Engineering Tools

Engineer



9400 function library _____

Reference manual EN

13350243

Lenze

Overview of technical documentation for 9400 Servo Drives

Project planning, selection & order

- 9400 Hardware Manual
- Catalogue / electronic catalogue (DSC - Drive Solution Catalogue)

Mounting & wiring

- MA 9400 HighLine
- MA - communication module
- MA - extension module
- MA - safety module
- MA - accessories
- MA - remote maintenance components

Parameter setting

- BA keypad
- SW - Lenze software »Engineer«
- SW - 9400 HighLine
- KHB - communication module
- SW - extension module
- SW - safety module
- SW - Lenze technology application
- **SW - 9400 function library** ← **This documentation**

Configuring & programming

- SW - Lenze software »Engineer«
- SW - Lenze software »PLC Designer«
- SW - 9400 HighLine
- KHB - communication module
- SW - extension module
- SW - safety module
- SW - Lenze technology application
- **SW - 9400 function library** ← **This documentation**

Drive commissioning

- Commissioning guide
- SW - 9400 HighLine
  - → Chapter "Commissioning"
  - → Chapter "Oscilloscope"
  - → Chapter "Diagnostics & fault analysis"
- Remote maintenance manual

Networking structure

- KHB - communication medium used

Legend:

- Printed documentation
- Online documentation (PDF/Engineer online help)

Abbreviations used:

BA	Operating Instructions
KHB	Communication Manual
MA	Mounting Instructions
SW	Software Manual

Contents

1 About this documentation 12
1.1 Conventions used 14
1.2 Definition of the notes used 15

2 Introduction 16
2.1 How to create applications: 16
2.2 What is a function block? 17
2.3 Parameterisable function blocks 18
2.4 Further objects of an interconnection 20
2.4.1 System block 21
2.4.2 Port block 21
2.4.3 Variable 22
2.4.4 User code 23
2.5 Functional ranges of an interconnection 24
2.6 Conventions used for input/output identifiers 27
2.7 Signal types & scaling 29
2.7.1 Data type check/implicit type conversion 30
2.8 Multitasking in the Servo Drive 9400 32
2.8.1 Runtime measurement 33
2.8.2 Optimising the runtime behaviour 35
2.8.3 Exchanging data between tasks 36

3 Working with the FB editor 37
3.1 Short overview of new functions in the FB Editor 37
3.2 User interface 38
3.2.1 Control and function elements 40
3.2.2 Toolbar 41
3.2.3 Search function 42
3.2.4 Task selection 43
3.2.5 Editor view/overview 44
3.2.6 Context menu 45
3.2.7 Overview window 46
3.3 Using the FB-Editor as "Viewer" 48
3.3.1 Following connections of inputs and outputs 49
3.3.2 Keyboard commands for navigation 50
3.4 Creating/reconfiguring the interconnection 51
3.4.1 Enabling the FB Editor 51
3.4.2 Insert/delete objects 52
3.4.2.1 Inserting a function block 53
3.4.2.2 Inserting a system block 56
3.4.2.3 Inserting a port block 58
3.4.2.4 Creating/inserting a variable 59
3.4.2.5 Configuring/inserting a user code 62
3.4.2.6 Inserting a comment 67
3.4.2.7 Deleting objects that are no longer required 70
3.4.2.8 Deleting variables and user codes from the application 71
3.4.3 Changing connector visibilities 72
3.4.4 Arranging objects in the drawing area 73
3.4.5 Creating/deleting connections 74
3.4.5.1 Creating a connection using the connection line 77
3.4.5.2 Creating a connection using port identifiers 78
3.4.5.3 Creating a connection via connection dialog 79
3.4.5.4 Deleting connections that are no longer required 80
3.4.6 Changing the processing order 81
3.4.7 Configuring the task properties 83

Contents

3.4.8 Configuring the exception handling 84
3.4.9 Compiling the completed interconnection 86
3.4.10 Testing the interconnection in the device 86
3.4.11 Copying interconnection elements (across all devices) 88
3.4.11.1 Insert options for copied elements 90
3.5 Printing the interconnection 92
3.6 Comparing interconnections 93
3.6.1 Comparison examples 94
3.6.2 Contents and representation of the comparison 96
3.7 Copying complete interconnection 97
3.8 Inserting complete interconnection from reference project 98

4 Short overview 100
4.1 Analog signal processing 100
4.2 Digital signal processing 102
4.3 Data type conversion 104
4.4 Mathematical functions 105
4.5 Conversion of physical units 105
4.6 Speed conditioning 106
4.7 Special functions 107
4.8 Output of status signals 108
4.9 FBs for system via electrical shaft 108
4.10 FBs for positioning tasks 111
4.11 FBs for winding technology 112
4.12 All function blocks [A-Z] 113
4.13 FB reference list - 9300 Servo PLC 123
4.14 FB reference list - 9300 servo inverter 131

5 Function blocks 134
5.1 L_CamClutchPos - clutch (path-controlled) 135
5.2 L_CamContactor - cam group with switch-on and switch-off dynamisation 138
5.2.1 Typical application 141
5.2.2 Data flow 142
5.2.3 Switching performance for position cams 143
5.2.4 Switching performance for position-time cams 144
5.2.5 Dynamic displacement of the cams 146
5.2.6 Selection of the cam data 146
5.2.7 Interconnection examples 147
5.3 L_CamCurve - curve interpolation for characteristics 149
5.3.1 Data flow 151
5.3.2 Function 152
5.3.3 Track switch-over 153
5.3.4 Interconnection examples 154
5.4 L_CamGetAxisData - reading out machine parameters from cam data 155
5.5 L_CamPosMarker - reading out position marks from cam data 158
5.5.1 Data flow 161
5.6 L_CamProfiler - curve interpolation for motion profiles 162
5.6.1 Data flow 166
5.6.2 Track switch-over 167
5.6.3 Feedforward control 167
5.6.4 Output of the machine parameters 167
5.7 L_CamSetContDataPDO - contactor data selection via inputs/PDOs 168
5.7.1 Exception handling 171
5.7.2 Interconnection example 171

Contents

5.8 L_CamSetContDataSDO - contactor data selection via codes/SDOs 172
5.8.1 Exception handling 174
5.8.2 Interconnection example 174
5.9 L_CamStretchAbs - stretching/compressing the position (absolute) 175
5.9.1 Typical application 177
5.9.2 Signal characteristics 178
5.9.3 Interconnection example 179
5.10 L_CamStretchFeed - stretching/compressing the position (relative) 180
5.10.1 Typical application 183
5.10.2 Signal characteristics 184
5.10.3 Interconnection example 185
5.11 L_CamSyncIn - synchronous/oversynchronous clutch-in 186
5.11.1 Application example: "Flying saw" 189
5.11.2 Synchronous clutch-in 191
5.11.3 Oversynchronous clutch-in 194
5.11.4 Signal characteristics 195
5.11.5 Interconnection example 196
5.12 L_Dc2BytesToWord - type converter 197
5.13 L_Dc2WordsToDWord- type converter 198
5.14 L_Dc4BytesToDWord - type converter 199
5.15 L_DcBitShift - shift operation 200
5.16 L_DcBitShiftByte - shift operation 201
5.17 L_DcBitShiftDWord - shift operation 202
5.18 L_DcBitShiftInt - shift operation 203
5.19 L_DcBitShiftWord - shift operation 204
5.20 L_DcBitsToByte - bit multiplexer 205
5.21 L_DcBitsToDWord - bit multiplexer 206
5.22 L_DcBitsToWord - bit multiplexer 208
5.23 L_DcByteBitand - bit combination 209
5.24 L_DcByteBitor - bit combination 210
5.25 L_DcByteBitxor - bit combination 211
5.26 L_DcByteToBits - bit demultiplexer 212
5.27 L_DcByteToInt - type converter 213
5.28 L_DcByteToWord - type converter 214
5.29 L_DcDIntToDWord - type converter 215
5.30 L_DcDIntToInt - type converter 216
5.31 L_DcDWordBitand - bit combination 217
5.32 L_DcDWordBitor - bit combination 218
5.33 L_DcDWordBitxor - bit combination 219
5.34 L_DcDWordTo2Words - type converter 220
5.35 L_DcDWordTo4Bytes - type converter 221
5.36 L_DcDWordToBits - bit demultiplexer 222
5.37 L_DcDWordToDInt - type converter 224
5.38 L_DcDWordToWord - type converter 225
5.39 L_DcGetBitOfByte - bit operation 226
5.40 L_DcGetBitOfDWord - bit operation 227
5.41 L_DcGetBitOfWord - bit operation 228
5.42 L_DcIntToDInt - type converter 229
5.43 L_DcIntToWord - type converter 230
5.44 L_DcNorm_aToNorm_n - signal converter 231
5.45 L_DcNorm_nToNorm_a - signal converter 232
5.46 L_DcNorm_nToSpeed_s - signal converter 233
5.47 L_DcResetBitOfByte - bit operation 234
5.48 L_DcResetBitOfDWord - bit operation 235
5.49 L_DcResetBitOfWord - bit operation 236

Contents

5.50 L_DcSetBitOfByte - bit operation 237
5.51 L_DcSetBitOfDWord - bit operation 238
5.52 L_DcSetBitOfWord - bit operation 239
5.53 L_DcSpeed_sToNorm_n - signal converter 240
5.54 L_DcSpeed_sToSpeed_v - signal converter 241
5.55 L_DcSpeed_vToSpeed_s - signal converter 242
5.56 L_DcWordBitand - bit combination 243
5.57 L_DcWordBitor - bit combination 244
5.58 L_DcWordBitxor - bit combination 245
5.59 L_DcWordTo2Bytes - type converter 246
5.60 L_DcWordToBits - bit demultiplexer 247
5.61 L_DcWordToByte - type converter 248
5.62 L_DcWordToDWord - type converter 249
5.63 L_DcWordToInt - type converter 250
5.64 L_DevApplErr - error tripping 251
5.65 L_DevApplErrFix - error tripping 253
5.66 L_DevBrakeStateDecoder - status signals of the basic function "Brake control" 255
5.67 L_DevDriveInterfaceStateDecoder - status signals of the drive interface 257
5.68 L_DevHomingStateDecoder - status signals of the basic function "Homing" 259
5.69 L_DevLimiterStateDecoder - status signals of the basic function "Limiter" 261
5.70 L_DevManualJogStateDecoder - status signals of the basic function "ManualJog" 263
5.71 L_DevParReadFix - read parameter 265
5.71.1 Example 268
5.72 L_DevParWriteFix - write parameter 271
5.73 L_DevPositionerStateDecoder - positioning status signals 275
5.74 L_DevReadParDInt - reading parameters of the DINT type 277
5.75 L_DevSMControlDecoder - control signals from the safety module 280
5.76 L_DevSMControlEncoder - control word for limiter 282
5.77 L_DevSMStateDecoder - status signals from the safety module 284
5.78 L_DevSMStateDecoderIO - status signals from the safety module 286
5.79 L_DevWriteParDInt - writing to parameters of the DINT type 287
5.80 L_EsClutchPos - clutch (time-controlled) 290
5.80.1 Signal characteristics 295
5.80.2 Interconnection example 297
5.81 L_EsEncoderConv - encoder signal conditioning 298
5.81.1 Typical application 301
5.81.2 Encoder settings 301
5.81.3 64 bit encoder signal evaluation 302
5.81.4 Signal characteristics 303
5.82 L_EsStretchIntegrate - synchronously stretching/compressing the master value 304
5.82.1 Signal characteristics 307
5.83 L_LdAddOffsetCyclic - offset addition 308
5.84 L_LdClutchAxisP - virtual clutch for synchronism 309
5.85 L_LdClutchV - virtual clutch for electronic gearboxes 312
5.86 L_LdConvAxisV - speed ratio 315
5.87 L_LdDifferentiateCyclic - cyclic differentiation 318
5.88 L_LdExtrapolate - extrapolation 319
5.88.1 Function 321
5.89 L_LdIntegrateCyclic - cyclic integration 322
5.90 L_LdLinearCoupling - master value connection 324
5.90.1 Interconnection example 326
5.91 L_LdMarkSync - mark synchronisation 327
5.91.1 Typical application 330
5.91.2 Signal characteristics 332
5.91.3 Interconnection example 333

Contents

5.92 L_LdMonitFollowError - following error monitoring 335
5.93 L_LdMPot - master value adjustment 337
5.93.1 Monitoring for counter overflow 339
5.94 L_LdPosCtrlLin - Master value adjustment via position 340
5.94.1 Monitoring for counter overflow 345
5.94.2 Interconnection examples 346
5.94.2.1 Printing roller trimming via x offset 346
5.94.2.2 Time-based compensation of mark correction 350
5.95 L_LdSetAxisVelocity - master value processing 354
5.96 L_LdStateDecoder - LineDrive status signals 358
5.97 L_LdToolControl - setpoint conditioning 360
5.98 L_LdSyncOperation - master value connection 363
5.98.1 Operating mode 0: Coupling via feed constants 368
5.98.2 Operating mode 1: Coupling via cycles 369
5.98.3 Operating mode 2: Free coupling 370
5.98.4 Angular trimming via x offset 371
5.99 L_LdVirtualMasterP - virtual master for synchronism 373
5.99.1 Functions 377
5.99.2 Monitoring for counter overflow 378
5.100 L_LdVirtualMasterV - virtual master for electronic gearboxes 379
5.100.1 Functions 382
5.100.2 Monitoring for counter overflow 383
5.101 L_LdZeroDetect - zero crossing detection 384
5.102 L_PosDecoderStatePositioner - positioning status signals 385
5.103 L_PosDecoderStateSequencer - status signals of the sequence control 387
5.104 L_PosGetProfile - profile data tables 388
5.105 L_PosGetProfileData - profile data output 389
5.106 L_PosGetTableAcc - acceleration table 392
5.107 L_PosGetTableJerk - S-ramp time table 394
5.108 L_PosGetTablePos - position table 395
5.109 L_PosGetTableSpeed - speed table 396
5.110 L_PosPositionerInterface - positioning interface 397
5.110.1 Interconnection 398
5.110.2 Applications 399
5.111 L_PosPositionerTable - profile data record management 400
5.111.1 Variable tables 407
5.111.2 "Teach" function 408
5.111.3 Positioning modes 409
5.111.4 Positioning with final speed 411
5.111.5 Touch probe positioning 412
5.111.6 Monitoring for counter overflow 413
5.112 L_PosProfileInterface - profile data interface 414
5.112.1 Positioning modes 416
5.112.2 Positioning with final speed 418
5.112.3 Touch probe positioning 419
5.112.4 Touch probe configuration 419
5.113 L_PosProfileTable - profile data record management (simple) 420
5.113.1 Profile data table 424
5.113.2 "Teach" function 425
5.113.3 Positioning modes 426
5.113.4 Touch probe positioning 428
5.113.5 Monitoring for counter overflow 429

Contents

5.114 L_PosSequencer - sequence control 430
5.114.1 Priorities of the control signals 436
5.114.2 States of the sequencer 437
5.114.3 Action types available for creating the positioning program 438
5.114.3.1 Action type "Program end" 439
5.114.3.2 Action type "Positioning" 440
5.114.3.3 Action type "Switching" 442
5.114.3.4 Action type "Branching" 443
5.114.3.5 Action type "Variable branching" 444
5.114.3.6 Action type "Homing" 445
5.114.3.7 Action type "Waiting" 446
5.114.3.8 Action type "Counter setting" 447
5.114.3.9 Action type "Counting" 448
5.114.3.10 Action type "Stand-by" 449
5.114.4 Example: Sequence table 450
5.115 L_PosSequencerStateDecoder - status signals of the sequence control 451
5.116 L_SdAccToUnit - acceleration conversion 452
5.117 L_SdDelayComp - dead time compensation 453
5.118 L_SdDifferentiate - differentiation 454
5.119 L_SdFactor - stretch factor 455
5.120 L_SdGetAxisData - Reading out machine parameters from axis data 456
5.121 L_SdGetPosition - position conversion 458
5.122 L_SdGetSpeed - speed conversion 459
5.123 L_SdIntegrate - integration of speed to position 460
5.124 L_SdIntegrateAxis - speed to position integration (with TP correction) 461
5.124.1 Typical application 463
5.124.2 Interconnection of the touch probe inputs 464
5.124.3 Signal characteristics 465
5.125 L_SdIntegrateLimit - integration of speed to position 467
5.125.1 Basics 469
5.125.2 Signal characteristics 470
5.126 L_SdInterExtrapolateAny - signal interpolation 471
5.126.1 Extrapolation function 472
5.126.2 Interpolation function 472
5.126.3 Signal characteristics 472
5.127 L_SdInterExtrapolatePosition - signal interpolation 474
5.127.1 Extrapolation function 476
5.127.2 Interpolation function 476
5.127.3 Telegram failure detection 477
5.127.4 Signal characteristics 477
5.128 L_SdInterpolate - signal interpolation 483
5.128.1 Function 484
5.129 L_SdLimitSpeed - speed limitation 486
5.130 L_SdMotorPot - motor potentiometer 488
5.130.1 Acceleration and deceleration time 490
5.130.2 Control signals 490
5.130.3 Signal characteristics 491
5.130.4 Deactivation function 492
5.130.5 Behaviour for incorrect parameterisation of the limit values 494
5.131 L_SdPosToUnit - position conversion 495
5.132 L_SdProcessController - PID controller with limitation 496
5.132.1 Controller characteristic 500
5.132.2 Adaptation of the controller gain via characteristic function 501
5.132.3 System deviation in the area of reduced sensitivity 502
5.132.4 Controller influence 502

Contents

5.132.5 Ramp function generator 503
5.132.6 Low pass in actual value path 503
5.133 L_SdRampGenerator - ramp function generator - S-ramp 504
5.133.1 Loading the ramp function generator 505
5.133.2 Acceleration and jerk 506
5.134 L_SdRampGeneratorAny - ramp function generator - S-ramp 507
5.135 L_SdRuntimeComp - runtime compensation 509
5.136 L_SdSetAxisData - machine parameters 510
5.136.1 Typical application 512
5.136.2 Selection and input of machine parameters 513
5.136.3 Resolution of an encoder revolution 514
5.136.3.1 Detecting the optimum resolution 516
5.136.4 Example 517
5.137 L_SdSetPosition - position conversion 518
5.137.1 Monitoring for counter overflow 519
5.138 L_SdSetSpeed - speed conversion 520
5.138.1 Monitoring for counter overflow 521
5.139 L_SdSpeedFilter - speed signal delay 522
5.139.1 Monitoring for counter overflow 523
5.140 L_SdSpeedSet - setpoint ramp generator 524
5.140.1 Setpoint path 528
5.140.2 Fixed setpoints 528
5.140.3 Ramp generator 529
5.140.4 Setting and selection of the Ti times 530
5.140.5 Display of the machine-based variables 530
5.141 L_SdSpeedToUnit - speed conversion 531
5.142 L_SdSwitchPoint - position switch points 532
5.142.1 Monitoring for counter overflow 534
5.143 L_SdTouchProbe - touch probe evaluation 535
5.143.1 Actual value interpolation (principle) 536
5.143.2 Example 536
5.144 L_SdUnitToAcc - acceleration conversion 537
5.145 L_SdUnitToPos - position conversion 538
5.146 L_SdUnitToSpeed - speed conversion 539
5.147 L_Tb5And - AND with 5 inputs 540
5.148 L_Tb5Nand - NAND with 5 inputs 541
5.149 L_Tb5Nor - NOR with 5 inputs 542
5.150 L_Tb5Or - OR with 5 inputs 543
5.151 L_Tb5Xor - XOR with 5 inputs 544
5.152 L_Tb8Select - 1-out-of-8 selector (for data type "DINT") 545
5.153 L_Tb8SelectByte - 1-out-of-8 selector (for data type "BYTE") 546
5.154 L_Tb8SelectWord - 1-out-of-8 selector (for data type "WORD") 547
5.155 L_TbAbs - absolute value 548
5.156 L_TbAdd - addition 549
5.157 L_TbAddLim - addition with limitation 550
5.158 L_TbAddSubLim - addition and subtraction with limitation 551
5.159 L_TbAnd - AND with 2 inputs 552
5.160 L_TbCompare - comparison 553
5.160.1 Function 1: dnIn1 = dnIn2 554
5.160.2 Function 2: dnIn1 > dnIn2 555
5.160.3 Function 3: dnIn1 < dnIn2 556
5.160.4 Function 4: |dnIn1| = |dnIn2| 557
5.160.5 Function 5: |dnIn1| > |dnIn2| 557
5.160.6 Function 6: |dnIn1| < |dnIn2| 557

Contents

5.161 L_TbCompare_n - scaled comparison 558
5.161.1 Function 1: dnIn1_n = dnIn2_n 559
5.161.2 Function 2: dnIn1_n > dnIn2_n 560
5.161.3 Function 3: dnIn1_n < dnIn2_n 561
5.161.4 Function 4: |dnIn1_n| = |dnIn2_n| 562
5.161.5 Function 5: |dnIn1_n| > |dnIn2_n| 562
5.161.6 Function 6: |dnIn1_n| < |dnIn2_n| 562
5.162 L_TbCount - up/downcounter 563
5.162.1 Function 1: Automatic counter reset 564
5.162.2 Function 2: Counter stop at upper limit 564
5.163 L_TbCurve - characteristic function 565
5.163.1 Selection of the characteristic 566
5.163.2 Characteristic mode 0: Entire range 568
5.163.3 Characteristic mode 1: Symmetrically to y axis 569
5.163.4 Characteristic mode 2: Symmetrically to zero point 569
5.164 L_TbCurveAny - Characteristic function 570
5.164.1 Selection of the characteristic 571
5.164.2 Characteristic mode 0: Entire range 573
5.164.3 Characteristic mode 1: Symmetrically to y axis 574
5.164.4 Characteristic mode 2: Symmetrically to zero point 574
5.165 L_TbDeadband - dead band with gain 575
5.166 L_TbDeadband_n - scaled dead band with gain 577
5.167 L_TbDelay - delay 579
5.167.1 Function 1: ON-delay 580
5.167.2 Function 2: OFF-delay 580
5.167.3 Function 3: General delay 581
5.168 L_TbDifferentiate - differentiator with low-pass filter 582
5.168.1 Function 583
5.168.2 Adapting the time reference 584
5.169 L_TbDiv - division 585
5.170 L_TbDiv_n - scaled division 586
5.171 L_TbFlipFlopD - D flipflop 587
5.172 L_TbFlipFlopRS - status-controlled RS flipflop 589
5.173 L_TbGainLim - gain with limitation 590
5.174 L_TbIntegrate - integration with limitation 591
5.174.1 Adapting the time reference 592
5.175 L_TbLimit - limitation (for "DINT" data type) 593
5.176 L_TbLimit_n - scaled limitation 594
5.177 L_TbLimitInt - limitation (for "INT" data type) 595
5.178 L_TbMaskOut - zone masking 596
5.179 L_TbMul - multiplication 598
5.180 L_TbMul_n - scaled multiplication with limitation 599
5.181 L_TbMulDivLim - multiplication and division with limitation 600
5.182 L_TbMulLim - multiplication with limitation 601
5.183 L_TbNand - NAND with 2 inputs 602
5.184 L_TbNeg - negation (for "DINT" data type) 603
5.185 L_TbNegInt - negation (for "INT" data type) 604
5.186 L_TbNegSel - optional negation (for "DINT" data type) 605
5.187 L_TbNegSelInt - optional negation (for "INT" data type) 606
5.188 L_TbNor - NOR with 2 inputs 607
5.189 L_TbNormalize - signal scaling with limitation 608
5.190 L_TbNot - negation 609
5.191 L_TbOr - OR with 2 inputs 610

Contents

5.192 L_TbOscillator - rectangular signal generator 611
5.192.1 Function 611
5.192.2 Optimising the accuracy of the output signal 612
5.193 L_TbPIController - PI controller with limitation 613
5.194 L_TbPT1Filter - delay 615
5.195 L_TbRateAction - rate action with limitation 616
5.196 L_TbSampleHold - sample & hold (for "DINT" data type) 618
5.197 L_TbSampleHoldWord - sample & hold (for "WORD"data type) 619
5.198 L_TbSelect - selector (for "DINT" data type) 620
5.199 L_TbSelectByte - selector (for "BYTE" data type) 621
5.200 L_TbSelectWord - selector (for "WORD" data type) 622
5.201 L_TbSub - subtraction 623
5.202 L_TbSubLim - subtraction with limitation 624
5.203 L_TbTransition - edge evaluation 625
5.203.1 Function 1: Evaluating rising edges 626
5.203.2 Function 2: Evaluating falling edges 626
5.203.3 Function 3: Evaluating rising and falling edges 627
5.204 L_TbXor - XOR with 2 inputs 628
5.205 L_WndCalcDiameter - diameter calculation 629
5.205.1 Setting an initial value 632
5.205.2 Calculating the diameter 632
5.205.3 Holding the current value 632
5.205.4 Selecting the direction of change/web break monitoring 633
5.205.5 Limit value monitoring 633
5.206 L_WndCalcLength - length calculation 634
5.206.1 Display of the length/residual length/residual runtime reached 636
5.206.2 Activating the braking process 636
5.206.3 Triggering a limit stop 636
5.206.4 Resetting the calculation & trigger signals 636
5.207 L_WndFrictionCompensation - friction compensation 637
5.207.1 Basics: Friction response 640
5.207.2 Basic interconnection 640
5.207.3 Identifying the friction characteristic automatically 641
5.207.4 Activating the friction compensation 643
5.208 L_WndIdentMInertia - identification of the moment of inertia 644
5.208.1 Basic interconnection 646
5.208.2 Carrying out the identification 646
5.209 L_WndStopCtrl - stop controller 649
5.209.1 Display of the residual length/residual runtime 651
5.209.2 Activating the braking process 651
5.209.3 Triggering a limit stop 651
5.209.4 Resetting the calculation & trigger signals 651
5.210 L_WndTensionCurve - tensile force characteristic 652
5.210.1 Selection of the characteristic 653
5.210.1.1 Characteristic for a linear course of the tensile force 653
5.210.1.2 Characteristic for a linear course of the torque 654
5.210.1.3 Course of the tensile force according to a specified characteristic 655

6 Table of attributes 656

Index 670

Your opinion is important to us 681

1 About this documentation

This documentation contains information on the Lenze function blocks which can be used in the L-force »Engineer« for the Servo Drives 9400.

The function block editor is available in the »Engineer« HighLevel for Servo Drives 9400 with the runtime software licence Motion Control HighLevel or higher.

Target group

This documentation is intended for all persons who want to reconfigure the application interconnection of the controller with the function block editor and extend it by individual functions.

Document history

Version			Description
6.7	08/2014	TD05	Error corrections
6.6	02/2013	TD05	Amendments & error corrections
6.5	12/2012	TD05	New layout
6.4	04/2012	TD05	Amendments & error corrections
6.3	11/2011	TD05	Amendments & error corrections
6.2	09/2011	TD05	Amendments & error corrections xtended by new function block: • LenzeToolbox.lib: L_TbCurveAny
6.1	06/2011	TD05	Amendments & error corrections
6.0	03/2011	TD05	Extended by "LenzeWinder.lib" function library with function blocks for winding technology: L_WndCalcDiameter, L_WndCalcLength, L_WndFrictionCompensation, L_WndIdentMInertia, L_WndStopCtrl, L_WndTensionCurve Amendments & error corrections
5.6	10/2010	TD05	Amendments & error corrections
5.5	03/2010	TD05	Amendments & error corrections Extended by new function blocks: • LenzeServoDrive.lib: L_SdIntegrateLimit, L_SdInterExtrapolateAny, L_SdInterExtrapolatePosition, L_SdRampGeneratorAny
5.4	02/2010	TD05	Error corrections
5.3	08/2009	TD05	Error corrections
5.2	08/2009	TD05	Amendments & error corrections Extended by new function block: • LenzeServoDrive.lib: L_SdGetAxisData
5.1	03/2009	TD05	New subchapter "Multitasking in the Servo Drive 9400" Extension of the main chapter "Working with the FB editor" by new functions of the »Engineer« V2.10
5.0	11/2008	TD05	Extension of the main chapter "Working with the FB editor" by new functions of the »Engineer« V2.9 Extended by new function blocks: • LenzeCam.lib: L_CamSetContDataPDO, L_CamSetContDataSDO • LenzeDataConversion.lib: L_DcBitShiftByte, L_DcBitShiftDWord, L_DcBitShiftInt, L_DcBitShiftWord, L_DcByteBitand, L_DcByteBitor, L_DcByteBitxor, L_DcByteToInt, L_DcWordBitand, L_DcWordBitor, L_DcWordBitxor • LenzeElectricalShaft.lib: L_EsClutchPos • LenzeToolbox.lib: L_Tb8SelectByte, L_Tb8SelectWord, L_TbLimitInt, L_TbNegInt, L_TbNegSelInt, L_TbSampleHoldWord, L_TbSelectByte, L_TbSelectWord New: Table of attributes with information required for a communication to the controller via parameters.

Version			Description
4.1	04/2008	TD05	Extended by new function block: • LenzeCam.lib: L_CamSyncIn
4.0	03/2008	TD05	New main chapter "Working with the FB editor" Extended by new function block: • LenzeServoDrive.lib: L_SdProcessController
3.0	09/2007	TD05	Extended by new function blocks: • LenzeCam.lib: L_CamClutchPos, L_CamContactor, L_CamCurve, L_CamGetAxisData, L_CamPosMarker, L_CamProfiler, L_CamStretchAbs, L_CamStretchFeed • LenzeElectricalShaft.lib: L_EsEncoderConv, L_EsStretchIntegrate • LenzeServoDrive.lib: L_SdIntegrateAxis, L_SdMotorPot • LenzeToolbox.lib: L_TbAddSubLim, L_TbMaskOut
2.1	03/2007	TD05	Extended by new function blocks: • LenzeDevice9400.lib: L_DevApplErr, L_DevApplErrFix, L_DevParReadFix, L_DevParWriteFix • LenzeServoDrive.lib: L_SdAccToUnit, L_SdSpeedToUnit, L_SdUnitToAcc • LenzeToolbox.lib: L_TbMulDivLim, L_TbNegSel
2.0	10/2006	TD05	Extended by new function blocks: • LenzeDevice9400.lib: L_DevPositionerStateDecoder, L_DevReadParDInt, L_DevWriteParDInt, L_DevSMControlDecoder, L_DevSMControlEncoder, L_DevSMStateDecoder, L_DevSMStateDecoderIO • LenzePositioning.lib: L_PosPositionerInterface, L_PosSequencerStateDecoder • LenzeServoDrive.lib: L_SdInterpolate
1.4	08/2006	TD05	First edition

1.1 Conventions used

This documentation uses the following conventions to distinguish between different types of information:

Type of information	Writing	Examples/notes
Spelling of numbers		
Decimal separators	Point	The decimal point is generally used. For example: 1234.56
Text		
Version information	Text colour blue	Information that is only valid for or from a certain software version of the controller is marked accordingly in this manual. Example: This function extension is available from software version V3.0!
Program name	» «	The Lenze »Engineer« PC software...
Window	*italics*	The *Message* window ... / The *Options* dialog box...
Variable identifier		Set *bEnable* to TRUE to...
Control element	**bold**	The **OK** button... / The **Copy** command... / The **Properties** tab... / The **Name** input field...
Sequence of menu commands		If the execution of a function requires several commands, the individual commands are separated by an arrow: Select **File→Open** to...
Shortcut	**<bold>**	Press **<F1>** to open the online help.
		If a command requires a combination of keys, a "+" is placed between the key symbols: Use **<Shift>+<ESC>** to...
Program code	`Courier`	`IF var1 < var2 THEN` `  a = a + 1` `  END IF`
Keyword	**`Courier bold`**	
Hyperlink	underlined	Optically highlighted reference to another topic. In this documentation activated by mouse-click.
Symbols		
Page reference	(📖 14)	Optically highlighted reference to another page. In this documentation activated by mouse-click.
Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

1.2 Definition of the notes used

This documentation uses the following signal words and symbols to indicate dangers and important information:

Safety instructions

Layout of the safety instructions:

Danger!

(characterises the type and severity of danger)

Note

(describes the danger and gives information about how to prevent dangerous situations)

Pictograph	Signal word	Meaning
	**Danger!**	**Danger of personal injury through dangerous electrical voltage** Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
	**Danger!**	**Danger of personal injury through a general source of danger** Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
	**Stop!**	**Danger of property damage** Reference to a possible danger that may result in property damage if the corresponding measures are not taken.

Application notes

Pictograph	Signal word	Meaning
	**Note!**	Important note to ensure trouble-free operation
	**Tip!**	Useful tip for easy handling
		Reference to another documentation

2 Introduction

2.1 How to create applications:

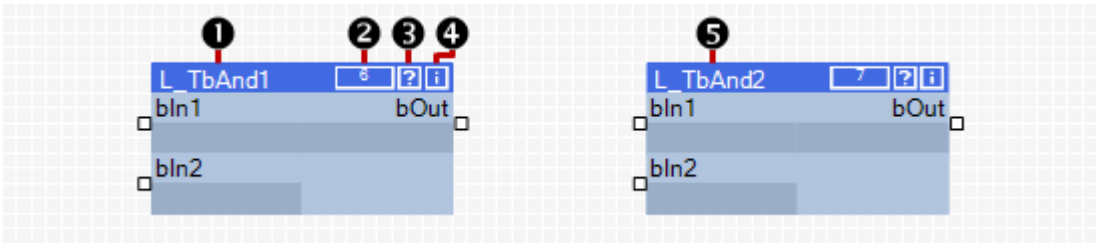
An application is the configured function of a device. To create an application for the controller, there are two solutions:

1. You use a ready-made technology application supplied by Lenze.
   - For this purpose, simply select the required technology application from the catalogue in the »Engineer« for the controller.
   - The selected technology application can be changed and extended by further functions, if required.
2. You create the application completely by yourself using the function block editor.
   - For this purpose, select no application for the controller. In this case, an "empty application" is assigned to the controller.

2.2 What is a function block?

A function block (FB) can be compared with an integrated circuit that contains a specific control logic and delivers one or several values when being executed.

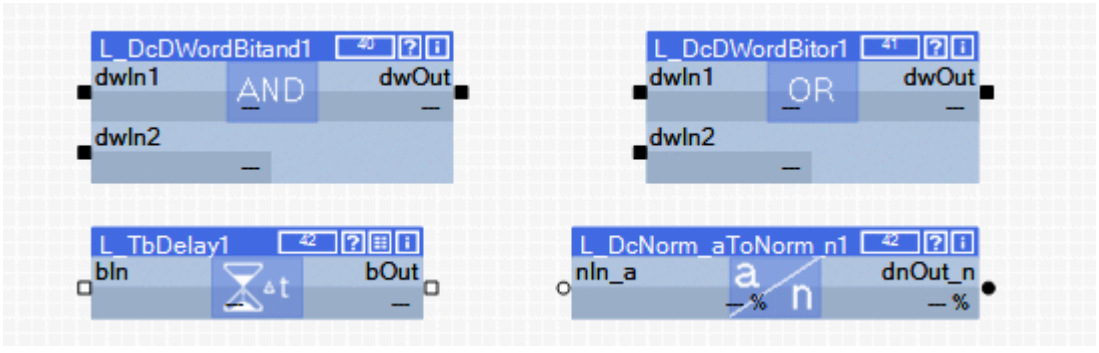
- Function blocks are classified in function libraries according to their functions.
- An instance (reproduction, copy) of the function block is always inserted into the interconnection.
- It is also possible to insert several instances of a function block into an interconnection.
- Each instance has an unequivocal identifier (the instance name) and an order index which defines the position at which the function block is calculated during runtime.



❶ Name of the instance of the function block (freely selectable)
❷ Order index (is simultaneously the button for changing the order index)
❸ Button for opening the online help for the function block
❹ Button for displaying/changing the function block properties
❺ Other instance of the function block

[2-1] Information on a function block in the function block editor

As of »Engineer« V2.9 icons are displayed for simple functions blocks which symbolise their basic functions to ensure a better readability of the interconnection.



[2-2] Display of symbolic icons for simple function blocks

Tip!

An overview of all available function blocks is provided in the chapter "Short overview". (📖 100)

A detailed description for each function block is provided in the chapter "Function blocks". (📖 134)

2.3 Parameterisable function blocks

Some function blocks have parameters which serve to change particular settings during operation, if required, or which display actual values & status information.

Parameters, codes & subcodes

All parameters are stored in "codes".

- The codes are numbered and indicated in the documentation with a preceding "C", e.g. "C00002".
- For the sake of clarity some codes contain "subcodes" in which parameters are stored. In the documentation, code and subcode are divided by a "/", e.g. "C00118/3".

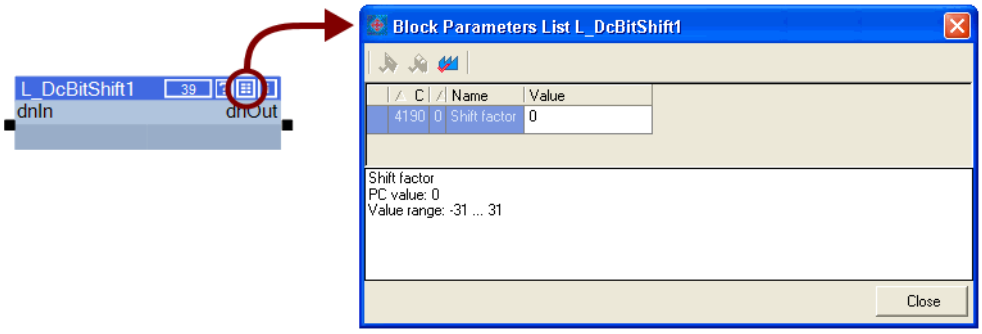
Code ranges

The codes for the 9400 Servo Drives are divided into the following ranges:

Assignment	Range	Contents
Firmware	C00000 ... C02999	Codes of the standard device
TA	C03000 ... C03099	Basic codes of the technology application
	C03100 ... C03499	Multiplexer codes of the technology application
	C03500 ... C03999	Codes of used function blocks
Libraries	C04000 ... C04199	LenzeToolbox
	C04200 ... C04399	LenzeServoDrive
	C04400 ... C04499	LenzeDataConversion
	C04500 ... C04799	LenzePositioning
	C04800 ... C04999	LenzeWinder
	C05000 ... C05299	LenzeLineDrive
	C05300 ... C05599	LenzeCam
	C05600 ... C05899	- Reserve -
	C05900 ... C05999	LenzeDevice9400
User	C06000 ... C08999	Free range for user codes
	C09000 ... C12999	- Reserve -
Modules	C13000 ... C13999	Extension/communication module in the MXI1 module receptacle
	C14000 ... C14999	Extension/communication module in the MXI2 module receptacle
	C15000 ... C15999	Safety module in the MSI module receptacle for safety engineering
	C16000 ...	- Reserve -

Parameter list

- By clicking the icon the parameter list of a function block can be opened.



Wildcard "unit" for the real unit of the machine

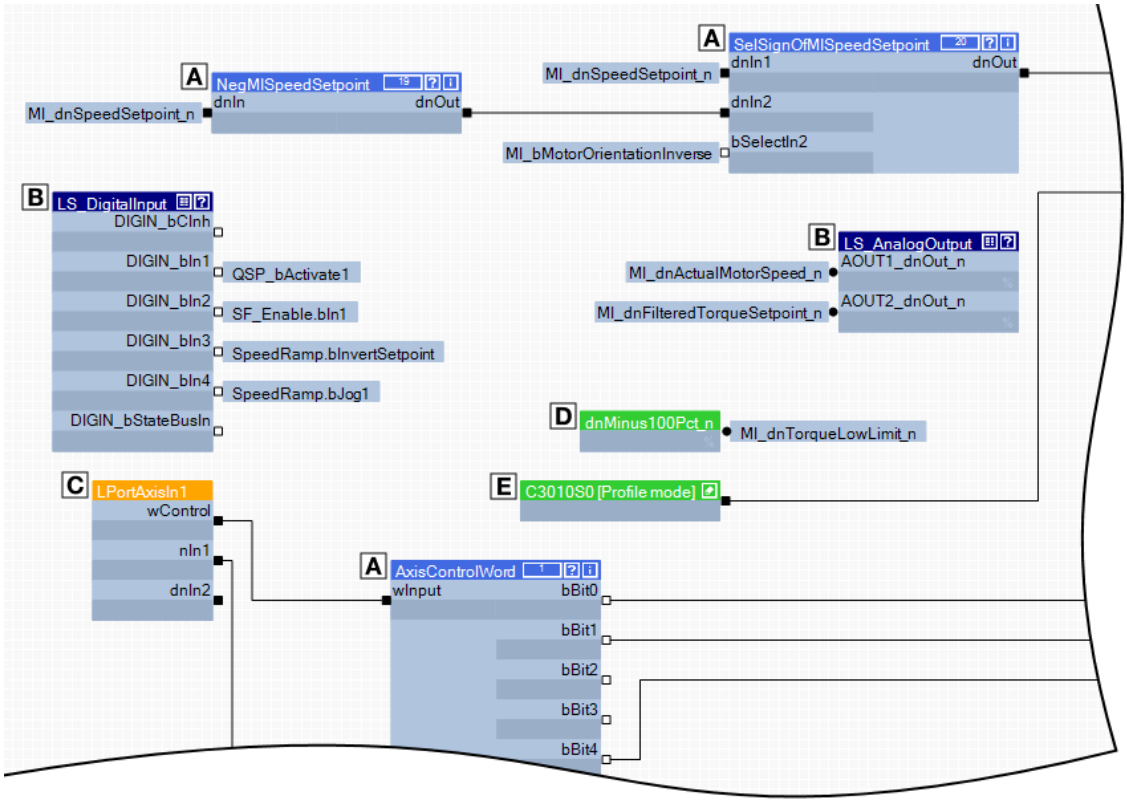
The parameters for a travel profile (e.g. position, speed, acceleration and deceleration) are selected in real units with regard to the slide, e.g. 1000 mm as relative position.

- The machine parameters of the drive interface (system block **LS_DriveInterface**) are used to define the real unit.
- This documentation uses the term "unit" in the parameter unit information only as a wildcard for the real unit of the machine.

Resolution	Value range	Number of decimal positions
32 bits	± 214748.3647	4

2.4 Further objects of an interconnection

In addition to function blocks, an interconnection also contains system blocks which are required for accessing the basic functions of the controller and the physical interfaces (e.g. digital inputs). Furthermore, an interconnection can contain port blocks as well as variables and user codes.

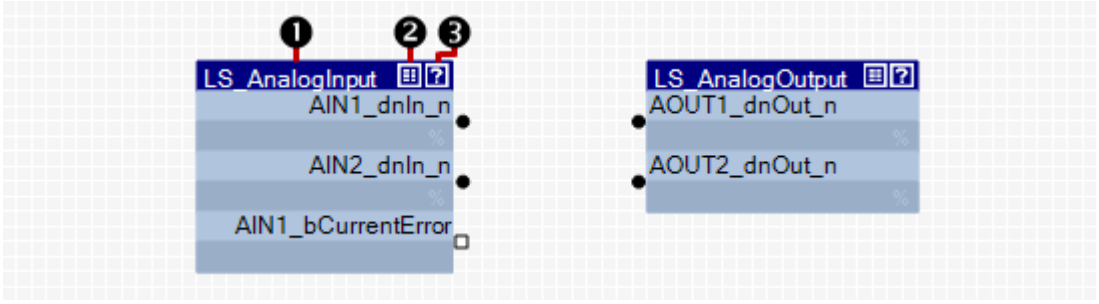


A Function block B System block C Port block D Variable E User code

2.4.1 System block

System blocks can be considered as a specific variant of a function block.

- In contrast to function blocks, system blocks are firmly integrated in the runtime system of the controller and therefore are not instanceable.
- System blocks partly activate real hardware, e. g. the digital and analog inputs/outputs and the motor control.
- The system blocks which are provided for a controller in the function block editor depend on the respective device type.



❶ Name of the system block
❷ Button for opening the parameterisation dialog or the parameter list for the system block
❸ Button for opening the online help for the system block

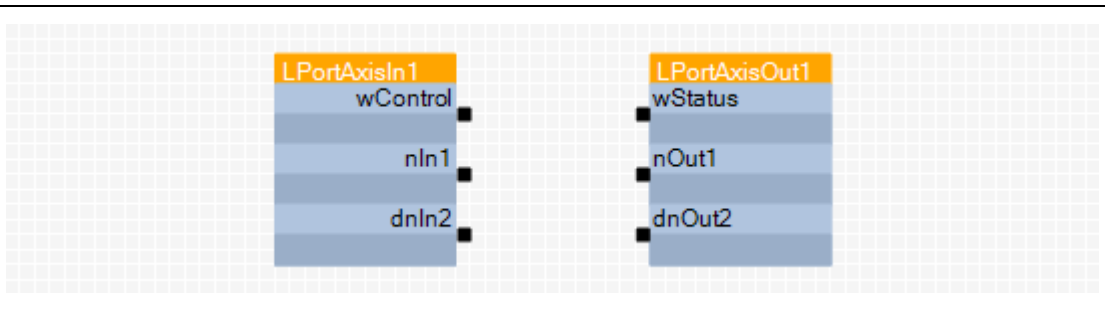
[2-3] Example: System blocks "LS_AnalogInput" and "LS_AnalogOutput" for representing the analog inputs/outputs

Tip!

The system blocks are described in detail in the software manual and in the online help for the controller.

2.4.2 Port block

All input/output ports of the application can be inserted in the interconnection in the form of port blocks, in order to receive access to the associated element variables.



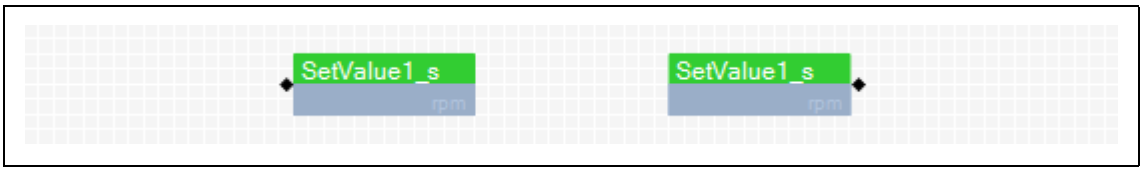
[2-4] Example: Input port "LPortAxisIn1" and output port "LPortAxisOut1"

Input/output ports enable, for instance, the data exchange in an interconnection or with a master control via a corresponding transmission medium (e.g. system bus).

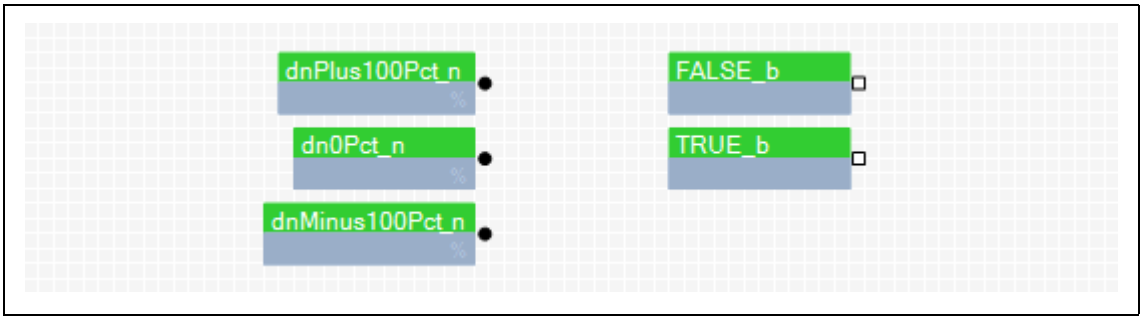
2.4.3 Variable

Variables can be used for a cross-task value exchange and for the use of constant values. The use of significantly named variables can make the interconnection more clear and easier to maintain.

- If you assign the "safe against mains failure" property to a variable, the value of the variable remains stored even after mains switching.
- A variable can be inserted exactly once with an input (for value assignment) into the interconnection and as often as required with an output (for read access).



[2-5] Example 1: Variable "SetValue1_s" on the left with input for value assignment and on the right with output for reading access



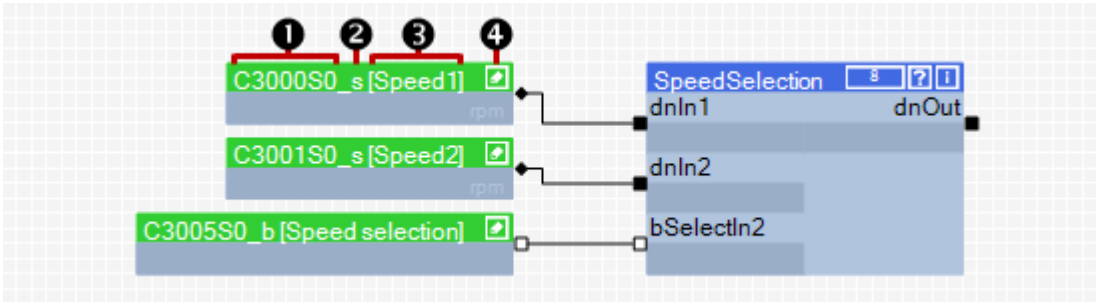
[2-6] Example 2: Variables (with output) for the constant values 100 %, 0 %, -100 % as well as FALSE and TRUE

Note!

Variables are only assigned at the end of the task cycle! Thus, the variable value will only be effective in the next task cycle!

2.4.4 User code

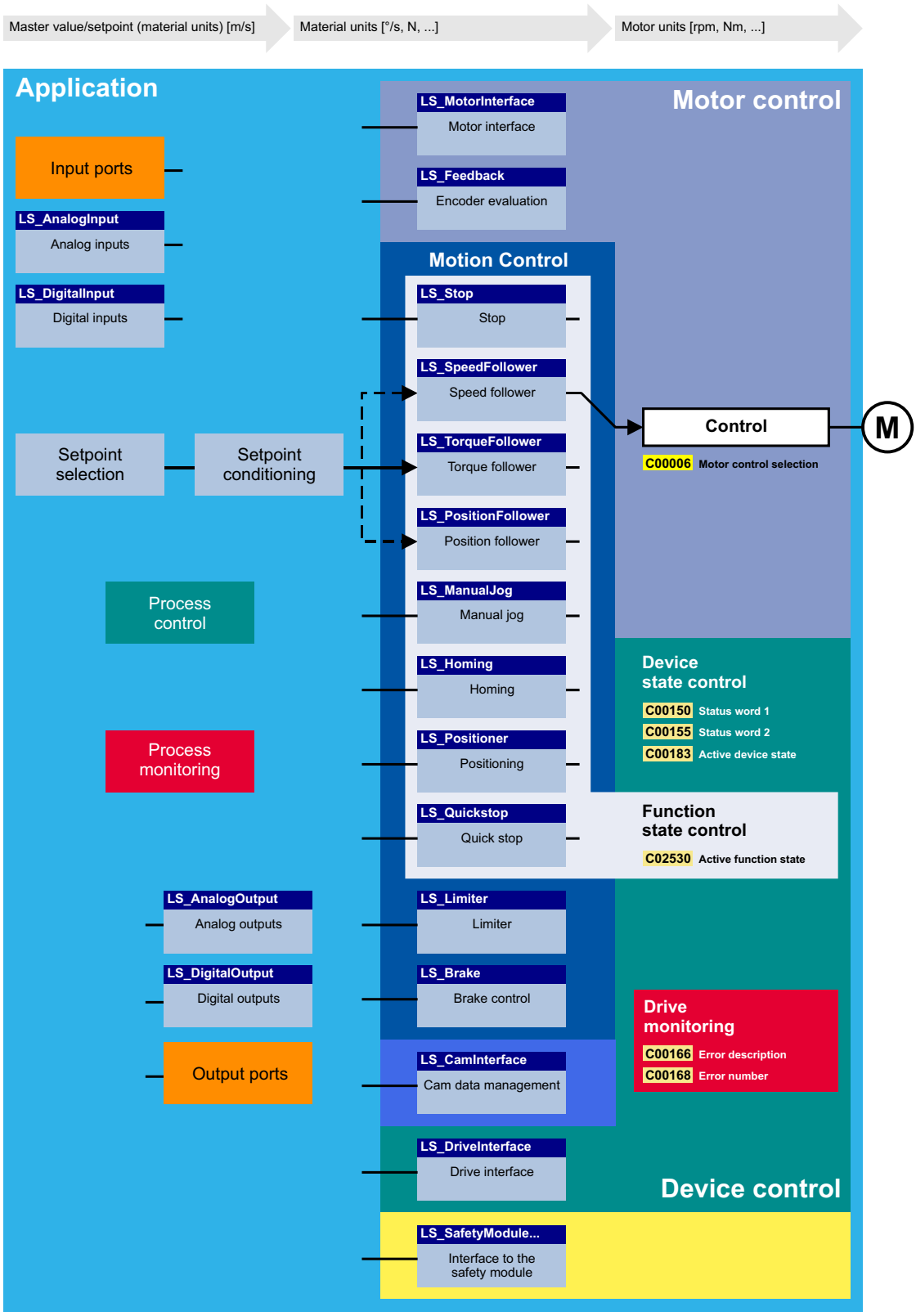
For the interconnection, user codes can be configured and linked with inputs/outputs of port blocks, function blocks, and system blocks. Thus, the linked inputs/outputs can be accessed using the keypad.



❶ Code number/number of subcodes
❷ Signal type entry
❸ Display text of the user code
❹ Button for opening the parameter list for the user code

[2-7] Example: User codes for the selection of a speed setpoint using the keypad

2.5 Functional ranges of an interconnection



[2-8] Schema: Functional ranges of an interconnection

The illustration [2-8] shows the different functional ranges within a typical application in a simplified way.

Application

Setpoints and control signals are read into the application and application-specific functions as setpoint conditioning, process control, and process monitoring are implemented by a corresponding function block interconnection.

- Here, you can access comprehensive Lenze function libraries, which contain, among other things, process controllers, arithmetic functions, logic modules as well as ramp generators and integrators.
- Actual values and status signals can be provided via the output terminals of the controller and via ports when communicating via a network.
- All parameters are selected and displayed in the real machine or material units.

Motor control

Here, all control and monitoring functions of the specific motor functions are implemented.

- All parameters are selected and displayed in motor units.
- In the function block editor, the motor control is accessed via the motor interface (system block **LS_MotorInterface**).

Device control

The device control contains all control and monitoring functions of the controller.

- In the function block editor, the device control is accessed via the drive interface (system block **LS_DriveInterface**).
- The machine constants for the motor end are also entered via the drive interface.

Basic drive functions (Motion Control)

Standard functions of the motion control (e.g. manual jog/inching, homing, brake control, positioning) are already included in the firmware of the controller as basic drive functions.

- In the function block editor, the basic drive functions are accessed via system blocks.
- The basic drive functions which are required in the interconnection in the form of a system block depend on the drive task to be solved.
- All parameters are selected and displayed in machine units. Thus, the user is able to work with the familiar units.

The implementation of the basic drive functions into the firmware offers many advantages for the user:

- The basic functions are always directly available.
- The interfaces to the basic functions are standardised and thus identical for all technology functions.
- Uniform dialogs provide a uniform "look & feel".
- The internal function state control guarantees that
  - only one basic function at a time controls the drive.
  - only one basic function with the highest priority is executed if several basic functions are activated at the same time.
  - the drive is always in a defined state, both in case of an error and in standard operation.
- All basic functions are synchronised to each other. You can change between the basic functions online while the drive is rotating.
- The processing speed is optimal.

2.6 Conventions used for input/output identifiers

This chapter describes the conventions used for the identifiers of the inputs/outputs of the function blocks. The conventions ensure a uniform and consistent terminology and make reading and comprehending the interconnection and application easier.



Tip!

The conventions used by Lenze are based on the "Hungarian Notation". This ensures that the most significant characteristics of the corresponding input/output (e.g. the data type) can be instantly recognised from its identifier.

An identifier consists of

- a data type entry
- an identifier (the "proper" name of the input/output)
- an (optional) signal type specification

Data type entry

The data type entry provides information about the data type of the corresponding input/output:

Data type entry	Meaning	Resolution	Value range
B	BOOL	1 bit	0 ≡ FALSE / 1 ≡ TRUE
by	BYTE	8 bits1	0 ... 255
dn	DINT	32 bits	-2147483647 ... 2147483647
dw	DWORD	32 bits	0 ... 4294967295
n	INT	16 bits	-32767 ... 32767
s	STRING	-	-
t	TIME	32 bits	-
W	WORD	16 bits	0 ... 65535

Identifier

The identifier is the proper name of the input/output and should indicate the application or function.

- Identifiers always start with a capital letter.
- If an identifier consists of several "words", then each "word" must start with a capital letter.
- All other letters are written in lower case.

Signal type entry

In general, it is possible to assign a certain signal type to the inputs and outputs of the Lenze function blocks. There are e.g. digital, scaled, position, acceleration and speed signals.

- A corresponding ending (preceded by an underscore) is added to the identifier of the corresponding input/output to indicate the signal type.

Signal type entry	Meaning	Resolution	Value range
_a	Scaled	16 bits	± 199.99 %
_n	Scaled	32 bits	± 200.00 %
_v	Speed	16 bits	± 30000.0 rpm
_s	Speed	32 bits	± 480000.0 rpm
_p	Position/angle	32 bits	-2^{31} ... 2^{31}-1 increments*
* As of controller software version V3.0 the resolution of an encoder revolution can be parameterised in C00100 (Lenze setting: 16 bits/encoder revolution).			

- An overview of the different signal types and their scaling is provided in the following subchapter "Signal types & scaling (📖 29)".

2.7 Signal types & scaling

With regard to the parameter setting & configuration of the controller it is very helpful to know the signal types and their scaling listed in the following table, which are used to process physical values (e.g. a speed or position) in the function block interconnection.

Signal type (data type)	Port symbol in the FB editor	Resolution	Value range (external)	Decimal positions/ Signal type indication in the identifier	
Scaled (INT)	○	16 bits	± 199.99 %	2	_a
Scaled (DINT)	●	32 bits	± 200.00 %	2	_n
Speed (INT)	◁ / ▷	16 bits	± 30000.0 rpm	1	_v
Speed (DINT)	◆	32 bits	± 480000.0 rpm	1	_s
Position/angle (DINT)	◀ / ▶	32 bits	-2^{31} ... 2^{31}-1 increments*	3	_p
Digital (BOOL)	□	1 bit	0 ≡ FALSE; 1 ≡ TRUE	0	
Acceleration (DINT)	■	32 bits	± 7.69 * 10^9 rpm/s	3	_x
Time	■	28 bits1	0 ... 268435.456 s	3	
Other (BYTE)	■	8 bits1	0 ... 255	0	
Other (WORD)	■	16 bits	0 ... 65535	0	
Other (DWORD)	■	32 bits	0 ... 4294967295	0	
Other (INT)	■	16 bits	-32767 ... 32767	0	
Other (DINT)	■	32 bits	-2147483647 ... 2147483647	0	

* As of controller software version V3.0 the resolution of an encoder revolution can be parameterised in C00100 (Lenze setting: 16 bits/encoder revolution).

Scaling of physical units

Signal type	Port symbol in the FB editor	Resolution	Scaling	
			external value	≡ internal value
Scaled (INT)	○	16 bits	100 %	≡ 2^{14} ≡ 16384
Scaled (DINT)	●	32 bits	100 %	≡ 2^{30} ≡ 1073741824
Speed (INT)	◁ / ▷	16 bits	15000 rpm	≡ 2^{14} ≡ 16384
Speed (DINT)	◆	32 bits	15000 rpm	≡ 2^{26} ≡ 67108864
Position/angle (DINT)	◀ / ▶	32 bits	1 encoder revolution	≡ 2^{16} increments*
Acceleration (DINT)	■	32 bits	15000000 rpm/s	≡ 2^{22} ≡ 4194304

* As of controller software version V3.0 the resolution of an encoder revolution can be parameterised in C00100 (Lenze setting: 16 bits/encoder revolution).

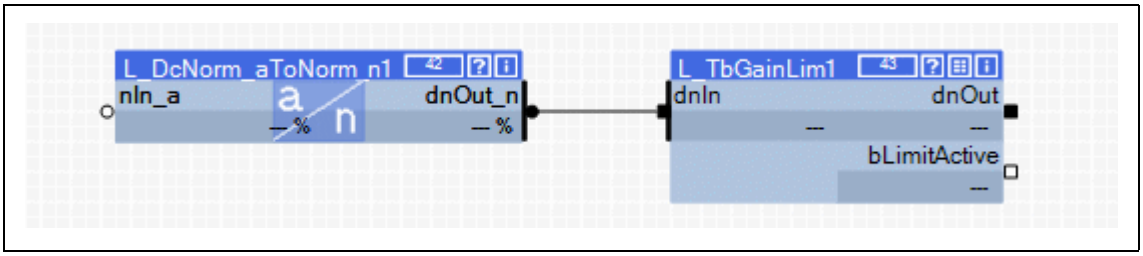
2.7.1 Data type check/implicit type conversion

For the »Engineer« up to and including V2.7 the following applies:

- Only inputs/outputs of the same signal type (with the same port symbol) can be connected in the »Engineer« function block editor.
- Non-standardised inputs/outputs can be connected if the input and output have the same resolution.
- If different types are to be connected, corresponding conversion modules must be placed in the function block editor.

As of the »Engineer« V2.9 the following applies:

- Signals of the DINT data type can now also be connected to signals of the DWORD data type. Signals of the INT data type can now also be connected to signals of the WORD data type.
- 32-bit-, 16-bit- and 8-bit data types can only be connected among each other.
- A type conversion does not require any conversion modules anymore which need more calculating time. Instead, the type conversion is implicit.
- The implicit type conversion is represented by a black bar at the port symbol in the function block editor.



[2-9] Representation of the implicit type conversion in the function block editor

- The following three tables show all permissible connections:

32-bit data types and signal types	Scaled (DINT)	Speed (DINT)	Position/ angle (DINT)	Accelerati- on (DINT)	Other (DWORD)	Other (DINT)	Port (UDINT)	Port (DINT)	Port (DWORD)
Scaled (DINT)	●				●	●		●	●
Speed (DINT)		●			●	●		●	●
Position/angle (DINT)			●		●	●		●	●
Acceleration (DINT)				●	●	●		●	●
Other (DWORD)	●	●	●	●	●	●	●	●	●
Other (DINT)	●	●	●	●	●	●		●	●
Port (UDINT)					●		●		●
Port (DINT)	●	●	●	●	●	●		●	●
Port (DWORD)	●	●	●	●	●	●	●	●	●

16-bit data types and signal types	Scaled (INT)	Speed (INT)	Other (WORD)	Other (INT)	Port (UINT)	Port (INT)	Port (WORD)	
Scaled (INT)	●		●	●		●	●	
Speed (INT)		●	●	●		●	●	
Other (WORD)	●	●	●	●	●	●	●	
Other (INT)	●	●	●	●		●	●	
Port (UINT)			●		●		●	
Port (INT)	●	●	●	●		●	●	
Port (WORD)	●	●	●	●	●	●	●	

8-bit data types and signal types	Other (BYTE)	Port (USINT)	Port (SINT)	Port (BYTE)	
Other (BYTE)	●	●	●	●	
Port (USINT)	●	●		●	
Port (SINT)	●		●	●	
Port (BYTE)	●	●	●	●	

2.8 Multitasking in the Servo Drive 9400

The Servo Drive 9400 has three different user tasks:

1. ApplicationTask
2. UserTask
3. IdleTask

ApplicationTask

The ApplicationTask is a time-controlled user task.

- Two different interval times (1 ms and 2 ms) can be set for the ApplicationTask.
- Compared to the other two user tasks, the ApplicationTask has the higher priority with regard to processing.
- The system blocks can only be accessed in the ApplicationTask. Thus, new setpoints are provided in the same interval time as defined for the ApplicationTask.

UserTask

The UserTask is, like the ApplicationTask, a time-controlled user task.

- Three different interval times (2 ms, 4 ms and 8 ms) can be set for the UserTask.
- The UserTask has a higher priority than the IdleTask.

IdleTask

The IdleTask is an idling user task.

- After the logic has been processed in the IdleTask, the IdleTask restarts. Thus, the IdleTask is not time-equidistant.
- The IdleTask has the lowest priority.
- The processing of the IdleTask is interrupted by the ApplicationTask and the UserTask.



Tip!

The task properties (interval time and runtime monitoring) are configured in a dialog which is called from the FB Editor.

▸ Configuring the task properties (📖 83)

2.8.1 Runtime measurement

When the application has been started, the controller continuously carries out a runtime measurement for the interval-controlled ApplicationTask, the interval-controlled UserTask and the idling IdleTask.

- The runtime is the time from the starting time of the task to the time where the processing of the task logic has been completed, including all interruptions (e.g. by higher-priority tasks).
- The runtime measurement also includes an operating system part of the Servo Drive 9400, the runtime of which depends among other things on the plugged-in modules.
- The current and maximum task runtimes are displayed in the following display parameters:

Parameter	Information	Lenze setting	
		Value	Unit
C02121/1	Current runtime of ApplicationTask	-	µs
C02121/2	Maximum runtime of ApplicationTask	-	µs
C02122/1	Current runtime of UserTask	-	µs
C02122/2	Maximum runtime of UserTask	-	µs
C02123/1	Current runtime of IdleTask	-	µs
C02123/2	Maximum runtime of IdleTask	-	µs
Grayed out = display parameter			

- The device command C00002 = "36: Reset runtime measurement" serves to reset the runtime measurement, e.g. the memory for the maximum values is reset to "0".

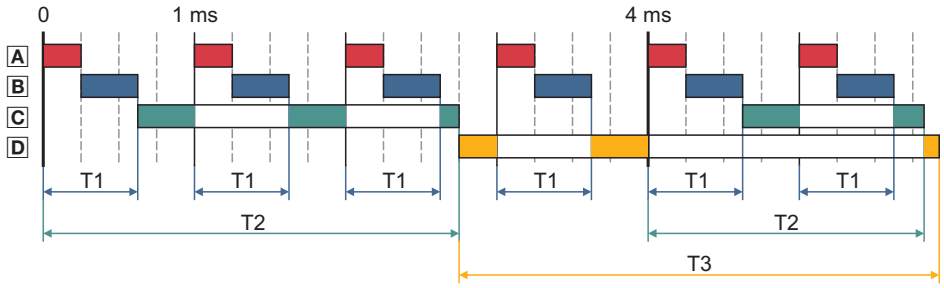


Note!

The runtime measurement is also reset by the following actions:
- Start application
- Reset/delete/restart program

Example of runtime measurement

In the following example, the ApplicationTask is called in a 1 ms interval. The ApplicationTask interrupts the UserTask which is called in a 4 ms interval. Both tasks, in turn, interrupt the IdleTask. The interruptions cause a runtime extension both in the UserTask and in the IdleTask. This means that already a small runtime extension has a significant effect on the runtime of the other two tasks!



A System task B Application task (here: Task interval = 1 ms) C User task (here: Task interval = 4 ms) D Idling idle task	T1 = runtime of the application task T2 = runtime of the user task T3 = runtime of the idle task

[2-10] Example: Runtimes of the different tasks

2.8.2 Optimising the runtime behaviour

The following points should be considered for an optimum runtime behaviour of the controller:

- Before you start programming, check which response times are required for which functions. This defines in which task the corresponding functions are to be programmed and which interval time must be set for the corresponding task. A subsequent optimisation of programs is very time-consuming and error-prone!
- High-priority, quick tasks must always be programmed runtime-efficiently. The less logic is executed in the quick task, the more resources are available for the other tasks!
- The runtimes of the time-controlled user tasks must not be higher than 95 % of the interval time of the corresponding task.
  - If, for example, the ApplicationTask is loaded with 70 %, only the 25 % are available for the UserTask and the IdleTask.
  - Check the runtimes by means of the runtime measurement. ▸ Runtime measurement (📖 33)
- If the runtime of the ApplicationTask is higher than 950 µs with an interval time of 1 ms, check first if the interval time of the ApplicationTask can be increased from 1 ms to 2 ms. This is the case if:
  - the setpoints in the application are generated in an interval higher than 2 ms, as for example with positioning drives or actuating drives with internal ramp generation or slave drives with a setpoint selection. Slave drives with a setpoint generation in an interval lower than 2 ms are excluded.
  - no time-critical operation of digital outputs is required as for example with a cam group.
- An increase of the interval time only causes a minimum deterioration of the drive behaviour if the motion setpoints are generated in the controller.
- The activation/deactivation of the three setpoint interpolators (C02550/1...3) has no influence on the task runtime.
- In a second step, the logics from the ApplicationTask should be sourced out into one of the other two tasks.
  - Time-critical logics should remain in the ApplicationTask.
  - Logics (modules) which require a time-equidistant execution and do not need to be executed in the ApplicationTask should be transferred to the UserTask.
  - Logics (modules) the execution of which is not time-critical and not time-equidistant should be transferred to the IdleTask.
- When the project is based on a technology application by Lenze, the logic which has no influence on the application function, e.g. non-used ports with connected signal converters, should be deleted from the ApplicationTask after activating the FB Editor.
  - Deleting non-required system blocks from the FB interconnection does not influence the task runtime.

2.8.3 Exchanging data between tasks

There are two options to exchange data between tasks:

1. via global variables
2. via user codes

In order to e.g. transfer a value from the ApplicationTask to the UserTask, first a global variable or a user code is defined. The value is assigned to this variable/code in the ApplicationTask and read from the same variable/code in the UserTask and e.g. assigned to a block input.

Note!

When creating the FB interconnection please bear in mind that the value of a global variable or user code may change while processing the lower-priority task when being interrupted by a higher-priority task.

The FB Editor prevents that the same variable or user code can be written several times. Thus, a variable or user code cannot be changed in different tasks.

3 Working with the FB editor

The function block editor (in the following called "FB editor) is available in the »Engineer« HighLevel 9400 Servo Drives with the runtime software licence Motion Control HighLevel or higher.

The FB editor serves to extend the supplied technology applications by individual functions. You can optionally also create an application completely of your own. Here you can access the comprehensive Lenze libraries which, among other things, contain process controllers, arithmetic functions, logic modules and ramp generators and integrators.

Note!

The illustrations of the FB Editor user interface and the dialog boxes in this documentation are based on the »Engineer« V2.9 (or higher).

Functions which are only available as of the Engineer version V2.9 (or higher) are marked correspondingly in this documentation.

3.1 Short overview of new functions in the FB Editor

As of »Engineer« V2.10

- Configuring the exception handling (📖 84)
- Inserting a comment with extended properties (📖 67)
  - The interior colour and text alignment of a comment can be changed via a properties dialog. The size of a comment can also be changed easily using the mouse pointer.
  - When using different interior colours you can use comments to graphically arrange areas that belong together in terms of function or separate them from other areas.

As of »Engineer« V2.9

- Data type check/implicit type conversion (📖 30)
- Testing the interconnection in the device (📖 86)
- Copying interconnection elements (across all devices) (📖 88)
- Comparing interconnections (📖 93)
- Copying complete interconnection (📖 97)
- Inserting complete interconnection from reference project (📖 98)

3.2 User interface

How to access the FB editor:

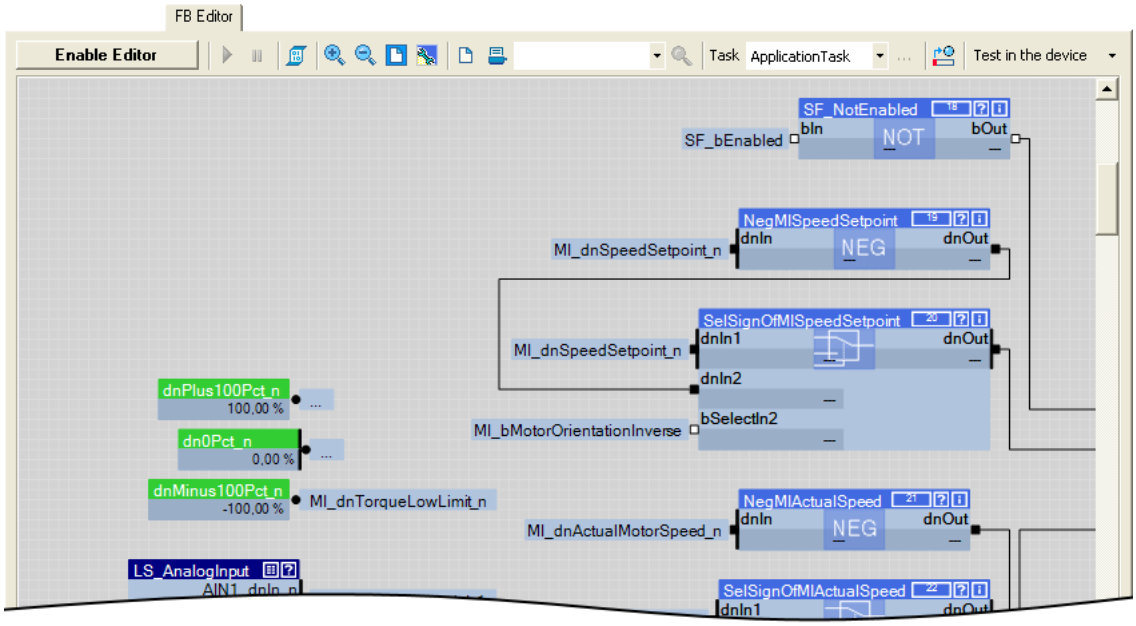
1. Go to *Project View* and select the 9400 controller.
2. Go to *Workspace* and select the **FB Editor** tab.

Note!

The first representation on the **FB Editor** tab depends on whether you have selected a technology application for the controller from the catalogue or not.

Controller with selected technology application

If you have selected a technology application for the controller from the catalogue, the FB Editor displays the interconnection of the technology application defined by Lenze:

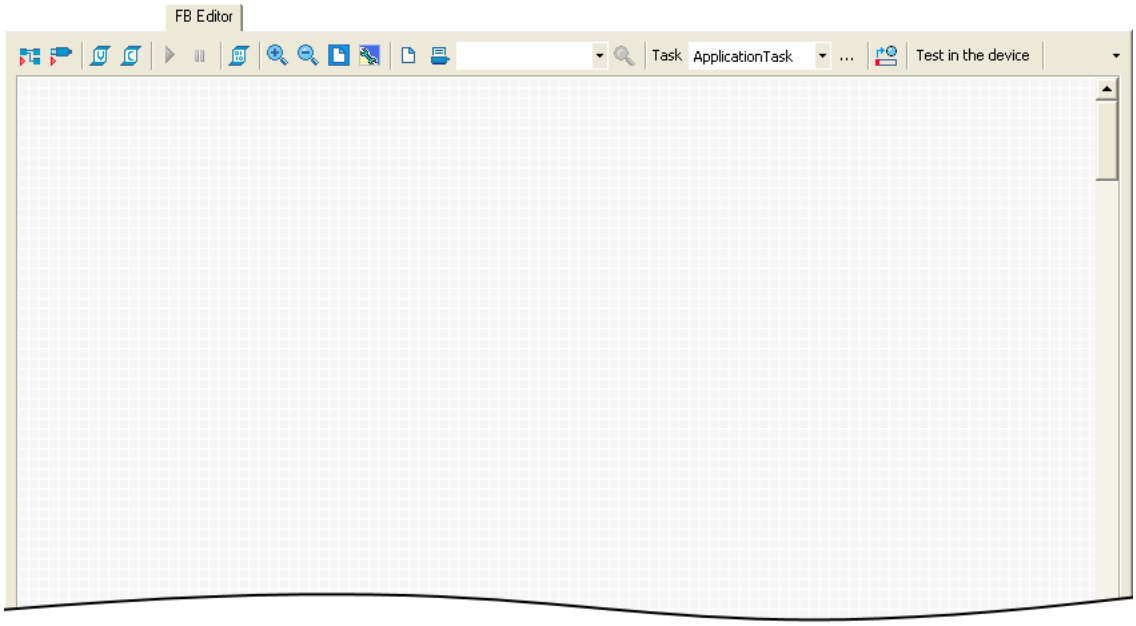


- First the FB Editor is in the "Viewer" mode, in which the displayed objects/connections cannot be changed. ▸ Using the FB-Editor as "Viewer" (📖 48)
- To be able to reconfigure the defined interconnection and to add further functions, the FB Editor must be enabled explicitly. ▸ Enabling the FB Editor (📖 51)

Controller with empty application

If no technology function has been selected from the catalogue when the controller has been entered into the *Project View*, the controller is automatically assigned to an "empty application".

In this case, the FB Editor contains an empty and already enabled drawing area where you can recreate the required interconnection from scratch:



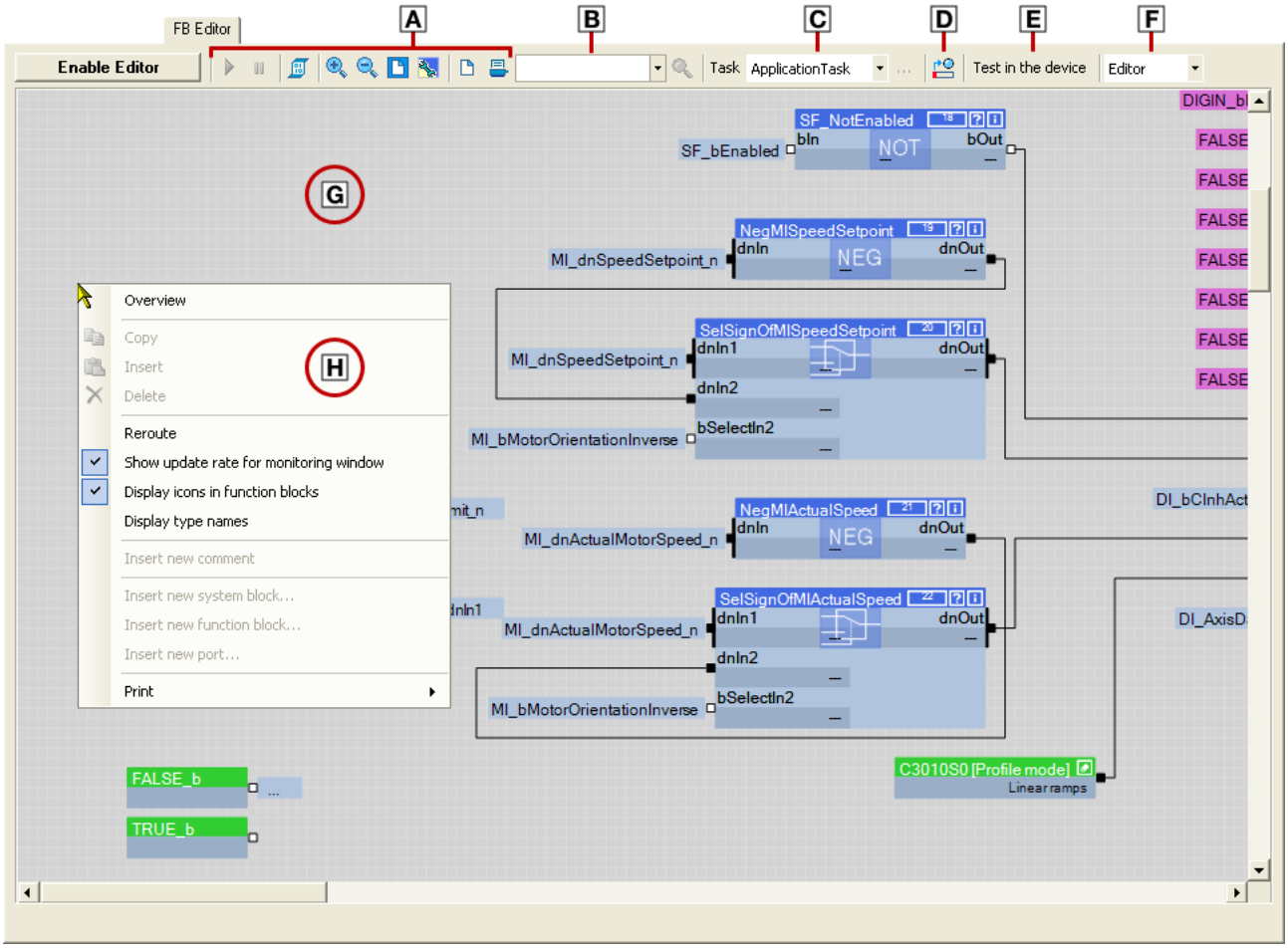
Tip!

You can replace the empty application any time by another technology application from the catalogue.

1. Select the controller in the *Project View*.
2. Select the command **Insert→Application**.

3.2.1 Control and function elements

The user interface of the FB Editor includes the following control and function elements:



A Toolbar
B Search function
C Task selection
D Configuring the exception handling
E Testing the interconnection in the device
F Editor view/overview
G Drawing area
H Context menu
Not shown: Overview window

Tip!

Go to the *»Engineer« toolbar* and click the icon to hide the *Project View* and the *Message Window*. This increases the *Workspace* available for the FB Editor. A renewed click on the symbol shows the *Project View* and the *Message Window* again.

3.2.2 Toolbar

The FB Editor is provided with an individual toolbar in the upper position which in the following text is called *FB Editor toolbar*.

- Click an icon to execute the corresponding function.

Icons/functions in the "Viewer" mode

Icon	Function
	Start online monitoring (as of »Engineer« V2.10)
	Interrupt online monitoring (as of »Engineer« V2.10)
	Stop online monitoring (as of »Engineer« V2.10)
	Enlarge view of interconnection
	Reduce view of interconnection
	Enlarge cutout of interconnection
	Show total interconnection in the drawing area
	Show FB type names (instead of FB instance names)
	Show print view
	Printing the interconnection (📖 92)
	Search function (📖 42)
	Configuring the exception handling (📖 84) (as of »Engineer« V2.10)

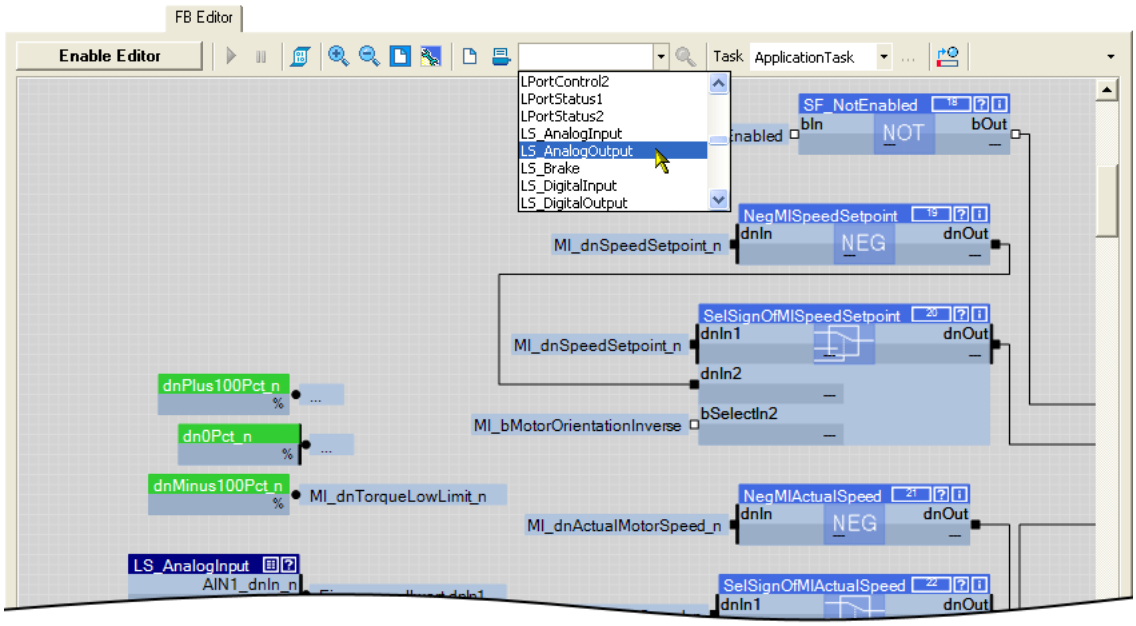
Additional icons/functions in the activated FB Editor

Icon	Function
	Insert function block or system block ▸ Inserting a function block (📖 53) ▸ Inserting a system block (📖 56)
	Inserting a port block (📖 58)
	Creating/inserting a variable (📖 59)
	Configuring/inserting a user code (📖 62)
	Compiling interconnection (📖 86)

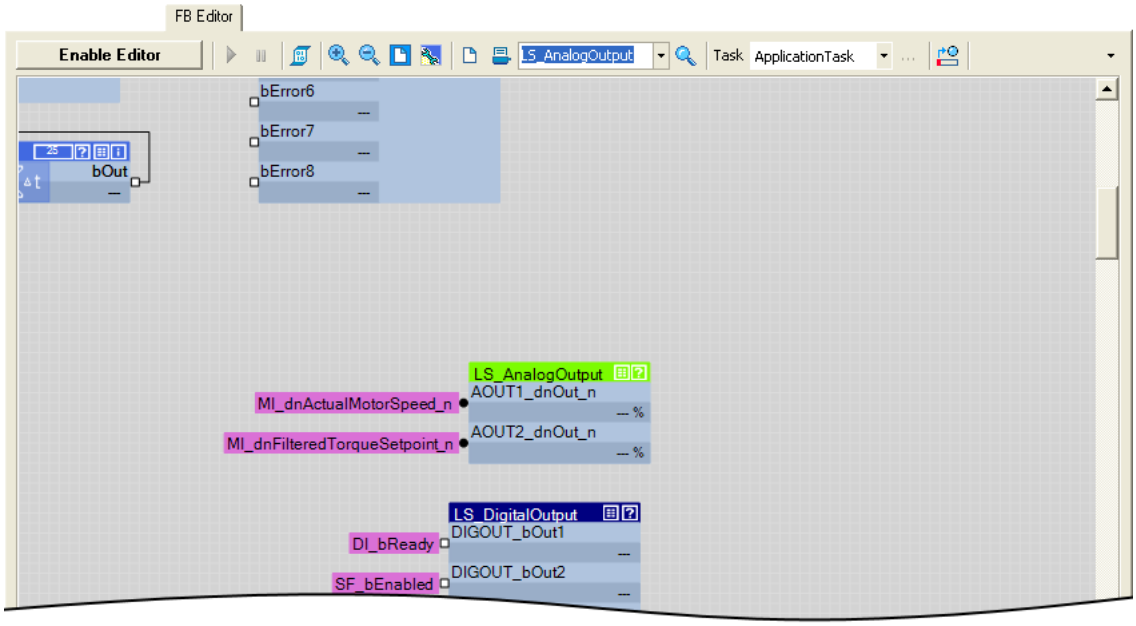
3.2.3 Search function

Use the search function to get quickly to a certain module of the interconnection.

- The list field of the search function contains all function blocks, system blocks, and port blocks of the interconnection:



- When you select a module in the list field, this module is zoomed in and selected at the same time (the following example shows the **LS_AnalogOutput** system block):



Tip!

You can also enter any search text in the input field.

- If you click the icon, the cutout is moved to the object which contains this search text.
- Another click on the icon leads to a new search. Thus, you can navigate successively to all objects which contain the entered search text.
- The search text does not consider case sensitivity.

3.2.4 Task selection

A task describes the runtime characteristics, such as the cycle time and the priority of one or several programs running within the controller. Together with the program, the task makes a runtime system.

- The task to be displayed in the drawing area or edited in the activated FB Editor can be selected via the list field (same name):



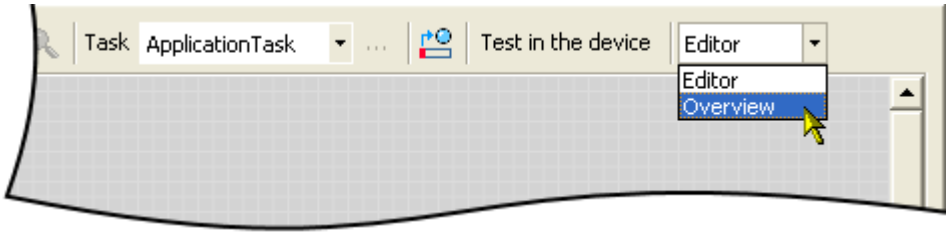
- By clicking the **...** button right beside the list field, you open a dialog field where you can see the properties (interval, type, watchdog, sensitivity) of the selected task. The task properties can also be changed, if required, in the activated FB Editor.

Related topics:

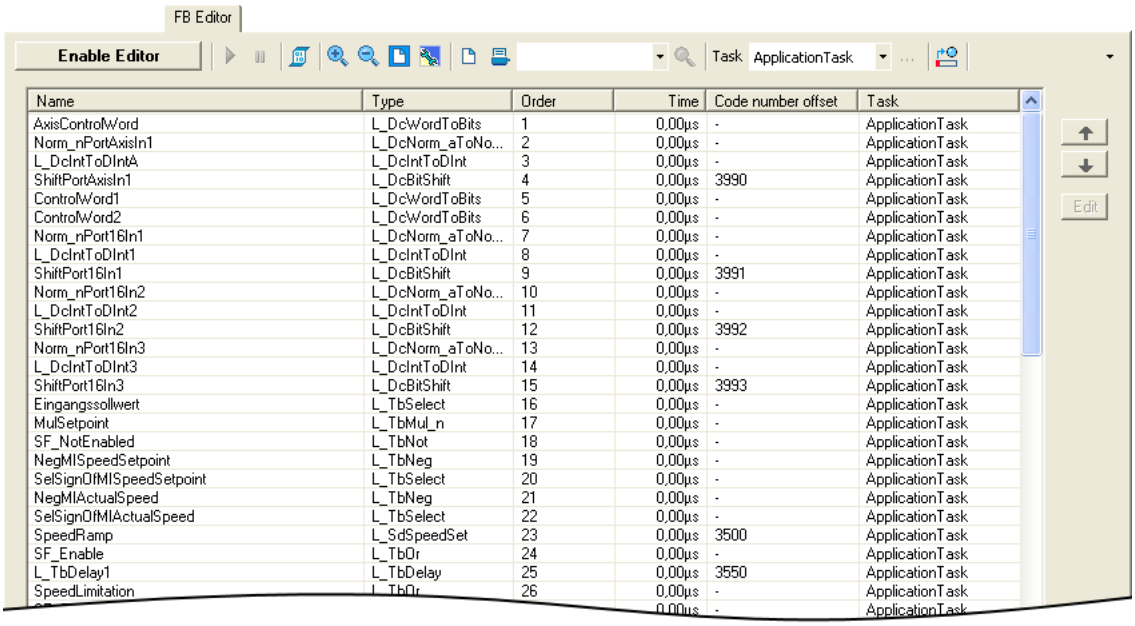
▸ Multitasking in the Servo Drive 9400 (📖 32)
▸ Configuring the task properties (📖 83)

3.2.5 Editor view/overview

Use the rightmost list field to change from the Editor to the overview and vice versa:



The overview displays all function blocks, system blocks, and port blocks of the interconnection in the order of their processing:

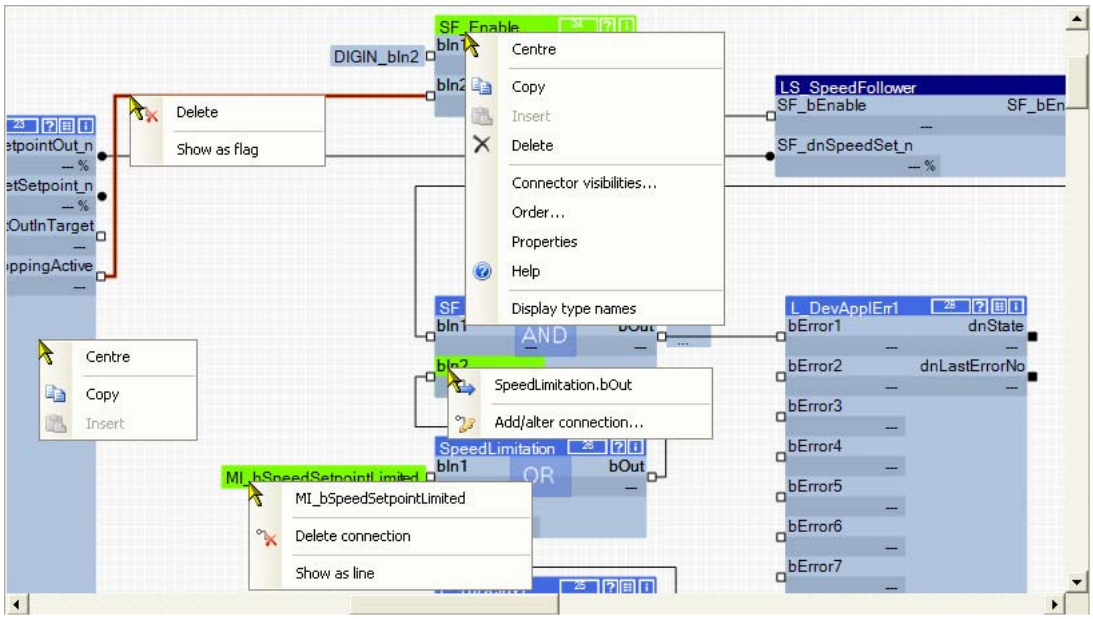


Name	Type	Order	Time	Code number offset	Task
AxisControlWord	L_DcWordToBits	1	0,00µs	-	ApplicationTask
Norm_nPortAxisIn1	L_DcNorm_aToNo...	2	0,00µs	-	ApplicationTask
L_DcIntToDIntA	L_DcIntToDInt	3	0,00µs	-	ApplicationTask
ShiftPortAxisIn1	L_DcBitShift	4	0,00µs	3990	ApplicationTask
ControlWord1	L_DcWordToBits	5	0,00µs	-	ApplicationTask
ControlWord2	L_DcWordToBits	6	0,00µs	-	ApplicationTask
Norm_nPort16In1	L_DcNorm_aToNo...	7	0,00µs	-	ApplicationTask
L_DcIntToDInt1	L_DcIntToDInt	8	0,00µs	-	ApplicationTask
ShiftPort16In1	L_DcBitShift	9	0,00µs	3991	ApplicationTask
Norm_nPort16In2	L_DcNorm_aToNo...	10	0,00µs	-	ApplicationTask
L_DcIntToDInt2	L_DcIntToDInt	11	0,00µs	-	ApplicationTask
ShiftPort16In2	L_DcBitShift	12	0,00µs	3992	ApplicationTask
Norm_nPort16In3	L_DcNorm_aToNo...	13	0,00µs	-	ApplicationTask
L_DcIntToDInt3	L_DcIntToDInt	14	0,00µs	-	ApplicationTask
ShiftPort16In3	L_DcBitShift	15	0,00µs	3993	ApplicationTask
Eingangssollwert	L_TbSelect	16	0,00µs	-	ApplicationTask
MulSetpoint	L_TbMul_n	17	0,00µs	-	ApplicationTask
SF_NotEnabled	L_TbNot	18	0,00µs	-	ApplicationTask
NegMISpeedSetpoint	L_TbNeg	19	0,00µs	-	ApplicationTask
SelSignOfMISpeedSetpoint	L_TbSelect	20	0,00µs	-	ApplicationTask
NegMIActualSpeed	L_TbNeg	21	0,00µs	-	ApplicationTask
SelSignOfMIActualSpeed	L_TbSelect	22	0,00µs	-	ApplicationTask
SpeedRamp	L_SdSpeedSet	23	0,00µs	3500	ApplicationTask
SF_Enable	L_TbOr	24	0,00µs	-	ApplicationTask
L_TbDelay1	L_TbDelay	25	0,00µs	3550	ApplicationTask
SpeedLimitation	L_TbOr	26	0,00µs	-	ApplicationTask

- As of the controller firmware V5.0 the average runtime is displayed for each function block.
- When the FB Editor is activated, the processing order of the function blocks can be changed via the arrow keys on the right side. ▸ Changing the processing order (📖 81)

3.2.6 Context menu

You can open a *context menu* via the right mouse button for each object (function block, variable, line, etc.) and for the drawing area:



- The contents of the *context menu* depend on the type of object you click on.
- As of the »Engineer« V2.10 small icons are shown in front of the entries in the *Context menu*.

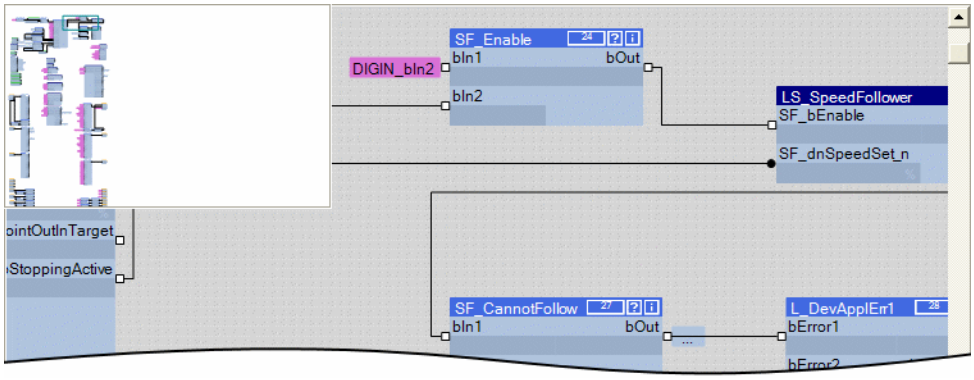
3.2.7 Overview window

The overview window shows the entire interconnection in a reduced view. The overview window serves to e.g. move quickly through a more complex interconnection.

How to show the monitor window:

Go to the *Context Menu* of the drawing area and select the **Overview Window**.

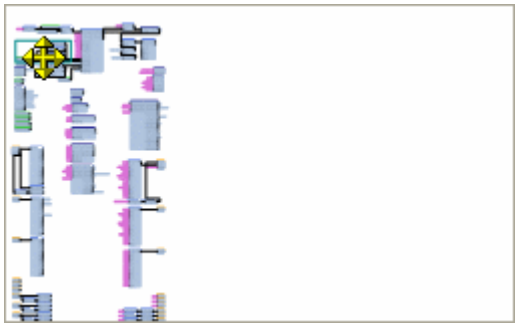
- If you execute this command again, the overview window is hidden again.



- The green frame in the overview window indicates the interconnection cutout that is currently displayed in the drawing area.
- Use the mouse pointer to shift and resize the cutout to be displayed.

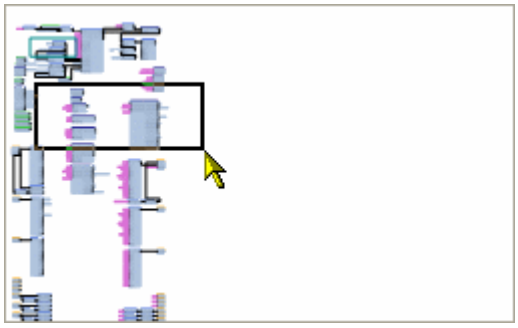
How to shift the cutout presented in the drawing area:

1. Position the mouse pointer to the green frame in the overview window.
2. Click left mouse button and shift the green frame to its new position by keeping the mouse button pressed, so that the desired cutout of the interconnection is displayed in the drawing area.



How to redefine the cutout to be presented:

In the overview window draw a frame around the area of the interconnection which is to be presented in the drawing window by keeping the left mouse button pressed:



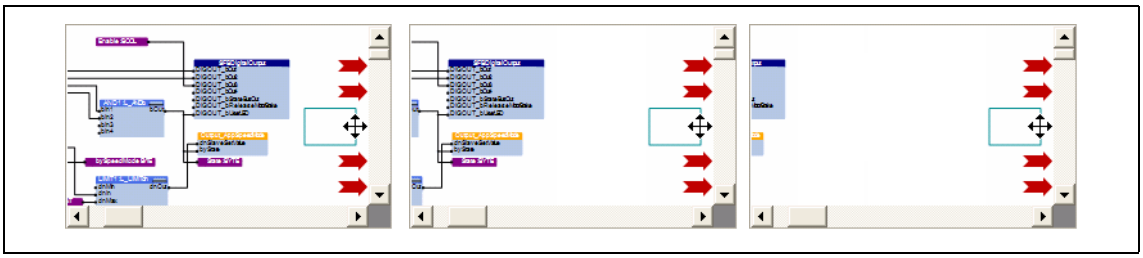
- The aspect ratio of the frame is automatically adapted to the aspect ratio of the drawing area.
- According to the size of the frame that is drawn, also the presentation size of the objects in the drawing area changes.

Tip!

Go to the *FB Editor toolbar* and click the icon to adapt the view size so that all objects included in the interconnection are visible in the drawing area.

Automatic scrolling ("AutoScroll function")

If you reach a window limitation in the drawing area when shifting an object or in the overview window when shifting the green frame, and if you then shortly hold the mouse pointer in this position, an automatic scrolling into the corresponding direction is carried out:



[3-1] Example: automatic scrolling to the right in the overview window

3.3 Using the FB-Editor as "Viewer"

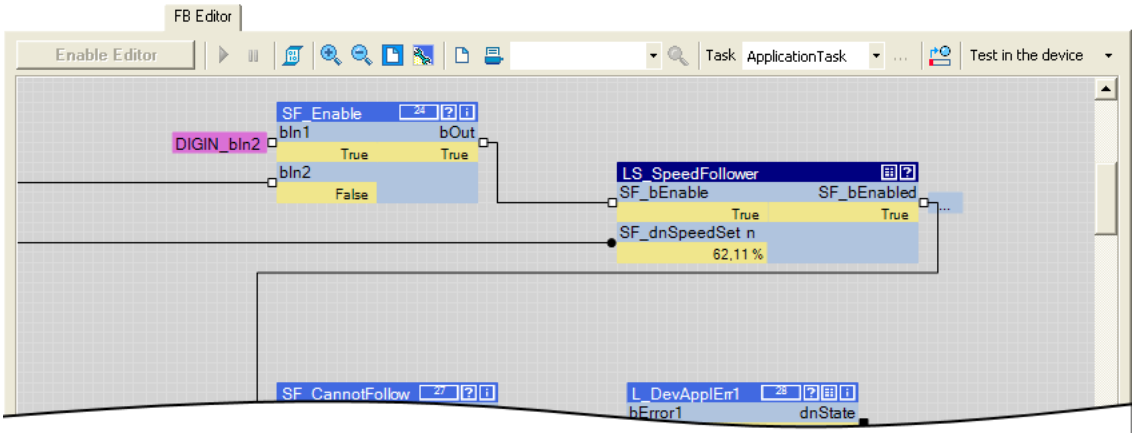
If you have selected a technology application from the catalogue for the controller, the FB Editor is first in the "Viewer" mode. The interconnection of the technology application is displayed but cannot be modified.

You can use this "Viewer" mode to:

- make a diagnosis of the application (when an online connection has been established),
- get a better understanding for the operating mode of the application,
- use the interconnection as an alternative parameterisation access.

Diagnostics of the application

When an online connection to the controller has been established, the current values are displayed at the inputs and outputs of the objects.



Getting a better understanding for the operating mode of the application

Make yourself familiar with the signal flow of the interconnection to get a better understanding of the operating mode of the application or individual functional areas.

The following buttons serve to get more information:

	Command	Function
	Help	Show online help for the function block.
	Help	Show online help for the system block.

Using the interconnection as an alternative parameterisation access

The following buttons serve to get to the parameterisation dialog or parameter list of an object:

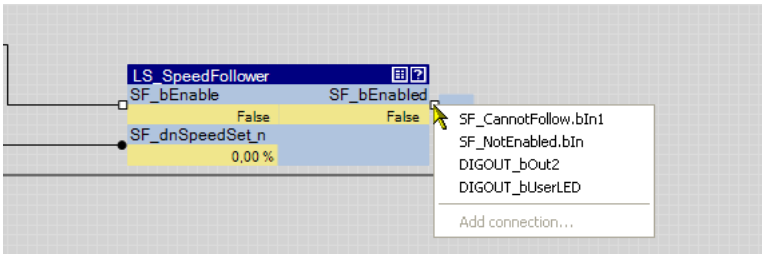
	Command	Function
	Parameter...	Open parameter list or parameterisation dialog of the function block. • Only if function block is parameterisable.
	Parameter...	Open parameter list or parameterisation dialog of the system block.
	Changing a value	Display/change value of the user code.

3.3.1 Following connections of inputs and outputs

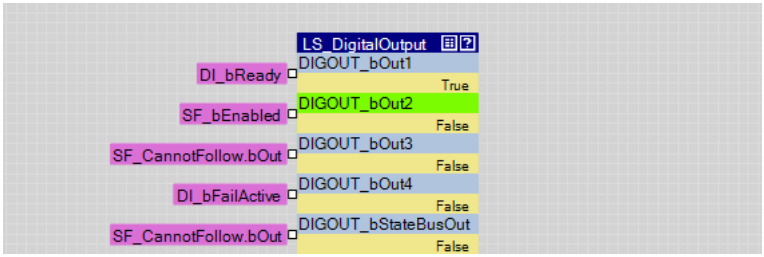
In addition to the Search function and the Overview window you can use the *context menu* of inputs and outputs to follow connections and quickly reach certain signals.

How to navigate from one output to another connected input:

1. Open the *context menu* (right mouse button) of the port symbol at the output.
   - If one output is connected to several inputs, three points ("...") are shown at the output instead of the concrete input identifier.
   - The *context menu* for the port symbol contains all inputs which are connected to the output:



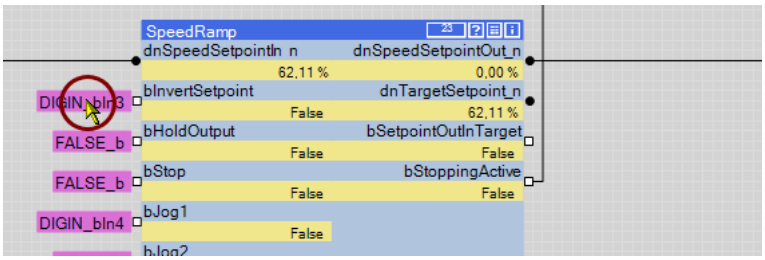
2. Select input in the *context menu* to which you want to navigate.
   - As a result, the selected input is displayed in the centre of the drawing area (in this example: Input DIGOUT_bOut2):



How to navigate from one input to another connected output:

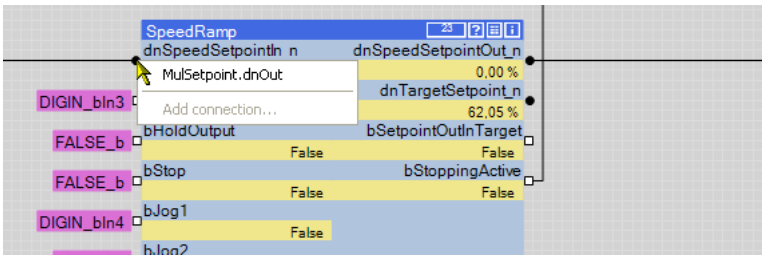
If the input is connected to a flag:

- Double-click the flag:



If the input is connected to a line:

- Open the *context menu* (right mouse button) of the port symbol at the input and select the output in the *context menu*.
- Since an output can only be connected to an input, the *context menu* contains only an output:



The output is displayed in the centre of the drawing area.

3.3.2 Keyboard commands for navigation

Keyboard command	Function
<Picture ▲ >	Scroll up
<Picture ▼ >	Scroll down
<Shift> + <picture ▲ >	Scroll to the left
<Shift> + <picture ▼ >	Scroll to the right
<POS1>	Scroll to the left edge of the interconnection
<END>	Scroll to the right edge of the interconnection
<Ctrl> + <Pos1>	Scroll to the left upper corner of the interconnection
<Ctrl> + <End>	Scroll to the right lower corner of the interconnection

3.4 Creating/reconfiguring the interconnection

How to proceed:

1. Enable FB Editor.
   - Only required if a Lenze technology application is to be reconfigured.
   - If no application or the "empty application" from the catalogue has been selected, the FB Editor is already enabled.
2. Insert required objects into the interconnection.
3. Hide unneeded inputs/outputs of function blocks and system blocks to obtain a clearly arranged interconnection.
4. Arrange the objects in the drawing area in a reasonable manner.
5. Establish the connections required for the desired function.
6. If required, change (optimise) the processing order of the function blocks.
7. Compile the completed interconnection.

Tip!

Detailed information on the individual steps can be obtained from the following subchapters!

3.4.1 Enabling the FB Editor

If you have selected a technology application from the catalogue for the controller, the FB Editor is first in the "Viewer" mode. The interconnection of the technology application is displayed but cannot be modified.

To be able to reconfigure the defined interconnection and to add further functions, the FB Editor must be enabled explicitly.

Note!

When the FB Editor is enabled, the parameter dialogs for the technology application are replaced by general dialogs. The dialogs for the parameterisation of the system and function blocks contained in the interconnection can still be used.

Furthermore, all multiplexers are removed from the interconnection and the currently parameterised signal combinations are replaced by connections.

How to enable the FB Editor:

1. When being in offline mode, press the **Enable Editor** button.
2. Confirm the query whether the FB Editor is to be enabled with **OK**.

3.4.2 Insert/delete objects

Objects can be inserted in the interconnection via the *FB Editor toolbar* and the *context menu* of the drawing area. The following subchapters provide detailed information on how to insert/delete the different objects.

Icon	Function
	Inserting a function block (📖 53)
	Inserting a system block (📖 56)
	Inserting a port block (📖 58)
	Creating/inserting a variable (📖 59)
	Configuring/inserting a user code (📖 62)
	Inserting a comment (📖 67)

Tip!

Use the *context menu* of the drawing area to insert a function block, system block, port block or comment directly to the current position of the mouse pointer in the drawing area.

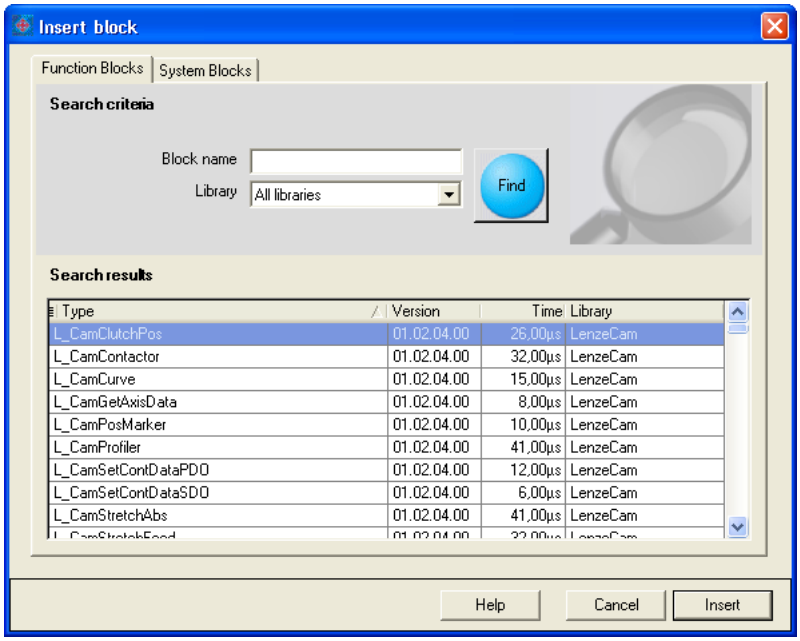
If you insert an object via the corresponding icon in the *FB Editor toolbar*, the object is always placed at the top left corner in the drawing area.

As of the »Engineer« V2.9, interconnection elements cannot only be copied within the same interconnection but also across all devices within the same project, as long as the devices stem from the same product family. ▸ Copying interconnection elements (across all devices) (📖 88)

3.4.2.1 Inserting a function block

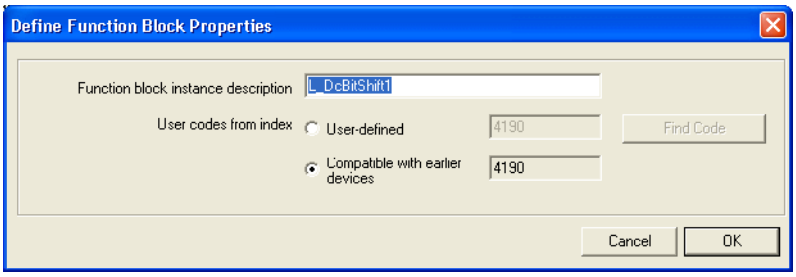
How to insert a function block into the interconnection:

1. Click the icon in the *FB editor toolbar*.
   - The *Insert block* dialog box appears.
2. Unless it is already displayed, select the **Function Blocks** tab.
   - All function blocks available are displayed in the **Search results** list field.

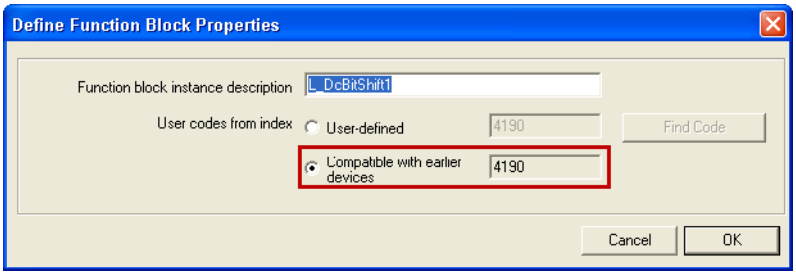


3. If required, define **search criterions** to accordingly narrow down the function blocks available:
   - **Block name**:
     String which must be contained in the name of the function block.
   - **Library**:
     Limitation of the selection to a specific function library.
4. After changing the search criteria, press the **Find** button to update the selection.
   - Then, only the function blocks complying with the features set in the search criteria are shown in the **Search Results** list field.
   - If no search criteria are set, all function blocks available are shown.
5. Select the function block to be inserted in the **Search results** list field.

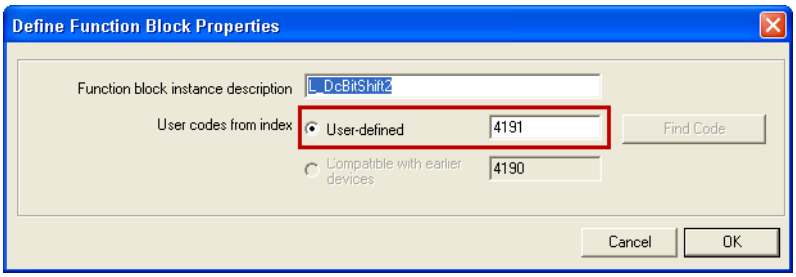
6. Press **Insert** button.
   - The *Define Function Block Properties* dialog box appears.



7. If required, change the name preselected for the function block into a more significant name.
   - The instance name must not already be allocated within the interconnection.
8. If required, adapt the index for a parameterisable function block from which on the assignment of user codes for the parameters is to start.
   - If the user codes for the compatibility with earlier devices are not yet assigned, these are suggested for the function block:



   - If the user codes for the compatibility with earlier devices are already assigned, e.g. since an instance of the function block is already available in the interconnection, a free (user-defined) index will be automatically suggested which can also be changed, if required:



9. Press **OK**.
   - The function block selected is inserted in the interconnection.

Note!

The parameters described in this documentation are always listed together with the user code index that results from the selection of the option **Compatible with earlier devices**!

Context menu for the function block

If you right-click on the header of a function block, a *context menu* opens via which you can execute the following functions in addition to the general processing functions (Copy, Insert, Delete):

	Command	Function
	Connector visibilities...	Define visible inputs and outputs for function blocks. ▸ Changing connector visibilities (📖 72)
	Order...	Change order index of the function block. ▸ Changing the processing order (📖 81)
	**Parameter...**	Open parameter list or parameterisation dialog of the function block. • Only if function block is parameterisable.
	**Properties**	Define function block properties.
	Display type names	Show FB type names (instead of FB instance names).

Related topics:

- Deleting objects that are no longer required (📖 70)
- Changing connector visibilities (📖 72)
- Arranging objects in the drawing area (📖 73)
- Creating/deleting connections (📖 74)
- Changing the processing order (📖 81)

3.4.2.2 Inserting a system block

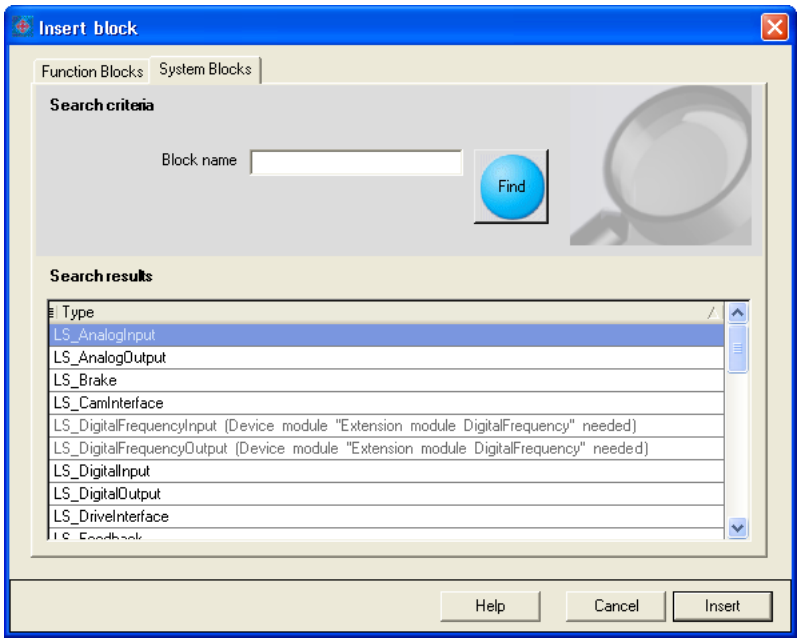
A system block is inserted similarly to the way a function block is inserted.

Note!

In the FB editor, system blocks are only available in the *ApplicationTask*!

How to insert a system block into the interconnection:

1. Click the icon in the *FB editor toolbar*.
   - The *Insert block* dialog box appears.
2. Unless it is already displayed, select the **System Blocks** tab.
   - All system blocks available are displayed in the **Search results** list field.



   - System blocks which are not included already in the interconnection are not shown in the list field anymore.
   - System blocks which are displayed in grey cannot be inserted. Either a certain extension module is required or the function is not supported due to the licence level given by the memory module.
3. If required, define **Search criterions** to accordingly narrow down the system blocks available:
   - **Block name**:
     String which must be contained in the name of the system block.
4. After changing the search criteria, press the **Find** button to update the selection.
   - Then, only the system blocks complying with the features set in the search criteria are shown in the **Search Results** list field.
   - If no search criteria are set, all system blocks available are shown.
5. Select the system block to be inserted in the **Search results** list field.

6. Press **Insert** button.
   - The system block selected is inserted in the interconnection.

Context menu for the system block

If you right-click on the header of a system block, a *context menu* opens via which you can execute the following functions in addition to the general processing functions (Copy, Insert, Delete):

	Command	Function
	Connector visibilities...	Define visible inputs and outputs for system blocks. ▸ Changing connector visibilities (📖 72)
	**Parameter...**	Open parameter list or parameterisation dialog of the system block.
	**Help**	Show online help for the system block.

Related topics:

- Deleting objects that are no longer required (📖 70)
- Changing connector visibilities (📖 72)
- Arranging objects in the drawing area (📖 73)
- Creating/deleting connections (📖 74)

3.4.2.3 Inserting a port block

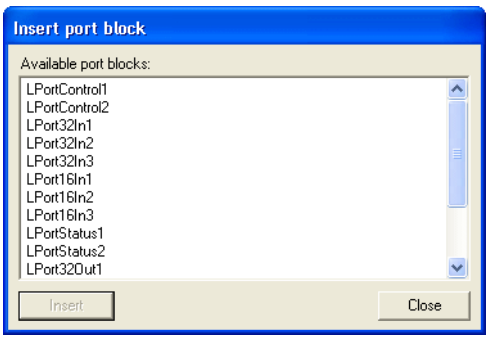
All input/output ports defined for the application on the **Ports** tab can be inserted into the interconnection in the form of port blocks in order to receive access to the associated element variables.

Tip!

You can change between the **Ports** and **FB Editor** tabs at any time to define new ports and afterwards insert them into the interconnection.

How to insert a port block into the interconnection:

1. Click the icon in the *FB editor toolbar*.
   - The *Insert port block* dialog box appears:



2. Select the desired port in the **Available port blocks** list field.
   - Ports which have already been inserted in the interconnection no longer are provided for selection in the **Available port blocks** list field.
3. Press **Insert** button.
   - The dialog box is closed and the port selected is inserted in the interconnection.

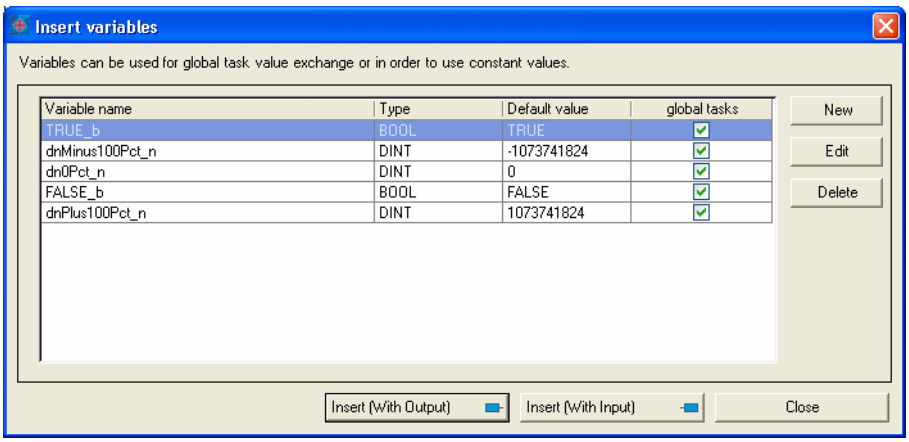
Related topics:

- Deleting objects that are no longer required (📖 70)
- Arranging objects in the drawing area (📖 73)
- Creating/deleting connections (📖 74)

3.4.2.4 Creating/inserting a variable

Click the icon in the *FB Editor toolbar* to open the *Insert variables* dialog box.

- The list field contains all variables created for the application:

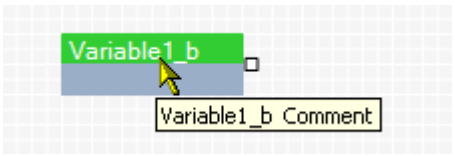


- The buttons serve to execute the following functions:

Button	Function
New	Create new variable.
Edit	Edit selected variable.
Delete	Delete selected variable. • You can only delete a variable if it is not used anymore in the interconnection.
Insert (with output)	Insert selected variable with output into the interconnection and close dialog box.
Insert (with input)	Insert selected variable with input into the interconnection and close dialog box. • For a value assignment (write access), a variable can be inserted exactly once into the interconnection. • If the selected variable is already included in the interconnection with input, this button is deactivated.
Close	Close dialog box without inserting a variable into the interconnection.

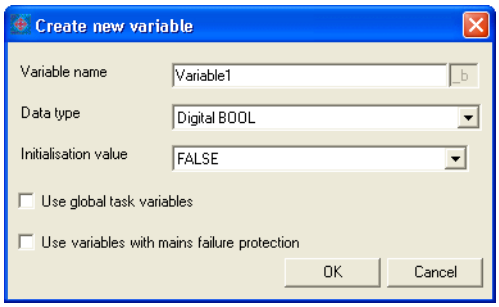
Tip!

As of the »Engineer« V2.10 a comment can be inserted for variables when being declared. The comment is then shown as "tooltip" in the drawing area if you keep the mouse pointer for a short time on the variable block:



How to create a new variable:

1. Go to the *Insert variables* dialog box and press the **New** button.
   - The *Create new variable* dialog box for the declaration of a new variable is displayed:



2. If required, change the variable name preselected in the **Variable name** input field into a more significant name.
   - The variable name must not already be allocated within the interconnection.
3. Select the data type for the variable from the **Data type** list field.
   - In accordance with IEC 61131-3 integer data types can be selected.
   - The data types are prenamed according to their designation.
4. If required, adapt the value preselected in the **Initialisation value** input field the variable is to be initialised with.
   - The initialisation value must be given in the internal quantity.
   - Signal types & scaling (📖 29)
5. Optionally enter a comment on the variable in the **Comment** input field.
6. Activate the **Use global task variables** control field if the variable is to be used in several tasks.
7. Activate the **Use variables with mains failure protection** if the value is to be saved with mains failure protection.
8. Press **OK**.
   - The *Create new variable* dialog box is closed and the newly created variable is listed in the *Insert variables* dialog box.

Insert a variable into the interconnection

A variable contained in the list field can be inserted exactly once with an input (for value assignment) into the interconnection and as often as required with an output (for read access).



How to insert a variable into the interconnection:

1. Go to the *Insert variables* dialog box and select the variable to be inserted in the list field.
2. Press the **Insert (with output)** or **Insert (with input)** button to insert a variable with input or output into the interconnection.



Note!

Variables are only assigned at the end of the task cycle! Thus, the variable value will only be effective in the next task cycle!

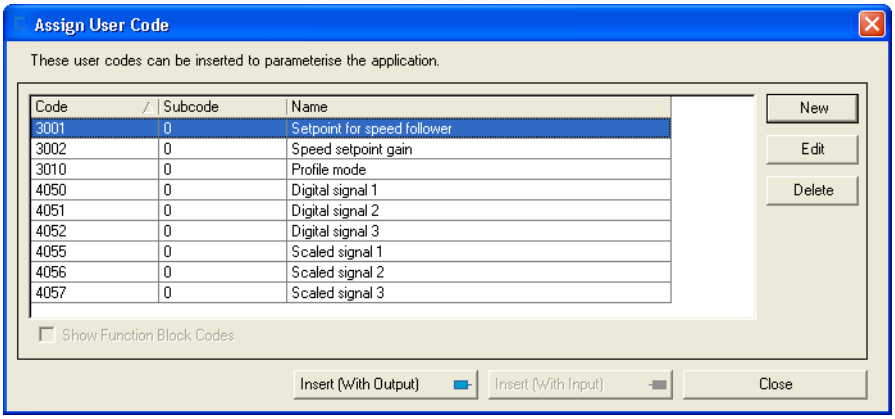
Related topics:

- Deleting objects that are no longer required (📖 70)
- Deleting variables and user codes from the application (📖 71)
- Arranging objects in the drawing area (📖 73)
- Creating/deleting connections (📖 74)

3.4.2.5 Configuring/inserting a user code

Click the icon in the *FB Editor toolbar* to open the *Assign User Code* dialog box.

- The list field contains all configured user codes created for the application:

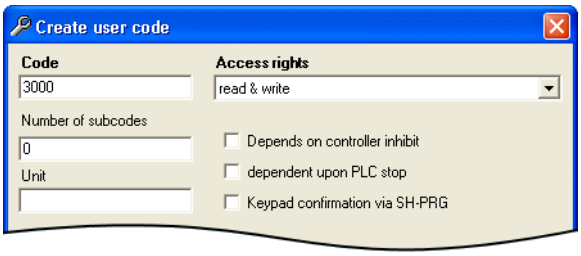


- The buttons serve to execute the following functions:

Button	Function
New	Configure new user codes.
Edit	Edit selected user code.
Delete	Delete selected user code. • You can only delete a user code if it is not used anymore in the interconnection.
Insert (with output)	Insert selected user code with output into the interconnection and close dialog box.
Insert (with input)	Insert selected user code with input into the interconnection and close dialog box. • For a value assignment (write access), a user code can be inserted exactly once into the interconnection. • If the selected user code is already included in the interconnection with input, this button is deactivated.
Close	Close dialog box without inserting a user code into the interconnection.

How to configure a new user code:

1. Go to the *Assign User Code* dialog box and press the **New** button.
   - The *Create user code* dialog box for configuring a new user code is displayed.



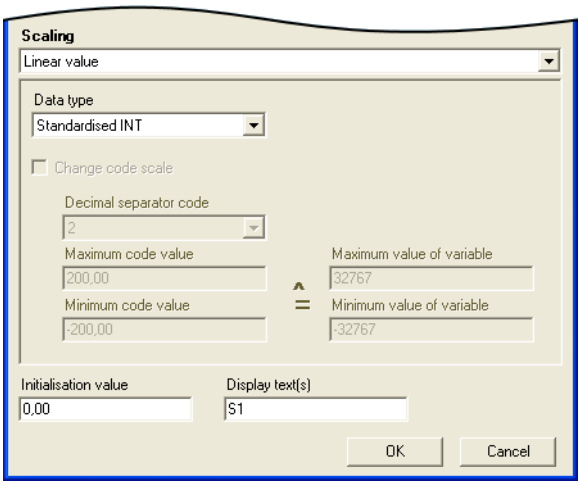
2. Configure the properties of the user code.
   - See the following section "Properties of a user code".
3. Press **OK**.
   - The *Create user code* dialog box is closed and the newly configured user code is listed in the *Assign User Code* dialog box.

Properties of a user code

Input field	Information	
Code	Code number • The next free code number is already entered into the input field. • The range of 3000 ... 9999 is available for user codes. • The code number must not already be allocated within the application.	
Access rights	Selection whether only read or also write access to user code should be possible. For a user code with write access, the following properties can be configured in addition:	
	Depends on controller inhibit	Set a checkmark in this control field if the user code should only be parameterised when the controller is inhibited.
	Dependent upon PLC stop	Set a checkmark into this control field if the user code should only be parameterised when the application is stopped.
	Keypad confirmation via SH-PRG	Set a checkmark into this control field if the parameterisation of the user code should be confirmed using the keypad.
Number of subcodes	Number of subcodes (0 ... 255) • If subcodes have been defined, the initialisation values and display texts are configured in the *Initialisation values* dialog box, which you open via the **Subcodes...** button.	
Unit	Optional text • The unit is displayed in the »Engineer« parameter list and in the keypad for the user code.	
Scaling	Selection whether the user code is to be displayed with a linear value or a selection list. • Depending on the selection, the other possible settings for the user code change.	

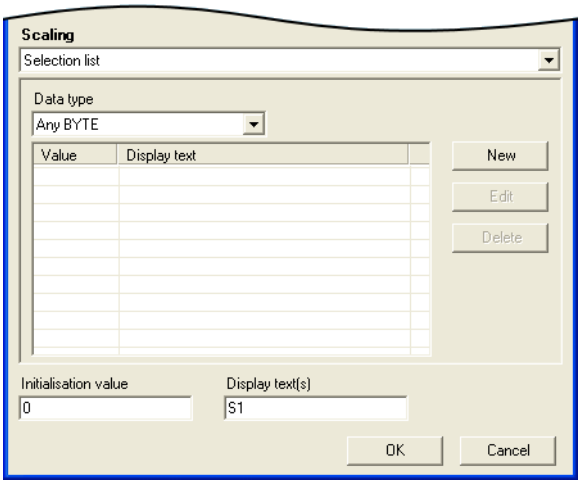
Input field	Information
Initialisation value	Value the user code is initialised with. • If subcodes have been defined, the initialisation values are configured in the *Initialisation values* dialog box instead, which you open via the **Subcodes...** button.
Display text(s)	Any display text for the user code • The display text is displayed in the FB Editor in the header of the user code, in the »Engineer« parameter list, and in the keypad for the user code. • If subcodes have been defined, the display texts are configured in the *Initialisation values* dialog box instead, which you open via the **Subcodes...** button.

User code with "Linear value" scaling



Input field	Information
Data type	Data type • The available data types are prenamed according to their designation.
Change code scaling	Set a checkmark in this control field if you want to change the preset scaling of the user code. • Then you can change the preset scaling and the number of decimal positions via the activated input fields. • The preset scaling can only be changed for the "Other BYTE/ WORD/ DWORD/INT/DINT" data types.

User code with "selection list" scaling



Input field	Information
Data type	Data type • The available data types are prenamed according to their designation.
Value	The value assigned to the selection
Display text	Any display text for the selection • The text is displayed in the »Engineer« parameter list and in the keypad in the selection list for the user code.

Button	Function
New	Create a new entry for the selection list.
Edit	Edit selected entry.
Delete	Delete selected entry from the selection list.

Insert a user code into the interconnection

A user code contained in the list field can be inserted exactly once with input (for value assignment) into the interconnection and as often as required with one output (for read access).

How to insert a user code into the interconnection:

1. Go to the *Assign User Code* dialog box and select the user code to be inserted in the list field.
2. Press the **Insert (with output)** or **Insert (with input)** button to insert a user code with input or output into the interconnection.

Context menu for user code

If you right-click on the header of a user code, a *context menu* opens via which you can execute the following functions in addition to the general processing functions (Copy, Insert, Delete):

	Command	Function
	Changing a value	Display/change value of the user code.

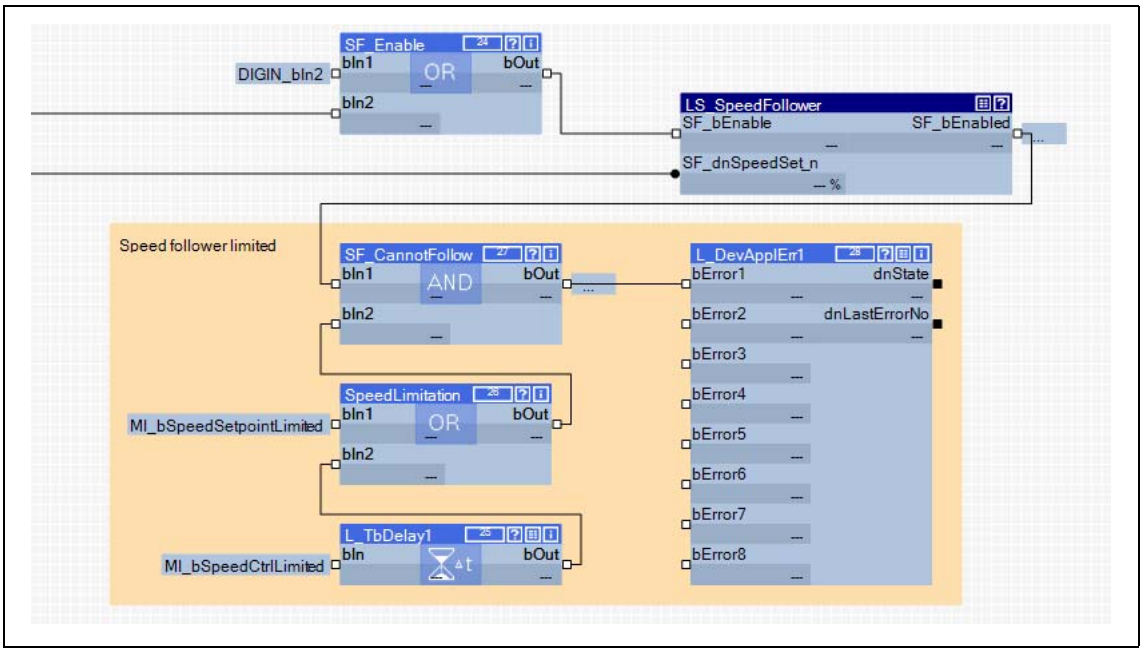
Related topics:

- Deleting objects that are no longer required (📖 70)
- Deleting variables and user codes from the application (📖 71)
- Arranging objects in the drawing area (📖 73)
- Creating/deleting connections (📖 74)

3.4.2.6 Inserting a comment

Comments can be entered into any position in the drawing area.

As of the »Engineer« V2.10 the interior colour and text alignment of a comment can be changed via a properties dialog. Now the sizes of comments can also be changed using the mouse pointer. When using different interior colours you can use comments to graphically arrange areas that belong together in terms of function or separate them from other areas:



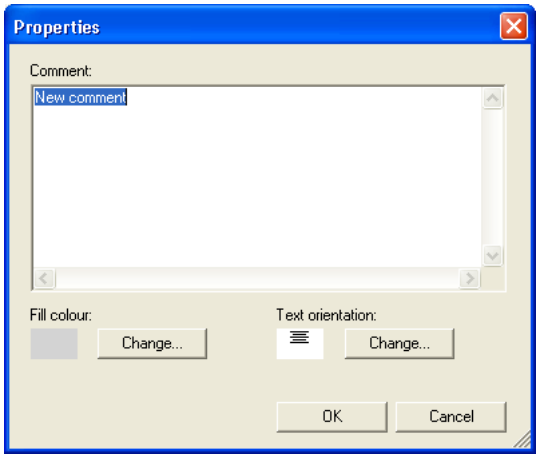
[3-2] Example: Graphical arrangement of FBs by means of a comment

Note!

The term "Arrangement" does not mean a logical arrangement of the function blocks. The comments are only graphical presentation elements of the FB Editor.

How to insert a new comment into the interconnection:

1. Move the mouse pointer to the (free) position in the drawing area where the comment is to be inserted.
2. Select the **New comment** command in the *context menu* (right mouse button).
   - The *Properties* dialog is displayed:



3. Enter the required comment into the text field.
4. Optional: Change preset interior colour.
   - For this purpose, click the left **Change...** button to open the *Colour* dialog box to select another interior colour.
5. Optional: Change preset text alignment.
   - For this purpose, click the right **Change...** button to open the *Text alignment* dialog box to select another interior colour.
6. Press **OK** to close the *Properties* dialog box and insert the comment.
   - After being inserted, the corner points of the comments are shown:



7. Optional: Change size of the comment.
   - For this purpose click one of the corner points with the left mouse button and enlarge the comment to the required size with the mouse button pressed.



8. Optional: Drag comment.
   - For this purpose click the comment with the left mouse button and move the comment to the required position with the mouse button pressed.



Tip!

The *Properties* dialog box for a comment already available can be opened by double-clicking the comment.

Related topics:

- Deleting objects that are no longer required (📖 70)
- Arranging objects in the drawing area (📖 73)
- Creating/deleting connections (📖 74)

3.4.2.7 Deleting objects that are no longer required

Objects that are no longer required can be easily deleted again. "Delete" only means that the object is removed from the drawing area. If you have deleted an object from the drawing area, you can reinsert it any time into the interconnection.

Note!

Deleting an object cannot be undone.

Together with the object, all available connections to this object are deleted.

How to delete objects that are no longer required:

1. Select objects to be deleted.
   - You can select a single object by clicking the header of the object.
   - You can select objects that are placed together by drawing a frame around these objects while keeping the mouse button pressed.
   - If you click the header of further objects while pressing **<Ctrl>**, these will be added to an already existing selection (multi-selection).
   - All selected objects are highlighted by a light green header.
2. Press **<Del>**.

Related topics:

- Deleting variables and user codes from the application (📖 71)
- Deleting connections that are no longer required (📖 80)

3.4.2.8 Deleting variables and user codes from the application

Variables and user codes that are no longer required can be also deleted completely from the application.



How to delete a variable completely from the application:

1. Delete all occurrences of the variable in the interconnection.
2. Click the icon to open the *Insert variables* dialog box.
3. Select the variable from the list field.
4. Press **Delete**.
5. Confirm the query whether the variable is to be deleted irrevocably with **OK**.
6. Press **Close** to close the *Insert variables* dialog box again.



How to delete a user code completely from the application:

1. Delete all occurrences of the user code in the interconnection.
2. Click the icon to open the *Assign User Code* dialog box.
3. Select the user code from the list field.
4. Press **Delete**.
5. Confirm the query whether the user code is to be deleted irrevocably with **OK**.
6. Press **Close** to close the *Assign User Code* dialog box again.

Related topics:

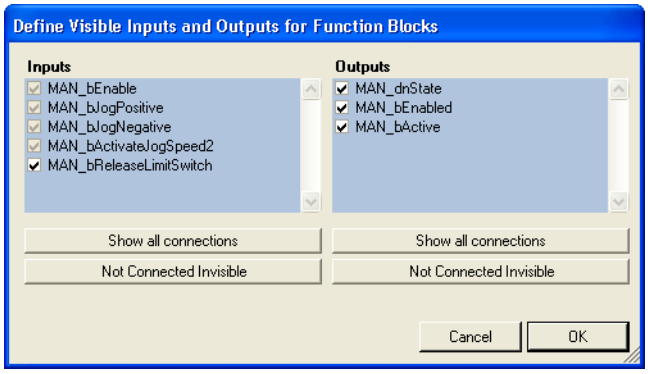
- Deleting objects that are no longer required (📖 70)
- Deleting connections that are no longer required (📖 80)

3.4.3 Changing connector visibilities

Inputs and outputs that are not connected can be hidden for each function and system block. This serves to reduce the dimension of the block. The interconnection becomes clearer.

How to define the visible inputs and outputs:

1. Go to the context menu of the block and select the **Connector visibilities** command.
   - The *Define Visible Inputs and Outputs for Function Blocks* is displayed:



   - All visible connections have a checkmark.
   - In case of a block that is inserted anew, all inputs and outputs are visible at first.
   - Inputs and outputs with a light grey checkbox are already connected and thus cannot be hidden.
2. By setting/removing the checkmarks or via the buttons you can define the visible inputs and outputs.
3. Press **OK** to accept the selected definition and close the dialog box.

3.4.4 Arranging objects in the drawing area

All objects can be freely arranged in the drawing area by dragging with the mouse.

We recommend to make an arrangement in which the required connections between the inputs and outputs can be created easily. A division into functional areas may also be sensible to get a better understanding of the application.

Objects which are already connected, can also be dragged to another (free) position in the drawing area. The available connections will be automatically re-routed after dragging.



How to drag an object:

1. Click the header of the object (and keep the button pressed).
2. Keep the button pressed and drag the object to the required position in the drawing area.
   - Via **<Esc>** you can cancel this action.



How to drag several objects at the same time:

1. Select the objects to be dragged.
   - You can select a single object by clicking the header of the object.
   - If you click the header of further objects while pressing **<Ctrl>**, these will be added to an already existing selection (multi-selection).
   - You can easily select objects that are placed together by drawing a frame around these objects while keeping the mouse button pressed.
   - All selected objects are highlighted by a light green header.
2. Keep the mouse button pressed on the header of one of the selected objects and drag it to the required position in the drawing area.
   - Via **<Esc>** you can cancel this action.



Note!

A red header indicates that the object overlaps with other objects in the drawing area!

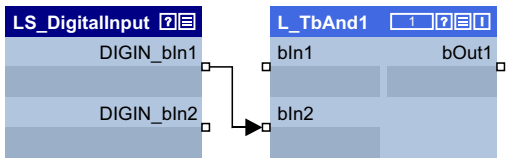
Arrange the objects so that no overlap occurs.

3.4.5 Creating/deleting connections

After adding objects and arranging them in a reasonable manner within the drawing area, you can create the connections between the available objects which are required for the desired function.

A connection always has a direction and therefore always has a source and a target.

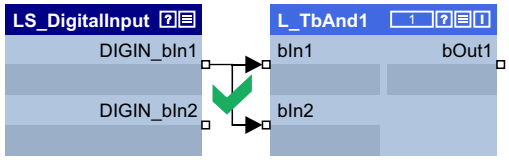
- An output represents a possible source in the interconnection.
- An input represents a possible target in the interconnection.



Permissible/impermissible connections

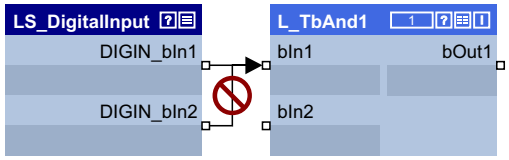
Several connections can lead from one output.

- Therefore it is always possible to start a new connection from an output.



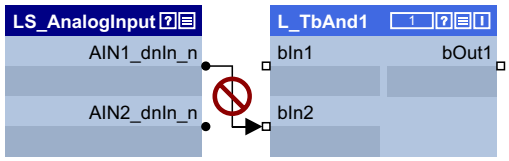
However, maximally one connection may end in an input.

- Therefore it is only possible to start a new connection from an input if there is no connection already ending in this input.



Only inputs/outputs of the same signal type (with the same port symbol) can be connected.

- Non-standardised inputs/outputs can be connected if the input and output have the same resolution.

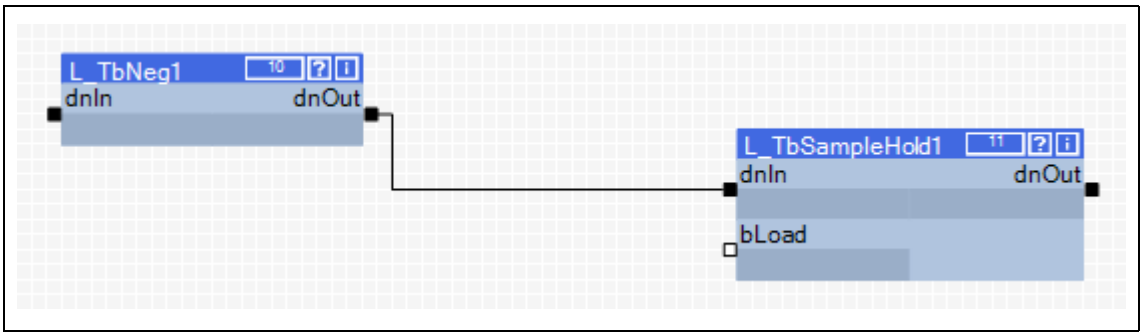


Tip!

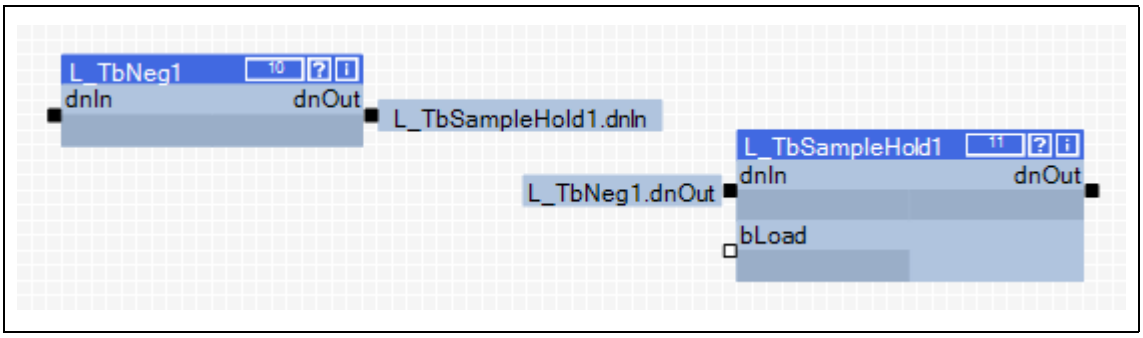
From »Engineer« V2.9, conversion blocks requiring additional computing time are not needed anymore for type conversion purposes. Instead, type conversion is performed implicitly. ▸ Data type check/implicit type conversion (📖 30)

Connection types

Connections can either be created by means of connection lines or port identifiers ("flags")



[3-3] Example 1: Connection via connection line

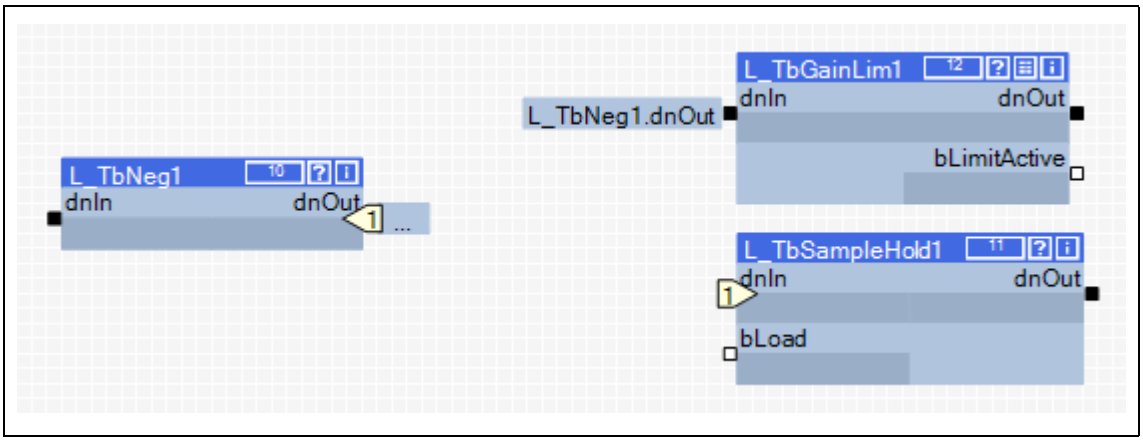


[3-4] Example 2: Connection via flags

Tip!

The commands **Show as flag** or **Show as line** in the *context menu* of a connection serve to change the representation of the connection at any time.

If both connection types are used for an output, the available connection lines are converted into small yellow flags with an automatically assigned number:



[3-5] Example 3: Connection of an output once with port identifier, once with connection line

- Three points ("...") are shown at the output instead of the concrete input identifier.
- The *context menu* for the port symbol contains all inputs which are connected to the output.

3.4.5.1 Creating a connection using the connection line

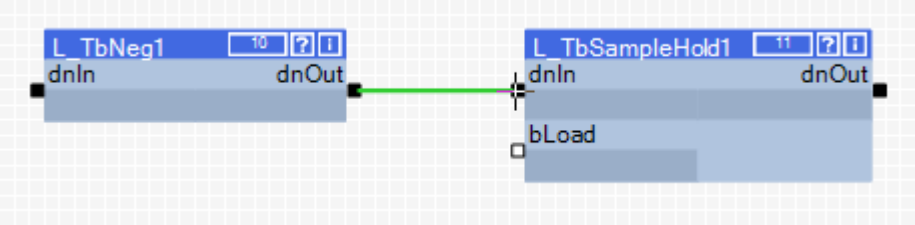
How to create a connection using the connection line:

1. Click the port symbol from which the new connection is to be started.
   - It is only possible to start a new connection from an input if there is no connection already ending in this input.
   - If you then move the mouse pointer away from the port symbol, a new connection is "drawn" from this port symbol.
   - Via **<Esc>** you can cancel this action.
2. Click the port symbol where the connection is to end.
   - Thereupon the corresponding connection is routed automatically if the connection is permissible.

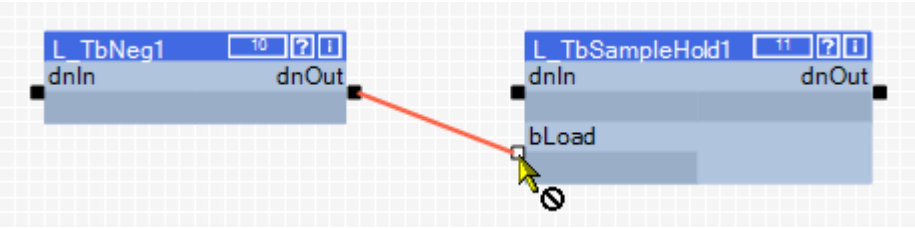
Tip!

If you move the mouse pointer across the port symbol while drawing a new connection, you can see whether the connection is permissible or not from the colour of the drawn line and from the mouse pointer symbol.

- Permissible connection:



- Impermissible connection (different port symbol):



The command **Show as flag** in the *context menu* of a line serves to change the representation of the connection at any time.

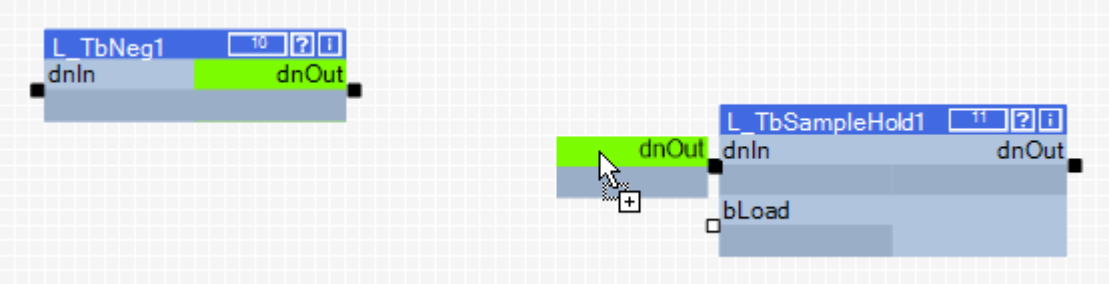
3.4.5.2 Creating a connection using port identifiers

How to create a connection with port identifiers:

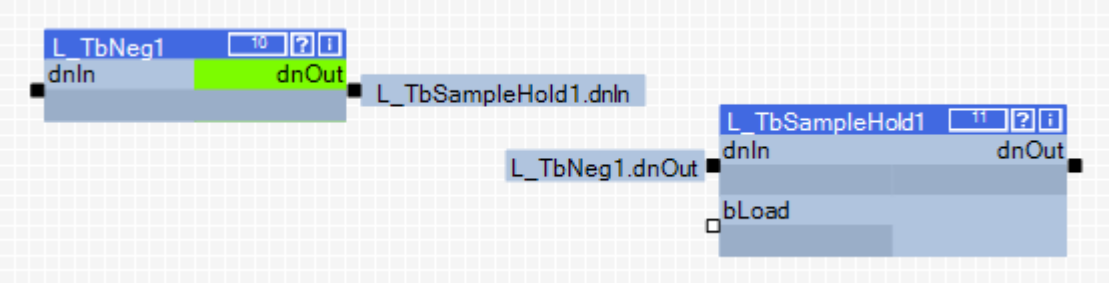
1. Click the port identifier.
   - The selected port is highlighted in light green:



2. Drag the port segment to the required port while keeping the left mouse button pressed:



After releasing the mouse button, the connection via port identifiers (flags) is created. The corresponding port identifier consists of the instance name and the name of the input/output:



Tip!

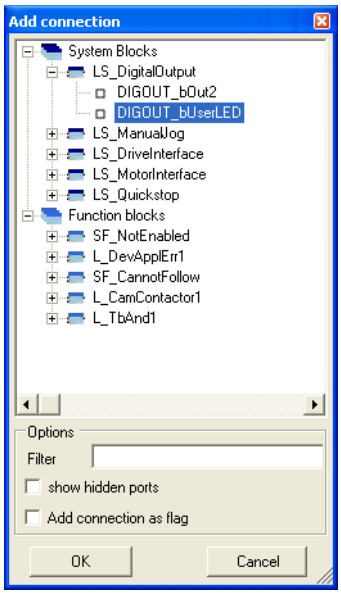
The command **Show as line** in the *context menu* of a flag serves to change the representation of the connection at any time.

3.4.5.3 Creating a connection via connection dialog

You can also create connections by means of a selection dialog instead of dragging by mouse. This especially makes sense if there is a great distance between the ports to be connected in the drawing area.

How to create a connection using the selection dialog:

1. Right-click the port identifier or click the port symbol from which the connection is to start.
   - The *context menu* for the port is displayed.
2. Go to the *context menu* for the port and select the **Add connection...** command.
   - The *Add connection* dialog box is displayed:



   - In a tree structure all inputs and outputs of the application are shown to which a connection is permissible.
   - You can enter an optional text into the **Filter** input field to reduce the selection to the blocks or ports which contain the entered text.
   - If you activate the **Show hidden ports** control field, the hidden ports for system and function blocks are shown as well.
3. Select the port where the connection is to end from the tree structure.
4. Activate the **Add connection as flag** control field if a port identifier (flag) is to be inserted instead of a connection line.
5. Press **OK** to create the connection to the selected port and close the dialog box.

3.4.5.4 Deleting connections that are no longer required



How to delete connection lines:

1. Select connection lines to be deleted.
   - Select a single connection line by directly clicking on the connection line with the right mouse button.
   - If you click further connection lines while pressing **<Ctrl>** they are added to an already existing selection (multi-selection).
   - All connection lines are highlighted in red.
2. Press **<Del>**.



How to delete port identifiers/flags:

1. Select the port identifiers to be deleted.
   - Select a single port identifier by directly clicking on the port identifier with the left mouse button.
   - If you click further port identifiers while pressing **<Ctrl>** they are added to an already existing selection (multi-selection).
   - All selected port identifiers are highlighted by a light green header.
2. Press **<Del>**.

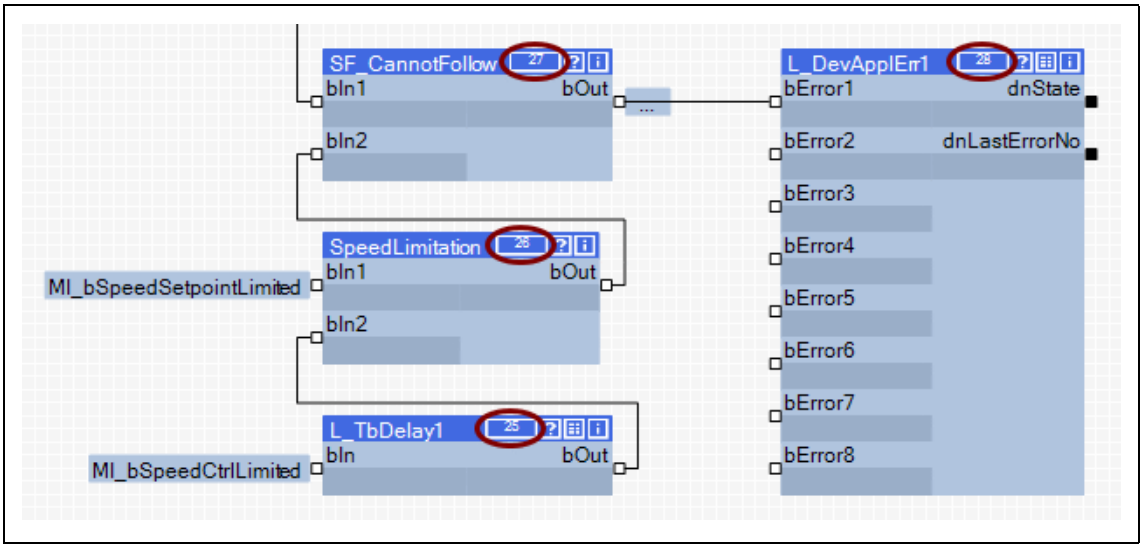
Related topics:

▸ Deleting objects that are no longer required (📖 70)

3.4.6 Changing the processing order

If you insert a function block into the interconnection, an order index is automatically assigned to this function block. By means of this order index it is defined in which order the individual function blocks are calculated with regard to the runtime.

- The first function block inserted contains the order index "1", the next function block inserted contains the order index "2", etc.
- The respective order index is displayed in the header of the function block in the rectangle after the instance name.



[3-6] Example: Function blocks with order index 25 ... 28

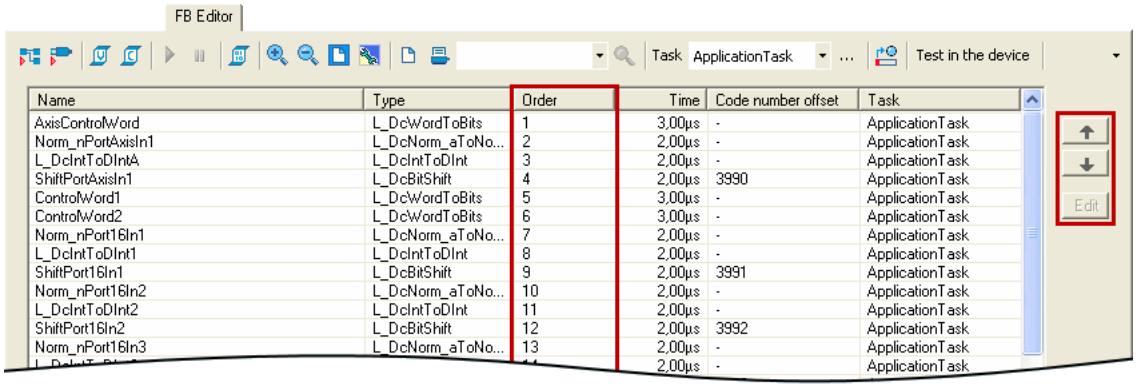
Note!

When a function block is shifted, its order index is maintained.

The order of processing has an impact on the result and has to be altered in specific cases.

How to change the processing order:

1. Use the list field at the top right to change from the Editor to the overview.
   - The overview displays all function blocks, system blocks, and port blocks of the interconnection in the order of their processing
   - In the "Order" column the order index of each function block is listed.

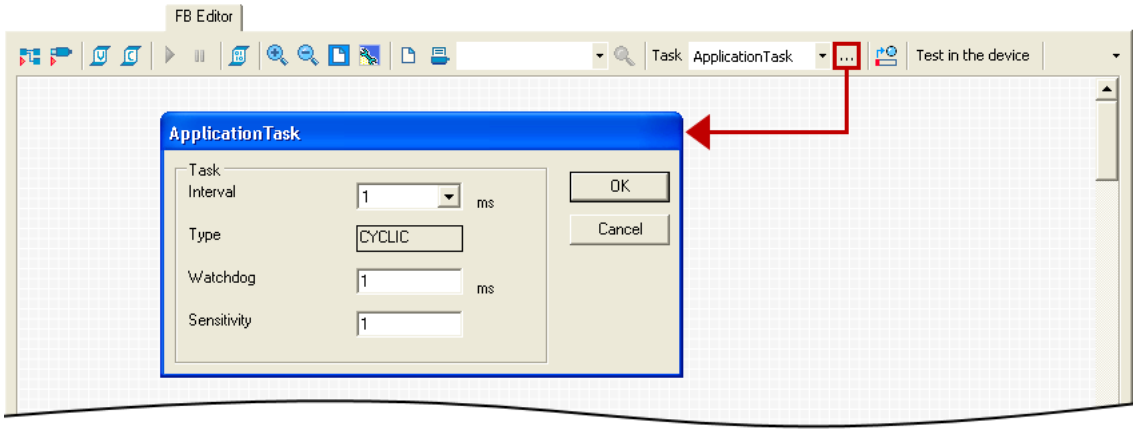


Name	Type	Order	Time	Code number offset	Task
AxisControlWord	L_DcWordToBits	1	3,00µs	-	ApplicationTask
Norm_nPortAxisIn1	L_DcNorm_aToNo...	2	2,00µs	-	ApplicationTask
L_DcIntToDIntA	L_DcIntToDInt	3	2,00µs	-	ApplicationTask
ShiftPortAxisIn1	L_DcBitShift	4	2,00µs	3990	ApplicationTask
ControlWord1	L_DcWordToBits	5	3,00µs	-	ApplicationTask
ControlWord2	L_DcWordToBits	6	3,00µs	-	ApplicationTask
Norm_nPort16In1	L_DcNorm_aToNo...	7	2,00µs	-	ApplicationTask
L_DcIntToDInt1	L_DcIntToDInt	8	2,00µs	-	ApplicationTask
ShiftPort16In1	L_DcBitShift	9	2,00µs	3991	ApplicationTask
Norm_nPort16In2	L_DcNorm_aToNo...	10	2,00µs	-	ApplicationTask
L_DcIntToDInt2	L_DcIntToDInt	11	2,00µs	-	ApplicationTask
ShiftPort16In2	L_DcBitShift	12	2,00µs	3992	ApplicationTask
Norm_nPort16In3	L_DcNorm_aToNo...	13	2,00µs	-	ApplicationTask
			2,00µs	-	ApplicationTask

2. Select the function block which is to receive a different position within the processing order.
3. Move the function block to the desired position using the ↑ and ↓ buttons.
   - The Edit button serves to directly assign another order index to the selected function block per dialog.

3.4.7 Configuring the task properties

By clicking the **...** button right beside the **Task** list field, you open a dialog box where you can see and, if required, change the properties (interval, type, watchdog, sensitivity) of the selected task.



Monitoring of the task runtime

The runtime of the UserTask and ApplicationTask is monitored by the controller. A "watchdog" is used for monitoring by continuously checking if both tasks are executed correctly within the defined time frame.

- The **Watchdog** input field serves to set the maximum runtime for the selected task.
- The **Sensitivity** input field serves to define how many times the set maximum runtime of the task can be exceeded before the monitoring mode is activated.
  - The sensitivity is provided with a counter which is reset to zero when the current runtime of the task is within the set maximum runtime again.
- When the monitoring mode of the task runtime is activated:
  - The "Error" response takes place in the Lenze setting, i.e. pulse inhibit and controller inhibit are set. The error response can be parameterised in C02111.
  - The error message "ApplicationTask: Overflow" or "UserTask: Overflow" is entered.

Note!

The runtime of the IdleTask is not monitored by the controller.

Related topics:

- Multitasking in the Servo Drive 9400 (📖 32)
- Task selection (📖 43)

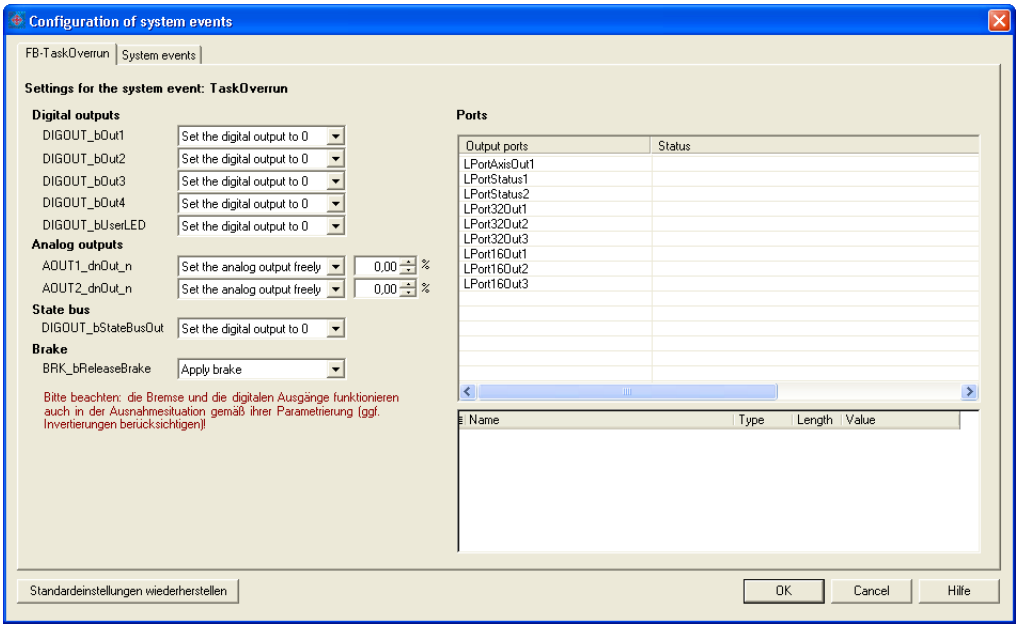
3.4.8 Configuring the exception handling

This function will be available as of »Engineer« V2.10!

For the 9400 HighLine controller, the behaviour after a task overflow can be configured for the analog and digital outputs, the brake control, and the output ports and thus be adapted to the corresponding application.

How to configure the exception handling:

1. Go to the *FB Editor toolbar* and click the icon to open the *Exception handling* dialog box:



   - The **FB-TaskOverrun** tab serves to configure the behaviour of the outputs of the controller and the output ports defined in the application in case of a task overflow.
   - In the **FB System Events** tab, the behaviour of the outputs of the drive controller and the application are only displayed and cannot be configured.

2. Carry out the desired configuration.
   - Each output can be configured individually. A free value can be set for the analog outputs (-200.00 ... 200.00 %).
   - If you select an output port in the "Ports" area on the right, all application variables for this output port are shown in the table below. In the "Value" column a value can be set for each application variable to which it is to be set if an event occurs can be specified.
   - If a value has been set for at least one application variable, the status "Exceptional behaviour parameterised" is shown for the corresponding output port.
   - If the **Restore standard settings** button is clicked, the default setting for task overflow is restored. In this case, all output terminals would be set to LOW level or 0 V in the event of a task overflow and the output ports would retain their last value.

3. Click **OK** to accept the configuration and close the dialog box.

Danger!

In case a task overflow occurs, the brake can be configured to "open". This setting should be used with care as the brake is then forcibly opened and does not close even if the drive controller is inhibited!

Note!

- To render the changes effective within the controller, the project has to be updated, and the changed application has to be transferred to the controller.
- During the reset or download of an application, all output signals are set to LOW level or 0 V for a short time (the state bus, in contrast, is set to HIGH level due to harware inversion).

3.4.9 Compiling the completed interconnection

How to compile the completed interconnection:

1. Click the icon in the *FB editor toolbar*.
   - The *Update project* dialog box appears.
2. Press the **Create** button to update the project.
   - After the update is completed successfully, a corresponding note appears.

3.4.10 Testing the interconnection in the device

This function will be available as of »Engineer« V2.9!

This function serves to transfer the compiled interconnection for testing to the volatile memory (RAM) of the controller after the FB interconnection has changed.

Since only the binary file and the connection table are transmitted to the device instead of the complete application, the transfer is quicker than with the regular function "Download program".

Note!

This function will be supported as of controller firmware V5.0.

After mains switching, the original application is reloaded which is stored in the memory module.

In order to store the application, which has been transferred to the volatile memory (RAM) for test purposes, completely and with mains failure protection in the memory module, the regular function "Download program" must be executed. Thus, the corresponding »Engineer« project is required to save the tested application with mains failure protection.

To check the current function block interconnection in the device:
Go to the *FB Editor toolbar* and click the **Test in the device** button.

Sequence

1. If the current status of the FB Editor does not correspond to the device update executed last, first the available function for updating a single device is called.
   - This generates all files to be transferred to the device.
   - A note may appear that first the networks should be updated.
2. If there was no online connection to the device yet, it will be established now.
3. A note appears that the controller is automatically inhibited for this function and the application is stopped and the application will be automatically started after the transfer and the controller will be enabled.
4. The »Engineer« checks if the function is supported by the device and if the memory available in the device is sufficient.
   - If not, the function is closed with a corresponding note. In this case, use the regular function "Download program".
5. The controller is inhibited and the function is stopped.
6. The application is transferred to the volatile memory of the device.
7. After successful download:
   - The transferred application is started automatically and the controller is enabled.
   - The value "99" is displayed in C00007 (active application).
   - The "Application program volatile in the device" is displayed in the *FB Editor status bar* .



Tip!

When an online connection has been established, the parameters of the application stored in the volatile memory can be read and changed using the »Engineer« as usual if the corresponding »Engineer« project is open.

3.4.11 Copying interconnection elements (across all devices)

This function will be available as of »Engineer« V2.9!

As of the »Engineer« V2.9, interconnection elements cannot only be copied within the same interconnection but also across all devices within the same project, as long as the devices stem from the same product family (e.g. Servo Drives 9400).

All types of blocks and comments can be copied to the clipboard via the **Copy** command or the **<Ctrl>+<c>** shortcut and then be inserted into the FB interconnection of the same or another project device of the same product family using the **Insert** command or the **<Ctrl>+<v>** shortcut.

- During the copy process into the clipboard, existing connections between copied blocks are copied as well, and the layout is kept too. Moreover, the separate technical objects (port definition, variable declaration, and user code declaration) are copied. Selected connections cannot be copied on their own.
- The **Insert** command is available if the clipboard is not empty and if it was copied from a device of the same product family. Within this product family, all device types (e.g. 9400 xxxxLine Vxx.xx) are permitted.
- After the **Insert** command has been selected, a dialog box is displayed which serves to select which elements are to be inserted from the clipboard and how to solve name conflicts, if any.
- After inserting the elements, they are marked in the target interconnection in order to be repositioned or deleted again to undo the insertion.
- Inserting from the clipboard can be repeated. The originally copied contents of the clipboard remains unchanged when it is inserted.



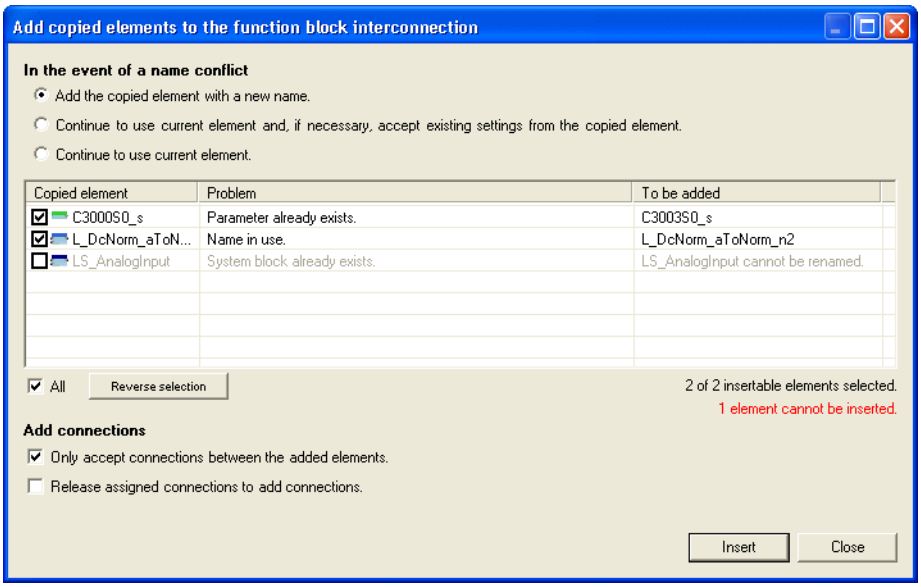
How to copy one or several interconnection elements:

1. Select the objects to be copied.
   - You can select a single object by clicking the header of the object.
   - If you click the header of further objects while pressing **<Ctrl>**, these will be added to an already existing selection (multi-selection).
   - You can easily select elements that are placed together by drawing a frame around these elements while keeping the mouse button pressed.
   - All selected objects are highlighted by a light green header.
2. Go to the *context menu* and select the **Copy** command (or **<Ctrl>+<c>**).
   - The selected elements are copied into the clipboard of the FB Editor.
3. If the elements are to be copied into a function block interconnection of another project device, change to the corresponding interconnection via the *project view*.
4. Go to the *context menu* and select the **Insert** command (or **<Ctrl>+<v>**).
5. Go to the *Insert copied elements into the FB interconnection* and select which elements from the clipboard are to be inserted and how to solve name conflicts, if any.
   - Detailed information on this dialog box can be obtained from the following subchapter "Insert options for copied elements". (📖 90)

6. Click **Insert** to insert the selected elements into the target interconnection as defined.
   - Only possible if at least one element in the list has been selected for insertion.
   - Insertion is also possible via the **<Enter>** button if at least one element is selected in the list for insertion.
   - The original layout and the relative position of the inserted blocks to each other are maintained.
   - When copying across the devices, you also insert the corresponding separate technical objects (port definition, variable declaration and user code declaration).
   - The inserted elements are deleted from the list. If the list is empty, the dialog box is closed and the connections are inserted depending on the selected option.
7. If there are still elements to be entered in the list, repeat steps 5 and 6 until all elements are inserted as intended.
8. Press **Close** to stop the insertion and close the dialog box.
   - You can also use **<Esc>** or **<Enter>** to close the dialog box if "Insert" is not active.
   - The elements inserted into the target interconnection so far are maintained.
   - The connections for the blocks inserted so far are inserted depending on the selected option.

3.4.11.1 Insert options for copied elements

If interconnection elements have been copied to the clipboard, the »Engineer« displays a list of all elements contained in the clipboard after the **Insert** command has been inserted in the *Add copied elements to the function block interconnection* dialog box:



The list shows the elements which can be added to the target interconnection without any further action, and the elements with problems and those which cannot be added.

- Each element only appears once in the list even if there are several problems.
- If there is at least one name conflict, there are three options of how to solve name conflicts. If the option is changed, the list is updated accordingly.
- The first column "Copied element" serves to define which elements are to be inserted by setting/resetting checkmarks.
  - In the default setting, all elements to be inserted are always selected.
  - When you deselect the **All** option, all elements in the list are deselected.
  - The **Reverse selection** button serves to reverse the existing selection.
- The second column "Problem" shows the currently existing problems.
- The third column "To be added" contains changes for conflicts with already available elements in the target interconnection to be able to accept the elements. The automatic suggestions depend on the selected option how to solve name conflicts.

- Connections are only inserted when the dialog box is closed, which applies to all modules inserted so far. They are displayed as lines or flags, like in the original, but re-routed.
  - When the option **Only accept connections between the added elements** is selected, only those connections are added which are between the inserted elements. If this option is deselected, also those connections are added which are between the added elements and external elements outside the selection if an element of the same type and instance name is available in the target interconnection.
  - When the **Release assigned connections ...** option is selected, connections are deleted in the target interconnection to be able to add connections from the clipboard to inputs already assigned.
- The buttons serve to execute the following functions:

Button	Function
Insert	Add elements selected in the list to the target interconnection • Only possible if at least one element in the list has been selected for insertion. • Insertion is also possible via the **<Enter>** button if at least one element is selected in the list for insertion. • The original layout and the relative position of the inserted blocks to each other are maintained. • When copying across the devices, you also insert the corresponding separate technical objects (port definition, variable declaration and user code declaration). • The inserted elements are deleted from the list. If the list is empty, the dialog box is closed and the connections are inserted depending on the selected option.
Close	Close dialog box. • You can also use **<Esc>** or **<Enter>** to close the dialog box if "Insert" is not active. • The elements inserted into the target interconnection so far are maintained. • The connections for the blocks inserted so far are inserted depending on the selected option.

Tip!

It is not compulsory to add all elements to be inserted in one step. If required, you can first add single elements and in further worksteps other elements of the list with another option for solving name conflicts.

3.5 Printing the interconnection

The interconnection can be printed for documentation purposes, optionally on one page, on four pages, or not scaled.

Tip!

By clicking the icon in the *FB Editor toolbar*, you can get a print view before printing.

How to print the interconnection:

1. Click the icon in the *FB editor toolbar*.
   - The *Circuit print size* dialog box is displayed.
2. Select the desired size and press **OK**.
   - The standard dialog box *Print* appears.
3. Press **OK** to start the printing process.

3.6 Comparing interconnections

This function will be available as of »Engineer« V2.9!

The comparison operation serves to compare function block interconnections for the device series 9400 within the project.

- In the project, it is distinguished between the status of the FB interconnection at the device update updated last ("last compiled") and the current processing status in the FB Editor ("processing").
- An offline<>online comparison and the comparison of two online devices are possible.
- The comparison result can be output as simple text. This can be copied to the clipboard and used for documentation purposes.

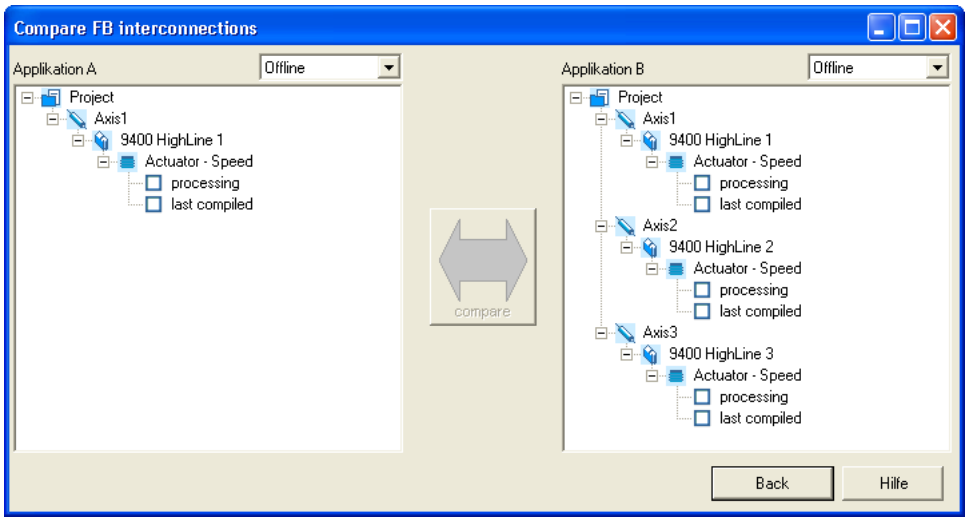
Note!

Only applications can be compared which have been enabled in the FB editor!

Parameter values, comments, module positions, line representations, and connector visibilities are not compared.

How to compare two FB interconnections:

1. Select the command **Application data→Compare FB interconnections...**.
   - The *Compare FB interconnections* dialog box is displayed:



   - For comparing two interconnections, the left project view displays **Application A** with only one selected device.
   - The right project view with **Application B** selected for the comparison shows all devices available in the project.
2. Select the applications to be compared in the project view represented on the left and right.
   - In order to execute a comparison with an online device, select "Online" in one of the two upper list fields. Then all available online devices are displayed for selection.
   - If you select "Online" in one of the two upper list fields, you can also compare the interconnections of two available online devices.

3. Click **Compare**.
   - If the comparison was executed successfully, the comparison result is displayed as simple text. ▸ Contents and representation of the comparison (📖 96)
   - If a comparison of the selected application is not possible, a corresponding message is displayed.
   - The dialog box remains open for a new selection.
4. In order to stop the comparison operation and close the dialog box: Press **Back**.

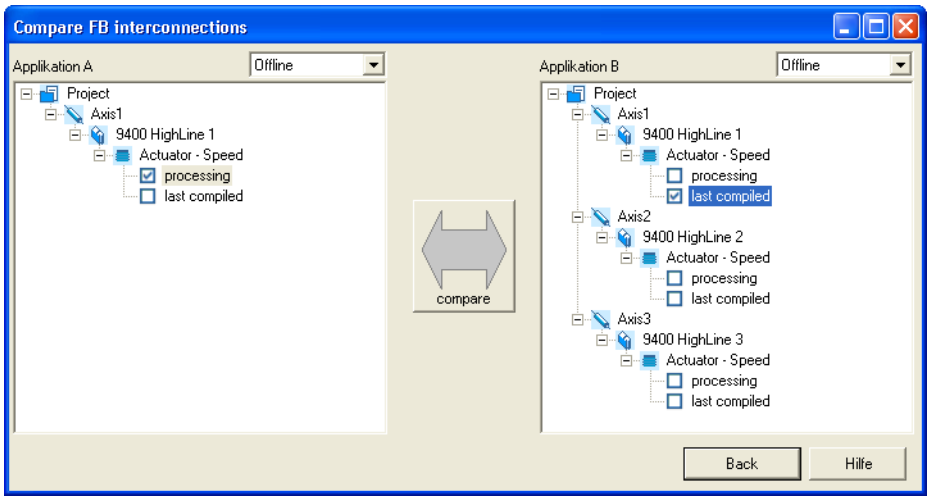
3.6.1 Comparison examples

The following comparisons can be executed among each other:

- FB interconnection in the Engineer (offline, processing)
- FB interconnection in the Engineer (offline, last compiled)
- FB interconnection in the controller (online)

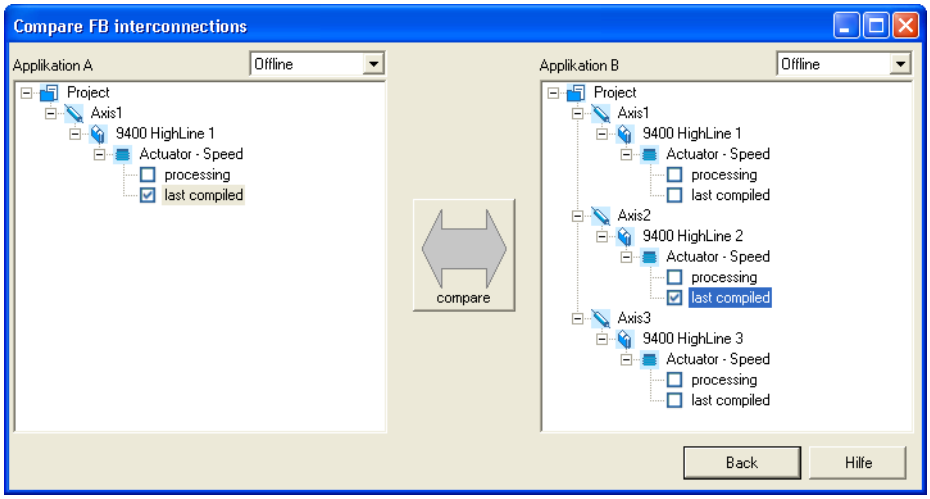
Example 1: Device A (processing) <--> device A (last compiled)

Comparison of the current processing status in the FB Editor with the status of the FB interconnection with the last device update:



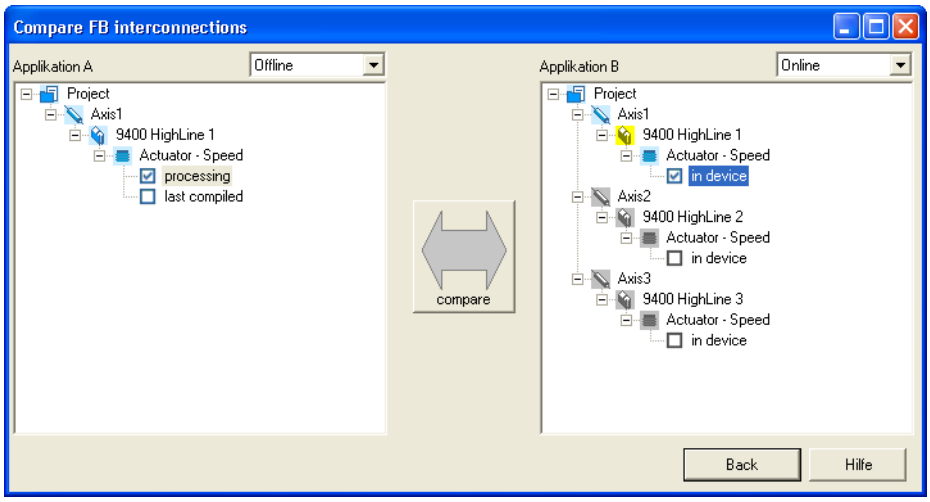
Example 2: Device A (last compiled) <--> device B (last compiled)

Comparison of FB interconnections of two different devices in the project:



Example 3: Device A (processing) <--> real device A (in the memory module)

Comparison of the current processing status in the FB Editor with the FB interconnection in the real device (in the memory module):



3.6.2 Contents and representation of the comparison

The following elements of an FB interconnection are considered in a comparison. The features of the elements to be compared are listed as well:

Element	Properties
Task	Name Cycle time Type
Function blocks to be instanced freely	Processing sequence Library version Instance name Type
System blocks	Name Version
Port blocks	Name Data types of the element variables
Variable declarations (local & global)	Variable name Data type
Variable assignments	Number
User parameter declarations	Name Code number Data type Setting range (lower limit, upper limit) Unit Variable instance name Variable setting range (lower limit, upper limit) Write/read permissions Dependencies on the controller inhibit and Plc-stop
User parameter assignments	Number Code number
Signal flows (connection lines)	Signal sources and target

Every object available in an FB interconnection is checked for existence in the other interconnection. If the object concerned is available but with another declaration or version, the deviating features are displayed by values in the report.

Example: The freely instanced module A is available in the project and the device but has been taken from a different library version in the device. Both version numbers are presented next to each other in the report in order to recognise the deviation.

3.7 Copying complete interconnection

This function will be available as of »Engineer« V2.9!

In contrast to copying/inserting selected interconnection elements via the clipboard, the function described in this chapter serves to replace the current FB interconnection of a device completely by the FB interconnection of another project device.

Note!

The complete FB interconnection can only be copied between devices of the same device type and version (e.g. 9400 HighLine V3).

A complete interconnection comprises:

- ApplicationTask, UserTask and IdleTask:
  - Function blocks (instances and parameter values)
  - System blocks (application and parameter values)
  - Port blocks (application)
  - Variables (variable declaration and value)
  - User codes (user code declaration and value)
  - Connections (fixed connections, MUXer and their setting)
  - Comments
  - Interconnection layout (arrangement of the modules)
- Global configuration:
  - Activation status
  - Task configuration
- Port definition of the ports used in the FB interconnection

How to copy the complete interconnection into another project device:

1. Select the application with the FB interconnection to be copied in the *project view*.
2. Select the command **Application data→Copy FB interconnections...**.
3. Go to *project view* and select the application which is to be inserted into the copied FB interconnection.
4. Select the command **Application data→Add FB interconnection...**.
   - The command can only be activated if an FB interconnection has been copied from a device of the same device type and version.
   - After the command has been executed, the module assembly is compared. If there are relevant deviations, the insertion is refused and a corresponding message is displayed.
   - If an insertion is possible, you are asked if the FB interconnection is to be inserted.
5. Confirm the question if the copied FB interconnection is to be inserted with **Yes**.
   - After the insertion, an update of the project is required.

3.8 Inserting complete interconnection from reference project

This function will be available as of »Engineer« V2.9!

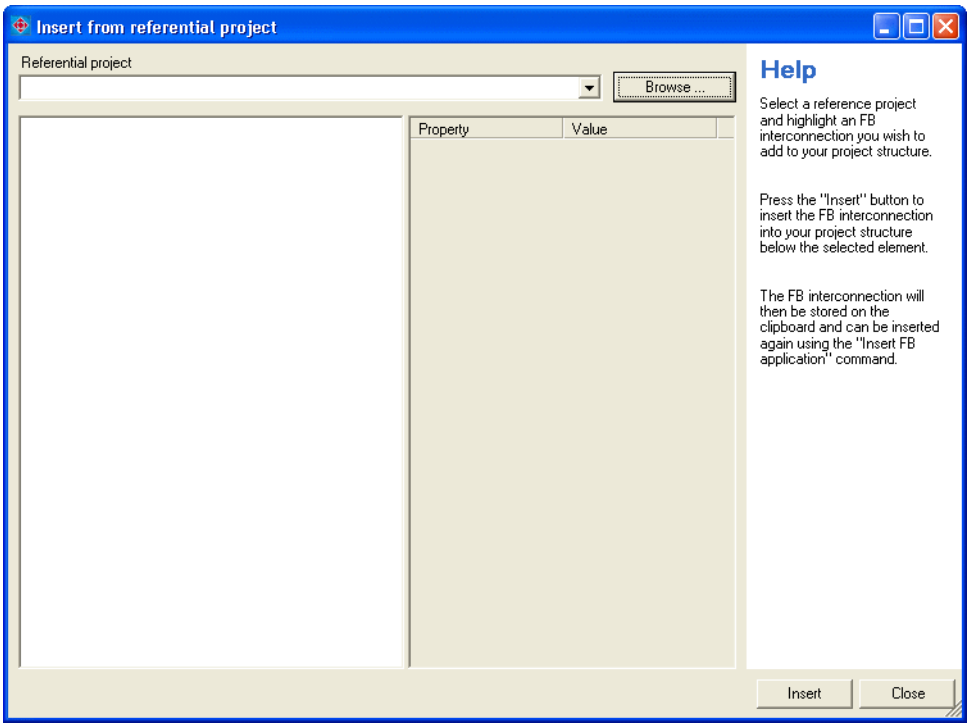
The function described in this chapter serves to replace the current FB interconnection of the device completely by the FB interconnection of a reference project.

Note!

Inserting from a reference project is only possible if the device in the reference project has the same device type and version as the project device (e.g. 9400 HighLine V3).

How to insert the complete interconnection from a reference project:

1. Go to the *project view* and select the command **Insert FB interconnection from referential project...** in the *context menu* of the application in which the FB interconnection of a referential project is to be inserted.
   - The *Insert from referential project* dialog box appears:



2. Select reference project.
   - Option 1: Press **Search...** to select the required reference project in the desktop environment in the *Open reference project* dialog box.
   - Option 2: Direct selection in the **Reference project** list field. This list field shows the reference projects already selected for quick selection.
   - After the selection, a project tree with all components of the selected reference project is displayed.
3. Select the application with the FB interconnection to be copied in the project tree.

4. Press **Insert** button.
   - The button can only be activated if the device in the reference project has the same device type and version as the project device.
   - After the button has been pressed, the module assembly is compared. If there are relevant deviations, the insertion is refused and a corresponding message is displayed.
   - If an insertion is possible, you are asked if the FB interconnection is to be inserted.
5. Confirm the question if the copied FB interconnection is to be inserted with **Yes**.
   - The FB interconnection is inserted and the *Insert from referential project* dialog box is closed.
   - After the insertion, an update of the project is required.

4 Short overview

4.1 Analog signal processing

Function block	Function	From library
L_SdDelayComp	Extrapolation • The FB extrapolates a signal by the time set.	LenzeServoDrive V01.00.xx.xx
L_SdProcessController	Comprehensive PID controller with setpoint and actual value processing as well as parameterisable characteristic function and limitation	LenzeServoDrive V02.05.xx.xx
L_Tb8Select	1-out-of-8 selector (for data type "DINT") • The FB connects one of eight input signals through to the output.	LenzeToolbox V01.00.xx.xx
L_Tb8SelectByte	1-out-of-8 selector (for data type "BYTE") • The FB connects one of eight input signals through to the output.	LenzeToolbox V02.05.xx.xx
L_Tb8SelectWord	1-out-of-8 selector (for data type "WORD") • The FB connects one of eight input signals through to the output.	LenzeToolbox V02.05.xx.xx
L_TbAbs	Absolute value generation • The function block converts a bipolar signal to a unipolar signal, i.e. the absolute value of the input signal is generated.	LenzeToolbox V01.00.xx.xx
L_TbCompare	Comparison • The function block compares two signals of the "DINT" data type and can be used e.g. for implementing a trigger.	LenzeToolbox V01.00.xx.xx
L_TbCompare_n	Comparison (scaled) • The function block compares two scaled signals and can be used e.g. for implementing a trigger.	LenzeToolbox V01.00.xx.xx
L_TbCurve	Characteristic function (scaled) • The FB produces the corresponding value of the Y axis for the X axis value with a parameterisable characteristic function y = f(x).	LenzeToolbox V01.00.xx.xx
L_TbCurveAny	Characteristic function • The FB produces the corresponding value of the Y axis for the X axis value with a parameterisable characteristic function y = f(x).	LenzeToolbox V02.06.xx.xx
L_TbDeadband	Dead band • The FB produces a symmetrical dead band around zero.	LenzeToolbox V01.00.xx.xx
L_TbDeadband_n	Dead band (scaled) • The FB produces a symmetrical dead band (scaled) around zero.	LenzeToolbox V01.00.xx.xx
L_TbDifferentiate	Differentiator with low pass	LenzeToolbox V01.00.xx.xx
L_TbGainLim	Gain with limitation	LenzeToolbox V01.00.xx.xx
L_TbLimit	Signal limitation (for data type "DINT") • The FB limits a signal of "DINT" data type to an adjustable value range.	LenzeToolbox V01.00.xx.xx
L_TbLimit_n	Signal limitation (scaled) • The FB limits a scaled signal [%] to an adjustable value range.	LenzeToolbox V01.00.xx.xx

Function block	Function	From library
L_TbLimitInt	Signal limitation (for data type "INT") • The FB limits a signal of "INT" data type to an adjustable value range.	LenzeToolbox V02.05.xx.xx
L_TbMaskOut	Zone masking • The FB masks up to four parameterisable blocking zones within a continuous signal characteristic.	LenzeToolbox V01.00.xx.xx
L_TbNeg	Negation (for data type "DINT") • The function block inverts the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output.	LenzeToolbox V01.00.xx.xx
L_TbNegInt	Negation (for data type "INT") • The function block inverts the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output.	LenzeToolbox V02.05.xx.xx
L_TbNegSel	Optional negation (for data type "DINT") • The function block inverts (optionally) the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output. • In contrast to the L_TbNeg FB, this FB is only inverted if the *bNeg* control input is set to TRUE.	LenzeToolbox V02.02.xx.xx
L_TbNegSelInt	Optional negation (for data type "INT") • The function block inverts (optionally) the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output. • In contrast to the L_TbNegInt FB, this FB is only inverted if the *bNeg* control input is set to TRUE.	LenzeToolbox V02.05.xx.xx
L_TbNormalize	Signal scaling • The FB scales any signal to a parameterisable reference variable.	LenzeToolbox V01.00.xx.xx
L_TbPIController	Simple PI controller with different control functions and parameterisable limitation	LenzeToolbox V01.00.xx.xx
L_TbPT1Filter	PT1 filter • The FB filters and delays analog signals.	LenzeToolbox V01.00.xx.xx
L_TbRateAction	Rate action function to compensate disturbing low passes	LenzeToolbox V01.00.xx.xx
L_TbSampleHold	Sample & Hold (for data type "DINT") • The FB stores a value of the "DINT" data type in order to keep it even after mains switching.	LenzeToolbox V01.00.xx.xx
L_TbSampleHoldWord	Sample & Hold (for data type "WORD") • The FB stores a value of the "WORD" data type in order to keep it even after mains switching.	LenzeToolbox V02.05.xx.xx
L_TbSelect	Selector (for data type "DINT") • The function block switches between two signals of "DINT" data type. The switch-over is controlled via a boolean input signal.	LenzeToolbox V01.00.xx.xx
L_TbSelectByte	Selector (for data type "BYTE") • The function block switches between two signals of "BYTE" data type. The switch-over is controlled via a boolean input signal.	LenzeToolbox V02.02.xx.xx
L_TbSelectWord	Selector (for data type "WORD") • The function block switches between two signals of "WORD" data type. The switch-over is controlled via a boolean input signal.	LenzeToolbox V02.05.xx.xx

4.2 Digital signal processing

Function block	Function	From library
L_TbCount	Upcounter and downcounter with adjustable limitation	LenzeToolbox V01.00.xx.xx
L_TbDelay	Binary delay element	LenzeToolbox V01.00.xx.xx
L_TbFlipFlopD	D flipflop • Can be used for evaluation and storage of digital signal edges.	LenzeToolbox V01.00.xx.xx
L_TbFlipFlopRS	RS flipflop • Can be used for evaluation and storage of digital signals.	LenzeToolbox V01.00.xx.xx
L_TbOscillator	Rectangular pulse generator with parameterisable pulse/dead time	LenzeToolbox V01.00.xx.xx
L_TbTransition	Signal edge evaluation • Can be used to generate timed and retriggerable pulses.	LenzeToolbox V01.00.xx.xx

Logic operations for "BOOL" data type

Function block	Logic operation	From library
L_TbAnd	AND (2 inputs)	LenzeToolbox V01.00.xx.xx
L_Tb5And	AND (5 inputs)	LenzeToolbox V01.00.xx.xx
L_TbOr	OR (2 inputs)	LenzeToolbox V01.00.xx.xx
L_Tb5Or	OR (5 inputs)	LenzeToolbox V01.00.xx.xx
L_TbNand	NOT-AND (2 inputs)	LenzeToolbox V01.00.xx.xx
L_Tb5Nand	NOT-AND (5 inputs)	LenzeToolbox V01.00.xx.xx
L_TbNor	NOT-OR (2 inputs)	LenzeToolbox V01.00.xx.xx
L_Tb5Nor	NOT-OR (5 inputs)	LenzeToolbox V01.00.xx.xx
L_TbXor	EXCLUSIVE-OR (2 inputs)	LenzeToolbox V01.00.xx.xx
L_Tb5Xor	EXCLUSIVE-OR (5 inputs)	LenzeToolbox V01.00.xx.xx
L_TbNot	Negation (inverter)	LenzeToolbox V01.00.xx.xx

Bit operations for "INT" data type

Function block	Function	From library
L_DcBitShiftInt	Bit shift operation	LenzeDataConversion V02.02.xx.xx

Bit operations for "DINT" data type

Function block	Function	From library
L_DcBitShift	Bit shift operation	LenzeDataConversion V01.00.xx.xx

Bit operations for "BYTE" data type

Function block	Function	From library
L_DcBitShiftByte	Bit shift operation	LenzeDataConversion V02.02.xx.xx
L_DcByteBitand	Bit-by-bit AND operation of two BYTE values.	LenzeDataConversion V02.02.xx.xx
L_DcByteBitor	Bit-by-bit OR operation of two BYTE values.	LenzeDataConversion V02.02.xx.xx
L_DcByteBitxor	Bit-by-bit EXCLUSIVE-OR operation of two BYTE values.	LenzeDataConversion V02.02.xx.xx
L_DcByteToBits	Bit demultiplexer • BYTE → BOOL (8 x)	LenzeDataConversion V01.00.xx.xx
L_DcGetBitOfByte	Bit operation • Output a status of a single bit.	LenzeDataConversion V01.00.xx.xx
L_DcSetBitOfByte	Bit operation • Set a single bit to "1".	LenzeDataConversion V01.00.xx.xx
L_DcResetBitOfByte	Bit operation • Reset a single bit to "0".	LenzeDataConversion V01.00.xx.xx

Bit operations for "WORD" data type

Function block	Function	From library
L_DcBitShiftWord	Bit shift operation	LenzeDataConversion V02.02.xx.xx
L_DcWordBitand	Bit-by-bit AND operation of two WORD values.	LenzeDataConversion V02.02.xx.xx
L_DcWordBitor	Bit-by-bit OR operation of two WORD values.	LenzeDataConversion V02.02.xx.xx
L_DcWordBitxor	Bit-by-bit EXCLUSIVE-OR operation of two WORD values.	LenzeDataConversion V02.02.xx.xx
L_DcWordToBits	Bit demultiplexer • WORD → BOOL (16 x)	LenzeDataConversion V01.00.xx.xx
L_DcGetBitOfWord	Bit operation • Output a status of a single bit.	LenzeDataConversion V01.00.xx.xx
L_DcSetBitOfWord	Bit operation • Set a single bit to "1".	LenzeDataConversion V01.00.xx.xx
L_DcResetBitOfWord	Bit operation • Reset a single bit to "0".	LenzeDataConversion V01.00.xx.xx

Bit operations for "DWORD" data type

Function block	Function	From library
L_DcBitShiftDWord	Bit shift operation	LenzeDataConversion V02.02.xx.xx
L_DcDWordToBits	Bit demultiplexer • DWORD → BOOL (32 x)	LenzeDataConversion V01.00.xx.xx
L_DcGetBitOfDWord	Bit operation • Output a status of a single bit.	LenzeDataConversion V01.00.xx.xx
L_DcSetBitOfDWord	Bit operation • Set a single bit to "1".	LenzeDataConversion V01.00.xx.xx
L_DcResetBitOfDWord	Bit operation • Reset a single bit to "0".	LenzeDataConversion V01.00.xx.xx
L_DcDWordBitand	Bit-by-bit AND operation of two DWORD values.	LenzeDataConversion V01.00.xx.xx

Function block	Function	From library
L_DcDWordBitor	Bit-by-bit OR operation of two DWORD values.	LenzeDataConversion V01.00.xx.xx
L_DcDWordBitxor	Bit-by-bit EXCLUSIVE-OR operation of two DWORD values.	LenzeDataConversion V01.00.xx.xx

4.3 Data type conversion

Function block	Function	From library
L_DcBitsToByte	Bit multiplexer • BOOL (8 x) → BYTE	LenzeDataConversion V01.00.xx.xx
L_DcBitsToWord	Bit multiplexer • BOOL (16 x) → WORD	LenzeDataConversion V01.00.xx.xx
L_DcBitsToDWord	Bit multiplexer • BOOL (32 x) → DWORD	LenzeDataConversion V01.00.xx.xx
L_DcByteToBits	Bit demultiplexer • BYTE → BOOL (8 x)	LenzeDataConversion V01.00.xx.xx
L_DcByteToInt	Type converter • BYTE → INT	LenzeDataConversion V02.02.xx.xx
L_DcByteToWord	Type converter • BYTE → WORD	LenzeDataConversion V01.00.xx.xx
L_Dc2BytesToWord	Type converter • BYTE (2 x) → WORD	LenzeDataConversion V01.00.xx.xx
L_Dc4BytesToDWord	Type converter • BYTE (4 x) → DWORD	LenzeDataConversion V01.00.xx.xx
L_DcIntToDInt	Type converter • INT → DINT	LenzeDataConversion V01.00.xx.xx
L_DcIntToWord	Type converter • INT → WORD	LenzeDataConversion V01.00.xx.xx
L_DcDIntToInt	Type converter • DINT → INT	LenzeDataConversion V01.00.xx.xx
L_DcDIntToDWord	Type converter • DINT → DWORD	LenzeDataConversion V01.00.xx.xx
L_DcWordToBits	Bit demultiplexer • WORD → BOOL (16 x)	LenzeDataConversion V01.00.xx.xx
L_DcWordTo2Bytes	Type converter • WORD → BYTE (2 x)	LenzeDataConversion V01.00.xx.xx
L_DcWordToByte	Type converter • WORD → BYTE	LenzeDataConversion V01.00.xx.xx
L_DcWordToInt	Type converter • WORD → INT	LenzeDataConversion V01.00.xx.xx
L_DcWordToDWord	Type converter • WORD → DWORD	LenzeDataConversion V01.00.xx.xx
L_Dc2WordsToDWord	Type converter • WORD (2 x) → DWORD	LenzeDataConversion V01.00.xx.xx
L_DcDWordToBits	Bit demultiplexer • DWORD → BOOL (32 x)	LenzeDataConversion V01.00.xx.xx
L_DcDWordTo4Bytes	Type converter • DWORD → BYTE (4 x)	LenzeDataConversion V01.00.xx.xx

Function block	Function	From library
L_DcDWordToDInt	Type converter • DWORD → DINT	LenzeDataConversion V01.00.xx.xx
L_DcDWordToWord	Type converter • DWORD → WORD	LenzeDataConversion V01.00.xx.xx
L_DcDWordTo2Words	Type converter • DWORD → WORD (2 x)	LenzeDataConversion V01.00.xx.xx

4.4 Mathematical functions

Function block	Mathematical function	From library
L_TbAbs	Absolute value generation	LenzeToolbox V01.00.xx.xx
L_TbAdd	Addition without limitation	LenzeToolbox V01.00.xx.xx
L_TbAddLim	Addition with limitation	LenzeToolbox V01.00.xx.xx
L_TbAddSubLim	Addition and subtraction with limitation	LenzeToolbox V02.04.xx.xx
L_TbDiv	Division with remainder	LenzeToolbox V01.00.xx.xx
L_TbDiv_n	Division divisor selected as scaled signal [%]	LenzeToolbox V01.00.xx.xx
L_TbIntegrate	Integration with limitation	LenzeToolbox V01.00.xx.xx
L_TbMul	Multiplication without limitation	LenzeToolbox V01.00.xx.xx
L_TbMul_n	Multiplication with the multiplicand selected as scaled signal [%]	LenzeToolbox V01.00.xx.xx
L_TbMulDivLim	Multiplication and division with limitation	LenzeToolbox V02.02.xx.xx
L_TbMulLim	Multiplication with limitation	LenzeToolbox V01.00.xx.xx
L_TbNeg	Sign inversion	LenzeToolbox V01.00.xx.xx
L_TbNegSel	Sign inversion (optional)	LenzeToolbox V02.02.xx.xx
L_TbSub	Subtraction without limitation	LenzeToolbox V01.00.xx.xx
L_TbSubLim	Subtraction with limitation	LenzeToolbox V01.00.xx.xx

4.5 Conversion of physical units

Function block	Function	From library
L_DcNorm_aToNorm_n	Signal converter • 16-bit signal (scaled) → 32-bit signal (scaled)	LenzeDataConversion V01.00.xx.xx
L_DcNorm_nToNorm_a	Signal converter • 32-bit signal (scaled) → 16-bit signal (scaled)	LenzeDataConversion V01.00.xx.xx
L_DcNorm_nToSpeed_s	Signal converter • 32-bit signal (scaled) → 32-bit-speed signal	LenzeDataConversion V01.00.xx.xx
L_DcSpeed_vToSpeed_s	Signal converter • 16-bit speed signal → 32-bit speed signal	LenzeDataConversion V01.00.xx.xx
L_DcSpeed_sToSpeed_v	Signal converter • 32-bit speed signal → 16-bit speed signal	LenzeDataConversion V01.00.xx.xx
L_DcSpeed_sToNorm_n	Signal converter • 32-bit speed signal → 32-bit signal (scaled)	LenzeDataConversion V01.00.xx.xx
L_SdDifferentiate	Differentiation (position → speed)	LenzeServoDrive V01.00.xx.xx
L_SdIntegrate	Integration (speed → position)	LenzeServoDrive V01.00.xx.xx

Function block	Function	From library
L_SdSetPosition	Position conversion • Position [unit] (selected as parameter value) → position [increments]	LenzeServoDrive V01.00.xx.xx
L_SdGetPosition	Position conversion • Position [increments] → position [unit] (displayed as parameter value)	LenzeServoDrive V01.00.xx.xx
L_SdPosToUnit	Position conversion • Position [increments] → position [unit]	LenzeServoDrive V01.00.xx.xx
L_SdUnitToPos	Position conversion • Position [unit] → position [increments]	LenzeServoDrive V01.00.xx.xx
L_SdSetSpeed	Speed conversion • Velocity [unit/s] (selected as parameter value) → speed	LenzeServoDrive V01.00.xx.xx
L_SdGetSpeed	Speed conversion • Speed → velocity [unit/s] (displayed as parameter value)	LenzeServoDrive V01.00.xx.xx
L_SdSpeedToUnit	Speed conversion • Speed → velocity [unit/s]	LenzeServoDrive V01.00.xx.xx
L_SdUnitToSpeed	Speed conversion • Velocity [unit/s] → speed	LenzeServoDrive V01.00.xx.xx
L_SdAccToUnit	Acceleration conversion • Speed variation/time → acceleration [unit/s^2]	LenzeServoDrive V01.00.xx.xx
L_SdUnitToAcc	Acceleration conversion • Acceleration [unit/s^2] → speed variation/time	LenzeServoDrive V01.00.xx.xx

4.6 Speed conditioning

Function block	Function	From library
L_SdDifferentiate	Differentiation (position → speed)	LenzeServoDrive V01.00.xx.xx
L_SdIntegrate	Integration (speed → position)	LenzeServoDrive V01.00.xx.xx
L_SdIntegrateAxis	Integration (speed → position) with TP correction	LenzeServoDrive V02.03.xx.xx
L_SdIntegrateLimit	Integration (speed → position) with adjustable upper/lower limit for return purposes	LenzeServoDrive V02.09.xx.xx
L_SdLimitSpeed	Speed limitation • Leads a speed setpoint to defined limit ranges.	LenzeServoDrive V01.00.xx.xx
L_SdSpeedFilter	Speed signal delay • Filters or decelerates speed signal according to the PT1 principle.	LenzeServoDrive V01.00.xx.xx
L_SdRampGenerator	Ramp generator (ramp function generator) with S-shaped ramps	LenzeServoDrive V01.00.xx.xx
L_SdRampGeneratorAny	Ramp generator (ramp function generator) with S-shaped ramps. • Unlike the L_SdRampGenerator FB, this FB is also able to process speeds (_s). Separate acceleration and deceleration times can be set.	LenzeServoDrive V02.09.xx.xx
L_SdSpeedSet	Ramp generator (ramp function generator) with S-shaped ramps, parameterisable fixed setpoints and comprehensive parameter setting and control options.	LenzeServoDrive V01.00.xx.xx

4.7 Special functions

Function block	Function	From library
L_DevApplErr	Error tripping • The FB can trip up to 8 different application messages out of the application. • Error ID and error response as well as the module ID can be set via parameters.	LenzeDevice9400 V01.00.xx.xx
L_DevApplErrFix	Error tripping • The FB can trip up to 8 different application messages out of the application. • Error ID and error response as well as the module ID can be set via FB inputs.	LenzeDevice9400 V02.00.xx.xx
L_DevParReadFix	One-time or cyclic reading of parameters • Only internal parameters (codes) of the controller can be read. • In contrast to the L_DevReadParDInt FB, the parameter of this FB to be read is addressed via FB inputs and not via parameters.	LenzeDevice9400 V02.00.xx.xx
L_DevParWriteFix	One-time or cyclic writing of parameters • Only internal parameters (codes) of the controller can be written. • In contrast to the L_DevWriteParDInt FB, the parameter of this FB to be written is addressed via FB inputs and not via parameters.	LenzeDevice9400 V02.00.xx.xx
L_DevReadParDInt	One-time or cyclic reading of parameters of "DINT" data type • Only internal parameters (codes) of the controller can be read.	LenzeDevice9400 V01.00.xx.xx
L_DevWriteParDInt	One-time or cyclic writing of parameters of the "DINT" data type • Only internal parameters (codes) of the controller can be written.	LenzeDevice9400 V02.00.xx.xx
L_DevSMControlEncoder	Control word for limiter • The FB generates the bit-coded control signal *dwControl* from individual boolean control signals for the SB **LS_Limiter** and thus allows control of the basic function "Limiter" if no SM3xx safety module is available.	LenzeDevice9400 V02.00.xx.xx
L_SdGetAxisData	Reading out machine parameters from axis data	LenzeServoDrive V02.07.xx.xx
L_SdInterpolate	Interpolation of position information • E.g. to compensate for longer bus transfer cycles or smooth low-resolution absolute value encoders.	LenzeServoDrive V02.00.xx.xx
L_SdInterExtrapolateAny	Interpolation/extrapolation of setpoints	LenzeServoDrive V02.00.xx.xx
L_SdInterExtrapolatePosition	Interpolation/extrapolation of position information	LenzeServoDrive V02.00.xx.xx
L_SdMotorPot	Motor potentiometer function • Can be used as an alternative setpoint source that is controlled via two inputs.	LenzeServoDrive V02.04.xx.xx
L_SdSetAxisData	Represented machine parameters of a master drive	LenzeServoDrive V01.00.xx.xx
L_SdTouchProbe	Touch probe processing • The FB takes over the interpolation of the input signal based on the time stamp handed over by a touch probe system block and outputs the interpolated value and the difference compared with the last input signal.	LenzeServoDrive V01.00.xx.xx

4.8 Output of status signals

The following FBs decode the status output signal *dnState*, which is made available by some system and function blocks, into single boolean status signals for further use in the FB interconnection:

Function block	Function	From library
L_DevDriveInterfaceStateDecoder	Status signals of the drive interface (SB **LS_DriveInterface**)	LenzeDevice9400 V01.00.xx.xx
L_DevManualJogStateDecoder	Status signals of the basic function "Manual jog" (SB **LS_ManualJog**)	LenzeDevice9400 V01.00.xx.xx
L_DevHomingStateDecoder	Status signals of the basic function "Homing" (SB **LS_Homing**)	LenzeDevice9400 V01.00.xx.xx
L_DevLimiterStateDecoder	Status signals of the basic function "Limiter" (SB **LS_Limiter**)	LenzeDevice9400 V01.00.xx.xx
L_DevBrakeStateDecoder	Status signals of the basic function "Brake control" (SB **LS_Brake**)	LenzeDevice9400 V01.00.xx.xx
L_DevPositionerStateDecoder	Status signals of the basic function "Positioning" (SB **LS_Positioner**) • From firmware 1.50, the FB L_DevPositionerStateDecoder replaces the FB L_PosDecoderStatePositioner.	LenzeDevice9400 V02.00.xx.xx
L_PosDecoderStatePositioner		LenzePositioning V01.00.xx.xx
L_PosSequencerStateDecoder	Status signals of the sequence control (FB L_PosSequencer) • From firmware 1.50, the FB L_PosSequencerStateDecoder replaces the FB L_PosDecoderStateSequencer.	LenzePositioning V02.00.xx.xx
L_PosDecoderStateSequencer		LenzePositioning V01.00.xx.xx
L_LdStateDecoder	Status signals of the FBs of the function library "LenzeLineDrive.lib"	LenzeLineDrive V01.00.xx.xx
L_DevSMControlDecoder	Control signals from the SM3xx safety module	LenzeDevice9400 V02.00.xx.xx
L_DevSMStateDecoder	Status signals from the SM3xx safety module	LenzeDevice9400 V02.00.xx.xx
L_DevSMStateDecoderIO	I/O status signals from the SM3xx safety module	LenzeDevice9400 V02.00.xx.xx

4.9 FBs for system via electrical shaft

General functions

Function block	Function	From library
L_EsEncoderConv	Encoder signal conditioning	LenzeElectricalShaft V01.00.xx.xx
L_EsStretchIntegrate	Stretching and compression of a master value in the "Modulo" or "Unlimited" traversing range	LenzeElectricalShaft V01.00.xx.xx

"Electronic cam" application

Function block	Function	From library
L_CamClutchPos	Clutch (path-controlled) • Clutch in or declutch a master position via a fifth order polynomial in a path-controlled manner.	LenzeCam V01.00.xx.xx
L_CamContactor	Cam group with switch-on and switch-off dynamisation	LenzeCam V01.00.xx.xx
L_CamCurve	Curve interpolation for characteristics	LenzeCam V01.00.xx.xx

Function block	Function	From library
L_CamGetAxisData	Reading out machine parameters from cam data	LenzeCam V01.00.xx.xx
L_CamPosMarker	Reading out position marks from cam data	LenzeCam V01.00.xx.xx
L_CamProfiler	Curve interpolation for motion profiles	LenzeCam V01.00.xx.xx
L_CamSetContDataPDO	Contactor data selection via inputs/PDOs • The inputs can be connected, for instance, to a corresponding port block to accept contactor data from a master control via process data objects (PDOs).	LenzeCam V01.02.xx.xx
L_CamSetContDataSDO	Contactor data selection via codes/SDOs • The codes can be written, for instance, via parameter data objects (SDOs) by a master control.	LenzeCam V01.02.xx.xx
L_CamStretchAbs	Stretch/compress position (absolute) • Simple absolute stretching and compression of a position in the limited traversing range.	LenzeCam V01.00.xx.xx
L_CamStretchFeed	Stretch/compress position (relative) • Relative stretching and compression of a position in the "Modulo" or "Unlimited" traversing range.	LenzeCam V01.00.xx.xx
L_CamSyncIn	Synchronous/oversynchronous clutch-in • Synchronising a tool (drive) to a running material (relative master speed).	LenzeCam V01.01.xx.xx
L_EsClutchPos	Clutch (time-controlled) • Clutch in or declutch a master position via a fifth order polynomial in a time-controlled manner.	LenzeElectricalShaft V01.01.xx.xx
L_LdVirtualMasterP	Virtual master for synchronism	LenzeLineDrive V01.00.xx.xx

Application "Electronic gearbox"

Function block	Function	From library
L_LdClutchV	Time-controlled virtual clutch with speed reference	LenzeLineDrive V01.00.xx.xx
L_LdExtrapolate	Extrapolation of position information • E.g. to compensate for longer bus transfer cycles or smooth low-resolution absolute value encoders.	LenzeLineDrive V01.00.xx.xx
L_LdMonitFollowError	Following error monitoring with adjustable switching threshold and hysteresis	LenzeLineDrive V01.00.xx.xx
L_LdMPot	Master value adjustment in positive/negative traversing direction with parameterisable speeds	LenzeLineDrive V01.00.xx.xx
L_LdSetAxisVelocity	Stretching/compression of the x axis and synchronisation via touch probe • Within the cycle, the master speed is integrated to the position.	LenzeLineDrive V01.00.xx.xx
L_LdStateDecoder	LineDrive status signals • Decoding of the status output signal *dnState* of an FB from the function library "LenzeLineDrive.lib" into individual boolean status signals for further use in the FB interconnection.	LenzeLineDrive V01.00.xx.xx
L_LdToolControl	Setpoint conditioning for the tool axis	LenzeLineDrive V01.00.xx.xx
L_LdVirtualMasterV	Virtual master for electronic gearbox	LenzeLineDrive V01.00.xx.xx
L_SdFactor	Stretch factor	LenzeServoDrive V01.00.xx.xx

Function block	Function	From library
L_SdRuntimeComp	Compensation of the runtime by a leading master value	LenzeServoDrive V01.00.xx.xx
L_SdSetAxisData	Represented machine parameters of a master drive	LenzeServoDrive V01.00.xx.xx

"Synchronism" application"

Function block	Function	From library
L_LdAddOffsetCyclic	Offset addition to a clocked position	LenzeLineDrive V01.00.xx.xx
L_LdClutchAxisP	Time-controlled virtual clutch with position reference	LenzeLineDrive V01.00.xx.xx
L_LdConvAxisV	Speed ratio between two axes	LenzeLineDrive V01.00.xx.xx
L_LdDifferentiateCyclic	Differentiation of a position considering the cycle	LenzeLineDrive V01.00.xx.xx
L_LdExtrapolate	Extrapolation of position information • E.g. to compensate for longer bus transfer cycles or smooth low-resolution absolute value encoders	LenzeLineDrive V01.00.xx.xx
L_LdIntegrateCyclic	Integration of a speed to a position considering the cycle	LenzeLineDrive V01.00.xx.xx
L_LdLinearCoupling	Master value connection	LenzeLineDrive V01.00.xx.xx
L_LdMarkSync	Mark synchronisation in synchronism and cam profiler applications	LenzeLineDrive V01.00.xx.xx
L_LdMonitFollowError	Following error monitoring with adjustable switching threshold and hysteresis	LenzeLineDrive V01.00.xx.xx
L_LdPosCtrlLin	Master value adjustment (related to position)	LenzeLineDrive V01.00.xx.xx
L_LdStateDecoder	LineDrive status signals • Decoding of the status output signal *dnState* of an FB from the function library "LenzeLineDrive.lib" into individual boolean status signals for further use in the FB interconnection.	LenzeLineDrive V01.00.xx.xx
L_LdSyncOperation	Master value connection	LenzeLineDrive V01.00.xx.xx
L_LdToolControl	Setpoint conditioning for the tool axis	LenzeLineDrive V01.00.xx.xx
L_LdVirtualMasterP	Virtual master for synchronism	LenzeLineDrive V01.00.xx.xx
L_LdZeroDetect	Detection of zero crossings in case of a clocked position	LenzeLineDrive V01.00.xx.xx
L_SdRuntimeComp	Compensation of the runtime by a leading master value	LenzeServoDrive V01.00.xx.xx
L_SdSetAxisData	Represented machine parameters of a master drive	LenzeServoDrive V01.00.xx.xx
L_SdSwitchPoint	Position switch points (cams)	LenzeServoDrive V01.00.xx.xx

4.10 FBs for positioning tasks

Function block	Function	From library
L_PosPositionerInterface	Interface to basic function "Positioning" • Use this interface to control the SB **LS_Positioner** via level-controlled signals, e.g. from a higher-level control.	LenzePositioning V02.00.xx.xx
L_PosPositionerTable	Profile data record management • Storing and managing (traversing) profiles and "teaching" positions, speeds, accelerations/decelerations and S-ramp times.	LenzePositioning V01.00.xx.xx
L_PosProfileTable	Profile data record management (simple) • Storing and managing up to four traversing profiles and "teaching" target positions. • Unlike the FB L_PosPositionerTable, this FB does not use any variable tables, but profile parameter data is directly entered under the assigned codes. • Furthermore, the position assigned to the input *dnExtPos_p* will be used as target position if profile number 1 is selected.	LenzePositioning V01.00.xx.xx
L_PosSequencer	Sequence control • Parameterisable sequence table for implementing a sequence control.	LenzePositioning V01.00.xx.xx
L_PosProfileInterface	Profile data interface • Profile data selection for the SB **LS_Positioner**.	LenzePositioning V01.00.xx.xx
L_SdSwitchPoint	Position switch points (cams)	LenzeServoDrive V01.00.xx.xx
L_LdMonitFollowError	Following error monitoring with adjustable switching threshold and hysteresis.	LenzeLineDrive V01.00.xx.xx
L_PosGetProfile	Profile data tables • Together with an FB instance of type L_PosPositionerTable or L_PosProfileTable, this FB provides five profile data sets that can be selected.	LenzePositioning V01.00.xx.xx
L_PosGetProfileData	Profile data output	LenzePositioning V01.00.xx.xx
L_PosGetTableAcc	Acceleration table • Together with an FB instance • ...of type L_PosPositionerTable this FB provides five acceleration values from the VTACC variable table. • ...of type L_PosProfileTable this FB provides accelerations which are directly defined in the profiles.	LenzePositioning V01.00.xx.xx
L_PosGetTableJerk	S-ramp time table • Together with an FB instance • ...of type L_PosPositionerTable this FB provides five S-ramp times from the VTJERK variable table . • ...of type L_PosProfileTable this FB provides S-ramp times which are directly defined in the profiles.	LenzePositioning V01.00.xx.xx
L_PosGetTablePos	Position table • Together with an FB instance • ...of type L_PosPositionerTable this FB provides five position values from the VTPOS variable table. • ...of type L_PosProfileTable this FB provides positions which are directly defined in the profiles.	LenzePositioning V01.00.xx.xx

Function block	Function	From library
L_PosGetTableSpeed	Speed table • Together with an FB instance • ...of type L_PosPositionerTable this FB provides five speed values from the VTSPEED variable table. • ...of type L_PosProfileTable this FB provides speeds which are directly defined in the profiles.	LenzePositioning V01.00.xx.xx
L_DevPositionerStateDecoder	Status signals of the basic function "Positioning" (SB **LS_Positioner**) • From firmware 1.50, the FB L_DevPositionerStateDecoder replaces the FB L_PosDecoderStatePositioner.	LenzeDevice9400 V02.00.xx.xx
L_PosDecoderStatePositioner		LenzePositioning V01.00.xx.xx
L_PosSequencerStateDecoder	Status signals of the sequence control (FB L_PosSequencer) • From firmware 1.50, the FB L_PosSequencerStateDecoder replaces the FB L_PosDecoderStateSequencer.	LenzePositioning V02.00.xx.xx
L_PosDecoderStateSequencer		LenzePositioning V01.00.xx.xx

4.11 FBs for winding technology

Function block	Function	From library
L_WndCalcDiameter	Diameter calculation	LenzeWinder V01.00.xx.xx
L_WndCalcLength	Length calculation	LenzeWinder V01.00.xx.xx
L_WndFrictionCompensation	Friction compensation	LenzeWinder V01.00.xx.xx
L_WndIdentMInertia	Identification of the moment of inertia	LenzeWinder V01.00.xx.xx
L_WndStopCtrl	Stop controller	LenzeWinder V01.00.xx.xx
L_WndTensionCurve	Tensile force characteristic	LenzeWinder V01.00.xx.xx

4.12 All function blocks [A-Z]

Function block	Function	From library
L_CamClutchPos	Clutch (path-controlled) • Clutch in or declutch a master position via a fifth order polynomial in a path-controlled manner.	LenzeCam V01.00.xx.xx
L_CamContactor	Cam group with switch-on and switch-off dynamisation	LenzeCam V01.00.xx.xx
L_CamCurve	Curve interpolation for characteristics	LenzeCam V01.00.xx.xx
L_CamGetAxisData	Reading out machine parameters from cam data	LenzeCam V01.00.xx.xx
L_CamPosMarker	Reading out position marks from cam data	LenzeCam V01.00.xx.xx
L_CamProfiler	Curve interpolation for motion profiles	LenzeCam V01.00.xx.xx
L_CamSetContDataPDO	Contactor data selection via inputs/PDOs • The inputs can be connected, for instance, to a corresponding port block to accept contactor data from a master control via process data objects (PDOs).	LenzeCam V01.02.xx.xx
L_CamSetContDataSDO	Contactor data selection via codes/SDOs • The codes can be written, for instance, via parameter data objects (SDOs) by a master control.	LenzeCam V01.02.xx.xx
L_CamStretchAbs	Stretch/compress position (absolute) • Simple absolute stretching and compression of a position in the limited traversing range.	LenzeCam V01.00.xx.xx
L_CamStretchFeed	Stretch/compress position (relative) • Relative stretching and compression of a position in the "Modulo" or "Unlimited" traversing range.	LenzeCam V01.00.xx.xx
L_CamSyncIn	Synchronous/oversynchronous clutch-in • Synchronising a tool (drive) to a running material (relative master speed).	LenzeCam V01.01.xx.xx
L_Dc2BytesToWord	Type converter • BYTE (2 x) ➞ WORD	LenzeDataConversion V01.00.xx.xx
L_Dc2WordsToDWord	Type converter • WORD (2 x) ➞ DWORD	LenzeDataConversion V01.00.xx.xx
L_Dc4BytesToDWord	Type converter • BYTE (4 x) ➞ DWORD	LenzeDataConversion V01.00.xx.xx
L_DcBitShift	Bit shift operation	LenzeDataConversion V01.00.xx.xx
L_DcBitShiftByte	Bit shift operation	LenzeDataConversion V02.02.xx.xx
L_DcBitShiftDWord	Bit shift operation	LenzeDataConversion V02.02.xx.xx
L_DcBitShiftInt	Bit shift operation	LenzeDataConversion V02.02.xx.xx
L_DcBitShiftWord	Bit shift operation	LenzeDataConversion V02.02.xx.xx
L_DcBitsToByte	Bit multiplexer • BOOL (8 x) ➞ BYTE	LenzeDataConversion V01.00.xx.xx
L_DcBitsToDWord	Bit multiplexer • BOOL (32 x) ➞ DWORD	LenzeDataConversion V01.00.xx.xx
L_DcBitsToWord	Bit multiplexer • BOOL (16 x) ➞ WORD	LenzeDataConversion V01.00.xx.xx
L_DcByteBitand	Bit-by-bit AND operation of two BYTE values	LenzeDataConversion V02.02.xx.xx
L_DcByteBitor	Bit-by-bit OR operation of two BYTE values	LenzeDataConversion V02.02.xx.xx
L_DcByteBitxor	Bit-by-bit EXCLUSIVE-OR operation of two BYTE values	LenzeDataConversion V02.02.xx.xx

Function block	Function	From library
L_DcByteToBits	Bit demultiplexer • BYTE → BOOL (8 x)	LenzeDataConversion V01.00.xx.xx
L_DcByteToBits	Bit demultiplexer • BYTE → BOOL (8 x)	LenzeDataConversion V01.00.xx.xx
L_DcByteToInt	Type converter • BYTE → INT	LenzeDataConversion V02.02.xx.xx
L_DcByteToWord	Type converter • BYTE → WORD	LenzeDataConversion V01.00.xx.xx
L_DcDIntToDWord	Type converter • DINT → DWORD	LenzeDataConversion V01.00.xx.xx
L_DcDIntToInt	Type converter • DINT → INT	LenzeDataConversion V01.00.xx.xx
L_DcDWordBitand	Bit-by-bit AND operation of two DWORD values	LenzeDataConversion V01.00.xx.xx
L_DcDWordBitor	Bit-by-bit OR operation of two DWORD values	LenzeDataConversion V01.00.xx.xx
L_DcDWordBitxor	Bit-by-bit EXCLUSIVE-OR operation of two DWORD values	LenzeDataConversion V01.00.xx.xx
L_DcDWordTo2Words	Type converter • DWORD → WORD (2 x)	LenzeDataConversion V01.00.xx.xx
L_DcDWordTo4Bytes	Type converter • DWORD → BYTE (4 x)	LenzeDataConversion V01.00.xx.xx
L_DcDWordToBits	Bit demultiplexer • DWORD → BOOL (32 x)	LenzeDataConversion V01.00.xx.xx
L_DcDWordToBits	Bit demultiplexer • DWORD → BOOL (32 x)	LenzeDataConversion V01.00.xx.xx
L_DcDWordToDInt	Type converter • DWORD → DINT	LenzeDataConversion V01.00.xx.xx
L_DcDWordToWord	Type converter • DWORD → WORD	LenzeDataConversion V01.00.xx.xx
L_DcGetBitOfByte	Bit operation • Output a status of a single bit.	LenzeDataConversion V01.00.xx.xx
L_DcGetBitOfDWord	Bit operation • Output a status of a single bit.	LenzeDataConversion V01.00.xx.xx
L_DcGetBitOfWord	Bit operation • Output a status of a single bit.	LenzeDataConversion V01.00.xx.xx
L_DcIntToDInt	Type converter • INT → DINT	LenzeDataConversion V01.00.xx.xx
L_DcIntToWord	Type converter • INT → WORD	LenzeDataConversion V01.00.xx.xx
L_DcNorm_aToNorm_n	Signal converter • 16-bit signal (scaled) → 32-bit signal (scaled)	LenzeDataConversion V01.00.xx.xx
L_DcNorm_nToNorm_a	Signal converter • 32-bit signal (scaled) → 16-bit signal (scaled)	LenzeDataConversion V01.00.xx.xx
L_DcNorm_nToSpeed_s	Signal converter • 32-bit signal (scaled) → 32-bit-speed signal	LenzeDataConversion V01.00.xx.xx
L_DcResetBitOfByte	Bit operation • Reset a single bit to "0".	LenzeDataConversion V01.00.xx.xx
L_DcResetBitOfDWord	Bit operation • Reset a single bit to "0".	LenzeDataConversion V01.00.xx.xx
L_DcResetBitOfWord	Bit operation • Reset a single bit to "0".	LenzeDataConversion V01.00.xx.xx

Function block	Function	From library
L_DcSetBitOfByte	Bit operation • Set a single bit to "1".	LenzeDataConversion V01.00.xx.xx
L_DcSetBitOfDWord	Bit operation • Set a single bit to "1".	LenzeDataConversion V01.00.xx.xx
L_DcSetBitOfWord	Bit operation • Set a single bit to "1".	LenzeDataConversion V01.00.xx.xx
L_DcSpeed_sToNorm_n	Signal converter • 32-bit speed signal → 32-bit signal (scaled)	LenzeDataConversion V01.00.xx.xx
L_DcSpeed_sToSpeed_v	Signal converter • 32-bit speed signal → 16-bit speed signal	LenzeDataConversion V01.00.xx.xx
L_DcSpeed_vToSpeed_s	Signal converter • 16-bit speed signal → 32-bit speed signal	LenzeDataConversion V01.00.xx.xx
L_DcWordBitand	Bit-by-bit AND operation of two WORD values	LenzeDataConversion V02.02.xx.xx
L_DcWordBitor	Bit-by-bit OR operation of two WORD values	LenzeDataConversion V02.02.xx.xx
L_DcWordBitxor	Bit-by-bit EXCLUSIVE-OR operation of two WORD values	LenzeDataConversion V02.02.xx.xx
L_DcWordTo2Bytes	Type converter • WORD → BYTE (2 x)	LenzeDataConversion V01.00.xx.xx
L_DcWordToBits	Bit demultiplexer • WORD → BOOL (16 x)	LenzeDataConversion V01.00.xx.xx
L_DcWordToBits	Bit demultiplexer • WORD → BOOL (16 x)	LenzeDataConversion V01.00.xx.xx
L_DcWordToByte	Type converter • WORD → BYTE	LenzeDataConversion V01.00.xx.xx
L_DcWordToDWord	Type converter • WORD → DWORD	LenzeDataConversion V01.00.xx.xx
L_DcWordToInt	Type converter • WORD → INT	LenzeDataConversion V01.00.xx.xx
L_DevApplErr	Error tripping • The FB can trip up to 8 different application messages out of the application. • Error ID and error response as well as the module ID can be set via parameters.	LenzeDevice9400 V01.00.xx.xx
L_DevApplErrFix	Error tripping • The FB can trip up to 8 different application messages out of the application. • Error ID and error response as well as the module ID can be set via FB inputs.	LenzeDevice9400 V02.00.xx.xx
L_DevBrakeStateDecoder	Status signals of the basic function "Brake control" (SB **LS_Brake**)	LenzeDevice9400 V01.00.xx.xx
L_DevDriveInterfaceStateDecoder	Status signals of the drive interface (SB **LS_DriveInterface**)	LenzeDevice9400 V01.00.xx.xx
L_DevHomingStateDecoder	Status signals of the basic function "Homing" (SB **LS_Homing**)	LenzeDevice9400 V01.00.xx.xx
L_DevLimiterStateDecoder	Status signals of the basic function "Limiter" (SB **LS_Limiter**)	LenzeDevice9400 V01.00.xx.xx
L_DevManualJogStateDecoder	Status signals of the basic function "Manual jog" (SB **LS_ManualJog**)	LenzeDevice9400 V01.00.xx.xx

Function block	Function	From library
L_DevParReadFix	One-time or cyclic reading of parameters • Only internal parameters (codes) of the controller can be read. • In contrast to the L_DevReadParDInt FB, the parameter of this FB to be read is addressed via FB inputs and not via parameters.	LenzeDevice9400 V02.00.xx.xx
L_DevParWriteFix	One-time or cyclic writing of parameters • Only internal parameters (codes) of the controller can be written. • In contrast to the L_DevWriteParDInt FB, the parameter of this FB to be written is addressed via FB inputs and not via parameters.	LenzeDevice9400 V02.00.xx.xx
L_DevPositionerStateDecoder	Status signals of the basic function "Positioning" (SB **LS_Positioner**)	LenzeDevice9400 V02.00.xx.xx
L_DevReadParDInt	One-time or cyclic reading of parameters of "DINT" data type • Only internal parameters (codes) of the controller can be read.	LenzeDevice9400 V01.00.xx.xx
L_DevSMControlDecoder	Control signals from the SM3xx safety module	LenzeDevice9400 V02.00.xx.xx
L_DevSMControlEncoder	Control word for limiter • The FB generates the bit-coded control signal *dwControl* from individual boolean control signals for the SB **LS_Limiter** and thus allows control of the basic function "Limiter" if no SM3xx safety module is available.	LenzeDevice9400 V02.00.xx.xx
L_DevSMStateDecoder	Status signals from the SM3xx safety module	LenzeDevice9400 V02.00.xx.xx
L_DevSMStateDecoderIO	I/O status signals from the SM3xx safety module	LenzeDevice9400 V02.00.xx.xx
L_DevWriteParDInt	One-time or cyclic writing of parameters of the "DINT" data type • Only internal parameters (codes) of the controller can be written.	LenzeDevice9400 V02.00.xx.xx
L_EsClutchPos	Clutch (time-controlled) • Clutch in or declutch a master position via a fifth order polynomial in a time-controlled manner.	LenzeElectricalShaft V01.01.xx.xx
L_EsEncoderConv	Encoder signal conditioning	LenzeElectricalShaft V01.00.xx.xx
L_EsStretchIntegrate	Stretching and compression of a master value in the "Modulo" or "Unlimited" traversing range	LenzeElectricalShaft V01.00.xx.xx
L_LdAddOffsetCyclic	Offset addition to a clocked position	LenzeLineDrive V01.00.xx.xx
L_LdClutchAxisP	Time-controlled virtual clutch with position reference	LenzeLineDrive V01.00.xx.xx
L_LdClutchV	Time-controlled virtual clutch with speed reference	LenzeLineDrive V01.00.xx.xx
L_LdConvAxisV	Speed ratio between two axes	LenzeLineDrive V01.00.xx.xx
L_LdDifferentiateCyclic	Differentiation of a position considering the cycle	LenzeLineDrive V01.00.xx.xx
L_LdExtrapolate	Extrapolation of position information • E.g. to compensate for longer bus transfer cycles or smooth low-resolution absolute value encoders.	LenzeLineDrive V01.00.xx.xx
L_LdIntegrateCyclic	Integration of a speed to a position considering the cycle	LenzeLineDrive V01.00.xx.xx
L_LdLinearCoupling	Master value connection	LenzeLineDrive V01.00.xx.xx

Function block	Function	From library
L_LdMarkSync	Mark synchronisation in synchronism and cam profiler applications	LenzeLineDrive V01.00.xx.xx
L_LdMonitFollowError	Following error monitoring with adjustable switching threshold and hysteresis	LenzeLineDrive V01.00.xx.xx
L_LdMPot	Master value adjustment in positive/negative traversing direction with parameterisable speeds	LenzeLineDrive V01.00.xx.xx
L_LdPosCtrlLin	Master value adjustment (related to position)	LenzeLineDrive V01.00.xx.xx
L_LdSetAxisVelocity	Stretching/compression of the x axis and synchronisation via touch probe • Within the cycle, the master speed is integrated to the position.	LenzeLineDrive V01.00.xx.xx
L_LdStateDecoder	LineDrive status signals • Decoding of the status output signal *dnState* of an FB from the function library "LenzeLineDrive.lib" into individual boolean status signals for further use in the FB interconnection.	LenzeLineDrive V01.00.xx.xx
L_LdSyncOperation	Master value connection	LenzeLineDrive V01.00.xx.xx
L_LdToolControl	Setpoint conditioning for the tool axis.	LenzeLineDrive V01.00.xx.xx
L_LdVirtualMasterP	Virtual master for synchronism	LenzeLineDrive V01.00.xx.xx
L_LdVirtualMasterV	Virtual master for electronic gearbox	LenzeLineDrive V01.00.xx.xx
L_LdZeroDetect	Detection of zero crossings in case of a clocked position	LenzeLineDrive V01.00.xx.xx
L_PosDecoderStatePositioner	Status signals of the basic function "Positioning" (SB **LS_Positioner**) • From firmware 1.50, the FB is replaced by the FB L_DevPositionerStateDecoder.	LenzePositioning V01.00.xx.xx
L_PosDecoderStateSequencer	Status signals of the sequence control (FB L_PosSequencer) • From firmware 1.50, the FB is replaced by the FB L_PosSequencerStateDecoder.	LenzePositioning V01.00.xx.xx
L_PosGetProfile	Profile data tables • Together with an FB instance of type L_PosPositionerTable or L_PosProfileTable, this FB provides five profile data sets that can be selected.	LenzePositioning V01.00.xx.xx
L_PosGetProfileData	Profile data output	LenzePositioning V01.00.xx.xx
L_PosGetTableAcc	Acceleration table • Together with an FB instance • ...of type L_PosPositionerTable this FB provides five acceleration values from the VTACC variable table. • ...of type L_PosProfileTable this FB provides accelerations which are directly defined in the profiles.	LenzePositioning V01.00.xx.xx
L_PosGetTableJerk	S-ramp time table • Together with an FB instance • ...of type L_PosPositionerTable this FB provides five S-ramp times from the VTJERK variable table . • ...of type L_PosProfileTable this FB provides S-ramp times which are directly defined in the profiles.	LenzePositioning V01.00.xx.xx

Function block	Function	From library
L_PosGetTablePos	Position table • Together with an FB instance • ...of type L_PosPositionerTable this FB provides five position values from the VTPOS variable table. • ...of type L_PosProfileTable this FB provides positions which are directly defined in the profiles.	LenzePositioning V01.00.xx.xx
L_PosGetTableSpeed	Speed table • Together with an FB instance • ...of type L_PosPositionerTable this FB provides five speed values from the VTSPEED variable table. • ...of type L_PosProfileTable this FB provides speeds which are directly defined in the profiles.	LenzePositioning V01.00.xx.xx
L_PosPositionerInterface	Interface to basic function "Positioning" • Use this interface to control the SB **LS_Positioner** via level-controlled signals, e.g. from a higher-level control.	LenzePositioning V02.00.xx.xx
L_PosPositionerTable	Profile data record management • Storing and managing (traversing) profiles and "teaching" positions, speeds, accelerations/decelerations and S-ramp times.	LenzePositioning V01.00.xx.xx
L_PosProfileInterface	Profile data interface • Profile data selection for the SB **LS_Positioner**.	LenzePositioning V01.00.xx.xx
L_PosProfileTable	Profile data record management (simple) • Storing and managing up to four traversing profiles and "teaching" target positions. • Unlike the FB L_PosPositionerTable, this FB does not use any variable tables, but profile parameter data is directly entered under the assigned codes. • Furthermore, the position assigned to the input *dnExtPos_p* will be used as target position if profile number 1 is selected.	LenzePositioning V01.00.xx.xx
L_PosSequencer	Sequence control • Parameterisable sequence table for implementing a sequence control.	LenzePositioning V01.00.xx.xx
L_PosSequencerStateDecoder	Status signals of the sequence control (FB L_PosSequencer)	LenzePositioning V02.00.xx.xx
L_SdAccToUnit	Acceleration conversion • Speed variation/time ➞ acceleration [unit/s^2]	LenzeServoDrive V01.00.xx.xx
L_SdDelayComp	Extrapolation • The FB extrapolates a signal by the time set.	LenzeServoDrive V01.00.xx.xx
L_SdDifferentiate	Differentiation (position ➞ speed)	LenzeServoDrive V01.00.xx.xx
L_SdFactor	Stretch factor	LenzeServoDrive V01.00.xx.xx
L_SdGetAxisData	Reading out machine parameters from axis data	LenzeServoDrive V02.07.xx.xx
L_SdGetPosition	Position conversion • Position [increments] ➞ position [unit] (displayed as parameter value)	LenzeServoDrive V01.00.xx.xx
L_SdGetSpeed	Speed conversion • Speed ➞ velocity [unit/s] (displayed as parameter value)	LenzeServoDrive V01.00.xx.xx
L_SdIntegrate	Integration (speed ➞ position)	LenzeServoDrive V01.00.xx.xx

Function block	Function	From library
L_SdIntegrateAxis	Integration (speed → position) with TP correction	LenzeServoDrive V02.03.xx.xx
L_SdIntegrateLimit	Integration (speed → position) with adjustable upper/lower limit for return purposes	LenzeServoDrive V02.09.xx.xx
L_SdInterExtrapolateAny	Interpolation/extrapolation of setpoints	LenzeServoDrive V02.09.xx.xx
L_SdInterExtrapolatePosition	Interpolation/extrapolation of position information	LenzeServoDrive V02.09.xx.xx
L_SdInterpolate	Interpolation of position information • E.g. to compensate for longer bus transfer cycles or smooth low-resolution absolute value encoders.	LenzeServoDrive V02.00.xx.xx
L_SdLimitSpeed	Speed limitation • Leads a speed setpoint to defined limit ranges.	LenzeServoDrive V01.00.xx.xx
L_SdMotorPot	Motor potentiometer function • Can be used as an alternative setpoint source that is controlled via two inputs.	LenzeServoDrive V02.04.xx.xx
L_SdPosToUnit	Position conversion • Position [increments] → position [unit]	LenzeServoDrive V01.00.xx.xx
L_SdProcessController	Comprehensive PID controller with setpoint and actual value processing as well as parameterisable characteristic function and limitation	LenzeServoDrive V02.05.xx.xx
L_SdRampGenerator	Ramp generator (ramp function generator) with S-shaped ramps	LenzeServoDrive V01.00.xx.xx
L_SdRampGeneratorAny	Ramp generator (ramp function generator) with S-shaped ramps. • Unlike the L_SdRampGenerator FB, this FB is also able to process speeds (_s). Separate acceleration and deceleration times can be set.	LenzeServoDrive V02.09.xx.xx
L_SdRuntimeComp	Compensation of the runtime by a leading master value	LenzeServoDrive V01.00.xx.xx
L_SdSetAxisData	Represented machine parameters of a master drive	LenzeServoDrive V01.00.xx.xx
L_SdSetPosition	Position conversion • Position [unit] (selected as parameter value) → position [increments]	LenzeServoDrive V01.00.xx.xx
L_SdSetSpeed	Speed conversion • Velocity [unit/s] (selected as parameter value) → speed	LenzeServoDrive V01.00.xx.xx
L_SdSpeedFilter	Speed signal delay • Filters or decelerates speed signal according to the PT1 principle.	LenzeServoDrive V01.00.xx.xx
L_SdSpeedSet	Ramp generator (ramp function generator) with S-shaped ramps, parameterisable fixed setpoints and comprehensive parameter setting and control options	LenzeServoDrive V01.00.xx.xx
L_SdSpeedToUnit	Speed conversion • Speed → velocity [unit/s]	LenzeServoDrive V01.00.xx.xx
L_SdSwitchPoint	Position switch points (cams)	LenzeServoDrive V01.00.xx.xx
L_SdTouchProbe	Touch probe processing • The FB takes over the interpolation of the input signal based on the time stamp handed over by a touch probe system block and outputs the interpolated value and the difference compared with the last input signal.	LenzeServoDrive V01.00.xx.xx
L_SdUnitToAcc	Acceleration conversion • Acceleration [unit/s^2] → speed variation/time	LenzeServoDrive V01.00.xx.xx

Function block	Function	From library
L_SdUnitToPos	Position conversion • Position [unit] ➞ position [increments]	LenzeServoDrive V01.00.xx.xx
L_SdUnitToSpeed	Speed conversion • Velocity [unit/s] ➞ speed	LenzeServoDrive V01.00.xx.xx
L_Tb5And	AND (5 inputs)	LenzeToolbox V01.00.xx.xx
L_Tb5Nand	NOT-AND (5 inputs)	LenzeToolbox V01.00.xx.xx
L_Tb5Nor	NOT-OR (5 inputs)	LenzeToolbox V01.00.xx.xx
L_Tb5Or	OR (5 inputs)	LenzeToolbox V01.00.xx.xx
L_Tb5Xor	EXCLUSIVE-OR (5 inputs)	LenzeToolbox V01.00.xx.xx
L_Tb8Select	1-out-of-8 selector (for data type "DINT") • The FB connects one of eight input signals through to the output.	LenzeToolbox V01.00.xx.xx
L_Tb8SelectByte	1-out-of-8 selector (for data type "BYTE") • The FB connects one of eight input signals through to the output.	LenzeToolbox V02.05.xx.xx
L_Tb8SelectWord	1-out-of-8 selector (for data type "WORD") • The FB connects one of eight input signals through to the output.	LenzeToolbox V02.05.xx.xx
L_TbAbs	Absolute value generation • The function block converts a bipolar signal into a unipolar signal, i.e. the absolute value of the input signal is generated.	LenzeToolbox V01.00.xx.xx
L_TbAdd	Addition without limitation	LenzeToolbox V01.00.xx.xx
L_TbAddLim	Addition with limitation	LenzeToolbox V01.00.xx.xx
L_TbAddSubLim	Addition and subtraction with limitation	LenzeToolbox V02.04.xx.xx
L_TbAnd	AND (2 inputs)	LenzeToolbox V01.00.xx.xx
L_TbCompare	Comparison • The function block compares two signals of the "DINT" data type and can be used e.g. for implementing a trigger.	LenzeToolbox V01.00.xx.xx
L_TbCompare_n	Comparison (scaled) • The function block compares two scaled signals and can be used e.g. for implementing a trigger.	LenzeToolbox V01.00.xx.xx
L_TbCount	Upcounter and downcounter with adjustable limitation	LenzeToolbox V01.00.xx.xx
L_TbCurve	Characteristic function (scaled) • The FB produces the corresponding value of the Y axis for the X axis value with a parameterisable characteristic function y = f(x).	LenzeToolbox V01.00.xx.xx
L_TbCurveAny	Characteristic function • The FB produces the corresponding value of the Y axis for the X axis value with a parameterisable characteristic function y = f(x).	LenzeToolbox V02.06.xx.xx
L_TbDeadband	Dead band • The FB produces a symmetrical dead band around zero.	LenzeToolbox V01.00.xx.xx
L_TbDeadband_n	Dead band (scaled) • The FB produces a symmetrical dead band (scaled) around zero.	LenzeToolbox V01.00.xx.xx
L_TbDelay	Binary delay element	LenzeToolbox V01.00.xx.xx
L_TbDifferentiate	Differentiator with low pass	LenzeToolbox V01.00.xx.xx
L_TbDiv	Division with remainder	LenzeToolbox V01.00.xx.xx

Function block	Function	From library
L_TbDiv_n	Division divisor selected as scaled signal [%]	LenzeToolbox V01.00.xx.xx
L_TbFlipFlopD	D flipflop • Can be used for evaluation and storage of digital signal edges.	LenzeToolbox V01.00.xx.xx
L_TbFlipFlopRS	RS flipflop • Can be used for evaluation and storage of digital signals.	LenzeToolbox V01.00.xx.xx
L_TbGainLim	Gain with limitation	LenzeToolbox V01.00.xx.xx
L_TbIntegrate	Integration with limitation	LenzeToolbox V01.00.xx.xx
L_TbLimit	Signal limitation (for data type "DINT") • The FB limits a signal of "DINT" data type to an adjustable value range.	LenzeToolbox V01.00.xx.xx
L_TbLimit_n	Signal limitation (scaled) • The FB limits a scaled signal [%] to an adjustable value range.	LenzeToolbox V01.00.xx.xx
L_TbLimitInt	Signal limitation (for data type "INT") • The FB limits a signal of "INT" data type to an adjustable value range.	LenzeToolbox V02.05.xx.xx
L_TbMaskOut	Zone masking • The FB masks up to four parameterisable blocking zones within a continuous signal characteristic.	LenzeToolbox V01.00.xx.xx
L_TbMul	Multiplication without limitation	LenzeToolbox V01.00.xx.xx
L_TbMul_n	Multiplication with the multiplicand selected as scaled signal [%]	LenzeToolbox V01.00.xx.xx
L_TbMulDivLim	Multiplication and division with limitation	LenzeToolbox V02.02.xx.xx
L_TbMulLim	Multiplication with limitation	LenzeToolbox V01.00.xx.xx
L_TbNand	NOT-AND (2 inputs)	LenzeToolbox V01.00.xx.xx
L_TbNeg	Negation (for data type "DINT") • The function block inverts the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output.	LenzeToolbox V01.00.xx.xx
L_TbNegInt	Negation (for data type "INT") • The function block inverts the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output.	LenzeToolbox V02.05.xx.xx
L_TbNegSel	Optional negation (for data type "DINT") • The function block inverts (optionally) the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output. • In contrast to the L_TbNeg FB, this FB is only inverted if the *bNeg* control input is set to TRUE.	LenzeToolbox V02.02.xx.xx
L_TbNegSelInt	Optional negation (for data type "INT") • The function block inverts (optionally) the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output. • In contrast to the L_TbNegInt FB, this FB is only inverted if the *bNeg* control input is set to TRUE.	LenzeToolbox V02.05.xx.xx
L_TbNor	NOT-OR (2 inputs)	LenzeToolbox V01.00.xx.xx
L_TbNormalize	Signal scaling • The FB scales any signal to a parameterisable reference variable.	LenzeToolbox V01.00.xx.xx
L_TbNot	Negation (inverter)	LenzeToolbox V01.00.xx.xx
L_TbOr	OR (2 inputs)	LenzeToolbox V01.00.xx.xx
L_TbOscillator	Rectangular pulse generator with parameterisable pulse/dead time	LenzeToolbox V01.00.xx.xx

Function block	Function	From library
L_TbPIController	Simple PI controller with different control functions and parameterisable limitation	LenzeToolbox V01.00.xx.xx
L_TbPT1Filter	PT1 filter • The FB filters and delays analog signals.	LenzeToolbox V01.00.xx.xx
L_TbRateAction	Rate action function to compensate disturbing low passes	LenzeToolbox V01.00.xx.xx
L_TbSampleHold	Sample & Hold (for data type "DINT") • The FB stores a value of the "DINT" data type in order to keep it even after mains switching.	LenzeToolbox V01.00.xx.xx
L_TbSampleHoldWord	Sample & Hold (for data type "WORD") • The FB stores a value of the "WORD" data type in order to keep it even after mains switching.	LenzeToolbox V02.05.xx.xx
L_TbSelect	Selector (for data type "DINT") • The function block switches between two signals of "DINT" data type. The switch-over is controlled via a boolean input signal.	LenzeToolbox V01.00.xx.xx
L_TbSelectByte	Selector (for data type "BYTE") • The function block switches between two signals of "BYTE" data type. The switch-over is controlled via a boolean input signal.	LenzeToolbox V02.02.xx.xx
L_TbSelectWord	Selector (for data type "WORD") • The function block switches between two signals of "WORD" data type. The switch-over is controlled via a boolean input signal.	LenzeToolbox V02.05.xx.xx
L_TbSub	Subtraction without limitation	LenzeToolbox V01.00.xx.xx
L_TbSubLim	Subtraction with limitation	LenzeToolbox V01.00.xx.xx
L_TbTransition	Signal edge evaluation • Can be used to generate timed and retriggerable pulses.	LenzeToolbox V01.00.xx.xx
L_TbXor	EXCLUSIVE-OR (2 inputs)	LenzeToolbox V01.00.xx.xx
L_WndCalcDiameter	Diameter calculation	LenzeWinder V01.00.xx.xx
L_WndCalcLength	Length calculation	LenzeWinder V01.00.xx.xx
L_WndFrictionCompensation	Friction compensation	LenzeWinder V01.00.xx.xx
L_WndIdentMInertia	Identification of the moment of inertia	LenzeWinder V01.00.xx.xx
L_WndStopCtrl	Stop controller	LenzeWinder V01.00.xx.xx
L_WndTensionCurve	Tensile force characteristic	LenzeWinder V01.00.xx.xx

4.13 FB reference list - 9300 Servo PLC

The following reference list shall assist you in finding the correct function block to simulate a function known from 9300 servo PLC.

Function block for 9300 servo PLC			Function block for 9400 Servo Drives	
Name	**Description**	**Function library**	**Name**	**Description**
L_32BitTransferDINT	Data-consistent copying of a DINT variable	Lenze32BitTransferDrv	-	-
L_32BitTransferDWORD	Data-consistent copying of a DWORD variable	Lenze32BitTransferDrv	-	-
L_8200CtrlWord	Control word for the frequency inverter	Lenze8200Drive	-	-
L_8200DataControl	Process data exchange with frequency inverter	Lenze8200Drive	-	-
L_8200Initialization	Initialise frequency inverter	Lenze8200Drive	-	-
L_8200Parameter	Initialise codes	Lenze8200Drive	-	-
L_8200StatusWord	Status word of frequency inverters	Lenze8200Drive	-	-
L_ABS	Absolute value generation	LenzeDrive.lib	L_TbAbs	Absolute value generation
L_ADD	Addition	LenzeDrive.lib	L_TbAdd	Addition without limitation
			L_TbAddLim	Addition with limitation
L_AifParMapClose*	Deactivate code mapping	LenzeAifParMapDrv	-	-
L_AifParMapInit*	Configure code mapping	LenzeAifParMapDrv	-	-
L_AifParMapOpen*	Activate code mapping	LenzeAifParMapDrv	-	-
L_AIN	Input gain & offset	LenzeDrive.lib	**LS_AnalogInput** SB	Analog inputs
L_AND	AND	LenzeDrive.lib	L_TbAnd	AND (2 inputs)
			L_Tb5And	AND (5 inputs)
L_ANEG	Inversion	LenzeDrive.lib	L_TbNeg	Negation
			L_TbNegSel	Negation (optional)
L_AOUT	Output gain & offset	LenzeDrive.lib	L_TbGainLim	Gain with limitation
			L_TbAdd	Addition without limitation
			L_TbAddLim	Addition with limitation
			L_TbAddSubLim	Addition and subtraction with limitation
L_ARIT	Arithmetic	LenzeDrive.lib	L_TbAdd	Addition without limitation
			L_TbAddLim	Addition with limitation
			L_TbAddSubLim	Addition and subtraction with limitation
			L_TbDiv	Division with remainder
			L_TbDiv_n	Division divisor selected as scaled signal [%]
			L_TbMul	Multiplication without limitation
			L_TbMul_n	Multiplication with the multiplicand selected as scaled signal [%]
			L_TbMulLim	Multiplication with limitation
			L_TbSub	Subtraction without limitation
			L_TbSubLim	Subtraction with limitation

* Function (no function block)

Function block for 9300 servo PLC			Function block for 9400 Servo Drives	
Name	**Description**	**Function library**	**Name**	**Description**
L_ARITPH	Arithmetic	LenzeDrive.lib	L_TbAdd	Addition without limitation
			L_TbAddLim	Addition with limitation
			L_TbAddSubLim	Addition and subtraction with limitation
			L_TbDiv	Division with remainder
			L_TbDiv_n	Division divisor selected as scaled signal [%]
			L_TbMul	Multiplication without limitation
			L_TbMul_n	Multiplication with the multiplicand selected as scaled signal [%]
			L_TbMulLim	Multiplication with limitation
			L_TbSub	Subtraction without limitation
			L_TbSubLim	Subtraction with limitation
L_ASW	Switch-over	LenzeDrive.lib	L_Tb8Select	1-from-8-selector
			L_TbSelect	Selector
L_BRK	Holding brake	Lenze9300Servo.lib	**LS_Brake** SB	Basic function "Brake control"
L_ByteArrayToDint	Type conversion	LenzeDrive.lib	-	-
L_CamAddVelocity	Speed addition	LenzeCamControl2	-	-
L_CamAdjustDfIn	Digital frequency adjustment	LenzeCamControl2	-	-
L_CamClutchXAxis	Virtual clutch	LenzeCamControl1	L_CamClutchPos	Virtual clutch (path-controlled)
			L_LdClutchAxisP	Time-controlled virtual clutch with position reference
			L_LdClutchV	Time-controlled virtual clutch with speed reference
L_CamContactor	Cam group	LenzeCamControl1	L_CamContactor	Cam group with switch-on and switch-off dynamisation
			L_SdSwitchPoint	Position switch points (cams)
L_CamContactorVPos	Dynamic adjustment of the cams/hysteresis	LenzeCamControl1	L_CamContactor	Cam group with switch-on and switch-off dynamisation
L_CamControl	Setpoint conditioning	LenzeCamControl1	**LS_CamInterface** SB	Basic drive function "Cam data management"
L_CamData	Profile management and curve profile generator	LenzeCamControl1	**LS_CamInterface** SB	Basic drive function "Cam data management"
			L_CamProfiler	Curve interpolation
			L_CamCurve	Characteristic interpolation
L_CamDifferentiate	Converter/differentiator	LenzeCamControl2	L_SdDifferentiate	Differentiation (position → speed)
L_CamEditProfileData	Online change of the profile points	LenzeCamControl2	**LS_CamInterface** SB	Basic drive function "Cam data management"
L_CamExtrapolate	Extrapolation	LenzeCamControl2	L_LdExtrapolate	Extrapolation of position information
			L_SdInterExtrapolateAny	Interpolation/extrapolation of setpoints
			L_SdInterExtrapolatePosition	Interpolation/extrapolation of position information

* Function (no function block)

Function block for 9300 servo PLC			Function block for 9400 Servo Drives	
Name	**Description**	**Function library**	**Name**	**Description**
L_CamGetPosition	Unit conversion	LenzeCamControl1	L_SdPosToUnit	Position [increments] → position [unit]
			L_SdGetPosition	Position [increments] → position [unit] (displayed as parameter value)
L_CamIntegrateVarLim	Phase integrator	LenzeCamControl2	L_SdIntegrate	Integration (speed → position)
L_CamLimitVelocity	Limiting element	LenzeCamControl1	L_SdLimitSpeed	Leads a speed setpoint to defined limit ranges
L_CamMachineData	Machine parameters	LenzeCamControl1	L_CamGetAxisData	Reading out machine parameters from cam data
			L_SdSetAxisData	Represented machine parameters of a master drive
L_CamMonitFollowError	Following error monitoring	LenzeCamControl1	L_LdMonitFollowError	Following error monitoring with adjustable switching threshold and hysteresis
L_CamPosCtrlLin	Linear profile generator	LenzeCamControl1	-	-
L_CamProfileData	Loading of important profile points	LenzeCamControl1	-	-
L_CamSectionShift	Section shift	LenzeCamControl2	-	-
L_CamSelPosition	Master value selection	LenzeCamControl2	L_Tb8Select	1-from-8-selector
L_CamSelVelocity	Digital frequency selection	LenzeCamControl2	L_Tb8Select	1-from-8-selector
L_CamSequence	Selection of a profile sequence	LenzeCamControl1	-	-
L_CamSetPosition	Unit conversion	LenzeCamControl1	L_SdUnitToPos	Position [unit] → position [increments]
			L_SdSetPosition	Position [unit] (selected as parameter value) → position [increments]
L_CamSetXAxisVelocity	Stretching & compressing the master value	LenzeCamControl1	L_CamStretchAbs	Stretch/compress position (absolute)
			L_CamStretchFeed	Stretch/compress position (relative)
			L_LdSetAxisVelocity	Stretching/compression of the X axis and synchronisation via touch probe. The master speed is integrated to the position within the cycle.
L_CamSetYAxisVelocity	Stretching & compressing the setpoint	LenzeCamControl1	L_CamStretchAbs	Stretch/compress position (absolute)
			L_CamStretchFeed	Stretch/compress position (relative)
L_CamVelocityFilter	Filter (PT1) for speed signals	LenzeCamControl2	L_TbPT1Filter	PT1 filter
L_CamVMasterPosition	Virtual master with positioning function	LenzeCamControl1	L_LdVirtualMasterP	Virtual master with position reference
L_CamVMasterVelocity	Virtual master	LenzeCamControl1	L_LdVirtualMasterV	Virtual master with speed reference
L_CanClose*	Deactivation	LenzeCanDrv	-	-
L_CanDSxClose*	Deactivate index mapping	LenzeCanDSxDrv	-	-
L_CanDSxCloseHeartBeat*	Deactivate "Heartbeat"	LenzeCanDSxDrv	-	-
L_CanDSx CloseNodeGuarding*	Deactivate "Node Guarding"	LenzeCanDSxDrv	-	-
L_CanDSxHeartBeat	Execute "Heartbeat"	LenzeCanDSxDrv	-	-
L_CanDSxInitIndexCode*	Configure index mapping	LenzeCanDSxDrv	-	-
L_CanDSxNodeGuarding	Execute "Node Guarding"	LenzeCanDSxDrv	-	-
L_CanDSxOpen*	Initialise CanDSx driver	LenzeCanDSxDrv	-	-
L_CanDSxOpenHeartBeat*	Initialise "Heartbeat"	LenzeCanDSxDrv	-	-
L_CanDSx OpenNodeGuarding*	Initialise "Node Guarding"	LenzeCanDSxDrv	-	-
L_CanGetRelocCobId*	Query - COB-ID area	LenzeCanDrv	-	-
* Function (no function block)				

Function block for 9300 servo PLC			Function block for 9400 Servo Drives	
Name	**Description**	**Function library**	**Name**	**Description**
L_CanGetState*	Driver status	LenzeCanDrv	-	-
L_CanInit*	Initialisation	LenzeCanDrv	-	-
L_CanPdoReceive	Receive CAN object	LenzeCanDrv	-	-
L_CanPdoTransmit	Transmit CAN object	LenzeCanDrv	-	-
L_CMP	Comparison	LenzeDrive.lib	L_TbCompare	Comparison
			L_TbCompare_n	Comparison (scaled)
L_CONV	Scaling	LenzeDrive.lib	L_TbGainLim	Gain with limitation
			L_TbNormalize	Signal scaling
L_CONVPA	Conversion of angle to analog	LenzeDrive.lib	-	-
L_CONVPP	Conversion of phase signal	LenzeDrive.lib	-	-
L_CONVVV	Conversion	LenzeDrive.lib	-	-
L_CONVX	Scaling with limitation	LenzeDrive.lib	L_TbNormalize	Signal scaling
L_CURVE	Characteristic function	LenzeDrive.lib	L_TbCurve	Characteristic function (scaled)
			L_TbCurveAny	Characteristic function
L_DB	Dead band	LenzeDrive.lib	L_TbDeadband	Dead band
			L_TbDeadband_n	Dead band (scaled)
L_DFRFG	Digital frequency ramp function generator	LenzeElectricalShaft	L_CamSyncIn	Synchronous/oversynchronous clutch-in
L_DFSET	Digital frequency processing	LenzeElectricalShaft	-	-
L_DIGDEL	Binary delay element	LenzeDrive.lib	L_TbDelay	Binary delay element
L_DintToByteArray	Type conversion	LenzeDrive.lib	-	-
L_Display9371BB	Transparent mode with operating unit 9371BB	LenzeDrive.lib	-	-
L_DT1_	Differentiation	LenzeDrive.lib	L_TbDifferentiate	Differentiator with low pass
L_FCNT	Up/downcounter	LenzeDrive.lib	L_TbCount	Upcounter and downcounter with adjustable limitation.
L_FIXSET	Programming of fixed setpoints	LenzeDrive.lib	L_SdSpeedSet	Ramp generator (ramp function generator) with S-shaped ramps, parameterisable fixed setpoints and comprehensive parameter setting and control options.
L_FLIP	D flipflop	LenzeDrive.lib	L_TbFlipFlopD	D flipflop
			L_TbFlipFlopRS	RS flipflop
L_FUNCodeIndexConv	Index of a code (function)	LenzeDrive.lib	-	-
L_FWM	Error tripping	LenzeDrive.lib	L_DevApplErr	Error tripping
			L_DevApplErrFix	Error tripping
L_GEARCOMP	Gearbox compensation	LenzeElectricalShaft	-	-
L_IOAInModule	Signal conversion	LenzeIOSystem	-	-
L_IOAOutModule	Signal conversion	LenzeIOSystem	-	-
L_IOCompactModule	Coordinate data	LenzeIOSystem	-	-
L_IOConvByteArrayToByte	Conversion function	LenzeIOSystem	-	-
L_IOConvByteToByteArray	Conversion function	LenzeIOSystem	-	-
L_IOCounterDataFromIO	Counter function	LenzeIOSystem	-	-
L_IOCounterDataToIO	Counter function	LenzeIOSystem	-	-
L_IOCounterDI ModuleDataFromIO	Counter function	LenzeIOSystem	-	-
L_IOCounterDI ModuleDataToIO	Counter function	LenzeIOSystem	-	-
L_IOData15	Coordinate data	LenzeIOSystem	-	-
L_IOData610	Coordinate data	LenzeIOSystem	-	-

* Function (no function block)

Function block for 9300 servo PLC			Function block for 9400 Servo Drives	
Name	**Description**	**Function library**	**Name**	**Description**
L_IODInModule	Bit conversion	LenzeIOSystem	-	-
L_IODOutModule	Bit conversion	LenzeIOSystem	-	-
L_IOParAIAOModule	Parameter function	LenzeIOSystem	-	-
L_IOParAInModule	Parameter function	LenzeIOSystem	-	-
L_IOParAOutModule	Parameter function	LenzeIOSystem	-	-
L_IOParComGuarding	Monitoring function	LenzeIOSystem	-	-
L_IOParCompactModule	Parameter function	LenzeIOSystem	-	-
L_IOParCounterDIModule	Counter function	LenzeIOSystem	-	-
L_IOParCounterModule	Counter function	LenzeIOSystem	-	-
L_IOParPDO15	Parameter function	LenzeIOSystem	-	-
L_IOParPDO610	Parameter function	LenzeIOSystem	-	-
L_IOParSSIModule	Parameter function	LenzeIOSystem	-	-
L_IOSSIDataFromIO	Counter function	LenzeIOSystem	-	-
L_IOSSIDataToIO	Counter function	LenzeIOSystem	-	-
L_LIM	Limit	LenzeDrive.lib	-	-
L_MCAutoResetInterface	Process control	LenzeMotionControl	-	-
L_MCGetPosition	Unit conversion	LenzeMotionControl	L_SdPosToUnit	Position [increments] → position [unit]
			L_SdGetPosition	Position [increments] → position [unit] (displayed as parameter value)
L_MCGetVelocity	Unit conversion	LenzeMotionControl	L_SdGetSpeed	Speed → velocity [unit/s] (displayed as parameter value)
L_MCInterpolator	Signal interpolation	LenzeMotionControl	-	-
L_MCMachineData	Machine parameters	LenzeMotionControl	**LS_DriveInterface** SB	The machine parameters for the drive/motor are set via the drive interface.
			L_SdSetAxisData	Represented machine parameters of a master drive.
L_MCPhiDiff	Integration and subtraction of speed signals	LenzeMotionControl	L_TbDifferentiate	Differentiator with low pass
L_MCPhiIntegrator	Integrator with TP set function	LenzeMotionControl	L_SdIntegrate	Integration (speed → position)
			L_TbIntegrate	Integration with limitation
L_MCPhiTrim	Phase trimming is speed-dependent	LenzeMotionControl	-	-
L_MCPhiTrimTime	Phase trimming is time-dependent	LenzeMotionControl	-	-
L_MCPositionCorrection	Feed correction	LenzeMotionControl	-	-
L_MCProfileGenerator	Profile generator	LenzeMotionControl	-	-
L_MCResidualDistance	Residual path calculation	LenzeMotionControl	-	-
L_MCSetAcceleration	Unit conversion	LenzeMotionControl	-	-
L_MCSetDeceleration	Unit conversion	LenzeMotionControl	-	-
L_MCSetPosition	Unit conversion	LenzeMotionControl	-	-
L_MCSetProfileConverter	Unit conversion	LenzeMotionControl	-	-
L_MCSetProfileValues	Unit conversion	LenzeMotionControl	-	-
L_MCSetVelocity	Unit conversion	LenzeMotionControl	-	-
L_MCSignalDerive	Signal differentiation	LenzeMotionControl	-	-
L_MCSwitchCam	Switching points	LenzeMotionControl	-	-
L_MCSwitchPoint	Switching points	LenzeMotionControl	L_SdSwitchPoint	Position switch points (cams)
L_MCTouchProbeCtrl	Touch probe pre-processing	LenzeMotionControl	L_SdTouchProbe	Touch probe processing
L_MemClearFlash*	Delete flash memory segment	LenzeMemDrv	-	-

* Function (no function block)

Function block for 9300 servo PLC			Function block for 9400 Servo Drives	
Name	**Description**	**Function library**	**Name**	**Description**
L_MemCopyFromRam*	Copy data from RAM into variable memory	LenzeMemDrv	-	-
L_MemCopyToRam*	Copy data from variable memory into RAM	LenzeMemDrv	-	-
L_MemGetStateDriver*	Driver status	LenzeMemDrv	-	-
L_MemGetStateOfMemory	PLC memory status	LenzeMemDrv	-	-
L_MemLoadRamFromFlash*	Copy data from flash memory into RAM	LenzeMemDrv	-	-
L_MemReadDINT*	Read DINT value out of RAM	LenzeMemDrv	-	-
L_MemReadDWORD*	Read DWORD value out of RAM	LenzeMemDrv	-	-
L_MemReadINT*	Read INT value out of RAM	LenzeMemDrv	-	-
L_MemReadWORD*	Read WORD value out of RAM	LenzeMemDrv	-	-
L_MemSaveRamToFlash*	Copy data from RAM into flash memory	LenzeMemDrv	-	-
L_MemWriteDINT*	Write DINT value into RAM	LenzeMemDrv	-	-
L_MemWriteDWORD*	Write DWORD value into RAM	LenzeMemDrv	-	-
L_MemWriteINT	Write INT value into RAM	LenzeMemDrv	-	-
L_MemWriteWORD*	Write WORD value into RAM	LenzeMemDrv	-	-
L_MFAIL	Mains failure control	Lenze9300Servo.lib	-	-
L_MPOT	Motor potentiometer	LenzeDrive.lib	L_SdMotorPot	Motor potentiometer
L_NOT	NOT	LenzeDrive.lib	L_TbNot	Negation (inverter)
L_NSET	Speed pre-processing	LenzeDrive.lib	L_SdSpeedSet	Ramp generator (ramp function generator) with S-shaped ramps, parameterisable fixed setpoints and comprehensive parameter setting and control options.
L_OR	OR	LenzeDrive.lib	L_TbOr	OR (2 inputs)
			L_Tb5Or	OR (5 inputs)
L_ParRead	Reading of codes	LenzeDrive.lib	L_DevParReadFix	One-time or cyclic reading of parameters
			L_DevReadParDInt	One-time or cyclic reading of parameters of data type DINT
L_ParWrite	Writing of codes	LenzeDrive.lib	L_DevParWriteFix	One-time or cyclic writing of parameters
			L_DevWriteParDInt	One-time or cyclic writing of parameters of data type DINT
L_PCTRL	Process controller	LenzeDrive.lib	-	-
L_PHADD	Addition	LenzeDrive.lib	L_TbAdd	Addition without limitation
			L_TbAddLim	Addition with limitation
			L_TbAddSubLim	Addition and subtraction with limitation
L_PHCMP	Comparison	LenzeDrive.lib	L_TbCompare	Comparison
			L_TbCompare_n	Comparison (scaled)
L_PHDIFF	Difference	LenzeDrive.lib	L_TbDifferentiate	Differentiator with low pass
L_PHDIV	Division	LenzeDrive.lib	L_TbDiv	Division with remainder
			L_TbDiv_n	Division divisor selected as scaled signal [%]
L_PHINT	Integration	LenzeDrive.lib	L_SdIntegrate	Integration (speed → position)
			L_SdIntegrateLimit	Integration (speed → position) with adjustable upper/lower limit for return purposes

* Function (no function block)

Function block for 9300 servo PLC			Function block for 9400 Servo Drives	
Name	**Description**	**Function library**	**Name**	**Description**
L_PHINTK	Integration	LenzeDrive.lib	L_SdIntegrate	Integration (speed → position)
			L_SdIntegrateLimit	Integration (speed → position) with adjustable upper/lower limit for return purposes
L_PT1_	Delay	LenzeDrive.lib	L_TbPT1Filter	PT1 filter
L_REF	Homing function	LenzeElectricalShaft	**LS_Homing** SB	Basic function "Homing"
L_RFG	Ramp function generator	LenzeDrive.lib	L_SdRampGenerator	Ramp generator (ramp function generator) with S-shaped ramps.
			L_SdRampGeneratorAny	Unlike the L_SdRampGenerator FB, this FB is also able to process speeds (_s). Separate acceleration and deceleration times can be set.
			L_SdSpeedSet	Ramp generator (ramp function generator) with S-shaped ramps, parameterisable fixed setpoints and comprehensive parameter setting and control options.
L_RLQ	Right/left quick stop	LenzeDrive.lib	-	-
L_Rs232Close	Deactivation (function)	LenzeFpiDrv	-	-
L_Rs232GetReceiveState	Receive status (function)	LenzeFpiDrv	-	-
L_Rs232GetSendState	Transmit status (function)	LenzeFpiDrv	-	-
L_Rs232Open	Initialisation (function)	LenzeFpiDrv	-	-
L_Rs232ReceiveData	Receiving data (function)	LenzeFpiDrv	-	-
L_Rs232SendData	Transmitting data (function)	LenzeFpiDrv	-	-
L_SH	Sample & Hold	LenzeDrive.lib	L_TbSampleHold	Sample & Hold
L_SRFG	Ramp function generator with S ramp	LenzeDrive.lib	L_SdRampGenerator	Ramp generator (ramp function generator) with S-shaped ramps.
			L_SdRampGeneratorAny	Unlike the L_SdRampGenerator FB, this FB is also able to process speeds (_s). Separate acceleration and deceleration times can be set.
			L_SdSpeedSet	Ramp generator (ramp function generator) with S-shaped ramps, parameterisable fixed setpoints and comprehensive parameter setting and control options.
L_TBConvBitsToByte*	BOOL (8 x) → BYTE	LenzePLCToolBox	L_DcBitsToByte	BOOL (8 x) → BYTE
L_TBConvBitsToDword*	BOOL (32 x) → DWORD	LenzePLCToolBox	L_DcBitsToDWord	BOOL (32 x) → DWORD
L_TBConvBitsToWord*	BOOL (16 x) → WORD	LenzePLCToolBox	L_DcBitsToWord	BOOL (16 x) → WORD
L_TBConvByteToBits	BYTE → BOOL (8 x)	LenzePLCToolBox	L_DcByteToBits	BYTE → BOOL (8 x)
L_TBConvCharToByte*	ASCII character → ASCII code	LenzePLCToolBox	-	-
L_TBConvDwordToBits	DWORD → BOOL (32 x)	LenzePLCToolBox	L_DcDWordToBits	DWORD → BOOL (32 x)
L_TBConvWordToBits	WORD → BOOL (16 x)	LenzePLCToolBox	L_DcWordToBits	WORD → BOOL (16 x)
L_TBGetBitOfByte*	Outputs a bit of a BYTE value	LenzePLCToolBox	L_DcGetBitOfByte	Outputs a bit of a BYTE value
L_TBGetBitOfDword*	Outputs a bit of a DWORD value	LenzePLCToolBox	L_DcGetBitOfDWord	Outputs a bit of a DWORD value
L_TBGetBitOfWord*	Outputs a bit of a WORD value	LenzePLCToolBox	-	-
L_TBResetBitOfByte*	Resets a bit of a BYTE value to "0".	LenzePLCToolBox	L_DcResetBitOfByte	Resets a bit of a BYTE value to "0".
L_TBResetBitOfDword*	Resets a bit of a DWORD value to "0".	LenzePLCToolBox	L_DcResetBitOfDWord	Resets a bit of a DWORD value to "0".
* Function (no function block)				

Function block for 9300 servo PLC			Function block for 9400 Servo Drives	
Name	**Description**	**Function library**	**Name**	**Description**
L_TBResetBitOfWord*	Resets a bit of a WORD value to "0".	LenzePLCToolBox	L_DcResetBitOfWord	Resets a bit of a WORD value to "0".
L_TBSetBitOfByte*	Sets a bit of a BYTE value to "1".	LenzePLCToolBox	L_DcSetBitOfByte	Sets a bit of a BYTE value to "1".
L_TBSetBitOfDword*	Sets a bit of a DWORD value to "1".	LenzePLCToolBox	L_DcSetBitOfDWord	Sets a bit of a DWORD value to "1".
L_TBSetBitOfWord*	Sets a bit of a WORD value to "1".	LenzePLCToolBox	L_DcSetBitOfWord	Sets a bit of a WORD value to "1".
L_TBSquareWave	Rectangular pulse generator with variable pulse/dead time.	LenzePLCToolBox	L_TbOscillator	Rectangular pulse generator with parameterisable pulse/dead time.
L_TpConfigDigInX	Configures a touch probe input (function)	LenzeTpDrv	-	-
L_TpGetLastScanDigIn1...4	Provides touch-probe signals	LenzeTpDrv	L_SdTouchProbe	Touch probe processing
L_TRANS	Edge evaluation	LenzeDrive.lib	L_TbTransition	Signal edge evaluation
L_WndAccTorque	Acceleration compensation	LenzeWinder	-	-
L_WndCalcDiameter	Diameter calculation	LenzeWinder	L_WndCalcDiameter	Diameter calculation
L_WndCalcLength	Length calculation	LenzeWinder	L_WndCalcLength	Length calculation
L_WndCalcThickness	Thickness calculation	LenzeWinder	-	-
L_WndConvActPos	Actual dancer value conditioning	LenzeWinder	-	-
L_WndConvMSet	Torque setpoint converter	LenzeWinder	-	-
L_WndConvNSet	Speed setpoint converter	LenzeWinder	-	-
L_WndFrictionCurve	Friction compensation	LenzeWinder	L_Wnd FrictionCompensation	Friction compensation
L_WndGearRatio	Gearbox factor changeover	LenzeWinder	-	-
L_WndIdentFriction	Identification of the friction characteristic	LenzeWinder	L_Wnd FrictionCompensation	Friction compensation
L_WndIdentMInertia	Identification of the moment of inertia	LenzeWinder	L_WndIdentMInertia	Identification of the moment of inertia
L_WndMInertia	Mass inertia calculation	LenzeWinder	-	-
L_WndProcessCtrl	Dancer / tension control	LenzeWinder	-	-
L_WndStopCtrl	Stop controller	LenzeWinder	L_WndStopCtrl	Stop controller
L_WndTensionCurve	Tensile force characteristic	LenzeWinder	L_WndTensionCurve	Tensile force characteristic
* Function (no function block)				

4.14 FB reference list - 9300 servo inverter

The following reference list shall assist you in finding the correct function block to simulate a function known from the 9300 servo inverter.

Function block for 9300 servo inverter		Function block for 9400 Servo Drives	
Name	**Description**	**Name**	**Description**
ABS	Absolute value generator	L_TbAbs	Absolute value generation
ADD	Addition block	L_TbAdd	Addition without limitation
		L_TbAddLim	Addition with limitation
		L_TbAddSubLim	Addition and subtraction with limitation
AIF-IN	Fieldbus	-	-
AIF-OUT	Fieldbus	-	-
AIN	Analog inputs	**LS_AnalogInput** SB	Analog inputs
AND	AND	L_TbAnd	AND (2 inputs)
		L_Tb5And	AND (5 inputs)
ANEG	Analog inverter	L_TbNeg	Sign inversion
AOUT	Analog outputs	**LS_AnalogOutput** SB	Analog outputs
ARIT	Arithmetic block	L_TbAdd	Addition without limitation
		L_TbAddLim	Addition with limitation
		L_TbAddSubLim	Addition and subtraction with limitation
		L_TbDiv	Division with remainder
		L_TbDiv_n	Division divisor selected as scaled signal [%]
		L_TbMul	Multiplication without limitation
		L_TbMul_n	Multiplication with the multiplicand selected as scaled signal [%]
		L_TbMulLim	Multiplication with limitation
		L_TbSub	Subtraction without limitation
		L_TbSubLim	Subtraction with limitation
ARITPH	32 bit arithmetic block	L_TbAdd	Addition without limitation
		L_TbAddLim	Addition with limitation
		L_TbAddSubLim	Addition and subtraction with limitation
		L_TbDiv	Division with remainder
		L_TbDiv_n	Division divisor selected as scaled signal [%]
		L_TbMul	Multiplication without limitation
		L_TbMul_n	Multiplication with the multiplicand selected as scaled signal [%]
		L_TbMulLim	Multiplication with limitation
		L_TbSub	Subtraction without limitation
		L_TbSubLim	Subtraction with limitation
ASW	Analog change-over switch	L_Tb8Select	1-from-8-selector
		L_TbSelect	Selector
BRK	Holding brake control	**LS_Brake** SB	Basic function "Brake control"
CAN-IN	System bus	Network configuration in the »Engineer«, interconnection via ports in the machine application.	
CAN-OUT			
CMP	Comparator	L_TbCompare	Comparison
		L_TbCompare_n	Comparison (scaled)
CONV	Conversion	L_TbGainLim	Gain with limitation
		L_TbNormalize	Signal scaling
CONVPHA	32 bit conversion	-	-
CONVPHPH	32 bit conversion	-	-

Function block for 9300 servo inverter		Function block for 9400 Servo Drives	
Name	**Description**	**Name**	**Description**
CONVPP	32 bit/16 bit conversion	-	-
CURVE	Characteristic function	L_TbCurve	Characteristic function (scaled)
		L_TbCurveAny	Characteristic function
DB	Dead band	L_TbDeadband	Dead band
		L_TbDeadband_n	Dead band (scaled)
DCTRL	Device control	**LS_DriveInterface** SB	Drive interface
DFIN	Digital frequency input	**LS_DigitalFrequencyInput** SB	Digital frequency input
DFOUT	Digital frequency output	**LS_DigitalFrequencyOutput** SB	Digital frequency output
DFRFG	Digital frequency ramp function generator	L_CamSyncIn	Synchronous/oversynchronous clutch-in
DFSET	Digital frequency processing	-	-
DIGDEL	Binary delay element	L_TbDelay	Binary delay element
DIGIN	Digital inputs	**LS_DigitalInput** SB	Digital inputs
DIGOUT	Digital outputs	**LS_DigitalOutput** SB	Digital outputs
DT1-1	Derivative-action element	L_TbDifferentiate	Differentiator with low pass
FCNT	Piece number	L_TbCount	Upcounter and downcounter with adjustable limitation.
FEVAN	Free analog input variable	-	-
FDO	Free digital outputs	-	-
FIXED	Constant signals	-	-
FIXSET	Fixed setpoints	L_SdSpeedSet	Ramp generator (ramp function generator) with S-shaped ramps, parameterisable fixed setpoints and comprehensive parameter setting and control options.
FLIP	D flipflop	L_TbFlipFlopD	D flipflop
		L_TbFlipFlopRS	RS flipflop
GEARCOMP	Gearbox torsion	-	-
LIM	Limiter	L_TbLimit	Signal limitation
		L_TbLimit_n	Signal limitation (scaled)
MCTRL	Motor control	**LS_MotorInterface** SB	Motor interface
MFAIL	Mains failure control	-	-
MLP	Motor phase failure detection	-	-
MONIT	Monitoring	-	-
MPOT	Motor potentiometer	L_SdMotorPot	Motor potentiometer
NOT	NOT	L_TbNot	Negation (inverter)
NSET	Speed setpoint conditioning	L_SdSpeedSet	Ramp generator (ramp function generator) with S-shaped ramps, parameterisable fixed setpoints and comprehensive parameter setting and control options.
OR	OR	L_TbOr	OR (2 inputs)
		L_Tb5Or	OR (5 inputs)
OSZ	Oscilloscope function	-	-
PCTRL	Process controller	-	-
PHADD	32 bit addition block	L_TbAdd	Addition without limitation
		L_TbAddLim	Addition with limitation
		L_TbAddSubLim	Addition and subtraction with limitation
PHCMP	Comparator	L_TbCompare	Comparison
		L_TbCompare_n	Comparison (scaled)
PHDIFF	32 bit setpoint/actual value comparison	L_TbDifferentiate	Differentiator with low pass

Function block for 9300 servo inverter		Function block for 9400 Servo Drives	
Name	**Description**	**Name**	**Description**
PHDIV	Conversion	L_TbDiv	Division with remainder
		L_TbDiv_n	Division divisor selected as scaled signal [%]
PHINT	Phase integrator	L_SdIntegrate	Integration (speed → position)
		L_SdIntegrateLimit	Integration (speed → position) with adjustable upper/lower limit for return purposes
PT1-1	First-order delay element	L_TbPT1Filter	PT1 filter
R/L/Q	QSP / setpoint inversion	-	-
REF	Homing function	**LS_Homing** SB	Basic function "Homing"
RFG	Ramp function generator	L_SdRampGenerator	Ramp generator (ramp function generator) with S-shaped ramps.
		L_SdRampGeneratorAny	Unlike the L_SdRampGenerator FB, this FB is also able to process speeds (_s). Separate acceleration and deceleration times can be set.
		L_SdSpeedSet	Ramp generator (ramp function generator) with S-shaped ramps, parameterisable fixed setpoints and comprehensive parameter setting and control options.
S&H	Sample & Hold	L_TbSampleHold	Sample & Hold
SRFG	S-shaped ramp function generator	L_SdRampGenerator	Ramp generator (ramp function generator) with S-shaped ramps.
		L_SdRampGeneratorAny	Unlike the L_SdRampGenerator FB, this FB is also able to process speeds (_s). Separate acceleration and deceleration times can be set.
		L_SdSpeedSet	Ramp generator (ramp function generator) with S-shaped ramps, parameterisable fixed setpoints and comprehensive parameter setting and control options.
STAT	Digital status signals	L_DevDriveInterfaceStateDecoder	Status signals of the drive interface
		L_DevManualJogStateDecoder	Status signals of the basic function "Manual jog"
		L_DevHomingStateDecoder	Status signals of the basic function "Homing"
		L_DevPositionerStateDecoder (from V2.0)	Status signals of the basic function "Positioning"
		L_PosDecoderStatePositioner	
		L_DevLimiterStateDecoder	Status signals of the basic function "Limiter"
		L_DevBrakeStateDecoder	Status signals of the basic function "Brake control"
		L_PosSequencerStateDecoder*	Status signals of the sequence control (FB L_PosSequencer)
		L_PosDecoderStateSequencer	
		L_LdStateDecoder	Status signals of the FBs of the function library "LenzeLineDrive.lib"
		L_DevSMControlDecoder (from V2.0) L_DevSMStateDecoder (from V2.0) L_DevSMStateDecoderIO (from V2.0)	Control/status signals from the SM3xx safety module
STATE-BUS	Statebus	**LS_DigitalInput** SB	Digital inputs
		LS_DigitalOutput SB	Digital outputs
STORE	Memory	-	-
SYNC	Multi-axis positioning	-	-
TRANS	Binary edge evaluation	L_TbTransition	Signal edge evaluation

5 Function blocks

This chapter describes all function blocks (FBs) that are provided by the »Engineer« function block editor for the "9400 Servo Drives".



Note!

The grayed out area at the beginning of each FB description informs you on the function library which contains the corresponding FB and provides information concerning the "9400 Servo Drives" that can be used with the FB.



Tip!

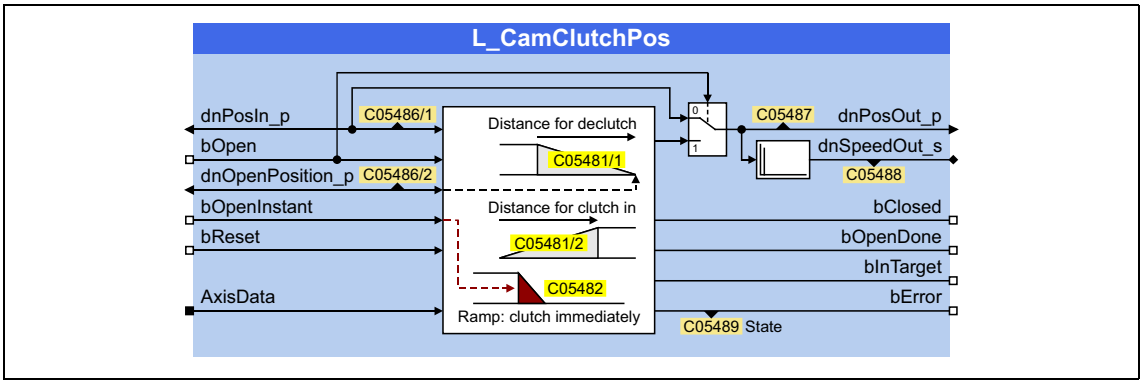
For an external access (e.g. from a master control) to the parameters of the parameterisable function blocks: The Table of attributes contains information required for communication to the controller via parameters. (📖 656)

5.1 L_CamClutchPos - clutch (path-controlled)

Function library:	LenzeCam	**FB permissible for firmware as of V3.0 only!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

This FB serves to implement a path-controlled clutching and declutching of the master position for all traversing ranges (Modulo/limited/unlimited) via a fifth-order polynomial.

- More functions:
  - Positive opening operation of the clutch (for emergency situations)
  - "Hard clutch-in"
- For time-controlled clutching and declutching, use the L_EsClutchPos FB. (📖 290)



Inputs

Identifier/data type	Information/possible settings	
dnPosIn_p DINT	Input position in [increments] • Selection of an external master position. • C05486/1 indicates the input position in the real unit of the machine.	
bOpen BOOL	Control of the clutch function	
	TRUE	"Open clutch" request (declutch process). • The *dnPosOut_p* output position from the declutch position disengages from the *dnPosIn_p* master position and is decelerated to the *dnOpenPosition_p* standstill position. • The declutch position indirectly results from the relative declutch path set in C05481/1. • If the *dnPosOut_p* output position has reached the standstill position, the two outputs *bInTarget* and *bOpenDone* are set to TRUE.
	FALSE	"Close clutch" request (clutch-in process). • The clutch-in process can only be started after the declutch process has been totally completed; an early abort of the declutch process is not possible. • The *dnPosOut_p* output position is resynchronised to the *dnPosIn_p* master position and then follows it. • The position from which the input and output position are synchronous again indirectly results from the relative clutch-in path set in C05481/2. • If the input and output position are synchronous again, the output *bClosed* is set to TRUE.
dnOpenPosition_p DINT	Standstill position after the declutch process in [increments] • C05486/2 indicates the standstill position in the real unit of the machine.	

Identifier/data type	Information/possible settings	
bOpenInstant BOOL	Positive opening operation of the clutch (for emergency situations) • This input has a higher priority than the input *bOpen*.	
	FALSE↗TRUE	The *dnPosOut_p* output position is separated immediately from the *dnPosIn_p* master position and brought to standstill within the deceleration time set in C05482. • It is not possible to abort the positive opening operation early. • If the *dnPosOut_p* output position is at standstill, the output *bOpenDone* is set to TRUE. • Afterwards a clutch-in process by a FALSE-TRUE edge at the input *bOpen* can only be effected again after the input *bOpenInstant* has been reset to FALSE.
bReset BOOL	Reset declutch process. • This function allows for a "hard clutch-in" and can be used for certain interruption mechanisms or for starting up the machine. • The reset of the declutch process should be only effected during controller inhibit or a decoupled setpoint path. • This input has the highest priority.	
	TRUE	The *dnPosOut_p* output position is set immediately to the *dnPosIn_p* master position. • The output *bClosed* is set to TRUE.
AxisData	Machine parameters • If the FB is to relate to the measuring system of the drive axis, connect this input to the output *DI_AxisData* of the **LS_DriveInterface** SB for accepting the machine parameters of the drive/motor. • If the FB within a cam application is to relate to the measuring system of the master selection (master), connect this input to the output *XAxisData* of the L_CamProfiler FB instead. (📖 162) • If this input is left unconnected, the position values are processed with the following resolution: 1 encoder revolution ≡ 2^{16} increments	

Outputs

Identifier/data type	Value/meaning	
dnPosOut_p DINT	Output position in [increments] • C05487 indicates the output position in the real unit of the machine.	
dnSpeedOut_s DINT	Output speed given as speed in [rpm] • C05488 indicates the output speed in the real unit of the machine.	
bClosed BOOL	Status signal to the clutch status	
	FALSE	The clutch is open, or an active clutch-in/declutch process is active. • The output position is at standstill or within an acceleration/deceleration phase.
	TRUE	The clutch is closed. • Output position *dnPosOut_p* and input position *dnPosIn_p* are synchronous.
bOpenDone BOOL	Status signal "Declutch process completed"	
	TRUE	The declutch process activated via *bOpen* or *bOpenInstant* has been totally completed; a clutch-in process can be effected again.
bInTarget BOOL	Status signal "Standstill position reached"	
	TRUE	The standstill position *dnOpenPosition_p* after a declutch process activated via *bOpen* has been reached.
bError BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameters C05489.

Parameter

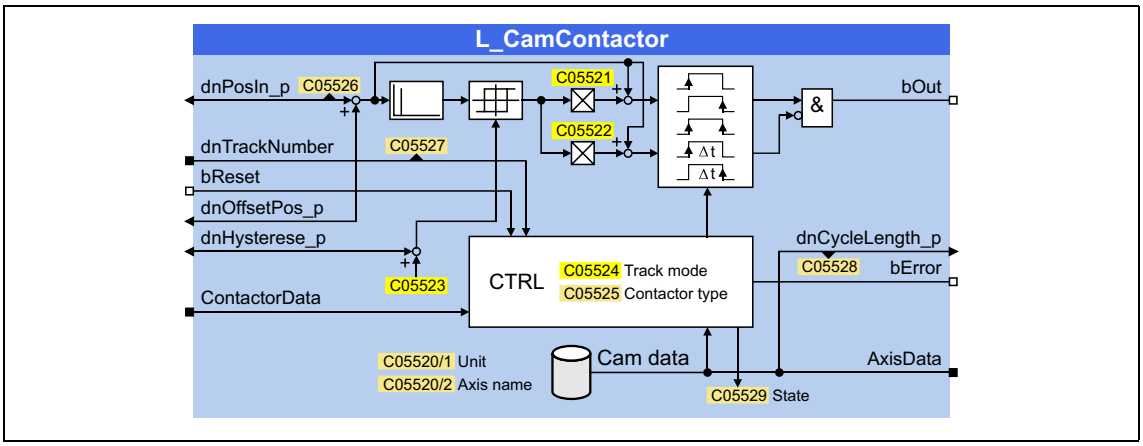
Parameter	Possible settings			Information
C05480/1	String of digits			Position unit • Read only
C05480/2	String of digits			Speed unit • Read only
C05481/1	-214748.3647	Unit	214748.3647	Relative declutch path • Relating to the *dnOpenPosition_p* standstill position. • Initialisation: 90.0000 unit
C05481/2	-214748.3647	Unit	214748.3647	Relative clutch-in path • Relating to the position from which the output position *dnPosOut_p* and input position *dnPosIn_p* are synchronous again. • Initialisation: 90.0000 unit
C05482	0.000	s	130.000	Deceleration time for positive opening operation • Initialisation: 1.000 s
C05486/1	-214748.3647	Unit	214748.3647	Input position • Display of the *dnPosIn_p* input signal in the real unit of the machine.
C05486/2	-214748.3647	Unit	214748.3647	Standstill position • Position after declutch process • Display of the *dnOpenPosition_p* input signal in the real unit of the machine.
C05487	-214748.3647	Unit	214748.3647	Output position • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C05488	-214748.3647	Unit/t	214748.3647	Output speed • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.
C05489	Status messages:			Status (bit coded) • Bits that are not itemised are reserved for future extensions. • The *bError* error output will only be set to TRUE if an error message (bit 16 ... 31) is issued. • Bit 31 is the default setting if one or more other error bits (bit 16 ... 30) are set.
	bits8	Clutch closed		
	bits9	Declutch position reached		
	Error messages:			
	bit16	Position is beyond the cycle		
	bits23	Invalid axis data structure		
	bits31	General error		

5.2 L_CamContactor - cam group with switch-on and switch-off dynamisation

Function library:	LenzeCam	**FB permissible for firmware as of V3.0 only!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

From the cam data under the current product number (specified by the **LS_CamInterface** SB), this FB reads the cam track selected via the input *dnTrackNumber* and outputs the cam status on the specified position.

- A cam track can contain up to four cams of the same type (path or time cam); however, the cam type can be different for each product.
- As of Engineer version 2.9 up to 16 cams per cam track are supported.
- By means of a parameterisable dynamic displacement of the cams, deceleration times of switching elements connected can be compensated.
- If further cam tracks are to be evaluated, further instances of this FB just have to be added to the application.



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Input position in [increments] • Position of the x axis or y axis (according to the application). • C05526 indicates the input position in the real unit of the machine.
dnTrackNumber DINT	Selection of the cam track of the cam data available in the memory module. **Note:** If the *ContactorData* input is assigned, the contactor data transferred via the *ContactorData* input are used. ▸ Selection of the cam data (📖 146)
bReset BOOL	Reset output signals & operating times TRUE: The outputs *bOut* and *bError* are reset to FALSE. • In the case of time cams, a time that has possibly started is reset. The "Cam ON" status is only output again if the switch-on condition has been met again.
dnOffsetPos_p DINT From library V01.02.xx.xx	Offset in [inc] • For position-dependent offset of the cam track.

Identifier/data type	Information/possible settings
dnHysteresePos_p DINT From library V01.02.xx.xx	Switching hysteresis in [increments] • Is added to the switching hysteresis set in C05523.
ContactorData From library V01.02.xx.xx	Cam data • Interface for optional acceptance of the contactor data from the FB L_CamSetContDataPDO or FB L_CamSetContDataSDO. These two FBs enable the direct selection of contactor data via inputs/PDOs or codes/SDOs (e.g. from a master control). • If contactor data are pending at this input, these will be used instead of the contactor data available in the cam data.

Outputs

Identifier/data type	Value/meaning	
bOut BOOL	Cam status of the cam track selected	
	FALSE	Cam switched off.
	TRUE	Cam switched on.
dnCycleLength_p DINT From library V01.01.xx.xx	Cycle in [increments] • C05528 indicates the cycle in the real unit of the machine.	
bError BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameters C05529.
AxisData	Machine parameters • Via this output the machine parameters used by the FB can be easily transferred to subsequent FBs.	

Parameter

Parameter	Possible settings			Information
C05520/1	String of digits			Position unit • Read only
C05520/2	String of digits			Designation of the measuring system • Read only
C05521	-1000.000	ms	1000.000	Switch-on position displacement • Initialisation: 0.000 ms
C05522	-1000.000	ms	1000.000	Output position displacement • Initialisation: 0.000 ms
C05523	-214748.3647	Unit	214748.3647	Hysteresis for switching threshold • If the input position is not stable and the output *bOut* oscillates.
C05524				Track switch-over mode
	0	In the zero crossing/cycle		Lenze setting • Use for modulo traversing range.
	1	Immediately		This setting should be used for a limited traversing range.

Parameter	Possible settings			Information
C05525				Cam type • Read only
	0	Unknown cam type		
	1	Positive travel cams		
	2	Negative travel cams		
	3	Bidirectional travel cams		
	11	Positive time cams		
	12	Negative time cams		
C05526	-214748.3647	Unit	214748.3647	Input position • Display of the *dnPosIn_p* input signal in the real unit of the machine.
C05527	0		2147483647	Cam track • Display of the track number *dnTrackNumber*.
C05528 From library V01.01.xx.xx	0.0000	Unit	214748.3647	Cycle • Display of the *dnCycleLength_p* output signal in the real unit of the machine.
C05529	Status messages:			Status (bit coded) • Bits that are not itemised are reserved for future extensions. • The *bError* error output will only be set to TRUE if an error message (bit 15 ... 31) is issued. • Bit 31 is the default setting if one or more other error bits (bit 15 ... 30) are set.
	bits6	Reset of cam active		
	bits10	Zero crossing active		
	bits11	Cam switched on		
	Error messages:			
	Bit15	Limitation of dynamic cam sampling		
	bit16	Position is beyond the cycle		
	bits23	Invalid axis data structure		
	bits25	Invalid data track number		
	bits26	Invalid data track type		
	bits31	General error		

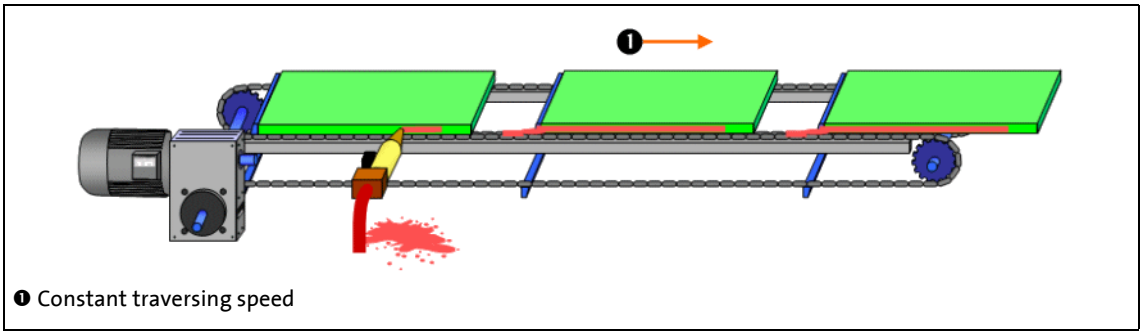
5.2.1 Typical application

Via the function of the cam group binary output signals with a reference to curves can be generated very simply.

Via digital switch cams for instance the quick control of external valves, glue nozzles, or similar peripheral elements can be carried out, which are connected to the digital outputs of the controller for this purpose. Of course the switch cams can also be used for controlling internal functions.

Example 1

Glueing of furniture panels by means of a glue nozzle controlled via a static cam signal (no compensation of the operating delay time of the glue valve):



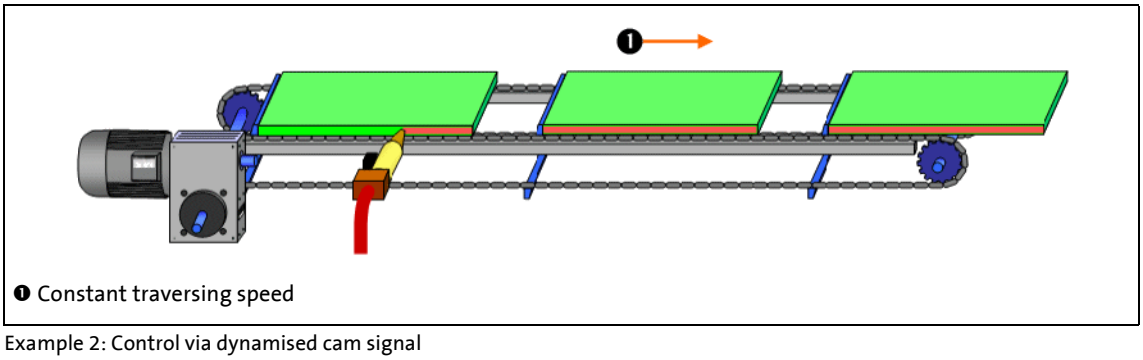
❶ Constant traversing speed

[5-1] Example 1: Control via static cam signal

Because the glue valve switches on/off in a delayed manner (operating delay time), the furniture panels are not glued from the leading edge. Correspondingly late the glue nozzle is switched off again: Due to the switch-off delay time, glue continues to emit from the nozzle although the furniture panel has already left the glue area.

Example 2

Glueing of furniture panels by means of a glue nozzle controlled via a dynamised cam signal (no compensation of the operating delay time of the glue valve):



❶ Constant traversing speed

[5-2] Example 2: Control via dynamised cam signal

5.2.2 Data flow

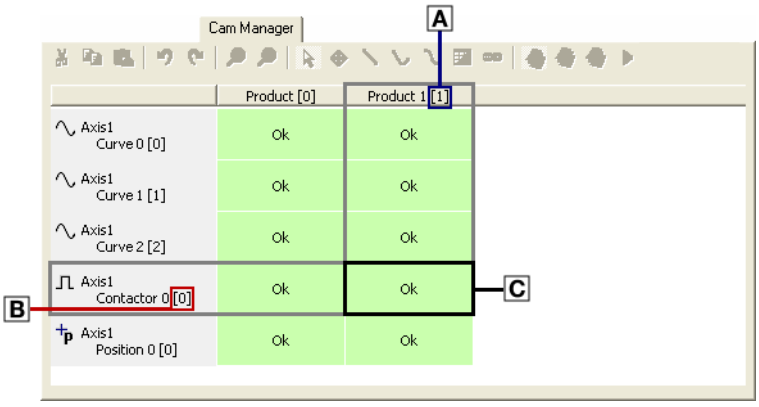
The cams are entered in the »Cam Designer«. The transmission of the cam data to the controller(s) – together with the corresponding parameter set – is effected via the »Engineer« by executing the command **Transfer parameter set to the device**. Afterwards the parameter set and the cam data have to be saved in the memory module with mains failure protection (e. g. with the device command C00002 = "11: Save starting parameters").

Tip!

The cams can also be entered directly via the parameters of the basic drive function "Cam data management" (**LS_CamInterface** SB) if cam data have already been downloaded to the controller.

Selection of the cam data to be used within the application

Within the application the selection of the cam data to be used is then carried out via the product number and the track number which have been defined for the axis in the »Cam Manager«:



Ⓐ Product number
Ⓑ Track number
Ⓒ Cam data

- The selection of the product number Ⓐ and therefore the selection of the product to be manufactured for the application is effected globally via the input *CAM_dnProductNumber* of the **LS_CamInterface** SB and has to be carried out synchronously for all cam drives in the interconnection.



- The selection of the track number Ⓑ of the cam data track, however, is effected individually via the input *dnTrackNumber* of the corresponding entity of the **L_CamContactor** FB.

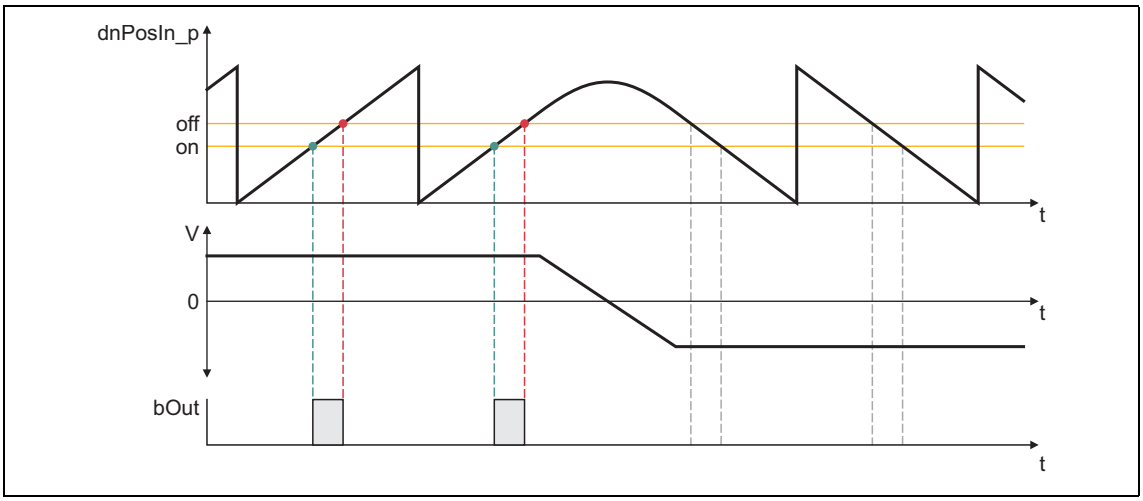


5.2.3 Switching performance for position cams

Position cams in positive effective direction

If the input position *dnPosIn_p* reaches the cam switch-on position in positive effective direction, the output *bOut* is set to TRUE and is reset to FALSE if the switch-off position is exceeded.

- If a change in direction is effected after the switch-on position has been reached, the output *bOut* is reset to FALSE immediately.

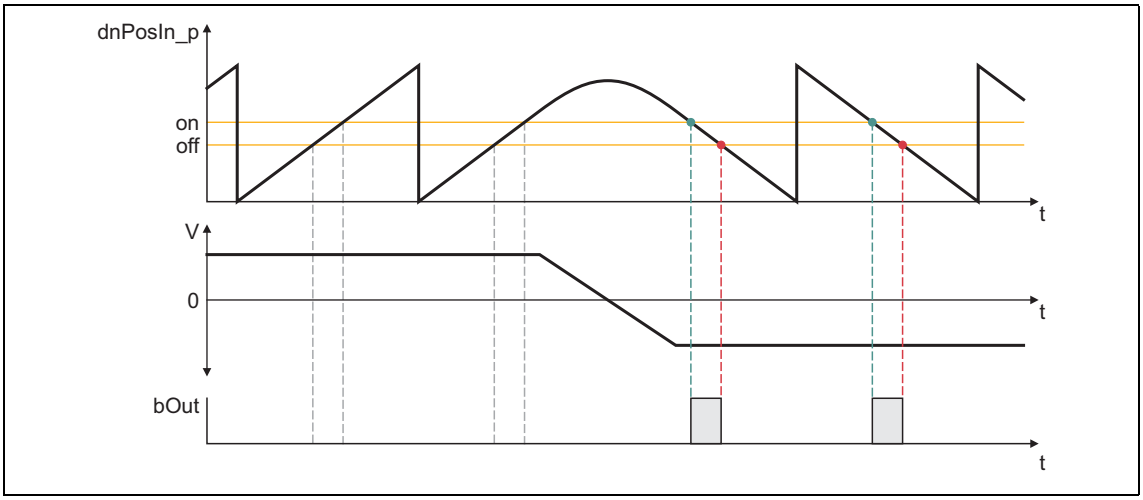


[5-3] Example: Position cams in positive effective direction

Position cams in negative effective direction

If the input position *dnPosIn_p* reaches the cam switch-on position in negative effective direction, the output *bOut* is set to TRUE and is reset to FALSE if the switch-off position is exceeded.

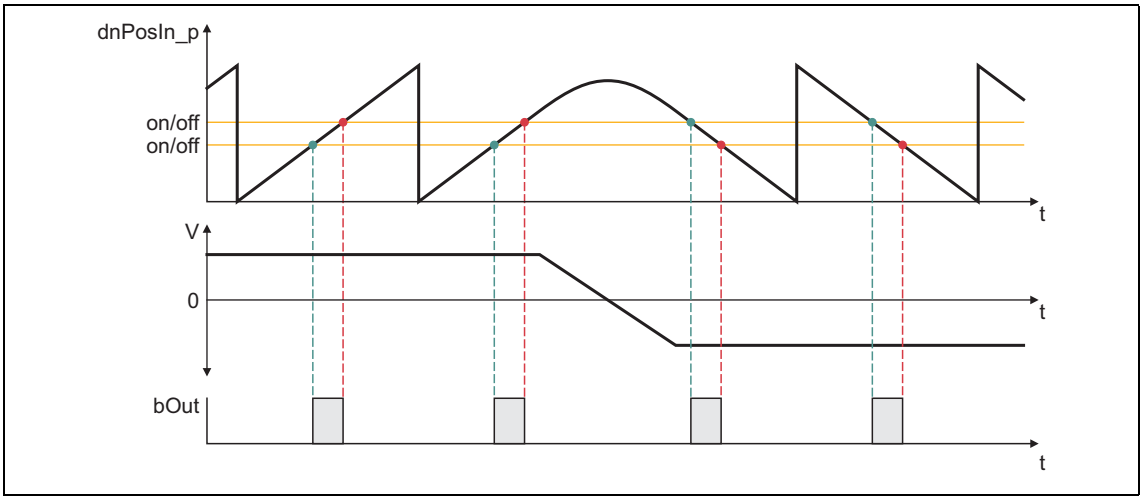
- If a change in direction is effected after the switch-on position has been reached, the output *bOut* is reset to FALSE immediately.



[5-4] Example: Position cams in negative effective direction

Position cams with bipolar effective direction

If the input position *dnPosIn_p* is positioned within the cam switch-on and switch-off position, the output *bOut* is set to TRUE, irrespective of the process flow direction.



[5-5] Example: Position cams with bipolar effective direction

5.2.4 Switching performance for position-time cams

Note!

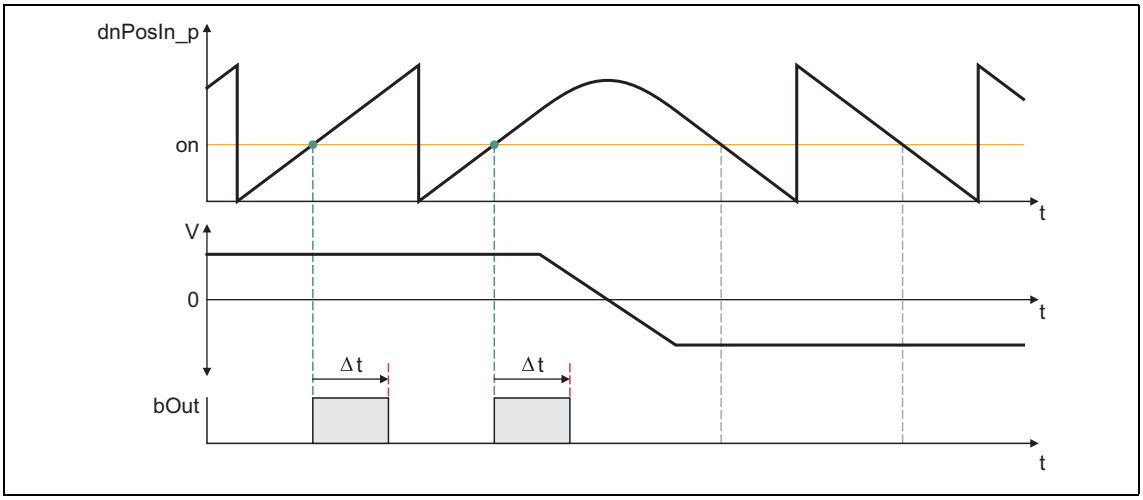
When position-time cams are used, overlapping switch-on states of subsequent cams can occur if the operating time of the position-time cams is longer than the interval to the switch-on position of the following position-time cams (post-trigger function).

If the operating time of a position-time cam is longer than the total cycle time, the output *bOut* remains on TRUE permanently.

Position-time cams in positive effective direction

If the input position *dnPosIn_p* reaches the cam switch-on position in positive effective direction, the output *bOut* is set to TRUE and is only reset to FALSE after the time defined has elapsed.

- Once the output is set to TRUE, a following change in direction has no impact on the cam signal.

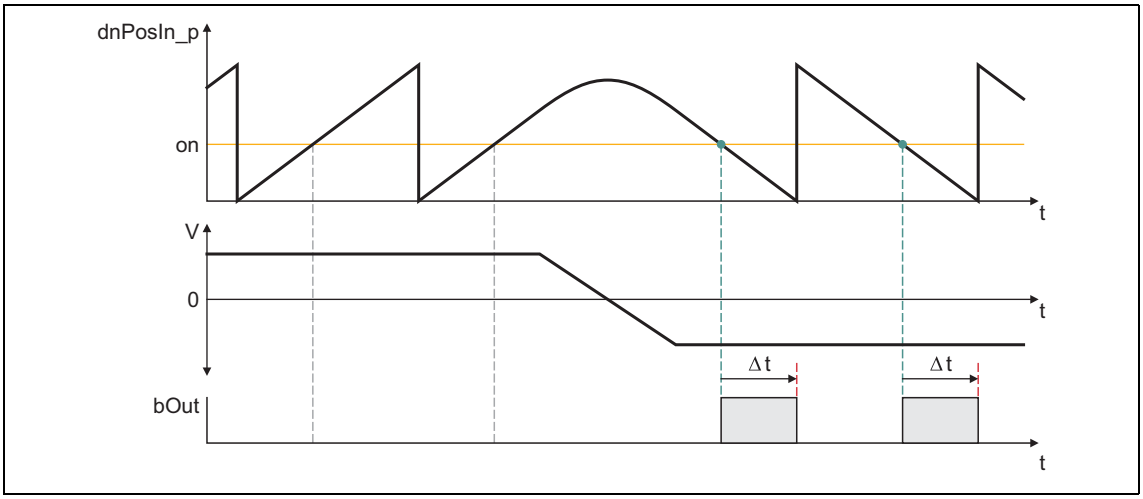


[5-6] Example: Position-time cams in positive effective direction

Position-time cams in negative effective direction

If the input position *dnPosIn_p* reaches the cam switch-on position in negative effective direction, the output *bOut* is set to TRUE and is only reset to FALSE after the time defined has elapsed.

- Once the output is set to TRUE, a following change in direction has no impact on the cam signal.

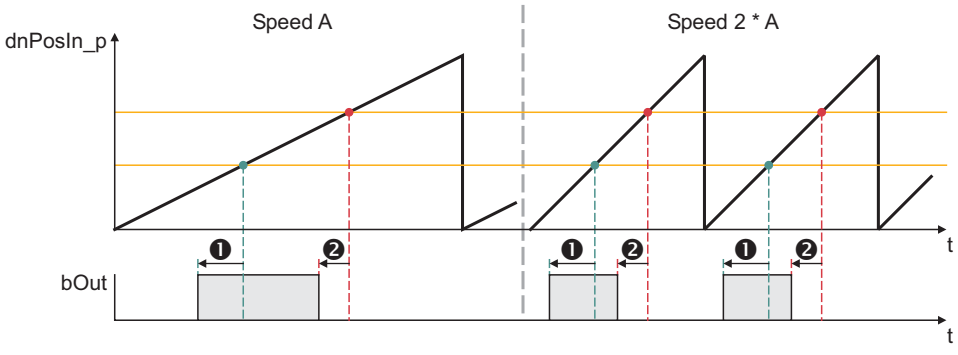


[5-7] Example: Position-time cams in negative effective direction

5.2.5 Dynamic displacement of the cams

In C05521 and C05522 a time displacement of the switch-on and switch-off positions can be set for the cams, e. g. to implement a compensation of delay times of switching elements connected.

- A positive time period causes the cam to be switched earlier, a negative time period accordingly causes the switching operation to occur later.
- For position-time cams only a time displacement of the switch-on position is effected via C05521; the setting in C05522 has no effect.
- Dynamic cam sampling is limited by the half-cycle period. If the limitation is activated, the *bError* output is set to TRUE and the limitation is signalled via error bit 15 in C05529.



❶ Positive time displacement of the switch-on position (causes earlier switch-on)
❷ Negative time displacement of the switch-off position (causes late switch-off)

[5-8] Displacement of the switch-on and switch-off position for different speeds

5.2.6 Selection of the cam data

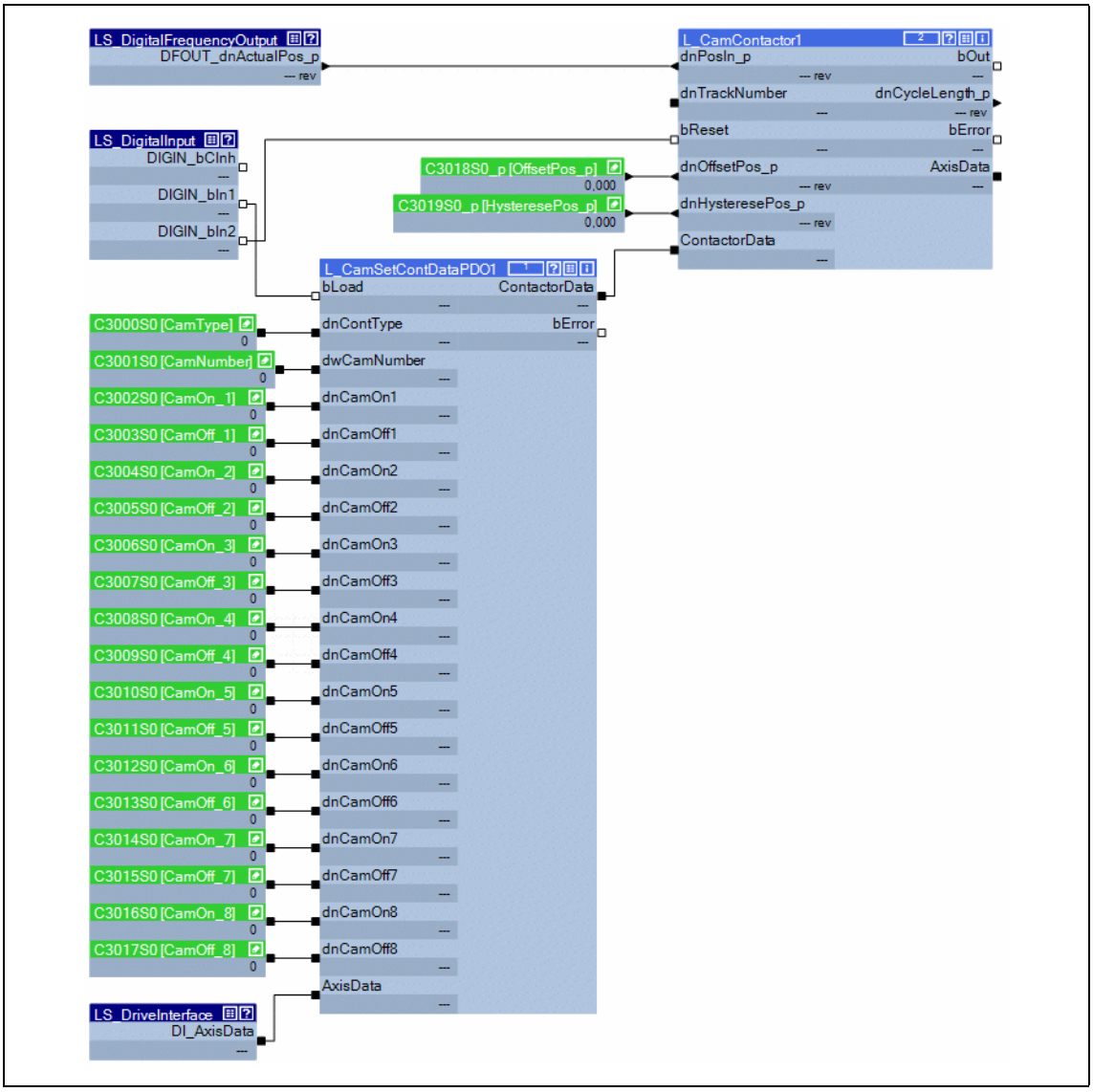
The contactor data can either be part of the cam data available in the memory module or be determined by a master control via PDOs or SDOs.

- If the FB uses the contactor data stored in the memory module, the corresponding track number must be determined in the *dnTrackNumber* input.
- The FB L_CamSetContDataPDO serves to determine the contactor data via inputs/PDOs. The contactor data are transferred to the FB **L_CamContactor** via the *ContactorData* input (see fig. [5-9]).
- The FB L_CamSetContDataSDO serves to determine the contactor data via inputs/SDOs. The contactor data are transferred to the FB **L_CamContactor** via the *ContactorData* input (see fig. [5-10]).

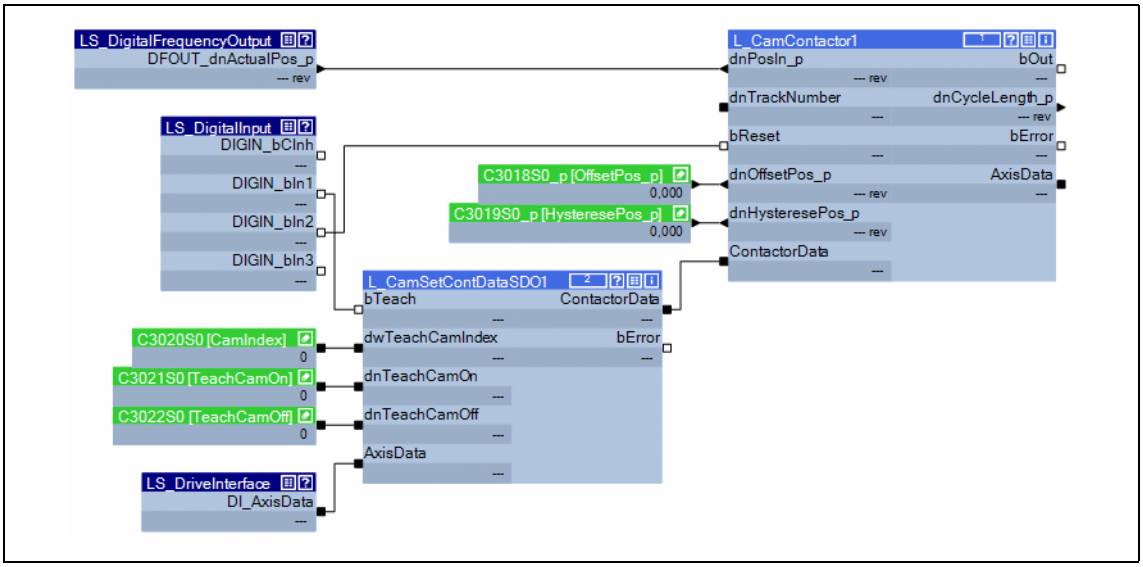
Note!

If the inputs *dnTrackNumber* and *ContactorData* are assigned simultaneously, the contactor data transferred via the *ContactorData* input is used. If this connection is separated, it is changed over to the contactor data stored in the memory module.

5.2.7 Interconnection examples



[5-9] Interconnection example 1: Selection of the contactor data via PDOs



[5-10] Interconnection example 2: Selection of the contactor data via SDOs

5.3 L_CamCurve - curve interpolation for characteristics

Function library:	LenzeCam	**FB permissible for firmware as of V3.0 only!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

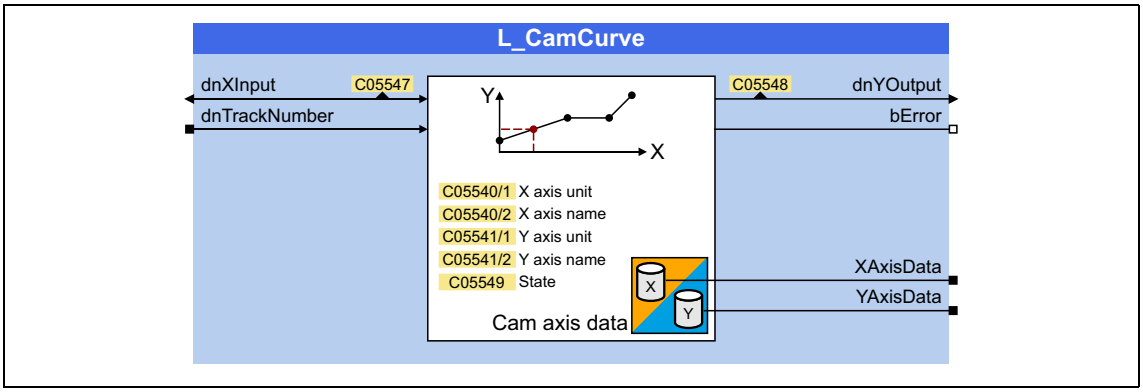
From the cam data under the current product number (specified by the **LS_CamInterface** SB), this FB reads the table of values of the curve track selected via the input *dnTrackNumber* and outputs the corresponding y value for the x value specified.

- Between two points of the table of values a linear interpolation takes place.
- This FB presents a reduced variant of the L_CamProfiler FB:
  - The FB does not process modulo measuring systems, but reverts solely to the interpolation point table stored.
  - The FB does not have a synchronised track change algorithm.
- For a curve interpolation of a motion profile the L_CamProfiler FB is to be used instead.

Stop!

Ensure that the x measuring system of the curve track matches the input signal, as otherwise serious malfunctions within the application may occur!

The y measuring system has to be defined as characteristic, otherwise an error is generated.



Inputs

Identifier/data type	Information/possible settings
dnXInput DINT	Input value (x axis) • C05546 indicates the input value in the real unit of the machine.
dnTrackNumber DINT	Selection of the curve track • The acceptance is effected immediately. • A switch-over of the track number (not product number) synchronised with the zero crossing can be implemented with an additional function block interconnection. ▸ Track switch-over (📖 153)

Outputs

Identifier/data type	Value/meaning	
dnYOutput DINT	Output value (y axis) • C05547 indicates the output value in the real unit of the machine.	
bError BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameters C05549.
XAxisData	Machine parameters of the x axis (zero for characteristic) • Via this output the machine parameters used by the FB can be easily transferred to subsequent FBs.	
YAxisData	Machine parameters of the y axis (zero for characteristic) • Via this output the machine parameters used by the FB can be easily transferred to subsequent FBs.	

Parameter

Parameter	Possible settings			Information
C05540/1	String of digits			Unit of the x axis • Read only
C05540/2	String of digits			Designation of the x axis • Read only
C05541/1	String of digits			Unit of the y axis • Read only
C05541/2	String of digits			Designation of the y axis • Read only
C05547	-214748.3647	Unit	214748.3647	Input position • Display of the *dnXInput* input signal in the real unit of the machine.
C05548	-214748.3647	Unit	214748.3647	Output position • Display of the output signal *dnYOutput* in the real unit of the machine.
C05549	Error messages:			Status (bit coded) • Bits that are not itemised are reserved for future extensions. • Bit 31 is the default setting if one or more other error bits (bit 16 ... 30) are set.
	bits25	Invalid data track number		
	bits26	Invalid data track type		
	bits31	General error		

5.3.1 Data flow

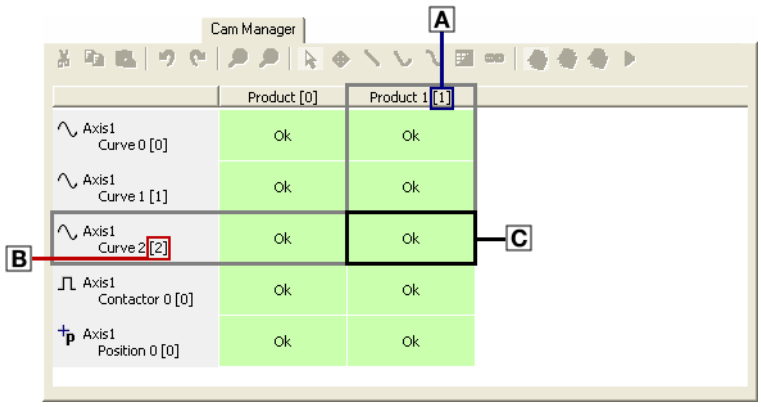
The characteristics (curves) are entered or imported in the »Cam Designer« by means of graphical objects (point, line, polynomial). The transmission of the cam data to the controller(s) – together with the corresponding parameter set – is effected via the »Engineer« by executing the command **Transfer parameter set to the device**. Afterwards the parameter set and the cam data have to be saved in the memory module with mains failure protection (e. g. with the device command C00002 = "11: Save starting parameters").

Tip!

The characteristics (curves) can also be entered directly via the parameters of the basic drive function "Cam data management" (**LS_CamInterface** SB) if cam data have already been downloaded to the controller.

Selection of the cam data to be used within the application

Within the application the selection of the data to be used is then carried out via the product number and the track number which have been defined for the axis in the »Cam Manager«:

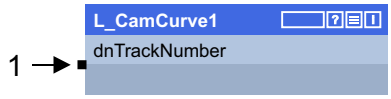


A Product number
B Track number
C Characteristic data

- The selection of the product number A is effected globally for the application via the input *CAM_dnProductNumber* of the **LS_CamInterface** SB:



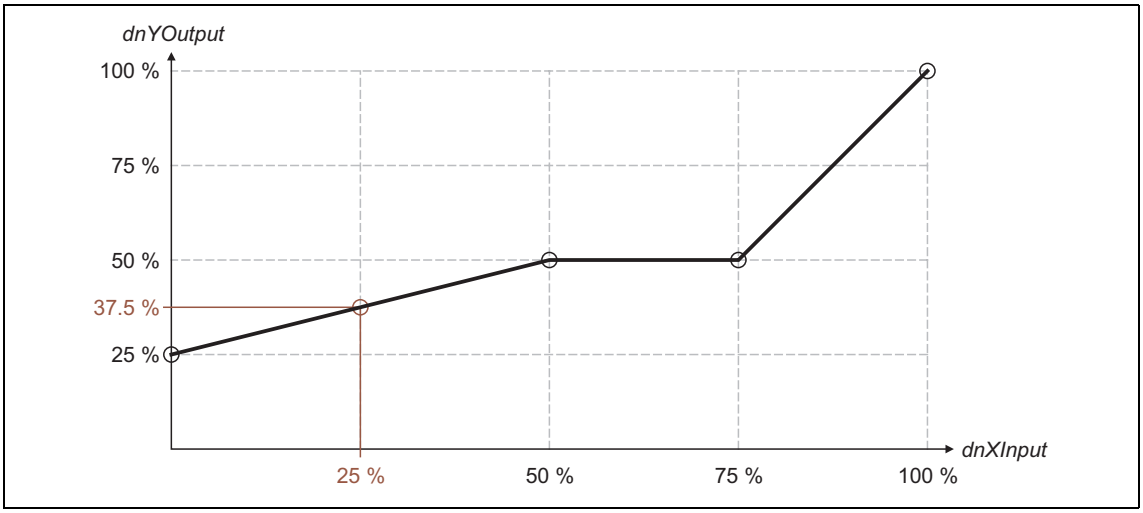
- The selection of the track number B of the characteristic, however, is effected individually via the input *dnTrackNumber* of the corresponding entity of the **L_CamCurve** FB.



5.3.2 Function

The following characteristic with 4 grid points and a percentage scaling of the x and y axis are given:

Grid point	1	2	3	4
x value	0 %	50 %	75 %	100 %
y value	25 %	50 %	50 %	100 %



[5-11] Example: characteristic

Assumption: At the input *dnXInput* the value "25 %" is available.

1. The FB first identifies the two adjacent grid points, in the example the grid points 1 and 2.
2. Afterwards the FB interpolates linearly between the two grid points and shows the interpolated value (in our example "37.5 %") at the output *dnYOutput*.

Scaling of characteristics

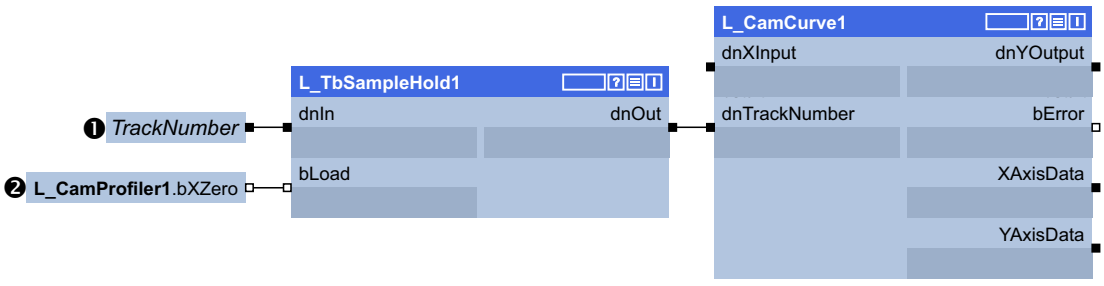
The scaling of characteristics cannot only be preselected in the »Cam Designer«/»Cam Editor« in percent, but can be set independently for the x axis and the y axis (e. g. percent, increments, or every other application unit of an existing measuring system).

Note!

Observe that the corresponding input and output signals are correctly processed according to the respective scaling.

5.3.3 Track switch-over

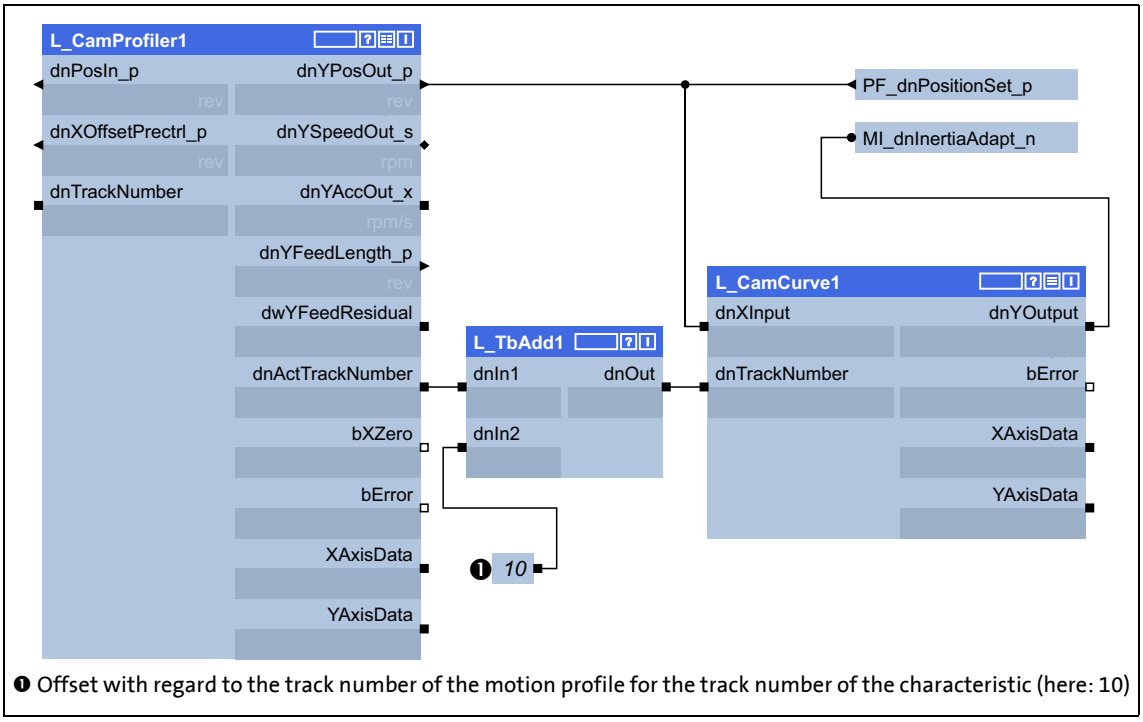
A switch-over to another characteristic via the input *dnTrackNumber* is effected immediately. A switch-over of the track number (not product number) synchronised with the zero crossing – derived from the display of the zero crossing at the L_CamProfiler FB – can be implemented with the following interconnection:



❶ Track number
❷ Status output signal "zero crossing" of the L_CamProfiler FB

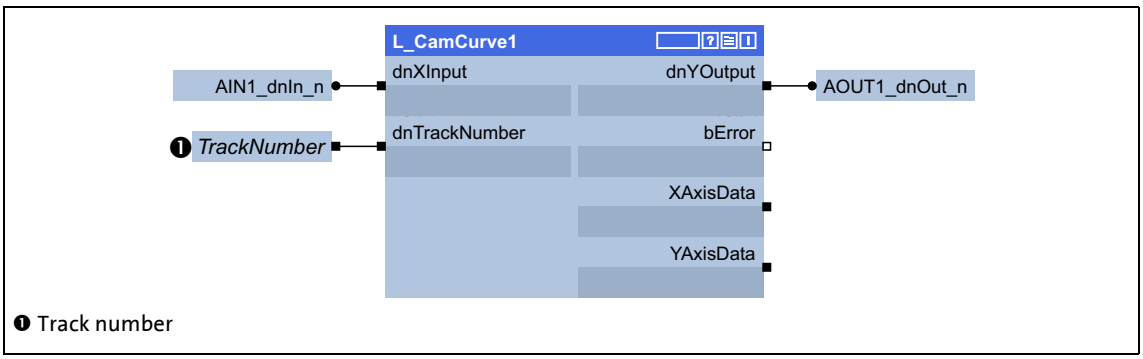
[5-12] Interconnection example for only accepting new cam data in the zero crossing of the x axis

5.3.4 Interconnection examples



❶ Offset with regard to the track number of the motion profile for the track number of the characteristic (here: 10)

[5-13] Interconnection example 1: Use of a characteristic for load adaption



❶ Track number

[5-14] Interconnection example 2: Simple analog characteristics

5.4 L_CamGetAxisData - reading out machine parameters from cam data

Function library:	LenzeCam	**FB only permissible for firmware from V3.0 onwards!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

This FB reads out the machine parameters of the axis selected via the input *dnAxisNumber* from the cam data and via the output *AxisData* provides them to subsequent FBs requiring these data for internal calculations (e. g. master value integrators, FBs for stretching/compression and other FBs for processing cam data).

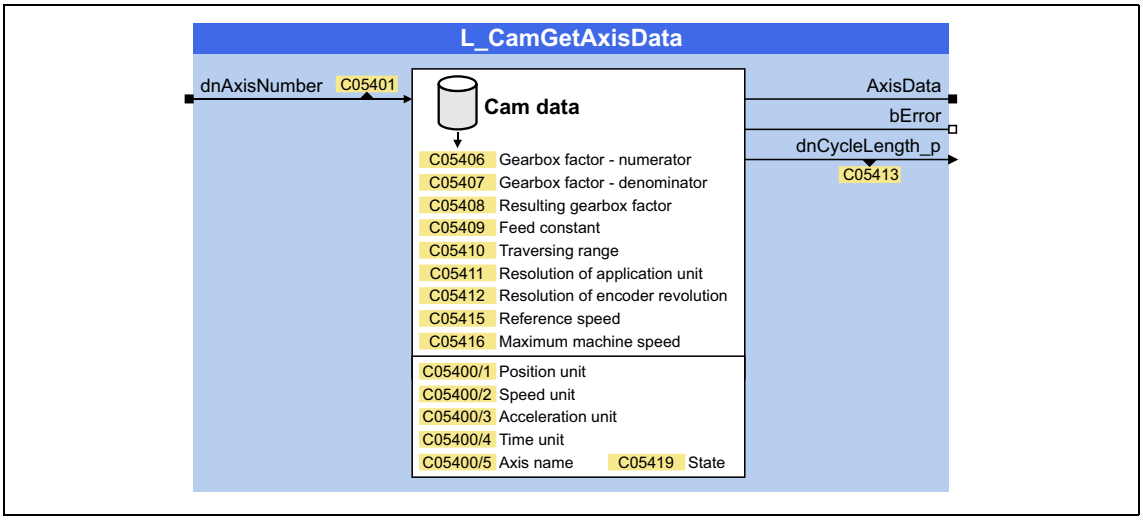
- If it is used within curve applications, the FB at least has to be instanced twice:
  - Instance 1 for the 1-measuring system (e. g. the measuring system of the x axis)
  - Instance 2 for the 2-measuring system (e. g. the measuring system of the y axis)
- Further instances for other measuring systems available (e. g. characteristic) are then to be added according to the application.

Note!

In order to enable the FB to read out machine parameters from a measuring system, the corresponding measuring system must be used in the »Cam Manager« at least once in a curve, cam, or position marker track!

Access to the machine parameters of the drive/motor

The machine parameters of the drive/motor via the output *DI_AxisData* of the **LS_DriveInterface** SB are provided to other function blocks requiring these data for internal calculations.



Inputs

Identifier/data type	Information/possible settings
dnAxisNumber DINT	Number of the axis the machine data of which are to be output at *AxisData*.

Outputs

Identifier/data type	Value/meaning	
AxisData	Machine parameters of the axis selected • In order to transfer the machine parameters to another FB, this output just has to be connected to the input *AxisData* of the corresponding FBs.	
bError BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameters C05419.
dnCycleLength_p DINT From library V01.01.xx.xx	Cycle in [increments] • C05413 indicates the cycle in the real unit of the machine.	

Parameters

Note!

Observe that the index of the codes listed in the following only applies to the first instance!

Parameter	Possible settings			Info
C05400/1	Character string			Position unit • Read only
C05400/2	Character string			Speed unit • Read only
C05400/3	Character string			Acceleration unit • Read only
C05400/4	Character string			Time unit • Read only
C05400/5	Character string			Axis designation • Read only
C05401	0		65535	Axis number • Display of the currently valid axis number from the axis data.
C05406	1		2147483647	Gearbox factor numerator • Read only
C05407	1		2147483647	Gearbox factor denominator • Read only
C05408	0.0000		214748.3647	Resulting gearbox factor • Read only
C05409	0.0001	Unit	214748.3647	Feed constant • Read only
C05410				Measuring system • Read only
	0	Unlimited		
	1	Limited		
	2	Modulo		
C05411	0.0000	Incr./unit	214748.3647	Resolution - application unit • Read only
C05412	10	Bit	24	Resolution of encoder revolution • Read only

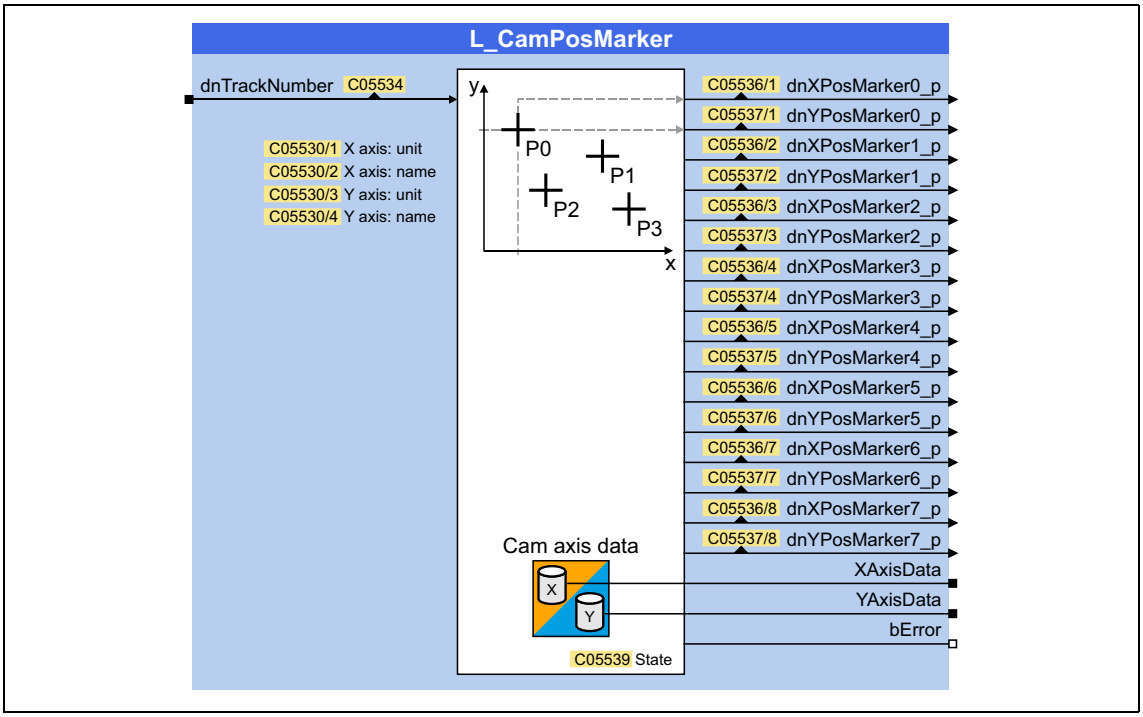
Parameter	Possible settings			Info
C05413	0.0000	Unit	214748.3647	Cycle • Read only
C05414	0.0000	Unit/t	214748.3647	Reference speed • Read only
C05415	0.0000	Unit/t	214748.3647	Max. machine speed • Read only
C05419	Error messages:			Status (bit-coded) • Bits that are not listed are reserved for future extensions. • Bit 31 is always set if one or more of the other error bits (bit 16 ... bit 30) are set.
	Bit16	Position is beyond the cycle		
	Bit 31	General error		

5.5 L_CamPosMarker - reading out position marks from cam data

Function library:	LenzeCam	**FB permissible for firmware as of V3.0 only!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

From the cam data under the current product number (specified via the **LS_CamInterface** SB), this FB reads out the track with position markers selected via the input *dnTrackNumber* and provides the corresponding x and y positions for processing within the function block interconnection.

- A position data track can contain up to four position markers.
- As of Engineer version 2.9 up to eight position markers per position data track are supported.



Inputs

Identifier/data type	Information/possible settings
dnTrackNumber DINT	Selection of the position marker track

Outputs

Identifier/data type	Value/meaning
dnXPosMarker0_p DINT	Position marker 0: x position in [increments] • C05536/1 indicates the position in the real unit of the machine.
dnYPosMarker0_p DINT	Position marker 0: y position in [increments] • C05537/1 indicates the position in the real unit of the machine.
dnXPosMarker1_p DINT	Position marker 1: x position in [increments] • C05536/2 indicates the position in the real unit of the machine.
dnYPosMarker1_p DINT	Position marker 1: y position in [increments] • C05537/2 indicates the position in the real unit of the machine.

Identifier/data type	Value/meaning
dnXPosMarker2_p DINT	Position marker 2: x position in [increments] • C05536/3 indicates the position in the real unit of the machine.
dnYPosMarker2_p DINT	Position marker 2: y position in [increments] • C05537/3 indicates the position in the real unit of the machine.
dnXPosMarker3_p DINT	Position marker 3: x position in [increments] • C05536/4 indicates the position in the real unit of the machine.
dnYPosMarker3_p DINT	Position marker 3: y position in [increments] • C05537/4 indicates the position in the real unit of the machine.
dnXPosMarker4_p DINT From library V01.02.xx.xx	Position marker 4: x position in [increments] • C05536/5 indicates the position in the real unit of the machine.
dnYPosMarker4_p DINT From library V01.02.xx.xx	Position marker 4: y position in [increments] • C05537/5 indicates the position in the real unit of the machine.
dnXPosMarker5_p DINT From library V01.02.xx.xx	Position marker 5: x position in [increments] • C05536/6 indicates the position in the real unit of the machine.
dnYPosMarker5_p DINT From library V01.02.xx.xx	Position marker 5: y position in [increments] • C05537/6 indicates the position in the real unit of the machine.
dnXPosMarker6_p DINT From library V01.02.xx.xx	Position marker 6: x position in [increments] • C05536/7 indicates the position in the real unit of the machine.
dnYPosMarker6_p DINT From library V01.02.xx.xx	Position marker 6: y position in [increments] • C05537/7 indicates the position in the real unit of the machine.
dnXPosMarker7_p DINT From library V01.02.xx.xx	Position marker 7: x position in [increments] • C05536/8 indicates the position in the real unit of the machine.
dnYPosMarker7_p DINT From library V01.02.xx.xx	Position marker 7: y position in [increments] • C05537/8 indicates the position in the real unit of the machine.
XAxisData	Machine parameters of the x axis • Via this output the machine parameters used by the FB can be easily transferred to subsequent FBs.
YAxisData	Machine parameters of the y axis • Via this output the machine parameters used by the FB can be easily transferred to subsequent FBs.
bError BOOL	"Error" status TRUE: An error has occurred (group signal). • For details see display parameter C05539.

Parameter

Parameter	Possible settings	Information
C05530/1	String of digits	Position unit of x axis • Read only
C05530/2	String of digits	Designation of the x axis • Read only
C05530/3	String of digits	Position unit of y axis • Read only
C05530/4	String of digits	Designation of the y axis • Read only

Parameter	Possible settings			Information
C05534	0		2147483647	Number of the position data track • Display of the *dnTrackNumber* input signal.
C05536/1...4	-214748.3647	Unit	214748.3647	Position marker 0 ... 3: x position • Display of the x positions in the real unit of the machine.
C05536/5...8 From library V01.02.xx.xx	-214748.3647	Unit	214748.3647	Position marker 4 ... 7: x position • Display of the x positions in the real unit of the machine.
C05537/1...4	-214748.3647	Unit	214748.3647	Position marker 0 ... 3: y position • Display of the y positions in the real unit of the machine.
C05537/5...8 From library V01.02.xx.xx	-214748.3647	Unit	214748.3647	Position marker 4 ... 7: y position • Display of the y positions in the real unit of the machine.
C05539	Error messages:			Status (bit coded) • Bits that are not itemised are reserved for future extensions. • Bit 31 is the default setting if one or more other error bits (bit 16 ... 30) are set.
	bits22	Invalid X axis data structure		
	bits23	Invalid Y axis data structure		
	bits25	Invalid data track number		
	bits31	General error		

5.5.1 Data flow

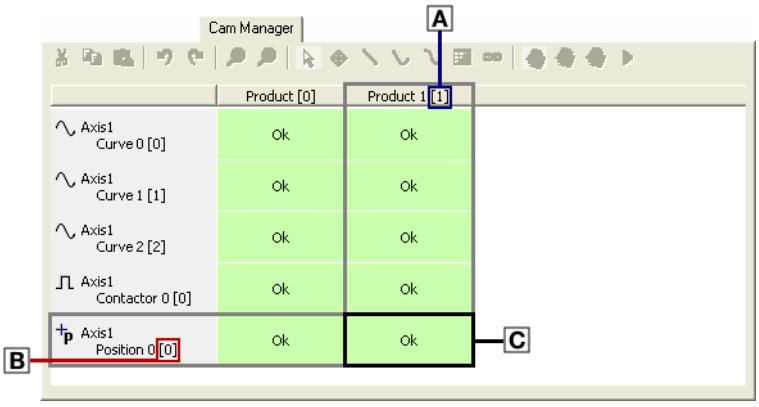
The position markers are entered in the »Cam Designer«. The transmission of the cam data to the controller(s) – together with the corresponding parameter set – is effected via the »Engineer« by executing the command **Transfer parameter set to the device**. Afterwards the parameter set and the cam data have to be saved in the memory module with mains failure protection (e. g. with the device command C00002 = "11: Save starting parameters").

Tip!

The position markers can also be entered directly via the parameters of the basic drive function "Cam data management" (**LS_CamInterface** SB) if cam data have already been downloaded to the controller.

Selection of the cam data to be used within the application

Within the application the selection of the position data to be used is then carried out via the product number and the track number which have been defined for the axis in the »Cam Manager«:



A Product number
B Track number
C Position data

- The selection of the product number A and therefore the selection of the product to be manufactured for the application is effected globally via the input *CAM_dnProductNumber* of the **LS_CamInterface** SB and has to be carried out synchronously for all cam drives in the interconnection.



- The selection of the track number B of the profile data track, however, is effected individually via the input *dnTrackNumber* of the corresponding entity of the **L_CamPosMarker** FB.



5.6 L_CamProfiler - curve interpolation for motion profiles

Function library:	LenzeCam	**FB only permissible for firmware from V3.0 onwards!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

From the cam data under the current product number (specified by the **LS_CamInterface**) SB, this FB reads the table of values of the curve track selected via the input *dnTrackNumber* and outputs the corresponding y position for the x position specified.

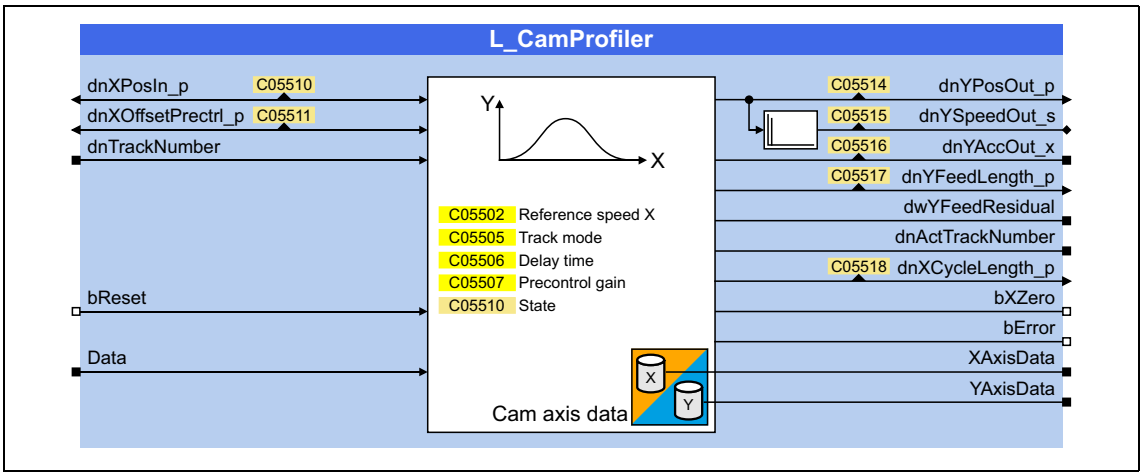
- Between two points of the table of values a linear interpolation is effected (similar to the **L_CamData** FB of the "Software Package CAM" for 9300ET).
- The y position is output in the corresponding y measuring system (modulo/limited/unlimited).
- Additionally, the y speed and the acceleration curve are output which are used for the acceleration feedforward control.
- Furthermore, for feed profiles (unlimited measuring system), the feed length for the use in a subsequent FB for stretching and compression is output.
- For the curve interpolation of a characteristic, as a reduced variant of this FB, the L_CamCurve FB is provided instead.

Stop!

Ensure that the selected curve track contains machine parameters matching the input signals, as otherwise serious malfunctions within the application may occur!

Note!

The minimum x value must be "0" to ensure that the FB can work without errors. If the x value range of a curve starts with x <> 0, an error will be output (*bError* = TRUE).



Inputs

Identifier/data type	Information/possible settings
dnXPosIn_p DINT	Input position in [increments] • Position of the x axis of an encoder via bus system. • C05510 indicates the input position in the real unit of the machine.
dnXOffsetPrectrl_p DINT	Offset for feedforward control in [increments] • Position offset for displacing the acceleration track shown at the output *dnYAccOut_x*. • C05511 indicates the position offset in the real unit of the machine.
dnTrackNumber DINT	Selection of the curve track • The acceptance is effected immediately, depending on the mode for the track switch-over selected in C05505, or in the next zero crossing of the x axis.
bReset BOOL From library V01.04.xx.xx	Reset output position FALSE↗TRUE: The *dnYPosOut_p* output is set to "0".
Data From library V01.04.xx.xx	Data interface – For Lenze service only! **For the "Electronic cam" application:** Leave the input blank. • If the *Data* input is not energised, the FB reads the table of values of the curve track selected via the *dnTrackNumber* input from the cam data under the current product number (specified by the **LS_CamInterface** SB) and outputs the corresponding y position for the x position specified. • In this case, the **LS_CamInterface** SB has full online change functionality.

Outputs

Identifier/data type	Value/meaning
dnYPosOut_p DINT	Output position in [increments] • Unlimited position of the y axis. • This output position can be directly transferred to the basic function "Position follower" (**LS_PositionFollower** SB), if an unlimited measuring system is used in the drive axis. • C05514 indicates the output position in the real unit of the machine.
dnYSpeedOut_s DINT	Output speed given as speed in [rpm] • C05515 indicates the output speed in the real unit of the machine.
dnYAccOut_x DINT	Acceleration feedforward value • The acceleration feedforward value results from the acceleration curve stored in the cam data and the mass information stored in the machine parameters for the y axis (YAxisData). • The output of the acceleration feedforward value is only effected if the motion profile selected also contains the corresponding information. • In some applications the dynamic performance can still be increased by means of an acceleration feedforward. • Output as speed change/time in [rpm/s] • Display parameter: C05516
dnYFeedLength_p DINT	Feed length of the current profile in [increments] • Only if a feed profile is available. • C05517 indicates the feed length in the real unit of the machine.
dwYFeedResidual DWORD	Residual value of the feed length of the current profile • Only if a feed profile is available.
dnActTrackNumber DINT	Number of the currently processed motion profile (current curve track number)
dnXCycleLength_p DINT From library V01.01.xx.xx	Cycle in [increments] • C05518 indicates the cycle in the real unit of the machine.

Identifier/data type	Value/meaning	
bXZero BOOL	Status signal "Zero position/zero crossing" • The input signal *dnXPosIn_p* is the reference.	
	FALSE	Master value is at any position in the master value cycle (register)
	TRUE	**Permanently:** Master value is at zero position. **For one cycle:** Zero crossing of the master value.
bError BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameter C05519.
XAxisData	Machine parameters of the x axis • Via this output the machine parameters used by the FB can be easily transferred to subsequent FBs.	
YAxisData	Machine parameters of the y axis • Via this output the machine parameters used by the FB can be easily transferred to subsequent FBs.	

Parameters

Parameter	Possible settings			Info
C05500/1	Character string			X position unit • Read only
C05500/2	Character string			X speed unit • Read only
C05500/3	Character string			X acceleration unit • Read only
C05500/4	Character string			Designation of the x axis • Read only
C05501/1	Character string			Y position unit • Read only
C05501/2	Character string			Y speed unit • Read only
C05501/3	Character string			Y acceleration unit • Read only
C05501/4	Character string			Designation of the y axis • Read only
C05502	-214748.3647	Unit/t	214748.3647	X reference speed • Initialisation: 0.0000 unit/t
C05505				Track switch-over mode
	0	In the zero crossing of the x axis		Lenze setting
	1	Immediately		
C05506	-8.000	ms	8.000	Dynamisation time • For trimming the feedforward control • Initialisation: 0.000 ms
C05507	0.00	%	200.00	Feed forward gain • Initialisation: 100.00 %
C05510	-214748.3647	Unit	214748.3647	Input position • Display of the *dnXPosIn_p* in the real unit of the machine.

Parameter	Possible settings			Info
C05511	-214748.3647	Unit	214748.3647	Feedforward control offset • Refers to the acceleration track shown at output *dnYAccOut_x*. • Display of the *dnXOffsetPrectrl_p* input signal in the real unit of the machine.
C05514	-214748.3647	Unit	214748.3647	Output position • Display of the *dnYPosOut_p* output signal in the real unit of the machine.
C05515	-214748.3647	Unit/t	214748.3647	Output speed • Display of the *dnYSpeedOut_s* output signal in the real unit of the machine.
C05516	-214748.3647	Unit/t^2	214748.3647	Acceleration feedforward control • Display of the *dnYAccOut_x* output signal in the real unit of the machine.
C05517	0.0000	Unit	214748.3647	Feed length • Display of the *dnYFeedLength_p* output signal in the real unit of the machine.
C05518 From library V01.01.xx.xx	0.0000	Unit	214748.3647	Cycle • Display of the *dnXCycleLength_p* output signal in the real unit of the machine.
C05519	Status messages:			Status (bit-coded) • Bits that are not listed are reserved for future extensions. • The *bError* error output will only be set to TRUE if an error message (Bit 16 ...Bit 31) is issued. • Bit 31 is always set if one or more of the other error bits (bit 16 ... bit 30) are set.
	bits10	Zero crossing active		
	Error messages:			
	Bit 16	Position is beyond the cycle		
	Bit 22	Invalid X axis data structure		
	bits23	Invalid Y axis data structure		
	bits25	Invalid data track number		
	Bit 26	Invalid data track type		
	Bit 31	General error		

5.6.1 Data flow

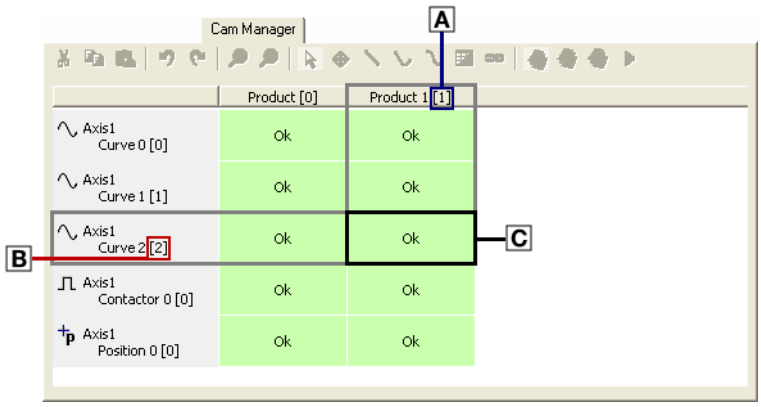
The motion profiles (curves) are entered or imported in the »Cam Designer« by means of graphical objects (point, line, polynomial). The transmission of the cam data to the controller(s) – together with the corresponding parameter set – is effected via the »Engineer« by executing the command **Transfer parameter set to the device**. Afterwards the parameter set and the cam data have to be saved in the memory module with mains failure protection (e. g. with the device command C00002 = "11: Save starting parameters").

Tip!

The motion profiles (curves) can also be entered directly via the parameters of the basic drive function "Cam data management" (**LS_CamInterface** SB) if cam data have already been downloaded to the controller.

Selection of the cam data to be used within the application

Within the application the selection of the profile data to be used is then carried out via the product number and the track number which have been defined for the axis in the »Cam Manager«:

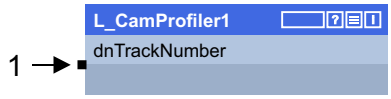


A Product number
B Track number
C Profile data

- The selection of the product number A and therefore the selection of the product to be manufactured for the application is effected globally via the input *CAM_dnProductNumber* of the **LS_CamInterface** SB and has to be carried out synchronously for all cam drives in the interconnection.



- The selection of the track number B of the motion profile, however, is effected individually via the input *dnTrackNumber* of the corresponding entity of the **L_CamProfiler** FB.



5.6.2 Track switch-over

A switch-over to another motion profile can be effected during operation.

- In the Lenze setting the acceptance of the new cam data is carried out in the next zero crossing of the x axis.
- Alternatively, the immediate acceptance of the new cam data can be selected in C05505.

5.6.3 Feedforward control

Via the output *dnYAccOut_x* the FB provides an acceleration feedforward value, if an acceleration curve for the motion profile selected is stored in the cam data, and corresponding mass information is stored in the machine parameters for the y axis.

- Optionally a dynamisation time can be set in C05506, by which the x position for the calculation of the acceleration signal is trimmed.
- Additionally a user-definable position offset can be specified via the input *dnXOffsetPrectrl_p*, which also displaces the acceleration track.
- Via the feedforward gain that can be set in C05507, additionally a systematic adjustment of the feedforward control can be carried out.

5.6.4 Output of the machine parameters

Via the outputs *XAxisData* and *YAxisData* the machine parameters of the x and y axis are provided for the optional further use in other FBs.

- For purposes of information, the designation of the x axis is displayed in C05500/4 and the designation of the y axis is shown in C05501/4.



Tip!

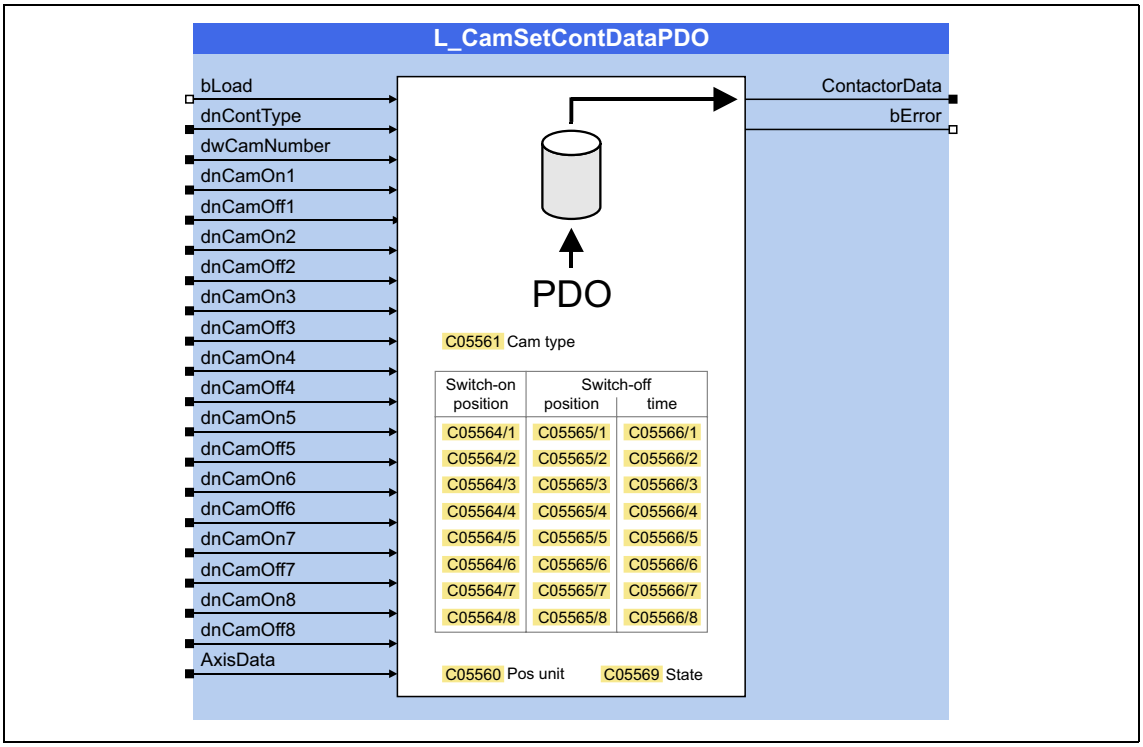
In more complex interconnections, for purposes of clarity, it is recommended to read out the machine parameters specifically via an instance of the L_CamGetAxisData FB instead of using the outputs *XAxisData* and *YAxisData*.

5.7 L_CamSetContDataPDO - contactor data selection via inputs/PDOs

Function library:	LenzeCam	**FB is available as of library V01.02.xx.xx! FB may only be used for firmware as of V3.0!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

This FB serves to select contactor data independent of the cam technology (CamDesigner/LC1 file) directly via inputs.

- The inputs can be connected, for instance, to a corresponding port block to accept contactor data from a master control via process data objects (PDOs).
- The read contactor data are transferred to the FB L_CamContactor via the *ContactorData* output.
- The read contactor data are saved with mains failure protection and thus will be available even after mains switching.
- The FB L_CamSetContDataSDO serves to select the contactor data via codes/SDOs.



Inputs

Identifier/data type	Information/possible settings	
bLoad BOOL	Accept contactor data	
	TRUE	The contactor data pending at the inputs are transferred via the *ContactorData* output to the FB L_CamContactor. • Changing the inputs has an immediate effect on the cams.
	TRUE↘FALSE	The contactor data valid last are saved internally with mains failure protection and transferred via the *ContactorData* output to the FB L_CamContactor. • Now, input changes do not affect the cams anymore. They remain unchanged.

Identifier/data type	Information/possible settings	
dnContType DINT	Cam type	
	1	Positive travel cams
	2	Negative travel cams
	3	Bidirectional travel cams
	11	Positive time cams
	12	Negative time cams
dwCamNumber DWORD	Number of cams in the track	
dnCamOn1 ... dnCamOn8 DINT	Cam switch-on position in [increments] • C05564/1...8 indicates the cam switch-on position in the real unit of the machine.	
dnCamOff1 ... dnCamOff8 DINT	With travel cams: Cam switch-off position in [increments] • C05565/1...8 indicates the cam switch-off position in the real unit of the machine.	
	With time cams: Cam break time in [s] • "e3" representation (fixed point with three decimal positions): "1" ≡ 1 ms, "1000" ≡ 1 s • Display parameter: C05566/1...8	
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	

Outputs

Identifier/data type	Value/meaning	
ContactorData	Output of the contactor data • Connect this output with the input of the same name of the FB L_CamContactor.	
bError BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameter C05569.

Parameter

Parameter	Possible settings			Information
C05560	String of digits			Position unit • Read only
C05561				Cam type • Read only
	1	Positive travel cams		
	2	Negative travel cams		
	3	Bidirectional travel cams		
	11	Positive time cams		
	12	Negative time cams		
C05564/1...8	-214748.3647	Unit	214748.3647	Cam switch-on position 1 ... 8 • Display of the input signal *dnCamOn1...8* in the real unit of the machine.

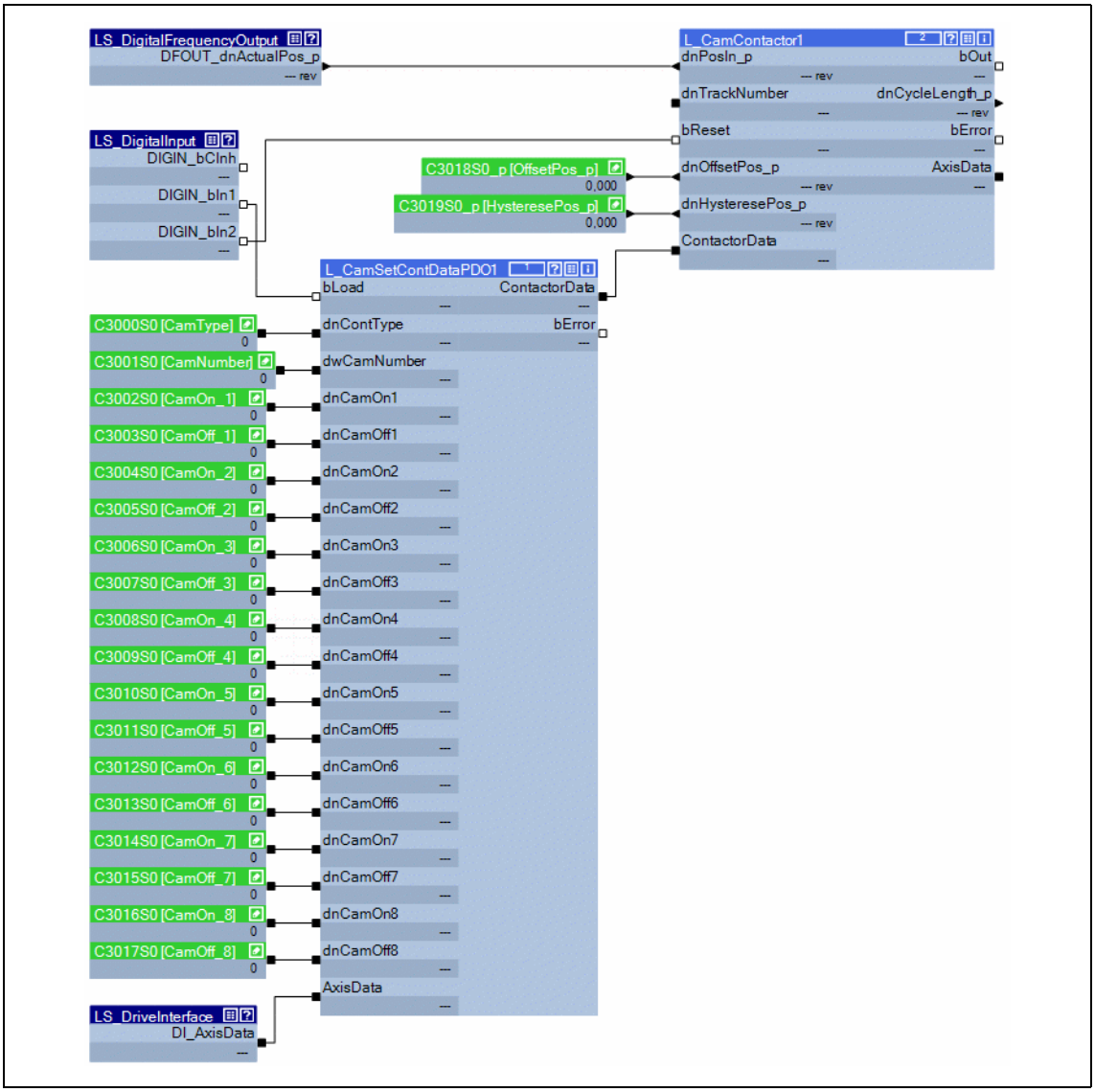
Parameter	Possible settings			Information
C05565/1...8	-214748.3647	Unit	214748.3647	Cam switch-off position 1 ... 8 • Only with travel cams • Display of the input signal *dnCamOff1...8* in the real unit of the machine.
C05566/1...8	0.000	s	134217.728	Cam break time 1 ... 8 • Only with time cams • Display of the input signal *dnCamOff1...8*.
C05567/1...8				Cam status 1 ... 8 • Read only
	0	Undefined cam		
	1	Cam is not active		
	2	Cam is active		
C05569	Status messages:			Status (bit coded) • Bits that are not itemised are reserved for future extensions. • The *bError* error output will only be set to TRUE if an error message (bit 16 ... 31) is issued. • Bit 31 is the default setting if one or more other error bits (bit 16 ... 30) are set.
	bits7	Invalid cam number		
	Error messages:			
	bits23	Invalid axis data structure		
	bits27	Invalid cam type		
	bits28	Invalid contactor data (wrong parameterisation)		
	bits31	General error		

5.7.1 Exception handling

The following table contains the behaviour of the FB in various special cases:

Special case	Behaviour
Invalid axis data	The *bError* output is set to TRUE.
Wrong parameterisation of the contactor data	A cam track is output without the wrong cams.
Selection of an invalid cam number	The number of cams is limited to 8, i.e. a cam track with max. 8 cams is output. In this case, the *bError* output is not set to TRUE.
Selection of an invalid cam type	No cam track is output. The *bError* output is set to TRUE.

5.7.2 Interconnection example



[5-15] Interconnection example: Selection of the cam data via PDOs

5.8 L_CamSetContDataSDO - contactor data selection via codes/SDOs

Function library:	LenzeCam	**FB is available as of library V01.02.xx.xx! FB may only be used for firmware as of V3.0!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

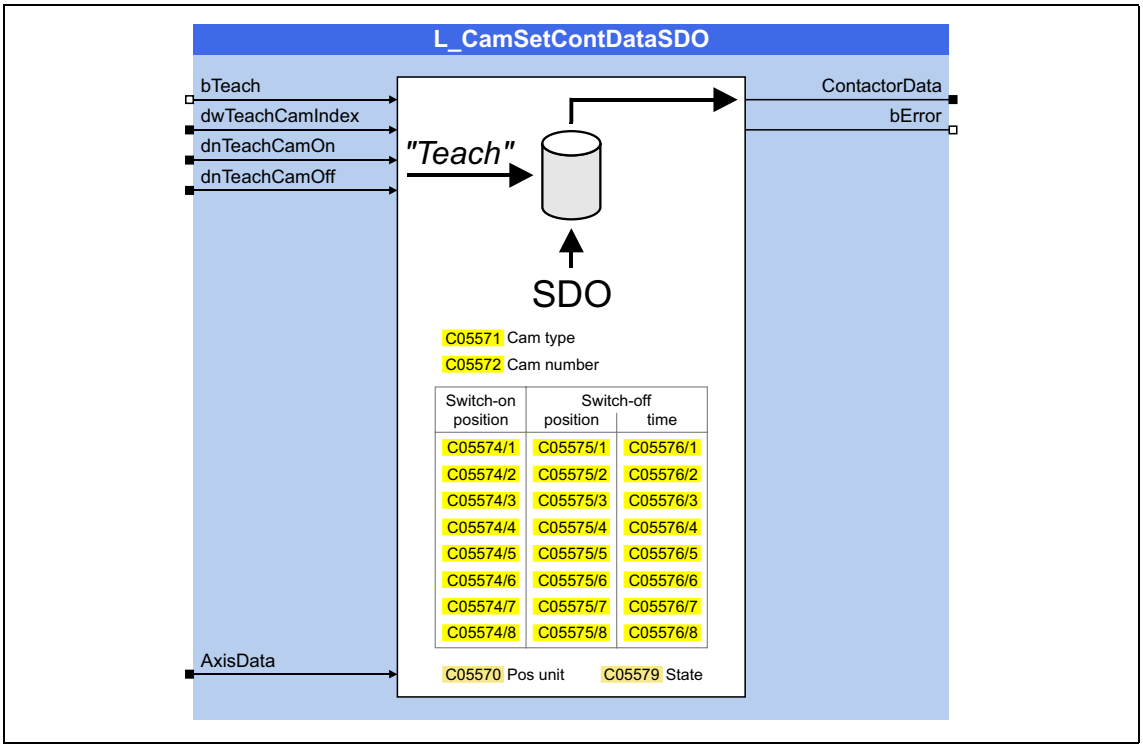
This FB serves to select contactor data independent of the cam technology (CamDesigner/LC1 file) directly via codes.

- The codes can be written, for instance, via parameter data objects (SDOs) by a master control.
- The FB supports "Teaching" of cams.
- The contactor data are transferred to the FB L_CamContactor via the *ContactorData* output.
- The FB L_CamSetContDataPDO serves to select the contactor data via inputs/PDOs.

Note!

After changing the contactor data during operation:

Save the parameter set and the cam data with mains failure protection in the memory module (e.g. using the device command C00002 = "11: Save start parameter").



Inputs

Identifier/data type	Information/possible settings	
bTeach BOOL	"Teach" cam	
	FALSE	"Teach" function is deactivated.
	TRUE	The cam parameters pending at the inputs *dnTeachCamOn* and *dnTeachCamOff* are accepted for the cam with the index *dwTeachCamIndex*.
dwTeachCamIndex DWORD	Selection of the cam to be "taught" • Index 0 ... 7 ≡ cam 1 ... 8	
dnTeachCamOn DINT	Cam switch-on position to be "taught" in [increments]	
dnTeachCamOff DINT	With travel cams: Cam switch-off position to be "taught" in [increments]	
	With time cams: Cam break time to be "taught" in [s] • "e3" representation (fixed point with three decimal positions): "1" ≡ 1 ms, "1000" ≡ 1 s	
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	

Outputs

Identifier/data type	Value/meaning	
ContactorData	Output of the contactor data • Connect this output with the input of the same name of the FB L_CamContactor.	
bError BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameter C05579.

Parameter

Parameter	Possible settings			Information
C05570	String of digits			Position unit • Read only
C05571				Cam type
	1	Positive travel cams		
	2	Negative travel cams		
	3	Bidirectional travel cams		
	11	Positive time cams		
	12	Negative time cams		
C05572	0		8	Number of cams
C05574/1...8	-214748.3647	Unit	214748.3647	Cam switch-on position 1 ... 8
C05575/1...8	-214748.3647	Unit	214748.3647	Cam switch-off position 1 ... 8 • Only with travel cams
C05576/1...8	0.000	s	134217.728	Cam break time 1 ... 8 • Only with time cams

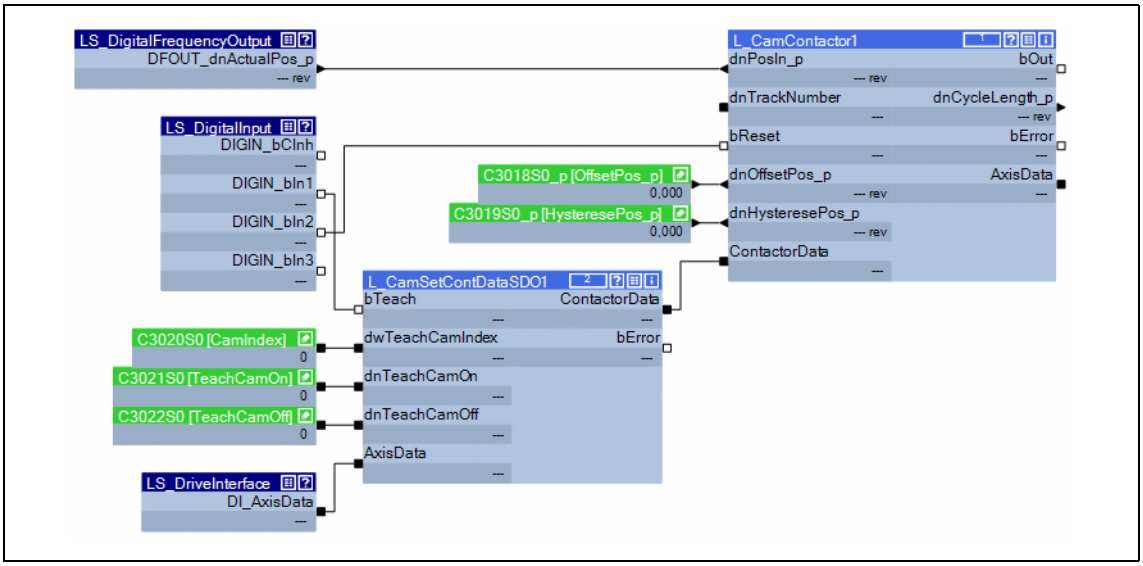
Parameter	Possible settings		Information
C05577/1...8			Cam status 1 ... 8 Read only
	0	Undefined cam	
	1	Cam is not active	
	2	Cam is active	
C05579	Status messages:		Status (bit coded) • Bits that are not itemised are reserved for future extensions. • The *bError* error output will only be set to TRUE if an error message (bit 16 ... 31) is issued. • Bit 31 is the default setting if one or more other error bits (bit 16 ... 30) are set.
	-	-	
	Error messages:		
	bits23	Invalid axis data structure	
	bits27	Invalid cam type	
	bits28	Invalid contactor data (wrong parameterisation)	
	bits31	General error	

5.8.1 Exception handling

The following table contains the behaviour of the FB in various special cases:

Special case	Behaviour
Invalid axis data	The *bError* output is set to TRUE.
Wrong parameterisation of the contactor data	A cam track is output without the wrong cams.
Error while teaching in	The cam is not taught in.
Changing the cam break time with active cam in the FB L_CamContactor	A value change for the cam break time in C05576/1...8 is only accepted after the break time set before in the FB L_CamContactor has elapsed.

5.8.2 Interconnection example



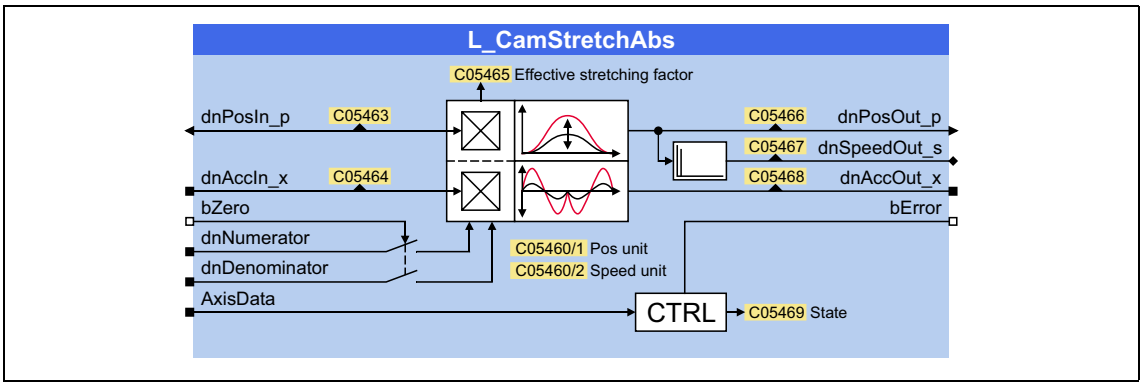
[5-16] Interconnection example: Selection of the contactor data via SDOs

5.9 L_CamStretchAbs - stretching/compressing the position (absolute)

Function library:	LenzeCam	**FB permissible for firmware as of V3.0 only!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

This FB is used for simple absolute stretching and compression of a position in the limited traversing range.

- Via the inputs *dnNumerator* and *dnDenominator* a free stretch factor for the position transmission can be defined. A changed stretch factor is only accepted internally with a TRUE signal at the input *bZero*.
- At the same time, the FB can stretch/compress the acceleration feedforward value, which the L_CamProfiler FB provides to downstream FBs.



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Input position in [increments] • C05463 indicates the input position in the real unit of the machine.
dnAccIn_x DINT	Acceleration feedforward value • For accepting the acceleration feedforward value for the y axis, given by the L_CamProfiler FB, just connect this input with its output *dnYAccOut_x*. • Selection as speed change/time in [rpm/s] • Display parameter: C05464
bZero BOOL	Accept changed stretch factor • Apply an acceptance signal to this input, which for instance signalises the zero crossing of the x axis (synchronised stretching/compression).
	TRUE: The changed stretch factor is accepted internally. • At the same time the status bit 10 ("Zero crossing active") is set in C05469.
dnNumerator DINT	Stretch factor (numerator) • $\pm 2^{31}-1$
dnDenominator DINT	Stretch factor (denominator) • 1 ... $2^{31}-1$ • A negative stretch factor can only be defined via *dnNumerator*. • In the case of an input value < 1, processing is continued internally with 1.
AxisData	Machine parameters • If the FB is to relate to the measuring system of the drive axis, connect this input to the output *DI_AxisData* of the **LS_DriveInterface** SB for accepting the machine parameters of the drive/motor. • If the FB within a cam application is to relate to the measuring system of the master selection (master), connect this input to the output *XAxisData* of the L_CamProfiler FB instead. (📖 162)

Outputs

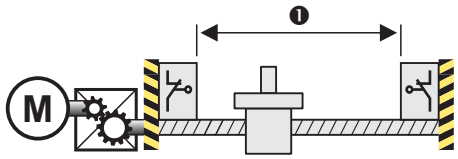
Identifier/data type	Value/meaning	
dnPosOut_p DINT	Output position in [increments] • Limitation can be effected (display via *bLimit*) • C05466 indicates the output position in the real unit of the machine.	
dnSpeedOut_s DINT	Output speed given as speed in [rpm] • C05467 indicates the output speed in the real unit of the machine.	
dnAccOut_x DINT	Acceleration feedforward value • Output as speed change/time in [rpm/s] • Display parameter: C05468	
bError BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameter C05469.

Parameter

Parameter	Possible settings			Information
C05460/1	String of digits			Position unit • Read only
C05460/2	String of digits			Speed unit • Read only
C05463	-214748.3647	Unit	214748.3647	Input position • Display of the *dnPosIn_p* input signal in the real unit of the machine.
C05464	-214748.3647	Unit/t^2	214748.3647	Input acceleration feedforward control • Display of the *dnAccIn_x* input signal in the real unit of the machine.
C05465	-214748.3647		214748.3647	Effective stretch factor
C05466	-214748.3647	Unit	214748.3647	Output position • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C05467	-214748.3647	Unit/t	214748.3647	Output speed • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.
C05468	-214748.3647	Unit/t^2	214748.3647	Output acceleration feedforward control • Display of the *dnAccOut_x* output signal in the real unit of the machine.
C05469	Status messages:			Status (bit coded) • Bits that are not itemised are reserved for future extensions. • The *bError* error output will only be set to TRUE if an error message (bit 16 ... 31) is issued. • Bit 31 is the default setting if one or more other error bits (bit 16 ... 30) are set.
	bits10	Zero crossing active		
	Error messages:			
	bits17	Initial speed too high		
	bits23	Invalid axis data structure		
	bits24	Stretched position too great		
	bits31	General error		

5.9.1 Typical application

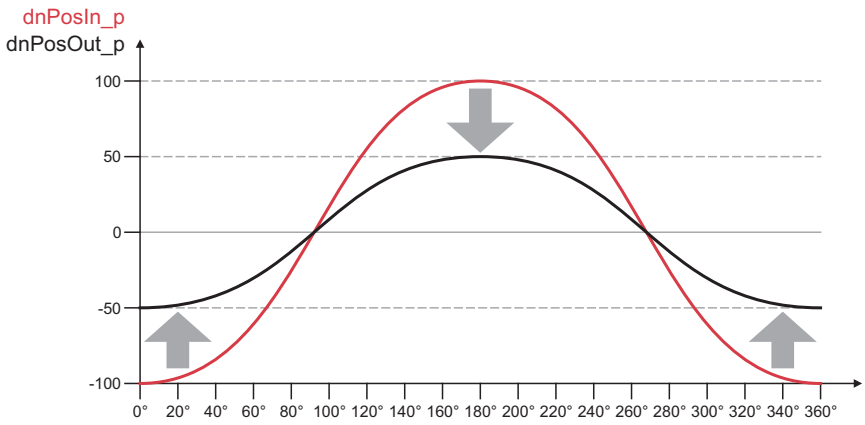
By means of this FB a traversing path within the limited traversing range can be simply changed via factors. By this, the motion profile can be maintained.



❶ Limited traversing range

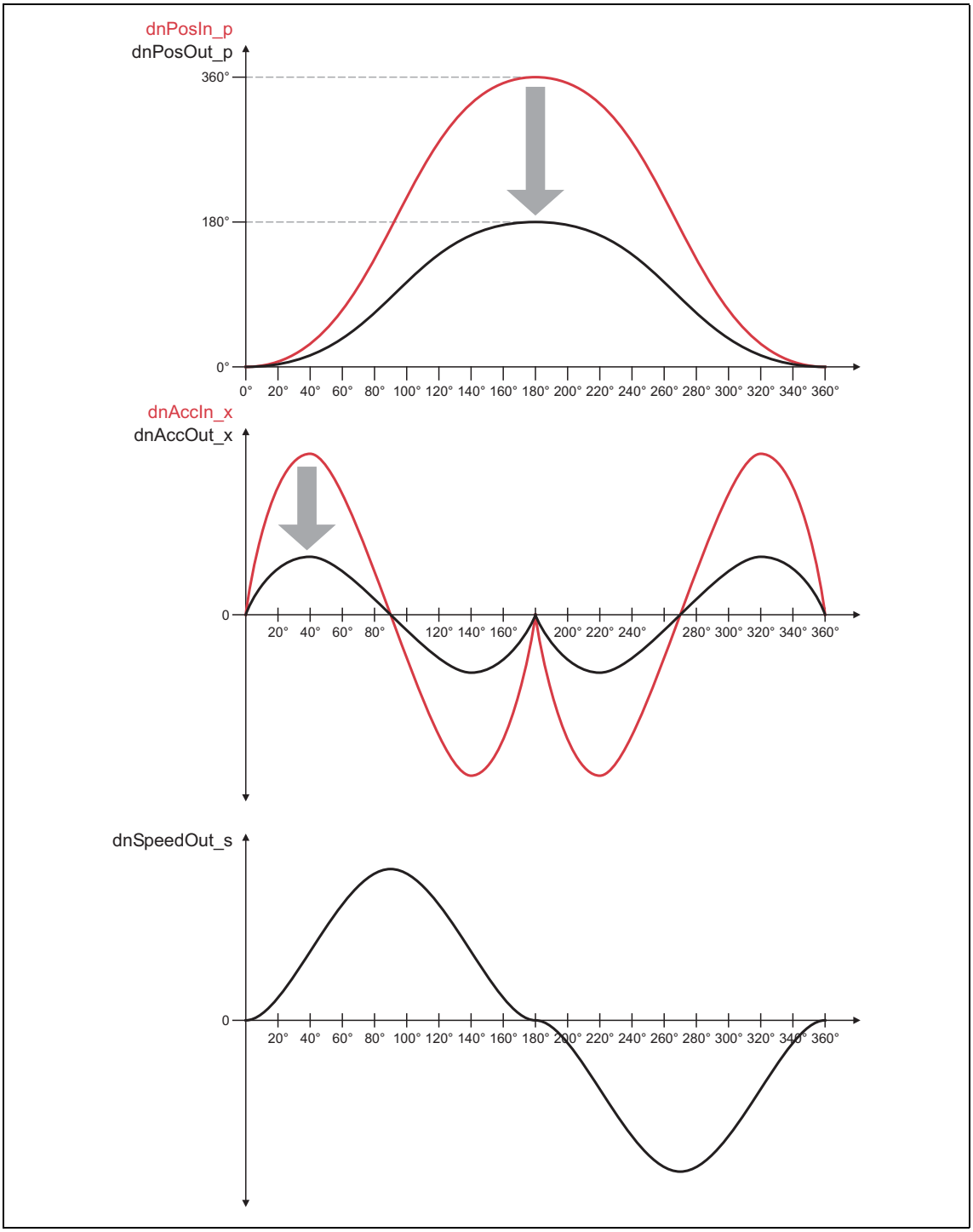
[5-17] Example: spindle drive

- Thereby the factors always relate to the zero point of the motion profile on the y axis.



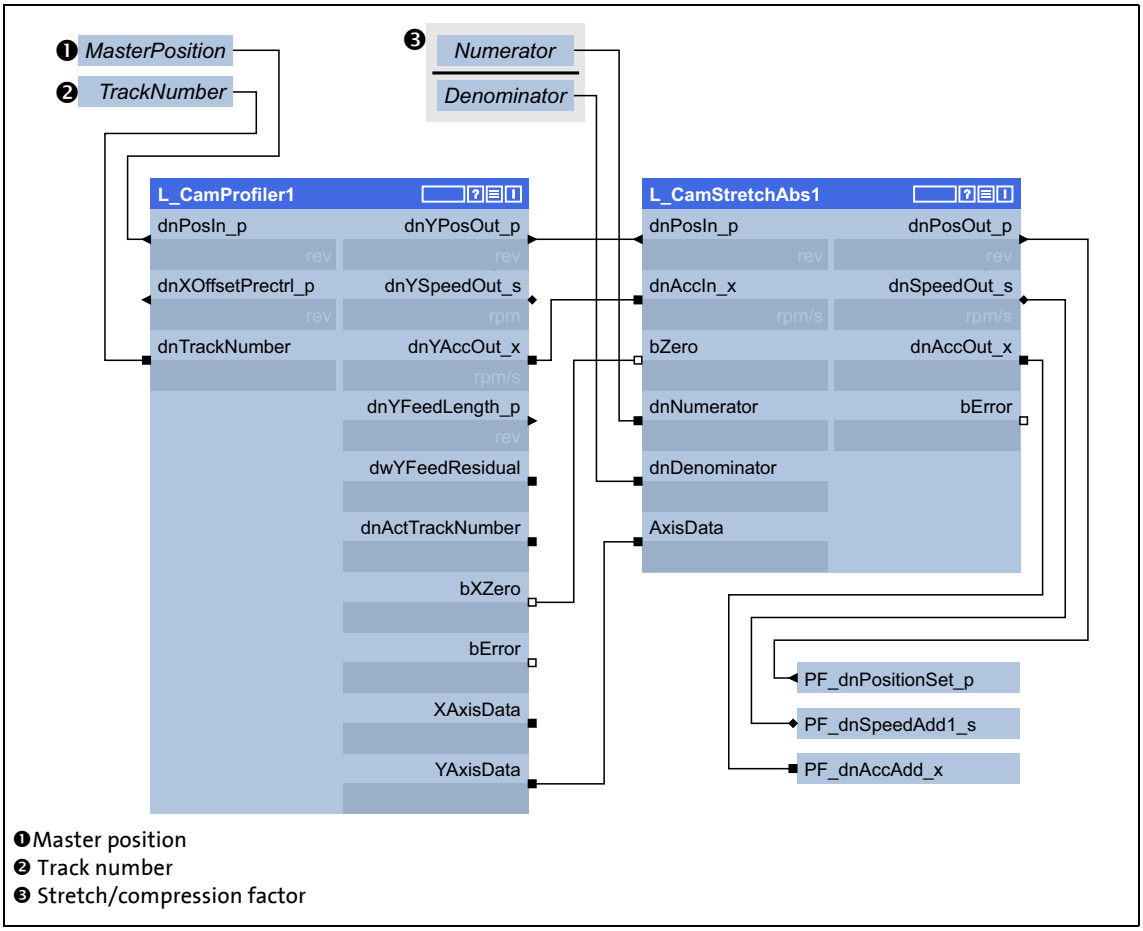
[5-18] Example: Decrease of the lift to 50 % with stretch factor = 0.5

5.9.2 Signal characteristics



[5-19] Example: Compression of the position with stretch factor = 0.5

5.9.3 Interconnection example



[5-20] Interconnection example

Note!

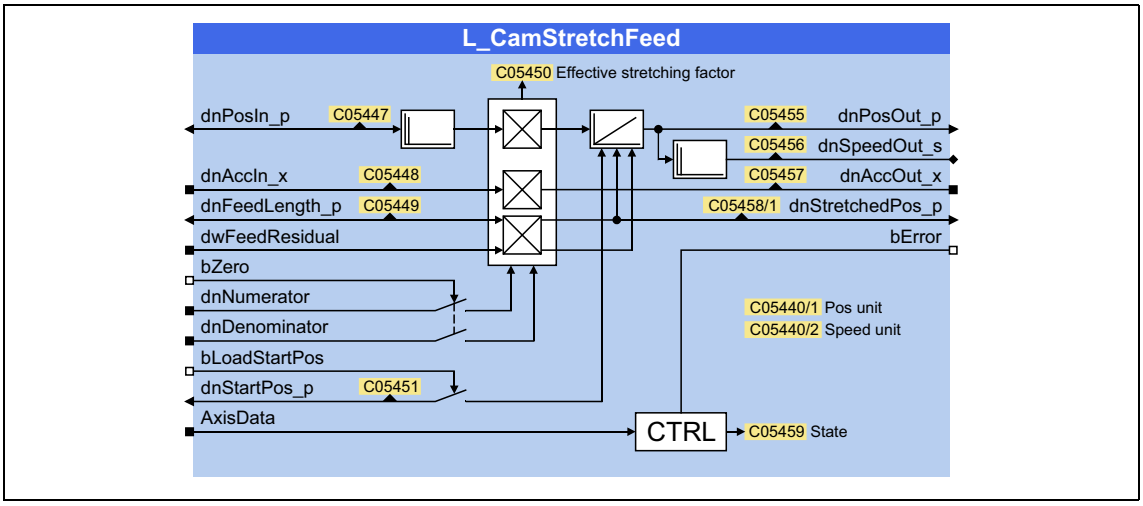
If the stretch/compression factor is to be altered online, it may only be changed over in the zero position of the motion profile, or it has to be directed via a profile generator.

5.10 L_CamStretchFeed - stretching/compressing the position (relative)

Function library:	LenzeCam	**FB permissible for firmware as of V3.0 only!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

This FB is used for relative stretching and compression of a position in the "modulo" or "unlimited" traversing range. For this the FB differentiates the input position to a position difference and stretches or compresses it. Afterwards the position difference stretched/compressed is integrated to a continuous position for the "unlimited" traversing range, or cycle-related for the "modulo" traversing range.

- The output continuous position can be directly transferred to the basic function "position follower" (**LS_PositionFollower** SB).
- Via the inputs *dnNumerator* and *dnDenominator* a free stretch factor for the speed transmission can be defined. A changed stretch factor is only immediately accepted internally with a TRUE signal at the input *bZero*.
- Via the input *bLoadStartPos* the position output can be set to the starting position pending at the input *dnStartPos_p*.



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Input position in [increments] • C05447 indicates the input position in the real unit of the machine.
dnAccIn_x DINT	Acceleration feedforward value • For accepting the acceleration feedforward value for the y axis, given by the L_CamProfiler FB, just connect this input with its output *dnYAccOut_x*. • Selection as speed change/time in [rpm/s] • Display parameter: C05448
dnFeedLength_p DINT	Feed length of the current profile in [increments] • For accepting the feed length for the y axis, output by the L_CamProfiler FB, you simply connect this input to its output *dnYFeedLength_p*. • Display parameter: C05449
dwFeedResidual DWORD	Residual value of the feed length of the current profile • For accepting the residual value for the y axis, output by the L_CamProfiler FB, you simply connect this input to its output *dwYFeedResidual*.

Identifier/data type	Information/possible settings
bZero BOOL	Accept changed stretch factor • Apply an acceptance signal to this input, which for instance signalises the zero crossing of the x axis (synchronised stretching/compression).
	TRUE: The changed stretch factor is immediately accepted internally.
dnNumerator DINT	Stretch factor (numerator) • $\pm 2^{31}$-1
dnDenominator DINT	Stretch factor (denominator) • 1 ... 2^{31}-1 • A negative stretch factor can only be defined via *dnNumerator*. • In the case of an input value < 1, processing is continued internally with 1.
bLoadStartPos BOOL	Load the integrator with starting position
	TRUE: Load the integrator with the value at the *dnStartPos_p* input.
dnStartPos_p DINT	Starting position in [inc] with which the integrator is loaded by setting *bLoadStartPos* to TRUE. • C05451 indicates the starting position in the real unit of the machine.
AxisData	Machine parameters • If the FB is to relate to the measuring system of the drive axis, connect this input to the output *DI_AxisData* of the **LS_DriveInterface** SB for accepting the machine parameters of the drive/motor. • If the FB within a cam application is to relate to the measuring system of the master selection (master), connect this input to the output *XAxisData* of the L_CamProfiler FB instead. (📖 162)

Outputs

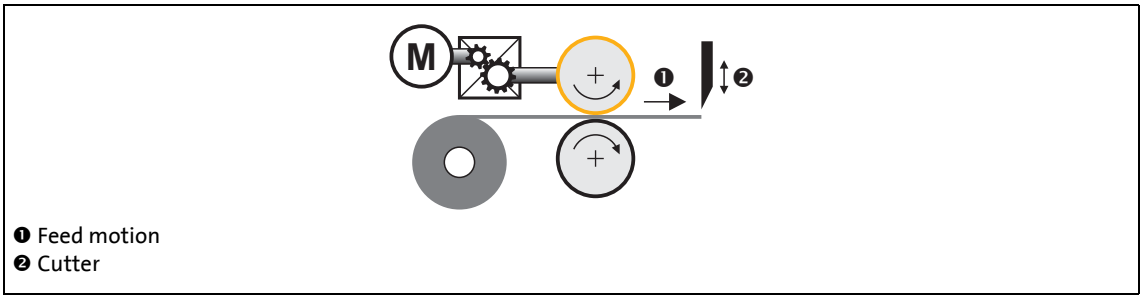
Identifier/data type	Value/meaning
dnPosOut_p DINT	Output position in [increments] • C05455 indicates the output position in the real unit of the machine.
dnSpeedOut_s DINT	Output speed given as speed in [rpm] • C05456 indicates the output speed in the real unit of the machine.
dnAccOut_x DINT	Acceleration feedforward value • Output as speed change/time in [rpm/s] • Display parameter: C05457
dnStretchedPos_p DINT	Output of the stretched/compressed input position • For example for position comparators. • The position is calculated from the stretch factors valid within the cycle multiplied with the input position, i. e. relating to the stretch factors at the input side, the output can only change in the next cycle. The loading of a starting position has no effect on the output. • C05458/1 indicates the stretched/compressed position in the real unit of the machine.
bError BOOL	"Error" status
	TRUE: An error has occurred (group signal). • For details see display parameters C05459.

Parameter

Parameter	Possible settings			Information
C05440/1	String of digits			Position unit • Read only
C05440/2	String of digits			Speed unit • Read only
C05447	-214748.3647	Unit	214748.3647	Input position • Display of the *dnPosIn_p* input signal in the real unit of the machine.
C05448	-214748.3647	Unit/t^2	214748.3647	Input acceleration feedforward control • Display of the *dnAccIn_x* input signal in the real unit of the machine.
C05449	-214748.3647	Unit	214748.3647	Feed length • Display of the *dnFeedLength_p* input signal in the real unit of the machine.
C05450	-214748.3647		214748.3647	Effective stretch factor
C05451	-214748.3647	Unit	214748.3647	Starting position • Display of the *dnStartPos_p* input signal in the real unit of the machine.
C05455	-214748.3647	Unit	214748.3647	Output position • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C05456	-214748.3647	Unit/t	214748.3647	Output speed • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.
C05457	-214748.3647	Unit/t^2	214748.3647	Output acceleration feedforward control • Display of the *dnAccOut_x* output signal in the real unit of the machine.
C05458/1	-214748.3647	Unit	214748.3647	Stretched position • Display of the *dnStretchPos_p* output signal in the real unit of the machine.
C05458/2	-214748.3647	Unit	214748.3647	Stretched feed length • Read only
C05459	Status messages:			Status (bit coded) • Bits that are not itemised are reserved for future extensions. • The *bError* error output will only be set to TRUE if an error message (bit 16 ... 31) is issued. • Bit 31 is the default setting if one or more other error bits (bit 16 ... 30) are set.
	bit 10	Zero crossing active		
	Error messages:			
	bit 16	*dnStartPos_p* is beyond the cycle		
	bit 17	Initial speed too high		
	bit 23	Invalid axis data structure		
	bit 24	Stretched position too great		
	bit 31	General error		

5.10.1 Typical application

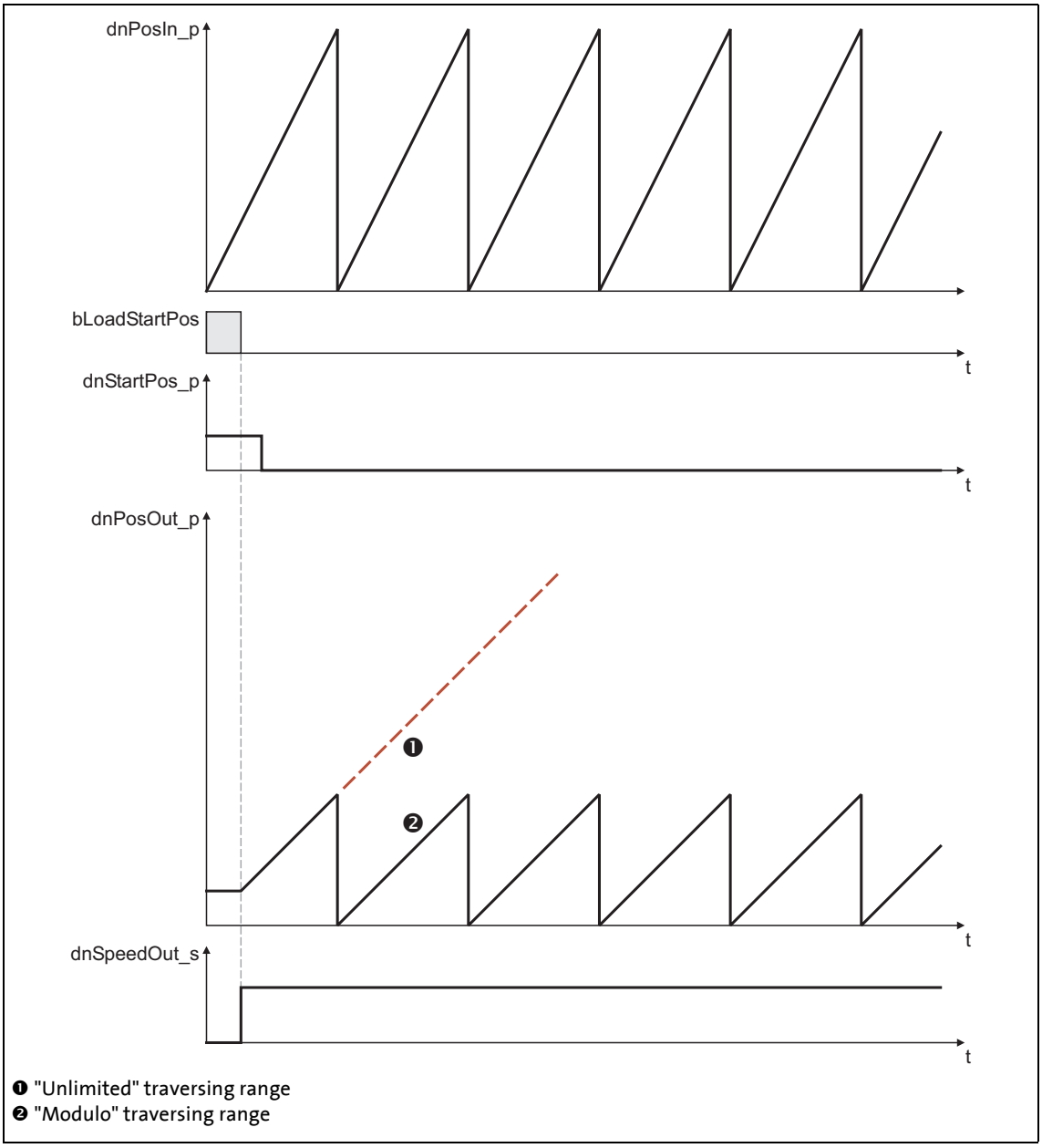
By means of this FB a feed motion with regard to the feed length at the **L_CamProfiler** FB can be simply altered via factors. Like this, the motion profile can be maintained.



❶ Feed motion
❷ Cutter

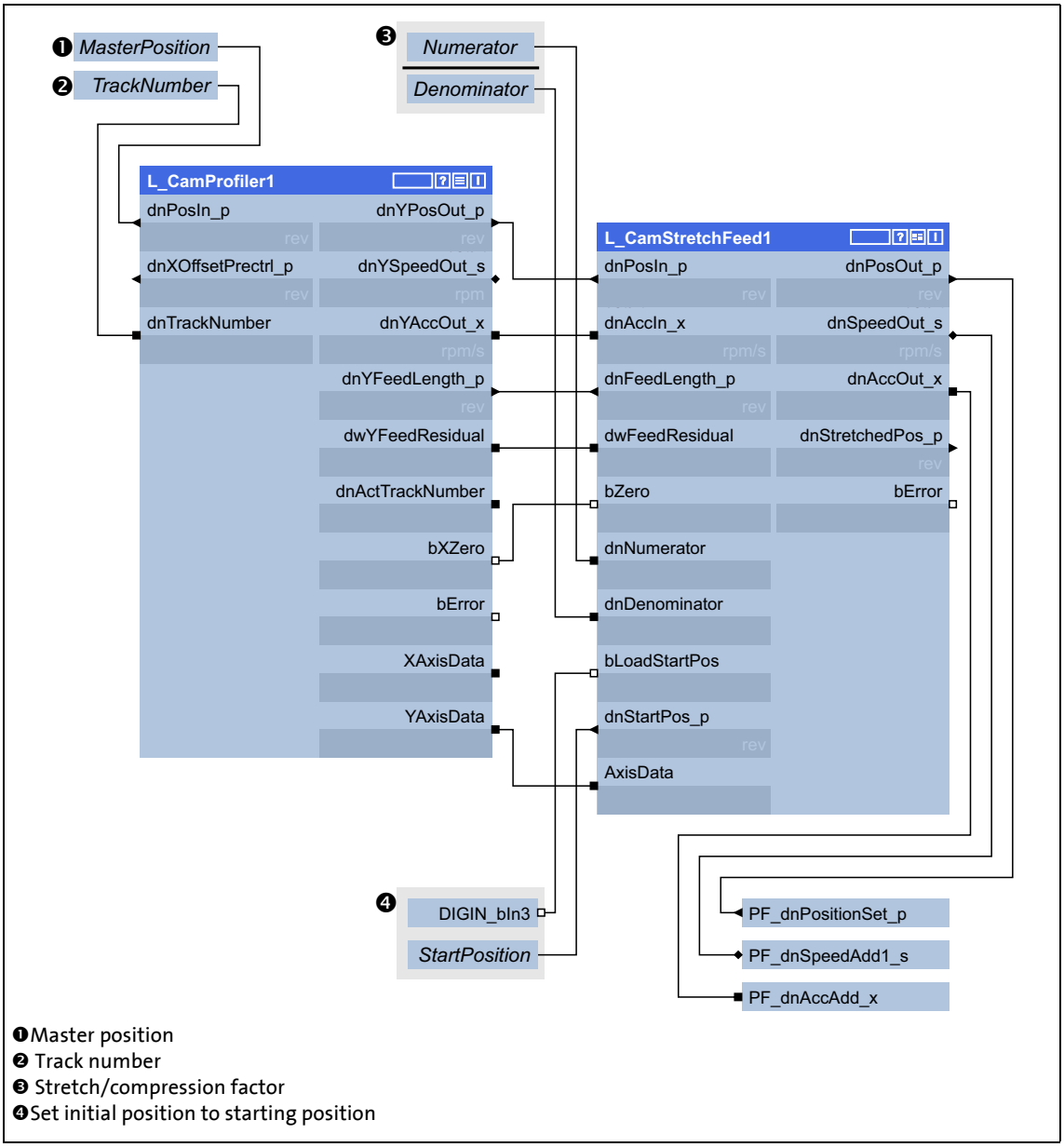
[5-21] Example: Material feed with subsequent cut

5.10.2 Signal characteristics



[5-22] Signal characteristics

5.10.3 Interconnection example



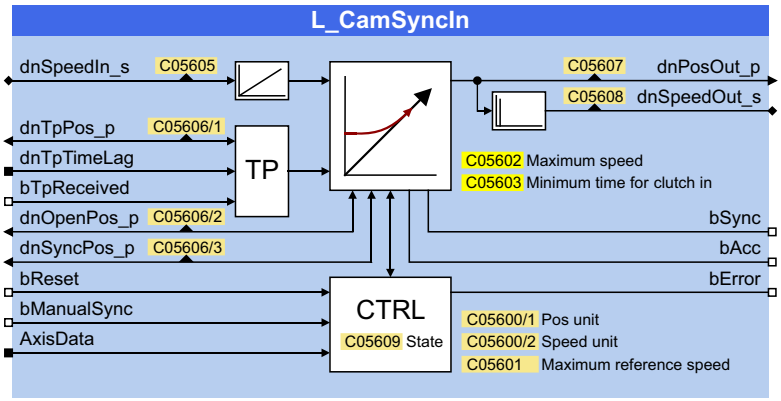
[5-23] Interconnection example

5.11 L_CamSyncIn - synchronous/oversynchronous clutch-in

Function library:	LenzeCam	**FB is available as of library V01.01.xx.xx! FB may only be used for firmware as of V3.0!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

This FB serves to synchronise a tool (drive) to a running material (relative master speed).

- Types of synchronisation:
  - Mark-controlled (touch probe)
  - Manually controlled (e.g. manually controlled top cut with flying saw)
- The synchronisation is path-based and functionally corresponds to a subset of the L_EsClutchPos clutch module. Synchronous or oversynchronous synchronisations are possible.
- The function corresponds to the "DFRFG" function of the 9300EK servo inverter and additionally supports a path-controlled clutch-in.



Inputs

Identifier/data type	Information/possible settings	
dnSpeedIn_s DINT	Input speed (master speed) as speed in [rpm] • Example: Material speed of the web with flying saw. • C05605 indicates the input speed in the real unit of the machine.	
dnTpPos_p DINT	Position of the touch probe sensor in [increments]. • C05606/1 indicates the position in the real unit of the machine.	
dnTpTimeLag DINT	Input for accepting the time stamp • Connect this input with the output *dnTouchProbeTimeLag* of the corresponding touch probe system block.	
bTpReceived BOOL	Input for taking over the status "Touch probe detected" • Connect this input with the output *dnTouchProbeReceived* of the corresponding touch probe system block.	
	FALSE↗TRUE	Touch probe detected. **Note:** At the same time, a speed must be applied at the *dnSpeedIn_s* input for the clutch-in ramp to be calculated. Otherwise, the clutch-in process will not be started in the FB.
dnOpenPos_p DINT	Clutch-in position in [increments] • Starting position for synchronising. • Is generally also used as target position for the **LS_Positioner** system block for re-positioning. • C05606/2 indicates the clutch-in position in the real unit of the machine.	

Identifier/data type	Information/possible settings	
dnSyncPos_p DINT	Synchronous position in [increments] • Starting position of the synchronous range. • C05606/3 indicates the synchronous position in the real unit of the machine.	
bReset BOOL	Reset function	
	TRUE	Function is reset to evaluate a new touch probe or the "Manual clutch-in" via *bManualSync* request.
bManualSync BOOL	Manual clutch-in • Example: Manually controlled top cut with flying saw.	
	TRUE	Start clutch-in process immediately. **Note:** At the same time, a speed must be applied at the *dnSpeedIn_s* input for the clutch-in ramp to be calculated. Otherwise, the clutch-in process will not be started in the FB.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**.	

Outputs

Identifier/data type	Value/meaning	
dnPosOut_p DINT	Output position in [increments] • C05607 indicates the output position in the real unit of the machine.	
dnSpeedOut_s DINT	Output speed given as speed in [rpm] • C05608 indicates the output speed in the real unit of the machine.	
bSync BOOL	"Clutched-in" status	
	TRUE	Input and output run synchronously.
bAcc BOOL	"Acceleration phase active" status	
	TRUE	Acceleration phase is active.
bError BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameter C05609.

Parameter

Parameter	Possible settings			Information
C05600/1	String of digits			Position unit • Read only
C05600/2	String of digits			Speed unit • Read only
C05601	0.0000	Unit/t	214748.3647	Reference speed • Read only
C05602	0.0000	Unit/t	214748.3647	Maximum speed • For oversynchronous clutch-in. • Initialisation: 0.0000 unit/t
C05603	0.001	s	100.000	Minimum clutch-in time • Time for the clutch-in process which must not be undershot. • Initialisation: 1.000 s
C05605	-214748.3647	Unit/t	214748.3647	Input speed • Display of the *dnSpeedIn_s* input signal in the real unit of the machine.

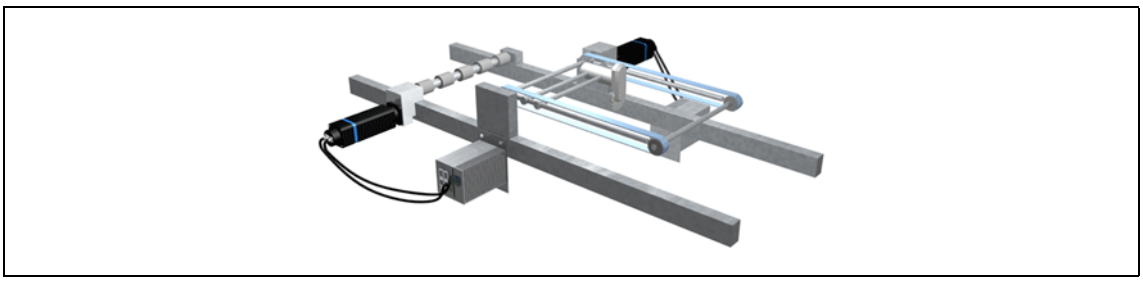
Parameter	Possible settings			Information
C05606/1	-214748.3647	Unit	214748.3647	Sensor position • Display of the *dnTpPos_p* input signal in the real unit of the machine.
C05606/2	-214748.3647	Unit	214748.3647	Clutch-in position • Display of the *dnOpenPos_p* input signal in the real unit of the machine.
C05606/3	-214748.3647	Unit	214748.3647	Synchronous position • Display of the *dnSyncPos_p* input signal in the real unit of the machine.
C05607	-214748.3647	Unit	214748.3647	Output position • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C05608	-214748.3647	Unit/t	214748.3647	Output speed • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.
C05609	Error messages:			Status (bit coded) • Bits that are not itemised are reserved for future extensions. • Bit 31 is the default setting if one or more other error bits (bit 16 ... 30) are set.
	bits18	Clutch-in not possible		
	Bit 19	Resulting acceleration too high		
	bits23	Invalid axis data structure		
	bits31	General error		

5.11.1 Application example: "Flying saw"

When a flying saw is used, the tool moves in synchronism with the moving material during the machining process. For this purpose, the tool must first be moved on a slide in parallel to the material.

To create the feed of the tool during the machining process, a second drive moves it with constant speed at 90° to the material. At the end of the machining process, the drive must be repositioned to its initial position.

A third drive from the tool drive group moves the tool, e.g., by means of the basic function "Positioning" (**LS_Positioner** system block).



[5-24] Example "flying saw"

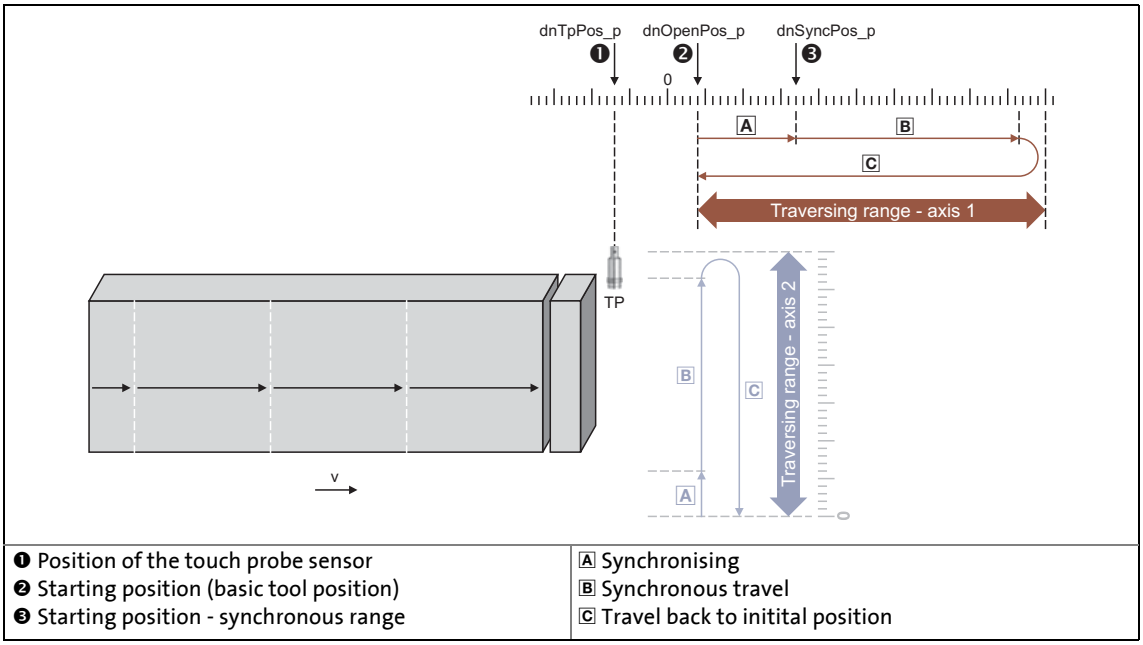
Motion sequence

The motion sequence of the flying saw drive consists of the following phases:

A. **Synchronising:** First the tool must be accelerated from the basic position to the material speed. When the material speed has been reached, the tool must be in the cutting position. The exact position of the points of intersection on the web is detected by a touch probe sensor.

B. **Synchronous travel:** The tool immerses into the material, then moves in synchronism with the material speed until the machining process is terminated including the lifting of the tool.

C. **Travel back to initial position:** Finally the tool is repositioned to the basic position, e.g. with the **LS_Positioner** system block.



❶ Position of the touch probe sensor	A Synchronising
❷ Starting position (basic tool position)	B Synchronous travel
❸ Starting position - synchronous range	C Travel back to initital position

[5-25] Phases and positions in the motion sequence of a "flying saw""

Synchronisation processes

In this example, always three movements are to be synchronised:

1. Running material to web (master value/master position).
2. Synchronising the slide with saw to web.
3. Releasing the cut with the saw.

Thus, three different measuring systems must be related to each other. In a path-controlled system, synchronising and the release of the cut can be started simultaneously, since the master value/master position ensure that both synchronisation processes are terminated at the same time. This, however, does not apply to the traverse path with different master speeds.

Operating mode

The traversing range has a starting position which can be defined via the *dnOpenPos_p* input. This starting position is also the target position for the **LS_Positioner** system block for repositioning.

As soon as the start of the synchronous range is reached, the cutting process can be started. The starting position of the synchronous range is defined via the *dnSyncPos_p* input.

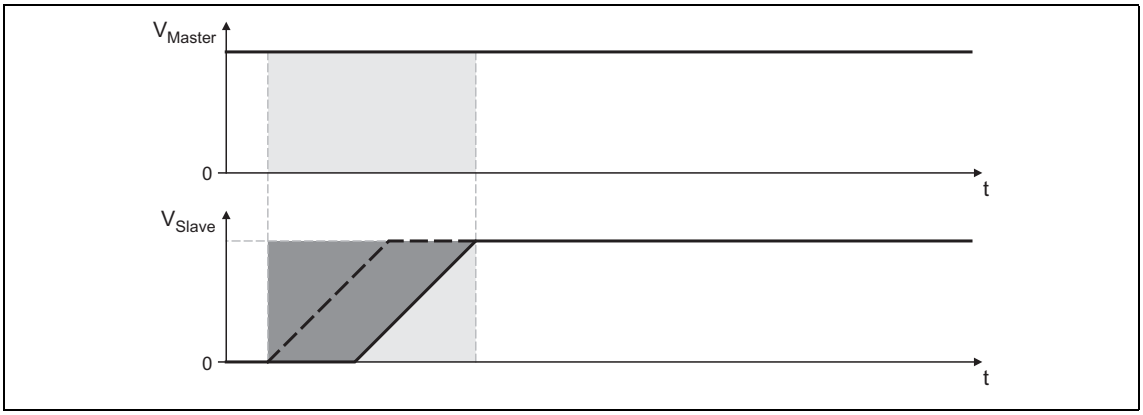
Afterwards the tool reports the end of the cutting process, e.g. using the boolean signals. This signal serves to activate the **LS_Positioner** system block to traverse the saw back into the *dnOpenPos_p* starting position.

During or after the termination of the re-positioning, the *bReset* input must be set from FALSE to TRUE once to activate the function block for the synchronisation of the next purpose.

A "top cut" can be activated by setting the *bManualSync* input to TRUE. In this case, synchronising is immediately started independent of the mark detection/position of the material.

5.11.2 Synchronous clutch-in

Due to the set ramps and the position of the touch probe sensor, no distance needs to be caught up. After clutch-in, the *dnSpeedOut_s* speed output follows the *dnSpeedIn_s* master speed.



[5-26] Principle "Synchronous clutch-in"

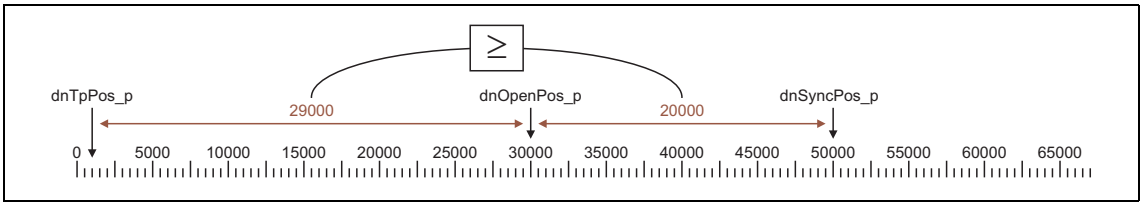
Tip!

Prefer the synchronous clutch-in to the oversynchronous clutch-in, since it prevents the material and tools from damage.

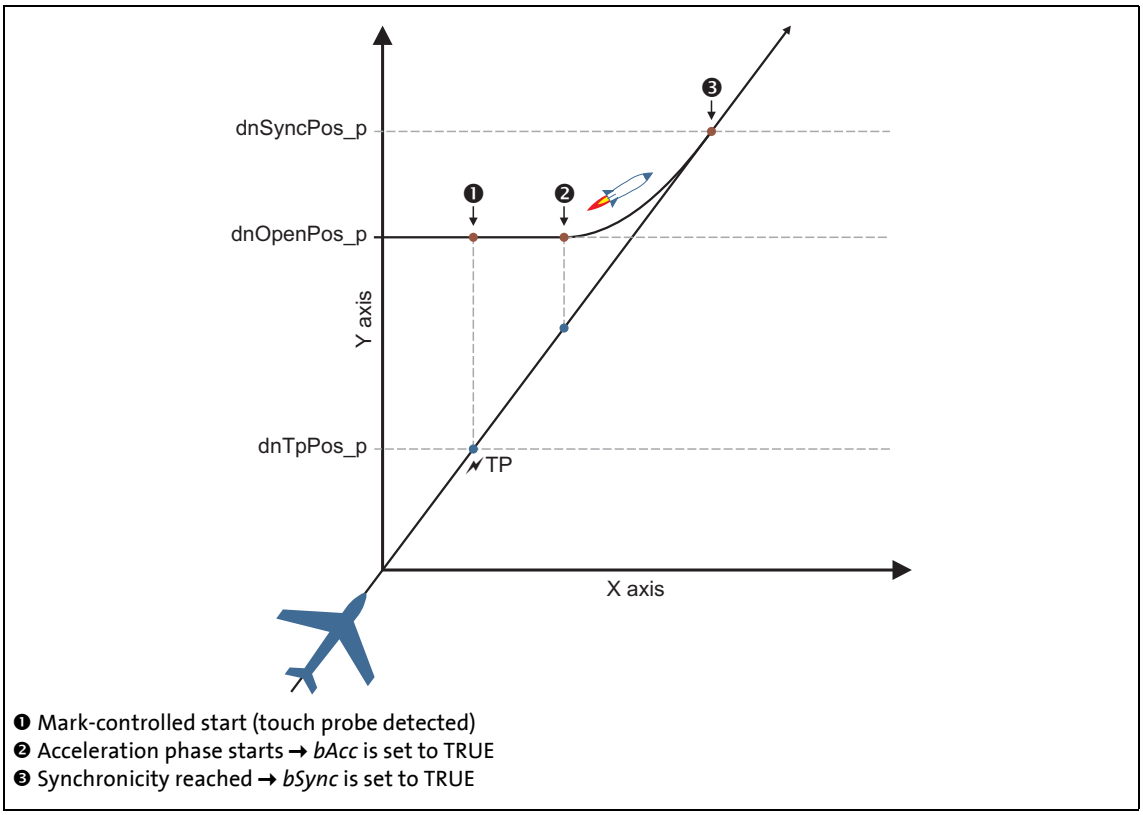
Preconditions

The following conditions must be fulfilled for synchronous clutch-in:

1. The difference between the position of the touch probe sensor and the basic position of the tool at standstill must be equal or greater than the difference between the basic function and the starting function of the synchronous range:



[5-27] Example: Distances of the individual positions



[5-28] Characteristics of positions for synchronous clutch-in

2. When *bTpReceived* or *bManualSync* are enabled, the resulting clutch-in time must not fall below the minimum clutch-in time parameterised in C05603.
   - The time refers to the reference speed when being reached and is thus reduced proportionately when the master speed is lower than the reference speed.

If the preconditions mentioned before are not fulfilled, clutch-in does not take place. The *bError* output is set to TRUE and the cause of error is displayed bit coded in C05609.

Sequence

Usually, the clutch-in process is started mark-controlled by setting the *bTpReceived* input to TRUE. For this purpose, the inputs *dnTpTimeLag* and *bTpReceived* must be connected to the touch probe system block for the digital input which is connected to the touch probe sensor.

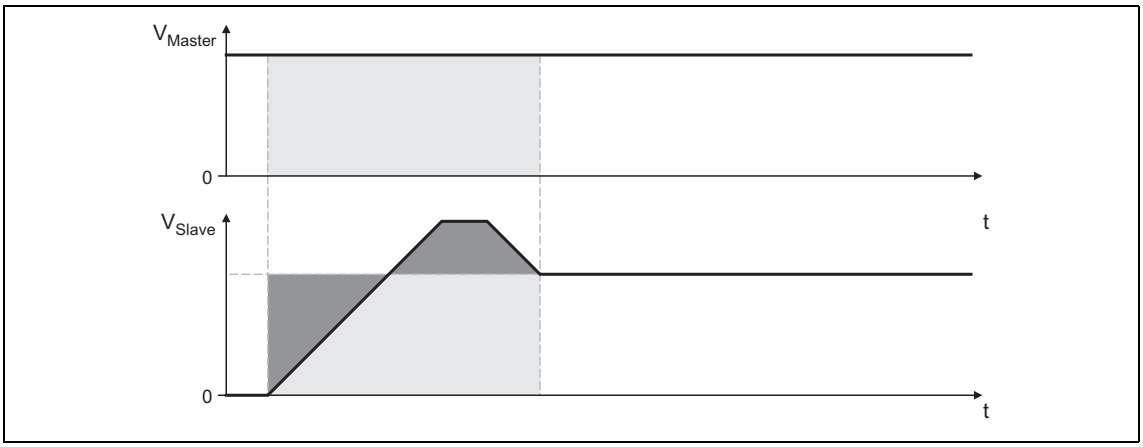
A manual clutch-in can be activated anytime via the *bManualSync* input (e.g. for "top cut" with a flying saw).

During the synchronising process, the *bAcc* output is set to TRUE. As soon as synchronicity has been reached (*dnPosOut_p* >= *dnSyncPos_p*), the *bSync* output is set to TRUE instead.

By setting the *bReset* input from FALSE to TRUE, the function block for the next synchronising can be activated.

5.11.3 Oversynchronous clutch-in

Oversynchronous clutch-in is always done in time-controlled mode. In contrast to the synchronous clutch-in, a distance must always be caught up. The distance may be caught up due to the defined ramp and/or the position of the touch probe sensor. After the catching-up process, the *dnSpeedOut_s* speed output follows the *dnSpeedIn_s* master speed.

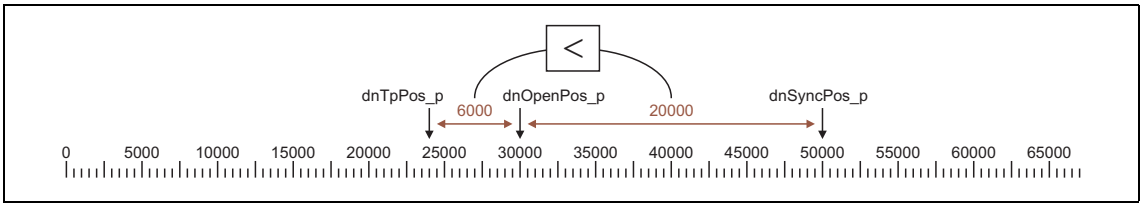


[5-29] Principle "Oversynchronous clutch-in"

Preconditions

The following conditions must be fulfilled for oversynchronous clutch-in:

1. The difference between the position of the touch probe sensor and the basic position of the tool at standstill must be lower than the difference between the basic function and the starting function of the synchronous range:



[5-30]

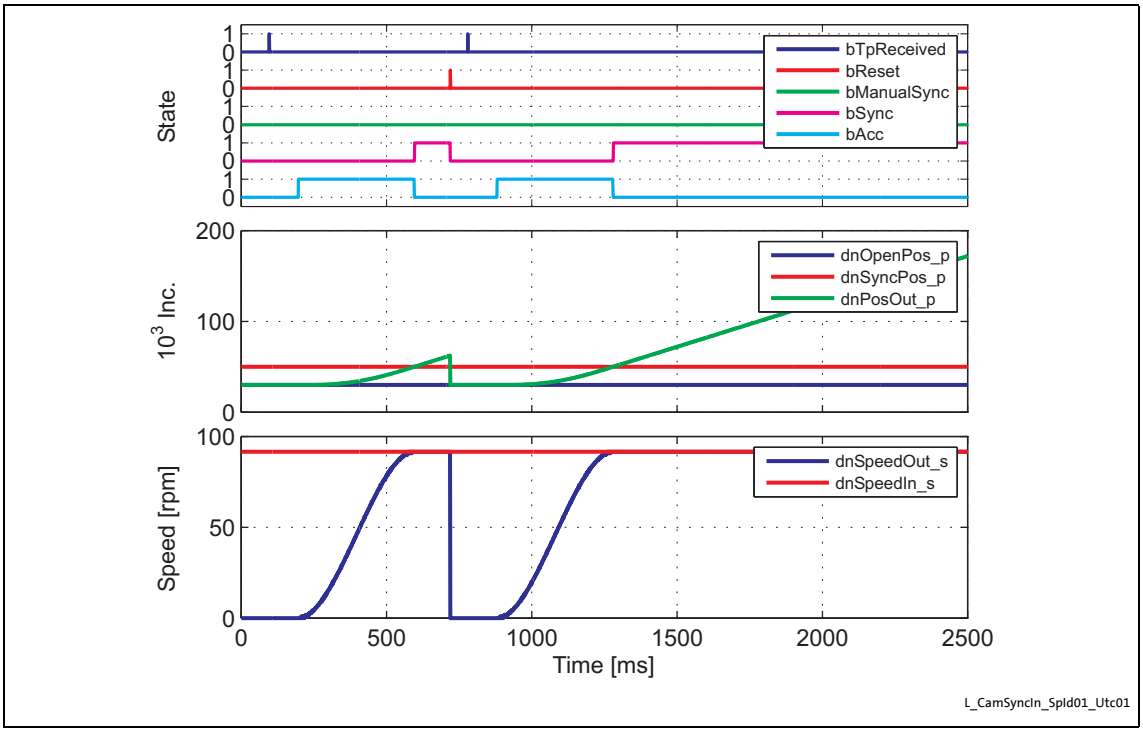
2. The maximum oversynchronous speed to be parameterised in C05602 must be greater than the master speed. The lower the distance of the touch probe sensor from the starting position of the synchronous range, the greater must be the oversynchronous speed.

If the preconditions mentioned before are not fulfilled, clutch-in does not take place. The *bError* output is set to TRUE and the cause of error is displayed bit-coded in C05609. Remedy: Increase value of C05602 until the error does not occur anymore.

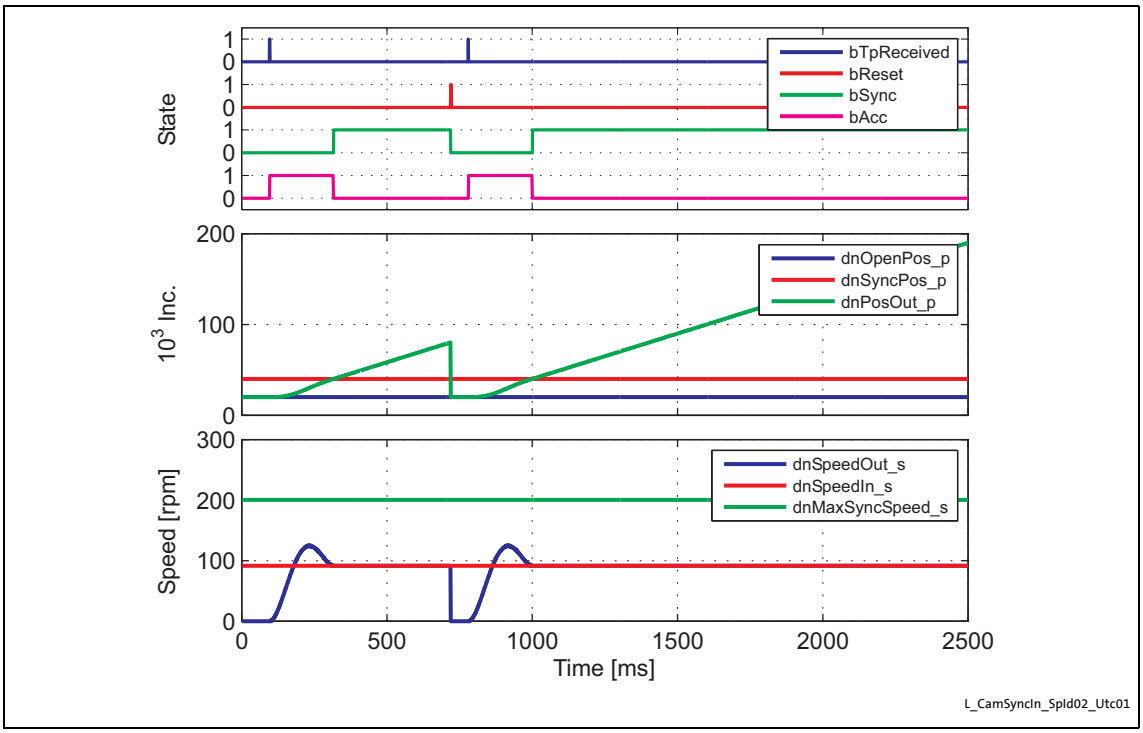
Sequence

The sequence corresponds to the sequence for a synchronous clutch-in described in the previous chapter. ▸ Synchronous clutch-in (📖 191)

5.11.4 Signal characteristics

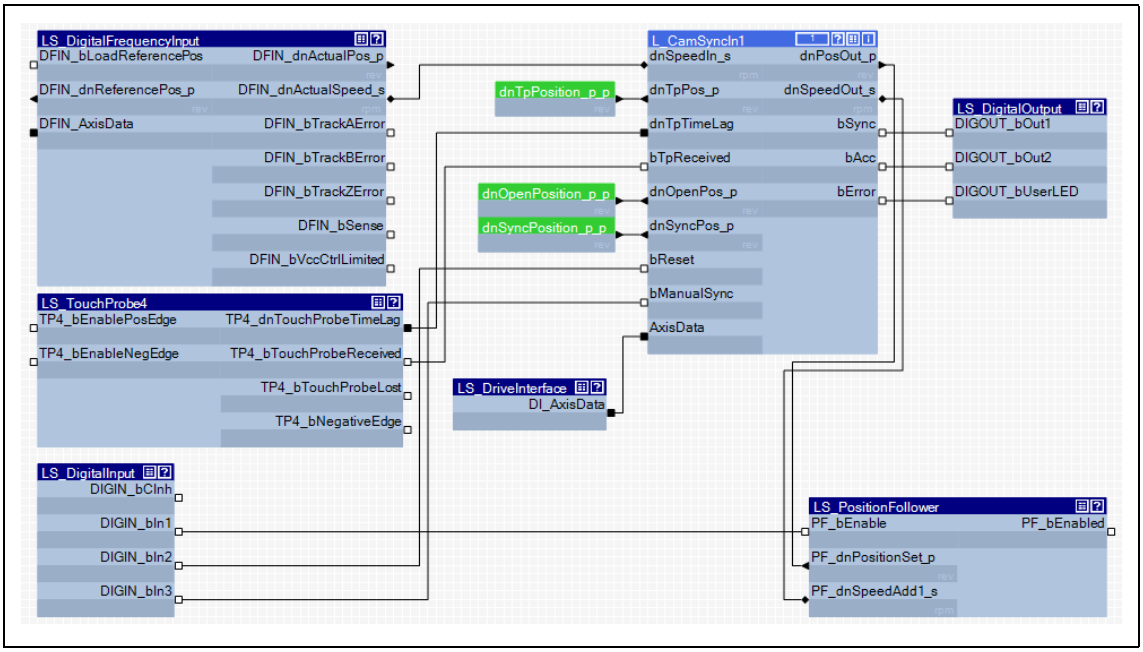


[5-31] Synchronous clutch-in



[5-32] Oversynchronous clutch-in

5.11.5 Interconnection example

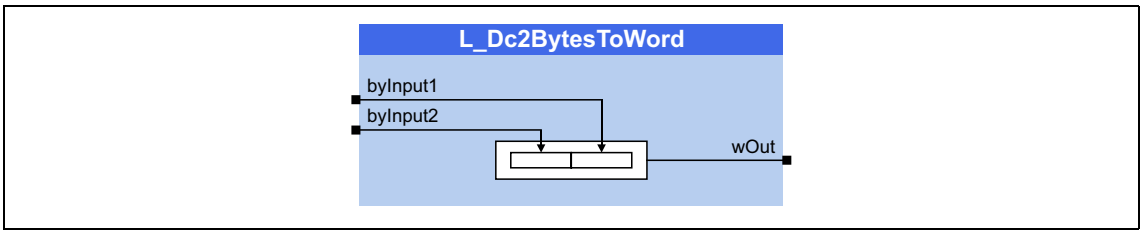


[5-33] Interconnection example "flying saw"

5.12 L_Dc2BytesToWord - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts two input values of type "BYTE" into an output value of type "WORD".



Inputs

Identifier/data type	Information/possible settings
byInput1 BYTE	Input value 1 ≡ bit 0 ... bit 7 of *wOut*
byInput2 BYTE	Input value 2 ≡ bit 8 ... bit 15 of *wOut*

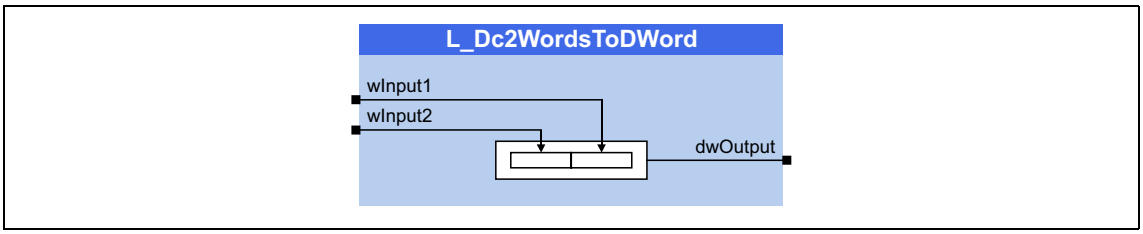
Outputs

Identifier/data type	Value/meaning
wOut WORD	Value of type "word" according to the transmitted input values

5.13 L_Dc2WordsToDWord- type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts two input values of type "WORD" into an output value of type "DWORD".



Inputs

Identifier/data type	Information/possible settings
wInput1 WORD	Input value 1 ≡ bit 0 ... bit 15 of *dwOutput*
wInput2 WORD	Input value 2 ≡ bit 16 ... bit 31 of *dwOutput*

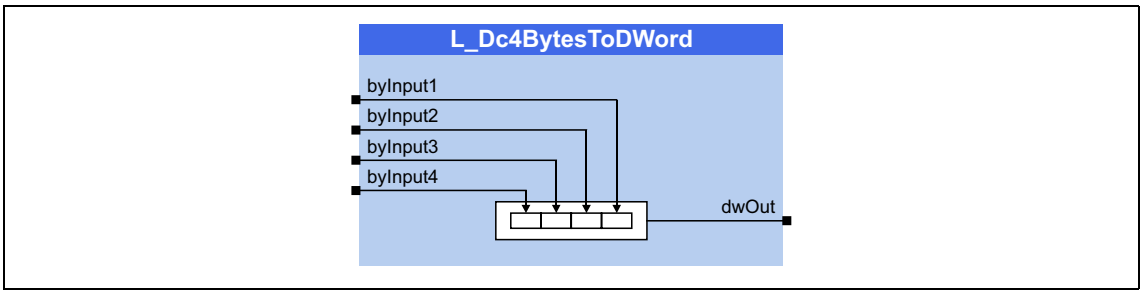
Outputs

Identifier/data type	Value/meaning
dwOutput DWORD	Value of type "double word" according to the transmitted input values

5.14 L_Dc4BytesToDWord - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts four input values of type "BYTE" into an output value of type "DWORD".



Inputs

Identifier/data type	Information/possible settings
byInput1 BYTE	Input value 1 ≡ bit 0 ... bit 7 of *dwOut*
byInput2 BYTE	Input value 2 ≡ bit 8 ... bit 15 of *dwOut*
byInput3 BYTE	Input value 3 ≡ bit 16 ... bit 23 of *dwOut*
byInput4 BYTE	Input value 4 ≡ bit 24 ... bit 31 of *dwOut*

Outputs

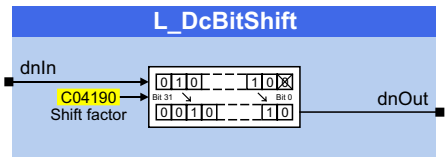
Identifier/data type	Value/meaning
dwOut DWORD	Value of type "double word" according to the transmitted input values

5.15 L_DcBitShift - shift operation

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a bit shift operation with a "DINT" value.

- The shift factor, i.e. the number of bits by which the input value is shifted to the left or right, is defined by a parameter.
- The result of the shift operation is output at the output *dnOut*.
- The sign remains.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal

Outputs

Identifier/data type	Value/meaning
dnOut DINT	Output signal according to the parameterised bit shift operation

Parameter

Parameter	Possible settings			Information
C04190	-31		31	Shift factor • Initialisation: 0

Example

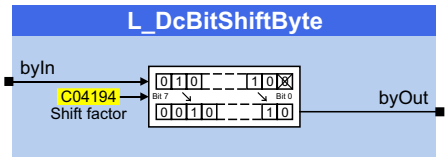


5.16 L_DcBitShiftByte - shift operation

Function library:	LenzeDataConversion	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out a bit shift operation with a "BYTE" value.

- The shift factor, i.e. the number of bits by which the input value is shifted to the left or right, is defined by a parameter.
- The result of the shift operation is output at the output *byOut*.



Inputs

Identifier/data type	Information/possible settings
byIn BYTE	Input signal

Outputs

Identifier/data type	Value/meaning
byOut BYTE	Output signal according to the parameterised bit shift operation

Parameter

Parameter	Possible settings			Information
C04194	-7		7	Shift factor • Initialisation: 0

Example

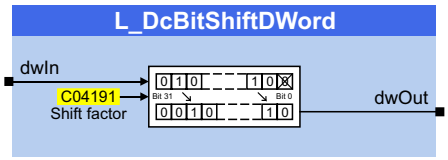


5.17 L_DcBitShiftDWord - shift operation

Function library:	LenzeDataConversion	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out a bit shift operation with a "DWORD" value.

- The shift factor, i.e. the number of bits by which the input value is shifted to the left or right, is defined by a parameter.
- The result of the shift operation is output at the output *dwOut*.



Inputs

Identifier/data type	Information/possible settings
dwIn DWORD	Input signal

Outputs

Identifier/data type	Value/meaning
dwOut DWORD	Output signal according to the parameterised bit shift operation

Parameter

Parameter	Possible settings			Information
C04191	-31		31	Shift factor • Initialisation: 0

Example

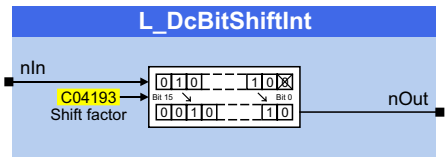


5.18 L_DcBitShiftInt - shift operation

Function library:	LenzeDataConversion	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out a bit shift operation with a "INT" value.

- The shift factor, i.e. the number of bits by which the input value is shifted to the left or right, is defined by a parameter.
- The result of the shift operation is output at the output *nOut*.
- The sign remains.



Inputs

Identifier/data type	Information/possible settings
nIn INT	Input signal

Outputs

Identifier/data type	Value/meaning
nOut INT	Output signal according to the parameterised bit shift operation

Parameter

Parameter	Possible settings			Information
C04193	-15		15	Shift factor • Initialisation: 0

Example

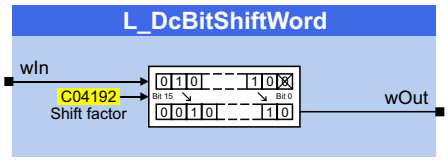


5.19 L_DcBitShiftWord - shift operation

Function library:	LenzeDataConversion	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out a bit shift operation with a "WORD" value.

- The shift factor, i.e. the number of bits by which the input value is shifted to the left or right, is defined by a parameter.
- The result of the shift operation is output at the output w*Out*.



Inputs

Identifier/data type	Information/possible settings
wIn WORD	Input signal

Outputs

Identifier/data type	Value/meaning
wOut WORD	Output signal according to the parameterised bit shift operation

Parameter

Parameter	Possible settings			Information
C04192	-15		15	Shift factor • Initialisation: 0

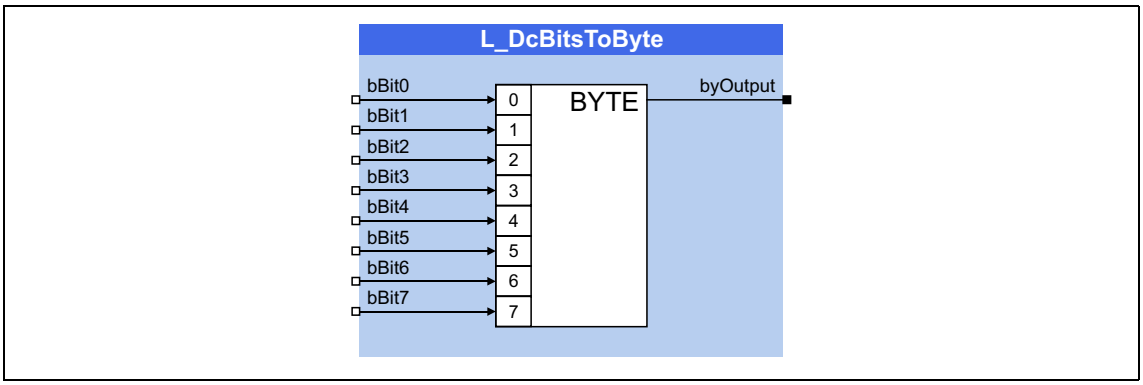
Example



5.20 L_DcBitsToByte - bit multiplexer

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts 8 single bit values into a value of type "BYTE".



Inputs

Identifier/data type	Information/possible settings
bBit0 BOOL	Input for bit value with valency 2^0
...	...
bBit7 BOOL	Input for bit valency 2^7

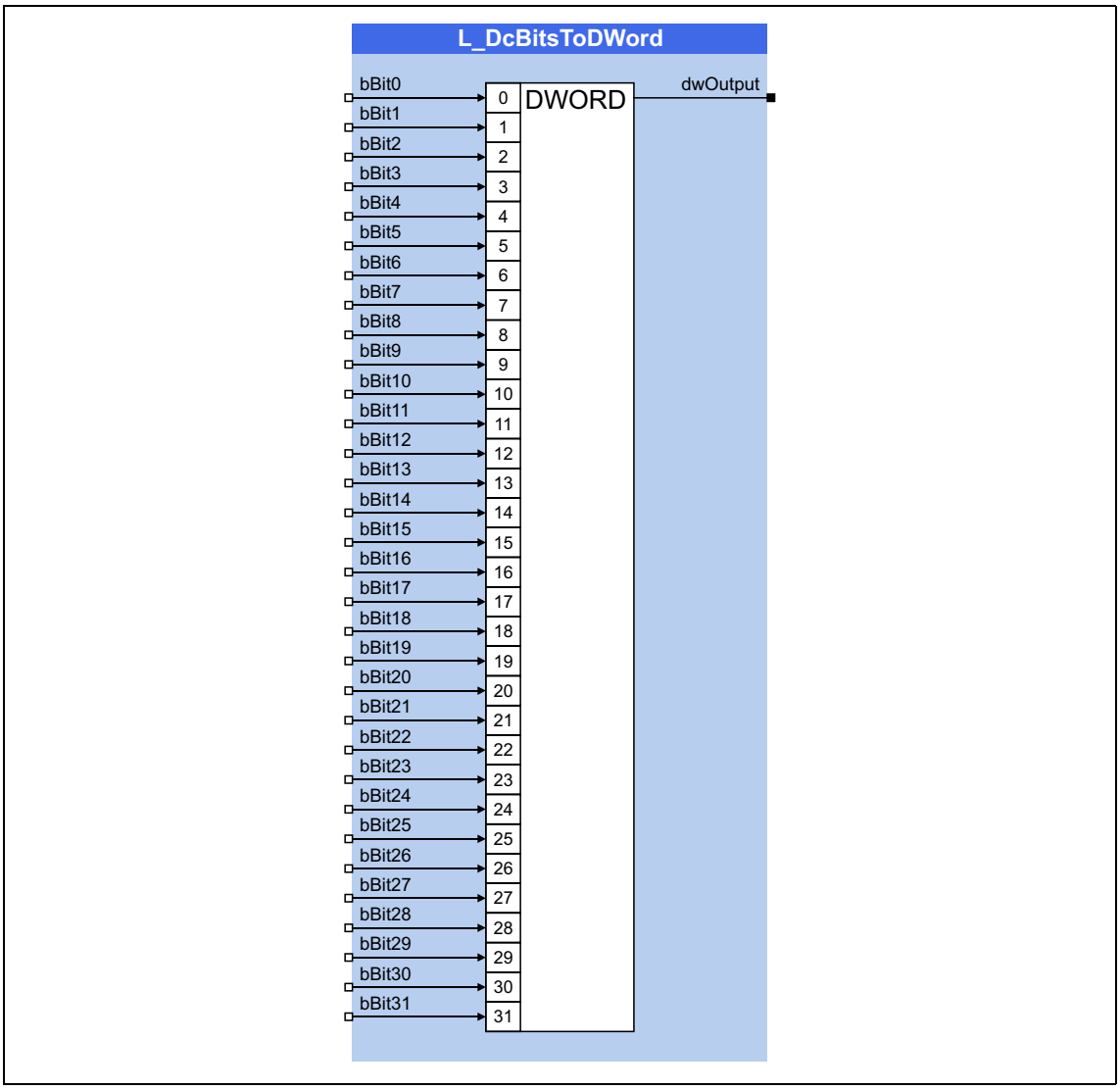
Outputs

Identifier/data type	Value/meaning
byOutput BYTE	Value of type "byte" according to the transmitted bit values

5.21 L_DcBitsToDWord - bit multiplexer

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts 32 single bit values into a value of "DWORD" type.



Inputs

Identifier/data type	Information/possible settings
bBit0 BOOL	Input for bit value with valency 2^0
...	...
bBit31 BOOL	Input for bit value with valency 2^{31}

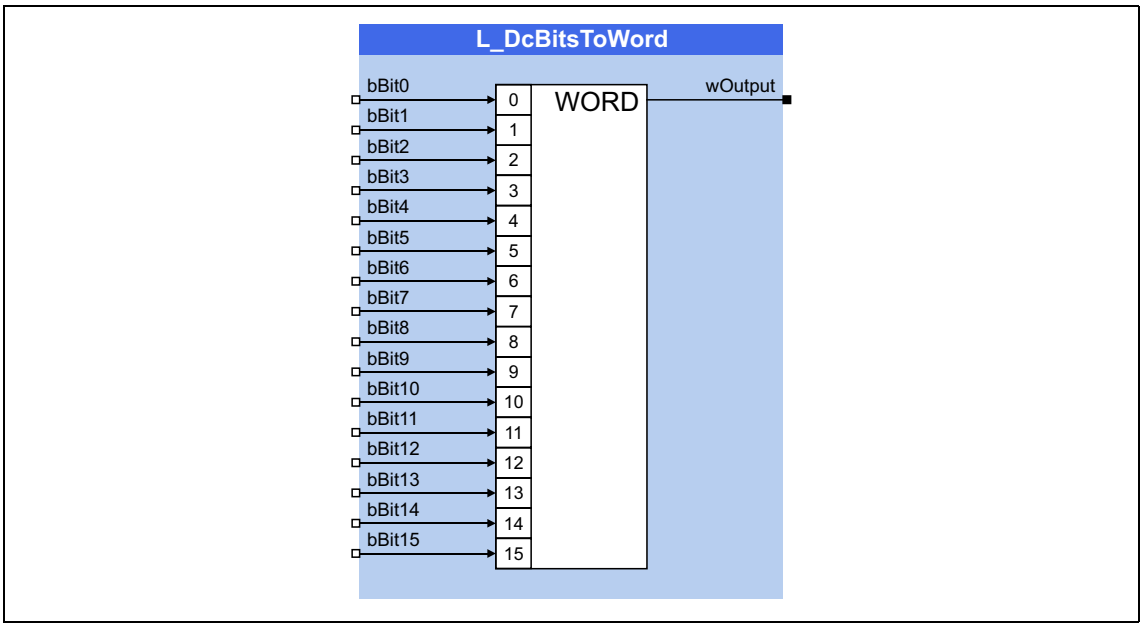
Outputs

Identifier/data type	Value/meaning
dwOutput DWORD	Value of type "double word" according to the transmitted bit values

5.22 L_DcBitsToWord - bit multiplexer

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts 16 single bit values into a value of "WORD" type.



Inputs

Identifier/data type	Information/possible settings
bBit0 BOOL	Input for bit value with valency 2^0
...	...
bBit15 BOOL	Input for bit value with valency 2^{15}

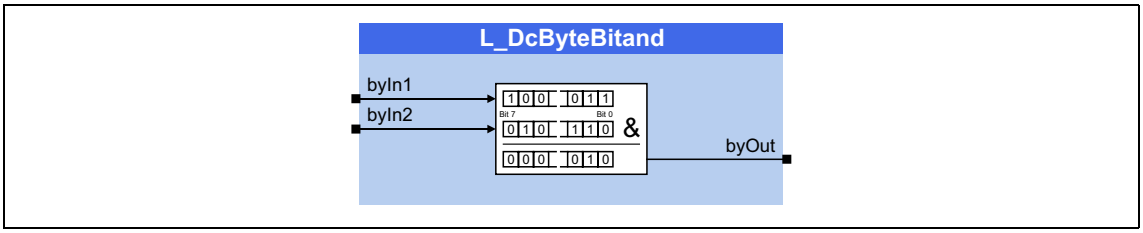
Outputs

Identifier/data type	Value/meaning
wOutput WORD	Value of "word" type according to the transmitted bit values.

5.23 L_DcByteBitand - bit combination

Function library:	LenzeDataConversion	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out a bit-by-bit AND operation of two input values of "BYTE" type and also outputs the results as "BYTE".



Inputs

Identifier/data type	Information/possible settings
byIn1 BYTE	Value 1 for bit-by-bit AND combination
byIn2 BYTE	Value 2 for bit-by-bit AND combination

Outputs

Identifier/data type	Value/meaning
byOut BYTE	Result of the bit-by-bit AND operation of *byIn1* and *byIn2*

Function

byIn1 - bit n	byIn2 - bit n	byOut - bit n
0	0	0
0	1	0
1	0	0
1	1	1

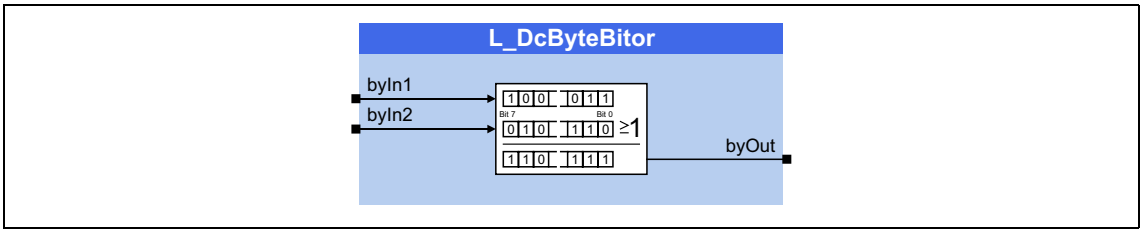
Example

```
byIn1 = 00101001
byIn2 = 01111000
        --------
byOut = 00101000
```

5.24 L_DcByteBitor - bit combination

Function library:	LenzeDataConversion	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out a bit-by-bit OR operation of two input values of "BYTE" type and also outputs the results as "BYTE".



Inputs

Identifier/data type	Information/possible settings
byIn1 BYTE	Value 1 for bit-by-bit OR combination
byIn2 BYTE	Value 2 for bit-by-bit OR combination

Outputs

Identifier/data type	Value/meaning
byOut BYTE	Result of the bit-by-bit OR operation of *byIn1* and *byIn2*

Function

byIn1 - bit n	byIn2 - bit n	byOut - bit n
0	0	0
0	1	1
1	0	1
1	1	1

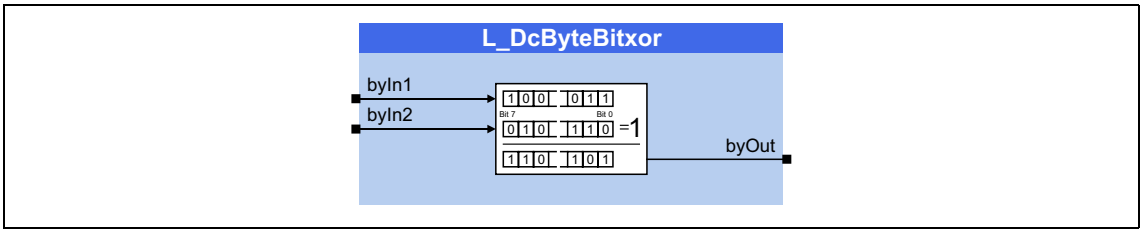
Example

```
byIn1 = 00101001
byIn2 = 01111000
        --------
byOut = 01111001
```

5.25 L_DcByteBitxor - bit combination

Function library:	LenzeDataConversion	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out a bit-by-bit EXCLUSIVE/OR operation of two input values of "BYTE" type and also outputs the results as "BYTE".



Inputs

Identifier/data type	Information/possible settings
byIn1 BYTE	Value 1 for bit-by-bit EXCLUSIVE OR combination
byIn2 BYTE	Value 2 for bit-by-bit EXCLUSIVE OR combination

Outputs

Identifier/data type	Value/meaning
byOut BYTE	Result of the bit-by-bit EXCLUSIVE/OR operation of *byIn1* and *byIn2*

Function

byIn1 - bit n	byIn2 - bit n	byOut - bit n
0	0	0
0	1	1
1	0	1
1	1	0

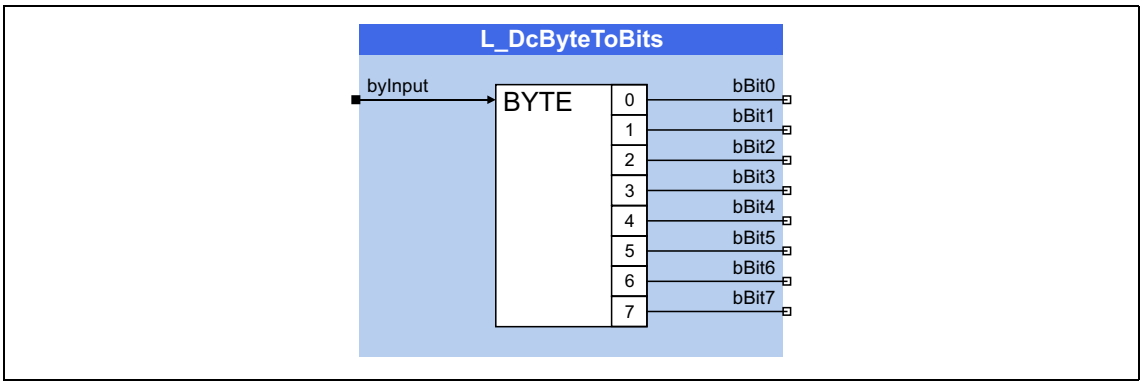
Example

```
byIn1 = 00101001
byIn2 = 01111000
        --------
byOut = 01010001
```

5.26 L_DcByteToBits - bit demultiplexer

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB outputs the 8 corresponding bit values for an input value of type "BYTE".



Inputs

Identifier/data type	Information/possible settings
byInput BYTE	Value to be converted

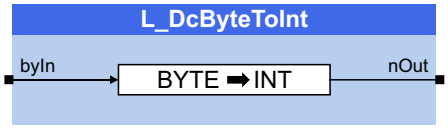
Outputs

Identifier/data type	Value/meaning
bBit0 BOOL	Bit 0 output of *byInput* (valency: 20)
bBit1 BOOL	Bit 1 output of *byInput* (valency: 2^1)
bBit2 BOOL	Bit 2 output of *byInput* (valency: 2^2))
...	...
bBit7 BOOL	Bit 7 output of *byInput* (valency: 27)

5.27 L_DcByteToInt - type converter

Function library:	LenzeDataConversion	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB converts a data type "BYTE" into the data type "INT".



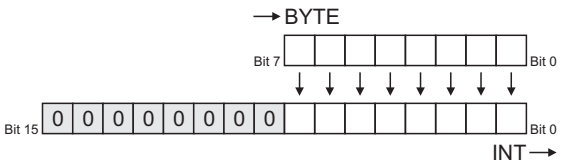
Inputs

Identifier/data type	Information/possible settings
byIn BYTE	Input signal

Outputs

Identifier/data type	Value/meaning
nOut INT	Output signal

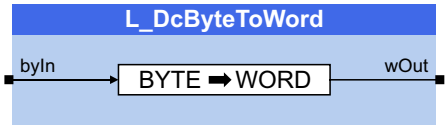
Function



5.28 L_DcByteToWord - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a data type "BYTE" into the data type "WORD".



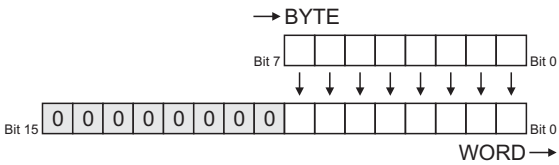
Inputs

Identifier/data type	Information/possible settings
byIn BYTE	Input signal

Outputs

Identifier/data type	Value/meaning
wOut WORD	Output signal

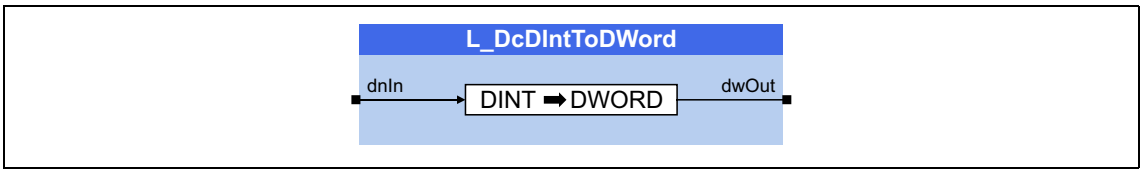
Function



5.29 L_DcDIntToDWord - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a data type "DINT" into the data type "DWORD".



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal

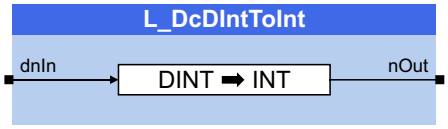
Outputs

Identifier/data type	Value/meaning
dwOut DWORD	Output signal

5.30 L_DcDIntToInt - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a data type "DINT" into the data type "INT".



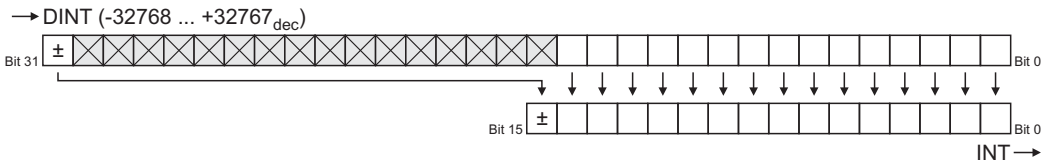
Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal • Internal limitation to -32768 ... +32767 (INT display area).

Outputs

Identifier/data type	Value/meaning
nOut INT	Output signal

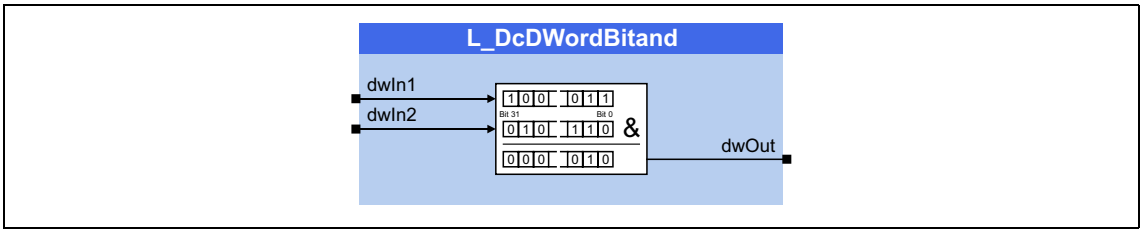
Function



5.31 L_DcDWordBitand - bit combination

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a bit-by-bit AND operation of two input values of "DWORD" type and also outputs the results as "DWORD".



Inputs

Identifier/data type	Information/possible settings
dwIn1 DWORD	Value 1 for bit-by-bit AND combination
dwIn2 DWORD	Value 2 for bit-by-bit AND combination

Outputs

Identifier/data type	Value/meaning
dwOut DWORD	Result of the bit-by-bit AND operation of *dwIn1* and *dwIn2*

Function

dwIn1 - bit n	dwIn2 - bit n	dwOut - bit n
0	0	0
0	1	0
1	0	0
1	1	1

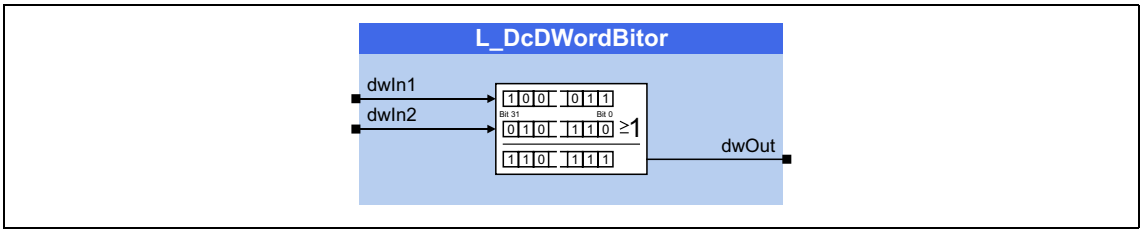
Example

```
dwIn1 = 00011001010011100001011000101000
dwIn2 = 01010001100001100100100001111000
        --------------------------------
dwOut = 00010001000001100000000000101000
```

5.32 L_DcDWordBitor - bit combination

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a bit-by-bit OR operation of two input values of "DWORD" type and also outputs the results as "DWORD".



Inputs

Identifier/data type	Information/possible settings
dwIn1 DWORD	Value 1 for bit-by-bit OR combination
dwIn2 DWORD	Value 2 for bit-by-bit OR combination

Outputs

Identifier/data type	Value/meaning
dwOut DWORD	Result of the bit-by-bit OR operation of *dwIn1* and *dwIn2*

Function

dwIn1 - bit n	dwIn2 - bit n	dwOut - bit n
0	0	0
0	1	1
1	0	1
1	1	1

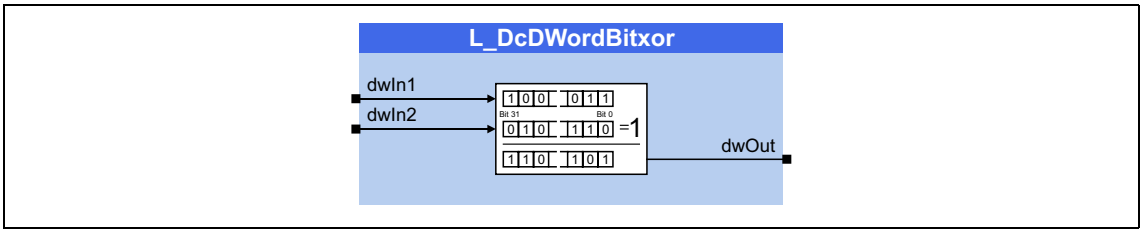
Example

```
dwIn1 = 00011001010011100001011000101000
dwIn2 = 01010001100001100100100001111000
        --------------------------------
dwOut = 01011001110011100101111001111000
```

5.33 L_DcDWordBitxor - bit combination

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a bit-by-bit EXCLUSIVE OR operation of two input values of "DWORD" type and also outputs the results as "DWORD".



Inputs

Identifier/data type	Information/possible settings
dwIn1 DWORD	Value 1 for bit-by-bit EXCLUSIVE OR combination
dwIn2 DWORD	Value 2 for bit-by-bit EXCLUSIVE OR combination

Outputs

Identifier/data type	Value/meaning
dwOut DWORD	Result of the bit-by-bit EXCLUSIVE OR operation of *dwIn1* and *dwIn2*

Function

dwIn1 - bit n	dwIn2 - bit n	dwOut - bit n
0	0	0
0	1	1
1	0	1
1	1	0

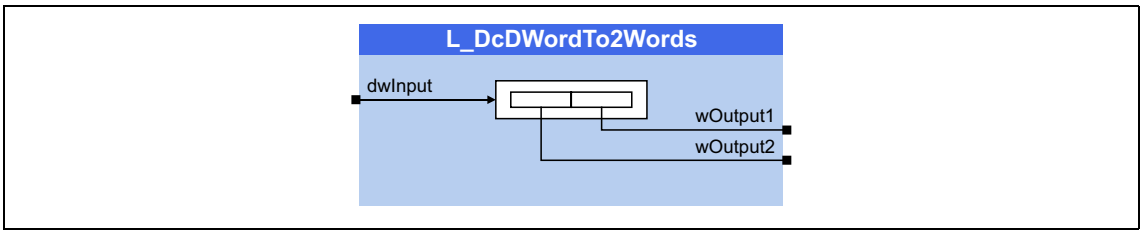
Example

```
dwIn1 = 00011001010011100001011000101000
dwIn2 = 01010001100001100100100001111000
        --------------------------------
dwOut = 01001000110010000101111001010000
```

5.34 L_DcDWordTo2Words - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts an input value of type "DWORD" into two output values of type "WORD".



Inputs

Identifier/data type	Information/possible settings
dwInput DWORD	Input value

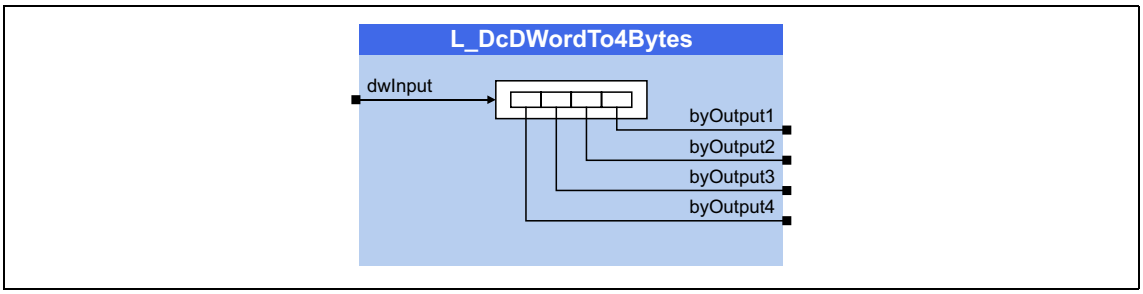
Outputs

Identifier/data type	Value/meaning
wOutput1 WORD	Output value 1 ≡ bit 0 ... bit 15 of *dwInput*
wOutput2 WORD	Output value 2 ≡ bit 16 ... bit 31 of *dwInput*

5.35 L_DcDWordTo4Bytes - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts an input value of type "DWORD" into four output values of type "BYTE".



Inputs

Identifier/data type	Information/possible settings
dwInput DWORD	Input value

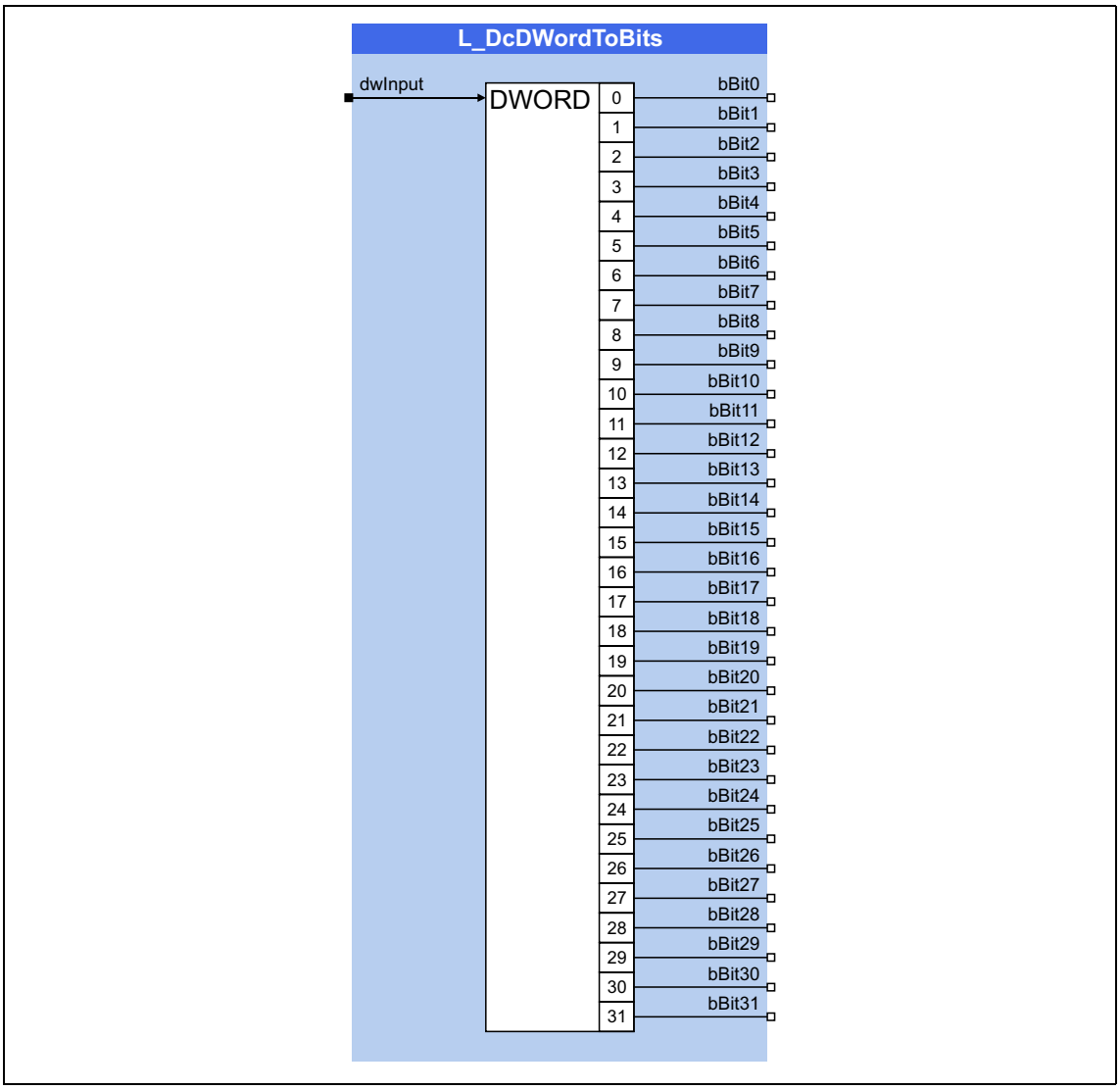
Outputs

Identifier/data type	Value/meaning
byOutput1 BYTE	Output value 1 ≡ bit 0 ... bit 7 of *dwInput*
byOutput2 BYTE	Output value 2 ≡ bit 8 ... bit 15 of *dwInput*
byOutput3 BYTE	Output value 3 ≡ bit 16 ... bit 23 of *dwInput*
byOutput4 BYTE	Output value 4 ≡ bit 24 ... bit 31 of *dwInput*

5.36 L_DcDWordToBits - bit demultiplexer

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB outputs the 32 corresponding bit values for an input value of "DWORD" type.



Inputs

Identifier/data type	Information/possible settings
dwInput DWORD	Value to be converted

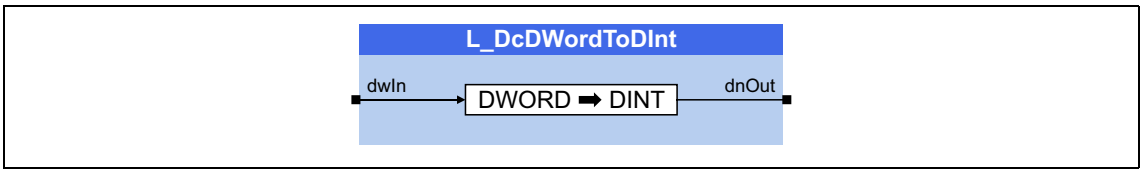
Outputs

Identifier/data type	Value/meaning
bBit0 BOOL	Bit 0 output of *dwInput* (valency: 2^0)
bBit1 BOOL	Bit 1 output of *dwInput* (valency: 21)
bBit2 BOOL	Bit 2 output of *dwInput* (valency: 2^2)
...	...
bBit31 BOOL	Bit 31 output of *dwInput* (valency: 2^{31})

5.37 L_DcDWordToDInt - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a data type "DWORD" into the data type "DINT".



Inputs

Identifier/data type	Information/possible settings
dwIn DWORD	Input signal

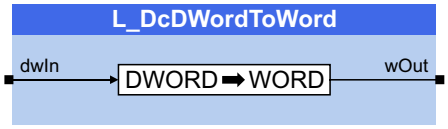
Outputs

Identifier/data type	Value/meaning
dnOut DINT	Output signal

5.38 L_DcDWordToWord - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a data type "DWORD" into the data type "WORD".



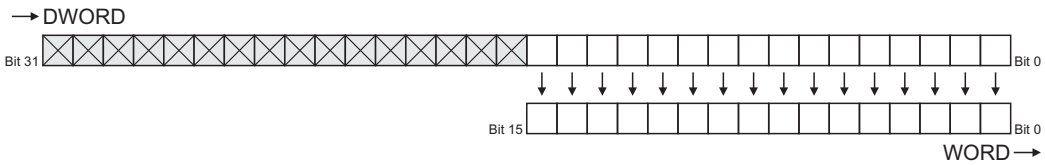
Inputs

Identifier/data type	Information/possible settings
dwIn DWORD	Input signal

Outputs

Identifier/data type	Value/meaning
wOut WORD	Output signal

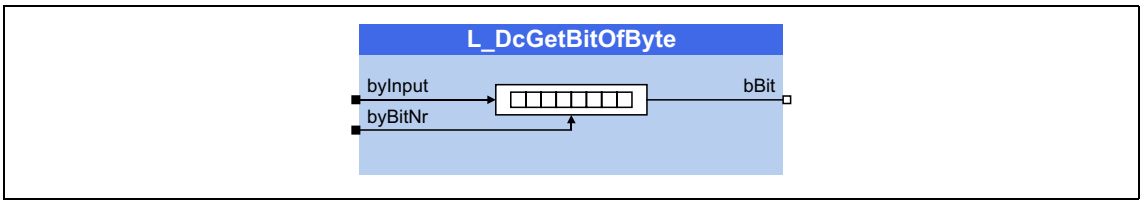
Function



5.39 L_DcGetBitOfByte - bit operation

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB returns the state of a single bit within a "BYTE" value.



Inputs

Identifier/data type	Information/possible settings
byInput BYTE	Input signal
byBitNr BYTE	No. (0 ... 7) of the bit of *byInput* the state of which is to be detected.

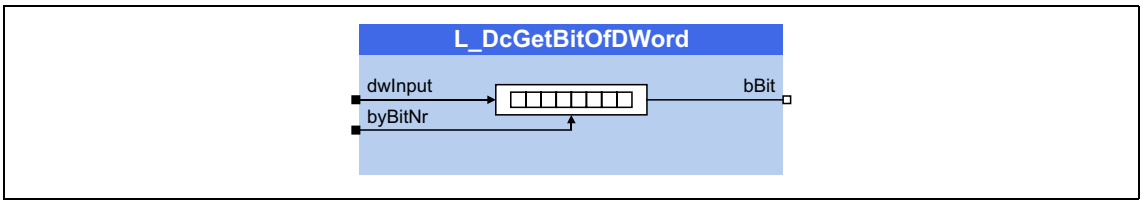
Outputs

Identifier/data type	Value/meaning
bBit BOOL	State of bit *byBitNr* of *byInput*.

5.40 L_DcGetBitOfDWord - bit operation

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB returns the state of a single bit within a "DWORD" value.



Inputs

Identifier/data type	Information/possible settings
dwInput DWORD	Input signal
byBitNr BYTE	No. (0 ... 31) of the bit of *dwInput* the state of which is to be detected.

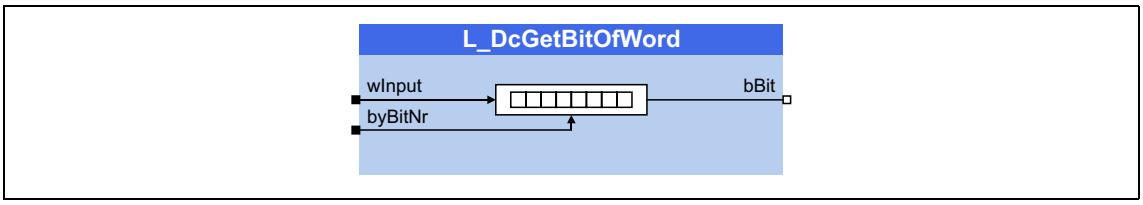
Outputs

Identifier/data type	Value/meaning
bBit BOOL	State of bit *byBitNr* of *dwInput*.

5.41 L_DcGetBitOfWord - bit operation

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB returns the state of a single bit within a "WORD" value.



Inputs

Identifier/data type	Information/possible settings
wInput WORD	Input signal
byBitNr BYTE	No. (0 ... 15) of the bit of *wInput* the state of which is to be detected.

Outputs

Identifier/data type	Value/meaning
bBit BOOL	State of bit *byBitNr* of *wInput*.

5.42 L_DcIntToDInt - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a data type "INT" into the data type "DINT".



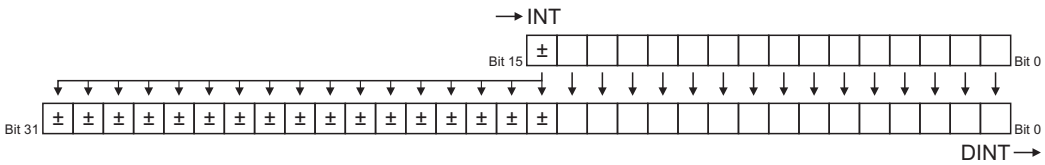
Inputs

Identifier/data type	Information/possible settings
nIn INT	Input signal

Outputs

Identifier/data type	Value/meaning
dnOut DINT	Output signal

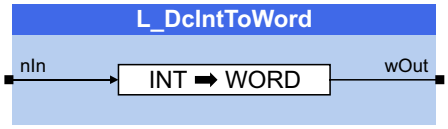
Function



5.43 L_DcIntToWord - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a data type "INT" into the data type "WORD" .



Inputs

Identifier/data type	Information/possible settings
nIn INT	Input signal

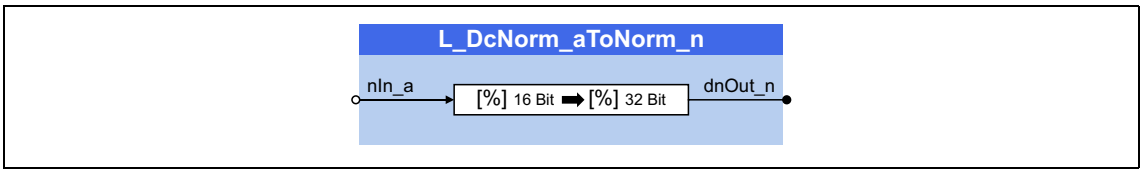
Outputs

Identifier/data type	Value/meaning
wOut WORD	Output signal

5.44 L_DcNorm_aToNorm_n - signal converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a scaled 16-bit signal into a scaled 32-bit signal.



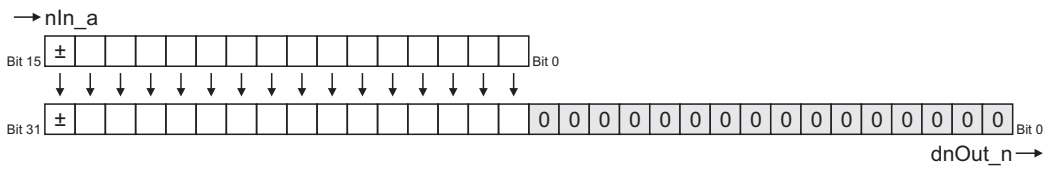
Inputs

Identifier/data type	Information/possible settings
nIn_a INT	Input signal • 100 % ≡ 214 ≡ 16384 **Note:** If the input information has a value of "-32768", the value will internally be limited to "-32767" and copied to the HighWord of the *dnOut_n* output. Therefore, this FB cannot be used to combine the LowWord and the HighWord in order to form a double word.

Outputs

Identifier/data type	Value/meaning
dnOut_n DINT	Output signal • 100 % ≡ 2^{30} ≡ 1073741824

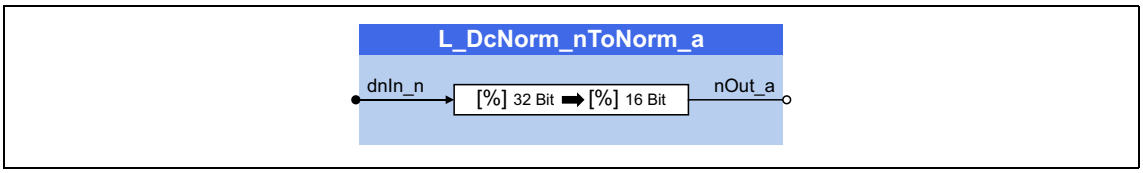
Function



5.45 L_DcNorm_nToNorm_a - signal converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a scaled 32-bit signal into a scaled 16-bit signal.



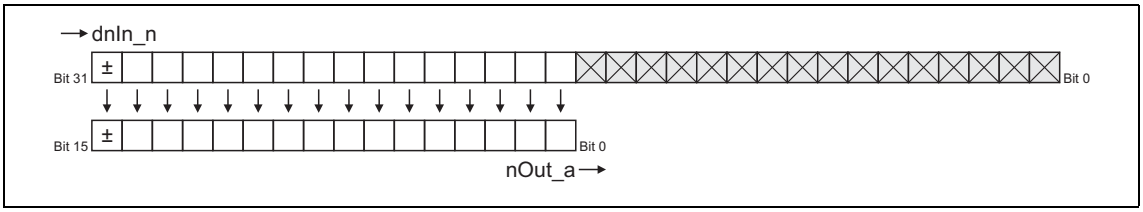
Inputs

Identifier/data type	Information/possible settings
dnIn_n DINT	Input signal • 100 % ≡ 2^{30} ≡ 1073741824

Outputs

Identifier/data type	Value/meaning
nOut_a INT	Output signal • 100 % ≡ 214 ≡ 16384

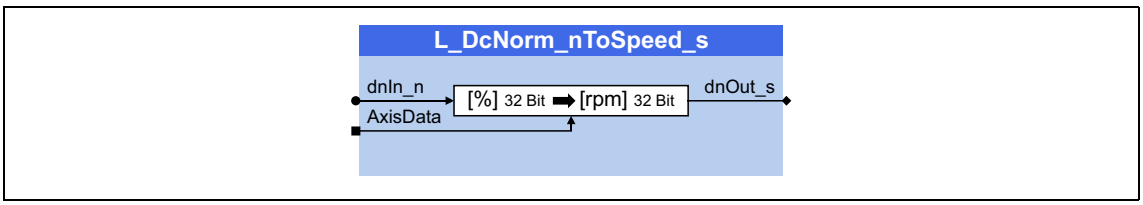
Function



5.46 L_DcNorm_nToSpeed_s - signal converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts a scaled 32-bit signal into a 32-bit speed signal.



Inputs

Identifier/data type	Information/possible settings
dnIn_n DINT	Input signal in [%] • 100 % ≡ 2^{30} ≡ 1073741824
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

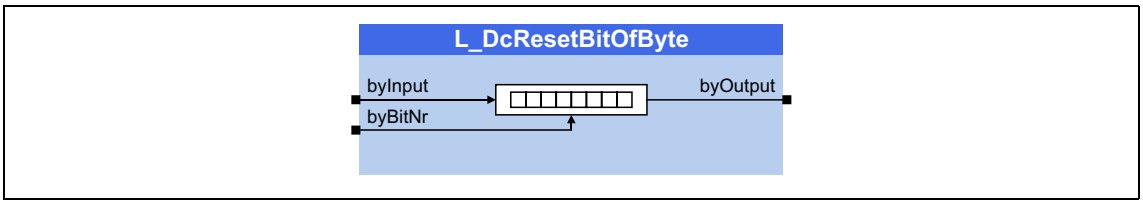
Outputs

Identifier/data type	Value/meaning
dnOut_s DINT	Output signal in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864

5.47 L_DcResetBitOfByte - bit operation

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB resets a single bit in a "BYTE" type value to "0".



Inputs

Identifier/data type	Information/possible settings
byInput BYTE	Input signal
byBitNr BYTE	No. (0 ... 7) of the bit to be reset.

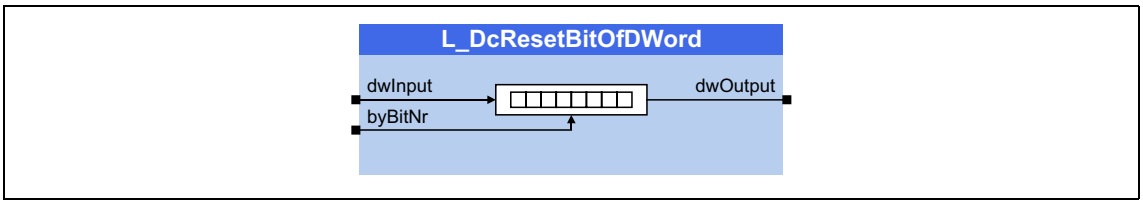
Outputs

Identifier/data type	Value/meaning
byOutput BYTE	Value of type "byte" resulting from the reset bit.

5.48 L_DcResetBitOfDWord - bit operation

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB resets a single bit in a "DWORD" type value to "0".



Inputs

Identifier/data type	Information/possible settings
dwInput DWORD	Input signal
byBitNr BYTE	No. (0 ... 31) of the bit to be reset.

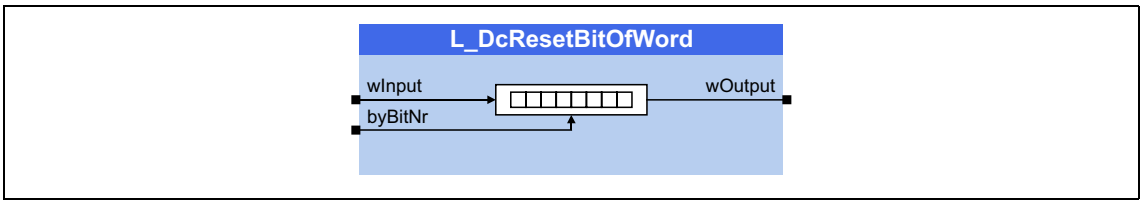
Outputs

Identifier/data type	Value/meaning
dwOutput DWORD	Value of type "double word" resulting from the reset bit.

5.49 L_DcResetBitOfWord - bit operation

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB resets a single bit in a "WORD" type value to "0".



Inputs

Identifier/data type	Information/possible settings
wInput WORD	Input signal
byBitNr BYTE	No. (0 ... 15) of the bit to be reset.

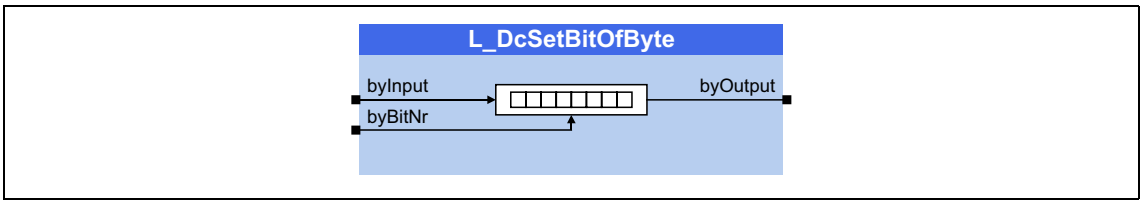
Outputs

Identifier/data type	Value/meaning
wOutput WORD	Value of type "word" resulting from the reset bit.

5.50 L_DcSetBitOfByte - bit operation

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB sets a single bit in a "BYTE" type value to "1".



Inputs

Identifier/data type	Information/possible settings
byInput BYTE	Input signal
byBitNr BYTE	No. (0 ... 7) of the bit to be set.

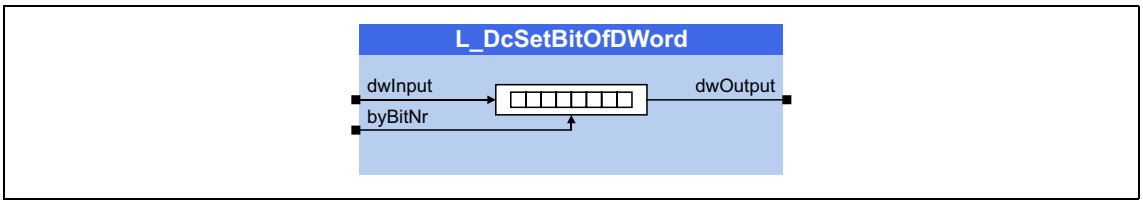
Outputs

Identifier/data type	Value/meaning
byOutput BYTE	Value of type "byte" resulting from the set bit.

5.51 L_DcSetBitOfDWord - bit operation

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB sets a single bit in a "DWORD" type value to "1".



Inputs

Identifier/data type	Information/possible settings
dwInput DWORD	Input signal
byBitNr BYTE	No. (0 ... 31) of the bit to be set.

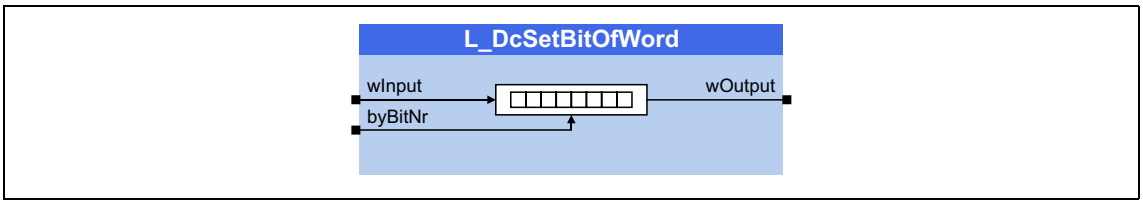
Outputs

Identifier/data type	Value/meaning
dwOutput DWORD	Value of type "double word" resulting from the set bit.

5.52 L_DcSetBitOfWord - bit operation

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB sets a single bit in a "WORD" type value to "1".



Inputs

Identifier/data type	Information/possible settings
wInput WORD	Input signal
byBitNr BYTE	No. (0 ... 15) of the bit to be set.

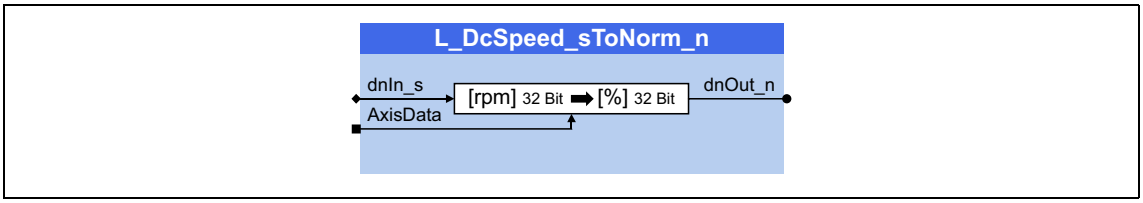
Outputs

Identifier/data type	Value/meaning
wOutput WORD	Value of type "word" resulting from the set bit.

5.53 L_DcSpeed_sToNorm_n - signal converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts a 32-bit speed signal into a scaled 32-bit signal.



Inputs

Identifier/data type	Information/possible settings
dnIn_s DINT	Input signal in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

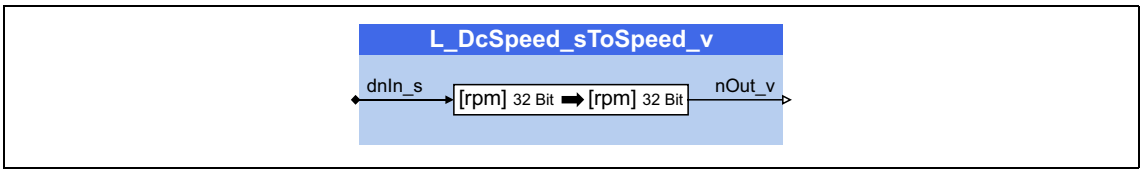
Outputs

Identifier/data type	Value/meaning
dnOut_n DINT	Output signal in [%] • 100 % ≡ 2^{30} ≡ 1073741824

5.54 L_DcSpeed_sToSpeed_v - signal converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a 32-bit speed signal into a 16-bit speed signal.



Inputs

Identifier/data type	Information/possible settings
dnIn_s DINT	Input signal in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864

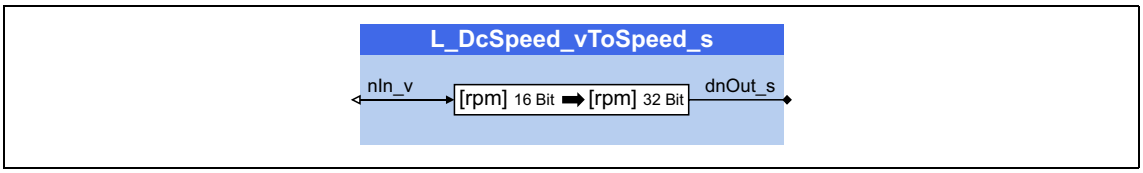
Outputs

Identifier/data type	Value/meaning
nOut_v INT	Output signal in [rpm] • 15000 rpm ≡ ^{214} ≡ 16384

5.55 L_DcSpeed_vToSpeed_s - signal converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a 16-bit speed signal into a 32-bit speed signal.



Inputs

Identifier/data type	Information/possible settings
nIn_v INT	Input signal in [rpm] • 15000 rpm ≡ 2^{14} ≡ 16384

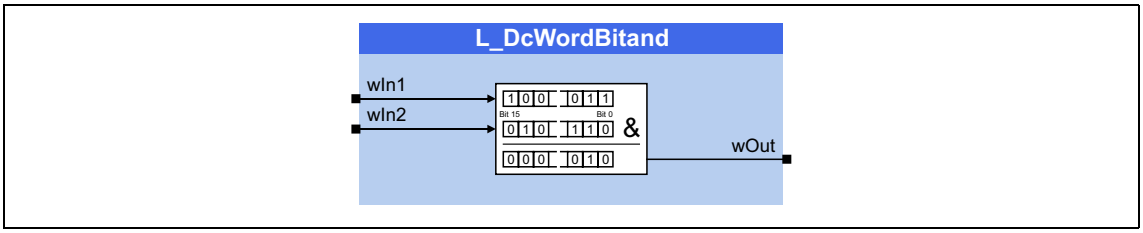
Outputs

Identifier/data type	Value/meaning
dnOut_s DINT	Output signal in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864

5.56 L_DcWordBitand - bit combination

Function library:	LenzeDataConversion	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out a bit-by-bit AND operation of two input values of "WORD" type and also outputs the results as "WORD".



Inputs

Identifier/data type	Information/possible settings
wIn1 WORD	Value 1 for bit-by-bit AND combination
wIn2 WORD	Value 2 for bit-by-bit AND combination

Outputs

Identifier/data type	Value/meaning
wOut WORD	Result of the bit-by-bit AND operation of *wIn1* and *wIn2*

Function

wIn1 - bit n	wIn2 - bit n	wOut - bit n
0	0	0
0	1	0
1	0	0
1	1	1

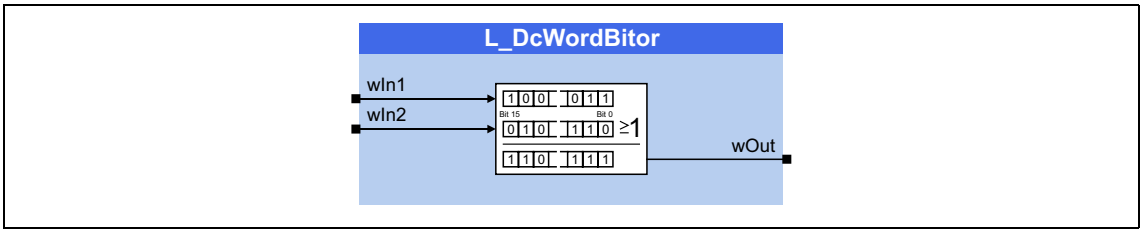
Example

```
wIn1 = 0001011000101000
wIn2 = 0100100001111000
       ----------------
wOut = 0000000000101000
```

5.57 L_DcWordBitor - bit combination

Function library:	LenzeDataConversion	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out a bit-by-bit OR operation of two input values of "WORD" type and also outputs the results as "WORD".



Inputs

Identifier/data type	Information/possible settings
wIn1 WORD	Value 1 for bit-by-bit OR combination
wIn2 WORD	Value 2 for bit-by-bit OR combination

Outputs

Identifier/data type	Value/meaning
wOut WORD	Result of the bit-by-bit OR operation of *wIn1* and *wIn2*

Function

wIn1 - bit n	wIn2 - bit n	wOut - bit n
0	0	0
0	1	1
1	0	1
1	1	1

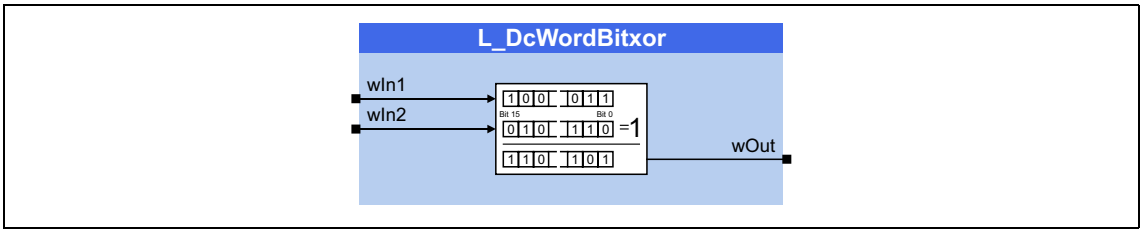
Example

```
wIn1 = 0001011000101000
wIn2 = 0100100001111000
       ----------------
wOut = 0101111001111000
```

5.58 L_DcWordBitxor - bit combination

Function library:	LenzeDataConversion	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out a bit-by-bit EXCLUSIVE OR operation of two input values of "WORD" type and also outputs the results as "WORD".



Inputs

Identifier/data type	Information/possible settings
wIn1 WORD	Value 1 for bit-by-bit EXCLUSIVE OR combination
wIn2 WORD	Value 2 for bit-by-bit EXCLUSIVE OR combination

Outputs

Identifier/data type	Value/meaning
wOut WORD	Result of the bit-by-bit EXCLUSIVE OR operation of *wIn1* and *wIn2*

Function

wIn1 - bit n	wIn2 - bit n	wOut - bit n
0	0	0
0	1	1
1	0	1
1	1	0

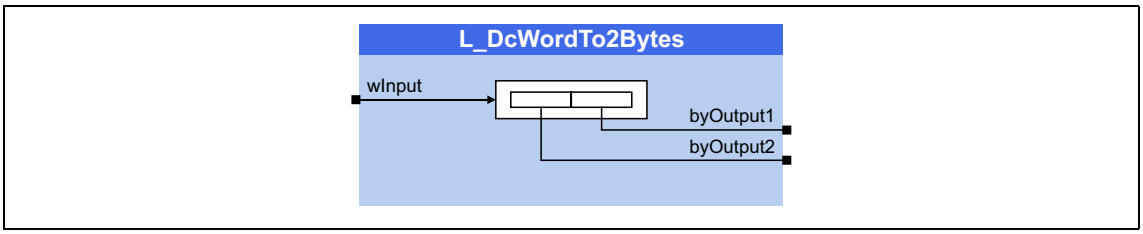
Example

```
wIn1 = 0001011000101000
wIn2 = 0100100001111000
       ----------------
wOut = 0101111001010000
```

5.59 L_DcWordTo2Bytes - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts an input value of type "WORD" into two output values of type "BYTE".



Inputs

Identifier/data type	Information/possible settings
wInput WORD	Input value

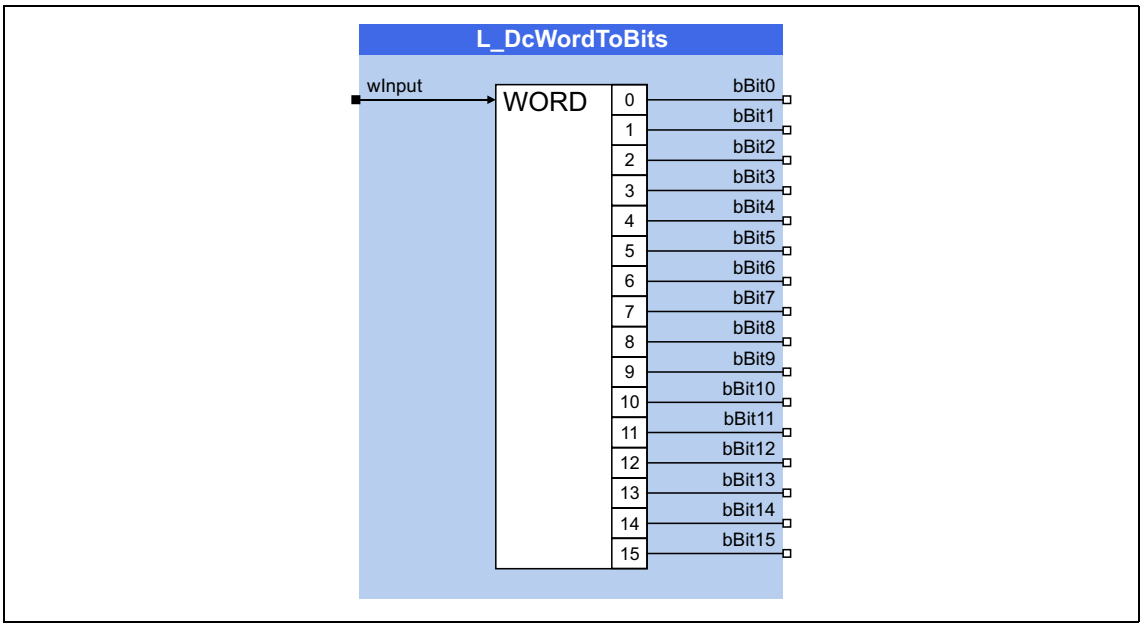
Outputs

Identifier/data type	Value/meaning
byOutput1 BYTE	Output value 1 ≡ bit 0 ... bit 7 of *wInput*
byOutput2 BYTE	Output value 2 ≡ bit 8 ... bit 15 of *wInput*

5.60 L_DcWordToBits - bit demultiplexer

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB outputs the 16 corresponding bit values for an input value of type "WORD".



Inputs

Identifier/data type	Information/possible settings
wInput WORD	Value to be converted

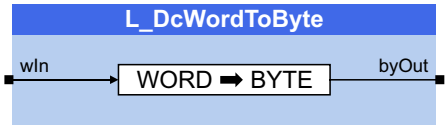
Outputs

Identifier/data type	Value/meaning
bBit0 BOOL	Bit 0 output of *byInput* (valency: 20)
bBit1 BOOL	Bit 1 output of *byInput* (valency: 2^1)
bBit2 BOOL	Bit 2 output of *byInput* (valency: 2^2))
...	...
bBit15 BOOL	Bit 15 output of *byInput* (valency: 2^{15})

5.61 L_DcWordToByte - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a data type "WORD" into the data type "BYTE" .



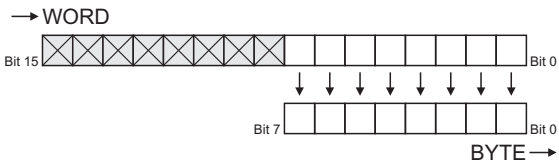
Inputs

Identifier/data type	Information/possible settings
wIn WORD	Input signal

Outputs

Identifier/data type	Value/meaning
byOut BYTE	Output signal

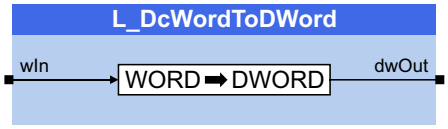
Function



5.62 L_DcWordToDWord - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a data type "WORD" into the data type "DWORD" .



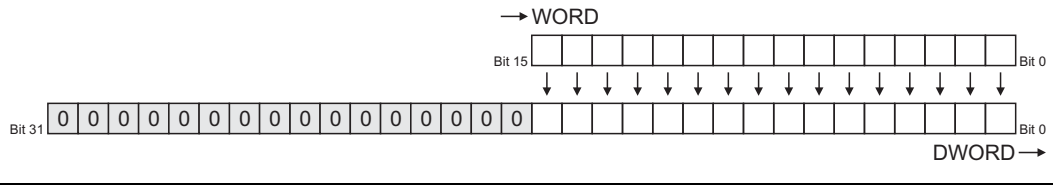
Inputs

Identifier/data type	Information/possible settings
wIn WORD	Input signal

Outputs

Identifier/data type	Value/meaning
dwOut DWORD	Output signal

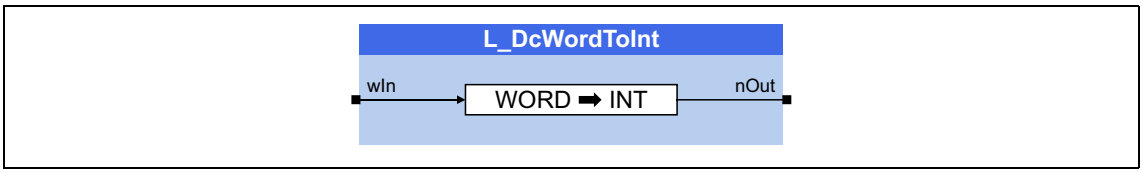
Function



5.63 L_DcWordToInt - type converter

Function library:	LenzeDataConversion
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts a data type "WORD" into the data type "INT".



Inputs

Identifier/data type	Information/possible settings
wIn WORD	Input signal

Outputs

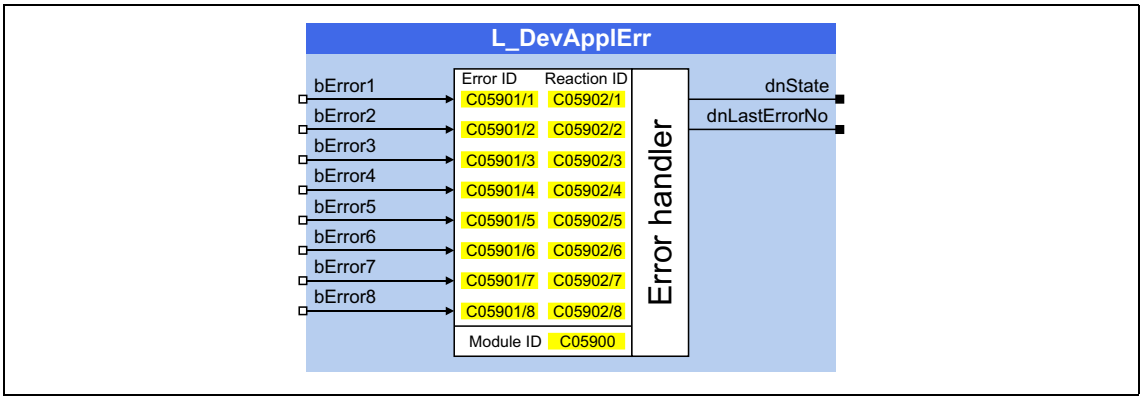
Identifier/data type	Value/meaning
nOut INT	Output signal

5.64 L_DevApplErr - error tripping

Function library:	LenzeDevice9400
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB is used for error handling in the application.

- Via the 8 boolean inputs up to 8 different application error messages with parameterisable module ID, error ID and error response can be released by the application.
- In contrast to FB L_DevApplErrFix the module ID; error ID, and the error response of this FB can be set via parameters and not via inputs.
- If more than 8 application error messages are required, the FB can be instanced.
- If several inputs are set to TRUE at the same time, the input with the lowest number will activate the error message.



Inputs

Identifier/data type		Information/possible settings	
bError1	BOOL	Input for error message 1 (highest priority)	
		FALSE↗TRUE	Error condition 1 is met and the parameterised application error message is output by the operating system.
		FALSE	Error condition 1 is no longer met.
...		...	
bError8	BOOL	Input for error message 8 (lowest priority)	
		FALSE↗TRUE	Error condition 8 is met and the parameterised application error message is output by the operating system.
		FALSE	Error condition 8 is no longer met.

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Display of the error message last transmitted to the operating system.	
	1	Error message 1 is active.
	2	Error message 2 is active.
	4	Error message 3 is active.
	8	Error message 4 is active.
	16	Error message 5 is active.
	32	Error message 6 is active.
	64	Error message 7 is active.
	128	Error message 8 is active.
dnLastErrorNo DINT	Error number of the error message last transmitted to the operating system.	

Parameter

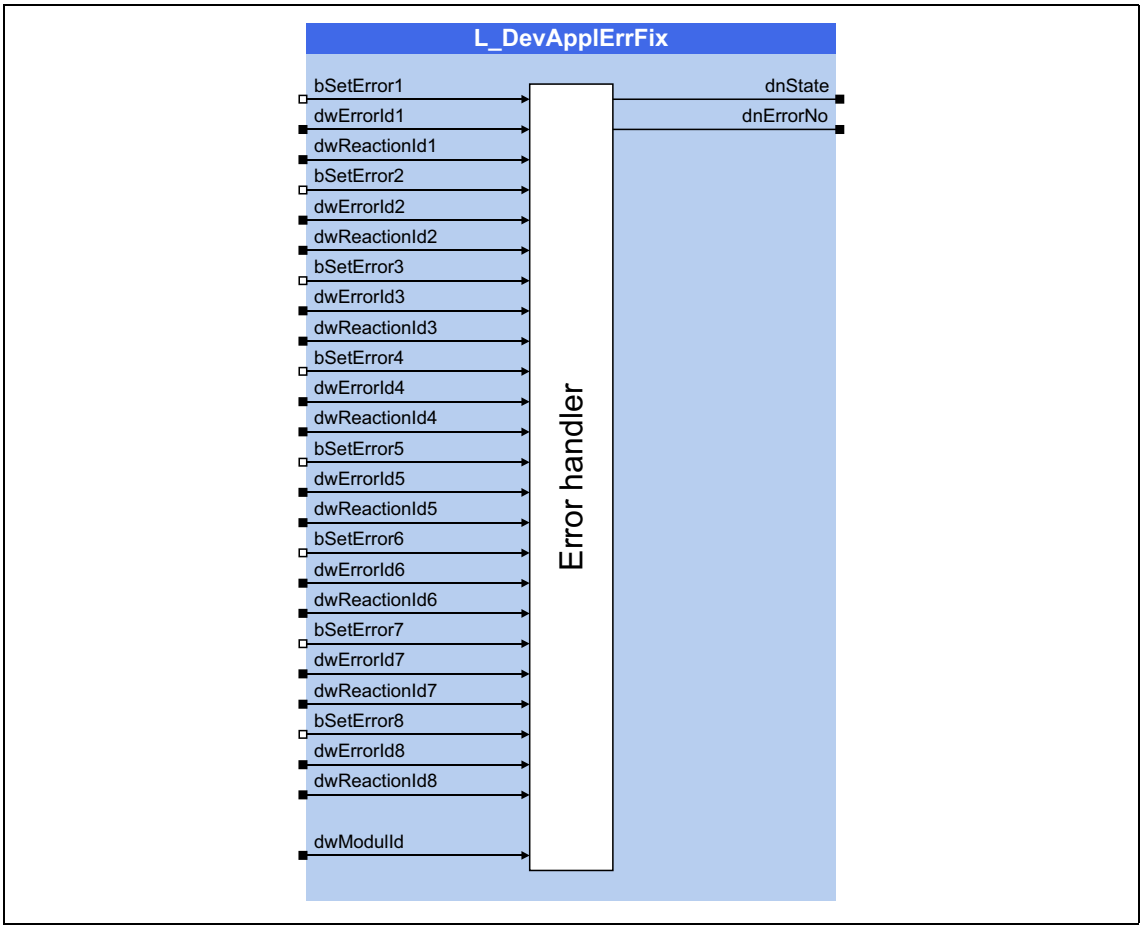
Parameter	Possible settings			Information
C05900	980		999	Module ID • Initialisation: 999
C05901/1...8	0		65535	Error ID • Initialisation: 0
C05902/1...8				Error response
	0	None		
	1	Error		
	2	Trouble		
	3	Quick stop by trouble		
	4	Warning locked		
	5	Warning		
	6	Information		

5.65 L_DevApplErrFix - error tripping

Function library:	LenzeDevice9400	**FB is available as of library V02.00!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB is used for error handling in the application.

- Via the 8 boolean inputs up to 8 different application error messages can be released by the application.
- In contrast to the L_DevApplErr FB the module ID, error ID, and the error response of this FB can be set via inputs and not via parameters.
- If more than 8 application error messages are required, the FB can be instanced.
- If several inputs are set to TRUE at the same time, the input with the lowest number will activate the error message.



Inputs

Identifier/data type	Information/possible settings	
bSetError1 BOOL	Input for error message 1 (highest priority)	
	FALSE↗TRUE	Error condition 1 is met and the parameterised application error message is output by the operating system.
	FALSE	Error condition 1 is no longer met.
...	...	
bSetError8 BOOL	Input for error message 8 (lowest priority)	
	FALSE↗TRUE	Error condition 8 is met and the parameterised application error message is output by the operating system.
	FALSE	Error condition 8 is no longer met.
dwErrorId1...8 DWORD	Error ID (0 ... 65535) for error message 1 ... 8	
dwReactionId1...8 DWORD	Error response for error message 1 ... 8	
	0	None
	1	Error
	2	Trouble
	3	Quick stop by trouble
	4	Warning locked
	5	Warning
	6	Information
dwModulId DWORD	Module ID (980 ... 990)	

Outputs

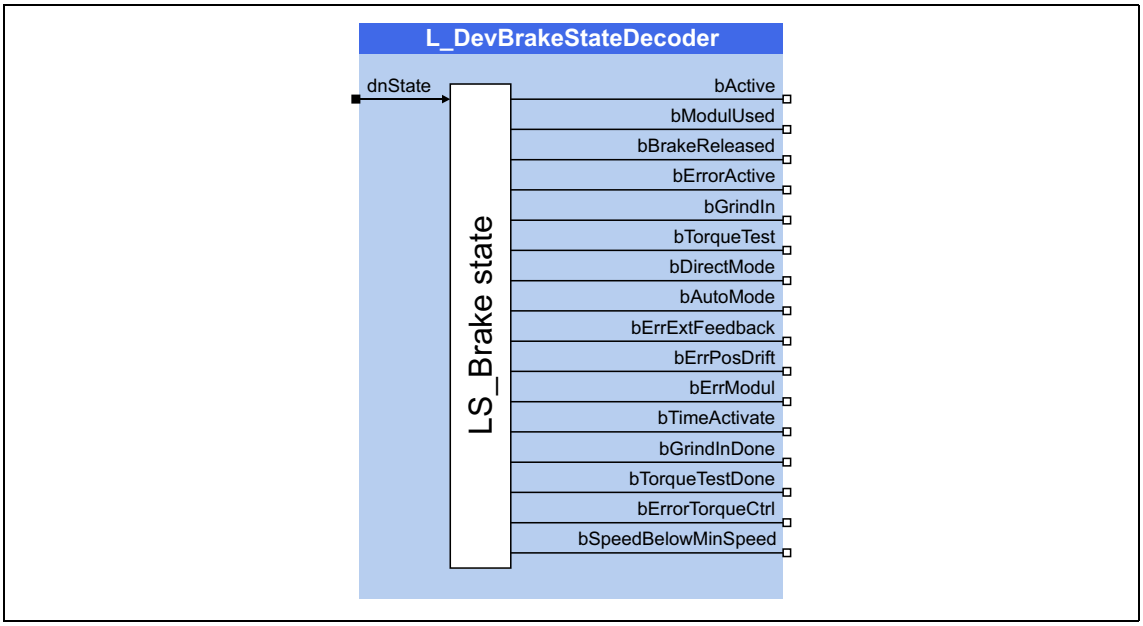
Identifier/data type	Value/meaning	
dnState DINT	Display of the error message last transmitted to the operating system.	
	1	Error message 1 is active.
	2	Error message 2 is active.
	4	Error message 3 is active.
	8	Error message 4 is active.
	16	Error message 5 is active.
	32	Error message 6 is active.
	64	Error message 7 is active.
	128	Error message 8 is active.
dnErrorNo DINT	Current error number	

5.66 L_DevBrakeStateDecoder - status signals of the basic function "Brake control"

Function library:	LenzeDevice9400
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB decodes the status output signal *dnState* of the SB **LS_Brake** into individual boolean status signals for further use in the FB interconnection.

- The functionality corresponds to a converter "DINT➞BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnState DINT	Input for the bit-coded status signal of the basic function "Brake control". • Connect this input with the output of the same name of the SB **LS_Brake**.

Outputs

The boolean outputs have the following meaning when the value is TRUE:

Bit*	Identifier	Meaning in case of TRUE
1	bActive	Brake control is active.
4	bModulUsed	Brake module is used.
8	bBrakeReleased	Brake status (internal status signal).
15	bErrorActive	Brake error is active (collective message).
16	bGrindIn	Status "Brake grinding-in".
17	bTorqueTest	Status "Brake test".
18	bDirectMode	Status "Direct control".
19	bAutoMode	Status "Automatic control".
20	bErrExtFeedback	Error: External feedback.
21	bErrPosDrift	Error: Position drift when brake is applied/during brake test.
22	bErrModul	Error: Brake module monitoring.
23	bTimeActivate	Information: Brake activation via waiting time.
24	bGrindInDone	Information: Brake grinding-in is completed.
25	bTorqueTestDone	Information: Brake test is completed.
26	bErrorTorqueCtrl	Error: feedforward control torque could not be established within one second.
27	bSpeedBelowMin-Speed	Information: Current speed has fallen below the threshold for the brake activation set in C02581.

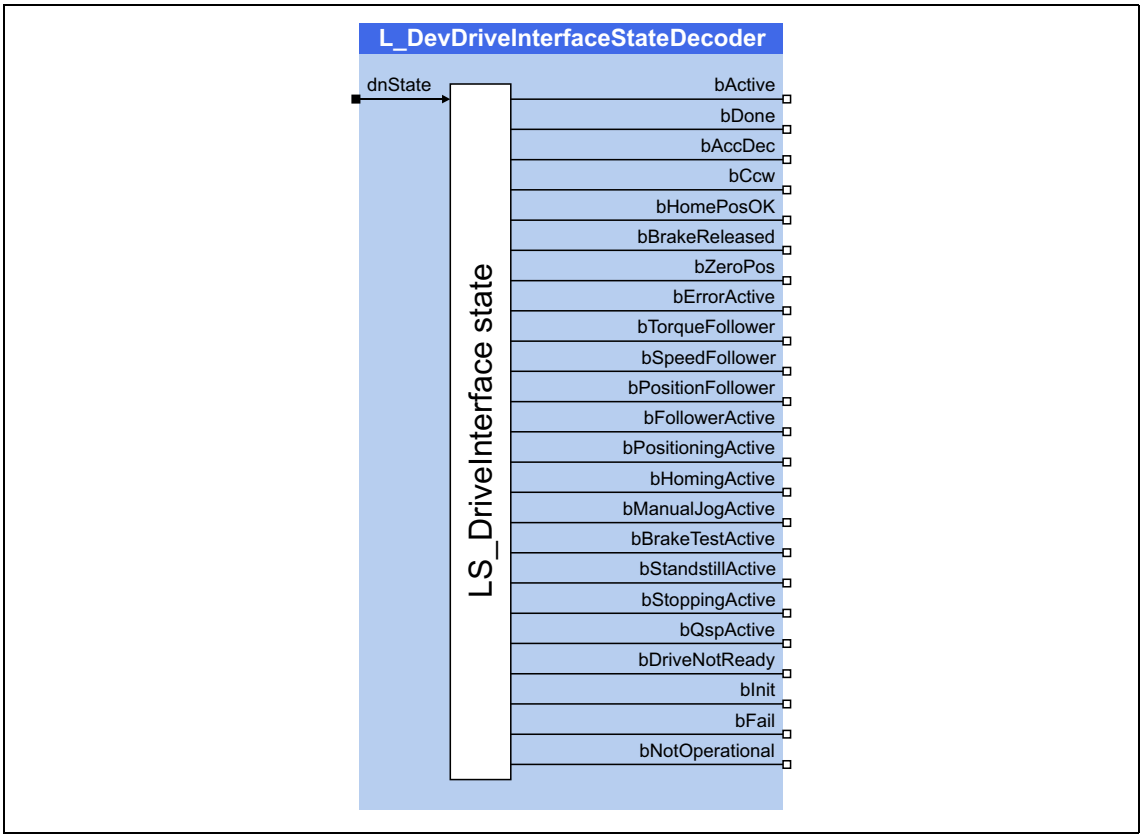
* Relating to the input signal *dnState*.

5.67 L_DevDriveInterfaceStateDecoder - status signals of the drive interface

Function library:	LenzeDevice9400
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB decodes the status output signal *dnState* of the SB **LS_DriveInterface** into individual boolean status signals for further use in the FB interconnection.

- The functionality corresponds to a converter "DINT➞BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnState DINT	Input for the bit-coded status signal of the drive interface. • Connect this input with the output of the same name of the SB **LS_DriveInterface**.

Outputs

The boolean outputs have the following meaning when the value is TRUE:

Bit*	Identifier	Meaning in case of TRUE
1	bActive	Basic function is active.
2	bDone	Basic function is completed.
3	bAccDec	Acceleration/deceleration phase is active.
5	bCcw	Counter-clockwise rotation is active.
7	bHomePosOK	Home position is known.
8	bBrakeReleased	Brake is released.
10	bZeroPos	Zero crossing detected or position = "0".
15	bErrorActive	Error is active.
16	bTorqueFollower	Torque follower is active.
17	bSpeedFollower	Speed follower is active.
18	bPositionFollower	Position follower is active.
19	bFollowerActive	Setpoint follower is active (group signal for bit 16 ...18.)
20	bPositioningActive	Positioning is active.
21	bHomingActive	Homing is active.
22	bManualJogActive	Manual jog is active.
23	bBrakeTestActive	Brake test is active.
24	bStandstillActive	Drive is at standstill.
25	bStoppingActive	Stopping (standard stop) is active.
26	bQspActive	Quick stop is active.
28	bDriveNotReady	Controller is not ready.
29	bInit	Controller is being initialised.
30	bFail	Error
31	bNotOperational	State machine for the basic functions is not ready.

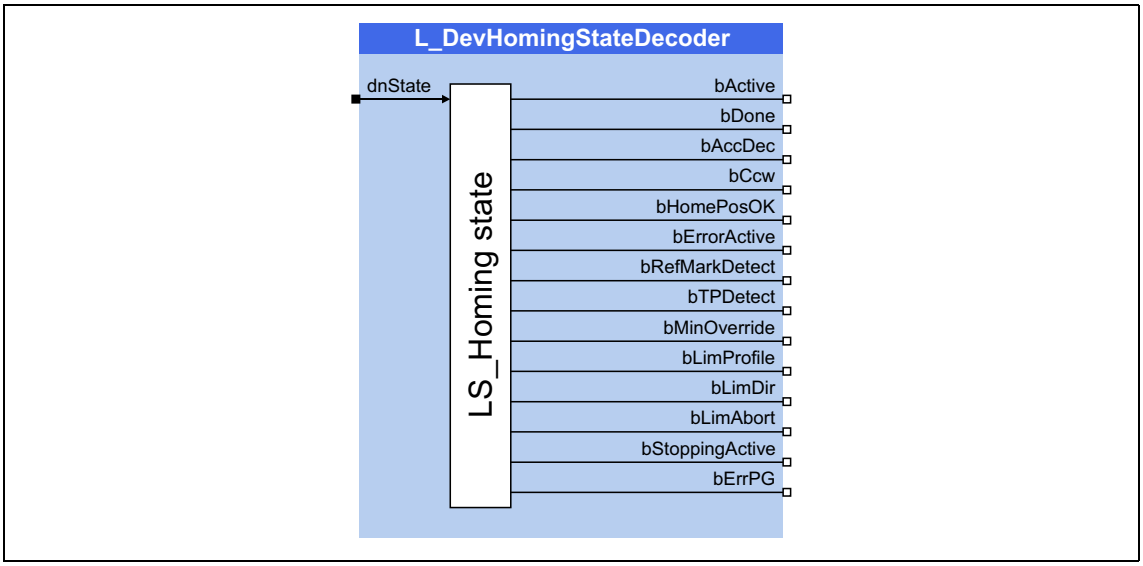
* Relating to the input signal *dnState*.

5.68 L_DevHomingStateDecoder - status signals of the basic function "Homing"

Function library:	LenzeDevice9400
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB decodes the status output signal *dnState* of the SB **LS_Homing** into individual boolean status signals for further use in the FB interconnection.

- The functionality corresponds to a converter "DINT➞BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnState DINT	Input for the bit-coded status signal of the basic function "Homing". • Connect this input with the output of the same name of the SB **LS_Homing**.

Outputs

The boolean outputs have the following meaning when the value is TRUE:

Bit*	Identifier	Meaning in case of TRUE
1	bActive	Homing is active.
2	bDone	Homing is completed.
3	bAccDec	Acceleration/deceleration phase is active.
5	bCcw	Counter-clockwise rotation is active.
7	bHomePosOK	Home position is known.
15	bErrorActive	Error is active.
16	bRefMarkDetect	Pre-stop (home switch) has been detected.
17	bTPDetect	Touch probe/zero pulse has been detected.
19	bMinOverride**	Override ≤ 1 %
21	bLimProfile	Profile data is limited by the basic function "Limiter" (SB **LS_Limiter**).
22	bLimDir	Direction is inhibited by the basic function "Limiter" (SB **LS_Limiter**).
23	bLimAbort	Abort by basic function "Limiter" (SB **LS_Limiter**).
25	bStoppingActive	Stopping (standard stop) is active.
30	bErrPG	Error during profile generation.

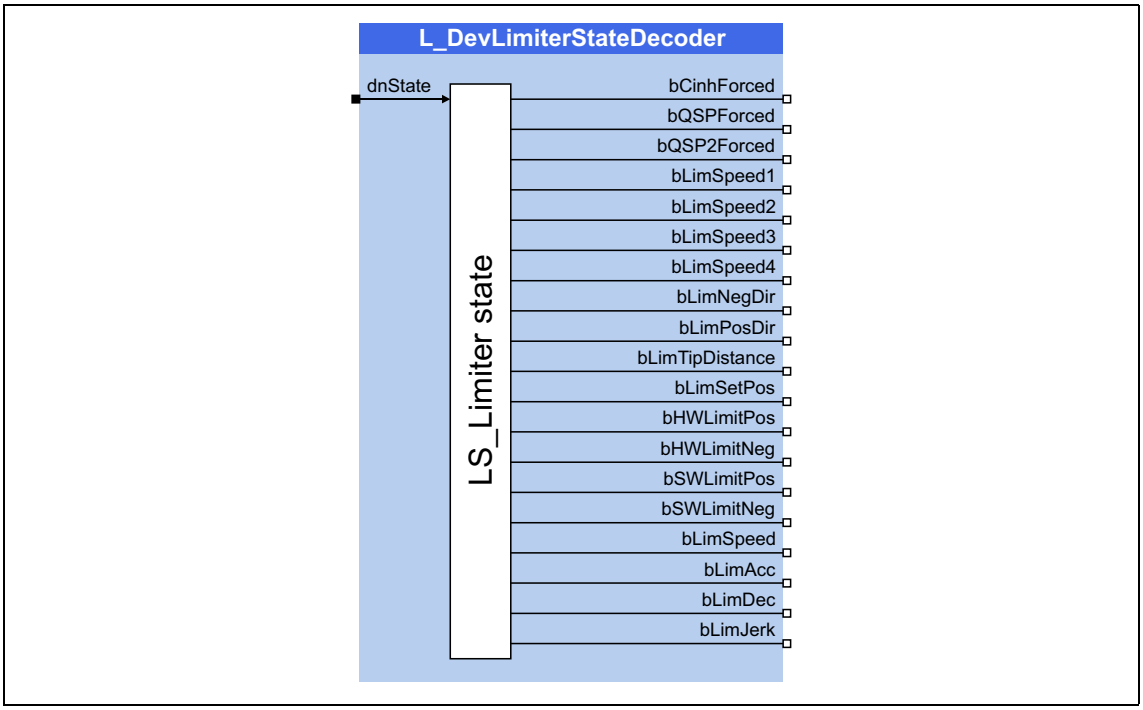
* With regard to the *dnState* input signal. ** As of library V02.04.xx.xx

5.69 L_DevLimiterStateDecoder - status signals of the basic function "Limiter"

Function library:	LenzeDevice9400
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB decodes the status output signal *dnState* of the SB **LS_Limiter** into individual boolean status signals for further use in the FB interconnection.

- The functionality corresponds to a converter "DINT➞BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnState DINT	Input for the bit-coded status signal of the basic function "Limiter". • Connect this input with the output of the same name of the SB **LS_Limiter**.

Outputs

The boolean outputs have the following meaning when the value is TRUE:

Bit*	Identifier	Meaning in case of TRUE
0	bCinhForced	Controller inhibit is initiated. (Safe torque off is requested.)
1	bQSPForced	Quick stop is initiated. (Safe stop 1 is requested.)
2	bQSP2Forced	Quick stop is initiated. (Safe stop 2 is requested.)
3	bLimSpeed1	Profile change due to speed limitation. (Limited speed 1 is requested.)
4	bLimSpeed2	Profile change due to speed limitation. (Limited speed 2 is requested.)
5	bLimSpeed3	Profile change due to speed limitation. (Limited speed 3 is requested.)
6	bLimSpeed4	Profile change due to speed limitation. (Limited speed 4 is requested.)
7	bLimNegDir	Only positive direction of rotation is permissible. • When the direction of rotation is negative while requesting "Only positive direction of rotation", the drive is braked to standstill.
8	bLimPosDir	Only negative direction of rotation is permissible. • When the direction of rotation is negative while requesting "Only negative direction of rotation", the drive is braked to standstill.
10	bLimTipDistance	Increment in manual jog mode is limited.
12	bLimSetPos	Limitation of the set position is active.
16	bHWLimitPos	Positive limit switch inhibits travel in positive direction.
17	bHWLimitNeg	Negative limit switch inhibits travel in positive direction.
18	bSWLimitPos	Positive software limit position inhibits travel in positive direction.
19	bSWLimitNeg	Negative software limit position inhibits travel in negative direction.
20	bLimSpeed	Limitation of speed is active.
21	bLimAcc	Limitation of acceleration is active.
22	bLimDec	Limitation of deceleration is active.
23	bLimJerk	Limitation of jerk is active (S-ramp time is increased).

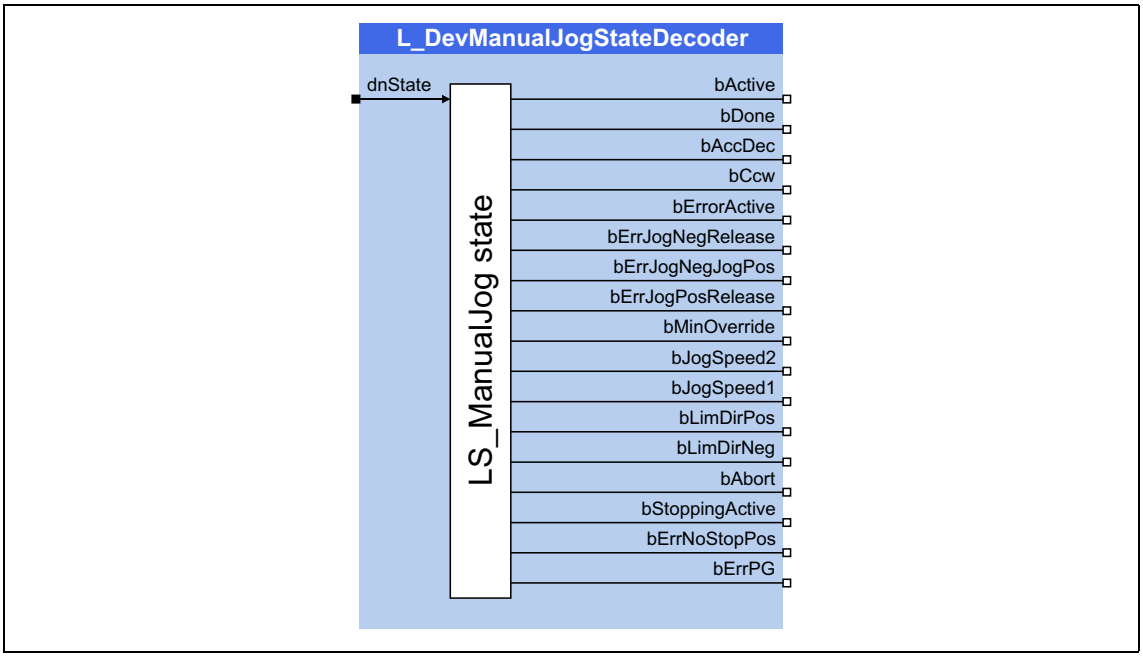
* Relating to the input signal *dnState*.

5.70 L_DevManualJogStateDecoder - status signals of the basic function "ManualJog"

Function library:	LenzeDevice9400
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB decodes the status output signal *dnState* of the SB **LS_ManualJog** into individual boolean status signals for further use in the FB interconnection.

- The functionality corresponds to a converter "DINT➞BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnState DINT	Input for the bit-coded status signal of the basic function "ManualJog". • Connect this input with the output of the same name of the SB **LS_ManualJog**.

Outputs

The boolean outputs have the following meaning when the value is TRUE:

Bit*	Identifier	Meaning in case of TRUE
1	bActive	Manual jog is active.
2	bDone	Manual jog is completed.
3	bAccDec	Acceleration/deceleration phase is active.
5	bCcw	Counter-clockwise rotation is active.
15	bErrorActive	Impermissible state (see bit 16, 17, 18, 22, 23).
16	bErrJogNegRelease	Stop by simultaneous selection of negative direction and retraction from limit switch.
17	bErrJogNegJogPos	Stop by simultaneous selection of positive and negative direction.
18	bErrJogPosRelease	Stop by simultaneous selection of positive direction and retraction from limit switch.
19	bMinOverride**	Override ≤ 1 %
20	bJogSpeed2	Manual jog speed 2 is active.
21	bJogSpeed1	Manual jog speed 1 is active.
22	bLimDirPos	Stop by selection of positive direction and simultaneous activation of the positive software limit position or the positive limit switch.
23	bLimDirNeg	Stop by selection of negative direction and simultaneous activation of the negative software limit position or the negative limit switch.
24	bAbort	General abort (ramping down of the speed setpoint) • E.g. when a direction initiator for manual jog is released or an impermissible state occurs (see bit 16, 17, 18, 22, 23).
25	bStoppingActive	Stopping is active.
27	bErrNoStopPos**	No stop position defined.
30	bErrPG	Error during profile generation.

* With regard to the *dnState* input signal. ** As of library V02.04.xx.xx

5.71 L_DevParReadFix - read parameter

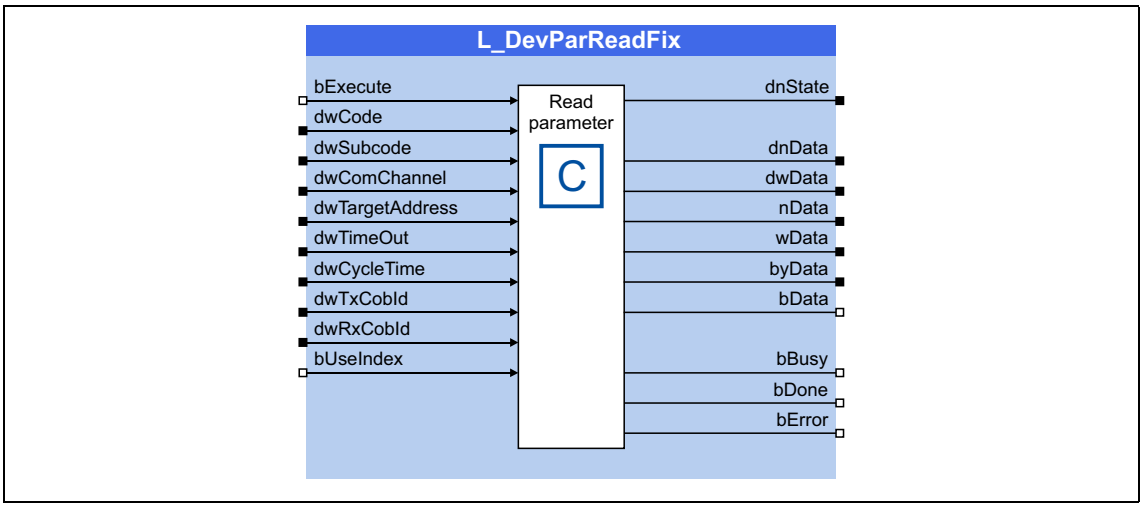
Function library:	LenzeDevice9400	**FB is available as of library V02.00.xx.xx! FB may only be used for firmware as of V1.50!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB reads a parameter and provides the read value to the application via the outputs *dnData* ... *bData*.

- All internal parameters (codes) of the controller can be read.
- From controller software version V08 and library V02.05.xx.xx, parameters of another node connected to the system bus (CAN) can also be read if a suitable SDO server channel has been parameterised at the remote node for this purpose.
  - The parameters of the remote node can be addressed by entering either the code or the CANopen index.
- In contrast to the L_DevReadParDInt FB, the parameter of this FB to be read is addressed via FB inputs and not via parameters.
- The FB supports both one-time and cyclic reading in an adjustable time interval.

Note!

The following applies to the 9400 ServoPLC: The FB must not be called with the same instance in several tasks.



Inputs

Identifier/data type		Information/possible settings	
bExecute	BOOL	Activates a read request.	
		FALSE↗TRUE	If the cycle time (*dwCycleTime*) = "0 ms" or the *dwCycleTime* input is not assigned: Read parameter value once which has been addressed via the inputs *dwCode*, *dwSubcode*, *dwComChannel* and *dwTargetAddress*.
			If cycle time (*dwCycleTime*) > "0 ms": Read parameter value cyclically which has been addressed via the inputs *dwCode*, *dwSubcode*, *dwComChannel* and *dwTargetAddress*.
		TRUE↘FALSE	Deactivate cyclic reading again.

Identifier/data type	Information/possible settings	
dwCode DWORD	• If *dwComChannel* = "1" or *bUseIndex* = FALSE: Code (0 ... 16000) • If *dwComChannel* = "2" or "3" and *bUseIndex* = TRUE: CANopen index (0x0 ... 0xFFFF)	
dwSubcode DWORD	Subcode (0 ... 255)	
dwComChannel DWORD	Interface • Up to and including controller software version V07, only the selection "1" (own device) is supported.	
	1	Own device
	2	CAN on-board
	3	CAN module
dwTargetAddress DWORD	This input does not have any function.	
dwTimeOut DWORD	Time-out (0 ... 10000 [ms]) • Time in which the read request must have been processed. • If the input is not assigned or if "0 s" is selected, time-out monitoring is deactivated.	
dwCycleTime DWORD	Cycle time (0 ... 100000 [ms]) • Interval for the cyclic reading of a parameter. • If an input is not assigned or if "0 ms" is selected, cyclic reading is deactivated.	
dwTxCobId DWORD From library V02.05.xx.xx	SDO client transmit identifier (0x0 ... 0x7FF) towards the remote CAN node • Identifier to be used for the communication with the remote node if *dwComChannel* = "2" or "3". • The identifier must correspond to an SDO server channel RX identifier (not to the basic SDO server channel 0). • If *dwComChannel* = "1", this input does not have any function.	
dwRxCobId DWORD From library V02.05.xx.xx	SDO client receive identifier (0x0 ... 0x7FF) from the remote CAN node • Identifier to be used for the communication with the remote node if *dwComChannel* = "2" or "3". • The identifier must correspond to an SDO server channel TX identifier (not to the basic SDO server channel 0). • If *dwComChannel* = "1", this input does not have any function.	
bUseIndex BOOL From library V02.05.xx.xx	Selection of the addressing of a parameter in the remote CAN node. • Selection only relevant if *dwComChannel* = "2" or "3".	
	FALSE	Value at *dwCode* is addressing a code (0 ... 16000).
	TRUE	Value at *dwCode* is addressing a CANopen index (0x0 ... 0xFFFF).

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit-coded) • Bits that are not listed have not been assigned with a status (always "0").	
	bits0	Read request has been sent to the operating system and the FB is waiting for a response from the device addressed (signal *bBusy*).
	bits1	Read request has been executed correctly (signal *bDone*).
	Bit 15	Error (group signal for bit 16 ...31).
	bit16	Configuration error: Invalid code.
	bits17	Configuration error: Invalid subcode.
	Bit 19	Configuration error: Invalid target address.
	Bit 20	Time-out error: The addressed device did not respond within the time-out time, the read request has been deleted.
	Bit 22	Error: Limit value has been exceeded.
	Bit 23	Error: Parameter reading is not permitted.
	Bit 24	Error: Access is not permitted.
	bits25	Error: No element of selection list.
	bit29	Error: Invalid identifier.
dnData DINT	Parameter set read (converted into different data types)	
dwData DWORD		
nData INT		
wData WORD		
byData BYTE		
bData BOOL		
bBusy BOOL	Status output "Read request transmitted"	
	TRUE	The read request has been sent to the operating system and the FB is waiting for a response from the device addressed.
	TRUE↘FALSE	The read request has been completed.
bDone BOOL	Status output "Read request executed correctly" • The output remains set until *bExecute* is reset to FALSE (however, at least for one task cycle).	
	TRUE	The read request has been executed correctly and the read parameter value is available at the *dnData* ... *bData* outputs.
bError BOOL	Status output "Error"	
	TRUE	An error has occurred during processing. • For details, please see the status output *dnState*.

Note!

The cycle time specified via the input *dwCycleTime* is only valid for the use of the FB in a 1-ms task! For a longer task cycle the cycle time is extended accordingly:

$$\text{Cycle time} = \frac{\text{dwCycleTime [ms]} \cdot \text{Task cycle [ms]}}{\text{1 [ms]}}$$

When the FB is used in the unsolicited task, the cycle time set is not effective. In this case, use a preceding FB L_TbOscillator instead to activate the read request cyclically:

- Connect the output *bOut* of the FB L_TbOscillator to the input *bExecute*.
- Set the desired cycle time via the parameters of the FB L_TbOscillator.

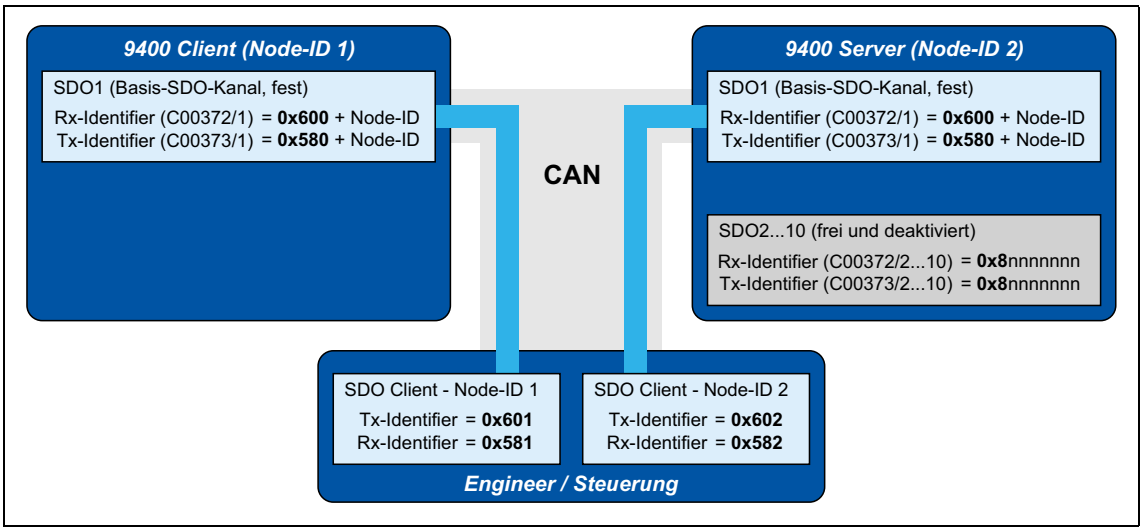
5.71.1 Example

The following example is intended to serve you as an installation guide for your own plant.

Task

The client, a 9400 controller, is supposed to read out code C00011 of the server, another 9400 controller, via a free SDO channel of the "CAN on board" system and provide the information to the application. At the same time, communication from the »Engineer« towards the client and the server is enabled via the basic SDO channel.

Initial situation



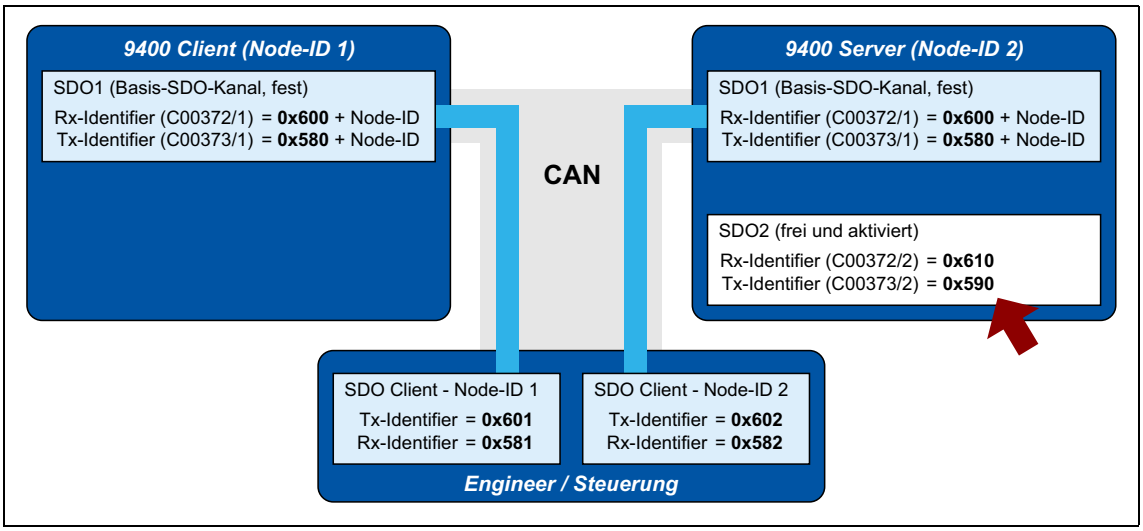
[5-34] Initial situation

- Parameter data channel 1 (the basic SDO channel) is always used for the communication towards the »Engineer«.
  - The basic SDO channel is permanently set to 0x600 + node number or 0x580 + node number, respectively, and can neither be changed nor deactivated according to CiA301.
- In the Lenze setting, the 9400 server has deactivated SDO server channels 2 ... 10 (bit 31 of the Rx/Tx identifiers has been set).

Step 1: Activate SDO server channel 2

To gain read access via the **L_DevParReadFix** FB, the remote node ("9400 server") must have a parameterised SDO server channel suitable for this purpose. SDO server channels 2 ... 10 can be set via codes C00372 and C00373 at the 9400 server.

- Subcodes 2 ... 10 stand for SDO server channels 2 ... 10.
- To activate SDO server channel 2, a receive identifier (RX identifier) must be entered in C00372/2 and a transmit identifier (TX identifier) must be entered in C00373/2 at the 9400 server:



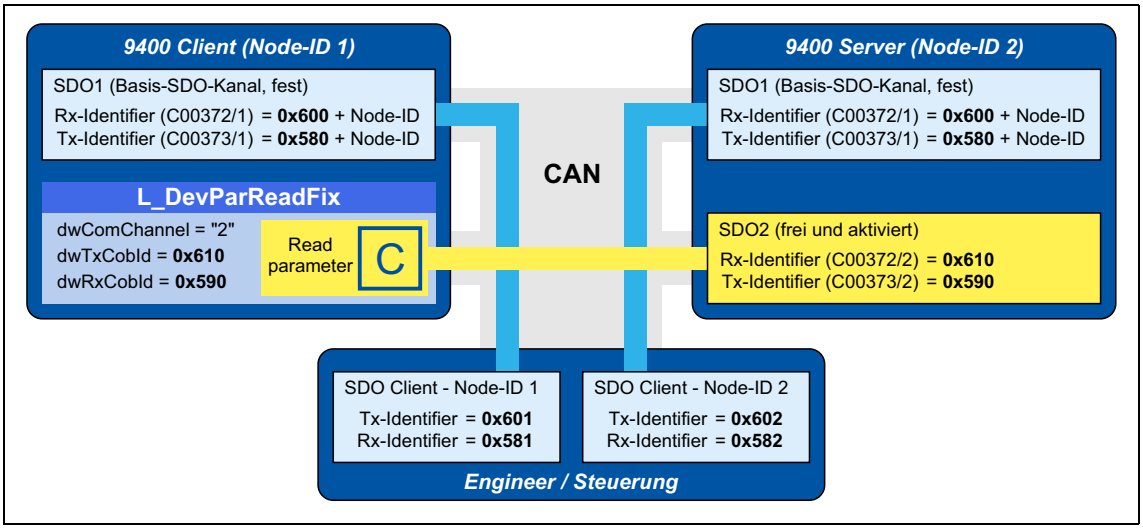
[5-35] Activation of parameter data channel 2 at the 9400 server

Now SDO server channel 2 which has been activated at the 9400 server can be used for communication via a client.

Step 2: Read access to the 9400 server

To gain read access to the parameters of the 9400 server, the **L_DevParReadFix** FB must be implemented into the application of the 9400 client.

- The receive identifier of the server (in our example: 0x610) must be assigned to the *dwTxCobId* input.
- The transmit identifier of the server (in our example: 0x590) must be assigned to the *dwRxCobId* input.
- A numerical value of "2" must be assigned to the *dwComChannel* input for the communication via "CAN on board" to be accomplished.



[5-36] Read access via parameter data channel 2 via the L_DevParReadFix FB

Due to the previously described measures, a connection has been established and the code addressed via inputs *dwCode* and *dwSubcode* can be queried in the remote server now.

Every time the block is executed via the *bExecute* input, the switching operation highlighted in yellow as shown in illustration [5-36] is carried out for a short time, the addressed code is queried, and a disconnection is made afterwards.

Write access with the FB L_DevParWriteFix is executed in the same way.

5.72 L_DevParWriteFix - write parameter

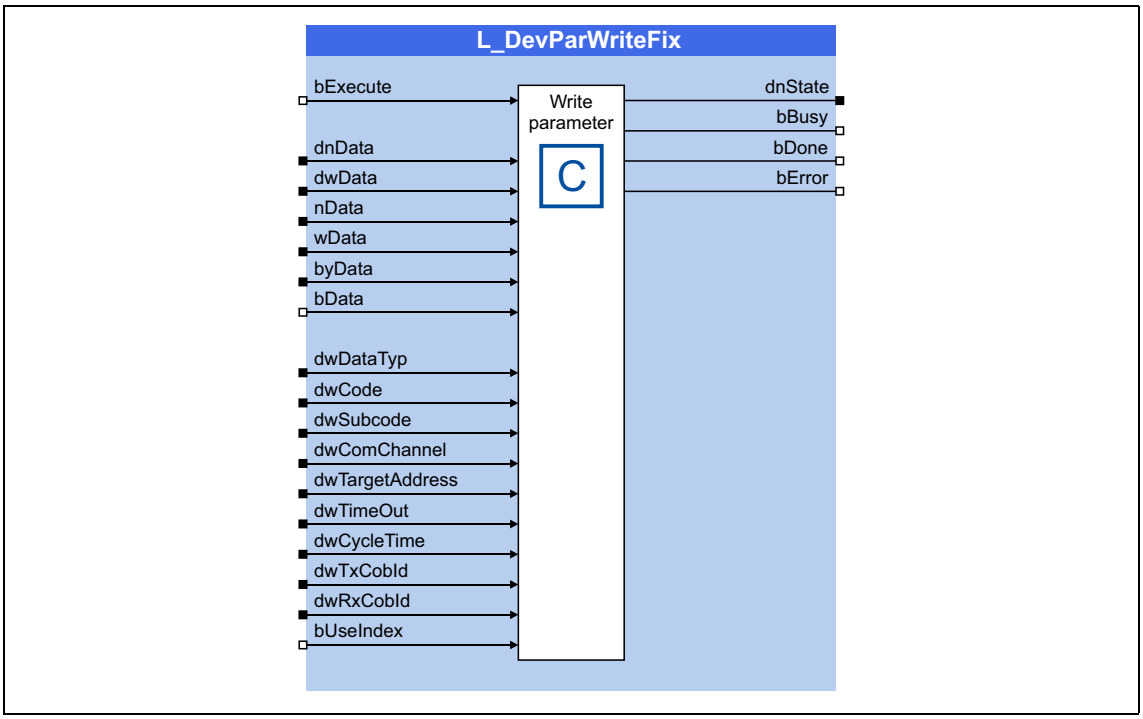
Function library:	LenzeDevice9400	**FB is available as of library V02.00.xx.xx! FB may only be used for firmware as of V1.50!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB writes a parameter with the value of the data input *dnData* ... *bData* which has been selected via the *dwDataTyp* input.

- All internal parameters (codes) of the controller can be written to.
- From controller software version V08 and library V02.05.xx.xx, parameters of another node connected to the system bus (CAN) can also be written to if a suitable SDO server channel has been parameterised at the remote node for this purpose.
  - The parameters of the remote node can be addressed by entering either the code or the CANopen index.
- In contrast to the L_DevWriteParDInt FB, the parameter of this FB to be written is addressed via FB inputs and not via parameters.
- The FB supports both one-time and cyclic writing in an adjustable time interval.

Note!

The following applies to the 9400 ServoPLC: The FB must not be called with the same instance in several tasks.



Inputs

Identifier/data type	Information/possible settings	
bExecute BOOL	Activates a write request.	
	FALSE↗TRUE	If the cycle time (*dwCycleTime*) = "0 ms" or the *dwCycleTime* input is not assigned: Write the parameter addressed via the inputs *dwCode*, *dwSubcode*, *dwComChannel* and *dwTargetAddress* once with the value of the data input which has been selected via *dwDataTyp*.
		If cycle time (*dwCycleTime*) > "0 ms": Write the parameter addressed via the inputs *dwCode*, *dwSubcode*, *dwComChannel* and *dwTargetAddress* cyclically with the value of the data input which has been selected via *dwDataTyp*.
	TRUE↘FALSE	Deactivate cyclic writing again.
dnData DINT	Parameter value to be written • The data input and thus the data type of the code to be written is selected via the *dwDataTyp* input.	
dwData DWORD		
nData INT		
wData WORD		
byData BYTE		
bData BOOL		
dwDataTyp DWORD	Selection of the data input / declaration of the data type	
	0	dnData data input / DINT data type
	1	dwData data input/ DWORD data type
	2	nData data input / INT data type
	3	wData data input / WORD data type
	4	byData data input/ BYTE data type
	5	bData data input/ BOOL data type
dwCode DWORD	• If *dwComChannel* = "1" or *bUseIndex* = FALSE: Code (0 ... 16000) • If *dwComChannel* = "2" or "3" and *bUseIndex* = TRUE: CANopen index (0x0 ... 0xFFFF)	
dwSubcode DWORD	Subcode (0 ... 255)	
dwComChannel DWORD	Interface • Up to and including controller software version V07, only the selection "1" (own device) is supported.	
	1	Own device
	2	CAN on-board
	3	CAN module
dwTargetAddress DWORD	This input does not have any function.	
dwTimeOut DWORD	Time-out (0 ... 10000 [ms]) • Time in which the write request must have been processed. • If the input is not assigned or if "0 s" is selected, time-out monitoring is deactivated.	

Identifier/data type		Information/possible settings
dwCycleTime DWORD		Cycle time (0 ... 100000 [ms]) • Interval for the cyclic writing of a parameter. • If an input is not assigned or if "0 ms" is selected, cyclic writing is deactivated.
dwTxCobId DWORD From library V02.05.xx.xx		SDO client transmit identifier (0x0 ... 0x7FF) towards the remote CAN node • Identifier to be used for the communication with the remote node if *dwComChannel* = "2" or "3". • The identifier must correspond to an SDO server channel RX identifier (not to the basic SDO server channel 0). • If *dwComChannel* = "1", this input does not have any function.
dwRxCobId DWORD From library V02.05.xx.xx		SDO client receive identifier (0x0 ... 0x7FF) from the remote CAN node • Identifier to be used for the communication with the remote node if *dwComChannel* = "2" or "3". • The identifier must correspond to an SDO server channel TX identifier (not to the basic SDO server channel 0). • If *dwComChannel* = "1", this input does not have any function.
bUseIndex BOOL From library V02.05.xx.xx		Selection of the addressing of a parameter in the remote CAN node. • Selection only relevant if *dwComChannel* = "2" or "3".
	FALSE	Value at *dwCode* is addressing a code (0 ... 16000).
	TRUE	Value at *dwCode* is addressing a CANopen index (0x0 ... 0xFFFF).

Outputs

Identifier/data type		Value/meaning
dnState DINT		Status (bit-coded) • Bits that are not listed have not been assigned with a status (always "0").
	bits0	Write request has been sent to the operating system and the FB is waiting for a response from the device addressed (signal *bBusy*).
	bits1	Write request has been executed correctly (signal *bDone*).
	Bit 15	Error (group signal for bit 16 ...31).
	bit16	Configuration error: Invalid code.
	bits17	Configuration error: Invalid subcode.
	Bit 19	Configuration error: Invalid target address.
	Bit 20	Time-out error: The addressed device did not respond within the time-out time, the write request has been deleted.
	Bit 21	Error: Invalid data type.
	Bit 22	Error: Limit value has been exceeded.
	Bit 24	Error: Access is not permitted.
	bits25	Error: No element of selection list.
	Bit 26	Error: Parameter writing is not permitted.
	bits27	Error: Parameter writing is only permitted when the controller is inhibited.
	Bit 28	Error: Parameter writing is only permitted when the application has stopped.
	bit29	Error: Invalid identifier.
bBusy BOOL		Status output "Write request transmitted"
	TRUE	The write request has been sent to the operating system and the FB is waiting for a response from the device addressed.
	TRUE↘FALSE	The write request has been completed.

Identifier/data type	Value/meaning	
bDone BOOL	Status output "Write request executed correctly" • The output remains set until *bExecute* is reset to FALSE (however, at least for one task cycle).	
	TRUE	The write request has been executed correctly.
bError BOOL	Status output "Error"	
	TRUE	An error has occurred during processing. • For details, please see the status output *dnState*.

Note!

The cycle time specified via the input *dwCycleTime* is only valid for the use of the FB in a 1-ms task! For a longer task cycle the cycle time is extended accordingly:

$$\text{Cycle time} = \frac{\text{dwCycleTime [ms]} \cdot \text{Task cycle [ms]}}{1\ \text{[ms]}}$$

When the FB is used in the unsolicited task, the cycle time set is not effective. In this case, use a preceding FB L_TbOscillator instead to activate the write request cyclically:

- Connect the output *bOut* of the FB L_TbOscillator to the input *bExecute*.
- Set the desired cycle time via the parameters of the FB L_TbOscillator.

Tip!

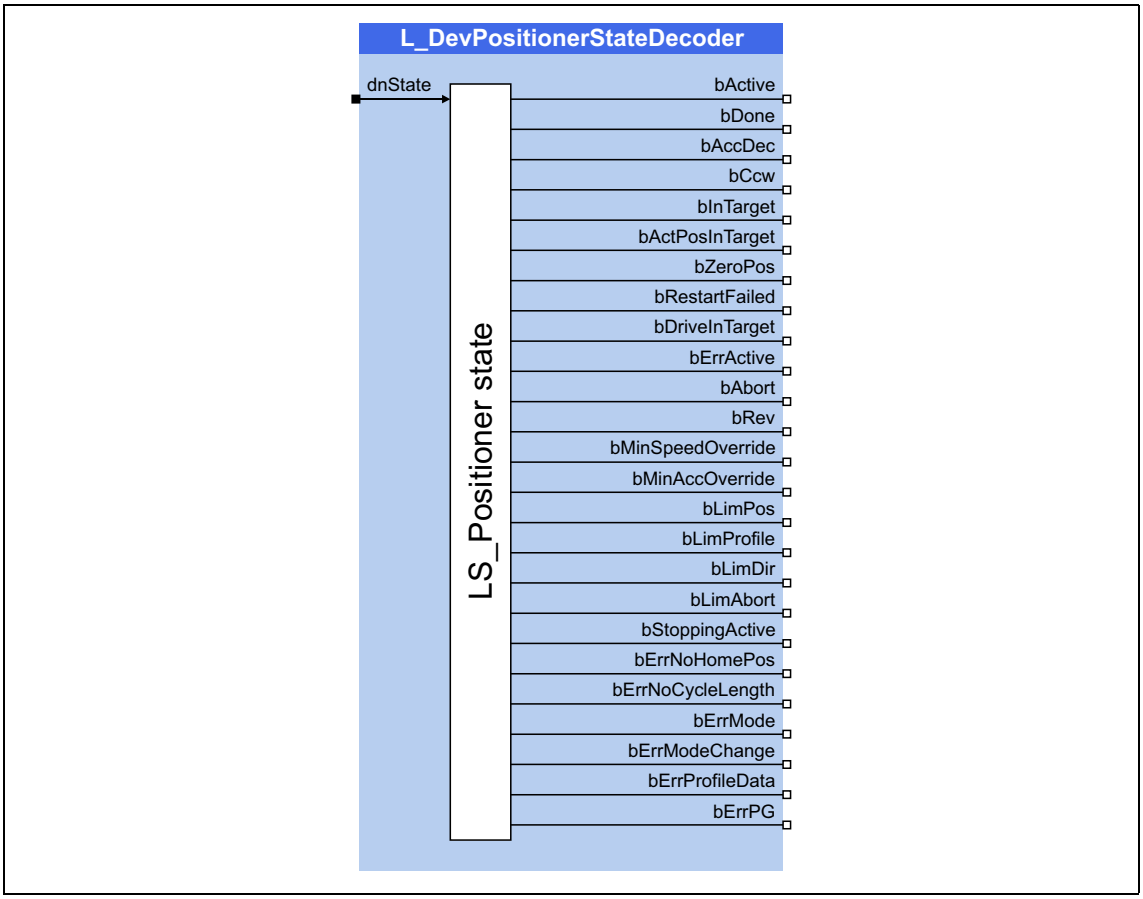
The description of FB L_DevParReadFix provides an application example.

5.73 L_DevPositionerStateDecoder - positioning status signals

Function library:	LenzeDevice9400	**FB available from library V02.00.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB decodes the *dnState* status output signal of the SB **LS_Positioner** into individual boolean status signals for further use in the function block interconnection.

- The functionality corresponds to a converter "DINT➞BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnState DINT	Input for the bit-coded status signal of the basic function "Positioning". • Connect this input with the output of the same name of the SB **LS_Positioner**.

Outputs

The boolean outputs have the following meaning when the value is TRUE:

Bit*	Identifier	Meaning in case of TRUE
1	bActive	Positioning is active.
2	bDone	Positioning is completed (all profiles have been processed).
3	bAccDec	Acceleration/deceleration phase is active.
4	bActPosInTarget	Actual position in the target • The actual position value of the drive has reached the target position of the profile to be traversed last within the tolerance window set in C02670.
5	bCcw	Counter-clockwise rotation is active.
6	bInTarget	Set position reached (in case of sequence profiles the drive continues to travel).
10	bZeroPos	Zero crossing in the "Modulo" positioning mode.
11	bRestartFailed	Positioning cannot be continued. • At the SB **LS_Positioner**, *POS_bRestart* is selected **AND** • *POS_bAbort* is set simultaneously **OR** • positioning is completed **OR** • a new positioning has been started simultaneously.
12	bDriveInTarget**	Drive in the target (actual position and set position in the target).
15	bErrActive	Error in basic function active (group signal).
16	bAbort	Positioning is aborted.
17	bRev	Reversing phase is active.
18	bMinSpeedOverride	Speed override ≤1 %
19	bMinAccOverride	Acceleration or deceleration override ≤1 %
20	bLimPos	Position is limited by the basic function "Limiter" (SB **LS_Limiter**).
21	bLimProfile	Profile data is limited by the basic function "Limiter" (SB **LS_Limiter**).
22	bLimDir	Direction is inhibited by the basic function "Limiter" (SB **LS_Limiter**).
23	bLimAbort	Abort by basic function "Limiter" (SB **LS_Limiter**).
24	bErrNoHomePos	Home position is not known.
25	bStoppingActive	Stopping is active. • Basic function is enabled for the first time but no positioning requested/active.
26	bErrNoCycleLength	Cycle is not known.
27	bErrMode	Invalid positioning mode.
28	bErrModeChange	Invalid change of positioning mode.
29	bErrProfileData	The profile data is not plausible or faulty: • No valid profile data structure is available (no signal connected to FB). • Parameters for acceleration/deceleration and/or speed = 0 • Override for acceleration/deceleration and/or speed = 0 • Set position > displayable value range • Position > cycle (in Modulo measuring systems)
30	bErrPG	Error during profile generation. • An internal error has occurred while calculating in the profile generator (group signal).

* With regard to the *dnState* input signal. ** As of library V02.04.xx.xx and controller software version V5.0

5.74 L_DevReadParDInt - reading parameters of the DINT type

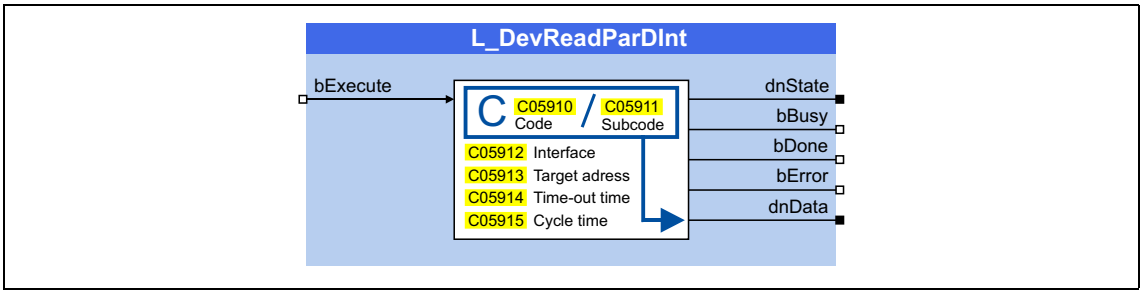
Function library:	LenzeDevice9400	**FB is available from library V02.00.xx.xx onwards! FB may only be used for firmware from V1.50 onwards!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB reads parameters of data type "DINT" and outputs the read value via the output *dnData* for the application.

- Only internal parameters (codes) of the controller can be read.
- In contrast to the L_DevParReadFix FB, the parameter of this FB to be read is addressed via parameters and not via FB inputs.
- The FB supports both one-time and cyclic reading in a parameterisable time interval.
- To read a parameter with another data type than "DINT", use the FB L_DevParReadFix. (📖 265)

Note!

The following applies to the 9400 ServoPLC: The FB may not be called with the same instance in several tasks.



Inputs

Identifier/data type		Information/possible settings	
bExecute	BOOL	Activate a read request.	
		FALSE↗TRUE	If cycle time (C05915) = "0 s": Read parameter value once which has been addressed via C05910 ...C05913.
			If cycle time (C05915) > "0 s": Read parameter value cyclically which has been addressed via C05910 ...C05913.
		TRUE↘FALSE	Deactivate cyclic reading again.

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded) • Bits that are not listed have not been assigned with a status (always "0").	
	Bit 0	Read request has been sent to the operating system and the FB is waiting for a response from the device addressed (signal *bBusy*).
	Bit 1	Read request has been executed correctly (signal *bDone*).
	Bit 15	Error (group signal for bit 16 ...31).
	Bit 16	Configuration error: Invalid code.
	Bit 17	Configuration error: Invalid subcode.
	Bit 19	Configuration error: Invalid target address.
	Bit 20	Time-out error: The addressed device did not respond within the time-out time, the read request has been deleted.
	Bit 22	Error: Limit value has been exceeded.
	Bit 23	Error: Parameter reading is not permitted.
	Bit 24	Error: Access is not permitted.
	Bit 25	Error: No element of selection list.
bBusy BOOL	Status output "Read request transmitted"	
	TRUE	The read request has been sent to the operating system and the FB is waiting for a response from the device addressed.
	TRUE↘FALSE	The read request has been completed.
bDone BOOL	Status output "Read request executed correctly" • The output remains set until *bExecute* is reset to FALSE (however, at least for one task cycle).	
	TRUE	The read request has been executed correctly and the read parameter value is available at the output *dnData*.
bError BOOL	Status output "Error"	
	TRUE	An error has occurred during processing. • For details, please see the status output *dnState*.
dnData DINT	Parameter value read	

Parameters

Parameter	Possible settings			Info
C05910	0		12000	Code • Initialisation: 2
C05911	0		255	Subcode • Initialisation: 0
C05912				Interface • Currently, only the Lenze setting "Own device" is supported.
	1	Own device		
	2	CAN on-board		
	3	Module in MXI1		
	4	Module in MXI2		
C05913	0		128	Target address • Selection of parameter channel for target device. • Setting is only required if an external parameter is to be read (C05912 > "1").

Parameter	Possible settings			Info
C05914	0.000	s	10.000	Time-out time • Time in which the read request must have been processed. • Initialisation: 0.000 s
C05915	0.000	s	100.000	Cycle time • Interval for the cyclic reading of a parameter. • Initialisation: 0.000 s (cyclic reading is deactivated)

Note!

The cycle time set in C05915 applies only to the use of the FB in a 1-ms task! For a longer task cycle the cycle time is prolonged accordingly:

$$\text{Cycle time} = \frac{\text{C5915 [s]} \cdot \text{Task cycle [ms]}}{\text{1 [ms]}}$$

If the FB is used in the unsolicited task, the cycle time set is not effective. In this case use an upstream FB L_TbOscillator instead, in order to cyclically trigger the read request:

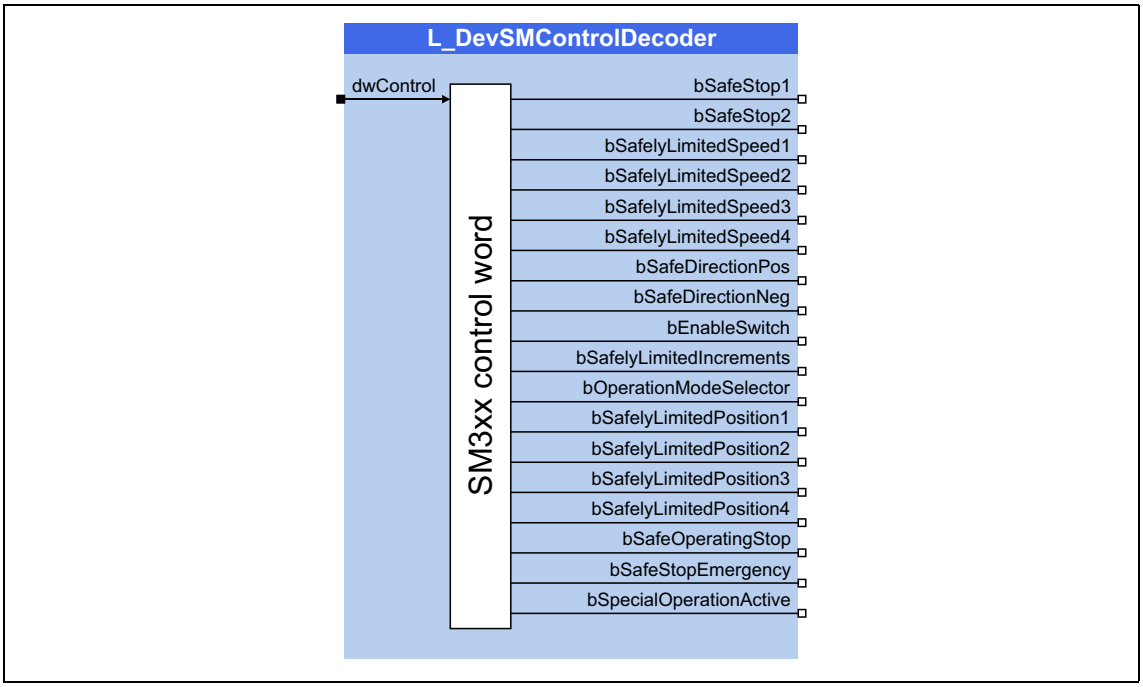
- Connect the output *bOut* of FB L_TbOscillator with the input *bExecute*.
- Set the desired cycle time via the parameters of the FB L_TbOscillator.

5.75 L_DevSMControlDecoder - control signals from the safety module

Function library:	LenzeDevice9400	**FB is available as of library V02.00.xx.xx! FB may only be used for firmware as of V1.50!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

The SM3xx safety module uses the bit-coded control signal *SM_dwControl* of the SB **LS_SafetyModuleInterface** to transmit information about requested or active safety functions to the application. The FB decodes the control signal into individual boolean control signals for further use in the FB interconnection.

- The corresponding actions (e.g. braking, braking to standstill, holding of the standstill position) must be executed by the application, e.g. via the basic function "Limiter".
- Several safety functions can be requested/be active at the same time.
- The safety functions that are supported depend on the safety module used.
- The FB functionality corresponds to a "DWORD→BOOL" converter, but the boolean outputs of the FB are called according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dwControl DWORD	Input for the bit-coded control signal from the safety module. • Connect this input with the output *SM_dwControl* of the SB **LS_SafetyModuleInterface**.

Outputs

Bit*	Identifier	Requested/active safety functions with TRUE
1	bSafeStop1	Safe stop 1 (SS1)
2	bSafeStop2	Safe stop 2 (SS2)
* Relating to the input signal *dwControl*.		

Bit*	Identifier	Requested/active safety functions with TRUE
3	bSafelyLimitedSpeed1	Safely limited speed 1 (SLS1)
4	bSafelyLimitedSpeed2	Safely limited speed 2 (SLS2)
5	bSafelyLimitedSpeed3	Safely limited speed 3 (SLS3)
6	bSafelyLimitedSpeed4	Safely limited speed 4 (SLS4)
7	bSafeDirectionPos	Safe positive direction of rotation (SDIp)
8	bSafeDirectionNeg	Safe negative direction of rotation (SDIn)
9	bSafeEnablingSwitch	Safe enable switch (ES)
10	bSafelyLimitedIncrements	Safely limited increments (SLI)
11	bOperationModeSelector	Safe operation mode selector (OMS)
12	bSafelyLimitedPosition1	Safely limited position 1 (SLP1)
13	bSafelyLimitedPosition2	Safely limited position 2 (SLP2)
14	bSafelyLimitedPosition3	Safely limited position 3 (SLP3)
15	bSafelyLimitedPosition4	Safely limited position 4 (SLP4)
16	bSafeOperatingStop	Safe operating stop (SOS)
23	bSafeStopEmergency	Emergency stop (SSE)
29	bSpecialOperationActive	Special operation is active

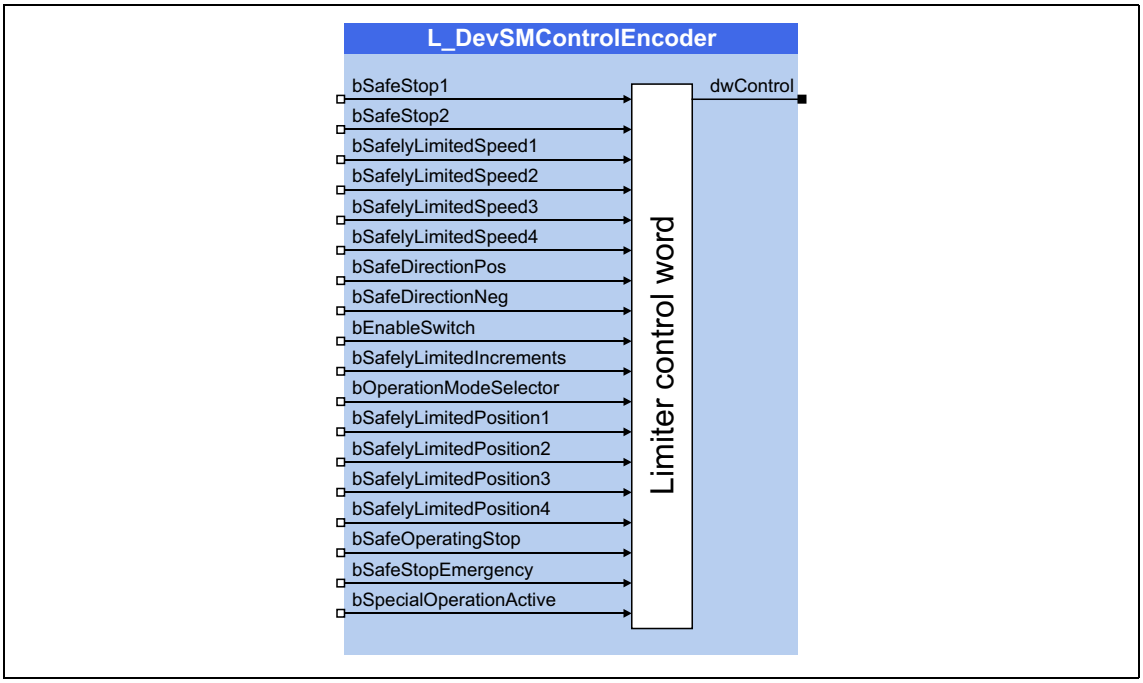
* Relating to the input signal *dwControl*.

5.76 L_DevSMControlEncoder - control word for limiter

Function library:	LenzeDevice9400	**FB is available as of library V02.00.xx.xx! FB may only be used for firmware as of V1.50!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB generates the bit-coded control signal *dwControl* from individual boolean control signals for the SB **LS_Limiter** and thus allows control of the basic function "Limiter" if no SM3xx safety module is available.

- The FB functionality corresponds to a "BOOL→DWORD" converter, but the boolean inputs of the FB are called according to their meaning.



Inputs

Bit*	Identifier	Requested/active safety functions with TRUE
1	bSafeStop1	Safe stop 1 (SS1)
2	bSafeStop2	Safe stop 2 (SS2)
3	bSafelyLimitedSpeed1	Safely limited speed 1 (SLS1)
4	bSafelyLimitedSpeed2	Safely limited speed 2 (SLS2)
5	bSafelyLimitedSpeed3	Safely limited speed 3 (SLS3)
6	bSafelyLimitedSpeed4	Safely limited speed 4 (SLS4)
7	bSafeDirectionPos	Safe positive direction of rotation (SDIp)
8	bSafeDirectionNeg	Safe negative direction of rotation (SDIn)
9	bSafeEnablingSwitch	Safe enable switch (ES)
10	bSafelyLimitedIncrements	Safely limited increments (SLI)
11	bOperationModeSelector	Safe operation mode selector (OMS)
12	bSafelyLimitedPosition1	Safely limited position 1 (SLP1)
13	bSafelyLimitedPosition2	Safely limited position 2 (SLP2)

* Relating to the output signal *dwControl*.

Bit*	Identifier	Requested/active safety functions with TRUE
14	bSafelyLimitedPosition3	Safely limited position 3 (SLP3)
15	bSafelyLimitedPosition4	Safely limited position 4 (SLP4)
16	bSafeOperatingStop	Safe operating stop (SOS)
23	bSafeStopEmergency	Emergency stop (SSE)
29	bSpecialOperationActive	Special operation is active

* Relating to the output signal *dwControl*.

Outputs

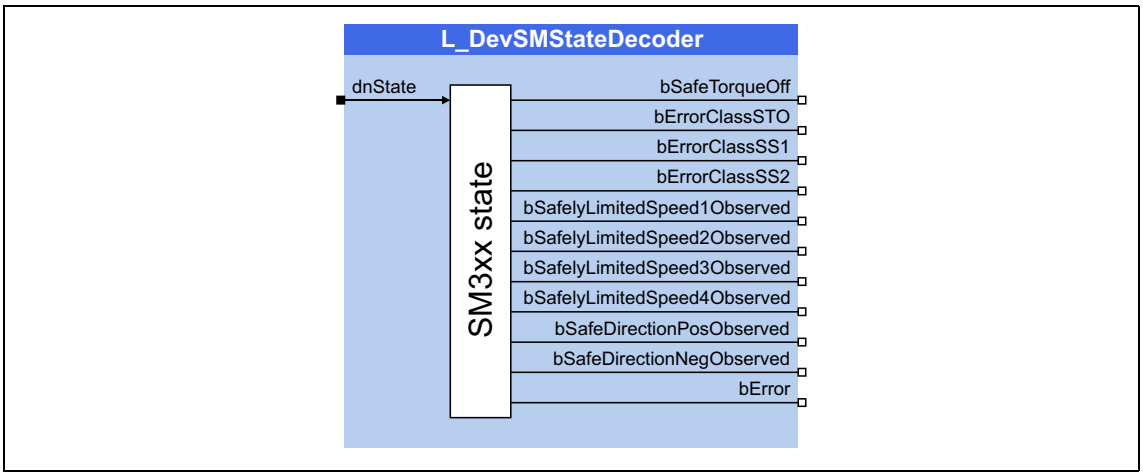
Identifier/data type	Information/possible settings
dwControl DWORD	Bit-coded control signal for the basic function "Limiter". • Connect this output with the input *LIM_dwControl* of the SB **LS_Limiter**.

5.77 L_DevSMStateDecoder - status signals from the safety module

Function library:	LenzeDevice9400	**FB is available as of library V02.00.xx.xx! FB may only be used for firmware as of V1.50!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

The SM3xx safety module uses the bit-coded status signal *SM_dnState* of the SB **LS_SafetyModuleInterface** to transmit the status of the safety functions to the application. The FB decodes the status signal into individual boolean status signals for further use in the FB interconnection.

- The FB functionality corresponds to a converter "DINT➞BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnState DINT	Input for the bit-coded status signal from the safety module. • Connect this input with the output *SM_dnState* of the SB **LS_SafetyModuleInterface**.

Outputs

Bit*	Identifier	Meaning in case of TRUE
0	bSafeTorqueOff	Safe torque off (STO) is active as normal stop.
3	bErrorClassSTO	Safe torque off (STO) is active as error stop.
4	bErrorClassSS1	Safe stop 1 (SS1) is active as error stop.
5	bErrorClassSS2	Safe stop 2 (SS2) is active as error stop.
8	bSafelyLimitedSpeed1Observed	Safely limited speed 1 (SLS1) is activated and observed.
9	bSafelyLimitedSpeed2Observed	Safely limited speed 2 (SLS2) is activated and observed.
10	bSafelyLimitedSpeed3Observed	Safely limited speed 3 (SLS3) is activated and observed.
11	bSafelyLimitedSpeed4Observed	Safely limited speed 4 (SLS4) is activated and observed.
12	bSafeDirectionPosObserved	Safe positive direction of rotation (SDIp) is activated and observed.
13	bSafeDirectionNegObserved	Safe negative direction of rotation (SDIn) is activated and observed.
14	bError	SM301 safety module is in an error state (Trouble or Warning).

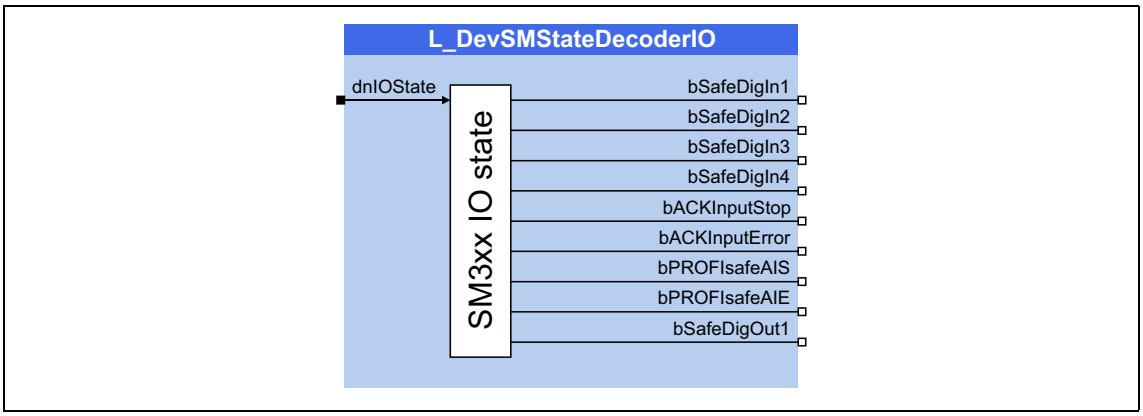
* Relating to the input signal *dnState*.

5.78 L_DevSMStateDecoderIO - status signals from the safety module

Function library:	LenzeDevice9400	**FB is available as of library V02.00.xx.xx! FB may only be used for firmware as of V1.50!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

The SM3xx safety module uses the bit-coded status signal *SM_dnIOState* of the SB **LS_SafetyModuleInterface** to transmit the status of the safe inputs and the safe output to the application. The FB decodes the status signal into individual boolean status signals for further use in the FB interconnection.

- The FB functionality corresponds to a converter "DINT➞BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnIOState DINT	Input for the bit-coded I/O status signal from the safety module. • Connect this input with the output *SM_dnIOState* of the SB **LS_SafetyModuleInterface**.

Outputs

Bit*	Identifier	Meaning in case of TRUE
0	bSafeDigIn1	Sensor input 1 in ON state.
1	bSafeDigIn2	Sensor input 2 in ON state.
2	bSafeDigIn3	Sensor input 3 in ON state.
3	bSafeDigIn4	Sensor input 4 in ON state.
5	bACKInputStop	Restart acknowledgement AIS via terminal (with TRUE↘FALSE).
6	bACKInputError	Error acknowledgement AIE via terminal (with TRUE↘FALSE).
8	bPROFIsafeAIS	Restart acknowledgement AIS via safety bus.
9	bPROFIsafeAIE	Error acknowledgement AIE via safety bus.
12	bSafeDigOut1	Safe output 1 (feedback output) in ON state.

* Relating to the input signal *dnIOState*.

5.79 L_DevWriteParDInt - writing to parameters of the DINT type

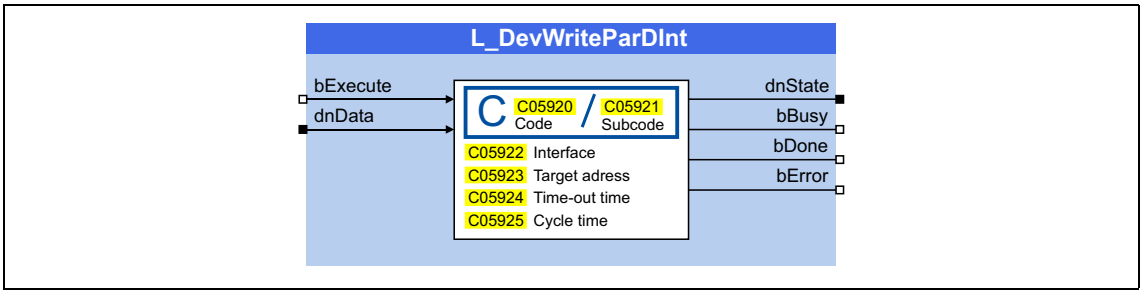
Function library:	LenzeDevice9400	**FB is available from library V02.00.xx.xx onwards! FB may only be used for firmware from V1.50 onwards!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB writes a parameter of data type "DINT" with the value assigned to the input *dnData*.

- Only internal parameters (codes) of the controller can be written.
- In contrast to the L_DevParWriteFix FB, the parameter of this FB to be written is addressed via parameters and not via FB inputs.
- The FB supports both one-time and cyclic writing in a parameterisable time interval.
- To write a different data type as "DINT" to a parameter, the FB L_DevParWriteFix is to be used instead. (📖 271)

Note!

The following applies to the 9400 ServoPLC: The FB may not be called with the same instance in several tasks.



Inputs

Identifier/data type		Information/possible settings	
bExecute	BOOL	Activate a write request.	
		FALSE↗TRUE	If cycle time (C05925) = "0 s": Write the parameter addressed via C05920 ...C05923 once with the value assigned to the *dnData* input.
			If cycle time (C05925) > "0 s": Write the parameter addressed via C05920 ...C05923 cyclically with the value assigned to the *dnData* input.
		TRUE↘FALSE	Deactivate cyclic writing again.

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded) • Bits that are not listed have not been assigned with a status (always "0").	
	Bit 0	Write request has been sent to the operating system and the FB is waiting for a response from the device addressed (signal *bBusy*).
	Bit 1	Write request has been executed correctly (signal *bDone*).
	Bit 15	Error (group signal for bit 16 ...31).
	Bit 16	Configuration error: Invalid code.
	Bit 17	Configuration error: Invalid subcode.
	Bit 19	Configuration error: Invalid target address.
	Bit 20	Time-out error: The addressed device did not respond within the time-out time, the write request has been deleted.
	Bit 21	Error: Invalid data type. • Use the table of attributes to check the data type of the parameter you want to write. If it is not a DINT data type, use the FB L_DevParWriteFix instead for writing the parameter.
	Bit 22	Error: Limit value has been exceeded.
	Bit 24	Error: Access is not permitted.
	Bit 25	Error: No element of selection list.
	Bit 26	Error: Parameter writing is not permitted.
	Bit 27	Error: Parameter writing is only permitted when the controller is inhibited.
	Bit 28	Error: Parameter writing is only permitted when the application has stopped.
bBusy BOOL	Status output "Write request transmitted"	
	TRUE	The write request has been sent to the operating system and the FB is waiting for a response from the device addressed.
	TRUE↘FALSE	The write request has been completed.
bDone BOOL	Status output "Write request executed correctly" • The output remains set until *bExecute* is reset to FALSE (however, at least for one task cycle).	
	TRUE	The write request has been executed correctly.
bError BOOL	Status output "Error"	
	TRUE	An error has occurred during processing. • For details, please see the status output *dnState*.

Parameters

Parameter	Possible settings			Info
C05920	0		12000	Code • Initialisation: 2
C05921	0		255	Subcode • Initialisation: 0
C05922				Interface • Currently only the Lenze setting "Own device" is supported.
	1	Own device		
	2	CAN on-board		
	3	Module in MXI1		
	4	Module in MXI2		

Parameter	Possible settings			Info
C05923	0		128	Target address • Selection of parameter channel for target device. • Setting is only required if an external parameter is to be written (C05922 > "1").
C05924	0.000	s	10.000	Time-out time • Time in which the write request must have been processed. • Initialisation: 0.000 s
C05925	0.000	s	100.000	Cycle time • Interval for the cyclic writing of a parameter. • Initialisation: 0.000 s (cyclic writing is deactivated)



Note!

The cycle time set in C05925 applies only to the use of the FB in a 1-ms task! For a longer task cycle the cycle time is prolonged accordingly:

$$\text{Cycle time} = \frac{\text{C5925 [s]} \cdot \text{Task cycle [ms]}}{\text{1 [ms]}}$$

If the FB is used in the unsolicited task, the cycle time set is not effective. In this case use an upstream FB L_TbOscillator instead, in order to cyclically trigger the write request:

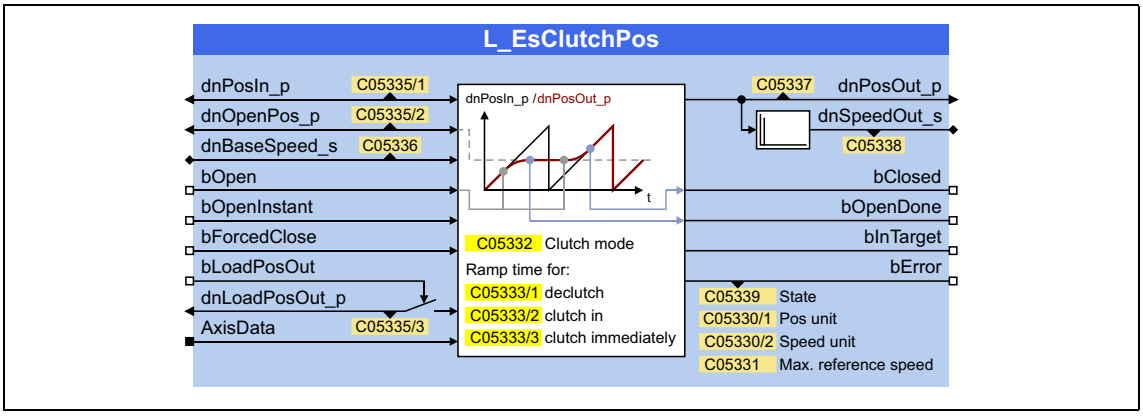
- Connect the output *bOut* of the FB L_TbOscillator with the input *bExecute*.
- Set the desired cycle time via the parameters of the FB L_TbOscillator.

5.80 L_EsClutchPos - clutch (time-controlled)

Function library:	LenzeElectricalShaft	**FB is available as of library V01.01!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB serves to implement a time-controlled clutching and declutching of the master position for all traversing ranges (Modulo/limited/unlimited) via a fifth-order polynomial.

- The FB supports the following clutch modes:
  - Declutching until standstill
  - Declutching to a basic speed
- More functions:
  - Positive opening operation of the clutch (for emergency situations)
  - "Hard clutch-in"
  - "Hard declutch"
- For path-controlled clutching and declutching, use the L_CamClutchPos FB instead. (📖 135)



Inputs

Identifier/data type	Information/possible settings	
dnPosIn_p DINT	Input position in [increments] • Selection of an external master position. • C05335/3 indicates the input position in the real unit of the machine.	
dnOpenPos_p DINT	Standstill position after the declutch process in [increments] • C05335/2 indicates the standstill position in the real unit of the machine.	
dnBaseSpeed_s DINT	Basic speed as speed in [rpm] • For clutch mode C05332 = "1: Clutch-in to basic speed". • C05336 indicates the basic speed in the real unit of the machine.	
bOpen BOOL	Control of the clutch function • The behaviour depends on the clutch-in mode set in C05332.	
	TRUE	"Open clutch" request (declutch process). • The declutch position results indirectly from the declutch time set in C05333/1. • When the declutch process is terminated, the *bOpenDone* output is set to TRUE.
		Clutch-in mode "Declutching until standstill": • The *dnPosOut_p* output position disengages from the *dnPosIn_p* master position within the declutch time set in C05333/1 and decelerated to the *dnOpenPos_p* standstill position. • When the *dnPosOut_p* output position has reached the standstill position, the *bInTarget* output is set to TRUE.
		Clutch-in mode "Declutching to basic speed": • The *dnPosOut_p* output position disengages from the *dnPosIn_p* master position within the declutch time set in C05333/1 and decelerated to the basic speed defined in the *dnBaseSpeed_s* input.
	FALSE	"Close clutch" request (clutch-in process). • The clutch-in process can only be started after the declutch process has been totally completed. An early abort of the declutch process is not possible. • The *dnPosOut_p* output position is synchronised from the standstill position (or basic speed) to *dnPosIn_p* master position again within the clutch-in time set in C05333/1 and follows it. • If the input and output position are synchronous again, the output *bClosed* is set to TRUE.
bOpenInstant BOOL	Positive opening operation of the clutch (for emergency situations) • This input has a higher priority than the input *bOpen*.	
	FALSE↗TRUE	The *dnPosOut_p* output position is separated immediately from the *dnPosIn_p* 3 master position and brought to standstill within the deceleration time set in C05333/3. • It is not possible to abort the positive opening operation early. • If the *dnPosOut_p* output position is at standstill, the output *bOpenDone* is set to TRUE. • Afterwards a clutch-in process by a FALSE-TRUE edge at the input *bOpen* can only be effected again after the input *bOpenInstant* has been reset to FALSE.
bForcedClose BOOL	Reset declutch process. • This function allows for a "hard clutch-in" and can be used for certain interruption mechanisms or for starting up the machine. • The reset of the declutch process should be only effected during controller inhibit or a decoupled setpoint path. • This input has the highest priority.	
	TRUE	The *dnPosOut_p* output position is set immediately to the *dnPosIn_p* master position. • The clutch is now clutched-in. • The output *bClosed* is set to TRUE.

Identifier/data type	Information/possible settings	
bLoadPosOut BOOL	Reset clutch-in process. • The function enables a "hard declutching". • This input has the second highest priority after *bForcedClose*.	
	TRUE	The *dnPosOut_p* output position is immediately set to the position defined via the *dnLoadPosOut_p* input. • The clutch is now declutched. • The *bOpenDone* output is set to TRUE.
dnLoadPosOut_p DINT	Position in [increments] to which the *dnPosOut_p* output position is set by setting *bLoadPosOut* to TRUE. • C05335/3 indicates the position in the real unit of the machine.	
AxisData	Machine parameters • If the FB is to relate to the measuring system of the drive axis, connect this input to the output *DI_AxisData* of the **LS_DriveInterface** SB for accepting the machine parameters of the drive/motor. • If the FB within a cam application is to relate to the measuring system of the master selection (master), connect this input to the output *XAxisData* of the L_CamProfiler FB instead. (📖 162) • If this input is left unconnected, the position values are processed with the following resolution: 1 encoder revolution ≡ 2^{16} increments	

Outputs

Identifier/data type	Value/meaning	
dnPosOut_p DINT	Output position in [increments] • C05337 indicates the output position in the real unit of the machine.	
dnSpeedOut_s DINT	Output speed given as speed in [rpm] • C05338 indicates the output speed in the real unit of the machine.	
bClosed BOOL	Status signal to the clutch status	
	FALSE	The clutch is open, or an active clutch-in/declutch process is active. • The output position is at standstill or within an acceleration/deceleration phase.
	TRUE	The clutch is closed. • Output position *dnPosOut_p* and input position *dnPosIn_p* are synchronous.
bOpenDone BOOL	Status signal "Declutch process completed"	
	TRUE	The declutch process activated via *bOpen* or *bOpenInstant* has been totally completed; a clutch-in process can be effected again.
bInTarget BOOL	Status signal "Standstill position reached"	
	TRUE	The standstill position *dnOpenPos_p* after a declutch process activated via *bOpen* has been reached.
bError BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameter C05339.

Parameter

Parameter	Possible settings			Information
C05330/1	String of digits			Position unit • Read only
C05330/2	String of digits			Speed unit • Read only
C05331	0.0000	Unit/t	214748.3647	Reference speed • Read only

Parameter	Possible settings			Information
C05332				Clutch mode
	0	Declutching until standstill (*dnOpenPos_p*).		
	1	Declutching to basic speed (*dnBaseSpeed_s*).		Lenze setting
C05333/1	0.001	s	10.000	Relative declutch time • With regard to the *dnOpenPos_p* standstill position or the *dnBaseSpeed_s* basic speed. • Initialisation: 1.000 s
C05333/2	0.001	s	10.000	Relative clutch-in time • Relating to the position from which the output position *dnPosOut_p* and input position *dnPosIn_p* are synchronous again. • Initialisation: 1.000 s
C05333/3	0.001	s	10.000	Deceleration time for positive opening operation • Relating to the reference speed. • Initialisation: 1.000 s
C05335/1	-214748.3647	Unit	214748.3647	Input position • Display of the *dnPosIn_p* input signal in the real unit of the machine.
C05335/2	-214748.3647	Unit	214748.3647	Standstill position • Position after declutch process • Display of the *dnOpenPos_p* input signal in the real unit of the machine.
C05335/3	-214748.3647	Unit	214748.3647	Position for "Hard declutching" • Position to which the *dnPosOut_p* output position is set immediately by setting *bLoadPosOut* to TRUE. • Display of the *dnLoadPosOut_p* input signal in the real unit of the machine.
C05336	-214748.3647	Unit/t	214748.3647	Basic speed • Display of the *dnBaseSpeed_s* input signal in the real unit of the machine.
C05337	-214748.3647	Unit	214748.3647	Output position • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C05338	-214748.3647	Unit/t	214748.3647	Output speed • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.

Parameter	Possible settings		Information
C05339	Status messages:		Status (bit coded) • Bits that are not itemised are reserved for future extensions. • The *bError* error output will only be set to TRUE if an error message (bit 16 ... 31) is issued. • Bit 31 is the default setting if one or more other error bits (bit 16 ... 30) are set.
	bits8	Clutch closed	
	bits9	Declutch position reached	
	bits14	Internal position overflow during declutch process	
	Bit15	Internal position overflow during clutch-in process	
	Error messages:		
	bit16	Position is beyond the cycle	
	bits23	Invalid axis data structure	
	bits31	General error	

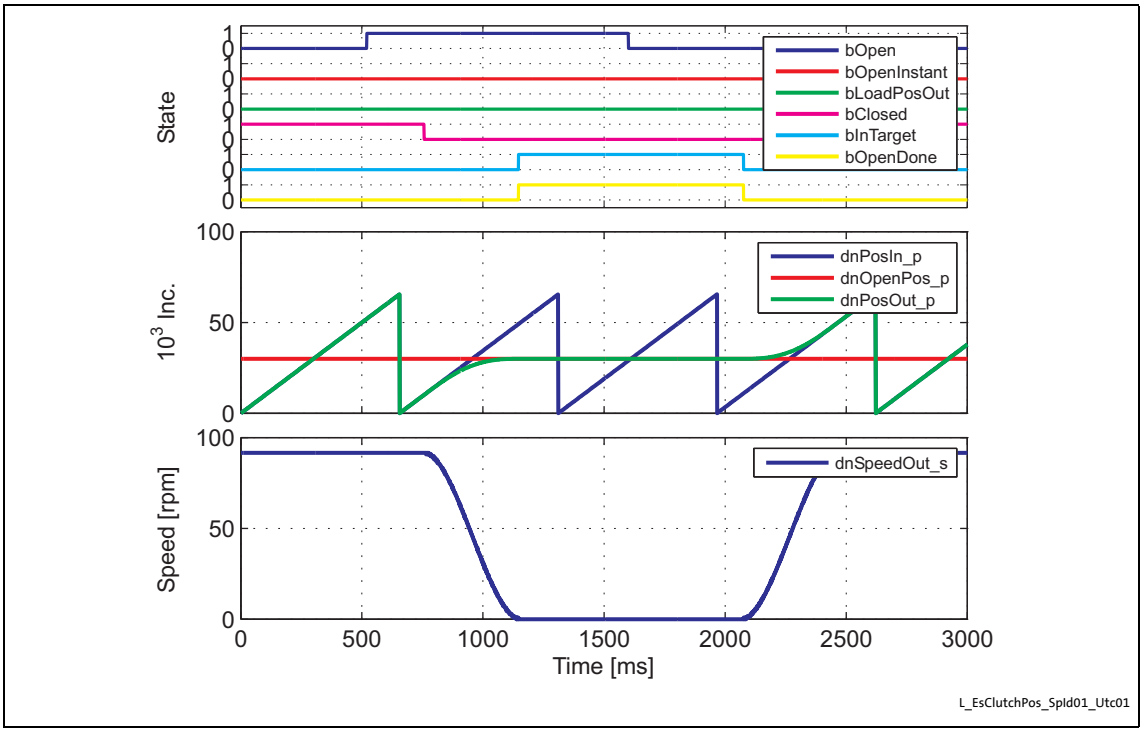
Note!

A clutch-in or declutch process may last up to 10 seconds, depending on the parameterised resolution of an encoder revolution in the machine parameters copied to the FB (*AxisData*).

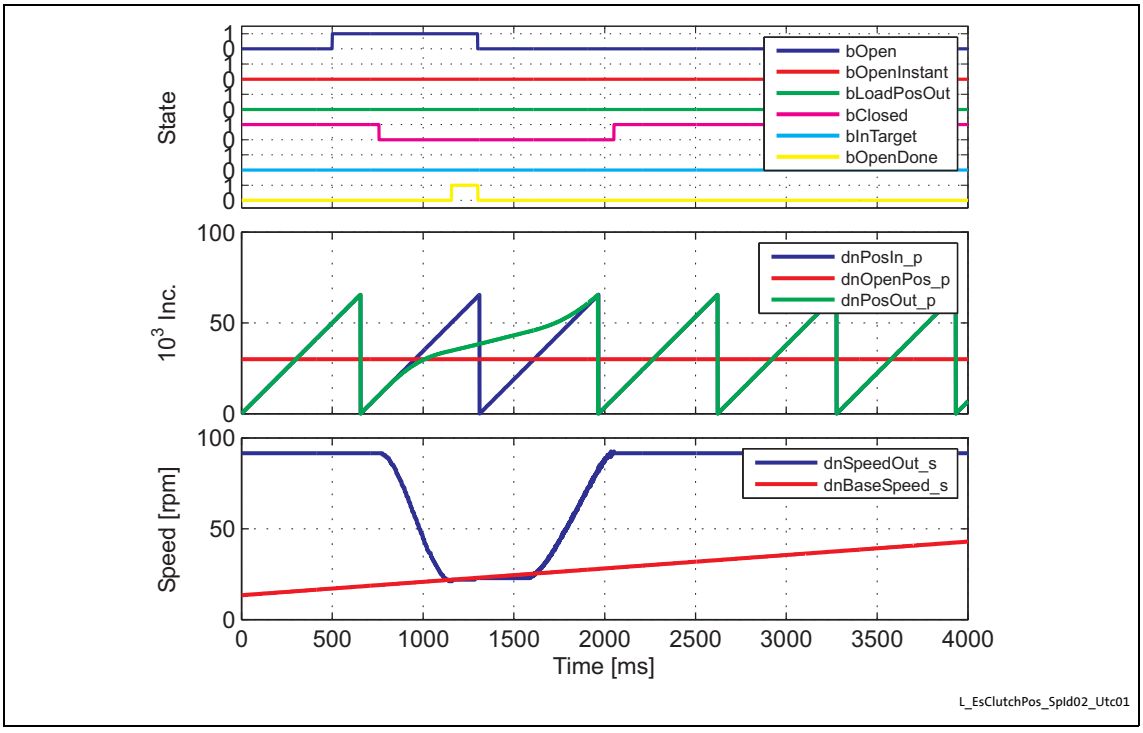
Example: If the maximum resolution of 24 bit/encoder revolution is set, the clutch-in path may be too long after only 2 s at maximum speed (internal position gets lost).

- In this case, clutching-in is executed via an emergency time ramp. This changes the course of speed and position during the clutch-in and declutch process.
- During the declutch process, you cannot stop at the standstill position *dnOpenPos_p* anymore! The time ramp, however, is maintained and the standstill position is within the cycle.
- An internal position overflow during the clutch-in/declutch process is displayed in C05339 via bit 14 and bit 15.

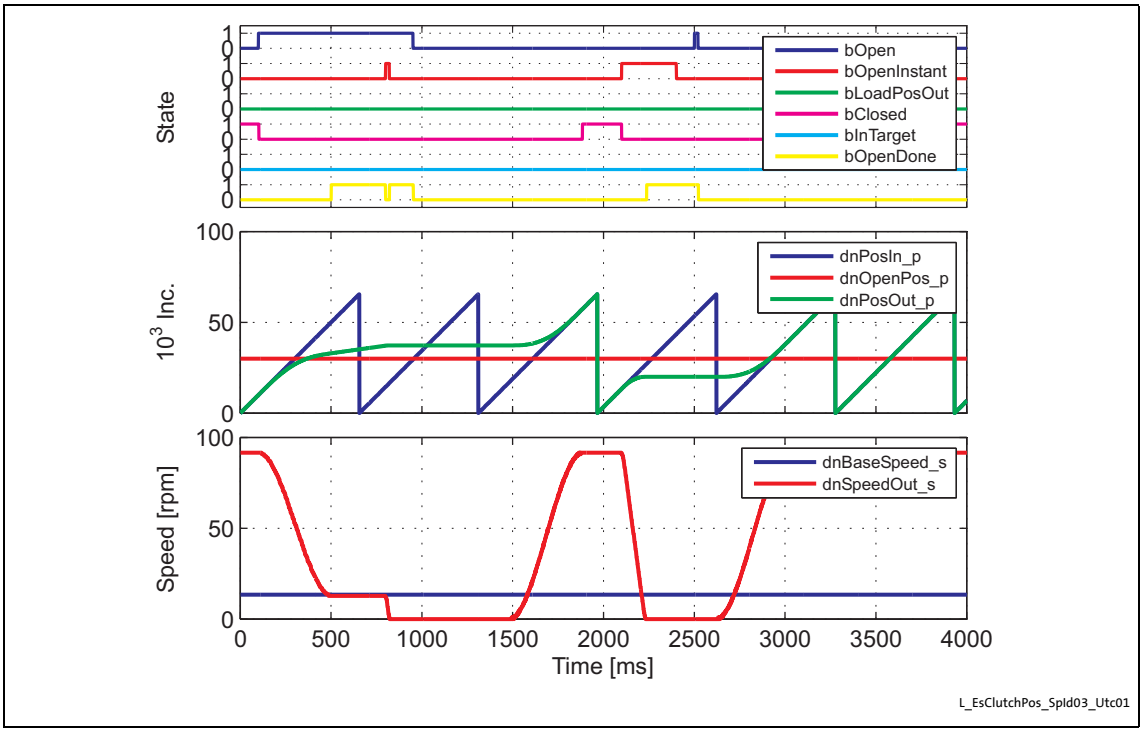
5.80.1 Signal characteristics



[5-37] Clutch-in mode "0: Declutching until standstill"

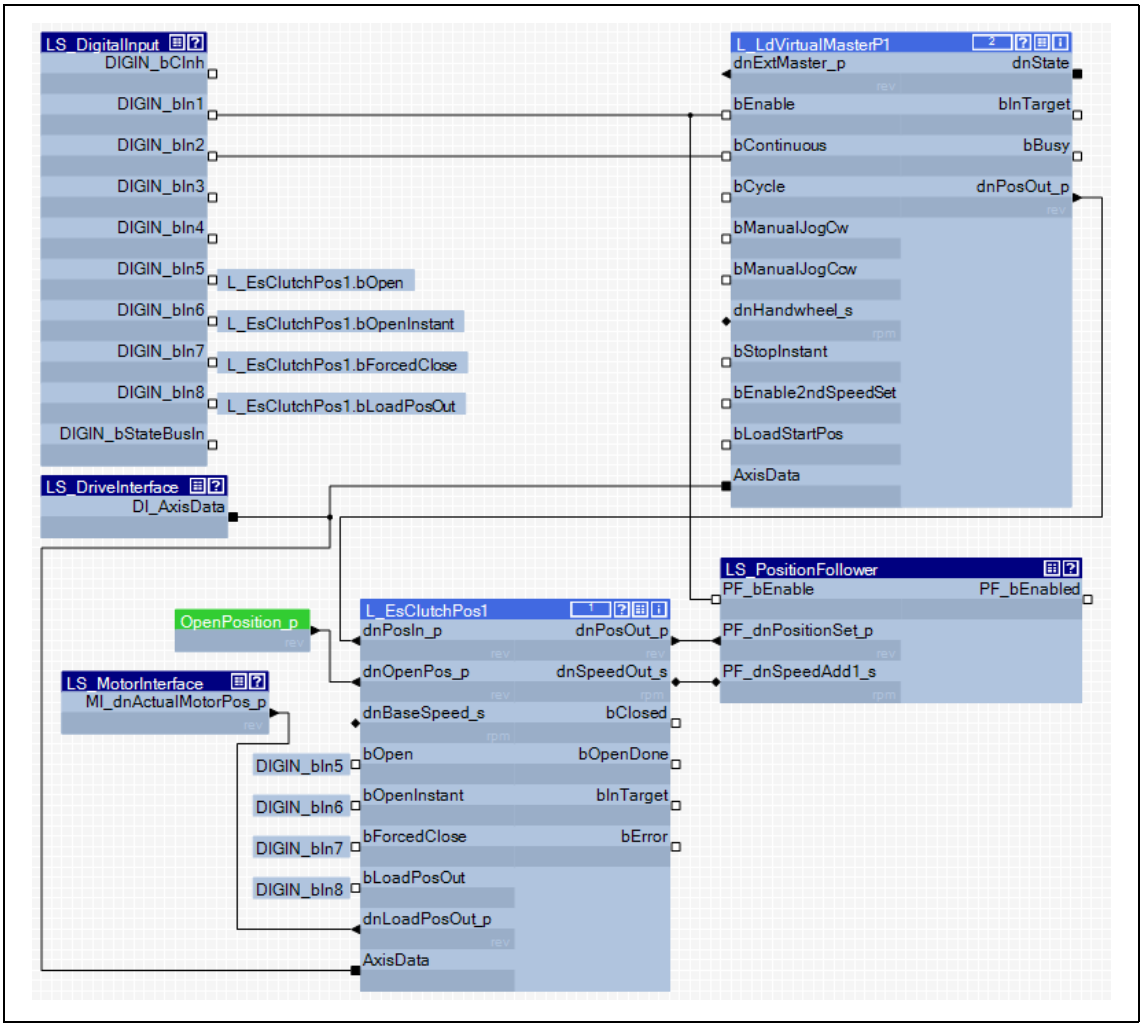


[5-38] Clutch-in mode "1: Declutching to basic speed"



[5-39] Positive opening operation of the clutch (*bOpenInstant* = TRUE)

5.80.2 Interconnection example



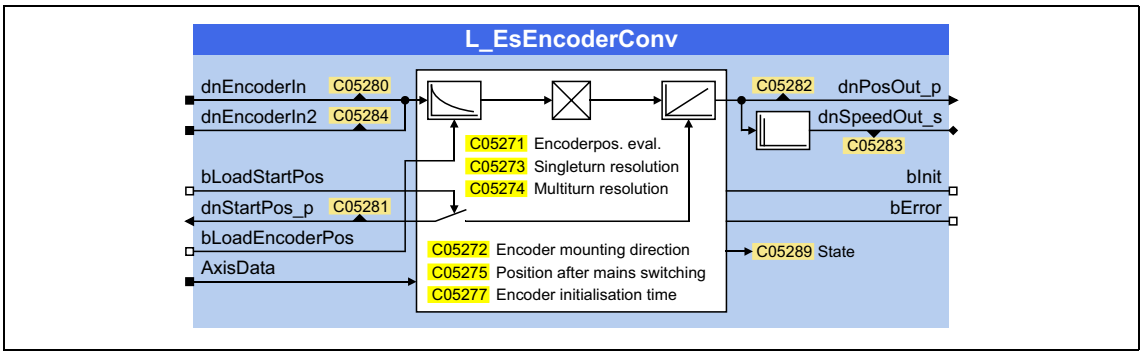
[5-40] Interconnection example

5.81 L_EsEncoderConv - encoder signal conditioning

Function library:	LenzeElectricalShaft
Runtime software licence:	☐ Motion Control HighLevel ☑ Motion Control TopLevel

This FB converts an actual position that is imported via a bus system, like for example system bus (CAN), Profibus, or ETHERNET Powerlink, by an external encoder, by means of the machine parameters transferred and the encoder resolution parameterised into a position in [increments] and provides it for further processing within the function block interconnection at the output *dnPosOut_p* .

- Via the input *bLoadStartPos* the position output can be set to the starting position pending at the input *dnStartPos_p*.
- The position that is output can be initialised/adjusted to the input value of an absolute value encoder via the *bLoadEncoderPos* input.
- In addition to the position, the resulting speed is shown at the output *dnSpeedOut_s*.
- The FB can process absolute value encoders and incremental encoders (single and multiturn encoders).



Inputs

Identifier/data type	Information/possible settings
dnEncoderIn DINT	External encoder signal (encoder counter) • As soon as the input value at the bus is changed, a valid encoder signal is recognised. • Display parameter: C05280
dnEncoderIn2 DINT From library V01.03.xx.xx	2nd external encoder signal • This input is only evaluated if "1: 64 bit encoder signal" has been selected in C05271. • Display parameter: C05284 ▸ 64 bit encoder signal evaluation (📖 302)
bLoadStartPos BOOL	Load integrator with starting position and reset error output. TRUE: Load the integrator with the value applied to input *dnStartPos_p* and reset *bError* to FALSE.
dnStartPos_p DINT	Starting position in [inc] with which the integrator is loaded by setting *bLoadStartPos* to TRUE. • C05281 indicates the starting position in the real unit of the machine.

Identifier/data type	Information/possible settings	
bLoadEncoderPos BOOL From library V01.03.xx.xx	Load integrator with absolute encoder position • Application: Initialisation of output positions for absolute value encoders	
	TRUE	Load integrator with encoder positions at the *dnEncoderIn* / *dnEncoderIn2* inputs • The resolution of an encoder revolution set in the machine parameters (AxisData) will be considered here. • Input signal changes still result in the generation of a speed value at the output.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters contained within the cam data can be read out via L_CamGetAxisData FB. In this case the FB output *AxisData* must be connected to this input. (📖 155) • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510) • If this input is released, the following scaling applies: 1 motor revolution ≡ 2^{16}	

Outputs

Identifier/data type	Value/meaning	
dnPosOut_p DINT	Master position output in [inc] • Overflow is possible (display via *bError*) • C05282 indicates the master position in the real unit of the machine.	
dnSpeedOut_s DINT	Master speed output in [rpm] • C05283 indicates the master speed in the real unit of the machine.	
bInit BOOL	Status signal "Encoder initialisation phase active"	
	TRUE	The initialisation time for the encoder set in C05277 has not elapsed yet, the FB does not yet carry out internal calculations.
bError BOOL	"Error" status • Status signal can be reset via *bLoadStartPos*.	
	TRUE	An error has occurred (group signal). • For details see display parameter C05289.

Parameter

Parameter	Possible settings		Information
C05270/1	String of digits		Position unit • Read only
C05270/2	String of digits		Speed unit • Read only
C05271 From library V01.03.xx.xx			Encoder position evaluation
	0	32 bit encoder signal • The 32 bit input position is transmitted via the *dnEncoderIn* input. • The resolution of the signal results from C05273 and C05274.	Lenze setting
	1	64 bit encoder signal • The 64 bit input position is transmitted via the *dnEnoderIn* and *dnEncoderIn2* input. • The resolution of the signal results from C05274.	▸ 64 bit encoder signal evaluation (📖 302)

Parameter	Possible settings			Information
C05272				Encoder mounting position
	0	On the right		Lenze setting
	1	On the left		
C05273	1		2147483647	Singleturn resolution • Setting of the number of increments. • Initialisation: 2048
C05274	1		2147483647	Multiturn resolution • Number of revolutions. • Initialisation: 1 (= singleturn)
C05275				Position after main switching
	0	Delete		Lenze setting
	1	Reconstruct		
C05277	0.010	s	60.000	Encoder initialisation time • Time after which the encoder supplies signals after switch-on. • Initialisation: 1.000 s
C05280	-2147483647	Steps	2147483647	Encoder input • Display of the *bEncoderI_n* input signal.
C05281	-214748.3647	Unit	214748.3647	Starting position • Display of the *dnStartPos_p* input signal in the real unit of the machine.
C05282	-214748.3647	Unit	214748.3647	Master position • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C05283	-214748.3647	Unit/t	214748.3647	Master speed • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.
C05284 From library V01.03.xx.xx	-2147483647	Rev	2147483647	2nd encoder input • Display of the *dnEncoderIn2* input signal.
C05289	Status messages:			Status (bit coded) • Bits that are not itemised are reserved for future extensions. • The *bError* error output will only be set to TRUE if an error message (bit 16 ... 31) is issued. • Bit 31 is the default setting if one or more other error bits (bit 16 ... 30) are set.
	bit 10	Zero crossing active		
	Error messages:			
	bit 16	Position is beyond the cycle		
	bit 17	Initial speed too high		
	bit 19	Encoder resolution too high		
	bit 21	Position skip ≥ half a cycle		
	bit 23	Invalid axis data structure		
	bit 25	Encoder input signal invalid		
	bit 31	General error		

5.81.1 Typical application

Via the FB a singleturn or multiturn absolute value bus encoder (CAN, Profibus) can be adapted to the application.

- Example: Master value encoder via system bus (CAN) for detecting the speed of the material path.
- The encoder information conditioned can be optionally used as absolute position or as master speed.
- Via the inputs and parameters the position output can be set to a position defined (set reference) which is reconstructed after mains switching.

5.81.2 Encoder settings

Encoder resolution

By means of the settings in C05273 (singleturn resolution) and C05274 (multiturn resolution) the FB recognises whether a singleturn or multiturn encoder is used.

- For an encoder of the "singleturn" type, for instance only the number of increments of the encoder has to be set in C05273, and C05274 remains on the Lenze setting "1".
- If a "multiturn" encoder is used, in addition to the number of increments the number of revolutions has to be set in C05274.

Note!

The product of a singleturn and multiturn resolution set must not exceed the numerical value of 2^{31}-1!

Please also see: ▸ 64 bit encoder signal evaluation (📖 302)

Encoder initialisation time

The FB can consider an encoder initialisation time after which the encoder for the first time supplies signals via the bus after switch-on.

- This initialisation time is to be set in C05277.
- Until this initialisation time has elapsed, the output *bInit* is set to TRUE, and the FB does not carry out internal calculations yet.

Behaviour after mains switching

In C05275 it can be selected whether the position is to be deleted after mains switching (initialised with "0"), or if the last value is to be reconstructed, which is only reasonable if an absolute value encoder is used.

5.81.3 64 bit encoder signal evaluation

This function extension will be available from library V01.03.xx.xx!

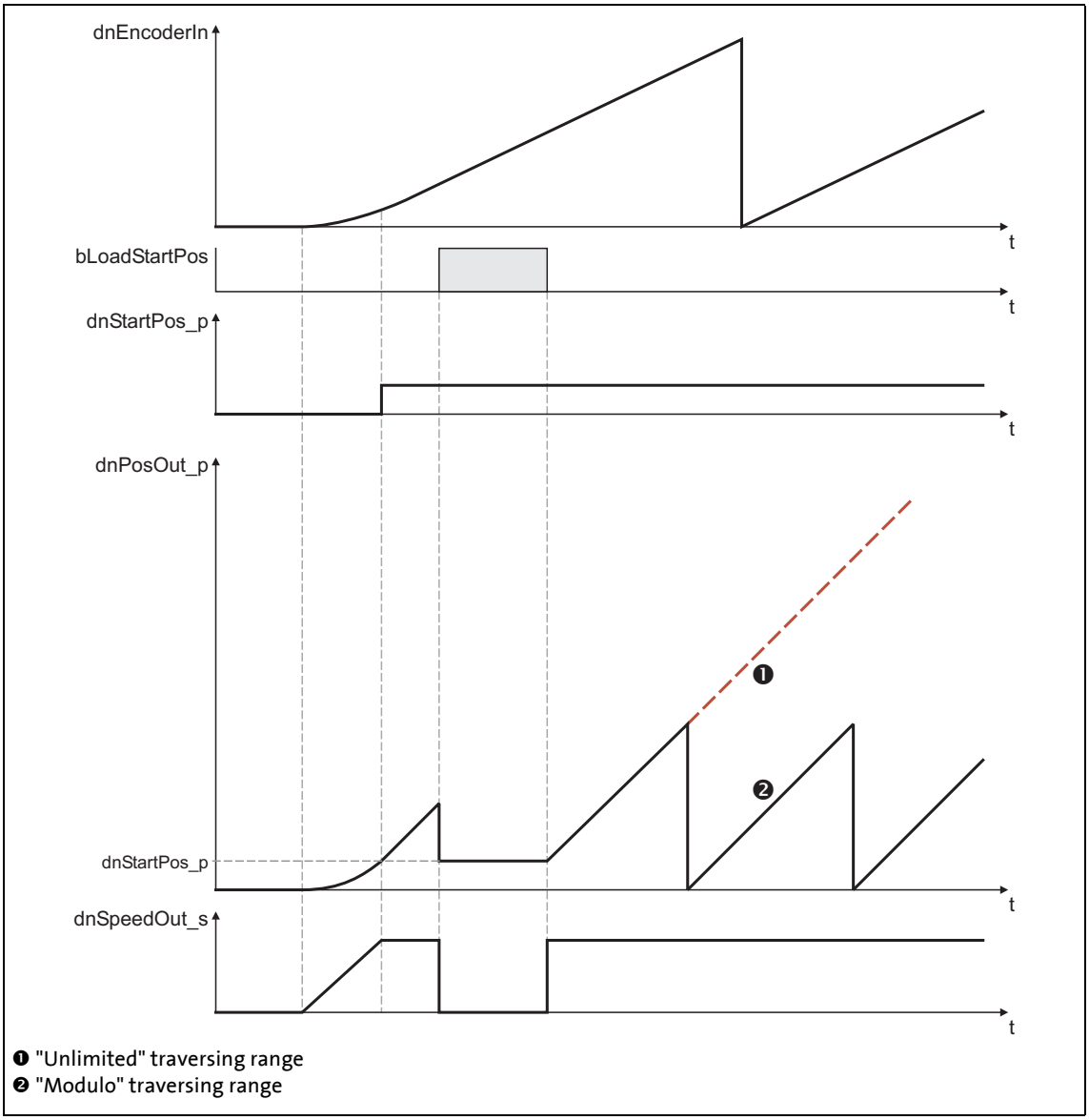
If "1: 64 bit encoder signal" has been selected in C05271, the *dnEncoderIn*2 input is evaluated in addition to the *dnEncoderIn* input for the second encoder signal:

- The position (number of increments) is transmitted within a revolution via the *dnEncoderIn* input. The position has a fixed resolution of 32 bits.
- The current number of encoder revolutions is transmitted via the *dnEncoderIn2* input.
- The (multiturn) resolution for the *dnEncoderIn2* input is set in C05274.
- C05273 (singleturn resolution) is inactive in this mode (is not evaluated).

How to convert encoder information / an encoder position from encoder input X8 into a _p position of the internal measuring system:

1. Connect the *ENC_dnEncoderOut1* output of the **LS_EncoderX8** system block to the *dnEncoderIn* input.
2. Connect the *ENC_dnEncoderOut2* output of the **LS_EncoderX8** system block to the *dnEncoderIn2* input.
3. Select "1: 64 bit encoder signal" in C05271 to ensure that the second input, *dnEncoderIn2*, is also evaluated.
4. Set the resolution of the multiturn encoder in C05274 (maximum number of revolutions).

5.81.4 Signal characteristics



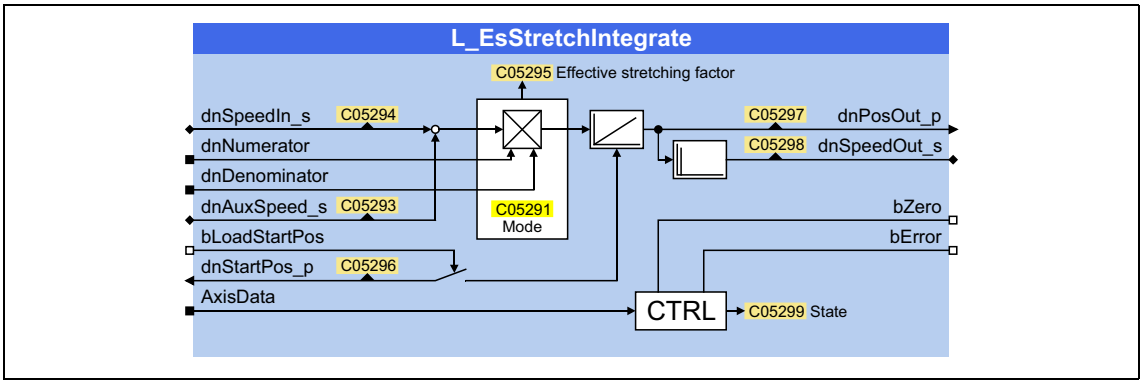
[5-41] Signal characteristics

5.82 L_EsStretchIntegrate - synchronously stretching/compressing the master value

Function library:	LenzeElectricalShaft
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to stretch and compress a master value in the "Modulo" or "Unlimited" traversing range. For this purpose, the FB integrates the input speed to a position and stretches or compresses it.

- Via the inputs *dnNumerator* and *dnDenominator* a free stretch factor can be defined for the speed ratio.
- In C05291 only the synchronous stretching/compression is preset as a mode, i. e. a new stretch factor is only accepted in the cycle transition of the position (modulo systems only).
- Via the input *bLoadStartPos* the position output can be set to the starting position pending at the input *dnStartPos_p*.
- In addition to the position, the resulting speed is shown at the output *dnSpeedOut_s*.



Inputs

Identifier/data type	Information/possible settings
dnSpeedIn_s DINT	Speed in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864 • Display parameter: C05294
dnNumerator DINT	Stretch factor (numerator) • $\pm 2^{31}$-1 • If an invalid value is selected, the ratio will be set to "0".
dnDenominator DINT	Stretch factor (denominator) • 0 ... 2^{31}-1 • A negative stretch factor can only be defined via *dnNumerator*. • In the case of an input value < 0, processing is continued internally with 1. • If an invalid value is selected, the ratio will be set to "0".
dnAuxSpeed_s DINT From library V01.01.xx.xx	Additional (compensation) speed in [rpm] • For example from mark correction. • 15000 rpm ≡ 2^{26} ≡ 67108864 • Display parameter: C05293
bLoadStartPos BOOL	Load the integrator with starting position TRUE: Load the integrator with the value at the *dnStartPos_p* input.

Identifier/data type	Information/possible settings
dnStartPos_p DINT	Starting position in [inc] with which the integrator is loaded by setting *bLoadStartPos* to TRUE. • C05295 indicates the starting position in the real unit of the machine.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters contained within the cam data can be read out via L_CamGetAxisData FB. In this case the FB output *AxisData* must be connected to this input. (📖 155) • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

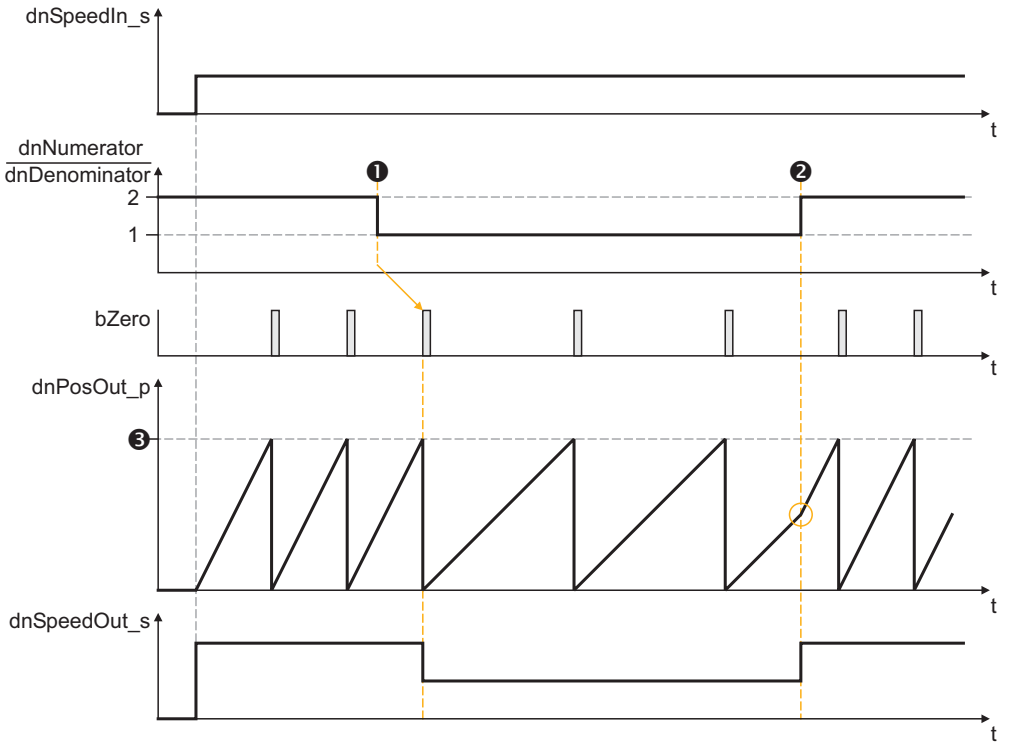
Identifier/data type	Value/meaning	
dnPosOut_p DINT	Master position output in [inc] • Overflow is possible (display via *bLimit*) • C05297 indicates the master position in the real unit of the machine.	
dnSpeedOut_s DINT	Master speed output in [rpm] • C05298 indicates the master speed in the real unit of the machine.	
bZero BOOL	Status signal "Zero position/zero crossing"	
	FALSE	Master value is at any position in the master value cycle (register)
	TRUE	**Permanently:** Master value is at zero position. **For one cycle:** Zero crossing of the master value.
		At the same time the status bit 10 ("Zero crossing active") is set in C05299.
bError BOOL	"Error" status • The status signal can be reset to FALSE via *bLoadStartPos* if the error was caused by a too great starting position at *dnStartPos_p* (see bit 16 in C05299).	
	TRUE	An error has occurred (group signal). • For details see display parameter C05299. • *bError* can be reset to FALSE via *bLoadStartPos* if the error was caused by a too great starting position at *dnStartPos_p* (see bit 16 in C05299).

Parameter

Parameter	Possible settings			Information
C05290/1	String of digits			Position unit • Read only
C05290/2	String of digits			Speed unit • Read only
C05291				Stretching/compression mode
	0	Synchronised • Acceptance of a new stretch factor only in the zero crossing.		Lenze setting
	1	Not synchronised • Immediate acceptance of a new stretch factor.		
C05293 From library V01.01.xx.xx	-2147483647	Unit/t	2147483647	Additional compensating speed • Display of the *dnAuxSpeed_s* input signal.

Parameter	Possible settings			Information
C05294	-2147483647	Unit/t	2147483647	Input speed • Display of the *dnSpeedIn_s* input signal.
C05295	-214748.3647		214748.3647	Effective stretch factor
C05296	-214748.3647	Unit	214748.3647	Starting position • Display of the *dnStartPos_p* input signal in the real unit of the machine.
C05297	-214748.3647	Unit	214748.3647	Master position • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C05298	-214748.3647	Unit/t	214748.3647	Master speed • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.
C05299	Status messages:			Status (bit coded) • Bits that are not itemised are reserved for future extensions. • The *bError* error output will only be set to TRUE if an error message (bit 16 ... 31) is issued. • Bit 31 is the default setting if one or more other error bits (bit 16 ... 30) are set.
	bits10	Zero crossing active		
	Error messages:			
	bit16	Position is beyond the cycle		
	bits17	Initial speed too high		
	bits21	Position jump is too big		
	bits23	Invalid axis data structure		
	bits31	General error		

5.82.1 Signal characteristics



❶ Synchronised stretching/compression (C05291 = "1: synchronised")
❷ Immediate acceptance of a new stretch factor (C05291 = "0: not synchronised")
❸Cycle

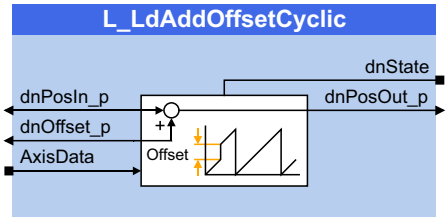
[5-42] Signal characteristics

5.83 L_LdAddOffsetCyclic - offset addition

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to add any offset position to a position signal with Modulo cycle.

- The clock pulse is created without remainder processing.



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Clocked position in [inc] • Apply the position signal with Modulo cycle to this input. • The position must match the measuring system and must not exceed twice the cycle length of the modulo system so that the output is still limited to permissible values (= cycle length of the modulo measuring system). Higher values cause a standstill at the FB output.
dnOffset_p DINT	Offset in [inc] • The offset may be a position of any size. This also applies to the "Modulo" traversing range.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

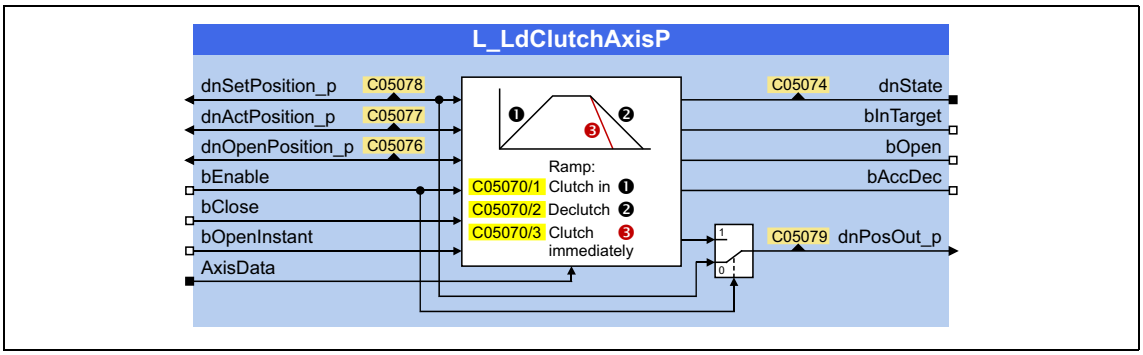
Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits10	Zero crossing active
	Bit15	Error
	bits18	No cycle available
	bits21	Input error (e.g. impermissible position jump or inverse direction of motion)
	bits23	No valid axis data structure
dnPosOut_p DINT	Position output in [inc]	

5.84 L_LdClutchAxisP - virtual clutch for synchronism

Function library:	LenzeLineDrive	**Do not use FB for new developments!** Replacement by L_EsClutchPos.
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB provides the "Time-controlled virtual clutch" function with position reference for the "Synchronism" application.



Inputs

Identifier/data type	Information/possible settings	
dnSetPosition_p DINT	Selection of an external master position in [inc] • E.g. by an absolute value encoder or a virtual master. • The external master position must be within the cycle. • C05078 indicates the position of the master shaft in the real unit of the machine.	
dnActPosition_p DINT	Internal master position of the clutch function in [inc] • The internal master position must be within the cycle. • C05077 indicates the master position in the real unit of the machine. • For a master value clutch, connect this input with the output *dnPosOut_p* of this FB.	
dnOpenPosition_p DINT	Selection of the stop position after declutch or selection of the starting position when engaging the master shaft in [inc]. • The stop position/starting position must be within the cycle. • C05076 indicates the stop position in the real unit of the machine.	
bEnable BOOL	Activate the clutch function • This input has the highest priority.	
	FALSE	Clutch function is deactivated. • The input *dnSetPosition_p* connected 1:1 to the output *dnPosOut_p*. No ramps are active.
	TRUE	Clutch function is activated.
bClose BOOL	Control of clutch function	
	FALSE	Open the clutch. • The master position *dnPosOut_p* is positioned via the set ramp to the stop position *dnOpenPosition_p*.
	TRUE	Close the clutch. • The master position *dnPosOut_p* is synchronised via the set ramp and follows the master shaft *dnSetPosition_p* after it has been clutched in.

Identifier/data type	Information/possible settings	
bOpenInstant BOOL	Positive opening operation of the clutch. • This input has the second highest priority (after *bEnable*).	
	FALSE↗TRUE	The master position *dnPosOut_p* is separated from the master shaft and stopped via the set ramp.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	Bit 1	State active
	Bit 2	Procedure completed
	Bit 3	Acceleration/deceleration phase
	Bit 8	Brake/clutch closed
	Bit 9	Waiting for clutch process
	Bit 15	Error
	Bit 16	Starting position is beyond the cycle
	Bit 18	No cycle available
	Bit 19	Maximum speed = "0"
	Bit 21	Input error (e.g. impermissible position jump or inverse direction of motion)
	Bit 23	No valid axis data structure
bInTarget BOOL	Status signal "Position reached"	
	TRUE	Position reached (clutch or declutch process or positive opening operation has been completed).
bOpen BOOL	Status signal "Clutch/declutch process is active"	
	FALSE	Clutch is closed, master position follows the master shaft.
	TRUE	Master position is synchronised to the master shaft after the clutch is closed or the master position is shut down after the clutch is opened. • *dnActPosition_p* and *dnSetPosition_p* are not synchronous!
bAccDec BOOL	Status signal "Ramp function of the clutch is active"	
	FALSE	Ramp function of the clutch is not active. • The clutch is closed or the master position is shut down.
	TRUE	Ramp function of the clutch is active. • The master position is synchronised or shut down. • The TRUE signal is pending immediately if the clutch or declutch process is started via the input *bClose*.
dnPosOut_p DINT	Master position output (current position of the master shaft in the cycle) in [inc] • C05079 indicates the current master position in the real unit of the machine.	

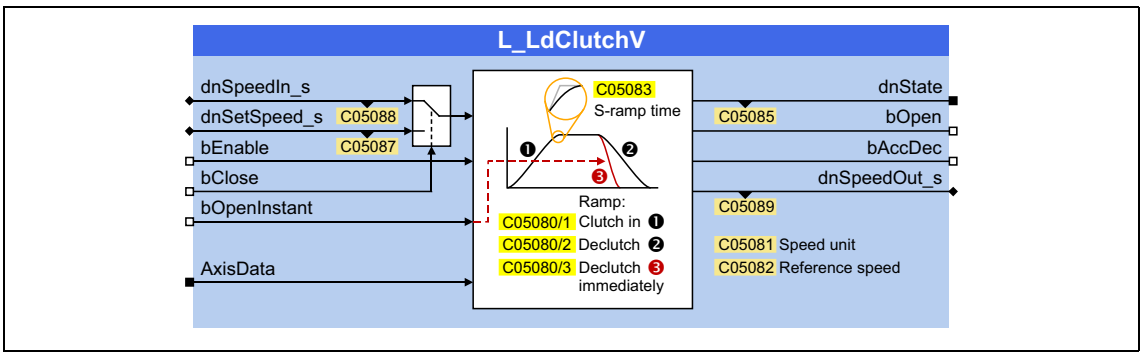
Parameters

Parameter	Possible settings			Info
C05070/1	0.010	s	130.000	Ramp: Clutch in • Acceleration ramp for synchronising to the master shaft. • This specification and the reference speed that exists in the axis data serve to calculate an internal acceleration. This acceleration serves to calculate the clutch-in path. • When a higher clutch-in time is set, the clutch-in path can also be executed via several cycles. • Initialisation: 1.000 s
C05070/2	0.010	s	130.000	Ramp: Declutch • Deceleration ramp for shutdown of the master position. • This specification and the reference speed that exists in the axis data serve to calculate an internal acceleration. This acceleration serves to calculate the declutch path. • When a higher declutch time is set, the declutch path can also be executed via several cycles. • Initialisation: 1.000 s
C05070/3	0.010	s	130.000	Ramp: Positive opening operation • Deceleration ramp for the positive opening operation. • Initialisation: 1.000 s
C05071/1 From library V02.02.xx.xx onwards	Character string			Position unit • Read only
C05071/2 From library V02.02.xx.xx onwards	Character string			Speed unit • Read only
C05072 From library V02.02.xx.xx onwards	-214748.3647	Unit/t	214748.3647	Reference speed • Read only • For input orientation for speed entries.
C05074	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.
C05076	-214748.3647	Unit	214748.3647	Set position after declutching • Display of the *dnOpenPosition_p* input signal in the real unit of the machine.
C05077	-214748.3647	Unit	214748.3647	Actual position of the master angle • Display of the *dnActPostion_p* input signal in the real unit of the machine.
C05078	-214748.3647	Unit	214748.3647	Master position at the input • Display of the *dnSetPostion_p* input signal in the real unit of the machine.
C05079	-214748.3647	Unit	214748.3647	Master position at the output • Display of the *dnPosOut_p* output signal in the real unit of the machine.

5.85 L_LdClutchV - virtual clutch for electronic gearboxes

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB provides the "Virtual clutch" function with speed reference for the "Electronic gearbox" application.



Inputs

Identifier/data type	Information/possible settings	
dnSpeedIn_s DINT	Setpoint speed • At this input, the FB expects a speed in [rpm]. The machine parameters at the input *AxisData* are used as speed reference. • C05088 indicates the setpoint speed in the real unit of the machine.	
dnSetSpeed_s DINT	Basic speed • At this input, the FB expects a speed in [rpm]. The machine parameters at the input *AxisData* are used as speed reference. • The selected speed is output at *dnSpeedOut_s* when the clutch is open. • C05087 indicates the basic speed in the real unit of the machine.	
bEnable BOOL	Activate the clutch function • This input has the highest priority.	
	FALSE	Clutch function is deactivated. • The input *dnSpeedIn_s* is connected through 1:1 to output *dnSpeedOut_s*. No ramps are active.
	TRUE	Clutch function is activated.
bClose BOOL	Control of clutch function	
	FALSE	Open the clutch. • The master speed *dnSpeedOut_s* is led to the basic speed *dnSetSpeed_s* via the set ramp.
	TRUE	Close the clutch. • The master speed *dnSpeedOut_s* is synchronised to the setpoint speed *dnSpeedIn_s* via the set ramp.
bOpenInstant BOOL	Positive opening operation of the clutch. • This input has the second highest priority (after *bEnable*).	
	FALSE↗TRUE	The master speed *dnSpeedOut_s* is separated from the master shaft and stopped via the set ramp.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits1	State active
	bits3	Acceleration/deceleration phase
	bits8	Brake/clutch closed
	Bit15	Error
	bits23	No valid axis data structure
bOpen BOOL	Status signal "Clutch/declutch process is active"	
	FALSE	Clutch is closed, *dnSpeedOut_s* follows *dnSpeedIn_s*.
	TRUE	*dnSpeedOut_s* is asynchronous to *dnSpeedIn_s* and is led via the internal profile generator.
bAccDec BOOL	Status signal "Ramp function of the clutch is active"	
	FALSE	Ramp function of the clutch is not active. • The clutch is closed or the master speed is shut down.
	TRUE	Ramp function of the clutch is active. • The master speed is accelerated or decelerated.
dnSpeedOut_s DINT	Speed output in [rpm] • C05089 indicates the speed in the real unit of the machine.	

Parameter

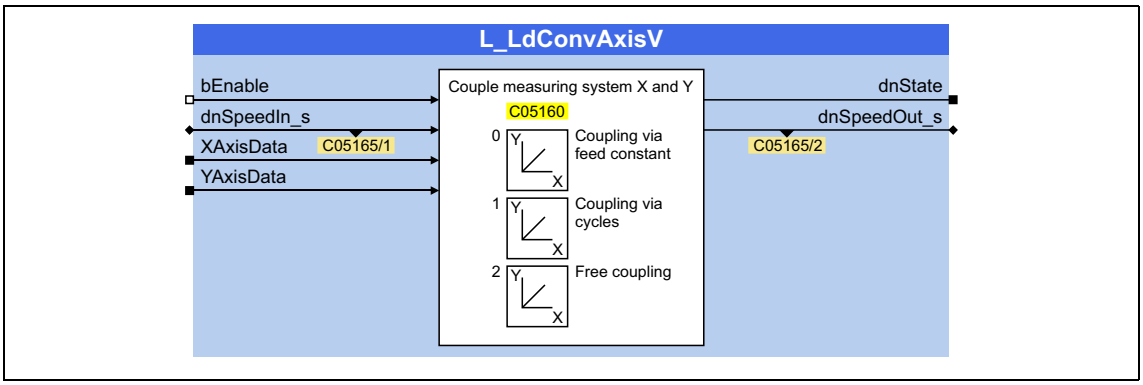
Parameter	Possible settings			Information
C05080/1	0.010	s	130.000	Ramp: Clutch in • Acceleration ramp for synchronising to *dnSpeedIn_s*. • Initialisation: 1.000 s
C05080/2	0.010	s	130.000	Ramp: Declutch • Deceleration ramp for synchronising to *dnSetSpeed_s*. • Initialisation: 1.000 s
C05080/3	0.010	s	130.000	Ramp: Positive opening operation • Deceleration ramp for the positive opening operation. • Initialisation: 1.000 s
C05081 As of library V02.02.xx.xx	String of digits			Speed unit • Read only
C05082 As of library V02.02.xx.xx	-214748.3647	Unit/t	214748.3647	Reference speed • Read only • For input orientation for speed entries.
C05083	0.010	s	130.000	S-ramp time • For jerk limitation. • Initialisation: 1.000 s
C05085	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.
C05087	-214748.3647	Unit	214748.3647	Setpoint speed after declutching • Read only • Calculated from the speed signal *dnSetSpeed_s*.

Parameter	Possible settings			Information
C05088	-214748.3647	Unit	214748.3647	Speed at the input • Read only • Calculated from the speed signal *dnSpeedIn_s*.
C05089	-214748.3647	Unit	214748.3647	Speed at the output • Read only • Calculated from the speed signal *dnSpeedOut_s*.

5.86 L_LdConvAxisV - speed ratio

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB transforms a speed between two axes.



Inputs

Identifier/data type	Information/possible settings	
bEnable BOOL	Activate function	
	FALSE	Output speed = "0".
	TRUE	Output speed = transformed input speed.
dnSpeedIn_s DINT	Selection of the master speed • At this input, the FB expects a speed in [rpm]. The machine parameters at the input *XAxisData* are used as speed reference. • C05165/1 indicates the master speed in the real unit of the machine.	
XAxisData	Machine parameters for scaling the master speed (master shaft) • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	
YAxisData	Machine parameters for scaling the setpoint speed (tool) • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**.	

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits14	Internal limitation for calculation
	Bit15	Error
	bits23	No valid axis data structure
dnSpeedOut_s DINT	Setpoint speed output in [rpm] • C05165/2 indicates the setpoint speed in the real unit of the machine.	

Parameter

Parameter	Possible settings				Information
C05160					Coupling measuring systems X and Y
	0	Coupling via feed constants			Lenze setting
	1	Coupling via cycles			
	2	Free coupling			
C05161/1	0		Unit	214748.3647	Cycle X • Read only
C05161/2	0		Unit	214748.3647	Cycle Y • Read only
C05162/1	0.0000			214748.3647	Coupling factor X • Initialisation: 0.0000
C05162/2	0.0000			214748.3647	Coupling factor Y • Initialisation: 0.0000
C05163/1	String of digits				User-defined unit X • Read only
C05163/2	String of digits				User-defined unit Y • Read only
C05165/1	-214748.3647		Unit/t	214748.3647	Speed X at the input • Read only • Calculated from the speed signal *dnSpeedIn_s*.
C05165/2	-214748.3647		Unit/t	214748.3647	Speed Y at the output • Read only • Calculated from the speed signal *dnSpeedOut_s*.
C05169	-2147483647			2147483647	Status • Display of the bit-coded output signal *dnState*.

Function

This FB transforms a speed between two axes, calculating the function of an electronic gearbox. The transformation works with remainder processing.

Operating mode 0 (unit coupling):

- In this operating mode, the FB couples both axes via the set feed constants.
- Via C05162/1 and C05162/2 the ratio of the corresponding units is set, i.e. the number of units of the X axis that corresponds to the number of units of the Y axis.

$$\text{Output value} = \text{Input value} \cdot \frac{\text{Gearbox constant Y}}{\text{Gearbox constant X}} \cdot \frac{\text{Feed constant X}}{\text{Feed constant Y}} \cdot \frac{\text{Coupling factor Y (C05162/2)}}{\text{Coupling factor X (C05162/1)}}$$

[5-43] Formula for operating mode 0

Operating mode 1 (clock equality):

- In this operating mode the FB makes use of the set cycles. The transformation displays one cycle of the X axis exactly in one cycle of the Y axis.

$$\text{Output value} = \text{Input value} \cdot \frac{\text{Gearbox constant Y}}{\text{Gearbox constant X}} \cdot \frac{\text{Feed constant X}}{\text{Feed constant Y}} \cdot \frac{\text{Cycle Y}}{\text{Cycle X}}$$

[5-44] Formula for operating mode 1

Operating mode 2 (free coupling):

- In this operating mode the set feed constants are not used for the transformation.
- Via C05162/1 and C05162/2 the revolution ratio is indicated, i.e. the number of revolutions of the master shaft that corresponds to the number of revolutions of the tool.

$$\text{Output value} = \text{Input value} \cdot \frac{\text{Gearbox constant Y}}{\text{Gearbox constant X}} \cdot \frac{\text{Coupling factor Y (C05162/2)}}{\text{Coupling factor X (C05162/1)}}$$

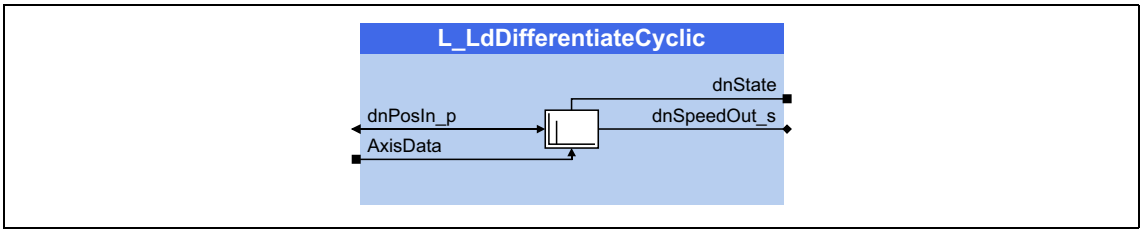
[5-45] Formula for operating mode 2

5.87 L_LdDifferentiateCyclic - cyclic differentiation

Function library:	LenzeLineDrive	**Do not use FB for new developments!** Replacement by L_SdDifferentiate.
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB differentiates a position with regard to a speed in consideration of the cycle.

- The cycle and shift factor for the position resolution are read out of the machine parameters (*AxisData*).
- The clock pulse is created with remainder processing.
- The FB is the counterpart of the FB L_LdIntegrateCyclic. (📖 322)



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Clocked position in [inc] • Scaling: 1 encoder revolution ≡ 2^{16} increments (or acc. to *AxisData*)
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

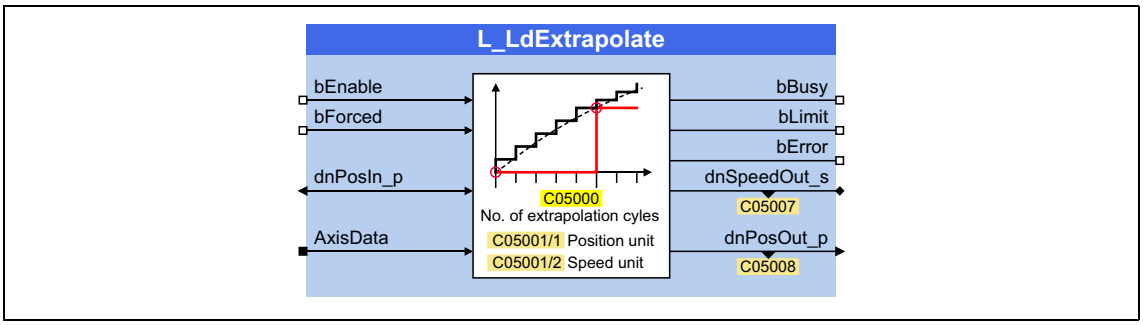
Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits10	Zero crossing active
	bits14	Internal limitation for calculation
	Bit15	Error
	bits18	No cycle available
	bits21	Input error (e.g. impermissible position jump or inverse direction of motion)
	bits23	No valid axis data structure
dnSpeedOut_s DINT	Speed • Scaling: 15000 rpm ≡ 2^{26} ≡ 67108864	

5.88 L_LdExtrapolate - extrapolation

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to extrapolate position information in order to e.g. compensate higher bus transmission cycles or smooth absolute value encoders with low resolution. A forced extrapolation is also possible.

- The LenzeServoDrive library (as of V02.00.xx.xx) provides the FB L_SdInterpolate for interpolation. (📖 483)



Inputs

Identifier/data type	Information/possible settings	
bEnable BOOL	Activate extrapolation	
	FALSE	The master value at *dnPosIn_p* is connected through 1:1 to the output *dnPosOut_p*.
	TRUE	The master value at *dnPosIn_p* is output to *dnPosOut_p* depending on the input *bForce* and the number of the extrapolation cycles set in C05000.
bForced BOOL	Forced extrapolation	
	FALSE	Extrapolation is only executed if the input value in the task cycle does not change, for instance, since the fieldbus cycle is longer and does not provide new position values in the task cycle.
	TRUE	Extrapolation is executed irrespective of master value changes.
dnPosIn_p DINT	Position of the external, and, as the case may be, insufficiently resolved master shaft in [inc].	
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	

Outputs

Identifier/data type	Value/meaning	
bBusy BOOL	Status signal "Extrapolation is executed"	
	TRUE	Extrapolation is executed.
bLimit BOOL	Status signal "Extrapolation limit reached"	
	TRUE	Extrapolation limit (C05000) reached.
bError BOOL	"Error" status signal	
	TRUE	Master position *dnPosIn_p* has exceeded the cycle defined in the machine parameters (*AxisData*).
dnSpeedOut_s DINT	Master speed output (in the cycle) in [rpm] • C05007 indicates the master speed in the real unit of the machine.	
dnPosOut_p DINT	Master position output (in the cycle) in [inc] • C05008 indicates the master position in the real unit of the machine.	

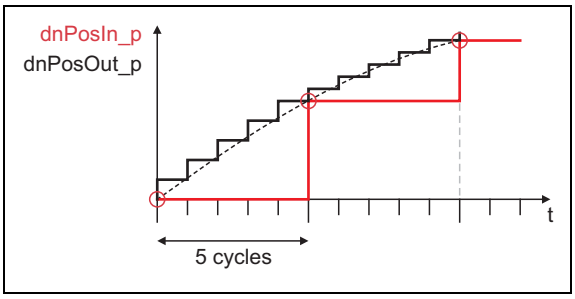
Parameter

Parameter	Possible settings			Information
C05000	1		127	Number of extrapolation cycles • Initialisation: 1
C05001/1 As of library V02.02.xx.xx	String of digits			Position unit • Read only
C05001/2 As of library V02.02.xx.xx	String of digits			Speed unit • Read only
C05007	-214748.3647	Unit/t	214748.3647	Speed at the output • Read only • Calculated from the speed signal *dnSpeedOut_s*.
C05008	-214748.3647	Unit	214748.3647	Position at the output • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C05009				Extrapolation cycles > selection • Read only
	0	No (OK)		
	1	Yes (error)		

5.88.1 Function

Compensation of bus transmission cycles > Sampling rate

If position information (e.g. the vertical shaft position) is exchanged between master and slave drives via a bus system with a transmission cycle higher than the sampling rate or between tasks with different cycle times, the setpoints cannot be processed in the control cycle of the drives. As a result, the position does not change linearly but stepwise at constant speed, which causes torque impulses.



[5-46] Example: Signal characteristic with five extrapolation cycles

- This FB serves to smooth the stepping by extrapolation.
- The number of the extrapolation cycles to be set in C05000 are defined according to the following formula:

$$C05000 = \frac{\text{Bus cycle [ms]}}{\text{Task cycle [ms]}}$$

Note!

Ensure that an integer value results for the number of extrapolation cycles so that the position information increases linearly and a speed calculation can be carried out correctly. In the case of integer conditions, an alternating speed characteristic results.

- If a smaller value is set than calculated according to the formula mentioned above, the extrapolation limit is reached and this is displayed by the output *bLimit*. In this case, *dnPosIn_p* is directly connected through to *dnPosOut_p*.
- If a higher value is set than calculated according to the formula mentioned above, it has no impact.

- By setting *bEnable* to TRUE, the extrapolation is activated.
- If *bEnable* is set to FALSE, the input signal is only looped through or the difference for speed output is created.

Smoothing of a low-resolution absolute value encoder

If the resolution of the absolute value encoder is so low that a new setpoint is not available for every new task call, this FB serves to "fine-interpolate" between the setpoints for a better smooth running.

- For this purpose enter instead of the bus cycle time the resolution converted at minimum speed into the formula mentioned above for calculating the extrapolation cycles.

Forced extrapolation

Forced extrapolation means that it is extrapolated up to the maximum value in C05000 irrespective whether the input signal has changed or not.

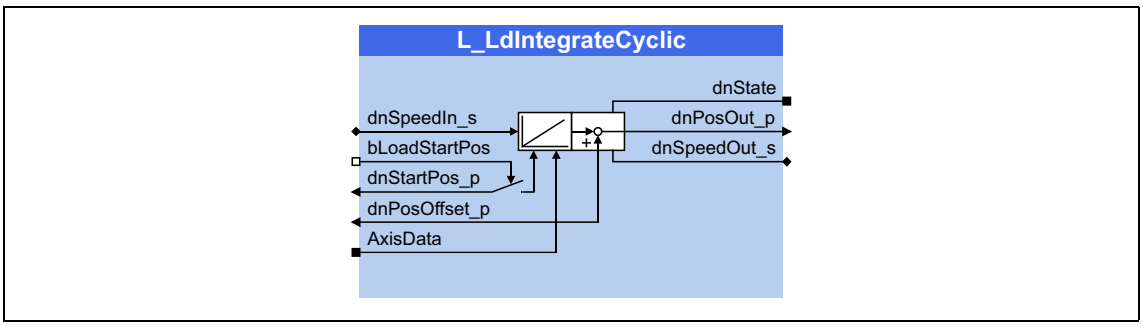
- The forced extrapolation is activated by setting *bForced* to TRUE.

5.89 L_LdIntegrateCyclic - cyclic integration

Function library:	LenzeLineDrive	**Do not use FB for new developments!** Replacement by L_SdIntegrateAxis.
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB integrates a speed in a position (by analogy with the FB L_SdIntegrate) in consideration of the cycle.

- It is possible to add a position offset to the output position, e.g. for trimming purposes. This function corresponds exactly to the FB L_LdAddOffsetCyclic. (📖 308)
- The cycle and shift factor for the position resolution are read out of the machine parameters (*AxisData*).
- The clock pulse is created with remainder processing.
- The FB is the counterpart of the FB L_LdDifferentiateCyclic. (📖 318)



Inputs

Identifier/data type	Information/possible settings
dnSpeedIn_s DINT	Speed • 15000 rpm ≡ 2^{26} ≡ 67108864
bLoadStartPos BOOL	Load the integrator with starting position TRUE: Load the integrator with the value at the *dnStartPos_p* input.
dnStartPos_p DINT	Starting position in [inc] with which the integrator is loaded by setting *bLoadStartPos* to TRUE.
dnPosOffset_p DINT	Position offset in [inc]
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510) • If this input is released, the following scaling applies: 1 motor revolution ≡ 2^{16}

Outputs

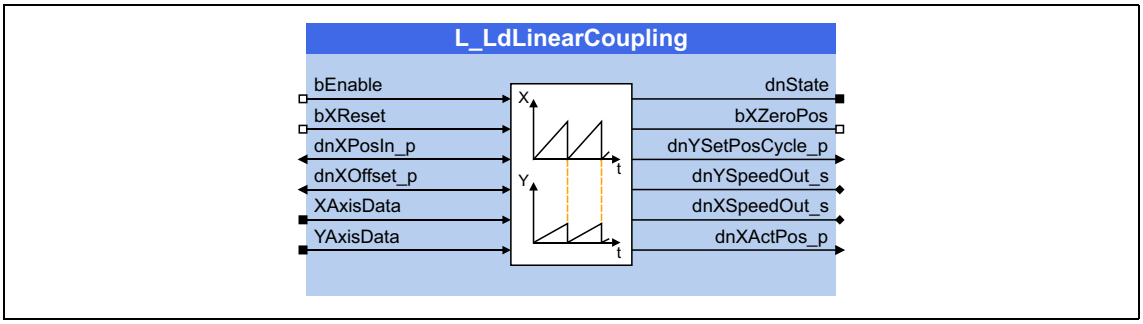
Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits10	Zero crossing active
	bits14	Internal limitation for calculation
	Bit15	Error
	bits18	No cycle available
	bits21	Input error (e.g. impermissible position jump or inverse direction of motion)
	bits23	No valid axis data structure
dnPosOut_p DINT	Position output in [inc]	
dnSpeedOut_s DINT	Speed output in [rpm] • Under consideration of the selected position offset.	

5.90 L_LdLinearCoupling - master value connection

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves as master value connection for the "Synchronism" application.

- The *dnYSetPosCycle_p* position value (Y cycle of the Y axis) is synchronous to the *dnXPosIn_p* master value (cycle of the X axis), i.e. if the master value overflows, a new cycle also starts for the *dnYSetPosCycle_p* output.
- The FB makes use of the set cycles of *XAxisData* and *YAxisData*. The transformation displays one cycle of the X axis exactly with residual values in one cycle of the Y axis.
- Internal calculation:
  dnYSetPosCycle = *dnXPosIn_p* * (cycle Y / cycle X)



Inputs

Identifier/data type	Information/possible settings	
bEnable BOOL	Activate function	
	FALSE	Function has been deactivated. • The *dnXSpeedOut_s* and *dnXActPos_p* outputs continue to follow the *dnXPosIn_p* and *dnXOffset_p* input positions. • A value of "0" is output at the *dnYSpeedOut_s* output. • The *dnYSetPosCycle_p* output is maintained on the last value. **Caution:** Consequently, a position step change on the master value end will not be displayed at the *dnYSpeedOut_s* and *dnYSetPosCycle_p* outputs!
	TRUE	Function is activated.
bXReset BOOL	Reset the master value selection *dnXPosIn_p* internally	
	FALSE	Output corresponds to *dnXPosIn_p* + *dnXOffset_p*.
	TRUE	Master value selection *dnXPosIn_p* is internally set to and held at "0". Output corresponds to *dnXOffset_p*.
dnXPosIn_p DINT	Master value selection in [inc] • C05186 indicates the master position in the real unit of the machine.	
dnXOffset_p DINT	Phase offset selection for the master value in [inc] • The phase offset is applied immediately. • C05187 indicates the phase offset in the real unit of the machine.	
XAxisData	Machine parameters for scaling the master value • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	
YAxisData	Machine parameters for scaling the setpoint • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**.	

Outputs

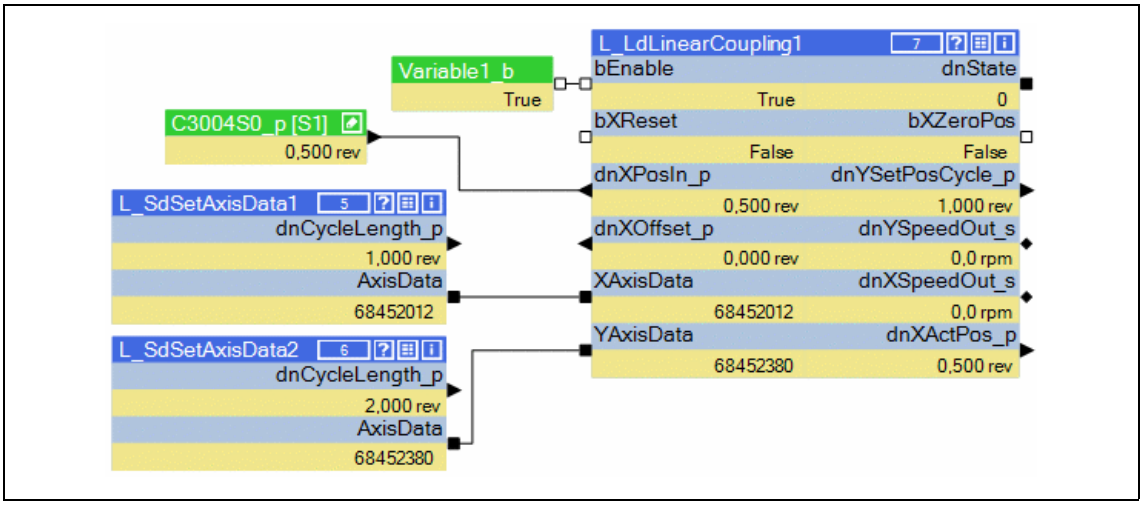
Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits10	Zero crossing active
	bits14	Internal limitation for calculation
	Bit15	Error
	bit16	Starting position is beyond the cycle
	bits18	No cycle available
	bits22	No cycle available for Y axis
	bits23	No valid axis data structure
bXZeroPos BOOL	Status signal "Zero position/zero crossing" • The input signal *dnXActPos_p* is the reference.	
	FALSE	Master value is at any position in the master value cycle (register)
	TRUE	**Permanently:** Master value is at the zero position of the master value integrator. **For one cycle:** Zero crossing of the master value integrator.
dnYSetPosCycle_p DINT	Setpoint position output in [inc] • Output of the position setpoint in the setpoint cycle as a function of the master value (master value cycle). • The return of master value cycle and setpoint cycle depends on the corresponding cycles set. Thus, the return is firmly linked to the master value cycle. • If the FB is deactivated (*bEnable* = FALSE), the output is maintained on the last set-point position. • C05189 indicates the setpoint position in the real unit of the machine.	
dnYSpeedOut_s DINT	Setpoint speed output in [rpm] • If the FB is deactivated (*bEnable* = FALSE), a value of "0" is output. • C05184 indicates the setpoint speed in the real unit of the machine.	
dnXSpeedOut_s DINT	Master speed output in [rpm] • If the FB is deactivated (*bEnable* = FALSE), the resulting master speed is output as before. • C05183 indicates the master speed in the real unit of the machine.	
dnXActPos_p DINT	Master position output in [inc] • Results from MODULO(*dnXOffset_p* + *dnXPosIn_p*) • If the FB is deactivated (*bEnable* = FALSE), the resulting master position is output as before. • C05188 indicates the master position in the real unit of the machine.	

Parameter

Parameter	Possible settings			Information
C05180/1 As of library V02.02.xx.xx	String of digits			X position unit • Read only
C05180/2 As of library V02.02.xx.xx	String of digits			X speed unit • Read only
C05180/3 As of library V02.02.xx.xx	String of digits			Y position unit • Read only
C05180/4 As of library V02.02.xx.xx	String of digits			Y speed unit • Read only
C05181	-214748.3647		214748.3647	Status • Display of the bit-coded output signal *dnState*.

Parameter	Possible settings			Information
C05183	-214748.3647	Unit/t	214748.3647	Speed X at the output • Read only • Calculated from the speed signal *dnXSpeedOut_s*.
C05184	-214748.3647	Unit/t	214748.3647	Speed Y at the output • Read only • Calculated from the speed signal *dnYSpeedOut_s*.
C05186	-214748.3647	Unit	214748.3647	Position X at the input • Display of the *dnXPosIn_p* input signal in the real unit of the machine.
C05187	-214748.3647	Unit	214748.3647	Trimming X • Display of the *dnXOffset_p* input signal in the real unit of the machine.
C05188	-214748.3647	Unit	214748.3647	Position X at the output • Display of the output signal *dnXActPos_p* in the real unit of the machine.
C05189	-214748.3647	Unit	214748.3647	Position Y at the output • Display of the output signal *dnYSetPosCycle_p* in the real unit of the machine.

5.90.1 Interconnection example



[5-47] Interconnection example

5.91 L_LdMarkSync - mark synchronisation

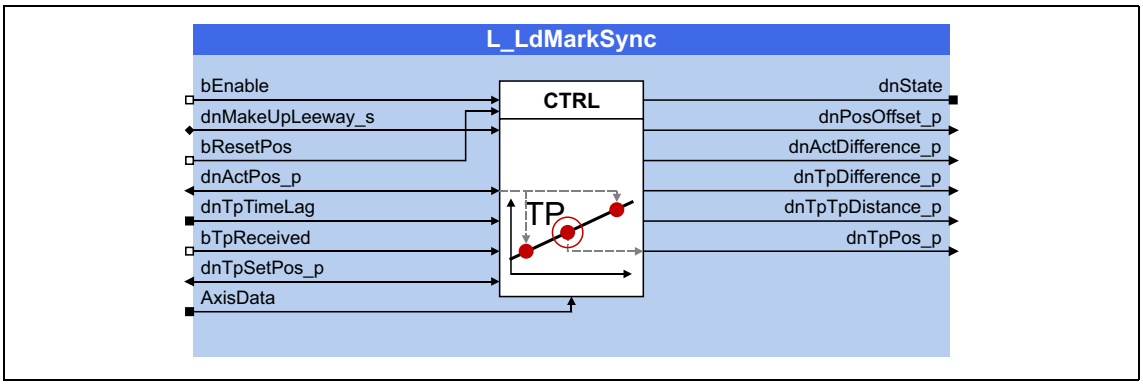
Function library:	LenzeLineDrive	
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

When mark synchronisation is performed, this FB calculates the position at which a mark signal (touch probe signal) occurred. Furthermore, the remaining distance to the setpoint position of the mark is observed and output.

- High-precision detection of the actual TP position for which a mark signal occurred.
- Application only in conjunction with modulo measuring systems.
- Comparison with a TP setpoint position at which the mark ideally occurs.
- Calculation of the TP deviation (= deviation between the actual TP position and the TP setpoint position).
- Feedback of the correcting signal and monitoring of the outstanding TP deviation.
- Distance measurement between two mark signals.

Note!

The generation of the correcting motion is not part of the L_LdMarkSync FB. For this purpose, use e.g. the L_LdPosCtrlLin FB (time-controlled correction) or the L_CamSyncIn FB (path-controlled correction).



Inputs

Identifier/data type	Information/possible settings	
bEnable BOOL	Activate FB	
	FALSE	FB is deactivated: Touch probe signals are not evaluated. A value of "0" is output at the *dnActDifference_p*, *dnTpDifference_p*, *dnTpTpDistance_p* and *dnTpPos_p* outputs. The value at the *dnPosOffset_p* output remains unchanged.
	TRUE	FB is activated: After a touch probe signal has occurred, the outputs display the described quantities. The monitoring of the remaining TP deviation (the *dnActDifference_p* output signal) is switched on.

Identifier/data type	Information/possible settings	
dnMakeUpLeeway_s DINT	Selection (addition) of the compensating speed as a speed signal • Scaling: $2^{26} \equiv 15000$ [rpm] • The *dnActDifference_p* output is set using this speed. The following applies from library V02.04.xx.xx: • The signal at this input is internally processed only as a sum so that the remaining TP distance (the *dnActDifference_p* output signal) can in fact be reduced with the defined *dnMakeUpLeeway_s* speed but never be increased. • If the remaining TP deviation *dnActDifference_p* has been compensated to zero completely, the signal does not have any effect anymore at this input.	
bResetPos BOOL	Reset internal offset integrator	
	FALSE	The internal offset integrator (the *dnPosOffset_p* output = sum of the compensated TP correction distances so far) is set via the performed TP correcting motions.
	TRUE	The internal offset integrator (the *dnPosOffset_p* output) is set to zero.
dnActPos_p DINT	Input for the acceptance of the current position of the system in [increments] for which a touch probe position is to be determined: • If the master value is corrected (x touch probe), the master position can be connected, e.g. the position output of the master value integrator. • If the drive axis is corrected (y touch probe), the actual position value, *FDB_dnActualPos_p*, can be connected. • Please observe that the modulo measuring system connected to the *AxisData* input must go with the position value specified here.	
dnTpTimeLag DINT	Input for accepting the touch probe time stamp • Connect this input with the output *dnTouchProbeTimeLag* of the corresponding touch probe system block. • The time stamp displays the time that has passed between the touch probe event and the calling of the ApplicationTask. • Scaling: $2^{20} \equiv 1048576 \equiv 1$ [ms]	
bTpReceived BOOL	Input for taking over the status "Touch probe detected" • Connect this input with the output *dnTouchProbeReceived* of the corresponding touch probe system block.	
	FALSE↗TRUE	Touch probe detected. • The actual position at the time when the touch probe is detected is calculated and output at the *dnTpPos_p* output. • The TP difference between the TP setpoint position, *dnTpSetPos_p*, and the actual position, *dnTpPos_p*, is output at the *dnTpDifference_p* output. • The remaining TP difference, *dnActDifference_p*, is set to the TP difference, *dnTpDifference_p*. • If a touch probe has already been detected in the last cycle, the position difference between the current and the previous touch probe is output at the *dnTpTpDistance_p* output.

Identifier/data type	Information/possible settings
dnTpSetPos_p DINT	Touch probe setpoint position in [inc] • At this input, specify the position at which the touch probe ideally occurs. • If the touch probe is detected at its (ideal) TP setpoint position, a correcting motion is not required (TP difference *dnTpDifference_p* = 0).
AxisData	Specification of the measuring system which is used to evaluate the touch probe event. The following sources can be used for this input: • If the master position is corrected (x touch probe), an arbitrary measuring system can be defined via the L_SdSetAxisData FB. In this case, its *AxisData* output must be connected to this input. • If an axis position is corrected (y touch probe), the *DI_AxisData* output of the **LS_DriveInterface** SB must be connected to this input. • Some of the cam technology function blocks (e.g. the L_CamProfiler FB) output the measuring systems defined via »Cam Designer«. In this case, the corresponding measuring system output must be connected to this input. **Note:** The connected measuring system must be a modulo measuring system.

Outputs

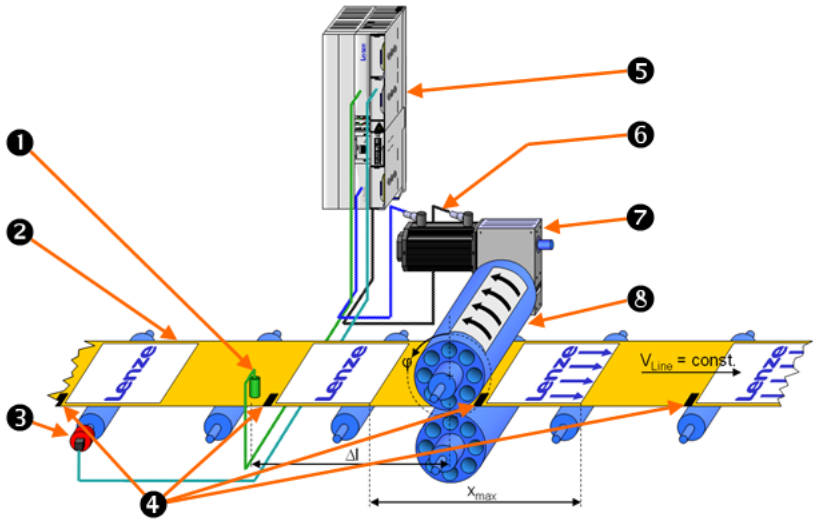
Identifier/data type	Value/meaning	
dnState DINT	Status	
	Bit15	Group error (see bit 16 ... 31)
	bits18	The specified modulo measuring system (the *AxisData* input) does not have a valid cycle (cycle = 0).
	bits20	The TP setpoint position at the *dnTpSetPos_p* input is outside of the permissible cycle of the modulo measuring system (the *AxisData* input).
	bits23	Error in the measuring system: The pointer at the *AxisData* input does not refer to a valid axis data structure.
dnPosOffset_p DINT	Aggregated position offset in [increments] (Aggregation of all correcting motions performed so far) • This output outputs an unlimited correction position. • If the TP is corrected at the axis side (y touch probe), the output can directly be connected to the *FDB_dnPosOffset_p* input of the **LS_Feedback** SB when the axis needs to be corrected.	
dnActDifference_p DINT	Remaining position difference in [increments] • If a touch probe is detected via the *bTpReceived* input, this output is set to the detected TP deviation, *dnTpDifference_p*. • Then, the remaining TP deviation is compensated to zero via the *dnMakeUpLeeway_s* input if a correcting motion is reported (correction distance monitoring). • The shorter distance to the TP setpoint position is always output. (Caution: The function block requires a modulo measuring system to work correctly!)	
dnTpDifference_p DINT	Last TP deviation in [inc] (TP deviation between the TP setpoint position, *dnTpSetPos_p*, and the actual position, *dnTpPos_p*, at which the touch probe has been detected.) • If a touch probe is detected via the *bTpReceived* input, the dnActDifference_p output is set to this value additionally. • The shorter distance to the TP setpoint position is always output. (Caution: The function block requires a modulo measuring system to work correctly!)	

Identifier/data type	Value/meaning
dnTpTpDistance_p DINT	Distance between the two touch probes last detected in [increments] • May e.g. serve to determine the print register (distance between two print marks). • Please make sure that the touch probe sensor is positioned in such a way that no cycle without touch probe detection is possible between two detected touch probes. In this case, the output would indicate a distance which is one cycle too short. This can be prevented by positioning the touch probe sensor in the middle of the cycle.
dnTpPos_p DINT	Detected TP position in [inc] (Actual TP position at which the current touch probe has been detected.) • The distance to the TP setpoint position is calculated using the actual TP position that is output here: *dnTpDifference_p = dnTpSetPos_p - dnTpPos_p*

5.91.1 Typical application

Many synchronous operations require a slave drive (tool) to move in synchronism with a master signal/master drive. This motion can be corrected using additional sensor technology.

In the following example, a print register has already been defined by print marks on the web. High-precision detection of these print marks in the controller is possible via fast digital inputs (touch probe inputs).



❶ Print-mark sensor
❷ Web including print register 0 ... x_{max} (print image: "Lenze")
❸ Master value encoder
❹ Print marks
❺ 9400 controller (slave drive = printing roller)
❻ Printing roller drive motor including feedback system
❼ Gearbox (slave drive)
❽ Printing roller including block (print image: "Arrows")

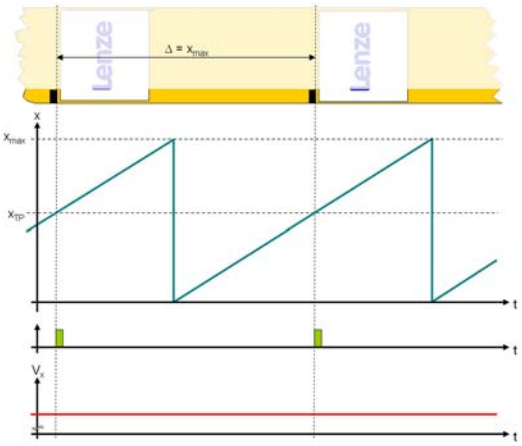
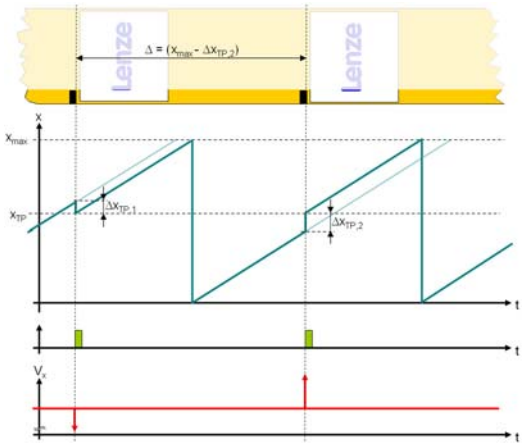
[5-48] Example: Printing roller drive including print-mark correction

Synchronisation to print marks

The "Arrows" print image is to be printed properly on to the "Lenze" print image on the web. For this purpose and in addition to the material speed signal (master value encoder), the printing roller drive receives a pulse from the print-mark sensor which is connected to one of the digital inputs of the controller.

If the distance Δl between the print-mark sensor and the twelve o'clock position ("kiss point") of the printing roller is known, the printing roller drive is able to correct the motion of the printing roller so that the "Arrows" print image is printed exactly at the required position on the web when the print-mark pulse is detected.

Since the production process in the example is periodical, one print-mark pulse appears per cycle. Ideally, the pulse always appears at a certain set position x_{TP} in the modulo cycle 0 ... x_{max}. If the print mark pulse is detected at another position than at its set position x_{TP}, there is a deviation Δx_{TP} between setpoint and actual position of the print mark pulse.

Signal characteristic 1: Print-mark pulses always appear exactly periodically with a distance of x_{max}.	**Signal characteristic 2:** Print-mark pulses appear approximately periodically (difference Δx_{TP} to x_{max}).
	
The touch probe signal is always detected at position $x = x_{TP}$ in the modulo cycle 0 ... x_{max}. In this case, position value x need not be corrected and shows an ideal saw-tooth profile if speed V_x is constant.	If the touch probe signals appear irregularly, the continuous modulo position x has not reached setpoint position x_{TP} yet, or is already past it. However, since the touch probe signal is always assigned to a defined setpoint position x_{TP}, position value x needs to be corrected accordingly: In the simplest case, position value x can be set to setpoint position x_{TP} if touch probe signal "hard" is detected. Thus, speed V_x shows corresponding discontinuities (delta pulses) at the times when the touch probe signals are detected.

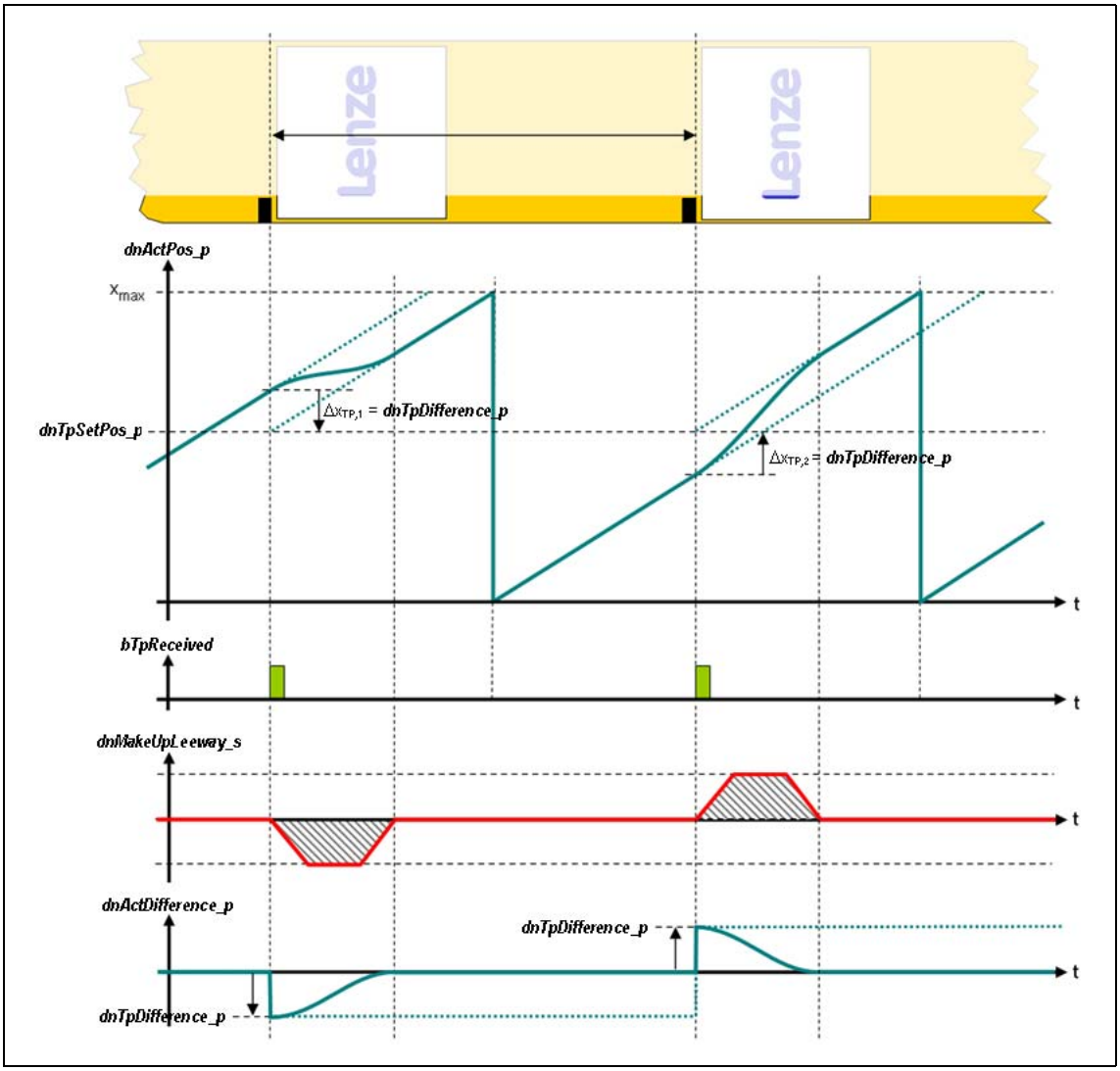
Tip!

Controlled compensation can be used to achieve "smooth" mark synchronisation to avoid delta pulses when correcting print marks.

▸ Interconnection example (📖 333)

5.91.2 Signal characteristics

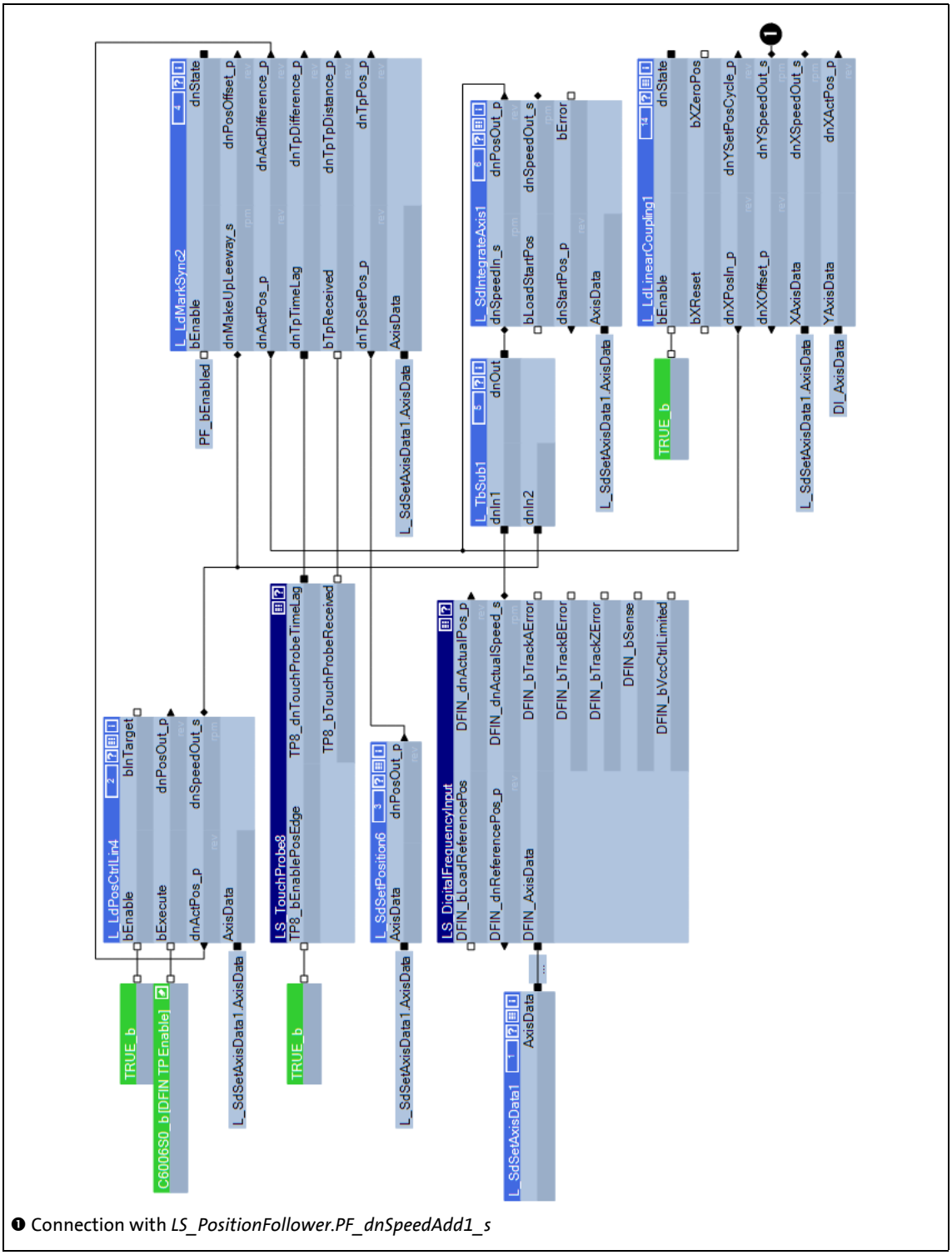
The following signal characteristics result during the current print-mark correction:



[5-49] Signal characteristics

5.91.3 Interconnection example

Interconnection example of a controlled compensation of print-mark deviation Δx_{TP} via a time-controlled positioning profile:



❶ Connection with *LS_PositionFollower.PF_dnSpeedAdd1_s*

[5-50] Model connection

Function block	Function
L_LdPosCtrlLin4	Time-controlled profile generation • Using the detected deviation (print-mark correction path), the L_LdPosCtrlLin FB generates a time-controlled speed setpoint. • The speed setpoint is provided at the *L_LdPosCtrlLin4.dnSpeedOut_s* output.
LS_TouchProbe8	Detection of the print-mark signal
L_SdSetPosition6	Selection of the set position of the print-mark sensor • The FB L_SdSetPosition converts the set position defined in the application unit [unit] into an incremental value. • The reference measuring system is the master measuring system which is defined via the FB L_SdSetAxisData.
L_LdMarkSync2	Detection of the correction distance Δx_{TP} • The FB L_LdMarkSync detects the correction distance Δx_{TP} based on the current x position and the set position of the print mark signal x_{TP} if a print mark pulse is detected via the SB **LS_TouchProbe8**. • The correction distance Δx_{TP} is provided at the *L_LdMarkSync2.dnActDifference_p* output.
L_SdSetAxisData1	Definition of the master measuring system • Using the L_SdSetAxisData FB, machine parameters of a higher-level drive can be displayed.
LS_DigitalFrequencyInput	Detection of the digital frequency and output as speed signal
L_TbSub1	Injection of the correction speed • The L_TbSub FB performs a subtraction without limitation.
L_SdIntegrateAxis1	Integration of the resulting speed signal to a master position (x position) • The L_SdIntegrateAxis FB integrates a speed into a position.
L_LdLinearCoupling1	Synchronous function • The FB L_LdLinearCoupling establishes the coupling between master and axis measuring system.

Operating mode

The core function of the interconnection is contained in the FB L_LdMarkSync2: On the basis of the current x position (integrator output *L_SdIntegrateAxis1.dnPosOut_p*) and the set position of the print-mark signal x_{TP} (output *L_SdSetPosition6.dnPosOut_p*) the FB detects the correction distance Δx_{TP} when a print mark pulse is detected.

In the example, the print-mark sensor is connected to the digital input DI8, touch probe is therefore detected via the **LS_TouchProbe8** system block.

The correction distance Δx_{TP} is led from the signal output *L_LdMarkSync2.dnActDifference_p* to the profile generator L_LdPosCtrlLin4 which generates a correction profile $V_{x,comp}$.

The correction speed $V_{x,comp}$ is superimposed on the real master speed (system variable *DFIN_dnActualSpeed_s*) via the FB L_TbSub1 and then integrated in the integrator block L_SdIntegrateAxis1 modulo to a master position 0 ... x_{max}.

Note!

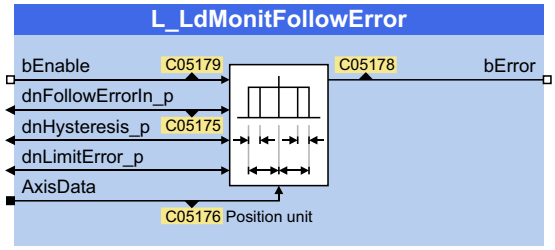
Touch probe corrections cannot be applied to absolute master positions, as for instance if selected via MotionBus.

A touch probe correction on the master value side requires a master speed as master value selection (e.g. via digital frequency coupling).

5.92 L_LdMonitFollowError - following error monitoring

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to implement a following error monitoring.



Inputs

Identifier/data type	Information/possible settings
bEnable BOOL	Activate monitoring function TRUE: Monitoring function is activated.
dnFollowErrorIn_p DINT	Following error in [inc] • C05175 indicates the following error in the real unit of the machine.
dnHysteresis_p DINT	Hysteresis for switching threshold in [inc]
dnLimitError_p DINT	Switching threshold for positive and negative following error in [inc]
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning
bError BOOL	Status signal "following error" TRUE: Following error of a higher switching threshold.

Parameter

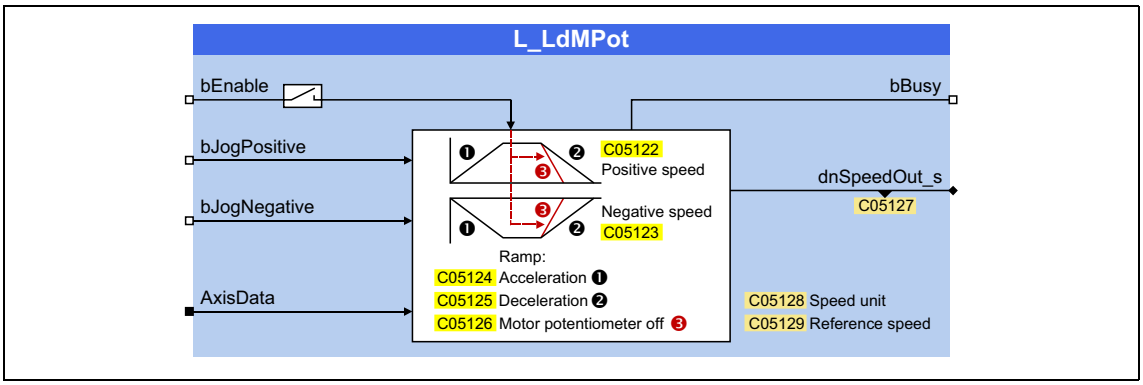
Parameter	Possible settings			Information
C05175	-214748.3647	Unit	214748.3647	Following error at the input • Display of the *dnFollowErrorIn_p* input signal in the real unit of the machine.
C05176 As of library V02.02.xx.xx	String of digits			Position unit • Read only

Parameter	Possible settings		Information
C05178			Following error outside the tolerance • Read only
	FALSE	Following error of a lower switching threshold	
	TRUE	Following error of a higher switching threshold	
C05179			Following error monitoring • Read only
	FALSE	Deactivated	
	TRUE	Activated	

5.93 L_LdMPot - master value adjustment

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to execute a master value adjustment.



Inputs

Identifier/data type	Information/possible settings	
bEnable BOOL	Activate master value adjustment	
	FALSE	Master value adjustment is/will be deactivated. • Output *dnSpeedOut_s* is braked to standstill via the disable ramp, if required.
	TRUE	Master value adjustment is activated.
bJogPositive BOOL	Adjustment in positive traversing direction	
	FALSE	Output *dnSpeedOut_s* is braked to standstill via the deceleration ramp.
	TRUE	Output *dnSpeedOut_s* is led to positive speed (C05122) via the start-up ramp.
bJogNegative BOOL	Adjustment in negative traversing direction	
	FALSE	Output *dnSpeedOut_s* is braked to standstill via the deceleration ramp.
	TRUE	Output *dnSpeedOut_s* is led to negative speed (C05123) via the start-up ramp.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	

Outputs

Identifier/data type	Value/meaning	
bBusy BOOL	Status signal "Master value adjustment is active"	
	FALSE	Master value adjustment is deactivated. • Output *dnSpeedOut_s* = "0".
	TRUE	Master value adjustment is activated. • Output *dnSpeedOut_s* ≠ "0".
dnSpeedOut_s DINT	Master speed output in [rpm] • C05127 indicates the master speed in the real unit of the machine.	

Parameter

Parameter	Possible settings			Information
C05122	-214000.0000	Unit/t	214000.0000	Positive speed • Initialisation: 1.0000 unit/t
C05123	-214000.0000	Unit/t	214000.0000	Negative speed • Initialisation: 1.0000 unit/t
C05124	0.001	s	130.000	Ramp: Acceleration • Initialisation: 1.000 s
C05125	0.001	s	130.000	Ramp: Deceleration • Initialisation: 1.000 s
C05126	0.001	s	130.000	Ramp: Motor potentiometer off • Initialisation: 1.000 s
C05127	-214748.3647	Unit	214748.3647	Speed at the output • Read only • Calculated from the speed signal *dnSpeedOut_s*.
C05128 As of library V02.02.xx.xx	String of digits			Speed unit • Read only
C05129 As of library V02.02.xx.xx	-214748.3647	Unit/t	214748.3647	Reference speed • Read only • For input orientation for speed entries.

5.93.1 Monitoring for counter overflow

This function extension will be available from library V2.02!

If the machine parameters transferred to the FB via the *AxisData* input are changed afterwards (e.g. the gearbox factors), this may cause a counter overflow when the speed values defined via C05122 and C05123 are converted from the real unit to the internal unit.

The FB causes a corresponding error message in case of an internal counter overflow due to a changed measuring system:

Error number		Error message in the logbook	Response
61669382	0x3AD0006	L_LdMPot:int.speed overflow (LS_DriveInterface)	Error
61669510	0x3AD0086	L_LdMPot:int.speed overflow (L_SdSetAxisData)	Error

5.94 L_LdPosCtrlLin - Master value adjustment via position

Function library:	LenzeLineDrive	
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB permits the time-controlled profile generation with linear acceleration/deceleration ramps. The following positioning functions, for instance, can be implemented via this FB:

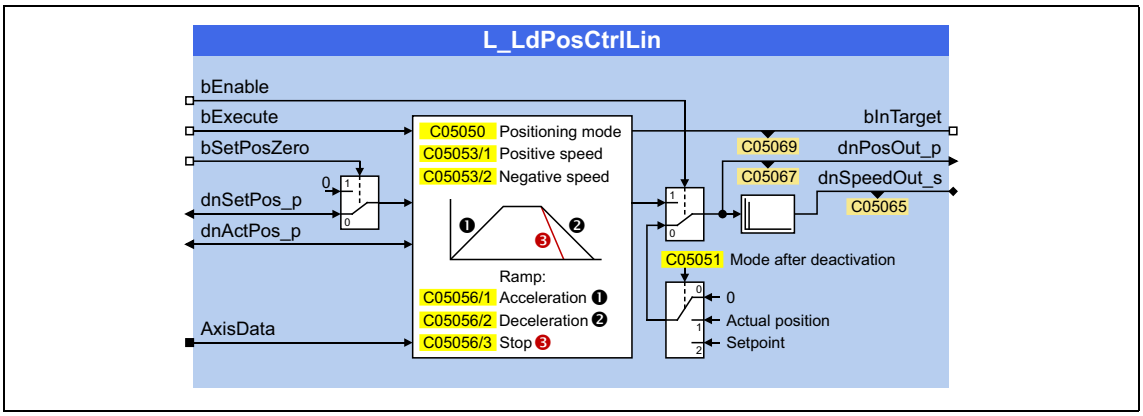
- Online trimming of offset positions on the master and drive end
- Positioning of the drive to the position of the electrical shaft
- Position override function: Separate the drive from the curve position and put it in a safe position

Note!

The FB always operates in the absolute measuring system independent of the setting in the *AxisData* machine parameters. Thus, the *dnPosOut_p* position output is not limited to a cycle length in case of a Modulo system.

Only the position value at the *dnPosOut_p* position output should be used in the application.

The *dnSpeedOut_s* speed output must not be integrated because the integrated position cannot be displayed completely using the *dnPosOut_p* output for internal technical reasons!



Inputs

Identifier/data type	Information/possible settings	
bEnable BOOL	Activate FB	
	FALSE	FB deactivated: • The *dnPosOut_p* output position and the *dnSpeedOut_s* output speed behave according to the deactivation mode set in C05051. • The *bInTarget* status signal is reset. **Caution:** Depending on the set deactivation mode, sudden changes may occur at the position and speed output when the FB (*bEnable* = TRUE↘FALSE) is deactivated! In this case, make sure that no drive motion is derived from these output signals!
	TRUE	FB activated: • The *dnPosOut_p* output position and the *dnSpeedOut_s* output speed display the required position profile according to the profile parameters and control signals set.
bExecute BOOL	Execute positioning according to the positioning mode set	
	FALSE	Abort of positioning (profile generator inactive): • When a running positioning mode is aborted, the speed is led to zero via the stop ramp set in C05056/3.
	FALSE↗TRUE	Executing the positioning: • Positioning is executed depending on the set profile parameters (ramp times, maximum speeds, positioning modes). • If the positioning mode 1 or 2 is set in C05050, a renewed FALSE↗TRUE edge is required at this control input for the start of each positioning profile.
bSetPosZero BOOL	Change over to internal zero position	
	FALSE	The *dnSetPos_p* input signal is used as position target for the traversing profile.
	TRUE	Zero is used as position target for the traversing profile.
dnSetPos_p DINT	Selection of the target position in [increments]: • Absolute position target (for absolute positioning modes 0 and 1) or traversing distance (for relative positioning mode 2).	
dnActPos_p DINT	Feedback of the current position in [increments] • The FB **L_LdPosCtrlLin** generates positioning profiles on the basis of setpoints only. Thus, only connect setpoint-based position signals to this input. • In the simplest case, directly return the *dnPosOut_p* position output to the *dnActPos_p* input. • Never connect this input to real position signals (e.g. *FDB_dnActualPos_p*).	
AxisData	Machine parameters • If the FB is to relate to the measuring system of the drive axis, connect this input to the output *DI_AxisData* of the **LS_DriveInterface** SB for accepting the machine parameters of the drive/motor. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510) • For a cam application, this FB can receive the *XAxisData/YAxisData* X/Y axis data structure of the L_CamProfiler FB. (📖 162)	

Outputs

Identifier/data type	Value/meaning	
bInTarget BOOL	Status signal "target position reached" • Display parameter: C05069 • The status signal is reset when the *bEnable* input is reset to FALSE. **Interconnection note:** Do not apply an inverted status signal to the enable input *PF_bEnable* of the SB **LS_PositionFollower**, since a residual path is not travelled anymore after the enable is cancelled (the setpoints *PF_dnPositionSet_p* and *PF_dnSpeedAdd1_s* are rejected in the same cycle).	
	FALSE	The traversing profile is still being executed or has been aborted, the current position *dnActPos_p* and the target position *dnSetPos_p* differ from each other.
	TRUE	The traversing profile has been completed, the current position *dnActPos_p* has reached the target position *dnSetPos_p*.
dnPosOut_p DINT	Output position in [increments] • Set position in traversing profile • In the case of simple positioning tasks (e.g. time-controlled offset adjustment), directly return this position output to the *dnActPos_p* input. • C05067 indicates the output position in the real unit of the machine. **Caution:** Depending on the deactivation mode set in C05051, sudden changes may occur at the position and speed out when the FB (*bEnable* = TRUE↘FALSE) is deactivated! In this case, make sure that no drive motion is derived from these output signals!	
dnSpeedOut_s DINT	Output speed given as speed in [rpm] • Setpoint speed in traversing profile • This signal directly results from the differentiation of the output position *dnPosOut_p*. • C05065 indicates the output speed in the real unit of the machine.	

Parameters

Parameter	Possible settings		Info
C05050			Positioning mode
	0	Absolute positioning without limit stop • When positioning is started (*bExecute* = TRUE), the outputs *dnPosOut_p* and *dnSpeedOut_s* are led to the target position with the set profile parameters. • The traverse path results from the difference between the current position *dnActPos_p* and the target position *dnSetPos_p*. • A new positioning profile is started as soon as the target position changes. An edge change at the *bExecute* control input is not required.	Lenze setting
	1	Absolute positioning with limit stop: • When positioning is started (*bExecute* = TRUE), the outputs *dnPosOut_p* and *dnSpeedOut_s* are led to the target position with the set profile parameters. • The traverse path results from the difference between the current position *dnActPos_p* and the target position *dnSetPos_p*. • After the set position has been reached (*bInTarget* = TRUE), a new positioning process requires a new FALSE↗TRUE edge at the *bExecute* control input.	
	2	Relative positioning with limit stop: • When positioning is started (*bExecute* = TRUE), the outputs *dnPosOut_p* and *dnSpeedOut_s* are led to the target position with the set profile parameters. • The traverse path is directly determined via the *dnSetPos_p* input. • After the set position has been reached (*bInTarget* = TRUE), a new positioning process requires a new FALSE↗TRUE edge at the *bExecute* control input.	

Parameter	Possible settings			Info
C05051				Mode after deactivation
	0	Positioning function deactivated: • The position output *dnPosOut_p* is set to zero. • The *bInTarget* status signal is reset to FALSE.		Lenze setting
	1	Positioning function changes in standby status: • The *dnPosOut_p* positioning output is kept on the current position *dnActPos_p*. • The *bInTarget* status signal is reset to FALSE.		
	2	Positioning function changes in standby status: • If *bSetPosZero* = FALSE: The *dnPosOut_p* position output is kept on the set position *dnSetPos_p*. • If *bSetPosZero* = TRUE: The *dnPosOut_p* position output is kept to zero. • The *bInTarget* status signal is reset to FALSE.		
C05052/1 From library V02.02.xx.xx onwards	Character string			Position unit • Read only
C05052/2 From library V02.02.xx.xx onwards	Character string			Speed unit • Read only
C05053/1	0.0000	Unit/t	214748.3647	Positive speed • Initialisation: 0.02 unit/t
C05053/2	0.0000	Unit/t	214748.3647	Negative speed • Initialisation: 0.02 unit/t
C05054 From library V02.02.xx.xx onwards	-214748.3647	Unit/t	214748.3647	Reference speed • Read only (determined from the machine parameters applied to the *AxisData* input) • For input orientation for speed entries.
C05056/1	0.010	s	130.000	Ramp: Acceleration • Initialisation: 1.000 s • Reference: • For acceleration in the positive direction: positive speed (C05053/1) • For acceleration in the negative direction: negative speed (C05053/2)
C05056/2	0.010	s	130.000	Ramp: Deceleration • Initialisation: 1.000 s • Reference: • For deceleration in the positive direction: positive speed (C05053/1) • For deceleration in the negative direction: negative speed (C05053/2)

Parameter	Possible settings			Info
C05056/3	0.010	s	130.000	Ramp: Stop • Initialisation: 1.000 s • Reference: • For deceleration in the positive direction: positive speed (C05053/1) • For deceleration in the negative direction: negative speed (C05053/2)
C05065	-2147483647	Unit/t	2147483647	Speed at the output • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.
C05067	-214748.3647	Unit	214748.3647	Position at the output • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C05069				Set position reached • Display of the *bInTarget* output signal.
	0	Set position not yet reached.		
	1	Set position reached.		

5.94.1 Monitoring for counter overflow

This function extension will be available from library V2.02!

If the machine parameters transferred to the FB via the *AxisData* input are changed afterwards (e.g. the gearbox factors), this may cause a counter overflow when the speed values defined via C05053/1...2 are converted from the real unit to the internal unit.

The FB causes a corresponding error message in case of an internal counter overflow due to a changed measuring system:

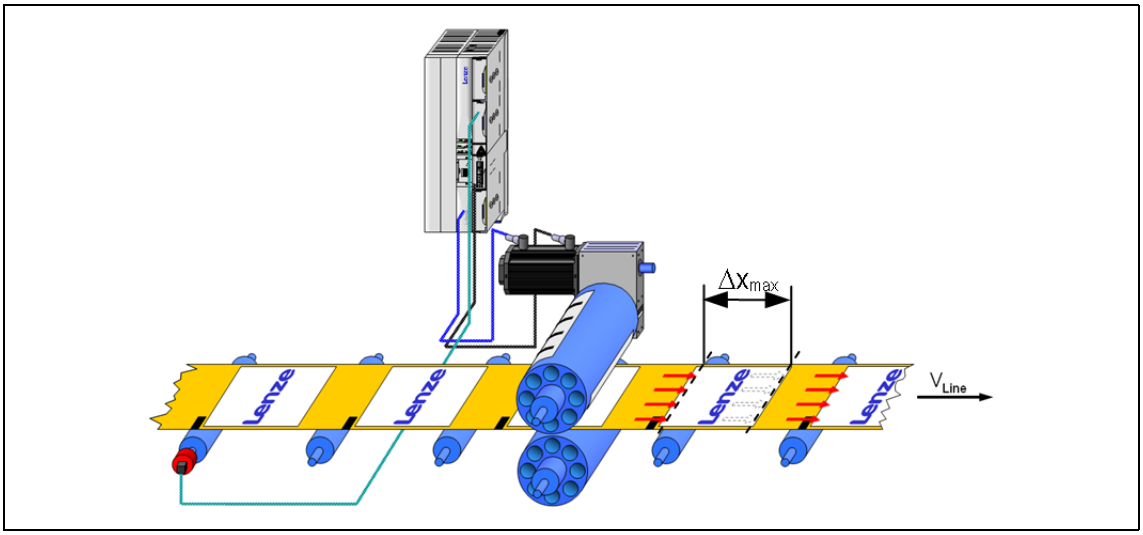
Error number		Error message in the logbook	Response
61669383	0x3AD0007	L_LdPosCtrlLin:int.speed overflow (LS_DriveInterface)	Error
61669511	0x3AD0087	L_LdPosCtrlLin:int.speed overflow (L_SdSetAxisData)	Error

5.94.2 Interconnection examples

5.94.2.1 Printing roller trimming via x offset

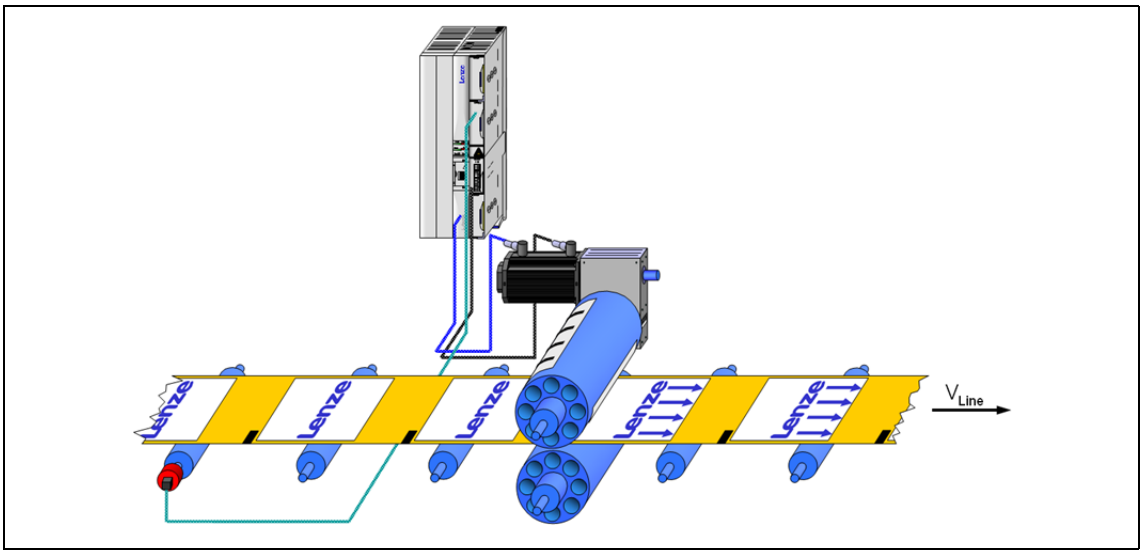
In many cases, an angular trimming of the slave drive to the master position is required. Usually, this angular trimming is executed via an x offset. This x offset virtually trims the master value and thus aligns the following axis correctly to the process.

- The following illustration (without x offset) shows the "arrows" print image shifted by Δx_{max} compared to the wanted position.



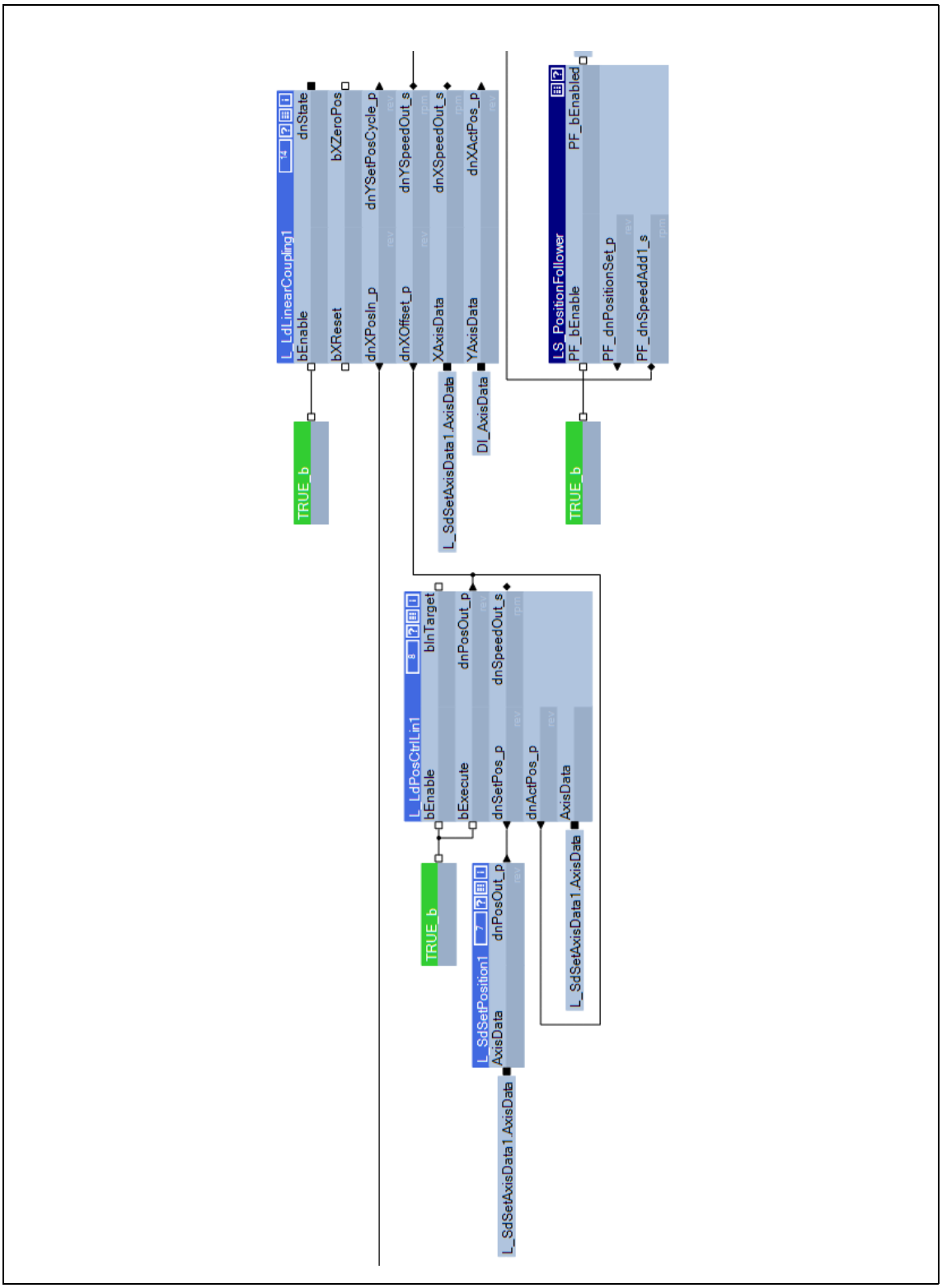
[5-51] Printing roller without angular trimming via x offset

- When x offset is used, the "arrows" print image is on the correct position:



[5-52] Printing roller with angular trimming via x offset

The FB **LdPosCtrlLin** serves to constantly adjust the x offset position. For this purpose the FB must be interconnected as follows in connection with the L_LdLinearCoupling synchronous function:



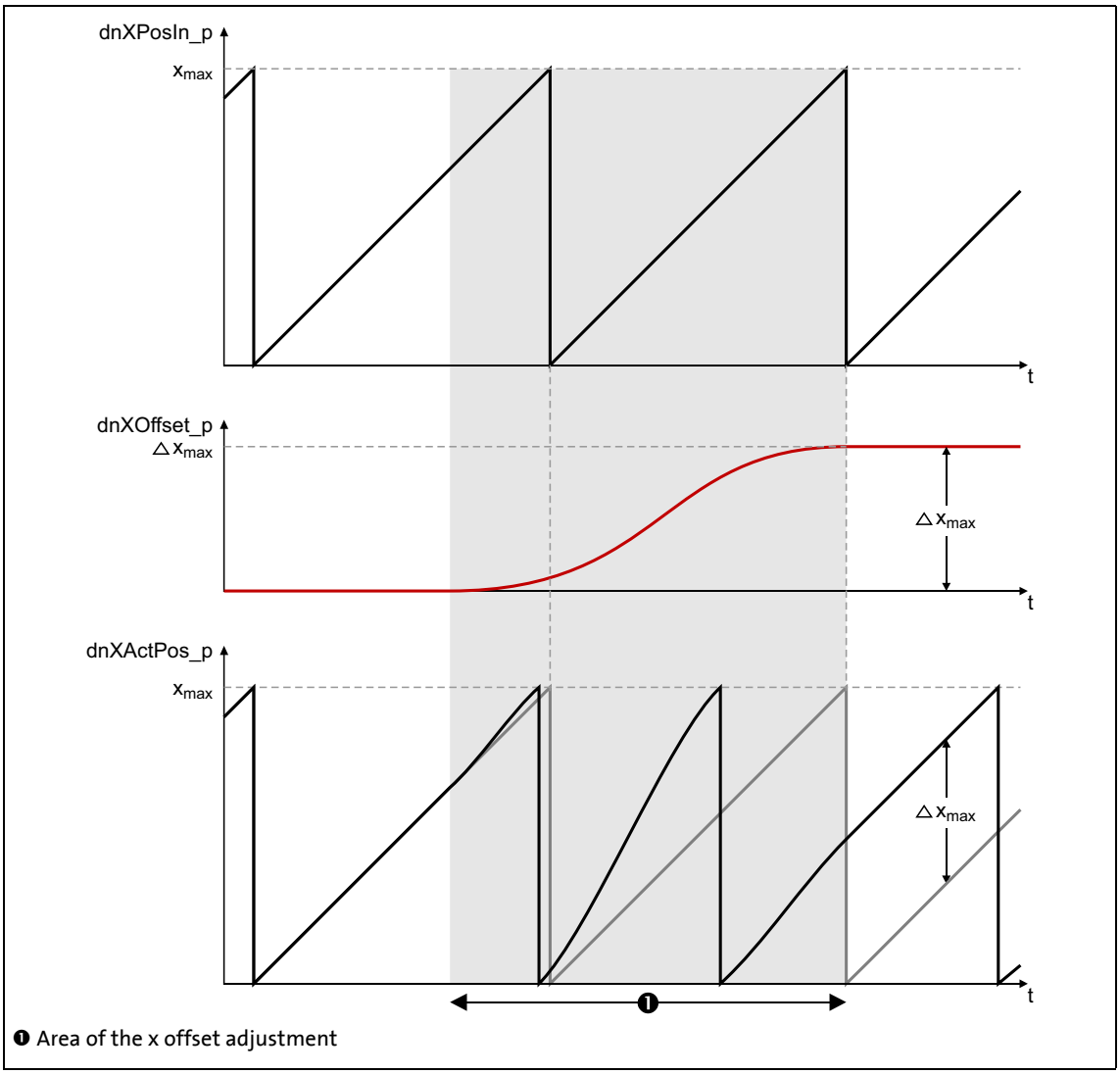
[5-53] Model connection

Function block	Function
L_SdSetPosition1	Selection of an x offset value • The FB L_SdSetPosition converts the position value defined in the application unit [unit] into an incremental value. • The reference measuring system is the master measuring system which is defined via the FB L_SdSetAxisData (not displayed).
L_LdPosCtrlLin1	Continuous adjustment of the x offset position • As described, the FB **L_LdPosCtrlLin** serves to constantly adjust the x offset position at the *dnPosOut_p* output. • The master measuring system serves as reference measuring system. • Important: The *dnPosOut_p* output position must be returned to the *dnActPos_p* input.
L_LdLinearCoupling1	Synchronous function • The FB L_LdLinearCoupling establishes the coupling between master and axis measuring system. • Moreover, this FB provides the option to consider an x offset in the master value.

Tip!

The cycle-related offsetting in the master measuring system (Modulo measuring system) is executed as in the FB L_LdAddOffsetCyclic.

The following signal characteristics exemplifies the x offset adjustment from zero to an offset value x_{max}:

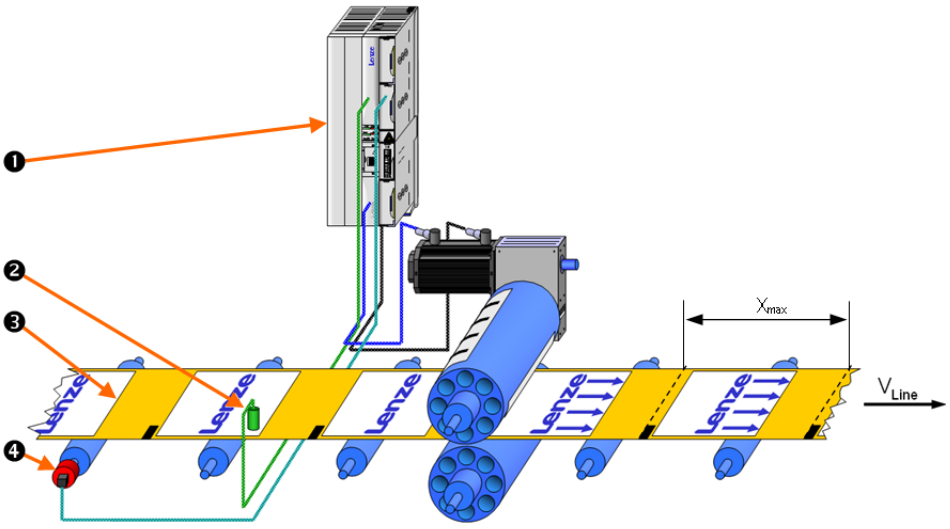


❶ Area of the x offset adjustment

[5-54] Signal characteristics - x offset adjustment

5.94.2.2 Time-based compensation of mark correction

In many production processes, a print register is already defined on the web, e.g. via print marks. These mark signals can be detected with high precision via fast digital inputs (touch probe inputs) in the controller:



❶ Slave drive
❷ Print-mark sensor
❸ Web
❹ Master value encoder
X_{max} = x-cycle of the register

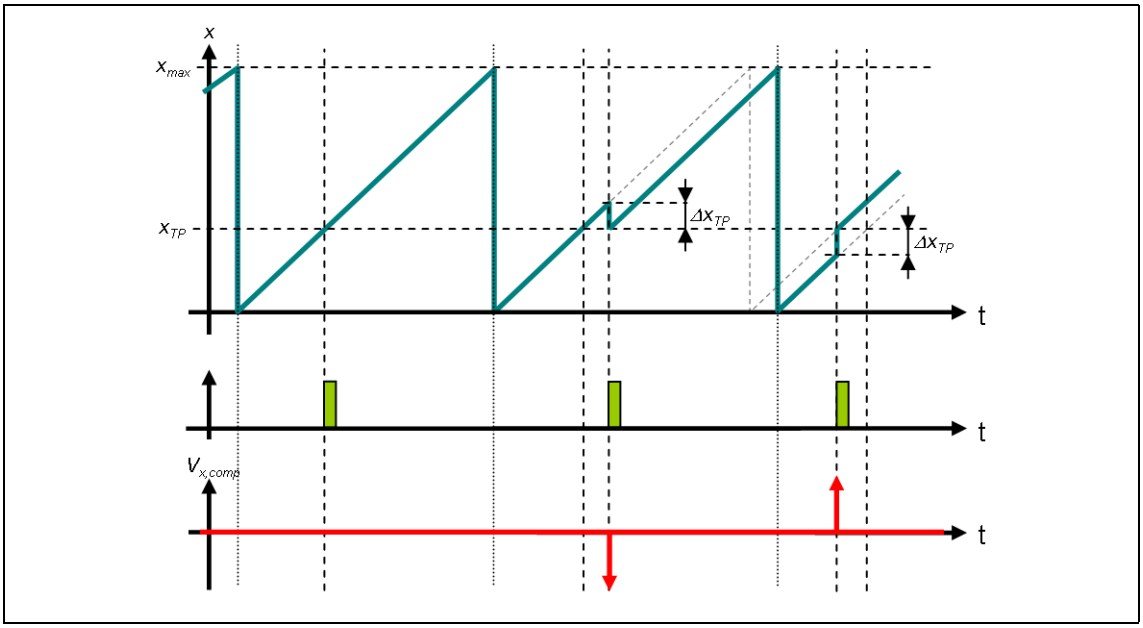
[5-55] Example : Material path with register

Since the production process is periodical, a print mark pulse appears per cycle. Ideally, the pulse always appears at a certain set position x_{TP} in the master cycle 0 ... x_{max}. If the print mark pulse is detected at another position than at its set position x_{TP}, there is a deviation Δx_{TP} between setpoint and actual position of the print mark pulse.

Options for compensating the print mark deviation:

A. Immediate compensation of Δx_{TP}

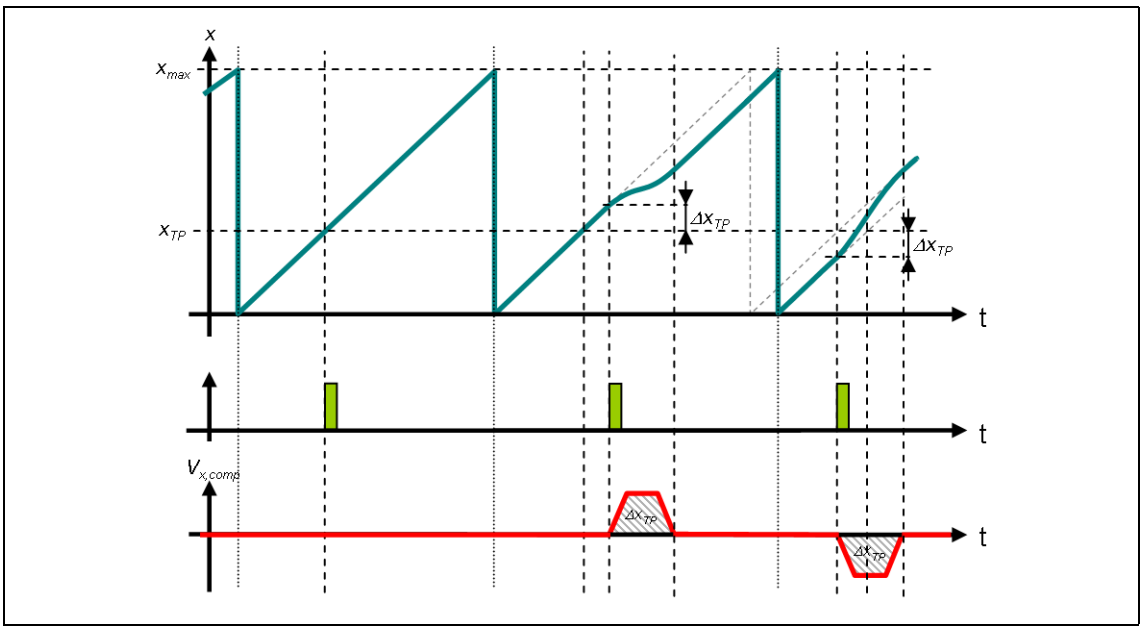
- During the compensation speed $V_{x,comp}$ delta pulses occur, since the x value is set to its set position x_{TP} when the print mark pulse occurs.



[5-56] Immediate compensation of the print mark deviation Δx_{TP}

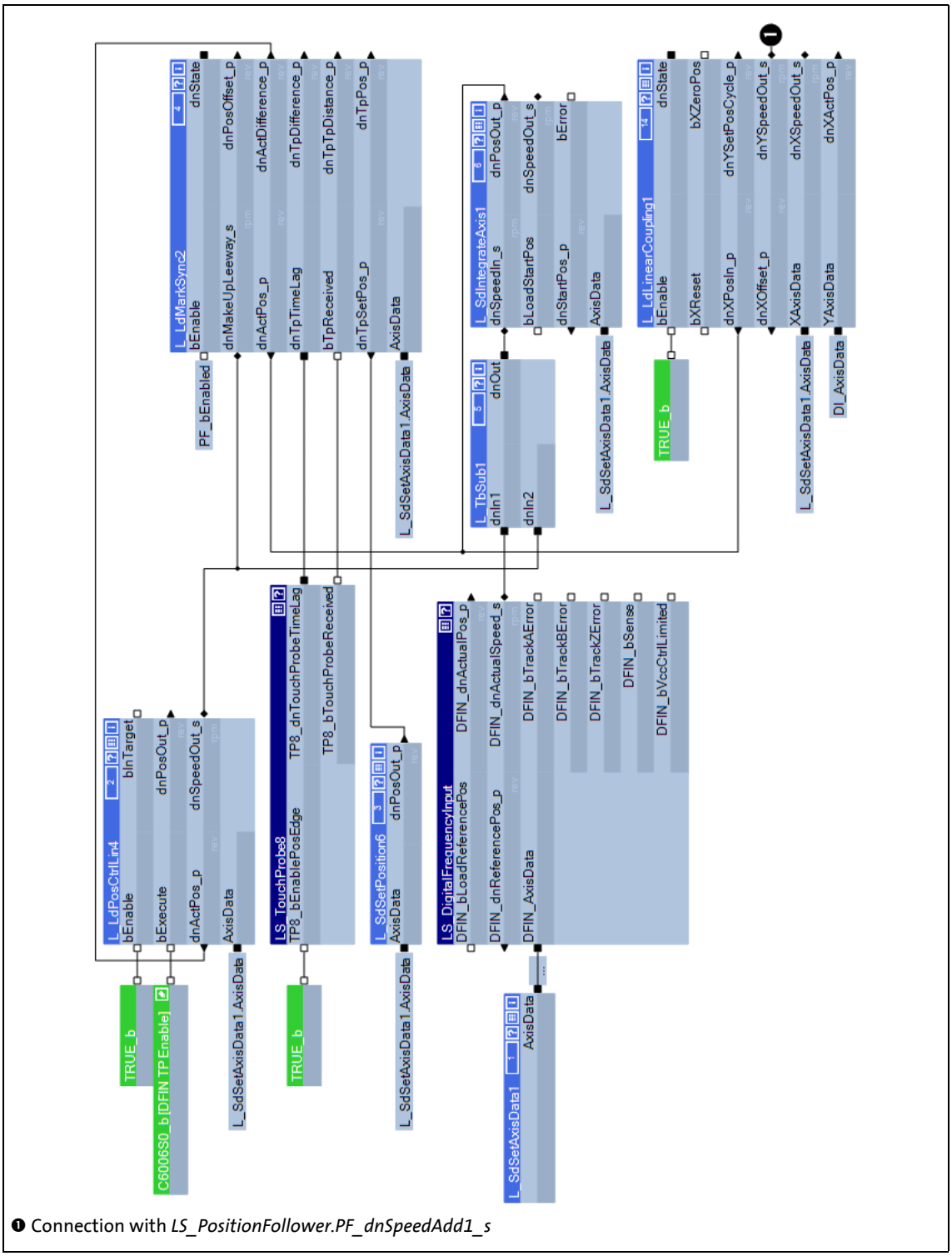
B. Compensation of Δx_{TP} via a higher-level compensation profile

- During the compensation speed $V_{x,comp}$, neither delta pulses nor jumps occur since the deviation from the set position of the print mark pulse Δx_{TP} is compensated via a profile generator:



[5-57] Compensation of the print mark deviation Δx_{TP} via a higher-level compensation profile

Interconnection example of a controlled compensation of print-mark deviation Δx_{TP} via a time-controlled positioning profile:



❶ Connection with *LS_PositionFollower.PF_dnSpeedAdd1_s*

[5-58] Model connection

Function block	Function
L_LdPosCtrlLin4	Time-controlled profile generation • Using the detected deviation (print-mark correction path), the L_LdPosCtrlLin FB generates a time-controlled speed setpoint. • The speed setpoint is provided at the *L_LdPosCtrlLin4.dnSpeedOut_s* output.
LS_TouchProbe8	Detection of the print-mark signal
L_SdSetPosition6	Selection of the set position of the print-mark sensor • The FB L_SdSetPosition converts the set position defined in the application unit [unit] into an incremental value. • The reference measuring system is the master measuring system which is defined via the FB L_SdSetAxisData.
L_LdMarkSync2	Detection of the correction distance Δx_{TP} • The FB L_LdMarkSync detects the correction distance Δx_{TP} based on the current x position and the set position of the print mark signal x_{TP} if a print mark pulse is detected via the SB **LS_TouchProbe8**. • The correction distance Δx_{TP} is provided at the *L_LdMarkSync2.dnActDifference_p* output.
L_SdSetAxisData1	Definition of the master measuring system • Using the L_SdSetAxisData FB, machine parameters of a higher-level drive can be displayed.
LS_DigitalFrequencyInput	Detection of the digital frequency and output as speed signal
L_TbSub1	Injection of the correction speed • The L_TbSub FB performs a subtraction without limitation.
L_SdIntegrateAxis1	Integration of the resulting speed signal to a master position (x position) • The L_SdIntegrateAxis FB integrates a speed into a position.
L_LdLinearCoupling1	Synchronous function • The FB L_LdLinearCoupling establishes the coupling between master and axis measuring system.

Operating mode

The core function of the interconnection is contained in the FB L_LdMarkSync2: On the basis of the current x position (integrator output *L_SdIntegrateAxis1.dnPosOut_p*) and the set position of the print-mark signal x_{TP} (output *L_SdSetPosition6.dnPosOut_p*) the FB detects the correction distance Δx_{TP} when a print mark pulse is detected.

In the example, the print-mark sensor is connected to the digital input DI8, touch probe is therefore detected via the **LS_TouchProbe8** system block.

The correction distance Δx_{TP} is led from the signal output *L_LdMarkSync2.dnActDifference_p* to the profile generator L_LdPosCtrlLin4 which generates a correction profile $V_{x,comp}$.

The correction speed $V_{x,comp}$ is superimposed on the real master speed (system variable *DFIN_dnActualSpeed_s*) via the FB L_TbSub1 and then integrated in the integrator block L_SdIntegrateAxis1 modulo to a master position 0 ... x_{max}.



Note!

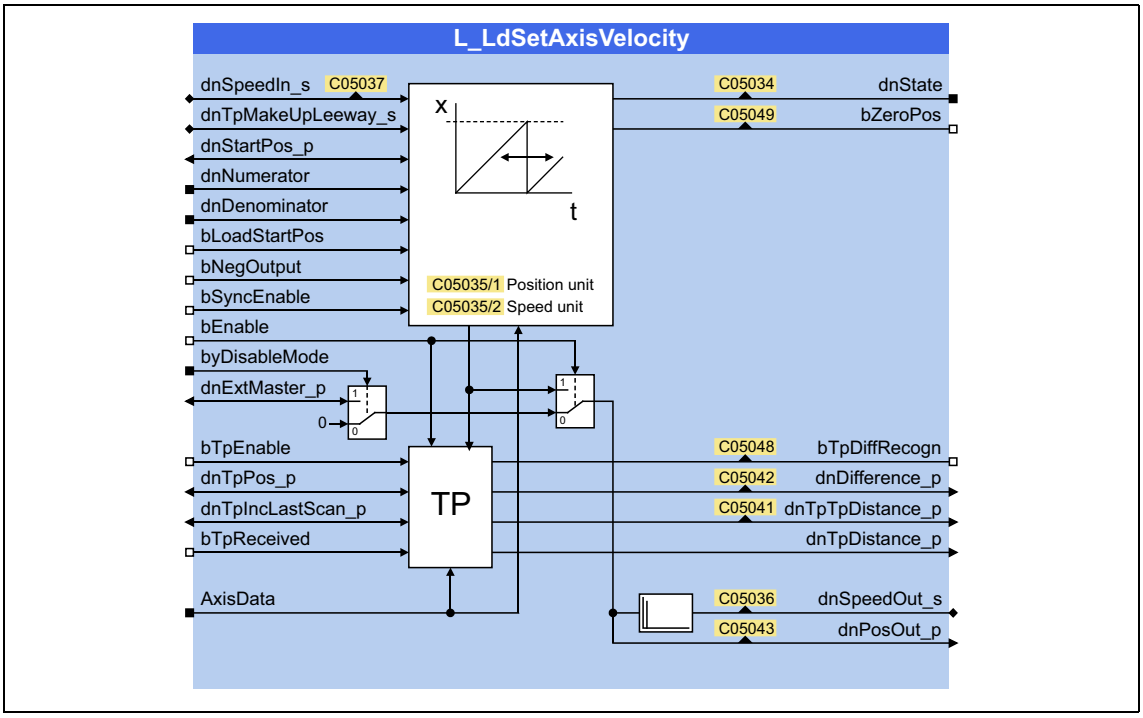
Touch probe corrections cannot be applied to absolute master positions, as for instance if selected via MotionBus.

A touch probe correction on the master value side requires a master speed as master value selection (e.g. via digital frequency coupling).

5.95 L_LdSetAxisVelocity - master value processing

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to extend/compress the X axis and execute a synchronisation via touch probe. The master speed is integrated to the position within the cycle.



Inputs

Identifier/data type	Information/possible settings
dnSpeedIn_s DINT	Selection of the master speed, e.g. from an encoder. • At this input, the FB expects a speed in [rpm]. The machine parameters at the input *AxisData* are used as speed reference.
dnTpMakeUpLeeway_s DINT	Selection (addition) of the TP correction speed to the master value • At this input, the FB expects a speed in [rpm]. The machine parameters at the input *AxisData* are used as speed reference.
dnStartPos_p DINT	Starting position of the master value integrator in [inc] • The selected master position is accepted with a FALSE↗TRUE edge at *bLoadStartPos*. • The value must be inside the range of the scaled master value cycle (register).
dnNumerator DINT	Factor (numerator) for extending and compressing the master axis • A change during operation is displayed as a stretch factor (electronic gearbox).
dnDenominator DINT	Factor (denominator) for extending and compressing the master axis • A change during operation is displayed as a stretch factor (electronic gearbox).
bLoadStartPos BOOL	Load starting position FALSE↗TRUE: Load starting position *dnStartPos_p* in the master value integrator.

Identifier/data type	Information/possible settings	
bNegOutput BOOL	Invert master speed • Direct sign reversal of the master speed. • The inversion of the master value gets active before the master value scaling and the stretch factor.	
	FALSE	CW rotation, the master value is not reversed.
	TRUE	CCW rotation, the master value is reversed.
bSyncEnable BOOL	Activate synchronised stretching/compression	
	FALSE	Changes in the stretch factor are effective immediately.
	TRUE	Changes in the stretch factor are only accepted during zero crossing.
bEnable BOOL	Activate FB	
	FALSE	Function block is deactivated, for behaviour see *byDisableMode*.
	TRUE	Function block activated.
	TRUE↘FALSE	Master shaft *dnPosOut_p* travels via the ramp for permanent operation with master speed to the external master position *dnExtMaster_p*.
byDisableMode BYTE	Behaviour in case of deactivated function block (*bEnable* = FALSE)	
	0	Speed output is not active, position remains.
	1	Position at *dnExtMaster_p* is accepted, speed output is active.
dnExtMaster_p DINT	Selection of an external master position in [inc], e.g. through an absolute value encoder. • Master value scaling is not considered. • The master position must be inside the cycle.	
bTpEnable BOOL	Activate touch probe evaluation	
	TRUE	Touch probe evaluation activated.
dnTpPos_p DINT	Assigned position of the touch probe sensor in the master value cycle in [inc] • Value is internally limited to 0 ... 2 x cycle.	
dnTpIncLastScan_p DINT	Number of increments counted since touch probe. • Connect this input with the output *dnTPCaptureValue1* of the touch probe system block used.	
bTpReceived BOOL	Control signal for touch probe correction • Connect this input with the output *bTouchProbeReceived* of the touch probe system block used.	
	FALSE↗TRUE	Touch probe detected.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status	
	bits4	Position change per cycle too big
	bits10	Zero crossing active
	bits14	Internal limitation for calculation
	Bit15	Error
	bit16	Starting position is beyond the cycle
	bits17	Error in the master signal (position or speed)
	bits18	No cycle available
	Bit 19	Maximum speed = "0"
	bits20	Position of the touch probe sensor is beyond the cycle
	bits21	Input error (e.g. impermissible position jump or inverse direction of motion)
	bits23	No valid axis data structure
bZeroPos BOOL	Status signal "Zero position/zero crossing"	
	FALSE	Master value is at any position in the master value cycle (register).
	TRUE	**Permanently:** Master value is at the zero position of the master value integrator. **For one cycle:** Zero crossing of the master value integrator.
bTpDiffRecogn BOOL	Status signal "Touch probe correction is active"	
	FALSE	Touch probe correction is not active or already executed in the current cycle.
	TRUE	Touch probe correction is active or not yet completed via the ramp function.
dnDifference_p DINT	Difference between *dnTpPos_p* and the position measured after detecting a touch probe in [inc].	
dnTpTpDistance_p DINT	Distance between the last two touch probes in [inc] • For instance to measure the real master value register.	
dnTpDistance_p DINT	Distance between the detected touch probe and the zero point of the integrator in [inc]	
dnSpeedOut_s DINT	Master speed output in [rpm] • C05036 indicates the master speed in the real unit of the machine.	
dnPosOut_p DINT	Master position output (position value of the master value integrator) in [inc] • C05043 indicates the master position in the real unit of the machine.	

Parameter

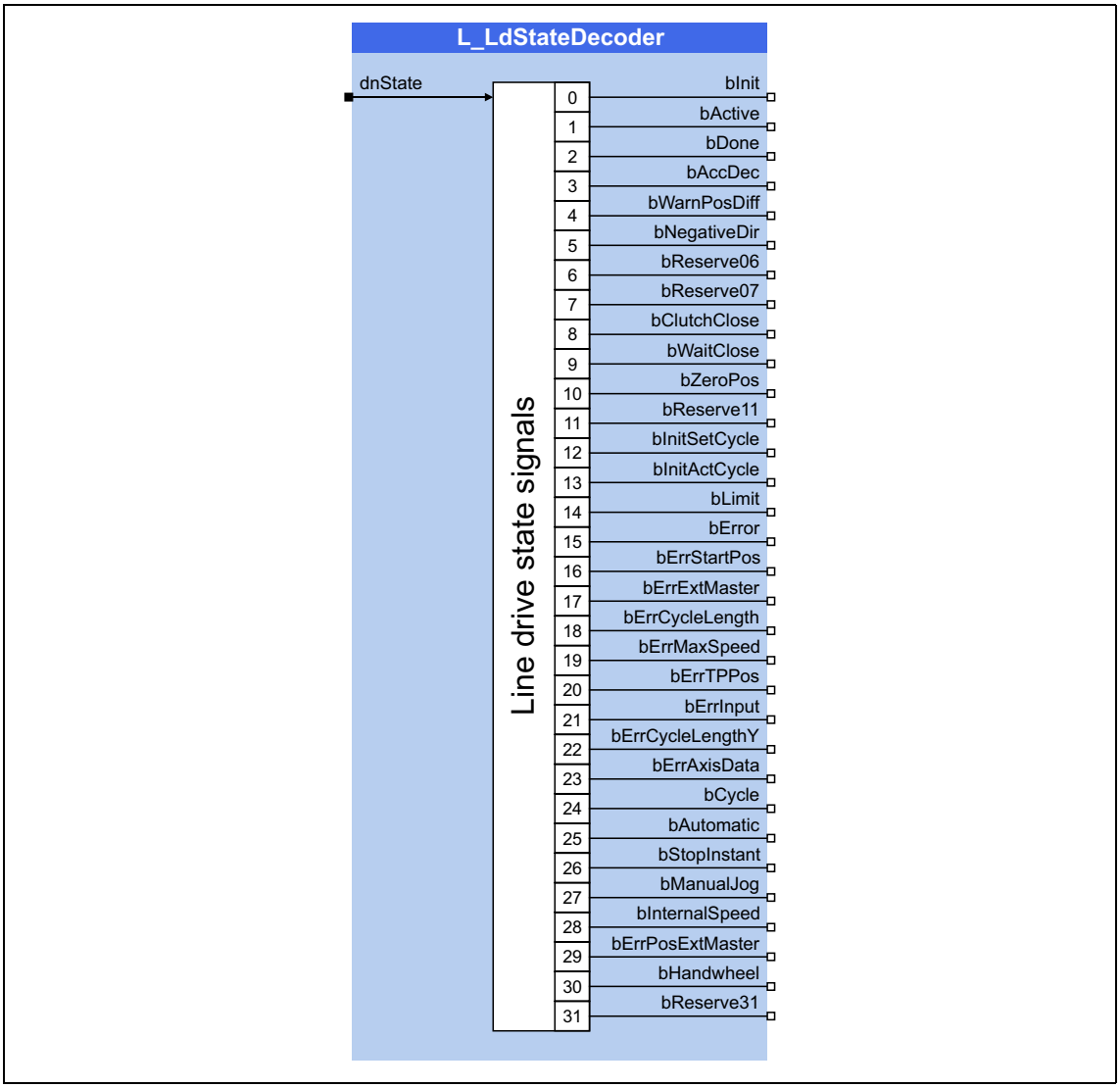
Parameter	Possible settings			Information
C05034	-214748.3647		214748.3647	Status • Display of the bit-coded output signal *dnState*.
C05035/1 As of library V02.02.xx.xx	String of digits			Position unit • Read only
C05035/2 As of library V02.02.xx.xx	String of digits			Speed unit • Read only
C05036	-214748.3647	Unit/t	214748.3647	Speed at the output • Read only • Calculated from the speed signal *dnSpeedOut_s*.
C05037 As of library V02.02.xx.xx	-214748.3647	Unit/t	214748.3647	Speed at the input • Read only • Calculated from the speed signal *dnSpeedIn_s*.
C05041	-214748.3647	Unit	214748.3647	Measured cycle • Display of the output signal *dnTpTpDistance_p* in the real unit of the machine.
C05042	-214748.3647	Unit	214748.3647	Measured TP deviation • Display of the output signal *dnDifference_p* in the real unit of the machine.
C05043	-214748.3647	Unit	214748.3647	Position at the output • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C05048				TP: Deviation detected • Display of the output signal *bTpDiffRecogn*.
	0	No deviation detected		
	1	Deviation detected		
C05049				Cycle start • Display of the output signal *bZeroPos*.
	0	No cycle start		
	1	Cycle start		

5.96 L_LdStateDecoder - LineDrive status signals

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB decodes the status output signal *dnState* of an FB from the function library "LenzeLineDrive.lib" into individual boolean status signals for further use in the FB interconnection.

- The functionality corresponds to a converter "DINT➞BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnState DINT	Input for the bit-coded status signal of an FB from the function library "LenzeLineDrive.lib". • Connect this input with the output of the same name of the corresponding FB.

Outputs

The boolean outputs have the following meaning when the value is TRUE:

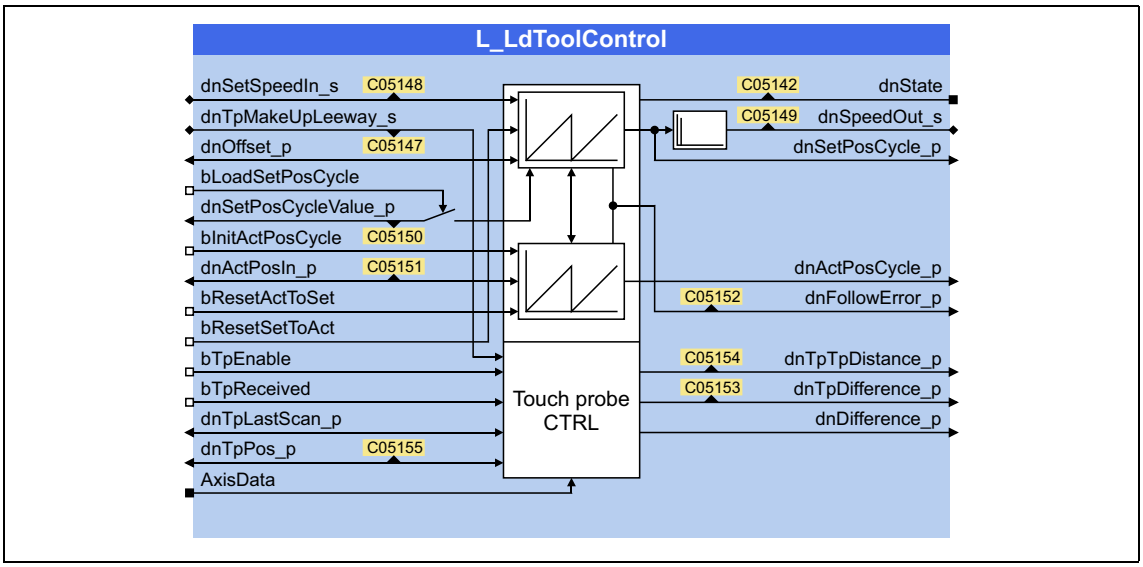
Bit*	Identifier	Meaning in case of TRUE
Status messages:		
0	bInit	Function is being initialised.
1	bActive	Function is active.
2	bDone	Function is completed.
3	bAccDec	Acceleration/deceleration phase is active.
4	bWarnPosDiff	Position change per cycle too big.
5	bNegativeDir	Counter-clockwise rotation
6	bReserve06	-
7	bReserve07	-
8	bClutchClose	Brake/clutch closed.
9	bWaitClose	Waiting for engagement process.
10	bZeroPos	Zero crossing is active.
11	bReserve11	-
12	bInitSetCycle	Initialisation of actual cycle.
13	bInitActCycle	Initialisation of setpoint cycle.
14	bLimit	Internal limitation for calculation.
24	bCycle	Single cycle
25	bAutomatic	Automatic mode
26	bStopInstant	Stopping is active.
27	bManualJog	Manual jog
28	bInternalSpeed	Internal setpoint speed is active.
30	bHandwheel	Electronic handwheel is active.
31	bReserve31	-
Error messages:		
15	bError	Error
16	bErrStartPos	Starting position is beyond the cycle.
17	bErrExtMaster	Error in the master signal (position or speed).
18	bErrCycleLength	No cycle available.
19	bErrMaxSpeed	Maximum speed = "0".
20	bErrTPPos	Position of the touch probe sensor is beyond the cycle.
21	bErrInput	Input error (e.g. impermissible position jump or inverse direction of motion).
22	bErrCycleLengthY	No cycle available for Y axis.
23	bErrAxisData	No valid axis data structure.
29	bErrPosExtMaster	The FB L_LdVirtualMasterV cannot approach the external target position because it is varying. Only fixed positions may be approached.

* Relating to the input signal *dnState*.

5.97 L_LdToolControl - setpoint conditioning

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to execute a setpoint conditioning.



Inputs

Identifier/data type	Information/possible settings
dnSpeedIn_s DINT	Selection of the speed setpoint for the machine axis (tool) • At this input, the FB expects a speed in [rpm]. The machine parameters at the input *AxisData* are used as speed reference. • C05148 indicates the setpoint speed in the real unit of the machine.
dnTpMakeUpLeeway_s DINT	Speed for touch probe correction • At this input, the FB expects a speed in [rpm]. The machine parameters at the input *AxisData* are used as speed reference. • C05147 indicates the correction speed in the real unit of the machine.
dnOffset_p DINT	Relative offset in [inc] • Is added to the speed setpoint *dnSpeedIn_s*.
bLoadSetPosCycle BOOL	Load selection for the master value adjustment TRUE: Load value at *dnSetPosCycleValue_p* into the SET integrator.
dnSetPosCycleValue_p DINT	Selection for the master value adjustment in [inc] • Load value for the SET integrator (setpoint position for the machine axis).
bInitActPosCycle BOOL	Initialise ACT integrator with the actual position TRUE: Load the actual position *dnActPosIn_p* into the ACT integrator.
dnActPosIn_p DINT	Actual position (from SB **LS_Feedback**) in [inc]
bResetActToSet BOOL	Delete following error TRUE: Transmits the value of the SET integrator to the ACT integrator. • The output value *dnSetPosCycle_p* is set to the output value *dnActPosCycle_p*.

Identifier/data type	Information/possible settings	
bResetSetToAct BOOL	Delete following error	
	TRUE	Transmits the value of the ACT integrator to the SET integrator. • The output value *dnActPosCycle_p* is set to the output value *dnSetPosCycle_p*.
bTpEnable BOOL	Activate touch probe evaluation	
	TRUE	Touch probe evaluation activated.
bTpReceived BOOL	Control signal for touch probe correction • Connect this input with the output *bTouchProbeReceived* of the touch probe system block used.	
	FALSE↗TRUE	Touch probe detected.
dnTpLastScan_p DINT	Number of increments counted since touch probe. • Connect this input with the output *dnTPCaptureValue1* of the touch probe system block used.	
dnTpPos_p DINT	Assigned position of the touch probe sensor in the setpoint cycle in [inc] • Value is internally limited to 0 ... 2 x cycle. • C05155 indicates the position in the real unit of the machine.	
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits4	Position change per cycle too big
	bits10	Zero crossing active
	bits12	Initialisation of actual clock
	bits13	Initialisation of setpoint clock
	Bit15	Error
	bits18	No cycle available
	Bit 19	Maximum speed = "0"
	bits20	Position of the touch probe sensor is beyond the cycle
	bits21	Input error (e.g. impermissible position jump or inverse direction of motion)
	bits23	No valid axis data structure
dnSpeedOut_s DINT	Speed setpoint for the position follower (SB **LS_PositionFollower**) for position control	
dnSetPosCycle_p DINT	Setpoint position in the cycle (with clock reference) in [inc] • Referring to zero point. • C05150 indicates the setpoint position in the real unit of the machine.	
dnActCycle_p DINT	Actual position in the cycle (with clock reference) in [inc] • Referring to zero point. • C05151 indicates the actual position in the real unit of the machine.	
dnFollowError_p DINT	Current following error in [inc] • C05152 indicates the following error in the real unit of the machine.	
dnTpTpDistance_p DINT	Distance between the last two touch probes in [inc] • For instance to measure the real actual register. • C05154 indicates the distance in the real unit of the machine.	

Identifier/data type	Value/meaning
dnTpDifference_p DINT	Difference between *dnTpPos_p* and the position measured after detecting a touch probe in [inc].
dnDifference_p DINT	Current difference between *dnTpPos_p* and the position measured after detecting a touch probe in [inc]. • C05153 indicates the deviation in the real unit of the machine.

Parameter

Parameter	Possible settings			Information
C05142	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.
C05143/1 As of library V02.02.xx.xx	String of digits			Position unit • Read only
C05143/2 As of library V02.02.xx.xx	String of digits			Speed unit • Read only
C05144 As of library V02.02.xx.xx	-214748.3647	Unit/t	214748.3647	Reference speed • Read only • For input orientation for speed entries.
C05147	-2147483647	Unit/t	2147483647	Correction speed at the input • Read only • Calculated from the speed signal *dnTpMakeUpLeeway_s*.
C05148	-2147483647	Unit/t	2147483647	Speed at the input • Read only • Calculated from the speed signal *dnSetSpeedIn_s*.
C05149	-2147483647	Unit/t	2147483647	Speed at the output • Read only • Calculated from the speed signal *dnSpeedOut_s*.
C05150	-214748.3647	Unit	214748.3647	Set position at the output • Display of the output signal *dnSetPosCycle_p* in the real unit of the machine.
C05151	-214748.3647	Unit	214748.3647	Actual position at the output • Display of the output signal *dnActPosCycle_p* in the real unit of the machine.
C05152	-214748.3647	Unit	214748.3647	Following error • Display of the output signal *dnFollowError_p* in the real unit of the machine.
C05153	-214748.3647	Unit	214748.3647	Measured TP deviation • Display of the output signal *dnDifference_p* in the real unit of the machine.
C05154	-214748.3647	Unit	214748.3647	Measured cycle • Display of the output signal *dnTpTpDistance_p* in the real unit of the machine.
C05155	-214748.3647	Unit	214748.3647	TP set position at the input • Display of the *dnTpPos_p* input signal in the real unit of the machine.

5.98 L_LdSyncOperation - master value connection

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

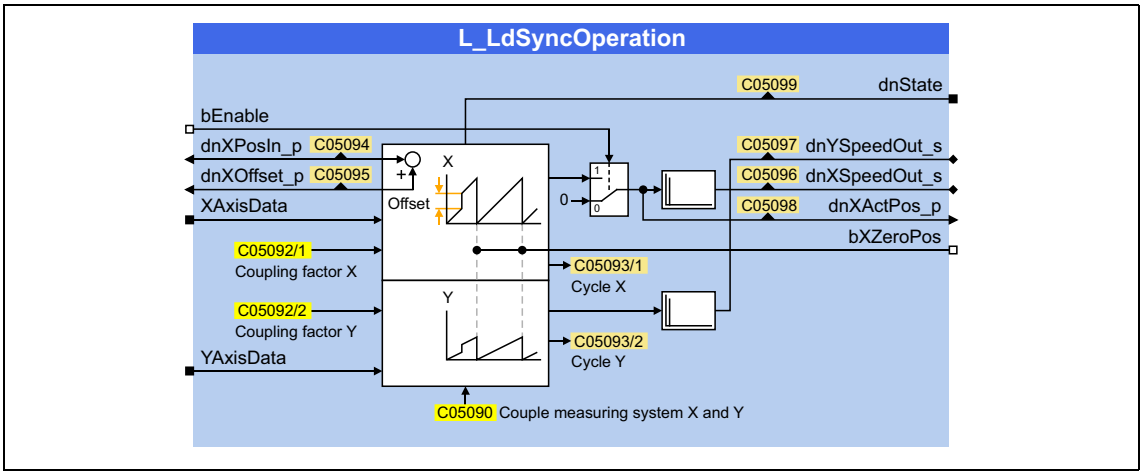
This FB implements synchronism between a master drive and a slave drive:

- Drift-free synchronism between master drive and slave drive (modulo measuring systems) with different types of couplings.
- Additional in-cycle compensation of angular trimming in the master measuring system (x offset).

Tip!

There are two other FB for synchronous operations:

- The L_LdConvAxisV FB generates the speed setpoint for the following axis from the master speed instead of the master position, as this FB does.
- Like this FB, the L_LdLinearCoupling FB accepts an x position as the master value, but generates a setpoint position for the following axis in addition to the setpoint speed.



Inputs

Identifier/data type	Information/possible settings	
bEnable BOOL	Activate function	
	FALSE	Function has been deactivated. • The *dnXSpeedOut_s* and *dnXActPos_p* outputs continue to follow the *dnXPosIn_p* and *dnXOffset_p* input positions. • A value of "0" is output at the *dnYSpeedOut_s* output. **Caution:** Thus, a position step change at the master value end will not be displayed at *dnYSpeedOut_s*!
	TRUE	Function is activated. • According to the coupling mode parameterised in C05090, the output signals follow the *input positions, dnXPosIn_p* and *dnXOffset_p*.

Identifier/data type	Information/possible settings
dnXPosIn_p DINT	Selection of the master position in [inc] • The master position must not exceed the permissible modulo value range of the master measuring system (the *XAxisData* input). • If the position changes considerably at this input, it may become impossible to display the values at the *dnXSpeedOut_s* and/or *dnYSpeedOut_s* outputs. In this case, the position reference between master drive and slave drive would get lost! • C05094 indicates the master position in the real unit of the machine.
dnXOffset_p DINT	Selection of the master value adjustment (x offset) in [increments] • It is permissible for the trimming value to be outside of the modulo value range of the master measuring system, but it must not be outside of the DINT value range minus the master cycle (DINT value range = $\pm 2^{31}$-1 [increments] ≡ ±32768 [rev]). • The resulting master value (the *dnXActPos_p* output) will be included in the modulo cycle of the master measuring system. • C05094 displays the effective x offset in the real unit of the master measuring system. • Sudden changes in the offset position cause sudden responses of the drive motion! Provide for smooth offset adjustment (e.g. via the L_LdPosCtrlLin FB).
XAxisData	Selection of the measuring system of the master drive for scaling the master value The following sources can be used for this input: • Any master measuring system can be defined via the L_SdSetAxisData FB. In this case, its *AxisData* output must be connected to this input. • Some of the cam technology function blocks (e.g. the L_CamProfiler FB) output the measuring systems defined via »Cam Designer«. In this case, the corresponding measuring system output must be connected to the *XAxisData* input. **Note:** In most cases, master measuring systems are of the "modulo" type. However, both limited and unlimited measuring systems are also possible. If the FB is operated in the "Coupling via cycles" operating mode (C05090 = 1), it is mandatory for the measuring system to be a modulo measuring system.
YAxisData	Selection of the measuring system of the slave drive for scaling the setpoint • This input typically receives drive measuring system data (the *DI_AxisData* output of the **LS_DriveInterface** SB). **Note:** Basically, all measuring system types (limited, unlimited, modulo) are permissible. If the FB is operated in the "Coupling via cycles" operating mode (C05090 = 1), it is mandatory for the measuring system to be a modulo measuring system.

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits10	Zero crossing active (corresponds to the *bXZeroPos* output signal)
	bits14	Internal limitation for calculation: The *dnYSpeedOut_s* and/or *dnYSpeedOut_s* speed outputs are not able to follow the change in the resulting *dnActXPos_p* master position. The reference to the master cycle gets lost.
	Bit15	Group error (see bit 16 ... 31)
	bit16	The master position at the *dnXPosIn_p* input is outside of the permissible modulo value range of the master measuring system (see *XAxisData* input).
	bits18	The master measuring system and/or the slave measuring system (the *XAxisData* and *YAxisData* inputs) are not modulo measuring systems, or the modulo cycle in one of the measuring systems is zero.
	bits21	Incorrect input value: • Error when selecting the offset position: The *dnXOffset_p* input exceeds the permissible DINT value range minus the master cycle. (DINT value range = $\pm 2^{31}$-1 [increments] ≡ ±32768 [rev]) • Gearbox denominator = "0"
	bits23	Error in the measuring system: • The pointers at the *XAxisData* and/or *YAxisData* inputs do not refer to a valid axis data structure. • If coupling is performed via cycles (C05090 = 1), modulo measuring systems must be specified at the *XAxisData* and/or *YAxisData* inputs.
dnYSpeedOut_s DINT	Output of the setpoint speed (y speed) as a speed signal • Scaling: 2^{26} ≡ 15000 [rpm] • The signal can be used for direct coupling to the **LS_PositionFollower** system block (the *PF_dnSpeedAdd1_s* input). • Shifts in the resulting master position (e.g. when switching-off the FB via *bEnable* = TRUE↘FALSE) will not be displayed via this signal. The position reference to the *dnXActPos_p* master cycle may get lost! • If the FB is deactivated (*bEnable* = FALSE), a value of "0" is output. • Using the measuring system data at the YAxisData input, the setpoint speed is indicated in the real unit of the machine in C05097.	
dnXSpeedOut_s DINT	Output of the resulting master speed as a speed signal (The resulting change in the *dnXPosIn_p* master position and the *dnXOffset_p* master value adjustment is displayed.) • Scaling: 2^{26} ≡ 15000 [rpm] • This signal can be used for the coupling of further slave drives. • Angular trimming via the *dnXOffset_p* input will be accounted for in this signal. • The signal directly results from the differentiation of the resulting *dnXActPos_p* master position. Considerable position step changes at *dnXActPos_p* lead to a limitation of this output. In this case, the signal does not display the position step change at *dnXActPos_p*. • If the FB is deactivated (*bEnable* = FALSE), the resulting master speed is output as before. • Using the measuring system data at the *XAxisData* input, the resulting master speed is displayed in the real unit of the master measuring system in C05096.	
dnXActPos_p DINT	Output of the resulting master position in [increments] • If *bEnable* = TRUE, the *dnXPosIn_p* master position offset against the *dnXOffset_p* offset value will be output. The output signal is limited to the modulo value range of the master measuring system (*XAxisData*). • If the FB is deactivated (*bEnable* = FALSE), the resulting master position is output as before. • Using the measuring system data at the *XAxisData* input, the resulting master position is displayed in the real unit of the master measuring system in C05098.	

Identifier/data type	Value/meaning	
bXZeroPos BOOL	Status signal: "Cycle zero crossing in the modulo cycle of the master measuring system"	
	FALSE	Cycle zero crossing did not occur: The resulting *dnXActPos_p* master position is non-zero, the resulting master value is in the x modulo cycle.
	TRUE	Cycle zero crossing occurred: • The *bXZeroPos* output is statically set to TRUE if the resulting *dnXActPos_p* master position is zero. • The *bXZeroPos* output is set to TRUE for one task cycle if the resulting *dnXActPos_p* master position performs a step change to the following x modulo cycle (*dnXSpeedOut_s* > 0: $x_{max} \rightarrow 0$) or to the previous x modulo cycle (*dnXSpeedOut_s* < 0: $0 \rightarrow x_{max}$).

Parameter

Parameter	Possible settings			Information
C05090				Coupling measuring systems X and Y
	0	Coupling via feed constants		One of the units of the master measuring system is mapped to one of the units of the slave measuring system (Lenze setting).
	1	Coupling via cycles		One of the cycles of the master measuring system is mapped to one of the cycles of the slave measuring system.
	2	Free coupling		Synchronism depends on the ratio between the gearbox factors of the master drive/slave drive and on the coupling factors, C05092/1 and C05092/2.
C05091/1	String of digits			User-defined unit X • Display of the unit of the master measuring system.
C05091/2	String of digits			User-defined unit Y • Display of the unit of the slave measuring system.
C05092/1	0.0000		214748.3647	Coupling factor x (master measuring system) • The value entered here is included in the resulting synchronism factor as the denominator term. • Initialisation: 0.0000
C05092/2	0.0000		214748.3647	Coupling factor y (slave measuring system) • The value entered here is included in the resulting synchronism factor as the numerator term. • Initialisation: 0.0000
C05093/1	0	Unit	2147483647	Cycle X • Display of the cycle of the master measuring system
C05093/2	0	Unit	2147483647	Cycle Y • Display of the cycle of the slave measuring system

Parameter	Possible settings			Information
C05094	-214748.3647	Unit	214748.3647	Position x at the input (master position) • Display of the *dnXPosIn_p* input signal in the real unit of the machine.
C05095	-214748.3647	Unit	214748.3647	Trimming X • Display of the *dnXOffset_p* input signal in the real unit of the machine.
C05096	-214748.3647	Unit/t	214748.3647	Speed X at the output • Display of the resulting master speed • Calculated from the speed signal *dnXSpeedOut_s*.
C05097	-214748.3647	Unit/t	214748.3647	Speed Y at the output • Display of the resulting setpoint speed • Calculated from the speed signal *dnYSpeedOut_s*.
C05098	-214748.3647	Unit	214748.3647	Position X at the output (resulting master position) • Display of the output signal *dnXActPos_p* in the real unit of the machine.
C05099	-214748.3647		214748.3647	Status • Display of the bit-coded output signal *dnState*.

5.98.1 Operating mode 0: Coupling via feed constants

In this operating mode, the FB couples the axes of the master drive and the slave drive via the set feed constants. The cycles of the master measuring system and the slave measuring system do not have any effect.

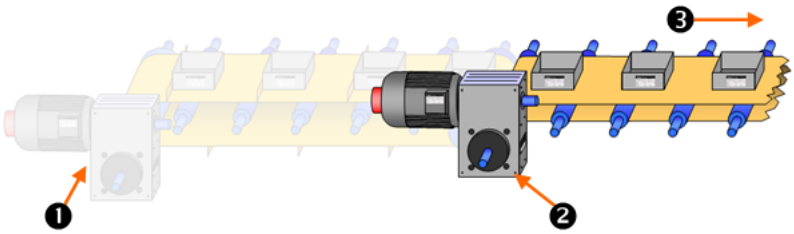
- Via C05092/1 and C05092/2 the ratio of the corresponding units is set, i.e. how many units of the master axis (x axis) correspond to how many units of the slave axis (y axis).
- The speed ratio between the slave drive and the master drive is calculated according to following formula:

$$\frac{\text{dnYSpeedOut\_s}}{\text{dnXSpeedOut\_s}} = \frac{\text{Gearbox constant Y}}{\text{Gearbox constant X}} \cdot \frac{\text{Feed constant X}}{\text{Feed constant Y}} \cdot \frac{\text{Coupling factor Y (C05092/2)}}{\text{Coupling factor X (C05092/1)}}$$

[5-59] Formula for operating mode 0

- The conversion according to the formula above is remainder considered and hence drift-free (consideration of remainders when performing divisions).

Application example: Synchronous conveying belts



❶ Master drive (conveying belt 1)
❷ Slave drive (conveying belt 2)
❸ Traversing speed

Master drive (conveying belt 1)		Slave drive (conveying belt 2)	
Gearbox constant X	1120 : 207	Gearbox constant Y	5 : 2
Feed constant X	200.0000 [mm]	Feed constant Y	90.0000 [mm]

According to the formula above, the speed ratio between the master drive and the slave drive is calculated as follows:

$$\frac{\text{dnYSpeedOut\_s}}{\text{dnXSpeedOut\_s}} = \frac{\frac{5}{2}}{\frac{1120}{207}} \cdot \frac{200.0000\ [\text{mm}]}{90.0000\ [\text{mm}]} \cdot \frac{1}{1} = 1.0268...$$

5.98.2 Operating mode 1: Coupling via cycles

In this operating mode, the FB couples the axes of the master drive and the slave drive via the set feed constants and the cycles. One of the cycles of the master drive is mapped on one of the cycles of the slave drive.

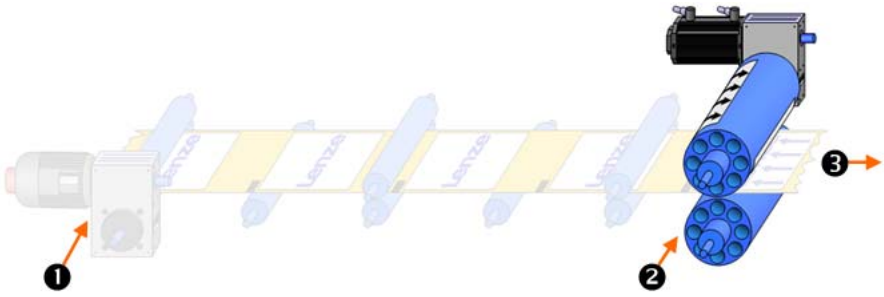
- In this type of coupling, both master measuring system and slave measuring system must be modulo measuring systems.
- The C05092/1 and C05092/2 coupling factors do not have any effect in this operating mode.
- The speed ratio between the slave drive and the master drive is calculated according to following formula:

$$\frac{\text{dnYSpeedOut\_s}}{\text{dnXSpeedOut\_s}} = \frac{\text{Gearbox constant Y}}{\text{Gearbox constant X}} \cdot \frac{\text{Feed constant X}}{\text{Feed constant Y}} \cdot \frac{\text{Cycle Y}}{\text{Cycle X}}$$

[5-60] Formula for operating mode 1

- The conversion according to the formula above is remainder considered and hence drift-free (consideration of remainders when performing divisions).

Application example: Printing roller



❶ Master drive (material feed)
❷ Slave drive (printing roller)
❸ Traversing speed

Master drive (material feed)		Slave drive (printing roller)	
Gearbox constant X	32 : 5	Gearbox constant Y	45 : 2
Feed constant X	160.0000 [mm]	Feed constant Y	360.0000 [°] (≡ 628.3185 [mm])
Cycle X	628.3185 [mm] (π * 200 [mm])	Cycle Y	360.0000 [°]

According to the formula above, the speed ratio between the master drive and the slave drive is calculated as follows:

$$\frac{\text{dnYSpeedOut\_s}}{\text{dnXSpeedOut\_s}} = \frac{\frac{45}{2}}{\frac{32}{5}} \cdot \frac{160.0000\ [\text{mm}]}{360.0000\ [°]} \cdot \frac{360.0000\ [°]}{628.3185\ [\text{mm}]} = 0.8952...$$

5.98.3 Operating mode 2: Free coupling

In this operating mode, the FB couples the axes of the master drive and the slave drive only via the gearbox factors of the master drive/slave drive and the coupling factors parameterised in C05092/1 and C05092/2.

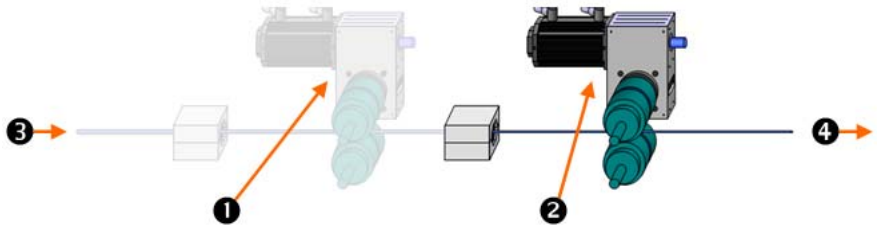
- Neither the feed constants nor the cycles of the master measuring system/slave measuring system have any effect on the synchronism ratio.
- Via C05092/1 and C05092/2 the revolution ratio is indicated, i.e. how many revolutions of the master shaft (x axis) correspond to how many revolutions of the following axis (x axis).
- The speed ratio between the slave drive and the master drive is calculated according to following formula:

$$\frac{\text{dnYSpeedOut\_s}}{\text{dnXSpeedOut\_s}} = \frac{\text{Gearbox constant Y}}{\text{Gearbox constant X}} \cdot \frac{\text{Coupling factor Y (C05092/2)}}{\text{Coupling factor X (C05092/1)}}$$

[5-61] Formula for operating mode 2

- The conversion according to the formula above is remainder considered and hence drift-free (consideration of remainders when performing divisions).

Application example: Wire drawing system with specification of the synchronism ratio K_y



❶ Master drive (wire drawing station 1)
❷ Slave drive (wire drawing station 2)
❸ Traversing speed
❹ K_y * traversing speed

Master drive (wire drawing station 1)		Slave drive (wire drawing station 2)	
Gearbox constant X	17 : 3	Gearbox constant Y	13 : 2
Feed constant X	160.0000 [mm]	Feed constant Y	160.0000 [mm]
Coupling factor X (C05092/1)	100 [%]	Coupling factor Y (C05092/2)	110 [%]

The speed ratio between the master drive and the slave drive is calculated according to the formula above, the synchronism ratio K_y being considered as the C05092/1 and C05092/1 coupling factors:

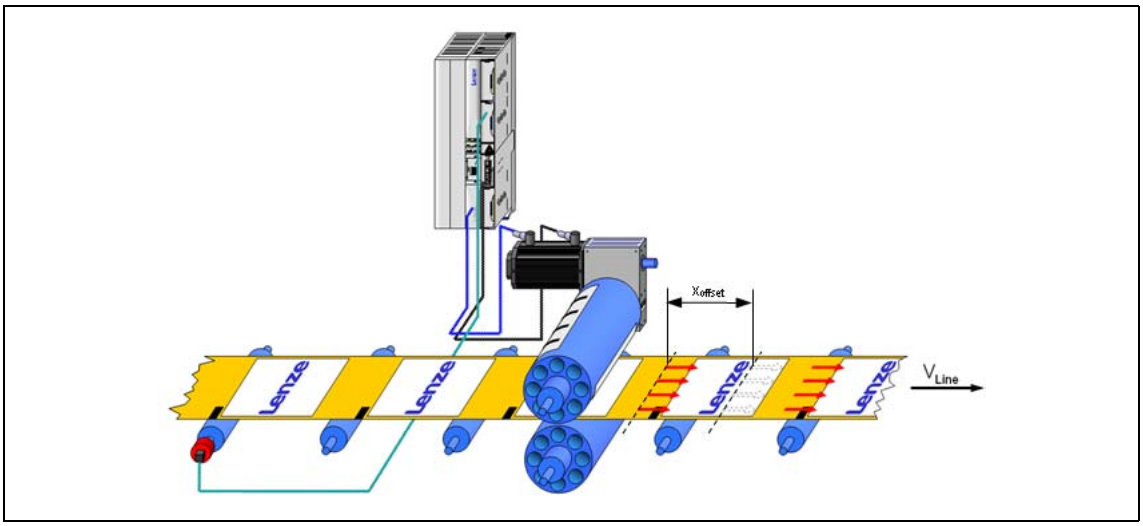
$$\frac{\text{dnYSpeedOut\_s}}{\text{dnXSpeedOut\_s}} = \frac{\frac{13}{2}}{\frac{17}{3}} \cdot \frac{110\ [\%]}{100\ [\%]} = 1.2618...$$

5.98.4 Angular trimming via x offset

In many production processes, several processing steps in a row are required for a product to be manufactured. Every machining process must be brought into line with the previous one to ensure the desired product quality. This is called angular trimming and normally achieved via a so-called x offset. The x offset leads to a virtual trimming of the master value and hence to a correct alignment of the following axis to the process.

Example 1: No angular trimming (without x offset)

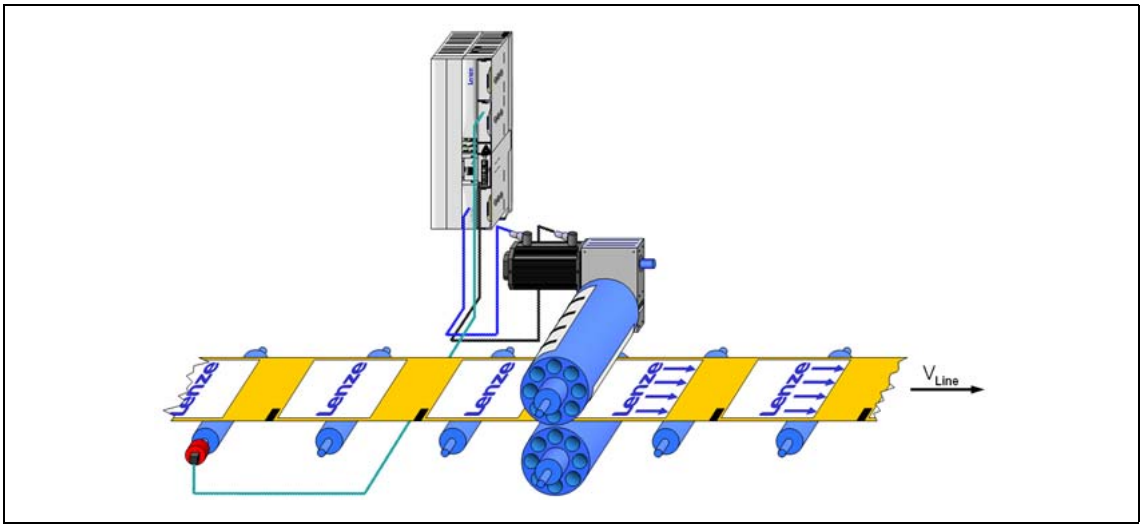
The "Arrows" print image is shifted by x_{offset} compared to the desired position:



[5-62] Example 1

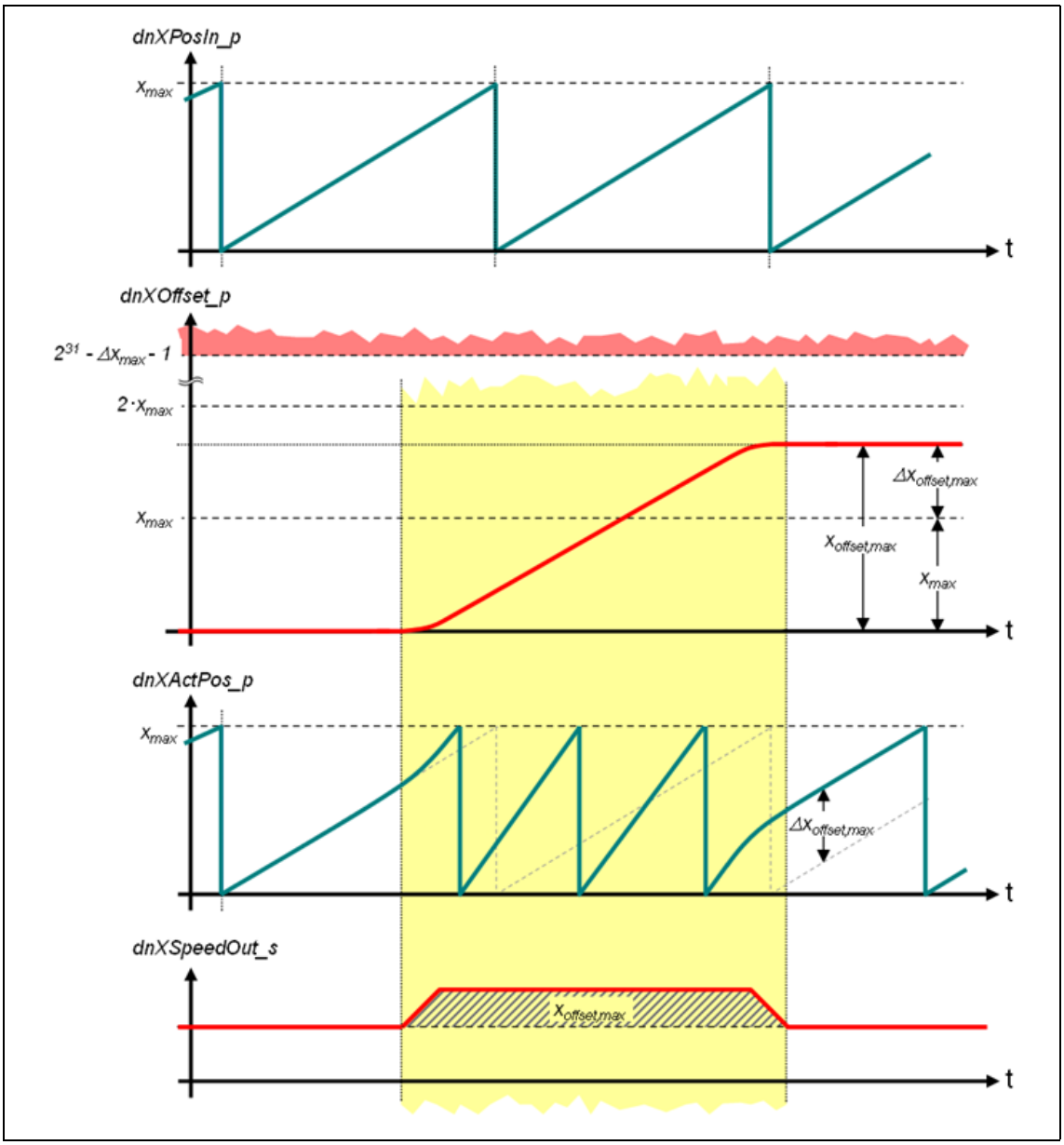
Example 2: Angular trimming via x offset

The "Arrows" print image is on the desired position:



[5-63] Example 2

The *dnXOffset_p* x offset is additively superimposed on the *dnXPosIn_p* master position and then projected again on the modulo cycle of the master measuring system. The resulting master position is output at the *dnXActPos_p* output.



[5-64] Example: Superimposing an offset on the master position

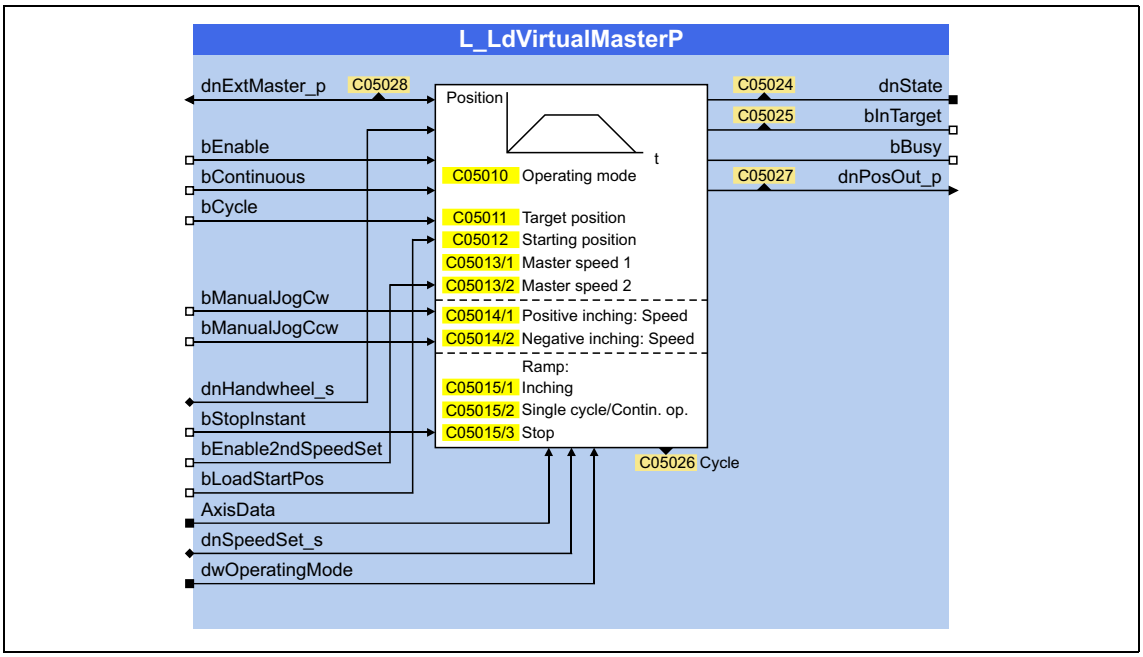
Tip!

- The *dnXOffset_p* offset value does not necessarily have to be limited to the modulo value range of the master cycle. However, it must not exceed the permissible DINT value range minus the master cycle.
  (DINT value range = $\pm 2^{31}$-1 [increments] ≡ ±32768 [rev])
- The x offset is offset in the **L_LdSyncOperation** FB in the same way as in the L_LdAddOffsetCyclic FB.

5.99 L_LdVirtualMasterP - virtual master for synchronism

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB provides the "Virtual master" function for the "Synchronism" application.



Inputs

Identifier/data type	Information/possible settings	
dnExtMaster_p DINT	Selection of an external master position in [inc] • E.g. through an absolute value encoder. • The master position must be inside the cycle. • C05028 indicates the external master position in the real unit of the machine.	
bEnable BOOL	Activate master function • This input has the highest priority.	
	FALSE	The input *dnExtMaster_p* connected 1:1 to the output *dnPosOut_p*. No ramps are active.
	TRUE	Ramps are activated.
	TRUE↘FALSE	Master shaft *dnPosOut_p* travels via the ramp for permanent operation with master speed to the external master position *dnExtMaster_p*. • Only a fixed master position can be approached. • If the master position changes, bit 29 of the status signal *dnState* will be set.
bContinuous BOOL	Control - continuous operation/production process • Only possible when: • C05010 = "0: Continuous operation" or • C05010 = "3: Enable speed and mode by process data" and *dwOperatingMode* = "0" (continuous operation). In this case, the speed is selected via the *dnSpeedSet_s* input.	
	TRUE	Continuous operation is started, master shaft *dnPosOut_p* travels continuously.
	TRUE↘FALSE	Continuous operation is stopped, master shaft *dnPosOut_p* travels to the stop position set in C05011.

Identifier/data type	Information/possible settings	
bCycle BOOL	Control - single cycle • Only possible when: • C05010 = "1: Inching" or • C05010 = "3: Enable speed and mode by process data" and *dwOperatingMode* = "1" (inching). In this case, the speed is selected via the *dnSpeedSet_s* input.	
	FALSE↗TRUE	Cyclic operation of the master shaft *dnPosOut_p* is started. • The master shaft travels to the stop position set in C05011. • A renewed cycle must be started with a renewed FALSE↗TRUE edge.
bManualJogCw BOOL	Inching - CW rotation • Only possible when: • C05010 = "1: Inching" or • C05010 = "3: Enable speed and mode by process data" and dwOperatingMode = "1" (inching). **Note:** The speeds specified in C05014/1 and C05014/2 are always valid for inching, independent of the preselected mode via code or input.	
	TRUE	Inching (manual jog) in positive direction.
bManualJogCcw BOOL	Inching - CCW rotation • Only possible when: • C05010 = "1: Inching" or • C05010 = "3: Enable speed and mode by process data" and dwOperatingMode = "1" (inching). **Note:** The speeds specified in C05014/1 and C05014/2 are always valid for inching, independent of the preselected mode via code or input.	
	TRUE	Inching (manual jog) in negative direction.
dnHandwheel_s DINT	Master speed for the handwheel function of the master shaft • At this input, the FB expects a speed in [rpm]. The machine parameters at the input *AxisData* are used as speed reference.	
bStopInstant BOOL	Stop master shaft • This input has the second highest priority (after *bEnable*).	
	FALSE↗TRUE	The master shaft *dnPosOut_p* is immediately braked to standstill via the set stop ramp.
bEnable2ndSpeedSet BOOL	Change over to second master speed	
	FALSE	Master speed 1 (C05013/1) is active.
	TRUE	Master speed 2 (C05013/2) is active.
bLoadStartPos BOOL	Set master shaft to starting position	
	FALSE↗TRUE	Master shaft *dnPosOut_p* is set to the starting position set in C05012.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510) **Note:** This FB only works with modulo measuring systems!	
dnSpeedSet_s DINT From library V02.07.xx.xx	Selection of master speed 1 via process date • This input is only active if the operating mode "3: Speed and mode via input" has been selected in C05010. **Note:** The master speed 2 is always selected via C05013/2.	

Identifier/data type	Information/possible settings	
dwOperatingMode DWORD From library V02.07.xx.xx	Selection of the operating mode via process date • This input is only active if the operating mode "3: Speed and mode via input" has been selected in C05010.	
	0	Continuous operation
	1	Inching
	2	Handwheel function

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	Bit 1	State active
	Bit 2	Procedure completed
	Bit 3	Acceleration/deceleration phase
	Bit 5	Counter-clockwise rotation
	Bit 10	Zero crossing active
	Bit 15	Error
	bit16	Starting position is beyond the cycle
	bits17	Error in the master signal (position or speed)
	Bit 18	No cycle available
	Bit 19	Maximum speed = "0"
	bits23	No valid axis data structure
	Bit 24	Single cycle
	bits25	Automatic mode
	Bit 26	Stopping is active
	Bit 27	Manual jog
	Bit 28	Internal setpoint speed is active
	Bit 29	Position difference - external master
	bits30	Electronic handwheel is active
bInTarget BOOL	Status signal "Master shaft has reached stop position"	
	TRUE	Master shaft *dnPosOut_p* has reached the stop position set in C05011.
bBusy BOOL	Status signal "master shaft is positioned"	
	TRUE	Master shaft *dnPosOut_p* is positioned.
dnPosOut_p DINT	Master position output (current position of the master shaft in the cycle) in [inc] • C05027 indicates the master position in the real unit of the machine.	

Parameter

Parameter	Possible settings		Info
C05010			VMaster: Operating mode
	0	Continuous operation	Lenze setting
	1	Inching	
	2	Handwheel function	
	3	Selection of speed and mode via input	Master speed 1 and the operating mode are selected via the corresponding inputs of the FB.

Parameter	Possible settings			Info
C05011	0.0000	Unit	214000.0000	Target position • The target position must be inside the cycle. • Initialisation: 0.0000 unit
C05012	0.0000	Unit	214000.0000	Starting position • The starting position must be inside the cycle. • Initialisation: 0.0000 unit
C05013/1	-214000.0000	Unit/t	214000.0000	Master speed 1 • Initialisation: 1.0000 unit/t
C05013/2	-214000.0000	Unit/t	214000.0000	Master speed 2 • Initialisation: 1.0000 unit/t
C05014/1	0.0000	Unit/t	214000.0000	Positive inching: Speed • Initialisation: 1.0000 unit/t
C05014/2	0.0000	Unit/t	214000.0000	Negative inching: Speed • Initialisation: 1.0000 unit/t
C05015/1	0.010	s	130.000	Ramp: Inching • Initialisation: 1.000 s
C05015/2	0.010	s	130.000	Ramp: Continuous operation/single cycle • Initialisation: 1.000 s
C05015/3	0.010	s	130.000	Ramp: Stop • Initialisation: 1.000 s
C05016/1 From library V02.02.xx.xx onwards	Character string			Position unit • Read only
C05016/2 From library V02.02.xx.xx onwards	Character string			Speed unit • Read only
C05017 From library V02.02.xx.xx onwards	-214748.3647	Unit/t	214748.3647	Reference speed • Read only • For input orientation for speed entries.
C05024	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.
C05025				Master position in the target • Like output *bInTarget* • Read only
	0	No		
	1	Yes		
C05026	-214748.3647	Unit	214748.3647	Cycle • Read only
C05027	-214748.3647	Unit	214748.3647	Master position at the output • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C05028	-214748.3647	Unit	214748.3647	External master position at the input • Display of the *dnExtMaster_p* input signal in the real unit of the machine.

5.99.1 Functions

Handwheel function

- Ensures an easy control of a machine run. The clutched drives are interconnected in a system.
- Incremental encoders via DFIN or multiple encoder input. An input must be selected.
- No ramp functions are active.
- Function is enabled via the control (PLC) by selecting the "handwheel function" operating mode (C05010 ="2") and setting *bEnable* to TRUE.

Manual jog (inching)

- Operation per HMI or keys via control (PLC). The locking is done in the PLC.
- Ramp functions are active.
- Parameters are defined once or changed via HMI.

Single cycle

- Operation per HMI or keys via control (PLC)
- Function starts with a FALSE↗TRUE edge at the *bCycle* input.
- Ramp functions are active.
- The data for production speed are used as parameters.
- The cycle starts at the current position and stops at the stop position (C05011).

Continuous operation

- Operation per HMI or keys via control (PLC)
- Function starts with a FALSE↗TRUE edge at the *bContinuous* input. The function remains active as long as the TRUE level is applied.
- Ramp functions are active.
- The data for production speed are used as parameters.
- The cycle starts at the current position and stops at the stop position (C05011).

Synchronise virtual master to the external master

- The virtual master is synchronised to an external position selection. Thus, the downstream system can be separated from the external master any time.
- Application: Modular machines: The external position is delivered by
  - MotionBus (Slave)
  - DFIN/multiple encoder input/FB **L_LdSetXAxisVelocity** (relative synchronism)
  - Fieldbus as real encoder (must be limited to the cycle set).
- The function is activated by setting *bEnable* to TRUE.

Profile generator/stopping at stop position:

- The process can be executed via several machine cycles (example: printing machines). The setting of the operating time is decisive.

Load starting position:

- The starting position set in C05012 is accepted by setting *bLoadStartPos* to TRUE.
- For this, the enable input *bEnable* must be set to TRUE.

Stop master shaft

- The master shaft is braked to standstill within the operating time set in C05015/3 setting *bStopInstant* to TRUE.
- For this, the enable input *bEnable* must be set to TRUE.
- This function has the highest priority.

Change over to second master speed

- By setting *bEnable2ndSpeedSet* to TRUE, the master speed 1 (C05013/1) is changed to master speed 2 (C05013/2).
- Ramp functions are active.

5.99.2 Monitoring for counter overflow

This function extension will be available from library V2.02!

If the machine parameters transferred to the FB via the *AxisData* input are changed afterwards (e.g. the gearbox factors), this may cause a counter overflow when the speed and position values defined via parameters are converted from the real unit to the internal unit.

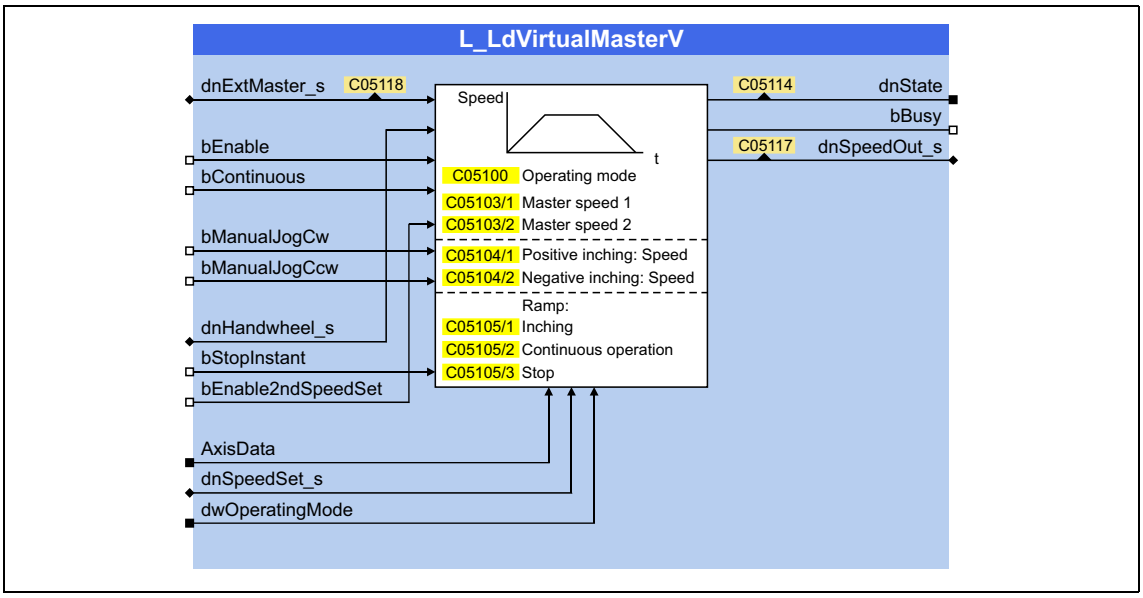
The FB causes a corresponding error message in case of an internal counter overflow due to a changed measuring system:

Error number		Error message in the logbook	Response
61669515	0x3AD008B	L_LdVirtualMasterP:int.speed overflow (LS_DriveInterface)	Error
61669387	0x3AD000B	L_LdVirtualMasterP:int.speed overflow (L_SdSetAxisData)	Error

5.100 L_LdVirtualMasterV - virtual master for electronic gearboxes

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB provides the "Virtual master" function for the "Electronic gearbox" application.



Inputs

Identifier/data type	Information/possible settings	
dnExtMaster_s DINT	Selection of an external master speed, e.g. from an encoder. • At this input, the FB expects a speed in [rpm]. The machine parameters at the input *AxisData* are used as speed reference. • C05118 indicates the external master speed in the real unit of the machine.	
bEnable BOOL	Activate master function • This input has the highest priority.	
	FALSE	The input *dnExtMaster_s* is connected through 1:1 to output *dnSpeedOut_s*. No ramps are active.
	TRUE	Master shaft *dnSpeedOut_s* is braked to standstill via the ramp function or follows the function selected.
	TRUE↘FALSE	Master shaft *dnSpeedOut_s* travels via the ramp for continuous operation with master speed *dnExtMaster_s*.
bContinuous BOOL	Control - continuous operation/production process • Only possible for the "continuous operation" mode set in C05100.	
	TRUE	Continuous operation is started, master shaft *dnSpeedOut_s* travels with setpoint speed.
	TRUE↘FALSE	Continuous operation is stopped, master shaft *dnSpeedOut_s* is braked to standstill via ramp for continuous operation.
bManualJogCw BOOL	Inching - CW rotation • Only possible for the "inching" mode set in C05100.	
	TRUE	Inching (manual jog) in positive direction.
bManualJogCcw BOOL	Inching - CCW rotation • Only possible for the "inching" mode set in C05100.	
	TRUE	Inching (manual jog) in negative direction.

Identifier/data type	Information/possible settings	
dnHandwheel_s DINT	Master speed for the handwheel function of the master shaft. • At this input, the FB expects a speed in [rpm]. The machine parameters at the input *AxisData* are used as speed reference.	
bStopInstant BOOL	Stop master shaft • This input has the second highest priority (after *bEnable*).	
	FALSE↗TRUE	The master shaft *dnSpeedOut_s* is immediately braked to standstill via the set stop ramp.
bEnable2ndSpeedSet BOOL	Change over to second master speed	
	FALSE	Master speed 1 (C05103/1) is active.
	TRUE	Master speed 2 (C05103/2) is active.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	
dnSpeedSet_s DINT As of library V02.07.xx.xx	Selection of master speed 1 via process date • This input is only active if the operating mode "3: Speed and mode via input" has been selected in C05100.	
dwOperatingMode DWORD As of library V02.07.xx.xx	Selection of the operating mode via process date • This input is only active if the operating mode "3: Speed and mode via input" has been selected in C05100.	
	0	Continuous operation
	1	Inching
	2	Handwheel function

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits1	State active
	bits3	Acceleration/deceleration phase
	bits5	Counter-clockwise rotation
	Bit15	Error
	bits17	Error in the master signal (position or speed)
	Bit 19	Maximum speed = "0"
	bits23	No valid axis data structure
	bits25	Automatic mode
	bits26	Stopping is active
	bits27	Manual jog
	bits28	Internal setpoint speed is active
	bits30	Electronic handwheel is active
bBusy BOOL	Status signal "Master shaft travels"	
	TRUE	Master shaft *dnSpeedOut_s* travels
dnSpeedOut_s DINT	Master speed output in [rpm] • C05117 indicates the master speed in the real unit of the machine.	

Parameter

Parameter	Possible settings			Information
C05100				VMaster: Operating mode
	0	Continuous operation		Lenze setting
	1	Inching		
	2	Handwheel function		
	3	Selection of speed and mode via input		Master speed 1 and the operating mode are selected via the corresponding inputs of the FB.
C05101 As of library V02.02.xx.xx	String of digits			Speed unit • Read only
C05102 As of library V02.02.xx.xx	-214748.3647	Unit/t	214748.3647	Reference speed • Read only • For input orientation for speed entries.
C05103/1	-214000.0000	Unit/t	214000.0000	Master speed 1 • Initialisation: 1.0000 unit/t
C05103/2	-214000.0000	Unit/t	214000.0000	Master speed 2 • Initialisation: 1.0000 unit/t
C05104/1	0.0000	Unit/t	214000.0000	Positive inching: Speed • Initialisation: 1.0000 unit/t
C05104/2	0.0000	Unit/t	214000.0000	Negative inching: Speed • Initialisation: 1.0000 unit/t
C05105/1	0.010	s	130.000	Ramp: Inching • Initialisation: 1.000 s
C05105/2	0.010	s	130.000	Ramp: Continuous operation/single cycle • Initialisation: 1.000 s
C05105/3	0.010	s	130.000	Ramp: Stop • Initialisation: 1.000 s
C05114	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.
C05117	-214748.3647	Unit	214748.3647	Speed at the output • Read only • Calculated from the speed signal *dnSpeedOut_s*.
C05118	-214748.3647	Unit	214748.3647	External speed at the input • Read only • Calculated from the speed signal *dnExtMaster_s*.

5.100.1 Functions

Handwheel function

- Ensures an easy control of a machine run. The clutched drives are interconnected in a system.
- Incremental encoders via DFIN or multiple encoder input. An input must be selected.
- No ramp functions are active.
- Function is enabled via the control (PLC) by selecting the "handwheel function" operating mode (C05100 ="2") and setting *bEnable* to TRUE.

Manual jog (inching)

- Operation per HMI or keys via control (PLC). The locking is done in the PLC.
- Ramp functions are active.
- Parameters are defined once or changed via HMI.

Continuous operation

- Operation per HMI or keys via control (PLC)
- Function starts with a FALSE↗TRUE edge at the *bContinuous* input. The function remains active as long as the TRUE level is applied.
- Ramp functions are active.
- The data for production speed are used as parameters.

Synchronise virtual master to the external master

- The virtual master is synchronised to an external speed selection. Thus, the downstream system can be separated from the external master any time.
- Application: Modular machines: The external speed is delivered by
  - MotionBus (Slave) as position selection
  - DFIN/multiple encoder input/FB **L_LdSetXAxisVelocity** (relative synchronism)
  - Fieldbus as real encoder as position selection (must be limited to the cycle set).
- The function is activated by setting *bEnable* to TRUE.

Stop master shaft

- The master shaft is braked to standstill within the operating time set in C05111 setting *bStopInstant* to TRUE.
- For this, the enable input *bEnable* must be set to TRUE.
- This function has the highest priority.

Change over to second master speed

- By setting *bEnable2ndSpeedSet* to TRUE, the master speed 1 (C05103/1) is changed to master speed 2 (C05103/2).
- Ramp functions are active.
- The function can be used e.g. for applications with reduced machine speed.

Second master speed for manual jog

- A change-over to master speed 2 via input *bEnable2ndSpeedSet* has also an effect on the "Manual jog" function:
  - Master speed 2 supersedes the manual speeds set in C05104/1 and C05104/2.
  - The direction of rotation is defined by the inputs *bManualJogCw* (positive) or *bManualJogCcw* (negative).

5.100.2 Monitoring for counter overflow

This function extension will be available from library V2.02!

If the machine parameters transferred to the FB via the *AxisData* input are changed afterwards (e.g. the gearbox factors), this may cause a counter overflow when the speed and position values defined via parameters are converted from the real unit to the internal unit.

The FB causes a corresponding error message in case of an internal counter overflow due to a changed measuring system:

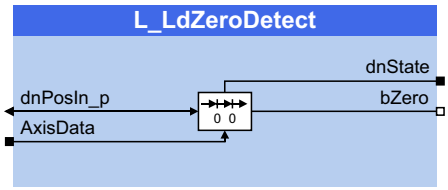
Error number		Error message in the logbook	Response
61669388	0x3AD000C	L_LdVirtualMasterV:int.speed overflow (LS_DriveInterface)	Error
61669516	0x3AD008C	L_LdVirtualMasterV:int.speed overflow (L_SdSetAxisData)	Error

5.101 L_LdZeroDetect - zero crossing detection

Function library:	LenzeLineDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB observes a clocked position and detects zero crossings.

- The cycle and shift factor for the position resolution are read out of the machine parameters (*AxisData*).



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Clocked position in [inc]
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

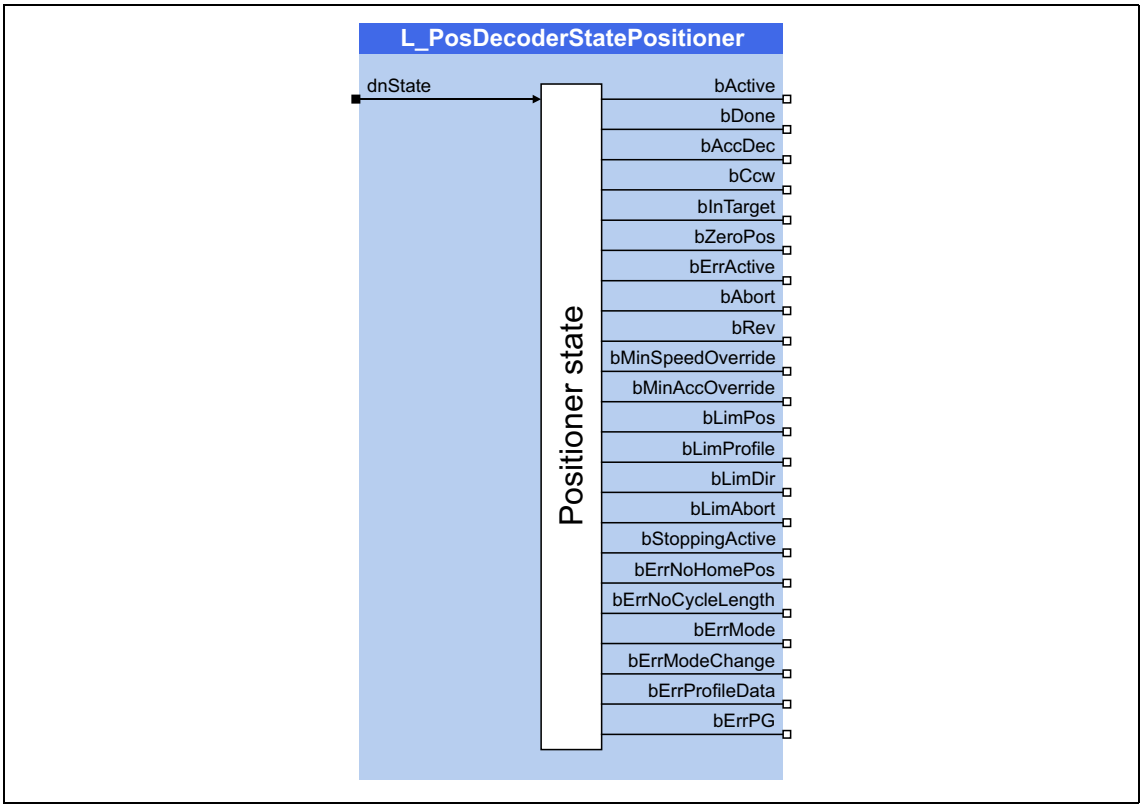
Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits10	Zero crossing active
	Bit15	Error
	bits18	No cycle available
	bits21	Input error (e.g. impermissible position jump or inverse direction of motion)
	bits23	No valid axis data structure
bZero BOOL	Status signal "Zero crossing"	
	TRUE	Zero crossing detected or position = "0"

5.102 L_PosDecoderStatePositioner - positioning status signals

Function library:	LenzePositioning	**Do not use FB for new developments!** Replacement by L_DevPositionerStateDecoder.
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB decodes the status output signal *dnState* of SB **LS_Positioner** into individual boolean status signals for further use in the FB interconnection.

- The functionality corresponds to a converter "DINT➞BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnState DINT	Input for the bit-coded status signal of the basic function "Positioning". • Connect this input with the output of the same name of the SB **LS_Positioner**.

Outputs

The boolean outputs have the following meaning when the value is TRUE:

Bit*	Identifier	Meaning in case of TRUE
Status messages:		
1	bActive	Positioning is active.
2	bDone	Positioning is completed (all profiles have been processed).
3	bAccDec	Acceleration/deceleration phase is active.
5	bCcw	Counter-clockwise rotation is active.
6	bInTarget	Set position reached (in case of sequence profiles the drive continues to travel).
10	bZeroPos	Zero crossing
16	bAbort	Positioning is aborted.
17	bRev	Reversing phase is active.
18	bMinSpeedOverride	Speed override ≤1 %
19	bMinAccOverride	Acceleration override ≤1 %
20	bLimPos	Position is limited by the basic function "Limiter" (SB **LS_Limiter**).
21	bLimProfile	Profile data is limited by the basic function "Limiter" (SB **LS_Limiter**).
22	bLimDir	Direction is inhibited by the basic function "Limiter" (SB **LS_Limiter**).
23	bLimAbort	Abort by basic function "Limiter" (SB **LS_Limiter**).
25	bStoppingActive	Stopping is active.
Error messages:		
15	bErrActive	Error is active.
24	bErrNoHomePos	Home position is not known.
26	bErrNoCycleLength	Cycle is not known.
27	bErrMode	Invalid positioning mode.
28	bErrModeChange	Invalid change of positioning mode.
29	bErrProfileData	Profile data is not plausible or faulty.
30	bErrPG	Error during profile generation.

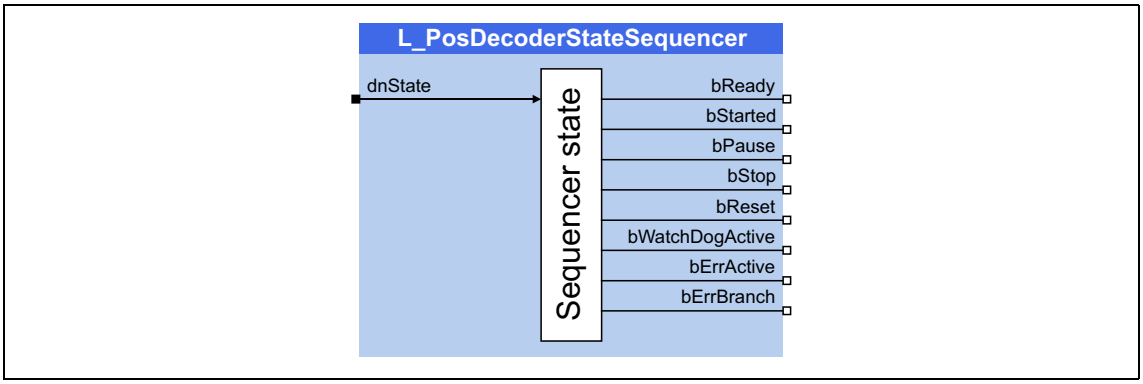
* Relating to the input signal *dnState*.

5.103 L_PosDecoderStateSequencer - status signals of the sequence control

Function library:	LenzePositioning	**Do not use FB for new developments!** Replacement by L_PosSequencerStateDecoder.
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB decodes the status output signal *dnState* of FB LS_PosSequencer into individual boolean status signals for further use in the FB interconnection.

- The functionality corresponds to a converter "DINT→BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnState DINT	Input for the bit coded status signal of the sequence control. • Connect this input with the output of the same name of the FB L_PosSequencer. (📖 430)

Outputs

The boolean outputs have the following meaning when the value is TRUE:

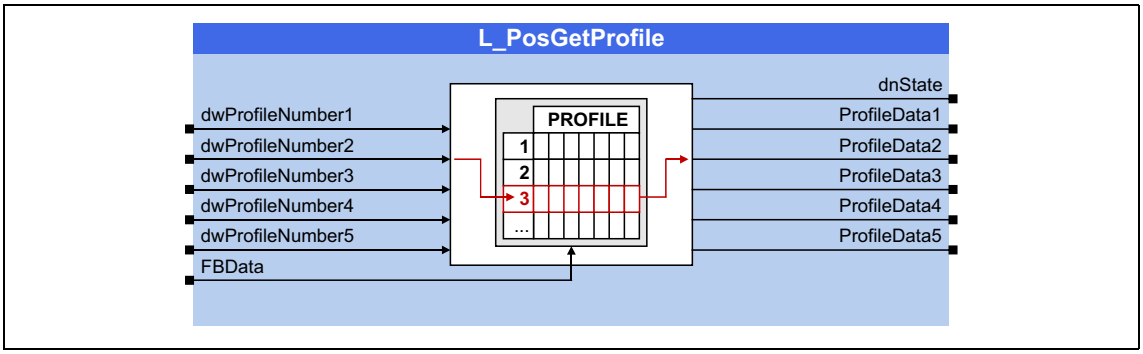
Bit*	Identifier	Meaning in case of TRUE
Status messages:		
16	bReady	Positioning program ready to start/program end reached.
17	bStarted	Positioning program is running.
18	bPause	Positioning program is started, pause is active.
19	bStop	Positioning program is stopped.
20	bReset	Positioning program is reset.
Error messages:		
22	bWatchDogActive	"Watchdog" monitoring function for positioning has been activated.
15	bErrActive	Error is active.
25	bErrBranch	Wrong value at branch input.

* Relating to the input signal *dnState*.

5.104 L_PosGetProfile - profile data tables

Function library:	LenzePositioning
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

Together with an FB instance of type L_PosPositionerTable or L_PosProfileTable this FB provides five profile data sets that can be selected.



Inputs

Identifier/data type	Information/possible settings	
dwProfileNumber**1...5** DWORD	Number of the profile whose profile data is to be output at *ProfileData1* ... *ProfileData5*.	
	In connection with an FB instance of type **L_PosPositionerTable**:	
	1 ... 75	Profiles 1 ... 75
	In connection with an FB instance of type **L_PosProfileTable**:	
	1 ... 4	Profile 1 ... 4
FBData	Interface for the transmission of the FB instance data. • Connect this input with the output of the same name of the FB instance of type **L_PosPositionerTable** or **L_PosProfileTable**.	

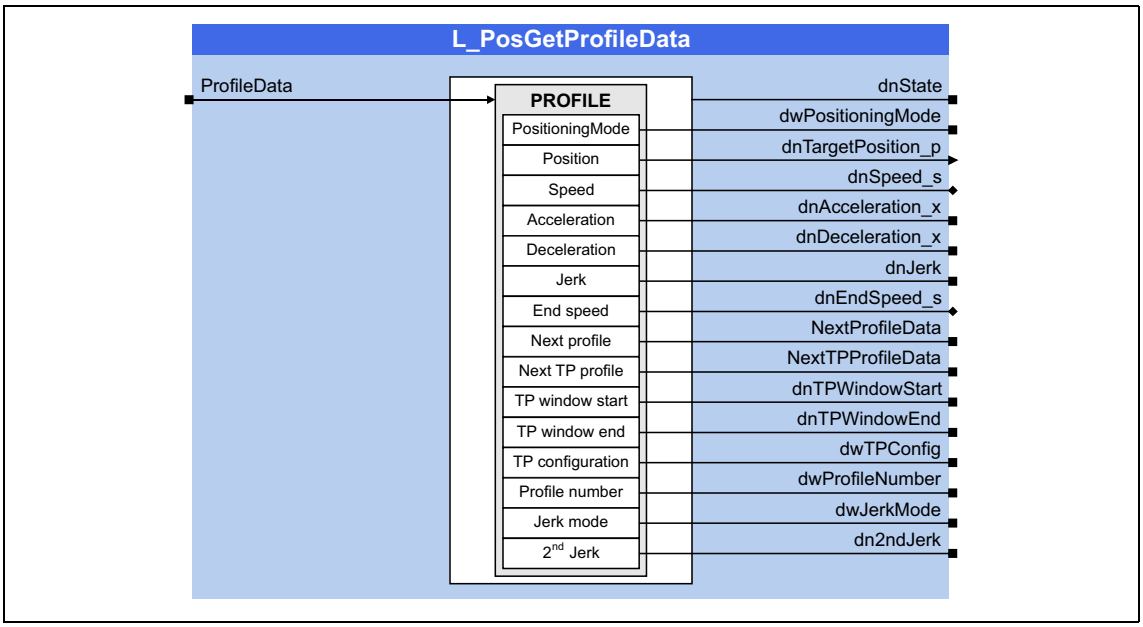
Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	Bit15	Error is active (collective message).
	bit16	*dwProfileNumber1* is not within the valid range.
	bits17	*dwProfileNumber2* is not within the valid range.
	bits18	*dwProfileNumber3* is not within the valid range.
	Bit 19	*dwProfileNumber4* is not within the valid range.
	bits20	*dwProfileNumber5* is not within the valid range.
	bits27	Error: Invalid FB instance data.
ProfileData**1...5**	Profile data of the profile selected via *dwProfileNumber1...5*.	

5.105 L_PosGetProfileData - profile data output

Function library:	LenzePositioning
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB outputs the profile data of a profile data set.



Inputs

Identifier/data type	Information/possible settings
ProfileData	Interface for the transmission of the profile data set.

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	Bit15	Error is active (collective message).
	bits29	Error: Invalid profile data.
dwPositioningMode DWORD	Positioning mode	
	1	Absolute
	2	Absolute TP
	5	Relative
	6	Relative TP
	7	Speed
	8	Speed TP
	11	Absolute CW
	12	Absolute CW TP
	13	Absolute CCW
	14	Absolute CCW TP
	15	Absolute shortest way
	16	Absolute shortest way TP
dnTargetPosition_p DINT	Target position in [inc]	
dnSpeed_s DINT	Traversing velocity as speed • Scaling: $2^{26} \equiv 15000$ [rpm]	
dnAcceleration_x DINT	Acceleration as speed variation/time • Scaling: $2^{22} \equiv 15000000$ [rpm/s]	
dnDeceleration_x DINT	Deceleration as speed variation/time • Scaling: $2^{22} \equiv 15000000$ [rpm/s]	
dnJerk DINT	S-ramp time in [ms]	
dnEndSpeed_s DINT	Final velocity as speed • Scaling: $2^{26} \equiv 15000$ [rpm]	
NextProfileData	Profile data of the standard sequence profile • For profile linkage.	
NextTPProfileData	Profile data of the TP sequence profile • For touch probe residual path positioning.	
dnTPWindowStart DINT	TP window starting position	
dnTPWindowEnd DINT	TP window end position	

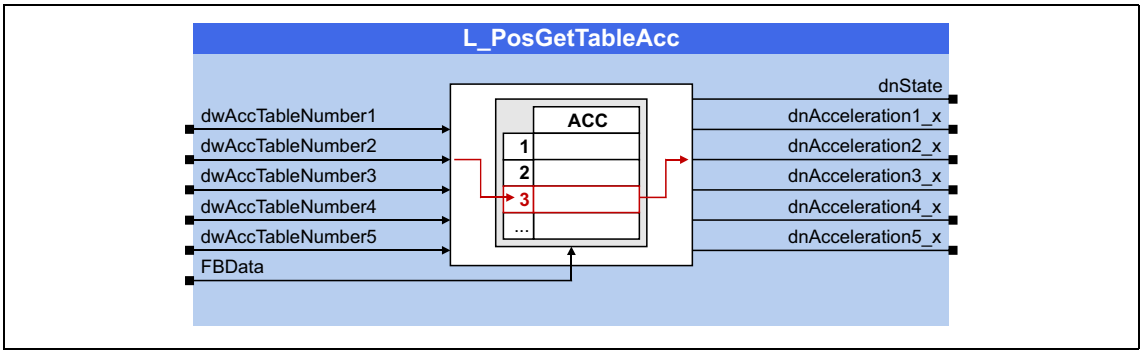
Identifier/data type	Value/meaning		
dwTPConfig DWORD	Touch-probe configuration (bit coded) • For the touch probe channels to be used, the corresponding bit must be set to "1":		
	bits0	TP channel 1, rising edge	Digital input 1
	bits1	TP channel 1, falling edge	
	bit 2	TP channel 2, rising edge	Digital input 2
	bits3	TP channel 2, falling edge	
	bits4	TP channel 3, rising edge	Digital input 3
	bits5	TP channel 3, falling edge	
	bits6	TP channel 4, rising edge	Digital input 4
	bits7	TP channel 4, falling edge	
	bits8	TP channel 5, rising edge	Digital input 5
	bits9	TP channel 5, falling edge	
	bits10	TP channel 6, rising edge	Digital input 6
	bits11	TP channel 6, falling edge	
	bits12	TP channel 7, rising edge	Digital input 7
	bits13	TP channel 7, falling edge	
	bits14	TP channel 8, rising edge	Digital input 8
	Bit15	TP channel 8, falling edge	
	bit16	TP channel 9	Motor encoder zero pulse
	bits18	TP channel 10	Load encoder zero pulse
	bits20	TP channel 11	DFIN zero pulse
	bits22	TP channel 12	DFOUT zero pulse
dwProfileNumber DWORD	Profile number of the profile data set		
dwJerkMode DWORD As of library V02.05.xx.xx	S-ramp mode • Will be supported as of controller software version V5.0. With a software version < V5.0, this output is always set to zero.		
	0	Second S-ramp time is not active. • The *dnJerk* S-ramp time is used for the acceleration and deceleration phase.	
	1	Second S-ramp time is active for deceleration. • The *dnJerk* S-ramp time is only used for the acceleration phase. • For the deceleration phase, however, the second *dn2ndJerk* S-ramp time is used.	
dn2ndJerk DINT As of library V02.05.xx.xx	Second S-ramp time in [ms] • Will be supported as of controller software version V5.0. With a software version < V5.0, this output is always set to zero. • The second S-ramp time is only used for S-ramp mode "1" for the deceleration phase.		

5.106 L_PosGetTableAcc - acceleration table

Function library:	LenzePositioning
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

Together with an FB instance of type L_PosPositionerTable this FB provides five acceleration values from the VTACC variable table.

In connection with an FB instance of type L_PosProfileTable it is possible to output the acceleration and deceleration values that are directly defined in the profiles.



Inputs

Identifier/data type	Information/possible settings	
dwAccTableNumber**1...5** DWORD	Number of the table position or profile whose acceleration value is to be output at *dnAcceleration1_x ... dnAcceleration5_x*.	
	In connection with an FB instance of type **L_PosPositionerTable**:	
	1 ... 50	Table position 1 ... 50 of the VTACC table
	In connection with an FB instance of type **L_PosProfileTable**:	
	1	Acceleration of profile 1
	2	Acceleration of profile 2
	3	Acceleration of profile 3
	4	Acceleration of profile 4
	5	Deceleration of profile 1
	6	Deceleration of profile 2
	7	Deceleration of profile 3
	8	Deceleration of profile 4
FBData	Interface for the transmission of the FB instance data. • Connect this input with the output of the same name of the FB instance of type **L_PosPositionerTable** or **L_PosProfileTable**.	

Outputs

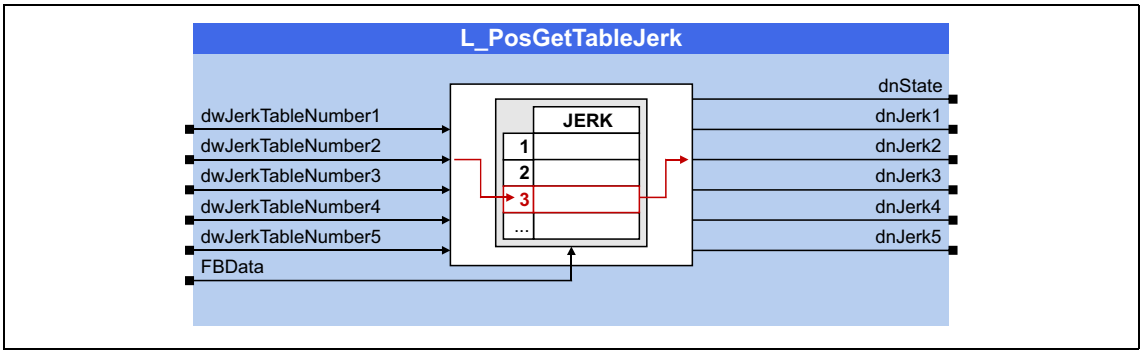
Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	Bit15	Error is active (collective message).
	bit16	*dwAccTableNumber1* is not within the valid range.
	bits17	*dwAccTableNumber2* is not within the valid range.
	bits18	*dwAccTableNumber3* is not within the valid range.
	Bit 19	*dwAccTableNumber4* is not within the valid range.
	bits20	*dwAccTableNumber5* is not within the valid range.
	bits27	Error: Invalid FB instance data.
dnAcceleration**1...5**_x DINT	Acceleration value parameterised at table position or in profile number *dwAccTableNumber1...5*. • Output as speed variation/time • Scaling: $2^{22} \equiv 15000000$ [rpm/s]	

5.107 L_PosGetTableJerk - S-ramp time table

Function library:	LenzePositioning
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

Together with an FB instance of type L_PosPositionerTable this FB provides five S-ramp times from the VTJERK variable table.

In connection with an FB instance of type L_PosProfileTable it is possible to output the S-ramp times that are directly defined in the profiles.



Inputs

Identifier/data type	Information/possible settings	
dwJerkTableNumber**1...5** DWORD	Number of the table position or profile, the speed value of which is to be output at *dnJerk1* ... *dnJerk5*.	
	In connection with an FB instance of type **L_PosPositionerTable**:	
	1 ... 50	Table position 1 ... 50 of the VTJERK table
	In connection with an FB instance of type **L_PosProfileTable**:	
	1 ... 4	Profile 1 ... 4
FBData	Interface for the transmission of the FB instance data. • Connect this input with the output of the same name of the FB instance of type **L_PosPositionerTable** or **L_PosProfileTable**.	

Outputs

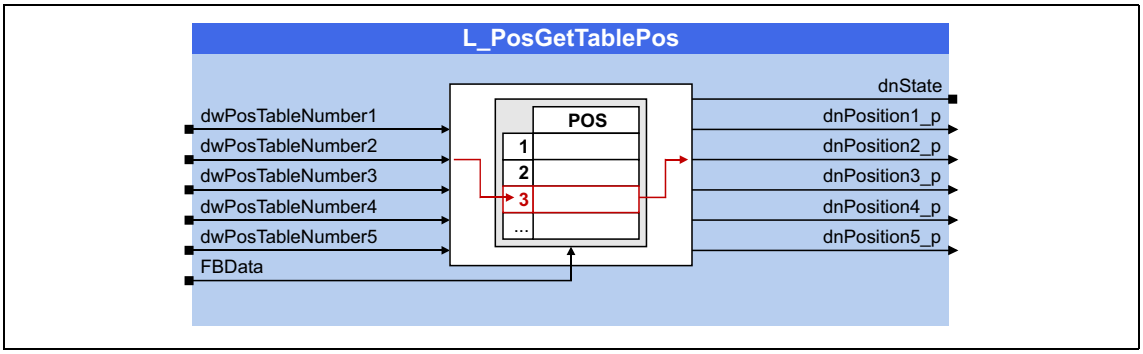
Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	Bit15	Error is active (collective message).
	bit16	*dwJerkTableNumber1* is not within the valid range.
	bits17	*dwJerkTableNumber2* is not within the valid range.
	bits18	*dwJerkTableNumber3* is not within the valid range.
	Bit 19	*dwJerkTableNumber4* is not within the valid range.
	bits20	*dwJerkTableNumber5* is not within the valid range.
	bits27	Error: Invalid FB instance data.
dnJerk**1...5**_s DINT	S-ramp time in [ms] parameterised at table position or in profile number *dwJerkTableNumber1...5*.	

5.108 L_PosGetTablePos - position table

Function library:	LenzePositioning
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

Together with an FB instance of type L_PosPositionerTable this FB provides five position values from the VTPOS variable table.

In connection with an FB instance of type L_PosProfileTable it is possible to output the positions that are directly defined in the profiles.



Inputs

Identifier/data type	Information/possible settings	
dwPosTableNumber**1...5** DWORD	Number of the table position or profile, the position value of which is to be output at *dnPosition1_p ... dnPosition5_p*.	
	In connection with an FB instance of type **L_PosPositionerTable**:	
	1 ... 50	Table position 1 ... 75 of the VTPOS table
	In connection with an FB instance of type **L_PosProfileTable**:	
	1 ... 4	Profile 1 ... 4
FBData	Interface for the transmission of the FB instance data. • Connect this input with the output of the same name of the FB instance of type **L_PosPositionerTable** or **L_PosProfileTable**.	

Outputs

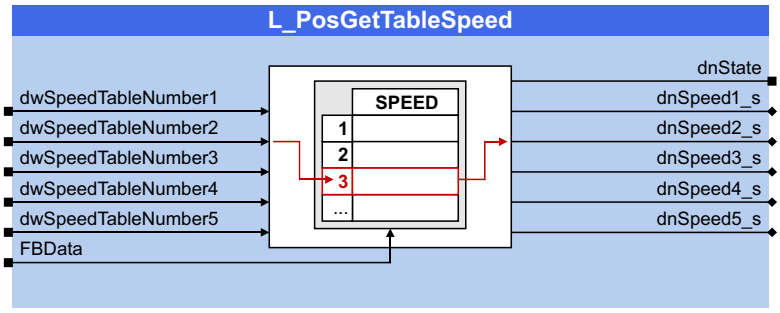
Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	Bit15	Error is active (collective message).
	bit16	*dwPosTableNumber1* is not within the valid range.
	bits17	*dwPosTableNumber2* is not within the valid range.
	bits18	*dwPosTableNumber3* is not within the valid range.
	Bit 19	*dwPosTableNumber4* is not within the valid range.
	bits20	*dwPosTableNumber5* is not within the valid range.
	bits27	Error: Invalid FB instance data.
dnPosition**1...5**_p DINT	Position in [inc] parameterised at table position or in profile number *dwPosTableNumber1...5*.	

5.109 L_PosGetTableSpeed - speed table

Function library:	LenzePositioning
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

Together with an FB instance of type L_PosPositionerTable this FB provides five speed values from the VTSPEED variable table.

In connection with an FB instance of type L_PosProfileTable it is possible to output the speed values that are directly defined in the profiles.



Inputs

Identifier/data type	Information/possible settings	
dwSpeedTableNumber**1...5** DWORD	Number of the table position or profile, the speed value of which is to be output at *dnSpeed1_s ... dnSpeed5_s*.	
	In connection with an FB instance of type **L_PosPositionerTable**:	
	1 ... 50	Table position 1 ... 50 of the VTSPEED table
	In connection with an FB instance of type **L_PosProfileTable**:	
	1 ... 4	Profile 1 ... 4
FBData	Interface for the transmission of the FB instance data. • Connect this input with the output of the same name of the FB instance of type **L_PosPositionerTable** or **L_PosProfileTable**.	

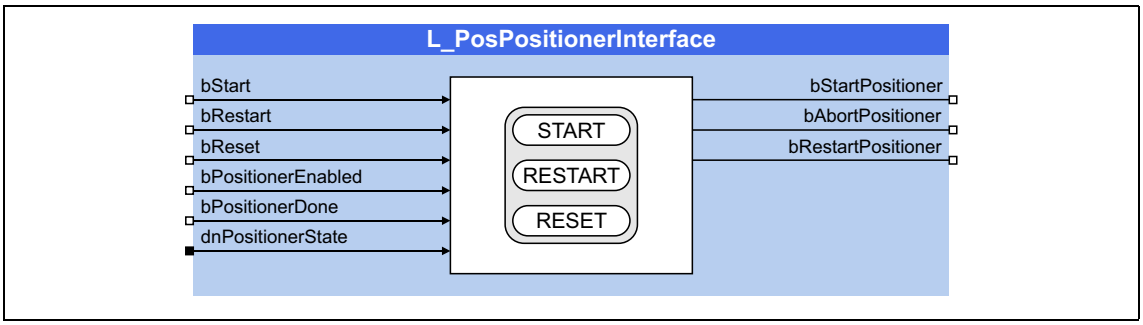
Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	Bit15	Error is active (collective message).
	bit16	*dwSpeedTableNumber1* is not within the valid range.
	bits17	*dwSpeedTableNumber2* is not within the valid range.
	bits18	*dwSpeedTableNumber3* is not within the valid range.
	Bit 19	*dwSpeedTableNumber4* is not within the valid range.
	bits20	*dwSpeedTableNumber5* is not within the valid range.
	bits27	Error: Invalid FB instance data.
dnSpeed**1...5**_s DINT	Velocity parameterised on table position or in profile number *dwSpeedTableNumber1...5* is output as speed. • Scaling: $2^{26} \equiv 15000$ [rpm]	

5.110 L_PosPositionerInterface - positioning interface

Function library:	LenzePositioning	**FB is available as of library V02.00.xx.xx! FB may only be used for firmware as of V1.50!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB provides an interface for the basic function "Positioning" for a higher-level control and allows to control the SB **LS_Positioner** via level-controlled signals.



Inputs

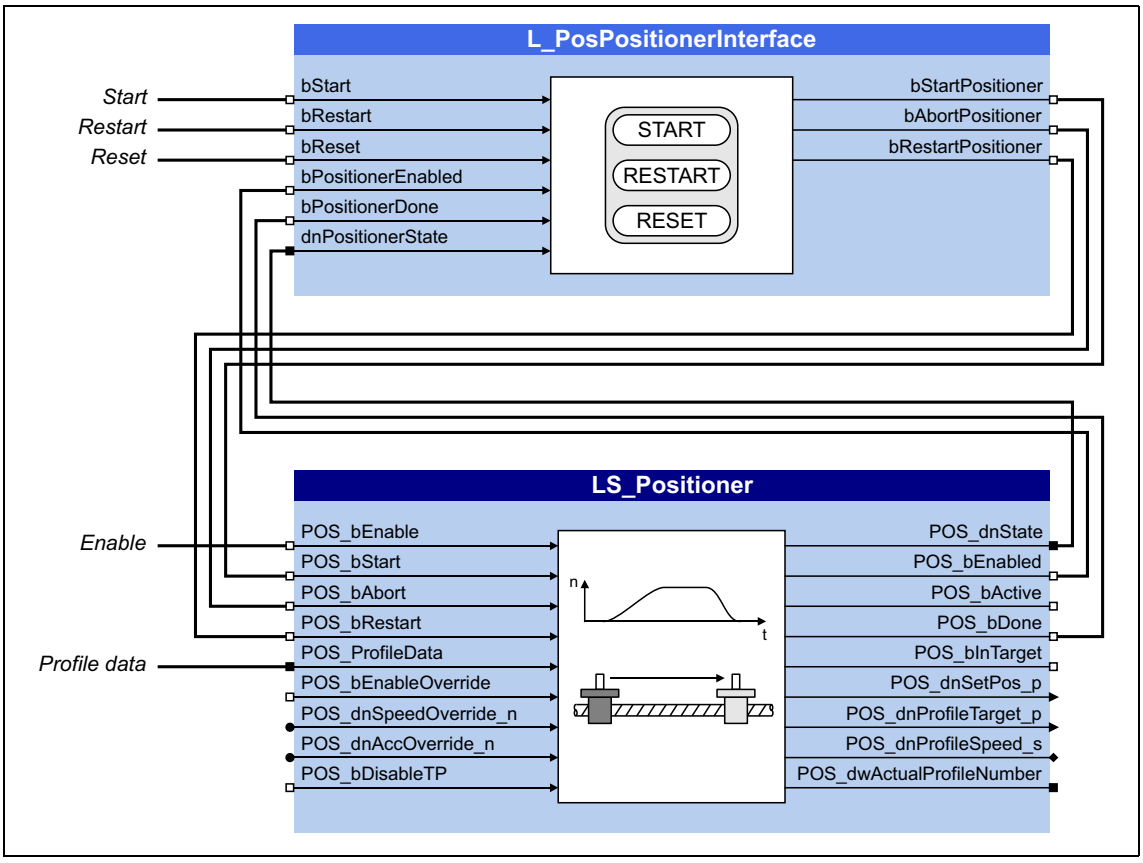
Identifier/data type	Information/possible settings	
bStart BOOL	Start/continue positioning	
	TRUE	Start/continue positioning.
	FALSE	Cancel positioning.
bRestart BOOL	Profile restart	
	FALSE↗TRUE	Read profile data once again considering the distance already traversed.
bReset BOOL	Profile reset	
	FALSE↗TRUE	Read profile data once again without considering the distance already traversed.
bPositionerEnabled BOOL	Interface to basic function "Positioning" • Connect this input with the output *POS_bEnabled* of the SB **LS_Positioner**.	
bPositionerDone BOOL	Interface to basic function "Positioning" • Connect this input with the output *POS_bDone* of the SB **LS_Positioner**.	
dnPositionerState DINT From library V02.07.xx.xx	Interface to basic function "Positioning" • Connect this input to the *POS_dnState* output of the **LS_Positioner** SB. • If this input is interconnected, the positioning process is restarted every time an error is detected or a restart of the **LS_Positioner** SB is impossible when the next edge at *bStart* is reached.	

Outputs

Identifier/data type	Value/meaning
bStartPositioner BOOL	Interface to basic function "Positioning" • Connect this output with the input *POS_bStart* of the SB **LS_Positioner**.
bAbortPositioner BOOL	Interface to basic function "Positioning" • Connect this output with the input *POS_bAbort* of the SB **LS_Positioner**.
bRestartPositioner BOOL	Interface to basic function "Positioning" • Connect this output with the input *POS_bRestart* of the SB **LS_Positioner**.

5.110.1 Interconnection

Connections required between the FB **L_PosPositionerInterface** and the SB **LS_Positioner**:



[5-65] Interconnection

Note!

If the *dnPositionerState* input is interconnected and positioning cannot be continued, the next positive edge at *bStart* will trigger a restart of the positioning process.

- If this behaviour is not required, do not interconnect the *dnPositionerState* input. In this case, a manual reset via the *bReset* input is necessary.

5.110.2 Applications

Application		FB control	Notes
Start positioning		*bStart* = TRUE	
Cancel/interrupt positioning		*bStart* = FALSE	
Continue positioning		*bStart* = FALSE↗TRUE	Profile data will be read again; the distance already traversed will be considered.
Update profile data during positioning (*bStart* = TRUE)		*bRestart* = FALSE↗TRUE	Profile data will be read again; the distance already traversed will be considered.
Restart positioning			
	after completion of positioning (*POS_bDone* = TRUE)	*bStart* = FALSE (≥ 1 cycle) *bStart* = TRUE **or** *bStart* = TRUE *bReset* = FALSE↗TRUE	
	after cancellation of positioning (*bStart* = FALSE)	*bReset* = FALSE↗TRUE *bStart* = FALSE↗TRUE	
	during positioning (*bStart* = TRUE)	*bReset* = FALSE↗TRUE	
Reset the status "Positioning completed" after end of positioning		*bStart* = FALSE	Status output *POS_bDone* will be reset to FALSE.

Continuing an interrupted positioning process

If the basic function "Positioning" is deactivated during the active positioning process (e. g. by quick stop or controller inhibit), the interrupted positioning is tried to be continued if the basic function "Positioning" is activated again, as far as the input *bStart* is set to TRUE.

- Distances of a relative positioning, that have already been covered, are taken into consideration.
- The positioning is not continued if incorrect profile data are available after the interruption, or if the **LS_Positioner** SB reports another error.
- If it is not possible to continue a positioning process, this is displayed via bit 11 of the status output *POS_dnState* at the **LS_Positioner** SB.

For controllers with software version lower than V3.0 the following applies:

- A positioning process can not be continued if another function state except for "Quick stop active", "Drive is stopped", or "Drive in standstill", has been active in the meantime. In this case the FB has to be reset with a FALSE-TRUE edge at the input *bReset*.

For controllers as of software version V3.0 the following applies:

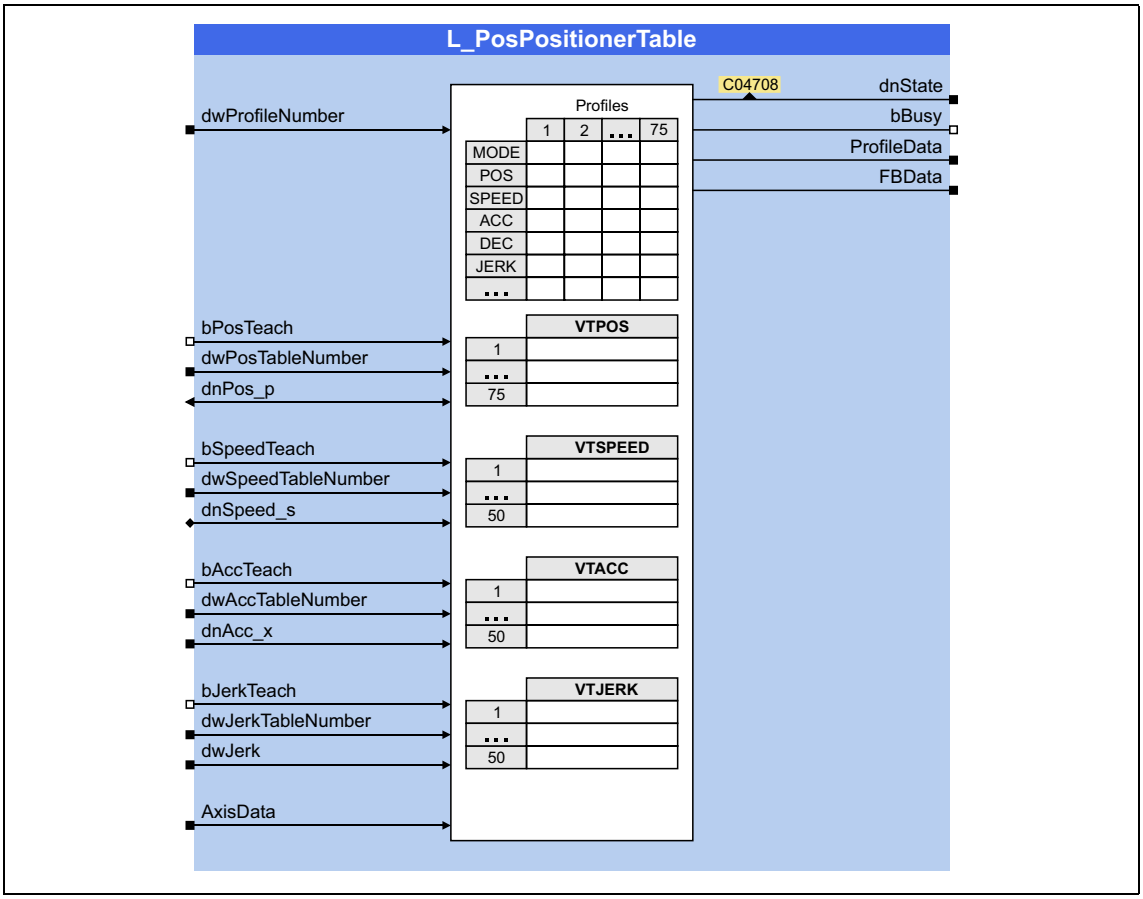
- A positioning process can not be continued if the machine parameters have been altered in the meantime, or if another referencing has taken place.

5.111 L_PosPositionerTable - profile data record management

Function library:	LenzePositioning	
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB is used to store and manage (traversing) profiles and to teach positions, speeds, accelerations/decelerations and S-ramp times.

- A profile describes a motion request which can be implemented into a rotary motion by the SB **LS_Positioner**.
- A profile is described via the following profile parameters: mode (type of positioning), position, speed, acceleration, deceleration, S-ramp time, final speed, standard sequence profile, TP sequence profile, TP window starting and end position.
  - For easier handling, the profile parameters position, speed, acceleration/deceleration and S-ramp time are kept in separate tables.
  - Thus, a positioning program for e.g. repeating speeds can always use the same table position. If this speed is to be changed, an adjustment of a table position can have an effect on several profiles.
- Another important task of this FB is the conversion of the table values according to the scaling/machine parameters selected in the SB **LS_DriveInterface**.



Inputs

Identifier/data type	Information/possible settings	
dwProfileNumber DWORD	Profile selection • Number of the profile the profile parameters of which are to be transferred to the SB **LS_Positioner** via the output *ProfileData*. • If a sequence profile is defined in the selected profile, this profile is only the first profile of an arbitrary linkage of profiles.	
	1...75	Profile number
bPosTeach BOOL	"Teach" position	
	TRUE	The position assigned to the input *dnPos_p* is stored at the VTPOS table position selected via the input *dwPosTableNumber*.
	TRUE↘FALSE	The value of the last cycle is maintained in the table.
dwPosTableNumber DWORD	Selection of the table position of the position to be "taught"	
	0	No selection (selection in case the input is not assigned)
	1...75	Table position 1 ... 75 of the VTPOS table
dnPos_p DINT	Position to be "taught" in [inc]	
bSpeedTeach BOOL	"Teach" speed	
	TRUE	The speed assigned to the input *dnSpeed_s* is stored at the VTSPEED table position selected via the input *dwSpeedTableNumber*.
	TRUE↘FALSE	The value of the last cycle is maintained in the table.
dwSpeedTableNumber DWORD	Selection of the table position for the speed to be "taught"	
	0	No selection (selection in case the input is not assigned)
	1...50	Table position 1 ... 50 of the VTSPEED table
dnSpeed_s DINT	Speed to be "taught" • At this input, the FB expects a speed. The machine parameters at the input *AxisData* are used as speed reference. • Scaling: $2^{26} \equiv 15000$ [rpm]	
bAccTeach BOOL	"Teach" acceleration/deceleration	
	TRUE	The acceleration/deceleration assigned to the input *dnAcc_s* is stored at the VTACC table position selected via the input *dwAccTableNumber*.
	TRUE↘FALSE	The value of the last cycle is maintained in the table.
dwAccTableNumber DWORD	Selection of the table position for the acceleration/deceleration speed to be "taught"	
	0	No selection (selection in case the input is not assigned)
	1...50	Table position 1 ... 50 of the VTACC table
dnAcc_x DINT	Acceleration/deceleration to be "taught" • At this input, the FB expects a speed variation/time. The machine parameters at the input *AxisData* are used as acceleration reference. • Scaling: $2^{22} \equiv 15000000$ [rpm/s]	
bJerkTeach BOOL	"Teach" S-ramp time	
	TRUE	The S-ramp time assigned to the input *dwJerk* is stored at the VTJERK table position selected via the input *dwJerkTableNumber*.
	TRUE↘FALSE	The value of the last cycle is maintained in the table.
dwJerkTableNumber DWORD	Selection of the table position for the S-ramp time to be "taught"	
	0	No selection (selection in case the input is not assigned)
	1...50	Table position 1 ... 50 of the VTJERK table

Identifier/data type	Information/possible settings
dwJerk DWORD	S-ramp time to be "taught" in [ms]
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits15	Error is active.
	bits16	Calculation is active.
	bits17	Position taught.
	bits18	Speed taught.
	bits19	Acceleration taught.
	bits20	S-ramp time taught.
	bits28	Error: Invalid machine parameters (*AxisData*).
	bits29	Error: Invalid profile number (*dwProfileNumber*).
	bits30	Error: Invalid teach profile number.
bBusy BOOL	Status signal "Conversion of the profile parameter is active"	
	FALSE	Profile parameters are ready.
	TRUE	Conversion is active, no access possible to the profile parameters at the moment.
ProfileData	Interface to basic function "Positioning" • Data structure with the profile parameters of the profile selected via the input *dwProfileNumber*. • Connect this output with the input of the same name of the SB **LS_Positioner**.	
FBData	FB interface for internal profiles/table data (L_PosGetProfile, L_PosGetTableAcc, L_PosGetTableJerk, L_PosGetTablePos, L_PosGetTableSpeed) • Data structure with the internal instance data.	

Parameter

Parameter	Possible settings		Information
C04700/1			"Teach" position • For execution via parameter channel.
	0↗1	The position assigned to the input *dnPos_p* is stored at the VTPOS table position selected via the input *dwPosTableNumber*.	
C04700/2			"Teach" speed • For execution via parameter channel.
	0↗1	The speed assigned to the input *dnSpeed_s* is stored at the VTSPEED table position selected via the input *dwSpeedTableNumber*.	

Parameter	Possible settings			Information
C04700/3				Teach acceleration • For execution via parameter channel.
	0↗1	The acceleration/deceleration assigned to the input *dnAcc_s* is stored at the VTACC table position selected via the input *dwAccTableNumber*.		
C04700/4				"Teach" S-ramp time • For execution via parameter channel.
	0↗1	The S-ramp time assigned to the input *dwJerk* is stored at the VTJERK table position selected via the input *dwJerkTableNumber*.		
C04701/1...75	-214748.3647	Unit	214748.3647	VTPOS: variable positions • For the selection of the target positions and the TP window starting or end positions for the profiles.
C04702/1...50	-214748.3647	Unit/s	214748.3647	VTSPEED: variable speeds • For the selection of the traversing and final speeds for the profiles.
C04703/1...50	0.0000	unit/s^2	214748.3647	VTACC: variable accelerations • For the selection of the accelerations and decelerations for the profiles.
C04704/1...50	0.000	s	2147483.647	VTJERK: variable S-ramp times • For the selection of the S-ramp times for the profiles.
C04705/1				DIS:bPosTeach • Display of the *bPosTeach* input signal.
	0	FALSE		
	1	TRUE		
C04705/2				DIS:bSpeedTeach • Display of the *bSpeedTeach* input signal.
	0	FALSE		
	1	TRUE		
C04705/3				DIS:bAccTeach • Display of the *bAccTeach* input signal.
	0	FALSE		
	1	TRUE		
C04705/4				DIS:bJerkTeach • Display of the *bJerkTeach* input signal.
	0	FALSE		
	1	TRUE		
C04706/1	0		100	DIS:VTPosNum • Display of the *dwPosTableNumber* input signal.
C04706/2	0		100	DIS:VTSpeedNum • Display of the *dwSpeedTableNumber* input signal.
C04706/3	0		100	DIS:VTAccNum • Display of the *dwAccTableNumber* input signal.
C04706/4	0		100	DIS:VTJerkNum • Display of the *dwJerkTableNumber* input signal.
C04706/5	0		100	DIS:ProfileNum • Display of the *dwProfileNumber* input signal.

Parameter	Possible settings			Information
C04707/1	-214748.3647	Unit	214748.3647	DIS:dnTeachPos • Display of the *dnPos_p* input signal in the real unit of the machine.
C04707/2	-214748.3647	Unit/s	214748.3647	DIS:dnTeachSpeed • Display of the *dnSpeed_s* input signal in the real unit of the machine.
C04707/1	-214748.3647	unit/s^2	214748.3647	DIS:dnTeachAcc • Display of the *dnAcc_x* input signal in the real unit of the machine.
C04707/1	-214748.3647	s	214748.3647	DIS:dnTeachJerk • Display of the *dwJerk* input signal in the real unit of the machine.
C04708	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.
C04710/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)				Positioning mode • For absolute positioning, the home position must be known. • Not all positioning modes are available in every traversing range! ▸ Positioning modes (📖 409)
	1	Absolute		
	2	Absolute TP		
	5	Relative		
	6	Relative TP		
	7	Speed		
	8	Speed TP		
	11	Absolute CW (modulo)		
	12	Absolute CW TP (modulo)		
	13	Absolute CCW (modulo)		
	14	Absolute CCW TP (modulo)		
	15	Absolute ShortestWay (modulo)		
	16	Absolute ShortestWay TP (modulo)		
C04711/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)				Position from VTPOS • Reference to VTPOS table position to define the profile parameter "Position".
	1	VTPOS position 1 (C04701/1)		
	2	VTPOS position 2 (C04701/2)		
	...	...		
	75	VTPOS position 75 (C04701/75)		
C04712/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)				Speed from VTSPEED • Reference to VTSPEED table position to define the profile parameter "Speed".
	1	VTSPEED position 1 (C04702/1)		
	2	VTSPEED position 2 (C04702/2)		
	...	...		
	50	VTSPEED position 50 (C04702/50)		
C04713/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)				Acceleration from VTACC • Reference to VTACC table position to define the profile parameter "Acceleration".
	1	VTACC position 1 (C04703/1)		
	2	VTACC position 2 (C04703/2)		
	...	...		
	50	VTACC position 50 (C04703/50)		

Parameter	Possible settings		Information
C04714/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)			Deceleration from VTACC • Reference to VTACC table position to define the profile parameter "Deceleration".
	1	VTACC position 1 (C04703/1)	
	2	VTACC position 2 (C04703/2)	
	...	...	
	50	VTACC position 50 (C04703/50)	
C04715/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)			S-ramp time from VTJERK • Reference to VTJERK table position to define the profile parameter "S-ramp time".
	1	VTJERK position 1 (C04704/1)	
	2	VTJERK position 2 (C04704/2)	
	...	...	
	50	VTJERK position 50 (C04704/50)	
C04716/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)			Final speed from VTSPEED • Reference to VTSPEED table position to define the profile parameter "Final speed". • A sequence profile must be available.
	1	VTSPEED position 1 (C04702/1)	
	2	VTSPEED position 2 (C04702/2)	
	...	...	
	50	VTSPEED position 50 (C04702/50)	
C04717/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)			TP window starting position from VTPOS • Reference to VTPOS table position to define the profile parameter "TP window starting position". • With a position value "0" the starting position equals the negative travel range limit.
	1	VTPOS position 1 (C04701/1)	
	2	VTPOS position 2 (C04701/2)	
	...	...	
	75	VTPOS position 75 (C04701/75)	
C04718/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)			TP window end position from VTPOS • Reference to VTPOS table position to define the profile parameter "TP window end position". • With a position value "0" the end position equals the positive travel range limit.
	1	VTPOS position 1 (C04701/1)	
	2	VTPOS position 2 (C04701/2)	
	...	...	
	75	VTPOS position 75 (C04701/75)	
C04719/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)			Sequence profile with TP • Sequence profile for touch probe residual path positioning.
	1	Profile no. 1	
	...	...	
	75	Profile no. 75	
C04720/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)			Sequence profile without TP • Sequence profile for profile linkage.
	0	No sequence profile (end of a profile chaining)	
	1	Profile no. 1	
	...	...	
	75	Profile no. 75	

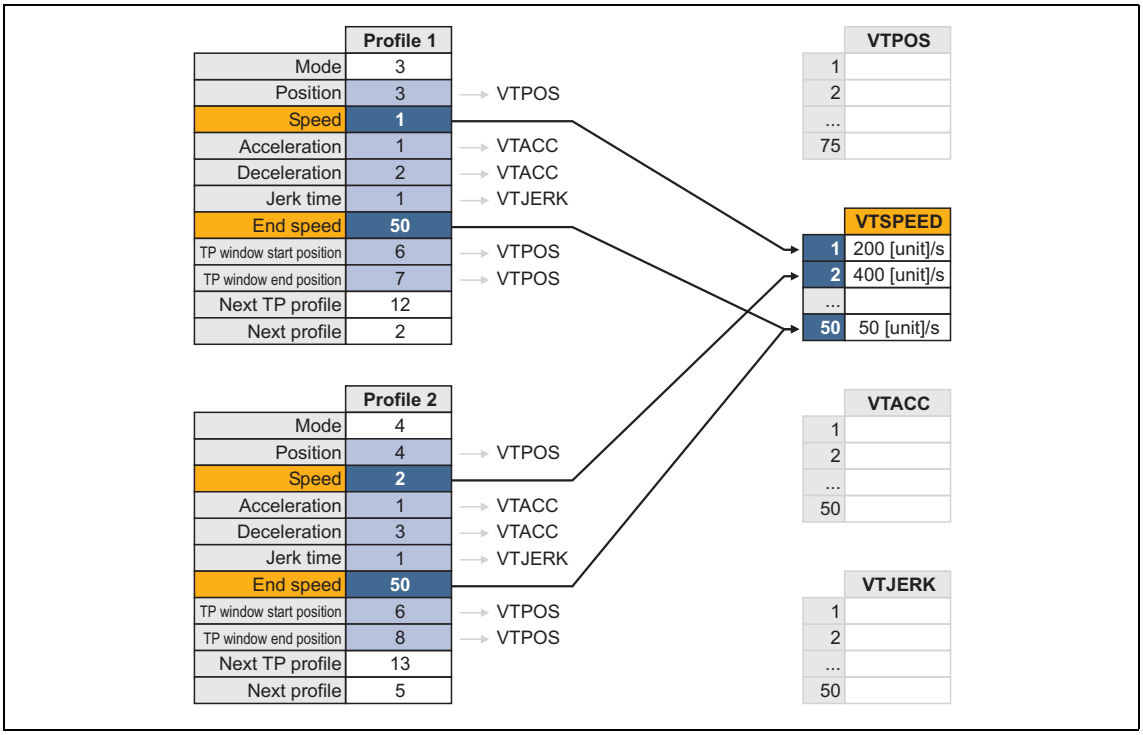
Parameter	Possible settings		Information
C04721/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75)	Value is bit-coded:		Touch probe configuration • The touch probe channels to be used are selected by setting the corresponding bits to "1".
	bits0	TP channel 1, rising edge	Digital input 1
	bits1	TP channel 1, falling edge	
	bit 2	TP channel 2, rising edge	Digital input 2
	bits3	TP channel 2, falling edge	
	bits4	TP channel 3, rising edge	Digital input 3
	bits5	TP channel 3, falling edge	
	bits6	TP channel 4, rising edge	Digital input 4
	bits7	TP channel 4, falling edge	
	bits8	TP channel 5, rising edge	Digital input 5
	bits9	TP channel 5, falling edge	
	bits10	TP channel 6, rising edge	Digital input 6
	bits11	TP channel 6, falling edge	
	bits12	TP channel 7, rising edge	Digital input 7
	bits13	TP channel 7, falling edge	
	bits14	TP channel 8, rising edge	Digital input 8
	bits15	TP channel 8, falling edge	
	bit16	TP channel 9	Motor encoder zero pulse
	bits18	TP channel 10	Load encoder zero pulse
	bits20	TP channel 11	DFIN zero pulse
	bits22	TP channel 12	DFOUT zero pulse
C04723/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75) From library V02.05.xx.xx			S-ramp mode • Will be supported as of controller software version V5.0. With software versions < V5.0, this parameter has no function.
	0	Second S-ramp time is not active. • The S-ramp time selected in C04715/1...75 from VTJERK is used for the acceleration and deceleration phase.	
	1	Second S-ramp time is active for deceleration. • The S-ramp time selected in C04715/1...75 from VTJERK is only used for the acceleration phase. • For the deceleration phase, however, the second S-ramp time selected in C04724/1...75 from VTJERK is used.	
C04724/1...75 (subcode 1 ... 75 ≡ profile no. 1 ... 75) From library V02.05.xx.xx			Second S-ramp time from VTJERK • Will be supported as of controller software version V5.0. With software versions < V5.0, this parameter has no function. • Reference to VTJERK table position to define the profile parameter "Second S-ramp time". • The second S-ramp time is only used for S-ramp mode "1" for the deceleration phase.
	1	VTJERK position 1 (C04704/1)	
	2	VTJERK position 2 (C04704/2)	
	...	...	
	50	VTJERK position 50 (C04704/50)	

5.111.1 Variable tables

To simplify parameter handling, the four most important physical sizes for profile parameters are stored in separate "variable tables".

	VTPos Positions	**VTSpeed** Speeds	**VTAcc** Acceleration/ deceleration	**VTJerk** S-ramp times
Unit	Unit	Unit/s	unit/s^2	s
Data format	DINT with four decimal positions			
Memory locations	75	50	50	50
Code	C04701/1...75	C04702/1...50	C04703/1...50	C04704/1...50
For profile parameters	Target position TP window starting position TP window end position	Speed Final speed	Acceleration Delay	S-ramp time

A value is assigned to a profile parameter by a reference to a table position of the assigned variable table. So not the value, but the table position which contains the value to be used is entered in the profile parameter.



[5-66] Principle: References to variable tables (here: References to VTSPEED)

- In the case of several references to the same table position, a change of the value in this table position affects several profiles at the same time. Thus, recurring profile parameters only need to be changed on one position.
- If e.g. in the case of a profile chaining, several profiles are to be executed with the same speed, the corresponding profile parameters "speed" can all refer to the same table position.

Conversion of the values into internal values

The values of the variable tables VTPOS, VTSPEED, VTACC and VTJERK are always entered in the application unit [unit].

- Since the SB **LS_Positioner** needs the profile parameters in the internal measuring system, the FB **L_PosPositionerTable** also stores the values of the variable tables in this measuring system.
- The values must only be reconverted by the FB if the machine parameters transmitted by the drive interface via the input *AxisData* have changed.
  - In this case the output *bBusy* is set to TRUE during the conversion and an access to the profile parameters is not possible.

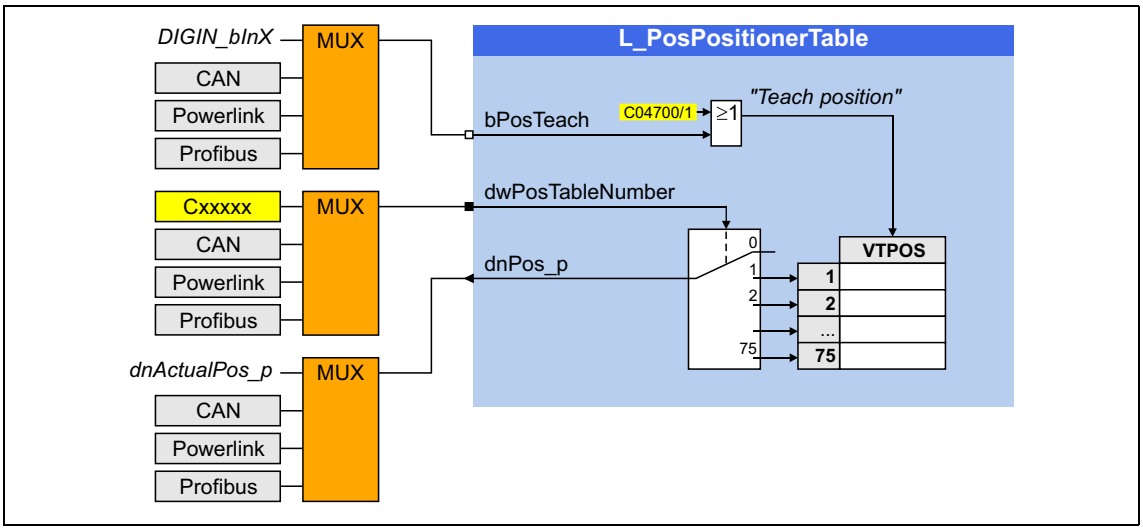
Note!

The values to be "taught" are exceptions. They must be selected in the internal measuring system.

5.111.2 "Teach" function

In addition to the direct value entry via parameters, the values can also be stored in the variable tables by means of the "teach" function.

The "teaching" of a value can be activated via an FB input or by describing a parameter (e.g. by master control):



[5-67] Principle of the "teach" function (here: "teaching" of positions)

Note!

If a position, speed, or acceleration are read out of the variable tables (e.g. with the FBs L_PosGetTablePos, L_PosGetTableSpeed or L_PosGetTableAcc) and this value is taught again into the FB, the value displayed may deviate in application units.

The deviation in the display can be explained by the conversion of the value from application units into internal units and back again to application units, but has no negative effect on the drive behaviour.

5.111.3 Positioning modes

Depending on the traversing range/application, you can select between different positioning modes which are described in the below table.

Note!

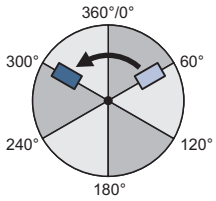
For absolute positioning, the home position must be known!

- If an absolute positioning process (positioning modes 1 ... 2 and 11 ... 16) is started although the home position is not known, an error message occurs.
  - In this case a programming error has occurred and the program flow must be reset.
  - If the error only occurs in a sequence profile, the last valid deceleration is used to decelerate the drive to standstill.

The modulo positioning modes may involve reversing processes!

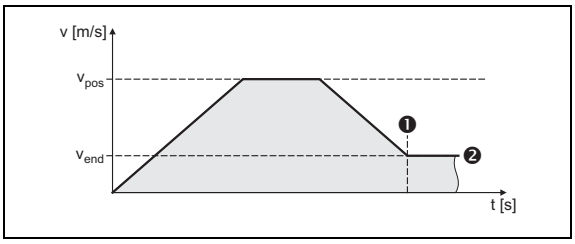
- If a modulo positioning process (positioning modes 11 ... 16) is started at an initial speed other than zero, the target position may be overtravelled and a reversing process may follow, depending on the selected deceleration / S-ramp time parameters. This would also cause e.g. counter-clockwise rotation in the "Absolute CW (modulo)" positioning mode.

Positioning mode		Supported traversing range		
		Unlimited	Limited	Modulo
1	**Absolute** The axis travels to an absolute position. • Reference for the absolute position is zero position. • The home position must be known. • The traversing range is limited: • to 214748.3647 [unit] • by the internal display area ($\pm 2^{31}$ increments)	●	●	
2	**Absolute TP** Like mode 1, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 412)	●	●	
5	**Relative** The axis is traversed by a distance. • Reference for the distance is the target position of the profile executed before. • The feed per positioning is limited: • to 214748.3647 [unit] • by the internal display area ($\pm 2^{31}$ increments)	●	●	●
6	**Relative TP** Like mode 5, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 412)	●	●	●
7	**Speed** Continuous constant travel. • This mode does not approach a defined position, but follows the profile. • Acceleration and deceleration are based on profile values. • The traversing direction is defined by the sign of the traversing speed. • Stopped through break signal.	●	●	
8	**Speed TP** Like mode 7, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 412)	●	●	

Positioning mode		Supported traversing range		
		Unlimited	Limited	Modulo
11	**Absolute CW (modulo)** The axis travels in CW direction to an absolute position. • Reference for the absolute position is zero position. • In this direction the zero position of the axis can be overtravelled.			●
12	**Absolute CW TP (modulo)** Like mode 11, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 412)			●
13	**Absolute CCW (modulo)** The axis travels in CCW direction to an absolute position. • Reference for the absolute position is zero position. • In this direction the zero position of the axis can be overtravelled.			●
14	**Absolute CCW TP (modulo)** Like mode 13, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 412)			●
15	**Absolute ShortestWay (modulo)** The axis travels to an absolute position in best time. • Reference for the absolute position is zero position. • The rotary table positioning is basically an absolute positioning with target positions between 0 and 360 angular degree [°]. In this mode the zero point can also be overtravelled if this is the shortest way to the target position:  			●
16	**Absolute ShortestWay TP (modulo)** Like mode 15, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 412)			●

5.111.4 Positioning with final speed

If a positioning process with a final speed other than zero is carried out, a velocity changeover / overchange can be realised, i.e. a second positioning process is started immediately once the target position is reached, and the drive does not come to a standstill at the first target position.



❶ Target position
❷ Final speed (in our example: other than zero)

[5-68] The principle of velocity changeover / overchange

- Normally, the drive has reached the final speed when the target position is reached.
- The profile is run through which is defined via "Profile data sequence profile without TP".
- The final speed must not exceed the maximum profile speed.
  - The speed is internally limited to the profile speed.
  - This is the maximum value for the final speed if the profile speed is reduced via override. (This behaviour may add to the cases described in the note below.)
  - The speed limitation is not affected by subsequent increases in the override value.

Note!

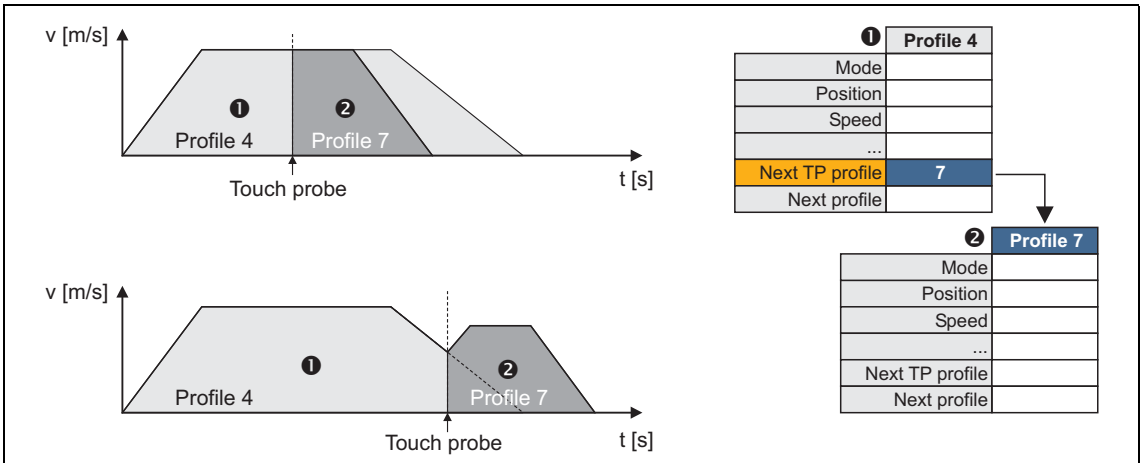
Positioning processes with final speeds other than zero do not permit reversing. In this case, the positioning process is aborted and an error is issued.

In the case of positioning processes with final speeds and if S-shaped ramps are used, a plausibility check as to whether the final speed can be reached is carried out when the profile is started. If the final speed cannot be reached, the positioning process is aborted and an error is issued. This is often the case if the positioning process has been paused and is then to be continued.

5.111.5 Touch probe positioning

With touch probe positioning, the profile is first processed according to the set profile parameters. If a touch probe is detected during profile processing, the profile automatically changes to the profile specified under the profile parameter "TP sequence profile".

- Here the current actual position is stored at the time of the touch probe activation (by a touch probe sensor).
- In the following relative positioning process, the "residual path" to this stored position is travelled according to the increments.



❶ "Starting profile"
❷ "Residual path profile" that is defined in the profile parameter "TP sequence profile" of the starting profile.

[5-69] Examples for a "residual path positioning" after a touch probe is detected

- The profile parameters "TP window starting position" and "TP window end position" the range in which touch probes are to be detected can be restricted.
  - If both profile parameters = "0", touch probe detection will be active for the whole profile/the whole traversing range.
- If no touch probe is detected and after the profile is executed, the positioning is continued with the profile defined in the profile parameter "Sequence profile without TP" (profile chaining).

Note!

If a profile is travelled with high speed and touch probe positioning is started, the residual path of which is smaller than the result from current speed and set deceleration ramp, the target position is "overtravelled".

- Normally a reversing movement occurs, i.e. the drive returns.
- If, for instance, a CCW rotation of the drive is forbidden by the safety module, the target cannot be approached since in this case the reversing movement is not permissible.

Further constellations are possible in connection with profile chaining in which an approach of the target position is impossible.

Touch probe configuration

The touch probe channel for positioning with touch probe detection can be selected in the »Engineer« via the parameter dialog of the FB **L_PositionerTable**.

- Go to the parameter dialog of the FB **L_PositionerTable** and press the **Setting up touch probe** button to open the *Setting up touch probe* dialog box and select the desired setting for the corresponding profile.
- The setting selected in this parameterisation dialog has a direct effect on the setting of C04721/1...75 ("touch probe configuration") and vice versa.
- For a direct setting of C04721/1...75 (e.g. via keypad) the following table lists the corresponding decimal values for all possible configurations.

Selection Touch probe channel	Touch probe response		
	Positive edge	Negative edge	Both edges
DIGIN 1	1	2	3
DIGIN 2	4	8	12
DIGIN 3	16	32	48
DIGIN 4	64	128	192
DIGIN 5	256	512	768
DIGIN 6	1024	2048	3072
DIGIN 7	4096	8192	12288
DIGIN 8	16384	32768	49152
Motor encoder zero pulse	65536		
Load encoder zero pulse	262144		
DFIN zero pulse	1048576		
DFOUT zero pulse	4194304		

- Example: For the selection of the touch probe channel "DIGIN 8" and the response to only one positive edge, the decimal value "16384" must be set in C04721.

5.111.6 Monitoring for counter overflow

This function extension will be available from library V2.02!

If the machine parameters transferred to the FB via the *AxisData* input are changed afterwards (e.g. the gearbox factors), this may cause a counter overflow when the speed, acceleration, and position values defined via parameters are converted from the real unit to the internal unit.

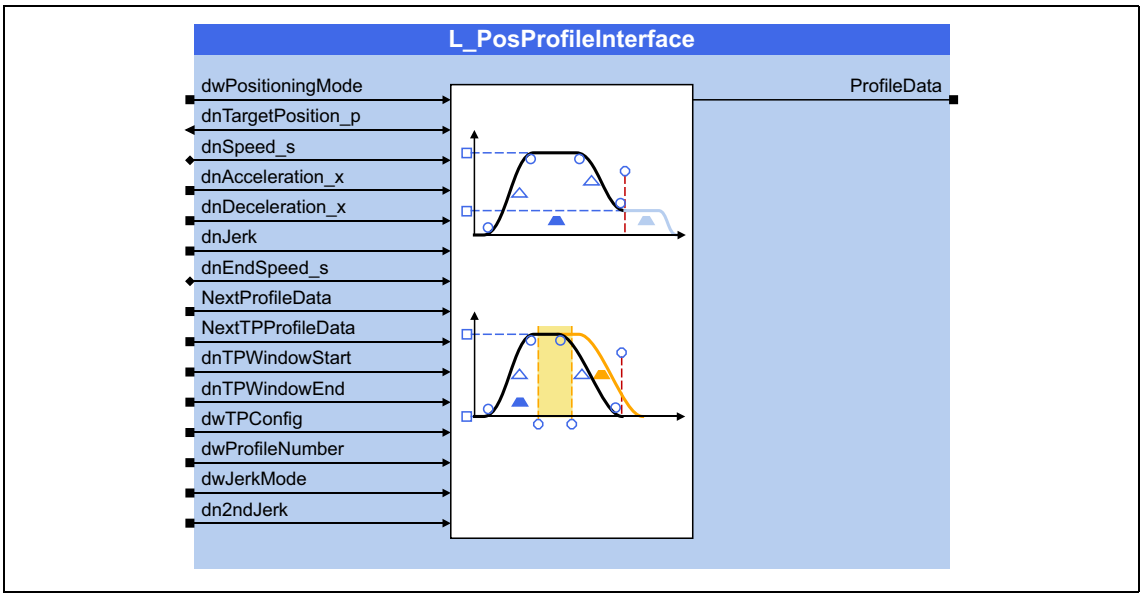
The FB causes a corresponding error message in case of an internal counter overflow due to a changed measuring system:

Error number		Error message in the logbook	Response
61736961	0x3AE0801	L_PosPositionerTable:int.pos. overflow (LS_DriveInterface)	Error
61737217	0x3AE0901	L_PosPositionerTable:int.pos. overflow (L_SdSetAxisData)	Error
61735937	0x3AE0401	L_PosPositionerTable:int.speed overflow (LS_DriveInterface)	Error
61736193	0x3AE0501	L_PosPositionerTable:int.speed overflow (L_SdSetAxisData)	Error
61735425	0x3AE0201	L_PosPositionerTable:int.accel. overflow (LS_DriveInterface)	Error
61735681	0x3AE0301	L_PosPositionerTable:int.accel. overflow (L_SdSetAxisData)	Error

5.112 L_PosProfileInterface - profile data interface

Function library:	LenzePositioning
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB provides a profile data record for the SB **LS_Positioner**.



Inputs

Identifier/data type	Information/possible settings	
dwPositioningMode DWORD	Positioning mode • For absolute positioning, the home position must be known! • Not all positioning modes are available in every traversing range! ▸ Positioning modes (📖 416)	
	1	Absolute
	2	Absolute TP
	5	Relative
	6	Relative TP
	7	Speed
	8	Speed TP
	11	Absolute CW (modulo)
	12	Absolute CW TP (modulo)
	13	Absolute CCW (modulo)
	14	Absolute CCW TP (modulo)
	15	Absolute ShortestWay (modulo)
	16	Absolute ShortestWay TP (modulo)
dnTargetPosition_p DINT	Target position in [inc]	
dnSpeed_s DINT	Speed • Entry as speed • Scaling: $2^{26} \equiv 15000$ [rpm]	

Identifier/data type	Information/possible settings		
dnAcceleration_x DINT	Acceleration • Entry as speed variation/time • Scaling: $2^{22} \equiv 15000000$ [rpm/s]		
dnDeceleration_x DINT	Delay • Entry as speed variation/time • Scaling: $2^{22} \equiv 15000000$ [rpm/s]		
dnJerk DINT	S-ramp time in [ms]		
dnEndSpeed_s DINT	Final speed • Entry as speed • Scaling: $2^{26} \equiv 15000$ [rpm]		
NextProfileData ProfileData	Profile data of the sequence profile without TP (for profile chaining) • This input can be connected e.g. with the output *ProfileData* of another instance of the FB **L_PosProfileInterface**.		
NextTPProfileData ProfileData	Profile data of the sequence profile with TP (for touch probe residual path positioning) • This input can be connected e.g. with the output *ProfileData* of another instance of the FB **L_PosProfileInterface**.		
dnTPWindowStart DINT	Area start for touch probe detection		
dnTPWindowEnd DINT	Area end for touch probe detection		
dwTPConfig DWORD	Touch-probe configuration (bit coded) • The touch probe channels to be used are selected by setting the corresponding bits to "1":		
	bits0	TP channel 1, rising edge	Digital input 1
	bits1	TP channel 1, falling edge	
	bits2	TP channel 2, rising edge	Digital input 2
	bits3	TP channel 2, falling edge	
	bits4	TP channel 3, rising edge	Digital input 3
	bits5	TP channel 3, falling edge	
	bits6	TP channel 4, rising edge	Digital input 4
	bits7	TP channel 4, falling edge	
	bits8	TP channel 5, rising edge	Digital input 5
	bits9	TP channel 5, falling edge	
	bits10	TP channel 6, rising edge	Digital input 6
	bits11	TP channel 6, falling edge	
	bits12	TP channel 7, rising edge	Digital input 7
	bits13	TP channel 7, falling edge	
	bits14	TP channel 8, rising edge	Digital input 8
	bits15	TP channel 8, falling edge	
	bit16	TP channel 9	Motor encoder zero pulse
	bits18	TP channel 10	Load encoder zero pulse
	bits20	TP channel 11	DFIN zero pulse
	bits22	TP channel 12	DFOUT zero pulse
dwProfileNumber DINT	Number assigned to the profile. • This number is displayed at the SB **LS_Positioner**.		

Identifier/data type	Information/possible settings	
dwJerkMode DWORD From library V02.05.xx.xx	S-ramp mode • Will be supported as of controller software version V5.0. With software versions < V5.0, this input has no function.	
	0	Second S-ramp time is not active. • The *dnJerk* S-ramp time is used for the acceleration and deceleration phase.
	1	Second S-ramp time is active for deceleration. • The *dnJerk* S-ramp time is only used for the acceleration phase. • For the deceleration phase, however, the second *dn2ndJerk* S-ramp time is used.
dn2ndJerk DINT From library V02.05.xx.xx	Second S-ramp time in [ms] • Will be supported as of controller software version V5.0. With software versions < V5.0, this input has no function. • The second S-ramp time is only used for S-ramp mode "1" for the deceleration phase.	

Outputs

Identifier/data type	Value/meaning
ProfileData	Profile data for the SB **LS_Positioner**

5.112.1 Positioning modes

Depending on the traversing range/application, you can select between different positioning modes which are described in the below table.

Note!

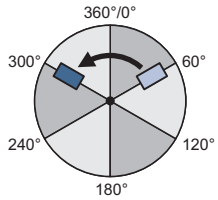
For absolute positioning, the home position must be known!

- If an absolute positioning process (positioning modes 1 ... 2 and 11 ... 16) is started although the home position is not known, an error message occurs.
  - In this case a programming error has occurred and the program flow must be reset.
  - If the error only occurs in a sequence profile, the last valid deceleration is used to decelerate the drive to standstill.

The modulo positioning modes may involve reversing processes!

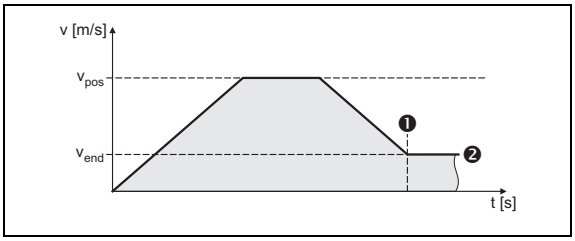
- If a modulo positioning process (positioning modes 11 ... 16) is started at an initial speed other than zero, the target position may be overtravelled and a reversing process may follow, depending on the selected deceleration / S-ramp time parameters. This would also cause e.g. counter-clockwise rotation in the "Absolute CW (modulo)" positioning mode.

Positioning mode		Supported traversing range		
		Unlimited	Limited	Modulo
1	**Absolute** The axis travels to an absolute position. • Reference for the absolute position is zero position. • The home position must be known. • The traversing range is limited: • to 214748.3647 [unit] • by the internal display area (±2^{31} increments)	●	●	
2	**Absolute TP** Like mode 1, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 419)	●	●	
5	**Relative** The axis is traversed by a distance. • Reference for the distance is the target position of the profile executed before. • The feed per positioning is limited: • to 214748.3647 [unit] • by the internal display area (±2^{31} increments)	●	●	●
6	**Relative TP** Like mode 5, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 419)	●	●	●
7	**Speed** Continuous constant travel. • This mode does not approach a defined position, but follows the profile. • Acceleration and deceleration are based on profile values. • The traversing direction is defined by the sign of the traversing speed. • Stopped through break signal.	●	●	
8	**Speed TP** Like mode 7, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 419)	●	●	
11	**Absolute CW (modulo)** The axis travels in CW direction to an absolute position. • Reference for the absolute position is zero position. • In this direction the zero position of the axis can be overtravelled.			●
12	**Absolute CW TP (modulo)** Like mode 11, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 419)			●
13	**Absolute CCW (modulo)** The axis travels in CCW direction to an absolute position. • Reference for the absolute position is zero position. • In this direction the zero position of the axis can be overtravelled.			●
14	**Absolute CCW TP (modulo)** Like mode 13, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 419)			●

Positioning mode		Supported traversing range		
		Unlimited	Limited	Modulo
15	**Absolute ShortestWay (modulo)** The axis travels to an absolute position in best time. • Reference for the absolute position is zero position. • The rotary table positioning is basically an absolute positioning with target positions between 0 and 360 angular degree [°]. In this mode the zero point can also be overtravelled if this is the shortest way to the target position:  			●
16	**Absolute ShortestWay TP (modulo)** Like mode 15, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 419)			●

5.112.2 Positioning with final speed

If a positioning process with a final speed other than zero is carried out, a velocity changeover / overchange can be realised, i.e. a second positioning process is started immediately once the target position is reached, and the drive does not come to a standstill at the first target position.



❶ Target position
❷ Final speed (in our example: other than zero)

[5-70] The principle of velocity changeover / overchange

- Normally, the drive has reached the final speed when the target position is reached.
- The profile is run through which is defined via "Profile data sequence profile without TP".
- The final speed must not exceed the maximum profile speed.
  - The speed is internally limited to the profile speed.
  - This is the maximum value for the final speed if the profile speed is reduced via override. (This behaviour may add to the cases described in the note below.)
  - The speed limitation is not affected by subsequent increases in the override value.

Note!

Positioning processes with final speeds other than zero do not permit reversing. In this case, the positioning process is aborted and an error is issued.

In the case of positioning processes with final speeds and if S-shaped ramps are used, a plausibility check as to whether the final speed can be reached is carried out when the profile is started. If the final speed cannot be reached, the positioning process is aborted and an error is issued. This is often the case if the positioning process has been paused and is then to be continued.

5.112.3 Touch probe positioning

With touch probe positioning, the profile is first processed according to the set profile parameters. If a touch probe is detected during profile processing, the profile automatically changes to the "sequence profile with TP", which has been defined via the *NextTPProfileData* input.

- If no touch probe is detected, positioning is continued with the sequence profile which has been defined via the *NextProfileData* input (profile linkage).
- In order to define the sequence profiles, both inputs *NextTPProfileData* and *NextProfileData* can be connected, for instance, with the *ProfileData* output of another instance of the FB **L_PosProfileInterface**.

Note!

If a profile is travelled with high speed and touch probe positioning is started, the residual path of which is smaller than the result from current speed and set deceleration ramp, the target position is "overtravelled".

- Normally a reversing movement occurs, i.e. the drive returns.
- If, for instance, a CCW rotation of the drive is forbidden by the safety module, the target cannot be approached since in this case the reversing movement is not permissible.

Further constellations are possible in connection with profile chaining in which an approach of the target position is impossible.

5.112.4 Touch probe configuration

The touch probe channel to be used for positioning with touch probe detection is selected bit-coded via the *dwTPConfig* input. The following table contains all possible configurations with the corresponding decimal values:

Selection Touch probe channel	Touch probe response		
	Positive edge	Negative edge	Both edges
DIGIN 1	1	2	3
DIGIN 2	4	8	12
DIGIN 3	16	32	48
DIGIN 4	64	128	192
DIGIN 5	256	512	768
DIGIN 6	1024	2048	3072
DIGIN 7	4096	8192	12288
DIGIN 8	16384	32768	49152
Motor encoder zero pulse	65536		
Load encoder zero pulse	262144		
DFIN zero pulse	1048576		
DFOUT zero pulse	4194304		

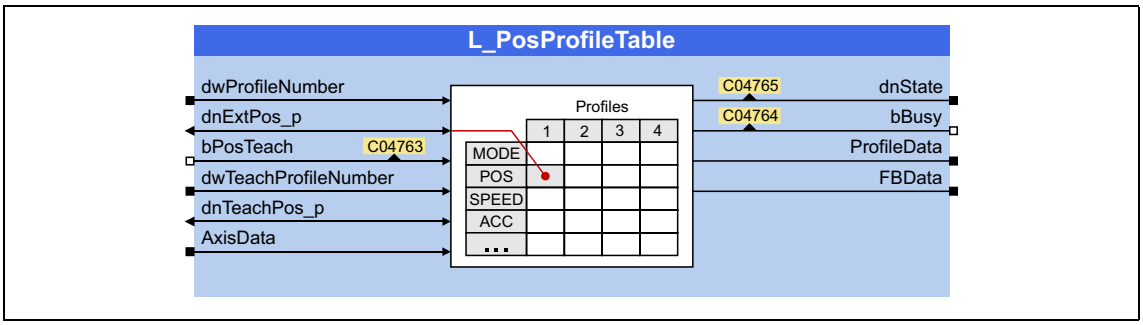
- Example: For the selection of the touch probe channel "DIGIN 8" and the response to only one positive edge, the decimal value "16384" must be set at the *dwTPConfig* input.

5.113 L_PosProfileTable - profile data record management (simple)

Function library:	LenzePositioning
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB is used to store and manage up to four traversing profiles and to "teach" target positions.

- A profile describes a motion request which can be implemented into a rotary motion by the SB **LS_Positioner**.
- A profile is described via the following profile parameters: Mode (type of positioning), position, speed, acceleration, deceleration, S-ramp time, TP sequence profile and TP selection.
- Unlike the FB L_PosPositionerTable, this FB does not use any variable tables, but profile parameter data is directly entered under the assigned codes.
- Furthermore, the position assigned to the input *dnExtPos_p* will be used as target position if profile number 1 is selected.
- Another important task of this FB is the conversion of the table values according to the scaling selected in the SB **LS_DriveInterface**.



Inputs

Identifier/data type	Information/possible settings	
dwProfileNumber DWORD	Profile selection • Number of the profile the profile parameters of which are to be transferred to the SB **LS_Positioner** via the output *ProfileData*.	
	1...4	Profile number
dnExtPos_p DINT	External target position in [inc] for profile number 1	
bPosTeach BOOL	"Teach" position	
	TRUE	The position assigned to the input *dnTeachPos_p* is stored in the profile selected via the input *dwTeachProfileNumber*.
	TRUE↘FALSE	The value of the last cycle is maintained in the table.
dwTeachProfileNumber DWORD	Selection of the profile for the position to be "taught"	
	0	No selection (selection in case the input is not assigned)
	1	Profile 1 • The position selected via the input *dnExtPos_p* will be used as target position for profile number 1.
	2...4	Profiles 2 ... 4

Identifier/data type	Information/possible settings
dnTeachPos_p DINT	Position to be "taught" in [inc]
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits15	Error is active.
	bits16	Calculation is active.
	bits17	Position taught.
	bits28	Error: Invalid machine parameters (*AxisData*).
	bits29	Error: Invalid profile number (*dwProfileNumber*).
	bits30	Error: Invalid teach profile number.
bBusy BOOL	Status signal "Conversion of the profile parameter is active"	
	FALSE	Profile parameters are ready.
	TRUE	Conversion is active, no access possible to the profile parameters at the moment.
ProfileData	Interface to basic function "Positioning" • Data structure with the profile parameters of the profile selected via the input *dwProfileNumber*. • Connect this output with the input of the same name of the SB **LS_Positioner**.	
FBData	FB interface for internal profiles/table data (L_PosGetProfile, L_PosGetTableAcc, L_PosGetTableJerk, L_PosGetTablePos, L_PosGetTableSpeed) • Data structure with the internal instance data.	

Parameter

Parameter	Possible settings			Information
C04750/1...4 (Subcode 1 ... 4 ≡ profile no. 1 ... 4)				Positioning mode • For absolute positioning, the home position must be known. • Not all positioning modes are available in every traversing range! ▸ Positioning modes (📖 426)
	1	Absolute		
	2	Absolute TP		
	5	Relative		
	6	Relative TP		
	7	Speed		
	8	Speed TP		
	11	Absolute CW (modulo)		
	12	Absolute CW TP (modulo)		
	13	Absolute CCW (modulo)		
	14	Absolute CCW TP (modulo)		
	15	Absolute ShortestWay (modulo)		
	16	Absolute ShortestWay TP (modulo)		
C04751/1...4 (Subcode 1 ... 4 ≡ profile no. 1 ... 4)	-214748.3647	Unit	214748.3647	Position

Parameter	Possible settings			Information
C04752/1...4 (Subcode 1 ... 4 ≡ profile no. 1 ... 4)	-214748.3647	Unit/t	214748.3647	Speed
C04753/1...4 (Subcode 1 ... 4 ≡ profile no. 1 ... 4)	0.0000	Unit/t^2	214748.3647	Acceleration
C04754/1...4 (Subcode 1 ... 4 ≡ profile no. 1 ... 4)	0.0000	Unit/t^2	214748.3647	Delay
C04755/1...4 (Subcode 1 ... 4 ≡ profile no. 1 ... 4)	0.000	s	2147483.647	S-ramp time
C04756/1...4 (Subcode 1 ... 4 ≡ profile no. 1 ... 4)				TP sequence profile • Sequence profile for touch probe residual path positioning.
	0	No residual path positioning		
	1	Profile no. 1		
	...	...		
	4	Profile no. 4		
C04757/1...4 (Subcode 1 ... 4 ≡ profile no. 1 ... 4)	Value is bit-coded:			Touch probe configuration • The touch probe channels to be used are selected by setting the corresponding bits to "1".
	bits0	TP channel 1, rising edge		Digital input 1
	bits1	TP channel 1, falling edge		
	bit 2	TP channel 2, rising edge		Digital input 2
	bits3	TP channel 2, falling edge		
	bits4	TP channel 3, rising edge		Digital input 3
	bits5	TP channel 3, falling edge		
	bits6	TP channel 4, rising edge		Digital input 4
	bits7	TP channel 4, falling edge		
	bits8	TP channel 5, rising edge		Digital input 5
	bits9	TP channel 5, falling edge		
	bits10	TP channel 6, rising edge		Digital input 6
	bits11	TP channel 6, falling edge		
	bits12	TP channel 7, rising edge		Digital input 7
	bits13	TP channel 7, falling edge		
	bits14	TP channel 8, rising edge		Digital input 8
	bits15	TP channel 8, falling edge		
	bit16	TP channel 9		Motor encoder zero pulse
	bits18	TP channel 10		Load encoder zero pulse
	bits20	TP channel 11		DFIN zero pulse
	bits22	TP channel 12		DFOUT zero pulse
C04758				"Teach" position • For execution via parameter channel.
	0↗1	The position assigned to the input *dnTeachPos_p* is stored in the profile selected via the input *dwTeachProfileNumber*.		
C04759/1	String of digits			Position unit • Read only
C04759/2	String of digits			Speed unit • Read only
C04759/3	String of digits			Acceleration unit • Read only

Parameter	Possible settings			Information
C04760/1	-214748.3647	Unit	214748.3647	External target position • Display of the *dnExtPos_p* input signal in the real unit of the machine.
C04760/2	-214748.3647	Unit	214748.3647	Teach position • Display of the *dnTeachPos_p* input signal in the real unit of the machine.
C04761/1	0		4	Selected profile • Display of the *dwProfileNumber* input signal.
C04761/2	0		4	Selected teach profile • Display of the *dwTeachProfileNumber* input signal.
C04763				DIS:bPosTeach • Display of the *bPosTeach* input signal.
	0	FALSE		
	1	TRUE		
C04764				DIS:bBusy • Display of the output signal *bBusy*.
	0	Profile parameters are ready.		
	1	Conversion is active, no access possible to the profile parameters at the moment.		
C04765	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.
C04767/1...4 (Subcode 1 ... 4 ≡ profile no. 1 ... 4) From library V02.05.xx.xx				S-ramp mode • Will be supported as of controller software version V5.0. With software versions < V5.0, this parameter has no function.
	0	Second S-ramp time is not active. • The S-ramp time parameterised in C04755/1...4 is used for the acceleration and deceleration phase.		
	1	Second S-ramp time is active for deceleration. • The S-ramp time parameterised in C04755/1...4 is only used for the acceleration phase. • For the deceleration phase, however, the second S-ramp time parameterised in C04768/1...4 is used.		
C04768/1...4 (Subcode 1 ... 4 ≡ profile no. 1 ... 4) From library V02.05.xx.xx	0.000		s	2147483.647	Second S-ramp time • Will be supported as of controller software version V5.0. With software versions < V5.0, this parameter has no function. • The second S-ramp time is only used for S-ramp mode "1" for the deceleration phase.

5.113.1 Profile data table

Profile parameter data is directly entered under the assigned codes.

- Exception: The position assigned to the input *dnExtPos_p* will be used as target position if profile number 1 is selected.

Profile parameters	Unit	Profile no. 1	Profile no. 2	Profile no. 3	Profile no. 4
Positioning mode	-	C04750/1	C04750/2	C04750/3	C04750/4
Position	Unit	*dnExtPos_p*	C04751/2	C04751/3	C04751/4
Speed	Unit/t	C04752/1	C04752/2	C04752/3	C04752/4
Acceleration	Unit/t^2	C04753/1	C04753/2	C04753/3	C04753/4
Delay	Unit/t^2	C04754/1	C04754/2	C04754/3	C04754/4
S-ramp time	s	C04755/1	C04755/2	C04755/3	C04755/4
TP sequence profile	-	C04756/1	C04756/2	C04756/3	C04756/4
TP selection	-	C04757/1	C04757/2	C04757/3	C04757/4

Conversion of the values into internal values

The profile parameters are always entered in the application unit [unit].

- Since the SB **LS_Positioner** needs the profile parameters in the internal measuring system, the FB **L_PosProfileTable** also stores the values in this measuring system.
- The values must only be reconverted by the FB if the machine parameters transmitted by the drive interface via the input *AxisData* have changed.
  - In this case the output *bBusy* is set to TRUE during the conversion and an access to the profile parameters is not possible.



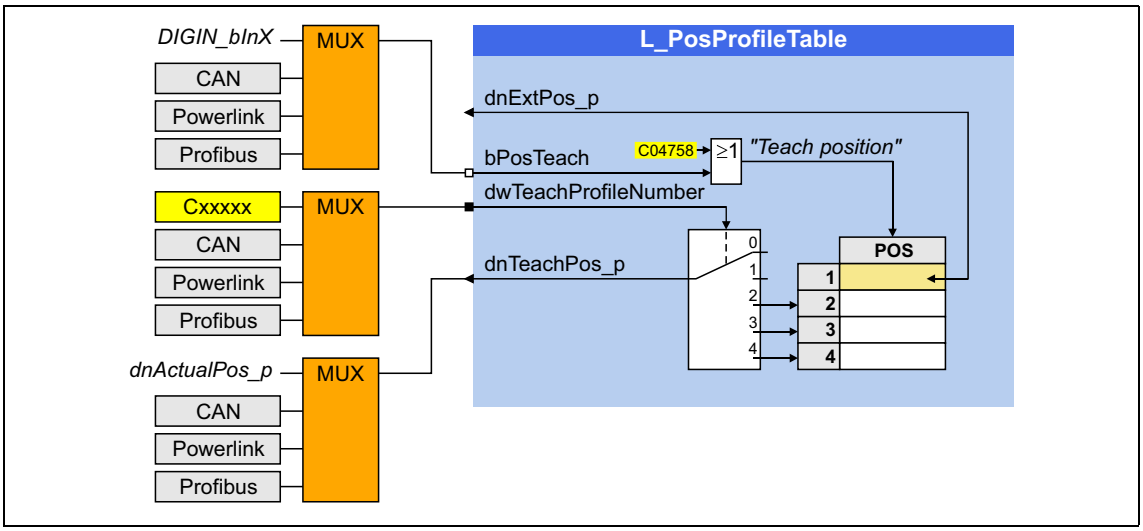
Note!

The values to be "taught" are exceptions. They must be selected in the internal measuring system.

5.113.2 "Teach" function

In addition to the direct value entry via parameters, the target position for a profile can also be stored by means of the "teach" function.

The "teaching" of a value can be activated via an FB input or by describing a parameter (e.g. by master control):



[5-71] "Teach" function principle

Note!

If a position is read out of the FB (e.g. with the FB L_PosGetTablePos) and this value is taught again to the FB, the value displayed may deviate in application units.

The deviation in the display can be explained by the conversion of the value from application units into internal units and back again to application units, but has no negative effect on the drive behaviour.

5.113.3 Positioning modes

Depending on the traversing range/application, you can select between different positioning modes which are described in the below table.

Note!

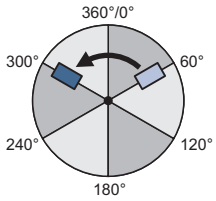
For absolute positioning, the home position must be known!

- If an absolute positioning process (positioning modes 1 ... 2 and 11 ... 16) is started although the home position is not known, an error message occurs.
  - In this case a programming error has occurred and the program flow must be reset.
  - If the error only occurs in a sequence profile, the last valid deceleration is used to decelerate the drive to standstill.

The modulo positioning modes may involve reversing processes!

- If a modulo positioning process (positioning modes 11 ... 16) is started at an initial speed other than zero, the target position may be overtravelled and a reversing process may follow, depending on the selected deceleration / S-ramp time parameters. This would also cause e.g. counter-clockwise rotation in the "Absolute CW (modulo)" positioning mode.

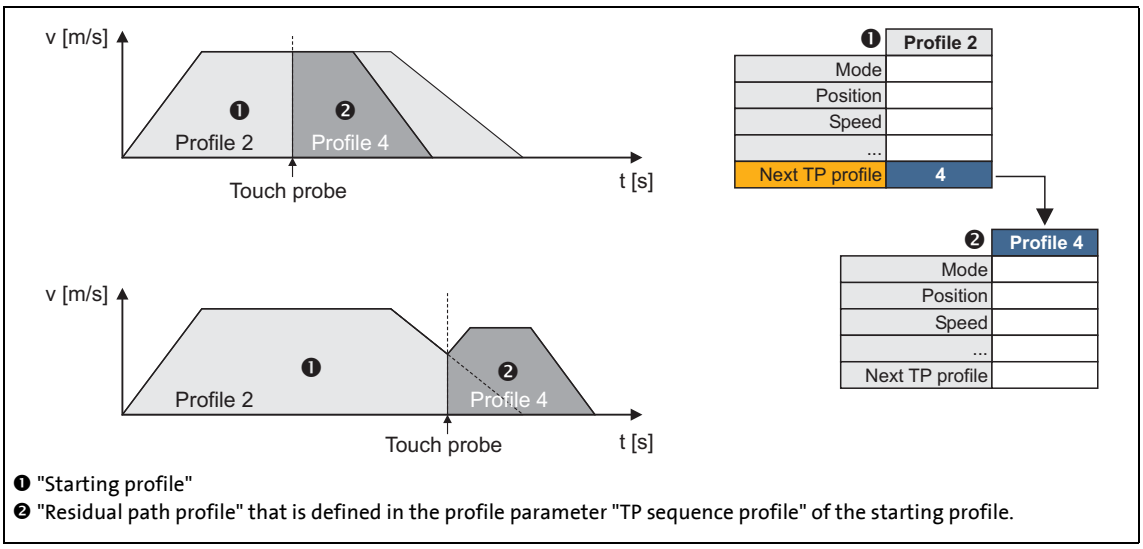
Positioning mode		Supported traversing range		
		Unlimited	Limited	Modulo
1	**Absolute** The axis travels to an absolute position. • Reference for the absolute position is zero position. • The home position must be known. • The traversing range is limited: • to 214748.3647 [unit] • by the internal display area ($\pm 2^{31}$ increments)	●	●	
2	**Absolute TP** Like mode 1, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 428)	●	●	
5	**Relative** The axis is traversed by a distance. • Reference for the distance is the target position of the profile executed before. • The feed per positioning is limited: • to 214748.3647 [unit] • by the internal display area ($\pm 2^{31}$ increments)	●	●	●
6	**Relative TP** Like mode 5, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 428)	●	●	●
7	**Speed** Continuous constant travel. • This mode does not approach a defined position, but follows the profile. • Acceleration and deceleration are based on profile values. • The traversing direction is defined by the sign of the traversing speed. • Stopped through break signal.	●	●	
8	**Speed TP** Like mode 7, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 428)	●	●	

Positioning mode		Supported traversing range		
		Unlimited	Limited	Modulo
11	**Absolute CW (modulo)** The axis travels in CW direction to an absolute position. • Reference for the absolute position is zero position. • In this direction the zero position of the axis can be overtravelled.			●
12	**Absolute CW TP (modulo)** Like mode 11, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 428)			●
13	**Absolute CCW (modulo)** The axis travels in CCW direction to an absolute position. • Reference for the absolute position is zero position. • In this direction the zero position of the axis can be overtravelled.			●
14	**Absolute CCW TP (modulo)** Like mode 13, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 428)			●
15	**Absolute ShortestWay (modulo)** The axis travels to an absolute position in best time. • Reference for the absolute position is zero position. • The rotary table positioning is basically an absolute positioning with target positions between 0 and 360 angular degree [°]. In this mode the zero point can also be overtravelled if this is the shortest way to the target position:  			●
16	**Absolute ShortestWay TP (modulo)** Like mode 15, but with profile change on touch probe detection. ▸ Touch probe positioning. (📖 428)			●

5.113.4 Touch probe positioning

With touch probe positioning, the profile is first processed according to the set profile parameters. If a touch probe is detected during profile processing, the profile automatically changes to the profile specified under the profile parameter "TP sequence profile".

- Here the current actual position is stored at the time of the touch probe activation (by a touch probe sensor).
- In the following relative positioning process, the "residual path" to this stored position is travelled according to the increments.



❶ "Starting profile"
❷ "Residual path profile" that is defined in the profile parameter "TP sequence profile" of the starting profile.

[5-72] Examples for a "residual path positioning" after a touch probe is detected

- If no touch probe is detected, positioning stops after profile processing without profile linkage.

Note!

If a profile is travelled with high speed and touch probe positioning is started, the residual path of which is smaller than the result from current speed and set deceleration ramp, the target position is "overtravelled".

- Normally a reversing movement occurs, i.e. the drive returns.
- If, for instance, a CCW rotation of the drive is forbidden by the safety module, the target cannot be approached since in this case the reversing movement is not permissible.

Further constellations are possible in connection with profile chaining in which an approach of the target position is impossible.

5.113.5 Monitoring for counter overflow

This function extension will be available from library V2.02!

If the machine parameters transferred to the FB via the *AxisData* input are changed afterwards (e.g. the gearbox factors), this may cause a counter overflow when the speed, acceleration, and position values defined via parameters are converted from the real unit to the internal unit.

The FB causes a corresponding error message in case of an internal counter overflow due to a changed measuring system:

Error number		Error message in the logbook	Response
61736966	0x3AE0806	L_PosProfileTable:int.pos. overflow (LS_DriveInterface)	Error
61737222	0x3AE0906	L_PosProfileTable:int.pos. overflow (L_SdSetAxisData)	Error
61735942	0x3AE0406	L_PosProfileTable:int.speed overflow (LS_DriveInterface)	Error
61736198	0x3AE0506	L_PosProfileTable:int.speed overflow (L_SdSetAxisData)	Error
61735430	0x3AE0206	L_PosProfileTable:int.accel. overflow (LS_DriveInterface)	Error
61735686	0x3AE0306	L_PosProfileTable:int.accel. overflow (L_SdSetAxisData)	Error

5.114 L_PosSequencer - sequence control

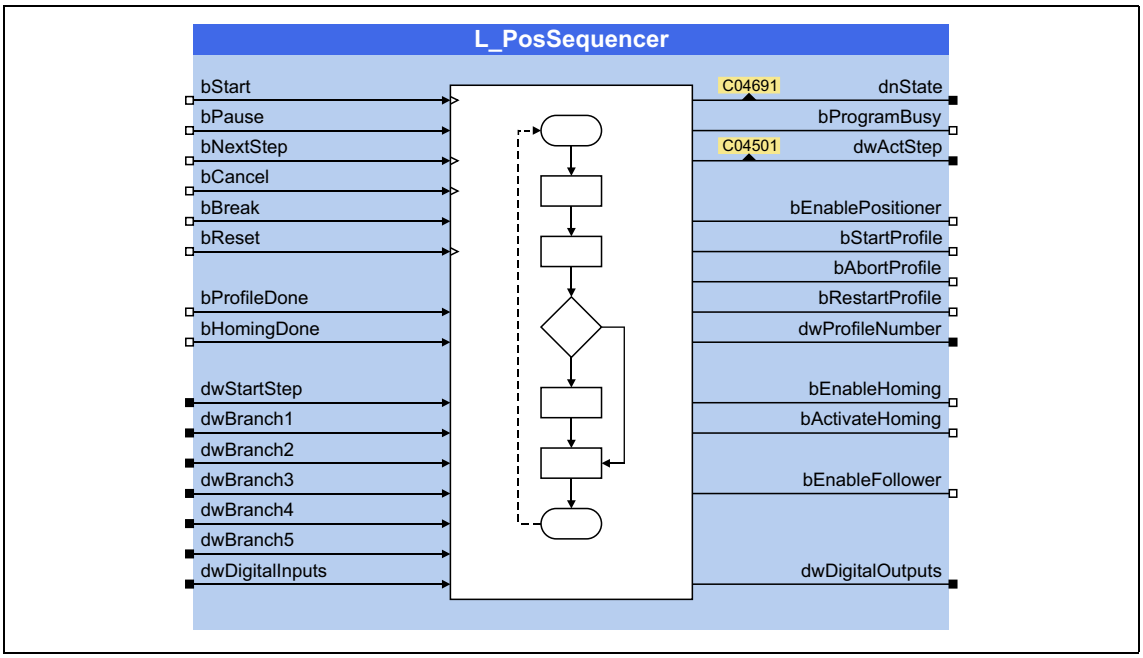
Function library:	LenzePositioning	
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

This FB processes a positioning program based on a sequence table.

- The sequence table is an important part of this FB and is indicated by a code with 100 subcodes. Each subcode contains a reference to a single program segment, also known as "action".
- Different action types are available which serve to implement, for instance, program branching, switching operations, waiting times and counters. The actions themselves are also indicated by codes.
  - The following action types are passive: Branching, variable branching, switching, counter setting, counting, waiting, standby, and program end. When these action types are processed, the deceleration ramp is active and a brake, if available, will be applied.
  - The action types positioning and homing are active since these both action types trigger the execution of the corresponding basic function via control outputs.

Note!

For a complete sequence control, the FB **L_PosSequencer** must be connected to further blocks, like the FB **L_PosPositionerTable** (profile data record management) and the SB **LS_Positioner** (basic function "positioning").



Inputs

Identifier/data type	Information/possible settings	
bStart BOOL	Start/continue positioning program • Display parameter: C04690/2	
	FALSE↗TRUE	The positioning program is started/continued. • Counters and outputs are not automatically reset by this. • A positioning program interrupted by *bBreak* is continued with the sequence step. • The started positioning program is processed until the end, even if *bStart* is reset to FALSE.
bPause BOOL	Interrupt the current step of the positioning program • Display parameter: C04690/3	
	TRUE	• An active positioning is interrupted. • The drive is stopped with the current profile deceleration. • A deceleration override is taken into account. • The sequence of an active timing element is stopped. • A "Standby" action is not interrupted, i.e. the output *bEnableFollower* remains TRUE.
	FALSE	• The interrupted positioning is completed. • The distance of the profile already covered is considered here. • Positioning processes with final speeds are not continued if the final speed cannot be reached anymore. • The deceleration of the stopped timing element is continued with the residual time.
bNextStep* BOOL From library V02.00.xx.xx	Cancel current step and go to next step	
	FALSE↗TRUE	The current positioning program step is cancelled immediately and the positioning program is continued with the next step listed in the sequence table.
bCancel BOOL	Cancel the current step and jump to the step defined in C04504 • Display parameter: C04690/6	
	FALSE↗TRUE	The current step of the positioning program is cancelled immediately and the positioning program is continued with the step defined in C04504.
bBreak BOOL	Interrupt positioning program • This input is intended for a linkage with drive-relevant status signals (e.g. for controller inhibit or quick stop). • If the interruption is cancelled again by a reset to FALSE, a new FALSE/TRUE edge is required at *bStart* so that the positioning program can be continued with the sequence step. • Display parameter: C04690/4	
	FALSE	Condition for the program start via *bStart*.
	TRUE	The active positioning program is interrupted. • If positioning is active, the drive is braked to standstill with the deceleration time for stop. • The system block **LS_Positioner** is deactivated. • The system block **LS_Homing** is deactivated. • The setpoint followers are deactivated. • The digital output signals keep their state.

Identifier/data type	Information/possible settings	
bReset BOOL	Reset positioning program. • Also possible when the positioning program is interrupted. • Display parameter: C04690/5	
	FALSE↗TRUE	The positioning program is reset. • If positioning is active, the drive is braked to standstill with the deceleration time for stop without considering an acceleration override. • The program flow is aborted ("program end"). • The digital output signals, counters and timing elements are reset. • In case a stand-by operation is active, it is aborted.
bProfileDone BOOL	Interface to basic function "Positioning" • Input for status signal "Profile has been executed". • Connect this input with the status output of the same name of the SB **LS_Positioner**. • Display parameter: C04690/7	
bHomingDone BOOL	Interface to basic function "Homing" • Input for status signal "Homing completed". • Connect this input with the status output of the same name of the SB **LS_Homing**. • Display parameter: C04690/1	
dwStartStep DWORD	Step of the sequence table with which the positioning program is to start. • Display parameter: C04692/6	
	0	Positioning program starts with step 1 of the sequence table.
	1 ... 100	Starting step
	Other	Positioning program does not start.
	Not assigned	Positioning program starts with step 1 of the sequence table.
dwBranch1 DWORD	Input signal for action 1 of type "Variable branch" • Display parameter: C04692/1 • If the positioning program contains the action 1 of type "Variable branch", a branch is carried out depending on the value applied to this input.	
	0	Branch to the next step.
	1	Branch to the step defined in C04540/1.
	2	Branch to the step defined in C04541/1.
	...	...
	20	Branch to the step defined in C04559/1.
	> 20	Branch to the next step.
dwBranch2 DWORD	Input signal for action 2 of type "Variable branch" • Display parameter: C04692/2 • If the positioning program contains the action 2 of type "Variable branch", a branch is carried out depending on the value applied to this input.	
	0	Branch to the next step.
	1	Branch to the step defined in C04540/2.
	2	Branch to the step defined in C04541/2.
	...	...
	20	Branch to the step defined in C04559/2.
	> 20	Branch to the next step.

Identifier/data type		Information/possible settings	
dwBranch3	DWORD	Input signal for action 3 of type "Variable branch" • Display parameter: C04692/3 • If the positioning program contains the action 3 of type "Variable branch", a branch is carried out depending on the value applied to this input.	
		0	Branch to the next step.
		1	Branch to the step defined in C04540/3.
		2	Branch to the step defined in C04541/3.
		...	...
		20	Branch to the step defined in C04559/3.
		> 20	Branch to the next step.
dwBranch4	DWORD	Input signal for action 4 of type "Variable branch" • Display parameter: C04692/4 • If the positioning program contains the action 4 of type "Variable branch", a branch is carried out depending on the value applied to this input.	
		0	Branch to the next step.
		1	Branch to the step defined in C04540/4.
		2	Branch to the step defined in C04541/4.
		...	...
		20	Branch to the step defined in C04559/4.
		> 20	Branch to the next step.
dwBranch5	DWORD	Input signal for action 5 of type "Variable branch" • Display parameter: C04692/5 • If the positioning program contains the action 5 of type "Variable branch", a branch is carried out depending on the value applied to this input.	
		0	Branch to the next step.
		1	Branch to the step defined in C04540/5.
		2	Branch to the step defined in C04541/5.
		...	...
		20	Branch to the step defined in C04559/5.
		> 20	Branch to the next step.
dwDigitalInputs	DWORD	Sequencer inputs 1 ... 32 (bit coded) • The action types "Positioning", Branching", "Waiting" and "Stand-by" have the parameter "Input for...". If this is non-zero, it describes the number of the sequencer input where the positioning program awaits the level defined before it executes the action. • For the transmission of individual control signals a BOOL-to-DWORD converter can be connected to this input. • Display parameter: C04693/1	
		bits0	Sequencer input 1
		bits1	Sequencer input 2
		...	...
		bits31	Sequencer input 32

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded) • Display parameter: C04691	
	bits15	Error is active.
	bits16	Positioning program ready to start/program end reached.
	bits17	Positioning program is running.
	bits18	Positioning program is started, pause is active.
	bits19	Positioning program is stopped.
	bits20	Positioning program is reset.
	bits21	Positioning program is completed.
	bits22	"Watchdog" monitoring function for positioning has been activated.
	bits25	Wrong value at branch input.
bProgramBusy BOOL	Status signal "Positioning program is running" • Display parameter: C04690/15	
	FALSE	Positioning program has not been started or stopped.
	TRUE	Positioning program is running.
dwActStep DWORD	Current step which is processed by the positioning program. • Display parameter: C04501	
	0	Positioning program has not been started.
	1 ... 100	Current step
bEnablePositioner BOOL	Interface to basic function "Positioning" • Control signal for enabling the basic function. • Connect this output with the input *bEnable* of the SB **LS_Positioner**. • Display parameter: C04690/10	
bStartProfile BOOL	Interface to basic function "Positioning" • Control signal for starting the positioning. • Connect this output with the input *bStart* of the SB **LS_Positioner**. • Display parameter: C04690/11	
bAbortProfile BOOL	Interface to basic function "Positioning" • Control signal for aborting the positioning. • Connect this output with the input *bAbort* of the SB **LS_Positioner**. • Display parameter: C04690/12	
bRestartProfile BOOL	Interface to basic function "Positioning" • Control signal for restarting the positioning. • Connect this output with the input *bRestart* of the SB **LS_Positioner**. • Display parameter: C04690/13	
dwProfileNumber DINT	Interface to profile data record management • Transmission of the profile number for selecting the next profile . • Connect this output with the input of the same name of the FB **L_PosPositionerTable**. • Display parameter: C04692/8	
bEnableHoming BOOL	Interface to basic function "Homing" • Control signal for enabling the basic function. • Connect this output with the input *bEnable* of the SB **LS_Homing**. • Display parameter: C04690/8	
bActivateHoming BOOL	Interface to basic function "Homing" • Control signal for starting homing. • This control signal is set to TRUE one cycle later than the *bEnableHoming* control signal. • Connect this output with the input of the same name of the SB **LS_Homing**. • Display parameter: C04690/9	

Identifier/data type	Value/meaning	
bEnableFollower BOOL	Enable signal for setpoint follower • This output is intended for the enable of a setpoint follower during a stand-by program step. • Connect this output, if required, with the input *bEnable* of the corresponding set-point follower (SB **LS_SpeedFollower**, SB **LS_TorqueFollower** or SB **LS_PositionFollower**). • Display parameter: C04690/10	
	FALSE	No enable since the basic function "Positioning" is active.
	TRUE	Enable is possible since stand-by program step is active.
dwDigitalOutputs DWORD	Sequencer outputs 1 ... 32 (bit coded) • The sequencer outputs can be set to "0" or "1" by means of the "Switching" action type during the runtime of the positioning program. • For the transmission of individual control signals a DWORD-to-BOOL converter can be connected to this input. • Display parameter: C04693/2	
	bits0	Sequencer output 1
	bits1	Sequencer output 2
	...	...
	bits31	Sequencer output 32

General parameters

Note!

The parameters of the different actions for the sequence tables are described in the corresponding action subchapters.

Parameter	Possible settings		Information
C04500/1...100			Sequence table • The calls for the actions required for the positioning program are stored under the subcodes. • This is how the basic sequence (except for the branches) is described.
	0	Program end	
	xxyyy	Action call • xx ≡ action type • yyy ≡ action number	
C04501	1 ...	... 100	Current program step • Read only
C04502			Status of sequence control • Read only
	0	READY: Positioning program is ready to start/program end has been reached.	
	1	RUN: Positioning program is running.	
	2	PAUSE: Current step is interrupted, pause is active.	
	3	BREAK: Positioning program is interrupted.	
	4	RESET: Positioning program has been reset.	
C04503			Sequence control • Bit-coded code for controlling the sequencer by a PC or master control via parameter channel. • By setting a bit to "1" the corresponding function is activated.
	bits0	Start	
	bits1	Pause	
	bit 2	BREAK	
	bits3	Cancel	
	bits4	Reset	
	bits5	NextStep	
C04504			Program step at abort • This step is processed when the input *bCancel* is set to TRUE.
	0	**Sequence step**	
	1...100	Step 1 ... 100	
	101	Program end	
C04505	Text		Current comment • Read only

5.114.1 Priorities of the control signals

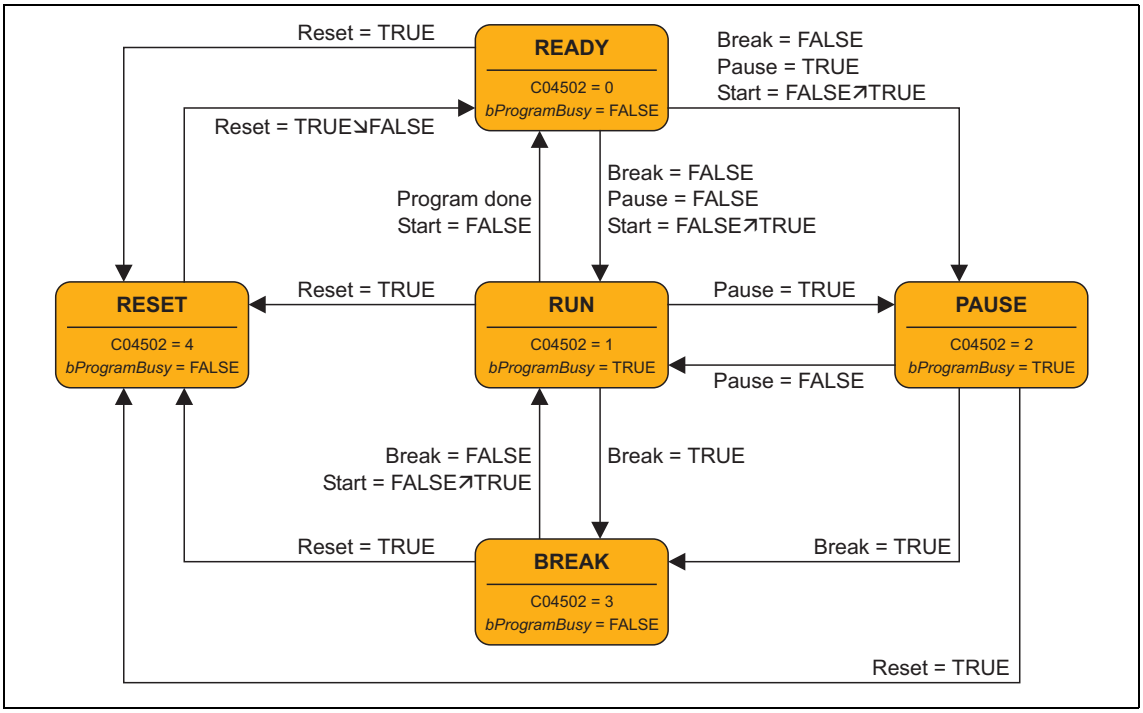
The boolean control signals are evaluated in the following sequence:

1. *bReset* 2. *bBreak* 3. *bPause* 4. *bStart* 5. *bCancel*

5.114.2 States of the sequencer

The internal state machine of the sequencer distinguishes the following five states:

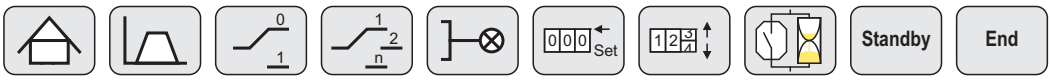
Status	Display C04502	Information
Ready	0	Positioning program ready to start/program end reached. • If the "Break" function is not activated, the positioning program can be started by e.g. a FALSE/TRUE edge at *bStart*.
RUN	1	Positioning program is running. • The sequence table is processed.
PAUSE	2	Positioning program is started, pause has been activated. • An active positioning is interrupted. • The drive is stopped with the current profile deceleration. • An acceleration override is not considered. • The sequence of an active timing element is stopped. • After the pause is cancelled, the positioning program and a positioning activated before or a stopped timing element are continued at the same position.
BREAK	3	Positioning program is interrupted. • If positioning is active, the drive is braked to standstill with the deceleration time for stop. • The system block **LS_Positioner** is deactivated. • The digital output signals keep their state. • After the interruption is cancelled, a new start signal is required to continue the positioning program with the sequence step.
Reset	4	Positioning program is reset. • If positioning is active, the drive is braked to standstill with the deceleration time for stop without considering an acceleration override. • The program flow is aborted ("program end"). • The digital output signals, counters and timing elements are reset. • In case a stand-by operation is active, it is aborted.



[5-73] Internal state machine of the sequencer

5.114.3 Action types available for creating the positioning program

The program flow of the multi-purpose positioning is selected according to a sequence table which can contain up to 100 references to "actions".



- An action comprises a clear functionality which is described with a few parameters.
- Different action types are available which serve to implement, for instance, program branching, switching operations, waiting times and counters.
- Each action type provides a certain number of actions which can be parameterised individually.
- One action can be called from several positions in the sequence table.
- After an action has been processed, the action in the next step of the sequence table is automatically processed unless it is jumped to another step in the sequence table due to a branch.
- One action can be maximally processed per computing cycle.
- The sequence table and the actions are displayed by parameters (codes with subcodes).
- Action calls are entered into the sequence table using a decimal number which contains the action type [A] and the action number [B].



Overview

Action	Action type [A]	Action number [B]	Number of available actions	Parameter range (first instance)
Program end	0	000	1	-
Positioning	1	001 ... 050	50	C04510 ... C04516
Switching	2	001 ... 025	25	C04520 ... C04523
Branching	3	001 ... 025	25	C04530 ... C04532
Variable branching	4	001 ... 005	5	C04540 ... C04559
Homing	5	000	1	-
Waiting	7	001 ... 025	25	C04570 ... C04573
Counter setting	8	001 ... 010	10	C04580 ... C04582
Counting	9	001 ... 025	25	C04590 ... C04593
Stand-by	10	001 ... 005	5	C04600 ... C04602

Detailed information of the action types can be found in the following subchapters.

5.114.3.1 Action type "Program end"



To define the program end in the sequence table, the action of type "Program end" is available.

Action call (entry in the sequence table)

0 (Leading zeros can be left out).

- If the input *bStart* is reset to FALSE while the positioning program is running, processing is only continued until the program end has been reached.
- In the Lenze setting all entries of the sequence table are set to "0" and thus to "program end".

5.114.3.2 Action type "Positioning"



To execute a profile, 50 actions of "Positioning" type are available.

Action call (entry in the sequence table)

1	x	x	x

with action number xxx = 001 ... 050

Waiting function

The execution of the set profile with activated waiting function is only started if the bit 1 ... 32 of the input signal *dwDigitalInputs* selected for the waiting function adopts the selected signal state ("0" or "1").

"Watchdog" monitoring function

You can select an individual "watchdog time" for every "positioning" action.

- If positioning takes longer than the selected watchdog time, there will be a branch to the next watchdog step.
- If the watchdog time is set to 0 s, the monitoring function is deactivated (Lenze setting).

Parameter

The available actions 1 ... 50 are indicated via the subcodes 1 ... 50.

Parameter	Possible settings		Information
C04510/1...50	Text		Comment on positioning step • Option for description of the function for keypad/HMI.
C04511/1...50			Input for waiting function • With the default setting "0" the waiting function is skipped and the profile execution is started immediately.
	0	**Waiting function is deactivated.**	
	1	Sequencer input 1 (Bit 0 of *dwDigitalInputs*)	
	2	Sequencer input 2 (Bit 1 of *dwDigitalInputs*)	
	...	...	
	32	Sequencer input 32 (Bit 31 of *dwDigitalInputs*)	
C04512/1...50			Signal state for waiting function • Only when the input for the waiting function has adopted this state, the profile is executed.
	0	Profile start when bit state is "0"	
	1	**Profile start when bit state is "1"**	
C04513/1...50			Profile number • A sequence profile can be set in the corresponding profile parameter.
	0	**No profile execution.**	
	1...100	Execute profile no. 1 ... 100.	
C04514/1...50			Sequence step • Step inside the sequence table which will be processed after the profile has been executed.
	0	**Sequence step**	
	1...100	Step 1 ... 100	
C04515/1...50			Watchdog time • Monitoring time within which positioning must be completed. Otherwise, there will be a branch to the next watchdog step defined under C04516.
	0	**Monitoring function is deactivated.**	
	> 0	Monitoring time in [s]	
C04516/1...50			Watchdog sequence step • Executed when the watchdog time is exceeded.
	0	**Sequence step**	
	1...100	Step 1 ... 100	

5.114.3.3 Action type "Switching"



In order to switch digital output signals 25 actions of type "Switching" are available.

- Each action can set two selectable bits of the output signal *dwDigitalOutputs* to "0" or "1" independently of each other.

Action call (entry in the sequence table)

2	x	x	x

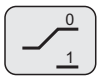
with action number xxx = 001 ... 025

Parameter

The available actions 1 ... 25 are indicated via the subcodes 1 ... 25.

Parameter	Possible settings		Information
C04520/1...25			Output for A switching
	0	**Switching is deactivated.**	
	1	Sequencer output 1 (Bit 0 of *dwDigitalOutputs*)	
	2	Sequencer output 2 (Bit 1 of *dwDigitalOutputs*)	
	...	...	
	32	Sequencer output 32 (Bit 31 of *dwDigitalOutputs*)	
C04521/1...25			Signal state for A switching
	0	Set bit to "0".	
	1	**Set bit to "1".**	
C04522/1...25			Output for B switching
	0	**Switching is deactivated.**	
	1	Sequencer output 1 (Bit 0 of *dwDigitalOutputs*)	
	2	Sequencer output 2 (Bit 1 of *dwDigitalOutputs*)	
	...	...	
	32	Sequencer output 32 (Bit 31 of *dwDigitalOutputs*)	
C04523/1...25			Signal state for B switching
	0	Set bit to "0".	
	1	**Set bit to "1".**	

5.114.3.4 Action type "Branching"



25 actions of type "Branching" are available for conditional and unconditional branches (jumps).

- A branch to the indicated step is executed when the comparison is deactivated or the selected bit 1 ... 32 of the input signal *dwDigitalInputs* has the selected signal state at the time of processing ("0" or "1"). If not, the sequence step is processed in the sequence table.

Action call (entry in the sequence table)

3	x	x	x

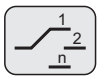
with action number xxx = 001 ... 025

Parameter

The available actions 1 ... 25 are indicated via the subcodes 1 ... 25.

Parameter	Possible settings		Information
C04530/1...25			Input for comparison
	0	**Unconditional branch** (The sequence step set in C04532/1...25 is processed).	
	1	Sequencer input 1 (Bit 0 of *dwDigitalInputs*)	
	2	Sequencer input 2 (Bit 1 of *dwDigitalInputs*)	
	...	...	
	32	Sequencer input 32 (Bit 31 of *dwDigitalInputs*)	
C04531/1...25			Comparison value • State which is compared with the bit to be evaluated of the input signal *dwDigitalInputs*.
	0	Comparison with "0"	
	1	**Comparison with "1"**	
C04532/1...25			Sequence step in case of equality • If the bit to be evaluated of the input signal *dwDigitalInputs* equals the comparison value, a branch to the step set here is executed. • In case of inequality or "0" setting, the next step in the sequence table is processed.
	0	**Sequence step**	
	1...100	Step 1 ... 100	

5.114.3.5 Action type "Variable branching"



For variable branches (jumps) 5 actions of type "Variable branching" are available.

- The branch to one of 20 possible steps is executed depending on the input signal *dwBranch1...5* at the time of processing.
- The input signals *dwBranch1...5* are firmly assigned to the five available actions
  - Input signal *dwBranch1* determines the branch for action 1,
  - Input signal *dwBranch2* determines the branch for action 2, etc.
- Example: When the input signal *dwBranch2* has the value "15" at the time of processing the action no. 2, it is branched to the step which is entered in the parameter "Sequence step at branch 15" for action no. 2 (C04554/2).

Action call (entry in the sequence table)

4	x	x	x

with action number xxx = 001 ... 005

Parameter

The available actions 1 ... 5 are indicated via the subcodes 1 ... 5.

Parameter	Possible settings		Information
C04540/1...5			Sequence step at branch value 1 • Step which is executed next when input *dwBranch1...5* = "1".
	0	**Branching is deactivated.** (The next step in the sequence table is processed.)	
	1...100	Step 1 ... 100	
C04541/1...5	See possible settings for C04540.		Sequence step at branch value 2
C04542/1...5			Sequence step at branch value 3
C04543/1...5			Sequence step at branch value 4
C04544/1...5			Sequence step at branch value 5
C04545/1...5			Sequence step at branch value 6
C04546/1...5			Sequence step at branch value 7
C04547/1...5			Sequence step at branch value 8
C04548/1...5			Sequence step at branch value 9
C04549/1...5			Sequence step at branch value 10
C04550/1...5			Sequence step at branch value 11
C04551/1...5			Sequence step at branch value 12
C04552/1...5			Sequence step at branch value 13
C04553/1...5			Sequence step at branch value 14
C04554/1...5			Sequence step at branch value 15
C04555/1...5			Sequence step at branch value 16
C04556/1...5			Sequence step at branch value 17
C04557/1...5			Sequence step at branch value 18
C04558/1...5			Sequence step at branch value 19
C04559/1...5			Sequence step at branch value 20

5.114.3.6 Action type "Homing"



In order to execute a homing function, the action of type "Homing" is available.

Action call (entry in the sequence table)

5	0	0	0

Note!

The "Homing" action has no own parameters.

The settings for homing are carried out via the parameters of the basic function "Homing".

Required connections

This functionality requires the following connections between the FB and the basic function "Homing":

- Output *bEnableHoming* → input *HM_bEnable* of the SB **LS_Homing**.
- Output *bActivateHoming* → input *HM_bActivateHoming* of the SB **LS_Homing**.
- Output *HM_bDone* of the SB **LS_Homing** → input *bHomingDone*.

Sequence

1. If the action is called, the SB **LS_Homing** is enabled via the *bEnableHoming* output.
2. One cycle later, the output *bActivateHoming* for activating homing is set to TRUE.
3. After the feedback via input *bHomingDone* that homing has been executed, the enable of the SB **LS_Homing** is deactivated again.
4. The program flow is continued.

5.114.3.7 Action type "Waiting"



25 actions of type "Waiting" are available for the insertion into the program flow.

- The sequence step is only processed after the waiting time has expired or when the selected bit 1 ... 32 of the input signal *dwDigitalInputs* has the selected signal state ("1" or "2") at the time of processing.

Waiting time C04571	Input C04572	Function
0 (deactivated)	0 (deactivated)	Sequence step is processed immediately (default setting).
> 0 [s]	0 (deactivated)	After the waiting time has expired, the sequence step will be processed.
> 0 [s]	1 ... 32	The sequence step will be processed when the input for the waiting time has the set state, but no later than after the set waiting time has elapsed.
0 (deactivated)	1 ... 32	If the input for the waiting function has the set state, the sequence step will be processed.

Action call (entry in the sequence table)

7	x	x	x

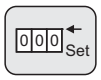
with action number xxx = 001 ... 025

Parameter

The available actions 1 ... 25 are indicated via the subcodes 1 ... 25.

Parameter	Possible settings		Information
C04570/1...25	Text		Comment on waiting step • Option for description of the function for keypad/HMI.
C04571/1...25	**0**	**Waiting time is deactivated.**	Waiting time
	> 0	Waiting time in [s]	
C04572/1...25			Input for waiting function
	0	**Input is deactivated.**	
	1	Sequencer input 1 (Bit 0 of *dwDigitalInputs*)	
	2	Sequencer input 2 (Bit 1 of *dwDigitalInputs*)	
	...	...	
	32	Sequencer input 32 (Bit 31 of *dwDigitalInputs*)	
C04573/1...50			Signal state for waiting function • Only when the input for the waiting function has adopted this state, the sequence step will be processed.
	0	Sequence step when bit state is "0"	
	1	**Sequence step when bit state is "1"**	

5.114.3.8 Action type "Counter setting"



10 actions of type "Counter setting" are available for setting one of the 10 available counters to a certain starting value.

- The 10 actions of type "Counter setting" are not permanently assigned to the 10 counters.
- You can use, for instance, an action of type "Counter setting" to set a counter to a value and later you can set the same counter with another action of type "Counter setting" to another value.

Action call (entry in the sequence table)

8	x	x	x

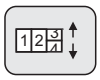
with action number xxx = 001 ... 010

Parameter

The available actions 1 ... 10 are indicated via the subcodes 1 ... 10.

Parameter	Possible settings			Information
C04580/1...10				Counter selection
	0	"Counter setting" is deactivated.		
	1	**Counter 1**		
	...	...		
	10	Counter 10		
C04581/1...10	-2147483647		2147483647	New counter content
C04582/1...10	-2147483647		2147483647	Current counter content • Read only

5.114.3.9 Action type "Counting"



25 actions of type "Counting" are available for counting processes.

- With each action processing the counter content of the respective counter is increased or reduced by the set step value (count upwards or downwards).
- When the comparison condition for comparing the counter content with an adjustable comparison value is fulfilled, a branch to any step is possible.
- 10 actions of type "Counter setting" are available for setting a counter to a starting value.

Action call (entry in the sequence table)

9	x	x	x

with action number xxx = 001 ... 025

Parameter

The available actions 1 ... 25 are indicated via the subcodes 1 ... 25.

Parameter	Possible settings			Information
C04590/1...25				Counter selection
	0	Counting is deactivated. (Sequence step is processed.)		
	1	**Counter 1**		
	...	...		
	10	Counter 10		
C04591/1...25	-2147483647		2147483647	Step value • Value by which the counter is increased or reduced. • Initialisation: 1
C04592/1...25	-2147483647		2147483647	Comparison value • Value with which the counter is compared.
C04593/1...25				Sequence step • When the set comparison condition is met, a branch to the step set here is executed. • If the condition is not fulfilled or with "0" setting, the next step in the sequence table is processed.
	0	**Sequence step**		
	1...100	Step 1 ... 100		
C04594/1...25				Comparison condition
	1	**Counter content = comparison value**		
	2	Counter content > comparison value		
	3	Counter content ≥ comparison value		
	4	Counter content < comparison value		
	5	Counter content ≤ comparison value		

5.114.3.10 Action type "Stand-by"



5 actions of type "Stand-by" are available for the temporary activation of a setpoint follower.

- For this purpose the output *bEnableFollower* must be connected to the enable input *bEnable* of the corresponding setpoint follower (SB **LS_SpeedFollower**, SB **LS_TorqueFollower** or SB **LS_PositionFollower**).
- The enable signal at the output *bEnableFollower* is applied until the condition for exiting the stand-by is met.

Action call (entry in the sequence table)

1	0	x	x	x

with action number xxx = 001 ... 005

Parameter

The available actions 1 ... 5 are indicated via the subcodes 1 ... 5.

Parameter	Possible settings		Information
C04600/1...5	Text		Comment on "Stand-by" step • Option for description of the function for keypad/HMI.
C04601/1...5			Input for "Stand-by" end
	1	Sequencer input 1 (Bit 0 of *dwDigitalInputs*)	
	2	Sequencer input 2 (Bit 1 of *dwDigitalInputs*)	
	...	...	
	32	Sequencer input 32 (Bit 31 of *dwDigitalInputs*)	
C04602/1...5			Signal state for "Stand-by" end • Only when the input for the end of "Stand-by" has reached this state, this stand-by step is exited and the sequence step is processed.
	0	Sequence step when bit state is "0"	
	1	**Sequence step when bit state is "1"**	

5.114.4 Example: Sequence table

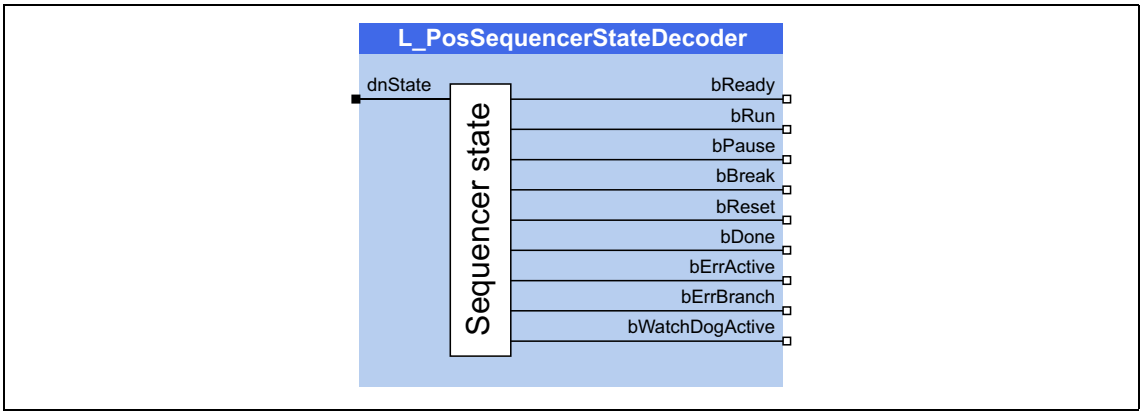
Parameter	Entry	≡ Action type/action no.		Meaning
C04500/1	5000		-	Execute homing
C04500/2	3001		1	Process action 1 of type "Branching" (3)
C04500/3	9005		5	Process action 5 of type "Counting" (9)
C04500/4	1001		1	Process action 1 of type "Positioning" (1)
C04500/5	7003		3	Process action 3 of type "Waiting" (7)
C04500/6	1002		2	Process action 2 of type "Positioning" (1)
C04500/7	3002		2	Process action 2 of type "Branching" (3)
C04500/8	9005		5	Process action 5 of type "Counting" (9)
C04500/9	3004		4	Process action 4 of type "Branching" (3)
C04500/...	yyxxx	yy	xxx	Process action *xxx* of type *yy*
C04500/24	1012		12	Process action 12 of type "Positioning" (1)
C04500/25	0		-	Program end
C04500/26	0		-	Program end
C04500/...	...	...	...	...
C04500/100	0		-	Program end

5.115 L_PosSequencerStateDecoder - status signals of the sequence control

Function library:	LenzePositioning	**FB is available as of library V02.00!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB decodes the status output signal *dnState* of FB LS_PosSequencer into individual boolean status signals for further use in the FB interconnection.

- The functionality corresponds to a converter "DINT➞BOOL", but the boolean outputs of this FB have the same names according to their meaning.



Inputs

Identifier/data type	Information/possible settings
dnState DINT	Input for the bit coded status signal of the sequence control. • Connect this input with the output of the same name of the FB L_PosSequencer. (📖 430)

Outputs

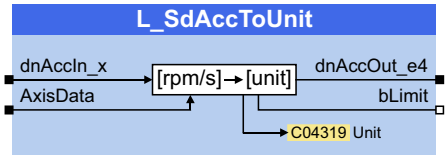
Bit*	Identifier	Meaning in case of TRUE
Status messages:		
16	bReady	Positioning program ready to start/program end reached.
17	bRun	Positioning program is running.
18	bPause	Positioning program is started, pause is active.
19	bBreak	Positioning program is stopped.
20	bReset	Positioning program is reset.
21	bDone	Positioning program is completed.
Error messages:		
15	bErrActive	Error is active.
22	bWatchDogActive	"Watchdog" monitoring function for positioning has been activated.
25	bErrBranch	Wrong value at branch input.

* Relating to the input signal *dnState*.

5.116 L_SdAccToUnit - acceleration conversion

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts an acceleration value defined in the internal unit [rpm/s] into an acceleration value in the real unit of the machine.



Inputs

Identifier/data type	Information/possible settings
dnAccIn_x DINT	Acceleration as speed variation/time in [rpm/s] • 15000000 rpm/s ≡ 2^{22} ≡ 4194304
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnAccOut_e4 DINT	Acceleration in [unit/s^2] • Output in "e4" format (fixed point with four decimal positions)	
bLimit BOOL	Status "Output signal is limited"	
	TRUE	The output signal is limited to the value range that can be displayed.

Parameter

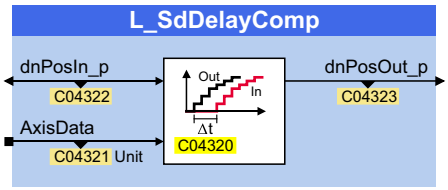
Parameter	Possible settings	Information
C04319	String from *AxisData*	Acceleration unit • Read only

5.117 L_SdDelayComp - dead time compensation

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB extrapolates a signal by a set time. In this way, the FB can compensate for the dead times of incoming signals resulting from the bus transfer.

- The FB considers a modulo measuring system and the cycle of the selected machine parameters.



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Position in [inc]
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning
dnPosOut_p DINT	Compensated position in [inc]

Parameter

Parameter	Possible settings			Information
C04320	-100.000	ms	100.000	Rate time • Initialisation: 0.000 ms
C04321 As of library V02.02.xx.xx	String of digits			Position unit • Read only
C04322 As of library V02.02.xx.xx	-214000.0000	Unit	214000.0000	Position at the input • Display of the *dnPosIn_p* input signal in the real unit of the machine.
C04323 As of library V02.02.xx.xx	-214000.0000	Unit	214000.0000	Position at the output • Display of the *dnPosOut_p* output signal in the real unit of the machine.

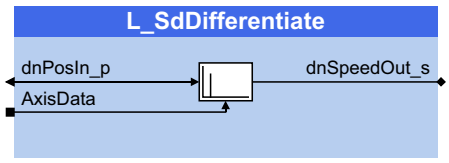
5.118 L_SdDifferentiate - differentiation

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB creates a speed signal from a position value.

Note!

As of library V02.00.xx.xx the FB is also suitable for clocked (modulo) positions!



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Position in [inc] • Scaling: 1 encoder revolution ≡ 2^{16} increments (or acc. to *AxisData*)
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

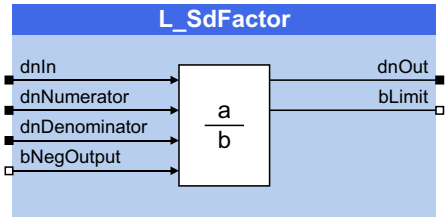
Identifier/data type	Value/meaning
dnSpeedOut_s DINT	Speed output in [rpm] • Scaling: 15000 rpm ≡ 2^{26} ≡ 67108864

5.119 L_SdFactor - stretch factor

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB provides the "stretch factor" function for the application "Electronic gearbox".

- The calculation is made with remainder handling.



Inputs

Identifier/data type	Information/possible settings	
dnIn DINT	Input signal	
dnNumerator DINT	Stretch factor (numerator) • 0 ... 2^{31}-1 • If an invalid value is selected, the gain will be set to "0".	
dnDenominator DINT	Stretch factor (denominator) • 1 ... 2^{31}-1 • If an invalid value is selected, the gain will be set to "0".	
bNegOutput BOOL	Reverse output signal	
	FALSE	Inversion is deactivated.
	TRUE	The output signal is inverted to the input signal if valid values are pending at the inputs *dnNumerator* and *dnDenominator*.

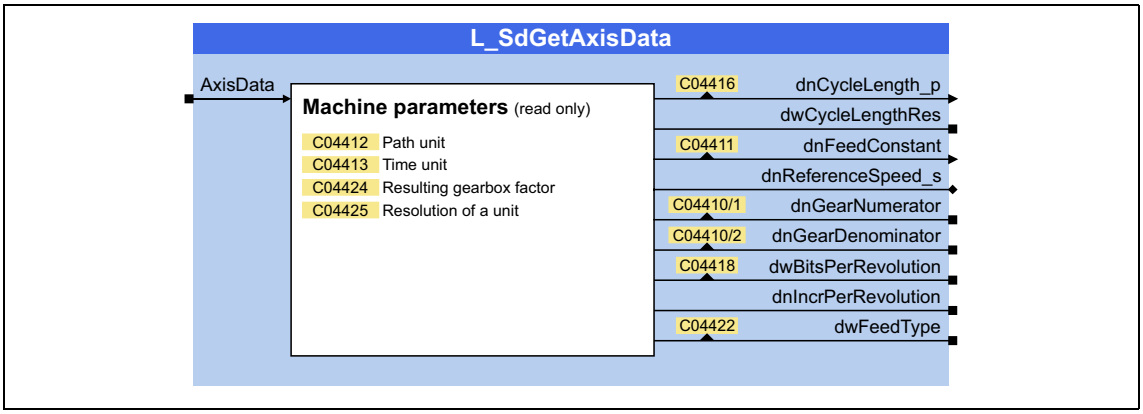
Outputs

Identifier/data type	Value/meaning	
dnOut DINT	Output signal	
bLimit BOOL	Status "Output signal is limited"	
	TRUE	The output signal is limited, the drive does not follow the setpoint.

5.120 L_SdGetAxisData - Reading out machine parameters from axis data

Function library:	LenzeServoDrive	**FB is available as of library V02.07.xx.xx!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

This FB reads out selected machine parameters from the copied axis data and makes them available to other FB which require them for internal calculations (e.g. master value integrators, FB for stretching/compression, and other FB for the edition of cam data).



Inputs

Identifier/data type	Information/possible settings
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning
dnCycleLength_p DINT	Cycle in [increments] • C04416 indicates the cycle in the real unit of the machine.
dwCycleLengthRes DWORD	Remainder of the cycle
dnFeedConstant DINT	Feed constant in [increments] • C04411 indicates the feed constant in the real unit of the machine.
dnReferenceSpeed_s DINT	Reference speed as a speed signal • Scaling: $2^{26} \equiv 15000$ [rpm] • C04420 indicates the reference speed in the real unit of the machine.
dnGearNumerator DINT	Gearbox factor (numerator) • The gearbox factor related to the current position control (C02570) and to the position encoder selection (C00490) is output. • Display parameter: C04410/1 • C04424 indicates the resulting gearbox factor.
dnGearDenominator DINT	Gearbox factor (denominator) • The gearbox factor related to the current position control (C02570) and to the position encoder selection (C00490) is output. • Display parameter: C04410/2 • C04424 indicates the resulting gearbox factor.

Identifier/data type	Value/meaning	
dwBitsPerRevolution DWORD	Resolution of an encoder revolution • Display parameter: C04418	
	10	Bit / encoder revolution
	...	...
	24	Bit / encoder revolution
dwIncrPerRevolution DWORD	Resolution of a revolution in [increments]	
dwFeedType DWORD	Traversing range • Display parameter: C04422	
	0	Unlimited
	1	Limited
	2	Modulo

Parameter

Note!

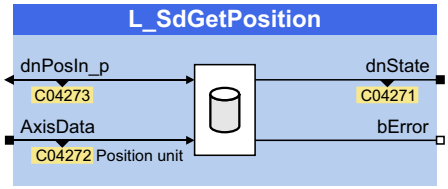
Observe that the index of the codes listed in the following only applies to the first instance!

Parameter	Possible settings			Information
C04410/1	1		2147483647	Gearbox factor (numerator) • Read only
C04410/2	1		2147483647	Gearbox factor (denominator) • Read only
C04411	0.0001	Unit/incr.	214748.3647	Feed constant • Read only
C04412	String of digits with max. 16 characters			Unit • Read only
C04413	String of digits with max. 8 characters			Time unit • Read only
C04414	String of digits with max. 16 characters			Axis name • Read only
C04416	0.0000	Unit	214748.3647	Cycle • Read only
C04418	10	Bit	24	Resolution of an encoder revolution • Read only
C04420	-214748.3647	Unit/t	214748.3647	Reference speed • Read only
C04422				Traversing range • Read only
	0	Unlimited		
	1	Limited		
	2	Modulo		
C04424	-2147483.647		2147483.647	Resulting gearbox factor • Read only
C04425	-214748.3647	Incr./unit	214748.3647	Resolution of a unit • Read only

5.121 L_SdGetPosition - position conversion

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts a position in [inc] into a position in the real unit of the machine. The converted position is displayed in C04273.



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Position in [inc] • Scaling: 1 encoder revolution ≡ 2^{16} increments (or acc. to *AxisData*)
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status	
	0	Ok - no error
	-1	No machine parameters are applied to the input *AxisData*.
	-12	Counter overflow of ±214748.3647 in the display parameter C04273.
bError BOOL	"Error" status signal	
	TRUE	The conversion was incorrect.

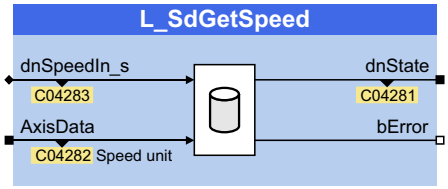
Parameter

Parameter	Possible settings			Information
C04271	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.
C04272 As of library V02.02.xx.xx	String of digits			Position unit • Read only
C04273	-214748.3647	Unit	214748.3647	Position value • Display of the *dnPosIn_p* input signal in the real unit of the machine.

5.122 L_SdGetSpeed - speed conversion

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts a speed into a velocity. The converted value is displayed in C04283.



Inputs

Identifier/data type	Information/possible settings
dnSpeedIn_s DINT	Speed in [rpm] • Scaling: 15000 rpm ≡ 2^{26} ≡ 67108864
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status	
	0	Ok - no error
	-1	No machine parameters are applied to the input *AxisData*.
	-12	Counter overflow of ±214748.3647 in the display parameter C04283.
bError BOOL	"Error" status signal	
	TRUE	The conversion was incorrect.

Parameter

Parameter	Possible settings			Information
C04281	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.
C04282 As of library V02.02.xx.xx	String of digits			Speed unit • Read only
C04283	-214748.3647	Unit/t	214748.3647	Speed value • Read only • Calculated from the speed signal *dnSpeedIn_s*.

5.123 L_SdIntegrate - integration of speed to position

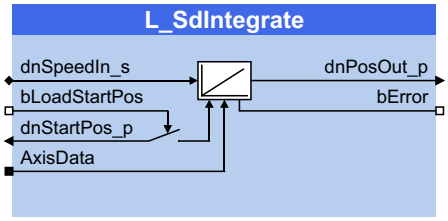
Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB integrates a speed to a position.

- The calculation is made with remainder handling.
- The *bLoadStartPos* input serves to load the starting position pending at the *dnStartPos_p* input into the integrator.

Note!

The FB is not suitable for modulo measuring systems!
(No clocked (Modulo) positions are provided.)



Inputs

Identifier/data type	Information/possible settings
dnSpeedIn_s DINT	Speed in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864
bLoadStartPos BOOL	Load integrator with starting position and reset overflow signal. TRUE: Load the integrator with the value applied to input *dnStartPos_p* and reset *bError* to FALSE.
dnStartPos_p DINT	Starting position in [inc] with which the integrator is loaded by setting *bLoadStartPos* to TRUE.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510) • If this input is released, the following scaling applies: 1 motor revolution ≡ 2^{16}

Outputs

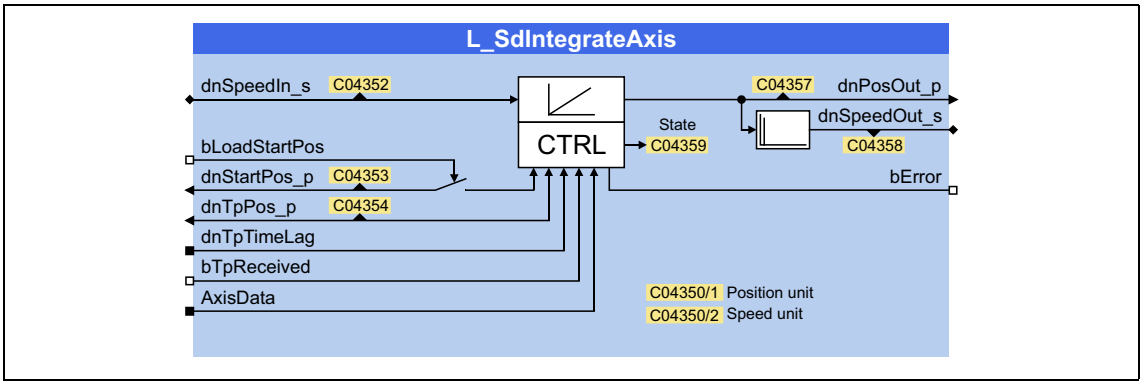
Identifier/data type	Value/meaning
dnPosOut_p DINT	Position in [inc] • Overflow is possible (display via *bError*)
bError BOOL	Status signal "Overflow occurs" • Status signal can be reset via *bLoadStartPos*. TRUE: Overflow has occurred.

5.124 L_SdIntegrateAxis - speed to position integration (with TP correction)

Function library:	LenzeServoDrive	**FB is available as of library V02.03.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB integrates a speed to a position.

- In contrast to the FB L_SdIntegrate, in the case of this FB a correction of the position output via touch probe can additionally be effected, i. e. the integrator considers the time from the detection of a touch probe to the next task cycle.
- The *bLoadStartPos* input serves to load the starting position pending at the *dnStartPos_p* input into the integrator.
- A touch-probe-exact (mark-controlled) start of the integrator can be carried out.
- In addition to the position, the resulting speed is shown at the output *dnSpeedOut_s*.
- The FB can be used for all measuring systems (modulo/limited/unlimited).



Inputs

Identifier/data type	Information/possible settings
dnSpeedIn_s DINT	Speed in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864 • Display parameter: C04352
bLoadStartPos BOOL	Load integrator with starting position and reset error output. TRUE: Load the integrator with the value applied to input *dnStartPos_p* and reset *bError* to FALSE.
dnStartPos_p DINT	Starting position in [inc] with which the integrator is loaded by setting *bLoadStartPos* to TRUE. • C04353 indicates the starting position in the real unit of the machine.
dnTpPos_p DINT	Position of the touch probe sensor in [increments]. • C04354 indicates the position in the real unit of the machine.
dnTpTimeLag DINT	Input for accepting the touch probe time stamp • Connect this input with the output *dnTouchProbeTimeLag* of the corresponding touch probe system block.

Identifier/data type	Information/possible settings
bTpReceived BOOL	Input for taking over the status "Touch probe detected" • Connect this input with the output *dnTouchProbeReceived* of the corresponding touch probe system block. • As long as the input *bLoadStartPos* is set to TRUE, this input has no effect.
	FALSE↗TRUE: Touch probe detected.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510) • If this input is released, the following scaling applies: 1 motor revolution ≡ 2^{16}

Outputs

Identifier/data type	Value/meaning
dnPosOut_p DINT	Position output in [inc] • Overflow is possible (display via *bError*) • C04357 indicates the position in the real unit of the machine.
dnSpeedOut_s DINT	Speed output in [rpm] • C04358 indicates the speed in the real unit of the machine.
bError BOOL	"Error" status • Status signal can be reset via *bLoadStartPos*.
	TRUE: An error has occurred (group signal). • For details see display parameter C05289.

Parameter

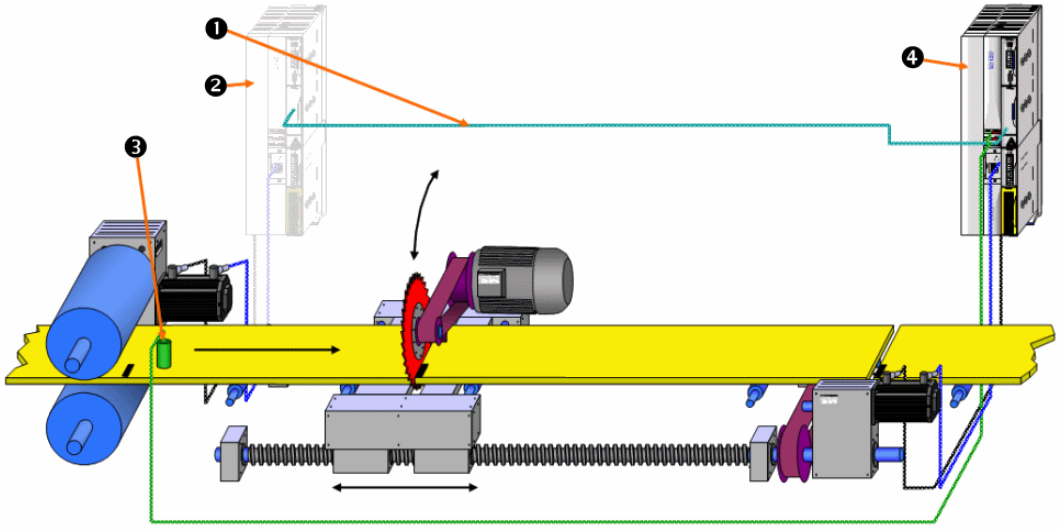
Parameter	Possible settings			Information
C04350/1	String of digits			Position unit • Read only
C04350/2	String of digits			Speed unit • Read only
C04352	-214748.3647	Unit/t	214748.3647	Speed input • Display of the *dnSpeedIn_s* input signal in the real unit of the machine.
C04353	-214748.3647	Unit	214748.3647	Starting position • Display of the *dnStartPos_p* input signal in the real unit of the machine.
C04354	-214748.3647	Unit	214748.3647	TP position • Display of the *dnTpPos_p* input signal in the real unit of the machine.
C04357	-214748.3647	Unit	214748.3647	Output position • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C04358	-214748.3647	Unit/t	214748.3647	Output speed • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.

Parameter	Possible settings		Information
C04359	Bit coded:		Status • Bits that are not itemised are reserved for future extensions.
	bits10	Zero crossing active	
	bit16	Position is beyond the cycle	
	bits17	Initial speed too high	
	bits23	Invalid axis data structure	
	bits31	General error	

5.124.1 Typical application

Mark-controlled start of a following axis

The following illustration shows the "flying saw" application. If no mark has been detected, the slave drive remains in its waiting position. Even if the material is fed, no material motion is shown via the position output of the FB, as no mark signal has been detected. Only if a mark has been detected via the touch probe sensor, the internal integrator starts and maps the material feed.



❶ Digital frequency connection
❷ Master drive (material feed)
❸ Touch probe sensor for mark-controlled start
❹ Slave drive (flying saw)

[5-74] Example: flying saw

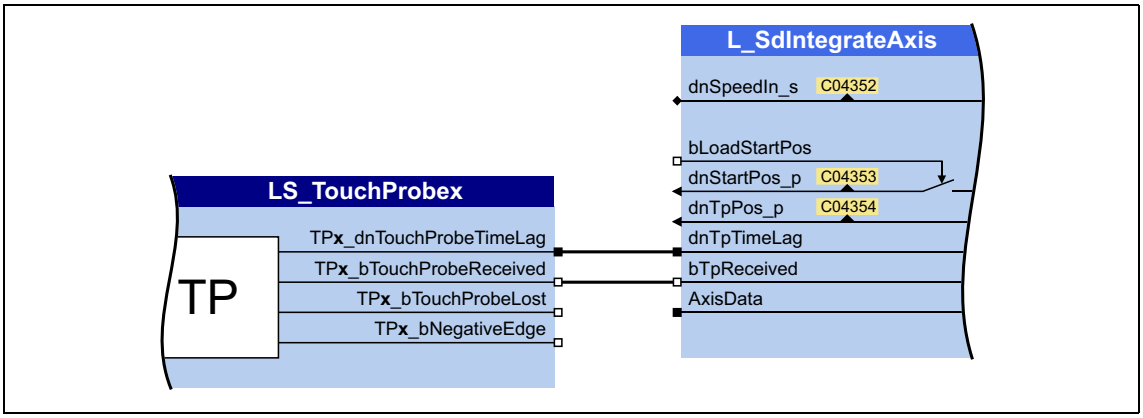
Measuring integrator for master position

In many cases integrators are not only required for motion control, but simply for generating comparison measuring systems (e. g. master cycle generation without direct influence on the drive motion).

The FB makes it possible to set the positions once, or even cyclically in a "hard" manner, via touch probe signals, which is desired in such cases.

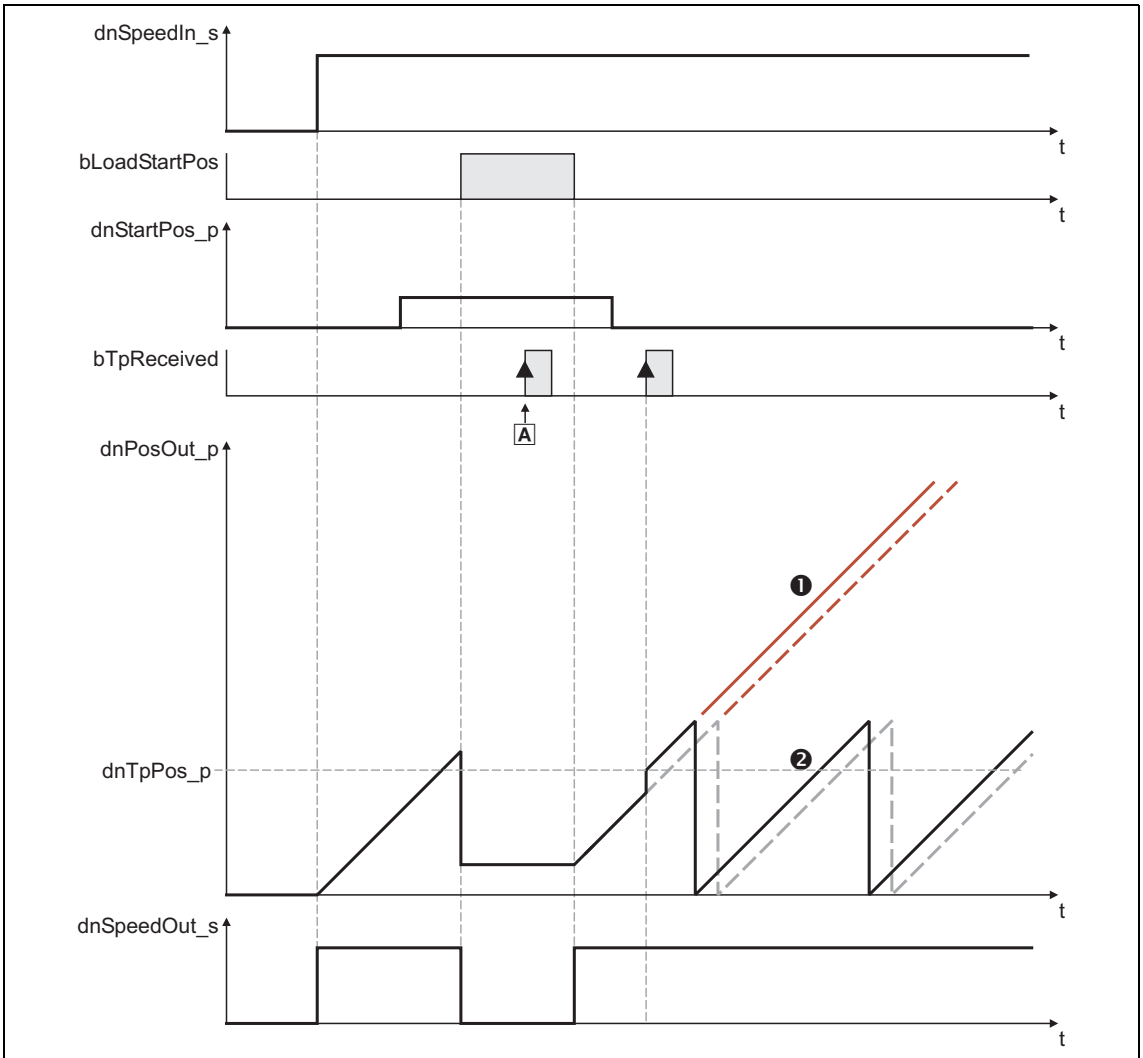
5.124.2 Interconnection of the touch probe inputs

For the position correction via touch probe the FB requires the touch probe time stamp and the "Touch probe detected" status signal from the **LS_TouchProbe** system block for the touch probe channel used:



[5-75] Example: Interconnection of the touch probe inputs of the L_SdIntegrateAxis FB

5.124.3 Signal characteristics

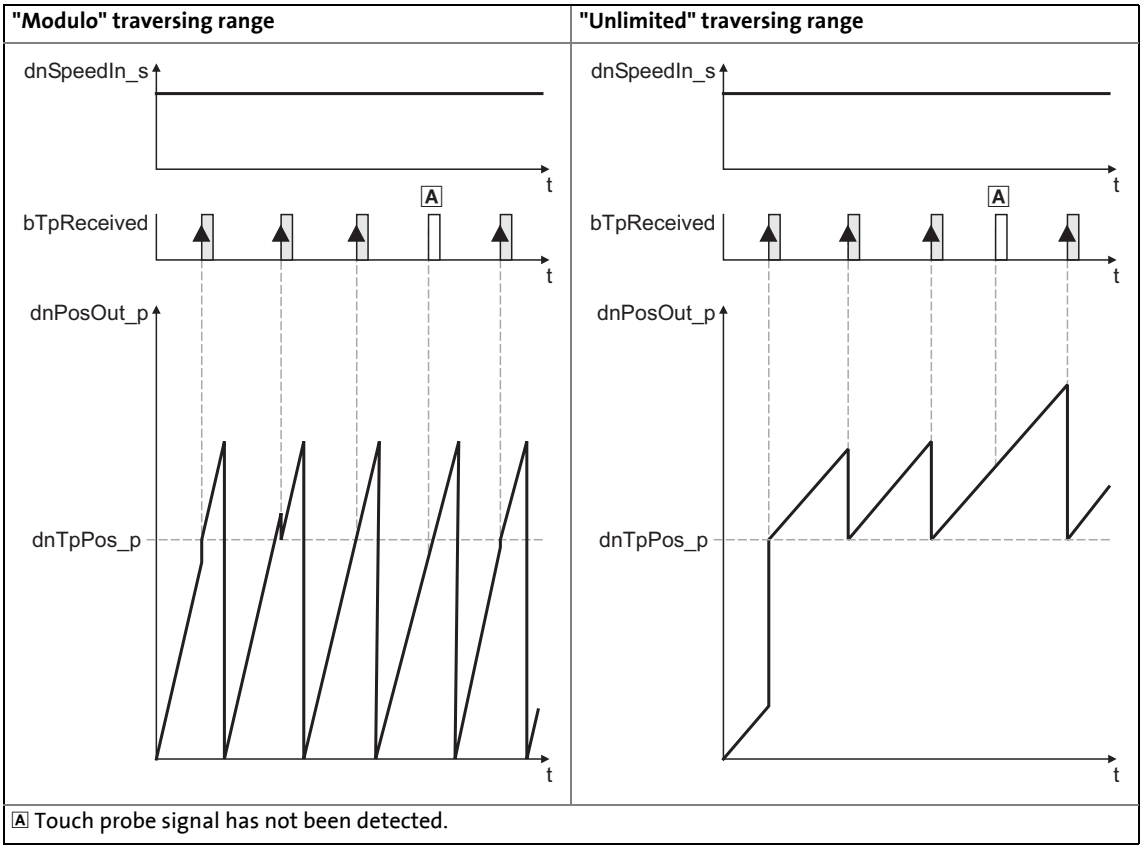


❶ "Unlimited" traversing range
❷ "Modulo" traversing range
A Touch probe signal has no effect, as *bLoadStartPos* is set to TRUE.

[5-76] Signal characteristics

Touch probe correction

In contrast to the FB L_SdIntegrate, in the case of this FB it is possible to correct the position output additionally via touch probe.



[5-77] Signal characteristics of touch probe correction

5.125 L_SdIntegrateLimit - integration of speed to position

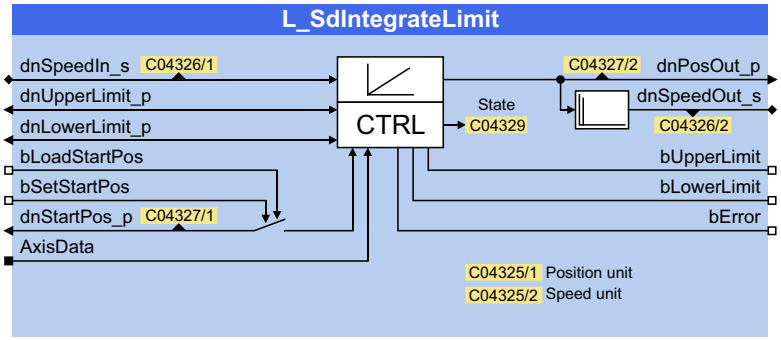
Function library:	LenzeServoDrive	**FB available from library V02.09.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB integrates a speed to a position.

- The cycle (return) can be defined via inputs *dnUpperLimit_p* and *dnLowerLimit_p*.
- The starting position applied at the *dnStartPos_p* input can be loaded into the integrator in a level-triggered manner via the *bLoadStartPos* input or in an edge-triggered manner via the *bSetStartPos* input.

Note!

During modulo operation, a cycle defined in the machine parameters that are applied (*AxisData*) will not be taken into account!



Inputs

Identifier/data type	Information/possible settings
dnSpeedIn_s DINT	Speed in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864 • Display parameter: C04326/1
dnUpperLimit_p DINT	Upper limit in [increments] for the position signal return
dnLowerLimit_p DINT	Lower limit in [increments] for the position signal return
bLoadStartPos BOOL	Load the integrator with starting position (level-triggered) TRUE: Load the integrator with the value at the *dnStartPos_p* input.
bSetStartPos BOOL	Load the integrator with starting position (edge-triggered) FALSE↗TRUE: Load the integrator with the value at the *dnStartPos_p* input. The speed is continued to be integrated immediately.
dnStartPos_p DINT	Starting position in [increments] used to load the integrator by setting *bLoadStartPos* to TRUE or by a FALSE-TRUE edge at *bSetStartPos*.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnPosOut_p DINT	Position in [inc] • Overflow at *dnUpperLimit_p* and *dnLowerLimit_p*.	
dnSpeedOut_s DINT	Speed output in [rpm] • C04326/2 indicates the speed in the real unit of the machine. • If *bLoadStartPos* = TRUE, "0" is provided.	
bUpperLimit BOOL	Status signal "Return from upper value is executed"	
	TRUE	The integrator has exceeded the value at *dnUpperLimit_p*. The integration process is continued at *dnLowerLimit_p* without any loss of increments.
bLowerLimit BOOL	Status signal "Return from lower value is executed"	
	TRUE	The integrator has exceeded the value at *dnLowerLimit_p*. The integration process is continued at *dnUpperLimit_p* without any loss of increments.
bError BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameter C04329.

Parameters

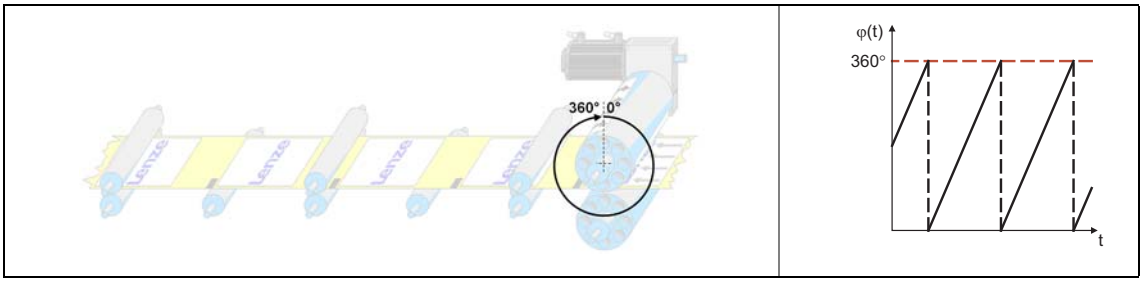
Parameter	Possible settings			Info
C04325/1	Character string			Position unit • Read only
C04325/2	Character string			Speed unit • Read only
C04326/1	-214748.3647	Unit/t	214748.3647	Speed input • Display of the *dnSpeedIn_s* input signal in the real unit of the machine.
C04326/2	-214748.3647	Unit/t	214748.3647	Output speed • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.
C04327/1	-214748.3647	Unit	214748.3647	Starting position • Display of the *dnStartPos_p* input signal in the real unit of the machine.
C04327/2	-214748.3647	Unit	214748.3647	Output position • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C04329	Status messages:			Status (bit-coded) • Bits that are not listed are reserved for future extensions. • The *bError* error output will only be set to TRUE if an error message (bit 15 ... 31) is issued. • Bit 31 is the default setting if one or more other error bits (bit 15 ... 30) are set.
	Bit 6	Upper limit reached/exceeded		
	Bit 7	Lower limit reached/exceeded		
	Bit 10	Zero crossing active		
	Error messages:			
	bit16	Position is beyond the cycle		
	bits17	Initial speed too high		
	Bit 23	Invalid axis data structure		
	bits31	General error		

5.125.1 Basics

Many machine structures and processes produce measuring systems which display a periodic feed motion (so-called modulo measuring systems). These measuring systems thus represent a position range of 0 ... s_{max}.

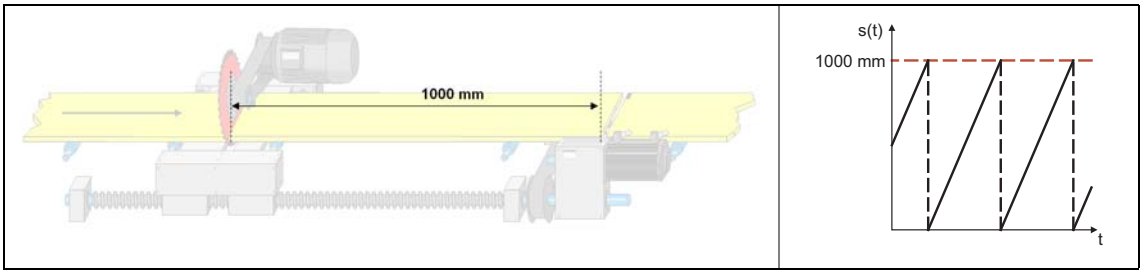
The following cases can be further distinguished:

- Case 1: Modulo measuring systems with a **reference to the mechanical system**
  - The modulo cycle results solely from the machine structure.



[5-78] Example of a reference to the mechanical system

- Case 2: Modulo measuring systems with a **process reference**
  - The modulo cycle is a process parameter (e.g. here product length = 1000 mm) and thus is variable.



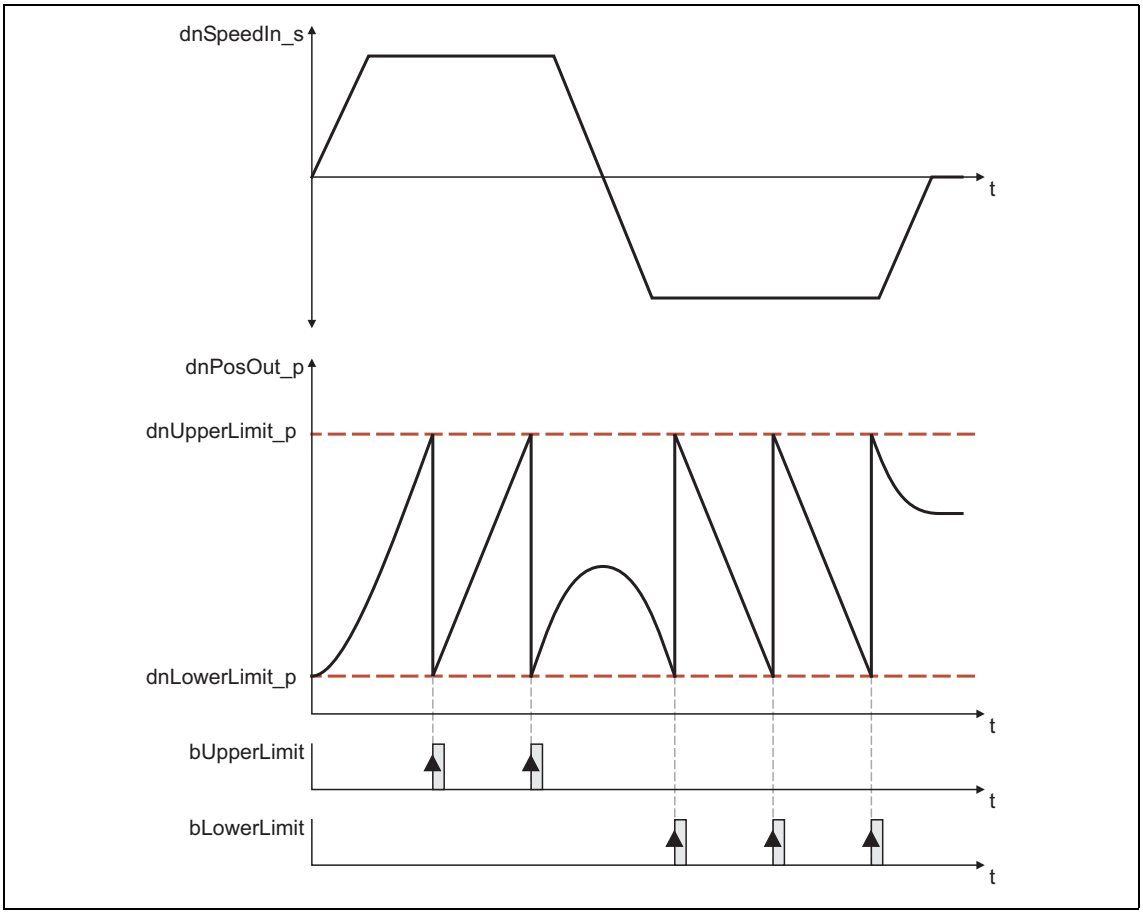
[5-79] Example of a process reference

Selection help

For both cases integrator function blocks are provided, calculating a modulo position from an input speed:

- Case 1: L_LdIntegrateCyclic, L_SdIntegrateAxis, L_EsStretchIntegrate
- Case 2: **L_SdIntegrateLimit**

5.125.2 Signal characteristics



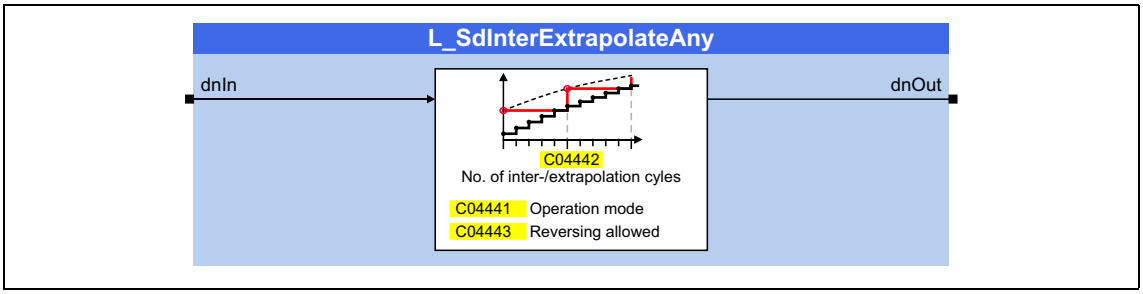
[5-80] Signal characteristics

5.126 L_SdInterExtrapolateAny - signal interpolation

Function library:	LenzeServoDrive	**FB available from library V02.09.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB serves to interpolate or extrapolate setpoints which have a greater resolution than the task cycle.

- The L_SdInterExtrapolatePosition FB serves to interpolate or extrapolate position information. (📖 474)



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input value (setpoint)

Outputs

Identifier/data type	Value/meaning
dnOut DINT	Output value (in the cycle)

Parameter

Parameter	Possible settings			Information
C04441				Operation mode
	0	Linear interpolation		Lenze setting
	1	Quadratic interpolation		
	2	Linear extrapolation		
	3	Quadratic extrapolation		
C04442	1		127	Number of interpolation/extrapolation steps • Task cycles per communication cycle • Initialisation: 1
C04443				Permit reversing during quadratic extrapolation
	0	Reversing not permitted		Lenze setting
	1	Reversing permitted		

5.126.1 Extrapolation function

The FB fills in missing process data by extrapolation. These process data may be missing because of stochastic errors (e.g. EMC interferences) or the bus cycle being greater than the task cycle of the application.

- Select a linear or a quadratic extrapolation in C04441.
- Set the number of extrapolation steps in C04442:

$$C04442 = \frac{\text{Bus cycle [ms]}}{\text{Task cycle [ms]}}$$

5.126.2 Interpolation function

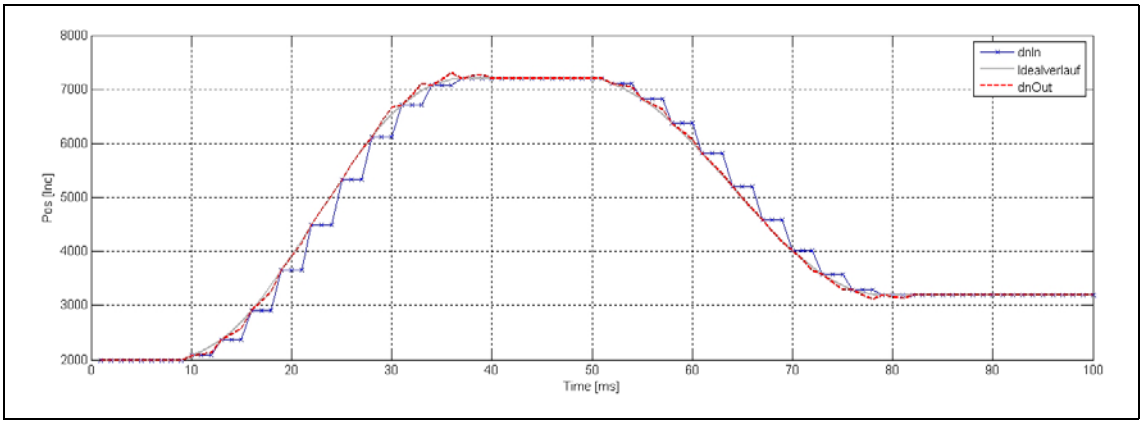
The FB fills in missing process data by interpolation. These process data are missing because the bus cycle is greater than the task cycle of the application.

- Select a linear or a quadratic interpolation in C04441.
- Set the number of interpolation steps in C04442:

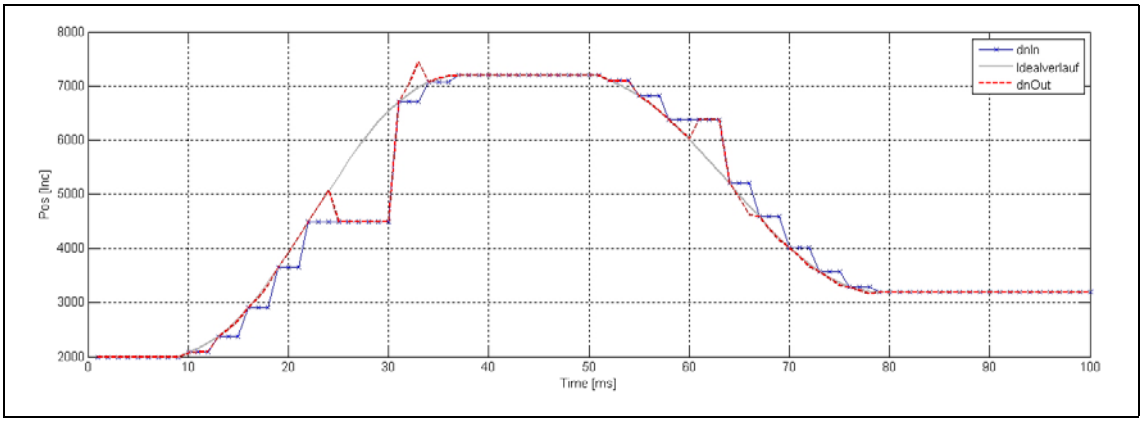
$$C04442 = \frac{\text{Bus cycle [ms]}}{\text{Task cycle [ms]}}$$

5.126.3 Signal characteristics

Extrapolation

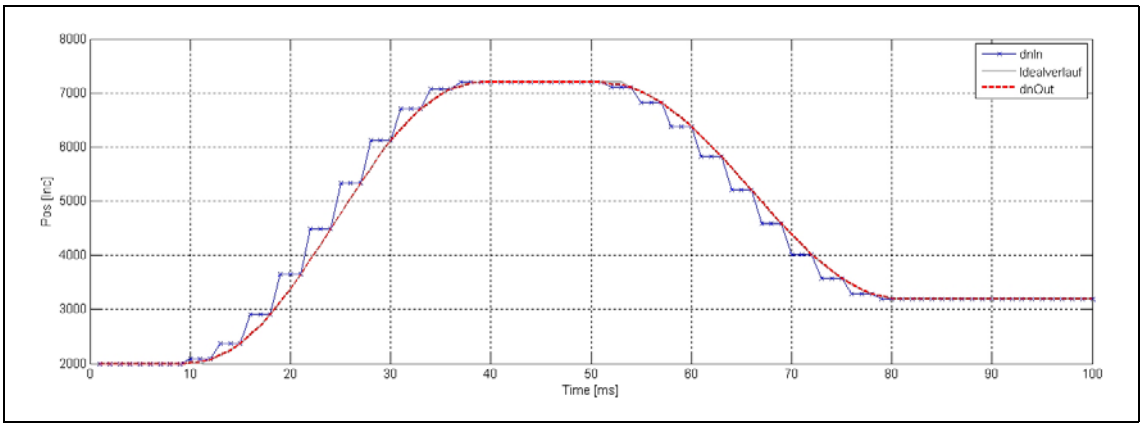


[5-81] Quadratic extrapolation

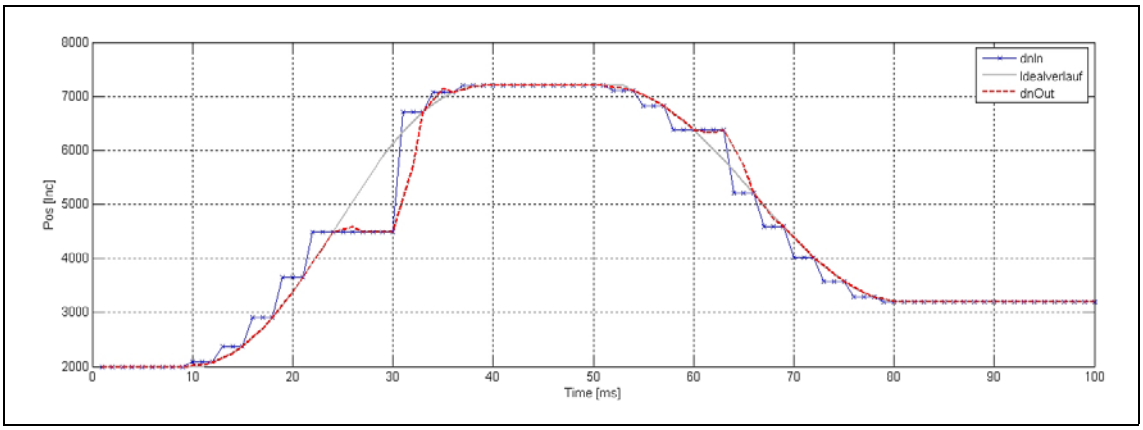


[5-82] Quadratic extrapolation with telegram failure

Interpolation



[5-83] Quadratic interpolation



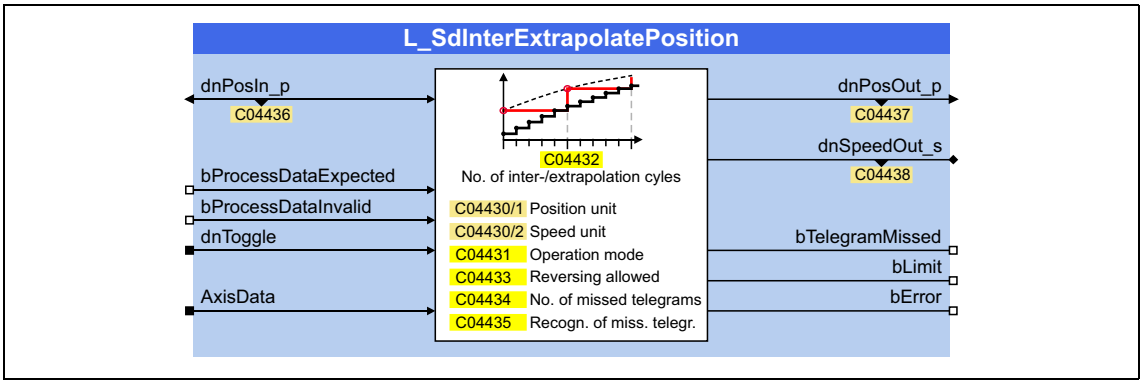
[5-84] Quadratic interpolation with telegram failure

5.127 L_SdInterExtrapolatePosition - signal interpolation

Function library:	LenzeServoDrive	**FB available from library V02.09.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB serves to interpolate or extrapolate position information with a greater resolution than the task cycle. Furthermore, the FB provides measures for filling in the gaps in the case of telegram failures.

- The L_SdInterExtrapolateAny FB serves to interpolate or extrapolate other data types. (📖 471)



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Input position (setpoint) in [increments]
bProcessDataExpected BOOL	SyncInput interface for telegram failure detection • This input is only evaluated if C04435 = "2". • If this input is set to TRUE (e.g. via the **LS_SyncInput** SB), the FB expects a new position value at the *dnPosIn_p* input at that time.
	FALSE: A new process datum at *dnPosIn_p* is not expected.
	TRUE: A new process datum at *dnPosIn_p* is expected.
bProcessDataInvalid BOOL	SyncInput interface for telegram failure detection • This input is only evaluated if C04435 = "2". • The signal is only valid in conjunction with *bProcessDataExpected*. • If this input is set to TRUE (e.g. via the **LS_SyncInput** SB), the FB invalidates the new position value at the *dnPosIn_p* input.
	FALSE: The process datum at *dnPosIn_p* is valid.
	TRUE: The process datum at *dnPosIn_p* is invalid.
dnToggle DINT	Toggle input for telegram failure detection • This input is only evaluated if C04435 = "1". • The input is supposed to receive a 32-bit value which had been received from the master control and which changes with every communication cycle. If the *dnToggle* input value remains unchanged, this behaviour will be evaluated as telegram failure.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnPosOut_p DINT	Output position (in the cycle) in [increments]	
dnSpeedOut_s DINT	Output speed (in the cycle) provided as speed in [min^{-1}] • 15000 rpm ≡ 2^{26} ≡ 67108864	
bTelegramMissed BOOL	Status signal "Telegram failure detected"	
	TRUE	Telegram failure has been detected.
bLimit BOOL	Status signal "Number of accepted telegram failures reached"	
	TRUE	The number of accepted telegram failures set in C04434 has been reached.
bError BOOL	"Error" status	
	TRUE	An error has occurred.

Parameter

Parameter	Possible settings			Information
C04430/1	String of digits			Position unit • Read only
C04430/2	String of digits			Speed unit • Read only
C04431				Operation mode
	0	Linear interpolation		Lenze setting
	1	Quadratic interpolation		
	2	Linear extrapolation		
	3	Quadratic extrapolation		
C04432	1		127	Number of interpolation/extrapolation steps • Task cycles per communication cycle • Initialisation: 1
C04433				Permit reversing during quadratic extrapolation
	0	Reversing not permitted		Lenze setting
	1	Reversing permitted		
C04434	0		50	Number of accepted telegram failures • at the same time number of communication cycles used for further interpolation/extrapolation. • Initialisation: 0
C04435				Selection: Telegram failure detection
	0	Inactive (Two identical input position values in a row are not evaluated as telegram failure.)		Lenze setting
	1	Toggle input		
	2	SyncInput interface		
C04436	-214748.3647	Unit	214748.3647	Position at the input • Display of the *dnPosIn_p* input signal in the real unit of the machine.

Parameter	Possible settings			Information
C04437	-214748.3647	Unit	214748.3647	Position at the output • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C04438	-214748.3647	Unit/t	214748.3647	Speed at the output • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.

5.127.1 Extrapolation function

The FB fills in missing position data by extrapolation. These position data may be missing because of stochastic errors (e.g. EMC interferences) or the bus cycle being greater than the task cycle of the application.

- Select a linear or a quadratic extrapolation in C04431.
- Set the number of extrapolation steps in C04432:

$$\text{C04432} = \frac{\text{Bus cycle [ms]}}{\text{Task cycle [ms]}}$$

Note!

A great number of interpolation/extrapolation steps may cause the extrapolated output signal to reverse, e.g. if the input signal with a rough resolution is brought to a standstill.

- Select in C04433 if reversing is permitted or not. In the Lenze setting of C04433, reversing is not permitted.

5.127.2 Interpolation function

The FB fills in missing position data by interpolation. These process data are missing because the bus cycle is greater than the task cycle of the application.

- Select a linear or a quadratic interpolation in C04431.
- Set the number of interpolation steps in C04432:

$$\text{C04432} = \frac{\text{Bus cycle [ms]}}{\text{Task cycle [ms]}}$$

5.127.3 Telegram failure detection

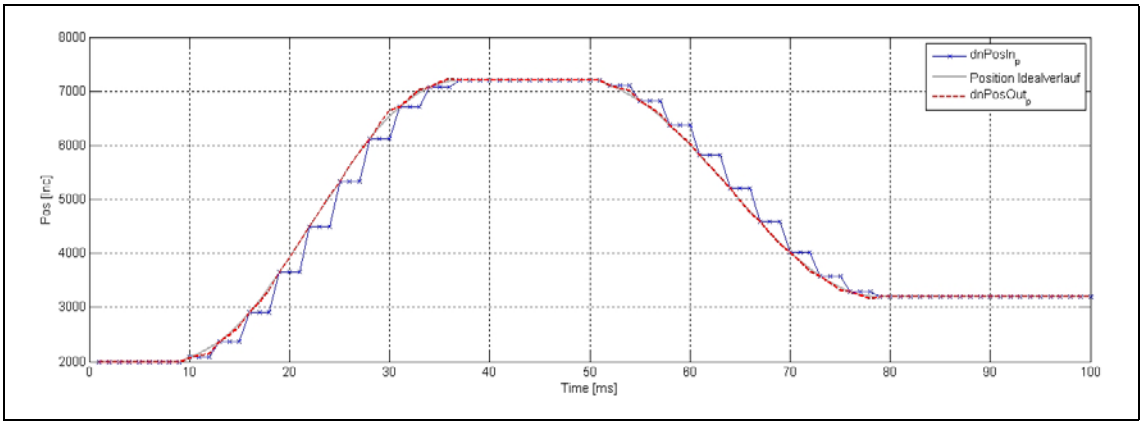
An optional telegram failure detection can be activated in C04435, either via toggle input or SyncInput interface (SB **LS_SyncInput**).

- Telegram failure detection serves to fill in missing process data values.
- The maximum number of accepted telegram failures before interpolation/extrapolation is stopped can be set in C04434.

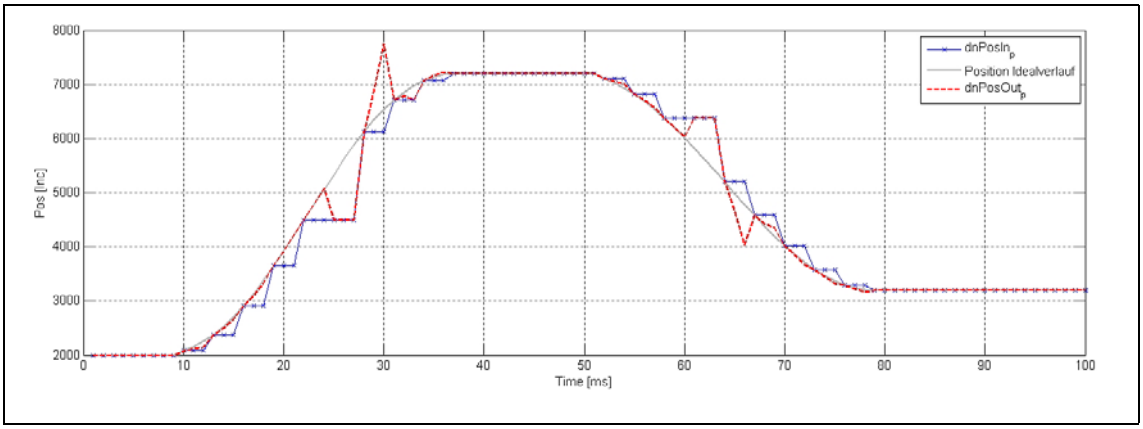
5.127.4 Signal characteristics

Extrapolation without telegram failure detection

Selection: Telegram failure detection (C04435) = "0: Deactivated"



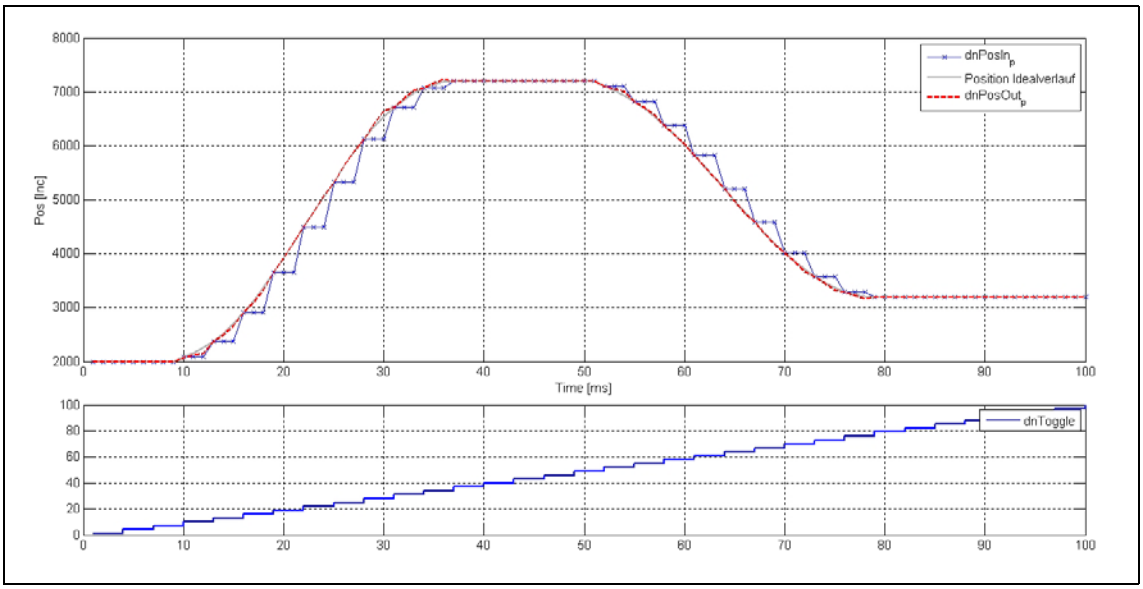
[5-85] Input values OK



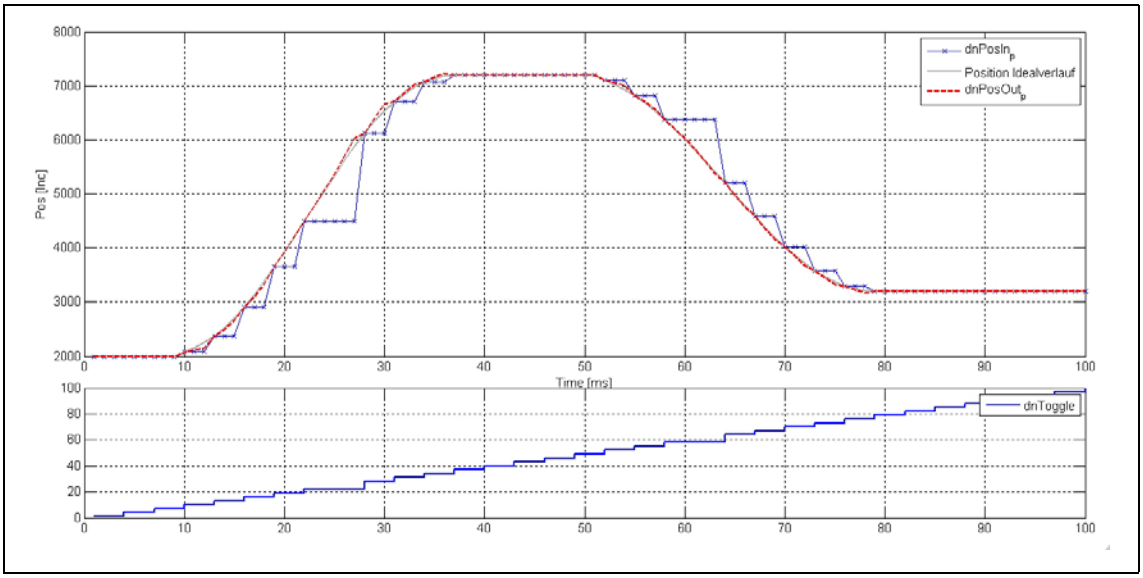
[5-86] Input values not OK, number of accepted telegram failures = 0

Extrapolation with telegram failure detection via toggle input

Selection: Telegram failure detection (C04435) = "1: Toggle input"



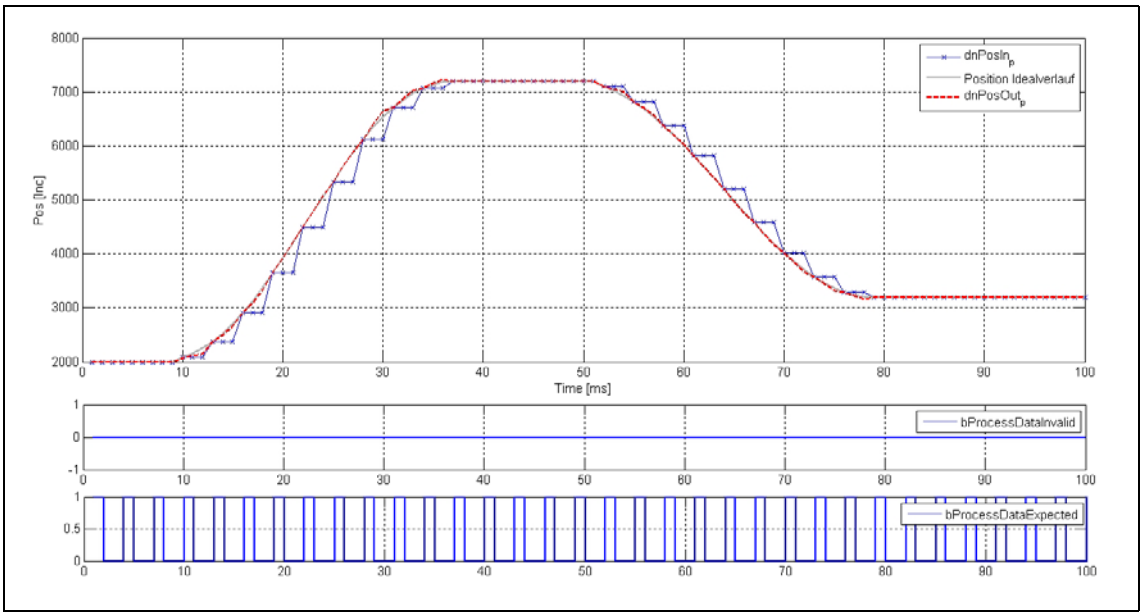
[5-87] Input values OK



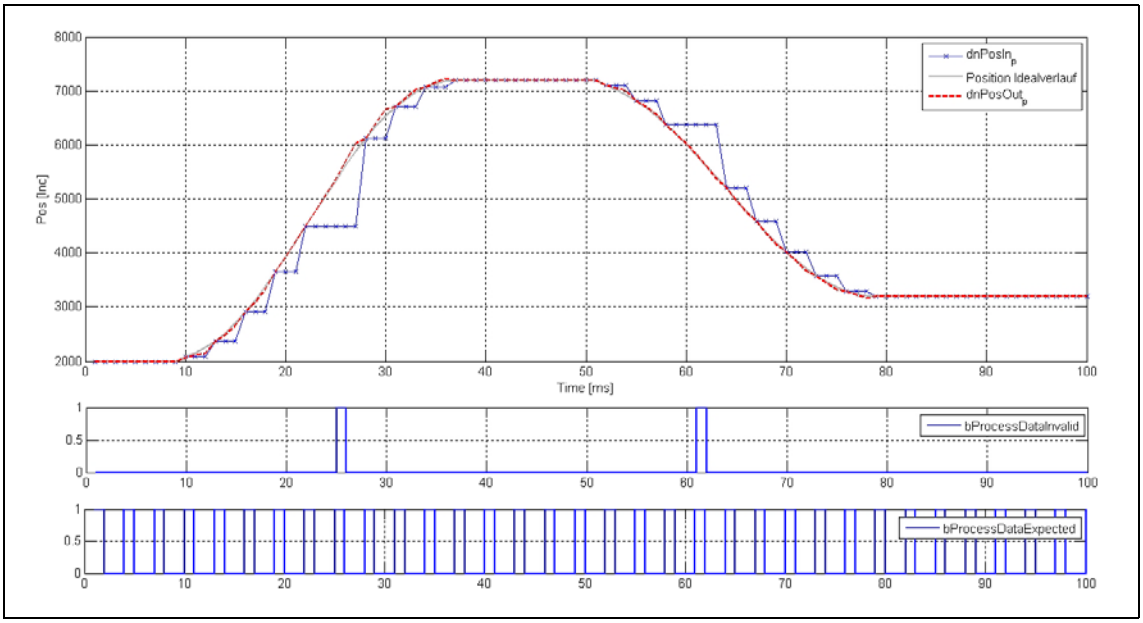
[5-88] Input values not OK, number of accepted telegram failures = 1

Extrapolation with telegram failure detection via SyncInput interface

Selection: Telegram failure detection (C04435) = "2: SyncInput interface"



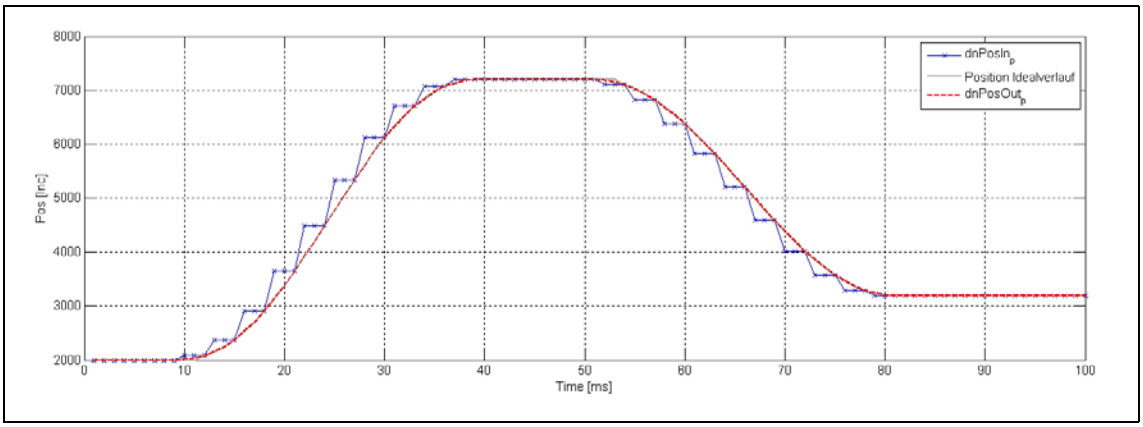
[5-89] Input values OK



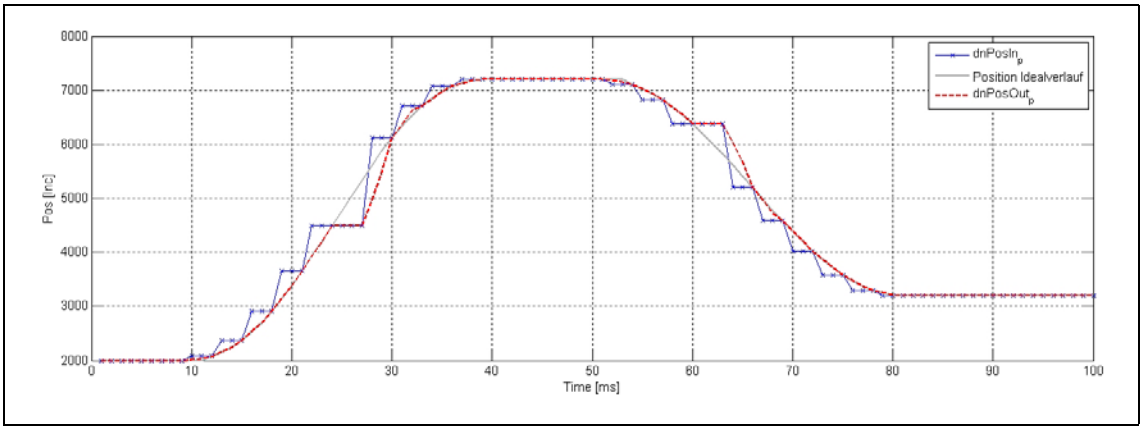
[5-90] Input values not OK, number of accepted telegram failures = 1

Interpolation without telegram failure detection

Selection: Telegram failure detection (C04435) = "0: Deactivated"



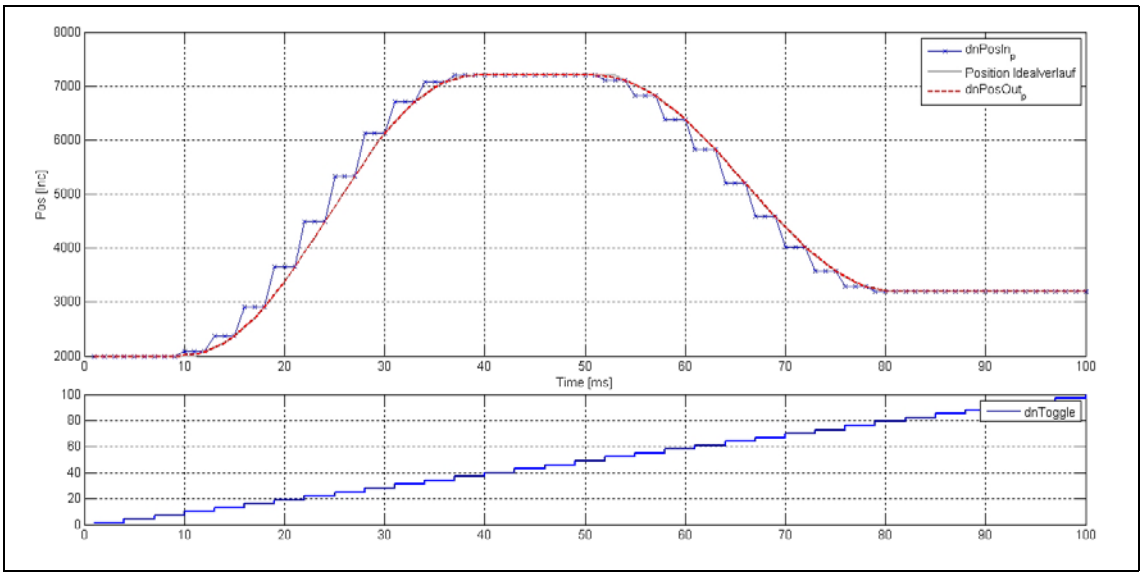
[5-91] Input values OK



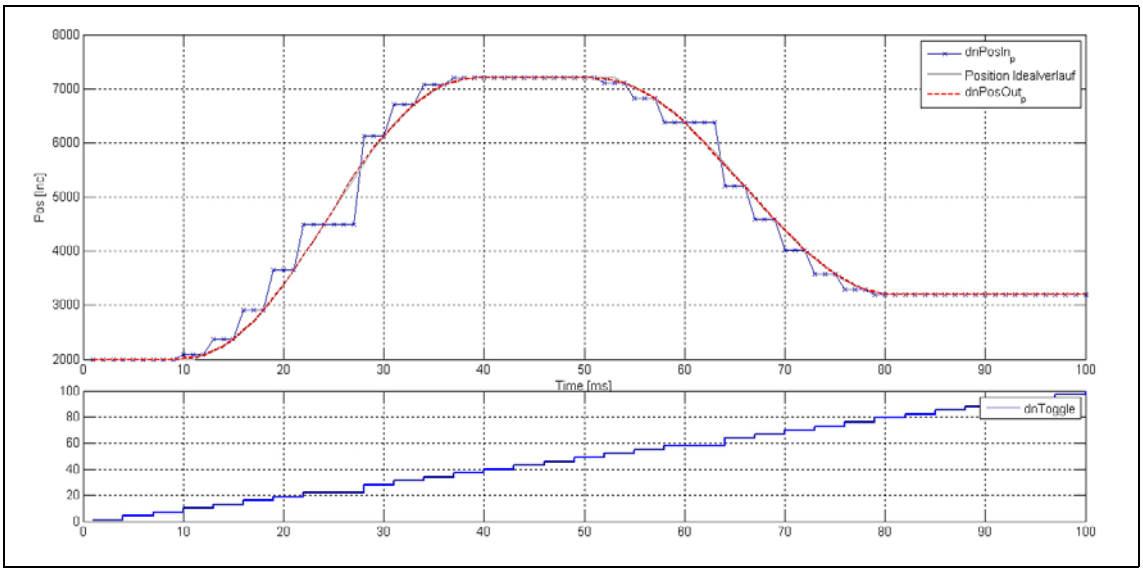
[5-92] Input values not OK, number of accepted telegram failures = 0

Interpolation with telegram failure detection via toggle input

Selection: Telegram failure detection (C04435) = "1: Toggle input"



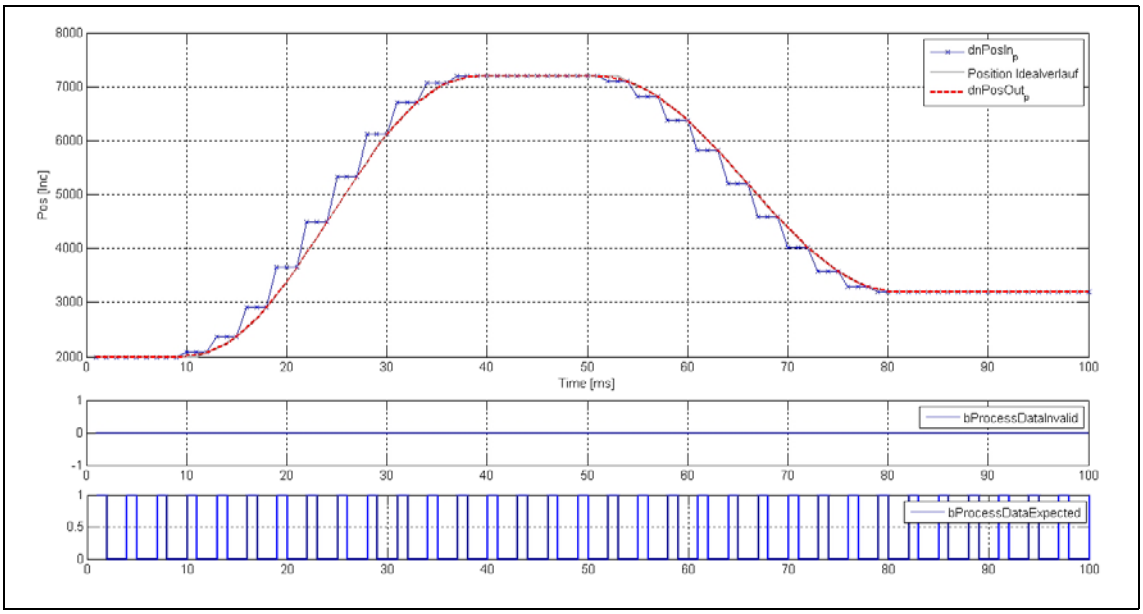
[5-93] Input values OK



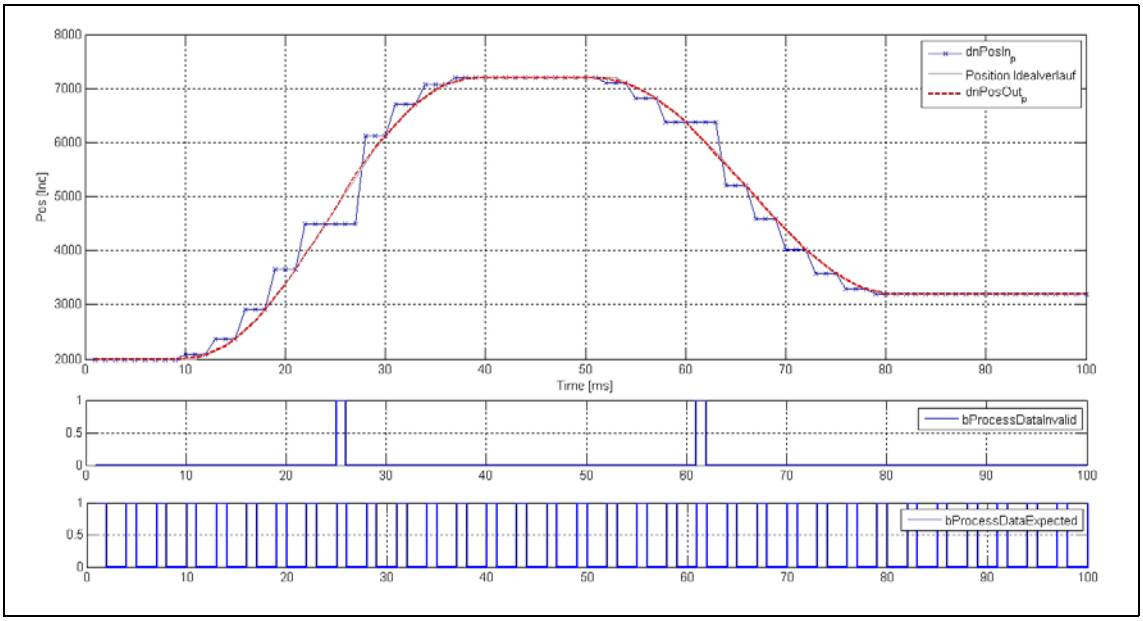
[5-94] Input values not OK, number of accepted telegram failures = 1

Interpolation with telegram failure detection via SyncInput interface

Selection: Telegram failure detection (C04435) = "2: SyncInput interface"



[5-95] Input values OK



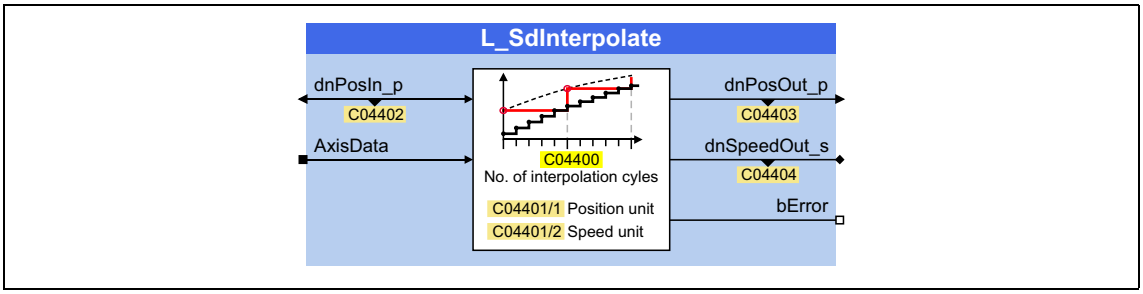
[5-96] Input values not OK, number of accepted telegram failures = 1

5.128 L_SdInterpolate - signal interpolation

Function library:	LenzeServoDrive	**FB available from library V02.00.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB is used to interpolate position information, e.g. to compensate for long bus transfer cycles or smooth low-resolution absolute-value encoders.

- The FB L_LdExtrapolate is available for extrapolation. (📖 319)



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Position of the external, and, as the case may be, insufficiently resolved master shaft in [increments].
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnPosOut_p DINT	Master position output (in the cycle) in [inc]	
dnSpeedOut_s DINT	Master speed output (in the cycle) in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864	
bError BOOL	"Error" status signal	
	TRUE	Master position *dnPosIn_p* has exceeded the cycle defined in the machine parameters (*AxisData*).

Parameter

Parameter	Possible settings			Information
C04400	1		16	Number of the interpolation cycles • Initialisation: 1
C04401/1 As of library V02.02.xx.xx	String of digits			Position unit • Read only
C04401/2 As of library V02.02.xx.xx	String of digits			Speed unit • Read only

Parameter	Possible settings			Information
C04402 As of library V02.02.xx.xx	-214748.3647	Unit	214748.3647	Position at the input • Display of the *dnPosIn_p* input signal in the real unit of the machine.
C04403 As of library V02.02.xx.xx	-214748.3647	Unit	214748.3647	Position at the output • Display of the *dnPosOut_p* output signal in the real unit of the machine.
C04404 As of library V02.02.xx.xx	-214748.3647	Unit/t	214748.3647	Speed at the output • Display of the *dnSpeedOut_s* output signal in the real unit of the machine.

5.128.1 Function

The number of interpolation cycles to be set under C04400 is calculated by means of the following formula:

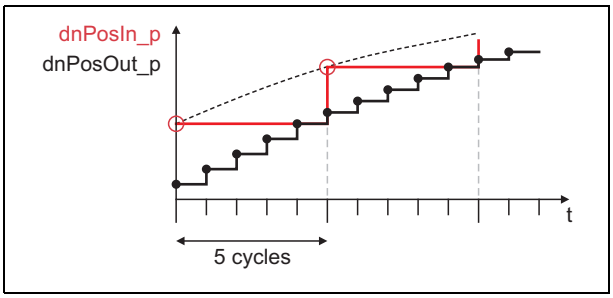
$$C04400 = \frac{\text{Bus cycle [ms]}}{\text{Task cycle [ms]}}$$

Example

If the 9400 HighLine controller (task cycle = 1 ms) receives a position value in a 5-ms cycle from a higher-level control at *dnPosIn_p*, set "5" under C04400.

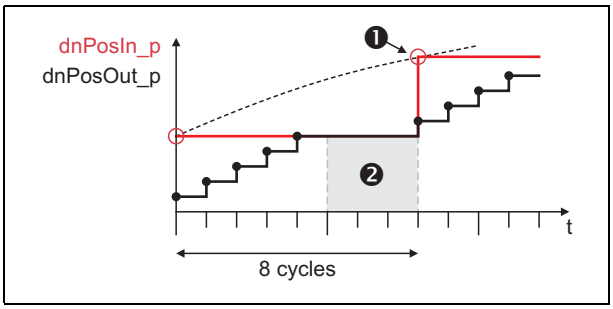
Applications

In the following examples, the number of interpolation cycles selected under C04400 is set to "5".



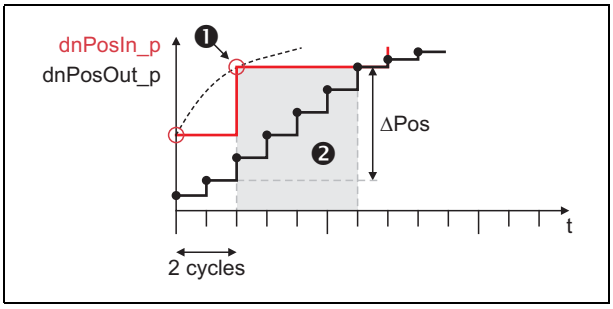
[5-97] Case 1: Cycle time of the input signal = number of interpolation cycles

- In every cycle, the interpolated position value is output at *dnPosOut_p*.



[5-98] Case 2: Cycle time of the input signal > number of interpolation cycles

- If a new value is only received after completion of the five interpolation cycles (❶), the output constantly outputs the input value after completion of the five interpolation cycles (❷).



[5-99] Case 3: Cycle time of the input signal < number of interpolation cycles

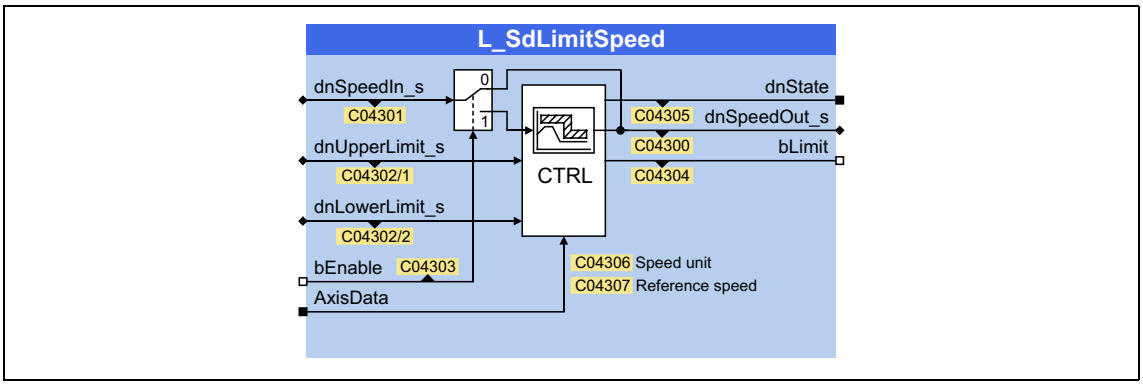
- If a new value is received before completion of the five interpolation cycles (❶), the new interpolation values are calculated by means of the difference between actual and setpoint position (❷).

5.129 L_SdLimitSpeed - speed limitation

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB is used to lead a speed setpoint to defined limit ranges.

- The overflow buffer has a sample rate of 8 revolutions/ms.



Inputs

Identifier/data type	Information/possible settings	
dnSpeedIn_s DINT	Speed input signal in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864	
dnUpperLimit_s DINT	Upper speed limit in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864	
dnLowerLimit_s DINT	Lower speed limit in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864	
bEnable BOOL	Activate speed limitation	
	FALSE	Speed limitation is deactivated. • The input signal *dnSpeedIn_s* is output unchanged to the output *dnSpeedOut_s*. • The overflow buffer is cleared.
	TRUE	Speed limitation is activated. • The input signal *dnSpeedIn_s* is output to the output *dnSpeedOut_s* considering the defined speed ranges.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded)	
	bits0	Ok - no error
	Bit15	Fault (group signal)
	bit16	Speed output signal is limited to the upper speed limit.
	bits17	Speed output signal is limited to the lower speed limit.
	bits18	No speed limits defined, the speed limitation is not active.
	Bit 19	Speed limits are invalid.
	bits20	Overflow of the overflow buffer
dnSpeedOut_s DINT	Speed output signal in [rpm]	
bLimit BOOL	Status signal "Speed limitation is active"	
	TRUE	The speed output signal is limited.

Parameter

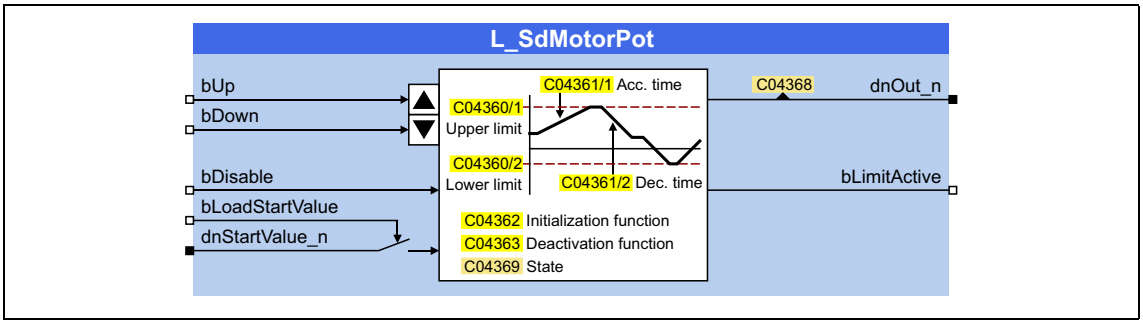
Parameter	Possible settings			Information
C04300	-214748.3647	Unit/t	214748.3647	Speed at the output • Read only • Calculated from the speed signal *dnSpeedOut_s*.
C04301	-214748.3647	Unit/t	214748.3647	Speed at the input • Read only • Calculated from the speed signal *dnSpeedIn_s*.
C04302/1 As of library V02.02.xx.xx	-214748.3647	Unit/t	214748.3647	Upper speed limit • Read only • Is calculated from the *dnUpperLimit_s* speed signal.
C04302/2 As of library V02.02.xx.xx	-214748.3647	Unit/t	214748.3647	Lower speed limit • Read only • Is calculated from the *dnLowerLimit_s* speed signal.
C04303				Limitation enabled • Display of the *bEnable* input signal.
	0	Speed limitation is deactivated.		
	1	Speed limitation is activated.		
C04304				Limitation active • Display of the output signal *bLimit*.
	0	Speed output signal is not limited.		
	1	Speed output signal is limited.		
C04305	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.
C04306 As of library V02.02.xx.xx	String of digits			Speed unit • Read only
C04307 As of library V02.02.xx.xx	-214748.3647	Unit/t	214748.3647	Reference speed • Read only • For input orientation for speed entries.

5.130 L_SdMotorPot - motor potentiometer

Function library:	LenzeServoDrive	**FB is available as of library V02.04.xx.xx!**
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel
Required retain memory:	212 bytes	

This FB replaces a hardware motor potentiometer and can be used as an alternative setpoint source controlled via two inputs.

- The signal output is effected via a ramp function generator with linear ramps.
- The acceleration and deceleration time and the upper and lower limit value can be parameterised independently of each other.
- The behaviour after mains switching can be parameterised; in the Lenze setting the last output value before the mains are switched off is re-established.
- The behaviour in the case of deactivation of the motor potentiometer function can also be parameterised.



Inputs

Identifier/data type	Information/possible settings	
bUp BOOL	Increase output value	
	TRUE	With the acceleration time set in C04361/1, *dnOut_n* is increased until the upper limit value has been reached (C04360/1). • If the input *bDown* is set to TRUE at the same time, *dnOut_n* is not changed.
bDown BOOL	Decrease output value	
	TRUE	With the deceleration time set in C04361/2, *dnOut_n* is decreased until the lower limit value has been reached (C04360/2). • If the input *bUp* is set to TRUE at the same time, *dnOut_n* is not changed.
bDisable BOOL	Deactivate motor potentiometer function • This input has the second highest priority after *bLoadStartValue*. • The behaviour of the output value *dnOut_n* in the case of deactivation can be selected in C04362; in the Lenze setting the last output value is maintained.	
	TRUE	Motor potentiometer function is deactivated.
bLoadStartValue BOOL	Load motor potentiometer with starting value • This input has the highest priority.	
	TRUE	*dnOut_n* is set to the starting value *dnStartValue_n*. • If the starting value is beyond the limit values parameterised, the output value *dnOut_n* is limited to the corresponding limit value.
dnStartValue_n DINT	Starting value in [%]	

Outputs

Identifier/data type	Value/meaning	
dnOut_n DINT	Output value in [%] • Display parameter: C04368	
bLimitActive BOOL	Status signal "Output value is limited"	
	TRUE	The output value *dnOut_n* lies on a limit value or lies beyond the limit values and is directed to the nearest limit value.

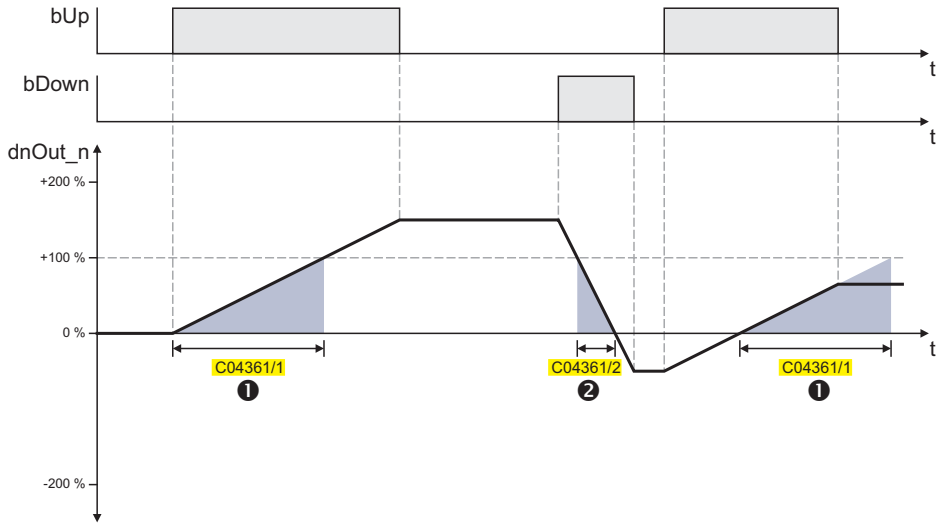
Parameter

Parameter	Possible settings			Information
C04360/1	-200.00	%	200.00	Upper limit value • Initialisation: 200.00 %
C04360/2	-200.00	%	200.00	Lower limit value • Initialisation: -200.00 %
C04361/1	0.001	s	999.999	Acceleration time • Relating to a change of the output value by 100 %. • Initialisation: 10 s
C04361/2	0.001	s	999.999	Deceleration time • Relating to a change of the output value by 100 %. • Initialisation: 10 s
C04362	The output value *dnOut_n*			Deactivation function • Behaviour of *dnOut_n* if *bDisable* is set to TRUE.
	0	...maintains its last value.		Lenze setting
	1	...is increased to 0 % with the deceleration time or increased with the acceleration time.		
	2	...is increased to the lower limit value with the deceleration time.		
	3	...is increased to the upper limit value with the acceleration time.		
	11	...is set to 0 % immediately.		**Important for the emergency stop function**
	12	...is set to the lower limit value immediately.		
	13	...is set to the upper limit value immediately.		
C04363	The output value *dnOut_n*			Initialisation function • Behaviour of *dnOut_n* after mains connection.
	0	...is set to its last value before mains switch-off.		Lenze setting
	1	...is set to 0 %.		
	2	...is set to the lower limit value.		
	3	...is set to the upper limit value.		
C04368	-200.00	%	200.00	Output setpoint • Display of the *dnOut_n* output value.

Parameter	Possible settings		Information
C04369	Bit coded:		Status • Bits that are not itemised are reserved for future extensions.
	bits5	Starting value loaded is limited	
	bits6	Upper limit value reached	
	bits7	Lower limit value reached	
	bits19	Lower limit value > upper limit value	
	bits31	General error	

5.130.1 Acceleration and deceleration time

The acceleration and deceleration time can be parameterised independently of each other in C04361/1 and C04361/2 and relate to a change in the output value *dnOut_n* by 100 %:



❶ C04361/1: acceleration time
❷ C04361/2: deceleration time

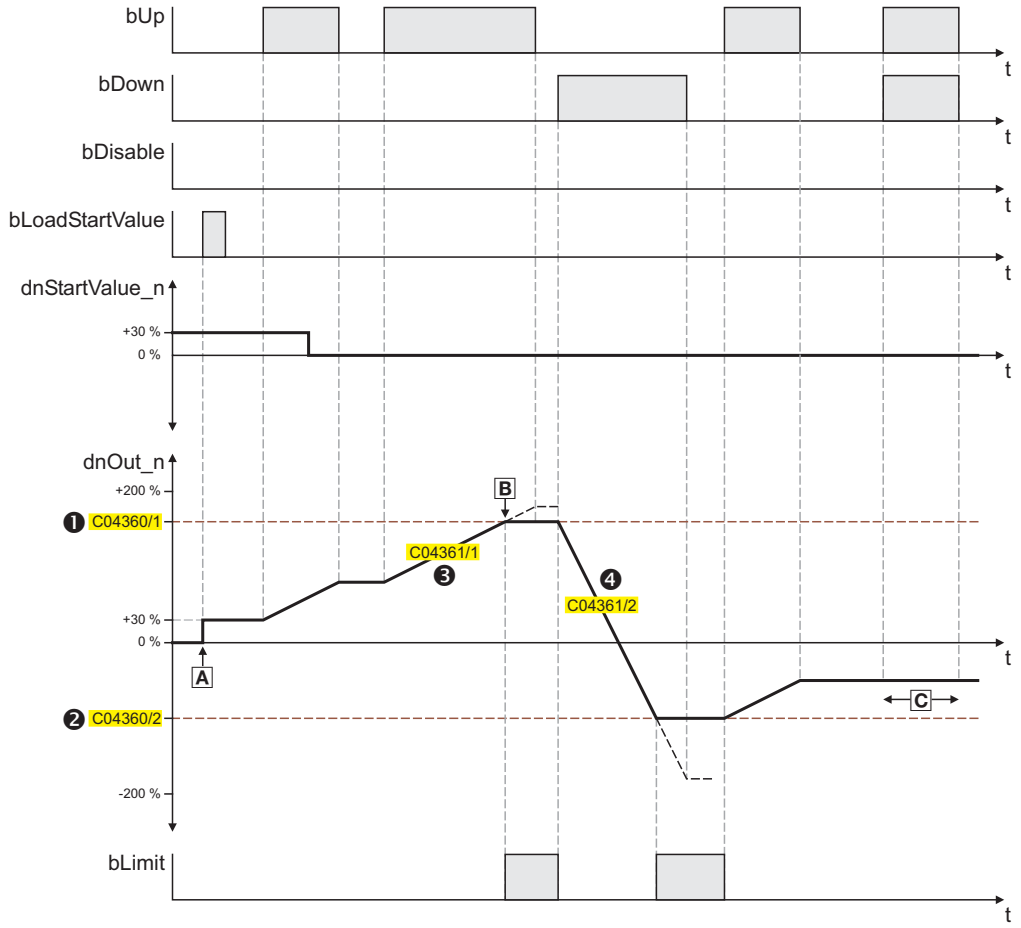
[5-100] Definition of the acceleration and deceleration time

5.130.2 Control signals

bUp	bDown	bDisable	bLoadStartValue	Function
FALSE	FALSE	FALSE	FALSE	*dnOut_n* remains unchanged.
TRUE	FALSE	FALSE	FALSE	**Increase output value** With the acceleration time set in C04361/1, *dnOut_n* is increased until the upper limit value has been reached (C04360/1).
FALSE	**TRUE**	FALSE	FALSE	**Decrease output value** With the deceleration time set in C04361/2, *dnOut_n* is decreased until the lower limit value has been reached (C04360/2).
TRUE	TRUE	FALSE	FALSE	*dnOut_n* remains unchanged.

bUp	bDown	bDisable	bLoadStartValue	Function
-	-	**TRUE**	FALSE	**Deactivate motor potentiometer function** *dnOut_n* behaves according to the deactivation function set in C04362.
-	-	-	**TRUE**	**Load starting value** *dnOut_n* is set to the value pending at input *dnStartValue_n*. • If the starting value is beyond the limit values parameterised, the output value *dnOut_n* is set to the corresponding limit value.

5.130.3 Signal characteristics

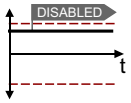
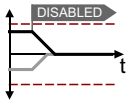
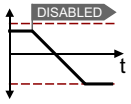
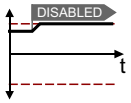
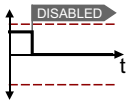
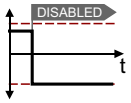
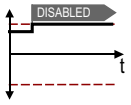


A Starting value is loaded
B Output value is limited
C Output value remains unchanged, as *bUp* and *bDown* are set to TRUE at the same time.
❶ C04360/1: upper limit value
❷ C04360/2: lower limit value
❸ C04361/1: acceleration time
❹ C04361/2: deceleration time

[5-101] Signal characteristics

5.130.4 Deactivation function

If the control input *bDisable* is set to TRUE, the control inputs *bUp* and *bDown* are deactivated and the output value *dnOut_n* is changed according to the deactivation function set in C04362:

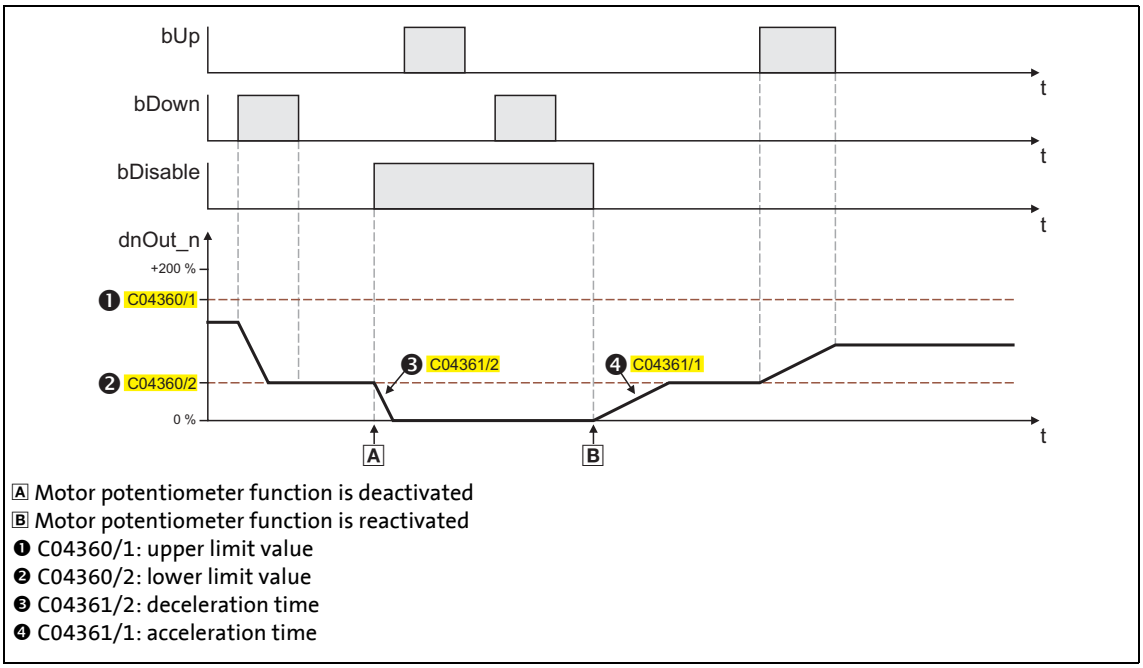
C04362		Deactivation function
0		*dnOut_n* maintains its last value.
1		*dnOut_n*is increased to 0 % with the deceleration time or increased with the acceleration time.
2		*dnOut_n* is decreased to the lower limit value (C04360/2) with the deceleration time (C04361/2).
3		*dnOut_n* is increased to the upper limit value (C04360/1) with the acceleration time (C04361/1).
11		*dnOut_n* is set to 0 % immediately.
12		*dnOut_n* is set to the lower limit value (C04360/2) immediately.
13		*dnOut_n* is set to the upper limit value (C04360/1) immediately.

Note!

If the motor potentiometer function is reactivated and *dnOut_n* is beyond the limit values parameterised, *dnOut_n* is lead to the nearest limit value under consideration of the acceleration/deceleration time.

This process is independent of the control signals *bUp* and *bDown*.

Example: Leading the output value to 0 % at deactivation



A Motor potentiometer function is deactivated
B Motor potentiometer function is reactivated
❶ C04360/1: upper limit value
❷ C04360/2: lower limit value
❸ C04361/2: deceleration time
❹ C04361/1: acceleration time

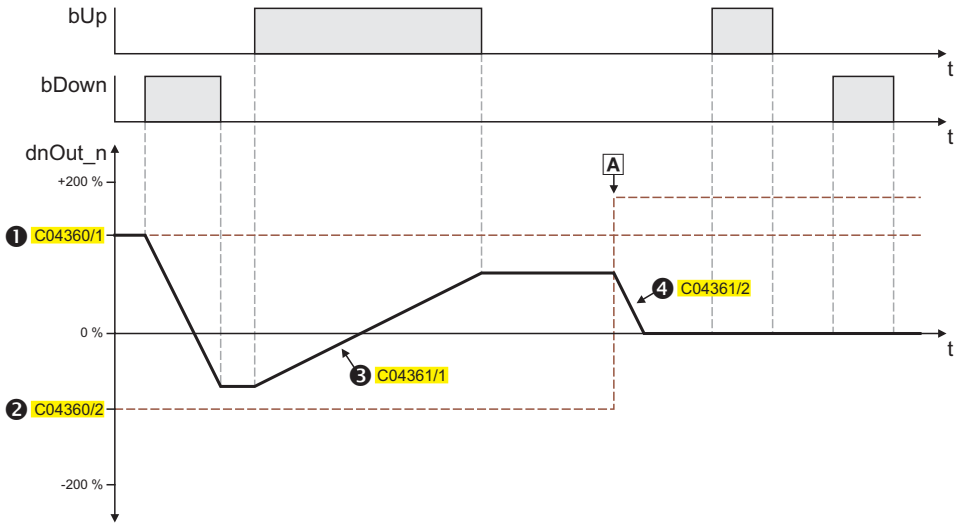
[5-102] Example: signal characteristics at deactivation function C04362 = 1

A. By setting the control input *bDisable* to TRUE, the motor potentiometer function is deactivated.
   - When the deactivation function C04362 = 1 is set, *dnOut_n* is decreased to 0 % under consideration of the deceleration time.

B. By resetting the control input *bDisable* to FALSE, the motor potentiometer function is reactivated.
   - As the lower limit value in the example is set greater 0 % and hence *dnOut_n* is beyond the limit values, *dnOut_n* is first lead back to the lower limit value by means of the acceleration time.

5.130.5 Behaviour for incorrect parameterisation of the limit values

If the upper limit value is parameterised smaller than the lower limit value, a bit coded error message is output via the display code C04369. As a response to this incorrect parameterisation, the control inputs *bUp* and *bDown* are deactivated and the output value *dnOut_n* is lead to 0 % with the deceleration/acceleration time set.



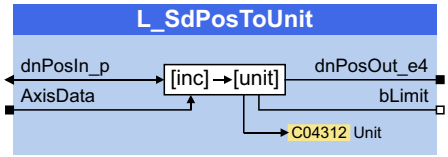
A Incorrect parameterisation (lower limit value > upper limit value)
❶ C04360/1: upper limit value
❷ C04360/2: lower limit value
❸ C04361/1: acceleration time
❹ C04361/2: deceleration time

[5-103] Example: signal characteristics for incorrect parameterisation of the limit values

5.131 L_SdPosToUnit - position conversion

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts a position defined in the internal unit [inc] into a position in the real unit of the machine.



Inputs

Identifier/data type	Information/possible settings
dnPosIn_p DINT	Position in [inc]
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnPosOut_e4 DINT	Position in [unit] • Output in "e4" format (fixed point with four decimal positions)	
bLimit BOOL	Status "Output signal is limited"	
	TRUE	The output signal is limited to the displayable value range or the *dnPosIn_p* input position is located beyond the software limit positions of the connected machine parameters (*AxisData*).

Parameter

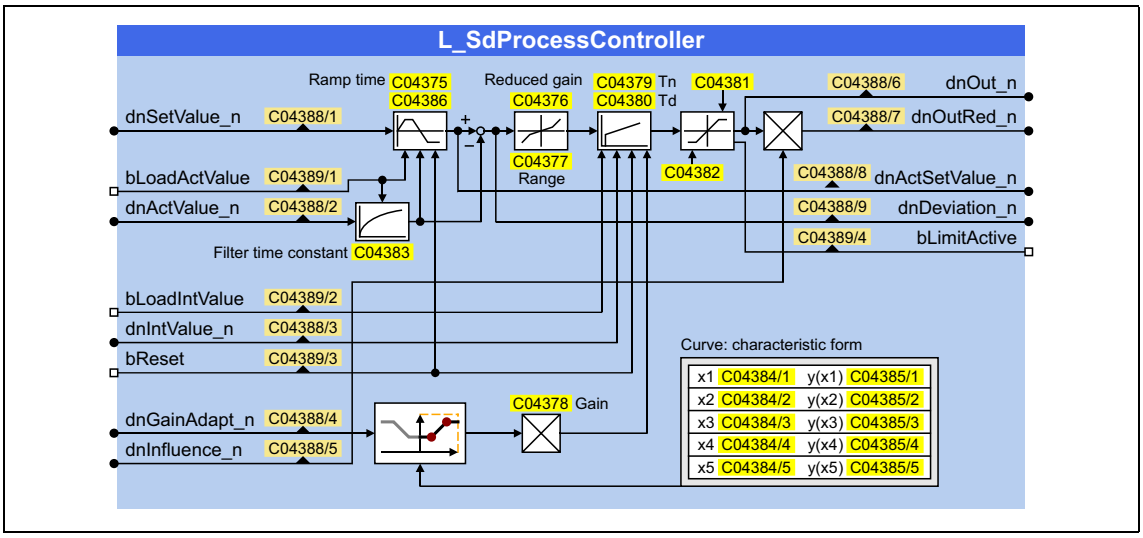
Parameter	Possible settings	Info
C04312	String from *AxisData*	Position unit • Read only

5.132 L_SdProcessController - PID controller with limitation

Function library:	LenzeServoDrive	**FB is available as of library V02.05.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This function block provides a comprehensive PID controller with setpoint and actual value processing as well as characteristic function.

- The function block basically consists of an interconnection of the following function blocks:
  - L_TbPT1Filter (📖 615)
  - L_SdRampGenerator (📖 504)
  - L_TbCurve (📖 565)
  - L_TbPIController (📖 613)
  - L_TbDifferentiate (📖 582)
- The value output to the output *dnOut_n* is internally limited to the limit range parameterised in C04381 and C04382 (default setting ±200 %).
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings	
dnSetValue_n DINT	Scaled process setpoint • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 • Display parameter: C04388/1	
bLoadActValue BOOL	Initialise ramp function generator and PT1 filter • Display parameter: C04389/1	
	FALSE	The ramp function generator runs with the acceleration/deceleration time set in C04375 from the actual value loaded via *dnActValue_n* to the setpoint applied to *dnSetValue_n*. • In the Lenze setting, the acceleration/deceleration time (C04375) is set to zero.
	TRUE	The *dnActValue_n* actual value is loaded into the ramp function generator and PT1 filter. The current system deviation is zero.
dnActValue_n DINT	Scaled actual process value for the ramp function generator and PT1 filter • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 • Display parameter: C04388/2	

Identifier/data type	Information/possible settings	
bLoadIntValue BOOL	Loading of the integrator • Display parameter: C04389/2	
	TRUE	Set the integral action component of the PID controller to the value applied to the input *dnIntValue_n*.
dnIntValue_n DINT	Scaled value on which the integral action component of the PID controller is adjusted by setting *bLoadIntValue* to TRUE. • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 • Display parameter: C04388/3	
bReset BOOL	Reset PID controller to zero • Display parameter: C04389/3	
	FALSE	PID controller is activated.
	TRUE	PID controller is reset to zero.
dnAdaptGain_n DINT	Scaled value for adapting the controller gain • The *dnAdaptGain_n* input value is first evaluated via a parameterisable characteristic function and then adapted to the controller gain (C04378). • In the characteristic default setting (y = 100 %), *dnAdaptGain_n* is ineffective. The controller gain is directly effective (C04378). • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 • Display parameter: C04388/4	
dnInfluence_n DINT	Scaled value for evaluating the *dnOut_n* output signal • The evaluated *dnOut_n* output signal is output at *dnOutRed_n*. • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 • Display parameter: C04388/5	

Outputs

Identifier/data type	Value/meaning	
dnOut_n DINT	Output signal • Is internally limited to the limit range parameterised in C04381 and C04382 (default setting ±200 %). • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 • Display parameter: C04388/6	
dnOutRed_n DINT	Output signal evaluated via *dnInfluence_n* input signal • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 • Display parameter: C04388/7	
dnActSetValue_n DINT	Current setpoint of the ramp function generator • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 • Display parameter: C04388/8	
dnDeviation_n DINT	System deviation • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 • Display parameter: C04388/9	
bLimitActive BOOL	Status signal "Limitation active" • Display parameter: C04389/4	
	TRUE	The output signal is limited.

Parameter

Parameter	Possible settings			Information
C04375	0.000	s	1000.000	Acceleration/deceleration ramp function generator • Initialisation: 0.000 s • Alternatively, it is possible from library V02.09.xx.xx to set a separate deceleration time in C04386. In the Lenze setting, the setting in C04386 is used as the deceleration time as usual.
C04376	0.00	%	100.00	Gain of the system deviation in the area of reduced sensitivity • Initialisation: 0.00 %
C04377	0.00	%	100.00	Area of system deviation with reduced gain/sensitivity • Initialisation: 0.00 %
C04378	0.0000		214748.3647	Controller gain • Setting 0.0000 resets the PID controller. • Initialisation: 1.0000
C04379	0.001	s	1000.000	Controller reset time • Setting 1000.000 s deactivates the I component. • Initialisation: 1.000 s
C04380	0.000	s	1000.000	Controller rate time • Setting 0.000 s deactivates the D component. • Initialisation: 0.000 s
C04381	0.00	%	200.00	Positive output limit value • Initialisation: 200.00 %
C04382	-200.00	%	0.00	Negative output limit value • Initialisation: -200.00 %
C04383	0.001	s	60.000	Filter time constant - actual value • With the setting 0.001 s the filter is not active. • Initialisation: 0.001 s
C04384/1...5	0.00	%	200.00	X values of the characteristic function • Subcodes 1 ... 5 correspond to the point values X1 ... X5. • Initialisation: X1, X3 ... X5 = 0.00 % X2 = 100.00 %
C04385/1...5	0.00	%	200.00	Y values of the characteristic function • Subcodes 1 ... 5 correspond to the point values Y1 ... Y5. • Initialisation: X1, X2 = 100.00 % X3 ... X5 = 0.00 %

Parameter	Possible settings			Information
C04386 From library V02.09.xx.xx	0.000	s	1000.001	Ramp function generator deceleration time • Only effective if ≠ 1000.001 s • Given a Lenze setting of 1000.001 s, the setting in C04375 is used as the deceleration time as usual. • Initialisation: 1000.001 s
C04388/1	-200.00	%	200.00	Process setpoint • Display of the *dnSetValue_n* input signal.
C04388/2	-200.00	%	200.00	Actual process value • Display of the *dnActValue_n* input signal.
C04388/3	-200.00	%	200.00	Load value - PID controller I component • Display of the *dnIntValue_n* input signal.
C04388/4	-200.00	%	200.00	Adaptation - controller gain • Display of the *dnAdaptGain_n* input signal.
C04388/5	-200.00	%	200.00	Evaluation of output signal • Display of the *dnInfluence_n* input signal.
C04388/6	-200.00	%	200.00	Setpoint at the output • Display of the *dnOut_n* output signal.
C04388/7	-200.00	%	200.00	Evaluated setpoint at the output • Display of the *dnOutRed_n* output signal.
C04388/8	-200.00	%	200.00	Current setpoint of ramp function generator • Display of the *dnActSetValue_n* output signal.
C04388/9	-200.00	%	200.00	System deviation • Display of the *dnDeviation_n* output signal.
C04389/1				Initialise ramp function generator and PT1 filter • Display of the *bLoadActValue* input signal.
	0	FALSE		
	1	TRUE		
C04389/2				Loading of the integrator • Display of the *bLoadIntValue* input signal.
	0	FALSE		
	1	TRUE		
C04389/3				Reset PID controller to zero • Display of the *bReset* input signal.
	0	FALSE		
	1	TRUE		
C04389/4				Limitation active • Display of the *bLimitActive* output signal.
	0	FALSE		
	1	TRUE		

5.132.1 Controller characteristic

The PID controller dynamics is parameterised according to the V_p gain, the T_n reset time, and the T_d rate time.

- In the Lenze setting, the function block operates as a PI controller and the D component is deactivated.

Gain (P component)

The controller gain is set in C04378.

- Setting "0.0000" resets the controller.
- The controller gain can be adapted via the *dnGainAdapt_n* input and a parameterisable characteristic function. ▸ Adaptation of the controller gain via characteristic function (📖 501)

Reset time (I component)

The reset time is set in C04379.

- Setting "1000.000 s" deactivates the I component.
- By setting the *bLoadIntValue* input to TRUE, the I component of the controller can be set to the value applied to the *dnIntValue_n* input.

Rate time (D component)

The rate time is set in C04380.

- Setting "0.000 s" deactivates the D component (Lenze setting). The PID controller thus becomes a PI controller or P controller if the I component is deactivated as well.

5.132.2 Adaptation of the controller gain via characteristic function

The controller gain can be adapted via the *dnGainAdapt_n* input and a parameterisable characteristic function.

- In the characteristic default setting (y = 100 %), *dnAdaptGain_n* is ineffective. The controller gain is directly effective (C04378).

$$P = C04378$$

- Only with the characteristic setting x1 = 0, y1 = 0, x2 = 100 % and y2 = 100 %, *dnAdaptGain_n* directly acts on the controller gain:

$$P = dnGainAdapt\_n \cdot C04378$$

- By changing the characteristic, the influence of *dnAdaptGain_n* on the controller gain can be evaluated in percent:

$$P = f(dnGainAdapt\_n) \cdot C04378$$

Selection of the characteristic

The characteristic can consist of up to five points which are defined by the subcodes of C04384 and C04385.

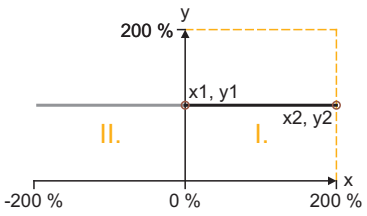
- Between the points a linear interpolation takes place.
- The characteristic is selected for the I. quadrant and mirrored at the Y axis.
- The same subcodes of C04384 and C04385 correspond to a pair of variates/point (x_n, y_n).

	1	2	2	2	5
Y	C04384/**1**	C04384/**2**	C04384/**3**	C04384/**4**	C04384/**5**
x	C04385/**1**	C04385/**2**	C04385/**3**	C04385/**4**	C04385/**5**

- The first pair of variates (C04384/1 and C04385/1) is always valid.
- The X values of the characteristic must be entered in ascending order (X1 < X2 < ... < X5).
- If the ascending chain of x values is interrupted, this corresponds to the end of the characteristic (see the following examples).
- Based on the last valid pair of variates, an extrapolation is executed to the end of the x value range (±200 %) on both sides.

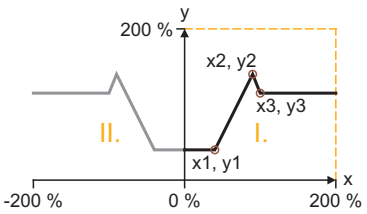
Characteristic examples

	1	2	3	4	5
x	0 %	100 %	0 %	0 %	0 %
Y	100 %	100 %	0 %	0 %	0 %



[5-1] Example 1 (Lenze setting)

	1	2	3	4	5
x	40 %	90 %	100 %	0 %	0 %
Y	40 %	140 %	115 %	0 %	0 %



[5-2] Example 2

5.132.3 System deviation in the area of reduced sensitivity

A reduced controller dynamics with low system deviations causes a positive influence on the damping behaviour of the control loop.

- C04377 serves to set a tolerance window, in which the system deviation is transferred to the controller with a slight gain.
- C04376 serves to set the required gain reduction in the defined tolerance window in percent.

5.132.4 Controller influence

The *dnInfluence_n* input serves to evaluate the *dnOut_n* controller output in percent. The evaluated output signal is provided at *dnOutRed_n*.

5.132.5 Ramp function generator

The *dnSetValue_n* input setpoint can be lead via a ramp function generator with linear ramps to prevent setpoint step-changes at the input.

The following applies up to and including library V02.08.xx.xx:

- The acceleration/deceleration times are set via C04375.

The following applies from library V02.09.xx.xx:

- The acceleration and deceleration times are set in C04375 and C04386.
  - If C04386 = "1000.001 s" (Lenze setting), the setting in C04375 is used for both the acceleration and the deceleration time, which corresponds to the behaviour so far (symmetrical acceleration/deceleration times).
  - If C04386 <> "1000.001 s", the setting in C04375 is used as the acceleration time and a separate setting in C04386 is used as the deceleration time.
- The acceleration time describes the time necessary to reach 100 %.
- The setting is done according to the formula:

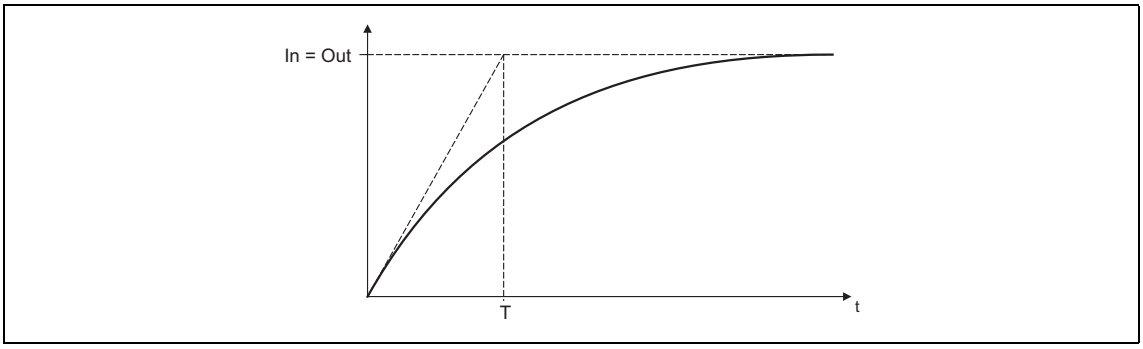
$$\frac{1\ \text{s} \cdot 100\ \%}{\text{C04375}\ [\text{s}]}$$

- In the Lenze setting, the ramp function generator is deactivated (C04375 = 0.000 s).
- By setting the *bLoadActValue* input to TRUE, the ramp function generator can be initialised with the value applied to the *dnActValue_n* input. This causes the system deviation to be zero.

5.132.6 Low pass in actual value path

The *dnActValue_n* actual value loaded by setting *bLoadActValue* to TRUE can be lead via a PT1 filter.

- The rate time is set in C04383.
- In the Lenze setting, the filter is deactivated (C04383 = 0.001 s).



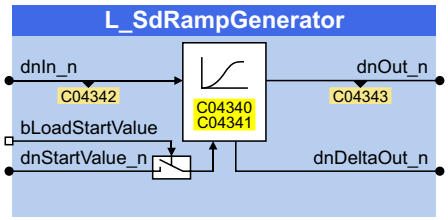
[5-104] PT1 filter function

5.133 L_SdRampGenerator - ramp function generator - S-ramp

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB is a ramp function generator with S-shaped ramps for limiting the increase of analog signals over time. The S-shape of the ramps results from a trapezoidal acceleration.

- The ramp function generator has a setting function for directly loading the value into the internal ramp generator.
- The acceleration/deceleration times are set via C04340.
- An S-ramp time for a jerk-free acceleration can be set in C04341.
- At the output *dnDeltaOut_n* the gradient dy/dt of the output signals *dnOut_n* is shown. This signal can for instance be used for torque feedforward control after a gain via the L_TbGainLim FB.
- An advanced version, the L_SdRampGeneratorAny FB, is available from library V02.09.xx.xx. This FB can be used to also process speeds (_s) and to set separate acceleration and deceleration times.



Inputs

Identifier/data type	Information/possible settings	
dnIn_n DINT	Scaled input signal	
bLoadStartValue BOOL	Initialising ramp function generator	
	FALSE	The ramp function generator runs with the Ti times set from the value loaded via *dnStartValue_n* to the value at *dnIn_n*.
	TRUE	At the output *dnOut_n* *dnStartValue_n* is output, *dnDeltaOut_n* remains on 0 %.
dnStartValue_n DINT	Scaled starting value for the ramp function generator • Acceptance when *bLoadStartValue* = TRUE	

Outputs

Identifier/data type	Value/meaning
dnOut_n DINT	Scaled output signal • Internally limited to ±200 %
dnDeltaOut_n DINT	Scaled acceleration of the ramp function generator • Time reference: 62.5 µs, i. e. 0.01 % correspond to a change of 160 % in the output signal within 1 s. • Internally limited to ±200 %

Parameter

Parameter	Possible settings			Information
C04340	0.001	s	1000.000	Acceleration / deceleration time • Initialisation: 1.000 s
C04341	0.001	s	10.000	S-ramp time • Initialisation: 0.100 s
C04342 As of library V02.02.xx.xx	-200.000	%	200.000	Setpoint at the input • Display of the *dnIn_n* input signal.
C04343 As of library V02.02.xx.xx	-200.000	%	200.000	Setpoint at the output • Display of the *dnOut_n* output signal.

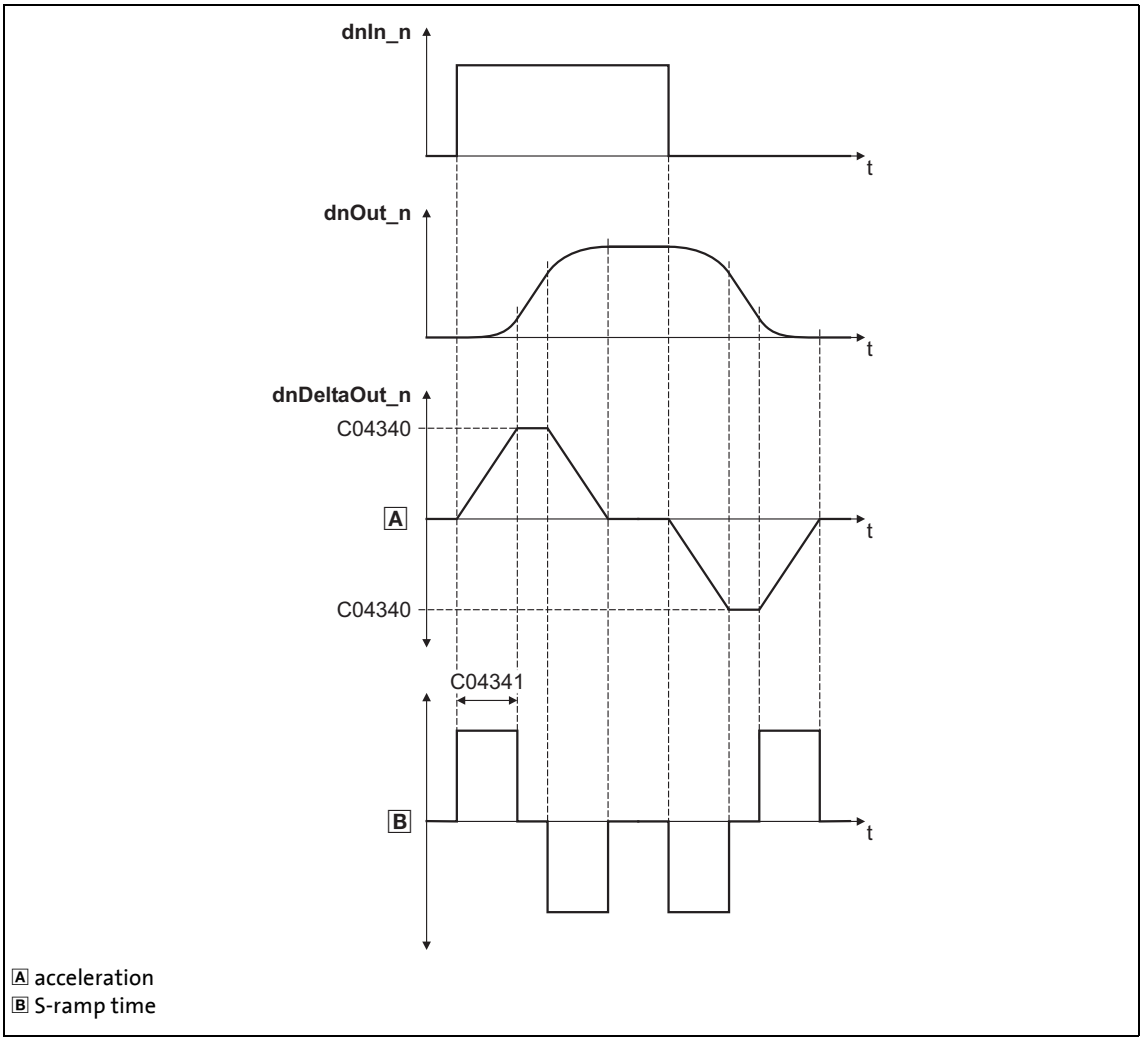
5.133.1 Loading the ramp function generator

By setting *bLoadStartValue* to TRUE the ramp function generator is loaded with the signal at *dnStartValue_n*.

- This value is accepted immediately and output to *nOut_n*. No s-shape ramp-up or ramp-down takes place.
- As long as *bLoadStartValue* = TRUE , the ramp function generator is inhibited.

5.133.2 Acceleration and jerk

The maximum acceleration and the jerk can be adjusted separately.



A acceleration
B S-ramp time

[5-105] Line diagram

Acceleration / deceleration time

The acceleration and deceleration time are set in C04340.

- The acceleration time describes the time necessary to reach 100 %.
- The setting is done according to the formula:

$$\frac{1\ \text{s} \cdot 100\ \%}{\text{C04340}\ [\text{s}]}$$

S-ramp time

An S-ramp time for a jerk-free acceleration of the drive can be set in C04341.

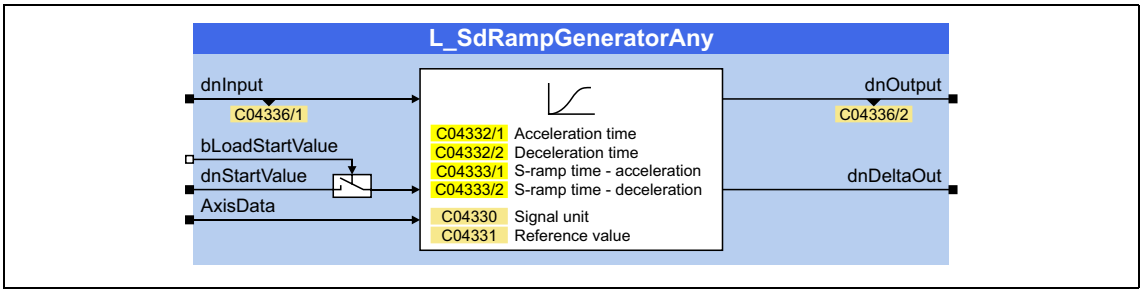
- The jerk is entered in [s] until the ramp function generator operates with the maximum acceleration.

5.134 L_SdRampGeneratorAny - ramp function generator - S-ramp

Function library:	LenzeServoDrive	**FB available from library V02.09.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB is a ramp function generator with S-shaped ramps for limiting the increase of analog signals over time. The S-shape of the ramps results from a trapezoidal acceleration.

- This FB is an advanced version of the L_SdRampGenerator FB:
  - Depending on the interconnection of AxisData, setpoints can be defined as scaled signals (_n) or speeds (_s).
  - Separate acceleration and deceleration times can be set via C04332/1 and C04332/2.
  - Separate S-ramp times for acceleration and deceleration to achieve jerk-free acceleration can be set in C04333/1 and C04333/2.
- The ramp function generator has a setting function for directly loading the value into the internal ramp generator.
- At the *dnDeltaOut* output, gradient dy/dt of the *dnOutput* output signal is provided. This signal can e.g. be used for torque feedforward control after being amplified via the L_TbGainLim FB.



Inputs

Identifier/data type	Information/possible settings
dnInput DINT	Input signal (scaled signal or speed) • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 (without *AxisData*) 15000 min^{-1} ≡ 2^{26} ≡ 67108864 (with *AxisData*)
bLoadStartValue BOOL	Initialising ramp function generator
	FALSE: The ramp function generator runs with the Ti times set from the value loaded via *dnStartValue* to the value at *dnInput*.
	TRUE: At the *dnOutput* output, *dnStartValue* is provided. *dnDeltaOut* remains at 0 % or 0 min^{-1}, respectively.
dnStartValue DINT	Starting value for the ramp function generator (scaled signal or speed) • Acceptance when *bLoadStartValue* = TRUE • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 (without *AxisData*) 15000 min^{-1} ≡ 2^{26} ≡ 67108864 (with *AxisData*)
AxisData	Machine parameters • If this input is not interconnected, the FB uses scaled signals. • If the FB receives machine parameters via this input, the FB uses speed signals instead. • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning
dnOutput DINT	Output signal • Scaling: 100 % ≡ 2^{30} ≡ 1073741824 (without *AxisData*) 15000 min^{-1} ≡ 2^{26} ≡ 67108864 (with *AxisData*)
dnDeltaOut DINT	Ramp function generator acceleration • Time reference: 62.5 µs

Parameter

Parameter	Possible settings			Information
C04330	String of digits			Display of the signal unit • "%" (without *AxisData*) or display of the application unit from *AxisData*.
C04331	0.0000		100.0000	Display of the reference value • 100.0000 (without *AxisData*) or reference speed from *AxisData*.
C04332/1	0.000	s	100.000	Acceleration time • The acceleration time refers to the time when the reference value is reached. • Setting as per the following formula: 1 s * reference / acceleration time [s] • Initialisation: 1.000 s
C04332/2	0.000	s	100.000	Deceleration time • Setting as per the following formula: 1 s * reference / deceleration time [s] • Initialisation: 1.000 s
C04333/1	0.000	s	10.000	S-ramp time - acceleration • The jerk is entered in [s] until the ramp function generator operates with the maximum acceleration. • Initialisation: 0.100 s
C04333/2	0.000	s	10.000	S-ramp time - deceleration • The jerk is defined as the time in [s] for the ramp function generator to reach max. deceleration. • Initialisation: 0.100 s
C04336/1	-214748.3647		214748.3647	Setpoint at the input • Read only
C04336/2	-214748.3647		214748.3647	Setpoint at the output • Read only



For a detailed functional description, see the L_SdRampGenerator FB.

5.135 L_SdRuntimeComp - runtime compensation

Function library:	LenzeServoDrive	**Do not use FB for new developments!** Replacement by L_SdDelayComp.
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

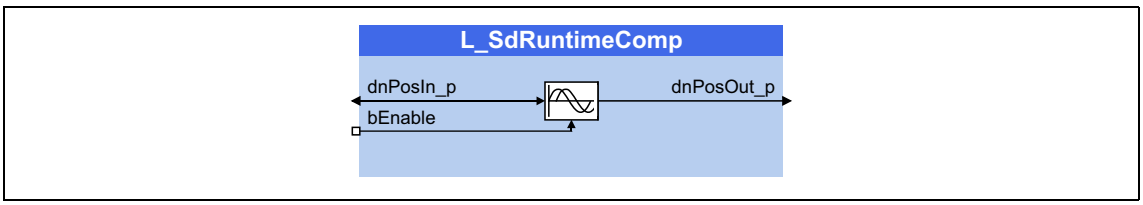
This FB is used for the runtime compensation of position signals that are, for instance, transmitted by means of a bus system.

- The output signal leads the input signal by the last position difference.
- The resulting rate time equals the task runtime of the program in which the FB is used.

Note!

The FB is not suitable for clocked (modulo) positions!

The FB L_SdDelayComp can be used for dead time compensation. (📖 453)



Inputs

Identifier/data type	Information/possible settings	
dnPosIn_p DINT	Position in [inc]	
bEnable BOOL	Activate compensation	
	TRUE	Compensation is activated.

Outputs

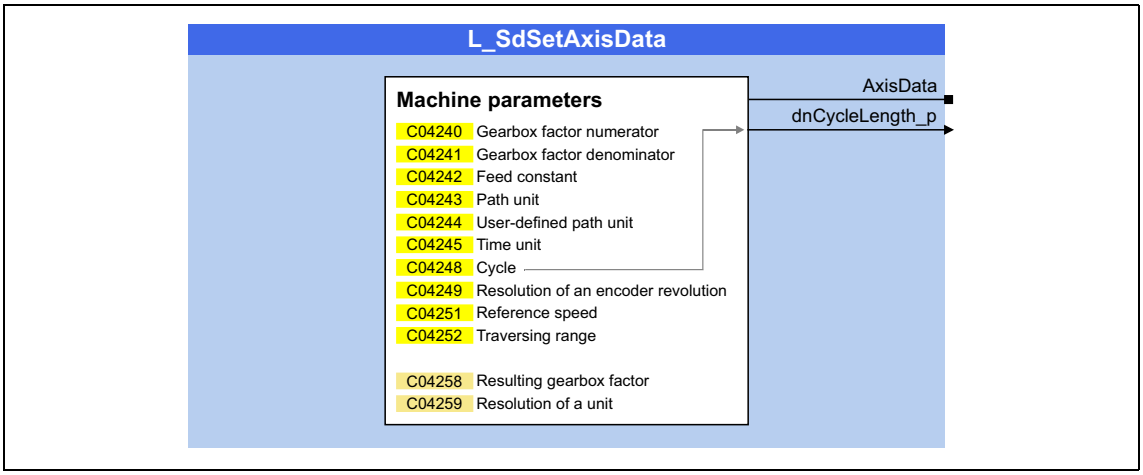
Identifier/data type	Value/meaning
dnPosOut_p DINT	Compensated position in [inc] • Leads the input signal *dnPosIn_p* by the last position difference.

5.136 L_SdSetAxisData - machine parameters

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to indicate the machine parameters of a master drive. The FB processes the machine parameters which you define with the physical units of the machine via parameters for the internal representation.

- The FB outputs a pointer to the data structure with the prepared machine parameters at the output *AxisData*.
- To transmit the machine parameters to another FB, which requires these data for internal calculations, the output *AxisData* must simply be connected to the input of the same name of the corresponding FB.



Outputs

Identifier/data type	Value/meaning
AxisData	Machine parameters • Pointer to a data structure the elements of which contain the machine parameters in the internal measuring system. • Via this "interface" the machine parameters can be provided for other FBs which require these data for internal calculations.
dnCycleLength_p DINT As of library V02.05.xx.xx	Cycle in [increments] • C04248 indicates the cycle in the real unit of the machine.

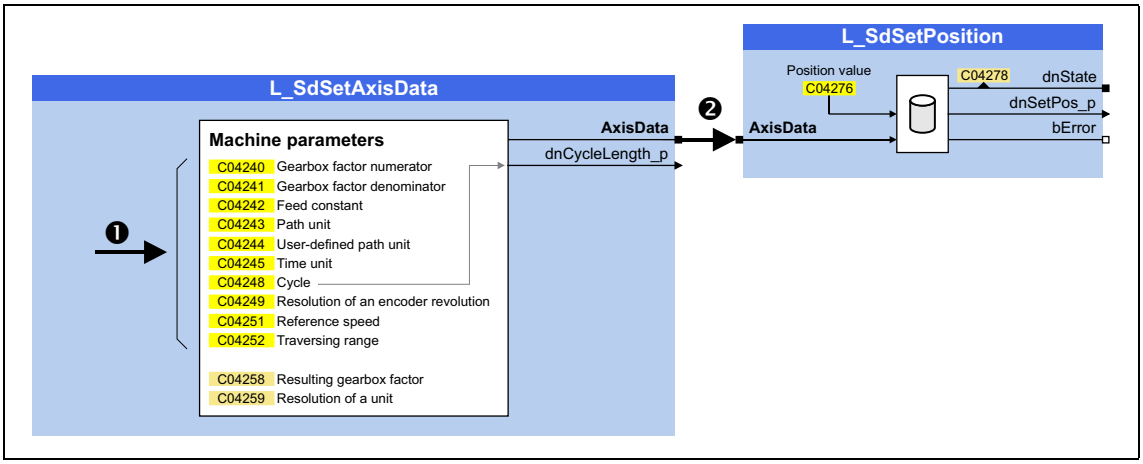
Parameter

Note!

Observe that the index of the codes listed in the following only applies to the first instance!

Parameter	Possible settings			Information
C04240	1		2147483647	Gearbox factor numerator • Initialisation: 1
C04241	1		2147483647	Gearbox factor denominator • Initialisation: 1
C04242	0.0001	unit/inc	214748.3647	Feed constant • Initialisation: 360 unit/inc
C04243				Path unit
	0	User-defined		Selection in C04244
	1	inc		
	2	µM		
	3	mm		
	4	M		
	5	inch		
	6	yard		
	7	°		Lenze setting
C04244	String of digits with max. 8 characters			User-defined path unit
C04245				Time unit • The time base unit is default set to "s" and cannot be changed!
	0	User-defined		
	1	ms		
	2	s		
	3	min		
	4	H		
C04246 As of library V02.07.xx.xx	String of digits with max. 16 characters			Axis name • Read only
C04248	0.0000	Unit	214748.3647	Cycle • Initialisation: 360 unit
C04249	10	Bit	24	Resolution of an encoder revolution • Initialisation: 16 bits
C04251	-214748.3647	Unit/t	214748.3647	Reference speed • Initialisation: 214748.3647 unit/t
C04252 As of library V02.00.xx.xx				Traversing range
	0	Unlimited		
	1	Limited		
	2	Modulo		Lenze setting
C04258	-2147483.647		2147483.647	Resulting gearbox factor • Read only
C04259	-214748.3647	Incr./unit	214748.3647	Resolution of a unit • Read only

5.136.1 Typical application



[5-106] Typical application

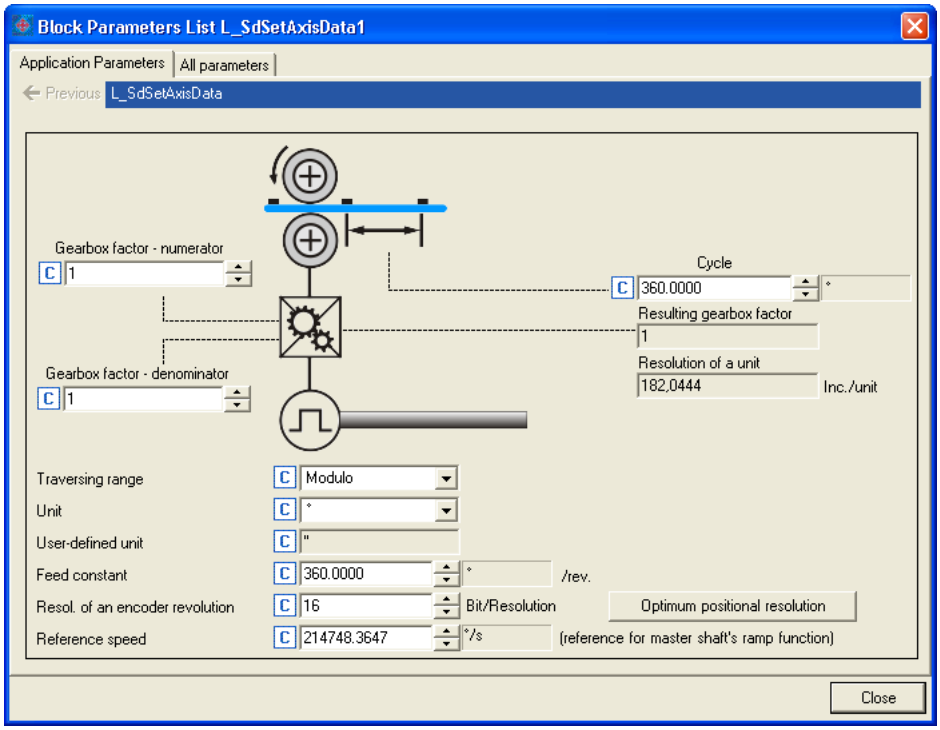
1. The machine data are specified with the physical units of the machine via the corresponding parameters. ❶
2. The FB converts the machine parameters into the internal representation and provides the data at the output *AxisData* as a data structure for transfer to other FBs. ❷
   - For library V01.xx the following applies: Always a modulo measuring system is output.
   - As of library V02.00 the following applies: The traversing range can be selected via C04252.

5.136.2 Selection and input of machine parameters

The following table lists absolutely necessary machine parameters:

Machine parameters	Setting	Unit	Information
Gearbox ratio - numerator	C04240	-	Input according to the gearbox nameplate.
Gearbox ratio - denominator	C04241	-	
Feed constant (F_c)	C04242	Unit	Enter the number in [unit] (e.g. [mm]), which is to be fed when the gearbox output revolves once.
Reference speed (v_{max})	C04251	Unit/t	This limit applies to the entire machine.
	Caution: The limit must be set in a way that it can be reached with max. motor speed: $V_{max} \leq n_{max} \cdot \frac{\text{Feed constant}}{60} \cdot \frac{\text{gearbox (denominator)}}{\text{gearbox (numerator)}}$ $\text{dwVmax} \leq \text{C00011} \cdot \frac{\text{C04242}}{60} \cdot \frac{\text{C04240}}{\text{C04241}}$		

The machine parameters can be easily entered via the parameterisation dialog of the FB **L_SdSetAxisData**.



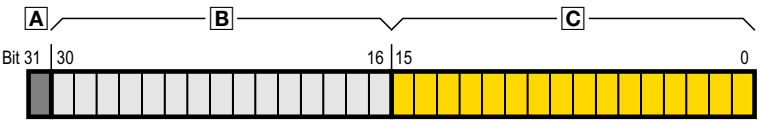
Tip!

The way how to detect the optimum resolution of the position values is described in the subchapter "Detecting the optimum resolution". (📖 516)

5.136.3 Resolution of an encoder revolution

C04249 serves to set the resolution - tailored to the application.

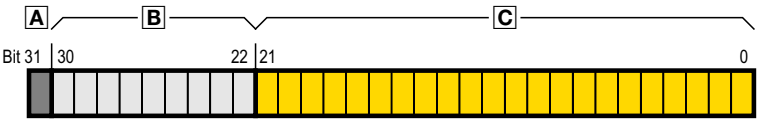
- The preset resolution of 16 bits/revolution is sufficient for standard applications.



A Sign bit
B Number of revolutions: 15 bits ≡ ±32767 revolutions
C Resolution of an encoder revolution: 16 bits ≡ 65536 increments/revolution

[5-107] Example: Standard resolution (16 bits/revolution)

- For higher-order applications, a higher resolution of the position values may lead to considerably improved control properties and positioning accuracies.
  - Finer resolution of the position targets ➞ improved positioning accuracy
  - Finer quantisation of the setpoints and actual values ➞ higher control quality
  - Higher loop gain can be adjusted ➞ less following errors
- However, a higher resolution results in a restricted number of encoder revolutions and only smaller traversing paths can be displayed.



A Sign bit
B Number of revolutions: 9 bits ≡ ±512 revolutions
C Resolution of an encoder revolution: 22 bits ≡ 4194304 increments/revolution

[5-108] Example: Higher resolution (22 bits/revolution) with restricted traversing range

Tip!

The way how to detect the optimum resolution of the position values is described in the following subchapter "Detecting the optimum resolution". (📖 516)

Note!

The position values (e.g. setpoints, actual values, parameters, ...) in the signal flow always use the set resolution, irrespective of the resolution the encoder supplies directly.

Resolution of the drive/motor

The machine parameters for the drive/motor are set via the drive interface (SB **LS_DriveInterface**).

Multi-axis systems

When being connected via the electrical shaft, the drive is at least provided with two measuring systems (master and slave).

- Each measuring system has an own resolution setting.
- The machine parameters (gearbox factors, feed constant, encoder resolution, and cycle) for the master measuring system or master value must be identical for all drives in the interconnection.

Technology applications "Electronic gearbox" and "Synchronism"

For both technology applications, the machine parameters of the master measuring system are defined on the *Application parameters* tab in the "Master value scaling" dialog level.

Electronic cam

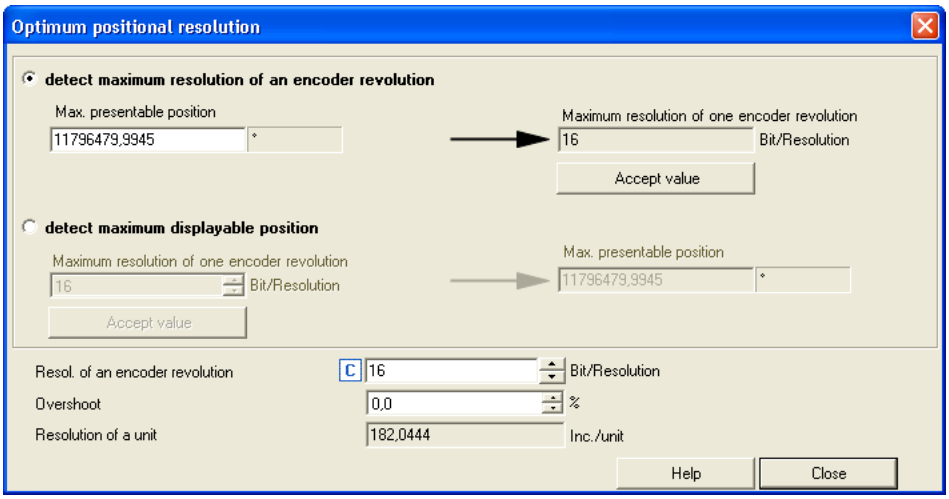
For electronic cams, the machine parameters of the master measuring system are defined on the *Measuring systems* tab of the electrical shaft.

5.136.3.1 Detecting the optimum resolution

How to detect the optimum resolution:

In the parameterisation dialog of the FB **L_SdSetAxisData**:

1. Set gearbox factors.
2. Set real unit of the machine.
3. Set feed constant.
4. Click **Optimum positional resolution**.
   - The *Optimum positional resolution* dialog box is displayed:



5. Enter the greatest position to be entered into a parameter during operation into the **Max. presentable position** input field.
   - If required, set an overshoot in the **Overshoot** input field for considering possibly occurring following errors (overshooting of the actual values).

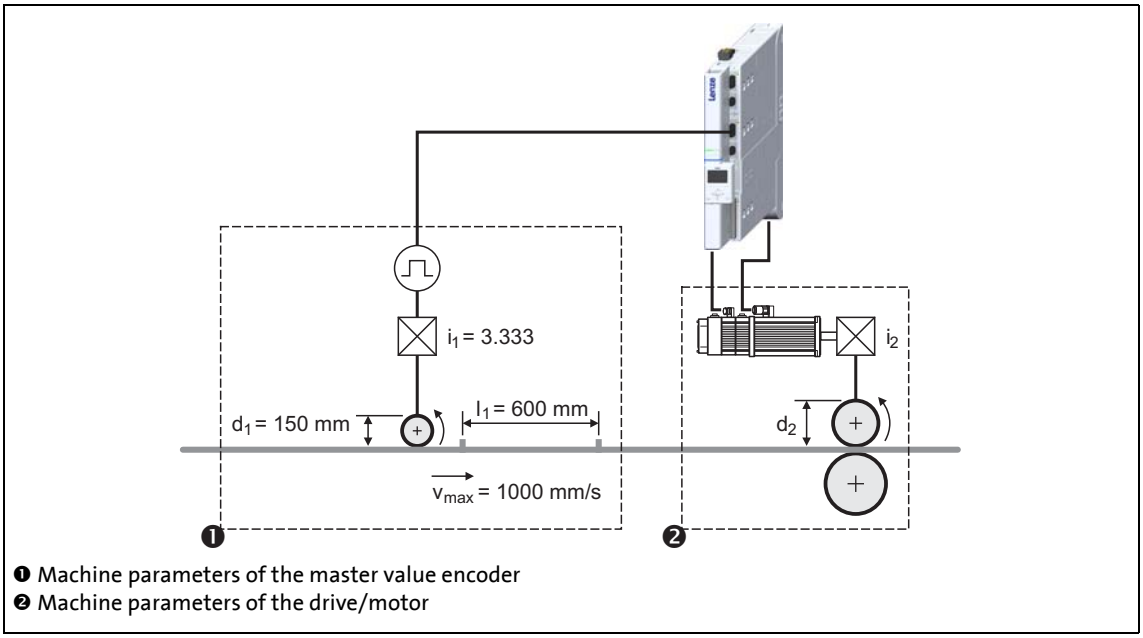
   After this, the maximum resolution for the entered position is displayed in the **Maximum resolution for encoder revolution** field.
6. Press **Accept value** to accept the resolution displayed in C04249.
7. Press **Close** to close the dialog box again.

Tip!

In order to display the max. presentable position for a predefined resolution, activate the second option **detect maximum displayable position**. Afterwards you can set the resolution for which the max. presentable position is to be displayed in the **Maximum resolution for encoder revolution** input field.

5.136.4 Example

The machine parameters of a master value encoder are to be displayed in the application for the purpose of master value processing.



❶ Machine parameters of the master value encoder
❷ Machine parameters of the drive/motor

[5-109] Schematic diagram of real master (master value encoder)

Settings

Machine parameters	Setting	Input	Information
Gearbox ratio - numerator	C04240	10	Enter gearbox ratio i_1 = 3.333 for the master value encoder as a quotient (numerator and denominator).
Gearbox ratio - denominator	C04241	3	
Feed constant	C04242	471.2389	Fc = d_1 * π = 150 mm * π = 471.2389 mm
Path unit	C04243	3	Unit = [mm]
Cycle	C04248	600	l_1 = 600 mm
Reference speed	C04251	1000	v_{max} = 1000 mm/s

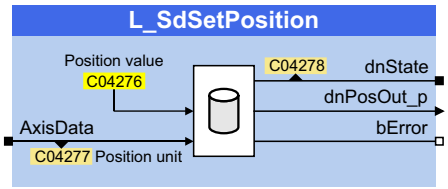
Tip!

The machine parameters for the drive/motor are set via the drive interface (SB **LS_DriveInterface**).

5.137 L_SdSetPosition - position conversion

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts a position which has been selected via C04276 in the real machine units into a position in [inc] and outputs it at *dnSetPosOut_p* for further processing within the FB interconnection.



Inputs

Identifier/data type	Information/possible settings
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510) • If this input is released, the following scaling applies: 1 motor revolution ≡ 2^{16}

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status	
	0	Ok - no error
	-12	Counter overflow of ±214748.3647 with position value (C04276). ▸ Monitoring for counter overflow (📖 519)
dnPosOut_p DINT	Position in [inc] • Scaling: 1 encoder revolution ≡ 2^{16} increments (or acc. to *AxisData*)	
bError BOOL	"Error" status signal	
	TRUE	The conversion was incorrect.

Parameter

Parameter	Possible settings			Information
C04276	-214000.0000	Unit	214000.0000	Position value • Initialisation: 0 unit
C04277 As of library V02.02.xx.xx	String of digits			Position unit • Read only
C04278	-2147483647		2147483647	Status • Read only

5.137.1 Monitoring for counter overflow

This function extension will be available from library V2.02!

If the machine parameters transferred to the FB via the *AxisData* input are changed afterwards (e.g. the gearbox factors), this may cause a counter overflow when the position value defined via C04276 is converted from the real unit to the internal unit.

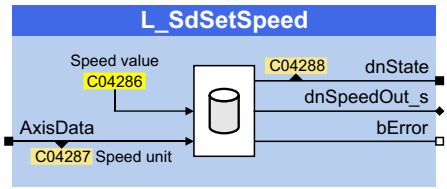
The FB reports an internal counter overflow due to a changed measuring system by the value "-12" at the *dnState* status output and activates a corresponding error message:

Error number		Error message in the logbook	Response
61603844	0x3AC0004	L_SdSetPosition:int.pos. overflow (LS_DriveInterface)	Error
61737217	0x3AE0901	L_SdSetPosition:int.pos. overflow (L_SdSetAxisData)	Error

5.138 L_SdSetSpeed - speed conversion

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts a velocity which has been selected via C04286 in the real machine units into a speed and outputs it at *dn'SpeedOut_s* for further processing in the FB interconnection.



Inputs

Identifier/data type	Information/possible settings
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status	
	0	Ok - no error
	-12	Counter overflow of ±214748.3647 with speed value (C04286). ▸ Monitoring for counter overflow (📖 521)
dnSpeedOut_s DINT	Speed output in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864	
bError BOOL	"Error" status signal	
	TRUE	The conversion was incorrect.

Parameter

Parameter	Possible settings			Information
C04286	-214748.3647	Unit/t	214748.3647	Speed value • Initialisation: 0 unit/t
C04287 As of library V02.02.xx.xx	String of digits			Speed unit • Read only
C04288	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.

5.138.1 Monitoring for counter overflow

This function extension will be available from library V2.02!

If the machine parameters transferred to the FB via the *AxisData* input are changed afterwards (e.g. the gearbox factors), this may cause a counter overflow when the speed value defined via C04286 is converted from the real unit to the internal unit.

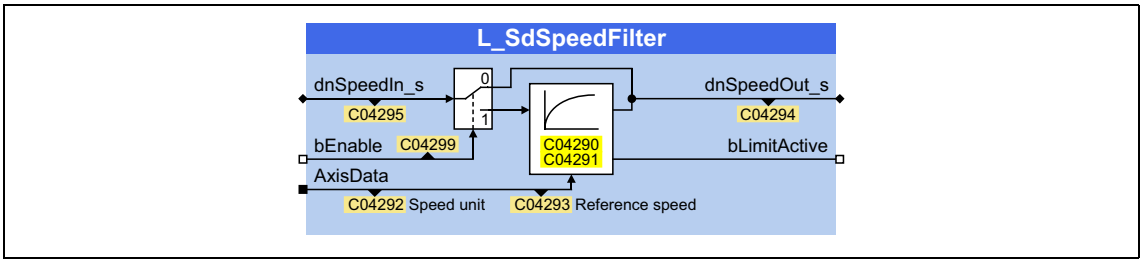
The FB reports an internal counter overflow due to a changed measuring system by the value "-12" at the *dnState* status output and activates a corresponding error message:

Error number		Error message in the logbook	Response
61603845	0x3AC0005	L_SdSetSpeed:int.speed overflow (LS_DriveInterface)	Error
61603973	0x3AC0085	L_SdSetSpeed:int.speed overflow (L_SdSetAxisData)	Error

5.139 L_SdSpeedFilter - speed signal delay

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB is used to filter or delay a speed signal according to the PT1 principle. The resulting position loss is compensated by an internal overflow buffer.



Inputs

Identifier/data type	Information/possible settings	
dnSpeedIn_s DINT	Speed selection in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864	
bEnable BOOL	Activate filter function	
	FALSE	Filter is deactivated. • Use these settings during the acceleration and deceleration phases to prevent a drifting of the position.
	TRUE	Filter is activated.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)	

Outputs

Identifier/data type	Value/meaning	
dnSpeedOut_s DINT	Filtered speed in [rpm]	
bLimitActive BOOL	Status signal "Overflow/underflow"	
	TRUE	An overflow/underflow has occurred.

Parameter

Parameter	Possible settings			Information
C04290	0.0000	Unit/s	214000.0000	Compensating speed • For compensating the phase differences. • Initialisation: 0.0010 unit/s
C04291	0.000	s	60.000	Filter time constant • Initialisation: 0.003 s
C04292 As of library V02.02.xx.xx	String of digits			Speed unit • Read only
C04293 As of library V02.02.xx.xx	-214748.3647	Unit/t	214748.3647	Reference speed • Read only • For input orientation for speed entries.
C04294	-214748.3647	Unit/t	214748.3647	Speed at the output • Read only • Calculated from the speed signal *dnSpeedOut_s*.
C04295	-214748.3647	Unit/t	214748.3647	Speed at the input • Read only • Calculated from the speed signal *dnSpeedIn_s*.
C04299				Filter is enabled • Read only
	0	Filter is not active		
	1	Filter is active		

5.139.1 Monitoring for counter overflow

This function extension will be available from library V2.02!

If the machine parameters transferred to the FB via the *AxisData* input are changed afterwards (e.g. the gearbox factors), this may cause a counter overflow when the speed value defined via C04290 is converted from the real unit to the internal unit.

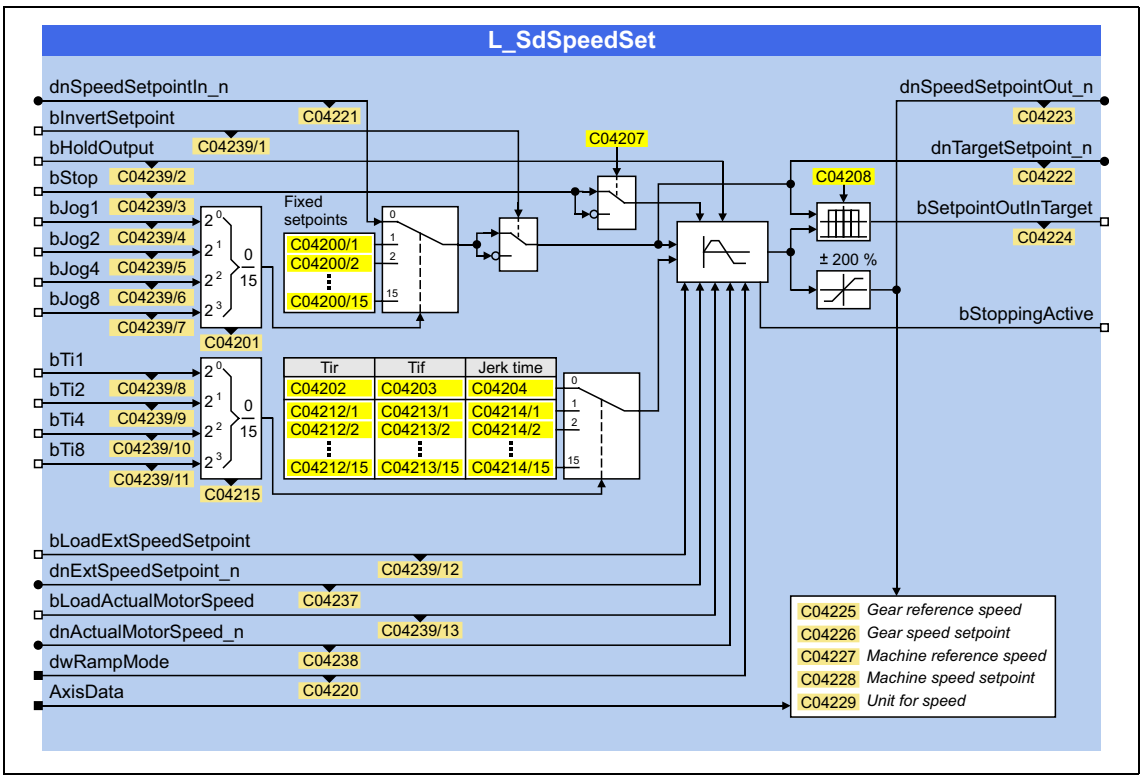
The FB causes a corresponding error message in case of an internal counter overflow due to a changed measuring system:

Error number		Error message in the logbook	Response
61603852	0x3AC000C	L_SdSpeedFilter:int.speed overflow (LS_DriveInterface)	Error
61603980	0x3AC008C	L_SdSpeedFilter:int.speed overflow (L_SdSetAxisData)	Error

5.140 L_SdSpeedSet - setpoint ramp generator

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB contains a ramp generator with comprehensive parameterisation and control options for conditioning a setpoint signal.



Inputs

Identifier/data type	Information/possible settings
dnSpeedSetpointIn_n DINT	Input setpoint in [%] • Relating to the reference speed (C00011)
bInvertSetpoint BOOL	Reversing the speed setpoint
	TRUE: Speed setpoint is inverted.
bHoldOutput BOOL	Holding (freezing) the output value
	TRUE: The current value of the ramp generator is held.
bStop BOOL	Activating "Ramp-down" to standstill • This input has priority over the input *bHoldOutput*.
	TRUE: The current value of the ramp generator is led to "0" via the Ti time set.
bJog1 ... bJog8 BOOL	Binary coded selection of the setpoint. • When all inputs are set to FALSE, the setpoint *dnSpeedSetpointIn_n* is selected, otherwise the corresponding fixed setpoint (C04200/1...15).
bTI1 ... bTI8 BOOL	Binary coded selection of the acceleration, deceleration and S-ramp time. • The times are set in the subcodes of C04212, C04213 and C04214. • The conversion is controlled via *dwRampMode* and *bStop*.

Identifier/data type	Information/possible settings	
bLoadExtSpeedSetpoint BOOL	Loading of the ramp generator with the motor-based speed setpoint *dnExtSpeedSetpoint_n*.	
	TRUE	The input signal *dnExtSpeedSetpoint_n* is loaded into the ramp generator.
dnExtSpeedSetpoint_n DINT	Motor-based speed setpoint in [%] • Is loaded into the ramp generator by setting *bLoadExtSpeedSetpoint* to TRUE.	
bLoadActualMotorSpeed BOOL	Loading of the ramp generator with the motor-based actual speed *dnActualMotorSpeed_n*.	
	TRUE	The input signal *dnActualMotorSpeed_n* is loaded into the ramp generator.
dnActualMotorSpeed_n DINT	Motor-based actual speed value in [%] • Is loaded into the ramp generator by setting *bLoadActualMotorSpeed* to TRUE.	
dwRampMode DWORD	Type of profile generation • Can also be set via C04220.	
	0	Profile generation with linear ramps. • Relating to the reference speed (C00011)
	1	Profile generation with jerk limitation (S ramps).
	10	Profile generation with linear ramps. • Relating to the reference speed (C00011) • In case of edge change at the *bStop* input with regard to the current setpoint (constant time). • The setpoint applied to the input *dnExtSpeedSetpoint_n* is adopted as starting value.
AxisData	Machine parameters • For calculation and display of reference values on the load side (output shaft speed, machine speed) the reference speed (C00011), the gearbox factor and the feed constant are required. • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**.	

Outputs

Identifier/data type	Value/meaning	
dnSpeedSetpointOut_n DINT	Output setpoint • Relating to the reference speed (C00011)	
bStoppingActive BOOL	Status signal "Ramp-down is active" • Ramp-down can be activated via the input *bStop*.	
	TRUE	The ramp generator leads the current output value to "0" (standstill).
	TRUE↘FALSE	The output value equals "0" (standstill).
dnTargetSetpoint_n DINT	Current target setpoint (at the input of the ramp generator) • Relating to the reference speed (C00011)	
bSetpointOutInTarget BOOL	Status signal "Default value reached"	
	TRUE	The output of the ramp generator has reached the default value (in consideration of the window set).

Parameter

Parameter	Possible settings			Information
C04200/1...15	-200.00	%	200.00	Fixed setpoint 1 ... 15 • Initialisation: 0.00 %
C04201	0		15	Active setpoint • Read only

Parameter	Possible settings			Information
C04202	0.000	s	1000.000	Basic acceleration time • Initialisation: 1.000 s
C04203	0.000	s	1000.000	Basic deceleration time • Initialisation: 1.000 s
C04204	0.000	s	10.000	Basic S-ramp time • Initialisation: 0.100 s
C04207				Polarity *bStop* input
	0	**HIGH active**		
	1	LOW active		
C04208	0.00	%	100.00	Tolerance setpoint reached • Initialisation: 1.00 %
C04212/1...15	0.000	s	1000.000	Acceleration time 1 ... 15 • Initialisation: 0.000 s
C04213/1...15	0.000	s	1000.000	Deceleration time 1 ... 15 • Initialisation: 0.000 s
C04214/1...15	0.000	s	1000.000	S-ramp time 1 ... 15 • Initialisation: 0.000 s
C04215	0		15	Active ramp parameters • Read only
C04220				Profile mode • Read only
	0	Profile generation with linear ramps • Relating to the reference speed (C00011)		
	1	Profile generation with jerk limitation (S ramps)		
	10	Profile generation with linear ramps • Relating to the reference speed (C00011) • In case of edge change at the *bStop* input with regard to the current setpoint (constant time). • The setpoint applied to the input *dnExtSpeedSetpoint_n* is adopted as starting value.		
C04221	-200.00	%	200.00	Setpoint at the input • Display of the *dnSpeedSetpointIn_n* input signal.
C04222	-200.00	%	200.00	Target setpoint • Display of the *dnTargetSetpoint_n* output signal
C04223	-200.00	%	200.00	Setpoint at the output • Display of the *dnSpeedSetpointOut_n*output signal.
C04224				Status "Target setpoint reached" • Display of the *bSetpointOutInTarget* output signal.
	0	Target setpoint not yet reached		
	1	Target setpoint reached		
C04225	0.000	rpm	50000.000	Load reference speed • Read only
C04226	0.000	rpm	50000.000	Setpoint speed - load • Read only
C04227	0.0000	Unit/t	214000.0000	Reference speed • Read only

Parameter	Possible settings			Information
C04228	-214000.0000	Unit/t	214000.0000	Setpoint speed • Read only
C04229	String from *AxisData*			Speed unit • Read only
C04237	-200.00	%	200.00	External setpoint • Display of the *dnExtSpeedSetpoint_n* input signal.
C04238	-200.00	%	200.00	Current motor speed • Display of the *dnActualMotorSpeed_n* input signal.
C04239/1				Reverse setpoint • Display of the *bInvertSetpoint* input signal.
	0	Setpoint not inverted		
	1	Setpoint inverted		
C04239/2				Hold setpoint output • Display of the *bHoldOutput* input signal.
	0	Setpoint output enabled		
	1	Hold setpoint output		
C04239/3				Stop • Display of the *bStop* input signal.
	0	Stop not active		
	1	Stop requested		
C04239/4...7				Jog input 1 ... 8 • Display of the *bJog1* ... *bJog8* input signals.
	0	Jog input not active		
	1	Jog input active		
C04239/8...11				Ti input 1 ... 8 • Display of the *bTi1* ... *bTi8* input signals.
	0	Ti input not active		
	1	Ti input active		
C04239/12				Load external setpoint • Display of the *bLoadExtSpeedSetpoint* input signal.
	0	Loading not active		
	1	Load external setpoint		
C04239/13				Load motor speed • Display of the *bLoadActualMotorSpeed* input signal.
	0	Loading not active		
	1	Load motor speed		

5.140.1 Setpoint path

- The signals in the setpoint path are limited to ±200.00 % of the motor Motor reference speed (C00011).
- The signal at *dnSpeedSetpointIn_n* is first led via the function "Selection of fixed setpoints".
- A selected fixed setpoint switches the input *dnSpeedSetpointIn_n* inactive and the subsequent signal conditioning uses the selected fixed setpoint.
- By setting the input *bInvertSetpoint* to TRUE the input signal of the ramp generator can be inverted.

5.140.2 Fixed setpoints

In addition to the direct setpoint selection via the input *dnSpeedSetpointIn_n* fixed setpoints can be preset in C04200/1...15.

- These fixed setpoints can be called binary coded any time via the selection inputs *bJog1* ... *bJog8*, so 15 options are available.

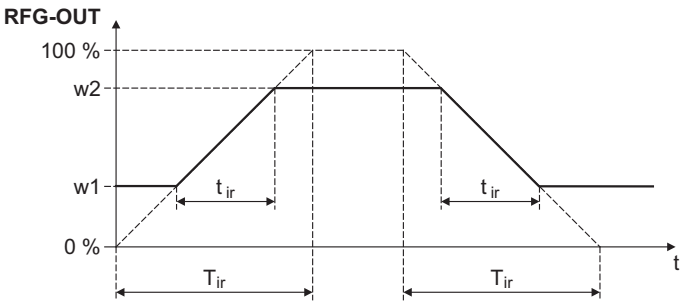
Selected inputs				Used Setpoint
bJog8	bJog4	bJog2	bJog1	
FALSE	FALSE	FALSE	FALSE	*dnSpeedSetpointIn_n*
FALSE	FALSE	FALSE	TRUE	C04200/1
FALSE	FALSE	TRUE	FALSE	C04200/2
FALSE	FALSE	TRUE	TRUE	C04200/3
FALSE	TRUE	FALSE	FALSE	C04200/4
FALSE	TRUE	FALSE	TRUE	C04200/5
FALSE	TRUE	TRUE	FALSE	C04200/6
FALSE	TRUE	TRUE	TRUE	C04200/7
TRUE	FALSE	FALSE	FALSE	C04200/8
TRUE	FALSE	FALSE	TRUE	C04200/9
TRUE	FALSE	TRUE	FALSE	C04200/10
TRUE	FALSE	TRUE	TRUE	C04200/11
TRUE	TRUE	FALSE	FALSE	C04200/12
TRUE	TRUE	FALSE	TRUE	C04200/13
TRUE	TRUE	TRUE	FALSE	C04200/14
TRUE	TRUE	TRUE	TRUE	C04200/15

- The number of selected inputs to be assigned depends on the number of required fixed setpoints.

Number of the fixed setpoints required	Number of the selection inputs to be assigned (bJog1 ... bJog8)
1	at least 1
1 ... 3	at least 2
4 ... 7	at least 3
8 ... 15	4

5.140.3 Ramp generator

Afterwards, the setpoint is led via a ramp generator with S-ramp characteristic. The ramp generator transfers setpoint step-changes at the input to a ramp.



w1, w2 = change of the setpoint as a function of t_{ir} or t_{if}
RFG-OUT = output of the ramp generator

[5-110] Acceleration and deceleration times

- t_{ir} and t_{if} are the desired times for changing between w1 and w2.
- S-ramps are possible if an S-ramp time is selected.
- The t_{ir}/t_{if} values are converted into the required Ti times according to the following formula:

$T_{ir} = t_{ir} \cdot \frac{100\,\%}{w2 - w1}$	$T_{if} = t_{if} \cdot \frac{100\,\%}{w2 - w1}$

5.140.4 Setting and selection of the Ti times

16 different Tir, Tif and S-ramp times can be selected via parameters for the ramp generator.

- The selection is carried out binary coded via the selection inputs *bTI1* ... *bTI8*:

Selected inputs				Selection		
bTI8	bTI4	bTI2	bTI1	Tir	Tif	S-ramp time
FALSE	FALSE	FALSE	FALSE	C04202	C04203	C04204
FALSE	FALSE	FALSE	TRUE	C04212/1	C04213/1	C04214/1
FALSE	FALSE	TRUE	FALSE	C04212/2	C04213/2	C04214/2
FALSE	FALSE	TRUE	TRUE	C04212/3	C04213/3	C04214/3
FALSE	TRUE	FALSE	FALSE	C04212/4	C04213/4	C04214/4
FALSE	TRUE	FALSE	TRUE	C04212/5	C04213/5	C04214/5
FALSE	TRUE	TRUE	FALSE	C04212/6	C04213/6	C04214/6
FALSE	TRUE	TRUE	TRUE	C04212/7	C04213/7	C04214/7
TRUE	FALSE	FALSE	FALSE	C04212/8	C04213/8	C04214/8
TRUE	FALSE	FALSE	TRUE	C04212/9	C04213/9	C04214/9
TRUE	FALSE	TRUE	FALSE	C04212/10	C04213/10	C04214/10
TRUE	FALSE	TRUE	TRUE	C04212/11	C04213/11	C04214/11
TRUE	TRUE	FALSE	FALSE	C04212/12	C04213/12	C04214/12
TRUE	TRUE	FALSE	TRUE	C04212/13	C04213/13	C04214/13
TRUE	TRUE	TRUE	FALSE	C04212/14	C04213/14	C04214/14
TRUE	TRUE	TRUE	TRUE	C04212/15	C04213/15	C04214/15

5.140.5 Display of the machine-based variables

For a better assessment of the created setpoints/parameterised fixed values (% value) the machine-based reference values are calculated and displayed:

Parameter	Information
C04225	Load reference speed
C04226	Setpoint speed - load
C04227	Machine reference speed
C04228	Machine setpoint speed

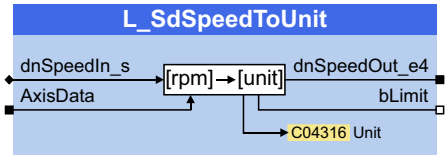
The current setpoint is also calculated and displayed correspondingly.

Parameter	Information
C04221	Setpoint at the input
C04222	Target setpoint
C04223	Setpoint at the output

5.141 L_SdSpeedToUnit - speed conversion

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts a speed value defined in the internal unit [rpm] into a speed value in the real unit of the machine.



Inputs

Identifier/data type	Information/possible settings
dnSpeedIn_s DINT	Speed given in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnSpeedOut_e4 DINT	Speed in [unit/s] • Output in "e4" format (fixed point with four decimal positions)	
bLimit BOOL	Status "Output signal is limited"	
	TRUE	The output signal is limited to the value range that can be displayed.

Parameter

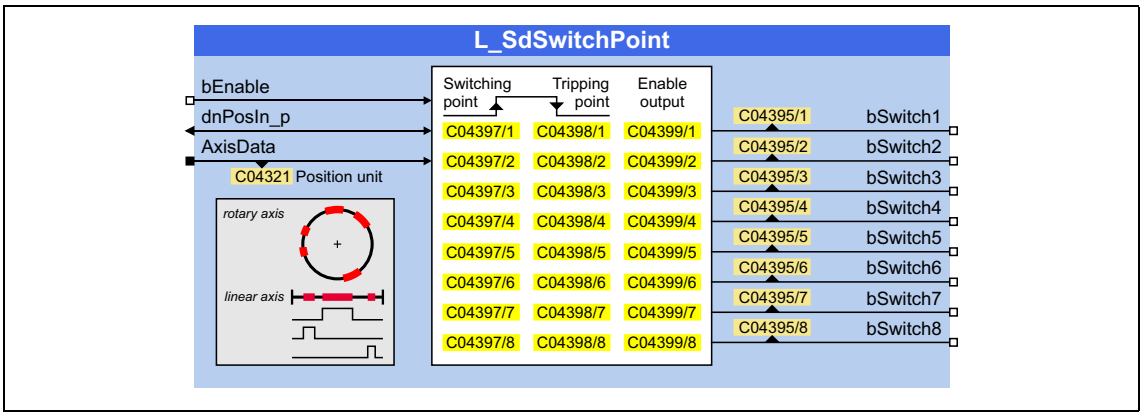
Parameter	Possible settings	Information
C04316	String from *AxisData*	Speed unit • Read only

5.142 L_SdSwitchPoint - position switch points

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to implement up to 8 simple position switch points (cams). When the current drive position is inside a range defined via parameters, the output assigned to this "switching area" is set to TRUE.

- The output setting does not depend on whether the switching range is approached by the positive or negative side.
- In the modulo measuring system (rotary table) the switching ranges can also be defined beyond the 0° position (e.g. switching on at 350° and off at 10° in CW rotation).



Inputs

Identifier/data type	Information/possible settings
bEnable BOOL	Enable FB TRUE: Position switch points are activated.
dnPosIn_p DINT	Input for accepting the current position • Connect this input with the position signal to be accepted. Depending on the application, this can be a setpoint or an actual position.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
bSwitch1 BOOL	Position switch point 1	
	TRUE	The current position (*dnPosIn_p*) is inside the switching range defined via C04397/1 and C04398/1.
bSwitch2 BOOL	Position switch point 2	
	TRUE	The current position (*dnPosIn_p*) is inside the switching range defined via C04397/2 and C04398/2.
bSwitch3 BOOL	Position switch point 3	
	TRUE	The current position (*dnPosIn_p*) is inside the switching range defined via C04397/3 and C04398/3.
bSwitch4 BOOL	Position switch point 4	
	TRUE	The current position (*dnPosIn_p*) is inside the switching range defined via C04397/4 and C04398/4.
bSwitch5 BOOL	Position switch point 5	
	TRUE	The current position (*dnPosIn_p*) is inside the switching range defined via C04397/5 and C04398/5.
bSwitch6 BOOL	Position switch point 6	
	TRUE	The current position (*dnPosIn_p*) is inside the switching range defined via C04397/6 and C04398/6.
bSwitch7 BOOL	Position switch point 7	
	TRUE	The current position (*dnPosIn_p*) is inside the switching range defined via C04397/7 and C04398/7.
bSwitch8 BOOL	Position switch point 8	
	TRUE	The current position (*dnPosIn_p*) is inside the switching range defined via C04397/8 and C04398/8.

Parameter

Parameter	Possible settings			Information
C04395/1...8				Status of switching range 1 ... 8 • Read only
	0	Switch off		
	1	Switch on		
C04396 As of library V02.02.xx.xx	String of digits			Position unit • Read only
C04397/1...8	-214748.3647	Unit	214748.3647	Switch-on position - switching range 1 ... 8
C04398/1...8	-214748.3647	Unit	214748.3647	Switch-off position - switching range 1 ... 8
C04399/1...8				Enable - switching range 1 ... 8
	0	Switching range not enabled		
	1	Switching range enabled		

5.142.1 Monitoring for counter overflow

This function extension will be available from library V2.02!

If the machine parameters transferred to the FB via the *AxisData* input are changed afterwards (e.g. the gearbox factors), this may cause a counter overflow when the position values defined via C04397/1...8 and C04398/1...8 converted from the real unit to the internal unit.

The FB causes a corresponding error message in case of an internal counter overflow due to a changed measuring system:

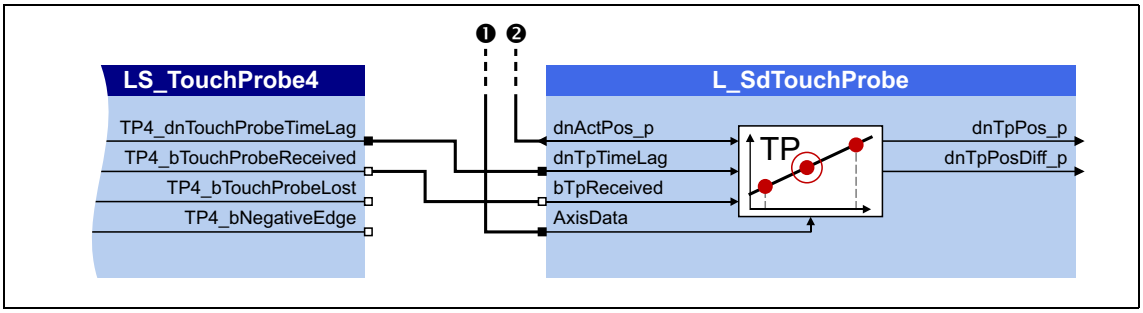
Error number		Error message in the logbook	Response
61603846	0x3AC0006	L_SdSwitchPoint:int.pos. overflow (LS_DriveInterface)	Error
61603974	0x3AC0086	L_SdSwitchPoint:int.pos. overflow (L_SdSetAxisData)	Error

5.143 L_SdTouchProbe - touch probe evaluation

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB takes over the interpolation of the input signal based on the time stamp handed over by a touch probe system block and outputs the interpolated value and the difference compared with the last input signal.

- The FB considers a modulo measuring system and the cycle of the selected machine parameters.



Inputs

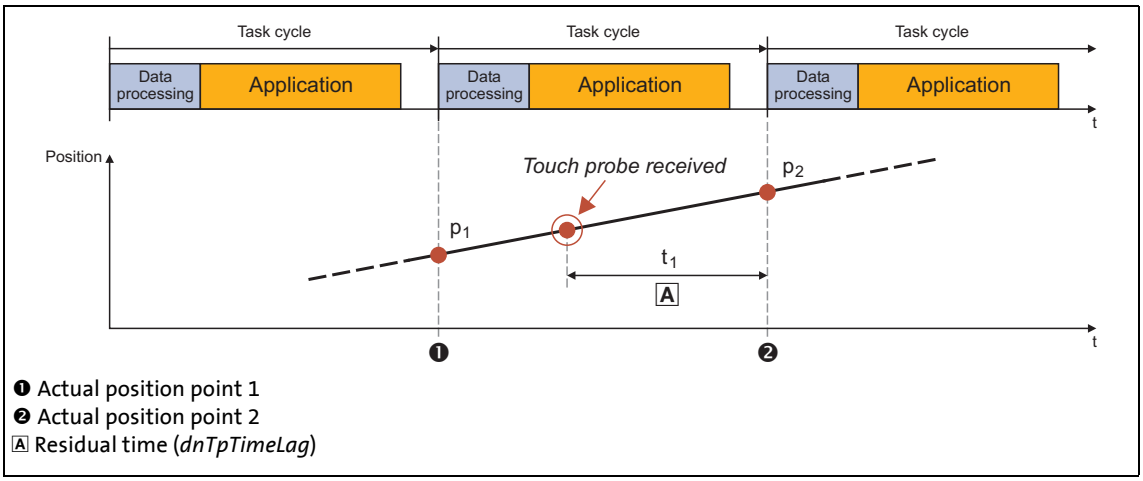
Identifier/data type	Information/possible settings
dnActPos_p DINT	Input for accepting the current position in [inc] • Connect this input with the signal to be interpolated, e.g. the master value of a line drive.
dnTpTimeLag DINT	Input for accepting the time stamp • Connect this input with the output *dnTouchProbeTimeLag* of the corresponding touch probe system block.
bTpReceived BOOL	Input for taking over the status "Touch probe detected" • Connect this input with the output *dnTouchProbeReceived* of the corresponding touch probe system block.
	FALSE↗TRUE | Touch probe detected.
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning
dnTpPos_p DINT	Interpolated TP position in [inc]
dnTpPosDiff_p DINT	Position difference in [increments] • Position difference between the last time the TP position was latched and the starting time of the task.

5.143.1 Actual value interpolation (principle)

If a touch probe is detected, the (residual) time to the following task cycle is determined and a time stamp is created from it. Based on this time stamp the FB **L_SdTouchProbe** can execute a linear interpolation between both actual position points. The result is the exact actual position at the time of the physical touch probe event.

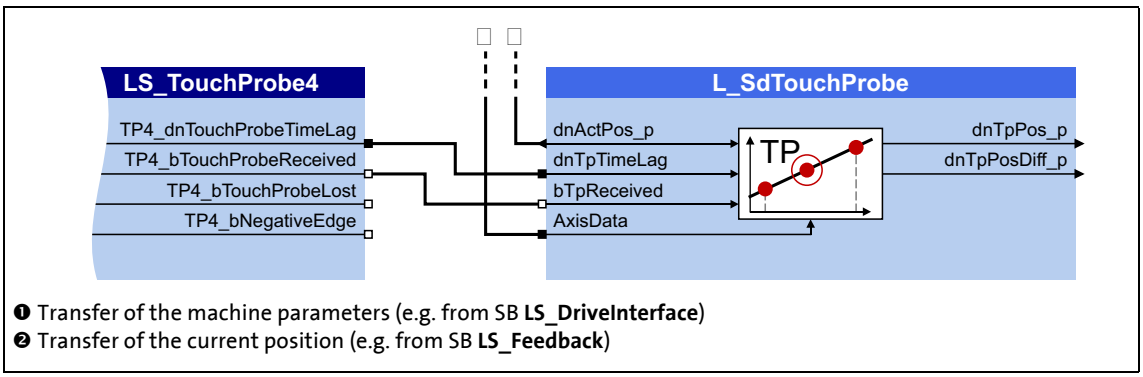


❶ Actual position point 1
❷ Actual position point 2
A Residual time (*dnTpTimeLag*)

[5-111] Actual value determination through linear interpolation (principle)

5.143.2 Example

Connections required for a touch probe detection via the digital input DI4:



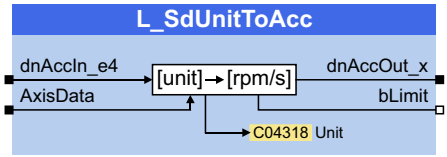
❶ Transfer of the machine parameters (e.g. from SB **LS_DriveInterface**)
❷ Transfer of the current position (e.g. from SB **LS_Feedback**)

[5-112] Transfer of the FB L_SdTouchProbe

5.144 L_SdUnitToAcc - acceleration conversion

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts an acceleration value defined in the real unit of the machine into an internal acceleration value.



Inputs

Identifier/data type	Information/possible settings
dnAccIn_e4 DINT	Acceleration in [unit/s^2] • Selection in "e4" format (fixed point with four decimal positions)
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnAccOut_x DINT	Acceleration as speed variation/time in [rpm/s] • 15000000 rpm/s ≡ 2^{22} ≡ 4194304	
bLimit BOOL	Status "Output signal is limited"	
	TRUE	The output signal is limited to the value range that can be displayed.

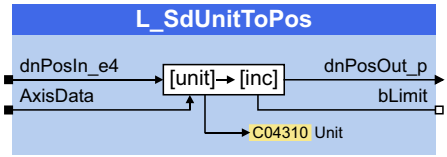
Parameter

Parameter	Possible settings	Information
C04318	String from *AxisData*	Acceleration unit • Read only

5.145 L_SdUnitToPos - position conversion

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts a position defined in the real unit of the machine into an internal position.



Inputs

Identifier/data type	Information/possible settings
dnPosIn_e4 DINT	Position in [unit] • Selection in "e4" format (fixed point with four decimal positions)
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnPosOut_p DINT	Position in [inc]	
bLimit BOOL	Status "Output signal is limited"	
	TRUE	The output signal is limited to the value range that can be displayed.

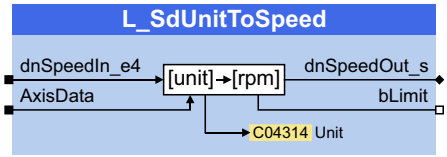
Parameter

Parameter	Possible settings	Information
C04310	String from *AxisData*	Position unit • Read only

5.146 L_SdUnitToSpeed - speed conversion

Function library:	LenzeServoDrive
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB uses the transmitted machine parameters and converts a speed defined in the real unit of the machine into an internal speed.



Inputs

Identifier/data type	Information/possible settings
dnSpeedIn_e4 DINT	Speed in [unit/s] • Selection in "e4" format (fixed point with four decimal positions)
AxisData	Machine parameters • For accepting the machine parameters of the drive/motor, connect this input with the output *DI_AxisData* of the SB **LS_DriveInterface**. • The machine parameters of a master drive can be displayed with the FB L_SdSetAxisData. In this case, the FB output *AxisData* must be connected to this input. (📖 510)

Outputs

Identifier/data type	Value/meaning	
dnSpeedOut_s DINT	Speed given in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864	
bLimit BOOL	Status "Output signal is limited"	
	TRUE	The output signal is limited to the value range that can be displayed.

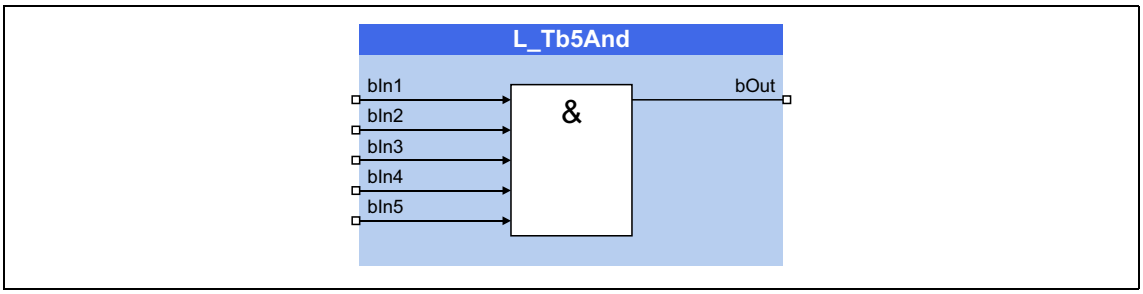
Parameter

Parameter	Possible settings	Information
C04314	String from *AxisData*	Speed unit • Read only

5.147 L_Tb5And - AND with 5 inputs

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a logic AND operation of binary signals.



Inputs

Identifier/data type	Information/possible settings
bIn1 BOOL	Input 1
bIn2 BOOL	Input 2
bIn3 BOOL	Input 3
bIn4 BOOL	Input 4
bIn5 BOOL	Input 5

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Result of the AND operation

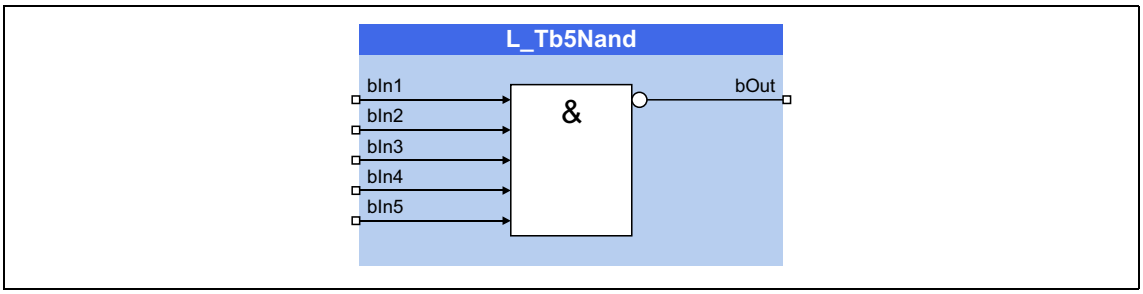
Tip!

If you want to combine less than five signals, set the unused inputs to TRUE or assign several inputs to the same signal.

5.148 L_Tb5Nand - NAND with 5 inputs

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a logic NAND operation of binary signals.



Inputs

Identifier/data type	Information/possible settings
bIn1 BOOL	Input 1
bIn2 BOOL	Input 2
bIn3 BOOL	Input 3
bIn4 BOOL	Input 4
bIn5 BOOL	Input 5

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Result of the NAND operation

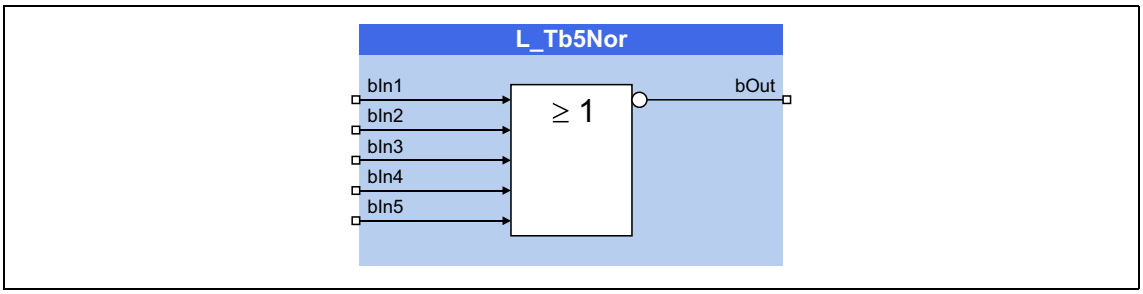
Tip!

If you want to combine less than five signals, set the unused inputs to TRUE or assign several inputs to the same signal.

5.149 L_Tb5Nor - NOR with 5 inputs

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a logic NOR operation of binary signals.



Inputs

Identifier/data type	Information/possible settings
bIn1 BOOL	Input 1
bIn2 BOOL	Input 2
bIn3 BOOL	Input 3
bIn4 BOOL	Input 4
bIn5 BOOL	Input 5

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Result of the NOR operation

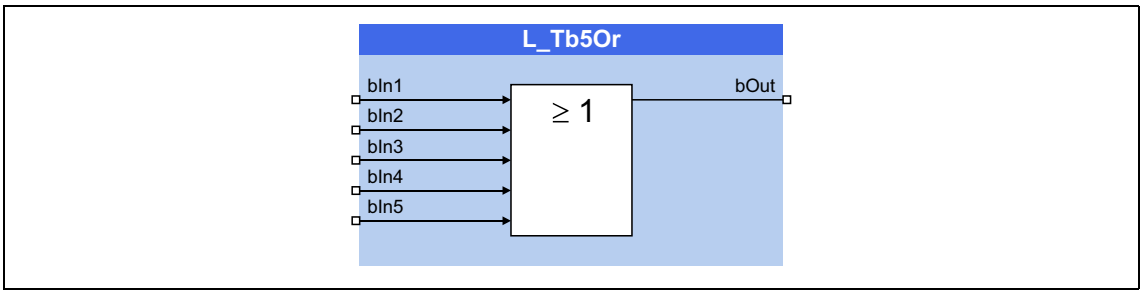
Tip!

If you want to combine less than five signals, set the unused inputs to FALSE.

5.150 L_Tb5Or - OR with 5 inputs

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a logic OR operation of binary signals.



Inputs

Identifier/data type	Information/possible settings
bIn1 BOOL	Input 1
bIn2 BOOL	Input 2
bIn3 BOOL	Input 3
bIn4 BOOL	Input 4
bIn5 BOOL	Input 5

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Result of the OR operation

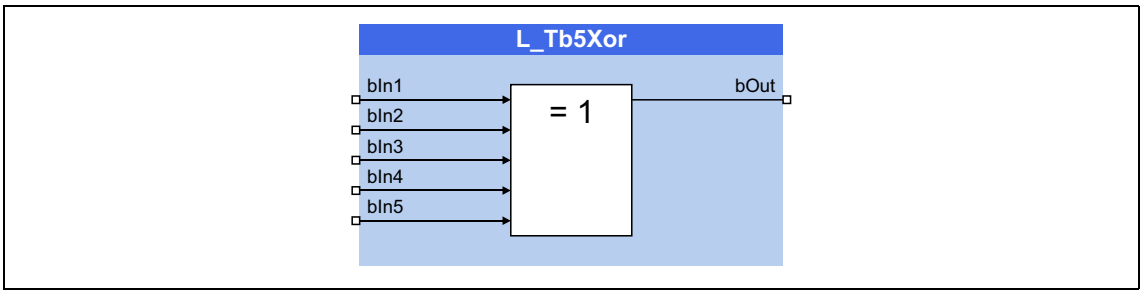
Tip!

If you want to combine less than five signals, set the unused inputs to FALSE.

5.151 L_Tb5Xor - XOR with 5 inputs

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a logic EXCLUSIVE OR operation of binary signals.



Inputs

Identifier/data type	Information/possible settings
bIn1 BOOL	Input 1
bIn2 BOOL	Input 2
bIn3 BOOL	Input 3
bIn4 BOOL	Input 4
bIn5 BOOL	Input 5

Outputs

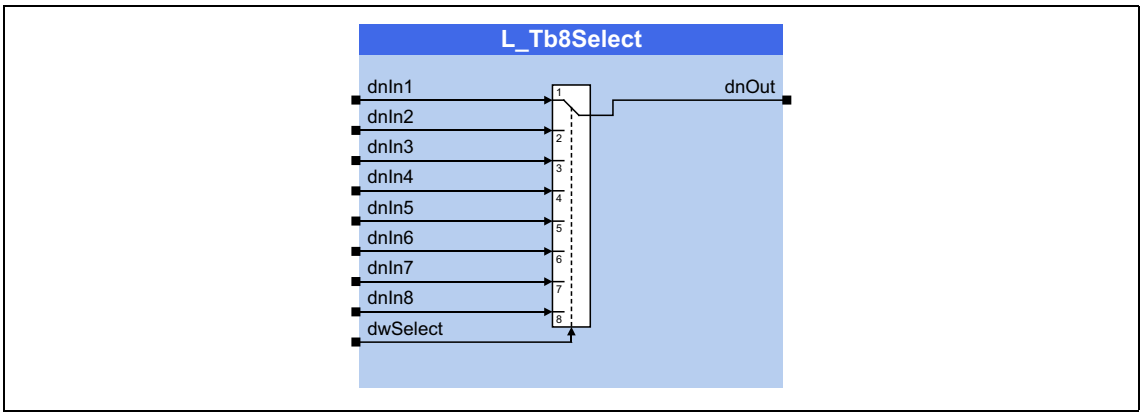
Identifier/data type	Value/meaning
bOut BOOL	Result of the exclusive OR operation

5.152 L_Tb8Select - 1-out-of-8 selector (for data type "DINT")

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB connects one of eight input signals of "DINT" type through to the output.

- The input is selected via the input *dwSelect*.



Inputs

Identifier/data type	Information/possible settings	
dnIn1 ... dnIn8 DINT	Input signal 1 ... 8	
dwSelect DWORD	Selection of the input signal for the output to *dnOut*	
	0	0
	1	*dnIn1*
	2	*dnIn2*
	3	*dnIn3*
	4	*dnIn4*
	5	*dnIn5*
	6	*dnIn6*
	7	*dnIn7*
	8	*dnIn8*
	> 8	0

Outputs

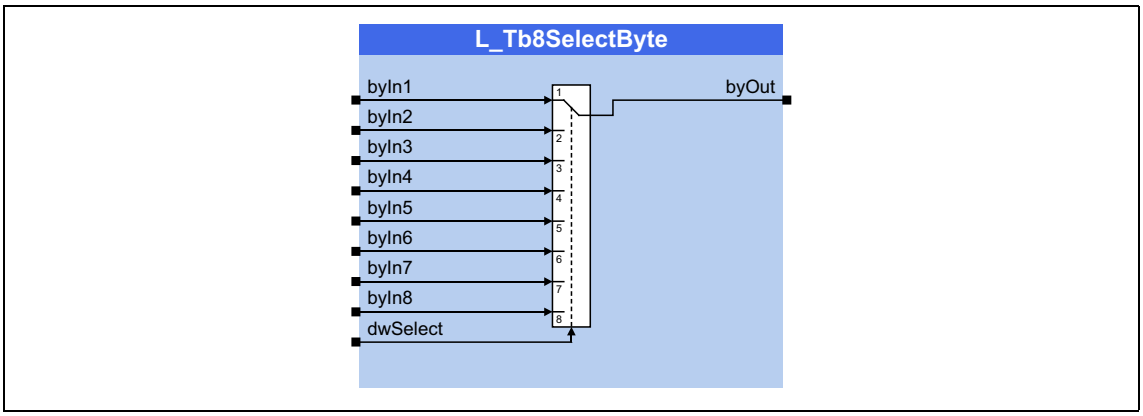
Identifier/data type	Value/meaning
dnOut DINT	Output signal

5.153 L_Tb8SelectByte - 1-out-of-8 selector (for data type "BYTE")

Function library:	LenzeToolbox	**FB is available as of library V02.05.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB connects one of eight input signals of "BYTE" type through to the output.

- The input is selected via the input *dwSelect*.



Inputs

Identifier/data type	Information/possible settings	
byIn1 ... byIn8 BYTE	Input signal 1 ... 8	
dwSelect DWORD	Selection of the input signal for the output to *byOut*	
	0	0
	1	*byIn1*
	2	*byIn2*
	3	*byIn3*
	4	*byIn4*
	5	*byIn5*
	6	*byIn6*
	7	*byIn7*
	8	*byIn8*
	> 8	0

Outputs

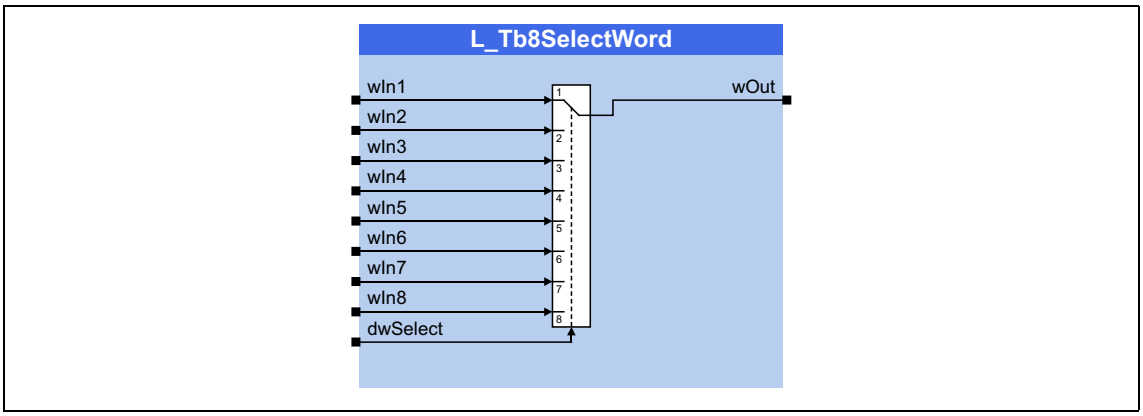
Identifier/data type	Value/meaning
byOut BYTE	Output signal

5.154 L_Tb8SelectWord - 1-out-of-8 selector (for data type "WORD")

Function library:	LenzeToolbox	**FB is available as of library V02.05.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB connects one of eight input signals of "WORD" type through to the output.

- The input is selected via the input *dwSelect*.



Inputs

Identifier/data type	Information/possible settings	
wIn1 ... wIn8 WORD	Input signal 1 ... 8	
dwSelect DWORD	Selection of the input signal for the output to *wOut*	
	0	0
	1	*wIn1*
	2	*wIn2*
	3	*wIn3*
	4	*wIn4*
	5	*wIn5*
	6	*wIn6*
	7	*wIn7*
	8	*wIn8*
	> 8	0

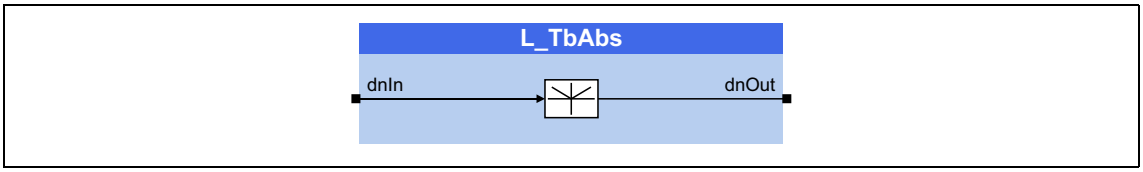
Outputs

Identifier/data type	Value/meaning
wOut WORD	Output signal

5.155 L_TbAbs - absolute value

Function library:	LenzeToolbox	
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This function block converts a bipolar signal into a unipolar signal, i.e. the absolute value of the input signal is generated.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal

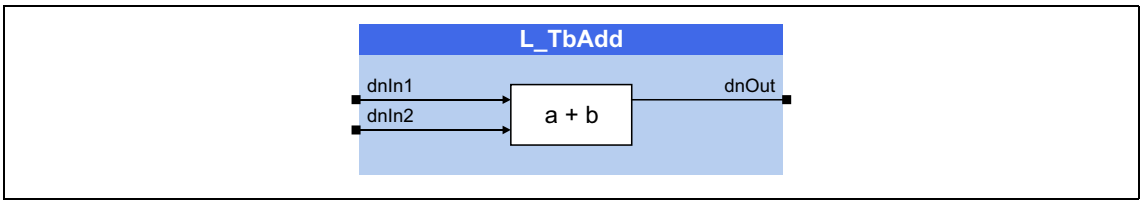
Outputs

Identifier/data type	Value/meaning
dnOut DINT	Output signal

5.156 L_TbAdd - addition

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out an addition without limitation.



Inputs

Identifier/data type	Information/possible settings
dnIn1 DINT	1st summand
dnIn2 DINT	2nd summand

Outputs

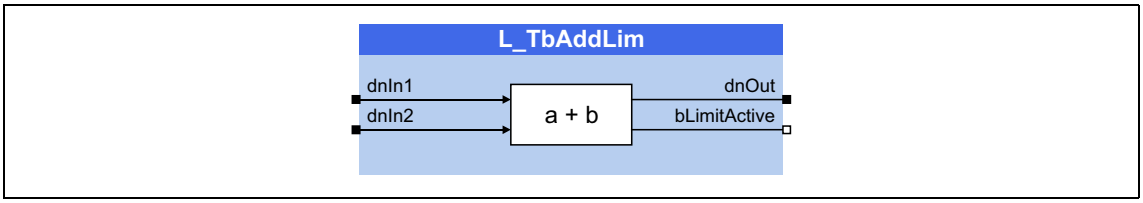
Identifier/data type	Value/meaning
dnOut DINT	Cumulative value (result of the addition) • No internal limitation, thus an overflow is possible: • If the result > 2^{31}-1: *dnOut* = *dnIn1* + *dnIn2* - 2^{32} • If the result < 2^{31}: *dnOut* = *dnIn1* + *dnIn2* + 2^{32}

5.157 L_TbAddLim - addition with limitation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out an addition with limitation.

- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnIn1 DINT	1st summand
dnIn2 DINT	2nd summand

Outputs

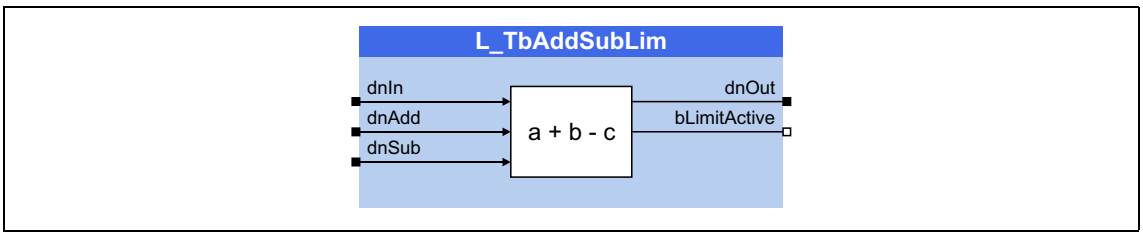
Identifier/data type	Value/meaning	
dnOut DINT	Cumulative value (result of the addition) • Internally limited to $\pm 2^{31}$-1	
bLimitActive BOOL	Status signal "Limitation active"	
	TRUE	The output signal is limited.

5.158 L_TbAddSubLim - addition and subtraction with limitation

Function library:	LenzeToolbox	**FB is available as of library V02.04.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out an addition with subsequent subtraction and limitation.

- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input value
dnAdd DINT	Value that is added to the input value.
dnSub DINT	Value that is subtracted from the input value.

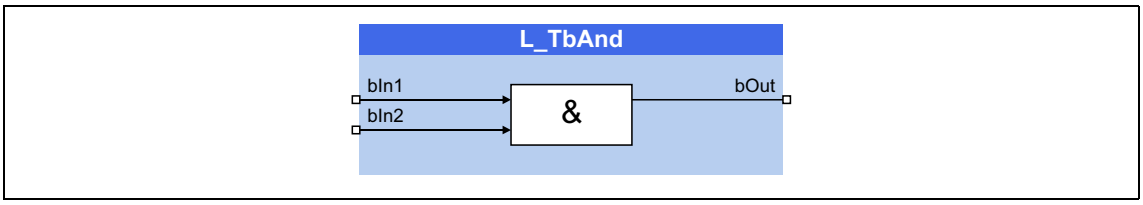
Outputs

Identifier/data type	Value/meaning	
dnOut DINT	Result of the calculation • Internally limited to $\pm 2^{31}$-1	
bLimitActive BOOL	Status signal "Limitation active"	
	TRUE	The output signal is limited.

5.159 L_TbAnd - AND with 2 inputs

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a logic AND operation of binary signals.



Inputs

Identifier/data type	Information/possible settings
bIn1 BOOL	Input 1
bIn2 BOOL	Input 2

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Result of the AND operation

Function

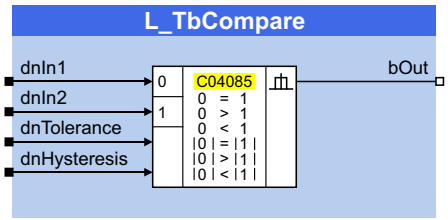
bIn2	bIn1	bOut
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

5.160 L_TbCompare - comparison

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This function block compares two signals of the "DINT" data type and can be used e.g. for implementing a trigger.

- The comparison function is selected via C04085.
- In contrast to the FB L_TbCompare_n, the window and hysteresis of this FB can be adjusted via inputs instead of parameters.



Inputs

Identifier/data type	Information/possible settings
dnIn1 DINT	Input signal 1
dnIn2 DINT	Input signal 2
dnTolerance DINT	Window for comparison operation • Value is internally limited to ≥0.
dnHysteresis DINT	Hysteresis for comparison operation • Value is internally limited to ≥0.

Outputs

Identifier/data type	Value/meaning	
bOut BOOL	Status signal "Comparison statement is true"	
	TRUE	The statement of the selected comparison mode is true.

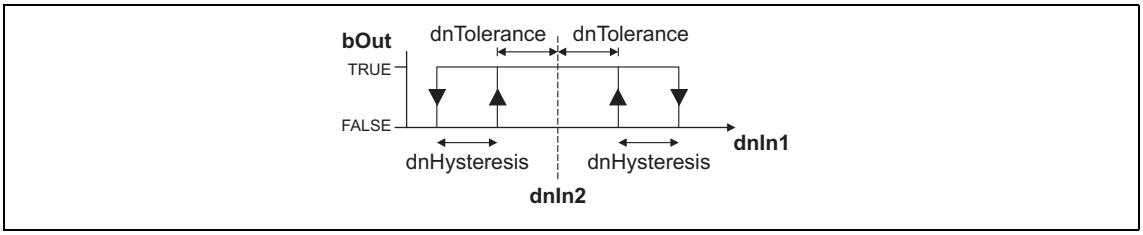
Parameter

Parameter	Possible settings		Information
C04085			Comparison operation
	0	**dnIn1 = dnIn2**	Lenze setting
	1	dnIn1 > dnIn2	
	2	dnIn1 < dnIn2	
	3	|dnIn1| = |dnIn2|	
	4	|dnIn1| > |dnIn2|	
	5	|dnIn1| < |dnIn2|	

5.160.1 Function 1: dnIn1 = dnIn2

Selection: C04085 = "0"

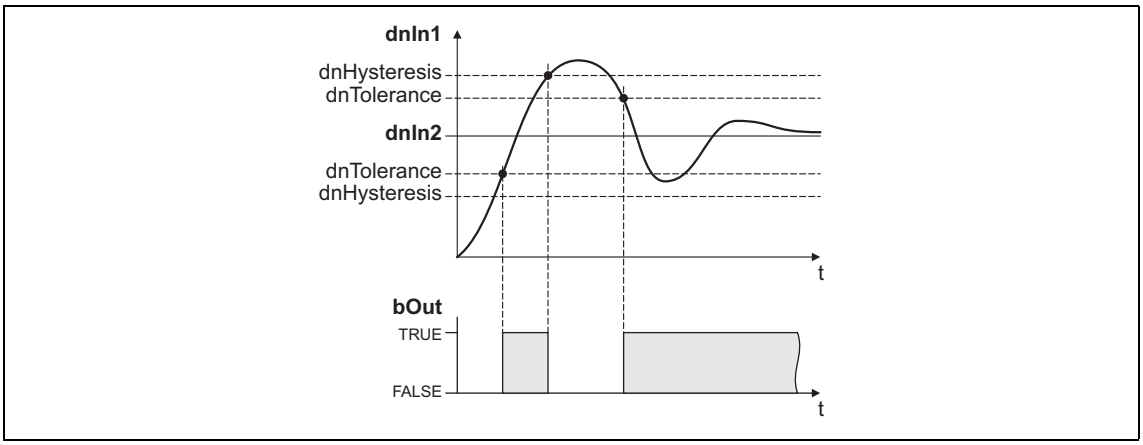
- This function compares two signals with regard to equality. They can, for instance, provide the comparison "actual speed equals setpoint speed " ($n_{act} = n_{set}$).
- Use the input *dnTolerance* to select the window within which the equality is to apply:
- Use the input *dnHysteresis* to select a hysteresis if the input signals are not stable and the output oscillates.



[5-113] Function 1: Switching performance

Tip!

With this function, the FB must be used in a fast task to achieve optimum sampling of the signals.

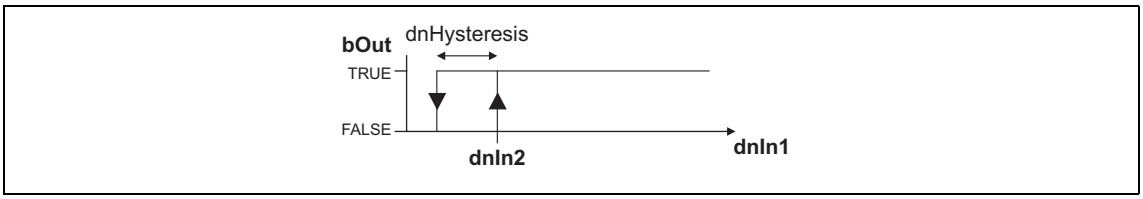


[5-114] Function 1: Example

5.160.2 Function 2: dnIn1 > dnIn2

Selection: C04085 = "1"

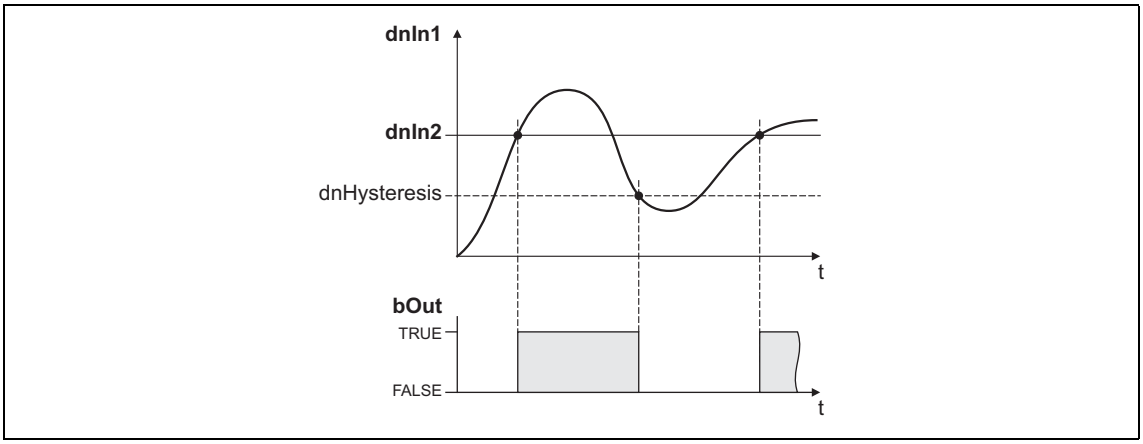
- This function serves, for instance, to implement the comparison "actual speed is higher than a limit value" ($n_{act} > n_x$) for one direction of rotation.



[5-115] Function 2: Switching performance

Functional sequence

1. If the value at d*nIn1* exceeds the value at *dnIn2*, *bOut* changes from FALSE to TRUE.
2. Only if the signal at *dnIn1* falls below the value at *dnIn2 - dnHysteresis* again, *bOut* changes from TRUE to FALSE.

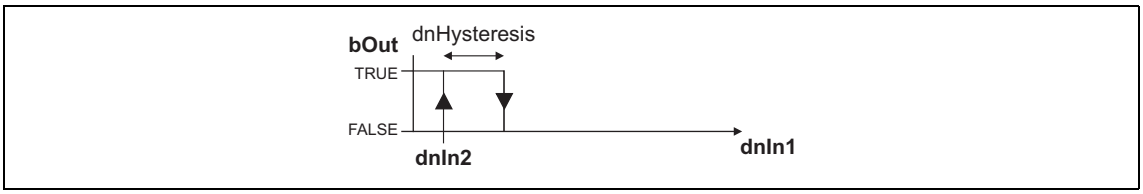


[5-116] Function 2: Example

5.160.3 Function 3: dnIn1 < dnIn2

Selection: C04085 = "2"

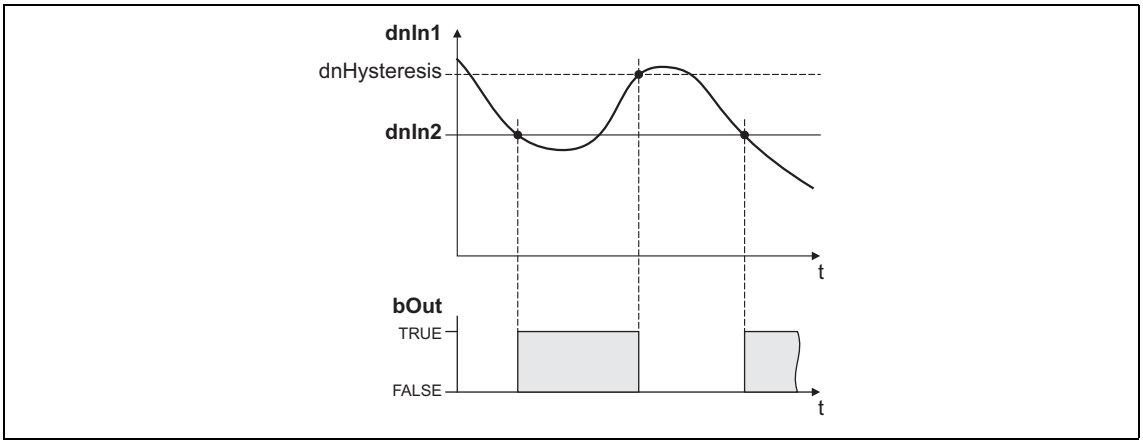
- This function serves, for instance, to implement the comparison "actual speed is lower than a limit value" (n_{act} n_x) for one direction of rotation.



[5-117] Function 3: Switching performance

Functional sequence

1. If the value at *dnIn1* falls below the value at *dnIn2*, *bOut_b* changes from FALSE to TRUE.
2. Only if the signal at *dnIn1* exceeds the value at *dnIn2 - dnHysteresis* again, *bOut_b* changes from TRUE to FALSE.



[5-118] Function 3: Example

5.160.4 Function 4: |dnIn1| = |dnIn2|

Selection: C04085 = "3"

- This function serves, for instance, to implement the comparison "$n_{act} = 0$".
- This function behaves like function 1. However, the amount is generated by the input signals before signal processing (without sign).
  ▸ Function 1: dnIn1 = dnIn2 (📖 554)

5.160.5 Function 5: |dnIn1| > |dnIn2|

Selection: C04085 = "4"

- This function serves, for instance, to implement the comparison "$|n_{act}| > |n_x|$" independent of the direction of rotation.
- This function behaves like function 2. However, the amount is generated by the input signals before signal processing (without sign).
  ▸ Function 2: dnIn1 > dnIn2 (📖 555)

5.160.6 Function 6: |dnIn1| < |dnIn2|

Selection: C04085 = "5"

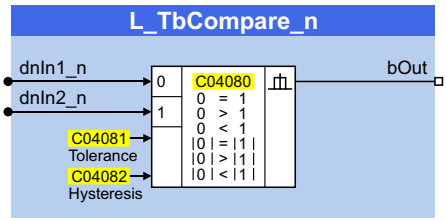
- This function serves to implement the comparison "$|n_{act}| < |n_x|$" independent of the direction of rotation.
- This function behaves like function 3. However, the amount is generated by the input signals before signal processing (without sign).
  ▸ Function 3: dnIn1 < dnIn2 (📖 556)

5.161 L_TbCompare_n - scaled comparison

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This function block compares two scaled signals and can be used e.g. for implementing a trigger.

- The comparison function is selected via C04080.
- In contrast to the FB L_TbCompare, the window and hysteresis of this FB can be adjusted via parameters instead of inputs.



Inputs

Identifier/data type	Information/possible settings
dnIn1_n DINT	Input signal 1 • Scaling: 100 % ≡ 2^{30} ≡ 1073741824
dnIn2_n DINT	Input signal 2 • Scaling: 100 % ≡ 2^{30} ≡ 1073741824

Outputs

Identifier/data type	Value/meaning	
bOut BOOL	Status signal "Comparison statement is true"	
	TRUE	The statement of the selected comparison mode is true.

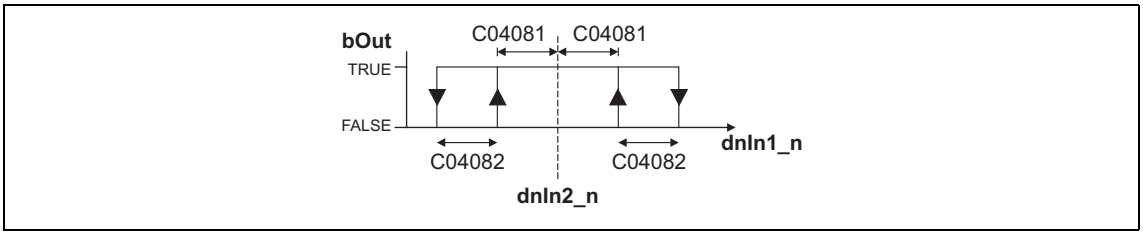
Parameter

Parameter	Possible settings			Information
C04080				Comparison operation
	0	**dnIn1_n = dnIn2_n**		Lenze setting
	1	dnIn1_n > dnIn2_n		
	2	dnIn1_n < dnIn2_n		
	3	|dnIn1_n| = |dnIn2_n|		
	4	|dnIn1_n| > |dnIn2_n|		
	5	|dnIn1_n| < |dnIn2_n|		
C04081	0.00	%	100.00	Tolerance • Initialisation: 1.00 %
C04082	0.00	%	100.00	Hysteresis • Initialisation: 1.00 %

5.161.1 Function 1: dnIn1_n = dnIn2_n

Selection: C04080 = "0"

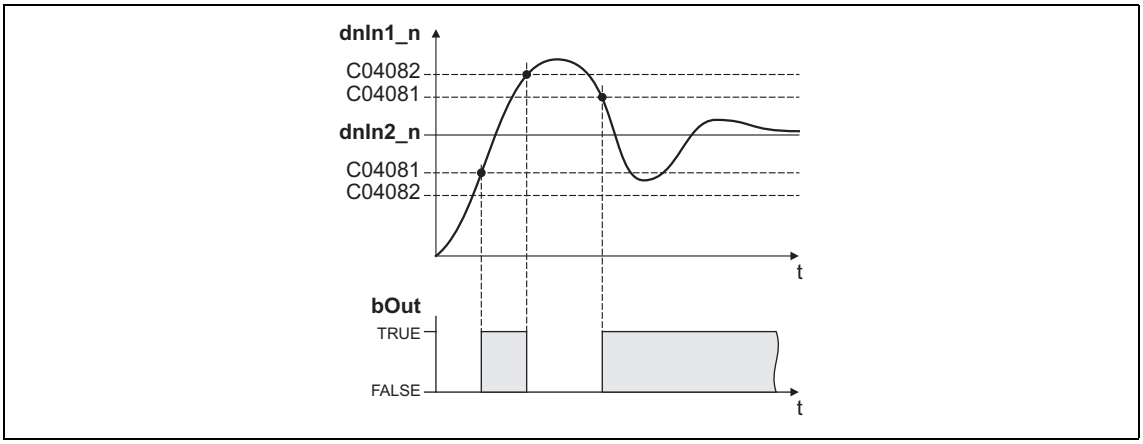
- This function compares two signals with regard to equality. They can, for instance, provide the comparison "actual speed equals setpoint speed " ($n_{act} = n_{set}$).
- Use C04081 to select the window within which the equality is to apply.
- Use C04082 to select a hysteresis if the input signals are not stable and the output oscillates.



[5-119] Function 1: Switching performance

Tip!

With this function, the FB must be used in a fast task to achieve optimum sampling of the signals.

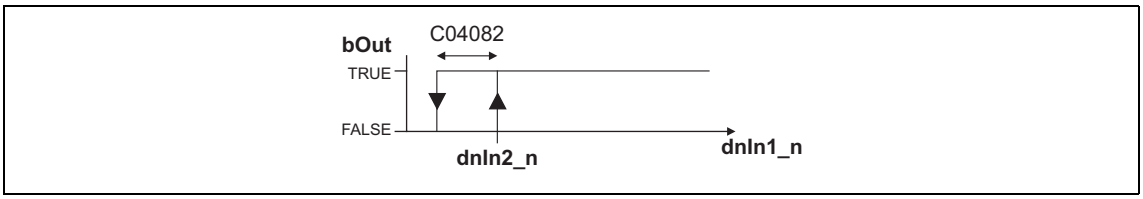


[5-120] Function 1: Example

5.161.2 Function 2: dnIn1_n > dnIn2_n

Selection: C04080 = "1"

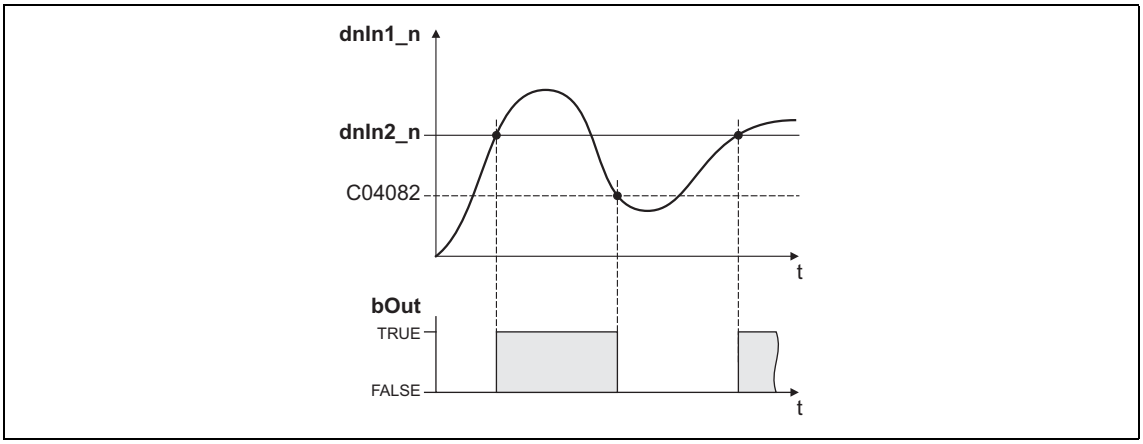
- This function serves, for instance, to implement the comparison "actual speed is higher than a limit value" ($n_{act} > n_x$) for one direction of rotation.



[5-121] Function 2: Switching performance

Functional sequence

1. If the value at d*nIn1_n* exceeds the value at *dnIn2_n*, *bOut* changes from FALSE to TRUE.
2. Only if the signal at *dnIn1_n* falls below the value at *dnIn2_n* - C04082 again, *bOut* changes from TRUE to FALSE.

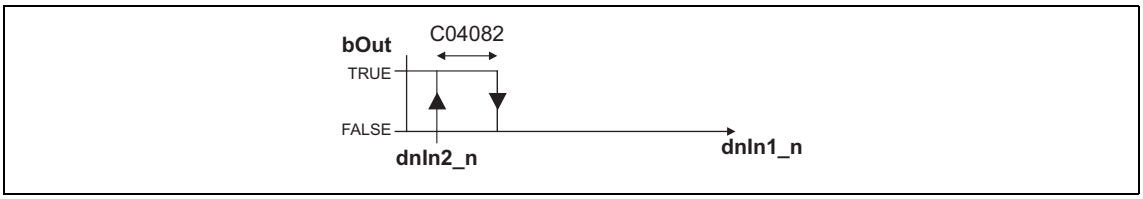


[5-122] Function 2: Example

5.161.3 Function 3: dnIn1_n < dnIn2_n

Selection: C04080 = "2"

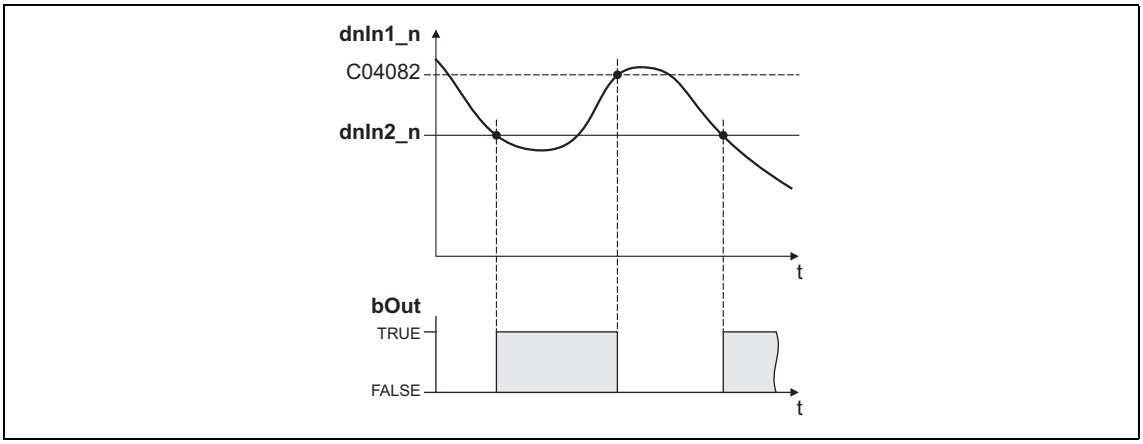
- This function serves, for instance, to implement the comparison "actual speed is lower than a limit value" (n_{act} n_x) for one direction of rotation.



[5-123] Function 3: Switching performance

Functional sequence

1. If the value at *dnIn1_n* falls below the value at *dnIn2_n*, *bOut_b* changes from FALSE to TRUE.
2. Only if the signal at *dnIn1_n* exceeds the value at *dnIn2_n* - C04082 again, *bOut_b* changes from TRUE to FALSE.



[5-124] Function 3: Example

5.161.4 Function 4: |dnIn1_n| = |dnIn2_n|

Selection: C04080 = "3"

- This function serves, for instance, to implement the comparison "$n_{act} = 0$".
- This function behaves like function 1. However, the amount is generated by the input signals before signal processing (without sign).
  ▸ Function 1: dnIn1_n = dnIn2_n (📖 559)

5.161.5 Function 5: |dnIn1_n| > |dnIn2_n|

Selection: C04080 = "4"

- This function serves, for instance, to implement the comparison "$|n_{act}| > |n_x|$" independent of the direction of rotation.
- This function behaves like function 2. However, the amount is generated by the input signals before signal processing (without sign).
  ▸ Function 2: dnIn1_n > dnIn2_n (📖 560)

5.161.6 Function 6: |dnIn1_n| < |dnIn2_n|

Selection: C04080 = "5"

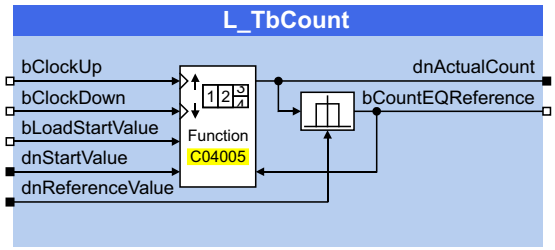
- This function serves to implement the comparison "$|n_{act}| < |n_x|$" independent of the direction of rotation.
- This function behaves like function 3. However, the amount is generated by the input signals before signal processing (without sign).
  ▸ Function 3: dnIn1_n < dnIn2_n (📖 561)

5.162 L_TbCount - up/downcounter

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB is a digital up/downcounter with adjustable limitation.

- Under C04005 you can select if the counter is to be stopped or reset automatically when reaching the limitation.



Inputs

Identifier/data type	Information/possible settings	
bClockUp BOOL	Upcounter	
	FALSE↗TRUE	Increase counter by "1".
bClockDown BOOL	Downcounter	
	FALSE↗TRUE	Reduce counter by "1".
bLoadStartValue BOOL	Accept starting value • This input has the highest priority.	
	TRUE	The starting value specified via *dnStartValue* is loaded into the counter.
dnStartValue DINT	Starting value	
dnReferenceValue DINT	Comparison value	

Outputs

Identifier/data type	Value/meaning	
dnActualCount DINT	Current counter content	
bCountEQReference BOOL	Result of the comparison of current counter content and comparison value	
	TRUE	$\lvert dnActualCount \rvert \geq \lvert dnReferenceValue \rvert$

Parameter

Parameter	Possible settings		Information
C04005			Reset function
	0	After reaching the reference value the counter is automatically reset to the starting value.	Lenze setting
	1	The reached reference value is maintained until a reset via *bLoadStartValue* is executed.	

5.162.1 Function 1: Automatic counter reset

Selection: C04005 = "0"

Functional sequence

1. If the amount of the current counter content ≥ the amount of the comparison value, the output *bCountEQReference* is set to TRUE (|*dnActualCount*| ≥ |*dnReferenceValue*|).
2. In the next cycle the counter is reset to the starting value *dnStartValue* and the output *bCountEQReference* is reset to FALSE.

5.162.2 Function 2: Counter stop at upper limit

Selection: C04005 = "1"

Functional sequence

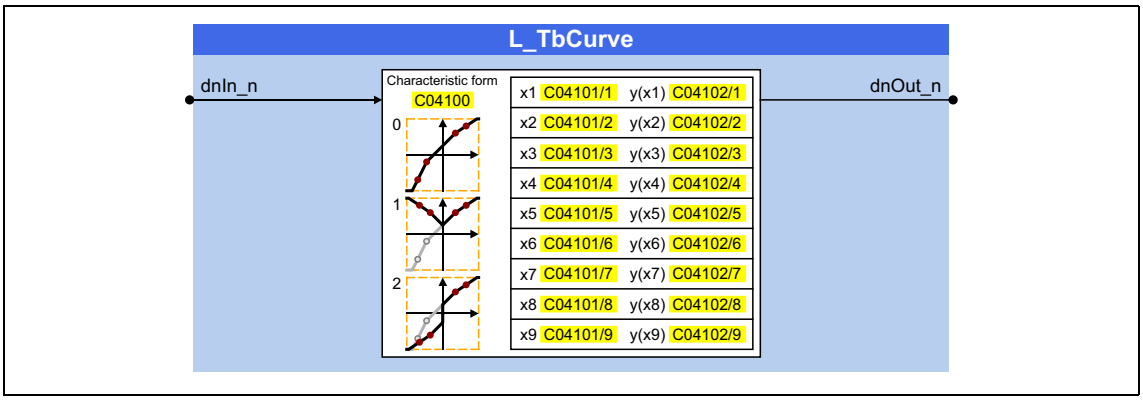
1. If the amount of the current counter content ≥ the amount of the comparison value, the output *bCountEQReference* is set to TRUE (|*dnActualCount*| ≥ |*dnReferenceValue*|).
2. The counter inputs *bClockUp* and *bClockDown* are deactivated at the same time, i.e. counting upwards or downwards is not possible anymore.
3. By setting *bLoadStartValue* to TRUE the counter is reset to the starting value *dnStartValue* and the counter inputs *bClockUp* and *bClockDown* are evaluated again.

5.163 L_TbCurve - characteristic function

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB represents a characteristic/curve function y = f(x). Here, the input signal corresponds to the x axis and the output signal corresponds to the y axis.

- The characteristic can consist of up to nine points which are defined by parameters.
- Between the points a linear interpolation takes place.
- The characteristic can be entered over the entire value range and can also be created symmetrically to the y axis or zero point by selecting the corresponding characteristic mode in C04100.



Inputs

Identifier/data type	Information/possible settings
dnIn_n DINT	Input signal

Outputs

Identifier/data type	Value/meaning
dnOut_n DINT	Output signal

Parameter

Parameter	Possible settings		Information
C04100			Characteristic mode
	0	Entire range	Lenze setting
	1	Symmetrically to the y axis	
	2	Symmetrically to the zero point	

Parameter	Possible settings			Information
C04101/1...9	-200.00	%	200.00	X values of the characteristic function • Subcodes 1 ... 9 correspond to the point values X1 ... X9. • Initialisation: 0.00 %
C04102/1...9	-200.00	%	200.00	Y values of the characteristic function • Subcodes 1 ... 9 correspond to the point values Y1 ... Y9. • Initialisation: 0.00 %

5.163.1 Selection of the characteristic

The up to nine points of the characteristic are selected via the subcodes of C04101 and C04102.

- The same subcodes of C04101 and C04102 correspond to a pair of variates/point (x_n, y_n).

	1	2	...	9
x	C04101/**1**	C04101/**2**	C04101/...	C04101/**9**
Y	C04102/**1**	C04102/**2**	C04102/...	C04102/**9**

- The first pair of variates (C04101/1 and C04102/1) is always valid.
- The X values of the characteristic must be entered in ascending order (X1 < X2 < ... < X9).
- If the ascending chain of x values is interrupted, this corresponds to the end of the characteristic (in the example X5 < X4):

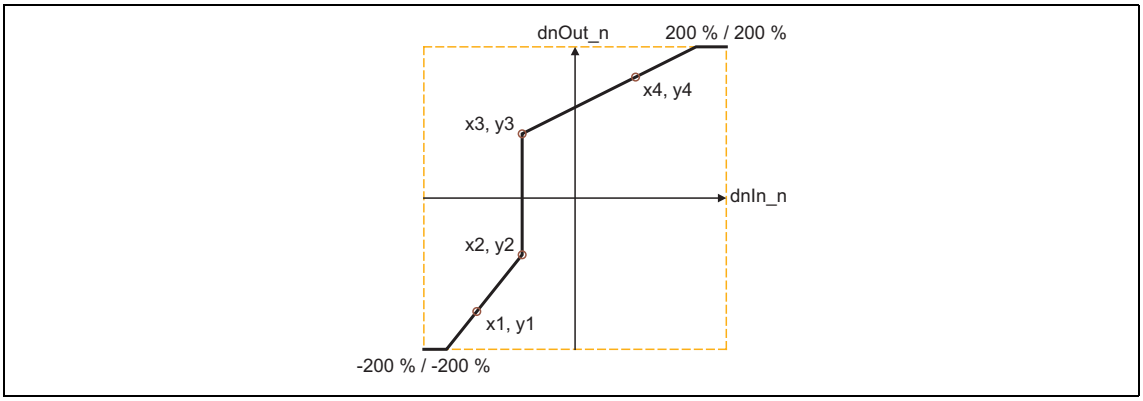
	1	2	3	4	5	6	7	8	9
x	-140 %	-90 %	70 %	145 %	0 %	0 %	0 %	0 %	0 %
Y	-140 %	-40 %	115 %	160 %	0 %	0 %	0 %	0 %	0 %

- Based on the last valid pair of variates, an extrapolation is executed to the end of the x value range (±200 %) on both sides.
- In the characteristic modes 1 and 2, only the range of the characteristic entered in the quadrants I. and IV. is used.

Special cases

- Jumps/discontinuities can be displayed by parameterising two successive x values the same way. A jump is only valid if a valid pair of variates exists after that, otherwise the extrapolation is executed starting from the last valid pair of variates.

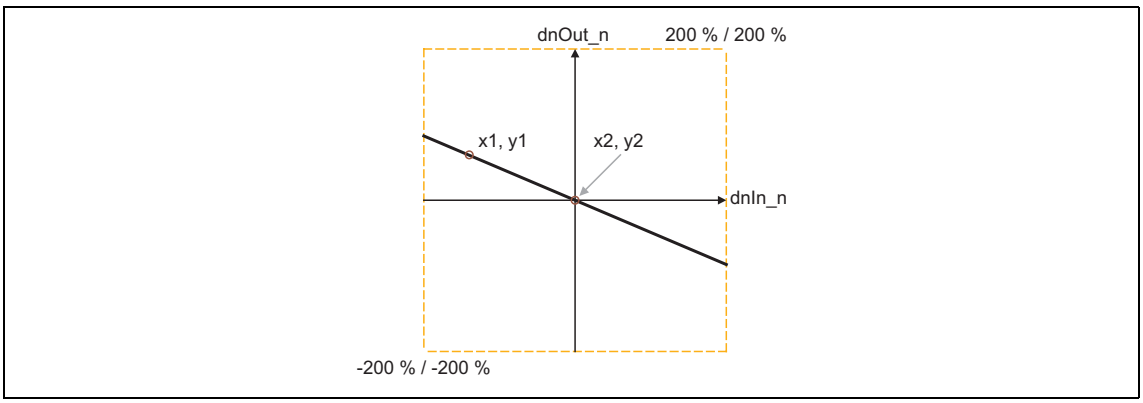
	1	2	3	4	5	6	7	8	9
x	-130 %	-70 %	-70 %	80 %	0 %	0 %	0 %	0 %	0 %
y(x)	-150 %	-75 %	85 %	160 %	0 %	0 %	0 %	0 %	0 %



[5-125] Example: Characteristic in the characteristic mode 0 (entire range) with jump

- If only one point X1 is defined in the negative range, X2 exists with the Lenze setting "0 %" as valid point (X2 > X1) and both points form a line:

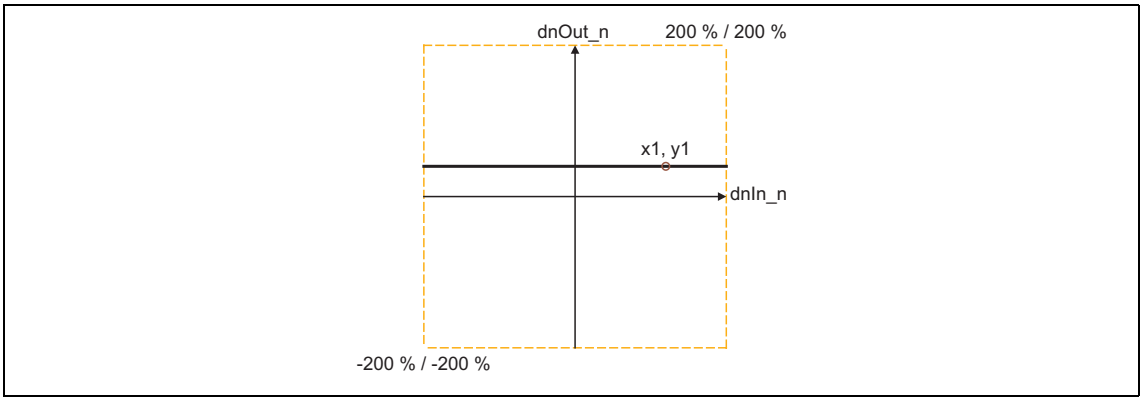
	1	2	3	4	5	6	7	8	9
x	-130 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
y(x)	40 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %



[5-126] Example: Characteristic in characteristic mode 0 (entire range) with only one defined point and X1 < 0

- If only one point X1 is defined in the positive range, a line is extrapolated:

	1	2	3	4	5	6	7	8	9
x	120 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
y(x)	40 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %



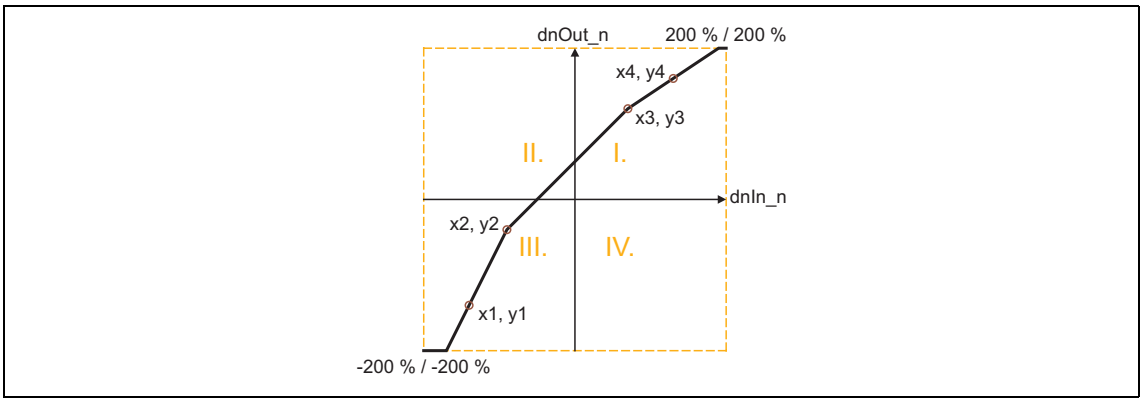
[5-127] Example: Characteristic in characteristic mode 0 (entire range) with only one defined point and X1 < 0

5.163.2 Characteristic mode 0: Entire range

If "Entire range" is selected in C04100, the entire characteristic range (quadrants I ... IV) is valid.

Example

	1	2	3	4	5	6	7	8	9
x	-140 %	-90 %	70 %	130 %	0 %	0 %	0 %	0 %	0 %
y(x)	-140 %	-40 %	120 %	160 %	0 %	0 %	0 %	0 %	0 %



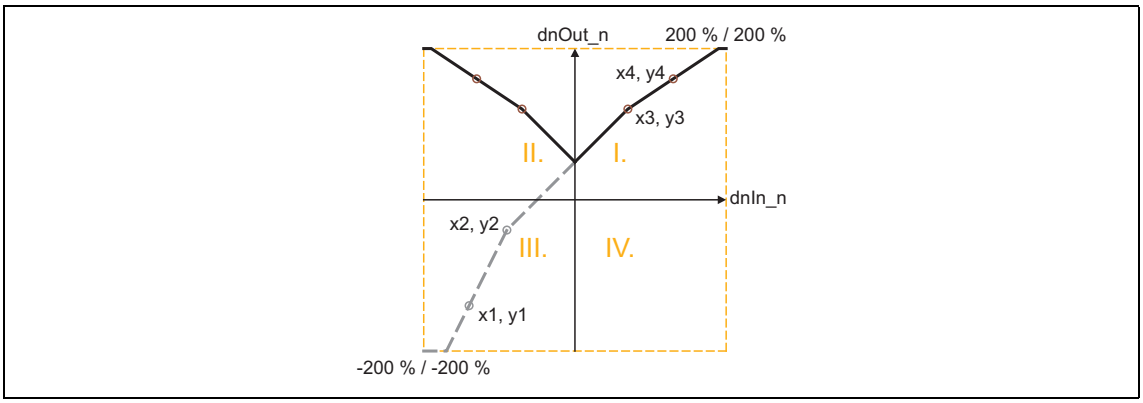
[5-128] Example: Characteristic in characteristic mode 0 (entire range)

5.163.3 Characteristic mode 1: Symmetrically to y axis

If "Symmetrically to y axis" is selected in C04100, the resulting characteristic is mirrored along the y axis in the I. and IV. quadrant.

Example

	1	2	3	4	5	6	7	8	9
x	-140 %	-90 %	70 %	130 %	0 %	0 %	0 %	0 %	0 %
y(x)	-140 %	-40 %	120 %	160 %	0 %	0 %	0 %	0 %	0 %



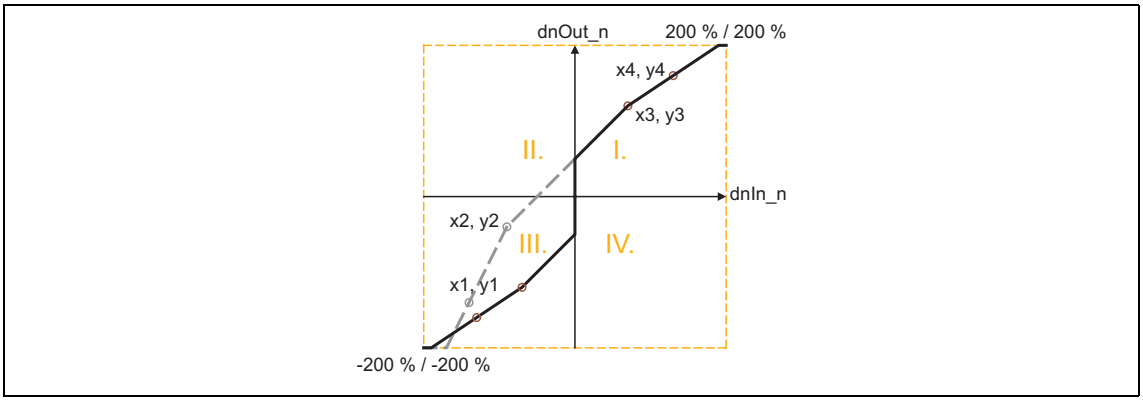
[5-129] Example: Characteristic in characteristic mode 1 (symmetrically to y axis)

5.163.4 Characteristic mode 2: Symmetrically to zero point

If "Symmetrically to zero point" is selected in C04100, the resulting characteristic is mirrored along the zero point in the I. and IV. quadrant.

Example

	1	2	3	4	5	6	7	8	9
x	-140 %	-90 %	70 %	130 %	0 %	0 %	0 %	0 %	0 %
y(x)	-140 %	-40 %	120 %	160 %	0 %	0 %	0 %	0 %	0 %



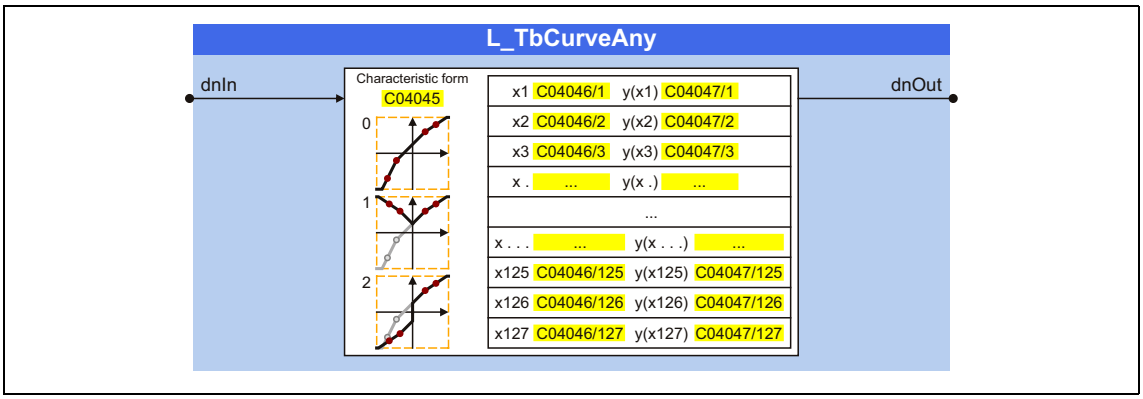
[5-130] Example: Characteristic in characteristic mode 2 (symmetrically to zero point)

5.164 L_TbCurveAny - Characteristic function

Function library:	LenzeToolbox	**FB is available as of library V02.06.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB represents a characteristic/curve function y = f(x). Here, the input signal corresponds to the x axis and the output signal corresponds to the y axis.

- The characteristic can consist of up to 127 points which are defined by parameters.
- Between the points a linear interpolation takes place.
- The characteristic can be entered over the entire value range and can also be created symmetrically to the y axis or zero point by selecting the corresponding characteristic mode in C04045.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal

Outputs

Identifier/data type	Value/meaning
dnOut DINT	Output signal

Parameters

Parameter	Possible settings		Info
C04045			Characteristic mode
	0	Entire range	Lenze setting
	1	Symmetrically to the y axis	
	2	Symmetrically to the zero point	

Parameter	Possible settings			Info
C04046/1...127	-2147483648		2147483647	X values of the characteristic function • Subcodes 1 ... 127 correspond to the point values X1 ... X127. • Initialisation: 0
C04047/1...127	-2147483648		2147483647	Y values of the characteristic function • Subcodes 1 ... 127 correspond to the point values Y1 ... Y127. • Initialisation: 0

Tip!

For a better understanding, in the following examples, the converted percentages referring to the limit values 2^{31}-1 and -(2^{31}) are entered instead of the real DINT values.

5.164.1 Selection of the characteristic

The up to 127 points of the characteristic are selected via the subcodes of C04046 and C04047.

- The same subcodes of C04046 and C04047 correspond to a pair of variates/point (x_n, y_n).

	1	2	...	127
x	C04046/**1**	C04046/**2**	C04046/...	C04046/127
Y	C04047/**1**	C04047/**2**	C04047/...	C04047/127

- The first pair of variates (C04046/1 and C04047/1) is always valid.
- The x values of the characteristic must be entered in ascending order (x1 < x2 < ... < x127).
- If the ascending chain of x values is interrupted, this corresponds to the end of the characteristic (in the example X5 < X4):

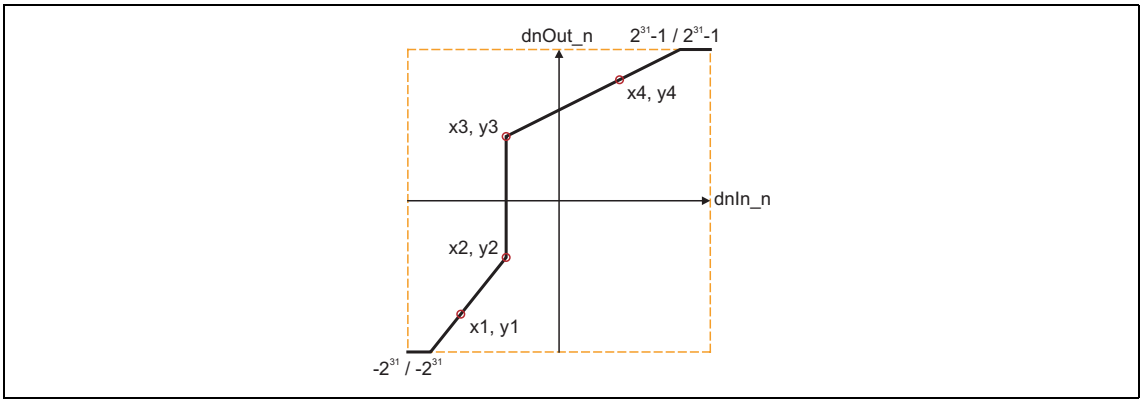
	1	2	3	4	5	6	...	126	127
x	-140 %	-90 %	70 %	145 %	0	0	0	0	0
y	-140 %	-40 %	115 %	160 %	0	0	0	0	0

- Based on the last valid pair of variates, an extrapolation is executed to the end of the x value range (2^{31})-1 on both sides.
- In the characteristic modes 1 and 2, only the range of the characteristic entered in the quadrants I. and IV. is used.

Special cases

- Jumps/discontinuities can be displayed by parameterising two successive x values the same way. A jump is only valid if a valid pair of variates exists after that, otherwise the extrapolation is executed starting from the last valid pair of variates.

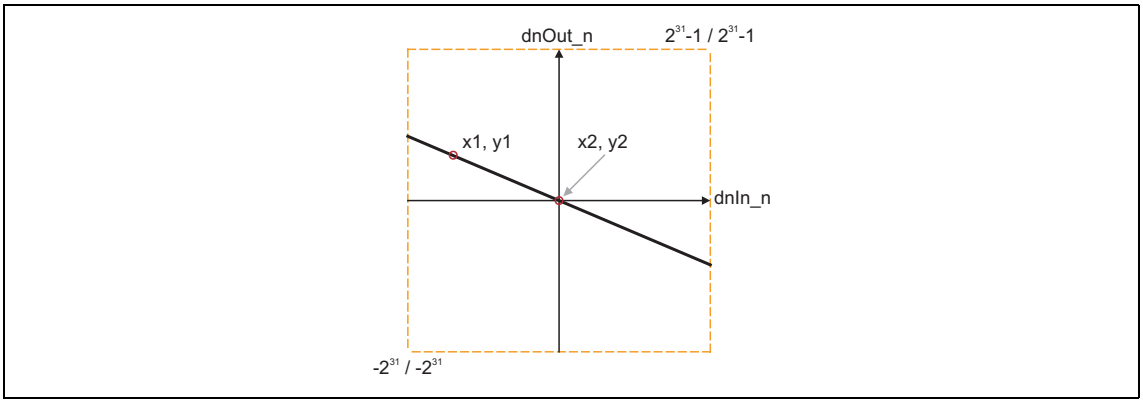
	1	2	3	4	5	6	...	126	127
x	-130 %	-70 %	-70 %	80 %	0	0	0	0	0
y(x)	-150 %	-75 %	85 %	160 %	0	0	0	0	0



[5-131] Example: Characteristic in the characteristic mode 0 (entire range) with jump

- If only one point X1 is defined in the negative range, X2 exists with the Lenze setting "0" as valid point (X2 > X1) and both points form a line:

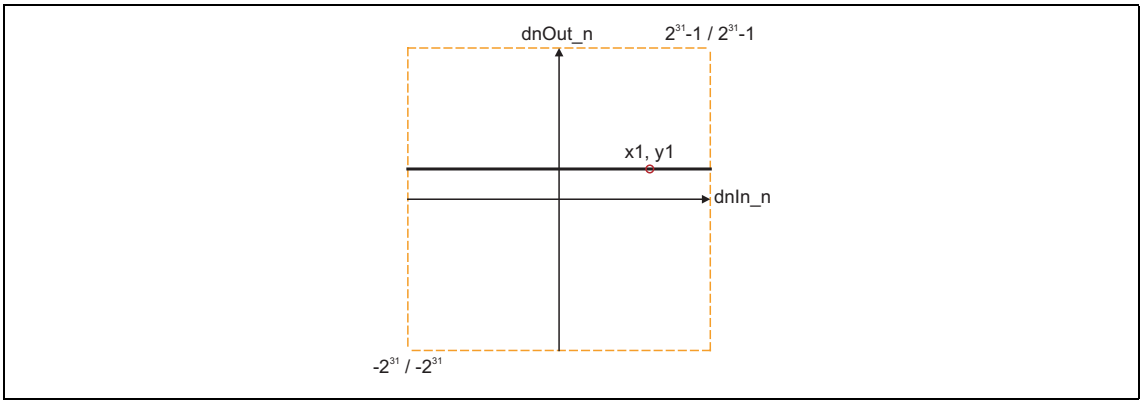
	1	2	3	4	5	6	...	126	127
x	-130 %	0	0	0	0	0	0	0	0
y(x)	40 %	0	0	0	0	0	0	0	0



[5-132] Example: Characteristic in characteristic mode 0 (entire range) with only one defined point and X1 < 0

- If only one point X1 is defined in the positive range, a line is extrapolated:

	1	2	3	4	5	6	...	126	127
x	120 %	0	0	0	0	0	0	0	0
y(x)	40 %	0	0	0	0	0	0	0	0



[5-133] Example: Characteristic in characteristic mode 0 (entire range) with only one defined point and X1 < 0

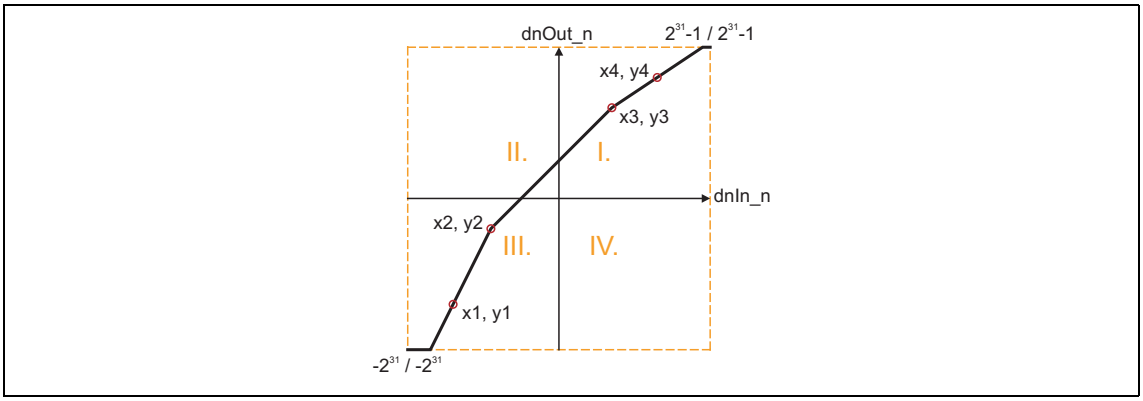
5.164.2 Characteristic mode 0: Entire range

If "Entire range" is selected in C04045, the entire characteristic range (quadrants I ... IV) is valid.

Note: For a better understanding, the corresponding percentages are entered instead of the real DINT values.

Example

	1	2	3	4	5	6	...	126	127
x	-140 %	-90 %	70 %	130 %	0	0	0	0	0
y(x)	-140 %	-40 %	120 %	160 %	0	0	0	0	0



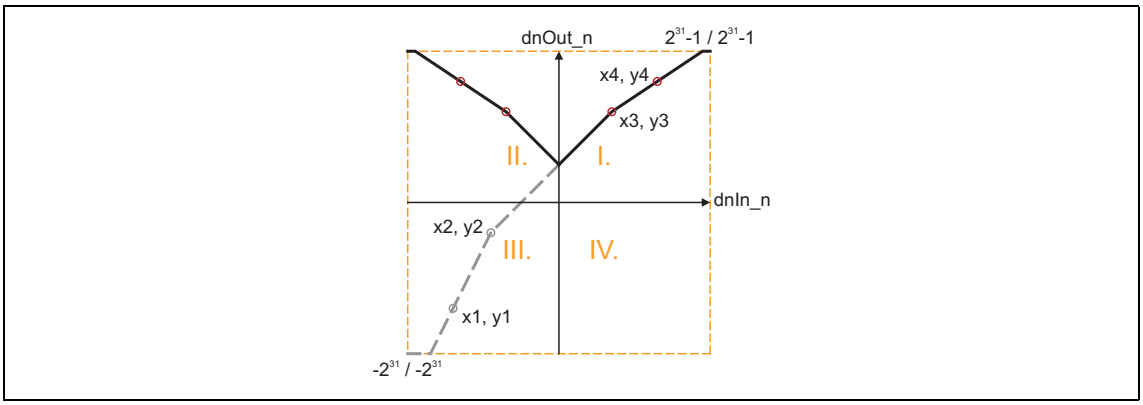
[5-134] Example: Characteristic in characteristic mode 0 (entire range)

5.164.3 Characteristic mode 1: Symmetrically to y axis

If "Symmetrically to y axis" is selected in C04045, the resulting characteristic is mirrored along the y axis in the I. and IV. quadrant.

Example

	1	2	3	4	5	6	...	126	127
x	-140 %	-90 %	70 %	130 %	0	0	0	0	0
y(x)	-140 %	-40 %	120 %	160 %	0	0	0	0	0



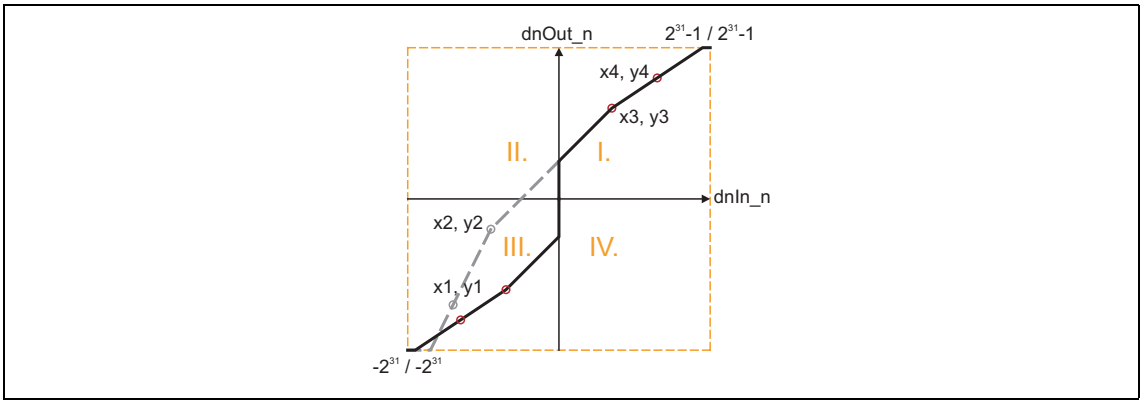
[5-135] Example: Characteristic in characteristic mode 1 (symmetrically to y axis)

5.164.4 Characteristic mode 2: Symmetrically to zero point

If "Symmetrically to zero point" is selected in C04045, the resulting characteristic is mirrored along the zero point in the I. and IV. quadrant.

Example

	1	2	3	4	5	6	...	126	127
x	-140 %	-90 %	70 %	130 %	0	0	0	0	0
y(x)	-140 %	-40 %	120 %	160 %	0	0	0	0	0



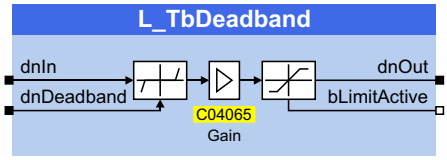
[5-136] Example: Characteristic in characteristic mode 2 (symmetrically to zero point)

5.165 L_TbDeadband - dead band with gain

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB produces a symmetrical dead band around zero.

- The attenuation of the input signal caused by the dead band can be compensated by a parameterisable gain.
- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.
- The *bLimitActive* output displays if a limitation is active.
- The function corresponds by approximation to the "DB" function of the 9300 servo inverter.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal
dnDeadband DINT	Dead band area for the input signal • Value is internally limited to ≥0.

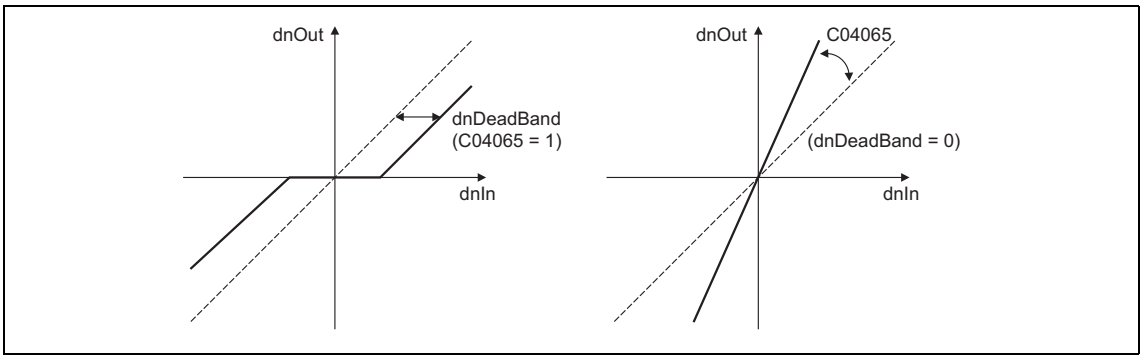
Outputs

Identifier/data type	Value/meaning	
dnOut DINT	Output signal • If |*dnIn*| ≤ *dnDeadband* the output is set to zero. • If |*dnIn*| > *dnDeadband* the difference multiplied by the parameterised gain is output as output signal. • The sign results from the sign of *dnIn* and the sign of the selected gain. • Internally limited to $\pm 2^{31}$-1	
bLimitActive BOOL	Output signal within the limitation	
	TRUE	The output signal *dnOut* is limited.

Parameter

Parameter	Possible settings			Information
C04065	-214748.3647		214748.3647	Gain • Initialisation: 1.0000

Function



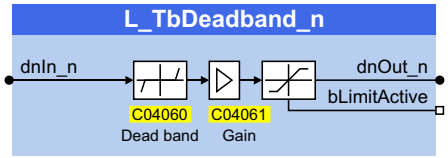
[5-137] Dead band and gain

5.166 L_TbDeadband_n - scaled dead band with gain

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB produces a symmetrical dead band around zero.

- The attenuation of the input signal caused by the dead band can be compensated by a parameterisable gain.
- The value output to *dnOut* is internally limited to ±200 %.
- The *bLimitActive* output displays if a limitation is active.
- The function corresponds by approximation to the "DB" function of the 9300 servo inverter.



Inputs

Identifier/data type	Information/possible settings
dnIn_n DINT	Input signal

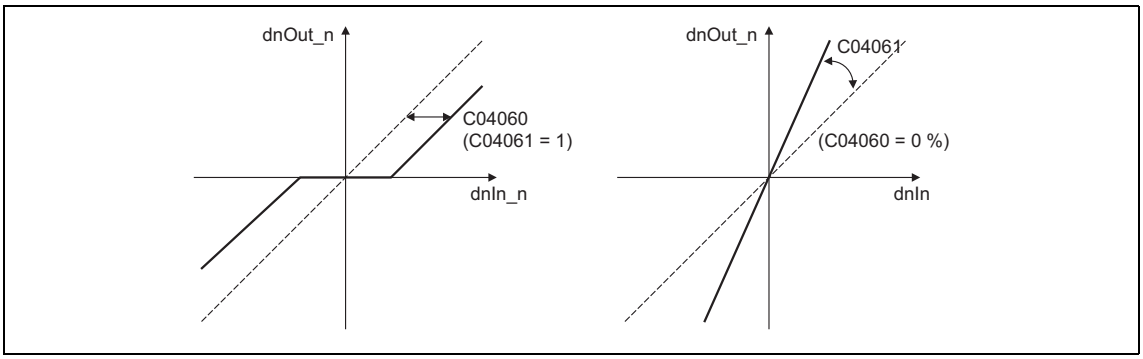
Outputs

Identifier/data type	Value/meaning
dnOut_n DINT	Output signal • If |*dnIn_n*| ≤ C04060 the output is set to zero. • If |*dnIn_n*| > C04060 the difference multiplied by the parameterised gain is output as output signal. • The sign results from the sign of *dnIn_n* and the sign of the selected gain. • Internally limited to ±200 %
bLimitActive BOOL	Output signal within the limitation TRUE: The output signal *dnOut_n* is limited.

Parameter

Parameter	Possible settings			Information
C04060	0.00	%	100.00	Dead band • Initialisation: 1.00 %
C04061	-214748.3647		214748.3647	Gain • Initialisation: 1.0000

Function



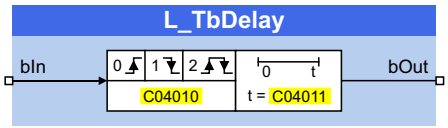
[5-138] Dead band and gain

5.167 L_TbDelay - delay

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB delays binary signals.

- The edge to be delayed is selected via the parameter C04010.
- The delay time is set via the parameter C04011.
- The function corresponds to the "DIGDEL" function of the 9300 servo inverter.



Inputs

Identifier/data type	Information/possible settings
bin BOOL	Input

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Output with time-delayed input signal

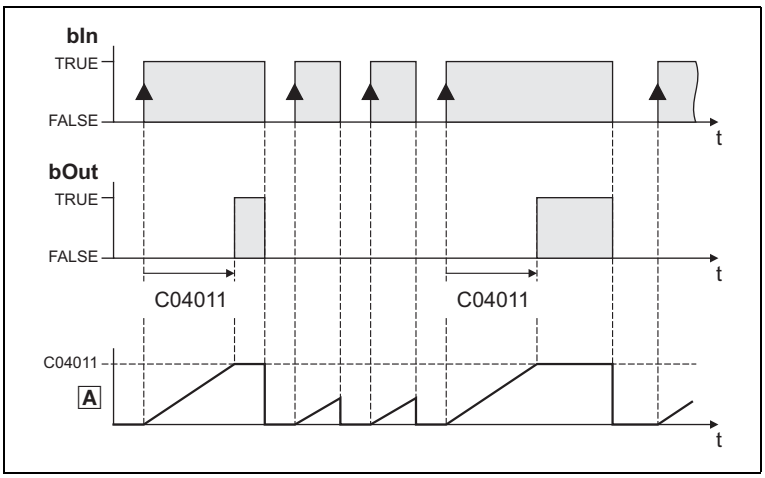
Parameter

Parameter	Possible settings			Information
C04010				Selection of the edge to be delayed
	0	Rising edge		Lenze setting
	1	Falling edge		
	2	Both edges		
C04011	0.000	s	60.000	Delay time • Initialisation: 0.001 s

5.167.1 Function 1: ON-delay

Selection: C04010 = "0" (Lenze setting)

- In this mode, the FB operates like a retriggerable monoflop.



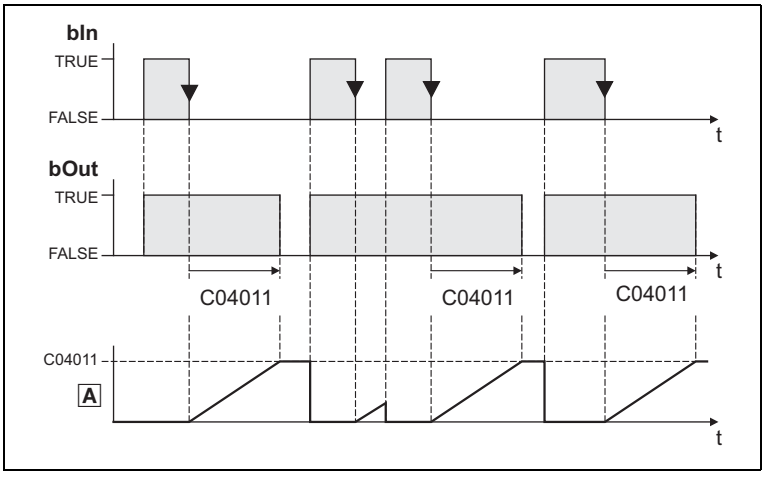
A = internal timing element

Functional sequence

1. A FALSE-TRUE edge at *bIn* starts the internal timing element (A).
2. After the delay time set via C04011 has elapsed, *bOut* is set to TRUE.
3. A TRUE-FALSE edge at b*In* sets *bOut* to FALSE again. Simultaneously, the internal timing element is reset.

5.167.2 Function 2: OFF-delay

Selection: C04010 = "1"



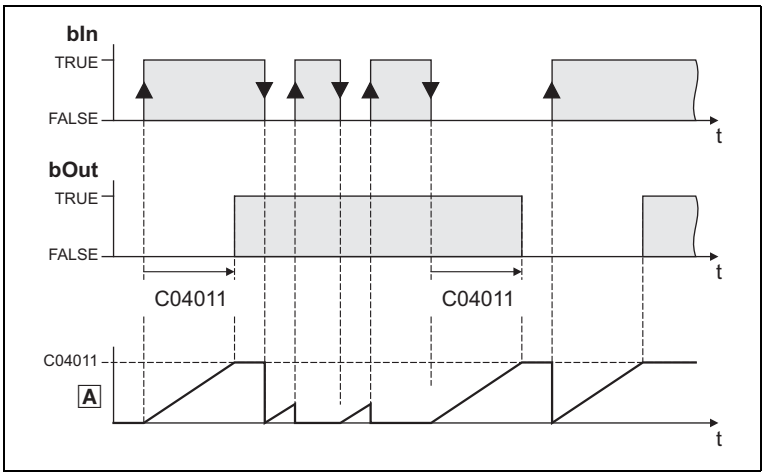
A = internal timing element

Functional sequence

1. A FALSE-TRUE edge at b*In* sets *bOut* to TRUE. Simultaneously, the internal timing element (A) is reset.
2. A TRUE-FALSE edge at b*In* starts the internal timing element.
3. After the delay time set via C04011 has elapsed, *bOut* is reset to FALSE.

5.167.3 Function 3: General delay

Selection: C04010 = "2"



A = internal timing element

Functional sequence

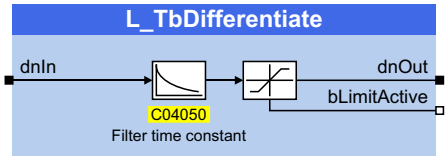
1. Any edge at b*In* resets and starts the internal timing element (A).
2. After the delay time set via C04011 has elapsed, the input signal *bIn* is output to *bOut*.

5.168 L_TbDifferentiate - differentiator with low-pass filter

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB represents a differentiator with low pass.

- The input signal is differentiated and filtered with the filter time constant set via the parameter C04050.
- The *dnOut* output signal has the unit [1/s] as time reference. ▸ Adapting the time reference (📖 584)
- The *dnOut* output signal is internally limited to $\pm 2^{31}$-1.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal

Outputs

Identifier/data type	Value/meaning	
dnOut DINT	Output signal • Internally limited to $\pm 2^{31}$-1	
bLimitActive BOOL	Status signal "Limitation active"	
	TRUE	The output signal is limited.

Parameter

Parameter	Possible settings			Information
C04050	0.001	s	60.000	Filter time constant T_{Filter} • Initialisation: 0.001 s

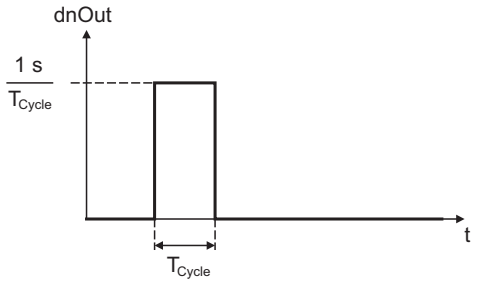
5.168.1 Function

Considering the signalling technique, the FB consists of the series connection of a time-discrete differentiator and a PT1 filter.

- The filtering of the output signal can be switched off with the setting of the filter time constant T_{Filter} to "0.001 s". In this case, the output signal results directly from:

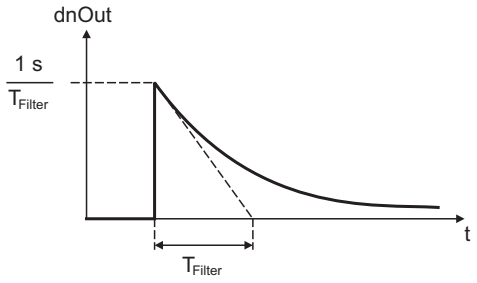
$$dnOut = \Delta dnIn \cdot \frac{1\ s}{\text{Task cycle time}}$$

[5-139] Formula for the output signal when the filter is switched off



[5-140] Step response when filter is switched off

- When the filter is switched on, each change of the input signal is evaluated with 1 s/T_{Filter}. Then the output signal tends to zero with the filter time constant T_{Filter}:



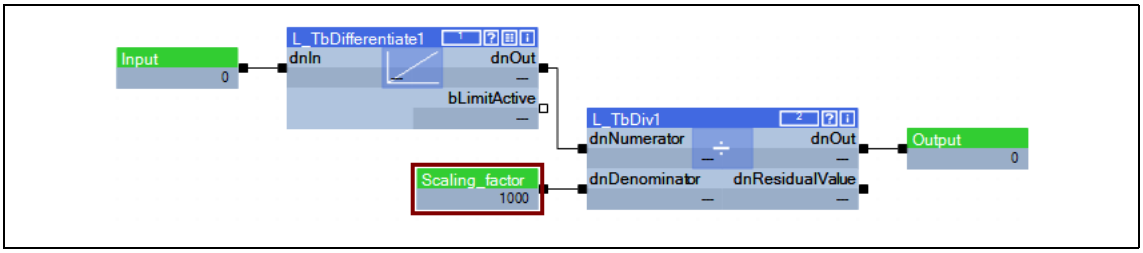
C04050: Filter time constant T_{Filter}

[5-141] Step response when the filter is switched on

5.168.2 Adapting the time reference

The *dnOut* output signal has the unit [1/s] as time reference.

The *dnOut* output signal can be adapted to the required unit by dividing the output signal by a corresponding scaling factor:



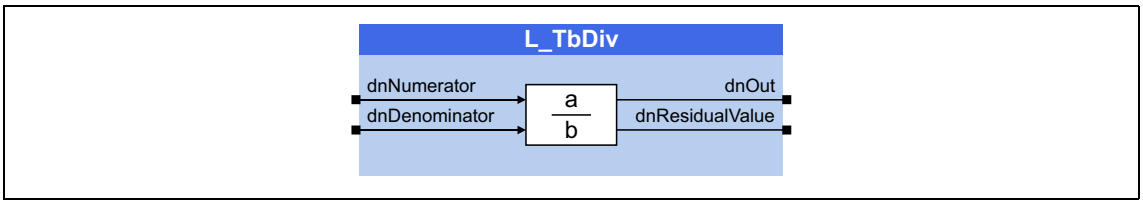
[5-142] Model connection for adapting the time reference

Example: Required unit of the signal *Output* = [1/ms] ➞ scaling factor = 1000

5.169 L_TbDiv - division

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a division with remainder.



Inputs

Identifier/data type	Information/possible settings
dnNumerator DINT	Dividend (numerator)
dnDenominator DINT	Divisor (denominator)

Outputs

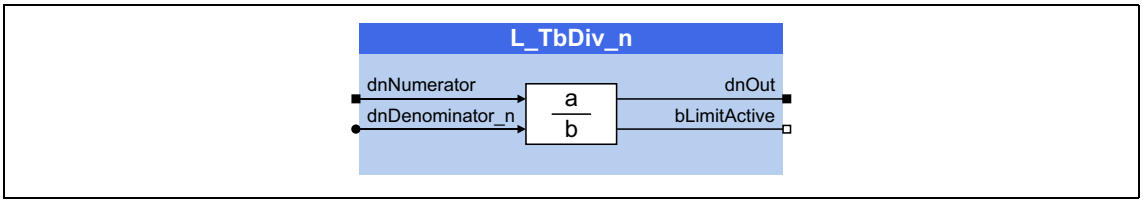
Identifier/data type	Value/meaning
dnOut DINT	Quotient value (integer result of division)
dnResidualValue DINT	Remainder

5.170 L_TbDiv_n - scaled division

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a division. The divisor (denominator) must be specified as a scaled signal [%].

- With a divisor of 100 % the dividend is output unchanged.
- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnNumerator DINT	Dividend (numerator)
dnDenominator_n DINT	Divisor (denominator) • Scaling: 100 % ≡ 2^{30} ≡ 1073741824

Outputs

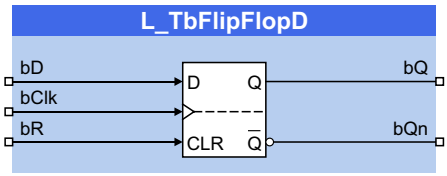
Identifier/data type	Value/meaning	
dnOut DINT	Quotient value (result of the division) • Internal limitation to $\pm 2^{31}$-1 (≡ ±200 %)	
bLimitActive BOOL	Status signal "Limitation active"	
	TRUE	The output signal is limited.

5.171 L_TbFlipFlopD - D flipflop

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB is created as a D flipflop and can be used to evaluate and save digital signal edges.

- With each rising edge at the clock input *bClk* the current status of the input *bD* is saved internally and output to the output *bOut*.
- The flipflop can be reset by a TRUE signal at the reset input *bR* which has the highest priority.



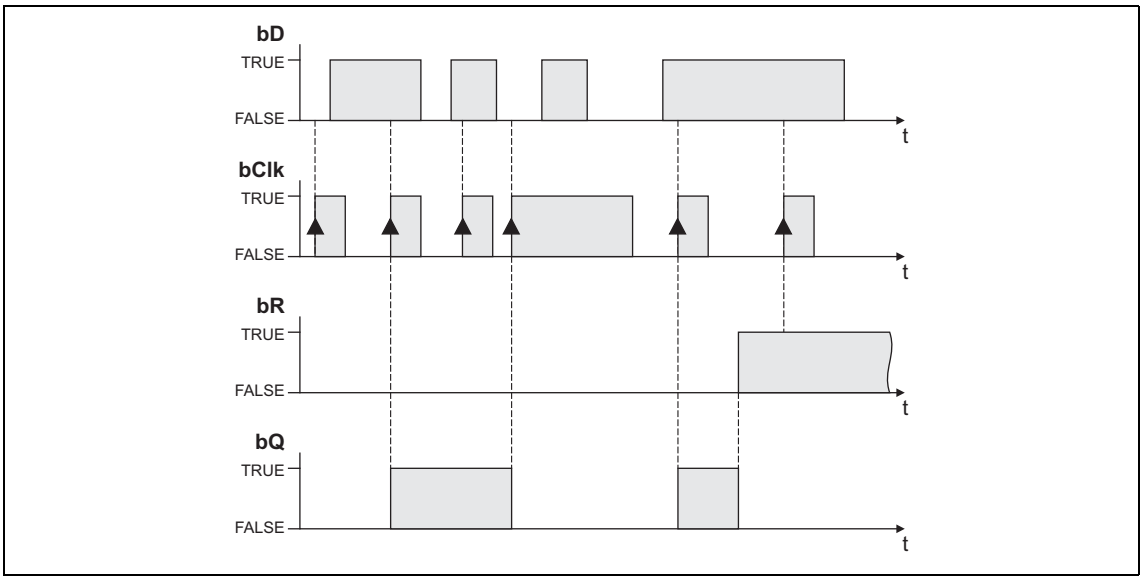
Inputs

Identifier/data type	Information/possible settings	
BD BOOL	Input	
bClk BOOL	Saving of input value	
	FALSE↗TRUE	The input value *bD* is saved internally and output to *bOut* until a new value is saved or the flipflop is reset via *bR*.
bR BOOL	Flipflop reset • This input has the highest priority.	
	TRUE	The output *bOut* is reset to FALSE.

Outputs

Identifier/data type	Value/meaning
bQ BOOL	State of the D flipflop
bQn BOOL	Inverted state of the D flipflop (≡ $\overline{bQ}$)

Function



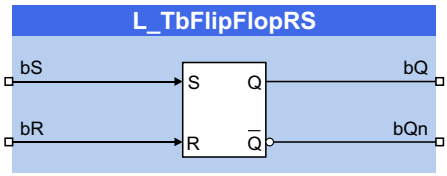
[5-143] Switching performance of the D flipflop

5.172 L_TbFlipFlopRS - status-controlled RS flipflop

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB is created as an RS flipflop and can be used to evaluate and save digital signals.

- The RS flipflop is set with a TRUE signal at the set input *bS* and reset with a TRUE signal at the reset input *bR*.
- If both inputs are applied with a FALSE signal, the output state is maintained.



Inputs

Identifier/data type	Information/possible settings	
bS BOOL	Set input	
	TRUE	Set RS flipflop.
bR BOOL	Reset input	
	TRUE	Reset RS flipflop.

Outputs

Identifier/data type	Value/meaning
bQ BOOL	State of the RS flipflop
bQn BOOL	Inverted state of the RS flipflop (≡ $\overline{bQ}$)

Function

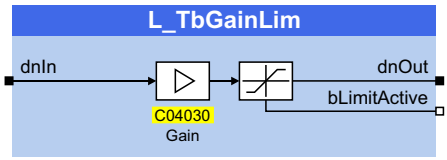
bS	bR	bQ	bQn
FALSE	FALSE	unchanged	unchanged
TRUE	FALSE	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE
TRUE	TRUE	FALSE	TRUE

5.173 L_TbGainLim - gain with limitation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB amplifies the input signal.

- The gain is set via the parameter C04030.
- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal

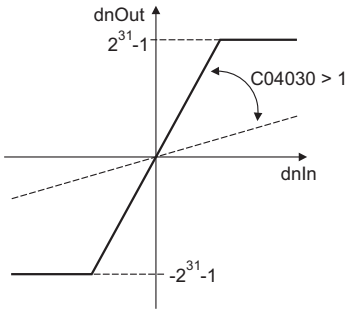
Outputs

Identifier/data type	Value/meaning	
dnOut DINT	Output signal • Internally limited to $\pm 2^{31}$-1	
bLimitActive BOOL	Output signal within the limitation	
	TRUE	The output signal *dnOut* is limited.

Parameter

Parameter	Possible settings			Information
C04030	-214748.3647		214748.3647	Gain • Initialisation: 1.0000

Function



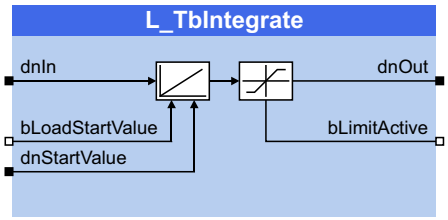
[5-144] Gain and limitation

5.174 L_TbIntegrate - integration with limitation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out an integration with limitation.

- The *dnIn* input signal has the unit [1/s] as time reference. ▸ Adapting the time reference (📖 592)
- The *dnOut* output signal is internally limited to $\pm 2^{31}$-1.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal
bLoadStartValue BOOL	Loading of the integrator TRUE: Load the integrator with the value at the *dnStartValue* input.
dnStartValue DINT	Value with which the integrator is loaded by setting *bLoadStartValue* to TRUE.

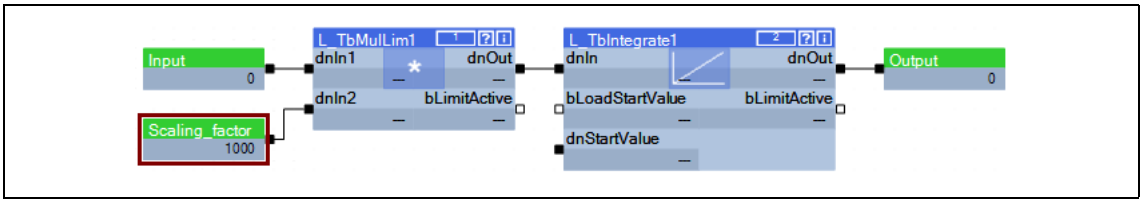
Outputs

Identifier/data type	Value/meaning
dnOut DINT	Output signal • Internally limited to $\pm 2^{31}$-1
bLimitActive BOOL	Status signal "Limitation active" TRUE: The output signal is limited.

5.174.1 Adapting the time reference

The *dnIn* input signal has the unit [1/s] as time reference.

An input signal can easily be adapted to the expected unit [1/s] by multiplying it by a corresponding scaling factor:



[5-145] Model connection for adapting the time reference

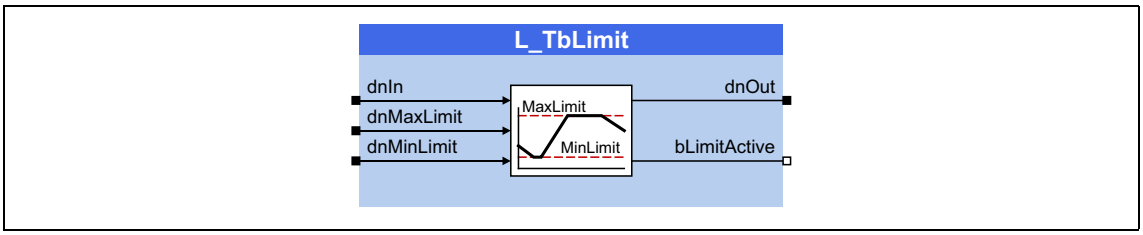
Example: Time reference of the *Input* input signal = [1/ms] ➞ scaling factor = 1000

5.175 L_TbLimit - limitation (for "DINT" data type)

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB limits a signal of "DINT" data type to an adjustable value range.

- The upper and lower limits of the value range are set via the inputs *dnMaxLimit* and *dnMinLimit*.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal
dnMaxLimit DINT	Upper limitation
dnMinLimit DINT	Lower limitation

Outputs

Identifier/data type	Value/meaning	
dnOut DINT	Output signal	
bLimitActive BOOL	Output signal within the limitation	
	TRUE	The output signal *dnOut* is limited.

Tip!

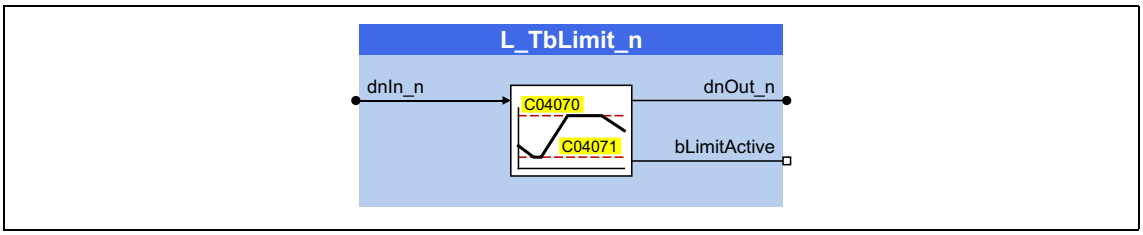
The lower limit must always be set smaller than the upper limit, otherwise the value "0" is output at *dnOut*.

5.176 L_TbLimit_n - scaled limitation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB limits a scaled signal [%] to an adjustable value range.

- The upper and lower limits of the value range are set via the parameters C04070 and C04071.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnIn_n DINT	Input signal • Scaling: 100 % ≡ 2^{30} ≡ 1073741824

Outputs

Identifier/data type	Value/meaning	
dnOut_n DINT	Output signal • Scaling: 100 % ≡ 2^{30} ≡ 1073741824	
bLimitActive BOOL	Output signal within the limitation	
	TRUE	The output signal *dnOut_n* is limited.

Parameter

Parameter	Possible settings			Information
C04070	-200.00	%	200.00	Upper limitation • Initialisation: 100.00 %
C04071	-200.00	%	200.00	Lower limitation • Initialisation: -100.00 %

Tip!

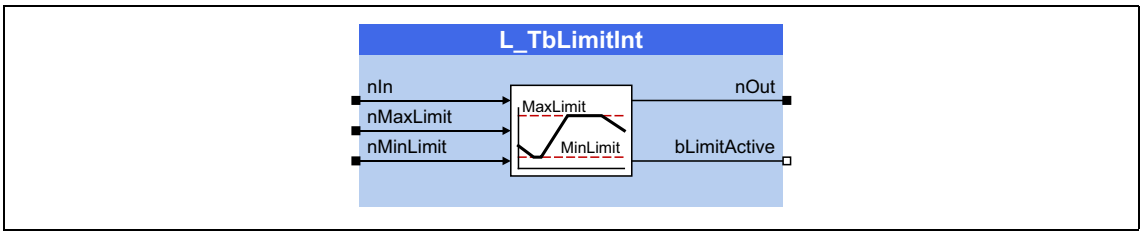
The lower limit must always be set smaller than the upper limit, otherwise the value "0" is output at *dnOut_n*.

5.177 L_TbLimitInt - limitation (for "INT" data type)

Function library:	LenzeToolbox	**FB is available as of library V02.05.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB limits a signal of "INT" data type to an adjustable value range.

- The upper and lower limits of the value range are set via the inputs *nMaxLimit* and *nMinLimit*.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
nIn INT	Input signal
nMaxLimit INT	Upper limitation
nMinLimit INT	Lower limitation

Outputs

Identifier/data type	Value/meaning	
nOut INT	Output signal	
bLimitActive BOOL	Output signal within the limitation	
	TRUE	The *nOut* output signal is limited.

Tip!

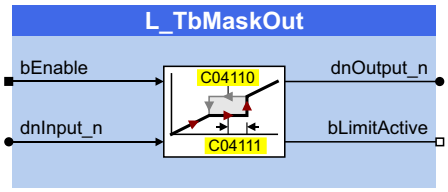
The lower limit must always be set smaller than the upper limit, otherwise the value "0" is output at *nOut*.

5.178 L_TbMaskOut - zone masking

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB masks up to four parameterisable blocking zones within a continuous signal characteristic.

- The definition of the upper and lower limits of the blocking zones is effected by specifying the mean value and the half width of the respective zone.
- The FB can be generally used (no measuring system reference).



Inputs

Identifier/data type	Information/possible settings	
bEnable BOOL	Activate zone masking	
	FALSE	The FB is deactivated. • The input signal is shown 1:1 at the output d*nOutput_n*.
	TRUE	The FB is activated. • A zone masking of the input signal according to the parameterised blocking zones is carried out.
dnInput_n DINT	Scaled input signal in [%]	

Outputs

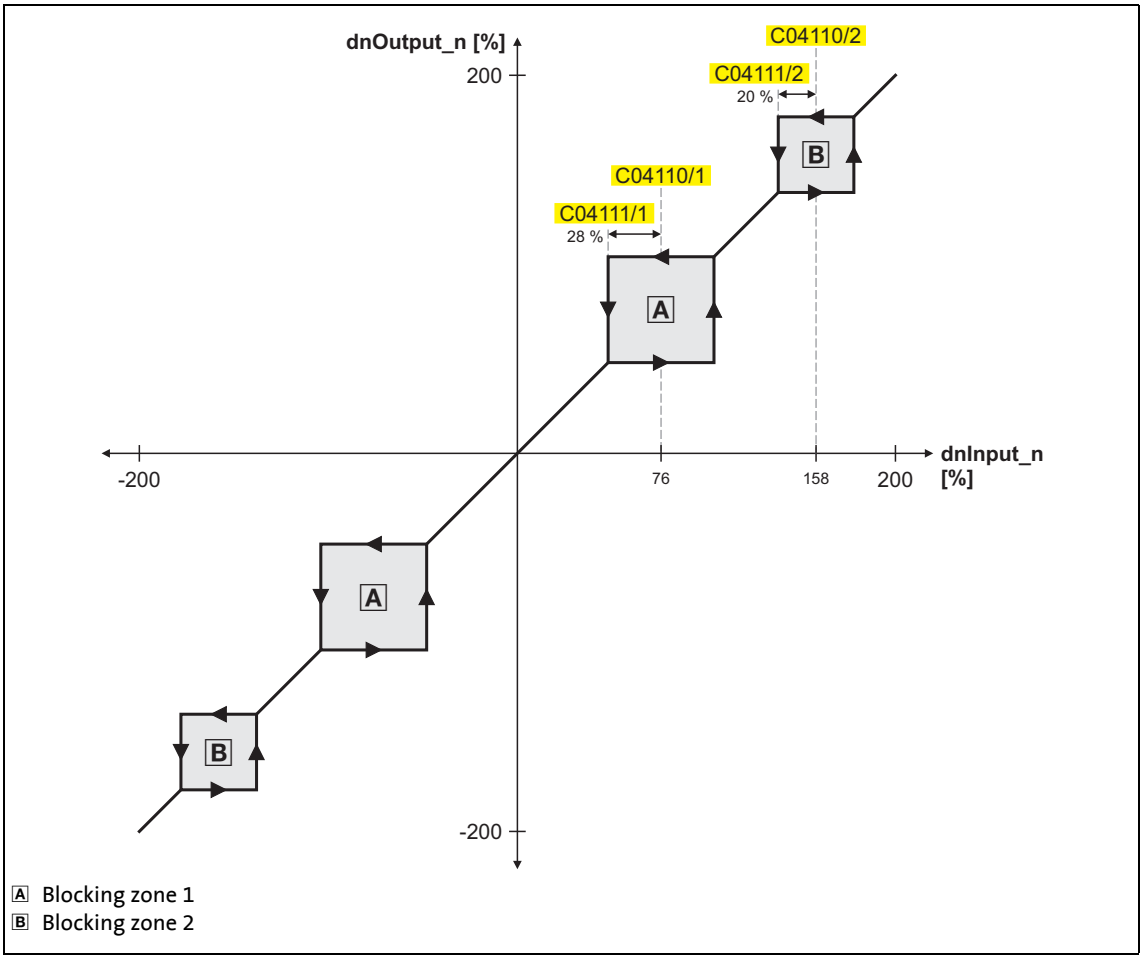
Identifier/data type	Value/meaning	
dnOutput_n DINT	Scaled output signal in [%] • If the FB is activated, the output signal is beyond the blocking zones.	
bLimitActive BOOL	Status signal "Limitation active"	
	TRUE	The input signal is within a blocking zone and is limited to the corresponding boundary value of the zone.

Parameter

Parameter	Possible settings			Information
C04110/1...4	0.00	%	200.00	Mean value of blocking zones 1 ... 4 • Initialisation: 0.00 %
C04111/1...4	0.00	%	100.00	Half width of blocking zones 1 ... 4 • Initialisation: 0.00 %

Function

The parameterised blocking zones act in the same way on negative input signals:



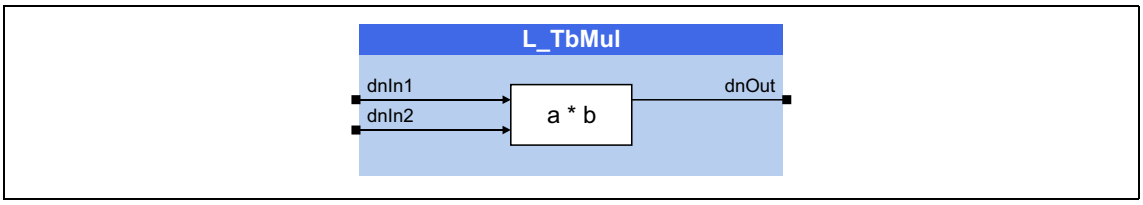
[5-146] Zone masking by means of parameterisable blocking zones

Parameter	Blocking zone 1	Blocking zone 2	Blocking zone 3	Blocking zone 4
Mean value	76 % (C04110/1)	158 % (C04110/2)	0 % (C04110/3)	0 % (C04110/4)
Half width	28 % (C04111/1)	20 % (C04111/2)	0 % (C04111/3)	0 % (C04111/4)

5.179 L_TbMul - multiplication

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a multiplication without limitation.



Inputs

Identifier/data type	Information/possible settings
dnIn1 DINT	Multiplier (1st factor)
dnIn2 DINT	Multiplicand (2nd factor)

Outputs

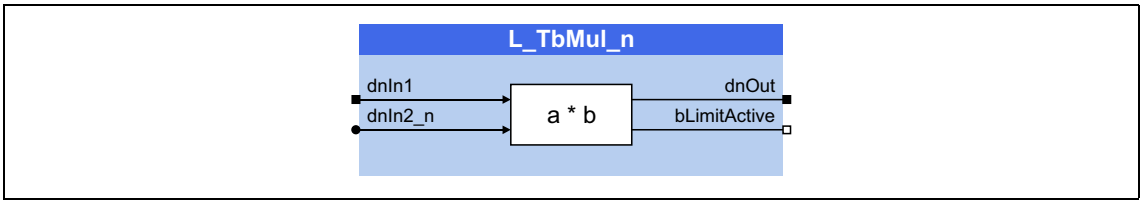
Identifier/data type	Value/meaning
dnOut DINT	Product value (result of the multiplication) with overflow • No internal limitation, thus an overflow is possible.

5.180 L_TbMul_n - scaled multiplication with limitation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a multiplication. The multiplicand (2. factor) must be specified as a scaled signal [%].

- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnIn1 DINT	Multiplier (1st factor)
dnIn2_n DINT	Multiplicand (2nd factor) • Scaling: 100 % ≡ 2^{30} ≡ 1073741824

Outputs

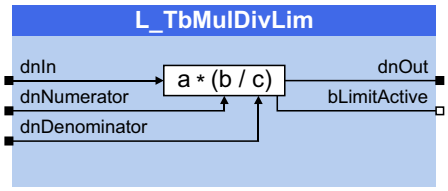
Identifier/data type	Value/meaning	
dnOut DINT	Product value (result of the multiplication) • Internally limited to $\pm 2^{31}$-1	
bLimitActive BOOL	Status signal "Limitation active"	
	TRUE	The output signal is limited.

5.181 L_TbMulDivLim - multiplication and division with limitation

Function library:	LenzeToolbox	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This FB carries out a multiplication with subsequent division and limitation.

- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Multiplier (1st factor)
dnNumerator DINT	Multiplicand (2nd factor)
dnDenominator DINT	Divisor

Outputs

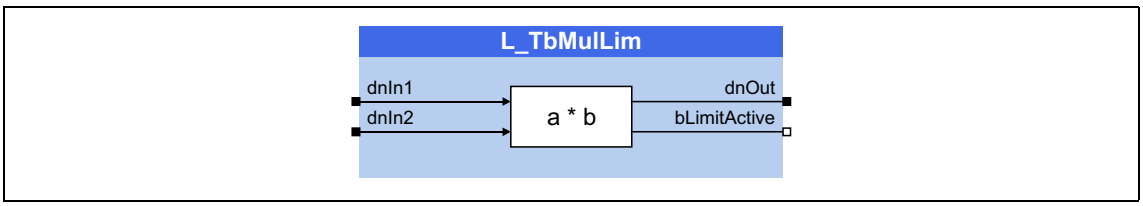
Identifier/data type	Value/meaning	
dnOut DINT	Result of the multiplication and division • Internally limited to $\pm 2^{31}$-1	
bLimitActive BOOL	Status signal "Limitation active"	
	TRUE	The output signal is limited.

5.182 L_TbMulLim - multiplication with limitation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a multiplication with limitation.

- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnIn1 DINT	Multiplier (1st factor)
dnIn2 DINT	Multiplicand (2nd factor)

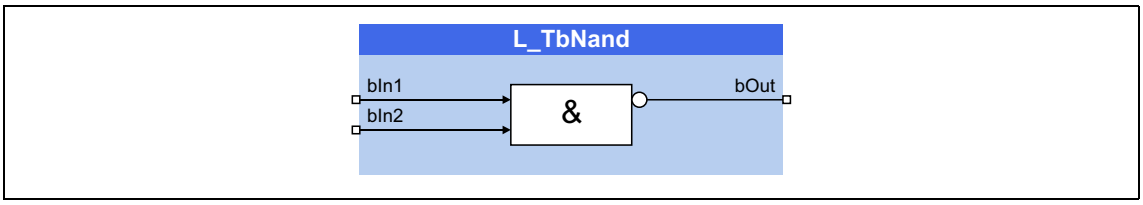
Outputs

Identifier/data type	Value/meaning	
dnOut DINT	Product value (result of the multiplication) • Internally limited to $\pm 2^{31}$-1	
bLimitActive BOOL	Status signal "Limitation active"	
	TRUE	The output signal is limited.

5.183 L_TbNand - NAND with 2 inputs

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a logic NAND operation of binary signals.



Inputs

Identifier/data type	Information/possible settings
bIn1 BOOL	Input 1
bIn2 BOOL	Input 2

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Result of the NAND operation

Function

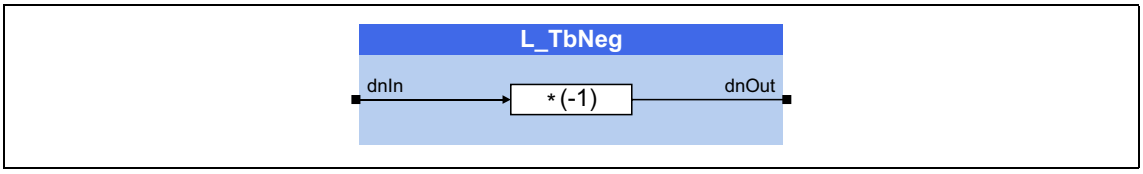
bIn2	bIn1	bOut
FALSE	FALSE	TRUE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	FALSE

5.184 L_TbNeg - negation (for "DINT" data type)

Function library:	LenzeToolbox	
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This function block inverts the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output.

- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.
- The L_TbNegSel FB is available for optional inversion.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal

Outputs

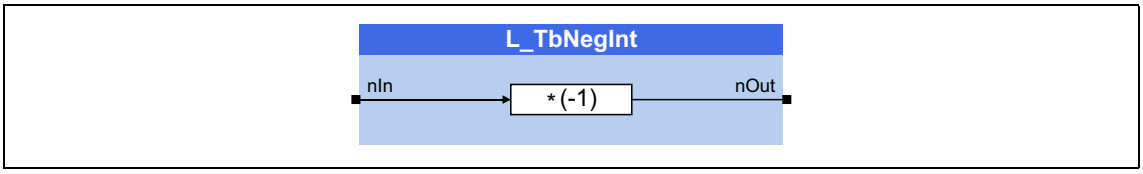
Identifier/data type	Value/meaning
dnOut DINT	Output signal

5.185 L_TbNegInt - negation (for "INT" data type)

Function library:	LenzeToolbox	**FB is available as of library V02.05.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This function block inverts the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output.

- The value output to output *nOut* is internally limited to $\pm 2^{15}$-1.
- The FB L_TbNegSelInt is available for optional inversion.



Inputs

Identifier/data type	Information/possible settings
nIn INT	Input signal

Outputs

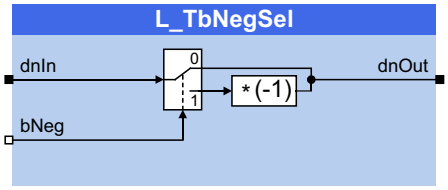
Identifier/data type	Value/meaning
nOut INT	Output signal

5.186 L_TbNegSel - optional negation (for "DINT" data type)

Function library:	LenzeToolbox	**FB is available as of library V02.02!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This function block inverts (optionally) the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output.

- In contrast to the L_TbNeg FB, this FB is only inverted if the *bNeg* control input is set to TRUE.
- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.



Inputs

Identifier/data type	Information/possible settings	
dnIn DINT	Input signal	
bNeg BOOL	Activate inversion	
	FALSE	*dnOut = dnIn*
	TRUE	*dnOut = -dnIn*

Outputs

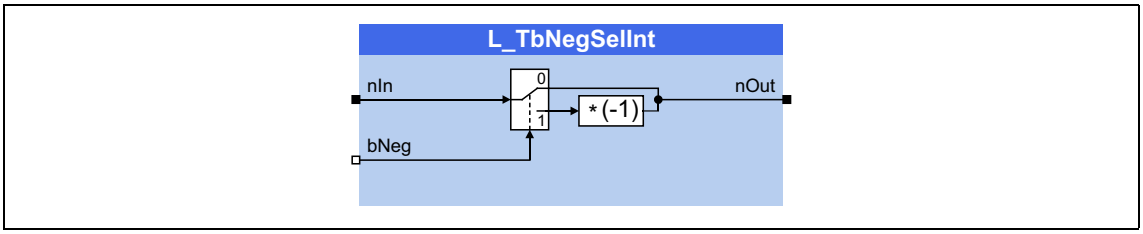
Identifier/data type	Value/meaning
dnOut DINT	Output signal

5.187 L_TbNegSelInt - optional negation (for "INT" data type)

Function library:	LenzeToolbox	**FB is available as of library V02.05.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This function block inverts (optionally) the sign of the input signal, i.e. the input signal is multiplied by -1 and is then output.

- In contrast to the L_TbNegInt FB, this FB is only inverted if the *bNeg* control input is set to TRUE.
- The value output to output *nOut* is internally limited to $\pm 2^{15}$-1.



Inputs

Identifier/data type	Information/possible settings	
nIn INT	Input signal	
bNeg BOOL	Activate inversion	
	FALSE	*nOut = nIn*
	TRUE	*nOut = -nIn*

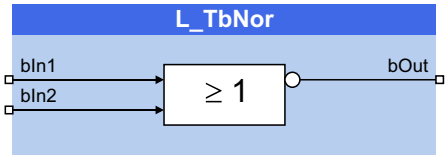
Outputs

Identifier/data type	Value/meaning
nOut INT	Output signal

5.188 L_TbNor - NOR with 2 inputs

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a logic NOR operation of binary signals.



Inputs

Identifier/data type	Information/possible settings
bIn1 BOOL	Input 1
bIn2 BOOL	Input 2

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Result of the NOR operation

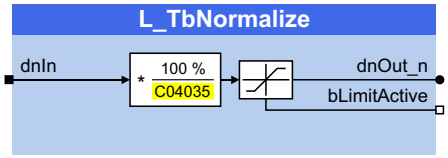
Function

bIn2	bIn1	bOut
FALSE	FALSE	TRUE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	FALSE

5.189 L_TbNormalize - signal scaling with limitation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB scales any signal to a parameterisable reference variable.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal

Outputs

Identifier/data type	Value/meaning	
dnOut_n DINT	Scaled output signal • Internally limited to ±200 %	
bLimitActive BOOL	Output signal within the limitation	
	TRUE	The output signal *dnOut_n* is limited.

Parameter

Parameter	Possible settings			Information
C04035	-2147483647		2147483647	Reference variable • Setting "0" is not possible. • Initialisation: 1073741824

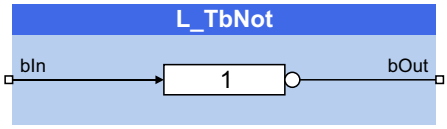
Function

$$dnOut\_n = \frac{dnIn}{C04035} \cdot 100\ \%$$

5.190 L_TbNot - negation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB negates a signal from the BOOL data type.



Inputs

Identifier/data type	Information/possible settings
bin BOOL	Input

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Result of the NOT operation

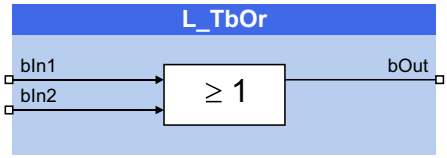
Function

bin	bOut
FALSE	TRUE
TRUE	FALSE

5.191 L_TbOr - OR with 2 inputs

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a logic OR operation of binary signals.



Inputs

Identifier/data type	Information/possible settings
bIn1 BOOL	Input 1
bIn2 BOOL	Input 2

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Result of the OR operation

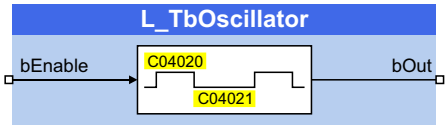
Function

bIn2	bIn1	bOut
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

5.192 L_TbOscillator - rectangular signal generator

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB can be used to generate a Boolean square wave signal with a variable high/low time.



Inputs

Identifier/data type	Information/possible settings	
bEnable BOOL	Activating the signal generator	
	FALSE	Signal generator deactivated, *bOut* = FALSE.
	TRUE	Signal generator activated.

Outputs

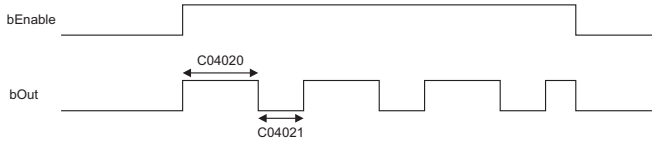
Identifier/data type	Value/meaning
bOut BOOL	Boolean square wave signal

Parameter

Parameter	Possible settings			Information
C04020	0.000	s	60.000	ON time • Initialisation: 0.001 s
C04021	0.000	s	60.000	OFF time • Initialisation: 0.001 s

5.192.1 Function

If the FB is activated by setting *bEnable* to TRUE, the set square wave signal is output to *bOut*, starting with the HIGH state:



- If *bEnable* is reset to FALSE, the output *bOut* is also immediately reset to FALSE.

5.192.2 Optimising the accuracy of the output signal

Enter only integer multiples of the task cycle time via C04020 and C04021 for a maximum accuracy of the rectangular signal to be output.

- In case of an asymmetrical entry of the pulse/pause time and non-compliance with the times selected as integer multiples of the task cycle time, the times selected are internally rounded up to integer multiples of the task cycle times.
- With a minimum timing below the task cycle time the permanent level arises according to the following rule: bOut = TRUE if C04020 > C04021.
- The following table lists the possible input signal states and the resulting output signal:

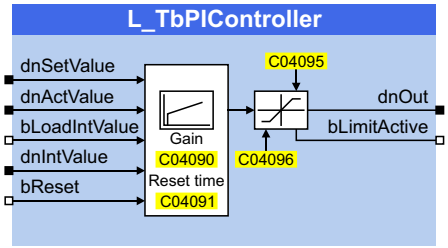
Task running time t	**On time** C04020	**Off time** C04021	**State of ON and OFF time**	**Output signal** bOut
1 ms	3 msec	2 ms	ON time, OFF time ≥ T ON time > OFF time	TRUE: 3 ms, FALSE: 2 ms
3 msec	3 msec	2 ms	ON time ≥ T OFF time < T ON time > OFF time	TRUE
3 msec	2 ms	1 ms	On time, OFF time < T ON time > OFF time	TRUE
3 msec	1 ms	2 ms	On time, OFF time < T ON time < OFF time	FALSE
3 msec	2 ms	2 ms	On time, OFF time < T ON time = OFF time	FALSE
3 msec	2 ms	3 msec	ON time < T OFF time ≥ T ON time < OFF time	FALSE
3 msec	3 msec	5 ms	ON time, OFF time ≥ T ON time < OFF time OFF time ≠ (n * T)	TRUE: 3 ms, FALSE: 6 ms
3 msec	5 ms	3 msec	ON time, OFF time ≥ T ON time > OFF time ON time ≠ (n * T)	TRUE: 6 ms, FALSE: 3 ms
3 msec	5 ms	4 ms	ON time, OFF time ≥ T ON time > OFF time ON time, OFF time ≠ (n * T)	TRUE: 6 ms, FALSE: 3 ms
3 msec	4 ms	5 ms	ON time, OFF time ≥ T ON time < OFF time ON time, OFF time ≠ (n * T)	TRUE: 3 ms, FALSE: 6 ms
3 msec	5 ms	5 ms	ON time, OFF time ≥ T ON time = OFF time ON time, OFF time ≠ (n * T)	TRUE: 6 ms, FALSE: 6 ms

5.193 L_TbPIController - PI controller with limitation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB provides a simple PI controller with different control functions.

- The value output to the output *dnOut* is internally limited to the limit range parameterised in C04095 and C04096 (default setting ±200 %).
- The *bLimitActive* output displays if a limitation is active.
- The function corresponds by approximation to the "PCTRL" function of the 9300 servo inverter.



Inputs

Identifier/data type	Information/possible settings	
dnSetValue DINT	Setpoint signal	
dnActValue DINT	Actual signal	
bLoadIntValue BOOL	Loading of the integrator	
	TRUE	Set the integral action component of the PI controller to the value applied to the input *dnIntValue*.
dnIntValue DINT	Value on which the integral action component of the PI controller is adjusted by setting *bLoadIntValue* to TRUE.	
bReset BOOL	Reset PI controller to zero.	
	TRUE	PI controller is reset to zero.

Outputs

Identifier/data type	Value/meaning	
dnOut DINT	Output signal • Is internally limited to the limit range parameterised in C04095 and C04096 (default setting ±200 %).	
bLimitActive BOOL	Status signal "Limitation active"	
	TRUE	The output signal is limited.

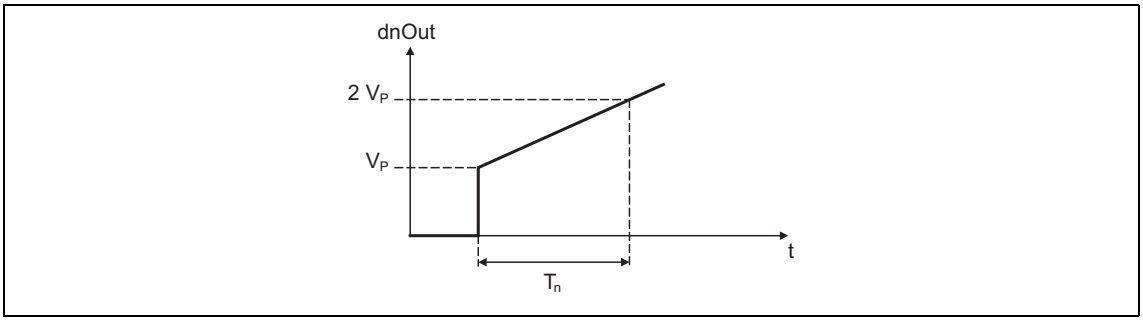
Parameter

Parameter	Possible settings			Information
C04090	0.0000		214748.3647	Controller gain • Setting 0.0000 resets the PI controller. • Initialisation: 1.0000
C04091	0.001	s	1000.000	Controller reset time • Setting 1000.000 resets the PI controller. • Initialisation: 1.000 s
C04095	0.00	%	200.00	Positive output limit value • Initialisation: 200.00 %
C04096	-200.00	%	0.00	Negative output limit value • Initialisation: -200.00 %

Function

The FB contains the calculation of the system deviation and the PI controller.

- The dynamics of the controller is parameterised according to the gain V_p (C04090) and the reset time T_n (C04091):



[5-147] Step response of the controller

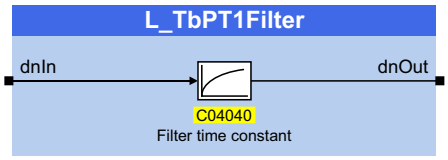
- By setting *bReset* to TRUE the controller is reset (P and I component).
- The I component of the controller can be set via the inputs *bLoadIntValue* and *dnIntValue*.
  - This function is also active when the controller is reset via the input *bReset*, i.e. the integral action component is loaded while a zero is output to the output. This provides the opportunity to reset the controller with a signal and simultaneously initialise the integral action component for the following enable:
- The correcting variable of the controller is automatically limited to the range $\pm 2^{31}$-1 (±200 %).
  - If a small setting range is required, the positive and negative output limit value can be set individually via C04095 and C04096.
  - When the controller reaches the output limitation, the P component has priority over the I component.

5.194 L_TbPT1Filter - delay

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB filters and delays analog signals.

- The filter time constant T is set via the parameter C04040.
- The gain is fixed at V = 1.
- The function corresponds to the "PT1" function of the 9300 servo inverter.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal

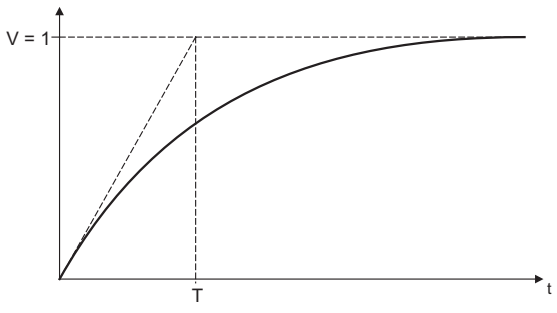
Outputs

Identifier/data type	Value/meaning
dnOut DINT	Output signal

Parameter

Parameter	Possible settings			Information
C04040	0.001	s	60.000	Filter time constant • With the setting 0.001 s the filter is not active. • Initialisation: 0.001 s

Function



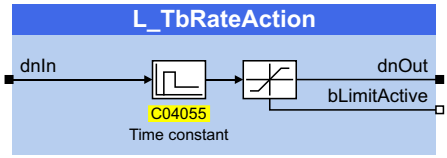
[5-148] Filter time constant T of the first-order delay element

5.195 L_TbRateAction - rate action with limitation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB contains a rate action function to compensate disturbing low passes.

- Therefore the rate time constant must be adjusted to the filter time constant of the low pass via the parameter C04055.
- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.



Inputs

Identifier/data type	Information/possible settings
dnIn DINT	Input signal

Outputs

Identifier/data type	Value/meaning	
dnOut DINT	Output signal • Internally limited to $\pm 2^{31}$-1	
bLimitActive BOOL	Status signal "Limitation active"	
	TRUE	The output signal is limited.

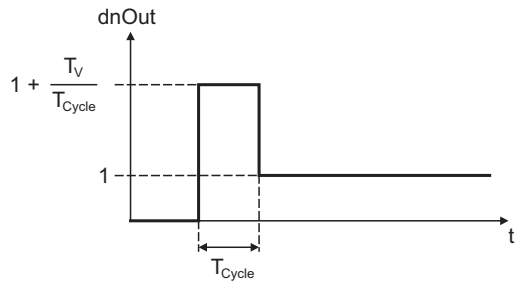
Parameter

Parameter	Possible settings			Information
C04055	0.000	s	60.000	Rate time constant T_V • Initialisation: 1.000 s

Function

The input signal is differentiated by means of the parameterised rate time constant T_V and added to the input signal.

- The differentiation of the input signal can be switched off by setting the rate time constant T_V lower than the task cycle time. In this case the differential component is zero and the input signal is output unchanged to the output.
- When the rate time constant T_V is higher than the task cycle time, the following step response is the result:



C04055: Rate time constant T_V

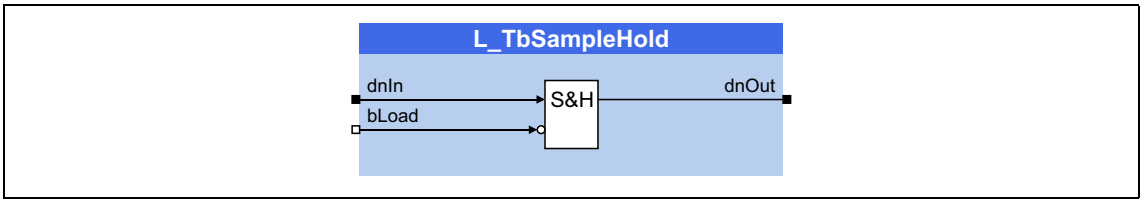
[5-149] Step response when the rate time constant > task cycle time

5.196 L_TbSampleHold - sample & hold (for "DINT" data type)

Function library:	LenzeToolbox	
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel
Required retain memory:	32 bytes	

This FB can save a signal from the "DINT" data type.

- The stored value is also available after mains disconnection.



Inputs

Identifier/data type	Information/possible settings	
dnIn DINT	Input signal	
bLoad BOOL	Saving of input signal	
	FALSE	The value valid last at *dnIn* is stored and output to *dnOut*. A signal change at *dnIn* does not change *dnOut*.
	TRUE	*dnIn* is output at the output *dnOut*.

Outputs

Identifier/data type	Value/meaning
dnOut DINT	Output signal

Behaviour when switching the mains

The value loaded last is stored permanently when the supply voltage is switched off and reloaded after the mains is switched on again.

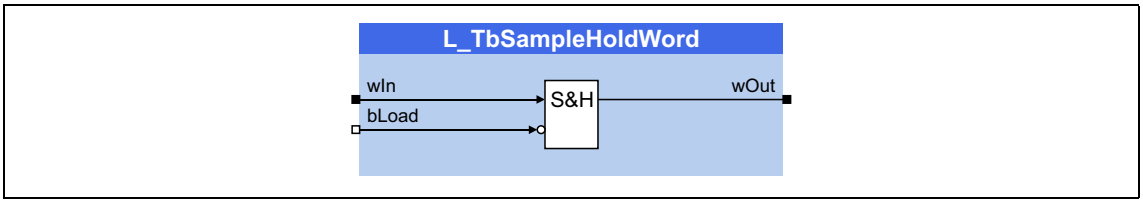
- In order that the stored value is not immediately overwritten by the current input signal at *dnIn* after switching on the mains again, *bLoad* must be set to FALSE when the mains is switched on again.

5.197 L_TbSampleHoldWord - sample & hold (for "WORD"data type)

Function library:	LenzeToolbox	**FB is available as of library V02.05.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel
Required retain memory:	28 bytes	

This FB can save a signal from the "WORD" data type.

- The stored value is also available after mains disconnection.



Inputs

Identifier/data type	Information/possible settings	
wIn WORD	Input signal	
bLoad BOOL	Saving of input signal	
	FALSE	The value valid last at *wIn* is saved and output to *wOut*. A signal change at *wIn* does not change *wOut*.
	TRUE	*wIn* is output at the output *wOut*.

Outputs

Identifier/data type	Value/meaning
wOut WORD	Output signal

Behaviour when switching the mains

The value loaded last is stored permanently when the supply voltage is switched off and reloaded after the mains is switched on again.

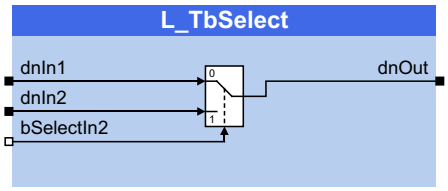
- In order that the stored value is not immediately overwritten by the current input signal at *wIn* after switching on the mains again, *bLoad* must be set to FALSE when the mains is switched on again.

5.198 L_TbSelect - selector (for "DINT" data type)

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This function block switches between two signals of "DINT" type. The switch-over is controlled via a boolean input signal.

- So it is, for example, possible to switch between an initial diameter and a calculated diameter during winding.



Inputs

Identifier/data type	Information/possible settings	
dnIn1 DINT	Input signal 1	
dnIn2 DINT	Input signal 2	
bSelectIn2 BOOL	Selection of the input signal for the output to *dnOut*	
	FALSE	*dnIn1*
	TRUE	*dnIn2*

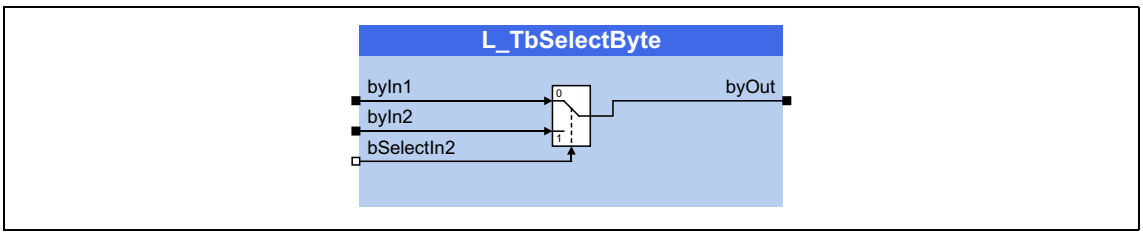
Outputs

Identifier/data type	Value/meaning
dnOut DINT	Output signal

5.199 L_TbSelectByte - selector (for "BYTE" data type)

Function library:	LenzeToolbox	**FB is available as of library V02.05.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This function block switches between two signals of "BYTE" type. The switch-over is controlled via a boolean input signal.



Inputs

Identifier/data type	Information/possible settings	
byIn1 BYTE	Input signal 1	
byIn2 BYTE	Input signal 2	
bSelectIn2 BOOL	Selection of the input signal for the output to *byOut*	
	FALSE	*byIn1*
	TRUE	*byIn2*

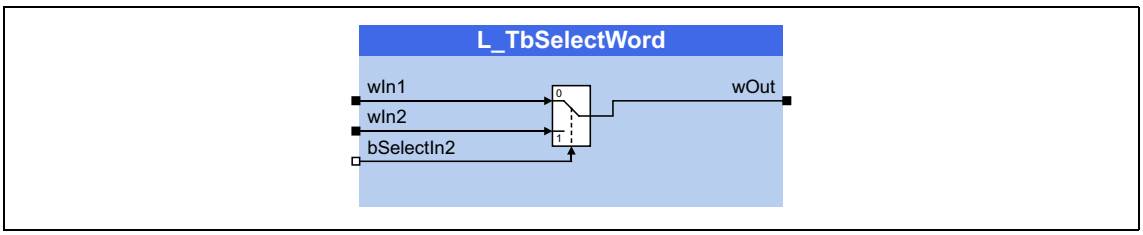
Outputs

Identifier/data type	Value/meaning
byOut BYTE	Output signal

5.200 L_TbSelectWord - selector (for "WORD" data type)

Function library:	LenzeToolbox	**FB is available as of library V02.05.xx.xx!**
Runtime software licence:	☑ Motion Control HighLevel	☑ Motion Control TopLevel

This function block switches between two signals of "WORD" type. The switch-over is controlled via a boolean input signal.



Inputs

Identifier/data type	Information/possible settings	
wIn1 WORD	Input signal 1	
wIn2 WORD	Input signal 2	
bSelectIn2 BOOL	Selection of the input signal for the output to *byOut*	
	FALSE	*wIn1*
	TRUE	*wIn2*

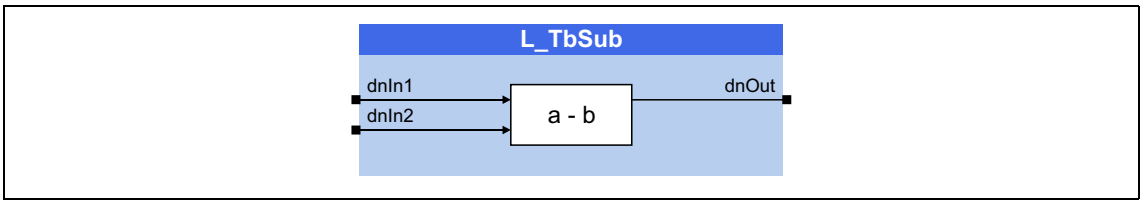
Outputs

Identifier/data type	Value/meaning
wOut WORD	Output signal

5.201 L_TbSub - subtraction

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a subtraction without limitation.



Inputs

Identifier/data type	Information/possible settings
dnIn1 DINT	Minuend
dnIn2 DINT	Subtrahend

Outputs

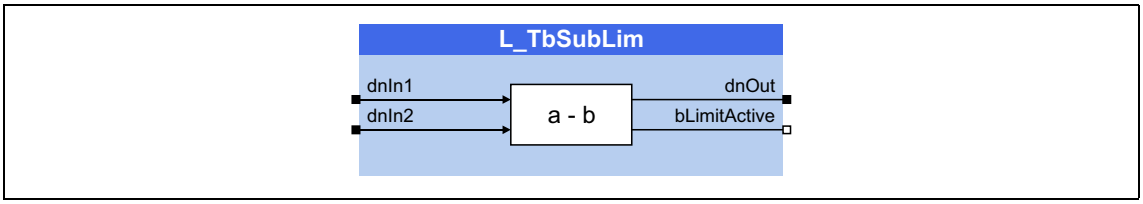
Identifier/data type	Value/meaning
dnOut DINT	Differential value (result of the subtraction) • No internal limitation, thus an overflow is possible: • If the result > 2^{31}-1: *dnOut* = *dnIn1* - *dnIn2* - 2^{32} • If the result < 2^{31}: *dnOut* = *dnIn1* - *dnIn2* + 2^{32}

5.202 L_TbSubLim - subtraction with limitation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a subtraction with limitation.

- The value output to output *dnOut* is internally limited to $\pm 2^{31}$-1.
- The *bLimitActive* output displays if a limitation is active.



Inputs

Identifier/data type	Information/possible settings
dnIn1 DINT	Minuend
dnIn2 DINT	Subtrahend

Outputs

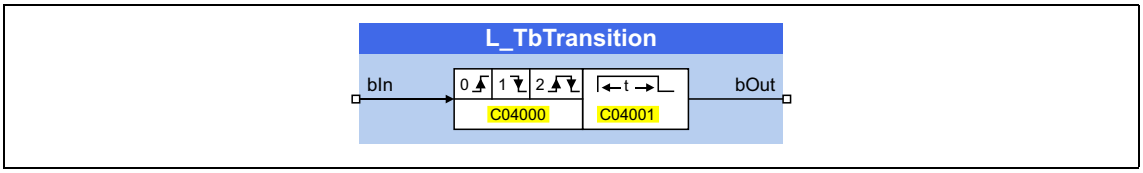
Identifier/data type	Value/meaning	
dnOut DINT	Differential value (result of the subtraction) • Internally limited to $\pm 2^{31}$-1	
bLimitActive BOOL	Status signal "Limitation active"	
	TRUE	The output signal is limited.

5.203 L_TbTransition - edge evaluation

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to evaluate digital signal edges and converts them into time-defined and retriggerable pulses.

- This function corresponds to the "TRANS" function of the 9300 servo inverter.



Inputs

Identifier/data type	Information/possible settings
bin BOOL	Input for edge evaluation • Function depends on the selection of the trigger edge in C04000.

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Output (retriggerable)

Parameter

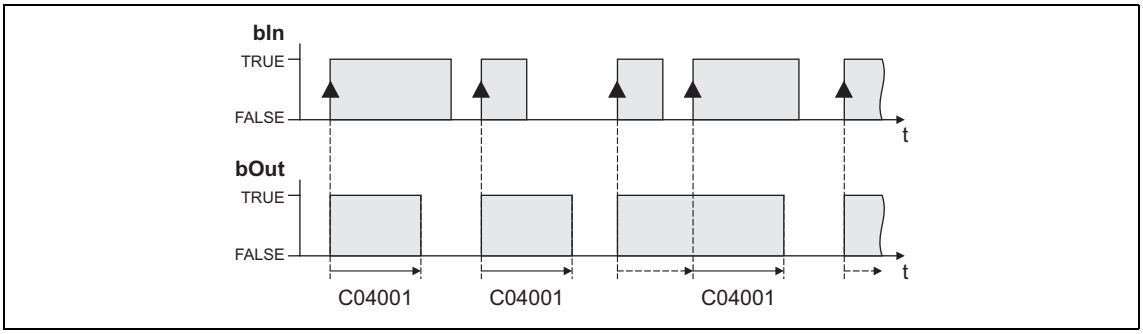
Parameter	Possible settings			Information
C04000				Trigger edge
	0	Rising edge		Lenze setting
	1	Falling edge		
	2	Both edges		
C04001	0.001	s	60.000	Pulse duration • Initialisation: 0.001 s

Note!

When the task interval time is set to >1 ms and the pulse duration set in C04001 is not an integer multiple of the interval time, the pulse duration is automatically extended to the next integer multiple of the interval time.

5.203.1 Function 1: Evaluating rising edges

C04000 = "0"



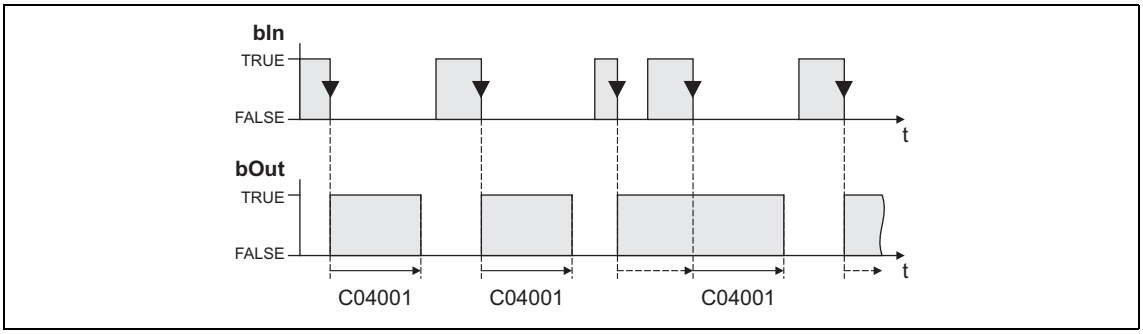
[5-150] Switching performance when evaluating rising edges

Functional sequence

1. A FALSE-TRUE edge at the input *bIn* sets the output *bOut* to TRUE.
2. After the time specified via C04001 the output *bOut* is reset to FALSE provided that no other FALSE/TRUE edge has taken place at the output *bIn* in the meantime.
   - If another FALSE/TRUE edge takes place at the *bIn* input, the time C04001 restarts to elapse, i.e. the output *bOut* is retriggerable.

5.203.2 Function 2: Evaluating falling edges

C04000 = "1"



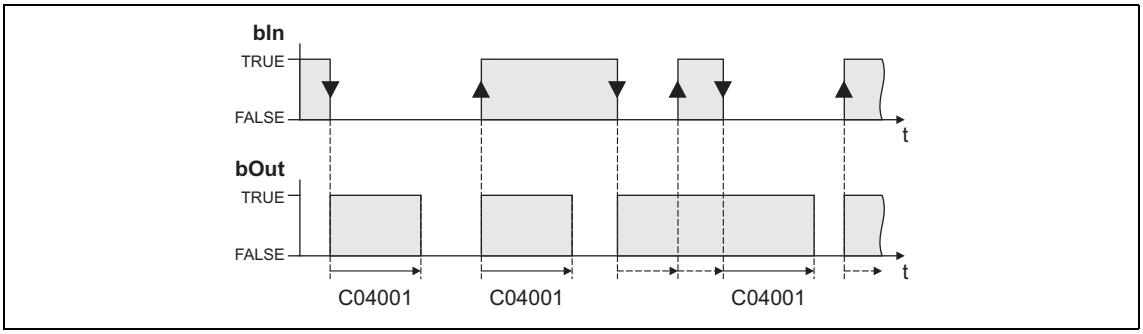
[5-151] Switching performance when evaluating falling edges

Functional sequence

1. A TRUE-FALSE edge at the input *bIn* sets the output *bOut* to TRUE.
2. After the time specified via C04001 the output *bOut* is reset to FALSE provided that no other TRUE/FALSE edge has taken place at the *bIn* input in the meantime.
   - If another TRUE/FALSE edge takes place at the *bIn* input, the time C04001 restarts to elapse, i.e. the output *bOut* is retriggerable.

5.203.3 Function 3: Evaluating rising and falling edges

C04000 = "2"



[5-152] Switching performance when evaluating rising and falling edges

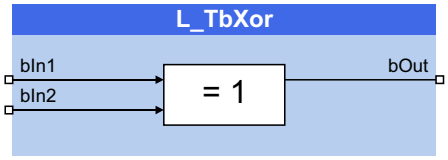
Functional sequence

1. A signal change (FALSE-TRUE or TRUE-FALSE edge) at the input *bIn* sets the output *bOut* to TRUE.
2. After the time specified via C04001 the output *bOut* is reset to FALSE provided that no other signal change has taken place at the *bIn* input in the meantime.
   - If another signal change takes place at the *bIn* input, the time C04001 restarts to elapse, i.e. the output *bOut* is retriggerable.

5.204 L_TbXor - XOR with 2 inputs

Function library:	LenzeToolbox
Runtime software licence:	☑ Motion Control HighLevel ☑ Motion Control TopLevel

This FB carries out a logic EXCLUSIVE OR operation of binary signals.



Inputs

Identifier/data type	Information/possible settings
bIn1 BOOL	Input 1
bIn2 BOOL	Input 2

Outputs

Identifier/data type	Value/meaning
bOut BOOL	Result of the exclusive OR operation

Function

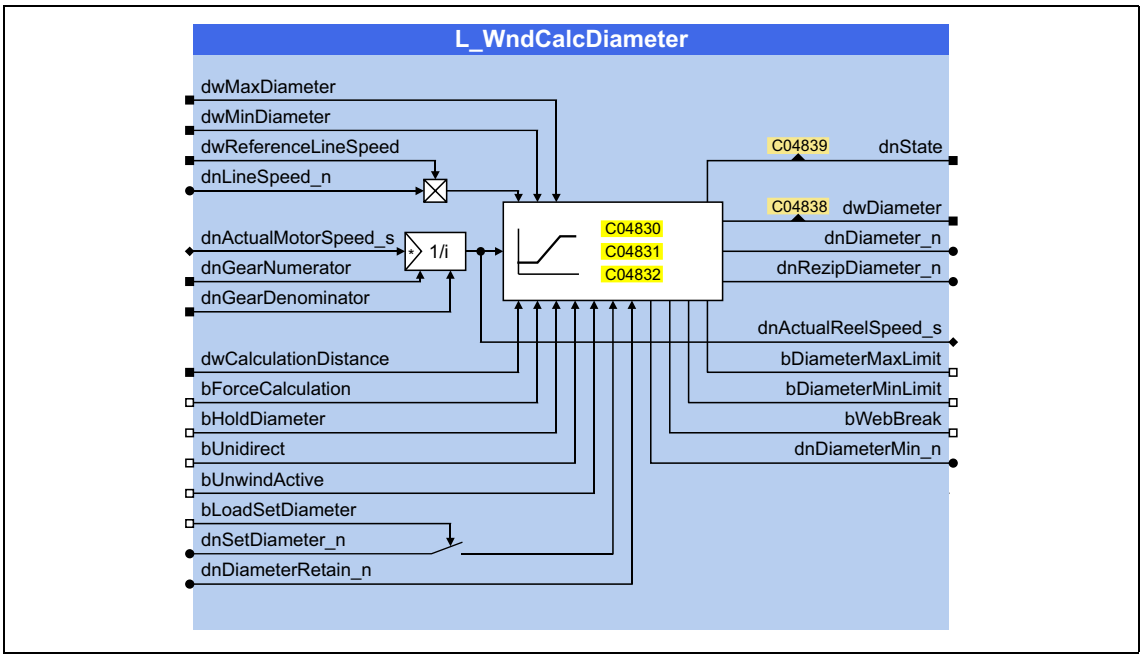
bIn2	bIn1	bOut
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	FALSE

5.205 L_WndCalcDiameter - diameter calculation

Function library:	LenzeWinder
Runtime software licence:	☐ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to calculate the reel diameter from the line speed and the reel speed.

- The output signal *dwDiameter* has a resolution of 1 µm and can display a maximum of 10 m.



Inputs

Identifier/data type	Information/possible settings
dwMaxDiameter DWORD	Maximum diameter in [µm] • 1000 µm ≡ 1 mm; 1000000 µm ≡ 1 m • Internal limitation to 10 m
dwMinDiameter DWORD	Minimum diameter in [µm] • 1000 µm ≡ 1 mm; 1000000 µm ≡ 1 m • Internal limitation to 10 m
dwReferenceLineSpeed DWORD	Maximum material speed in [cm/min] • 2500 cm/min ≡ 25.00 m/min • Internal limitation to 2000 m/min
dnLineSpeed_n DINT	Material speed setpoint in [%] • 100 % ≡ maximum material speed (*dwReferenceLineSpeed*)
dnActualMotorSpeed_s DINT	Current motor speed in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864
dnGearNumerator DINT	Gearbox factor (numerator)
dnGearDenominator DINT	Gearbox factor (denominator)
dwCalculationDistance DWORD	Reel shaft distance (number of revolutions) after which a recalculation of the diameter is carried out. • 1000 ≡ one revolution (with three decimal positions) • Internal limitation to 1000000, i. e. the diameter calculation can be maximally averaged by 1000 revolutions.

Identifier/data type	Information/possible settings	
bForceCalculation BOOL	Forcing of the diameter calculation	
	FALSE↗TRUE	Forcing of the diameter calculation • The next diameter calculation is carried out after the reel shaft distance specified via *dwCalculationDistance* has been elapsed.
bHoldDiameter BOOL	Hold last diameter value	
	TRUE	The diameter value output last is not overwritten with new values.
bUnidirect BOOL	Enable of only one direction of change and activation of the web break monitoring	
	TRUE	Only the direction of change defined via *bUnwindActive* is enabled.
bUnwindActive BOOL	Selection of the direction of change enabled • Only in the case of *bUnidirect* = TRUE	
	FALSE	Clockwise rotation (Cw)
	TRUE	Counter-clockwise rotation (Ccw)
bLoadSetDiameter BOOL	Acceptance of the preselected initial value • This input has a higher priority than the input *bHoldDiameter*.	
	TRUE	Accept value at *dnSetDiameter_n*.
dnSetDiameter_n DINT	Selection of an initial value/external diameter signal in [%] • 100 % ≡ maximum diameter (*dwMaxDiameter*) • Value is accepted by setting *bLoadSetDiameter* to TRUE.	
dnDiameterRetain_n DINT	Input for reading in the current diameter value after mains switching: 1. For this, connect the output *dnDiameter_n* with a retain variable. 2. Then connect the retain variable with this input.	

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded) • Bits that are not listed have not been assigned with a status (always "0").	
	Status messages:	
	Bit 12	Diameter value was calculated/read from the memory.
	Bit 13	Diameter value was initialised with *dwMinDiameter*.
	Bit 14	Diameter value is/was loaded via *bLoadSetDiameter*/ *dnSetDiameter_n*.
	Bit 15	Diameter value is/was held (*bHoldDiameter*).
	Error messages:	
	Bit 29	Input value *dwMinDiameter* ≥ *dwMaxDiameter*.
	Bit 30	Internal reset of the diameter calculation (integration overflow by *dnLineSpeed_n*).
dwDiameter DWORD	Current diameter in [µm] • 1000 µm ≡ 1 mm; 1000000 µm ≡ 1 m • Internal limitation to 10 m • Display parameter: C04838	
dnDiameter_n DINT	Current diameter in [%] • 100 % ≡ maximum diameter (*dwMaxDiameter*)	
dnRecipDiameter_n DINT	Reciprocal value of the current diameter in [%] • 100 % ≡ minimum diameter (*dwMinDiameter*)	
dnActualReelSpeed_s DINT	Current reel speed in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864 • The current reel speed results from the motor speed (*dnActualMotorSpeed_s*) and the gearbox factor (*dnGearNumerator*/*dnGearDenominator*).	

Identifier/data type	Value/meaning	
bDiameterMaxLimit BOOL	Limit value monitoring	
	TRUE	Upper limit value *dwMaxDiameter* reached.
bDiameterMinLimit BOOL	Limit value monitoring	
	TRUE	Lower limit value *dwMinDiameter* reached.
bWebBreak BOOL	Web break monitoring	
	TRUE	Web break after passing through the monitor window set in C04832.
dnDiameterMin_n DINT	Minimum diameter in [%] • 100 % ≡ maximum diameter (*dwMaxDiameter*)	

Parameter

Parameter	Possible settings			Info
C04830	0.000	Rev	1000.000	Calculation interval • Display of the *dwCalculationDistance* input signal.
C04831	0.01	s	3.00	Diameter filter time constant • Initialisation: 0.10 s
C04832	0.00	%	100.00	Window for web break monitoring • Only relevant for *bUnidirect* = TRUE. • Relating to the maximum diameter *dwMaxDiameter*. • Initialisation: 10.00 %
C04838	0	µm	10000000	Current diameter • Display of the output signal *dwDiameter*.
C04839	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.

5.205.1 Setting an initial value

At the input *dnSetDiameter_n* you can specify an initial value or an external diameter signal.

- This value is accepted if *bLoadSetDiameter* is set to TRUE.
- During the acceptance, the diameter calculation is reset and the filter for the diameter calculation is loaded with the initial value specified.

5.205.2 Calculating the diameter

For calculating the reel diameter, the material speed and the current reel speed are cyclically integrated.

- The material speed is specified via *dnLineSpeed_n*.
- The reel speed results from the motor speed (*dnActualMotorSpeed_s*) and the current gearbox factor (*dnGearNumerator*/*dnGearDenominator*).
- The calculation cycle is defined via the input *dwCalculationDistance*.
- At the end of each integration interval a new diameter value results from the division of the integrator values.
- An absolute value generation and smoothing via a first order low pass is effected. The time constant of this filter can be set in C04831. The output *dwDiameter* is not filtered.

Converting the diameter to 1/D

For evaluating the material speed with 1/D (speed-controlled winders), the reciprocal value of the reel diameter is provided via output *dnRezipDiameter_n*.

- This value refers to the limit value *dwMinDiameter*.

5.205.3 Holding the current value

By setting *bHoldDiameter* to TRUE, the last diameter value *dnDiameter_n* that is output is held internally.

- The diameter calculation is reset and the filter for the diameter calculation is loaded with the diameter value that is held internally.

5.205.4 Selecting the direction of change/web break monitoring

By setting *bUnidirect* to TRUE, only one direction of change is enabled for the diameter calculation, and at the same time the web break monitoring is activated.

- The permissible direction of change is defined via *bUnwindActive*.
- A newly calculated value is only accepted if it exceeds the value in the permissible direction which is saved last.

After a web break the successively calculated diameter values in the majority of cases go to the opposite direction very quickly.

- In C04832 you can define the maximally permissible deviation in the opposite direction by the exceedance of which the output *bWebBreak* is set to TRUE.
- If *bUnidirect* is set to FALSE, both directions of change are enabled, and the internal memory is always overwritten with the recalculated value. By this, the web break monitoring is switched off.

5.205.5 Limit value monitoring

Via *dwMaxDiameter* and *dwMinDiameter* you define limit values for the diameter values specified by the achievement/exceedance of which the corresponding output (*bDiameterMaxLimit*/ *bDiameterMinLimit*) is set to TRUE.

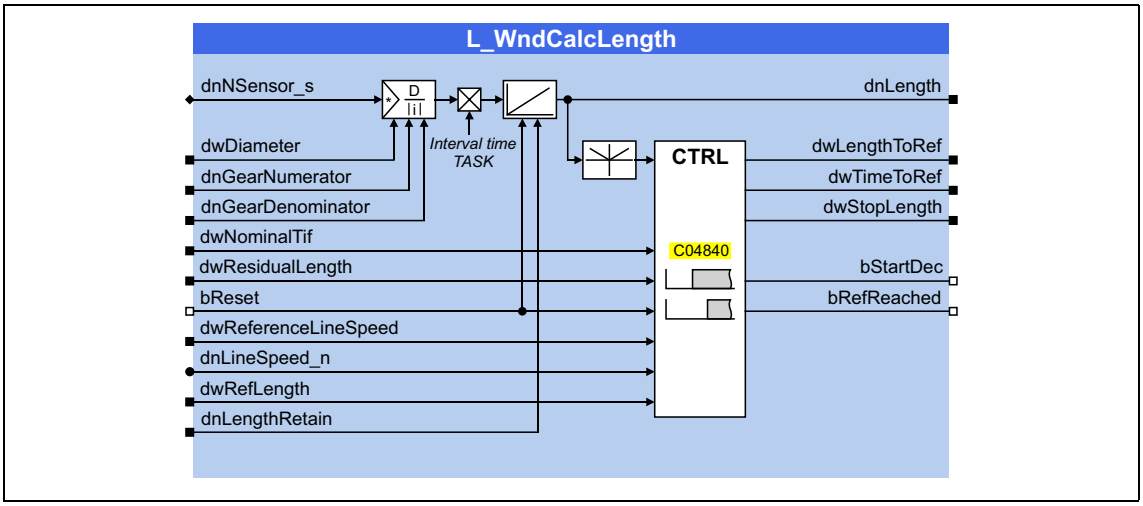
- The unfiltered diameter value *dwDiameter* and the filtered diameter value *dnDiameter_n* are limited.
- The limit values are entered in [µm].
- The hysteresis for resetting *bDiameterMaxLimit*/*bDiameterMinLimit* is set fixedly to 1 % of *dwMaxDiameter*.
- For the parameterisation of further FBs, the value in *dwMinDiameter* relating to *dwMaxDiameter* on a percentage basis is provided via output *dnDiameterMin_n*.

5.206 L_WndCalcLength - length calculation

Function library:	LenzeWinder
Runtime software licence:	☐ Motion Control HighLevel ☑ Motion Control TopLevel

On the basis of the speed and the diameter of a roll moving along with the material, this FB serves to carry out a continuous length calculation.

- If a gearbox is switched between the roll and the speed sensor, the sensor speed is converted to the speed of the roll according to the gearbox factor.
- The output signal *dnLength* has a resolution of 1 mm and can display a maximum of 2000 km.



Inputs

Identifier/data type	Information/possible settings	
dnNSensor_s DINT	Current encoder speed in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864	
dwDiameter DWORD	Diameter of the impeller/roll in [µm] • 1000 µm ≡ 1 mm; 1000000 µm ≡ 1 m • Internal limitation to 10 m	
dnGearNumerator DINT	Gearbox factor (numerator) of the gearbox between the impeller/roll and the speed sensor	
dnGearDenominator DINT	Gearbox factor (denominator) of the gearbox between the impeller/roll and the speed sensor	
dwNominalTif DWORD	Rated deceleration time for braking process in [ms] • Is triggered if the pre-switch-off length has been reached (output *bStartDec* = TRUE). • The reference value can be selected in C04840 (Lenze setting: material speed set-point). • 1000 ms ≡ 1 s • Internal limitation to 10000 s.	
dwResidualLength DWORD	Remaining residual length after the winding process in [mm] • 1000 mm ≡ 1 m; 1000000 mm ≡ 1 km • Internal limitation to 1000 km	
bReset BOOL	Reset length acquisition	
	TRUE	The length acquisition and the digital output signals *bStartDec* and *bRefReached* are reset.

Identifier/data type	Information/possible settings
dwReferenceLineSpeed DWORD	Maximum material speed in [cm/min] • 2500 cm/min ≡ 25.00 m/min • Internal limitation to 2000 m/min
dnLineSpeed_n DINT	Material speed setpoint in [%] • 100 % ≡ maximum material speed (*dwReferenceLineSpeed*)
dwRefLength DWORD	The length to be winded up during a winding process in [mm] • 1000 mm ≡ 1 m; 1000000 mm ≡ 1 km • Internal limitation to 2000 km
dnLengthRetain DINT	Input for reading in the current length after mains switching: 1. For this, connect the output *dnLength_n* with a retain variable. 2. Then connect the retain variable with this input.

Outputs

Identifier/data type	Value/meaning	
dnLength DINT	Current length in [mm]	
dwLengthToRef DWORD	Residual length until reaching the reference value in [mm] • Internal limitation to 2000 km	
dwTimeToRef DWORD	Residual runtime until reaching the reference value in [s] • The calculation depends on C04840, either on the basis of the material speed setpoint or the maximum material speed. • This value is only calculated at every 2. call of the FB (alternating with *dwStopLength*).	
dwStopLength DWORD	Change in length during the braking phase in [mm] • This value is only calculated at every 2. call of the FB (alternating with *dwTimeToRef*).	
bStartDec BOOL	Control signal for starting the braking process	
	TRUE	Pre-switch-off length reached.
bRefReached BOOL	Control signal for triggering the limit stop	
	TRUE	Reference length *dwRefLength* reached.

Parameters

Parameter	Possible settings		Info
C04840	Reference value =		Stop length of reference value • Reference value for the rated deceleration time *dwNominalTif* • Initialisation: maximum material speed
	0	Maximum material speed (*dwReferenceLineSpeed*)	
	1	Material speed setpoint (*dnLineSpeed_n*)	

5.206.1 Display of the length/residual length/residual runtime reached

Via the output *dnLength* the length reached is displayed as a physical value [mm]. Negative values can also be shown.

Via the output *dwLengthToRef* the residual length until reaching the reference value (*dwRefLength*) is provided.

- Assuming that the maximum material speed *dwReferenceLineSpeed* is run, a corresponding residual runtime *dwTimeToRef* results from the residual length.

5.206.2 Activating the braking process

For the activation of the braking process, the current length is compared to the remaining length after the stop (*dwResidualLength*), including the change in length during the braking phase (*dwStopLength*).

- The change in length during the braking phase is output at *dwStopLength*. It depends on the current material speed which is calculated from *dwReferenceLineSpeed* and *dnLineSpeed_n*, and on the reference deceleration time *dwNominalTif*.
- The signal for activating the braking process is output at *bStartDec*.

5.206.3 Triggering a limit stop

For triggering a limit stop, the current length is compared to the reference length.

- The signal for triggering the limit stop is output at *bRefReached*.

5.206.4 Resetting the calculation & trigger signals

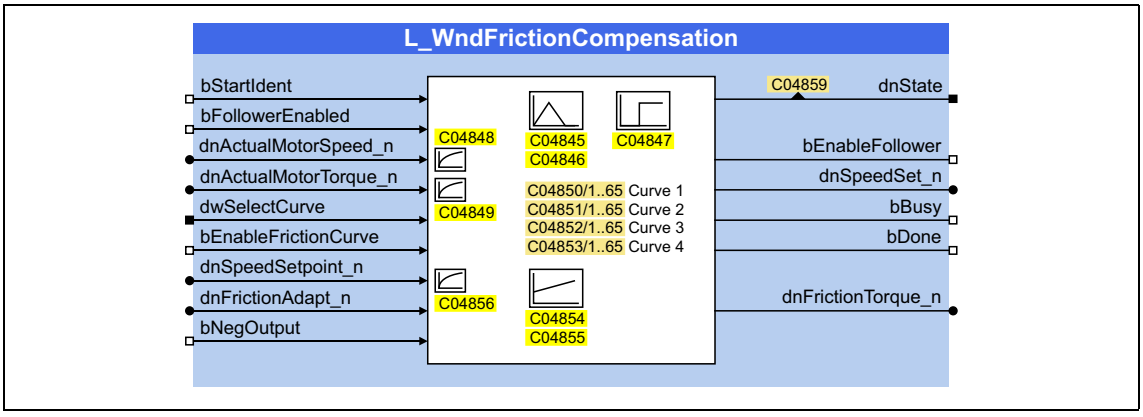
By setting *bReset* to TRUE, you can reset the length calculation and the two digital outputs for the activation of the braking process/limit stop.

5.207 L_WndFrictionCompensation - friction compensation

Function library:	LenzeWinder	
Runtime software licence:	☐ Motion Control HighLevel	☑ Motion Control TopLevel

This FB contains the two functions "friction identification" and "friction compensation".

- With the "friction identification" function the friction characteristic can be identified automatically.
  - For this, the FB accepts the drive control on request and carries out the identification.
  - For saving the automatically identified friction characteristic, four memory locations are provided. The selection of memory locations 1 ... 4 is effected via the input *dwSelectCurve*.
  - The friction characteristic includes 65 values which are equally distributed across the speed range of 0 ... n_{max} (C00011).
- With the "friction compensation" function, a speed-dependent compensation of the friction torque can be carried out on the basis of a linear or a previously automatically determined friction characteristic.
  - If the "friction compensation" function is activated, the input *dwSelectCurve* on the other hand serves to select the friction characteristic to be used.
  - If the input *dwSelectCurve* is = 0 or unassigned, a parameterisable linear characteristic is used for the friction compensation.



Inputs

Identifier/data type	Information/possible settings
bStartIdent BOOL	Start identification of the friction characteristic TRUE | Start identification.
bFollowerEnabled BOOL	Interface for the basic function "Speed follower" • Connect this input with the output *SF_bEnabled* of the SB **LS_SpeedFollower**.
dnActualMotorSpeed_n DINT	Current speed of the motor shaft in [%] • 100 % ≡ motor reference speed (C00011)
dnActualMotorTorque_n DINT	Current actual torque of the motor in [%] • 100 % ≡ motor reference torque (C00057/2)

Identifier/data type	Information/possible settings	
dwSelectCurve DWORD	**Friction identification** (*bStartIdent* = True): Selection of the memory area within which the friction values identified are saved. 4 characteristics can be saved in the FB at the maximum.	
	1 ... 4	Memory area 1 ... 4
	Friction compensation (*bStartIdent* = FALSE, *bEnableFrictionCurve* = TRUE): Selection of the characteristic for the friction compensation.	
	0	Linear friction characteristic (adjustable in C04854 and C04855)
	1 ... 4	Identified friction characteristic (memory area 1 ... 4)
bEnableFrictionCurve BOOL	Activate friction compensation	
	FALSE	Friction torque is reset to "0".
	TRUE	Friction compensation is activated.
dnSpeedSetpoint_n DINT	Speed reference value for characteristic in [%] • 100 % ≡ motor reference speed (C00011)	
dnFrictionAdapt_n DINT	Input for the proportional evaluation of the characteristic in [%] • Internal limitation to 0 ... 200 %	
bNegOutput BOOL	Inversion of the output signal	
	TRUE	Output signal *dnFrictionTorque_n* is displayed in an inverted manner.

Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded) • Bits that are not listed have not been assigned with a status (always "0").	
	Status messages:	
	Bit 0	Identification started. Waiting for activation of the basic function "Speed follower".
	Bit 1	Waiting for motor standstill.
	Bit 2	Drive is accelerated to the specified speed.
	Bit 3	Drive has reached the specified speed and is braked to standstill.
	Bit 4	The friction torque is calculated.
	Bit 5	Identification completed successfully.
	Error messages:	
	Bit 16	Identification cancelled (*bStartIdent* = FALSE).
	Bit 17	Identification cancelled (*bFollowerEnabled* = FALSE).
	Bit 18	Specified maximum speed not reached.
	Bit 19	Acceleration time for the friction absorption too short.
	Bit 20	Speed reached at the end of acceleration process too low.
bEnableFollower BOOL	Interface for the basic function "Speed follower" • Connect this output to the input *SF_bEnable* of SB **LS_SpeedFollower**.	
dnSpeedSet_n DINT	Interface for the basic function "Speed follower" • Connect this output to the input *SF_dnSpeedSet_n* of SB **LS_SpeedFollower**.	
bBusy BOOL	Status signal "Detection of measured value active"	
	TRUE	Detection of measured value active.
bDone BOOL	Status signal "Identification completed successfully"	
	TRUE	Identification completed successfully.
dnFrictionTorque_n DINT	Current friction torque in [%] • 100 % ≡ motor reference torque (C00057/2)	

Parameters for the identification of the friction characteristic

Parameter	Possible settings			Info
C04845	5.00	%	100.00	Max. motor speed - ident • 100 % ≡ motor reference speed (C00011) • Initialisation: 100.00 %
C04846	1	s	3600	Acceleration/deceleration time - ident • Initialisation: 30 s
C04847	0	ms	1000	Start delay - ident • Initialisation: 500 ms
C04848	0	ms	100	Current speed filter time - ident • Initialisation: 5 ms
C04849	0	ms	1000	Current torque filter time - ident • Initialisation: 5 ms
C04850/1...65	0.00	%	200.00	Friction characteristic 1 • Display of the identified friction values 1 ... 65
C04851/1...65	0.00	%	200.00	Friction characteristic 2 • Display of the identified friction values 1 ... 65
C04852/1...65	0.00	%	200.00	Friction characteristic 3 • Display of the identified friction values 1 ... 65
C04853/1...65	0.00	%	200.00	Friction characteristic 4 • Display of the identified friction values 1 ... 65

Parameters for friction compensation

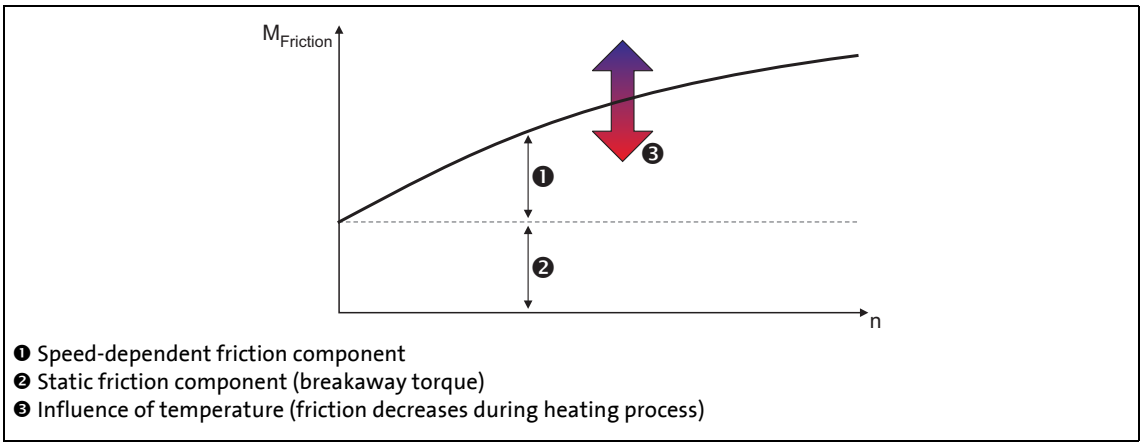
Parameter	Possible settings			Info
C04854	0.00	%	200.00	Starting friction - comp • Initialisation: 0.00 %
C04855	0.00	%	200.00	Final friction - comp • Initialisation: 0.00 %
C04856	0	ms	100	Filter time setpoint speed - comp • Filter time constant for speed reference value *dnSpeedSetpoint_n*. • Initialisation: 0 ms

Status display

Parameter	Possible settings			Info
C04859	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.

5.207.1 Basics: Friction response

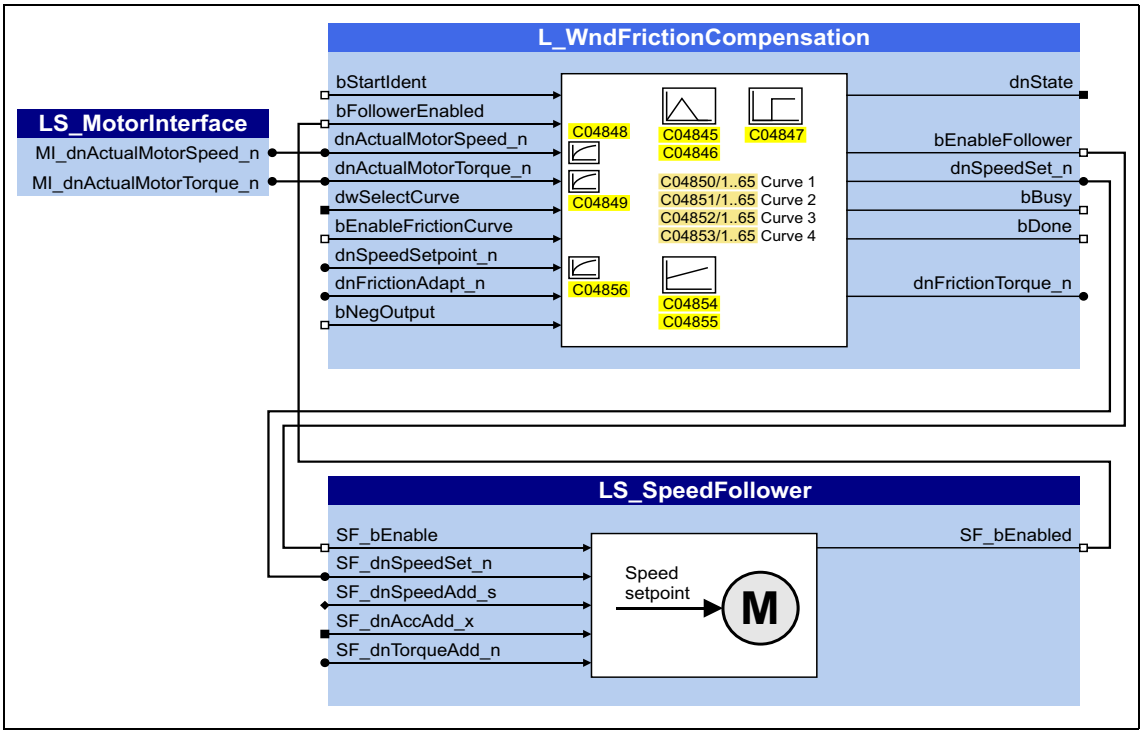
Starting with the breakaway torque, the friction generally increases with the speed. However, apart from the speed, also the temperature and torque (e.g. if a gearbox is used) can influence the friction response.



❶ Speed-dependent friction component
❷ Static friction component (breakaway torque)
❸ Influence of temperature (friction decreases during heating process)

[5-153] Friction response

5.207.2 Basic interconnection



[5-154] Basic interconnection with the basic function "Speed follower"

5.207.3 Identifying the friction characteristic automatically

Danger!

For the identification of the friction characteristic, the drive is accelerated to the maximum speed set in C04845 and braked to standstill again with the acceleration/deceleration time set in C04846.

Select the memory location for the characteristic

Define the memory location (1 ... 4) on which the characteristic is to be saved after a successful identification via input *dwSelectCurve*.

Starting the identification

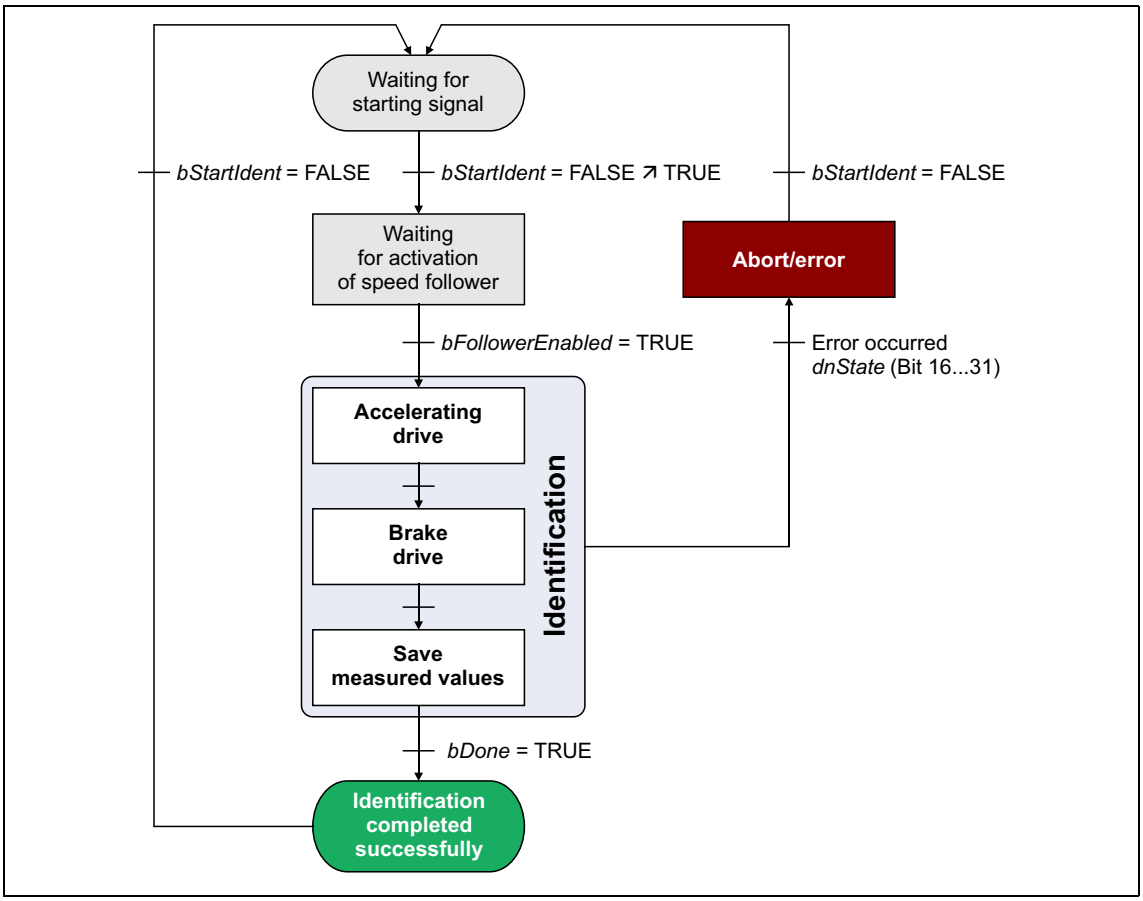
By setting the input *bStartIdent* from FALSE to TRUE, you start the identification.

Identification procedure

1. The basic function "Speed follower" is activated.
   - Output *bBusy* changes to TRUE.
2. If the drive is still coasting it is waited until the motor stops coasting first.
3. With the acceleration/deceleration time set in C04846 , the drive is accelerated to the maximum speed set in C04845.
4. When the drive has reached the maximum speed, it is braked to standstill again with the acceleration/deceleration time set in C04846.
5. The friction characteristic is saved in the memory area selected via the input *dwSelectCurve*.
6. After the identification has been completed successfully, the output *bBusy* is reset to FALSE, and the output *bDone* is set to TRUE.

State machine for the identification

The identification is effected according to the following state machine:



[5-155] State machine for the identification run

Cancelling the identification

If the input *bStartIdent* is reset to FALSE, the active identification is cancelled immediately. Resetting the input *bFollowerEnabled* to FALSE also results in the immediate abort.

- The output *bBusy* is reset to FALSE and the drive is brought to a standstill via quick stop.
- The cause for the abort is shown via status output *dnState*.

Delay of pulse enable

After activation of the function, the identification does not start immediately, but only after the delay time set in C04847 has elapsed. By this, the magnetic flux of the motor can fully build up first (only required for asynchronous machines).

- If required, the presetting is to be increased in the case of machines with great rotor time constants.

Smoothing the actual value signals

If incremental encoders are used, it may be required to filter the actual speed value signal.

- The time constant for the filter can be set in C04848; in the majority of cases the setting "10 ms" is sufficient.
- For smoothing the friction characteristic, it can also be helpful to filter the actual torque value signal; the time constant for this filter can be set in C04849.

5.207.4 Activating the friction compensation

The friction compensation is active if the input *bStartIdent* is set to FALSE, and the input *bEnableFrictionCurve* is set to TRUE.

- The current speed is preselected via the input *dnSpeedSetpoint_n*.
- If the actual speed value exceeds 100 %, the characteristic is extrapolated linearly.

Use of a linear characteristic

If the input *dwSelectCurve* is = 0 or unassigned, a linear characteristic is used for the friction compensation.

- The linear characteristic is specified via parameters C04854 and C04855:
  - C04854 = initial friction torque at speed "0"
  - C04855 = friction torque at maximum speed (C00011)
- The characteristic is limited internally to 0 ... 200 %.

Use of a previously automatically identified friction characteristic

If the input *dwSelectCurve* = 1 ... 4, the friction characteristic saved on the corresponding memory location is used instead of the linear characteristic.

▸ Identifying the friction characteristic automatically (📖 641)

Temperature compensation

Via the input *dnFrictionAdapt_n* a gain for the characteristic on a percentage basis is specified, for instance in order to compensate changes in the friction torque due to temperature changes.

Smoothing the speed reference value

For smoothing the speed reference value *dnSpeedSetpoint_n*, a filter time constant can be set in C04856.

Display of the precontrol signal

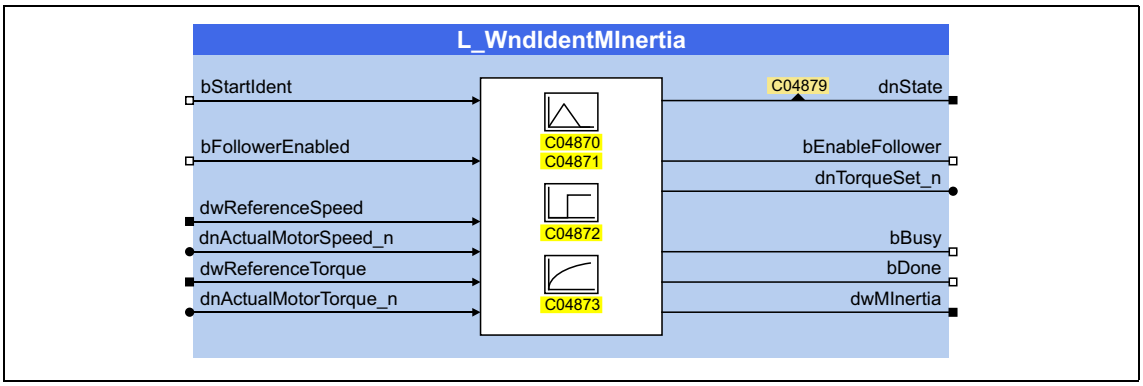
The precontrol signal generated is provided at output *dnFrictionTorque_n*. By setting the input *bNegOutput* to TRUE, you can output this signal in an inverted manner (change of direction of rotation).

5.208 L_WndIdentMInertia - identification of the moment of inertia

Function library:	LenzeWinder
Runtime software licence:	☐ Motion Control HighLevel ☑ Motion Control TopLevel

By means of this FB, the current moment of inertia can be identified automatically. The FB takes over the drive control on request and carries out the identification.

- Via *dnState* the current status of the identification run is displayed, or the cause that may have lead to an early abort is shown.
- After the identification run has been completed successfully, the moment of inertia is output via *dwMInertia*.



Inputs

Identifier/data type	Information/possible settings
bStartIdent BOOL	Start identification of the moment of inertia TRUE | Start identification.
bFollowerEnabled BOOL	Interface for the basic function "Torque follower" • Connect this input with the output *TF_bEnabled* of the SB **LS_TorqueFollower**.
dwReferenceSpeed DWORD	Motor reference speed in [rpm] • 15000 rpm ≡ 2^{26} ≡ 67108864 • Internal limitation to 50000 rpm • Connect this input to the output *MI_dwReferenceSpeed* of the SB **LS_MotorInterface**.
dnActualMotorSpeed_n DINT	Current speed of the motor shaft in [%] • 100 % ≡ motor reference speed (*dwReferenceSpeed*) • Connect this input to the output *MI_dnActualMotorSpeed_n* of the SB **LS_MotorInterface**.
dwReferenceTorque DWORD	Motor reference torque in [mNm] • 1000 mNm ≡ 1 Nm • Internal limitation to 3000 Nm. • Connect this input to the output *MI_dwReferenceTorque* of the SB **LS_MotorInterface**.
dnActualMotorTorque_n DINT	Current actual torque of the motor in [%] • 100 % ≡ motor reference torque (*dwReferenceTorque*) • Connect this input to the output *MI_dnActualMotorTorque_n* of the SB **LS_MotorInterface**.

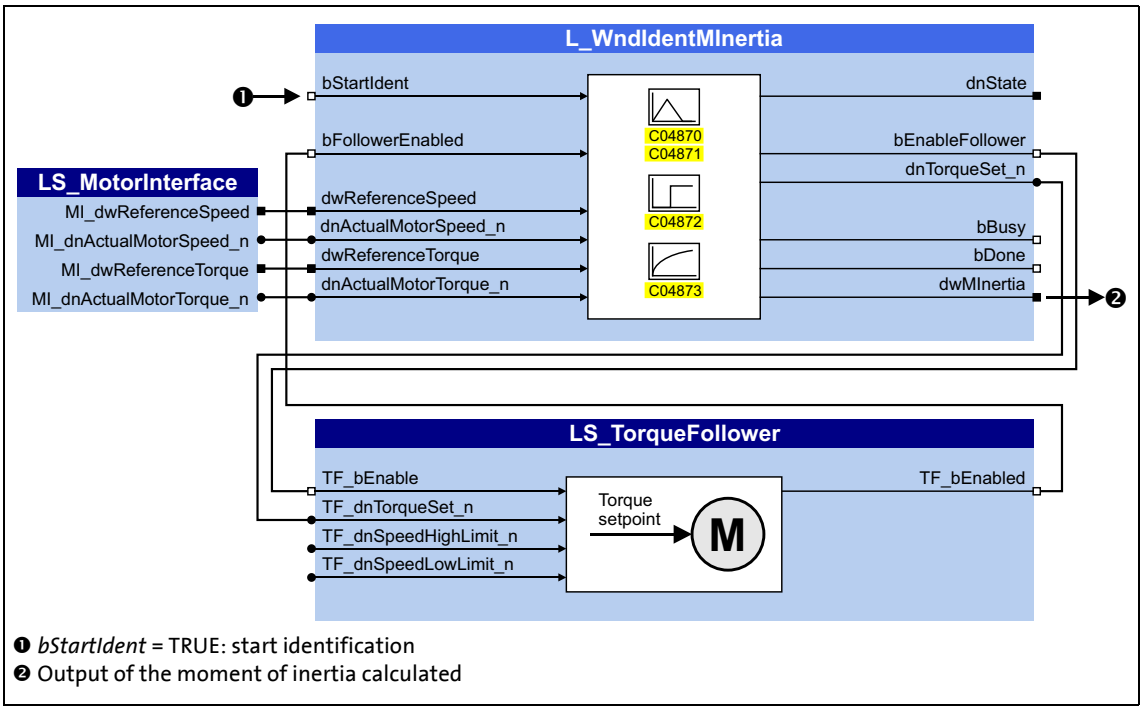
Outputs

Identifier/data type	Value/meaning	
dnState DINT	Status (bit coded) • Bits that are not listed have not been assigned with a status (always "0").	
	Status messages:	
	Bit 0	Identification started. Waiting for activation of the basic function "Torque follower".
	Bit 1	Waiting for motor standstill.
	Bit 2	Drive is accelerated to the specified speed.
	Bit 3	Drive has reached the specified speed and is braked to standstill.
	Bit 4	The moment of inertia (*dwMInertia*) is calculated.
	Bit 5	Identification completed successfully.
	Error messages:	
	Bit 16	Identification cancelled (*bStartIdent* = FALSE).
	Bit 17	Identification cancelled (*bFollowerEnabled* = FALSE).
	Bit 18	Specified maximum speed not reached.
	Bit 20	Speed reached at the end of acceleration process too low.
bEnableFollower BOOL	Interface for the basic function "Torque follower" • Connect this output with the input *TF_bEnable* of the SB **LS_TorqueFollower**.	
dnTorqueSet_n DINT	Interface for the basic function "Torque follower" • Connect this output with the input *TF_dnTorqueSet_n* of the SB **LS_TorqueFollower**.	
bBusy BOOL	Status signal "Detection of measured value active"	
	TRUE	Detection of measured value active.
bDone BOOL	Status signal "Identification completed successfully"	
	TRUE	Identification completed successfully.
dwMinertia DWORD	Identified moment of inertia in [0.1 kg cm^2]	

Parameters

Parameter	Possible settings			Info
C04870	5.00	%	100.00	Maximum motor speed • 100 % ≡ motor reference speed (C00011) • Initialisation: 10.00 %
C04871	-100.00	%	100.00	Maximum motor torque • 100 % ≡ motor reference torque (C00057/2) • Initialisation: 25.00 %
C04872	0	ms	1000	Start delay • Initialisation: 500 ms
C04873	0	ms	50	Current speed filter time • Initialisation: 5 ms
C04879	-2147483647		2147483647	Status • Display of the bit-coded output signal *dnState*.

5.208.1 Basic interconnection



[5-156] Basic interconnection with the basic function "Speed follower"

5.208.2 Carrying out the identification

Danger!

For the identification of the current moment of inertia, the drive is accelerated to the maximum speed set in C04870 and is braked to standstill again.

Note!

The rate of change of the speed results from the torque setpoint set in C04871 and the present moment of inertia.

Starting the identification

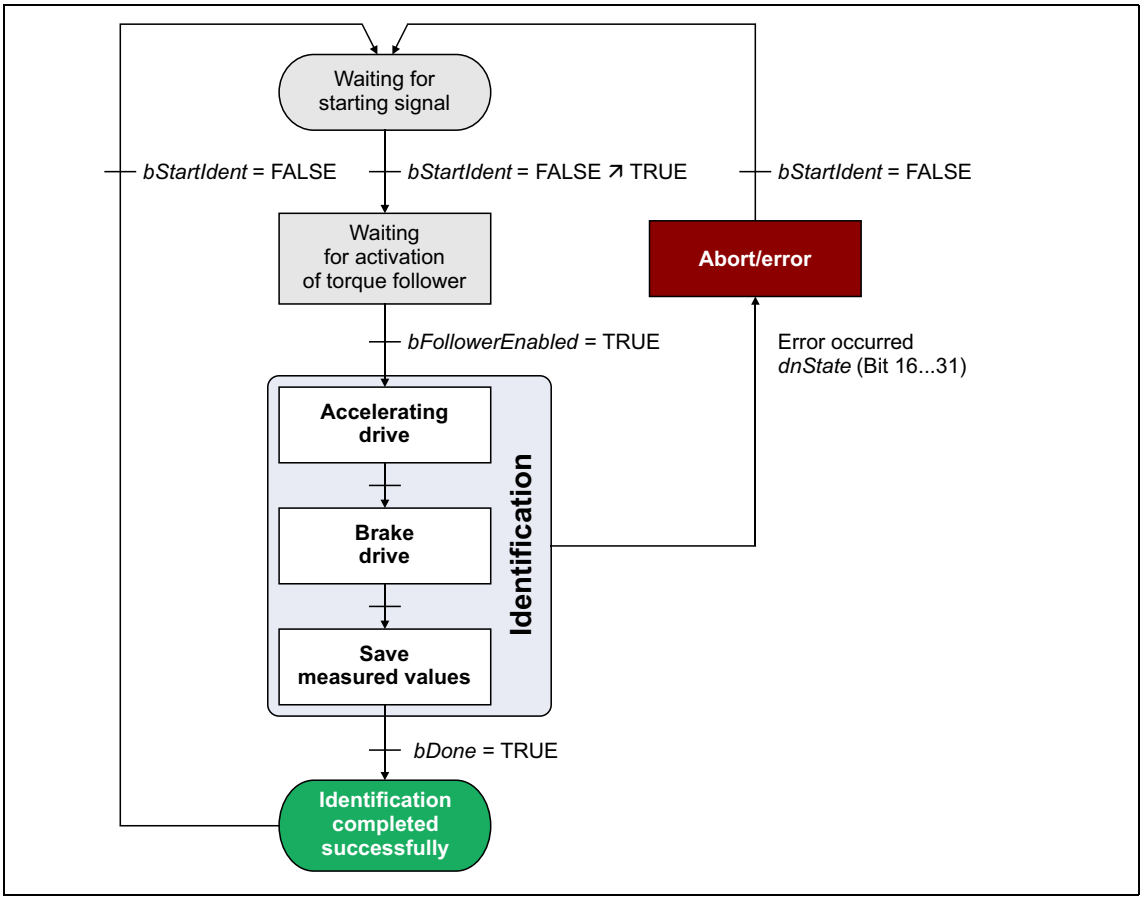
By setting the input *bStartIdent* from FALSE to TRUE, you start the identification.

Identification procedure

1. The basic function "Torque follower" is activated.
   - Output *bBusy* changes to TRUE.
2. If the drive is still coasting it is waited until the motor stops coasting first.
3. The drive is accelerated to the specified maximum speed (C04870) with the specified torque set-point (C04871).
4. When the drive has reached the maximum speed, it is braked to standstill again with the inverse torque setpoint.
5. The present moment of inertia is calculated.
   - At the output *dwMInertia* the calculated value is displayed as a physical value in the unit [0.01 kg cm^2].
6. After the identification has been completed successfully, the output *bBusy* is reset to FALSE, and the output *bDone* is set to TRUE.

State machine for the identification

The identification is effected according to the following state machine:



[5-157] State machine for the identification run

Cancelling the identification

If the input *bStartIdent* is reset to FALSE, the active identification is cancelled immediately. Resetting the input *bFollowerEnabled* to FALSE also results in the immediate abort.

- The output *bBusy* is reset to FALSE and the drive is brought to a standstill via quick stop.
- The cause for the abort is shown via status output *dnState*.

Delay of pulse enable

After activation of the function, the identification does not start immediately, but only after the delay time set in C04872 has elapsed. By this, the magnetic flux of the motor can fully build up first (only required for asynchronous machines).

- If required, the presetting is to be increased in the case of machines with great rotor time constants.

Smoothing the actual speed value signal

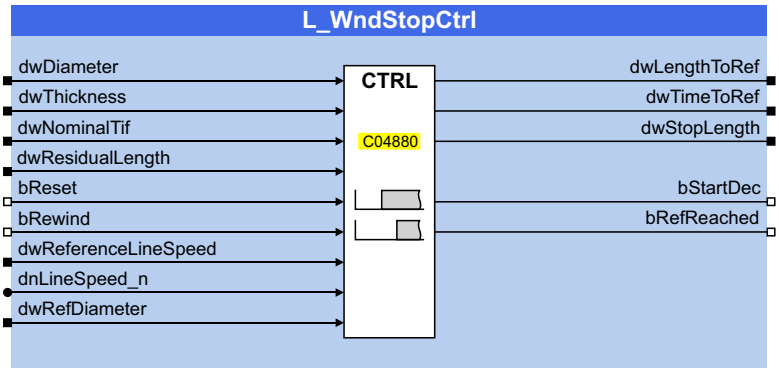
If incremental encoders are used, it may be required to filter the actual speed value signal.

- The time constant for the filter can be set in C04873; in the majority of cases the setting "10 ms" is sufficient.

5.209 L_WndStopCtrl - stop controller

Function library:	LenzeWinder
Runtime software licence:	☐ Motion Control HighLevel ☑ Motion Control TopLevel

This FB serves to monitor the residual length to the reference diameter, on the basis of the current diameter and the material thickness. Output signals for an accurately timed braking and a limit stop are provided.



Inputs

Identifier/data type	Information/possible settings	
dwDiameter DWORD	Current reel diameter in [µm] • 1000 µm ≡ 1 mm; 1000000 µm ≡ 1 m • Internal limitation to 10 m	
dwThickness DWORD	Current material thickness in [µm] • 1000 µm ≡ 1 mm • Internal limitation to 30 mm	
dwNominalTif DWORD	Rated deceleration time for braking process in [ms] • Is triggered if the pre-switch-off length has been reached (output *bStartDec* = TRUE). • The reference value can be selected in C04880 (Lenze setting: material speed set-point). • 1000 ms ≡ 1 s • Internal limitation to 10000 s.	
dwResidualLength DWORD	Remaining residual length after the stop in [mm] • 1000 mm ≡ 1 m; 1000000 mm ≡ 1 km • Internal limitation to 1000 km	
bReset BOOL	Reset stop controller	
	TRUE	The stop controller and the digital output signals *bStartDec* and *bRefReached* are reset.
bRewind BOOL	Selection of unwinding/rewinding operation	
	FALSE	Unwinding operation
	TRUE	Rewinding operation
dwReferenceLineSpeed DWORD	Maximum material speed in [cm/min] • 2500 cm/min ≡ 25.00 m/min • Internal limitation to 2000 m/min	

Identifier/data type	Information/possible settings
dnLineSpeed_n DINT	Material speed setpoint in [%] • 100 % ≡ maximum material speed (*dwReferenceLineSpeed*)
dwRefDiameter DWORD	Reference diameter for pre-switch-off and switch-off signal in [µm] • 1000 µm ≡ 1 mm; 1000000 µm ≡ 1 m • Internal limitation to 10 m

Outputs

Identifier/data type	Value/meaning	
dwLengthToRef DWORD	Residual length until reaching the reference value in [mm] • Internal limitation to 2000 km	
dwTimeToRef DWORD	Residual runtime until reaching the reference value in [s] • The calculation depends on C04880, either on the basis of the material speed set-point or the maximum material speed. • This value is only calculated at every 2. call of the FB (alternating with *dwStopLength*).	
dwStopLength DWORD	Change in length during the braking phase in [mm] • This value is only calculated at every 2. call of the FB (alternating with *dwTimeToRef*).	
bStartDec BOOL	Control signal for starting the braking process	
	TRUE	Pre-switch-off length reached.
bRefReached BOOL	Control signal for triggering the limit stop	
	TRUE	Reference length *dwRefLength* reached.

Parameters

Parameter	Possible settings		Info
C04880	Reference value =		Stop length of reference value • Reference value for the rated deceleration time *dwNominalTif* • Initialisation: maximum material speed
	0	Maximum material speed (*dwReferenceLineSpeed*)	
	1	Material speed setpoint (*dnLineSpeed_n*)	

5.209.1 Display of the residual length/residual runtime

Via the output *dwLengthToRef* the residual length until reaching the reference diameter (*dwRefDiameter*) is provided.

- Via *bRewind* you define whether the reference diameter is to be smaller (unwinder) or greater (rewinder) than the current diameter.
- If the reference diameter is reached or exceeded, the residual length is zero.
- Assuming that the maximum material speed *dwReferenceLineSpeed* is run, a corresponding residual runtime *dwTimeToRef* results from the residual length.

5.209.2 Activating the braking process

For the activation of the braking process, the residual length until reaching the reference diameter is compared to the remaining length after the stop (*dwResidualLength*), including the change in length during the braking phase (*dwStopLength*).

- The change in length during the braking phase is output at *dwStopLength*. It depends on the current material speed which is calculated from *dwReferenceLineSpeed* and *dnLineSpeed_n*, and on the reference deceleration time *dwNominalTif*.
- The signal for activating the braking process is output at *bStartDec*.

5.209.3 Triggering a limit stop

For the activation of a limit stop, the residual length until reaching the reference diameter is compared to zero (current diameter equals reference diameter).

- The signal for triggering the limit stop is output at *bRefReached*.

5.209.4 Resetting the calculation & trigger signals

By setting *bReset* to TRUE, you can reset the stop controller and the two digital outputs for the activation of the braking process/limit stop.

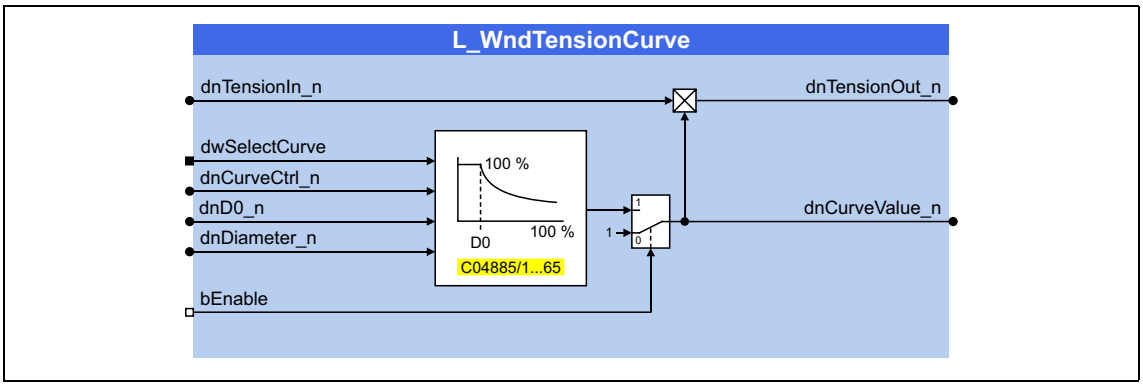
5.210 L_WndTensionCurve - tensile force characteristic

Function library:	LenzeWinder
Runtime software licence:	☐ Motion Control HighLevel ☑ Motion Control TopLevel

In the case of centre winders, the drive torque is transmitted from the centre across the individual layers to the winding surface. For smooth materials and large diameter areas in the majority of cases it is required to reduce the tensile force in a manner depending on the diameter.

This FB serves to generate a tensile force characteristic depending on the diameter.

- This characteristic is distinguished by an initial area with a constant evaluation (100 %) and a second area within which the tensile force is adapted to the diameter.



Inputs

Identifier/data type	Information/possible settings	
dnTensionIn_n DINT	Maximum setpoint of the tensile force in [%]	
dwSelectCurve DWORD	Selection of the characteristic	
	0	Linear tensile force characteristic
	1	Linear torque characteristic
	2	Course of the tensile force according to a specified characteristic (C04885/1...65)
dnCurveCtrl_n DINT	Gradient of the characteristic (torque characteristic) in [%] • Internal limitation to -100 ... +200 % • At 100 % there is a constant tensile force characteristic and therefore a torque proportionally increasing with D.	
dnD0_n DINT	Starting point of the characteristic • 100 % ≡ maximum diameter	
dnDiameter_n DINT	Current diameter • 100 % ≡ maximum diameter	
bEnable BOOL	Activate tensile force setpoint	
	FALSE	Tensile force setpoint is not affected.
	TRUE	Tensile force setpoint is multiplied with the characteristic value.

Outputs

Identifier/data type	Value/meaning
dnTensionOut_n DINT	Current tensile force setpoint in [%]
dnCurveValue_n DINT	Current characteristic value in [%]

Parameters

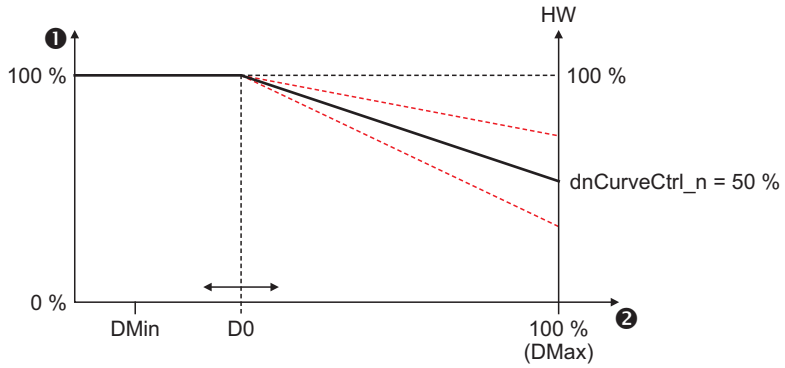
Parameter	Possible settings			Info
C04885/1...65	1	%	200	Table of characteristic values • The 65 values in [%] are distributed equally across the diameter area of *dnD0_n* to the maximum diameter.

5.210.1 Selection of the characteristic

The part of the characteristic that depends on the diameter can be generated for a linear course of the tensile force, a linear course of the torque, or on the basis of a specified characteristic. The corresponding selection is carried out via the input *dwSelectCurve*.

5.210.1.1 Characteristic for a linear course of the tensile force

Characteristic
❶ Tensile force
❷ Diameter



Function for D > D0:

$$100\,\% - \frac{D - D0}{100\,\% - D0} \cdot (100\,\% - HW)$$

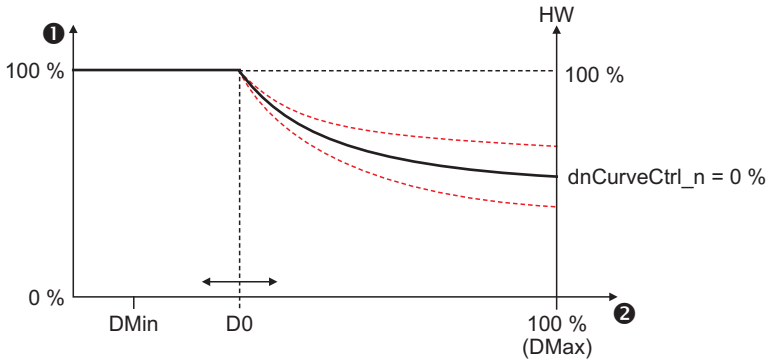
D	Current diameter (*dnDiameter_n*)
D0	Starting point for diameter dependency (*dnD0_n*)
HW	Evaluation of tensile force (*dnCurveCtrl_n*)

[5-158] Characteristic for a linear course of the tensile force (*dwSelectCurve* = 0)

- Via *dnD0_n* you define the diameter at which the reduction of the tensile force is to start.

5.210.1.2 Characteristic for a linear course of the torque

Characteristic
❶ Tensile force
❷ Diameter



Function for D > D0:

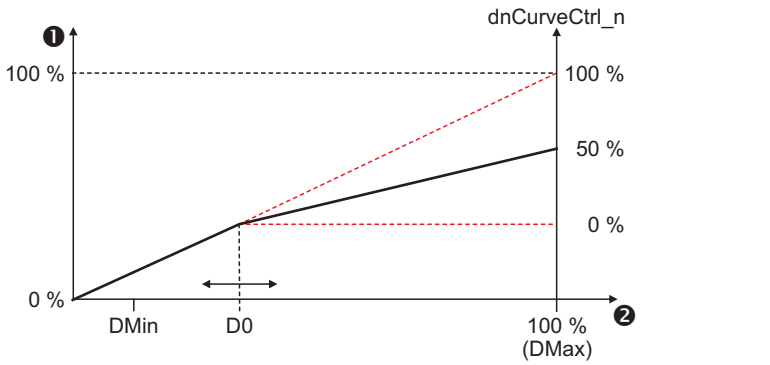
$$HW + \frac{(100\,\% - HW) \cdot D0}{D}$$

D	Current diameter (*dnDiameter_n*)
D0	Starting point for diameter dependency (*dnD0_n*)
HW	Evaluation of tensile force (*dnCurveCtrl_n*)

[5-159] Characteristic for a linear course of the torque (*dwSelectCurve* = 1)

- Via *dnD0_n* you define the diameter at which the reduction of the tensile force is to start.
- Since this characteristic selection focuses on the linear course of the torque, in this case you define the steepness of the torque curve via *dnCurveCtrl_n*:
  - At *dnCurveCtrl_n* = 100 % the tensile force remains unchanged.
  - At *dnCurveCtrl_n* = 0 % the tensile force is reduced to D0/DMax.

Characteristic
❶ Torque
❷ Diameter

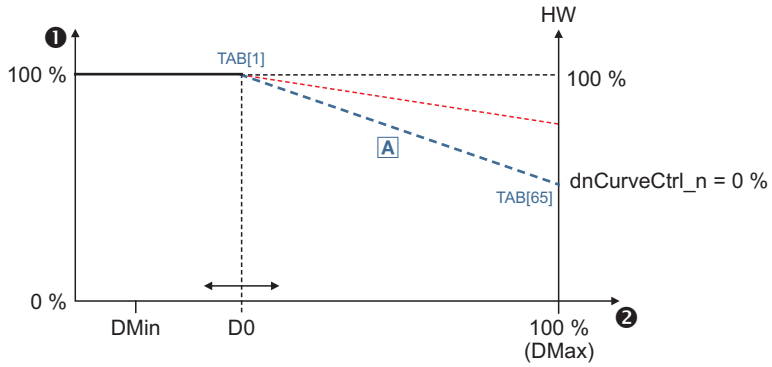


D	Current diameter (*dnDiameter_n*)
D0	Starting point for diameter dependency (*dnD0_n*)

5.210.1.3 Course of the tensile force according to a specified characteristic

The characteristic (A) can be parameterised in C04885/x and contains 65 values, equally distributed across the diameter area of D0 ... DMax:

Characteristic
❶ Tensile force
❷ Diameter



Function for D > D0:

$$100\,\% - \frac{(100\,\% - \mathrm{TAB[D\text{-}D0]}) \cdot (100\,\% - \mathrm{HW})}{100\,\%}$$

A	Table function (e.g. straight line)
TAB[D-D0]	Table value [1...65] for D-D0
D0	Starting point for diameter dependency (*dnD0_n*)
HW	Evaluation of tensile force (*dnCurveCtrl_n*)

[5-160] Course of the tensile force according to a specified characteristic (*dwSelectCurve* = 2)

- Via *dnD0_n* you define the start of the table area (D0).
- Via *dnCurveCtrl_n* the gradient of the characteristic can be altered.
  - At *dnCurveCtrl_n* = 0 % the gradient of the characteristic remains unchanged.

6 Table of attributes

The table of attributes contains information required for a communication to the controller via parameters.

How to read the table of attributes:

Column		Meaning	Entry	
Code		Parameter name	Cxxxxx	
Name		Parameter short text (display text)	*Text*	
Index	dec	Index under which the parameter will be addressed. The subindex for array variables corresponds to the Lenze subcode number.	24575 - Lenze code number	Only required if accessed via a bus system.
	hex		$5FFF_h$ - Lenze code number	
Data	DS	Data structure	S	Single variable (only one parameter element)
			A	Array variable (several parameter elements)
	DA	Number of array elements (subcodes)	*Number*	
	DT	Data type	BITFIELD_8	1 byte bit coded
			BITFIELD_16	2 bytes bit coded
			BITFIELD_32	4 bytes bit coded
			INTEGER_8	1 byte with sign
			INTEGER_16	2 bytes with sign
			INTEGER_32	4 bytes with sign
			UNSIGNED_8	1 byte without sign
			UNSIGNED_16	2 bytes without sign
			UNSIGNED_32	4 bytes without sign
			VISIBLE_STRING	ASCII string
	Factor	Factor for data transmission via a bus system, depending on the number of decimal positions	*Factor*	1 ≡ no decimal positions 10 ≡ 1 decimal position 100 ≡ 2 decimal positions 1000 ≡ 3 decimal positions 10000 ≡ 4 decimal positions
Access	R	Read access	☑ Reading permitted	
	W	Write access	☑ Writing permitted	
	CINH	Controller inhibit required	☑ Writing is only possible if the controller is inhibited	

Table of attributes

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
L_CamClutchPos										
C05480	Position unit	19095	4A97	A	2	VISIBLE_STRING	1	☑		
C05481	Declutch distance	19094	4A96	A	2	INTEGER_32	10000	☑	☑	
C05482	Ramp: Positive opening operation	19093	4A95	S	1	INTEGER_32	1000	☑	☑	
C05486	Input position	19089	4A91	A	2	INTEGER_32	10000	☑		
C05487	Output position	19088	4A90	S	1	INTEGER_32	10000	☑		
C05488	Output speed	19087	4A8F	S	1	INTEGER_32	10000	☑		
C05489	Status	19086	4A8E	S	1	INTEGER_32	1	☑		
L_CamContactor										
C05520	Position unit	19055	4A6F	A	2	VISIBLE_STRING	1	☑		
C05521	Dynamisation - switch-on position	19054	4A6E	S	1	INTEGER_32	1000	☑	☑	
C05522	Dynamisation - switch-off position	19053	4A6D	S	1	INTEGER_32	1000	☑	☑	
C05523	Hysteresis window	19052	4A6C	S	1	INTEGER_32	10000	☑	☑	
C05524	Track switch-over mode	19051	4A6B	S	1	INTEGER_32	1	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C05525	Cam type	19050	4A6A	S	1	INTEGER_32	1	☑		
C05526	Input position	19049	4A69	S	1	INTEGER_32	10000	☑		
C05527	Track number	19048	4A68	S	1	INTEGER_32	1	☑		
C05528	Cycle	19047	4A67	S	1	INTEGER_32	10000	☑		
C05529	Status	19046	4A66	S	1	INTEGER_32	1	☑		
L_CamCurve										
C05540	X position unit	19035	4A5B	A	2	VISIBLE_STRING	1	☑		
C05541	Y position unit	19034	4A5A	A	2	VISIBLE_STRING	1	☑		
C05547	Input value	19028	4A54	S	1	INTEGER_32	1	☑		
C05548	Output value	19027	4A53	S	1	INTEGER_32	1	☑		
C05549	Status	19026	4A52	S	1	INTEGER_32	1	☑		
L_CamGetAxisData										
C05400	Position unit	19175	4AE7	A	5	VISIBLE_STRING	1	☑		
C05401	Designation of the axis	19174	4AE6	S	1	INTEGER_32	1	☑		
C05406	Gearbox ratio - numerator	19169	4AE1	S	1	INTEGER_32	1	☑		
C05407	Gearbox ratio - denominator	19168	4AE0	S	1	INTEGER_32	1	☑		
C05408	Resulting gearbox factor	19167	4ADF	S	1	INTEGER_32	10000	☑		
C05409	Feed constant	19166	4ADE	S	1	INTEGER_32	10000	☑		
C05410	Measuring system	19165	4ADD	S	1	INTEGER_32	1	☑		
C05411	Resolution of a user unit	19164	4ADC	S	1	INTEGER_32	10000	☑		
C05412	Resolution of encoder revolution	19163	4ADB	S	1	INTEGER_32	1	☑		
C05413	Cycle	19162	4ADA	S	1	INTEGER_32	10000	☑		
C05414	Reference speed	19161	4AD9	S	1	INTEGER_32	10000	☑		
C05415	Maximum machine speed	19160	4AD8	S	1	INTEGER_32	10000	☑		
C05419	Status	19156	4AD4	S	1	INTEGER_32	1	☑		
L_CamPosMarker										
C05530	X position unit	19045	4A65	A	4	VISIBLE_STRING	1	☑		
C05534	Track number	19041	4A61	S	1	INTEGER_32	1	☑		
C05536	X coordinate of the position marker	19039	4A5F	A	4	INTEGER_32	10000	☑		
C05537	Y coordinate of the position marker	19038	4A5E	A	4	INTEGER_32	10000	☑		
C05539	Status	19036	4A5C	S	1	INTEGER_32	1	☑		
L_CamProfiler										
C05500	X position unit	19075	4A83	A	4	VISIBLE_STRING	1	☑		
C05501	Y position unit	19074	4A82	A	4	VISIBLE_STRING	1	☑		
C05502	X reference speed	19073	4A81	S	1	INTEGER_32	10000	☑		
C05505	Track switch-over mode	19070	4A7E	S	1	INTEGER_32	1	☑	☑	
C05506	Dynamisation time	19069	4A7D	S	1	INTEGER_32	1000	☑	☑	
C05507	Feed forward gain	19068	4A7C	S	1	INTEGER_32	100	☑	☑	
C05510	Input position X	19065	4A79	S	1	INTEGER_32	10000	☑		
C05511	Offset feedforward control X	19064	4A78	S	1	INTEGER_32	10000	☑		
C05514	Output position Y	19061	4A75	S	1	INTEGER_32	10000	☑		
C05515	Initial speed Y	19060	4A74	S	1	INTEGER_32	10000	☑		
C05516	Output acceleration Y	19059	4A73	S	1	INTEGER_32	10000	☑		
C05517	Feed length Y	19058	4A72	S	1	INTEGER_32	10000	☑		
C05518	Cycle X	19057	4A71	S	1	INTEGER_32	10000	☑		
C05519	Status	19056	4A70	S	1	INTEGER_32	1	☑		
L_CamSetContDataPDO										
C05560	Position unit	19015	4A47	S	1	VISIBLE_STRING	1	☑		
C05561	Cam type	19014	4A46	S	1	INTEGER_32	1	☑		
C05564	Switch-on position	19011	4A43	A	8	INTEGER_32	10000	☑		
C05565	Switch-off position	19010	4A42	A	8	INTEGER_32	10000	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C05566	Break time	19009	4A41	A	8	INTEGER_32	1000	☑		
C05569	Status	19006	4A3E	S	1	INTEGER_32	1	☑		
L_CamSetContDataSDO										
C05570	Position unit	19005	4A3D	S	1	VISIBLE_STRING	1	☑		
C05571	Cam type	19004	4A3C	S	1	INTEGER_32	1	☑	☑	
C05572	Number of cams	19003	4A3B	S	1	UNSIGNED_32	1	☑	☑	
C05574	Switch-on position	19001	4A39	A	8	INTEGER_32	10000	☑	☑	
C05575	Switch-off position	19000	4A38	A	8	INTEGER_32	10000	☑	☑	
C05576	Break time	18999	4A37	A	8	INTEGER_32	1000	☑	☑	
C05579	Status	18996	4A34	S	1	INTEGER_32	1	☑		
L_CamStretchAbs										
C05460	Position unit	19115	4AAB	A	3	VISIBLE_STRING	1	☑		
C05463	Input position	19112	4AA8	S	1	INTEGER_32	10000	☑		
C05464	Input acceleration	19111	4AA7	S	1	INTEGER_32	10000	☑		
C05465	Effective stretch factor	19110	4AA6	S	1	INTEGER_32	10000	☑		
C05466	Output position	19109	4AA5	S	1	INTEGER_32	10000	☑		
C05467	Output speed	19108	4AA4	S	1	INTEGER_32	10000	☑		
C05468	Output acceleration	19107	4AA3	S	1	INTEGER_32	10000	☑		
C05469	Status	19106	4AA2	S	1	INTEGER_32	1	☑		
L_CamStretchFeed										
C05440	Position unit	19135	4ABF	A	3	VISIBLE_STRING	1	☑		
C05447	Input position	19128	4AB8	S	1	INTEGER_32	10000	☑		
C05448	Input acceleration	19127	4AB7	S	1	INTEGER_32	10000	☑		
C05449	Feed length	19126	4AB6	S	1	INTEGER_32	10000	☑		
C05450	Effective stretch factor	19125	4AB5	S	1	INTEGER_32	10000	☑		
C05451	Starting position	19124	4AB4	S	1	INTEGER_32	10000	☑		
C05455	Output position	19120	4AB0	S	1	INTEGER_32	10000	☑		
C05456	Output speed	19119	4AAF	S	1	INTEGER_32	10000	☑		
C05457	Output acceleration	19118	4AAE	S	1	INTEGER_32	10000	☑		
C05458	Stretched position	19117	4AAD	A	2	INTEGER_32	10000	☑		
C05459	Status	19116	4AAC	S	1	INTEGER_32	1	☑		
L_CamSyncIn										
C05600	Unit	18975	4A1F	A	2	VISIBLE_STRING	1	☑		
C05601	Reference speed	18974	4A1E	S	1	INTEGER_32	10000	☑		
C05602	Maximum speed	18973	4A1D	S	1	INTEGER_32	10000	☑	☑	
C05603	Minimum clutch-in time	18972	4A1C	S	1	INTEGER_32	1000	☑	☑	
C05605	Input speed	18970	4A1A	S	1	INTEGER_32	10000	☑		
C05606	Sensor position	18969	4A19	A	3	INTEGER_32	10000	☑		
C05607	Output position	18968	4A18	S	1	INTEGER_32	10000	☑		
C05608	Output speed	18967	4A17	S	1	INTEGER_32	10000	☑		
C05609	Status	18966	4A16	S	1	INTEGER_32	1	☑		
L_DcBitShift										
C04190	Shift factor	20385	4FA1	S	1	INTEGER_32	1	☑	☑	
L_DcBitShiftByte										
C04194	Shift factor	20381	4F9D	S	1	INTEGER_32	1	☑	☑	
L_DcBitShiftDWord										
C04191	Shift factor	20384	4FA0	S	1	INTEGER_32	1	☑	☑	
L_DcBitShiftInt										
C04193	Shift factor	20382	4F9E	S	1	INTEGER_32	1	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
L_DcBitShiftWord										
C04192	Shift factor	20383	4F9F	S	1	INTEGER_32	1	☑	☑	
L_DevApplErr										
C05900	Function block: ID	18675	48F3	S	1	UNSIGNED_16	1	☑	☑	
C05901	Error number	18674	48F2	A	8	UNSIGNED_16	1	☑	☑	
C05902	Response in case of error	18673	48F1	A	8	UNSIGNED_8	1	☑	☑	
L_DevReadParDInt										
C05910	Code	18665	48E9	S	1	INTEGER_32	1	☑	☑	
C05911	Subcode	18664	48E8	S	1	INTEGER_32	1	☑	☑	
C05912	Hardware interface	18663	48E7	S	1	INTEGER_32	1	☑	☑	
C05913	Target address	18662	48E6	S	1	INTEGER_32	1	☑	☑	
C05914	Abort time	18661	48E5	S	1	INTEGER_32	1000	☑	☑	
C05915	Cyclic reading time	18660	48E4	S	1	INTEGER_32	1000	☑	☑	
L_DevWriteParDInt										
C05920	Code	18655	48DF	S	1	INTEGER_32	1	☑	☑	
C05921	Subcode	18654	48DE	S	1	INTEGER_32	1	☑	☑	
C05922	Hardware interface	18653	48DD	S	1	INTEGER_32	1	☑	☑	
C05923	Target address	18652	48DC	S	1	INTEGER_32	1	☑	☑	
C05924	Abort time	18651	48DB	S	1	INTEGER_32	1000	☑	☑	
C05925	Cyclic writing time	18650	48DA	S	1	INTEGER_32	1000	☑	☑	
L_EsClutchPos										
C05330	Unit	19245	4B2D	A	2	VISIBLE_STRING	1	☑		
C05331	Reference speed	19244	4B2C	S	1	INTEGER_32	10000	☑		
C05332	Mode	19243	4B2B	S	1	INTEGER_32	1	☑	☑	
C05334	Clutch-in time	19241	4B29	A	3	INTEGER_32	1000	☑	☑	
C05335	Input position	19240	4B28	A	3	INTEGER_32	10000	☑		
C05336	Basic speed	19239	4B27	S	1	INTEGER_32	10000	☑		
C05337	Output position	19238	4B26	S	1	INTEGER_32	10000	☑		
C05338	Output speed	19237	4B25	S	1	INTEGER_32	10000	☑		
C05339	Internal status	19236	4B24	S	1	INTEGER_32	1	☑		
L_EsEncoderConv										
C05270	Position unit	19305	4B69	A	2	VISIBLE_STRING	1	☑		
C05271	Encoder position evaluation	19304	4B68	S	1	INTEGER_32	1	☑	☑	
C05272	Encoder mounting position	19303	4B67	S	1	INTEGER_32	1	☑	☑	
C05273	Singleturn resolution	19302	4B66	S	1	INTEGER_32	1	☑	☑	
C05274	Multiturn resolution	19301	4B65	S	1	INTEGER_32	1	☑	☑	
C05275	Mode of position reconstruction	19300	4B64	S	1	INTEGER_32	1	☑	☑	
C05277	Initialisation time of the encoder	19298	4B62	S	1	INTEGER_32	1000	☑	☑	
C05280	Encoder input	19295	4B5F	S	1	INTEGER_32	1	☑		
C05281	Starting position	19294	4B5E	S	1	INTEGER_32	10000	☑		
C05282	Position output	19293	4B5D	S	1	INTEGER_32	10000	☑		
C05283	Speed output	19292	4B5C	S	1	INTEGER_32	10000	☑		
C05284	2nd encoder input	19291	4B5B	S	1	INTEGER_32	1	☑		
C05289	Status	19286	4B56	S	1	INTEGER_32	1	☑		
L_EsStretchIntegrate										
C05290	Position unit	19285	4B55	A	2	VISIBLE_STRING	1	☑		
C05291	Synchronised stretching/compression	19284	4B54	S	1	UNSIGNED_8	1	☑	☑	
C05293	Additional speed	19282	4B52	S	1	INTEGER_32	10000	☑		
C05294	Speed input	19281	4B51	S	1	INTEGER_32	10000	☑		
C05295	Effective stretch factor	19280	4B50	S	1	INTEGER_32	10000	☑		
C05296	Starting position	19279	4B4F	S	1	INTEGER_32	10000	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C05297	Position output	19278	4B4E	S	1	INTEGER_32	10000	☑		
C05298	Speed output	19277	4B4D	S	1	INTEGER_32	10000	☑		
C05299	Status	19276	4B4C	S	1	INTEGER_32	1	☑		
L_LdClutchAxisP										
C05070	Ramps for coupling	19505	4C31	A	3	INTEGER_32	1000	☑	☑	
C05071	Unit	19504	4C30	A	2	VISIBLE_STRING	1	☑		
C05072	Reference speed	19503	4C2F	S	1	INTEGER_32	10000	☑		
C05074	Status	19501	4C2D	S	1	INTEGER_32	1	☑		
C05076	Set position after declutching	19499	4C2B	S	1	INTEGER_32	10000	☑		
C05077	Actual position of the master angle	19498	4C2A	S	1	INTEGER_32	10000	☑		
C05078	Master position at the input	19497	4C29	S	1	INTEGER_32	10000	☑		
C05079	Master position at the output	19496	4C28	S	1	INTEGER_32	10000	☑		
L_LdClutchV										
C05080	Ramps for coupling	19495	4C27	A	3	UNSIGNED_32	1000	☑	☑	
C05081	Speed unit	19494	4C26	S	1	VISIBLE_STRING	1	☑		
C05082	Reference speed	19493	4C25	S	1	INTEGER_32	10000	☑		
C05083	S-ramp time	19492	4C24	S	1	UNSIGNED_32	1000	☑	☑	
C05085	Status	19490	4C22	S	1	INTEGER_32	1	☑		
C05087	Setpoint speed after declutching	19488	4C20	S	1	INTEGER_32	10000	☑		
C05088	Speed at the input	19487	4C1F	S	1	INTEGER_32	10000	☑		
C05089	Speed at the output	19486	4C1E	S	1	INTEGER_32	10000	☑		
L_LdConvAxisV										
C05160	Coupling measuring systems X and Y	19415	4BD7	S	1	UNSIGNED_32	1	☑	☑	☑
C05161	Cycles	19414	4BD6	A	2	INTEGER_32	10000	☑		
C05162	Coupling factors	19413	4BD5	A	2	INTEGER_32	10000	☑	☑	☑
C05163	User-defined unit	19412	4BD4	A	2	VISIBLE_STRING	1	☑		
C05165	Speeds	19410	4BD2	A	2	INTEGER_32	10000	☑		
C05169	Status	19406	4BCE	S	1	INTEGER_32	1	☑		
L_LdExtrapolate										
C05000	Number of extrapolation cycles	19575	4C77	S	1	UNSIGNED_16	1	☑	☑	
C05001	Unit	19574	4C76	A	2	VISIBLE_STRING	1	☑		
C05007	Speed at the output	19568	4C70	S	1	INTEGER_32	10000	☑		
C05008	Position at the output	19567	4C6F	S	1	INTEGER_32	10000	☑		
C05009	Extrapolation cycl. > selection	19566	4C6E	S	1	UNSIGNED_8	1	☑		
L_LdLinearCoupling										
C05181	Status	19394	4BC2	S	1	INTEGER_32	1	☑		
C05182	Unit	19393	4BC1	A	4	VISIBLE_STRING	1	☑		
C05183	Speed X at the output	19392	4BC0	S	1	INTEGER_32	10000	☑		
C05184	Speed Y at the output	19391	4BBF	S	1	INTEGER_32	10000	☑		
C05186	Position X at the input	19389	4BBD	S	1	INTEGER_32	10000	☑		
C05187	Trimming X	19388	4BBC	S	1	INTEGER_32	10000	☑		
C05188	Position X at the output	19387	4BBB	S	1	INTEGER_32	10000	☑		
C05189	Position Y at the output	19386	4BBA	S	1	INTEGER_32	10000	☑		
L_LdMonitFollowError										
C05175	Following error at the input	19400	4BC8	S	1	INTEGER_32	10000	☑		
C05176	Position unit	19399	4BC7	S	1	VISIBLE_STRING	1	☑		
C05178	Following error outside the tolerance	19397	4BC5	S	1	UNSIGNED_8	1	☑		
C05179	Following error monitoring is active.	19396	4BC4	S	1	UNSIGNED_8	1	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
L_LdMPot										
C05122	Positive speed	19453	4BFD	S	1	INTEGER_32	10000	☑	☑	
C05123	Negative speed	19452	4BFC	S	1	INTEGER_32	10000	☑	☑	
C05124	Ramp: Acceleration	19451	4BFB	S	1	UNSIGNED_32	1000	☑	☑	
C05125	Ramp: Deceleration	19450	4BFA	S	1	UNSIGNED_32	1000	☑	☑	
C05126	Ramp: Motor potentiometer off	19449	4BF9	S	1	UNSIGNED_32	1000	☑	☑	
C05127	Speed at the output	19448	4BF8	S	1	INTEGER_32	10000	☑		
C05128	Speed unit	19447	4BF7	S	1	VISIBLE_STRING	1	☑		
C05129	Reference speed	19446	4BF6	S	1	INTEGER_32	10000	☑		
L_LdPosCtrlLin										
C05050	Positioning mode	19525	4C45	S	1	UNSIGNED_32	1	☑	☑	
C05051	Mode after deactivation	19524	4C44	S	1	UNSIGNED_32	1	☑	☑	
C05052	Unit	19523	4C43	A	2	VISIBLE_STRING	1	☑		
C05053	Speeds	19522	4C42	A	2	INTEGER_32	10000	☑	☑	
C05054	Reference speed	19521	4C41	S	1	INTEGER_32	10000	☑		
C05056	Ramps	19519	4C3F	A	3	INTEGER_32	1000	☑	☑	
C05065	Speed at the output	19510	4C36	S	1	INTEGER_32	10000	☑		
C05067	Position at the output	19508	4C34	S	1	INTEGER_32	10000	☑		
C05069	Set position reached	19506	4C32	S	1	UNSIGNED_8	1	☑		
L_LdSetAxisVelocity										
C05034	Status	19541	4C55	S	1	INTEGER_32	1	☑		
C05035	Unit	19540	4C54	A	2	VISIBLE_STRING	1	☑		
C05036	Speed at the output	19539	4C53	S	1	INTEGER_32	10000	☑		
C05037	Master speed	19538	4C52	S	1	INTEGER_32	10000	☑		
C05041	Measured cycle	19534	4C4E	S	1	INTEGER_32	10000	☑		
C05042	Measured TP deviation	19533	4C4D	S	1	INTEGER_32	10000	☑		
C05043	Position at the output	19532	4C4C	S	1	INTEGER_32	10000	☑		
C05048	TP: Deviation detected	19527	4C47	S	1	UNSIGNED_8	1	☑		
C05049	Cycle start	19526	4C46	S	1	UNSIGNED_8	1	☑		
L_LdSyncOperation										
C05090	Coupling measuring systems X and Y	19485	4C1D	S	1	UNSIGNED_32	1	☑	☑	☑
C05091	User-defined units	19484	4C1C	A	2	VISIBLE_STRING	1	☑		
C05092	Coupling factors	19483	4C1B	A	2	INTEGER_32	10000	☑	☑	☑
C05093	Cycles	19482	4C1A	A	2	INTEGER_32	10000	☑		
C05094	Position X at the input	19481	4C19	S	1	INTEGER_32	10000	☑		
C05095	Trimming X	19480	4C18	S	1	INTEGER_32	10000	☑		
C05096	Speed X at the output	19479	4C17	S	1	INTEGER_32	10000	☑		
C05097	Speed Y at the output	19478	4C16	S	1	INTEGER_32	10000	☑		
C05098	Position X at the output	19477	4C15	S	1	INTEGER_32	10000	☑		
C05099	Status	19476	4C14	S	1	INTEGER_32	1	☑		
L_LdToolControl										
C05142	Status	19433	4BE9	S	1	INTEGER_32	1	☑		
C05143	Unit	19432	4BE8	A	2	VISIBLE_STRING	1	☑		
C05144	Reference speed	19431	4BE7	S	1	INTEGER_32	10000	☑		
C05147	Correction speed at the input	19428	4BE4	S	1	INTEGER_32	10000	☑		
C05148	Speed at the input	19427	4BE3	S	1	INTEGER_32	10000	☑		
C05149	Speed at the output	19426	4BE2	S	1	INTEGER_32	10000	☑		
C05150	Set position at the output	19425	4BE1	S	1	INTEGER_32	10000	☑		
C05151	Actual position at the output	19424	4BE0	S	1	INTEGER_32	10000	☑		
C05152	Following error	19423	4BDF	S	1	INTEGER_32	10000	☑		
C05153	Measured TP deviation	19422	4BDE	S	1	INTEGER_32	10000	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C05154	Measured cycle	19421	4BDD	S	1	INTEGER_32	10000	☑		
C05155	TP set position at the input	19420	4BDC	S	1	INTEGER_32	10000	☑		
L_LdVirtualMasterP										
C05010	VMaster: Operating mode	19565	4C6D	S	1	UNSIGNED_32	1	☑	☑	
C05011	Target position	19564	4C6C	S	1	INTEGER_32	10000	☑	☑	
C05012	Starting position	19563	4C6B	S	1	INTEGER_32	10000	☑	☑	
C05013	Master speed	19562	4C6A	A	2	INTEGER_32	10000	☑	☑	
C05014	Inching: Speed	19561	4C69	A	2	INTEGER_32	10000	☑	☑	
C05015	VMaster: Ramps	19560	4C68	A	3	INTEGER_32	1000	☑	☑	
C05016	Unit	19559	4C67	A	2	VISIBLE_STRING	1	☑		
C05017	Reference speed	19558	4C66	S	1	INTEGER_32	10000	☑		
C05024	Status	19551	4C5F	S	1	INTEGER_32	1	☑		
C05025	Master position in the target	19550	4C5E	S	1	UNSIGNED_8	1	☑		
C05026	Cycle	19549	4C5D	S	1	INTEGER_32	10000	☑		
C05027	Master position at the output	19548	4C5C	S	1	INTEGER_32	10000	☑		
C05028	External master position at the input	19547	4C5B	S	1	INTEGER_32	10000	☑		
L_LdVirtualMasterV										
C05100	VMaster: Operating mode	19475	4C13	S	1	UNSIGNED_32	1	☑	☑	
C05101	Speed unit	19474	4C12	S	1	VISIBLE_STRING	1	☑		
C05102	Reference speed	19473	4C11	S	1	INTEGER_32	10000	☑		
C05103	Master speed	19472	4C10	A	2	INTEGER_32	10000	☑	☑	
C05104	Inching: Speed	19471	4C0F	A	2	INTEGER_32	10000	☑	☑	
C05105	VMaster: Ramps	19470	4C0E	A	3	UNSIGNED_32	1000	☑	☑	
C05114	Status	19461	4C05	S	1	INTEGER_32	1	☑		
C05117	Speed at the output	19458	4C02	S	1	INTEGER_32	10000	☑		
C05118	External speed at the input	19457	4C01	S	1	INTEGER_32	10000	☑		
L_PosPositionerTable										
C04700	Teaching	19875	4DA3	A	4	UNSIGNED_8	1	☑	☑	
C04701	Table: Position x	19874	4DA2	A	75	INTEGER_32	10000	☑	☑	
C04702	Table: Speed x	19873	4DA1	A	50	INTEGER_32	10000	☑	☑	
C04703	Table: Acceleration x	19872	4DA0	A	50	INTEGER_32	10000	☑	☑	
C04704	Table: S-ramp time x	19871	4D9F	A	50	INTEGER_32	1000	☑	☑	
C04705	DIS:bPosTeach	19870	4D9E	A	4	UNSIGNED_8	1	☑		
C04706	Table position	19869	4D9D	A	5	UNSIGNED_32	1	☑		
C04707	DIS:dnTeachPos	19868	4D9C	A	4	INTEGER_32	10000	☑		
C04708	DIS: dnState	19867	4D9B	S	1	INTEGER_32	1	☑		
C04709	Position unit	19866	4D9A	A	3	VISIBLE_STRING	1	☑		
C04710	Positioning mode	19865	4D99	A	75	UNSIGNED_16	1	☑	☑	
C04711	Position profile n = table pos.	19864	4D98	A	75	UNSIGNED_16	1	☑	☑	
C04712	Speed profile n = table pos.	19863	4D97	A	75	UNSIGNED_16	1	☑	☑	
C04713	Accel. profile n = table pos.	19862	4D96	A	75	UNSIGNED_16	1	☑	☑	
C04714	Decel. profile n = table pos.	19861	4D95	A	75	UNSIGNED_16	1	☑	☑	
C04715	S-ramp time profile n = table pos.	19860	4D94	A	75	UNSIGNED_16	1	☑	☑	
C04716	Final speed profile n = table pos.	19859	4D93	A	75	UNSIGNED_16	1	☑	☑	
C04717	Starting pos. TP profile n = table pos.	19858	4D92	A	75	UNSIGNED_16	1	☑	☑	
C04718	End pos. TP profile n = table pos.	19857	4D91	A	75	UNSIGNED_16	1	☑	☑	
C04719	Sequence profile for profile n with TP	19856	4D90	A	75	UNSIGNED_16	1	☑	☑	
C04720	Sequence profile for profile n without TP	19855	4D8F	A	75	UNSIGNED_16	1	☑	☑	
C04721	TP evaluation for profile n	19854	4D8E	A	75	UNSIGNED_32	1	☑	☑	
C04723	S-ramp mode	19852	4D8C	A	75	UNSIGNED_16	1	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C04724	Second S-ramp time = table pos.	19851	4D8B	A	75	UNSIGNED_16	1	☑	☑	
L_PosProfileTable										
C04750	Positioning mode	19825	4D71	A	4	UNSIGNED_32	1	☑	☑	
C04751	Position profile n	19824	4D70	A	4	INTEGER_32	10000	☑	☑	
C04752	Speed profile n	19823	4D6F	A	4	INTEGER_32	10000	☑	☑	
C04753	Acceleration profile n	19822	4D6E	A	4	INTEGER_32	10000	☑	☑	
C04754	Deceleration profile n	19821	4D6D	A	4	INTEGER_32	10000	☑	☑	
C04755	S-ramp time profile n	19820	4D6C	A	4	UNSIGNED_32	1000	☑	☑	
C04756	Sequence profile for profile n with TP	19819	4D6B	A	4	UNSIGNED_32	1	☑	☑	
C04757	TP evaluation for profile n	19818	4D6A	A	4	UNSIGNED_32	1	☑	☑	
C04758	"Teach" position	19817	4D69	S	1	UNSIGNED_8	1	☑	☑	
C04759	Position unit	19816	4D68	A	3	VISIBLE_STRING	1	☑		
C04760	DIS: dnExtPos	19815	4D67	A	2	INTEGER_32	10000	☑		
C04761	DIS: dwProfileNumber	19814	4D66	A	2	UNSIGNED_32	1	☑		
C04763	DIS:bPosTeach	19812	4D64	S	1	UNSIGNED_8	1	☑		
C04764	DIS:bBusy	19811	4D63	S	1	UNSIGNED_8	1	☑		
C04765	DIS: dnState	19810	4D62	S	1	INTEGER_32	1	☑		
C04767	S-ramp mode profile n	19808	4D60	A	4	UNSIGNED_32	1	☑	☑	
C04768	Second S-ramp time profile n	19807	4D5F	A	4	UNSIGNED_32	1000	☑	☑	
L_PosSequencer										
C04500	Action in sequence step n	20075	4E6B	A	100	UNSIGNED_32	1	☑	☑	
C04501	Current step	20074	4E6A	S	1	UNSIGNED_32	1	☑		
C04502	Sequence control: Status	20073	4E69	S	1	UNSIGNED_32	1	☑		
C04503	Sequence control: Control word	20072	4E68	S	1	UNSIGNED_32	1	☑	☑	
C04504	Step for bCancel = TRUE	20071	4E67	S	1	UNSIGNED_32	1	☑	☑	
C04505	Current step: Comment	20070	4E66	S	1	VISIBLE_STRING	1	☑		
C04510	Pos. action n: Comment	20065	4E61	A	50	VISIBLE_STRING	1	☑	☑	
C04511	Pos. action n: Start with	20064	4E60	A	50	UNSIGNED_32	1	☑	☑	
C04512	Pos. action n: input polarity	20063	4E5F	A	50	UNSIGNED_8	1	☑	☑	
C04513	Pos. action n: Profile number	20062	4E5E	A	50	UNSIGNED_32	1	☑	☑	
C04514	Pos. action n: Jump destination	20061	4E5D	A	50	UNSIGNED_32	1	☑	☑	
C04515	Pos. action n: Monitoring time	20060	4E5C	A	50	UNSIGNED_32	1000	☑	☑	
C04516	Pos. action n: Jump destination monit.	20059	4E5B	A	50	UNSIGNED_32	1	☑	☑	
C04520	Switch. act. n: output switch. A	20055	4E57	A	25	UNSIGNED_32	1	☑	☑	
C04521	Switch. act. n: pol. switch. A	20054	4E56	A	25	UNSIGNED_8	1	☑	☑	
C04522	Switch. act. n: output switch. B	20053	4E55	A	25	UNSIGNED_32	1	☑	☑	
C04523	Switch. act. n: Pol. switch. B	20052	4E54	A	25	UNSIGNED_8	1	☑	☑	
C04530	Branch act. n: Input for jump	20045	4E4D	A	25	UNSIGNED_32	1	☑	☑	
C04531	Branch act. n: Input polarity	20044	4E4C	A	25	UNSIGNED_8	1	☑	☑	
C04532	Branch act. n: Jump destination	20043	4E4B	A	25	UNSIGNED_32	1	☑	☑	
C04540	Var. branch n: Jump destin. 1	20035	4E43	A	5	UNSIGNED_32	1	☑	☑	
C04541	Var. branch n: Jump destin. 2	20034	4E42	A	5	UNSIGNED_32	1	☑	☑	
C04542	Var. branch n: Jump destin. 3	20033	4E41	A	5	UNSIGNED_32	1	☑	☑	
C04543	Var. branch n: Jump destin. 4	20032	4E40	A	5	UNSIGNED_32	1	☑	☑	
C04544	Var. branch n: Jump destin. 5	20031	4E3F	A	5	UNSIGNED_32	1	☑	☑	
C04545	Var. branch n: Jump destin. 6	20030	4E3E	A	5	UNSIGNED_32	1	☑	☑	
C04546	Var. branch n: Jump destin. 7	20029	4E3D	A	5	UNSIGNED_32	1	☑	☑	
C04547	Var. branch n: Jump destin. 8	20028	4E3C	A	5	UNSIGNED_32	1	☑	☑	
C04548	Var. branch n: Jump destin. 9	20027	4E3B	A	5	UNSIGNED_32	1	☑	☑	
C04549	Var. branch n: Jump destin. 10	20026	4E3A	A	5	UNSIGNED_32	1	☑	☑	
C04550	Var. branch n: Jump destin. 11	20025	4E39	A	5	UNSIGNED_32	1	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C04551	Var. branch n: Jump destin. 12	20024	4E38	A	5	UNSIGNED_32	1	☑	☑	
C04552	Var. branch n: Jump destin. 13	20023	4E37	A	5	UNSIGNED_32	1	☑	☑	
C04553	Var. branch n: Jump destin. 14	20022	4E36	A	5	UNSIGNED_32	1	☑	☑	
C04554	Var. branch n: Jump destin. 15	20021	4E35	A	5	UNSIGNED_32	1	☑	☑	
C04555	Var. branch n: Jump destin. 16	20020	4E34	A	5	UNSIGNED_32	1	☑	☑	
C04556	Var. branch n: Jump destin. 17	20019	4E33	A	5	UNSIGNED_32	1	☑	☑	
C04557	Var. branch n: Jump destin. 18	20018	4E32	A	5	UNSIGNED_32	1	☑	☑	
C04558	Var. branch n: Jump destin. 19	20017	4E31	A	5	UNSIGNED_32	1	☑	☑	
C04559	Var. branch n: Jump destin. 20	20016	4E30	A	5	UNSIGNED_32	1	☑	☑	
C04570	Wait. act. n: Comment	20005	4E25	A	25	VISIBLE_STRING	1	☑	☑	
C04571	Wait. act. n: Waiting time	20004	4E24	A	25	UNSIGNED_32	1000	☑	☑	
C04572	Wait. act. n: Input for "Next""	20003	4E23	A	25	UNSIGNED_32	1	☑	☑	
C04573	Wait. act. n: Input polarity	20002	4E22	A	25	UNSIGNED_8	1	☑	☑	
C04580	Counter set act. n: Counter no.	19995	4E1B	A	10	UNSIGNED_32	1	☑	☑	
C04581	Counter set act. n: Starting value	19994	4E1A	A	10	INTEGER_32	1	☑	☑	
C04582	Counter n: Current counter content	19993	4E19	A	10	INTEGER_32	1	☑		
C04590	Count. act. n: Counter no.	19985	4E11	A	25	UNSIGNED_32	1	☑	☑	
C04591	Count. act. n: Step value	19984	4E10	A	25	INTEGER_32	1	☑	☑	
C04592	Count. act. n: comparison value	19983	4E0F	A	25	INTEGER_32	1	☑	☑	
C04593	Count. act. n: Jump destination	19982	4E0E	A	25	UNSIGNED_32	1	☑	☑	
C04594	Count. act. n: Comparison op.	19981	4E0D	A	25	UNSIGNED_32	1	☑	☑	
C04600	Stand-by act. n: Comment	19975	4E07	A	5	VISIBLE_STRING	1	☑	☑	
C04601	Stand-by act.n: Input for "End""	19974	4E06	A	5	UNSIGNED_32	1	☑	☑	
C04602	Stand-by act. n: Input polarity	19973	4E05	A	5	UNSIGNED_8	1	☑	☑	
C04690	DIS: bHomingDone	19885	4DAD	A	15	UNSIGNED_8	1	☑		
C04691	DIS: dnState	19884	4DAC	S	1	INTEGER_32	1	☑		
C04692	DIS: dwBranch1	19883	4DAB	A	8	UNSIGNED_32	1	☑		
C04693	DIS: dwDigitalInputs	19882	4DAA	A	2	UNSIGNED_32	1	☑		
L_SdAccToUnit										
C04319	Unit	20256	4F20	S	1	VISIBLE_STRING	1	☑		
L_SdDelayComp										
C04320	Rate time	20255	4F1F	S	1	INTEGER_32	1000	☑	☑	
C04321	Position unit	20254	4F1E	S	1	VISIBLE_STRING	1	☑		
C04322	Position at the input	20253	4F1D	S	1	INTEGER_32	10000	☑		
C04323	Position at the output	20252	4F1C	S	1	INTEGER_32	10000	☑		
L_SdGetAxisData										
C04410	Gearbox factor	20165	4EC5	A	2	INTEGER_32	1	☑		
C04411	Feed constant	20164	4EC4	S	1	INTEGER_32	10000	☑		
C04412	Unit	20163	4EC3	S	1	VISIBLE_STRING	1	☑		
C04413	Time unit	20162	4EC2	S	1	VISIBLE_STRING	1	☑		
C04414	Axis name	20161	4EC1	S	1	VISIBLE_STRING	1	☑		
C04416	Cycle	20159	4EBF	S	1	INTEGER_32	10000	☑		
C04418	Resolution of an encoder revolution	20157	4EBD	S	1	UNSIGNED_32	1	☑		
C04420	Reference speed	20155	4EBB	S	1	INTEGER_32	10000	☑		
C04422	Traversing range	20153	4EB9	S	1	UNSIGNED_32	1	☑		
C04424	Resulting gearbox factor	20151	4EB7	S	1	INTEGER_32	1000	☑		
C04425	Resolution of a unit	20150	4EB6	S	1	INTEGER_32	10000	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
L_SdGetPosition										
C04271	Status	20304	4F50	S	1	INTEGER_32	1	☑		
C04272	Position unit	20303	4F4F	S	1	VISIBLE_STRING	1	☑		
C04273	Position value	20302	4F4E	S	1	INTEGER_32	10000	☑		
L_SdGetSpeed										
C04281	Status	20294	4F46	S	1	INTEGER_32	1	☑		
C04282	Speed unit	20293	4F45	S	1	VISIBLE_STRING	1	☑		
C04283	Speed value	20292	4F44	S	1	INTEGER_32	10000	☑		
L_SdIntegrateAxis										
C04350	Position unit	20225	4F01	A	2	VISIBLE_STRING	1	☑		
C04352	Speed input	20223	4EFF	S	1	INTEGER_32	10000	☑		
C04353	Starting position	20222	4EFE	S	1	INTEGER_32	10000	☑		
C04354	TP position	20221	4EFD	S	1	INTEGER_32	10000	☑		
C04357	Output position	20218	4EFA	S	1	INTEGER_32	10000	☑		
C04358	Speed output	20217	4EF9	S	1	INTEGER_32	10000	☑		
C04359	Status	20216	4EF8	S	1	INTEGER_32	1	☑		
L_SdIntegrateLimit										
C04325	Position unit	20250	4F1A	A	2	VISIBLE_STRING	1	☑		
C04326	Speed at the input	20249	4F19	A	2	INTEGER_32	10000	☑		
C04327	Starting position	20248	4F18	A	2	INTEGER_32	10000	☑		
C04329	Status	20246	4F16	S	1	INTEGER_32	1	☑		
L_SdInterExtrapolateAny										
C04445	Operation mode	20130	4EA2	S	1	UNSIGNED_32	1	☑	☑	
C04446	Number of interpolation/extrapolation steps	20129	4EA1	S	1	UNSIGNED_32	1	☑	☑	
C04447	Change of direction of rotation during extrapolation	20128	4EA0	S	1	UNSIGNED_8	1	☑	☑	
L_SdInterExtrapolatePosition										
C04430	Position unit	20145	4EB1	A	2	VISIBLE_STRING	1	☑		
C04431	Operation mode	20144	4EB0	S	1	UNSIGNED_32	1	☑	☑	
C04432	Number of interpolation/extrapolation steps	20143	4EAF	S	1	UNSIGNED_32	1	☑	☑	
C04433	Change of direction of rotation during extrapolation	20142	4EAE	S	1	UNSIGNED_8	1	☑	☑	
C04434	Number of the interpolation cycles	20141	4EAD	S	1	UNSIGNED_32	1	☑	☑	
C04435	Telegram failure detection	20140	4EAC	S	1	UNSIGNED_32	1	☑	☑	
C04436	Position at the input	20139	4EAB	S	1	INTEGER_32	10000	☑		
C04437	Position at the output	20138	4EAA	S	1	INTEGER_32	10000	☑		
C04438	Speed at the output	20137	4EA9	S	1	INTEGER_32	10000	☑		
L_SdInterpolate										
C04400	Number of the interpolation cycles	20175	4ECF	S	1	UNSIGNED_32	1	☑	☑	
C04401	Position unit	20174	4ECE	A	2	VISIBLE_STRING	1	☑		
C04402	Position at the input	20173	4ECD	S	1	INTEGER_32	10000	☑		
C04403	Position at the output	20172	4ECC	S	1	INTEGER_32	10000	☑		
C04404	Speed at the output	20171	4ECB	S	1	INTEGER_32	10000	☑		
L_SdLimitSpeed										
C04300	Speed at the output	20275	4F33	S	1	INTEGER_32	10000	☑		
C04301	Speed at the input	20274	4F32	S	1	INTEGER_32	10000	☑		
C04302	Speed limits	20273	4F31	A	2	INTEGER_32	10000	☑		
C04303	Limitation enabled	20272	4F30	S	1	UNSIGNED_8	1	☑		
C04304	Limitation active	20271	4F2F	S	1	UNSIGNED_8	1	☑		
C04305	Status	20270	4F2E	S	1	INTEGER_32	1	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C04306	Speed unit	20269	4F2D	S	1	VISIBLE_STRING	1	☑		
C04307	Reference speed	20268	4F2C	S	1	INTEGER_32	10000	☑		
L_SdMotorPot										
C04360	Upper limit value	20215	4EF7	A	2	INTEGER_32	100	☑	☑	
C04361	Acceleration time	20214	4EF6	A	2	INTEGER_32	1000	☑	☑	
C04362	Deactivation function	20213	4EF5	S	1	INTEGER_32	1	☑	☑	
C04363	Initialisation function	20212	4EF4	S	1	INTEGER_32	1	☑	☑	
C04368	Output setpoint	20207	4EEF	S	1	INTEGER_32	100	☑		
C04369	Status	20206	4EEE	S	1	INTEGER_32	1	☑		
L_SdPosToUnit										
C04312	Unit	20263	4F27	S	1	VISIBLE_STRING	1	☑		
L_SdProcessController										
C04375	Acceleration / deceleration time	20200	4EE8	S	1	INTEGER_32	1000	☑	☑	
C04376	Gain of red. system deviation	20199	4EE7	S	1	INTEGER_32	100	☑	☑	
C04377	Field of red. system deviation	20198	4EE6	S	1	INTEGER_32	100	☑	☑	
C04378	Controller gain	20197	4EE5	S	1	INTEGER_32	10000	☑	☑	
C04379	Controller reset time	20196	4EE4	S	1	INTEGER_32	1000	☑	☑	
C04380	Controller rate time	20195	4EE3	S	1	INTEGER_32	1000	☑	☑	
C04381	Positive output limit value	20194	4EE2	S	1	INTEGER_32	100	☑	☑	
C04382	Negative output limit value	20193	4EE1	S	1	INTEGER_32	100	☑	☑	
C04383	Filter time	20192	4EE0	S	1	INTEGER_32	1000	☑	☑	
C04384	Characteristic - controller adaptation	20191	4EDF	A	5	INTEGER_32	100	☑	☑	
C04385	Characteristic - controller adaptation y(x1)	20190	4EDE	A	5	INTEGER_32	100	☑	☑	
C04386	Ramp function generator deceleration time	20189	4EDD	S	1	INTEGER_32	1000	☑	☑	
C04388	Process setpoint	20187	4EDB	A	9	INTEGER_32	100	☑		
C04389	Load actual value	20186	4EDA	A	4	INTEGER_32	1	☑		
L_SdRampGenerator										
C04340	Acceleration time/deceleration time	20235	4F0B	S	1	UNSIGNED_32	1000	☑	☑	
C04341	S-ramp time	20234	4F0A	S	1	UNSIGNED_32	1000	☑	☑	
C04342	Setpoint at the input	20233	4F09	S	1	INTEGER_32	100	☑		
C04343	Setpoint at the output	20232	4F08	S	1	INTEGER_32	100	☑		
L_SdRampGeneratorAny										
C04330	Signal unit	20245	4F15	S	1	VISIBLE_STRING	1	☑		
C04331	Reference value	20244	4F14	S	1	UNSIGNED_32	10000	☑		
C04332	Acceleration / deceleration time	20243	4F13	A	2	UNSIGNED_32	1000	☑	☑	
C04333	S-ramp time for acceleration/deceleration	20242	4F12	A	2	UNSIGNED_32	1000	☑	☑	
C04336	Setpoint at the input/output	20239	4F0F	A	2	INTEGER_32	10000	☑		
L_SdSetAxisData										
C04240	Gearbox factor numerator	20335	4F6F	S	1	INTEGER_32	1	☑	☑	☑
C04241	Gearbox factor denominator	20334	4F6E	S	1	INTEGER_32	1	☑	☑	☑
C04242	Feed constant	20333	4F6D	S	1	INTEGER_32	10000	☑	☑	☑
C04243	Unit	20332	4F6C	S	1	UNSIGNED_32	1	☑	☑	
C04244	User-defined unit	20331	4F6B	S	1	VISIBLE_STRING	1	☑	☑	
C04245	Time unit	20330	4F6A	S	1	UNSIGNED_32	1	☑		
C04246	Axis name	20329	4F69	A	2	VISIBLE_STRING	1	☑		
C04248	Cycle	20327	4F67	S	1	INTEGER_32	10000	☑	☑	☑
C04249	Resolution of an encoder revolution	20326	4F66	S	1	UNSIGNED_32	1	☑	☑	☑
C04251	Reference speed	20324	4F64	S	1	INTEGER_32	10000	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C04252	Traversing range	20323	4F63	S	1	INTEGER_32	1	☑	☑	☑
C04258	Resulting gearbox factor	20317	4F5D	S	1	INTEGER_32	1000	☑		
C04259	Resolution of a unit	20316	4F5C	S	1	INTEGER_32	10000	☑		
L_SdSetPosition										
C04276	Position value	20299	4F4B	S	1	INTEGER_32	10000	☑	☑	
C04277	Position unit	20298	4F4A	S	1	VISIBLE_STRING	1	☑		
C04278	Status	20297	4F49	S	1	INTEGER_32	1	☑		
L_SdSetSpeed										
C04286	Speed value	20289	4F41	S	1	INTEGER_32	10000	☑	☑	
C04287	Speed unit	20288	4F40	S	1	VISIBLE_STRING	1	☑		
C04288	Status	20287	4F3F	S	1	INTEGER_32	1	☑		
L_SdSpeedFilter										
C04290	Compensating speed	20285	4F3D	S	1	INTEGER_32	10000	☑	☑	
C04291	Filter time constant	20284	4F3C	S	1	INTEGER_32	1000	☑	☑	
C04292	Speed unit	20283	4F3B	S	1	VISIBLE_STRING	1	☑		
C04293	Reference speed	20282	4F3A	S	1	INTEGER_32	10000	☑		
C04294	Speed at the output	20281	4F39	S	1	INTEGER_32	10000	☑		
C04295	Speed at the input	20280	4F38	S	1	INTEGER_32	10000	☑		
C04299	Filter is enabled	20276	4F34	S	1	UNSIGNED_8	1	☑		
L_SdSpeedSet										
C04200	Fixed setpoint x	20375	4F97	A	15	INTEGER_32	100	☑	☑	
C04201	Active setpoint	20374	4F96	S	1	UNSIGNED_32	1	☑		
C04202	Basic acceleration time	20373	4F95	S	1	INTEGER_32	1000	☑	☑	
C04203	Basic deceleration time	20372	4F94	S	1	INTEGER_32	1000	☑	☑	
C04204	Basic S-ramp time	20371	4F93	S	1	INTEGER_32	1000	☑	☑	
C04207	Polarity bStop input	20368	4F90	S	1	UNSIGNED_8	1	☑	☑	
C04208	Tolerance: Setpoint reached	20367	4F8F	S	1	INTEGER_32	100	☑	☑	
C04212	Acceleration time x	20363	4F8B	A	15	INTEGER_32	1000	☑	☑	
C04213	Deceleration time x	20362	4F8A	A	15	INTEGER_32	1000	☑	☑	
C04214	S-ramp time x	20361	4F89	A	15	INTEGER_32	1000	☑	☑	
C04215	Active ramp parameters	20360	4F88	S	1	UNSIGNED_32	1	☑		
C04220	Profile mode	20355	4F83	S	1	UNSIGNED_32	1	☑		
C04221	Setpoint at the input	20354	4F82	S	1	INTEGER_32	100	☑		
C04222	Target setpoint	20353	4F81	S	1	INTEGER_32	100	☑		
C04223	Setpoint at the output	20352	4F80	S	1	INTEGER_32	100	☑		
C04224	Target setpoint reached	20351	4F7F	S	1	UNSIGNED_8	1	☑		
C04225	Load reference speed	20350	4F7E	S	1	INTEGER_32	1	☑		
C04226	Setpoint speed - load	20349	4F7D	S	1	INTEGER_32	1	☑		
C04227	Reference speed	20348	4F7C	S	1	INTEGER_32	10000	☑		
C04228	Setpoint speed	20347	4F7B	S	1	INTEGER_32	10000	☑		
C04229	Speed unit	20346	4F7A	S	1	VISIBLE_STRING	1	☑		
C04237	External setpoint	20338	4F72	S	1	INTEGER_32	100	☑		
C04238	Current motor speed	20337	4F71	S	1	INTEGER_32	100	☑		
C04239	Status: Digital inputs	20336	4F70	A	13	UNSIGNED_8	1	☑		
L_SdSpeedToUnit										
C04316	Unit	20259	4F23	S	1	VISIBLE_STRING	1	☑		
L_SdSwitchPoint										
C04395	Output x: Status	20180	4ED4	A	8	UNSIGNED_8	1	☑		
C04396	Position unit	20179	4ED3	S	1	VISIBLE_STRING	1	☑		
C04397	Output x: Switch-on position	20178	4ED2	A	8	INTEGER_32	10000	☑	☑	
C04398	Output x: Switch-off position	20177	4ED1	A	8	INTEGER_32	10000	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C04399	Enable output x	20176	4ED0	A	8	UNSIGNED_8	1	☑	☑	
L_SdUnitToAcc										
C04318	Unit	20257	4F21	S	1	VISIBLE_STRING	1	☑		
L_SdUnitToPos										
C04310	Unit	20265	4F29	S	1	VISIBLE_STRING	1	☑		
L_SdUnitToSpeed										
C04314	Unit	20261	4F25	S	1	VISIBLE_STRING	1	☑		
L_TbCompare										
C04085	Comparison operation	20490	500A	S	1	UNSIGNED_32	1	☑	☑	
L_TbCompare_n										
C04080	Comparison operation	20495	500F	S	1	UNSIGNED_32	1	☑	☑	
C04081	Tolerance	20494	500E	S	1	INTEGER_32	100	☑	☑	
C04082	Hysteresis	20493	500D	S	1	INTEGER_32	100	☑	☑	
L_TbCount										
C04005	Function: |Counter| >= |Ref.|	20570	505A	S	1	UNSIGNED_32	1	☑	☑	
L_TbCurve										
C04100	Characteristic shape	20475	4FFB	S	1	UNSIGNED_32	1	☑	☑	
C04101	x coordinates	20474	4FFA	A	9	INTEGER_32	100	☑	☑	
C04102	y coordinates	20473	4FF9	A	9	INTEGER_32	100	☑	☑	
L_TbCurveAny										
C04045	Characteristic shape	20530	5032	E	1	UNSIGNED_32	1	☑	☑	
C04046	x coordinates	20529	5031	A	127	INTEGER_32	100	☑	☑	
C04047	y coordinates	20528	5030	A	127	INTEGER_32	100	☑	☑	
L_TbDeadband										
C04065	Gain	20510	501E	S	1	INTEGER_32	10000	☑	☑	
L_TbDeadband_n										
C04060	Dead band	20515	5023	S	1	INTEGER_32	100	☑	☑	
C04061	Gain	20514	5022	S	1	INTEGER_32	10000	☑	☑	
L_TbDelay										
C04010	Select delayed edge	20565	5055	S	1	UNSIGNED_32	1	☑	☑	
C04011	Delay time	20564	5054	S	1	UNSIGNED_32	1000	☑	☑	
L_TbDifferentiate										
C04050	Filter time constant	20525	502D	S	1	INTEGER_32	1000	☑	☑	
L_TbGainLim										
C04030	Gain	20545	5041	S	1	INTEGER_32	10000	☑	☑	
L_TbLimit_n										
C04070	Upper limitation	20505	5019	S	1	INTEGER_32	100	☑	☑	
C04071	Lower limitation	20504	5018	S	1	INTEGER_32	100	☑	☑	
L_TbMaskOut										
C04110	Mean value - blocking zones	20465	4FF1	A	4	INTEGER_32	100	☑	☑	
C04111	Width - blocking zones	20464	4FF0	A	4	INTEGER_32	100	☑	☑	
L_TbNormalize										
C04035	Reference value	20540	503C	S	1	INTEGER_32	1	☑	☑	
L_TbOscillator										
C04020	Time: Output = HIGH	20555	504B	S	1	UNSIGNED_32	1000	☑	☑	
C04021	Time: Output = LOW	20554	504A	S	1	UNSIGNED_32	1000	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
L_TbPIController										
C04090	Gain	20485	5005	S	1	INTEGER_32	10000	☑	☑	
C04091	Reset time	20484	5004	S	1	INTEGER_32	1000	☑	☑	
C04095	Upper limitation	20480	5000	S	1	INTEGER_32	100	☑	☑	
C04096	Lower limitation	20479	4FFF	S	1	INTEGER_32	100	☑	☑	
L_TbPT1Filter										
C04040	Filter time constant	20535	5037	S	1	INTEGER_32	1000	☑	☑	
L_TbRateAction										
C04055	Time constant	20520	5028	S	1	INTEGER_32	1000	☑	☑	
L_TbTransition										
C04000	Select trigger edge	20575	505F	S	1	UNSIGNED_32	1	☑	☑	
C04001	Time: Output = HIGH	20574	505E	S	1	UNSIGNED_32	1000	☑	☑	

Index

Symbols

"Flying saw" application 186
"Unit" wildcard 19

Numbers

64 bit encoder signal evaluation 302

A

ABS
  See L_TbAbs 548
Absolute value 548
Absolute value encoder 298
Acceleration conversion 452, 537
Acceleration table 392
ADD
  See L_TbAdd 549
  See L_TbAddLim 550
  See L_TbAddSubLim 551
Addition 549, 550, 551
AND
  See L_Tb5And 540
  See L_TbAnd 552
AND with five inputs 540
AND with two inputs 552
ANEG
  See L_TbNeg 603
Application notes 15
ARIT
  See L_TbAdd 549
  See L_TbAddLim 550
  See L_TbAddSubLim 551
  See L_TbDiv 585
  See L_TbDiv_n 586
  See L_TbMul 598
  See L_TbMul_n 599
  See L_TbMulLim 601
  See L_TbSub 623
  See L_TbSubLim 624
ASW
  See L_Tb8Select 545
  See L_TbSelect 620

B

Behaviour of the outputs 84
Bit combination 209, 210, 211, 217, 218, 219, 243, 244, 245
Bit demultiplexer 212, 222, 247
Bit multiplexer 205, 206, 208
Bit operation 226, 227, 228, 234, 235, 236, 237, 238, 239

C

C4000 625
C4001 625
C4005 564
C4010 579
C4011 579
C4020 611
C4021 611
C4030 590
C4035 608
C4040 615
C4045 570
C4046 571
C4047 571
C4050 582
C4055 616
C4060 577
C4061 577
C4065 575
C4070 594
C4071 594
C4080 558
C4081 558
C4082 558
C4085 553
C4090 614
C4091 614
C4095 614
C4096 614
C4100 565
C4101 566
C4102 566
C4110 596
C4111 596
C4190 200
C4191 202
C4192 204
C4193 203
C4194 201
C4200 525
C4201 525
C4202 526
C4203 526
C4204 526
C4207 526
C4208 526
C4212 526
C4213 526
C4214 526
C4215 526
C4220 526
C4221 526
C4222 526
C4223 526
C4224 526
C4225 526
C4226 526

C4227 526
C4228 527
C4229 527
C4237 527
C4238 527
C4239 527
C4240 511
C4241 511
C4242 511
C4243 511
C4244 511
C4245 511
C4246 511
C4248 511
C4249 511
C4251 511
C4252 511
C4258 511
C4259 511
C4271 458
C4272 458
C4273 458
C4276 518
C4277 518
C4278 518
C4281 459
C4282 459
C4283 459
C4286 520
C4287 520
C4288 520
C4290 523
C4291 523
C4292 523
C4293 523
C4294 523
C4295 523
C4299 523
C4300 487
C4301 487
C4302 487
C4303 487
C4304 487
C4305 487
C4306 487
C4307 487
C4310 538
C4312 495
C4314 539
C4316 531
C4318 537
C4319 452
C4320 453
C4321 453
C4322 453
C4323 453
C4325 468
C4326 468
C4327 468
C4329 468
C4330 508
C4331 508
C4332 508
C4333 508
C4336 508
C4340 505
C4341 505
C4342 505
C4343 505
C4350 462
C4352 462
C4353 462
C4354 462
C4357 462
C4358 462
C4359 463
C4360 489
C4361 489
C4362 489
C4363 489
C4368 489
C4369 490
C4375 498
C4376 498
C4377 498
C4378 498
C4379 498
C4380 498
C4381 498
C4382 498
C4383 498
C4384 498
C4385 498
C4386 499
C4388 499
C4395 533
C4396 533
C4397 533
C4398 533
C4399 533
C4400 483
C4401 483
C4402 484
C4403 484
C4404 484
C4410 457

C4411 457
C4412 457
C4413 457
C4414 457
C4416 457
C4418 457
C4420 457
C4422 457
C4424 457
C4425 457
C4430 475
C4431 475
C4432 475
C4433 475
C4434 475
C4435 475
C4436 475
C4437 476
C4438 476
C4441 471
C4442 471
C4443 471
C4500 436
C4501 436
C4502 436
C4503 436
C4504 436
C4505 436
C4510 441
C4511 441
C4512 441
C4513 441
C4514 441
C4515 441
C4516 441
C4520 442
C4521 442
C4522 442
C4523 442
C4530 443
C4531 443
C4532 443
C4540 444
C4541 444
C4542 444
C4543 444
C4544 444
C4545 444
C4546 444
C4547 444
C4548 444
C4549 444
C4550 444
C4551 444
C4552 444
C4553 444
C4554 444
C4555 444
C4556 444
C4557 444
C4558 444
C4559 444
C4570 446
C4571 446
C4572 446
C4573 446
C4580 447
C4581 447
C4582 447
C4590 448
C4591 448
C4592 448
C4593 448
C4594 448
C4600 449
C4601 449
C4602 449
C4700 402
C4701 403
C4702 403
C4703 403
C4705 403
C4706 403
C4707 404
C4708 404
C4710 404
C4711 404
C4712 404
C4713 404
C4714 405
C4715 405
C4716 405
C4717 405
C4718 405
C4719 405
C4720 405
C4721 406
C4723 406
C4724 406
C4750 421
C4751 421
C4752 422
C4753 422
C4754 422
C4755 422
C4756 422

C4757 422
C4758 422
C4759 422
C4760 423
C4761 423
C4763 423
C4764 423
C4765 423
C4767 423
C4768 423
C4830 631
C4831 631
C4832 631
C4838 631
C4839 631
C4840 635
C4845 639
C4846 639
C4847 639
C4848 639
C4849 639
C4850 639
C4851 639
C4852 639
C4853 639
C4854 639
C4855 639
C4856 639
C4859 639
C4870 645
C4871 645
C4872 645
C4873 645
C4879 645
C4880 650
C4885 653
C5000 320
C5001 320
C5007 320
C5008 320
C5009 320
C5010 375
C5011 376
C5012 376
C5013 376
C5014 376
C5015 376
C5016 376
C5017 376
C5025 376
C5026 376
C5027 376
C5028 376
C5034 357
C5035 357
C5036 357
C5037 357
C5041 357
C5042 357
C5043 357
C5048 357
C5049 357
C5050 343
C5051 344
C5052 344
C5053 344
C5054 344
C5056 344, 345
C5065 345
C5067 345
C5069 345
C5070 311
C5071 311
C5072 311
C5074 311
C5076 311
C5077 311
C5078 311
C5079 311
C5080 313
C5081 313
C5082 313
C5083 313
C5085 313
C5087 313
C5088 314
C5089 314
C5090 366
C5091 366
C5092 366
C5093 366
C5094 367
C5095 367
C5096 367
C5097 367
C5098 367
C5099 367
C5100 381
C5101 381
C5102 381
C5103 381
C5104 381
C5105 381
C5114 381
C5117 381
C5118 381

C5122 338
C5123 338
C5124 338
C5125 338
C5126 338
C5127 338
C5128 338
C5129 338
C5142 362
C5143 362
C5144 362
C5147 362
C5148 362
C5149 362
C5150 362
C5151 362
C5152 362
C5153 362
C5154 362
C5155 362
C5160 316
C5161 316
C5162 316
C5163 316
C5165 316
C5169 316
C5175 335
C5176 335
C5178 336
C5179 336
C5180 325
C5181 325
C5183 326
C5184 326
C5186 326
C5187 326
C5188 326
C5189 326
C5270 299
C5271 299
C5272 300
C5273 300
C5274 300
C5275 300
C5277 300
C5280 300
C5281 300
C5282 300
C5283 300
C5284 300
C5289 300
C5290 305
C5291 305
C5293 305
C5294 306
C5295 306
C5296 306
C5297 306
C5298 306
C5299 306
C5330 292
C5331 292
C5332 293
C5333 293
C5335 293
C5336 293
C5337 293
C5338 293
C5339 294
C5400 156
C5401 156
C5406 156
C5407 156
C5408 156
C5409 156
C5410 156
C5411 156
C5412 156
C5413 157
C5414 157
C5415 157
C5419 157
C5440 182
C5447 182
C5448 182
C5449 182
C5450 182
C5451 182
C5455 182
C5456 182
C5457 182
C5458 182
C5459 182
C5460 176
C5463 176
C5464 176
C5465 176
C5466 176
C5467 176
C5468 176
C5469 176
C5480 137
C5481 137
C5482 137
C5486 137
C5487 137

C5488 137
C5489 137
C5500 164
C5501 164
C5502 164
C5505 164
C5506 164
C5507 164
C5510 164
C5511 165
C5514 165
C5515 165
C5516 165
C5517 165
C5518 165
C5519 165
C5520 139
C5521 139
C5522 139
C5523 139
C5524 139
C5525 140
C5526 140
C5527 140
C5528 140
C5529 140
C5530 159
C5534 160
C5536 160
C5537 160
C5539 160
C5540 150
C5541 150
C5547 150
C5548 150
C5549 150
C5560 169
C5561 169
C5564 169
C5565 170
C5566 170
C5567 170
C5569 170
C5570 173
C5571 173
C5572 173
C5574 173
C5575 173
C5576 173
C5577 174
C5579 174
C5600 187
C5601 187
C5602 187
C5603 187
C5605 187
C5606 188
C5607 188
C5608 188
C5609 188
C5900 252
C5901 252
C5902 252
C5910 278
C5911 278
C5912 278
C5913 278
C5914 279
C5915 279
C5920 288
C5921 288
C5922 288
C5923 289
C5924 289
C5925 289
Cam 532
Cam data via PDO 168
Cams 532
Characteristic function 565, 570
Characteristic interpolation 149
Clutch (path-controlled) 135
Clutch (time-controlled) 290
CMP
  See L_TbCompare 553
  See L_TbCompare_n 558
Comparator 553, 558
Comparing applications 93
Comparing FB interconnections 93
Comparing interconnections 93
Comparison 553, 558
Compressing (position) 175, 180
Configuring system events 84
Configuring the task properties 83
Contactor data selection 172
Control signal from the safety module 280
Control word for limiter 282
CONV
  See L_TbGainLim 590
  See L_TbNormalize 608
Conventions used 14
Conventions used for variable identifiers 27
Copying complete interconnection 97
Copying elements 88
Copying FB interconnection 97
Copying interconnection 97
Copying interconnection elements 88

Copying objects 88
Counter 563
CURVE
  See L_TbCurve 565
  See L_TbCurveAny 570
Curve interpolation 162

D
Data type check 30
Data type entry 27
DB
  See L_TbDeadband 575
  See L_TbDeadband_n 577
Dead band with gain 575
Dead band with gain (scaled) 577
Dead time compensation 453
Delay 579, 615
Diameter calculation 629
Differentiation 454
Differentiation (cyclic) 318
Differentiator with low-pass filter 582
DIGDEL
  See L_TbDelay 579
Division 585, 586, 600
Download to RAM 86
DT1-1
  See L_TbDifferentiate 582

E
Edge evaluation 625
E-mail to Lenze 681
Encoder signal conversion 298
Error tripping 251, 253
Exception handling 84
Extrapolation 319

F
FB Editor 37
FBE_insert_user_code 62
FCNT
  See L_TbCount 563
Feedback to Lenze 681
FIXSET
  See L_SdSpeedSet 524
FLIP
  See L_TbFlipFlopD 587
  See L_TbFlipFlopRS 589
Flipflop 587
Flying saw 189
Following error monitoring 335
Friction compensation 637
Function block 17
Function block editor 37
Function library 17

G
Gain 590
General delay 581

I
Identification of the moment of inertia 644
Implicit type conversion 30
Incremental encoder 298
Insert options for copied elements 90
Inserting complete interconnection from reference project 98
Inserting copied elements 90
Inserting FB interconnection from reference project 98
Integration 322, 460, 461, 467, 591

L
L_CamClutchPos 135
L_CamContactor 138
L_CamCurve 149
L_CamGetAxisData 155
L_CamPosMarker 158
L_CamProfiler 162
L_CamSetContDataPDO 168
L_CamSetContDataSDO 172
L_CamStretchAbs 175
L_CamStretchFeed 180
L_CamSyncIn 186
L_Dc2BytesToWord 197
L_Dc2WordsToDWord 198
L_Dc4BytesToDWord 199
L_DcBitShift 200
L_DcBitShiftByte 201
L_DcBitShiftDWord 202
L_DcBitShiftInt 203
L_DcBitShiftWord 204
L_DcBitsToByte 205
L_DcBitsToDWord 206
L_DcBitsToWord 208
L_DcByteBitand 209
L_DcByteBitor 210
L_DcByteBitxor 211
L_DcByteToBits 212
L_DcByteToInt 213
L_DcByteToWord 214
L_DcDIntToDWord 215
L_DcDIntToInt 216
L_DcDWordBitand 217
L_DcDWordBitor 218
L_DcDWordBitxor 219
L_DcDWordTo2Words 220
L_DcDWordTo4Bytes 221
L_DcDWordToBits 222
L_DcDWordToDInt 224
L_DcDWordToWord 225

L_DcGetBitOfByte 226
L_DcGetBitOfDWord 227
L_DcGetBitOfWord 228
L_DcIntToDInt 229
L_DcIntToWord 230
L_DcNorm_aToNorm_n 231
L_DcNorm_nToNorm_a 232
L_DcNorm_nToSpeed_s 233
L_DcResetBitOfByte 234
L_DcResetBitOfDWord 235
L_DcResetBitOfWord 236
L_DcSetBitOfByte 237
L_DcSetBitOfDWord 238
L_DcSetBitOfWord 239
L_DcSpeed_sToNorm_n 240
L_DcSpeed_sToSpeed_v 241
L_DcSpeed_vToSpeed_s 242
L_DcWordBitand 243
L_DcWordBitor 244
L_DcWordBitxor 245
L_DcWordTo2Bytes 246
L_DcWordToBits 247
L_DcWordToByte 248
L_DcWordToDWord 249
L_DcWordToInt 250
L_DevApplErr 251
L_DevApplErrFix 253
L_DevBrakeStateDecoder 255
L_DevDriveInterfaceStateDecoder 257
L_DevHomingStateDecoder 259
L_DevLimiterStateDecoder 261
L_DevManualJogStateDecoder 263
L_DevParReadFix 265
L_DevParWriteFix 271
L_DevPositionerStateDecoder 275
L_DevReadParDInt 277
L_DevSMControlDecoder 280
L_DevSMControlEncoder 282
L_DevSMStateDecoder 284
L_DevSMStateDecoderIO 286
L_DevWriteParDInt 287
L_EsClutchPos 290
L_EsEncoderConv 298
L_EsStretchIntegrate 304
L_LdAddOffsetCyclic 308
L_LdClutchAxisP 309
L_LdClutchV 312
L_LdConvAxisV 315
L_LdDifferentiateCyclic 318
L_LdExtrapolate 319
L_LdIntegrateCyclic 322
L_LdLinearCoupling 324
L_LdMarkSync 327
L_LdMonitFollowError 335
L_LdMPot 337
L_LdPosCtrlLin 340
L_LdSetAxisVelocity 354
L_LdStateDecoder 358
L_LdSyncOperation 363
L_LdToolControl 360
L_LdVirtualMasterP 373
L_LdVirtualMasterV 379
L_LdZeroDetect 384
L_PosDecoderStatePositioner 385
L_PosDecoderStateSequencer 387
L_PosGetProfile 388
L_PosGetProfileData 389
L_PosGetTableAcc 392
L_PosGetTableJerk 394
L_PosGetTablePos 395
L_PosGetTableSpeed 396
L_PosPositionerInterface 397
L_PosPositionerTable 400
L_PosProfileInterface 414
L_PosProfileTable 420
L_PosSequencer 430
L_PosSequencerStateDecoder 451
L_SdAccToUnit 452
L_SdDelayComp 453
L_SdDifferentiate 454
L_SdFactor 455
L_SdGetAxisData 456
L_SdGetPosition 458
L_SdGetSpeed 459
L_SdIntegrate 460
L_SdIntegrateAxis 461
L_SdIntegrateLimit 467
L_SdInterExtrapolateAny 471
L_SdInterExtrapolatePosition 474
L_SdInterpolate 483
L_SdLimitSpeed 486
L_SdMotorPot 488
L_SdPosToUnit 495
L_SdProcessController 496
L_SdRampGenerator 504
L_SdRampGeneratorAny 507
L_SdRuntimeComp 509
L_SdSetAxisData 510
L_SdSetPosition 518
L_SdSetSpeed 520
L_SdSpeedFilter 522
L_SdSpeedSet 524
L_SdSpeedToUnit 531
L_SdSwitchPoint 532
L_SdTouchProbe 535
L_SdUnitToAcc 537

L_SdUnitToPos 538
L_SdUnitToSpeed 539
L_Tb5And 540
L_Tb5Nand 541
L_Tb5Nor 542
L_Tb5Or 543
L_Tb5Xor 544
L_Tb8Select 545
L_Tb8SelectByte 546
L_Tb8SelectWord 547
L_TbAbs 548
L_TbAdd 549
L_TbAddLim 550
L_TbAddSubLim 551
L_TbAnd 552
L_TbCompare 553
L_TbCompare_n 558
L_TbCount 563
L_TbCurve 565
L_TbCurveAny 570
L_TbDeadband 575
L_TbDeadband_n 577
L_TbDelay 579
L_TbDifferentiate 582
L_TbDiv 585
L_TbDiv_n 586
L_TbFlipFlopD 587
L_TbFlipFlopRS 589
L_TbGainLim 590
L_TbIntegrate 591
L_TbLimit 593
L_TbLimit_n 594
L_TbLimitInt 595
L_TbMaskOut 596
L_TbMul 598
L_TbMul_n 599
L_TbMulDivLim 600
L_TbMulLim 601
L_TbNand 602
L_TbNeg 603
L_TbNegInt 604
L_TbNegSel 605
L_TbNegSelInt 606
L_TbNor 607
L_TbNormalize 608
L_TbNot 609
L_TbOr 610
L_TbOscillator 611
L_TbPIController 613
L_TbPT1Filter 615
L_TbRateAction 616
L_TbSampleHold 618
L_TbSampleHoldWord 619
L_TbSelect 620
L_TbSelectByte 621
L_TbSelectWord 622
L_TbSub 623
L_TbSubLim 624
L_TbTransition 625
L_TbXor 628
L_WndCalcDiameter 629
L_WndCalcLength 634
L_WndFrictionCompensation 637
L_WndIdentMInertia 644
L_WndStopCtrl 649
L_WndTensionCurve 652
Layout of the safety information 15
Layout of the safety instructions 15
Length calculation 634
LIM
  See L_TbLimit 593
  See L_TbLimit_n 594
Limit 593, 594, 595

M

Machine parameters 510
Mark synchronisation 327
Master value adjustment 337
Master value adjustment via position 340
Master value connection 324, 363
Master value processing 354
Monitoring of the task runtime 83
Motor potentiometer 488
Multiplication 598, 599, 600, 601

N

NAND with five inputs 541
NAND with two inputs 602
Negation 603, 604, 605, 606, 609
NOR with five inputs 542
NOR with two inputs 607
NOT
  See L_TbNot 609
NSET
  See L_SdSpeedSet 524

O

OFF-delay 580
Offset addition 308
ON-delay 580
OR
  See L_Tb5Or 543
  See L_TbOr 610
OR with five inputs 543
OR with two inputs 610
Overchange 411, 418

Oversynchronous clutch-in 186

P

Parameterisable function blocks 18
PCTRL
See L_TbPIController 613
PHADD
See L_TbAdd 549
See L_TbAddLim 550
See L_TbAddSubLim 551
PHCMP
See L_TbCompare 553
See L_TbCompare_n 558
PHDIFF
See L_TbDifferentiate 582
PHDIV
See L_TbDiv 585
See L_TbDiv_n 586
PHINT
See L_SdIntegrate 460
See L_SdIntegrateLimit 467
PI controller 613
PID controller with limitation 496
Position conversion 458, 495, 518, 538
Position table 395
Positioning interface 397
Positioning with final speed 411, 418
Printing the interconnection 92
Process controller 496, 613
Profile data interface 414
Profile data output 389
Profile data record management 400, 420
Profile data tables 388
PT1-1
See L_TbPT1Filter 615

R

Ramp function generator 524
Ramp function generator with S ramp 504, 507
Ramp generator 524
Rate action 616
Read code 265
Read parameter 265
Reading a code 277
Reading out cams from cam data 138
Reading out machine parameters from axis data 456
Reading out machine parameters from cam data 155
Reading out position marks from cam data 158
Reading parameters 277
Real unit 19
Rectangular signal generator 611
Reference project 98
RFG
See L_SdRampGenerator 504
See L_SdSpeedSet 524
RS flipflop (status-controlled) 589
Runtime compensation 509

S

S&H
See L_TbSampleHold 618
Safety instructions 15
Sample & Hold 618, 619
Scaling of physical units 29
Selector 620, 621, 622
Selector for data type "BYTE" 546
Selector for data type "DINT" 545
Selector for data type "WORD" 547
Sequence control 430
Setpoint conditioning 360
Setpoint ramp generator 524
Shift operation 200, 201, 202, 203, 204
Signal converter 231, 232, 233, 240, 241, 242
Signal interpolation 471, 474, 483
Signal scaling 608
Signal type entry 28
Speed conversion 459, 520, 531, 539
Speed limitation 486
Speed ratio 315
Speed signal delay 522
Speed table 396
S-ramp time table 394
SRFG
See L_SdRampGenerator 504
See L_SdSpeedSet 524
Status signals (drive interface) 257
Status signals (LineDrive-FBs) 358
Status signals (LS_Brake) 255
Status signals (LS_Homing) 259
Status signals (LS_Limiter) 261
Status signals (LS_ManualJog) 263
Status signals (positioning) 275, 385
Status signals (sequence control) 387, 451
Status signals from the safety module 284, 286
Stop controller 649
Stretch factor (electronic gearbox) 455
Stretching & compressing the master value 304
Stretching (position) 175, 180
Subtraction 551, 623, 624
Switching points 532
Symbolic icons 17
Synchronous clutch-in 186
Synchronous stretching & compression 304
Synchronous/oversynchronous clutch-in 186

T

Table of attributes 656
Task selection 43
Task types 32
TaskOverrun 84
Tensile force characteristic 652
Test in the device 86
Testing the interconnection in the device 86
Touch probe 461
Touch probe evaluation 535
TRANS
  See L_TbTransition 625
Type conversion (implicit) 30
Type converter 197, 198, 199, 213, 214, 215, 216, 220, 221, 224, 225, 229, 230, 246, 248, 249, 250

U

Unit (wildcard) 19
Up/downcounter 563

V

Velocity changeover 411, 418
Virtual clutch (electronic gearbox) 312
Virtual clutch (synchronism) 309
Virtual master (electronic gearbox) 379
Virtual master (synchronism) 373

W

Watchdog 83
Write code 271
Write parameter 271
Writing to a code 287
Writing to parameters 287

X

XOR with five inputs 544
XOR with two inputs 628

Z

Zero crossing detection 384
Zone masking 596

FEEDBACK

Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

If you have suggestions for improvement, please e-mail us to:

feedback-docu@Lenze.de

Thank you for your support.

Your Lenze documentation team

CE

Lenze Automation GmbH
Hans-Lenze-Str. 1
D-31855 Aerzen
Germany
+49 (0)51 54 / 82-0
+49 (0)51 54 / 82-28 00
Lenze@Lenze.de
www.Lenze.com

Service
Lenze Service GmbH
Breslauer Straße 3
D-32699 Extertal
Germany
00 80 00 / 24 4 68 77 (24 h helpline)
+49 (0)51 54 / 82-11 12
Service@Lenze.de

Lenze