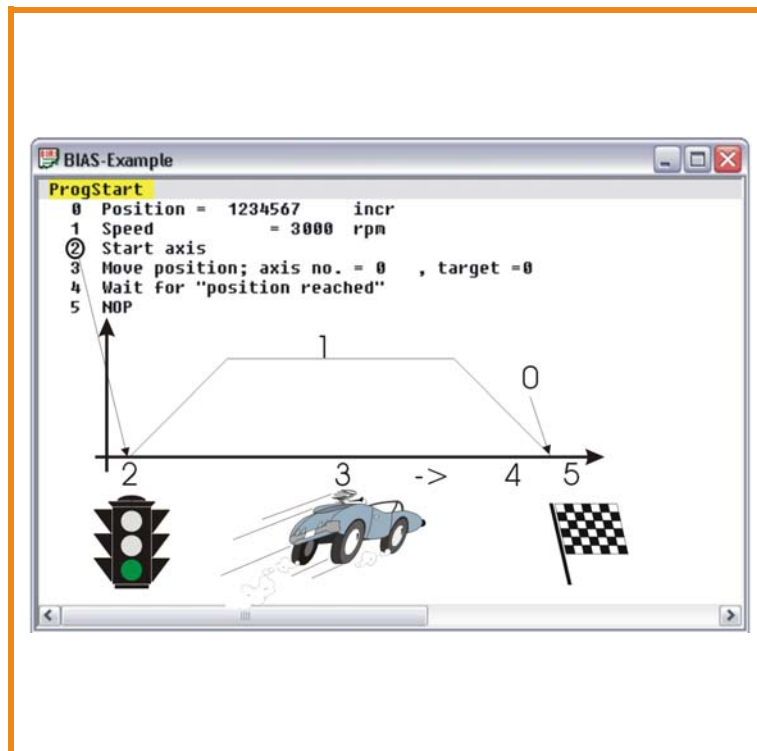


BIAS - Befehle



UL: 07-01-08-02



Produkt - Handbuch 631

UL: 07-01-05-06



Produkt - Handbuch 635

UL: 07-02-08-03



Produkt - Handbuch 637

UL: 07-02-09-01



Produkt - Handbuch 637+

UL: 07-02-10-01



Produkt - Handbuch 637f

CD-ROM:



EASYRIDER® Windows - Software

©SSD Drives GmbH.

Alle Rechte vorbehalten. Kein Teil der Beschreibung darf in irgendeiner Form, ohne Zustimmung der Gesellschaft vervielfältigt oder weiter verarbeitet werden.

Änderungen sind ohne vorherige Ankündigung vorbehalten.

SSD Drives hat für seine Produkte teilweise Warenzeichenschutz und Gebrauchsmusterschutz eintragen lassen. Aus dem Überlassen der Beschreibungen darf nicht angenommen werden, dass damit eine Übertragung von irgendwelchen Rechten stattfindet.

Hergestellt in Deutschland, 2004

Das Wichtigste zuerst	6
1 Allgemein 630 Programmiersprache BIAS	7
1.0 Einleitung	7
1.1 Programmaufbau	8
1.2 Abarbeitung eines BIAS-Programms.....	8
1.3 Abarbeitung eines SPS-Programms.....	9
1.4 Abarbeitung eines Mathematik-Programms	9
1.5 BIAS-Beispielprogramme	10
2 BIAS Befehlsübersicht.....	22
3 BIAS - Befehlsgruppe.....	23
3.1 BIAS-Befehlsgruppe „Fahrbefehle“	23
3.2 BIAS-Befehlsgruppe „Parameterbefehle“	24
3.3 BIAS Befehlsgruppe “Variablenbefehle“	26
3.4 BIAS Befehlsgruppe “Merkerbefehle“	26
3.5 BIAS Befehlsgruppe “Bedingungsbefehle“	27
3.6 BIAS Befehlsgruppe “Programmsteuerbefehle“	27
3.7 BIAS Befehlsgruppe “Mathematikbefehle“	28
3.8 BIAS Befehlsgruppe “Ausgangsbefehle“	28
3.9 BIAS Befehlsgruppe “CAN-Befehle“	29
3.10 BIAS Befehlsgruppe “637f V8.xx Befehle“	29
4 BIAS - Befehl	30
4 - 00 BIAS-Befehl “Fahre Position“	30
4 - 01 BIAS-Befehl “Fahre Kettenposition“	31
4 - 02 BIAS-Befehl “Fahre Referenz“	32
4 - 03 BIAS-Befehl “Fahre unendlich Positiv“	34
4 - 04 BIAS-Befehl “Fahre unendlich Negativ“	35
4 - 05 BIAS-Befehl “Fahre Synchron“	36
4 - 06 BIAS-Befehl “Fahre Synchronprofil“	38
4 - 07 BIAS-Befehl “Synchroneinstellungen 1“	39
4 - 08 BIAS-Befehl “Synchroneinstellungen 2“	40
4 - 09 BIAS-Befehl “Fahre PID, Drehzahl“	42
4 - 0A BIAS-Befehl “Fahre PID, Moment“	44
4 - 0B BIAS-Befehl “Sollposition [AchsNr] = Konst.“	46
4 - 0C BIAS-Befehl “Sollposition [AchsNr] = [Variable X]“	47
4 - 0D BIAS-Befehl “Fahre relativ“	48
4 - 0E BIAS-Befehl “Starte Achse“	49
4 - 0F BIAS-Befehl “Stoppe Achse“	50
4- 10 BIAS-Befehl “Fahre Position + Parameter“	51
4 - 11 BIAS-Befehl “Fahre Kettenposition + Parameter“	52
4 - 12 BIAS-Befehl “Fahre Referenz +Parameter“	53
4 - 13 BIAS-Befehl “Fahre unendlich Positiv+Parameter“	55
4 - 14 BIAS-Befehl “Fahre unendlich Negativ+Parameter“	56
4 - 15 BIAS-Befehl “Fahre Synchron+Parameter“	57
4 - 16 BIAS-Befehl “Fahre Analogwert + Integrator“	58
4 - 17 BIAS-Befehl “Fahre Drehzahl + Integrator“	59
4 - 1A BIAS-Befehl “Taktlänge = Konst.“	60
4 - 1B BIAS-Befehl “Taktlänge = [Variable X]“	60
4 - 1C BIAS-Befehl “Lade Parametersatz X = [Variable [Y]]	61
4 - 1F BIAS-Befehl “Stoppe Achse + Parameter“	65
4 - 20 BIAS-Befehl “Weg = Konst.“	66
4 - 21 BIAS-Befehl “Geschwindigkeit = Konst.“	67
4 - 22 BIAS-Befehl “Beschleunigung = Konst.“	68
4 - 23 BIAS-Befehl “Verzögerung = Konst.“	69
4 - 24 BIAS-Befehl “Koppelfaktor = Konst.“	70
4 - 25 BIAS-Befehl “Position erreicht" Fenster = Konst.	71
4 - 26 BIAS-Befehl “Restweg = Konst.“	72

	Seite
4 - 27	BIAS-Befehl "Rampenfilter = Konst.,[Variable Y]" 73
4 - 28	BIAS-Befehl "Istposition X = Konst." 74
4 - 29	BIAS-Befehl "Wenn Istposition X ? Konst. dann Springe" 75
4 - 2A	BIAS-Befehl "Wenn Istposition X ? [Variable Y] dann Springe" 76
4 - 2B	BIAS-Befehl "Sensorfenster = Konst." 77
4 - 2C	BIAS-Befehl "Sensorfenster = Konst." 79
4 - 2D	BIAS-Befehl "Sensoreinstellung 1 = Konst." 80
4 - 2E	BIAS-Befehl "Sensoreinstellung 2 = Konst." 81
4 - 2F	BIAS-Befehl "Parameterübernahme" 82
4 - 30	BIAS-Befehl "Weg = [Variable X]" 83
4 - 31	BIAS-Befehl "Geschwindigkeit = [Variable X]" 84
4 - 32	BIAS-Befehl "Beschleunigung = [Variable X]" 85
4 - 33	BIAS-Befehl "Verzögerung = [Variable X]" 86
4 - 34	BIAS-Befehl "Koppelfaktor = [Variable X]" 87
4 - 35	BIAS-Command "Position reached" window = [variable X]" 88
4 - 36	BIAS-Befehl "Restweg = [Variable X]" 89
4 - 37	BIAS-Befehl "Maximalstrom = [Variable X]" 90
4 - 38	BIAS-Befehl "Istposition X = [Variable Y]" 91
4 - 39	BIAS-Befehl "Analogausgang X = [Variable Y]" 92
4 - 3A	BIAS-Befehl "PID Skalierung" 93
4 - 3B	BIAS-Befehl "Sensorfenster = [Variable X]" 94
4 - 3C	BIAS-Befehl "Sensorposition = [Variable X]" 96
4 - 3D	BIAS-Befehl "Sensoreinstellung 1 = [Variable X]" 97
4 - 3E	BIAS-Befehl "Sensoreinstellung 2 = [Variable X]" 98
4 - 3F	BIAS-Befehl "PID Parameter" 99
4 - 40	BIAS-Befehl "[Variable X] = Weg" 100
4 - 41	BIAS-Befehl "[Variable X] = Geschwindigkeit" 101
4 - 42	BIAS-Befehl "[Variable X] = Beschleunigung" 102
4 - 43	BIAS-Befehl "[Variable X] = Verzögerung" 103
4 - 44	BIAS-Befehl "[Variable X] = Koppelfaktor" 104
4 - 45	BIAS-Befehl "[Variable X] = Satznummer" 105
4 - 46	BIAS-Befehl "[Variable X] = Istposition Y" 106
4 - 47	BIAS-Befehl "[Variable X] = Analogeingang" 107
4 - 48	BIAS-Befehl "[Variable X] = Latchposition Y" 108
4 - 49	BIAS-Befehl "[Variable X] = Drehzahl Y" 109
4 - 4A	BIAS-Befehl "[Variable X] = Latchstatus Y" 110
4 - 4B	BIAS-Befehl "[Variable X] = Position Y" 111
4 - 4C	BIAS-Befehl "[Variable X] = Wert Y" 112
4 - 4D	BIAS-Befehl "[Variable X] = Achsstatus, AchsNr. Y" 120
4 - 50	BIAS-Befehl "NOP" 122
4 - 51	BIAS-Befehl "Programmende, Modus =" 123
4 - 52	BIAS-Befehl "Unterprogramm" 124
4 - 53	BIAS-Befehl "Unterprogrammende" 125
4 - 54	BIAS-Befehl "SPS-Programm" 126
4 - 55	BIAS-Befehl "Springe Konst." 127
4 - 46	BIAS-Befehl "Springe [Variable X]" 128
4 - 57	BIAS-Befehl "BIAS – Abarbeitungszeiger = Konst." 129
4 - 58	BIAS-Befehl "Warte auf "Position erreicht"" 130
4 - 59	BIAS-Befehl "Wartezeit = Konst." 131
4 - 5A	BIAS-Befehl "Wartezeit = [Variable X]" 132
4 - 5B	BIAS-Befehl "BIAS-Abarbeitungszeiger = [Variable X]" 133
4 - 5C	BIAS-Befehl "Springe [Var.[X]]; Länge = Y ; ab Zeile" 134
4 - 5D	BIAS-Befehl "Bearbeite X Befehle" 135
4 - 5F	BIAS-Befehl "Virtuelles Programm" 136
4 - 60	BIAS-Befehl "Merker X = Konst." 137
4 - 61	BIAS-Befehl "Wenn Merker X ? Konst. dann Springe" 138
4 - 62	BIAS-Befehl "Merker X = Merker Y" 139
4 - 63	BIAS-Befehl "Merker X = Eingang Y" 140
4 - 64	BIAS-Befehl "Merker X = Ausgang Y" 141
4 - 65	BIAS-Befehl "Merker X = Merker Y & Merker Z" 142
4 - 66	BIAS-Befehl "Merker X = Merker Y Merker Z" 143

	Seite
4 - 67	BIAS-Befehl "Merker X = Merker Y ^ Merker Z" 144
4 - 68	BIAS-Befehl "Merker X = !Merker Y" 145
4 - 69	BIAS-Befehl "Merker X = Status Y" 146
4 - 6A	BIAS-Befehl "Wenn Status X ? Konst. dann Springe" 147
4 - 6B	BIAS-Befehl "Modus X = Konst." 148
4 - 6C	BIAS-Befehl "Merker X = [Variable Y], Anzahl Z" 149
4 - 6D	BIAS-Befehl "[Variable X]. Bit [Y] = Konst." 150
4 - 6E	BIAS-Befehl "Wenn[Variable X]. Bit Y ? Konst. dann Springe" 151
4 - 6F	BIAS-Befehl "Achszustand, AchsNr. X, Bit Y =Konst.[Merker Z]" 152
4 -70	BIAS-Befehl "Wenn Eingang X ? Konst. dann Springe" 154
4 -71	BIAS-Befehl "Wenn Ausgang X ? Konst. dann Springe" 156
4 - 72	BIAS-Befehl "Ausgang X = Konst." 158
4 - 73	BIAS-Befehl "Ausgang X = Merker Y" 159
4 - 78	BIAS-Befehl "IBT-Maskennummer = X, LED Status [Variable Y]" 160
4 - 79	BIAS-Befehl "IBT-Meldungsnummer =X, LED Status[VariableY]" 161
4 - 7A	BIAS-Befehl "CAN-Kommando = [Variable X]" 162
4 - 7B	BIAS-Befehl "IBT-Datentransfer" 164
4 - 7C	BIAS-Befehl "CAN2-Kommando = [Variable X]" 165
4 - 80	BIAS-Befehl "[Variable X] = Konst." 167
4 - 81	BIAS-Befehl "Wenn [Variable X] ? Konst." 168
4 - 82	BIAS-Befehl "[Variable X] = [Variable Y] + Konst." 170
4 - 83	BIAS-Befehl "[Variable X] = [Variable Y] - Konst." 171
4 - 84	BIAS-Befehl "[Variable X] = [Variable Y] * Konst." 172
4 - 85	BIAS-Befehl "[Variable X] = [Variable Y] / Konst." 173
4 - 86	BIAS-Befehl "[Variable X] = Merker Y" 174
4 -87	BIAS-Befehl "[Variable X] = [Variable Y].Bit Z ; Anzahl" 175
4 - 88	BIAS-Befehl "[Variable X] = [Variable Y]" 176
4 - 89	BIAS-Befehl "Wenn [Variable X] ? [Variable Y]" 177
4 - 8A	BIAS-Befehl "[Variable X] = [Variable Y] + [Variable Z]" 179
4 - 8B	BIAS-Befehl "[Variable X] = [Variable Y] - [Variable Z]" 180
4 - 8C	BIAS-Befehl "[Variable X] = [Variable Y] * [Variable Z]" 181
4 - 8D	BIAS-Befehl "[Variable X] = [Variable Y] / [Variable Z]" 182
4 - 8E	BIAS-Befehl "[Teachvariable X] = [Variable Y]" 183
4 - 8F	BIAS-Befehl "[Variable X] = [Teachvariable Y]" 184
4 - 90	BIAS-Befehl "Mathematikprogramm" 185
4 -91	BIAS-Befehl "Profilinitialisierung" 186
4 - 92	BIAS-Befehl "Profilaktlänge = [Variable X]" 187
4 - 93	BIAS-Befehl "[Variable X] = Profilwert" 188
4 - 94	BIAS-Befehl "Profilwert = [Variable X]" 189
4 - 97	BIAS-Befehl "Tabelle speichern" 190
4 - A0	BIAS-Befehl "Tabelle ([Variable X])= Konst." 191
4 - A1	BIAS-Befehl "Tabelle ([Variable X]) = [Y Variable Z]" 192
4 - A2	BIAS-Befehl "[X Variable Y] = Tabelle ([Variable Z])" 193
4 - A3	BIAS-Befehl "[W Variable X] = [Y Variable Z]" 194
4 - A4	BIAS-Befehl "[X Variable Y] = Konst." 195
4 - A5	BIAS-Befehl "[Variable [X]] = Konst." 196
4 - A6	BIAS-Befehl "[Variable [X]] = [Variable Y]" 197
4 - A7	BIAS-Befehl "[Variable X] = [Variable [Y]]" 198
4 - A8	BIAS-Befehl "[Variable X] = [Variable Y] ? [Variable Z]" 199
4 - A9	BIAS-Befehl "[Variable X] = [Variable Y] ? Konst." 200
4 - B0	BIAS-Befehl "[D_V Variable X] = [D_V Variable Y] + [D_V Variable Z]" 201
4 - B1	BIAS-Befehl "[D_V Variable X] = [D_V Variable Y] - [D_V Variable Z]" 202
4 - B2	BIAS-Befehl "[D_V Variable X] = [D_V Variable Y] * [D_V Variable Z]" 203
4 - B3	BIAS-Befehl "[D_V Variable X] = [D_V Variable Y] / [D_V Variable Z]" 204
4 - B4	BIAS-Befehl "Wenn [D_V Variable X] ? [D_V Variable Y]" 205
4 - B5	BIAS-Befehl "[D_V Variable X] = SIN {[D_V Variable Y]}" 207
4 - B6	BIAS-Befehl "[D_V Variable X] = COS {[D_V Variable Y]}" 208
4 - B7	BIAS-Befehl "[D_V Variable X] = SQRT {[D_V Variable Y]}" 209
5	Änderungsliste..... 210

Wir bedanken uns für das Vertrauen, das Sie unserem Produkt entgegenbringen. Die vorliegende Betriebsanleitung dient der Übersicht von technischen Daten und Eigenschaften.

Bitte lesen Sie vor Einsatz des Produktes diese Bedienungsanleitung.

Bei Rückfragen wenden Sie sich bitte an Ihren nächsten SSD Drives - Ansprechpartner.

Der nicht sachgemäße Einsatz des Produktes im Zusammenhang mit lebensgefährlicher Spannung kann zu Verletzungen führen.

Des Weiteren können dadurch Beschädigungen an Motoren oder Produkten auftreten. Berücksichtigen Sie deshalb bitte unbedingt unsere Sicherheitshinweise.

Sicherheitshinweise

Wir gehen davon aus, dass Sie als Fachmann mit den einschlägigen Sicherheitsregeln, insbesondere nach VDE 0100, VDE 0113, VDE 0160, EN 50178 den Unfallverhütungsvorschriften der Berufsgenossenschaft und den DIN-Vorschriften vertraut sind und mit ihnen umgehen können.

Weiterhin sind die Bestimmungen nach den relevanten europäischen Richtlinien einzuhalten.

Je nach Einsatzart sind weitere nationale Normen, wie z. B. UL, DIN zu beachten. Wenn der Einsatz unserer Produkte im Zusammenhang mit Komponenten anderer Hersteller erfolgt, sind auch deren Betriebsanleitungen unbedingt zu beachten.

630 Programmiersprache “BIAS“

1.0 Einleitung

Die Programmiersprache “BIAS“

Bedienersprache für intelligente Antriebs – Steuerungen

wurde entwickelt um die Programmierung komplexer und doch übersichtlicher Programme zu ermöglichen. Dazu wurden die BIAS-Befehle entsprechend ihrer Funktion in die folgenden 12 Befehlsgruppen untergliedert:

0. Fahrbefehle
1. Fahrbefehle + Parameter
2. Parameterbefehle
3. "Parameter aus Variablen" - Befehle
4. "Parameter in Variablen" - Befehle
5. Steuerbefehle
6. Merkerbefehle
7. Ein-/ Ausgangsbefehle
8. Variablenbefehle
9. Mathematikbefehle 1
10. Mathematikbefehle 2
11. Fließkommabefehle

BIAS - [Befehlsübersicht](#)

Ab der Softwareversion 5.10 besteht bei den Geräten der 630 Serie in den zusätzlichen Befehlsgruppen 9 - 11 die Möglichkeit Rechenoperationen speziell für die Online- Berechnung von Synchronprofilen zu programmieren. Weitere Informationen finden sie im Handbuch Motion-Control (UL:10.6.7).

Sie haben mit diesen Befehlen die Möglichkeit den geforderten Maschinenablauf in Schrittketten zu programmieren. Die Größe eines Programms ist auf maximal 1500 BIAS-Befehle begrenzt. Die Erstellung der Programme erfolgt mit der EASYRIDER-Software am PC und kann mittels serieller Kommunikation in den Regler übertragen werden.

Wenn Sie das BIAS-Programm mit der **EASYRIDER** Oberfläche erstellen, stehen Ihnen Sprunglabel , Kommentare und eine Einheit für die Positionsvorgaben zur Verfügung.

Eine weitere Möglichkeit ist die Programmierung bzw. das Senden und die Steuerung des BIAS-Programms über einen Feldbus. Die dafür erforderliche Befehlskodierung ist in den Befehlsbeschreibungen dargestellt

1.1 Programmaufbau

Ein BIAS-Programm besteht aus 3 grundlegenden Speicherbereichen.

1. Die Programmdefinition:

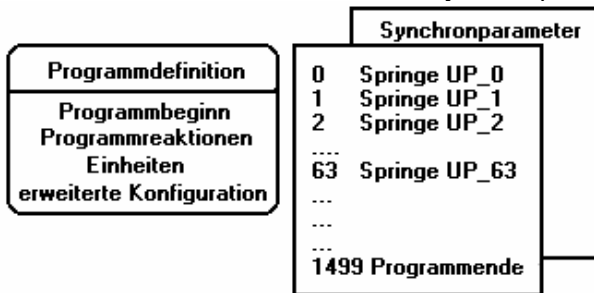
enthält alle Definitionen zum Start und der Abarbeitung eines BIAS-Programms, die Eingaben zur Definition einer Einheit für die Positionsvorgabe und die notwendigen Konfigurationen der Ein- und Ausgänge.

2. Der Befehlsspeicher:

enthält bis zu 1500 BIAS-Befehle.

Die Synchronparameter

enthalten die Definitionen für die 16 Synchronprofilblöcke und die 2048 Stützstellen.



Alle 3 Speicherbereiche sind zusammengefasst als ein BIAS-Programm zu verwalten.

In der DOS-Version der EASYRIDER Software hat die Datei die Erweiterung *.ASB.

In der EASYRIDER für Windows Software ist die Dateierweiterung *.WBD

In der EASYRIDER für Windows Version können mit der DOS Version erstellte Programme (*.ASB) eingelesen werden.

Beim Abspeichern der Programme wird dann in die Windowsversion (*.WBD) konvertiert.

Die DOS-Version vom EASYRIDER ist nicht in der Lage *.WBD Dateien einzulesen !!

1.2 Abarbeitung eines BIAS-Programms

In der Betriebsart 5 "Lageregelung mit BIAS-Abarbeitung" wird nach der Aktivierung der Reglerendstufe die BIAS-Abarbeitung gestartet.

Der erste auszuführende BIAS-Satz wird in der BIAS-Programmdefinition (Parameter "Programmbeginn") festgelegt.

Anschließend arbeitet der Regler sequentiell jeden Fahrkurvengeneratorzyklus einen BIAS-Befehl ab.

Erreicht die BIAS-Abarbeitung einen Fahrbefehl kann dieser durch die Low-High-Flanke des Starteingangs gestartet werden.

Serie	Eingang	Konfiguration
635/ 637/637+/637f:	X10.11	"Starteingang BIAS" (Funktion 0)
631:	X10.9	"Starteingang" (Funktion 3)

Alternativ dazu werden Fahrbefehle gestartet, wenn über den BIAS-Befehl "Starte Achse" vor dem Fahrbefehl die Startkennung gesetzt wird.

Nach erfolgtem Start werden die folgenden Sätze abgearbeitet.

Folgt nach einem Fahrbefehl der Befehl "Warte auf Position erreicht" wird erst nach Erreichen der Zielposition mit der Satzabarbeitung fortgefahren.

Reglertyp	Fahrkurvengeneratorzyklus
631/635/637	1,899ms
637+/637f	0,844ms

1.3 Abarbeitung eines SPS-Programms

Parallel zu der sequentiellen Abarbeitung des BIAS-Programms kann ein zyklisches SPS-Programm für übergeordnete Überwachungsaufgaben gestartet werden.

Der Start des SPS-Programms erfolgt mit der Abarbeitung des BIAS-Befehls "SPS-Programm".

Nach der Aktivierung des SPS-Programms werden die ab der angegebenen Satznummer programmierten SPS-Befehle abgearbeitet.

Der Befehl "Programmende, **Modus = 0**" innerhalb eines SPS-Programms bewirkt den Rücksprung zum Start des SPS-Programms.

Der Regler arbeitet sequentiell jeden Fahrkurvengeneratorzyklus ein SPS-Befehl ab.

In der BIAS-Programmdefinition (Parameter "Programmreaktion SPS-Programm") kann die Reaktion des SPS-Programms auf die Deaktivierung der Endstufe eingestellt werden. Damit ist es möglich das SPS-Programm auch während der Deaktivierung des Reglers weiter abarbeiten zu lassen.

Wird in diesem Modus der Befehl "SPS -Programm ..." auf der **1. Zeile** der BIAS- Programmabarbeitung programmiert, so läuft das SPS - Programm auch ohne Aktivierung des Reglers an.

Innerhalb eines SPS-Programmes sind nicht alle BIAS-Befehle erlaubt.

In der Befehlsübersicht in Kapitel 3 sind die erlaubten Befehle aufgelistet.

Eine Überprüfung der erlaubten Befehle erfolgt erst zum Zeitpunkt der Programmausführung auf dem Regler!

Reglertyp	Fahrkurvengeneratorzyklus
631/635/637	1,899ms
637+/637f	0,844ms

1.4 Abarbeitung eines Mathematik-Programms

Parallel zu der sequentiellen Abarbeitung des BIAS-Programms und/oder SPS-Programmes kann eine 3. Task als Mathematik-Programm für übergeordnete Berechnungsfunktionen gestartet werden.

Der Start des Mathematik-Programms erfolgt mit der Abarbeitung des BIAS-Befehls "Mathematik-Programm".

Nach der Aktivierung des Mathematik-Programms werden die ab der angegebenen Satznummer programmierten Mathematik-Befehle abgearbeitet.

Der Befehl "Programmende **Modus = 0**" innerhalb eines Mathematik - Programms bewirkt den Rücksprung zum Start des Mathematik-Programms.

Der Befehl "Programmende **Modus = 3**" beendet das Mathematik-Programm.

Die Abarbeitung der Mathematikbefehle erfolgt in der interruptfreien Rechenzeit des Reglers.

Bei einer Standardanwendung werden etwa 10 Befehle in 2ms bearbeitet.

In der BIAS-Programmdefinition (Parameter "Programmreaktion Math-Programm") kann die Reaktion des Mathematik-Programms auf die Deaktivierung der Endstufe eingestellt werden. Damit ist es möglich das Mathematik-Programm auch während der Deaktivierung des Reglers weiter abarbeiten zu lassen.

Wird in diesem Modus der Befehl " Mathematik -Programm ..." auf der **1. Zeile** bzw. **2. Zeile** (falls SPS-Programm auf 1. Zeile) der BIAS- Programmabarbeitung programmiert oder auf der Zeile 0 des BIAS- Programmes, so läuft das Mathematik - Programm auch ohne Aktivierung des Reglers an.

1.5 BIAS-Beispielprogramme

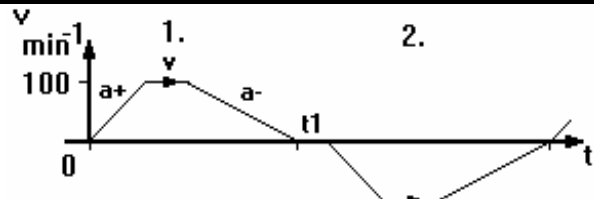
Die Beispielprogramme können in der EASYRIDER Software im BIAS - Editor im Menue „Datei“ unter dem Menüpunkt „BIAS-Beispiel laden“ aufgerufen werden.

1. BIAS-Applikationsbeispiel: Reversierbetrieb freie Motorwelle

Erklärung: 1. Absolutpositionierung auf die Position 16384 Inkremente
2. Wartezeit von 1 Sekunde
3. Absolutpositionierung auf die Position 0 Inkremente

Dateiname: BSP_001.wbd

Voraussetzungen: Betriebsart 5 ("Lageregelung mit BIAS-Abarbeitung")



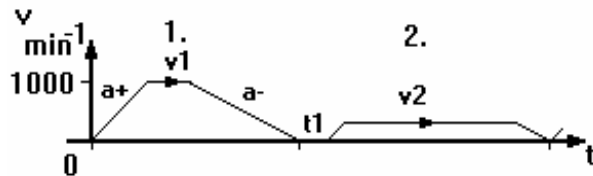
Programmlisting

```
* *****
* !!! Achtung !!! Das Programm führt zu automatischen Bewegungen der Achse !
* *****
PROG_START:
* Startpunkt des BIAS-Programms nach jeder Aktivierung der Endstufe
0 Istposition 1 = 0          INKR
* *****
* Definition des Verfahrensprofils für die 1. Positionierung
1 Beschleunigung = 5000 min-1/s
* Definition der Beschleunigungsrampe a+
2 Geschwindigkeit = 100 min-1
* Definition der Maximalgeschwindigkeit v
3 Verzögerung = 2500 min-1/s
* Definition der Verzögerungsrampe a-
4 Weg = 16384 INKR
* Definition der 1. Zielposition in Inkrementen
* *****
5 Starte Achse
* Starterlaubnis erteilen (Startkennung setzen)
6 Fahre Position, AchsNr. = 0, Ziel = 0
* 1. Positionierung als Absolutpositionierung starten
* *****
7 Warte auf Pos. erreicht
* BIAS-Abarbeitung bis zum Erreichen der Zielposition unterbrechen
8 Warte Zeit 1000 ms
* Wartezeit (t1) von 1000 ms
* *****
* Definition des Verfahrensprofils für die 2. Positionierung
9 Weg = 0 INKR
* Definition der 2. Zielposition, die Definitionen für die Beschleunigungsrampe, die Maximalgeschwindigkeit
* und/oder die Verzögerungsrampe werden nicht verändert, sie behalten also ihre Werte und Gültigkeit.
10 Starte Achse
* Starterlaubnis erteilen (Startkennung setzen)
11 Fahre Position, AchsNr. = 0, Ziel = 0
* 2. Positionierung als Absolutpositionierung starten
12 Warte auf Pos. erreicht
* BIAS-Abarbeitung bis zum Erreichen der Zielposition unterbrechen
13 Programmende, Modus 0
* BIAS-Abarbeitung am Programmbeginn fortführen (auf dem Satz 0, oder am Label "PROG_START")
```

Erklärung: 2. **BIAS-Applikationsbeispiel: Vorschub freie Motorwelle**
 1. Relativpositionierung in positiver Richtung um 100000 Inkremente
 2. Wartezeit von 1 Sekunde
 3. Relativpositionierung in positiver Richtung um 32700 Inkremente

Dateiname: BSP_002. wbd

Voraussetzungen: Betriebsart 5 ("Lageregelung mit BIAS-Abarbeitung")



Programmlisting

```

* *****
* !!! Achtung !!! * Das Programm führt zu automatischen Bewegungen der Achse !
* *****
PROG_START:
* Definition des Verfahrensprofils für die 1. Positionierung
0 Beschleunigung = 5000 min-1/s
* Definition der Beschleunigungsrampe a+
1 Verzögerung = 2500 min-1/s
* Definition der Verzögerungsrampe a-
Start_Pos:
2 Geschwindigkeit = 1000 min-1
  Definition der Maximalgeschwindigkeit v1
3 Weg = 100000 INKR
* Definition der 1. Zielposition in Inkrementen (bei hoher Resolverauflösung ca. 6,1 Motorumdrehungen)
* *****
4 Starte Achse
* Starterlaubnis erteilen (Startkennung setzen)
5 Fahre Kettenposition, AchsNr. = 0, Ziel = 0
* 1. Positionierung als Relativpositionierung starten
* *****
6 Warte auf "Pos. erreicht"
* BIAS-Abarbeitung bis zum Erreichen der Zielposition unterbrechen
7 Warte Zeit 1000 ms
* zusätzliche Wartezeit von 1000 ms einfügen
* *****
* Definition des Verfahrensprofils für die 2. Positionierung
8 Weg = 32700 INKR
* Definition der 2. Zielposition (bei hoher Resolverauflösung ca. 2 Motorumdrehungen)
9 Geschwindigkeit = 250 min-1
* Definition der Maximalgeschwindigkeit v2
* *****
10 Starte Achse
* Starterlaubnis erteilen (Startkennung setzen)
11 Fahre Kettenposition, AchsNr. = 0, Ziel = 0
* 2. Positionierung als Relativpositionierung starten
12 Warte auf "Pos. erreicht"
* BIAS-Abarbeitung bis zum Erreichen der Zielposition unterbrechen
* *****
13 Springe Start_Pos
* BIAS-Abarbeitung am Label "Start_Pos" fortführen
* *****
  
```

3.1 BIAS-Applikationsbeispiel: Lineareinheit 635/637

Erklärung:

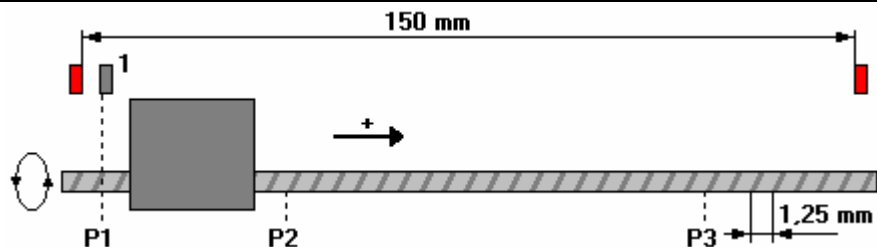
1. Referenzfahrt (Sensor + folgendem Nullimpuls in negativer Richtung)
2. Absolutpositionierung auf die Position P2 (25 mm)
3. Start nach externem Signal (Low-High-Flanke Eingang X10.11) auf die Position P3 (125 mm)
4. Start nach externem Signal (Low-High-Flanke Eingang X10.11) auf die Position P2 (25 mm)
5. Weiter mit 3.

Dateiname: BSP_003. wbd

Voraussetzungen: Betriebsart 5 ("Lageregelung mit BIAS-Abarbeitung")
 Starteingang X10.11 verdrahten und mit Funktion 0 (Starteingang BIAS) konfigurieren
 Referenzsensor (1) verdrahtet auf X10.24 und Konfiguration X10.24 mit Funktion 1 (Ref.sensor)

Spindeltrieb mit
 Steigung = 1,25 mm
 Gesamtlänge = 150 mm

1 = Referenzsensor
 P1 = Referenzposition
 P2 = Zielposition 1
 P3 = Zielposition 2



Programmlisting

```

* *****
* !!! Achtung !!!
* Das Programm führt zu automatischen Bewegungen der Achse !
* Notwendige Eingaben für die Einheit " mm" in den BIAS-Definitionen Einheit (Regler)
* Zähnezah Abtrieb = 100;   Zähnezah Antrieb = 100 (Übersetzung 1:1)
* hohe Geberauflösung     = J (16384 Inkremente / Motorumdrehung)
* Weg pro Abtriebsumdrehung = 1.25 (Spindelsteigung)
* Anzeigetext der Einheit   = mm
* *****

PROG_START:
* Definition des Verfahrens für die Referenzfahrt P1
0 Beschleunigung = 2000 min-1/s
1 Verzögerung = 2000 min-1/s
2 Starte Achse
3 Fahre Referenz; Modus = 11; v = 100 min-1, s = 0 mm
* Modus 11 = Low-High Flanke Referenzsensor + folgende Resolvernnullstellung in negativer Richtung +
* automatische Richtungswahl
4 Warte auf Pos. erreicht
* Referenzfahrt beendet
* *****

Pos_Loop:
* Definition des Verfahrens für die 1. Positionierung P2 (25 mm)
5 Beschleunigung = 25000 min-1/s
6 Verzögerung = 15000 min-1/s
7 Fahre Position; v = 2500 min-1, s = 25 mm
* Starterlaubnis wird über Low-High Flanke X10.11 erteilt
8 Warte auf "Pos. erreicht"
* 1. Positionierung P2 beendet
* *****

* Definition des Verfahrens für die 2. Positionierung P3 (125 mm)
9 Beschleunigung = 5000 min-1/s
10 Verzögerung = 5000 min-1/s
11 Fahre Position; v = 1500 min-1, s = 125 mm
12 Warte auf "Pos. erreicht"
* 2. Positionierung P3 beendet
13 Springe Pos_Loop
    
```

3.2 BIAS-Applikationsbeispiel: Lineareinheit 631

Erklärung: 1. Referenzfahrt (Sensor + folgendem Nullimpuls in negativer Richtung)
 2. Absolutpositionierung auf die Position P2 (25 mm)
 3. Start nach externem Signal (Low-High-Flanke Eingang X10.11) auf die Position P3 (125 mm)

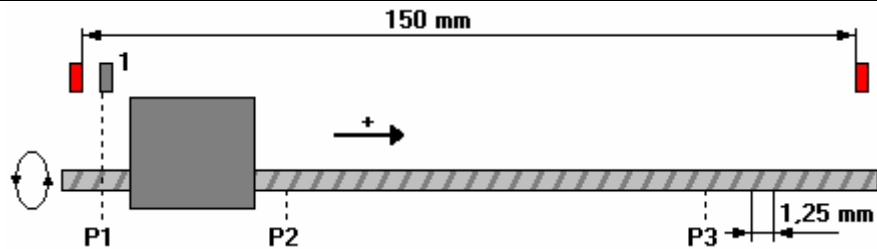
mm)
 4. Start nach externem Signal (Low-High-Flanke Eingang X10.11) auf die Position P2 (25 mm)
 5. Weiter mit 3.

Dateiname: B631_003. wbd

Voraussetzungen: Betriebsart 5 ("Lageregelung mit BIAS-Abarbeitung")
 Starteingang X10.9 verdrahten und mit Funktion 0 (Starteingang BIAS) konfigurieren
 Referenzsensor (1) verdrahtet auf X10.8 und Konfiguration X10.8 mit Funktion 1 (Ref.sensor)

Spindeltrieb mit
 Steigung = 1,25 mm
 Gesamtlänge = 150 mm

1 = Referenzsensor
 P1 = Referenzposition
 P2 = Zielposition 1
 P3 = Zielposition 2



Programmlisting

```

* *****
* !!! Achtung !!!
* Das Programm führt zu automatischen Bewegungen der Achse !
* Notwendige Eingaben für die Einheit " mm" in den BIAS-Definitionen Einheit (Regler)
* Zähnezah Abtrieb = 100;   Zähnezah Antrieb = 100 (Übersetzung 1:1)
* hohe Geberauflösung      = J (16384 Inkremente / Motorumdrehung)
* Weg pro Abtriebsumdrehung = 1.25 (Spindelsteigung)
* Anzeigetext der Einheit   = mm
* *****

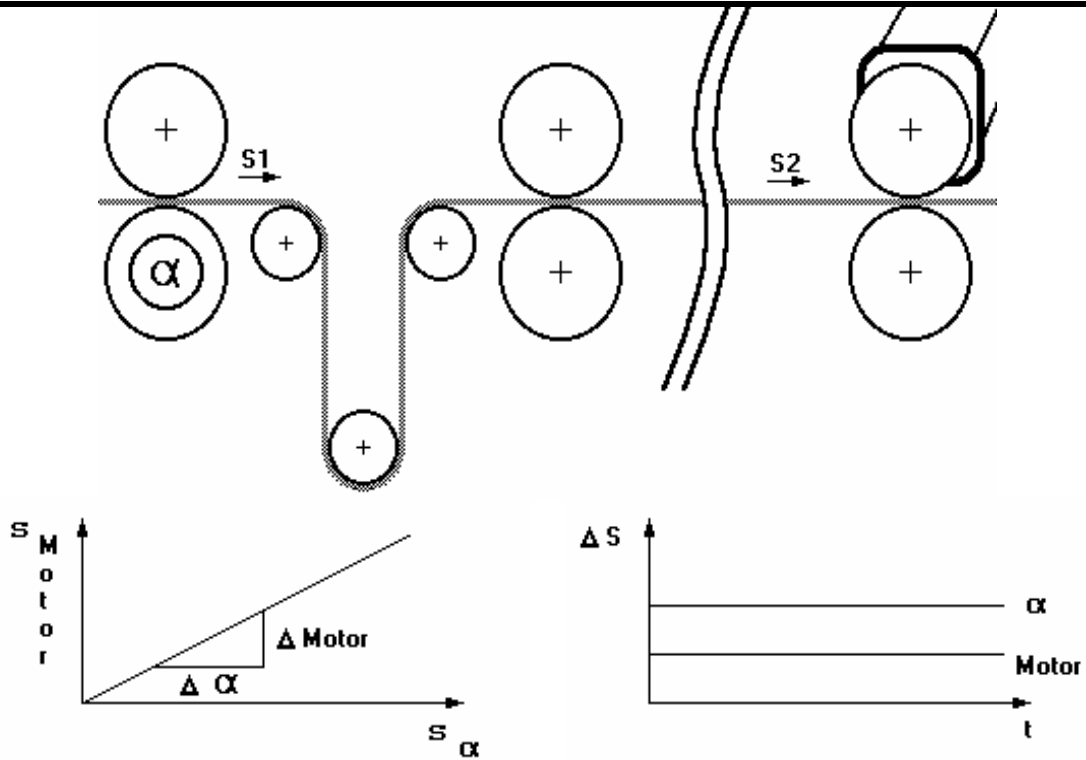
PROG_START:
* Definition des Verfahrsprofils für die Referenzfahrt P1
0 Beschleunigung = 2000 min-1/s
1 Verzögerung = 2000 min-1/s
2 Starte Achse
3 Fahre Referenz; Modus = 11; v = 100 min-1, s = 0 mm
* Modus 11 = Low-High Flanke Referenzsensor + folgende Resolvernulldstellung in negativer Richtung +
* automatische Richtungswahl
4 Warte auf "Pos. erreicht"
* Referenzfahrt beendet
* *****

Pos_Loop:
* Definition des Verfahrsprofils für die 1. Positionierung P2 (25 mm)
5 Beschleunigung = 25000 min-1/s
6 Verzögerung = 15000 min-1/s
7 Fahre Position; v = 2500 min-1, s = 25 mm
* Starterlaubnis wird über Low-High Flanke X10.11 erteilt
8 Warte auf "Pos. erreicht"
* 1. Positionierung P2 beendet
* *****

* Definition des Verfahrsprofils für die 2. Positionierung P3 (125 mm)
9 Beschleunigung = 5000 min-1/s
10 Verzögerung = 5000 min-1/s
11 Fahre Position; v = 1500 min-1, s = 125 mm
12 Warte auf Pos. erreicht
* 2. Positionierung P3 beendet
13 Springe Pos_Loop
  
```

4.1 BIAS-Applikationsbeispiel: Elektronisches Getriebe 635/637/637+/637f

Erklärung: 1. Positionssynchrones Fahren zum externen Inkrementalgeber (α) solange der Eingang X10.11 aktiv (1) ist.
Dateiname: BSP_004. wbd
Voraussetzungen: Betriebsart 5 ("Lageregelung mit BIAS-Abarbeitung")
 Eingang X10.11 konfiguriert mit der Funktion 3 (BIAS-Eingang 11)
 X40-Anschluß als Inkrementaleingang konfiguriert



Programmlisting

```

* *****
* !!! Achtung !!!
* Notwendige E/A - Konfigurationen (BIAS-Definitionen Konfiguration)
* Eingang X10.11 Funktion 3 (BIAS-Eingang 11)
* X40-Modus 1 (Inkrementaleingang)
* X40-Auflösung entsprechend dem externen Geber (8192 Inkremente / Umdrehung)
* *****

PROG_START
0 Wenn Eingang 11 == 0 dann Springe PROG_START
1 Weg = 0 INKR
* Deaktivierung der Synchronlauffunktion "Formatüberwachung"
2 Starte Achse
3 Fahre Synchron; Koppelfaktor = 0.5
    
$$\frac{\Delta s_1}{\Delta t} = \frac{\Delta s_2}{\Delta t}; s_1 = 8192; s_2 = 4096; \text{Koppelfaktor} = \frac{s_2}{s_1} = \frac{4096}{8192} = 0,5$$

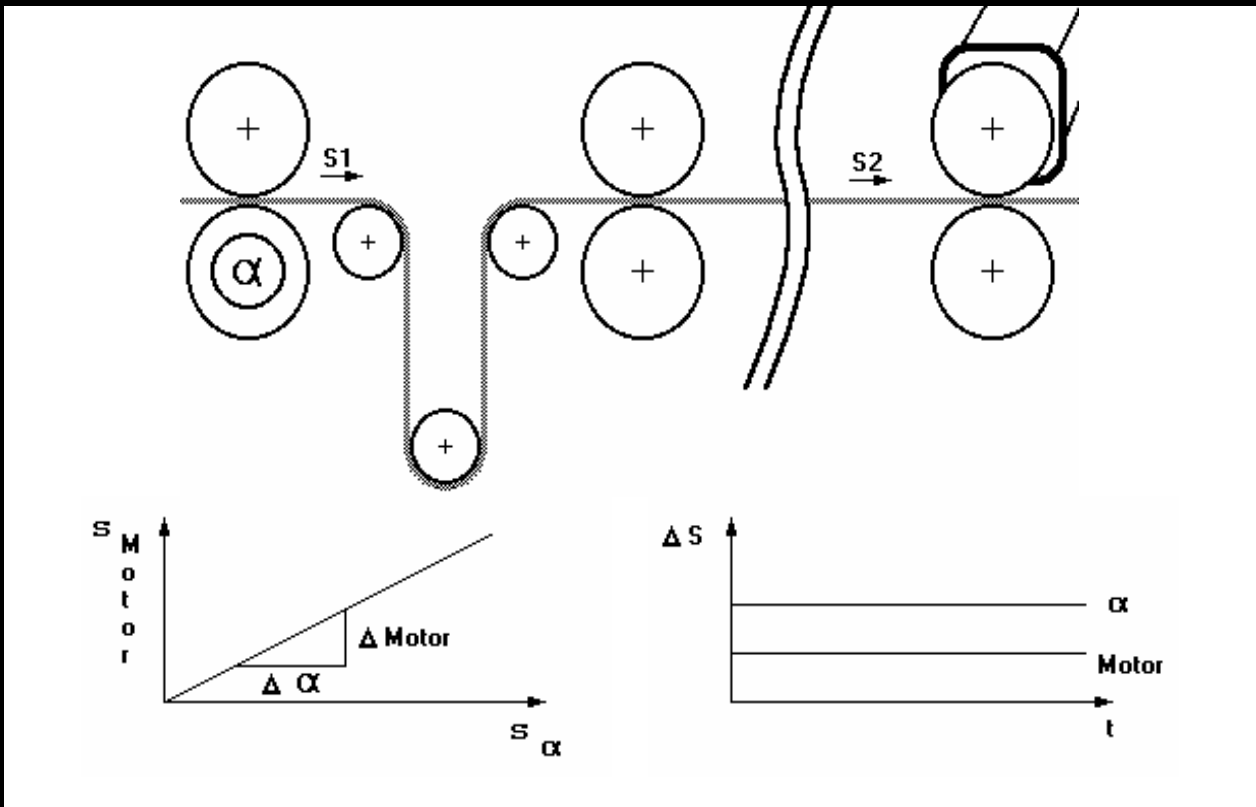
* Forderung:
FAHRE:
4 Wenn Eingang 11 == 1 dann Springe FAHRE
5 Stoppe Achse; Modus = 1 a = 5000 min /s
6 Programmende, Modus 0
    
```

4.2 BIAS-Applikationsbeispiel: Elektronisches Getriebe 631

Erklärung: 1. Positionssynchrones Fahren zum externen Inkrementalgeber (α) solange der Eingang X10.11 aktiv (1) ist.

Dateiname: B631_004. wbd

Voraussetzungen: Betriebsart 5 ("Lageregelung mit BIAS-Abarbeitung")
Eingang X10.8 konfiguriert mit der Funktion 3 (BIAS-Eingang 8)
X40-Anschluß als Inkrementaleingang konfiguriert



Programmlisting

```

* *****
* !!! Achtung !!!
* Notwendige E/A - Konfigurationen (BIAS-Definitionen Konfiguration)
* Eingang X10.9 Funktion 3 (BIAS-Eingang 11)
* X40-Modus 1 (Inkrementaleingang)
* X40-Auflösung entsprechend dem externen Geber (8192 Inkremente / Umdrehung)
* *****

PROG_START
0 Wenn Eingang 8 == 0 dann Springe PROG_START
1 Weg = 0 INKR
* Deaktivierung der Synchronlauffunktion "Formatüberwachung"
2 Starte Achse
3 Fahre Synchron; Koppelfaktor = 0.5
    
$$\frac{\Delta S_1}{\Delta t} = \frac{\Delta S_2}{\Delta t} \quad s_1 = 8192; s_2 = 4096; \text{Koppelfaktor} = \frac{s_2}{s_1} = \frac{4096}{8192} = 0,5$$

* Forderung:
FAHRE:
4 Wenn Eingang 8 == 1 dann Springe FAHRE
5 Stoppe Achse; Modus = 1 a- = 5000 min /s
6 Programmende, Modus 0
    
```

4.3 BIAS-Applikationsbeispiel: Elektronisches Getriebe 637f

Erklärung:

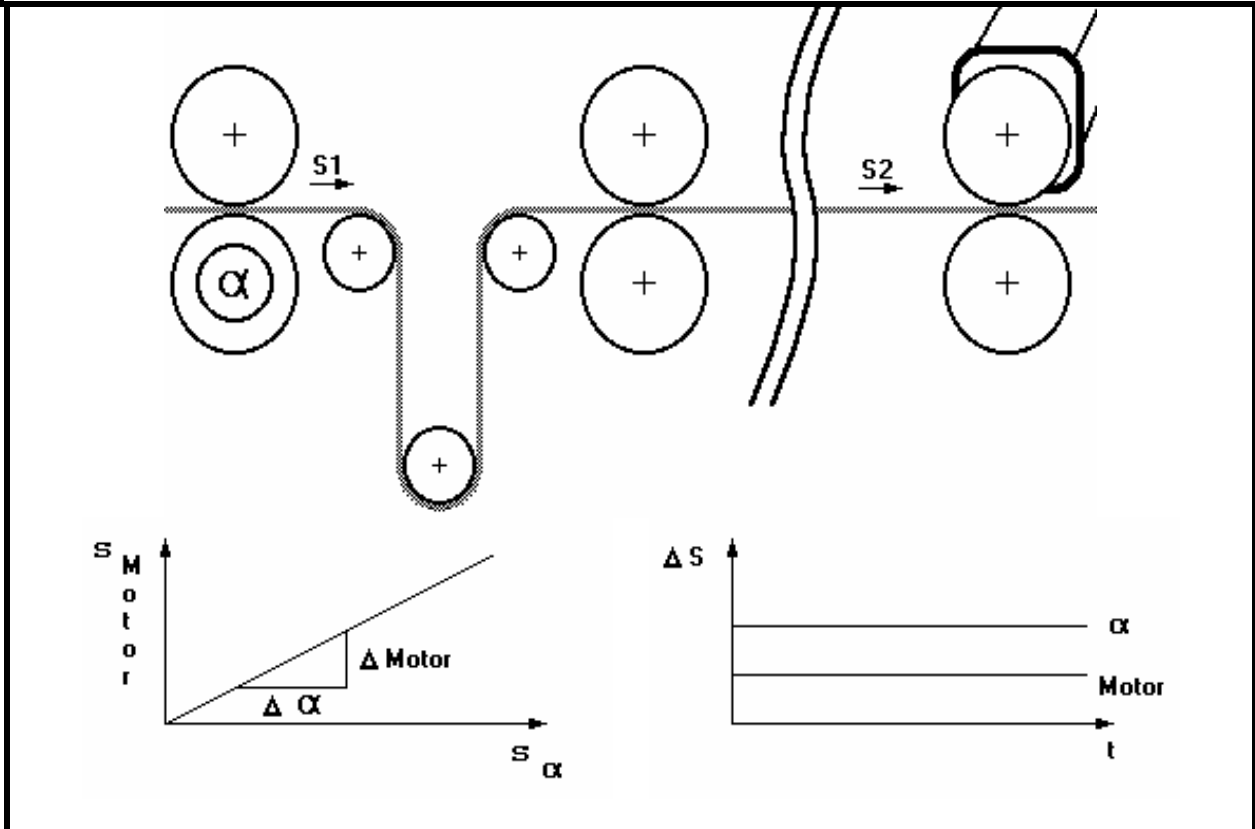
1. Positionssynchrones Fahren zum externen Inkrementalgeber (α)
solange der Eingang X10.11 aktiv (1) ist.

Dateiname:

BSP_004f.wbd

Voraussetzungen:

Betriebsart 5 ("Lageregelung mit BIAS-Abarbeitung")
Eingang X10.11 konfiguriert mit der Funktion 3 (BIAS-Eingang 11)
X40-Anschluß als Inkrementaleingang konfiguriert



Programmlisting

```

* 4. BIAS-Applikationsbeispiel: Elektronisches Getriebe 637f
* *****
* !!! Achtung !!!
* Notwendige E/A-Konfigurationen (BIAS-Definitionen Konfiguration)
*   Eingang X10.11 Funktion 3 (BIAS-Eingang 11)
*   X40-Modus 1 (Inkrementaleingang)
*   X40-Auflösung entsprechend dem externen Geber
* *****

ProgStart:
0 Istposition 1 = 0      INKR
1 Istposition 2 = 0      INKR
2 Modus 8 = 1
3 Unterprogramm Para_Sync
4 Lade Parametersatz 1 = [Variable [23 ]]*ZeigerVar10  *
5 Wenn Eingang 11 == 0 dann Springe ProgStart
6 Starte Achse
7 Fahre Synchron; AchsNr. = 1 , Ziel = 1 , Leitwert = 1
FAHRE:
8 Wenn Eingang 11 == 1 dann Springe FAHRE
9 Stoppe Achse; Modus = 1 ; AchsNr. = 1
10 Modus 8 = 0
11 Programmende, Modus = 0
* *****
*   Parametersatz des Typ 1 für einen Synchronlauf
* *****

Para_Sync:
12 [Variable 10 ] = 0      *Startwinkel  *
13 [Variable 11 ] = 0      *Totzeit   *
* *****

          Δs1 = Δs2
* Forderung:  Δt = Δt ; s1 = 8192; s2 = 4096; Koppelfaktor =  $\frac{s_2}{s_1} = \frac{4096}{8192} = 0,5$ 
* Übersetzungsverhältnis = 0,5
* Bei einer Geberaufösung von 4096 INKR pro Umdehung
14 [Variable 12 ] = 4096      *KopZaehler  *
15 [Variable 13 ] = 8192      *KopNenner   *
* *****

16 [Variable 14 ] = 10000      *MoveAcc     *
17 [Variable 15 ] = 10000      *MoveCont    *
18 [Variable 16 ] = 10000      *MoveDec     *
19 [Variable 17 ] = 10000      *MoveWait    *
20 [Variable 18 ] = 0          *CAM_Start_o *
21 [Variable 19 ] = 0          *Cam_Nr      *
22 [Variable 20 ] = 10000      *SkalZaehler *
23 [Variable 21 ] = 10000      *SkalNenner  *
24 [Variable 23 ] = 10         *ZeigerVar10 *
25 Merker 0 = 0                *Start_Nocke *
26 Merker 1 = 0                *Start_Richt *
27 Merker 2 = 0                *Gear_Trig_o *
28 Merker 3 = 0                *Backstop    *
29 Merker 4 = 0                *Clutch_on   *
30 Merker 5 = 0                *Ramp_Richtu *
31 Merker 6 = 0                *Ramp_Up     *
32 Merker 7 = 0                *Ramp_Down   *
33 Merker 8 = 0                *Ramp_Wait   *
34 Merker 9 = 0                *Move_Linked *
35 Merker 10 = 0               *Merge_Linke *
36 Merker 11 = 0               *Caming_on   *
37 Merker 12 = 0               *Stop_Cam    *
38 Merker 13 = 0               *Cam_Backsto *
39 Merker 14 = 0               *Res1        *
40 Merker 15 = 0               *Res2        *
41 [Variable 22 ] = Merker 0 , Anzahl = 16 *Bits, Start_Nocke*
42 Unterprogrammende

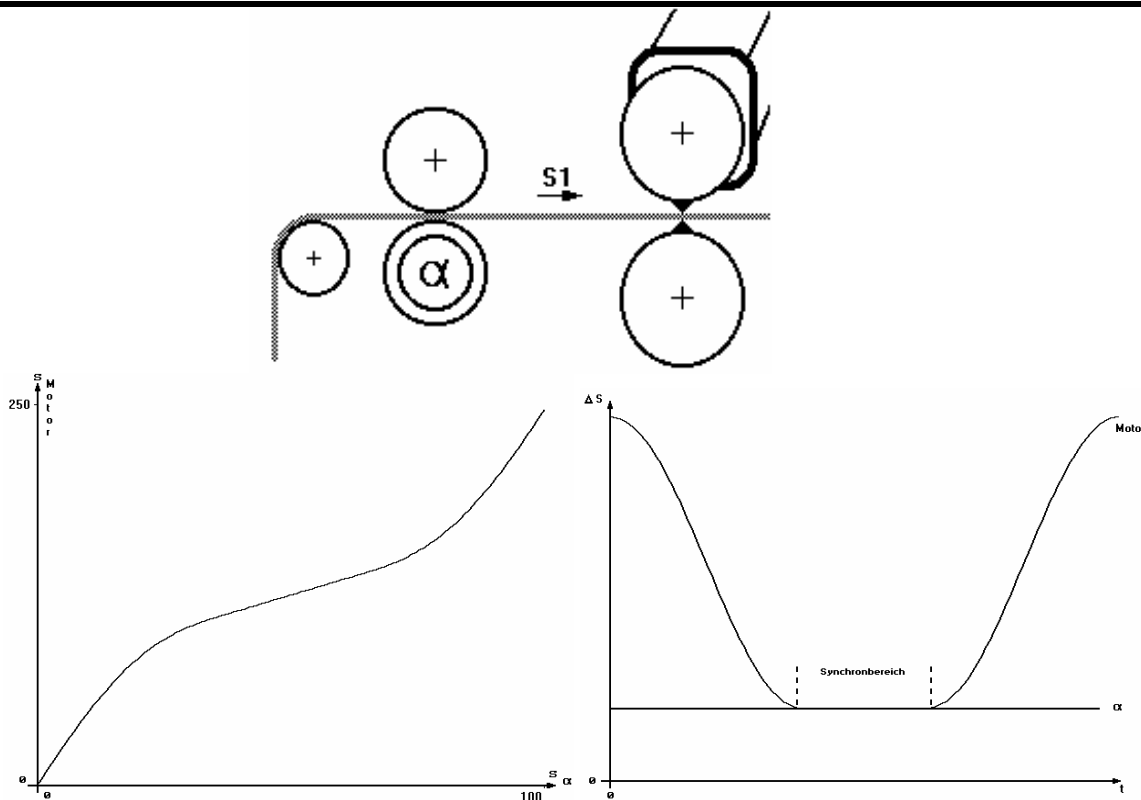
```

5.1 BIAS-Applikationsbeispiel: Elektronisches Kurvengetriebe 635/637

Erklärung: 1. Positionssynchrones Fahren zum externen Inkrementalgeber (α) solange der Eingang X10.11 aktiv (1) ist.

Dateiname: BSP_005. wbd

Voraussetzungen: Betriebsart 5 ("Lageregelung mit BIAS-Abarbeitung")
Eingang X10.11 konfiguriert mit der Funktion 3 (BIAS-Eingang 11)
X40-Anschluß als Inkrementaleingang konfiguriert



Programmlisting

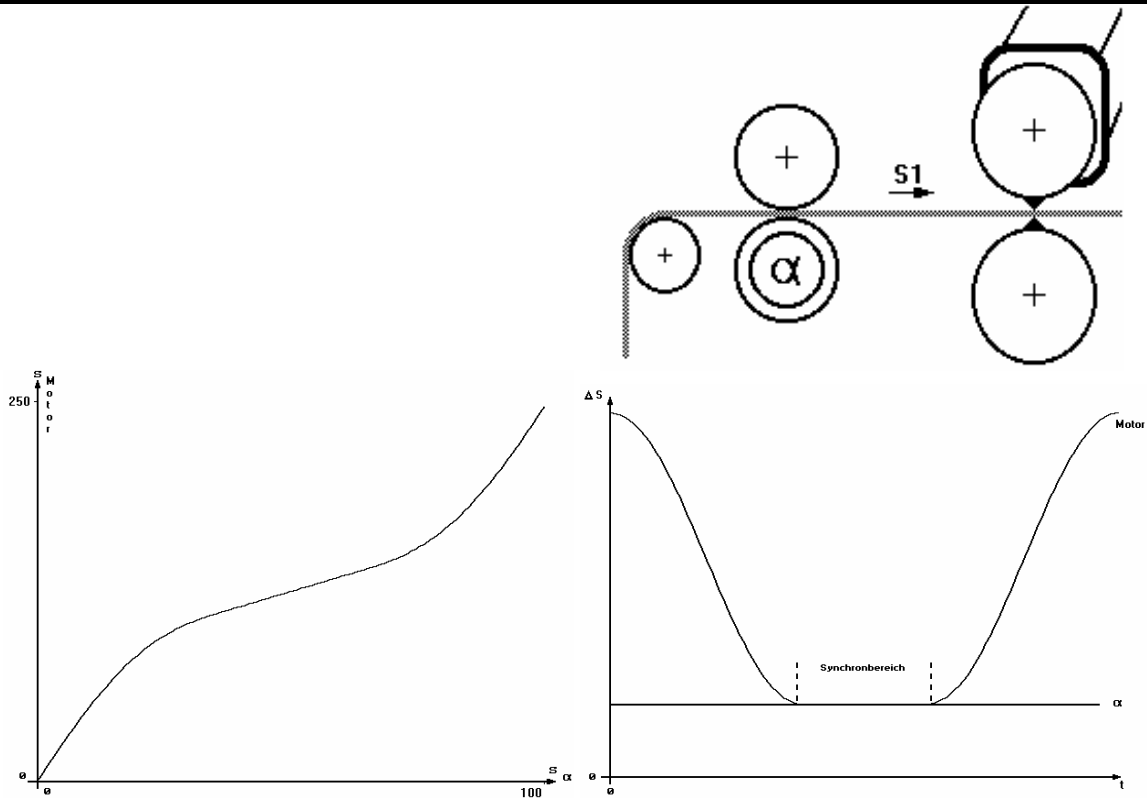
```
* *****
PROG_START:
0  Wenn Eingang 11 == 0 dann Springe PROG_START
1  Koppelfaktor = 1
2  Starte Achse
3  Fahre Synchronprofil 0
* *****
* Notwendige Eingaben für die Synchronprofilberechnung des Profils 0
* Synchronprofil Stanze = 6
* Inkrementkorrektur = N
* Mastertaktlänge = 100 mm
* Slavetaktlänge = 250 mm
* Anzahl der Stützstellen = 128
* Länge des Synchronbereichs = 25 mm
* *****
FAHRE:
4  Wenn Eingang 11 == 1 dann Springe FAHRE
5  Stoppe Achse; Modus = 1 a- = 5000 min /s
6  Programmende, Modus 0
```

5.2 BIAS-Applikationsbeispiel: Elektronisches Kurvengetriebe 631

Erklärung: 1. Positionssynchrones Fahren zum externen Inkrementalgeber (α) solange der Eingang X10.11 aktiv (1) ist.

Dateiname: B631_004. wbd

Voraussetzungen: Betriebsart 5 ("Lageregelung mit BIAS-Abarbeitung")
Eingang X10.8 konfiguriert mit der Funktion 3 (BIAS-Eingang 11)
X40-Anschluß als Inkrementaleingang konfiguriert



Programmlisting

```

* *****
PROG_START:
0 Wenn Eingang 11 == 0 dann Springe PROG_START
1 Koppelfaktor = 1
2 Starte Achse
3 Fahre Synchronprofil 0
* *****
* Notwendige Eingaben für die Synchronprofilberechnung des Profils 0
* Synchronprofil Stanze = 6
* Inkrementkorrektur = N
* Mastertaktlänge = 100 mm
* Slavetaktlänge = 250 mm
* Anzahl der Stützstellen = 128
* Länge des Synchronbereichs = 25 mm
* *****
FAHRE:
4 Wenn Eingang 8 == 1 dann Springe FAHRE
5 Stoppe Achse; Modus = 1 a- = 5000 min /s
6 Programmende, Modus 0

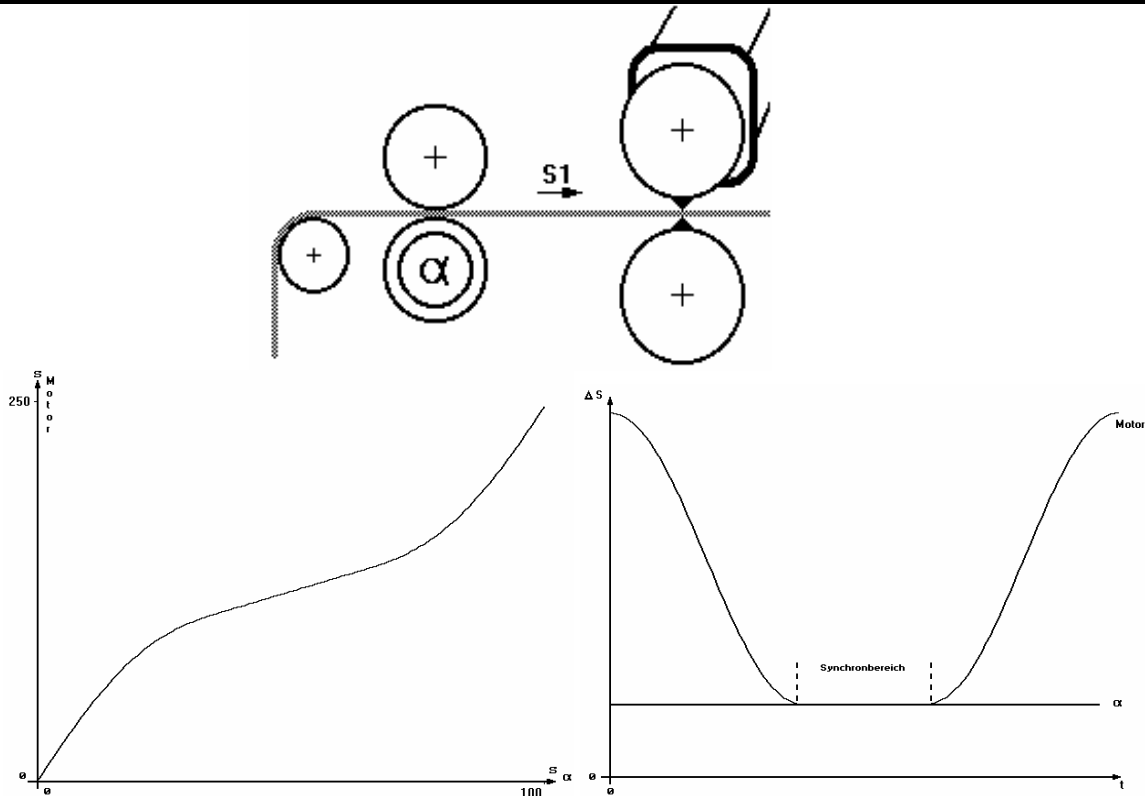
```

5.3 BIAS-Applikationsbeispiel: Elektronisches Kurvengetriebe 637f

Erklärung: 1. Positionssynchrones Fahren zum externen Inkrementalgeber (α) solange der Eingang X10.11 aktiv (1) ist.

Dateiname: BSP_005f.wbd

Voraussetzungen: Betriebsart 5 ("Lageregelung mit BIAS-Abarbeitung")
Eingang X10.11 konfiguriert mit der Funktion 3 (BIAS-Eingang 11)
X40-Anschluß als Inkrementaleingang konfiguriert



Programmlisting

```

* *****
*
* 5. BIAS-Applikationsbeispiel: Kurvengetriebe 637f
*
* *****
* !!! Achtung !!!
* Notwendige E/A-Konfigurationen (BIAS-Definitionen Konfiguration)
*   Eingang X10.11 Funktion 3 (BIAS-Eingang 11)
*   X40-Modus 1 (Inkrementaleingang)
*   X40-Auflösung entsprechend dem externen Geber
* *****
ProgStart:
0 Istposition 1 = 0      INKR
1 Istposition 2 = 0      INKR
2 Modus 8 = 1
3 Unterprogramm Para_Sync
4 Lade Parametersatz 1 = [Variable [23]]*ZeigerVar10      *
5 Wenn Eingang 11 == 0 dann Springe ProgStart
6 Starte Achse
7 Fahre Synchron; AchsNr. = 1 , Ziel = 1 , Leitwert = 1
FAHRE:
8 Wenn Eingang 11 == 1 dann Springe FAHRE
9 Stoppe Achse; Modus = 1 ; AchsNr. = 1
10 Modus 8 = 0
11 Programmende, Modus = 0
* *****

```

```

*
*   Parametersatz des Typ 1 für einen Synchronprofil 0
*
*   *****
Para_Sync:
12 [Variable 10 ] = 0                *Startwinkel      *
13 [Variable 11 ] = 0                *Totzeit          *
*   *****
*   Übersetzungsverhältnis = 1,0
*   Bei einer Geberaufösung von 4096 INKR pro Umdehung
*   *****
14 [Variable 12 ] = 4096              *KopZaehler       *
15 [Variable 13 ] = 4096              *KopNenner        *
16 [Variable 14 ] = 10000             *MoveAcc          *
17 [Variable 15 ] = 10000             *MoveCont         *
18 [Variable 16 ] = 10000             *MoveDec          *
19 [Variable 17 ] = 10000             *MoveWait         *
20 [Variable 18 ] = 0                 *CAM_Start_o     *
21 [Variable 19 ] = 0                 *Cam_Nr          *
22 [Variable 20 ] = 10000             *SkalZaehler      *
23 [Variable 21 ] = 10000             *SkalNenner       *
24 [Variable 23 ] = 10                *ZeigerVar10     *
25 Merker 0 = 0                       *Start_Nocke     *
26 Merker 1 = 0                       *Start_Richt     *
27 Merker 2 = 0                       *Gear_Trig_o     *
28 Merker 3 = 0                       *Backstop        *
29 Merker 4 = 0                       *Clutch_on       *
30 Merker 5 = 0                       *Ramp_Richtu     *
31 Merker 6 = 0                       *Ramp_Up         *
32 Merker 7 = 0                       *Ramp_Down       *
33 Merker 8 = 0                       *Ramp_Wait       *
34 Merker 9 = 0                       *Move_Linked     *
35 Merker 10 = 0                      *Merge_Linke     *
36 Merker 11 = 1                      *Caming_on       *
37 Merker 12 = 0                      *Stop_Cam        *
38 Merker 13 = 0                      *Cam_Backsto     *
39 Merker 14 = 0                      *Res1            *
40 Merker 15 = 0                      *Res2            *
41 [Variable 22 ] = Merker 0 , Anzahl = 16 *Bits, Start_Nocke*
42 Unterprogrammende

```



Weg =	[Variable X] = Weg	BIAS-Abarbeitungszeiger	[Variable X] =Merker Y	Profilwert = [Variable X]	Tabelle speichern	SPS-Programm
Der Befehl ist nur in der BIAS-Task erlaubt	Der Befehl ist in der BIAS, SPS und MATH-Task erlaubt	Der Befehl ist nur in der SPS und MATH-Task erlaubt	Der Befehl ist nur in der BIAS und SPS-Task erlaubt	Der Befehl ist nur in der MATH-Task erlaubt	Der Befehl ist nur in der SPS-Task erlaubt	Der Befehl ist nur in der BIAS und MATH-Task erlaubt

	0	1	2	3	4	5	6	7	8	9	A	B
0	Fahre_Position	Fahre_Position + Parameter	Weg = Konst.	Weg = [Variable X]	[Variable X] = Weg	NOP	Merker X = Konst.	Wenn Eingang X ? Konst.	[Variable X] = Konst.	Mathematik programm	Tabelle [Variable X] = Konst.	[D_Variable X] = [D_Variable Y] + [D_Variable Z]
1	Fahre Kettenposition	Fahre Kettenposition + Parameter	Geschwindigkeit = Konst.	Geschwindigkeit = [Variable X]	[Variable X] = Geschwindigkeit	Programmende	Wenn Merker X ? Konst.	Wenn Ausgang X ? Konst.	Wenn [Variable X] ? Konst.	Profilinitialisierung = Konst.	Tabelle [Variable X] = [Y_Variable Z]	[D_Variable X] = [D_Variable Y] - [D_Variable Z]
2	Fahre Referenz	Fahre Referenz + Parameter	Beschleunigung = Konst.	Beschleunigung = [Variable X]	[Variable X] = Beschleunigung	Unterprogramm	Merker X = Merker Y	Ausgang X = Konst.	[Variable X] = [Variable Y] + Konst.	Profilaktlänge = [Variable X]	[X_Variable Y] = Tabelle [Variable Z]	[D_Variable X] = [D_Variable Y] * [D_Variable Z]
3	Fahre unendlich Positiv	Fahre unendlich Positiv + Parameter	Verzögerung = Konst.	Verzögerung = [Variable X]	[Variable X] = Verzögerung	Unterprogrammende	Merker X = Eingang Y	Ausgang X = Merker Y	[Variable X] = [Variable Y] - Konst.	[Variable X] = Profilwert	[W_Variable X] = [Y_Variable Z]	[D_Variable X] = [D_Variable Y] / [D_Variable Z]
4	Fahre unendlich Negativ	Fahre unendlich Negativ + Parameter	Koppelfaktor = Konst.	Koppelfaktor = [Variable X]	[Variable X] = Koppelfaktor	SPS-Programm	Merker X = Ausgang Y	_____	[Variable X] = [Variable Y] * Konst.	Profilwert = [Variable X]	[X_Variable Y] = Konst.	Wenn [D_Variable X] ? [D_Variable Y]
5	Fahre Synchron	Fahre Synchron + Parameter	"Position erreicht" Fenster = Konst.	"Position erreicht" Fenster = [Variable X]	[Variable X] = Satznummer	Springe Konst.	Merker X = Merker Y & Merker Z	_____	[Variable X] = [Variable Y] / Konst.	_____	[Variable X] = Konst.	[D_Variable X] = SIN ([D_Variable Y])
6	Fahre Synchronprofil	Fahre Analogwert + Integrator	Restweg = Konst.	Restweg = [Variable X]	[Variable X] = Istposition Y	Springe [Variable X]	Merker X = Merker Y Merker Z	_____	[Variable X] = Merker Y	_____	[Variable X] = [Variable Y]	[D_Variable X] = COS ([D_Variable Y])
7	Synchroneinstellungen 1	Fahre Drehzahl + Integrator	Rampenfilter = Konst., [Variable X]	Maximalstrom = [Variable X]	[Variable X] = Analogeingang Y	BIAS-Abarbeitungszeiger = Konst.	Merker X = Merker Y ^ Merker Z	_____	[Variable X] = [Variable Y].Bit Z Anzahl	Tabelle speichern	[Variable X] = [Variable Y]	[D_Variable X] = SQRT ([D_Variable Y])
8	Synchroneinstellungen 2	_____	Istposition X = Konst.	Istposition X = [Variable Y]	[Variable X] = Latchposition Y	Warte auf "Position erreicht"	Merker X = ! Merker Y	IBT- Maskennummer = Konst.	[Variable X] = [Variable Y]	_____	[Variable X] = [Variable Y] ? [Variable Z]	_____
9	Fahre_PID : Drehzahl	_____	Wenn Istposition X ? Konst.	Analogausgang X = [Variable Y]	[Variable X] = Drehzahl Y	Wartezeit = Konst.	Merker X = Status Y	IBT- Meldungsnummer = Konst.	Wenn [Variable X] ? [Variable Y]	_____	[Variable X] = [Variable Y] ? Konst.	_____
A	Fahre_PID : Moment	Taktlänge =	Wenn Istposition X ? [Variable Y]	PID Skalierung	[Variable X] = Latchstatus Y	Wartezeit = [Variable X]	Wenn Status X ? Konst.	CAN Kommando = [Variable X]	[Variable X] = [Variable Y] + [Variable Z]	_____	_____	_____
B	Sollposition [AchsNr] = konst.	Taktlänge = [Variable X]	Sensorfenster = Konst.	Sensorfenster = [Variable X]	[Variable X] = Position Y; AchsNr.	BIAS-Abarb.zeiger = [Variable X]	Modus X = Konst.	IBT Datentransfer	[Variable X] = [Variable Y] - [Variable Z]	_____	_____	_____
C	Sollposition [AchsNr] = [Variable X]	Lade Parametersatz X = [Variable Y]	Sensorposition = Konst.	Sensorposition = [Variable X]	[Variable X] = Wert Y	Springe [Var.[X]]; Länge = Y : ab Zeile	Merker X = [Variable Y]	CAN2 Kommando = [Variable X]	[Variable X] = [Variable Y] * [Variable Z]	_____	_____	_____
D	Fahre relativ	_____	Sensoreinstellungen 1 = Konst.	Sensoreinstellungen 1 = [Variable X]	[Variable X] = Achsstatus, AchsNr. Y	Bearbeite X Befehle	[Variable X].Bit[Y] = Konst.	_____	[Variable X] = [Variable Y] / [Variable Z]	_____	_____	_____
E	Starte Achse	_____	Sensoreinstellungen 2 = Konst.	Sensoreinstellungen 2 = [Variable X]	_____	_____	Wenn[Var. X]. Bit Y == Konst. dann Springe	_____	[Teachvariable X] = [Variable Y]	_____	_____	_____
F	Stoppe Achse	Stoppe Achse + Parameter	Parameterübernahme	PID Parameter	_____	Virtuelles Programm	Achszustand, AchsNr. X, Bit Y = Konst., [Merker Z]	_____	[Variable X] = [Teachvariable Y]	_____	_____	_____

- Befehlsgruppe ["Fahrbefehle"](#)
- Befehlsgruppe ["Parameterbefehle"](#)
- Befehlsgruppe ["Variablenbefehle"](#)
- Befehlsgruppe ["Merkerbefehle"](#)
- Befehlsgruppe ["Bedingungsbehele"](#)

- Befehlsgruppe ["Programmsteuerbefehle"](#)
- Befehlsgruppe ["Mathematikbefehle"](#)
- Befehlsgruppe ["Ausgangsbefehle"](#)
- Befehlsgruppe ["CAN-Befehle"](#)
- Befehlsgruppe ["637 f Befehle"](#)

3.1 BIAS-Befehlsgruppe „Fahrbefehle“

Befehlsname	Nr.	Verwendbar in Programmzyklus...		
		BIAS	SPS	MATH
Fahrbefehle				
Fahre Kettenposition	01	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre PID, Drehzahl	09	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre PID, Moment	0A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre Position	00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre Referenz	02	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre relativ	0D	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre Synchron	05	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre Synchronprofil	06	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre unendlich Negativ	04	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre unendlich Positiv	03	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Starte Achse	0E	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Stoppe Achse	0F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Fahrbefehle mit Parametern				
Fahre Drehzahl + Integrator	17	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre Kettenposition + Parameter	11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre Position + Parameter	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre Referenz + Parameter	12	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre Synchron + Parameter	15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre unendlich mit Analogwert + Integrator	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre unendlich Negativ + Parameter	14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fahre unendlich Positiv + Parameter	13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Stoppe Achse + Parameter	1F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

3.2 BIAS-Befehlsgruppe „Parameterbefehle“

Befehlsname	Nr.	Verwendbar in Programmzyklus...		
		BIAS	SPS	MATH
Parameter = Konstante				
Beschleunigung = Konst.	22	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Geschwindigkeit = Konst.	21	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Istposition X = Konst.	28	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Koppelfaktor = Konst.	24	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
"Position erreicht" Fenster = Konst.	25	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Rampenfilter = Konst.: [Variable X]	27	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Restweg = Konst.	26	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sensoreinstellungen 1 = Konst.	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sensoreinstellungen 2 = Konst.	2E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sensorfenster = Konst.	2B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sensorposition = Konst.	2C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sollposition [Achsnr] = Konst.	0B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Synchronereinstellungen 1; Modus X	07	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Synchronereinstellungen 2; Linear X	08	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Taktlänge = Konst.	1A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Verzögerung = Konst.	23	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Weg = Konst.	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Parameterübernahme	2F	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Parameter = Variable				
Analogausgang X = [Variable Y]	39	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Beschleunigung = [Variable X]	32	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Geschwindigkeit = [Variable X]	31	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Istposition X = [Variable Y]	38	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Koppelfaktor = [Variable X]	34	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Lade Parametersatz X = [Variable [Y]]	1C	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Maximalstrom = [Variable X]	37	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PID Parameter	3F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PID Skalierung	3A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
"Position erreicht" Fenster = [Variable X]	35	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Restweg = [Variable X]	36	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sensoreinstellungen 1 = [Variable X]	3D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sensoreinstellungen 2 = [Variable X]	3E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sensorfenster; Modus = [Variable X]	3B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sensorposition = [Variable X]	3C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sollposition [Achsnr] = [Variable X]	0C	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Taktlänge = [Variable X]	1B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Verzögerung = [Variable X]	33	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Weg = [Variable X]	30	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Variable = Parameter

[Variable X] = Analogeingang Y	47	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Achsstatus, Achsnr. Y	4D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Beschleunigung	42	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Drehzahl Y	49	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Geschwindigkeit	41	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Istposition Y	46	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Koppelfaktor	44	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Latchposition Y	48	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Latchzustand Y	4A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Position Y	4B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Satznummer	45	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Verzögerung	43	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Weg	40	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = Wert Y	4C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

3.3 BIAS Befehlsgruppe "Variablenbefehle"

Befehlsname	Nr.	Verwendbar in Programmzyklus...		
		BIAS	SPS	MATH
Variablen-Befehle				
<u>[X Variable Y] = Konst.</u>	A4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Longinteger-Variablenbefehle				
<u>[Variable X] = [Variable Y].Bit Z, Anzahl</u>	87	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = [Variable Y] - [Variable Z]</u>	8B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = [Variable Y] - Konst.</u>	83	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable. X] = [Variable Y] * [Variable Z]</u>	8C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = [Variable Y] * Konst.</u>	84	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = [Variable Y] / [Variable Z]</u>	8D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = [Variable Y] / Konst.</u>	85	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = [Variable Y] + [Var. Z]</u>	8A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = [Variable Y] + Konst.</u>	82	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable [X]] = [Variable Y]</u>	A6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable [X]] = Konst.</u>	A5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = [Variable [Y]]</u>	A7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = [Variable Y]</u>	88	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = Merker Y, Anzahl Z</u>	86	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<u>[Variable X].BIT Y = Konst.</u>	6D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = Konst.</u>	80	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = [Teachvariable Y]</u>	8F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Teachvariable X]= [Variable Y]</u>	8E	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<u>[Variable X] = [Variable Y] ? [Variable Z]</u>	A8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>[Variable X] = [Variable Y] ? Konst</u>	A9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

3.4 BIAS Befehlsgruppe "Merkerbefehle"

Befehlsname	Nr.	Verwendbar in Programmzyklus...		
		BIAS	SPS	MATH
Merker-Befehle				
<u>Achsstaus, AchsNr. X, Bit Y = Konst., [Merker Z]</u>	6F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<u>Merker X = !Merker Y</u>	68	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Merker X = [Variable Y], Anzahl Z</u>	6C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<u>Merker X = Ausgang Y</u>	64	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Merker X = Eingang Y</u>	63	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Merker X = Merker Y & Merker Z</u>	65	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Merker X = Merker Y ^ Merker Z</u>	67	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Merker X = Merker Y Merker Z</u>	66	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Merker X = Merker Y</u>	62	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Merker X = Status Y</u>	69	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Merker X = Konst.</u>	60	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

3.5 BIAS Befehlsgruppe "Bedingungsbehele"

Befehlsname	Nr.	Verwendbar in Programmzyklus...		
		BIAS	SPS	MATH
Bedingungs-Befehle				
Wenn [D Var. X] ? [D Var. Y] dann Springe	B4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Wenn [Var. X] ? [Var. Y] dann Springe	89	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Wenn [Var. X] ? Konst. dann Springe	81	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Wenn Ausgang X ? Konst. dann Springe	71	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Wenn Eingang X ? Konst. dann Springe	70	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Wenn Istposition X ? Konst. dann Springe	29	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Wenn Istposition X [Var. Y] dann Springe	2A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Wenn Merker X ? dann Springe	61	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Wenn Status X ? Konst. dann Springe	6A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Wenn [Variable X].BIT Y == Konst.	6E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

3.6 BIAS Befehlsgruppe "Programmsteuerbefehle"

Befehlsname	Nr.	Verwendbar in Programmzyklus...		
		BIAS	SPS	MATH
Programmsteuer-Befehle				
Bearbeite X Befehle	5D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
BIAS-Abarbeitungszeiger = [Variable X]	5B	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
BIAS-Abarbeitungszeiger	57	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Mathematikprogramm	90	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modus X = Konst.	6B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NOP	50	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Programmende. Modus =	51	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Springe [Variable X]	56	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Springe [Variable [X]];Länge:ab Konst.	5C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Springe Konst.	55	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SPS-Programm	54	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Unterprogramm	52	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Unterprogrammende	53	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Virtuelles Programm	5F	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Warte auf "Position erreicht"	58	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Wartezeit = [Variable X]	5A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Wartezeit = Konst.	59	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3.7 BIAS Befehlsgruppe "Mathematikbefehle"

Befehlsname	Nr.	Verwendbar in Programmzyklus...		
		BIAS	SPS	MATH
Profile Befehle				
[Variable X] = Profilwert Y, Z	93	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[X Variable Y] = Konst.	A4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[X Variable Y] = Tabelle ([Variable Z])	A2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Profilinitialisierung = Konst.	91	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Profiltaktlänge = [Variable X]	92	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Profilwert Y, Z = [Variable X]	94	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Tabelle ([Variable X]) = [Y Variable Z]	A1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Tabelle ([Variable X]) = Konst.	A0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Tabelle speichern	97	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Fließkomma Variablen-Befehle				
[D Variable X] = [D Variable Y] - [D Variable Z]	B1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[D Variable X] = [D Variable Y] * [D Variable Z]	B2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[D Variable X] = [D Variable Y] / [D Variable Z]	B3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[D Variable X] = [D Variable Y] + [D Variable Z]	B0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[D Variable X] = COS {[D Variable Y]}	B6	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[D Variable X] = SIN {[D Variable Y]}	B5	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[D Variable X] = SQRT {[D Variable Y]}	B7	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Variablen Typenumwandlung				
[W Variable X] = [Y Variable Z]	A3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

3.8 BIAS Befehlsgruppe "Ausgangsbefehle"

Befehlsname	Nr.	Verwendbar in Programmzyklus...		
		BIAS	SPS	MATH
Ausgangsbefehle				
Ausgang X = Merker Y	73	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ausgang X = Konst.	72	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Analogausgang X = [Variable Y]	39	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

3.9 BIAS Befehlsgruppe "CAN-Befehle"

Befehlsname	Nr.	Verwendbar in Programmzyklus...		
		BIAS	SPS	MATH
CAN-Befehle				
CAN-Kommando = [Variable X]	7A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CAN2-Kommando = [Variable X]	7C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
IBT-Datentransfer	7B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
IBT-Maskennummer = X, LED Status [Var. Y]	78	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
IBT-Meldungsnummer = X, LED Status [Var. Y]	79	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

3.10 BIAS Befehlsgruppe "637f V8.xx Befehle"

Befehlsname	Nr.	Verwendbar in Programmzyklus...		
		BIAS	SPS	MATH
637f V8.xx-Befehle				
Achszustand, AchsNr. X, Bit Y = Konst, [Merker Z]	6F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Bearbeite X Befehle	5D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CAN2-Kommando = [Variable X]	7C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Fahre relativ, AchsNr. X, Ziel Y	0D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Lade Parametersatz X = [Variable [Y]]	1C	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sollposition [AchsNr] = Konst.	0B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sollposition [AchsNr] = [Variable X]	0C	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Springe [Var.[X]]; Länge=Y ; ab Z	5C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
[Variable X] = Achsstatus, AchsNr.Y	4D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X].BIT Y = Konst.	6D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = [Variable Y].BIT Z, Anzahl	87	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Wenn [Variable X].BIT Y == Konst.	6E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = [Variable Y] ? [Variable Z]	A8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable X] = [Variable Y] ? Konst.	A9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

4 - 00 BIAS-Befehl "Fahre Position"

Fahre Position; AchsNr. = X , Ziel = Y

Erklärung:

Der Befehl "Fahre Position" startet die Achse auf die zuletzt definierte Zielposition.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:
631/635/637/ 637+:	keine	-
637f mit Multi-Achsen-Funktion: (Modus 8 = 1)	X: Achsnummer Y: Ziel	1 - 3 1 = Sollwert in den Lageregler 2 = Sollwert auf CAN2-PDO3 4 = Sollwert auf CAN2-PDO4 8 = Sollwert auf X40-Ausgang Die Parallelschaltung der Sollwertziele ist durch Addition der Funktionscodierung möglich.

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Für die Fahrkurvengenerierung gelten immer die zuletzt eingestellten Parameter für: "Weg", Beschleunigung, Geschwindigkeit, Verzögerung, Rampenfilter

Beim Einschalten des Reglers sind diese Parameter mit den Einstellungen aus dem EASYRIDER Menü "Inbetriebnahme Lageregler" vorgeladen.

Bei der Verwendung der Multi-Achsfunktion beim Reglertyp 637f müssen die Parameter über den Befehl "Lade [Parametersatz 0 = \[Variable\[X\]\]](#)" vorgeladen werden.

Beispiel:

- 0 Weg = 1000 INKR
- 1 Starte Achse
- 2 Fahre Position; AchsNr. = 0 , Ziel = 0
- 3 Warte auf "Pos. erreicht"
- 4....

Ergebnis:

Die Achse fährt auf die Absolutposition 1000 Inkremente. Die BIAS-Abarbeitung wird mit dem Satz 4 nach dem Erreichen [der Position](#) fortgeführt.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
- **635/ 637/ 637+/ 637f**
Bei Verwendung des Starteingangs X10.11 ist in der "Eingangs-Konfiguration" die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
- **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- Die momentane Sollposition kann mit dem BIAS-Befehl [Variable [X](#)] = [Position 1](#) ermittelt werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
00	X	Y	00	00	00	00	00

Parameter: D1 = Achsnummer: 1...3
D2 = Ziel: 1...15

4 - 01 BIAS-Befehl "Fahre Kettenposition"

Fahre Kettenposition; AchsNr. = X , Ziel = Y

Erklärung:

Der Befehl "Fahre Kettenposition" startet die Achse auf die Position, die sich aus der Summe der aktuellen Zielposition plus der im Befehl "Weg" definierten relativen Distanz ergibt.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
631/635/637/637+:	keine	-	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
637f mit Multi-Achsen-Funktion: (Modus 8 = 1)	X: Achsnummer Y: Ziel	1 - 3 1 = Sollwert in den Lageregler 2 = Sollwert auf CAN2-PDO3 4 = Sollwert auf CAN2-PDO4 8 = Sollwert auf X40-Ausgang Die Parallelschaltung der Sollwertziele ist durch Addition der Funktionscodierung möglich	

Für die Fahrkurvengenerierung gelten immer die zuletzt eingestellten Parameter für: "Weg", Beschleunigung, Geschwindigkeit, Verzögerung, Rampenfilter

Beim Einschalten des Reglers sind diese Parameter mit den Einstellungen aus dem EASYRIDER Menü "Inbetriebnahme Lageregler" vorgeladen.

Bei der Verwendung der Multi-Achsfunktion beim Reglertyp 637f müssen die Parameter über den Befehl "Lade [Parametersatz 0 = \[Variable\[X\]\]](#)" vorgeladen werden.

Beispiel:

0 Weg = 1000 INKR

1 Starte Achse

2 Fahre Kettenposition, AchsNr. = 0, Ziel 0

3 Warte auf "Pos. erreicht"

4

Ergebnis:

Die Achse fährt von der momentanen Position 1000 Inkremente weiter.

Die BIAS-Abarbeitung wird mit dem Satz 4 nach dem Erreichen [der Position](#) fortgeführt.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/ 637+/ 637f**
Bei Verwendung des Starteingangs X10.11 ist in der Konfiguration die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- Die momentane Sollposition kann mit dem BIAS-Befehl [Variable [X1](#) = [Position 1](#)] ermittelt werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
01	X	Y	00	00	00	00	00

Parameter: D1 = Achsnummer: 1...3
D2 = Ziel: 1...15

4 - 02 BIAS-Befehl "Fahre Referenz"

Fahre Referenz, Modus= X , [Variable Y]

Erklärung:

Der Befehl "Fahre Referenz" startet die Referenzfahrt der Achse entsprechend dem angegebenen Referenzmodus.

631 [Beispiel laden](#)

635/637 [Beispiel laden](#)

637+ [Beispiel laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
631/635/637/ 637+:	X: Modus	Modus: 0...23	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
637f:	X: Modus Y: Variable	Modus: 0...23 , 24 oder 128, 192 0 -255	

Für die Fahrkurvengenerierung gelten immer die zuletzt eingestellten Parameter für: "Weg", Beschleunigung, Geschwindigkeit, Verzögerung, Rampenfilter

Beim Einschalten des Reglers sind diese Parameter mit den Einstellungen aus dem EASYRIDER Menü "Inbetriebnahme Lageregler" vorgeladen.

Beispiel:

0 Weg = 4000 INKR

1 Starte Achse

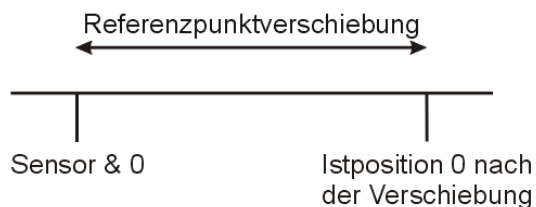
2 Fahre Referenz, Modus 16

3 Warte auf "Pos. erreicht"

4

Ergebnis:

Die Achse referiert auf die der Low-High-Flanke des Referenzsensor folgenden Gebernullstellung in positiver Richtung (Sensor & 0-Punkt).



Von dieser Position aus wird eine Referenzpunktverschiebung um +4000 Inkremente vorgenommen.

Hinweis:

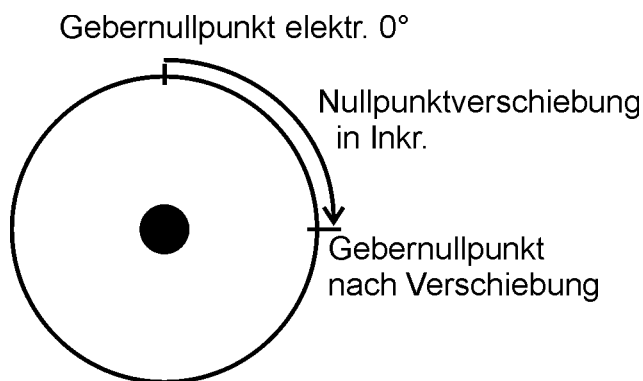
- Geber können Resolver oder HIPERFACE sein mit ihrem entsprechenden Nullpunkt bei Regelung auf Istposition 2 wird der Nullimpuls des externen Geber ausgewertet.
- Für die Modi 0 -11 muss der Parameter "Weg" auf Null gesetzt werden.
- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/ 637+/ 637f**
Bei Verwendung des Starteingangs X10.11 ist in der Konfiguration die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- **635/ 637/ 637+/ 637f**
Bei der Benutzung eines Modus mit Referenzsensor X10.24 ist in der "Eingangs-Konfiguration" die Funktion 1 (Referenzsensor) zu selektieren.

- **637f**
Ist der 637f Regler mit der Optionsplatine C28 bestückt kann optional der Eingang X120.4 als Referenzsensor konfiguriert werden. Der Referenzsensoreingang am X10 Stecker ist dann nicht in Funktion.

Modus = 24, die Referenzkennung (und der Referiert-Ausgang) werden ohne Positionierung gesetzt.

Modus = 128, der Modus wird aus dem Inhalt der angegebenen Variablen geladen.

Modus =192, der Inhalt der angegebenen Variable wird als Gebernullpunktverschiebung geladen.
Achtung: Dieser Modus startet **nicht** die Achse. Die Verschiebung wird erst nach einem erneutem Referenzfahrbefehl mit Gebernullpunktmodus mit ausgewertet.



- **631**
Bei der Benutzung eines Modus mit Referenzsensor muss für einen der Eingänge (X10.8, X10.9 oder X10.10) in der Konfiguration die Funktion 1 (Referenzsensor) selektiert sein.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
02	X	Y	00	00	00	00	00

Parameter: D1 = Referenzmodus: 0...24, 128, 192
D2 = Variable 0-255

4 - 03 BIAS-Befehl "Fahre unendlich Positiv"

Fahre unendlich Positiv; AchsNr. = X, Ziel = Y

Erklärung:

Der Befehl "Fahre unendlich Positiv" startet in positiver Richtung auf die aktuell definierte Geschwindigkeit.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
631/635/637/637+:	keine	-	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
637f mit Multi-Achsen-Funktion: (Modus 8 = 1)	X: Achsnummer Y: Ziel	1 - 3 1 = Sollwert in den Lageregler 2 = Sollwert auf CAN2-PDO3 4 = Sollwert auf CAN2-PDO4 8 = Sollwert auf X40-Ausgang Die Parallelschaltung der Sollwertziele ist durch Addition der Funktionscodierung möglich.	

Für die Fahrkurvengenerierung gelten immer die zuletzt eingestellten Parameter für: "Weg", Beschleunigung, Geschwindigkeit, Verzögerung, Rampenfilter

Beim Einschalten des Reglers sind diese Parameter mit den Einstellungen aus dem EASYRIDER Menü "Inbetriebnahme Lageregler" vorgeladen. Bei der Verwendung der Multi-Achsfunktion beim Reglertyp 637f müssen die Parameter über den Befehl "Lade [Parametersatz 0 = \[Variable\[X\]\]](#)" vorgeladen werden.

Beispiel:

0 Geschwindigkeit = 1000 rpm
1 Starte Achse
2 Fahre unendlich Positiv, AchsNr. = 0 , Ziel = 0
3 Warte Zeit 1000 ms
4 Stoppe Achse ; Modus 1
5

Ergebnis:

Die Achse wird 1 Sekunde lang in positiver Richtung verfahren.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/637+/- 637f**
Bei Verwendung des Starteingangs X10.11 ist in der "Eingangs-Konfiguration" die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- Geschwindigkeitsänderungen können während der Fahrt über den Befehl "Parameterübernahme" vorgenommen werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
03	X	Y	00	00	00	00	00

Parameter: D1 = Achsnummer: 1...3
D2 = Ziel: 1...15

4 - 04 BIAS-Befehl "Fahre unendlich Negativ"

Fahre unendlich Negativ; AchsNr. = X, Ziel = Y

Erklärung:

Der Befehl "Fahre unendlich Negativ" startet in negativer Richtung auf die aktuell definierte Geschwindigkeit.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
631/635/637/ 637+:	keine	-	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
637f mit Multi-Achsen-Funktion (Modus 8 = 1)	X: Achsnummer Y: Ziel	1 - 3 1 = Sollwert in den Lageregler 2 = Sollwert auf CAN2-PDO3 4 = Sollwert auf CAN2-PDO4 8 = Sollwert auf X40-Ausgang Die Parallelschaltung der Sollwertziele ist durch Addition der Funktionscodierung möglich.	

Für die Fahrkurvengenerierung gelten immer die zuletzt eingestellten Parameter für: Beschleunigung, Geschwindigkeit, Verzögerung, Rampenfilter

Beim Einschalten des Reglers sind diese Parameter mit den Einstellungen aus dem EASYRIDER Menü "Inbetriebnahme Lageregler" vorgeladen.

Bei der Verwendung der Multi-Achsfunktion beim Reglertyp 637f müssen die Parameter über den Befehl "Lade [Parametersatz 0 = \[Variable\[X\]\]](#)" vorgeladen werden.

Beispiel:

0 Geschwindigkeit = 1000 rpm

1 Starte Achse

2 Fahre unendlich Negativ, AchsNr = 0, Ziel = 0

3 Warte Zeit 1000 ms

4 Stoppe Achse ; Modus 1

5

Ergebnis:

Die Achse wird 1 Sekunde lang in negativer Richtung verfahren.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/ 637+/ 637f**
Bei Verwendung des Starteingangs X10.11 ist in der "Eingangs-Konfiguration" die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- Geschwindigkeitsänderungen können während der Fahrt über den Befehl "Parameterübernahme" vorgenommen werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
04	X	Y	00	00	00	00	00

Parameter: D1 = Achsnummer: 1...3
D2 = Ziel: 1...15

4 - 05 BIAS-Befehl "Fahre Synchron"

Fahre Synchron; AchsNr. = X , Ziel = Y , Leitwert = Z

Erklärung:

Der Befehl "Fahre Synchron" startet die Achse mit dem eingestellten Koppelfaktor synchron zu den Masterinkrementen.

[Beispiel laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
631/635/637/ 637+:	keine	-	
637f mit Multi-Achsen-Funktion: (Modus 8 = 1)	X: Achsnummer Y: Ziel Z: Leitwert	1 - 3 1 = Sollwert in den Lageregler 2 = Sollwert auf CAN2-PDO3 4 = Sollwert auf CAN2-PDO4 8 = Sollwert auf X40-Ausgang Die Parallelschaltung der Sollwertziele ist durch Addition der Funktionscodierung möglich. 1 = X40 Istposition 2 2 = Istposition 3 (CAN2-SSI-PDO1) 4 = Istposition 4 CAN2-PDO3 8 = Istposition 5 CAN2-PDO4 16 = Sollposition von Achse 1 32 = Sollposition von Achse 2 64 = Sollposition von Achse 3 Die Addition der Leitwerte ist durch Addition der Funktionscodierung möglich. Achtung: Es darf nicht die Sollposition der Achse, die als Achsnummer X angewählt wurde, als Leitwert aufgeschaltet werden.	

Bei der Verwendung der Multi-Achsfunktion beim Reglertyp 637f müssen die Parameter über den Befehl "Lade [Parametersatz X = \[Variable\[Y\]\]](#)" vorgeladen werden.

Weitere [Informationen über Synchronfunktionen der 630 Serie](#)

Beispiel:

0 Taktlänge = 0 INKR

1 Koppelfaktor = 1.0

2 Fahre Synchron; AchsNr. = 0 , Ziel = 0 , Leitwert = 0

3 Wenn Eingang 11 == 1 dann Springe 3

4 Stoppe Achse ; Modus 1

5 ...

Ergebnis:

Nach der Low-High-Flanke des Starteingangs fährt die Achse synchron (1 Masterinkrement am X40 Eingang = 1 Slave Inkrement), bis der Starteingang 0 wird.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte Achse"
optional : Sensor; Modus=128
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/ 637+/ 637f**
Bei Verwendung des Starteingangs X10.11 ist in der "Eingangs-Konfiguration" die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- Der X40-Modus ist in der Zählerkonfiguration als Impulseingang zu selektieren.
- Die Veränderung des Übersetzungsverhältnisses ist während der Synchronfahrt über den Befehl "Koppelfaktor =" möglich.
- Der vorangestellte Befehl "Taktlänge =" definiert die Formatlänge für Sonderfunktionen(Auf-, Absynchronisieren, Offset, Sensorkorrektur). Mit Taktlänge=0 sind diese Funktionen nicht aktiv.
- Die Leitwerte werden als Positionsdifferenzen vorgegeben.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
05	X	Y	Z	00	00	00	00

Parameter: D1 = Achsnummer: 1...3
 D2 = Ziel: 1...15
 D3 = Leitwert: 1...127

4 - 06 BIAS-Befehl "Fahre Synchronprofil"

Fahre Synchronprofil X

Erklärung:

Der Befehl "Fahre Synchronprofil" startet die Achse mit der angegebenen Profilnummer synchron zu den Masterinkrementen.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle	Profilnummer	0-15	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Weitere [Informationen über Synchronprofilfunktionen der 630 Serie](#)

Beispiel:

0 Koppelfaktor = 1.0

1 Fahre Synchronprofil 0

2 Wenn Eingang 11 == 1 dann Springe 2

3 Stoppe Achse; Modus 1

4

Ergebnis:

Das im Synchronprofil 0 berechnete Profil wird solange abgearbeitet, bis der Starteingang 11 (9 bei 631) 0 wird.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/637+/637f**
Bei Verwendung des Starteingangs X10.11 ist in der "Eingangs-Konfiguration" die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- Der X40-Modus ist in der Zählerkonfiguration als Impulseingang zu selektieren.
- Vor dem Aktivieren des Befehls muß das zugehörige Synchronprofil berechnet worden sein (siehe "Programm", "Synchronprofilverwaltung").
- Der vorangestellte Befehl "Koppelfaktor = " definiert eine Skalierung der Mastertaktlänge (resultierende Mastertaktlänge = berechnete Mastertaktlänge * Koppelfaktor).

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
06	X	00	00	00	00	00	00

Parameter: D1 = Profilnummer 0...15

4 - 07 BIAS-Befehl "Synchronereinstellungen 1"

Synch.einstellungen 1; Modus=X; Offset=[Var.Y]; Startoffset=[Var.Z]

Erklärung:

Mit dem Befehl "Synchronereinstellungen 1" ist es möglich, durch die Einstellung eines Modus und/oder der Definition eines Offset in der Variablen X auf einen laufenden Synchron-Fahrbehl einzuwirken.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle	X: Modus Y: Offsetvariable Z: Startoffsetvar.	0...255 0...255 0...255	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Weitere [Informationen über Synchronfunktionen der 630 Serie](#)

Beispiel:

0 Synch.einstel.; Modus=0 ; Offset=[Var.0]; Startoffset=[Var.0]

Ergebnis:

Mit dem Parameter **Modus** sind folgende Funktionen aktivierbar:

- 0 keine Funktion
- 1 Abbremsen auf Geschwindigkeit 0 innerhalb der halben Formatlänge.
- 2 Anfahren auf Synchrongeschwindigkeit innerhalb der halben Formatlänge.
- 3 reserviert (kundenspezifische Sonderfunktion)

Die Reaktion auf eine Offsetveränderung kann über folgende Kombinationen modifiziert werden:

Modus + 32 **Offsetwert** wird direkt geladen (nicht erst am Formattrigger)

Modus + 64 Offsetwert wird als Offset2 (Profilverschiebung auf der Masterachse) geladen

Modus + 128 **Start** der Synchronfahrt an der im Parameter "Startoffset" angegebenen Masterposition

Der Parameter **Offset** definiert die Variable, deren Inhalt dem Synchronlauf als absolute Verschiebung auf der Slaveachse (Standard) oder der Masterachse (Modus + 64) aufgeschaltet wird.

Der Parameter **Startoffset** definiert die Variable, deren Inhalt den Startpunkt für die Modi 1 und 2 gegenüber dem Formatbeginn verschiebt (Standard) oder der die Position auf der Masterachse definiert an der der Synchronlauf gestartet wird (Modus + 128).

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
07	X	Y	Z	00	00	00	00

Parameter: D1 = Modus: 0..255
 D2 = Variablenindex für Offsetwert: 0..255
 D3 = Variablenindex für Startoffsetwert: 0..255

4 - 08 BIAS-Befehl "Synchronereinstellungen 2"

Synch.einstellungen 2; Linear= X ,Modus= Y ,Wert = Z

Erklärung:

Mit dem Befehl "Synchronereinstellungen 2" ist es möglich, das Löschen des Zählers beim Einschalten des Synchronlaufs zu unterdrücken (Linearachse).

Über die Parameter Modus und Wert können verschiedene Einstellungen des Synchronlaufs vorgenommen werden.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle	X: Linear Y: <u>Modus</u> Z: <u>Wert</u>	0/1 (1= Zähler nicht löschen) 0...255 0...65535	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

0 Synch.einstellungen 2; Linear= 0 ,Modus= 0 ,Wert = 0

Ergebnis:

Modus = 1 Wert = 0 ..500 Verschiebung auf der Slaveachse, Offsetkorrekturgeschwindigkeit als %-Wert der aktuellen Synchrongeschwindigkeit

Modus = 2 Wert = 0 Verschiebung auf der Slaveachse, Offsetkorrekturgeschwindigkeit als Inkremente/Abtastzyklus.

Hinweis:

Ist keiner der beiden Modi gesetzt oder der gesetzte Wert = 0, dann wird die Standard-Offsetkorrekturgeschwindigkeit von Offset/256 benutzt.

Dies bedeutet, daß der Offset innerhalb von 512 ms korrigiert wird.

Modus = 3	Wert = 0 Wert = 1	Bidirektionale Bewegungen der Achse erlaubt Bewegungen nur in positiver Zählrichtung der Achse erlaubt
Modus = 4	Wert = 0 Wert = 1	Rampenfilter beeinflusst alle Positionierungen Rampenfilter beeinflusst nur die Offsetkorrektur der Achse während des Synchronlaufs.
Modus = 5	Wert = 0 ..500	Verschiebung auf der Masterachse, Offsetkorrekturgeschwindigkeit als %-Wert der aktuellen Synchrongeschwindigkeit
Modus = 6	Wert = 0	Verschiebung auf der Masterachse, Offsetkorrekturgeschwindigkeit als Inkremente/Abtastzyklus.
Modus = 7	Wert = 0 Wert = 1	positive Synchronstart Richtung negative Synchronstart Richtung
Modus = 8	Wert = 0 Wert = 1 Wert = 2	Koppelfaktor mit 1/256 Standardgenauigkeit Koppelfaktor mit 1/65536 Genauigkeit Der Koppelfaktor wird definiert mit dem Befehl Koppelfaktor = [Variable(x)] als Bruch zwischen dem High und Low word der Variable (X)

Beispiel:
 0 [Variable 0] = 3
 1 [Variable 1] = 1
 2 [Variable 3] = [Variable 1] * 65536
 3 [Variable 2] = [Variable 0] + [Variable 3]
 4 Synch.einstellungen 2; Linear= 1,Modus= 8 ,Wert = 2
 5 Koppelfaktor = [Variable 2] definiert den Koppelfaktor von 1/3.

- Modus = 9** Wert = 0 Führt am Ende des Synchronprofils mit dem aktuell definierten Profil fort.
Nur 637f
- Wert = 1 Beendet den Synchronlauf am Ende des Synchronprofils.
- Modus = 10** Wert = 1 Ist diese Kennung gesetzt, werden bei einer Unterbrechung (Stop, Positionierung, Regler deaktiv) des Synchronlaufs alle für die Synchronisierung relevanten Werte und Parameter eingefroren und bei einem Neustart des Synchronlaufs wieder aktiviert. Das Wiedereinschalten des Synchronlaufs erfolgt inkrementell, d.h. ohne Positionssprung beim Slave.
Nur 637f
- Während der Unterbrechung einlaufende Masterinkremente werden ignoriert.
- Wert = 0 Soll der Synchronlauf wieder vom Anfang gestartet werden, muss die „Einfrier Kennung“ zurückgenommen werden
- Modus = 11** Wert = 1 Verhindert das Überschreiben der Mastertaktlänge, wenn aus einem Synchronlauf ein absoluter oder relativer Fahrbefehl gestartet wird. (Da der Befehl "Weg=" die Taktlänge überschreiben würde.)
Nur 631/635/637
- Modus = 12** Wert = 0 Keine Ausgangsskalierungsänderung während des Betriebs möglich.
Nur 637f
- Wert = 1 Die Ausgangsskalierung (Slaveinkremente) des CAM-Profiles ist mit einem Bruch einstellbar.
Variable 250 = Zähler
Variable 251 = Nenner
- Modus = 13** Wert = 0 Das Rückwärtsdrehen des Master über einen Formatwechsel wird nicht überwacht.
Nur 637f
- Wert = 1 Rückwärtsdrehen bis zum Formatwechsel überwachen bzw. ermöglichen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
08	X	Y	Z(low)	Z(high)	00	00	00

Parameter: D1 = Linear: 0, 1
D2 = Modus: 0-255
D3, D4 = Wert: 0 - 65535

4 - 09 BIAS-Befehl "Fahre PID, Drehzahl"

Fahre PID; Drehzahl; Sollwert= X , Istwert=Y , Modus=Z

Erklärung:

Der Befehl "Fahre PID, Drehzahl" ermöglicht die Regelung der Istdrehzahl entsprechend der Regeldifferenz zwischen Soll- und Istwert über die eingestellten Parameter des PID Reglers.

Die Parameter des PID Reglers werden mit den Befehlen "PID [Skalierung](#)" und "PID [Parameter](#)" eingestellt.

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle	X: Sollwert Y: Istwert Z: Modus	0...4 0...4 0...255	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Weitere [Informationen über den PID-Regler der 630 Serie](#)

Beispiel:

PROG_START:

0 [Variable 0] = 200
 1 [Variable 1] = 100
 2 [Variable 2] = 3
 3 [Variable 3] = 0
 4 [Variable 4] = 500
 5 [Variable 5] = -1000
 6 PID Parameter ; Start=[Variable 0]
 7 [Variable 6] = 2500
 8 [Variable 7] = 2500
 9 [Variable 8] = -2500
 10 [Variable 9] = 2500
 11 [Variable 10] = 18000
 12 [Variable 11] = 100
 13 PID Skalierung; Start=[Variable 6]
 * Start execution PID , Output = speed Setpoint
 14 Starte Achse
**15 Fahre PID; Drehzahl; Sollwert= 4 , Istwert=0,
 Modus=7**

Loop:

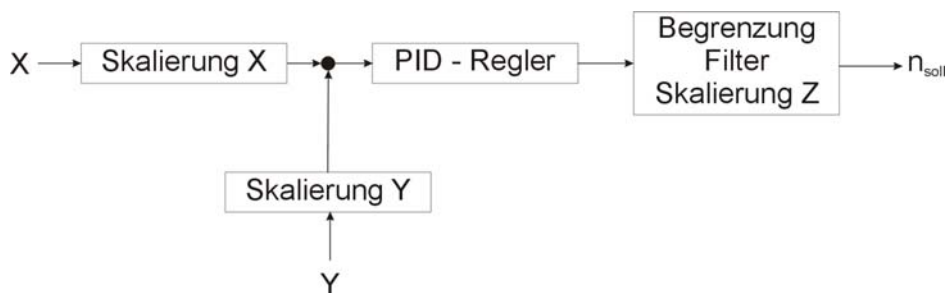
16 Springe Loop

Ergebnis:

* P-Gain $200/4096=0.048$
 * Integration-Time $100*1.89 \text{ msec}=189 \text{ msec}$
 * Derivative Time $3*1,89\text{msec}=5,67\text{mSec}$
 * Filtertimeconstant
 * PID Output limit high
 * PID Output limit low
 * load PID Parameters
 * Scaling Setpoint numerator
 * Scaling Setpoint denominator
 * Scaling actual value numerator
 * Scaling actual value denominator
 * Scaling PID Output numerator
 * Scaling PID Output denominator
 * Overtake Scalings
 * Setpoint =Variable 254 =0, Actual value
 =ana.in.1,PID enable,I enable, Feed Forward on.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/637+/637f**
Bei Verwendung des Starteingangs X10.11 ist in der Konfiguration die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.



Sollwert X Erklärung

0	Analogeingang 1 ($\pm 10V = \pm 2047$)	
1	Analogeingang 2 ($\pm 10V = \pm 2047$)	
2	Istgeschwindigkeit 1 intern	
3	631/635/637	637+/637f
	Istgeschwindigkeit 2 (Inkremente /1,899 ms)	Istgeschwindigkeit 2 (Inkremente /0,844 ms)
4	BIAS-Variable 254	

Istwert Y Erklärung

0	Analogeingang 1 ($\pm 10V = \pm 2047$)	
1	Analogeingang 2 ($\pm 10V = \pm 2047$)	
2	Istgeschwindigkeit 1 intern	
3	631/635/637	637+/637f
	Istgeschwindigkeit 2 (Inkremente /1,899 ms)	Istgeschwindigkeit 2 (Inkremente /0,844 ms)
4	BIAS-Variable 255	

Modus: Der Parameter Modus enthält mehrere bitkodierte Schalterfunktionen.

Bit 0:	PID-Freigabe,	0 = PID-Regler deaktiviert (alle Anteile = 0) 1 = PID-Regler aktiviert
Bit 1:	I-Anteil-Freigabe	0 = I-Anteil gesperrt 1 = I-Anteil aktiv
Bit 2:	Vorsteuerungsfreigabe	1 = Vorsteuerung aktiviert
Bit 3:	Vorsteuerungsauswahl	0 = Vorsteuerung standard 1 = Vorsteuerung mit Var. 253
Bit 4:	Integrator-Anteil stoppen	0 = Integrator läuft 1 = Integrator stoppen (der momentan aktuelle Integratorwert bleibt erhalten)
Bit 5...6:	Reserviert	immer 0
Bit 7:	Reiner Analoges Drehzahlregler (ohne PID-Funktion)	0 = Standard PID Funktion 1 = reiner analoger Drehzahlregler

Befehlskodierung:

Code **D1**

09 **X**

D	D	D	D	D	D
2	3	4	5	6	7
Y	Z	00	00	00	00

Parameter: D1 = Sollwert (0-4)

D2 = Istwert (0-4)

D3 = Modus (0 – 255)

4 - 0A BIAS-Befehl "Fahre PID, Moment"

Fahre PID; Moment ; Sollwert= X , Istwert=Y , Modus=Z

Erklärung:

Der Befehl "Fahre PID, Moment" ermöglicht die Regelung der Iststromes entsprechend der Regeldifferenz zwischen Soll (X)- und Istwert (Y) über die eingestellten Parameter des PID Reglers.

Die Parameter des PID Reglers werden mit den Befehlen "PID [Skalierung](#)" und "PID [Parameter](#)" eingestellt.

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle	X: Sollwert Y: Istwert Z: Modus	0...4 0...4 0...255	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Weitere [Informationen über den PID-Regler der 630 Serie](#)

Beispiel:

PROG_START:

0 [Variable 0] = 200
 1 [Variable 1] = 100
 2 [Variable 2] = 3
 3 [Variable 3] = 0
 4 [Variable 4] = 500
 5 [Variable 5] = -1000
 6 PID Parameter ; Start=[Variable 0]
 7 [Variable 6] = 2500
 8 [Variable 7] = 2500
 9 [Variable 8] = -2500
 10 [Variable 9] = 2500
 11 [Variable 10] = 18000
 12 [Variable 11] = 100
 13 PID Skalierung; Start=[Variable 6]
 * Start execution PID , Output = speed Setpoint

14 Starte Achse

15 Fahre PID; Moment; Sollwert= 4 , Istwert=0 , Modus=7 * Setpoint =Variable 254 =0, Actual value

Loop:

16 Springe Loop

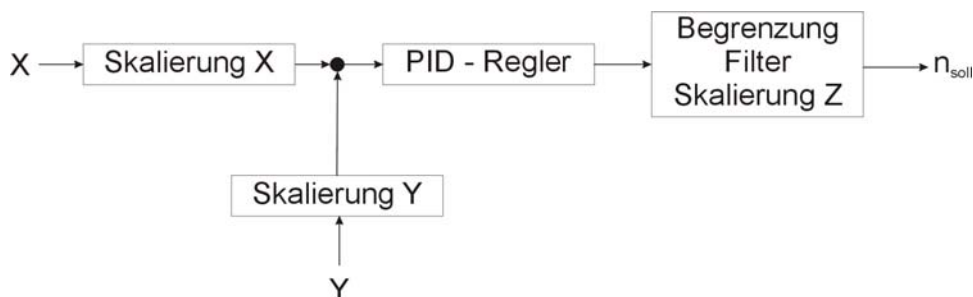
Ergebnis:

* P-Gain 200/4096=0.048
 * Integration-Time 100*1.89 msec=189 msec
 * Derivative Time 3*1,89msec=5,67mSec
 * Filtertimeconstant
 * PID Output limit high
 * PID Output limit low
 * load PID Parameters
 * Scaling Setpoint numerator
 * Scaling Setpoint denominator
 * Scaling actual value numerator
 * Scaling actual value denominator
 * Scaling PID Output numerator
 * Scaling PID Output denominator
 * Overtake Scalings

=ana.in.1,PID enable,I enable, Feed Forward on.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/ 637+ / 637f**
Bei Verwendung des Starteingangs X10.11 ist in der Konfiguration die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.



Sollwert Erklärung

t X

0	Analogeingang 1 ($\pm 10V = \pm 2047$)	
1	Analogeingang 2 ($\pm 10V = \pm 2047$)	
2	Istgeschwindigkeit 1 intern	
3	631/635/637	637+/637f
	Istgeschwindigkeit 2 (Inkrement /1,899 ms)	Istgeschwindigkeit 2 (Inkrement /0,844 ms)
4	BIAS-Variable 254	

Istwert Erklärung

t Y

0	Analogeingang 1 ($\pm 10V = \pm 2047$)	
1	Analogeingang 2 ($\pm 10V = \pm 2047$)	
2	Istgeschwindigkeit 1 intern	
3	631/635/637	637+/637f
	Istgeschwindigkeit 2 (Inkrement /1,899 ms)	Istgeschwindigkeit 2 (Inkrement /0,844 ms)
4	BIAS-Variable 255	

Modus: Der Parameter Modus enthält mehrere bitkodierte Schalterfunktionen.

Bit 0:	PID-Freigabe,	0 = PID-Regler deaktiviert (alle Anteile = 0) 1 = PID-Regler aktiviert
Bit 1:	I-Anteil-Freigabe	0 = I-Anteil gesperrt 1 = I-Anteil aktiv
Bit 2:	Vorsteuerungsfreigabe	1 = Vorsteuerung aktiviert
Bit 3:	Vorsteuerungsauswahl	0 = Vorsteuerung standard 1 = Vorsteuerung mit Var. 253
Bit 4:	Integrator-Anteil stoppen	0 = Integrator läuft 1 = Integrator stoppen (der momentan aktuelle Integratorwert bleibt erhalten)
Bit 5...6:	Reserviert	immer 0
Bit 7:	Reiner Analoges Momentenregler (ohne PID-Funktion)	0 = Standard PID Funktion 1 = reiner analoger Momrentenregler

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
0A	X	Y	Z	00	00	00	00

Parameter: D1 = Sollwert (0-4)
D2 = Istwert (0-4)
D3 = Modus (0 – 255)

4 - 0B BIAS-Befehl "Sollposition [AchsNr] = Konst."

Sollposition [AchsNr X] = Y INKR

Erklärung:

Mit dem Befehl "Sollposition [AchsNr] = Konst." ist es möglich, die Sollposition der selektierten Achse auf einen bestimmten Wert zu setzen.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
637f mit Multi-Multi-Achsen-Funktion (Modus 8 = 1)	X: Achsnummer Y: Konstante	1 - 3 +/- 231	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

0 Sollposition [AchsNr 1] = 10000 INKR

Ergebnis:

Die Sollposition der Achse 1 wird auf 10000 Inkremente geladen.

Hinweis:

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
0B	X	Y(low)			Y(high)	00	00

Parameter: D1 = Achse 1 - 3
D2-D5 = Konstante +/- 231

4 - 0C BIAS-Befehl "Sollposition [Achsnr] = [Variable X]"

Sollposition [Achsnr X] = [Variable Y]

Erklärung:

Mit dem Befehl "Sollposition [Achsnr X] = [Variable Y]" ist es möglich, die Sollposition der selektierten Achse auf den Inhalt der definierten Variablen zu setzen.

[Beispiel laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
637f mit Multi-Achsen-Funktion: (Modus 8 = 1)	X: Achsnummer	1 - 3	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
	Y: Variable	0...255	

Beispiel:

0 [Variable 0] = 10000

1 Sollposition [Achsnr 1] = [Variable 0]

Ergebnis:

Die Sollposition der Achse 1 wird auf 10000 Inkremente geladen.

Hinweis:

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
0C	X	Y	00	00	00	00	00

Parameter: D1 = Achsnummer: 1 - 3
D2 = Variablennummer: 0...255

4 - 0D BIAS-Befehl "Fahre relativ"

Fahre relativ; AchsNr. = X , Ziel = Y

Erklärung:

Der Befehl "Fahre Relativ" startet die Achse X auf die Position, die sich aus der Summe der aktuellen Sollposition plus der im Parametersatz definierten relativen Distanz ergibt.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:
637f mit Multi-Achsen-Funktion: (Modus 8 = 1)	X: Achsnummer Y: Ziel	1 - 3 1 = Sollwert in den Lageregler 2 = Sollwert auf CAN2-PDO3 4 = Sollwert auf CAN2-PDO4 8 = Sollwert auf X40-Ausgang Die Parallelschaltung der Sollwertziele ist durch Addition der Funktionscodierung möglich.

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Dieser Befehl ist nur mit der Verwendung der Multi-Achsfunktion beim Reglertyp 637f erlaubt, daher müssen die Parameter über den Befehl "Lade Parametersatz 0 = [Variable[X]]" vorgeladen werden.

Beispiel:

- 0 [Variable 10] = 100000
- 1 [Variable 11] = 1000
- 2 [Variable 12] = 4000
- 3 [Variable 13] = 4000
- 4 [Variable 0] = 10
- 5 Lade Parametersatz 0 = [Variable [0]]
- 6 Starte Achse
- 7 Fahre relativ; AchsNr. = 1 , Ziel = 1

Ergebnis:

Ab der momentanen Sollposition startet die Achse mit der Geschwindigkeit 1000 um 100000 Inkremente weiter.

Hinweis:

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
0D	X	Y	00	00	00	00	00

Parameter: D1 = Achsnummer: 1 - 3
D2 = Ziel: 1...15

4 - 0E BIAS-Befehl "Starte Achse"

Erklärung:

Der Befehl "Starte Achse" setzt die erforderliche Startkennung für einen folgenden Fahrbefehl.

Mit diesem Befehl kann ein Fahrbefehl ohne externe Startbedingung ausgeführt werden.

Jeder Fahrbefehl setzt die Startkennung wieder zurück.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle	keine	-	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

0 Starte Achse

- 1 Fahre Kettenposition
- 2 Warte auf "Pos. erreicht"
- 3 Fahre Kettenposition
- 4

Ergebnis:

Der erste Fahrbefehl "Fahre Kettenposition" wird sofort ausgeführt.

Nachdem die Zielposition erreicht ist (Satz 2), wird die zweite Positionierung erst durch die Low-High-Flanke des Starteingangs gestartet.

Hinweis:

- Der Befehl "Starte Achse" ersetzt die Low-High-Flanke des Starteingangs X10.11 (X10.9 beim 631) als gültige Startkennung.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
0E	00	00	00	00	00	00	00

4 - 0F BIAS-Befehl "Stoppe Achse"

Stoppe Achse; Modus = X ; AchsNr. = Y

Erklärung:

Der Befehl "Stoppe Achse; Modus =" stoppt den Antrieb mit der im Modus angegebenen Funktion.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
631/635/637/ 637+:	X: Modus	0/1 0 = Stopp abrupt (Sollposition = Istposition) 1 = Stopp geführt (mit der aktuellen "Verzögerung")	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
637f mit Multi-Achsen-Funktion: (Modus 8 = 1)	Y: Achsennummer	0 Stoppt alle Achsen und beendet die Multiachsfunktion 1 – 3 Führt immer geführten Stop der angegebenen Achse aus.	

Beispiel:

0 Starte Achse
1 Fahre unendlich Positiv
2 Warte Zeit 1000 ms
3 Verzögerung = 10000 rpm/s
4 Stoppe Achse, Modus 1
5 Warte auf "Pos. erreicht"

Ergebnis:

Der Fahrbefehl "Fahre unendlich Positiv" wird nach 1 Sekunde über die definierte Verzögerungsrampe von 10000 rpm/s gestoppt.

Hinweis:

- Soll die Achse über eine bestimmte Verzögerungsrampe gestoppt werden, empfiehlt sich die Benutzung des Befehls "Stoppe [Achse + Parameter](#)".
- Nach dem Befehl "Stoppe Achse" empfiehlt sich den Befehl "Warte [auf "Pos. erreicht"](#)" in die BIAS-Abarbeitung einfügen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
0F	X	Y	00	00	00	00	00

Parameter: D1 = Modus 0 = Stopp abrupt (Sollposition = Istposition)
D1 = Modus 1 = Stopp geführt (mit der aktuellen "Verzögerung")
D2 = AchsNr 1...3

4- 10 BIAS-Befehl "Fahre Position + Parameter"

Fahre Position; v = X rpm, s= Y INKR

Erklärung:

Der Befehl "Fahre Position + Parameter" startet die Achse mit der im Parameter "v=" angegebenen Geschwindigkeit auf die im Parameter "s=" definierte Position.

Beispiel [laden](#)

Parameter:

v = 1...12000 rpm

s = ± 231 Inkremente

Für die Fahrkurvengenerierung gelten immer die zuletzt eingestellten Parameter für: Beschleunigung, [biascom21-D.htm](#) Verzögerung, Rampenfilter

Beim Einschalten des Reglers sind diese Parameter mit den Einstellungen aus dem EASYRIDER Menü "Inbetriebnahme Lageregler" vorgeladen.

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

0 Starte Achse

1 Fahre Position; v= 100 rpm, s= 1000 INKR

2 Warte auf "Pos. erreicht"

3....

Ergebnis:

Die Achse fährt auf die Absolutposition 1000 Inkremente mit einer Geschwindigkeit von 100 rpm.

Die BIAS-Abarbeitung wird mit dem Satz 3 nach dem Erreichen [der Position](#) fortgeführt.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/637+/637f**
Bei Verwendung des Starteingangs X10.11 ist in der "Eingangs-Konfiguration" die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- **631/ 635/ 637**
Die maximal zulässige Geschwindigkeit hängt von der konfigurierten Resolverauflösung ab. Bei hoher Resolverauflösung sind lediglich 4000 rpm zulässig.
- Bei diesem Befehl ist die Verwendung der Einheitenkonvertierung der EASYRIDER-Software möglich. Bei Verwendung der berechneten Einheiten können Änderungen der eingegebenen Position auftreten, da diese Positionen immer auf eine ganzzahlige Inkrementanzahl gerundet werden müssen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
10	X(low)	X(high)	Y(low)			Y(high)	00

Parameter: D1, D2 = Geschwindigkeit 1....12000 rpm
D3...D6 = Absolutposition ±231 Inkremente

4 - 11 BIAS-Befehl "Fahre Kettenposition + Parameter"

Fahre Kettenposition; v = X rpm, s= Y INKR

Erklärung:

Der Befehl "Fahre Kettenposition + Parameter" startet die Achse mit der im Parameter "v=" angegebenen Geschwindigkeit auf die Position, die sich aus der Summe der aktuellen Zielposition plus der im Parameter "s=" definierten Position ergibt.

Beispiel [laden](#)

Parameter:

v = 1...12000 rpm

s = ± 231 Inkremente

Für die Fahrkurvengenerierung gelten immer die zuletzt eingestellten Parameter für: Beschleunigung, [biascom21-D.htm](#) Verzögerung, Rampenfilter

Beim Einschalten des Reglers sind diese Parameter mit den Einstellungen aus dem Menü "Inbetriebnahme Lageregler" vorgeladen.

- BIAS-Programm
 SPS-Programm
 Mathematik-Programm

Beispiel:

0 Starte Achse

1 Fahre Kettenposition; v= 100 rpm, s= 1000 INKR

2 Warte auf "Pos. erreicht"

3....

Ergebnis:

Die Achse fährt von der aktuellen Sollposition +1000 Inkremente mit einer Geschwindigkeit von 100 rpm.

Die BIAS-Abarbeitung wird mit dem Satz 3 nach dem Erreichen [der Position](#) fortgeführt.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/637+/637f**
Bei Verwendung des Starteingangs X10.11 ist in der "Eingangs-Konfiguration" die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- **631/ 635/ 637**
Die maximal zulässige Geschwindigkeit hängt von der konfigurierten Resolverauflösung ab. Bei hoher Resolverauflösung sind lediglich 4000 rpm zulässig.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
11	X(low)	X(high)	Y(low)			Y(high)	00

Parameter: D1,D2 = Geschwindigkeit 1....12000 rpm
 D3...D6 = Relativposition ±231 Inkremente

4 - 12 BIAS-Befehl "Fahre Referenz +Parameter"

Fahre Referenz; Modus=X ; v = Y rpm , s= Z INKR

Erklärung:

Der Befehl "Fahre Referenz+ Parameter" startet die Referenzfahrt mit der im Parameter "v=" angegebenen Geschwindigkeit entsprechend dem angegebenen "Modus".

Im Parameter "s=" wird die Referenzpunktverschiebung definiert.

631 [Beispiel laden](#)

635/637 [Beispiel laden](#)

637+ [Beispiel laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X: Modus	Modus: <u>0...23</u>	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
	Y: Geschwindigkeit	v =1...12000 rpm (Geschwindigkeit)	
	Z: Referenzpunktverschiebung	s =± 231 Inkremente (Weg)	

Für die Fahrkurvengenerierung gelten immer die zuletzt eingestellten Parameter für: Beschleunigung, Verzögerung, Rampenfilter

Beim Einschalten des Reglers sind diese Parameter mit den Einstellungen aus dem Menue Inbetriebnahme Lageregler vorgeladen.

Beispiel:

0 Starte Achse

1 Fahre Referenz, Modus 16, v= 100rpm, s=4000 Inkr

2 Warte auf "Pos. erreicht"

Ergebnis:

Die Achse referiert auf die der Low-High-Flanke des Referenzsensor folgenden Gebernullstellung in positiver Richtung (Sensor & 0-Punkt).



Von dieser Position aus wird eine Referenzpunktverschiebung um +4000 Inkremente vorgenommen.

Hinweis:

Geber können Resolver oder HIPERFACE sein mit ihrem entsprechenden Nullpunkt bei Regelung auf Istposition 2 wird der Nullimpuls des externen Geber ausgewertet.

Für die Modi 0 -11 muss der Parameter "Weg" auf Null gesetzt werden.

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte Achse"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
- 635/ 637/ 637+/637f
Bei Verwendung des Starteingangs X10.11 ist in der Konfiguration die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
- 631
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 ([Starteingang](#)) zu selektieren.
- 635/ 637/ 637+/637f
Bei Benutzung eines Modus mit Referenzsensor X10.24 ist in der "Eingangs-Konfiguration" die Funktion 1 (Referenzsensor) zu selektieren.
 - **631**
Bei Benutzung eines Modus mit Referenzsensor muß für einen der Eingänge (X10.8, X10.9 oder X10.10) in der Konfiguration die Funktion 1 (Referenzsensor) selektiert sein.
 - **631/ 635/ 637** Die maximal zulässige Geschwindigkeit hängt von der konfigurierten Resolverauflösung ab.
Bei hoher Resolverauflösung sind lediglich 4000 rpm zulässig.
- **637f**
Ist der 637f Regler mit der Optionsplatine C28 bestückt kann optional der Eingang X120.4 als Referenzsensor konfiguriert werden. Der Referenzsensoreingang am X10 Stecker ist dann nicht in Funktion.
-
-
- Befehlskodierung:

Code**D1**

D2	D3	D4	D5	D6	D7
Y(low)	Y(high)	Z(low)			Z(high)

12

X

Parameter: D1 = Referenzmodus: 0...23
 D2, D3 = Geschwindigkeit 1....12000 rpm
 D4...D7 = Referenzverschiebung, Position $\pm 2^{31}$ Inkremente

4 - 13 BIAS-Befehl "Fahre unendlich Positiv+Parameter"

Fahre positiv; v= X rpm , a+ = Y rpm/s , a- = Z rpm/s

Erklärung:

Der Befehl "Fahre unendlich Positiv + Parameter" startet die Achse mit den angegebenen Parametern "v=" (Geschwindigkeit), "a+=" (Beschleunigung) und "a-=" (Verzögerung) in positiver Richtung.

[631 Beispiel laden](#)

[635/637 Beispiel laden](#)

[637+ Beispiel laden](#)

Parameter:

v = 1...12000 rpm

a+= 5...320000 rpm/s

a-= 5...320000 rpm/s

Für die Fahrkurvengenerierung gelten immer die zuletzt eingestellten Parameter für: [Rampenfilter](#)

Beim Einschalten des Reglers sind diese Parameter mit den Einstellungen aus dem Menue Inbetriebnahme Lageregler vorgeladen.

BIAS-Programm

SPS-Programm

Mathematik-Programm

Beispiel:

0 Fahre positiv; v= 900 rpm, a+= 200rpm/s, a-= 5000 rpm/s

1 Wenn Eingang 11 == 1 dann Springe 1

2 Stoppe Achse ; Modus 1

3 ...

Ergebnis:

Die Achse wird in positiver Richtung verfahren sowie der Starteingang aktiviert wird.

Nach dem Deaktivieren des Starteingangs wird die Achse gestoppt.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl ["Starte Achse"](#)
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/ 637+/637f**
Bei Verwendung des Starteingangs X10.11 ist in der Konfiguration die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- **631/ 635/ 637** Die maximal zulässige Geschwindigkeit hängt von der konfigurierten Resolverauflösung ab.
Bei hoher Resolverauflösung sind lediglich 4000 rpm zulässig.

.Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
13	X(low)	X(high)	Y(low)	Y(high)	Z(low)	Z(high)	00

Parameter: D1,D2 = Geschwindigkeit 1....12000 [rpm]
 D3,D4 = Beschleunigung 1....64000 [5*rpm/s]
 D5,D6 = Verzögerung 1....64000 [5*rpm/s]

4 - 14 BIAS-Befehl "Fahre unendlich Negativ+Parameter"

Fahre negativ; v= X rpm , a+ = Y rpm/s , a- = Z rpm/s

Erklärung:

Der Befehl "Fahre unendlich Negativ + Parameter" startet die Achse mit den angegebenen Parametern "v=" (Geschwindigkeit), "a+=" (Beschleunigung) und "a-=" (Verzögerung) in negativer Richtung.

[631 Beispiel laden](#)

[635/637 Beispiel laden](#)

[637+ Beispiel laden](#)

Parameter:

v = 1...12000 rpm

a+= 5...320000 rpm/s

a-= 5...320000 rpm/s

Für die Fahrkurvengenerierung gelten immer die zuletzt eingestellten Parameter für: Rampenfilter

Beim Einschalten des Reglers sind diese Parameter mit den Einstellungen aus dem Menue Inbetriebnahme Lageregler vorgeladen.

BIAS-Programm

SPS-Programm

Mathematik-Programm

Beispiel:

0 Fahre negativ; v= 900 rpm, a+= 200rpm/s, a-= 5000 rpm/s

1 Wenn Eingang 11 == 1 dann Springe 1

2 Stoppe Achse ; Modus 1

3 ...

Ergebnis:

Die Achse wird in negativer Richtung verfahren sowie der Starteingang aktiviert wird.

Nach dem Deaktivieren des Starteingangs wird die Achse gestoppt.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/ 637+/637f**
Bei Verwendung des Starteingangs X10.11 ist in der Konfiguration die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- **631/ 635/ 637** Die maximal zulässige Geschwindigkeit hängt von der konfigurierten Resolverauflösung ab.
Bei hoher Resolverauflösung sind lediglich 4000 rpm zulässig.

.Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
14	X(low)	X(high)	Y(low)	Y(high)	Z(low)	Z(high)	00

Parameter: D1, D2 = Geschwindigkeit 1...12000 [rpm]
 D3, D4 = Beschleunigung 1...64000 [5*rpm/s]
 D5, D6 = Verzögerung 1...64000 [5*rpm/s]

4 - 15 BIAS-Befehl "Fahre Synchron+Parameter"

Fahre Synchron; Koppelfaktor = X

Erklärung:

Der Befehl "Fahre Synchron + Parameter" startet die Achse mit dem angegebenen Koppelfaktor synchron zu den Masterinkrementen.

[631 Beispiel laden](#)

[635/637 Beispiel laden](#)

[637+ Beispiel laden](#)

Reglertyp:	Parameter:	Zulässige Eingabe:
Alle	X: Standard:	= Koppelfaktor +/-127, mit 1/256 Genauigkeit
	X: erweitert:	Synch.einstellungen 2 ; Linear= 0 .Modus= 8 .Wert = 1 = Koppelfaktor +/-127, mit 1/65536 Genauigkeit

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Für die Fahrkurvengenerierung gelten immer die zuletzt eingestellten Parameter für: Rampenfilter

Beim Einschalten des Reglers sind diese Parameter mit den Einstellungen aus dem Menue Inbetriebnahme Lageregler vorgeladen.

[Weitere Informationen über Synchronfunktionen der 630 Serie](#)

Beispiel:

- 0 Taktlänge = 0 INKR
- 1 Fahre Synchron; Koppelfaktor = 1.0**
- 2 Wenn Eingang 11 == 1 dann Springe 2
- 3 Stoppe Achse ; Modus 1
- 4 ...

Ergebnis:

Nach der Low-High-Flanke des Starteingangs fährt die Achse synchron (1 Masterinkrement am X40 Eingang = 1 Slave Inkrement), bis der Starteingang 0 wird.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
optional : Sensor; [Modus=128](#)
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/ 637+/637f**
Bei Verwendung des Starteingangs X10.11 ist in der Konfiguration die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- Der X40-Modus ist in der Zählerkonfiguration als Impulseingang zu selektieren.
- Die Veränderung des Übersetzungsverhältnisses ist während der Synchronfahrt über den Befehl "Koppelfaktor [=](#)" möglich.
- Der vorangestellte Befehl "Taktlänge [=](#)" definiert die Formatlänge für Sonderfunktionen(Auf-, Absynchronisieren, Offset, Sensorkorrektur). Mit Taktlänge=0 sind diese Funktionen nicht aktiv.

.Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
15	X(low)	X(low)	X(high)	X(VZ)	00	00	00

Parameter: D1-D4= Koppelfaktor, X(high) = +/-127

4 - 16 BIAS-Befehl "Fahre Analogwert + Integrator"

Fahre unendlich mit Analogwert; Integrator = X rpm/s

Erklärung:

Der Befehl "Fahre Analogwert + Integrator" startet die Achse auf die sich aus dem analogen Sollwert (635/637/637+/637f: X10.5 zu X10.18; 631: X10.1 zu X10.2) mal der Drehzahlsollwertnormierung ergebenden Geschwindigkeit. Die zulässige Änderungsgeschwindigkeit der Drehzahl kann über den Integratorwert begrenzt werden (0 schaltet die Begrenzung aus).

631 [Beispiel laden](#)

635/637 [Beispiel laden](#)

637+ [Beispiel laden](#)

Parameter: Integrator = 0...160000 rpm/s; 0 = Integrator deaktiviert

- BIAS-Programm
 SPS-Programm
 Mathematik-Programm

Beispiel:

0 Fahre unendlich mit Analogwert; Integrator = 10000 rpm/s

- 1 Wenn Eingang X == 1 dann Springe 1
- 2 Stoppe Achse; Modus 1

Ergebnis:

Die Achse fährt entsprechend der analogen Sollwertvorgabe (wie Betriebsart "Drehzahlregelung"), solange der Eingang X aktiv ist.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11(X10.9 beim 631)
 - **635/ 637/ 637+/637f**
Bei Verwendung des Starteingangs X10.11 ist in der Konfiguration die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- Der Befehl "[Rampenfilter =](#)" muss vor diesem Fahrbefehl deaktiviert werden.
- Die Parameter Beschleunigung und Verzögerung haben **keinen** Einfluss auf die Rampen

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
16	X(low)	X(high)	00	00	00	00	00

Parameter: D1,D2= Integrator [5*rpm/s]

4 - 17 BIAS-Befehl "Fahre Drehzahl + Integrator"

Fahre Drehzahl, Drehzahl = X rpm , Integr. = Y rpm/s

Erklärung:

Der Befehl "Fahre Drehzahl + Integrator" startet die Achse auf die angegebene Drehzahl unter Umgehung des Lagereglers. Dies entspricht der Betriebsart 1 "Drehzahlregelung". Die zulässige Änderungsgeschwindigkeit der Drehzahl kann über den Integratorwert begrenzt werden (0 schaltet die Begrenzung aus).

631 [Beispiel laden](#)
 635/637 [Beispiel laden](#)
 637+ [Beispiel laden](#)

Parameter: X: v = 1...12000 rpm
 Y: Integrator = 0...160000 rpm/s; 0 = Integ. deaktiviert

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

- 0 Starte Achse
- 1 Fahre Drehzahl, Drehzahl = 1000 rpm , Integr. = 10000 rpm/s
- 2 Wenn Eingang X == 1 dann Springe 1
- 3 Stoppe Achse; Modus 1

Ergebnis:

Die Achse fährt entsprechend der Vorgabe auf eine Drehzahl von + 1000 min⁻¹, solange der Eingang X aktiv ist.

Hinweis:

- Vor dem Start wird auf eine gültige Startkennung gewartet:
 - a.) vorangestellt der Befehl "Starte [Achse](#)"
 - b.) Low-High-Flanke des Starteingangs X10.11 (X10.9 beim 631)
 - **635/ 637/ 637+/637f**
Bei Verwendung des Starteingangs X10.11 ist in der Konfiguration die Funktion 0 für den Eingang X10.11 (Starteingang) zu selektieren.
 - **631**
Bei Verwendung des Starteingangs X10.9 ist in der Konfiguration die Funktion 3 für den Eingang X10.9 (Starteingang) zu selektieren.
- Der Befehl "[Rampenfilter =](#)" muss vor diesem Fahrbefehl deaktiviert werden.
- Die Parameter Beschleunigung und Verzögerung haben **keinen** Einfluss auf die Rampen

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
17	X(low)	X(high)	Y(low)	Y(high)	00	00	00

Parameter: D1,D2= Geschwindigkeit [rpm]
 D3,D4= Integrator [5*rpm/s]

4 - 1A BIAS-Befehl "Taktlänge = Konst."

Taktlänge = X INKR

Erklärung:

Der Befehl "Taktlänge =" definiert die Mastertaktlänge für Synchronanwendungen.

Der Wert 0 deaktiviert die Formatüberwachung.

631 [Beispiel laden](#)

635/637 [Beispiel laden](#)

637+ [Beispiel laden](#)

Parameter: X: s = ± 231 Inkremente

BIAS-Programm

SPS-Programm

Mathematik-Programm

Weitere [Informationen über Synchronfunktionen der 630 Serie](#)

Beispiel:

Ergebnis:

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
1A	X(low)			X(high)	00	00	00

Parameter: D1-D4= Taktlänge: ± 231

4 - 1B BIAS-Befehl "Taktlänge = [Variable X]"

Taktlänge = [Variable X]

Erklärung:

Der Befehl "Taktlänge = [Variable X]" definiert die Mastertaktlänge durch den Variableninhalt X in Inkrementen für Synchronanwendungen.

Ein Variableninhalt von 0 deaktiviert die Formatüberwachung.

Der Wert 0 deaktiviert die Formatüberwachung.

631 [Beispiel laden](#)

635/637 [Beispiel laden](#)

637+ [Beispiel laden](#)

Parameter: X: 0 -255

BIAS-Programm

SPS-Programm

Mathematik-Programm

Weitere [Informationen über Synchronfunktionen der 630 Serie](#)

Beispiel:

Ergebnis:

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
1B	X	00	00	00	00	00	00

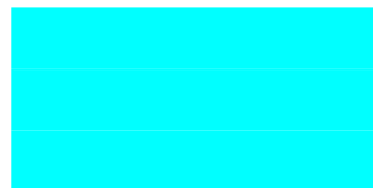
Parameter: D1= Variablennummer: 0...255

4 - 1C BIAS-Befehl "Lade Parametersatz X = [Variable [Y]]"

Lade Parametersatz X = [Variable [Y]]

Erklärung:

Der Befehl "Lade Parametersatz X = [Variable[Y]]" definiert durch den Inhalt der Zeigervariablen Y die Startvariable des entsprechenden Parametersatzes vom Typ X.



Achsnnummer/ Nocken (bei Typ 2):	Parameter: X			<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
	Typ 0: Parametersatz für Positionierbefehle	Typ 1: Parametersatz für Synchronfahrt	Typ 2: Parametersatz für Nockenfunktion	
1	0	1	8	
2	2	3	9	
3	4	5	10	
4	6 (Reserve)	7 (Reserve)	11	
Parameters: Y				
Zeigervariable Y auf die Startvariable des entsprechenden Parametersatzes vom Typ X.			0 - 255	
Weitere Informationen über Synchronfunktionen der 630 Serie				

Beispiel:

Ergebnis:

Hinweis:

- Die Funktion "Synchronlauf" erhält ihre Steuerparameter vor dem Start aus den Variablen, die im Parametersatz angegeben sind und im Zustand "Aktiv" aus dem Achszustand.
- Die Beschleunigung und Verzögerung werden in 5 min-1/s übernommen

Parametersatz: Typ 0:

Parametersatz für Positionierbefehle

Var. + 0: Position	Zielposition in Inkrementen(+/-2 ³¹)
Var. + 1: Geschwindigkeit	Max. Geschwindigkeit ,immer positiv, in rpm
Var. + 2: Beschleunigung	Beschleunigung (5-64000) in rpm/(min*Sec)
Var. + 3: Verzögerung	Verzögerung(5-64000) in rpm/(min*sec)

Parametersatz: Typ 1:

Parametersatz für Synchronbefehle

0: Startwinkel	Definiert den Einschaltpunkt des Synchronlaufs. Einheit Inkremente des ausgewählten Leitwertes für den Synchronlauf.
1: Totzeitkomp.,Sync	Die durch die Abtastung, mechanische Verzögerung und elektrische Schaltzeiten entstehende Phasenverschiebung zwischen Master und Slave wird durch Addition eines geschwindigkeitsproportionalen Offsets zum SSI-Geberwinkel kompensiert. Die Eingabe erfolgt in %. 100% entspricht der exakten Kompensation der Verzögerung der SSI-Abtastung im Regler (844uSec)
2: Koppelfaktor Zähler	Übersetzungsverhältnis Zähler 32bit Integer Wert.
3: Koppelfaktor Nenner	Übersetzungsverhältnis Nenner 31bit unsigned Integer Wert.
4: Move_Linked_ACC	Fahrweg des Slaves in Inkrementen während des Auframpfen, Fahrweg = Eingabe/2.
5: Move_Linked_CONT	Fahrweg des Slaves in Inkrementen
6: Move_Linked_DEC	Fahrweg des Slaves in Inkrementen während dem Abrampfen Fahrweg = Eingabe/2.
7: Move_Linked_WAIT	Fahrweg in Inkrementen während des Wartens
8:CAM_Startoffset	Verschiebung des Anfangspunkts des CAM-Profiles in Richtung der Masterachse. Der Start erfolgt auf der Slaveachse inkrementell
9: CAM-Nr.	Gibt an welches CAM-Profil zur Ausführung kommt. Generell ist es momentan möglich insgesamt 2048 Profil Stützstellen im Regler zu speichern. Die dann auf 2-16 verschiedene Profile verteilt werden können. An einer Erweiterung der Stützstellenzahl wird gearbeitet. Für dieses Projekt wird mit 2 verschiedenen Profilen <1024 Stützstellen begonnen.
10: Skalierung Zähler	Definiert den Zählerwert für die Ausgangsskalierung
11: Skalierung Nenner	Definiert den Nennerwert für die Ausgangsskalierung
12: Sync. Einstellungen	Bit Code für Synchronlaufoptionen

Bit Code für Synchronlaufoptionen

Bit	Name	Bedeutung
0	START_NOCKEN	Anwahl der Funktion für Synchronlauf Start auf definiertem Masterwinkel (Bit 0 =1)
1	START_RICHTUNG	Richtung in der der Start auf definiertem Masterwinkel erfolgen soll (0= Positive Zählrichtung)
2	GEARBOX_TRIGGER_ON	Einschalten der Triggerfunktion für die Gearbox. Änderungen des Koppelfaktors werden beim Triggerzeitpunkt (bei Erreichen der Formatlänge (Koppelfaktor Nenner) übernommen. (Trigger On =1). Für sofortige Übernahme des Koppelfaktors muss das Bit Gearbox_Trigger_ext. nach dem Schreiben des neuen Koppelfaktors gesetzt werden (Trigger On =0).
3	BACKSTOP	Koppelt die Synchronisierung bei rückwärts drehendem Master ab, solange bis der Punkt der Auskopplung wieder vorwärts überfahren wird. Vorwärts, Rückwärts wird definiert aus dem Bit START_RICHTUNG Steht die Masterposition hinter der Auskoppelposition Wird das Fehlerbit BACK_STOP_ERROR gesetzt
4	CLUTCH_ON	Einschalten der Kupplungsfunktion (Bit 4 =1)
5	RAMP_RICHTUNG_Y	Slave Richtung der Kupplungsfunktion
6	RAMP_UP	Startet das synchrone Auframpen der Kupplungsfunktion
7	RAMP_DOWN	Startet das synchrone Abrampen der Kupplungsfunktion
8	RAMP_WAIT	Startet das synchrone Warten der Kupplungsfunktion
9	MOVE_LINKED	Bei MOVE_LINKED = 1, werden die Distanzen Move_Linked_ACC, Move_Linked_Cont und Move_Linked_DEC nacheinander abgefahren.
10	MERGE_MOVE_LINKED	Mit einer positiven Flanke des Merkers "Merge_Linked_Move" werden die neuen Formate übernommen und ein neues Profil gestartet. Nach dem Neustart wird das Bit "Merge_Linked_Move" gelöscht.
11	CAMING_ON	Schaltet den CAM- Funktionsblock ein
12	STOP_CAM_ENDE	Stoppt den Synchronlauf am Ende der Kurvenscheibe
13	CAM_BACKSTOP_OFF <u>Diese Funktion ist zur Zeit in Vorbereitung!!!</u>	Bei rückwärts drehendem Leitwert stoppt der CAM am Profilanfang, solange bis der Leitwert wieder in positiver Richtung über den Ausschaltpunkt gelaufen ist. Mit diesem Merker kann die Abarbeitung des CAM-Profiles in negativer Leitwert Richtung fortgesetzt werden
14	RESERVE 1	
15	RESERVE 2	Wichtig ! <u>Bit 15 = 0</u>

Parametersatz: Typ 2

Parametersatz für Nocken

0: Modus	Taktlänge des Nockenschaltwerkes. In dieser Applikation 65536 Inkr == 360 Grad am SSI
1: Modulo 1-4	Taktlänge des Nockenschaltwerkes. In dieser Applikation 65536 Inkr == 360 Grad am SSI
2: Einschaltpunkt 1-4	Definiert den Einschaltpunkt des Ausganges bei den Modi 2,4 und 8. Einheit: Inkremente des SSI-Leitwertes.
3: Ausschaltpunkt 1-4	Definiert den Ausschaltpunkt des Ausganges beim Modus 2. Einheit: Inkremente des SSI-Leitwertes.
4: Einschaltdauer 1-4	Definiert die Ein/Ausschaltdauer des Ausganges beim Modus 8. Einheit: ms
5: Totzeitkompensation 1-4	Die durch die Abtastung, mechanische Verzögerung und elektrische Schaltzeiten entstehende Phasenverschiebung zwischen Master und Slave wird durch Addition eines geschwindigkeitsproportionalen Offsets zum SSI-Leitwert kompensiert. Die Eingabe erfolgt in %. 100% entspricht der exakten Kompensation der Verzögerung der SSI-Abtastung im Regler (422µs).

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
1C	X	Y	00	00	00	00	00

Parameter: D1= Parametersatznummer: 0...255
D2= Variablennummer: 0...255

4 - 1F BIAS-Befehl "Stoppe Achse + Parameter"

Stoppe Achse; Modus = X a- = Y rpm/s

Erklärung:

Der Befehl "Stoppe Achse; Modus = X a- = Y" stoppt den Antrieb mit der im Modus angegebenen Funktion.

Beispiel [laden](#)

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle	X: Modus	0/1 0 = Stopp abrupt (Sollposition = Istposition) 1 = Stopp geführt	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
	Y: Verzögerung	a-= 5...320000 rpm/s	

Beispiel:

0 Starte Achse
 1 Fahre unendlich Positiv
 2 Warte Zeit 1000 ms
3 Stoppe Achse, Modus 1 a-= 10000 rpm/s
 4 Warte auf "Pos. erreicht"

Ergebnis:

Der Fahrbefehl "Fahre unendlich Positiv" wird nach 1 Sekunde über die definierte Verzögerungsrampe von 10000 rpm/s gestoppt.

Hinweis:

- Nach dem Befehl "Stoppe Achse + Parameter" empfiehlt sich den Befehl "Warte [auf "Pos. erreicht"](#)" in die BIAS-Abarbeitung einfügen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
1F	X	Y(low)	Y(high)	00	00	00	00

Parameter: D1 = Modus 0 = Stopp abrupt (Sollposition = Istposition)
 D1 = Modus 1 = Stopp geführt
 D2, D3 = Verzögerung 1....64000 [5*rpm/s]

4 - 20 BIAS-Befehl "Weg = Konst."

$$\text{Weg} = X \quad \text{INKR}$$

Erklärung:

Der Befehl "Weg = Konst. " definiert die Sollposition für die Fahrbefehle aus der angegebenen Variablen X. Der definierte Weg ist bis zum nächsten Weg-Befehl aktuell.

Reglertyp:	Parameter	Zulässige Eingabe:	<input checked="" type="checkbox"/> BIAS-Programm
Alle	X:	± 231 Inkremente	<input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

0 Weg = 10000 INKR
1 Starte Achse
2 Fahre Kettenposition
3 Warte auf "Pos erreicht"
4 Springe 4

Ergebnis:

Es werden 10000 Inkremente auf die aktuelle Zielposition addiert und die Achse gestartet.

Hinweis:

- Bei der Verwendung von Fahrbefehlen mit direkter Positionsangabe wird der Befehl überschrieben.
- Bei diesem Befehl ist die Verwendung der Einheitenkonvertierung der EASYRIDER-Software möglich. Bei Verwendung der berechneten Einheiten können Änderungen der eingegebenen Position auftreten, da diese Positionen immer auf eine ganzzahlige Inkrementanzahl gerundet werden müssen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
20	X(low)			X(high)	00	00	00

Parameter: D1...D4 = Absolutposition ±231 Inkremente

4 - 21 BIAS-Befehl "Geschwindigkeit = Konst."

Geschwindigkeit = X rpm

Erklärung:

Der Befehl " Geschwindigkeit = Konst." definiert die Maximalgeschwindigkeit für die folgenden Fahrbefehle. Die definierte Geschwindigkeit bleibt bis zum nächsten Geschwindigkeitsbefehl aktuell.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	1...12000	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

- 0 **Geschwindigkeit = 1000 rpm**
- 1 Weg = 10000 INKR
- 2 Starte Achse
- 3 Fahre Kettenposition
- 4 Warte auf "Pos erreicht"
- 5 Springe 5

Ergebnis:

Die Achse fährt ab der momentanen Position relativ 10000 Inkremente mit einer Maximalgeschwindigkeit von 1000 rpm weiter.

Hinweis:

1. Bei der Verwendung von Fahrbefehlen mit direkter Geschwindigkeitsangabe wird der Befehl überschrieben.
2. **631/635/637:** Die maximal zulässige Geschwindigkeit hängt von der konfigurierten Resolverauflösung ab. Bei hoher Resolverauflösung sind lediglich 4000 rpm zulässig.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
21	X(low)	X(high)	00	00	00	00	00

Parameter: D1,D2 = Geschwindigkeit 1....12000 [rpm]

4 - 22 BIAS-Befehl "Beschleunigung = Konst."

Beschleunigung = X rpm/s

Erklärung:

Der Befehl "Beschleunigung = Konst." definiert die Beschleunigungsrampe für die folgenden Fahrbefehle. Die definierte Beschleunigung bleibt bis zum nächsten Beschleunigungsbefehl aktuell.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	5...320000	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

0 Beschleunigung = 5000 rpm/s
 1 Geschwindigkeit = 1000
 2 Weg = 10000 INKR
 3 Starte Achse
 4 Fahre Kettenposition
 5 Warte auf "Pos erreicht"
 6 Springe 6

Ergebnis:

Die Achse fährt ab der momentanen Position relativ 10000 Inkremente weiter. Dabei werden folgende Werte eingehalten:

- Die Geschwindigkeitsänderung beträgt während der Beschleunigungsphase 5000 rpm pro Sekunde.
- Die Maximalgeschwindigkeit beträgt 1000 rpm.

Hinweis:

- Bei der Verwendung von Fahrbefehlen mit direkter Beschleunigungsangabe wird der Befehl überschrieben.
- Der Beschleunigungsparameter wird bei einer Drehzahländerung von 0 rpm auf eine best. Drehzahl, bzw. bei einer Drehzahländerung ohne Vorzeichenwechsel verwendet.
 - Wird während eines aktiven Fahrbefehls die Drehzahl verringert (ohne Vorzeichenwechsel) wird der Beschleunigungsparameter für die Drehzahländerung verwendet.
 - Verläuft die Drehzahländerung über 0 rpm, wird bis 0 rpm der Verzögerungsparameter und bis zur gewünschten Drehzahl mit geänderten Vorzeichen der Beschleunigungsparameter verwendet.

Minimale Beschleunigung auf 4000rpm

Bsp.:

Beschl. = 5 rpm/s

$$t = \frac{4000 \text{ rpm}}{5 \frac{\text{rpm}}{\text{s}}} = 800 \text{ s}$$

==>

Maximale Beschleunigung auf 4000rpm

Bsp.:

Beschl. = 320000 rpm/s

$$t = \frac{4000 \text{ rpm}}{320000 \frac{\text{rpm}}{\text{s}}} = 0,0125 \text{ s} = 12,5 \text{ ms}$$

==>

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
22	X(low)	X(high)	00	00	00	00	00

Parameter: D1,D2 = Beschleunigung 1....64000 [5*rpm/s]

4 - 23 BIAS-Befehl "Verzögerung = Konst."

$$\text{Verzögerung} = X \quad \text{rpm/s}$$

Erklärung:

Der Befehl "Verzögerung = Konst." definiert die Verzögerungsrampe für die folgenden Fahrbefehle. Die definierte Verzögerung bleibt bis zum nächsten Verzögerungsbefehl aktuell.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	5...320000	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

- 0 **Verzögerung = 10000 rpm/s**
- 1 Beschleunigung = 5000 rpm/s
- 2 Geschwindigkeit = 1000
- 3 Weg = 10000 INKR
- 4 Starte Achse
- 5 Fahre Kettenposition
- 6 Warte auf "Pos erreicht"
- 7 Springe 7

Ergebnis:

Die Achse fährt ab der momentanen Position relativ 10000 Inkremente weiter. Dabei werden folgende Werte eingehalten:

1. Die Geschwindigkeitsänderung beträgt während der Beschleunigungsphase 5000 rpm pro Sekunde.
2. Die Maximalgeschwindigkeit beträgt 1000 rpm.
3. Die Geschwindigkeitsänderung beträgt während der Verzögerungsphase 10000 rpm pro Sekunde.

Hinweis:

- Bei der Verwendung von Fahrbefehlen mit direkter Verzögerungsangabe wird der Befehl überschrieben.
- Der Verzögerungsparameter wird bei einer Drehzahländerung auf 0 rpm verwendet.
 - Wird während eines aktiven Fahrbefehls die Drehzahl verringert (ohne Vorzeichenwechsel) wird der Beschleunigungsparameter für die Drehzahländerung verwendet.
 - Verläuft die Drehzahländerung über 0 rpm, wird bis 0 rpm der Verzögerungsparameter und bis zur gewünschten Drehzahl mit geänderten Vorzeichen der Beschleunigungsparameter verwendet.

.Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
23	X(low)	X(high)	00	00	00	00	00

Parameter: D1,D2 = Verzögerung 1....64000 [5*rpm/s]

4 - 24 BIAS-Befehl "Koppelfaktor = Konst."

Koppelfaktor = X

Erklärung:

Der Befehl "Koppelfaktor = Konst." definiert den Koppelfaktor für den Fahrbefehl "Fahre Synchron".

Unter Verwendung des Befehls "Synchronereinstellungen 2" mit Modus 8 kann die Genauigkeit der Auflösung selektiert werden (Werte 0 oder 1).

Reglertyp:	Parameter:	Zulässige Eingabe:	
Alle	X: Standard:	= Koppelfaktor +/-127, mit 1/256 Genauigkeit	<input checked="" type="checkbox"/> BIAS-Programm
	X: erweitert:	Synch.einstellungen <u>2</u> ; <u>Linear= 0</u> <u>.Modus= 8</u> <u>.Wert = 1</u> = Koppelfaktor +/-127, mit 1/65536 Genauigkeit	<input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Weitere [Informationen über Synchronfunktionen der 630 Serie](#)

Beispiel: 635/637/637+/637f

0 Koppelfaktor = 1.0
1 Fahre Synchron
2 Wenn Eingang 11 == 1 dann Springe 2
3 Stoppe Achse ; Modus 1

Ergebnis:

Die Achse fährt synchron bis der Eingang 11 abfällt. Dabei wird pro 1 am X40-Anschluss eingezählten Inkrement(Masterinkrement) 1 Inkrement verfahren.

Beispiel: 631

0 Koppelfaktor = 1.0
1 Fahre Synchron
2 Wenn Eingang 9 == 1 dann Springe 2
3 Stoppe Achse ; Modus 1

Ergebnis:

Die Achse fährt synchron bis der Eingang 9 abfällt. Dabei wird pro 1 am X40-Anschluss eingezählten Inkrement(Masterinkrement) 1 Inkrement verfahren.

Hinweis:

- Der Parameter "Koppelfaktor" muss vor dem Befehl "Fahre Synchron" definiert sein.
- Veränderungen des Koppelfaktors während des Synchronlaufs werden sofort aktiviert! (ohne Befehl "Parameterübernahme").

.Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
24	X(low)	X(low)	X(high)	X(VZ)	00	00	00

Parameter: D1-D4= Koppelfaktor, X(high) = +/-127

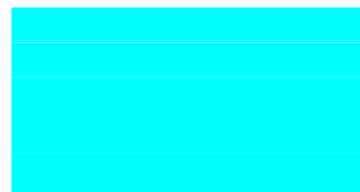
4 - 25 BIAS-Befehl "Position erreicht" Fenster = Konst.

"Pos. erreicht" Fenster = X

Erklärung:

Der Befehl "Position erreicht" Fenster= Konst." legt den Bereich fest, innerhalb dessen die angefahrene Position als erreicht gilt. Der Bereich ergibt sich aus der Sollposition ± des definierten Fensterwertes in Inkremente.

Siehe auch: [Position erreicht](#).



Reglertyp:	Parameter:	Zulässige Eingabe:	
Alle	X:	1...32000	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

0 "Position erreicht" Fenster = 400

- 1 Weg = 10000 Inkr.
- 2 Starte Achse
- 3 Fahre Position
- 4 Warte auf "Position erreicht"
- 5

Ergebnis:

Die BIAS Abarbeitung wird auf Satz 5 weitergeschaltet sobald der Fahrkurvengenerator für die Positionierung berechnet ist und sich die aktuelle Istposition der Achse innerhalb des Bereiches 9600 bis 10400 Inkremente befindet.

Siehe auch: Parameter "[Position erreicht Fenster](#)"

Hinweis:

1. **635/637/637+/637f:** Ist der Ausgang X10.12 mit der Funktion 0 (Position erreicht) konfiguriert, spiegelt dieser die Einhaltung des "Position erreicht"-Bereichs nach aussen wieder. (Ausgang = 1 = Istposition innerhalb des "Position erreicht"-Fensters).
2. **631:** Ist der Ausgang X10.6 mit der Funktion 0 (Position erreicht) konfiguriert, spiegelt dieser die Einhaltung des "Position erreicht"-Bereichs nach aussen wieder. (Ausgang = 1 = Istposition innerhalb des "Position erreicht"-Fensters).

.Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
25	X(low)	X(high)	00	00	00	00	00

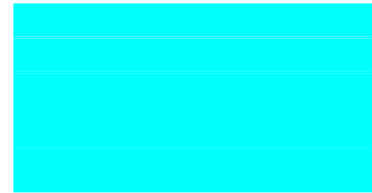
Parameter: D1,D2 = +/-Fensterwert 1....32000 Inkremente

4 - 26 BIAS-Befehl "Restweg = Konst."

Restweg = X INKR

Erklärung:

Mit dem Befehl "Restweg = Konst." wird die Position, die bei der Positionierung mit Sensor, auf den gemessenen Latchwert (gespeicherte Istposition bei der Low-High-Flanke des Sensors) aufaddiert wird, definiert. Der definierte Restweg bleibt bis zum nächsten Restweg Befehl aktuell.

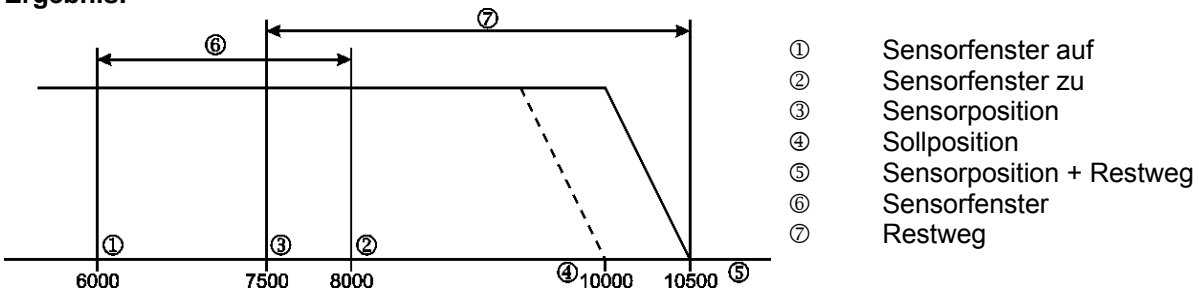


Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	± 231 Inkremente	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

- 0 Istposition 1 = 0 INKR
- 1 Weg = 10000 INKR
- 2 Sensor; Modus= 10 Auf= 6000 Inkr Zu= 8000 Inkr
- 3 Restweg = 3000**
- 4 Fahre Position
- 5 Warte auf "Pos erreicht"
- 6 Springe 6

Ergebnis:



Die Achse fährt die Position 10000 Inkremente absolut an. Dabei wird zwischen den Positionen 6000 Inkremente und 8000 Inkremente die erste Low-High-Flanke des Sensors X10.4 (X10.9 bei 631) (7500 Inkremente) zur Restwegpositionierung ausgewertet. Bei einer angenommenen Sensorposition von 7500 Inkrementen beträgt die neue Zielposition 7500 Inkremente + Restweg (3000 Inkremente) = 10500 Inkremente.

Hinweis:

- Dieser Befehl ist nur in Verbindung mit dem Befehl "Sensorfenster; [Modus =; Modus 10](#)" zu verwenden.
- Bei diesem Befehl ist die Verwendung der Einheitenkonvertierung der EASYRIDER-Software möglich. Bei Verwendung der berechneten Einheiten können Änderungen der eingegebenen Position auftreten, da diese Positionen immer auf eine ganzzahlige Inkrementanzahl gerundet werden müssen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
26	X(low)			X(high)	00	00	00

Parameter: D1...D4 = Restweg ±231 Inkremente

4 - 27 BIAS-Befehl "Rampenfilter = Konst.,[Variable Y]"

Rampenfilter = X , [Variable Y]

Erklärung:

Der Befehl "Rampenfilter = Konst.,[Variable Y]" parametrier die Filterzeitkonstante des Rampenfilters.
 Der Befehl ermöglicht somit eine Ruckbegrenzung der Beschleunigungs- und Bremsrampe.



Reglertyp:	Parameter:	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
631/635/637 637+:	X: X:	0, 4, 8, 12, 16, 20, 24, 28, 32 0 - 7	
637f:	X: Y: Variable	0 - 8 oder 255 0 - 255 zulässiger Variableninhalt 0 - 8	

Beispiel:

Hinweis:

- Bei der Verwendung der Fahrbefehle "Fahre PID, Drehzahl" "Fahre PID, Moment" "Fahre Analogwert" "Fahre Drehzahl" muss der Rampenfilter mit 0 geladen werden.
- Der Wert 0 schaltet die Ruckbegrenzung aus.
- **637f:** Ist der Wert = 255 → wird der Rampenfilter aus dem Inhalt der angegebenen Variablen geladen.

Siehe auch: [Parameter Rampenfilter](#)

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
27	X	Y	00	00	00	00	00

Parameter: D1= Rampenfilterkonstante (siehe [Erklärung](#))
 D2= Rampenfilter aus Variable: 0...255

4 - 28 BIAS-Befehl "Istposition X = Konst."

Istposition X = Y INKR

Erklärung:

Der Befehl "Istposition X = Konst." lädt den angegebenen Positionswert in die definierte Istposition (Zählerpreset).

Reglertyp:	Parameter:	Zulässige Eingabe:	
Alle:	X:	1 = interner Istwert 2 = X40-Eingang	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
	Y:	± 231 Inkremente	

Beispiel:

0 Istposition 1 = 0 INKR
 1 Weg = 10000 INKR
 2 Geschwindigkeit = 1000 rpm
 3 Beschleunigung = 2000 rpm/s
 4 Verzögerung = 2000 rpm/s
 5 Starke Achse
 6 Fahre Position; AchsNr. = 0 , Ziel = 0

Ergebnis:

Es wird der interne Istwert auf 0 Inkrementen gesetzt.
 Anschließend fährt der Antrieb auf die absolute Position von 10000 Inkrementen.

Hinweis:

- Die Istposition die für die Lageregelung verwendet wird, sollte nur im Stillstand verändert werden, da dies einem Stoppbefehl ohne Rampe entspricht!
 Nach dem Setzen einer neuen Istposition, werden die Distanzparameter (Weg = Konst.) auch auf diesen Wert gesetzt.
- Bei diesem Befehl ist die Verwendung der Einheitenkonvertierung der EASYRIDER-Software möglich. Bei Verwendung der berechneten Einheiten können Änderungen der eingegebenen Position auftreten, da diese Positionen immer auf eine ganzzahlige Inkrementanzahl gerundet werden müssen.
- 637f:** Dieser Befehl hat im Modus "Multiaxis" keine Auswirkung. Die angewählte Istposition wird nicht geändert.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
28	X	Y(low)			Y(high)	00	00

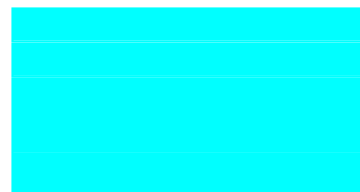
Parameter: D1 = 1 / 2
 D2...D5 = Absolutposition ±231 Inkremente

4 - 29 BIAS-Befehl "Wenn Istposition X ? Konst. dann Springe"

Wenn Istposition X ? Y INKR dann Springe Z

Erklärung:

Der Befehl "Wenn Istposition X ? Konst. dann Springe" erlaubt Programmverzweigungen durch den Vergleich des angewählten Positionswertes X mit der Konstanten Y. Trifft der Vergleich zu wird die Abarbeitung an der angegeben Adresse fortgeführt.



Reglertyp:	Parameter:	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm
Alle:	X:	1: = interner Istwert 2: = X40-Eingang 3: = CAN-Bus	<input checked="" type="checkbox"/> SPS-Programm
	?	> größer (0) < kleiner (1) == gleich (2) >= größer gleich (3) <= kleiner gleich (4) != ungleich (5) -> positiv (6) -< negativ (7)	<input checked="" type="checkbox"/> Mathematik-Programm
	Y:	± 231 Inkremente	
	Z:Adresse:	0...1499 Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.	

Beispiel:

0 Ausgang 20 = 0
1 **Wenn Istposition 1 < 1000 dann Springe 1**
2 Ausgang 20 = 1

Ergebnis:

Der Ausgang 20 wird im Positionsbereich 1000 Inkremente bis 10000 Inkremente gesetzt.

LOOP:

3 **Wenn Istposition 1 < 10000 dann Springe LOOP**
4 Springe 0

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7.0-D7.3	D7.4-D7.7
29	X	Y(low)		Y(high)		Z(low)	Z(high)	?

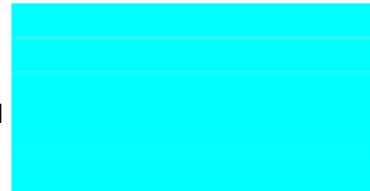
Parameter: D1 = Istposition 1,2,3 (siehe [Erklärung](#))
D2...D5 = Absolutposition ±231 Inkremente
D6,D7.0-D7.3 = Sprungadresse 0...1499
D7.4-D7.7 = Vergleichsbedingung 0...7 (siehe [Erklärung](#))

4 - 2A BIAS-Befehl "Wenn Istposition X ? [Variable Y] dann Springe"

Wenn Istposition X ? [Variable Y] dann Springe Z

Erklärung:

Der Befehl "Wenn Istposition X ? [Variable Y] dann Springe" erlaubt Programmverzweigungen durch den Vergleich des angewählten Positionswertes X mit dem Inhalt der Variablen Y. Trifft der Vergleich zu wird die Abarbeitung an der angegeben Adresse fortgeführt.



Reglertyp:	Parameter:	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm
Alle:	X:	1: = interner Istwert 2: = X40-Eingang 3: = CAN-Bus	<input checked="" type="checkbox"/> SPS-Programm
	?	> größer (0) < kleiner (1) == gleich (2) >= größer gleich (3) <= kleiner gleich (4) != ungleich (5) -> positiv (6) -< negativ (7)	<input checked="" type="checkbox"/> Mathematik-Programm
	Y:	0-255	
	Z:Adresse:	0...1499 Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.	

Beispiel:

- 0 Variable 0 = 1000
- 1 Variable 1 = 10000
- 2 Ausgang 20 = 0
- 3 Wenn Istposition 1 < [Var. 0] dann Springe 3**
- 4 Ausgang 20 = 1

Ergebnis:

Der Ausgang 20 wird im Positionsbereich 1000 Inkremente bis 10000 Inkremente gesetzt.

LOOP:

- 5 Wenn Istposition 1 < [Var. 1] dann Springe LOOP**
- 6 Springe 2

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
2A	X	?	Y	Z(low)	Z(high)	00	00

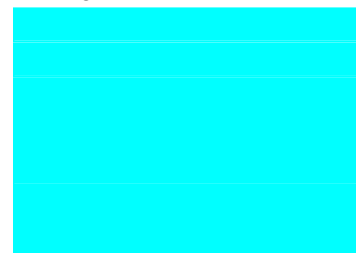
Parameter: D1 = Istposition: 1,2,3 (siehe [Erklärung](#))
 D2 = Vergleichsbedingung 0...255
 D3 = Variablennummer 0...7 (siehe [Erklärung](#))
 D4,D5 = Sprungadresse 0...1499

4 - 2B BIAS-Befehl "Sensorfenster = Konst."

Sensor; Modus= X ; Auf = Y Inkrem.; Zu = Z Inkrem.

Erklärung:

Der Befehl "Sensorfenster" lädt den **Modus** und die Fensterwerte für die Sensorsteuerung der 630 Geräte.
 Die Sensorsteuerung ist lediglich für Positionierungen in positiver Richtung definiert.
 Ein aufgetretener Sensorimpuls deaktiviert die angewählte Funktion automatisch.



Reglertyp:	Parameter :	Zulässige Eingaben:
Alle:	X: Modus	0 = beide Sensorfunktionen ausschalten. 1 = Sensorfunktion Synchronlauf einschalten. Latcheingang <u>2</u> auf Istpos2. 2 = kundenspezifische Sonderfunktion. 3 = Sensorfunktion Synchronlauf einschalten. Latcheingang 2 auf Istpos1. 4 = Quadraloc Funktion ohne Autokorrektur Latch auf Istpos1. 5 = Quadraloc Funktion mit Autokorrektur Latch auf Istpos1. 6 = Quadraloc Funktion ohne Autokorrektur Latch auf Istpos2. 7 = Quadraloc Funktion mit Autokorrektur. Latch auf Istpos2. 10 = Sensorfunktion mit Restwegfunktion einschalten. Latcheingang <u>1</u> auf Istpos1. 11 = Sensorfunktion für Latcheingang 1 Latch auf Istpos1. Die Überwachung des Sensorfensters ist deaktiviert. 12 = Sensorfunktion für Latcheingang 2 Latch auf Istpos1. Die Überwachung des Sensorfensters ist deaktiviert. 64 = Sensorfunktion für Latcheingang 1 Latch auf Istpos2. Die Überwachung des Sensorfensters ist deaktiviert. 65 = Sensorfunktion für Latcheingang 2 Latch auf Istpos2. Die Überwachung des Sensorfensters ist deaktiviert. 128 = Sensorfunktion Synchronlauf einschalten. Latcheingang 2 auf Istpos2. 254 = Sensorfunktion für Latcheingang 2 ausschalten. 255 = Sensorfunktion für Latcheingang 1 ausschalten.
	Y:	224 Inkremente
	Z:	224 Inkremente

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel: für Modus 10:

- 0 Weg = 10000 INKR
- 1 Restweg = 2000 INKR
- 2 **Sensor; Modus=10 ;Auf=6000 Inkrem.;Zu=9000 Inkrem.**
- 3 Starte Achse
- 4 Fahre Position

Ergebnis:

Die Achse wird auf die absolute Position positioniert. Sollte im Positionsbereich von 6000 bis 9000 Inkrementen eine Low-High-Flanke des Sensoreinganges X10.4 auftreten, wird ab dieser Position ein Restweg von 2000 Inkrementen gefahren.

Hinweis:

Latcheingang 1:

635/637/637+/637f Eingang X10.4 mit Funktion 0 konfigurieren,

631 Eingang X10.9 mit Funktion 5 konfigurieren.

Ob ein Interrupt aufgetreten ist kann mit dem Befehl "Variable [\[X\]=Latchstatus 1](#)" abgefragt werden. Der Latchwert kann mit dem Befehl "Variable [\[X\]=Latchposition 1](#)" gelesen werden.

Latcheingang 2:

635/637/637+/637f Eingang X10.25 mit Funktion 0 konfigurieren,

631 Eingang X10.10 mit Funktion 5 konfigurieren.

Die Überwachung der maximalen Korrekturlänge ist aktiviert.

siehe auch Sensoreinstellungen 1 und Sensoreinstellungen 2

Ob ein Interrupt aufgetreten ist kann mit dem Befehl "Variable [\[X\]=Latchstatus 2](#)" abgefragt werden.

Der Latchwert kann mit dem Befehl "Variable [\[X\]=Latchposition 2](#)" gelesen werden.

1. Bei Synchronanwendungen werden die Fensterwerte relativ zum Formatbeginn angegeben.
2. Wird für den Parameter Sensorfenster Zu der Wert 0 angegeben wird das Sensorfenster nicht geschlossen.

Befehlskodierung:

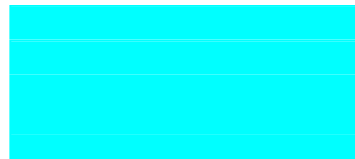
Code	D1	D2	D3	D4	D5	D6	D7
2B	X	Y(low)		Y(high)	Z(low)		Z(high)

Parameter: D1 = Modus (siehe [Erklärung](#))
 D2-D4 = Fenster auf 224 Inkremente
 D5-D7 = Fenster zu 224 Inkremente

4 - 2C BIAS-Befehl "Sensorfenster = Konst."

Erklärung:

Der Befehl "Sensorposition = Konst." definiert den Abstand, in dem der Sensor vom Startpunkt aus erwartet wird.
Die Sensorposition wird für die Auswertung der maximalen Korrektur im Synchronlauf benötigt.



Reglertyp:	Parameter	Zulässige Eingaben:
	:	
Alle:	Konst.	± 231

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

Ergebnis:

Hinweis:

- Dieser Befehl ist nur in Verbindung mit den Befehlen "Sensorfenster, [Modus 1](#), "Sensoreinstellungen [1](#)" und "Sensoreinstellungen [2](#)" zu verwenden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
2C	X(low)			X(high)	00	00	00

Parameter: D1- D4 = Position: ± 231

4 - 2D BIAS-Befehl "Sensoreinstellung 1 = Konst."

Sensoreinstellungen; Korrektur= X Mittelwert= Y

Erklärung:

Der Befehl "Sensoreinstellungen 1" definiert den maximal zulässigen Wert für die Korrektur der Position entsprechend der gelatchten Sensorposition im Sensormodus 1.

Der Parameter Mittelwert gibt die Anzahl der Latchpositionen an deren Mittelwert für die Positionskorrektur ausgewertet werden soll.

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X: Korrektur: Y: Mittelwert:	224 Inkremente 0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

Ergebnis:

Hinweis:

- Dieser Befehl ist nur in Verbindung mit den Befehlen "Sensorfenster, [Modus 1](#)"; "Sensorposition" und "Sensoreinstellungen [2](#)" zu verwenden.

Befehlskodierung:

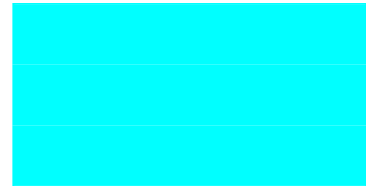
Code	D1	D2	D3	D4	D5	D6	D7
2D	X (low)		X (high)	Y	00	00	00

Parameter: D1 - D3 = Korrekturwert: 224
D4 = Mittelwert 0...255

4 - 2E BIAS-Befehl "Sensoreinstellung 2 = Konst."

Erklärung:

Der Befehl "Sensoreinstellungen 2" definiert die Zeit, innerhalb derer die Positionskorrektur ausgeführt werden soll.



Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
Alle:	Konst.	65535 in ms	

Beispiel:

Ergebnis:

Hinweis:

- Dieser Befehl ist nur in Verbindung mit den Befehlen "Sensorfenster, [Modus 1](#)", "Sensorposition" und "Sensoreinstellungen [1](#)" zu verwenden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
2E	X (low)	X (high)	00	00	00	00	00

Parameter: D1, D2 = Korrekturzeit: 0...65535

4 - 2F BIAS-Befehl "Parameterübernahme"

Erklärung:

Der Befehl "Parameterübernahme" erlaubt die Übernahme der Parameter Weg, Geschwindigkeit, Beschleunigung, Verzögerung und „Position erreicht“ -Fenster während eines laufenden Fahrbefehls.

Während einer laufenden Synchronfahrt hat der Befehl keine Auswirkungen.

637f: Im Multi-Achsen Modus ist dieser Befehl nicht erlaubt!

Reglertyp:	Parameter	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm
	:		<input type="checkbox"/> SPS-Programm
Alle:	Keine	---	<input type="checkbox"/> Mathematik-Programm

Beispiel:

0 Weg = 100000 Inkr.
 1 Geschwindigkeit = 100 rpm
 2 Starte Achse
 3 Fahre Position
 4 Geschwindigkeit = 1000 rpm
 5 Wenn Istposition 1 < 10000 Inkr dann Springe 5
 6 **Parameterübernahme**
 7 Warte auf "Pos. erreicht"

Ergebnis:

Die Achse fährt mit einer Geschwindigkeit von 100 rpm bis zur Position 10000 Inkr. Danach wird die neue Geschwindigkeit 1000 rpm übernommen.

Hinweis:

- Es werden jeweils die zuletzt geladenen Parameterwerte übernommen.

Befehlskodierung:

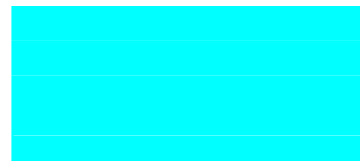
Code	D1	D2	D3	D4	D5	D6	D7
2F	00	00	00	00	00	00	00

Parameter:

4 - 30 BIAS-Befehl "Weg = [Variable X]"

Erklärung:

Der Befehl "Weg = [Variable X]" definiert die Sollposition für die Fahrbefehle aus der angegebenen Variablen X. Der definierte Weg ist bis zum nächsten Wegbefehl aktuell.



Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	0...255 Zulässiger Variableninhalt ± 231 Inkremente	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

- 0 [Variable 0] = 10000
- 1 **Weg = [Variable 0]**
- 2 Starte Achse
- 3 Fahre Kettenposition
- 4 Warte auf "Pos erreicht"
- 5 Springe 2

Ergebnis:

Es werden 10000 Inkremente auf die aktuelle Zielposition addiert und die Achse gestartet.

Hinweis:

- Bei der Verwendung von Fahrbefehlen mit direkter Positionsangabe wird der Befehl überschrieben.
- Der Inhalt der Variablen wird als inkrementelle Positionsvorgabe interpretiert.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
30	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
zulässiger Variableninhalt: ± 231

4 - 31 BIAS-Befehl "Geschwindigkeit = [Variable X]"

Erklärung:

Der Befehl " Geschwindigkeit = [Variable X]" definiert die Maximalgeschwindigkeit entsprechend dem Inhalt der angegebenen Variablen X für die folgenden Fahrbefehle. Die definierte Geschwindigkeit bleibt bis zum nächsten Geschwindigkeitsbefehl aktuell.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	0...255 zulässiger Variableninhalt: 1...12000	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

0 [Variable 0] = 1000
 1 **Geschwindigkeit = [Variable 0]**
 2 Weg = 10000 INKR
 3 Starte Achse
 4 Fahre Kettenposition
 5 Warte auf "Pos erreicht"
 6 Springe 3

Ergebnis:

Die Achse fährt ab der momentanen Zielposition additiv 10000 Inkremente mit einer Maximalgeschwindigkeit von 1000 rpm weiter.

Hinweis:

- Bei der Verwendung von Fahrbefehlen mit direkter Geschwindigkeitsangabe wird der Befehl überschrieben.
- Der Inhalt der Variablen wird als Geschwindigkeitsvorgabe in Umdrehungen pro Minute interpretiert.
- 631/635/637:** Die maximal zulässige Geschwindigkeit hängt von der konfigurierten Resolverauflösung ab. Bei hoher Resolverauflösung sind lediglich 4000 rpm zulässig.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
31	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 32 BIAS-Befehl "Beschleunigung = [Variable X]"

Erklärung:

Der Befehl "Beschleunigung = [Variable X]" definiert die Beschleunigungsrampe entsprechend dem Inhalt der angegebenen Variablen X für die folgenden Fahrbefehle. Die definierte Beschleunigung bleibt bis zum nächsten Beschleunigungsbefehl aktuell.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	0...255 zulässiger Variableninhalt: 1...64000	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

- 0 [Variable 0] = 1000
- 1 **Beschleunigung = [Variable 0]**
- 2 Geschwindigkeit = 1000
- 3 Weg = 10000 INKR
- 4 Starte Achse
- 5 Fahre Kettenposition
- 6 Warte auf "Pos erreicht"
- 7 Springe 4

Ergebnis:

- Die Achse fährt ab der momentanen Zielposition additiv 10000 Inkremente weiter. Dabei werden folgende Werte eingehalten:
- 3. Die Geschwindigkeitsänderung beträgt während der Beschleunigungsphase 5000 rpm pro Sekunde.
 - 4. Die Maximalgeschwindigkeit beträgt 1000 rpm.

Hinweis:

1. Bei der Verwendung von Fahrbefehlen mit direkter Beschleunigungsangabe wird der Befehl überschrieben.
2. Der Inhalt der Variablen wird als Beschleunigungsvorgabe in 5 rpm/s interpretiert.
- Der Beschleunigungsparameter wird bei einer Drehzahländerung von 0 rpm auf eine best. Drehzahl, bzw. bei einer Drehzahländerung ohne Vorzeichenwechsel verwendet.
 - Wird während eines aktiven Fahrbefehls die Drehzahl verringert (ohne Vorzeichenwechsel) wird der Beschleunigungsparameter für die Drehzahländerung verwendet.
 - Verläuft die Drehzahländerung über 0 rpm, wird bis 0 rpm der Verzögerungsparameter und bis zur gewünschten Drehzahl mit geänderten Vorzeichen der Beschleunigungsparameter verwendet.

Minimale Beschleunigung auf 4000rpm

Bsp.:

Beschl. = 5 rpm/s

$$t = \frac{4000 \text{rpm}}{5 \frac{\text{rpm}}{\text{s}}} = 800\text{s}$$

==>

Maximale Beschleunigung auf 4000rpm

Bsp.:

Beschl. = 320000 rpm/s

$$t = \frac{4000 \text{rpm}}{320000 \frac{\text{rpm}}{\text{s}}} = 0,0125\text{s} = 12,5\text{ms}$$

==>

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
32	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 33 BIAS-Befehl "Verzögerung = [Variable X]"

Erklärung:

Der Befehl "Verzögerung = [Variable X]" definiert die Verzögerungsrampe entsprechend dem Inhalt der angegebenen Variablen X für die folgenden Fahrbefehle. Die definierte Verzögerung bleibt bis zum nächsten Verzögerungsbefehl aktuell.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	0...255 zulässiger Variableninhalt: 1...64000	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

0 [Variable 0] = 2000
 1 **Verzögerung = [Variable 0]**
 2 Beschleunigung = 5000 rpm/s
 3 Geschwindigkeit = 1000
 4 Weg = 10000 INKR
 5 Starte Achse
 6 Fahre Kettenposition
 7 Warte auf "Pos erreicht"
 8 Springe 4

Ergebnis:

Die Achse fährt ab der momentanen Zielposition additiv 10000 Inkremente weiter. Dabei werden folgende Werte eingehalten:

- Die Geschwindigkeitsänderung beträgt während der Beschleunigungsphase 5000 rpm pro Sekunde.
- Die Maximalgeschwindigkeit beträgt 1000 rpm.
- Die Geschwindigkeitsänderung beträgt während der Verzögerungsphase 10000 rpm pro Sekunde.

Hinweis:

- Bei der Verwendung von Fahrbefehlen mit direkter Verzögerungsangabe wird der Befehl überschrieben.
- Der Inhalt der Variablen wird als Verzögerungsvorgabe in 5 rpm/s interpretiert.
- Der Verzögerungsparameter wird bei einer Drehzahländerung auf 0 rpm verwendet.
 - Wird während eines aktiven Fahrbefehls die Drehzahl verringert (ohne Vorzeichenwechsel) wird der Beschleunigungsparameter für die Drehzahländerung verwendet.
 - Verläuft die Drehzahländerung über 0 rpm, wird bis 0 rpm der Verzögerungsparameter und bis zur gewünschten Drehzahl mit geänderten Vorzeichen der Beschleunigungsparameter verwendet.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
33	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 34 BIAS-Befehl "Koppelfaktor = [Variable X]"

Erklärung:

Der Befehl "Koppelfaktor = [Variable X]" definiert den Koppelfaktor für den Fahrbefehl "Fahre Synchron" aus der angegebenen Variablen. Unter Verwendung des Befehls "Synchronereinstellungen 2" mit Modus 8 kann die Genauigkeit der Auflösung selektiert werden (Werte 0, 1 oder 2).

Reglertyp:	Parameter:	Zulässige Eingabe:	
Alle	X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
		zulässiger Variableninhalt siehe: Synchronereinstellungen <u>2</u>	

Beispiel:

635/637/637+/637f

- 0 [Variable 0] = 256
- 1 **Koppelfaktor = [Variable 0]**
- 2 Fahre Synchron
- 3 Wenn Eingang 11 == 1 dann Springe 3
- 4 Stoppe Achse ; Modus 1

Ergebnis:

Die Achse fährt synchron bis der Eingang 11 abfällt. Dabei wird pro 1 am X40-Anschluss eingezählten Inkrement(Masterinkrement) 1 Inkrement verfahren.

Beispiel:

631

- 0 [Variable 0] = 256
- 1 **Koppelfaktor = [Variable 0]**
- 2 Fahre Synchron
- 3 Wenn Eingang 9 == 1 dann Springe 3
- 4 Stoppe Achse ; Modus 1

Ergebnis:

Die Achse fährt synchron bis der Eingang 9 abfällt. Dabei wird pro 1 am X40-Anschluss eingezählten Inkrement(Masterinkrement) 1 Inkrement verfahren.

Hinweis:

- Der Parameter "Koppelfaktor" muss vor dem Befehl "Fahre Synchron" definiert sein.
- Veränderungen des Koppelfaktors während des Synchronlaufs werden sofort aktiviert (ohne Befehl "Parameterübernahme").

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
34	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 35 BIAS-Command "Position reached" window = [variable X]"

Pos. window = [variable X]

Explanation:

The command "Position reached window = [variable X]" determines the range within the position moved to is regarded as reached.

The range results from the setpoint position \pm the defined window.

Drive type:	Parameters:	Permissible inputs:	
All	X:	0...255 permitted variable content 1...32000	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

Example:

0 [Variable 0] = 400

1 Pos. window = [variable 0]

2 Position = 10000 incr

3 Start axis

4 Move position

5 Wait for "position reached"

Result:

The BIAS execution is continued as soon as the ramp generation of the move command is ended and the actual position of the axis is located in the range of 9600 to 10400 increments

Note:

- **635/637/637+/ 637f**
If output X10.12 is configured with the function 0 (position reached), it will reflect the "position reached" window (output = 1 = actual position within the "position reached" window).
- **631**
If output X10.6 is configured with the function 0 (position reached), it will reflect the "position reached" window (output = 1 = actual position within the "position reached" window).
- The variable content is interpreted as a window specification in increments.

Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
35	X	00	00	00	00	00	00

Parameters: D1 = Variable number: 0...255

4 - 36 BIAS-Befehl "Restweg = [Variable X]"

Erklärung:

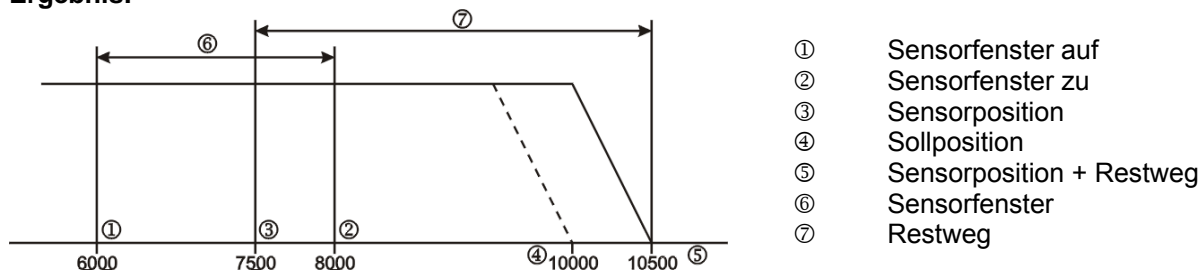
Mit dem Befehl "Restweg=[Variable X]" wird die Position, die bei der Positionierung mit Sensor, auf den gemessenen Latchwert (gespeicherte Istposition bei der Low-HighFlanke des Sensors) aufaddiert wird, durch den Inhalt der angegebenen Variablen definiert. Der definierte Restweg bleibt bis zum nächsten Restweg Befehl aktuell.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	0...255 Zulässiger Variableninhalt ± 231 Inkremente	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

- 0 Istposition 1 = 0 INKR
- 1 Weg = 10000 INKR
- 2 Sensor;Modus= 10 Auf= 6000 Inkr Zu= 8000 Inkr
- 3 [Variable 0] = 3000
- 4 **Restweg = [Variable 0]**
- 5 Fahre Position
- 6 Warte auf "Pos erreicht"
- 7 Springe 0

Ergebnis:



Die Achse fährt die Position 10000 Inkremente absolut an. Dabei wird zwischen den Positionen 6000 Inkremente und 8000 Inkremente die erste Low-High-Flanke des Sensors X10.4 (X10.9 bei 631) (7500 Inkremente) zur Restwegpositionierung ausgewertet. Bei einer angenommenen Sensorposition von 7500 Inkrementen beträgt die neue Zielposition 7500 Inkremente + Restweg (3000 Inkremente) = 10500 Inkremente.

Hinweis:

- Dieser Befehl ist nur in Verbindung mit dem Befehl "Sensorfenster; [Modus =; Modus 10](#)" zu verwenden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
36	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 37 BIAS-Befehl "Maximalstrom = [Variable X]"

Maximalstrom = [Variable X], Modus = Y

Erklärung:

Der Befehl "Maximalstrom = [Variable X]" definiert den maximal zulässigen Strom des Regler und ermöglicht somit eine variable Momentenbegrenzung der Achse.

Reglertyp:	Parameter	Zulässige Eingabe:	
	X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	Y:	0: Der Inhalt der Variablen wird in 1/10 Ampere ausgewertet.	
		1: Der Inhalt der Variablen wird in D/A-Stufen ausgewertet	
		Zulässiger Variableninhalt:	
	Modus 0:	(2 x Reglernennstrom) x 10	
631	Modus 1:	0 - 6061 = 100% Reglermaximalstrom.	
635/637/637+/637 f		0 - 12123 = 100% Reglermaximalstrom.	

Beispiel:

0 [Variable 10] = 10
 1 [Variable 11] = 100
 2 Weg = 10000 INKR
 3 **Maximalstrom = [Variable 11], Modus = 0**
 4 Starte Achse
 5 Fahre Position
 6 Wenn Istposition 1 < 7500 INKR dann Springe 6
 7 **Maximalstrom = [Variable 10], Modus = 0**
 8 Warte auf "Pos erreicht"
 9 Springe 3

Ergebnis:

Die Position 10000 INKR wird angefahren im Bereich von 0 bis 7500 INKR beträgt der Maximalstrom höchstens 10 Aeff. Im Bereich über 7500 INKR wird der Maximalstrom auf 1 Aeff begrenzt.

Hinweis: Der angegebene Maximalstrom ist nur dann maßgebend, wenn es keine weiteren Begrenzungen auf kleinere Maximalströme gibt.

- Die externe Strombegrenzung über Analogeingang 2 ist aktiviert (nicht bei 631!).
- Die Funktion Strombegrenzung bei Warnung, d.h. Abregelung bei Effektivwertüberschreitung ist aktiviert.
- Der Maximalstrom ist in der "Konfiguration Motor" prozentual begrenzt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
37	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
 D2 = Modus 1/ 2

4 - 38 BIAS-Befehl "Istposition X = [Variable Y]"

Istposition X = [Variable Y]

Erklärung:

Der Befehl "Istposition X = [Variable Y]" lädt den Inhalt der angegebenen Variablen in die definierte Istposition (Zählerpreset).

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle:	X:	1 = interner Istwert 2 = X40-Eingang	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
	Y:	0...255	

Beispiel:

- 0 [Variable 0] = 0
- 1 **Istposition 1 = [Variable 0]**
- 2 Weg = 10000 INKR
- 3 Geschwindigkeit = 1000 rpm
- 4 Beschleunigung = 2000 rpm/s
- 5 Verzögerung = 2000 rpm/s
- 6 Starte Achse
- 7 Fahre Position; AchsNr. = 0 , Ziel = 0

Ergebnis:

Es wird der interne Istwert auf 0 Inkrementen gesetzt.
Anschließend fährt der Antrieb auf die absolute Position von 10000 Inkrementen.

Hinweis:

- Die Istposition die für die Lageregelung verwendet wird, sollte nur im Stillstand verändert werden, da dies einem Stoppbefehl ohne Rampe entspricht.
Nach dem Setzen einer neuen Istposition, werden die Distanzparameter (Weg = Konst.) auch auf diesen Wert gesetzt.
- **637f:** Dieser Befehl hat im Modus "Multiaxis" keine Auswirkung. Die angewählte Istposition wird nicht geändert.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
38	X	Y	00	00	00	00	00

Parameter: D1 = Modus 1/ 2
D2 = Variablennummer: 0...255

4 - 39 BIAS-Befehl "Analogausgang X = [Variable Y]"

Analogausgang X = [Variable Y]

Erklärung:

Der Befehl "Analogausgang X = [Variable Y]" lädt den Inhalt der angegebenen Variable in den angegebenen Analogausgang.

Reglertyp:	Parameter	Zulässige Eingabe:	
631: 635/ 637/ 637+/- 637f	Keine X: Y:	Befehl steht nicht zur Verfügung 1 = MP 1 (X10.17) 2 = MP 2 (X10.6) 0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
		Zulässiger Variableninhalt	
635/ 637:	bei X = 1/ 2	-64...+63	entspr. -10...+10V
637+/-637f:	bei X = 1 MP1 bei X = 2 MP2	-512...+511 -128...+127	entspr. -10...+10V entspr. -10...+10V

Beispiel für : 635/ 637:

0 [Variable 0] = -64

1 Analogausgang 1 = [Variable 0]

Ergebnis:

Am Analogausgang MP1 liegt eine Spannung von -10V an.

Hinweis:

- Der entsprechende Analogausgang muss als BIAS-Analogausgang konfiguriert sein (siehe "Konfiguration", "Ausgänge")

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
39	X	Y	00	00	00	00	00

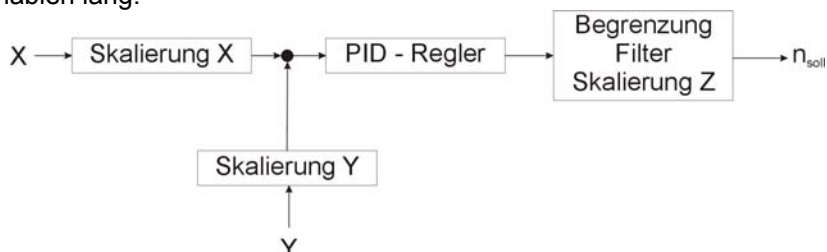
Parameter: D1 = MP 1/ 2
D2 = Variablennummer: 0...255

4 - 3A BIAS-Befehl "PID Skalierung"

Erklärung:

Der Befehl "PID Skalierung" legt den Variablenbereich für die Skalierung des PID-Reglers fest.

Der Variablenbereich beginnt mit der angegebenen Variablen und ist 6 Variablen lang.



Reglertyp:	Parameter	Erklärung:	Zulässiger Variableninhalt
Alle	n	Skalierung X Zähler	±32767
	n+1	Skalierung X Nenner	±32767
	n+2	Skalierung Y Zähler	±32767
	n+3	Skalierung Y Nenner	±32767
	n+4	Skalierung Z Zähler	±32767
	n+5	Skalierung Z Nenner	±32767

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel :

- 0 [Variable 5] = 1250
- 1 [Variable 6] = 1000
- 2 [Variable 7] = 900
- 3 [Variable 8] = 1000
- 4 [Variable 9] = 500
- 5 [Variable 10] = 1000
- 6 **PID Skalierung; Start = [Variable 5]**

Ergebnis:

Der Befehl "PID Skalierung" normiert den Sollwert mit einem Faktor von 1.25, den Istwert mit 0.9 und den PID Ausgang mit Faktor 0.5.

Hinweis:

$$X1 = X \cdot \frac{[\text{Variable } n]}{[\text{Variable } n + 1]}$$

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
3A	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 3B BIAS-Befehl "Sensorfenster = [Variable X]"

Sensor; Modus = [Var. X]; Auf = [Var. Y]; Zu = [Var. Z]

Erklärung:

Der Befehl "Sensorfenster" lädt den **Modus** und die Fensterwerte aus den angegebenen Variablen für die Sensorsteuerung der 630 Geräte. Die Sensorsteuerung ist lediglich für Positionierungen in positiver Richtung definiert. Ein aufgetretener Sensorimpuls deaktiviert die angewählte Funktion automatisch.

Reglertyp:	Parameter :	Zulässige Eingaben:	
Alle:	X: Modus	0-255 zulässiger Variableninhalt: 0 = beide Sensorfunktionen ausschalten. 1 = Sensorfunktion Synchronlauf einschalten. Latcheingang <u>2</u> auf Istpos2. 2 = kundenspezifische Sonderfunktion. 3 = Sensorfunktion Synchronlauf einschalten. Latcheingang 2 auf Istpos1. 4 = Quadraloc Funktion ohne Autokorrektur Latch auf Istpos1. 5 = Quadraloc Funktion mit Autokorrektur Latch auf Istpos1. 6 = Quadraloc Funktion ohne Autokorrektur Latch auf Istpos2. 7 = Quadraloc Funktion mit Autokorrektur. Latch auf Istpos2. 10 = Sensorfunktion mit Restwegfunktion einschalten. Latcheingang <u>1</u> auf Istpos1. 11 = Sensorfunktion für Latcheingang 1 Latch auf Istpos1. Die Überwachung des Sensorfensters ist deaktiviert. 12 = Sensorfunktion für Latcheingang 2 Latch auf Istpos1. Die Überwachung des Sensorfensters ist deaktiviert. 64 = Sensorfunktion für Latcheingang 1 Latch auf Istpos2. Die Überwachung des Sensorfensters ist deaktiviert. 65 = Sensorfunktion für Latcheingang 2 Latch auf Istpos2. Die Überwachung des Sensorfensters ist deaktiviert. 128 = Sensorfunktion Synchronlauf einschalten. Latcheingang 2 auf Istpos2. 254 = Sensorfunktion für Latcheingang 2 ausschalten. 255 = Sensorfunktion für Latcheingang 1 ausschalten.	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	Y:	0-255 zulässiger Variableninhalt 224 Inkremente	
	Z:	0-255 zulässiger Variableninhalt 224 Inkremente	

Beispiel: für Modus 10:

- 0 Variable 0= 10
- 1 Variable 1= 6000
- 2 Variable 2= 9000
- 3 Weg = 10000 INKR
- 4 Restweg = 2000 INKR
- 5 **Sensor; Modus=[Var. 0];Auf=[Var. 1];Zu=[Var. 2]**
- 6 Starte Achse

Ergebnis:

Die Achse wird auf die absolute Position positioniert. Sollte im Positionsbereich von 6000 bis 9000 Inkrementen eine Low-High-Flanke des Sensoreinganges X10.4 auftreten, wird ab dieser Position ein Restweg von 2000 Inkrementen gefahren.

7 Fahre Position

Hinweis:

Latcheingang 1:

635/637/637+/637f Eingang X10.4 mit Funktion 0 konfigurieren,

631 Eingang X10.9 mit Funktion 5 konfigurieren.

Ob ein Interrupt aufgetreten ist kann mit dem Befehl "Variable [\[X\]=Latchstatus 1](#)" abgefragt werden.

Der Latchwert kann mit dem Befehl "Variable [\[X\]=Latchposition 1](#)" gelesen werden.
werden.

Latcheingang 2:

635/637/637+/637f Eingang X10.25 mit Funktion 0 konfigurieren,

631 Eingang X10.10 mit Funktion 5 konfigurieren.

Die Überwachung der maximalen Korrekturlänge ist aktiviert.

siehe auch Sensoreinstellungen 1 und Sensoreinstellungen 2

Ob ein Interrupt aufgetreten ist kann mit dem Befehl "Variable [\[X\]=Latchstatus 2](#)" abgefragt werden.

Der Latchwert kann mit dem Befehl "Variable [\[X\]=Latchposition 2](#)" gelesen werden.

1. Bei Synchronanwendungen werden die Fensterwerte relativ zum Formatbeginn angegeben.
2. Wird für den Parameter Sensorfenster Zu der Wert 0 angegeben wird das Sensorfenster nicht geschlossen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
3B	X	Y	Z	00	00	00	00

Parameter: D1 = Modus (siehe [Erklärung](#))
D2 = Variable 0-255 Fenster auf 224 Inkremente
D3 = Variable 0-255 Fenster zu 224 Inkremente

4 - 3C BIAS-Befehl "Sensorposition = [Variable X]"

Erklärung:

Der Befehl "Sensorposition = [Variable X]" definiert mit dem Inhalt der angegebenen Variablen den Abstand, in dem der Sensor vom Startpunkt aus erwartet wird.

Die Sensorposition wird für die Auswertung der maximalen Korrektur im Synchronlauf.

Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm
Alle:	X: Modus	0...255	<input checked="" type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

Ergebnis:

Hinweis:

- Dieser Befehl ist nur in Verbindung mit den Befehlen "Sensorfenster, [Modus 1](#), "Sensoreinstellungen [1](#)" und "Sensoreinstellungen [2](#)" zu verwenden.
- Der Variableninhalt wird als Position in Inkremente interpretiert.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
3C	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 3D BIAS-Befehl "Sensoreinstellung 1 = [Variable X]"

Sensoreinstellungen; Korrektur= [Var. X]; Mittelwert= [Var. Y]

Erklärung:

Der Befehl "Sensoreinstellungen 1 = [Variable X]" definiert den maximal zulässigen Wert für die Korrektur der Position entsprechend der gelatchten Sensorposition im Sensormodus 1 aus dem Inhalt der Variablen X. Der Inhalt der Variablen Y gibt die Anzahl der Latchpositionen an deren Mittelwert für die Positionskorrektur ausgewertet werden soll.

Reglertyp:	Parameter	Zulässige Eingaben:	
	:		<input checked="" type="checkbox"/> BIAS-Programm
			<input checked="" type="checkbox"/> SPS-Programm
			<input type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255 Zulässiger Variableninhalt: ± 224 Inkremente	
	Y:	0...255 Zulässiger Variableninhalt: 0...255	

Beispiel:

Ergebnis:

Hinweis:

- Dieser Befehl ist nur in Verbindung mit den Befehlen "Sensorfenster, Modus 1 = [Variable X]"; "Sensorposition = [Variable X]" und "Sensoreinstellungen 2 = [Variable X]" zu verwenden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
3D	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
D2 = Variablennummer: 0...255

4 - 3E BIAS-Befehl "Sensoreinstellung 2 = [Variable X]"

Erklärung:

Der Befehl "Sensoreinstellungen 2" definiert mit dem Inhalt der angegebenen Variablen die Zeit, innerhalb derer die Positionskorrektur ausgeführt werden soll.

Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
Alle:	X: Modus	0...255 Zulässiger Variableninhalt: 65535 in ms	

Beispiel:

Ergebnis:

Hinweis:

- Dieser Befehl ist nur in Verbindung mit den Befehlen "Sensorfenster, Modus 1 = [Variable X]"; "Sensorposition = [Variable X]" und "Sensoreinstellungen 1 = [Variable X]" zu verwenden.

Befehlskodierung:

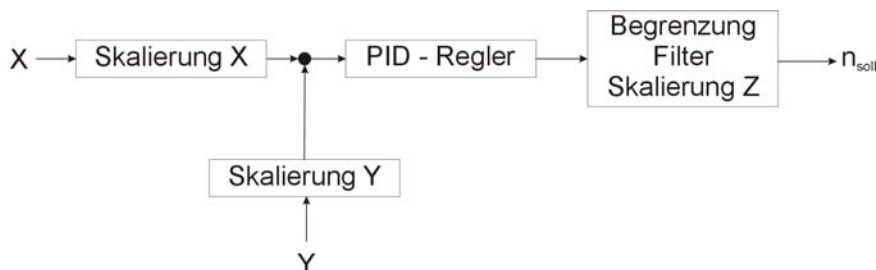
Code	D1	D2	D3	D4	D5	D6	D7
3E	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 3F BIAS-Befehl "PID Parameter"

Erklärung:

Der Befehl "PID Parameter" legt den Variablenbereich für die Parametrierung fest. Der Variablenbereich beginnt mit der angegebenen Variablen und ist 6 Variablen lang.



Reglertyp:	Parameter:	Zulässige Eingaben:	Bezeichnung	Zulässige Variableninhalte	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
631/635/637	n	0...249	P-Anteil	0..32767 * 1/4096	
	n+1		Int. zeitkonst.	0..32767 * 1,899ms	
	n+2		Dif. zeitkonst.	0..32767 * 1,899ms	
	n+3		Filterzeitkonst.	0..32767 * 1,899ms	
	n+4		Begrenzung max	± 32767	
	n+5		Begrenzung min	± 32767	
	n	0...249	P-Anteil	0..32767 * 1/4096	
	n+1		Int. zeitkonst.	0..32767 * 0,844ms	
	n+2		Dif. zeitkonst.	0..32767 * 0,844ms	
	n+3		Filterzeitkonst.	0..32767 * 0,844ms	
	n+4		Begrenzung max	± 32767	
	n+5		Begrenzung min	± 32767	

Beispiel:

Ergebnis:

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
3F	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 40 BIAS-Befehl "[Variable X] = Weg"

Erklärung:

Der Befehl "[Variable X] = Weg" lädt den aktuellen definierten Wegparameter in die angegebene Variable.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 [Variable 0] = Weg

1 [Variable 0] = [Variable 0] + 1000

2 Weg = [Variable 0]

3 Parameterübernahme

Ergebnis:

Die aktuell definierte Sollposition wird um 1000 Inkremente erhöht und aktiviert.

Hinweis:

- Der Wegparameter wird in Inkrementen übernommen

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
40	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 41 BIAS-Befehl “[Variable X] = Geschwindigkeit“

Erklärung:

Der Befehl “[Variable X] = Geschwindigkeit“ lädt die aktuell definierte Sollgeschwindigkeit in die angegebene Variable.

Reglertyp:	Parameter:	Zulässige Eingabe:	
Alle	X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 [Variable 0] = Geschwindigkeit

1 [Variable 0] = [Variable 0] + 100

2 Geschwindigkeit = [Variable 0]

3 Parameterübernahme

Ergebnis:

Die aktuell definierte Sollgeschwindigkeit wird um 100 rpm erhöht und aktiviert.

Hinweis:

- Die Sollgeschwindigkeit wird in min-1 übernommen.
- **631/635/637:** Die maximal zulässige Geschwindigkeit hängt von der konfigurierten Resol verauflösung ab. Bei hoher Resol verauflösung sind lediglich 4000 rpm zulässig.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
41	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 42 BIAS-Befehl “[Variable X] = Beschleunigung“

Erklärung:

Der Befehl “[Variable X] = Beschleunigung“ lädt die aktuellen definierte Beschleunigung in die angegebene Variable.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 [Variable 0] = Beschleunigung
 1 [Variable 0] = [Variable 0] + 100
 2 Beschleunigung = [Variable 0]
 3 Parameterübernahme

Ergebnis:

Die aktuell definierte Beschleunigung wird um 500 rpm/s erhöht und aktiviert.

Hinweis:

- Die Beschleunigung wird in 5 min⁻¹/s übernommen

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
42	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 43 BIAS-Befehl “[Variable X] = Verzögerung“

Erklärung:

Der Befehl “[Variable X] = Verzögerung“ lädt die aktuellen definierte Verzögerung in die angegebene Variable.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

- 0 [Variable 0] = Verzögerung
- 1 [Variable 0] = [Variable 0] + 120
- 2 Verzögerung = [Variable 0]
- 3 Parameterübernahme

Ergebnis:

Die aktuell definierte Verzögerung wird um 600 rpm/s erhöht und aktiviert.

Hinweis:

- Der Inhalt der Variablen wird als Verzögerungsvorgabe in 5 rpm/s interpretiert.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
43	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 44 BIAS-Befehl "[Variable X] = Koppelfaktor"

Erklärung:

Der Befehl "[Variable X] = Koppelfaktor" lädt den aktuell definierten Koppelfaktor in die angegebene Variable X.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm

Beispiel für Synchronereinstellungen 2, Modus 8, Wert 0 (standard):

```

0   [Variable 1] = Istposition 1
1   [Variable 0] = Koppelfaktor
2   [Variable 0] = [Variable 0] - 1
3   [Variable 1] = [Variable 1] + 4096
4   Wenn Istposition 1 < [Variable 1] dann Springe 4
5   Koppelfaktor = [Variable 0]
6   Wenn Istposition < 655360 Springe 2
  
```

Ergebnis:

Bis zur Slaveposition 655360 Inkremente wird alle 4096 Inkremente wird der Koppelfaktor um 1/256 verringert.

Hinweis:

- Beachten Sie die Festlegung der Koppelfaktordefinition in den Befehlen "Koppelfaktor = [Variable X]" oder "Koppelfaktor =" und "Synchronereinstellungen 2"

Befehlskodierung:

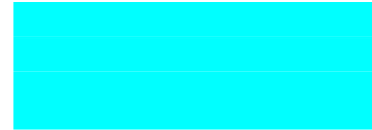
Code	D1	D2	D3	D4	D5	D6	D7
44	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 45 BIAS-Befehl "[Variable X] = Satznummer"

Erklärung:

Der Befehl "[Variable X] = Satznummer " lädt die aktuelle bearbeitete Satznummer der gültigen Abarbeitung in die angegebene Variable X.



Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 ...
 1 ...
 2 **[Variable 1] = Satznummer**
 3 ...

Ergebnis:

Die Satznummer 2 wird in der Variablen 1 gespeichert.

Hinweis:

- Mit diesem Befehl kann der Programmablaufzeiger gespeichert und für weitere Entscheidungen dann verwendet werden.
- siehe auch: Befehl "[BIAS Abarbeitungszeiger = \[Variable X\]](#)".

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
45	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 46 BIAS-Befehl "[Variable X] = Istposition Y"

Erklärung:

Der Befehl "[Variable X] = Istposition Y" lädt die aktuelle Istposition Y in die angegebene Variable X.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X: Y:	0...255 1: interne Istposition (X30) 2: X40 Zähleringang 3: CAN-Bus Absolutwertgeber	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
nur 637f:		3: CAN2 PDO1 SSI –Leitwert 4: CAN2 PDO3 Leitwert 5: CAN2 PDO4 Leitwert	

Beispiel für 635/ 637/ 637+/ 637f:

```

0   [Variable 1] = Istposition 1
1   [Variable 1] = [Variable 1] + 4096
2   Ausgang 13 = 1
3   Wenn Istposition 1 < [Variable 1] dann Springe 3
4   Ausgang 13 = 0

```

Ergebnis:

Der BIAS-Ausgang 13 wird an der aktuellen Position gesetzt und 4096 Inkremente später gelöscht.

Hinweis:

- Die Position wird in Inkrementen übernommen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
46	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
D2 = Istposition: 1...5 (siehe [Erklärung](#))

4 - 47 BIAS-Befehl "[Variable X] = Analogeingang"

Erklärung:

Der Befehl "[Variable X] = Analogeingang Y" lädt den Wert des definierten Analogeinganges Y in die angegebene Variable.

Reglertyp:	Parameter	Zulässige Eingabe:	
631	X: Y:	0...255 1 = X10.1-X10.2 (Sollwerteingang)	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
635/ 637/ 637+ / 637f		1 = X10.5- X10.18 (Sollwerteingang) 2 = X10.10- X10.19 (externe Strombegrenzung)	

Beispiel:

- 0 [Variable 0] = Analogeingang 2
- 1 [Variable 0] = [Variable 0] / 2
- 2 [Variable 0] = [Variable 0]* -1
- 3 [Variable 1] = [Variable 0] + 2048
- 4 Geschwindigkeit = [Variable 1]
- 5 Parameterübernahme
- 6 Springe 0

Ergebnis:

Die Achse fährt entsprechend der Analogspannung am Analogeingang 2. Dabei folgt die Sollgeschwindigkeit folgendem Zusammenhang:

$$v = 2048 - \frac{[Variable 0]}{4095} \cdot \frac{V_{analog}}{(10V)}$$

Hinweis:

- Für den **Analogeingang 1** beträgt der Wertebereich ± 2047Digits (±11 Bit; ±10V)
Beim 637f werden nur 12Bit anstatt die 14 Bit aus Kompatibilitätsgründen verwendet.
- Für den **Analogeingang 2** beträgt der Wertebereich ± 4095 Digits.
(Real nur ±9 Bit; ±10V)

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
47	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
D2 = Analogeingang (siehe [Erklärung](#))

4 - 48 BIAS-Befehl "[Variable X] = Latchposition Y"

Erklärung:

Der Befehl "[Variable X] = Latchposition Y" lädt die aktuell definierte Latchposition Y in die angegebene Variable X.

Reglertyp:	Parameter	Zulässige Eingabe:	
631:	X: Y:	0...255 1 = Latchwert X10.9 2 = Latchwert X10.10	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
635/ 637/ 637+/- 637f:		1 = Latchwert X10.4 2 = Latchwert X10.25	

Beispiel:

0 [Variable 0] = Latchstatus 1
 1 Wenn [Variable 0] <= 0 dann Sprunge 0
 2 **Variable 1] = Latchposition 1**

Ergebnis:

Die Position des Latchimpulses wird in der Variable 1 gespeichert.

Hinweis:

- Dieser Befehl ist im Zusammenhang mit dem Befehl "Sensorfenster; [Modus X](#)" zu benutzen.
- Die Latchposition ist nur aktuell, wenn im Befehl "[Variable X]= Latchstatus Y" ein Wert > 0 zurückgegeben wird.
- Die Latchpositionen werden in Inkrementen übernommen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
48	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
 D2 = Latchposition 1/2 (siehe [Erklärung](#))

4 - 49 BIAS-Befehl "[Variable X] = Drehzahl Y"

Erklärung:

Der Befehl "[Variable X] = Drehzahl Y " lädt die aktuell definierte Drehzahl Y in die angegebene Variable X.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle	X: Y:	0...255 1 = interner Istposition (X30) 2 = X40-Zählereingang	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
nur 637f		3 = SSI-Istposition3 Drehzahl	

Beispiel 635/ 637/ 637+/ 637f:

- 0** [Variable 0] = Drehzahl 2
1 Wenn [Variable 0] >= 0 dann Springe 3
2 [Variable 0] = [Variable 0]* -1
3 Wenn [Variable 0] >= 100 dann Springe 6
4 Ausgang 13 = 1
5 Springe 0
6 Ausgang 13 = 0
7 Springe 0

Ergebnis:

Der Ausgang X10.13 wird gesetzt sowie die aus den am X40-Zählereingang ankommenden Inkrementen gebildete Istdrehzahl kleiner 100 rpm ist.

Hinweis:

- Die Drehzahl wird in rpm übernommen und ist vorzeichenbehaftet.
- Für die Drehzahl 2 muss die korrekte Inkrementauflösung des externen Gebers in der "Konfiguration", "Zähler", "[X40-Auflösung](#)" angegeben sein.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
49	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
D2 = Drehzahl 1,2,3 (siehe [Erklärung](#))

4 - 4A BIAS-Befehl "[Variable X] = Latchstatus Y"

Erklärung:

Der Befehl "[Variable X] = Latchstatus Y" lädt den aktuell definierten Latchstatus Y in die angegebene Variable X.

Reglertyp:	Parameter	Zulässige Eingabe:	
631:	X: Y:	0...255 1 = Latchwert X10.9 2 = Latchwert X10.10	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
635/ 637/ 637+/- 637f:		1 = Latchwert X10.4 2 = Latchwert X10.25	

Zustand:

0	Latch noch nicht aufgetreten
-1	kein Latch innerhalb des Sensorfensters aufgetreten
1	Latch ausgeführt (Modus 10)
2	Latch ausgeführt (Modus 11)
12	Latch ausgeführt (Modus 12)
64	Latch ausgeführt (Modus 64)
65	Latch ausgeführt (Modus 65)

Beispiel:

0 [Variable 0] = Latchstatus 1
1 Wenn [Variable 0] <= 0 dann Sprunge 0
2 [Variable 1] = Latchposition 1

Ergebnis:

Die Position des Latchimpulses 1 wird in der Variablen 1 gespeichert.

Hinweis:

- Dieser Befehl ist im Zusammenhang mit dem Befehl "Sensorfenster; [Modus X](#)" zu benutzen.
- Die Latchpositionen werden in Inkrementen übernommen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
4A	X	Y	00	00	00	00	00

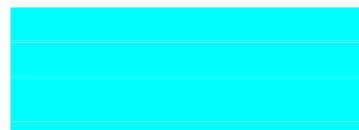
Parameter: D1 = Variablennummer: 0...255
D2 = Latchstatus 1/2 (siehe [Erklärung](#))

4 - 4B BIAS-Befehl "[Variable X] = Position Y"

[Variable X] = Position Y ; AchsNr. Z

Erklärung:

Der Befehl "[Variable X] = Position Y" lädt den Inhalt der definierten Position in die angegebene Variable X.



Reglertyp:	Parameter	Zulässige Eingabe:	
631/ 635/ 637/ 637+/637f	X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	Y: Positionswert	1 = aktuelle Sollposition 2 = interne Zielposition	
637f mit Multi-Achsen-Funktion:: (Modus 8 = 1)	Y: Positionswert	1 = aktuelle Sollposition (Achse Z) 2 = interne Zielposition (Achse Z) 3 = Bremsenpunkt (Achse Z) 4 = Profiltaktlänge (Achse Z)	
	Z: Achsnummer	1-3	

Beispiel:

Ergebnis:

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
4B	X	Y	Z	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
 D2 = Positionswert (siehe [Erklärung](#))
 D3 = Achsnummer: 1...3

4 - 4C BIAS-Befehl "[Variable X] = Wert Y"

Erklärung:

Der Befehl "[Variable X] = Wert Y" lädt den definierten Wert Y in die angegebene Variable.

Reglertyp:	Parameter	Zulässige Eingabe:	
Alle:	X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	Y:	0 = Isteffektivstrom <u>(+/-) in 1/10 Ampere</u> 1 = Betrag <u>des Isteffektivstrom in 1/10 Ampere</u> 2 = Feldbusstatus 3 = <u>aktuelle Verfahrensgeschwindigkeit in Umdrehungen pro Minute</u> 4 = <u>Regelabweichung des Lagereglers in Inkrementen (Schleppfehler)</u> 5 = Sollwert der PID-Funktion 6 = Istwert der PID-Funktion 7 = <u>aktuelle Regeldifferenz der PID-Funktion</u> 8 = <u>aktueller Integralwert der PID-Funktion</u> 9 = <u>Ausgabewert multipliziert mit 256 der PID-Funktion</u> 10 = <u>Zeitzählerwert (in 2 ms Stufen)</u> 11 = <u>aktuell zulässiger Maximalstrom in 1/10 Ampere</u> 12 = <u>aktuell zulässiger Maximalstrom in D/A Steps</u> 13 = <u>aktuell gemessener Isteffektivstrom in D/A Steps</u> 14 = <u>Ucc- Zwischenkreisspannung in Volt</u> 15 = IBT-Maskennummer 16 = <u>Regler-Effektivstrom in 1/10 Ampere</u> 17 = Synchronlauf <u>Geschwindigkeit</u> 18 = Feldbusknotennummer 19 = Ein-/Ausgangsstatus <u>von X120</u> 20 = I ² t <u>Motor</u> 21 = I ² t <u>Regler</u> 22 = CAN2-BUS-Knoten- <u>Überwachungsstatus</u> 23 = Rotorposition <u>bei positiver Flanke des Referenzsensors (ab Firmware V8.20 nur bei 637f)</u>	

Wert = 0:
lädt den aktuell gemessenen Strom vorzeichenrichtig in 1/10 Ampere in die angegebene Variable.

<p>Beispiel: Der Befehl wird bei einem Regler 635 -05 ausgeführt (5 Ampere Nennstrom). Variable 2 = Wert 0</p>	<p>Ergebnis: Die Variable 2 wird mit dem aktuell gemessenen Strom in 1/10 Ampere geladen. Fließt z.B -1 Ampere Strom hat die Variable 2 den Inhalt -10.</p>
---	--

Wert = 1:
lädt der Betrag des aktuell gemessenen Stromes in 1/10 Ampere in die angegebene Variable.

<p>Beispiel: Der Befehl wird bei einem Regler 635 -05 ausgeführt (5 Ampere Nennstrom). 0 Variable 2 = Wert 1</p>	<p>Ergebnis: Die Variable 2 wird mit dem Betrag des aktuell gemessenen Stromes in 1/10 Ampere geladen. Fließt z.B -1 Ampere Strom hat die Variable 2 den Inhalt 10.</p>
--	--

Wert = 2:
lädt die folgenden 4 Byte Statusinformationen in die angegebene Variable.
Diese Information kann z.B über Feldbuschnittstellen ausgelesen werden und weiterverarbeitet werden.

<p>Beispiel: 0 Variable 2 = Wert 2</p>	<p>Ergebnis: Die Variable 2 wird mit den aktuellen Statusinformationen geladen</p>
--	---

Byte 0
Ein- [Ausgangsstatus 631](#) Eingangsstatus low Byte: 635/637 637+/637f

Byte 1 und Byte 2
Fehlerwort: 631 635/637 637f

Byte 3
Statuswort 2 low Byte 631 635/637 637f

Wert = 10:

lädt den aktuellen Zeitählerwert in die angegebene Variable.

Hinweis: 631/ 635 /637
Der interne Zeitähler wird alle 2 ms um 1 erhöht.

637+ / 637f
Der interne Zeitähler wird alle 0,844 ms um 1 erhöht.

Unter Verwendung dieses Befehles können freidefinierbare Zeitfunktionen im BIAS-Programm verarbeitet werden.

Beispiel für 631/ 635 /637:

ProgStart:

```
0 Istposition 1 = 0      INKR
1 Starte Achse
2 Fahre positiv; v= 100 rpm , a+= 5000 rpm/s , a-= 5000 rpm/s
3 [Variable 0 ] = Wert 10
4 [Variable 1 ] = [Variable 0 ] + 2500
```

WAIT:

```
5 Wenn Istposition 1 > 100000      INKR dann Springe STOP
6 [Variable 0 ] = Wert 10
7 Wenn [Variable 0 ] -< [Variable 1 ] dann Springe WAIT
```

STOP:

```
8 Stoppe Achse; Modus = 1
9 NOP
10 NOP
```

Ergebnis:

Die Achse soll die Bewegung nach Überschreiten der Istposition > 10000 oder spätestens nach 5 Sekunden gestoppt werden.

Wert = 11:

lädt den aktuell erlaubten Maximalstrom in 1/10 Ampere in die angegebene Variable.

Hinweis: Der Maximalstrom kann durch folgende Funktionen beeinflusst werden:

- Die externe Strombegrenzung über Analogeingang 2 ist aktiviert (nicht bei 631!).
- Die Funktion Strombegrenzung bei Warnung, d.h. Abregelung bei Effektivwertüberschreitung ist aktiviert.
- Der Maximalstrom ist in der "Konfiguration Motor" prozentual begrenzt.
- Der Maximalstrom ist durch den BIAS-Befehl "Maximalstrom = Variable" bestimmt.

Generell ist immer der kleinste aktivierte Wert für die Maximalstrombegrenzung gültig.

Beispiel:

Der Befehl wird bei einem Regler 635 -05 ausgeführt (5 Ampere Nennstrom) und in der "Konfiguration Motor" ist der Maximalstrom auf 50 % begrenzt.

Ergebnis:

Die Variable 2 wird mit dem Wert des momentan gültigen Maximalstromes in 1/10 Ampere geladen.
Die Variable 2 hat den Inhalt 50.

0 Variable 2 = Wert 11

Wert = 12:

lädt den aktuell erlaubten Maximalstrom mit der internen Normierung in die angegebene Variable.

Hinweis: Der Maximalstrom kann durch folgende Funktionen beeinflusst werden:

- Die externe Strombegrenzung über Analogeingang 2 ist aktiviert (nicht bei 631 !).
- Die Funktion Strombegrenzung bei Warnung, d.h. Abregelung bei Effektivwertüberschreitung ist aktiviert.
- Der Maximalstrom ist in der "Konfiguration Motor" prozentual begrenzt.
- Der Maximalstrom ist durch den BIAS-Befehl "Maximalstrom = Variable" bestimmt.

Generell ist immer der kleinste aktivierte Wert für die Maximalstrombegrenzung gültig.

Hinweis: 631: Der Reglermaximalstrom von 100% hat den internen Wert 6143.

635/637/637+/637f: Der Reglermaximalstrom von 100% hat den internen Wert 12123.

Beispiel:

Der Befehl wird bei einem Regler 635 -05 ausgeführt (5 Ampere Nennstrom) und in der "Konfiguration Motor" ist der Maximalstrom auf 50 % begrenzt.0

Variable 2 = Wert 11

Ergebnis:

Die Variable 2 wird mit dem Wert des momentan gültigen Maximalstromes mit der internen Normierung geladen.

Die Variable 2 hat den Inhalt von ca.6060.

0 Variable 2 = Wert 12

Wert = 13:

lädt den aktuell gemessenen Strom vorzeichenrichtig mit der internen Normierung in die angegebene Variable.

Hinweis: 631: Der Reglermaximalstrom von 100% hat den internen Wert 6143.

635/637/637+/637f: Der Reglermaximalstrom von 100% hat den internen Wert 12123.

Beispiel:

Befehl wird bei einem Regler 635 -05 ausgeführt (5 Ampere Nennstrom).

0 Variable 2 = Wert 13

Ergebnis:

Die Variable 2 wird mit dem aktuell gemessenen Strom in der internen Normierung geladen. Fließt z.B. 5 Ampere Strom, hat die Variable 2 den Inhalt von ca. 6060

Wert = 14:

lädt die aktuell gemessene Zwischenkreisspannung in Volt in die angegebene Variable.

Beispiel:

Die Zwischenkreisspannung beträgt 322 V

0 Variable 2 = Wert 14

Ergebnis:

Die Variable 2 wird mit dem Wert der Zwischenkreisspannung geladen.
Die Variable 2 hat den Inhalt 322.

Wert = 15:

lädt die aktuelle Maskennummer des Bediengerätes IBT in die angegebene Variable.

Hinweis: Dieser Befehl ist nur bei Verwendung des Bediengerätes IBT mit CAN-Bus-Anschluss zu verwenden.

Mit dieser Information kann im BIAS-Programm ausgewählt werden, ob eine Maskenumschaltung über den Befehl "IBT-Maskennummer" erlaubt wird.

Beispiel:

Die Anzeige des Bedienfeldes befindet sich in der Hauptmaske (4).

0 Variable 2 = Wert 15

Ergebnis:

Die Variable 2 wird mit der aktuellen Maskennummer geladen. Die Variable 2 hat den Inhalt 4.

Wert = 16:

lädt den Regler-Effektivstrom in 1/10 Ampere in die angegebene Variable.

Beispiel:

Der Befehl wird bei einem Regler 635 -05 ausgeführt (5 Ampere Nennstrom 10 Ampere Effektivstrom).

0 Variable 2 = Wert 16

Ergebnis:

Die Variable 2 wird mit dem Regler-Effektivstrom in 1/10 Ampere geladen.
Die Variable 2 hat den Inhalt 100.

Wert = 17:

lädt die aktuell gemessene Synchrongeschwindigkeit (Δ Inkremente) in die angegebene Variable.
Diese Geschwindigkeit gilt nicht für den Multiachs- Synchronlauf.

$$v = \frac{\Delta \text{Inkr} \times 60 \frac{\text{s}}{\text{min}}}{65535 \text{Inkr} \times \text{Abtastzeitlageregler}}$$

Abtastzeitlageregler:

637+, 637f: 0,844ms

631, 635, 637: 1,899ms

Beispiel für 637+/637f:

Die Synchrongeschwindigkeit beträgt 1000rpm.

0 Variable 2 = Wert 17

Ergebnis:

Die Variable 2 wird mit dem Wert der Synchrongeschwindigkeit geladen.

Die Variable 2 hat den Inhalt 921.

Wert = 18:

lädt die aktuelle Knotennummer bzw. Achsnummer in die angegebene Variable.

Beispiel:

Die Knotennummer beträgt 3.

0 Variable 2 = Wert 18

Ergebnis:

Die Variable 2 wird mit dem Wert der Knotennummer geladen.

Die Variable 2 hat den Inhalt 3.

Wert = 19:

lädt den aktuellen Ein-/ Ausgangsstatus von dem Stecker X120 in die angegebene Variable.

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung.

Bit	Beschreibung
0	Eingang 1.
1	Eingang 2.
2	Eingang 3.
3	Eingang 4.
4	Ausgang 1.
5	Ausgang 2.
6	Ausgang 3.
7	Ausgang 4.
8	Nocken 1.
9	Nocken 2.
10	Nocken 3.
11	Nocken 4.
12	Reserve.
13	Reserve.
14	Nockenschaltwerk On.
15	Nocken Backstop.

Beispiel:

Die Eingänge 2 und 3, sowie die Ausgänge 1 und 4 führen High-Pegel, die restlichen Zustände führen low Pegel.

Ergebnis:

Die Variable 2 wird mit dem Wert des Ein-/Ausgangsstatus geladen.
Die Variable 2 hat den Inhalt 150. (96hex)

0 Variable 2 = Wert 19

Wert = 20:

lädt die aktuelle Motorauslastung in die angegebene Variable.

Die 100% der Auslastung ist auf $I_{\text{Nenn}}^2 \cdot \sqrt{2}$ normiert.

Beispiel:

Die Motorauslastung beträgt 100%.
Der Befehl wird bei einem Motor AC M2n0290-4-/1-3 (6 Ampere Nennstrom)

Ergebnis:

Die Variable 2 wird mit dem Wert der Motorauslastung geladen.
Die Variable 2 hat den Inhalt 51.

0 Variable 2 = Wert 20

Wert = 21:

Lädt die aktuelle Reglerauslastung in die angegebene Variable.

Die 100% der Auslastung ist auf $I_{\text{Nenn}}^2 \cdot \sqrt{2}$ normiert.

Beispiel:

Die Reglerauslastung beträgt 100%.
Der Befehl wird bei einem Regler 637f -20 ausgeführt (20 Ampere Nennstrom).

Ergebnis:

Die Variable 2 wird mit dem Wert der Reglerauslastung geladen.
Die Variable 2 hat den Inhalt 560.

0 Variable 2 = Wert 21

Wert = 22:

Lädt den aktuellen CAN2-Bus-Knoten-Überwachungsstatus.

Beispiel:

0 Variable 2 = Wert 22

Ergebnis:

Die Variable wird mit dem Wert des CAN2-Bus-Knoten-Überwachungsstatus geladen.
Das Bit der anfragenden Achse ist nicht gesetzt.

Wert = 23:

Lädt die Rotorposition bei positiver Flanke des Referenzsensors bei Referenzfahrt auf Referenzsensor.
Siehe BIAS-Befehl "[Fahre Referenz](#)"
(ab Firmware V8.20 nur bei 637f)

Beispiel:

Ergebnis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
4C	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
D2 = Wert 0...23

4 - 4D BIAS-Befehl "[Variable X] = Achsstatus, AchsNr. Y"

Erklärung:

Der Befehl "[Variable X] = Achsstatus, AchsNr. Y" lädt den aktuellen Status der angegebenen Achse Y in die angegebene Variable X.

Reglertyp:	Parameter	Zulässige Eingabe:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
637f:	X:	0...255	
	Y:	1...3	

Bit	Achsstatus 1-3	
0	Done	Dieser Merker signalisiert, dass kein Multiachs Fahrbefehl ausgeführt wurde bzw. dass ein Multiachs Fahrbefehl abgeschlossen wurde. Dieser Merker ist nach dem Starten eines Fahrbefehls für mindestens die "Pos. erreicht" Zeit (EASYRIDER Parameter) gelöscht. Hinweis: Ein Stop Befehl mit einer AchsNr. != 0 gleicht einem Fahrbefehl!
1	Beschleunigung	Achse X beschleunigt
2	Kontinuierlich	Achse X fährt mit konstanter Geschwindigkeit
3	Verzögerung	Achse X verzögert
4	Flanke Startwinkel	Signalisiert Synchronstart erfolgt (Synchronlauf aktiviert, Start Nocken Überfahren)
5	Trigger Gearbox Ext	Trigger zur Gearbox-Parameterübernahme. Der Merker wird von dem Funktionsblock Gearbox während des Synchronlauf gesetzt und kann im BIAS gelöscht werden.
6	Trigger Kupplung Ext	Trigger für manuelle Parameterübernahme (Der Merker muss im BIAS gesetzt bzw. gelöscht werden) z.B. Synchronisation des Parameterumschaltens mit der Außenwelt
7	Trigger CAM Ext	Trigger für manuelle Parameterübernahme (Der Merker muss im BIAS gelöscht werden) z.B. Synchronisation des Parameterumschaltens mit der Außenwelt
8	Gearbox Trigger	Trigger zur Parameterübernahme der Getriebefunktion. Der Merker wird von dem Getriebefunktionsblock für einen 1 Zyklus gesetzt.
9	Clutch Trigger	Trigger Kupplung- Parameter -Übernahme 1 Zyklus
10	CAM Trigger	Trigger zur Parameterübernahme der CAM-Funktion. Der Merker wird von dem CAM-Funktionsblock für einen 1 Zyklus gesetzt.
11	First CAM	Intern (1.CAM lade)
12	Backstop Error	Fehler Master dreht in falsche Richtung
13	Start Fehler	Startwinkel beim Einschalten schon überschritten
14	Clutch Error	Parameter oder Steuerfehler Kupplung
15	CAM Start Error	Fehler in CAM Daten
16	Start Nocken	Anwahl der Funktion für Synchronlauf Start auf definiertem Masterwinkel (Bit = 1)
17	Start Richtung	Richtung in der der Start auf definiertem Masterwinkel erfolgen soll (0= Positive Zählrichtung)

18	Gearbox Trigger On	Einschalten der Triggerfunktion für die Gearbox. Änderungen des Koppelfaktors werden beim Triggerzeitpunkt (bei Erreichen der Formatlänge (Koppelfaktor Nenner) übernommen. („Gearbox Trigger On“ =1). Für sofortige Übernahme des Koppelfaktors muss der Merker “Bit Gearbox Trigger Ext“ nach dem Schreiben des neuen Koppelfaktors gesetzt werden (Trigger On =0).
19	Backstop On	Koppelt die Synchronisierung bei rückwärts drehendem Master ab, solange bis der Punkt der Auskopplung wieder vorwärts überfahren wird. Vorwärts, Rückwärts wird definiert aus dem Merker “Start Richtung“ Steht die Masterposition hinter der Auskoppelposition Wird das Fehlerbit “Backstop Error“ gesetzt
20	Clutch On	Einschalten der Kupplungsfunktion (Bit 4 =1)
21	Ramp Richtung	Slave Richtung der Kupplungsfunktion
22	Ramp Up	Aktiviert das Auframpen
23	Ramp Down	Aktiviert das Abrampen
24	Ramp Wait	Aktiviert die Wartefunktion
25	Move Linked	Bei MOVE_LINKED = 1, werden die Distanzen Move_Linked_ACC, Move_Linked_Cont und Move_Linked_DEC nacheinander abgefahren.
26	Merge Move Linked	Mit einer positiven Flanke des Merkers “Merge_Linked_Move“ werden die neuen Formate übernommen und ein neues Profil gestartet. Nach dem Neustart wird das Bit “Merge_Linked_Move“ gelöscht.
27	Caming On	Schaltet den CAM-Funktionsblock ein
28	Stop CAM Ende	Stoppt den Synchronlauf am Ende der Kurvenscheibe
29	CAM Backstop <u>Diese Funktion ist zur Zeit in Vorbereitung!!!</u>	Bei rückwärts drehendem Leitwert stoppt der CAM am Profilanfang, solange bis der Leitwert wieder in positiver Richtung über den Ausschaltpunkt gelaufen ist. Mit diesem Merker kann die Abarbeitung des CAM-Profiles in negativer Leitwert Richtung fortgesetzt werden.
30	Reserve	
31	Reserve	

Beispiel: 9 ... 10 [Variable 0] = Achsstatus, AchsNr. = 2 11 Wenn [Variable 0].Bit 12 == 1 dann Springe Fehler 12 ... Fehler: 60 ...	Ergebnis: Der Achsstatus der Achse 2 wird der Variablen 0 zugewiesen. Wenn der Merker Backstop Error gesetzt ist, verzweigt das Programm an die Sprungmarke “Fehler“ ansonsten läuft das Programm bei Zeile 12 weiter.
---	---

Hinweis:

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung.

Befehlskodierung:

Cod e	D1	D2	D3	D4	D5	D6	D7
4D	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
 D2 = Achsnummer: 1...3

4 - 50 BIAS-Befehl "NOP"

Erklärung:

Der Befehl „NOP“ dient als Leerzeile und es wird der darauf folgende Satz bearbeitet. (no operation)

Parameter:

keine

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

0 NOP
 1 NOP
 2 [Variable 0] = 1
 3 Springe 3

Ergebnis:

Es wird für 2 Zyklen kein (NOP) Befehl verarbeitet.

Hinweis: NOP-Zeilen verlangsamen die Programmabarbeitung

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
50	00	00	00	00	00	00	00

4 - 51 BIAS-Befehl "Programmende, Modus ="

Programmende, Modus = X

Erklärung:

Der Befehl "Programmende, Modus =" setzt die Abarbeitung entsprechend dem im Parameter "Modus" angegebenen Wert fort.

Parameter:	Zulässige Eingabe:
X: Modus:	0...6
0 =	Rücksprung auf den Programmstartsatz der aufrufenden Task (BIAS-, SPS- oder Mathematik-Programm)
1 =	Beendet die Abarbeitung eines BIAS-Programms
2 =	Beendet die Abarbeitung eines SPS-Programms
3 =	Beendet die Abarbeitung eines Mathematik-Programms
4 =	Deaktiviert den Regler
5 =	Aktiviert den Regler (nur im SPS-Programm mit "Weiterabarbeitung während Deaktivierung" möglich und wenn Regler durch Modus 4 deaktiviert wurde).
6 =	führt einen Reglerreset (nur im SPS-Programm mit "Weiterabarbeitung während Deaktivierung" möglich)

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

ProgStart:

- 0 [Variable 0] = 0
- 1 [Variable 1] = 0
- 2 Mathematikprogramm MATH
- 3 SPS-Programm SPS
- 4 Programmende, Modus = 1**

SPS:

- 5 [Variable 0] = [Variable 0] + 1
- 6 Programmende, Modus = 0**

MATH:

- 7 [Variable 1] = [Variable 1] + 1
- 8 Programmende, Modus = 0**

Ergebnis:

Nach dem Starten der BIAS-Abarbeitung werden die Mathematik-Task und die SPS-Task gestartet und die BIAS-Abarbeitung anschließend gestoppt. Die Mathematik-Task und die SPS-Task werden weiter zyklisch abgearbeitet. (Rücksprung auf den Programmstartsatz der aufrufenden Task)

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
51	X	00	00	00	00	00	00

Parameter: D1 = Modus 0-6

4 - 52 BIAS-Befehl "Unterprogramm"

Unterprogramm X

Erklärung:

Der Befehl "Unterprogramm X" setzt den BIAS-Abarbeitungszeiger auf die angegebene Satznummer.

Gleichzeitig wird die nächste Satznummer als Rücksprungadresse im Unterprogramm-Stapelspeicher gesichert.

Die Stapeltiefe (d.h. eine Verschachtelung von Unterprogrammen) ist auf **128** Unterprogrammssprünge begrenzt. Ein Unterprogramm **muss** mit dem Befehl "Unterprogrammende" beendet werden.

Parameter:

X: Adresse:

Zulässige Eingabe:

0...1499

Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

ProgStart:

0 Unterprogramm UP_1
1 Sprünge 1

UP_1:

2 Unterprogramm UP_2
3 [Variable 0] = 0
4 Unterprogrammende

UP_2:

5 [Variable 1] = 123
6 Unterprogrammende

Ergebnis:

Nach dem Aktivieren der BIAS-Abarbeitung wird das Unterprogramm 1 aufgerufen. Darin wird das Unterprogramm 2 aufgerufen und nach dem Rücksprung aus den Unterprogrammen das Programm auf Zeile 1 fortgeführt.

Hinweis:

- BIAS-, SPS- und Mathematik-Programme haben von einander getrennte Unterprogramm-Stapelspeicher.
- Bei Verwendung des Befehles "BIAS-Abarbeitungszeiger" wird der BIAS-Unterprogrammstapelspeicher gelöscht!

Befehlskodierung:

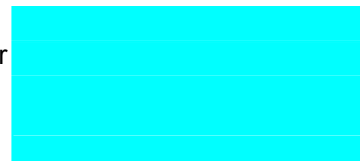
Code	D1	D2	D3	D4	D5	D6	D7
52	X(low)	X(high)	00	00	00	00	00

Parameter: D1,D2 = Sprungadresse: 0...1499

4 - 53 BIAS-Befehl "Unterprogrammende"

Erklärung:

Der Befehl "Unterprogrammende" lädt die im Unterprogramm-Stapelspeicher abgespeicherte Rücksprungadresse in den Satzzeiger und beendet das Unterprogramm der aufrufenden Task.



Parameter:

keine

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

ProgStart:

- 0 Unterprogramm UP_1
- 1 Sprunge 1

UP_1:

- 2 Unterprogramm UP_2
- 3 [Variable 0] = 0
- 4 **Unterprogrammende**

UP_2:

- 5 [Variable 1] = 123
- 6 **Unterprogrammende**

Ergebnis:

Nach dem Aktivieren der BIAS-Abarbeitung wird das Unterprogramm 1 aufgerufen. Darin wird das Unterprogramm 2 aufgerufen und nach dem Rücksprung aus den Unterprogrammen das Programm auf Zeile 1 fortgeführt.

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
53	00	00	00	00	00	00	00

4 - 54 BIAS-Befehl "SPS-Programm"

SPS-Programm X

Erklärung:

Der Befehl "SPS-Programm" setzt den SPS-Abarbeitungszeiger auf die angegebene Satznummer. Mit diesem Befehl kann parallel zur BIAS-Satzabarbeitung ein zyklisches SPS-Programm gestartet werden. Die angegebene Satznummer wird im SPS-Programm beim Befehl "Programmende, Modus =0" als Rücksprungadresse genommen.

Parameter:

X: Adresse:

Zulässige Eingabe:

0...1499

Die maximal zulässige Sprungadresse entspricht der Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

ProgStart:

- 0 [Variable 0] = 0
- 1 [Variable 1] = 0
- 2 Mathematikprogramm MATH
- 3 SPS-Programm SPS**
- 4 Programmende, Modus = 1

SPS:

- 5 [Variable 0] = [Variable 0] + 1
- 6 Programmende, Modus = 0

MATH:

- 7 [Variable 1] = [Variable 1] + 1
- 8 Programmende, Modus = 0

Ergebnis:

Nach dem Starten der BIAS-Abarbeitung werden die Mathematik-Task und die SPS-Task gestartet und die BIAS-Abarbeitung anschließend gestoppt. Die Mathematik-Task und die SPS-Task werden weiter zyklisch abgearbeitet. (Rücksprung auf den Programmstartsatz der aufrufenden Task)

Hinweis:

- Führen Sie den Befehl "SPS-Programm" nicht zyklisch aus, da ansonsten die SPS-Abarbeitung ständig an der angegebenen Satznummer beginnt.
- Innerhalb eines SPS- Programm sind nicht alle BIAS-Befehle erlaubt.
- In der BIAS- Programmdefinition (Parameter "Programmreaktion SPS- Programm") kann die Reaktion des SPS- Programms auf die Deaktivierung der Endstufe eingestellt werden. Damit ist es möglich das SPS- Programm auch während der Deaktivierung des Reglers weiterarbeiten zu lassen.
- Wird in diesem Modus der Befehl " SPS -Programm ..." auf der **1. Zeile** der BIAS- Programmabarbeitung programmiert, so läuft das SPS - Programm auch ohne Aktivierung des Reglers an.
- **Achtung beim Modus "bei Deaktivierung weiterarbeiten"!!**
Beim Senden eines neuen BIAS- Programms mit Änderung der Zeilennummern (Verschiebung von Programmteilen) können unkontrollierte Maschinenbewegungen und Programmreaktionen die Folge sein!

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
54	X(low)	X(high)	00	00	00	00	00

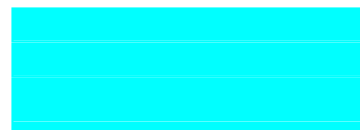
Parameter: D1,D2 = Sprungadresse: 0...1499

4 - 55 BIAS-Befehl "Springe Konst."

Springe Konst.

Erklärung:

Der Befehl "Springe = Konst." setzt den Abarbeitungszeiger auf die angegebene Satznummer.



Parameter	Zulässige Eingabe:	
Konst.:	0...1499 Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

```

0 Wenn Merker 0 == 0 dann Springe 3
1 [Variable 0] = 5
2 Springe 4
3 [Variable 0] = 1
4 [Variable 0] = [Variable 0]* 100
5 Springe [Variable 0]
```

Ergebnis:

Die Abarbeitung wird bei nicht gesetztem Merker 0 im Satz 3 fortgesetzt und die Variable 0 mit 100 multipliziert.

Hinweis:

1. Die maximal zulässige Adresse ist abhängig von der Länge des BIAS-Programms und entspricht der Satznummer des letzten BIAS-Satzes.
2. Alternativ zu den Satznummern können Label (Sprungmarken) verwendet werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
55	X(low)	X(high)	00	00	00	00	00

Parameter: D1, D2 = Sprungadresse: 0...1499

4 - 46 BIAS-Befehl "Springe [Variable X]"

Springe [Variable X]

Erklärung:

Der Befehl "Springe [Variable X]" setzt den Abarbeitungszeiger auf die, dem Inhalt der angegebenen Variablen entsprechenden Satznummer.

Parameter

Zulässige Eingabe:

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

X:

0...255

zulässiger Variableninhalt

0...1499

Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.

Beispiel:

```
0 Wenn Merker 0 == 0 dann Springe 3
1 [Variable 0] = 5
2 Springe 4
3 [Variable 0] = 1
4 [Variable 0] = [Variable 0]* 100
5 Springe [Variable 0]
```

Ergebnis:

Die Abarbeitung wird bei gesetztem Merker 0 im Satz 500 und bei nicht gesetztem Merker 0 im Satz 100 fortgesetzt.

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
56	X	00	00	00	00	00	00

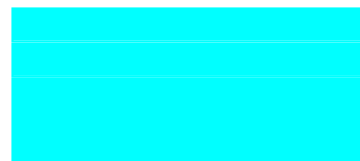
Parameter: D1 = Variablennummer: 0...255

4 - 57 BIAS-Befehl "BIAS – Abarbeitungszeiger = Konst."

BIAS-Abarbeitungszeiger = Konst.

Erklärung:

Der Befehl "BIAS – Abarbeitungszeiger = Konst." setzt den BIAS-Abarbeitungszeiger vom SPS- oder Mathematik-Programm aus auf den abgegeben Wert.



Parameter:

Adresse:

Zulässige Eingabe:

0...1499

Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

0 SPS-Programm 50

Stoppe:

10 Stoppe Achse, Modus 1, a- = 10000 rpm/s

11 Springe 11

...

50 Wenn Istpos.1 > 500000 INKR dann Springe Stop

51 Wenn Istpos.1 < 0 INKR dann Springe Stop

52 Programmende

Stop:

53 BIAS-Abarbeitungszeiger = Stoppe

54 Programmende

Ergebnis:

Die Abarbeitung des BIAS-Programms wird im Satz 10 (Label Stoppe) fortgesetzt sowie die Istposition grösser 500000 Inkremente oder kleiner 0 Inkremente ist.

Hinweis:

1. Die maximal zulässige Adresse ist abhängig von der Länge des BIAS-Programms und entspricht der Satznummer des letzten BIAS-Satzes.
2. Alternativ zu den Satznummern können Label (Sprungmarken) verwendet werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
57	X(low)	X(high)	00	00	00	00	00

Parameter: D1, D2 = Sprungadresse: 0...1499

4 - 58 BIAS-Befehl "Warte auf "Position erreicht""

Warte auf "Pos. erreicht"

Erklärung:

Der Befehl "Warte auf Position erreicht" wartet auf die Beendigung der Fahrprofilgenerierung einer Positionierung und darauf das das Statusbit „BIT_POS_ERREICHT“ auf 1 geht.
Danach wird der nächste BIAS-Befehl ausgeführt.

Parameter:

keine

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

```
0 Weg = 10000 INKR
1 "Pos. erreicht" Fenster = 100
2 Fahre Position
3 Warte auf "Position erreicht"
4 ...
```

Ergebnis:

Die BIAS-Abarbeitung wird im Satz 4 erst fortgesetzt nachdem die Positionierung beendet ist und sich die Istposition im Bereich von 9900 bis 10100 Inkremente befindet.

Hinweis:

- Das Bit „Position erreicht“ wird auch bei aktivem Endschalter (richtungsabhängig) gesetzt!
- Das Bit ist im Modus 8 (Multiachsfunktion) immer gesetzt. Im Multiachsbetrieb muss der Achsstatus abgefragt werden.
siehe auch:
 - BIAS-Befehl "[Wenn Status == 8 dann Springe](#)"

BIAS-Befehl "[Position erreicht Fenster = konst.](#)"

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
58	00	00	00	00	00	00	00

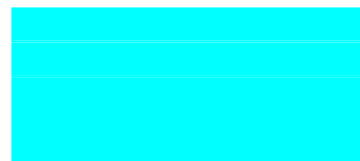
Parameter: -

4 - 59 BIAS-Befehl "Wartezeit = Konst."

Wartezeit X ms

Erklärung:

Der Befehl "Wartezeit = Konst." startet die angegebene Wartezeit. Erst nach dem Ablauf dieser Wartezeit wird der nächste BIAS-Befehl ausgeführt.



Parameter:

Konst.

Zulässige Eingabe:

0...131070 ms

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

- 1 ...
- 2 Fahre Position
- 3 Warte auf "Position erreicht"
- 4 **Warte Zeit 1000 ms**
- 5 ...

Ergebnis:

Die BIAS-Abarbeitung wird 1 Sekunde nach der Positionierung im Satz 5 fortgesetzt.

Hinweis:

- Bei der Eingabe sind nur gerade Wartezeiten zulässig. Ungerade Wartezeiten werden abgerundet.
- Bei einer Satznummeranwahl über Strobefunktion wird die Wartezeit gelöscht.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
59	X(low)	X(high)	00	00	00	00	00

Parameter: D1, D2 = Wartezeit: 0...131070

4 - 5A BIAS-Befehl "Wartezeit = [Variable X]"

Wartezeit [Variable X]

Erklärung:

Der Befehl "Wartezeit = [Variable X]" startet die in Variable X angegebene Wartezeit. Erst nach dem Ablauf dieser Wartezeit wird der nächste BIAS-Befehl ausgeführt.

Parameter

Zulässige Eingabe:

X:

0...255

zulässiger Variableninhalt:

0...65535

- BIAS-Programm
 SPS-Programm
 Mathematik-Programm

Beispiel:

- 1 [Variable 0] = 1000
- 2 Fahre Position
- 3 Warte auf "Position erreicht"
- 4 **Wartezeit [Variable 0]**
- 5 ...

Ergebnis:

Die BIAS-Abarbeitung wird 1 Sekunde nach der Positionierung im Satz 5 fortgesetzt.

Hinweis:

1. Der Variableninhalt wird als Zeitangabe in ms interpretiert.
 Beim 637+ und 637f wird in 1 ms Stufen gezählt
 Beim 631,635,637 wird in 2 ms Stufen gezählt. Ungerade Wartezeiten werden abgerundet.
2. Bei einer Satznummeranwahl über Strobefunktion wird die Wartezeit gelöscht.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
5A	X	00	00	00	00	00	00

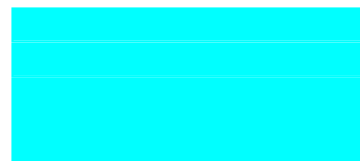
Parameter: D1 = Variablennummer: 0...255

4 - 5B BIAS-Befehl "BIAS-Abarbeitungszeiger = [Variable X]"

BIAS-Abarbeitungszeiger = [Variable X]

Erklärung:

Der Befehl "BIAS-Abarbeitungszeiger = [Variable X]" setzt den BIAS-Abarbeitungszeiger vom SPS-Programm aus auf den in der Variablen X angegebenen Wert.



Parameter	Zulässige Eingaben:	
X:	0...255 Die zulässige Sprungadresse entspricht der Länge des BIAS-Programms.	<input type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

```

0 [Variable 0] = 10
1 SPS-Programm 50
Stoppe:
10 Stoppe Achse, Modus 1, a- = 10000 rpm/s
11 Springe 11
...
50 Wenn Istpos.1 > 500000 INKR dann
   Springe Stop
51 Wenn Istpos.1 < 0 INKR dann Springe
   Stop
52 Programmende
Stop:
53 BIAS-Abarbeitungszeiger = [Variable 0 ]
54 Programmende
    
```

Ergebnis:

Die Abarbeitung des BIAS-Programms wird im Satz 10 fortgesetzt sowie die Istposition grösser 500000 Inkremente oder kleiner 0 Inkremente ist.

Hinweis:

1. Die maximal zulässige Adresse ist abhängig von der Länge des BIAS-Programms und entspricht der Satznummer des letzten BIAS-Satzes.
2. Alternativ zu den Satznummern können Label (Sprungmarken) verwendet werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
5B	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255

4 - 5C BIAS-Befehl "Springe [Var.[X]]; Länge = Y ; ab Zeile"

Springe [Var.[X]]; Länge = Y ; ab Z

Erklärung:

Der Befehl "Springe [Var.[X]]; Länge = Y ; ab Zeile" erlaubt Programmverzweigungen abhängig vom Inhalt der Variablen X. Die Variable X darf den Wert Y nicht überschreiten. Die Befehle ab der Zeile Z dürfen nur Sprungbefehle ("Springe Konst." bzw. "Springe [Variable X]") sein! Der 1. Befehl am Sprungziel wird noch im gleichen Programmzyklus abgearbeitet.

Parameter:	Zulässige Eingabe:	
X:	0...255 (Variableninhalt X <= Y)	<input checked="" type="checkbox"/> BIAS-Programm
Y:	0...99	<input checked="" type="checkbox"/> SPS-Programm
Z:	0...1499 (Y+Zeile)<1499	<input type="checkbox"/> Mathematik-Programm

Beispiel:

10 Springe [Var.[64]]; Länge=6 ; ab JUMP
JUMP:

```
11 Springe STANDSTILL
12 Springe MOVE_ABS
13 Springe MOVE_REL
14 Springe MOVE_HOME
15 Springe MOVE_INF
16 Springe STOP_AXIS
```

STANDSTILL:

```
17 Springe 10
```

MOVE_ABS:

```
18 Starte Achse
19 Fahre Position
```

Ergebnis:

Mit dem Abarbeiten der Zeile 10 wird abhängig vom Inhalt der Variablen 64 der Befehl am angegebenen Sprungziel ausgeführt.

Hat die Variable 64 den Wert 1 wird ein Sprung zur Sprungmarke MOVE_ABS durchgeführt und der Befehl "Starte Achse" auf Zeile 18 ausgeführt.

Hinweis:

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung.
- Dieser Befehl steht beim 631, 635, 637 ab der Firmware V6.19 zur Verfügung.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
5C	X	Y	Z(low)	Z(high)	00	00	00

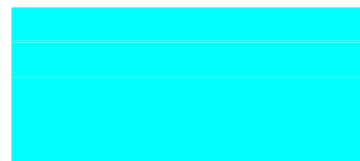
Parameter: D1 = Variablennummer: 0...255
D2 = Länge: 0...99
D3,D4 = Sprungadresse: 0...1499

4 - 5D BIAS-Befehl "Bearbeite X Befehle"

Bearbeite X Befehle

Erklärung:

Der Befehl "Bearbeite X Befehle" legt in dem Parameter X die Anzahl der Befehle fest, die in einem Zyklus bearbeitet werden.
Der Befehl selbst wird nicht mitgezählt!



Parameter	Zulässige Eingabe:	
X:	0...9	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

- 0 Wartezeit 1000 ms
- 1 Bearbeite 6 Befehle**
- 2 Weg = 10000 INKR
- 3 Geschwindigkeit = 1000 rpm
- 4 Beschleunigung = 5000 rpm/s
- 5 Verzögerung = 5000 rpm/s
- 6 Starte Achse
- 7 Fahre Position; AchsNr. = 0 , Ziel = 0
- 8 Warte auf "Pos. erreicht"
- 9 NOP

Ergebnis:

Nach der Wartezeit von 1 Sekunde wird der Start im nächsten Zyklus sofort ausgeführt. Die Befehle 2-7 werden in einem Zyklus verarbeitet.

Hinweis:

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung
- Der Befehl wird ignoriert falls er innerhalb eines schon gestarteten Befehlszyklus vorkommt. In diesem Fall wird sofort der nächste Befehl ausgeführt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
5D	X	00	00	00	00	00	00

Parameter: D1: Befehlsanzahl 0...9

4 - 5F BIAS-Befehl "Virtuelles Programm"

Virtuelles Programm a= W, b = X, c = Y, d = Z

Erklärung:

Der Befehl "Virtuelles Programm" ist zu Testzwecken implementiert und **nicht zu benutzen!**

Parameter	Zulässige Eingabe	
W:	0...255	<input checked="" type="checkbox"/> BIAS-Programm
X:	0...255	<input type="checkbox"/> SPS-Programm
Y:	0...255	<input type="checkbox"/> Mathematik-Programm
Z:	0...255	
Weiter Informationen finden Sie in der Dokumentation Configuration trajectory generator		

Beispiel:

Ergebnis:

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
5F	W	X	Y	Z	00	00	00

Parameter: D1 = Variablennummer: 0...255
 D2 = Variablennummer: 0...255
 D3 = Variablennummer: 0...255
 D4 = Variablennummer: 0...255

4 - 60 BIAS-Befehl "Merker X = Konst."

Merker X = Konst.

Erklärung:

Der Befehl "Merker X = Konst." weist dem Merker X die Konstante 0 oder 1 zu.

Parameter	Zulässige Eingabe	
X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm
konst.:	0/1	<input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 Merker 255 = 1

Ergebnis:

Der Merker 255 wird gesetzt.

Hinweis:

- Nach dem Einschalten des Reglers sind alle Merker gelöscht.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
60	X	Y	00	00	00	00	00

Parameter: D1 = Merkenummer 0..255
D2 = Konst. 0 / 1

4 - 61 BIAS-Befehl "Wenn Merker X ? Konst. dann Springe"

Wenn Merker X ? Konst. dann Springe Y

Erklärung:

Der Befehl "Wenn Merker X ? konst dann Springe" erlaubt Programmverzweigungen durch dem Vergleich des angewählten Merkers X mit der Konstanten 0 bzw. 1.
Trifft der Vergleich zu wird die Abarbeitung an der angegebenen Adresse fortgeführt.

Parameter	Zulässige Eingabe:	Bemerkung:	
X:	0...255		<input checked="" type="checkbox"/> BIAS-Programm
?:	== !=	gleich ungleich	<input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
Konst.:	0/1		
Y:	0...1499		
Adresse:	Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden)		

Beispiel:

0 Wenn Merker 0 == 1 dann Springe 3

- 1 Ausgang 12 = 0
- 2 Springe 4
- 3 Ausgang 12 = 1
- 4 Springe 4

Ergebnis:

Ist der Merker 0 gesetzt, wird der Ausgang 12 auf gesetzt.
(Ausgang 12 muss als BIAS-Ausgang konfiguriert sein.)

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
61	X	?	Konst.	Y(low)	Y(high)	00	00

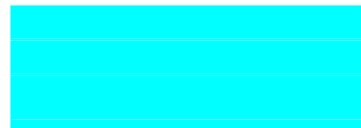
Parameter: D1 = Merker Nummer: 0...255
D2 = Bedingung: == (0) / != (1)
D3 = Konstante: 0/1
D4, D5 = Sprungadresse: 0..1499

4 - 62 BIAS-Befehl "Merker X = Merker Y"

Merker X = Merker Y

Erklärung:

Der Befehl "Merker X = Merker Y" kopiert den Zustand des Merkers Y in den Merker X.



Parameter

Zulässige Eingabe:

X:

0...255

BIAS-Programm

Y:

0...255

SPS-Programm

Mathematik-Programm

Beispiel:

0 Merker 0 = Merker 1

Ergebnis:

Der Merker 1 wird in den Merker 0 kopiert.

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
62	X	Y	00	00	00	00	00

Parameter: D1 = Merker Nummer 0...255
D2 = Merker Nummer 0...255

4 - 63 BIAS-Befehl "Merker X = Eingang Y"

Merker X = Eingang Y

Erklärung:

Der Befehl "Merker X = Eingang Y" kopiert den Zustand des Eingang Y in den Merker X.

Reglertyp:	Parameter:	Zulässige Eingabe:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	X:	0...255	
631:	Y:	7, 8, 9, 10 Pinnr. an X10-Klemme	
635/637/ 637+/637f:	Y:	2, 4, 11, 15, 22, 24, 25 Pinnr. an X10- Buchse	
	Teilbestückte E/A-Optionsplatine RP-EA5 (Steckplatz B)	101, 102, 106, 107, 108 Pinnr. an COM2 Sub-D-Buchse	
	Vollbestückte E/A-Optionsplatine RP-EAE (Steckplatz C)	201-208, 211-216 Pinnr. an X200 Sub-D-Buchse	
637f:	2 x CAN + 4 Ausgänge und 4 Eingänge RP-2C8 (Steckplatz B):	121 – 124 Pinnr. an X120 Klemme	
	Y: 2 Eingänge RP-SBT (Steckplatz C)	91, 93 Pinnr. an X290 Klemme 91: Regleraktivierung 93. Sicherer Halt	

Beispiel:

0 Merker 0 = Eingang 2

Ergebnis:

Der "Merker 0 spiegelt den Zustand des Eingang X10.2 wieder.

Hinweis: Die Eingänge können unabhängig von ihrer Konfiguration abgefragt werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
63	X	Y	00	00	00	00	00

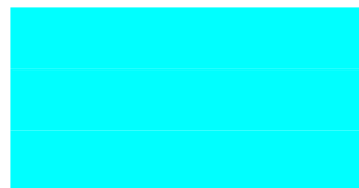
Parameter: D1 = Merker Nummer: 0...255
D2 = Nummer des Eingangs: (siehe [Erklärung](#))

4 - 64 BIAS-Befehl "Merker X = Ausgang Y"

Merker X = Ausgang Y

Erklärung:

Der Befehl "Merker X = Ausgang Y" kopiert den Zustand des Ausgangs Y in den Merker X.



Reglertyp:	Parameter:	Zulässige Eingabe:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
631:	X:	0...255	
	Y:	5, 6 Pinnr. an X10-Klemme	
635/637/ 637+/637f	Y:	8, 12, 13, 20, 23 Pinnr. an X10- Buchse	
	X: Teilbestückte E/A-Optionsplatine RP-EA5 (Steckplatz B)	109, 110 Pinnr. an COM2 Sub-D-Buchse	
	X: Vollbestückte E/A-Optionsplatine RP-EAE (Steckplatz C)	209-210, 217-224 Pinnr. an X200 Sub-D-Buchse	
637f	X: 2 x CAN + 4 Ausgänge und 4 Eingänge RP-2C8 (Steckplatz B):	125 – 128 Pinnr. an X120 Klemme	

Beispiel:

0 Merker 0 = Ausgang 12

Ergebnis:

Der Merker 0 spiegelt den Zustand des Ausgangs X10.12 wieder.

Hinweis: Die Ausgänge können unabhängig von ihrer Konfiguration abgefragt werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
64	X	Y	00	00	00	00	00

Parameter: D1 = Merker Nummer: 0...255
 D2 = Nummer des Ausgangs: (siehe [Erklärung](#))

4 - 65 BIAS-Befehl "Merker X = Merker Y & Merker Z"

Merker X = Merker Y & Merker Z

Erklärung:

Der Befehl "Merker X = Merker Y & Merker Z" kopiert das Ergebnis der logischen UND- Verknüpfung des Merkers Y und Merkers Z in den Merker X.

Parameter	Zulässige Eingabe:	
X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm
Y:	0...255	<input checked="" type="checkbox"/> SPS-Programm
Z:	0...255	<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 Merker 0 = 0

1 Merker 1 = 1

2 Merker 2 = Merker 1 & Merker 0

Ergebnis:

Merker 2 = 0

Hinweis:

Verknüpfungstabelle

:

X	=	Y	&	Z
0	=	0	&	0
0	=	0	&	1
0	=	1	&	0
1	=	1	&	1

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
65	X	Y	Z	00	00	00	00

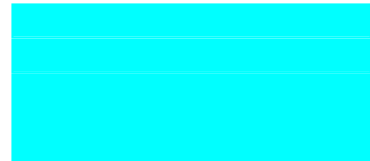
Parameter: D1 = Merker Nummer 0...255
 D2 = Merker Nummer 0...255
 D3 = Merker Nummer 0...255

4 - 66 BIAS-Befehl "Merker X = Merker Y | Merker Z"

Merker X = Merker Y | Merker Z

Erklärung:

Der Befehl "Merker X = Merker Y | Merker Z" kopiert das Ergebnis der logischen ODER- Verknüpfung des Merkers Y und Merkers Z in den Merker X.



Parameter

Zulässige Eingabe:

X: 0...255
Y: 0...255
Z: 0...255

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

0 Merker 0 = 0
1 Merker 1 = 1
2 **Merker 2 = Merker 1 | Merker 0**

Ergebnis:

Merker 2 = 1

Hinweis:

Verknüpfungstabelle

:

X	=	Y		Z
0	=	0		0
1	=	0		1
1	=	1		0
1	=	1		1

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
66	X	Y	Z	00	00	00	00

Parameter: D1 = Merker Nummer 0...255
D2 = Merker Nummer 0...255
D3 = Merker Nummer 0...255

4 - 67 BIAS-Befehl "Merker X = Merker Y ^ Merker Z"

Merker X = Merker Y ^ Merker Z

Erklärung:

Der Befehl "Merker X = Merker Y ^ Merker Z" kopiert das Ergebnis der logischen EXCLUSIV-ODER- Verknüpfung des Merkers Y und Merkers Z in den Merker X.

Parameter

Zulässige Eingabe:

X: 0...255
Y: 0...255
Z: 0...255

BIAS-Programm
 SPS-Programm
 Mathematik-Programm

Beispiel:

0 Merker 0 = 0
1 Merker 1 = 1
2 **Merker 2 = Merker 1 ^ Merker 0**

Ergebnis:

Merker 2 = 0

Hinweis:

Verknüpfungstabelle

:

X	=	Y	^	Z
0	=	0	^	0
1	=	0	^	1
1	=	1	^	0
0	=	1	^	1

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
67	X	Y	Z	00	00	00	00

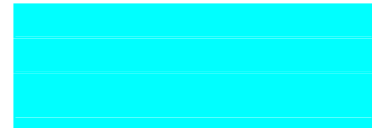
Parameter: D1 = Merker Nummer 0...255
D2 = Merker Nummer 0...255
D3 = Merker Nummer 0...255

4 - 68 BIAS-Befehl "Merker X = !Merker Y"

Merker X = !Merker Y

Erklärung:

Der Befehl "Merker X = !Merker Y" kopiert den negierten Zustand des Merkers Y in den Merker X.



Parameter	Zulässige Eingabe:	
X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm
Y:	0...255	<input checked="" type="checkbox"/> SPS-Programm
		<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 Merker 0 = ! Merker 0

Ergebnis:

Der Merker 0 wurde negiert.

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
68	X	Y	00	00	00	00	00

Parameter: D1 = Merker Nummer 0...255
D2 = Merker Nummer 0...255

4 - 69 BIAS-Befehl "Merker X = Status Y"

Merker X = Status Y

Erklärung:

Der Befehl "Merker X = Status Y" kopiert den aktuellen Status Y in den Merker X.

Parameter	Zulässige Eingabe:	Bezeichnung	
X:	0...255		<input checked="" type="checkbox"/> BIAS-Programm
Y:	0	Position erreicht	<input checked="" type="checkbox"/> SPS-Programm
	1	Endstufe deaktiviert	<input checked="" type="checkbox"/> Mathematik-Programm
	2	Start Format (Synchron)	
	3	intern (knock down bit)	
	4	Offsetkorrektur ausgeführt	
	5	Trigger Profilumschaltung	
	6	Druckmarkenfehler	
	7	Referiert	
	8	Zielposition erreicht	
	9	IBT Datentransfer läuft	
	10	Achse in Lageregelung (Phase Fertig)	
	11	Achse in Bremsrampe	
	12	Achse im Stillstand + Rampenfilterfunktion abgeschlossen (neuer Rampenfilterwert kann geladen werden.	
	13	Tabelle speichern läuft (nur bei 637f)	
	14	pos. Positionsbegrenzung (nur bei 637f)	
	15	neg. Positionsbegrenzung (nur bei 637f)	
	16	Endschalter erkannt	
	17	Geschwindigkeitsbegrenzung überschritten (nur bei 637f)	

Beispiel:

0 Merker 1 = Status 11

Ergebnis:

Der Merker 1 zeigt an ob sich der Lagesollwert in der Bremsrampe befindet.

Hinweis:

- Das Bit 8 "Zielposition erreicht" ist im Modus 8 (Multiachsfunktion) immer gesetzt. Im Multiachsbetrieb muss der Achsstatus abgefragt werden. "[Variabe X] = Achsstatus, Achs Nr."
- Der Status 2 (Synchronlauf Formatumschaltung) ist nur einen BIAS-Befehl lang gesetzt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
69	X	Y	00	00	00	00	00

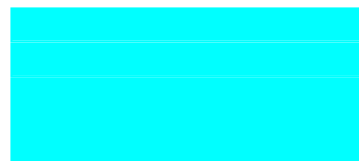
Parameter: D1 = Merker Nummer: 0...255
D2 = Status: 0...17

4 - 6A BIAS-Befehl "Wenn Status X ? Konst. dann Springe"

Wenn Status X ? Konst. dann Springe Y

Erklärung:

Der Befehl "Wenn Status X ?" erlaubt Programmverzweigungen durch den Vergleich des angewählten Status X mit der Konstanten 0 bzw. 1. Trifft der Vergleich zu wird die Abarbeitung an der angegeben Adresse fortgesetzt.



Parameter	Zulässige Eingabe:	Bezeichnung	<input checked="" type="checkbox"/> BIAS-Programm
Y:	0	Position erreicht	<input checked="" type="checkbox"/> SPS-Programm
	1	Endstufe deaktiviert	<input checked="" type="checkbox"/> Mathematik-Programm
	2	Start Format (Synchron)	
	3	intern (knock down bit)	
	4	Offsetkorrektur ausgeführt	
	5	Trigger Profilumschaltung	
	6	Druckmarkenfehler	
	7	Referiert	
	8	Zielposition erreicht	
	9	IBT Datentransfer läuft	
	10	Achse in Lageregelung (Phase Fertig)	
	11	Achse in Bremsrampe	
	12	Achse im Stillstand + Rampenfilterfunktion abgeschlossen (neuer Rampenfilterwert kann geladen werden.	
	13	Tabelle speichern läuft (nur bei 637f)	
	14	pos. Positionsbegrenzung (nur bei 637f)	
	15	neg. Positionsbegrenzung (nur bei 637f)	
	16	Endschalter erkannt	
17	Geschwindigkeitsbegrenzung überschritten (nur bei 637f)		
?:	==	gleich	
	!=	ungleich	
Konst.:	0/1		
Y:	0...1499		
Adresse:	Die maximal zulässige Sprungadresse entspricht der Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.		

Beispiel:

Ergebnis:

Hinweis:

- Das Bit 8 "Zielposition erreicht" ist im Modus 8 (Multiachsfunktion) immer gesetzt. Im Multiachsbetrieb muss der Achsstatus abgefragt werden. "[Variabe X] = Achsstatus, Achs Nr."
- Der Status 2 (Synchronlauf Formatumschaltung) ist nur einen BIAS-Befehl lang gesetzt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
6A	X	?	Konst	Y(low)	Y(high)	00	00

Parameter: D1 = Status 0...17
D2 = Bedingung: == (0)/ != (1)
D3 = Konstante: 0/ 1
D4, D5 = Sprungadresse: 0...1499

4 - 6B BIAS-Befehl "Modus X = Konst."

Erklärung:

Der Befehl "Modus X = Konst." schaltet Zusatzfunktionen frei.

Parameter

X:

Konst.:

Zulässige Eingabe:

0...255

231

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Modus

Zusatzfunktion

0	Schiebefaktoren (nur beim 637f)	
4	0 = Rampensteilheit standard	1 = Rampensteilheit vervierfacht (nur beim 637f)
8	0 = Multiachsbetrieb aus	1 = Multiachsbetrieb (nur beim 637f)
22	0 = CAN2-BUS-Knoten-Überwachungswert zurücksetzen (nur beim 637f mit RP 2CA/ RP 2C8)	

Beispiel:

0 Modus 8 = 1

1 [Variable 10] = 200000

2 [Variable 11] = 1000

3 [Variable 12] = 2000

4 [Variable 13] = 2000

5 [Variable 14] = 10

6 Lade Parametersatz 0 = [Variable [14]]

7 Starte Achse

8 Fahre Position; AchsNr. = 1 , Ziel = 1

9 [Variable 0] = Achsstatus, AchsNr. = 1

10 Wenn [Variable 0].Bit 0 == 0 dann Springe 9

11 ...

Hinweis:

Ergebnis:

Der Multiachsmodus wird aktiviert. Die Variablen definieren den Parametersatz für eine Positionierung.

Die Achse fährt auf die Position 200000 Inkremente und nach Erreichen dieser Position wird die Zeile 11 angesprungen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
6B	X	Y (low)			Y (high)	00	00

Parameter: D1 = Modus: 0...255
D2-D5 = Konstante: ± 231

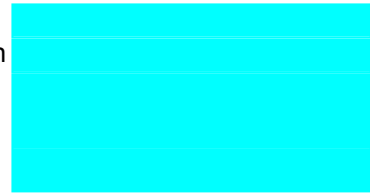
4 - 6C BIAS-Befehl "Merker X = [Variable Y], Anzahl Z"

Merker X = [Variable Y], Anzahl = Z

Erklärung:

Der Befehl "Merker X = [Variable Y], Anzahl Z" kopiert die Z niederwertigsten Bits der angegeben Variablen Y in den Merkerbereich beginnend ab dem Merker X.

Merker X = Bit 0



Parameter:	Zulässige Eingabe:	
X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm
Y:	0...255	<input checked="" type="checkbox"/> SPS-Programm
Z:	1...32	<input type="checkbox"/> Mathematik-Programm
beachte: (X+Z)<257		

Beispiel:

Die 4 niederwertigsten Bit der Variablen 0 werden auf die Merker 0 – 3 kopiert.

0 Variable 0 = 15
 1 Merker 0 = Variable 0, Anzahl 4

Ergebnis:

Merker 0 = 1
 Merker 1 = 1
 Merker 2 = 1
 Merker 3 = 1

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
6C	X	Y	Z	00	00	00	00

Parameter: D1 = Merker Nummer: 0...255
 D2 = Variablennummer: 0...255
 D3 = Bitanzahl: 1...32

4 - 6D BIAS-Befehl “[Variable X]. Bit [Y] = Konst.“

[Variable X].Bit Y = Konst. ,[Merker Z]

Erklärung:

Der Befehl “[Variable X]. Bit [Y] = Konst. , Merker Z“ setzt oder löscht das ausgewählte Bit Y der ausgewählten Variablen X.

Ist in der Konstanten der Wert 255 eingetragen, so wird der Inhalt des Merkers Z in die Bitposition der Variablen kopiert.

Parameter	Zulässige Eingabe:	
X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm
Y:	0...31	<input checked="" type="checkbox"/> SPS-Programm
Konst.:	0/1 , 255	<input checked="" type="checkbox"/> Mathematik-Programm
Z:	0...255	

Beispiel:

0 Merker 10 = 1

1 [Variable 0].Bit 15 = 0 ,[Merker 0]

2 [Variable 1].Bit 1 = 255 ,[Merker 10]

Ergebnis:

Bit 15 der Variable 0 wird gelöscht und

Bit 1 der Variable 1 wird (durch Merker 10 = 1) gesetzt.

Hinweis

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung.
- Dieser Befehl steht beim 631, 635, 637 ab der Firmware V6.19 zur Verfügung.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
6D	X	Y	Konst.	Z	00	00	00

Parameter: D1 = Variablennummer: 0...255
 D2 = Bitnummer 0...31
 D3 = Konstante: 0/1 , 255
 D4 = Merkenummer: 0...255

4 - 6E BIAS-Befehl "Wenn[Variable X]. Bit Y ? Konst. dann Springe"

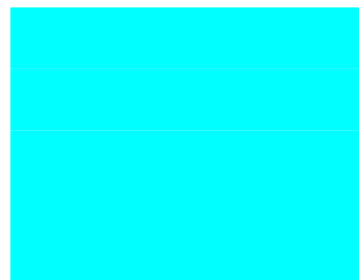
Wenn [Variable X].Bit Y ? dann Springe Z

Erklärung:

Der Befehl " Wenn[Var. X]. Bit Y ? Konst. dann Springe" erlaubt Programmverzweigungen durch den Vergleich des angewählten Bits Y der angewählten Variable X mit einer Konstanten.

Ist die Bedingung erfüllt, wird die BIAS-Abarbeitung an der angegebenen Adresse fortgesetzt.

Ansonsten wird der folgende BIAS-Befehl ausgeführt.



Parameter	Zulässige Eingabe:	Bezeichnung:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
X:	0...255		
Y:	0...31		
?:	== !=	gleich ungleich	
Z: Adresse:	0...1499 Die maximal zulässige Sprungadresse entspricht der Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.		

Beispiel:

```
LOOP:
  0 Wenn [Variable 10 ].Bit 3 == 0 dann Springe LOOP
  1....
```

Ergebnis:

Das Programm wird erst bei Zeile 1 weiterbearbeitet falls Bit 3 der Variable 10 auf 1 geht.

Hinweis:

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung.
- Dieser Befehl steht beim 631, 635, 637 ab der Firmware V6.19 zur Verfügung.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
6E	X	Y	?	Z(low)	Z(high)	00	00

Parameter: D1 = Variablennummer 0...255
 D2 = Bitnummer 0...31
 D3 = Bedingung == (0)/ != (1)
 D4, D5 = Sprungadresse 0...1499

4 - 6F BIAS-Befehl "Achszustand, AchsNr. X, Bit Y =Konst.[Merker Z]"

Achszustand, AchsNr. X, Bit Y = Konst, [Merker Z]

Erklärung:

Der Befehl "Achszustand, AchsNr. X , Bit Y = Konst. , [Merker Z]" setzt oder löscht das angegebene Bit des Achszustandes der angegebenen Achse X.

Ist in der Konstanten der Wert 255 eingetragen, so wird der Inhalt des Merkers Z in die Bitposition des Achszustandes kopiert.

Der Befehl wird im Zusammenhang mit der Multiachsen-Funktion des 637f Reglers verwendet.

Reglertyp	Parameter	Zulässige Eingabe:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
637f	X:	1..3	
	Y:	5-7, 21-24, 26, 28	
	Konst.	0/1/ 255	
	Z:	0...255	

Parameter Y:

5	Trigger Gearbox Ext	Trigger zur Gearbox-Parameterübernahme. Der Merker wird von dem Funktionsblock Gearbox während des Synchronlauf gesetzt und kann im BIAS gelöscht werden.
6	Trigger Kupplung Ext	Trigger für manuelle Parameterübernahme (Der Merker muss im BIAS gesetzt bzw. gelöscht werden) z.B. Synchronisation des Parameterumschaltens mit der Außenwelt
7	Trigger CAM Ext	Trigger für manuelle Parameterübernahme (Der Merker muss im BIAS gelöscht werden) z.B. Synchronisation des Parameterumschaltens mit der Außenwelt

21	Ramp Richtung	Slave Richtung der Kupplungsfunktion
22	Ramp Up	Aktiviert auframpen
23	Ramp Down	Aktiviert abrampen
24	Ramp Wait	Aktiviert Wartezeit

26	Merge Move Linked	Mit einer positiven Flanke des Merkers "Merge_Linked_Move" werden die neuen Formate übernommen und ein neues Profil gestartet. Nach dem Neustart wird das Bit "Merge_Linked_Move" gelöscht.
----	-------------------	---

28	Stop CAM Ende	Stoppt den Synchronlauf am Ende der Kurvenscheibe
----	---------------	---

Alle nicht aufgeführten Merker dürfen und können nicht verändert werden.

Beispiel:

```

35 ...
36 [Variable 1 ] = Achsstatus, AchsNr. = 1
37 Wenn [Variable 1 ].Bit 6 == 0 dann Springe SYNC_SSI1
38 Achszustand, AchsNr. 1 , Bit 23 = 1 , [Merker 0 ]
39 Achszustand, AchsNr. 1 , Bit 24 = 1 , [Merker 0 ]
40 Achszustand, AchsNr. 1 , Bit 6 = 0 , [Merker 0 ]
41 ..

```

Ergebnis:

Die Merker "Ramp Down" und "Ramp Wait" werden gesetzt und der Merker "Trigger Kupplung Ext" wird gelöscht, wenn das Bit 6 des Achsstatus der Achse 1 gesetzt ist.

SYNC_SSI1:

Hinweis:

- Dieser Befehl steht nur beim 637f mit einer Firmware V8.19 zur Verfügung
- Die Funktion "Synchronlauf" erhält ihre Steuerparameter vor dem Start aus den Variablen, die im Parametersatz angegeben sind und im Zustand "Aktiv" aus dem Achszustand.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
6F	X	Y	Konst.	Z	00	00	00

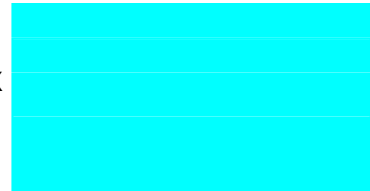
Parameter: D1 = Achsnummer 0...3
D2 = Bitnummer 5-7, 21-24, 28
D3 = Konstante 0/1, 255
D4 = Merker Nummer 0...255

4 -70 BIAS-Befehl "Wenn Eingang X ? Konst. dann Springe"

Wenn Eingang X ? Konst. dann Springe Y

Erklärung:

Der Befehl "Wenn Eingang X ? Konst. dann Springe" erlaubt Programmverzweigungen durch den Vergleich des angewählten Eingangs X mit der Konstanten 0 bzw. 1. Trifft der Vergleich zu wird die Abarbeitung an der angegebenen Adresse fortgeführt.



Reglertyp:	Parameter:	Zulässige Eingaben:	
631:	X:	7, 8, 9, 10 Pinnr. an X10-Klemme	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
635/637/ 637+/637f:	X:	2, 4, 11, 15, 22, 24, 25 Pinnr. an X10- Buchse	
	X: Teilbestückte E/A-Optionsplatine RP-EA5 (Steckplatz B)	101, 102, 106, 107, 108 Pinnr. an COM2 Sub-D-Buchse	
	X: Vollbestückte E/A-Optionsplatine RP-EAE (Steckplatz C)	201-208, 211-216 Pinnr. an X200 Sub-D-Buchse	
637f:	X: 2 x CAN + 4 Ausgänge und 4 Eingänge RP-2C8 (Steckplatz B):	121 – 124 Pinnr. an X120 Klemme	
	X: 2 Eingänge RP-SBT (Steckplatz C)	91, 93 Pinnr. an X290 Klemme 91: Regleraktivierung 93. Sicherer Halt	
	?	== gleich != ungleich	
	Konst.	0/1	
	Y Adresse:	0...1499 Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.	

Beispiel:

LOOP:

```

0 Wenn Eingang 2 == 1 dann
  Springe E2_high
1 [Variable 0 ] = [Variable 0 ] - 1
2 Springe LOOP

```

E2_high:

```

3 [Variable 0 ] = [Variable 0 ] + 1
4 Springe LOOP

```

Ergebnis:

Ist der Eingang 2 eingeschaltet wird die Variable 0 zyklisch um 1 inkrementiert .
Ist der Eingang 2 ausgeschaltet wird die Variable 0 zyklisch um 1 dekrementiert .

Die Eingänge können unabhängig von ihrer Konfiguration abgefragt werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
70	X	?	Konst	Y(low)	Y(high)	00	00

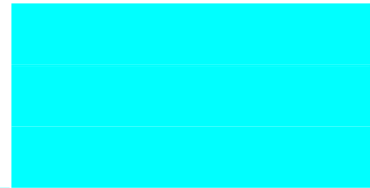
Parameter: D1 = Eingangspinnummer: ([siehe Erklärung](#))
D2 = Vergleichsbedingung: == (0) oder != (1)
D3 = Vergleichskonstante: 0 / 1
D4,D5 = Sprungadresse : 0...1499

4 -71 BIAS-Befehl "Wenn Ausgang X ? Konst. dann Springe"

Wenn Ausgang X ? Konst. dann Springe Y

Erklärung:

Der Befehl "Wenn Ausgang X ? Konst. dann Springe" erlaubt Programmverzweigungen durch den Vergleich des angewählten Ausgangs X mit der Konstanten 0 bzw. 1. Trifft der Vergleich zu wird die Abarbeitung an der angegebenen Adresse fortgeführt.



Reglertyp:	Parameter:	Zulässige Eingabe:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
631:	X:	5, 6 Pinnr. an X10-Klemme	
635/637/ 637+/637f:	X:	8, 12, 13, 20, 23 Pinnr. an X10- Buchse	
	X: Teilbestückte E/A-Optionsplatine RP-EA5 (Steckplatz B)	109, 110 Pinnr. an COM2 Sub-D-Buchse	
	X: Vollbestückte E/A-Optionsplatine RP-EAE (Steckplatz C)	209-210, 217-224 Pinnr. an X200 Sub-D-Buchse	
637f:	X: 2 x CAN + 4 Ausgänge und 4 Eingänge RP-2C8 (Steckplatz B):	125 – 128 Pinnr. an X120 Klemme	
	?	== gleich != ungleich	
	Konst.	0/1	
	Y: Adresse:	0...1499 Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.	

Beispiel:	Wenn Ausgang 12 == 1 dann	Ergebnis:
0		Ist der Ausgang 12 gesetzt, wird der Merker 1 gesetzt.
Springe 3		
1	Merker 1 = 0	
2	Springe 4	
3	Merker = 1	
4	Springe 4	

Ausgänge können unabhängig von ihrer Konfiguration abgefragt werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
71	X	?	Konst	Y(low)	Y(high)	00	00

Parameter: D1 = Ausgangspinnummer: ([siehe Erklärung](#))
D2 = Vergleichsbedingung: == (0) / != (1)
D3 = Vergleichskonstante 0 / 1
D4, D5 = Sprungadresse 0...1499

4 - 72 BIAS-Befehl "Ausgang X = Konst."

Ausgang X = Konst.

Erklärung:

Der Befehl "Ausgang X " weist dem Ausgang X die Konstante 0 oder 1 zu.

Reglertyp:	Parameter:	Zulässige Eingabe:	
631:	X:	5, 6 Pinnr. an X10-Klemme	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
635/637/ 637+/637f:	X:	12, 13, 20, 23 Pinnr. an X10- Buchse	
	X: Teilbestückte E/A-Optionsplatine RP-EA5 (Steckplatz B)	109, 110 Pinnr. an COM2 Sub-D-Buchse	
	X: Vollbestückte E/A-Optionsplatine RP-EAE (Steckplatz C)	209, 210, 217-224 Pinnr. an X200 Sub-D-Buchse	
637f:	X: 2 x CAN + 4 Ausgänge und 4 Eingänge RP-2C8 (Steckplatz B):	125– 128 Pinnr. an X120 Klemme	

Beispiel:

0 Wenn Eingang 2 == 1 dann Springe 3
 1 **Ausgang 12 = 0**
 2 Springe 4
 3 **Ausgang 12 = 1**
 4 Springe 4

Ergebnis:

Ist der Eingang 2 gesetzt, wird der Ausgang 12 gesetzt.
 (Ausgang 12 muss als BIAS-Ausgang konfiguriert sein.)

Hinweis:

- Der Befehl ist nur mit den Ausgängen gültig, die in der E/A-Konfiguration als BIAS-Ausgänge (Funktion 2) definiert sind.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
72	X	Konst.	00	00	00	00	00

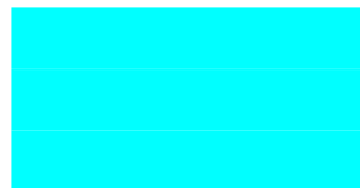
Parameter: D1 = Ausgangspinnennummer (siehe [Erklärung](#))
 D2 = Konstante: 0/1

4 - 73 BIAS-Befehl "Ausgang X = Merker Y"

Ausgang X = Merker Y

Erklärung:

Der Befehl "Ausgang X = Merker Y" weist dem Ausgang X den Zustand des Merkers Y zu.



Reglertyp:	Parameter:	Zulässige Eingabe	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	Y:	0...255	
631:	X:	5, 6 Pinnr. an X10-Klemme	
635/637/ 637+/637f:		12, 13, 20, 23 Pinnr. an X10- Buchse	
	X: Teilbestückte E/A-Optionsplatine RP-EA5 (Steckplatz B)	109, 110 Pinnr. an COM2 Sub-D- Buchse	
	X: Vollbestückte E/A-Optionsplatine RP-EAE (Steckplatz C)	209, 210, 217-227 Pinnr. an X200 Sub-D- Buchse	
637f:	X: 2 x CAN + 4 Ausgänge und 4 Eingänge RP-2C8 (Steckplatz B):	121 – 124 Pinnr. an X120 Klemme	

Beispiel:

0 Merker 0 = Eingang 2
 1 **Ausgang 12 = Merker 0**

Ergebnis:

Der Eingang 2 wird auf den Ausgang 12 gespiegelt.

Hinweis:

- Der Befehl ist nur mit den Ausgängen gültig, die in der Konfiguration als BIAS- Ausgänge (Funktion 2) definiert sind.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
73	X	Y	00	00	00	00	00

Parameter:

D1 = Ausgangspinnnummer: (siehe [Erklärung](#))
 D2 = Merker Nummer: 0...255

4 - 78 BIAS-Befehl "IBT-Maskennummer = X, LED Status [Variable Y]"

Erklärung:

Der Befehl "IBT-Maskennummer = X, LED Status [Variable Y]" aktiviert die ausgewählte Bildschirmmaske des IBT (intelligentes Bedienerterminal) und lädt den Zustand der Funktionstasten LED's.

Parameter:	Bezeichnung:	Zulässige Eingabe	
X:	Maskennummer:	0...9999	<input checked="" type="checkbox"/> BIAS-Programm
Y:		0..255	<input checked="" type="checkbox"/> SPS-Programm
			<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 Variable 0 = 0

1 **IBT-Maskennummer = 4, LED- Status = [Variable 0]**

Ergebnis:

Die Hauptmaske auf dem Display wird aktiviert. Alle LED's werden ausgeschaltet.

Hinweis:

- Der Inhalt der Variable Y für den LED Status muss aus der Summe der folgenden Werte gebildet werden.

Taste	ein	blinkend
F1	32768	16384
F2	8192	4096
F3	2048	1024
F4	512	256
F5	128	64
F6	32	16

Befehlskodierung:

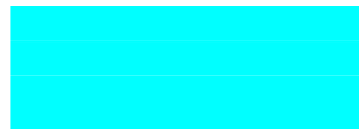
Code	D1	D2	D3	D4	D5	D6	D7
78	X(low)	X(high)	Y	00	00	00	00

Parameter: D1,D2 = Nummer der IBT Maske
D3 = Variablennummer: 0...255

4 - 79 BIAS-Befehl "IBT-Meldungsnummer =X, LED Status[VariableY]"

Erklärung:

Der Befehl "IBT-Meldungsnummer = X, LED Status [Variable Y]" aktiviert die Registrierung der angegebenen Meldung im seriellen Meldungspuffer des IBT und lädt den Zustand der Funktionstasten LED's



Parameter:	Bezeichnung:	Zulässige Eingabe:	
X:	Meldungsnummer	0...9999	<input checked="" type="checkbox"/> BIAS-Programm
Z:	Datensatznummer	0..255	<input checked="" type="checkbox"/> SPS-Programm
			<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 Variable 0 = 48

1 **IBT-Meldungsnummer = 10, LED-Status = [Variable 0]**

Ergebnis:

Die Meldungsmaske 10 wird auf dem IBT-Display aktiviert. Die LED der Funktionstasten 6 blinkt. Alle anderen LED's werden ausgeschaltet.

Hinweis:

- Die angegebene Meldung muss dem IBT bekannt sein.
- Der Inhalt der Variable Y für den LED Status muss aus der Summe der folgenden Werte gebildet werden.

Taste	ein	blinkend
F1	32768	16384
F2	8192	4096
F3	2048	1024
F4	512	256
F5	128	64
F6	32	16

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
79	X(low)	X(high)	Y	00	00	00	00

Parameter: D1,D2 = Nummer der IBT Meldung
D3 = Variablennummer: 0...255

4 - 7A BIAS-Befehl "CAN-Kommando = [Variable X]"

CAN-Kommando; Start=[Variable X]

Erklärung:

Der Befehl "CAN-Kommando = [Variable X]" kann ein Telegramm über den CAN-BUS senden bzw. empfangen.

- a. **CAN-Kommando senden**
 Der Variablenbereich beginnt mit der angegebenen Variable und ist 5 Variablen lang.
 Die erste Variable gibt den Identifier an.
 Mit den Bits 11 – 14 kann die Länge eingestellt werden.
 Mit dem Bit 14 kann zwischen fest definierter und variabler Datenlänge gewählt werden.
 Mit den Bits 11 - 13 wird die variable Länge binärkodiert.
 (siehe Bitkodierung des Identifier X)
 Die zweite bis fünfte Variable beinhalten die zu sendenden Daten.
- b. CAN-Kommando empfangen
 Der Variablenbereich beginnt mit der angegebenen Variable und ist 5 Variablen lang.
 Die erste Variable gibt den Identifier mit dem gesetzten Remote Bit (Bit 15 der angegebenen Variable) für das Empfangen von Daten.
 Die zweite Variable gibt die Startvariable an in denen die Empfangsdaten gespeichert werden. Insgesamt belegt der Empfangsdatenbereich 14 Variablen.
 (siehe Variablebereich und Datentyp)

a.) CAN-Kommando senden

Parameter:	Bezeichnung:	Zulässige Eingabe	
X	Modus	216	<input checked="" type="checkbox"/> BIAS-Programm
X + 1	Datenbyte 0, 1	216	<input checked="" type="checkbox"/> SPS-Programm
X + 2	Datenbyte 2 ,3	216	<input type="checkbox"/> Mathematik-Programm
X + 3	Datenbyte 4 ,5	216	
X + 4	Datenbyte 6 ,7	216	

Bitkodierung des Modus X

Bit 15	Bit 14		Bit 13	Bit 12	Bit 11	Bit 10 – 0
0 = Senden	0 = feste Länge	8 Byte	22	21	20	Identifier 0...2047
	1 = variable Länge	0 bis 7 Byte	Länge bitkodiert 0...7			
1 = Remote-anforderung	-					Identifier 0...2047

Beispiel: Sende Login

```

0   [Var. 0] = 200
1   [Var. 1] = 1
3   [Var. 2] = 0
4   [Var. 3] = 0
5   [Var. 4] = 0
6   CAN-Kommando; Start = [Variable 0]
7   ...

```

Ergebnis:

Mit dem Abarbeiten der Zeile 6 wird der Identifier 200 mit dem Inhalt 1 im Datenbyte 0 über die CAN-Busschnittstelle gesendet.

b.) CAN-Kommando Empfangen

Parameter:	Bezeichnung:	Zulässige Eingabe
X	Identifizier + Remote Bit (Bit 15)	0 ... 2047 + 32768
X + 1	Variablenstartadresse für die Empfangsdaten	0 ... 242
X + 2	reserviert	0
X + 3	reserviert	0
X + 4	reserviert	0

Die Variable X+1 gibt die Nummer der Variablen an, ab der die Empfangsdaten eingetragen werden.
Die 1. Variable der Empfangsdaten ist solange -1 bis die Antwort von der Gegenstation empfangen wurde.

Dabei werden folgende Variablen mit den entsprechenden Datenformaten verwendet:

CAN	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7
Zielvar. char	(X+1)+0	(X+1)+1	(X+1)+2	(X+1)+3	(X+1)+4	(X+1)+5	(X+1)+6	(X+1)+7
Zielvar. word	(X+1)+8		(X+1)+9		(X+1)+10		(X+1)+11	
Zielvar. longword	(X+1)+12				(X+1)+13			

Beispiel: Receive Status

```

1 [Variable 0 ] = 201
2 [Variable 0 ].Bit 15 = 1, [Merker 0 ]
3 [Variable 1 ] = 10
4 [Variable 2 ] = 0
5 [Variable 3 ] = 0
6 [Variable 4 ] = 0
7 CAN-Kommando; Start=[Variable 0 ]
WAIT_CAN:
8 Wenn [Variable 10 ] == -1 dann Springe WAIT_CAN
9 ...

```

Ergebnis:

Mit dem Abarbeiten der Zeile 7 wird der Identifizier 201 mit Remote-Kennung über die CAN-Busschnittstelle abgesandt. Die Gegenstation mit dem Senden-Identifizier 201 antwortet.
Wenn die Variable 10 ungleich dem Wert -1 wird, sind die empfangenen Daten beginnend ab Variable 10 abgespeichert.

CAN	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7
char	Var. 10	Var. 11	Var. 12	Var. 13	Var. 14	Var. 15	Var. 16	Var. 17
word	Var. 18		Var. 19		Var. 20		Var. 21	
longword	Var. 22				Var. 23			

Hinweis:

- Weitere Informationen zur Kodierung der CAN-Kommandos und dem Inhalt der Datenbytes erhalten Sie in der CAN-BUS Dokumentation. (UL: 07-05-03-02... und UL: 07-05-03-03...)
- Die Kodierung der Datenbyteanzahl ist ab der Firmware 8.19 möglich.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
7A	X	00	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...251

4 - 7B BIAS-Befehl "IBT-Datentransfer"

IBT-Datentransfer, Modus = X , Rezept = Y , Datensatz = Z

Erklärung:

Der Befehl "IBT- Datentransfer" aktiviert den Datentransfer vom bzw. zum IBT.

Modus 0: Transfer vom IBT zum Regler

Modus 1: Transfer vom Regler zum IBT

Parameter:	Bezeichnung:	Zulässige Eingabe	
X:	Modus:	0/1	<input checked="" type="checkbox"/> BIAS-Programm
Y:	Rezeptnummer	1..250	<input checked="" type="checkbox"/> SPS-Programm
Z:	Datensatznummer	0..250	<input type="checkbox"/> Mathematik-Programm

Beispiel:

Ergebnis:

Hinweis:

- Die angegebene Rezept- und Datensatznummer muss dem IBT bekannt sein.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
7B	X	Y	Z	00	00	00	00

Parameter:

D1 = Modus: 0/1
 D2 = Rezeptnummer: 0...250
 D3 = Datensatz: 0...250

4 - 7C BIAS-Befehl "CAN2-Kommando = [Variable X]"

CAN2-Kommando; Start=[Variable X]

Erklärung:

Der Befehl "CAN2-Kommando = [Variable X]" ermöglicht es, ein Telegramm über den CAN2-BUS zu senden bzw. empfangen.

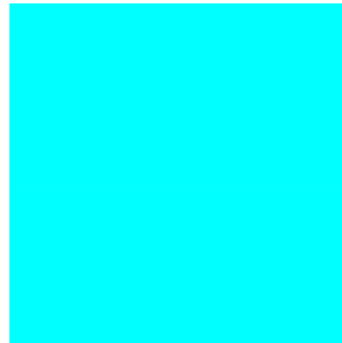
Es können bis zu 8 Byte über die Objekte 5 und 6 gesendet werden.

Bei den Empfängern werden die Empfangsdaten auf die Variable 246 und 247 (Objekt 5) bzw. 248 und 249 (Objekt 6) kopiert.

Entsprechend der in der Konfiguration kodierten Länge werden die nicht verwendeten Datenbytes beim Sender unterdrückt.

Vor dem Umparametrieren eines Senders muss der Modus "Aus" eingeschaltet werden.

Parameter: X	Bezeichnung: Variablennummer:	Zulässige Eingabe: 0...251
	Parameter X:	
Zulässiger Variableninhalt		
X:	Konfiguration:	
X+1:	Datenbyte 0 - 3	232
X+2:	Datenbyte 4 - 7	232



- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Konfiguration bitkodiert								
Bit 31 -16	Bit 15 – 12	Bit 11 - 8				Bit 7	Bit 6 – 0	
Verzögerungszeit	Datenbyte-anzahl	Modus				Ziel	Identifizier-offset	
Zeit bis nächstes CAN2-Kommando im Modus "Zyklisch senden" übertragen wird. 1 = 0,844ms 65536 = 55,3s	0...8	27	26	25	24	Aus Empfangen 1 mal senden Zyklisch senden	0 = Objekt 5 ab ID 300h 1 = Objekt 6 ab ID 380h	0 ... 7F
		0	0	0	0			
		0	0	0	1			
		0	0	1	0			
		0	1	1	0			

Beispiel 1: CAN2-Sender Objekt 5 mit 8 Byte

- 0 [Variable 0] = 0x4A0
- 1 [Variable 1] = 0x8
- 2 [Variable 2] = 0x6
- 3 [Variable 3] = 0x7F
- 4 [Variable 0] = [Variable 0] << 0x4
- 5 [Variable 0] = [Variable 0] + [Variable 1]
- 6 [Variable 0] = [Variable 0] << 0x4
- 7 [Variable 0] = [Variable 0] + [Variable 2]
- 8 [Variable 0] = [Variable 0] << 0x8
- 9 [Variable 20] = [Variable 0] + [Variable 3]
- 10 [Variable 21] = 0x150
- 11 [Variable 22] = 0x250
- 12 **CAN2-Kommando; Start=[Variable 20]**

Ergebnis:

Die Variable 0 wird mit dem Wert für die Verzögerungszeit von 1sec, die Variable 1 mit der Datenlänge von 8 Byte, die Variable 2 mit dem Modus 6 und die Variable 3 mit den Identifier Offset von 7Fh vorgeladen. Die folgenden 6 Befehle fassen die Konfiguration in Variable 20 zusammen. Der Variableninhalt der Variablen 20 ist: 0x04A0867F. Das Objekt 5 wird als Sender mit dem Identifier 37Fh initialisiert. Anschließend wird der Inhalt der Variablen 21 und 22 zyklisch jede Sekunde 1mal über den CAN2-Bus gesendet.

CAN2-Empfänger Objekt 5

5 [Variable 20] = 0x17F
 6 **CAN2-Kommando; Start=[Variable 20]**

Ergebnis:

Das Objekt 5 mit der ID 37Fh wird als Empfänger eingeschaltet.

Sender	Var. 21	Var 22
	Byte 0- 3	Byte 4-7
	0x150	0x250



Var. 246	Var 247	Empfänger
Byte 0 – 3	Byte 7 - 4	
0x150	0x250	

Beispiel 2: CAN2-Sender Objekt 6 mit 6 Byte

3 [Variable 20] = 0x6280
 4 [Variable 21] = 0xFFEEDDCC
 5 [Variable 22] = 0x88664422
 6 **CAN2-Kommando; Start=[Variable 20]**

Ergebnis:

Es werden 6 Byte der Variablen 21 und 22 einmal über des Objekt 6 mit der ID 380h über den CAN2-Bus gesendet

CAN2-Empfänger Objekt 6

5 [Variable 20] = 0x180
 6 **CAN2-Kommando; Start=[Variable 20]**

Ergebnis:

Das Objekt 6 mit der ID 380h wird als Empfänger eingeschaltet.

Sender	Var. 21	Var 22
	Byte 0- 3	Byte 4-7
	0xFFEEDDCC C	0x8866442 2



Var. 247	Var 246	Empfänger
Byte 4 - 7	Byte 0 – 3	
0xYYYY4422	0xFFEEDDCC	

YY = Bytes nicht verändert

Beispiel 3: CAN2-Sender Objekt 5 deaktivieren

0 [Variable 0] = 0x0
 1 **CAN2-Kommando; Start=[Variable 20]**

Ergebnis:

Das Objekt 5 wird ausgeschaltet und kann jetzt neu konfiguriert werden.

Hinweis:

- Nur bei 637f mit CAN 2 ab Firmware 8.19 möglich!
- Beachten Sie die Buslast bei der Verwendung dieses Befehls in Kombination mit den fest definierten CAN2- Telegrammen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
7C	X	00	00	00	00	00	00

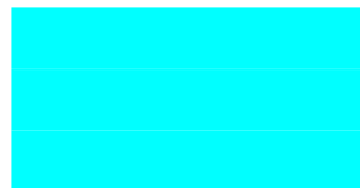
Parameter: D1 = Variablennummer: 0...251

4 - 80 BIAS-Befehl "[Variable X] = Konst."

[Variable X] = Konst.

Erklärung:

Der Befehl "[Variable X] = Konst." weist der Variablen X den Wert der Konstanten zu.



Reglertyp:	Parameter:	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm
Alle:	X:	0-255	<input checked="" type="checkbox"/> SPS-Programm
	Y: Konst.	± 231	<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 [Variable 0] = 100000

Ergebnis:

Die Variable 0 wird mit dem Wert 100000 geladen.

Hinweis:

1. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
2. Die Variableninhalte werden in der "EASYRIDER-Diagnose", "BIAS" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
80	X	Y(low)			Y(high)	00	00

Parameter: D1 = Variablennummer:0...255
D2-D5 = Konstante: ± 231

4 - 81 BIAS-Befehl "Wenn [Variable X] ? Konst."

Wenn [Variable X] ? dann Springe Y

Erklärung:

Der Befehl "Wenn [Variable X] ? Konst." erlaubt Programmverzweigungen durch den Vergleich der Variablen X mit der Konstanten.

Ist die angegebene Bedingung erfüllt, wird die BIAS-Abarbeitung an der angegebenen Adresse fortgeführt. Andernfalls wird der nächste BIAS-Befehl ausgeführt.

Reglertyp:	Parameter :	Zulässige Eingaben:	
Alle:	X:	0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	?:	> größer (0) < kleiner (1) == gleich (2) >= größer gleich (3) <= kleiner gleich (4) != ungleich (5) >+ positiv (6) <- negativ (7)	
	Y: Adresse:	0...1499 Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.	
		Wert in () für Befehlskodierung s.u.	

Beispiel:

```

0 Starte Achse
1 Fahre Position
2 Ausgang 20 = 0
3 [Variable 0] = Drehzahl 1
4 Wenn [Variable 0] < 500 dann Springe 3
5 Ausgang 20 = 1
  
```

LOOP:

```

6 [Variable 0] = Drehzahl 1
7 Wenn [Variable 0] < 500 dann Springe 2
8 Wenn [Variable 0] < 2500 dann Springe LOOP
9 Springe 2
  
```

Ergebnis:

Der Ausgang 20 wird gesetzt, wenn sich die Istgeschwindigkeit im Bereich von 500 bis 2500 rpm befindet.

Hinweis:

1. Die maximal zulässige Adresse ist abhängig von der Länge des BIAS-Programms und entspricht der Satznummer des letzten BIAS-Satzes.
2. Alternativ zu den Satznummern können Label (Sprungmarken) verwendet werden.
3. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
4. Die Variableninhalte werden in der "Diagnose", "BIAS-Diagnoseseite" zyklisch angezeigt.
5. Der Vergleich beider Werte wird durch die Subtraktion der Konstanten vom Variableninhalt ausgeführt.
Der bedingte Sprung ist vom Vorzeichen des Ergebnis abhängig.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6.7..D6.4	D6.3..D6.0	D7	
81	X	Y(low)			Y(high)		?	Z	Z

Parameter:
 D1 = Variablennummer: 0...255
 D2– D5= Konstante: $\pm 2^{31}$
 D6.7...D6.4: Vergleichbedingungen: 0...7 ([siehe Erklärung](#))
 D6.3..D6.0, D7: Sprungadresse: 0...1499

4 - 82 BIAS-Befehl "[Variable X] = [Variable Y] + Konst."

$$[\text{Variable X}] = [\text{Variable Y}] + \text{Konst.}$$

Erklärung:

Der Befehl "[Variable X] = [Variable Y] + Konst." addiert die Konstante zu dem Inhalt der Variablen Y und speichert das Ergebnis in der Variablen X ab.

Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255	
	Y:	0...255	
	Konst.:	± 231	

Beispiel:

```

0 Wenn Eingang 11 == 0 dann Springe 4
1 [Variable 0] = Weg
2 [Variable 0] = [Variable 0] + 15000
3 Weg = [Variable 0]
4 Starte Achse
5 Fahre Position
  
```

Ergebnis:

Wenn der Eingang 11 aktiv ist, fährt die Achse 15000 Inkremente weiter.

Hinweis:

1. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
2. Die Variableninhalte werden in der "Diagnose", "BIAS- Diagnosesseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
82	X	Y	Z(low)			Z(high)	00

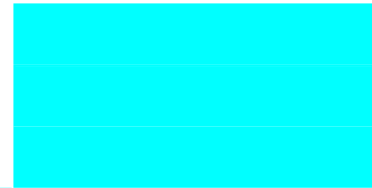
Parameter: D1 = Variablennummer 0...255
 D2 = Variablennummer 0...255
 D3 – D6 = Konstante ± 231

4 - 83 BIAS-Befehl "[Variable X] = [Variable Y] - Konst."

$$[\text{Variable X}] = [\text{Variable Y}] - \text{Konst.}$$

Erklärung:

Der Befehl "[Variable X] = [Variable Y] - Konst." subtrahiert die Konstante vom Inhalt der Variablen Y und speichert das Ergebnis in der Variablen X ab.



Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255	
	Y:	0...255	
	Konst.:	± 231	

Beispiel:

- 0 Wenn Eingang 11 == 0 dann Springe 4
- 1 [Variable 0] = Weg
- 2 **[Variable 0] = [Variable 0] - 15000**
- 3 Weg = [Variable 0]
- 4 Starte Achse
- 5 Fahre Position

Ergebnis:

Wenn der Eingang 11 aktiv ist, fährt die Achse 15000 Inkremente weniger.

Hinweis:

3. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
4. Die Variableninhalte werden in der "Diagnose", "BIAS- Diagnosesseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
83	X	Y	Z(low)			Z(high)	00

Parameter: D1 = Variablennummer 0...255
D2 = Variablennummer 0...255
D3 – D6 = Konstante ± 231

4 - 84 BIAS-Befehl "[Variable X] = [Variable Y] * Konst."

$$[\text{Variable X}] = [\text{Variable Y}] * \text{Konst.}$$

Erklärung:

Der Befehl "[Variable X] = [Variable Y] * Konst." multipliziert den Inhalt der Variablen Y mit der Konstante und speichert das Ergebnis in der Variablen X ab.

Reglertyp:	Parameter	Zulässige Eingaben:	
	:		<input checked="" type="checkbox"/> BIAS-Programm
			<input checked="" type="checkbox"/> SPS-Programm
			<input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255	
	Y:	0...255	
	Konst.:	± 231	

Beispiel:

0 Wenn Eingang 11 == 0 dann Springe 4
 1 [Variable 0] = Weg
 2 **[Variable 0] = [Variable 0] * 5**
 3 Weg = [Variable 0]
 4 Starte Achse
 5 Fahre Position

Ergebnis:

Wenn der Eingang 11 aktiv ist, fährt die Achse den 5-fachen Weg.

Hinweis:

- Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
- Die Variableninhalte werden in der "Diagnose", "BIAS- Diagnosesseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
84	X	Y	Z(low)			Z(high)	00

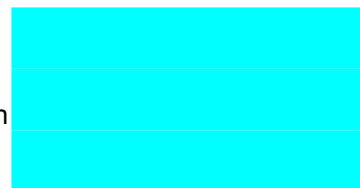
Parameter: D1 = Variablennummer 0...255
 D2 = Variablennummer 0...255
 D3 – D6 = Konstante ± 231

4 - 85 BIAS-Befehl "[Variable X] = [Variable Y] / Konst."

$$[\text{Variable X}] = [\text{Variable Y}] / \text{Konst.}$$

Erklärung:

Der Befehl "[Variable X] = [Variable Y] / Konst." dividiert den Inhalt der Variablen Y durch die Konstante und speichert das Ergebnis in der Variablen X.



Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm
Alle:	X:	0...255	<input checked="" type="checkbox"/> SPS-Programm
	Y:	0...255	<input checked="" type="checkbox"/> Mathematik-Programm
	Konst.:	± 231	

Beispiel:

- 0 Wenn Eingang 11 == 0 dann Springe 4
- 1 [Variable 0] = Weg
- 2 **[Variable 0] = [Variable 0] / 5**
- 3 Weg = [Variable 0]
- 4 Starte Achse
- 5 Fahre Position

Ergebnis:

Wenn der Eingang 11 aktiv ist, fährt die Achse 1/5 des Weges.

Hinweis:

7. Die Division liefert ein ganzzahliges Ergebnis.
8. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
9. Die Variableninhalte werden in der "Diagnose", "BIAS- Diagnoseseseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
85	X	Y	Z(low)			Z(high)	00

Parameter: D1 = Variablennummer 0...255
D2 = Variablennummer 0...255
D3 – D6 = Konstante ± 231

4 - 86 BIAS-Befehl "[Variable X] = Merker Y"

[Variable X] = Merker Y , Anzahl = Z

Erklärung:

Der Befehl [Variable X] = Merker, Anzahl = Z." weist der Variablen X auf den Bitpositionen 0 bis (Anzahl-1) die Wertigkeit von Merker Y bis Y+(Anzahl-1) zu.

Reglertyp:	Parameter	Zulässige Eingaben:	
	:		<input checked="" type="checkbox"/> BIAS-Programm
			<input checked="" type="checkbox"/> SPS-Programm
			<input type="checkbox"/> Mathematik-Programm
Alle:	X:	0 - 255	
	Y:	0 – max. 255 -(Anzahl W-1)	
	Z:	1 - 32	

Beispiel:

0 [Merker 0] = 0
 1 [Merker 1] = 1
 2 [Merker 2] = 0
 3 [Merker 3] = 1
 4 [Merker 4] = 0
 5 [Merker 5] = 1
 6 [Merker 6] = 0
 7 [Merker 7] = 1
 8 [Variable 8] = [Merker 0] , Anzahl = 8

Ergebnis:

Die Variable 8 wird auf den Bitpositionen 0 -7 der Reihe nach mit den der Wertigkeit von Merker 0 – 7 geladen.

Variable 8 = 204d ; 0CCh, ..0010101010b

Hinweis:

1. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
2. Die Variableninhalte werden in der "EASYRIDER-Diagnose", "BIAS" zyklisch angezeigt.
3. Die Zielvariable X wird vor der Zuweisung gelöscht, d.h. die nicht verwendeten Bitwerte sind 0.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
86	X	Y	Z	00	00	00	00

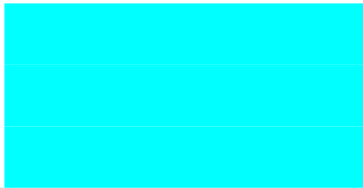
Parameter: D1 = Variablennummer:0...255
 D2 = Merker Nummer: 0...255
 D3 = Anzahl: 1...32

4 -87 BIAS-Befehl “[Variable X] = [Variable Y].Bit Z ; Anzahl“

[Variable X] = [Variable Y].Bit Z , Anzahl = W

Erklärung:

Der Befehl [Variable X] = [Variable Y].Bit Z, Anzahl = W." weist der Variablen X auf den Bitpositionen 0 bis (Anzahl-1) die Wertigkeit von Bit Z der Variablen Y bis Y+(Anzahl-1) zu .



Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
637f:	X:	0 - 255	
	Y:	0 – max. 255 -(Anzahl W-1)	
	Z:	0 - 31	
	W:	1 - 32	

Beispiel:

- 0 [Variable 0] = 1
- 1 [Variable 1] = 3
- 2 [Variable 2] = 1
- 3 [Variable 3] = 3
- 4 [Variable 4] = 1
- 5 [Variable 5] = 3
- 6 [Variable 6] = 1
- 7 [Variable 7] = 3
- 8 [Variable 8] = [Variable 0].Bit 0 , Anzahl = 8
- 9 [Variable 9] = [Variable 0].Bit 1 , Anzahl = 8

Ergebnis:

Die Variable 8 wird auf den Bitpositionen 0 -7 der Reihe nach mit den der Wertigkeit von Bit 0 der Variablen 0 – 7 geladen.

Variable 8 = 255d ; 0FFh, ...0011111111b
 Variable 9 = 204d ; 0CCh, ..0010101010b

Hinweis:

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung.
- Dieser Befehl steht beim 631, 635, 637 ab der Firmware V6.19 zur Verfügung.
- Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
- Die Variableninhalte werden in der "EASYRIDER-Diagnose", "BIAS" zyklisch angezeigt.
- Die Zielvariable X wird vor der Zuweisung gelöscht, d.h. die nicht verwendeten Bitwerte sind 0.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
87	X	Y	Z	W	00	00	00

Parameter: D1 = Variablennummer:0...255
 D2 = Variablennummer:0...255
 D3 = Bitnummer: 0...31
 D4 = Anzahl: 1...32

4 - 88 BIAS-Befehl “[Variable X] = [Variable Y]“

[Variable X] = [Variable Y]

Erklärung:

Der Befehl “[Variable X] = [Variable Y]“ weist der Variablen X den Inhalt der Variablen Y zu.

Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255	
	Y:	0...255	

Beispiel:

0 [Variable 0] = Weg
 1 [Variable 1] = [Variable 0]
 2 [Variable 0] = [Variable 0] / 5
 3 [Variable 0] = [Variable 0] + [Variable 1]
 4 Weg = [Variable 0]

Ergebnis:

Der Parameter Weg wird mit 1,2 multipliziert.

Hinweis:

1. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
2. Die Variableninhalte werden in der "Diagnose", "BIAS-Diagnoseseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
88	X	Y	00	00	00	00	00

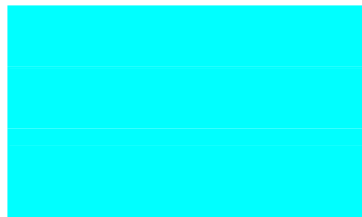
Parameter: D1 = Variablennummer:0...255
 D2 = Variablennummer:0...255

4 - 89 BIAS-Befehl "Wenn [Variable X] ? [Variable Y]"

Wenn [Variable X] ? [Variable Y] dann Springe Z

Erklärung:

Der Befehl "Wenn [Variable X] ? [Variable Y]" erlaubt Programmverzweigungen durch den Vergleich der Variablen X mit der Variablen Y. Ist die angegebene Bedingung erfüllt, wird die BIAS-Abarbeitung an der angegebenen Adresse fortgeführt. Andernfalls wird der nächste BIAS-Befehl ausgeführt.



Reglertyp:	Parameter	Zulässige Eingaben:	
	:		<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255	
	?:	> größer (0) < kleiner (1) == gleich (2) >= größer gleich (3) <= kleiner gleich (4) != ungleich (5) -> positiv (6) -< negativ (7)	
		Wert in () für Befehlskodierung s.u.	
	Y:	0...255	
	Z: Adresse:	0...1499 Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.	

Beispiel:

- 0 Starte Achse
- 1 Fahre Position
- 2 [Variable 1] = 500
- 3 [Variable 2] = 2500
- 4 Ausgang 20 = 0
- 5 [Variable 0] = Drehzahl 1
- 6 **Wenn [Variable 0] < [Variable 1] dann Springe 5**
- 7 Ausgang 20 = 1

LOOP:

- 8 [Variable 0] = Drehzahl 1
- 9 **Wenn [Variable 0] < [Variable 1] dann Springe 4**
- 10 **Wenn [Variable 0] < [Variable 2] dann Springe LOOP**
- 11 Springe 2

Ergebnis:

Der Ausgang 20 wird gesetzt, wenn sich die Istgeschwindigkeit im Bereich von 500 bis 2500 rpm befindet.

Hinweis:

3. Die maximal zulässige Adresse ist abhängig von der Länge des BIAS-Programms und entspricht der Satznummer des letzten BIAS-Satzes.
4. Alternativ zu den Satznummern können Label (Sprungmarken) verwendet werden.
5. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
6. Die Variableninhalte werden in der "Diagnose", "BIAS-Diagnoseseite" zyklisch angezeigt.

7. Der Vergleich beider Werte wird durch die Subtraktion der beiden Variableninhalt voneinander ausgeführt.
Der bedingte Sprung ist vom Vorzeichen des Ergebnis abhängig.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
89	X	?	Y	Z(low)	Z(high)	00	00

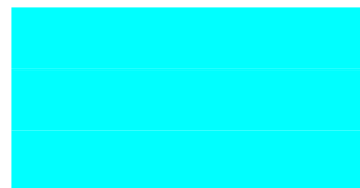
Parameter: D1 = Variablennummer: 0...255
 D2 = Vergleichsbedingung 0...7 (siehe [Erklärung](#))
 D3 = Variablennummer: 0...255
 D4,D5 = Sprungadresse 0-1499

4 - 8A BIAS-Befehl “[Variable X] = [Variable Y] + [Variable Z]“

$$[\text{Variable X}] = [\text{Variable Y}] + [\text{Variable Z}]$$

Erklärung:

Der Befehl “[Variable X] = [Variable Y] + [Variable Z]“ addiert den Inhalt der Variablen Y und zum Inhalt der Variablen Z und speichert das Ergebnis in Variable X.



Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255	
	Y:	0...255	
	Z:	0...255	

Beispiel:

0 Wenn Eingang 11 == 0 Springe 5
 1 [Variable 1] = 15000
 2 [Variable 2] = Weg
 3 **[Variable 0] = [Variable 0] + [Variable 1]**
 4 Weg = [Variable 0]
 5 Starte Achse
 7 Fahre Position

Ergebnis:

Wenn der Eingang 11 aktiv ist, fährt die Achse 15000 Inkremente weiter.

Hinweis:

- Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
- Die Variableninhalte werden in der "Diagnose", "BIAS-Diagnoseseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
8A	X	Y	Z	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
 D2 = Variablennummer: 0...255
 D3 = Variablennummer: 0...255

4 - 8B BIAS-Befehl “[Variable X] = [Variable Y] - [Variable Z]”

$$[\text{Variable X}] = [\text{Variable Y}] - [\text{Variable Z}]$$

Erklärung:

Der Befehl “[Variable X] = [Variable Y] - [Variable Z]” subtrahiert den Inhalt der Variablen Z von dem Inhalt der Variablen Y und speichert das Ergebnis in Variable X.

Reglertyp:	Parameter:	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255	
	Y:	0...255	
	Z:	0...255	

Beispiel:

0 Wenn Eingang 11 == 0 Sringe 5
 1 [Variable 1] = 15000
 2 [Variable 2] = Weg
 3 **[Variable 0] = [Variable 0] - [Variable 1]**
 4 Weg = [Variable 0]
 5 Starte Achse
 7 Fahre Position

Ergebnis:

Wenn der Eingang 11 aktiv ist, fährt die Achse 15000 Inkremente kürzer.

Hinweis:

- Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
- Die Variableninhalte werden in der "Diagnose", "BIAS-Diagnoseseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
8B	X	Y	Z	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
 D2 = Variablennummer: 0...255
 D3 = Variablennummer: 0...255

4 - 8C BIAS-Befehl “[Variable X] = [Variable Y] * [Variable Z]“

$$[\text{Variable X}] = [\text{Variable Y}] * [\text{Variable Z}]$$

Erklärung:

Der Befehl “[Variable X] = [Variable Y] * [Variable Z]“ multipliziert den Inhalt der Variablen Z mit dem Inhalt der Variablen Y und speichert das Ergebnis in Variable X.

Reglertyp:	Parameter:	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255	
	Y:	0...255	
	Z:	0...255	

Beispiel:

0 Wenn Eingang 11 == 0 Sringe 5
 1 [Variable 1] = 5
 2 [Variable 2] = Weg
 3 **[Variable 0] = [Variable 0] * [Variable 1]**
 4 Weg = [Variable 0]
 5 Starte Achse
 7 Fahre Position

Ergebnis:

Wenn der Eingang 11 aktiv ist, fährt die Achse den 5-fachen Weg..

Hinweis:

- 12. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
- 13. Die Variableninhalte werden in der "Diagnose", "BIAS-Diagnoseseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
8C	X	Y	Z	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
 D2 = Variablennummer: 0...255
 D3 = Variablennummer: 0...255

4 - 8D BIAS-Befehl “[Variable X] = [Variable Y] / [Variable Z]“

$$[\text{Variable X}] = [\text{Variable Y}] / [\text{Variable Z}]$$

Erklärung:

Der Befehl “[Variable X] = [Variable Y] / [Variable Z]“ dividiert den Inhalt der Variablen Y durch den Inhalt der Variablen Z und speichert das Ergebnis in Variable X.

Reglertyp:	Parameter:	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255	
	Y:	0...255	
	Z:	0...255	

Beispiel:

0 Wenn Eingang 11 == 0 Sringe 5
 1 [Variable 1] = 5
 2 [Variable 2] = Weg
 3 **[Variable 0] = [Variable 0] / [Variable 1]**
 4 Weg = [Variable 0]
 5 Starte Achse
 7 Fahre Position

Ergebnis:

Wenn der Eingang 11 aktiv ist, fährt die Achse 1/5 des Weges.

Hinweis:

- Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
- Die Variableninhalte werden in der "Diagnose", "BIAS-Diagnoseseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
8D	X	Y	Z	00	00	00	00

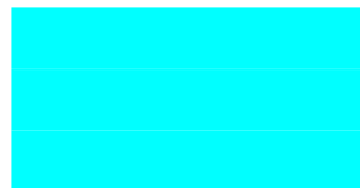
Parameter: D1 = Variablennummer: 0...255
 D2 = Variablennummer: 0...255
 D3 = Variablennummer: 0...255

4 - 8E BIAS-Befehl “[Teachvariable X] = [Variable Y]“

[Teachvariable X] = [Variable Y]

Erklärung:

Der Befehl “[Teachvariable X] = [Variable Y]“ weist der netzausfallsicheren Variablen X (Teachvariable) den Inhalt der Variablen Y zu.



Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
Alle:	X:	0...15	
	Y:	0...255	

Beispiel:

0 [Variable 0] = Istposition 1
1 [Teachvariable 1] = [Variable 0]

Ergebnis:

Die aktuelle Istposition 1 wird in der Teachvariable 1 netzausfallsicher gespeichert. Sie kann zum Beispiel als spätere Zielposition benutzt werden.

Hinweis:

- Während der Ausführung dieses Befehls ist die BIAS-Abarbeitung für ca. 10ms unterbrochen.
- Beachten Sie die maximalen Speicherzyklen des EEPROM beim zyklischen Schreiben diesen Befehls
(ca. 100.000 Speicherzyklen)

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
8E	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...15
D2 = Variablennummer: 0...255

4 - 8F BIAS-Befehl “[Variable X] = [Teachvariable Y]“

[Variable X] = [Teachvariable Y]

Erklärung:

Der Befehl “[Variable X] = [Teachvariable Y]“ weist der Variablen X den Inhalt der netzausfallsicheren Variablen Y (Teachvariable) zu.

Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm
			<input checked="" type="checkbox"/> SPS-Programm
			<input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255	
	Y:	0...15	

Beispiel:

0 [Variable 0] = [Teachvariable 1]
1 Weg = [Variable 0]

Ergebnis:

Der Inhalt der Teachvariablen 1 wird als Sollposition geladen.

Hinweis:

16. Nach dem Einschalten des Reglers werden die Inhalte aller Teachvariablen automatisch vorgeladen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
8F	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer: 0...255
D2 = Variablennummer: 0...15

4 - 90 BIAS-Befehl "Mathematikprogramm"

Erklärung:

Der Befehl "Mathematikprogramm" startet die Abarbeitung einer dritten Task. Diese wird in den nicht von Interrupts belegten Zeiten abgearbeitet, d.h. die Abarbeitung ist zeitlich nicht determiniert.

Reglertyp:	Parameter :	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm
Alle:	Adresse:	0 - 1499	
		Die maximal zulässige Sprungadresse entspricht Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden	

Beispiel:

0 Programmende, Modus = 3
1 **Mathematikprogramm Mathe**

Ergebnis:

Das laufende Mathematikprogramm wird beendet und das Programm ab dem Sprunglabel Mathe gestartet.

Hinweis:

- Führen Sie den Befehl "Mathematik-Programm" nicht zyklisch aus, da ansonsten die Mathematik-Abarbeitung ständig an der angegebenen Satznummer beginnt.
- Innerhalb eines Mathematik - Programms sind nicht alle BIAS- Befehle erlaubt.
- In der BIAS- Programmdefinition (Parameter „Programmreaktion Math- Programm“) kann die Reaktion des Mathematik - Programms bei der Deaktivierung der Endstufe eingestellt werden. Damit ist es möglich das Mathematik - Programm auch während der Deaktivierung des Reglers weiterarbeiten zu lassen.
- Wird in diesem Modus der Befehl " Mathematik -Programm ..." auf der **1. Zeile** bzw. **2. Zeile** (falls SPS- Programm auf 1. Zeile) der BIAS- Programmabarbeitung programmiert, so läuft das Mathematik - Programm auch ohne Aktivierung des Reglers an.
- Achtung bei beim Modus "bei Deaktivierung weiterarbeiten"!!**
Beim Senden eines neuen BIAS- Programms mit Änderung in der Zeilennummern (Verschiebung von Programmteilen) können unkontrollierte Maschinenbewegungen die Folge sein!

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
90	X(low)	X(high)	00	00	00	00	00

Parameter: D1, D2 : Sprungadresse 0...1499

4 -91 BIAS-Befehl "Profilinitialisierung"

Profilinit.; Profilnr.= X ,Stützstellen = Y ,Startadresse = Z

Erklärung:

Der Befehl "Profilinitialisierung" initialisiert den angegebenen Synchronprofilparameterblock X. Dieser Befehl sollte immer zusammen mit dem Befehl "Profiltaktlänge" benutzt werden.

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X: Profilnummer	0...15	<input type="checkbox"/> BIAS-Programm
	Y: Stützstellen	1...1024	<input type="checkbox"/> SPS-Programm
	Z: Startadresse	0...2047	<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

ProgStart:
0 Profilinit.; Profilnr.=0 ,Stützstellen=128 ,Startadresse=0

Ergebnis:

Das Profil 0 wird mir 128 Stützstellen ab der Startadresse 0 initialisiert.

Hinweis:

Die Werte können auch mit dem Befehl "Profilwert, Profil X , Wert Y = [Variable Z]" geladen werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
91	X	Y(low	Y(high	Z(low	Z(high)	00	00

Parameter: D1: Profilnummer: 0...15
D2, D3: Stützstellen: 1...1024
D4, D5: Startadresse: 0...2047

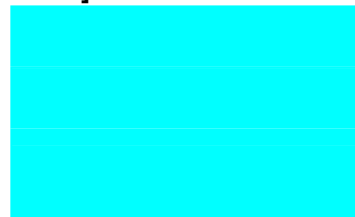
4 - 92 BIAS-Befehl "Profiltaktlänge = [Variable X]"

Profillänge; Profilnr.= X ,Master=[Var. Y] ,Slave=[Var. Z]

Erklärung:

Der Befehl "Profiltaktlänge" legt die Taktlängen für Master und Slave des angegebenen Synchronprofilparameterblocks X anhand des Inhalts der Variablen Y und Z fest.

Dieser Befehl sollte immer zusammen mit dem Befehl "Profilinitialisierung" benutzt werden.



Reglertyp:	Parameter:	Zulässige Eingaben:
Alle:	X: Profilnummer	0...15
	Y:	0...255
	Z:	0...255

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

ProgStart:

0 [Variable 10] = 100000

1 [Variable 11] = 80000

2 **Profillänge; Profilnr.=0 ,Master=[Var. 10] ,Slave=[Var. 11]**

Ergebnis:

Das Profil 0 wird mit Mastertaktlänge von 100000 Inkr. und Slavetaktlänge von 80000 Inkrementen initialisiert.

Hinweis:

Die Werte können auch mit dem Befehl "Profilwert, [Profil X , Wert Y = \[Variable Z \]](#)" geladen werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
92	X	Y	Z	00	00	00	00

Parameter: D1 = Profilnummer: 0...15
 D2 = Variablennummer:0...255
 D3 = Variablennummer:0...255

4 - 93 BIAS-Befehl "[Variable X] = Profilwert"

[Variable X] = Profilwert, Profil Y, Wert Z

Erklärung:

Der Befehl "[Variable X] = Profilwert, Profil Y, Wert Z" lädt den Wert des ausgewählten Parameter Z des Synchronprofilparameterblocks Y in die Variable X.

Reglertyp:	Parameter:	Zulässige Eingaben:
Alle:	X:	0...255
	Y: Profilnummer	0...15
	Z:	0 Stützstellenanzahl 1 Startadresse der 1. Stützstelle 2 Mastertaktlänge 3 Slavetaktlänge 4...99 Reserve

- BIAS-Programm
 SPS-Programm
 Mathematik-Programm

Beispiel:

ProgStart:

- 0 [Variable 0] = Profilwert, Profil 0 , Wert 0
- 1 [Variable 1] = Profilwert, Profil 0 , Wert 1
- 2 [Variable 2] = Profilwert, Profil 0 , Wert 2
- 3 [Variable 3] = Profilwert, Profil 0 , Wert 3

Ergebnis:

Die Parameter des Profils 0 werden der Reihe nach in die Variablen 0 – 3 geladen.

Hinweis:

Mit diesem Befehl ist es möglich importierte Profile auf gültige Wertebereiche zu überprüfen.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
93	X	Y	Z	00	00	00	00

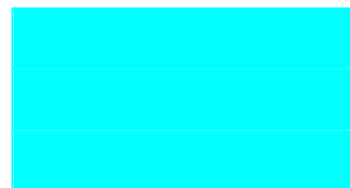
Parameter: D1 = Variablennummer:0...255
 D2 = Profilnummer: 0...15
 D3 = Wert: 0...99

4 - 94 BIAS-Befehl "Profilwert = [Variable X]"

Profilwert, Profil X , Wert Y = [Variable Z]

Erklärung:

Der Befehl "Profilwert, Profil X, Wert Y = [Variable Z]" schreibt den Wert der Variablen Z in den ausgewählten Wert Y des Synchronprofilparameterblocks X.



Reglertyp:	Parameter:	Zulässige Eingaben:
Alle:	X: Profilnummer	0...15
	Y:	0 Stützstellenanzahl 1 Startadresse der 1. Stützstelle 2 Mastertaktlänge 3 Slavetaktlänge 4...99 Reserve
	Z:	0...255

- BIAS-Programm
- SPS-Programm
- Mathematik-Programm

Beispiel:

ProgStart:
 0 [Variable 10] = 128
 1 **Profilwert, Profil 0 , Wert 0 = [Variable 10]**
 2 NOP

Ergebnis:

Das Profil 0 wird mit der Stützstellenanzahl 128 parametrieret.

Hinweis:

Änderungen von Profilparametern während das Profil abgearbeitet wird sind nicht erlaubt!

Befehlskodierung:

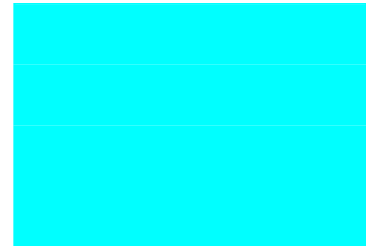
Code	D1	D2	D3	D4	D5	D6	D7
94	X	Y	Z	00	00	00	00

Parameter: D1 = Profilnummer: 0...15
 D2 = Wert 0...99
 D3 = Variablennummer 0...255

4 - 97 BIAS-Befehl "Tabelle speichern"

Erklärung:

Der Befehl "Tabelle Speichern" startet die Speicherung der gesamten Profiltabelle.
 Die Ausführung des SPS-Programms wird für die Zeit der Speicherung unterbrochen.
 Das heißt die Abarbeitungszeit dieses Befehls überschreitet den 2ms Standardwert.



Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	keine	--	<input type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input type="checkbox"/> Mathematik-Programm

Beispiel:

0 Wenn Merker 1 == 0 dann Springe NotSave
 1 Programmende 4
 2 **Tabelle Speichern**
 3 Programmende 5
 4 Merker 1 = 0
 NotSave:

Ergebnis:

Wenn Merker 1 gesetzt ist wird die Profiltabelle gespeichert. Dazu wird als erstes der Regler deaktiviert anschließend die Tabelle gespeichert. Als Abschluss wird der Regler wieder aktiviert.

Hinweis:

- Die Ausführung dieses Befehls ist nur im deaktivierten Zustand des Reglers möglich, d.h. als Abarbeitungsmodus des SPS-Programms muß der Wert 2 "Weiterabarbeitung bei Deaktivierung" konfiguriert sein.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
97	00	00	00	00	00	00	00

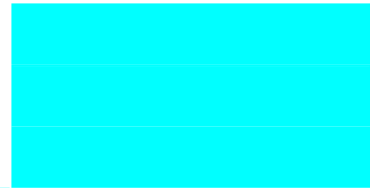
Parameter:

4 - A0 BIAS-Befehl "Tabelle ([Variable X])= Konst."

Tabelle ([Variable X]) = Konst.

Erklärung:

Der Befehl "Tabelle([Variable X]) = Konst." lädt die Konstante in die, durch die Variablen X adressierte, Profilspeicherstelle.



Reglertyp:	Parameter:	Zulässige Eingaben:	<input type="checkbox"/> BIAS-Programm
Alle:	X:	0-255 Zulässiger Variableninhalt 0...4095	<input type="checkbox"/> SPS-Programm
	Y: Konst.	± 231	<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

ProgStart:

0 [Variable 10] = 0

1 **Tabelle([Variable 10])= 1000**

2 NOP

Ergebnis:

Der Wert 1000 wird auf den Tabellenspeicherplatz 0 geladen.

Hinweis:

Mit dem Befehl "Tabelle [speichern](#)" können die Werte der Tabelle netzausfallsicher gespeichert werden.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
A0	X	Y(low			Y(high	00	00
)		

Parameter: D1 = Variablennummer:0...255
D2-D5 = Konstante : ± 231

4 - A1 BIAS-Befehl "Tabelle ([Variable X]) = [Y Variable Z]"

Tabelle ([Variable X]) = [Y Variable Z]

Erklärung:

Der Befehl "Tabelle ([Variable X]) = [Y Variable Z]" lädt den Inhalt der Typvariablen Z in die, durch die Variablen X adressierte, Profilspeicherstelle.

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X:	0-255 Zulässiger Variableninhalt: 0...4095	<input type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	Y: Typvorwahl	L = Long Integer F = Float D = Double Float	
	Z:	0...255	

Beispiel:

ProgStart:

```
0 [Variable 0 ] = 1000
1 [Variable 10 ] = 0
2 Tabelle([Variable 10 ])= [L Variable 0 ]
3 NOP
```

Ergebnis:

Der Wert 1000 wird auf den Tabellenspeicherplatz 0 geladen.

Hinweis:

1. Mit dem Befehl "Tabelle speichern" können die Werte der Tabelle netzausfallsicher gespeichert werden.
2. Typvariablen werden vor der Zuweisung in das Format Long Integer gewandelt.
3. Es stehen ihnen 3 verschiedene Datentypvarianten zur Verfügung. Alle Typvarianten sind in getrennten Speicherbereichen angelegt.
Long Integer Variablen 0...255 entsprechen den BIAS-Variablen.
Float Variablen 0...255 entsprechen dem 4-Byte IEEE Format.
Double Float Variablen 0...255 entsprechen dem 8-Byte IEEE Format.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
A1	X	Y	Z	00	00	00	00

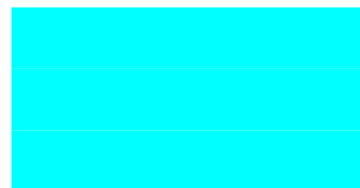
Parameter: D1 = Variablennummer:0...255
D2 = Datentypvorwahl: (Long = 0, Float = 1, Double = 2)
D3 = Variablennummer:0...255

4 - A2 BIAS-Befehl "[X Variable Y] = Tabelle ([Variable Z])"

[X Variable Y] = Tabelle ([Variable Z])

Erklärung:

Der Befehl "[X Variable Y] = Tabelle ([Variable Z])" lädt den Inhalt, der durch die Variablen Z adressierten Profilspeicherstelle, in die Typvariable X.



Reglertyp:	Parameter:	Zulässige Eingaben:	<input type="checkbox"/> BIAS-Programm
Alle:	X: Typvorwahl	L = Long Integer F = Float D = Double Float	<input type="checkbox"/> SPS-Programm
	Y:	0-255	<input checked="" type="checkbox"/> Mathematik-Programm
	Z:	0-255 Zulässiger Variableninhalt: 0...4095	

Beispiel:

```
ProgStart:
0 [Variable 10] = 0
1 [L Variable 5 ] = Tabelle([Variable 10 ])
2 NOP
```

Ergebnis:

Der Wert auf dem Tabellenspeicherplatz 0 wird in die Long Integer Variable 5 geladen.

Hinweis:

- Typvariablen werden vor der Zuweisung in das Format Long Integer gewandelt.
- Es stehen ihnen 3 verschiedene Datentypvarianten zur Verfügung. Alle Typvarianten sind in getrennten Speicherbereichen angelegt.
Long Integer Variablen 0...255 entsprechen den BIAS-Variablen.
Float Variablen 0...255 entsprechen dem 4-Byte IEEE Format.
Double Float Variablen 0...255 entsprechen dem 8-Byte IEEE Format.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
A2	X	Y	Z	00	00	00	00

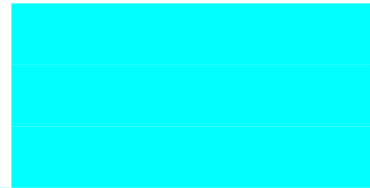
Parameter: D1 = Datentypvorwahl: (Long = 0, Float = 1, Double = 2)
D2 = Variablennummer:0...255
D3 = Variablennummer:0...255

4 - A3 BIAS-Befehl “[W Variable X] = [Y Variable Z]“

[W Variable X] = [Y Variable Z]

Erklärung:

Der Befehl "[W Variable X] = [Y Variable Z]" weist der Typvariablen X den Wert der Typvariablen Z zu.



Reglertyp:	Parameter:	Zulässige Eingaben:	<input type="checkbox"/> BIAS-Programm
Alle:	W: Typvorwahl	L = Long Integer F = Float D = Double Float	<input type="checkbox"/> SPS-Programm
	X:	0-255	<input checked="" type="checkbox"/> Mathematik-Programm
	Y: Typvorwahl	L = Long Integer F = Float D = Double Float	
	Z:	0-255	

Beispiel:

0 [D Variable 0] = 0
 1 [D Variable 1] = 0.250000000
 2 Wenn [D_Variable 1] -< [D_Variable 0] dann Springe Fehler
 3 [D_Variable 2] = SQRT([D_Variable 1])
 4 [L_Variable 1] = [D_Variable 2]
 5 ...
 Fehler:
 50 ...

Ergebnis:

Die Double Variable 2 wird in die Long Integer Variable 1 geladen.

Hinweis:

- 6. Typvariablen werden vor der Zuweisung in das Format Long Integer gewandelt.
- 7. Es stehen ihnen 3 verschiedene Datentypvarianten zur Verfügung. Alle Typvarianten sind in getrennten Speicherbereichen angelegt.
Long Integer Variablen 0...255 entsprechen den BIAS-Variablen.
Float Variablen 0...255 entsprechen dem 4-Byte IEEE Format.
Double Float Variablen 0...255 entsprechen dem 8-Byte IEEE Format.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
A3	W	X	Y	Z	00	00	00

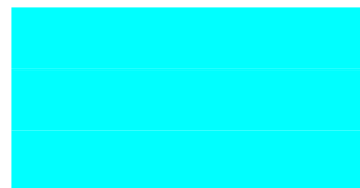
Parameter: D1 = Datentypvorwahl: (Long = 0, Float = 1, Double = 2)
 D2 = Variablennummer:0...255
 D3 = Datentypvorwahl: (Long = 0, Float = 1, Double = 2)
 D4 = Variablennummer:0...255

4 - A4 BIAS-Befehl "[X Variable Y] = Konst."

[X Variable Y] = Konst.

Erklärung:

Der Befehl "[X Variable Y] = Konst." weist der Typvariablen X den Wert der Konstanten zu.



Reglertyp:	Parameter:	Zulässige Eingaben:	<input type="checkbox"/> BIAS-Programm
Alle:	X: Typvorwahl	L = Long Integer F = Float D = Double Float	<input type="checkbox"/> SPS-Programm
	Y:	0-255	<input checked="" type="checkbox"/> Mathematik-Programm
	Konst.	± 231	

Beispiel:

```
0 [D Variable 0 ] = 0
1 [D Variable 1 ] = 0.250000000
2 Wenn [D_Variable 1 ] -< [D_Variable 0 ] dann Springe Fehler
3 [D_Variable 1 ] = SQRT([D_Variable 1 ])
4 ...
Fehler:
50 ...
```

Ergebnis:

Aus der Typvariable 1 wird, wenn der Inhalt nicht negativ ist, die Quadratwurzel gebildet und in die Typvariable 1 zurückgespeichert, sonst wird die BIAS-Abarbeitung an der Sprungmarke Fehler fortgesetzt

Hinweis:

- Typvariablen werden vor der Zuweisung in das Format Long Integer gewandelt.
- Es stehen ihnen 3 verschiedene Datentypvarianten zur Verfügung. Alle Typvarianten sind in getrennten Speicherbereichen angelegt.
Long Integer Variablen 0...255 entsprechen den BIAS-Variablen.
Float Variablen 0...255 entsprechen dem 4-Byte IEEE Format.
Double Float Variablen 0...255 entsprechen dem 8-Byte IEEE Format.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
A4	X	Y	Z(low)			Z(high)	
							00

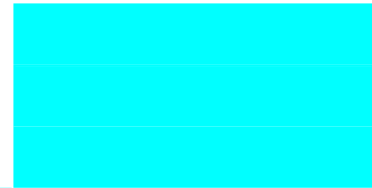
Parameter: D1 = Datentypvorwahl: (Long = 0, Float = 1, Double = 2)
D2 = Variablennummer: 0...255
D3 – D6 = Konstante: ± 231

4 - A5 BIAS-Befehl "[Variable [X]] = Konst."

[Variable [X]] = Konst.

Erklärung:

Der Befehl "[Variable [X]] = Konst." lädt die Konstanten in die Variable, die durch den Inhalt der Variablen X adressiert wird.



Reglertyp:	Parameter:	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm
Alle:	X:	0...255 Zulässiger Variableninhalt: 0...255	<input checked="" type="checkbox"/> SPS-Programm
	Konst.	± 231	<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 [Variable 0] = 10
1 [Variable [0]] = 10000

Ergebnis:

Die Variable 10 wird mit dem Wert 10000 geladen

Hinweis:

- 10. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
- 11. Die Variableninhalte werden in der "Diagnose", "BIAS- Diagnosesseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
A5	X	Y(low)			Y(high	00	00
)		

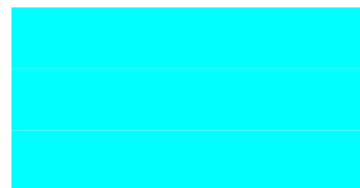
Parameter: D1 = Variablennummer:0...255
D2, D3 = Konstante: ± 231

4 - A6 BIAS-Befehl "[Variable [X]] = [Variable Y]

[Variable [X]] = [Variable Y]

Erklärung:

Der Befehl "[Variable [X]] = [Variable Y]" lädt den Inhalt der Variable Y in die Variable, die durch den Inhalt der Variablen X adressiert wird.



Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X:	0...255 Zulässiger Variableninhalt: 0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	Y	0...255	

Beispiel:

0 [Variable 0] = 10
 1 [Variable 1] = 100000
 2 [Variable [0]] = [Variable 1]

Ergebnis:

Die Variable 10 wird mit dem Wert 100000 geladen

Hinweis:

12. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
13. Die Variableninhalte werden in der "Diagnose", "BIAS- Diagnoseseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
A6	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer:0...255
 D2 = Variablennummer:0...255

4 - A7 BIAS-Befehl "[Variable X] = [Variable [Y]]"

[Variable X] = [Variable [Y]]

Erklärung:

Der Befehl "[Variable X] = [Variable [Y]]" lädt den Inhalt der Variablen, die durch den Inhalt der Variablen Y adressiert wird, in die Variable X.

Reglertyp:	Parameter:	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm
Alle:	X:	0...255	<input checked="" type="checkbox"/> SPS-Programm
	Y	0...255 Zulässiger Variableninhalt: 0...255	<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 [Variable 0] = 10
 1 [Variable 10] = 100000
 2 [Variable 1] = [Variable [0]]

Ergebnis:

Die Variable 1 wird mit dem Wert 100000 geladen

Hinweis:

14. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
15. Die Variableninhalte werden in der "Diagnose", "BIAS- Diagnoseseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
A7	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer:0...255
 D2 = Variablennummer:0...255

4 - A8 BIAS-Befehl “[Variable X] = [Variable Y] ? [Variable Z]“

Erklärung:

Der Befehl “[Variable X] = [Variable Y] ? [Variable Z]“ weist der Variablen X das Ergebnis der ausgewählten logischen Verknüpfung aus den Variablen Y und Z zu.

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X: Y:	0...255 0...255	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	Z:	0...255	
	?	& UND (11) ODER (12) >> Schiebe Rechts (13) << Schiebe Links (14) rl Rotiere Links (15) rr Rotiere Rechts (16) ^ Exklusive ODER (17) Wert in () für Befehlskodierung s.u. Zulässiger Variableninhalt der Variablen Z bei der Befehlskodierung 13 – 16 1...31	

Beispiel:

0 [Variable 0] = 1
 1 [Variable 1] = 15
 2 [Variable 2] = [Variable 0] << [Variable 1]

Ergebnis:

Die Variable 0 wird um 15 Bit nach links geschoben. Die Variable 2 hat den Wert 65536.

Hinweis:

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung.
- Dieser Befehl steht beim 631, 635, 637 ab der Firmware V6.19 zur Verfügung.
- Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
- Die Variableninhalte werden in der "Diagnose", "BIAS- Diagnosesseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
A8	X	Y	?	Z	00	00	00

Parameter: D1 = Variablennummer: 0...255
 D2 = Variablennummer: 0...255
 D3 = logische Verknüpfung (11...17) (siehe [Erklärung](#))
 D4 = Variablennummer: 0...255

4 - A9 BIAS-Befehl “[Variable X] = [Variable Y] ? Konst.“

[Variable X] = [Variable Y] ? Z

Erklärung:

Der Befehl “[Variable X] = [Variable Y] ? Konst.“ weist der Variablen X das Ergebnis der ausgewählten logischen Verknüpfung aus den Variablen Y und Konstanten zu.

Reglertyp:	Parameter:	Zulässige Eingaben:	<input checked="" type="checkbox"/> BIAS-Programm <input checked="" type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X: Y:	0...255 0...255	
	Konst:	± 231	
	?	& UND (11) ODER (12) >> Schiebe Rechts (13) << Schiebe Links (14) rl Rotiere Links (15) rr Rotiere Rechts (16) ^ Exklusive ODER (17) Wert in () für Befehlskodierung s.u. Zulässiger Inhalt der Konstanten bei der Befehlskodierung 13 – 16 1...31	

Beispiel:

0 [Variable 0] = 0x0000 00FF
 1 [Variable 1] = [Variable 0] | 0x0000 FF00

Ergebnis:

Die Variable 0 wird mit der Konstanten ODER verknüpft und das Ergebnis in der Variablen 1 gespeichert. Die Variable 1 hat den Wert 65535.

Hinweis:

- Dieser Befehl steht beim 637f ab der Firmware V8.19 zur Verfügung.
- Dieser Befehl steht beim 631, 635, 637 ab der Firmware V6.19 zur Verfügung.
- Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
- Die Variableninhalte werden in der "Diagnose", "BIAS- Diagnoseseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
A9	X	Y	?	Z (low)			Z (high)

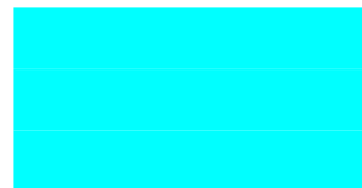
Parameter: D1 = Variablennummer: 0...255
 D2 = Variablennummer: 0...255
 D3 = logische Verknüpfung (11...17) (siehe [Erklärung](#))
 D4 - D7 = Konstante: ± 231 Ausnahme D3 = 13, 14, 15, 16

4 - B0 BIAS-Befehl "[D_Variable X] = [D_Variable Y] + [D_Variable Z]"

$$[D_Variable X] = [D_Variable Y] + [D_Variable Z]$$

Erklärung:

Der Befehl "[D_Variable X] = [D_Variable Y] + [D_Variable Z]" addiert den Inhalt der Double-Variablen Y zu dem Inhalt der Double-Variablen Z und speichert die Summe in der Double-Variablen X.



Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X:	0...255	<input type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	Y	0...255	
	Z	0...255	

Beispiel:

0 [D Variable 0] = 1.250000000

1 [D Variable 1] = 0.250000000

2 [D_Variable 2] = [D_Variable 0] + [D_Variable 1]

Ergebnis:

Es werden die Double-Variablen 0 und 1 addiert und die Summe wird in der Double-Variablen 2 gespeichert.

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
B0	X	Y	Z	00	00	00	00

Parameter: D1: Variablennummer: 0...255

D2: Variablennummer: 0...255

D3: Variablennummer: 0...255

4 - B1 BIAS-Befehl “[D_Variable X] = [D_Variable Y] - [D_Variable Z]

$$[D_Variable X] = [D_Variable Y] - [D_Variable Z]$$

Erklärung:

Der Befehl “[D_Variable X] = [D_Variable Y] - [D_Variable Z]“ subtrahiert den Inhalt der Double-Variablen Z von dem Inhalt der Double-Variablen Y und speichert die Differenz in der Double-Variablen X.

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X:	0...255	<input type="checkbox"/> BIAS-Programm
	Y	0...255	<input type="checkbox"/> SPS-Programm
	Z	0...255	<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 [D Variable 0] = 1.250000000

1 [D Variable 1] = 0.250000000

2 [D_Variable 2] = [D_Variable 0] - [D_Variable 1]

Ergebnis:

Es wird der Inhalt der Double-Variablen 1 von dem Inhalt Double- Variablen 0 subtrahiert und die Differenz wird in der Double-Variablen 2 gespeichert.

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
B1	X	Y	Z	00	00	00	00

Parameter: D1: Variablennummer: 0...255

D2: Variablennummer: 0...255

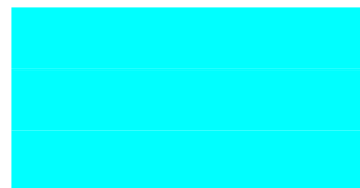
D3: Variablennummer: 0...255

4 - B2 BIAS-Befehl "[D_Variable X] = [D_Variable Y] * [D_Variable Z]

$$[D_Variable X] = [D_Variable Y] * [D_Variable Z]$$

Erklärung:

Der Befehl "[D_Variable X] = [D_Variable Y] * [D_Variable Z]" multipliziert den Inhalt der Double-Variablen Y mit dem Inhalt der Double-Variablen Z und speichert das Produkt in der Double-Variablen X.



Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X:	0...255	<input type="checkbox"/> BIAS-Programm
	Y	0...255	<input type="checkbox"/> SPS-Programm
	Z	0...255	<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 [D Variable 0] = 1.250000000

1 [D Variable 1] = 0.250000000

2 [D_Variable 2] = [D_Variable 0] * [D_Variable 1]

Ergebnis:

Es werden die Double-Variablen 0 und 1 multipliziert und das Produkt wird in der Double-Variablen 2 gespeichert.

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
B2	X	Y	Z	00	00	00	00

Parameter: D1: Variablennummer: 0...255

D2: Variablennummer: 0...255

D3: Variablennummer: 0...255

4 - B3 BIAS-Befehl “[D_Variable X] = [D_Variable Y] / [D_Variable Z]

$$[D_Variable X] = [D_Variable Y] / [D_Variable Z]$$

Erklärung:

Der Befehl “[D_Variable X] = [D_Variable Y] / [D_Variable Z]“ dividiert den Inhalt der Double-Variablen Y durch den Inhalt der Double-Variablen Z und speichert den Quotienten in der Double-Variablen X.

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X:	0...255	<input type="checkbox"/> BIAS-Programm
	Y	0...255	<input type="checkbox"/> SPS-Programm
	Z	0...255	<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 [D Variable 0] = 1.250000000

1 [D Variable 1] = 0.250000000

2 [D_Variable 2] = [D_Variable 0] / [D_Variable 1]

Ergebnis:

Die Double-Variablen 0 wird durch die Double-Variable 1 dividiert und der Quotient wird in der Double-Variablen 2 gespeichert.

Hinweis:

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
B3	X	Y	Z	00	00	00	00

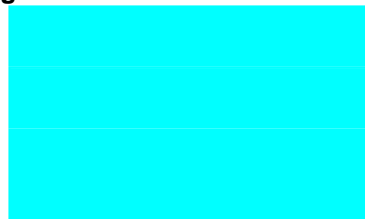
Parameter: D1 = Variablennummer:0...255
D2 = Variablennummer:0...255
D3 = Variablennummer:0...255

4 - B4 BIAS-Befehl "Wenn [D_Variable X] ? [D_Variable Y]

Wenn [D_Variable X] ? [D_Variable Y] dann Springe Z

Erklärung:

Der Befehl "Wenn [D_Variable X] ? [D_Variable Y]" erlaubt Programmverzweigungen durch den Vergleich der Double-Variablen X mit der Double-Variablen Y. Ist die angegebene Bedingung erfüllt, wird die BIAS-Abarbeitung an der angegebenen Adresse fortgesetzt. Andernfalls wird der folgende BIAS-Befehl ausgeführt.



Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X:	0...255	<input type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	?:	> größer (0) < kleiner (1) == gleich (2) >= größer gleich (3) <= kleiner gleich (4) != ungleich (5) -> positiv (6) -< negativ (7) Wert in () für Befehlskodierung s.u.	
	Y	0...255	
	Z: Adresse:	0...1499 Die maximal zulässige Sprungadresse entspricht der Länge des BIAS-Programms. Alternativ zu den Zeilennummern können Label (Sprungmarken) verwendet werden.	

Beispiel:

```

0 [D Variable 0 ] = 0
1 [D Variable 1 ] = 0.250000000
2 Wenn [D_Variable 1 ] -< [D_Variable 0 ] dann Springe Fehler
3 [D_Variable 1 ] = SQRT([D_Variable 1 ])
4 ...

```

Fehler:

```
50 ...
```

Ergebnis:

Aus der Typvariable 1 wird, wenn der Inhalt nicht negativ ist, die Quadratwurzel gebildet und in die Typvariable 1 zurückgespeichert, sonst wird die BIAS-Abarbeitung an der Sprungmarke Fehler fortgesetzt

Hinweis:

Die maximal zulässige Adresse ist abhängig von der Länge des BIAS-Programms und entspricht der Satznummer des letzten BIAS-Satzes.

Alternativ zu den Satznummern können Label (Sprungmarken) verwendet werden.

17. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
18. Die Variableninhalte werden in der "Diagnose", "BIAS-Diagnoseseite" zyklisch angezeigt.
19. Der Vergleich beider Werte wird durch die Subtraktion der Konstanten vom Variableninhalt ausgeführt.
Der bedingte Sprung ist vom Vorzeichen des Ergebnis abhängig.
20. Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
B4	X	?	Y	Z(low)	Z(high)	00	00

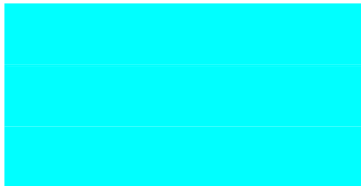
Parameter: D1 = Variablennummer:0...255
D2 = Bedingung: 0...7 ([siehe Erklärung](#))
D3 = Variablennummer:0...255
D4, D5 = Adresse: 0...1499

4 - B5 BIAS-Befehl "[D_Variable X] = SIN {[D_Variable Y]}"

$$[D_Variable\ X] = \text{SIN}\{[D_Variable\ Y]\}$$

Erklärung:

Der Befehl "[D_Variable X] = SIN{[D_Variable Y]}" berechnet den Sinus aus dem Inhalt der Double-Variable Y und speichert sie in die Double-Variable X. Der Winkel muss als Radiant in der Double-Variablen Y stehen.



Reglertyp:	Parameter:	Zulässige Eingaben:	<input type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
Alle:	X:	0...255	
	Y	0...255	

Beispiel:

- 0 [D Variable 0] = 30
- 1 [D Variable 1] = 180
- 2 [D Variable 2] = 3.141592741
- 3 [D_Variable 3] = [D_Variable 0] * [D_Variable 2]
- 4 [D_Variable 3] = [D_Variable 3] / [D_Variable 1]
- 5 [D_Variable 0] = SIN([D_Variable 3])

Ergebnis:

Es soll der Sinus von 30° gebildet werden. Dazu wird zu erst der Radiant gebildet und in der Double-Variablen 3 gespeichert. Aus der Double-Variablen 3 kann der Sinuswert berechnet werden und das Ergebnis wird in die Double-Variable 0 gespeichert.

Hinweis:

- 16. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
- 17. Die Double-Variableninhalte werden in der "Diagnose", "BIAS- Mathematikseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
B5	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer:0...255
 D2 = Variablennummer:0...255

4 - B6 BIAS-Befehl "[D_Variable X] = COS {[D_Variable Y]}"

$$[D_Variable\ X] = \text{COS}\{[D_Variable\ Y]\}$$

Erklärung:

Der Befehl "[D_Variable X] = COS{[D_Variable Y]}" berechnet den Kosinus aus dem Inhalt der Double-Variable Y und speichert sie in die Double-Variable X.

Der Winkel muss als Radiant in der Double-Variablen Y stehen.

Reglertyp:	Parameter:	Zulässige Eingaben:	
Alle:	X:	0...255	<input type="checkbox"/> BIAS-Programm <input type="checkbox"/> SPS-Programm <input checked="" type="checkbox"/> Mathematik-Programm
	Y	0...255	

Beispiel:

0 [D Variable 0] = 30

1 [D Variable 1] = 180

2 [D Variable 2] = 3.141592741

3 [D_Variable 3] = [D_Variable 0] * [D_Variable 2]

4 [D_Variable 3] = [D_Variable 3] / [D_Variable 1]

5 [D_Variable 0] = COS([D_Variable 3])

Ergebnis:

Es soll der Kosinus von 30° gebildet werden.

Dazu wird zu erst der Radiant gebildet und in der Double-Variablen 3 gespeichert.

Aus der Double-Variablen 3 kann der Kosinuswert berechnet werden und das Ergebnis wird in die Double-Variable 0 gespeichert.

Hinweis:

18. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.

19. Die Double-Variableninhalte werden in der "Diagnose", "BIAS- Mathematikseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
B6	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer:0...255

D2 = Variablennummer:0...255

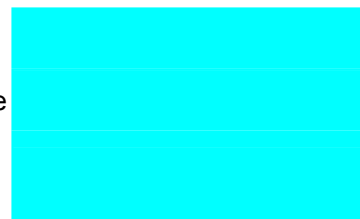
4 - B7 BIAS-Befehl "[D_Variable X] = SQRT {[D_Variable Y]}"

$$[D_Variable X] = SQRT\{[D_Variable Y]\}$$

Erklärung:

Der Befehl "[D_Variable X] = SQRT{[D_Variable Y]}" berechnet die Quadratwurzel aus dem Inhalt der Double-Variable Y und speichert sie in die Double-Variable X.

Die Double-Variable Y muss positiv sein, aus einer negativen Wert kann keine Wurzel gebildet werden.



Reglertyp:	Parameter:	Zulässige Eingaben:	<input type="checkbox"/> BIAS-Programm
Alle:	X:	0...255	<input type="checkbox"/> SPS-Programm
	Y	0...255	<input checked="" type="checkbox"/> Mathematik-Programm

Beispiel:

0 [D Variable 0] = 0
 1 [D Variable 1] = 0.250000000
 2 Wenn [D_Variable 1] -< [D_Variable 0] dann Springe Fehler
 3 [D_Variable 1] = SQRT([D_Variable 1])
 4 ...
 Fehler:
 50 ...

Ergebnis:

Aus der Typvariable 1 wird, wenn der Inhalt nicht negativ ist, die Quadratwurzel gebildet und in die Typvariable 1 zurückgespeichert, sonst wird die BIAS-Abarbeitung an der Sprungmarke Fehler fortgesetzt

Hinweis:

- 20. Nach dem Einschalten des Reglers haben alle Variablen den Inhalt 0.
- 21. Die Double-Variableninhalte werden in der "Diagnose", "BIAS- Mathematikseite" zyklisch angezeigt.

Befehlskodierung:

Code	D1	D2	D3	D4	D5	D6	D7
B7	X	Y	00	00	00	00	00

Parameter: D1 = Variablennummer:0...255
 D2 = Variablennummer:0...255

AUSTRALIEN
Eurotherm Pty Ltd
Unit 1
20-22 Foundry Road
Seven Hills
New South Wales 2147
Tel: +61 2 9838 0099
Fax: +61 2 9838 9288

CHINA
Eurotherm Pty Ltd
Apt. 1805, 8 Building Hua Wei Li
Chao Yang District,
Beijing 100021
Tel: +86 10 87785520
Fax: +86 10 87790272

DÄNEMARK
Eurotherm GmbH
Enghavevej 11
DK-7100 Vejle
Tel: +45 70 201311
Fax: +45 70 201312

DEUTSCHLAND
SSD DRIVES GmbH
Von-Humboldt-Straße 10
64646 Heppenheim
Tel: +49 6252 7982-00
Fax: +49 6252 7982-05

ENGLAND
SSD Drives Ltd
New Courtwick Lane
Littlehampton
West Sussex BN17 7RZ
Tel: +44 1903 737000
Fax: +44 1903 737100

FRANKREICH
SSD Drives SAS
15 Avenue de Norvège
Villebon sur Yvette
91953 Courtaboeuf Cedex / Paris
Tel: +33 1 69 185151
Fax: +33 1 69 185159

HONG KONG
Eurotherm Ltd
Unit D
18/F Gee Chang Hong Centre
65 Wong Chuk Hang Road
Aberdeen
Tel: +852 2873 3826
Fax: +852 2870 0148

INDIEN
Eurotherm DEL India Ltd
152, Developed Plots Estate
Perungudi
Chennai 600 096, India
Tel: +91 44 2496 1129
Fax: +91 44 2496 1831

IRLAND
SSD Drives
2004/4 Orchard Ave
Citywest Business Park
Naas Rd, Dublin 24
Tel: +353 1 4691800
Fax: +353 1 4691300

ITALIEN
SSD Drives SpA
Via Gran Sasso 9
20030 Lentate Sul Seveso
Milano
Tel: +39 0362 557308
Fax: +39 0362 557312

JAPAN
PTI Japan Ltd
7F, Yurakucho Building
10-1, Yuakucho 1-Chome
Chiyoda-ku, Tokyo 100-0006
Tel: +81 3 32132111
Fax: +81 3 32131900

KANADA
SSD Drives Inc
880 Laurentian Drive
Burlington
Ontario
Canada, L7N 3V6
Tel: +1 905 333-7787
Fax: +1 905 632-0107

KOREA
Myungshin Drives Co. Ltd.
1308, Daeryung Techno Town
8th Bldg., 481-11 Gasan-Dong,
Geumcheon-Gu,
Seoul 153-803
Tel: +82 2 2163 6677
Fax: +82 2 2163 8982

NIEDERLANDE
Eurotherm BV
Genielaan 4
2404CH
Alphen aan den Rijn
Tel: +31 172 411 752
Fax: +31 172 417 260

POLEN
OBR-USN
ul. Batorego 107
PL 87-100 Torun
Tel: +48 56 62340-21
Fax: +48 56 62344-25

RUMÄNIEN
Servosisteme SRL
Sibiu 17
061535 Bukarest
Tel: +40 723348999
Fax: +40 214131290

SPANIEN
Eurotherm Espana S.A.
Pol. Ind. Alcobendas
C/ La Granja, 74
28108 Madrid
Tel: +34 91 661 60 01
Fax: +34 91 661 90 93

SCHWEDEN
SSD Drives AB
Montörgatan 7
S-30260 Halmstad
Tel: +46 35 177300
Fax: +46 35 108407

SCHWEIZ
Indur Antriebstechnik AG
Margarethenstraße 87
CH 4008 Basel
Tel: +41 61 27929-00
Fax: +41 61 27929-10

U.S.A
SSD Drives Inc.
9225 Forsyth Park Drive
Charlotte
North Carolina 28273-3884
Tel: +1 704 588 3246
Fax: +1 704 588 3249

Weitere Niederlassungen und Vertretungen in:

Ägypten · Argentinien · Bangladesch · Brasilien · Chile · Costa Rica · Ecuador · Griechenland · Indonesien · Island · Israel
Kolumbien · Kuwait · Litauen · Malaysia · Marokko · Mexico · Neuseeland · Nigeria · Peru · Philippinen · Portugal
Österreich · Saudi Arabien · Singapur · Slowenien · Sri Lanka · Süd Afrika · Taiwan · Thailand · Tschechien
Türkei · Ungarn · Vereinigte Arabische Emirate · Vietnam · Zypern

SSD Drives GmbH

Im Sand 14 76669 Bad Schönborn Tel.: +49 7253 9404-0, Fax: +49 7253 9404-99
www.ssddrives.com · ssd@ssddrives.de