

Handbuch Steuerung Teach-Programmierung

Andras Steuerungssysteme GmbH
Westerstraße 93A
28199 Bremen



**ANDRAS
SYSTEM**

Wir finden den Weg

- Irrtümer und Änderungen vorbehalten -

Ergänzungen / Änderungen

- Sicherheitsräume
- Systemmasse
- Systemmasse der Palettierung
- Palettierung substituieren
- Korrektur von nicht eindeutigen Textpassagen in der Befehlsbeschreibung
- Korrektur in den Sicherheitsräumen der Systemmasse
- Überarbeitung der Befehlstexte
- Dokumentation der Anzeigeform für Palettierbefehle
- Funktion „Konfig-Text --> SPS-Text“
- Funktion „Import von Palettiertabellen aus CSV“
- Dokumentation der Rechen-Funktionen für Palettiertabellen
- Befehle 68 und 69
- Ergänzung im Kapitel Programmbezeichnung
- Ergänzung im Kapitel Konfiguration
- Hinweis zum Testen unter Stop passiv
- Befehle 30, 64, 65, 72 und 80
- Onlinehilfe im Teach-In
- Geschwindigkeitstabelle

Inhaltsverzeichnis

Ergänzungen / Änderungen.....	2
Inhaltsverzeichnis.....	3
Abbildungsverzeichnis.....	5
Vorwort.....	6
Grundlagen der Teach-Programmierung.....	7
Programme und Teachebenen.....	7
Zeitverhalten.....	7
Konfiguration.....	8
Genereller Aufbau einer Zeile.....	10
Einzeilige und zweizeilige Darstellung.....	10
Bedeutung und Editierung der einzelnen Felder.....	11
Verschlüsselung von Palettier tabellen.....	12
Bedeutung der Option Schrittweise.....	12
Bedeutung der Option Override.....	13
Sequenzprogrammierung.....	14
Palettierung.....	14
Systemmasse der Palettierung.....	15
Palettierung substituieren.....	16
Darstellung des Felds Position innerhalb der Teach-Programmierung bei Palettierbefehlen.....	17
Rechnen innerhalb von Palettier tabellen.....	17
Importieren von Palettier tabellen aus einer CSV-Datei.....	18
Editierung von Teach-Programmen.....	19
Maske zur reinen Programmierung.....	20
Zweiteilige Arbeitsmaske.....	21
Hinweis zum Testen unter Stop passiv.....	23
Onlinehilfe in der zweiteiligen Arbeitsmaske.....	23
Teach-In Funktion.....	24
Befehlsbeschreibung der Teach-Programmierung.....	25
Befehl 0: Fahre auf Mass / Starte Ablauf mit Auftrag.....	26
Befehl 1: Fahre auf Mass aus Palettierung / Starte Ablauf mit Auftrag aus Palettierung.....	27
Befehl 2: Fahre auf Mass / Starte Ablauf mit Auftrag.....	28
Befehl 3: Fahre auf Mass aus Palettierung / Starte Ablauf mit Auftrag aus Palettierung.....	29
Befehl 6: Sprung wenn Istwert < Mass / Sprung wenn Step > Mass.....	30
Befehl 7: Sprung wenn Istwert < Mass aus Palettierung / Sprung wenn Step > Mass aus Palettierung.....	31
Befehl 8: Sprung wenn Istwert > Mass / Sprung wenn Step < Mass.....	32
Befehl 9: Sprung wenn Istwert > Mass aus Palettierung / Sprung wenn Step < Mass aus Palettierung.....	33
Befehl 10: Sprung wenn Istwert = Mass / Sprung wenn Step = Mass.....	34
Befehl 11: Sprung wenn Istwert = Mass aus Palettierung / Sprung wenn Step = Mass aus Palettierung.....	35
Befehl 12: Sprung wenn Istwert <> Mass / Sprung wenn Step <> Mass.....	36
Befehl 13: Sprung wenn Istwert <> Mass aus Palettierung / Sprung wenn Step <> Mass aus Palettierung.....	37
Befehl 20: Verfahre relativ um Mass, gekappte Kurve.....	38
Befehl 21: Verfahre relativ um Mass aus Palettierung, gekappte Kurve.....	39
Befehl 22: Verfahre relativ um Mass, komprimierte Kurve.....	40
Befehl 23: Verfahre relativ um Mass aus Palettierung, komprimierte Kurve.....	41
Befehl 24: Fahre auf Mass mit reduziertem Strom.....	42
Befehl 25: Fahre auf Mass aus Palettierung mit reduziertem Strom.....	43
Befehl 26: Sprung wenn innerhalb Sollwertfenster Mass.....	44
Befehl 27: Sprung wenn innerhalb Sollwertfenster Mass aus Palettierung.....	45
Befehl 28: Sprung wenn ausserhalb Sollwertfenster Mass.....	46
Befehl 29: Sprung wenn ausserhalb Sollwertfenster Mass aus Palettierung.....	47
Befehl 30: Kurve laden.....	48
Befehl 36: Sprung wenn innerhalb Startwertfenster Mass.....	48
Befehl 37: Sprung wenn innerhalb Startwertfenster Mass aus Palettierung.....	49
Befehl 38: Sprung wenn ausserhalb Startwertfenster Mass.....	50
Befehl 39: Sprung wenn ausserhalb Startwertfenster Mass aus Palettierung.....	51
Befehl 40: Fahre Slave auf Mass, Rampe gekappt.....	52
Befehl 41: Fahre Slave auf Mass aus Palettierung, Rampe gekappt.....	52
Befehl 42: Fahre Slave auf Mass, Rampe komprimiert.....	52
Befehl 43: Fahre Slave auf Mass aus Palettierung, Rampe komprimiert.....	52
Befehl 44: Fahre Slave auf Mass, Strom reduziert.....	53
Befehl 45: Fahre Slave auf Mass aus Palettierung, Strom reduziert.....	53
Befehl 46: Warte bis Slave-Position < Mass.....	53
Befehl 47: Warte bis Slave-Position < Mass aus Palettierung.....	53
Befehl 48: Warte bis Slave-Position > Mass.....	54
Befehl 49: Warte bis Slave-Position > Mass aus Palettierung.....	54
Befehl 50: Lösche Ausgang Mass im Slave.....	54

Befehl 51: Lösche Ausgang Mass aus Palettierung im Slave	54
Befehl 52: Setze Ausgang Mass im Slave	55
Befehl 53: Setze Ausgang Mass aus Palettierung im Slave	55
Befehl 54: Warte bis Eingang im Slave low ist	55
Befehl 55: Warte bis Eingang aus Palettierung im Slave low ist	55
Befehl 56: Warte bis Eingang im Slave high ist	56
Befehl 57: Warte bis Eingang aus Palettierung im Slave high ist	56
Befehl 58: Warte Mass Millisekunden	56
Befehl 59: Warte Mass aus Palettierung Millisekunden	56
Befehl 60: Lösche das Programm im Slave	57
Befehl 61: Reset des Programms im Slave	57
Befehl 62: Starte das Programm im Slave	57
Befehl 63: Warte auf das Programm im Slave	57
Befehl 64: Kurve einkuppeln	59
Befehl 65: Kurve auskuppeln	59
Befehl 68: Addiere Mass auf den nachfolgenden Befehl	59
Befehl 69: Addiere Mass aus Palettierung auf den nachfolgenden Befehl	60
Befehl 71: Teachebene aktiv?	60
Befehl 72: Teilenummer minus 1	61
Befehl 73: Teachebene starten	61
Befehl 74: Teachebene stoppen	61
Befehl 75: Warte bis Zeile >	62
Befehl 76: Warte bis Zeile <	62
Befehl 77: Teilenummer löschen	63
Befehl 78: Teilenummer plus 1	63
Befehl 79: Tabelle umschalten	63
Befehl 80: Bahn umrechnen	64
Befehl 81: Sprung wenn 2. Teachebene aktiv	64
Befehl 82: 2. Teachebene starten, Stop passiv	64
Befehl 83: Startbedingung löschen	65
Befehl 84: Ablaufprogrammierung löschen	65
Befehl 87: Stop-Abfrage aktiv	65
Befehl 88: Stop-Abfrage passiv	65
Befehl 89: Unterbrechnung nach Zeile	65
Befehl 90: Stop-Uhr starten	66
Befehl 91: Stop-Uhr Restart	66
Befehl 92: Stop-Uhr anzeigen	66
Befehl 93: 2. Teachebene starten	66
Befehl 94: 2. Teachebene stoppen	66
Befehl 95: Warte bis Zeile >	67
Befehl 96: Warte bis Zeile <	67
Befehl 97: Teilenummer löschen	67
Befehl 98: Teilenummer plus 1	67
Befehl 99: Sprung nach Zeile	67
Sonstiges zur Teach-Programmierung	69
Programmbezeichnung	69
Programmverwaltung	70
Gruppenzuordnung der Teachebenen	72
Handfunktionen	73
Teilenummern editieren	74
Onlinehilfe der Teach-Programmierung	75
Sicherheitsräume	75
Systemmasse	77
Neue Systemmasse-Tabelle anlegen	77
Systemmasse teachen und editieren	77
Sicherheitsräume der Systemmasse	78
Konfig-Text --> SPS-Text	79
Geschwindigkeitstabelle	80

Abbildungsverzeichnis

Abbildung 1: Konfiguration Teach-In	9
Abbildung 2: Einzeilige Darstellung einer Teach-Zeile	10
Abbildung 3: Zweizeilige Darstellung einer Teach-Zeile	10
Abbildung 4: Editor für die Option Schrittweise	12
Abbildung 5: Editor für die Option Override.....	13
Abbildung 6: Systemmasse der Palettierung	15
Abbildung 7: Palettierung substituieren	16
Abbildung 8: Auswahl Teach-In	19
Abbildung 9: Maske zur reinen Programmierung	20
Abbildung 10: Zweiteilige Arbeitsmaske	21
Abbildung 11: Programmbezeichnung editieren	69
Abbildung 12: Aktuelles Programm festlegen.....	70
Abbildung 13: Inhaltsverzeichnis der Programmauswahl	71
Abbildung 14: Gruppenzuordnung der Teachebenen	72
Abbildung 15: Handfunktionen.....	73
Abbildung 16: Teilenummern editieren	74
Abbildung 17: Onlinehilfe	75
Abbildung 18: Sicherheitsräume	76
Abbildung 19: Neue Systemmasse-Tabelle erstellen	77
Abbildung 20: Systemmasse teachen und editieren	78
Abbildung 21: Geschwindigkeitstabelle Teach-In	80

Vorwort

Diese Dokumentation wendet sich in erster Linie an die Programmierer einer Maschine. Das System ist grundsätzlich frei konfigurierbar, so dass auf die Nennung von Menüpunkten in dieser Dokumentation verzichtet wird. Wo die einzelnen Masken und Funktionen zu finden sind, ist von den Programmierern der Steuerung frei einstellbar und aus diesem Grund vom Lieferanten der Steuerung nicht dokumentierbar.

Auch der Aufbau und das Layout der einzelnen Masken kann verändert werden. Wie dies geschieht ist in der Dokumentation „Handbuch Editor-Konzept“ beschrieben. Alle Funktionen und Masken, die hier beschrieben werden, entsprechen in ihrem Layout der Version, die von der Andras Steuerungssysteme GmbH als Auslieferungszustand festgelegt worden ist.

Grundlagen der Teach-Programmierung

In diesem Bereich geht es um die Grundlagen der Teach-Programmierung und um die Zusammenhänge der einzelnen Bereiche.

Programme und Teachebenen

Das Betriebssystem unterscheidet innerhalb der Teach-Programmierung grundsätzlich zwischen Programmen und Teachebenen.

Ein Programm ist durch eine zweistellige Nummer gekennzeichnet und beinhaltet neben den Dateien der Teach-Programmierung auch noch andere Dateien, wie beispielsweise Sicherheitsräume, Konfigurationen und Bezeichnungen. Alle Dateien, die programmabhängig sind beinhalten in ihrem Dateinamen die zweistellige Programmnummer. Zum Aufruf solcher Datendateien aus einem Menü heraus kennt das Betriebssystem die Zeichenkombination @@, die automatisch durch die Nummer des aktiven Programms ersetzt wird. Es können mehrere Programm gleichzeitig auf der Ramdisk der CPU gehalten werden, aktiv laufen kann aber immer nur ein Programm zur Zeit.

Innerhalb eines Programms kann es bis zu 9 Teach-Prozesse geben, die parallel laufen können. Diese Prozesse nennen sich Teachebene und sind mit den Zahlen 1 bis 9 nummeriert. Jede Teachebene stellt einen eigenen Teach-Prozess dar und wird vom Betriebssystem zeilenweise abgearbeitet. Beim Start eines Programms wird vom Betriebssystem nur die Teachebene 1 auf der Zeile 1 gestartet. Sollen parallel weitere Teachebenen laufen, so müssen diese durch einen entsprechenden Befehl angestartet werden.

Zeitverhalten

Grundsätzlich wird jede Teachebene zeilenweise abgearbeitet und gibt die Rechenzeit erst dann wieder frei, wenn ein Sprung durchgeführt wird oder wenn die Teachebene warten muss (z.B. auf die Antwort eines Antriebs innerhalb des Bussystems). Die Teachebene erhält erneut Rechenzeit, wenn ein anderen Prozess diese Teachebene anspricht (z.B. weil die Antwort des Antriebs im Bustreiber angekommen ist). Spricht kein anderer Prozess die Teachebene an, erhält diese auf jeden Fall erneut nach 20 Millisekunden Rechenzeit. Die einzelnen Teachebenen sind innerhalb des Betriebssystems mit Prioritäten versehen, wobei die Teachebene mit der höchsten Priorität als erstes Rechenzeit erhält. Innerhalb der Prioritätenliste des Betriebssystems hat die Teachebene 1 von allen Teachebenen die höchste und die Teachebene 9 die niedrigste Priorität.

Konfiguration

Innerhalb der Teach-Programmierung kommt der Konfiguration eine zentrale Bedeutung zu. Alle Bewegungen, die in einer Teach-Zeile angesprochen werden, sind auf die Konfiguration bezogen. In der Konfiguration werden also alle Bewegungen (Objekte) festgelegt, die innerhalb der Teach-Programmierung angesprochen werden. Bewegungen, die nicht in der Konfiguration hinterlegt sind, können in der Programmierung auch nicht angesprochen werden.

Die Konfiguration wird in Dateien mit der Dateierdung CFG gespeichert. Dabei gibt es verschiedene Möglichkeiten, die sich daraus ergeben, wofür die Konfigurationen gültig sind. Näheres hierzu erläutert die nachfolgende Tabelle

Gültigkeit der Konfiguration	Art des Dateinamen	Beispiel
Konfigurationen, die nur programmabhängig sind, gelten für das angegebene Programm, dort aber für alle Gruppen und Teach-Ebenen	TEACH + zweistellig Programmnummer + .CFG	TEACH03.CFG für Programm 3
Konfigurationen, die programm- und gruppenabhängig sind, gelten für das angegebene Programm und dort nur für die Teach-Ebenen die der angegebenen Gruppe zugeordnet sind.	TEACH + zweistellige Programmnummer + einstellige Gruppennummer + .CFG	TEACH991.CFG für Programm 99, Gruppe 1

Innerhalb der Programmierung sucht das Betriebssystem grundsätzlich zuerst nach eine programm- und gruppenabhängigen Konfigurationsdatei. Nur wenn diese nicht gefunden wird, benutzt das Betriebssystem die rein programmabhängige Konfigurationsdatei.

Die Konfigurationsdateien sind rekordmässig aufgebaut, wobei jeder Rekord ein Objekt beschreibt. Die Rekordnummer innerhalb der Datei ist zugleich die Bewegungsnummer. Jedes Objekt wird entweder durch eine Achse oder durch einen Ablauf repräsentiert. Wenn das Objekt eine Achse ist, so muss im Feld Achse die Achsnummer eingetragen werden. Wenn das Objekt durch einen Ablauf repräsentiert wird, so muss im Feld Ablauf die Ablaufnummer eingetragen werden. Insgesamt kann jede Konfigurationsdatei maximal 99 Bewegungen beinhalten.

Dem Objekt kann ein Bewegungstext zugewiesen werden. Dieser Bewegungstext wird in der zentralen Textdatei unter dem Schlüssel CFGTEXT...+eingeebene Nummer des Bewegungstextes (also beispielsweise CFGTEXT...100 für den Bewegungstext 100) abgespeichert. Innerhalb der Maske zur Editierung der Konfiguration wird angegeben, welcher Bewegungstext verwendet werden soll (Feld Textnr. Bewegung) und der entsprechende Text kann editiert werden.

Aus der Konfiguration werden auch die Softkeys abgeleitet, die für das entsprechende Objekt angezeigt werden, wenn innerhalb des Teach-In das entsprechende Objekt angewählt wird. Bei Achsen sind diese Softkeys systembedingt vorgegeben und können nicht verändert werden. Bei Abläufen kann festgelegt werden, welcher Text einem Softkey zugewiesen wird und welcher Auftrag an den Ablauf gesendet wird, wenn der Softkey gedrückt wird. Die Betextung der Softkeys für Abläufe erfolgt – wie der Bewegungstext selber auch – über eine Textnummer, die in der entsprechenden Maske eingegeben werden kann. Auch hier kann der Text editiert werden. Soll ein Softkey keinen Text und keine Funktion erhalten, so wird die Bewegungsnummer 0 eingetragen. Zusätzlich muss für jeden der vier Softkeys zusätzlich angegeben werden, welchen Auftrag der Ablauf erhalten soll, wenn der Softkey gedrückt wird (Feld Auftrag Softkey). Werden bei Objekten, die als Achse definiert sind, Texte und Aufträge vorgegeben, so werden diese ignoriert.

Grundsätzlich kann jede Achse oder jeder Ablauf mehrmals innerhalb der Konfiguration vorkommen und dabei auch mit unterschiedlichen Texten versehen werden.

KONFIG TEACHIN		
AKTUELLE DATEI :	TEACH021.CFG	
BEWEGUNGSNUMMER . .	1	
ACHSE	0	
ABLAUF	110	
TEXTNR BEWEGUNG . .	100	HUB LIFT 2
TEXTNR. SOFTKEY 1	101	POS. BAND 1
AUFTRAG SOFTKEY 1	1	
TEXTNR. SOFTKEY 2	102	POS. BAND 2
AUFTRAG SOFTKEY 2	2	
TEXTNR. SOFTKEY 3	103	POS. BAND 3
AUFTRAG SOFTKEY 3	3	
TEXTNR. SOFTKEY 4	104	POS. BAND 4
AUFTRAG SOFTKEY 4	4	
<< SATZ-1 SATZ+1		

Abbildung 1: Konfiguration Teach-In

Innerhalb der Maske zur Editierung der Konfiguration kann mit den Softkeys Satz-1 und Satz+1 zwischen einzelnen Datensätzen hin- und hergeschaltet werden. In der zweiten Softkeyleiste sind die Funktionen zum Einfügen (Satz eingef.) und Entfernen (Satz entf.) von Datensätzen untergebracht. Mit dem Softkey Drucken kann die Konfiguration in eine Textdatei auf Diskette ausgegeben werden.

Neben der Konfiguration für die Teach-Programmierung existiert noch eine Konfiguration für die Handfunktionen. Diese Konfiguration ist genauso aufgebaut, wie die Konfiguration für die Teach-Programmierung und wird auch mit der selben Maske editiert. Der Unterschied liegt lediglich darin, dass die Konfiguration der Handfunktionen ausschließlich programmabhängig ist und den Dateinamen HANDB + zweistellige Programmnummer + .CFG trägt (Beispiel: HANDB02.CFG für das Programm 2). Eine gruppenabhängige Konfiguration für die Handfunktionen gibt es nicht.

Hinweis: *Aufträge innerhalb der Ablaufprogrammierung können normalerweise 5stellig sein. Innerhalb der Konfiguration können aber nur Aufträge von 0 bis 255 eingetragen werden!*

Genereller Aufbau einer Zeile

In diesem Kapitel wird der generelle Aufbau einer Zeile innerhalb der Teach-Programmierung erläutert. Dabei wird nicht nur auf die Besonderheiten der einzelnen Felder, sondern auch auf die Besonderheiten während der Editierung eingegangen.

Einzeilige und zweizeilige Darstellung

Je nach Maske wird eine Zeile der Programmierung im Bild einzeilig oder zweizeilig angezeigt. Die Bedeutung der einzelnen Felder ist davon nicht betroffen. Bei der zweizeiligen Darstellung wird in der unteren Zeile grundsätzlich die textmässige Beschreibung des Befehls ausgegeben. Bei der einzeiligen Darstellung werden zunächst einmal nur die Werte in den Feldern angezeigt. Hat ein Feld keine Bedeutung für den entsprechenden Befehl, dann wird in diesem Feld ein Hilfetext zu dem Befehl angezeigt.

NR.	BEF.	BEWEGUNG	POSITION/MASS	GESCHW.	WARTEN
1	0	ENTNAHMEHUB	GRUNDST.	0	JA

Abbildung 2: Einzeilige Darstellung einer Teach-Zeile

NR.	BEF.	BEWEGUNG	POSITION / MASS	GESCHW.	WARTE
1	0	ENTNAHMEHUB STARTE ABLAUF MIT JOB	GRUNDST.	0	JA

Abbildung 3: Zweizeilige Darstellung einer Teach-Zeile

Bedeutung und Editierung der einzelnen Felder

Die Bedeutung der einzelnen Felder und die Besonderheiten des Feldes bei der Editierung werden in der nachfolgenden Tabelle dargestellt. Viele Felder sind in ihrer Bedeutung abhängig von dem verwendeten Befehl. Diese Abhängigkeiten werden in der Befehlsbeschreibung näher erläutert. Die Tabelle dient daher nur als grobe Orientierung und zur Erläuterung des Editierverfahrens.

Feld	Bedeutung	Editierung
Nr.	Zeilennummer des Programms, zugleich Satznummer des Rekords in der Datei	Ist diese Zeile Spungziel eines Befehls, so steht vor der Nummer ein Sternchen
Bef.	Befehl der ausgeführt werden soll	Keine Besonderheiten
Bewegung	Bewegungsnummer in der Konfiguration auf die sich der Befehl bezieht, wenn der Befehl eine Bewegungsnummer benötigt.	Wird eine Bewegungsnummer eingegeben, die in der Konfiguration definiert ist, so wird anstelle der Nummer der entsprechende Bewegungstext angezeigt. Eingegeben wird immer die Nummer.
Position / Mass	Bei Befehlen für Achsen steht hier die Zielposition, bei Befehlen für Abläufe die Auftragsnummer. Bei indirekten Befehlen wird hier die Palettierung verschlüsselt.	Bei direkten Befehlen für Abläufe wird anstelle der Auftragsnummer der entsprechende Text aus der Konfiguration angezeigt. Eingegeben wird immer die Nummer.
Geschw.	Bei Befehlen für Achsen steht hier die Angabe mit wie viel Promille der maximalen Geschwindigkeit gefahren werden soll. In der Softwareständen ab 10.03.2006 kann der Wert größer als 1000 sein. Werte größer 1000 werden dann als Zugriff aus die Geschwindigkeitstabelle interpretiert. Bei Befehlen, die einen Sprung ausführen, steht hier die Zeile, auf die gesprungen werden soll	Bei Befehlen, die einen Sprung ausführen, wird hinter der Zahl ein Sternchen ausgegeben, damit sofort ersichtlich ist, dass es sich um eine Sprungadresse handelt.
Warten	Angabe, ob nach Ausführung des Befehls gewartet werden soll, bis die Achse das Ziel erreicht hat, bzw. bis der Ablauf den Auftrag vollständig abgearbeitet hat.	Eingegeben wird eine Zahl. Dabei steht 0 für Ja, 1 für Nein, 2 für Bahnkoppelung, 10 für Ja mit Schrittweise, 11 für Nein mit Schrittweise und 12 für Bahnkoppelung mit Schrittweise. Wird eine der Schrittweise-Optionen verwendet, so steht hinter dem Warten-Text ein Sternchen, also beispielsweise „JA*“ für Ja mit Schrittweise.

Verschlüsselung von Palettiertabellen

In einigen Befehlen wird im Feld Position / Mass nicht die tatsächliche Position angegeben, sondern es wird in diesem Feld verschlüsselt, aus welcher Palettiertabelle die Position geholt werden soll. Die Verschlüsselung der Palettiertabelle erfolgt dabei nach folgendem Schema:

10000er+1000er-Stelle = Nummer der Tabelle
 100er-Stelle = Art der Tabelle (1=Teile, 2=Etagen, 3=Addiert)
 10er+1er-Stelle = Spalte in der Tabelle

Wird die 10000er und die 1000er-Stelle nicht angegeben, so wird die als Default gesetzte Tabelle verwendet. Diese Defaulttabelle kann mittels Befehl gesetzt werden. Die verwendete Zeile innerhalb der Palettiertabelle ergibt sich aus dem Teilezähler.

Beispiele:

98123 – Teiletabelle mit der Nummer 98, Spalte 23
 01201 – Etagentabelle mit der Nummer 1, Spalte 1
 201 – Etagentabelle die als Default gesetzt ist, Spalte 1

Bedeutung der Option Schrittweise

Im Feld Warten kann angegeben werden, dass die Option Schrittweise aktiv ist (siehe Beschreibung der Editierung zum Feld Warten). Schrittweise bedeutet, dass die entsprechende Zeile vollständig ausgeführt wird und anschließend der Startmarker gelöscht wird. Durch diese Option bleibt das Programm nach Ausführung der Zeile stehen und muss durch einen neuen Startbefehl gestartet werden. Die Schrittweise-Option kann global ein- und ausgeschaltet werden. Ist die Option ausgeschaltet, wirken die Schrittweise-Einträge genauso wie die entsprechenden „normalen“ Einträge, d.h. beispielsweise Warten Ja mit Schrittweise wird zu Warten Ja. Die Schrittweise-Option ist besonders dazu geeignet, eine Teach-Programmierung sequenzweise zu testen.



Abbildung 4: Editor für die Option Schrittweise

Wenn der Editor für die Option Schrittweise eingeblendet wird, zeigt er den aktuellen Zustand (aktiv oder inaktiv). Der Zustand der Option kann durch Softkeys verändert werden. Mit der Taste Zurück wird der Editor wieder ausgeblendet.

Bedeutung der Option Override

Die Teach-Programmierung ist mit einem generellen Override versehen. Mit dieser Option kann die Geschwindigkeit der Anlage herabgesetzt werden. Beim Override handelt es sich um eine Prozentzahl, die angibt, um wieviel die eingetragenen Geschwindigkeiten zusätzlich reduziert werden. Der Override kann Werte zwischen 1 und 100 Prozent haben und wird durch einen entsprechenden Editor verändert, der in die Masken der Automatik und der Teach-Programmierung durch einen Softkey eingeblendet werden kann.



Abbildung 5: Editor für die Option Override

Wenn der Editor für die Option Override eingeblendet wird, zeigt er den aktuellen Prozentsatz. Der Zustand der Option kann durch Softkeys verändert werden. Mit der Taste Zurück wird der Editor wieder ausgeblendet.

Der Override begrenzt die eingetragene Geschwindigkeit einer Teach-Zeile. Hierzu ein paar Beispiele:

Promille in der Teach-Zeile	Prozentsatz des Override	Geschwindigkeit, die an die Achse gesendet wird
1000 Promille	100%	1000 Promille
500 Promille	100%	500 Promille
1000 Promille	50%	500 Promille
500 Promille	50%	500 Promille

Der Override wirkt erst, wenn an die Achse erneut ein Fahrbefehl gesendet wird. Wird der Override verändert, während sich die Achse bereits bewegt, so gilt der neue Override für diese Achse erst, wenn die Achse einen neuen Positionierbefehl erhält!

Sequenzprogrammierung

Die Teach-Programmierung bietet die Möglichkeit, Teile des Programms als sogenannte Sequenz auf ein dezentrales Antriebselement zu laden und die Programmausführung auf dem Antriebselement durchführen zu lassen. Dieses Verfahren führt zu erheblich beschleunigten Ausführungen und Reaktionszeiten, da das Programm auf dem dezentralen Element ausgeführt wird. Durchführbar ist die Sequenzprogrammierung derzeit nur mit Antrieben der Andrive Antriebstechnik GmbH. Die entsprechenden Antriebe müssen werkseitig mit einer speziellen Firmware ausgestattet sein. **Die Sequenz darf nur Befehle für das Antriebselement enthalten, auf das die Sequenz geladen wird**, da die dezentralen Antriebselemente untereinander nicht kommunizieren können.

Eine Sequenz muss zunächst vollständig auf das Antriebselement geladen werden. Hierzu dienen die Befehle 40 bis 59. Wenn die Sequenz vollständig geladen ist, kann die Sequenz mit den Befehlen 62 oder 63 gestartet werden. Innerhalb eine Sequenz sind keine Sprünge möglich.

Palettierung

Jedes Programm kann bis zu 99 Palettiertabellen für Teile und bis zu 99 Teilenummern für Etagen haben. Die Datenstruktur der Palettierung und damit auch die entsprechenden Masken sind kundenspezifisch definierbar und können in jeder Anlage anders sein. Aus diesem Grund werden an dieser Stelle keine Masken angezeigt. Die Datenbeschreibung für die Etagentabellen steht in der Datei PT.DBH, die Datenbeschreibung für die Teiletabellen steht in der Datei PE.DBH.

Die Bildung des Dateinamens ergibt sich aus der folgenden Tabelle:

Art der Tabelle	Regel für den Dateinamen	Beispiel
Etagen	PE + Nummer der Tabelle + PG + Nummer des Programms + .PE	PE03PG02.PE Tabelle 3 für Programm 2
Teile	PT + Nummer der Tabelle + PG + Nummer des Programms + .PT	PT77PG02.PT Tabelle 77 für Programm 2

Das Betriebssystem verfügt über eine Funktion *Neue Palettierung anlegen*, mit dieser Funktion wird eine neue Palettiertabelle angelegt, die die eingegebene Nummer erhält. Dabei werden immer sowohl die Etagen- als auch die Teiletabelle mit der eingegebenen Nummer angelegt.

Jede Palettierung hat ihren eigenen Teilezähler. Dieser Teilezähler wird in der Automatikmaske und in den Arbeitsmasken der Teach-Programmierung angezeigt und hat folgendes Aussehen: Nummer der Palette / Etagenzähler / Teilezähler, also beispielsweise 99/7/3 für Tabelle 99, Etage 7, Teil 3.

Aus der Programmierung heraus kann nur der Teilezähler per Befehl verändert werden. Steht der Teilezähler am Ende seiner Palettiertabelle und wird dann um eins erhöht, so springt der Teilezähler auf den Wert 1 und der Etagenzähler wird um 1 erhöht. Das Ende einer Tabelle wird erkannt, wenn entweder die Datei keine weiteren Datensätze enthält, oder wenn die nächste Zeile in allen Spalten mit Nullen belegt ist. Wenn auch der Etagenzähler an seinem Ende angekommen ist, springen Teile- und Etagenzähler auf 1.

Palettierung substituieren

Es besteht die Möglichkeit, die Adressierung einer Palettiertabelle zu substituieren. Dabei wird anstelle der adressierten Palettiertabelle die Substitutionstabelle verwendet, allerdings wird die Systemmasse-Tabelle der adressierten Palettiertabelle benutzt. Diese Möglichkeit ist vor allem dann sinnvoll, wenn gleiche Palettierungen an mehreren Stellen vorkommen und sich ausschließlich der Koordinatenursprung der Palettierung verändert.

In der entsprechenden Eingabemaske wird angegeben, welche Quelltablelle durch welche Zieltabelle ersetzt werden soll.

TABELLEN SUBSTITUIEREN

QUELLE	ZIEL
1	7
0	0

SATZ EINF.

SATZ ENTF.

SEITE-

SEITE+

Abbildung 7: Palettierung substituieren

Im obigen Beispiel soll die Palettiertabelle 1 durch die Tabelle 7 ersetzt werden. Wird nun im Teach-Programm die Palettiertabelle 1 angesprochen, so verwendet das Betriebssystem statt dessen die Tabelle 7, als Koordinatenursprung werden aber die zur Tabelle 1 gehörenden Werte benutzt.

Diese Funktion ist in Systemen mit einem Versionsstand ab 17.02.2004 implementiert!

Darstellung des Felds Position innerhalb der Teach-Programmierung bei Palettierbefehlen

Wird innerhalb der Teach-Programmierung ein Befehl benutzt, der sich auf eine Palettierung oder auf ein Systemmass bezieht, so wird der entsprechende Wert im Feld „Position“ verschlüsselt und wie folgt dargestellt:

Werte	Darstellungsform
0 bis 97	Text aus den Systemmassen
98	Fester Text: „Minimales Mass“
99	Fester Text: „Maximales Mass“
100 bis 99399	Tabellennummer / Art der Tabelle / Spalte Beispiel: 99/1/45

Diese Darstellung ist in Systemen mit einem Versionsstand ab 04.03.2004 implementiert!

Rechnen innerhalb von Palettiertabellen

Das Bild zur Bearbeitung von Palettiertabellen stellt verschiedene Rechen-Funktionen zur Verfügung, die innerhalb des Menüpunkts „Rechnen“ angezeigt sind. Alle Rechen-Funktionen setzen voraus, dass der Cursor auf dem ersten Feld steht, welches von der Funktion betroffen ist. Alle Rechen-Funktionen arbeiten grundsätzlich vertikal, d.h. sie beziehen sich auf immer das gleiche Feld über mehrer Zeilen der Tabelle hinweg.

Funktion	Wirkungsweise
Reihe	Zunächst muss die Anzahl der Zeilen angegeben werden, die von der Funktion betroffen sein sollen. Anschließend muss ein Wert eingegeben werden, um den das nächste Feld jeweils erhöht werden soll. Die Firmware nimmt dann den Wert aus der vorherigen Zeile, erhöht ihn um den angegebenen Wert und schreibt in die aktuelle Zeile. Wird als Wert 2 eingegeben und der erste Wert ist 0, dann entsteht eine Reihe mit den Werten 2, 4, 6, 8,
Addieren	Zunächst muss die Anzahl der Zeilen angegeben werden, die von der Funktion betroffen sein sollen. Anschließend muss ein Wert eingegeben werden, der auf den aktuellen Wert des Feldes addiert werden soll. Negative Werte führen zu einer Subtraktion. Die Firmware nimmt dann für alle markierten Felder den aktuellen Wert, addiert den eingegebenen Wert und schreibt das Ergebnis zurück in das Feld.
Duplizieren	Zunächst muss die Anzahl der Zeilen angegeben werden, die von der Funktion betroffen sein sollen. Anschließend muss angegeben werden, wie oft der Block kopiert werden soll. Die Firmware kopiert den angegebenen Block dann x-Mal untereinander. Voraussetzung ist es, dass für das Kopieren ausreichend Zeilen angelegt sind. Diese Funktion fügt der Tabelle keine neuen Zeilen an!

Importieren von Palettiertabellen aus einer CSV-Datei

Im Untermenü „Rechnen“ einer Palettiertabelle steht die Funktion „Import“ zur Verfügung. Mit dieser Funktion ist es möglich, die Palettiertabelle aus einer CSV-Datei zu importieren. Dabei muss die CSV-Datei pro Zeile genau eine Zeile der Tabelle beinhalten und mit CR LF abgeschlossen sein. Die einzelnen Felder einer Zeile müssen in der CSV-Datei mit einem Semikolon getrennt sein. Mehr zum Thema CSV-Datei findet sich in der Dokumentation *Handbuch Steuerung – Allgemeine Funktionen, Kapitel Generelles zu CSV-Dateien*.

Beim Aufruf der Funktion muss zunächst die gewünschte CSV-Datei ausgewählt werden. Diese CSV-Datei muss sich auf einer Diskette befinden. Die Firmware kontrolliert dann, ob die Anzahl der Felder innerhalb der CSV-Datei mit der Anzahl der Felder innerhalb der Datenbeschreibung der Palettiertabelle überein stimmt. Ist dies nicht der Fall, wird der Import abgelehnt.

Beim Importieren wird zunächst die aktuell angezeigte Tabelle gelöscht. Anschließend werden die Daten aus der CSV-Datei gelesen und in die Palettiertabelle eingetragen.

Diese Funktion steht in Systemen mit einem Versionsstand ab 10.05.2004 zur Verfügung.

Editierung von Teach-Programmen

Für die Editierung bzw. Erstellung von Teach-Programmen stehen grundsätzlich verschiedene Masken und verschiedene Verfahren zur Verfügung.



Allen Masken und Verfahren ist aber gleich, dass zunächst die gewünschte Datei mit dem Teach-Programm ausgewählt werden muss. Dabei wird jede Teachebene in einer eigenen Datei gespeichert. Der Dateiname ergibt sich aus der Nummer der Teachebene und der Nummer des aktuellen Programms und wird nach der Regel TEACH + Nummer der Teachebene + Programmnummer gebildet. TEACH102 ist also die erste Teachebene des Programms 02.

Die Auswahlliste beinhaltet grundsätzlich alle Teachebenen, die für ein Programm bereits existieren. In der Liste kann keine Datei für eine neue Teachebene angelegt werden. Hierzu dient die Funktion **Neue Teachebene anlegen**, die sich im Konfigurationsmenü des Teach-In befindet.

Zur Auswahl der Teachebene, die bearbeitet werden soll, wird in der entsprechenden Dateiliste die gewünschte Datei markiert. Zur Navigation in der Dateiliste können die Cursorastern verwendet werden. Mit der *Enter*-Taste wird die Auswahl bestätigt, mit der *Rück*-Taste wird die Auswahl abgebrochen.

Abbildung 8: Auswahl Teach-In

Maske zur reinen Programmierung

Es existiert eine Maske zur reinen Programmierung einer Teachebene. Diese Maske füllt den gesamten Bildschirm aus und ermöglicht es, ein Programm vollständig zu editieren. Dabei wird das gewünschte Programm über die Tastatur eingegeben.

TEACHIN PROGRAMMIEREN						
AKTUELLE DATEI: TEACH102.TI						
NR.	BEF.	BEWEGUNG	POSITION / MASS	GESCHW.	WARTE	
1	0	ENTNAHMEHUB STARTE ABLAUF MIT JOB	GRUNDST.	0	JA	
2	73	PROGRAMM STARTEN	2.0+	10	JA	
3	73	PROGRAMM STARTEN	3.0+	10	JA	
4	73	PROGRAMM STARTEN	4.0+	10	JA	
5	12	EINLEGEHUB SPRUNG WENN ISTWERT <> MASS NACH GESCHW.	0.0+	0	*	JA
6	0	0	0.0+	0	JA	
7	0	0	0.0+	0	JA	
8	0	0	0.0+	0	JA	

<< SEITE- SEITE+

Abbildung 9: Maske zur reinen Programmierung

In dieser Maske können Zeilen eingefügt und gelöscht werden. Die entsprechenden Softkeys befinden sich in der zweiten Softkeyreihe. Hier finden sich auch die Druckfunktionen, die mehrere Funktionen zur Verfügung stellen:

- **In Datei** druckt das aktuell angezeigte Programm in eine Textdatei auf Diskette und verwendet dabei eine zweizeilige Darstellungsform
- **Einzeilig** druckt das aktuell angezeigte Programm in eine Textdatei auf Diskette und verwendet dabei eine einzeilige Darstellungsform
- **Alle** druckt alle Teachebenen eines Programms in Textdateien auf Diskette. Dabei wird für jede Teachebene eine neue Textdatei erzeugt.

Wird im Feld Nr. eine Zahl eingegeben, so springt der Cursor auf die entsprechende Zeile und öffnet den Editiermodus. Ist die eingegebene Nummer höher als die Anzahl der Zeilen der Teachebene, so springt der Cursor auf die letzte Zeile der Teachebene.

Zweiteilige Arbeitsmaske

In der Arbeitsmaske kann nicht nur die ausgewählte Teachebene editiert werden, sondern in dieser Maske können auch die einzelnen, konfigurierten Bewegungen ausgeführt werden und die Programmierung kann getestet werden. Die Arbeitsmaske ist zweiteilig, wobei im oberen Bildteil die konfigurierten Bewegungen aufgeführt werden. Im unteren Bildteil befindet sich die Programmierung der Teachebene. Zwischen den beiden Bildteilen kann mit der *Tab*-Taste hin und her gesprungen werden.

ISTWERT: 0.000+		Sollwert: 0.000+		1/02	1/ 1/ 1
X-ACHSE GRUPPE 2		Y-ACHSE GRUPPE 2		Z-ACHSE GRUPPE 2	
ENTNAHMEHUB		EINLEGEHUB		C-ACHSE	
WERKZEUG		VAK.ROHT.EIN		VAK.ROHT.AUS	
SGM <-> HG		VAK.FERTIGT.EIN		VAK.FERTIGT.AUS	
BLECHAUFNAHME		TÜR ANWAHL		VERWALTUNG	
NR.	BEF.	BEWEGUNG	POSITION/MASS	GESCHW.	WARTEN
1	0	ENTNAHMEHUB	GRUNDST.	0	JA
2	73	PROGRAMM STARTEN	2.0+	10	JA
3	73	PROGRAMM STARTEN	3.0+	10	JA
4	73	PROGRAMM STARTEN	4.0+	10	JA
5	12	EINLEGEHUB	0.0+	0	* JA
6	0		0.0+	0	JA
7	0		0.0+	0	JA
8	0		0.0+	0	JA
SYSTEMDRUCK ZU NIEDRIG					INAKTIV
TIPPEN -		TIPPEN +		1/10 MINUS	
TIPPEN -		TIPPEN +		1/10 PLUS	

Abbildung 10: Zweiteilige Arbeitsmaske

Im oberen Bildteil sind die einzelnen Bewegungen aufgeführt, wie sie in der Konfiguration definiert worden sind. In diesem Bild können die Bewegungen mittels Softkeys ausgeführt werden. Dabei beziehen sich die Softkeys immer auf die Bewegung, die aktuell durch den Cursor markiert sind. Handelt es sich bei der Bewegung um Achsen, so ist die Belegung der Softkeys immer gleich:

Softkey	Bedeutung
Tippen -	Bewegt die Achse in negativer Fahrtrichtung. Die Bewegung erfolgt, solange der Softkey gedrückt ist, dabei wird die Bewegung langsam schneller. Wird der Softkey losgelassen, wird die Bewegung gestoppt.
Tippen +	Bewegt die Achse in positiver Fahrtrichtung. Die Bewegung erfolgt, solange der Softkey gedrückt ist, dabei wird die Bewegung langsam schneller. Wird der Softkey losgelassen, wird die Bewegung gestoppt.
1/10 Minus	Fährt die Achse – ausgehend von der aktuellen Istposition – bei jedem Drücken um 1/10 Millimeter in negativer Richtung
1/10 Plus	Fährt die Achse – ausgehend von der aktuellen Istposition – bei jedem Drücken um 1/10 Millimeter in positiver Richtung

Handelt es sich bei der markierten Bewegung um einen Ablauf, so werden auf den Softkeys die Texte dargestellt, die in der Konfiguration für den entsprechenden Softkey definiert worden sind. Wird ein Softkey gedrückt, so wird dem Ablauf der Auftrag gesendet, der in der Konfiguration für diesen Softkey vorgesehen ist.

Im unteren Teil der Arbeitsmaske wird die Teachebene angezeigt. Die Editierung entspricht dabei im wesentlichen den Verfahren, wie es im Kapitel „Maske zur reinen Programmierung“ beschrieben worden ist – allerdings ist die Darstellung einzeilig und es gibt keine Druckfunktionen.

Im unteren Teil befinden sich auch die Testfunktionen. Hier kann eine Teachebene betrieben werden, ähnlich als ob sie im Automatikbild laufen würde. Um das Programm der Teachebene einfach vollständig laufen zu lassen, wird die Start-Taste benutzt. Mit der Stop-Taste wird dann die Programmausführung unterbrochen. Während die Teachebene im Test läuft, wird der Cursor im Bild nachgeführt – d.h. er steht immer auf der Zeile, auf der die Ausführung aktuell arbeitet. Eine Editierung ist in dieser Zeit nicht möglich. Die Softkeys geben verschiedene Möglichkeiten, die nachfolgend erläutert werden:

Softkey	Bedeutung
Satz einf.	Fügt an der Position des Cursors eine Zeile ein. Während dieser Softkey sichtbar ist, kann die Teachebene nicht getestet werden!
Satz entf.	Löscht die Zeile, an der der Cursor aktuell steht.
Aktiv / Inaktiv	Schaltet das aktive Teach-In ein und aus. Mehr dazu im Kapitel Teach-In-Funktion. Der aktuelle Zustand (aktiv oder inaktiv) wird über dem vierten Softkey in der Bemerkungszeile angezeigt.
Override	Ruft das Bild zum Verändern des Override auf.
Einzel	Führt einen Einzelschritt-Test durch. Die Zeile, auf der der Cursor aktuell steht, wird ausgeführt.
Breakp.	Setzt einen Breakpoint auf eine Zeile, die eingegeben werden muss. Wenn die Ausführung mit Start begonnen wird und dann an die mit einem Breakpoint markiert Zeile gelangt, wird die Ausführung unterbrochen.
Schritt w.	Ruft das Bild zum Verändern der Schrittweise-Option auf
Softkeys	Blendet die userdefinierten Softkeys ein, die aus der Ablaufprogrammierung gesetzt werden können. Solange die userdefinierten Softkey eingeblendet sind, stehen keine anderen Softkeys zur Verfügung. Um in die Softkeys der Teach-Programmierung zurück zu kommen, muss innerhalb der userdefinierten Softkeys die Rück-Taste verwendet werden.

Hinweis zum Testen unter Stop passiv

Wird der Test des TeachIn mit Start begonnen und läuft dann über den Befehl *Stop passiv*, so läuft das TeachIn auch bei Betätigung der Stop-Taste weiter. Das TeachIn hält erst an, wenn der Befehl *Stop aktiv* ausgeführt worden ist. Der Cursor steht dann auf der Zeile hinter dem Befehl *Stop aktiv*. Solange diese Zeile nicht erreicht ist, wird vom Bedienterminal keine Taste angenommen.

Wird auf einer Zeile mit dem Befehl *Stop passiv* der Softkey *Einzelstschritt* gedrückt, so beginnt das TeachIn zu laufen. Es läuft dann solange, bis der Befehl *Stop aktiv* ausgeführt worden ist und steht anschließend auf der Zeile dahinter.

Onlinehilfe in der zweiteiligen Arbeitsmaske

Befindet sich der Cursor der zweiteiligen Arbeitsmaske des Teach-In innerhalb des Programms, so ist die VOR-Taste mit Hilfsfunktionen ausgestattet, die einer Onlinehilfe entsprechen. Dabei werden bei Betätigung der VOR-Taste, abhängig von dem Feld, auf dem der Cursor steht, Informationen eingeblendet. Diese Informationen werden grundsätzlich in einem zusätzlichen Bild untergebracht, welches mit der RÜCK-Taste verlassen werden muss, um wieder das Editierbild zu erreichen.

Die VOR-Taste ist dabei wie folgt belegt:

- (a) steht der Cursor auf der Zeilennummer, so wird der Verbundtest des Teach-In geöffnet
- (b) steht der Cursor auf dem Befehl, so wird eine Liste aller möglichen Teach-In-Befehle geöffnet
- (c) steht der Cursor auf der Bewegung, so wird eine Liste aller konfigurierten Bewegungen, die zu dieser Teach-Ebene gehören, eingeblendet
- (d) steht der Cursor auf dem Feld Position und handelt es sich bei der Bewegung um einen Ablauf, so wird eine Liste aller zu diesem Ablauf gehörenden, betexteten Jobs angezeigt
- (e) steht der Cursor auf dem Feld Position und handelt es sich bei der Bewegung um eine Achse, so wird die resultierende Position angezeigt. Bei dieser Berechnung werden die aktuellen Teilezähler und die entsprechenden Systemmasse berücksichtigt. Die angezeigte Position entspricht als dem Wert, auf den die Achse fahren würde, wenn diese Zeile unter den aktuellen Bedingungen angestartet wird
- (f) steht der Cursor auf dem Feld Geschwindigkeit und handelt es sich bei dem Befehl um einen Sprung, so wird die Zeile angezeigt, auf die der Sprung erfolgt. Diese Funktion ist nur wirksam, wenn der Sprung innerhalb der Teach-Ebene erfolgt.

Alle eingeblendeten Hilfsinformationen sind read-only, können also an dieser Stelle nicht geändert werden!

Teach-In Funktion

Die zweiteilige Arbeitsmaske ist mit einer Teach-In Funktion ausgestattet. Dabei kann eine Bewegung zunächst manuell ausgeführt werden und dann als Resultat in das Programm übernommen werden. Diese Funktion steht nur zur Verfügung, wenn sie explizit aktiviert worden ist. Aktivieren und deaktivieren kann man die Funktion mittels des Softkeys Aktiv / Inaktiv innerhalb des unteren Bildteils. Der aktuelle Zustand (aktiv oder inaktiv) wird in der Meldezeile links über dem vierten Softkey angezeigt.

Um die Teach-In-Funktion zu nutzen, wird die gewünschte Bewegung ausgeführt – beispielsweise wird eine Achse auf eine bestimmte Position gefahren. Wird dann die Enter-Taste verwendet, so wird diese Bewegung in das Programm eingetragen. Verwendet wird dabei die Programmzeile, in der sich der Cursor aktuell befindet – sollte sich in dieser Programmzeile bereits ein Befehl befinden, so wird dieser teilweise überschrieben! Übernommen werden die Bewegungsnummer und die Zielposition bzw. der Auftrag. Der eigentliche Befehl, die Geschwindigkeit und das Feld Warten behalten den Wert, der zuvor dort gestanden hat.

Beim Teach-In einer Achse wird mit der Übernahme in das Feld Bewegung die Bewegungsnummer der Achse eingetragen, in das Feld Position wird die Istposition der Achse eingetragen, wie sie zum Zeitpunkt des Drückens der Enter-Taste gewesen ist.

Beim Teach-In eines Ablaufs wird mit der Übernahme in das Feld Bewegung die Bewegungsnummer des Ablaufs eingetragen. In das Feld Position wird der Auftrag eingetragen, der zu dem Softkey gehört, der für diesen Ablauf zuletzt gedrückt worden ist.

Nachdem die Bewegung in die Teacheben eingetragen worden ist, wird der Cursor automatisch auf die nächste Zeile gesetzt. Befindet sich der Cursor bereits in der letzten Zeile, so bleibt er dort – die Teach-In Funktion legt keine neue Zeile an!

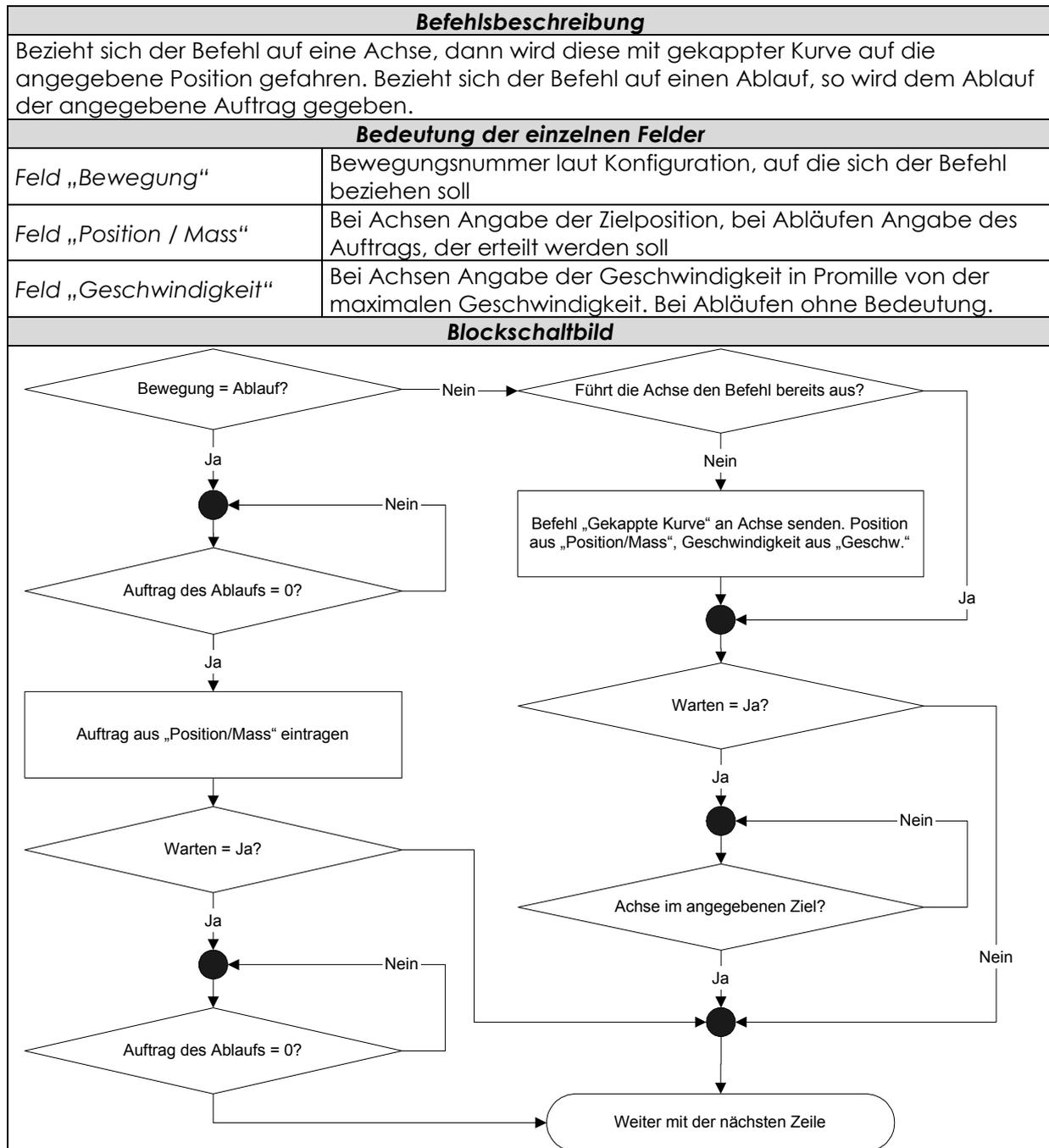
Befehlsbeschreibung der Teach-Programmierung

In diesem Kapitel werden die einzelnen Befehle der Teach-Programmierung näher erläutert. Hierzu wird eine kurze Erklärung gegeben und ein Blockschaltbild angezeigt. Dokumentiert sind nur die Befehle, die in den OALSYS-Betriebssystemen implementiert sind. Nicht dokumentierte Befehlsnummern werden auch nicht ausgeführt, d.h. entsprechende Zeilen werden vom Betriebssystem vollständig ignoriert.

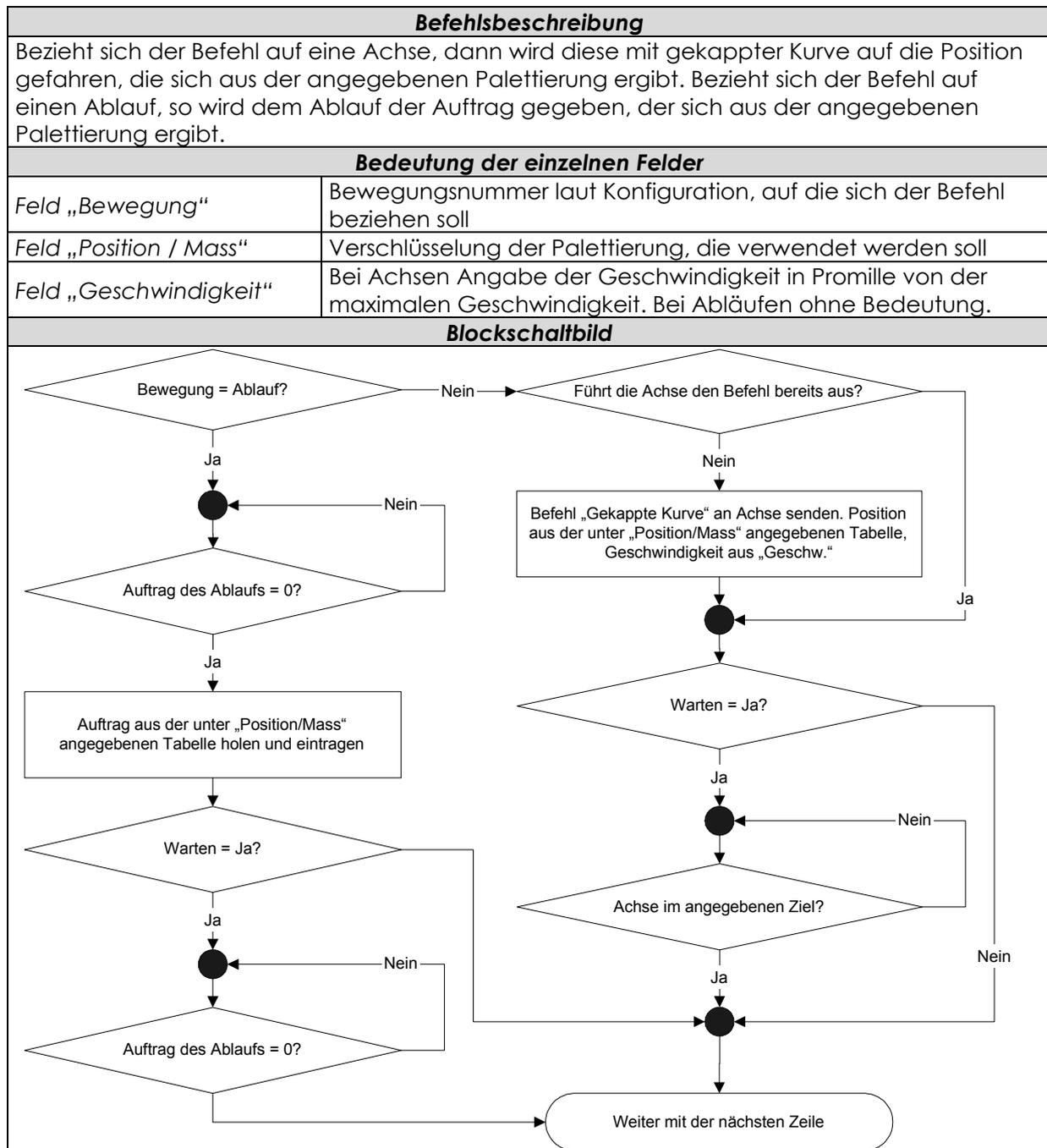
Die nachfolgende Tabelle gibt eine Übersicht der aktuell nicht implementierten Befehle:

Befehl	Beschreibung
4	Starten einer Synchronfahrt ab Punkt Mass
5	Starten einer Synchronfahrt ab Punkt Mass aus Palettierung
14	Fahre auf ersten Punkt von Bahnsegment Mass
15	Fahre auf ersten Punkt von Bahnsegment Mass aus Palettierung
16	Starte Bahnsegment Mass
17	Starte Bahnsegment Mass aus Palettierung
18	Beenden der Synchronfahrt
19	Beenden der Synchronfahrt
32	Sprung wenn Bahnbezugspunkt = Mass
33	Sprung wenn Bahnbezugspunkt = Mass aus Palettierung
34	Setze Bahnbezugspunkt = Mass
35	Setze Bahnbezugspunkt = Mass aus Palettierung
80	Bahn senden

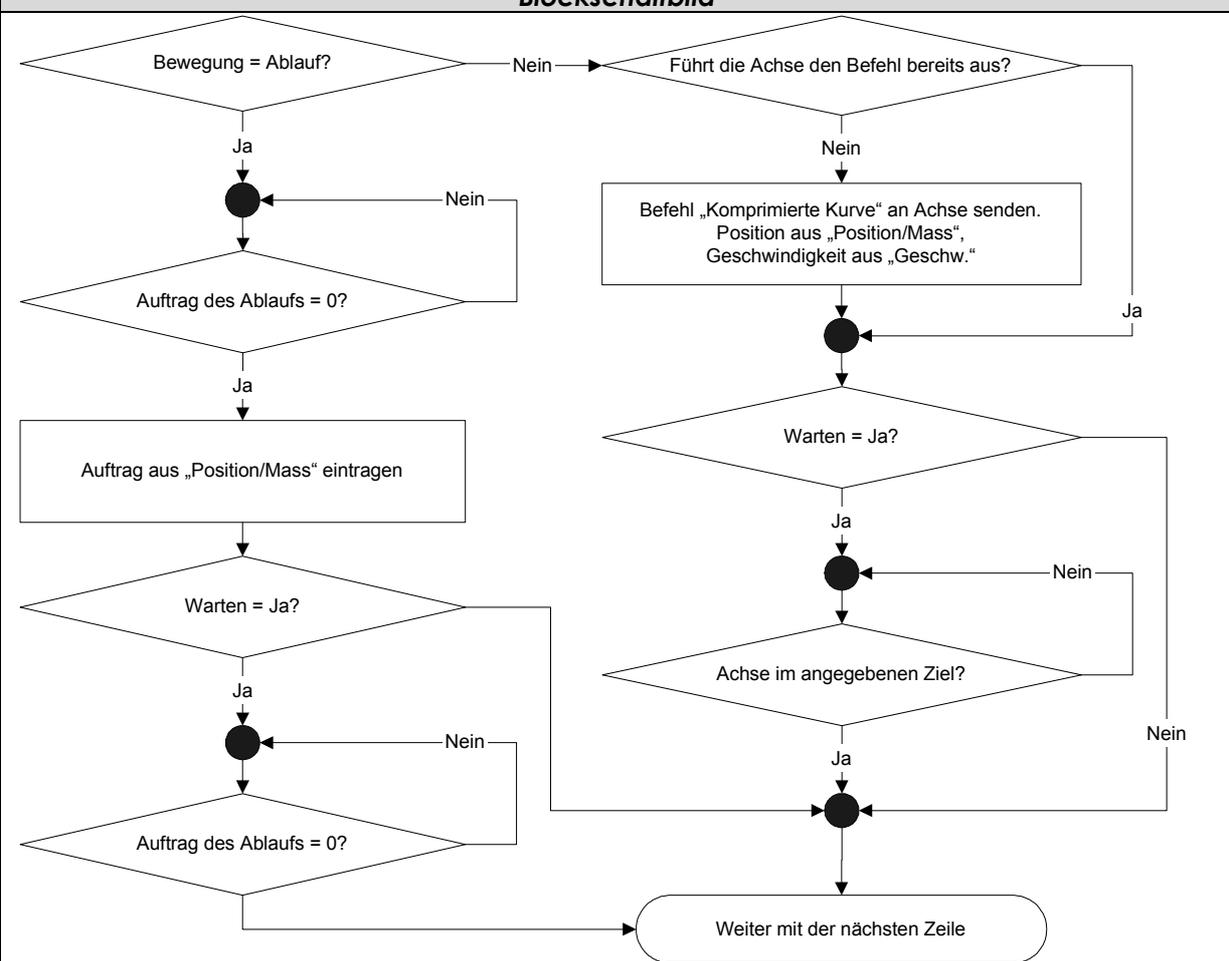
Befehl 0: Fahre auf Mass / Starte Ablauf mit Auftrag



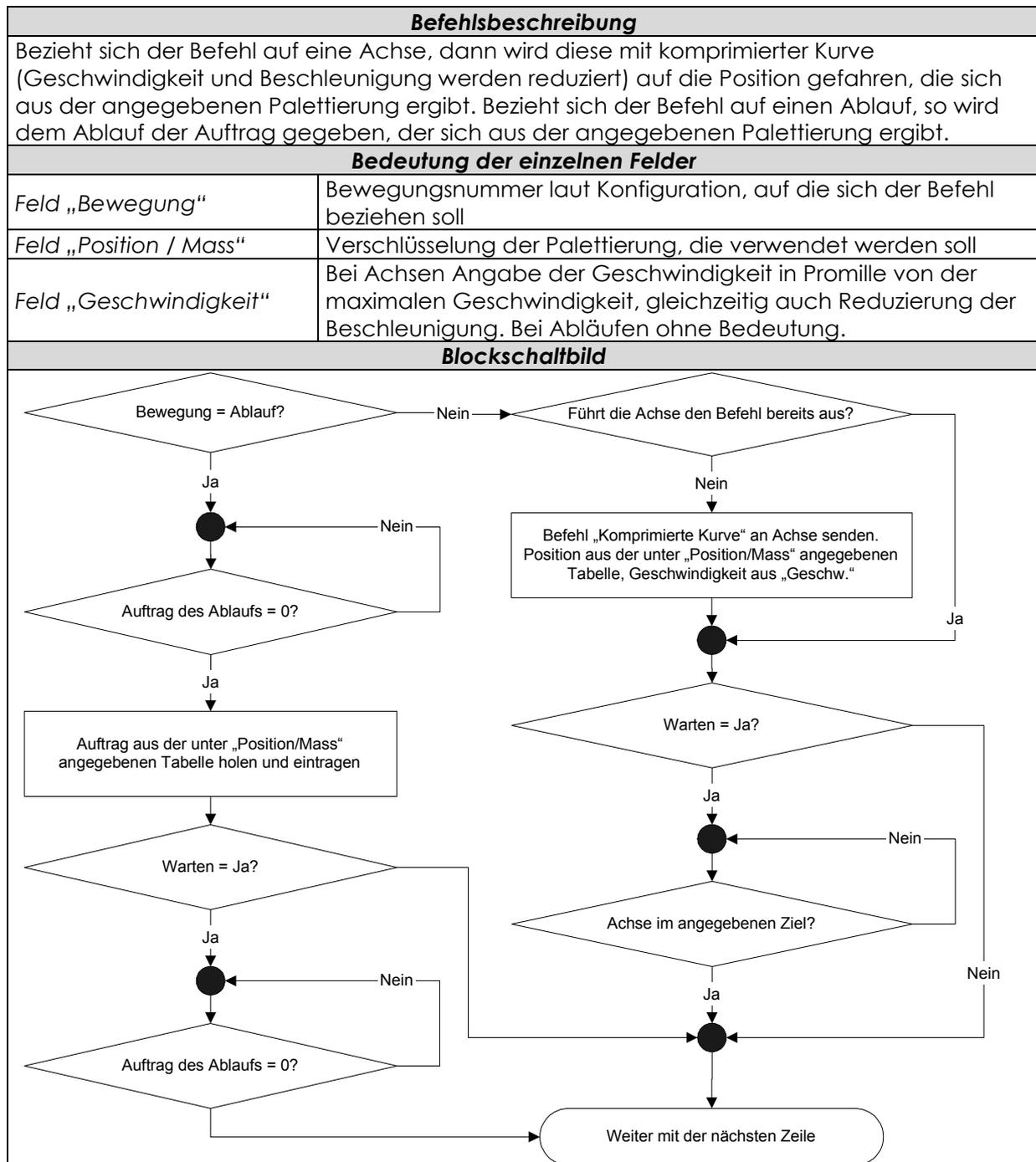
Befehl 1: Fahre auf Mass aus Palettierung / Starte Ablauf mit Auftrag aus Palettierung



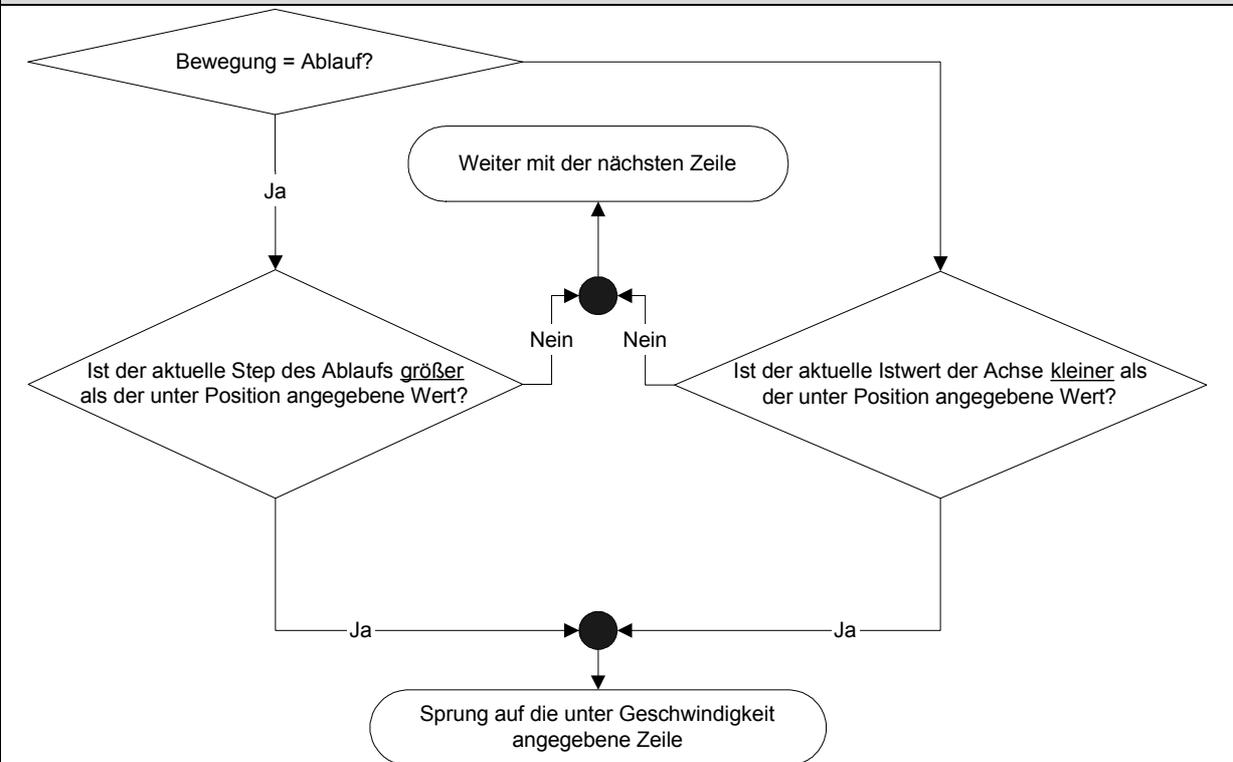
Befehl 2: Fahre auf Mass / Starte Ablauf mit Auftrag

Befehlsbeschreibung	
Bezieht sich der Befehl auf eine Achse, dann wird diese mit komprimierter Kurve (Geschwindigkeit und Beschleunigung werden reduziert) auf die angegebene Position gefahren. Bezieht sich der Befehl auf einen Ablauf, so wird dem Ablauf der angegebene Auftrag gegeben.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Bei Achsen Angabe der Zielposition, bei Abläufen Angabe des Auftrags, der erteilt werden soll
Feld „Geschwindigkeit“	Bei Achsen Angabe der Geschwindigkeit in Promille von der maximalen Geschwindigkeit, gleichzeitig auch Reduzierung der Beschleunigung. Bei Abläufen ohne Bedeutung.
Blockschaltbild	
 <pre> graph TD Start(()) --> D1{Bewegung = Ablauf?} D1 -- Ja --> J1(()) D1 -- Nein --> D2{Führt die Achse den Befehl bereits aus?} D2 -- Ja --> J2(()) D2 -- Nein --> B1[Befehl „Komprimierte Kurve“ an Achse senden. Position aus „Position/Mass“, Geschwindigkeit aus „Geschw.“] B1 --> J2 J1 --> D3{Warten = Ja?} J2 --> D3 D3 -- Ja --> J3(()) D3 -- Nein --> D4{Achse im angegebenen Ziel?} D4 -- Ja --> J3 D4 -- Nein --> D5{Auftrag des Ablaufs = 0?} D5 -- Ja --> B2[Auftrag aus „Position/Mass“ eintragen] B2 --> D3 D5 -- Nein --> J3 D3 --> J3 D3 --> D5 D5 --> End([Weiter mit der nächsten Zeile]) J3 --> End </pre>	

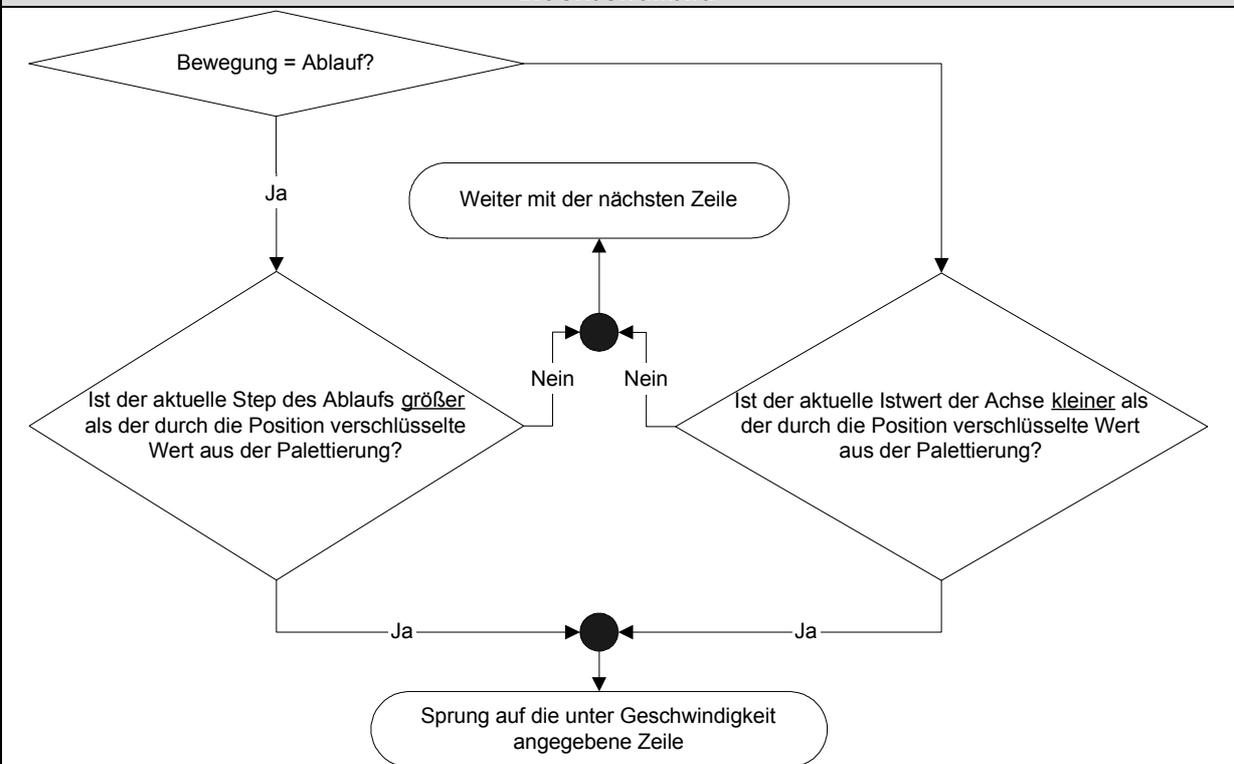
Befehl 3: Fahre auf Mass aus Palettierung / Starte Ablauf mit Auftrag aus Palettierung



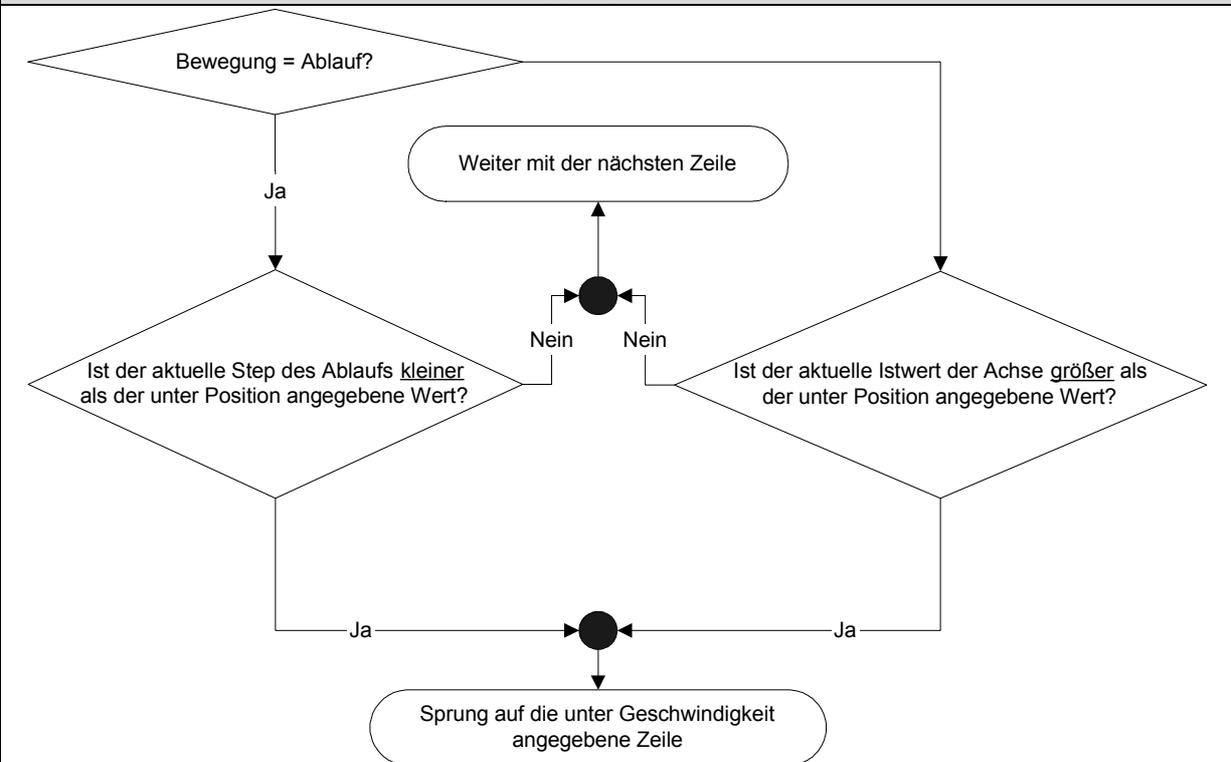
Befehl 6: Sprung wenn Istwert < Mass / Sprung wenn Step > Mass

Befehlsbeschreibung	
<p>Bezieht sich der Befehl auf eine Achse, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der aktuelle Istwert der Achse <u>kleiner</u> als die angegebene Position ist. Bezieht sich der Befehl auf einen Ablauf, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der Step des Ablaufs <u>größer</u> als die angegebene Position ist. <i>Dieser Befehl arbeitet immer im Modus Warten Nein.</i></p>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Bei Achsen Angabe der Vergleichsposition, bei Abläufen Angabe des Steps, mit dem verglichen werden soll
Feld „Geschwindigkeit“	Angabe der Zeile, auf die gesprungen werden soll, wenn die Bedingung erfüllt ist..
Blockschaltbild	
 <pre> graph TD Start(()) --> Q1{Bewegung = Ablauf?} Q1 -- Ja --> Q2{Ist der aktuelle Step des Ablaufs größer als der unter Position angegebene Wert?} Q1 -- Nein --> Q3{Ist der aktuelle Istwert der Achse kleiner als der unter Position angegebene Wert?} Q2 -- Ja --> J1(()) Q2 -- Nein --> J2(()) Q3 -- Ja --> J1 Q3 -- Nein --> J2 J1 --> A1(Sprung auf die unter Geschwindigkeit angegebene Zeile) J2 --> A2(Weiter mit der nächsten Zeile) A1 --> J2 A2 --> J2 J2 --> End(()) </pre> <p>The flowchart starts with a decision diamond 'Bewegung = Ablauf?'. If 'Ja', it proceeds to 'Ist der aktuelle Step des Ablaufs größer als der unter Position angegebene Wert?'. If 'Nein', it proceeds to 'Ist der aktuelle Istwert der Achse kleiner als der unter Position angegebene Wert?'. Both 'Ja' answers lead to a central junction point. From this junction, a 'Ja' path leads to 'Sprung auf die unter Geschwindigkeit angegebene Zeile', and a 'Nein' path leads to 'Weiter mit der nächsten Zeile'. Both paths then merge back into the central junction point.</p>	

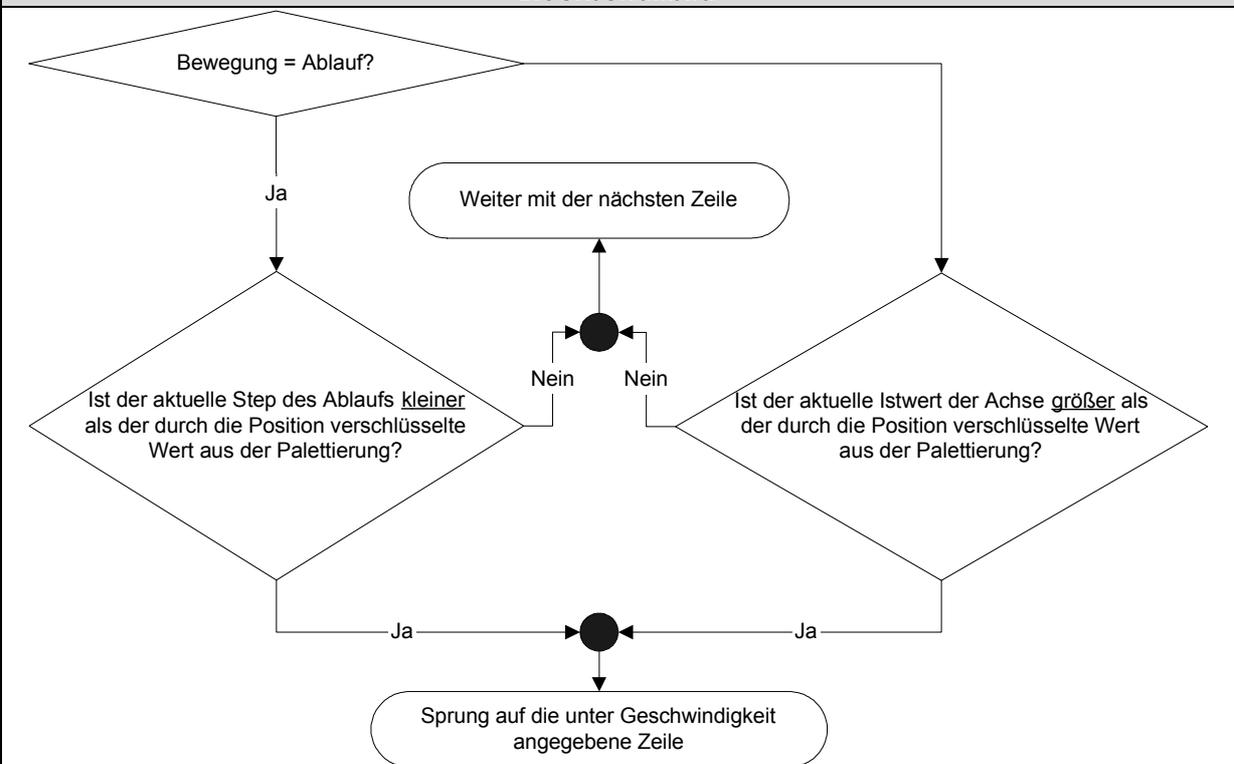
Befehl 7: Sprung wenn Istwert < Mass aus Palettierung / Sprung wenn Step > Mass aus Palettierung

Befehlsbeschreibung	
<p>Bezieht sich der Befehl auf eine Achse, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der aktuelle Istwert der Achse <u>kleiner</u> als die durch die Palettierung verschlüsselte Position ist. Bezieht sich der Befehl auf einen Ablauf, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der Step des Ablaufs <u>größer</u> als der durch die Palettierung verschlüsselte Step ist. <i>Dieser Befehl arbeitet immer im Modus Warten Nein.</i></p>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Angabe der Zeile, auf die gesprungen werden soll, wenn die Bedingung erfüllt ist..
Blockschaltbild	
 <pre> graph TD Start(()) --> Q1{Bewegung = Ablauf?} Q1 -- Ja --> Q2{Ist der aktuelle Step des Ablaufs größer als der durch die Position verschlüsselte Wert aus der Palettierung?} Q1 -- Nein --> Q3{Ist der aktuelle Istwert der Achse kleiner als der durch die Position verschlüsselte Wert aus der Palettierung?} Q2 -- Ja --> J1(()) Q2 -- Nein --> J2(()) Q3 -- Ja --> J1 Q3 -- Nein --> J2 J1 --> Action1(Sprung auf die unter Geschwindigkeit angegebene Zeile) J2 --> Action2(Weiter mit der nächsten Zeile) Action1 --> End(()) Action2 --> End </pre>	

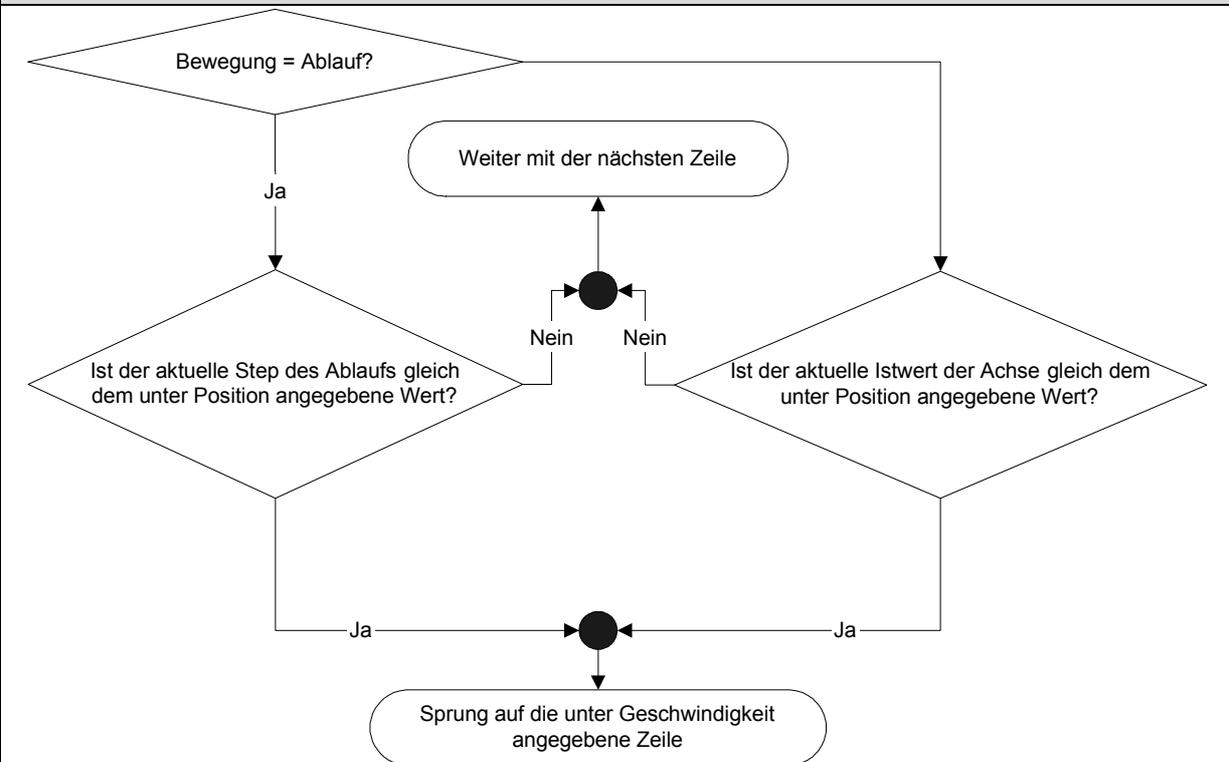
Befehl 8: Sprung wenn Istwert > Mass / Sprung wenn Step < Mass

Befehlsbeschreibung	
<p>Bezieht sich der Befehl auf eine Achse, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der aktuelle Istwert der Achse <u>größer</u> als die angegebene Position ist. Bezieht sich der Befehl auf einen Ablauf, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der Step des Ablaufs <u>kleiner</u> als die angegebene Position ist. <i>Dieser Befehl arbeitet immer im Modus Warten Nein.</i></p>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Bei Achsen Angabe der Vergleichsposition, bei Abläufen Angabe des Steps, mit dem verglichen werden soll
Feld „Geschwindigkeit“	Angabe der Zeile, auf die gesprungen werden soll, wenn die Bedingung erfüllt ist..
Blockschaltbild	
 <pre> graph TD Start(()) --> Q1{Bewegung = Ablauf?} Q1 -- Ja --> Q2{Ist der aktuelle Step des Ablaufs kleiner als der unter Position angegebene Wert?} Q1 -- Nein --> Q3{Ist der aktuelle Istwert der Achse größer als der unter Position angegebene Wert?} Q2 -- Ja --> J1(()) Q2 -- Nein --> J2(()) Q3 -- Ja --> J1 Q3 -- Nein --> J2 J1 --> S1(Sprung auf die unter Geschwindigkeit angegebene Zeile) J2 --> S2(Weiter mit der nächsten Zeile) S1 --> J2 S2 --> J2 </pre> <p>The flowchart illustrates the logic of Befehl 8. It starts with a decision diamond 'Bewegung = Ablauf?'. If 'Ja' (Yes), it proceeds to 'Ist der aktuelle Step des Ablaufs kleiner als der unter Position angegebene Wert?'. If 'Nein' (No), it proceeds to 'Ist der aktuelle Istwert der Achse größer als der unter Position angegebene Wert?'. Both 'Ja' paths lead to a central junction point, which then leads to the action 'Sprung auf die unter Geschwindigkeit angegebene Zeile'. Both 'Nein' paths lead to another central junction point, which then leads to the action 'Weiter mit der nächsten Zeile'. There are also feedback loops from both action boxes back to their respective junction points.</p>	

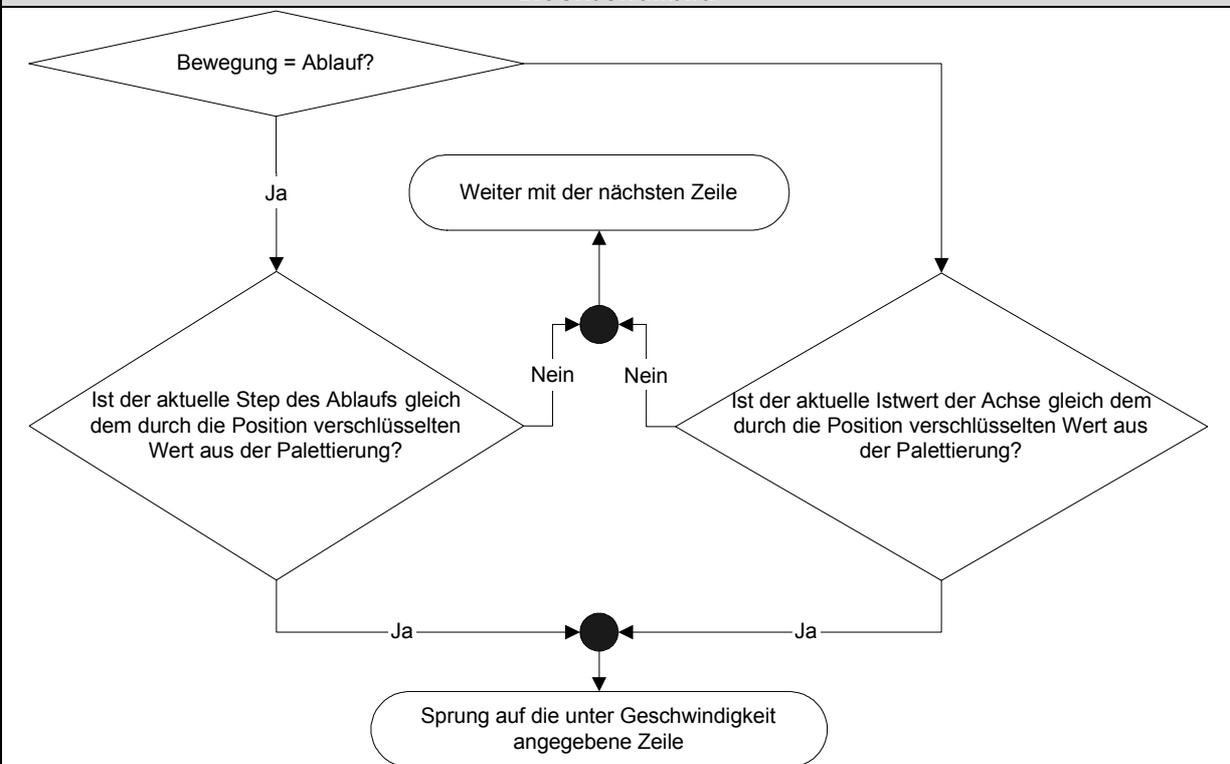
Befehl 9: Sprung wenn Istwert > Mass aus Palettierung / Sprung wenn Step < Mass aus Palettierung

Befehlsbeschreibung	
<p>Bezieht sich der Befehl auf eine Achse, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der aktuelle Istwert der Achse <u>größer</u> als die durch die Palettierung verschlüsselte Position ist. Bezieht sich der Befehl auf einen Ablauf, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der Step des Ablaufs <u>kleiner</u> als der durch die Palettierung verschlüsselte Step ist. <i>Dieser Befehl arbeitet immer im Modus Warten Nein.</i></p>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Angabe der Zeile, auf die gesprungen werden soll, wenn die Bedingung erfüllt ist..
Blockschaltbild	
 <pre> graph TD Start(()) --> Q1{Bewegung = Ablauf?} Q1 -- Ja --> Q2{Ist der aktuelle Step des Ablaufs kleiner als der durch die Position verschlüsselte Wert aus der Palettierung?} Q1 -- Nein --> Q3{Ist der aktuelle Istwert der Achse größer als der durch die Position verschlüsselte Wert aus der Palettierung?} Q2 -- Ja --> J1(()) Q2 -- Nein --> J2(()) Q3 -- Ja --> J1 Q3 -- Nein --> J2 J1 --> Action1(Sprung auf die unter Geschwindigkeit angegebene Zeile) J2 --> Action2(Weiter mit der nächsten Zeile) </pre>	

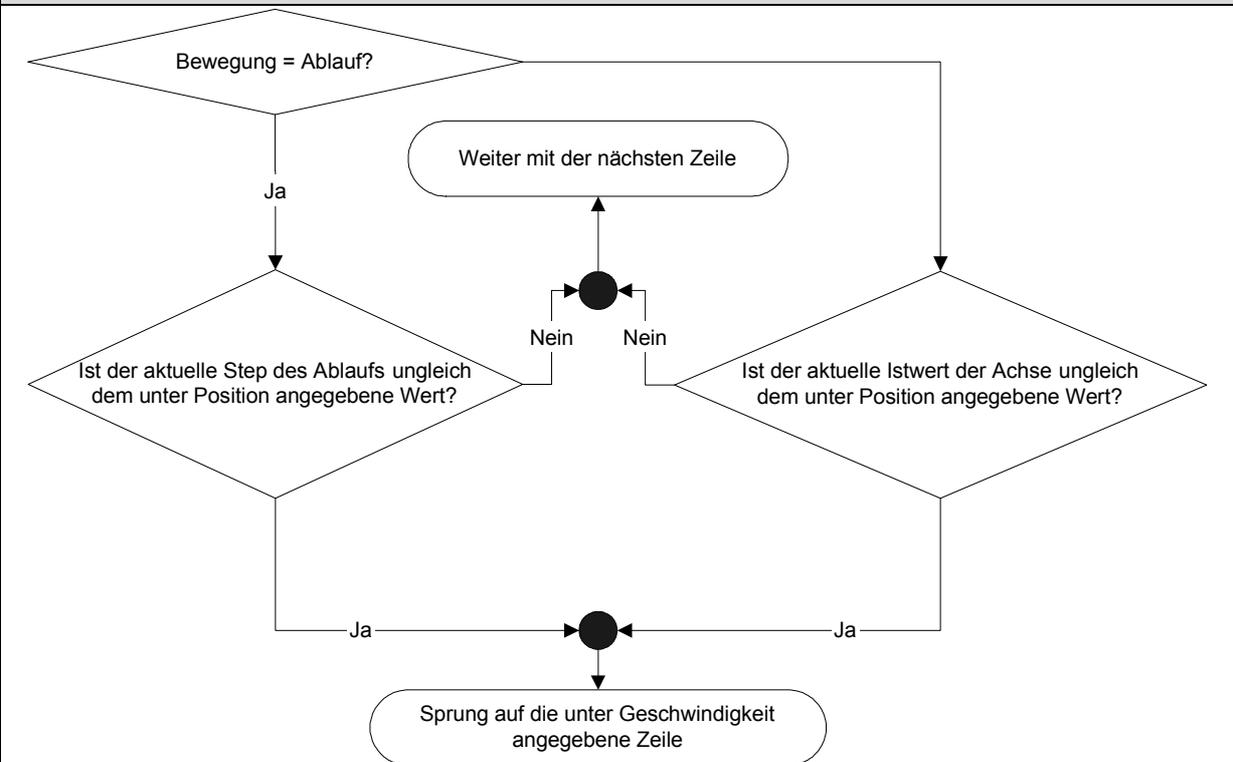
Befehl 10: Sprung wenn Istwert = Mass / Sprung wenn Step = Mass

Befehlsbeschreibung	
<p>Bezieht sich der Befehl auf eine Achse, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der aktuelle Istwert der Achse <u>gleich</u> der angegebenen Position ist. Bezieht sich der Befehl auf einen Ablauf, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der Step des Ablaufs <u>gleich</u> dem unter Position ist angegebenen Wert ist. <i>Dieser Befehl arbeitet immer im Modus Warten Nein.</i></p>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Bei Achsen Angabe der Vergleichsposition, bei Abläufen Angabe des Steps, mit dem verglichen werden soll
Feld „Geschwindigkeit“	Angabe der Zeile, auf die gesprungen werden soll, wenn die Bedingung erfüllt ist..
Blockschaltbild	
 <pre> graph TD Start(()) --> Q1{Bewegung = Ablauf?} Q1 -- Ja --> Q2{Ist der aktuelle Step des Ablaufs gleich dem unter Position angegebene Wert?} Q1 -- Nein --> Q3{Ist der aktuelle Istwert der Achse gleich dem unter Position angegebene Wert?} Q2 -- Ja --> J1(()) Q2 -- Nein --> J2(()) Q3 -- Ja --> J1 Q3 -- Nein --> J2 J1 --> A1[Springung auf die unter Geschwindigkeit angegebene Zeile] J2 --> A2[Weiter mit der nächsten Zeile] A1 --> J2 A2 --> J2 </pre>	

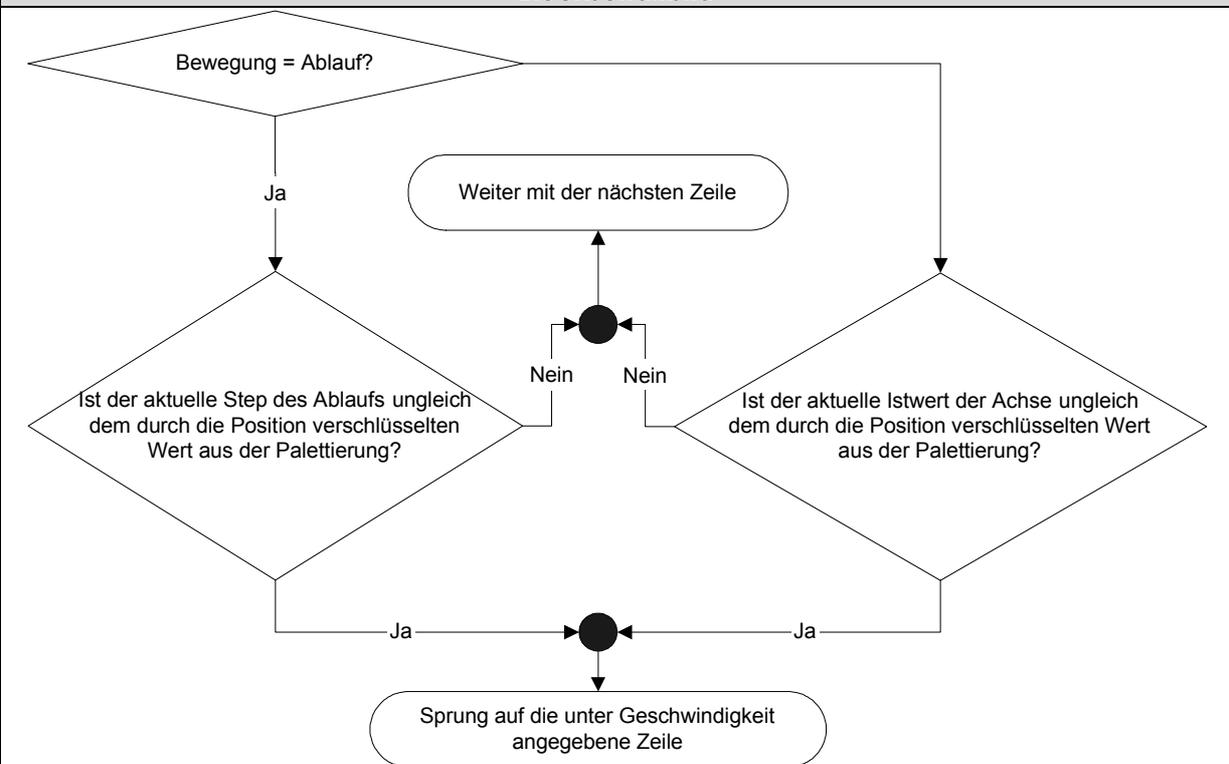
Befehl 11: Sprung wenn Istwert = Mass aus Palettierung / Sprung wenn Step = Mass aus Palettierung

Befehlsbeschreibung	
<p>Bezieht sich der Befehl auf eine Achse, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der aktuelle Istwert der Achse <u>gleich</u> der durch die Palettierung verschlüsselten Position ist. Bezieht sich der Befehl auf einen Ablauf, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der Step des Ablaufs <u>gleich</u> dem durch die Palettierung verschlüsselten Step ist. <i>Dieser Befehl arbeitet immer im Modus Warten Nein.</i></p>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Angabe der Zeile, auf die gesprungen werden soll, wenn die Bedingung erfüllt ist..
Blockschaltbild	
 <pre> graph TD Start(()) --> Q1{Bewegung = Ablauf?} Q1 -- Ja --> Q2{Ist der aktuelle Step des Ablaufs gleich dem durch die Position verschlüsselten Wert aus der Palettierung?} Q1 --> Q3{Ist der aktuelle Istwert der Achse gleich dem durch die Position verschlüsselten Wert aus der Palettierung?} Q2 -- Ja --> J1(()) Q2 -- Nein --> J2(()) Q3 -- Ja --> J1 Q3 -- Nein --> J2 J1 --> Action1(Sprung auf die unter Geschwindigkeit angegebene Zeile) J2 --> Action2(Weiter mit der nächsten Zeile) Action1 --> End(()) Action2 --> End </pre>	

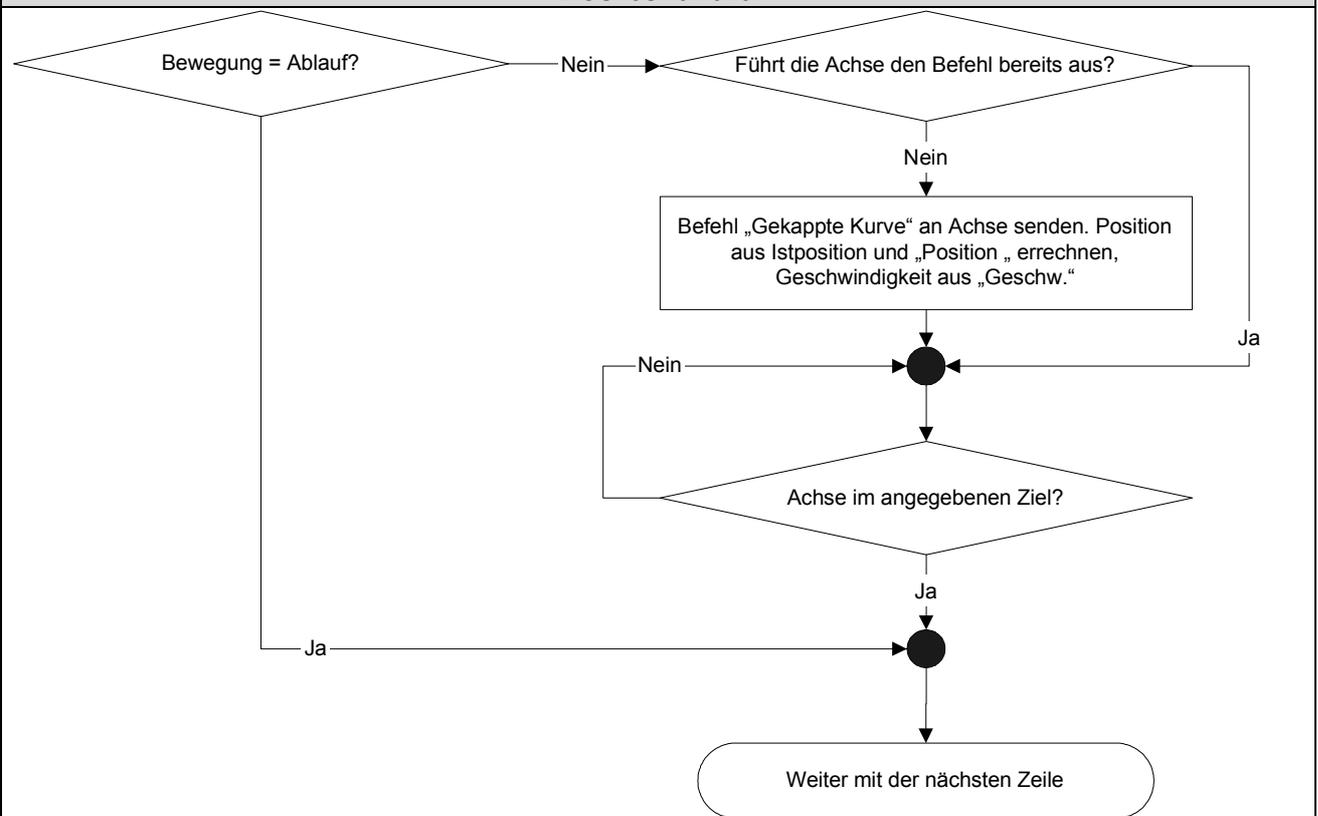
Befehl 12: Sprung wenn Istwert <> Mass / Sprung wenn Step <> Mass

Befehlsbeschreibung	
<p>Bezieht sich der Befehl auf eine Achse, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der aktuelle Istwert der Achse <u>ungleich</u> der angegebenen Position ist. Bezieht sich der Befehl auf einen Ablauf, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der Step des Ablaufs <u>ungleich</u> dem unter Position ist angegebenen Wert ist. <i>Dieser Befehl arbeitet immer im Modus Warten Nein.</i></p>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Bei Achsen Angabe der Vergleichsposition, bei Abläufen Angabe des Steps, mit dem verglichen werden soll
Feld „Geschwindigkeit“	Angabe der Zeile, auf die gesprungen werden soll, wenn die Bedingung erfüllt ist..
Blockschaltbild	
 <pre> graph TD Start(()) --> Q1{Bewegung = Ablauf?} Q1 -- Ja --> Q2{Ist der aktuelle Step des Ablaufs ungleich dem unter Position angegebene Wert?} Q1 -- Nein --> Q3{Ist der aktuelle Istwert der Achse ungleich dem unter Position angegebene Wert?} Q2 -- Ja --> J1(()) Q2 -- Nein --> J2(()) Q3 -- Ja --> J1 Q3 -- Nein --> J2 J1 --> A1[Sprung auf die unter Geschwindigkeit angegebene Zeile] J2 --> A2[Weiter mit der nächsten Zeile] A1 --> J2 A2 --> J2 </pre>	

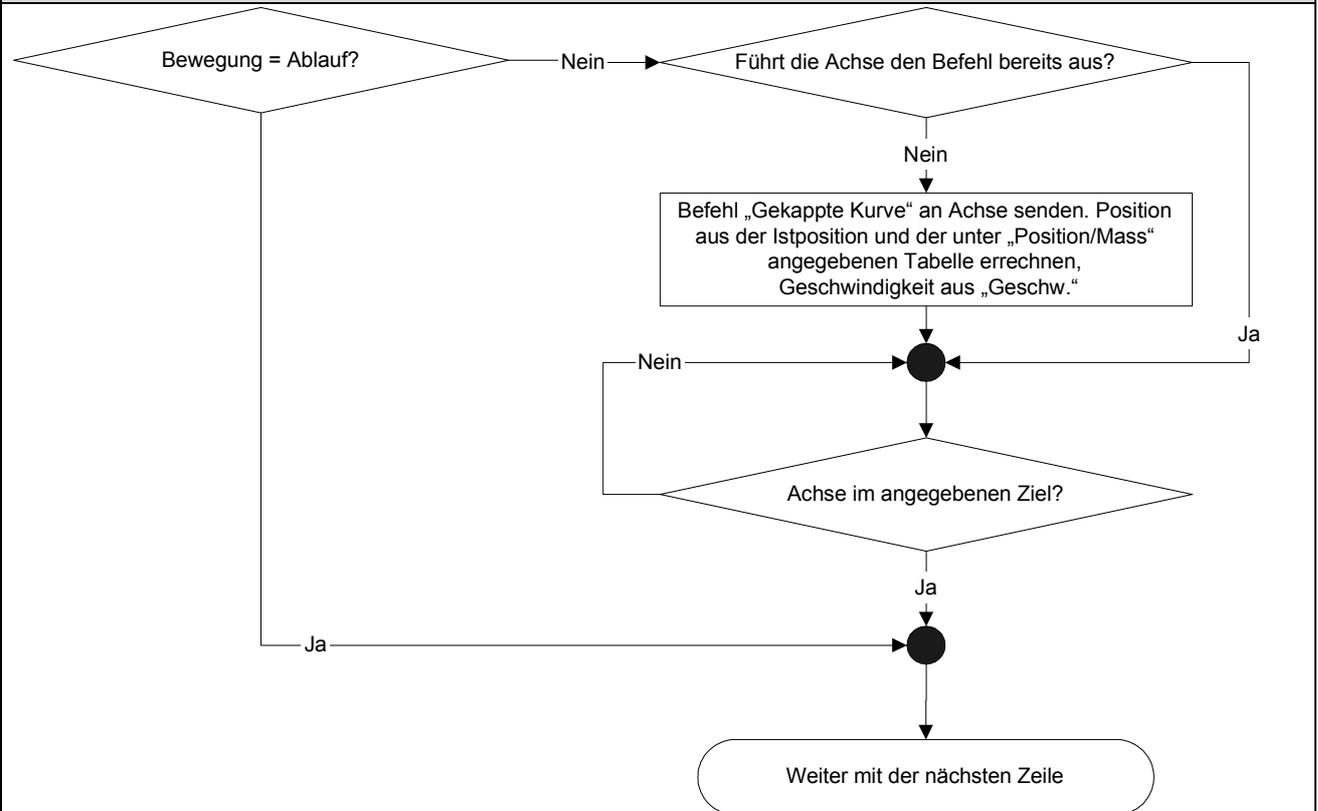
Befehl 13: Sprung wenn Istwert <> Mass aus Palettierung / Sprung wenn Step <> Mass aus Palettierung

Befehlsbeschreibung	
<p>Bezieht sich der Befehl auf eine Achse, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der aktuelle Istwert der Achse <u>ungleich</u> der durch die Palettierung verschlüsselten Position ist. Bezieht sich der Befehl auf einen Ablauf, dann springt die Programmierung auf die unter Geschwindigkeit angegebene Zeile, wenn der Step des Ablaufs <u>ungleich</u> dem durch die Palettierung verschlüsselten Step ist. <i>Dieser Befehl arbeitet immer im Modus Warten Nein.</i></p>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Angabe der Zeile, auf die gesprungen werden soll, wenn die Bedingung erfüllt ist..
Blockschaltbild	
 <pre> graph TD Start(()) --> Q1{Bewegung = Ablauf?} Q1 -- Ja --> Q2{Ist der aktuelle Step des Ablaufs ungleich dem durch die Position verschlüsselten Wert aus der Palettierung?} Q1 -- Nein --> Q3{Ist der aktuelle Istwert der Achse ungleich dem durch die Position verschlüsselten Wert aus der Palettierung?} Q2 -- Ja --> J1(()) Q2 -- Nein --> J2(()) Q3 -- Ja --> J1 Q3 -- Nein --> J2 J1 --> Action1(Sprung auf die unter Geschwindigkeit angegebene Zeile) J2 --> Action2(Weiter mit der nächsten Zeile) Action1 --> Start Action2 --> Start </pre> <p>The flowchart starts with a decision diamond 'Bewegung = Ablauf?'. If 'Ja', it proceeds to 'Ist der aktuelle Step des Ablaufs ungleich dem durch die Position verschlüsselten Wert aus der Palettierung?'. If 'Nein', it proceeds to 'Ist der aktuelle Istwert der Achse ungleich dem durch die Position verschlüsselten Wert aus der Palettierung?'. Both 'Ja' answers lead to a central junction point, which then leads to the action 'Sprung auf die unter Geschwindigkeit angegebene Zeile'. Both 'Nein' answers lead to another central junction point, which then leads to the action 'Weiter mit der nächsten Zeile'. Both actions loop back to the start of the flowchart.</p>	

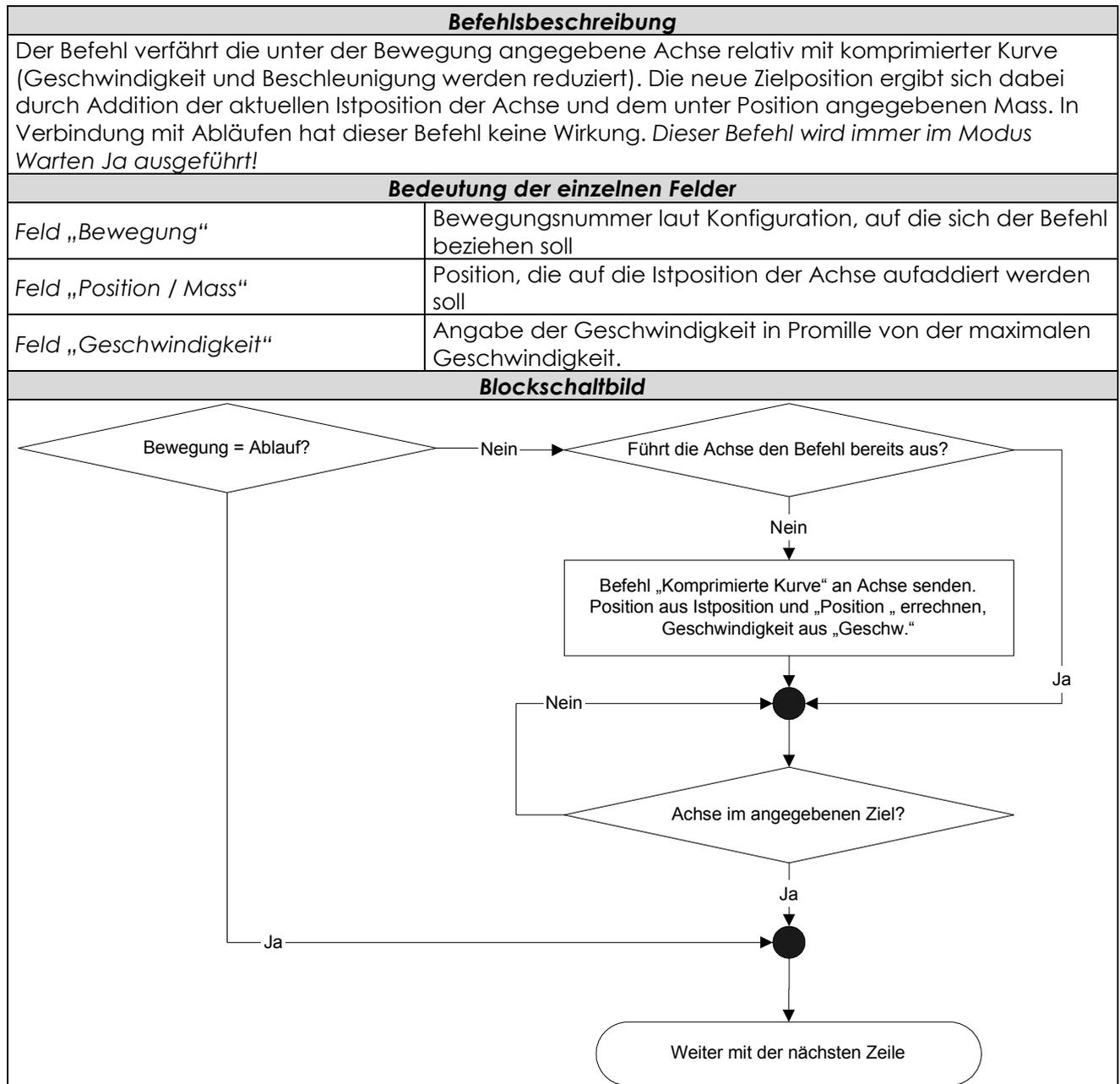
Befehl 20: Verfahre relativ um Mass, gekappte Kurve

Befehlsbeschreibung	
Der Befehl verfährt die unter der Bewegung angegebene Achse relativ mit gekappter Kurve. Die neue Zielposition ergibt sich dabei durch Addition der aktuellen Istposition der Achse und dem unter Position angegebenen Mass. In Verbindung mit Abläufen hat dieser Befehl keine Wirkung. <i>Dieser Befehl wird immer im Modus Warten Ja ausgeführt!</i>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Position, die auf die Istposition der Achse aufaddiert werden soll
Feld „Geschwindigkeit“	Angabe der Geschwindigkeit in Promille von der maximalen Geschwindigkeit.
Blockschaltbild	
 <pre> graph TD Start(()) --> D1{Bewegung = Ablauf?} D1 -- Ja --> M1(()) D1 -- Nein --> D2{Führt die Achse den Befehl bereits aus?} D2 -- Ja --> M1 D2 -- Nein --> P[Befehl „Gekappte Kurve“ an Achse senden. Position aus Istposition und „Position „ errechnen, Geschwindigkeit aus „Geschw.“] P --> M2(()) M2 --> D3{Achse im angegebenen Ziel?} D3 -- Ja --> M1 D3 -- Nein --> D2 M1 --> End([Weiter mit der nächsten Zeile]) </pre>	

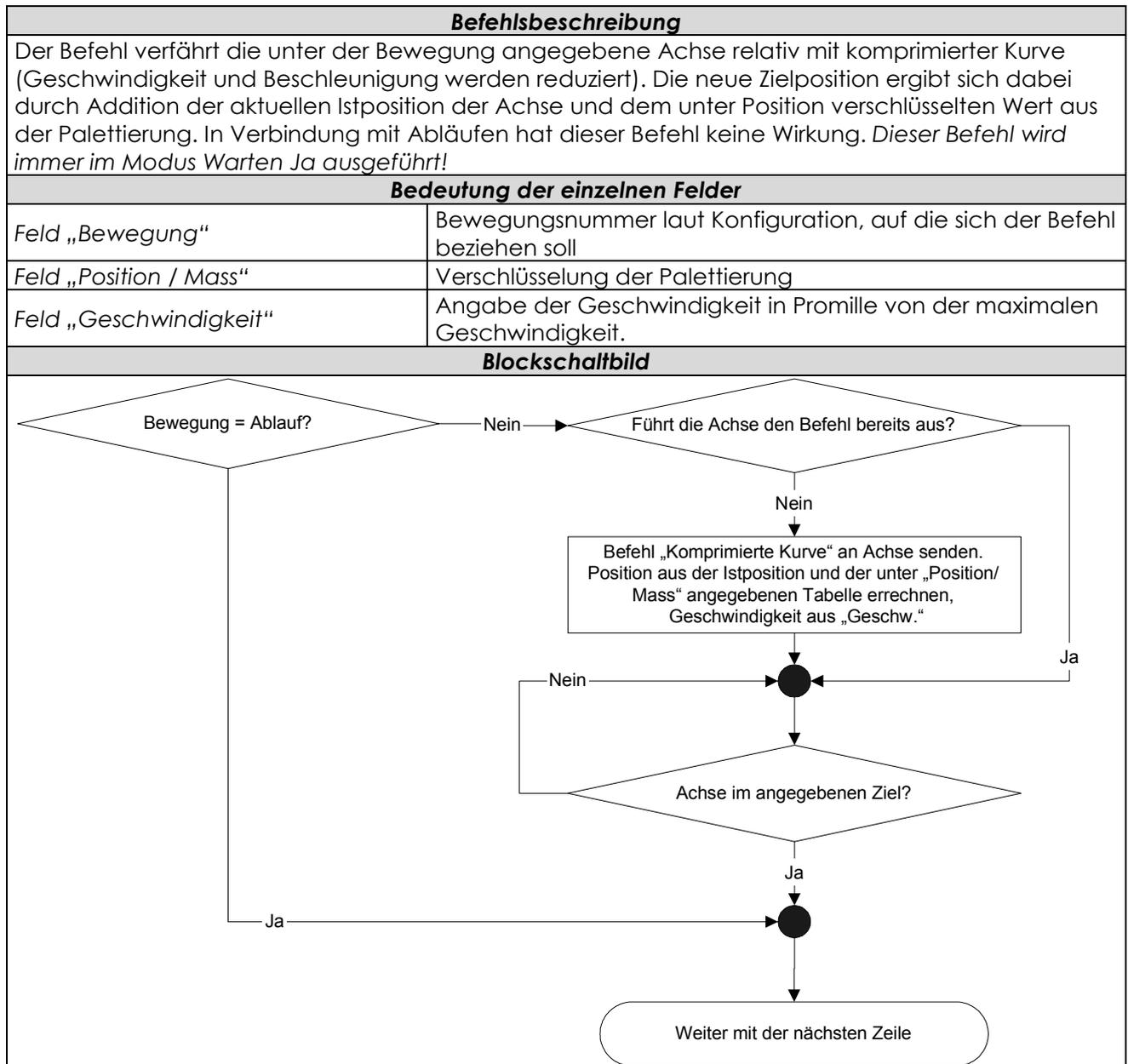
Befehl 21: Verfahre relativ um Mass aus Palettierung, gekappte Kurve

Befehlsbeschreibung	
Der Befehl verfährt die unter der Bewegung angegebene Achse relativ mit gekappter Kurve. Die neue Zielposition ergibt sich dabei durch Addition der aktuellen Istposition der Achse und dem unter Position verschlüsselten Wert aus der Palettierung. In Verbindung mit Abläufen hat dieser Befehl keine Wirkung. <i>Dieser Befehl wird immer im Modus Warten Ja ausgeführt!</i>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Angabe der Geschwindigkeit in Promille von der maximalen Geschwindigkeit.
Blockschaltbild	
 <pre> graph TD Start{Bewegung = Ablauf?} -- Ja --> Conn1(()) Start -- Nein --> Conn2{Führt die Achse den Befehl bereits aus?} Conn2 -- Ja --> Conn1 Conn2 -- Nein --> Process[Befehl „Gekappte Kurve“ an Achse senden. Position aus der Istposition und der unter „Position/Mass“ angegebenen Tabelle errechnen, Geschwindigkeit aus „Geschw.“] Process --> Conn3(()) Conn3 --> Conn4{Achse im angegebenen Ziel?} Conn4 -- Ja --> Conn1 Conn4 -- Nein --> Conn3 Conn1 --> End([Weiter mit der nächsten Zeile]) </pre>	

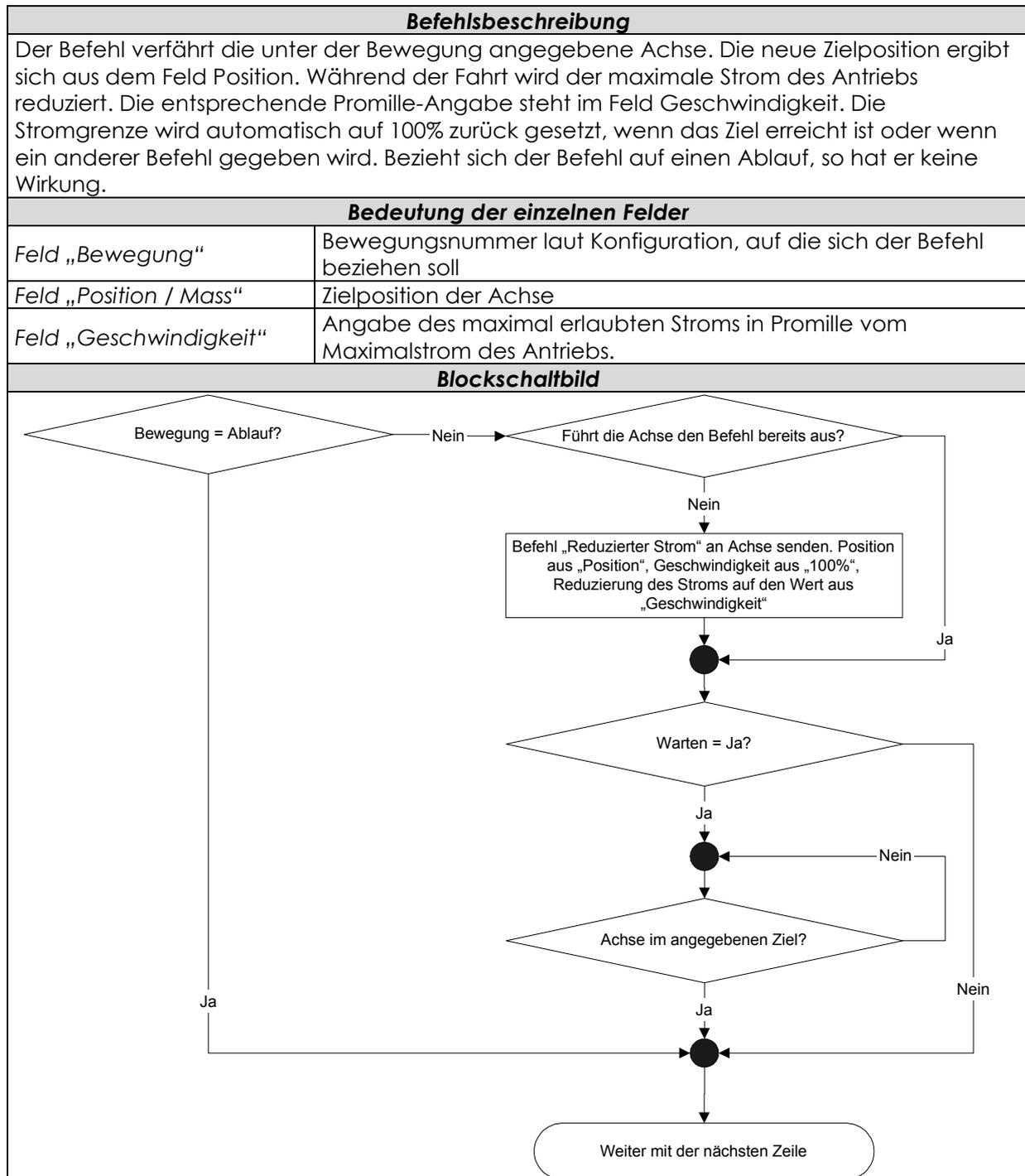
Befehl 22: Verfahre relativ um Mass, komprimierte Kurve



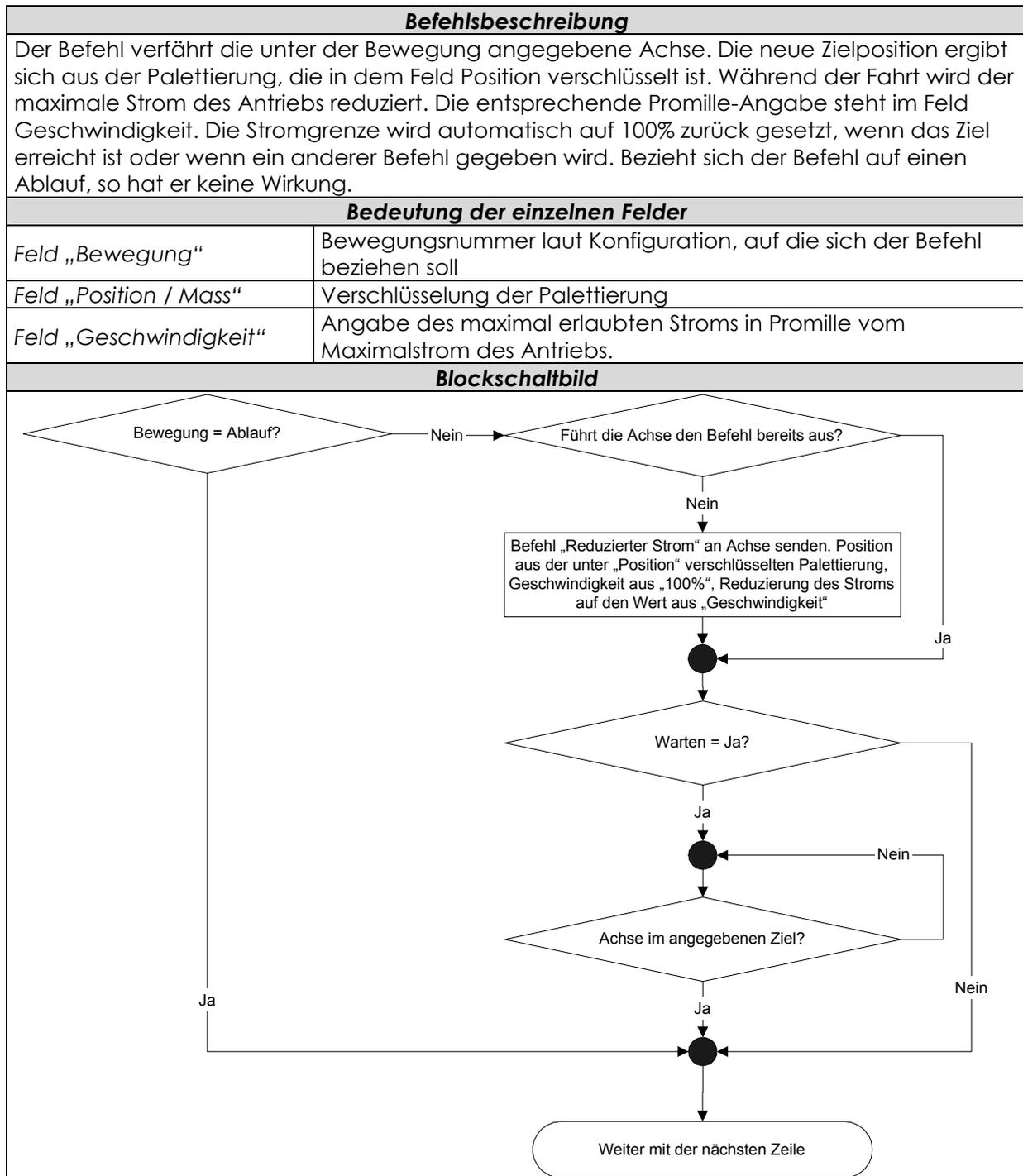
Befehl 23: Verfahre relativ um Mass aus Palettierung, komprimierte Kurve



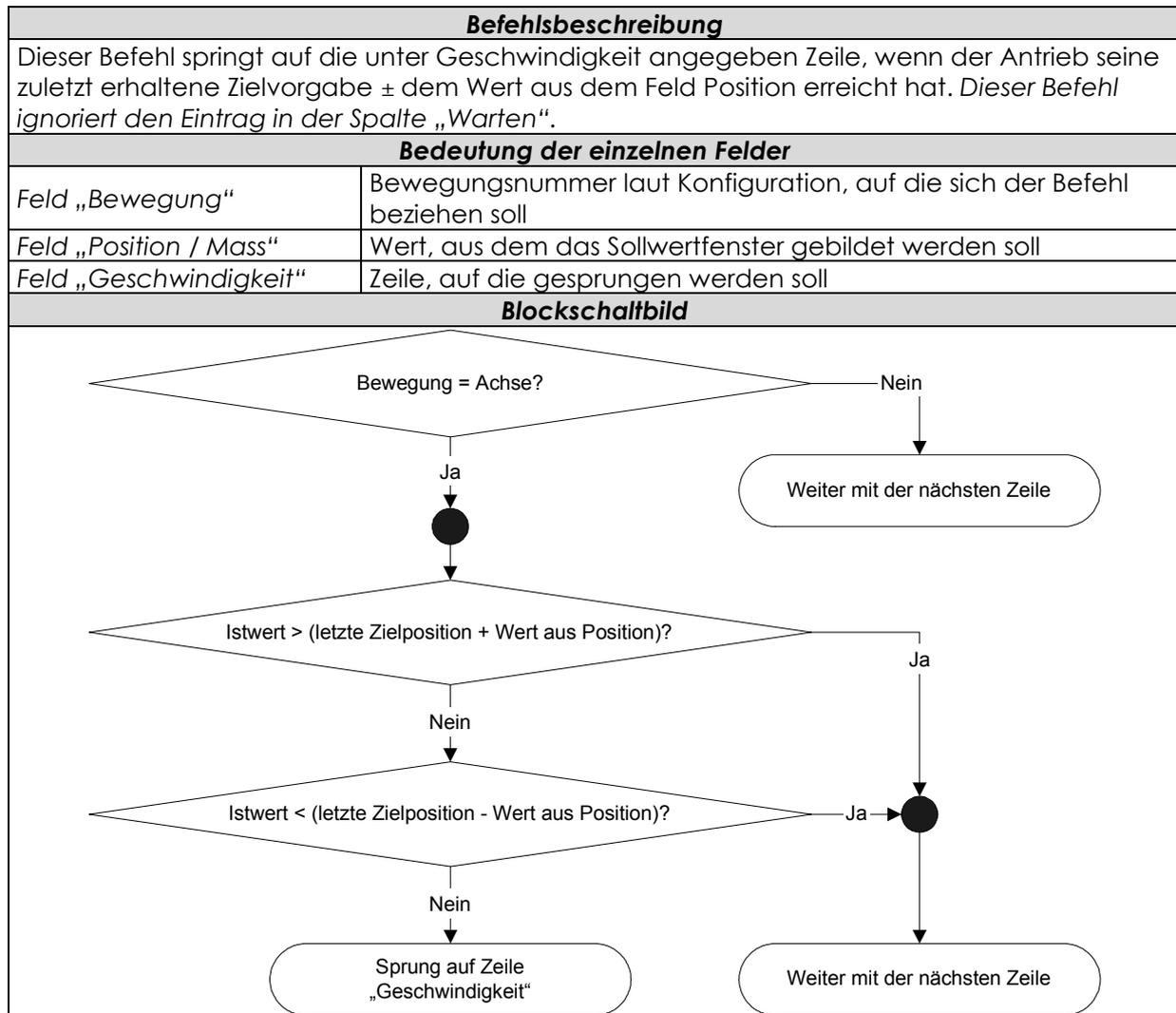
Befehl 24: Fahre auf Mass mit reduziertem Strom



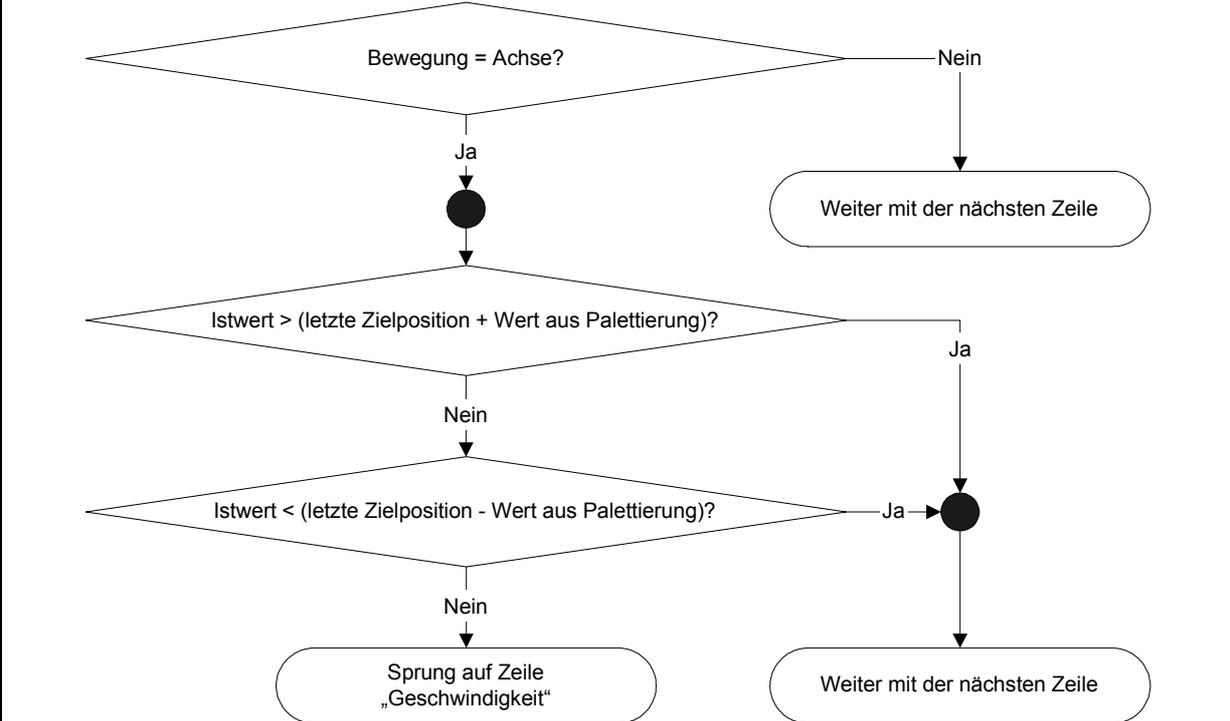
Befehl 25: Fahre auf Mass aus Palettierung mit reduziertem Strom



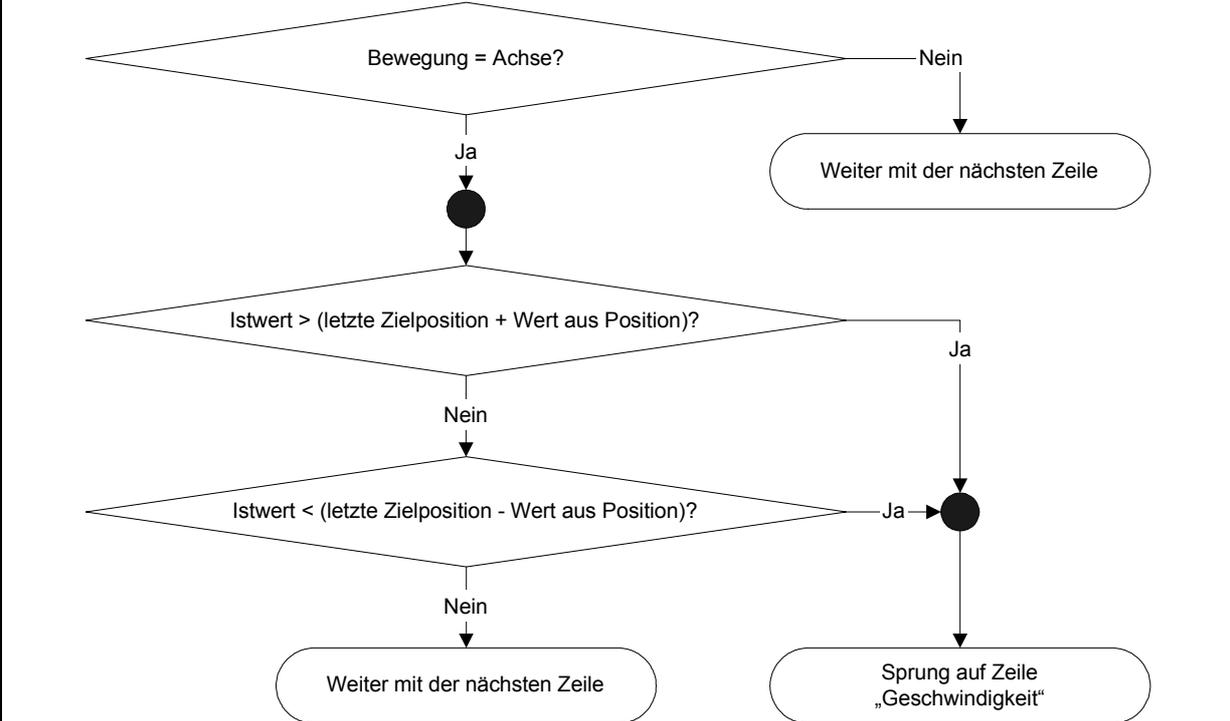
Befehl 26: Sprung wenn innerhalb Sollwertfenster Mass



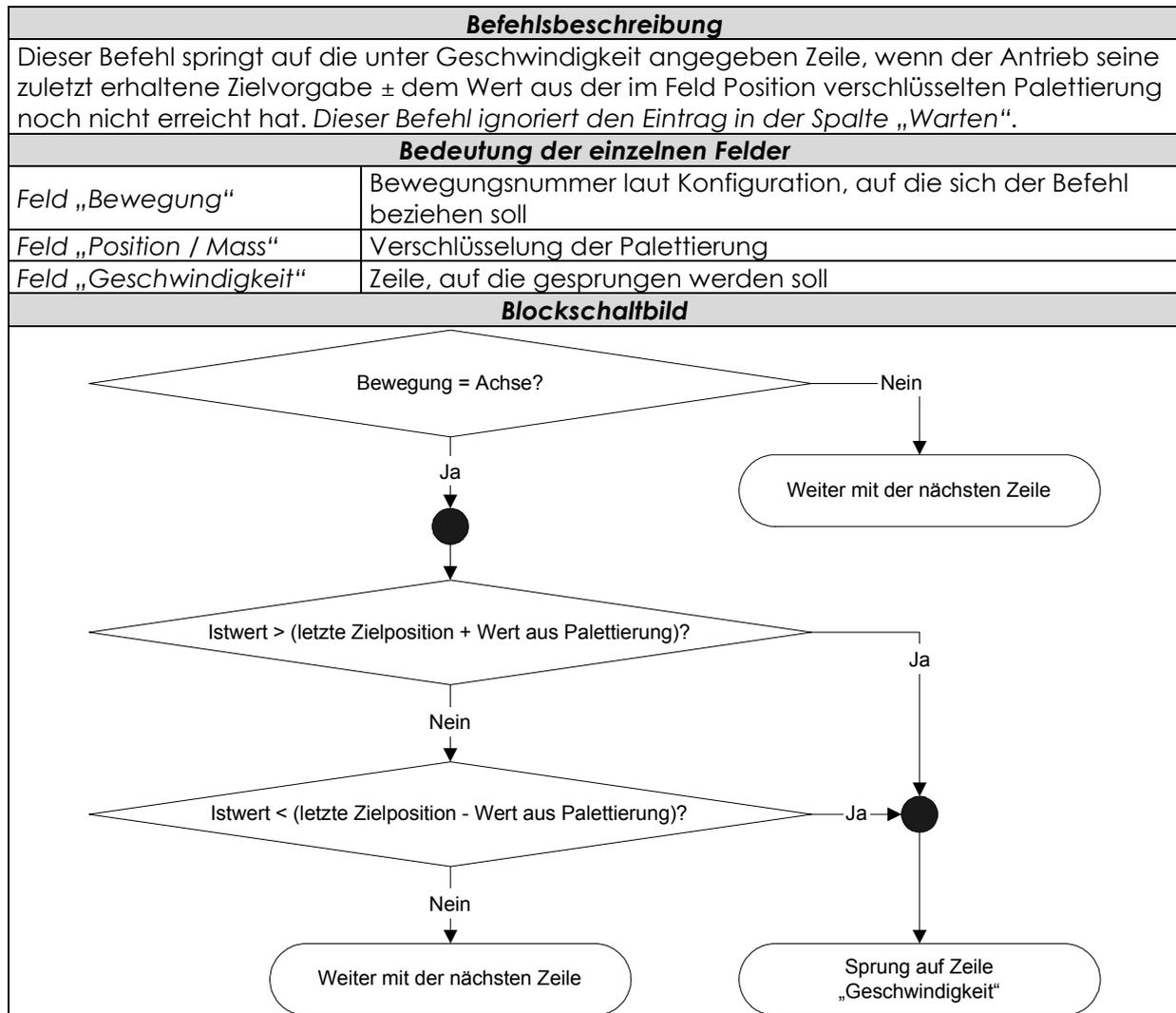
Befehl 27: Sprung wenn innerhalb Sollwertfenster Mass aus Palettierung

Befehlsbeschreibung	
Dieser Befehl springt auf die unter Geschwindigkeit angegeben Zeile, wenn der Antrieb seine zuletzt erhaltene Zielvorgabe \pm dem Wert aus der im Feld Position verschlüsselten Palettierung erreicht hat. <i>Dieser Befehl ignoriert den Eintrag in der Spalte „Warten“.</i>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Zeile, auf die gesprungen werden soll
Blockschaltbild	
 <pre> graph TD Start(()) --> D1{Bewegung = Achse?} D1 -- Ja --> J1(()) D1 -- Nein --> E1(Weiter mit der nächsten Zeile) J1 --> D2{Istwert > (letzte Zielposition + Wert aus Palettierung)?} D2 -- Ja --> J2(()) D2 -- Nein --> D3{Istwert < (letzte Zielposition - Wert aus Palettierung)?} D3 -- Ja --> J2 D3 -- Nein --> E2(Sprung auf Zeile „Geschwindigkeit“) J2 --> E3(Weiter mit der nächsten Zeile) </pre>	

Befehl 28: Sprung wenn ausserhalb Sollwertfenster Mass

Befehlsbeschreibung	
Dieser Befehl springt auf die unter Geschwindigkeit angegeben Zeile, wenn der Antrieb seine zuletzt erhaltene Zielvorgabe \pm dem Wert aus dem Feld Position noch nicht erreicht hat. Dieser Befehl ignoriert den Eintrag in der Spalte „Warten“.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Wert, aus dem das Sollwertfenster gebildet werden soll
Feld „Geschwindigkeit“	Zeile, auf die gesprungen werden soll
Blockschaltbild	
 <pre> graph TD Start(()) --> D1{Bewegung = Achse?} D1 -- Ja --> J1(()) D1 -- Nein --> E1(Weiter mit der nächsten Zeile) J1 --> D2{Istwert > (letzte Zielposition + Wert aus Position)?} D2 -- Ja --> J2(()) D2 -- Nein --> D3{Istwert < (letzte Zielposition - Wert aus Position)?} D3 -- Ja --> J2 D3 -- Nein --> E2(Weiter mit der nächsten Zeile) J2 --> E3(Sprung auf Zeile „Geschwindigkeit“) </pre>	

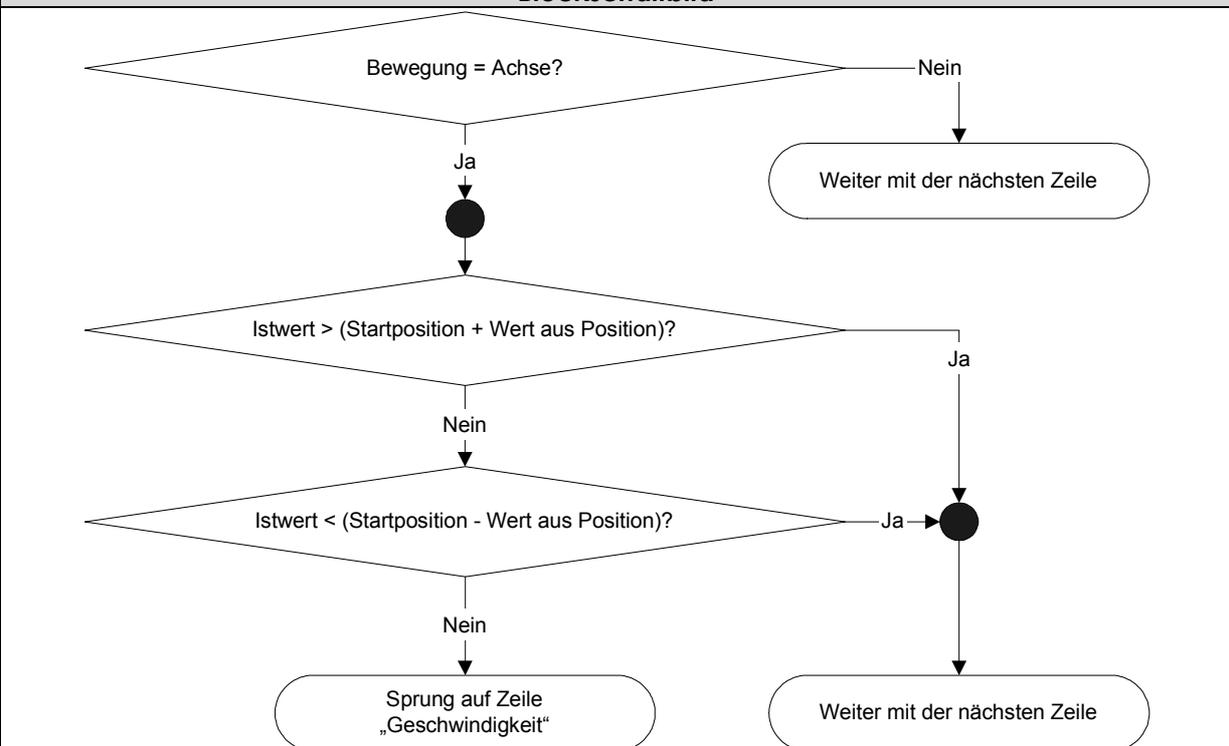
Befehl 29: Sprung wenn ausserhalb Sollwertfenster Mass aus Palettierung



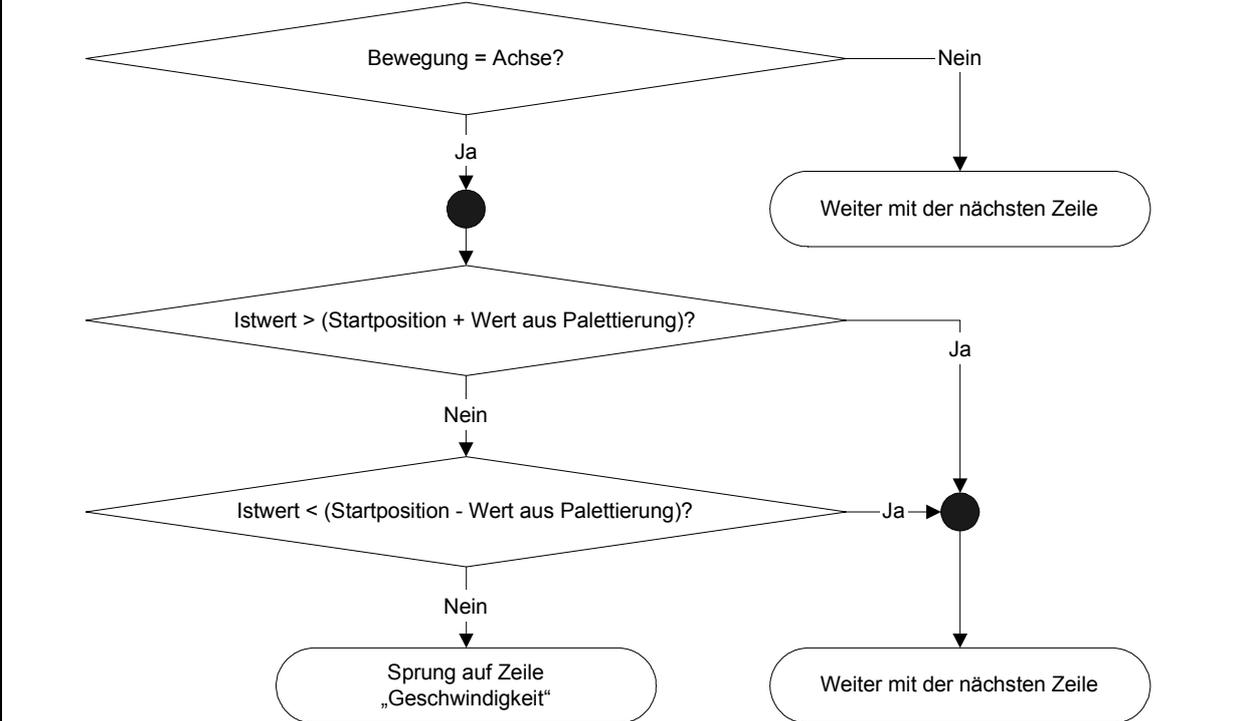
Befehl 30: Kurve laden

Befehlsbeschreibung	
Dieser Befehl lädt die Bahnpunkte einer Kurve in den SEW-Antrieb. Dabei werden die Bahnpunkte geladen, die sich in der Datei AXSE+Achsnummer+Programmnummer . SEW befinden, also beispielsweise AXSE0199.SEW für Achse 1, Programm 99.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfig, bei der der Befehl wirksam werden soll. Es muss sich dabei um einen SEW-Antrieb handeln.
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine. Das Teach-In wartet an dieser Stelle immer, bis die Kurve vollständig geladen ist!

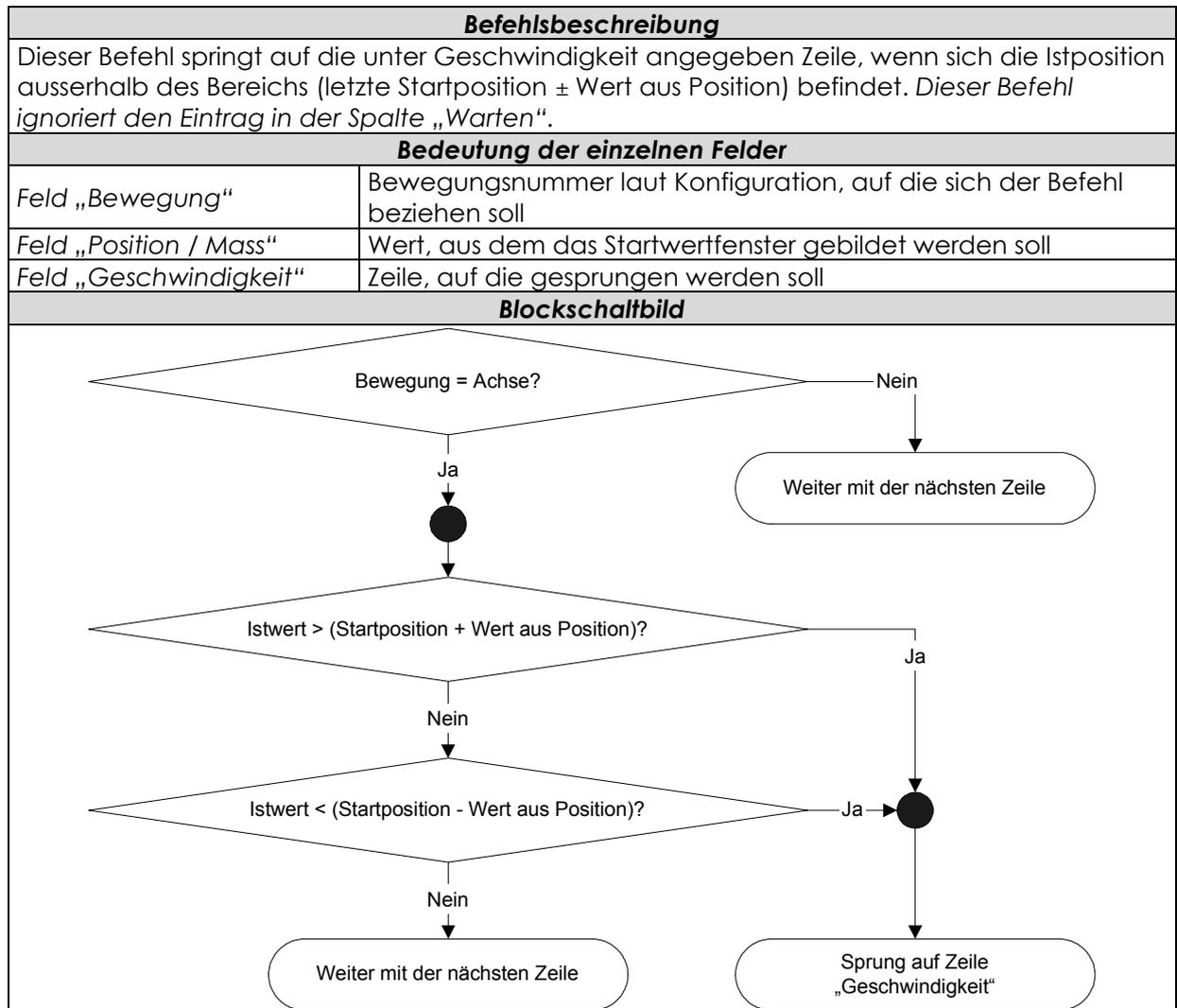
Befehl 36: Sprung wenn innerhalb Startwertfenster Mass

Befehlsbeschreibung	
Dieser Befehl springt auf die unter Geschwindigkeit angegebene Zeile, wenn sich die Istposition innerhalb des Bereichs (letzte Startposition ± Wert aus Position) befindet. <i>Dieser Befehl ignoriert den Eintrag in der Spalte „Warten“.</i>	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Wert, aus dem das Startwertfenster gebildet werden soll
Feld „Geschwindigkeit“	Zeile, auf die gesprungen werden soll
Blockschaltbild	
 <pre> graph TD A{Bewegung = Achse?} -- Ja --> B(()) A -- Nein --> C(Weiter mit der nächsten Zeile) B --> D{Istwert > (Startposition + Wert aus Position)?} D -- Ja --> E(()) D -- Nein --> F{Istwert < (Startposition - Wert aus Position)?} F -- Ja --> E F -- Nein --> G(Sprung auf Zeile „Geschwindigkeit“) E --> H(Weiter mit der nächsten Zeile) </pre>	

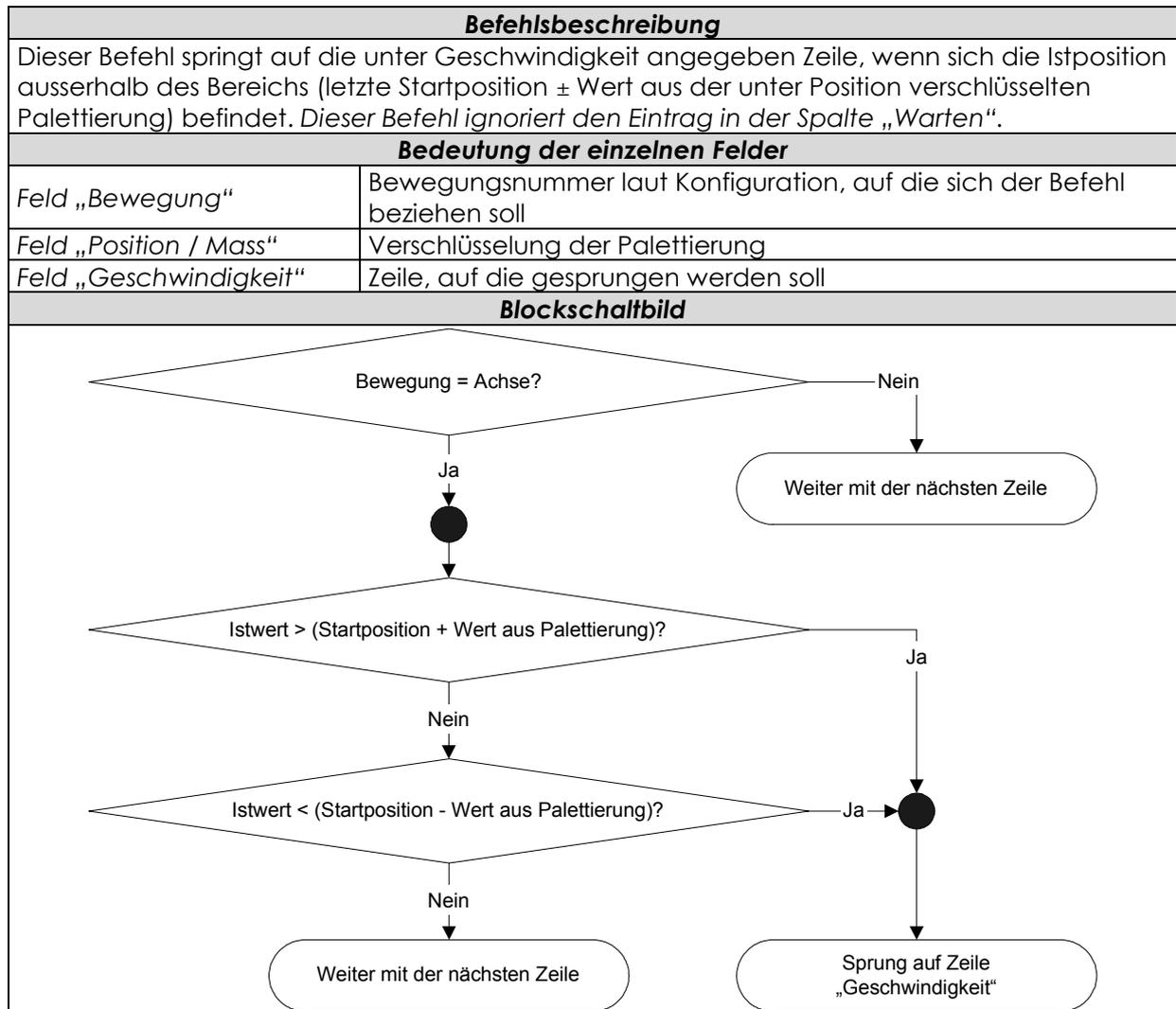
Befehl 37: Sprung wenn innerhalb Startwertfenster Mass aus Palettierung

Befehlsbeschreibung	
Dieser Befehl springt auf die unter Geschwindigkeit angegebene Zeile, wenn sich die Istposition innerhalb des Bereichs (letzte Startposition \pm Wert aus der unter Position verschlüsselten Palettierung) befindet. Dieser Befehl ignoriert den Eintrag in der Spalte „Warten“.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfiguration, auf die sich der Befehl beziehen soll
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Zeile, auf die gesprungen werden soll
Blockschaltbild	
 <pre> graph TD Start(()) --> D1{Bewegung = Achse?} D1 -- Ja --> J1(()) D1 -- Nein --> R1(Weiter mit der nächsten Zeile) J1 --> D2{Istwert > (Startposition + Wert aus Palettierung)?} D2 -- Ja --> J2(()) D2 -- Nein --> D3{Istwert < (Startposition - Wert aus Palettierung)?} D3 -- Ja --> J2 D3 -- Nein --> R2(Sprung auf Zeile „Geschwindigkeit“) J2 --> R3(Weiter mit der nächsten Zeile) </pre>	

Befehl 38: Sprung wenn ausserhalb Startwertfenster Mass



Befehl 39: Sprung wenn ausserhalb Startwertfenster Mass aus Palettierung



Befehl 40: Fahre Slave auf Mass, Rampe gekappt

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Im Prinzip handelt es sich um den Befehl 0, mit dem Unterschied, dass der Befehl nicht ausgeführt wird, sondern als Teil einer Sequenz in den dezentralen Antrieb geladen wird.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Siehe Befehl 0
Feld „Geschwindigkeit“	Siehe Befehl 0

Befehl 41: Fahre Slave auf Mass aus Palettierung, Rampe gekappt

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Im Prinzip handelt es sich um den Befehl 1, mit dem Unterschied, dass der Befehl nicht ausgeführt wird, sondern als Teil einer Sequenz in den dezentralen Antrieb geladen wird.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Siehe Befehl 1
Feld „Geschwindigkeit“	Siehe Befehl 1

Befehl 42: Fahre Slave auf Mass, Rampe komprimiert

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Im Prinzip handelt es sich um den Befehl 2, mit dem Unterschied, dass der Befehl nicht ausgeführt wird, sondern als Teil einer Sequenz in den dezentralen Antrieb geladen wird.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Siehe Befehl 2
Feld „Geschwindigkeit“	Siehe Befehl 2

Befehl 43: Fahre Slave auf Mass aus Palettierung, Rampe komprimiert

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Im Prinzip handelt es sich um den Befehl 3, mit dem Unterschied, dass der Befehl nicht ausgeführt wird, sondern als Teil einer Sequenz in den dezentralen Antrieb geladen wird.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Siehe Befehl 3
Feld „Geschwindigkeit“	Siehe Befehl 3

Befehl 44: Fahre Slave auf Mass, Strom reduziert

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Im Prinzip handelt es sich um den Befehl 24, mit dem Unterschied, dass der Befehl nicht ausgeführt wird, sondern als Teil einer Sequenz in den dezentralen Antrieb geladen wird.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Siehe Befehl 24
Feld „Geschwindigkeit“	Siehe Befehl 24

Befehl 45: Fahre Slave auf Mass aus Palettierung, Strom reduziert

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Im Prinzip handelt es sich um den Befehl 25, mit dem Unterschied, dass der Befehl nicht ausgeführt wird, sondern als Teil einer Sequenz in den dezentralen Antrieb geladen wird.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Siehe Befehl 25
Feld „Geschwindigkeit“	Siehe Befehl 25

Befehl 46: Warte bis Slave-Position < Mass

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle gewartet wird, bis die Istposition des Antriebs kleiner als der im Feld Position angegebene Wert ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Position mit der verglichen werden soll
Feld „Geschwindigkeit“	Keine

Befehl 47: Warte bis Slave-Position < Mass aus Palettierung

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle gewartet wird, bis die Istposition des Antriebs kleiner als der Wert ist, der im Feld Position als Palettierung verschlüsselt ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Keine

Befehl 48: Warte bis Slave-Position > Mass

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle gewartet wird, bis die Istposition des Antriebs größer als der im Feld Position angegebene Wert ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Position mit der verglichen werden soll
Feld „Geschwindigkeit“	Keine

Befehl 49: Warte bis Slave-Position > Mass aus Palettierung

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle gewartet wird, bis die Istposition des Antriebs größer als der Wert ist, der im Feld Position als Palettierung verschlüsselt ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Keine

Befehl 50: Lösche Ausgang Mass im Slave

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle der im Feld Position angegebene interne Ausgang des Antriebs gelöscht wird.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Ausgang, der im Antrieb gelöscht werden soll
Feld „Geschwindigkeit“	Keine

Befehl 51: Lösche Ausgang Mass aus Palettierung im Slave

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle der interne Ausgang des Antriebs, der in der Palettiertabelle, die durch das Feld Position verschlüsselt ist, angegebene ist, gelöscht wird.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Keine

Befehl 52: Setze Ausgang Mass im Slave

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle der im Feld Position angegebene interne Ausgang des Antriebs gesetzt wird.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Ausgang, der im Antrieb gesetzt werden soll
Feld „Geschwindigkeit“	Keine

Befehl 53: Setze Ausgang Mass aus Palettierung im Slave

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle der interne Ausgang des Antriebs, der in der Palettiertabelle, die durch das Feld Position verschlüsselt ist, angegebene ist, gesetzt wird.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Keine

Befehl 54: Warte bis Eingang im Slave low ist

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle gewartet wird, bis der interne Eingang des Antriebs, der im Feld Position genannt ist, low ist..	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Eingang der kontrolliert werden soll
Feld „Geschwindigkeit“	Keine

Befehl 55: Warte bis Eingang aus Palettierung im Slave low ist

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle gewartet wird, bis der interne Eingang des Antriebs, der über das Feld Position in der Palettiertabelle genannt ist, low ist..	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Verschlüsselung der Palettiertabelle
Feld „Geschwindigkeit“	Keine

Befehl 56: Warte bis Eingang im Slave high ist

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle gewartet wird, bis der interne Eingang des Antriebs, der im Feld Position genannt ist, high ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Eingang der kontrolliert werden soll
Feld „Geschwindigkeit“	Keine

Befehl 57: Warte bis Eingang aus Palettierung im Slave high ist

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle gewartet wird, bis der interne Eingang des Antriebs, der über das Feld Position in der Palettiertabelle genannt ist, high ist..	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Verschlüsselung der Palettiertabelle
Feld „Geschwindigkeit“	Keine

Befehl 58: Warte Mass Millisekunden

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle gewartet wird, bis soviel Millisekunden vergangen sind, wie im Feld Position stehen.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Millisekunden, die gewartet werden soll
Feld „Geschwindigkeit“	Keine

Befehl 59: Warte Mass aus Palettierung Millisekunden

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass bei der Ausführung der Sequenz an dieser Stelle gewartet wird, bis soviel Millisekunden vergangen sind, wie im in der über das Feld Position angesprochenen Palettiertabelle stehen.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Keine

Befehl 60: Lösche das Programm im Slave

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Dieser Befehl führt dazu, dass das im Slave hinterlegte Programm vollständig gelöscht wird.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 61: Reset des Programms im Slave

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Dieser Befehl führt dazu, dass ein eventuell laufende Sequenz im Slave abgebrochen wird und die aktuelle Zeile der Sequenz im Antrieb auf 1 gesetzt wird. Die Ausführung wird mit diesem Befehl nicht wieder gestartet!	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 62: Starte das Programm im Slave

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Der Befehl führt dazu, dass die Ausführung der Sequenz im Antrieb gestartet wird. Gestartet wird auf der Zeile, auf der die Sequenz aktuell steht. Ist dieser Befehl mit Warten Ja in der Teachebene eingetragen, so wartet die Teachebene bis der Antrieb die Sequenz vollständig abgearbeitet hat. Wenn der Antrieb meldet, dass die Sequenz mit einem Fehler oder einer Störung beendet worden ist, dann springt dieser Befehl auf die Zeile, die im Feld Geschwindigkeit angegeben ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Zeile, auf die gesprungen wird, wenn innerhalb der Sequenz ein Fehler oder eine Störung aufgetreten ist.

Befehl 63: Warte auf das Programm im Slave

Befehlsbeschreibung	
Hierbei handelt es sich um einen Befehl zur Sequenzprogrammierung. Dieser Befehl veranlasst die Teachebene darauf zu warten, dass der Antrieb die Sequenz vollständig abgearbeitet hat. Er ist nur sinnvoll, wenn die Sequenz zuvor mit dem Befehl 62 und Warten Nein gestartet worden ist. Dieser Befehl startet die Sequenz nicht! Wenn der Antrieb meldet, dass die Sequenz mit einem Fehler oder einer Störung beendet worden ist, dann springt dieser Befehl auf die Zeile, die im Feld Geschwindigkeit angegeben ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer der Achse, auf die die Sequenz geladen werden soll.
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Zeile, auf die gesprungen wird, wenn innerhalb der Sequenz ein

	Fehler oder eine Störung aufgetreten ist.
--	---

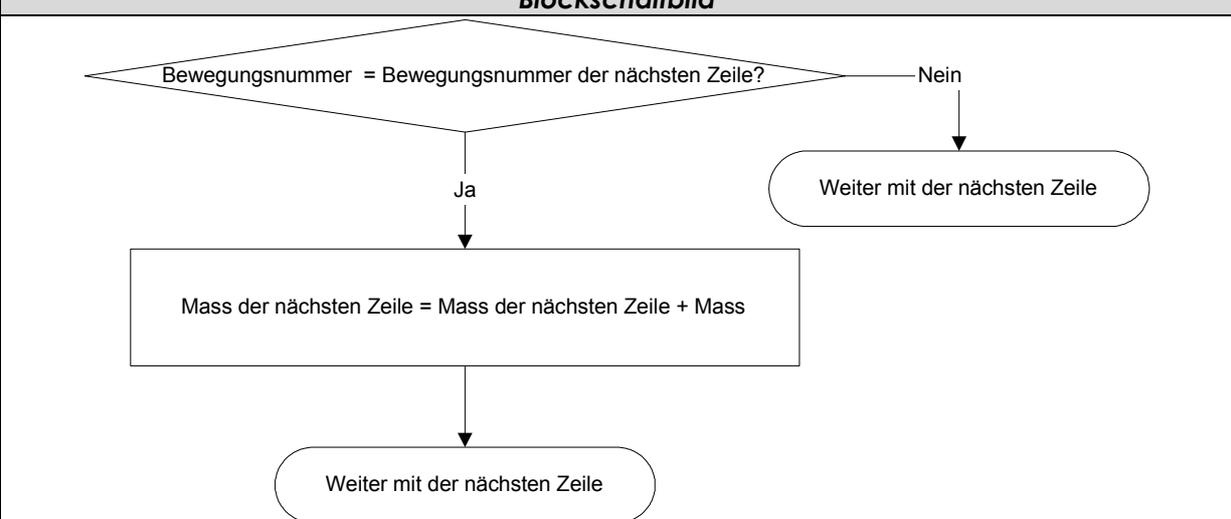
Befehl 64: Kurve einkuppeln

Befehlsbeschreibung	
Dieser Befehl kuppelt die Kurve eines SEW-Antriebs ein. Die Kurve bleibt eingekuppelt, bis sie mit dem entsprechenden Befehl wieder ausgekuppelt wird. Solange die Achse eingekuppelt ist, reagiert diese auf keinen anderen Befehl!	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfig, bei der der Befehl wirksam werden soll. Es muss sich dabei um einen SEW-Antrieb handeln.
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

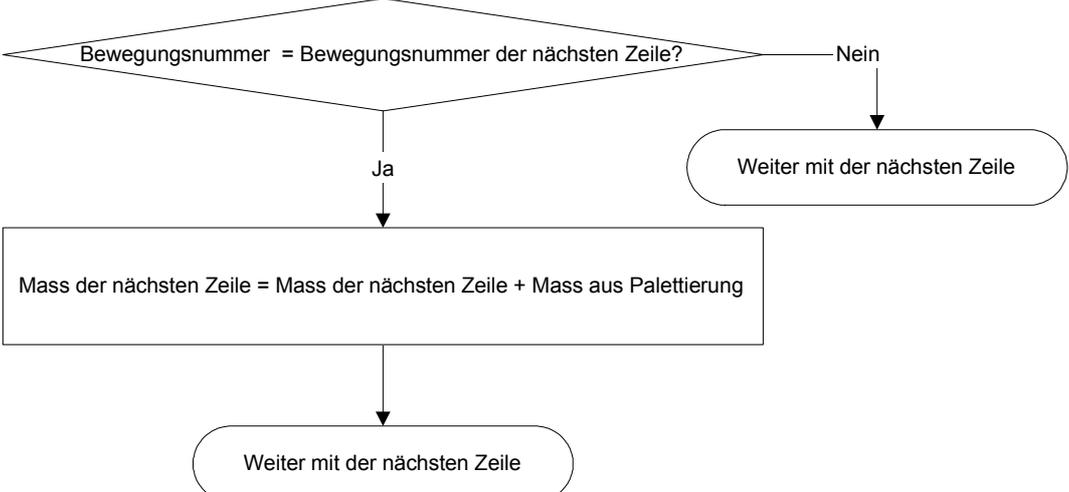
Befehl 65: Kurve auskuppeln

Befehlsbeschreibung	
Dieser Befehl kuppelt die Kurve eines SEW-Antriebs aus.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer laut Konfig, bei der der Befehl wirksam werden soll. Es muss sich dabei um einen SEW-Antrieb handeln.
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

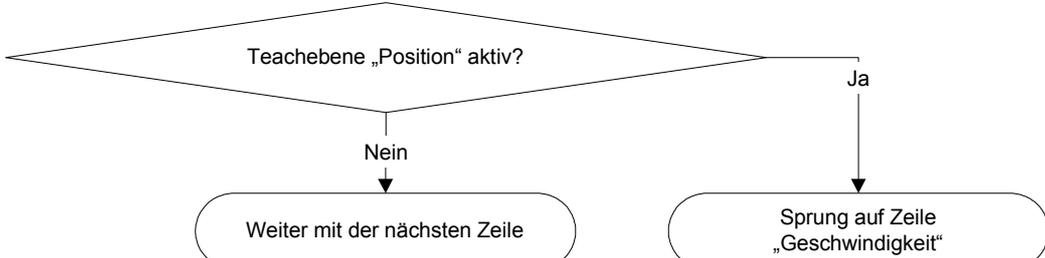
Befehl 68: Addiere Mass auf den nachfolgenden Befehl

Befehlsbeschreibung	
Ist die Bewegungsnummer des Befehls gleich der Bewegungsnummer der nächsten Zeile, so wird das Mass des Befehls auf das Mass der nächsten Zeile aufaddiert. Ist die Bewegungsnummer ungleich, so ist der Befehl wirkungslos.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer für die der Befehl gelten soll
Feld „Position / Mass“	Mass, dass addiert werden soll
Feld „Geschwindigkeit“	Keine Bedeutung
Blockschaltbild	
 <pre> graph TD A{Bewegungsnummer = Bewegungsnummer der nächsten Zeile?} -- Ja --> B[Mass der nächsten Zeile = Mass der nächsten Zeile + Mass] A -- Nein --> C(Weiter mit der nächsten Zeile) B --> D(Weiter mit der nächsten Zeile) </pre>	

Befehl 69: Addiere Mass aus Palettierung auf den nachfolgenden Befehl

Befehlsbeschreibung	
Ist die Bewegungsnummer des Befehls gleich der Bewegungsnummer der nächsten Zeile, so wird das Mass, verschlüsselt durch die Palettierstabelle auf das Mass der nächsten Zeile aufaddiert. Ist die Bewegungsnummer ungleich, so ist der Befehl wirkungslos.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Bewegungsnummer für die der Befehl gelten soll
Feld „Position / Mass“	Verschlüsselung der Palettierung
Feld „Geschwindigkeit“	Keine Bedeutung
Blockschaltbild	
 <pre> graph TD A{Bewegungsnummer = Bewegungsnummer der nächsten Zeile?} -- Ja --> B[Mass der nächsten Zeile = Mass der nächsten Zeile + Mass aus Palettierung] A -- Nein --> C(Weiter mit der nächsten Zeile) B --> D(Weiter mit der nächsten Zeile) </pre>	

Befehl 71: Teachebene aktiv?

Befehlsbeschreibung	
Dieser Befehl springt auf die unter Geschwindigkeit angegebene Zeile, wenn die unter Position angegebene Teachebene aktiv ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Nummer der Teachebene, die getestet werden soll
Feld „Geschwindigkeit“	Zeile, auf die gesprungen werden soll
Blockschaltbild	
 <pre> graph TD A{Teachebene „Position“ aktiv?} -- Ja --> B(Sprung auf Zeile „Geschwindigkeit“) A -- Nein --> C(Weiter mit der nächsten Zeile) </pre>	

Befehl 72: Teilenummer minus 1

Befehlsbeschreibung	
Dieser Befehl verringert die Teilenummer in der Palettiertabelle, die im Feld Position verschlüsselt ist, um den Wert 1 und macht anschließend die verschlüsselte Palettiertabelle zur Default-Palettiertabelle.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Verschlüsselung der Palettiertabelle, wobei die Art der Tabelle und die Spalte in der Tabelle keine Bedeutung haben. Es werden nur die 10000er und die 1000er Stelle der Verschlüsselung verwendet.
Feld „Geschwindigkeit“	Keine

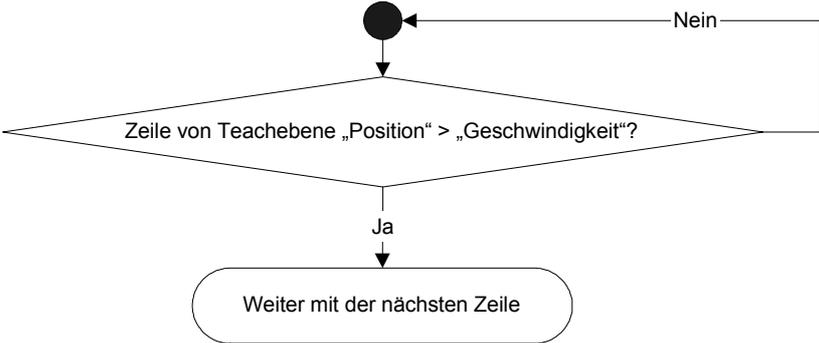
Befehl 73: Teachebene starten

Befehlsbeschreibung	
Dieser Befehl startet die Teachebene, die im Feld Position angegeben ist auf der Zeile, die im Feld Geschwindigkeit angegeben ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Nummer der Teachebene, die gestartet werden soll
Feld „Geschwindigkeit“	Zeile auf der die gewünschte Teachebene gestartet werden soll

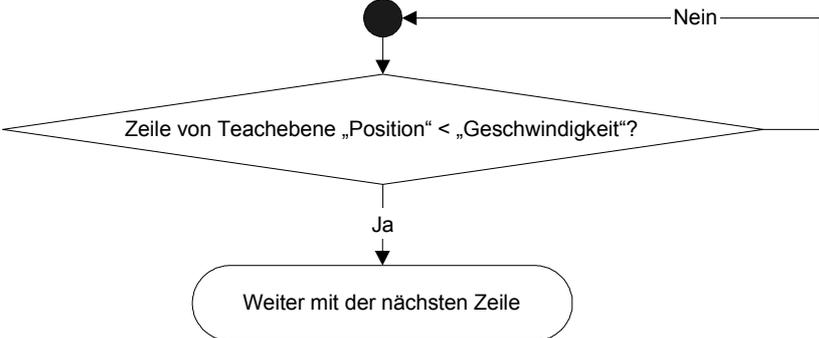
Befehl 74: Teachebene stoppen

Befehlsbeschreibung	
Dieser Befehl hält die Ausführung der im Feld Position angegebenen Teachebene an.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Nummer der Teachebene, die angehalten werden soll
Feld „Geschwindigkeit“	Keine

Befehl 75: Warte bis Zeile >

Befehlsbeschreibung	
Dieser Befehl wartet, bis die auszuführende Zeilennummer der Teachebene, die unter dem Feld Position angegeben ist, größer ist als der Wert, der unter Geschwindigkeit angegeben ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Nummer der Teachebene, auf die gewartet werden soll
Feld „Geschwindigkeit“	Nummer der Zeile, auf die gewartet werden soll
Blockschaltbild	
 <pre> graph TD Start(()) --> Decision{Zeile von Teachebene „Position“ > „Geschwindigkeit“?} Decision -- Ja --> End([Weiter mit der nächsten Zeile]) Decision -- Nein --> Start </pre>	

Befehl 76: Warte bis Zeile <

Befehlsbeschreibung	
Dieser Befehl wartet, bis die auszuführende Zeilennummer der Teachebene, die unter dem Feld Position angegeben ist, kleiner ist als der Wert, der unter Geschwindigkeit angegeben ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Nummer der Teachebene, auf die gewartet werden soll
Feld „Geschwindigkeit“	Nummer der Zeile, auf die gewartet werden soll
Blockschaltbild	
 <pre> graph TD Start(()) --> Decision{Zeile von Teachebene „Position“ < „Geschwindigkeit“?} Decision -- Ja --> End([Weiter mit der nächsten Zeile]) Decision -- Nein --> Start </pre>	

Befehl 77: Teilenummer löschen

Befehlsbeschreibung	
Dieser Befehl löscht die Teilenummer in der Palettiertabelle, die im Feld Position verschlüsselt ist und macht anschließend die verschlüsselte Palettiertabelle zur Default-Palettiertabelle.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Verschlüsselung der Palettiertabelle, wobei die Art der Tabelle und die Spalte in der Tabelle keine Bedeutung haben. Es werden nur die 10000er und die 1000er Stelle der Verschlüsselung verwendet.
Feld „Geschwindigkeit“	Keine

Befehl 78: Teilenummer plus 1

Befehlsbeschreibung	
Dieser Befehl erhöht die Teilenummer in der Palettiertabelle, die im Feld Position verschlüsselt ist, um den Wert 1 und macht anschließend die verschlüsselte Palettiertabelle zur Default-Palettiertabelle.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Verschlüsselung der Palettiertabelle, wobei die Art der Tabelle und die Spalte in der Tabelle keine Bedeutung haben. Es werden nur die 10000er und die 1000er Stelle der Verschlüsselung verwendet.
Feld „Geschwindigkeit“	Keine

Befehl 79: Tabelle umschalten

Befehlsbeschreibung	
Dieser Befehl schaltet die Default-Palettiertabelle um und setzt diese auf die Tabelle, die im Feld Position verschlüsselt ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Verschlüsselung der Palettiertabelle, wobei die Art der Tabelle und die Spalte in der Tabelle keine Bedeutung haben. Es werden nur die 10000er und die 1000er Stelle der Verschlüsselung verwendet.
Feld „Geschwindigkeit“	Keine

Befehl 80: Bahn umrechnen

Befehlsbeschreibung	
Dieser Befehl berechnet aus einer Kurvendatei die entsprechenden Bahnpunktdateien für die beteiligten Achsen. Als Kurvendatei wird die Datei verwendet, die zu dem aktuellen Programm und der Gruppe der Teach-Ebene gehört. Als Ergebnis der Umrechnung entstehen für die beteiligten Achsen die entsprechenden Bahnpunktdateien, die dann mit dem Befehl 30 in die Achsen geladen werden können.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine, das Teach-In wartet an dieser Stelle immer, bis die Umrechnung vollständig gelaufen ist.

Befehl 81: Sprung wenn 2. Teachebene aktiv

Befehlsbeschreibung	
Wird dieser Befehl in der Teachebene 1 gegeben, so springt die Teachebene 1 auf die unter Geschwindigkeit angegebene Zeile, wenn die Teachebene 2 aktiv ist. Wird dieser Befehl in einer anderen Teachebene gegeben, so springt diese Teachebene auf die unter Geschwindigkeit angegebene Zeile, wenn die Teachebene 1 aktiv ist.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Zeile, auf die gesprungen werden soll

Befehl 82: 2. Teachebene starten, Stop passiv

Befehlsbeschreibung	
Wird dieser Befehl in der Teachebene 1 gegeben, so wird die Teachebene 2 angestartet. Wird dieser Befehl in einer anderen Teachebene gegeben, so wird die Teachebene 1 gestartet. In beiden Fällen wird bei der an zu startenden Teachebene der Modus „Stop passiv“ aktiviert.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 83: Startbedingung löschen

Befehlsbeschreibung	
Dieser Befehl löscht die Startbedingung der Gruppe, zu der die Teachebene gehört. Damit wird die Ausführung der eigenen Teachebene angehalten.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 84: Ablaufprogrammierung löschen

Befehlsbeschreibung	
Dieser Befehl führt einen Reset auf der gesamten Ablaufprogrammierung durch.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 87: Stop-Abfrage aktiv

Befehlsbeschreibung	
Dieser Befehl setzt die Stop-Abfrage auf aktiv. Stop-Befehle (entweder durch die Stop-Taste am Terminal oder durch einen externen Stop) werden wieder ausgeführt.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 88: Stop-Abfrage passiv

Befehlsbeschreibung	
Dieser Befehl setzt die Stop-Abfrage auf passiv. Stop-Befehle (entweder durch die Stop-Taste am Terminal oder durch einen externen Stop) werden ignoriert. Es ist unbedingt darauf zu achten, dass die Stop-Abfrage wieder auf aktiv gesetzt wird, da die Maschine sonst nicht angehalten werden kann.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 89: Unterbrechung nach Zeile

Befehlsbeschreibung	
Dieser Befehl armiert einen Sprung für eine Unterbrechung. Tritt eine Unterbrechung ein, so springt die Teacheben auf die angegebene Zeile	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Zeile, auf die bei Unterbrechung gesprungen werden soll

Befehl 90: Stop-Uhr starten

Befehlsbeschreibung	
Dieser Befehl startet die interne Stop-Uhr und zeigt sie ständig aktualisiert an.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 91: Stop-Uhr Restart

Befehlsbeschreibung	
Dieser Befehl zeigt den aktuellen Wert der Stop-Uhr an, setzt die Uhr dann zurück und startet sie erneut. In der Anzeige bleibt der „alte“ Wert stehen.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 92: Stop-Uhr anzeigen

Befehlsbeschreibung	
Dieser Befehl zeigt den aktuellen Wert der Stop-Uhr an. Die Uhr läuft im Hintergrund weiter und im Bild wird sie nicht mehr aktualisiert.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 93: 2. Teachebene starten

Befehlsbeschreibung	
Wird dieser Befehl in der Teachebene 1 gegeben, so wird die Teachebene 2 angestartet. Wird dieser Befehl in einer anderen Teachebene gegeben, so wird die Teachebene 1 gestartet.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 94: 2. Teachebene stoppen

Befehlsbeschreibung	
Wird dieser Befehl in der Teachebene 1 gegeben, so wird die Teachebene 2 angehalten. Wird dieser Befehl in einer anderen Teachebene gegeben, so wird die Teachebene 1 angehalten.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 95: Warte bis Zeile >

Befehlsbeschreibung	
Wird dieser Befehl in der Teachebene 1 gegeben, so wartet die Teachebene 1, bis die Teachebene 2 auf einer Zeile größer dem Wert im Feld Geschwindigkeit steht. Wird dieser Befehl in einer anderen Teachebene gegeben, so wartet diese Teachebene, bis die Teachebene 1 auf einer Zeile größer dem Wert im Feld Geschwindigkeit steht.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Nummer der Zeile, auf die gewartet werden soll

Befehl 96: Warte bis Zeile <

Befehlsbeschreibung	
Wird dieser Befehl in der Teachebene 1 gegeben, so wartet die Teachebene 1, bis die Teachebene 2 auf einer Zeile kleiner dem Wert im Feld Geschwindigkeit steht. Wird dieser Befehl in einer anderen Teachebene gegeben, so wartet diese Teachebene, bis die Teachebene 1 auf einer Zeile kleiner dem Wert im Feld Geschwindigkeit steht.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Nummer der Zeile, auf die gewartet werden soll

Befehl 97: Teilenummer löschen

Befehlsbeschreibung	
Diese Befehl löscht die Teilenummer in der Default-Palettirtabelle der Teachebene.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 98: Teilenummer plus 1

Befehlsbeschreibung	
Dieser Befehl erhöht die Teilenummer in der Default-Palettirtabelle der Teachebene um den Wert 1.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Keine

Befehl 99: Sprung nach Zeile

Befehlsbeschreibung	
Diese Befehl führt einen unbedingten Sprung auf die unter „Geschwindigkeit“ angegebene Zeile durch.	
Bedeutung der einzelnen Felder	
Feld „Bewegung“	Keine
Feld „Position / Mass“	Keine
Feld „Geschwindigkeit“	Zeile, auf die gesprungen werden soll

Programmverwaltung

Wie bereits im Kapitel Grundlagen der Teach-Programmierung beschrieben, kann es innerhalb des Systems verschiedene Programme geben. Diese Programme können die Nummern 1 bis 99 tragen. Gearbeitet werden kann stets nur mit einem Programm; dieses Programm wird aktuelles Programm genannt. Das aktuelle Programm kann in einer entsprechenden Maske festgelegt werden und wird zusammen mit den Konstanten gespeichert. Innerhalb dieser Maske stehen auch Funktionen zur Verwaltung der Programme zur Verfügung. So können Programme beispielsweise gelöscht oder kopiert werden. Die Programmverwaltung achtet dabei darauf, dass alle Dateien, die zu einem Programm gehören, gelöscht bzw. kopiert werden.

Nachfolgend wird die Maske zur Eingabe der aktuellen Programmnummer vorgestellt:



The image shows a graphical user interface for setting the current program. At the top, a grey bar contains the text 'AKTUELLES PROGRAMM FESTLEGEN'. Below this, a label 'PROGRAMMNUMMER:' is followed by a black input field containing the number '2'. At the bottom of the screen, there are four dark grey buttons with white text: 'ZURUECK', 'INHALT', and two empty buttons.

Abbildung 12: Aktuelles Programm festlegen

In der Maske zum Festlegen der aktuellen Programmnummer, kann die gewünschte Nummer direkt eingegeben werden. Beim Aufrufen der Maske beinhaltet das Feld die aktuell eingestellte Programmnummer. Der Softkey Inhalt führt zu einer Übersicht, die alle Programme, die sich auf der Ramdisk befinden, beinhaltet.

Gruppenzuordnung der Teachebenen

Die einzelnen Teachebenen können einer Gruppe zugeordnet werden. Dabei ist es durchaus möglich, dass mehrere Teachebenen zu einer Gruppe gehören. Die Gruppenzuordnung der einzelnen Teachebenen kann in einer Maske eingegeben werden:

ZUORDNUNGSTABELLE TEACHIN

NUMMER	GRUPPE
1	2
2	3
3	4
4	1
5	0
6	0
7	0
8	0
9	0

SEITE-

SEITE+

Abbildung 14: Gruppenzuordnung der Teachebenen

Wenn eine Teachebene einer Gruppe zugeordnet ist, so sind für diese Teachebene die Start- und Stopzustände und –merker der jeweiligen Gruppendefinition relevant.

Handfunktionen

Das Betriebssystem beinhaltet eine Maske für Handfunktionen. Dabei handelt es sich um ein Bild, welches dem oberen Teil der zweiteiligen Arbeitsmaske (siehe entsprechendes Kapitel) im Layout und in der Funktionalität gleicht. Die Handfunktionen sind dafür gedacht, einzelne Bewegungen innerhalb einer Maschine manuell ausführen zu können. Für die Handfunktionen wird eine Konfigurationsdatei benötigt, die den gleichen Aufbau wie die Konfiguration der Teach-Programmierung hat. Diese Datei trägt den Dateinamen HANDB+Programmnummer+.CFG, also beispielsweise HANDB02.CFG für das Programm 2.

Die nachfolgende Abbildung zeigt ein Beispiel für die Handfunktionen:

HANDFUNKTIONEN			
ISTWERT:	Sollwert:		
HUB LIFT 2	HUB LIFT 2	BAND LIFT 2	
AUTOMATIK LIFT 2	ETAGENBÄNDER	ETAGENBÄNDER	
----	----	----	
----	----	----	
----	LICHTGITTER	VERWALTUNG	
SYSTEMDRUCK ZU NIEDRIG			
POS. BAND 1	POS. BAND 2	POS. BAND 3	POS. BAND 4

Abbildung 15: Handfunktionen

Im Bild für die Handfunktionen sind die einzelnen Bewegungen aufgeführt, wie sie in der Konfiguration definiert worden sind. In diesem Bild können die Bewegungen mittels Softkeys ausgeführt werden. Dabei beziehen sich die Softkeys immer auf die Bewegung, die aktuell durch den Cursor markiert sind. Handelt es sich bei der Bewegung um Achsen, so ist die Belegung der Softkeys immer gleich:

Softkey	Bedeutung
Tippen -	Bewegt die Achse in negativer Fahrtrichtung. Die Bewegung erfolgt, solange der Softkey gedrückt ist, dabei wird die Bewegung langsam schneller. Wird der Softkey losgelassen, wird die Bewegung gestoppt.
Tippen +	Bewegt die Achse in positiver Fahrtrichtung. Die Bewegung erfolgt, solange der Softkey gedrückt ist, dabei wird die Bewegung langsam schneller. Wird der Softkey losgelassen, wird die Bewegung gestoppt.
1/10 Minus	Fährt die Achse – ausgehend von der aktuellen Istposition – bei jedem Drücken um 1/10 Millimeter in negativer Richtung
1/10 Plus	Fährt die Achse – ausgehend von der aktuellen Istposition – bei jedem Drücken um 1/10 Millimeter in positiver Richtung

Handelt es sich bei der markierten Bewegung um einen Ablauf, so werden auf den Softkeys die Texte dargestellt, die in der Konfiguration für den entsprechenden Softkey definiert worden sind. Wird ein Softkey gedrückt, so wird dem Ablauf der Auftrag gesendet, der in der Konfiguration für diesen Softkey vorgesehen ist.

Teilenummern editieren

Jede Palettiertabelle hat einen eigenen Teilezähler. Dieser Teilezähler besteht aus dem Etagezähler und dem Zähler für die Teilenummer. Beide Werte können in einer entsprechenden Maske manuell verändert werden:

TABELLE	ETAGE	TEILENUMMER
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	1
12	1	1
13	1	1
14	1	1
15	1	1
16	1	1
17	1	1

Abbildung 16: Teilenummern editieren

Onlinehilfe der Teach-Programmierung

Die Teach-Programmierung verfügt über ein Bild, in dem alle verfügbaren Opcodes mit ihrer Bezeichnung aufgeführt sind. Dieses Bild kann als Onlinehilfe verwendet werden. Die entsprechende Opcodetabelle kann mit dem Softkey Drucken in eine Textdatei oder in eine Excel-lesbare Datei ausgegeben werden.

HILFE: TEACHIN-BEFEHLE	
LISTE DER TEACHIN-BEFEHLE	
NR.	BEZEICHNUNG
0	FAHREN AUF MASS MIT GEKAPPTER KURVE
1	FAHREN AUF (MASS) MIT GEKAPPTER KURVE
2	FAHREN AUF MASS MIT KOMPRIMIERTER KURVE
3	FAHREN AUF (MASS) MIT KOMPRIMIERTER KURVE
4	STARTEN SYNCHRONFAHRT AB DEM PUNKT MASS
5	STARTEN SYNCHRONFAHRT AB DEM PUNKT (MASS)
6	SPRUNG WENN ISTWERT < MASS NACH GESCHW.
7	SPRUNG WENN ISTWERT < (MASS) NACH GESCHW.
8	SPRUNG WENN ISTWERT > MASS NACH GESCHW.
9	SPRUNG WENN ISTWERT > (MASS) NACH GESCHW.
10	SPRUNG WENN ISTWERT = MASS NACH GESCHW.
11	SPRUNG WENN ISTWERT = (MASS) NACH GESCHW.
12	SPRUNG WENN ISTWERT <> MASS NACH GESCHW.
13	SPRUNG WENN ISTWERT <> (MASS) NACH GESCHW.
14	FAHRE AUF ERSTEN PUNKT VON SEGMENT MASS
15	FAHRE AUF ERSTEN PUNKT VON SEGMENT (MASS)
16	STARTE BAHNSEGMENT MASS
17	STARTE BAHNSEGMENT (MASS)

<< **DRUCKEN** **SEITE-** **SEITE+**

Abbildung 17: Onlinehilfe

Sicherheitsräume

Ein Sicherheitsraum stellt einen Block von maximal vier Bewegungen dar, mit dem eine Position im Raum beschrieben werden kann. Bei den Bewegungen kann es sich um eine Achse oder um einen Ablauf handeln. Für jede einzelne Bewegung wird festgelegt, von welchem Mass bis zu welchem Mass die Bedingung gelten soll. Handelt es sich bei der Bewegung um einen Ablauf, so gilt das Mass als der Step, den der entsprechende Ablauf aktuell meldet. Ein Sicherheitsraum gilt als verletzt, wenn alle Bewegungen die beschriebene Bedingung erfüllen. Erfüllt nur eine der genannten Bewegungen die Bedingung nicht, so ist der Sicherheitsraum nicht verletzt. Melden Abläufe einen Step Null, so gilt die entsprechende Bedingung als erfüllt.

Die Sicherheitsräume werden vom System automatisch überprüft. Ist ein Sicherheitsraum verletzt, so wird die Störung „Sicherheitsraum verletzt“ ausgelöst und die Maschine angehalten. Die Überprüfung findet einerseits vor Auslösung einer Fahrbewegung statt, andererseits aber auch während der Ausführung der Bewegung. Steht bereits vor Auslösung der Fahrbewegung fest, dass der Sicherheitsraum verletzt sein wird, so wird die Bewegung gar nicht erst ausgelöst.

Neben den Sicherheitsräumen gibt es zusätzlich noch die SPS-Sicherheitsräume, die aber nicht automatisch überprüft werden sondern lediglich von der SPS abgefragt werden können. Die SPS-Sicherheitsräume sind im Handbuch Ablaufprogrammierung dokumentiert.



Die Datei mit den Sicherheitsräumen ist eine rekordbasierte Datei. Wird ein Datensatz eingefügt oder gelöscht, so verschieben sich alle dahinter liegenden Sicherheitsräume und es ist eventuell notwendig, die abfragenden Prozesse entsprechend zu ändern!

Sicherheitsräume können mit einem Text versehen werden. Der Textschlüssel für Sicherheitsräume lautet TISICH.....+Nummer, also beispielsweise TISICH.....1 für den Sicherheitsraum 1. In dem entsprechenden Editor können die Sicherheitsräume in eine Textdatei ausgegeben werden (Softkey „Drucken“).

STANDARD-SICHERHEITSRAUME

SICHERHEITSRAEUME BEARBEITEN

LAUFENDE NUMMER 1

UEBERWACHUNG SGM

BEWEGUNG	GRP	BEWEGUNGSTEXT	VON MASS	BIS MASS
1	1	X-ACHSE	250+	300+
2	1	Y-ACHSE	700+	900+
3	1	Z-ACHSE	1525+	2000+
0	0		0+	0+

<<
DRUCKEN
SATZ-1
SATZ+1

Abbildung 18: Sicherheitsräume

Der Sicherheitsraum 1 aus dem oben gezeigten Beispiel gilt als verletzt, wenn die X-Achse auf einem Mass zwischen 250 und 300, die Y-Achse auf einem Mass zwischen 700 und 900 und die Z-Achse auf einem Mass zwischen 1525 und 2000 steht.

Systemmasse

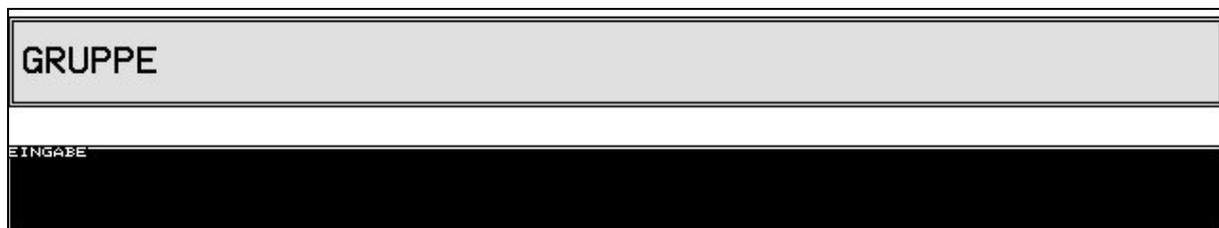
Die Systemmasse sind prinzipiell eine Palettiertabelle, die im System fest hinterlegt ist. Diese Tabelle ist grundsätzlich programm- und gruppenabhängig und beinhaltet pro Datensatz einen Eintrag aus den ersten sechs Bewegungen der zugehörigen Konfigurationsdatei. Der Dateiname setzt sich zusammen aus **SYSMA + Programmnummer(zweistellig) + Gruppennummer + POS**, also beispielsweise **SYSMA991.POS** für die Gruppe 1 des Programms 99. Auch in Systemen, die keine Gruppen benutzen, ist die Systemmasse-Tabelle gruppenabhängig und trägt dann die Gruppennummer Null.

Jede Systemmasse-Tabelle kann maximal 97 Positionen beinhalten, wobei jede Position mit einem Text belegt werden kann. Der Textschlüssel der einzelnen Positionen setzt sich zusammen aus **SYSPOS + Gruppennummer + + Positionsnummer**, also beispielsweise **SYSPOS1. . . .45** für die Position 45 innerhalb der Gruppe 1.

Innerhalb der Teachprogrammierung werden die Systemmasse durch einen Palettierungsbefehl angesprochen. Dabei entfallen für die Verschlüsselung der Palettierung die Tabellenummer und die Art der Tabelle. Ein Palettierungsbefehl, der mit dem Operanden 45 angesprochen wird, bezieht sich also auf die Systemmasse-Tabelle der Gruppe, die zu der ausführenden Teachebene gehört und innerhalb dieser Tabelle auf den Datensatz (die Position) 45.

Neue Systemmasse-Tabelle anlegen

Zum Anlegen einer neuen Systemmasse-Tabelle gibt es eine eigene Funktion. Diese Funktion erfordert die Eingabe der Gruppennummer, für die die Systemmasse-Tabelle angelegt werden soll.



The image shows a screenshot of a software interface. At the top, there is a grey header bar with the word "GRUPPE" in white capital letters. Below this header is a large black rectangular area. In the top-left corner of this black area, the word "EINGABE" is written in small white capital letters, indicating an input field for the group number.

Abbildung 19: Neue Systemmasse-Tabelle erstellen

Wenn für die eingegebene Gruppennummer bereits eine Systemmasse-Tabelle besteht, erscheint eine Fehlermeldung. Im anderen Fall werden die entsprechende Tabelle und die zugehörige Sicherheitsraumdatei erzeugt.

Systemmasse teachen und editieren

Zum Anlegen der Systemmasse steht ein Bild zur Verfügung, welches im oberen Teil das bekannte Teachbild und im unteren Teil die Systemmasse-Tabelle beinhaltet. Im oberen Bildteil können wie gewohnt die verschiedenen Bewegungen ausgeführt werden und die Anlage kann auf die gewünschte Position platziert werden.

Im unteren Bildteil sind die einzelnen Position sichtbar und können an dieser Stelle editiert werden. Dies gilt auch für die Betextung der einzelnen Positionen. Mit Hilfe des Softkeys Übernehmen werden die Istpositionen der ersten sechs Bewegungen aus der Konfigdatei in die Zeile der Tabelle eingetragen, auf der aktuell der Cursor steht.

Sowohl beim manuelle Editieren, als auch beim Übernehmen wird kontrolliert, ob die neuen Werte die Sicherheitsraumbeschreibung der Systemmasse verletzen. Ist dies der Fall, werden die Werte nicht übernommen und es erscheint eine Fehlermeldung.

ISTWERT:		Sollwert:			1/	1/	1
X-ACHSE GRUPPE 2		Y-ACHSE GRUPPE 2			Z-ACHSE GRUPPE 2		
ENTNAHMEHUB		EINLEGEHUB			C-ACHSE		
WERKZEUG		VAK.ROHT.EIN			VAK.ROHT.AUS		
SGM <-> HG		VAK.FERTIGT.EIN			VAK.FERTIGT.AUS		
BLECHAUFNAHME		TÜR ANWAHL			VERWALTUNG		
POSITION		BEW.1	BEW.2	BEW.3	BEW.4	BEW.5	BEW.6
1	SYSP0S2.....1	0.0+	0.0+	0.0+	0.0+	0.0+	0.0+
2	SYSP0S2.....2	0.0+	0.0+	0.0+	0.0+	0.0+	0.0+
3	SYSP0S2.....3	0.0+	0.0+	0.0+	0.0+	0.0+	0.0+
4	SYSP0S2.....4	0.0+	0.0+	0.0+	0.0+	0.0+	0.0+
5	SYSP0S2.....5	0.5+	0.0+	0.0+	0.0+	0.0+	0.0+
SYSTEMDRUCK ZU NIEDRIG							
<<		UEBERNAHME		SEITE-		SEITE+	

Abbildung 20: Systemmasse teachen und editieren

Die Funktionen zum Löschen und Einfügen von Datensätzen befinden sich in der zweiten Softkeyreihe. Solange diese Softkeyreihe sichtbar ist, ist es nicht möglich, in den oberen Bildteil zu wechseln.

Sicherheitsräume der Systemmasse

Zu jeder Systemmasse-Tabelle gehört eine entsprechende Sicherheitsraumdatei, die den Dateinamen der Systemmasse-Tabelle mit der Endung SAS trägt. Das Funktionsprinzip entspricht den normalen Sicherheitsräumen, allerdings wird die Sicherheitsraumdatei der Systemmasse nur verwendet, wenn die Tabelle geteacht oder manuell editiert wird. Sie dient also lediglich dazu, abzusichern, dass keine unzulässigen Positionen innerhalb der Systemmasse-Tabelle aufgeführt werden.

Der erste Sicherheitsraum dieser Datei bezieht sich auf die erste Zeile der Systemmasse-Tabelle, der zweite Sicherheitsraum bezieht sich auf die zweite Zeile usw.

Ein einzelner Sicherheitsraum besteht aus sechs Zeilen, wobei die erste Zeile sich auf die erste Position der Systemmasse, die zweite Zeile auf die zweite Position der Systemmasse usw. bezieht. In den Sicherheitsräumen der Systemmasse gibt es keine Bewegungsnummern.

Konfig-Text --> SPS-Text

Mit dieser Funktion ist es möglich, Texte aus der Konfiguration des Teach In in die Betextung der SPS zu übernehmen. Beim Aufruf der Funktion muss zunächst die Konfigurationsdatei ausgewählt werden, für die die Betextung übernommen werden soll. Anschließend wird noch einmal nachgefragt, ob die Texte wirklich übernommen werden sollen. Dies ist notwendig, da der Vorgang **nicht rückgängig** gemacht werden kann.

Bei der Übernahme der Texte liest die Firmware die gesamte, ausgewählte Konfigurationsdatei ein und verfährt dann mit jedem Datensatz nach folgendem Schema:

- Handelt es sich bei der Bewegung um eine Achse (Achsennummer ungleich Null), so wird der Bewegungstext in die Achsbezeichnung der angegebenen Achse kopiert
- Handelt es sich bei der Bewegung um einen Ablauf (Ablaufnummer ungleich Null), so wird der angegebene Bewegungstext in die Ablaufbezeichnung des angegebenen Ablaufs kopiert.
- Bei Abläufen wird für die vier definierten Softkeys noch die Jobbezeichnung kopiert, wenn der Auftrag des Softkeys ungleich Null ist. Dabei wird die entsprechende Softkeybezeichnung auf die Jobnummer kopiert, die als Auftrag für diesen Softkey definiert ist. Jobbezeichnungen sind dabei grundsätzlich ablauf- und auftragsabhängig.

Bereits bestehende Texte innerhalb der SPS-Betextung werden mit dieser Funktion überschrieben! Nach dem Erzeugen der Texte muss das Terminal **nicht** neu gestartet werden.

Diese Funktion ist in Systemen mit einem Versionsstand ab 10.05.2004 implementiert!

Beispiel 1:

Bewegungstext = „Y-Achse“, Achse = 1, Ablauf = 0

Die Firmware erzeugt eine Achsbezeichnung „Y-Achse“ für die Achse 1.

Beispiel 2:

Bewegungstext = „Förderband“, Achse = 0; Ablauf = 23, Text für Softkey1 = „Einschalten“,

Auftrag für Softkey1 = 10, Text für Softkey2 = „Ausschalten“, Auftrag für Softkey2 = 15

Die Firmware erzeugt eine Ablaufbezeichnung „Förderband“ für den Ablauf 23.

Die Firmware erzeugt einen Text „Einschalten“ für den Job 10 des Ablaufs 23.

Die Firmware erzeugt einen Text „Ausschalten“ für den Job 15 des Ablaufs 23.

Geschwindigkeitstabelle

In Softwareständen ab 10.03.2006 verfügt das Teach-In über eine Geschwindigkeitstabelle. Diese Tabelle ist programmabhängig und trägt den Dateinamen TEACH@@.VTB. Diese Tabelle ist rekordbasiert und weißt pro Rekord einen Geschwindigkeitswert auf. Jeder Rekord kann dabei betextet werden, wobei sich der Textschlüssel auf dem fixen Teil TEACHV und der Rekordnummer ergibt (Beispiel: TEACHV.....1 für den ersten Eintrag).

Die Geschwindigkeitstabelle wird innerhalb des Teach-Programms über das Feld Geschwindigkeit adressiert. Bei Befehlen, die dieses Feld als echte Geschwindigkeit interpretieren führen Werte über 1000 dazu, dass auf die Geschwindigkeitstabelle zugegriffen wird. Dabei gibt der eingetragene Wert minus 1000 die Zeile innerhalb der Geschwindigkeitstabelle an (Beispiele: 1001 verwendet die Geschwindigkeit, die in der Zeile 1 der Geschwindigkeitstabelle eingetragen ist; 1005 verwendet die Geschwindigkeit, die in der Zeile 5 der Geschwindigkeitstabelle eingetragen ist).

Wird die angegebene Zeile in der Geschwindigkeitstabelle oder die Geschwindigkeitstabelle an sich nicht gefunden, so wird als Geschwindigkeit 1000 Promille verwendet.

Die nachfolgende Abbildung zeigt das Editierbild für die Geschwindigkeitstabelle:



GESCHWINDIGKEITSTABELLE		
Nr.	BEZEICHNUNG	WERT
1	TEACHV.....1	10
2	TEACHV.....2	20
3	TEACHV.....3	300
4	TEACHV.....4	1000
5	TEACHV.....5	0
6	TEACHV.....6	0
7	TEACHV.....7	0
8	TEACHV.....8	0
9	TEACHV.....9	0

Abbildung 21: Geschwindigkeitstabelle Teach-In