

MotionCtrlSW-SG5-SG7

Manual

Motion Control SW

- **C1100 Servo Drive Series**
- **I1100 Servo Drive Series**
- **C1250 Servo Drive Series**
- **E1200 Servo Drive Series**
- **C1400 Servo Drive Series**
- **E1400 Servo Drive Series**



© 2022 NTI AG

This work is protected by copyright.

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, microfilm, storing in an information retrieval system, not even for didactical use, or translating, in whole or in part, without the prior written consent of NTI AG.

LinMot® is a registered trademark of NTI AG.

Note

The information in this documentation reflects the stage of development at the time of press and is therefore without obligation. NTI AG reserves itself the right to make changes at any time and without notice to reflect further technical advance or product improvement.

NTI AG
LinMot
Bodenaeckerstrasse 2
CH-8957 Spreitenbach

Tel.: +41 56 419 91 91
Fax.: +41 56 419 91 92
Email: office@LinMot.com
Homepage: www.LinMot.com

Table of Contents

1 System Overview	16
1.1 References	16
1.2 Definitions, Items, Shortcuts	16
1.3 Data types	16
2 Motion Control Interfaces	17
3 State Machine	18
3.1 State 0: Not Ready To Switch On	19
3.2 State 1: Switch On Disabled	19
3.3 State 2: Ready To Switch On	20
3.4 State 3: Setup Error State	20
3.5 State 4: Error State	20
3.6 State 5: HW Test	20
3.7 State 6: Ready to Operate	20
3.8 State 7: Brake release Delay	20
3.9 State 8: Operation Enabled	20
3.10 State 9: Homing	21
3.11 State 10: Clearance Check	21
3.12 State 11: Going To Initial Position	21
3.13 State 12: Aborting	21
3.14 State 13: Freezing	21
3.15 State 14: Error Behaviour Quick Stop	21
3.16 State 15: Going To Position	21
3.17 State 16: Jogging +	21
3.18 State 17: Jogging -	21
3.19 State 18: Linearizing	21
3.20 State 19: Phase Searching	22
3.21 State 20: Special Mode	22
3.22 State 21: Brake set Delay	22
3.23 Building the Control Word	22
3.24 Control Word	22
3.25 Status Word	23
3.26 Warn Word	24
4 Motion Command Interface	26
4.1 Motion Command Interface	26
4.1.1 Command Header	26
4.1.1.1 Master ID	26
4.1.1.2 Sub ID	26
4.1.1.3 Command Count	26
4.2 Overview Motion Commands	26
4.3 Detailed Motion Command Description	35
4.3.1 No Operation (000xh)	35
4.3.2 Write Interface Control Word (001xh)	35
4.3.3 Write Live Parameter (002xh)	35

4.3.4 Write X4 Intf Outputs with Mask (003xh)	36
4.3.5 Select Position Controller Set (005xh)	36
4.3.6 Clear Event Evaluation (008xh)	36
4.3.7 Master Homing (009xh)	36
4.3.8 Reset I Position Controller (00Axh)	37
4.3.9 Reset (00Fh)	37
4.3.10 VAI Go To Pos (010xh)	37
4.3.11 VAI Increment Dem Pos (011xh)	37
4.3.12 VAI Increment Target Pos (012xh)	38
4.3.13 VAI Go To Pos From Act Pos And Act Vel (013xh)	38
4.3.14 VAI Go To Pos From Act Pos Starting With Dem Vel = 0 (014xh)	38
4.3.15 VAI Increment Act Pos (015xh)	38
4.3.16 VAI Increment Act Pos Starting With Dem Vel = 0 (016xh)	39
4.3.17 VAI Stop (017xh)	39
4.3.18 VAI Go To Pos After Actual Command (018xh)	39
4.3.19 VAI Go To Analog Pos (019xh)	40
4.3.20 VAI Go To Pos On Rising Trigger Event (01Axh)	40
4.3.21 VAI Increment Target Pos On Rising Trigger Event (01Bxh)	40
4.3.22 VAI Go To Pos On Falling Trigger Event (01Cxh)	41
4.3.23 VAI Increment Target Pos On Falling Trigger Event (01Dxh)	41
4.3.24 VAI Change Motion Parameters On Positive Position Transition (01Exh)	41
4.3.25 VAI Change Motion Parameters On Negative Position Transition (01Fh)	42
4.3.26 Predef VAI Go To Pos (020xh)	42
4.3.27 Predef VAI Increment Dem Pos (021xh)	42
4.3.28 Predef VAI Increment Target Pos (022xh)	42
4.3.29 Predef VAI Go To Pos From Act Pos and Act Vel (023xh)	43
4.3.30 Predef VAI Go To Pos From Act Pos Starting With Dem Vel = 0 (024xh)	43

4.3.31	Predef VAI Stop (027xh)	43
4.3.32	Predef VAI Go To Pos After Actual Command (028xh)	43
4.3.33	Predef VAI Go To Pos On Rising Trigger Event (02Axh)	43
4.3.34	Predef VAI Increment Target Pos On Rising Trigger Event (02Bxh)	44
4.3.35	Predef VAI Go To Pos On Falling Trigger Event (02Cxh)	44
4.3.36	Predef VAI Go To Pos On Falling Trigger Event (02Dxh)	44
4.3.37	Predef VAI Infinite Motion Positive Direction (02Exh)	44
4.3.38	Predef VAI Infinite Motion Negative Direction (02Fxh)	44
4.3.39	P Stream With Slave Generated Time Stamp (030xh)	45
4.3.40	PV Stream With Slave Generated Time Stamp (031xh)	45
4.3.41	P Stream With Slave Generated Time Stamp and Configured Period Time (032xh)	45
4.3.42	PV Stream With Slave Generated Time Stamp and Configured Period Time (033xh)	45
4.3.43	PVA Stream With Slave Generated Time Stamp (034xh)	46
4.3.44	PVA Stream With Slave Generated Time Stamp and Configured Period Time (035xh)	46
4.3.45	PVA Stream With Master Generated Time Stamp (03Axh) EtherNet/IP CIP Sync	46
4.3.46	Stop Streaming (03Fxh)	47
4.3.47	Time Curve With Default Parameters (040xh)	47
4.3.48	Time Curve With Default Parameters From Act Pos (041xh)	47
4.3.49	Time Curve To Pos With Default Speed (042xh)	47
4.3.50	Time Curve To Pos With Adjustable Time (043xh)	48
4.3.51	Time Curve With Adjustable Offset, Time Scale & Amplitude Scale (044xh)	48
4.3.52	Time Curve With Adjustable Offset, Time & Amplitude Scale (045xh)	48
4.3.53	Time Curve With Adjustable Offset, Time & Amplitude Scale On Rising Trigger Event (046xh)	48
4.3.54	Time Curve With Adjustable Offset, Time & Amplitude Scale On Falling Trigger Event (047xh)	49
4.3.55	Time Curve To Pos With Default Speed On Rising Trigger Event (04Axh)	49
4.3.56	Time Curve To Pos With Default Speed On Falling Trigger Event (04Cxh)	49
4.3.57	Time Curve To Pos With Adjustable Time On Rising Trigger Event (04Exh)	49

4.3.58 Time Curve To Pos With Adjustable Time On Falling Trigger Event (04F _{xh})	50
4.3.59 Modify Curve Start Address in RAM (050 _{xh})	50
4.3.60 Modify Curve Info Block 16 Bit Value in RAM (051 _{xh})	50
4.3.61 Modify Curve Info Block 32 Bit Value in RAM (052 _{xh})	51
4.3.62 Modify Curve Data Block 32 Bit Value in RAM (054 _{xh})	51
4.3.63 Modify Curve Data Block 64 Bit Value in RAM (055 _{xh})	51
4.3.64 Modify Curve Data Block 96 Bit Value in RAM (056 _{xh})	52
4.3.65 Reset and disable CAM motion (060 _{xh})	52
4.3.66 Enable CAM Motion on next Z (061 _{xh})	52
4.3.67 Start Encoder Cam From Actual Count and Pos (063 _{xh})	52
4.3.68 Setup Encoder Cam On Rising Trigger Event With Delay Counts (069 _{xh})	53
4.3.69 Setup Encoder Cam On Rising Trigger Event With Delay Counts, Target Pos and Length (06A _{xh})	53
4.3.70 Setup Encoder Cam On Falling Trigger Event With Delay Counts (06B _{xh})	53
4.3.71 Setup Encoder Cam On Falling Trigger Event With Delay Counts, Target Pos and Length (06C _{xh})	53
4.3.72 Setup Encoder Cam On Rising Trigger Event With Delay Counts, Amplitude scale and Length (06D _{xh})	54
4.3.73 Setup Encoder Cam On Falling Trigger Event With Delay Counts, Amplitude scale and Length (06E _{xh})	54
4.3.74 Start VAI Encoder Position Indexing (070 _{xh})	54
4.3.75 Start Predef VAI Encoder Position Indexing (071 _{xh})	55
4.3.76 Stop Position Indexing and VAI Go To Pos (07E _{xh})	55
4.3.77 Stop Position Indexing and VAI Go To Pos (07F _{xh})	55
4.3.78 VAI 16 Bit Go To Pos (090 _{xh})	55
4.3.79 VAI 16 Bit Increment Dem Pos (091 _{xh})	56
4.3.80 VAI 16 Bit Increment Target Pos (092 _{xh})	56
4.3.81 VAI 16 Bit Go To Pos From Act Pos And Act Vel (093 _{xh})	56
4.3.82 VAI 16 Bit Go To Pos From Act Pos Starting With Dem Vel = 0 (094 _{xh})	57
4.3.83 VAI 16 Bit Increment Act Pos (095 _{xh})	57
4.3.84 VAI 16 Bit Increment Act Pos Starting With Dem Vel = 0 (096 _{xh})	57

4.3.85 VAI 16 Bit Stop (097xh)	58
4.3.86 VAI 16 Bit Go To Pos After Actual Command (098xh)	58
4.3.87 VAI 16 Bit Go To Pos On Rising Trigger Event (09Axh)	58
4.3.88 VAI 16 Bit Increment Target Pos On Rising Trigger Event (09Bxh)	58
4.3.89 VAI 16 Bit Go To Pos On Falling Trigger Event (09Cxh)	59
4.3.90 VAI 16 Bit Increment Target Pos On Falling Trigger Event (09Dxh)	59
4.3.91 VAI 16 Bit Change Motion Parameters On Positive Position Transition (09Exh)	59
4.3.92 VAI 16 Bit Change Motion Parameters On Negative Position Transition (09Fxh)	60
4.3.93 Predef VAI 16 Bit Go To Pos (0A0xh)	60
4.3.94 Predef VAI 16 Bit Increment Dem Pos (0A1xh)	60
4.3.95 Predef VAI 16 Bit Increment Target Pos (0A2xh)	60
4.3.96 Predef VAI 16 Bit Go To Pos From Act Pos And Act Vel (0A3xh)	61
4.3.97 Predef VAI 16 Bit Go To Pos From Act Pos Starting With Dem Vel = 0 (0A4xh)	61
4.3.98 Predef VAI 16 Bit Stop (0A7xh)	61
4.3.99 Predef VAI 16 Bit Go To Pos After Actual Command (0A8xh)	61
4.3.100 Predef VAI 16 Bit Go To Pos On Rising Trigger Event (0AAxh)	62
4.3.101 Predef VAI 16 Bit Increment Target Pos On Rising Trigger Event (0ABxh)	62
4.3.102 Predef VAI 16 Bit Go To Pos On Falling Trigger Event (0ACxh)	62
4.3.103 Predef VAI 16 Bit Increment Target Pos On Falling Trigger Event (0ADxh)	62
4.3.104 VAI Predef Acc Go To Pos (0B0xh)	63
4.3.105 VAI Predef Acc Increment Dem Pos (0B1xh)	63
4.3.106 VAI Predef Acc Increment Target Pos (0B2xh)	63
4.3.107 VAI Predef Acc Go To Pos From Act Pos And Act Vel (0B3xh)	63
4.3.108 VAI Predef Acc Go To Pos From Act Pos Starting With Dem Vel = 0 (0B4xh)	64
4.3.109 VAI Predef Acc Go To Pos After Actual Command (0B8xh)	64
4.3.110 VAI Predef Acc Go To Pos On Rising Trigger Event (0BAxh)	64
4.3.111 VAI Predef Acc Increment Target Pos On Rising Trigger Event (0BBxh)	64

4.3.112 VAI Predef Acc Go To Pos On Falling Trigger Event (0BCxh)	65
4.3.113 VAI Predef Acc Increment Target Pos On Falling Trigger Event (0BDxh)	65
4.3.114 VAI Dec=Acc Go To Pos (0C0xh)	65
4.3.115 VAI Dec=Acc Increment Dem Pos (0C1xh)	65
4.3.116 VAI Dec=Acc Increment Target Pos (0C2xh)	66
4.3.117 VAI Dec=Acc Go To Pos From Act Pos And Act Vel (0C3xh)	66
4.3.118 VAI Dec=Acc Go To Pos From Act Pos Starting With Dem Vel = 0 (0C4xh)	66
4.3.119 VAI Dec=Acc Go To Pos With Max Curr (0C5xh)	66
4.3.120 VAI Dec=Acc Go To Pos From Act Pos And Vel With Max Curr (0C6xh)	67
4.3.121 VAI Dec=Acc Go To Pos From Act Pos And Vel = 0 With Max Curr (0C7xh)	67
4.3.122 VAI Dec=Acc Go To Pos After Actual Command (0C8xh)	67
4.3.123 VAI Dec=Acc Go To Pos On Rising Trigger Event (0CAxh)	68
4.3.124 VAI Dec=Acc Increment Target Pos On Rising Trigger Event (0CBxh)	68
4.3.125 VAI Dec=Acc Go To Pos On Falling Trigger Event (0CCxh)	68
4.3.126 VAI Dec=Acc Increment Target Pos On Falling Trigger Event (0CDxh)	68
4.3.127 VAI Dec=AccVAI Infinite Motion Positive Direction (0CExh)	69
4.3.128 VAI Dec=Acc VAI Infinite Motion Negative Direction (0CFxh)	69
4.3.129 VAI Increment Captured Pos (0D0xh)	69
4.3.130 VAI 16 Bit Dec=Acc Go To Pos (0D1xh)	69
4.3.131 Predef VAI Increment Act Pos by Cmd Tab Var1 (0D2xh)	70
4.3.132 VAI Go To Cmd Tab Var1 Pos (0D4xh)	70
4.3.133 VAI Go To Cmd Tab Var2 Pos (0D5xh)	70
4.3.134 VAI Go To Cmd Tab Var1 Pos From Act Pos And Act Vel (0D6xh)	71
4.3.135 VAI Go To Cmd Tab Var2 Pos From Act Pos And Act Vel (0D7xh)	71
4.3.136 VAI Stop at Angle (0D8xh)	71
4.3.137 VAI Increment Act Pos Starting With Dem Vel = 0 Reset I-Contr (0D9xh)	71
4.3.138 VAI Start Trig Rise Config VAI Command (0DCxh)	72

4.3.139	VAI Start Trig Fall Config VAI Command (0DDxh)	72
4.3.140	VAI Start Trig Rise Config VAI Command (0DExh)	72
4.3.141	VAI Start Trig Fall Config VAI Command (0DFxh)	72
4.3.142	Sin VA Go To Pos (0E0xh)	73
4.3.143	Sin VA Increment Demand Pos (0E1xh)	73
4.3.144	Sin VA Go To Pos From Actual Pos (0E4xh)	73
4.3.145	Sin VA Increment Actual Pos (0E6xh)	73
4.3.146	Sin VA Go To Pos After Actual Command (0E8xh)	74
4.3.147	Sin VA Go To Analog Pos (0E9xh)	74
4.3.148	Sin VA Go To Pos On Rising Trigger Event (0EAxh)	74
4.3.149	Sin VA Increment Demand Pos On Rising Trigger Event (0EBxh)	75
4.3.150	Sin VA Go To Pos On Falling Trigger Event (0ECxh)	75
4.3.151	Sin VA Increment Demand Pos On Falling Trigger Event (0EDxh)	75
4.3.152	Bestehorn VAJ Go To Pos (0F0xh)	76
4.3.153	Bestehorn VAJ Increment Demand Pos (0F1xh)	76
4.3.154	Bestehorn VAJ 16 BitGo To Pos (0F2xh)	76
4.3.155	Bestehorn VAJ Go To Pos From Actual Pos (0F4xh)	76
4.3.156	Bestehorn VAJ Increment Actual Pos (0F6xh)	77
4.3.157	Bestehorn VAJ Go To Pos After Actual Command (0F8xh)	77
4.3.158	Bestehorn VAJ Go To Analog Pos (0F9xh)	77
4.3.159	Bestehorn VAJ Go To Pos On Rising Trigger Event (0FAxh)	78
4.3.160	Bestehorn VAJ Increment Demand Pos On Rising Trigger Event (0FBxh)	78
4.3.161	Bestehorn VAJ Go To Pos On Falling Trigger Event (0FCxh)	79
4.3.162	Bestehorn VAJ Increment Demand Pos On Falling Trigger Event (0FDxh)	79
4.3.163	Encoder Cam Enable (100xh)	79
4.3.164	Encoder Cam Disable (101xh)	80
4.3.165	Encoder Cam Go To Sync Pos (102xh)	80

4.3.166 Encoder Cam Set Value (104xh)	80
4.3.167 Encoder Cam y Define Curve With Default Parameters (1y0xh)	80
4.3.168 Encoder Cam y Define Curve From Act Pos (1y1xh)	80
4.3.169 Encoder Cam y Define Curve To Pos (1y2xh)	81
4.3.170 Encoder Cam y Define Curve From Pos To Pos In Counts (1y3xh)	81
4.3.171 Encoder Cam y Define Curve To Pos In Counts (1y4xh)	81
4.3.172 Encoder Cam y Define Curve With Amplitude Scale In Counts (1y5xh)	81
4.3.173 Encoder Cam y Enable (1y6xh)	82
4.3.174 Encoder Cam y Disable (1y7xh)	82
4.3.175 Encoder Cam y Change Amplitude Scale and Length (1y8xh)	82
4.3.176 Start Command Table Command (200xh)	82
4.3.177 Start Command Table Command On Rising Trigger Event (201xh)	82
4.3.178 Start Command Table Command On Falling Trigger Event (202xh)	83
4.3.179 Modify Command Table 16 bit Parameter in RAM (208xh)	83
4.3.180 Modify Command Table 32 bit Parameter in RAM (209xh)	83
4.3.181 Wait Time (210xh)	83
4.3.182 Wait Until Motion Finished (211xh)	84
4.3.183 Wait Until In Target Position (212xh)	84
4.3.184 Wait Until Rising Trigger Event (213xh)	84
4.3.185 Wait Until Falling Trigger Event (214xh)	84
4.3.186 Wait Until Demand Position Greater Than (220xh)	84
4.3.187 Wait Until Demand Position Less Than (221xh)	84
4.3.188 Wait Until Actual Position Greater Than (222xh)	85
4.3.189 Wait Until Actual Position Less Than (223xh)	85
4.3.190 Wait Until Difference Position Greater Than (224xh)	85
4.3.191 Wait Until Difference Position Less Than (225xh)	85
4.3.192 Wait Until Difference Position Unsigned Greater Than (226xh)	85

4.3.193 Wait Until Difference Position Unsigned Less Than (227xh)	86
4.3.194 Wait Until Demand Velocity Greater Than (228xh)	86
4.3.195 Wait Until Demand Velocity Less Than (229xh)	86
4.3.196 Wait Until Actual Velocity Greater Than (22Axh)	86
4.3.197 Wait Until Actual Velocity Less Than (22Bxh)	86
4.3.198 Wait Until Current Greater Than (22Exh)	87
4.3.199 Wait Until Current Less Than (22Fh)	87
4.3.200 Set Cmd Table Var 1 To (240xh)	87
4.3.201 Add To Cmd Table Var 1 (241xh)	87
4.3.202 Set Cmd Table Var 2 To (242xh)	87
4.3.203 Add To Cmd Table Var 2 (243xh)	88
4.3.204 Write Cmd Table Var 1 To UPID RAM value (248xh)	88
4.3.205 Write Cmd Table Var 2 To UPID RAM value (249xh)	88
4.3.206 Write UPID RAM Value To Cmd Table Var 1 (24Cxh)	88
4.3.207 Write UPID RAM Value To Cmd Table Var 2 (24Dxh)	88
4.3.208 Write UPID RAM value To UPID ROM value (24Exh)	89
4.3.209 IF Cmd Table Var 1 Less Than (250xh)	89
4.3.210 IF Cmd Table Var 1 Greater Than (251xh)	89
4.3.211 IF Cmd Table Var 2 Less Than (252xh)	89
4.3.212 IF Cmd Table Var 2 Greater Than (253xh)	90
4.3.213 IF Cmd Table Var 1 Less Than UPID Value (256xh)	90
4.3.214 IF Cmd Table Var 2 Less Than UPID Value (257xh)	90
4.3.215 IF Demand Position Less Than (258xh)	90
4.3.216 IF Demand Position Greater Than (259xh)	91
4.3.217 IF Actual Position Less Than (25Axh)	91
4.3.218 IF Actual Position Greater Than (25Bxh)	91
4.3.219 IF Difference Position Less Than (25Cxh)	91

4.3.220 IF Difference Position Greater Than (25Dxh)	92
4.3.221 IF Current Less Than (25Exh)	92
4.3.222 IF Current Greater Than (25Fxh)	92
4.3.223 IF Analog Val On X4.4 Less Than (260xh)	92
4.3.224 IF Masked X4 Input Value Equal Than (262xh)	93
4.3.225 IF Masked X6 Input Value Equal Than (263xh)	93
4.3.226 IF Masked Status Word Equal Than (264xh)	93
4.3.227 IF Masked Warn Word Equal Than (265xh)	94
4.3.228 IF CAM Counts Less Than (266xh)	94
4.3.229 Encoder Winding Start With Default Parameters (300xh)	94
4.3.230 Encoder Winding Start With Default Parameters At Revolutions (301xh)	94
4.3.231 Encoder Winding Stop Adaptation Of Left/Right Position and Disturbance (304xh)	94
4.3.232 Encoder Winding Restart Adaptation Of Left/Right Position and Disturbance (305xh)	95
4.3.233 Encoder Curve Winding Start With Default Parameters (310xh)	95
4.3.234 Encoder Curve Winding Start With Default Parameters At Revolutions (311xh)	95
4.3.235 VAI Go To Pos With Higher Force Ctrl Limit (380xh)	95
4.3.236 VAI Go To Pos From Act Pos And Reset Force Control (381xh)	96
4.3.237 Force Ctrl Change Target Force (382xh)	96
4.3.238 VAI Go To Pos With Higher Force Ctrl Limit and Target Force (383xh)	96
4.3.239 VAI Go To Pos With Lower Force Ctrl Limit (384xh)	96
4.3.240 VAI Go To Pos With Lower Force Ctrl Limit and Target Force (385xh)	97
4.3.241 VAI Go To Pos From Act Pos And Reset Force Control Set I (386xh)	97
4.3.242 VAI Increment Act Pos And Reset Force Control Set I (387xh)	97
4.3.243 VAI Inc Act Pos With Higher Force Ctrl Limit and Target Force (388xh)	98
4.3.244 VAI Inc Act Pos With Lower Force Ctrl Limit and Target Force (389xh)	98
4.3.245 VAI Go To Pos Tara Diff Analog In -10V..10V (38Axh)	98
4.3.246 Standstill Tara Diff Analog In -10V..10V (38Bxh)	100

4.3.247	Current Command Mode (390xh)	100
4.3.248	Closed Loop Force Command Mode Rel (398xh)	100
4.3.249	Closed Loop Force Command Mode Abs (399xh)	100
4.3.250	Change to Position Controlled Mode (39Fxh)	101
4.3.251	VAJI Go To Pos (3A0xh)	101
4.3.252	VAJI Increment Dem Pos (3A1xh)	101
4.3.253	VAJI Increment Target Pos (3A2xh)	102
4.3.254	VAJI Go To Pos From Act Pos Starting With Dem Vel = 0 and Dem Acc = 0 (3A4xh)	102
4.3.255	VAJI Increment Act Pos Starting With Dem Vel = 0 and Dem Acc = 0 (3A6xh)	103
4.3.256	VAJI Go To Pos On Rising Trigger Event (3AAxh)	103
4.3.257	VAJI Increment Target Pos On Rising Trigger Event (3ABxh)	104
4.3.258	VAJI Go To Pos On Falling Trigger Event (3ACxh)	104
4.3.259	VAJI Increment Target Pos On Falling Trigger Event (3ADxh)	104
4.3.260	Predef VAJI Go To Pos (3B0xh)	105
4.3.261	Predef VAJI Increment Dem Pos (3B1xh)	105
4.3.262	Predef VAJI Increment Target Pos (3B2xh)	105
4.3.263	Predef VAJI Go To Pos From Act Pos Starting With Dem Vel = 0 and Dem Acc = 0 (3B4xh)	105
4.3.264	Predef VAJI Go To Pos On Rising Trigger Event (3BAxh)	106
4.3.265	Predef VAJI Increment Target Pos On Rising Trigger Event (3BBxh)	106
4.3.266	Predef VAJI Go To Pos On Falling Trigger Event (3BCxh)	106
4.3.267	Predef VAJI Increment Target Pos On Falling Trigger Event (3BDxh)	106
4.3.268	VAI Modulo Go To Pos Positive Direction (3C0xh)	107
4.3.269	VAI Modulo Go To Pos Negative Direction (3C1xh)	107
4.3.270	VAI Modulo Go To Pos Shortest Way (3C2xh)	107
5	Set-point Generation	108
5.1	<i>VA-Interpolator</i>	108
5.1.1	Input Variables	109
5.1.2	Output Variables	109

5.2	<i>VAJ-Interpolator</i>	109
5.2.1	Input Variables	110
5.2.2	Output Variables	110
5.3	<i>Sine VA Motion</i>	111
5.3.1	Input Variables	111
5.3.2	Output Variables	112
5.4	<i>Bestehorn VAJ Motion</i>	112
5.4.1	Input Variables	113
5.4.2	Output Variables	113
5.5	<i>P/PV/PVA-Stream</i>	113
5.6	<i>Cam Motions</i>	114
5.6.1	Triggered Cam Motions	114
5.6.2	Repeated Cam Motions with the Modulo CamMode	114
6	Command Table	115
7	Drive Configuration	116
7.1	<i>Power Bridge</i>	116
7.2	<i>X4 I/O Definitions</i>	116
7.2.1	Brake Output	116
7.2.1.1	Brake Operation with Abort Behavior	117
7.2.1.2	Brake Operation with Quick Stop Behavior	117
7.2.1.3	Brake Operation with Switch On Behavior	118
7.2.2	Trigger Input	118
7.2.2.1	Direct Trigger Mode	118
7.2.2.2	Inhibited Trigger Mode	118
7.2.2.3	Delayed Trigger Mode	119
7.2.2.4	Inhibited & Delayed Trigger Mode	119
7.2.3	X4.8 and X4.9 Limit Switches	120
7.2.4	X4.10 and X4.11 PTC 1 and PTC 2	120
7.3	<i>Master Encoder</i>	120
7.4	<i>Monitoring</i>	121
7.4.1	Logic Supply Voltage	121
7.4.2	Motor Supply Voltage	121
7.4.2.1	Phase Switch On Test	122
7.4.3	Regeneration Resistor	122
7.4.4	Temperature Monitoring	123
7.4.5	Cycle Time Monitoring	124

7.5	<i>PosCtrlStructure</i>	125
8	Motor Configuration	125
8.1	<i>Generic Motor Temperature Calculated</i>	125
8.2	<i>External Position Sensor</i>	126
8.2.1	Serial Synchronous Interface (SSI)	126
8.2.2	Bidirectional Serial Synchronous Interface (BiSS-C)	127
8.2.3	Bidirectional Serial Synchronous Interface (BiSS-B)	128
8.2.4	EnDat2	129
8.2.5	Incremental AB Encoder	130
8.2.6	Incremental ABZ Encoder	131
8.2.7	Commutation	131
8.3	<i>Modulo Functionality</i>	132
9	State Machine Setup	136
9.1	<i>Linearizing</i>	136
9.1.1	Mode	136
9.1.2	System Compensation Selection	136
9.1.3	Linearizing Config	137
10	Error Code List	137

1 System Overview

This user Manual describes the Motion Control SW functionality of the LinMot servo drives.

1.1 References

Ref	Title	Source
1	Installation_Guide_C1100.pdf	www.linmot.com
1	Installation_Guide_C1250.pdf	www.linmot.com
1	Installation_Guide_E1200.pdf	www.linmot.com
2	Installation_Guide_E1400.pdf	www.linmot.com
3	Usermanual_LinMot-Talk.pdf	www.linmot.com

The documentation is distributed with the LinMot-Talk configuration software or can be downloaded from the Internet from the download section of our homepage.

1.2 Definitions, Items, Shortcuts

Shortcut	Meaning
LM	LinMot linear motor
OS	Operating system (Software)
MC (SW)	Motion Control (Software)
Intf	Interface (Software)
Appl	Application (Software)
VAI	VA-Interpolator (Max velocity limited acceleration position interpolator)
Pos	Position
Vel	Velocity
Acc	Acceleration
Dec	Deceleration
UPID	Unique Parameter ID (16 bit)

1.3 Data types

Type	Range/Format	Num of bytes
Bool	Boolean, False/True	1/8
Byte	0..255	1
Char	ASCII	1
String	Array of char last char = 00h	X
SInt16	-32'768..32'767	2
UInt16	0..65'535	2
SInt32	-2'147'483'648..2'147'483'647	4
UInt32	0..4'294'967'295	4
Float		4

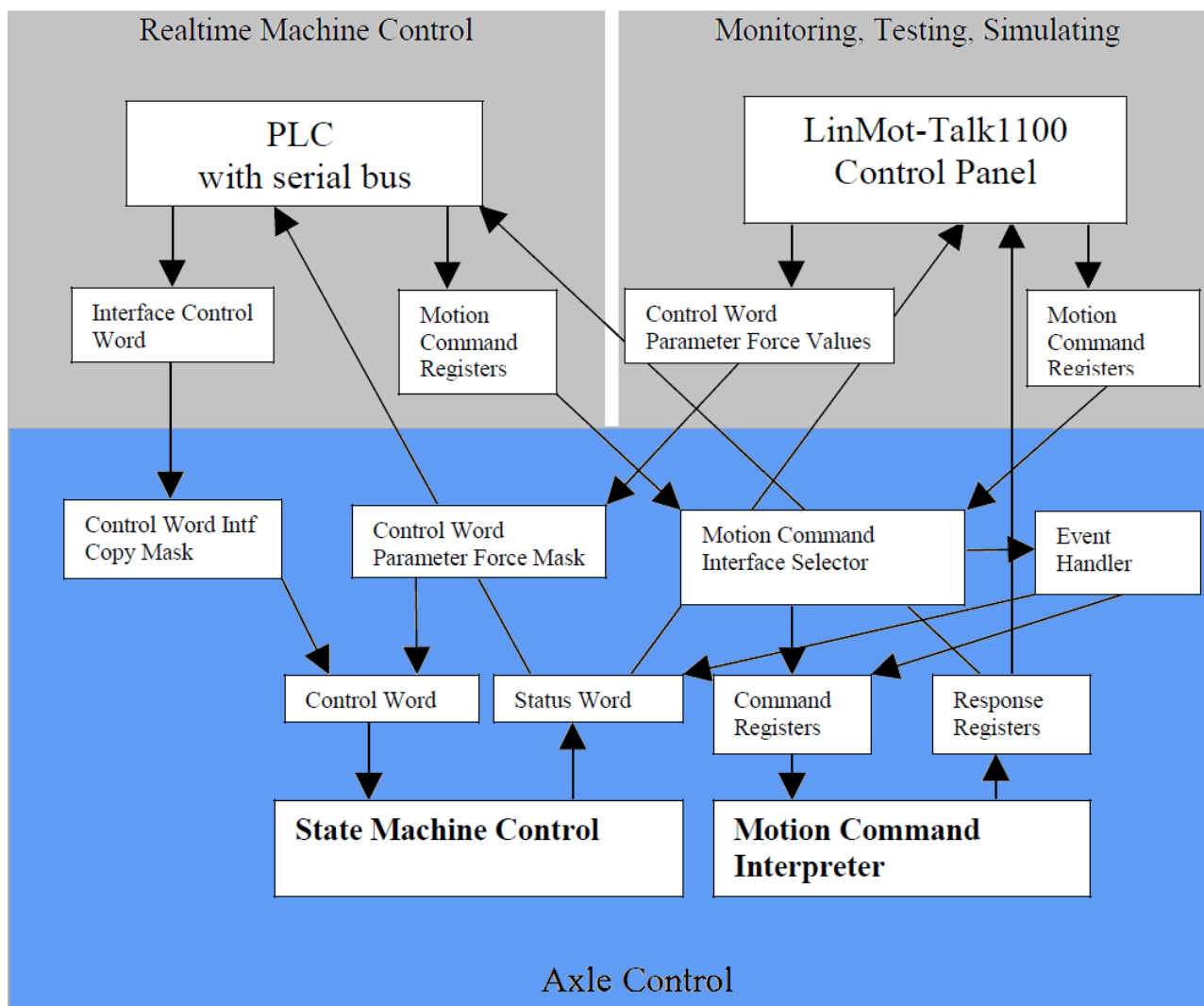
2 Motion Control Interfaces

For controlling the behavior of the motion control SW, two different Interfaces are available. For controlling the main state machine, a bit coded **control word** can be used. For controlling the motion functionality a memory mapped **motion command interface** can be used. These two instances are mapped via an interface SW to an upper control system (PLC, IPC, PC, ...). The interfacing is done with digital I/Os or a serial link like Profibus DP, CAN bus (CANopen), RS485, RS422 or RS232 (LinRS protocol), or one of the following Ethernet protocols:

- PROFINET
- Ethernet/IP
- EtherCAT
- SERCOS III
- POWERLINK
- LinUDP

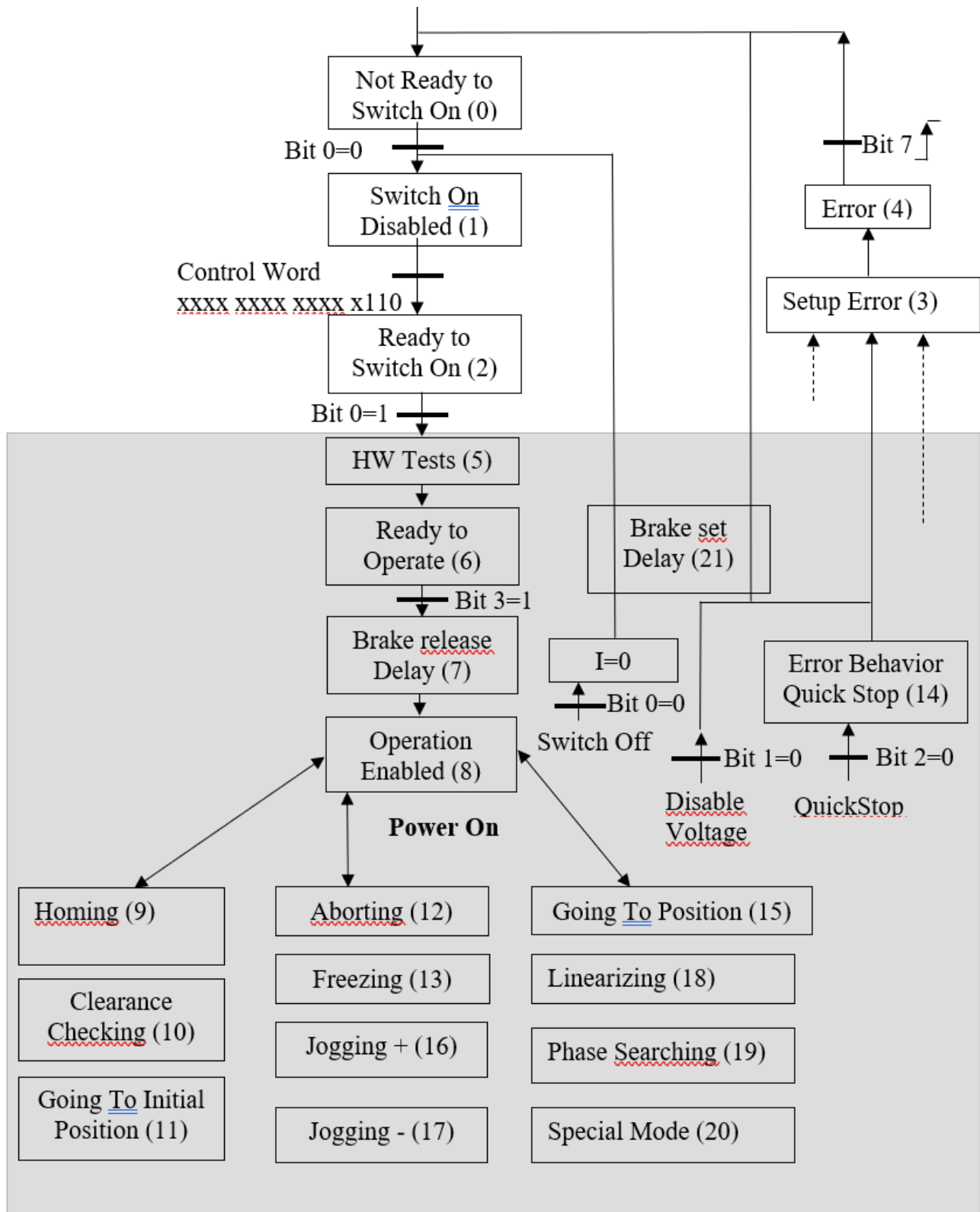
With LinMot-Talk the control over the control word can be taken bit by bit, for testing and debugging. Unused control word bits can be forced by parameter value.

Also the control of the motion command interpreter can be switched to the control panel of the LinMot-Talk software for testing. All this can be done while the system is running, so be careful using this features on a running machine!



3 State Machine

The main behavior of the axes is controlled with the control word, it is shown in the following state diagram.



The state machine can be followed in the PLCs with fieldbus using the StateVar. This response word can be configured for any supported fieldbus.

State Var															
Main State								Sub State							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The State Var is divided into two sections: the Main State section (high byte) contains directly the number of the state machine, the content of the Sub State (low byte) is state depending.

State Var	
Main State	Sub State
00: Not Ready To Switch On	0
01: Switch On Disabled	0
02: Ready To Switch On	0
03: Setup Error	Error Code which will be logged
04: Error	Logged Error Code
05: HW Tests	0 (Not yet defined)
06: Ready To Operate	0 (Not yet defined)
07: -	
08: Operation Enabled	Bits 0..3: Motion Command Count Bit 4: Event Handler Active Bit 5: Motion Active Bit 6: In Target Position Bit 7: Homed
09: Homing	0Fh: Homing Finished
10: Clearance Check	0Fh: Clearance Check Finished
11: Going To Initial Position	0Fh: Going To Initial Position Finished
12: Aborting	Not yet defined
13: Freezing	Not yet defined
14: Quick Stop (Error Behaviour)	Not yet defined
15: Going To Position	0Fh: Going To Position Finished
16: Jogging +	01h: Moving positive 0Fh: Jogging +Finished
17: Jogging -	01h: Moving negative 0Fh: Jogging -Finished
18: Linearizing	Not yet defined
19: Phase Search	Not yet defined
20: Special Mode	Not yet defined
21: Brake Delay	Not yet defined

3.1 State 0: Not Ready To Switch On

In this state the release of control word bit 0 switch on is awaited. As soon as this bit is cleared a change to state 1 is performed. This behavior avoids self starting if all necessary bits for a start are set correctly in the control word.

3.2 State 1: Switch On Disabled

The state machine rests in this state as long as the bits 1 or 2 of the control word are cleared.

3.3 State 2: Ready To Switch On

The state machine rests in this state as long as the bit 0 is cleared.

3.4 State 3: Setup Error State

The state machine rests in this state as long the bits 0 is cleared.

3.5 State 4: Error State

The error state can be acknowledged with a rising edge of the control word bit 7 'Error Acknowledge'. If the error is fatal, bit 12 'Fatal Error' in the status word is set, no error acknowledgment is possible.

In the case of a fatal error, the error has to be checked, and the problem has to be solved before a reset or power cycle is done for resetting the error.

3.6 State 5: HW Test

The HW Test state is an intermediate state before turning on the power stage of the servo drive. If everything seems to be ok the servo changes to state 6 without any user action. The test takes about 300ms.

3.7 State 6: Ready to Operate

In this state the motor is either position controlled or with demand current = 0 and under voltage, but no motion commands are accepted. The mode is configurable with UPID 6300h.

Sending motion commands in this state will generate the error 'Motion command sent in wrong state' and a state change to the error state will be performed.

Clearing the control word bit 3 'Enable Operation' in state 8 or higher will stop immediately the set point generation and a state transition to 6 is performed. Clearing the bit while a motion is in execution a following error might be generated.

3.8 State 7: Brake release Delay

If the brake functionality is defined, the drive enter this state and waits until release delay time is over before entering state 8, this to avoid motions with half released brake.

3.9 State 8: Operation Enabled

This is the state of the normal operation in which the motion commands are executed. It is strongly recommended to use the State Var for the motion command synchronization with any fieldbus system.

State Var															
Main State = 8								Homed	In Target Position	Motion Active	Event Handler	Motion Command Count			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

In the high byte stands the number of the main state = 8. In the low byte stands in the lowest 4 bits the actual interpreted 'Motion Command Count', bit 4 indicates if the event handler is active, in bit 5 stands the status word bit 'Motion Active', in bit 6 the status word bit 'In Target Position' and in bit 7 the status word bit 'Homed'. Because the 'Motion Command Count' echo and this status word bits are located in the same byte no data consistency problem is possible with any fieldbus.

A new motion command can be setup when the Motion Command Count has changed to the last sent and the 'Motion Active' bit is 0 or the 'In Target Position' bit is 1 if an exact positioning is required.

3.10 State 9: Homing

The homing state is used to define the position of the system according a mechanical reference, a home switch or an index.

For LinMot motors the slider home position at this home position is taken to compensate edge effects.

In the home sequence a position check of two positions and the motion to an initial position can be added.

Hint: If a mechanical stop homing mode is chosen, the initial position should be a little apart from this mechanical stop to avoid overheating of the motor.

3.11 State 10: Clearance Check

Setting the Clearance Check bit in the Control Word, two positions are moved to, to check if the whole motion range is free. Normally this action is added to the homing sequence to ensure that the homing was done correctly.

3.12 State 11: Going To Initial Position

Setting the Go To Initial Position bit in the control word, the servo moves to the initial position, normally used to move away from the mechanical stop after homing, to protect the motor from overheating at the mechanical stop. After an error it is also recommended to move to a defined position again.

3.13 State 12: Aborting

Clearing the /Abort bit in the control word initiates a quick stop. After the motion has stopped the servo rests position controlled. Setting the bit again the servo drive rests in position until a new motion command is executed.

3.14 State 13: Freezing

Clearing the /Freeze bit in the control word initiates a quick stop. After the motion is stopped the servo rests position controlled. Setting the bit again the servo drive will finish the frozen motion (e.g. if it was a VAI command). Curve motion can be frozen but not restarted by releasing this bit, setting the bit again the motor moves at the target position of the last VAI command, if never used a VAI command it will go to the initial position.

3.15 State 14: Error Behaviour Quick Stop

Most of the errors, which can occur during an active motion, cause a quick stop behavior to stop the motion. After the quick stop is finished the motor is no longer position controlled.

3.16 State 15: Going To Position

Setting the Go To Position bit in the control word, the servo moves to the defined position, recommendable for example, after an error, to move to a defined position again.

3.17 State 16: Jogging +

Setting the Jog Move + bit in the control word, the servo moves either a defined position increment or to the maximal position with a limited speed. Releasing the bit will stop the motion.

3.18 State 17: Jogging -

Setting the Jog Move - bit in the control word, the servo moves either a defined position decrement or to the minimal position with a limited speed. Releasing the bit will stop the motion

3.19 State 18: Linearizing

The linearizing state is used to correct position feedback parameters, to improve the linearity of the position feedback.

3.20 State 19: Phase Searching

The phase search is only defined for three phase EC motors with hall switches and ABZ-sensors to find the commutation offset for to the sensor. It cannot be guaranteed that this feature will work for all kinds of EC motors. The found offset can be found in the variable section Calculated Commutation Offset (UPID: 1C1Bh), and has to be set manually to the parameter Phase Angle (UPID 11F2h).

3.21 State 20: Special Mode

The Special Mode is available only on the B1100 servo drives. In this state the current command mode over the analog input is available. For using this mode see the [4].

3.22 State 21: Brake set Delay

If the brake functionality is defined, the drive rests the in the position controlled mode for the brake apply delay time to make sure that the brake is closed when switch off the power stage.

3.23 Building the Control Word

The Control Word can be accessed bit by bit from different sources with different priorities. The highest priorities have the bits that are forced by parameters. The second highest priority has the control panel of the LinMot-Talk software, if logged in with the SW. The next lower priorities have the bits that are defined on the X4 IOs as control word input bits. The lowest priority have bits which are set over the interface (normally a serial fieldbus connection), so in the Ctrl Word Interface Copy mask all bits can be selected, without causing any problems, but bits which should not be accessed through the interface can be masked out.

3.24 Control Word

With the Control Word (16Bit) the main state machine of the servo drive can be accessed. Following table shows the meaning of each bit:

Bit Name	Val	Meaning	Remark
0 Switch On	0	OFF1	A-Stop, -> Current = 0, power switches disabled
	1	ON	State change from switch on disabled to ready to switch on
1 Voltage Enable	0	OFF2	Power switches disabled without microcontroller action
	1	Operation	
2 /Quick Stop	0	OFF3	Quick Stop -> Current = 0 -> H-Bridges disabled
	1	Operation	
3 Enable Operation	0	Operation disabled	Position controller active Motion Commands disabled
	1	Operation enable	Position controller active Motion Commands enabled
4 /Abort	0	Abort	Quick Stop position control rests active, motion command is cleared.
	1	Operation	
5 /Freeze	0	Freeze motion	Quick Stop position control rests active, Target position not cleared, curves motions are aborted
	1	Operation	Rising edge will reactivate motion command
6 Go To Position	0		
	1	Go To Position	Go to fixed parameterized Position. Wait for release of signal.
7 Error Acknowledge	0		
	1	Error Acknowledge	Rising edge of signal acknowledges error

Bit Name	Val	Meaning	Remark
8 Jog Move +	0		
	1		Jog Move +
9 Jog Move -	0		
	1		Jog Move -
10 Special Mode	0		
	1	Special Mode	Special Mode
11 Home	0	Stop Homing	
	1	Homing	At startup bit 11 Status word is cleared, until procedure is finished.
12 Clearance Check	0	Stop Clearance Check	
	1	Clearance Check	Enable Clearance Check Movements
13 Go To Initial Position	0		
	1	Go To initial Position	Rising edge will start go to initial position
14 Reserved	0		
	1		Reserved
15 Phase Search	0	Stop Phase Search	
	1	Phase Search	Enable Phase Search Movements

3.25 Status Word

Following table shows the meaning of the single bits: ↗

Bit Name	Val	Meaning	Remark
0 Operation Enabled	0		State Nr < 8
	1	Operation Enabled	State Nr 8 or higher (copied to drive EN LED)
1 Switch On Active	0	Switch On Disabled	Control Word Bit 0
	1	Switch On Enabled	
2 Enable Operation	0	Operation Disabled	Control Word Bit 3
	1	Operation	
3 Error	0	No Error	
	1	Error	Acknowledge with Control word Bit 7 (Reset Error)
4 Voltage Enable	0	Power Bridge Off	Control Word Bit 1
	1	Operation	
5 /Quick Stop	0	Active	Control Word Bit 2
	1	Operation	
6 Switch On Locked	0	Not Locked	
	1	Switch On Locked	Release with 0 of Control word bit 0 (Switch On)
7 Warning	0	Warning not active	No bit is set in the Warn Word
	1	Warning active	One or more bits in the Warn Word are set
8 Event Handler Active	0	Event Handler Inactive	Event Handler cleared or disabled
	1	Event Handler Active	Event Handler setup

Bit Name	Val	Meaning	Remark
9 Special Motion Active	0	Normal Operation	
	1	Special Command runs	Special motion commands (Homing, ..) runs
10 In Target Position	0	Not In Pos	Motion active or actual position out of window
	1	In Pos	Actual position after motion in window
11 Homed	0	Motor not homed	Incremental sensor not homed (referenced)
	1	Motor homed	Position sensor system valid
12 Fatal Error	0		
	1	Fatal Error	A fatal error can not be acknowledged!
13 Motion Active	0	No Motion	Setpoint generation inactive
	1	Motion active	Setpoint generation (VAI, curve) active
14 Range Indicator 1	0	Not In Range 1	Defined UPID is not in Range 1
	1	In Range1	Defined UPID is in Range 1
15 Range Indicator 2	0	Not In Range 2	Defined UPID is not in Range 2
	1	In Range2	Defined UPID is in Range 2

3.26 Warn Word

Following table shows the meaning of the single bits of the Warn Word:

Bit Name	Val	Meaning
0 Motor Hot Sensor	0	Normal Operation
	1	Motor Temperature Sensor On
1 Motor Short Time Overload I ^Δ t	0	Normal Operation
	1	Calculated Motor Temperature Reached Warn Limit
2 Motor Supply Voltage Low	0	Normal Operation
	1	Motor Supply Voltage Reached Low Warn Limit
3 Motor Supply Voltage High	0	Normal Operation
	1	Motor Supply Voltage Reached High Warn Limit
4 Position Lag Always	0	Normal Operation
	1	Position Error during Moving Reached Warn Limit
5 Reserved	0	
	1	
6 Drive Hot	0	Normal Operation
	1	Temperature on servo drive High
7 Motor Not Homed	0	Normal Operation
	1	Warning Motor Not Homed Yet
8 PTC Sensor 1 Hot	0	Normal Operation
	1	PTC Temperature Sensor 1 On
9 Reserved PTC 2	0	Normal Operation
	1	PTC Temperature Sensor 2 On
10	0	Normal Operation

Bit Name	Val	Meaning
	1	Regenerative Resistor Temperature Hot Calculated
11 Speed Lag Always	0	Normal Operation
	1	Speed Lag is above Warn Limit
12 Position Sensor	0	Normal Operation
	1	Position is in Warn condition
13 Reserved	0	
	1	
14 Interface Warn Flag	0	Normal Operation
	1	Warn Flag Of Interface SW layer
15 Application Warn Flag	0	Normal Operation
	1	Warn Flag Of Application SW layer

Normally the warn word bits are used to react in conditions before the drive goes into the error state. E.g. a typical reaction on the warning 'Motor Temperature Sensor' would be a stop of the machine, before the drive goes into the error state and the motor goes out of control to avoid crashes.

4 Motion Command Interface

4.1 Motion Command Interface

The motion command interface consists of one word that contains the command ID, and up to 16 command parameter words. Example: 'VA-Interpolator 16 bit Go To Absolute Position'

Word	Description	Example of command
1.	Command Header with ID	Go To Absolute Position Immediate
2.	1. Command Parameter	Position
3.	2. Command Parameter	Maximal Speed
4.	3. Command Parameter	Acceleration
5.	4. Command Parameter	Deceleration
6.-16.	5.-15. Command Parameter	Not used

4.1.1 Command Header

Master ID								Sub ID				Command Count			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The header of the Motion command is split into three parts:

- Master ID
- Sub ID
- Command Count

4.1.1.1 Master ID

The master ID specifies the command group.

4.1.1.2 Sub ID

The sub ID is used to identify different commands from the same command group.

4.1.1.3 Command Count

A new command will only be executed, if the value of the command count has changed. In the easiest way bit 0 can be toggled.

4.2 Overview Motion Commands

Master ID	Sub ID	C	C	E	E	Description
		1	1	1	1	
		1	2	2	4	
		0	5	0	0	
		0	0	0	0	
00h	0h	X	X	X	X	No Operation
	1h	X	X	X	X	Write Interface Control Word
	2h	X	X	X	X	Write Live Parameter
	3h	X	X	X	X	Write X4/X14 Intf Outputs with Mask
	5h	X	X	X	X	Select Position Controller Set
	8h	X	X	X	X	Clear Event Evaluation

Master ID	Sub ID	C 1 1 0 0	C 1 2 5 0 0	E 1 2 0 0	E 1 4 0 0	Description
	9h	X	X	X	X	Master Homing
	Ah	X	X	X	X	Reset I Position Controller
	Fh	X	X	X	X	Reset
01h	0h	X	X	X	X	VAI Go To Pos
	1h	X	X	X	X	VAI Increment Dem Pos
	2h	X	X	X	X	VAI Increment Target Pos
	3h	X	X	X	X	VAI Go To Pos From Act Pos And Act Vel
	4h	X	X	X	X	VAI Go To Pos From Act Pos Starting With Dem Vel = 0
	5h	X	X	X	X	VAI Increment Act Pos
	6h	X	X	X	X	VAI Increment Act Pos Starting with Dem Vel = 0
	7h	X	X	X	X	VAI Stop
	8h	X	X	X	X	VAI Go To Pos After Actual Command
	9h	X	X	X	X	VAI Go To Analog Pos
	Ah	X	X	X	X	VAI Go To Pos On Rising Trigger Event
	Bh	X	X	X	X	VAI Increment Target Pos On Rising Trigger Event
	Ch	X	X	X	X	VAI Go To Pos On Falling Trigger Event
	Dh	X	X	X	X	VAI Increment Target Pos On Falling Trigger Event
	Eh	X	X	X	X	VAI Change Motion Parameters On Positive Position Transition
	Fh	X	X	X	X	VAI Change Motion Parameters On Negative Position Transition
02h	0h	X	X	X	X	Predef VAI Go To Pos
	1h	X	X	X	X	Predef VAI Increment Dem Pos
	2h	X	X	X	X	Predef VAI Increment Target Pos
	3h	X	X	X	X	Predef VAI Go To Pos From Act Pos And Act Vel
	4h	X	X	X	X	Predef VAI Go To Pos From Act Pos Starting With Dem Vel = 0
	7h	X	X	X	X	Predef VAI Stop With Quick Stop Deceleration
	8h	X	X	X	X	Predef VAI Go To Pos After Actual Command
	Ah	X	X	X	X	Predef VAI Go To Pos On Rising Trigger Event
	Bh	X	X	X	X	Predef VAI Increment Target Pos On Rising Trigger Event
	Ch	X	X	X	X	Predef VAI Go To Pos On Falling Trigger Event
	Dh	X	X	X	X	Predef VAI Increment Target Pos On Falling Trigger Event
	Eh	X	X	X	X	Predef VAI Infinite Motion Positive Direction
	Fh	X	X	X	X	Predef VAI Infinite Motion Negative Direction
03h	0h	X	X	X	X	P Stream With Slave Generated Time Stamp
	1h	X	X	X	X	PV Stream With Slave Generated Time Stamp
	2h	X	X	X	X	P Stream With Slave Generated Time Stamp and Configured Period Time
	3h	X	X	X	X	PV Stream With Slave Generated Time Stamp and Configured Period Time

Master ID	Sub ID	C 1	C 2	E 1	E 2	Description
		1	2	2	4	
		0	5	0	0	
		0	0	0	0	
	4h	X	X	X	X	PVA Stream With Slave Generated Time Stamp
	5h	X	X	X	X	PV Stream With Slave Generated Time Stamp and Configured Period Time
	Ah		X			PVA Stream With Master Generated Time Stamp
	Fh	X	X	X	X	Stop Streaming
04h	0h	X	X	X	X	Time Curve With Default Parameters
	1h	X	X	X	X	Time Curve With Default Parameters From Act Pos
	2h	X	X	X	X	Time Curve To Pos With Default Speed
	3h	X	X	X	X	Time Curve To Pos With Adjustable Time
	4h	X	X	X	X	Time Curve With Adjustable Offset, Time Scale & Amplitude Scale
	5h	X	X	X	X	Time Curve With Adjustable Offset, Time & Amplitude Scale
	6h	X	X	X	X	Time Curve With Adjustable Offset, Time & Amplitude Scale On Rising Trigger Event
	7h	X	X	X	X	Time Curve With Adjustable Offset, Time & Amplitude Scale On Falling Trigger Event
	Ah	X	X	X	X	Time Curve To Pos With Default Speed On Rising Trigger Event
	Ch	X	X	X	X	Time Curve To Pos With Default Speed On Falling Trigger Event
	Eh	X	X	X	X	Time Curve To Pos With Adjustable Time On Rising Trigger Event
	Fh	X	X	X	X	Time Curve To Pos With Adjustable Time On Falling Trigger Event
05h	0h			X	X	Modify Curve Start Address in RAM
	1h			X	X	Modify Curve Info Block 16 Bit Value in RAM
	2h			X	X	Modify Curve Info block 32 Bit Value in RAM
	4h			X	X	Modify Curve Data Block 32 Bit in RAM
	5h			X	X	Modify Curve Data Block 64 Bit in RAM
	6h			X	X	Modify Curve Data Block 96 Bit in RAM
06h	0h			X	X	Reset and disable CAM motion
	1h			X	X	Enable CAM Motion on next Z
	3h	X	X	X	X	Start Encoder CAM From Actual Count and Pos
	4h			X	X	Setup Encoder CAM from Actual Counts With Delay Counts
	9h			X	X	Setup Encoder CAM On Rise Trigger Event With Delay Counts
	Ah			X	X	Setup Encoder CANM On Rise Trigger Event With Delay Counts, Target Pos and Length
	Bh			X	X	Setup Encoder CAM On Fall Trigger Event With Delay Counts
	Ch			X	X	Setup Encoder CAM On Fall Trigger Event With Delay Counts, Target Pos and Length
	Dh			X	X	Setup Encoder CAM On Rise Trigger Event With Delay Counts Amplitude scale and length

Master ID	Sub ID	C 1	C 2	E 1	E 2	Description
		1	1	1	1	
		1	2	2	4	
		0	5	0	0	
		0	0	0	0	
	Eh			X	X	Setup Encoder CAM On Fall Trigger Event With Delay Counts Amplitude scale and length
07h	0h	X	X	X	X	Start VAI Encoder Position Indexing
	1h	X	X	X	X	Start Predef VAI Encoder Position Indexing
	2h			X	X	Start Encoder Position Indexing with VA-Filters
	Eh	X	X	X	X	Stop Position Indexing And VAI Go To Pos
	Fh	X	X	X	X	Stop Position Indexing And Predefined VAI Go To Pos
09h	0h	X	X	X	X	VAI 16 Bit Go To Pos
	1h	X	X	X	X	VAI 16 Bit Increment Dem Pos
	2h	X	X	X	X	VAI 16 Bit Increment Target Pos
	3h	X	X	X	X	VAI 16 Bit Go To Pos From Act Pos And Act Vel
	4h	X	X	X	X	VAI 16 Bit Go To Pos From Act Pos Starting With Dem Vel = 0
	5h	X	X	X	X	VAI 16Bit Increment Act Pos
	6h	X	X	X	X	VAI Increment Act Pos Starting with Dem Vel = 0
	7h	X	X	X	X	VAI 16 Bit Stop
	8h	X	X	X	X	VAI 16 Bit Go To Pos After Actual Command
	Ah	X	X	X	X	VAI 16 Bit Go To Pos On Rising Trigger Event
	Bh	X	X	X	X	VAI 16 Bit Increment Target Pos On Rising Trigger Event
	Ch	X	X	X	X	VAI 16 Bit Go To Pos On Falling Trigger Event
	Dh	X	X	X	X	VAI 16 Bit Increment Target Pos On Falling Trigger Event
	Eh	X	X	X	X	VAI 16 Bit Change Motion Parameters On Positive Position Transition
	Fh	X	X	X	X	VAI 16 Bit Change Motion Parameters On Negative Position Transition
0Ah	0h	X	X	X	X	Predef VAI 16 Bit Go To Pos
	1h	X	X	X	X	Predef VAI 16 Bit Increment Dem Pos
	2h	X	X	X	X	Predef VAI 16 Bit Increment Target Pos
	3h	X	X	X	X	Predef VAI 16 Bit Go To Pos From Act Pos And Act Vel
	4h	X	X	X	X	Predef VAI 16 Bit Go To Pos From Act Pos Starting With Dem Vel = 0
	7h	X	X	X	X	Predef VAI 16 Bit Stop With Quick Stop Deceleration
	8h	X	X	X	X	Predef VAI 16 Bit Go To Pos After Actual Command
	Ah	X	X	X	X	Predef VAI 16 Bit Go To Pos On Rising Trigger Event
	Bh	X	X	X	X	Predef VAI 16 Bit Increment Target Pos On Rising Trigger Event
	Ch	X	X	X	X	Predef VAI 16 Bit Go To Pos On Falling Trigger Event
	Dh	X	X	X	X	Predef VAI 16 Bit Increment Target Pos On Falling Trigger Event
0Bh	0h			X	X	VAI Predef Acc Go To Pos
	1h			X	X	VAI Predef Acc Increment Dem Pos

Master ID	Sub ID	C 1	C 2	E 1	E 2	Description
		1	1	1	1	
		1	2	2	4	
		0	5	0	0	
		0	0	0	0	
	2h			X	X	VAI Predef Acc Increment Target Pos
	3h			X	X	VAI Predef Acc Go To Pos From Act Pos And Act Vel
	4h			X	X	VAI Predef Acc Go To Pos From Act Pos Starting With Dem Vel = 0
	8h			X	X	VAI Predef Acc Go To Pos After Actual Command
	Ah			X	X	VAI Predef Acc Go To Pos On Rising Trigger Event
	Bh			X	X	VAI Predef Acc Increment Target Pos On Rising Trigger Event
	Ch			X	X	VAI Predef Acc Go To Pos On Falling Trigger Event
	Dh			X	X	VAI Predef Acc Increment Target Pos On Falling Trigger Event
0Ch	0h	X	X	X	X	VAI Dec=Acc Go To Pos
	1h	X	X	X	X	VAI Dec=Acc Increment Dem Pos
	2h	X	X	X	X	VAI Dec=Acc Increment Target Pos
	3h	X	X	X	X	VAI Dec=Acc Go To Pos From Act Pos And Act Vel
	4h	X	X	X	X	VAI Dec=Acc Go To Pos From Act Pos Starting With Dem Vel = 0
	5h	X	X	X	X	VAI Dec=Acc Go To Pos With Max Curr
	6h	X	X	X	X	VAI Dec=Acc Go To Pos From Act Pos And Act Vel With Max Curr
	7h	X	X	X	X	VAI Dec=Acc Go To Pos From Act Pos, Dem Vel = 0 and With Max Curr
	8h	X	X	X	X	VAI Dec=Acc Go To Pos After Actual Command
	Ah	X	X	X	X	VAI Dec=Acc Go To Pos On Rising Trigger Event
	Bh	X	X	X	X	VAI Dec=Acc Increment Target Pos On Rising Trigger Event
	Ch	X	X	X	X	VAI Dec=Acc Go To Pos On Falling Trigger Event
	Dh	X	X	X	X	VAI Dec=Acc Increment Target Pos On Falling Trigger Event
	Eh	X	X	X	X	VAI Dec=Acc Infinite Motion Positive Direction
	Fh	X	X	X	X	VAI Dec=Acc Infinite Motion Negative Direction
0Dh	0h	X	X	X	X	VAI Go Relative To Captured Pos
	1h	X	X	X	X	VAI Dec=Acc 16 Bit Go To Pos
	2h	x				Predef VAI Increment Act Pos by Cmd Tab var 1
	4h	X	X	X	X	VAI Go To Cmd Table Var 1 Pos
	5h	X	X	X	X	VAI Go To Cmd Table Var 2 Pos
	6h	X	X	X	X	VAI Go To Cmd Table Var 1 Pos From Act Pos And Act Vel
	7h	X	X	X	X	VAI Go To Cmd Table Var 2 Pos From Act Pos And Act Vel
	8h		X	X	X	VAI Stop at Angle
	9h	X	X	X	X	VAI Increment Act Pos Starting with Dem Vel = 0 Reset I-Contr
	Ch	X	X	X	X	VAI Start Trig Rise Config VAI Command incremental
	Dh	X	X	X	X	VAI Start Trig Rise Config VAI Command incremental
	Eh	X	X	X	X	VAI Start Trig Rise Config VAI Command

Master ID	Sub ID	C 1	C 2	E 1	E 2	Description
		1	1	1	1	
		1	2	2	4	
		0	5	0	0	
		0	0	0	0	
	Fh	X	X	X	X	VAI Start Trig Rise Config VAI Command
0Eh	0h	X	X	X	X	Sin VA Go To Pos
	1h	X	X	X	X	Sin VA Increment Demand Pos
	4h	X	X	X	X	Sin VA Go To Pos From Actual Pos
	6h	X	X	X	X	Sin VA Increment Actual Pos
	8h	X	X	X	X	Sin VA Go To Pos After Actual Command
	9h	X	X	X	X	Sin VA Go To Analog Pos
	Ah	X	X	X	X	Sin VA Go To Pos On Rising Trigger Event
	Bh	X	X	X	X	Sin VA Increment Demand Pos On Rising Trigger Event
	Ch	X	X	X	X	Sin VA Go To Pos On Falling Trigger Event
	Dh	X	X	X	X	Sin VA Increment Demand Pos On Falling Trigger Event
0Fh	0h	X	X	X	X	Bestehorn VAJ Go To Pos
	1h	X	X	X	X	Bestehorn VAJ Increment Demand Pos
	2h	X				Bestehorn VAJ 16 Bit Go To Pos
	4h	X	X	X	X	Bestehorn VAJ Go To Pos From Actual Pos
	6h	X	X	X	X	Bestehorn VAJ Increment Actual Pos
	8h	X	X	X	X	Bestehorn VAJ Go To Pos After Actual Command
	9h	X	X	X	X	Bestehorn VAJ Go To Analog Pos
	Ah	X	X	X	X	Bestehorn VAJ Go To Pos On Rising Trigger Event
	Bh	X	X	X	X	Bestehorn VAJ Increment Demand Pos On Rising Trigger Event
	Ch	X	X	X	X	Bestehorn VAJ Go To Pos On Falling Trigger Event
	Dh	X	X	X	X	Bestehorn VAJ Increment Demand Pos On Falling Trigger Event
10h	0h			X	X	Encoder CAM Enable
	1h			X	X	Encoder CAM Disable
	2h			X	X	Encoder CAM Go To Sync Pos
	4h			X	X	Encoder CAM Set Value
11h	0h			X	X	Encoder CAM 1 Define Curve With Default Parameters
	1h			X	X	Encoder CAM 1 Define Curve From Act Pos
	2h			X	X	Encoder CAM 1 Define Curve To Pos
	3h			X	X	Encoder CAM 1 Define Curve From Pos To Pos In Counts
	4h			X	X	Encoder CAM 1 Define Curve To Pos In Counts
	5h			X	X	Encoder CAM 1 Define Curve with Amplitude Scale In Counts
	6h			X	X	Encoder CAM 1 Enable
	7h			X	X	Encoder CAM 1 Disable
	8h			X	X	Encoder CAM 1 Change Amplitude Scale and Length

Master ID	Sub ID	C 1	C 2	E 1	E 2	Description
		1 1 0 0	1 2 5 0	1 2 0 0	1 4 0 0	
12h	0h			X	X	Encoder CAM 2 Define Curve With Default Parameters
	1h			X	X	Encoder CAM 2 Define Curve From Act Pos
	2h			X	X	Encoder CAM 2 Define Curve To Pos
	3h			X	X	Encoder CAM 2 Define Curve From Pos To Pos In Counts
	4h			X	X	Encoder CAM 2 Define Curve To Pos In Counts
	5h			X	X	Encoder CAM 2 Define Curve with Amplitude Scale In Counts
	6h			X	X	Encoder CAM 2 Enable
	7h			X	X	Encoder CAM 2 Disable
	8h			X	X	Encoder CAM 2 Change Amplitude Scale and Length
20h	0h	X	X	X	X	Start Command Table Command
	1h	X	X	X	X	Start Command Table Command On Rising Trigger Event
	2h	X	X	X	X	Start Command Table Command On Falling Trigger Event
	8h		X	X	X	Modify Command Table 16 bit Parameter in RAM
	9h		X	X	X	Modify Command Table 32 bit Parameter in RAM
21h	0h	X	X	X	X	Wait Time
	1h	X	X	X	X	Wait Until Motion Finished
	2h	X	X	X	X	Wait Until In Target Position
	3h	X	X	X	X	Wait Until Rising Trigger Edge
	4h	X	X	X	X	Wait Until Falling Trigger Edge
	8h	X	X	X	X	Wait Time Defined With Cmd Table Var 1
	9h	X	X	X	X	Wait Time Defined With Cmd Table Var 2
22h	0h	X	X	X	X	Wait Until Demand Position Greater Than
	1h	X	X	X	X	Wait Until Demand Position Less Than
	2h	X	X	X	X	Wait Until Actual Position Greater Than
	3h	X	X	X	X	Wait Until Actual Position Less Than
	4h	X	X	X	X	Wait Until Difference Position Greater Than
	5h	X	X	X	X	Wait Until Difference Position Less Than
	6h	X	X	X	X	Wait Until Difference Position Unsigned Greater Than
	7h	X	X	X	X	Wait Until Difference Position Unsigned Less Than
	8h	X	X	X	X	Wait Until Demand Velocity Greater Than
	9h	X	X	X	X	Wait Until Demand Velocity Less Than
	Ah	X	X	X	X	Wait Until Actual Velocity Greater Than
	Bh	X	X	X	X	Wait Until Actual Velocity Less Than
	Eh	X	X	X	X	Wait Until Current Greater Than
	Fh	X	X	X	X	Wait Until Current Less Than

Master ID	Sub ID	C 1 1 0 0	C 1 2 5 0 0	E 1 2 2 0 0	E 1 4 0 0 0	Description
24h	0h	X	X	X	X	Set Cmd Table Var 1 To
	1h	X	X	X	X	Add To Cmd Table Var 1
	2h	X	X	X	X	Set Cmd Table Var 2 To
	3h	X	X	X	X	Add To Cmd Table Var 2
	8h	X	X	X	X	Write Cmd Table Var 1 To UPID RAM value
	9h	X	X	X	X	Write Cmd Table Var 2 To UPID RAM value
	Ch	X	X	X	X	Write UPID RAM value To Cmd Table Var 1
	Dh	X	X	X	X	Write UPID RAM value To Cmd Table Var 2
	Eh	X	X	X	X	Write UPID RAM value To UPID ROM value
25h	0h	X	X	X	X	IF Cmd Table Var 1 Less Than
	1h	X	X	X	X	IF Cmd Table Var 1 Greater Than
	2h	X	X	X	X	IF Cmd Table Var 2 Less Than
	3h	X	X	X	X	IF Cmd Table Var 2 Greater Than
	6h	X	X	X	X	IF Cmd Table Var 1 Less Than UPID Value
	7h	X	X	X	X	IF Cmd Table Var 2 Less Than UPID Value
	8h	X	X	X	X	IF Demand Position Less Than
	9h	X	X	X	X	IF Demand Greater Than
	Ah	X	X	X	X	IF Actual Position Less Than
	Bh	X	X	X	X	IF Actual Greater Than
	Ch	X	X	X	X	IF Difference Position Less Than
	Dh	X	X	X	X	IF Difference Greater Than
	Eh	X	X	X	X	IF Current Less Than
	Fh	X	X	X	X	IF Current Greater Than
26h	0h	X	X	X	X	IF Analog Val On X4.4 Less Than
	2h	X	X	X	X	IF Masked X4 Input Value Equal Than
	3h			X	X	IF Masked X6 input Value Equal Than
	4h	X	X	X	X	IF Masked Status Word Equal Than
	5h	X	X	X	X	IF Masked Warn Word Equal Than
	6h			X	X	IF CAM Counts Less Than
30h	0h			X		Encoder Winding Start With Default Parameters
	1h			X		Encoder Winding Start With Default Parameters At Revolutions
	4h			X		Encoder Winding Stop Adaptation Of Left/Right Position and Disturbance
	5h			X		Encoder Winding Restart Adaptation Of Left/Right Position and Disturbance
31h	0h			X		Encoder Curve Winding Start With Default Parameters

Master ID	Sub ID	C 1	C 2	E 1	E 2	Description
		1	1	1	1	
		1	2	2	4	
		0	5	0	0	
		0	0	0	0	
	1h			X		Encoder Curve Winding Start With Default Parameters At Revolutions
38h	0h	F	F	F	F	VAI Go To Pos With Force Ctrl Limit
	1h	F	F	F	F	VAI Go To Pos From Act Pos And Reset Force Control
	2h	F	F	F	F	Force Ctrl Change Target Force
	3h	F	F	F	F	VAI Go To Pos With Force Ctrl Limit and Target Force
	4h	F	F	F	F	VAI Go To Pos With Lower Force Ctrl Limit
	5h	F	F	F	F	VAI Go To Pos With Lower Force Ctrl Limit and Target Force
	6h	F	F	F	F	VAI Go To Pos From Act Pos And Reset Force Control Set I
	7h	F	F	F	F	VAI Increment Act Pos And Reset Force Control Set I
	8h	F	F	F	F	VAI Inc Act Pos With Higher Force Ctrl Limit and Target Force
	9h	F	F	F	F	VAI Inc Act Pos With Lower Force Ctrl Limit and Target Force
	Ah		F			VAI Go To Pos Tara Diff Analog In -10V..10V
	Bh		F			Standstill Tara Diff Analog In -10V..10V
39h	0h	X	X	X	X	Current Command Mode
	8h	X	X		X	Closed Loop Force Command Mode Rel
	9h	X	X		X	Closed Loop Force Command Mode Abs
	Fh		X	X	X	Change to Position Controlled mode
3Ah	0h		X		X	VAJI Go To Pos
	1h		X		X	VAJI Increment Dem Pos
	2h		X		X	VAJI Increment Target Pos
	4h		X		X	VAJI Go To Pos From Act Pos Starting With Dem Vel = 0 and Dem Acc = 0
	6h		X		X	VAJI Increment Act Pos Starting with Dem Vel = 0 and Dem Acc = 0
	Ah		X		X	VAJI Go To Pos On Rising Trigger Event
	Bh		X		X	VAJI Increment Target Pos On Rising Trigger Event
	Ch		X		X	VAJI Go To Pos On Falling Trigger Event
	Dh		X		X	VAJI Increment Target Pos On Falling Trigger Event
3Bh	0h		X		X	Predef VAJI Go To Pos
	1h		X		X	Predef VAJI Increment Dem Pos
	2h		X		X	Predef VAJI Increment Target Pos
	4h		X		X	Predef VAJI Go To Pos From Act Pos Starting With Dem Vel = 0 and Dem Acc = 0
	Ah		X		X	Predef VAJI Go To Pos On Rising Trigger Event
	Bh		X		X	Predef VAJI Increment Target Pos On Rising Trigger Event
	Ch		X		X	Predef VAJI Go To Pos On Falling Trigger Event
	Dh		X		X	Predef VAJI Increment Target Pos On Falling Trigger Event

Master ID	Sub ID	C	C	E	E	Description
		1	1	1	1	
		1	2	2	4	
		0	5	0	0	
		0	0	0	0	
3Ch	0h		X			VAI Modulo Go To Pos Positive Direction
	1h		X			VAI Modulo Go To Pos Negative Direction
	2h		X			VAI Modulo Go To Pos Shortest Way

F: only with Force Control Key

4.3 Detailed Motion Command Description

4.3.1 No Operation (000xh)

Name	Byte Offset	Description	Type	Unit
Header	0	No Operation (000xh)	UInt16	-

This command does nothing. It can be sent in any operational state. Could be used to set the command count nibble to defined value, 0 for example.

4.3.2 Write Interface Control Word (001xh)

Name	Byte Offset	Description	Type	Unit
Header	0	001xh: Write Interface Control Word	UInt16	-
1. Par	2	Interface Control Word	UInt16	-

This command allows writing the control word through the motion command interface. The fieldbus interfaces (PROFINET, Ethernet/IP, EtherCAT CANOpen, Profibus, LinRS, POWERLINK) offer other ways to access the control word directly. Mostly a direct access is more comfortable than the way over the motion command interface.

4.3.3 Write Live Parameter (002xh)

Name	Byte Offset	Description	Type	Unit
Header	0	002xh: Write Live Parameter	UInt16	-
1. Par	2	UPID (Unique Parameter ID)	UInt16	-
2. Par	4	Parameter Value, the Unit depends on Parameter	Div	Div

This command allows writing any live parameter's ram value through the motion command interface. The parameter has to be specified by its UPID (Unique Parameter ID). In order to keep the interface as simple as possible any parameter can be accessed as 32bit integer value. The drive's operating system will filter out the relevant number of bits for parameters with smaller data size (e.g. only the lowest bit is considered for Boolean parameters).

The fieldbus interfaces (CANOpen, DeviceNet, Profibus, LinRS, POWERLINK, EtherCAT) offer other ways to read and write parameter values directly. Mostly a direct access is more comfortable than the way over the motion command interface.

4.3.4 Write X4 Intf Outputs with Mask (003xh)

Name	Byte Offset	Description	Type	Unit
Header	0	003xh: Write X4 Intf Outputs with Mask	UInt16	-
1. Par	2	Bit Mask; Bit 0 = X4.3 Bit 1 = X4.4...	UInt16	-
2. Par	4	Bit Value; Bit 0 = X4.3, Bit 1 X4.4...	UInt16	-

This command allows writing the configured X4 interface outputs with a write mask through the motion command interface. To write an output, the corresponding bit in the mask must be set. Bit 0 is mapped to output X4.3, bit 1 to output X4.4 etc.

4.3.5 Select Position Controller Set (005xh)

Name	Byte Offset	Description	Type	Unit
Header	0	005xh: Select Position Controller Set	UInt16	-
1. Par	2	Controller Set Selection (0 = Set A, 1 = Set B)	UInt16	-

This command selects the active position controller set (A/B) UPID 0x1393. For set A the ID is 0 and for Set B the ID is 1.

4.3.6 Clear Event Evaluation (008xh)

Name	Byte Offset	Description	Type	Unit
Header	0	008xh: Clear Event Evaluation	UInt16	-

This command resets the event handler. The event handler becomes active, if a motion command has been sent, that does not immediately start, but waits with its execution until other conditions are fulfilled (e.g. command 'VAI Go To Pos On Rising Trigger Event'). The bit 8 of the status word shows, if the event handler is active.

Once the event handler becomes active, it remains active, until it is deactivated with this clear command. As long the event handler is active, the command to be executed on the event situation will be restarted each time the event condition is fulfilled.

4.3.7 Master Homing (009xh)

Name	Byte Offset	Description	Type	Unit
Header	0	009xh: Master Homing	UInt16	-
1. Par	2	Home Position	SInt32	0.1 um

This command can be used, if the master system knows the home position without going to the home state in the state machine. The passed value of the home position is stored in the RAM value of the parameter Home Position (UPID 13C7h), then the corresponding value of the parameter Slider Home Position (UPID 13CAh) is calculated and stored in the RAM value. Then a homing at actual position is done without going into the homing state.

4.3.8 Reset I Position Controller (00Axh)

Name	Byte Offset	Description	Type	Unit
Header	0	00Axh: Reset I Position Controller	UInt16	-

This command resets the I-part of the position controller.

4.3.9 Reset (00Fxx)

Name	Byte Offset	Description	Type	Unit
Header	0	00Fxx: Reset	UInt16	-

This command resets all firmware instances of the drive. Use this command with count = 0, otherwise the drive reboots cyclic!

4.3.10 VAI Go To Pos (010xx)

Name	Byte Offset	Description	Type	Unit
Header	0	010xx: VAI Go To Pos	UInt16	-
1. Par	2	Target Position	Int32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command sets a new Target Position and defines the maximal velocity acceleration and deceleration for the movement. The command execution starts immediately when the command has been sent. The set points (Demand Position, Demand Velocity and Demand Acceleration) are calculated by the internal Velocity Acceleration Interpolator (VAI). This command initializes the VAI with the current Demand Position and Demand Velocity value. Therefore it is possible to start a new command while a former command is still being executed.

4.3.11 VAI Increment Dem Pos (011xx)

Name	Byte Offset	Description	Type	Unit
Header	0	011xx:VAI Increment Dem Pos	UInt16	-
1. Par	2	Position Increment	Int32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command sets a new target position and defines the maximal velocity, acceleration and deceleration for going there. The new target position value will be determined by the firmware. It is calculated by adding the position increment argument to the demand position value. The demand position is the actual position setpoint on which the motor is controlled. The demand position value moves towards the target position value while a motion command is in execution.

4.3.12 VAI Increment Target Pos (012xh)

Name	Byte Offset	Description	Type	Unit
Header	0	012xh: VAI Increment Target Pos	UInt16	-
1. Par	2	Position Increment	SIInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command sets a new target position and defines the maximal velocity, acceleration and deceleration for going there. The new target position value will be determined by the firmware. It is calculated by adding the position increment argument to the (former) target position. The target position is the motion's end position and doesn't change during the execution of a motion command.

4.3.13 VAI Go To Pos From Act Pos And Act Vel (013xh)

Name	Byte Offset	Description	Type	Unit
Header	0	013xh: VAI Go To Pos From Act Pos And Act Vel	UInt16	-
1. Par	2	Target Position	SIInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command starts the new VAI set point generation from the actual position and actual velocity. Can be used after a press command.

4.3.14 VAI Go To Pos From Act Pos Starting With Dem Vel = 0 (014xh)

Name	Byte Offset	Description	Type	Unit
Header	0	014xh: VAI Go To Pos From Act Pos Starting With Dem Vel =0	UInt16	-
1. Par	2	Target Position	SIInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command starts the new VAI setpoint generation from the actual position and the start velocity is forced to zero. Can be used after a press command.

4.3.15 VAI Increment Act Pos (015xh)

Name	Byte Offset	Description	Type	Unit
Header	0	015xh: VAI Increment Act Pos	UInt16	-
1. Par	2	Position Increment	SIInt32	0.1 um

Name	Byte Offset	Description	Type	Unit
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command sets a new target position and defines the maximal velocity, acceleration and deceleration for going there. The new target position value will be determined by the firmware. It is calculated by adding the position increment argument to the actual position. The actual position is the effective motor position. This command can be used to perform a retraction move after a press command. If the position increment argument is zero, this command defines the actual motor position as new setpoint.

4.3.16 VAI Increment Act Pos Starting With Dem Vel = 0 (016xh)

Name	Byte Offset	Description	Type	Unit
Header	0	016xh: VAI Increment Act Pos Starting With Dem Vel = 0	UInt16	-
1. Par	2	Position Increment	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command starts the new VAI set point generation from the actual position and the start velocity is forced to zero. This command defines the maximal velocity, acceleration and deceleration for going to the target position. The new target position value will be determined by the firmware. It is calculated by adding the position increment argument to the actual position. The actual position is the effective motor position. This command can be used to perform a retraction move after a press command. If the position increment argument is zero, then this command defines the actual motor position as new set point.

4.3.17 VAI Stop (017xh)

Name	Byte Offset	Description	Type	Unit
Header	0	017xh: VAI Stop	UInt16	-
1. Par	2	Deceleration	UInt32	1E-5 m/s ²

This command can be used to stop the motor (deceleration to velocity zero). The command execution starts immediately after the command has been sent. The command initializes the VAI with the current Demand Position and Demand Velocity value.

4.3.18 VAI Go To Pos After Actual Command (018xh)

Name	Byte Offset	Description	Type	Unit
Header	0	018xh: VAI Go To Pos After Actual Command	UInt16	-
1. Par	2	Target Position	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command waits until the actual motion set point generation has finished, then starts the new defined VAI motion.

4.3.19 VAI Go To Analog Pos (019xh)

Name	Byte Offset	Description	Type	Unit
Header	0	019xh: VAI Go To Analog Pos	UInt16	-
1. Par	2	Maximal Velocity	UInt32	1E-6 m/s
2. Par	6	Acceleration	UInt32	1E-5 m/s ²
3. Par	10	Deceleration	UInt32	1E-5 m/s ²

This command sets a new Target Position Calculated from the analog input on X4.4 and defines the maximal velocity, acceleration and deceleration for the movement. The command execution starts immediately after the command has been sent. The setpoints (Demand Position, Demand Velocity and Demand Acceleration) are calculated by the internal Velocity Acceleration Interpolator (VAI). This command initializes the VAI with the current Demand Position and Demand Velocity value. Therefore it is possible to start a new command while a former command is still being executed.

4.3.20 VAI Go To Pos On Rising Trigger Event (01Axh)

Name	Byte Offset	Description	Type	Unit
Header	0	01Axh: VAI Go To Pos On Rising Trigger Event	UInt16	-
1. Par	2	Target Position	SLnt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command defines a new VAI motion command (see description of command 'VAI Go To Pos (010xh)'). The command will be started on each rising edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new VAI motion on the rising trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'. The trigger input must be configured with parameter UPID 0x1039.

4.3.21 VAI Increment Target Pos On Rising Trigger Event (01Bxh)

Name	Byte Offset	Description	Type	Unit
Header	0	01Bxh: VAI Increment Target Pos On Rising Trigger Event	UInt16	-
1. Par	2	Position Increment	SLnt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command defines a new VAI motion command (relative move, see description of command 'VAI Increment Target Pos (012xh)'). The command will be started on each rising edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new VAI motion on the rising

trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'. The trigger input must be configured with parameter UPID 0x1039.

4.3.22 VAI Go To Pos On Falling Trigger Event (01Cxh)

Name	Byte Offset	Description	Type	Unit
Header	0	01Cxh: VAI Go To Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	UInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command defines a new VAI motion command (see description of command 'VAI Go To Pos (010xh)'). The command will be started on each falling edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new VAI motion on the falling trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'. The trigger input must be configured with parameter UPID 0x1039.

4.3.23 VAI Increment Target Pos On Falling Trigger Event (01Dxh)

Name	Byte Offset	Description	Type	Unit
Header	0	01Dxh: VAI Increment Target Pos On Falling Trigger Event	UInt16	-
1. Par	2	Position Increment	UInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command defines a new VAI motion command (relative move, see description of command 'VAI Increment Target Pos (012xh)'). The command will be started on each falling edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new VAI motion on the falling trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'. The trigger input must be configured with parameter UPID 0x039.

4.3.24 VAI Change Motion Parameters On Positive Position Transition (01Exh)

Name	Byte Offset	Description	Type	Unit
Header	0	01Exh: VAI Change Motion Parameters On Positive Position Transition	UInt16	-
1. Par	2	Transition Event Position	UInt32	0.1 um
2. Par	6	Max Velocity After Event	UInt32	1E-6 m/s
3. Par	10	Acceleration After Event	UInt32	1E-5 m/s ²
4. Par	14	Deceleration After Event	UInt32	1E-5 m/s ²

This motion command moves an event change position, an event maximal speed, an event acceleration and an event deceleration to the event instance, and starts the event evaluation. As soon as the demand position

crosses the event change position in the positive direction the VAI is changed with event values, the target position rests unchanged. As long the position transition event handler is active, only absolute VAI motion commands should be used (otherwise the VAI Target Position is not correctly defined).

4.3.25 VAI Change Motion Parameters On Negative Position Transition (01Fhx)

Name	Byte Offset	Description	Type	Unit
Header	0	01Fhx: VAI Change Motion Parameters On Negative Position Transition	UInt16	-
1. Par	2	Transition Event Position	SInt32	0.1 um
2. Par	6	Max Velocity After Event	UInt32	1E-6 m/s
3. Par	10	Acceleration After Event	UInt32	1E-5 m/s ²
4. Par	14	Deceleration After Event	UInt32	1E-5 m/s ²

This motion command moves an event change position, an event maximal speed, an event acceleration and an event deceleration to the event instance, and starts the event evaluation. As soon as the demand position crosses the event change position in the negative direction the VAI is changed with event values, the target position rests unchanged. As long the position transition event handler is active, only absolute VAI motion commands should be used (otherwise the VAI Target Position is not correctly defined).

4.3.26 Predef VAI Go To Pos (020hx)

Name	Byte Offset	Description	Type	Unit
Header	0	020hx: Predef VAI Go To Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command works like the command 'VAI Go To Pos (010hx)'. The VAI parameters (Maximal Velocity, Acceleration and Deceleration) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0.

4.3.27 Predef VAI Increment Dem Pos (021hx)

Name	Byte Offset	Description	Type	Unit
Header	0	021hx: Predef VAI Increment Dem Pos	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um

This command works like the command 'VAI Increment Dem Pos (011hx)'. The VAI parameters (Maximal Velocity, Acceleration and Deceleration) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0.

4.3.28 Predef VAI Increment Target Pos (022hx)

Name	Byte Offset	Description	Type	Unit
Header	0	022hx: Predef VAI Increment Target Pos	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um

This command works like the command 'VAI Increment Target Pos (012xh)'. The VAI parameters (Maximal Velocity, Acceleration and Deceleration) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0.

4.3.29 Predef VAI Go To Pos From Act Pos and Act Vel (023xh)

Name	Byte Offset	Description	Type	Unit
Header	0	023xh: Predef VAI Go To Pos From Act Pos And Act Vel	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command starts the new VAI setpoint generation from the actual position, can be used after a press command.

4.3.30 Predef VAI Go To Pos From Act Pos Starting With Dem Vel = 0 (024xh)

Name	Byte Offset	Description	Type	Unit
Header	0	024xh: Predef VAI Go To Pos From Act Pos With Dem Vel = 0	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command starts the new VAI-Setpoint generation from the actual position and the start velocity is forced to zero can be used after a press command.

4.3.31 Predef VAI Stop (027xh)

Name	Byte Offset	Description	Type	Unit
Header	0	017xh: Predef VAI Stop With Quick Stop Deceleration	UInt16	-

This command can be used to stop the motor (deceleration to velocity zero). Command execution starts immediately when the command has been sent. The command initializes the VAI with the current Demand Position and Demand Velocity value. It takes the deceleration value from parameter UPID 0x1721.

4.3.32 Predef VAI Go To Pos After Actual Command (028xh)

Name	Byte Offset	Description	Type	Unit
Header	0	028xh: Predef VAI Go To Pos After Actual Command	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command waits until the actual motion set point generation has finished, then starts the new defined VAI motion.

4.3.33 Predef VAI Go To Pos On Rising Trigger Event (02Axh)

Name	Byte Offset	Description	Type	Unit
Header	0	02Axh: Predef VAI Go To Pos On Rising Trigger Event	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command works like the command 'VAI Go To Pos On Rising Trigger Event (01Axh)'. The VAI parameters (Maximal Velocity, Acceleration and Deceleration) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0.

4.3.34 Predef VAI Increment Target Pos On Rising Trigger Event (02Bxh)

Name	Byte Offset	Description	Type	Unit
Header	0	02Bxh: Predef VAI Increment Target Pos On Rising Trigger Event	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um

This command works like the command 'VAI Increment Target Pos On Rising Trigger Event (01Bxh)'. The VAI parameters (Maximal Velocity, Acceleration and Deceleration) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0.

4.3.35 Predef VAI Go To Pos On Falling Trigger Event (02Cxh)

Name	Byte Offset	Description	Type	Unit
Header	0	02Cxh: Predef VAI Go To Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command works like the command 'VAI Go To Pos On Falling Trigger Event (01Cxh)'. The VAI parameters (Maximal Velocity, Acceleration and Deceleration) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0.

4.3.36 Predef VAI Go To Pos On Falling Trigger Event (02Dxh)

Name	Byte Offset	Description	Type	Unit
Header	0	02Cxh: Predef VAI Go To Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command works like the command 'VAI Increment Target Pos On Falling Trigger Event (01Dxh)'. The VAI parameters (Maximal Velocity, Acceleration and Deceleration) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0.

4.3.37 Predef VAI Infinite Motion Positive Direction (02Exh)

Name	Byte Offset	Description	Type	Unit
Header	0	02Exh: Predef VAI Infinite Motion Positive Direction	UInt16	-

This command starts a infinite motion in positive direction. The VAI parameters (Maximal Velocity, Acceleration and Deceleration) are not given as command arguments here, but are predefined through parameters UPID 0x14BE UPID 0x14BF and UPID 0x14C0.

4.3.38 Predef VAI Infinite Motion Negative Direction (02Fhx)

Name	Byte Offset	Description	Type	Unit
Header	0	02Fhx: Predef VAI Infinite Motion Negative Direction	UInt16	-

This command starts an infinite motion in negative direction. The VAI parameters (Maximal Velocity, Acceleration and Deceleration) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0.

4.3.39 P Stream With Slave Generated Time Stamp (030xh)

Name	Byte Offset	Description	Type	Unit
Header	0	030xh: P Stream With Slave Generated Time Stamp	UInt16	-
1. Par	2	Position	SInt32	0.1 um

Position streaming mode: The time stamp is generated by the interface (receive time stamp of T0), the streaming period has to be in the time range 2..5ms. For good results the streaming period has to be as constant as possible.

4.3.40 PV Stream With Slave Generated Time Stamp (031xh)

Name	Byte Offset	Description	Type	Unit
Header	0	031xh: PV Stream With Slave Generated Time Stamp	UInt16	-
1. Par	2	Position	SInt32	0.1 um
2. Par	6	Velocity	SInt32	1 um/s

Position and velocity streaming mode: Preferred streaming mode. The time stamp is generated by the interface (receive time stamp of T0), the streaming period has to be in the time range 2..5ms. Less sensitive to deviations of the streaming period.

4.3.41 P Stream With Slave Generated Time Stamp and Configured Period Time (032xh)

Name	Byte Offset	Description	Type	Unit
Header	0	032xh: P Stream With Slave Generated Time Stamp and Configured Period Time	UInt16	-
1. Par	2	Position	SInt32	0.1 um

Position streaming mode: The time stamp is generated by the interface (receive time stamp of T0), the streaming period has to be in the time range 2..5ms. For good results the streaming period has to be as constant as possible. For the derivation of the velocity and the acceleration the configured streaming period time 14E6h is taken to minimize communication time jittering.

4.3.42 PV Stream With Slave Generated Time Stamp and Configured Period Time (033xh)

Name	Byte Offset	Description	Type	Unit
Header	0	033xh: PV Stream With Slave Generated Time Stamp Stamp and Configured Period Time	UInt16	-
1. Par	2	Position	SInt32	0.1 um
2. Par	6	Velocity	SInt32	1 um/s

This command can be used for cyclic streaming of position and velocity setpoints. This is possible if master provides NC functionality. The commands have to be sent strictly cyclic in order to prevent buffer

underflow/buffer overflow errors. The period length must be in the time range of 0.4ms .. 10ms. The setpoint time stamp is generated by the slave's interface when the command is received (receive time stamp). The first two streaming commands are used to determine the nominal period length. The setpoints are then delayed 1.5 times the period length. Between the setpoints, the slave performs a fine interpolation. Since acceleration setpoint derivation is less sensitive to bus jitters, it is recommended to use this command (PV streaming) instead of the simple P streaming command whenever possible. For the derivation of the velocity and the acceleration the configured streaming period time 14E6h is taken to minimize communication time jittering.

4.3.43 PVA Stream With Slave Generated Time Stamp (034xh)

Name	Byte Offset	Description	Type	Unit
Header	0	034xh: PVA Stream With Slave Generated Time Stamp	UInt16	-
1. Par	2	Position	SIInt32	0.1 µm
2. Par	6	Velocity	SIInt32	1 µm/s
3. Par	10	Acceleration	SIInt32	1E-5m/s ²

This command can be used for cyclic position, velocity and acceleration setpoint streaming. This is possible if master provides NC functionality. The commands have to be sent strictly cyclic in order to prevent buffer underflow/buffer overflow errors. The period length should be in the time range of 0.4ms .. 10ms. The setpoint time stamp is generated by the slave's interface when the command is received (receive time stamp). The first two streaming commands are used to determine the nominal period length. The setpoints are then delayed 1.5 times the period length. Between the setpoints, the slave performs a fine interpolation. For good results the streaming period has to be as constant as possible.

4.3.44 PVA Stream With Slave Generated Time Stamp and Configured Period Time (035xh)

Name	Byte Offset	Description	Type	Unit
Header	0	035xh: PVA Stream With Slave Generated Time Stamp and Configured Period Time	UInt16	-
1. Par	2	Position	SIInt32	0.1 µm
2. Par	6	Velocity	SIInt32	1 µm/s
3. Par	10	Acceleration	SIInt32	1E-5m/s ²

This command can be used for cyclic streaming of position, velocity and acceleration setpoints. This is possible if master provides NC functionality. The commands have to be sent strictly cyclic in order to prevent buffer underflow/buffer overflow errors. The period length must be in the time range of 0.4ms .. 10ms. The setpoint time stamp is generated by the slave's interface when the command is received (receive time stamp). The first two streaming commands are used to determine the nominal period length. The setpoints are then delayed 1.5 times the period length. Between the setpoints, the slave performs a fine interpolation. Since acceleration setpoint derivation is less sensitive to bus jitters, it is recommended to use this command (PVA streaming) instead of the simple P streaming command whenever possible. For the derivation of the velocity and the acceleration the configured streaming period time 14E6h is taken to minimize communication time jittering.

4.3.45 PVA Stream With Master Generated Time Stamp (03Axh) EtherNet/IP CIP Sync

Name	Byte Offset	Description	Type	Unit
Header	0	03Axh: PVA Stream With Master Generated Time Stamp	UInt16	-
1. Par	2	Position	SIInt32	0.1 µm
2. Par	6	Velocity	SIInt32	1 µm/s

Name	Byte Offset	Description	Type	Unit
3. Par	10	Acceleration	SInt32	1E-5m/s2

This command can be used for cyclic streaming of position, velocity and acceleration setpoints with CIP Sync of EtherNet/IP. The setpoint time stamp is generated by the master when the command is created and sent in the CIP Sync specific output assembly to the slave. In order to prevent buffer underflow/buffer overflow errors, the commands have to be sent strictly cyclic. The period length must be in the time range of 0.4ms .. 10ms and preset by the configuration assembly of the EtherNet/IP interface. The first streaming command is only used for streaming period validation. The setpoints are active at the time defined by the master timestamp. Between the setpoints, the slave performs a fine interpolation. Since the timestamp is generated by the master, the streaming is less sensitive to bus jitter. The master and slave clocks need to be synchronized over PTP. See documentation of the EtherNet/IP interface for more information.

4.3.46 Stop Streaming (03Fhx)

Name	Byte Offset	Description	Type	Unit
Header	0	03Fhx: Stop Streaming	UInt16	-

This command is used to leave the streaming mode. If the master stops sending streaming commands without using this command, the slave will generate a buffer underflow error.

4.3.47 Time Curve With Default Parameters (040xh)

Name	Byte Offset	Description	Type	Unit
Header	0	040xh: Time Curve With Default Parameters	UInt16	-
1. Par	2	Curve ID	UInt16	1..100

This command starts a time curve profile. The curve with the given ID must be stored on the drive. Offset, amplitude and time scaling are defined by the values of parameters UPID 0x14C9, UPID 0x14CA and UPID 0x14CB (default parameters).

4.3.48 Time Curve With Default Parameters From Act Pos (041xh)

Name	Byte Offset	Description	Type	Unit
Header	0	41xh: Time Curve With Default Parameters From Act Pos	UInt16	-
1. Par	2	Curve ID	UInt16	1..100

This command sets the curve offset to the actual position set point then starts the specified time curve with the default parameters (curve time scaling and curve amplitude scaling).

4.3.49 Time Curve To Pos With Default Speed (042xh)

Name	Byte Offset	Description	Type	Unit
Header	0	042xh: Time Curve To Pos With Default Speed	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Target Position	SInt32	0.1 um

This command sets the curve offset to the actual demand position and scales the curve the way that the target position is reached at the end (the scaling range is –2000%..2000%, if this range is exceeded an error will be generated) then starts the specified time curve with the default curve speed parameter.

4.3.50 Time Curve To Pos With Adjustable Time (043xh)

Name	Byte Offset	Description	Type	Unit
Header	0	043xh: Time Curve To Pos With Adjustable Time	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Target Position	SInt32	0.1 um
3. Par	8	Curve Time	UInt32	10us

This command sets the curve offset to the demand position and scales the curve such that the target position is reached at the end (the scaling range is –2000%..2000%, if this range is exceeded an error will be generated). The curve time is taken from the motion command.

4.3.51 Time Curve With Adjustable Offset, Time Scale & Amplitude Scale (044xh)

Name	Byte Offset	Description	Type	Unit
Header	0	044xh: Time Curve With Adjustable Offset, Time Scale & Amplitude Scale	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Offset	SInt32	0.1 um
3. Par	8	Time Scale [0..200%]	UInt16	0.01%
4. Par	10	Amplitude Scale [-2000%..2000%]	SInt16	0.1%

With this command all the curve parameters are defined.

4.3.52 Time Curve With Adjustable Offset, Time & Amplitude Scale (045xh)

Name	Byte Offset	Description	Type	Unit
Header	0	045xh: Time Curve With Adjustable Offset, Time & Amplitude Scale	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Offset	SInt32	0.1 um
3. Par	8	Curve Time	UInt32	10us
4. Par	12	Amplitude Scale [-2000%..2000%]	SInt16	0.1%

With this command all the curve parameters are defined, unlike command 44xh this command defines the absolute curve time.

4.3.53 Time Curve With Adjustable Offset, Time & Amplitude Scale On Rising Trigger Event (046xh)

Name	Byte Offset	Description	Type	Unit
Header	0	046xh: Time Curve With Adjustable Offset, Time & Amplitude Scale On Rise Trigger Event	UInt16	-

Name	Byte Offset	Description	Type	Unit
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Offset	SInt32	0.1 um
3. Par	8	Curve Time	UInt32	10us
4. Par	12	Amplitude Scale [-2000%..2000%]	SInt16	0.1%

On a rising trigger event start command 045xh.

4.3.54 Time Curve With Adjustable Offset, Time & Amplitude Scale On Falling Trigger Event (047xh)

Name	Byte Offset	Description	Type	Unit
Header	0	047xh: Time Curve With Adjustable Offset, Time & Amplitude Scale On Falling Trigger Event	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Offset	SInt32	0.1 um
3. Par	8	Curve Time	UInt32	10us
4. Par	12	Amplitude Scale [-2000%..2000%]	SInt16	0.1%

On a falling trigger event start command 045xh.

4.3.55 Time Curve To Pos With Default Speed On Rising Trigger Event (04Axh)

Name	Byte Offset	Description	Type	Unit
Header	0	04Axh: Time Curve To Pos With Default Speed On Rising Trigger Event	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Target Position	SInt32	0.1 um

On a rising trigger event start command 042xh.

4.3.56 Time Curve To Pos With Default Speed On Falling Trigger Event (04Cxh)

Name	Byte Offset	Description	Type	Unit
Header	0	04Cxh: Time Curve To Pos With Default Speed On Falling Trigger Event	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Target Position	SInt32	0.1 um

On a falling trigger event start command 042xh.

4.3.57 Time Curve To Pos With Adjustable Time On Rising Trigger Event (04Exh)

Name	Byte Offset	Description	Type	Unit
Header	0	04Exh: Time Curve Go To Absolute Position With Adjustable Time On Rising Trigger Event	UInt16	-

Name	Byte Offset	Description	Type	Unit
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Target Position	SInt32	0.1 µm
3 Par	8	Curve Time	UInt32	10us

On a rising trigger event start command 043xh.

4.3.58 Time Curve To Pos With Adjustable Time On Falling Trigger Event (04Fxx)

Name	Byte Offset	Description	Type	Unit
Header	0	04Fxx: Time Curve To Pos With Adjustable Time On Falling Trigger Event	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Target Position	SInt32	0.1 µm
3 Par	8	Curve Time	UInt32	10us

On a falling trigger event start command 043xh.

4.3.59 Modify Curve Start Address in RAM (050xx)

Name	Byte Offset	Description	Type	Unit
Header	0	050xx: Modify Curve Start Address in RAM	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Address Offset	UInt16	-

This command allows locating the curve start address (points to the start of the curve info header). A value of 0FFFh means no curve has been defined, can be used to clear a curve. When using this command the whole memory management of the curve data has to be done by the superior system! A curve can only be defined with a valid start address. If a start address is defined, the curve (curve info header and curve data points) has to be completely defined, otherwise this will lead to an unpredictable behavior!

4.3.60 Modify Curve Info Block 16 Bit Value in RAM (051xx)

Name	Byte Offset	Description	Type	Unit
Header	0	051xx: Modify Curve Info Block 16 Bit Value in RAM	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Offset in Byte	UInt16	0..
3. Par	6	Value	Sint16	-

This command allows to modify a 16 bit value (or half of a 32 bit value) in RAM within the curve info block of an existing curve profile with the given ID. The offset argument defines on which position within the curve info block the modification takes place (byte offset). In order to modify for instance the value of x-Length you first have to set the low word of the new curve length with Offset 32 and then the high word with offset 34.

The fieldbus interfaces (PROFINET, EtherNet/IP, EtherCAT, POWERLINK, SERCOS III, LinUDP, CANOpen, LinRS) offer a safer way to download and modify curve profiles (Curve Service). Writing with wrong offsets, may overwrite and destroy data of other curves and may lead to unpredictable behavior!

4.3.61 Modify Curve Info Block 32 Bit Value in RAM (052xh)

Name	Byte Offset	Description	Type	Unit
Header	0	052xh: Modify Curve Info Block 32 Bit Value in RAM	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Offset in Byte	UInt16	0..
3. Par	6	Value	Sint32	-

This command allows to modify a 32 bit value in RAM within the curve data block of an already existing curve profile with the given ID. The offset argument defines on which position within the curve data block the modification takes place (byte offset). In order to modify for instance the first position setpoint value (Sint32) you have to set the new position with offset 0. The second setpoint has offset 4, and so on. The fieldbus interfaces (PROFINET, EtherNet/IP, EtherCAT, POWERLINK, SERCOS III, LinUDP, CANOpen, LinRS) offer a safer way to download and modify curve profiles (Curve Service). Writing with wrong offsets, may overwrite and destroy data of other curves and may lead to unpredictable behavior!

4.3.62 Modify Curve Data Block 32 Bit Value in RAM (054xh)

Name	Byte Offset	Description	Type	Unit
Header	0	054xh: Modify Curve Data Block 32 Bit Value in RAM	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Offset in Byte	UInt16	0..
3. Par	6	Curve point value	Sint32	-

This command allows to modify in RAM a 32 bit value within the Curve Data Block of a curve with existing curve start address and curve info header block with the given ID. The offset argument defines on which position within the curve data block the modification takes place (byte offset). In order to modify for instance the first position setpoint value (Sint32) you have to set the new position with offset 0. The second setpoint has offset 4, and so on. The fieldbus interfaces (PROFINET, EtherNet/IP, EtherCAT, POWERLINK, SERCOS III, LinUDP, CANOpen, LinRS) offer a safer way to download and modify curve profiles (Curve Service). Writing with wrong offsets, may overwrite and destroy data of other curves and may lead to unpredictable behavior!

4.3.63 Modify Curve Data Block 64 Bit Value in RAM (055xh)

Name	Byte Offset	Description	Type	Unit
Header	0	055xh: Modify Curve Data Block 64 Bit Value in RAM	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Offset in Byte	UInt16	0..
3. Par	6	Curve point value	Sint32	-
4. Par	10	Curve point value	Sint32	-

This command allows to modify in RAM two 32 bit value within the curve data block of a curve with existing curve start address and curve info header block with the given ID. The offset argument defines on which position within the curve data block the modification takes place (byte offset). In order to modify for instance the first position setpoint value (Sint32) you have to set the new position with offset 0. The second setpoint has offset 4, and so on. The fieldbus interfaces (PROFINET, EtherNet/IP, EtherCAT, POWERLINK, SERCOS III, LinUDP, CANOpen, LinRS) offer a safer way to download and modify curve profiles (Curve Service). Writing with wrong offsets, may overwrite and destroy data of other curves and may lead to unpredictable behavior.

4.3.64 Modify Curve Data Block 96 Bit Value in RAM (056xh)

Name	Byte Offset	Description	Type	Unit
Header	0	056xh: Modify Curve Data Block 96 Bit Value in RAM	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Offset in Byte	UInt16	0..
3. Par	6	Curve point value	SInt32	-
4. Par	10	Curve point value	SInt32	-
4. Par	14	Curve point value	SInt32	-

This command allows to modify in RAM three 32 bit value within the curve data block of a curve with existing curve start address and curve info header block with the given ID. The offset argument defines on which position within the curve data block the modification takes place (byte offset). In order to modify for instance the first position setpoint value (SInt32) you have to set the new position with offset 0. The second setpoint has offset 4, and so on. The fieldbus interfaces (PROFINET, EtherNet/IP, EtherCAT, POWERLINK, SERCOS III, LinUDP, CANOpen, LinRS) offer a safer way to download and modify curve profiles (Curve Service). Writing with wrong offsets, may overwrite and destroy data of other curves and may lead to unpredictable behavior!

4.3.65 Reset and disable CAM motion (060xh)

Name	Byte Offset	Description	Type	Unit
Header	0	060xh: Reset and disable CAM motion	UInt16	-

The actual CAM motion is disabled.

4.3.66 Enable CAM Motion on next Z (061xh)

Name	Byte Offset	Description	Type	Unit
Header	0	061xh: Enable CAM Motion on next Z	UInt16	-

The actual CAM motion is enabled and restarted at next Z.

4.3.67 Start Encoder Cam From Actual Count and Pos (063xh)

Name	Byte Offset	Description	Type	Unit
Header	0	063xh: Start Encoder Cam From Actual Count and Pos	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Length	SInt32	1 Incr
3. Par	8	Stroke	SInt32	0.1 um

Start a CAM curve with the specified curve ID, curve length [incr] and curve stroke.

4.3.68 Setup Encoder Cam On Rising Trigger Event With Delay Counts (069xh)

Name	Byte Offset	Description	Type	Unit
Header	0	069xh: Setup Encoder Cam On Rising Trigger Event With Delay Counts	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Delay Count	UInt32	1 Incr

Setup in the event handler to start a cam curve on the rising trigger event with the specified curve ID and the specified delay counts. The specified curve ID is written to the RAM value of UPID 154Ah, and the specified cam start delay is written to the RAM value of UPID 154Ch. Use the infinite cam Length (UPID1527h) for triggered cam motions.

4.3.69 Setup Encoder Cam On Rising Trigger Event With Delay Counts, Target Pos and Length (06Axh)

Name	Byte Offset	Description	Type	Unit
Header	0	06Axh: Setup Encoder Cam On Rise Trigger Event With Delay Counts, Target Pos and Length	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Delay Count	SInt32	1 Incr
3. Par	8	Target Position	SInt32	0.1 um
4. Par	12	Curve length	SInt32	1 Incr

Setup in the event handler to start a cam curve on the rising trigger event with the specified curve ID and the specified delay counts to specified target position in specified counts. The specified curve ID is written to the RAM value of UPID 154Ah, and the specified cam start delay is written to the RAM value of UPID 154Ch. Use the infinite cam length (UPID1527h) for triggered cam motions.

4.3.70 Setup Encoder Cam On Falling Trigger Event With Delay Counts (06Bxh)

Name	Byte Offset	Description	Type	Unit
Header	0	06Bxh: Setup Encoder Cam On Falling Trigger Event With Delay Counts	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Delay Count	UInt32	1 Incr

Setup in the event handler to start a cam curve on the falling trigger event with the specified curve ID and the specified delay counts. The specified curve ID is written to the RAM value of UPID 154Bh, and the specified cam start delay is written to the RAM value of UPID 154Dh. Use the infinite cam length (UPID1527h) for triggered cam motions.

4.3.71 Setup Encoder Cam On Falling Trigger Event With Delay Counts, Target Pos and Length (06Cxh)

Name	Byte Offset	Description	Type	Unit
Header	0	06Cxh: Setup Encoder Cam On Falling Trigger Event With Delay Counts, Target Pos and Length	UInt16	-
1. Par	2	Curve ID	UInt16	1..100

Name	Byte Offset	Description	Type	Unit
2. Par	4	Curve Start Delay Count	SInt32	1 Incr
3. Par	8	Target Position	SInt32	0.1 um
4. Par	12	Curve length	SInt32	1 Incr

Setup in the event handler to start a cam curve on the falling trigger event with the specified curve ID and the specified delay counts to specified target position in specified counts. The specified curve ID is written to the RAM value of UPID 154Bh, and the specified cam start delay is written to the RAM value of UPID 154Dh. Use the infinite cam length (UPID1527h) for triggered cam motions.

4.3.72 Setup Encoder Cam On Rising Trigger Event With Delay Counts, Amplitude scale and Length (06Dxh)

Name	Byte Offset	Description	Type	Unit
Header	0	06Dxh: Setup Encoder Cam On Rise Trigger Event With Delay Counts, Amplitude scale and Length	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Delay Count	SInt32	1 Incr
3. Par	8	Amplitude Scale	Sint16	0.1 %
4. Par	10	Curve length in Counts	SInt32	1 Incr

Setup in the event handler to start a cam curve on the rising trigger event with the specified curve ID and the specified delay counts to specified target position in specified counts. The specified curve ID is written to the RAM value of UPID 154Ah, and the specified cam start delay is written to the RAM value of UPID 154Ch. Use the infinite cam length (UPID1527h) for triggered cam motions.

4.3.73 Setup Encoder Cam On Falling Trigger Event With Delay Counts, Amplitude scale and Length (06Exh)

Name	Byte Offset	Description	Type	Unit
Header	0	06Exh: Setup Encoder Cam On Fall Trigger Event With Delay Counts, Amplitude scale and Length	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Delay Count	SInt32	1 Incr
3. Par	8	Amplitude Scale	Sint16	0.1 %
4. Par	10	Curve length in Counts	SInt32	1 Incr

Setup in the event handler to start a cam curve on the rising trigger event with the specified curve ID and the specified delay counts to specified target position in specified counts. The specified curve ID is written to the RAM value of UPID 154Ah, and the specified cam start delay is written to the RAM value of UPID 154Ch. Use the infinite cam length (UPID1527h) for triggered cam motions.

4.3.74 Start VAI Encoder Position Indexing (070xh)

Name	Byte Offset	Description	Type	Unit
Header	0	070xh: Start VAI Encoder Position Indexing	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

Name	Byte Offset	Description	Type	Unit
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

Changes to the VAI encoder position indexing mode, the actual encoder count is to the target position. To stop the indexing mode use one of the commands (008xh), (07Exh) or (07Fhx).

4.3.75 Start Predef VAI Encoder Position Indexing (071xh)

Name	Byte Offset	Description	Type	Unit
Header	0	071xh: Start Predef VAI Encoder Position Indexing	UInt16	-
1. Par	2	Target Position	SInt32	0.1 μ m

Changes to the VAI encoder position indexing mode, the actual encoder count is to the target position. The VAI parameters maximal velocity, acceleration and deceleration are ones taken at calling time from the predefined VAI parameter set (UPIDs: 14BEh, 14BF and 14C0h). To stop the indexing mode use one of the commands (008xh), (07Exh) or (07Fhx).

4.3.76 Stop Position Indexing and VAI Go To Pos (07Exh)

Name	Byte Offset	Description	Type	Unit
Header	0	07Exh: Stop Position Indexing And VAI Go To Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command stops the position indexing mode and moves to the defined target position with a VAI command.

4.3.77 Stop Position Indexing and VAI Go To Pos (07Fhx)

Name	Byte Offset	Description	Type	Unit
Header	0	07Vxh: Stop Position Indexing And VAI Go To Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 μ m

This command clears the position indexing mode and moves to the defined target position with a Predefined VAI command.

4.3.78 VAI 16 Bit Go To Pos (090xh)

Name	Byte Offset	Description	Type	Unit
Header	0	090xh: VAI 16 Bit Go To Pos	UInt16	-
1. Par	2	Target Position	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled

Name	Byte Offset	Description	Type	Unit
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command is similar to the 010xh command, but the parameters are only 16 bit and scaled. The scaling is according to the parameter settings under \Motion Control SW\ Motion Interface\ 16 Bit Interface Scaling\.

4.3.79 VAI 16 Bit Increment Dem Pos (091xh)

Name	Byte Offset	Description	Type	Unit
Header	0	091xh: VAI 16 Bit Go To Relative Position	UInt16	-
1. Par	2	Position Increment	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command works like the command 'VAI Increment Dem Pos (011xh)', but the arguments for this command have to be defined as 16 bit values. The corresponding scaling factors between the transmitted 16 bit values and the internally used 32 bit values have to be defined by using parameters UPID 0x1455, UPID 0x1456 and UPID 0x1457.

4.3.80 VAI 16 Bit Increment Target Pos (092xh)

Name	Byte Offset	Description	Type	Unit
Header	0	092xh: VAI 16 Bit Increment Target Pos	UInt16	-
1. Par	2	Position Increment	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command works like the command 'VAI Increment Target Pos (012xh)', but the arguments for this command have to be defined as 16 bit values. The corresponding scaling factors between the transmitted 16 bit values and the internally used 32 bit values have to be defined by using parameters UPID 0x1455, UPID 0x1456 and UPID 0x1457.

4.3.81 VAI 16 Bit Go To Pos From Act Pos And Act Vel (093xh)

Name	Byte Offset	Description	Type	Unit
Header	0	093xh: VAI 16 Bit Go To Pos From Act Pos And Act Vel	UInt16	-
1. Par	2	Target Position	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command starts the new VAI set point generation from the actual position, can be used after a press command.

4.3.82 VAI 16 Bit Go To Pos From Act Pos Starting With Dem Vel = 0 (094xh)

Name	Byte Offset	Description	Type	Unit
Header	0	094xh: VAI 16 Bit Go To Pos From Act Pos Starting With Dem Vel = 0	UInt16	-
1. Par	2	Target Position	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command starts a new VAI set point generation from the actual position and the start velocity is forced to zero. Can be used after a press command.

4.3.83 VAI 16 Bit Increment Act Pos (095xh)

Name	Byte Offset	Description	Type	Unit
Header	0	095xh: VAI 16 Bit Increment Act Pos	UInt16	-
1. Par	2	Position Increment	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command works like the command 'VAI Increment Act Pos (015xh)', but the arguments for this command have to be defined as 16 bit values. The corresponding scaling factors between the transmitted 16 bit values and the internally used 32 bit values have to be defined by using parameters UPID 0x1455, UPID 0x1456 and UPID 0x1457.

4.3.84 VAI 16 Bit Increment Act Pos Starting With Dem Vel = 0 (096xh)

Name	Byte Offset	Description	Type	Unit
Header	0	096xh: VAI 16 Bit Increment Act Pos	UInt16	-
1. Par	2	Position Increment	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command works like the command 'VAI Increment Act Pos Starting with Dem Vel = 0 (016xh)', but the arguments for this command have to be defined as 16 bit values. The corresponding scaling factors between the transmitted 16 bit values and the internally used 32 bit values have to be defined by using parameters UPID 0x1455, UPID 0x1456 and UPID 0x1457.

4.3.85 VAI 16 Bit Stop (097xh)

Name	Byte Offset	Description	Type	Unit
Header	0	097xh: VAI 16 Bit Stop	UInt16	-
1. Par	2	Deceleration	UInt16	Scaled

This command works like the command 'VAI Stop (017xh)', but the deceleration arguments for this command has to be defined as 16 bit value. The corresponding scaling factor between the transmitted 16 bit value and the internally used 32 bit value has to be defined by using parameter UPID 0x1457.

4.3.86 VAI 16 Bit Go To Pos After Actual Command (098xh)

Name	Byte Offset	Description	Type	Unit
Header	0	098xh: VAI 16 Bit Go To Pos After Actual Command	UInt16	-
1. Par	2	Target Position	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command works like the command 'VAI Go To Pos After Actual Command (018xh)', but the arguments for this command have to be defined as 16 bit values. The corresponding scaling factors between the transmitted 16 bit values and the internally used 32 bit values have to be defined by using parameters UPID 0x1455, UPID 0x1456 and UPID 0x1457.

4.3.87 VAI 16 Bit Go To Pos On Rising Trigger Event (09Axh)

Name	Byte Offset	Description	Type	Unit
Header	0	9Axh: VAI 16 Bit Go To Pos On Rising Trigger Event	UInt16	-
1. Par	2	Target Position	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command works like the command 'VAI Go To Pos On Rising Trigger Event (01Axh)', but the arguments for this command have to be defined as 16 bit values. The corresponding scaling factors between the transmitted 16 bit values and the internally used 32 bit values have to be defined by using parameters UPID 0x1455, UPID 0x1456 and UPID 0x1457.

4.3.88 VAI 16 Bit Increment Target Pos On Rising Trigger Event (09Bxh)

Name	Byte Offset	Description	Type	Unit
Header	0	9Bxh: VAI 16 Bit Increment Target Pos On Rising Trigger Event	UInt16	-
1. Par	2	Position Increment	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled

Name	Byte Offset	Description	Type	Unit
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command works like the command 'VAI Increment Target Pos On Rising Trigger Event (01Bxh)', but the arguments for this command have to be defined as 16 bit values. The corresponding scaling factors between the transmitted 16 bit values and the internally used 32 bit values have to be defined by using parameters UPID 0x1455, UPID 0x1456 and UPID 0x1457.

4.3.89 VAI 16 Bit Go To Pos On Falling Trigger Event (09Cxh)

Name	Byte Offset	Description	Type	Unit
Header	0	09Cxh: VAI 16 Bit Go To Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command works like the command 'VAI Go To Pos On Falling Trigger Event (01Cxh)', but the arguments for this command have to be defined as 16 bit values. The corresponding scaling factors between the transmitted 16 bit values and the internally used 32 bit values have to be defined by using parameters UPID 0x1455, UPID 0x1456 and UPID 0x1457.

4.3.90 VAI 16 Bit Increment Target Pos On Falling Trigger Event (09Dxh)

Name	Byte Offset	Description	Type	Unit
Header	0	09Dxh: VAI 16 Bit Increment Target Pos On Falling Trigger Event	UInt16	-
1. Par	2	Position Increment	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled
3. Par	6	Acceleration	UInt16	Scaled
4. Par	8	Deceleration	UInt16	Scaled

This command works like the command 'VAI Increment Target Pos On Falling Trigger Event (01Dxh)', but the arguments for this command have to be defined as 16 bit values. The corresponding scaling factors between the transmitted 16 bit values and the internally used 32 bit values have to be defined by using parameters UPID 0x1455, UPID 0x1456 and UPID 0x1457.

4.3.91 VAI 16 Bit Change Motion Parameters On Positive Position Transition (09Exh)

Name	Byte Offset	Description	Type	Unit
Header	0	09Exh: VAI 16 Bit Change Motion Parameters On Positive Position Transition	UInt16	-
1. Par	2	Transition Event Position	SInt16	Scaled
2. Par	4	Max Velocity After Event	UInt16	Scaled
3. Par	6	Acceleration After Event	UInt16	Scaled

Name	Byte Offset	Description	Type	Unit
4. Par	8	Deceleration After Event	UInt16	Scaled

As soon as the demand position crosses the event change position in positive direction the parameters for velocity, acceleration and deceleration will be changed to the values defined in the command.

4.3.92 VAI 16 Bit Change Motion Parameters On Negative Position Transition (09Fhx)

Name	Byte Offset	Description	Type	Unit
Header	0	09Exh: VAI Change Motion Parameters on Negative Position Transition	UInt16	-
1. Par	2	Transition Event Position	SInt16	Scaled
2. Par	4	Max Velocity After Event	UInt16	Scaled
3. Par	6	Acceleration After Event	UInt16	Scaled
4. Par	8	Deceleration After Event	UInt16	Scaled

As soon as the demand position crosses the event change position in negative direction the parameters for velocity, acceleration and deceleration will be changed to the values defined in the command.

4.3.93 Predef VAI 16 Bit Go To Pos (0A0hx)

Name	Byte Offset	Description	Type	Unit
Header	0	0A0hx: Predef VAI 16 Bit Go To Pos	UInt16	-
1. Par	2	Target Position	SInt16	Scaled

This command works like the command 'VAI Go To Pos (010hx)'. The VAI parameters maximal velocity, acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0. The Target Position argument has to be defined as 16 bit value. The corresponding scaling factor between the transmitted 16 bit value and the internally used 32 bit value is defined by parameter UPID 0x1455.

4.3.94 Predef VAI 16 Bit Increment Dem Pos (0A1hx)

Name	Byte Offset	Description	Type	Unit
Header	0	0A1hx: Predef VAI 16 Bit Increment Dem Pos	UInt16	-
1. Par	2	Position Increment	SInt16	Scaled

This command works like the command 'VAI Increment Dem Pos (011hx)'. The VAI parameters maximal velocity, acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0. The Position Increment argument has to be defined as 16 bit value. The corresponding scaling factor between the transmitted 16 bit value and the internally used 32 bit value is defined by parameter UPID 0x1455.

4.3.95 Predef VAI 16 Bit Increment Target Pos (0A2hx)

Name	Byte Offset	Description	Type	Unit
Header	0	0A2hx: Predef VAI 16 Bit Increment Target Pos	UInt16	-

Name	Byte Offset	Description	Type	Unit
1. Par	2	Position Increment	SInt16	Scaled

This command works like the command 'VAI Increment Target Pos (012xh)'. The VAI parameters maximal velocity, acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0. The Position Increment argument has to be defined as 16 bit value. The corresponding scaling factor between the transmitted 16 bit value and the internally used 32 bit value is defined by parameter UPID 0x1455.

4.3.96 Predef VAI 16 Bit Go To Pos From Act Pos And Act Vel (0A3xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0A3xh: VAI 16 Bit Go To Pos From Act Pos And Act Vel	UInt16	-
1. Par	2	Target Position	SInt16	Scaled

This command starts a new VAI set point generation from the actual position, can be used after a press command.

4.3.97 Predef VAI 16 Bit Go To Pos From Act Pos Starting With Dem Vel = 0 (0A4xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0A4xh: Predef VAI 16 Bit Go To Pos From Act Pos Starting With Dem Vel = 0	UInt16	-
1. Par	2	Target Position	SInt16	Scaled

This command starts a new VAI set point generation from the actual position and the start velocity is forced to zero. Can be used after a press command.

4.3.98 Predef VAI 16 Bit Stop (0A7xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0A7xh: Predef VAI Stop With Quick Stop Deceleration	UInt16	-

This command can be used to stop the motor (deceleration to velocity zero). Command execution starts immediately when the command has been sent. The command initializes the VAI with the current Demand Position and Demand Velocity value. It takes the deceleration value from parameter UPID 0x1721.

4.3.99 Predef VAI 16 Bit Go To Pos After Actual Command (0A8xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0A8xh: Predef VAI 16 Bit Go To Pos After Actual Command	UInt16	-
1. Par	2	Target Position	SInt16	Scaled

This command waits until the actual motion set point generation has finished, then starts the new defined VAI motion.

4.3.100 Predef VAI 16 Bit Go To Pos On Rising Trigger Event (0AAxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0AAxh: Predef VAI 16 Bit Go To Pos On Rising Trigger Event	UInt16	-
1. Par	2	Target Position	SInt16	Scaled

This command works like the command 'VAI Go To Pos On Rising Trigger Event (01Axh)'. The VAI parameters maximal velocity, acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0. The Target Position argument has to be defined as 16 bit value. The corresponding scaling factor between the transmitted 16 bit value and the internally used 32 bit value is defined by parameter UPID 0x1455.

4.3.101 Predef VAI 16 Bit Increment Target Pos On Rising Trigger Event (0ABxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0ABxh: Predef VAI 16 Bit Increment Target Pos On Rising Trigger Event	UInt16	-
1. Par	2	Position Increment	SInt16	Scaled

This command works like the command 'VAI Increment Target Pos On Rising Trigger Event (01Bxh)'. The VAI parameters maximal velocity, acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0. The Position Increment argument has to be defined as 16 bit value. The corresponding scaling factor between the transmitted 16 bit value and the internally used 32 bit value is defined by parameter UPID 0x1455.

4.3.102 Predef VAI 16 Bit Go To Pos On Falling Trigger Event (0ACxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0ACxh: VAI 16 Bit Go To Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command works like the command 'VAI Go To Pos On Falling Trigger Event (01Cxh)'. The VAI parameters maximal velocity, acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0. The Target Position argument has to be defined as 16 bit value. The corresponding scaling factor between the transmitted 16 bit value and the internally used 32 bit value is defined by parameter UPID 0x1455.

4.3.103 Predef VAI 16 Bit Increment Target Pos On Falling Trigger Event (0ADxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0ADxh: Predef VAI 16 Bit Increment Target Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	SInt16	Scaled

This command works like the command 'VAI Increment Target Pos On Falling Trigger Event (01Dxh)'. The VAI parameters maximal velocity, acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BE, UPID 0x14BF and UPID 0x14C0. The Position Increment argument has to be defined as 16 bit value. The corresponding scaling factor between the transmitted 16 bit value and the internally used 32 bit value is defined by parameter UPID 0x1455.

4.3.104 VAI Predef Acc Go To Pos (0B0xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0B0xh: VAI Predef Acc Go To Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

This command works like the command 'VAI Go To Pos (010xh)'. The VAI parameters acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BF and UPID 0x14C0.

4.3.105 VAI Predef Acc Increment Dem Pos (0B1xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0B1xh:VAI Predef Acc Increment Dem Pos	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

This command works like the command 'VAI Increment Dem Pos (011xh)'. The VAI parameters acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BF and UPID 0x14C0.

4.3.106 VAI Predef Acc Increment Target Pos (0B2xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0B2xh: VAI Predef Acc Increment Target Pos	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

This command works like the command 'VAI Increment Target Pos (012xh)'. The VAI parameters acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BF and UPID 0x14C0.

4.3.107 VAI Predef Acc Go To Pos From Act Pos And Act Vel (0B3xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0B3xh: VAI Predef Acc Go To Pos From Act Pos And Act Vel	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

This command starts a new VAI setpoint generation from the actual position and with actual velocity. Can be used after a press command.

4.3.108 VAI Predef Acc Go To Pos From Act Pos Starting With Dem Vel = 0 (0B4xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0B4xh: VAI Predef Acc Go To Pos From Act Pos Starting With Dem Vel =0	UInt16	-
1. Par	2	Target Position	SLint32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

This command starts a new VAI setpoint generation from the actual position and the start velocity is forced to zero. Can be used after a press command.

4.3.109 VAI Predef Acc Go To Pos After Actual Command (0B8xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0B8xh: VAI Predef Acc Go To Pos After Actual Command	UInt16	-
1. Par	2	Target Position	SLint32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

This command waits until the actual motion setpoint generation has finished, then starts the new defined VAI motion.

4.3.110 VAI Predef Acc Go To Pos On Rising Trigger Event (0BAxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0BAxh: VAI Predef Acc Go To Pos On Rising Trigger Event	UInt16	-
1. Par	2	Target Position	SLint32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

This command works like the command 'VAI Go To Pos On Rising Trigger Event (01Axh)'. The VAI parameters acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BF and UPID 0x14C0.

4.3.111 VAI Predef Acc Increment Target Pos On Rising Trigger Event (0BBxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0BBxh: VAI Predef Acc Increment Target Pos On Rising Trigger Event	UInt16	-
1. Par	2	Position Increment	SLint32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

This command works like the command 'VAI Increment Target Pos On Rising Trigger Event (01Bxh)'. The VAI parameters acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BF and UPID 0x14C0.

4.3.112 VAI Predef Acc Go To Pos On Falling Trigger Event (0BCxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0BCxh: VAI Predef Acc Go To Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	SLint32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

This command works like the command 'VAI Go To Pos On Falling Trigger Event (01Cxh)'. The VAI parameters acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BF and UPID 0x14C0.

4.3.113 VAI Predef Acc Increment Target Pos On Falling Trigger Event (0BDxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0BDxh: VAI Predef Acc Increment Target Pos On Falling Trigger Event	UInt16	-
1. Par	2	Position Increment	SLint32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

This command works like the command 'VAI Increment Target Pos On Falling Trigger Event (01Dxh)'. The VAI parameters acceleration and deceleration are not given as command arguments here, but are predefined through the parameters UPID 0x14BF and UPID 0x14C0.

4.3.114 VAI Dec=Acc Go To Pos (0C0xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0C0xh: VAI Dec=Acc Go To Pos	UInt16	-
1. Par	2	Target Position	SLint32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command works like the command 'VAI Go To Pos (010xh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration.

4.3.115 VAI Dec=Acc Increment Dem Pos (0C1xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0C1xh: VAI Dec=Acc Increment Dem Pos	UInt16	-
1. Par	2	Position Increment	SLint32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command works like the command 'VAI Increment Dem Pos (011xh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration.

4.3.116 VAI Dec=Acc Increment Target Pos (0C2xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0C2xh: VAI Dec=Acc Increment Target Pos	UInt16	-
1. Par	2	Position Increment	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command works like the command 'VAI Increment Target Pos (012xh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration.

4.3.117 VAI Dec=Acc Go To Pos From Act Pos And Act Vel (0C3xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0C3xh: VAI Dec=Acc Go To Pos From Act Pos And Act Vel	UInt16	-
1. Par	2	Target Position	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command works like the command 'VAI Go To Pos From Act Pos And Act Vel (013xh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration.

4.3.118 VAI Dec=Acc Go To Pos From Act Pos Starting With Dem Vel = 0 (0C4xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0C4xh: VAI Dec=Acc Go To Pos From Act Pos Starting With Dem Vel =0	UInt16	-
1. Par	2	Target Position	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command works like the command 'VAI Go To Pos From Act Pos Starting with Dem Vel = 0 (014xh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration.

4.3.119 VAI Dec=Acc Go To Pos With Max Curr (0C5xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0C5xh: VAI Dec=Acc Go To Pos With Max Curr	UInt16	-
1. Par	2	Target Position	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²

Name	Byte Offset	Description	Type	Unit
4. Par	14	Demand Curr Limit	UInt16	1mA

This command works like the command 'VAI Go To Pos (010xh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration. Limits the maximal current of the actual position controller set to the defined value.

4.3.120 VAI Dec=Acc Go To Pos From Act Pos And Vel With Max Curr (0C6xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0C6xh: VAI Dec=Acc Go To Pos From Act Pos And Vel With Max Curr	UInt16	-
1. Par	2	Target Position	SLint32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²
4. Par	14	Demand Curr Limit	UInt16	1mA

This command works like the command 'VAI Go To Pos From Act Pos And Act Vel (013xh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration. Limits the maximal current of the actual position controller set to the defined value.

4.3.121 VAI Dec=Acc Go To Pos From Act Pos And Vel = 0 With Max Curr (0C7xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0C7xh: VAI Dec=Acc Go To Pos From Act Pos And Vel = 0 With Max Curr	UInt16	-
1. Par	2	Target Position	SLint32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²
4. Par	14	Demand Curr Limit	UInt16	1mA

This command works like the command 'VAI Go To Pos From Act Pos And Act Vel (013xh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration. Limits the maximal current of the actual position controller set to the defined value.

4.3.122 VAI Dec=Acc Go To Pos After Actual Command (0C8xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0C8xh: VAI Dec=Acc Go To Pos After Actual Command	UInt16	-
1. Par	2	Target Position	SLint32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command works like the command 'VAI Go To Pos After Actual Command (018xh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration.

4.3.123 VAI Dec=Acc Go To Pos On Rising Trigger Event (0CAxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0CAxh: VAI Dec=Acc Go To Pos On Rising Trigger Event	UInt16	-
1. Par	2	Target Position	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command works like the command 'VAI Go To Pos On Rising Trigger Event (01Axh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration.

4.3.124 VAI Dec=Acc Increment Target Pos On Rising Trigger Event (0CBxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0CBxh: VAI Dec=Acc Increment Target Pos On Rising Trigger Event	UInt16	-
1. Par	2	Position Increment	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command works like the command 'VAI Increment Target Pos On Rising Trigger Event (01Bxh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration.

4.3.125 VAI Dec=Acc Go To Pos On Falling Trigger Event (0CCxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0CCxh: VAI Dec=Acc Go To Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command works like the command 'VAI Go To Pos On Falling Trigger Event (01Cxh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration.

4.3.126 VAI Dec=Acc Increment Target Pos On Falling Trigger Event (0CDxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0CDxh: VAI Dec=Acc Increment Target Pos On Falling Trigger Event	UInt16	-
1. Par	2	Position Increment	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command works like the command 'VAI Increment Target Pos On Falling Trigger Event (01Dxh)', but here there aren't two different arguments for acceleration and deceleration. The Acceleration command argument value is also used for deceleration.

4.3.127 VAI Dec=AccVAI Infinite Motion Positive Direction (0CExh)

Name	Byte Offset	Description	Type	Unit
Header	0	0CExh: This command starts an infinite motion in positive direction	UInt16	-
1. Par	2	Maximal Velocity	UInt32	1E-6 m/s
2. Par	6	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command starts an infinite motion in positive direction.

4.3.128 VAI Dec=Acc VAI Infinite Motion Negative Direction (0CFxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0CFxh: This command starts an infinite motion in negative direction.	UInt16	-
1. Par	2	Maximal Velocity	UInt32	1E-6 m/s
2. Par	6	Acceleration = Deceleration	UInt32	1E-5 m/s ²

This command starts an infinite motion in negative direction.

4.3.129 VAI Increment Captured Pos (0D0xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0D0xh: VAI Increment Captured Pos	UInt16	-
1. Par	2	Captured Position Increment	SInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command sets a new Target Position and defines the maximal velocity, acceleration and deceleration for going there. The Target Position is calculated by adding the Position Increment argument and the value of the capture variable UPID 0x1E62. The command execution starts immediately when the command has been sent. The setpoints (Demand Position, Demand Velocity and Demand Acceleration) are calculated by the internal Velocity Acceleration Interpolator (VAI). This command initializes the VAI with the current Demand Position and Demand Velocity value. Therefore it is possible to start a new command, while execution of a former command is not fully completed.

4.3.130 VAI 16 Bit Dec=Acc Go To Pos (0D1xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0D1xh: VAI 16 Bit Dec=Acc Go To Pos	UInt16	-
1. Par	2	Target Position	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled

3. Par 6 Acceleration / Deceleration

UInt16 Scaled

This command sets a new Target Position and defines the maximal velocity, acceleration and deceleration for going there. The Target Position is calculated by adding the Position Increment argument and the value of the capture variable UPID 0x1E62. The command execution starts immediately when the command has been sent. The setpoints (Demand Position, Demand Velocity and Demand Acceleration) are calculated by the internal Velocity Acceleration Interpolator (VAI). This command initializes the VAI with the current Demand Position and Demand Velocity value. Therefore it is possible to start a new command, while execution of a former command is not fully completed.

4.3.131 Predef VAI Increment Act Pos by Cmd Tab Var1 (0D2xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0D2xh: Predef VAI Increment Act Pos by Cmd Tab Var 1	UInt16	-

Increment the actual position with the command table var 1 position with the predefined motion command parameters.

4.3.132 VAI Go To Cmd Tab Var1 Pos (0D4xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0D4xh: VAI Go To Cmd Tab Var1 Pos	UInt16	-
1. Par	2	Maximal Velocity	UInt32	1E-6 m/s
2. Par	6	Acceleration	UInt32	1E-5 m/s ²
3. Par	10	Deceleration	UInt32	1E-5 m/s ²

This command sets a new Target Position and defines the maximal velocity, acceleration and deceleration for going there. The command execution starts immediately when the command has been sent. The setpoints (Demand Position, Demand Velocity and Demand Acceleration) are calculated by the internal Velocity Acceleration Interpolator (VAI). This command initializes the VAI with the current Demand Position and Demand Velocity value. Therefore it is possible to start a new command, while execution of a former command is not fully completed.

4.3.133 VAI Go To Cmd Tab Var2 Pos (0D5xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0D5xh: VAI Go To Cmd Tab Var2 Pos	UInt16	-
1. Par	2	Maximal Velocity	UInt32	1E-6 m/s
2. Par	6	Acceleration	UInt32	1E-5 m/s ²
3. Par	10	Deceleration	UInt32	1E-5 m/s ²

This command sets a new Target Position and defines the maximal velocity, acceleration and deceleration for going there. The command execution starts immediately when the command has been sent. The setpoints (Demand Position, Demand Velocity and Demand Acceleration) are calculated by the internal Velocity Acceleration Interpolator (VAI). This command initializes the VAI with the current Demand Position and Demand Velocity value. Therefore it is possible to start a new command, while execution of a former command is not fully completed.

4.3.134 VAI Go To Cmd Tab Var1 Pos From Act Pos And Act Vel (0D6xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0D6xh: VAI Go To Cmd Tab Var1 Pos From Act Pos And Act Vel	UInt16	-
1. Par	2	Maximal Velocity	UInt32	1E-6 m/s
2. Par	6	Acceleration	UInt32	1E-5 m/s ²
3. Par	10	Deceleration	UInt32	1E-5 m/s ²

This command sets a new Target Position and defines the maximal velocity, acceleration and deceleration for going there. The command execution starts immediately when the command has been sent. The setpoints (Demand Position, Demand Velocity and Demand Acceleration) are calculated by the internal Velocity Acceleration Interpolator (VAI). This command initializes the VAI with the current Actual Position and Actual Velocity value. Therefore it is possible to start a new command, while execution of a former command is not fully completed.

4.3.135 VAI Go To Cmd Tab Var2 Pos From Act Pos And Act Vel (0D7xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0D7xh: VAI Go To Cmd Tab Var2 Pos From Act Pos And Act Vel	UInt16	-
1. Par	2	Maximal Velocity	UInt32	1E-6 m/s
2. Par	6	Acceleration	UInt32	1E-5 m/s ²
3. Par	10	Deceleration	UInt32	1E-5 m/s ²

This command sets a new Target Position and defines the maximal velocity, acceleration and deceleration for going there. The command execution starts immediately when the command has been sent. The setpoints (Demand Position, Demand Velocity and Demand Acceleration) are calculated by the internal Velocity Acceleration Interpolator (VAI). This command initializes the VAI with the current Actual Position and Actual Velocity value. Therefore it is possible to start a new command, while execution of a former command is not fully completed.

4.3.136 VAI Stop at Angle (0D8xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0D8xh: VAI Stop at Angle	UInt16	-
1. Par	2	Angle	UInt16	360°/2 ¹⁶
2. Par	4	Deceleration	UInt32	1E-5 m/s ²

This command can be used to stop the motor at a specified angle with the specified deceleration (deceleration to velocity zero). This command could only be used on ABZ sensor feedback on X13 and if 360° (one revolution) is a power of 2 number.

4.3.137 VAI Increment Act Pos Starting With Dem Vel = 0 Reset I-Contr (0D9xh)

Name	Byte Offset	Description	Type	Unit
Header	0	016xh: VAI Increment Act Pos Starting With Dem Vel = 0 Reset I-Contr	UInt16	-

1. Par	2	Position Increment	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command resets the I-part of the position controller and starts the new VAI setpoint generation from the actual position and the start velocity is forced to zero. This command defines the maximal velocity, acceleration and deceleration for going to the target position. The new target position value will be determined by the firmware. It is calculated by adding the position increment argument to the actual position. The actual position is the effective motor position. This command can be used to perform a retraction move after a press command. If the position increment argument is zero, then this command defines the actual motor position as new setpoint.

4.3.138 VAI Start Trig Rise Config VAI Command (0DCxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0DCxh: VAI Start Trig Rise Config VAI Command incremental	UInt16	-

This command starts the VAI motion command incremental, defined with the parameters in Trig Rise Config of the Triggered VA-Interpolator Run Mode settings.

4.3.139 VAI Start Trig Fall Config VAI Command (0DDxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0DDxh: VAI Start Trig Fall Config VAI Command incremental	UInt16	-

This command starts the VAI motion command incremental, defined with the parameters in Trig Fall Config of the Triggered VA-Interpolator Run Mode settings.

4.3.140 VAI Start Trig Rise Config VAI Command (0DExh)

Name	Byte Offset	Description	Type	Unit
Header	0	0DExh: VAI Start Trig Rise Config VAI Command	UInt16	-

This command starts the VAI motion command, defined with the parameters in Trig Rise Config of the Triggered VA-Interpolator Run Mode settings.

4.3.141 VAI Start Trig Fall Config VAI Command (0DFxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0DFxh: VAI Start Trig Fall Config VAI Command	UInt16	-

This command starts the VAI motion command, defined with the parameters in Trig Fall Config of the Triggered VA-Interpolator Run Mode settings.

4.3.142 Sin VA Go To Pos (0E0xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0E0xh: Sin VA Go To Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²

Half period sine motion profile, regarding the limitations of maximal acceleration and maximal velocity.

4.3.143 Sin VA Increment Demand Pos (0E1xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0E1xh: Sin VA Increment Demand Pos	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²

Half period sine motion profile, regarding the limitations of maximal acceleration and maximal velocity. The new Target Position value will be determined by the firmware. It is calculated by adding the Position Increment argument to the Demand Position value (relative move). The command execution starts immediately when the command has been sent.

4.3.144 Sin VA Go To Pos From Actual Pos (0E4xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0E4xh: Sin VA Go To Pos From Actual Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²

Half period sine motion profile, regarding the limitations of maximal acceleration and maximal velocity. The command execution starts immediately when the command has been sent. This command should be used if the Actual Position does not match with the current Demand Position value, but it can be assumed that the motor stands still (Actual Velocity assumed to be zero, e.g. because the motor stands on a hard stop). This can happen after a Press command, where the actual motor position is defined through external conditions and the motor can not and had not to follow the demand position. By starting this command, the former accepted difference between Actual Position and Demand Position can be eliminated.

4.3.145 Sin VA Increment Actual Pos (0E6xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0E6xh: Sin VA Increment Actual Pos	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s

Name	Byte Offset	Description	Type	Unit
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²

Half period sine motion profile, regarding the limitations of maximal acceleration and maximal velocity. The new Target Position value will be determined by the firmware. It is calculated by adding the Position Increment argument to the Actual Position value (relative move). The command execution starts immediately when the command has been sent.

4.3.146 Sin VA Go To Pos After Actual Command (0E8xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0E8xh: Sin VA Go To Pos After Actual Command	UInt16	-
1. Par	2	Target Position	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²

Half period sine motion profile, regarding the limitations of maximal acceleration and maximal velocity. The command execution is delayed until the former command has been completed. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new VAI motion command as soon as bit 13 of the Status Word (Motion Active) is 0. After the event, the Event Handler deactivates itself.

4.3.147 Sin VA Go To Analog Pos (0E9xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0E9xh: Sin VA Go To Analog Pos	UInt16	-
1. Par	2	Maximal Velocity	UInt32	1E-6 m/s
2. Par	6	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²

Half period sine motion profile, regarding the limitations of maximal acceleration and maximal velocity. This command sets a new Target Position Calculated from the analog input on X4.4 or X20.

4.3.148 Sin VA Go To Pos On Rising Trigger Event (0EAxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0EAxh: Sin VA Go To Pos On Rising Trigger Event	UInt16	-
1. Par	2	Target Position	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²

This command defines a new Cos motion command (see description of command 'Sin VA Go To Pos (0E0xh)'). The command will be started on each rising edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new Sin VA motion on the rising trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'. The trigger input must be configured with parameter UPID 0x1039.

4.3.149 Sin VA Increment Demand Pos On Rising Trigger Event (0EBxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0EBxh: Sin VA Increment Demand Pos On Rising Trigger Event	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²

This command defines a new Sin VA motion command (relative move, see description of command 'Sin VA Increment Demand Pos (0E1xh)'). The command will be started on each rising edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new Sin VA motion on the rising trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'. The trigger input must be configured with parameter UPID 0x1039.

4.3.150 Sin VA Go To Pos On Falling Trigger Event (0ECxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0ECxh: Sin VA Go To Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²

This command defines a new Sin VA motion command (see description of command 'Sin VA Go To Pos (0E0xh)'). The command will be started on each falling edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new Sin VA motion on the falling trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'. The trigger input must be configured with parameter UPID 0x1039.

4.3.151 Sin VA Increment Demand Pos On Falling Trigger Event (0EDxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0EDxh: Sin VA Increment Demand Pos On Falling Trigger Event	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²

This command defines a new Sin VA motion command (relative move, see description of command 'Sin VA Increment Demand Pos (0E1xh)'). The command will be started on each falling edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new Sin VA motion on the falling trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'. The trigger input must be configured with parameter UPID 0x1039.

4.3.152 Bestehorn VAJ Go To Pos (0F0xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0F0xh: Bestehorn VAJ Go To Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Jerk	UInt32	1E-4 m/s ³

Bestehorn Motion Profile, regarding the limitations maximal jerk, acceleration and maximal speed.

4.3.153 Bestehorn VAJ Increment Demand Pos (0F1xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0F1xh: Bestehorn VAJ Increment Demand Pos	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Jerk	UInt32	1E-4 m/s ³

Bestehorn Motion Profile, regarding the limitations maximal jerk, acceleration and maximal speed. The new Target Position value will be determined by the firmware. It is calculated by adding the Position Increment argument to the Demand Position value (relative move). The command execution starts immediately when the command has been sent.

4.3.154 Bestehorn VAJ 16 BitGo To Pos (0F2xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0F2xh: Bestehorn VAJ 16 Bit Go To Pos	UInt16	-
1. Par	2	Target Position	SInt16	Scaled
2. Par	4	Maximal Velocity	UInt16	Scaled
3. Par	6	Maximal Acceleration/Deceleration	UInt16	Scaled
4. Par	8	Maximal Jerk	UInt16	Scaled

Bestehorn Motion Profile, regarding the limitations maximal jerk, acceleration and maximal speed. With 16 bit motion parameters: UPID 0x1455 for Position, UPID 0x1456 for Maximal Velocity, UPID 0x1457 for Acceleration and UPID 0x1458 for Jerk Scaling.

4.3.155 Bestehorn VAJ Go To Pos From Actual Pos (0F4xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0F4xh: Bestehorn VAJ Go To Pos From Actual Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

Name	Byte Offset	Description	Type	Unit
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Jerk	UInt32	1E-4 m/s ³

Bestehorn Motion Profile, regarding the limitations maximal jerk, acceleration and maximal speed. The command execution starts immediately when the command has been sent. This command should be used if the Actual Position does not match with the current Demand Position value, but it can be assumed that the motor stands still (Actual Velocity assumed to be zero, e.g. because the motor stands on a hard stop). This can happen after a Press command, where the actual motor position is defined through external conditions and the motor cannot and had not to follow the demand position. By starting this command, the former accepted difference between Actual Position and Demand Position can be eliminated.

4.3.156 Bestehorn VAJ Increment Actual Pos (0F6xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0F6xh: Bestehorn VAJ Increment Actual Pos	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Jerk	UInt32	1E-4 m/s ³

Bestehorn Motion Profile, regarding the limitations maximal jerk, acceleration and maximal speed. The new Target Position value will be determined by the firmware. It is calculated by adding the Position Increment argument to the Actual Position value (relative move). The command execution starts immediately when the command has been sent.

4.3.157 Bestehorn VAJ Go To Pos After Actual Command (0F8xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0F8xh: Bestehorn VAJ Go To Pos After Actual Command	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Jerk	UInt32	1E-4 m/s ³

Bestehorn Motion Profile, regarding the limitations maximal jerk, acceleration and maximal speed. The command execution is delayed until the former command has been completed. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new VAI motion command as soon as bit 13 of the Status Word (Motion Active) is 0. After the event, the Event Handler deactivates itself.

4.3.158 Bestehorn VAJ Go To Analog Pos (0F9xh)

Name	Byte Offset	Description	Type	Unit
Header	0	0F9xh: Bestehorn VAJ Increment Actual Pos	UInt16	-

Name	Byte Offset	Description	Type	Unit
1. Par	2	Maximal Velocity	UInt32	1E-6 m/s
2. Par	6	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²
3. Par	10	Maximal Jerk	UInt32	1E-4 m/s ³

Bestehorn Motion Profile, regarding the limitations maximal jerk, acceleration and maximal speed. This command sets a new Target Position Calculated from the analog input on X4.4 or X20.

4.3.159 Bestehorn VAJ Go To Pos On Rising Trigger Event (0FAxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0FAxh: Bestehorn VAJ Go To Pos On Rising Trigger Event	UInt16	-
1. Par	2	Target Position	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Jerk	UInt32	1E-4 m/s ³

This command defines a new Bestehorn motion command (see description of command Bestehorn VAJ Go To Pos (0F0xh)). The command will be started on each rising edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal.

The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new Bestehorn motion on the rising trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'.

The trigger input must be configured with parameter UPID 0x1039.

4.3.160 Bestehorn VAJ Increment Demand Pos On Rising Trigger Event (0FBxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0FBxh: Bestehorn VAJ Increment Demand Pos On Rising Trigger Event	UInt16	-
1. Par	2	Position Increment	UInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Jerk	UInt32	1E-4 m/s ³

This command defines a new Bestehorn motion command (relative move, see description of command Bestehorn VAJ Increment Demand Pos (0F1xh)). The command will be started on each rising edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal.

The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new Bestehorn motion on the rising trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'.

The trigger input must be configured with parameter UPID 0x1039.

4.3.161 Bestehorn VAJ Go To Pos On Falling Trigger Event (0FCxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0FCxh: Bestehorn VAJ Go To Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Jerk	UInt32	1E-4 m/s ³

This command defines a new Bestehorn motion command (see description of command Bestehorn VAJ Go To Pos (0F0xh)). The command will be started on each falling edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal.

The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new Bestehorn motion on the falling trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'.

The trigger input must be configured with parameter UPID 0x1039.

4.3.162 Bestehorn VAJ Increment Demand Pos On Falling Trigger Event (0FDxh)

Name	Byte Offset	Description	Type	Unit
Header	0	0FDxh: Bestehorn VAJ Increment Demand Pos On Falling Trigger Event	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration/Deceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Jerk	UInt32	1E-4 m/s ³

This command defines a new Bestehorn motion command (relative move, see description of command Bestehorn VAJ Increment Demand Pos (0F1xh)). The command will be started on each falling edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal.

The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new Bestehorn motion on the falling trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'.

The trigger input must be configured with parameter UPID 0x1039.

4.3.163 Encoder Cam Enable (100xh)

Name	Byte Offset	Description	Type	Unit
Header	0	100xh: Encoder CAM Enable	UInt16	-

Global enable of encoder CAM.

4.3.164 Encoder Cam Disable (101xh)

Name	Byte Offset	Description	Type	Unit
Header	0	101xh: Encoder CAM Disable	UInt16	-

Global disable of encoder CAM.

4.3.165 Encoder Cam Go To Sync Pos (102xh)

Name	Byte Offset	Description	Type	Unit
Header	0	102xh: Encoder CAM Go To Sync Pos	UInt16	-

Go to specified encoder CAM synchronous position. Used for synchronization with standing encoder.

4.3.166 Encoder Cam Set Value (104xh)

Name	Byte Offset	Description	Type	Unit
Header	0	104xh: Encoder CAM Set Value	UInt16	-
1. Par	2	Counter Value	SInt32	1 Incr.

Set encoder CAM value to specified value.

4.3.167 Encoder Cam y Define Curve With Default Parameters (1y0xh)

At the moment two cams can be defined. In the following y=1 stands for cam 1 and y=2 stands for cam 2.

Name	Byte Offset	Description	Type	Unit
Header	0	1y0xh: Encoder Cam y Define Curve With Default Parameters (y=1..2)	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Count	SInt32	1 Incr.

Define Encoder CAM 1 curve with default parameters.

4.3.168 Encoder Cam y Define Curve From Act Pos (1y1xh)

Name	Byte Offset	Description	Type	Unit
Header	0	1y1xh: Encoder Cam y Define Curve From Act Pos (y=1..2)	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Count	SInt32	1 Incr.

If the encoder value is in the range of cam y, the motor has to be at the start position of the cam y during the command setup, to set the position start point of cam y correctly.

4.3.169 Encoder Cam y Define Curve To Pos (1y2xh)

Name	Byte Offset	Description	Type	Unit
Header	0	1y2xh: Encoder Cam y Define Curve To Pos (y=1..2)	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Count	SInt32	1 Incr
3. Par	8	Target Position	SInt32	0.1 um

If the encoder value is in the range of cam y, the motor has to be at the start position of the cam y during command setup, to set the position start point correctly of cam y.

4.3.170 Encoder Cam y Define Curve From Pos To Pos In Counts (1y3xh)

Name	Byte Offset	Description	Type	Unit
Header	0	1y3xh: Encoder Cam y Define Curve To Pos (y=1..2)	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Count	SInt32	1 Incr
3. Par	8	Start Position	SInt32	0.1 um
4. Par	12	Target Position	SInt32	0.1 um
5. Par	16	CAM Length In Counts	SInt32	1 Incr

Define Encoder CAM 1 curve start from actual position and go to specified position.

4.3.171 Encoder Cam y Define Curve To Pos In Counts (1y4xh)

Name	Byte Offset	Description	Type	Unit
Header	0	1y4xh: Encoder Cam y Define Curve To Pos (y=1..2)	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Count	SInt32	1 Incr
3. Par	8	Target Position	SInt32	0.1 um
4. Par	12	CAM Length In Counts	SInt32	1 Incr

If the encoder value is in the range of cam y, the motor has to be at the start position of the cam y during command setup, to set the position start point correctly of cam y.

4.3.172 Encoder Cam y Define Curve With Amplitude Scale In Counts (1y5xh)

Name	Byte Offset	Description	Type	Unit
Header	0	1y5xh: Encoder CAM y Define Curve To Pos (y=1..2)	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Curve Start Count	SInt32	1 Incr
3. Par	8	Amplitude Scale	SInt16	0.1%

Name	Byte Offset	Description	Type	Unit
4. Par	10	CAM Length In Counts	Sint32	1 Incr.

The values for the amplitude scale and cam length are updated immediately, for this reason this command should only be used outside the cam y (old and new) definition. For changing the amplitude scale and/or length during cam y is active, use the command 1y8xh instead.

4.3.173 Encoder Cam y Enable (1y6xh)

Name	Byte Offset	Description	Type	Unit
Header	0	1y6xh Encoder Cam y Enable; (y=1..2)	UInt16	-

Enable Encoder CAM 1.

4.3.174 Encoder Cam y Disable (1y7xh)

Name	Byte Offset	Description	Type	Unit
Header	0	1y7xh Encoder Cam y Disable; (y=1..2)	UInt16	-

Disable Encoder CAM 1.

4.3.175 Encoder Cam y Change Amplitude Scale and Length (1y8xh)

Name	Byte Offset	Description	Type	Unit
Header	0	1y8xh: Encoder Cam y Define Curve To Pos (y=1..2)	UInt16	-
1. Par	2	Amplitude Scale	Sint16	0.1%
2. Par	4	CAM Length In Counts	Sint32	1 Incr.

The new values for the amplitude scale and cam length are updated at next cam y start event.

4.3.176 Start Command Table Command (200xh)

Name	Byte Offset	Description	Type	Unit
Header	0	200xh: Start Command Table Command	UInt16	-
1. Par	2	Command Table ID	UInt16	1..255

This command starts a command table instruction, defined with the Command Table ID.

4.3.177 Start Command Table Command On Rising Trigger Event (201xh)

Name	Byte Offset	Description	Type	Unit
Header	0	201xh: Start Command Table Command On Rising Trigger Event	UInt16	-
1. Par	2	Command Table ID	UInt16	1..255

This command starts a command table instruction on a rising trigger event.

4.3.178 Start Command Table Command On Falling Trigger Event (202xh)

Name	Byte Offset	Description	Type	Unit
Header	0	202xh: Start Command Table Command On Falling Trigger Event	UInt16	-
1. Par	2	Command Table ID	UInt16	1..255

This command starts a command table instruction on a falling trigger event.

4.3.179 Modify Command Table 16 bit Parameter in RAM (208xh)

Name	Byte Offset	Description	Type	Unit
Header	0	208xh: Modify Command Table 16 bit Parameter in RAM	UInt16	-
1. Par	2	Command Table ID	UInt16	1..255
2. Par	4	Parameter Offset	UInt16	0..3Eh
3. Par	6	Parameter Value	Sint16	-

This command modifies a single 16 bit parameter of the specified command table entry with the specified offset to specified value. Within the command table entry the link ID has offset = 2 the motion command header has offset = 4 and the first motion command parameter has offset = 6.

4.3.180 Modify Command Table 32 bit Parameter in RAM (209xh)

Name	Byte Offset	Description	Type	Unit
Header	0	209xh: Modify Command Table 32 bit Parameter in RAM	UInt16	-
1. Par	2	Command Table ID	UInt16	1..255
2. Par	4	Parameter Offset	UInt16	0..3Eh
3. Par	6	Parameter Value	Sint32	-

This command modifies a single 32 bit parameter of the specified command table entry with the specified offset to specified value. Within the command table entry the link ID has offset = 2 the motion command header has offset = 4 and the first motion command parameter has offset = 6.

4.3.181 Wait Time (210xh)

Name	Byte Offset	Description	Type	Unit
Header	0	210xh: Wait Time	UInt16	-
1. Par	2	Time	UInt32	100us

Can be used in a linked command table sequence.

4.3.182 Wait Until Motion Finished (211xh)

Name	Byte Offset	Description	Type	Unit
Header	0	211xh: Wait Until Motion Finished	UInt16	-

Can be used in a linked command table sequence.

4.3.183 Wait Until In Target Position (212xh)

Name	Byte Offset	Description	Type	Unit
Header	0	212xh: Wait Until Motion Finished	UInt16	-

Can be used in a linked command table sequence.

4.3.184 Wait Until Rising Trigger Event (213xh)

Name	Byte Offset	Description	Type	Unit
Header	0	213xh: Wait Until Rising Trigger	UInt16	-

Can be used in a linked command table sequence.

4.3.185 Wait Until Falling Trigger Event (214xh)

Name	Byte Offset	Description	Type	Unit
Header	0	214xh: Wait Until Falling Trigger	UInt16	-

Can be used in a linked command table sequence.

4.3.186 Wait Until Demand Position Greater Than (220xh)

Name	Byte Offset	Description	Type	Unit
Header	0	220xh: Wait Until Demand Position Greater Than	UInt16	-
1. Par	2	Dem Pos Trig Level	SInt32	0.1 um

Can be used in a linked command table sequence.

4.3.187 Wait Until Demand Position Less Than (221xh)

Name	Byte Offset	Description	Type	Unit
Header	0	221xh: Wait Until Demand Position Less Than	UInt16	-
1. Par	2	Dem Pos Trig Level	SInt32	0.1 um

Can be used in a linked command table sequence.

4.3.188 Wait Until Actual Position Greater Than (222xh)

Name	Byte Offset	Description	Type	Unit
Header	0	222xh: Wait Until Actual Position Greater Than	UInt16	-
1. Par	2	Act Pos Trig Level	SInt32	0.1 um

Can be used in a linked command table sequence.

4.3.189 Wait Until Actual Position Less Than (223xh)

Name	Byte Offset	Description	Type	Unit
Header	0	223xh: Wait Until Actual Position Less Than	UInt16	-
1. Par	2	Act Pos Trig Level	SInt32	0.1 um

Can be used in a linked command table sequence.

4.3.190 Wait Until Difference Position Greater Than (224xh)

Name	Byte Offset	Description	Type	Unit
Header	0	224xh: Wait Until Difference Position Greater Than	UInt16	-
1. Par	2	Diff Pos Trig Level	SInt32	0.1 um

Can be used in a linked command table sequence.

4.3.191 Wait Until Difference Position Less Than (225xh)

Name	Byte Offset	Description	Type	Unit
Header	0	225xh: Wait Until Difference Position Less Than	UInt16	-
1. Par	2	Diff Pos Trig Level	SInt32	0.1 um

Can be used in a linked command table sequence.

4.3.192 Wait Until Difference Position Unsigned Greater Than (226xh)

Name	Byte Offset	Description	Type	Unit
Header	0	226xh: Wait Until Difference Position Unsigned Greater Than	UInt16	-
1. Par	2	Diff Pos Trig Level	SInt32	0.1 um

Can be used in a linked command table sequence.

4.3.193 Wait Until Difference Position Unsigned Less Than (227xh)

Name	Byte Offset	Description	Type	Unit
Header	0	227xh: Wait Until Difference Position Unsigned Less Than	UInt16	-
1. Par	2	Diff Pos Trig Level	SInt32	0.1 um

Can be used in a linked command table sequence.

4.3.194 Wait Until Demand Velocity Greater Than (228xh)

Name	Byte Offset	Description	Type	Unit
Header	0	228xh: Wait Until Demand Velocity Greater Than	UInt16	-
1. Par	2	Dem Vel Trig Level	SInt32	1E-6 m/s

Can be used in a linked command table sequence.

4.3.195 Wait Until Demand Velocity Less Than (229xh)

Name	Byte Offset	Description	Type	Unit
Header	0	229xh: Wait Until Demand Velocity Less Than	UInt16	-
1. Par	2	Dem Vel Trig Level	SInt32	1E-6 m/s

Can be used in a linked command table sequence.

4.3.196 Wait Until Actual Velocity Greater Than (22Axh)

Name	Byte Offset	Description	Type	Unit
Header	0	22Axh: Wait Until Actual Velocity Greater Than	UInt16	-
1. Par	2	Act Vel Trig Level	SInt32	1E-6 m/s

Can be used in linked command table sequence.

4.3.197 Wait Until Actual Velocity Less Than (22Bxh)

Name	Byte Offset	Description	Type	Unit
Header	0	22Bxh: Wait Until Actual Velocity Less Than	UInt16	-
1. Par	2	Act Vel Trig Level	SInt32	1E-6 m/s

Can be used in a linked command table sequence.

4.3.198 Wait Until Current Greater Than (22Exh)

Name	Byte Offset	Description	Type	Unit
Header	0	22Exh: Wait Until Current Greater Than	UInt16	-
1. Par	2	Dem Curr Vel Trig Level	Sint16	1mA

Can be used in linked command table sequence.

4.3.199 Wait Until Current Less Than (22Fhx)

Name	Byte Offset	Description	Type	Unit
Header	0	22Fhx: Wait Until Current Less Than	UInt16	-
1. Par	2	Dem Curr Trig Level	Sint16	1mA

Can be used in a linked command table sequence.

4.3.200 Set Cmd Table Var 1 To (240xh)

Name	Byte Offset	Description	Type	Unit
Header	0	240xh: Set Cmd Table Var 1 To	UInt16	-
1. Par	2	Set value	Sint32	-

Set Command Table Var 1 to specified value.

4.3.201 Add To Cmd Table Var 1 (241xh)

Name	Byte Offset	Description	Type	Unit
Header	0	241xh: Add To Cmd Table Var 1	UInt16	-
1. Par	2	Add value	Sint32	-

Add to Command Table Var 1 specified value.

4.3.202 Set Cmd Table Var 2 To (242xh)

Name	Byte Offset	Description	Type	Unit
Header	0	242xh: Set Cmd Table Var 2 To	UInt16	-
1. Par	2	Set value	Sint32	-

Set Command Table Var 2 to specified value.

4.3.203 Add To Cmd Table Var 2 (243xh)

Name	Byte Offset	Description	Type	Unit
Header	0	243xh: Add To Cmd Table Var 2	UInt16	-
1. Par	2	Add value	Sint32	-

Add to Command Table Var 2 specified value.

4.3.204 Write Cmd Table Var 1 To UPID RAM value (248xh)

Name	Byte Offset	Description	Type	Unit
Header	0	248xh: Write Cmd Table Var 1 To UPID RAM value	UInt16	-
1. Par	2	UPID (Unique Parameter ID)	UInt16	-

This command copies the RAM value of Cmd Table Var 1 to the live parameter with the given UPID.

4.3.205 Write Cmd Table Var 2 To UPID RAM value (249xh)

Name	Byte Offset	Description	Type	Unit
Header	0	249xh: Write Cmd Table Var 2 To UPID RAM value	UInt16	-
1. Par	2	UPID (Unique Parameter ID)	UInt16	-

This command copies the RAM value of Cmd Table Var 2 to the live parameter with the given UPID.

4.3.206 Write UPID RAM Value To Cmd Table Var 1 (24Cxh)

Name	Byte Offset	Description	Type	Unit
Header	0	24Cxh: Write UPID RAM value To Cmd Table Var 1	UInt16	-
1. Par	2	UPID (Unique Parameter ID)	UInt16	-

This command copies the RAM value of the live parameter with the given UPID To Cmd Table Var 1.

4.3.207 Write UPID RAM Value To Cmd Table Var 2 (24Dxh)

Name	Byte Offset	Description	Type	Unit
Header	0	24Dxh: Write UPID RAM value To Cmd Table Var 2	UInt16	-
1. Par	2	UPID (Unique Parameter ID)	UInt16	-

This command copies the RAM value of the live parameter with the given UPID To Cmd Table Var 2.

4.3.208 Write UPID RAM value To UPID ROM value (24Exh)

Name	Byte Offset	Description	Type	Unit
Header	0	24Exh: Write UPID RAM value To UPID ROM value	UInt16	-
1. Par	2	UPID (Unique Parameter ID)	UInt16	-

This command copies the RAM value of the live parameter to its ROM value.

4.3.209 IF Cmd Table Var 1 Less Than (250xh)

Name	Byte Offset	Description	Type	Unit
Header	0	250xh: IF Cmd Table Var 1 Less Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Command Table Variable 1 (UPID 0x1E72) is less than the condition value.

4.3.210 IF Cmd Table Var 1 Greater Than (251xh)

Name	Byte Offset	Description	Type	Unit
Header	0	251xh: IF Cmd Table Var 1 Greater Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Command Table Variable 1 (UPID 0x1E72) is greater than the condition value.

4.3.211 IF Cmd Table Var 2 Less Than (252xh)

Name	Byte Offset	Description	Type	Unit
Header	0	252xh: IF Cmd Table Var 2 Less Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Command Table Variable 2 (UPID 0x1E73) is less than the condition value.

4.3.212 IF Cmd Table Var 2 Greater Than (253xh)

Name	Byte Offset	Description	Type	Unit
Header	0	253xh: IF Cmd Table Var 2 Greater Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Command Table Variable 2 (UPID 0x1E73) is greater than the condition value.

4.3.213 IF Cmd Table Var 1 Less Than UPID Value (256xh)

Name	Byte Offset	Description	Type	Unit
Header	0	250xh: IF Cmd Table Var 1 Less Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Command Table Variable 1 (UPID 0x1E72) is less than the value of the variable defined by the UPID.

4.3.214 IF Cmd Table Var 2 Less Than UPID Value (257xh)

Name	Byte Offset	Description	Type	Unit
Header	0	250xh: IF Cmd Table Var 1 Less Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Command Table Variable 2 (UPID 0x1E73) is less than the value of the variable defined by the UPID.

4.3.215 IF Demand Position Less Than (258xh)

Name	Byte Offset	Description	Type	Unit
Header	0	258xh: IF Demand Position Less Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Demand Position (UPID 0x1B8A) is less than the condition value.

4.3.216 IF Demand Position Greater Than (259xh)

Name	Byte Offset	Description	Type	Unit
Header	0	259xh: IF Demand Position Greater Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Demand Position (UPID 0x1B8A) is greater than the condition value.

4.3.217 IF Actual Position Less Than (25Axh)

Name	Byte Offset	Description	Type	Unit
Header	0	25Axh: IF Actual Position Less Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Actual Position (UPID 0x1B8D) is less than the condition value.

4.3.218 IF Actual Position Greater Than (25Bxh)

Name	Byte Offset	Description	Type	Unit
Header	0	25Bxh: IF Actual Position Greater Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Actual Position (UPID 0x1B8D) is greater than the condition value.

4.3.219 IF Difference Position Less Than (25Cxh)

Name	Byte Offset	Description	Type	Unit
Header	0	25Cxh: IF Difference Position Less Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Difference Position (UPID 0x1B90) is less than the condition value.

4.3.220 IF Difference Position Greater Than (25Dxh)

Name	Byte Offset	Description	Type	Unit
Header	0	25Dxh: IF Difference Position Greater Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Difference Position (UPID 0x1B90) is greater than the condition value.

4.3.221 IF Current Less Than (25Exh)

Name	Byte Offset	Description	Type	Unit
Header	0	25Cxh: IF Current Less Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Demand Current (UPID 0x1B93) is less than the condition value.

4.3.222 IF Current Greater Than (25Fxh)

Name	Byte Offset	Description	Type	Unit
Header	0	25Fxh: IF Current Greater Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if Demand Current (UPID 0x1B93) is greater than the condition value.

4.3.223 IF Analog Val On X4.4 Less Than (260xh)

Name	Byte Offset	Description	Type	Unit
Header	0	260xh: IF Analog Val On X4.4 Less Than	UInt16	-
1. Par	2	Condition Value	Sint32	-
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if analog value on X4.4 (UPID 0x1CA4) is less than the condition value.

4.3.224 IF Masked X4 Input Value Equal Than (262xh)

Name	Byte Offset	Description	Type	Unit
Header	0	262xh: IF Masked X4 Input Value Equal Than	UInt16	-
1. Par	2	X4 Bit Mask	UInt16	
2. Par	4	Condition Value	UInt16	-
3. Par	6	Command Table ID IF TRUE	UInt16	1..255
4. Par	8	Command Table ID IF FALSE	UInt16	1..255

According to the masked input pattern this command executes a branch. With the X4 Bit Mask the relevant inputs are filtered and then compared with the X4 Compare Value. An input is relevant if the corresponding bit in the Bit Mask is set. The logical comparison is: IF((X4_Inputs & X4_Bit_Mask) = X4_Compare_Value) THEN ... Bit 0 corresponds with X4.3, bit 8 with X4.11.

4.3.225 IF Masked X6 Input Value Equal Than (263xh)

Name	Byte Offset	Description	Type	Unit
Header	0	263xh: IF Masked X6 Input Value Equal Than	UInt16	-
1. Par	2	X6 Bit Mask	UInt16	
2. Par	4	Condition Value	UInt16	-
3. Par	6	Command Table ID IF TRUE	UInt16	1..255
4. Par	8	Command Table ID IF FALSE	UInt16	1..255

According to the masked input pattern this command executes a branch. With the X6 Bit Mask the relevant inputs are filtered and then compared with the X6 Compare Value. An input is relevant if the corresponding bit in the Bit Mask is set. The logical comparison is: IF((X6_Inputs & X6_Bit_Mask) = X6_Compare_Value) THEN ... Bit 0 corresponds with X6.1, bit 7 with X6.17.

4.3.226 IF Masked Status Word Equal Than (264xh)

Name	Byte Offset	Description	Type	Unit
Header	0	264xh IF Masked Status Word Equal Than	UInt16	-
1. Par	2	Status Word Bit Mask	UInt16	
2. Par	4	Condition Value	UInt16	-
3. Par	6	Command Table ID IF TRUE	UInt16	1..255
4. Par	8	Command Table ID IF FALSE	UInt16	1..255

According to the value of the state word a branch is executed. The relevant bits are represented in the Bit Mask with 1. The logical comparison is: IF((Status_Word & Status_Word_Bit_Mask) = Status_Word_Compare_Value) THEN...

4.3.227 IF Masked Warn Word Equal Than (265xh)

Name	Byte Offset	Description	Type	Unit
Header	0	265xh IF Masked Warn Word Equal Than	UInt16	-
1. Par	2	Warn Word Bit Mask	UInt16	
2. Par	4	Condition Value	UInt16	-
3. Par	6	Command Table ID IF TRUE	UInt16	1..255
4. Par	8	Command Table ID IF FALSE	UInt16	1..255

According to the value of the warn word a branch is executed. The relevant bits are represented in the Bit Mask with 1. The logical comparison is: IF((Warn_Word & Warn_Word_Bit_Mask) = Warn_Word_Compare_Value) THEN....

4.3.228 IF CAM Counts Less Than (266xh)

Name	Byte Offset	Description	Type	Unit
Header	0	256xh: IF CAM Counts Less Than	UInt16	-
1. Par	2	Condition Value	Sint32	Counts
2. Par	6	Command Table ID IF TRUE	UInt16	1..255
3. Par	8	Command Table ID IF FALSE	UInt16	1..255

Command Table branch condition: if CAM Counts (UPID 0x1C50) is less than the condition value.

4.3.229 Encoder Winding Start With Default Parameters (300xh)

Name	Byte Offset	Description	Type	Unit
Header	0	300xh: Encoder Winding Start With Default Parameters	UInt16	-

Start Encoder Winding with default Parameter.

4.3.230 Encoder Winding Start With Default Parameters At Revolutions (301xh)

Name	Byte Offset	Description	Type	Unit
Header	0	301xh: Encoder Winding Start With Default Parameters At Revolutions	UInt16	-
1. Par	2	Revolution Counts To Start	Sint32	Rev

Restart Encoder Winding with default Parameter at revolutions.

4.3.231 Encoder Winding Stop Adaptation Of Left/Right Position and Disturbance (304xh)

Name	Byte Offset	Description	Type	Unit
Header	0	304xh: Encoder Winding Stop Adaptation Of Left/Right Position And Disturbance.	UInt16	-

Stop adaptation of the disturbance.

4.3.232 Encoder Winding Restart Adaptation Of Left/Right Position and Disturbance (305xh)

Name	Byte Offset	Description	Type	Unit
Header	0	305xh: Encoder Winding Restart Adaptation Of Left/Right Position And Disturbance	UInt16	-

Restart adaptation of disturbance.

4.3.233 Encoder Curve Winding Start With Default Parameters (310xh)

Name	Byte Offset	Description	Type	Unit
Header	0	310xh: Encoder Curve Winding Start With Default Parameters	UInt16	-
1. Par	2	Curve ID	UInt16	1..100

Start Encoder Curve Winding with default Parameter.

4.3.234 Encoder Curve Winding Start With Default Parameters At Revolutions (311xh)

Name	Byte Offset	Description	Type	Unit
Header	0	311xh: Encoder Curve Winding Start With Default Parameters At Revolutions	UInt16	-
1. Par	2	Curve ID	UInt16	1..100
2. Par	4	Revolution Counts To Start	SInt32	1 Rev

Restart Encoder Curve Winding with default Parameter at revolutions.

4.3.235 VAI Go To Pos With Higher Force Ctrl Limit (380xh)

Name	Byte Offset	Description	Type	Unit
Header	0	380xh: VAI Go To Pos With Higher Force Ctrl Limit	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s2
4. Par	14	Force Limit	Sint16	0.1 N

Moves to the defined target position, if the measured force reaches the higher defined value the drive switches to the force control mode with target force = force limit.

To change back to position control mode use motion command **VAI Go To Pos From Act Pos And Reset Force Control (381xh)**.

4.3.236 VAI Go To Pos From Act Pos And Reset Force Control (381xh)

Name	Byte Offset	Description	Type	Unit
Header	0	381xh: VAI Go To Pos From Act Pos And Reset Force Control	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

Reinstalls the position control mode and moves to the defined target position.

4.3.237 Force Ctrl Change Target Force (382xh)

Name	Byte Offset	Description	Type	Unit
Header	0	382xh: Force Ctrl Change Target Force	UInt16	-
1. Par	2	Target Force	Sint16	0.1 N

This command can be used to change the target force during the force control mode. This command activates the force control mode if not active.

4.3.238 VAI Go To Pos With Higher Force Ctrl Limit and Target Force (383xh)

Name	Byte Offset	Description	Type	Unit
Header	0	383xh: VAI Go To Pos With Higher Force Ctrl Limit and Target Force	UInt16	-
1. Par	2	Target Position	Sint32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Force Limit	Sint16	0.1 N
5. Par	16	Target Force	Sint16	0.1 N

Moves to the defined target position, if the measured force reaches the higher force limit the drive switches to the force control mode with target force = target force.

To change back to position control mode use motion command VAI Go To Pos From Act Pos And Reset Force Control (381xh).

4.3.239 VAI Go To Pos With Lower Force Ctrl Limit (384xh)

Name	Byte Offset	Description	Type	Unit
Header	0	384xh: VAI Go To Pos With Lower Force Ctrl Limit	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Force Limit	Sint16	0.1 N

Moves to the defined target position, if the measured force reaches the lower defined value the drive switches to the force control mode with target force = force limit.

To change back to position control mode use motion command **VAI Go To Pos From Act Pos And Reset Force Control (381xh)**.

4.3.240 VAI Go To Pos With Lower Force Ctrl Limit and Target Force (385xh)

Name	Byte Offset	Description	Type	Unit
Header	0	385xh: VAI Go To Pos With Lower Force Ctrl Limit and Target Force	UInt16	-
1. Par	2	Target Position	Sint32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Force Limit	Sint16	0.1 N
5. Par	16	Target Force	Sint16	0.1 N

Moves to the defined target position, if the measured force reaches the lower force limit the drive switches to the force control mode with target force = target force.

To change back to position control mode use motion command **VAI Go To Pos From Act Pos And Reset Force Control (381xh)**.

4.3.241 VAI Go To Pos From Act Pos And Reset Force Control Set I (386xh)

Name	Byte Offset	Description	Type	Unit
Header	0	386xh: VAI Go To Pos From Act Pos And Reset Force Control Set I	UInt16	-
1. Par	2	Target Position	SInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

Reinstalls the position control mode and moves to the defined target position. The I part of the position controller is set to the last force control current.

4.3.242 VAI Increment Act Pos And Reset Force Control Set I (387xh)

Name	Byte Offset	Description	Type	Unit
Header	0	387xh: VAI Increment Act Pos And Reset Force Control Set I	UInt16	-
1. Par	2	Target Position Increment	SInt32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

Reinstalls the position control mode and moves to the incremented target position. The I part of the position controller is set to the last force control current.

4.3.243 VAI Inc Act Pos With Higher Force Ctrl Limit and Target Force (388xh)

Name	Byte Offset	Description	Type	Unit
Header	0	388xh: VAI Inc Act Pos With Higher Force Ctrl Limit and Target Force	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Force Limit	Sint16	0.1 N
5. Par	16	Target Force	Sint16	0.1 N

VAI increment actual position, if the measured force reaches the defined value, the drive switches to the force control mode with Target Force = Target Force. To change back to position control mode use motion command VAI Go To Pos From Act Pos And Reset Force Control (381xh)!.

4.3.244 VAI Inc Act Pos With Lower Force Ctrl Limit and Target Force (389xh)

Name	Byte Offset	Description	Type	Unit
Header	0	389xh: VAI Inc Act Pos With Lower Force Ctrl Limit and Target Force	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Force Limit	Sint16	0.1 N
5. Par	16	Target Force	Sint16	0.1 N

VAI increment actual position, if the measured force reaches the defined value, the drive switches to the force control mode with Target Force = Target Force. To change back to position control mode use motion command VAI Go To Pos From Act Pos And Reset Force Control (381xh)!.

4.3.245 VAI Go To Pos Tara Diff Analog In -10V..10V (38Axh)

Name	Byte Offset	Description	Type	Unit
Header	0	38Axh: VAI Go To Pos Tara Diff Analog In -10V..10V	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²
5. Par	18	Time Delay	UInt16	ms
6. Par	20	Capture Time	UInt16	ms

This command sets a new Target Position and defines the maximal velocity, acceleration and deceleration for the movement. The command execution starts immediately when the command has been sent. The setpoints (Demand Position, Demand Velocity and Demand Acceleration) are calculated by the internal Velocity

Acceleration Interpolator (VAI). This command initializes the VAI with the current Demand Position and Demand Velocity value. Therefore it is possible to start a new command while a former command is still being executed.

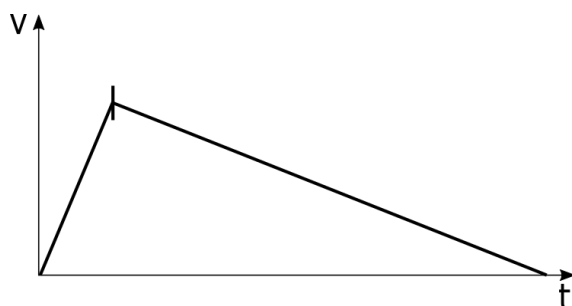
This command additionally captures the *Measured Force* (UPID 0x1EA1) while the *Demand Velocity* is in maximal velocity state and stores its average as *0V Raw Data Offset* (UPID 0x1798).

The measurement of the *Measured Force* can be delayed by the parameter *Time Delay* and is captured during the time *Capture Time*.

Bit 9 in Status Word (Special Motion Active) indicates if tara calculation is active.

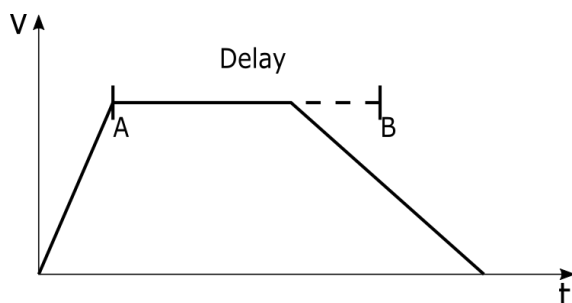
Depending on the parameter values of *Time Delay* and *Capture Time* four different cases are possible. To get the best result, try to choose your parameter values as described in [Case 4](#).

Case 1: Movement does not reach maximal velocity state.



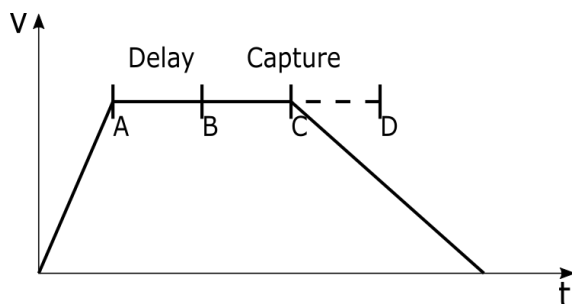
As there is no maximal velocity state, the *Measured Force* is not captured and therefore the *0V Raw Data Offset* does not change.

Case 2: *Time Delay* is longer than maximal velocity state.



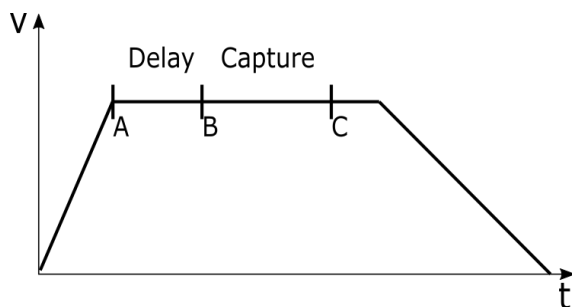
As the parametrized delay is too long, the *Measured Force* is not captured and therefore the *0V Raw Data Offset* does not change.

Case 3: *Time Delay* is shorter than maximal velocity state and *Time Delay + Capture Time* is longer than maximal velocity state.



The *Measured Force* is measured from B to C and the new *0V Raw Data Offset* is calculated at the end of the motion.

Case 4: *Time Delay* is shorter than maximal velocity state and *Time Delay + Capture Time* is shorter than maximal velocity state.



The *Measured Force* is measured from B to C and the new *0V Raw Data Offset* is calculated at C.

Error: If *Capture Time* is set to 0, the error 0x5E (Cfg Err: Par Range Error) is set.

4.3.246 Standstill Tara Diff Analog In -10V..10V (38Bxh)

Name	Byte Offset	Description	Type	Unit
Header	0	38Bxh: Standstill Tara Analog In -10V..10V	UInt16	-
1. Par	2	Capture Time	UInt16	ms

This command captures the Measured Force (UPID 0x1EA1) over the parametrized "Capture Time" and stores its average as "0V Raw Data Offset" (UPID 0x1798).

Bit 9 in Status Word (Special Motion Active) indicates if tara calculation is active.

Error: If "Capture Time" is set to 0, the error 0x5E (Cfg Err: Par Range Error) is set.

4.3.247 Current Command Mode (390xh)

Name	Byte Offset	Description	Type	Unit
Header	0	390xh: Current Command Mode	UInt16	-
1. Par	2	Demand Current	Sint32	0.001 A

Change in Current Command mode if not yet and set demand current.

4.3.248 Closed Loop Force Command Mode Rel (398xh)

Name	Byte Offset	Description	Type	Unit
Header	0	398xh: Current Command Mode	UInt16	-
1. Par	2	Target Force Relative	Sint16	0.002%

Change to the closed loop force control mode and set the target force. The value 5000 is 500% which is the maximal force/torque value.

4.3.249 Closed Loop Force Command Mode Abs (399xh)

Name	Byte Offset	Description	Type	Unit
Header	0	398xh: Current Command Mode	UInt16	-
1. Par	2	Target Force Absolut	Sint16	0.1N

Name	Byte Offset	Description	Type	Unit
------	-------------	-------------	------	------

Change to the closed loop force control mode and set the target force. The value of motion parameter 1 is directly written to the Target Force (UPID 1EA0h).

4.3.250 Change to Position Controlled Mode (39Fxxh)

Name	Byte Offset	Description	Type	Unit
Header	0	39Fxxh: Change to Position Controlled Mode	UInt16	-

Go back to position controlled mode.

4.3.251 VAJI Go To Pos (3A0xxh)

Name	Byte Offset	Description	Type	Unit
Header	0	3A0xxh: VAJI Go To Pos	UInt16	-
1. Par	2	Target Position	Sint32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Deceleration	UInt32	1E-5 m/s ²
5. Par	18	Jerk	UInt32	1E-4 m/s ³

This command sets a new Target Position and defines the maximal velocity, maximal acceleration, maximal deceleration, and jerk for the movement. The command execution starts one cycle after the command has been sent. The setpoints (Demand Position, Demand Velocity, Demand Acceleration, and Demand Jerk) are calculated by the internal Velocity Acceleration Jerk Interpolator (VAJI). This command initializes the VAJI with the current Demand Position, Demand Velocity, and Demand Acceleration value. Therefore it is possible to start a new command while a former command is still being executed.

4.3.252 VAJI Increment Dem Pos (3A1xxh)

Name	Byte Offset	Description	Type	Unit
Header	0	3A1xxh:VAJI Increment Dem Pos	UInt16	-
1. Par	2	Position Increment	Sint32	0.1 μ m
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Deceleration	UInt32	1E-5 m/s ²
5. Par	18	Jerk	UInt32	1E-4 m/s ³

This command sets a new Target Position and defines the maximal velocity, maximal acceleration, maximal deceleration, and jerk for the movement. The new Target Position value will be determined by the firmware. It is calculated by adding the Position Increment argument to the Demand Position value (relative move). The command execution starts one cycle after the command has been sent. The setpoints (Demand Position, Demand Velocity, Demand Acceleration, and Demand Jerk) are calculated by the internal Velocity Acceleration Jerk Interpolator (VAJI). This command initializes the VAJI with the current Demand Position,

Demand Velocity, and Demand Acceleration value. Therefore it is possible to start a new command while the execution of a former command is still being executed.

4.3.253 VAJI Increment Target Pos (3A2xh)

Name	Byte Offset	Description	Type	Unit
Header	0	3A2xh: VAJI Increment Target Pos	UInt16	-
1. Par	2	Position Increment	SIInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Deceleration	UInt32	1E-5 m/s ²
5. Par	18	Jerk	UInt32	1E-4 m/s ³

This command sets a new Target Position and defines the maximal velocity, maximal acceleration, maximal deceleration, and jerk for going there. The new Target Position value will be determined by the firmware. It is calculated by adding the Position Increment argument to the current VAJI Target Position value (relative move). The command execution starts one cycle after the command has been sent. The setpoints (Demand Position, Demand Velocity, Demand Acceleration, and Demand Jerk) are calculated by the internal Velocity Acceleration Jerk Interpolator (VAJI). This command initializes the VAJI with the current Demand Position, Demand Velocity, and Demand Acceleration value. Therefore it is possible to start a new command while the execution of a former command is still being executed.

This command should be used only after the execution of other VAI/VAJI commands (not directly after time curves, encoder curves, streaming commands, etc.). Otherwise the VAJI Target Position value is not correctly initialized, which may lead to unpredictable behavior.

4.3.254 VAJI Go To Pos From Act Pos Starting With Dem Vel = 0 and Dem Acc = 0 (3A4xh)

Name	Byte Offset	Description	Type	Unit
Header	0	3A4xh: VAJI Go To Pos From Act Pos Starting With Dem Vel =0 and Dem Acc = 0	UInt16	-
1. Par	2	Target Position	SIInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Deceleration	UInt32	1E-5 m/s ²
5. Par	18	Jerk	UInt32	1E-4 m/s ³

This command sets a new Target Position and defines the maximal velocity, maximal acceleration, maximal deceleration, and jerk for going there. The command execution starts one cycle after the command has been sent. The setpoints (Demand Position, Demand Velocity, Demand Acceleration, and Demand Jerk) are calculated by the internal Velocity Acceleration Jerk Interpolator (VAJI). This command initializes the VAJI with the Actual Position (Demand Position = Actual Position) and sets the Demand Velocity and Demand Acceleration values to zero.

This command should be used if the Actual Position does not match with the current Demand Position value, but it can be assumed that the motor stands still (Actual Velocity and Actual Acceleration assumed to be zero, e.g. because the motor stands on a hard stop). This can happen after a Press command, where the actual motor position is defined through external conditions and the motor could not and had not to follow the demand position. By starting this command, the former accepted difference between Actual Position and Demand Position can be eliminated.

4.3.255 VAJI Increment Act Pos Starting With Dem Vel = 0 and Dem Acc = 0 (3A6xh)

Name	Byte Offset	Description	Type	Unit
Header	0	3A6xh: VAI Increment Act Pos Starting With Dem Vel = 0 and Dem Acc = 0	UInt16	-
1. Par	2	Position Increment	SLint32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Deceleration	UInt32	1E-5 m/s ²
5. Par	18	Jerk	UInt32	1E-4 m/s ³

This command sets a new Target Position and defines the maximal velocity, maximal acceleration, maximal deceleration, and jerk for the movement. The new Target Position value will be determined by the firmware. It is calculated by adding the Position Increment argument to the Actual Position value (relative move). The command execution starts one cycle after the command has been sent. The setpoints (Demand Position, Demand Velocity, Demand Acceleration, and Demand Jerk) are calculated by the internal Velocity Acceleration Jerk Interpolator (VAJI). This command initializes the VAJI with the Actual Position (Demand Position = Actual Position) and sets the Demand Velocity and Demand Acceleration values to zero. This command should be used if the Actual Position does not match with the current Demand Position value, but it can be assumed that the motor stands still (Actual Velocity and Actual Acceleration assumed to be zero, e.g. because the motor stands on a hard stop). This can happen after a Press command, where the actual motor position is defined through external conditions and the motor could not and had not to follow the demand position. By starting this command, the former accepted difference between Actual Position and Demand Position can be eliminated. With this command a retraction move for a certain distance (command argument Position Increment) can be performed. If the Position Increment value is zero, this command can be used to reset the Target Position (and the Demand Position) value exactly to the Actual Position.

4.3.256 VAJI Go To Pos On Rising Trigger Event (3AAxh)

Name	Byte Offset	Description	Type	Unit
Header	0	3AAxh: VAJI Go To Pos On Rising Trigger Event	UInt16	-
1. Par	2	Target Position	SLint32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Deceleration	UInt32	1E-5 m/s ²
5. Par	18	Jerk	UInt32	1E-4 m/s ³

This command defines a new VAJI motion command (see description of command 'VAJI Go To Pos (3A10xh)'). The command will be started one cycle after each rising edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new VAJI motion on the rising trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'. The trigger input must be configured with parameter UPID 0x1039.

4.3.257 VAJI Increment Target Pos On Rising Trigger Event (3ABxh)

Name	Byte Offset	Description	Type	Unit
Header	0	3ABxh: VAJI Increment Target Pos On Rising Trigger Event	UInt16	-
1. Par	2	Position Increment	UInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Deceleration	UInt32	1E-5 m/s ²
5. Par	18	Jerk	UInt32	1E-4 m/s ³

This command defines a new VAJI motion command (relative move, see description of command 'VAJI Increment Target Pos (3A2xh)'). The command will be started one cycle after each rising edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new VAJI motion on the rising trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'.

The trigger input must be configured with parameter UPID 0x1039.

4.3.258 VAJI Go To Pos On Falling Trigger Event (3ACxh)

Name	Byte Offset	Description	Type	Unit
Header	0	3ACxh: VAJI Go To Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	UInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Deceleration	UInt32	1E-5 m/s ²
5. Par	18	Jerk	UInt32	1E-4 m/s ³

This command defines a new VAJI motion command (see description of command 'VAJI Go To Pos (3A0xh)'). The command will be started one cycle after each falling edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal.

The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new VAJI motion on the falling trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'.

The trigger input must be configured with parameter UPID 0x1039.

4.3.259 VAJI Increment Target Pos On Falling Trigger Event (3ADxh)

Name	Byte Offset	Description	Type	Unit
Header	0	3ADxh: VAJI Increment Target Pos On Falling Trigger Event	UInt16	-
1. Par	2	Position Increment	UInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Maximal Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Maximal Deceleration	UInt32	1E-5 m/s ²
5. Par	18	Jerk	UInt32	1E-4 m/s ³

This command defines a new VAJI motion command (relative move, see description of command 'VAJI Increment Target Pos (3A2xh)'). The command will be started one cycle after each falling edge on the Trigger signal. This command is used to synchronize the execution of the motion command to a hardware signal. The command activates the Event Handler (see Status Word bit 8). The Event Handler starts the new VAJI motion on the falling trigger event. The Event Handler can be deactivated by using the command 'Clear Event Evaluation (008xh)'.

The trigger input must be configured with parameter UPID 0x1039.

4.3.260 Predef VAJI Go To Pos (3B0xh)

Name	Byte Offset	Description	Type	Unit
Header	0	3B0xh: Predef VAJI Go To Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command works like the command 'VAJI Go To Pos (3A0xh)'. The VAJI parameters (Maximal Velocity, Maximal Acceleration, Maximal Deceleration, and Jerk) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF, UPID 0x14C0, and UPID 0x14C1.

4.3.261 Predef VAJI Increment Dem Pos (3B1xh)

Name	Byte Offset	Description	Type	Unit
Header	0	3B1xh: Predef VAJI Increment Dem Pos	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um

This command works like the command 'VAJI Increment Dem Pos (3A1xh)'. The VAJI parameters (Maximal Velocity, Maximal Acceleration, Maximal Deceleration, and Jerk) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF, UPID 0x14C0, and UPID 0x14C1.

4.3.262 Predef VAJI Increment Target Pos (3B2xh)

Name	Byte Offset	Description	Type	Unit
Header	0	3B2xh: Predef VAJI Increment Target Pos	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um

This command works like the command 'VAJI Increment Target Pos (3A2xh)'. The VAJI parameters (Maximal Velocity, Maximal Acceleration, Maximal Deceleration, and Jerk) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF, UPID 0x14C0, and UPID 0x14C1.

4.3.263 Predef VAJI Go To Pos From Act Pos Starting With Dem Vel = 0 and Dem Acc = 0 (3B4xh)

Name	Byte Offset	Description	Type	Unit
Header	0	3B4xh: Predef VAJI Go To Pos From Act Pos With Dem Vel = 0 and Dem Acc = 0	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command works like the command 'VAJI Go To Pos From Act Pos Starting with Dem Vel = 0 and Dem Acc = 0 (3A4xh)'. The VAJI parameters (Maximal Velocity, Maximal Acceleration, Maximal Deceleration, and

Jerk) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF, UPID 0x14C0, and UPID 0x14C1.

4.3.264 Predef VAJI Go To Pos On Rising Trigger Event (3BAxh)

Name	Byte Offset	Description	Type	Unit
Header	0	3BAxh: Predef VAJI Go To Pos On Rising Trigger Event	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command works like the command 'VAJI Go To Pos On Rising Trigger Event (3AAxh)'. The VAJI parameters (Maximal Velocity, Maximal Acceleration, Maximal Deceleration, and Jerk) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF, UPID 0x14C0, and UPID 0x14C1.

4.3.265 Predef VAJI Increment Target Pos On Rising Trigger Event (3BBxh)

Name	Byte Offset	Description	Type	Unit
Header	0	3BBxh: Predef VAJI Increment Target Pos On Rising Trigger Event	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um

This command works like the command 'VAJI Increment Target Pos On Rising Trigger Event (3ABxh)'. The VAJI parameters (Maximal Velocity, Maximal Acceleration, Maximal Deceleration, and Jerk) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF, UPID 0x14C0, and UPID 0x14C1.

4.3.266 Predef VAJI Go To Pos On Falling Trigger Event (3BCxh)

Name	Byte Offset	Description	Type	Unit
Header	0	3BCxh: Predef VAJI Go To Pos On Falling Trigger Event	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um

This command works like the command 'VAJI Go To Pos On Falling Trigger Event (3ACxh)'. The VAJI parameters (Maximal Velocity, Maximal Acceleration, Maximal Deceleration, and Jerk) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF, UPID 0x14C0, and UPID 0x14C1.

4.3.267 Predef VAJI Increment Target Pos On Falling Trigger Event (3BDxh)

Name	Byte Offset	Description	Type	Unit
Header	0	3BDxh: Predef VAJI Increment Target Pos On Falling Trigger Event	UInt16	-
1. Par	2	Position Increment	SInt32	0.1 um

This command works like the command 'VAJI Increment Target Pos On Falling Trigger Event (3ADxh)'. The VAJI parameters (Maximal Velocity, Maximal Acceleration, Maximal Deceleration, and Jerk) are not given as command arguments here, but are predefined through parameters UPID 0x14BE, UPID 0x14BF, UPID 0x14C0, and UPID 0x14C1.

4.3.268 VAI Modulo Go To Pos Positive Direction (3C0xh)

Name	Byte Offset	Description	Type	Unit
Header	0	010xh: VAI Go To Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²

This command sets a new Target Position in positive direction and defines the maximal velocity, acceleration, and deceleration for the movement. The command execution starts immediately when the command has been sent. The setpoints (Demand Position, Demand Velocity, and Demand Acceleration) are calculated by the internal Velocity Acceleration Interpolator (VAI). This command initializes the VAI with the current Demand Position and Demand Velocity value. Therefore it is possible to start a new command while a former command is still being executed.

4.3.269 VAI Modulo Go To Pos Negative Direction (3C1xh)

Name	Byte Offset	Description	Type	Unit
Header	0	010xh: VAI Go To Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command sets a new Target Position in negative direction and defines the maximal velocity, acceleration, and deceleration for the movement. The command execution starts immediately when the command has been sent. The setpoints (Demand Position, Demand Velocity, and Demand Acceleration) are calculated by the internal Velocity Acceleration Interpolator (VAI). This command initializes the VAI with the current Demand Position and Demand Velocity value. Therefore it is possible to start a new command while a former command is still being executed.

4.3.270 VAI Modulo Go To Pos Shortest Way (3C2xh)

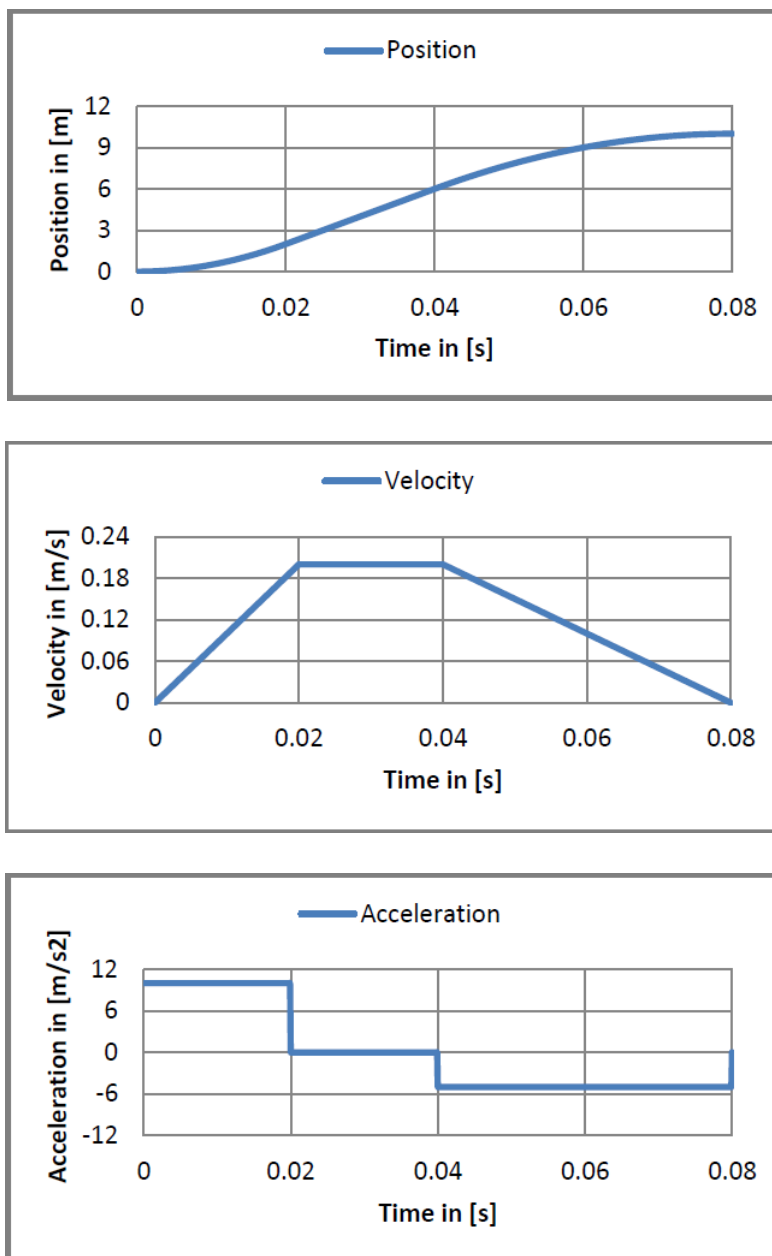
Name	Byte Offset	Description	Type	Unit
Header	0	010xh: VAI Go To Pos	UInt16	-
1. Par	2	Target Position	SInt32	0.1 um
2. Par	6	Maximal Velocity	UInt32	1E-6 m/s
3. Par	10	Acceleration	UInt32	1E-5 m/s ²
4. Par	14	Deceleration	UInt32	1E-5 m/s ²

This command sets a new Target Position and defines the maximal velocity, acceleration, and deceleration for the Shortest Way movement. The command execution starts immediately when the command has been sent. The setpoints (Demand Position, Demand Velocity, and Demand Acceleration) are calculated by the internal Velocity Acceleration Interpolator (VAI). This command initializes the VAI with the current Demand Position and Demand Velocity value. Therefore it is possible to start a new command while a former command is still being executed.

5 Set-point Generation

5.1 VA-Interpolator

The VA-Interpolator generates a position curve from one position to another due to the parameter values of acceleration, deceleration, and maximal speed. A new target position can be set even if the old target position was not reached.



5.1.1 Input Variables

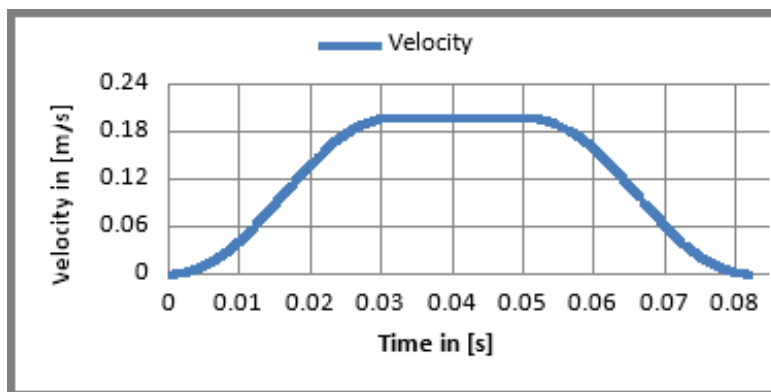
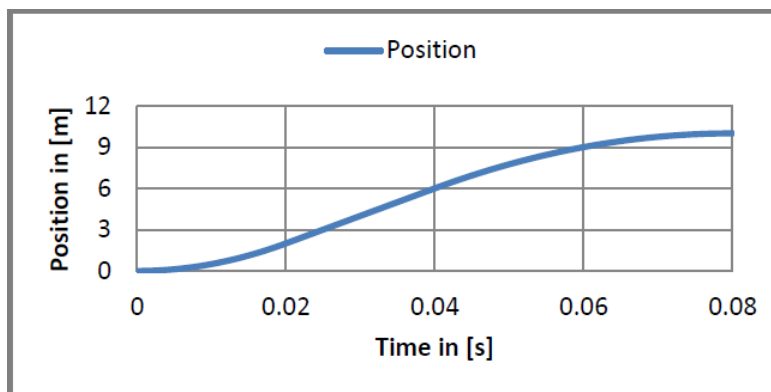
Name	Type	Valid range	Unit
Target Position	SInt32	-8000'0000h ... 7FFF'FFFFh	0.1µm
Maximal Speed	UInt32	0h ... 7FFF'FFFFh	1E-6m/s
Acceleration	UInt32	0h ... 7FFF'FFFFh	1E-5m/s ²
Deceleration	UInt32	0h ... 7FFF'FFFFh	1E-5m/s ²

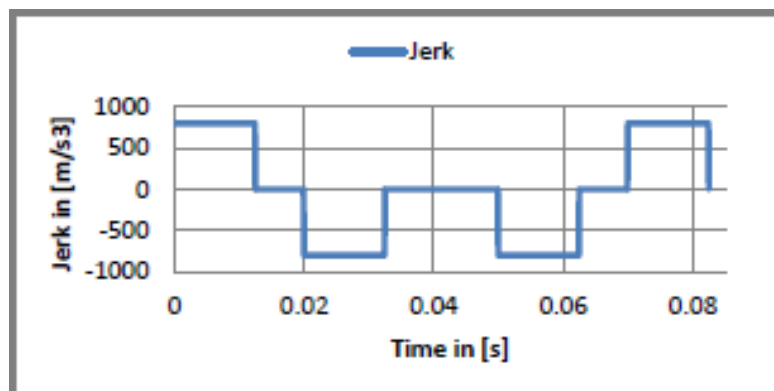
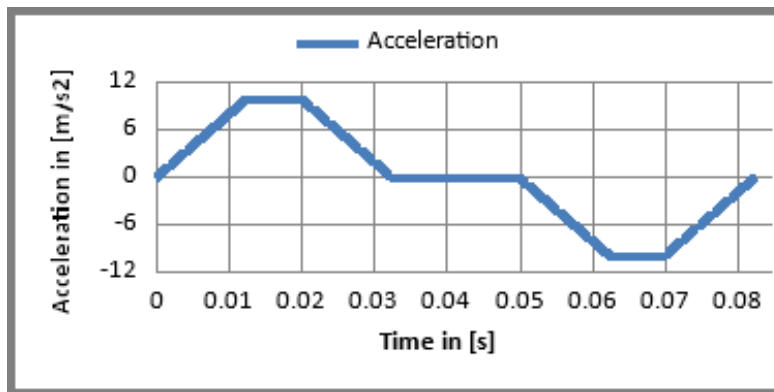
5.1.2 Output Variables

Name	Type	Valid range	Unit
Position	SInt32	-8000'0000h ... 7FFF'FFFFh	0.1µm
Velocity	SInt32	-8000'0000h ... 7FFF'FFFFh	1E-6m/s
Acceleration	SInt32	-8000'0000h ... 7FFF'FFFFh	1E-5m/s ²

5.2 VAJ-Interpolator

The VAJ-Interpolator generates a position curve from one position to another due to the parameter values of jerk, maximal acceleration, maximal deceleration, and maximal speed. A new target position can be set even if the old target position was not reached.





5.2.1 Input Variables

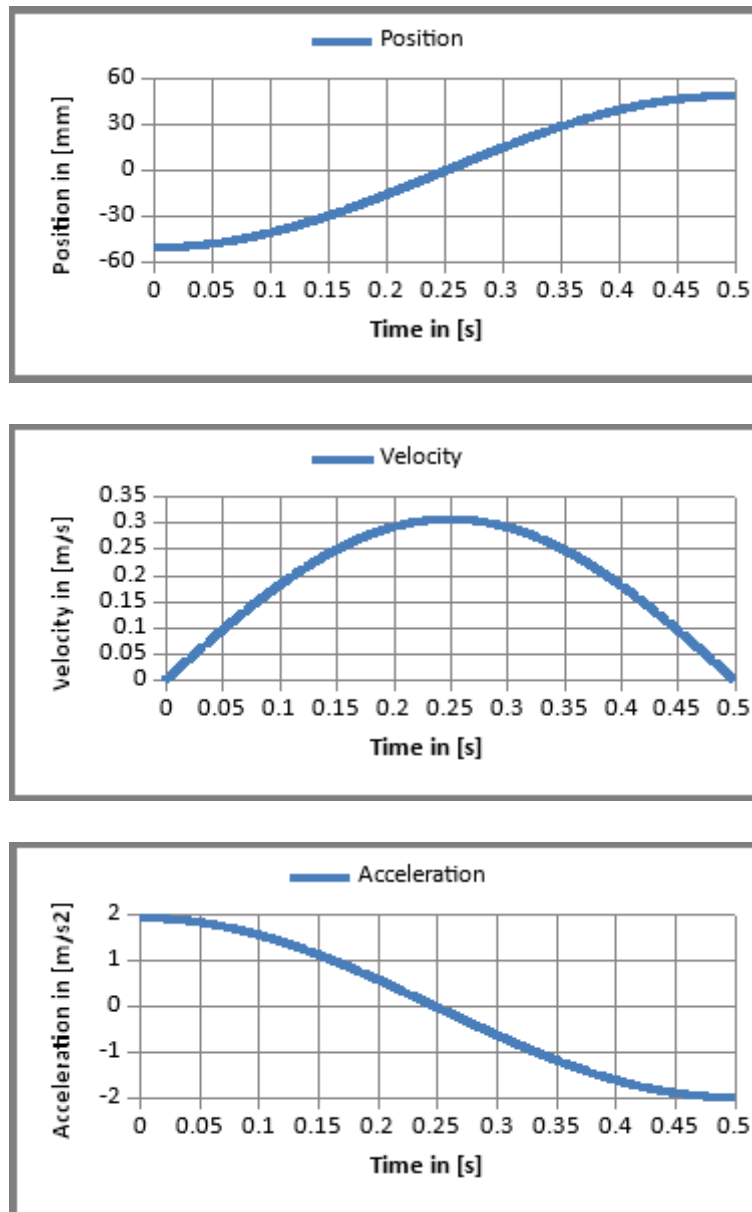
Name	Type	Valid range	Unit
Target Position	SInt32	-8000'0000h ... 7FFF'FFFFh	0.1µm
Maximal Speed	UInt32	0h ... 7FFF'FFFFh	1E-6m/s
Maximal Acceleration	UInt32	0h ... 7FFF'FFFFh	1E-5m/s²
Maximal Deceleration	UInt32	0h ... 7FFF'FFFFh	1E-5m/s²
Jerk	UInt32	0h ... 5FFF'FFFFh	1E-4m/s³

5.2.2 Output Variables

Name	Type	Valid range	Unit
Position	SInt32	-8000'0000h ... 7FFF'FFFFh	0.1µm
Velocity	SInt32	-8000'0000h ... 7FFF'FFFFh	1E-6m/s
Acceleration	SInt32	-8000'0000h ... 7FFF'FFFFh	1E-5m/s²
Jerk	SInt32	-8000'0000h ... 7FFF'FFFFh	1E-4m/s³

5.3 Sine VA Motion

The Sine generates a position curve from one position to another due to the parameter values of target position, maximal speed and acceleration. A new target position can only be started when the previous motion was finished.



5.3.1 Input Variables

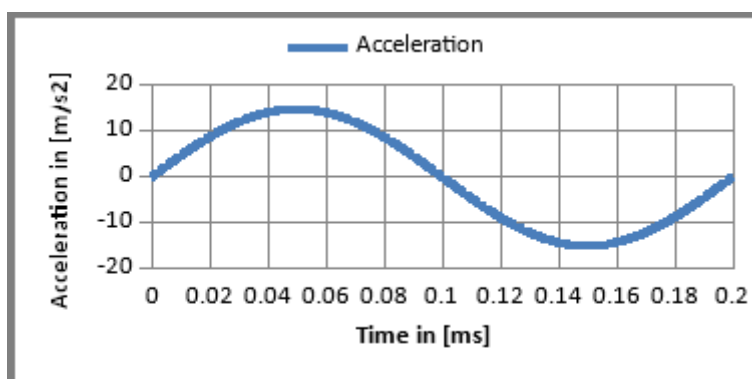
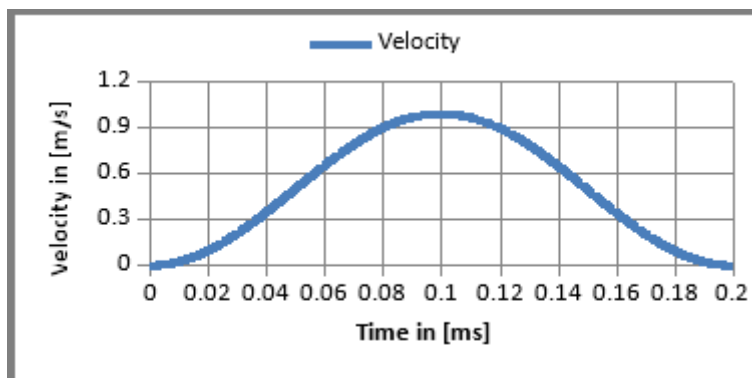
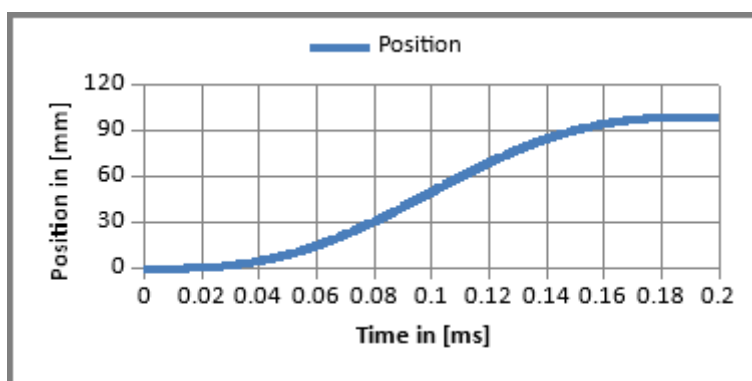
Name	Type	Valid range	Unit
Target Position	SInt32	-8000'0000h ... 7FFF'FFFFh	0.1µm
Maximal Speed	UInt32	0h ... 7FFF'FFFFh	1E-6m/s
Maximal Acceleration/Deceleration	UInt32	0h ... 7FFF'FFFFh	1E-5m/s²

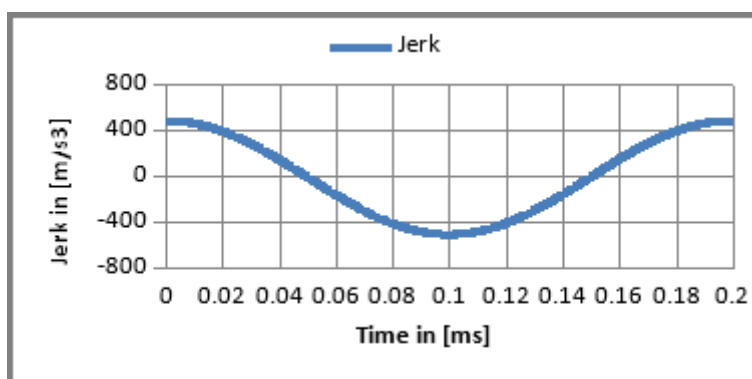
5.3.2 Output Variables

Name	Type	Valid range	Unit
Position	SInt32	-8000'0000h ... 7FFF'FFFFh	0.1µm
Velocity	SInt32	-8000'0000h ... 7FFF'FFFFh	1E-6m/s
Acceleration	SInt32	-8000'0000h ... 7FFF'FFFFh	1E-5m/s²

5.4 Besthorn VAJ Motion

The Besthorn generates a position curve from one position to another due to the parameter values of target position, maximal speed acceleration and jerk. A new target position can only be started when the previous motion was finished.





5.4.1 Input Variables

Name	Type	Valid range	Unit
Target Position	SInt32	-8000'0000h ... 7FFF'FFFFh	0.1µm
Maximal Speed	UInt32	0h ... 7FFF'FFFFh	1E-6m/s
Maximal Acceleration/Deceleration	UInt32	0h ... 7FFF'FFFFh	1E-5m/s²
Maximal Jerk	UInt32	0h ... 7FFF'FFFFh	1E-4m/s³

5.4.2 Output Variables

Name	Type	Valid range	Unit
Position	SInt32	-8000'0000h ... 7FFF'FFFFh	0.1µm
Velocity	SInt32	-8000'0000h ... 7FFF'FFFFh	1E-6m/s
Acceleration	SInt32	-8000'0000h ... 7FFF'FFFFh	1E-5m/s²

5.5 P/PV/PVA-Stream

For masters with NC (Numerical Control) capabilities, the software supports cyclic streaming modes of the position and velocity, or position only. The streaming has to be strictly cyclic in the period range 2ms to 5ms. This feature is supported with all fieldbus variants like EtherCAT, Ethernet/IP, POWERLINK, SERCOS 3, Profibus DP, CAN Open¹.

Different modes are supported:

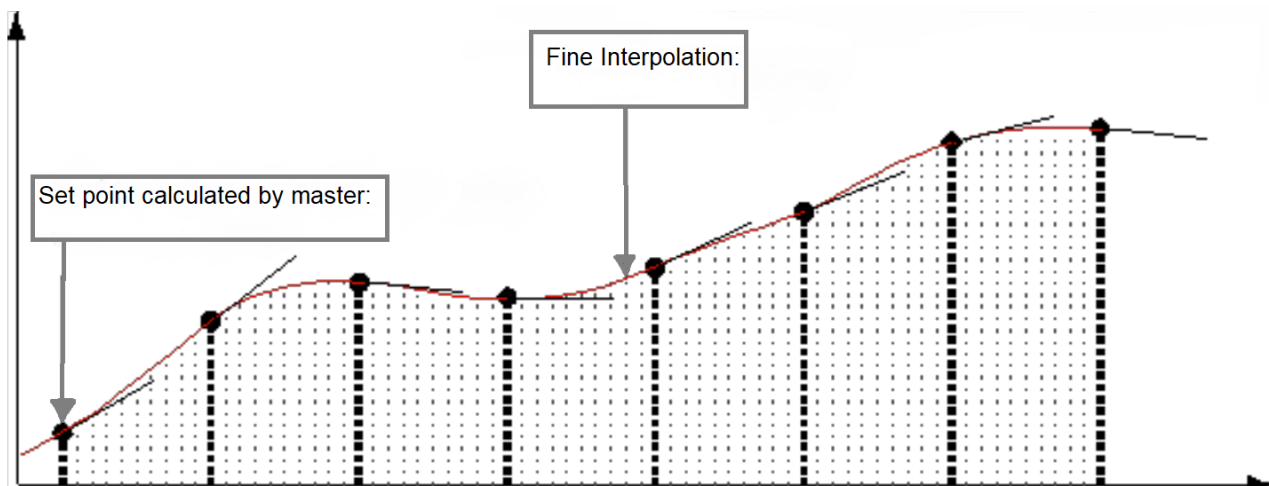
In the first mode (Motion command: 030xh) the master only streams the position.

In the second mode (Motion command: 031xh) the master streams position and velocity, whenever possible use the PV-streaming mode, because the acceleration derivation is less sensitive to bus jitters than in the position only streaming mode.

The third mode (Motion command: 032xh) is like the first mode, but for the derivation of the velocity and the acceleration a configured period time (UPID 14E6h) is taken, instead of the slave receive time stamp. This minimizes the bad influence of the transmission jitter.

The position setpoint generation in these modes is delayed 1.5 times the streaming cycle time, e.g. with 2ms streaming period time the delay is 3ms.

¹ P(V)-Streaming with CANOpen is only possible with transmission type 1 (synchronous transmission)



5.6 Cam Motions

NOTE: Cam motions are not available in the B1100 servo drives.

For high speed synchronization different modes of cam motions are supported. The motions are defined with cam curves, which can be defined with the curve tool of the LinMot-Talk software.

5.6.1 Triggered Cam Motions

For triggered cam motions setup the master encoder and then set cam mode (UPID1527h) to infinite length. Then use the motion commands 069xh or 06Bxh to setup the cam curves which should be started at the trigger event, or use the triggered cam curve run mode.

5.6.2 Repeated Cam Motions with the Modulo CamMode

For repeated cam motions setup the master encoder and then set cam mode (UPID1527h) to modulo cam length. The cam length can be defined independently from the master encoder length! So with a 8000 counts ABZ master encoder also 16'000 counts or 4'000 counts cam length is possible.

With the following sequence a startup or resynchronization to a standing master encoder is possible:

- Move to start position of cam 1 motion e.g. VAI GoTo Pos (010xh)
- Define cam 1 motion e.g. Encoder CAM 1 Define Curve To Pos (112xh)
- Move to start position of cam 2 motion e.g. VAI GoTo Pos (010xh)
- Define cam 2 motion e.g. Encoder CAM 2 Define Curve To Pos (112xh)
- Go to synchronized cam position with (102xh)
- After synchronized cam position is reached enable cam motion with (100xh)
- Start turning the encoder

With the following sequence a (re)synchronization to a moving master encoder is possible:

- Disable cam start enable (UPID 1528h) and wait until cam enabled vanishes, or clear also cam enabled (1BA9h)
- Move to start position of cam 1 motion e.g. VAI GoTo Pos (010xh)
- Define cam 1 motion e.g. Encoder CAM 1 Define Curve To Pos (112xh)
- Define cam 2 motion e.g. Encoder CAM 2 Define Curve To Pos (112xh)
- After the synchronized cam position is reached, enable the cam motion with (100xh)
- Enable cam start enable (UPID 1528h)

6 Command Table

NOTE: The command table in the B1100 servo drives is limited to 31 entries, and is flash only, for this reason it is not possible to write or modify the table over a serial interface!

The command table functionality can be used for programming sequences directly in the drive. The following examples illustrate the possibilities of the command table. The shown command table can be loaded from the defaults 'CT_Example.lmc'.

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Sequenced Entry
1	Go To 50mm	VAI Go To Pos	Pos: 50 mm	Vel: 0.5 m/s	Acc: 2 m/s ²	Dec: 1 m/s ²	2 (Wait Motion ...)
2	Wait Motion Done	Wait until Motion Finished					3 (GoTo 0 mm)
3	GoTo 0 mm	VAI Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s ²	Dec: 10 m/s ²	None
4							
5	Reset CT Var1	Set Cmd Table Var 1 To	Set Val: 0				6 (GoTo 50mm)
6	GoTo 50mm	VAI Go To Pos	Pos: 50 mm	Vel: 1 m/s	Acc: 10 m/s ²	Dec: 10 m/s ²	7 (Wait In Pos)
7	Wait In Pos	Wait until In Target Position					8 (GoTo 0 mm)
8	GoTo 0 mm	VAI Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s ²	Dec: 10 m/s ²	9 (Wait Motion ...)
9	Wait Motion Done	Wait until Motion Finished					10 (Inc CT Var1)
10	Inc CT Var1	Add To Cmd Table Var 1	Add Val: 1				11 (IF Var1 < 5)
11	IF Var1 < 5	IF Cmd Table Var 1 Less Than	Val: 5	True Cmd ID: 6 (...)	False Cmd ID: 12...		None
12	End Seq	No Operation					None
13							
14	Reduce Force	Write Live Parameter	UPID: 13A6h (M...	Value: 1 A			15 (GoTo 50mm)
15	GoTo 50mm	VAI Go To Pos	Pos: 50 mm	Vel: 0.05 m/s	Acc: 1 m/s ²	Dec: 1 m/s ²	16 (Wait Motion ...)
16	Wait Motion Done	Wait until Motion Finished					17 (Test For Part)
17	Test For Part	IF Actual Position Less Than	Val: 48 mm	True Cmd ID: 18...	False Cmd ID: 21...		None
18	GoTo 0 mm	VAI Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s ²	Dec: 10 m/s ²	19 (Set Normal F...
19	Set Normal Force	Write Live Parameter	UPID: 13A6h (M...	Value: 4 A			20 (CT Var1 = 0)
20	CT Var1 = 0	Set Cmd Table Var 1 To	Set Val: 0				None
21	GoTo 0 mm	VAI Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s ²	Dec: 10 m/s ²	22 (Set Normal F...
22	Set Normal Force	Write Live Parameter	UPID: 13A6h (M...	Value: 4 A			23 (CT Var1 = 1)
23	CT Var1 = 1	Set Cmd Table Var 1 To	Set Val: 1				None
24							
25							

The first sequence ID1, ID2 and ID3 shows a simple sequence GoTo 50mm wait until motion is completed and then go back to 0mm.

The second sequence ID5..12 repeats 5 times the Sequence GoTo 50mm GoTo 0mm

1. ID 5: Set Command Table To Var 1= 0
2. ID 6: GoTo 50mm
3. ID 7: Wait Until In Target Position
4. ID 8: GoTo 0mm
5. ID 9: Wait until Motion Done
6. ID 10: Increment Command Table Var 1
7. ID 11: If Command Table To Var 1 < 5 Then GoTo ID 6 Else GoTo ID 12
8. ID 12: No Operation End of Sequence

The third sequence ID14..23 changes the maximal current of the position controller set A (UPID 13A6), goes to 50mm wait until motion completed, tests if part is present (Act Pos < 48), goes back to 0mm, sets Normal Force (Current) and sets CT Var 1 = 0 if part was present else set CT Var 1 = 1.

7 Drive Configuration

The parameter configuration is normally done with LinMot-Talk software [3]. The UPIDs, over which the parameter can be accessed, are the same for C1100, C1250, E1100, E1200 and E1400 drives, but are different for the B1100 drives. In this documentation the C1100/C1250/E1100/E1200/E1400 UPIDs are used. If a UPID for a B1100 servo drive is needed, a conversion list can be generated with the LinMot-Talk software.

7.1 Power Bridge

The servo drives are divided into four different power classes. The drives without power extension on the name have a maximal current of 8A, the high current (name extension HC) variant has a maximal current of 15A and the extreme current (name extension XC) has a maximal current of 25A. The E1200 series is only available as ultra current controllers (name extension UC), with a current maximum of 32A.

7.2 X4 I/O Definitions

The functionality of most IOs can be programmed as a control word input bit or status word output bit, or they can be used as interface IO and read out or written over a serial bus interface. Apart from this general functionality a few IOs have a special functionality.

Descriptor	General Purpose IO	Special Functions
X4.3	Yes	Brake (Output)
X4.4	Yes	Analog In
X4.5	Yes	Capture Input
X4.6	Yes	Trigger (Input)
X4.7	Yes	Home Switch (Input)
X4.8	Yes	Limit IN (Input)
X4.9	Yes	Limit OUT (Input) / 24V Step (Input)
X4.10	Yes	PTC 1 (Input) / 24V Direction (Input)
X4.11	Yes	PTC 2 (Input)
X4.12	No	SVE Safety Voltage Enable (Input)

7.2.1 Brake Output

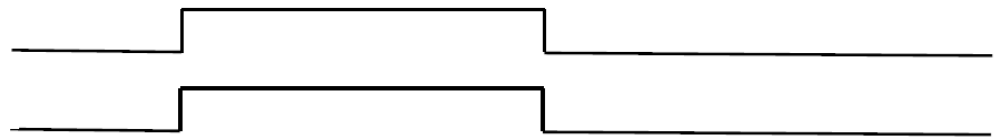
The Brake Output can be used to control directly a valve of a pneumatic brake module. The cases in which the brake has to be applied or released are configured over the brake mode parameters. The brake output is controlled from the state machine.

Parameter Name	UPID	Description
Status Word: Operation Enabled	1717h	The brake is released (Brake Output = 24V), when bit 0 of the status word (Operation Enabled) is set. Otherwise the brake is applied.
Ctrl Word: /Abort	1718h	The brake is applied (Brake Output = 0V) when entering the Aborting State (12) and released (Brake Output = 24V) when going to Operation Enabled State (8) again.
Quick Stop	1716h	Special brake behavior with Quick Stop (Brake Mode Status Word: Operation Enabled has also to be set)
Switch On	1719h	The Brake apply delay time is applied when switching of the operation or the power stage, when switching on the release delay time is applied.

The following figure shows the behavior when only the brake mode “Status Word: Operation Enabled” is selected. If only this switch (with UPID 1717h) is activated, no apply or release delay time is regarded.

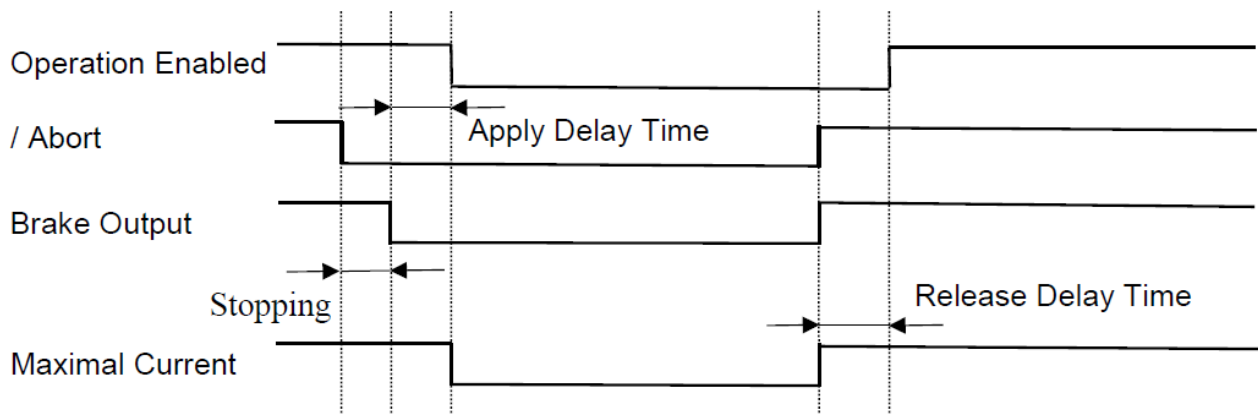
Operation Enabled

Brake Output



7.2.1.1 Brake Operation with Abort Behavior

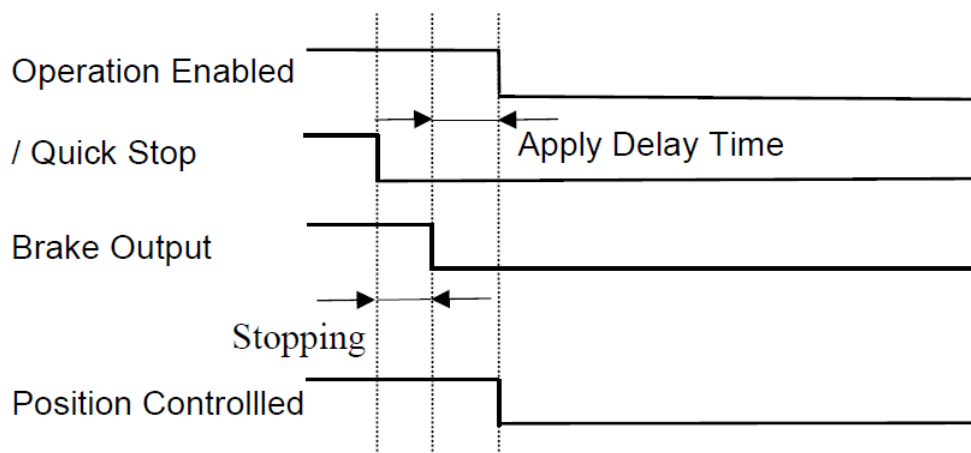
When set the brake mode to “Ctrl Word: /Abort”, the brake is applied in state Aborting (12) and the maximal current of the motor is set to 0A. If using this behavior, ensure that the maximal current (UPID 13A6h and 13BAh) is not set over a serial bus during the state Aborting (12)!



Parameter Name	UPID	Description
Apply Delay Time	171Bh	Delay time after brake output goes to 0V until the maximal current of the motor is set to 0A (UPID 13A6h and 13BAh).
Release Delay Time	171Ch	Delay time of the Status Word bit: Operation Enabled after the motion has stopped.

7.2.1.2 Brake Operation with Quick Stop Behavior

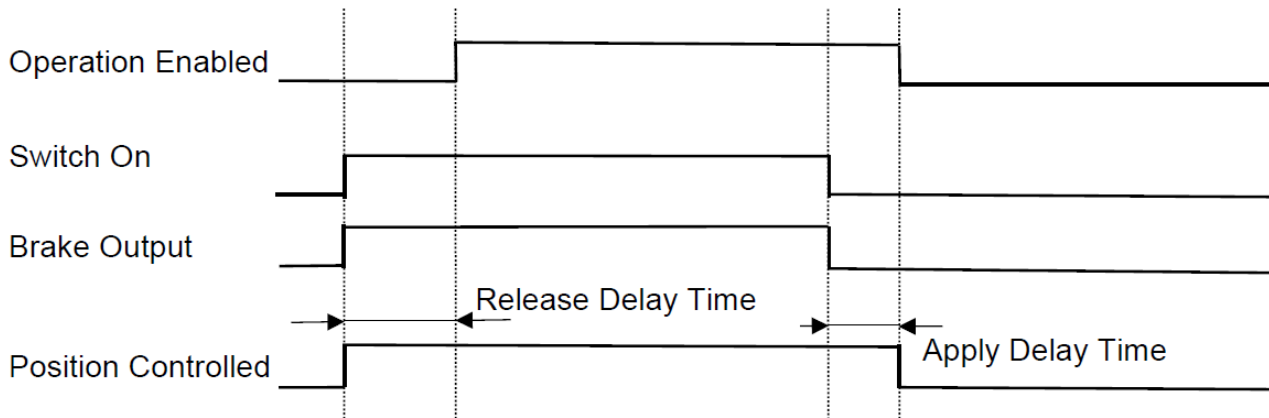
If also the Quick Stop brake behavior is selected the brake is applied as soon as the motion has stopped, then the reset of the Operation Enabled bit is delayed by the apply delay time, which then also powers off the motor.



Parameter Name	UPID	Description
Apply Delay Time	171Bh	Delay time of the Status Word bit: Operation Enabled after the motion has stopped.

7.2.1.3 Brake Operation with Switch On Behavior

If also the SwitchOn brake behavior is selected, the brake Release Delay Time is waited before entering the state Operation Enabled, when switching on. When Switching off the Apply Delay Time is waited before the position controller is switched off.



Parameter Name	UPID	Description
Apply Delay Time	171Bh	Delay time after brake output goes to 0V until the maximal current of the motor is set to 0A (UPID 13A6h and 13BAh).
Release Delay Time	171Ch	Delay time of the Status Word bit: Operation Enabled after the motion has stopped.

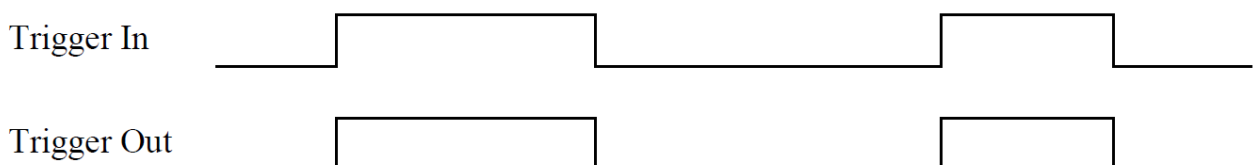
7.2.2 Trigger Input

For the trigger input, which is evaluated in the motion control task, different evaluation modes are supported.

Parameter Name	UPID	Description
Trigger Modes	170Ch	Trigger mode selection: <ul style="list-style-type: none"> • 0: None • 1: Direct • 2: Inhibited • 3: Delayed • 4: Inhibited & Delayed

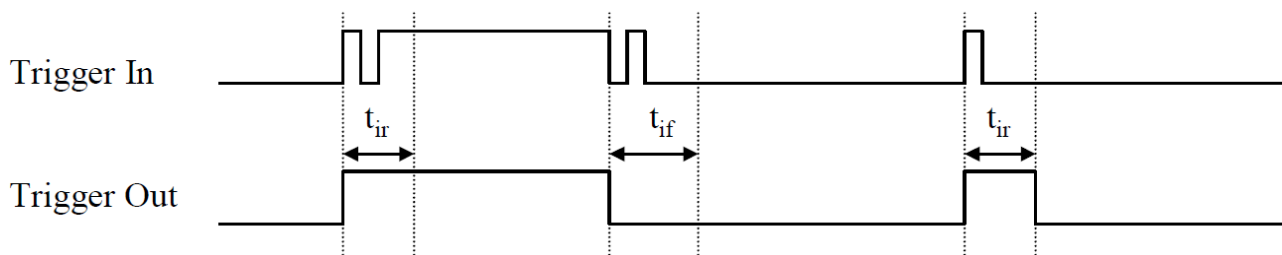
7.2.2.1 Direct Trigger Mode

In the Direct Trigger Mode copies the trigger input directly copied to the trigger output, which is used by the MC-SW. No parameter configuration is needed for this mode.



7.2.2.2 Inhibited Trigger Mode

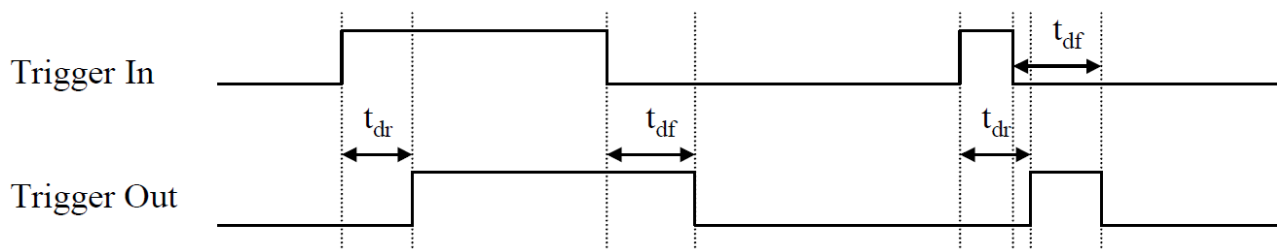
The Inhibit Trigger Mode can be used to debounce a jittering trigger input without delay. This method doesn't increase the noise immunity!



Parameter Name	UPID	Description
Rise Inhibit Time	170Dh	Inhibit time after rising edge of trigger in signal (tir)
Fall Inhibit Time	170Eh	Inhibit time after falling edge of trigger in signal (tif)

7.2.2.3 Delayed Trigger Mode

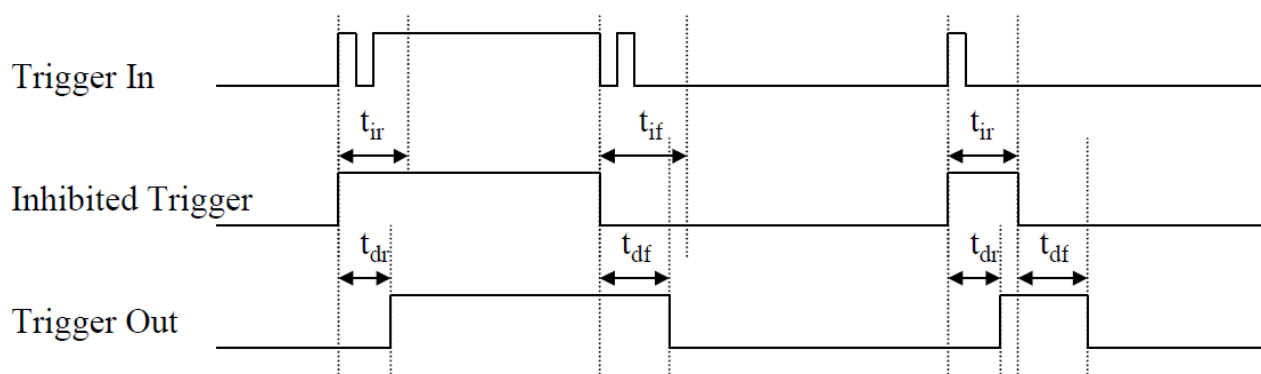
The Delayed Trigger Mode can be used to delay an action following the rising or falling trigger event.



Parameter Name	UPID	Description
Rise Delay Time	170Fh	Trigger Out delay time after rising edge of Trigger In signal (tdr)
Fall Delay Time	1710h	Trigger Out delay time after falling edge of Trigger In signal (tdf)

7.2.2.4 Inhibited & Delayed Trigger Mode

The Inhibited & Delayed Trigger Mode first debounces the trigger input signal with the inhibit time. The inhibited trigger signal is then delayed with the rise/fall delay time and then copied to the trigger out signal.



Parameter Name	UPID	Description
Rise Inhibit Time	170Dh	Inhibit time after rising edge of Trigger In signal (tir)
Fall Inhibit Time	170Eh	Inhibit time after falling edge of Trigger In signal (tif)
Rise Delay Time	170Fh	Trigger Out Delay time after rising edge of Trigger In signal (tdr)
Fall Delay Time	1710h	Trigger Out Delay time after falling edge of Trigger In signal (tdf)

7.2.3 X4.8 and X4.9 Limit Switches

If on IO pin X4.8 and/or on X4.9 a limit switch is defined, the error behavior in case of an active limit switch can be configured. The error is only generated on a powered motor which is homed, so it is possible to exit the error state with a new homing request. For that reason also the homed bit in the status word is cleared if moving into a limit switch.

Parameter Name	UPID	Description
Error Behavior	121Bh	Error behavior of the configured limit switches: <ul style="list-style-type: none">• 0: No Error (maybe only used for homing)• 1: Power Off• 2: Quick Stop

7.2.4 X4.10 and X4.11 PTC 1 and PTC 2

On IO pins X4.10 and X4.11 PTC sensors can be connected to supervise an over temperature of a motor and/or the transformer supply. The other side of the PTC has to be connected to 24V. If the resistance is below 2kOhm the input is high. If it rises above 15kOhm the output is regarded as low. Typically, the transition low to high is between 5kOhm and 7kOhm. If the input rises, the corresponding bit in the warn word will be set after the PTC Warn Time. After the PTC error time the corresponding error is set. The error only can be acknowledged if the corresponding PTC warn bit has vanished.

7.3 Master Encoder

The master encoder can be used for applications where high-speed synchronization is needed, such as winding application, indexing positioning mode, etc.

The parameter Encoder Source defines whether the master encoder is connected to the X10 (RJ45) or the X12 (DSUB 9) connector. X10 is looped through to the master encoder connector X11 (RJ45). This simplifies the loop through cabling of the encoder signals.

Parameter Name	UPID	Description
Encoder Source	172Ah	Defines the source of the Master Encoder: <ul style="list-style-type: none">• 0: None• 1: Encoder Input X10• 2: Ext Sensor Input X12

The encoder type parameter defines the signal type of the master encoder. At any rate the signals have to be differential RS422. When using a SD(Z) master encoder, the step signal has to be connected to A, /A, Dir signal to B, /B and the Zero to Z, /Z.

Parameter Name	UPID	Description
Encoder Type	128Eh	Defines the Master Encoder Type: <ul style="list-style-type: none">• 0: None• 1: ABZ• 2: AB• 4: Step Dir (SD)• 5: Step Dir Zero (SDZ)

The decoding parameter defines how many edges of the signals are evaluated. For AB(Z) encoder types 1x, 2x, 4x decoding is valid; for SD(Z) encoder types only 1x and 2x decoding is valid. If 4x decoding is selected a 2x decoding is forced without error generation.

Parameter Name	UPID	Description
Decoding	128Fh	Defines the Master Encoder decoding:

Parameter Name	UPID	Description
		<ul style="list-style-type: none"> 0: 1x 1: 2x 2: 4x

With the direction parameter the count direction of the encoder evaluation can be defined.

Parameter Name	UPID	Description
Direction	1290h	Defines the Master Encoder direction: <ul style="list-style-type: none"> 0: Positive 1: Negative

The Counts/Revolution parameter defines the number of counts per revolution for ABZ or SDZ master encoders.

Parameter Name	UPID	Description
Counts/ Revolution	1291h	Only used for ABZ or SDZ encoders, to define the counts per revolution.

The Speed Filter Time parameter defines the time over which the master encoder speed is calculated. Bigger values generate a smoother master encoder speed but generate a bigger delay time of the speed, which may be a problem if the master encoder speed varies quickly.

Parameter Name	UPID	Description
Speed Filter Time	1293h	Time over which the Master Encoder speed is generated.

7.4 Monitoring

All servo drive series support enhanced monitoring features of the supply voltages and the board temperatures.

7.4.1 Logic Supply Voltage

The logic supply voltage must be in the range from 20V to 28V when using the default parameter settings. If the supply voltage goes out of this range, an error will be generated. If the supply voltage goes below approx. 18V the servo drive shuts down.

Parameter Name	UPID	Description
Logic Supply Voltage Too Low	100Eh	If the logic supply voltage sinks below this limit the error 01h will be generated.

Parameter Name	UPID	Description
Logic Supply Voltage Too High	1010h	If the logic supply voltage rises above this limit the error 02h will be generated.

7.4.2 Motor Supply Voltage

The motor supply voltage can be monitored with different warn and error levels.

Parameter Name	UPID	Description
Motor Supply Voltage Low	1018h	If the motor supply voltage sinks below this limit the warn bit Motor Supply Voltage Low (bit 2) is set.

Parameter Name	UPID	Description
Motor Supply Voltage High	1019h	If the logic supply voltage rises above this limit the warn bit Motor Supply Voltage High (bit 3) is set.

Parameter Name	UPID	Description
Motor Supply Voltage Too Low	101Ah	If the logic supply voltage sinks below this limit the error 03h will be generated.
Motor Supply Voltage Too High	101Bh	If the logic supply voltage rises above this limit the error 04h will be generated.

7.4.2.1 Phase Switch On Test

The parameters in the Phase Switch On Test section are used in the HW Tests State (State 5) before enabling the power stage. If the motor power supply is on and there is no ground path in the motor (inclusive cabling) the phase voltage is approx. 6.5V when in power off state. When moving the motor the induced voltage may lead to HW test error.

Parameter Name	UPID	Description
Phase Voltage Low Level	102Ch	If one of the phase voltages is below this limit before powering up an error will be generated.
Phase Voltage High Level	102Dh	If one of the phase voltages is above this limit before powering up an error will be generated.

As a next motor cabling test step, one single end of a phase is set to half of the motor supply voltage. At this time, the current into the servo drive has to be below the Phase Test Max Incurrent value. An error will be generated otherwise.

Parameter Name	UPID	Description
Phase Test Max Incurrent	102Eh	If the current rises above this limit if one edge of a phase is set to a voltage an error will be generated.

7.4.3 Regeneration Resistor

The regeneration resistor terminals on X1 can be used for energy dissipation, when the motor is decelerating.

Parameter Name	UPID	Description
Enable	101Dh	The regeneration resistor output can be activated with this parameter: <ul style="list-style-type: none">• 0: Disable• 1: Enable

In the configuration section the switch on and off voltage levels for the regeneration resistor can be defined. The turn on voltage has to be at minimum 0.5V higher than the turn off voltage. Ensure that the idle motor supply voltage is lower than the turn off voltage!

Parameter Name	UPID	Description
Turn On Voltage	101Eh	If the Motor Supply Voltage rises above this limit the low side switch of the regeneration output is activated.
Turn Off Voltage	101Fh	If the Motor Supply Voltage sinks below this limit the low side switch of the regeneration output is turned off.

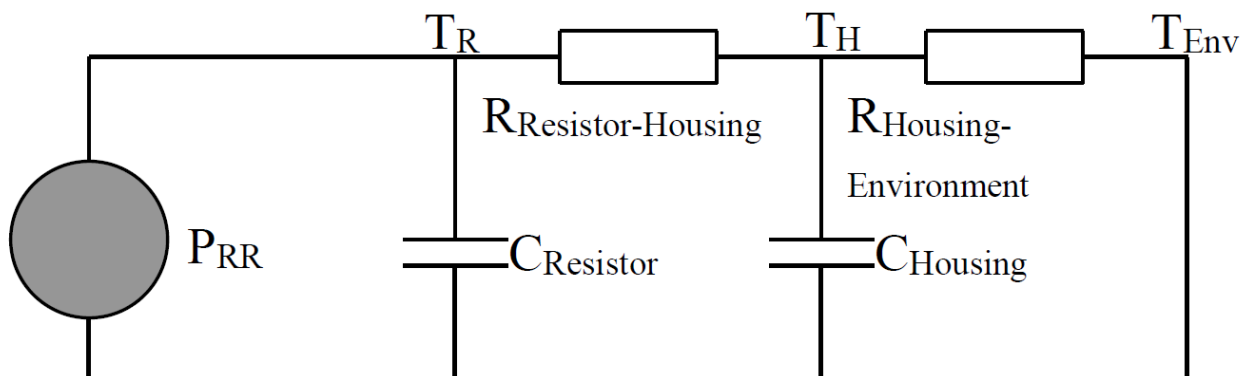
With the RR Temp Calculated section, the temperature model of the regeneration resistor is parameterized. This is used for avoiding damages to the regeneration resistor if the energy dissipation rises over the resistor's capabilities.

Parameter Name	UPID	Description
RR Resistance	1022h	Resistance value of the regeneration resistor. This value is used for calculating how much heat energy is generated in the resistor at the current DC link voltage.
Warning Temp	1024h	If Calc Temp RR Winding (UPID1C0Dh) of the calculated temperature model rises above this value, bit 10 of the Warn Word is set.
Error Temp	1025h	If Calc Temp RR Winding (UPID1C0Dh) of the calculated temperature model rises above this value, the drive goes to error state with the error 15h.

In the Temp Model Parameters section the regeneration resistor's temperature model is defined.

Parameter Name	UPID	Description
C Resistor	1026h	Heat capacity of the resistor winding itself.
R Winding Housing	1027h	Thermal resistance value between resistor winding and its housing.
C Housing	1028h	Heat capacity of the resistor housing.
R Housing Environment	1029h	Thermal resistance value between resistor housing and environment.
Environment Temperature	102Ah	Environmental temperature of the regeneration resistor.

The figure below shows the implemented temperature model for the regeneration resistor.



7.4.4 Temperature Monitoring

The E1100 servo drive hardware contains eight absolute temperature sensors for thermal protection. On the B1100 drive is one sensor placed.

Parameter Name	UPID	Description
Temp Sens Warn Level	1040h	If the maximal board temperature rises above this level, a warning is generated (bit 6 in Warn Word is set).
Temp Sens Error Level	1041h	If the maximal board temperature rises above this level, the error is generated (error codes 10h..17h).

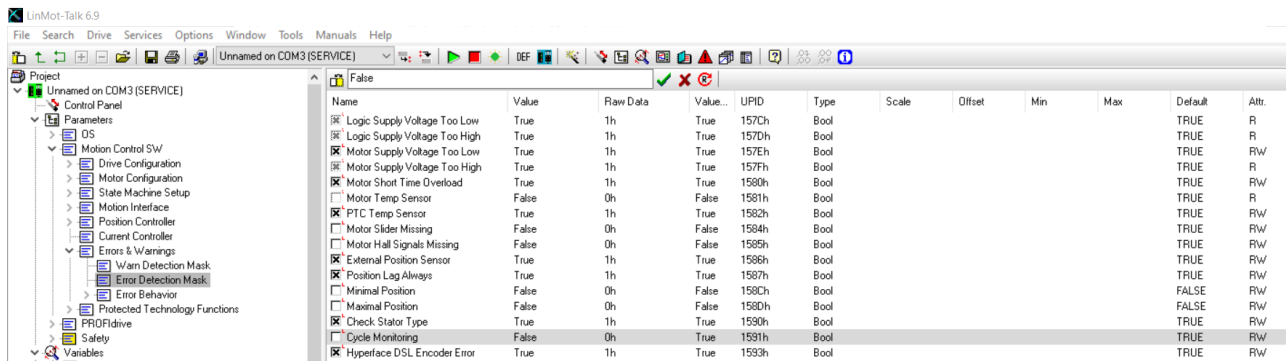
The table below shows the mapped variables of the temperature monitoring.

Variable Name	UPID	Description
Power Bridge Temp Ph1+	1BC6h	Temperature of power driver Phase 1+ (E1100 only)
Power Bridge Temp Ph1-	1BC7h	Temperature of power driver Phase 1- (E1100 only)
Power Bridge Temp Ph2+	1BC8h	Temperature of power driver Phase 2+ (E1100 only)
Power Bridge Temp Ph2-	1BC9h	Temperature of power driver Phase 2- (E1100 only)
Power Bridge Temp Ph2+	1BC9h	Temperature of power driver Phase 2+ (E1100 only)
Power Bridge Temp DCLV+	1BCAh	Temperature of power drivers DC link voltage
Temp RR Driver	1BCBh	Temperature of Regeneration Resistor low side driver
Connector Temp X3	1BCCh	Temperature of motor connector X3
Temp Core	1BCDh	Temperature near microcontroller core (E1100 only)
Max Drive Temp	1BCEh	Maximal temperature of above variables

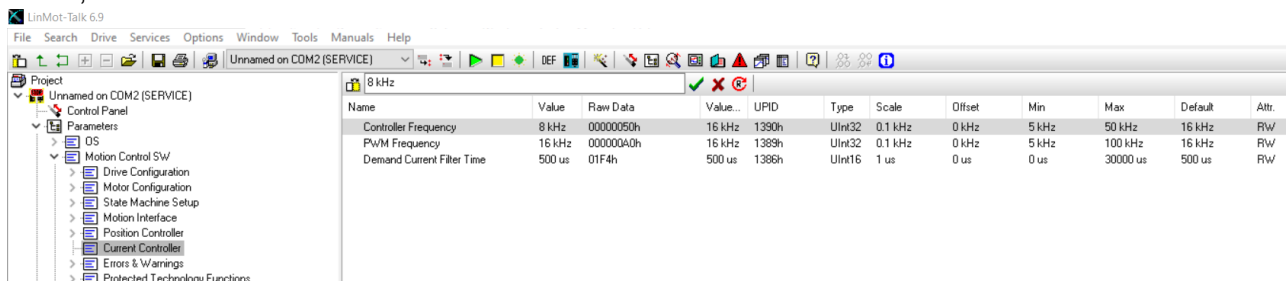
7.4.5 Cycle Time Monitoring

By default a cycle time monitoring of the cyclic tasks[0..3] is activated, in some configurations it is possible that the errors 0x90..0x93 "RunTime error: Too few calculation time for cycle 0..3" occurs, which means that needed cycle time was greater than the maximal possible.

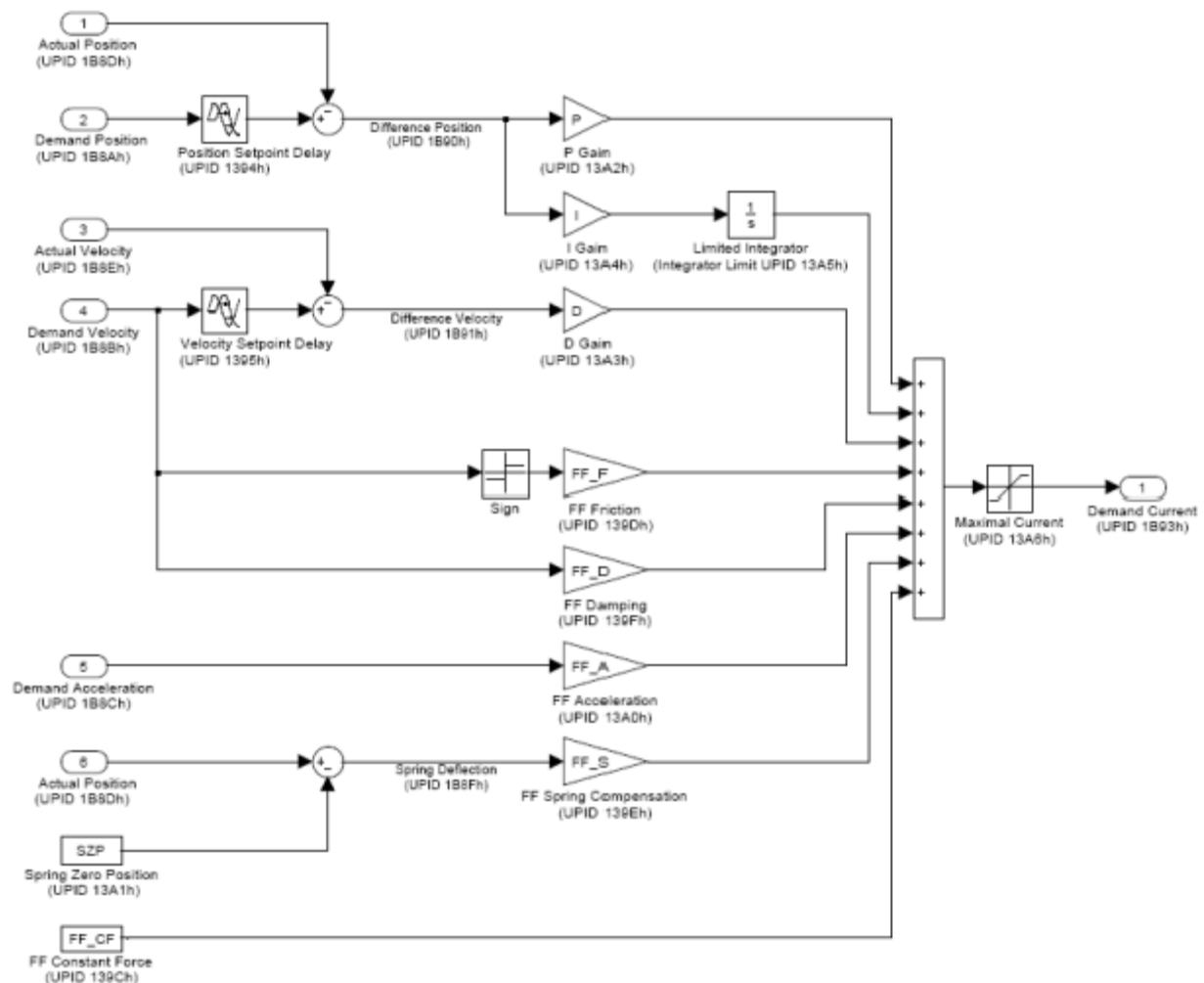
Because this are normally seldom events, in most cases the best and easiest way to avoid this error, is to deactivate the error generation for these events. To deactivate the error generation just set the parameter with the UPID 1591h to false.



If this action leads to a unwanted behavior, these errors can also be avoided by a halving the controller frequency. To do this, reduce the default value of the parameter with the UPID 1390h to its half. No other values, than the half of the default value should be used!



7.5 PosCtrlStructure

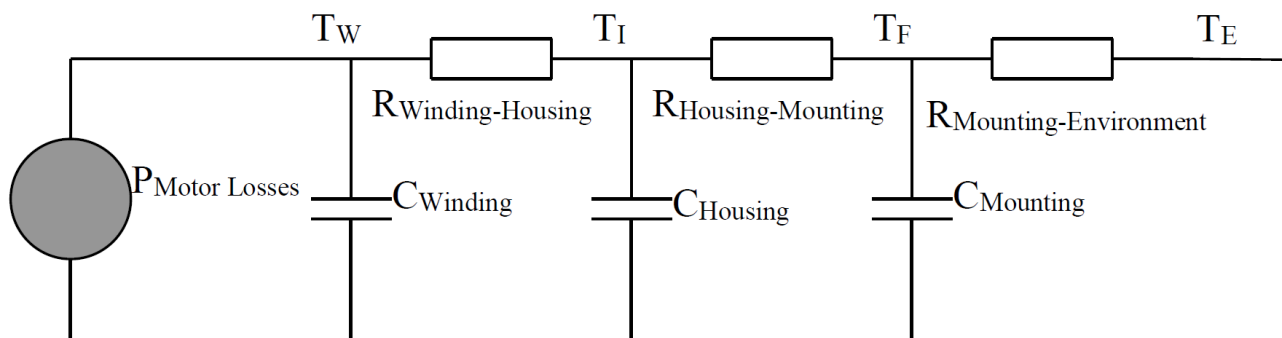


8 Motor Configuration

The motor usually is set up with the motor wizard, which sets all needed parameters. Therefore a detailed description of the parameters will follow in the future.

8.1 Generic Motor Temperature Calculated

For third parties motors a generic calculated motor temperature model is used to adapt the winding resistance and to detect excess temperature.



C Winding: UPID 120Ch
R Winding-Housing: UPID 1210h

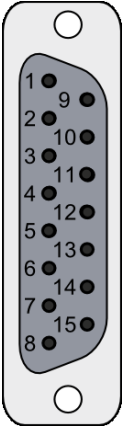
C Housing: UPID 1211h
 R Housing-Mounting: UPID 1212h
 C Mounting: UPID 1213h
 R Mounting-Environment: UPID 1214h

The sum of all R defines the static power losses (consider also TW and TE). With the capacitance the thermal time constant can be influenced. The bigger the thermal capacitance the slower the temperature will rise.

8.2 External Position Sensor

The LinMot motors position accuracy is increased by an additional external position sensor which is connected to X13.

8.2.1 Serial Synchronous Interface (SSI)

SSI		
	Pin	Assignment
	1	5VDC
	2	n.c.
	3	n.c.
	4	Data-
	5	GND
	6	n.c.
	7	n.c.
	8	Clock-
	9	n.c.
	10	n.c.
	11	Data+
	12	n.c.
	13	n.c.
	14	n.c.
	15	Clock+

The Serial Synchronous Interface (SSI) defines a digital serial communication interface with only six wires: two (differential) clock lines, two (differential) data lines, supply voltage (typically 5VDC), and sensor ground (typically 0VDC).

The SSI parameters are either configured automatically by the motor wizard or manually in the parameter tree:

Parameters\Motion Control SW\Motor Configuration\Position Feedback\Feedback on X13

- Select the Sensor Type (UPID 0x125C): SSI
- Select the X13 Sensor Type Selection (UPID 0x1197): Linear or Rotative
- Select General Absolute Sensor Configuration:
 - Resolution (linear sensor) [um]
 - Full Turn Length (rotative sensor) [-]
 - Number of Position Bits [-] (for rotative motors: number of singleturn bits)
 - Number of Multiturn Position Bits [-]
- Select SSI/BiSS Configuration:
 - Count Direction. Positive or Negative with respect to motor direction.
 - Position Decoding: Binary or Gray code.

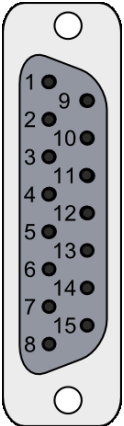
Additionally chose the SSI mode for position recovery in the parameter tree:

Parameters\Motion Control SW\State Machine Setup\Homing\Power Up Position Recovery\Mode>Select SSI

For third-party motors the commutation information can also be taken from the absolute sensor signal:

Parameters\Motion Control SW\Motor Configuration\Commutation\Angle Info Source>Select Absolute Sensor

8.2.2 Bidirectional Serial Synchronous Interface (BiSS-C)

BiSS-C		
	Pin	Assignment
	1	5VDC
	2	n.c.
	3	n.c.
	4	Data-
	5	GND
	6	n.c.
	7	n.c.
	8	Clock-
	9	n.c.
	10	n.c.
	11	Data+
	12	n.c.
	13	n.c.
	14	n.c.
	15	Clock+

The Bidirectional Serial Synchronous Interface (BiSS-C) defines a digital serial communication interface which is hardware compatible with the SSI interface.

The BiSS-C parameters are either configured automatically by the motor wizard or manually in the parameter tree:

Parameters\Motion Control SW\Motor Configuration\Position Feedback\Feedback on X13

- Select the Sensor Type (UPID 0x125C): BiSS-C
- Select the X13 Sensor Type Selection (UPID 0x1197): Linear or Rotative
- Select General Absolute Sensor Configuration:
 - Resolution (linear sensor) [um]
 - Full Turn Length (rotative sensor) [-]
 - Number of Position Bits [-] (for rotative motors: number of singleturn bits)
 - Number of Multiturn Position Bits [-]
- Select SSI/BiSS Configuration:
 - Number of CRC Bits
 - CRC Polynom
 - Count Direction. Positive or Negative with respect to motor direction.

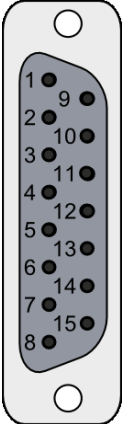
Additionally chose the BiSS-C mode for position recovery in the parameter tree:

Parameters\Motion Control SW\State Machine Setup\Homing\Power Up Position Recovery\Mode\Select BiSS-C

For third-party motors the commutation information can also be taken from the absolute encoder signal:

Parameters\Motion Control SW\Motor Configuration\Commutation\Angle Info Source\Select Absolute Sensor

8.2.3 Bidirectional Serial Synchronous Interface (BiSS-B)

BiSS-B		
	Pin	Assignment
	1	5VDC
	2	n.c.
	3	n.c.
	4	Data-
	5	GND
	6	n.c.
	7	n.c.
	8	Clock-
	9	n.c.
	10	n.c.
	11	Data+
	12	n.c.
	13	n.c.
	14	n.c.
	15	Clock+

The Bidirectional Serial Synchronous Interface (BiSS-B) defines a digital serial communication interface which is hardware compatible with the SSI interface.

The BiSS-B parameters are either configured automatically by the motor wizard or manually in the parameter tree:

Parameters\Motion Control SW\Motor Configuration\Position Feedback\Feedback on X13

- Select the Sensor Type (UPID 0x125C): BiSS-B
- Select the X13 Sensor Type Selection (UPID 0x1197): Linear or Rotative
- Select General Absolute Sensor Configuration:
 - Resolution (linear sensor) [um]
 - Full Turn Length (rotative sensor) [-]
 - Number of Position Bits [-] (for rotative motors: number of singleturn bits)
 - Number of Multiturn Position Bits [-]
- Select SSI/BiSS Configuration:
 - Count Direction. Positive or Negative with respect to motor direction.

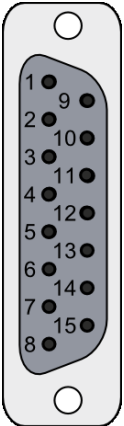
Additionally chose the BiSS-B mode for position recovery in the parameter tree:

Parameters\Motion Control SW\State Machine Setup\Homing\Power Up Position Recovery\Mode\Select BiSS-B

For third-party motors the commutation information can also be taken from the absolute encoder signal:

Parameters\Motion Control SW\Motor Configuration\Commutation\Angle Info Source\Select Absolute Sensor

8.2.4 EnDat2

EnDat2.x		
	Pin	Assignment
	1	5VDC
	2	n.c.
	3	n.c.
	4	Data-
	5	GND
	6	n.c.
	7	n.c.
	8	Clock-
	9	n.c.
	10	n.c.
	11	Data+
	12	n.c.
	13	n.c.
	14	n.c.
	15	Clock+

The EnDat2.x Interface defines a digital serial communication interface which is hardware compatible with the SSI interface.

The EnDat2.x parameters are either configured automatically by the motor wizard or manually in the parameter tree:

Parameters\Motion Control SW\Motor Configuration\Position Feedback\Feedback on X13

- Select the Sensor Type (UPID 0x125C): EnDat2.1 or EnDat2.2
- Select the X13 Sensor Type Selection (UPID 0x1197): Linear or Rotative
- Select General Absolute Sensor Configuration (*):
 - Full Turn Length (rotative encoder) [-]
- Select EnDat2 Configuration:
 - Positive Count Direction: Right/Clockwise or Left/Counter-Clockwise.

Additionally chose the EnDat2.x mode for position recovery in the parameter tree:

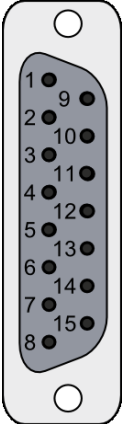
Parameters\Motion Control SW\State Machine Setup\Homing\Power Up Position Recovery\Mode\Select EnDat2.1 or EnDat2.2

For third-party motors the commutation information can also be taken from the absolute encoder signal:

Parameters\Motion Control SW\Motor Configuration\Commutation\Angle Info Source\Select Absolute Sensor

(*) *EnDat2 encoders automatically configure the other sensor parameters at power-up. Endat2 is only available on LinMot drives C1250, C1450 and E14x0.*

8.2.5 Incremental AB Encoder

AB Encoder		
	Pin	Assignment
	1	5VDC
	2	A-
	3	B-
	4	n.c.
	5	GND
	6	n.c.
	7	n.c.
	8	n.c.
	9	A+
	10	B+
	11	n.c.
	12	n.c.
	13	n.c.
	14	n.c.
	15	n.c.

The Incremental AB Encoder defines a digital serial communication with only six wires: two (differential) A lines, two (differential) B lines, supply voltage (typically 5VDC), and sensor ground (typically 0VDC).

The Incremental AB parameters are either configured automatically by the motor wizard or manually in the parameter tree:

Parameters\Motion Control SW\Motor Configuration\Position Feedback\Feedback on X13

- Select the Sensor Type (UPID 0x125C): Incremental AB Encoder
- Select the X13 Sensor Type Selection (UPID 0x1197): Linear or Rotative
- Select Sensor Configuration to configure the encoder:
 - Resolution (1/4 period length) [um]
 - Counts per Revolution [-]
 - Count Direction. Positive or Negative with respect to motor direction.

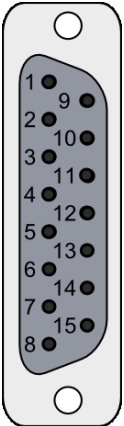
If the external sensor has an additional absolute sensor, then chose the mode for position recovery in the parameter tree:

Parameters\Motion Control SW\State Machine Setup\Homing\Power Up Position Recovery\Mode\Select the correct absolute sensor.

For third-party motors the commutation information can also be taken from the incremental encoder signal:

Parameters\Motion Control SW\Motor Configuration\Commutation\Angle Info Source\Select AB Sensor Position on X13

8.2.6 Incremental ABZ Encoder

ABZ Encoder		
	Pin	Assignment
	1	5VDC
	2	A-
	3	B-
	4	Z-
	5	GND
	6	n.c.
	7	n.c.
	8	n.c.
	9	A+
	10	B+
	11	Z+
	12	n.c.
	13	n.c.
	14	n.c.
	15	n.c.

Several Incremental AB Encoder have an additional Z signal, which serves as a position reference. The Incremental ABZ Encoder is hardware-compatible to the AB Encoder with the additional (differential) Z line.

The Incremental ABZ parameters are either configured automatically by the motor wizard or manually in the parameter tree:

Parameters\Motion Control SW\Motor Configuration\Position Feedback\Feedback on X13

- Select the Sensor Type (UPID 0x125C): Incremental ABZ Encoder
- Select the X13 Sensor Type Selection (UPID 0x1197): Linear or Rotative
- Select Sensor Configuration to configure the encoder:
 - Resolution (1/4 period length) [um]
 - Counts per Revolution [-]
 - Count Direction. Positive or Negative with respect to motor direction.

If the external sensor has an additional absolute sensor, then chose the mode for position recovery in the parameter tree:

Parameters\Motion Control SW\State Machine Setup\Homing\Power Up Position Recovery\Mode\Select the correct absolute sensor.

For third-party motors the commutation information can also be taken from the incremental encoder signal:

Parameters\Motion Control SW\Motor Configuration\Commutation\Angle Info Source\Select ABZ Sensor Position on X13

8.2.7 Commutation

The initial commutation angle is determined depending the chosen Init Mode (UPID 0x11F0). The Init Mode is either configured automatically by the motor wizard or manually in the parameter tree:

Parameters\Motion Control SW\Motor Configuration\Commutation\Init Mode

The table below shows the possible combinations of the Angle Info Source (UPID 0x11EF) and the Init Mode:

		Commutation					
Init Mode	None	SIN/COS On X3 Motor Sensor	AB Sensor Position On X13	ABZ Sensor Position On X13	Hall Switches On X3	Hall Switches On X13	Absolute Sensor
		x			x	x	
	Hall Switches On X3 Until First Z			x			
	Hall Switches On X13 Until First Z			x			
	Hall Switches On X3 Until Hall Switch Edge		x				
	Hall Switches On X13 Until Hall Switch Edge		x				
	Phase Search		x				x
	Absolute Position						x
Motor Communication			x				x

Warning: all other combinations will throw a configuration error (0x6B, Cfg Err: Commutation not Supp) or may result in an unstable behavior.

Workaround for ABZ Incremental Encoder without Hall Switches: set Sensor Type (UPID 0x125C) and Angle Info Source (UPID 0x11EF) parameters as AB Incremental Encoder.

8.3 Modulo Functionality

(supported LinMot drives: C1250)

The modulo functionality is useful whenever rotary motors are controlled by LinMot drives. For that, two additional variables are defined, which run synchronously to the variables Actual Position (UPID 0x1B8D) and Demand Position (UPID 0x1B8A):

- Actual Position Modulo (UPID 0x1B99)
- Demand Position Modulo (UPID 0x1B9A)

Their range is defined by the variables Minimal Position (UPID 0x11C2) and Maximal Position (UPID 0x11C3). Hence, instead of may having an overflow (or underflow) at 214'748.3647 mm (or -214'748.3648 mm) a modulo range can be parametrized.

Note: The control panel always shows the variables Actual Position (UPID 0x1B8D) and Demand Position (UPID 0x1B8A), since internal drive control is based on these variables.

Parameters for Modulo Operation

Four modulo parameters are configured in the parameter tree:

Parameters\Motion Control SW\Motor Configuration\Modulo Configuration

- Minimal Position (UPID 0x11C2):
Position where the variable Actual Position Modulo (UPID 0x1B99) underflows to Maximal Position (UPID 0x11C3).
- Maximal Position (UPID 0x11C3):
Position where the variable Actual Position Modulo (UPID 0x1B99) overflows to Minimal Position (UPID 0x11C2).
- Start-up Position (UPID 0x11C4):
Start-up value of the variable Actual Position Modulo (UPID 0x1B99) and Demand Position Modulo (UPID 0x1B9A) if Power Up Position Recovery (UPID 0x13E0) is set to 0 (None).
- Modulo Mode (UPID 0x11C5):
 - Off
 - On

Motion Commands for Modulo Operation

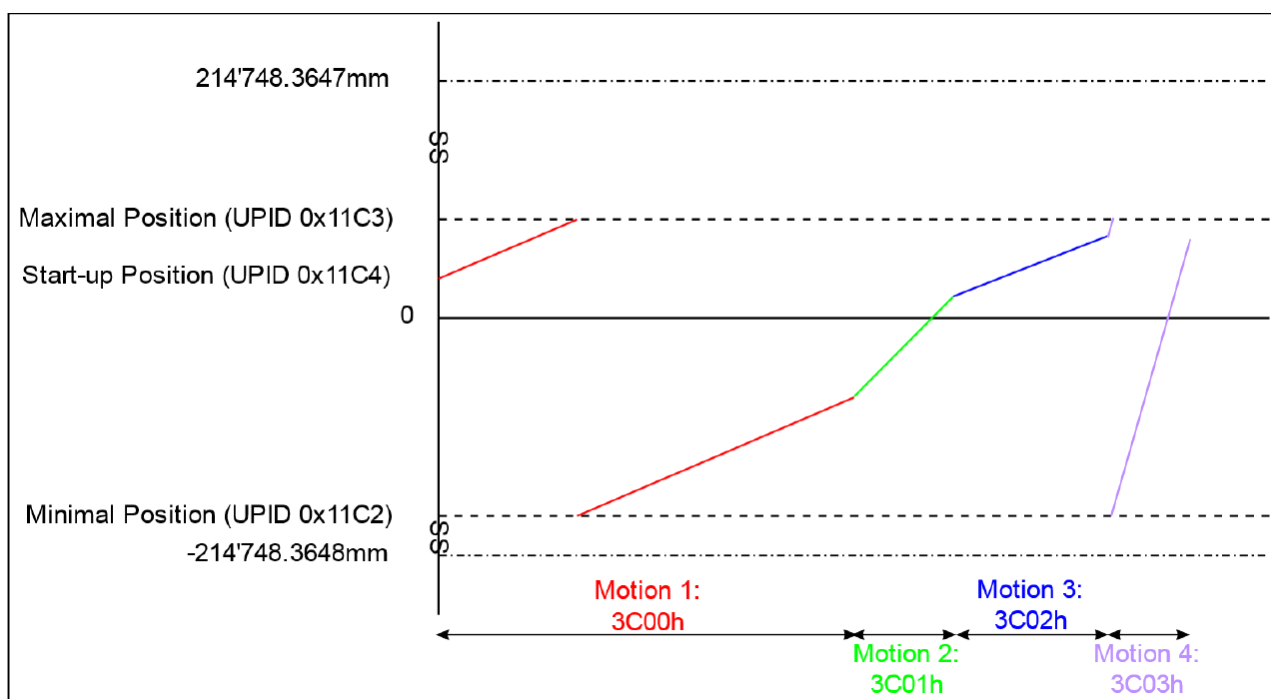
Three motion commands for Modulo operation are defined:

- *3C0xh: VAI Modulo Go To Pos Positive Direction:*
If a new target position is set, the motor turns in positive direction until the target position is reached. Three different cases are possible:
 - The target position is greater than the actual position. In this case the motor turns in positive direction.
 - The target position is smaller than the actual position. In this case the motor turns in positive direction and the variable Modulo Actual Position (0x1B99) variable overflows at Maximal Position (UPID 0x11C3).
 - The target position equals the actual position. In this case the motor will not move.

Example (see figure below):

Maximal Position: +120 mm
Minimal Position: -240 mm
Start-up Position: +50 mm

Motion 1: 3C00h, Target Position = -100 mm → Stroke = 210 mm
Motion 2: 3C01h, Target Position = +30 mm → Stroke = 130 mm
Motion 3: 3C02h, Target Position = +100 mm → Stroke = 70 mm
Motion 4: 3C03h, Target Position = +99 mm → Stroke = 359 mm

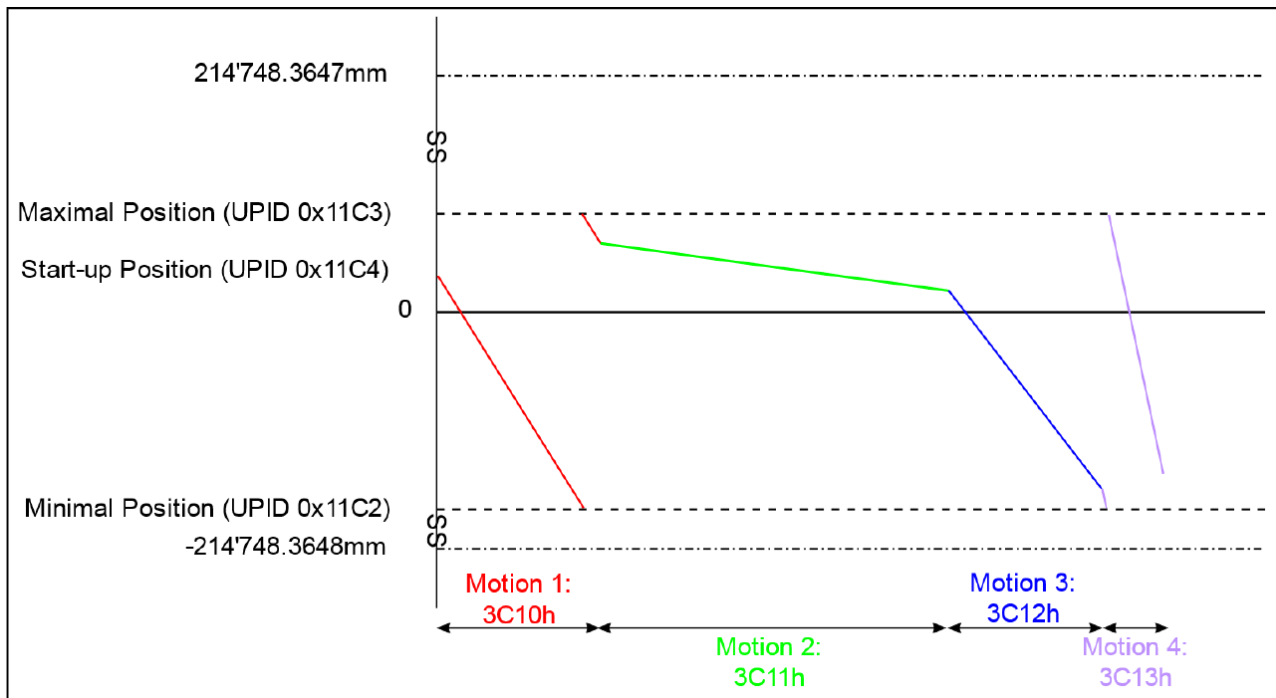


- **3C1xh: VAI Modulo Go To Pos Negative Direction:**
If a new target position is set, the motor turns in negative direction until the target position is reached. Three different cases are possible:
 - The target position is smaller than the actual position. In this case the motor turns in negative direction.
 - The target position is greater than the actual position. In this case the motor turns in negative direction and the variable Modulo Actual Position (UPID 0x1B99) variable underflows at Minimal Position (UPID 0x11C2).
 - The target position equals the actual position. In this case the motor will not move.

Example (see figure below):

Maximal Position: +120 mm
Minimal Position: -240 mm
Start-up Position: +50 mm

Motion 1: 3C10h, Target Position = +100 mm → Stroke = -310 mm
Motion 2: 3C11h, Target Position = +30 mm → Stroke = -70 mm
Motion 3: 3C12h, Target Position = -200 mm → Stroke = -270 mm
Motion 4: 3C13h, Target Position = -190 mm → Stroke = -350 mm

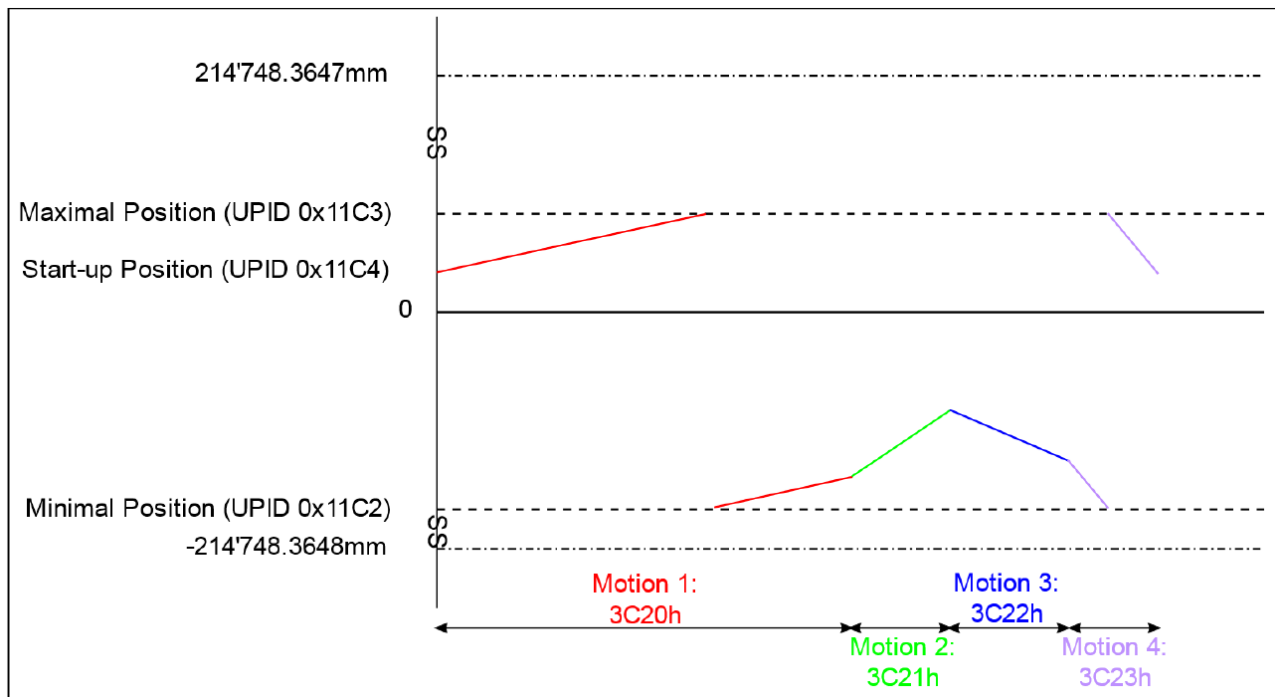


- **3C2xh: VAI Modulo Go To Pos Shortest Way:**
If a new target position is set, the motor turns in the direction of the shortest way.

Example (see figure below):

Maximal Position: +120 mm
Minimal Position: -240 mm
Start-up Position: +50 mm

Motion 1: 3C20h, Target Position = -200 mm → Stroke = 110 mm
Motion 2: 3C21h, Target Position = -120 mm → Stroke = 80 mm
Motion 3: 3C22h, Target Position = -180 mm → Stroke = -60 mm
Motion 4: 3C23h, Target Position = +50 mm → Stroke = -130 mm



9 State Machine Setup

In the following chapter the configuration of the state machine is described. Normally the after installation and running the 'Motor Wizard' there is no more configuration needed.

9.1 Linearizing

In this chapter some system self optimizations are configurable.

9.1.1 Mode

With this parameter the the self tuning mode can be selected.

Parameter Name	UPID	Description
Mode	173Eh	Defines which optimization shall be done if started: <ul style="list-style-type: none">• 0: None• 4: Motor Hall Offset & Amp Correction• 2: System Compensation (generation of a position depening friction and reluctance table)

9.1.2 System Compensation Selection

The parameters in this directory define the setup of the system compensation, which consists of the generation of a position depending friction and reluctance table.

This parameter defines if the correction tables are linear or rotative.

Parameter Name	UPID	Description
Compensation Type	174Fh	Defines the type of the generated compensation tables: <ul style="list-style-type: none">• 0: Linear• 1: Rotative

With this parameter the position controller set for the optimization motion can be selected.

Parameter Name	UPID	Description
Ctrl Par selection	1735h	Defines the postion controller parameter set which is active during the optimization motion: <ul style="list-style-type: none">• 0: Set A• 1: Set B

With this parameter it is defined where the generated compensation tables are stored.

Parameter Name	UPID	Description
Safe in Flash	1736h	Storing location of the generated compensation tables: <ul style="list-style-type: none">• 0: Safe in RAM• 1: Safe in ROM

Select if friction table shall be generated.

Parameter Name	UPID	Description
Friction Table	174Ah	If selected the friction compensation table will be generated

Select if reluctance table shall be generated.

Parameter Name	UPID	Description
Reluctance Table	174Bh	If selected the reluctance compensation table will be generated

9.1.3 Linearizing Config

In this section the motion and with this the also the possible range of the optimization are configurable.

Start position of the linearizing motion. For a rotative system compensation the distance minimal to maximal position should be at least 2 revolutions.

Parameter Name	UPID	Description
Minimal Position	173Fh	Start point of the optimization motion

End position of the linearizing motion. For a rotative system compensation the distance minimal to maximal position should be at least 2 revolutions.

Parameter Name	UPID	Description
Maximal Position	1740h	End point of the optimization motion

Parameter configuration of the Maximal Speed of the VAI motion.

Parameter Name	UPID	Description
Max Speed	1742h	Maximal speed of the optimization motion

Parameter configuration of the Acceleration of the VAI motion.

Parameter Name	UPID	Description
Acceöleration	1743h	Acceleration of the optimization motion

10 Error Code List

An actual error code list can be generated with the LinMot-Talk SW, the procedure to do this, is described in the manual "LinMot-Talk 6 Configuration Software" chapter 2.4 Errors.

Contact & Support

SWITZERLAND

NTI AG

Bodenaeckerstrasse 2
CH-8957 Spreitenbach

Sales and Administration:

+41 56 419 91 91
office@linmot.com

Tech. Support:

+41 56 544 71 00
support@linmot.com
<http://www.linmot.com/support>

Tech. Support (Skype):

support.linmot

Fax:

+41 56 419 91 92

Web:

<http://www.linmot.com>

USA

LinMot USA Inc.

N1922 State Road 120, Unit 1
Lake Geneva, WI 53147
USA

Phone:

262-743-2555

E-Mail:

usasales@linmot.com

Web:

<http://www.linmot-usa.com/>

Please visit <http://www.linmot.com/contact> to find the distribution close to you.

Smart solutions are...

