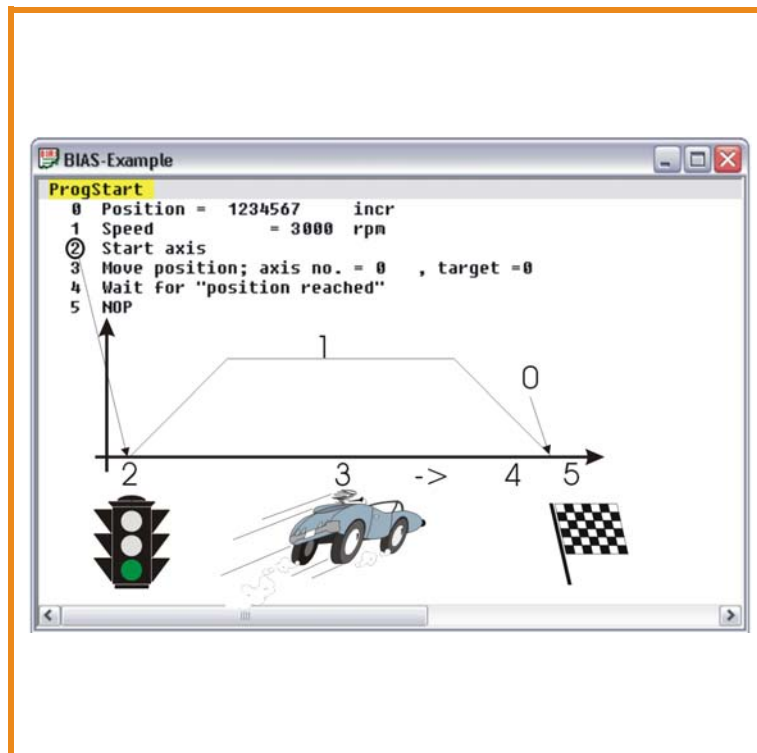


# BIAS

## BIAS - Command



UL: 07-01-08-02



Product manual 631

---

UL: 07-01-05-06



Product manual 635

---

UL: 07-02-08-03



Product manual 637

---

UL: 07-02-09-01



Product manual 637+

---

UL: 07-02-10-01



Product manual 637f

---

CD-ROM:



EASYRIDER® Windows - Software

---

**©SSD Drives GmbH.**

All rights reserved. No portion of this description may be produced or processed in any form without the consent of the company.

Changes are subject to change without notice.

**SSD Drives** has registered in part trademark protection and legal protection of designs. The handing over of the descriptions may not be construed as the transfer of any rights.

Made in Germany, 2004

	Page
<b>The most important thing first .....</b>	<b>6</b>
<b>1 General .....</b>	<b>7</b>
1.0 Introduction .....	7
1.1. Program layout.....	8
1.2. Execute a BIAS program .....	8
1.3. Execute a PLC program .....	9
1.4. Execute a Mathematics program .....	9
1.5. BIAS example programs.....	10
<b>2 BIAS commands overview.....</b>	<b>22</b>
<b>3 BIAS-Command group .....</b>	<b>23</b>
3.1 BIAS-Command group "Move commands".....	23
3.2 BIAS-Command group „Parameter commands“ .....	24
3.3 BIAS Command group "Variable commands" .....	26
3.4 BIAS Command group "Flag commands".....	26
3.5 BIAS Command group "Conditional jump commands".....	27
3.6 BIAS Commands group "Program control commands".....	27
3.7 BIAS Command group "Mathematic commands" .....	28
3.8 BIAS Command group "Output commands".....	28
3.9 BIAS Command group "CAN-Commands".....	29
3.10 BIAS Command group "637f V8.xx commands" .....	29
<b>4 BIAS - Command .....</b>	<b>30</b>
4 - 00 BIAS-Command "Move position" .....	30
4 - 01 BIAS-Command "Move incremental position" .....	31
4 - 02 BIAS-Command "Move datum".....	32
4 - 03 BIAS-Command "Move infinite positive".....	34
4 - 04 BIAS-Command "Move infinite negative" .....	35
4 - 05 BIAS-Command "Move synchron".....	36
4 -06 BIAS-Command "Move CAM profile".....	38
4 - 07 BIAS-Command "Synchronous settings 1".....	39
4 - 08 BIAS-Command "Synchronous settings 2".....	40
4 - 09 BIAS-Command "Move PID, speed".....	42
4 - 0A BIAS-Command "Move PID, torque" .....	44
4 - 0B BIAS-Command "Set position [axis no. ] = const. " .....	46
4 - 0C BIAS-Command "Set position [axis no. ] = [variable X]".....	47
4 - 0D BIAS-Command "Move relative" .....	48
4 - 0E BIAS-Command "Start axis" .....	49
4 - 0F BIAS-Command "Stop axis".....	50
4 - 10 BIAS-Command "Move position + parameter" .....	51
4 - 11 BIAS-Command "Move incremental position + parameter".....	52
4 - 12 BIAS-Command "Move datum + parameter".....	53
4 - 13 BIAS-Command "Move infinite positive + parameter".....	55
4 - 14 BIAS-Command "Move infinite negative + parameter".....	56
4 - 15 BIAS-Command "Move synchron + parameter" .....	57
4 - 16 BIAS-Command "Move analogue value + integrator" .....	58
4 - 17 BIAS-Command "Move speed + integrator".....	59
4 - 1A BIAS command "Cycle length = Const. ".....	60
4 - 1B BIAS command "Cycle length = [variable X] " .....	60
4 - 1C BIAS-Command "Load parameter set X = [variable [Y]]".....	61
4 - 1F BIAS-Command "Stop axis + parameter".....	65
4 - 20 BIAS-Command "Position = const. ".....	66
4 - 21 BIAS-Command "Speed = const. " .....	67
4 - 22 BIAS-Command "Acceleration = const. ".....	68
4 - 23 BIAS-Command "Deceleration = const. " .....	69
4 - 24 BIAS-Command "Gear factor = const. " .....	70
4 - 25 BIAS-Command "Position reached" window = const. ....	71
4 - 26 BIAS-Command "Remaining position= const. " .....	72

	Page
4 - 27 BIAS-Command "Ramp filter = const.,[variable Y]"	73
4 - 28 BIAS-Command "Actual position X = const."	74
4 - 29 BIAS-Command "If actual position X ? const. then jump"	75
4 - 2A BIAS-Command "If actual position X ? [variable Y] then jump"	76
4 - 2B BIAS-Command "Sensor window = const."	77
4 - 2C BIAS-Command "Sensor position = const."	79
4 - 2D BIAS-Command "Sensor adjustment 1 = const."	80
4 - 2E BIAS-Command "Sensor adjustment 2 = const."	81
4 - 2F BIAS-Command "Update parameter"	82
4 - 30 BIAS-Command "Position = [variable X]"	83
4 - 31 BIAS-Command "Speed = [variable X]"	84
4 - 32 BIAS-Command "Acceleration = [variable X]"	85
4 - 33 BIAS-Command "Deceleration = [variable X]"	86
4 - 34 BIAS-Command "Gear factor = [variable X]"	87
4 - 35 BIAS-Command "Position reached" window = [variable X]"	88
4 - 36 BIAS-Command "Remaining position = [variable X]"	89
4 - 37 BIAS-Command "Maximum current = [variable X]"	90
4 - 38 BIAS-Command "Actual position X = [variable Y]"	91
4 - 39 BIAS-Command "Analog output X = [variable Y]"	92
4 - 3A BIAS-Command "PID scaling"	93
4 - 3B BIAS-Command "Sensor window = [variable X]"	94
4 - 3C BIAS-Command "Sensor position = [variable X]"	96
4 - 3D BIAS-Command "Sensor adjustments 1 = [variable X]"	97
4 - 3 BIAS-Command "Sensor adjustments 2 = [variable X]"	98
4 - 3F BIAS-Command "PID parameter"	99
4 - 40 BIAS-Command "[Variable X] = position"	100
4 - 41 BIAS-Command "[Variable X] = speed"	101
4 - 42 BIAS-Command "[Variable X] = acceleration"	102
4 - 43 BIAS-Command "[Variable X] = deceleration"	103
4 - 44 BIAS-Command "[Variable X] = gear factor"	104
4 - 45 BIAS-Command "[Variable X] = block number"	105
4 - 46 BIAS-Command "[Variable X] = actual position Y"	106
4 - 47 BIAS-Command "[Variable X] = analog input Y"	107
4 - 48 BIAS-Command "[Variable X] = latch position Y"	108
4 - 49 BIAS-Command "[Variable X] = actual speed Y"	109
4 - 4A BIAS-Command "[Variable X] = latch status Y"	110
4 - 4B BIAS-Command "[Variable X] = position Y"	111
4 - 4C BIAS-Command "[Variable X] = value Y"	112
4 - 4D BIAS-Command "[Variable X] = axis status, axis no. Y"	120
4 - 50 BIAS-Command "NOP"	122
4 - 51 BIAS-Command "End of program, mode ="	123
4 - 52 BIAS-Command "Sub-program"	124
4 - 53 BIAS-Command "End of sub-program "	125
4 - 54 BIAS-Command "PLC-program"	126
4 - 55 BIAS-Command "Jump const."	127
4 - 56 BIAS-Command "Jump [variable X]"	128
4 - 57 BIAS-Command "BIAS-execution pointer = const."	129
4 - 58 BIAS-Command "Wait for "position reached""	130
4 - 59 BIAS-Command "Wait time = const."	131
4 - 5A BIAS-Commands "Wait time = [variable X]"	132
4 - 5B BIAS-Command "BIAS-execution pointer = [variable X]"	133
4 - 5C BIAS-Command "Jump [var.[ X ]]; length = Y ; from line"	134
4 - 5D BIAS-Command "Execute X commands"	135
4 - 5F BIAS-Command "Virtual program"	136
4 - 60 BIAS-Command "Flag X = const."	137
4 - 61 BIAS-Command "If flag X ? const. then jump "	138
4 - 62 BIAS-Command "Flag X = flag Y"	139
4 - 63 BIAS-Command "Flag X = input Y"	140
4 - 64 BIAS-Command "Flag X = output Y"	141
4 - 65 BIAS-Command "Flag X = flag Y & flag Z"	142
4 - 66 BIAS-Command "Flag X = flag Y   flag Z"	143

	Page
4 - 67 BIAS-Command "Flag X = flag Y ^ flag Z"	144
4 - 68 BIAS-Command "Flag X = !flag Y"	145
4 - 69 BIAS-Command "Flag X = status Y"	146
4 - 6A BIAS-Command "If status X ? const. then jump"	147
4 - 6B BIAS-Command "Mode X = const."	148
4 - 6C BIAS-Command "Flag X = [variable Y], number Z"	149
4 - 6D BIAS-Command "[Variable X]. bit [Y] = const."	150
4 - 6E BIAS-Command "If [variable X]. bit Y ? const. then jump"	151
4 - 6F BIAS-Command "Axis state, axis no. X, bit Y = const. [flag Z]"	152
4 - 70 BIAS-Command "If input X ? const. then jump"	154
4 - 71 BIAS-Command " If output X ? const. then jump"	156
4 - 72 BIAS-Command "Output X = const."	158
4 - 73 BIAS-Command "Output X = flag Y"	159
4 - 78 BIAS-Command "IBT-mask number = X, LED status [variable Y]"	160
4 - 79 BIAS-Command "IBT-notification number = X,LED status[variable Y]"	161
4 - 7A BIAS-Command "CAN-command = [variable X]"	162
4 - 7B BIAS-Command "IBT- data transfer"	164
4 - 7C BIAS-Command "CAN2-command = [variable X]"	165
4 - 80 BIAS-Command "[Variable X] = const."	167
4 - 81 BIAS-Command "If [variable X] ? const."	168
4 - 82 BIAS-Command "[Variable X] = [variable Y] + const."	170
4 - 83 BIAS-Command "[Variable X] = [variable Y] - const."	171
4 - 84 BIAS-Command "[Variable X] = [variable Y] * const."	172
4 - 85 BIAS-Command "[Variable X] = [variable Y] / const."	173
4 - 86 BIAS-Command "[Variable X] = flag Y"	174
4 - 87 BIAS-Command "[Variable X] = [variable Y].bit Z ; number"	175
4 - 88 BIAS-Command "[Variable X] = [variable Y]"	176
4 - 89 BIAS-Command "If [variable X] ? [variable Y]"	177
4 - 8A BIAS-Command "[Variable X] = [variable Y] + [variable Z]"	179
4 - 8B BIAS-Command "[Variable X] = [variable Y] - [variable Z]"	180
4 - 8C BIAS-Command "[Variable X] = [variable Y] * [variable Z]"	181
4 - 8D BIAS-Command "[Variable X] = [variable Y] / [variable Z]"	182
4 - 8E BIAS-Command "[Teachvariable X] = [variable Y]"	183
4 - 8F BIAS-Command "[Variable X] = [teachvariable Y]"	184
4 - 90 BIAS-Command "Mathematic program"	185
4 - 91 BIAS-Command "Profile initialization"	186
4 - 92 BIAS-Command "Profile cycle length = [variable X]"	187
4 - 93 BIAS-Command "[Variable X] = profile value"	188
4 - 94 BIAS-Command "Profile value = [variable X]"	189
4 - 97 BIAS-Command " Save table"	190
4 - A0 BIAS-Command "Table ([variable X])= const."	191
4 - A1 BIAS-Command "Table ([variable X]) = [Y variable Z]"	192
4 - A2 BIAS-Command " [X variable Y] = table ([variable Z])"	193
4 - A3 BIAS-Command "[W variable X] = [Y variable Z]"	194
4 - A4 BIAS-Command "[X variable Y] = const."	195
4 - A5 BIAS-Command "[Variable [X]] = const."	196
4 - A6 BIAS-Command "[Variable [X]] = [variable Y]"	197
4 - A7 BIAS-Command "[Variable X] = [variable [Y]]"	198
4 - A8 BIAS-Command "[Variable X ] = [variable Y ] ? [variable Z]"	199
4 - A9 BIAS-Command "[Variable X ] = [variable Y ] ? const."	200
4 - B0 BIAS-Command "[D_variable X]=[D_variable Y]+[D_variable Z]"	201
4 - B1 BIAS-Command "[D_variable X]=[D_variable Y]-[D_variable Z]"	202
4 - B2 BIAS-Command "[D_variable X] = [D_variable Y] * [D_variable Z]"	203
4 - B3 BIAS-Command "[D_variable X] = [D_variable Y] / [D_variable Z]"	204
4 - B4 BIAS-Command "If [D_variable X] ? [D_variable Y]"	205
4 - B5 BIAS-Command "[D_variable X] = SIN ([D_variable Y])"	207
4 - B6 BIAS-Command "[D_variable X] = COS ([D_variable Y])"	208
4 - B7 BIAS-Command "[D_variable X] = SQRT ([D_variable Y])"	209
<b>5 Modification Record</b>	<b>210</b>

Thanks for your confidence choosing our product.

These operating instructions present themselves as an overview of the technical data and features.

Please read the operating instructions before operating the product.

If you have any questions, please contact your nearest SSD Drives representative. Improper application of the product in combination with dangerous voltage can lead to injuries.

In addition, damage can also occur to motors or other products. Therefore please observe our safety precautions strictly.

### **Safety precautions**

We assume that, as an expert, you are familiar with the relevant safety regulations, especially in accordance with VDE 0100, VDE 0113, VDE 0160, EN 50178, the accident prevention regulations of the employers liability insurance company and the DIN regulations and that you are able to use and apply them. As well, relevant European Directives must be observed.

Depending on the kind of application, additional regulations e.g. UL, DIN are subject to be observed.

If our products are operated in connection with components from other manufacturers, their operating instructions are also subject to be observed strictly.

## 630 Programming language BIAS

### 1.0 Introduction

The programming language "BIAS"

#### Bedienersprache für intelligente Antriebs – Steuerungen

was developed to allow the programming of complex and yet clear programs.

Therefore the BIAS commands were divided according to their function into the 9 following command groups:

0. Move command
1. Move command + parameters
2. Parameter commands
3. "Parameter from variables" - commands
4. "Parameter into variables" - commands
5. Control commands
6. Flag commands
7. In-/ output commands
8. Variable commands
9. Mathematics commands 1
10. Mathematics commands 2
11. Floating point commands

#### BIAS – [Command overview](#)

With the softwareversion 5.10 you have the possibility in the drives of the 630 series to program in the additional command groups 10 - 12 mathematical operations. i.e. to calculate online the synchronous profiles.

For further information see motion control manual (UL:10.6.7)

With these commands you will be able to program the required machine process in chains of steps

The size of a program is limited to a maximum of 1500 BIAS commands

The designing of the programs occurs with EASYRIDER software at the PC and can be transmitted into the servo drive via serial communication.

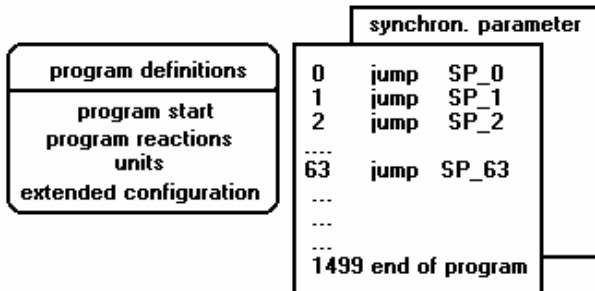
If you create the BIAS program with the **EASYRIDER** shell, jump labels, comments and a unit for the position presettings are provided.

A further possibility is programming or transmitting and controlling the BIAS program via a field bus respectively. The necessary command coding is listed in the command instruction.

**1.1. Program layout**

A BIAS program consists of 3 basic memory areas.

1. **The program definition:**  
contains all definitions for starting and processing a BIAS program, the entries for defining a unit for position presetting and the necessary configurations of the inputs and outputs.
2. **The command memory:**  
contains up to 1500 BIAS commands.
3. **The synchronous parameters:**  
contain the definitions for the 16 synchronous profile blocks and the 2048 supporting points.



The basic memory areas are part of the BIAS program.

In the EASYRIDER-DOS Software this file has the extension \*.ASB.

In the EASYRIDER for Windows Software the extension is \*.WBD.

In the EASYRIDER for Windows Software it is possible to read files with the DOS extension (\*.ASB).

With saving the file in the Windows Version it is converted in the Windows version (\*.WBD).

With the DOS version it is not possible to read files with the \*.WBD extension !!

**1.2. Execute a BIAS program**

The BIAS processing is started in operating mode 5 "position control with BIAS processing" after activating the output stage of the regulator.

The first BIAS block to be executed is determined in the BIAS program definition (Parameter "program start").

After that, the regulator processes one BIAS command sequentially every trajectory cycle.

If the BIAS processing encounters a move command, it can be started with the Low-High slope of the start input.

Serie	Input	Configuration
635/ 637/637+/637f:	X10.11	"Start input BIAS" (Function 0)
631:	X10.9	"Start input " (Function 3)

Alternatively, move commands are started when the start identifier is set before the move command, via the BIAS command "Start axis".

The following blocks will be processed after a successful start.

If the command, "Wait for "position reached"" follows a move command, block processing will only be continued after the target position is reached.

Drive type	Trajectory cycle
631/635/637	1,899ms
637+/637f	0,844ms



### 1.3. Execute a PLC program

A cyclic PLC program for supervisory monitoring tasks can be started parallel to the sequential processing of a BIAS program

The PLC program is started by processing the BIAS command, "**PLC program**".

After the PLC program is activated the programmed PLC commands are processed as of the specified block number.

The command "**end of program, mode = 0**" within a PLC program causes a jump back to the start of the PLC program.

The regulator processes one PLC command sequentially every trajectory cycle.

The reaction of the PLC program to the deactivation of the output stage can be adjusted in the BIAS program definition (parameter "program reaction PLC program"). Thus it is possible to allow the PLC program to continue to process also during the deactivation of the regulator. Is in this mode the first command of the BIAS execution the command "PLC program" the PLC task starts automatically independently of the state (deactive/active) of the drive.

In the plc-loop not all of the BIAS commands are allowed.

In chapter 3 command overview the allowed commands are listed.

The check of allowed commands is done by the drive during run time!

Drive type	Trajectory cycle
631/635/637	1,899ms
637+/637f	0,844ms

### 1.4. Execute a Mathematics program

A 3 task as math program for supervisory calculation can be started parallel to the sequential processing of a BIAS program and/or PLC program.

The mathematics-program is started by processing the BIAS command, "**Mathematics program**".

After the mathematics program is activated the programmed mathematics commands are processed as of the specified block number.

The command "**end of program, mode =0**" within a mathematic - program causes a jump back to the start of the mathematics program.

The command "**end of program, mode =3**" cancels the mathematics program.

The reaction of the mathematics program to the deactivation of the output stage can be adjusted in the BIAS program definition (parameter "program reaction mathematics program"). Thus it is possible to allow the mathematics program to continue to process also during the deactivation of the regulator.

In this mode the command „Mathematic program“ is executed at the first or second line (if the PLC program is on line 1)of the BIAS program or at line 0 , if the drive is not enabled.

The calculation of the mathematics commands is done in the interruptfree calculation time of the drive. In a standard application approx. 10 commands are processed every 2ms

1.5. BIAS example programs

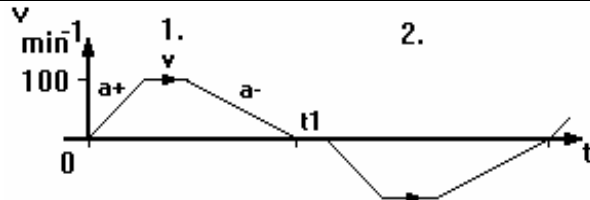
The example programs you will find in the BIAS editor; menu "File"; submenu "load BIAS example"

**1. BIAS-application example: Reverse operation free motor shaft**

**Explanation:** 1. Absolute positioning to the position of 16384 increments  
 2. Waiting time of 1 second  
 3. Absolute positioning to the position of 0 increments

**Filename:** EXP\_001.wbd

**Requirements:** operation mode 5 ("Position control with BIAS-execution")



Programlisting

```

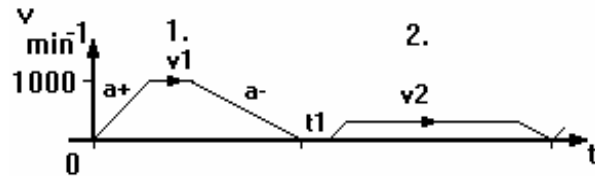
* *****
* !!! Attention !!!
* The program leads to automatic movements of the axis!
PROG_START:
0 Actual position 1 = 0 INCR
* *****
* Definition of the movement profile for the 1st positioning
1 Acceleration = 5000 rpm/s
* Definition of the acceleration ramp a+
2 Speed = 100 rpm
* Definition of the maximum speed v
3 Deceleration = 2500 rpm/s
* Definition of the deceleration ramp a-
4 Position = 16384 INCR
* Definition of the 1st target position in increments
* *****
5 Start axis
* Give start permission (set start mark)
6 Move position, axis no.= 0, target = 0
* start 1st positioning as absolute positioning
* *****
7 Wait for "position reached"
* interrupt BIAS-execution until reaching the target position
8 Wait time 1000 ms
* waiting time (t1) of 1000 ms
* *****
* Definition of the movement profile for the 2nd positioning
9 Position = 0 INCR
* Definition of the 2nd target position, the definitions for the acceleration ramp, the maximum speed
* and/or the deceleration ramp are not changed, so they retain their values and validity
* *****
10 Start axis
* Give start permission (set start mark)
11 Move position, axis no.= 0, target = 0
* start 2nd positioning as absolute positioning
12 Wait for "position reached"
* interrupt BIAS-execution until reaching the target position
* *****
13 End of program, mode 0
* Continue BIAS-execution at the beginning of the program (at block 0 or at label "PROG_START")
* *****
    
```

### 2. BIAS-application example: Feed free motor shaft

**Explanation:** 1. Relative positioning of 100000 increments in the positive direction  
 2. Waiting time of 1 second  
 3. Relative positioning of 32700 increments in the positive direction

**Filename:** EXP\_002.wbd

**Requirements:** Operation mode 5 ("Position control with BIAS-execution")



### Programlisting

```

* *****
* !!! Attention !!!
* The program leads to automatic movements of the axis!
* *****

PROG_START:
* Starting point of the BIAS-Program after every activation of the power stage
* *****

* Definition of the movement profile for the 1st positioning
0 Acceleration = 5000 rpm/s
* Definition of the acceleration ramp a+
1 Deceleration = 2500 rpm/s
* Definition of the deceleration ramp a-

START_POS:
2 Speed = 1000 rpm
* Definition of the maximum speed v1
3 Position = 100000 INCR
* .....Definition of the 1st target position in increments (high resolver resolution ca. 6.1 motor revolutions)
4 Start axis
* Give start permission (set start mark)
5 Move incremental, axis no.= 0, target = 0
* start 1st positioning as relative positioning
* *****

6 Wait for position reached
* interrupt BIAS-execution until reaching the target position
7 Wait time 1000 ms
* waiting time (t1) of 1000 ms
* *****

* Definition of the movement profile for the 2nd positioning
8 Position = 32700 INCR
* Definition of the 2nd target position (high resolver resolution circa 2 motor revolutions)
9 Speed = 250 rpm
* Definition of the maximum speed v2
10 Start axis
* Give start permission (set start mark)
11 Move incremental, axis no.= 0, target = 0
* start 2nd positioning as relative positioning
12 Wait for "position reached"
* interrupt BIAS-execution until reaching the target position
* *****

13 Jump START_POS
* Continue BIAS-execution at the label "START_POS"
* *****
  
```

**3.1 BIAS-application example: Linear unit 635/637**

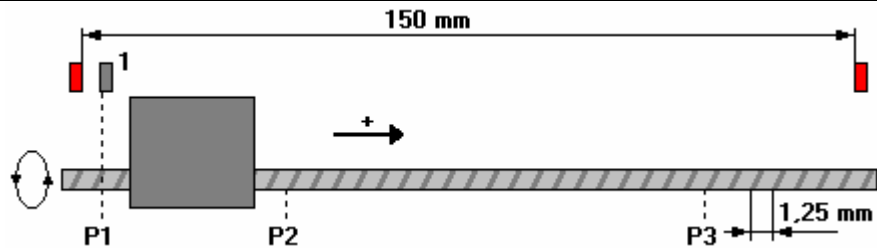
**Explanation:** 1. Reference run (sensor +following zero impulse in negative direction)  
 2. Absolute positioning to position P2 (25 mm)  
 3. Start after external signal (low-high-slope input X10.11) to position P3 (125 mm)  
 4. Start after external signal (low-high-slope input X10.11) to position P2 (25 mm)  
 5. Continue with 3.

**Filename:** EXP\_003.wbd

**Requirements:** Operation mode 5 ("Position control with BIAS-execution")  
 Wire start input X10.11 and configure with function 0 (BIAS start input)  
 Reference sensor (1) wired to X10.24 and configuration X10.24 with function 1 (ref. sensor)

**Spindle operation with**  
 gradient = 1,25 mm  
 total length = 150 mm

1 = reference sensor  
 P1 = reference position  
 P2 = target position 1  
 P3 = target position 2



**Programlisting**

```
* *****
* !!! Attention !!!
* The program leads to automatic movements of the axis!
* Required inputs for the unit " mm" at BIAS-program definitions
* Unit (Drive):
* cog count output drive = 100; cog count input drive = 100 (gear factor 1:1)
* high resolver resolution = Y (16384 increments / motor revolution)
* distance per output turn = 1.25 (gradient of the spindle)
* displayed text for unit = mm
* *****
```

**PROG\_START:**

```
* Definition of the movement profile for the reference run
0 Acceleration = 2000 rpm/s
1 Deceleration = 2000 rpm/s
2 Start axis
3 Move datum, mode = 11, v = 100 rpm, s = 0 mm
4 Wait for position reached
* *****
```

**Pos\_Loop:**

```
* Definition of the movement profile for the 1st positioning P2
5 Acceleration = 25000 rpm/s
6 Deceleration = 15000 rpm/s
7 Move position; v = 2500 rpm, s = 25 mm
* start permission is given via the low-high slope X10.11
8 Wait for "position reached"
* 1st positioning ended
* *****
```

**Definition of the movement profile for the 2nd positioning P3**

```
9 Acceleration = 5000 rpm/s
10 Deceleration = 5000 rpm/s
11 Move position; v = 1500 rpm, s = 125 mm
12 Wait for "position reached"
* 2nd positioning ended
13 Jump Pos_Loop
* *****
```

### 3.2 BIAS-application example: Linear unit 631

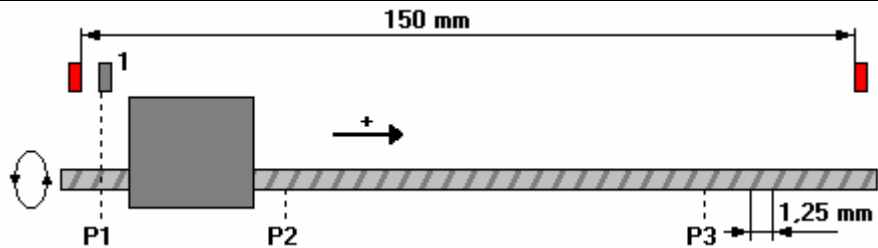
**Explanation:** 1. Reference run (sensor +following zero impulse in negative direction)  
 2. Absolute positioning to position P2 (25 mm)  
 3. Start after external signal (low-high-slope input X10.11) to position P3 (125 mm)  
 4. Start after external signal (low-high-slope input X10.11) to position P2 (25 mm)  
 5. Continue with 3.

**Filename:** E631\_003.wbd

**Requirements:** Operation mode 5 ("Position control with BIAS-execution")  
 Wire start input X10.8 and configure with function 0 (BIAS start input)  
 Reference sensor (1) wired to X10.9 and configuration X10.9 with function 1 (ref. sensor)

**Spindle operation with**  
 gradient = 1,25 mm  
 total length = 150 mm

1 = reference sensor  
 P1 = reference position  
 P2 = target position 1  
 P3 = target position 2



#### Programming

```
* *****
* !!! Attention !!!
* The program leads to automatic movements of the axis!
* Required inputs for the unit " mm" at BIAS-program definitions
* Unit (Drive):
* cog count output drive = 100; cog count input drive = 100 (gear factor 1:1)
* high resolver resolution = Y (16384 increments / motor revolution)
* distance per output turn = 1.25 (gradient of the spindle)
* displayed text for unit = mm
* *****
```

#### PROG\_START:

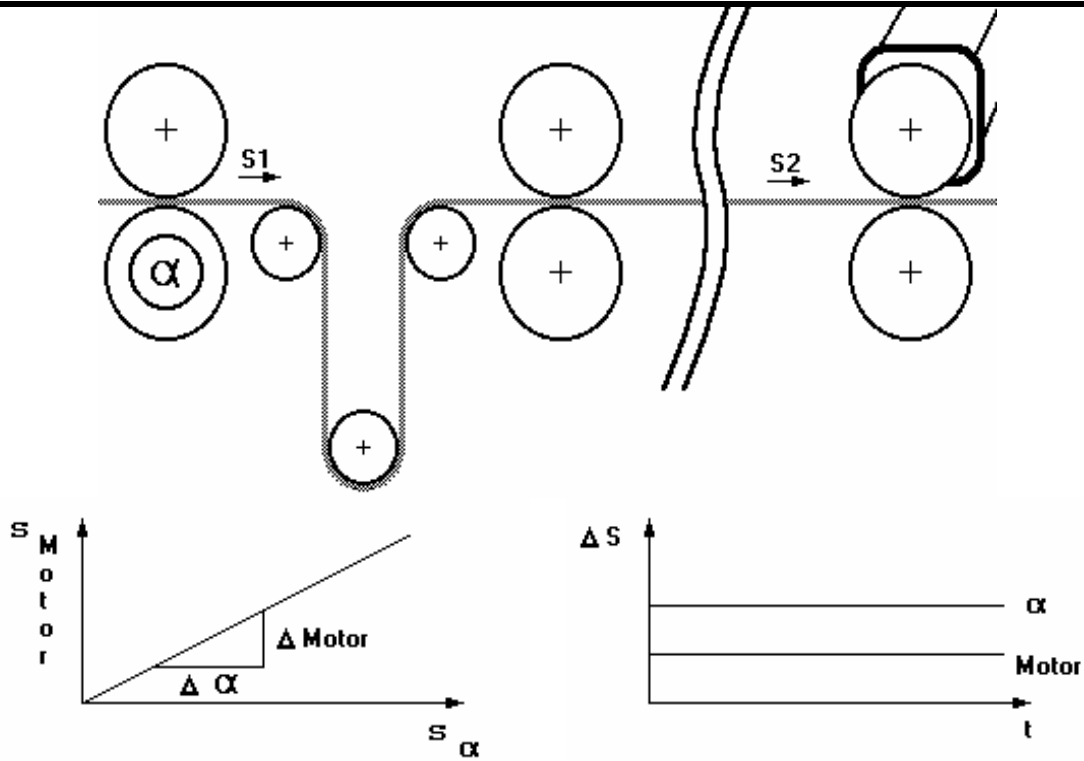
```
* Definition of the movement profile for the reference run
0 Acceleration = 2000 rpm/s
1 Deceleration = 2000 rpm/s
2 Start axis
3 Move datum, mode = 11, v = 100 rpm, s = 0 mm
4 Wait for position reached
* *****
```

#### Pos\_Loop:

```
* Definition of the movement profile for the 1st positioning P2
5 Acceleration = 25000 rpm/s
6 Deceleration = 15000 rpm/s
7 Move position; v = 2500 rpm, s = 25 mm
* start permission is given via the low-high slope X10.11
8 Wait for "position reached"
* 1st positioning ended
* *****
* Definition of the movement profile for the 2nd positioning P3
9 Acceleration = 5000 rpm/s
10 Deceleration = 5000 rpm/s
11 Move position; v = 1500 rpm, s = 125 mm
12 Wait for "position reached"
* 2nd positioning ended
13 Jump Pos_Loop
* *****
```

**4.1 BIAS-application example: Electronic gear box 635/637/637+/637f**

**Explanation:** 1. Move position synchronous to the external incremental encoder ( $\alpha$ ) as long as the input X10.11 is active.  
**Filename:** EXP\_004.wbd  
**Requirements:** Operation mode 5 ("Position control with BIAS-execution")  
 Wire start input X10.11 and configure with function 3 (BIAS input 11)  
 Configure X40-connector with function 1 (incremental input)  
 X40-resolution depending the external encoder (8192 increments / revolution)



**Programmlisting**

```

* *****
* !!! Attention !!!
* necessary I/O-configurations (BIAS-program definition configuration)
* input X10.11 function 3 (BIAS-input 11)
* Configure X40-connector with function 1 (incremental input)
* X40-resolution depending the external encoder (8192 increments / revolution)
* *****

PROG_START:
0 If 11 == 0 then jump PROG_START
1 Position = 0 INCR
* Additional synchron function "format check" deactivated
1 Start axis
2 Move synchron, gear factor = 0.5

       $\frac{\Delta S_1}{\Delta t} = \frac{\Delta S_2}{\Delta t}$  ; s1 = 8192; s2= 4096; gear factor =  $\frac{s_2}{s_1} = \frac{4096}{8192} = 0,5$ 
* Requirement:

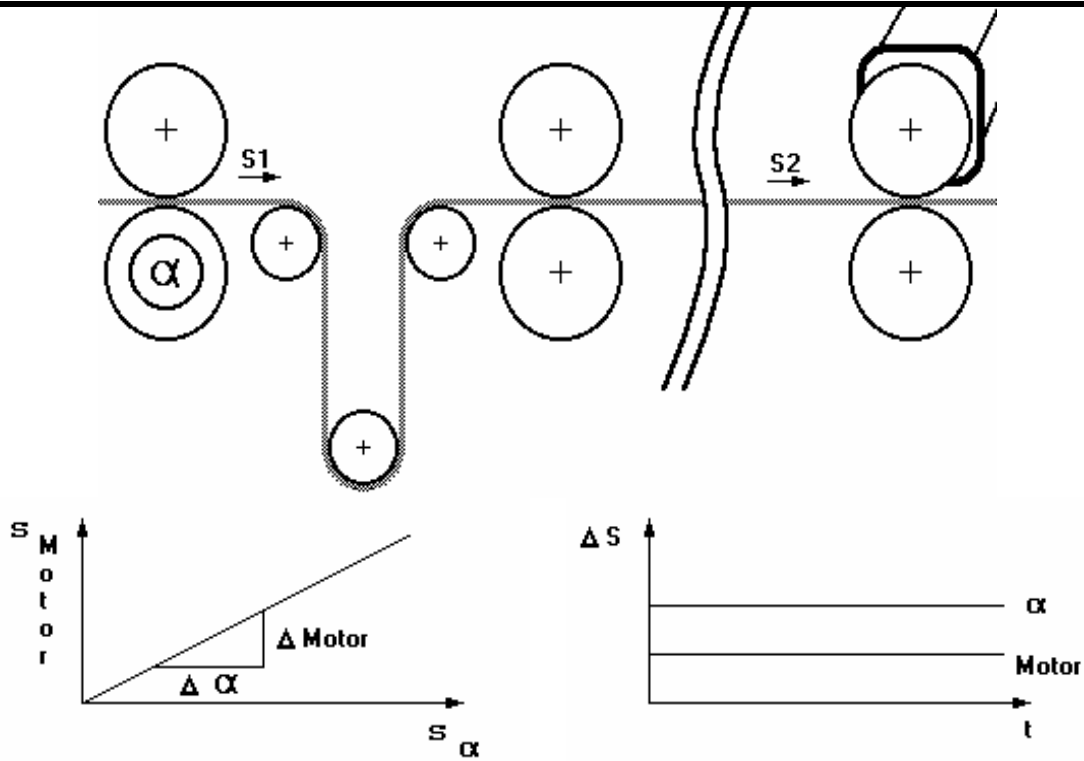
MOVE:
3 If input 11 == 1 then jump MOVE
4 Stop axis, mode = 1 a= 5000 rpm /s
5 End of program, mode 0
    
```

### 4.2 BIAS-application example: Electronic gear box 631

**Explanation:** 1. Move position synchronous to the external incremental encoder ( $\alpha$ ) as long as the input X10.11 is active.

**Filename:** EXP\_004.wbd

**Requirements:** Operation mode 5 ("Position control with BIAS-execution")  
 Wire start input X10.11 and configure with function 3 (BIAS input 11)  
 Configure X40-connector with function 1 (incremental input)  
 X40-resolution depending the external encoder (8192 increments / revolution)



#### Programmlisting

```

* *****
* !!! Attention !!!
* necessary I/O-configurations (BIAS-program definition configuration)
* input X10.8 function 3 (BIAS-input 8)
* Configure X40-connector with function 1 (incremental input)
* X40-resolution depending the external encoder (8192 increments / revolution)
* *****

PROG_START:
0 If 11 == 0 then jump PROG_START
1 Position = 0 INCR
* Additional synchron function "format check" deactivated
2 Start axis
3 Move synchron, gear factor = 0.5


$$\frac{\Delta S_1}{\Delta t} = \frac{\Delta S_2}{\Delta t} ; s_1 = 8192; s_2 = 4096; \text{gear factor} = \frac{s_2}{s_1} = \frac{4096}{8192} = 0,5$$

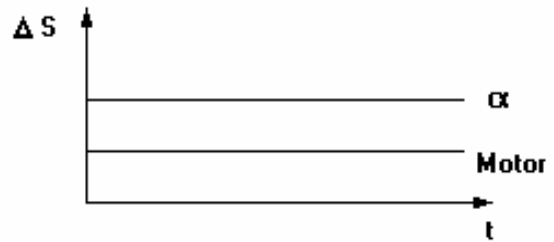
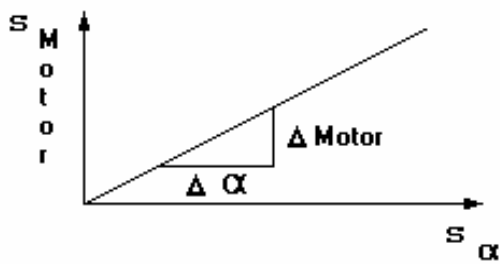
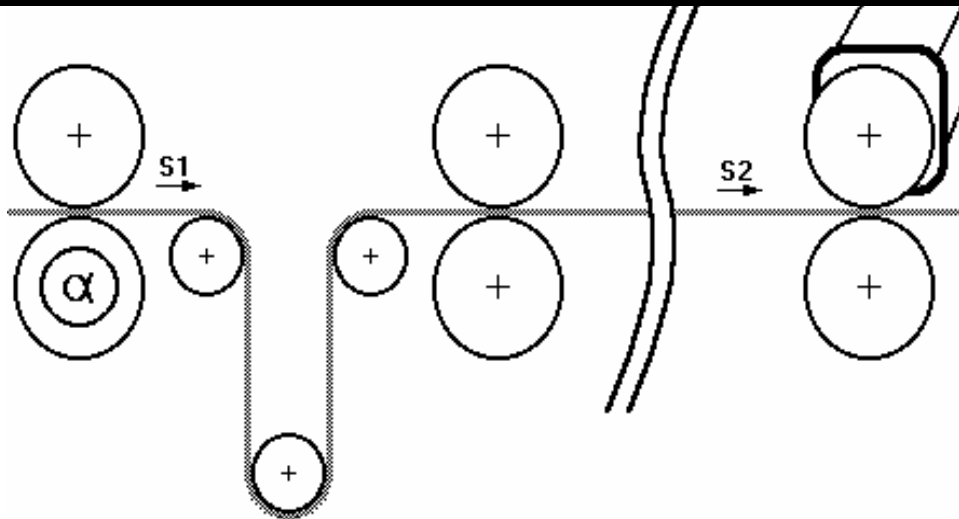

MOVE:
4 If input 8 == 1 then jump MOVE
5 Stop axis, mode = 1 a = 5000 rpm /s
6 end of program, mode 0
  
```

4.3 BIAS-application example: Electronic gear box 637f

**Explanation:** 1. Move position synchronous to the external incremental encoder ( $\alpha$ ) as long as the input X10.11 is active.

**Filename:** EXP\_004f.wbd

**Requirements:** Operation mode 5 ("Position control with BIAS-execution")  
 Wire start input X10.11 and configure with function 3 (BIAS input 11)  
 Configure X40-connector with function 1 (incremental input)  
 X40-resolution depending the external encoder (8192 increments / revolution)





### Programmlisting

```

* *****
* !!! Attention !!!
* necessary I/O-configurations (BIAS-program definition configuration)
* input X10.11 function 3 (BIAS-input 11)
* Configure X40-connector with function 1 (incremental input)
* X40-resolution depending the external encoder (8192 increments / revolution)
* *****

```

#### ProgStart:

```

0 Act. posit. 1 = 0      INKR
1 Act. posit. 2 = 0      INKR
2 Mode 8 = 1
3 Sub-program Para_Sync
4 Load parameter set 1 = [variable [23 ]]          *PointerVa10*
5 If input 11 == 0 then jump ProgStart
6 Start axis
7 Move synchron; axis no. = 1 , target =1 , reference =1
FAHRE:
8 If input 11 == 1 then jump FAHRE
9 Stop axis; mode = 1 ; axis no. = 1
10 Mode 8 = 0
11 End of program, mode = 0

```

```

* Parameter set of type 1 for Synchronism *

```

#### Para\_Sync:

```

12 [Variable 10 ] = 0          *Start angle*
13 [Variable 11 ] = 0          *Delay Time *

```

```

* *****
* Requirement:  $\frac{\Delta s_1}{\Delta t} = \frac{\Delta s_2}{\Delta t}$  ; s1 = 8192; s2= 4096; gear factor =  $\frac{s_2}{s_1} = \frac{4096}{8192} = 0,5$ 

```

```

* Gear ratio = 0,5 *
* At an external encoder resolution of 4096 Incr *

```

```

14 [Variable 12 ] = 4096          *GearNumerat*
15 [Variable 13 ] = 8192          *KopDenomina*

```

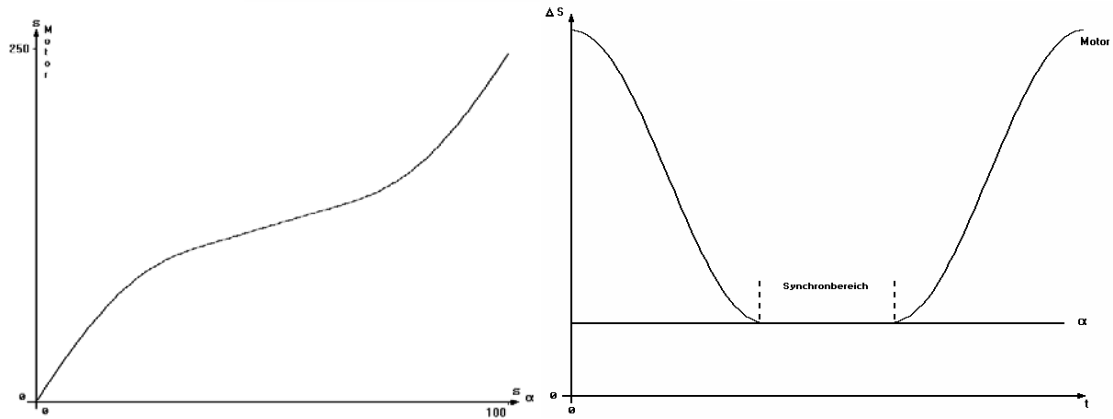
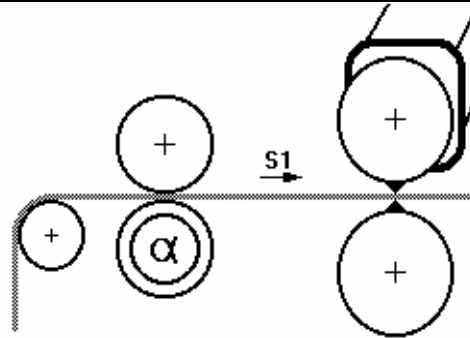
```

16 [Variable 14 ] = 10000          *MoveAcc *
17 [Variable 15 ] = 10000          *MoveCont *
18 [Variable 16 ] = 10000          *MoceDec *
19 [Variable 17 ] = 10000          *MoveWait *
20 [Variable 18 ] = 0              *CAM_Start_o*
21 [Variable 19 ] = 0              *Cam_Nr *
22 [Variable 20 ] = 10000          *ScalNumerat*
23 [Variable 21 ] = 10000          *ScalDenomin*
24 [Variable 23 ] = 10              *PointerVa10*
25 Flag 0 = 0                      *Start_Angle*
26 Flag 1 = 0                      *Start_Direc*
27 Flag 2 = 0                      *Gear_Trig_o*
28 Flag 3 = 0                      *Backstop *
29 Flag 4 = 0                      *Clutch_on *
30 Flag 5 = 0                      *Ramp_Direct*
31 Flag 6 = 0                      *Ramp_Up *
32 Flag 7 = 0                      *Ramp_Down *
33 Flag 8 = 0                      *Ramp_Wait *
34 Flag 9 = 0                      *Move_Linked*
35 Flag 10 = 0                     *Merge_Linke*
36 Flag 11 = 0                     *Caming_on *
37 Flag 12 = 0                     *Stop_Cam *
38 Flag 13 = 0                     *Cam_Backsto*
39 Flag 14 = 0                     *Res1 *
40 Flag 15 = 0                     *Res2 *
41 [Variable 22 ] = flag 0 , number = 16 *Bits , Start_Angle*
42 End of sub-program

```

**5.1 BIAS-application example: Electronic cam profile 635/637**

**Explanation:** 1. Move position synchronous to the external incremental encoder ( $\alpha$ ) as long as the input X10.11 is active.  
**Filename:** EXP\_005.wbd  
**Requirements:** Operation mode 5 ("Position control with BIAS-execution")  
 Wire start input X10.11 and configure with function 3 (BIAS input 11)  
 Configure X40-connector with function 1 (incremental input)  
 X40-resolution depending the external encoder (8192 increments / revolution)



**Programmlisting**

```

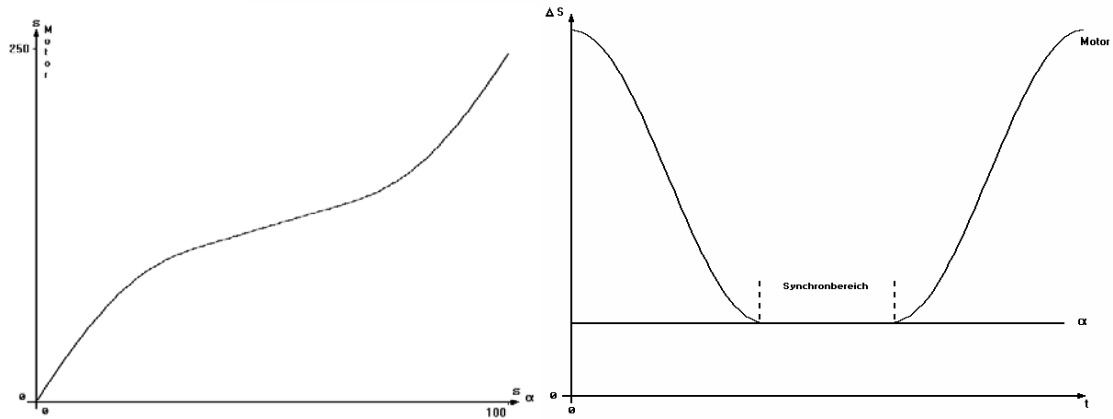
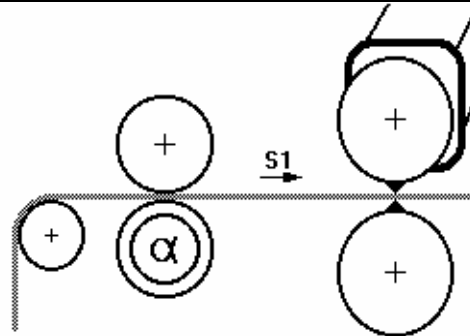
* *****
PROG_START:
0  If 11 == 0 then jump PROG_START
1  Gear factor = 1
2  Start axis
3  Move CAM-profile 0
* *****
* necessary inputs for cam profile calculation, profile 0
* Necessary inputs for the cam profile calculation, profile 0
* synchr. profile punch = 6
* incremental correction = N
* master distance per cycle = 100 mm
* slave distance per cycle = 250 mm
* number of supporting points = 128
* synchronous distance = 25 mm
* *****
MOVE:
4  If input 11 == 1 then jump MOVE
5  Stop axis, mode = 1 a- = 5000 rpm /s
6  End of program, mode 0
    
```

### 5.2 BIAS-application example: Electronic cam profile 631

**Explanation:** 1. Move position synchronous to the external incremental encoder ( $\alpha$ ) as long as the input X10.11 is active.

**Filename:** E631\_005.wbd

**Requirements:** Operation mode 5 ("Position control with BIAS-execution")  
 Wire start input X10.8 and configure with function 3 (BIAS input 8)  
 Configure X40-connector with function 1 (incremental input)  
 X40-resolution depending the external encoder (8192 increments / revolution)



#### Programmlisting

```

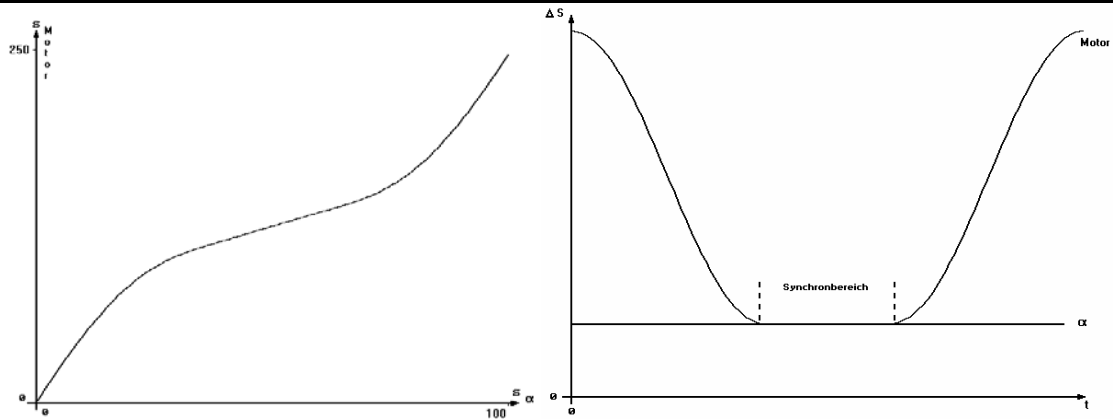
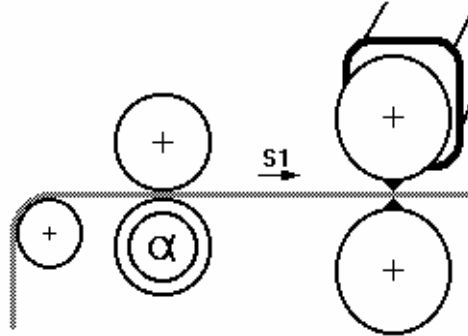
* *****
PROG_START:
0  If  8 == 0 then jump  PROG_START
1  Gear factor = 1
2  Start axis
3  Move CAM-profile 0
* *****
* necessary inputs for cam profile calculation, profile 0
* Necessary inputs for the cam profile calculation, profile 0
* synchr. profile  punch    = 6
* incremental correction   = N
* master distance per cycle = 100 mm
* slave distance per cycle  = 250 mm
* number of supporting points = 128
* synchronous distance     = 25 mm
* *****
MOVE:
4  If input  8 == 1 then jump MOVE
5  Stop axis, mode = 1  a- = 5000 rpm /s
6  End of program, mode 0
  
```

**5.3 BIAS-application example: Electronic cam profile 637f**

**Explanation:** 1. Move position synchronous to the external incremental encoder ( $\alpha$ ) as long as the input X10.11 is active.

**Filename:** EXP\_005f.wbd

**Requirements:** Operation mode 5 ("Position control with BIAS-execution")  
 Wire start input X10.11 and configure with function 3 (BIAS input 11)  
 Configure X40-connector with function 1 (incremental input)  
 X40-resolution depending the external encoder (8192 increments / revolution)



### Programmlisting

```

* *****
* !!! Attention !!!
* necessary I/O-configurations (BIAS-program definition configuration)
* input X10.11 function 3 (BIAS-input 11)
* Configure X40-connector with function 1 (incremental input)
* X40-resolution depending the external encoder (8192 increments / revolution)
* *****

```

### ProgStart:

```

0 Act. posit. 1 = 0      INKR
1 Act. posit. 2 = 0      INKR
2 Mode 8 = 1
3 Sub-program Para_Sync
4 Load parameter set 1 = [variable [23 ]]          *PointerVa10*
5 If input 11 == 0 then jump ProgStart
6 Start axis
7 Move synchron; axis no. = 1 , target =1 , reference =1
FAHRE:
8 If input 11 == 1 then jump FAHRE
9 Stop axis; mode = 1 ; axis no. = 1
10 Mode 8 = 0
11 End of program, mode = 0

```

```

*
*      Parameter set of type 1 for CAM profile 0
*
* *****

```

### Para\_Sync:

```

12 [Variable 10 ] = 0          *Start angle*
13 [Variable 11 ] = 0          *Delay Time *
* *****
*      Gear ratio = 0,5
*      At an external encoder resolution of 4096 Incr
* *****
14 [Variable 12 ] = 4096          *GearNumerat*
15 [Variable 13 ] = 4096          *KopDenomina*
16 [Variable 14 ] = 10000        *MoveAcc *
17 [Variable 15 ] = 10000        *MoveCont *
18 [Variable 16 ] = 10000        *MoceDec *
19 [Variable 17 ] = 10000        *MoveWait *
20 [Variable 18 ] = 0            *CAM_Start_o*
21 [Variable 19 ] = 0            *Cam_Nr *
22 [Variable 20 ] = 10000        *ScalNumerat*
23 [Variable 21 ] = 10000        *ScalDenomin*
24 [Variable 23 ] = 10          *PointerVa10*
25 Flag 0 = 0                    *Start_Angle*
26 Flag 1 = 0                    *Start_Direc*
27 Flag 2 = 0                    *Gear_Trig_o*
28 Flag 3 = 0                    *Backstop *
29 Flag 4 = 0                    *Clutch_on *
30 Flag 5 = 0                    *Ramp_Direct*
31 Flag 6 = 0                    *Ramp_Up *
32 Flag 7 = 0                    *Ramp_Down *
33 Flag 8 = 0                    *Ramp_Wait *
34 Flag 9 = 0                    *Move_Linked*
35 Flag 10 = 0                   *Merge_Linke*
36 Flag 11 = 1                   *Caming_on *
37 Flag 12 = 0                   *Stop_Cam *
38 Flag 13 = 0                   *Cam_Backsto*
39 Flag 14 = 0                   *Res1 *
40 Flag 15 = 0                   *Res2 *
41 [Variable 22 ] = flag 0 , number = 16 *Bits , Start_Angle*
42 End of sub-program

```



Position = const.	[Variable X] = position	BIAS-execution pointer	[Variable X] = flag Y	Profile value = [variable X]	Save table	PLC-program
This command is only permitted in the BIAS- task	This command is only permitted in the BIAS, PLC and MATH-Task	This command is only permitted in the PLC and MATH-Task	This command is only permitted in the BIAS and PLC -Task	This command is only permitted in the MATH-Task	This command is only permitted in the MATH-Task	This command is only permitted in the BIAS and MATH-Task

	0	1	2	3	4	5	6	7	8	9	A	B
0	Move_position	Move_position + parameter	Position = const.	Position = [variable X]	[Variable X] = position	NOP	Flag_X = const.	If_input X ? const.	[Variable X] = const.	Mathematic program	Table [[variable X]] = const.	[D_Variable X] = [D_Variable Y] + [D_Variable Z]
1	Move_incremental position	Move_incremental position + parameter	Speed = const.	Speed = [variable X]	[Variable X] = speed	End_of program	If_flag X ? const.	If_output X ? const.	If [variable X] ? const.	Profile_initialization = const.	Table [[variable X]] = [Y_Variable Z]	[D_Variable X] = [D_Variable Y] - [D_Variable Z]
2	Move_datum	Move_datum + parameter	Acceleration = const.	Acceleration = [Variable X]	[Variable X] = acceleration	Sub-program	Flag_X = flag Y	Output_X = const.	[Variable X] = [variable Y] + const.	Profile_cycle_length = [variable X]	[X_Variable Y] = Table [[variable Z]]	[D_Variable X] = [D_Variable Y] * [D_Variable Z]
3	Move_infinite positive	Move_infinite positive + parameter	Deceleration = const.	Deceleration = [variable X]	[Variable X] = deceleration	End_of Sub-program	Flag_X = input Y	Output_X = flag Y	[Variable X] = [variable Y] - const.	[Variable X] = profile_value	[W_Variable X] = [Y_Variable Z]	[D_Variable X] = [D_Variable Y] / [D_Variable Z]
4	Move_infinite negative	Move_infinite negative + parameter	Gear factor = const.	Gear_factor = [Variable X]	[Variable X] = gear factor	PLC-program	Flag_X = output Y	_____	[Variable X] = [variable Y] * const.	Profile_value = [variable X]	[X_Variable Y] = const.	If [D_Variable X] ? [D_Variable Y]
5	Move_synchron	Move_synchron + parameter	"Position_reached" window = const.	"Position_reached" window = [variable X]	[Variable X] = block number	Jump_const.	Flag_X = flag Y & flag Z	_____	[Variable X] = [variable Y] / const.	_____	[Variable X] = const.	[D_Variable X] = SIN ([D_Variable Y])
6	Move_CAM profile	Move_analogue value + integrator	Remaining_position = const.	Remaining_position = [variable X]	[Variable X] = actual position Y	Jump [variable X]	Flag_X = flag Y   flag Z	_____	[Variable X] = flag Y	_____	[Variable X] = [variable Y]	[D_Variable X] = COS ([D_Variable Y])
7	Synchronous settings 1	Move_speed + integrator	Ramp_filter = const. [variable X]	Maximal_current = [variable X]	[Variable X] = analogue input Y	BIAS-Execution pointer = const.	Flag_X = flag Y ^ flag Z	_____	[Variable X] = [variable Y].bit Z number	Save table	[Variable X] = [variable Y]	[D_Variable X] = SQRT ([D_Variable Y])
8	Synchronous settings 2	_____	Actual_position X = const.	Actual_position X = [variable Y]	[Variable X] = latch position Y	Wait for "position reached"	Flag_X = ! flag Y	IBT_mask_number = const.	[Variable X] = [variable Y]	_____	[Variable X] = [variable Y] ? [variable Z]	_____
9	Move_PID: speed	_____	If_actual_position X ? const.	Analogue_output X = [variable Y]	[Variable X] = actual speed Y	Wait time = const.	Flag_X = status Y	IBT_notification number = const.	If [variable X] ? [variable Y]	_____	[Variable X] = [variable Y] ? const.	_____
A	Move_PID: torque	Cycle_length = const.	If_actual_position X ? [variable Y]	PID_scaling	[Variable X] = latch status Y	Wait time = [variable X]	If_status X ? const.	CAN_Command = [variable X]	[Variable X] = [variable Y] + [variable Z]	_____	_____	_____
B	Set_point [axis no.] = const.	Cycle_length = [variable X]	Sensor_window = const.	Sensor_window = [variable X]	[Variable X] = position Y; axis no.	BIAS-execution pointer = [variable X]	Modus_X = const.	IBT_data_transfer	[Variable X] = [variable Y] - [variable Z]	_____	_____	_____
C	Set_point [axis no.] = [variable X]	Load_parameter set X = [variable Y]	Sensor_position = const.	Sensor_position = [variable X]	[Variable X] = value Y	Jump [var.X]; length = const.; from	Flag_X = [variable Y]	CAN2_Command = [variable X]	[Variable X] = [variable Y] * [variable Z]	_____	_____	_____
D	Move_relative	_____	Sensor adjustment 1 = const.	Sensor adjustment 1 = [variable X]	[Variable X] = axis status; axis no. Y	Execute X commands	[Variable X].bit[Y] = const.	_____	[Variable X] = [variable Y] / [variable Z]	_____	_____	_____
E	Start_axis	_____	Sensor adjustment 2 = const.	Sensor adjustment 2 = [variable X]	_____	_____	If [var. X].bit Y == const. then jump	_____	[Teachvariable X] = [variable Y]	_____	_____	_____
F	Stop_axis	Stop_axis ± parameter	Update_parameter	PID_parameter	_____	Virtual_program	Axis_state, axis no. X, bit Y = const., [flag Z]	_____	[Variable X] = [teachvariable Y]	_____	_____	_____

- Command group "Move commands"
- Command group "Parameter commands"
- Command group "Variable commands"
- Command group "Flag commands"
- Command group "Conditional jump commands"

- Command group "Program control commands"
- Command group "Mathematic commands"
- Command group "Output commands"
- Command group "CAN-Commands"
- Command group "637f commands"

## 3.1 BIAS-Command group “Move commands“

Command name	No.	Applicable in program task...		
		BIAS	PLC	MATH
<b>Move commands</b>				
Move <a href="#">incremental position</a>	01	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">PID, speed</a>	09	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">PID, torque</a>	0A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">position</a>	00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">datum</a>	02	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">relative</a>	0D	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">synchron</a>	05	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">CAM profile</a>	06	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">infinite negative</a>	04	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">infinite positive</a>	03	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Start <a href="#">axis</a>	0E	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Stop <a href="#">axis</a>	0F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>Move commands with parameter</b>				
		<b>BIAS</b>	<b>PLC</b>	<b>MATH</b>
Move <a href="#">speed + integrator</a>	17	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">incremental position + parameter</a>	11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">position + parameter</a>	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">datum + parameter</a>	12	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">synchron + parameter</a>	15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">Analogue value + integrator</a>	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">infinite negative + parameter</a>	14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move <a href="#">infinite positive + parameter</a>	13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Stop <a href="#">axis + parameter</a>	1F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### 3.2 BIAS-Command group „Parameter commands“

Command name	No.	Applicable in program task...		
		BIAS	PLC	MATH
<b>Parameter = Constant</b>				
Acceleration <a href="#">= const.</a>	22	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Speed <a href="#">= const.</a>	21	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Actual <a href="#">position X = const.</a>	28	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gear <a href="#">factor = const.</a>	24	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
“Position <a href="#">reached“ window = const.</a>	25	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ramp <a href="#">filter = const.; [variable X]</a>	27	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Remaining <a href="#">position = const.</a>	26	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sensor <a href="#">adjustment 1 = const.</a>	2D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sensor <a href="#">adjustment 2 = const.</a>	2E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sensor <a href="#">window = const.</a>	2B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sensor <a href="#">position = const.</a>	2C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Set <a href="#">point [axis no] = const.</a>	0B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Synchronous <a href="#">settings 1; modus X ....</a>	07	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Synchronous <a href="#">settings 2; linear X ....</a>	08	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cycle <a href="#">length = const.</a>	1A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Deceleration <a href="#">= const.</a>	23	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Position <a href="#">= const.</a>	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Update <a href="#">parameter</a>	2F	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>Parameter = Variable</b>				
Analogue <a href="#">output X = [variable Y]</a>	39	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Acceleration <a href="#">= [variable X]</a>	32	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Speed <a href="#">= [variable X]</a>	31	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Actual <a href="#">position X = [variable Y]</a>	38	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gear <a href="#">factor = [variable X]</a>	34	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Load <a href="#">parameter set X = [variable [Y]]</a>	1C	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Maximal <a href="#">current = [variable X]</a>	37	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PID <a href="#">parameter</a>	3F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PID <a href="#">scaling</a>	3A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
“Position <a href="#">reached“ window = [variable X]</a>	35	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Remaining <a href="#">position = [variable X]</a>	36	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sensor <a href="#">adjustment 1 = [variable X]</a>	3D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sensor <a href="#">adjustment 2 = [variable X]</a>	3E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sensor <a href="#">window; modus = [variable X]</a>	3B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sensor <a href="#">position = [variable X]</a>	3C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Set <a href="#">point [axis no] = [variable X]</a>	0C	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cycle <a href="#">length = [variable X]</a>	1B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Deceleration <a href="#">= [variable X]</a>	33	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Position <a href="#">= [variable X]</a>	30	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



Variable = Parameter

[Variable <a href="#">X</a> ] = <a href="#">analogue input Y</a>	47	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">axis status, axis no. Y</a>	4D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">acceleration</a>	42	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">actual speed Y</a>	49	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">speed</a>	41	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">actual position Y</a>	46	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">gear factor</a>	44	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">latch position Y</a>	48	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">latch state Y</a>	4A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">position Y</a>	4B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">set number</a>	45	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">deceleration</a>	43	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">position</a>	40	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <a href="#">X</a> ] = <a href="#">value Y</a>	4C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

### 3.3 BIAS Command group “Variable commands”

Command name	No.	Applicable in program task...		
		BIAS	PLC	MATH
<b>Variables commands</b>				
[X_Variable <u>Y</u> ] = const.	A4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Longinteger-variables commands</b>				
[Variable <u>X</u> ] = [variable Y].bit Z, number	87	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = [variable Y] - [variable Z]	8B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = [variable Y] - const.	83	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable. <u>X</u> ] = [variable Y] * [variable Z]	8C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = [variable Y] * const.	84	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = [variable Y] / [variable Z]	8D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = [variable Y] / const.	85	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = [variable Y] + [Var. Z]	8A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = [variable Y] + const.	82	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable [ <u>X</u> ]] = [variable Y]	A6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable [ <u>X</u> ]] = const.	A5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = [variable [Y]]	A7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = [variable Y]	88	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = flag Y, number Z	86	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
[Variable <u>X</u> ].bit Y = const.	6D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = const.	80	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = [teachvariable Y]	8F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Teachvariable <u>X</u> ] = [variable Y]	8E	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Variable <u>X</u> ] = [variable Y] ? [variable Z]	A8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
[Variable <u>X</u> ] = [variable Y] ? const	A9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

### 3.4 BIAS Command group “Flag commands”

Command name	No.	Applicable in program task...		
		BIAS	PLC	MATH
<b>Flag commands</b>				
Axis <u>state</u> , axis no. X, bit Y = const., [flag Z]	6F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Flag <u>X</u> = ! flag Y	68	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Flag <u>X</u> = [variable Y], number Z	6C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Flag <u>X</u> = output Y	64	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Flag <u>X</u> = input Y	63	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Flag <u>X</u> = flag Y & flag Z	65	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Flag <u>X</u> = flag Y ^ flag Z	67	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Flag <u>X</u> = flag Y   flag Z	66	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Flag <u>X</u> = flag Y	62	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Flag <u>X</u> = state Y	69	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Flag <u>X</u> = const.	60	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## 3.5 BIAS Command group “Conditional jump commands”

Command name	No.	Applicable in program task...		
		BIAS	PLC	MATH
<b>Conditional jump commands</b>				
If <a href="#">[D Var. X] ? [D Var. Y] then jump</a>	B4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
If <a href="#">[var. X] ? [var. Y] then jump</a>	89	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
If <a href="#">[var. X] ? const. then jump</a>	81	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
If <a href="#">output X ? const. then jump</a>	71	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
If <a href="#">input X ? const. then jump</a>	70	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
If <a href="#">actual position X ? const. then jump</a>	29	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
If <a href="#">actual position X [var. Y] then jump</a>	2A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
If <a href="#">flag X ? then jump</a>	61	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
If <a href="#">state X ? const. then jump</a>	6A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
If <a href="#">[variable X].bit Y == const.</a>	6E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## 3.6 BIAS Commands group “Program control commands”

Command name	No.	Applicable in program task...		
		BIAS	PLC	MATH
<b>Program control commands</b>				
Execute <a href="#">X commands</a>	5D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
BIAS- <a href="#">execution pointer = [variable X]</a>	5B	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
BIAS- <a href="#">execution pointer</a>	57	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Mathematic <a href="#">program</a>	90	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modus <a href="#">X = const.</a>	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"/>
End <a href="#">of program, modus =</a>	51	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Jump <a href="#">[variable X]</a>	56	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Jump <a href="#">[variable X]: length; const.</a>	5C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Jump <a href="#">const.</a>	55	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PLC- <a href="#">program</a>	54	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Sub <a href="#">program</a>	52	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
End <a href="#">of sub program</a>	53	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Virtual <a href="#">program</a>	5F	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Wait <a href="#">for "position reached"</a>	58	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Wait <a href="#">time = [variable X]</a>	5A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Wait <a href="#">time = const.</a>	59	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### 3.7 BIAS Command group “Mathematic commands“

Command name	No.	Applicable in program task...		
		BIAS	PLC	MATH
<b>Profile Commands</b>				
[Variable X] = profile value Y, Z	93	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[X_Variable Y] = const.	A4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[X_Variable Y] = table ([variable Z])	A2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Profile initialization = const.	91	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Profile cycle length = [variable X]	92	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Profile value Y, Z = [variable X]	94	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Table ([Variable X]) = [Y_Variable Z]	A1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Table ([Variable X]) = const.	A0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Save table	97	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>floating point Variable commands</b>				
[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"/>
<b>Variable type change command</b>				
[W_Variable X] = [Y_Variable Z]	A3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

### 3.8 BIAS Command group “Output commands“

Command name	No.	Applicable in program task...		
		BIAS	PLC	MATH
<b>Output commands</b>				
Output X = flag Y	73	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Output X = const.	72	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Analogue output X = [variable Y]	39	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## 3.9 BIAS Command group “CAN-Commands“

Command name	No.	Applicable in program task...		
		BIAS	PLC	MATH
<b>CAN commands</b>				
<a href="#">CAN-Command = [variable X]</a>	7A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<a href="#">CAN2 Command = [variable X]</a>	7C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<a href="#">IBT-data transfer</a>	7B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<a href="#">IBT-mask number = X, LED status [var. Y]</a>	78	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">IBT-notification number = X, LED status [var. Y]</a>	79	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## 3.10 BIAS Command group “637f V8.xx commands“

Command name	No.	Applicable in program task...		
		BIAS	PLC	MATH
<b>637f V8.xx- Commands</b>				
<a href="#">Axis state, axis no. X, bit Y = const, [flag Z]</a>	6F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<a href="#">CAN2 Command = [variable X]</a>	7C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<a href="#">Execute X commands</a>	5D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<a href="#">Move relative, axis no. X, target Y</a>	0D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Load parameter set X = [variable [Y]]</a>	1C	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<a href="#">Set point [axis no.] = const.</a>	0B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<a href="#">Set point [axis no.] = [variable X]</a>	0C	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<a href="#">Jump [var.[X ]]; length=Y ; from Z</a>	5C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<a href="#">[Variable X] = axis status, axis no.Y</a>	4D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">[Variable X].bit Y = const.</a>	6D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">[Variable X] = [variable Y].bit Z, number</a>	87	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">If [variable X].bit Y == const.</a>	6E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">[Variable X] = [variable Y] ? [variable Z]</a>	A8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">[Variable X] = [variable Y] ? const.</a>	A9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## 4 - 00 BIAS-Command "Move position"

Move position; axis no. = X , target = Y

### Explanation:

The command "Move position" starts the axis on the target position defined last.

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
631/635/637/ 637+:	none	-	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
637f with Multi-Axis-Function: (Mode 8 = 1)	X: Axis number Y: Target	1 - 3 1 = Reference for position controller 2 = Reference for CAN2-PDO3 4 = Reference for CAN2-PDO4 8 = Reference for X40 output The parallel connection of the setpoint targets is possible by addition of the function coding.	

To the generation of move curve always apply the last adjusted parameters for: Position<sub>1</sub>, Acceleration<sub>1</sub>, Speed<sub>1</sub>, Deceleration<sub>1</sub>, Ramp [filter](#)

When switching on of the controller, these parameters are pre-loaded with the settings from EASYRIDER menu "Start-up position controller".

When using the Multi-Axis-Function with the controller type 637f the parameters must be pre-loaded over the command "Load [parameterset 0 = \[variable\[X\]\]](#)".

### Example:

0 Position = 1000 INCR  
1 Start axis  
**2 Move position; axis no. = 0 , target = 0**  
3 Wait for "position reached"  
4....

### Result:

The axis moves 1000 increments on the absolute position. BIAS processing is continued with the set 4 after the [position reached](#).

### Note:

- Before the start a valid starting identification is waited for:
  - a.) preceding the command "Start [axis](#)"
  - b.) Low-High impulse of the Start input X10.11 (X10.9 with the 631)
    - **635/ 637/ 637+/ 637f**  
When using the start input X10.11 the function 0 for the input X10.11 (Start input) is to be selected in the "input configuration".
    - **631**  
When using the start input X10.9 the function 3 for the input X10.9 (Start input) is to be selected in the configuration.
- The momentary set position can be determined with the BIAS command [\[Variable X\] = position 1.](#)

### Command coding:

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

Parameters: D1 = Axis number: 1...3  
D2 = Target: 1...15

## 4 - 01 BIAS-Command "Move incremental position"

Move incremental position; axis no. = X , target = Y

### Explanation:

The command "Move incremental position" starts the axis on the position, which results from the sum of the current target position plus the relative distance defined in the command "Position= ".

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
631/635/637/637+:	none	-	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
637f with Multi-Axis-Function: (Mode 8 = 1)	X: Axis number Y: Target	1 - 3  1 = Reference for position controller 2 = Reference for CAN2-PDO3 4 = Reference for CAN2-PDO4 8 = Reference for X40 output The parallel connection of the setpoint targets is possible by addition of the function coding.	

To the generation of move curve always apply the last adjusted parameters for: Position, Acceleration, Speed, Deceleration, Ramp [filter](#)

When switching on of the controller, these parameters are pre-loaded with the settings from EASYRIDER menue "Start-up position controller".

When using the Multi-Axis-Function with the controller type 637f the parameters must be pre-loaded over the command "Load [parameter set 0 = \[Variable\[X\]\]](#)".

### Example:

0 Position = 1000 INCR

1 Start Axis

2 Move incremental position; axis no.=0, target=0

3 Wait for "position reached"

4....

### Result:

The axis moves 1000 increments.

BIAS processing is continued with the set 4 after the [position reached](#) command.

### Note:

- Before the start a valid starting identification is waited for:
  - a.) preceding the command "Start [axis](#)"
  - b.) Low-High impulse of the Start input X10.11 (X10.9 with the 631)
    - **635/ 637/ 637+/ 637f**  
When using the start input X10.11 the function 0 for the input X10.11 (Start input) is to be selected in the configuration.
    - **631**  
When using the start input X10.9 the function 3 for the input X10.9 (Start input) is to be selected in the configuration.
- The target position can be determined with the BIAS command [Variable [X](#)] = [position 2](#) .

### Command coding:

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

Parameters: D1 = Axis number: 1...3  
D2 = Target: 1...15

4 - 02 BIAS-Command "Move datum"

Move datum, mode= X , [variable Y ]

**Explanation:**

The command "Move datum" starts the reference move (homing) of the axis corresponding to the specified reference mode.

[631 load example](#)

[635/637 load example](#)

[637+ load example](#)

Drive type:	Parameters:	Permissible inputs:
631/635/637/ 637+:	X: Mode	Mode: <a href="#">0...23</a>
637f:	X: Mode Y: Variable	<a href="#">Mode: 0...23, 24 oder 128, 192</a> 0 -255

- BIASprogram
- PLC program
- Mathematics

To the generation of move curve always apply the last adjusted parameters for: Position, Acceleration, Speed, Deceleration, Ramp [filter](#).

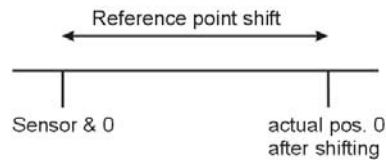
After switching the drive on these parameters are initialized with the values of the tuning position loop menu.

**Example:**

- 0 Position = 4000 INKR
- 1 Start axis
- 2 Move datum, mode 16
- 3 Wait for "position reached"
- 4 ....

**Result:**

The axis moves in the positive direction to the Low-High slope of the reference sensor and then the zero point of the encoder.



Then it moves +4000 increments from this position and then sets the actual position to zero.



**Note:**

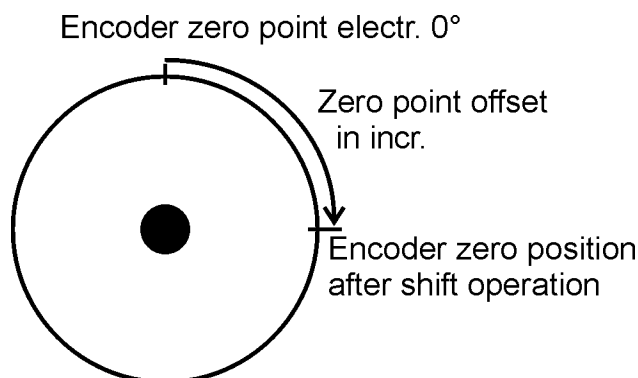
- The encoder could be resolver or HIPERFACE®  
In position control with actual position 2 (X40) the 0-position of the external encoder is used.
- Before starting a valid start mark has to occur:
  - a.) the command "Start axis" before the move command or
  - b.) low-high impulse of the start input X10.11 (X10.9 at 631)
    - **635/ 637/ 637+/637f**  
When using the start input X10.11 the function 0 (start input) has to be selected for the input X10.11 in the configuration.
    - **631**  
When using the start input X10.9 the function 3 (start input) has to be selected for the input X10.9 in the configuration.
- **635/ 637/ 637+/637f**  
When using the reference sensor input X10.24 the function 1 (reference sensor) has to be selected for the input X10.24 in the configuration.
- **631**  
When using the reference sensor for one of the inputs (X10.8, X10.9 or X10.10) the function 1 (reference sensor) has to be selected in the configuration.
- **631/ 635/ 637**  
The maximum permissible speed depends on the resolver resolution configuration.  
With a high resolver resolution only 4000 rpm is allowed.
- **637f**  
When the drive 637f is mounted by the option board C28 the input X120.4 can configured as reference sensor. The reference sensor input configuration at the X10.24 clamp has no function.

**Mode = 24**, the reference bit (and the output initialized) are set immediately.

**Mode = 128**, the value of the specified variable is loaded.

**Mode =192**, the value of the specified variable is used as encoder zero point shift.

**Attention:** With this mode the axis do **not** start. The offset is calculated only after a new Move datum command with the mode encoder zero point.



**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>02</b>	<b>X</b>	<b>Y</b>	00	00	00	00	00

**Parameters:** D1 = Reference mode: 0...24, 128, 192  
D2 = Variable 0-255

## 4 - 03 BIAS-Command "Move infinite positive"

Move infinite positive; axis no. = X, target = Y

### Explanation:

The command "Move infinite positive" starts the movement of the axis in a positive direction at the currently defined speed.

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
631/635/637/ 637+:	None	-	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
637f With Multi-Axis- function: (Mode 8 = 1)	X: Axis number  Y: Target	1 - 3 1 = reference for position controller 2 = reference for CAN2-PDO3 4 = reference for CAN2-PDO4 8 = reference for X40-output At using more reference value target the function codes are able to add.	

The setpoint calculation always uses the last defined parameters of: "Position", Acceleration, Speed, Deceleration, Ramp [filter](#)  
After switching the drive on these parameters are initialized with the values of the tuning position loop menu.  
At using the multi axis functions by the drive 637f the parameters must loaded with the command "Load [parameter set 0 = \[variable \[X\]\]](#)".

### Example:

0 Speed = 1000 rpm

1 Start axis

2 Move infinite positive, axis no. = 0, target = 0

3 Wait time 1000 ms

4 Stop axis; mode 1

5 ....

### Result:

The axis moves 1s in positive direction.

### Note:

- The command "Start [axis](#)" before the move command
  - low-high impulse of the start input X10.11 (X10.9 on 631)
    - 635/ 637/ 637+/ 637f**  
When using the start input X10.11 the function 0 (start input) has to be selected in the configuration
    - 631**  
When using the start input X10.9 the function 3 (start input) has to be selected in the configuration.
- Speed changes can be carried out during the run with the command "Update [parameter](#)".

### Command coding:

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

Parameters: D1 = Axis number: 1...3  
D2 = Target: 1...15

## 4 - 04 BIAS-Command "Move infinite negative"

Move infinite negative; axis no. = X, target = Y

### Explanation:

The command "Move infinite negative" starts the movement of the axis in a negative direction at the currently defined speed.

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
631/635/637/637+:	none	-	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
637f with Multi-Axis-Function: (Mode 8 = 1)	X: Axis number Y: Target	1 - 3  1 = reference for position controller 2 = reference for CAN2-PDO3 4 = reference for CAN2-PDO4 8 = reference for X40 output The parallel connection of the setpoint targets is possible by addition of the function coding.	

To the generation of move curve always apply the last adjusted parameters for: Acceleration, Speed, Deceleration, Ramp [filter](#)

When switching on of the controller, these parameters are pre-loaded with the settings from EASYRIDER menu "Start-up position controller".

When using the Multi-Axis-Function with the controller type 637f the parameters must be pre-loaded over the command "Load [parameterset 0 = \[variable\[X\]\]"](#).

### Example:

0 Speed = 1000 rpm  
1 Start axis  
**2 Move infinite negative, axis no = 0, target = 0**  
3 Wait time 1000 ms  
4 Stop axis ; mode 1  
5 ....

### Result::

The axis moves 1s in the negative direction.

### Note:

- Before the start a valid starting identification is waited for:
  - a.) preceding the command "Start [axis](#)"
  - b.) Low-High impulse of the Start input X10.11 (X10.9 with the 631)
    - **635/ 637/ 637+/ 637f**  
When using the start input X10.11 the function 0 for the input X10.11 (Start input) is to be selected in the "input configuration".
    - **631**  
When using the start input X10.9 the function 3 for the input X10.9 (Start input) is to be selected in the configuration.
- Speed changes can be carried out during the run with the command "Update [parameter](#)".

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>04</b>	<b>X</b>	<b>Y</b>	00	00	00	00	00

Parameters: D1 = axis number: 1...3  
D2 = Target: 1...15

## 4 - 05 BIAS-Command "Move synchron"

Move synchron; axis no = X , target = Y , reference = Z

### Explanation:

The command "Move synchron" starts the movement of the axis with the adjusted gear factor synchronous to the master increments.

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
631/635/637/637+:	none	-	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
637f with Multi-Axis-function: (Mode 8 = 1)	<p><b>X: Axis number</b></p> <p><b>Y: Target</b></p> <p><b>Z: Reference</b></p>	<p>1 - 3</p> <p>1 = Setpoint into position controller</p> <p>2 = Setpoint on CAN2-PDO3</p> <p>4 = Setpoint on CAN2-PDO4</p> <p>8 = Setpoint on X40-output</p> <p>The parallel switch of targets is possible by addition of the function codes.</p> <p>1 = X40 actual position 2</p> <p>2 = Actual position 3 (CAN2-SSI-PDO1)</p> <p>4 = Actual position 4 CAN2-PDO3</p> <p>8 = Actual position 5 CAN2-PDO4</p> <p>16 = Setpoint position of Axis 1</p> <p>32 = Setpoint position of Axis 2</p> <p>64 = Setpoint position of Axis 3</p> <p>The sum of the reference value is allowed.</p> <p><b>Attention:</b> It is not permitted to select the setpoint of the specified axis as reference.</p>	

At using the multi-axis functions at the drive 637f the parameter must load with the command "Load [parameterset X = \[variable\[Y\]\]](#)".

Further [information about the synchronous function of the 630 Series](#)

### Example 635/637/637+/637f:

0 Cycle length = 0 INKR

1 Gear factor = 1.0

**2 Move synchron; axis no. = 0, target = 0 , setpoint = 0**

3 If input 11 == 1 then jump 3

4 Stop axis ; mode 1

5 ...

### Example 631:

0 Cycle length = 0 INKR

1 Gear factor = 1.0

**2 Move synchron; axis no. = 0, target = 0 , setpoint = 0**

3 If input 9 == 1 then jump 3

4 Stop axis ; mode 1

5 ...

### Note:

- Before starting a valid start mark has to occur:
  - a. The command "Start [axis](#)" before the move command.  
optional : Sensor; [mode=128](#)
  - b. low-high impulse of the start input X10.11 (X10.9 on 631)
    - **635/ 637/ 637+/637f**  
When using the start input X10.11 the function 0 (start input) has to be selected for the input X10.11 in the configuration.

### Result:

After the Low-High slope of the start input 11 the axis runs synchronously (per master increment = gear factor increments), until the start input drops.

After the Low-High slope of the start input 9 the axis runs synchronously (per master increment = gear factor increments), until the start input drops.

- **631**  
When using the start input X10.9 the function 3 (start input) has to be selected for the input X10.9 in the configuration.
- X40 has to be configured for encoder input in the counter configuration.
- The gear ratio can change with the command "Gear factor =" during a synchronous move command
- The command "Cycle length =" placed before the move command defines the length of the format.
- The setpoints are specified as position differences.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>05</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	00	00	00	00

**Parameters:** D1 = Axis number: 1...3  
 D2 = Target: 1...15  
 D3 = Reference 1...127

## 4 -06 BIAS-Command "Move CAM profile"

### Move CAM-profile X

#### Explanation:

The command "Move CAM profile" starts the movement of the axis with the specified profile number synchronous to the master increments.

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
All	X: Profile number	0-15	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

Further [information about the synchronous profile function of the 630 Series](#)

#### Example 635/637/ 637+/ 637f:

0 Gear factor = 1.0

##### 1 Move CAM-profile 0

2 If input 11 == 1 then jump 2

3 Stop axis; mode 1

4 .....

#### Result:

The profile calculated in the CAM profile 0 is processed until input 11 drops.

#### Example 631:

0 Gear factor = 1.0

##### 1 Move CAM-profile 0

2 If input 9 == 1 then jump 2

3 Stop axis; mode 1

4 .....

#### Result:

The profile calculated in the CAM profile 0 is processed until input 9 drops.

#### Note:

- Before starting a valid start command has to be encountered by either:
  - The command "Start [axis](#)" before the move command low-high impulse of the start input X10.11 (X10.9 on 631)
  - a. **635/ 637/637+/637f:**  
When using the start input X10.11 the function 0 (start input) has to be selected in the configuration.
  - b. **631**  
When using the start input X10.9 the function 3 (start input) has to be selected in the configuration.
- X40 has to be configured for encoder input in the counter configuration..
- Before enabling the command the corresponding profile has to be calculated ("Program", "Manage CAM profiles)
- The command "Gear [factor =](#)" placed before the move command defines a scaling of the master cycle length (resulting master cycle length = calculated master cycle length \* Gear factor)

#### Command coding:

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

**Parameters:** D1 = Profile number 0...15

## 4 - 07 BIAS-Command “Synchronous settings 1“

Synchronous settings 1; mode=X; offset=[var.Y ]; start offset=[var.Z ]

### Explanation:

With the command "Synchronous settings 1" it is possible to influence a synchronous move command in progress by setting a mode and/or the definition of an offset in variable X.

[Load example](#)

Drive type:	Parameters:	Permissible inputs:	
All	X: <a href="#">Modus</a> Y: <a href="#">Offsetvariable</a> Z: <a href="#">Startoffsetvar.</a>	0...255 0...255 0...255	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

Further [information about the synchronous functions of the 630 Series](#)

### Example:

0 Synchronous settings 1.; mode=0; offset=[var.0]; start offset=[var.0]

### Result:

### Note:

- The following functions are possible with the parameter **mode**:
  - No function
  - Brake to 0 speed within half of the cycle length.
  - Move to synchronous speed within half of the cycle length.
  - Reserved (customer depending special function)
- The reaction to offset modifications can be influenced by the following combinations:

**Mode + 32** Offset value will be loaded immediately (not only on the format trigger point).

**Mode + 64** Offset value will be loaded as offset 2 (profile shifting on the master axis).

**Mode + 128** Start point for the synchronous move on the master position given in the parameter start offset

The parameter **offset** defines the variable, whose content is added to the synchronous run as an absolute shifting on the slave axis (default) or on the master axis (mode + 64).

The parameter **start offset** defines the variable, whose content shifts the starting point for the modes 1 and 2 with regard to the format beginning (default) or defines the position on the master axis on which the synchronous run is started (mode + 128).

### Command coding:

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

**Parameters:** D1 = Mode: 0..255  
 D2 = Variable number for offset value: 0..255  
 D3 = Variable number for start offset value: 0..255

**4 - 08 BIAS-Command “Synchronous settings 2“**

Synchronous settings 2; linear= X , mode= Y ,value = Z

**Explanation:**

With the command "synchronous settings 2" it is possible to suppress the resetting of the counter on the format trigger and when starting the synchronous run (linear axis).  
 With the parameters mode and value\_ it is possible to define various synchron parameters.

[Load example](#)

Drive type:	Parameters:	Permissible inputs:	
All	X: Linear Y: <a href="#">Mode</a> Z: <a href="#">Value</a>	0/1 (1= don't clear the counter) 0...255 0...65535	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

**Example:**

0 Synchronous settings 2; linear= 0, mode= 0, value = 0

**Result:**

- Mode = 1** Value = 0..500 Phase shift on slave axis offset, adjust speed as % of the actual synchron speed
- Mode = 2** Value = 0..65535 Phase shift on slave axis offset, adjust speed in increments per sampling cycle

**Note:**

- If no mode is set or the values are zero, the default offset adjust speed is offset/256, that means in 512 msec the new offset will be adjusted.

- Mode = 3** Value = 0 Movement of the axis either directional is possible  
Value = 1 Movement only in a positive count direction is possible
- Mode = 4** Value = 0 Ramp filter will effect the complete position setpoint  
Value = 1 Ramp filter will affect only the offset adjustment on the slave axis in synchronous mode
- Mode = 5** Value =0 ..500 Phase shift on the master axis offset, adjust speed as % of the actual synchron speed
- Mode = 6** Value =0 .... Phase shift on the master axis offset, adjust speed in increments per sampling cycle
- Mode = 7** Value = 0 Positive synchron start direction  
Value = 1 Negative synchron start direction
- Mode = 8** Value = 0 Gear ratio resolution 1/256 (standard)  
Value = 1 Gear ratio resolution 1/65536  
Value = 2 Gear ratio is defined with the command gear factor = [variable(x)] as a fraction between the high and the low word of variable (x)

Example:

- 0 [Variable 0 ] = 3
  - 1 [Variable 1 ] = 1
  - 2 [Variable 3 ] = [variable 1 ] \* 65536
  - 3 [Variable 2 ] = [variable 0 ] + [variable 3 ]
  - 4 Synchronous settings 2; linear= 1 ,mode = 8 , val.= 2
  - 5 Gear factor = [variable 2 ]
- Defines a gear ratio of 1/3



<b>Mode = 9 only 637f</b>	Value = 0	At the end of the synchron profile the actual defined profile is continoused.
	Value = 1	The synchronism is finished at the end of the synchron profile.
<b>Mode = 10 only 637f</b>	Value = 1	With this mode it is possible to save all parameters and values of the synchronous move during a stop or deactive state. At a restart the synchronism is activated with the stored parameters. During a stop or deactive state the incomming increments of the master are ignored.
	Value = 0	Disables mode 10, stored synchronous parameters are ignored.
<b>Mode = 11 only 631/635/637</b>	Value = 1	The overwriting of the master cycle length is not allowed, if an absolute or relative move command is called during a synchron move (The command "Position = " overwrites the cycle lenght.)
	Value = 0	The scale out function is switched off.
<b>Mode = 12 only 637f</b>	Value = 1	The scale out function is enabled. The scaling factor can adjusted during the sychronism. Variable 250 = numerator Variable 251 = denominator
	Value = 0	The reverse rotation of the master over a format change is not checked.
<b>Mode = 13 only 637f</b>	Value = 1	The reverse rotation to the format change is checked or is permitted.

**Command coding:**

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

**Parameters:** D1 = Linear: 0, 1  
D2 = Mode: 0-255  
D3, D4 = Value: 0 - 65535

## 4 - 09 BIAS-Command "Move PID, speed"

Move PID; speed; set value= X , act. value=Y , mode=Z

### Explanation:

The command "Move PID, speed" enables the controlling of the actual speed according to the control error between the set point and the actual value specified in the parameter of the PID controller (see also commands "PID [scaling](#)" and "PID [parameter](#)").

Drive type:	Parameters:	Permissible inputs:	
All	X: <a href="#">Set value</a> Y: <a href="#">Actual value</a> Z: <a href="#">Mode</a>	0...4 0...4 0...255	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

Further [information about the PID-controller of the 630 series](#)

### Example:

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 scaling; start=[variable 6 ]
* Start execution PID , output = speed setpoint
14 Start axis

```

**15 Move PID; speed; set value= 4 , act. value=0, mode=7**

Loop:

16 Jump Loop

### Note:

- Before starting a valid start command has to be encountered by either:
  - a.) the command "Start [axis](#)" before the move command, or
  - b.) low-high impulse of the start input X10.11 (X10.9 on 631)
    - **635/ 637/ 637+/ 637f**  
When using the start input X10.11 the function 0 (start input) has to be selected in the configuration.
    - **631**  
When using the start input X10.9 the function 3 (start input) has to be selected in the configuration.

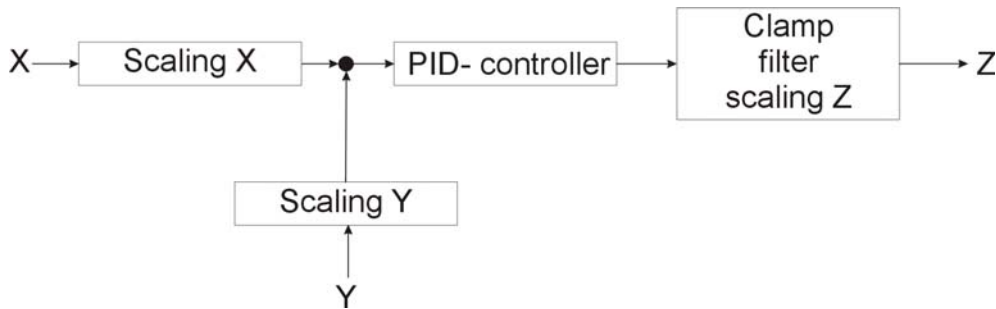
### Ergebnis:

```

* 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

```

\* Setpoint =Variable 254 =0, Actual value =ana.in.1,PID enable,I enable, Feed Forward on.



### Setpoint X Explanation:

0	Analogue input 1 ( $\pm 10V = \pm 2047$ )	
1	Analogue input 2 ( $\pm 10V = \pm 2047$ )	
2	Actual speed <a href="#">1 intern</a>	
3	<b>631/635/637</b>	<b>637+/637f</b>
	Act. speed 2 (increments /1,899 ms)	Act. speed 2 (increments /0,844 ms)
4	BIAS-variable 254	

### Actual value Y Explanation

0	Analogue input 1 ( $\pm 10V = \pm 2047$ )	
1	Analogue input 2 ( $\pm 10V = \pm 2047$ )	
2	Actual speed <a href="#">1 intern</a>	
3	<b>631/635/637</b>	<b>637+/637f</b>
	Act. speed 2 (increments /1,899 ms)	Act. speed 2 (increments /0,844 ms)
4	BIAS-variable 255	

**Mode:** The parameter "Mode" contains several bit-coded switches.

<b>Bit 0:</b>	PID enable	0 = PID controller deactivated (all parts = 0) 1 = PID controller activated
<b>Bit 1:</b>	I part enable	0 = I part deactivated 1 = I part activated
<b>Bit 2:</b>	Feed forward enable	1 = Feed forward activated
<b>Bit 3:</b>	Feed forward selection	0 = Feed forward standard (Default) 1 = Feed forward with Var. 253
<b>Bit 4:</b>	Stop integral part	0 = Integral runs 1 = Integral stop (the last actual integral value is valid)
<b>Bit 5...6:</b>	Reserved	always 0
<b>Bit 7:</b>	Real analog speed control (without PID function)	0 = Standard PID function 1 = Real analog speed control

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>09</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	00	00	00	00

**Parameters:** D1 = Set value (0-4)  
D2 = Actual value (0-4)  
D3 = Mode (0 – 255)

## 4 - 0A BIAS-Command "Move PID, torque"

Move PID, torque ; setpoint = X , act. value = Y , mode = Z

### Explanation:

The command "Move PID, torque" enables the controlling of the actual current according the control error between the set point and the actual value specified in the parameter of the PID controller.

(see also commands "PID [Scaling](#)" and "PID [parameters](#)").

Drive type:	Parameters:	Permissible inputs:	
All	X: <u>Setpoint</u> Y: <u>Act. value</u> Z: <u>Mode</u>	0...4 0...4 0...255	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

Further [information about the PID controller of 630 series](#)

### Example:

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 scaling; start=[variable 6 ]

\* Start execution PID, output = speed setpoint

14 Start axis

**15 Move PID; torque; setpoint=4, act. value=0, mode=7**

Loop:

16 Jump Loop

### Note:

- Before starting a valid start command has to be encountered by either:
  - a.) the command "Start [axis](#)" before the move command, or
  - b.) low-high impulse of the start input X10.11 (X10.9 on 631)
    - **635/ 637/ 637+/ 637f**  
When using the start input X10.11 the function 0 (start input) has to be selected in the configuration.
    - **631**  
When using the start input X10.9 the function 3 (start input) has to be selected in the configuration.

### Result:

\* 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

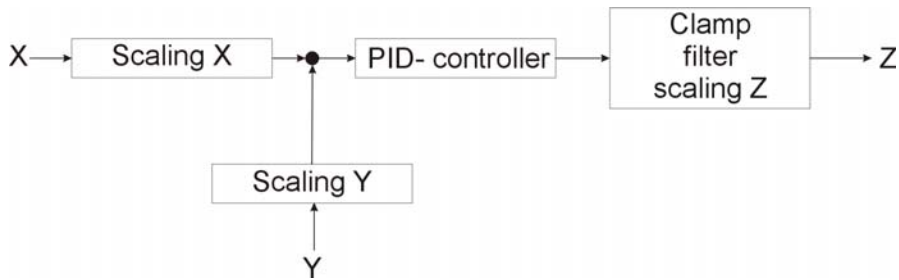
\* Scaling PID Output numerator

\* Scaling PID Output denominator

\* Overtake Scalings

\* Setpoint =Variable 254 =0, Actual value

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



Setpoint X	Explanation	
0	Analogue input 1 ( $\pm 10V = \pm 2047$ )	
1	Analogue input 2 ( $\pm 10V = \pm 2047$ )	
2	Actual speed <a href="#">1 intern</a>	
3	<b>631/635/637</b>	<b>637+/637f</b>
	Act. speed 2 (increments /1,899 ms)	Act. speed 2 (increments /0,844 ms)
4	BIAS variable 254	

Act. value Y	Explanation	
0	Analogue input 1 ( $\pm 10V = \pm 2047$ )	
1	Analogue input 2 ( $\pm 10V = \pm 2047$ )	
2	Actual speed <a href="#">1 intern</a>	
3	<b>631/635/637</b>	<b>637+/637f</b>
	Act. speed 2 (increments /1,899 ms)	Act. speed 2 (increments /0,844 ms)
4	BIAS variable 255	

**Mode:** The parameter "Mode" contains several bit-coded switches.

<b>Bit 0:</b>	PID enable	0 = PID controller deactivated (all parts = 0) 1 = PID controller activated
<b>Bit 1:</b>	I part enable	0 = I part deactivated 1 = I part activated
<b>Bit 2:</b>	Feed forward enable	1 = Feed forward activated
<b>Bit 3:</b>	Feed forward selection	0 = Feed forward standard (Default) 1 = Feed forward with Var. 253
<b>Bit 4:</b>	Stop integral part	0 = Integral runs 1 = Integral stop (the last actual integral value is valid)
<b>Bit 5...6:</b>	Reserved	always 0
<b>Bit 7:</b>	Real analog speed control (without PID function)	0 = Standard PID function 1 = Real analog speed control

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>0A</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	00	00	00	00

**Parameters:** D1 = Setpoint (0-4)  
D2 = Act. value (0-4)  
D3 = Mode (0 – 255)

#### 4 - 0B BIAS-Command "Set position [axis no. ] = const."

Set position [axis no.X] = Y INCR

##### Explanation:

With the command "Set position [axis no. ] = const." it is possible to set the set position of the selected axis to a certain value.

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
637f with Multi-Axis-Function (Mode 8 = 1)	X: Axis number Y: Constant	1 - 3 +/- 231	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

##### Example:

0 Set position [axis no.1] = 10000 INCR

##### Result:

The set position of the axis 1 is loaded on 10000 increments.

##### Note:

- This command is available with the 637f starting from the firmware V8.19

##### Command coding:

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

Parameters: D1 = Axis 1 - 3  
D2-D5 = Constant +/- 231

## 4 - 0C BIAS-Command "Set position [axis no. ] = [variable X]"

Set position [axis no.X] = [variable Y]

### Explanation:

With the command "Set position [axis no.X] = [variable Y]" it is possible to set the set position of the selected axis on contents of the defined variable.

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
637f with Multi-Axis-Function (Mode 8 = 1)	X: Axis number Y: Variable	1 - 3 0...255	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

### Example:

0 [Variable 0] = 10000

1 Set position [axis no.1] = [variable 0]

### Result:

The set position of the axis 1 is loaded on 10000 increments.

### Note:

- This command is available with the 637f starting from the firmware V8.19

### Command coding:

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

Parameters: D1 = Axis number: 1 - 3  
D2 = Variable number: 0...255

## 4 - 0D BIAS-Command "Move relative"

Move relative; axisno. = X, target = Y

### Explanation:

The command "Move relative" starts the axis X on the position, which results from the sum of the current set position plus the relative distance defined in the parameter set.

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
637f with Multi-Axis-Function: (Mode 8 = 1)	X: Axis number Y: Target	1 - 3 1 = Setpoint into position controller 2 = Setpoint on CAN2-PDO3 4 = Setpoint on CAN2-PDO4 8 = Setpoint on X40 output  The parallel connection of the setpoint targets is possible by addition of the function coding.	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

This command is permitted only with the use of the Multi-Axes-Function with the controller type 637f, therefore the parameters must be pre-loaded over the command "Load [parameter set 0 = \[variable\[X\]\]](#)".

### Example:

- 0 [Variable 10 ] = 100000
- 1 [Variable 11 ] = 1000
- 2 [Variable 12 ] = 4000
- 3 [Variable 13 ] = 4000
- 4 [Variable 0 ] = 10
- 5 Load parameter set 0 = [variable [0]]
- 6 Start axis
- 7 Move relative; axis no. = 1 , target = 1

### Result

Starting from the momentary set position the axis moves on with the speed 1000 for 100000 increments.

### Note:

- This command is available with the 637f starting from the firmware V8.19

### Command coding:

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

Parameters: D1 = Axis number: 1 - 3  
D2 = Target: 1...15



## 4 - 0E BIAS-Command "Start axis"

**Explanation:**

The command "Start axis" places the start mark for the following move command. With this command a move command can be executed without external start condition. Every move command resets the start mark.

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
All	none	-	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

**Example:**

**0 Start Axis**

- 1 Move incremental position
- 2 Wait for "position reached"
- 3 Move incremental position
- 4 ....

**Result:**

The first move command "Move incremental position" is executed immediately. After reaching the target position (block 2) the second move command waits for a Low-High impulse of the start input X 10.11 (X 10.9 with the 631).

**Note:**

- The command "Start axis" replaces the Low-High impulse of the start input X10.11 (X10.9 with the 631) as a valid start mark.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>0E</b>	00	00	00	00	00	00	00

## 4 - 0F BIAS-Command "Stop axis"

Stop axis ; mode = X ; axis no. = Y

### Explanation:

The command "Stop axis; mode =" stops drive movements with the function specified in the parameter mode.

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
631/635/637/ 637+:	X: Mode	<b>0/1</b> mode 0 = stop abruptly (target position = actual position) mode 1 = stop guided (with the current deceleration ramp)	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
637f with Multi-Axis-Function: (Mode 8 = 1)	Y: Axis number	<b>0</b> Stops all axis and disables the multiaxis function <b>1 – 3</b> guided stop of the selceted axis	

### Example:

0 Start axis  
 1 Move infinite positive  
 2 Wait time 1000 ms  
 3 Deceleration = 10000 rpm/s  
**4 Stop Axis, Mode 1**  
 5 Wait for "position reached"

### Result:

The move command "move infinite positive" is stopped after 1 second with the defined deceleration ramp of 10000 rpm/s.

### Note:

- If the axis is to be stopped over a certain deceleration ramp, the use of the command "Stop [axis + parameter](#)" is recommended.
- After the command "Stop axis" it is recommended to insert the command Wait [for "Pos. reached"](#) into BIAS processing.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>0F</b>	<b>X</b>	<b>Y</b>	00	00	00	00	00

**Parameters:** D1 = Mode 0 = stop abruptly (target position = actual position)  
 D1 = Mode 1 = stop guided (with the current deceleration ramp)  
 D2 = AxisNo. 1...3

## 4 - 10 BIAS-Command "Move position + parameter"

Move position; v = X rpm, s= Y INCR

### Explanation:

The command "Move position + parameter" starts the axis at the speed specified in the parameter "v=" to the position defined in the parameter "s="

[Load example](#)

### Parameters:

v = 1...12000 rpm

s = ± 231 Increments

The setpoint calculation always uses the last defined parameters of: acceleration, [biascom21-E.htm](#) deceleration, Ramp [filter](#)

After power on these parameters are initialized with the values of the tuning position loop.

- BIAS program
- PLC program
- Mathematics

### Example:

0 Start axis

**1 Move position; v= 100 rpm, s= 1000 INCR**

2 Wait for "position. reached"

3....

### Result:

The axis will move to the absolute position 1000 increments at a speed of 100 rpm.

The BIAS processing is continued with block 3 after reaching [the position](#).

### Note:

- Before starting a valid start mark has to occur:
  - a.) low-high impulse of the start input X10.11 (X10.9 at 631) or
  - b.) the command "Start axis" before the move command.
- **635/ 637/ 637+/637f**  
When using the start input X10.11 the function 0 (start input) has to be selected for the input X10.11 in the configuration
- **631**  
When using the start input X10.9 the function 3 (start input) has to be selected for the input X10.9 in the configuration.
- **631/ 635 / 637**  
The maximum permissible speed depends on the resolver resolution configuration. With a high resolver resolution only 4000 rpm is allowed.

### Command coding:

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

**Parameters:** D1, D2 = Speed 1....12000 rpm  
D3...D6 = Absolute position ±231 Increments

**4 - 11 BIAS-Command “Move incremental position + parameter“**

Move incremental position; v = X rpm, s= Y INCR

**Explanation:**

The command "move incremental position + parameter" starts the axis at the speed specified in the parameters "v=" to the position resulting from the current position + the position defined in the parameter "s".

[Load example](#)

**Parameters:**

v = 1...12000 rpm

s = ± 231 Increments

The setpoint calculation always uses the last defined parameters of: acceleration, [biascom21-E.htm](#) deceleration, Ramp [filter](#)

After power on these parameters are initialized with the values of the tuning position loop.

- BIAS program
- PLC program
- Mathematics

**Example:**

0 Start axis

**1 Move incremental position; v= 100 rpm, s= 1000 INCR**

2 Wait for "position reached"

3....

**Result:**

The axis moves 1000 increments from the current position at a speed of 100 rpm.

The BIAS processing is continued with block 3 after reaching the position reaching [the position](#).

**Note:**

- Before starting a valid start mark has to occur:
  - a.) low-high impulse of the start input X10.11 (X10.9 at 631) or
  - b.) the command "Start [axis](#)" before the move command
    - **635/ 637/637+/637f**  
When using the start input X10.11 the function 0 (start input) has to be selected for the input X10.11 in the configuration.
    - **631**  
When using the start input X10.9 the function 3 (start input) has to be selected for the input X10.9 in the configuration.
- **631/ 635/ 637**  
The maximum permissible speed depends on the resolver resolution configuration. With a high resolver resolution only 4000 rpm is allowed.

**Command coding:**

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

**Parameters:** D1,D2 = Speed 1....12000 rpm  
D3...D6 = Relative position ±231 Increments

## 4 - 12 BIAS-Command "Move datum + parameter"

Move datum; mode=X ; v = Y rpm , s= Z INCR

**Explanation:**

The command "move datum +" starts the reference move at the speed specified in the parameter "v=" corresponding to the specified mode.

The reference shift is defined in the parameter "s=".

[631 load example](#)

[635/637 load example](#)

[637+ load example](#)

Drive type:	Parameters:	Permissible inputs:
All	<b>X: Mode</b>	Mode: <a href="#">0...23</a>
	<b>Y: Speed</b>	v =1...12000 rpm (Speed)
	<b>Z: Reference point shift</b>	s =± 231 Increments (Position)

- BIAS program
- PLC program
- Mathematics

The setpoint calculation always uses the last defined parameters of: Acceleration, Deceleration, Ramp [filter](#)

After power on these parameters are initialized with the values of the tuning position loop menu.

**Example:**

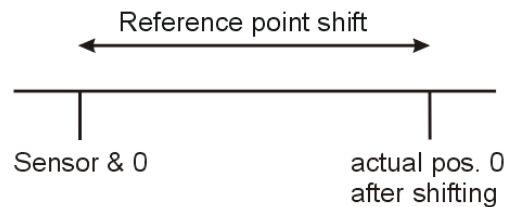
0 Start axis

1 Move datum, mode 16, v= 100rpm, s=4000 Inkr

2 Wait for "position reached"

**Result:**

The axis moves in the positive direction to the Low-High slope of the reference sensor and the encoder zero position.



The final reference position is shifted by +4000 increments from this position.

**Note:**

- The encoder could be resolver or HIPERFACE®  
In position control with actual position 2 (X40) the 0-position of the external encoder is used.
- Before starting a valid start mark has to occur:
  - a.) the command "Start [axis](#)" before the move command or
  - b.) low-high impulse of the start input X10.11 (X10.9 at 631)
    - **635/ 637/ 637+/637f**  
When using the start input X10.11 the function 0 (start input) has to be selected for the input X10.11 in the configuration.
    - **631**  
When using the start input X10.9 the function 3 (start input) has to be selected for the input X10.9 in the configuration.
- **635/ 637/ 637+/637f**  
When using the reference sensor input X10.24 the function 1 (reference sensor) has to be selected for the input X10.24 in the configuration.
- **631**  
When using the reference sensor for one of the inputs (X10.8, X10.9 or X10.10) the function 1 (reference sensor) has to be selected in the configuration.
- **631/ 635/ 637**  
The maximum permissible speed depends on the resolver resolution configuration.  
With a high resolver resolution only 4000 rpm is allowed.
- **637f**  
When the drive 637f is mounted by the option board C28 the input X120.4 can configured as reference sensor. The reference sensor input configuration at the X10.24 clamp has no function.

**Command coding:**

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

**Parameters:**

D1 =	Reference mode: 0...23
D2, D3 =	Speed 1....12000 rpm
D4...D7 =	Reference point shift, position $\pm 231$ Increments

## 4 - 13 BIAS-Command “Move infinite positive + parameter“

Move positive ; v= X rpm , a+ = Y rpm/s , a- = Z rpm/s

### Explanation:

The command "Move infinite positive + parameter" starts the axis with the specified parameters "v=" (speed), "a+" (acceleration) and a- (deceleration) in the positive direction.

[631 load example](#)

[635/637 load example](#)

[637+ load example](#)

### Parameters:

v = 1...12000 rpm

a+ = 5...320000 rpm/s

a- = 5...320000 rpm/s

The setpoint calculation always uses the last defined parameters of: Ramp [filter](#).

After power on these parameters are initialized with the values of the tuning position loop menu.

BIAS program

PLC program

Mathematics

### Example:

**0 Move positive; v= 900 rpm, a+= 200rpm/s, a-= 5000 rpm/s**

1 If input 11 == 1 then jump 1

2 Stop axis ; mode 1

3 ...

### Result:

The axis moves in the positive direction as soon as the start input is activated.

After deactivation of the start input the axis is stopped.

### Note:

- Before starting a valid start mark has to occur:
  - a.) the command "Start [axis](#)" [before the move command](#)
  - b.) low-high impulse of the start input X10.11 (X10.9 at 631)
    - **635/ 637/ 637+/637f**  
When using the start input X10.11 the function 0 (start input) has to be selected for the input X10.11 in the configuration
    - **631**  
When using the start input X10.9 the function 3 (start input) has to be selected for the input X10.9 in the configuration.
- **631/ 635/ 637** The maximum permissible speed depends on the resolver resolution configuration. With a high resolver resolution only 4000 rpm is allowed.

### Command coding:

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

**Parameters:** D1,D2 = Speed 1...12000 [rpm]  
 D3,D4 = Acceleration 1...64000 [5\*rpm/s]  
 D5,D6 = Deceleration 1...64000 [5\*rpm/s]

#### 4 - 14 BIAS-Command "Move infinite negative + parameter"

Move negative; v= X rpm , a+ = Y rpm/s , a- = Z rpm/s

##### Explanation:

The command "Move infinite negative + parameter" starts the axis with the specified parameters "v=" (speed), "a+" (acceleration) and a- (deceleration) in the negative direction.

[631 load example](#)

[635/637 load example](#)

[637+ load example](#)

##### Parameters:

v = 1...12000 rpm

a+= 5...320000 rpm/s

a-= 5...320000 rpm/s

The setpoint calculation always uses the last defined parameters of: Ramp [filter](#)

After power on these parameters are initialized with the values of the tuning position loop menu.

BIAS program

PLC program

Mathematics

##### Example:

**0 Move negative; v= 900 rpm, a+= 200rpm/s, a-= 5000 rpm/s**

1 If input 11 == 1 then jump 1

2 Stop axis ; mode 1

3 ...

##### Result:

The axis moves in the negative direction as soon as the start input is activated.

After deactivation of the start input the axis is stopped.

##### Note:

- Before starting a valid start mark has to occur:
  - c. The command "Start [axis](#)" before the move command.
  - d. low-high impulse of the start input X10.11 (X10.9 on 631)
    - **635/ 637/ 637+/637f**  
When using the start input X10.11 the function 0 (start input) has to be selected for the input X10.11 in the configuration.
    - **631**  
When using the start input X10.9 the function 3 (start input) has to be selected for the input X10.9 in the configuration.
- **631/ 635/ 637**  
The maximum permissible speed depends on the resolver resolution configuration. With a high resolver resolution only 4000 rpm is allowed.

##### Command coding:

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

**Parameters:** D1, D2 = Speed 1...12000 [rpm]

D3, D4 = Acceleration 1...64000 [5\*rpm/s]

D5, D6 = Deceleration 1...64000 [5\*rpm/s]



## 4 - 15 BIAS-Command “Move synchron + parameter“

Move synchron; gear factor = X

### Explanation:

The command "Move synchron + parameter" starts the axis with the specified gear factor synchronous to the master increments.

[631 load example](#)

[635/637 load example](#)

[637+ load example](#)

Drive type:	Parameters:	Permissible inputs:
All	<b>X: Standard:</b>  <b>X: extended:</b>	= Gear factor +/-127, with 1/256 accuracy Synchronous <a href="#">settings 2; linear= 0, mode= 8 ,value = 1</a> = Gear factor +/-127, with 1/65536 accuracy

- BIAS program
- PLC program
- Mathematics

The setpoint calculation always uses the last defined parameters of: Ramp [filter](#)

After power on these parameters are initialized with the values of the tuning position loop menu.

[Further Information about synchronous functions of 630 series](#)

### Example:

0 Cycle length = 0 INCR

1 **Move synchron; gear factor = 1.0**

2 If input 11 == 1 then jump 2

3 Stop axis ; mode 1

4 ...

### Result:

After the Low-High slope of the start input the axis moves synchronously (per master increment = gear factor \* increments), until deactivation of the start input.

### Note:

- Before starting a valid start mark has to occur:
  - a. the command "Start [axis](#)" before the move command optional : sensor; [mode=128](#)
  - b. low-high impulse of the start input X10.11 (X10.9 at 631)
    - **635/ 637/ 637+/637f**  
When using the start input X10.11 the function 0 (start input) has to be selected for the input X10.11 in the configuration.
    - **631**  
When using the start input X10.9 the function 3 (start input) has to be selected for the input X10.9 in the configuration.
- The function X40 has to be selected as incremental input in the counter configuration.
- Changing the gear ratio is possible during the synchronous run with the command “Gear [factor =](#)”.
- The command "Cycle [length =](#)" placed before the move command defines the length of the format.

### Command coding:

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

**Parameters:** D1-D4= Gear factor, X(high) = +/-127

## 4 - 16 BIAS-Command "Move analogue value + integrator"

Move infinite with analog value; integrator = X rpm/s

### Explanation:

The command "Move infinite with analog value" starts the axis to the speed resulting from the analog setpoint X10.5 to X10.18(631 X10.1 to X10.2) multiplied by the speed setpoint scaling value.

The permitted rate of speed change can be limited by the parameter "integrator" (0 = no limiting)

631 [load example](#)

635/637 [load example](#)

637+ [load example](#)

**Parameters: Integrator = 0...160000 rpm/s; 0 = Integrator deactivated**

- BIAS program
- PLC program
- Mathematics

### Example:

**0 Move infinite with analog value; integrator = 10000 rpm/s**

- 1 If input X == 1 then 1
- 2 Stop axis; mode 1

### Result:

The axis moves according to the analog setpoint value (as operating mode "speed control"), as long as input X is active.

The maximum possible speed change is limited to 10000 rpm/s.

### Note:

- Before starting a valid start mark has to occur:
  - a. The command "Start [axis](#)" [before the move command](#).
  - b. Low-high impulse of the start input X10.11 (X10.9 at 631)
    - **635/ 637/ 637+/637f**  
When using the start input X10.11 the function 0 (start input) has to be selected for the input X10.11 in the configuration.
    - **631**  
When using the start input X10.9 the function 3 (start input) has to be selected for the input X10.9 in the configuration
- The command "Ramp [filter](#) =" must deactivate before this move command.
- The parameter acceleration and deceleration haven't influence to the ramps.

### Command coding:

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

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

## 4 - 17 BIAS-Command "Move speed + integrator"

Move speed, speed =X rpm , integrat.= Y rpm/s

### Explanation:

The command "Move speed + integrator" starts the axis according to value "Speed" by-passing the position controller. This corresponds to the operating mode "speed control". The permitted rate of speed change can be limited by the parameter "integrator" (0 = no limiting)

631 [load example](#)

635/637 [load example](#)

637+ [load example](#)

**Parameters** X: v =1...12000 rpm  
Y: Integrator = 0...160000 rpm/s; 0 = Integ. deactive

- BIAS program
- PLC program
- Mathematics

### Example:

- 0 Start axis
- 1 Move speed ,speed =1000 rpm , integrat.= 10000 rpm/s
- 2 If input X == 1 then jump 1
- 3 Stop axis; mode 1

### Result:

The axis moves according to the set speed of 1000 rpm (as operating mode "speed control") as long as the start input X is active. During the process, speed changes are limited to a maximum of 10000 rpm/s.

### Note:

- Before starting a valid start command has to be encountered by either:
  - a.) the command "Start [axis](#)" before the move command.
  - b.) low-high impulse of the start input X10.11 (X10.9 at 631)
    - **635/ 637/ 637+/637f**  
When using the start input X10.11 the function 0 (start input) has to be selected for the input X10.11 in the configuration.
    - **631**  
When using the start input X10.9 the function 3 (start input) has to be selected for the input X10.9 in the configuration.
- The command "Ramp [filter =](#)" must deactivat before thhis move move command.
- The parameter acceleration and decelleration haven't influence to the ramps

### Command coding:

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

**Parameters:** D1,D2= Speed [rpm]  
D3,D4= Integrator [5\*rpm/s]

#### 4 - 1A BIAS command "Cycle length = Const."

Cycle length = X INCR

##### Explanation:

The command "Cycle length=" specifies the master cycle length for synchronous applications.

A value of 0 deactivates the format check.

[631\\_Load\\_example](#)

[635/637\\_Load\\_example](#)

[637+\\_Load\\_example](#)

**Parameter:** X: s = ± 231 Increments

- BIAS program  
 PLC program  
 Mathematics

Further [information about synchronous functions of the 630 series](#)

**Example:**

**Result:**

**Note:**

**Command coding:**

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

**Parameters:** D1-D4= Cycle length: ± 231

#### 4 - 1B BIAS command "Cycle length = [variable X] "

Cycle length = [variable X]

##### Explanation:

The command "Cycle length = [variable X]" specifies the master cycle length in increments specified by the content of [variable X] for synchronous applications.

A variable content of 0 deactivates the format check.

A value of 0 deactivates the format check.

[631\\_Load\\_example](#)

[635/637\\_Load\\_example](#)

[637+\\_Load\\_example](#)

**Parameter:** X: 0 -255

- BIAS program  
 PLC program  
 Mathematics

Further [information about synchronous functions of the 630 series](#)

**Example:**

**Result:**

**Note:**

**Command coding:**

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

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

## 4 - 1C BIAS-Command “Load parameter set X = [variable [Y]]“

Load parameter set X = [variable [Y]]

**Explanation:**

The command “Load parameter set X = [variable [Y]]” defines the starting variable of the appropriate parameter set of type X by the contents of the variable Y.



Axis no./ cam (by type 2):	Parameters: X		
	Type 0: Parameter set for positioning commands	Type 1: Parameter set for synchronous move	Type 2: Parameter set for cam function
1	0	1	8
2	2	3	9
3	4	5	10
4	6 (Reserved)	7 (Reserved)	11

**Parameters: Y**  
**0 - 255**

Pointer variable Y on the starting variable of the appropriate parameters set of type X

Further [information about synchronous functions of the 630 series](#)

- BIAS program
- PLC Program
- Mathematics

**Example:**

**Result:**

**Note:**

- The function “Synchronous run” receives their control parameter before the start from the variables, which are indicated in the parameter set and in the state “Active” from the axis state.
- Acceleration and deceleration are transferred to 5min-1/s.

**Parameterset: Type 0:**

**Parameterset for positioning commands**

<b>Var. + 0: Position</b>	Target position in increments (+/-2 <sup>31</sup> )
<b>Var. + 1: Speed</b>	Max. speed, every time positive, in rpm
<b>Var. + 2: Acceleration</b>	Acceleration (5-64000) in rpm/(min*Sec)
<b>Var. + 3: Deceleration</b>	Deceleration (5-64000) in rpm/(min*sec)

Parameter set: Type 1:

Parameterset for synchronous commands

<b>0: Start angle</b>	Defines switch-on point of the synchronous movement . Unit: Increments of the selected reference for the synchronous movement.
<b>1: Delay time comp., Sync.</b>	The phase shift between master and slave, resulting from the scanning, mechanical delay and electrical switching times, is compensated by addition of a speed-proportional offset to the reference of the synchronous move. The input is made in %. 100% corresponds to the accurate compensation of the delay caused by the scanning in the controller (844uSec) .
<b>2: Gear factor numerator</b>	Gear factor numerator: 32bit signed Integer value.
<b>3: Gear factor denominator</b>	Gear factor denominator: 31bit unsigned Integer value.
<b>4: Move_Linked_ACC</b>	Distance in increments during ramp-up of the slave; distance = input/2
<b>5: Move_Linked_CONT</b>	Distance in increments during synchronous movement of the slave
<b>6: Move_Linked_DEC</b>	Distance in increments during ramp-down of the slave; distance = input/2
<b>7: Move_Linked_WAIT</b>	Distance in increments during waiting. It is only used by MOVE_LINKED = 0.
<b>8: CAM_Startoffset</b>	Shift of the start point of the CAM profile toward the master axis. The start takes place on the slave axis incrementally.
<b>9: CAM No.</b>	Indicates which CAM profile comes to execution. General altogether 2048 profile supporting points are momentarily possible to be stored in the controller. Which can be distributed then on 2-16 different profiles.
<b>10: Scaling numerator</b>	This parameter defines numerator of the output scaling.
<b>11: Scaling denominator</b>	This parameter defines denominator of the output scaling.
<b>12: Sync. settings</b>	<a href="#">Bit code for synchronous run options</a>

## Bit Code for synchronous movement options

Bit	Name	Meaning
0	START_ANGLE	Selection of the function for synchronous movement start on defined master angle (Bit 0 =1).
1	START_DIRECTION	Direction in which the start is to take place on defined master angle (0 = positive counting direction).
2	GEARBOX_TRIGGER_ON	Switch on of the trigger function for the gearbox. The gear factor is took over at the trigger point at format length reached (gear factor denominator) ("Gearbox Trigger On" =1). If the gear factor is took over immediately the flag "bit gearbox Trigger Ext" must set to 1 after the new gear factor was written (Trigger On =0).
3	Backstop On	Disables the synchronous move during a back rotation of the master at the beginning of the Profil, until the master reaches the backstop position. The direction is defined in the flag "Start direction" If the position of the master is behind the backstop position the error bit "Backstop Error" is set to 1.
4	CLUTCH_ON	Switch-on of the clutch function (Bit 4 =1)
5	RAMP_DIRECTION_Y	Slave direction of the clutch function
6	RAMP_UP	Starts the synchronous ramp-up of the clutch function
7	RAMP_DOWN	Starts the synchronous ramp-down of the clutch function
8	RAMP_WAIT	Starts the synchronous waiting of the clutch function
9	MOVE_LINKED	Move Linked = 1, moves the positions Move Linked ACC, Move Linked Cont and Move Linked DEC consecutively.
10	MERGE_MOVE_LINKED	Start a new profile,by a positive slope of the flag " Merge Move Linked" the new formats are took over and the flag "Merge Move Linked" is cleared after the restart.
11	CAMING_ON	Switches on the CAM function block.
12	STOP_CAM_END	Stops the synchronous movement at end of the profile.
13	CAM_BACKSTOP_OFF <b><u>This function is still in preparation !!!</u></b>	If the reference is turning backwards, the movement of the CAM is stopped at the beginning of the profile, until the reference is crossing again the switch off point in positive direction. With this flag it is possible to carry on CAM execution in negative direction of the reference.
14	RESERVE 1	
15	RESERVE 2	<b>Important !</b> Bit 15 = 0

## Parameter set: Type 2

## Parameter set for cams

0: Mode	Cam switching device mode
1: Modulo 1-4	Cycle length of the cam switching device. in this application 65536 Incr == 360 degrees at the SSI.
2: Switch-on point 1-4	Defines switch-on point of the output with modes 2,4 and 8. Unit: Increments of the SSI trajectory
3: Switch-off point 1-4	Defines switch-off point of the output with modes 2. Unit: Increments of the SSI trajectory
4: Cycle duration 1-4	Defines the cyclic duration of the output with mode 8. Unit: ms
5: Delay time comp. 1-4	The phase shift between master and slave, resulting from the scanning, mechanical delay and electrical switching times, is compensate by addition of the speed-proportional offset to SSI trajectory. The input is made in %. 100% correspond to the accurate compensation of the delay of the SSI scanning in the controller (422µs).

## Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>1C</b>	<b>X</b>	<b>Y</b>	00	00	00	00	00

**Parameters:** D1= Parameter set number: 0...255  
D2= Variable number: 0...255



## 4 - 1F BIAS-Command "Stop axis + parameter"

Stop axis; mode = X a- = Y rpm/s

### Explanation:

The command "Stop Axis; mode = X a- = Y" stops the drive with the function indicated in the mode.

Load [example](#)

Drive type:	Parameters:	Permissible inputs:	
All	<b>X: Mode</b>	<b>0/1</b> 0 = Stop abruptly (target position = actual position) 1 = Stop guided	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
	<b>Y: Deceleration</b>	a-= 5...320000 rpm/s	

### Example:

0 Start axis  
 1 Move infinite positive  
 2 Wait time 1000 ms  
 3 Stop axis, mode 1 a-= 10000 rpm/s  
 4 Wait for "position reached"

### Result:

The move command "move infinite positive" is stopped after 1 seconds with the specified deceleration ramp of 10000 rpm/s.

### Note:

- After the command "Stop axis + parameter" the use of the command "Wait [for "Position reached"](#)" is recommended.

### Command coding:

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

**Parameters:** D1 = Mode 0 = Stop abruptly (target position = actual position)  
 D1 = Mode 1 = Stop guided  
 D2, D3 = Deceleration 1...64000 [5\*rpm/s]

## 4 - 20 BIAS-Command "Position = const."

Position = X INCR

### Explanation:

The command "Position =" defines the setpoint position for the move commands. The defined position remains in effect until the next move command.

Drive type:	Parameters:	Permissible inputs:	
All	X:	± 231 increments	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

### Example:

**0** Position = 10000 INKR  
**1** Start axis  
**2** Move incremental position  
**3** Wait for "position reached"  
**4** Jump 1

### Result:

The axis moves 10000 increments on a Low-High slope of the start input.

### Note:

- When using commands with direct position specification, the command will be overwritten.
- When using calculated units, changes in the position entered can occur since these positions always have to be rounded to an integral number of increments.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>20</b>	<b>X(low)</b>			<b>X(high)</b>	00	00	00

**Parameters:** D1...D4 = Absolut position ±231 increments

## 4 - 21 BIAS-Command "Speed = const."

Speed = X rpm

**Explanation:**

The command "Speed =" defines the maximum speed for the move commands.  
The defined speed remains in effect until the next speed command.



Drive type:	Parameters:	Permissible inputs
All	X:	1...12000

- BIAS program
- PLC program
- Mathematics

**Example:**

- 0 Speed = 1000 rpm
- 1 Position = 10000 INKR
- 2 Start axis
- 3 Move incremental position
- 4 Wait for "position reached
- 5 Jump 5

**Result:**

The axis moves 10000 increments from the current actual position at 1000 rpm.

**Note:**

- When using commands with direct speed specification, the command will be overwritten.
- **631/635/637/The** maximum permissible speed depends on the configured resolver resolution. With a high resolver resolution only 4000 rpm is allowed.

**Command coding:**

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

Parameters: D1,D2 = speed 1....12000 [rpm]

**4 - 22 BIAS-Command "Acceleration = const."**

Acceleration = X    rpm/s

**Explanation:**

The command "Acceleration =" defines the acceleration ramp for the move commands.  
The defined acceleration remains in effect until the next acceleration command.



Drive type:	Parameters:	Permissible inputs	
All	X:	5...320000	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

**Example:**

- 0    **Acceleration = 5000 rpm/s**
- 1    Speed = 1000
- 2    Position = 10000 INKR
- 3    Start axis
- 4    Move incremental position
- 5    Wait for "position reached"
- 6    Jump 6

**Result:**

The axis moves 10000 increments from the current actual position at 1000 rpm and an acceleration of 10000 rpm/s.

**Note:**

- When using commands with direct acceleration specification, the command will be overwritten.
- The parameter acceleration is used by the change of speed from 0 rpm to a defined speed or by change of speed with the same sign.
  - The parameter acceleration is used by an active move command if the speed is decreased (without different sign).
  - The variation of the speed over 0 rpm, the parameter deceleration is used to 0 rpm and the parameter acceleration is used to the defined new speed.

With minimum acceleration to 4000rpm

Exp.:

a = 5 rpm/s

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

==>

With maximum acceleration to 4000rpm

Exp.:

a = 320000 rpm/s

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

==>

**Command coding:**

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

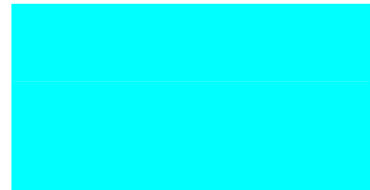
**Parameters:** D1,D2 = Acceleration: 1...64000 [5\*rpm/s]

## 4 - 23 BIAS-Command “Deceleration = const.”

Deceleration = X rpm/s

**Explanation:**

The command “Deceleration =” defines the deceleration ramp for the move commands.  
The defined deceleration remains in effect until the next deceleration command.



Move type:	Parameters:	Permissible inputs:			
		0.320000	<input checked="" type="checkbox"/>	BIAS	program
			<input type="checkbox"/>	PLC	program
			<input type="checkbox"/>	Mathematics	

**Example:**

- 0 Deceleration = 10000 rpm/s
- 1 Acceleration = 5000 rpm/s
- 2 Speed = 1000
- 3 Position = 10000 INKR
- 4 Start axis
- 5 Move incremental position
- 6 Wait for “Pos reached”
- 7 Jump 7

**Result:**

The axis moves 10000 increments from the current actual position at 1000 rpm with an acceleration rate of 10000 rpm/s and a deceleration rate of 5000 rpm/s

**Note:**

- When using commands with direct deceleration specification, the command will be overwritten.
- The parameter deceleration is used by the change of speed to 0 rpm
  - The parameter acceleration is used by an active move command if the speed is decreased (without different sign).
  - The variation of the speed over 0 rpm, the parameter deceleration is used to 0 rpm and the parameter acceleration is used to the defined new speed.

**Command coding:**

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

Parameters: D1,D2 = Deceleration 1....64000 [5\*rpm/s]

## 4 - 24 BIAS-Command "Gear factor = const."

Gear factor = X

### Explanation:

The command "gear factor =" determines the gear ratio between master and slave (the axis following the master) during a synchronous run ("Move [synchron](#)").

The gear factor specifies the number of increments to be moved per master increment.

With the command "Synchronous [settings 2](#)" with mode 8 it is possible to select the resolution of the gear factor (value 0, 1 or 2).

Drive type:	Parameters:	Permissible inputs:	
All	<b>X: standard:</b>  <b>X: extended:</b>	= Gear factor +/-127, with 1/256 accuracy adjustment see: Synchronous <a href="#">settings 2</a> ; <a href="#">linear= 0, mode= 8, value = 1</a> = Gear factor +/-127, with 1/65536 accuracy adjustment	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

Further [more information about synchron functionen of the 630 Series](#)

### Example: 635/637/637+/637f

```

0   Gear factor = 1.0
1   Move synchron
2   If input 11 == 1 then jump 2
3   Stop axis; mode 1
  
```

### Result:

The axis moves synchronously until the start input 11 drops. With this, 1 increments are traversed per 1 increments (master increments) counted on the X40-input

### Example:631

```

0   Gear factor = 1.0
1   Move synchron
2   If input 9 == 1 then jump 2
3   Stop axis; mode 1
  
```

### Result:

The axis moves synchronously until the start input 9 drops. With this, 1 increments are traversed per 1 increments (master increments) counted on the X40-input.

### Note:

- The parameter "gear factor" must be defined before the command "move synchron".
- Changes of the gear factor during the synchronous run will be activated immediately (without the command "Update [parameter](#)").

### Command coding:

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

**Parameters:** D1-D4= Gear factor, X(high) = +/-127

## 4 - 25 BIAS-Command "Position reached" window = const.

"Pos. reached" window = X

### Explanation:

The command "Position reached window =" defines the range in which the position to be moved to is regarded as reached.

The range results from the setpoint position ± the defined window.

See also: [Position reached](#).

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

### Example:

**0 "Position reached" window = 400**

- 1 Position = 10000 incr
- 2 Start axis
- 3 Move position
- 4 Wait for "position reached"
- 5 .....

### Result:

The BIAS execution is continued with line 5 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.

See also: Parameter ["Position reached" window](#)

### Note:

1. **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).
2. **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).

### Command coding:

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

Parameters: D1,D2 = +/-window value 1....32000 Increments

## 4 - 26 BIAS-Command "Remaining position= const."

Remaining position = X INKR

### Explanation:

The command "Remaining position =" defines the position, which is added together with the latch value measured, when positioning with a sensor. The defined remaining position remains in effect until the next remaining position command.

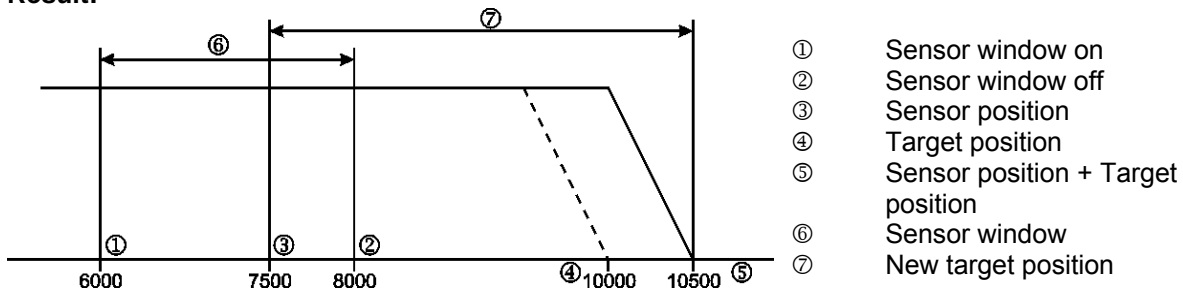
Drive type:	Parameters:	Permissible inputs:	
All	X:	± 231 Increments	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

### Example:

```

0 Actual position 1 = 0 INCR
1 Position = 10000 INCR
2 Sensor; mode= 10 on= 6000 Incr off= 8000 Incr
3 Remaining position = 3000
4 Move position
5 Wait for "position reached"
6 Jump 6
  
```

### Result:



The axis starts to move to the absolute position of 10000 increments.

If a Low-High slope on the sensor X10.4 (X10.9 on 631) is evaluated between the positions 6000 and 8000 increments then the remaining position is added to the latch position and moved to.

With an assumed sensor position of 7500 increments, the new target position is:

7500 increments + remaining position 3000 increments = 10500 increments

### Note:

- This command is only to be used in combination with the command "Sensor [window; mode =; mode 10](#)".
- When using calculated units, changes in the position entered can occur since these positions always have to be rounded to an integral number of increments.

### Command coding:

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

**Parameters:** D1...D4 = Remaining position ±231 Increments

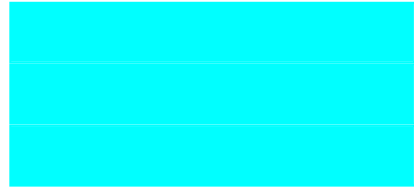


## 4 - 27 BIAS-Command “Ramp filter = const.,[variable Y]“

Ramp filter = X , [variable Y ]

### Explanation:

The command "Ramp filter =" sets the filter constant of the ramp filter. The command thus allows a torque dampening (S-curve) of the acceleration and braking ramp.



Drive type:	Parameters:	Permissible inputs:	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
631/635/637 637+:	X: X:	0, 4, 8, 12, 16, 20, 24, 28, 32 0 - 7	
637f:	X: Y: variable	0 - 8 or 255 0 - 255 permitted variable content 0 - 8	

### Example:

### Result:

### Note:

- At the move commands
  - “Move PID, speed“
  - “ Move PID, torque“
  - “ Move analogue value“
  - “ Move speed“
 the ramp filter must be loaded with 0.
- The value 0 switches the ramp filter off.
- 637f:** If the value X is 255, the ramp filter is loaded by the content of the specified variable.

### Command coding:

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

**Parameters:** D1= Ramp filter constant (see [explanation](#))  
 D2= Ramp filter form variable: 0...255

#### 4 - 28 BIAS-Command "Actual position X = const."

Act. posit. X = incr

##### Explanation:

The command "Actual position X =" presets the specified position value into the actual position defined.

Drive type:	Parameters:	Permissible inputs:	
All	X:	1 = internal position 2 = X40-input	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
	Y:	± 231 Increments	

##### Example:

0 Act posit. 1 = 0 INCR  
 1 Position = 10000 INCR  
 2 Speed = 1000 rpm  
 3 Acceleration = 2000 rpm/s  
 4 Deceleration = 2000 rpm/s  
 5 Start axis  
 6 Move position; axis no. = 0 , target = 0

##### Result:

The internal position is set to 0.  
 After the drive will move to the absolute position of 10000 increments.

##### Note:

- The actual position used for position control should only be changed at standstill, since this corresponds to a stop command without ramp.
- When using calculated units, changes in the position entered can occur since these positions always have to be rounded to an integral number of increments
- **637f:**  
This command has no effects in the mode "Multi axis". The specified actual position is not changed.

##### Command coding:

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

Parameters: D1 = 1 / 2  
 D2...D5 = Absolute position ±231 Increments

## 4 - 29 BIAS-Command "If actual position X ? const. then jump"

If actual pos X ? Y INKR then jump Z

### Explanation:

The command "If actual pos. X ? const." allows program branches by comparing the actual value X with the constant.  
If the condition is true the execution will be continued at the specified command line. Otherwise the next BIAS command line is executed.

Drive type:	Parameters:	Permissible inputs:	
All	X:	1: = internal position 2: = X40-input 3: = CAN-Bus	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	?	> greater than (0) < less than (1) == equal to (2) >= greater than or equal to (3) <= less than or equal to (4) != not equal to (5) > positive (6) < negative (7)	
	Y:	± 231 Increments	
	Z: Address:	0...1499 The maximum permissible jump address corresponds to the number of BIAS command lines in the program and corresponds to the number of the last BIAS command line. As an alternative to the line numbers labels (jump labels) can be used	

### Example:

```

0 Output 20 = 0
1 If actual pos. 1 < 1000 then jump 1
2 Output 20 = 1
LOOP:
3 If actual pos. 1 < 10000 then jump LOOP
4 Jump 4
    
```

### Result:

The output 20 is set to 1 in the position range of 1000 to 10000 increments.

### Note:

- The comparison of both values is carried out by a subtraction of the constant factor from the actual position.
- The conditional jumps depend only on the sign of the result for the conditions "difference positive" and "difference negative"

### Command coding:

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)		?

**Parameters:**

D1 =	Actual position	1,2,3 (see <a href="#">explanation</a> )
D2...D5 =	Absolute position	±231 Increments
D6,D7.0-D7.3 =	Jump address	0...1499
D7.4-D7.7 =	Comparison condition	0...7 (see <a href="#">explanation</a> )

**4 - 2A BIAS-Command “If actual position X ? [variable Y] then jump”**

If actual pos. X ? [variable Y] then jump Z

**Explanation:**

The command "If actual pos. X ? [variable Y]" allows program branches by comparing the actual value X with the contents of the specified variable.

If the condition is true the execution will be continued at the specified command line. Otherwise the next BIAS command line is executed.



<b>Drive type:</b>	<b>Parameters:</b>	<b>Permissible inputs:</b>	<input checked="" type="checkbox"/> BIAS program
All	<b>X:</b>	1: = internal position 2: = X40-input 3: = CAN-Bus	<input checked="" type="checkbox"/> PLC program
	<b>?</b>	> greater than (0) < less than (1) == equal to (2) >= greater than or equal to (3) <= less than or equal to (4) != not equal to (5) >- positive (6) <- negative (7)	<input checked="" type="checkbox"/> Mathematics
	<b>Y:</b>	0-255	
	<b>Z:Address:</b>	0...1499 The maximum permissible jump address corresponds to the number of BIAS command lines in the program and corresponds to the number of the last BIAS command line. As an alternative to the line numbers labels (jump labels) can be used.	

**Example:**

```
0 Variable 0 = 1000
1 Variable 1 = 10000
2 Output 20 = 0
3 If actual pos. 1 < [var. 0] then jump 3
4 Output 20 = 1
```

**Result:**

The output 20 is set to 0 in the position range of 1000 to 10000 increments.

**LOOP:**

```
5 If actual pos. 1 < [var. 1] then jump LOOP
6 Jump 2
```

**Note:**

**Command coding:**

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

**Parameters:** D1 = Actual position: 1,2,3 (see [explanation](#))  
 D2 = Comparison condition 0...255  
 D3 = Variable number 0...7 (see [explanation](#))  
 D4,D5 = Jump address 0...1499

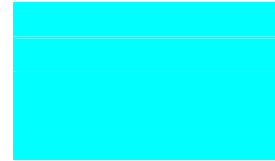
## 4 - 2B BIAS-Command "Sensor window = const."

Sensor; mode= X ; on = Y Increment.; off = Z Increment.

### Explanation:

The command "Sensor window = const." loads the mode and the window values for controlling the sensor functions.

The sensor control is only defined for positioning in a positive direction.



Drive type:	Parameters :	Permissible inputs:
All	<b>X: Mode</b>	<ul style="list-style-type: none"> <li><b>0</b> = Switch off the sensor functions.</li> <li><b>1</b> = Switch on sensor function in synchronous run with latch <a href="#">input 2</a>.</li> <li><b>2</b> = Customer specific special function.</li> <li><b>3</b> = Switch on the sensor function in synchronism with latch input 2.</li> <li><b>4</b> = Switch on the Quadraloc function without auto correction in synchronous run. Latch on real position 1.</li> <li><b>5</b> = Switch on the Quadraloc function with auto correction in synchronous run. Latch on real position 1.</li> <li><b>6</b> = Switch on the Quadraloc function without auto correction in synchronous run. Latch on real position 2.</li> <li><b>7</b> = Switch on the Quadraloc function with auto correction in synchronous run. Latch on real position 2.</li> <li><b>10</b> = Switch on sensor function with "remaining position =" for latch <a href="#">input 1</a></li> <li><b>11</b> = Switch on sensor function for latch input 1. The monitoring of the sensor window is deactivated.</li> <li><b>12</b> = Switch on interrupt function for latch input 2 The monitoring of the sensor window is deactivated.</li> <li><b>64</b> = Switch on interrupt function for latch input 1. The monitoring of the sensor window is deactivated</li> <li><b>65</b> = Switch on interrupt function for latch input 2. The monitoring of the sensor window is deactivated.</li> <li><b>128</b> = Switch on the interrupt function for latch input 1.</li> <li><b>254</b> = Switch off sensor function for latch input 1</li> <li><b>255</b> = Switch off sensor function for latch input 2</li> </ul>
	<b>Y:</b>	<b>224 Increments</b>
	<b>Z:</b>	<b>224 Increments</b>

- BIAS program
- PLC program
- Mathematics

**Example:** for mode 10:

```
0 Position = 10000 INCR
1 Remaining position = 2000 INCR
2 Sensor; mode=10 ;on=6000 Increment.; off =9000 Increment.
3 Start axis
4 Move position
```

**Result:**

The axis starts moving to the absolute position. Should a Low-High slope of the Latch input 1 (635/637/637+/637f X10.4, **631** X10.9) occur in the position range of 6000 to 9000 increments, 2000 increments will be added to the Latch position and moved to.

**Note:**

**Latch input 1:**

635/637/637+/637f input X10.4 configured for function 0,  
631 input X10.9 configured for function 5.

The latch status can be queried with the command "Variable [\[X\] = latch status 1](#)".

The latch position can be queried with the command "Variable [\[X\] = latch position 1](#)".

**Latch input 2:**

635/637/637+/637f input X10.25 configured for function 0,  
631 input X10.10 configured for function 5.

The monitoring of maximum correction length is activated.

See also "Sensor [adjustment 1](#)" und "Sensor [adjustment 2](#)"

The latch status can be queried with the command "Variable [\[X\] = latch status 2](#)".

The latch position can be queried with the command "Variable [\[X\] = latch position 2](#)".

- With synchronous applications the window values are specified relative to the beginning of the cycle.

If the value 0 is specified for the parameter sensor window off, the sensor window will not be turned off.

**Command coding:**

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

**Parameters:** D1 = Mode (see [explanation](#))  
D2-D4 = window on 224 Increments  
D5-D7 = window off 224 Increments

## 4 - 2C BIAS-Command "Sensor position = const."

**Explanation:**

The command "Sensor position = const." defines the starting position in which the sensor is expected.

The sensor position is required for the evaluation of the maximum correction in synchronous move.

Drive type:	Parameters	Permissible inputs:	
	:		<input checked="" type="checkbox"/> BIAS program
			<input checked="" type="checkbox"/> PLC program
All	const.	± 231	<input type="checkbox"/> Mathematics

**Example:**

**Result:**

**Note:**

- This command is only to be used in connection with the commands "Sensor [window , mode 1](#)", "Sensor [adjustment 1](#)" and "Sensor [adjustment 2](#)".
- The variable content is required for position in increments.

**Command coding:**

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

**Parameters:** D1- D4 = Position: ± 231

#### 4 - 2D BIAS-Command "Sensor adjustment 1 = const."

Sensoradjustment ; correct. = X increm.; average= Y

##### Explanation:

The command "Sensor adjustments 1 = const." defines the maximum permitted value for the correction of the position according to the latched sensor position in "sensor window; mode = 1".

The parameter average specifies the number of latch positions whose average values are to be evaluated for the position correction.

Drive type:	Parameters:	Permissible inputs:	
All	X: Correction: Y: Average:	224 Increments 0...255	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

Example:

Result:

##### Note:

- This command is only to be used in connection with the commands "Sensor [window , mode 1](#)", "Sensor [position](#)" and "Sensor [adjustment 2](#)".

##### Command coding:

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

Parameters: D1 - D3 = Correction: 224  
D4 = Average 0...255



## 4 - 2E BIAS-Command "Sensor adjustment 2 = const."

**Explanation:**

The command "Sensor adjustment 2 = const." defines the time within which the position correction is to be carried out.

Drive type:	Parameters:	Permissible inputs:	
All	const.	65535 in ms	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

**Example:**

**Result:**

**Note:**

- This command is only to be used in connection with the commands "Sensor [window, mode 1](#)", "Sensor [position](#)" and "Sensor [adjustment 1](#)".

**Command coding:**

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

**Parameters:** D1, D2 = Correction time: 0...65535

## 4 - 2F BIAS-Command "Update parameter"

### Explanation:

The command "Update parameter" allows the parameters: position, speed, acceleration, deceleration, and window to be updated during a move command in progress.

During a synchronous run in progress this command has no effect.

637f: With Multi-Axis Mode this command is not allowed!

Drive type:	Parameters:	Permissible inputs:	
All	none	---	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

### Example:

```

0   Position = 100000 Inkr.
1   Speed = 100 rpm
2   Start axis
3   Move position
4   Speed = 1000 rpm
5   If actual pos. 1 < 10000 Incr then jump 5
6   Update parameter
7   Wait for "position reached"
  
```

### Result:

The axis moves at a speed of 100 rpm to the position of 10000 increments then at a speed of 1000 rpm to the position of 100000 increments.

### Note:

- Each time, the last defined parameter values are updated.

### Command coding:

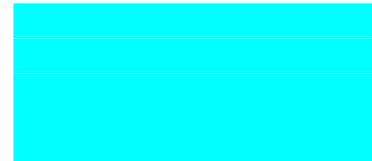
Code	D1	D2	D3	D4	D5	D6	D7
<b>2F</b>	00	00	00	00	00	00	00

### Parameters:

## 4 - 30 BIAS-Command "Position = [variable X]"

**Explanation:**

The command "Position = [variable X]" defines the setpoint position for the move commands from the contents of the variable X.  
The defined position is in effect until the next position command.



Drive type:	Parameters:	Permissible inputs:	
All	X:	0...255 permitted variable content ± 231 increments	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

**Example:**

```

0 [Variable 0] = 10000
1 Position = [variable 0]
2 Start Axis
3 Move in incremental position
4 Wait for "position reached"
5 Jump 2
  
```

**Result:**

The axis moves 10000 increments from the current actual position.

**Note:**

- When using move commands with direct position specification, the command will be overwritten.
- The content of the variable is interpreted as incremental position specification.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>30</b>	<b>X</b>	00	00	00	00	00	00

**Parameters:** D1 = Variable number: 0...255  
permissible variable content: ± 231

#### 4 - 31 BIAS-Command "Speed = [variable X]"

##### Explanation:

The command "Speed = [variable X]" defines the maximum speed for the move commands from the contents of the specified variable X.  
The defined speed remains in effect until the next speed command.

Drive type:	Parameters:	Permissible inputs:	
All	X:	0...255 permitted variable content: 1...12000	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

##### Example:

```

0 [Variable 0] = 1000
1 Speed = [variable 0]
2 Position = 10000 incr
3 Start axis
4 Move incremental position
5 Wait for "position reached"
6 Jump 3
  
```

##### Result:

The axis moves 10000 increments further relatively from the current actual position at 1000 rpm.

##### Note:

- When using move commands with direct speed specification, the command will be overwritten.
- The content of the variable is interpreted as a speed specification in rpm.
- **631/635/637**  
The maximum permissible speed depends on the configured resolver resolution.  
With a high resolver resolution only 4000 rpm is allowed.

##### Command coding:

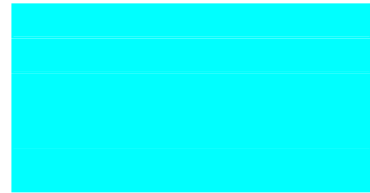
Code	D1	D2	D3	D4	D5	D6	D7
31	X	00	00	00	00	00	00

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

## 4 - 32 BIAS-Command "Acceleration = [variable X]"

**Explanation:**

The command "Acceleration = [variable X]" defines the acceleration ramps for the move commands from the contents of the specified variable X. The defined acceleration remains in effect until the next acceleration command.



Dirve type:	Parameters:	Permissible inputs:	
All	X:	0...255 permitted Variable content: 1...64000	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

**Explanation:**

- 0 [Variable 0] = 1000
- 1 **Acceleration = [variable 0]**
- 2 Speed = 1000
- 3 Position = 10000 INCR
- 4 Start axiss
- 5 Move incremental position
- 6 Wait for "position reached"
- 7 Jump 4

**Result:**

The axis moves 10000 increments further relatively from the current actual position at 1000 rpm and an acceleration of 10000 rpm/s..

**Note:**

- When using move commands with direct acceleration specification, the command will be overwritten.
- The content of the variable is interpreted as an acceleration specification in 5 rpm/s (variable x \* 5 = acceleration rate).
- The parameter acceleration is used by the change of speed from 0 rpm to a defined speed or by change of speed with the same sign.
  - The parameter acceleration is used by an active move command if the speed is decreased (without different sign).
  - The variation of the speed over 0 rpm, the parameter deceleration is used to 0 rpm and the parameter acceleration is used to the defined new speed.

With minimum acceleration to 4000rpm

Exp.:

a = 5 rpm/s

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

With maximum acceleration to 4000rpm

Exp.:

a = 32000 rpm/s

$$\implies t = \frac{4000\text{rpm}}{32000 \frac{\text{rpm}}{\text{s}}} = 0,0125\text{s} = 12,5\text{ms}$$

**Command coding:**

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

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

### 4 - 33 BIAS-Command "Deceleration = [variable X]"

#### Explanation:

The command "Deceleration = [variable X]" defines the deceleration ramps for the move commands from the contents of the specified variable X. The defined deceleration remains in effect until the next deceleration command.

Drive type:	Parameters:	Permissible inputs:	
All	X:	0...255 permitted variable content: 1...64000	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

#### Example:

```

0 [Variable 0] = 2000
1 Deceleration = [variable 0]
2 Acceleration = 5000 rpm/s
3 Speed = 1000
4 Position = 10000 INCR
5 Start axis
6 Move incremental position
7 Wait for "position reached"
8 Jump 4
  
```

#### Result:

The axis moves 10000 increments further relatively from the current actual position at 1000 rpm with an acceleration rate of 5000 rpm/s and a deceleration rate of 2000 rpm/s.

#### Note:

- When using move commands with direct deceleration specification, the command will be overwritten.
- The content of the variable is interpreted as a deceleration specification in 5 rpm/s (variable x \* 5 = deceleration rate).
- The parameter deceleration is used by the change of speed to 0 rpm
  - The parameter acceleration is used by an active move command if the speed is decreased (without different sign).
  - The variation of the speed over 0 rpm, the parameter deceleration is used to 0 rpm and the parameter acceleration is used to the defined new speed.

#### Command coding:

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

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

## 4 - 34 BIAS-Command "Gear factor = [variable X]"

**Explanation:**

The command "Gear factor = [variable X]" defines the gear factor for the move command "[Move synchron](#)" from the contents of the specified variable. It is the ratio between master and slave (the axis following the master) during a synchronous run.

With the command "[Synchron adjustment 2](#)" with mode 8 is possible to select the resolution of the gear factor (value 0, 1 or 2).

Drive type:	Parameters:	Permissible inputs	
All	X:	0...255	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
		permitted variable content see: <a href="#">Synchron adjustment 2</a>	

**Example: 635/637/637+/637f**  
 0 [Variable 0] = 256  
 1 **Gear factor = [variable 0]**  
 2 Move synchron  
 3 If input 11 == 1 then jump 3  
 4 Stop axis ; mode 1

**Result:**  
 The axis moves synchronously until the start input 11 drops.  
 At the same time 1 increment is traversed per 1 increment (master increment) counted in on the X40 connection.

**Example: 631**  
 0 [Variable 0] = 256  
 1 **Gear factor = [variable 0]**  
 2 Move synchron  
 3 If input 11 == 9 then jump 3  
 4 Stop axis ; mode 1

**Result:**  
 The axis moves synchronously until the start input 9 drops.  
 At the same time 1 increment is traversed per 1 increment (master increment) counted in on the X40 connection.

**Note:**

- The parameter "gear factor" must be defined before the command "move synchron".
- The content of the variable is interpreted as gear factor \* 256.
- Changes of the gear factor during the synchronous run will be activated immediately (without the command "[Update parameter](#)").

**Command coding:**

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

**Parameters:** D1 = Variable number: 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-Command "Remaining position = [variable X]"

### Explanation:

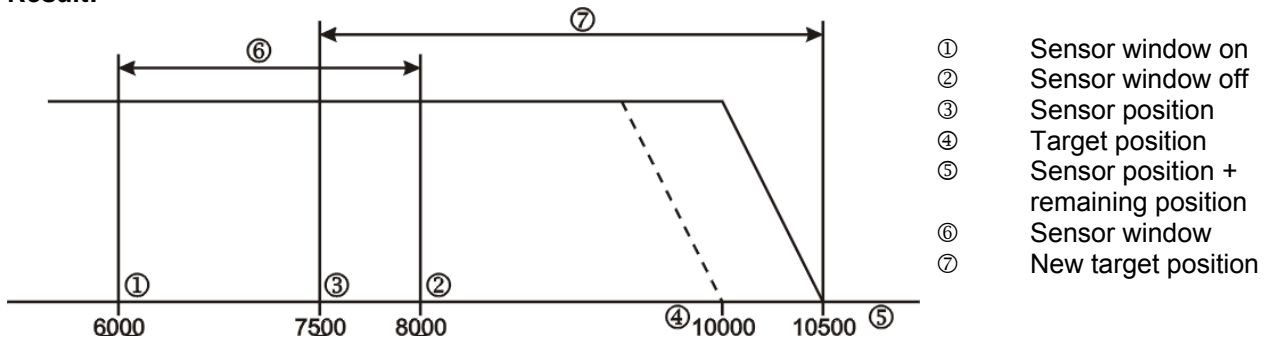
The command "Remaining position = [variable X]" defines the variable where the remaining position is stored, which is added together with the latch value measured, when positioning with a sensor.  
The defined remaining position remains in effect until the next remaining position command.

Drive type:	Parameters:	Permissible Inputs	
All	X:	0...255 permitted variable content: ± 231 Increments	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

### Example:

- 0 Actual position 1 = 0 INCR
- 1 Position = 10000 INCR
- 2 Sensor; mode= 10 on= 6000 Incr off= 8000 Incr
- 3 [Variable 0] = 3000
- 4 **Remaining position = [variable 0]**
- 5 Move position
- 6 Wait for "position reached"
- 7 Jump 0

### Result:



The axis starts to move to the absolute position of 10000 increments.  
If a Low-High slope on the sensor X10.4 (X10.9 on 631) is evaluated between the positions of 6000 and 8000 increments then the remaining position is added to the latch position and moved to.  
With an assumed sensor position of 7500 increments, the new target position is: 7500 increments + remaining position 3000 increments = 10500 increments

### Note:

- This command is only to be used in combination with the command "Sensor [window; mode =; mode 10](#)".

### Command coding:

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

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

### 4 - 37 BIAS-Command "Maximum current = [variable X]"

Maximum current = [variable X ], mode = Y

#### Explanation:

The command "Maximum current = [variable X]" defines the maximum permitted current of the drive and thus allows a variable torque limiting of the axis.

Drive type:	Parameters:	Permissible inputs:	
	X:	0...255	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Y:	0: The content of the variable is evaluated in 1/10 amps. 1: The content of the variable is evaluated in D/A-steps.	
		<b>Permitted variable content:</b> (2 x maximum current of the drive) x 10	
631	Mode 0:	0 - 6061 = 100% maximum current of the drive. 0 - 12123 = 100% maximum current of the drive.	
635/637/637+/637 f	Mode 1:		

#### Example:

```

0 [Variable 10] = 10
1 [Variable 11] = 100
2 Position = 10000 INCR
3 Maximum current = [variable 11], mode = 0
4 Start axis
5 Move position
6 If actual position 1 < 7500 INCR then jump 6
7 Maximum current = [variable 10], mode = 0
8 Wait for "position reached"
9 Jump 3
  
```

#### Result:

The position of 10000 increments is moved to. In the range of 0 to 7500 increments the maximum current is limited to 10 A rms. In the range of 7500 increments to 10000 increments the maximum current is limited to 1 A rms.

#### Note:

The specified maximum current is only taken into account if there are no other smaller limitations to the maximum current. Possibilities are:

- The external current limitation with analog input 2 is active (not 631 !).
- The function current limitation at warning is activated.
- The maximum current is limited in the menu "commissioning motor"

#### Command coding:

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

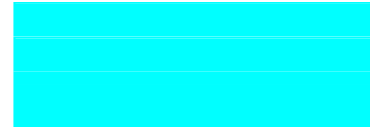
**Parameters:** D1 = Variable number: 0...255  
D2 = Mode 1/ 2

## 4 - 38 BIAS-Command "Actual position X = [variable Y]"

Actual position X = [variable Y]

**Explanation:**

The command "Actual position X = [variable Y]" loads the contents of the specified variable into the actual position defined (counter preset).



Drive type:	Parameters:	Permissible inputs:	
All	X:	1 = internal position 2 = X40-input	<input checked="" type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
	Y:	0...255	

**Example:**

- 0 [Variable 0] = 0
- 1 **Actual position 1 = [variable 0]**
- 2 Position = 10000 INCR
- 3 Speed = 1000 rpm
- 4 Acceleration = 2000 rpm/s
- 5 Deceleration = 2000 rpm/s
- 6 Start axis
- 7 Move position; axis no. = 0 , target = 0

**Result:**

The internal position is set to 0 increments. After the drive will move to the absolute position of 10000 increments.

**Note:**

- The actual position used for position control should only be changed at standstill, since this corresponds to a stop command without ramp.
- This command has no effects in the mode "Multi axis". The specified actual position is not changed.

**Command coding:**

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

**Parameters:** D1 = Mode 1/ 2  
D2 = Variable number: 0...255

#### 4 - 39 BIAS-Command “Analog output X = [variable Y]“

Analog output X = [variable Y ]

##### Explanation:

The command "analog output X = [variable Y]" loads the contents of the specified variable into the analog output defined.

Drive type:	Parameters:	Permissible inputs:	
631: 635/ 637/ 637+/- 637f	none X:  Y:	This command is not available on the 631 drive 1 = MP 1 (X10.17) 2 = MP 2 (X10.6)  0...255	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
		Permitted variable content	
635/ 637:	X = 1/ 2	-64...+63 corresponding to: -10...+10V	
637+/-637f:	X = 1 MP1  X = 2 MP2	-512...+511 corresponding to: -10...+10V -128...+127 corresponding to: -10...+10V	

##### Example: 635/ 637:

0 [Variable 0] = -64  
1 Analog output 1 = [Variable 0]

##### Result:

An analog voltage of -10V is sent out on the MP 1 (X10.17).

##### Note:

- In order to be able to use the analog output function 1 (BIAS) it must be selected in the configuration for the respective analog output.

##### Command coding:

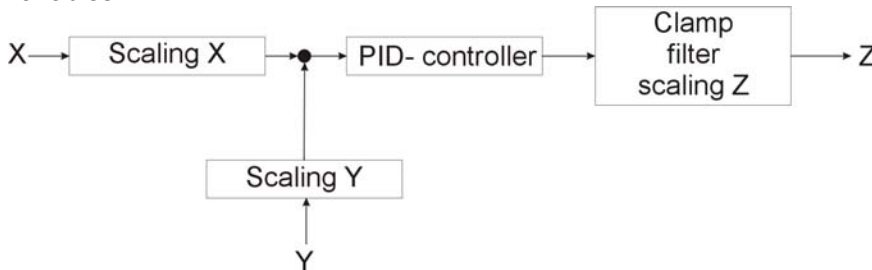
Code	D1	D2	D3	D4	D5	D6	D7
39	X	Y	00	00	00	00	00

Parameters: D1 = MP 1/ 2  
D2 = Variable number: 0...255

## 4 - 3A BIAS-Command "PID scaling"

**Explanation:**

The command "PID scaling" defines the variable area for the scaling parameters of the PID-controller. The variable area starts with the specified variable and has a length of 6 variables.



Drive type:	Parameters:	Explanation::	Permitted variable content
All	n	scaling X numerator	± 32767
	n+1	scaling X denominator	± 32767
	n+2	scaling Y numerator	± 32767
	n+3	scaling Y denominator	± 32767
	n+4	scaling Z numerator	± 32767
	n+5	scaling Z denominator	± 32767

- BIAS program
- PLC program
- Mathematics

**Example :**

- 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 scaling; start = [variable 5]**

**Result:**

The command PID scaling scales the set point value with a factor of 1.25, the actual value with 0.9 and the PID output (adjustment value) with 0.5.

**Note:**

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

**Command coding:**

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

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

**4 - 3B BIAS-Command “Sensor window = [variable X]“**

Sensor; mode = [var. X ]; on = [var. Y ]; off = [var. Z ]

**Explanation:**

The command "Sensor window = [variable X]" loads the contents of the specified variables into the mode and the window values for controlling the sensor functions. The sensor control is **only defined for** positionings in a **positive direction**. If a sensor impuls occurs the selected function is deactivated automatical.

Drive type:	Parameters:	Permissible inputs:	
All	X: Mode	<b>0-255</b> <b>permitted variable content:</b> <b>0</b> = Switch off the sensor functions. <b>1</b> = Switch on sensor function in synchronous run with latch <a href="#">input 2</a> . <b>2</b> = Customer specific special function. <b>3</b> = Switch on the sensor function in synchronism with latch input 2 on actual position 1. <b>4</b> = Switch on the Quadraloc function without auto correction in synchronous run. Latch on actual position 1. <b>5</b> = Switch on the Quadraloc function with auto correction in synchronous run. Latch on actual position 1. <b>6</b> = Switch on the Quadraloc function without auto correction in synchronous run. Latch on actual position 2. <b>7</b> = Switch on the Quadraloc function with auto correction in synchronous run. Latch on actual position 2. <b>10</b> = Switch on sensor function with “remaining position =” for latch <a href="#">input 1</a> on actual position 1 <b>11</b> = Switch on sensor function for latch input 1 on actual position 1. The monitoring of the sensor window is deactivated. <b>12</b> = Switch on interrupt function for latch input 2 on actual position 1. The monitoring of the sensor window is deactivated. <b>64</b> = Switch on interrupt function for latch input 1 on actual position 2. The monitoring of the sensor window is deactivated <b>65</b> = Switch on interrupt function for latch input 2 on actual position 2. The monitoring of the sensor window is deactivated. <b>128</b> = Switch on the interrupt function for latch input 1. <b>254</b> = Switch off sensor function for latch input 2. <b>255</b> = Switch off sensor function for latch input 1.	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Y:	<b>0-255</b> <b>permitted variable content 224 increments</b>	
	Z:	<b>0-255</b> <b>permitted variable content 224 increments</b>	

**Example:** for mode 10:

- 0 Variable 0= 10
- 1 Variable 1= 6000
- 2 Variable 2= 9000

**Result:**

The axis starts moving to the absolute position. Should a Low-High slope of the

- |  |  |
|--|--|
| <p>3 Position = 10000 INCR<br/>         4 Remaining position = 2000 INCR<br/>         5 <b>Sensor; mode=[var. 0]; on=[var. 1]; off=[var. 2]</b><br/>         6 Start axis<br/>         7 Move position</p> | <p>Latch input 1 occur in the position range of 6000 to 9000 increments, 2000 increments will be added to the latch position and moved to.</p> |
|--|--|

**Note:**

**Latch input 1:**

635/637/637+/637f input X10.4 configured for function 0,  
 631 input X10.9 configured for function 5.  
 The latch status can be queried with the command "Variable [\[X\] = latch status 1](#)".  
 The latch position can be queried with the command "Variable [\[X\] = latch position 1](#)".

**Latch input 2:**

635/637/637+/637f input X10.25 configured for function 0,  
 631 input X10.10 configured for function 5.  
 The monitoring of maximum correction length is activated.  
  
 See also "Sensor [adjustment 1](#)" und "Sensor [adjustment 2](#)"  
 The latch status can be queried with the command "Variable [\[X\] = latch status 2](#)".  
 The latch position can be queried with the command "Variable [\[X\] = latch position 2](#)".

- With synchronous applications the window values are specified relative to the beginning of the cycle.
- If the value 0 is specified for the parameter sensor window off, the sensor window will not be turned off.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>3B</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	00	00	00	00

**Parameters:** D1 = Mode (see [explanation](#))  
 D2 = Variable 0-255: window on 224 increments  
 D3 = Variable 0-255 window off 224 increments

#### 4 - 3C BIAS-Command "Sensor position = [variable X]"

##### Explanation:

The command "Sensor position = [variable X]" defines the starting position in which the sensor is expected.

The sensor position is required for the evaluation of the maximum correction in synchronous move.

Drive type	Parameters	Permissible inputs:	
	:		<input checked="" type="checkbox"/> BIAS program
			<input checked="" type="checkbox"/> PLC program
			<input type="checkbox"/> Mathematics
All:	X: Mode	0...255	

##### Example:

##### Result:

##### Note:

1. This command is only to be used in connection with the commands "Sensor [window , mode 1](#)", "Sensor [adjustment 1](#)" and "Sensor [adjustment 2](#)".
2. The variable content is required for position in increments.

##### Command coding:

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

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



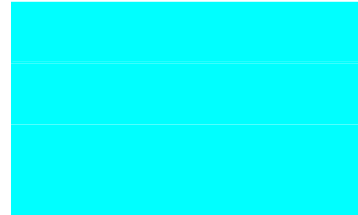
## 4 - 3D BIAS-Command "Sensor adjustments 1 = [variable X]"

Sensoradjustment; correction = [var. X ]; average= [var. Y]

### Explanation::

The command "Sensor adjustments 1= [variable X]" defines the maximum permitted value for the correction of the position according to the latched sensor position in "sensor window; mode = 1".

The parameter average specifies the number of latch positions whose average values are to be evaluated for the position correction.



Drive type:	Parameters:	Permissible inputs:	
All	X:  Y:	0...255 permitted variable content: ± 224 increments 0...255 permitted variable content: 0...255	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

### Example:

### Result:

### Note:

- This command is only to be used in connection with the commands "Sensor window, mode 1 = [variable X]"; "Sensor position = [variable X]" and "Sensor adjustment 2 = [variable X]".

### Command coding:

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

Parameters: D1 = Variable number: 0...255  
D2 = Variable number: 0...255

### 4 - 3 BIAS-Command "Sensor adjustments 2 = [variable X]"

#### Explanation:

The command "Sensor adjustment 2 = [Variable X]" defines the time within which the position correction is to be carried out.

Drive type:	Parameters:	Permissible inputs;	
All	X:	0...255 permitted variable content: 65535 in ms	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

Example:

Result:

#### Note:

- This command is only to be used in connection with the commands "Sensor [window, mode 1 = \[variable X\]](#)"; "Sensor [position = \[variable X\]](#)" and "Sensor [adjustment 1 = \[variable X\]](#)"

#### Command coding:

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

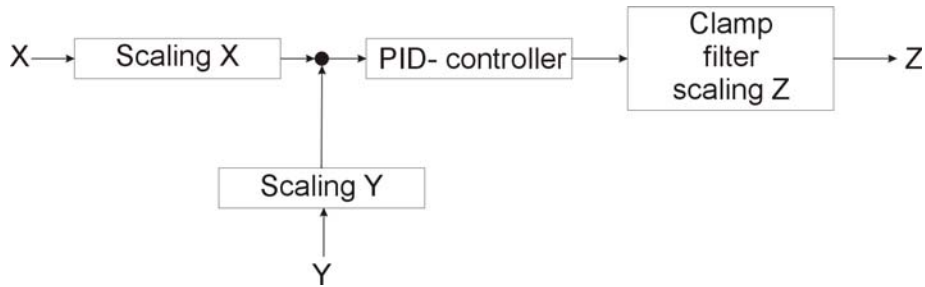
**Parameters:** D1 = Variable number: 0...255

## 4 - 3F BIAS-Command "PID parameter"

### Explanation:

The command "PID parameter" defines the variable area for the parameters of the PID-controller.

The variable area starts with the specified variable and has a length of 6 variables.



Drive type:	Parameters:	Permissible inputs:	Meaning	Permitted variable content
631/ 635/ 637	n n+1 n+2 n+3 n+4 n+5	0...249	P-gain Integration time Deviation time Filter time constant Clamp maximum Clamp minimum	0..32767 * 1/4096 0..32767 * 1,899ms 0..32767 * 1,899ms 0..32767 * 1,899ms ± 32767 ± 32767
637+/ 637f	n n+1 n+2 n+3 n+4 n+5	0...249	P-gain Integration time Deviation time Filter time constant Clamp maximum Clamp minimum	0..32767 * 1/4096 0..32767 * 0,844ms 0..32767 * 0,844ms 0..32767 * 0,844ms ± 32767 ± 32767

- BIAS program
- PLC program
- Mathematics

Example:

Result:

Note:

Command coding:

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

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

#### 4 - 40 BIAS-Command “[Variable X] = position“

##### Explanation:

The command "[Variable X] = position" loads the current position into the specified variable X.

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

##### Example:

0 [Variable 0] = position

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

2 Position = [variable 0]

3 Update parameter

##### Result:

The actual set point is increased by 1000 increments and is activated.

##### Note:

- The position is a value in increments

##### Command coding:

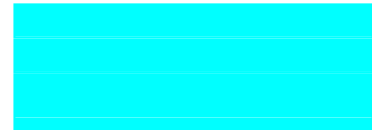
Code	D1	D2	D3	D4	D5	D6	D7
40	X	00	00	00	00	00	00

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

## 4 - 41 BIAS-Command “[Variable X] = speed“

**Explanation:**

The command “[Variable X] = speed” loads the current speed into the specified variable X.



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

**Example:**

```

0 [Variable 0] = speed
1 [Variable 0] = [variable 0] + 100
2 Speed = [variable 0]
3 Update parameter
    
```

**Result:**

The actual current speed is increased by 100rpm and is activated.

**Note:**

- The setpoint speed is a value in rpm.
- **631/ 635/ 637**  
The maximum permissible speed is subject of the configuration of the resolver resolution. If the resolver resolution is configured to high resolution the maximum permissible speed is 4000rpm.

**Command coding:**

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

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

#### 4 - 42 BIAS-Command “[Variable X] = acceleration”

##### Explanation:

The command “[Variable X] = acceleration” loads the current acceleration into the specified variable X.

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

##### Example:

**0** [Variable 0] = acceleration

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

2 Acceleration = [variable 0]

3 Update parameter

##### Result:

The current defined acceleration is increased by 500 rpm/s and activated.

##### Note:

- The acceleration is a value in 5 rpm/s.

##### Command coding:

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

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

## 4 - 43 BIAS-Command “[Variable X] = deceleration”

### Explanation:

The command "[Variable X] = deceleration" loads the current deceleration into the specified variable X.

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

### Example:

**0 [Variable 0] = deceleration**

- 1 [Variable 0] = [variable 0] + 120
- 2 Deceleration = [variable 0]
- 3 Update parameter

### Result:

The current defined deceleration is increased by 600 rpm/s and activated.

### Note:

- The deceleration is a value in 5 rpm/s.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>43</b>	<b>X</b>	00	00	00	00	00	00

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

#### 4 - 44 BIAS-Command "[Variable X] = gear factor"

##### Explanation:

The command "[Variable X] = gear factor" loads the current gear factor into the specified variable X.

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

##### Example synchronous setting 2, mode 8, value 0 (standard):

```

0   [Variable 1] = actual pos 1
1   [Variable 0] = gear factor
2   [Variable 0] = [variable 0] - 1
3   [Variable 1] = [variable 1] + 4096
4   If actual pos 1 < [variable 1] then jump 4
5   Gear factor = [variable 0]
6   If actual pos 1 < 655360 then jump 2

```

##### Result:

Up to the slave position of 655360 increments, every 4096 increments the gear factor is decreased by 1/256.

##### Note:

- The content of the variable is defined in the commands "Gear [factor = \[variable X\]](#)" and "Gear [factor =](#)" und "Synchronous [settings 2](#)"

##### Command coding:

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

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



## 4 - 45 BIAS-Command “[Variable X] = block number”

### Explanation:

The command "[Variable X] = block number" loads the current block number of the BIAS process into the specified variable X.

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

### Example:

```

0   ...
1   ...
2   [Variable 1] = block number
3   ...
  
```

### Result:

The block number 2 is stored in variable 1.

### Note:

- This command allows storing of the program steps.
- see also: command "BIAS [execution pointer = \[variable X\]](#)".

### Command coding:

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

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

#### 4 - 46 BIAS-Command “[Variable X] = actual position Y”

##### Explanation:

The command “[Variable X] = actual position Y” loads the value of the defined actual position Y into the specified variable X.

Drive type:	Parameters:	Permissible inputs:	
All	X: Y:	0...255 1: internal position (X30) 2: X40 input 3: CAN-Bus absolute encoder	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
only 637f:		3: CAN2 PDO1 SSI – reference value 4: CAN2 PDO3 reference value 5: CAN2 PDO4 reference value	

##### Example for 635/ 637/ 637+/ 637f:

```

0   [Variable 1] = actual pos. 1
1   [Variable 1] = [variable 1] + 4096
2   Output 13 = 1
3   If actual pos. 1 < [variable 1] then jump 3
4   Output 13 = 0

```

##### Result:

The BIAS output 13 is set as of the current position and reset 4096 increments later.

##### Note:

- The position is a value in increments.

##### Command coding:

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

Parameters: D1 = Variable number: 0...255  
D2 = Actual position: 1...5 (see [explanation](#))

## 4 - 47 BIAS-Command “[Variable X] = analog input Y”

### Explanation:

The command “[Variable X] = analog input Y” loads the value of the defined analog input Y into the specified variable.

Drive type:	Parameters:	Permissible inputs:	
631	X: Y:	<b>0...255</b> 1 = X10.1-X10.2 (setpoint input)	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
635/ 637/ 637+/- 637f		1 = X10.5- X10.18 (setpoint input) 2 = X10.10- X10.19 (external current limitation)	

### Example:

```

0 [Variable 0] = analog input 2
1 [Variable 0] = [variable 0] / 2
2 [Variable 0] = [variable 0]* -1
3 [Variable 1] = [variable 0] + 2048
4 Speed = [variable 1]
5 Update parameter
6 Jump 0

```

### Result:

The axis moves corresponding to the analog voltage on analog input 2.

With the setpoint speed as follows:

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

### Note:

- The value range of the **analog input 1** is 2047Digits (±11 Bit; ±10V)  
The 637f use only 12bit (real 14Bit)!
- The value range of the **analog input 2** is ± 4095 Digits.  
(Real only ±9 Bit; ±10V)

### Command coding:

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

**Parameters:** D1 = Variable number: 0...255  
D2 = Analog input (see [explanation](#))

#### 4 - 48 BIAS-Command “[Variable X] = latch position Y”

##### Explanation:

The command “[Variable X] = latch position Y” loads the value of the defined latch position Y into the specified variable X.

Drive type:	Parameters:	Permissible inputs:	
631:	X: Y:	<b>0...255</b> 1 = Latch value X10.9 2 = Latch value X10.10	<input checked="" type="checkbox"/> BIAS program
635/ 637/ 637+/- 637f:		1 = Latch value X10.4 2 = Latch value X10.25	<input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics

##### Example:

```
0 [Variable 0] = latch status 1
1 If [variable 0] <= 0 then jump 0
2 [Variable 1] = latch position 1
```

##### Result:

The actual position at the latch pulse is stored in the variable 1

##### Note:

- This command can be used in connection with the command "Sensor [window; mode X](#)" to be used.
- The latch position is only current, if a value > 0 is returned with the command “[Variable [X](#)] = [latch status Y](#)”
- The latch position is a value in increments.

##### Command coding:

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

**Parameters:** D1 = Variable number: 0...255  
D2 = Latch position 1/2 (see [explanation](#))

## 4 - 49 BIAS-Command “[Variable X] = actual speed Y“

**Explanation:**

The command "[Variable X] = actual speed Y" loads the value of the defined speed Y into the specified variable X.

Drive type:	Parameters:	Permissible inputs:	
All	X: Y:	0...255  1 = internal actual position (X30) 2 = X40-input	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
only 637f		3 = SSI-actual position 3 actual speed	

**Example 635/ 637/ 637+/ 637f:**

```

0      [Variable 0] = actual speed 2
1      If [variable 0] >= 0 then jump 3
2      [Variable 0] = [variable 0]* -1
3      If [variable 0] >= 100 then jump 6
4      Output 13 = 1
5      Jump 0
6      Output 13 = 0
7      Jump 0
    
```

**Result:**

The output X10.13 is set as well as the speed calculated from the increment coming in on the X40 counter input is less than 100 rpm.

**Note:**

- The speed is a value in rpm and is marked with a plus or minus sign.
- For speed 2 the correct increment resolution of the external encoder must be specified in "Commissioning", "Counter", "X40 resolution".

**Command coding:**

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

**Parameters:** D1 = Variable number: 0...255  
D2 = Actual speed 1,2,3 (see [explanation](#))

#### 4 - 4A BIAS-Command "[Variable X] = latch status Y"

##### Explanation:

The command "[Variable X] = latch status Y" loads the status of the defined latch input Y into the specified variable X.

Drive type:	Parameters:	Permissible inputs:	
631:	X: Y:	0...255 1 = Latch value X10.9 2 = Latch value X10.10	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
635/ 637/ 637+/- 637f:		1 = Latch value X10.4 2 = Latch value X10.25	

##### Status:

0	Latch not yet appeared
-1	No latch appeared within the sensor window
1	Latch executed (mode 10)
2	Latch executed (mode 11)
12	Latch executed (mode 12)
64	Latch executed (mode 64)
65	Latch executed (mode 65)

##### Example:

```

0 [Variable 0] = latch status 1
1 If [variable 0] <= 0 then jump 0
2 [Variable 1] = latch position 1

```

##### Result:

The actual position at the latch pulse is stored in the variable 1.

##### Note:

- This command can be used in connection with the command "Sensor [window; mode X](#)" zu benutzen.
- The latch position is a value in increments.

##### Command coding:

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

Parameters: D1 = Variable number: 0...255  
D2 = Latch status 1/2 (see [explanation](#))

## 4 - 4B BIAS-Command "[Variable X] = position Y"

[Variable X ] = position Y ; axis no. Z

### Explanation:

The command "[Variable X] = position Y" stores the content of the defined position into the specified variable X.

Drive type:	Parameters:	Permissible inputs:	
631/ 635/ 637/ 637+/ 637f	X:  Y: Position value	0...255  1 = actual setpoint position 2 = internal target position	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
637f with Multi-Axis- Function: (Mode 8 = 1)	Y: Position value  Z: Axis number	1 = actual setpoint position (axis Z) 2 = internal target position (axis Z) 3 = Brake distance (axis Z) 4 = Profile cycle length (axis Z)  1-3	

### Example for 635/637/637+:

- 0 [Variable 0] = position 2
- 1 [Variable 0] = [variable 0] – 2000
- 2 If actual pos. 1 < variable 0] then jump 2
- 3 Output 13 = 1

### Result:

The output 13 is set 2000 increments before reaching the target position.

### Note:

1. The position is a value in increments.
2. The actual setpoint position represents the position course during a positioning. It gives, therefore, the respective setpoint position at a certain point in time.
3. The internal target position specifies the target of a positioning as an absolute value.
4. The braking distance specifies the necessary path for the braking ramp in increments.

### Command coding:

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

**Parameters:** D1 = Variable number: 0...255  
 D2 = Position value (see [explanation](#))  
 D3 = Axis number: 1...3

#### 4 - 4C BIAS-Command “[Variable X] = value Y“

##### Explanation:

The command “[Variable X] = value Y” loads the defined value in the specified variable X.

Drive type:	Parameters:	Permissible inputs:	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
All:	X:  Y:	0...255  0 = Actual <a href="#">current with sign in 1/10 Amps</a> 1 = Actual <a href="#">effective current without sign in 1/10 Amps</a> 2 = Field <a href="#">bus status</a> 3 = Actual speed in rpm 4 = Position deviation of the position controller (trail error) 5 = Setpoint of the PID-function 6 = Actual value of the PID-function 7 = Actual deviation of the PID-function 8 = Actual integral value of the PID-function 9 = Output value multiplied by 256 of the PID-function  10 = Time <a href="#">value (in 2 ms steps)</a> 11 = actual <a href="#">maximum current in 1/10 Amps</a> 12 = actual <a href="#">maximum current in D/A Steps</a> 13 = actual <a href="#">effective current in D/A Steps</a> 14 = Ucc <a href="#">in Volts</a> 15 = IBT-Mask <a href="#">number</a> 16 = Drive <a href="#">maximum current in 1/10 Amps</a> 17 = Synchronism <a href="#">speed</a> 18 = Fieldbus <a href="#">node number</a> 19 = Input-/output <a href="#">status of X120</a> 20 = I <sup>2</sup> t <a href="#">Motor</a> 21 = I <sup>2</sup> t <a href="#">Drive</a> 22 = CAN2-Bus-node <a href="#">monitoring state</a> 23 = Rotor <a href="#">postion at positive slope of the reference sensor</a> (from Firmware V8.20 only 637f)	

##### Value = 0:

loads the actual current with sign in 1/10 Amps in the selected variable.

##### Example:

The command is executed with a drive 635-05 (5 Amps nominal current)

Variable 2 = value 0

##### Result:

The variable 2 is loaded with the actual current with sign in 1/10 Amps.

The drive is producing -1 amp of current; the variable 2 has the content -10.



Value = 1:

loads the actual current without sign in 1/10 Amps in the selected variable

**Example:**

The command is executed with a drive 635-05 (5 Amps nominal current).

0 Variable 2 = value 1

**Result:**

The variable 2 is loaded with the actual current without sign in 1/10 Amps.  
The drive is producing -1 amp of current; the variable 2 has the content 10

Value = 2:

loads the following 4-byte status information in the selected variable.

**Example:**

0 Variable 2 = value 2

**Result:**

The variable 2 is loaded with the actual status information.

**Byte 0**

In- [output status 631](#) Input status low byte: 635/637 637+/637f

**Byte 1 and Byte 2**

Error word 631 635/637 637f

**Byte 3**

Status word 2 low Byte 631 635/637 637f

Value = 10:

loads the actual timer value in the selected variable

**Note:** 631/ 635 /637

Der The internal timer value is increased every 2 ms by 1.

## 637+ / 637f

Der The internal timer value is increased every 0,844 ms by 1.

With this function it is possible to define free running timers in the BIAS - processing

**Example** for 631/ 635 /637:

ProgStart:

```

0   Actual position 1 = 0      INKR
1   Start axis
2   Move positive; v= 100 rpm , a+= 5000 rpm/s , a-= 5000 rpm/s
3   [Variable 0 ] = value 10
4   [Variable 1 ] = [variable 0 ] + 2500
WAIT:
5   If actual position 1 > 100000   INKR then jump STOP
6   [Variable 0 ] = value 10
7   If [Variable 0 ] <= [variable 1 ] then jump WAIT
STOP:
8   Stop axis; mode = 1
9   NOP
10  NOP

```

**Result:**

The drive should be stopped after reaching the actual position > 10000 or after 5 seconds.

Value = 11:

loads the actual permitted maximum current in 1/10 Amps in the selected variable.

**Note:** The maximum current can be influenced with the following functions:

- a. The external current limitation with analog input 2 is active (not 631 !).
- b. The function current limitation at warning activated.
- c. The maximum current in the menu "commissioning motor"
- d. The maximum current defined with the BIAS-command "Maximum current = variable"

Gener The lowest value is valid

**Example:**

The command is executed with a drive 635-05 (5 Amps nominal current) and in "commissioning motor" the maximum current is limited to 50 %.

**Result:**

The variable 2 is loaded with the actual valid maximum current in 1/10 Amps.

The variable 2 has the content 25.

**0 Variable 2 = value 11**

Value = 12:

loads the actual permitted maximum current with the internal notation in the selected variable

**Note:** The maximum current can be influenced with the following functions:

- a. The external current limitation with analog input 2 is active (not 631 !).
- b. The function current limitation at warning is activated.
- c. The maximum current in the menu "commissioning motor" is limited.
- d. The maximum current defined with the BIAS-command "Maximal current = [variable X]".

The lowest value is valid!

**Note:** 631                                    The maximum current of the drive: 100% has the internal value 6060.

635/637/637+/637f:                    The maximum current of the drive: 100% has the internal value 12123.

**Example:**

The command is executed with a drive 635-05 (5 Amps nominal current) and in "commissioning motor" the maximum current is limited to 50 %.

**Result:**

The variable 2 is loaded with the actual valid maximum current with the internal notation. The variable 2 has the content 6060.

**0        Variable 2 = value 12**

Value = 13:

loads the actual measured current with the internal notation in the selected variable..

**Note:** 631:                                    The maximum current of the drive: 100% has the internal value 6060.

635/637/637+/637f:                    The maximum current of the drive: 100% has the internal value 12123.

**Example:**

The command is executed with a drive 635-05 (5 Amps nominal current).

**Result:**

The variable 2 is loaded with amount of the actual measured current with the internal notation. The drive is producing 5 amps of current; the variable 2 has the content 6060.

**0        Variable 2 = value 13**

Value = 14:

loads the actual DC-Bus voltage in the selected variable.

**Example:**

The DC-Bus voltage is 322 V

**0 Variable 2 = value 14**

**Result:**

The variable 2 is loaded with the value of the DC-Bus voltage.

The variable 2 has the content of 322.

Value = 15:

loads the actual mask number of the IBT in the selected variable.

**Note:**

This command is only valid when you use an IBT with CAN-Bus. With this information it is possible to decide the change of the display mask with the BIAS-command "IBT-Mask [number](#)".

**Example:**

The display of the IBT shows the main mask (4).

**0 Variable 2 = value 15**

**Result:**

The variable 2 is loaded with the actual mask number.

The variable 2 has the content of 4.

Value = 16:

loads the maximum current of the drive in 1/10 Amps in the selected variable.

**Example:**

The command is calculated with the drive 635-05 (5 Amps nominal current → 10 Amps maximum current).

**0 Variable 2 = value 16**

**Result:**

The variable 2 is loaded with the maximum current of the drive in 1/10 Amps.

The variable 2 has the content of 100.

Value = 17:

Loads the actual measured synchronous speed ( $\Delta$ increments) in the specified variable.  
This speed is not valid in the multi axis synchronous mode.

$$v = \frac{\Delta \text{Incr} \times 60 \frac{\text{s}}{\text{min}}}{65535 \text{Incr} \times \text{sampling time position control}}$$

sampling time position control:

637+, 637f: 0,844ms

631, 635, 637: 1,899ms

**Example 637+ / 637f:**

The synchronous speed is 1000rpm

**0 Variable 2 = value 17**

**Result:**

The variable 2 is loaded by the value of the Synchronous speed.

The variable 2 has the value 921.

Value = 18:

loads the actual node or axis number in the specified variable.

**Example:**

The defined node number is 3.

**0 Variable 2 = value 18**

**Result:**

The variable 2 is loaded by the value of the node number.

The variable 2 has the value 3.

Value = 19:

Loads the actual input/ output status from clamp X120 in the specified variable.

- This command is available with the 637f as of firmware V8.19.

Bit	Meaning
0	Input 1.
1	Input 2.
2	Input 3.
3	Input 4.
4	Output 1.
5	Output 2.
6	Output 3.
7	Output 4.
8	Cam 1.
9	Cam 2.
10	Cam 3.
11	Cam 4.
12	Reserve.
13	Reserve.
14	cam-operated switchgroup On.
15	Cam Backstop.

**Example:**

The inputs 2 and 3, and the outputs 1 and 4 have High- level

**0 Variable 2 = value 19**

**Result:**

The variable 2 is loaded with the value by the input/ output status.  
The variable 2 has the content 150. (96hex).

Value = 20:

load the actual motor performance in the specified variable.

The 100% of the performance is normalized to.  $I_{\text{Nominal}}^2 \cdot \sqrt{2}$

**Example:**

The motor performance is 100%.  
The command is executed with a motor  
AC M2n0290-4-/1-3 (6 Amps nominal current)

**0 Variable 2 = value 20**

**Result:**

The variable 2 is loaded with the value by the motor performance.  
The variable 2 has the content 51.

Value = 21:

load the actual drive performance in the specified variable.

The 100% of the performance is normalized to  $I_{\text{Nominal}}^2 \cdot \sqrt{2}$ .

**Example:**

The drive performance is 100%.  
The command is executed with a drive 637f-20 (20 Amps nominal current)

**Result:**

The variable 2 is loaded with the value by the drive performance.  
The variable 2 has the content 560.

**0 Variable 2 = value 21**

Value = 22:

load the actual CAN2-Bus-node monitoring state.

**Example:**

**0 Variable 2 = value 22**

**Result:**

The variable 2 is loaded with the value by the CAN2-Bus-node monitoring state.  
The bit of the asking axis is not set to one.

value = 23:

load the rotor position at positive slope of the reference sensor at homing mode with reference sensor.  
See BIAS-command "[Move datum](#)"  
(from Firmware V8.20 only 637f)

**Example:**

**Result:**

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>4C</b>	<b>X</b>	<b>Y</b>	00	00	00	00	00

**Parameters:** D1 = Variable number: 0...255  
D2 = Value 0...22

#### 4 - 4D BIAS-Command “[Variable X] = axis status, axis no. Y“

##### Explanation:

The command "[Variable X] = axis status, axis no. Y" loads the actual status of the specified axis Y in the specified variable X.

Drive type:	Parameters:	Permissible inputs:	
637f:	X:	0...255	<input checked="" type="checkbox"/> BIAS program
	Y:	1...3	<input checked="" type="checkbox"/> SPS program
			<input checked="" type="checkbox"/> Mathematics

Bit	Meaning	Axis status of axis 1 - 3
0	Done	This flag signals that either no multi axis move command was started up to now, or the last commanded multi axis move command was terminated. The signal will stay on low after start of a move command for the minimum ““Pos reached” time” (EASYRIDER parameter). <b>NOTE: A stop command with axis no != 0 is also a move command !!!</b>
1	Acceleration	Axis X accelerates
2	Continuous	Axis X moves with constant speed
3	Deceleration	Axis X decelerates
4	Slope start angle	Signal that synchronous start happens (synchronism is active, start cam drive through)
5	Trigger Gearbox Ext	Trigger for update the gearbox parameters. The flag is set to 1 by the gearbox function block of the synchronous movement and can be reset by BIAS.
6	Trigger Clutch Ext	Trigger for manual update parameter (The flag must set to 1 or set to 0 by BIAS) e.g. synchronisation of switching parameters with other drives or PLC's
7	Trigger CAM Ext	Trigger for manual update parameter (The flag must set to 1 or set to 0 by BIAS) e.g. synchronisation of switching parameters with other drives or PLC's
8	Gearbox Trigger	Trigger for update the gearbox parameters. The flag is set to 1 by the gearbox function block of the synchronous move for one sampling cycle.
9	Clutch Trigger	Trigger update clutch factor 1 cycle
10	CAM Trigger	Trigger for update the the caming parameters The flag is set to 1 by the CAM function block of the synchronous move for one sampling cycle.
11	First CAM	Reserved for Internal diagnosis
12	Backstop Error	Error master moves in wrong direction
13	Start Error	Start angle is drive through by switch on the synchronism
14	Clutch Error	Parameter or control error clutch
15	CAM Start Error	Error in CAM data
16	Start Angle	Selection of the function for synchronous movement start on defined master angle (Bit 0 =1).
17	Start direction	Direction in which the start is to take place on defined master angle (0 = positive counting direction).
18	Gearbox Trigger On	Switch on of the trigger function for the gearbox. The gear factor is took over at the trigger point at format length reached (gear factor denominator) (“Gearbox Trigger On“ =1). If the gear factor is took over immediately the flag “bit gearbox Trigger Ext“ must set to 1 after the new gear factor was written (Trigger On =0).
19	Back stop	Stops the synchronous move if the master turns backwards. In this case the errorbit „Backstop error“ is set. The slave is stopped until the master is crossing again the switch off point in positive direction. Forward or backward is defined with the flag “Start direction“



20	Clutch On	Switch on the clutch function (Bit 4 =1)
21	Ramp direction	Slave direction of the clutch function
22	Ramp Up	Starts the synchronous ramp-up of the clutch function
23	Ramp Down	Starts the synchronous ramp-down of the clutch function
24	Ramp Wait	Starts the synchronous wait of the clutch function
25	Move Linked	Move Linked = 1, moves the positions Move Linked ACC, Move Linked Cont and Move Linked DEC consecutively.
26	Merge Move Linked	Start a new profile, by a positive slope of the flag " Merge Move Linked" the new formats are took over and the flag "Merge Move Linked" is cleared after the restart.
27	Coming On	Switch on the CAM-Function block
28	Stop CAM End	Stops the synchronous movement at the end of cam profile
29	CAM Backstop <b><u>This function is still in preparation !!!</u></b>	If the reference is turning backwards, the movement of the CAM is stopped at the beginning of the profile, until the reference is crossing again the switch off point in positive direction. With this flag it is possible to carry on CAM execution in negative direction of the reference.
30	Reserve	
31	Reserve	

**Example:**

```

9 ...
10 [Variable 0 ] = axis status, axis no. = 2
11 If [variable 0 ].Bit 12 == 1 then jump Error
12 ...

```

```

Error:
60 ...

```

**Result:**

The axis status of the axis 2 is allocated the variable 2.  
If the flag Backstop Error is set to 1 the program continuous at the label "Error" otherwise the program continuous at line 12.

**Note:**

- This command is available with the 637f as of firmware V8.19

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>4D</b>	<b>X</b>	<b>Y</b>	00	00	00	00	00

**Parameters:** D1 = Variable number: 0...255  
D2 = axis number: 1...3

## 4 - 50 BIAS-Command "NOP"

### Explanation:

The command "NOP" serves as a line space and the following block is worked on.

### Parameters:

none

- BIAS program
- PLC program
- Mathematics

### Example:

```

0 NOP
1 NOP
2 [Variable 0 ] = 1
3 Jump 3

```

### Result:

The assignment of the variable 0 is delayed for 2 cycles.  
(The NOP's in line 0 and 1).

**Note:** NOP-lines make the program execution slowly.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
50	00	00	00	00	00	00	00

## 4 - 51 BIAS-Command “End of program, mode =“

End of program, mode = X

**Explanation:**

The command “End of program, mode =“ continue the execution depending on the value of the parameter “mode”.

**Parameters:**

**Permissible inputs:**

**X: Mode:**

**0...6**

- BIAS program
- PLC program
- Mathematics

- 0 = Return to program start block of the calling task (BIAS-, PLC- oder Mathematic-program)
- 1 = stop the execution of the BIAS program
- 2 = stop the execution of the PLC program
- 3 = stop the execution of the Mathematic program
- 4 = Deactivates the drive
- 5 = Activate the drive (only possible within the PLC-program with “Continue execution at reactivation” and if the drive is deactivated with mode 4).
- 6 = execute a drive reset (only possible within the PLC-program with “Continue execution at reactivation”)

**Example:**

ProgStart:

- 0 [Variable 0 ] = 0
- 1 [Variable 1 ] = 0
- 2 Mathematic program MATH
- 3 PLC-program PLC
- 4 End of program, mode = 1**

PLC:

- 5 [Variable 0 ] = [variable 0 ] + 1
- 6 End of program, mode = 0**

MATH:

- 7 [Variable 1 ] = [variable 1 ] + 1
- 8 End of program, mode = 0**

**Result:**

After the start the BIAS- program calles the mathematic and PLC task. Then the BIAS- program is stoped. The mathematic and the PLC task are executed cyclic (return to program start block of the calling task)

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>51</b>	<b>X</b>	00	00	00	00	00	00

**Parameters:** D1 = Mode 0-6

**4 - 52 BIAS-Command “Sub-program“**

**Sub-program X**

**Explanation:**

The command "Sub-program" sets the BIAS-execution pointer to the specified set block-number.  
 At the same time the following block-number is stored as the return address in the sub-program-stack.  
 The stack depth is limited to 128 sub-programs.  
 The return is programmed by the command “End\_of sub-program“.



**Parameters:**

**X:** Address:

**Permissible inputs:**

**0...1499**

The maximum address depends on the length of the BIAS-program and corresponds to the block-number of the last BIAS-block.  
 Instead of block-numbers, labels can be used.

- BIAS program
- PLC program
- Mathematics

**Example:**

ProgStart:

**0 Sub-program UP\_1**

1 Springe 1

**UP\_1:**

**2 Sub-program UP\_2**

3 [Variable 0 ] = 0

4 End of sub-program

**UP\_2:**

5 [Variable 1 ] = 123

6 End of sub-program

**Result:**

At first the BIAS execution calls the sub program UP\_1.  
 The sub program UP\_1 calls the sub program UP\_2.  
 After all sub program has stopped the BIAS execution is continued with line 1.

**Note:**

- BIAS-, PLC- und Mathematic- programs have separated sub-program-stacks.
- If the command “BIAS execution pointer“ is used, the BIAS-program-stack is cleared!

**Command coding:**

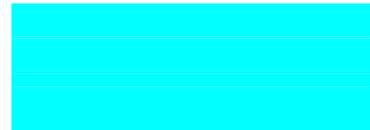
Code	D1	D2	D3	D4	D5	D6	D7
52	X(low)	X(high)	00	00	00	00	00

**Parameters:** D1,D2 = Jump address: 0...1499

## 4 - 53 BIAS-Command “End of sub-program “

**Explanation:**

The Command “End of sub-program“ loads the return address stored in the sub program stack and concludes the subroutine.



**Parameters:**

None

- BIAS program
- PLC program
- Mathematics

**Example:**

ProgStart:

0 Sub-program UP\_1  
1 Jump 1

UP\_1:

2 Sub-program UP\_2  
3 [Variable 0 ] = 0  
4 **End of sub-program**

UP\_2:

5 [Variable 1 ] = 123  
6 **End of sub-program**

**Result:**

At first the BIAS execution calls the sub-program UP\_1. The sub-program UP\_1 calls the sub-program UP\_2. After all sub-programs are finished the BIAS execution is continued with line 1.

**Note:**

Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>53</b>	00	00	00	00	00	00	00

## 4 - 54 BIAS-Command "PLC-program"

### PLC-program X

#### Explanation:

The command "PLC-program" sets the PLC-execution pointer to the specified block-number.

With this command a cyclic program can be started parallel to the BIAS-program execution.

The specified block-number can be taken as a return address in the PLC-program with the command "End of program".

#### Parameters:

X: Address:

#### Permissible inputs:

0...1499

the maximum address depends on the length of the BIAS-program and corresponds to the block-number of the last BIAS-block. Instead of block-numbers, labels can be used.

- BIAS program
- PLC program
- Mathematics

#### Example:

ProgStart:

- 0 [Variable 0 ] = 0
- 1 [Variable 1 ] = 0
- 2 Mathematic program MATH
- 3 PLC-program PLC**
- 4 End of Program, mode = 1

#### PLC:

- 5 [Variable 0 ] = [variable 0 ] + 1
- 6 End of Program, mode = 0

#### MATH:

- 7 [Variable 1 ] = [variable 1 ] + 1
- 8 End of Program, mode= 0

#### Result:

After the start the BIAS- program calls the mathematic and PLC task. Then the BIAS- program is stopped. The mathematic and the PLC task are executed cyclic (return to program start block of the calling task)

#### Note:

- Do not execute the command "PLC-program cyclically, because the PLC-processing will always begin at the specified block-number.
- In the PLC- loop not all of the BIAS commands are allowed.
- The reaction of the PLC program to the deactivation of the output stage can be adjusted in the BIAS program definition (parameter "program reaction PLC program") Thus it is possible to allow the PLC program to allow the PLC program to continue to process also during the deactivation of the regulator.
  - Is in this mode the first command of the BIAS execution the command "PLC program" the PLC task starts automaticly independently of the state (deactive/active) of the drive.
- **Attention: With this mode "Continue execution at reactivation"!! Download of a BIAS-program with changed command lines may result in uncontrolled movements and program reactions!**

#### Command coding:

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

Parameters: D1,D2 = Jump address: 0...1499

## 4 - 55 BIAS-Command "Jump const."

### Jump const.

**Explanation:**

The command "Jump const." sets the execution pointer to the specified block number.



**Parameters**

**Const.:**

**Permissible inputs:**

**0...1499**

The maximum address depends on the length of the BIAS-program and corresponds to the block-number of the last BIAS-block. Instead of block-numbers, labels can be used.

- BIAS program
- PLC program
- Mathematics

**Example:**

```

0 If flag 0 == 0 then jump 3
1 [Variable 0] = 5
2 Jump 4
3 [Variable 0] = 1
4 [Variable 0] = [variable 0]* 100
5 Jump [variable 0]
```

**Result:**

If the flag 0 is not set the execution is continued in line 3 and the variable 0 is multiplied with 100.

**Note:**

1. The maximum address depends on the length of the BIAS-program and corresponds to the block-number of the last BIAS-block.
2. Instead of block-numbers, labels can be used.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>55</b>	<b>X(low)</b>	<b>X(high)</b>	00	00	00	00	00

**Parameters:** D1, D2 = Jump address: 0...1499

## 4 - 56 BIAS-Command “Jump [variable X]“

### Jump [variable X ]

#### Explanation:

The command “Jump [variable X]“ sets the execution pointer to the value of the variable X.

#### Parameters

#### X:

#### Permissible inputs:

**0...255**

Permissible variable content

**0...1499**

The maximum address depends on the length of the BIAS-program and corresponds to the block-number of the last BIAS-block.

Instead of block-numbers, labels can be used.

- BIAS program
- PLC program
- Mathematics

#### Example:

```

0 If flag 0 == 0 then jump 3
1 [Variable 0] = 5
2 Jump 4
3 [Variable 0] = 1
4 [Variable 0] = [variable 0]* 100
5 Jump [variable 0]
```

#### Result:

If the flag 0 is set to 0 the execution is continued in line 500 and if the flag 0 is not set to 1 the execution is continued in line 100.

#### Note:

#### Command coding:

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

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



## 4 - 57 BIAS-Command “BIAS-execution pointer = const.”

**BIAS-execution pointer = const.**

**Explanation:**

The command “BIAS-execution pointer = const.” sets the execution pointer of the BIAS program to the specified block number.



**Parameters:**

Address:

**Permissible inputs:**

**0...1499**

The maximum address depends on the length of the BIAS-program and corresponds to the block-number of the last BIAS-block. Instead of block-numbers, labels can be used

- BIAS program
- PLC program
- Mathematics

**Example:**

0 PLC-program 50

**Stop\_1:**

10 Stop axis, mode 1, a- = 10000 rpm/s

11 Jump 11

...

50 If actual position > 500000 INKR then jump Stop

51 If actual position < 0 INKR then jump Stop

52 End of program

Stop:

**53 BIAS-execution pointer Stop\_1**

54 End of program

**Result:**

The processing of the BIAS program is continued in block 10 (label Stops) as soon as the actual position is greater than 500000 increments or smaller than 0 increments.

**Note:**

1. The maximum address depends on the length of the BIAS-program and corresponds to the block-number of the last BIAS-block.
2. Instead of block-numbers, labels can be used

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>57</b>	<b>X(low)</b>	<b>X(high)</b>	00	00	00	00	00

**Parameters:** D1, D2 = Jump address: 0...1499

## 4 - 58 BIAS-Command "Wait for "position reached" "

### Wait for "position reached"

#### Explanation:

The command "Wait for "position reached"" waits for the the calculation of the trajectory generator and then for the position +/- the "position reached" window.

After reaching this position the "position reached bit" is set to 1 and the next BIAS command is executed.

#### Parameters:

none

- BIAS program
- PLC program
- Mathematics

#### Example:

```
0 Position = 10000 INKR
1 "Pos. reached" window = 1
2 Move position
3 Wait for "position reached"
4 ...
```

#### Result:

The BIAS execution will only continue in line 4, if moving to position ends and the actual position is between 9900 up to 10100 increments.

#### Note:

- The bit "position reached" will set to 1 if a limit switch is active (dependent on direction)!
- The Bit "position reached" is always set in the mode 8 (Multi axis function)  
To check the target position reached signal use the command "[Variable X] = axis status, axis no." in the multi axis mode.

see also:

- BIAS-command "If status == 8 then jump"

BIAS-command "Position reached window = const."

Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
58	00	00	00	00	00	00	00

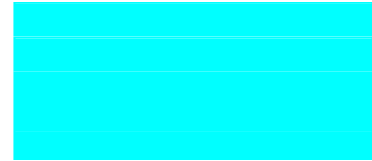
Parameters: -

## 4 - 59 BIAS-Command "Wait time = const."

Wait time X ms

**Explanation:**

The command "Wait time" starts the specified waiting time. Only after this waiting time runs out will the next BIAS command be executed.



**Parameters:**

**Permissible inputs:**

Const.

0...131070 ms

- BIAS program
- PLC program
- Mathematics

**Example:**

```

1 ...
2 Move position
3 Wait for "position reached"
4 Wait time 1000 ms
5 ...
    
```

**Result:**

The BIAS execution will continue at line 5, 1 second after reaching the position.

**Note:**

- Only even waiting periods are permissible. Odd waiting periods get roundet off.
- When selecting a block number via the strobe function the waiting time is deleted.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>59</b>	<b>X(low)</b>	<b>X(high)</b>	00	00	00	00	00

**Parameters:** D1, D2 = Wait time: 0...131070

## 4 - 5A BIAS-Commands "Wait time = [variable X]"

Wait time [variable X ]

### Explanation:

The command "Wait time [variable X]" starts the waiting time specified in the variable X.

Only after this waiting time runs out will the next BIAS command be executed.

### Parameters:

X:

### Permissible inputs:

0...255

permissible variable content:

0...65535

BIAS program

PLC program

Mathematics

### Example:

```

1 [ Variable 0] = 1000
2 Move position
3 Wait for "position reached"
4 Wait time [variable 0]
5 ...

```

### Result:

The BIAS execution will continue at line 5, 1 second after reached the position..

### Note:

- The variable content is interpreted as a time specified in ms.  
The drive types 637+ and 637f count the time in 1 ms steps  
The drive types 631,635 and 637 count the time in 2 ms steps. Odd waiting periods get rounded off.
- When selecting a block number via the strobe function the waiting time is deleted

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
5A	X	00	00	00	00	00	00

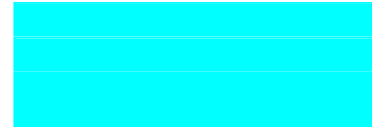
Parameters: D1 = Variable number: 0...255

## 4 - 5B BIAS-Command "BIAS-execution pointer = [variable X]"

BIAS-execution pointer = [variable X ]

### Explanation:

The command "BIAS-execution pointer = [variable X]" sets the BIAS execution pointer to the block-number specified in Variable X.



### Parameters:

X:

### Permissible inputs:

0...255

The maximum permitted variable content depends on the length of the BIAS program and corresponds to the block-number of the last BIAS block. Instead of block-numbers, labels can be used

- BIAS program
- PLC program
- Mathematics

### Example:

```

0 [Variable 0] = 10
1 PLC-program 50
...
Stop_1:
10 Stop axis, mode 1, a- = 10000 rpm/s
11 Jump 11
...
50 If actual position > 500000 INCR then jump Stop
51 If actual position < 0 INCR then jump Stop
52 End of program
Stop:
53 BIAS-execution pointer = [variable 0 ]
54 End of program
    
```

### Result:

The processing of the BIAS program is continued in line 10 (label Stop\_1) if the actual position is greater than 500000 or less than 0 increments.

### Note:

1. The maximum address depends on the length of the BIAS-program and corresponds to the block-number of the last BIAS-block.
2. Instead of block-numbers, labels can be used.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
5B	X	00	00	00	00	00	00

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

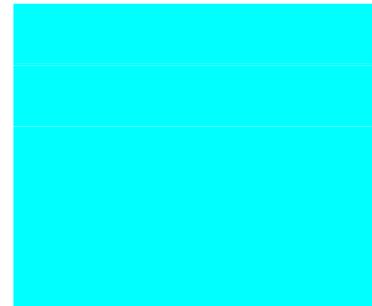
**4 - 5C BIAS-Command “Jump [var.[ X ]]; length = Y ; from line“**

**Jump [var.[ X]]; length= Y ; from Z**

**Explanation:**

The command “Jump [var.[ X ]]; length = Y ; from line“ allows program branches dependent on the content of the specified variable X. The content of the variable X is not allowed to be greater than the length Y. The command beginning from line Z must be jump commands (e.g. “Jump const.” or “Jump [variable X])!

The first command at the jump target is executed in the same cycle.



<b>Parameters:</b>	<b>Permissible inputs:</b>	<input checked="" type="checkbox"/> BIAS program
<b>X:</b>	<b>0...255 (Variable content X &lt;= Y)</b>	<input checked="" type="checkbox"/> PLC program
<b>Y:</b>	<b>0...99</b>	<input type="checkbox"/> Mathematics
<b>Z:</b>	<b>0...1499</b>	
	<b>(Y+line)&lt;1499</b>	

**Example:**

```

10 Jump [var.[64 ]]; length=6; from JUMP
JUMP:
11 Jump STANDSTILL
12 Jump MOVE_ABS
13 Jump MOVE_REL
14 Jump MOVE_HOME
15 Jump MOVE_INF
16 Jump STOP_AXIS
STANDSTILL:
17 Jump 10
MOVE_ABS:
18 Starts axis
19 Move position
.....
    
```

**Result:**

If the line 10 is executed the programm will be continue dependent on the content of the specified variable 64 with the next command at the jump target.

The content of the variable 64 is 1, then the program continues at the label MOVE\_ABS with executing the command “Start axis”.

**Note:**

- This command is available with the 637f as of firmware V8.19
- This command is available with the 631/635/637 as of firmware V6.19

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
5C	X	Y	Z(low)	Z(high)	00	00	00

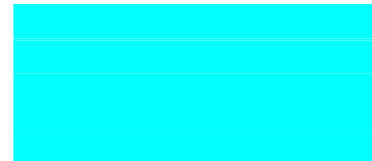
**Parameters:** D1 = Variable number: 0...255  
 D2 = Length: 0...99  
 D3,D4 = Jump address: 0...1499

## 4 - 5D BIAS-Command "Execute X commands"

### Execute X commands

#### Explanation:

The command "Execute X commands" executes the number of commands which are defined in the parameter X in one cycle. The command is not counted itself!



#### Parameters

X:

#### Permissible inputs:

0...9

- BIAS program
- PLC program
- Mathematics

#### Example:

- 0 Wait time 1000 ms
- 1 Execute 6 commands**
- 2 Position = 10000 INCR
- 3 Speed = 1000 rpm
- 4 Acceleration = 5000 rpm/s
- 5 Deceleration = 5000 rpm/s
- 6 Start axis
- 7 Move position; axis no. = 0 , target = 0
- 8 Wait for "pos. reached"
- 9 NOP

#### Result:

After the delay time of 1 second the positioning is started in the next cycle immediately. The commands 2 to 7 are executed in one cycle.

#### Note:

- This command is available with the 637f as of firmware V8.19
- The command is ignored if this command is in the block of the commands which are executed in one cycle..  
In this case the next command is executed.

#### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>5D</b>	<b>X</b>	00	00	00	00	00	00

**Parameters:** D1: Number of commands 0...9

## 4 - 5F BIAS-Command "Virtual program"

Virtual program a= W, b = X, c = Y, d = Z

### Explanation:

The command "Virtual program" is implemented only for test purposes and not for use!

Parameters	Permissible inputs:	
W:	0...255	<input checked="" type="checkbox"/> BIAS program
X:	0...255	<input type="checkbox"/> PLC program
Y:	0...255	<input type="checkbox"/> Mathematics
Z:	0...255	
Further informations are in the manual Configuration <a href="#">trajectory generator</a>		

### Example:

### Result:

### Note:

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
5F	W	X	Y	Z	00	00	00

**Parameters:** D1 = Variable number: 0...255  
 D2 = Variable number: 0...255  
 D3 = Variable number: 0...255  
 D4 = Variable number: 0...255

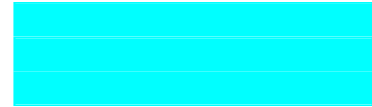


## 4 - 60 BIAS-Command "Flag X = const."

Flag X = const.

### Explanation:

The command "Flag X =" assigns the constant 0 or 1 to the flag.



Parameters	Permissible inputs	
X:	0...255	<input checked="" type="checkbox"/> BIAS program
const.:	0/1	<input checked="" type="checkbox"/> PLC program
		<input checked="" type="checkbox"/> Mathematics

### Example:

0 Flag 255 = 1

### Result:

The flag 255 is set to 0.

### Note:

- After power on all flags are cleared.

### Command coding:

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

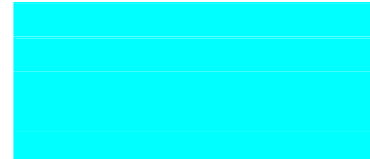
**Parameters:** D1 = flag number 0..255  
D2 = const. 0 / 1

**4 - 61 BIAS-Command “If flag X ? const. then jump “**

If flag X ? const. then jump Y

**Explanation:**

The command “If flag X ? const then jump Y“ allows program branches through the comparison of the selected flag with the constant 0 or 1. If the condition is true the processing will be continued with the specified command.



Parameters	Permissible inputs:	Comment:	
X:	0...255	equal to not equal to	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
?:	== !=		
Const.:	0/1		
Y:	0...1499		
Address:	The maximum address depends on the length of the BIAS-program and corresponds to the block-number of the last BIAS-block.		

**Example:**

**0** If flag 0 == 1 then jump 3  
**1** Output12 = 0  
**2** Jump 4  
**3** Output12 = 1  
**4** Jump 4

**Result:**

If the flag 0 is set to 1, output 12 will be set to 1. (Output 12 must be configured as a BIAS output).

**Note:**

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
61	X	?	Const.	Y(low)	Y(high)	00	00

**Parameters:** D1 = Flag number: 0...255  
 D2 = Comparison condition == (0) / != (1)  
 D3 = Constant: 0/1  
 D4, D5 = Jump address: 0..1499

## 4 - 62 BIAS-Command "Flag X = flag Y"

Flag X = flag Y

### Explanation:

The command "Flag X = flag Y" assigns the content of the flag Y to the flag X.

### Parameters

X:

Y:

### Permissible inputs:

0...255

0...255

BIAS program

PLC program

Mathematics

### Example:

0 Flag 0 = flag 1

### Result:

The state of the flag 1 is copied into the flag 0.

### Note:

Command coding:

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

**Parameters:** D1 = Flag number 0...255

D2 = Flag number 0...255

## 4 - 63 BIAS-Command "Flag X = input Y"

Flag X = input Y

### Explanation:

The command "Flag X = input Y" assigns the state of the input Y to the flag X.

Drive type:	Parameters:	Permissible inputs:	
631:	X:	0...255	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Y:	7, 8, 9, 10 Pin no. at X10-clamp	
635/637/ 637+/637f:	Y:	2, 4, 11, 15, 22, 24, 25 Pin no. at X10- female	
	Y: I/O-Option board RP-EA5 (Slot B)	101, 102, 106, 107, 108 Pin no. at COM2 Sub-D-female	
	Y: I/O-Option board RP-EAE (Slot C)	201-208, 211-216 Pin no at X200 Sub-D-female	
637f:	Y: 2 x CAN + 4 outputs and 4 inputs RP-2C8 (Slot B):	121 – 124 Pin no. at X120 clamp	
	Y: 2 Inputs RP-SBT (Slot C)	91, 93 Pin no. at X290 clamp 91: Activation Drive 93. Safe standstill	

### Example:

0 Flag 0 = input 2

### Result:

The flag 0 reflects the state of the input X10.2.

### Note:

- The inputs can be scanned independently of their configuration.

### Command coding:

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

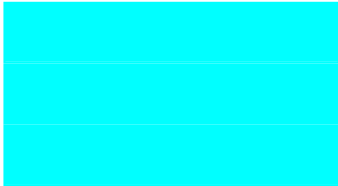
Parameters: D1 = Flag number: 0...255  
 D2 = Number of the input: (see [explanation](#))

## 4 - 64 BIAS-Command “Flag X = output Y“

Flag X = Output Y

**Explanation:**

The command "Flag X = output Y" assigns the state of the output to the flag X.



Drive type:	Parameters:	Permissible inputs:	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
631:	X:	0...255	
	Y:	5, 6 Pin no. at X10-clamp	
635/637/ 637+/637f	Y:	8, 12, 13, 20, 23 Pin no. at X10- clamp	
	X: I/O-Option board RP-EA5 (Slot B)	109, 110 Pin no. at COM2 Sub-D-female	
	X: I/O-Option board RP-EAE (Slot C)	209-210, 217-224 Pin no. at X200 Sub-D-female	
637f	X: 2 x CAN + 4 outputs and 4 inputs RP-2C8 (Slot B):	125 – 128 Pin no. at X120 clamp	

**Example:**

0 Flag 0 = output 12

**Result:**

The flag 0 reflects the state of the output X10.12.

**Note:**

- The outputs can be scanned independently of their configuration.

**Command coding**

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

**Parameters:** D1 = Flag number: 0...255  
 D2 = Number of the output: (see [explanation](#))

#### 4 - 65 BIAS-Command "Flag X = flag Y & flag Z"

Flag X = flag Y & flag Z

##### Explanation:

The command "Flag X = flag Y & flag Z" makes an "AND" operation with the flag Y and the flag Z and stores the result in the flag X.

##### Parameters:

X:

Y:

Z:

##### Permissible inputs:

0...255

0...255

0...255

BIAS program

PLC program

Mathematics

##### Example:

0 Flag 0 = 0

1 Flag 1 = 1

2 **Flag 2 = flag 1 & flag 0**

##### Result:

Flag 2 = 0

##### Note :

Operation table:

**X = Y & Z**

0 = 0 & 0

0 = 0 & 1

0 = 1 & 0

1 = 1 & 1

##### Command coding:

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

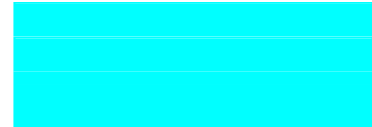
**Parameter:** D1 = Flag number 0...255  
D2 = Flag number 0...255  
D3 = Flag number 0...255

## 4 - 66 BIAS-Command “Flag X = flag Y | flag Z”

$$\text{Flag X} = \text{flag Y} \mid \text{flag Z}$$

**Explanation:**

The command "Flag X = flag Y | flag Z" makes an "OR" operation with the flag Y and the flag Z and stores the result in the flag X.



**Parameters:**

**Permissible inputs**

**X:** 0...255  
**Y:** 0...255  
**Z:** 0...255

- BIAS program
- PLC program
- Mathematics

**Example:**

0 Flag 0 = 0  
 1 Flag 1 = 1  
 2 **Flag 2 = flag 1 | flag 0**

**Result:**

Flag 2 = 1

**Note:**

Operation table:

X	=	Y		Z
0	=	0		0
1	=	0		1
1	=	1		0
1	=	1		1

**Command coding:**

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

**Parameter:** D1 = Flag number 0...255  
 D2 = Flag number 0...255  
 D3 = Flag number 0...255

#### 4 - 67 BIAS-Command "Flag X = flag Y ^ flag Z"

$$\text{Flag X} = \text{flag Y} \wedge \text{flag Z}$$

##### Explanation:

The command "Flag X = flag Y ^ flag Z" makes an "exclusive OR" operation with the flag Y and the flag Z and stores the result in the flag X

##### Parameters:

**X:** 0...255  
**Y:** 0...255  
**Z:** 0...255

##### Permissible inputs:

- BIAS program
- PLC program
- Mathematics

##### Example:

0 Flag 0 = 0  
 1 Flag 1 = 1  
 2 **Flag 2 = flag 1 ^ flag 0**

##### Result:

Flag 2 = 0

##### Note:

Operation table:

<b>X</b>	<b>=</b>	<b>Y</b>	<b>^</b>	<b>Z</b>
0	=	0	^	0
1	=	0	^	1
1	=	1	^	0
0	=	1	^	1

##### Command coding:

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

**Parameters:** D1 = Flag number 0...255  
 D2 = Flag number 0...255  
 D3 = Flag number 0...255

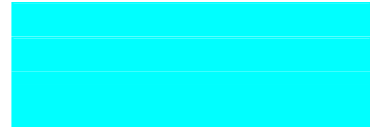


## 4 - 68 BIAS-Command "Flag X = !flag Y"

Flag X = !flag Y

**Explanation:**

The command "Flag X = !flag Y" assigns the negated content of the flag Y to the flag X.



Parameters	Permissible inputs	
X:	0...255	<input checked="" type="checkbox"/> BIAS program
Y:	0...255	<input checked="" type="checkbox"/> PLC program
		<input checked="" type="checkbox"/> Mathematics

**Example:**

0 Flag 0 = 0  
 1 Flag 1 = ! flag 0

**Result:**

The flag 1 ist set to 1

**Note:**

**Command coding:**

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

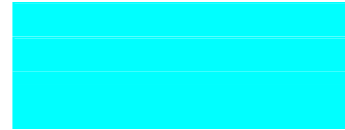
**Parameters:** D1 = Flag number 0...255  
 D2 = Flag number 0...255

**4 - 69 BIAS-Command “Flag X = status Y“**

Flag X = status Y

**Explanation:**

The command "Flag X = Status Y" copies the current status to flag X.



Parameters:	Permissible inputs:	Meaning:	
<b>X:</b>	<b>0...255</b>		<input checked="" type="checkbox"/> BIAS program
<b>Y:</b>	<b>0</b>	position reached	<input checked="" type="checkbox"/> PLC program
	<b>1</b>	output stage inactive	<input checked="" type="checkbox"/> Mathematics
	<b>2</b>	format check executed (Synchron)	
	<b>3</b>	intern (knock down bit)	
	<b>4</b>	offset correction executed	
	<b>5</b>	trigger profile switch	
	<b>6</b>	sensor latch error	
	<b>7</b>	"move datum" successfully done	
	<b>8</b>	target position reached	
	<b>9</b>	IBT data transfer runs	
	<b>10</b>	axis in position control (standstill)	
	<b>11</b>	axis in deceleration ramp	
	<b>12</b>	ramp filter function done (new value can be loaded)	
	<b>13</b>	table saved runs (only for 637f)	
	<b>14</b>	pos. position limit (only for 637f)	
	<b>15</b>	neg. position limit (only for 637f)	
	<b>16</b>	limit switches detected	
	<b>17</b>	speed limit reached (only for 637f)	

**Example:**

0 Flag 1 = status 11

**Result:**

The flag 1 shows if the position set value is in the deceleration ramp.

**Note:**

- The Bit 8 "target position reached" is always set in the mode 8 (Multi axis function)  
To check the target position reached signal use the command "[Variable X] = axis status, axis no." in the multi axis mode.
- The status 2 (format check synchronous move) is set only on one BIAS-cycle.

**Command coding:**

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

**Parameters:** D1 = Flag number: 0...255  
D2 = Status: 0...17

## 4 - 6A BIAS-Command "If status X ? const. then jump"

If status X ? const. then jump Y

**Explanation:**

The command "If status X ?" allows program branches through the comparison of the selected status X with the constants 0 and 1.

If the comparison is true, the processing is continued with the specified command.

Parameters	Permissible inputs:	Meaning	
<b>Y:</b>	<b>0</b>	position reached	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	<b>1</b>	output stage inactive	
	<b>2</b>	format check executed (Synchron)	
	<b>3</b>	intern (knock down bit)	
	<b>4</b>	offset correction executed	
	<b>5</b>	trigger profile switch	
	<b>6</b>	sensor latch error	
	<b>7</b>	"move datum" successfully done	
	<b>8</b>	target position reached	
	<b>9</b>	IBT data transfer runs	
	<b>10</b>	axis in position control (standstill)	
	<b>11</b>	axis in deceleration ramp	
	<b>12</b>	ramp filter function done (new value can be loaded)	
	<b>13</b>	table saved runs (only for 637f)	
	<b>14</b>	pos. position limit (only for 637f)	
	<b>15</b>	neg. position limit (only for 637f)	
	<b>16</b>	limit switches detected	
	<b>17</b>	speed limit reached (only for 637f)	
<b>?:</b>	<b>==</b>	<b>equal to</b>	
	<b>!=</b>	<b>not equal to</b>	
<b>Const.:</b>	<b>0/1</b>		
<b>Y:</b> <b>Address:</b>	<b>0...1499</b>	The maximum permitted address depends on the length of the BIAS program and corresponds to the block-number of the last BIAS block. As an alternative to the block-numbers, labels (jump labels) can be used.	

**Example:**

**Result:**

**Note:**

- The Bit 8 "target position reached" is always set in the mode 8 (Multi axis function)  
To check the target position reached signal use the command "[Variable X] = axis status, axis no." in the multi axis mode.
- The status 2 (format check synchronous move) is set only on one BIAS-cycle.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>6A</b>	<b>X</b>	<b>?</b>	<b>Const</b>	<b>Y(low)</b>	<b>Y(high)</b>	00	00

**Parameters:** D1 = Status 0...17  
 D2 = Comparison condition: == (0)/ != (1)  
 D3 = Constant: 0/ 1  
 D4, D5 = Jump address: ...1499

## 4 - 6B BIAS-Command "Mode X = const."

### Explanation:

The command "Mode =const." loads the specified mode with the constant value.

Parameters	Permissible inputs:	
X:	0...255	<input checked="" type="checkbox"/> BIAS program
Const.:	231	<input checked="" type="checkbox"/> PLC program
		<input checked="" type="checkbox"/> Mathematics

Mode	Special functions
0	Shift factors (only 637f)
4	0 = Ramp steepness standard 1 = Ramp steepness multiplied by 4 (only 637f)
8	0 = Multi axis mode off 1 = Multi axis mode (only 637f)
22	0 = CAN2-Bus-node monitoring value reset (only 637f with RP 2CA/ RP 2C8)

### Example:

```

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 Load parameter set 0 = [variable [14 ]]
7 Start axis
8 Move position; Axis no. = 1 , target = 1
9 [Variable 0 ] = Axis status, Axis no. = 1
10 If [variable 0 ].Bit 0 == 0 then jump 9
11 ...

```

### Result:

The multi axis mode is activated. The variables define the parameter set for a positioning. The axis moves on the position 200000 increments and after the axis reached this position the line 11 is executed.

### Note:

#### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
6B	X	Y (low)			Y (high)	00	00

Parameters: D1 = Mode: 0...255  
D2-D5 = Constant: ± 231

## 4 - 6C BIAS-Command "Flag X = [variable Y], number Z"

Flag X = [variable Y ], number = Z

### Explanation:

The command "Flag X = [variable Y], number Z" copies the Z lowest bits of the content variable Y in the flag range starting with flag X.

Flag X = Bit 0



### Parameters:

X:  
Y:  
Z:

### Permissible inputs:

0...255  
0...255  
1...32  
note: (X+Z)<257

- BIAS program
- PLC program
- Mathematics

### Example:

The 4 lowest bits of the variable 0 are copied in the flags 0 to 3.

0 Variable 0 = 15  
1 Flag 0 = variable 0, number 4

### Result:

Flag 0 = 1  
Flag 1 = 1  
Flag 2 = 1  
Flag 3 = 1

### Note:

#### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
6C	X	Y	Z	00	00	00	00

Parameters: D1 = Flag number: 0...255  
D2 = Variable number: 0...255  
D3 = Number of bits: 1...32

#### 4 - 6D BIAS-Command “[Variable X]. bit [Y] = const.”

[Variable X ].bit Y = const. ,[flag Z ]

##### Explanation:

The command “[ Variable X ]. bit [Y] = const. , flag Z“ sets or clears the specified bit Y of the specified variable X.

If in the constant the value 255 is registered, the content of the flag Z is copied in the specified bit Y of the variable X.

Parameters	Permissible inputs:	
X:	0...255	<input checked="" type="checkbox"/> BIAS program
Y:	0...31	<input checked="" type="checkbox"/> PLC program
Const.:	0/1 , 255	<input checked="" type="checkbox"/> Mathematics
Z:	0...255	

##### Example:

0 Flag 10 = 1

1 [Variable 0 ].bit 15 = 0 ,[flag 0 ]

2 [Variable 1 ].bit 1 = 255 ,[flag 10 ]

##### Result:

The bit 15 of the variable 0 is cleared and the bit 1 of the variable 1 is set (by the flag 10= 1).

##### Note:

- This command is available with the 637f as of firmware V8.19.
- This command is available with the 631/635/637 as of firmware V6.19.

##### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
6D	X	Y	Const.	Z	00	00	00

**Parameters:** D1 = Variable number: 0...255  
D2 = Bit number 0...31  
D3 = Constant: 0/1 , 255  
D4 = Flag number: 0...255

## 4 -6E BIAS-Command “If [variable X]. bit Y ? const. then jump“

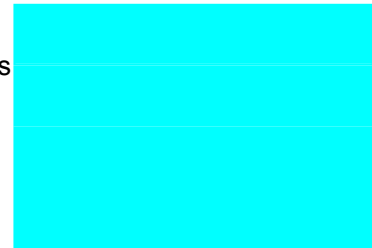
If [variable X]. bit Y ? const. then jump Z

### Explanation:

The command “ If [var. X]. bit Y ? const. then jump“ allows program branches by comparing the bit Y of the variable X with the constant.

If the condition is true the execution will be continued at the specified command line.

Otherwise the next BIAS command is executed.



Parameters:	Permissible inputs:	Meaning:	
X:	0...255		<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
Y:	0...31		
?:	== !=	equal to not equal to	
Z: Address:	0...1499 The maximum permissible jump address corresponds to the length of the BIAS program. As an alternative to the line numbers labels (jump labels) can be used.		

### Example:

LOOP:

```
0 If [variable 10 ].bit 3 == 0 then jump LOOP
1....
```

### Result:

The program is executed at line 1 if the bit 3 of the variable 10 is set to 1.

### Note:

- This command is available with the 637f as of firmware V8.19.
- This command is available with the 631/635/637 as of firmware V6.19.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
6E	X	Y	?	Z(low)	Z(high)	00	00

**Parameters:** D1 = Variable number                    0...255  
 D2 = Bit number                                        0...31  
 D3 = Comparison condition                        == (0)/ != (1)  
 D4, D5 = Jump address                            0...1499

#### 4 - 6F BIAS-Command “Axis state, axis no. X, bit Y = const. [flag Z]“

Axis state, axis no. X, bit Y = const, [flag Z ]

##### Explanation:

The command “Axis state, axis no. X , bit Y = const. , [flag Z]“ sets or clears the specified bit of the axis state of the specified axis number X.

If in the constant the value 255 is registered, the content of the flag Z is copied in the bit of the axis state.

This command is used in context with the multi axis function

Drive type:	Parameters:	Permissible inputs:	
637f	X: Y:	1..3 5-7, 21-24, 28	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
	Const.	0/1/ 255	
	Z:	0...255	

##### Parameters Y:

Bit	Name	Meaning
5	Trigger Gearbox Ext	Trigger for update the gearbox parameters. The flag is set to 1 by the gearbox function block of the synchronous movement and can be reset by BIAS.
6	Trigger Clutch Ext	Trigger for manual update parameter (The flag must set to 1 or set to 0 by BIAS) e.g. synchronisation of switching parameters with other drives or PLC's
7	Trigger CAM Ext	Trigger for manual update parameter (The flag must set to 1 or set to 0 by BIAS) e.g. synchronisation of switching parameters with other drives or PLC's
21	Ramp direction	Slave direction of the clutch function
22	Ramp Up	Starts the synchronous ramp-up of the clutch function
23	Ramp Down	Starts the synchronous ramp-down of the clutch function
24	Ramp Wait	Starts the synchronous wait of the clutch function
26	Merge Move Linked	Start a new profile, by a positive slope of the flag " Merge Move Linked" the new formats are took over and the flag "Merge Move Linked" is cleared after the restart.
28	Stop CAM End	Stops the synchronous movement at the end of cam profile

All not listed flags are not allowed or not able to change



**Example:**

```

35 ...
36 [Variable 1 ] = axis status, axis no. = 1
37 If [variable 1 ].bit 6 == 0 the jump SYNC_SSI1
38 Axis state, axis no 1 , bit 23 = 1 , [flag 0 ]
39 Axis state, axis no 1 , bit 24 = 1 , [flag 0 ]
40 Axis state, axis no 1 , bit 6 = 0 , [flag 0 ]
41 ..

```

SYNC\_SSI1:

**Result:**

The flag “Ramp down“and “Ramp Wait“ are set and the flag “Trigger clutch Ext“ is cleared if the bit 6 of the axis status is set to 1.

**Note:**

- This command is available with the 637f as of firmware V8.19
- The function “synchronous move“ gets their parameters from the variable of the specified parameter set before start and after the parameters are defined in the axis state.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>6F</b>	<b>X</b>	<b>Y</b>	<b>Const.</b>	<b>Z</b>	00	00	00

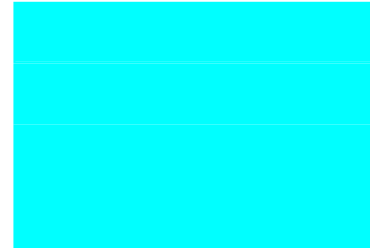
**Parameters:** D1 = Axis number 0...3  
D2 = Bit number 5-7, 21-24, 28  
D3 = Constant 0/1, 255  
D4 = Flag number 0...255

**4 - 70 BIAS-Command "If input X ? const. then jump"**

If input X ? const. then jump

**Explanation:**

The command "If input X ?" allows program branches through the comparison of the selected input X with the constant 0 or 1. If the condition is true the execution will be continued at the specified command line. Otherwise the next BIAS command is executed.



Drive type	Parameters	Permissible input:	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
631:	X:	<b>7, 8, 9, 10</b> Pin no. at X10- clamp	
635/637/ 637+/637f:	X:	<b>2, 4, 11, 15, 22, 24, 25</b> Pin no. at X10- female	
	X:  I/O-Option board RP-EA5 (Slot B)	<b>101, 102, 106, 107, 108</b> Pin no. at COM2 Sub-D-female	
	X:  I/O-Option board RP-EAE (Slot C)	<b>201-208, 211-216</b> Pin no. at X200 Sub-D-female	
637f:	X: 2 x CAN + 4 outputs and 4 inputs RP-2C8 (Slot B):	<b>121 – 124</b> Pin no. at X120 clamp	
	X: 2 Inputs RP-SBT (Slot C)	<b>91, 93</b> Pin no. at X290 clamp 91: Activation Drive 93. Safe standstill	
	?	<b>==</b> equal to <b>!=</b> not equal to	
	<b>const.</b>	<b>0/1</b>	
	Y: Address:	<b>0...1499</b> The maximum permissible jump address corresponds to the length of the BIAS program. As an alternative to the line numbers labels (jump labels) can be used.	

**Example:**

```

0 If input 2 == 1 then jump 3
1 Output 12 = 0
2 Jump 4
3 Output 12 = 1
4 Jump 4
    
```

**Result:**

If input 2 is set, the output 12 is set to 1. (Output 12 has to be configured as BIAS-output.)

**Note:**

- The inputs can be scanned independently of their configuration.
- The maximum address depends on the length of the BIAS-program and corresponds to the block-number of the last BIAS-block.  
Instead of block-numbers, labels can be used.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>70</b>	<b>X</b>	<b>?</b>	<b>Const.</b>	<b>Y(low)</b>	<b>Y(high)</b> )	00	00

**Parameters:** D1 = Input pin number: (see [explanation](#))  
 D2 = Compare condition: == (0) or != (1)  
 D3 = Compare constant: 0 / 1  
 D4,D5 = Address : 0...1499

## 4 - 71 BIAS-Command “ If output X ? const. then jump“

If output X ? const. then jump Y

### Explanation:

The command "If output X ?" allows program branches through the comparison of the selected output X with the constant 0 or 1. If the condition is true the execution will be continued at the specified command line. Otherwise the next BIAS command is executed.

Drive type:	Parameters:	Permissible inputs	
631:	X:	<b>5, 6</b> Pin no. at X10-clamp	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
635/637/ 637+/637f:	X:	<b>8, 12, 13, 20, 23</b> Pin no. at X10- female	
	X: I/O-Option board RP-EA5 (Slot B)	<b>109, 110</b> Pin no. at COM2 Sub-D-female	
	X: I/O-Option board RP-EAE (Slot C)	<b>209-210, 217-224</b> Pin no. at X200 Sub-D-female	
637f:	X: 2 x CAN + 4 outputs and 4 inputs RP-2C8 (Slot B):	<b>125 – 128</b> Pin no. at X120 clamp	
	?	== equal to != not equal to	
	Const.	<b>0/1</b>	
	Y: Address:	<b>0...1499</b> The maximum permissible jump address corresponds to the length of the BIAS program. As an alternative to the line numbers labels (jump labels) can be used.	

### Example:

```

0 If output 12 == 1 then jump 3
1 Flag 1 = 0
2 Jump 4
3 Flag 1 = 1
4 Jump 4
  
```

### Result:

If output 12 is set, the flag 1 is set to 1.

### Note:

- The output number corresponds to the pin number on the X10 or X200 connection.
- The outputs can be scanned independently of their configuration.

- The maximum address depends on the length of the BIAS-program and corresponds to the block-number of the last BIAS-block.
- Instead of block-numbers, labels can be used.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
71	X	?	const.	Y(low)	Y(high)	00	00

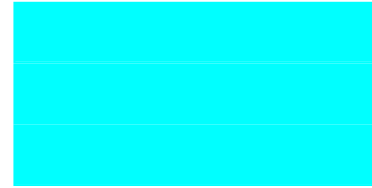
**Parameters:** D1 = Output pin number: (see [explanation](#))  
 D2 = Compare condition: == (0) / != (1)  
 D3 = Compare constant: 0 / 1  
 D4, D5 = Address 0...1499

## 4 - 72 BIAS-Command "Output X = const."

Output X = const.

### Explanation:

The command "Output X =" assigns the constant 0 or 1 to the output.



Drive type:	Parameters:	Permissible inputs	
631:	X:	5, 6 Pin no. at X10-clamp	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
635/637/ 637+/637f:	X:	12, 13, 20, 23 Pin no. at X10- female	
	X: I/O-Option board RP-EA5 (Slot B)	109, 110 Pin no. at COM2 Sub-D-female	
	X: I/O-Option board RP-EAE (Slot C)	209, 210, 217-224 Pin no. at X200 Sub-D-female	
637f:	X: 2 x CAN + 4 outputs and 4 inputs RP-2C8 (Slot B):	125– 128 Pin no. at X120 clamp	

### Example:

```

0   If input 2 == 1 then jump3
1   Output 12 = 0
2   Jump 4
3   Output 12 = 1
4   Jump 4
  
```

### Result:

1st If input 2 is set, the output 12 is set to 1.  
(Output 12 has to be configured as BIAS-output.)

### Note:

- The command is only valid with outputs defined as BIAS-outputs (function 2) in the configuration.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
72	X	Const.	00	00	00	00	00

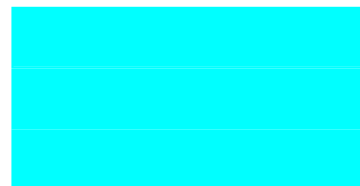
Parameters: D1 = Output pin number (see [explanation](#))  
D2 = Constant: 0/1

## 4 - 73 BIAS-Command "Output X = flag Y"

Output X = flag Y

**Explanation:**

The command "Output X = flag Y" assigns the state of the flag Y to the output.



Drive type:	Parameters:	Permissible inputs	
	Y:	0...255	<input checked="" type="checkbox"/> BIAS program
			<input checked="" type="checkbox"/> PLC program
			<input checked="" type="checkbox"/> Mathematics
631:	X:	5, 6 Pin no. at X10-clamp	
635/637/ 637+/637f:		12, 13, 20, 23 Pin no. at X10- female	
	X: I/O-Option board RP-EA5 (Slot B)	109, 110 Pin no. at COM2 Sub-D- female	
	X: I/O-Option board RP-EAE (Slot C)	209, 210, 217-227 Pin no. at X200 Sub-D-female	
637f:	X: 2 x CAN + 4 outputs and 4 inputs RP-2C8 (Slot B):	121 – 124 Pin no. at X120 clamp	

**Example:**

0 Flag 0 = input 2  
1 **Output 12 = flag 0**

**Result:**

The state of the input 2 is assigned to the output 12.  
(Output 12 has to be configured as BIAS-output.)

**Note:**

- The command is only valid with outputs defined as BIAS-outputs (function 2) in the configuration.

**Command coding:**

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

**Parameters:**

D1 = Output pin number: (see [explanation](#))  
D2 = Flag number: 0...255

#### 4 - 78 BIAS-Command "IBT-mask number = X, LED status [variable Y]"

##### Explanation:

The command "IBT mask number = " activates the selected screen mask of the IBT and the status of the function key LED's of the IBT.

Parameters:	Meaning:	Permissible inputs:	
X:	Mask number:	0...9999	<input checked="" type="checkbox"/> BIAS program
Y:		0..255	<input checked="" type="checkbox"/> PLC program
			<input checked="" type="checkbox"/> Mathematics

##### Example:

0 Variable 0 = 0

1 **IBT-Mask number = 4, LED- status = [variable 0]**

##### Result:

The main mask 4 on the IBT display is activated. All the LED's are switched off.

##### Note:

- The LED-status results of the sum of the following values

Key	On	flash
F1	32768	16384
F2	8192	4096
F3	2048	1024
F4	512	256
F5	128	64
F6	32	16

##### Command coding:

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

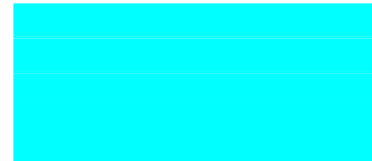
**Parameters:** D1,D2 = Number of the IBT Mask  
D3 = Variable number: 0...255



## 4 - 79 BIAS-Command "IBT-notification number = X,LED status[variable Y]"

**Explanation:**

The command "IBT notification number = " activates the notification of the specified message in the serial message buffer of the IBT and the status of the function key LED's of the IBT.



Parameters:	Meaning:	Permissible inputs:	
X:	Notification number	0...9999	<input checked="" type="checkbox"/> BIAS program
Z:	Dats set number	0..255	<input checked="" type="checkbox"/> PLC program
			<input checked="" type="checkbox"/> Mathematics

**Example:**

0 Variable 0 = 48  
 1 **IBT- notification number = 10, LED-status = [variable 0]**

**Result:**

The notification mask 10 on the IBT display is activated. The LED of the function key 6 flashes. All the other LED's are switched off.

**Note:**

- The specified mask must be known to the IBT.
- The LED-status results of the sum of the following values .

Kex	on	flash
F1	32768	16384
F2	8192	4096
F3	2048	1024
F4	512	256
F5	128	64
F6	32	16

**Command coding:**

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

**Parameters:** D1,D2 = Number of IBT notification  
 D3 = Variable number: 0...255

**4 - 7A BIAS-Command "CAN-command = [variable X]"**

CAN-command; start=[variable X ]

**Explanation:**

The command "CAN-command" allows to transmit or to receive data via the CAN-Bus.

**a. CAN-command transmit**

The variable range starts with the specified variable X and has a length of 5 variables.

The first variable specifies the identifier.

The bits 11 – 14 define the length of the command.

The bit no.14 selects between fixed and variable data length.

The bits 11 - 13 define the length binary coded.

(see bit coding of the identifier X)

The second to fifth variable includes the transmit data.

**b. CAN-command receive**

The variable range of the remote CAN -command starts with the specified variable X and has a length of 5 variables.

The first variable specifies the identifier with the Remote Bit (Bit 15 = 1) for the receive command.

The second variable selects the number of the variable, where the receive data is written in.

The receive data range uses 14 variables.

(see [variable range und data type](#))

**a.) CAN-command transmit**

Parameters:	Name:	Permissible inputs
X	Mode	216
X + 1	Data byte 0, 1	216
X + 2	Data byte 2 ,3	216
X + 3	Data byte 4 ,5	216
X + 4	Data byte 6 ,7	216

- BIAS program
- PLC program
- Mathematics

**Bit coding of mode X**

Bit 15	Bit 14		Bit 13	Bit 12	Bit 11	Bit 10 – 0
0 = transmit	0 = fixed length	8 Byte	22	21	20	Identifier 0...2047
	1 = variable length	0 to 7 Byte	bit coding length 0...7			
1 = Remote-request	-					Identifier 0...2047

**Example: Transmit Login**

```

0   [Var. 0] = 200
1   [Var. 1] = 1
3   [Var. 2] = 0
4   [Var. 3] = 0
5   [Var. 4] = 0
6   CAN-command; start = [variable 0]
7   ...
    
```

**Result:**

With execution of line 6 the identifier 200 with control word 1 (LOGIN) will be transmitted via the CAN-Bus.

## b.) CAN-command Receive

Parameters:	Name:	Permissible inputs
X	Identifier + Remote Bit (Bit 15)	0 ... 2047 + 32768
X + 1	Variable start address for receive data	0 ... 242
X + 2	reserved	0
X + 3	reserved	0
X + 4	reserved	0

The variable x+1 selects the number of the variable, where the receive data is written in.  
The 1st variable of the receive data field has the value -1 until the complete answer is received.

The following variables and data formats are used:

CAN	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7
Target var. 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
Target var. word	(X+1)+8		(X+1)+9		(X+1)+10		(X+1)+11	
Target var. longword	(X+1)+12				(X+1)+13			

### Example: Receive Status

```

1 [Variable 0 ] = 201
2 [Variable 0 ].bit 15 = 1, [flag 0 ]
3 [Variable 1 ] = 10
4 [Variable 2 ] = 0
5 [Variable 3 ] = 0
6 [Variable 4 ] = 0
7 CAN-command; start=[variable 0 ]
WAIT_CAN:
8 If [variable 10 ] == -1 than jump WAIT_CAN
9 ...

```

### Result:

With execution of line 7 the identifier 201 with remote bit will be transmitted via the CAN-Bus.  
The station with the transmit identifier 201 answers.  
When the variable 10 gets a value not equal -1 the received data will be stored starting with variable 10.

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			

### Note:

- Further information for the CAN-commando coding see CAN-BUS manual. (UL: 07-05-03-02... und UL: 07-05-03-03...)
- The coding of the data length is available with the 637f as of firmware V8.19
- The coding of the data length is available with the 631/ 635/ 637 as of firmware V6.19

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
7A	X	00	00	00	00	00	00

Parameters: D1 = Variable number: 0...251

## 4 - 7B BIAS-Command "IBT- data transfer"

IBT-data transfer, mode = X , recipe = Y, dataset = Z

### Explanations:

The command "IBT- data transfer" controls the dataset transfer between the IBT and the drive.

Mode 0: Transfer from the IBT to the drive

Mode 1: Transfer from the drive to the IBT

Parameters:	Name:	Permissible inputs:	
X:	mode:	0/1	<input checked="" type="checkbox"/> BIAS program
Y:	recipe number	1..250	<input checked="" type="checkbox"/> PLC program
Z:	dataset number	0..250	<input type="checkbox"/> Mathematics

### Example:

### Result:

### Note:

- The IBT must know the recipe number and the dataset number.

### Command code:

Code	D1	D2	D3	D4	D5	D6	D7
<b>7B</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	00	00	00	00

### Parameters:

D1 = Mode: 0/1  
 D2 = Recipe number: 0...250  
 D3 = Dataset: 0...250

## 4 - 7C BIAS-Command "CAN2-command = [variable X]"

**CAN2-command; start=[variable X ]**

### Explanation:

The command "CAN2-command = [variable X]" permits to receive or transmit a telegram by the CAN2-BUS.

It is allowed to transmit up to 8 byte with the objects 5 and 6

The receiver copies the data in the variables 246 and 247 (object 5) or 248 and 249 (object 6).

The transmitter transmits only the number of data as configured in the configuration.

Before the configuration of an object is changed, the mode "Off" must be turned on.

Parameters:	Name: Variable number:	Permissible input:	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
<b>X</b>	Parameter X:	0...251	
<b>Permissible variable content</b>			
<b>X:</b>	Configuration:		
<b>X+1:</b>	Data byte 0 - 3	232	
<b>X+2:</b>	Data byte 4 - 7	232	

Configuration bit coded								
Bit 31 -16 Delay time	Bit 15 – 12 Number of Data byte	Bit 11 - 8 Mode				Bit 7 Target	Bit 6 – 0 Identifier- offset	
Time until the next CAN2-command in the mode "Cyclic transmit" is transmitted. 1 = 0,844ms 65536 = 55,3s	0...8	27	26	25	24		0 = Objekt 5 from ID 300h 1 = Object 6 from ID 380h	0 ... 7F
		0	0	0	0	Off		
		0	0	0	1	receive		
		0	0	1	0	1 time transmit		
		0	1	1	0	Cyclic transmit		

### Example 1: CAN2-Receive object 5 with 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-command; start=[variable 20]

```

### Result:

The variable 0 is loaded with the value for the delay time 1s, the variable 1 with 8 for the data length of 8 byte, the variable 2 with 6 for the mode 6 and the variable 3 with identifier offset 7Fh.

The following 6 command combine the configuration in variable 20. The variable content is 0x04A0867F.

With the command "CAN2- command" in line 12 the object 5 is initialized as transmitter with the ID 37Fh and the variables 21 and 22 are transmitted cyclic every second.

**CAN2-Receive object 5**

5 [Variable 20 ] = 0x17F  
 6 **CAN2-command; start=[variable 20]**

**Result:**

The object 5 is switched on as a receiver with the ID 37Fh.

Transmitter	Var. 21	Var. 22
	Byte 0- 3	Byte 4-7
	0x150	0x 250



Var. 246	Var. 247	Receiver
Byte 0 – 3	Byte 4 - 7	
0x150	0x 250	

**Example 2: CAN2-Transmit object 6 with 6 Byte**

3 [Variable 20 ] = 0x6280  
 4 [Variable 21 ] = 0xFFEEDDC  
 5 [Variable 22 ] = 0x8866442  
 6 **CAN2-command; start=[variable 20]**

**Result:**

With the command “CAN2- command” in line 6 the object 6 once transmits the variables 21 and 22.

**CAN2- Receive object 6**

5 [Variable 20 ] = 0x180  
 6 **CAN2-command; start=[variable 20]**

**Result:**

The object 6 is switched on as a receiver with the ID 380h.

Transmitter	Var. 21	Var. 22
	Byte 0 - 3	Byte 4-7
	0xFFEEDDC C	0x8866442 2



Var. 248	Var. 249	Receiver
Byte 0 - 3	Byte 4 – 7	
0x FFEEDDC	0xYYYY4422	

YY = Bytes not change

**Example 3:CAN2-Transmit object 5 deactivate**

0 [Variable 0 ] = 0x0  
 1 **CAN2-command; start=[variable 20]**

**Result:**

The object 5 is switched off.  
 A new configuration is possible now.

**Note:**

- This command is available with the 637f as of firmware V8.19
- The user of this command has to take note of the bus load in combination with fixed CAN2 telegrams

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>7C</b>	<b>X</b>	00	00	00	00	00	00

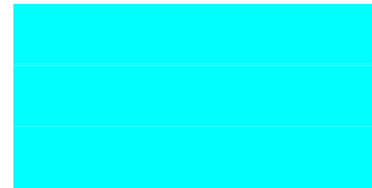
**Parameters:** D1 = variable number: 0...251

## 4 - 80 BIAS-Command "[Variable X] = const."

[Variable X ] = const.

**Explanation:**

The command "[Variable X] = " assigns the value of the constant to the variable X.



<b>Drive type:</b>	<b>Parameters:</b>	<b>Permissible inputs:</b>	<input checked="" type="checkbox"/> BIAS program
<b>All:</b>	<b>X:</b>	<b>0-255</b>	<input checked="" type="checkbox"/> PLC program
	<b>Y: Const.</b>	<b>± 231</b>	<input checked="" type="checkbox"/> Mathematics

**Example:**

0 [Variable 0] = 100000

**Result:**

The content of the variable 1 is 100000.

**Note:**

1. After switching the drive on all variables have the content 0.
2. The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

**Command coding:**

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

**Parameters:** D1 = Variable number:0...255  
D2-D5 = Constant: ± 231

## 4 - 81 BIAS-Command "If [variable X] ? const."

If [variable X] ? then jump Y

### Explanation:

The command "If [variable X] ? const." allows program branches by comparing the variable X with the constant.  
If the condition is true the execution will be continued at the specified command line.  
Otherwise the next BIAS command is executed.

Drive type:	Parameters	Permissible inputs:	
	:		<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
All:	X:	0...255	
	?:	> greater than (0) < lower than (1) == equal to (2) >= greater than or equal to (3) <= lower than or equal to (4) != not equal to (5) -> positive (6) -< negative (7)	
		value in () for command coding	
	Y: Address:	0...1499 The maximum permissible jump address corresponds to the length of the BIAS program. As an alternative to the line numbers labels (jump labels) can be used.	

### Example:

```

0 Start axis
1 Move position
2 Output 20 = 0
3 [Variable 0] = actual speed 1
4 If [variable 0] < 500 then jump 3
5 Output 20 = 1

```

### LOOP:

```

6 [Variable 0] = actual speed 1
7 If [variable 0] < 500 then jump 2
8 If [variable 0] < 2500 then jump LOOP
9 Jump 2

```

### Result:

The output 20 is set if the actual speed is greater or equal than 500 rpm and lower than 2500 rpm.

### Note:

1. The maximum permissible jump address corresponds to the length of the BIAS program.
2. As an alternative to the line numbers labels (jump labels) can be used.
3. After switching the drive on all variables have the content 0.
4. The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.
5. The comparison of both values is carried out by a subtraction of the constant factor from the contents of the variable.  
The conditional jump depends on the sign of the result.



## Command coding:

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

**Parameters:**  
 D1 = Variable number: 0...255  
 D2– D5= Constant: ± 231  
 D6.7...D6.4: Comparison condition: 0...7 (see [explanation](#))  
 D6.3..D6.0, D7: Address: 0...1499

## 4 - 82 BIAS-Command "[Variable X] = [variable Y] + const."

[Variable X ] = [variable Y ] + const.

### Explanation:

The command "[Variable X] = [variable Y] + const." adds the content of the variable Y to the constant and stores the sum in the variable X.

Drive type:	Parameters	Permissible inputs:	
	:		<input checked="" type="checkbox"/> BIAS program
			<input checked="" type="checkbox"/> PLC program
			<input checked="" type="checkbox"/> Mathematics
All:	X:	0...255	
	Y:	0...255	
	Const.:	± 231	

### Example:

```

0 If input 11 == 0 then jump 4
1 [Variable 0] = position
2 [Variable 0] = [variable 0] + 15000
3 Position = [variable 0]
4 Start axis
5 Move position

```

### Result:

If the input 11 is active, the axis moves 15000 increments further..

### Note:

1. After switching the drive on all variables have the content 0.
2. The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>82</b>	<b>X</b>	<b>Y</b>	<b>Z(low)</b>		<b>Z(high)</b>		<b>00</b>

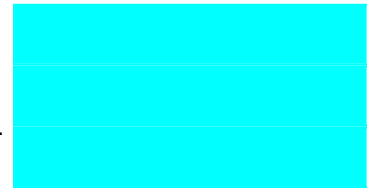
**Parameters:** D1 = Variable number 0...255  
D2 = Variable number 0...255  
D3 – D6 = Constant ± 231

## 4 - 83 BIAS-Command "[Variable X] = [variable Y] - const."

[Variable X ] = [variable Y ] – const.

### Explanation:

The command "[Variable X] = [variable Y] - const." subtracts the constant from the content of the variable Y and stores the difference in the variable X..



Drive type:	Parameters	Permissible inputs:	
	:		<input checked="" type="checkbox"/> BIAS program
			<input checked="" type="checkbox"/> PLC program
			<input checked="" type="checkbox"/> Mathematics
All:	X:	0...255	
	Y:	0...255	
	Const.:	± 231	

### Example:

```

0 If input 11 == 0 then jump 4
1 [Variable 0] = position
2 [Variable 0] = [variable 0] - 15000
3 Position = [variable 0]
4 Start axis
5 Move position
    
```

### Result:

If the input 11 is active, the axis moves 15000 fewer increments.

### Note:

- After switching the drive on all variables have the content 0.
- The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

### Command coding:

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

**Parameters:** D1 = Variable number 0...255  
 D2 = Variable number 0...255  
 D3 – D6 = Constant ± 231

#### 4 - 84 BIAS-Command "[Variable X] = [variable Y] \* const."

[Variable X ] = [variable Y ] \* const.

##### Explanation:

The command "[Variable X] = [variable Y] \* const." multiplies the content of the variable Y with the constant and stores the result in the variable X.

Drive type:	Parameters	Permissible inputs:	
	:		<input checked="" type="checkbox"/> BIAS program
			<input checked="" type="checkbox"/> PLC program
			<input checked="" type="checkbox"/> Mathematics
All:	X:	0...255	
	Y:	0...255	
	Const.:	± 231	

##### Example:

```

0 If input 11 == 0 then jump 4
1 [Variable 0] = position
2 [Variable 0] = [variable 0] * 5
3 Position = [variable 0]
4 Start axis
5 Move position

```

##### Result:

If the input 11 is active, the axis moves 5 times the position.

##### Note:

- After switching the drive on all variables have the content 0.
- The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

##### Command coding:

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

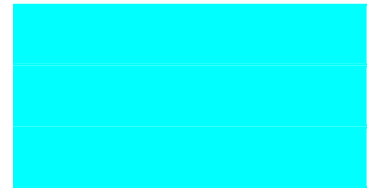
**Parameters:** D1 = Variable number 0...255  
 D2 = Variable number 0...255  
 D3 – D6 = Constant ± 231

## 4 - 85 BIAS-Command "[Variable X] = [variable Y] / const."

[Variable X ] = [variable Y ] / const.

### Explanation:

The command "[Variable X] = [variable Y] / const." divides the content of the variable Y with the constant and stores the integer result in the variable X.



Drive type:	Parameters	Permissible inputs:	
	:		<input checked="" type="checkbox"/> BIAS program
			<input checked="" type="checkbox"/> PLC program
			<input checked="" type="checkbox"/> Mathematics
All:	X:	0...255	
	Y:	0...255	
	Const.:	± 231	

### Example:

```

0 If input 11 == 0 then jump 4
1 [Variable 0] = position
2 [Variable 0] = [variable 0] / 5
3 Position = [variable 0]
4 Start axis
5 Move position
    
```

### Result:

If the input 11 is active, the axis moves 1/5 the position..

### Note:

- After switching the drive on all variables have the content 0.
- The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

### Command coding:

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

Parameters: D1 = Variable number 0...255  
 D2 = Variable number 0...255  
 D3 – D6 = Constant ± 231

#### 4 - 86 BIAS-Command "[Variable X] = flag Y"

[Variable X ] = flag Y , number = Z

##### Explanation:

The command "[Variable X] = flag Y, number = Z" copy the status of the desired Z flags starting with flag Y bit wise to the content of the variable X.(flag Y = bit 0).

Drive type:	Parameters	Permissible inputs:	
	:		<input checked="" type="checkbox"/> BIAS program
			<input checked="" type="checkbox"/> PLC program
			<input type="checkbox"/> Mathematics
All:	X:	0 - 255	
	Y:	0 – max. 255 -(number W-1)	
	Z:	1 - 32	

##### Example:

0 [Flag 0 ] = 0  
 1 [Flag 1 ] = 1  
 2 [Flag 2 ] = 0  
 3 [Flag 3 ] = 1  
 4 [Variable 10] = flag 0 , number = 3

##### Result

The value 5 (Flag 0 = 1 + Flag 2 = 4) is loaded into variable 10.

##### Note:

1. After switching the drive on all flags have the content 0.
2. The flag contents will be displayed cyclically in the "diagnosis", "BIAS" page.
3. Before the vaue is allocated the target variable is deleted,viz the not used bits are set to 0.

##### Command coding:

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

**Parameters:** D1 = Variable number: 0...255  
 D2 = Flag number: 0...255  
 D3 = Number: 1...32

## 4 - 87 BIAS-Command “[Variable X] = [variable Y].bit Z ; number”

[Variable X ] = [variable Y ].bit Z , number = W

### Explanation:

The command “[Variable X] = [variable Y].bit Z, number = W.” allocates the variable X on the bit position 0 to (number -1) bit wise with the value of the bit Z from the variable Y to Y+(number-1)

Drive type :	Parameters :	Permissible inputs:	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
All:	X:	0 - 255	
	Y:	0 – max. 255 -(number W-1)	
	Z:	0 - 31	
	W:	1 - 32	

### Example:

- 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 , number = 8
- 9 [Variable 9 ] = [Variable 0 ].Bit 1 , number = 8

### Result:

The variable 8 is loaded on the bit position 0 – 7 bit wise with the value of the bit 0 from the variables 0 – 7.

Variable 8 = 255d ; 0FFh, ...0011111111b

The variable 9 is loaded with the value of the bit 1

Variable 9 = 204d ; 0CCh, ..0010101010b

### Note:

- This command is available with the 637f as of firmware V8.19.
- This command is available with the 631, 635, 637 as of firmware V8.19
- After switching the drive on all variables have the content 0.
- The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.
- Before the vaue is allocated the target variable is deleted,viz the not used bits are set to

### Command coding:

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

- Parameter:**
- D1 = Variable number: 0...255
  - D2 = Variable number: 0...255
  - D3 = Bit number: 0...31
  - D4 = Number: 1...32

## 4 - 88 BIAS-Command "[Variable X] = [variable Y]"

[Variable X ] = [variable Y ]

### Explanation:

The command "[Variable X] = [variable Y]." assigns the content of the variable Y to the variable X.

Drive type:	Parameters	Permissible inputs:	
	:		<input checked="" type="checkbox"/> BIAS program
			<input checked="" type="checkbox"/> PLC program
			<input checked="" type="checkbox"/> Mathematics
All:	X:	0...255	
	Y:	0...255	

### Example:

0 [Variable 0] = position  
 1 [Variable 1] = [variable 0]  
 2 [Variable 0] = [variable 0] / 5  
 3 [Variable 0] = [variable 0] + [variable 1]  
 4 Position = [variable 0]

### Result:

The parameter position is multiplied by 1.2.

### Note:

1. After switching the drive on all variables have the content 0.
2. The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

### Command coding:

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

Parameters: D1 = Variable number: 0...255  
 D2 = Variable number: 0...255



## 4 - 89 BIAS-Command "If [variable X] ? [variable Y]"

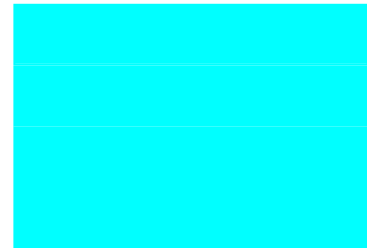
If [Variable X ] ? [variable Y ] then jump Z

### Explanation:

The command "If [variable X] ? [variable Y]" allows program branches by comparing the variable X with the variable Y.

If the condition is true the execution will be continued at the specified command line.

Otherwise the next BIAS command is executed.



Drive type:	Parameters	Permissible inputs:	
	:		<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
All:	X:	0...255	
	?:	> greater than (0) < lower than (1) == equal to (2) >= greater than or equal to (3) <= lower than or equal to (4) != bot equal to (5) -> positive (6) -< negative (7)	
		value in () for the command coding s.u.	
	Y:	0...255	
	Z: Address:	0...1499 The maximum permissible jump address corresponds to the length of the BIAS program. As an alternative to the line numbers, labels (jump labels) can be used.	

### Example:

```

0 Start axis
1 Move position
2 [Variable 1] = 500
3 [Variable 2] = 2500
4 Output 20 = 0
5 [Variable 0] = actual speed 1
6 If [variable 0] < [variable 1] then jump 5
7 Ausgang 20 = 1
LOOP:
8 [Variable 0] = actual speed 1
9 If [variable 0] < [variable 1] then jump 4
10 If [Variable 0] < [Variable 2] then jump LOOP
11 Jump 2
  
```

### Result:

The output 20 is set if the actual speed is greater or equal than 500 rpm and lower than 2500 rpm.

The maximum permissible jump address corresponds to the length of the BIAS program.

1. As an alternative to the line numbers, labels (jump labels) can be used.
2. After switching the drive on all variables have the content 0.
3. The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

4. The comparison of both values is carried out by a subtraction of the constant factor from the contents of the variable.  
The conditional jump depends on the sign of the result..

#### Command coding:

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

D1 = Variable number: 0...255  
 D2 = Comparison condition 0...7 ([see explanation](#))  
 D3 = Variable number: 0...255  
 D4,D5 = Address 0-1499

## 4 - 8A BIAS-Command “[Variable X] = [variable Y] + [variable Z]”

$$[\text{Variable X}] = [\text{variable Y}] + [\text{variable Z}]$$

### Explanation:

The command "[Variable X] = [variable Y] + [variable Z]" adds the content of the variable Z to the variable Y and stores the sum in the variable X.

Drive type:	Parameters	Permissible inputs	
	:		<input checked="" type="checkbox"/> BIAS program
			<input checked="" type="checkbox"/> PLC program
			<input checked="" type="checkbox"/> Mathematics
All:	X:	0...255	
	Y:	0...255	
	Z:	0...255	

### Example:

```

0   If input 11 == 0 then jump 5
1   [Variable 1] = 15000
2   [Variable 2] = position
3   [Variable 0] = [variable 0] + [variable 1]
4   Position = [Variable 0]
5   Start axis
7   Move position
  
```

### Result:

If the input 11 is active, the axis moves 15000 increments further.

### Note:

1. After switching the drive all variables have the content 0.
2. The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
8A	X	Y	Z	00	00	00	00

Parameters: D1 = Variable number: 0...255  
 D2 = Variable number: 0...255  
 D3 = Variable number: 0...255

#### 4 - 8B BIAS-Command “[Variable X] = [variable Y] - [variable Z]”

[Variable X ] = [variable Y ] - [variable Z ]

##### Explanation:

The command “[Variable X] = [variable Y] - [variable Z]” subtracts the content of the variable Z from the content of the variable Y and stores the difference in the variable X

Drive type:	Parameters:	Permissible inputs:	
All:	X:	0...255	<input checked="" type="checkbox"/> BIAS program
	Y:	0...255	<input checked="" type="checkbox"/> PLC program
	Z:	0...255	<input checked="" type="checkbox"/> Mathematics

##### Example:

```

0 If input 11 == 0 then jump 5
1 [Variable 1] = 15000
2 [Variable 0] = position
3 [Variable 0] = [variable 0] - [variable 1]
4 Position = [variable 0]
5 Start axis
6 Move position

```

##### Result:

If the input 11 is active, the axis moves 15000 fewer increments.

##### Note:

1. After switching the drive on all variables have the content 0.
2. The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

##### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>8B</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	00	00	00	00

**Parameters:** D1 = Variable number: 0...255  
D2 = Variable number: 0...255  
D3 = Variable number: 0...255

## 4 - 8C BIAS-Command “[Variable X] = [variable Y] \* [variable Z]”

[Variable X ] = [variable Y ] \* [variable Z ]

### Explanation:

The command "[Variable X] = [variable Y] \* [variable Z]" multiplies the content of the variable Y with the content of the variable Z and stores the result in the variable X.

Drive type:	Parameters:	Permissible inputs:	
All:	X:	0...255	<input checked="" type="checkbox"/> BIAS program
	Y:	0...255	<input checked="" type="checkbox"/> PLC program
	Z:	0...255	<input checked="" type="checkbox"/> Mathematics

### Example:

```

0   If Input 11 == 0 then jump 5
1   [Variable 1] = 5
2   [Variable 2] = position
3   [Variable 0] = [variable 0] * [variable 1]
4   Weg = [variable 0]
5   Start axis
7   Move position
  
```

### Result:

If the input 11 is active, the axis moves 5 times the position.

### Note:

- After switching the drive on all variables have the content 0.
- The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>8C</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	00	00	00	00

**Parameters:** D1 = Variable number: 0...255  
 D2 = Variable number: 0...255  
 D3 = Variable number: 0...255

#### 4 - 8D BIAS-Command “[Variable X] = [variable Y] / [variable Z]”

[Variable X ] = [variable Y ] / [variable Z ]

##### Explanation:

The command "[Variable X] = [variable Y] / [variable Z]" divides the content of the variable Y with the content of the variable Z and stores the integer result in the variable X.

Drive type:	Parameters:	Permissible inputs:	
All:	X:	0...255	<input checked="" type="checkbox"/> BIAS program
	Y:	0...255	<input checked="" type="checkbox"/> PLC program
	Z:	0...255	<input checked="" type="checkbox"/> Mathematics

##### Example:

```

0   If input 1 == 0 then jump 5
1   [Variable 1] = 5
2   [Variable 2] = position
3   [Variable 0] = [variable 0] / [variable 1]
4   Position = [variable 0]
5   Start axis
7   Move position

```

##### Result:

If the input 11 is active, the axis moves 1/5 the position.

##### Note:

1. The result of the division has no remainder.
2. After switching the drive on all variables have the content 0.
3. The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

##### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
8D	X	Y	Z	00	00	00	00

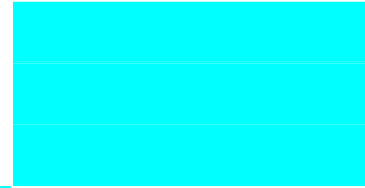
Parameter: D1 = Variable number: 0...255  
D2 = Variable number: 0...255  
D3 = Variable number: 0...255

## 4 - 8E BIAS-Command “[Teachvariable X] = [variable Y]”

[Teachvariable X ] = [variable Y ]

### Explanation:

The BIAS command "[Teachvariable X] = [variable Y]" assigns the contents of variable Y to the permanently stored variable X (teach variable).



Drive type:	Parameters	Permissible inputs	
	:		<input checked="" type="checkbox"/> BIAS program
All:	X:	0...15	<input type="checkbox"/> PLC program
	Y:	0...255	<input type="checkbox"/> Mathematics

### Example:

0 [Variable 0] = actual position 1  
 1 [Teachvariable 1] = [variable 0]

### Result:

The current actual position 1 is stored permanently in the teach variable 1. It can for example be used later as a target position.

### Note:

- During the execution of this command the BIAS processing is stopped for approx. 10 ms.
- Because of the limitation of save cycles from the EEPROM this command must not executed cyclically (appr. 100.000 save cycles)

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
8E	X	Y	00	00	00	00	00

Parameters: D1 = Variable number: 0...15  
 D2 = Variable number: 0...255

## 4 - 8F BIAS-Command “[Variable X] = [teachvariable Y]”

[Variable X ] = [teachvariable Y ]

### Explanation:

The command “[Variable X] = [teachvariable Y]” assigns the content of the permanently stored variable Y (teach variable) to the variable X.

Drive type:	Parameters	Permissible inputs:	
	:		<input checked="" type="checkbox"/> BIAS program
			<input checked="" type="checkbox"/> PLC program
			<input checked="" type="checkbox"/> Mathematics
All:	X:	0...255	
	Y:	0...15	

### Example:

0 [Variable 0] = [teachvariable 1]  
1 Position = [variable 0]

### Result:

The content of the teach variable 1 is loaded as setpoint position.

### Note:

- After switching the drive on the contents of all teach variables are automatically preloaded.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
8F	X	Y	00	00	00	00	00

Parameters: D1 = Variable number: 0...255  
D2 = Variable number: 0...15



## 4 - 90 BIAS-Command "Mathematic program"

**Explanation:**

The command "Mathematics program" starts the execution of a third task. This task will be executed in the time that isn't used for the interrupt execution. The execution time of the math-program is not determined.

<b>Drive type:</b>	<b>Parameters</b> :	<b>Permissible inputs:</b>	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input type="checkbox"/> Mathematics
<b>All:</b>	Address:	<b>0 – 1499</b>	
		The maximum permissible jump address corresponds to the length of the BIAS-program. As an alternative to the line numbers labels (jump label) can be used.	

**Example:**

```
0 End of program, mode = 3
1 Mathematics program Math
```

**Result:**

The executed Mathematics program is terminated and then the new program at the label Math started.

**Note:**

- The execution of this command is only permitted, if no MATH-program is executed. In any other case the execution of the calling task is terminated with the error message "start error".
- In the mathematic task are not all BIAS-Command permitted
- In the BIAS-program definitions (Parameters“ Program reaction to drive deactivation/ activation) can set the reaction of the mathematic task at deactivation of the drive. So the mathematic task can continue to work after the deactivation of the drive.
- In this mode the command „Mathematic program“ is executed at the first or second line (if the PLC program is on line 1), if the drive is not enabled.
- **Attention at the Mode “Continue execution after reactivation“!!**  
If you download a new BIAS-Program with modifications in the line numbers, the mechanic can make uncontrolled movements

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>90</b>	<b>X(low)</b>	<b>X(high)</b>	00	00	00	00	00

**Parameters:** D1, D2 : Address 0...1499

## 4 - 91 BIAS-Command "Profile initialization"

Profile init.; profilnr.= X ,supp. point = Y, start address = Z

### Explanation:

The command "profile initializing" initializes the specified synchronous parameter block. This command should be used in combination with the command "profile [cycle length](#)".

Drive type:	Parameters:	Permissible inputs:	
All:	X: Profile number	0...15	<input type="checkbox"/> BIAS program
	Y: Supporting points	1...1024	<input type="checkbox"/> PLC program
	Z: Start address	0...2047	<input checked="" type="checkbox"/> Mathematics

### Example:

ProgStart:

0 Profile init.;profilnr.=0 ,supp. point =128, start address =0

### Result:

The profile is initialized with 128 supporting points and it starts address 0

### Note:

The values can load with the command "Profile [value, profile X , value Y = \[variable Z \]](#)", too.

### Command coding:

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

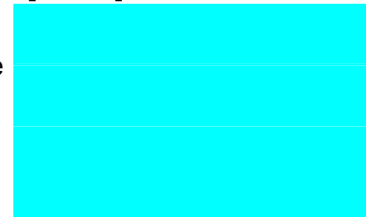
**Parameters:** D1: Profile number: 0...15  
 D2, D3: Supporting points: 1...1024  
 D4, D5: Start address: 0...2047

## 4 - 92 BIAS-Command "Profile cycle length = [variable X]"

Profile cycle length; profile no.= X ,master=[var. Y ] ,slave=[var. Z ]

### Explanation::

The command "profile length" loads the cycle length for the master and slave of the specified synchronous profile block with the contents of the variables. This command should be used in combination with the command "Profile [initialization](#)".



Drive type:	Parameters:	Permissible inputs:	
All:	X: Profile number	0...15	<input type="checkbox"/> BIAS program
	Y:	0...255	<input type="checkbox"/> PLC program
	Z:	0...255	<input checked="" type="checkbox"/> Mathematics

### Example:

ProgStart:

0 [Variable 10 ] = 100000

1 [Variable 11 ] = 80000

**2 Profile cycle length; profile no.= 0 ,master=[var. 10] ,slave=[var. 11 ]**

### Result:

The profil 0 is initialized with a master cycle length of 100000 increments and a slave cylce length of 80000 increments.

### Note:

The values can load with the command "Profile [value, profile X , value Y = \[variable Z \]](#)", too.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>92</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	00	00	00	00

**Parameters:** D1 = Profile number: 0...15  
 D2 = Variable number: 0...255  
 D3 = Variable number: 0...255

## 4 - 93 BIAS-Command "[Variable X] = profile value"

[Variable X] = profile value, profile Y, value Z

### Explanation:

The command "[Variable X]=profile value, profile Y value Z" loads the desired parameter Z of the cam profile Y in the variable X.

Drive type:	Parameters:	Permissible inputs:	
All:	X:	0...255	<input type="checkbox"/> BIAS program
	Y: Profile number	0...15	<input type="checkbox"/> PLC program
	Z:	0      Number of supporting points 1      Address of first supporting point 2      Master cycle length 3      Slave cycle length    Reserve	<input checked="" type="checkbox"/> Mathematics

### Example:

ProgStart:

0 [Variable 0 ] = profile value, profile 0 , value 0  
 1 [Variable 1 ] = profile value, profile 0 , value 1  
 2 [Variable 2 ] = profile value, profile 0 , value 2  
 3 [Variable 3 ] = profile value, profile 0 , value 3

### Result:

The parameters are loaded in the variables 0 – 3.

### Note :

With this command you can check imported profile for valid valuation.

### Command coding:

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

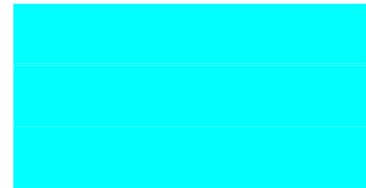
**Parameters:** D1 = Variable number: 0...255  
 D2 = Profile number: 0...15  
 D3 = Value: 0...99

## 4 - 94 BIAS-Command "Profile value = [variable X]"

Profile value, profile X , value Y = [variable Z ]

### Explanation:

The command "Profile value, profile X, value Y=[variable X]" stores the content of the variable Z in the desired value Y of the cam profile X.



Drive type:	Parameters:	Permissible inputs:	
All:	X: Profile number	0...15	<input type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Y:	0 Number of supporting points 1 Address of first supporting point 2 Master cycle length 3 Slave cycle length 4...99 Reserve	
	Z:	0...255	

### Example:

```
ProgStart:
0 [Variable 10 ] = 128
1 Profile value, profile 0 , value 0 = [variable 10 ]
2 NOP
```

### Result:

The profile 0 has 128 supporting points.

### Note:

- Modification of profile parameters are not permitted!

### Command coding:

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

Parameters: D1 = Profile number: 0...15  
 D2 = Value 0...99  
 D3 = Variable number 0...255

## 4 - 97 BIAS-Command " Save table"

### Explanation:

The command "Save table" starts the saving of the entire profile table. For the execution of this command it is necessary to interrupt the PLC-program execution. It means the execution time of this command is larger than the 2ms default value.

Drive type :	Parameters:	Permissible inputs:	
All:	None	---	<input type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input type="checkbox"/> Mathematics

### Example:

```

0   If Flag 1 == 0 then jump NotSave
1   End of program 4
2   Save table
3   End of program 5
4   Flag 1 = 0
NotSave:

```

### Result:

If the flag 1 is set, the table will save. Beside the the drive is deactivated, the table save and the drive is activated again..

### Note:

- The execution of this command is possible only in the deactivated status of the servo drive.
- I.e. for the execution mode of the PLC program the value 2 "go on execution on deactivation" must be configured.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>97</b>	00	00	00	00	00	00	00

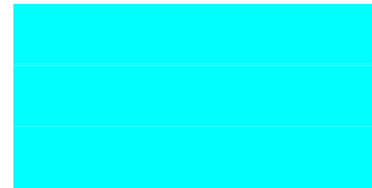
### Parameters:

## 4 - A0 BIAS-Command "Table ([variable X])= const."

Table ([variable X ]) = const.

**Explanation:**

The command "Table([var. X]) = const." loads the constant value into the profile memory address specified by the variable X.



Drive type	Parameters:	Permissible inputs:	
All:	X:	0-255 Permitted variable content 0...4095	<input type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Y: const.	± 231	

**Example:**

```
ProgStart:
0 [Variable 10 ] = 0
1 Table([variable 10 ])= 1000
2 NOP
```

**Result:**

The value 1000 is loaded into the profile memory address 0.

**Note:**

- The command "Save [table](#)" is able to save values of the table permanent.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
A0	X	Y(low)			Y(high )	00	00

**Parameters:** D1 = Variable number: 0...255  
D2-D5 = Constant : ± 231

#### 4 - A1 BIAS-Command "Table ([variable X]) = [Y variable Z]"

Table ([variable X]) = [Y variable Z]

##### Explanation:

The command "Table([variable X]) = [Y variable Z]" loads the contents of the Y type variable Z into the profile memory address specified by the variable X.

Drive type:	Parameters:	Permissible inputs	
All:	X:	0-255 Permitted variable content: 0...4095	<input type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Y: type selection	L = Long Integer F = Float D = Double Float	
	Z:	0...255	

##### Example:

```
ProgStart:
0 [Variable 0 ] = 1000
1 [Variable 10 ] = 0
2 Table([variable 10 ])= [L variable 0 ]
3 NOP
```

##### Result:

The value is loaded into the profile memory address 0.

##### Note:

1. The command "Save [table](#)" is able to save the values of the table permanent.
2. A type conversion to type Long Integer will be executed before assigning a type variable.
3. 3 different type variables are available to you.  
All type variables are stored in different memory areas.  
The long integer variables 0...255 corresponds to the BIAS-variables 0...255.  
The float variables 0...255 corresponds to the 4-byte IEEE format.  
The double float variables 0...255 corresponds to the 8-byte IEEE format.

##### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
A1	X	Y	Z	00	00	00	00

**Parameters:** D1 = Variable number: 0...255  
D2 = Data type selection: (Long = 0, Float = 1, Double = 2)  
D3 = Variable number: 0...255

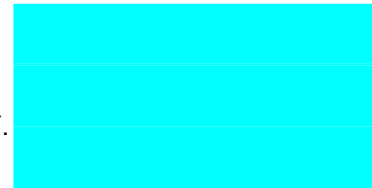


## 4 - A2 BIAS-Command “ [X variable Y] = table ([variable Z])“

[X variable Y] = table ([variable Z])

### Explanation:

The command "[X variable Y] = table([variable Z])" loads the contents of the profile memory address specified by the variable Z into the X type variable Y.



Drive type:	Parameters:	Permissible inputs	
All:	X: Type selection	L = Long Integer F = Float D = Double Float	<input type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Y:	0-255	
	Z:	0-255 Permitted variable content 0...4095	

### Example:

```
ProgStart:
0 [Variable 10 ] = 0
1 [L variable 5 ] = table([variable 10 ])
2 NOP
```

### Result:

The value of the profile memory address 10 is loaded into the long integer variable 5.

### Note:

- The command "Save [table](#)" is able to save the values of the table permanent.
- A type conversion to type Long Integer will be executed before assigning a type variable.
- 3 different type variables are available to you.  
All type variables are stored in different memory areas.  
The long integer variables 0...255 corresponds to the BIAS-variables 0...255.  
The float variables 0...255 corresponds to the 4-byte IEEE format.  
The double float variables 0...255 corresponds to the 8-byte IEEE format.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
A2	X	Y	Z	00	00	00	00

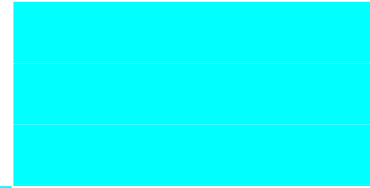
**Parameters:** D1 = Type selection (Long = 0, Float = 1, Double = 2)  
D2 = Variable number: 0...255  
D3 = Variable number: 0...255

**4 - A3 BIAS-Command “[W variable X] = [Y variable Z]”**

[W variable X] = [Y variable Z]

**Explanation:**

The command “[W variable X] = [Y variable Z]” loads the contents of the Y type variable Z into the W type variable X.



Drive type:	Parameters:	Permissible inputs:	
All:	W: Type selection	L = Long integer F = Float D = Double float	<input type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	X:	0-255	
	Y: Type selection	L = Long integer F = Float D = Double float	
	Z:	0-255	

**Example:**

```

0 [D_variable 0 ] = 0
1 [D_variable 1 ] = 0.250000000
2 If [D_variable 1 ] < [D_variable 0 ] then jump error
3 [D_variable 2 ] = SQRT([D_variable 1 ])
4 [L_variable 1 ] = [D_variable 2 ]
...
error:
50 ...
    
```

**Result:**

The double variable 2 is loaded into the long integer variable 1.

**Note:**

7. A type conversion to type Long Integer will be executed before assigning a type variable.
8. 3 different type variables are available to you.  
 All type variables are stored in different memory areas.  
 The long integer variables 0...255 corresponds to the BIAS-variables 0...255.  
 The float variables 0...255 corresponds to the 4-byte IEEE format.  
 The double float variables 0...255 corresponds to the 8-byte IEEE format.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
A3	W	X	Y	Z	00	00	00

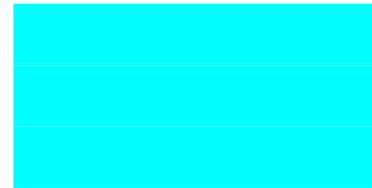
**Parameters:** D1 = Type selection: (Long = 0, Float = 1, Double = 2)  
 D2 = Variable number: 0...255  
 D3 = Type selection: (Long = 0, Float = 1, Double = 2)  
 D4 = Variable number: 0...255

## 4 - A4 BIAS-Command "[X variable Y] = const."

[X variable Y] = const.

**Explanation:**

The command "[X variable Y] = const." loads the constant value into the X type variable Y.



Drive type:	Parameters:	Permissible inputs	
All:	X: Type selection:	L = Long integer F = Float D = Double float	<input type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Y:	0-255	
	Const.	± 231	

**Example:**

```

0 [D_variable 0 ] = 0
1 [D_variable 1 ] = 0.25000000
2 If [D_variable 1 ] -< [D_variable 0 ] the jump Error
3 [D_variable 1 ] = SQRT([D_variable 1 ])
4 ...
Error:
50 ...
    
```

**Result:**

If the content of the type variable 1 isn't negative, the square root is calculated and the result is stored back into the type variable 1. If the content of the type variable is negative, the BIAS execution pointer is set at the label Error (line 50).

**Note:**

- It is possible to use a constant floating point value for the assignment of a float or double float variable.
- 3 different type variables are available to you. All type variables are stored in different memory areas. The **long integer** variables 0...255 corresponds to the BIAS-variables 0...255. The **float variables** 0...255 corresponds to the 4-byte IEEE format. The **double float** variables 0...255 corresponds to the 8-byte IEEE format.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
A4	X	Y	Z(low)			Z(high)	00

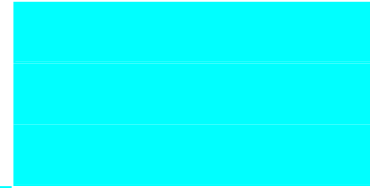
**Parameters:** D1 = Type selection: (Long = 0, Float = 1, Double = 2)  
D2 = Variable number: 0...255  
D3 – D6 = Constant: ± 231

**4 - A5 BIAS-Command "[Variable [X]] = const."**

[Variable [X]] = const.

**Explanation:**

The command "[Variable [X]] = const " assigns the constant value to the variable addressed by the content of the variable X.



Drive type:	Parameters:	Permissible inputs	
All:	X:	0...255 permitted variable content: 0...255	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Const.	± 231	

**Example:**

0 [Variable 0] = 10  
 1 [Variable [0]] = 10000

**Result:**

The variable 10 is loaded with the value 10000.

**Note :**

1. After switching the drive on all variables have the content 0.
2. The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
A5	X	Y(low)			Y(high )	00	00

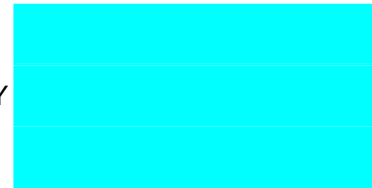
**Parameters:** D1 = Variable number: 0...255  
 D2, D3 = Constant: ± 231

## 4 - A6 BIAS-Command "[Variable [X]] = [variable Y]

[Variable [X]] = [variable Y]

### Explanation:

The command "[Variable [X]] = [variable Y]" assigns the content of variable Y to the variable addressed by the content of the variable X.



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

### Example:

0 [Variable 0] = 10  
 1 [Variable 1] = 100000  
 2 [Variable [0]] = [variable 1]

### Result:

The content of the variable 10 is 100000.

### Note:

1. After switching the drive on all variables have the content 0
2. The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>A6</b>	<b>X</b>	<b>Y</b>	00	00	00	00	00

**Parameters:** D1 = Variable number: 0...255  
 D2 = Variable number: 0...255

## 4 - A7 BIAS-Command "[Variable X] = [variable [Y]]"

[Variable X] = [variable [Y]]

### Explanation:

The command "[Variable X] = [variable [Y]]" assigns the content of variable addressed by the content of the variable Y to the variable X.

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

### Example:

0 [Variable 0] = 10  
1 [Variable 10] = 100000  
2 [Variable 1] = [variable [0]]

### Result:

The content of the variable 1 is 100000.

### Note:

1. After switching the drive on all variables have the content 0
2. The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>A7</b>	<b>X</b>	<b>Y</b>	00	00	00	00	00

**Parameters:** D1 = Variable number: 0...255  
D2 = Variable number: 0...255

## 4 - A8 BIAS-Command “[Variable X ] = [variable Y ] ? [variable Z]“

**Explanation:**

The command “[Variable X ] = [variable Y ] ? [variable Z]“ allocates the result from the logical operation between the selected variables Y and Z into the variable X.

Drive type :	Parameters:	Permissible inputs:	
All:	X: Y:	0...255 0...255	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Z:	0...255	
	?	& AND (11)   OR (12) >> Shift right (13) << Shift left (14) rl Rotate left (15) rr Rotate right (16) ^ XOR (17) Value in ( ) for the command coding  Permitted variable content of the variable Z for command coding 13 – 16 1...31	

**Example:**

0 [Variable 0] = 1  
 1 [Variable 1] = 15  
 2 [Variable 2] = [variable 0] << [variable 1]

**Result:**

The variable 0 is shifted 15 Bit to left. The content of the variable 2 is 65536.

**Note :**

- This command is available with the 637f as of firmware V8.19
- This command is available with the 631/635/637 as of firmware V6.19
- After switching the drive on all variables have the content 0
- The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
A8	X	Y	?	Z	00	00	00

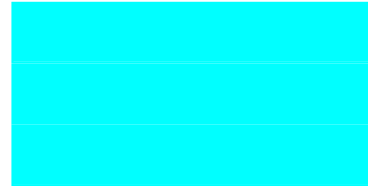
**Parameters:** D1 = Variable number: 0...255  
 D2 = Variable number: 0...255  
 D3 = Logic operation (11...17) (see [explanation](#))  
 D4 = Variable number: 0...255

**4 - A9 BIAS-Command “[Variable X ] = [variable Y ] ? const.”**

[Variable X ] = [variable Y ] ? Z

**Explanation:**

The Command “[Variable X ] = [variable Y ] ? const.” allocates the result from the logical operation between the selected variable Y and the constant into the variable X.



Drive type :	Parameters:	Permissible inputs:	<input checked="" type="checkbox"/> BIAS program <input checked="" type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
All:	X: Y:	0...255 0...255	
	Const.:	± 231	
	?	& AND (11)   OR (12) >> Shift right (13) << Shift left (14) rl Rotate Left (15) rr Rotate right (16) ^ XOR (17) Value in ( ) for the command coding s.u. Permitted variable content of the variable Z for command coding 13 – 16 1...31	

**Example:**

0 [Variable 0] = 0x0000 00FF  
 1 [Variable 1] = [variable 0] | 0x0000 FF00

**Result:**

The variable 0 is combined logic OR with the constant and the result is saved into the variable 1. The content of the variable 1 is 65535.

**Note:**

- This command is available with the 637f as of firmware V8.19
- This command is available with the 631/635/637 as of firmware V6.19
- After switching the drive on all variables have the content 0
- The variable contents will be displayed cyclically in the "diagnosis", "BIAS" page.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
A9	X	Y	?	Z (low)			Z (high)

**Parameters:** D1 = Variable number: 0...255  
 D2 = Variable number: 0...255  
 D3 = Logic operation (11...17) (see [explanation](#))  
 D4 - D7 = Constant: ± 231 exception D3 = 13, 14, 15, 16



## 4 - B0 BIAS-Command "[D\_variable X]=[D\_variable Y]+[D\_variable Z]"

$$[D\_variable X] = [D\_variable Y] + [D\_variable Z]$$

### Explanation:

The command "[D\_var. X] = [D\_var.Y] + [D\_var.Z]" adds the contents of the double variable Z to the contents of the double variable Y and stores the sum into the double variable X.

Drive type:	Parameters:	Permissible inputs:	
All:	X:	0...255	<input type="checkbox"/> BIAS program
	Y:	0...255	<input type="checkbox"/> PLC program
	Z:	0...255	<input checked="" type="checkbox"/> Mathematics

### Example:

0 [D\_variable 0 ] = 1.250000000

1 [D\_variable 1 ] = 0.250000000

2 [D\_Variable 2 ] = [D\_variable 0 ] + [D\_variable 1 ]

### Result:

The double variable 0 and 1 are added and the sum is stored in the double- variable 2.

### Note:

1. After switching on all variables have the content 0.
2. The double variable contents will be displayed cyclically in the "diagnosis", "Mathematics" page.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
B0	X	Y	Z	00	00	00	00

**Parameters:** D1: Variable number: 0...255  
 D2: Variable number: 0...255  
 D3: Variable number: 0...255

#### 4 - B1 BIAS-Command "[D\_variable X]=[D\_variable Y]-[D\_variable Z]"

$$[D\_variable\ X] = [D\_variable\ Y] - [D\_variable\ Z]$$

##### Explanation:

The command "[D\_var. X] = [D\_var.Y] - [D\_var.Z]" subtracts the contents of the double variable Z from the contents of the double variable Y and stores the difference into the double variable X.

Drive type:	Parameters:	Permissible input:	
All:	X:	0...255	<input type="checkbox"/> BIAS program
	Y:	0...255	<input type="checkbox"/> PLC program
	Z:	0...255	<input checked="" type="checkbox"/> Mathematics

##### Example:

0 [D\_variable 0 ] = 1.250000000

1 [D\_variable 1 ] = 0.250000000

2 [D\_variable 2 ] = [D\_variable 0 ] - [D\_variable 1 ]

##### Result:

The contents of the double variable 1 is subtracted from the contents of the double variable 0. The difference is stored in the double variable 2

##### Note:

- After switching on all variables have the content 0.
- The double variable contents will be displayed cyclically in the "diagnosis", "Mathematics" page.

##### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>B1</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	00	00	00	00

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

D2: Variable number: 0...255

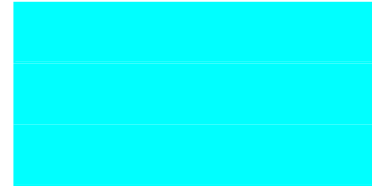
D3: Variable number: 0...255

## 4 - B2 BIAS-Command "[D\_variable X] = [D\_variable Y] \* [D\_variable Z]"

$$[D\_variable\ X] = [D\_variable\ Y] * [D\_variable\ Z]$$

### Explanation:

The command "[D\_var. X] = [D\_var.Y] \* [D\_var.Z]" multiplies the contents of the double variable Z with the contents of the double variable Y and stores the product into the double variable X.



Drive type:	Parameters:	Permissible inputs:	
All:	X:	0...255	<input type="checkbox"/> BIAS program
	Y:	0...255	<input type="checkbox"/> PLC program
	Z:	0...255	<input checked="" type="checkbox"/> Mathematics

### Example:

0 [D\_variable 0 ] = 1.250000000

1 [D\_variable 1 ] = 0.250000000

2 [D\_variable 2 ] = [D\_variable 0 ] \* [D\_variable 1 ]

### Result:

The content of the double variable 0 and 1 are multiplied and the product is stored into the double variable 2.

### Note:

- After switching on all variables have the content 0.
- The double variable contents will be displayed cyclically in the "diagnosis", "Mathematics" page.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>B2</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	00	00	00	00

**Parameters:** D1: Variable number: 0...255  
 D2: Variable number: 0...255  
 D3: Variable number: 0...255

#### 4 - B3 BIAS-Command "[D\_variable X] = [D\_variable Y] / [D\_variable Z]

$$[D\_variable\ X] = [D\_variable\ Y] / [D\_variable\ Z]$$

##### Explanation:

The command "[D\_var. X] = [D\_var.Y] / [D\_var.Z]" divides the contents of the double variable Z by the contents of the double variable Y and stores the quotient into the double variable X.

Drive type:	Parameters:	Permissible inputs:	
All:	X:	0...255	<input type="checkbox"/> BIAS program
	Y:	0...255	<input type="checkbox"/> PLC program
	Z:	0...255	<input checked="" type="checkbox"/> Mathematics

##### Example:

0 [D\_variable 0 ] = 1.250000000

1 [D\_variable 1 ] = 0.250000000

2 [D\_variable 2 ] = [D\_variable 0 ] / [D\_variable 1 ]

##### Result:

The content of the double variable 1 is divided by the content of the double variable 0 and the quotient is stored into the double variable 2.

##### Note:

- After switching on all variables have the content 0.
- The double variable contents will be displayed cyclically in the "diagnosis", "Mathematics" page.

##### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
<b>B3</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	00	00	00	00

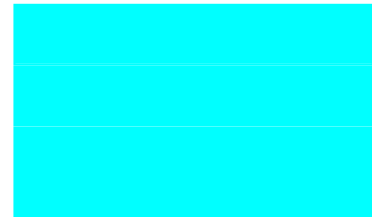
**Parameters:** D1 = Variable number: 0...255  
 D2 = Variable number: 0...255  
 D3 = Variable number: 0...255

## 4 - B4 BIAS-Command "If [D\_variable X] ? [D\_variable Y]"

If [D\_variable X] ? [D\_variable Y] then jump Z

**Explanation:**

The command "If [D\_var.X] ? [D\_var.Y]" allows program branches by comparing the double variable X with the double variable Y. If the condition is true the execution will be continued at the specified command line. Otherwise the next BIAS command is executed.



Drive type:	Parameters:	Permissible inputs:	
All:	X:	0...255	<input type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	?:	> greater than (0) < lower than (1) == equal to (2) >= greater than or equal to(3) <= lower than or equal to (4) != not equal to (5) > positive (6) < negative (7)	
		value in ( ) for the command coding s.u.	
	Y:	0...255	
	Z: address:	0...1499 The maximum permissible jump address corresponds to the length of the BIAS program. As an alternative to the line numbers labels (jump labels) can be used.	

**Example:**

```

0 [D_variable 0 ] = 0
1 [D_variable 1 ] = 0.250000000
2 If [D_variable 1 ] < [D_variable 0 ] then jump error
3 [D_variable 1 ] = SQRT([D_variable 1 ])
4 ...
error:
50 ...
  
```

**Result:**

If the double variable 1 is not negative, the program calculates the square root of the double-variable 1 and stores the result back. If the double variable 1 is negative the execution continue at the label "error"

**Note:**

5. The maximum permissible jump address corresponds to the length of the BIAS program.
6. As an alternative to the line numbers labels (jump labels) can be used.
7. After switching on all variables have the content 0.
8. The double variable contents will be displayed cyclically in the "diagnosis", "Mathematics" page.
9. The comparison of both, values is carried out by a subtraction of the variable Y from the variable X. The conditional jump depends on the sign of the result.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
<b>B4</b>	<b>X</b>	<b>?</b>	<b>Y</b>	<b>Z(low)</b>	<b>Z(high)</b>	00	00

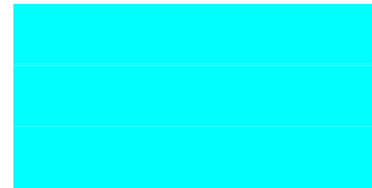
**Parameters:** D1 = Variable number: 0...255  
 D2 = Condition: 0...7 (see [explanation](#))  
 D3 = Variable number: 0...255  
 D4, D5 = address: 0...1499

## 4 - B5 BIAS-Command "[D\_variable X] = SIN ([D\_variable Y])

$$[D\_variable\ X] = SIN([D\_variable\ Y])$$

**Explanation:**

The command "[D\_var. X] = SIN([D\_var.Y])" calculates the sinus of the double-variable Y and stores the result in the double-variable X.



Drive type:	Parameters:	Permissible inputs:	
All:	X:	0...255	<input type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Y:	0...255	

**Example:**

- 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 ])

**Result:**

The example calculates the sinus of 30°. At first the radian is calculated and stored in the double variable 3. After this the sinus value is calculated and the result is stored in the double variable 0.

**Note:**

- 9. After switching on all variables have the content 0.
- 10. The double variable contents will be displayed cyclically in the "diagnosis", "Mathematics" page.

**Command coding:**

Code	D1	D2	D3	D4	D5	D6	D7
B5	X	Y	00	00	00	00	00

**Parameters:** D1 = Variable number: 0...255  
 D2 = Variable number: 0...255

#### 4 - B6 BIAS-Command "[D\_variable X] = COS ([D\_variable Y])

[D\_variable X] = COS([D\_variable Y])

##### Explanation:

The command "[D\_var. X] = COS([D\_var.Y])" calculates the cosinus of the double variable Y and stores the result in the double-variable X.

Drive type:	Parameters:	Permissible inputs:	
All:	X:	0...255	<input type="checkbox"/> BIAS program
	Y:	0...255	<input type="checkbox"/> PLC program
			<input checked="" type="checkbox"/> Mathematics

##### Example:

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 ])

##### Result:

The example calculates the cosinus of 30°. At first the radian is calculated and stored in the double variable 3. After this the cosinus value is calculated and the result is stored in the double variable 0.

##### Note:

- After switching on all variables have the content 0.
- The double variable contents will be displayed cyclically in the "diagnosis", "Mathematics" page.

##### Commands coding:

Code	D1	D2	D3	D4	D5	D6	D7
B6	X	Y	00	00	00	00	00

**Parameters:** D1 = Variable number: 0...255  
 D2 = Variable number: 0...255

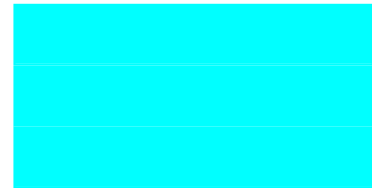


## 4 - B7 BIAS-Command "[D\_variable X] = SQRT ([D\_variable Y])

[D\_variable X] = SQRT([D\_variable Y])

### Explanation:

The command "[D\_var. X] = SQRT([D\_var.Y])" calculates the square root of the double-variable Y and stores the result in the double-variable X.



Drive type:	Parameters:	Permissible inputs:	
All:	X:	0...255	<input type="checkbox"/> BIAS program <input type="checkbox"/> PLC program <input checked="" type="checkbox"/> Mathematics
	Y:	0...255	

### Example:

```

0 [D_variable 0 ] = 0
1 [D_variable 1 ] = 0.250000000
2 If [D_variable 1 ] -< [D_variable 0 ] then jump Error
3 [D_variable 1 ] = SQRT([D_variable 1 ])
4 ...
Error:
50 ...
  
```

### Result:

If the double-variable 1 is not negativ then the square root is calculated and the result is stored in the double variable 1.  
If the double variable is negative the BIAS execution pointer is set to the line "Error".

### Note:

- After switching on all variables have the content 0.
- The double variable contents will be displayed cyclically in the "diagnosis", "Mathematics" page.

### Command coding:

Code	D1	D2	D3	D4	D5	D6	D7
B7	X	Y	00	00	00	00	00

**Parameters:** D1 = Variable number: 0...255  
D2 = Variable number: 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