

PROGRAMMIERHANDBUCH
HARL-U1 Ver. 2

Hirata

Copyright 1995 by HIRATA Robotics GmbH

Am Sägewerk 7 D-55124 Mainz

Alle Rechte vorbehalten. Kein Teil des Handbuches darf in irgendeiner Form (Druck, Fotokopie oder einem anderen Verfahren) ohne schriftliche Genehmigung der HIRATA Robotics GmbH reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

INHALTSVERZEICHNIS

Kapitel	Inhalt	Seite
1	Allgemeines	1
2	Grundsätzliche Funktionen	2
2.1	Programmstruktur	2
2.2	Speicher und Zeitgeber	3
3	Bedienungsanleitung	4
3.1	Hauptmenü	4
3.2	Programmnummer und Programmschritt	5
3.3	Eingeben oder Verändern von Programmen	6
3.4	Setzen von Zeitgebern	8
3.5	Offset-Tabelle	8
3.6	Programminitialisierung	9
3.7	Kopieren von Programmen	11
3.8	Eingabe von bewegungsrelevanter Daten	12
3.9	Info	14
3.10	Palettierung	14
3.10.1	Palettierdaten	14
3.10.2	Palettentabelle	15
3.11	Handbetrieb	16
3.12	Programmüberprüfung	17
3.13	Sichern und Laden von Daten mittels Memory-Karte	19
3.14	Eingänge lesen / Ausgänge setzen	23
3.15	Aufzeichnung des Programmablaufes	24
4.	Ein- und Ausgangssignale	25
4.1	Eingangssignale	25
4.2	Ausgangssignale	27
4.3	Ein-/Ausgangserweiterung	29

INHALTSVERZEICHNIS

Kapitel	Inhalt	Seite
5	Fehlermeldungen	30
6	Befehlsliste	31
7	Beschreibung der Befehle	33
8	Programmbeispiel	82
9	Zeitdiagramme der Ansteuerung	87
10	Anhang	89
10.1	Ein- und Ausgangssignalverarbeitung während der Roboterbewegung	89
10.2	Externe Positionsadressierung	91
10.3	Erweiterte Roboterfunktionen	92
10.3.1	Positionskalibrierung durch Visionsystem	92
10.3.2	Positionsverschiebung durch Offset-Tabelle	92
10.3.3	Bewegungsänderung	93
10.4	Anwendungsbeispiel	95
10.4.1	Beschreibung des Bewegungsablaufes	95
10.4.2	Ein- und Ausgangsbelegung	95
10.4.3	Flußdiagramm	96
10.4.4	Beispielprogramm	97
10.5	Anwendung des R MOV-Befehls	100
10.5.1	Warnung - Wichtige Hinweise zur Benutzung	100
10.5.2	Programmbeispiel	101

1 Allgemeines

Dieses Handbuch beschreibt die Roboterprogrammiersprache HARL-U1, Hirata Assembly Robot Language U1.

HARL-U1 ermöglicht die Steuerung von Aktuatoren und Initiatoren, die in dem direkten Umfeld des Roboters sind, um zum Beispiel einfache Pick&Place-Aufgaben durchzuführen.

Mit Hilfe des Teachpults wird das HARL-U1-Programm eingegeben oder geändert. Auch die OFFLINE-Programmierung über PC ist möglich.

Die Beschreibung der Bedienungs- und Eingabeprozedur für Positionen wird im Operationhandbuch beschrieben.

Die Beschreibung der Ansteuerung und die dazugehörige Schnittstellenbeschreibung befindet sich jedoch in diesem Handbuch.

2. Grundfunktionen

2.1 Programmstruktur

In der Programmiersprache gibt es drei verschiedene Programmtypen, Haupt-, Unter- und Handprogramm. Jedes Programm besteht aus Programmschritten mit Schrittnummern, Befehlen und Daten.

(1) Hauptprogramm

Das Hauptprogramm beschreibt den Bewegungsablauf des Roboters und die Steuerung der Peripherie wie Greifer, Zuführungen usw.

Die maximale Anzahl von Hauptprogrammen beträgt 16 (00 - 15) und die maximale Schrittzahl pro Hauptprogramm beträgt 999.

Über Eingänge kann das entsprechende Programm angewählt und gestartet werden. Die Steuerung ist ein Single-Task-System, d.h., es kann nur ein Programm nicht zwei oder mehrere Programme gleichzeitig ausgeführt werden.

Wird das HARL-U1-Programm gestartet, beginnt immer zuerst das Hauptprogramm zu arbeiten.

(2) Unterprogramm

Immer wiederkehrende und gleiche Ablaufrouinen eines Programmes können als Unterprogramme geschrieben werden. Diese Unterprogramme werden aus dem Hauptprogramm aufgerufen und es wird dadurch Speicherplatz gespart.

Die maximale Anzahl von Unterprogrammen beträgt 99 und die maximale Schrittzahl pro Unterprogramm beträgt 999.

(3) Handprogramm

Das Handprogramm ist eine Art Unterprogramm, mit dem Sie nur Aktuatoren (Ausgänge) des Roboters z.B. Greifer steuern sollten. Dadurch brauchen Sie diese Steuerbefehle nicht im Hauptprogramm programmieren.

Die Handprogramme werden vom Hauptprogramm aus aufgerufen.

Der Unterschied zwischen Hand- und Unterprogramm besteht darin, daß Sie Handprogramme von dem Teachpult aus aufrufen und ausführen können.

Die maximale Anzahl von Handprogrammen beträgt 99 und die maximale Schrittzahl pro Handprogramm beträgt 999.

Hinweise

1. Jedes Hauptprogramm ist unabhängig von den anderen, sie beeinflussen sich nicht gegenseitig. Unter- und Handprogramme jedoch können gemeinschaftlich von allen Hauptprogrammen aus aufgerufen werden.
2. Die maximale Anzahl von Programmschritten von allen Programmen (Haupt-, Unter- und Handprogramm) beträgt 6.500.
Bevor Sie mit der Eingabe von Programmen beginnen, definieren Sie die Parameter für
MAIN, SUB und HAND gemäß Kapitel 3.2.

2.2 Speicher und Zeitgeber

(1) Datenregister

Es gibt 16 Datenregister (00 - 15). Jedes Register kann Werte zwischen 0 und 999 speichern.

Werden Befehle benutzt, in denen Datenregister vorkommen, können arithmetische Berechnungen (Addition, Subtraktion), Werteerhöhung/-verringern oder -vergleich angewendet werden.

Wird einmal ein Wert in ein Datenregister eingespeichert, wird dieser solange gehalten, bis ein neuer Wert programmiert wird.

Die Daten bleiben auch bei ausgeschalteter Steuerung erhalten.

(2) Bitspeicher

Die Programmiersprache HARL-U1 verfügt über 32 Bitspeicher (Adressierung 32 - 63), in denen der Wert 0 oder 1 gespeichert werden kann.

Wird einmal ein Wert in ein Datenregister eingespeichert, wird dieser solange gehalten, bis ein neuer Wert programmiert wird.

Der Bitspeicher 63 wird jedoch zur Speicherung des Interrupt-Signals benutzt.

Dadurch wechselt der Zustand dieses Speicher entsprechend des Interrupt-Eingangs.

Die Daten bleiben auch bei ausgeschalteter Steuerung erhalten.

(3) Zeitgeber (Timer)

Die Programmiersprache HARL-U1 verfügt über 16 Zeitgeber (Adressierung 00 - 15), die einen Zeitbereich von 0,01 bis 9,99 Sekunden haben können.

Der Wert eines jeden Zeitgebers muß am Anfang in dem entsprechenden Untermenüpunkt (Kapitel 3.4) gesetzt werden.

3 Bedienungsanleitung

3.1 Hauptmenü

Durch gleichzeitiges Drücken der Tasten "FUNC" und "seq" in den Betriebsarten **KEY-IN** oder **RO-TEACH** oder **LI-TEACH** auf dem Teachpult, wechselt die Anzeige auf nachfolgendes Bild (Bild 3.1).

Die Anzeige zeigt das Hauptmenü, im dem Sie die entsprechende Funktion über die numerischen Tasten "1" bis "5" anwählen können.

S E Q U E N C E M E N U	
1 . E D I T	2 . M A N U A L
3 . U T I L	4 . S Y S T E M
5 . P A L L E T	

Bild 3.1 Hauptmenü

Die Hauptmenüpunkte haben folgende Untermenüpunkte

1. EDIT

- | | | |
|----|--------------|---|
| 1. | MAIN | Eingeben oder Verändern von Hauptprogrammen |
| 2. | SUB | Eingeben oder Verändern von Unterprogrammen |
| 3. | HAND | Eingeben oder Verändern von Handprogrammen |
| 4. | TIMER | Eingeben oder Verändern von Zeitgeberwerten |
| 5. | TABLE | Eingeben oder Verändern von Offset-Tabellen |

2. MANUAL Handbetrieb: Ausführen der Handprogramme

3. UTIL

- | | | |
|----|-------------|--|
| 1. | INIT | Anwahl der Programmnummer durch die Taste "dec/INC |
| 2. | COPY | Kopieren von Programmen und Initialisierungsdaten |
| 3. | DATA | Eingabe von Erweiterungsdaten |
| 4. | INFO | Anzeige der HARL-U1 Version und Typ. |

4. SYSTEM Festlegung der Programmnummer und -anfang (Startschritt)

5. PALLET

- | | | |
|----|---------------------|---|
| 1. | PALLET DATA | Eingabe der Palettierdaten |
| 2. | PALLET TABLE | Eingabe der Daten für den PALLET-Befehl |

3.2 Programmnummer und Programmschritt

Die Programmnummer und die Programmschritte für die Haupt-, Unter- und Handprogramme sowie deren Zuordnung können Sie innerhalb der erlaubten Speicherbereiche frei wählen.

< Wie werden die Programmnummern und deren Anfangsprogrammschritte geordnet? >

1. Drücken Sie Taste "4" im Hauptmenü, um den Menüpunkt SYSTEM anzuwählen.
2. Geben Sie die Daten der folgenden Parameter ein.

MAIN PROGRAM MAX	Größe Programmnummer für Hauptprogramme
MAIN STEP MAX	Größe Anzahl von Schritten für Hauptprogramm
SUB PROGRAM MAX	Größe Programmnummer für Unterprogramme
SUB STEP MAX	Größe Anzahl von Schritten für Unterprogramm
HAND PROGRAM MAX	Größe Programmnummer für Handprogramme
HAND STEP MAX	Größe Anzahl von Schritten für Handprogramm
3. Nachdem Sie die Daten eingegeben haben, drücken Sie die Taste "SHIFT", so daß die LED in der Taste leuchtet. Dann drücken Sie die Tasten "FUNC" und "READ" gleichzeitig. Dann wird "SYSTEM SET OK?" angezeigt. Drücken Sie jetzt die Taste "ENTER", um die korrekte Dateneingabe und Übernahme in den Speicher zu bestätigen.
4. Wenn Sie die Taste "F1" drücken, werden die momentan gespeicherten Werte angezeigt. Drücken Sie irgendeine Taste, um zur ursprünglichen Anzeige zurückzukehren.

Maximale Speicherbegrenzungen

MAIN SIZE	4096	Maximale Größe für Hauptprogramme
SUB SIZE	1000	Maximale Größe für Unterprogramme
HAND SIZE	1000	Maximale Größe für Handprogramme
TOTAL SIZE	6096/6500	Maximale Größe für alle Programme zusammen

Hinweise

- | | | |
|----|---|-------|
| 1. | Bevor Sie mit der Eingabe eines Programmes beginnen, definieren Sie zuerst die Programmnummern und die Programmschritte. | |
| 2. | Die höchste Programmnummer für Unter- und Handprogramme ist 99 und die größte Anzahl von Programmschritten ist 999. Die höchste Programmnummer für Hauptprogramme ist 15. | |
| 3. | Die maximale Anzahl von allen Programmschritten zusammen ist 6500. | |
| 4. | Werkseitig programmierte Werte sind: | |
| | Hauptprogramme | : 15 |
| | Unter- und Handprogramme | : 20 |
| | Größe Anzahl Programmschritte Hauptprogramm | : 256 |
| | Größe Anzahl Programmschritte Unter- Handprg. | : 50 |

3.3 Eingeben oder Verändern von Programmen

- (1) Drücken Sie die Taste "1" (EDIT) im Hauptmenü. Dann erscheint folgende Anzeige.

S E Q U E N C E		E D I T	
1 .	M A I N	2 .	S U B
3 .	H A N D	4 .	T I M E R
5 .	T A B L E		

Bild 3.2 Programmeingabemenü

- (2) Drücken Sie jetzt die Taste "1" (MAIN), "2" (SUB) oer "3" (HAND). Danach erscheint Bild 3.3

P R O G R A M		S E L E C T	
M A I N	P R O G R A M	N O :	0 0
P u s h E N D k e y			

Bild 3.3 Programmnummerauswahl bei Taste "1" (MAIN)

- (3) Geben Sie die gewünschte Programmnummer ein und drücken Sie die Taste "END". Das entsprechende Programm wird geladen und der Anfang des Programmes angezeigt.

M A I N	P R O G R A M	0 1
0 0 0	E N D	
0 0 1	E N D	
0 0 2	E N D	

Bild 3.4 Anzeige des Programmanfangs des Hauptprogramms 1

- (4) Wenn Sie einen neuen Befehl eingeben wollen, muß der Cursor mit der Taste "UP/DOWN" auf die entsprechende Programmzeile bewegt werden.
- (5) Drücken Sie die Taste "i.O./SEL", um die Befehlstabelle anzuwählen. Die Befehlstabelle besteht aus sechs Seiten. Durch Drücken der Taste "i.O./SEL" wird immer um eine Seite weitergeblättert
- (6) Wählen Sie den entsprechenden Befehl mit Hilfe der numerischen Tasten "1" bis "9" aus.

COMMAND LIST 1 / 6		
1 . NOP	2 . END	3 . STOP
4 . TIM	5 . JUMP	6 . BRN 0
7 . BRN 1	8 . BRN T	9 . PCHK

Bild 3.5 Seite 1 der Befehlstabelle

- (7) Wird ein Befehl angewählt, bei dem die Eingabe eines Wertes erforderlich ist, springt der Cursor nach Anwähl des Befehls an die Stelle, an der der Wert eingegeben werden muß. Danach ist der Wert mit Hilfe der numerischen Tasten einzugeben.
- (8) Nachdem alle Eingaben gemacht worden sind, drücken Sie die Taste "ENTER" einmal. Danach erscheint die Anzeige "ENTER OK?". Durch erneutes Drücken der Taste "ENTER" wird die korrekte Eingabe bestätigt und in den Speicher übernommen. Danach können Sie den nächsten Programmschritt eingeben.
- (9) Die Eingabe des nächsten Befehls erfolgt nun in gleicher Weise wie Schritt 4 bis 8.
- (10) Am Ende eines Programmes muß immer der END-Befehl eingegeben werden.

Hinweis

- | |
|--|
| <ol style="list-style-type: none"> 1. Bei der Eingabe von Unter- und Handprogrammen verfahren Sie analog. 2. Wird die Taste "spd/CAN" gedrückt, springt die Anzeige in das vorherige Menü. |
|--|

3.4 Setzen von Zeitgebern

- (1) Drücken Sie die Taste "4" (TIMER) im Menüpunkt EDIT, um die Zeitgeber programmieren zu können.

Zeitgebernummer : TIM0 bis TIM15

Zeitbereich : 0.01 s bis 9.99 s

T I M E R E D I T	
T I M 0 0	1 . 0 0
T I M 0 1	0 . 0 5
T I M 0 2	0 . 1 5

Zeitgeber programmierte Zeit

Bild 3.6 Eingabemenü für Zeitgeber

- (2) Bewegen Sie den Kursors zum entsprechenden Zeitgeber durch Drücken der Taste "up/DOWN".
- (3) Geben Sie die gewünschte Zeit mit Hilfe der numerischen Tastatur ein.
- (4) Sind alle Eingaben gemacht worden, drücken Sie einmal die Taste "ENTER". Danach erscheint die Anzeige "ENTER OK?". Durch erneutes Drücken der Taste "ENTER" wird die korrekte Eingabe bestätigt und in den Speicher übernommen. Danach wird der nächste Zeitgeber angezeigt.

3.5 OFFSET-Tabelle

Die Werte der OFFSET-Tabelle werden benutzt, wenn der Roboter durch den Befehl "R_MOV" oder durch den optionalen Wert "2" eines Befehls bewegt wird. Lesen Sie bitte in Kapitel 10 über die Verwendung der OFFSET-Tabelle.

Nachfolgend wird beschrieben, wie Daten in die OFFSET-Tabelle eingegeben werden können.

- (1) Drücken Sie die Taste "5" (TABLE) im Menüpunkt EDIT. Danach wird Bild 3.7 angezeigt.

O f f s e t T a b l e N o . 0 0			
A	0 0 0 0 . 0 0	B	0 0 0 0 . 0 0
Z	0 0 0 0 . 0 0	W	0 0 0 0 . 0 0

Bild 3.7 OFFSET-Tabelle

- (2) Wählen Sie die gewünschte OFFSET-Tabellennummer (00 - 99) mit Hilfe der numerischen Tasten oder der Taste "dec/INC" aus.
- (3) Bewegen Sie den Cursor mit Hilfe der Taste "up/DOWN" zur gewünschten Koordinaten oder drücken Sie die entsprechende Achs-Tipptaste und geben Sie dann den Wert ein.
- (4) Sind alle Eingaben gemacht worden, drücken Sie einmal die Taste "ENTER". Danach erscheint die Anzeige "ENTER OK?". Durch erneutes Drücken der Taste "ENTER" wird die korrekte Eingabe bestätigt und in den Speicher übernommen.

3.6 Programminitialisierung

Drücken Sie die Taste "3" (UTIL) im Hauptmenü, um den Untermenüpunkt UTIL aufzurufen. Danach erscheint Bild 3.8

S E Q U E N C E		U T I L I T Y	
1 .	I N I T	2 .	C O P Y
3 .	D A T A	4 .	I N F O

Bild 3.8 UTILITY Menü

- (1) Drücken Sie die Taste "1" (INIT) im Untermenü UTIL. Danach wird Bild 3.9 aufgerufen.

S E Q U E N C E		I N I T	
T Y P E		[M A I N P R O G]	
P R O G . N O .		0 0	
S T A R T	S T E P	0 0 0	
E N D		S T E P	0 0 0
P u s h E N D K e y			

Bild 3.9 Programminitialisierung

- (2) Wählen Sie die Programm- oder Datenart an, die initialisiert werden soll, in dem Sie die Taste "io/SEL" oder numerische Taste "0" bis "7" drücken. Die Zuordnung der numerischen Tasten ist:

Anzeige	Taste	Beschreibung
Main Prog.	"0"	Hauptprogramm
Sub Prog.	"1"	Unterprogramm
Hand Prog.	"2"	Handprogramm
Timer Data	"3"	Zeitgeberwert
Reg. Data	"4"	Datenregisterwert
Memory Bit	"5"	Merkerwert
Offset Table	"6"	Offset-Tabelle
All Data	"7"	Alle Programme und Daten

- (3) Eingeben der Nummer des Programmes bei PROG.NO., das initialisiert werden soll. Wenn ein Zeitgeber, Datenregister, Merker oder alle Daten initialisiert werden sollen, ist die Programmnummer PROG. NO. ungültig.
- (4) Bei einem Programm muß zuerst die Startadresse bei START STEP eingegeben werden.
Bei Zeitgebern, Datenregistern oder Merkern ist die Anfangsnummer einzugeben.
Bei der Initialisierung aller Daten ist jeder Wert ungültig, der bei START STEP eingegeben wird.

- (5) Bei einem Programm muß der letzte Programmschritt bei END STEP eingegeben werden.
Bei Zeitgebern, Datenregistern oder Merkern ist die Endnummer einzugeben.

Bei der Initialisierung aller Daten ist jeder Wert ungültig, der bei END STEP eingegeben wird.
- (6) Nach der Eingabe der Anfangs- und Endwerte ist die Taste "END" zu drücken. Dann wird "INIT OK?" angezeigt. Sind alle Werte für die Initialisierung korrekt, ist die Taste "ENTER" zu drücken. Danach beginnt die Initialisierung.
- (7) Ist die Initialisierung beendet, wird "COMPLETED" angezeigt.

Hinweis

Wenn ein Programm initialisiert wurde, steht in jeder Programmzeile der END-Befehl.
Wenn Speicher initialisiert wurden, sind die Werte auf Null gesetzt.

Achtung

Wenn "ALL DATA" angewählt wurde, werden alle Programme und Daten initialisiert.

3.7 Kopieren von Programmen

- (1) Drücken Sie die Taste "2" (COPY) im Untermenü UTIL. Danach wird Bild 3.10 aufgerufen.

S E Q U E N C E C O P Y			
S	P R O G . T Y P E	[M A I N]	—> Quellprogramm (Source) Typ, von dem kopiert wird
S	P R O G . N O .	0 0	—> Quellprogrammnummer von der kopiert wird
S	S T A R T S T E P	0 0 0	—> Quellprogramm Schrittnummer ab der kopiert wird
S	E N D S T E P	0 0 0	—> Quellprogramm Schrittnummer bis zu der kopiert wird
D	P R O G . T Y P E	[M A I N]	—> Zielprogramm (Destination) Typ, in den kopiert wird
D	P R O G . N O .	0 0	—> Zielprogramm Schrittnummer in die kopiert wird
D	S E T S T E P	0 0 0	—> Zielprogramm Schrittnummer bis zu der kopiert wird
P u s h E N D K e y			
- - - - -			

Bild 3.10 Programm kopieren

- (2) Wählen Sie die Parameter durch Drücken der Taste "up/DOWN" an.
- 1) *S PROG. TYPE*
Anwahl des Programmtyps, aus der kopiert werden soll.
MAIN/SUB/HAND kann durch die Taste "io/SEL" oder "0", "1", "2" angewählt werden.
 - 2) *S PROG. NO.*
Anwahl der Programmnummer, aus der kopiert werden soll.
 - 3) *START STEP*
Startadresse, ab der kopiert werden soll
 - 4) *END STEP*
Endadresse, bis zu der kopiert werden soll
 - 5) *D PROG. TYPE*
Anwahl der Programmtyp, in die kopiert werden soll.
 - 6) *D PROG. NO.*
Anwahl der Programmnummer, in die kopiert werden soll.
 - 7) *D SET STEP*
Anfangsadresse auf die kopiert werden soll.
- (3) Nach der Eingabe aller Werte drücken Sie die Taste "END" .
Dann wird "COPY OK?" angezeigt. Sind alle Werte für den Kopiervorgang korrekt, drücken Sie die Taste "ENTER" . Danach wird der Kopiervorgang ausgeführt.

3.8 Eingabe von bewegungsrelevanter Daten des Roboters

Weitere Parameter mit bewegungsrelevantem Charakter sind in Kapitel 10.3 beschrieben.

- (1) Drücken Sie die Taste "2" (COPY) im Untermenü UTIL. Danach wird Bild 3.10 aufgerufen.

EXPAND DATA SET	
PULL LIMIT	0010.00
ARCH MOTION	ON
TABLE DATA	00

ZUPDIS1	0000.00
ZDNDIS1	0000.00
ZUPDIS2	0000.00
ZDNDIS2	0000.00
ZUPDIS3	0000.00
ZDNDIS3	0000.00
PULLRATE	0000.00
I/O STATUS	0000.00
-----bottom-----	

Bild 3.11 Tabelle bewegungsrelevanter Parameter

1) PULL LIMIT

Bestimmt die untere Grenze des PULL-UP-Wertes.

Dieser Wert wird zur Bewegungssteuerung des Roboters herangezogen, wenn der optionale Wert (Data) eines Verfabrbefehls (MOVE command) 04 ist.

Bei der Funktion ZUP DIS wird dieser Wert als Pull-up-Wert der Z-Achse benutzt. Soll diese Funktion nicht benutzt werden, setzen Sie den Wert auf den größten Z-Achsenwert, der angefahren wird.

2) ARCH MOTION (Bogenbewegung)

Dieser Wert wird zur Bewegungssteuerung des Roboters herangezogen, wenn der optionale Wert (Data) eines Verfabrbefehls (MOVE command) 04 ist und eine Bogenbewegung ausgeführt werden soll oder nicht (ON=ja / OFF=nein).

3) TABLE DATA

Dieser Wert wird zur Bewegungssteuerung des Roboters herangezogen, wenn der optionale Wert (Data) eines Verfabrbefehls (MOVE command) 02 ist.

Der Wert spezifiziert welche Tabelle zur Positionierung benutzt werden soll.

- 4) ZUP DIS 1 - 3 (Z-Achsen aufwärts Abstand)
 ZDN DIS 1 - 3 (Z-Achsen abwärts Abstand)

Dieser Wert wird zur Bewegungssteuerung des Roboters herangezogen, wenn der optionale Wert (Data) eines Verfahrbefehls (MOVE command) 04 ist. Die Z-Achse bewegt sich von der momentanen Position aus um den dort eingegebenen Betrag nach oben oder unten. Drei unterschiedliche Werte können gespeichert und mit dem entsprechenden Parameter aufgerufen werden.

Optionaler Wert	Parameter
14	ZUP/ZDN DIS 1
24	ZUP/ZDN DIS 2
34	ZUP/ZDN DIS 3

- 5) PULL RATE

Dieser Wert wird zur Bewegungssteuerung des Roboters herangezogen, wenn der optionale Wert (Data) eines Verfahrbefehls (MOVE command) 04 ist. Mit diesem Parameter kann der interne Pull-Up-Wert gesetzt werden, indem der ZUP/DN DIS vergrößert wird. Der Wert dieses Parameters kann zwischen 1.0 und 2.0 liegen.

- 6) I/O STATUS

Mit diesem Parameter wird festgelegt, ob die Ausgänge im Fall eines NOTAUS während des Automatikbetriebes zurückgesetzt werden oder nicht.

I/O Status	Bedeutung
0	Ausgangszustand bleibt erhalten
1	Ausgangszustand wird zurückgesetzt

3.9 INFO

- (1) Drücken Sie die Taste "4" (INFO) im Untermenü UTIL, danach wird die HARL-U1 Softwareversion und der Softwaretyp angezeigt.

3.10 Palettierung

3.10.1 Palettierdaten

Durch eine optionale Berechnungssoftware können die Palettierdaten mittels eines Computers in die Steuerung übertragen werden. Drücken Sie die Taste "1" (PALLET DATA) nachdem Sie die Taste "5" (PALLET) im Hauptmenü gedrückt haben, um die Steuerung zum Datenempfang vorzubereiten.

Nach der Übertragung sollten Sie folgende Daten auf ihre Richtigkeit überprüfen: TEACHING ADDRESS, OPEN ADDRESS, WORK STEP, M DATA, F CODE, S CODE und Werte des X,Y,Z,W OFFSET.

3.10.2 Palettentabelle

Wenn Sie den Befehl PALLET (PALT) im Programm verwenden, werden die jeweiligen Daten der im Befehl angesprochenen Palettentabelle angesprochen. Wird der Palettierungsbefehl angewendet, müssen die entsprechenden Daten im voraus eingegeben werden.

- (1) Drücken Sie die Taste "5" (PALLET), wenn Sie sich im Hauptmenü (Bild 3.1) befinden.
- (2) Drücken Sie die Taste "2" (PALLET TABLE). Danach wird Bild 3.12 angezeigt.

PALLET	TABLE	No.	00
TOTAL	STEP		000
TOTAL	NUMBER		000
1	STEP	NUMBER	000
USER	DATA	1	000
USER	DATA	2	000
USER	DATA	3	000
USER	DATA	4	000
USER	DATA	5	000
USER	DATA	6	000
USER	DATA	7	000

Bild 3.12 Paletten-Tabelle

Taste "dec/INC" : Ändert die Nummer der Paletten-Tabelle

Taste " up/DOWN" : Bewegt den Cursor auf- und abwärts

Taste "ENTER" : Dateneingabe

3.11 Handbetrieb

Nach der Eingabe des HARL-U1-Programmes kann die Funktion der Unterprogramme manuell mit Hilfe des Teachpultes überprüft werden.

Achtung

Wenn der Handbetrieb für Greifer und anderen Endeffektoren ausprobiert wird, kann es durch Fehlbedienung oder Fehlprogrammierung zu unerwarteten Reaktionen kommen. Verhalten Sie sich vorsichtig.

- (1) Durch Drücken der Taste "6" für MANUAL (siehe Bild 3.1) wird das entsprechende Untermenü aufgerufen und die Anzeige zeigt nachfolgendes Bild.

MANUAL OPERATION			
0 0	0 1	0 2	0 3
0 4	0 5	0 6	0 7
P u s h A X I S K e y			

Bild 3.13 Menü Handbetrieb

- (2) Wählen Sie die Nummer des Handprogrammes durch Drücken der Taste "io/SEL" an. Bei jedem Drücken der Taste springt die Programmnummer um eins weiter.
- (3) Das Handprogramm wird ausgeführt, wenn die entsprechende zugeordnete Achs-Tipptaste gedrückt wird.

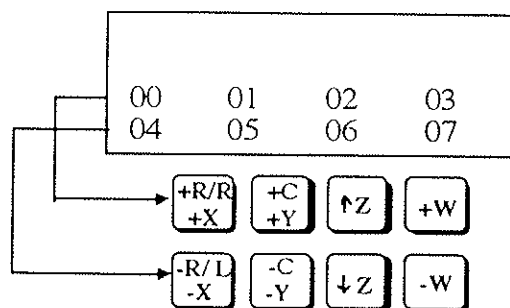


Bild 3.14 Zuordnung der Programmnummer zur Achs-Tipptaste

- (4) Während der Programmausführung wird "SEQUENCE EXECUTING" angezeigt. Nachdem die Programmausführung beendet ist, wird "COMPLETED" angezeigt und die Anzeige kehrt zu Bild 3.13 zurück.

Hinweis

Mit der Taste "CAN" kann der Vorgang jederzeit unterbrochen werden und die Anzeige kehrt zum vorherigen Menü zurück.

3.12 Programmüberprüfung

Eine Einzelschrittabarbeitung eines HARL-U1-Programmes ist möglich. Dabei kann der momentane Zustand von Speichern, Ein- und Ausgängen beobachtet werden.

- (1) Wählen Sie die Betriebsart CHECK am Teachpult. Drücken Sie gleichzeitig die Tasten "FUNC" und "seq", dann erscheint nachfolgendes Bild.

SEQUENCE CHECK	
MAIN PROGRAM No 00	→ Programmnummer
RUN MODE [STEP]	→ Angewählte Betriebsart
WAIT COMM. [WAIT]	→ Befehl WAIT aktiv / nicht aktiv
MOVE COMM. [MOVE]	→ Befehl MOVE aktiv / nicht aktiv
Push END Key	

Bild 3.15 Programmüberprüfung

Drücken Sie die Taste "up/DOWN", um den Cursor zu bewegen.

- 1) *MAIN PROGRAM NO.*

Anwahl des Hauptprogramms, das überprüft werden soll.

- 2) *RUN MODE [STEP/ALL]*

Anwahl, ob das Porgramm Schritt für Schritt oder kontinuierlich abgearbeitet werden soll.

Durch Drücken der Taste "io/SEL" wird STEP oder ALL angewählt.

- 3) *WAIT COMM. [WAIT/NO WAIT]*

Anwahl, ob WAIT-Befehle ausgeführt werden sollen oder nicht.

Durch Drücken der Taste "io/SEL" kann der Zustand zwischen WAIT und NO WAIT umgeschaltet werden. Wenn NO WAIT aktiv ist, werden die Befehle WAIT0, WAIT1 und WAITT nicht ausgeführt. Jedoch der WAITZ-Befehl wird auf jeden Fall ausgeführt, um die Roboterbewegung zu bestätigen.

4) *MOVE COMM. [MOVE/NO MOVE]*

Anwahl, ob MOVE-Befehle ausgeführt werden sollen oder nicht.

Durch Drücken der Taste "io/SEL" kann der Zustand zwischen MOVE und NO MOVE umgeschaltet werden. Wenn NO MOVE aktiv ist, werden die Befehle NMOVE, PMOVE, CMOVE, SMOVE, RMOVE, ACAL, HOLD, Z PULL UP, Z DN, Z UP, FINISH Z und MOVE STOP nicht ausgeführt. Dadurch können die Ein- und Ausgangssignale überprüft werden, ohne daß sich der Roboter bewegt.

- (2) Nachdem diese Daten eingegeben worden sind, drücken Sie die Taste "END". Danach wird untenstehendes Bild angezeigt.

MAIN PROGRAM No 00	—> Programmnummer
000 NOP	—> Programmschritt und Befehl
001 END	
Push START Key	

Bild 3.16 Anzeige Überprüfung

- (3) Drücken Sie die Taste "START" damit der angezeigte Befehl ausgeführt wird. Nachdem die Anweisung korrekt ausgeführt wurde, wird "COMPLETED" angezeigt und die nächste Programmzeile angezeigt.
- (4) Das Programm kann von jedem Programmschritt aus gestartet werden. Zum Vor- oder Zurückspringen im Programm drücken Sie entsprechend oft die Taste "dec/INC".
- (5) Während das Programm ausgeführt wird, können folgende Daten durch Drücken der korrespondierenden Achs-Tipptasten beobachtet werden.

Zeitgeber	Anzeige Taste "+ X" oder "-X"
Datenregister	Anzeige Taste "+ Y" oder "-Y"
Ein-/Ausgänge	Anzeige Taste "+ Z" oder "-Z"
Merker	Anzeige Taste "+ W" oder "-W"

Hinweis

1. Wenn die Taste "spd/CAN" gedrückt wird, wird vorherige Anzeige angezeigt.
2. Wenn "ALL" im "RUN MODE" angewählt werden soll, muß die Taste "START" gedrückt gehalten werden. Wenn die Taste "START" zwischenzeitlich losgelassen wird, wird die Betriebsart "STEP" angewählt.

3. Während der Ausführung des MOVE-Befehls muß die Taste "START" gedrückt bleiben.
4. MOVE-Befehle können nur ausgeführt werden, wenn vorher ein Kalibrierlauf ausgeführt wurde.

3.13 Sichern und Laden von Daten mittels Memory-Karte

Das HARL-U1-Programm kann auf eine Memory-Karte (128 kByte Type) gespeichert werden.

- (1) Nachdem die Memory-Karte in den Leser der Steuerung eingeschoben ist, wird nach dem gleichzeitigen Drücken der Tasten "FUNC" und "m.c." folgendes Bild angezeigt.

MEMORY CARD	
1. POS / SG	3. EDIT
2. SEQUENCE	4. UTIL
5. VERIFY	

Bild 3.17 Bedienung der Memory-Karte

- (2) Wenn ein Programm gespeichert oder geladen werden soll, muß die Taste "task/2" (SEQUENCE) gedrückt werden. Danach wird nachfolgendes Bild angezeigt.

MEMORY CARD (SEQ)	
1. ALL - LOAD	3. LOAD
2. ALL - SAVE	4. SAVE

Bild 3.18 Speichern und Laden eines Programmes

- 1) Wenn alle Programme von der Memory-Karte in die Steuerung geladen werden sollen, drücken Sie die Taste "cal/1" (ALL-LOAD). Danach wird nachfolgendes Bild angezeigt.

LOAD MEMORY CARD
MEMCARD >> ROBOT
(SEQUENCE)
ALL LOAD OK?

Bild 3.19 Laden von allen Programmen

Drücken Sie die Taste "ENTER" um den Ladevorgang zu starten. Nachdem der Ladevorgang abgeschlossen ist, wird COMPLETED angezeigt und die Anzeige kehrt zu Bild 3.18 zurück.

- 2) Wenn alle Programme von der Steuerung in die Memory-Karte geladen werden sollen, drücken Sie die Taste "task/2" (ALL-SAVE) und dann die Taste "ENTER", um dem Sicherungsvorgang zu starten.

Haben Sie ALL-SAVE angewählt, besteht folgender Zusammenhang zwischen Dateinummern (files) und den Datentypen.

Hauptprogramm : 9876100 bis 9876199

Unterprogramm : 9876200 bis 9876299

Handprogramm : 9876300 bis 9876399

Zeitgeberwerte : 9876400

Offset-Tabelle : 9876500

Systemdaten : 9876600

- 3) Wenn ein bestimmtes Programm oder bestimmte Daten von der Memory-Karte in die Steuerung geladen werden sollen, drücken Sie die Taste "3" (LOAD). Danach erscheint folgendes Bild.

SEQUENCE	LOAD
FILE NAME	0 0 0 0 0 0 0
Push END Key	

Bild 3.20 Laden von bestimmten Daten

Nachdem der Dateiname (Nummer) eingegeben worden ist, drücken Sie die Taste "END". Danach wird "SEQ LOAD OK?" angezeigt. Bestätigen Sie durch Drücken der Taste "ENTER", daß die Daten korrekt sind. Dann beginnt der Ladevorgang.

- 4) Wenn ein bestimmtes Programm oder bestimmte Daten von der Steuerung in die Memory-Karte geladen werden sollen, drücken Sie die Taste "4" (SAVE). Danach erscheint folgendes Bild.

S E Q U E N C E S A V E		
K I N D	N A M E	[M A I N] 0 0
F I L E	N A M E	0 0 0 0 0 0 0
P u s h E N D K e y		

Bild 3.21 Sichern von bestimmten Daten

Danach wählen Sie die Programm- oder Datentype durch Drücken der Taste "io/SEL". Die Type, die ausgewählt werden kann, ist "MAIN" (Hauptprogramm), "SUB" (Unterprogramm), "HAND" (Handprogramm), "TIME" (Zeitgeberwerte), "TABLE" (Offset-Tabellenwerte) oder "U1SYS" (HARL-U1 Systemdaten).

"MAIN", "SUB" oder "HAND" wird bei KIND NAME angegeben und danach die Programmnummer.

Nachdem die Dateinummer eingegeben wurde, drücken Sie die Taste "END". Danach wird "SEQ SAVE OK?" angezeigt. Drücken Sie die Taste "ENTER", um den Sicherungsvorgang zu starten.

(3) Nachdem Programme oder Daten abgespeichert worden sind, überprüfen Sie die Speicherung durch nachfolgende Prozedur.

1) Drücken Sie die Taste "mot/3" (EDIT), wenn Bild 3.17 angezeigt wird. Danach erscheint folgendes Bild.

F I L E E D I T		
1. FILE	2. LOAD	3. SAVE
4. COPY	5. DEL	6. DUMP
7. FIND		

Bild 3.22 Editier-Anzeige

2) Drücken Sie die Taste "cal/1" (FILE)

F I L E S		
F I L E S T Y P E		
[P o s i t i o n D a t a]		
P u s h E N D K e y		

Bild 3.23 Datei-Anzeige

Wählen Sie die entsprechende Datei durch Drücken der Taste "io/SEL" oder der numerischen Tasten "0" bis "6".

POSITION DATA	0 : Positionsdatei
SYSTEM DATA	1 : Systemparameter-Datei
MEMORY DATA	2 : Speicherdaten-Datei
MAIN PROGRAM	3 : Hauptprogramm-Datei
SUB PROGRAM	4 : Unterprogramm-Datei
HAND PROGRAM	5 : Handprogramm-Datei
TIMER DATA	6 : Zeitgeberwerte
OFFSET TABLE	7 : Offset-Tabellenwerte
HARL-U1 SYSTEM	8 : HARL-U1 Systemdaten
ALL	9 : Alle Werte

- 3) Nachdem Sie die Dateitype angewählt haben, drücken Sie die Taste "END". Dann wird "FILES TYPE OK?" angezeigt.

Drücken Sie dann die Taste "ENTER" und die folgenden Daten der gespeicherten Datei werden angezeigt.

0 0 0 0 1 0 0	M A I N	0 0 0 2 5 6 4
0 0 0 0 2 0 1	S U B	0 0 0 0 5 0 4
0 0 0 0 3 0 4	H A N D	0 0 0 0 5 0 4
0 0 0 0 1 1 2	M A I N	0 0 0 2 5 6 4

Bild 3.24 Datei-Daten

- 4) Gibt es fünf oder mehr Dateien, drücken Sie irgendeine Taste, damit die nächste Dateiinformation angezeigt wird.
- 5) Wenn alle Datei angezeigt wurden, wird "COMPLETED" angezeigt und die Anzeige zeigt Bild 3.23.

Hinweis

1. Tritt während der Benutzung der Memory-Karte eine Störung oder Fehler auf, informieren Sie sich bitte im Operation Handbuch über deren Behebung.

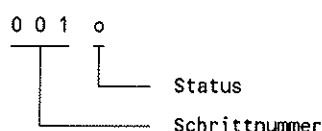
3.15 Aufzeichnung des Programmablaufs

Der Programmablauf im Automatikbetrieb oder bei der Programmüberprüfung kann aufgezeichnet werden. Diese Funktion ist sehr hilfreich, um die einzelnen Programmschritte zu überprüfen.

Drücken Sie die Tasten "FUNC" und "task" solange gleichzeitig, bis nachfolgendes Bild angezeigt wird.

S T E P H I S T O R Y			
M 0 1	[0 0 0]	°->	M 0 1 [0 0 1] °->
M 0 1	[0 0 2]	°->	M 0 1 [0 0 3] °->
S 0 2	[0 0 0]	°->	S 0 2 [0 0 1] °->

Bild 3.26 Aufzeichnung des Programmablaufs



* Status

- 0 korrekte Ausführung
- n Hardware-Fehler
- p Positionsfehler
- s Programmfehler
- e Programmende

* Schrittnummer

- XXX Schrittnummer des Hauptprogramms
- SXX Schrittnummer des Unterprogramms
- HXX Schrittnummer des Handprogramms
- XX Schrittnummer

Wollen Sie diese Betriebsart beenden, sind abermals die Tasten "FUNC" und "task" gleichzeitig zu drücken. Die Taste "CAN" hat in dieser Betriebsart keine Wirkung.

4 Ein- und Ausgangssignale

4.1 Eingangssignale

Die Tabelle 4.1 zeigt die Zuordnung der Eingangssignale

Nr.	Signalname	Bedeutung
IN00	PROG.START	Programm Startsignal
IN01	PROG.SEL 0	Programmanwahl Bit 0
IN02	PROG.SEL 1	Programmanwahl Bit 1
IN03	PROG.SEL 2	Programmanwahl Bit 2
IN04	PROG.SEL 3	Programmanwahl Bit 3
IN05	STOP	Programm Stoppsignal
IN06	HAND.SEL	Handprogramm Signal
IN07	RESTART	Programm Restartsignal
IN08	frei	frei für Anwendung
IN09	frei	frei für Anwendung
IN10	frei	frei für Anwendung
IN11	frei	frei für Anwendung
IN12	frei	frei für Anwendung
IN13	frei	frei für Anwendung
IN14	frei	frei für Anwendung
IN15	frei	frei für Anwendung

Tabelle 4.1

(1) PROG.START (IN00) : Programmstart

Dieses Signal dient zum Starten des HARL-U1-Programms durch externe Ansteuerung.

Wenn dieses Signal angelegt wird, startet das Programm, das durch den Signalcode an den Eingängen 1 bis 4 dekodiert worden ist. Wird das Handprogramm-Signal (IN06) und das Programm Startsignal angelegt, wird das entsprechende Handprogramm ausgeführt.

(2) PROG.SEL (IN01 - IN04) : Programmwahl

An diesen vier Eingängen kann die Programmnummer des Haupt- oder Handprogramms binär codiert angelegt werden. Bevor das Programm Startsignal IN00 angelegt wird, müssen diese Eingänge gesetzt sein.

Es können bis zu 16 Programme codiert werden. Die Tabelle 4.2 zeigt die entsprechende Codierung.

Prog-Nr.	IN 01	IN 02	IN 03	IN 04
00	AUS	AUS	AUS	AUS
01	EIN	AUS	AUS	AUS
02	AUS	EIN	AUS	AUS
03	EIN	EIN	AUS	AUS
04	AUS	AUS	EIN	AUS
05	EIN	AUS	EIN	AUS
06	AUS	EIN	EIN	AUS
07	EIN	EIN	EIN	AUS
08	AUS	AUS	AUS	EIN
09	EIN	AUS	AUS	EIN
10	AUS	EIN	AUS	EIN
11	EIN	EIN	AUS	EIN
12	AUS	AUS	EIN	EIN
13	EIN	AUS	EIN	EIN
14	AUS	EIN	EIN	EIN
15	EIN	EIN	EIN	EIN

Tabelle 4.2

(3) STOP (IN05) : Programmstop

Mit diesem Eingang kann die Ausführung des Programmes gestoppt werden.

Wird dieser Eingang während der Ausführung eines Programmes gesetzt, wird die Programmausführung unterbrochen und das Ausgangssignal PROG.END (OUT00) gesetzt. Die Bewegung des Roboters oder Handhabungsgerätes wird ebenso sofort unterbrochen.

(4) HAND.SEL (IN06) : Handprogrammanwahl

Mit diesem Signal wird entweder das Haupt- oder Handprogramm angewählt. Wird dieser Eingang gesetzt, ist das entsprechende Handprogramm angewählt.

Die Handprogramme 16 bis 19 können auf diese Art und Weise nicht ausgeführt werden.

(5) RESTART (IN07) : Programm Restart

Mit diesem Signal kann ein unterbrochenes Programm fortgesetzt werden.

Wenn das Programmstart Signal (IN00) gesetzt und das Programm durch das Programmstop Signal (IN05) unterbrochen wurde, kann der Programmablauf durch das Restartsignal an der Stelle fortgesetzt werden, an der es unterbrochen wurde.

(6) Freie Eingänge für Anwendung (IN08 bis IN15)

Die Eingänge 8 bis 15 sind nicht vorbelegt und stehen dem Anwender frei zur Verfügung.

4.2 Ausgangssignale

Die Tabelle 4.3 zeigt die Zuordnung der Ausgangssignale

Nr.	Signalname	Bedeutung
OUT00	PROG.END	Programmende Signal
OUT01	ERROR 0	Fehlercode Bit 0
OUT02	ERROR 1	Fehlercode Bit 1
OUT03	AUTO	Betriebsart Automatik
OUT04	B.P	Grundstellung
OUT05	Z SAFETY	Z-Achse auf Sicherheitshöhe
OUT06	CPC OUT	Bahnsteuerungssignal
OUT07	A-ACAL	Kalibriersignal
OUT08	Frei	frei für Anwendung
OUT09	Frei	frei für Anwendung
OUT10	Frei	frei für Anwendung
OUT11	Frei	frei für Anwendung
OUT12	Frei	frei für Anwendung
OUT13	Frei	frei für Anwendung
OUT14	Frei	frei für Anwendung
OUT15	Frei	frei für Anwendung

Tabelle 4.3 Zuordnung der Ausgangssignale

(1) PROG.END (OUT00) : Programmende

Dieser Ausgang zeigt das Ende eines Hauptprogramms an.

Wird der END-Befehl in dem Hauptprogramm ausgeführt oder das Programmstop Signal angelegt, wird dieser Ausgang gesetzt.

Wird das nächste Programm Startsignal angelegt, wird dieser Ausgang gelöscht.

(2) ERROR (OUT01 - OUT02) : Fehlercode

Tritt während der Programmausführung ein Fehler auf, wird ein Fehlercode an diesen Ausgängen ausgegeben. Der Code gibt an, ob es sich um einen Programm-, Positionsdaten- oder Hardwarefehler handelt.

Ist ein Befehl oder ein Wert im Programm falsch, erscheint der Programmfehler. Bei falscher Positionsadresse oder falschen Positionsdaten erscheint Positionsfehler. Tritt ein OVERRUN oder NOTAUS auf, wird der Code für Hardwarefehler gesetzt.

Die Tabelle 4.4 zeigt den Fehlercode

Fehlerart	ERROR 0	ERROR 1
Programmfehler	EIN	AUS
Positionsdatenfehler	AUS	EIN
Hardwarefehler	EIN	EIN

Tabelle 4.4 Fehlercodierung und Bedeutung

(3) AUTO (OUT03) : Automatik

Dieses Signal ist gesetzt, wenn sich die Steuerung im Automatikbetrieb befindet.

Wird der Schlüsselschalter auf dem Teachpult auf ONLINE geschaltet oder ist kein Teachpult an der Steuerung angeschlossen, befindet sich die Steuerung im Automatikbetrieb und dieser Ausgang wird gesetzt.

(4) B.P (OUT04) : Grundstellung

Dieses Signal wird bei dem PASS PTP-Betrieb benutzt oder wenn die beiden Hauptachsen A(X), B(Y) eine bestimmte Position haben.

Genauere Erläuterungen finden Sie im Operation Handbuch.

(5) Z SAFETY (OUT05) : Z-Achse auf Sicherheitshöhe

Wenn sich die Z-Achse des Roboters oberhalb der Sicherheitshöhe befindet, die mit dem Systemparameter "SAFETY ZONE" definiert wurde, wird dieser Ausgang gesetzt.

Genauere Erläuterungen finden Sie im Operation Handbuch.

(6) CPC OUT (OUT06) : Ausgabe während Bahnsteuerung

Dieser Ausgang wird während einer Bahnbewegung ein- und ausgeschaltet.

Genauere Erläuterungen finden Sie im Operation Handbuch.

(7) ACAL (OUT07) : Kalibriersignal

Wenn der Roboter oder das Handhabungsgerät kalibriert ist, ist dieser Ausgang gesetzt.

(8) Freie Ausgänge für Anwendung (OUT08 bis OUT15)

Die Ausgänge 8 bis 15 sind frei für die Anwendung verfügbar.

4.3 Ein-/Ausgangserweiterung (Option)

Durch eine Erweiterungskarte kann die Anzahl der Ein- und Ausgänge um jeweils 16 Ein- und 16- Ausgänge erweitert werden. Diese Ein- und Ausgänge stehen dem Anwender frei zur Verfügung.

Nr.	Bedeutung	Nr.	Bedeutung
IN16	frei für Anwendung	OUT16	frei für Anwendung
IN17	frei für Anwendung	OUT17	frei für Anwendung
IN18	frei für Anwendung	OUT18	frei für Anwendung
..		...	
..		...	
IN29	frei für Anwendung	OUT29	frei für Anwendung
IN30	frei für Anwendung	OUT30	frei für Anwendung
IN35	frei für Anwendung	OUT31	frei für Anwendung

Tabelle 4.5 Optionale Ein- und Ausgänge

5 Fehlermeldungen

Ist eine Dateneingabe falsch oder tritt eine Fehlbedienung auf, werden folgende Fehlermeldungen zusätzlich zu den im Operation Handbuch beschriebenen angezeigt.

Fehlermeldung	Erklärung
PROGRAM NOT FOUND	Das entsprechende Programm fehlt
DUPLICATE PROG.NAME	Die Programmnummer ist doppelt
CHECK SUM ERROR	Prüfsummenfehler im RAM / Daten fehlerhaft
STEP OVER	Die Anzahl der Programmschritte ist zu groß
PROGRAM NO. ERROR	Die Programmnummer ist falsch
STEP ERROR	Der Programmschritt ist fehlerhaft
TIMER NO. ERROR	Die Zeitgebernummer ist fehlerhaft
REG. NO. ERROR	Die Datenregisternummer ist fehlerhaft
DATA ERROR	Der eingegebene Wert ist fehlerhaft
POS ADDRESS ERROR	Die Positionsadresse liegt außerhalb des Bereichs
COMMAND ERROR	Die Programmanweisung ist falsch oder fehlerhaft
MOVE INCOMPLETED	Die Bewegung wurde unterbrochen
ACAL ERROR	Während dem Kalibriervorgang trat ein Fehler auf
I/O NO. ERROR	Die Ein- oder Ausgangsnummer ist fehlerhaft
DUPLICATE MOVE COMM	Während der Ausführung einer Roboterbewegung wurde ein weiterer Bewegungsbefehl gesendet
NESTING OVER	Eine Schachtelung im Unter- oder Handprogramm wurde mehr als dreimal durchgeführt.

Tabelle 5.1 Fehlermeldungen

6 Befehlsliste

Befehlsname	Befehl	Funktion
NOP-Befehl	NOP	Keine Befehlsausführung, Programm geht zum nächsten Befehl
END-Befehl	END	Beendet das Programm
STOP-Befehl	STOP	Stoppt die Programmabarbeitung nachdem der momentane Befehl ausgeführt wurde
Sprungbefehl	JUMP	Das Programm springt auf den spezifizierten Programmschritt
Verzweigungsbefehle	BRNO	Das Programm verzweigt auf den spezifizierten Programmschritt, wenn der Eingang oder Merker Null ist.
	BRN1	Das Programm verzweigt auf den spezifizierten Programmschritt, wenn der Eingang oder Merker eins ist.
	BRNT	Das Programm verzweigt auf den spezifizierten Programmschritt, wenn der Zeitgeber abgelaufen ist.
Unterprogrammaufruf	SUB	Aufrufen eines Unterprogramms
Handprogrammaufruf	HAND	Aufrufen eines Handprogramms
Geschwindigkeitsbefehl	SPEED	Steht für Geschwindigkeit Bestimmt die Geschwindigkeit des Roboters in %.
Beschleunigungsbefehl	ACC	Steht für Beschleunigung bzw. Verzögerung Bestimmt die Beschleunigung/Verzögerung des Roboters in %.
Zeitgeber setzen	TIMER	Steht für die Intervall-Zeitgeber Der Inhalt des spezifizierten Zeitgebers wird auf den Intervall-Zeitgeber gesetzt.
Ausgabebefehl	E_ON	Steht für Ausgang oder Bitmerker setzen
	E_OFF	Steht für Ausgang oder Bitmerker zurücksetzen
Datenregister setzen	D SET	Setzt ein Datenregister auf einen bestimmten Wert.
Datenregister erhöhen	D INC	Erhöht den Inhalt eines Datenregisters um eins
Datenregister erniedrigen	D DEC	Verringert den Inhalt eines Datenregisters um eins
Datenregister Addition	D ADD	Addiert den Inhalt zweier Datenregister
Datenregister Subtraktion	D SUB	Subtrahiert den Inhalt zweier Datenregister
Datenregister Substitut.	D EQU	Ersetzt den Inhalt eines Datenregisters mit dem eines andern
Datenregistervergleich	D CMP	Vergleicht den Inhalt zweier Datenregister. Wenn der Inhalt gleich ist, verzweigt das Programm zu dem spezifizierten Programmschritt.
Wartebefehle	WAIT0	Steht für Verweilen des Programms Das Programm verweilt auf diesem Befehl, bis spezifizierter Eingang Null wird.
	WAIT1	Steht für Verweilen des Programms Das Programm verweilt auf diesem Befehl, bis spezifizierter Eingang eins wird.
	WAITP	Wird in Verbindung mit dem R_MOV-Befehl verwendet Der Roboter passiert erst dann die Position, wenn der peripherie Prozeß beendet ist.
	WAITT	Steht für Verweilen des Programms Das Programm verweilt auf diesem Befehl, bis spezifizierter Zeitgeber abgelaufen ist.
	WAITZ	Steht für Verweilen des Programms Das Programm verweilt auf diesem Befehl, bis sich die Z-Achse innerhalb der Z SAFETY ZONE befindet.

Befehlsname	Befehl	Funktion
Spezielle Sprungbefehle	P_CHK	Steht für Positionsüberprüfung Das Programm verzweigt auf den spezifizierten Programmschritt, wenn der Roboter seine Positionierung beendet hat.
	C_CHK	Steht für Kalibrierungsüberprüfung Das Programm verzweigt auf den spezifizierten Programmschritt, wenn der Roboter kalibriert ist.
	H_CHK	Steht für Verriegelungsüberprüfung Das Programm verzweigt auf den spezifizierten Programmschritt, wenn die Achsen des Roboters verriegelt sind.
	Z_CHK	Steht für Z-Achsenüberprüfung Das Programm verzweigt auf den spezifizierten Programmschritt, die Z-Achse innerhalb der SAFETY ZONE befindet.
Interruptbefehl	INTST	Steht für Interruptsignaldefinition Spezifiziert einen Eingang als Interrupt-Eingang
Verriegelungsbefehl	HOLD	Schaltet die Servoverriegelung der Achsen ein oder aus
Kalibrier-Befehl	ACAL	Ausführen des Kalibriervorgangs
Bewegungsbefehle	N_MOV	Steht für N_MOVE Der Roboter fährt zu der spezifizierten Positionsadresse. Das Programm verweilt an diesem Befehl bis die Positionierung beendet ist.
	P_MOV	Steht für P_MOVE Der Roboter fährt zu der durch das Datenregister spezifizierten Positionsadresse. Das Programm verweilt an diesem Befehl bis die Positionierung beendet ist.
	C_MOV	Steht für C_MOVE Der Roboter fährt zu der spezifizierten Positionsadresse, parallel wird der nächste Schritt im Programm abgearbeitet.
	S_MOV	Steht für S_MOVE Der Roboter fährt zu der durch das Datenregister spezifizierten Positionsadresse, parallel wird der nächste Schritt im Programm abgearbeitet.
	R_MOV	Steht für R_MOVE Der Roboter fährt zu der durch die Offset-Tabelle erzeugte Positionsadresse
	FIN_Z	Erlaubt das Absenken der Z-Achse
Bewegungsstop	MVSTP	Stoppt die Roboterbewegung
Bewegungsende	MVEND	Steht für Bewegungsende Das Programm verweilt, bis die Bewegung beendet ist.
Z-Achsenbewegungsbefehl	Z_PUL	Die Z-Achse wird an der momenten Position auf die PULL-UP-Ebene hochgefahren.
Z-Achsenbewegungsbefehl	Z_UP	Die Z-Achse wird um den Wert, der in dem spezifizierten Datenregister steht, angehoben.
Z-Achsenbewegungsbefehl	Z_DN	Die Z-Achse wird um den Wert, der in dem spezifizierten Datenregister steht, abgesenkt.
Speicherkarten-Lesebefehl	MLOAD	Lesen der spezifizierten Positionsadresse, die in der Memorykarte gespeichert ist
E/A-Bedienungsbefehl	D_IN	Der Status des Eingangports wird in die entsprechenden Datenregister geschrieben
	D_OT	Der Status der Datenregister wird auf den entsprechenden Ausgangsport geschrieben
Visionbefehl	VSCE	Aufrufen der spezifizierten Szene im Visionsystem
Palettierungsroutine	PALT	Aufruf der Palettierungsroutine Diese Routine ist derzeit nicht verfügbar.

7 Beschreibung der Befehle

ACAL	ACAL
-------------	-------------

Funktion Ausführen eines automatischen Kalibrierlaufes (A-CAL)

Format ACAL

Beispiel ACAL

Erklärung Das Gerät führt einen Referenz- oder Kalibrierlauf aus. Ist der Kalibrier-
vorgang beendet, führt das Programm den nächsten Schritt aus.

Notizen

ACC (Accel)		ACC (Accel)
<i>Funktion</i>	Setzen der Beschleunigung / Verzögerung des Gerätes	
<i>Format</i>	ACC <Wert1>	Wert1 : 0 bis 100% Beschleunigungsfaktor
<i>Beispiel</i>	ACC 080	
<i>Erklärung</i>	Wurde der Wert einmal gesetzt, bleibt er solange gültig, bis er überschrieben wird. Die tatsächliche Beschleunigung bzw. Verzögerung wird prozentual von dem Wert der Systemparameter ACCEL A,B,W berechnet.	
<i>Notizen</i>		

BRN0 (Branch 0 (off))	BRN0 (Branch 0 (off))
-----------------------	-----------------------

<i>Funktion</i>	Das Programm verzweigt zu der angegebenen Schrittnummer, wenn der spezifizierte Eingang oder der Merker Null (off) ist.		
<i>Format</i>	BRN0 <Wert1> <Wert2>	<Wert1>	00 - 31 Eingangsnummer 32 - 63 Merker Nummer
		<Wert2>	000 - 999 Schrittnummer
<i>Beispiel</i>	BRN0 10 045		
<i>Erklärung</i>	Das Programm verzweigt auf die Schrittnummer des <Wert2>, wenn der <Wert1> des Eingangs oder Merkers Null ist. Ist der <Wert1> gleich 1 (on) verzweigt das Programm nicht und führt den nächsten Programmschritt aus.		
<i>Notizen</i>			

BRN1 (Branch 1 (on))**BRN1 (Branch 1 (on))**

<i>Funktion</i>	Das Programm verzweigt zu der angegebenen Schrittnummer, wenn der spezifizierte Eingang oder der Merker eins (on) ist.		
<i>Format</i>	BRN1 <Wert1> <Wert2>	<Wert1>	00 - 31 Eingangsnummer 32 - 63 Merckernummer
		<Wert2>	000 - 999 Schrittnummer
<i>Beispiel</i>	BRN1 10 045		
<i>Erklärung</i>	Das Programm verzweigt auf die Schrittnummer des <Wert2>, wenn der <Wert1> des Eingangs oder Merkers eins ist. Ist der <Wert1> gleich Null (off) verzweigt das Programm nicht und führt den nächsten Programmschritt aus.		
<i>Notizen</i>			

BRNT (Branch Timer)		BRNT (Branch Timer)	
<i>Funktion</i>	Das Programm verzweigt zu der angegebenen Schrittnummer, wenn der spezifizierte Zeitgeber abgelaufen ist.		
<i>Format</i>	BRNT <Wert1> <Wert2>	<Wert1>	0 oder 1 Intervallzeitgebernr.
		<Wert2>	000 - 999 Schrittnummer
<i>Beispiel</i>	BRNT 1 045		
<i>Erklärung</i>	Das Programm verzweigt auf die Schrittnummer des <Wert2>, wenn der <Wert1> des Intervall-Zeitgeber abgelaufen ist. Ist der Zeitgeber noch nicht gelaufen, wird der nächste Programmschritt ausgeführt. Der Intervall-Zeitgeber muß erst gesetzt werden, bevor dieser Befehl ausgeführt wird.		
<i>Notizen</i>			

C CHK (Calibration Check)		CHK (Calibration Check)	
<i>Funktion</i>	Das Programm verzweigt zu der angegebenen Schrittnummer, wenn das Gerät kalibriert ist. Ist das Gerät nicht kalibriert, wird der nächste Programmschnitt ausgeführt.		
<i>Format</i>	C_CHK <Wert1>	<Wert1>	000 - 999 Schrittnummer
<i>Beispiel</i>	C_CHK 080		
<i>Erklärung</i>	Das Programm verzweigt auf die Schrittnummer des <Wert1>, wenn das Gerät kalibriert ist. Ist das Gerät nicht kalibriert, verzweigt das Programm nicht und führt den nächsten Programmschritt aus.		
<i>Notizen</i>			

C MOV

<i>Erklärung</i>	Der größte Wert von <Wert1> wird durch den Systemparameter "ADDRESS MAX" bestimmt. Wenn dieser Befehl ausgeführt wird, beginnt der Roboter sich zu der spezifizierten Position zu bewegen und der nächste Programmschritt wird ausgeführt. Dieser Befehl muß zusammen mit den Befehlen FIN_Z und MVEND (MOVE END) benutzt werden.
------------------	--

1. Mit einem Sprungbefehl (JUMP oder BRN0) kann nicht von C_MOVE zu MOVE_END oder zwischen C_MOVE und MVEND von außerhalb hinein gesprungen werden.
2. Normalerweise ist der <Wert2> auf 00 gesetzt. Die Funktionen für andere Werte werden in Kapitel 10 des Handbuches beschrieben.

D ADD (Data Register Addition)		D ADD (Data Register Addition)	
<i>Funktion</i>	Addiert den Inhalt von zwei Datenregistern miteinander		
<i>Format</i>	D_ADD <Wert1> <Wert2>	<Wert1>	00 - 15 Datenregisternummer
		<Wert2>	00 - 15 Datenregisternummer
<i>Beispiel</i>	D_ADD 13 14		
<i>Erklärung</i>	Die Inhalte der beiden Datenregister, die durch <Wert1> und <Wert2> spezifiziert werden, werden addiert und das Ergebnis in das Datenregister von <Wert1> abgelegt.		
	$D<Wert1> \text{ Wert} = D<Wert1> \text{ Wert} + D<Wert2> \text{ Wert}$		
<i>Notizen</i>			

D CMP (Data Register Comparison)		CMP (Data Register Comparison)	
<i>Funktion</i>	Vergleicht den Inhalt von zwei Datenregistern miteinander. Ist der Inhalt gleich, springt das Programm zu dem spezifizierten Programmschritt.		
<i>Format</i>	D_CMP <Wert1> <Wert2> <Wert3> <Wert1>	00 - 15 Datenregisternr.	
		<Wert2>	00 - 15 Datenregisternr.
		<Wert3>	000 - 999 Schrittnummer
<i>Beispiel</i>	C_CMP 10 11 056		
<i>Erklärung</i>	Die Inhalte der beiden Datenregister, die durch <Wert1> und <Wert2> spezifiziert werden, werden miteinander verglichen. Ist das Ergebnis gleich, springt das Programm zu der spezifizierten Schrittnummer, wenn nicht, wird der nächste Programmschritt ausgeführt.		
<i>Notizen</i>			

D DEC (Data Register Decrement)**D DEC (Data Register Decrement)**

Funktion Vom Inhalt des Datenregisters wird eins subtrahiert.

Format D_DEC <Wert1> <Wert1> 00 - 15 Datenregisternummer

Beispiel D_DEC 13

Erklärung Vom Inhalt des Datenregisters, das durch <Wert1> spezifiziert wird, wird eins abgezogen.

$$D<Wert1> \text{ Wert} = D<Wert1> \text{ Wert} - 1$$

Hinweis

Werte kleiner Null werden nicht reduziert.

Notizen

D INC (Data Register Increment)**D INC (Data Register Increment)**

Funktion Erhöht den Inhalt eines Datenregisters um eins.

Format D_INC <Wert1> <Wert1> 00 - 15 Datenregisternummer

Beispiel D_INC 11

Erklärung Der Inhalt des Datenregisters, das durch <Wert1> spezifiziert wird, wird um eins erhöht.

$$D<Wert1> \text{ Wert} = D<Wert1> \text{ Wert} + 1$$

Notizen

D OT (Data Output)**D OT (Data Output)**

Funktion Setzt den Zustand des Datenregisters auf das Ausgangsport.

Format D_OT <Wert1> <Wert2> <Wert3>

 <Wert1> 00 - 15 Datenregister

 <Wert2> 08 - 31 Ausgangsnummer

 <Wert3> 00 - 31 Datenbreite

Beispiel D_OT 10 08 05

Erklärung Der Zustand des spezifizierten Datenregister <Wert1> wird auf das spezifizierte Ausgangsport <Wert2> mit einer Datenbreite von <Wert3> übertragen.
Im Beispiel wird der Zustand des Datenregisters 10 auf die Ausgänge 8 bis 12 übertragen.

Notizen

D SUB (Data Register Subtraction)		D SUB (Data Register Subtraction)	
<i>Funktion</i>	Subtrahiert den Inhalt zweier Datenregister voneinander.		
<i>Format</i>	D_SUB <Wert1> <Wert2>	<Wert1>	00 - 15 Datenregisternummer
		<Wert2>	00 - 15 Datenregisternummer
<i>Beispiel</i>	D_SUB 13 14		
<i>Erklärung</i>	Die Inhalte der beiden Datenregister, die durch <Wert1> und <Wert2> spezifiziert werden, werden subtrahiert und das Ergebnis in das Datenregister von <Wert1> abgelegt.		
	$D<Wert1> Wert = D<Wert1> Wert - D<Wert2> Wert$		
<i>Notizen</i>			

END (Program End)	END (Program End)
--------------------------	--------------------------

Funktion Beendet die Programmabarbeitung.

Format END

Beispiel END

Erklärung Wird dieser Befehl im Hauptprogramm benutzt, wird das Programm beendet und das Programmende Signal wird ausgegeben.

Wird dieser Befehl im Unter- oder Handprogramm benutzt, wird der vorherige Programmschritt der Schleife im Hauptprogramm ausgeführt.

Notizen

E OF (External Output Off)		E OF (External Output Off)	
<i>Funktion</i>	Setzt den spezifizierten Ausgang oder Merker auf Null (off).		
<i>Format</i>	E_OF <Wert1>	<Wert1>	00 - 31 Ausgangsnummer 32 - 63 Merckernummer
<i>Beispiel</i>	D_OF 40		
<i>Erklärung</i>	Wenn der <Wert1> zwischen 00 und 31 liegt, wird der entsprechende Ausgang auf Null gesetzt. Liegt der <Wert1> zwischen 32 und 63, wird der entsprechende Merker auf Null gesetzt. Wird der Wert gesetzt, bleibt er solange erhalten, bis er neu gesetzt wird.		
<i>Notizen</i>			

FIN Z (Finish Z)	FIN Z (Finish Z)
<i>Funktion</i>	Gibt die Abwärtsbewegung der Z-Achse frei.
<i>Format</i>	FIN_Z
<i>Beispiel</i>	FIN_Z
<i>Erklärung</i>	Während der S MOVE- oder C MOVE-Befehl ausgeführt wird, kann die Abwärtsbewegung der Z-Achse der jeweiligen Bewegung nicht ausgeführt werden, solange dieser Befehl nicht ausgeführt wurde. (siehe Kapitel 10 für die Bedingungen der Z-Abwärtsbewegung)
<i>Notizen</i>	

HAND (Hand Program)**HAND (Hand Program)**

Funktion Führt das spezifizierte Handprogramm aus.

Format HAND <Wert1> <Wert1> 00 - 19 Handprogrammnummer

Beispiel HAND 10

Erklärung Führt das durch den <Wert1> spezifizierte Handprogramm aus.

Man beachte, daß die maximale Verschachtelung von Unter- und Handprogrammen drei beträgt.

Notizen

H CHK (Hold Check)		H CHK (Hold Check)	
<i>Funktion</i>	Überprüft, ob das Gerät im HOLD (Servoverriegelung) ist. Wenn das Gerät im Hold ist, verzweigt das Programm zu der spezifizierten Adresse.		
<i>Format</i>	H_CHK <Wert1>	<Wert1>	000 - 999 Programmschritt
<i>Beispiel</i>	H_CHK 021		
<i>Erklärung</i>	Wenn die Verriegelung eingeschaltet ist, springt das Programm auf die spezifizierte Adresse. Wenn die Verriegelung ausgeschaltet ist, wird der nächste Programmschritt ausgeführt.		
<i>Notizen</i>			

HOLD**HOLD**

<i>Funktion</i>	Schaltet die Servoverriegelung der Achsen ein und aus.		
<i>Format</i>	HOLD <Wert1>	<Wert1>	0 : Verriegelung aller Achsen aus 1 : Verriegelung A-Achse ein 2 : Verriegelung B-Achse ein 3 : Verriegelung Z-Achse ein 4 : Verriegelung W-Achse ein 5 : Verriegelung aller Achsen ein
<i>Beispiel</i>	HOLD 1		
<i>Erklärung</i>	Je nach <Wert1> wird die Servoverriegelung der jeweiligen Achse eingeschaltet.		
<i>Notizen</i>			

INTST (Interrupt Signal Set)	INTST (Interrupt Signal Set)
------------------------------	------------------------------

Funktion Bestimmt einen Eingang als Interrupt-Eingang. (Unterbrechung)

Format INTST <Wert1> <Wert1> 00 - 31 Eingangsnummer

Beispiel INTST 09

Erklärung Wenn der durch den <Wert1> spezifizierte Eingang eins wird, während ein S MOVE- oder C MOVE-Befehl ausgeführt wird, wird die Roboterbewegung sofort unterbrochen und das Programm läuft direkt zu dem MOVE END-Befehl weiter. Das Programm fährt dann nach dem MOVE END-Befehl fort. Wird der Interrupt-Eingang eins, wird der Merker 63 auch auf eins gesetzt, damit der Interrupt-Zustand vom Programm erfaßt werden kann.

Notizen

JUMP (Step Jump)		JUMP (Step Jump)	
<i>Funktion</i>	Springt ohne Bedingung auf die spezifizierte Adresse.		
<i>Format</i>	JUMP <Wert1>	<Wert1>	000 - 999 Programmschritt
<i>Beispiel</i>	JUMP 100		
<i>Erklärung</i>	Springt auf die Adresse, die durch <Wert1> spezifiziert ist. Für den Sprung werden keine Bedingungen abgefragt.		
<i>Notizen</i>			

MVEND (Move End)		MVEND (Move End)
<i>Funktion</i>	Das Programm läuft parallel zu einer Roboterbewegung weiter, bis es auf diesen Befehl kommt.	
<i>Format</i>	MVEND	
<i>Beispiel</i>	MVEND	
<i>Erklärung</i>	Wurde ein S MOVE oder C MOVE-Befehl ausgeführt, wird das Programm parallel zur Roboterbewegung ausgeführt, bis es den MVEND-Befehl erreicht. Dort verweilt das Programm, bis der Roboter seine Zielposition erreicht hat.	
<i>Notizen</i>		

MLOAD (Memory-Karte laden)**MLOAD (Memory-Karte laden)**

Funktion Liest die Positionsdaten aus einer Datei, die auf der Memory-Karte gespeichert ist.

Format MLOAD <Wert1> <Wert1> 00 - 15 Datenregister

Beispiel MLOAD 10

Erklärung Liest die Positionsdaten aus einer Datei, die auf der Memory-Karte gespeichert ist. Die zulesende Dateinummer steht in dem Datenregister, das durch den Wert1 aufgerufen wird.
Die Positionsdaten müssen vorher auf die Memory-Karte geladen worden sein.

Notizen

MVSTP (Move Stop)	MVSTP (Move Stop)
<i>Funktion</i>	Wenn dieser Befehl ausgeführt wird, wird eine Bewegung des Roboters sofort unterbrochen.
<i>Format</i>	MVSTP
<i>Beispiel</i>	MVSTP
<i>Erklärung</i>	Wird ein S MOV, C MOV oder R MOV-Befehl ausgeführt, bleibt der Roboter bei diesem Befehl sofort stehen, das Programm springt direkt zum MVEND-Befehl und der nächste Befehl nach MVEND wird ausgeführt. Siehe Kapitel 10.
<i>Notizen</i>	

N MOV**N MOV**

<i>Funktion</i>	Der Roboter fährt auf die spezifizierte Positionsadresse		
<i>Format</i>	N_MOV <Wert1> <Wert2>	<Wert1>	000 - 999 Positionsnummer
		<Wert2>	00 - 99 optionaler Wert
<i>Beispiel</i>	N_MOV 200 00		
<i>Erklärung</i>	Der größte Wert von <Wert1> wird durch den Systemparameter "ADDRESS MAX" bestimmt. Wenn dieser Befehl ausgeführt wird, beginnt der Roboter sich zu der spezifizierten Position zu bewegen. Wurde die Position ordnungsgemäß erreicht, wird der nächste Programmschritt ausgeführt.		

Hinweis

1. Normalerweise ist der <Wert2> auf 00 gesetzt. Die Funktionen für andere Werte werden in Kapitel 10 des Handbuches beschrieben.

Notizen

NOP (No Operation)	NOP (No Operation)
---------------------------	---------------------------

Funktion Keine Porgrammoperation. Der nächste Schritt wird ausgeführt.

Format NOP

Beispiel NOP

Erklärung Keine Porgrammoperation. Der nächste Schritt wird ausgeführt.

Notizen

P CHK (Position Check)		P CHK (Position Check)	
<i>Funktion</i>	Springt zu der spezifizierten Schrittnummer, wenn der Roboter die Position ordnungsgemäß angefahren hat. Wenn nicht, wird der nächste Programmschritt ausgeführt.		
<i>Format</i>	P_CHK <Wert1>	<Wert1>	000 - 999 Schrittnummer
<i>Beispiel</i>	P_CHK 100		
<i>Erklärung</i>	Wenn der Roboter die Position erreicht und gestoppt hat, springt das Programm auf die mit <Wert1> spezifizierte Schrittnummer. Wenn nicht wird der nächste Programmschritt ausgeführt.		
<i>Notizen</i>			

P MOV

<i>Format</i>	P_MOV <Wert1> <Wert2>	<Wert1>	00 - 15 Datenregisternummer
		<Wert2>	00 - 99 optionaler Wert

Erklärung Der größte Wert von <Wert1> wird durch den Systemparameter "ADDRESS MAX" bestimmt.

Wenn dieser Befehl ausgeführt wird, beginnt der Roboter sich zu der spezifizierten Position zu bewegen. Wurde die Position ordnungsgemäß erreicht, wird der nächste Programmschritt ausgeführt.

1. Normalerweise ist der <Wert2> auf 00 gesetzt. Die Funktionen für andere Werte werden in Kapitel 10 des Handbuches beschrieben.

65

R MOV**R MOV**

Funktion Der Roboter positioniert auf einer erzeugten Bahnposition und bedient Ein- und Ausgänge während der Bewegung.

Format R_MOV <Wert1> <Wert2> <Wert3>

<Wert1> 00 - 15 Datenregister, das die Referenzadresse enthält

<Wert2> 00 - 15 Datenregister, das die Offset-Tabellennummer enthält

<Wert3> 00 - 99 Anzahl von Bahnpunkten, die erzeugt werden sollen.

Beispiel R_MOV 10 05 04

Erklärung Dieser Befehl wird zur schnelleren Palettierung benutzt. Der Roboter bewegt sich zur der Positionsadresse, die durch den Inhalt des Datenregisters <Wert1> spezifiziert ist und passiert dabei die Raumpunkte die durch den <Wert3> spezifiziert sind. Die zupassierenden Raumpunkte werden automatisch erzeugt, dabei werden die Daten der Offset-Tabelle herangezogen, die durch den Inhalt des Datenregisters <Wert2> bestimmt ist.

Hinweise

1. Dieser Befehl muß in Verbindung mit dem WAITP- und MVEND-Befehl verwendet werden.
2. Eine Programmverzweigung aus oder in den Abschnitt zwischen dem ersten R_MOV- und MVEND-Befehl ist nicht zulässig.

Notizen

S MOV

Dieser Befehl muß zusammen mit den Befehlen `FINISH_Z` (FINI) und `MOVE END` (MVED) benutzt werden.

1. Mit einem Sprungbefehl (JUMP oder BRN0) kann nicht von C_MOV zu MVEND oder zwischen C_MOV und MVEND von außerhalb hinein gesprungen werden.
2. Normalerweise ist der <Wert2> auf 00 gesetzt. Die Funktionen für andere Werte werden in Kapitel 10 des Handbuches beschrieben.

67

SPEED**SPEED**

Funktion Setzen der Achsengeschwindigkeit des Roboters.

Format SPEED <Wert1> <Wert1> 0 - 100% Geschwindigkeitsfaktor

Beispiel SPEED 100

Erklärung Wird der Wert einmal gesetzt, bleibt er solange gültig, bis er erneut gesetzt wird.

Die tatsächliche Geschwindigkeit hängt jedoch auch von den Systemparametern und dem F-Code ab.

Siehe Systemparameter und F-Code im Operation Handbuch.

Notizen

STOP (Program Stop)	STOP (Program Stop)
---------------------	---------------------

Funktion Programmausführung stoppen.

Format STOP

Beispiel STOP

Erklärung Das Programm wird gestoppt, sobald der momentan abgearbeitete Befehl beendet ist.

Um das Programm wieder zu starten, muß das Programm Startsignal erneut angelegt werden.

Notizen

SUB (Sub Program)		SUB (Sub Program)	
<i>Funktion</i>	Aufruf eines Unterprogramms		
<i>Format</i>	SUB <Wert1>	<Wert1>	00 - 19 Unterporgrammnummer
<i>Beispiel</i>	SUB 12		
<i>Erklärung</i>	Das Unterprogramm, das durch <Wert1> spezifiziert ist, wird ausgeführt. Die maximale Verschachtelungstiefe von Unter- und Handprogrammen beträgt drei Ebenen.		
<i>Notizen</i>			

TIMER (Interval Timer)		TIMER (Interval Timer)	
<i>Funktion</i>	Setzt den Zeitwert des Intervall-Zeitgebers.		
<i>Format</i>	TIMER <Wert1> <Wert2>	<Wert1>	0 - 1 Intervall-Zeitgebernr.
		<Wert2>	00 -15 Zeitgebernummer
<i>Beispiel</i>	TIMER 1 10		
<i>Erklärung</i>	Es gibt zwei Intervall-Zeitgeber (0 und 1). Der Wert des Zeitgebers, der durch <Wert2> spezifiziert wird, wird auf den Interval-Zeitgeber übertragen, der durch <Wert1> spezifiziert wird. Wenn der Befehl ausgeführt wurde, beginnt der Zeitgeber zu laufen. Ob der Zeitgeber abgelaufen ist oder nicht, kann mit dem BRNT-Befehl überprüft werden. Siehe auch BRNT-Befehl.		
<i>Notizen</i>			

VSCE (Vision Scene Set)**VSCE (Vision Scene Set)**

Funktion Setzt das Visionsystem.

Format VSCE < Wert1 > < Wert1 > 00 - 15 Szenennummer

Beispiel VSCE 02

Erklärung Ruft die entsprechende Szenennummer im Vision-System auf.

ACHTUNG

Bitte benutzen Sie diesen Befehl auf keinen fall, wenn kein Vision-System angeschlossen ist.

Notizen

Z CHK (Z Safety Check)		Z CHK (Z Safety Check)	
<i>Funktion</i>	Das Programm springt zu der spezifizierten Adresse, wenn sich die Z-Achse oberhalb der Sicherheitsebene befindet. Wenn nicht, wird der nächste Programmschritt ausgeführt.		
<i>Format</i>	Z_CHK <Wert1>	<Wert1>	000 - 999 Schrittnummer
<i>Beispiel</i>	Z_CHK 050		
<i>Erklärung</i>	Das Porgramm springt auf den Programmschritt, der durch <Wert1> spezifiziert ist, wenn sich die Z-Achse oberhalb der durch dem Parameter Z SAFTEY ZONE definierte Sicherheitsebene befindet. Wenn nicht, wird der nächste Programmschritt ausgeführt.		
<i>Notizen</i>			

Z DN (Z Down)	Z DN (Z Down)
---------------	---------------

Funktion Nur die Z-Achse verfährt nach unten um den Abstand, der in dem spezifizierten Datenregister steht.

Format Z_DN <Wert1> <Wert1> 00 - 15 Datenregisternummer

Beispiel Z_DN 04

Erklärung Die Z-Achse verfährt von ihrer momentanen Position um den Betrag nach unten, der in dem durch <Wert1> spezifizierten Datenregister steht.

Notizen

Z PUL (Z Pull-up)**Z PUL (Z Pull up)**

<i>Funktion</i>	Nur die Z-Achse verfährt auf die Höhe der PULL-UP-Ebene, die in dem Systemparameter "PULL UP" gesetzt ist.
<i>Format</i>	Z_PUL
<i>Beispiel</i>	Z_PUL
<i>Erklärung</i>	Die Z-Achse verfährt von ihrer momentanen Position auf die PULL-UP-Ebene hoch. Die PULL-UP-Ebene wird in den Systemparametern gesetzt.
<i>Notizen</i>	

Z UP**Z UP**

Funktion Nur die Z-Achse verfährt nach oben um den Abstand, der in dem spezifizierten Datenregister steht.

Format Z_UP <Wert1> <Wert1> 00 - 15 Datenregisternummer

Beispiel Z_UP 04

Erklärung Die Z-Achse verfährt von ihrer momentanen Position um den Betrag nach oben, der in dem durch <Wert1> spezifizierten Datenregister steht.

Die Z-Achse fährt jedoch nicht über den Wert der PULL-UP-Ebene hinaus.

Notizen

WAIT0 (Wait 0 (OFF))**WAIT0 (Wait 0 (OFF))**

Funktion Warten bis das spezifizierte Eingangssignal Null wird.

Format WAIT0 <Wert1> <Wert1> 00 - 31 Eingangsnummer

Beispiel WAIT0 12

Erklärung Der Programmablauf verweilt solange an diesem Befehl, bis der durch
 <Wert1> spezifizierte Eingang Null wird.

Nur Eingänge können mit diesem Befehl verarbeitet werden.
Merker können nicht verwendet werden.

Notizen

WAIT1 (Wait 1 (ON))**WAIT1 (Wait 1 (ON))**

Funktion Warten bis das spezifizierte Eingangssignal eins wird.

Format WAIT1 <Wert1> <Wert1> 00 - 31 Eingangsnummer

Beispiel WAIT1 11

Erklärung Der Programmablauf verweilt solange an diesem Befehl, bis der durch
 <Wert1> spezifizierte Eingang eins wird.

Nur Eingänge können mit diesem Befehl verarbeitet werden.
Merker können nicht verwendet werden.

Notizen

WAITP	WAITP
--------------	--------------

Funktion Der Roboter wartet an einem Verschleifpunkt, bis der periphere Ablauf abgeschlossen ist.

Format WAITP

Beispiel WAITP

Erklärung Wenn der R MOV-Befehl ausgeführt wird, wartet der Roboter solange an einem Verschleifpunkt (Pass point) bis der Programmablauf, der die Peripherie steuert, beendet ist.
Der WAITP-Befehl ist erforderlich, wenn der R_MOV-Befehl benutzt wird.

Notizen

WAITT (Wait Timer)**WAITT (Wait Timer)**

Funktion Warten bis der spezifizierte Zeitgeber abgelaufen ist.

Format WAITT <Wert1> <Wert1> 00 - 15 Zeitgebern timer

Beispiel WAITT 05

Erklärung Der Programmablauf verweilt solange an diesem Befehl, bis der durch
<Wert1> spezifizierte Zeitgeber abgelaufen ist.

Notizen

WAITZ**WAITZ**

Funktion Warten bis die Z-Achse die Höhe der Z SAFTEY ZONE erreicht hat.

Format WAITZ

Beispiel WAITZ

Erklärung Der Programmablauf verweilt solange an diesem Befehl, bis die Z-Achse die Höhe des Z SAFTEY ZONE Wert erreicht hat.

Dieser Befehl kann nur in Verbindung mit den C_MOV- und S_MOV-Befehlen verwendet werden.

Wird der Befehl aufgerufen, während der Roboter sich nicht bewegt, wird der Befehl ignoriert und der nächste Befehl wird ausgeführt.

Notizen

8 Programmbeispiel

In diesem Beispiel verfügt der Roboter über zwei Greifer und Z-Ausgleichseinheiten. Die Greifer sind an einem Schwenkmodul montiert. Der Roboter nimmt das Werkstück von einer Palette auf und platziert es auf einem Werkstückträger.

Die Ein- und Ausgänge sind wie folgt belegt.

Eingang	Funktion	Ausgang	Funktion
IN 08	Schwenkmodul Stellung 1	OUT 08	Greifer auf/zu
IN 09	Schwenkmodul Stellung 2	OUT 09	Schwenkmodul
IN 10	Ausgleichssenor	OUT 10	Startlampe
IN 11	Starttaste	OUT 11	Fehlerlampe
IN 12	Versorgung erlaubt	OUT 12	
IN 13	Aufnehmen erlaubt	OUT 13	
IN 14		OUT 14	
IN 15		OUT 15	

Tabelle 8.1

< Bedingungen >

- (1) OUT 08 EIN : Greifer auf
AUS : Greifer zu
- (2) OUT 09 EIN : Schwenkmodul Stellung 2
AUS : Schwenkmodul Stellung 1
- (3) Wenn der Eingang 10 während einer Pick- und Place-Operation eins wird, soll die Z-Achse des Roboters nach oben fahren. Durch Anlegen des Reset-Signals (IN 14), beginnt das Programm am Anfang.

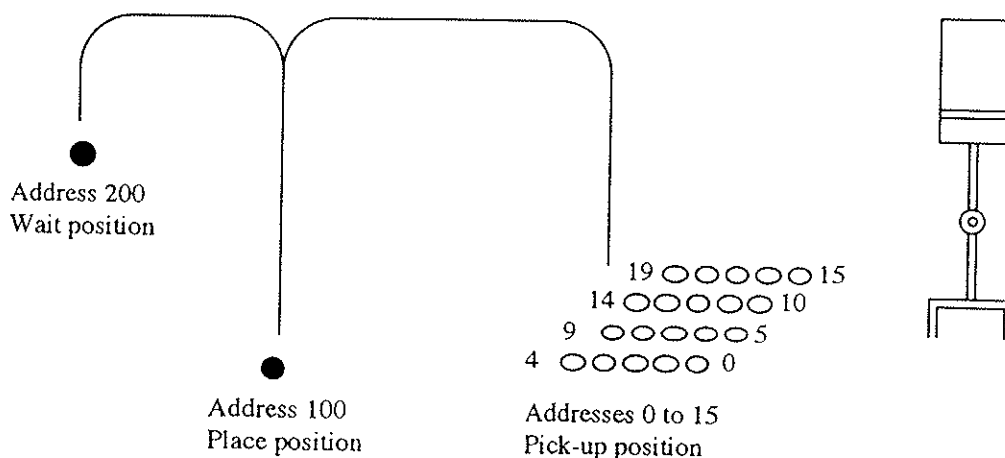


Bild 8.1 Bewegungsablauf

Listung des Hauptprogramms [MAIN 00] Werkstückbereitstellung

000	INTS	10	: Setze Z-Ausgleichssignal als Interruptsignal
001	SUB	15	: Aufrufen Unterprogramm 15 (Initialisierung)
002	SPEED	100	: Setze Robotergerwindigkeit
003	C_CHK	005	: Prüfe Kalibrierungszustand, wenn ok gehe zu 5
004	ACAL		: Kalibriervorgang durchführen
005	HOLD	5	: Servoverriegelung für alle Achsen an
006	NMOV	200	: Fahre auf Warteposition
007	WAIT1	11	: Warte auf Startsignal
008	E_ON	10	: Startlampe an
009	D_SET	00 000	: Setze Palettenzähler
010	D_SET	00 020	: Setze Wiederholungszähler für Pick-operation
011	SMOV	00	: Fahre zur Palettenposition
012	WAITZ		: Warte bis Z-Achse innerhalb Sicherheitszone
013	TIMER	0 00	: Setze Interval-Zeitgeber
014	HAND	00	: Aufrufen Handprogramm 00 (Schwenkmodul Stellung 1)
015	BRN1	08 020	: Ist Schwenkmodulsensor an?
016	BRNT	0 018	: Zeitgeber abgelaufen?
017	JUMP	015	: Springe zurück
018	E_ON	63	: Fehlermerker setzen
019	MVSTP		: Roboterbewegung unterbrechen
020	WAIT1	13	: Warte bis Aufnehmen erlaubt
021	FIN_Z		: Z-Achse freigeben
022	MVEND		: Warte bis Roboterbewegung fertig
023	BRN1	63 050	: Springe zu Fehlerroutine, wenn Merker 63 = 1
024	HAND	01	: Aufrufen Handprogramm 01 (Greifer zu)

025	D_INC	00	: Palettenzähler erhöhen
026	C_MOV	100	: Fahre zu Ablageposition
027	WAITZ		: Warte bis Z-Achse innerhalb Sicherheitszone
028	TIMER	0 00	: Setze Intervall-Zeitgeber
029	HAND	02	: Aufrufen Handprogramm 02 (Schwenkmodul Stellung 2)
030	BRN1	09 035	: Ist Schwenkmodulsensor an?
031	BRNT	0 033	: Zeitgeber abgelaufen?
032	JUMP	030	: Springe zurück
033	E_ON	63	: Fehlermerker setzen
034	MVSTP		: Roboterbewegung unterbrechen
035	WAIT1	12	: Warte bis Versorgung erlaubt
036	FIN_Z		: Z-Achse freigeben
037	MVEND		: Warte bis Roboterbewegung fertig
038	BRN1	63 050	: Springe zu Fehlerroutine, wenn Merker 63 = 1
039	HAND	03	: Aufrufen Handprogramm 03 (Greifer auf)
040	D_CMP	00 01 042	: Ist Palette abgearbeitet?
041	JUMP	10	: Rücksprung Hauptschleife
042	SUB	15	: Aufrufen Unterprogramm 15 (Greifer initialisieren)
043	N_MOV	200	: Fahre zur Warteposition
044	E_OFF	10	: Startlampe aus
045	END		: Programmende

Auflistung Hauptprogramm**Fehlerroutine**

050	Z_PUL		: Z-Achse fährt auf PULL-UP-Ebene
051	E_ON	11	: Fehlerlampe ein
052	WAIT1	14	: Warte auf Reset-Signal
053	E_OFF	11	: Fehlerlampe aus
054	E_OFF	63	: Fehlermerker zurücksetzen
055	JUMP	006	: Springe zum Programmanfang

Auflistung Unterprogramm [SUB 15]**Initialisierung**

000	HAND	00	: Aufrufen Handprogramm 00
001	HAND	01	: Aufrufen Handprogramm 01
002	E_OFF	10	: Startlampe aus
003	E_OFF	11	: Fehlerlampe aus
004	E_OFF	63	: Fehlermerker zurücksetzen
005	END		: Ende Unterprogramm

Auflistung Handprogramm [HAND 00]**Schwenkmodul Stellung 1**

000	E_OFF	09	: Ausgang 9 aus
001	END		: Ende Handprogramm

Auflistung Handprogramm [HAND 01]**Greifer auf**

000	E_ON	08	: Ausgang 8 ein
001	END		: Ende Handprogramm

Auflistung Handprogramm [HAND 02]**Schwenkmodul Stellung 2**

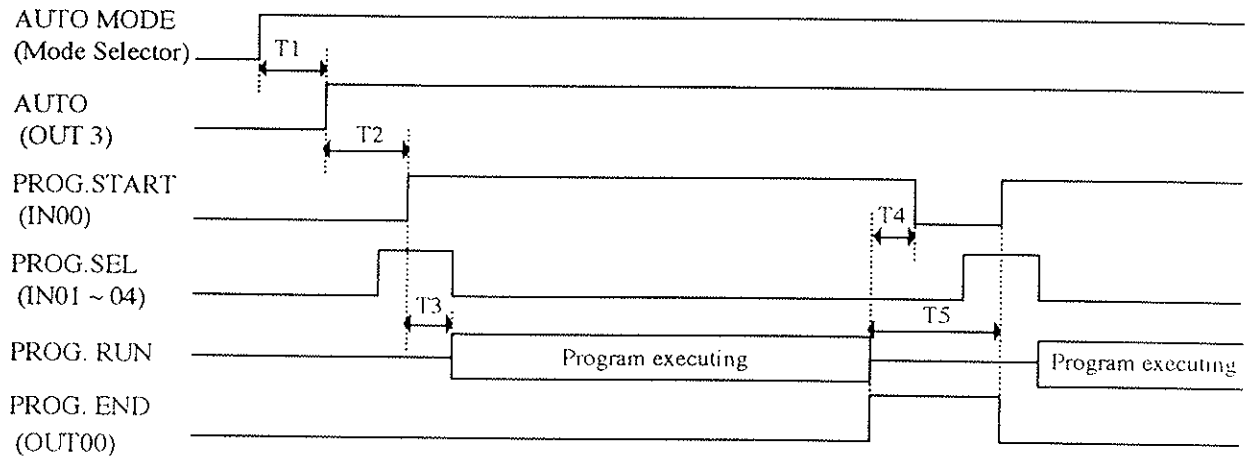
000 E_ON 09 : Ausgang 9 ein
001 END : Ende Handprogramm

Auflistung Handprogramm [HAND 03]**Greifer zu**

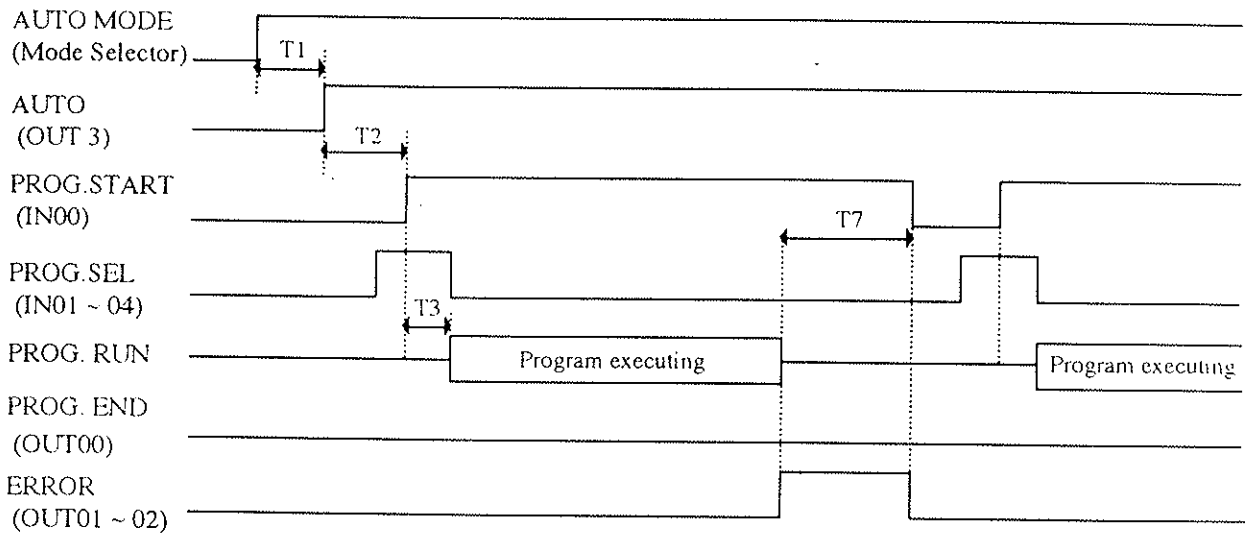
000 E_OFF 08 : Ausgang 8 ein
001 END : Ende Handprogramm

9 Zeitdiagramme

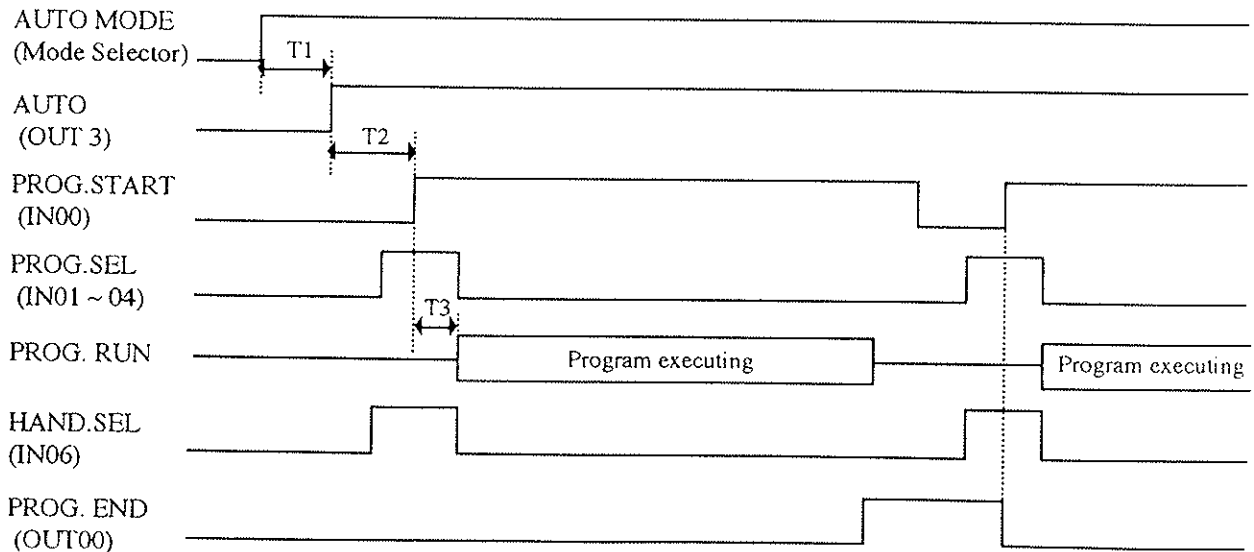
<Normal operation>



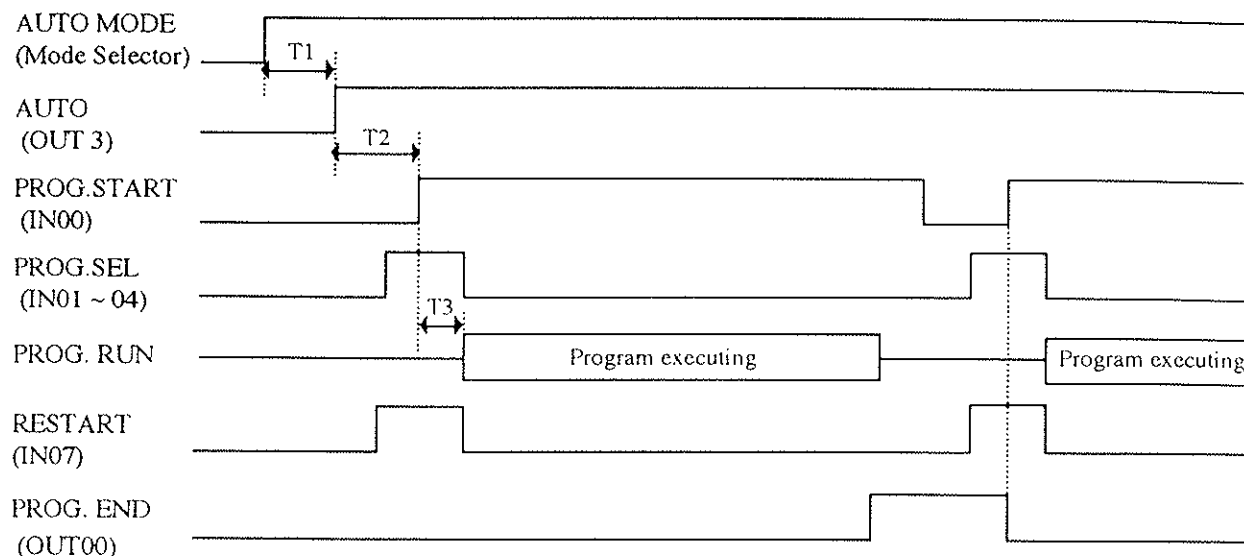
<Error occurance>



<Hand select>



<Restart>



- T1 : 1,3 s (2,5 s nach Netzspannung ein)
- T2 : 10 ms (nach Ausgang AUTO bis Programmstart akzeptiert wird)
- T3 : 20 ms (nach Programmstart bis zur ersten Befehlsausführung)
- T4 : 15 ms (nach Programmende bis zum Zurücksetzen des Programmstart)
- T5 : 50 ms (nach Programmende bis nächster Programmstart akzeptiert wird)
- T6 : 10 ms (zwischen Fehlererkennung und Setzen Fehlerausgang)
- T7 : 45 ms (nach Setzen Fehlerausgang bis Programmstart bestätigt wird)

Hinweis

1. Die Programmauswahlsignale (Handwahl- und Resetsignal) müssen angelegt werden, bevor das Programmstart-Signal angelegt wird.
2. Die Zeit T5 zwischen Programmende und neuem Programmstart sollte länger als 50 ms sein.
3. Der Ausgang "Programmende" wird zu Beginn des Programm-Start-Signals zurückgesetzt.
4. Der NOT AUS-Fehler wird durch die steigende Flanke des Programmstart-Signals nicht zurückgesetzt. Wenn NOT AUS zurückgesetzt wird, wird der Status am Beginn der Betriebsart "AUTO" initialisiert.
5. Die Ausgänge OUT08 bis OUT31 werden nicht zurückgesetzt solange die Netzspannung eingeschaltet bleibt.
6. Der Fehlerausgang, die freien Anwenderausgänge und der Programmende-Ausgang werden mit der steigenden Flanke des Programmstart-Signals zurückgesetzt.

10 Anhang

10.1 Ein- und Ausgangssignalverarbeitung während der Roboterbewegung

Dieses Kapitel beschreibt die Verarbeitung der Ein- und Ausgangssignale während einer Roboterbewegung. Diese Prozedur kann die Taktzeit einer Anwendung reduzieren, da während der Roboter zu einer Position fährt, Ein- und Ausgangssignale verarbeitet werden können.

Verfügt ein Roboter zum Beispiel über ein Schwenkmodul mit zwei Greifern, so kann das Schwenkmodul drehen, während der Roboter zur nächsten Position fährt.

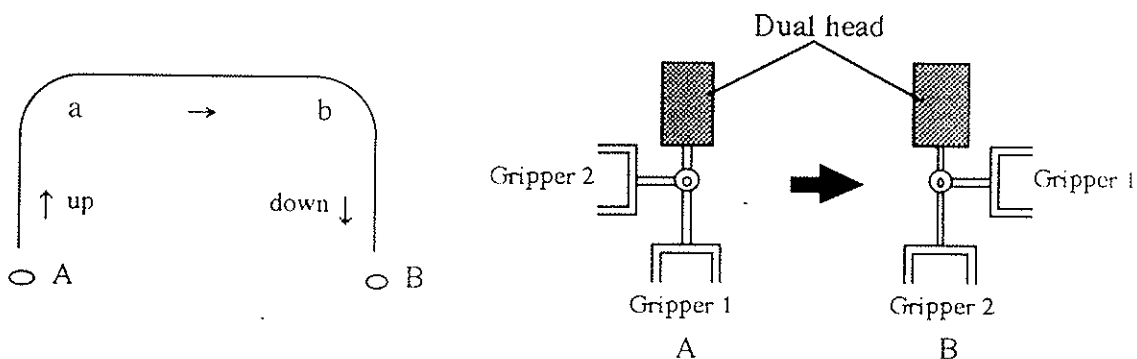


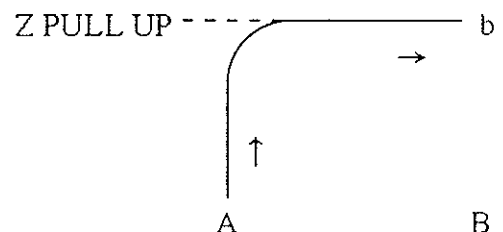
Bild 10.1

Wird der Greifer 1 an der Position "A" und der Greifer 2 an der Position "B" benutzt, so kann das Schwenkmodul mit der Schwenkbewegung beginnen, während die Z-Achse von "A" nach "a" fährt. Wenn die Drehbewegung noch nicht abgeschlossen ist, bevor der Roboter an der Position "b" angekommen ist, bleibt der Roboter in Position "b" stehen. Erst wenn der Schwenkvorgang abgeschlossen ist, fährt die Z-Achse nach unten.

Um diesen Vorgang zu programmieren, müssen die Befehle "S_MOV", "C_MOV", "FIN_Z" und "MVEND" benutzt werden.

Hinweise zur Ein- und Ausgangssignalverarbeitung während der Bewegung

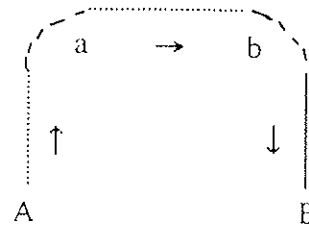
- (1) Mit dem S_MOV- oder C_MOV-Befehl wird der Roboter angewiesen, nur bis zur Position "b" zu fahren und die Z-Achse nicht nach unten (Position B) zu bewegen, solange die Drehung des Schwenkmoduls nicht abgeschlossen ist.



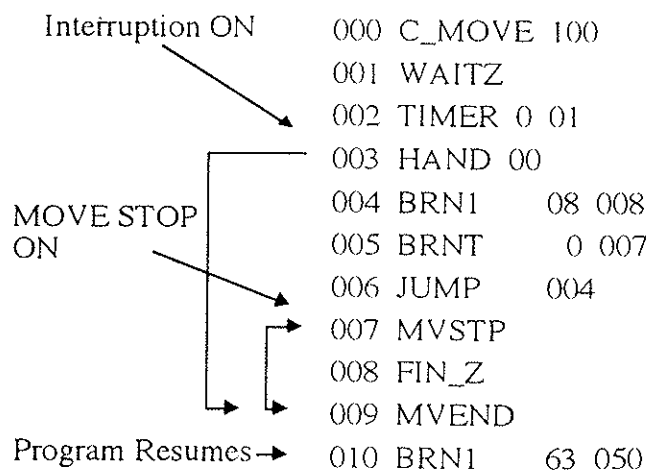
- (2) Nachdem durch den S MOV- oder C MOV-Befehl die Bewegung gestartet wurde, führt die Steuerung die nächsten Befehle im Programm aus.

- (3) Wird der FIN_Z-Befehl ausgeführt, fährt die Z-Achse nach unten.

Wurde der FIN_Z-Befehl während der Bewegung von "A" nach "b" ausgeführt, bleibt die Z-Achse nicht in der Position "b" stehen, sondern fährt sofort auf "B" weiter.



- (4) Wird während eines aktivierten C MOV- oder S MOV-Befehles ein MVSTP-Befehl ausgeführt, springt der Programmablauf direkt zu dem Befehl "MVEND" und der nachfolgende Programmschritt wird ausgeführt.



- (5) Wird während eines aktivierten C MOV- oder S MOV-Befehles das Interrupt-Signal gesetzt, springt der Programmablauf direkt zu dem Befehl "MVEND" und der nachfolgende Programmschritt wird ausgeführt.

Außerdem wird der Merker 63 gesetzt, so daß das Auftreten des Interrupts gespeichert wird.

Hinweis

Nach einem C MOV- oder S MOV-Befehl muß immer ein MVEND-Befehl benutzt werden.

10.2 Externe Positionsadressierung

Der folgende Abschnitt beschreibt, wie von einer externen Steuerung eine direkte Positionsadressierung der Roboterbewegung erfolgen kann und das Erreichen der vorgegebenen Position durch Ausgangssignale quittiert wird.

An den Eingängen 9 bis 15 wird die Adressierung vorgenommen. Die gewünschte Positionsadresse wird binär codiert dort angelegt und als gleicher Code nach Erreichen der Position wieder an den Ausgängen 9 bis 15 ausgegeben. Der Eingang 8 startet die Bewegung und der Ausgang 8 quittiert das Erreichen der Position.

Eingang		Ausgang	
IN 08	Start	OUT 08	Bewegung beendet
IN 09	Adressbit 0	OUT 09	Adressbit 0
IN 10	Adressbit 1	OUT 10	Adressbit 1
IN 11	Adressbit 2	OUT 11	Adressbit 2
IN 12	Adressbit 3	OUT 12	Adressbit 3
IN 13	Adressbit 4	OUT 13	Adressbit 4
IN 14	Adressbit 5	OUT 14	Adressbit 5
IN 15	Adressbit 6	OUT 15	Adressbit 6

Tabelle 10 Ein- /Ausgangsbelegung

Programm

000	WAIT1	08	Roboter bereit, wartet auf Startsignal
001	D_IN	01 09 03	Lade Daten für Positionsadressierung
002	P_MOV	01 00	Verfahre Roboter
003	E_ON	08	Positionierung beendet
004	D_OT	01 09 07	Gebe angefahrene Position aus
005	END		Programmende

In Schritt 000 wartet der Roboter auf das Start-Signal. Bevor das Start-Signal angelegt wird, muß die Adressencodierung an den Eingängen 9 - 15 erfolgt sein. Im obigen Beispiel werden jedoch nur die Adressen 9 bis 11 eingelesen.

09	10	11	12	13	14	15	Eingang
1	1	1	0	0	1	0	Ein / Aus
2^0	2^1	2^2	2^3	2^4	2^5	2^6	

Ist der Zustand der Eingänge wie in obiger Tabelle gezeigt, wird der Wert 7 in das Datenregister 01 geschrieben. Der Roboter fährt dann in Schritt 002 auf die Position, die in Positionsadress 007 abgespeichert worden ist. Nachdem der Roboter die Positionierung beendet hat, wird der Ausgang 8 gesetzt und die Codierung (7) der Positionsadresse an den Ausgängen 9 bis 15 ausgegeben.

10.3 Erweiterte Roboterfunktionen

Mit dieser Funktion kann der Bewegungsablauf des Roboters während der Roboterbewegung geändert werden. Der <Wert2> der Befehle C_MOV, N_MOV, P_MOV und S_MOV hat folgenden Einfluß auf den Bewegungsablauf.

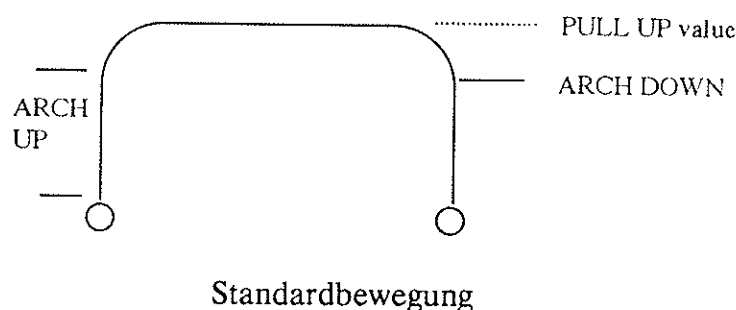
Wert 2	Beschreibung
00	Normal
01	Positionskalibrierung durch Visionsystem
02	Positionsoffset durch Offset-Tabelle
04	Bewegungsänderung

10.3.1 Positionskalibrierung durch Visionsystem

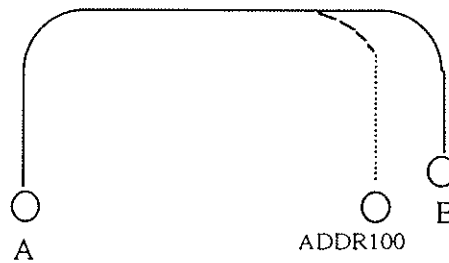
Dieser Funktionswert kann eingesetzt werden, wenn das Robotersystem mit einem Hirata-Visionsystem gekoppelt ist. Setzen Sie die Szenennummer (SCENE) in der Vision-Steuerung mit dem VSCE-Befehl. Weitere Informationen erhalten Sie darüber im Bedienungshandbuch des Visionsystems.

10.3.2 Positionsverschiebung durch Offset-Tabelle

Diese Funktion kann dazu benutzt werden, den Roboter relativ zu einer Zielposition zu positionieren. Der Betrag der Verschiebung (Offset) muß vorher berechnet und in die Offset-Tabelle eingegeben werden. Siehe Kapitel 3.5 und 3.8.



Tabellenwert = 00 Anwahl des Datenregisters, dessen Inhalt die Offset-Tabellennummer darstellt



Verschiebebewegung

Tabellenwert = 00

Die Koordinatenwerte der Offset-Tabelle 00 werden wie folgt gesetzt:

A = 100.00 B = 150.00
Z = -10.00 W = 30.00

Sind die obigen Daten programmiert und folgende Befehle werden ausgeführt,

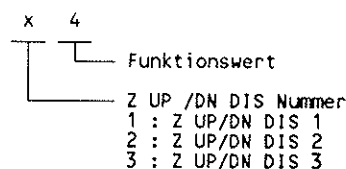
D_SET 00 000	Anwahl der Offset-Tabelle
N_MOV 100 02	Fahre zur Offset-Position

dann verfährt der Roboter von Position A zur Position B, die sich relativ um die obengenannten Werte zur Position der Adresse 100 ergibt.

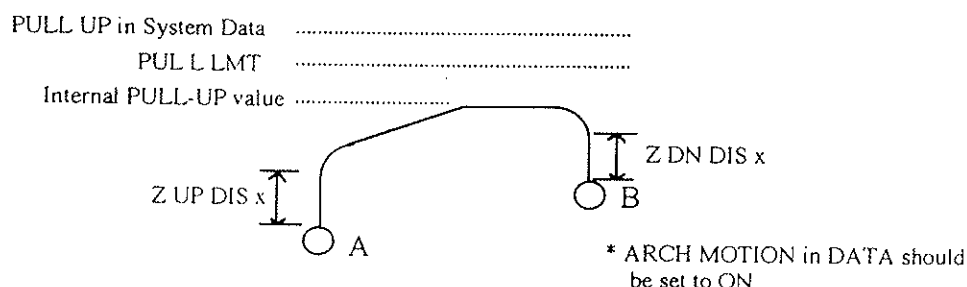
10.3.3 Bewegungsänderung

Mit dieser Funktion kann der Roboter auf der kürzesten Bahn zwischen seiner momentanen und Zielposition bewegt werden. Sämtliche bewegungsspezifische Systemparameter werden in diesem Falle ignoriert. Diese spezielle Bewegung wird durch die Parametersatz des Untermenüpunkte UTIL / DATA SET bestimmt. Siehe Kapitel 3.8.

Die Parameter Z UP/DN DIS 1 bis 3 werden folgendermaßen ausgewählt:



Wird der Funktionswert auf 24 gesetzt, sind die Werte der Parameter Z UP DIS 2 und Z DN DIS 2 aktiv und der Roboter bewegt wie in nachfolgender Skizze dargestellt.



Schema der Roboterbewegung bei Funktionswert x4

Hinweise

1. Der interne PULL-UP-Wert wird wie folgt berechnet:

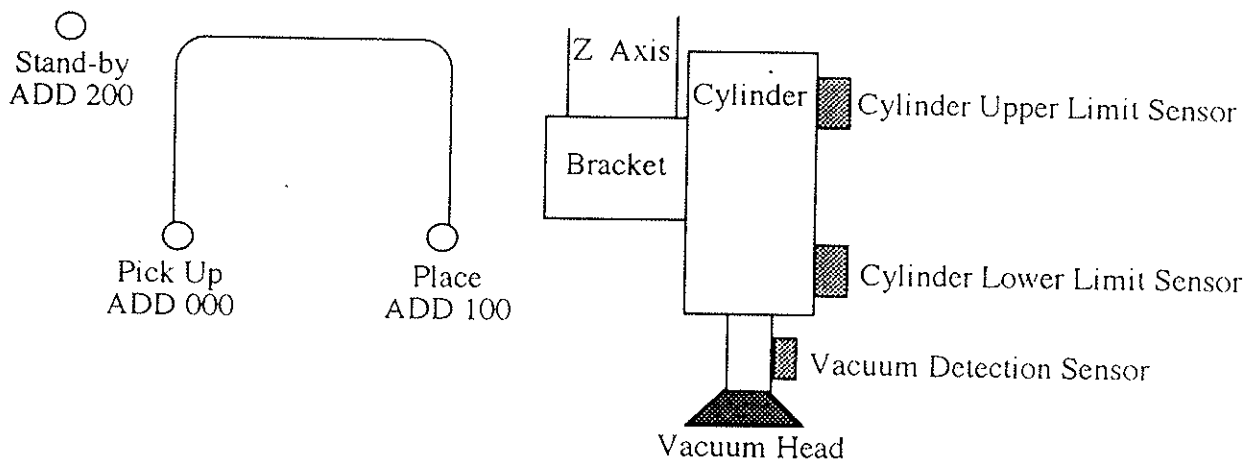
$$\text{Z-Achsen-Zielposition von B} - (\text{ZUP/DN DIS} \times \text{PULL UP RATE})$$
2. Der interne PULL-UP-Wert wird nicht über den Wert von PUL L LMT gesetzt.
3. Der Standardwert für den Wert von PULL UP RATE ist 2. Setzen Sie den Wert zwischen 1.0 und 2.0.
4. wird die Z UP/DN DIS Nummer größer 3 gesetzt, wird Z UP/DN DIS 1 benutzt.
5. Funktionswerte 02 und 04 können gleichzeitig benutzt werden. Benutzen Sie dann als Funktionswerte 16, 26, oder 36 für den entsprechenden Verfahrbefehl.
6. Funktionwert 01 kann nicht mit anderen Funktionen kombiniert werden.
7. Die Parameterwerte für die Funktionswerte 02 und 04 müssen vorher in die Offset-Tabelle eingetragen werden.

10.4 Anwendungsbeispiel

10.4.1 Beschreibung des Bewegungsablaufes

Dieser Abschnitt beschreibt eine Programmierbeispiel in HARL_U1 unter folgenden Bedingungen:

1. Der Roboter nimmt ein Teil auf und placiert es, nachdem er das Signal "Zyklusstop" erhalten hat.
2. Das Greifersystem besteht aus einem Kurzhubzylinder und einem Vakuumgreifer, das an der Z-Achse des Roboters befestigt ist.
3. Der Kurzhubzylinder nimmt das Teil an der unteren Position auf oder setzt es dort ab.
4. Positionsbelegung und Greiferkonfiguration entsprechen nachstehender Skizze,

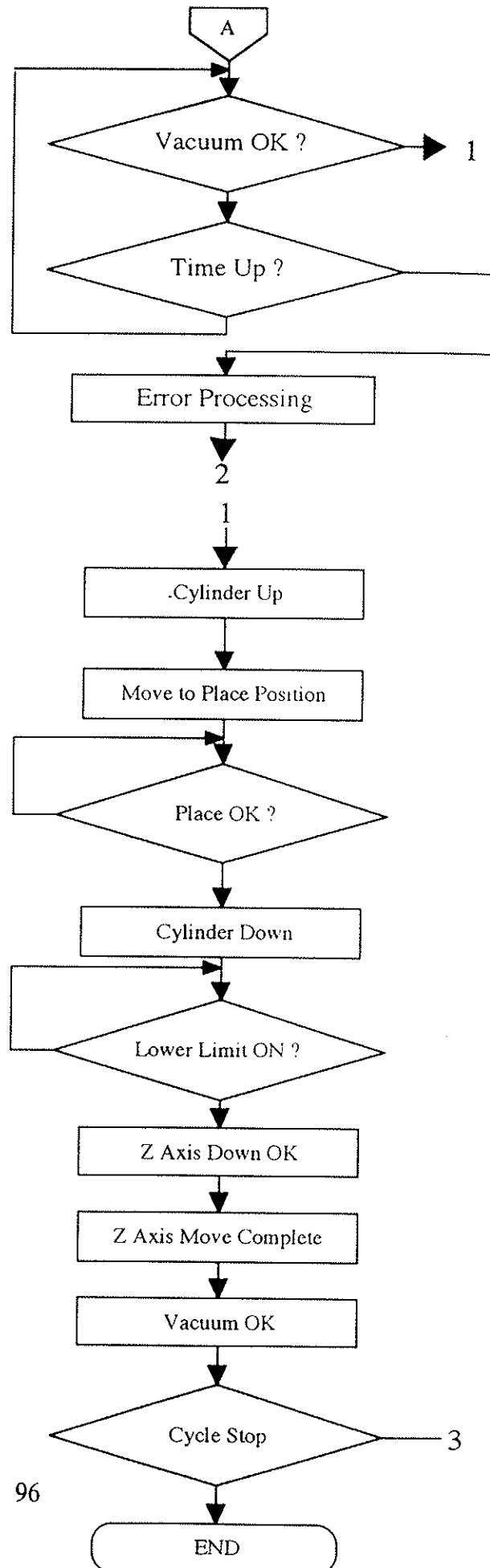
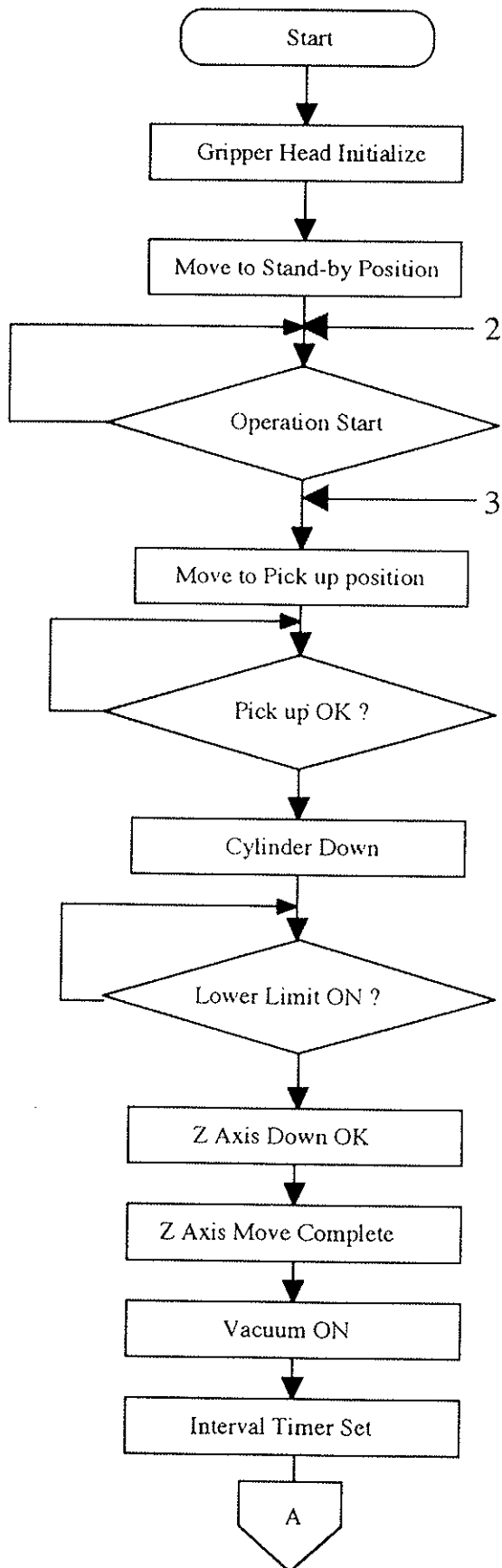


10.4.2 Ein- und Ausgangsbelegung

Eingang	Bedeutung	Ausgang	Bedeutung
08	Start	08	In Betrieb
09	Teil aufgenommen ok	09	Fehler
10	Teil abgelegt ok	10	Kurzhubzylinder ausfahren
11	Reset	11	Vakuum ein
12	Kurzhubzylinder oben	12	
13	Kurzhubzylinder unten	13	
14	Vakuum ok	14	
15	Zyklusstop	15	

10.4.3

Flußdiagramm



10.4.4 Beispielprogramm

(1) Hauptprogramm (MAIN 00 : Teil aufnehmen und absetzen)

000	SUB	10	: Greifersystem initialisieren
001	N_MOV	200 00	: Auf Stand-By-Position fahren
002	E_ON	08	: Ausgang "in Betrieb" ein
003	WAIT1	08	: Warte auf Start-Signal
004	C_MOV	000	: Auf Aufnahmeposition fahren
005	WAIT1	09	: Warte bis Aufnehmen ok ist
006	HAND	00	: Unterprogramm Zylinder abwärts
007	WAIT1	13	: Warte bis Zylinder unten ist
008	FIN_Z		: Warte ist Z-Achse unten ist
009	MVEND		: Bewegung beendet
010	HAND	02	: Unterprogramm Vakuum ein
011	TIMER	0 01	: Intervallzeitgeber setzen
012	BRNT1	14 017	: Abfrage Vakuum ok?
013	BRNT	0 015	: Überwachungszeit abgelaufen?
014	JUMP	012	: Wiederhole Abfrage
015	SUB	00	: Fehlerroutine
016	JUMP	012	: Wiederhole Abfrage
017	HAND	01	: Unterprogramm Zylinder aufwärts
018	C_MOV	100 00	: Fahre zur Ablageposition
019	WAIT1	10	: Warte bis Ablegen ok ist
020	HAND	00	: Unterprogramm Zylinder abwärts
021	WAIT1	13	: Warte bis Zylinder unten ist

	022	FIN_Z		: Z-Achsenbewegung ok
	023	MVEND		: Bewegung beendet
	024	E_OF	08	: Ausgang "In Betrieb" aus
	025	BRN0	15 003	: Prüfe: Zyklusende
	026	END		: Programmende
(2)	Unterprogramm	(SUB 10		: Greifersystem initialisieren)
	000	HAND	01	: Unterprogramm Zylinder aufwärts
	001	HAND	03	: Unterprogramm Vakuum aus
	002	END		: Unterprogramm Ende
(3)	Unterprogramm	(SUB 00		: Fehlerbehandlung)
	000	ACAL		: Roboterkalibrierung ausführen
	001	HAND	01	: Unterprogramm Zylinder aufwärts
	002	E_ON	09	: Ausgang "Fehler" ein
	003	WAIT1	11	: Warte auf Reset
	004	E_OFF	09	: Ausgang "Fehler" aus
	005	END		
(4)	Handprogramm	(HAND 00		: Zylinder abwärts)
	000	E_OFF	10	: Ausgang aus
	001	END		
(5)	Handprogramm	(HAND 01		: Zylinder aufwärts)
	000	E_ON	10	: Ausgang ein
	001	END		

(6) Handprogramm (HAND 02 : Vakkum ein)
000 E_ON 11 : Ausgang ein
001 END

(7) Handprogramm (HAND 03 : Vakkum aus)
000 E_OFF 11 : Ausgang aus
001 END

10.5 Anwendung des R_MOV-Befehls

10.5.1 Warnung - Wichtige Hinweise zur Benutzung des Befehls

9

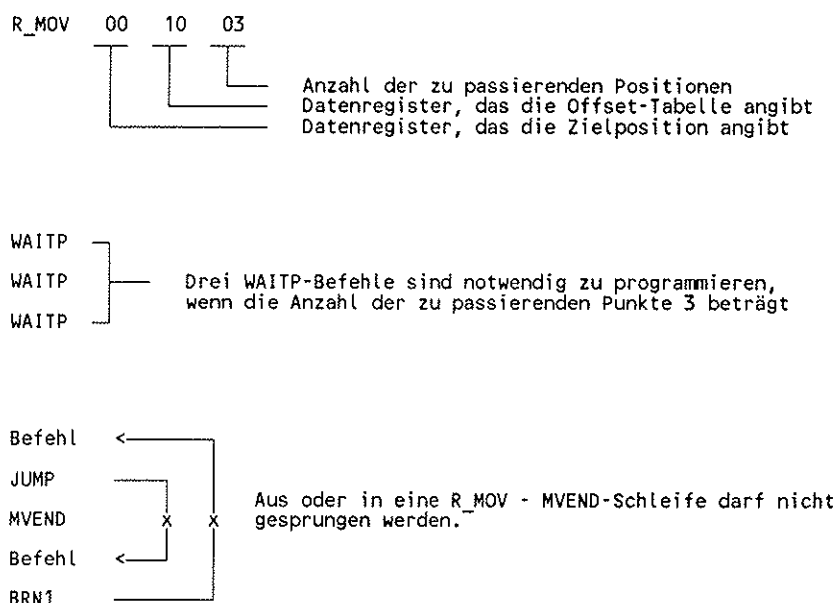
Der R_MOV-Befehl kann zur schnelleren Palettierung verwendet werden. Der Roboter bewegt sich dabei auf einer Bahn zur Zielposition, die durch Parameter in einer Offset-Tabelle erzeugt wird. Außerdem ist die Bedienung von Ein- und Ausgängen während der Bewegung möglich.

Wenn das Programm oder Programmdatei bei der Benutzung des R_MOV-Befehls falsch oder unvollständig sind, kann es passieren, daß der Roboter zu unerwarteten Position fährt. Wenn Sie den R_MOV-Befehl benutzen wollen, lesen Sie dieses Kapitel sorgfältig durch.

Bitte beachten Sie nachfolgende Hinweise bei der Verwendung des R_MOV-Befehls.

- (1) Der R_MOV-Befehl darf immer nur im Zusammenhang mit dem WAITP- und MVEND-Befehl benutzt werden.
- (2) Der Programmablauf darf nie zwischen den ersten R_MOV-Befehl und dem MVEND-Befehl springen.
- (3) Der WAITP-Befehl muß für jede erzeugte Position zwischen der R_MOV- und MVEND-Schleife eingegeben werden.
- (4) Geben Sie der Werte der benutzten Offset-Tabelle zuerst ein, bevor Sie den R_MOV-Befehl benutzen.

Befehls-Beispiel



10.5.2 Programmbeispiele

(1) Schritt 1 Erstellen der Positionsdaten

Wenn der R MOV-Befehl verwendet werden soll, ist es nötig, die entsprechenden Positionsdaten zu erstellen und in die entsprechende Offset-Tabelle abzulegen. Das folgende Beispiel zeigt, wie sich der Roboter unter Zuhilfenahme des R_MOV-Befehls bewegt.

< Dateneingabe >

Datenregister 6 = 100

Datenregister 8 = 10

Koordinaten der Positionsadresse 100 :

X = 100.00 Y = 200.00 Z = 85.00 W = 150.00

Offset-Tabelle 10 X = -10.00 Y = -20.00 Z = -75.00 W = 0.00

Offset-Tabelle 11 X = -10.00 Y = -30.00 Z = -50.00 W = 0.00

Offset-Tabelle 12 X = -0.00 Y = -0.00 Z = -0.00 W = 0.00

Offset-Tabelle 13 X = -0.00 Y = -0.00 Z = -50.00 W = 0.00

< Programm >

R MOV 06 08 04

WAITP

WAITP

WAITP

WAITP

MVEND

< Roboterbewegung >

Vier Positionsdaten werden erzeugt unter Zuhilfenahme der Offset-Tabellen 10 - 13, wobei als Ausgangspunkt die Koordinaten der Positionsadresse 100 herangezogen werden. Die entsprechende Zuweisung ist in Datenregister 6 gespeichert. Danach bewegt sich der Roboter in einer Verschleißbewegung zu diesen errechneten Positionen. Die Hilfspositionen werden wie folgt berechnet.

Koordinaten der Referenzposition + Koordinaten der Offset-Tabelle

Im Falle des obigen Beispiels ergeben sich folgende Positionen:

P1 : X = 90.00 Y = 180.00 Z = 10.00 W = 150.00

P2 : X = 90.00 Y = 170.00 Z = 35.00 W = 150.00

P3 : X = 100.00 Y = 200.00 Z = 85.00 W = 150.00

P4 : X = 100.00 Y = 200.00 Z = 35.00 W = 150.00

Bitte beachten Sie, daß der M- und S-Code von P1 bis P4 automatisch auf M=39 und S=00 gesetzt werden und daß der F-Code von P1 bis P4 gleich der von der Referenzadresse (PM100) ist.

(2) Schritt 2 Kontinuierliche Roboterbewegung

Werden mehrere R MOV-befehle programmiert, bewegt sich der Roboter kontinuierlich ohne seine Bewegung zu unterbrechen. In diesem Fall sollte der R MOV-Befehl in zwei Schritten programmiert werden, bevor die Positionen erzeugt werden.

Das folgende Beispiel zeigt eine kontinuierliche Palettierbewegung, bei der der Roboter die Werkstücke von einem laufenden Transportband aufnimmt und auf einer Palette ablegt.

< Dateneingabe >

Geteachte Positionen

PM 001 (Transportband) X = 100.00 Y = 200.00 Z = 200.00 W = 150.00

PM 100 (Palette) X = 500.00 Y = 400.00 Z = 150.00 W = 100.00

Auf PM 100 - 110 liegen die Palettenpositionen, PM 100 ist die Anfangsposition der Palette.

Offset-Tabelle für die Transportbandpositionen

Offset-Tabelle 10 X = -10.00 Y = -20.00 Z = -100.00 W = 0.00

Offset-Tabelle 11 X = -10.00 Y = -30.00 Z = -50.00 W = 0.00

Offset-Tabelle 12 X = 0.00 Y = 0.00 Z = 0.00 W = 0.00

Offset-Tabelle 13 X = 10.00 Y = 20.00 Z = -70.00 W = 0.00

Offset-Tabelle 14 X = 10.00 Y = 30.00 Z = -90.00 W = 0.00

Offset-Tabelle für die Palettenpositionen

Offset-Tabelle 20 X = -10.00 Y = -20.00 Z = -50.00 W = 0.00

Offset-Tabelle 21 X = -10.00 Y = -30.00 Z = -25.00 W = 0.00

Offset-Tabelle 22 X = 0.00 Y = 0.00 Z = 0.00 W = 0.00

Offset-Tabelle 23 X = 0.00 Y = 0.00 Z = -100.00 W = 0.00

Ein- / Ausgänge

IN 08 : Vakuummelder OUT 08 : Vakuum ein

Datenregister

DT 00 Adressen am Transportband

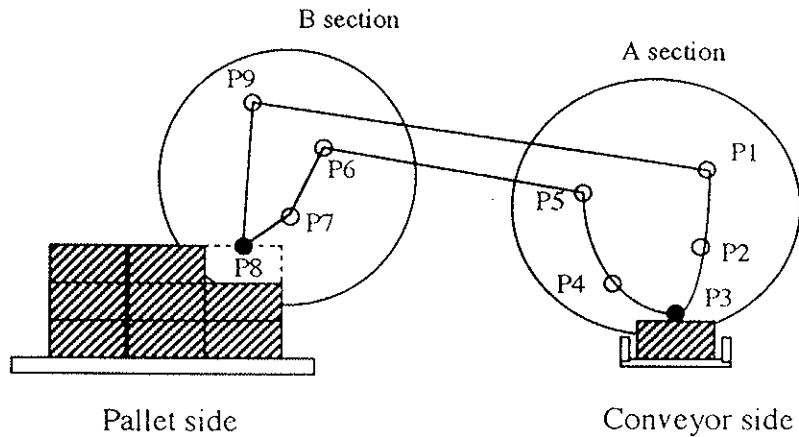
DT 01 Startadresse der Palette

DT 02 Endadresse der Palette

DT 03 Offset-Tabelle am Transportband

DT 04 Offset-Tabelle auf Palette

< Roboterbewegung >



- P1 : Startposition über Transportband
- P2 : Verschleifposition zum Abgreifen
- P3 : Abgreifposition
- P4 : Vakuumüberprüfungsposition
- P5 : Enposition des Abgreifvorgangs
- P6 : Startposition der Palettierung
- P7 : Verschleifposition zum Palettieren
- P8 : Vakuumabschaltposition
- P9 : Ablageposition

< Programmbeispiel >

000	ACAL		: Roboterkalibrierlauf
001	SPED	100	: Geschwindigkeit setzen
[C1] 002	D_SET	00 001	: Anfangsadresse Transportband
003	D_SET	01 100	: Anfangsadresse Palette
004	D_SET	02 110	: Endadresse Palette
005	D_SET	03 010	: Offset-Tabelle Transportband
006	D_SET	04 020	: Offset-Tabelle Palette
007	NOP		
008	NOP		
009	NOP		

[C2]	010	R_MOV	00 03 05	: Erzeugt Positionsdaten im A-Bereich und bewegt Roboter
	011	WAITP		: Programm verweilt, bis Position P1 erreicht
	012	NOP		
	013	WAITP		: Programm verweilt, bis Position P2 erreicht
[C3]	014	R_MOV	01 04 04	: Erzeugt Positionsdaten im B-Bereich und bewegt Roboter
	015	WAITP		: Programm verweilt, bis Position P3 erreicht
	016	E_ON	09	: Vakuum einschalten
	017	WAITP		: Programm verweilt, bis Position P4 erreicht
	018	BRN1	08 021	: Prüfe Vakuum
	019	E_ON	32	: Vakuumfehlermerker setzen
	020	MVSTP		: Unterbreche Roboterbewegung
	021	WAITP		: Programm verweilt, bis Position P5 erreicht
	022	NOP		
	023	WAITP		: Programm verweilt, bis Position P6 erreicht
	024	NOP		
	025	WAITP		: Programm verweilt, bis Position P7 erreicht
[C4]	026	R_MOV	00 03 05	: Erzeugt Positionsdaten im A-Bereich und bewegt Roboter
	027	WAITP		: Programm verweilt, bis Position P8 erreicht
	028	E_OF	09	: Vakuum ausschalten
	029	WAITP		: Programm verweilt, bis Position P9 erreicht
[C5]	030	D_INC	01	: Erhöhe Palettenpositionszähler
[C6]	031	D_CMP	00 02 033	: Vergleiche Palettenposition mit max. Position
[C7]	032	JUMP	011	: Wiederhole Vorgang
[C8]	033	MVSTP		: Unterbreche Roboterbewegung
	034	MVEND		: Zyklusende

Bemerkungen:

- [C1] : Initialisierung des Programms Schritt 2 bis 6
- [C2] : Erzeugt die Positionen im Bereich A und bewegt den Roboter entsprechend
- [C3] : Erzeugt die Positionen im Bereich B der Position P2 im voraus
- [C4] : Erzeugt die Positionen im Bereich A der Position P7 im voraus
- [C5] : Erhöht den Palettenpositionszähler um eins
- [C6] : Vergleicht die momentanen Stand des Palettenzählers mit der max. zulässigen
- [C7] : Springt zurück und wiederholt den Zyklus
- [C8] : Stoppt die Roboterbewegung

(3) Schritt 3 Palettierprogramm mit dem PALT-Befehl

Der PALT-befehl ist sehr hilfreich, wenn verschiedene Palettierbilder abgearbeitet werden müssen. Jede Paletteninformation (Lagenanzahl, Gesamtanzahl der Teile, Anzahl der Teile pro Lage und andere Anwendungs-daten) müssen dann vorher in die Paletten-Tabelle eingegeben werden. (siehe Kapitel 3.10.2)

Bis zu 32 verschiedene Paletteninformationen können in eine Paletten-Tabelle eingegeben werden.

< Dateneingabe >

Datenregister 5 : Nummer der Paletten-Tabelle
Inhalt der Paletten-Tabelle 12:

1	: Gesamtschritte	: 2
2	: Gesamtanzahl	: 12
3	: 1. Schrittnummer	: 6
4	: Anwenderwert 1	: 001
5	: Anwenderwert 2	: 100
6	: Anwenderwert 3	: 110
7	: Anwenderwert 4	: 010
8	: Anwenderwert 5	: 020

< Programmbeispiel >

Der [C1]-Abschnitt im obigen Programmbeispiel kann ab Schritt 2 durch den PALT-Befehl folgendermaßen ersetzt werden:

[C1]	002	D_IN	05 10 04	: Lade Paletteninformation über Eingangsport in Register 5
	003	PALT	05 04 00	: Anfangsadresse Transportband
	004	PALT	05 05 01	: Anfangsadresse Palette
	005	PALT	05 06 02	: Endadresse Palette
	006	PALT	05 07 03	: Offset-Tabelle Transportband
	007	PALT	05 08 04	: Offset-Tabelle Palette
	008	NOP		
	009	NOP		

(4) Hinweise

- 1) Wenn der R_MOV-Befehl benutzt wird, fährt der Roboter auf die berechneten Hilfspositionen in PTP zu. Er erreicht diese aber nicht exakt, sondern verschleift diese zur nächsten.
- 2) Stimmt die Anzahl der erzeugten Positionen nicht mit der der programmierten WAITP-Befehle überein, fährt der Roboter zu der berechneten Position und führt dann den Befehl aus, der nach dem MVEND-befehl steht.