktuellen Positionsdaten

HINWEIS

Ein Umschalten zwischen den Menüs zur Editierung von Befehlen und zur Editierung von Positionen erfolgt über die gleichzeitige Betätigung der Tasten [POS/CHAR] und [ADD ↑] oder [POS/CHAR] und [RPL ↓].

Position anfahren

Die Handspitze (TCP: Tool Center Point) des Roboters kann zu einer bereits definierten Position gefahren werden. Das Anfahren der Position erfolgt über den MO- oder MS-Bewegungs-befehl.

Bezeichnung	Bewegung
MO-Bewegungsbefehl	Die definierte Position wird mittels Gelenk-Interpolation angefahren.
MS-Bewegungsbefehl	Die definierte Position wird mittels Linear-Interpolation angefahren. Es wird keine Verfahrbewegung ausgeführt, wenn der Stellungsmerker der aktuellen Position von der anzufahrenden Position abweicht.

Tab. 7-9: Anfahren einer definierten Position

Das Anfahren einer Position über den MO-Bewegungsbefehl ist nur im Gelenk-JOG-Modus, möglich. Ändern Sie den JOG-Modus, falls nötig.

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung								
①	<table border="1"> <tr> <td>X, Y, Z</td> <td>LOW</td> </tr> <tr> <td>X:</td> <td>+80.09</td> </tr> <tr> <td>Y:</td> <td>-21.78</td> </tr> <tr> <td>Z:</td> <td>+137.36</td> </tr> </table>	X, Y, Z	LOW	X:	+80.09	Y:	-21.78	Z:	+137.36		Es wird auf den Gelenk-JOG-Betrieb umgeschaltet.
X, Y, Z	LOW										
X:	+80.09										
Y:	-21.78										
Z:	+137.36										
②	<table border="1"> <tr> <td>JOINT</td> <td>LOW</td> </tr> <tr> <td>X:</td> <td>+34.50</td> </tr> <tr> <td>Y:</td> <td>+20.00</td> </tr> <tr> <td>Z:</td> <td>+80.00</td> </tr> </table>	JOINT	LOW	X:	+34.50	Y:	+20.00	Z:	+80.00		Das Menü für den Gelenk-JOG-Betrieb wird angezeigt.
JOINT	LOW										
X:	+34.50										
Y:	+20.00										
Z:	+80.00										
③	<table border="1"> <tr> <td>MO . POS (P2)</td> <td></td> </tr> <tr> <td>X:</td> <td>+132.30</td> </tr> <tr> <td>Y:</td> <td>+254.10</td> </tr> <tr> <td>Z:</td> <td>+32.00</td> </tr> </table>	MO . POS (P2)		X:	+132.30	Y:	+254.10	Z:	+32.00		Bei Tastenbetätigung wird die angezeigte Position mittels Gelenk-Interpolation angefahren. Der Roboter stoppt, sobald die [INP/EXE]-Taste losgelassen wird.
MO . POS (P2)											
X:	+132.30										
Y:	+254.10										
Z:	+32.00										

Tab. 7-10: Anfahren einer definierten Position über den MO-Bewegungsbefehl

Das Anfahren einer Position über den MS-Bewegungsbefehl ist nur im Werkzeug-JOG-Modus, möglich. Ändern Sie den JOG-Modus, falls nötig.

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung								
①	<table border="1"> <tr> <td>JOINT</td> <td>LOW</td> </tr> <tr> <td>X:</td> <td>+34.50</td> </tr> <tr> <td>Y:</td> <td>+20.00</td> </tr> <tr> <td>Z:</td> <td>+80.00</td> </tr> </table>	JOINT	LOW	X:	+34.50	Y:	+20.00	Z:	+80.00		Es wird auf den Werkzeug-JOG-Betrieb umgeschaltet.
JOINT	LOW										
X:	+34.50										
Y:	+20.00										
Z:	+80.00										
②	<table border="1"> <tr> <td>TOOL</td> <td>LOW</td> </tr> <tr> <td>X:</td> <td>+80.09</td> </tr> <tr> <td>Y:</td> <td>-21.78</td> </tr> <tr> <td>Z:</td> <td>+137.36</td> </tr> </table>	TOOL	LOW	X:	+80.09	Y:	-21.78	Z:	+137.36		Das Menü für den Werkzeug-JOG-Betrieb wird angezeigt.
TOOL	LOW										
X:	+80.09										
Y:	-21.78										
Z:	+137.36										
③	<table border="1"> <tr> <td>MS . POS (P2)</td> <td>)</td> </tr> <tr> <td>X:</td> <td>+132.30</td> </tr> <tr> <td>Y:</td> <td>+254.10</td> </tr> <tr> <td>Z:</td> <td>+32.00</td> </tr> </table>	MS . POS (P2))	X:	+132.30	Y:	+254.10	Z:	+32.00		Bei Tastenbetätigung wird die angezeigte Position mittels Gelenk-Interpolation angefahren. Der Roboter stoppt, sobald die [INP/EXE]-Taste losgelassen wird. ①
MS . POS (P2))										
X:	+132.30										
Y:	+254.10										
Z:	+32.00										

Tab. 7-11: Anfahren einer definierten Position über den MS-Bewegungsbefehl

① Ist die Ausführung einer linearen Verfahrbewegung von der aktuellen Position zur Zielposition nicht möglich, verbleibt der Roboter im Stillstand.

Positionsdaten ersetzen


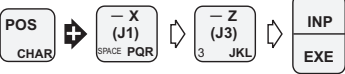

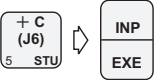
Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre>PR: 1 ST: 8 LN: 80 80 MVS P3 CODE EDIT</pre>		Nach der Tastenbetätigung wird das Menü zur Editierung der Positionsdaten angezeigt.
②	<pre>MO . POS () X: +0.00 Y: +0.00 Z: +0.00</pre>		Die Positionsnummer „3“ wird eingegeben. Danach wird die Eingabe bestätigt und es erscheint die Anzeige der aktuell gespeicherten Position.
③	<pre>MO . POS (P3) X: +132.30 Y: +354.10 Z: +132.00</pre>		Bewegen Sie den Roboter im JOG-Betrieb zu der Position, die angeglich werden soll.
④	<pre>JOINT LOW X: +34.50 Y: +20.00 Z: +80.00</pre>		Die neue Position wird angefahren.
⑤	<pre>MO . POS (P3) X: +132.30 Y: +354.10 Z: +132.00</pre>	<p>[STEP/MOVE]-Taste gedrückt halten</p>	Es ertönt ein Summton und eine Bestätigungsabfrage wird angezeigt.
⑥	<pre>MO . POS (P3) X: +132.30 Y: -284.10 Z: +132.00</pre>		Das Ersetzen der Position wird bestätigt. Nach der Anzeige von „REPLACING“ und einem weiteren Summton ist die neue Position wirksam.
⑦	<pre>MO . POS (P3) X: +132.30 Y: -284.10 Z: +132.00</pre>		Das Ersetzen ist jetzt abgeschlossen.

Tab. 7-12: Beispiel zum Ersetzen von Positionsdaten

Beschreibung

- Aufruf einer Positionsvariablen
Geben Sie den Positionsvariablennamen, der aufgerufen werden soll, in die Klammern hinter dem Befehl MO.POS ein und betätigen Sie anschließend die [INP/EXE]-Taste. Mit den Tasten [+ /FORWD] (vorwärts) und [- /BACKWD] (rückwärts) können Sie durch die Variablenanzeige „blättern“.
- Prüfen des editierten Roboterprogramms
Überprüfen Sie nach erfolgter Editierung des Roboterprogramms die ausgeführten Korrekturen im Schrittbetrieb.













Positionsdaten ändern

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre>PR:1 ST:8 LN:80 80 MVS P3 CODE EDIT</pre>		Nach der Tastenbetätigung wird das Menü zur Editierung der Positionsdaten angezeigt.
②	<pre>MO. POS () X: +0.00 Y: +0.00 Z: +0.00</pre>		Die Positionsnummer „3“ wird eingegeben. Danach wird die Eingabe bestätigt und es erscheint die Anzeige der aktuell gespeicherten Position.
③	<pre>MO. POS (P3) X: +132.30 Y: +354.10 Z: +132.00</pre>		Der Cursor wird zum Y-Eingabefeld und zur „4“ bewegt.
④	<pre>MO. POS (P3) X: +132.30 Y: +354.10 Z: +132.00</pre>		Die Ziffer „4“ wird durch die Ziffer „5“ überschrieben.
⑤	<pre>MO. POS (P3) X: +132.30 Y: +355.10 Z: +132.00</pre>		Der neue Koordinatenwert wird angezeigt.

Tab. 7-13: Beispiel zum Ändern von Positionsdaten

Positionsdaten löschen

Es können nur Positionsdaten gelöscht werden, die nicht vom aktuell ausgeführten Programm verwendet werden. Bei einem Versuch, eine Position zu löschen, die im aktuellen Programm verwendet wird, erfolgt eine Fehlermeldung.

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre>PR:1 ST:8 LN:80 80 MVS P3 CODE EDIT</pre>	 →  oder 	Nach der Tastenbetätigung wird das Menü zur Editierung der Positionsdaten angezeigt.
②	<pre>MO.POS () X: +0.00 Y: +0.00 Z: +0.00</pre>	 →  →  → 	Die Positionsnummer „4“ wird eingegeben. Danach wird die Eingabe bestätigt und es erscheint die Anzeige der aktuell gespeicherten Position.
③	<pre>MO.POS (P4) X: +2.98 Y: +354.10 Z: +132.00</pre>	 →  →  [STEP/MOVE]-Taste gedrückt halten	Es ertönt ein Summton und eine Bestätigungsabfrage wird angezeigt.
④	<pre>MO.POS (P4) X: +2.98 Y: +35.10 DELETE?</pre>	 → 	Die Koordinaten der Position 4 werden gelöscht. Es ertönt ein weiterer Summton.
⑤	<pre>MO.POS (P4) X: ----- Y: ----- Z: -----</pre>		Der Löschvorgang ist ausgeführt.

Tab. 7-14: Beispiel zum Löschen von Positionsdaten

Programm speichern

Neu erstellte oder überarbeitete Programme werden mit einer der folgenden Operationen gespeichert.

- Drücken Sie die [MENU]-Taste. Das Hauptmenü wird angezeigt.
- Stellen Sie den T/B [ENABLE/DISABLE]-Schalter in die Position DISABLE.

HINWEIS

Wird bei angezeigtem Programmeingabe-Bildschirm die Spannungsversorgung abgeschaltet, werden die eingegebenen Programmabschnitte gelöscht.

7.2.3 Roboterprogramm testen

Nach der Programmerstellung sollten Sie das Programm „testen“. Mit Testen ist die Suche nach und die Beseitigung von Programmfehlern gemeint.

Das Testen des Programms erfolgt mit der Teaching Box.



ACHTUNG:

Testen Sie unbedingt jedes Roboterprogramm vor einem automatischen Betriebseinsatz!

Programm schrittweise ausführen (vorwärts)

Die Programmausführung erfolgt zeilenweise in Vorwärtsrichtung. Betätigen Sie bei der Ausführung der folgenden Schritte den Totmannschalter auf der Rückseite der Teaching Box.

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre>MO . POS (P1) X: +132.30 Y: +354.10 Z: +132.00</pre>		Nach der Tastenbetätigung wird das Menü zur Editierung der Befehle angezeigt.
②	<pre>PR:1 ST:1 LN:10 10 MOV P1 CODE EDIT</pre>	oder →	Der Roboter startet die Verfahrbewegung. Wird die [INP/EXE]-Taste während der Bewegung losgelassen, stoppt der Roboter.
③	<pre>PR:1 ST:2 LN:20 20 MOV P2 CODE EDIT</pre>	oder →	Nach Abarbeitung einer Zeile wird die nächste Zeile angezeigt. Die schrittweise Ausführung des Programms erfolgt in derselben Weise.
④	<pre>PR:1 ST:13 LN:130 130 END CODE EDIT</pre>		Ist die Roboterbewegung oder die Programmausführung fehlerhaft, gehen Sie wie unter „Programm schrittweise ausführen (rückwärts)“ vor.

Tab. 7-15: Schrittweise Ausführung eines Programms (vorwärts)



ACHTUNG:




Achten Sie bei der schrittweisen Ausführung des Programms genau auf die Bewegung des Roboters. Treten während der Roboterbewegung Unregelmäßigkeiten auf (z. B. Kollisionsgefahr mit umliegenden Einrichtungen usw.), lassen Sie die [INP/EXE] oder den Totmannschalter los oder drücken Sie den Totmannschalter ganz durch, um den Roboter zu stoppen.

Beschreibung

- **Schrittbetrieb**
 Im Schrittbetrieb wird das Roboterprogramm zeilenweise abgearbeitet. Die Verfahrensgeschwindigkeit ist niedrig und der Roboter stoppt nach Abarbeitung jeder Zeile, um eine Überprüfung der Programmfunktionen und der Verfahrensbewegung zu ermöglichen. Während des Schrittbetriebs leuchtet die grüne LED des START-Tasters am Steuergerät.
- **Stoppen des Roboters im Betrieb**
 - durch Betätigung des NOT-HALT-Schalters
 Die Servoversorgung wird abgeschaltet und der Roboter stoppt sofort. Setzen Sie zur Überprüfung des Betriebs den Fehler zurück, schalten Sie die Servoversorgung wieder ein und führen Sie das Programm im Schrittbetrieb aus.
 - durch Loslassen oder Durchdrücken des Totmannschalters
 Die Servoversorgung wird abgeschaltet und der Roboter stoppt sofort. Es erscheint die Fehlermeldung 2000. Setzen Sie zur Überprüfung des Betriebs den Fehler zurück, betätigen Sie den Totmannschalter bis zur Mittelstellung, schalten Sie die Servoversorgung über die Taste [SVO ON] wieder ein und führen Sie das Programm im Schrittbetrieb aus.
 - durch Loslassen der [EXE]-Taste
 Die Ausführung des Schrittes wird unterbrochen. Die Servoversorgung wird nicht abgeschaltet. Betätigen Sie zum Fortsetzen der Programmausführung die [EXE]-Taste.

Programm schrittweise ausführen (rückwärts)

Nach Abarbeitung einer Zeile wird die vorherige Zeile aufgerufen und ausgeführt. Die Funktion kann nur bei Interpolationsbefehlen verwendet werden. Die maximale Anzahl der Zeilen für die Programmausführung in Rückwärtsrichtung ist 4.

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre>MO . POS (P1) X: +132.30 Y: +354.10 Z: +132.00</pre>		Nach der Tastenbetätigung wird das Menü zur Editierung der Befehle angezeigt.
②	<pre>PR:1 ST:1 LN:10 10 MOV P1 CODE EDIT</pre>		Starten Sie die schrittweise Programmausführung in Vorwärtsrichtung wie zuvor beschrieben.
③	<pre>PR:1 ST:2 LN:20 20 MOV P2 CODE EDIT</pre>		Nach Abarbeitung einer Zeile wird die vorherige Zeile aufgerufen.
④	<pre>PR:1 ST:1 LN:10 10 MOV P1 CODE EDIT</pre>		Die vorherige Zeile wird angezeigt. Wird die [INP/EXE]-Taste während der Bewegung losgelassen, stoppt der Roboter.

Tab. 7-16: Schrittweise Ausführung eines Programms (rückwärts)

7.3 Servospannung ein-/ausschalten

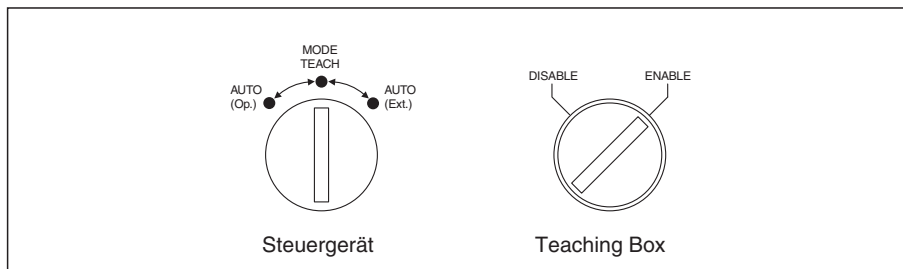
Aus Sicherheitsgründen kann die Servospannung im Teach-Modus nur bei betätigtem Totmannschalter eingeschaltet werden. Betätigen Sie den Totmannschalter bis zur Mittelstellung, bevor Sie die Servospannung einschalten.

HINWEIS

Die Bremsen werden automatisch aktiviert, wenn die Servospannung ausgeschaltet wird.

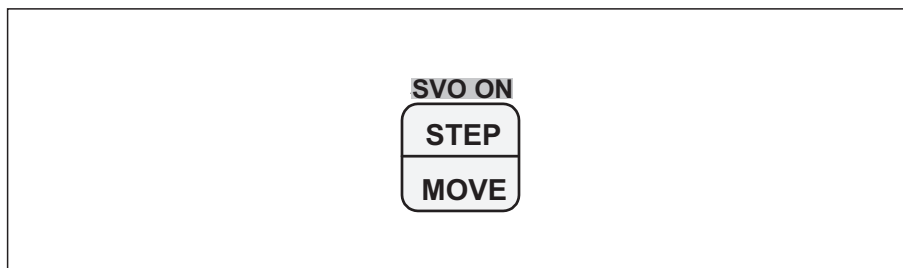
Servospannung über die [SVO ON]-Taste der Teaching Box einschalten

- ① Stellen Sie den [MODE]-Schalter des Steuergerätes auf „TEACH“ und den [ENABLE/DISABLE]-Schalter der Teaching Box auf „ENABLE“.



R000713C







- ② Betätigen Sie die [SVO ON]-Taste ([STEP/MOVE]-Taste), um die Servospannung einzuschalten.



step_mov

Servospannung über die Teaching Box einschalten

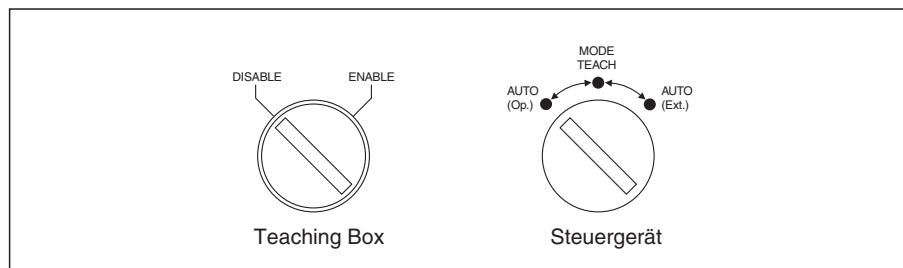
Stellen Sie den [MODE]-Schalter des Steuergerätes auf „TEACH“ und den [ENABLE/DISABLE]-Schalter der Teaching Box auf „ENABLE“.

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<div style="border: 1px solid black; padding: 5px;"> <MENU> 1. TEACH 2. RUN 3. FILE 4. MONI 5. MAINT 6. SET </div>		Das Menü RUN wird durch Eingabe der Ziffer „2“ aufgerufen.
②	<div style="border: 1px solid black; padding: 5px;"> <RUN> 1. SERVO </div>		Der Menüpunkt SERVO wird ausgewählt.
③	<div style="border: 1px solid black; padding: 5px;"> <SERVO> SERVO ON (■) 0: OFF 1: ON </div>	 →  oder  → 	Die Servospannung wird ein- oder ausgeschaltet (0: AUS, 1: EIN).
④	<div style="border: 1px solid black; padding: 5px;"> <SERVO> SERVO ON (■) 0: OFF 1: ON </div>		Die Servospannung ist jetzt ein- oder ausgeschaltet.

Tab. 7-17: Servospannung ein-/ausschalten

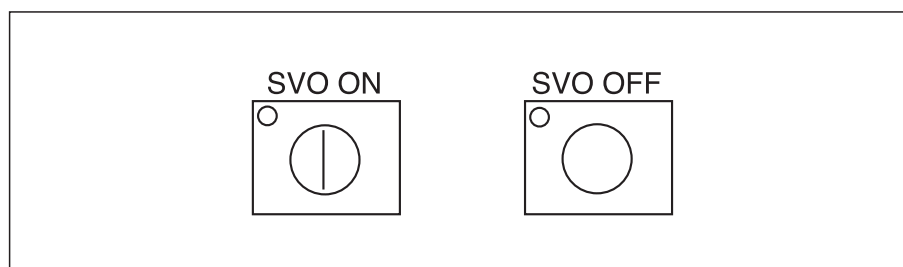
Servospannung über das Steuergerät einschalten

- ① Stellen Sie den [ENABLE/DISABLE]-Schalter der Teaching Box auf „DISABLE“ und den [MODE]-Schalter des Steuergerätes auf „AUTO (Op.)“.



R000707C

- ② Betätigen Sie die [SVO ON]-Taste, um die Servospannung einzuschalten. Die LED auf der [SVO ON]-Taste leuchtet. Betätigen Sie die [SVO OFF]-Taste, um die Servospannung auszuschalten. Die LED auf der [SVO OFF]-Taste leuchtet.



R000715C

7.4 Automatikbetrieb

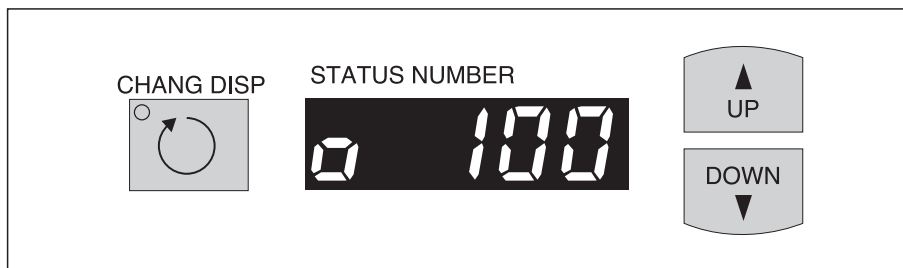
7.4.1 Geschwindigkeit einstellen

Die Geschwindigkeit kann über die Teaching Box oder das Steuergerät festgelegt werden. Die aktuelle Arbeitsgeschwindigkeit ergibt sich dabei aus:

$$\text{Aktuelle Arbeitsgeschwindigkeit} = \text{Einstellwert über Steuergerät (Teaching Box)} \times \text{Einstellwert im Programm}$$

Einstellung über das Steuergerät

- ① Betätigen Sie zweimal die Taste [CHNG DISP], um den Wert der Geschwindigkeitsübersteuerung „OVERRIDE“ anzuzeigen.
- ② Stellen Sie die Geschwindigkeitsübersteuerung über die Tasten ▼ und ▲ ein. Der Wert kann in folgenden Schritten eingestellt werden: 10 → 20 → 30 → 40 → 50 → 60 → 70 → 80 → 90 → 100 %.



R000705C

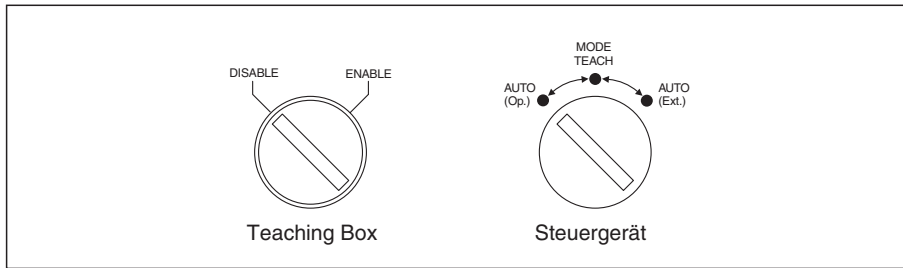
Einstellung über die Teaching Box

Die Einstellung der Geschwindigkeit über die Teaching Box erfolgt über die Tasten [STEP/MOVE] und [+ /FORWD] oder [STEP/MOVE] und [- /BACKWD]. Der Wert kann in folgenden Schritten eingestellt werden: LOW → HIGH → 3 → 5 → 10 → 30 → 50 → 70 → 100 %. Im Modus LOW und HIGH ist nur ein Schrittbetrieb möglich.

Die aktuell eingestellte Geschwindigkeit erscheint in der oberen rechten Ecke der Anzeige.

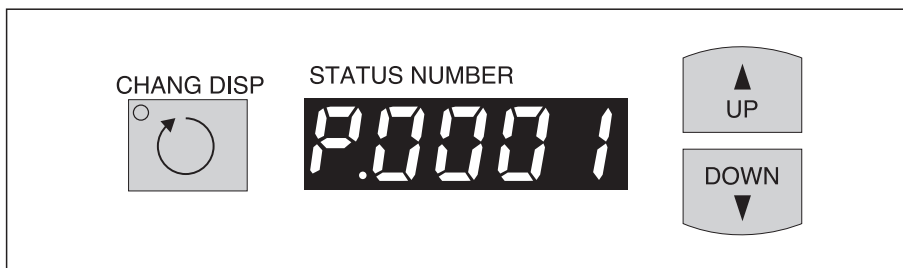
7.4.2 Auswahl der Programmnummer

- ① Stellen Sie den [ENABLE/DISABLE]-Schalter der Teaching Box auf „DISABLE“ und den [MODE]-Schalter des Steuergerätes auf „AUTO (Op.)“.



R000707C

- ② Betätigen Sie die [CHANG DISP]-Taste, um die Programmnummer anzuzeigen.
- ③ Wählen Sie die Programmnummer über die Tasten ▼ und ▲.



R000708C

7.4.3 Starten des Automatikbetriebs



GEFAHR:

Stellen Sie vor einem Start des Automatikbetriebs sicher, dass die folgenden Bedingungen erfüllt sind:

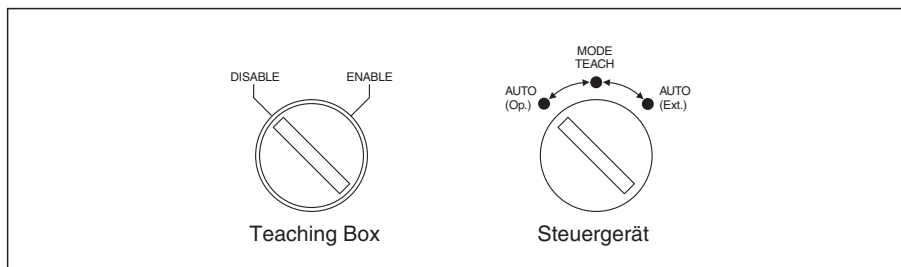
Es dürfen sich keine Personen im Umkreis des Roboters befinden.

Die Tür der Sicherheitsumzäunung muss geschlossen sein. Ein Öffnen der Tür muss zur sofortigen Unterbrechung des Roboterbetriebs führen.

Es dürfen sich keine Gegenstände innerhalb des Bewegungsbereiches des Roboters befinden, die zum Betrieb des Roboters nicht unbedingt notwendig sind (z. B. Werkzeuge usw.).

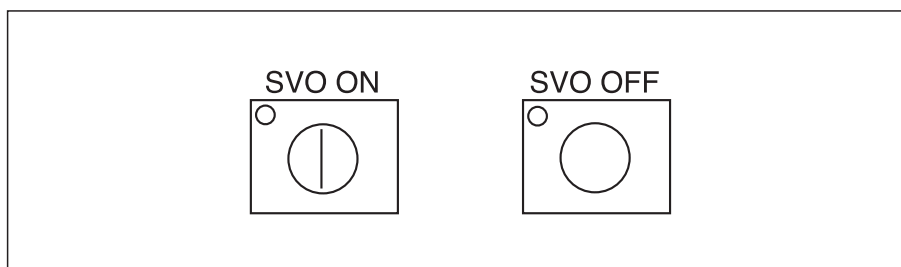
Überprüfen Sie die einwandfreie Funktion des Programms vor Ausführung des Automatikbetriebs im Schrittbetrieb.

- ① Stellen Sie den [ENABLE/DISABLE]-Schalter der Teaching Box auf „DISABLE“ und den [MODE]-Schalter des Steuergerätes auf „AUTO (Op.)“.



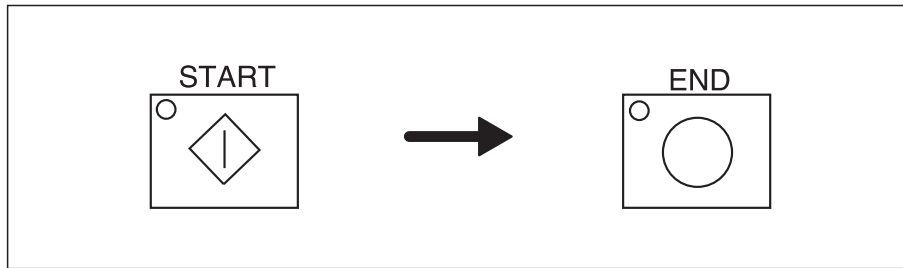
R000707C

- ② Betätigen Sie die [SVO ON]-Taste, um die Servospannung einzuschalten. Die LED auf der [SVO ON]-Taste leuchtet. Betätigen Sie die [SVO OFF]-Taste, um die Servospannung auszuschalten. Die LED auf der [SVO OFF]-Taste leuchtet.



R000715C

- ③ Betätigen Sie die [START]-Taste, um den kontinuierlichen Automatikbetrieb zu starten. Bei Betätigung der [END]-Taste stoppt das Programm nach Ausführung eines Zyklus.



R000709C

**Gefahr:**

Prüfen Sie vor dem Starten des Automatikbetriebs, dass die korrekte Programmnummer ausgewählt wurde.

Unterbrechen Sie den Automatikbetrieb sofort durch Betätigung des NOT-HALT-Schalters, falls Ihnen Unregelmäßigkeiten auffallen.

7.4.4 Stoppen des Automatikbetriebs

Der Automatikbetrieb kann über das Steuergerät oder über die Teaching Box durch Betätigung der [STOP]-Taste gestoppt werden.

Bei Betätigung der [STOP]-Taste wird das Programm unterbrochen und der Roboter bis zum Stillstand abgebremst.

HINWEIS | Zur Ausführung der Stoppfunktion ist kein Betriebsrecht erforderlich.

7.4.5 Fortsetzung des Automatikbetriebs aus dem Stoppzustand

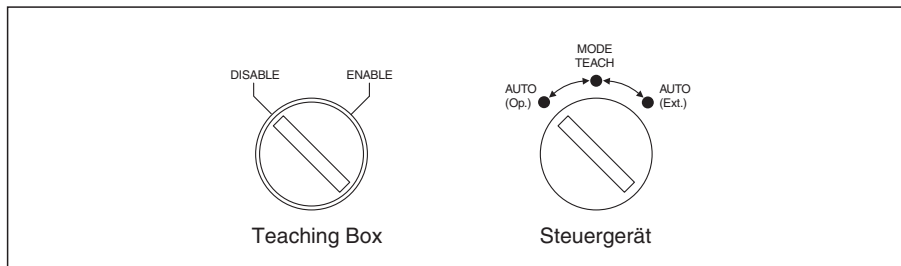
Zum Fortsetzen des Automatikbetriebs aus dem Stoppzustand führen Sie die Schritte aus, die im Absatz „Starten des Automatikbetriebs“ (siehe Seite 7-21) beschrieben werden.

7.4.6 Programm zurücksetzen

Der Stoppzustand des Programmes wird aufgehoben und die Programmsteuerung springt zum Programmstart.

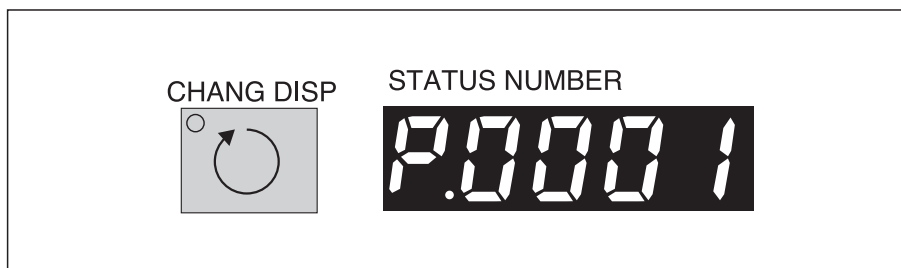
Rücksetzen über das Steuergerät

- ① Stellen Sie den [ENABLE/DISABLE]-Schalter der Teaching Box auf „DISABLE“ und den [MODE]-Schalter des Steuergerätes auf „AUTO (Op.)“.



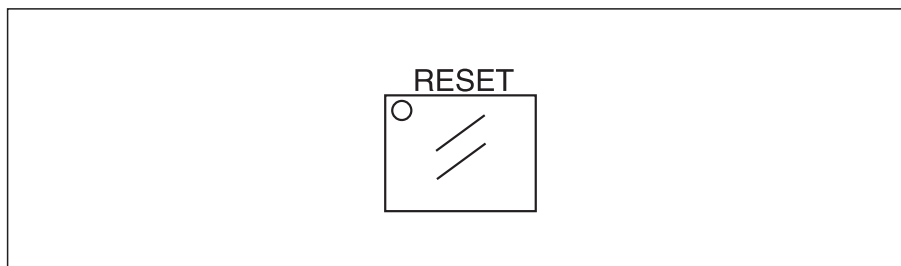
R000707C

- ② Betätigen Sie die [CHANG DISP]-Taste, um die Programmnummer anzuzeigen.



R000711C

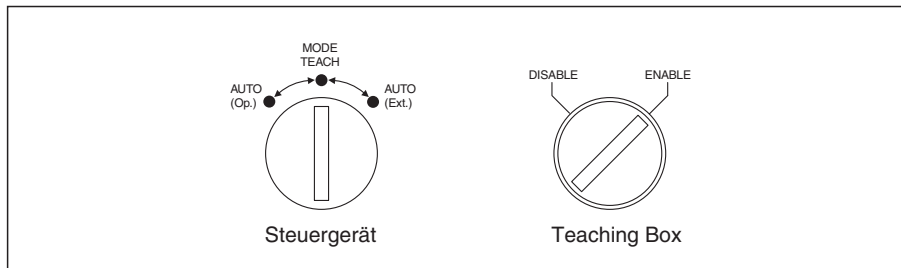
- ③ Betätigen Sie die [RESET]-Taste. Die STOP-LED erlischt.



R000712C

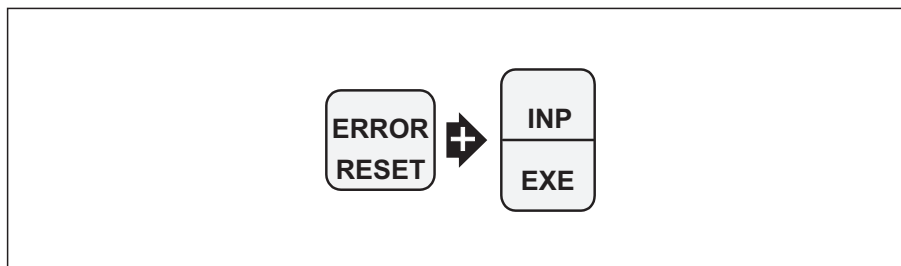
Rücksetzen über die Teaching Box

- ① Stellen Sie den [MODE]-Schalter des Steuergerätes auf „TEACH“ und den [ENABLE/DISABLE]-Schalter der Teaching Box auf „ENABLE“.



R000713C

- ② Betätigen Sie die [INP/EXE]-Taste, während Sie die [ERROR/RESET]-Taste gedrückt halten.



R000714C

HINWEISE









Während der Programmausführung kann ein Programm nicht zurückgesetzt werden. Das Programm muss zuerst gestoppt werden. Vor Ausführung des Rücksetzvorgangs über das Steuergerät muss die entsprechende Programmnummer im Display angezeigt werden.

Die LED der [STOP]-Taste erlischt, wenn das Programm zurückgesetzt ist.

7.5 Programmverwaltungsfunktionen

7.5.1 Programmverzeichnis anzeigen

Die Funktion ermöglicht die Anzeige der im Steuergerät gespeicherten Programme und deren Eigenschaften.

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT 6 . SET</pre>		Das Menü FILE wird durch Eingabe der Ziffer „3“ aufgerufen.
②	<pre><FILE> 1 . DIR 2 . COPY 3 . RENAME 4 . DELETE</pre>		Der Menüpunkt DIR wird ausgewählt. Oben rechts erscheint die Anzahl der gespeicherten Programme.
③	<pre><DIR> 7 1 99-10-10 2 99-11-29 5 99-12-08</pre>	 	Mit den Cursortasten können die weiteren Programmnamen zur Anzeige gebracht werden.
④	<pre><DIR> 7 10 99-12-10 13 99-08-29 17 99-09-10</pre>		Es wird die Zeit der Programmerstellung angezeigt (Stunde, Minute, Sekunde). Über die Taste [DEL ←] gelangen Sie zur vorherigen Anzeige zurück.
⑤	<pre><DIR> 7 10 03:58:02 13 23:05:32 17 15:48:39</pre>		Es wird die Dateigröße in Bytes angezeigt. Oben rechts erscheint die Gesamtspeicherbelegung.
⑥	<pre><DIR> 25639 10 348 13 1978 17 3873</pre>		Es wird angezeigt, ob ein Programm geschützt ist.
⑦	<pre><DIR> PROTECT 10 OFF (1) 13 ON (1)</pre>		Es wird angezeigt, ob eine Variable geschützt ist.
⑧	<pre><DIR> POS . PROTECT 10 ON (1) 13 ON (1)</pre>		Eine detaillierte Beschreibung zum Schutz von Programmen finden Sie auf der nächsten Seite.

Tab. 7-18: Beispiel zum Anzeigen des Programmverzeichnisses


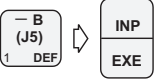
7.5.2 Programm schützen

Programmschutzfunktion

Die Funktion verhindert ein unbeabsichtigtes Löschen eines Programms und Programmänderungen.

Rufen Sie zuerst das Programmverzeichnis auf (siehe Seite 7-25).

Im folgenden Beispiel wird das Programm Nr. 2 geschützt:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><DIR> PROTECT 1 OFF (0) 2 OFF (0)</pre>		Bewegen Sie den Cursor zu dem Programm, dessen Schreibschutz eingestellt werden soll.
②	<pre><DIR> PROTECT 1 OFF (0) 2 OFF (0)</pre>		Der Programmschutz für Programm Nr. 2 wird eingeschaltet.
③	<pre><DIR> PROTECT 1 OFF (0) 2 ON (1)</pre>		Programm Nr. 2 ist jetzt geschützt.

Tab. 7-19: Beispiel zum Schützen eines Programms

Beschreibung


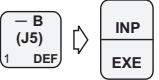
- Programme werden gegen die Operationen DELETE, RENAME und Programmänderungen geschützt.
- Beim Kopieren eines Programms wird der Schutzstatus nicht mitkopiert.
- Beim Löschen des Speichers über „INITIALIZATION“ (siehe Abschnitt 7.7.2) wird der Schutzstatus ignoriert und das Programm gelöscht.

Variablenschutzfunktion

Die Funktion verhindert ein unbeabsichtigtes Löschen und Ändern von Variablen.

Rufen Sie zuerst das Programmverzeichnis auf (siehe Seite 7-25).

Im folgenden Beispiel werden die Variablen von Programm Nr. 2 geschützt:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><DIR> POS . PROTECT 1 OFF (0) 2 OFF (0)</pre>		Bewegen Sie den Cursor zu dem Programm, dessen Variablen geschützt werden sollen.
②	<pre><DIR> POS . PROTECT 1 OFF (0) 2 OFF (0)</pre>		Der Variablenschutz für Programm Nr. 2 wird eingeschaltet.
③	<pre><DIR> POS . PROTECT 1 OFF (0) 2 ON (1)</pre>		Die Variablen in Programm Nr. 2 sind jetzt geschützt.

Tab. 7-20: Beispiel zum Schützen von Variablen





Beschreibung

- Variablen werden gegen das Überschreiben durch Positionsdaten, gegen Änderungen und gegen Substitutionen bei fehlerhafter Programmausführung geschützt.
- Beim Kopieren eines Programms wird der Schutzstatus nicht mitkopiert.
- Beim Löschen des Speichers über „INITIALIZATION“ (siehe Seite 7.7.2) wird der Schutzstatus ignoriert und die Variablen gelöscht.

7.5.3 Programm kopieren

Die Funktion dient zum Kopieren eines Roboterprogramms.

Im folgenden Beispiel wird das Programm Nr. 1 kopiert und unter dem neuen Programmnamen Nr. 5 nochmals abgespeichert:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 .TEACH 2 .RUN 3 .FILE 4 .MONI 5 .MAINT 6 .SET</pre>		Das Menü FILE wird durch Eingabe der Ziffer „3“ aufgerufen.
②	<pre><FILE> 1 .DIR 2 .COPY 3 .RENAME 4 .DELETE</pre>		Der Menüpunkt COPY wird ausgewählt.
③	<pre><COPY> FROM () TO () INPUT SOURCE</pre>		Der Name (Nr. 1) des Programms, das kopiert werden soll, wird eingegeben.
④	<pre><COPY> FROM (1) TO (5) INPUT DEST.</pre>		Der neue Programmname (Nr. 5) wird eingegeben. Anschließend wird die Eingabe bestätigt und der Kopiervorgang wird ausgeführt.
⑤	<pre><FILE> 1 .DIR .COPY 3 .RENAME 4 .DELETE</pre>		Nach Abschluss des Kopiervorgangs wird das Menü FILE angezeigt.

Tab. 7-21: Beispiel zum Kopieren eines Roboterprogramms





Beschreibung

- Es wird eine Fehlermeldung angezeigt, wenn derselbe Programmname zweimal angegeben wird, d. h. wenn Quelle und Ziel eines Kopiervorgangs identisch sind.
- Der Schutzstatus von Programmen und Variablen wird nicht mitkopiert.
- Es wird eine Fehlermeldung angezeigt, wenn das Zielprogramm belegt ist.

7.5.4 Programmnamen ändern

Die Funktion dient zum Ändern eines Programmnamens.

Im folgenden Beispiel wird der Programmname von „1“ auf „5“ geändert:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 .TEACH 2 .RUN 3 .FILE 4 .MONI 5 .MAINT 6 .SET</pre>		Das Menü FILE wird durch Eingabe der Ziffer „3“ aufgerufen.
②	<pre><FILE> 1 .DIR 2 .COPY 3 .RENAME 4 .DELETE</pre>		Der Menüpunkt RENAME wird ausgewählt.
③	<pre><RENAME> FROM (1) TO () INPUT SOURCE</pre>		Der Name (Nr. 1) des Programms, das umbenannt werden soll, wird eingegeben.
④	<pre><RENAME> FROM (1) TO (5) INPUT DEST.</pre>		Der neue Programmname (Nr. 5) wird eingegeben. Anschließend wird die Eingabe bestätigt und der Änderungsvorgang wird ausgeführt.
⑤	<pre><FILE> 1 .DIR 2 .COPY 3 .RENAME 4 .DELETE</pre>		Nach Abschluss des Änderungsvorgangs wird das Menü FILE angezeigt.

Tab. 7-22: Beispiel zum Ändern eines Programmnamens




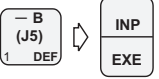
Beschreibung

- Wird derselbe Programmname zweimal angegeben, erfolgt eine Fehlermeldung.
- Der Programmname eines geschützten Programms oder eines Programms mit geschützten Variablen kann nicht geändert werden. Schalten Sie gegebenenfalls die Schutzfunktion aus.

7.5.5 Programm löschen

Die Funktion dient zum Löschen eines gespeicherten Programms.

Im folgenden Beispiel wird Programm Nr. 1 gelöscht:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 .TEACH 2 .RUN 3 .FILE 4 .MONI 5 .MAINT 6 .SET</pre>		Das Menü FILE wird durch Eingabe der Ziffer „3“ aufgerufen.
②	<pre><FILE> 1 .DIR 2 .COPY 3 .RENAME 4 .DELETE</pre>		Der Menüpunkt DELETE wird ausgewählt.
③	<pre><DELETE> DELETE (1) INPUT DEL.FILE</pre>		Der Name (Nr. 1) des Programms, das gelöscht werden soll, wird eingegeben. Nach der Eingabe wird eine Bestätigungsabfrage angezeigt.
④	<pre><DELETE> DELETE 1 OK? (1) 1 : EXECUTE</pre>		Die Abfrage wird durch Eingabe der Ziffer „1“ bestätigt.
⑤	<pre><FILE> 1 .DIR 2 .COPY 3 .RENAME 4 .DELETE</pre>		Nach Abschluss des Änderungsvorgangs wird das Menü FILE angezeigt.

Tab. 7-23: Beispiel zum Löschen eines Programms

HINWEIS




Ein geschütztes Programm oder eine Programm mit geschützten Variablen kann nicht gelöscht werden. Schalten Sie gegebenenfalls die Schutzfunktion aus.

7.6 Monitor-Funktionen

7.6.1 Monitor-Funktion für Eingangssignale

Die Funktion ermöglicht die Anzeige des Eingangssignalstatus.

Im folgenden Beispiel wird der Status der Eingangsbits 8 bis 15 angezeigt:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT 6 . SET</pre>		Das Menü MONITOR wird durch Eingabe der Ziffer „4“ aufgerufen.
②	<pre><MONI> 1 . INPUT 2 . OUTPUT 3 . VAR 4 . ERROR</pre>		Der Menüpunkt INPUT wird ausgewählt.
③	<pre><INPUT> NUMBER (8) BIT: 76543210 DATA (00000000)</pre>		Die Ziffer „8“ wird eingegeben und anschließend wird die Eingabe bestätigt.
④	<pre><INPUT> NUMBER (8) BIT: 54321098 DATA (01001011)</pre>		Der Bitstatus der Eingangsbits 8 bis 15 wird angezeigt.

Tab. 7-24: Beispiel zum Überprüfen der Bitzustände der Eingangssignale











HINWEIS

Die Anzeige der Eingangsbitzustände kann auch ohne zugewiesene Betriebsrechte der Teaching Box erfolgen.

7.6.2 Monitor-Funktion für Ausgangssignale

Die Funktion ermöglicht die Anzeige und Einstellung der Ausgangssignalzustände.

Im folgenden Beispiel wird das 8. Ausgangsbit eingeschaltet:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 .TEACH 2 .RUN 3 .FILE 4 .MONI 5 .MAINT 6 .SET</pre>		Das Menü MONITOR wird durch Eingabe der Ziffer „4“ aufgerufen.
②	<pre><MONI> 1 .INPUT 2 .OUTPUT 3 .VAR 4 .ERROR</pre>		Der Menüpunkt OUTPUT wird ausgewählt.
③	<pre><OUTPUT> NUMBER (8) BIT: 76543210 DATA (00000000)</pre>	  	Die Ziffer „8“ wird eingegeben und anschließend wird die Eingabe bestätigt. Der Bitstatus der Ausgangsbits 8 bis 15 wird angezeigt.
④	<pre><OUTPUT> NUMBER (8) BIT: 54321098 DATA (01101000)</pre>	 	Bewegen Sie den Cursor zum 8. Bit.
⑤	<pre><OUTPUT> NUMBER (8) BIT: 54321098 DATA (01101001)</pre>	  	Der Signalzustand von Bit 8 wird auf „1“ gesetzt. Anschließend wird die Dateneingabe bestätigt.
⑥	<pre><OUTPUT> NUMBER (8) BIT: 54321098 DATA (01101001)</pre>		Der Signalzustand von Bit 8 ist auf „1“ gesetzt.

Tab. 7-25: Beispiel zum Einstellen der Bitzustände der Ausgangssignale

7.6.3 Monitor-Funktion für Variable

Die Funktion ermöglicht die Anzeige und Einstellung von Variablen.

Im folgenden Beispiel wird die numerische Variable M8 in Programm Nr. 1 von „2“ auf „5“ gesetzt:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT 6 . SET</pre>		Das Menü MONITOR wird durch Eingabe der Ziffer „4“ aufgerufen.
②	<pre><MONI> 1 . INPUT 2 . OUTPUT 3 . VAR 4 . ERROR</pre>		Der Menüpunkt VAR wird ausgewählt.
③	<pre><VAR> (1) SELECT</pre>		Die Ziffer „1“ zur Auswahl des Programms Nr. 1 wird eingegeben.
④	<pre><VAR> V . NAME (M) DATA () SET V . NAME</pre>		Der Variablenname „M8“ wird eingegeben.
⑤	<pre><VAR> V . NAME (M8) DATA (+ 2) SET V . NAME</pre>		Der aktuelle Wert „+2“ wird durch den neuen Wert „+5“ überschrieben.
⑥	<pre><VAR> V . NAME (M8) DATA (+ 5) SET V . NAME</pre>		Der Wert der Variablen M8 ist jetzt „+5“.

Tab. 7-26: Beispiel zum Ändern von Variablenwerten





HINWEISE

Die Anzeige von Variablen kann auch ohne zugewiesene Betriebsrechte der Teaching Box erfolgen. Für eine Änderung von Variablen benötigen Sie jedoch die Betriebsrechte.

Roboterstatusvariablen können nicht direkt angezeigt werden. Schreiben Sie den Wert zuerst in eine Programmvariable und zeigen Sie die Programmvariable an.

7.6.4 Liste der aufgetretenen Fehlermeldungen

Die bisher aufgetretenen Fehlermeldungen werden in einer Liste angezeigt. Diese Funktion ist für eine Störungssuche sehr hilfreich.

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 .TEACH 2 .RUN 3 .FILE 4 .MONI 5 .MAINT 6 .SET</pre>		Das Menü MONITOR wird durch Eingabe der Ziffer „4“ aufgerufen.
②	<pre><MONI> 1 .INPUT 2 .OUTPUT 3 .VAR 4 .ERROR</pre>		Der Menüpunkt ERROR wird ausgewählt.
③	<pre><ERROR HISTORY>- 1 99-08-10 10:20 2000 SERVO OFF</pre>	 	Mit den Cursortasten können die weiteren Fehlermeldungen zur Anzeige gebracht werden.
④	<pre><ERROR HISTORY>- 2 99-08-10 10:12 3110 ARGUMENT</pre>		Anzeige des folgenden Fehlers

Tab. 7-27: Beispiel zum Anzeigen der Liste der aufgetretenen Fehlermeldungen

HINWEIS

Die Anzeige von Fehlermeldungen kann auch ohne zugewiesene Betriebsrechte der Teaching Box erfolgen.

7.7 Zusatzfunktionen

7.7.1 Parameter einstellen

Die Funktionen der parallelen Ein-/Ausgabeschchnittstelle, die Werkzeuglänge usw. sind über Parameter festgelegt. Diese Parameter können eingestellt werden.

Im folgenden Beispiel wird der Wert für die Z-Achse des Parameters „MEXTL (Werkzeugdaten)“ von 0 auf 100 mm gesetzt:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT 6 . SET</pre>		Das Menü MAINTENANCE wird durch Eingabe der Ziffer „5“ aufgerufen.
②	<pre><MAINT> 1 . PARAM 2 . INIT 3 . BRAKE 4 . ORIGIN 5 . POWER</pre>		Das Menü PARAMETER wird ausgewählt.
③	<pre><PARAM> () () () SET PARAM.NAME</pre>		Die Zeichen „MEXTL“ werden eingegeben. Anschließend wird der Cursor zum Eingabefeld für die Achsennummer bewegt.
④	<pre><PARAM> (MEXTL) () () SET ELEMENT</pre>		Es wird die Achsennummer „3“ eingegeben.
⑤	<pre><PARAM> (MEXTL) (3) (+0.00) SET ELEMENT</pre>		Der aktuelle Wert „+0.00“ wird angezeigt. Anschließend wird der Cursor zum Feld für die Eingabe des neuen Wertes bewegt.
⑥	<pre><PARAM> (MEXTL) (3) (+0.00) SET ELEMENT</pre>		Der Neue Wert „+100“ wird eingegeben.
⑦	<pre><PARAM> (MEXTL) (3) (+100.00) SET ELEMENT</pre>		Schalten Sie die Spannungsversorgung aus und wieder ein, damit der neue Wert wirksam werden kann.




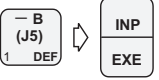
Tab. 7-28: Beispiel zum Einstellen eines Parameters

HINWEIS

Damit ein neuer Parameterwert wirksam wird, muss die Spannungsversorgung des Steuergerätes aus- und wieder eingeschaltet werden.

7.7.2 Alle gespeicherten Programme löschen

Diese Funktion löscht alle gespeicherten Programme mit einem Bedienschritt:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT 6 . SET</pre>		Das Menü MAINTENANCE wird durch Eingabe der Ziffer „5“ aufgerufen.
②	<pre><MAINT> 1 . PARAM 2 . INIT 3 . BRAKE 4 . ORIGIN 5 . POWER</pre>		Das Menü INIT wird ausgewählt.
③	<pre><INIT> INIT (█) 1 . PROGRAM 2 . BATT.</pre>		Die Ziffer „1“ wird eingegeben. Anschließend wird die Eingabe bestätigt.
④	<pre><INIT> PROGRAM OK? (█) 1 : EXECUTE</pre>		Die Abfrage wird durch Eingabe der Ziffer „1“ bestätigt. Der Löschvorgang wird ausgeführt.
⑤	<pre><INIT> INIT (█) 1 . PROGRAM 2 . BATT.</pre>		Nach Ausführung des Löschvorgangs wird das Menü INIT angezeigt.

Tab. 7-29: Beispiel zum Löschen aller gespeicherten Programme

HINWEIS

Es werden auch geschützte Programme und Programme mit geschützten Variablen gelöscht.

7.7.3 Batteriezähler zurücksetzen

Der Batteriezähler erfasst die Betriebszeit der Batterien im Roboterarm und im Steuergerät und dient als Referenz für die Warnmeldung zum Austausch der Batterien. Setzen Sie daher nach dem Austauschen der Batterien unbedingt den Batteriezähler zurück.

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT 6 . SET</pre>		Das Menü MAINTENANCE wird durch Eingabe der Ziffer „5“ aufgerufen.
②	<pre><MAINT> 1 . PARAM 2 . INIT 3 . BRAKE 4 . ORIGIN 5 . POWER</pre>		Das Menü INIT wird ausgewählt.
③	<pre><INIT> INIT (█) 1 . PROGRAM 2 . BATT.</pre>		Die Ziffer „2“ wird eingegeben. Anschließend wird die Eingabe bestätigt.
④	<pre><INIT> BATT. OK? (█) 1 : EXECUTE</pre>		Die Abfrage wird durch Eingabe der Ziffer „1“ bestätigt. Der Rücksetzvorgang wird ausgeführt.
⑤	<pre><INIT> INIT (█) 1 . PROGRAM 2 . BATT.</pre>		Nach Ausführung des Rücksetzvorgangs wird das Menü INIT angezeigt.

Tab. 7-30: Beispiel zum Zurücksetzen des Batteriezählers

HINWEIS

Bei verbrauchter Batterie wird eine Warnmeldung ausgegeben. Die Gebrauchszeit der Batterie wird ab dem Zurücksetzen des Batteriezählers erfasst. Setzen Sie daher nach einem Austausch der Batterien den Batteriezähler zurück, um eine korrekte Erfassung der Gebrauchszeit zu gewährleisten.

7.7.4 Gelenkbremsen lösen

Die Bremsen für die Robotergelenke können bei ausgeschalteter Servospannung gelöst werden. Der Roboterarm kann dann direkt manuell bewegt werden.



ACHTUNG

Beachten Sie, dass der Roboterarm aufgrund des Eigengewichts bei gelösten Bremsen heruntersinken kann. Unterstützen Sie daher den Roboterarm vor dem Lösen der Bremsen.

Schalten Sie erst über die Teaching Box mit dem SERVO-Menü die Servos aus (siehe Abs. 7.3). Anschließend bringen Sie den Totmannschalter in die Mittelstellung und folgen den Anweisungen aus Tab. 7-32.

Achse	Anzeige auf der Teaching Box							
	1	2	3	4	5	6	7	8
J1	1	0	0	0	0	0	0	0
J2	0	1	0	0	0	0	0	0
J3	0	0	1	0	0	0	0	0
J4	0	0	0	1	0	0	0	0
J5	0	0	0	0	1	0	0	0
J6	0	0	0	0	0	1	0	0

Tab. 7-31:
Gelenkbremsen und Anzeige



Im folgenden Beispiel wird beim Roboter RP-1AH die Gelenkbremse der Achse J1 gelöst:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT 6 . SET</pre>		Das Menü MAINTENANCE wird durch Eingabe der Ziffer „5“ aufgerufen.
②	<pre><MAINT> 1 . PARAM 2 . INIT 3 . BRAKE 4 . ORIGIN 5 . POWER</pre>		Das Menü BRAKE wird ausgewählt.
③	<pre><BRAKE>12345678 BRAKE (00000000) 0 : LOCK 1 : FREE</pre>		Es werden die Achsen auf „1“ gesetzt, deren Bremsen gelöst werden sollen.
④	<pre><BRAKE>12345678 BRAKE (10000000) 0 : LOCK 1 : FREE</pre>		Die Bremsen sind gelöst, solange die Tasten betätigt sind. Wird eine der Tasten losgelassen, sind alle Bremsen wieder aktiv.

Tab. 7-32: Beispiel zum Lösen der Gelenkbremsen

7.7.5 Batterie und Einschaltzeit anzeigen

Die Betriebszeit des Steuergerätes und die verbleibende Lebensdauer der Batterie in Stunden werden angezeigt.



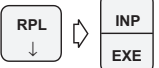
Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT 6 . SET</pre>		Das Menü MAINTENANCE wird durch Eingabe der Ziffer „5“ aufgerufen.
②	<pre><MAINT> 1 . PARAM 2 . INIT 3 . BRAKE 4 . ORIGIN 5 . POWER</pre>		Das Menü POWER wird ausgewählt.
③	<pre><HOUR DATA> Hr POWER ON: 1258 BATTERY: 4649</pre>		Es werden die Betriebszeit und die verbleibende Lebensdauer der Batterie angezeigt.

Tab. 7-33: Beispiel zum Anzeigen der Einschaltzeit und Restpufferung

7.7.6 Uhrzeit und Datum einstellen

Das Steuergerät ist mit einer internen Uhr für Uhrzeit- und Datumsfunktionen ausgerüstet. Diese Datumsfunktion nutzt das Steuergerät z. B. zum Eintragen des Erstellungsdatums in ein Programm.

Die interne Uhr sollten Sie nach der erstmaligen Inbetriebnahme des Roboters und danach in regelmäßigen Abständen kontrollieren. Nachfolgend wird das Einstellen von Uhrzeit und Datum beschrieben:

Nr.	Display-Darstellung	Tastenbetätigungen	Beschreibung
①	<pre><MENU> 1 . TEACH 2 . RUN 3 . FILE 4 . MONI 5 . MAINT . SET</pre>		Das Menü SET wird durch Eingabe der Ziffer „6“ aufgerufen.
②	<pre><SET> 1 . CLOCK</pre>		Im Menü SET wird der Menüpunkt CLOCK aufgerufen. Nach der Tastenbetätigung wird die aktuelle Uhrzeit und das aktuelle Datum angezeigt.
③	<pre><CLOCK> DATE (99-12-07) TIME (23:58:17) INPUT DATE</pre>		Stellen Sie das richtige Datum ein. Bestätigen Sie die Eingabe mit der [INP/EXE]-Taste. Wechseln Sie mit der Taste [RPL ↓] in die Zeile zur Einstellung der Uhrzeit.
③	<pre><CLOCK> DATE (99-10-07) TIME (23:58:17) INPUT DATE</pre>		Die Einstellung der Uhrzeit erfolgt in derselben Weise wie die Einstellung des Datums.

Tab. 7-34: Beispiel zum Einstellen von Uhrzeit/Datum

8 MELFA-BASIC IV

Die nachfolgenden Abschnitte enthalten eine Auflistung aller in MELFA-BASIC IV verwendeten Datentypen und deren Anwendungsmöglichkeiten.

8.1 Begriffserklärung

8.1.1 Anweisung

Eine Anweisung ist die kleinste Einheit eines Programms. Sie besteht aus einem Befehl und einem Befehlsparameter.

Beispiel ▾

MOV P1 = Anweisung

MOV = Befehl

P1 = Befehlsparameter

△

8.1.2 Angehängte Anweisung

Bei Interpolationsbefehlen ist es möglich, eine Verknüpfung an die Anweisung anzuhängen. Durch Anhängen einer Verknüpfung können bestimmte Befehle parallel zum Interpolationsbefehl ausgeführt werden. Es darf pro Zeile nur eine Verknüpfung angehängt werden.

Beispiel ▾

Folgender Befehl bewirkt, dass die Position P1 mittels Gelenk-Interpolation angefahren und gleichzeitig das Ausgangsbit 17 auf „1“ gesetzt wird:

MOV P1 WTH M_OUT(17) = 1

△

Mit Hilfe der Befehle WTH und WTHIF können Verknüpfungen an eine Anweisung angehängt werden.

8.1.3 Zeilen

Eine Zeile besteht aus einer Zeilennummer und einer Anweisung oder zwei Anweisungen, wenn zusätzlich eine Konjunktion angehängt ist.

Die Zeilenlänge darf maximal 127 Zeichen betragen. Das Zeilenendzeichen wird dabei nicht mitgezählt.

HINWEIS

In der MELFA-BASIC-Programmiersprache ist es nicht erlaubt, mehrere durch Semikolons getrennte Anweisungen in eine Zeile zu setzen, wie es bei vielen BASIC-Dialekten möglich ist.

8.1.4 Zeilennummern und Marken

Zeilennummern

Für die einwandfreie Funktion eines Programmes müssen die Zeilennummern in aufsteigender Reihenfolge angeordnet sein. Beim Abspeichern wird das Programm in dieser Reihenfolge im Speicher abgelegt. Der Wertebereich für die Zeilennummern beträgt 1 bis 32 767.

Eine Ausnahme bildet die direkte Befehlsausführung:

HINWEIS

Bei fehlender Zeilennummer wird eine Anweisung direkt nach Betätigung der Eingabetaste ausgeführt. Die Anweisung wird dabei nicht gespeichert.

Marken

Eine Marke ist ein benutzerdefiniertes Wort, das ein Sprungziel festlegt. Erzeugt wird eine Marke durch Eingabe des Asterisks-Zeichens (*) hinter der Zeilennummer und einer alphanumerischen Zeichenkette. Hierbei wird auch zwischen Groß- und Kleinbuchstaben unterschieden. Beginnt die Zeichenkette mit dem Zeichen „L“, kann als nächstes Zeichen der Unterstrich (_) verwendet werden. Das erste Zeichen muss ein Buchstabe sein. Die Länge der Sprungmarke darf maximal 8 Zeichen betragen.

Beispiel ▾

```
120 *ABLAGE;  
170 *BAND_1
```

**HINWEIS**

Für Markennamen dürfen keine reservierten Wörter (z. B. DLY, HOPEN usw.), keine Namen, die mit einem Symbol oder einer Zahl beginnen und keine Namen, die schon für eine Variable oder eine Funktion vergeben wurden, benutzt werden.

8.1.5 Zeichentypen

Die in MELFA-BASIC IV verwendbaren Zeichentypen sind in Tab. 8-1 aufgeführt.

Verwendbare Zeichen in MELFA-BASIC IV	
Alphanumerische Zeichen	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 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
Zahlen	0 1 2 3 4 5 6 7 8 9
Symbole	! " # \$ % & () * + - . , / : ; = < > ? @ ` [¥] ^ { } ~
Leerstellen	Leerzeichen
Unterstrich	_

Tab. 8-1: In MELFA-BASIC IV verwendbare Zeichen

HINWEIS

Kleinbuchstaben, die in Kommentaren oder in Zeichenketten verwendet werden, werden auch als Kleinbuchstaben abgespeichert. In allen anderen Fällen werden sie zu Großbuchstaben konvertiert, sobald das Programm gelesen wird.

8.1.6 Zeichen mit besonderer Bedeutung

Unterstrich (_)

Der Unterstrich wird bei Variablenamen als zweites Zeichen verwendet, wenn diese als programmexterne Variablen benutzt werden.

Beispiele ▾

P_CURR
M_01
M_ABC



Apostroph (')

Der Apostroph wird vor einen Kommentar gesetzt. Es hat die gleiche Funktion wie der REM-Befehl (Kennzeichnung eines Kommentars).

Beispiele ▾

100 MOV P1 'GET	GET wird als Kommentar deklariert
150 'GET PARTS	Entspricht der Zeile: 150 REM GET PARTS



Asterisks (*)

Das Asterisks-Zeichen wird vor alle Sprungmarken gesetzt. Die Sprungmarke darf maximal aus 8 Zeichen bestehen.

Beispiel ▾

200 *LADEN

Komma (,)

Das Komma dient bei Angabe mehrerer Parameter oder Suffixe zur Trennung.

Beispiel ▾

P1 = (100, 150, ...)



Punkt (.)

Der Punkt dient als Dezimalpunkt und zur Unterteilung der einzelnen Komponenten bei mehrteiligen Daten wie Positions- und Gelenkvariablen.

Beispiel ▾

M1 = P2. X Schreibt die X-Koordinate der Positionsvariablen P2 in die Variable M1



Leerzeichen

In Zeichenketten und Kommentaren wird das Leerzeichen wie jedes andere Zeichen interpretiert. Zwischen einzelnen Daten, nach Zeilennummern und Anweisungen dient es zur Trennung.

Im Eingabeformat bei der detaillierten Befehlsbeschreibung (siehe Abs. 9.3) wird ein notwendiges Leerzeichen durch das Zeichen „□“ dargestellt.

8.1.7 Datentypen

Datentypen umfassen Werte, Positionsdaten, Gelenkdaten und Zeichen. Bei Zahlen unterscheidet man zwischen reellen und ganzen Zahlen.

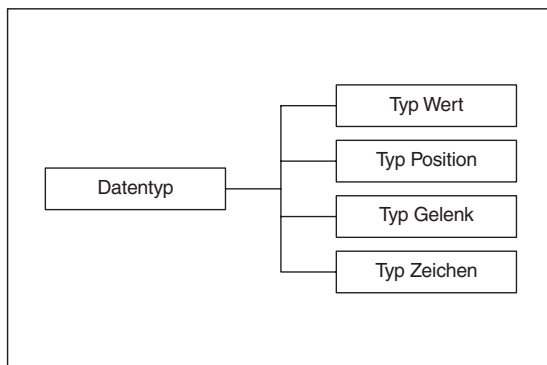


Abb. 8-1:
Datentypen

R000717C

8.1.8 Konstanten

Man unterscheidet fünf Arten von Konstanten:

- Numerische Konstanten
- Alphanumerische Konstanten
- Positionskonstanten
- Gelenkkonstanten
- Winkelkonstanten

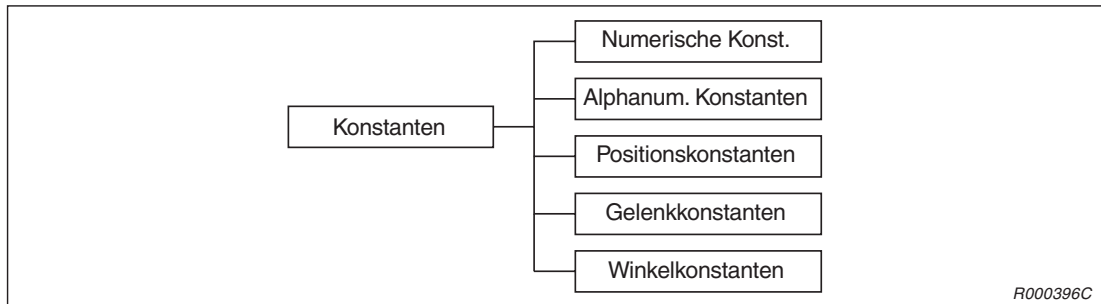


Abb. 8-2: Konstanten

Numerische Konstanten

Folgende Abbildung zeigt die Syntax numerischer Konstanten:

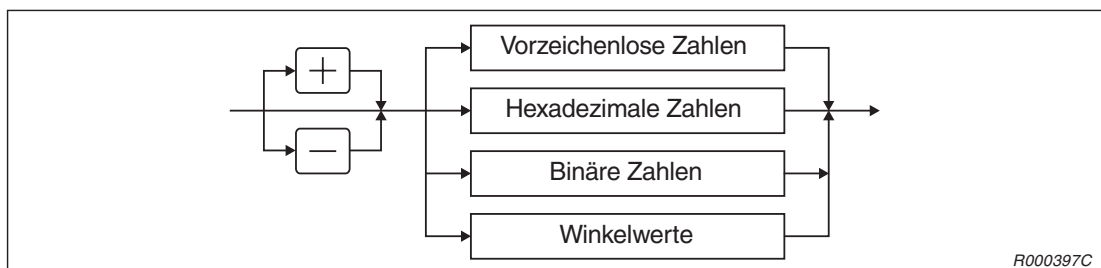


Abb. 8-3: Numerische Konstanten

Vorzeichenlose Zahlen

Folgende Abbildung zeigt die Syntax vorzeichenloser Integer-Zahlen:

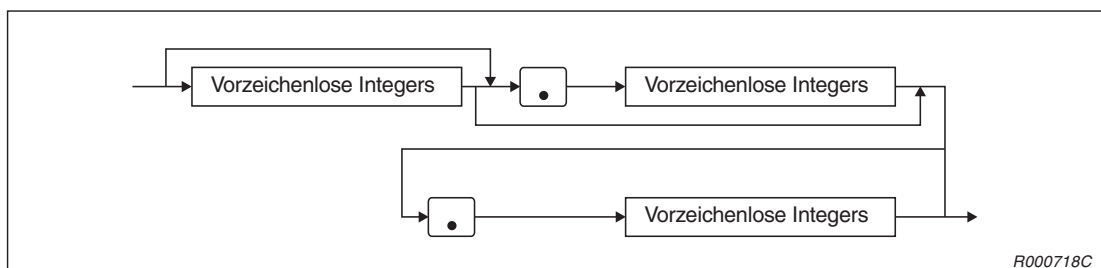


Abb. 8-4: Vorzeichenlose Integer-Zahlen

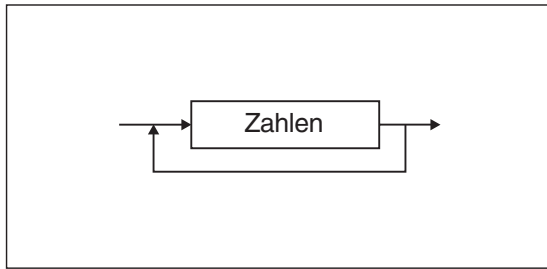


Abb. 8-5:
Vorzeichenlose Zahlen

R000399C

Der Wertebereich für Zahlen ist $-1.701411834604692E+38$ bis $1.701411834604692E+38$.

Beispiele ▾

1; 17; 1256



Hexadezimale Zahlen

Folgende Abbildung zeigt die Syntax hexadezimaler Zahlen:

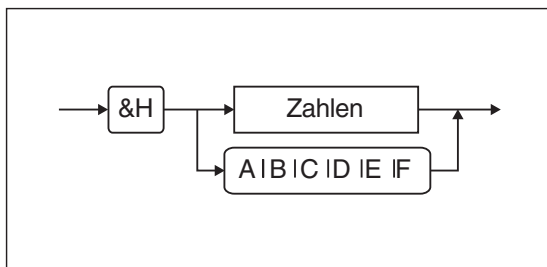


Abb. 8-6:
Hexadezimale Zahlen

R000400C

Der Wertebereich hexadezimaler Zahlen ist &H0000 bis &HFFFF.

Beispiele ▾

&H132; &HC011; &H1AC4



Binäre Zahlen

Folgende Abbildung zeigt die Syntax binärer Zahlen:

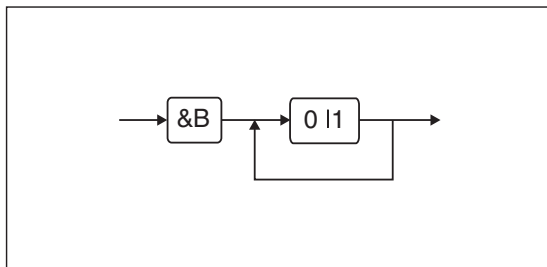


Abb. 8-7:
Binäre Zahlen

R000401C

Der Wertebereich binärer Zahlen ist &B0000000000000000 bis &B1111111111111111.

Beispiele ▾

&B010011; &B1101



Positionskonstanten

Folgende Abbildung zeigt die Syntax der Positionskonstanten:

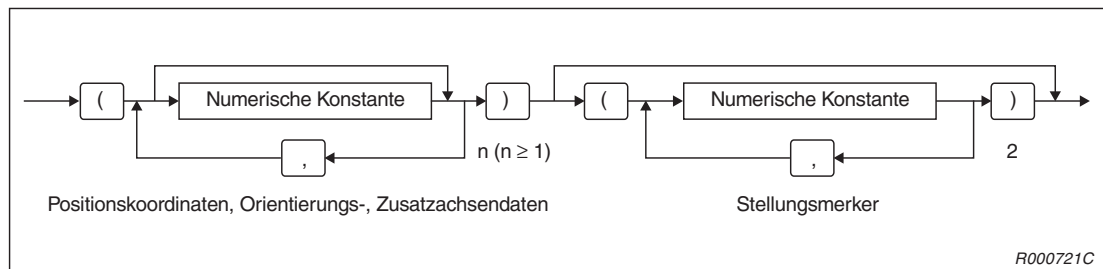


Abb. 8-8: Positionskonstanten

Die Koordinaten, Stellungsdaten und die zusätzlichen Achsensdaten haben folgende Struktur und Bedeutung:

Struktur: X, Y, Z, A, B, C, L1, L2

- X, Y, Z sind die Daten der Koordinaten. Sie geben die Position der Handspitze (TCP = Tool Center Point) des Roboters im kartesischen Koordinatensystem wieder. Sie werden in mm angegeben.
- A, B, C sind Orientierungsdaten der Roboterhand. Sie geben die Orientierung der Hand im Raum wieder. Ihre Einheit ist Grad.
- L1, L2 sind zusätzliche Achsensdaten. Sie geben die Koordinaten für die zusätzlichen Achsen 1 und 2 an. Ihre Einheit ist mm oder Grad.

Bedeutung der Stellungsdaten (fehlen die Stellungsdaten, werden die Standardeinstellungen verwendet):

Struktur: FL1, FL2

- FL1: Stellungsmerker
Standardeinstellung = 7 (Der Einstellbereich ist 0 bis 7. Jedes Bit zeigt die R/A/F-Stellung an.)
- FL2: Multirotationsinformation
Standardeinstellung = 0 (Der Einstellbereich ist 0 bis +429496725. Informationen für 8 Achsen werden mit 4 Bits für 1 Achse dargestellt (-8 bis 7).

Beispiele ▾

P1 = (X,Y,Z,A,B,C)(FL1,FL2)
P1 = (X,Y,Z,A,B,C,L1,L2)(FL1,FL2)

△

HINWEISE

Es ist nicht erforderlich, die Koordinaten und Stellungsdaten für sämtliche 8 Achsen anzugeben. Bei unvollständigen Angaben werden die folgenden Achsensdaten als undefiniert verarbeitet. Geben Sie die Daten für einen 4-achsigen Roboter (Achsenkonfiguration: X, Y, Z, C) wie folgt an: (X, Y, Z, , , C) oder (X, Y, Z, 0, 0, C).

Eine Positionskonstante darf keine Variable als Element enthalten.

Gelenkkonstanten

Folgende Abbildung zeigt die Syntax der Gelenkkonstanten:

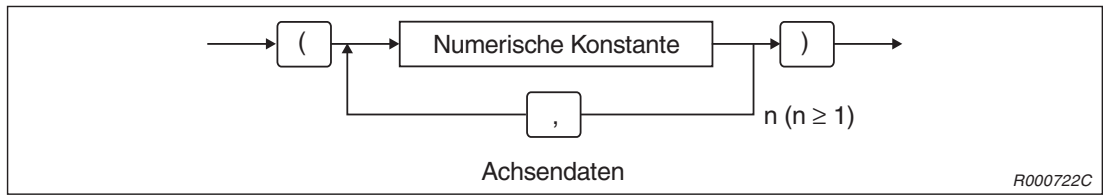


Abb. 8-9: Gelenkkonstanten

Struktur und Bedeutung der Achsendaten:

Struktur: J1, J2, J3, J4, J5, J6, J7, J8

- J1 bis J6: Achsendaten (in mm oder Grad)
- J7 und J8: Daten der Zusatzachsen (in mm oder Grad)

HINWEIS | Eine Gelenkkonstante darf keine Variable als Element enthalten.

Zeichenkettenkonstanten

Folgende Abbildung zeigt die Syntax der Zeichenkettenkonstanten:

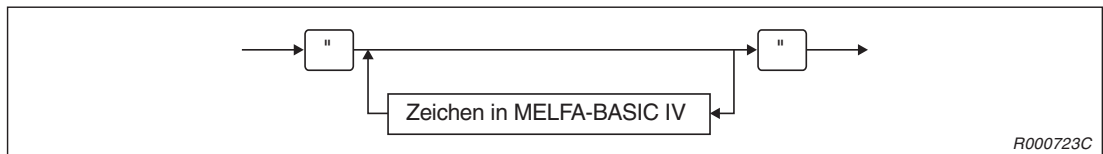


Abb. 8-10: Zeichenkettenkonstanten

Die maximale Länge einer Zeichenkette darf 127 Zeichen betragen. Bei mehr als 127 Zeichen wird ein Syntax-Fehler gemeldet.

Beispiel ▾ "Gefertigte Teile ="



Winkelbetrag

Der Winkelbetrag wird in Grad (nicht in Radiant) angegeben. Folgende Abbildung zeigt die Syntax des Winkelbetrages:

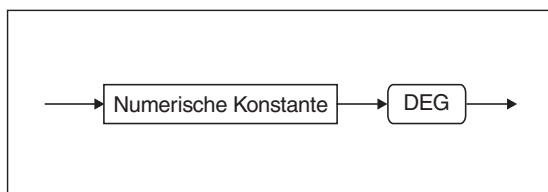


Abb. 8-11: Winkelbetrag

R000724C

Beispiel ▾ Der Sinus eines 90°-Winkels wird folgendermaßen dargestellt: SIN(90DEG).



8.1.9 Variablen

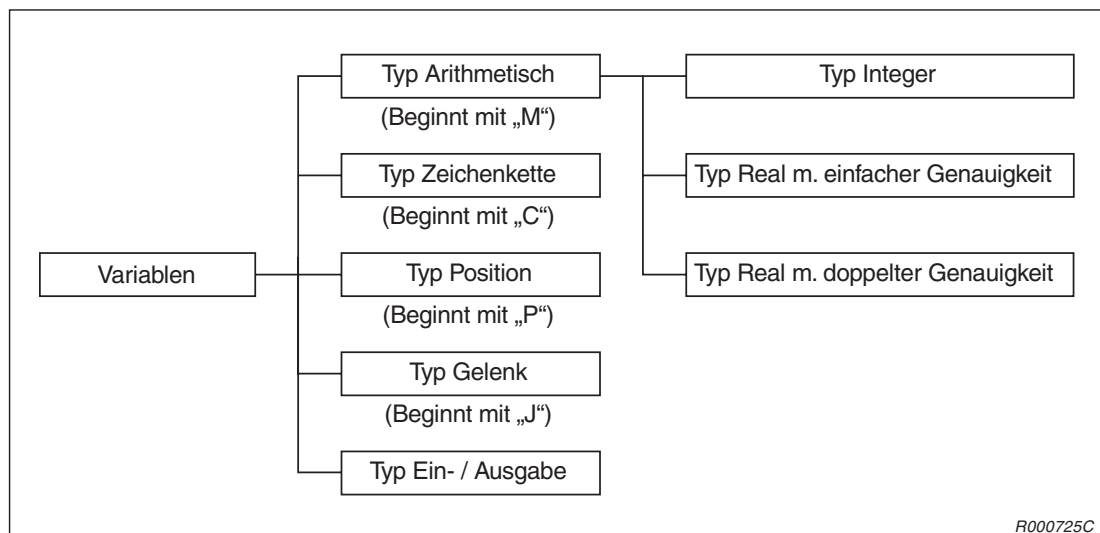
Die folgenden Variablenwerte können eingesetzt werden. Sie unterscheiden sich anhand der Werte, die in ihnen abgelegt werden.

- Arithmetische Variablen, Zeichenvariablen, Positionsvariablen, Gelenkvariablen und Ein- und Ausgabevariablen
- Arithmetische Variablen lassen sich weiterhin in ganze Zahlen (Integer), reelle Zahlen (Real) mit einfacher Genauigkeit und reelle Zahlen mit doppelter Genauigkeit unterteilen.

Variablen können nach ihrem Verwendungsbereich in lokale und globale Variablen eingeteilt werden.

- Lokale Variablen werden innerhalb eines Programms verwendet.
- Roboterstatusvariablen, externe Variablen und benutzerdefinierte externe Variablen sind globale Variablen und können auch programmübergreifend verwendet werden (benutzerdefinierte externe Variablen sind durch einen Unterstrich () als zweites Zeichen gekennzeichnet).

Folgende Abbildung zeigt die Syntax von Variablen:



R000725C

Abb. 8-12: Variablen

HINWEIS

Variablen werden beim Generieren und beim Laden oder Zurücksetzen eines Programms nicht gelöscht.

Arithmetische Variablen

Folgende Abbildung zeigt die Syntax von arithmetischen Variablen:

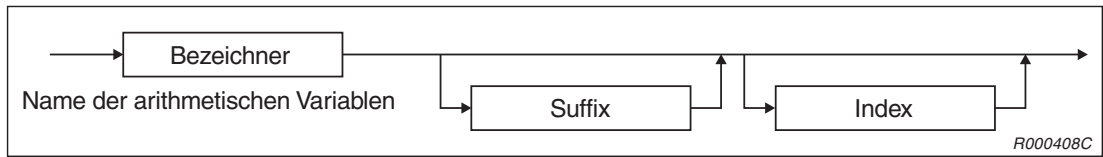


Abb. 8-13: Arithmetische Variablen

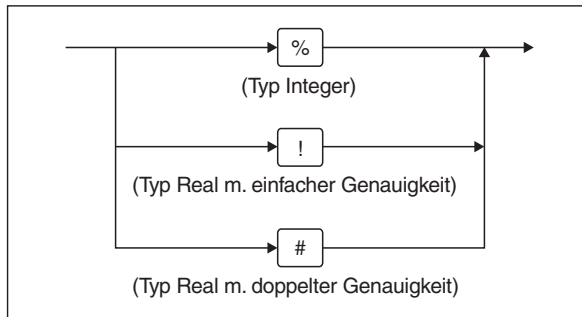


Abb. 8-14: Suffix

R000410C

HINWEISE

Fehlt das Suffix in der Variablendeklaration, so wird die Variable als reelle Zahl mit einfacher Genauigkeit interpretiert.

Werden Zahlen mit einfacher Genauigkeit und Zahlen mit doppelter Genauigkeit gemeinsam verarbeitet oder ersetzt, tritt ein Fehler auf.

Informationen über die Indizierung finden Sie in Abs. 8.1.10 „Feldvariablen“ in diesem Handbuch.

Typ	Bereich
Integer	-32 768 bis 32 767
Reelle Zahlen mit einfacher Genauigkeit	-1,70141E+38 bis 1,70141E+38
Reelle Zahlen mit doppelter Genauigkeit	-1,701411834604692E+38 bis 1,701411834604692E+38

Tab. 8-2: Wertebereich

Zeichenkettenvariablen

Folgende Abbildung zeigt die Syntax von Zeichenketten-Variablen:

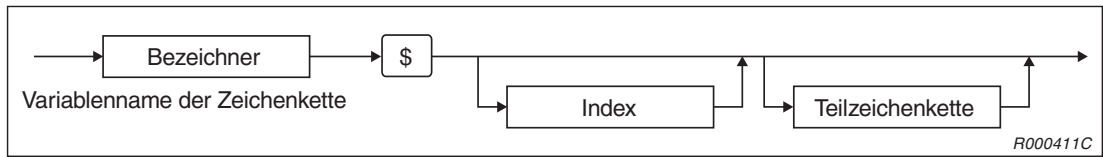


Abb. 8-15: Zeichenkettenvariablen

Beispiel ▾

C1\$

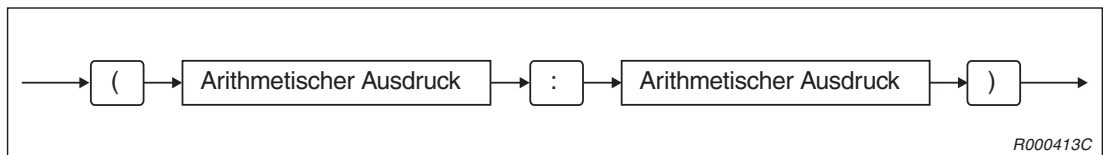


Abb. 8-16: Teilzeichenkette

HINWEIS

Informationen über die Indizierung finden Sie im Abs. 8.1.10 „Feldvariablen“ in diesem Handbuch.

Positionsvariablen

Folgende Abbildung zeigt die Syntax von Positionsvariablen:

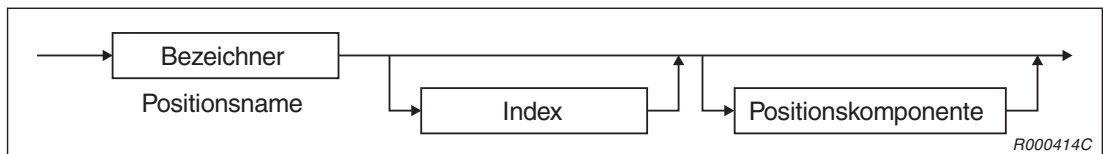


Abb. 8-17: Positionsvariablen



Abb. 8-18: Positionskomponenten

Die Übertragung der Z-Komponente der Positionsvariablen P1 in die numerische Variable M1 erfolgt mit: M1 = P1.Z. Die Einheit der Winkelkomponenten (z. B. P1.A) ist RAD.

HINWEIS

Informationen über die Indizierung finden Sie im Abs. 8.1.10 „Feldvariablen“ in diesem Handbuch.

Gelenkvariablen

Folgende Abbildung zeigt die Syntax von Gelenkvariablen:

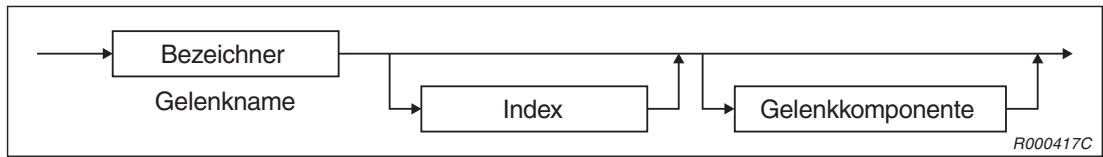


Abb. 8-19: Gelenkvariablen

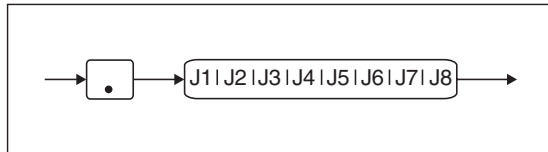


Abb. 8-20:
Gelenkkomponenten

HINWEISE

Informationen über die Indizierung finden Sie im Abs. 8.1.10 „Feldvariablen“ in diesem Handbuch.

E/A-Variablen

Folgende Abbildung zeigt die Syntax von E/A-Variablen:

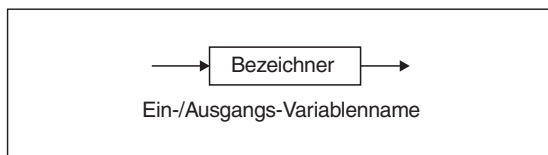


Abb. 8-21:
E/A-Variablen

8.1.10 Feldvariablen

Arithmetische Variablen, Zeichenkettenvariablen, Positionsvariablen und Gelenkvariablen können in Feldvariablen verwendet werden. Die Festlegung der Feldvariablenelemente finden Sie im Abschnitt für die Indizierung der jeweiligen Variablen. Die Syntax für die Indizierung zeigt folgende Abbildung:

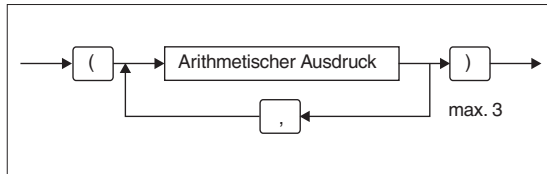


Abb. 8-22:
Indizierung

R000420C

HINWEISE

Deklarieren Sie die Feldvariablen, bevor Sie sie verwenden (siehe auch Abs. 9.3.23 „DIM-Befehl“).

Eine Feldvariable darf aus maximal 3 Dimensionen bestehen.

Eindimensionale Feldvariable: M1(10)

Zweidimensionale Feldvariable: M1(3,2)

Dreidimensionale Feldvariable: M1(3,2,5)

Wenn Sie die Indizierung nach dem Namen der Feldvariablen in Klammern setzen, können Sie mit einem Variablennamen mehrdimensionale Daten darstellen.

Es ist möglich, einen variablen arithmetischen Ausdruck als Index zu verwenden.

8.1.11 Externe Variablen

Externe Variablen sind durch einen Unterstrich () an der zweiten Stellen des Bezeichners (Variablenname) gekennzeichnet. Die Werte können programmübergreifend verwendet werden. Externe Variablen ermöglichen somit den Austausch von Daten zwischen Programmen.

Man unterscheidet vier Typen von externen Variablen:

- numerische Variablen
- Positionsvariablen
- Gelenkvariablen
- Zeichenkettenvariablen

Als externe Variablen bezeichnet man programmexterne Variablen, Roboterstatusvariablen mit festgelegten Namen sowie benutzerdefinierte externe Variablen, deren Namen durch den Benutzer definiert werden.

Programmexterne Variablen

In folgender Tabelle sind die verfügbaren programmexternen Variablen aufgeführt. Die Variablennamen sind festgelegt, die Anwendung kann vom Anwender definiert werden.

Datentyp	Variablenname	Anzahl
Position	P_00 bis P_19	20
Positions-Feldvariable (Anzahl der Elemente: 10)	P_100() bis P_104()	5
Gelenk	J_00 bis J_19	20
Gelenk-Feldvariable (Anzahl der Elemente: 10)	J_100() bis J_104()	5
Numerisch	M_00 bis M_19	20
Numerische Feldvariable (Anzahl der Elemente: 10)	M_100() bis M_104()	5
Zeichenketten	C_00 bis C_19	20
Zeichenketten-Feldvariablen (Anzahl der Elemente: 10)	C_100() bis C_104()	5

Tab. 8-3: Programmexterne Variablen

Benutzerdefinierte externe Variablen

Eine über den Befehl DEF deklarierte Variable, mit einem Unterstrich () an der zweiten Stelle im Variablennamen, wird als externe Variable verarbeitet. Dazu muss die Variable im Basisprogramm deklariert sein. Ist die Variable nicht im Basisprogramm deklariert, sondern in einem anderen Programm, so wird sie nicht als externe, sondern als lokale Variable behandelt.

Der Variablenname und die Anwendung einer benutzerdefinierten Variablen können vom Anwender definiert werden.

Beispiel ▾

```
DEF POS P_CHECK
```

Deklariert die Variable CHECK als Positionsvariable



HINWEIS

Bei Verwendung von benutzerdefinierten Variablen wird ein Basisprogramm benötigt. Das Basisprogramm wird über den Parameter PRGUSR definiert. Das Basisprogramm enthält die über die Befehle DEF INTE, DEF POS festgelegte Deklaration der benutzerdefinierten Variablen usw. Dieses Programm wird nicht ausgeführt.

Roboterstatusvariablen

Roboterstatusvariablen werden verwendet, um einen schnellen Zugriff auf den Roboterzustand zu ermöglichen oder um ihn zu ändern. Die Variablennamen und die Funktion der Roboterstatusvariablen sind vordefiniert.

Nr.	Variablenname	Feldelement (Anzahl der Elemente) ^①	Inhalt	Zugriff	Datentyp
1	P_CURR	Roboter- nummer (4)	Augenblicksposition (kartesisch)	Lesen	Position
2	J_CURR	Roboter- nummer (4)	Augenblicksposition (Gelenk)	Lesen	Gelenk
3	P_TOOL	Roboter- nummer (4)	Zuletzt festgelegte Werkzeug- konvertierungsdaten	Lesen	Position
4	P_BASE	Roboter- nummer (4)	Zuletzt festgelegte Basis- konvertierungsdaten	Lesen	Position
5	P_NTOOL	Roboter- nummer (4)	Standardwert der Werkzeug- konvertierungsdaten	Lesen	Position
6	P_NBASE	Roboter- nummer (4)	Standardwert der Basis- konvertierungsdaten	Lesen	Position
7	M_OVRD	Programm- platznummer (32)	Zuletzt festgelegter Über- steuerungswert (gültig für das gesamte Programm)	Lesen	Integer
8	M_JOVRD	Programm- platznummer (32)	Zuletzt festgelegter Über- steuerungswert (gültig nur bei Gelenk-Interpolation)	Lesen	Integer
9	M_LINE	Programm- platznummer (32)	Aktuell ausgeführte Zeilennummer	Lesen	Integer
10	M_SKIPCQ	Programm- platznummer (32)	Wird nach Ausführung eines SKIP-Befehls auf „1“, sonst auf „0“ gesetzt	Lesen	Integer
11	M_RATIO	Programm- platznummer (32)	Position bezogen auf die Ziel- position	Lesen	Integer
12	M_RDST	Programm- platznummer (32)	Restverfahrweg zur Zielposition der Komponenten X, Y und Z	Lesen	Real mit einfacher Genauigkeit
13	M_SPD	Programm- platznummer (32)	Zuletzt festgelegter Geschwin- digkeitswert (gültig für Linear- und Kreisinterpolation)	Lesen	Real mit einfacher Genauigkeit
14	M_ACL	Programm- platznummer (32)	Zuletzt festgelegte Beschleuni- gungszeit	Lesen	Real mit einfacher Genauigkeit
15	M_DACL	Programm- platznummer (32)	Zuletzt festgelegte Abbremszeit	Lesen	Real mit einfacher Genauigkeit
16	M_NOVRD	Programm- platznummer (32)	Systemstandardwert (M_OVRD Standardwert)	Lesen	Real mit einfacher Genauigkeit
17	M_NJOVRD	Programm- platznummer (32)	Systemstandardwert (M_JOVRD Standardwert)	Lesen	Real mit einfacher Genauigkeit
18	M_NSPD	Programm- platznummer (32)	Systemstandardwert (M_SPD Standardwert)	Lesen	Real mit einfacher Genauigkeit

Tab. 8-4: Roboterstatusvariablen (1)

Nr.	Variablenname	Feldelement (Anzahl der Elemente) ^①	Inhalt	Zugriff	Datentyp
19	M_NACL	Programmplatznummer (32)	Systemstandardwert (M_ACL Standardwert)	Lesen	Real mit einfacher Genauigkeit
20	M_NDAACL	Programmplatznummer (32)	Systemstandardwert (M_DACL Standardwert)	Lesen	Real mit einfacher Genauigkeit
21	M_RSPD	Programmplatznummer (32)	Aktuelle Geschwindigkeit (gültig für Linear- und Kreis-Interpolation)	Lesen	Real mit einfacher Genauigkeit
22	C_MECHA	Programmplatznummer (32)	Name des Roboters	Lesen	Zeichenkette
23	M_ACLSTS	Programmplatznummer (32)	Aktueller Status der Beschleunigung-/Abbremsung 0 = gestoppt; 1 = beschleunigt; 2 = konstante Geschwindigkeit; 3 = bremst	Lesen	Integer
24	M_PI	—	Kreiszahl (3.1415...)	Lesen	Real mit doppelter Genauigkeit
25	M_EXP	—	Basis des natürlichen Logarithmus (2.71828...)	Lesen	Real mit doppelter Genauigkeit
26	M_G	—	Erdbeschleunigung (9.80665)	Lesen	Real mit doppelter Genauigkeit
27	C_MAKER	—	Herstellerinformation (max. 64 Zeichen)	Lesen	Zeichenkette
28	C_USER	—	Benutzerinformation	Lesen	Zeichenkette
29	C_DATE	—	Aktuelles Datum Jahr/Monat/Datum	Lesen	Zeichenkette
30	C_TIME	—	Aktuelle Zeit Stunde/Minute/Sekunde	Lesen	Zeichenkette
31	M_BTIME	—	Restzeit der Batterie	Lesen	Integer
32	TIMER	Timernummer (8)	Zeitdauer ab Bezugszeit [ms]	Lesen	Real mit einfacher Genauigkeit
33	M_RUN	Programmplatznummer (32)	Betriebsanzeige	Lesen/Schreiben	Integer
34	M_WAI	Programmplatznummer (32)	Wartestatus	Lesen	Integer
35	M_ERR	Programmplatznummer (32)	Fehlermeldung	Lesen	Integer
36	M_SVO	Roboter- nummer (4)	Servospannung EIN	Lesen	Integer
37	M_CYS	Programmplatznummer (32)	Kontinuierlicher Betrieb EIN, zyklischer Betrieb EIN	Lesen	Integer
38	M_UAR	Roboter- nummer (4)	Bitdaten, Bit 0: Bereich 1 ... Bit 7: Bereich 2	Lesen	Integer
39	C_PRG	Programmplatznummer (32)	Programmname für Ausführung	Lesen	Zeichenkette
40	M_ON	—	Eine „1“ wird gesetzt	Lesen	Integer
41	M_OFF	—	Eine „0“ wird gesetzt	Lesen	Integer

Tab. 8-4: *Roboterstatusvariablen (2)*

Nr.	Variablenname	Feldelement (Anzahl der Elemente) ^①	Inhalt	Zugriff	Datentyp
42	M_IN	Eingangsnnummer (32767)	Allgemeine Bit-Schnittstelle, Bit-Eingang: 0 = AUS, 1 = EIN	Lesen	Integer
43	M_INB	Eingangsnnummer (32767)	Allgemeine Bit-Schnittstelle, Byte-Eingang	Lesen	Integer
44	M_INW	Eingangsnnummer (32767)	Allgemeine Bit-Schnittstelle, Wort-Eingang	Lesen	Integer
45	M_OUT	Ausgangsnummer (32767)	Allgemeine Bit-Schnittstelle, Bit-Ausgang: 0 = AUS, 1 = EIN	Lesen/Schreiben	Integer
46	M_OUTB	Ausgangsnummer (32767)	Allgemeine Bit-Schnittstelle, Byte-Ausgang	Lesen/Schreiben	Integer
47	M_OUTW	Ausgangsnummer (32767)	Allgemeine Bit-Schnittstelle, Wort-Ausgang	Lesen/Schreiben	Integer
48	M_DIN	Eingangsnnummer (32767)	Allgemeine Wort-Schnittstelle, Wort-Eingang	Lesen	Integer
49	M_DOUT	Ausgangsnummer (32767)	Allgemeine Wort-Schnittstelle, Wort-Eingang	Lesen/Schreiben	Integer
50	P_SAFE	Roboter- nummer (4)	Position des Rückzugpunktes	Lesen	Position
51	J_ORIGIN	Roboter- nummer (4)	Gelenkkoordinaten des Referenzpunktes	Lesen	Gelenk
52	P_ZERO	—	Nullpunkt	Lesen	Position
53	M_ERRLVL	—	Schwelle für Lesefehler: (H/L/C = 3/2/1)	Lesen	Integer
54	M_ERRNO	—	Fehlernummer lesen	Lesen	Integer

Tab. 8-4: *Roboterstatusvariablen (3)*

- ① Roboternummer: Festlegung der Roboter-
nummer im Multitask-Betrieb
 Programmplatznummer: Festlegung des Programmplatzes im Multitask-Betrieb
 Eingangsnummer: Bit-Nummer des Eingangssignals
 Ausgangsnummer: Bit-Nummer des Ausgangssignals

8.1.12 Logische Werte

Logische Werte geben die Resultate von Vergleichsoperationen oder Ein- und Ausgangszuständen wieder. Ein Ergebnis ungleich 0 entspricht dem Wert „wahr“ und ein Ergebnis gleich 0 entspricht dem Wert „unwahr“. Ergebnisse werden im Integer-Format angezeigt. Bei Substitutionen wird für den Wert „wahr“ eine 1 gesetzt. Folgende Tabelle zeigt die logischen Werte und deren Bedeutung:

Durch eine 1 dargestellte Zustände	Durch eine 0 dargestellte Zustände
Ergebnis einer Vergleichsoperation (falls wahr)	Ergebnis einer Vergleichsoperation (falls unwahr)
Ergebnis einer logischen Operation (falls wahr)	Ergebnis einer logischen Operation (falls unwahr)
Schalter EIN	Schalter AUS
Ein-/Ausgangsignal EIN	Ein-/Ausgangsignal AUS
Handgreifer offen (Stromfluss durch die Hand)	Handgreifer geschlossen (kein Stromfluss durch die Hand)
Einstellungen für Freigabe oder Gültigkeit (wie zum Beispiel bei Interrupts)	Einstellungen für Sperren oder Ungültigkeit (wie zum Beispiel bei Interrupts)

Tab. 8-5: *Logische Werte und deren Bedeutung*

8.1.13 Funktionen

Mit dem Argument einer Funktion wird eine durch die Funktion festgelegte Rechenoperation durchgeführt. Das Ergebnis kann ein numerischer Typ oder eine Zeichenkette sein.

Benutzerdefinierte Funktionen

Benutzerdefinierte Funktionen werden mit dem Befehl DEF FN erstellt.

Beispiel ▾

DEF FNMADD(MA, MB) = MA + MB



Funktionen beginnen mit den Zeichen „FN“. Das dritte Zeichen dient zur Beschreibung des Datentyps (Zeichenkette: C, Numerischer Wert: M, Position: P, Gelenk: J). Es können bis zu 8 Zeichen verwendet werden.

Fest definierte Funktionen

Folgende Tabelle zeigt die fest definierten Funktionen:

Funktionsart	Funktionsname (Format)	Bedeutung	Ergebnis
Numerische Funktionen	ABS (<Numerischer Ausdruck>)	Bildet den Betrag	Numerischer Wert
	CINT (<Numerischer Ausdruck>)	Rundet den dezimalen Wert zu einer Integer-Zahl	
	DEG (<Numerischer Ausdruck: radian>)	Wandelt die Einheit des Winkels von Radian (rad) in Grad (deg) um	
	EXP (<Numerischer Ausdruck>)	Berechnet den Wert der Exponentialfunktion	
	FIX (<Numerischer Ausdruck>)	Erzeugt einen Integer-Sektor	
	INT (<Numerischer Ausdruck>)	Erzeugt die größtmögliche Integer-Zahl, die kleiner als der Wert des numerischen Ausdrucks ist	
	LEN (<Ausdruck für eine Zeichenkette>)	Berechnet die Länge der Zeichenkette	
	LN (<Numerischer Ausdruck>)	Berechnet den natürlichen Logarithmus	
	LOG (<Numerischer Ausdruck>)	Berechnet den dekadischen Logarithmus	
	MAX (<Numerischer Ausdruck>...)	Berechnet den Maximalwert der numerischen Ausdrücke	
	MIN (<Numerischer Ausdruck>...)	Berechnet den Minimalwert der numerischen Ausdrücke	
	RAD (<Numerischer Ausdruck: deg>)	Wandelt die Einheit des Winkels von Grad (deg) in Radian (rad) um	
	SGN (<Numerischer Ausdruck>)	Prüft das Vorzeichen des numerischen Ausdrucks	
	SQR (<Numerischer Ausdruck>)	Berechnet die Quadratwurzel	
	STRPOS (<Ausdruck für eine Zeichenkette>, <Ausdruck für eine Zeichenkette>)	Gibt die Position der zweiten Zeichenkette innerhalb der ersten Zeichenkette an	
RND (<Numerischer Ausdruck>)	Ermittelt eine Zufallszahl		

Tab. 8-6: Fest definierte Funktionen (1)

Funktionsart	Funktionsname (Format)	Bedeutung	Ergebnis
Numerische Funktionen	ASC (<Typ Zeichenkette>)	Erzeugt den ASCII-Code für das erste Zeichen in der Zeichenkette	Numerischer Wert
	CVI (<Typ Zeichenkette>)	Wandelt eine 2-Byte-Zeichenkette in einen Integer-Wert um	
	CVS(<Typ Zeichenkette>)	Wandelt eine 4-Byte-Zeichenkette in einen Real-Wert mit einfacher Genauigkeit um	
	CVD(<Typ Zeichenkette>)	Wandelt eine 8-Byte-Zeichenkette in einen Real-Wert mit doppelter Genauigkeit um	
	VAL (<Typ Zeichenkette>)	Wandelt eine Zeichenkette in einen numerischen Wert um	
Trigonometrische Funktionen	ATN (<Numerischer Ausdruck>)	Berechnet den Arcus Tangens (Einheit: rad) Definitionsbereich: numerischer Wert, $-\pi/2$ bis $+\pi/2$	Numerischer Wert
	ATN2 (<Numerischer Ausdruck>, <Numerischer Ausdruck>)	Berechnet den Arcus Tangens (Einheit: rad) ($\Theta = \text{ATN2}(\Delta y, \Delta x)$) Definitionsbereich: numerischer Wert von Δy und Δx ungleich 0, $-\pi$ bis $+\pi$	
	COS (<Numerischer Ausdruck>)	Berechnet den Kosinus (Einheit: rad) Definitionsbereich: numerischer Wert, -1 bis $+1$	
	SIN (<Numerischer Ausdruck>)	Berechnet den Sinus (Einheit: rad) Definitionsbereich: numerischer Wert, -1 bis $+1$	
	TAN (<Numerischer Ausdruck>)	Berechnet den Tangens (Einheit: rad) Definitionsbereich: numerischer Wertebereich	
Zeichenkettenfunktionen	BIN\$ (<Numerischer Ausdruck>)	Wandelt den Wert des numerischen Ausdrucks in eine binäre Zeichenkette um	Zeichenkette
	CHR\$ (<Numerischer Ausdruck>)	Erzeugt ein Zeichen, das dem Wert des numerischen Ausdrucks entspricht	
	HEX\$ (<Numerischer Ausdruck>)	Wandelt den Wert des numerischen Ausdrucks in eine hexadezimale Zeichenkette um	
	LEFT\$ (<Zeichenkette>, <Numerischer Ausdruck>)	Erzeugt einen Teil der Zeichenkette Die Länge der erzeugten Zeichenkette, beginnend mit dem linken Zeichen, ist im zweiten Argument festgelegt.	
	MID\$ (<Zeichenkette>, <Numerischer Ausdruck>, <Numerischer Ausdruck>)	Erzeugt einen Teil der Zeichenkette Die Länge der erzeugten Zeichenkette ist im dritten, die Position von links im zweiten Argument festgelegt.	
	MIRROR\$ (<Typ Zeichenkette>)	Spiegelung der binären Bits der Zeichenkette	
	MKI\$ (<Numerischer Ausdruck>)	Wandelt den Wert des numerischen Ausdrucks in eine 2-Byte-Zeichenkette um	
	MKS\$ (<Numerischer Ausdruck>)	Wandelt den Wert des numerischen Ausdrucks in eine 4-Byte-Zeichenkette um	
	MKD\$ (<Numerischer Ausdruck>)	Wandelt den Wert des numerischen Ausdrucks in eine 8-Byte-Zeichenkette um	
	RIGHT\$ (<Zeichenkette>, <Numerischer Ausdruck>)	Erzeugt einen Teil der Zeichenkette Die Länge der erzeugten Zeichenkette, beginnend mit dem rechten Zeichen, ist im zweiten Argument festgelegt.	
STR\$ (<Numerischer Ausdruck>)	Wandelt den Wert des numerischen Ausdrucks in eine dezimale Zeichenkette um		

Tab. 8-6: Fest definierte Funktionen (2)

Funktionsart	Funktionsname (Format)	Bedeutung	Ergebnis
Positions-variablen	DIST (<Position>, <Position>)	Berechnet den Abstand zwischen zwei Punkten	Position
	FRAM (<Position 1>, <Position 2>, <Position 3>)	Berechnet das Koordinatensystem über 3 Punkte Position 1 entspricht dem Flächenursprung, Position 2 dem Punkt in der Fläche der X- und Position 3 dem Punkt in der Fläche der Y-Achse. Der Flächenursprungspunkt und die Stellung sind durch die X-, Y- und Z-Koordinaten der 3 Positionen beschrieben und können über den Rücksetzwert (Position) zurückgesetzt werden. Die Ausführung erfolgt ohne Berücksichtigung des Roboters über 6 Achsen und 3 Dimensionen.	
	RDFL1 (<Position>, <Numerischer Wert>)	Überträgt den Stellungsmerker der festgelegten Position als Zeichenkette Argument <numerischer Wert>: 0 = R/L, 1 = A/B, 2 = F/N	Zeichenkette
	SETFL1 (<Position>, <Zeichen>)	Änderung des Stellungsmerkers der festgelegten Position Die zu ändernden Daten werden über Zeichen definiert (R/L/A/B/F/N).	
	RDFL2 (<Position>, <Numerischer Wert>)	Überträgt die Multirotationsdaten der festgelegten Position als numerischen Wert (-2 bis 1) Das Argument <numerischer Ausdruck> überträgt die Achsennummer (1 bis 8).	Numerischer Wert
	SETFL2 (<Position>, <Numerischer Wert>, <Numerischer Wert>)	Änderung der Multirotationsdaten der festgelegten Position als numerischer Wert (-2 bis 1) Die linke Seite des Ausdrucks entspricht der Achsennummer, die geändert werden soll, die rechte Seite entspricht dem Wert.	
	ALIGN (<Position>)	Setzt den Wert der Position mit dem kleinstmöglichen senkrechten oder waagerechten Abstand zur Stellung (A, B, C) der Position 1	
	INV (<Position>)	Positionsumkehr	Position
	PTOJ (<Position>)	Konvertiert die Positions- in Gelenkdaten	Gelenk
	JTOP (<Position>)	Konvertiert die Gelenk- in Positionsdaten	Position
	ZONE (<Position 1>, <Position 2>, <Position 3>)	Prüft, ob die Position 1 im Bereich zwischen den Positionen 2 und 3 liegt (außerhalb = 0, innerhalb = 1)	Numerischer Wert

Tab. 8-6: Fest definierte Funktionen (3)

8.1.14 Konvertierte Datentypen

Numerische Variablen müssen in MELFA-BASIC IV nicht als Integer- oder reelle Zahl deklariert werden. In Abhängigkeit von der ausgeführten Operation werden die Daten automatisch konvertiert. Dabei kann das Ergebnis in Abhängigkeit von der Reihenfolge der Datentypen unterschiedlich sein. Folgende Tabelle zeigt einige Beispiele:

Linkes Argument	Operation	Rechtes Argument	Ergebnis
15 (Typ Numerisch)	AND	256 (Typ Numerisch)	15 (Typ Numerisch)
P1 (Typ Position)	*	M1 (Typ Numerisch)	P2 (Typ Position)
M1 (Typ Numerisch)	*	P1 (Typ Position)	FEHLER

Tab. 8-7: Operationsergebnisse in Abhängigkeit der Datenreihenfolge

Konvertierung der Datentypen in Abhängigkeit der Operation

Folgende Tabelle zeigt die Konvertierung von Datentypen in Abhängigkeit von der ausgeführten Operation. Bei der Angabe von logischen Operationen ist die logische Negation ausgenommen.

Typ linkes Argument	Operation	Typ rechtes Argument					
		Zeichenkette	Numerischer Wert		Position	Gelenk	
			Integer	Real			
Zeichenkette	Substitution	Zeichenkette	—	—	—	—	
	Addition	Zeichenkette	—	—	—	—	
	Vergleich	Integer	—	—	—	—	
Numerischer Wert	Integer	Addition	—	Integer	Real	—	—
		Subtraktion	—	Integer	Real	—	—
		Multiplikation	—	Integer	Real	—	—
		Division	—	Integer	Real	—	—
		Integer-Division	—	Integer	Integer	—	—
		Modulo	—	Integer	Integer	—	—
		Exponential	—	Integer	Integer	—	—
		Substitution	—	Integer	Integer	—	—
		Vergleich	—	Integer	Integer	—	—
	Logisch	—	Integer	Integer	—	—	
	Real	Addition	—	Real	Real	—	—
		Subtraktion	—	Real	Real	—	—
		Multiplikation	—	Real	Real	—	—
		Division	—	Real	Real	—	—
		Integer-Division	—	Integer	Integer	—	—
		Modulo	—	Integer	Integer	—	—
		Exponential	—	Integer	Real	—	—
		Substitution	—	Integer	Real	—	—
Vergleich		—	Integer	Integer	—	—	
Logisch	—	Integer	Integer	—	—		

Tab. 8-8: Konvertierung der Datentypen (1)

Typ linkes Argument	Operation	Typ rechtes Argument				
		Zeichenkette	Numerischer Wert		Position	Gelenk
			Integer	Real		
Position	Addition	—	—	—	Position	—
	Subtraktion	—	—	—	Position	—
	Multiplikation	—	Position	Position	Position	—
	Division	—	Position	Position	Position	—
	Integer-Division	—	—	—	—	—
	Modulo	—	—	—	—	—
	Exponential	—	—	—	—	—
	Substitution	—	—	—	Position	—
	Vergleich	—	—	—	—	—
	Logisch	—	—	—	—	—
Gelenk	Addition	—	—	—	—	Gelenk
	Subtraktion	—	—	—	—	Gelenk
	Multiplikation	—	Gelenk	Gelenk	—	—
	Division	—	Gelenk	Gelenk	—	Gelenk
	Integer-Division	—	—	—	—	—
	Modulo	—	—	—	—	—
	Exponential	—	—	—	—	—
	Substitution	—	—	—	—	Gelenk
	Vergleich	—	—	—	—	—
Logisch	—	—	—	—	—	
Nur linkes Argument	Vorzeichenumkehr	—	Integer	Integer	Position	Gelenk
	Negation NOT	—	Integer	Integer	—	—

Tab. 8-8: Konvertierung der Datentypen (2)


HINWEISE

| Eine Division durch „0“ ist nicht möglich.

| Bei der Ausführung exponentieller, modulo und logischer Operationen werden reelle Zahlen vor der Verarbeitung in ganzzahlige Werte umgewandelt und abgerundet.

8.1.15 Rangfolge von Operationen

Werden in einem Ausdruck mehrere Operationen ausgeführt, gilt die in folgender Tabelle dargestellte Rangfolge:

Operation (Operator)	Typ der Operation	Priorität
Operation in Klammern ()	—	Hoch  Niedrig
Funktion	Funktion	
Exponent (^)	Arithmetische Operation	
Operation mit einem Argument (+, -)	Arithmetische Operation	
* /	Arithmetische Operation	
\	Arithmetische Operation	
MOD	Arithmetische Operation	
+ -	Arithmetische Operation	
<< >>	Logische Operation	
Vergleichsoperation (=, <>, ><, <, <=, =<, >, >=, ==)	Vergleichsoperation	
NOT	Logische Operation	
AND	Logische Operation	
OR	Logische Operation	
XOR	Logische Operation	

Tab. 8-9: Rangfolge von Operationen

8.1.16 Programmebenen

Beim Entwurf eines Programms muss die Anzahl der Ebenen und die Struktur festgelegt werden. Werden die in folgender Tabelle aufgeführten Befehle verwendet, erweitert sich die Programmstruktur um eine Ebene. Für jeden Befehl gibt es eine maximale Anzahl der Ebenen. Wird diese Anzahl überschritten erfolgt eine Fehlermeldung.

Anzahl der Ebenen	Verfügbare Befehle
16 Ebenen	Wiederholschleifen (FOR ~ NEXT, WHILE ~ WEND)
8 Ebenen	Funktionsaufruf (CALLP)
8 Ebenen	Unterprogrammaufruf (GOSUB) ^①

Tab. 8-10: Programmebenen

^① Bei jedem Rücksprung aus einem Unterprogramm wird die Programmstruktur eine Ebene flacher.

8.1.17 Reservierte Wörter

Reservierte Wörter haben im System eine bestimmte, festliegende Bedeutung. Sie dürfen zum Beispiel nicht als Programmname etc. vergeben werden.

9 MELFA-BASIC-IV-Befehle

9.1 Allgemeine Hinweise

In den nachfolgenden Abschnitten finden Sie eine Auflistung aller MELFA-BASIC-IV-Befehle und deren Anwendungsmöglichkeiten.

9.1.1 Beschreibung des verwendeten Formats

Funktion

Hier finden Sie eine Funktionsbeschreibung des Befehls.

Eingabeformat

Hier finden Sie das genaue Format zur Eingabe des Befehls. Befehlsparameter werden in spitzen Klammern „<>“ angegeben. Die eckige Klammer „[]“ kennzeichnet die wahlfreien Befehlsparameter. Die notwendige Eingabe eines Leerzeichens wird durch „□“ dargestellt.

Programmbeispiel

Hier finden Sie die Verwendung des Befehls in einem Beispielprogramm.

Erläuterung

Hier finden Sie eine detaillierte Beschreibung, Besonderheiten usw. des Befehls.

9.2 Übersicht der MELFA-BASIC-IV-Befehle

Befehl		Funktion	Abschnitt	Seite
ACCEL	(Accelerate)	Beschleunigung einstellen	9.3.1	9-4
ACT	(Act)	Interrupt freigeben/sperrern	9.3.2	9-5
BASE	(Base)	Basis	9.3.3	9-7
CALLP	(Call P)	Programm aufrufen	9.3.4	9-8
CLOSE	(Close)	Datei schließen	9.3.5	9-9
CLR	(Clear)	Löschen	9.3.6	9-10
CMP POS	(Compliance Posture)	Achsenweichheit aktivieren	9.3.7	9-11
CMP TOOL	(Compliance Tool)	Achsenweichheit im Werkzeugkoordinatensystem aktivieren	9.3.8	9-12
CMP OFF	(Compliance OFF)	Achsenweichheit deaktivieren	9.3.9	9-13
CMPG	(Compliance Gain)	Achsenweichheit einstellen	9.3.10	9-14
CNT	(Continuous)	Roboterbewegung steuern	9.3.11	9-15
COM OFF	(Communication OFF)	Kommunikations-Interrupt sperren	9.3.12	9-17
COM ON	(Communication ON)	Kommunikations-Interrupt freigeben	9.3.13	9-18
COM STOP	(Communication Stop)	Kommunikations-Interrupt stoppen	9.3.14	9-19
DEF ACT	(Define act)	Interrupt-Prozess definieren	9.3.15	9-20
DEF FN	(Define function)	Funktion definieren	9.3.16	9-22
DEF PLT	(Define pallet)	Palette definieren	9.3.17	9-23
DEF INTE/ FLOAT/ DOUBLE	(Define Integer/ Float/Double)	Arithmetische Variable deklarieren	9.3.18	9-25
DEF IO	(Define IO)	Ein-/Ausgangsvariable definieren	9.3.19	9-26
DEF JNT	(Define Joint)	Gelenkvariable definieren	9.3.20	9-28
DEF POS	(Define Position)	Positionsvariable definieren	9.3.21	9-29
DEF CHAR	(Define Character)	Zeichenkettenvariable definieren	9.3.22	9-30
DIM	(Dim)	Dimension definieren	9.3.23	9-31
DLY	(Delay)	Verzögerung einstellen	9.3.24	9-32
ERROR	(Error)	Fehler generieren	9.3.25	9-33
END	(End)	Programmende	9.3.26	9-34
FINE	(Fine)	Feinpositionierung	9.3.27	9-35
FOR-NEXT	(For-next)	Programmschleife	9.3.28	9-36
FPRM	(FPRM)	Parameter definieren	9.3.29	9-38
GETM	(Get Mechanism)	Roboter definieren	9.3.30	9-39
GOSUB	(Go Subroutine)	Sprung zu einem Unterprogramm	9.3.31	9-40
GOTO	(Go To)	Sprung zu einer Programmzeile oder Marke	9.3.32	9-41
HLT	(Halt)	Programmablauf stoppen	9.3.33	9-42
HOPEN/ HCLOSE	(Hand open/Hand close)	Handgreiferzustand festlegen	9.3.34	9-42
IF THEN ELSE	(If Then Else)	WENN ... DANN ... SONST-Schleife	9.3.35	9-44
INPUT #	(Input)	Eingabe	9.3.36	9-45
JOVRD	(J override)	Übersteuerung Gelenk-Interpolation	9.3.37	9-46
JRC	(Joint Roll Change)	Gelenkposition verändern	9.3.38	9-47

Tab. 9-1: Übersicht der Befehle (1)

Befehl		Funktion	Abschnitt	Seite
Label	(Label)	Sprungmarke	9.3.39	9-48
LOADSET	(Load set)	Hand- und Werkstückbedingung einstellen	9.3.40	9-50
MOV	(Move)	Bewegung mit Gelenk-Interpolation	9.3.41	9-52
MVC	(Move C)	Kreis-Interpolation	9.3.42	9-53
MVR	(Move R)	Kreis-Interpolation	9.3.43	9-54
MVR2	(Move R2)	Kreis-Interpolation	9.3.44	9-56
MVR3	(Move R3)	Kreis-Interpolation	9.3.45	9-58
MVS	(Move S)	Bewegung mit Linear-Interpolation	9.3.46	9-60
OADL	(Optimum Acceleration/Deceleration)	Optimale Beschleunigung/Abbremsung	9.3.47	9-62
ON COM GOSUB	(ON Communication Go Subroutine)	Sprung zu einem Unterprogramm	9.3.48	9-63
ON-GOSUB	(ON GOSUB)	Sprung zu einem Unterprogramm	9.3.49	9-64
ON GOTO	(On go to)	Programmverzweigung	9.3.50	9-65
OPEN	(Open)	Datei öffnen	9.3.51	9-66
OVRD	(Override)	Übersteuerung	9.3.52	9-68
PLT	(Pallet)	Koordinaten für Palette berechnen	9.3.53	9-69
PRINT #	(Print)	Daten übertragen	9.3.54	9-70
RELM	(Release mechanism)	Roboterzuordnung aufheben	9.3.55	9-72
REM	(Remarks)	Kommentar	9.3.56	9-73
RETURN	(Return)	Rücksprung zum Hauptprogramm	9.3.57	9-73
SELECT CASE	(Select case)	Prozess ausführen	9.3.58	9-75
SERVO	(Servo)	Servo ein-/ausschalten	9.3.59	9-77
SKIP	(Skip)	Sprung in die nächste Zeile	9.3.60	9-78
SPD	(Speed)	Geschwindigkeit festlegen	9.3.61	9-79
TOOL	(Tool)	Werkzeug-Konvertierungsdaten	9.3.62	9-80
TORQ	(Torque)	Prozess ausführen	9.3.63	9-81
WHILE~WEND	While End	Programmschleife	9.3.65	9-83
WTH	(With)	Anweisung hinzufügen	9.3.66	9-84
WTHIF	(With If)	Anweisung hinzufügen, wenn ...	9.3.67	9-85
XLOAD	(X Load)	Programm laden	9.3.68	9-86
XRUN	(X Run)	Programm starten	9.3.69	9-87
XSTP	(X Stop)	Programm stoppen	9.3.70	9-88
XRST	(X Reset)	Programm zurücksetzen	9.3.71	9-89

Tab. 9-1: Übersicht der Befehle (2)

9.3 Detaillierte Befehlsbeschreibung

In diesem Abschnitt finden Sie eine detaillierte Beschreibung sowie Programmbeispiele zur Anwendung der Befehle.

9.3.1 ACCEL (Accelerate)

Funktion: Beschleunigung und Abbremsung einstellen

Legt den Wert für die Beschleunigung und Abbremsung in Prozent fest.

Eingabeformat

```
ACCEL □ [<Beschleunigung>] [, <Abbremsung>]
```

<Beschleunigung/Abbremsung>

Legt den Prozentwert der Beschleunigung/Abbremsung vom Stillstand bis zur maximalen Geschwindigkeit fest

Die Werte können als Konstante oder Variable angegeben werden. Erfolgt keine Angabe, wer-

den

die Werte auf 100, 100 [%] gesetzt.

Programmbeispiel

10	ACCEL 50,100	Beschleunigung/Abbremsung für große Last einstellen (bei einem Grundwert der Beschleunigungs-/Abbremszeit von 0,2 s ist eine Beschleunigungszeit von 0,4 s und eine Abbremszeit von 0,2 s wirksam)
20	MOV P1	Position P1 mittels Gelenk-Interpolation anfahren
30	ACCEL 100,100	Beschleunigung/Abbremsung für Standardlast einstellen
40	MOV P2	Position P2 mittels Gelenk-Interpolation anfahren

Erläuterung

- Die maximale Beschleunigung/Abbremsung ist vom verwendeten Robotertyp abhängig. Stellen Sie die Beschleunigung/Abbremsung als Prozentwert des Maximalwertes ein. Als Standardwerte sind die Werte 100, 100 eingestellt.

- Die Beschleunigungs- bzw. Abbremszeit berechnet sich aus:

$$\text{Zeit} = \frac{100\%}{\text{Beschleunigung}[\%]} \times 0,2\text{s}$$

- Die über diesen Befehl eingestellte Beschleunigung/Abbremsung wird beim Zurücksetzen des Programms und bei Ausführung der END-Anweisung auf die Standardwerte zurückgesetzt.

- Die Verfahrkurve für eine kontinuierliche, gleichmäßige Bewegung (CNT freigegeben) kann von der Verfahrkurve mit Beschleunigung abweichen. Die Größe der Abweichung ist abhängig vom Wert der eingestellten Beschleunigungszeit. Für eine gleichmäßige Bewegung mit einer konstanten Geschwindigkeit sollten die Beschleunigungs- und die Abbremszeit gleich sein. In der Grundeinstellung ist die CNT-Einstellung gesperrt.

9.3.2 ACT (Act)

Funktion: Interrupt freigeben/sperrern

Freigeben oder sperren des Interrupts.

Eingabeformat

```
ACT □ <Priorität>=<freigeben/sperrern>
```

<Priorität>	Gibt den Interrupt frei oder sperrt ihn $1 \leq \text{Priorität} \leq 8$ Legt die mit der Anweisung DEF ACT definierte Priorität des Interrupts fest Hinter dem ACT-Befehl muss ein Leerzeichen stehen. Die Schreibweise ACT1 wird als Anweisung zur Deklaration einer Variablen gewertet.
<freigeben/sperrern>	freigeben = 1, sperren = 0

Programmbeispiel

10 DEF ACT 1,M_IN(1) = 1 GOSUB *INTR	Weist den Eingang 1 dem Interrupt 1 zu
20 MOV P1	Position P1 mittels Gelenk-Interpolation anfahren
30 ACT 1 = 1	Interrupt 1 freigeben
40 MOV P2	Position P2 mittels Gelenk-Interpolation anfahren
50 ACT 1 = 0	Interrupt 1 sperren
60 GOTO 20	Sprung zur Programmzeile 20
100 *INTR	Ändert sich das Eingangssignal 1 auf EIN (1) während der Roboter sich von P1 nach P2 bewegt, wird der Roboter gestoppt.
110 HLT	
120 RETURN 0	Rücksprung aus der Interruptroutine

Erläuterung

- Beim Programmstart ist der Interrupt mit der Priorität 0 freigegeben. Wenn der Interrupt mit der Priorität 0 gesperrt ist, werden die Interrupts der Prioritäten 1 bis 8 nicht freigegeben, auch wenn sie auf freigegeben gesetzt sind.
- Die Interrupts der Prioritäten 1 bis 8 sind beim Programmstart gesperrt.
- Ein Interrupt kann nur ausgeführt werden, wenn folgende Bedingungen erfüllt sind: Der Interrupt der Priorität 0 ist freigegeben. Der Status der DEF ACT-Anweisung ist definiert worden. Der Interrupt, der in der DEF ACT-Anweisung festgelegt wurde, ist durch die ACT-Anweisung freigegeben.
- Ein Rücksprung aus einer Interruptroutine kann entweder durch RETURN 0 oder RETURN 1 erfolgen.
- Auch wenn der Roboter sich in einer Interpolation befindet, wird ein mit DEF ACT definierter Interrupt ausgeführt.
- Während eines Interruptprozesses wird der entsprechende Interrupt auf gesperrt gesetzt.
- Ein Kommunikations-Interrupt hat eine höhere Priorität als ein mit DEF ACT definierter Interrupt.
- Die Reihenfolge der Prioritäten ist: COM > ACT > WTHIF (WTH) > Impulsausgang.

9.3.3 BASE (Base)

Funktion: Basis-Konvertierungsdaten

Legt die Basis-Transformationsdaten fest.

Eingabeformat

```
BASE □ <Basis-Transformationsdaten>
```

<Basis-Transformationsdaten>

Legt die Basis-Transformationsdaten in Form von Positionskoordinaten fest

Programmbeispiel

10	BASE (50,100,0,0,0,0,0,0)	Eingabe der Basis-Transformationsdaten als Konstante
20	MVS P1	Position P1 mittels Linear-Interpolation anfahren
30	BASE P2	Eingabe der Basis-Transformationsdaten als Variable
40	MVS P1	Position P1 mittels Linear-Interpolation anfahren
50	BASE P_BASE	Zurücksetzen der Basis-Transformationsdaten auf den Standardwert

Erläuterung

- Die X-, Y- und Z-Koordinaten geben die parallele Verschiebung des Basiskoordinatensystems in Bezug auf das Weltkoordinatensystem an. Die Basis-Transformationsdaten können ausschließlich mit dem BASE-Befehl geändert werden. Die Komponenten A, B und C geben dabei die Drehwinkel des Basiskoordinatensystems in Bezug auf das Weltkoordinatensystem an.
X = Paralleler Abstand zur X-Achse
Y = Paralleler Abstand zur Y-Achse
Z = Paralleler Abstand zur Z-Achse
A = Drehwinkel um die X-Achse
B = Drehwinkel um die Y-Achse
C = Drehwinkel um die Z-Achse
L1 = Weg der Zusatzachse 1
L2 = Weg der Zusatzachse 2
- Aus Sicht des Koordinatenursprungs werden Winkel in Uhrzeigerrichtung positiv gewertet.
- Die Werte der Stellungsmerker sind ohne Bedeutung.
- Die Änderungen des mit dem BASE-Befehl geänderten Basiskoordinatensystems bleiben auch nach Ausschalten der Spannungsversorgung erhalten.
- Der Standardwert ist P_NBASE = (0,0,0,0,0,0)(0,0). Er wird ohne Berücksichtigung des verwendeten Robotertyps für 6 Achsen und dreidimensional berechnet.

9.3.4 CALLP (Call P)

Funktion: Programm aufrufen

Führt das aufgerufene Programm aus (siehe auch GOSUB-Befehl für Unterprogrammaufrufe).

Eingabeformat

```
CALLP □ "<Programmname>" [, <Parameter> [, <Parameter>] ... ]
```

<Programmname>	Legt den Programmnamen als Zeichenkettenkonstante oder Zeichenkettenvariable fest
<Parameter>	Legt die Variablen fest, die beim Aufruf des Programmes übergeben werden Es können maximal 16 Variablen übergeben werden.

Programmbeispiel

```
10 CALLP "P10",M1,P1,P2      Aufruf des Programms P10 und Übergabe der
                             Variablen M1, P1, P2
```

```
Programm "P10"
10 FPRM M1,P1,P2
```

Erläuterung

- Weicht eine Variable in der CALLP-Anweisung von der im Programm definierten Variablen (FPRM) ab, erfolgt eine Fehlermeldung.
- Weicht die Anzahl der Variablen in der CALLP-Anweisung von der Anzahl der im Programm definierten Variablen ab, erfolgt eine Fehlermeldung.
- Wird das Programm zurückgesetzt, geht die Steuerung auf den Anfang des Hauptprogramms zurück.
- Das aufgerufene Programm hat keinen Einfluss auf die Anweisungen DEF ACT, DEF FN, DEF PLT, DIM im aufrufenden Programm. Sobald das aufgerufene Programm zurückspringt, werden sie wieder gültig.
- Die Geschwindigkeits- und Werkzeugdaten (TCP) bleiben gültig.
- Es können bis zu 7 Programme von einem Programm aus aufgerufen werden. Dieser Wert beinhaltet auch die vom aufgerufenen Programm ausgeführten Unterprogramme.
- Eine geöffnete und ausgeführte Datei kann von keiner anderen Anwendung aufgerufen und verwendet werden. Es kann wiederholt von derselben Anwendung aufgerufen werden.
- Ein Programm kann sich nicht selber aufrufen.
- Das aufrufende Programm kann nicht aufgerufen werden.
- Die Parameter, die an ein Programm übergeben werden sollen, werden dort mit dem FRPM-Befehl festgelegt.
- Die Ergebnisse, die im aufgerufenen Programm berechnet wurden, können nicht über die Parameter an das aufrufende Programm übergeben werden. Verwenden Sie externe Variablen, um die Werte zu übergeben.

9.3.5 CLOSE (Close)

Funktion: Datei schließen

Schließt die festgelegte Datei.

Eingabeformat

```
CLOSE □ [[#]<Dateinummer>[, [[#]<Dateinummer> ...]
```

<Dateinummer> Legt die Dateinummer der zu schließenden Datei fest

Programmbeispiel

10 OPEN "COM1:" AS#1	„COM1:“ wird als Datei Nummer 1 geöffnet
20 PRINT #1,M1	Überträgt den Inhalt von M1 in Datei Nummer 1
100 INPUT #1,M2	Liest die Daten von Datei Nummer 1 in M2 ein
110 CLOSE #1	Datei Nummer 1 schließen
200 CLOSE	Alle geöffneten Dateien schließen

Erläuterung

- Eine geöffnete Ein-/Ausgabedatei lagert die Daten in den Zwischenspeicher aus, sobald die CLOSE-Anweisung ausgeführt wird. Der unterbrochene Ausgabeprozess kann somit korrekt zum Abschluss gebracht werden.
- Durch die Ausführung der END-Anweisung wird eine Datei ebenfalls geschlossen.

9.3.6 CLR (Clear)

Funktion: Löschfunktion

Zurücksetzen der allgemeinen Ausgänge, der lokalen Variablen und globalen Variablen.

Eingabeformat

```
CLR □ <Ausführung>
```

- | | |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Ausführung> | <ul style="list-style-type: none"> 0: Die allgemeinen Ausgangsbits, die lokalen und globalen Variablen werden zurückgesetzt. 1: Die allgemeinen Ausgänge werden auf ein voreingestelltes Bit-Muster zurückgesetzt. Die Bit-Muster werden über die Parameter ORST0 bis ORST224 vorgegeben (0: AUS, 1: EIN, *: HALTEN). 2: Alle lokalen numerischen Variablen und alle im Programm verwendeten numerischen Feldvariablen werden auf „0“ gesetzt. 3: Alle globalen numerischen Variablen und numerischen Feldvariablen, die programmübergreifend verwendet werden, werden auf „0“ gesetzt.
Die Ausführung kann als Konstante oder Variable vorgegeben werden. |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

9.3.7 CMP POS (Compliance Posture)

Funktion: Achsenweichheit aktivieren

Legt fest, welche Achse im kartesischen Koordinatensystem weich geschaltet werden soll.

Eingabeformat

```
CMP  POS, <Achse>
```

<Achse> Legt über ein Bit-Muster die Achse fest, für die die Weichheit eingestellt werden soll
00000000: X, Y, Z, ... C, L1, L2

Programmbeispiel

10	MOV P1	Position P1 mittels Gelenk-Interpolation anfahren
20	CMPG 0.5,0.5,1.0,0.5,0.5	Einstellung der Weichheit
30	CMP POS, &B00011011	Achsenweichheit aktivieren
40	MVS P2	Position P2 mittels Linear-Interpolation anfahren
50	M_OUT(10) = 1	Ausgang 10 auf „1“ setzen
60	MVS P1	Position P1 mittels Linear-Interpolation anfahren
70	CMP OFF	Achsenweichheit deaktivieren

Erläuterung

- Die Weichheit einer Roboterachse kann im kartesischen Koordinatensystem festgelegt werden.
- Wird beim senkrechten Einsetzen eines Bolzens in eine Bohrung die Weichheit für die Achsen X, Y, A und B aktiviert, wird der Bolzen sanft in die Bohrung geführt.
- Der Wert der Weichheit wird über den Befehl CMPG eingestellt.
- Die Einstellung der Weichheit bleibt auch bei einer Programmunterbrechung bis zur Ausführung des Befehls CMP OFF oder bis zum Aus- und Wiedereinschalten der Versorgungsspannung aktiviert.
- Ist die Weichheit aktiviert, kann der Roboter keine Position erreichen, die außerhalb des Verfahrwegbereiches der Gelenke liegt.
- Ist die Abweichung der aktuellen Position von der Zielposition größer als der in Parameter NMPCDIST eingestellte Wert, stoppt der Roboter. Der Roboter stoppt nicht während der Programmausführung.
- Wird der Befehl CMP POS ausgeführt, während der Befehl CMP TOOL aktiv ist, erfolgt eine Fehlermeldung. Heben Sie die Einstellung der Weichheit über den Befehl CMP OFF auf, bevor Sie den Befehl CMP POS ausführen.
- Schaltet sich bei aktivierter Weichheitseinstellung die Servospannung ein, kann sich die Position des Roboters ändern.
- Bei aktivierter Weichheitseinstellung ist ein JOG-Betrieb möglich.

HINWEIS

Die Einstellung ist nur bei bestimmten Robotermodellen (detaillierte Hinweise finden Sie im Technischen Handbuch des jeweiligen Roboters) möglich.

9.3.8 CMP TOOL (Compliance Tool)

Funktion: Achsenweichheit im Werkzeugkoordinatensystem aktivieren

Legt fest, welche Achse im Werkzeugkoordinatensystem weich geschaltet werden soll.

Eingabeformat

```
CMP  TOOL, <Achse>
```

<Achse> Legt über ein Bit-Muster die Achse fest, für die die Weichheit eingestellt werden soll
 00000000: X, Y, Z, ... C, L1, L2

Programmbeispiel

10	MOV P1	Position P1 mittels Gelenk-Interpolation anfahren
20	CMPG 0.5,0.5,1.0,0.5,0.5	Einstellung der Weichheit
30	CMP TOOL, &B00011011	Achsenweichheit aktivieren
40	MVS P2	Position P2 mittels Linear-Interpolation anfahren
50	M_OUT(10) = 1	Ausgang 10 auf „1“ setzen
60	MVS P1	Position P1 mittels Linear-Interpolation anfahren
70	CMP OFF	Achsenweichheit deaktivieren

Erläuterung

- Die Weichheit einer Roboterachse kann im Werkzeugkoordinatensystem festgelegt werden.
- Wird beim senkrechten Einsetzen eines Bolzens in eine Bohrung die Weichheit für die Achsen X, Y, A und B aktiviert, wird der Bolzen sanft in die Bohrung geführt.
- Der Wert der Weichheit wird über den Befehl CMPG eingestellt.
- Die Einstellung der Weichheit bleibt auch bei einer Programmunterbrechung bis zur Ausführung des Befehls CMP OFF oder bis zum Aus- und Wiedereinschalten der Versorgungsspannung aktiviert.
- Ist die Weichheit aktiviert, kann der Roboter keine Position erreichen, die außerhalb des Verfahrwegbereiches der Gelenke liegt.
- Ist die Abweichung der aktuellen Position von der Zielposition größer als der in Parameter NMPCDIST eingestellte Wert, stoppt der Roboter. Der Roboter stoppt nicht während der Programmausführung.
- Wird der Befehl CMP POS ausgeführt, während der Befehl CMP TOOL aktiv ist, erfolgt eine Fehlermeldung. Heben Sie die Einstellung der Weichheit über den Befehl CMP OFF auf, bevor Sie den Befehl CMP TOOL ausführen.
- Schaltet sich bei aktivierter Weichheitseinstellung die Servospannung ein, kann sich die Position des Roboters ändern.
- Bei aktivierter Weichheitseinstellung ist ein JOG-Betrieb möglich.

HINWEIS | Die Einstellung ist nur bei bestimmten Robotermodellen (detaillierte Hinweise finden Sie im Technischen Handbuch des jeweiligen Roboters) möglich.

9.3.9 CMP OFF (Compliance OFF)

Funktion: Achsenweichheit deaktivieren

Deaktiviert die eingestellte Weichheit.

Eingabeformat

CMP OFF

Erläuterung

- Die eingestellte Weichheit wird deaktiviert.

HINWEIS

Die Einstellung ist nur bei bestimmten Robotermodellen (detaillierte Hinweise finden Sie im Technischen Handbuch des jeweiligen Roboters) möglich.

9.3.10 CMPG (Compliance Gain)

Funktion: Achsenweichheit einstellen

Legt den Wert der Weichheit einer Achse fest.

Eingabeformat

```
CMPG □    [<Einstellung X-Achse>], [<Einstellung Y-Achse>]
          [<Einstellung Z-Achse>], [<Einstellung A-Achse>]
          [<Einstellung B-Achse>], [<Einstellung C-Achse>]
          [<Einstellung L1-Achse>], [<Einstellung L2-Achse>]
```

<Einstellung X- bis L2-Achse>

Der Grad der Weichheit für jede Achse kann eingestellt werden.
 Die Einstellung „1“ bedeutet Normalbetrieb, die Einstellung „0.0“ entspricht der größten Weichheit.
 Bei fehlender Angabe wird der aktuelle Wert verwendet.

Erläuterung

- Der Grad der Weichheit kann für jede Achse festgelegt werden.
- Die Aktivierung der Weichheit erfolgt über die Befehle CMP POS oder CMP TOOL.
- Über den Befehl CMPG wird eine Kraft ähnlich einer Federkraft eingestellt, deren Größe von der Abweichung zwischen Zielposition und der aktuellen Position abhängig ist. Die Federkonstante wird über den CMPG-Befehl eingestellt.
- Die Abweichung zwischen der Zielposition und der aktuellen Position kann aus der Variablen M_CMPDST ausgelesen werden. Über diese Variable kann das erfolgreiche Einsetzen eines Bolzens abgefragt werden.
- Bei kleinen Einstellwerten kann die Position bei Aktivierung der Weichheit durch den Befehl CMP POS oder den Befehl CMP TOOL aufgrund des Eigengewichts des Roboterarms nach unten sinken. Stellen Sie die Achsenweichheit schrittweise ein.
- Bei aktivierter Achsenweichheit kann der Grad der Weichheit auch während des Betriebs verändert werden.

9.3.11 CNT (Control)

Funktion: Roboterbewegung steuern

Legt die Steuerung für eine kontinuierliche und gleichmäßige Bewegung fest.

Eingabeformat

```
CNT □<freigeben/sperrern>[,<numerischer Wert 1>]
      [,<numerischer Wert 2>]
```

<freigeben/sperrern>	Legt den Anfang und das Ende einer kontinuierlichen und gleichmäßigen oder einer beschleunigten und abgebremsten Roboterbewegung fest freigeben = 1, gesperrt = 0
<numerischer Wert 1>	Legt den Anfangspunktabstand der kontinuierlichen Bewegung in mm fest
<numerischer Wert 2>	Legt den Endpunktabstand der kontinuierlichen Bewegung in mm fest

Programmbeispiel

10 CNT 0	Sperren der CNT-Einstellung
20 MVS P1	Position P1 mittels Linear-Interpolation und Beschleunigung/Verzögerung anfahren
30 CNT 1	Freigeben der CNT-Einstellung
40 MVS P2	Position P2 mittels Linear-Interpolation und kontinuierlicher gleichmäßiger Geschwindigkeit anfahren
50 CNT 1, 100, 200	Anfangspunktabstand der kontinuierlichen Bewegung auf 100 mm und Endpunktstand der kontinuierlichen Bewegung auf 200 mm festlegen
60 MVS P3	Position P3 mittels Linear-Interpolation und kontinuierlicher gleichmäßiger Geschwindigkeit anfahren
70 CNT 1, 300	Anfangspunktabstand der kontinuierlichen Bewegung auf 300 mm und Endpunktstand der kontinuierlichen Bewegung auf 300 mm festlegen
80 MOV P2	Position P2 mittels Gelenk-Interpolation und kontinuierlicher gleichmäßiger Geschwindigkeit anfahren
90 CNT 0	Sperren der CNT-Einstellung
100 MOV P5	Position P5 mittels Gelenk-Interpolation und Beschleunigung/Verzögerung anfahren

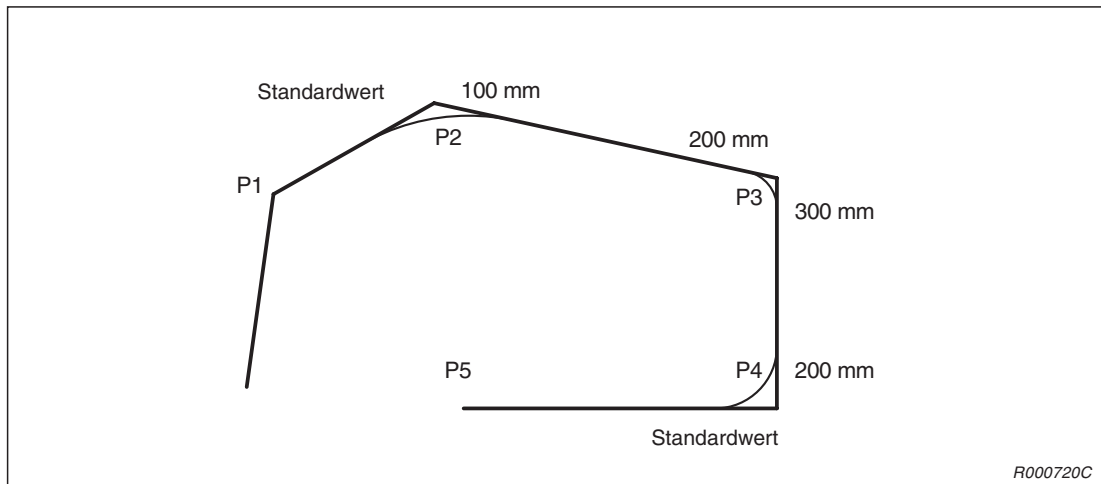


Abb. 9-1: *Verfahrweg für die verschiedenen CNT-Einstellungen*

Erläuterung

- Um die Abstände der Start- und der Endposition der kontinuierlichen Bewegung zu verkleinern, ist einer der beiden eingestellten Werte zu verringern.
- Standardmäßig ist die CNT-Einstellung gesperrt.
- Die numerischen Werte 1 und 2 werden bei fehlender Angabe auf den Standardwert gesetzt.
- Fehlt die Angabe für den numerischen Wert 2, wird er auf den gleichen Wert wie der numerische Wert 1 gesetzt.
- Bei freigegebener CNT-Einstellung ist die FINE-Einstellung gesperrt.
- Bei kleinen numerischen Werten kann das Durchlaufen des Verfahrweges länger dauern als bei deaktivierter CNT-Einstellung.

9.3.12 COM OFF (Communication OFF)

Funktion: Kommunikations-Interrupt sperren

Sperrt die Interrupts von Kommunikationskanälen.

Eingabeformat

```
COM  [( <Nummer des Kommunikationskanals> )]  OFF
```

<Nummer des Kommunikationskanals> Legt die Nummer des Kommunikationskanals fest (z. B. 1, 2 oder 3)

Programmbeispiel

```
10 COM(1) OFF           Sperrt den Kommunikations-Interrupt des Kanals 1
```

Erläuterung

- Nach Ausführung des COM OFF-Befehls bleibt der Interrupt auch bei anstehenden Nachrichten gesperrt.
- Informationen über Kommunikationskanäle sind unter dem Befehl OPEN zu finden.

9.3.13 COM ON (Communication ON)

Funktion: Kommunikations-Interrupt freigeben

Gibt die Interrupts von Kommunikationskanälen frei.

Eingabeformat

```
COM □ [(<Nummer des Kommunikationskanals>)] □ ON
```

<Nummer des Kommunikationskanals> Legt die Nummer des Kommunikationskanals fest (z. B. 1, 2 oder 3)

Programmbeispiel

```
10 COM(1) ON           Gibt den Kommunikations-Interrupt des Kanals 1 frei
```

Erläuterung

- Informationen über Kommunikationskanäle sind unter dem Befehl OPEN zu finden.

9.3.14 COM STOP (Communication STOP)

Funktion: Kommunikations-Interrupt stoppen

Stoppt die Interrupts von Kommunikationskanälen.

Eingabeformat

```
COM □ [( <Nummer des Kommunikationskanals> )] □ STOP
```

<Nummer des Kommunikationskanals> Legt die Nummer des Kommunikationskanals fest (z. B. 1, 2 oder 3)

Programmbeispiel

10 COM(1) STOP Gibt den Kommunikations-Interrupt des Kanals frei 1

Erläuterung

- Nach Ausführung des COM STOP-Befehls wird der Interrupt auch bei anstehenden Nachrichten nicht generiert. Die anstehenden Daten und der Interrupt werden aufgezeichnet und beim nächsten Öffnen des Kanals abgearbeitet.
- Informationen über Kommunikationskanäle sind unter dem Befehl OPEN zu finden.

9.3.15 DEF ACT (Define act)

Funktion: Interrupt-Prozess definieren

Legt den Status und die Ausführung eines Interrupts fest.

Eingabeformat

```
DEF □ ACT □ <Priorität>, <Ausdruck> □ <Prozess> [, L]
```

- <Priorität> Gibt die Priorität des Interrupts an
1 ≤ Priorität ≤ 8
- <Ausdruck> Folgende Formate können für den Interrupt-Status verwendet werden (siehe auch Syntaxdiagramm):
<Num. Datentyp> <Vergleichsoperator> <Num. Datentyp> oder
<Num. Datentyp> <Logischer Operator> <Num. Datentyp>.
Die Angabe <Numerischer Datentyp> bezieht sich auf:
<Numerische Konstanten>|<Numerische Variablen>|<Numerische Feldvariablen>|<Komponentendaten>.
- <Prozess> Legt eine GOTO- oder GOSUB-Anweisung fest, die bei einem Interrupt ausgeführt wird
- L Die Angabe „L“ legt fest, dass der Interrupt-Prozess nach Abarbeitung der Zeile ausgeführt wird. Bei fehlender Angabe wird der Interrupt-Prozess sofort ausgeführt.

Programmbeispiel

10	DEF ACT 1, M_IN(17) = 1 GOSUB 100	Definiert einen Unterprogramm- sprung zu Zeile 100, wenn der Status des allgemeinen Eingangssignals Nummer 17 = EIN ist
20	DEF ACT 2, MFG1 AND MFG2 GOTO 200	Definiert einen Programmsprung zu Zeile 200, wenn das Resultat der UND-Verknüpfung von MFG1 und MFG2 „wahr“ ist
30	DEF ACT 3, M_TIMER(1) > 10.5 GOSUB 300	Definiert nach Ablauf von 10,5 s einen Unterprogramm- sprung zu Zeile 300
100	M_TIMER(1) = 0	Zähler zurücksetzen
110	ACT 3 = 1	Interrupt 3 freigeben

Erläuterung

- Die Prioritäten der Interrupts sind in aufsteigender Reihenfolge von 1 bis 8 festgelegt.
- Über die Priorität können bis zu 8 Interrupts unterschieden werden.
- Haben zwei Interrupts dieselbe Priorität, ist der später definierte Interrupt vorrangig.
- Der DEF ACT-Befehl definiert nur den Interrupt. Mit dem ACT-Befehl wird der Status des Interrupts festgelegt.
- Der Kommunikations-Interrupt (COM) hat eine höhere Priorität als Interrupts, die mit dem DEF ACT-Befehl definiert wurden.
- DEF ACT-Definitionen sind nur in dem Programm wirksam, in dem sie definiert wurden. In einem Unterprogramm müssen sie gegebenenfalls neu definiert werden.
- Wird ein Interrupt durch eine GOTO-Anweisung in einem DEF ACT-Befehl generiert, bleibt der Interrupt während der Abarbeitung des verbleibenden Programmteils erhalten und es werden nur Interrupts höherer Priorität akzeptiert. Der Interrupt kann durch die Ausführung der END-Anweisung deaktiviert werden.

9.3.16 DEF FN (Define function)

Funktion: Funktion definieren

Definiert eine Funktion und legt den Namen fest.

Eingabeformat

```
DEF □ FN <Name> [( <Formalparameter> [, <Formalparameter>] ... )]
                = <Funktionsausdruck>
```

- <Name> Besteht aus einem Zeichen zur Identifizierung und einer Zeichenkette
- <Formalparameter> Legt die Variablen der Funktion fest
Es können maximal 16 Variablen verwendet werden.
- <Funktionsausdruck> Legt die Rechenoperation fest.

Programmbeispiel

```
10 DEF FNMAVE(MA,MB) = (MA+MB)/2      Legt fest, dass durch FNMAVE der Durchschnitt von zwei numerischen Variablen gebildet wird
20 MDATA1 = 20                        Weist MDATA1 den Wert 20 zu
30 MDATA2 = 30                        Weist MDATA2 den Wert 30 zu
40 MAVE = FNMAVE(MDATA1,MDATA2)    Der Durchschnitt von 20 und 30 (= 25) wird der numerischen Variablen MAVE zugewiesen.
```

Erläuterung

- Durch FN und <Name> wird der Name der Funktion festgelegt. Der Funktionsname kann bis zu 8 Zeichen lang sein.
Beispiel:
Numerischer Typ ... FNMAX Identifizierungszeichen: M
Zeichenkettentyp ... FNCAME\$ Identifizierungszeichen: C (Wird durch „\$“ abgeschlossen)
- Eine mit DEF FN definierte Funktion heißt benutzerdefinierte Funktion.
- Es können Funktionen bis zu maximal einer Zeilenlänge beschrieben werden.
- Im <Ausdruck> können fest definierte und schon vorher vom Benutzer definierte Funktionen verwendet werden. In diesem Fall können 16 Ebenen von benutzerdefinierten Funktionen verwendet werden.
- Wenn die Variablen im Funktionsausdruck nicht in den Formelparametern aufgeführt wurden, werden für die Variablen die augenblicklichen Werte verarbeitet. Es tritt eine Fehlermeldung auf, wenn die Anzahl oder der Typ der verwendeten Variablen (numerische oder Zeichenkette) von den deklarierten abweichen.
- Eine benutzerdefinierte Funktion steht nur in dem Programm zur Verfügung, in dem sie definiert worden ist. Sie kann von einem anderen Programm nicht durch einen CALLP-Befehl aufgerufen werden.

9.3.17 DEF PLT (Define pallet)

Funktion: Palette definieren

Definiert eine Palette.

Eingabeformat

```
DEF □ PLT □ <Palettennummer>, <Bezugsposition>,
           <Spaltenendpunkt A>, <Zeilenendpunkt B>,
           [<Paletteneckpunkt, der gegenüber der
           Bezugsposition liegt>],
           <Anzahl der Spaltengitterpunkte A>,
           <Anzahl der Zeilengitterpunkte B>,
           <Bewegungsrichtung>
```

<Palettennummer>	Legt die Nummer der eingesetzten Palette fest $1 \leq \text{Palettennummer} \leq 8$
<Bezugsposition>	Legt den Anfangspunkt der Palette fest Es können Ausdrücke mit Positionsoperationen verwendet werden.
<Spaltenendpunkt A>	Legt einen der Endpunkte der Palette fest Dient als Zwischenposition für kreisförmige Paletten Es können Ausdrücke mit Positionsoperationen verwendet werden.
<Zeilenendpunkt B>	Legt einen der Endpunkte der Palette fest Dient als Zwischenposition für kreisförmige Paletten Es können Ausdrücke mit Positionsoperationen verwendet werden.
<Paletteneckpunkt, der gegenüber der Bezugsposition liegt>	Legt den Punkt fest, der gegenüber der Bezugsposition liegt. Für kreisförmige Paletten ohne Bedeutung Es können Ausdrücke mit Positionsoperationen verwendet werden.
<Anzahl der Spaltengitterpunkte>	Legt die Gitterpunkte der Palette in Spaltenrichtung fest Legt bei kreisförmiger Palette die Positionen zwischen dem Anfangs- und Endpunkt fest Es können Ausdrücke mit numerischen Operationen verwendet werden.
<Anzahl der Zeilengitterpunkte>	Legt die Gitterpunkte der Palette in Zeilenrichtung fest Für kreisförmige Paletten ohne Bedeutung (0 usw., muss festgelegt werden) Es können Ausdrücke mit numerischen Operationen verwendet werden.
<Bewegungsrichtung>	Die Eingabe von „1“, „2“ oder „3“ legt die Bewegungsrichtung fest. 1 = Zickzack 2 = Bewegungsrichtung beibehalten (z. B. von links nach rechts) 3 = kreisförmige Bewegung

Programmbeispiel

10 DEF PLT 1,P1,P2,P3, ,4,3,1 Palettendefinition mit 3 Punkten
20 DEF PLT 1,P1,P2,P3,P4,4,3,1 Palettendefinition mit 4 Punkten

Erläuterung

- Es können 3 oder 4 Punkte einer Palette definiert werden. Die Festlegung von 3 Punkten ist einfacher zu programmieren. Durch die Festlegung von 4 Punkten erreicht man eine höhere Präzision.
- Der Befehl steht nur innerhalb des ausgeführten Programms zur Verfügung. Er kann nicht von einem anderen Programm aufgerufen werden. Falls nötig, muss er erneut definiert werden.
- Die Anzahl der Zeilen- und Spaltengitterpunkte muss größer als Null sein. Ansonsten erfolgt eine Fehlermeldung.
- Wenn das Produkt <Anzahl der Spaltengitterpunkte> x <Anzahl der Zeilengitterpunkte> den Wert 32 767 überschreitet, erfolgt eine Fehlermeldung.
- Die Angabe der Anzahl der Zeilengitterpunkte ist für kreisförmige Paletten ohne Bedeutung, darf aber nicht weggelassen werden. Der Paletteneckpunkt, der gegenüber der Bezugsposition liegt, ist wirkungslos, auch wenn er festgelegt wurde.

9.3.18 DEF INTE/FLOAT/DOUBLE (Define Integer/Float/Double)

Funktion: Arithmetische Variable deklarieren

Definiert eine arithmetische Variable.

Eingabeformat

```
DEF □ INTE □ <Name einer arithmetischen Variablen>
    [,<Name einer arithmetischen Variablen>] ...
DEF □ FLOAT □ <Name einer arithmetischen Variablen>
    [,<Name einer arithmetischen Variablen>] ...
DEF □ DOUBLE □ <Name einer arithmetischen Variablen>
    [,<Name einer arithmetischen Variablen>] ...
```

<Name einer arithmetischen Variablen>

Legt den Variablennamen fest

Programmbeispiel

10	DEF INTE WORK	Deklariert WORK als Namen einer arithmetischen Variablen
20	WORK = 100	Setzt den Wert der arithmetischen Variablen WORK auf 100

Erläuterung

- Der Variablenname kann bis zu 8 Zeichen lang sein. Detaillierte Hinweise zu den verwendbaren Zeichentypen finden Sie in Abs. 8.1.5.
- Bei Deklaration mehrerer Variablennamen dürfen maximal 123 Zeichen (inklusive Befehl) in einer Zeile verwendet werden.
- Eine mit dem Befehl INTE deklarierte Variable ist vom Typ Integer (–32768 bis +32767).
- Eine mit dem Befehl FLOAT deklarierte Variable ist eine arithmetische Variable mit einfacher Genauigkeit ($\pm 1,70141E+18$).
- Eine mit dem Befehl DOUBLE deklarierte Variable ist eine arithmetische Variable mit doppelter Genauigkeit ($\pm 1,701411834604692E+308$).

9.3.19 DEF IO (Define IO)

Funktion: Ein-/Ausgangsvariable definieren

Definiert eine Ein-/Ausgangsvariable.

Eingabeformat

```
DEF □ IO □ <E/A-Variablenname> = <Variablentyp>,
          <E/A-Bitnummer>] [, <Maskeninformation>]
```

- <E/A-Variablenname> Legt den Variablennamen fest
- <Variablentyp> Legt fest, ob die Variable vom Typ BIT, BYTE, WORD oder INTEGER ist
- <E/A-Bitnummer> Legt die Nummer des Eingangs-/Ausgangsbits fest
- <Maskeninformation> Legt fest, ob ein bestimmtes Signal zugelassen wird

Programmbeispiel

- 10 DEF IO PORT1 = BIT,6 Weist der Ein-/Ausgangsvariablen mit dem Namen PORT1 das Ein-/Ausgangsbit Nr. 6 vom Typ BIT zu
- 20 DEF IO PORT2 = BYTE,8,&H0F Weist der Ein-/Ausgangsvariablen mit dem Namen PORT2 das Ein-/Ausgangsbit Nr. 8 vom Typ BYTE zu und legt die Maske auf hexadezimal 0F fest

Erläuterung

- Der Variablenname kann bis zu 8 Zeichen lang sein. Detaillierte Hinweise zu den verwendbaren Zeichentypen finden Sie in Abs. 8.1.5.
- Ist eine Maske angegeben, wird nur ein bestimmtes Signal zugelassen.

Beispiel ▾

In Zeile 20 wird ein Maskierungsprozess mit dem hexadezimalen Wert 0F ausgeführt, so dass die Bits Nummer 8 bis 11 zugelassen und die Bits Nummer 12 bis 15 gesperrt werden.

```
zugelassen  gesperrt
1 1 1 1 0 0 0 0
8                            15 (E/A-Bitnummer)
```



Beispiel ▾

Mit Hilfe der DEF I/O-Funktion ist es möglich, sowohl einen Eingang, als auch einen Ausgang zu definieren. Generell lässt sich die Funktion wie folgt beschreiben:

Wird die Variable auf der linken Seite eines mathematischen Ausdrucks benutzt, wird sie als Ausgang festgelegt.

Wird die Variable auf der rechten Seite eines mathematischen Ausdrucks eingesetzt, wird sie als Eingang festgelegt.

10 DEF IO PORT1 = BIT,6

20 PORT1 = 1

Ausgang 6 wird eingeschaltet

30 M1 = PORT1

Das Eingangssignal von Eingang 6 wird in M1 eingelesen.

40 IF PORT1 = 1 THEN GOTO 100

Sonderfall!

Eingang 6 wird auf log. 1 geprüft

△

9.3.20 DEF JNT (Define Joint)

Funktion: Gelenkvariable definieren

Definiert eine Gelenkvariable.

Eingabeformat

```
DEF □ JNT □ <Gelenkvariablenname> [, <Gelenkvariablenname>] ...
```

<Gelenkvariablenname> Legt den Namen der Gelenkvariablen fest

Programmbeispiel

10	DEF JNT SAFE	Deklariert SAFE als Namen einer Gelenkvariablen
20	MOV SAFE	Fährt die Position SAFE an

Erläuterung

- Der Variablenname darf aus maximal 8 Zeichen bestehen. Detaillierte Hinweise zu den verwendbaren Zeichentypen finden Sie in Abs. 8.1.5.
- Für die Variablen gelten die Regeln für Gelenkvariablen.
- Die mit DEF JNT deklarierten Variablen können wie die mit „J“ festgelegten Variablen verwendet werden.

9.3.21 DEF POS (Define Position)

Funktion: Positionsvariable definieren

Definiert eine Positionsvariable.

Eingabeformat

```
DEF □ POS □ <Positionsvariablenname>
      [,<Positionsvariablenname>] ...
```

<Positionsvariablenname>

Legt den Namen der Positionsvariablen fest

Programmbeispiel

```
10 DEF POS WORKSET
```

Deklariert WORKSET als Namen einer Positionsvariablen

```
20 MOV WORKSET
```

Führt die Position WORKSET an

Erläuterung

- Der Variablenname darf aus maximal 8 Zeichen bestehen. Detaillierte Hinweise zu den verwendbaren Zeichentypen finden Sie in Abs. 8.1.5.
- Bei Deklaration mehrerer Variablenamen dürfen maximal 123 Zeichen (inklusive des Befehls) in einer Zeile verwendet werden.
- Für die Variablen gelten die Regeln für Positionsvariablen.
- Die mit DEF POS deklarierten Variablen können wie die mit „P“ festgelegten Variablen verwendet werden.

9.3.22 DEF CHAR (Define Character)

Funktion: Zeichenkettenvariable definieren

Definiert eine Zeichenkettenvariable.

Eingabeformat

```
DEF □ CHAR □ <Name der Zeichenkettenvariablen>
           [,<Name der Zeichenkettenvariablen>] ...
```

<Name der Zeichenkettenvariablen> Legt den Namen der Zeichenkettenvariablen fest

Programmbeispiel

```
10 DEF CHAR MESSAGE                      Deklariert MESSAGE als Namen einer
                                           Zeichenkettenvariablen
20 MESSAGE = "WORKSET"                    Schreibt die Zeichenkette WORKSET in die
                                           Zeichenkettenvariable MESSAGE
```

Erläuterung

- Der Variablenname darf aus maximal 8 Zeichen bestehen. Detaillierte Hinweise zu den verwendbaren Zeichentypen finden Sie in Abs. 8.1.5.
- Bei Deklaration mehrerer Variablenamen dürfen maximal 123 Zeichen (inklusive des Befehls) in einer Zeile verwendet werden.
- Für die Variablen gelten die Regeln für Zeichenkettenvariablen.
- Die mit DEF CHAR deklarierten Variablen können wie die mit „C“ festgelegten Variablen verwendet werden.

9.3.23 DIM (Dim)

Funktion: Dimension definieren

Legt die Anzahl der Elemente bei Feldvariablen fest. Es sind bis zu drei Dimensionen möglich.

Eingabeformat

```
DIM □ <Variablenname>(<max. Indexwert>, [, <max. Indexwert>])
      [, <Variablenname>(<max. Indexwert>
      [, <max. Indexwert>])]
```

<Variablenname> Legt den Namen für die Feldvariable fest

<maximaler Indexwert> Konstante, die die Anzahl der Elemente einer Feldvariablen festlegt
 $1 \leq \text{Maximaler Indexwert} \leq 999$
 Der maximale Indexwert darf nur Konstanten enthalten.
 Ausdrücke mit numerischen Operationen sind nicht erlaubt.

Programmbeispiel

10	DIM PDATA(10)	Deklariert die Positions-Feldvariable PDATA als eine Variable mit 10 Elementen
20	DIM MDATA(2,3)	Deklariert die zweidimensionale numerische Feldvariable MDATA als eine Variable mit 2 x 3 Elementen

Erläuterung

- Es sind ein-, zwei- und dreidimensionale Felder erlaubt.
- Bei Überschreitung des Wertebereiches für den maximalen Indexwert erfolgt bei der Ausführung der DIM-Anweisung eine Fehlermeldung.
- Bei Ausführung der DIM-Anweisung sind die Feldvariablen auf keine Standardwerte gesetzt, sondern nicht definiert.
- Es können keine Felder ohne DIM-Anweisung verwendet werden. Ist die Anzahl der Elemente eine reelle Zahl, wird die Zahl automatisch auf eine Integer-Zahl gerundet.
- Die DIM-Anweisung steht nur innerhalb des ausgeführten Programmes zur Verfügung. Sie kann nicht von einem anderen Programm aufgerufen werden.
- Bei Verwendung in einem Unterprogramm muss die DIM-Anweisung erneut definiert werden. Ein teachbarer Wert kann direkt in eine mit der DIM-Anweisung deklarierte Positions-Feldvariable geschrieben oder aus ihr ausgelesen werden.

9.3.24 DLY (Delay)

Funktion: Verzögerung einstellen

Als einzelner Befehl wird zur festgelegten Zeit der Wartestatus erzeugt. Wird der DLY-Befehl für einen zusätzlichen Impulsausgang genutzt, wird die Impulsdauer festgelegt.

Eingabeformat

```
DLY □ <Zeit>
```

<Zeit> Legt die Dauer des Wartestatus oder die Impulsdauer in Sekunden fest
Der kleinste einstellbare Wert ist 0,05 s.

Programmbeispiel

```
10 DLY 30                      Wartezeit von 30 Sekunden
20 M_OUT(17) = 1 DLY 0.5      Setzt das allgemeine Ausgangssignal 17 für 0,5 s auf „1“
```

Erläuterung

- Der DLY-Befehl wird verwendet, um in Programmen Verzögerungszeiten zu erzeugen. Ebenso lässt sich die Impulsdauer eines Ausgangssignals in der OUT-Anweisung festlegen.
- Der Impulsausgang wird gleichzeitig mit Ausführung des in der nächsten Zeile stehenden Befehls gesetzt.
- Es können bis zu 4 Impulsausgänge gleichzeitig gesteuert werden. Wird dieser Wert überschritten, kommt es bei Ausführung des Befehls zu einer Fehlermeldung.
- Nach Ablauf der festgesetzten Zeit wird wieder der Zustand vor Ausführung des Befehls angenommen.
- Wird während der festgesetzten Zeit eine END-Anweisung, die letzte Zeile des Programms oder ein NOT-HALT ausgeführt, behält der Impulsausgang seinen gegenwärtigen Zustand bei.
- Die Reihenfolge der Prioritäten ist:
COM > ACT > WTHIF (WTH) > Impulsausgang (Zeitintervall aktiv).
- Wird während der Ausgabe eines Impulses ein Stoppsignal eingegeben, wird die Ausgabe des Impulses nicht unterbrochen.

Beispiel ▾

Wird in folgendem Programm in Zeile 20 ein Stoppsignal eingegeben, wird die Ausführung des Programms unterbrochen, der Ausgangszustand bleibt jedoch erhalten.

```
10 M_OUT(17) = 1
20 DLY 10
30 M_OUT(17) = 0
```



9.3.25 ERROR (Error)

Funktion: Fehler generieren

Im Anwendungsprogramm wird ein Fehler erzeugt.

Eingabeformat

```
ERROR □ <Fehlernummer>
```

<Fehlernummer> Es kann eine Konstante oder ein numerischer Ausdruck festgelegt werden.
 $9000 \leq \text{Fehlernummer} \leq 9299$

Programmbeispiel

100 ERROR 9000 Generiert den Fehler mit der Fehlernummer 9000

Erläuterung

- Der ERROR-Befehl wird verwendet, um einen Fehler an ein externes Gerät zu übertragen.
- In Abhängigkeit der Fehlernummer reagiert das System wie in folgender Tabelle gezeigt:

Fehlernummer	Systemverhalten
9000–9099 (schwerer Fehler)	Die Programmausführung wird unterbrochen und die Servospannung abgeschaltet. Der Fehler wird bei einem Reset zurückgesetzt.
9100–9199 (leichter Fehler)	Die Programmausführung wird unterbrochen. Der Fehler wird bei einem Reset zurückgesetzt.
9200–9299 (Warnung)	Die Programmausführung wird nicht unterbrochen. Der Fehlerausgang wird bei einem Reset eingeschaltet.

Tab. 9-2: Systemverhalten in Abhängigkeit der Fehlernummer

9.3.26 END (End)

Funktion: Programmende

Beendet das Programm.

Eingabeformat

END

Programmbeispiel

100 END

Erläuterung

- Es können mehrere END-Anweisungen in einem Programm ausgeführt werden.
- Es muss nicht zwingend eine END-Anweisung an das Ende eines Programms gesetzt werden.
- Eine END-Anweisung, die durch einen CALLP-Befehl aufgerufen wird, übergibt die Kontrolle an das Programm, in dem der CALLP-Befehl ausgeführt wurde.
- Eine END-Anweisung im Hauptprogramm schließt alle geöffneten Dateien.
- Bei Ausführung der END-Anweisung werden die durch folgende Befehle eingestellten Werte zurückgesetzt: SPD, ACCEL, OADL, JOVRD, OVRD, FINE und CNT.
- Das Ausführen der END-Anweisung führt nicht zwangsläufig zum Beenden des Programms. Durch das Umstellen des Parameters SLOTON auf den Wert 2 und Verändern des Parameters SLT1 von ..., REP, ..., ... auf ..., CYC, ..., ... bleibt der Roboter bei Erreichen von END stehen.

9.3.27 FINE (Fine)

Funktion: Feinpositionierung

Legt den Status bei der Beendigung eines Interpolationsbefehls fest, wenn die CNT-Einstellung gesperrt ist.

Eingabeformat

```
FINE  <sperr/en/Anzahl der Impulse> [, <Achse>]
```

<Anzahl der Impulse>	Festlegung der Anzahl der Impulse zur Positionierung Bei einer Einstellung von „0“ (Standardwert) ist die Feinpositionierung deaktiviert.
<Achse>	Festlegung der Achse, für die die Feinpositionierung ausgeführt werden soll Bei fehlender Angabe ist die Feinpositionierung für alle Achsen aktiviert.

Programmbeispiel

10	FINE 300	FINE-Einstellung auf 300 Impulse festlegen
20	MOV P1	Position P1 mittels Gelenk-Interpolation anfahren
30	FINE 100,2	Feinpositionierung für Achse 2 auf 100 Impulse festlegen
40	MOV P2	Position P2 mittels Gelenk-Interpolation anfahren
50	FINE 0	Feinpositionierung deaktivieren
60	MOV P3	Position P3 mittels Gelenk-Interpolation anfahren
70	FINE 100	Feinpositionierung für alle Achsen 100 Impulse festlegen
80	MOV P4	Position P2 mittels Gelenk-Interpolation anfahren

Erläuterung

- Während einer Programmabarbeitung ist die FINE-Einstellung solange gesperrt, bis sie durch das Programm freigegeben wird. Sobald die FINE-Einstellung freigegeben wurde, bleibt sie solange freigegeben, bis sie erneut gesperrt wird.
- Nach Abarbeitung des Programms wird die FINE-Einstellung gesperrt.
- Ist die CNT-Einstellung freigegeben, wird der FINE-Befehl ignoriert. Er wird auch dann ignoriert, wenn er freigegeben ist (d. h. er wird als gesperrt interpretiert, die Einstellung bleibt jedoch erhalten).

9.3.28 FOR-NEXT (For-Next)

Funktion: Programmschleife

Dieser Befehl bewirkt eine Wiederholung des Programmteils, der zwischen der FOR-Anweisung und der NEXT-Anweisung steht. Der Programmteil wird solange wiederholt, bis die Abbruchbedingungen erfüllt sind.

Eingabeformat

```
FOR □ <Zähler> = <Vorgabewert> TO <Endwert> [STEP<Schrittwert>]
:
NEXT □ [<Zähler 1> [,<Zähler 2>] ...]
```

<Zähler>	Der numerische Datentyp gibt die Anzahl der Wiederholungen der Programmschleife an. Gilt auch für Zähler 1 und Zähler 2 usw.
<Vorgabewert>	Gibt den Startwert des Zählers vor
<Endwert>	Gibt den Endwert des Zählers vor
<Schrittwert>	Legt die Schrittweite des Zählers fest

Programmbeispiel

Programm zur Addition der Zahlen 1 bis 10

10	MSUM = 0	Weist MSUM den Wert 0 zu
20	FOR M1 = 1 TO 10 STEP 1	Setze den Vorgabewert, den Endwert und die Schrittweite der Variablen M1
30	MSUM = MSUM + M1	Addiert M1 zu der numerischen Variablen MSUM
40	NEXT M1	Sprung zu Zeile 20
50	END	Programmende

Speichert das Produkt zweier numerischer Variablen in eine zweidimensionale Feldvariable (Beispiel für verschachtelte FOR-NEXT-Programmschleifen)

10	DIM MBOX(10,10)	Reserviert Speicherplatz für eine 10 x 10 Feldvariable
20	FOR M1 = 1 TO 10 STEP 1	Erhöht den Zähler der numerischen Variablen M1 von 1 bis 10 um 1 und springt zu Zeile 70, sobald der Wert 10 überschritten ist („STEP 1“ kann weggelassen werden).
30	FOR M2 = 1 TO 10 STEP 1	Erhöht den Zähler der numerischen Variablen M2 von 1 bis 10 um 1 und springt zu Zeile 60, sobald der Wert 10 überschritten ist („STEP 1“ kann weggelassen werden).
40	MBOX(M1,M2) = MX * MY	Ersetzt die Elemente der Feldvariablen MBOX(M1,M2) durch das Produkt MX * MY.
50	NEXT M2	Sprung zu Zeile 30
60	NEXT M1	Sprung zu Zeile 20
70	END	Programmende

Erläuterung

- Die Programmschleife wird nicht ausgeführt, wenn:
 Der <Vorgabewert> größer als der <Endwert> und der <Schrittwert> positiv ist.
 Der <Vorgabewert> kleiner als der <Endwert> und der <Schrittwert> negativ ist.
- Widersprechen sich die FOR- und die NEXT-Anweisung, erfolgt eine Fehlermeldung. Werden FOR-NEXT-Programmschleifen verschachtelt verwendet und sie haben den gleichen Endwert, ist es möglich, nur eine NEXT-Anweisung zu verwenden. Die Zeilen 50 und 60 im Programmbeispiel können zu einer Zeile zusammengefasst werden: NEXT M2,M1.
- Steht die NEXT-Anweisung in unmittelbarer Beziehung zur nächsten FOR-Anweisung, können die Variablennamen in der NEXT-Anweisung weggelassen werden. „M2“ in Zeile 50 und „M1“ in Zeile 60 im Programmbeispiel können weggelassen werden.
- Programmebenen
 Es ist möglich, FOR-NEXT-Programmschleifen zwischen weiteren FOR-NEXT-Anweisungen zu verwenden. Mit jeder FOR-NEXT-Programmschleife erhöht sich die Zahl der Programmebenen um 1. Ein Programm darf aus maximal 16 Programmebenen bestehen. Bei mehr als 16 Ebenen erfolgt eine Fehlermeldung.
- Schleifen Verschachtelungen

Beispiel ▾

```

10  FOR
    :   :
    :   FOR
    :   :   :
    :   FOR
    :   :   :
    :   :
    :   :
    :   NEXT
    :   :
    :   NEXT
    :   :
    :   NEXT
    
```

Richtig!

```

10  FOR
    :   :
    :   FOR
    :   :   :
    :   FOR
    :   :   :
    :   :
    :   :
    :   NEXT
    :   :
    :   NEXT
    :   :
    :   NEXT
    
```

Falsch!



9.3.29 FPRM (FPRM)

Funktion: Parameter definieren

Legt im Unterprogramm die Reihenfolge, den Typ und die Anzahl von Parametern fest, die von der CALLP-Anweisung eines Hauptprogramms übergeben werden.

Eingabeformat

```
FPRM □ <Formalparameter>[, <Formalparameter>] ...
```

<Formalparameter> Variablenname im Unterprogramm,
 Es können alle Variablen verwendet werden.
 Es dürfen maximal 16 Variablen verwendet werden.

Programmbeispiel

10 FPRM M1,P1,P2 Festlegung der Datentypen, der Reihenfolge und
 der Anzahl

Wenn das Unterprogramm den Namen 20 trägt, so ist die Übergabe der Variablen M4, P3 und P5 vom Hauptprogramm an das Unterprogramm in die Variablen M100, P1 und P30 wie folgt zu definieren:

Hauptprogramm:

```
:
120 CALLP "P20", M4, P3, P5
:
```

Unterprogramm:

```
10 FPRM M100, P1, P30
```

Die Variableninhalte von M4 wird an M100, P3 an P1 und P5 an P30 übergeben.

Erläuterung

- Der FPRM-Befehl wird nicht benötigt, wenn im aufgerufenen Unterprogramm keine Parameter verwendet werden.
- Eine Variable, die nicht als Formalparameter aufgeführt ist, behält ihren aktuellen Wert bei.
- Programmebenen
 Der Aufruf von Unterprogrammen erlaubt eine Programmstruktur mit mehreren Ebenen. Jeder Programmaufruf bedeutet dabei eine zusätzliche Ebene. Es dürfen maximal 7 Ebenen verwendet werden. Bei mehr als 7 Ebenen erfolgt eine Fehlermeldung.
- Rechenergebnisse eines Unterprogramms können nicht mittels temporärer Parameter in das Hauptprogramm übertragen werden. Verwenden Sie zu diesem Zweck externe Variablen.

9.3.30 GETM (Get Mechanism)

Funktion: Roboter definieren

Definiert einen Roboter.

Eingabeformat

```
GETM □ <Roboter Nummer>
```

<Roboter Nummer> Festlegung der Roboter Nummer über einen numerischen Wert oder eine Variable
Der Roboterarm in einem Standardsystem wird als Roboter 1 definiert.

Programmbeispiel

10	GETM 1	Definition des Roboters 1
20	SERVO ON	Servospannung des Roboters 1 einschalten
30	MOV P1	Position 1 mittels Gelenk-Interpolation anfahren
40	MVS P2	Position 2 mittels Linear-Interpolation anfahren
45	P3 = P_CURR	Aktuelle Position von Roboter 1 in P3 übertragen
50	SERVO OFF	Servospannung des Roboters 1 ausschalten
60	RELM	Definition von Roboter 1 aufheben
70	END	Programmende

Erläuterung

- Bewegungsbefehle und Befehle zum Ein- und Ausschalten der Servomotoren können nur nach Definition eines Roboters über den GETM-Befehl ausgeführt werden.
- Der GETM-Befehl kann nicht für einen Roboter ausgeführt werden, der bereits über den Befehl definiert worden ist.
- Bei fehlender Angabe des Arguments werden den Roboterstatusvariablen, die eine Roboterdefinition benötigen, die Werte des aktuellen Roboters zugewiesen.
- Der Anwendung 1 ist standardmäßig der Roboter 1 zugeordnet.
- Bei einem Programmstopp wird der Befehl RELM automatisch durch das System ausgeführt. Bei einem Neustart des Programms wird der Befehl GETM automatisch ausgeführt.

9.3.31 GOSUB (Go Subroutine)

Funktion: Sprung zu einem Unterprogramm

Bewirkt einen Sprung zu einem Unterprogramm, das mit einer festgelegten Zeilennummer oder einer Marke beginnt.

Eingabeformat

```
GOSUB □ <Sprungziel>
```

<Sprungziel> Legt eine Zeilennummer oder eine Marke fest

Programmbeispiel

10	GOSUB 300	Sprung zum Unterprogramm (Zeile 300)
20	GOSUB *LABEL	Sprung zum Unterprogramm LABEL
30	MOV P10	Position P10 anfahren
40	MOV P11	Position P11 anfahren
50	GOTO 10	Sprung zur Programmzeile 10
	:	
300	MOV P1	Position P1 anfahren
310	MOV P2	Position P2 anfahren
320	RETURN	Rücksprung zur Zeile 20
	:	
500	*LABEL	Sprungmarke „LABEL“ festgelegt
510	MOV P8	Position P8 anfahren
520	MOV P20	Position P20 anfahren
530	RETURN	Rücksprung zur Zeile 30
	:	

Erläuterung

- Zum Aufruf eines Unterprogramms im Basisprogramm verwenden Sie als erstes und zweites Zeichen „L_“. Existiert der aufgerufene Markenname auch im lokalen Programm, erfolgt der Sprung zu dieser Marke.

9.3.32 GOTO (Go To)

Funktion: Sprung zu einer Programmzeile oder Marke

Bewirkt einen unbedingten Sprung zu einer festgelegten Zeilennummer oder Marke.

Eingabeformat

GOTO □ <Sprungziel>

<Sprungziel> Legt eine Zeilennummer oder eine Marke fest

Programmbeispiel

100 GOTO 500	Sprung in die Programmzeile 500
:	
500 MOV P1	Position P1 anfahren
:	
700 GOTO *LABEL	Sprung zum Unterprogramm LABEL
:	
900 *LABEL	Sprungmarke „LABEL“ festgelegt

Erläuterung

- Der Sprungs in ein Basisprogramm ist nicht möglich. Es erfolgt eine Fehlermeldung.

9.3.33 HLT (Halt)

Funktion: Programmablauf stoppen

Stoppt die Roboterbewegung und den Programmablauf.

Eingabeformat

HLT

Erläuterung

- Unterbricht den Programmablauf und stoppt den Roboter mit der definierten Abbremszeit.
- Ein Neustart kann über die Teaching Box oder durch ein externes Start-Signal erfolgen. Der Programmstart beginnt eine Zeile nach dem HLT-Befehl. Wurde der HLT-Befehl in einer Verknüpfung ausgeführt, startet das Programm in der Zeile, in der es unterbrochen wurde.

9.3.34 HOPEN/HCLOSE (Hand Open/Hand Close)

Funktion: Handgreiferzustand festlegen

Legt den Handgreiferzustand (offen/geschlossen) fest.

Eingabeformat

```
HOPEN □ <Handnummer> [, <Start-Greifkraft>, <Halte-Greifkraft>,
      <Haltezeit für Start-Greifkraft>]
HCLOSE □ <Handnummer>
```

<Handnummer>	Festlegung der Handnummer $1 \leq \text{Handnummer} \leq 8$
<Start-Greifkraft>	Legt die Greifkraft zum Öffnen und Schließen der motorbetriebenen Greifhand fest $0 \leq \text{Start-Greifkraft} \leq 63$ (63 = 3,5 kg) Standardwert: 63 Bei fehlender Angabe wird der vorherige Wert verwendet.
<Halte-Greifkraft>	Legt die Greifkraft fest, die zur Aufrechterhaltung des Handgreiferzustandes der motorbetriebenen Greifhand benötigt wird $0 \leq \text{Halte-Greifkraft} \leq 63$ (63 = 3,5 kg) Standardwert: 63 Bei fehlender Angabe wird der vorherige Wert verwendet.
<Haltezeit für Start-Greifkraft>	Legt die Zeitdauer für die Start-Greifkraft der motorbetriebenen Greifhand fest Standardwert: 0,3 s

Programmbeispiel

```
10 HOPEN 1   Öffnet Hand 1
20 HCLOSE 2  Schließt Hand 2
```

Erläuterung

- Die Handausführung (einfach/doppelt) wird in Parameter HANDTYPE festgelegt.
- Das Handsteuersignal (auf/zu) nach Einschalten der Versorgungsspannung kann über Parameter HANDINIT festgelegt werden.
- Der Handgreiferzustand kann über die Roboterstatusvariable M_HNDCQ abgefragt werden. Auf das Signal kann auch über die Eingangssignale Nr. 900 bis 907 (bei Definition eines Roboters) zugegriffen werden.

```
10 HCLOSE 1           Schließt Hand 1
20 IF M_HNDCQ(1) <> 1 GOTO 20  Wartestatus, bis die Roboterstatusvariable
                               M_HNDCQ(1) gleich 1 ist
30 MOV P1             Position 1 mittels Gelenk-Interpolation
                               anfahren
```

- Auf den Handgreifer bezogene Parameter finden Sie im Kap. 5 „Parameter“.

9.3.35 IF ... THEN ... ELSE (If Then Else)

Funktion: WENN ... DANN ... SONST

WENN eine bestimmte Bedingung zutrifft, DANN führe Anweisung 1 aus, SONST führe Anweisung 2 aus.

Eingabeformat

```
IF  <Ausdruck>  THEN  <Anweisung>  [ELSE <Anweisung>]
```

<Ausdruck> Beschreibt einen booleschen Ausdruck

<Anweisung> Anweisung nach THEN wird ausgeführt, wenn das Ergebnis des booleschen Ausdrucks wahr ist. Anweisung nach ELSE wird ausgeführt, wenn das Ergebnis des booleschen Ausdrucks unwahr ist.

Programmbeispiel

100 IF M1 > 10 THEN 1000	Sprung zu Zeile 1000, falls M1 größer 10
110 IF M1 > 10 GOTO 1000 ELSE GOTO 2000	Sprung zu Zeile 1000, falls M1 größer 10, sonst Sprung zu Zeile 2000

9.3.36 INPUT # (Input)

Funktion: Eingabe

Liest Daten aus Dateien oder Eingabegeräten im ASCII-Format.

Eingabeformat

```
INPUT □ # <Dateinummer>, <Datename> [, <Datename>] ...
```

<Dateinummer>	Legt die Dateinummer fest $1 \leq \text{Dateinummer} \leq 8$
<Datename>	Name der Variablen in die die Daten übertragen werden Es können alle Variablen verwendet werden.

Programmbeispiel

10 OPEN "COM:" AS #1	Weist der RS232C-Schnittstelle die Datei Nr. 1 zu
20 INPUT #1,M1	Erfolgt eine Eingabe von der Tastatur, wird dieser Wert in die numerische Variable M1 übertragen.

Erläuterung

- Überträgt Eingangsdaten aus Dateien (oder von Eingabegeräten), die mittels OPEN-Anweisung geöffnet worden sind, in eine Variable. Ist die OPEN-Anweisung nicht ausgeführt worden, erfolgt eine Fehlermeldung.
- Der übertragene Datentyp und der Variablentyp müssen übereinstimmen.
- Werden mehrere Variablennamen angegeben, müssen sie durch Kommas getrennt werden.
- Bei Ausführung der INPUT-Anweisung wartet das System auf eine Eingabe. Bei Betätigung der Eingabetaste (CR und LF) werden die Eingangsdaten in die Variablen übertragen.
- Beim Senden von Werten an den Roboter ist vor den Daten die Buchstabenfolge PRN zu setzen.
Beispiel:
PRN 50 für die Übergabe des Wertes 50
- Bei Eingabe mehrerer Elemente werden die Elemente der Reihe nach übertragen.
Beispiel:
Bei Eingabe einer Zeichenkette, eines numerischen Wertes und einer Position
10 INPUT #1,C1\$,M1,P1
PRN MELFA,125.75,(130.5,-117.2,55.1,16.2,0,0)(1,0)
„MELFA“ wird in C1\$ übertragen,
125.75 in M1 und
(130.5,-117.2,55.1,16.2,0,0)(1,0) in P1

9.3.37 JOVRD (J Override)

Funktion: Übersteuerung

Legt die Geschwindigkeit für die Gelenk-Interpolation fest.

Eingabeformat

```
JOVRD □ <Übersteuerungswert>
```

<Übersteuerungswert> Legt den prozentualen Übersteuerungswert als reelle Zahl oder als numerischen Ausdruck fest
 $1 \leq \text{Übersteuerungswert} \leq 100.0$

Programmbeispiel

```
10 JOVRD 50           Übersteuerung auf den Wert 50 % einstellen
20 MOV P1            Position P1 anfahren
30 JOVRD M_NOVRD     Standardwert einstellen
```

Erläuterung

- Der JOVRD-Befehl ist nur bei der Gelenk-Interpolation wirksam.
- Die aktuelle Arbeitsgeschwindigkeit ergibt sich folgendermaßen:

$$\text{Gelenk-Interpolation} = \frac{\text{Übersteuerungswert der T/B- oder des Steuergeräts}}{100} \times \text{Einstellwert des OVRD-Befehls} \times \text{Einstellwert des JOVRD-Befehls}$$

- Der Maximalwert der Arbeitsgeschwindigkeit ist 100 %. Der Standardwert der Arbeitsgeschwindigkeit beträgt 100 % der Standardeinstellung (M_NOVRD).
- Die Arbeitsgeschwindigkeit wird bei Ausführung der END-Anweisung und bei einem Reset auf den Standardwert zurückgesetzt.

9.3.38 JRC (Joint Roll Change)

Funktion: Gelenkposition verändern

Roboterarmachse J6

Überschreibt die aktuelle Position durch Addition von $\pm 360^\circ$ zur aktuellen Gelenkposition der J6-Achse.

Benutzerdefinierte Achse

Überschreibt die aktuelle Position durch Addition oder Subtraktion eines festgelegten Wertes zur aktuellen Gelenkposition der gewählten Achse. Der Anwender kann den Wert in einem Parameter festlegen. Als Achse kann sowohl eine Gelenkachse wie auch eine lineare Achse gewählt werden.

Veränderung der Grundposition eines Gelenkes

Eingabeformat

JRC <input type="checkbox"/> <[+]1/-1/0>[, <Achsennummer>] . . .

<+1>	<p>Addiert einen definierten Wert zur aktuellen Position einer gewählte Gelenkachse. Der Wert, um den die Position verändert wird, ist im Parameter JRCQTT festgelegt.</p> <p>Bei der Gelenkachse J6 ist dieser Wert auf 360° festgelegt.</p>
<-1>	<p>Subtrahiert einen definierten Wert von der aktuellen Position einer gewählte Gelenkachse. Der Wert, um den die Position verändert wird, ist im Parameter JRCQTT festgelegt.</p> <p>Bei der Gelenkachse J6 ist dieser Wert auf 360° festgelegt.</p>
<0>	<p>Der aktuelle Wert einer festgelegten Achse wird als Grundposition in den Parameter JRCORG geschrieben.</p> <p>Dieser Befehl kann nur bei einer benutzerdefinierten Achse ausgeführt werden.</p>
<Achsennummer>	<p>Die Zielachse wird durch Jx angegeben, wobei x eine Zahl zwischen 1 und 8 ist.</p> <p>Die Standardachse ist die Roboterarmachse J6.</p>

Verwendete Parameter

JRCEXE	Freigabe des JRC-Befehls (freigegeben/gesperrt = 1/0) Standardeinstellung JRCEXE = 0
JRCQTT	Festlegung der Werte, um die die festgelegten Achsen mit dem JRC-Befehl verschoben werden. Dieser Wert ist bei der Standardachse J6 auf 360° festgelegt.
JRCORG	Festlegung der Grundposition für JRC = 0 (kann nur bei zusätzlichen und benutzerdefinierten Achsen eingestellt werden)

Programmbeispiel

10	MOV P1	Position 1 anfahren
20	JRC 1	Addition von 360° zur aktuellen Position der J6-Achse
30	MOV P1	Position 1 anfahren

9.3.39 LABEL (Label)

Funktion: Sprungmarke

Legt ein Sprungziel fest.

Eingabeformat

*<Name der Marke>

<Name der Marke> Legt den Namen der Marke über eine Zeichenkette fest
Das erste Zeichen muss ein Buchstabe sein.
Die maximale Länge beträgt 8 Zeichen (das (*)-Zeichen wird nicht mitgezählt).

Programmbeispiel

10 *SUB1 Name der Marke ist *SUB1

Erläuterung

- Es erfolgt keine Fehlermeldung, wenn die Marke während eines Programmablaufes nicht aufgerufen wird.
- Ist die gleiche Marke in einem Programm mehrmals definiert, erfolgt eine Fehlermeldung.

9.3.40 LOADSET (Load Set)

Funktion: Hand- und Werkstückbedingung einstellen

Legt die Hand- und Werkstückbedingungen für eine optimale Beschleunigung/Abbremsung fest.

Eingabeformat

```
LOADSET □ <Handnummer>, <Werkstücknummer>
```

- <Handnummer> Legt die Hand für die Einstellung der Größe und des Gewichts für die optimale Beschleunigung/Abbremsung fest
($1 \leq \text{HNDDAT} \leq 8$)
- <Werkstücknummer> Legt das Werkstück für die Einstellung der Größe und des Gewichts für die optimale Beschleunigung/Abbremsung fest
($1 \leq \text{WRKDAT} \leq 8$)

Programmbeispiel

- 5 OADL ON Optimale Beschleunigung/Abbremsung freigeben
- 10 LOADSET 1,1 Optimale Beschleunigung/Abbremsung für Hand 1 und Werkstück 1 einstellen
- 20 MOV P1 Position 1 mittels Gelenk-Interpolation und optimaler Beschleunigung/Abbremsung anfahren
- 30 MOV P2 Position 2 mittels Gelenk-Interpolation und optimaler Beschleunigung/Abbremsung anfahren
- 40 LOADSET 1,2 Optimale Beschleunigung/Abbremsung für Hand 1 und Werkstück 2 einstellen
- 50 MOV P1 Position 1 mittels Gelenk-Interpolation und optimaler Beschleunigung/Abbremsung anfahren
- 60 MOV P2 Position 2 mittels Gelenk-Interpolation und optimaler Beschleunigung/Abbremsung anfahren

Erläuterung

- Die Funktion ermöglicht die Ausführung der Roboterbewegung für verschiedene Handdaten und Werkstücke mit optimaler Beschleunigung/Abbremsung.
- Beim Programmstart wird für die Hand die maximale Last gesetzt.
- Werden mehrere Variablennamen angegeben, müssen sie durch Kommas getrennt werden.
- Stellen Sie das Gewicht, die Maße (X, Y, Z) und den Schwerpunkt (X, Y, Z) der Hand in Parameter HANDDAT (HANDDAT 1 bis 8) ein.
- Stellen Sie das Gewicht, die Maße (X, Y, Z) und den Schwerpunkt (X, Y, Z) des Werkstücks in Parameter WRKDAT (WRKDAT 1 bis 8) ein.
- Die Hand- und Werkzeugbedingungen für die optimale Beschleunigung/Abbremsung werden beim Zurücksetzen des Programms und bei Ausführung der END-Anweisung auf die Standardwerte zurückgesetzt.
- Als Standardwert wird die Handbedingung auf den Lastnennwert und die Werkzeugbedingung auf „keine“ (0 kg) gesetzt.
- Eine detaillierte Beschreibung zur Freigabe der optimalen Beschleunigung/Abbremsung finden Sie in Abs. 9.3.47 „OADL-Befehl“.

9.3.41 MOV (Move)

Funktion: Bewegung mit Gelenk-Interpolation

Bewegt die Handspitze mittels Gelenk-Interpolation zu einer festgelegten Position.

Eingabeformat

```
MOV □ <Zielposition> [, [<Abstand>] [<Interpolationstyp>]]
      [<Verknüpfungsbedingung>]
```

<Zielposition>	Legt die Zielposition fest
<Abstand>	Legt den Verfahrwegbetrag in Werkzeugrichtung auf der Z-Achse fest (Abstand zur Zielposition)
<Interpolationstyp>	Legt den Interpolationstyp fest: TYPE <Numerische Konstante 1>,<Numerische Konstante 2> Numerische Konstante 1 ... indirekte/direkte Anfahrt = 1/0 Numerische Konstante 2 ... reserviert Standardwerte: 1, 0 (indirekte Anfahrt)
<Verknüpfungsbedingung>	Es können die Verknüpfungen WTH und WTHIF verwendet werden.

Programmbeispiel

10	MOV P1 TYPE 1,0	Position 1 anfahren
20	MOV P2	Position 2 anfahren
30	MOV (PLT 1,10),100.0 WTH M_OUT(17) = 1	Palette 1 anfahren und Ausgangsbit 17 auf 1 setzen
40	MOV P4+P5,50.0 WTHIF M_IN(18)=1,M_OUT(20) = 1	Position (4+5) anfahren, wenn Eingangsbit 18 gleich 1 ist, setze Ausgangsbit 20 auf 1

Erläuterung

- Durch die Verknüpfungsbedingungen über die WTH- und WTHIF-Anweisung können die Verfahrbewegung und die Signalausgabe synchronisiert werden.
- Über die numerische Konstante 1 zur Festlegung des Interpolationstyps wird die Interpolation der Stellung definiert.
- Bei der indirekten Anfahrt wird die Position exakt mit der geteachten Stellung angefahren. In Abhängigkeit der geteachten Stellung kann die direkte Anfahrt gewählt werden.
- Bei der direkten Anfahrt wird die Stellung beim Start bis zur Stellung bei Erreichen der Zielposition mit weniger Bewegungen geändert.
- Die Auswahl zwischen direkter und indirekter Anfahrt ist bei einem Bewegungsbereich der Stellungsachse von 180° oder mehr von Bedeutung.
- Liegt die Zielposition bei angewählter direkter Anfahrt außerhalb des Bewegungsbereiches, ist es möglich, dass die Achse in umgekehrter Richtung mit indirekter Anfahrt bewegt wird.
- Die numerische Konstante 2 ist für die Gelenk-Interpolation bedeutungslos.

9.3.42 MVC (Move C)

Funktion: Kreis-Interpolation

Bewegt die Handspitze mittels 3D-Kreis-Interpolation entlang eines durch Startposition, Zwischenposition 1, Zwischenposition 2 und Startposition festgelegten Kreises.

Eingabeformat

```
MVC  <Startposition>, <Zwischenposition 1>, <Zwischenposition 2>
       [<Verknüpfungsbedingung>]
```

<Startposition>	Legt den Start- und Endpunkt des Kreises fest
<Zwischenposition 1>	Legt die erste Zwischenposition auf dem Kreisumfang fest
<Zwischenposition 2>	Legt die zweite Zwischenposition auf dem Kreisumfang fest
<Verknüpfungsbedingung>	Es können die Verknüpfungen WTH und WTHIF verwendet werden.

Programmbeispiel

10	MVC P1,P2,P3	Bewegung entlang des durch P1, P2 und P3 festgelegten Kreises
20	MVC P1,J2,P3	Bewegung entlang des durch P1, J2 und P3 festgelegten Kreises
30	MVC P1,P2,P3 WTH M_OUT(17) = 1	Bewegung entlang des durch P1, P2 und P3 festgelegten Kreises und Setzen des Ausgangsbits 17 auf 1
40	MVC P3,(PLT 1,5),P4 WTHIF M_IN(20) = 1,M_OUT(21) = 1	Bewegung entlang des durch P3, (PLT 1,5) und P4 festgelegten Kreises und Setzen des Ausgangsbits 21, falls Eingangsbit 20 gleich 1

Erläuterung

- Mittels Kreis-Interpolation bewegt sich die Handspitze des Roboters auf dem Kreisumfang des durch die 3 Punkte festgelegten Kreises (360°).
- Während der Kreis-Interpolation bleibt die Orientierung des Roboters unverändert.
- Entspricht die momentane Position nicht der Startposition, fährt der Roboter die Startposition mittels Linear-Interpolation an.
- Wird die Kreisbogenbewegung fortgesetzt, wenn die Kreis-Interpolation unterbrochen und nach einem JOG-Betrieb neu gestartet wurde, bewegt sich der Roboter mittels Linear-Interpolation zu der Position an der die Bewegung unterbrochen wurde und setzt dort die Kreisbogenbewegung fort.

Erläuterung

- Mittels Kreis-Interpolation bewegt sich der Roboterarm auf dem Kreisbogen, der durch die 3 Punkte festgelegt ist.

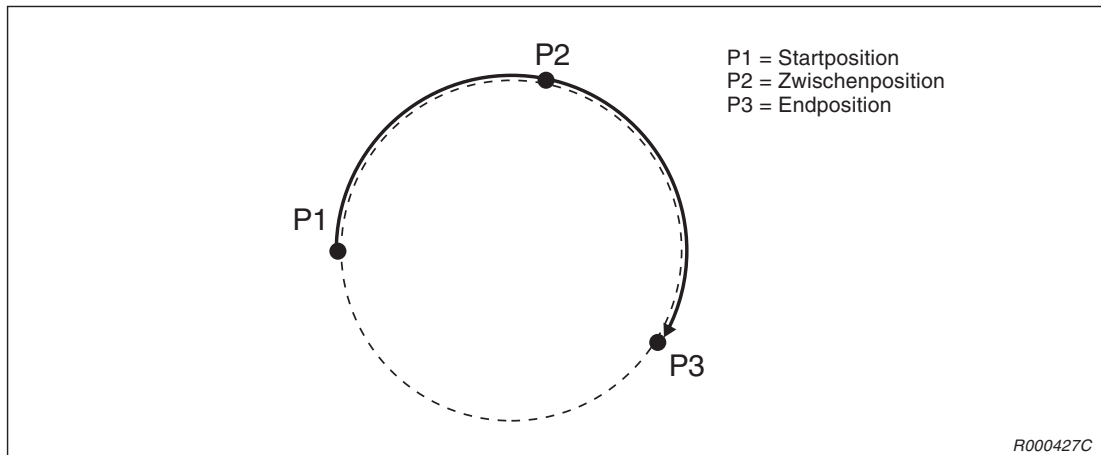


Abb. 9-2: Beispiel zur Kreis-Interpolation über eine Zwischenposition

- Die Roboterstellung wird vom Startpunkt zum Endpunkt interpoliert. Die Stellung der Zwischenposition hat keinen Einfluss.
- Entspricht die aktuelle Position nicht der Startposition, fährt der Roboter die Startposition mittels Linear-Interpolation an.
- Wird die Kreisbogenbewegung fortgesetzt, wenn die Kreis-Interpolation unterbrochen und nach einem JOG-Betrieb neu gestartet wurde, bewegt sich der Roboter mittels Gelenk-Interpolation zu der Stopp-Position und setzt dort die Kreisbogenbewegung fort.
- Weichen die Stellungsmerker der Start- und Endposition für eine andere Interpolationsmethode als die kartesische Interpolation für 3 Achsen voneinander ab, erfolgt eine Fehlermeldung.
- Der Roboter verfährt mit Linear-Interpolation, wenn zwei der drei Positionen gleich sind oder alle Positionen auf einer Geraden liegen. Es erfolgt keine Fehlermeldung.
- Wird über die numerische Konstante 2 die kartesische Interpolation für 3 Achsen gewählt, ist die numerische Konstante 1 unwirksam und der Roboter bewegt sich mit der geteachten Orientierung.
- Die numerische Konstante 2 legt den Interpolationstyp der Stellung fest. Bei einer Interpolation im Koordinatensystem X, Y, Z, J4, J5 und J6 wird die kartesische Interpolation für 3 Achsen verwendet, um den Roboter in die Nähe eines bestimmten Punktes zu bewegen.

Erläuterung

- Mittels Kreis-Interpolation bewegt sich die Roboterhand auf dem Kreisbogen, der durch die 3 Punkte festgelegt ist. Die Bewegung geht nicht durch die Referenzposition.

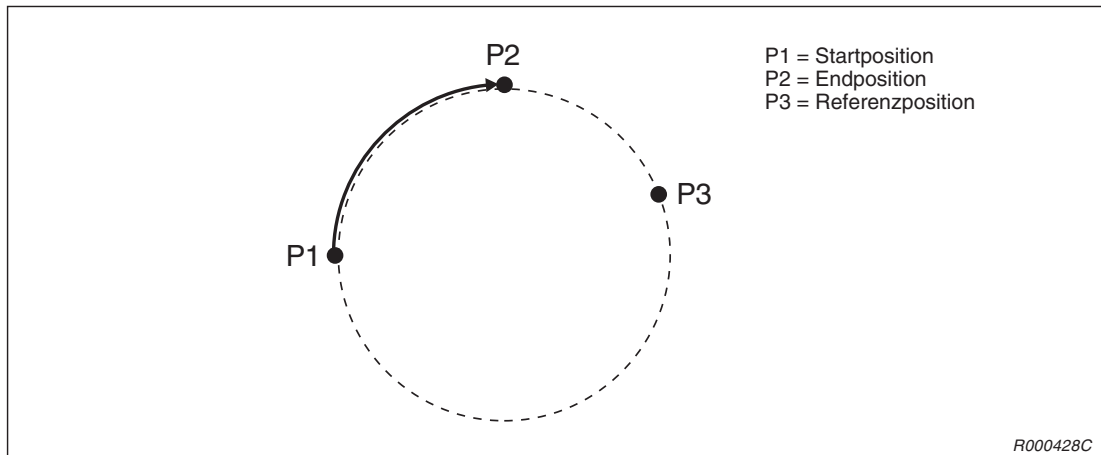


Abb. 9-3: Beispiel zur Kreis-Interpolation über einen Referenzpunkt

- Die Roboterstellung wird vom Startpunkt zum Endpunkt interpoliert. Die Stellung des Referenzpunktes hat keinen Einfluss.
- Entspricht die aktuelle Position nicht der Startposition, fährt der Roboter automatisch die Startposition mittels Linear-Interpolation an.
- Wird die Kreisbogenbewegung fortgesetzt, wenn die Kreis-Interpolation unterbrochen und nach einem JOG-Betrieb neu gestartet wurde, bewegt sich der Roboter mittels Gelenk-Interpolation zu der Stopp-Position und setzt dort die Kreisbogenbewegung fort.
- Der Roboter bewegt sich in die Richtung entlang des Kreisbogens, die nicht durch die Referenzposition geht.
- Weichen die Stellungsmerker der Start- und Endposition für eine andere Interpolationsmethode als die kartesische Interpolation für 3 Achsen voneinander ab, erfolgt eine Fehlermeldung.
- Der Roboter verfährt mit Linear-Interpolation, wenn zwei der drei Positionen gleich sind oder alle Positionen auf einer Geraden liegen. Es erfolgt keine Fehlermeldung.
- Wird über die numerische Konstante 2 die kartesische Interpolation für 3 Achsen gewählt, ist die numerische Konstante 1 unwirksam und der Roboter bewegt sich mit der geteachten Orientierung.
- Die numerische Konstante 2 legt den Interpolationstyp der Stellung fest. Bei einer Interpolation im Koordinatensystem X, Y, Z, J4, J5 und J6 wird die kartesische Interpolation für 3 Achsen verwendet, um den Roboter in die Nähe eines bestimmten Punktes zu bewegen.

Erläuterung

- Mittels Kreis-Interpolation bewegt sich die Roboterhand auf dem Kreisbogen, der durch die 3 Punkte festgelegt ist.

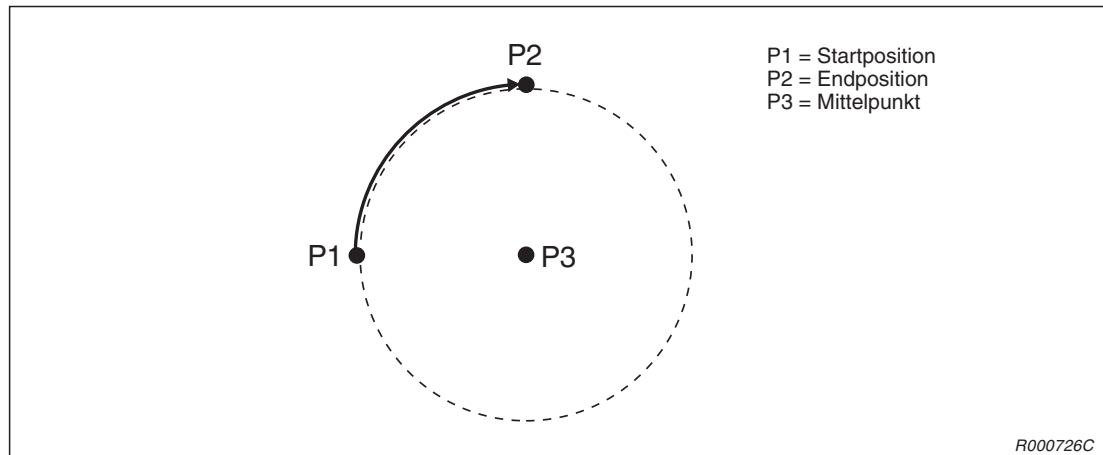


Abb. 9-4: Beispiel zur Kreis-Interpolation über einen Mittelpunkt

- Die Roboterstellung wird vom Startpunkt zum Endpunkt interpoliert. Die Stellung des Referenzpunktes hat keinen Einfluss.
- Entspricht die aktuelle Position nicht der Startposition, fährt der Roboter automatisch die Startposition mittels Linear-Interpolation an.
- Wird die Kreisbogenbewegung fortgesetzt, wenn die Kreis-Interpolation unterbrochen und nach einem JOG-Betrieb neu gestartet wurde, bewegt sich der Roboter mittels Gelenk-Interpolation zu der Stopp-Position und setzt dort die Kreisbogenbewegung fort.
- Weichen die Stellungsmerker der Start- und Endposition für eine andere Interpolationsmethode als die kartesische Interpolation für 3 Achsen voneinander ab, erfolgt eine Fehlermeldung.
- Der Roboter verfährt mit Linear-Interpolation, wenn zwei der drei Positionen gleich sind oder alle Positionen auf einer Geraden liegen. Es erfolgt keine Fehlermeldung.
- Wird über die numerische Konstante 2 die kartesische Interpolation für 3 Achsen gewählt, ist die numerische Konstante 1 unwirksam und der Roboter bewegt sich mit der geteachten Orientierung.
- Der Zentriwinkel vom Start- bis zum Endpunkt ist: $0 < \text{Zentriwinkel} < 180^\circ$.
- Legen Sie die Positionen so fest, dass die Differenz zwischen Mittelpunkt und Endpunkt und die Differenz zwischen Mittelpunkt und Startpunkt größer als 0,01 mm ist.
- Die numerische Konstante 2 legt den Interpolationstyp der Stellung fest. Bei einer Interpolation im Koordinatensystem X, Y, Z, J4, J5 und J6 wird die kartesische Interpolation für 3 Achsen verwendet, um den Roboter in die Nähe eines bestimmten Punktes zu bewegen.
- Stimmen der Start- und Endpunkt oder alle 3 Punkte überein, erfolgt keine Fehlermeldung. Der nächste Befehl wird ausgeführt. Ändert sich in dieser Zeit die Stellung wird nur die Stellung interpoliert.

9.3.46 MVS (Move S)

Funktion: geradlinige Bewegung

Bewegt die Handspitze mittels Linear-Interpolation zur festgesetzten Position.

Eingabeformat 1

```
MVS □ <Zielposition> [, <Abstand>]
      □ [<Interpolationstyp>] □ [<Verknüpfungsbedingung>]
```

Eingabeformat 2

```
MVS □ , <Verfahrbetrag> □ [<Interpolationstyp>]
```

- <Zielposition> Legt die Zielposition fest
- <Abstand> Legt den Verfahrbetrag in Werkzeugrichtung auf der Z-Achse fest (Abstand zur Zielposition/±-Richtung)
- <Interpolationstyp> Legt den Interpolationstyp fest:
TYPE <Numerische Konstante 1>, <Numerische Konstante 2>
Numerische Konstante 1 ... indirekte/direkte Anfahrt = 1/0
Numerische Konstante 2 ... 3 Achsen kartesisch/Drehung = 1/0
Standardwerte: 0, 0 (indirekte Anfahrt, Drehung)
- <Verknüpfungsbedingung> Es können die Verknüpfungen WTH und WTHIF verwendet werden.
- <Verfahrbetrag> Legt den Verfahrbetrag von der Augenblicksposition in Werkzeuglängsrichtung auf der Z-Achse fest (Abstand zur Zielposition)
Bei einem positiven Betrag bewegt sich die Handspitze in Werkzeuglängsrichtung nach vorne. Bei Angabe eines negativen Verfahrbetrags wird die Handspitze in Werkzeuglängsrichtung zurückgefahren.

Programmbeispiel

- 10 MVS PLT1,10,-100.0 WTH M_OUT(17) = 1 Position 100 mm über dem Gitterpunkt 10 der Palette 1 mittels Linear-Interpolation anfahren und Ausgangsbit 17 auf 1 setzen
- 20 MVS P4+P5,-50.0 WTHIF M_IN(18) = 1,M_OUT(20) = 1 Position anfahren, die um 50 mm in Werkzeuglängsrichtung entfernt von (P4+P5) liegt und Ausgangsbit 20 auf 1 setzen, falls Eingangsbit 18 gleich 1 ist
- 30 MVS ,-50 Position anfahren, die 50 mm in Werkzeuglängsrichtung von der aktuellen Position entfernt ist

Erläuterung

- Dieser Befehl verfährt die Handspitze entlang einer geraden Linie zur festgelegten Position.
- Die Roboterstellung wird vom Startpunkt zum Endpunkt interpoliert.
- Wird die geradlinige Bewegung fortgesetzt, wenn die Linear-Interpolation unterbrochen und nach einem JOG-Betrieb neu gestartet wurde, bewegt sich der Roboter mittels Gelenk-Interpolation zu der Stopp-Position und setzt dort die geradlinige Bewegung fort.
- Beginnt ein Neustart mit Befehlen, die sich auf die aktuelle Position beziehen (z. B. MVS, 100), bewegt sich die Handspitze um den verbleibenden Verfahrbetrag.
- Weichen die Stellungsmerker der Start- und Endposition für eine andere Interpolationsmethode als die kartesische Interpolation für 3 Achsen voneinander ab, erfolgt eine Fehlermeldung.
- Wird über die numerische Konstante 2 die kartesische Interpolation für 3 Achsen gewählt, ist die numerische Konstante 1 unwirksam und der Roboter bewegt sich mit der geteachten Orientierung.

9.3.47 OADL (Optimum Acceleration/Deceleration)

Funktion: Optimale Beschleunigung/Abbremsung

Legt die optimale Beschleunigungs-/Abbremszeit in Abhängigkeit von der Lasteinstellung der Hand fest.

Eingabeformat

OADL <Schalter>

<Schalter> Legt fest, ob die Einstellung für die optimale Beschleunigung/Abbremsung freigegeben (OADL ON) oder gesperrt (OADL OFF) ist.

Programmbeispiel

10	OADL ON	Gibt die optimale Beschleunigung/Abbremsung frei
20	MOV P1	Position 1 anfahren
30	LOADSET 1,1	Optimale Beschleunigung/Abbremsung für Hand 1 und Werkstück 1 einstellen
40	MOV P2	Position 2 anfahren
50	HOPEN 1	Öffnet Hand 1
60	MOV P3	Position 3 anfahren
70	HCLOSE 1	Schließt Hand 1
80	MOV P4	Position 4 anfahren
90	LOADSET 1,2	Optimale Beschleunigung/Abbremsung für Hand 1 und Werkstück 2 einstellen
100	MOV P5	Position 5 anfahren
110	HCLOSE 1	Umschaltung auf Werkstück 2 zur automatischen Veränderung der Beschl./Verzögerungsrampe
120	MOV P6	Position 6 anfahren

Parameter HNDHOLD ist auf 0, 1 gesetzt.

Erläuterung

- Der Roboter bewegt sich mit der optimalen Beschleunigung/Abbremsung für die über den LOADSET-Befehl eingestellten Bedingungen für die Hand und das Werkstück.
- Die Zuordnungen für das Öffnen/Schließen einer Hand und den Befehlen HOPEN oder HCLOSE erfolgen über die Parameter HNDHOLD 1 bis 8.
- Die OADL-Einstellung ist standardmäßig deaktiviert.
- Der OADL-Befehl ist so lange aktiv (OADL ON), bis er auf OFF gesetzt wird (OADL OFF) oder bis die END-Anweisung ausgeführt wird.

9.3.48 ON COM GOSUB (ON Communication Go Subroutine)

Funktion: Sprung zu einem Unterprogramm

Legt den Sprung in ein Unterprogramm fest, wenn ein Interrupt von einem Kommunikationskanal anliegt.

Eingabeformat

ON <input type="checkbox"/> COM <input type="checkbox"/> [(<input type="checkbox"/> <Dateinummer>)] <input type="checkbox"/> GOSUB <input type="checkbox"/> <Sprungziel>

<Dateinummer> Legt die Nummer des Kommunikationskanals
($1 \leq \text{Kommunikationskanal} \leq 2$) fest

<Sprungziel> Legt eine Zeilennummer oder eine Marke fest

Programmbeispiel

10	OPEN "COM1:" AS #1	Öffnet Kommunikationskanal 1 als Datei Nr. 1
20	OPEN "COM3:" AS #2	Öffnet Kommunikationskanal 3 als Datei Nr. 2
30	ON COM GOSUB 300	Springt zur Zeile 300, wenn auf dem Kommunikationskanal Nummer 1 ein Interrupt anliegt
40	ON COM(2) GOSUB *RECV	Springt zu Marke *RECV, wenn auf dem Kommunikationskanal Nummer 3 ein Interrupt anliegt
50	COM(1) ON	Gibt den Kommunikations-Interrupt der Datei Nr. 1 frei
60	COM(2) ON	Gibt den Kommunikations-Interrupt der Datei Nr. 2 frei
70	COM OFF	Sperrt den Kommunikations-Interrupt der Datei Nr. 1
80	COM(2) OFF	Sperrt den Kommunikations-Interrupt der Datei Nr. 2

Erläuterung

- Bei fehlender Nummer des Kommunikationskanals wird der Standardwert 1 gesetzt.
- Die Prioritäten der Interrupts werden mit steigender Dateinummer kleiner.
- Bei anliegendem Interrupt wird die Roboterbewegung gestoppt.
- Mit der COM STOP-Anweisung kann der Interrupt ignoriert werden und die Roboterbewegung wird nicht unterbrochen.
- In der Grundeinstellung sind die Interrupts gesperrt. Mit dem COM ON-Befehl wird ein Interrupt wieder zugelassen.

9.3.49 ON GOSUB (ON GOSUB)

Funktion: Sprung zu einem Unterprogramm

Legt den Sprung zu einer festgelegten Zeilennummer oder einer Marke eines Unterprogramms fest.

Eingabeformat

```
ON  <Ausdruck>  GOSUB  [<Sprungziel>] [, [<Sprungziel>]] ...
```

<Ausdruck> Legt fest, zu welcher Zeilennummer oder Marke das Programm verzweigt wird

<Sprungziel> Legt eine Zeilennummer oder Marke fest

Programmbeispiel

```
10 ON M1 GOSUB 1000, *SUBPR      Springt zur Zeile 1000, wenn der Wert
                                   des Ausdrucks 1 ist und zur Marke *SUBPR,
                                   wenn der Wert 2 ist
```

Erläuterung

- Der Wert des Ausdrucks legt fest, zu welcher Zeilennummer oder Marke das Programm verzweigt wird.
Beispiel: Ist der Wert der Variablen 2, wird das zweite Sprungziel aufgerufen.
- Wird eine Zeilennummer oder Marke aufgerufen, die nicht existiert oder zweimal definiert ist, erfolgt eine Fehlermeldung.

Wert von <Ausdruck>	Steuerung
Reelle Zahl	Der Wert wird zu einer Integer-Zahl gerundet.
Wenn der Wert des Ausdrucks gleich 0 ist oder wenn der Wert größer als die Anzahl der Zeilen oder Marken ist	Steuerung springt in die nächste Zeile.
Wenn der Wert negativ oder größer als 32767 ist	ERROR
Zeilennummer oder Marke ist nicht angegeben	ERROR

Tab. 9-3: Werte des Ausdrucks und deren Verarbeitung

9.3.50 ON ... GOTO (On Go To)

Funktion: Programmverzweigung

Legt den Sprung zu einer festgelegten Zeilennummer oder einer Marke fest.

Eingabeformat

```
ON  <Ausdruck>  GOTO  [<Sprungziel>] [, [<Sprungziel>]] ...
```

<Ausdruck> Legt fest, zu welcher Zeilennummer oder Marke das Programm verzweigt wird

<Sprungziel> Legt eine Zeilennummer oder Marke fest

Programmbeispiel

```
10 ON M1 GOTO 1000, *SUBPR     Springt zur Zeile 1000, wenn der Wert
                               des Ausdrucks 1 ist und zur Marke *SUBPR,
                               wenn der Wert 2 ist
```

Erläuterung

- Der Wert des Ausdrucks legt fest, zu welcher Zeilennummer oder Marke das Programm verzweigt wird.
Beispiel: Ist der Wert des Ausdrucks 2, wird das zweite Sprungziel aufgerufen.
- Gibt der Wert des Ausdrucks ein nicht vorhandenes Sprungziel an, erfolgt eine Fehlermeldung.

Wert von <Ausdruck>	Steuerung
Reelle Zahl	Der Wert wird zu einer Integer-Zahl gerundet.
Wenn der Wert des Ausdrucks gleich 0 ist oder wenn der Wert größer als die Anzahl der Zeilen oder Marken ist	Steuerung springt in die nächste Zeile.
Wenn der Wert negativ oder größer als 32767 ist	ERROR
Zeilennummer oder Marke ist nicht angegeben	ERROR

Tab. 9-4: Werte des Ausdrucks und deren Verarbeitung

9.3.51 OPEN (Open)

Funktion: Datei öffnen

Öffnet eine Datei.

Eingabeformat

```
OPEN  "<Dateibezeichnung>"  [FOR <Modus>] 
AS  [#]<Dateinummer>
```

<Dateibezeichnung> Gibt den Namen der Datei oder des Kommunikationskanals an
 Die Dateibezeichnung "<Dateiname des Kommunikationskanals>:" wird zum Öffnen von Kommunikationskanälen verwendet.
 Die Dateibezeichnung "<Dateiname>" wird zum Öffnen anderer Dateien verwendet.

<Modus> Legt die Methode fest, mit der auf eine Datei zugegriffen wird
 Die Angabe kann zum Öffnen von Kommunikationskanälen weggelassen werden.

- Keine Angabe = wahlfreier Modus
 Dieser Modus wird beim Zugriff auf die Kommunikationskanäle verwendet.
- INPUT = Eingabemodus
 Liest Daten von einer vorhandenen Datei ein
- OUTPUT = Ausgabemodus (neue Datei)
 Legt eine neue Datei an und schreibt Daten in diese Datei
- APPEND = Ausgabemodus (vorhandene Datei)
 Hängt Daten an das Ende einer vorhandenen Datei an

Dateibezeichnung	Dateiname	Zugriffsmethode
Dateiname	Maximal 16 Zeichen	INPUT, PRINT, APPEND
Kommunikationskanal	COM 1 :, COM 2 :, COM 3 :	Keine Angabe = Wahlfreier Modus

Tab. 9-5: Dateibezeichnung und Zugriffsmethode

<Dateinummer> Legt eine Konstante im folgenden Bereich fest:
 für einen Interrupt eines Kommunikationskanals: 1 bis 3,
 ohne Interrupt eines Kommunikationskanals: 1 bis 8

Programmbeispiel

10	OPEN "COM1:" AS #1	Öffnet den RS232C-Kommunikationskanal als Datei Nr. 1
20	MOV P_01	Position P_01 anfahren
30	PRINT #1,P_CURR	Sendet die Daten der aktuellen Position an einen an der RS232 angeschlossenen Empfänger, z. B. Terminal, SPS ... Format: (100.00,200.00,300.00,400.00)(7.0)
40	INPUT #1,M1,M2,M3	Liest die Daten „101.00,202.00,303.00“ im ASCII-Format ein und schreibt sie in die Variablen M1, M2 und M3
50	P_01.X = M1	Schreibt die Daten aus M1 in die X-Komponente der globalen Variablen P_01
60	P_01.Y = M2	Schreibt die Daten aus M2 in die Y-Komponente der globalen Variablen P_01
70	P_01.C = M3	Schreibt die Daten aus M3 in die C-Komponente der globalen Variablen P_01
80	CLOSE	Schließt alle geöffneten Dateien
90	END	Programmende

Erläuterung

- Die Datei wird über die Dateibezeichnung geöffnet und es wird eine Dateinummer festgelegt. Schreib- oder Lesevorgänge werden einer Datei durch Angabe der Dateinummer zugeordnet.
- Ein Kommunikationskanal wird wie eine Datei behandelt.

Kommunikationskanal	Schnittstelle
COM 1	Standard RS232C
COM 2	Reserviert
COM 3	Reserviert

Tab. 9-6: Zuordnung der Schnittstellen

9.3.52 OVRD (Override)

Funktion: Übersteuerung

Legt den Programmwert für die Geschwindigkeits-Übersteuerung fest.

Eingabeformat

```
OVRD  <Übersteuerungswert>
```

<Übersteuerungswert> Legt den prozentualen Übersteuerungswert fest
 $1 \leq \text{Übersteuerungswert} \leq 100.0$
 Bei einer Einstellung eines Wertes außerhalb des Einstellbereiches erfolgt eine Fehlermeldung.

Programmbeispiel

10	OVRD 50	Übersteuerung auf den Wert 50 % einstellen
20	MOV P1	Position P1 mittels Gelenk-Interpolation anfahren
30	MVS P2	Position P2 mittels Linear-Interpolation anfahren
40	OVRD M_NOVRD	Standardwert einstellen

Erläuterung

- Legt den prozentualen Übersteuerungswert für die Arbeitsgeschwindigkeit des Roboters fest.
- Der OVRD-Befehl ist unabhängig von der Art der Interpolation wirksam.
- Die aktuelle Arbeitsgeschwindigkeit ergibt sich folgendermaßen:

Gelenk-Interpolation	=	Einstellung der T/B oder des Steuergeräts	x	Einstellwert des OVRD-Befehls	x	Einstellwert des JOVRD-Befehls
Linear-Interpolation	=	Einstellung der T/B oder des Steuergeräts	x	Einstellwert des OVRD-Befehls	x	Einstellwert des SPD-Befehls

- Der Maximalwert der Arbeitsgeschwindigkeit ist 100 %. Der Standardwert der Arbeitsgeschwindigkeit beträgt 100 % der Standardeinstellung (M_NOVRD).
- Der Standardwert bleibt so lange wirksam, bis der OVRD-Befehl ausgeführt wird. Die so festgesetzte Arbeitsgeschwindigkeit kann durch einen weiteren OVRD-Befehl geändert werden.
- Die durch einen OVRD-Befehl festgelegte Arbeitsgeschwindigkeit bleibt so lange erhalten, bis erneut ein OVRD-Befehl, eine END-Anweisung oder ein Reset ausgeführt wird. Nach Ausführung der END-Anweisung oder eines Resets ist der Standardwert wieder gültig.
- Liegt der Übersteuerungswert außerhalb des Wertebereiches des Roboters, erfolgt eine Fehlermeldung. Der Wert muss zwischen 0 und 100 % liegen.

9.3.53 PLT (Pallet)

Funktion: Koordinaten für Palette berechnen

Berechnet die Koordinaten eines Gitterpunktes der festgelegten Palette und weist die berechneten Koordinaten der festgelegten Position zu.

Eingabeformat

```
PLT □ <Palettennummer>, <numerischer Operationsausdruck>
```

<Palettennummer>	Wählt eine vorher mit dem DEF PLT-Befehl definierte Palette aus $1 \leq \text{Palettennummer} \leq 8$
<numerischer Operationsausdruck>	Legt die Positionsnummer für die berechneten Koordinaten fest

Programmbeispiel

10 M1 = 0	Setzt M1 auf 0
20 DEF PLT 1,P10,P11,P12,P13,8,8,1	Definiert Palette Nummer 1
30 M1 = M1 + 1	Erhöht den Wert von M1 um 1
40 P1=PLT 1,M1	Weist P1 die Koordinaten des Gitterpunktes M1 zu
50 MOV P1,-50	Position anfahren, die um 50 mm in Werkzeugzeuglängsrichtung von Position 1 entfernt liegt
60 MVS P1	Position 1 mittels Linear-Interpolation anfahren
70 MOV (PLT1,5) TYPE 1,1	Position 5 anfahren

Erläuterung

- Dieser Befehl berechnet die Koordinaten eines Gitterpunktes einer Palette, die vorher mit dem DEF PLT-Befehl definiert wurde, und weist sie einer Position zu.
- Die Palettennummern müssen im Bereich von 1 bis 8 liegen. Es können bis zu acht Paletten gleichzeitig definiert sein.
- Die Position des Gitterpunktes kann in Abhängigkeit der festgelegten Bewegungsrichtung (siehe DEF PLT-Befehl) unterschiedlich sein.
- Wird ein Gitterpunkt festgelegt, der außerhalb der Zeilen oder Spalten der definierten Palette liegt, erfolgt eine Fehlermeldung.
- Ist ein Palettengitterpunkt, der in einem Bewegungsbefehl als Zielposition angegeben ist, nicht in Klammern aufgeführt, erfolgt eine Fehlermeldung (siehe Programmbeispiel).

9.3.54 PRINT (Print)

Funktion: Daten übertragen

Überträgt Daten in eine Datei oder an ein Ausgabegerät.

Eingabeformat

```
PRINT □ #<Dateinummer> □ [, [<Ausdruck>;] ... [<Ausdruck> [ ; ]]
```

- <Dateinummer> Bezieht sich auf die im OPEN-Befehl festgelegte Dateinummer
1 ≤ Dateinummer ≤ 8
- <Ausdruck> Legt eine numerische Variable, eine Positionsvariable
oder eine Zeichenkette fest

Programmbeispiel

10	OPEN "COM1" AS #1	Öffnet den RS232C-Kommunikationskanal als Datei Nr. 1
20	MDATA = 150	Setzt MDATA auf 150
30	PRINT #1, "***PRINT TEST***"	Gibt die Zeichenkette ***PRINT TEST*** aus
40	PRINT #1	Gibt eine Leerzeile aus
50	PRINT #1, "MDATA =", MDATA	Gibt die Zeichenkette MDATA = und den Wert von MDATA aus, (150)
60	PRINT #1	Gibt eine Leerzeile aus
70	PRINT #1, "*****"	Gibt die Zeichenkette ***** aus
80	END	Programmende

Folgendes Ergebnis wird ausgegeben:

PRINT TEST

MDATA = 150

Erläuterung

- Fehlt eine Angabe für <Ausdruck>, wird ein „Carriage Return“ ausgegeben.
- Bei fehlender <Dateinummer> wird der Standardwert 1 verwendet.
- Ausgabeformat der Daten:
Der Platz für die Ausgabe von <Ausdruck> ist in Einheiten von 10 festgelegt. Werden bei der Ausgabe mehrere Ausdrücke angegeben, muss ein Komma zwischen den einzelnen Ausdrücken stehen.
Bei Trennung der Ausdrücke durch Semikolons werden sie ohne Zwischenraum ausgegeben.
- Nach jeder PRINT-Anweisung wird ein „Carriage Return“ ausgeführt.

Beispiel ▾

```
10 M1 = 123.5
20 P1 = (130.5,-117.2,55.1,16.2,0.0,0.0)(1,0)
```

nach Eingabe von

```
30 PRINT #1,"OUTPUT TEST",M1,P1 wird
```

```
OUTPUT TEST      123.5      (130.5,-117.2,55.1,16.2,0.0,0.0)(1,0) ausgegeben
```

nach Eingabe von

```
30 PRINT #1,"OUTPUT TEST";M1;P1 wird
```

```
OUTPUT TEST 123.5(130.5,-117.2,55.1,16.2,0.0,0.0)(1,0) ausgegeben
```

Werden die Ausdrücke durch ein Komma oder ein Semikolon getrennt, wird kein „Carriage Return“ zugelassen. Die Ausdrücke werden in einer Zeile ausgegeben. Wird kein Komma oder Semikolon eingegeben, wird ein „Carriage Return“ zugelassen.

Nach Eingabe von

```
30 PRINT #1,"OUTPUT TEST",
```

```
40 PRINT #1,M1;
```

```
50 PRINT #1,P1 wird
```

```
OUTPUT TEST 123.5(130.5,-117.2,55.1,16.2,0.0,0.0)(1,0) ausgegeben.
```

△

9.3.55 RELM (Release Mechanism)

Funktion: Roboterzuordnung aufheben

Hebt die Zuordnung eines Roboters auf.

Eingabeformat

RELM

Erläuterung

- Hebt die aktuelle Zurordnung eines Roboters auf.
- Bei einem Programmstopp durch ein Interrupt-Signal wird der Befehl RELM automatisch durch das System ausgeführt.

9.3.56 REM (Remarks)

Funktion: Kommentar

Ermöglicht dem Programmierer, einen Kommentar zu schreiben.

Eingabeformat

```
REM □ [<Kommentar>]
```

<Kommentar>

Es können Zeichenketten bis zur Länge einer Zeile eingegeben werden.

Programmbeispiel

10	REM ***Hauptprogramm***	Legt die Zeichenkette ***Hauptprogramm***
20	‘***Hauptprogramm***	als Kommentar fest
30	MOV P1	Position 1 mittels Gelenk-Interpolation anfahren

Erläuterung

- Die REM-Anweisung kann durch ein halbes Anführungszeichen (‘) abgekürzt werden.
- Ein Kommentar kann hinter einem Befehl in derselben Zeile stehen.

9.3.57 RETURN (Return)

Funktion: Rücksprung zum Hauptprogramm

Springt beim Rücksprung aus einem Unterprogramm in die Zeile nach dem GOSUB-Befehl.

Springt beim Rücksprung aus einer Interrupt-Routine in die Zeile zurück, in der der Interrupt aufgetreten ist oder in die nächste Zeile.

Eingabeformat

Beim Rücksprung aus einem Unterprogramm:

```
RETURN
```

Beim Rücksprung aus einer Interrupt-Routine:

```
RETURN <Rücksprungziel>
```

<Rücksprungziel> Legt die Zeile fest, zu der die Steuerung zurückspringt, nachdem eine Interrupt-Routine abgearbeitet wurde
 0 ... Springt in die Zeile, in der der Interrupt aufgetreten ist.
 1 ... Springt eine Zeile hinter die Zeile, in der der Interrupt aufgetreten ist.

Erläuterung

- Es erfolgt eine Fehlermeldung, wenn bei einem RETURN-Befehl in einem Unterprogramm ein Rücksprungziel angegeben wurde. Es erfolgt eine Fehlermeldung, wenn das Rücksprungziel in einer Interrupt-Routine nicht angegeben wurde.

9.3.58 SELECT CASE

Funktion: Prozess ausführen

Führt in Abhängigkeit einer Bedingung einen von mehreren Prozessen aus.

Eingabeformat

```
SELECT □ <Auswahl>
  CASE □ <Ausdruck>
    [<Prozess>]
    BREAK
  CASE □ <Ausdruck>
    [<Prozess>]
    BREAK

    :

  CASE □ <Ausdruck>
    [<Prozess>]
    BREAK
  DEFAULT
    [<Prozess>]
    BREAK
END □ SELECT
```

- | | |
|------------|--------------------------------------------------------------------------------------------------------------------------|
| <Auswahl> | Legt einen numerischen Ausdruck fest |
| <Ausdruck> | Legt einen numerischen Ausdruck fest
Der Typ muss mit dem der Bedingung übereinstimmen. |
| <Prozess> | Legt die auszuführende Anweisung (inklusive Verzweigungen
oder Wiederholschleifen), eine Zeilennummer oder Marke fest |

Programmbeispiel

10	SELECT MCNT	Auswahl der numerischen Variablen MCNT
20	M1 = 10	Diese Zeile wird nicht ausgeführt.
30	CASE IS <= 10	Fahre Position P2 an, falls CASE kleiner gleich 10 ist
40	MOV P1	
50	BREAK	Sprung hinter die END SELECT-Anweisung
60	CASE 11	Fahre Position P2 an, falls CASE gleich 11 ist
70	MOV P2	
80	BREAK	Sprung hinter die END SELECT-Anweisung
90	CASE 12	Fahre Position P3 an, falls CASE gleich 12 ist
100	MOV P3	
110	BREAK	Sprung hinter die END SELECT-Anweisung
120	CASE 13 TO 18	Fahre Position P4 an, falls CASE größer gleich 13 oder kleiner gleich 18 ist
130	MOV P4	
140	BREAK	Sprung hinter die END SELECT-Anweisung
150	DEFAULT	Setzt Ausgangsbit 10 auf „1“, falls MCNT keinem der oben genannten Werte oder Wertebereiche entspricht
160	M_OUT(10) = 1	
170	BREAK	Sprung hinter die END SELECT-Anweisung
180	END SELECT	

Erläuterung

- Wird eine der Bedingungen der CASE-Anweisung erfüllt, wird der Prozess bis zur nächsten CASE-, DEFAULT- oder ENDSELECT-Anweisung ausgeführt.
- Wird keine der CASE-Bedingungen erfüllt, wird der DEFAULT-Prozess ausgeführt. Ist kein DEFAULT-Prozess definiert, springt das Programm eine Zeile hinter die END SELECT-Anweisung.
- Eine SELECT-Anweisung muss immer durch eine END SELECT-Anweisung abgeschlossen werden.
- Bei Ausführung einer END SELECT-Anweisung, der keine SELECT-Anweisung vorausgeht, erfolgt eine Fehlermeldung. Wird eine END-Anweisung zwischen den beiden Anweisungen oder die letzte Programmzeile ausgeführt, stoppt das Programm in dieser Zeile .
- Innerhalb eines SELECT-CASE-Blocks können keine weiteren SELECT-CASE-Blöcke ausgeführt werden. Die Ausführung von WHILE WEND- oder FOR-NEXT-Schleifen ist jedoch möglich.

9.3.59 SERVO (Servo)

Funktion: Servo ein-/auschalten

Schaltet die Servospannung ein oder aus.

Eingabeformat

SERVO <input type="checkbox"/> ON

SERVO <input type="checkbox"/> OFF

Programmbeispiel

10	SERVO ON	Schaltet die Servospannung ein
20	IF M_SVO <> 1 GOTO 20	Wartestatus, bis Servoversorgung eingeschaltet ist
30	SPD M_NSPD	Geschwindigkeit auf Standardwert setzen

Erläuterung

- Die Servospannung für alle Achsen wird ein- oder ausgeschaltet.

9.3.60 SKIP (Skip)

Funktion: Sprung in die nächste Zeile

Die Programmsteuerung springt in die nächste Zeile.

Eingabeformat

SKIP

Programmbeispiel

20 MOV P1 WTHIF M_IN(17) = 0,SKIP

Fährt Position 1 mittels Gelenk-Interpolation an und unterbricht die Roboterbewegung, wenn das Eingangsbit Nummer 17 gleich 0 wird
Die Programmsteuerung springt in die nächste Zeile.

Erläuterung

- Dieser Befehl wird mit anderen Befehlen in Verbindung mit WTH und WTHIF verwendet. Bei Ausführung des Befehls in Verbindung mit WTH oder WTHIF wird die Programmabarbeitung innerhalb der Zeile unterbrochen und die Programmsteuerung springt in die nächste Zeile. Die Ausführung einer SKIP-Anweisung kann über die Roboterstatusvariable M_SKIPCQ geprüft werden.

9.3.61 SPD (Speed)

Funktion: Geschwindigkeit festlegen

Legt die Geschwindigkeit für lineare und kreisförmige Bewegungen fest.

Eingabeformat

```
SPD □ <Geschwindigkeitswert>
```

<Geschwindigkeitswert>

Legt die Geschwindigkeit als reelle Zahl in mm/s fest

Programmbeispiel

10	SPD 100	Legt die Geschwindigkeit auf 100 mm/s fest
20	MVS P1	Fährt Position P1 mittels Linear-Interpolation an
30	SPD M_NS PD	Setzt die Geschwindigkeit auf den Standardwert

Erläuterung

- Der SPD-Befehl ist nur bei linearen und kreisförmigen Bewegungen des Roboters wirksam.
- Der aktuelle Übersteuerungswert ergibt sich aus:

Aktueller Übersteuerungswert	=	Übersteuerungswert der T/B oder des Steuergeräts	x	Einstellwert des OVRD-Befehls	x	Einstellwert des SPD-Befehls
------------------------------	---	--------------------------------------------------	---	-------------------------------	---	------------------------------

- Bei Verwendung des Standardwerts verfährt der Roboter immer mit der maximal möglichen Geschwindigkeit.
- Der Standardwert ist so lange gültig, bis mit dem SPD-Befehl ein neuer Wert festgelegt wird. Dieser kann durch den SPD-Befehl wieder geändert werden.
- Die über den SPD-Befehl festgelegte Geschwindigkeit wird bei der Ausführung der END-Anweisung auf den Standardwert zurückgesetzt.

9.3.62 TOOL (Tool)

Funktion: Werkzeug-Konvertierungsdaten

Legt die Werkzeug-Konvertierungsdaten fest (Verschiebung des TCP).

Eingabeformat

```
TOOL □ <Werkzeug-Konvertierungsdaten>
```

Programmbeispiel

```
10 TOOL (100,100,0,0,0,0)    Legt die Werkzeug-Konvertierungsdaten fest
20 MVS P2                    Position 2 anfahren
30 TOOL P_NTOOL              Setzt Werkzeug-Konvertierungsdaten auf den
                              Standardwert P_NTOOL
```

Erläuterung

- Es wird der Standardwert (P_NTOOL) verwendet, bis ein TOOL-Befehl ausgeführt wird. Ist der TOOL-Befehl ausgeführt, sind die Werkzeug-Konvertierungsdaten so lange gültig, bis der TOOL-Befehl erneut ausgeführt wird.
- Die mit dem TOOL-Befehl festgelegten Werkzeug-Konvertierungsdaten werden im Parameter MEXTL gespeichert. Der Wert bleibt auch nach Ausschalten der Spannungsversorgung des Steuergeräts erhalten.

Beispiel ▾

```
TOOL (0,65,145,0,0,0)
```

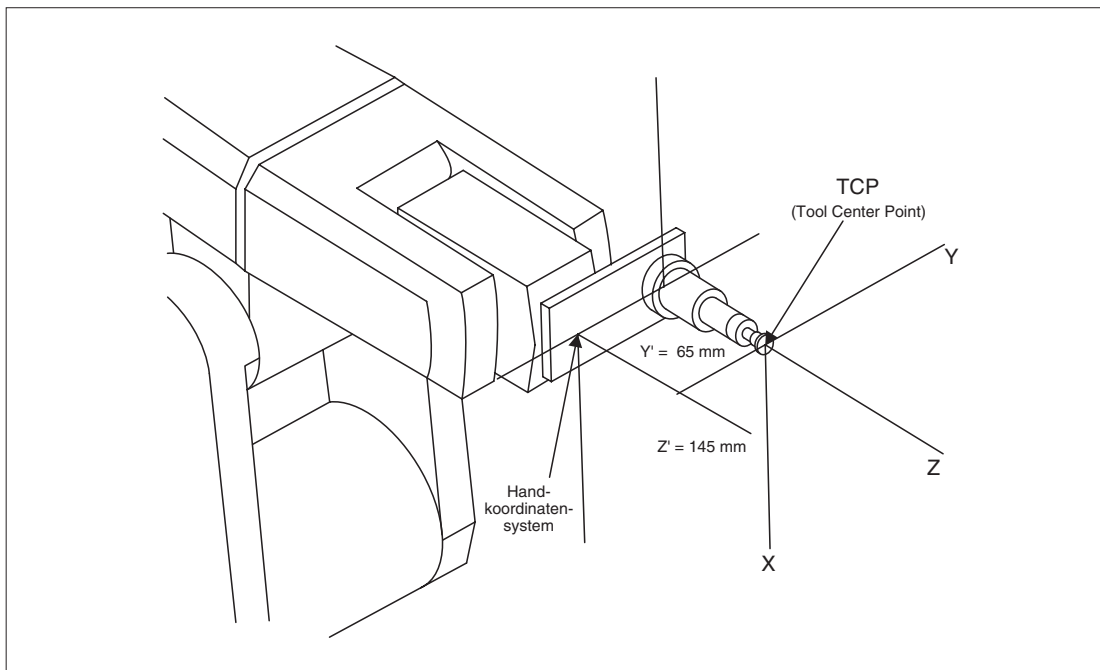


Abb. 9-5: Übergang vom Handkoordinatensystem zum Werkzeugkoordinatensystem TCP

9.3.63 TORQ (Torque)

Funktion: Drehmomentgrenze definieren

Legt die Drehmomentgrenze für eine Achse fest.

Eingabeformat

```
TORQ □ <Achsennummer>, <numerischer Wert>
```

<Achsennummer>	Legt die Nummer der Achse fest $1 \leq \text{Achsennummer} \leq 8$
<numerischer Wert>	Legt den prozentualen Grenzwert des Drehmoments, der von einer Achse erzeugt wird, fest $1 \leq \text{Grenzwert} \leq 100 \%$

Programmbeispiel

10 TORQ 4, 80	Legt das maximale Drehmoment für Achse 4 auf 80 % fest
20 MVS P1	Position 1 anfahren

Erläuterung

- Es wird das maximale Drehmoment für eine Achse festgelegt. Der Grenzwert wird in Prozent, bezogen auf den Standardwert, eingestellt.
- Die Drehmomentgrenze wird verwendet, wenn z. B. beim Greifen mit einer Servo-Greifhand eine bestimmte Greifkraft nicht überschritten werden darf.

9.3.64 WAIT (Wait)

Funktion: Wartestatus definieren

Legt einen Wartestatus in Abhängigkeit von einer Variablen fest.

Eingabeformat

```
WAIT □ <numerische Variable>=<numerische Konstante>
```

<p><numerische Variable></p> <p><numerische Konstante></p>	<p>Legt eine mit M beginnende Variable fest</p> <p>Legt eine numerische Konstante fest</p>
------------------------------------------------------------------------	--------------------------------------------------------------------------------------------

Programmbeispiel

10	WAIT M_IN(1) = 1	Wartestatus bis Eingangsbit 1 gleich 1 ist
20	WAIT M_IN(3) = 0	Wartestatus bis Eingangsbit 3 gleich 0 ist
30	WAIT M_RUN(2) = 1	Wartestatus bis Betriebssignal der Anwendung 2 gleich 1 ist
40	WAIT M_01 = 1	Wartestatus bis die externe Variable M_01 gleich 1 ist

Erläuterung

- Der Befehl wird zur Unterbrechung eines Programms bis zu einer Signaleingabe und während des Multitaskings verwendet.

9.3.65 WHILE ~ WEND (While End)

Funktion: Programmschleife

Solange die Schleifenbedingung wahr ist, wird das Programm zwischen der WHILE- und der WEND-Anweisung wiederholt.

Eingabeformat

```
WHILE □ <Schleifenbedingung>
:
WEND
```

<Schleifenbedingung> Legt die Abarbeitung der Schleife über eine Vergleichsbedingung fest

Programmbeispiel

20	WHILE (M1 >= -5) AND (M1 <= 5)	Wiederholt den Programmblock, solange M1 zwischen -5 und +5 liegt und springt zu Zeile 60, wenn M1 außerhalb des Wertebereichs liegt
30	M1 = -(M1 + 1)	Addiert 1 zu M1 und kehrt das Vorzeichen um
40	PRINT #1,M1	Gibt den Wert von M1 aus
50	WEND	Springt zurück zur WHILE-Anweisung (Zeile 20)
60	END	Programmende

Erläuterung

- Der Programmblock zwischen WHILE und WEND wird wiederholt, solange die Schleifenbedingung wahr (M1 zwischen -5 bis +5) ist.
- Ist die Schleifenbedingung unwahr (M1 außerhalb von -5 bis +5), springt das Programm eine Zeile hinter die WEND-Anweisung.

9.3.66 WTH (With)

Funktion: Anweisung hinzufügen

Während einer Interpolationsbewegung wird eine zusätzliche Anweisung ausgeführt.

Eingabeformat

```
WTH □ <Anweisung>
```

<Anweisung> Legt die zusätzlich ausgeführte Anweisung fest
 Es dürfen folgende Operationen ausgeführt werden:
 – <num. Datentyp B><Substitutionsoperator><num. Datentyp A>
 [Substitutionen, Signal-Anweisungen (siehe entsprechendes
 Syntaxdiagramm)]
 – HLT-Anweisung
 – SKIP-Anweisung.

Programmbeispiel

```
10  MOV P1 WTH M_OUT(17) = 1 DLY M1 + 2 Position P1 anfahren und Ausgangs-
                                     bit 17 für die Zeit von (M1+ 2)
                                     Sekunden auf 1 setzen
```

Erläuterung

- Dieser Befehl wird dazu verwendet, während einer Interpolationsbewegung eine zusätzliche Anweisung auszuführen.
- Es erfolgt eine Fehlermeldung, wenn die Anweisung nicht angegeben wird.
- Die Anweisung wird mit Beginn der Roboterbewegung ausgeführt.
- Die Prioritäten der Interrupts sind:
 COM > ACT > WTHIF (WTH) > Impulsausgang.

9.3.67 WTHIF (With If)

Funktion: Anweisung hinzufügen, wenn ...

Während einer Interpolationsbewegung wird eine bedingte, zusätzliche Anweisung ausgeführt.

Eingabeformat

WTHIF □ <Bedingung>, <Anweisung>

<Bedingung>	Legt die Bedingung fest, bei der die zusätzliche Anweisung ausgeführt wird (siehe auch ACT)
<Anweisung>	Legt die zusätzlich ausgeführte Anweisung fest (siehe auch WTH) Es dürfen folgende Operationen ausgeführt werden: <ul style="list-style-type: none"> – <num. Datentyp B><Substitutionsoperator><num. Datentyp A> – HLT-Anweisung – SKIP-Anweisung.

Programmbeispiel

10	MOV P1 WTHIF M_IN(17) = 1,HLT	Position 1 anfahren und Programm stoppen, falls das Eingangsbit Nummer 17 gleich 1 ist
20	MVS P2 WTHIF M_RSPD>200,M_OUT(17) = 1 DLY M1 + 2	Position 2 anfahren und das Ausgangsbit Nummer 17 für die Zeit von (M1 + 2) Sekunden auf 1 setzen, falls M_RSPD > 200
30	MVS P3 WTHIF M_MRATIO>15,M_OUT(1) = 1	Position 3 anfahren und das Ausgangsbit Nummer 1 auf 1 setzen, falls M_MRATIO > 15

Erläuterung

- Dieser Befehl wird dazu verwendet, während einer Interpolationsbewegung eine zusätzliche, bedingte Anweisung auszuführen.
- Die Anweisung wird mit Beginn der Roboterbewegung ausgeführt.

9.3.68 XLOAD (X Load)

Funktion: Programm laden

Lädt ein Programm in einen festgelegten Programmplatz (Slot/Task).

Eingabeformat

```
XLOAD □ <Programmplatznummer> <Programmname>
```

<Programmplatznummer> Legt die Nummer des Programmplatzes fest
 <Programmname> Legt den Namen des Programms fest

Programmbeispiel

10	XLOAD 2,"10"	Auswahl des Programms 10 für Programmplatz 2
20	XRUN 2	Startet Programmplatz 2
30	WAIT M_RUN(2) = 1	Wartestatus bis das Betriebssignal des Programmplatzes 2 gleich 1 ist

Erläuterung

- Ist das gewählte Programm bereits einem anderen Programmplatz zugewiesen, erfolgt bei der Programmausführung eine Fehlermeldung.
- Ist das gewählte Programm editiert worden, erfolgt bei der Programmausführung eine Fehlermeldung.
- Die Festlegung des Programmnamens erfolgt in Anführungszeichen.

9.3.69 XRUN (X Run)

Funktion: Programm starten

Startet die parallele Ausführung der gewählten Programme.

Eingabeformat

```
XRUN □ <Programmplatznummer> <Programmname> [, <Ausführung>]
```

<Programmplatznummer>	Legt die Nummer des Programmplatzes fest
<Programmname>	Legt den Namen des Programms fest
<Ausführung>	Legt die Ausführung des Programms fest 0 = kontinuierlich, 1 = zyklisch

Programmbeispiel

10	XRUN 1,"1"	Startet Programm 1 als Programmplatz 1
20	XRUN 3,"2",1	Startet Programm 2 als Programmplatz 3 im zyklischen Betrieb

Erläuterung

- Wird der gewählte Programmplatz bereits verwendet, erfolgt bei der Programmausführung eine Fehlermeldung.
- Wird der XRUN-Befehl im Wartestatus eines in der Programmmitte gestoppten Programms ausgeführt, startet der Betrieb im kontinuierlichen Modus.
- Die Festlegung des Programmnamens erfolgt in Anführungszeichen.
- Fehlt die Angabe für die Ausführung, wird das Programm im aktuellen Modus ausgeführt.

9.3.70 XSTP (X Stop)

Funktion: Programm stoppen

Stoppt die Ausführung des Programms des gewählten Programmplatzes (Slot/Task).

Eingabeformat

```
XSTP □ <Programmplatznummer>
```

<Programmplatznummer> Legt die Nummer des Programmplatzes fest

Programmbeispiel

```
10  XRUN 2                    Startet Programmplatz 2
   :
100 XSTP 2                    Stoppt Programmplatz 2
110 WAIT M_WAI(2) = 1        Wartestatus, bis Programmplatz 2 gestoppt ist
   :
200 XRUN 2                    Startet Programmplatz 2
```

Erläuterung

- Wird ein bereits gestoppter Programmplatz gestoppt, erfolgt keine Fehlermeldung.
- Der XSTP-Befehl stoppt auch ein konstant ausgeführtes Programm.

9.3.71 XRST (X Reset)

Funktion: Programm zurücksetzen

Setzt die Steuerung des Programms des festgelegten Programmplatzes von der aktuellen Zeile auf den Programmanfang zurück.

Eingabeformat

```
XRST □ <Programmplatznummer>
```

<Programmplatznummer> Legt die Nummer des Programmplatzes fest

Programmbeispiel

10	XRUN 2	Startet Programmplatz 2
20	WAIT M_RUN(2) = 1	Wartestatus, bis Betriebssignal des Programmplatzes 2 gleich 1 ist
	:	
	:	
100	XSTP 2	Stoppt Programmplatz 2
110	WAIT M_WAI(2) = 1	Wartestatus, bis Programmplatz 2 gestoppt ist
	:	
150	XRST 2	Programmplatz 2 zurücksetzen
160	WAIT M_PSA(2) = 1	Wartestatus, bis Programmplatz 2 gewählt wurde
	:	
200	XRUN 2	Startet Programmplatz 2
210	WAIT M_RUN(2) = 1	Wartestatus, bis Betriebssignal des Programmplatzes 2 gleich 1 ist
	:	

Erläuterung

- Ein Zurücksetzen des Programms ist nur im gestoppten Zustand des Programmplatzes möglich.

9.3.72 SUBSTITUTE (Substitute)

Funktion: Daten ersetzen

Das Ergebnis einer Operation wird in eine Variable oder Feldvariable übertragen.

Eingabeformat 1

```
<Variablenname> = <Ausdruck 1>
```

Eingabeformat 2

```
<Variablenname> = <Ausdruck 1> DLY <Ausdruck 2>
```

- <Variablenname> Legt den Namen der Variablen fest, in die die Daten übertragen werden (siehe auch Syntaxdiagramme der Variablentypen)
- <Ausdruck 1> Daten, die in die Variable übertragen werden
- <Ausdruck 2> Legt die Verzögerungszeit fest

Programmbeispiel

- 10 P100 = P1 + P2 * 2 Überträgt das Ergebnis der Operation P1 + P2 * 2 in die Variable P100
- 20 M_OUT(17) = 1 DLY 10.0 Setzt Ausgangsbit Nummer 17 für 10 Sekunden auf 1

Erläuterung

- Wird der Befehl für einen Impulsausgang verwendet, wird der Impuls parallel zu den Befehlen der nachfolgenden Zeilen geschaltet.
- Nach Ablauf der eingestellten Verzögerungszeit nimmt der Impulsausgang den Wert an, den er vor Ausführung des Befehles hatte.
- Wird während der festgesetzten Zeit eine END-Anweisung, die letzte Zeile des Programms oder ein NOT-HALT ausgeführt, behält der Impulsausgang seinen gegenwärtigen Zustand bei. Der Ausgangszustand während der Unterbrechung kann über die Systemparameter eingestellt werden, wenn der Ausgang den Zustand annehmen soll, den er vor der Impulsabgabe oder nach abgelaufener Verzögerungszeit haben soll.

A Anhang

A.1 Fehlerdiagnose

Bei Auftreten eines Fehlers wird am Steuergerät eine 5-stellige Fehlernummer auf dem Display „STATUS NUMBER“ angezeigt (z. B. C0010). Die LED auf dem RESET-Taster leuchtet.

Wird eine Taste der Teaching Box (z. B. MENU-Taste) betätigt, erscheint eine 4-stellige Fehlernummer auf dem Display der Teaching Box. Das erste Zeichen der Fehlernummer wird nicht angezeigt. Es erscheint z. B. „0010“ für „C0010“.

In folgender Tabelle sind die Fehlernummern, die Fehlerursachen und die Gegenmaßnahmen aufgeführt. Lässt sich ein Fehler durch die aufgeführten Gegenmaßnahmen nicht beseitigen, setzen Sie sich mit Ihrem Vertriebspartner in Verbindung.

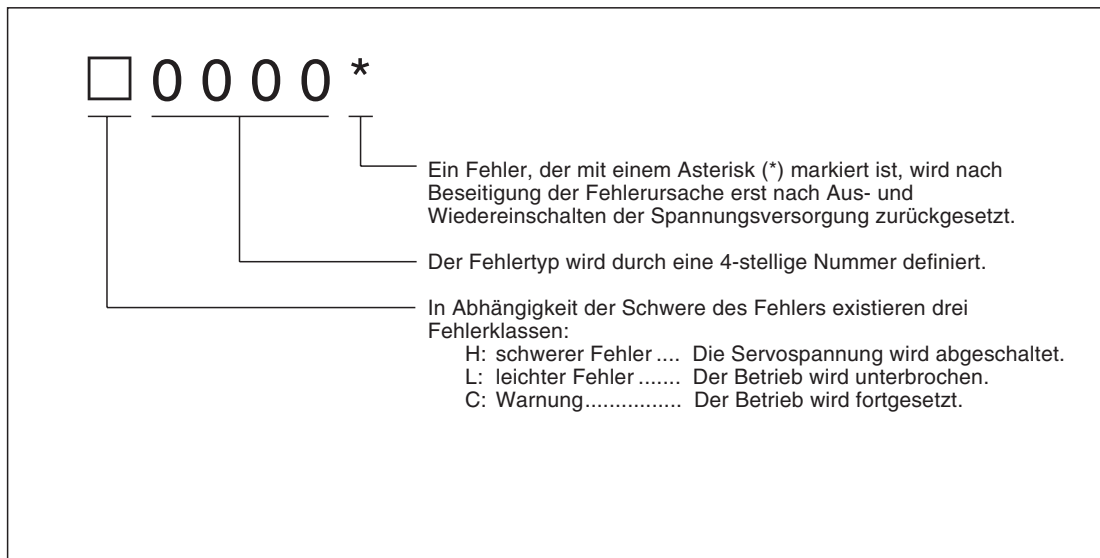


Abb. A-1: Aufbau einer Fehlermeldung

HINWEIS

Die letzte Stelle der Fehlernummer kann eine Achsennummer anzeigen.
Bsp.: Die Fehlernummer H0931 bedeutet Überstrom des Motors der Achse Nr. 1.

A.1.1 Übersicht der Fehlercodes

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
C0010	Datei wurde wegen Versionsabweichung initialisiert.	System startet in diesem Zustand nicht.	Datei wird automatisch initialisiert.
C0011	Systemdaten wurden wegen Versionsabweichung initialisiert.	System startet in diesem Zustand nicht.	Datei wird automatisch initialisiert.
H0012*	Ausschaltroutine wurde nicht vorschriftsmäßig beendet.	Das System ist instabil.	Backup-Daten laden
H0020*	Der Backup-Datenname ist bereits vorhanden.	Der Name ist bereits vergeben.	Namen ändern
H0021*	Kein Anlegen weiterer Backup-Daten möglich	Überlauf des Steuerbereiches	Steuerbereich über Software vergrößern
H0022*	Überlauf des Backup-Bereichs	Zu kleiner Bereich	Backup-Bereich über Software vergrößern
H0030*	Eingabe eines Handfehlersignals	Einstellfehler der Benutzerdaten	Muss zum Rücksetzen der JOG-Betrieb ausgeführt werden, Totmannschalter betätigen
H0031*	Eingabe eines Fehlers in der Pneumatikversorgung der Hand	Einstellfehler der Benutzerdaten	Muss zum Rücksetzen der JOG-Betrieb ausgeführt werden, Totmannschalter betätigen
H0041*	CRC-Fehler im externen E/A-Kanal 1	Fehler auf der Kommunikationsleitung des externen E/A-Kanals 1	Prüfen des Übertragungskabels und der Spannungsversorgung des externen Geräts
H0042*	CRC-Fehler im externen E/A-Kanal 2	Fehler auf der Kommunikationsleitung des externen E/A-Kanals 2	
H0043*	CRC-Fehler im externen E/A-Kanal 3	Fehler auf der Kommunikationsleitung des externen E/A-Kanals 3	
H0050	Eingabe eines externen NOT-HALT-Signals	Das externe Stoppsignal wurde eingegeben.	Externen NOT-HALT-Kreis prüfen
H0051	Fehler im Anschluss des externen NOT-HALT-Kreises		
H0052	Eingabe des externen NOT-HALT-Signals für die Zusatzachse 1	Das externe Stoppsignal wurde in die Schnittstellenkarte der Zusatzachse eingegeben.	Externen NOT-HALT-Kreis 1 der Zusatzachse prüfen
H0053	Eingabe des externen NOT-HALT-Signals für die Zusatzachse 1	Das externe Stoppsignal wurde in den Zusatzverstärker eingegeben.	Externen NOT-HALT-Kreis 2 der Zusatzachse prüfen
H0060	Eingabe des NOT-HALT-Signals über das Steuergerät	Das Stoppsignal wurde über das Steuergerät eingegeben.	NOT-HALT-Kreis des Steuergeräts prüfen
H0061	Fehler im Anschluss des NOT-HALT-Kreises vom Steuergerät		
H0070	Eingabe des NOT-HALT-Signals über die Teaching Box	Das Stoppsignal wurde über die Teaching Box eingegeben.	NOT-HALT-Kreis der Teaching Box prüfen
H0071	Fehler im Anschluss des NOT-HALT-Kreises der Teaching Box		

Tab. A-1: Fehlercodes (1)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
H0072	Fehler beim Abschalten der Teaching Box	Fehlerhaftes Abschalten der Teaching Box	Funktion des Schalters prüfen
H0073	Fehler in der Verbindungsleitung des Schalters zum Abschalten der Teaching Box	Schalter arbeitet nicht korrekt	
H0074	Fehler im System des [ENABLE/DIASABLE]-Schalters der Teaching Box	[ENABLE/DIASABLE]-Schalter arbeitet nicht korrekt	[ENABLE/DIASABLE]-Schalter prüfen
H0081*	Sicherung für die motorbetriebene Greifhand defekt	Sicherung für die motorbetriebene Greifhand defekt	Sicherung wechseln
H0082*	Sicherung für die pneumatische Greifhand defekt	Sicherung für die pneumatische Greifhand defekt	
H0083*	Sicherung der Spannungsversorgung der pneumatischen Greifhand defekt	Sicherung der Spannungsversorgung der pneumatischen Greifhand defekt	
H0084*	Sicherung des Steuergerätes defekt	Sicherung des Steuergerätes defekt	
H0085*	Sicherung der Spannungsversorgung des externen NOT-HALT-Kreises defekt	Sicherung der Spannungsversorgung des externen NOT-HALT-Kreises defekt	
H0090*	Sicherung der parallelen E/A-Schnittstelle defekt	Sicherung der parallelen E/A-Schnittstelle defekt	
H0091	Signal ist bereits einem speziellen Ausgang zugeordnet.	Signal kann nicht 2-mal zugeordnet werden.	Ausgangsnummer oder zugeordneten Parameter ändern
H0500*	Einstellfehler Codierschalter	Fehlerhafte Einstellung des Codierschalters zur Festlegung der beiden internen Zusatzachsenansteuerungen	Einstellung korrigieren
H0510*	Einstellung des externen NOT-Halt-Schalters	Widersprüchliche Einstellung des Codierschalters des Spannungswandlers und des Parameters SVPTYP	
H0520*	Fehlerhafte Achseneinstellung	Die Einstellung der Achse eines Roboters ist auch einem weiteren Roboter zugewiesen.	
H0530*	Speicherfehler im Servoverstärker	Prüfsumme im Servoverstärker fehlerhaft	Versorgungsspannung aus- und wieder einschalten
H0540*	Software-Fehler im Servoverstärker	Zeitüberschreitung bei der Verarbeitung der Daten im Servoverstärker	
H0560*	Fehler im A/D-Wandler des Servoverstärkers	Fehler im A/D-Wandler des Servoverstärkers bei der Initialisierung	
H0630*	Fehler des Servoverstärkers bei nicht verwendeter Achse	Fehler im Leistungsteil einer nicht zum Betrieb verwendeten Achse	
H0670*	CPU-Fehler des seriellen Encoders	Fehler der CPU durch seriellen Encoder	
H0680*	LED-Fehler des seriellen Encoders	Impulse der LED des seriellen Encoders fehlerhaft	
H0700*	Schaltfehler des Schützes	Schütz ist trotz READY OFF-Signal eingeschaltet.	

Tab. A-1: Fehlercodes (2)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
H0710*	Schaltfehler des Überstromrelais der Spannungsversorgung	Überstromrelais der Spannungsversorgung schaltet sich nicht ein.	Versorgungsspannung aus- und wieder einschalten
H0720*	Fehlerhafte Spannungsversorgung (Watch Dog)	Zeitüberschreitung in der Spannungswandlerroutine	
H0730*	Schaltfehler des Überstromrelais der Spannungsversorgung	Überstromrelais der Spannungsversorgung schaltet sich nicht aus.	
H0740*	Fehler im Hauptkreis der Spannungsversorgung	Ladefehler der Kapazität	
H0750*	Speicherfehler des Spannungswandlers	Fehler im Speicherkreis des Spannungswandlers	
H0760*	Spannungswandler- oder A/D-Wandler-Fehler	Fehler im Spannungswandler oder im A/D-Wandler	
H0780*	Watch Dog des Servoverstärkers	Zeitüberschreitung bei der Datenverarbeitung des Servoverstärkers	Versorgungsspannung aus- und wieder einschalten
H0790*	Leiterplattenfehler des Servoverstärkers	Fehler auf der Leiterplatte des Servoverstärkers	
H0800*	Timerfehler im Servoverstärker	Fehler des Timers im Servoverstärker	
H0810*	Unterspannung des Servoverstärkers	Die Spannung zwischen den Klemmen P und N ist auf einen Wert kleiner gleich 200 V abgefallen.	Prüfen des eingangsseitigen Schützes und der Versorgungsspannung
H0820*	Erdschluss eines Servomotors	Erdschlussfehler vom Servomotor	Kabelanschluss prüfen
H0830*	Überspannung des Servoverstärkers	Die Spannung zwischen den Klemmen P und N ist auf einen Wert größer gleich 400 V angestiegen.	Prüfen der Spannungsversorgung und des Werts des regenerativen Widerstandes
H0850*	Offene Phase/fehlerhafter Spannungswert	Eingangsspannung entspricht nicht der eingestellten Spannung.	Eingangsspannung und Wahlschalter zur Einstellung der Eingangsspannung prüfen
H0860	Versorgungsspannung zu hoch	Die Versorgungsspannung zwischen L+ und L- ist größer als 410 V.	
H0880*	Überhitzung des Bremswiderstandes	Bremswiderstand wurde überhitzt.	Versorgungsspannung aus- und nach einer Wartezeit wieder einschalten
H0890	Überhitzung des Servomotors	Thermosicherung des Motors oder des Encoders wurde aktiviert.	Bremswiderstand prüfen
H0910*	Geschwindigkeitsüberschreitung des Servoverstärkers	Überschreitung der Motorgeschwindigkeit um den Faktor 1,2	Versorgungsspannung aus- und wieder einschalten
H0920*	Überstrom im Leistungsteil des Servoverstärker	Überstrom im Servoverstärker oder im Leistungsteil	Anschluss des Motorkabels und Isolation prüfen
H0930*	Motorüberstrom	Der Strom durch den Motor ist zu groß.	Motorkabel prüfen

Tab. A-1: Fehlercodes (3)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
H0940	Überlast 1	Überlast Motor oder Servoverstärker	Last prüfen, Roboter darf nicht mit umliegenden Einrichtungen kollidieren
H0950	Überlast 2	Stromfluss von mindestens 95 % des maximalen Ausgangsstroms für eine Sekunde oder länger	
H0960	Positionsabweichung 1	Bei eingeschaltetem Servo ist die eingestellte Abweichung der aktuellen Position zur Zielposition zu groß.	Last prüfen
H0970	Positionsabweichung 2	Bei eingeschaltetem Servo ist die eingestellte Abweichung der aktuellen Position zur Zielposition zu groß.	Einwirkung externer Kräfte bei eingeschaltetem Servo prüfen
H0980	Positionsabweichung 3	Kein Motorstrom bei Auftreten der Fehlermeldung für Positionsabweichung 1	Motoranschluss prüfen
H1010	Kollisionsüberwachung 2	Fehler bei Kollisionsüberwachung 2	Kollisionsursache beseitigen
H1020	Regenerative Überlast des Servoverstärkers der Zusatzachse	Überhitzung des regenerativen Bremswiderstandes	Prüfen der regenerativen Lastkapazität und des Parameters SVPTYPE
H1030*	Regenerative Überlast des Spannungswandlers	Die regenerative Kapazität des Spannungswandlers ist überschritten worden.	15 min bei eingeschalteter Versorgungsspannung warten, dann Versorgungsspannung aus- und wieder einschalten
H1040*	Keine Kommunikation mit Encoder	Kein Kommunikationsaufbau zwischen Servoverstärker und Encoder	Prüfen des Encoderkabels und des Anschlusses
H1070*	Kommunikationsfehler mit Encoder	Unterbrechung der Kommunikation mit dem Encoder	
H1080*	Kommunikationsfehler des Servoverstärkers	Unterbrechung der Kommunikation zwischen Steuergerät und Servoverstärker	Kabelanschluss und Kabel prüfen
H1090*	Initialisierungsfehler des Servoverstärkers	Fehler bei der Initialisierung des Servoverstärkers	Einstellungen der Achsen prüfen (Parameter, Codierschalter)
H1100*	Kommunikationsfehler des Servoverstärkers	Unterbrechung der Kommunikation zwischen Steuergerät und Servoverstärker	Kabelanschluss und Kabel prüfen
H1110*	RS232C-Kommunikationsfehler mit Servoverstärker	Kommunikationsfehler zwischen Servoverstärker und PC	
H1120*	Verlust der Absolutposition	Daten der Absolutposition sind gelöscht.	Batteriespannung und Encoderkabel prüfen
H1130*	Encoderdaten fehlerhaft	Encoderdaten pro Umdrehung sind fehlerhaft.	Vorgang wiederholen und Umgebungsbedingungen prüfen

Tab. A-1: Fehlercodes (4)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
H1140*	CRC-Fehler der Daten des Servoverstärkers	Prüfsummenfehler in den Daten zum Steuergerät	Kabelanschluss und Kabel prüfen
H1150*	Fehlerhafter Befehlswert der Kommunikationsdaten des Servoverstärkers	Die Befehlsdaten für die Zielposition sind zu groß.	
H1160*	Fehler in der Datenlänge zum Servoverstärker	Die Länge der Daten vom Steuergerät ist fehlerhaft.	
H1170*	Kommunikationsdaten zum Servoverstärker fehlerhaft	Die Daten vom Steuergerät sind fehlerhaft.	
H1180*	Fehler 1 der Rückmeldeimpulse des Servoverstärkers	Fehlende Impulse im Rückmeldesignal vom Encoder	Encoderkabel und Anschluss prüfen
H1200*	CRC-Fehler in den Kommunikationsdaten des Servoverstärkers	In den Kommunikationsdaten vom Servoverstärker ist ein CRC-Fehler aufgetreten.	Encoderkabel und Anschluss prüfen
H1210*	ID-Fehler in den Kommunikationsdaten des Servoverstärkers	In den Kommunikationsdaten vom Servoverstärker ist ein ID-Fehler aufgetreten.	
H1220*	Fehler der Achsennummer in den Kommunikationsdaten des Servoverstärkers	In den Kommunikationsdaten vom Servoverstärker ist ein Fehler der Achsennummer aufgetreten.	
H1230*	Sub-ID-Fehler in den Kommunikationsdaten des Servoverstärkers	In den Kommunikationsdaten vom Servoverstärker ist ein Sub-ID-Fehler aufgetreten.	
H1240*	Datennummernfehler in den Kommunikationsdaten des Servoverstärkers	In den Kommunikationsdaten vom Servoverstärker ist ein Datennummernfehler aufgetreten.	
H1250*	Parameterfehler des Servoverstärkers	Fehlerhafter Einstellung eines Parameters des Servoverstärkers	Parameter auf einen zulässigen Wert einstellen
C1290	Absolute Positionsabweichung des Servoverstärkers	Abweichung der Daten der Absolutwertposition beim Einschalten	Prüfen, ob die Achse aufgrund des Eigengewichts oder durch äußere Krafteinwirkung beim Einschalten bewegt wurde
C1320	Fehler des schnellen Zählers für die seriellen Multirotationsdaten des Servoverstärkers	Fehlerhafte Multirotationsdaten des Encoders	Versorgungsspannung aus- und wieder einschalten
C1330	Niedrige Batteriespannung des Encoders	Batteriespannung für den Encoder ist abgesunken.	Batterie wechseln
C1340	Regenerative Überlast des Servoverstärkers	Regenerative Energie des Zusatzachsenverstärkers von 80 % oder mehr	Prüfen des Parameters zur Einstellung der regenerativen Kapazität
C1350	Überlast des Servoverstärkers	Überlast von 80 % oder mehr	Last prüfen, Roboter darf nicht mit umliegenden Einrichtungen kollidieren
C1360	Warnung des Zählers für Absolutwertposition	Fehlerhafter Encoderzähler	Batterie und Encoderkabel prüfen
C1370	Parameterfehler des Servoverstärkers	Einstellung eines Parameters außerhalb des zulässigen Wertebereichs	Parameter auf einen zulässigen Wert einstellen
C1400	Kurzzeitige Überschreitung der regenerativen Energie des Servoverstärkers	Kurzzeitige Überschreitung der regenerativen Energie des Spannungswandlers	Regenerative Kapazität prüfen
C1420	Regenerative Überlast des Servoverstärkers	Regenerative Energie von 80 % oder mehr	Geschwindigkeit des Roboters verringern

Tab. A-1: Fehlercodes (5)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
C1430	Hauptkreis des Servoverstärkers der Zusatzachse ausgeschaltet	Der Servo wurde bei ausgeschaltetem Hauptkreis eingeschaltet.	Hauptkreis einschalten
H1440*	Software-Fehler 2 im Servoverstärker	Prozessorfehler	Versorgungsspannung aus- und wieder einschalten
H1570	Unregistrierter Servofehler	Es ist ein unregistrierter Servofehler aufgetreten.	Kann der Fehler nicht zurückgesetzt werden, schalten Sie die Spannungsversorgung aus und wieder ein.
C1580	Unregistrierte Servowarnung	Es ist eine unregistrierte Servowarnung aufgetreten.	Kann die Warnung nicht zurückgesetzt werden, schalten Sie die Spannungsversorgung aus und wieder ein.
H1600*	Kein Roboter definiert	Keiner der Roboter wurde definiert	Mindestens einen Roboter definieren
H1610*	Robotername ungültig	Der Robotername ist ungültig oder nicht registriert.	Korrekt einstellen
C1620	Roboter Nummer ungültig	Die gewählte Roboter Nummer ist ungültig.	Korrekte Roboter Nummer einstellen
C1630	Während eines Servofehlers kann die Funktion Servo ON nicht ausgeführt werden.	Während eines Servofehlers kann die Servospannung nicht eingeschaltet werden.	Servofehler zurücksetzen und dann Servospannung einschalten
C1640	Bei ausgeschaltetem Totmannschalter kann die Funktion Servo ON nicht ausgeführt werden.	Bei ausgeschaltetem Totmannschalter kann die Servospannung nicht eingeschaltet werden.	Totmannschalter betätigen und dann Servospannung einschalten
C1650	Bei gelöster Bremse kann die Funktion Servo ON nicht ausgeführt werden.	Bei gelöster Bremse kann die Servospannung nicht eingeschaltet werden.	Bremsen aktivieren und dann Servospannung einschalten
C1660	Während des Servo-ON-Vorgangs kann die Funktion Servo ON nicht ausgeführt werden.	Während des Servo-ON-Vorgangs kann die Servospannung nicht eingeschaltet werden.	Servospannung nicht während des Servo-ON-Vorgangs einschalten
C1670	Servo-OFF-Zustand aktiv	Der Servo-OFF-Zustand ist aktiv.	Servospannung nicht im Servo-OFF-Zustand ausschalten
H1680	Zeitüberschreitung beim Servo-ON-Vorgang	Die Servos sind nicht innerhalb der zulässigen Zeit eingeschaltet worden.	Servoverstärker auf Fehler prüfen
H1681	Fehlerhaftes Abschalten der Servospannung	Die Servospannung wurde ungewollt abgeschaltet.	Servoverstärker und Haupt-CPU prüfen
C1690	Während der Totmannschalter ausgeschaltet ist, kann keine Bremse gelöst werden.	Es kann keine Bremse gelöst werden, während der Totmannschalter ausgeschaltet ist.	Totmannschalter einschalten und dann Bremse lösen
C1700	Bei aktiviertem NOT-HALT kann keine Bremse gelöst werden.	Es kann keine Bremse gelöst werden, wenn das NOT-HALT-Signal eingegeben wird.	NOT-HALT-Status aufheben und dann Bremse lösen
C1710	Im Servo-ON-Zustand kann keine Bremse gesteuert werden.	Bei eingeschalteter Servospannung kann keine Bremse gesteuert werden.	Servospannung ausschalten und dann Bremsen steuern
C1720	Während die Bremsen gelöst werden, kann kein weiterer Prozess zum Lösen der Bremsen ausgeführt werden.	Es kann kein weiterer Prozess zum Lösen der Bremsen ausgeführt werden, während die Bremsen gelöst werden.	Bremsen nicht lösen, während der Prozess „Bremsen lösen“ aktiv ist

Tab. A-1: Fehlercodes (6)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
C1730	Während die Bremsen aktiviert werden, kann kein weiterer Prozess zur Aktivierung der Bremsen ausgeführt werden.	Es kann kein weiterer Prozess zur Aktivierung der Bremsen ausgeführt werden, während die Bremsen aktiviert werden.	Bremsen nicht aktivieren, während der Prozess „Bremsen aktivieren“ aktiv ist
C1740	Servoparameter können nicht geändert werden.	Während der Einstellung von Parametern können keine weiteren Parameter eingestellt werden.	Einstellung wiederholen
C1750	Einstellung der Servoparameter ist fehlgeschlagen.	Es konnte keine Einstellung der Servoparameter vorgenommen werden.	
C1760	Ungültige Daten der Grundposition	Die Daten der Grundposition sind nicht korrekt eingestellt.	Korrekte Daten der Grundposition einstellen
C1770	Einstellung der Daten der Grundposition nicht abgeschlossen	Die Grundposition ist nicht eingestellt.	Einstellung wiederholen
C1780	Ungültige Achseneinstellung für Grundposition	Fehlerhafte Einstellung der Grundposition einer Achse	Setzen der Grundposition dieser Achse
C1781	Keine Einstellung der Grundposition im Servo-ON-Zustand	Es wurde versucht, die Grundposition bei eingeschalteter Servospannung einzustellen.	Vor der Einstellung der Grundposition Servospannung ausschalten
H1790*	Fehlerhafte Einstellung der Fahrweggrenzen	Einstellung des Parameters MEJAR ist fehlerhaft	Einstellung des Parameters MEJAR korrigieren
H1800*	Fehlerhafte Einstellung der Fahrweggrenzen für Absolutwertpositionierung	Einstellung des Parameters MEMAR ist fehlerhaft	Einstellung des Parameters MEMAR korrigieren
H1810*	Einstellfehler der benutzerdefinierten Grundposition	Einstellung des Parameters USERORG ist fehlerhaft	Einstellung des Parameters USERORG korrigieren
L1820	Positionsdaten stimmen nicht überein. Grundposition überprüfen	Positionsdaten bei ausgeschalteter Spannungsversorgung verändert	Überprüfung der Grundposition Bei Verschiebung diese neu einstellen
L2000	Servospannung ist ausgeschaltet.	Zeitüberschreitung beim Einschalten der Servospannung	Servospannung einschalten und Neustart ausführen
L2010	Keine Impulsausgabe	Fehlerhafte Festlegung des Impulsausgangs	Programm korrigieren
L2020	Lesen eines externen Befehls	Während des Lesens eines externen Befehls konnte ein Befehl nicht ausgeführt werden.	
L2030	Anforderung JOG-Betrieb nicht akzeptiert	Die Anforderung zum JOG-Betrieb wurde ausgeführt aber nicht akzeptiert.	Anforderung wiederholen
L2031	Fehlerhafte Einstellung der Dimensionen	Einstellung der Parameter JOGTSJ oder JOGJSP ist fehlerhaft	Dimension von 5 oder kleiner einstellen

Tab. A-1: Fehlercodes (7)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
L2040	Überschreitung des Absolutwertbereiches	Der Absolutwertbereich wurde überschritten.	Totmannschalter betätigen und Roboter im JOG-Betrieb in den zulässigen Bereich zurückfahren
L2041	Überschreitung des Absolutwertbereiches der Achse 1 in +-Richtung	Der Bereich des Absolutwerts der Achse 1 in +-Richtung wurde überschritten.	
L2042	Überschreitung des Absolutwertbereiches der Achse 2 in +-Richtung	Der Bereich des Absolutwerts der Achse 2 in +-Richtung wurde überschritten.	
L2043	Überschreitung des Absolutwertbereiches der Achse 3 in +-Richtung	Der Bereich des Absolutwerts der Achse 3 in +-Richtung wurde überschritten.	
L2044	Überschreitung des Absolutwertbereiches der Achse 4 in +-Richtung	Der Bereich des Absolutwerts der Achse 4 in +-Richtung wurde überschritten.	Totmannschalter betätigen und Roboter im JOG-Betrieb in den zulässigen Bereich zurückfahren
L2045	Überschreitung des Absolutwertbereiches der Achse 5 in +-Richtung	Der Bereich des Absolutwerts der Achse 5 in +-Richtung wurde überschritten.	
L2046	Überschreitung des Absolutwertbereiches der Achse 6 in +-Richtung	Der Bereich des Absolutwerts der Achse 6 in +-Richtung wurde überschritten.	
L2047	Überschreitung des Absolutwertbereiches der Achse 7 in +-Richtung	Der Bereich des Absolutwerts der Achse 7 in +-Richtung wurde überschritten.	
L2048	Überschreitung des Absolutwertbereiches der Achse 8 in +-Richtung	Der Bereich des Absolutwerts der Achse 8 in +-Richtung wurde überschritten.	
L2051	Überschreitung des Absolutwertbereiches der Achse 1 in --Richtung	Der Bereich des Absolutwerts der Achse 1 in --Richtung wurde überschritten.	
L2052	Überschreitung des Absolutwertbereiches der Achse 2 in --Richtung	Der Bereich des Absolutwerts der Achse 2 in --Richtung wurde überschritten.	
L2053	Überschreitung des Absolutwertbereiches der Achse 3 in --Richtung	Der Bereich des Absolutwerts der Achse 3 in --Richtung wurde überschritten.	
L2054	Überschreitung des Absolutwertbereiches der Achse 4 in --Richtung	Der Bereich des Absolutwerts der Achse 4 in --Richtung wurde überschritten.	
L2055	Überschreitung des Absolutwertbereiches der Achse 5 in --Richtung	Der Bereich des Absolutwerts der Achse 5 in --Richtung wurde überschritten.	
L2056	Überschreitung des Absolutwertbereiches der Achse 6 in --Richtung	Der Bereich des Absolutwerts der Achse 6 in --Richtung wurde überschritten.	
L2057	Überschreitung des Absolutwertbereiches der Achse 7 in --Richtung	Der Bereich des Absolutwerts der Achse 7 in --Richtung wurde überschritten.	
L2058	Überschreitung des Absolutwertbereiches der Achse 8 in --Richtung	Der Bereich des Absolutwerts der Achse 8 in --Richtung wurde überschritten.	
H2091	Überschreitung des Überlagerungsbereichs 1	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 1 zu bewegen.	

Tab. A-1: Fehlercodes (8)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
H2092	Überschreitung des Überlagerungsbereichs 2	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 2 zu bewegen.	Position im Betrieb prüfen
H2093	Überschreitung des Überlagerungsbereichs 3	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 3 zu bewegen.	
H2094	Überschreitung des Überlagerungsbereichs 4	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 4 zu bewegen.	
H2095	Überschreitung des Überlagerungsbereichs 5	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 5 zu bewegen.	
H2096	Überschreitung des Überlagerungsbereichs 6	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 6 zu bewegen.	
H2097	Überschreitung des Überlagerungsbereichs 7	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 7 zu bewegen.	
H2098	Überschreitung des Überlagerungsbereichs 8	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 8 zu bewegen.	
H2099	Überschreitung des Überlagerungsbereichs 9	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 9 zu bewegen.	
H2100	Überschreitung des Überlagerungsbereichs 10	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 10 zu bewegen.	
H2101	Überschreitung des Überlagerungsbereichs 11	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 11 zu bewegen.	
H2102	Überschreitung des Überlagerungsbereichs 12	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 12 zu bewegen.	
H2103	Überschreitung des Überlagerungsbereichs 13	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 13 zu bewegen.	
H2104	Überschreitung des Überlagerungsbereichs 14	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 14 zu bewegen.	
H2105	Überschreitung des Überlagerungsbereichs 15	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 15 zu bewegen.	
H2106	Überschreitung des Überlagerungsbereichs 16	Es wurde versucht, den Roboter außerhalb des Überlagerungsbereichs 16 zu bewegen.	
H2111	Überschreitung der Verkehrswegsbegrenzungsebene 1	Es wurde versucht, den Roboter außerhalb des Verkehrswegsbegrenzungsebene 1 zu bewegen.	
H2112	Überschreitung der Verkehrswegsbegrenzungsebene 2	Es wurde versucht, den Roboter außerhalb des Verkehrswegsbegrenzungsebene 2 zu bewegen.	
H2113	Überschreitung der Verkehrswegsbegrenzungsebene 3	Es wurde versucht, den Roboter außerhalb des Verkehrswegsbegrenzungsebene 3 zu bewegen.	

Tab. A-1: Fehlercodes (9)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
H2114	Überschreitung der Verfahrwegsbegrenzungsebene 4	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 4 zu bewegen.	Position im Betrieb prüfen
H2115	Überschreitung der Verfahrwegsbegrenzungsebene 5	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 5 zu bewegen.	
H2116	Überschreitung der Verfahrwegsbegrenzungsebene 6	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 6 zu bewegen.	
H2117	Überschreitung der Verfahrwegsbegrenzungsebene 7	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 7 zu bewegen.	
H2118	Überschreitung der Verfahrwegsbegrenzungsebene 8	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 8 zu bewegen.	
H2119	Überschreitung der Verfahrwegsbegrenzungsebene 9	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 9 zu bewegen.	
H2120	Überschreitung der Verfahrwegsbegrenzungsebene 10	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 10 zu bewegen.	
H2121	Überschreitung der Verfahrwegsbegrenzungsebene 11	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 11 zu bewegen.	
H2122	Überschreitung der Verfahrwegsbegrenzungsebene 12	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 12 zu bewegen.	
H2123	Überschreitung der Verfahrwegsbegrenzungsebene 13	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 13 zu bewegen.	
H2124	Überschreitung der Verfahrwegsbegrenzungsebene 14	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 14 zu bewegen.	
H2125	Überschreitung der Verfahrwegsbegrenzungsebene 15	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 15 zu bewegen.	
H2126	Überschreitung der Verfahrwegsbegrenzungsebene 16	Es wurde versucht, den Roboter außerhalb des Verfahrwegsbegrenzungsebene 16 zu bewegen.	
H2129	Fehlerhafte Daten zur Einstellung der Verfahrwegsbegrenzungsebene	Daten zur Festlegung der Verfahrwegsbegrenzungsebene sind fehlerhaft	

Tab. A-1: Fehlercodes (10)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
H2130	Geschwindigkeitsüberschreitung	Geschwindigkeitsgrenzwert wurde überschritten.	Roboterstellung im Betrieb prüfen
H2131	Geschwindigkeitsüberschreitung Achse 1	Geschwindigkeitsgrenzwert der Achse 1 wurde überschritten.	
H2132	Geschwindigkeitsüberschreitung Achse 2	Geschwindigkeitsgrenzwert der Achse 2 wurde überschritten.	
H2133	Geschwindigkeitsüberschreitung Achse 3	Geschwindigkeitsgrenzwert der Achse 3 wurde überschritten.	
H2134	Geschwindigkeitsüberschreitung Achse 4	Geschwindigkeitsgrenzwert der Achse 4 wurde überschritten.	
H2135	Geschwindigkeitsüberschreitung Achse 5	Geschwindigkeitsgrenzwert der Achse 5 wurde überschritten.	
H2136	Geschwindigkeitsüberschreitung Achse 6	Geschwindigkeitsgrenzwert der Achse 6 wurde überschritten.	
H2137	Geschwindigkeitsüberschreitung Achse 7	Geschwindigkeitsgrenzwert der Achse 7 wurde überschritten.	
H2138	Geschwindigkeitsüberschreitung Achse 8	Geschwindigkeitsgrenzwert der Achse 8 wurde überschritten.	
H2140	Überschreitung des Absolutwertbereiches	Der Bereich des Absolutwerts der Achse 1 in +-Richtung wurde überschritten.	Totmannschalter betätigen und Roboter im JOG-Betrieb in den zulässigen Bereich zurückfahren
H2141	Überschreitung des Absolutwertbereiches der Achse 1 in +-Richtung	Der Bereich des Absolutwerts der Achse 1 in +-Richtung wurde überschritten.	
H2142	Überschreitung des Absolutwertbereiches der Achse 2 in +-Richtung	Der Bereich des Absolutwerts der Achse 2 in +-Richtung wurde überschritten.	
H2143	Überschreitung des Absolutwertbereiches der Achse 3 in +-Richtung	Der Bereich des Absolutwerts der Achse 3 in +-Richtung wurde überschritten.	
H2144	Überschreitung des Absolutwertbereiches der Achse 4 in +-Richtung	Der Bereich des Absolutwerts der Achse 4 in +-Richtung wurde überschritten.	
H2145	Überschreitung des Absolutwertbereiches der Achse 5 in +-Richtung	Der Bereich des Absolutwerts der Achse 5 in +-Richtung wurde überschritten.	
H2146	Überschreitung des Absolutwertbereiches der Achse 6 in +-Richtung	Der Bereich des Absolutwerts der Achse 6 in +-Richtung wurde überschritten.	
H2147	Überschreitung des Absolutwertbereiches der Achse 7 in +-Richtung	Der Bereich des Absolutwerts der Achse 7 in +-Richtung wurde überschritten.	
H2148	Überschreitung des Absolutwertbereiches der Achse 8 in +-Richtung	Der Bereich des Absolutwerts der Achse 8 in +-Richtung wurde überschritten.	

Tab. A-1: Fehlercodes (11)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
H2151	Überschreitung des Absolutwertbereiches der Achse 1 in --Richtung	Der Bereich des Absolutwerts der Achse 1 in --Richtung wurde überschritten.	Totmannschalter betätigen und Roboter im JOG-Betrieb in den zulässigen Bereich zurückfahren
H2152	Überschreitung des Absolutwertbereiches der Achse 2 in --Richtung	Der Bereich des Absolutwerts der Achse 2 in --Richtung wurde überschritten.	
H2153	Überschreitung des Absolutwertbereiches der Achse 3 in --Richtung	Der Bereich des Absolutwerts der Achse 3 in --Richtung wurde überschritten.	
H2154	Überschreitung des Absolutwertbereiches der Achse 4 in --Richtung	Der Bereich des Absolutwerts der Achse 4 in --Richtung wurde überschritten.	
H2155	Überschreitung des Absolutwertbereiches der Achse 5 in --Richtung	Der Bereich des Absolutwerts der Achse 5 in --Richtung wurde überschritten.	
H2156	Überschreitung des Absolutwertbereiches der Achse 6 in --Richtung	Der Bereich des Absolutwerts der Achse 6 in --Richtung wurde überschritten.	
H2157	Überschreitung des Absolutwertbereiches der Achse 7 in --Richtung	Der Bereich des Absolutwerts der Achse 7 in --Richtung wurde überschritten.	
H2158	Überschreitung des Absolutwertbereiches der Achse 8 in --Richtung	Der Bereich des Absolutwerts der Achse 8 in --Richtung wurde überschritten.	
H2160	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen	Der Verfahrwegbereich für Gelenkbewegungen wurde überschritten.	Roboterstellung im Betrieb prüfen
H2161	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 1 in +-Richtung	Der Verfahrwegbereich der Achse 1 für Gelenkbewegungen wurde in +-Richtung überschritten.	
H2162	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 2 in +-Richtung	Der Verfahrwegbereich der Achse 2 für Gelenkbewegungen wurde in +-Richtung überschritten.	
H2163	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 3 in +-Richtung	Der Verfahrwegbereich der Achse 3 für Gelenkbewegungen wurde in +-Richtung überschritten.	
H2164	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 4 in +-Richtung	Der Verfahrwegbereich der Achse 4 für Gelenkbewegungen wurde in +-Richtung überschritten.	
H2165	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 5 in +-Richtung	Der Verfahrwegbereich der Achse 5 für Gelenkbewegungen wurde in +-Richtung überschritten.	
H2166	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 6 in +-Richtung	Der Verfahrwegbereich der Achse 6 für Gelenkbewegungen wurde in +-Richtung überschritten.	
H2167	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 7 in +-Richtung	Der Verfahrwegbereich der Achse 7 für Gelenkbewegungen wurde in +-Richtung überschritten.	

Tab. A-1: Fehlercodes (12)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
H2168	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 8 in +-Richtung	Der Verfahrwegbereich der Achse 8 für Gelenkbewegungen wurde in +-Richtung überschritten.	Roboterstellung im Betrieb prüfen
H2171	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 1 in --Richtung	Der Verfahrwegbereich der Achse 1 für Gelenkbewegungen wurde in --Richtung überschritten.	
H2172	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 2 in --Richtung	Der Verfahrwegbereich der Achse 2 für Gelenkbewegungen wurde in --Richtung überschritten.	
H2173	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 3 in --Richtung	Der Verfahrwegbereich der Achse 3 für Gelenkbewegungen wurde in --Richtung überschritten.	
H2174	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 4 in --Richtung	Der Verfahrwegbereich der Achse 4 für Gelenkbewegungen wurde in --Richtung überschritten.	
H2175	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 5 in --Richtung	Der Verfahrwegbereich der Achse 5 für Gelenkbewegungen wurde in --Richtung überschritten.	
H2176	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 6 in --Richtung	Der Verfahrwegbereich der Achse 6 für Gelenkbewegungen wurde in --Richtung überschritten.	
H2177	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 7 in --Richtung	Der Verfahrwegbereich der Achse 7 für Gelenkbewegungen wurde in --Richtung überschritten.	
H2178	Überschreitung der Verfahrweggrenzen für Gelenkbewegungen der Achse 8 in --Richtung	Der Verfahrwegbereich der Achse 8 für Gelenkbewegungen wurde in --Richtung überschritten.	
H2180	Überschreitung der orthogonalen Bereichsgrenze	Die orthogonale Bereichsgrenze wurden überschritten.	
H2181	Überschreitung des orthogonalen Bereiches der Achse X in +-Richtung	Die orthogonale Bereichsgrenze der X-Achse wurden in +-Richtung überschritten.	
H2182	Überschreitung des orthogonalen Bereiches der Achse Y in +-Richtung	Die orthogonale Bereichsgrenze der Y-Achse wurden in +-Richtung überschritten.	
H2183	Überschreitung des orthogonalen Bereiches der Achse Z in +-Richtung	Die orthogonale Bereichsgrenze der Z-Achse wurden in +-Richtung überschritten.	
L2600	Positionsdaten außerhalb des Verfahrwegbereiches	Positionsdaten liegen außerhalb des Verfahrwegbereiches.	Position korrigieren
L2601	Startposition außerhalb des Verfahrwegbereiches	Startposition liegt außerhalb des Verfahrwegbereiches.	
L2602	Zielposition außerhalb des Verfahrwegbereiches	Zielposition liegt außerhalb des Verfahrwegbereiches.	
L2603	Zwischenposition außerhalb des Verfahrwegbereiches	Zwischenposition liegt außerhalb des Verfahrwegbereiches.	

Tab. A-1: Fehlercodes (13)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
L2700	CMP-Befehlsmodus fehlerhaft	Festgelegter Modus weicht vom aktuellen Modus ab	CMP OFF-Befehl ausführen und dann festlegen
L2800	Positionsdaten fehlerhaft	Die Positionsdaten sind für den Interpolationsprozess fehlerhaft.	Position korrigieren
L2801	Positionsdaten der Startposition fehlerhaft	Die Positionsdaten der Startposition sind für den Interpolationsprozess fehlerhaft.	
L2802	Positionsdaten der Zielposition fehlerhaft	Die Positionsdaten der Zielposition sind für den Interpolationsprozess fehlerhaft.	
L2803	Positionsdaten der Hilfsposition fehlerhaft	Die Positionsdaten der Hilfsposition sind für den Interpolationsprozess fehlerhaft.	
L2810	Stellungsmerker fehlerhaft	Der Stellungsmerker der Start- und der Endposition müssen für Linear- bzw. Kreis-Interpolation passen.	
H2820	Übersteuerung der Beschleunigung/Abbremsung fehlerhaft	Die Werte zur Festlegung der Übersteuerung der Beschleunigung/Abbremsung sind fehlerhaft.	Korrekten Wert einstellen
H2830	Systemfehler	Es wurde eine für die Stellung ungültige Interpolationsmethode verwendet.	MITSUBISHI-Vertriebspartner kontaktieren
H2840*	Fehler beim Lesen der Parameter für den Interpolationsprozess	Parameter sind nicht eingestellt.	Parameter korrekt einstellen
H2850	Systemfehler	Ungültige Norm	MITSUBISHI-Vertriebspartner kontaktieren
H2860	Systemfehler	Es wurde eine ungültige Interpolationsmethode verwendet.	
H2870	Systemfehler	Die Positionsdaten für die Interpolation wurden nicht definiert.	
H2880*	Kein Speicherplatz für die Daten des Interpolationsprozesses	Zu wenig Speicherplatz	
H2890	Systemfehler	Undefinierte Fehlernummer	
L2900	Systemfehler M O O	—	
L3100	Interner Prozessorfehler	—	
L3110	Bereichsfehler eines Befehlsparameters	Befehlsparameter liegt außerhalb des zulässigen Einstellbereiches.	Bereich prüfen und Eingabe wiederholen
L3120	Anzahl der Befehlsparameter fehlerhaft	Anzahl der Befehlsparameter entspricht nicht dem zulässigen Wert.	Anzahl der Befehlsparameter prüfen und korrigieren
L3130	Datei bereits geöffnet	Es wurde versucht, eine bereits geöffnete Datei zu öffnen.	Dateinummer prüfen und evtl. richtige Datei öffnen
L3140	Öffnen einer Datei nicht möglich	Die Datei kann nicht geöffnet werden.	
L3150	Kein Schreibvorgang möglich, da die Zugriffsmethode auf „INPUT“ gesetzt ist	Kein Schreibvorgang möglich, da die Zugriffsmethode auf „INPUT“ gesetzt ist	Dateinummer prüfen, Zugriffsmethode prüfen und Vorgang wiederholen
L3170	Kein Schreibvorgang möglich, da die Zugriffsmethode auf „OUTPUT“ gesetzt ist	Kein Schreibvorgang möglich, da die Zugriffsmethode auf „OUTPUT“ gesetzt ist	

Tab. A-1: Fehlercodes (14)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
L3180	Systemfehler Einlesen von Felddaten nicht möglich	—	MITSUBISHI-Vertriebspartner kontaktieren
L3200	Datei kann nicht gelesen werden.	Ein Lesen der Datei ist nicht möglich.	Dateiinhalte prüfen
L3210	Variable kann nicht geschrieben werden.	Ein Schreiben der Variablen ist nicht möglich.	Schreibschutzstatus der Variablen prüfen
L3220	Zu viele Programmverschachtelungen	Die Anzahl der zulässigen Programmverschachtelungen ist überschritten worden.	Programm korrigieren und Ausführung wiederholen
L3230	Anzahl der FOR- und NEXT-Anweisungen ist nicht identisch.	Die Anzahl der FOR- und NEXT-Anweisungen ist nicht identisch.	
L3240	Die Anzahl der Programmebenen übersteigt 16.	Die Anzahl der Programmebenen übersteigt 16.	
L3250	Die Anzahl der WHILE- und WHEN-Anweisungen ist nicht identisch.	Die Anzahl der WHILE- und WHEN-Anweisungen ist nicht identisch.	
L3260	Parallele Ausführung der festgelegten Programme nicht möglich	Parallele Ausführung der festgelegten Programme nicht möglich	Einzelne Programme auswählen und ausführen
L3270	Befehlsgröße überschritten	Befehlsgröße überschritten	Maximal 256 Zeichen verwenden
L3280	Ausführung ohne GETM-Befehl nicht möglich	Ausführung ohne GETM-Befehl nicht möglich	GETM-Befehl ausführen und Vorgang wiederholen
L3281	Keine Ausführung im Betrieb	Keine Ausführung im Betrieb	Keine Ausführung im Betrieb
L3282	Starten nicht möglich	Fehlerhafte Programmwahl oder Attributeinstellung	Fehlerhafte Programmwahl oder Attributeinstellung
L3283	Vorheriger Befehl wird ausgeführt	Vorheriger Befehl wird ausgeführt	Vorheriger Befehl wird ausgeführt
L3284	Keine Ausführung während der Editierung möglich	Keine Ausführung während der Editierung möglich	Keine Ausführung während der Editierung möglich
L3285	Programmwahl angeben	Programmwahl angeben	Programmwahl angeben
L3286	Start eines leeren Programms	Start eines leeren Programms	Start eines leeren Programms
L3287	Programm kann nicht bei Startbedingungen ERR und ALWAYS gestartet werden.	Programm kann nicht bei Startbedingungen ERR und ALWAYS gestartet werden.	Programm kann nicht bei Startbedingungen ERR und ALWAYS gestartet werden.
L3288	Keine Ausführung während einer Programmeditierung möglich	Keine Ausführung während einer Programmeditierung möglich	Keine Ausführung während einer Programmeditierung möglich
L3289	Das in der Programmplatzliste ausgewählte Programm existiert nicht.	Das in der Programmplatzliste ausgewählte Programm existiert nicht.	Das in der Programmplatzliste ausgewählte Programm existiert nicht.
L3290	Keine Ausführung der Systemanwendung möglich	Keine Ausführung der Systemanwendung möglich	Prüfen, ob eine andere Anwendung (Benutzeranwendung) ausgeführt wird
L3300	Keine Ausführung einer Benutzeranwendung möglich	Keine Ausführung einer Benutzeranwendung möglich	Prüfen, ob die Systemanwendung ausgeführt wird
L3310	Der Befehl XRUN kann nicht ausgeführt werden, wenn das Programm bereits abgearbeitet wird.	Der Befehl XRUN kann nicht ausgeführt werden, wenn das Programm bereits abgearbeitet wird.	Programm prüfen

Tab. A-1: Fehlercodes (15)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
L3320	Der Befehl XRUN kann nicht ausgeführt werden, wenn kein Programm gewählt wurde.	Der Befehl XRUN kann nicht ausgeführt werden, wenn kein Programm gewählt wurde.	Programm prüfen
L3330	Der Befehl XSTP kann nicht ausgeführt werden, wenn kein Programm gewählt wurde.	Der Befehl XSTP kann nicht ausgeführt werden, wenn kein Programm gewählt wurde.	
L3340	Der Befehl XRST kann nicht ausgeführt werden, wenn kein Programm gewählt wurde.	Der Befehl XRST kann nicht ausgeführt werden, wenn kein Programm gewählt wurde.	
L3350	Der Befehl XRST kann nicht ausgeführt werden, wenn das Programm abgearbeitet wird.	Der Befehl XRST kann nicht ausgeführt werden, wenn das Programm abgearbeitet wird.	
L3360	Die Programmwahl muss vor Ausführung des XLOAD-Befehls freigegeben worden sein.	Der Befehl XLOAD kann nicht ausgeführt werden, wenn die Programmwahl nicht freigegeben worden ist.	Freigabe des Programms und anschließend XLOAD-Befehl ausführen
L3361	Fehler bei der Programmfestlegung in der Anwendungsliste	Fehler bei der Programmfestlegung in der Anwendungsliste	Fehler bei der Programmfestlegung in der Anwendungsliste
L3400	Es wurden mehr als 7 Programmaufrufe mit dem CALLP-Befehl ausgeführt.	Es wurden mehr als 7 Programmaufrufe mit dem CALLP-Befehl ausgeführt.	Programm prüfen
L3500	Das mit INPUT angegebene Format ist fehlerhaft.	Das mit INPUT angegebene Format ist fehlerhaft.	Format prüfen
L3600	Sprungziel existiert nicht.	Sprungziel existiert nicht.	Sprungziel prüfen
L3700	Verwendung einer nicht definierten Variablen	Verwendung einer nicht definierten Variablen	Variable vor Verwendung definieren
L3710	Es wurden mehr als 7 Programmaufrufe mit dem CALLP-Befehl ausgeführt.	Es wurden mehr als 7 Programmaufrufe mit dem CALLP-Befehl ausgeführt.	Programm prüfen
L3810	Typ des Arguments fehlerhaft	Typ des Arguments fehlerhaft	Korrekten Typ des Arguments festlegen
L3820	Nicht definierter Code	Syntaxfehler bei der Eingabe	Syntax prüfen
L3830	Ausführung des GETM-Befehls nicht möglich	Ausführung des GETM-Befehls nicht möglich	Prüfen, ob der festgelegte Roboter mit einer anderen Anwendung verwendet wird
L3840	Ausführung des RETURN-Befehls ohne GOSUB-Befehl	Ausführung des RETURN-Befehls ohne GOSUB-Befehl	Programm prüfen
L3850	Verwendung einer nicht definierten Palette	Verwendung einer nicht definierten Palette	Palette vor Verwendung definieren
L3860	Fehlerhafte Positionsdaten	Fehlerhafte Positionsdaten	Positionsdaten prüfen
L3870	Der festgelegte Roboter kann nicht definiert werden.	Der festgelegte Roboter kann nicht definiert werden.	Prüfen der Roboternummer
L3880	Die festgelegte Programmnummer ist ungültig.	Die festgelegte Programmnummer ist ungültig.	Korrekte Programmnummer eingeben
L3890	Fehlerhafter Betriebsbefehl	Fehlerhafter Betriebsbefehl	Programm zurücksetzen
L3900	Systemfehler, außerhalb des Überlagerungsbereiches	—	Variablen prüfen
C4000	Zeitüberschreitung	Programmfehler	Programm prüfen
C4100	Zu große Anzahl der registrierten Programme	Zu große Anzahl der registrierten Programme	Nicht verwendete Programme löschen

Tab. A-1: Fehlercodes (16)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
C4110	Zu wenig Speicherplatz	Die Anzahl der Programme und die Datenmenge sind größer als der verfügbare Speicherplatz.	Nicht verwendete Programme oder Daten löschen
C4120	Programmname zu lang	Die Länge des Programmnamens darf höchstens 12 Zeichen und 3 Zusatzzeichen betragen.	Länge des Programmnamens auf einen zulässigen Wert kürzen
C4130	Programmname enthält ein ungültiges Zeichen.	Programmname enthält ein ungültiges Zeichen.	Es dürfen nur Buchstaben und Ziffern verwendet werden.
C4140	Das festgelegte Programm wurde nicht gefunden.	Das festgelegte Programm wurde nicht gefunden.	Gültiges Programm festlegen oder festgelegtes Programm erstellen
C4150	Fehler im Dateiinhalt	Fehler im Dateiinhalt	Datei löschen
C4160	Kein gültiges Roboterprogramm	Das ausgewählte Programm ist kein gültiges Roboterprogramm.	Gültiges Programm auswählen
C4170	Programm wird editiert.	Programm wird editiert.	Editiertes Programm schließen
C4180	Programm wird ausgeführt.	Programm wird ausgeführt.	Programm stoppen
C4190	Programmausführung wird vorbereitet.	Programmausführung wird vorbereitet.	Programm prüfen
C4200	Es können keine Daten in die Datei geschrieben werden.	1: Schreibschutz aktiv oder 2: Speicherplatz zu niedrig	1. Schreibschutz aufheben 2. Nicht verwendete Dateien löschen
C4210	Die eingegebene Anweisung ist zu lang.	Die Länge der Anweisung darf höchstens 127 Zeichen betragen.	Länge der Anweisung auf einen zulässigen Wert kürzen
C4220	Die eingegebene Anweisung weist einen Syntaxfehler auf.	Die eingegebene Anweisung weist einen Syntaxfehler auf.	Die eingegebene Anweisung weist einen Syntaxfehler auf.
C4230	Inhalt oder Syntax fehlerhaft	Inhalt oder Syntax fehlerhaft	Inhalt prüfen und gültige Zeilennummer eingeben
C4240	Schreibgeschütztes Programm	Schreibgeschütztes Programm	Schreibschutz aufheben
C4250	Keine weiteren Zeilen oder Variablen	Die Anzahl der gelesenen Zeilen oder Variablen übersteigt den maximal zulässigen Wert.	Programme prüfen
C4300	Der eingegebene Variablenname ist zu lang.	Die maximale Länge eines Variablennamens beträgt 8 Zeichen.	Variablennamen auf 8 Zeichen kürzen
C4310	Ungültiges Zeichen im Variablen- oder Markennamen	Es wurde ein anderes Zeichen als A bis Z oder 0 bis 9 verwendet.	Keine ungültigen Zeichen verwenden
C4320	Variable ist schreibgeschützt.	Variable ist schreibgeschützt.	1. Verwenden einer speicherbaren Variablen 2. Schreibschutz aufheben
C4330	Es wurde versucht, eine nicht lesbare Variable zu lesen.	Variable ist schreibgeschützt.	1. Verwenden einer speicherbaren Variablen 2. Schreibschutz aufheben
C4340	Nicht definierte Variable	Die Variable ist nicht deklariert.	Variable deklarieren

Tab. A-1: Fehlercodes (17)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
C4350	Doppelte Deklaration einer Variablen	Bereits definierte Variablen können nicht mit der DIM- oder DEF-Anweisung neu definiert werden.	1. Variablenamen ändern und definieren 2. Definierte Variable löschen
C4360	Dieselbe Variable kann nicht öfter als 65535-mal verwendet werden.	Bei 10 P1 = P1 + P2 wird 2-mal auf P1 und 1-mal auf P2 zugegriffen.	Programm so umschreiben, dass die Zahl der Zugriffe auf dieselbe Variable vermindert wird.
C4370	Fehlerhaftes Feldelement	1. Feldelement liegt nicht im zulässigen Bereich. 2. Variable ist kein Feld.	1. Setzen Sie das Feldelement auf einen Wert zwischen 1 und der max. Anzahl der Feldelemente. 2. Verwenden Sie kein Feldelement.
C4380	In einer Anweisung verwendete Variablen können nicht gelöscht werden.	In einer Anweisung verwendete Variablen können nicht gelöscht werden.	Löschen Sie die Anweisung mit der verwendeten Variablen.
C4390	Fehler in der Kombination der Variablentypen	Die benutzerdefinierten globalen Variablen teilen sich in lokale Typen und Basistypen auf.	Variablentypen anpassen
C4400	Fehler in unnummerierten Daten	Fehler in unnummerierten Daten	Programm prüfen und Daten zurücksetzen
C4410	Fehler in unnummerierten Daten	Fehler in unnummerierten Daten	Daten zurücksetzen
C4420	Es dürfen keine Zeilennummern größer als 32767 verwendet werden.	Die neue Zeilennummer oder Zeilennummernlücke überschreitet den zulässigen Wert.	Keine Zeilennummern über 32767 verwenden
C4430	Das gesuchte Zeichen konnte nicht gefunden werden.	Das gesuchte Zeichen konnte nicht gefunden werden.	Programm prüfen
C4440	Die Marke existiert bereits.	Eine bereits definierte Marke kann kein zweites Mal definiert werden.	1. Markennamen ändern und definieren oder 2. Definierten Markennamen löschen
C4800	Das Basisprogramm des Systems kann nicht verwendet werden.	Das Basisprogramm des Systems ist in den Parametern nicht korrekt festgelegt.	Legen Sie das Basisprogramm des Systems korrekt in den Parametern fest.
C4900	Systemfehler	—	MITSUBISHI-Vertriebspartner kontaktieren
H5000	Fehlerhaftes Einschalten des Teach-Modus	Der [ENABLE/DISABLE]-Schalter der Teaching Box wurde im Automatikbetrieb umgeschaltet.	Stellen Sie den [ENABLE/DISABLE]-Schalter der Teaching Box auf DISABLE oder wählen Sie den Teach-Modus.
L5010	Signal zur Freigabe des Automatikbetriebs AUS	Signal der Freigabe des Automatikbetriebs AUS	Signal zur Freigabe des Automatikbetriebs einschalten oder Teach-Modus wählen.
L5100	Für die festgelegte Anwendung wurde kein Programm gewählt.	Für die festgelegte Anwendung wurde kein Programm gewählt.	Programm für die festgelegte Anwendung wählen.
L5110	Kein kontinuierlicher Betrieb möglich	Es wurde ein anderes Programm gewählt.	Wahl des richtigen Programms
L5120	Programm kann nicht gewählt werden.	Die Programmwahl der festgelegten Anwendung ist nicht freigegeben.	Anwendung initialisieren oder Programm zurücksetzen

Tab. A-1: Fehlercodes (18)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
L5130	Servospannung kann nicht eingeschaltet werden.	Der Vorgang Servo AUS ist aktiv.	Warten Sie, bis die Servospannung ausgeschaltet ist und schalten Sie dann die Servospannung ein.
L5140	Lesen nicht möglich	Es wird gerade ein Lese- oder Editierungsvorgang ausgeführt.	Editierte Datei schließen und Daten nach Abschluss des Lesevorgangs lesen
L5150	Keine Ausführung, da Grundposition nicht eingestellt	Grundposition nicht eingestellt	Grundposition einstellen
L5200*	Anzahl der parallel ausgeführten Programme hat den Maximalwert erreicht.	Anzahl der parallel ausgeführten Programme hat den Maximalwert erreicht.	Anzahl der parallel ausgeführten Programme reduzieren
L5210*	Anzahl der definierten Roboter hat den Maximalwert erreicht.	Anzahl der definierten Roboter hat den Maximalwert erreicht.	Anzahl der definierten Roboter reduzieren
L5400	Roboter können nicht definiert werden.	Roboter können nicht definiert werden.	Unabhängige Roboter Nummer definieren
L5410	Ein nicht existierender Modus wurde festgelegt.	Ein nicht existierender Modus wurde festgelegt.	Gültigen Modus festlegen
L5420	Ungültige Festlegung eines Programms	Ungültige Festlegung eines Programms	Gültiges Programm festlegen
L5430	Ungültige Festlegung eines Roboters	Ungültige Festlegung eines Roboters	Gültigen Roboter festlegen
L5600	Keine Ausführung im Fehlerfall möglich	Keine Ausführung im Fehlerfall möglich	Fehler zurücksetzen
C5610	Keine Ausführung bei Eingabe des Stopp-Signals möglich	Keine Ausführung bei Eingabe des Stopp-Signals möglich	Stopp-Signal ausschalten
L5620	Keine Ausführung bei Eingabe des Zyklus-Stopp-Signals möglich	Keine Ausführung bei Eingabe des Zyklus-Stopp-Signals möglich	Zyklus-Stopp-Signal ausschalten
L5630	Keine Ausführung bei Eingabe des Signals Servo OFF möglich	Keine Ausführung bei Eingabe des Signals Servo OFF möglich	Servo OFF-Signal ausschalten
L5640	Keine Ausführung während des Betriebs	Keine Ausführung während des Betriebs	Betrieb stoppen und Ausführung wiederholen
L5650	Keine Ausführung während des Stopp-Vorgangs	Keine Ausführung während des Stopp-Vorgangs	Stopp-Vorgang beenden und Ausführung wiederholen
L6010	Ein ungültiger Befehl wurde gesendet.	Es wurde ein nicht registrierter Befehl gesendet.	Korrekten Befehl eingeben
L6020	Betriebsrecht nicht aktiviert	Es wurden keine Betriebsrechte zugewiesen.	Betriebsrecht zuweisen
L6030	Betriebsrecht zum Editieren nicht aktiviert	Es wurden keine Betriebsrechte zum Editieren zugewiesen.	Betriebsrecht zum Editieren zuweisen
L6040	Ungültige Gerätenummer	Die Gerätenummer ist nicht registriert.	Gültige Gerätenummer angeben
C6050	Die Blockdatei kann nicht geöffnet werden.	Die Blockdatei kann nicht geöffnet werden.	Datei prüfen und korrekte Datei festlegen
C6060	Teach-Modus nicht aktiviert	Teach-Modus nicht aktiviert	Parameter im Teach-Modus schreiben
C6070	Zeit kann nicht eingestellt werden.	Zeit kann nicht eingestellt werden.	Betrieb unterbrechen und Servospannung ausschalten

Tab. A-1: Fehlercodes (19)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
C6500	Die Kommunikationsleitung wurde nicht über das Anwendungsprogramm geöffnet.	Es wurde im Programm kein OPEN-Befehl ausgeführt.	OPEN-Befehl ausführen und dann PRINT-Befehl
L6600	Ungültige Signalnummer	Die angegebene Signalnummer wurde nicht definiert.	Korrekte Signalnummer angeben
L6610	Handsensorsignal kann nicht geschrieben werden.	Ungültiges Signal	Gültiges Signal verwenden
L6620	Eingangssignal kann nicht in den für den Roboter festgelegten Bereich geschrieben werden.	Ungültiges Signal	Gültiges Signal verwenden
L6630	Eingangssignal kann nicht geschrieben werden.	Aktueller Eingangssignalmodus	Pseudo-Eingangssignal setzen
L6640	Fehlerhafte Einstellung eines speziellen Parameters	Parametereinstellung fehlerhaft	Einstellung des Parameters korrigieren
L6650	Doppelte Definition eines speziellen Ausgangssignals	Ungültige Parametereinstellung	Parameter korrigieren
L6651	Doppelte Definition eines speziellen Ausgangssignals und des Parameters HANDTYPE	Ungültige Parametereinstellung	
L6660	Schreiben eines speziellen Ausgangssignals nicht möglich	Fehlerhafte Programmeinstellung	Programm korrigieren
L6670	Ungültiges Bitmuster zum Zurücksetzen der Ausgänge	Die Parameter wurden nicht in Gruppen von 8 gesetzt.	Parameter korrigieren
L6800*	Pseudo-Eingangszustand löschen, Versorgungsspannung ausschalten	Pseudo-Eingangszustand wurde gelöscht	Versorgungsspannung ausschalten
C6900	Start im Pseudo-Eingangssignal-Modus	Parametereinstellungen	Setzen Sie die Parameter bei Verwendung aktueller Signale zurück und schalten Sie die Versorgungsspannung aus und wieder ein.
C7000	Die zu kopierende Quelldatei kann nicht gefunden werden.	Die zu kopierende Quelldatei kann nicht gefunden werden.	Korrekten Dateinamen angeben
C7010	Die zu löschende Zieldatei konnte nicht gefunden werden.	Die zu löschende Zieldatei kann nicht gefunden werden.	
C7020	Die Datei, die umbenannt werden soll, kann nicht gefunden werden.	Die Datei, die umbenannt werden soll, kann nicht gefunden werden.	
H7030*	Parameteränderungsdatei zu groß	Änderungskapazität zu groß	MITSUBISHI-Vertriebspartner kontaktieren
C7040	Änderung des Parameters unzulässig	Änderung des Parameters ist aus Sicherheitsgründen nicht zulässig.	
H7050	Dateifehler	Datei ist fehlerhaft.	Datei nicht verwenden
H7060*	System-RAM-Ressourcen erschöpft	Speicherkapazität wurde überschritten.	MITSUBISHI-Vertriebspartner kontaktieren
C7070	Speicherkapazität erschöpft	Datei überschreitet die Speicherkapazität.	Nicht verwendete Daten oder Dateien löschen
C7500	Keine Batteriespannung	Batterie leer	Batterie austauschen und Daten laden

Tab. A-1: Fehlercodes (20)

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
C7510	Batteriespannung zu niedrig	Batterie leer	Batterie austauschen
C7520	Gebrauchszeit der Batterie überschritten	Batterie leer	
H7600*	Fehlerhafte Roboternummer der Zusatzachse	Die Nummer des Roboters der Zusatzachse ist fehlerhaft.	Einstellung des Parameters AXMENO korrigieren
H7601*	Fehlerhafte Zusatzachsennummer	Die Nummer der Zusatzachse ist fehlerhaft.	Einstellung des Parameters AXJNO korrigieren
H7602*	Doppelte Zusatzachsennummer	Die zugewiesene Zusatzachsennummer wird bereits verwendet.	
H7603*	Fehlerhafte Einheit der Zusatzachse	Die Einheit für die Zusatzachse ist fehlerhaft.	Einstellung des Parameters AXUNT korrigieren
H7604*	Fehlerhafte Beschleunigungszeit der Zusatzachse	Die Beschleunigungszeit der Zusatzachse ist nicht korrekt eingestellt.	Einstellung des Parameters AXACC korrigieren
H7605*	Fehlerhafte Abbremszeit der Zusatzachse	Die Abbremszeit der Zusatzachse ist nicht korrekt eingestellt.	Einstellung des Parameters AXDEC korrigieren
H7606*	Der Nenner zur Festlegung des Übersetzungsverhältnisses der Zusatzachse ist fehlerhaft.	Die Einstellung des Nenners zur Festlegung des Übersetzungsverhältnisses der Zusatzachse ist nicht korrekt.	Einstellung des Parameters AXGRTN korrigieren
H7607*	Der Zähler zur Festlegung des Übersetzungsverhältnisses der Zusatzachse ist fehlerhaft.	Die Einstellung des Zählers zur Festlegung des Übersetzungsverhältnisses der Zusatzachse ist nicht korrekt.	Einstellung des Parameters AXGRTD korrigieren
H7608*	Merker für die Einstellung des Zusatzachsenmotors fehlerhaft	Der Freigabemerker des Zusatzachsenmotors ist nicht korrekt.	Einstellung des Parameters AXMOTSET korrigieren
H7609*	Fehlerhafte Nenngeschwindigkeit der Zusatzachse	Die Nenngeschwindigkeit der Zusatzachse ist nicht korrekt.	Einstellung des Parameters AXMREV korrigieren
H7610*	Fehlerhafte Maximalgeschwindigkeit der Zusatzachse	Die Maximalgeschwindigkeit der Zusatzachse ist nicht korrekt.	Einstellung des Parameters AXJMX korrigieren
H7611*	Fehlerhafte Einstellung der Impulszahl des Encoders	Die Impulszahl des Encoders ist fehlerhaft eingestellt.	Einstellung des Parameters AXENCR korrigieren
H7612*	Fehlerhafte Einstellung der JOG-Zeitkonstante der Zusatzachse	Die Einstellung der Zeitkonstanten der Zusatzachse für den JOG-Betrieb ist nicht korrekt.	Einstellung des Parameters AXJOGTS korrigieren

Tab. A-1: Fehlercodes (21)

Index

A

ACL-Befehl	9-4
ACT-Befehl	9-5
Anweisung	8-1
angehängte	3-37
Aufbau	3-1
Ausgangssignale	3-32
Automatikbetrieb	7-19

B

BASE-Befehl	9-7
Batterie	
anzeigen	7-39
Batteriezüher	
zurücksetzen	7-37
Befehle	
detaillierte Beschreibung	9-4
Übersicht	9-2
Beschleunigungszeit	3-13
Betriebsrechte	2-8
Bremszeit	3-13

C

CALLP-Befehl	9-8
CLOSE-Befehl	9-9
CLR-Befehl	9-10
CMP OFF-Befehl	9-13
CMP POS-Befehl	9-11
CMP TOOL-Befehl	9-12
CMPG-Befehl	9-14
CNT-Befehl	9-15
COM OFF-Befehl	9-17
COM ON-Befehl	9-18
COM STOP-Befehl	9-19

D

Datentypen	8-5
Datum	
einstellen	7-40
DEF ACT-Befehl	9-20
DEF CHAR-Befehl	9-30
DEF DOUBLE-Befehl	9-25
DEF FLOAT-Befehl	9-25
DEF FN-Befehl	9-22
DEF INTE-Befehl	9-25
DEF IO-Befehl	9-26
DEF JNT-Befehl	9-28
DEF PLT-Befehl	9-23
DEF POS-Befehl	9-29
DIM-Befehl	9-31
DLY-Befehl	9-32

E

Ein-/Ausgänge	
externe	6-1
Einführung	1-1
Eingangssignale	3-31
Einschaltzeit	
anzeigen	7-39
END-Befehl	9-34
ERROR-Befehl	9-33

F

Fehler	
Codes	A-2
Diagnose	A-1
Übersicht	A-2
Feinpositionierung	3-16
FINE-Befehl	9-35
FOR-NEXT-Befehl	9-36
FPRM-Befehl	9-38
Funktionen	8-20

G

Gelenkbremsen	
lösen ·	7-38
Gelenk-Interpolation ·	3-4
Geschwindigkeit ·	3-13
GETM-Befehl ·	9-39
GOSUB-Befehl ·	9-40
GOTO-Befehl ·	9-41

H

HCLOSE-Befehl ·	9-43
HLT-Befehl ·	9-42
HOPEN-Befehl ·	9-43

I

IF ... THEN ... ELSE-Befehl ·	9-44
INPUT#-Befehl ·	9-45
Interrupt ·	3-27

J

JOVRD-Befehl ·	9-46
JRC-Befehl ·	9-47

K

Kommunikationsfunktionen ·	3-33
Konstanten	
Einteilung ·	8-6
Gelenkkonstanten ·	8-9
Positionskonstanten ·	8-8
Zeichenkettenkonstanten ·	8-9
Kontinuierliche Bewegung ·	3-11
Kreis-Interpolation ·	3-9

L

LABEL-Befehl ·	9-49
Linear-Interpolation ·	3-6
LOADSET-Befehl ·	9-50
Logische Werte ·	8-19

M

Marken ·	8-2
MELFA-BASIC-IV-Befehle ·	9-1
Menüpunkt	
auswählen ·	7-2
Monitor-Funktion	
Ausgangssignale ·	7-32
Eingangssignale ·	7-31
Fehlermeldungen ·	7-34
Variable ·	7-33
MOV-Befehl ·	9-52
Movement Position-Befehl ·	9-52
Multitask-Funktion ·	4-1
MVC-Befehl ·	9-53
MVR2-Befehl ·	9-56
MVR3-Befehl ·	9-58
MVR-Befehl ·	9-54
MVS-Befehl ·	9-60

O

OADL-Befehl ·	9-62
ON ... GOTO-Befehl ·	9-65
ON COM GOSUB-Befehl ·	9-63
ON-GOSUB-Befehl ·	9-64
OPEN-Befehl ·	9-66
OVRD-Befehl ·	9-68

P

Palettierungsfunktion	3-20
Parallele Ein-/Ausgabeschnittstelle	6-1
Parameter	
einstellen	7-35
spezielle	6-6
Übersicht	5-1
PLT-Befehl	9-69
Positionsdaten	
ändern	7-13
anfahren	7-10
eingeben	7-9
ersetzen	7-12
löschen	7-14
PRINT-Befehl	9-70
Programm	
alle löschen	7-36
Aufbau	3-1
editieren	7-7
erstellen	7-4
kopieren	7-28
löschen	7-30
Namen ändern	7-29
schützen	7-26
testen	7-15
Verzeichnis anzeigen	7-25
Programmebenen	8-25
Programmschleife	3-26

R

RELM-Befehl	9-72,9-73,9-74
REM-Befehl	9-73
Reservierte Wörter	8-25
RETURN-Befehl	9-74
Roboterstatusvariablen	8-16

S

SELECT CASE-Befehl	9-75
SERVO-Befehl	9-77
Servospannung	
ein-/ausschalten	7-17
Sicherheitshinweise	
grundlegende	1-1
SKIP-Befehl	9-78
SPD-Befehl	9-79
Steuergerät	
Bedien- und Signalelemente	2-1
Stopp	3-30
SUBSTITUTE-Befehl	9-90

T

Teaching Box	
Bedienelemente	2-5
Bedienung	7-1
Display-Anzeigen	2-3
Funktionen	2-3
Timer	3-29
TOOL-Befehl	9-80
TORQ-Befehl	9-81

U

Uhrzeit	
einstellen	7-40
Unterprogramm	3-28

V

Variablen	
arithmetische	8-11
Ein-/Ausgangsvariablen	8-13
Einteilung	8-10
externe	8-15
Feldvariablen	8-14
Gelenkvariablen	8-13
Positionsvariablen	8-12
Roboterstatusvariablen	8-16
Zeichenkettenvariablen	8-12
Verzweigungen	3-24

W

WAIT-Befehl	9-82
Wartezeit	3-24
WHILE ~ WEND-Befehl	9-83
WTH-Befehl	9-84
WTHIF-Befehl	9-85

X

XLOAD-Befehl	9-86
XRST-Befehl	9-89
XRUN-Befehl	9-87
XSTP-Befehl	9-88

Z

Zeichen	
mit besonderer Bedeutung	8-4
Typen	8-3
Zeilen	8-2
Zeilennummern	8-2
Zeitablaufdiagramm	6-13

HEADQUARTERS

MITSUBISHI ELECTRIC EUROPE B.V.
 German Branch
 Gothaer Straße 8
 D-40880 Ratingen
 Telefon: +49 (0) 21 02 / 486-0
 Telefax: +49 (0) 21 02 / 4 86-1 12
 E-Mail: megfa-mail@meg.mee.com

MITSUBISHI ELECTRIC EUROPE B.V.
 FRANCE
 25, Boulevard des Bouvets
 F-92741 Nanterre Cedex
 Telefon: +33 1 55 68 55 68
 Telefax: +33 1 49 01 07 25
 E-Mail: factory.automation@fra.mee.com

MITSUBISHI ELECTRIC EUROPE B.V.
 Italian Branch
 C.D. Colleoni - P. Perseo Ing. 2
 Via Paracelso 12
 I-20041 Agrate Brianza (MI)
 Telefon: +39 (0) 39 / 60 53 1
 Telefax: +39 (0) 39 / 60 53 312
 E-Mail: factory.automation@it.mee.com

MITSUBISHI ELECTRIC EUROPE B.V.
 Pol. Ind. Can Magi-C.
 Calle Joan Buscallá, 2-4 AC 420
 E-08190 Sant Cugat del Vallés
 Telefon: +34 (9) 3 / 565 31 31
 Telefax: +34 (9) 3 / 589 29 48

MITSUBISHI ELECTRIC EUROPE B.V.
 UK Branch
 Travellers Lane
 GB-Hatfield Herts. AL10 8 XB
 Telefon: +44 (0) 1707 / 27 61 00
 Telefax: +44 (0) 1707 / 27 86 95

MITSUBISHI ELECTRIC CORPORATION
 Mitsubishi Denki Bldg.
 2-2-3 Marunouchi Chiyoda-Ku
 Tokyo 100-8310
 Telefon: +81 (0) 3 / 32 18 31 76
 Telefax: +81 (0) 3 / 32 18 24 22

MITSUBISHI ELECTRIC AUTOMATION
 500 Corporate Woods Parkway
 Vernon Hills, Illinois 60061
 Telefon: +1 (0) 847 / 478 21 00
 Telefax: +1 (0) 847 / 478 22 83

**EUROPÄISCHE
 VERTRETUNGEN**

Belgien
 Getronics b.v.
 Control Systems
 Pontbeeklaan 43
 B-1731 Asse-Zellik
 Telefon: +32 (0) 2 / 4 67 17 51
 Telefax: +32 (0) 2 / 4 67 17 45
 E-Mail: infoautomation@getronics.com

Dänemark
 Louis Poulsen
 Geminivej 32
 DK-2670 Greve
 Telefon: +45 (0) 43 / 95 95 95
 Telefax: +45 (0) 43 / 95 95 91
 E-Mail: lpia@lpmail.com

Finland
 Beijer Electronics OY
 Elanontie 5
 FIN-01510 Vantaa
 Telefon: +358 (0) 9 / 615 20 11
 Telefax: +358 (0) 9 / 615 20 500
 E-Mail: info@beijer.fi

Irland
 MITSUBISHI ELECTRIC EUROPE B.V. - Irish Branch
 Westgate Business Park
 IRL-Dublin 24
 Telefon: +353 (0) 1 / 419 88 00
 Telefax: +353 (0) 1 / 419 88 90
 E-mail: sales.info@meuk.mee.com

Niederlande
 Getronics
 Industrial Automation B.V.
 Control Systems
 Donauweg 10
 NL-1043 AJ-Amsterdam
 Telefon: +31 (0) 20 / 586 15 92
 Telefax: +31 (0) 20 / 586 19 27
 E-Mail: infoautomation@getronics.com

Norwegen
 Beijer Electronics AS
 Teglverksveien 1
 N-3002 Drammen
 Telefon: +47 (0) 32 / 24 30 00
 Telefax: +47 (0) 32 / 84 85 77
 E-Mail: —

**EUROPÄISCHE
 VERTRETUNGEN**

Österreich
 GEVA GmbH
 Wiener Straße 89
 A-2500 Baden
 Telefon: +43 (0) 2252 / 85 55 20
 Telefax: +43 (0) 2252 / 488 60
 E-Mail: office@geva.co.at

Polen
 MPL Technology SP. z.o.o
 ul. Wroclawska 53
 PL-30011 Kraków
 Telefon: +48 (0) 12 / 632 28 85
 Telefax: +48 (0) 12 / 632 47 82
 E-Mail: mpl@krakow.ipl.net

Rumänien
 Sirius
 Trading & Services srl
 Bd. Ghica nr. 112, Bl. 41
 RO-72235 Bucaresti 2
 Telefon: +40 (0) 1 / 210 55 11
 Telefax: +40 (0) 1 / 210 55 11
 E-Mail: sirius_t_s@fx.ro

Schweden
 Beijer Electronics AB
 Box 325
 S-20123 Malmö
 Telefon: +46 (0) 40 / 35 86 00
 Telefax: +46 (0) 40 / 93 23 01
 E-Mail: info@elc.beijer.se

Schweiz
 ECONOTEC AG
 Postfach 282
 CH-8309 Nürensdorf
 Telefon: +41 (0) 1 / 838 48 11
 Telefax: +41 (0) 1 / 838 48 12
 E-Mail: info@econotec.ch

Slowenien
 INEA d.o.o.
 Ljubljanska 80
 SI-61230 Domžale
 Telefon: +386 (0) 1 / 721 80 00
 Telefax: +386 (0) 1 / 724 16 72
 E-Mail: inea@inea.si

Tschechien
 AUTOCONT s.r.o.
 Nemocnici 12
 CZ-70100 Ostrava 1
 Telefon: +420 (0) 69 / 615 21 11
 Telefax: +420 (0) 69 / 615 21 12
 E-Mail: petr.pustovka@autocont.cz

Türkei
 GTS
 Darülaceze Cad. No. 43A KAT: 2
 TR-80270 Okmeydanı-Istanbul
 Telefon: +90 (0) 212 / 320 1640
 Telefax: +90 (0) 212 / 320 1649
 E-Mail: gts@turk.net

**VERKAUFSBÜROS
 DEUTSCHLAND**

MITSUBISHI ELECTRIC EUROPE B.V.
 DGZ-Ring Nr. 7
 D-13086 Berlin
 Telefon: (0 30) 4 71 05 32
 Telefax: (0 30) 4 71 54 71

MITSUBISHI ELECTRIC EUROPE B.V.
 Revierstraße 5
 D-44379 Dortmund
 Telefon: (02 31) 96 70 41-0
 Telefax: (02 31) 96 70 41-41

MITSUBISHI ELECTRIC EUROPE B.V.
 Brunnenweg 7
 D-64331 Weiterstadt
 Telefon: (0 61 50) 13 99 0
 Telefax: (0 61 50) 13 99 99

MITSUBISHI ELECTRIC EUROPE B.V.
 Kurze Straße 40
 D-70794 Filderstadt-Bonlanden
 Telefon: (07 11) 77 05 98-0
 Telefax: (07 11) 77 05 98-79

MITSUBISHI ELECTRIC EUROPE B.V.
 Am Söldnermoos 8
 D-85399 Hallbergmoos
 Telefon: (08 11) 99 87 4-0
 Telefax: (08 11) 99 87 4-10

**VERTRETUNG
 MITTLERER OSTEN**

ILAN & GAVISH LTD
 Automation Service
 24 Shenkar St., Qiryat-Arie 49513
 IL-49001 Petach-Tikva
 Telefon: +972 (0) 3 / 922 18 24
 Telefax: +972 (0) 3 / 972 39 24 07 61
 E-Mail: iandg@internet-zahav.net

VERTRETUNGEN EURASIEN

MITSUBISHI ELECTRIC EUROPE B.V.
 12/1 Goncharnaya St, suite 3C
 RUS-109240 Moskow
 Telefon: +7 (0) 95 / 915-8602
 Telefax: +7 (0) 95 / 915-8603