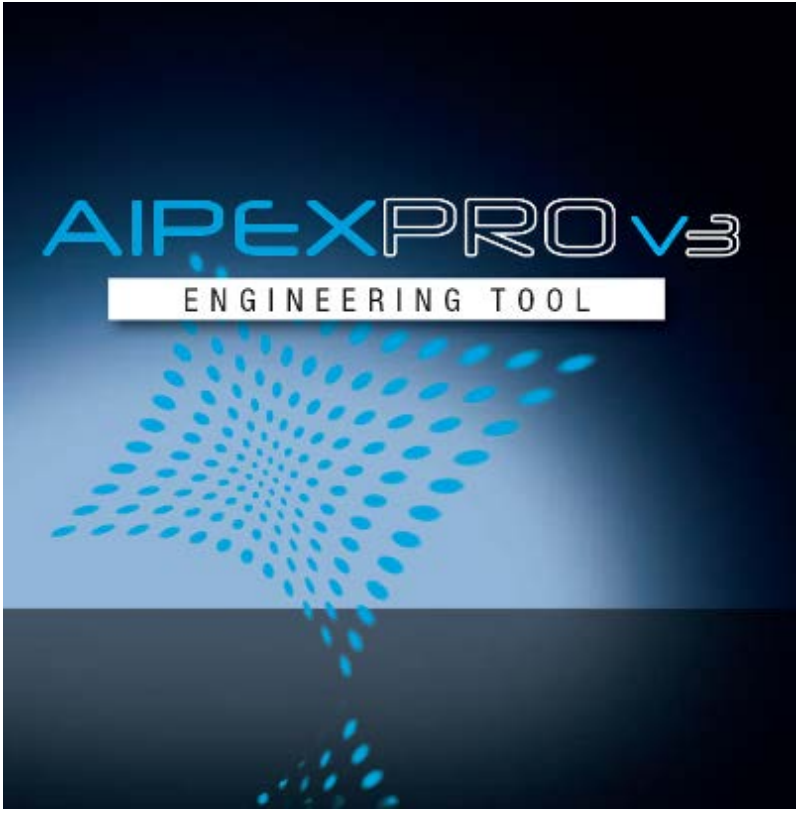


AMKmotion
Softwarebeschreibung
AIPEX PRO V3
AFL AMKmotion Funktionsbibliotheken

Version: 2023/09
Teile-Nr.: 204979
"Original Dokumentation"

AMKmotion
MEMBER OF THE ARBURG FAMILY

Impressum

Name: PDK_204979_Software_AIPEX_PRO_V3_de

Version:

Version: 2023/09	
Änderung	**Kurzzeichen**
• neues AMKmotion Design	LeS

Bisherige Version: 2019/45

Produktstand:

Produkt	Firmware Version (Teile-Nr.)
PC Software	AIPEX PRO V3.04 2018/47 (207211)
Software AFL für CODESYS V2	AFL V3 Version 3.14 2015/41 (206005)
Software AFL für CODESYS V3	AFL V4 Version 3.5.5.0 2015/41 (206004)

Schutzvermerk: © AMKmotion GmbH + Co KG

Weitergabe sowie Vervielfältigung dieser Unterlagen, Verwertung und Mitteilung ihres Inhalts wird nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zum Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Vorbehalt: Änderungen im Inhalt der Dokumentation und Liefermöglichkeit der Produkte sind vorbehalten.

Herausgeber: AMKmotion GmbH + Co KG
Gaußstraße 37-39
73230 Kirchheim unter Teck
Germany

Phone +49 7021 50 05-0
Fax +49 7021 50 05-176

E-Mail info@amk-motion.com

Registergericht: AG Stuttgart, HRA 230681, Kirchheim unter Teck,
Ust.-Id.-Nr.: DE 145 912 804
Komplementär: AMKmotion Verwaltungsgesellschaft mbH, HRB 774646

Service: Phone +49 7021 50 05-190, Fax -193

Zur schnellen und zuverlässigen Behebung der Störung tragen Sie bei, wenn Sie unseren Service informieren über:

- die Typenschildangaben der Geräte
- die Softwareversion
- die Gerätekonstellation und die Applikation
- die Art der Störung, vermutete Ausfallursache
- die Diagnosemeldungen (Fehlernummern)

E-Mail service@amk-motion.com

Internetadresse: www.amk-motion.com

Inhalt

Impressum **2**
1 Zu dieser Dokumentation **13**
1.1 Dokumentstruktur 13
1.2 Aufbewahrung 13
1.3 Zielgruppe 14
1.4 Zweck 14
1.5 Darstellungskonventionen 14
1.6 Weiterführende Dokumente 14
2 Zu Ihrer Sicherheit **16**
2.1 Grundlegende Hinweise für Ihrer Sicherheit 16
2.2 Darstellung der Sicherheitshinweise 16
2.3 Gefahrenklassen 16
2.4 Bestimmungsgemäße Verwendung 17
3 AIPEX PRO **18**
3.1 Installationsanleitung 18
3.1.1 Voraussetzungen 18
3.1.2 Wichtige Hinweise 18
3.1.3 Installation 18
3.2 Servicepack 35
3.3 Programmübersicht 36
3.3.1 Programmübersicht - Symbole und Icons 37
3.3.2 Programmübersicht - Anzeigefilter 39
3.3.3 Programmübersicht - Teilansichten 40
3.4 Kommunikation PC - AMK Gerät 43
3.4.1 Kommunikationsserver 43
3.4.1.1 Kommunikationsserver Einstellungen 44
3.4.2 CAN Schnittstelle 46
3.4.2.1 CAN Kommunikationseinstellungen 46
3.4.3 COM Schnittstelle 48
3.4.3.1 COM Kommunikationseinstellungen 49
3.4.3.2 Serielles AMK Verbindungskabel RS232 (PC-AMK) 51
3.4.3.3 USB-RS232 Konverter 52
3.4.4 EtherCAT Schnittstelle 54
3.4.4.1 EtherCAT Kommunikationseinstellungen 55
3.4.5 Ethernet Schnittstelle 57
3.4.5.1 Ethernet Kommunikationseinstellungen 58
3.4.6 SERCOS III Schnittstelle 66
3.4.6.1 SERCOS III Kommunikationseinstellungen 67
3.4.7 USB Schnittstelle 69
3.4.7.1 USB Kommunikationseinstellungen 69
3.4.8 AIPEX PRO add in gateway for TwinCAT V2 69
3.4.9 Kommunikation testen 69
3.5 Reiter 70
3.5.1 Konfiguration 70
3.5.1.1 Konfiguration - Anleitung 71
3.5.1.1.1 Eigenschaften - Gerät 71
3.5.1.1.2 Interface 73
3.5.1.1.3 PLC 73
3.5.1.1.4 Motor 75
3.5.2 Parameter 75
3.5.2.1 Parameter - Anleitung 76

3.5.2.1.1 Übersicht 76
3.5.2.1.2 Darstellung und Strukturierung von Listen 77
3.5.2.1.3 Kontextmenü Parameter 78
3.5.2.1.4 ID Eigenschaften 80
3.5.2.2 Parameter - Auswahl 81
3.5.2.3 Parameter - Online Hilfe 82
3.5.2.4 Parameter - Kommentare 83
3.5.2.5 Parameter - Suchfunktion 84
3.5.3 Nachrichten 84
3.5.3.1 Nachrichten - Anzeigefilter 85
3.5.3.2 Nachrichten - Allgemeine Optionen 86
3.5.3.2.1 ACC-Bus 86
3.5.3.2.2 EtherCAT 87
3.5.3.3 Nachrichten - Spezifische Knotenoptionen 88
3.5.3.3.1 ACC-Bus 88
3.5.3.3.2 EtherCAT 89
3.5.3.4 Nachrichten - PDO Optionen 90
3.5.3.5 Nachrichten - PLC Optionen 91
3.5.4 Scope 92
3.5.4.1 Scope - Anleitung 92
3.5.4.2 Scope - Maximale Aufzeichnungszeit 95
3.5.4.3 Interpretation der Lagesollwerte und Lageistwerte 96
3.5.5 Diagnose 96
3.6 Menüleiste 97
3.6.1 Projekt 97
3.6.2 Online 100
3.6.3 Ansicht 102
3.6.4 Extras 103
3.6.4.1 Extras - Optionen 104
3.6.4.1.1 Basiseinstellungen 105
3.6.4.1.2 PC Kommunikation 106
3.6.4.1.3 Konfiguration erstellen 107
3.6.4.1.4 Datenaktualisierung 110
3.6.4.2 Extras - Projekt Einstellungen 112
3.6.4.2.1 Basiseinstellungen 112
3.6.4.2.2 Konfiguration erstellen (projektspezifisch) 113
3.6.5 Inbetriebnahme 116
3.6.5.1 Inbetriebnahme - Monitor 120
3.6.5.2 Testgenerator 121
3.6.6 Konfiguration 125
3.6.7 '?' - Dokumentationen 126
3.6.8 Direktmode 127
3.6.8.1 Verbindung im Direktmode aktivieren 129
3.6.8.2 Parameter 131
3.6.8.3 Temporäre Parameter 132
3.6.8.4 Diagnose 133
3.6.8.5 Kommunikation 134
3.6.8.6 PLC 136
3.6.8.7 Spezialfunktionen 137
3.6.8.8 Systeminfo 139
3.6.8.9 Funktionale Sicherheit 140
3.7 Funktionen 141
3.7.1 Aktualisierung der Geräte-Firmware 141
3.7.2 Daten vergleichen 145

3.7.3 Diagnose mit AIPEX PRO 147
3.7.4 Einloggen 149
3.7.5 Gerätedaten auslesen und speichern 151
3.7.6 Geräteposition im EtherCAT Bus verschieben 154
3.7.7 Importieren einer externen Gerätebeschreibung 154
3.7.8 Modular Device konfigurieren 157
3.7.9 Oszilloskop konfigurieren 162
3.7.10 Testgenerator 165
3.7.11 Überspielen eines Offline Projekts in ein Gerät 169
4 AFL (AMK Function Libraries) 171
4.1 AmkAfl Standardbausteine 171
4.1.1 Einführung AFL Standardbausteine für CODESYS V3 171
4.1.1.1 Übersicht der AMK Bibliotheken 171
4.1.1.2 Verwendung der AFL Standardbausteine 172
4.1.1.3 Programmaufbau mit CODESYS V3 und AFL Standardbausteinen 174
4.1.1.4 Visualisierung Standardbausteine 178
4.1.1.5 Übersicht Standardbausteine und Achsstrukturen 178
4.1.1.5.1 Erweiterte Standard Achsen 179
4.1.1.5.2 Strukturen 181
4.1.2 Einführung AFL Standardbausteine für CODESYS V2 182
4.1.2.1 Übersicht der AMK Bibliotheken 182
4.1.2.2 Verwendung der AFL Standardbausteine 183
4.1.2.3 Programmaufbau mit CODESYS V2 und AFL Standardbausteinen 184
4.1.2.4 Visualisierung Standardbausteine 187
4.1.2.5 Übersicht Standardbausteine und Achsstrukturen 187
4.1.2.5.1 Erweiterte Standard Achsen 188
4.1.2.5.2 Strukturen 190
4.1.3 Beschreibung der Strukturen (ST) 191
4.1.3.1 ST_CONTROL (ST) 191
4.1.3.1.1 ST_PLC_STATUSBITS (ST) 191
4.1.3.1.2 ST_PLC_CONTROLBITS (ST) 192
4.1.3.1.3 arstDiagnosis (ARR_ST) 192
4.1.3.2 ST_AXIS_DRIVE (ST) 193
4.1.3.2.1 ST_ACTUAL_VALUES (ST) 193
4.1.3.2.2 ST_STATUSBITS (ST) 194
4.1.3.2.3 ST_SETPOINTS (ST) 195
4.1.3.2.4 ST_CONTROLBITS (ST) 195
4.1.3.2.5 ST_IDS (ST) 196
4.1.3.2.6 ST_PLC_ID_LIST (ST) (vorläufig) 196
4.1.3.2.7 arstDiagnosis (ARR_ST) 196
4.1.4 Beschreibung der Standardbausteine (FB) 197
4.1.4.1 Einspeisung und PLC 197
4.1.4.1.1 STANDARD_CONTROL (FB) 197
4.1.4.1.2 STANDARD_CONTROL_iSA (FB) 199
4.1.4.1.3 STANDARD_KE (FB) 201
4.1.4.2 Standard Achse 202
4.1.4.2.1 STANDARD_AXIS (FB) (STANDARD_AXIS_KWZ, STANDARD_AXIS_IDT4, STANDARD_AXIS_ihX) 202
4.1.4.2.2 Modifizierte 'Standard Achsen' für ihX Antriebe 209
4.1.4.2.3 STANDARD_AXIS_POSITION_INTERPOSED (FB) 210
4.1.4.2.4 STANDARD_AXIS_POSITION_TO_THE_MARK (FB) 219
4.1.4.2.5 STANDARD_AXIS_PRINTMARK_INSETTER_CONT (FB) 228
4.1.4.2.6 STANDARD_AXIS_PRINTMARK_INSETTER_INTERVAL (FB) 240
4.1.4.2.7 STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_CONT (FB) 252

4.1.4.2.8 STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_DISCONT (FB) 264
4.1.4.2.9 STANDARD_FOLLOW_AXIS_GEAR (FB) 276
4.1.4.2.10 STANDARD_FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2 (FB) 284
4.1.4.2.11 STANDARD_FOLLOW_AXIS_SWITCHING_TABLE_WITH_OUTGEAR (FB) 293
4.1.4.2.12 STANDARD_FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2 (FB) 301
4.1.4.2.13 STANDARD_FOLLOW_AXIS_TABLE_CONTINUOUS (FB) 311
4.1.4.2.14 STANDARD_FOLLOW_TABLE_ENGAGE_DISENG_AXIS (FB) 319
4.1.4.2.15 STANDARD_FOLLOW_TABLE_START_AUTOSTOP_AXIS (FB) 328
4.1.4.2.16 STANDARD_WINDER_DANCER_AXIS (FB) 337
4.1.4.2.17 STANDARD_WINDER_TORQUE_AXIS (FB) 347
4.2 AMK AFL Applikationsbausteine 366
4.2.1 Bibliotheksbeschreibung AmkAfl.lib 366
4.2.1.1 05_StandardLevel 366
4.2.1.1.1 01_Comparison Functions 366
4.2.1.1.1.1 COMPARE_2VALUES (FB) 366
4.2.1.1.1.2 fboCompare_ActVal_GT_Threshold (FB) 367
4.2.1.1.2 03_Position 367
4.2.1.1.2.1 Support 367
4.2.1.1.2.2 POSITION_ABSOLUT (FB) 372
4.2.1.1.2.3 POSITION_ABSOLUT_VAJ (FB) 375
4.2.1.1.2.4 POSITION_CHECK_LIMIT_SWITCH (FB) 377
4.2.1.1.2.5 POSITION_HOMING_FIXED_STOP (FB) 378
4.2.1.1.2.6 POSITION_JERK_1 (FB) 380
4.2.1.1.2.7 POSITION_RELATIV (FB) 383
4.2.1.1.2.8 POSITION_RELATIV_INTERPOSED (FB) 386
4.2.1.1.2.9 POSITION_RELATIV_RETRIGGER_VAJ (FB) 388
4.2.1.1.2.10 POSITION_RELATIV_VAJ (FB) 390
4.2.1.1.3 04_Manual 393
4.2.1.1.3.1 MANUAL_DETERMINE_INERTIA (FB) 393
4.2.1.1.3.2 MANUAL_JOG (FB) 396
4.2.1.1.3.3 MANUAL_JOG_VAJ (FB) 398
4.2.1.1.3.4 MANUAL_STEP (FB) 400
4.2.1.1.3.5 MANUAL_VELOCITY (FB) 401
4.2.1.1.3.6 MANUAL_VELOCITY_1 (FB) 403
4.2.1.1.4 06_Follow 404
4.2.1.1.4.1 FOLLOW_AXIS_CAM_CONT (FB) 404
4.2.1.1.4.2 FOLLOW_AXIS_CONTINUOUS_TAB1_TAB2_OUTGEAR32 (FB) 408
4.2.1.1.4.3 FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2 (FB) 410
4.2.1.1.4.4 FOLLOW_AXIS_CRANK_DRIVE_EMULATION (FB) 412
4.2.1.1.4.5 FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2 (FB) 414
4.2.1.1.4.6 FOLLOW_AXIS_TABLE_CONTINUOUS (FB) 416
4.2.1.1.4.7 FOLLOW_AXIS_TABLE_ENGAGE_DISENGAGE (FB) 418
4.2.1.1.4.8 FOLLOW_AXIS_TABLE_START_AUTOSTOP (FB) 420
4.2.1.1.4.9 FOLLOW_AXIS_TABLE_START_AUTOSTOP_1 (FB) 422
4.2.1.1.4.10 FOLLOW_RATIO_INC_64 (FB) 424
4.2.1.1.5 07_Convert/Calc 426
4.2.1.1.5.1 CALC_PROF_2_FORMAT_POLY (FB) 426
4.2.1.1.5.2 CONVERT_ANALOG_TO_DIGITAL (FB) 427
4.2.1.1.5.3 CONVERT_DIGITAL_TO_ANALOG (FB) 428
4.2.1.1.5.4 CONVERT_VELOCITY (FB) 429
4.2.1.1.5.5 fbrInertiaHollowCylinder (F) 431
4.2.1.1.5.6 fbrInertiaPlainCylinder (F) 431
4.2.1.1.5.7 fbyChar_To_ASCIICode (F) 432
4.2.1.1.5.8 fdi_Convert_InkPerSec_to_0_0001rpm (F) 432

4.2.1.1.5.9 fdiConvert_TwoInt_to_Dint (F) 433
4.2.1.1.5.10 fdiSwap_AllBytes_of_Dint (F) 433
4.2.1.1.5.11 fdiSwap_HLBytes_of_Dint (F) 433
4.2.1.1.5.12 fdiSwap_HLWords_of_Dint (F) 434
4.2.1.1.5.13 fiConvert_Bcd_to_Int (F) 434
4.2.1.1.5.14 fiConvertTorqueNmToPercent (F) 434
4.2.1.1.5.15 fiSwap_Bytes_of_Int (F) 435
4.2.1.1.5.16 frGradToRad (F) 435
4.2.1.1.5.17 frRadToGrad (F) 435
4.2.1.1.5.18 fstrASCIICode_To_Char (F) 436
4.2.1.1.5.19 fstrConvert_BoolToString (F) 436
4.2.1.1.5.20 fstrConvert_DintToString (F) 436
4.2.1.1.5.21 fudCalc_Ramp_Acceleration (F) 437
4.2.1.1.5.22 fudiConvert_Bcd_to_Udint (F) 437
4.2.1.1.5.23 fuiConvert_Bcd_to_Uint (F) 438
4.2.1.1.5.24 udfConvert_Acceleration (F) 438
4.2.1.1.6 08_Feedforward 439
4.2.1.1.6.1 FEED_FOR_DEADTIME (FB) 439
4.2.1.1.6.2 FEED_FOR_TORQUE (FB) 440
4.2.1.1.6.3 FEED_FOR_VELOCITY (FB) 441
4.2.1.1.7 09_Filter 442
4.2.1.1.7.1 FILTER_BLOCKING_NEGATIVE_COUNT (FB) 442
4.2.1.1.7.2 FILTER_BLOCKING_POSITIVE_COUNT (FB) 442
4.2.1.1.7.3 FILTER_HYSTERESIS (FB) 443
4.2.1.1.7.4 FILTER_MEAN_VALUE_DINT_MAX20 (FB) 444
4.2.1.1.7.5 FILTER_MEAN_VALUE_MAX20 (FB) 445
4.2.1.1.7.6 FILTER_PT1 (FB) 445
4.2.1.1.7.7 FILTER_RAMP_INCRDECR_TIMECONST (FB) 446
4.2.1.1.7.8 FILTER_RAMP_MINMAX_TIME (FB) 448
4.2.1.1.7.9 FILTER_SETPOINT (FB) 449
4.2.1.1.8 10_Buffer 450
4.2.1.1.8.1 BUFFER_FIFO_DINT_MAX_50 (FB) 450
4.2.1.1.8.2 BUFFER_FIFO_INT_MAX50 (FB) 451
4.2.1.1.8.3 BUFFER_SHIFT_REG_INT_MAX10 (FB) 452
4.2.1.1.9 11_Modulo 453
4.2.1.1.9.1 fdiAddModulo (F) 453
4.2.1.1.9.2 fdiModulo_DiffModulo (F) 454
4.2.1.1.9.3 fdiModulo_Modulo (F) 455
4.2.1.1.9.4 MODULO_COUNT (FB) 455
4.2.1.1.9.5 MODULO_DEMODULO_COUNT (FB) 456
4.2.1.1.9.6 MODULO_MODCOUNT_GEAR (FB) 457
4.2.1.1.9.7 MODULO_MODULO1_TO_MODULO2_COUNT (FB) 458
4.2.1.1.10 12_Increment 459
4.2.1.1.10.1 INCREMENT_DIFF_STAGE (FB) 459
4.2.1.1.10.2 INCREMENT_DIFF_STAGE_I_TO_DI (FB) 460
4.2.1.1.10.3 INCREMENT_MULTIPLEX_2_INT_TO_1_DINT (FB) 461
4.2.1.1.10.4 INCREMENT_MULTIPLEX_2_TO_1_DINT (FB) 462
4.2.1.1.10.5 INCREMENT_MULTIPLEX_4_TO_1_DINT (FB) 463
4.2.1.1.10.6 INCREMENT_OFFSET (FB) 464
4.2.1.1.10.7 INCREMENT_SUMMATION (FB) 464
4.2.1.1.11 13_Timing control 465
4.2.1.1.11.1 TIMING_CONTROL_BLINK (FB) 465
4.2.1.1.11.2 TIMING_CONTROL_CLOCK (FB) 466
4.2.1.1.12 14_I_O_Handling 467

4.2.1.1.12.1 I_O_DEBOUNCED_SWITCH (FB) 467
4.2.1.1.12.2 I_O_EDGE_CONTROL (FB) 468
4.2.1.1.12.3 I_O_FLANK_TRIGGER (FB) 468
4.2.1.1.12.4 I_O_FLANK_TRIGGER_ACKNOWLEDGEMENT (FB) 469
4.2.1.1.12.5 I_O_LATCH_BUTTON (FB) 470
4.2.1.1.12.6 I_O_PLUS_MINUS_BUTTON (FB) 471
4.2.1.1.13 17_Date 472
4.2.1.1.13.1 DATE_COMPAR_BIT (FB) 472
4.2.1.1.13.2 DATE_CONVERT_DATA_TIME (FB) 473
4.2.1.1.13.3 DATE_PAYDAY (FB) 474
4.2.1.1.13.4 fboDate_Leap_Year (F) 475
4.2.1.1.13.5 fdiDate_Diff_Time (F) 475
4.2.1.1.13.6 fiDate_Get_Day_of_the_Week (F) 476
4.2.1.1.13.7 fiDate_Get_Day_of_the_Year (F) 477
4.2.1.1.13.8 fusiDate_Get_Calendar_Week_EU (F) 477
4.2.1.1.13.9 fusiDate_Get_Calendar_Week_USA (F) 478
4.2.1.1.13.10 fusiDate_Get_count_Days_in_Month (F) 479
4.2.1.1.13.11 fusiDate_Get_Eastern (F) 480
4.2.1.1.14 18_File 481
4.2.1.1.14.1 Support 481
4.2.1.1.14.2 FILE_COPY (FB) 483
4.2.1.1.14.3 FILE_DELETE (FB) 485
4.2.1.1.14.4 FILE_FIND (FB) 486
4.2.1.1.14.5 FILE_READ (FB) 487
4.2.1.1.14.6 FILE_WRITE (FB) 489
4.2.1.1.15 19_Stringfunctions 490
4.2.1.1.15.1 fiString_Find (F) 490
4.2.1.1.15.2 fiString_Len (F) 491
4.2.1.1.15.3 fstrConCat (F) 491
4.2.1.1.15.4 fstrString_Append 492
4.2.1.1.15.5 fstrString_Append1 (F) 493
4.2.1.1.15.6 fstrString_Compare (F) 493
4.2.1.1.15.7 fstrString_Left (F) 494
4.2.1.1.15.8 fstrString_Mid (F) 495
4.2.1.1.15.9 fstrString_Right (F) 495
4.2.1.1.16 20_Virtual Master 496
4.2.1.1.16.1 VIRTUAL_RFGEN (FB) 496
4.2.1.1.16.2 VIRTUAL_VGEN (FB) 497
4.2.1.1.16.3 VIRTUAL_VGEN_A (FB) 499
4.2.1.1.16.4 VIRTUAL_VGEN_AJ (FB) 500
4.2.1.1.17 21_Data 502
4.2.1.1.17.1 DATA_TABLE_SORT (FB) 502
4.2.1.2 06_TechnologyLevel 503
4.2.1.2.1 01_Emergency stop 503
4.2.1.2.1.1 EMERGENCY_BREAK_POSITION (FB) 503
4.2.1.2.1.2 EMERGENCY_BREAK_VELOCITY (FB) 504
4.2.1.2.2 02_Print marks 505
4.2.1.2.2.1 PRINT_MARK_CONTROL (FB) 505
4.2.1.2.2.2 PRINT_MARK_CORRECT (FB) 510
4.2.1.2.2.3 PRINT_MARK_CORROUT_CONTROL (FB) 511
4.2.1.2.2.4 PRINT_MARK_DETECT (FB) 512
4.2.1.2.2.5 PRINT_MARK_INSETTER_CONT (FB) 515
4.2.1.2.2.6 PRINT_MARK_INSETTER_INTERVAL (FB) 521
4.2.1.2.2.7 PRINT_MARK_IPO (FB) 528

4.2.1.2.2.8 PRINT_MARK_POSITION_TO_THE_MARK (FB) 529
4.2.1.2.2.9 PRINT_MARK_REGISTER_CONTROL_CONTINUOUS (FB) 533
4.2.1.2.2.10 PRINT_MARK_REGISTER_CONTROL_DISCONT (FB) 539
4.2.1.2.2.11 PRINT_MARK_TABLE_INVERTER (FB) 546
4.2.1.2.3 03_Winder 548
4.2.1.2.3.1 WINDER_ANALOG_TO_PULSE (FB) 548
4.2.1.2.3.2 WINDER_DANCER_BIN_OUT (FB) 549
4.2.1.2.3.3 WINDER_CHARACTERISTICS (FB) 551
4.2.1.2.3.4 WINDER_CALC_DM (FB) 553
4.2.1.2.3.5 WINDER_CALC_WEB_SPEED (FB) 555
4.2.1.2.3.6 WINDER_CD_REEL_TO_MOTOR_SPEED (FB) 556
4.2.1.2.3.7 WINDER_CALC_FRICTION (FB) 558
4.2.1.2.3.8 WINDER_CALC_INERTIA (FB) 559
4.2.1.2.3.9 WINDER_CALC_TORQUE (FB) 560
4.2.1.2.3.10 WINDER_CT_REEL_TO_MOTOR_SPEED (FB) 562
4.2.1.2.3.11 WINDER_PID_CTRL (FB) 563
4.2.1.2.3.12 WINDER_SCALE_FIND_INERTIA_SPEED (FB) 567
4.2.1.2.3.13 WINDER_TIGHTWND_REACHED (FB) 567
4.2.1.2.3.14 WINDER_DANCER_CONTROL (FB) 568
4.2.1.2.3.15 WINDER_TORQUE_CONTROL (FB) 578
4.2.1.2.4 04_Recipe 587
4.2.1.2.4.1 prgLoad_Save_Recipe_To_File (PRG) 587
4.2.1.2.5 05_Alarm 590
4.2.1.2.5.1 Alarmtabelle 590
4.2.1.2.5.2 FILE_MANAGER 591
4.2.1.2.5.3 Support 594
4.2.1.2.5.4 FB_READ_ERR_HISTORIE (FB) 595
4.2.1.2.5.5 fboClrErr (F) 597
4.2.1.2.5.6 fboSetErr (F) 598
4.2.1.2.5.7 fusiSetErrTabRow (F) 598
4.2.1.2.5.8 fusiSetFormat (F) 599
4.2.1.2.5.9 fusiSetHistTabRow (F) 599
4.2.1.2.5.10 PRG_AUTO_DEL (PRG) 600
4.2.1.2.5.11 PRG_ERR_HANDLER (PRG) 601
4.2.1.2.5.12 Visualisierung 602
4.2.1.2.6 06_Commands 604
4.2.1.2.6.1 CMD_HOME (FB) 604
4.2.1.2.7 07_NC Motion 605
4.2.1.2.7.1 NC_MOTION 605
4.2.1.2.7.2 FAST Modul 606
4.2.1.2.7.3 FIFO 608
4.2.1.2.7.4 Funktionen 610
4.2.1.2.7.5 Module 610
4.2.1.2.8 08_Sequence 624
4.2.1.2.8.1 SEQUENCE_STATE_CHECK (FB) 624
4.2.1.2.8.2 SEQUENCE_STATE_TIMEOUT (FB) 625
4.2.1.2.9 09_Parameter 626
4.2.1.2.9.1 SAVE_PARAMETER (FB) 626
4.2.1.2.9.2 RESTORE_PARAMETER (FB) 627
4.2.1.3 07_Visualization 629
4.2.1.3.1 NUMPAD (FB) 629
4.2.1.4 08_Devices 630
4.2.1.4.1 fstrGetControllerTyp (F) 630
4.2.1.5 Anhang 631

4.2.2 Bibliotheksbeschreibung AmkAflConfig.lib 632
4.2.2.1 02_Parameter Access 632
4.2.2.1.1 ID_READ_10IDS_DINT (FB) 632
4.2.2.1.2 ID_READ_5IDS_DINT (FB) 634
4.2.2.1.3 ID_READ_NODE_LIST (FB) 635
4.2.2.1.4 ID_WRITE_10IDS_DINT (FB) 636
4.2.2.1.5 ID_WRITE_5IDS_DINT (FB) 638
4.2.2.1.6 ID_WRITE_5IDS_TEMP_DINT (FB) 639
4.2.2.1.7 ID_WRITE_ID26 (FB) 641
4.2.2.2 03_Actual Level 642
4.2.2.2.1 01_Status 642
4.2.2.2.1.1 GET_STATUSBITS (FB) 642
4.2.2.2.2 02_Actual_Value 644
4.2.2.2.2.1 GET_RECTANGULAR_PULSE_INPUT (FB) 644
4.2.2.2.2.2 GET_STATUSBITS_ACTUAL_VALUE_GEAR (FB) 645
4.2.2.2.2.3 GET_STATUSBITS_ACTUAL_VALUE_GEAR_WITHOUT_RECT (FB) 647
4.2.2.3 04_Setpoint_Level 649
4.2.2.3.1 01_Setpoint 649
4.2.2.3.1.1 SET_SET_POINTS_AND_FEED_FORWARD_GEAR (FB) 649
4.2.2.3.1.2 SET_SET_POINTS_GEAR (FB) 652
4.2.2.4 06_Diagnose 654
4.2.2.4.1 DIAGNOSE_ERROR (FB) 654
4.2.2.4.2 DIAGNOSE_TRACE_CSV (FB) 655
4.2.2.4.3 DIAGNOSE_TRACE_VAL1_TO_VAL10 (FB) 656
4.2.2.5 old version 657
4.2.2.5.1 GET_ACTUAL_VALUE (FB) 657
4.2.2.5.2 GET_STATUSBITS_ACTUAL_VALUE (FB) 659
4.2.2.5.3 SET_CONTROLBITS_SETPOINT_SETVALUES (FB) 661
4.2.2.5.4 SET_SET_POINTS_AND_FEED_FORWARD (FB) 663
4.2.2.5.5 SET_SETPOINT_POSITION_ABSOLUTE (FB) 666
4.2.2.6 Support 667
4.2.2.6.1 CONTROL_UE_RF (FB) 667
4.2.2.6.2 RATIO_INC_ABS (FB) 668
4.3 FIRST STEPS AFL Standardbausteine 669
4.3.1 FIRST STEPS mit CODESYS V3 und AFL Standard Bausteine 669
4.3.1.1 Voraussetzungen 669
4.3.1.2 CODESYS Versionsauswahl 669
4.3.1.3 Grund Einstellungen AIPEX PRO 670
4.3.1.4 PLC Projekt erzeugen (Online Projekt) 672
4.3.1.5 Standard Funktionsbausteine importieren 681
4.3.1.6 Standard Control 683
4.3.1.7 Standard Einspeisung 687
4.3.1.8 Standard AXIS 690
4.3.1.9 PLC Programm und Nachrichtenkonfiguration erstellen 695
4.3.1.10 PLC Projekt testen 698
4.3.1.11 Visualisierung 701
4.3.1.11.1 Visualisierung Standardbausteine 703
4.3.1.12 Funktionen 706
4.3.1.12.1 I/O Zugriff 706
4.3.1.12.1.1 Externe I/O-Klemme (asynchron) 706
4.3.1.12.1.2 Externe I/O-Klemme (zyklisch) 710
4.3.1.12.1.3 Time Stamp EAs und XFC Klemmen 715
4.3.1.12.1.4 EA Klemme Steuerung (asynchon) 715
4.3.1.12.2 Erweiterte Variablenzugriffe 719

4.3.1.12.3 EtherCAT Querkommunikation zwischen zwei AMK Steuerungen 723
4.3.1.12.3.1 Vorbereitung 724
4.3.1.12.3.2 PLC Programm Master Steuerung 725
4.3.1.12.3.3 PLC Programm Slave Steuerung 733
4.3.1.12.4 Kurvenscheibe 737
4.3.1.12.5 PLC Projekt importieren 737
4.3.1.13 Generelle Informationen 741
4.3.1.13.1 Versionierung CODESYS V3 741
4.3.1.13.2 Automatische Buskonfiguration 743
4.3.1.13.2.1 Einschränkung der automatischen Buskonfiguration 747
4.3.1.13.3 Zielsystemauswahl für CODESYS V3 750
4.3.1.13.3.1 Auswahl AMK Zielsysteme 751
4.3.1.13.3.2 Geräteidentifizierung 752
4.3.1.13.3.3 AIPEX PRO CODESYS Projekte mit CODESYS Optionen 753
4.3.1.13.3.4 CODESYS Optionen deaktivieren 756
4.3.1.13.3.5 AMK Zielsystems Wechsel 758
4.3.1.13.4 Bibliotheksverwalter 760
4.3.1.13.5 Default Taskkonfiguration 763
4.3.1.13.6 Diagnoseinformationen 763
4.3.1.13.7 Steuerungskonfiguration 764
4.3.2 FIRST STEPS mit CODESYS V2 und AFL Standard Bausteine 766
4.3.2.1 Basis Einstellungen 766
4.3.2.1.1 Konfiguration erstellen (allgemein) 766
4.3.2.1.2 Basiseinstellungen 769
4.3.2.1.3 Konfiguration erstellen (projektspezifisch) 769
4.3.2.2 PLC Projekt erzeugen 771
4.3.2.3 Standard Einspeisung importieren 776
4.3.2.4 Standard Einspeisung instanzieren 777
4.3.2.5 Basis Steuerung importieren 780
4.3.2.6 Standard Steuerung importieren 781
4.3.2.7 Standard Steuerung instanzieren 782
4.3.2.8 Basis Achse importieren 785
4.3.2.8.1 Beschreibung Basis Achse 785
4.3.2.9 Standard Achse importieren 786
4.3.2.10 Achsstruktur deklarieren 787
4.3.2.11 Standard Achse instanzieren 789
4.3.2.12 Synchrone Aktionen 792
4.3.2.13 Istwerte lesen 793
4.3.2.14 Statusbits lesen 794
4.3.2.15 Diagnose lesen 795
4.3.2.16 Visualisierung 796
4.3.2.17 Programm übertragen 800
4.3.2.18 Schnell zum Ziel 804
4.3.2.18.1 Diagnose Array (arstDiagnosis) visualisieren 804
4.3.2.18.2 Externe asynchrone EA Klemme 809
4.3.2.18.3 Externe zyklische EA Klemme 814
4.3.2.18.4 Erweiterte Variablenzugriffe 820
4.3.2.18.5 Kurvenscheibe erstellen 823
4.3.2.18.6 PLC Projekt von AIPEXPRO importieren 830
4.3.2.18.7 Statusbits erweitern 836
4.3.2.18.8 Zugriff auf Steuerungs EAs 840
4.4 Visualisierung von AFL Bausteinen 843
4.4.1 CODESYS V3 843
4.4.2 CODESYS V2 850

Glossar **853**
Ihre Meinung zählt! **856**

1 Zu dieser Dokumentation

1.1 Dokumentstruktur

Thema	Kapitel	Kapitelnummer
Gültigkeit, Verwendung und Zielsetzung des Dokuments	Impressum	-
	Zu diesem Dokument	1
Zu Ihrer Sicherheit	Grundlegende Hinweise	2
	Darstellung der Sicherheitshinweise	
	Gefahrenklassen	
	Bestimmungsgemäße Verwendung	
AIPEX PRO Beschreibung der Software mit praktischen Anwendungsmöglichkeiten.	Installationsanleitung	3
	Servicepack	
	Programmübersicht	
	Kommunikation PC - AMK Gerät	
	Reiter (Konfiguration, Parameter, Nachrichten, Scope, Diagnose)	
	Menüleiste	
	Funktionen	
AFL Function Libraries Beinhaltet die Beschreibung der AFL Standard- und AFL Applikationsbausteine.	AFL AMK Function Libraries	4
AFL Standardbausteine Mit AFL Standardbausteinen wird die Funktionalität der AMK Geräte in ein PLC Projekt eingebunden. AFL Standardbausteinen bestehen aus AFL Applikationsbausteinen.	Einführung AFL Standardbausteine für CODESYS V3	4.1
	Einführung AFL Standardbausteine für CODESYS V2	
	Beschreibung der Strukturen (ST)	
	Beschreibung der Standardbausteine (FB)	
AFL Applikationsbausteine AMK stellt mit den AFL Applikationsbausteinen komplexere Funktionen zur Verfügung. AFL Applikationsbausteinen bestehen aus den PLC Grundbausteinen (separates Dokument).	Bibliotheksbeschreibung AmkAfl.lib	4.2
	Bibliotheksbeschreibung AmkAflConfig.lib	
FIRST STEPS mit AFL Standardbausteinen Programmierbeispiele für CODESYS V3 und CODESYS V2 mit AFL Standardbausteinen.	FIRST STEPS mit CODESYS V3 und AFL Standard Bausteine	4.3
	FIRST STEPS mit CODESYS V2 und AFL Standard Bausteine	
Vorgefertigte Visualisierungselemente für AFL Funktionsbausteine.	Visualisierung von AFL Bausteinen	4.4
Abkürzungen und Begriffserklärungen	Glossar	-

1.2 Aufbewahrung

Dieses Dokument muss ständig dort verfügbar und einsehbar sein, wo das Produkt im Einsatz ist. Wird das Produkt an einem anderen Ort eingesetzt oder wechselt den Besitzer, muss das Dokument mitgegeben werden.

1.3 Zielgruppe

Dieses Dokument muss von jeder Person gelesen, verstanden und beachtet werden, die berechtigt ist und beabsichtigt, eine der folgenden Arbeiten auszuführen:

- Projektieren
- Anschließen
- Parametrieren
- Inbetriebnehmen
- Service und Störungsbeseitigung
- Austausch
- Software Installieren
- PLC programmieren

1.4 Zweck

Dieses Dokument richtet sich an alle Personen, die mit dem Produkt umgehen, und informiert zu folgenden Themen:

- Inbetriebnahme und Betrieb
- Diagnose
- Softwareinstallation
- FIRST STEPS Anleitung

1.5 Darstellungskonventionen

Darstellung	Bedeutung
	Diese Textstelle verdient Ihre besondere Aufmerksamkeit!
	Das Symbol Hand zeigt in den Beispielen, auf welche Menüpunkte oder Tasten in einer Software geklickt werden muss.
RMT	Klicken Sie mit der 'Rechten Maus Taste'
0x	0x gefolgt von einer Hexadezimalzahl, z. B. 0x500A
'Namen'	In Hochkomma werden Namen dargestellt, z. B. Parameter, Variablen, usw.
'Text'	Menüpunkte und Tasten in einer Software oder Bedieneinheit, z. B.: Bestätigen Sie mit **'OK'** im Menü **'Optionen'**, um die Funktion 'PLC Programm löschen' aufzurufen
→	Ablauf einer Eingabe- / Bedienfolge z. B. **'Start'** → **'Alle Programme'** → **'Zubehör'** → **'Editor'** z. B. 0 → 1 Flanke
Siehe 'Kapitelname' auf Seite x	Ausführbarer Querverweis in elektronischen Ausgabemedien

1.6 Weiterführende Dokumente

Inbetriebnahmebeschreibung

AMK Teile-Nr.	Titel
204539	Erstinbetriebnahme KE/KW
204737	Erstinbetriebnahme Dezentrale Antriebe
204665	Anbindung von Fremdsteuerungen an AMK Antriebe

Softwarebeschreibung

AMK Teile-Nr.	Titel
205210	Softwarebeschreibung AmkLibraries
204072	AIPEX PRO add in gateway for TwinCAT

AMK Teile-Nr.	Titel
203771	Softwarebeschreibung ATF - AMK Tool Flasher
205171	AIPEX PRO Inbetriebnahme

2 Zu Ihrer Sicherheit

2.1 Grundlegende Hinweise für Ihrer Sicherheit

- Bei elektrischen Antriebssystemen treten prinzipbedingt Gefahren auf, die Tod oder schwere Körperverletzungen verursachen können:
  - Elektrische Gefährdung (z. B. Stromschlag durch Berühren elektrischer Anschlüsse)
  - Mechanische Gefährdung (z. B. Quetschen, Einziehen durch die Rotation der Motorwelle)
  - Thermische Gefährdung (z. B. Verbrennungen beim Berühren heißer Oberflächen)
- Die Gefahren treten insbesondere bei der Inbetriebnahme, während des Betriebes und im Service- oder Wartungsfall auf.
- Sicherheitshinweise in der Dokumentation und auf dem Produkt warnen vor den Gefahren.
- Die Sicherheitshinweise müssen vor der Installation und Produktverwendung gelesen und verstanden worden sein. In den produktbegleitenden Dokumenten weisen handlungsbezogene Warnhinweise auf direkt bevorstehende Gefahren hin und müssen unmittelbaren Einfluss auf die Handlung des Anwenders haben.
- AMKmotion Produkte müssen im Originalzustand belassen werden, d.h. an der Hardware darf keine bauliche Veränderung vorgenommen werden und Software darf nicht dekompiliert und der Quellcode geändert werden.
- Beschädigte oder fehlerhafte Produkte dürfen nicht eingebaut oder in Betrieb genommen werden.
- Anlagen, in die AMKmotion Produkte eingebaut werden, dürfen erst in Betrieb genommen werden (Aufnahme der bestimmungsgemäßen Verwendung), wenn festgestellt ist, dass alle dafür relevanten Normen, Gesetze und Richtlinien eingehalten sind, z. B. Niederspannungsrichtline, EMV-Richtlinie und Maschinenrichtlinie und möglicherweise weitere Produktnormen. Die Verantwortung dabei hat der Anlagenbauer.
- Die Geräte müssen wie in den Gerätebeschreibungen beschrieben montiert, angeschlossen und betrieben werden. Die technischen Daten und die geforderten Umgebungsbedingungen sind zu jeder Zeit einzuhalten.

2.2 Darstellung der Sicherheitshinweise

Alle Sicherheitshinweise sind wie folgt aufgebaut:

⚠SIGNALWORT	
	Art und Quelle der Gefahr Folge(n) bei Nichtbeachtung **Gegenmaßnahmen:** • ...

2.3 Gefahrenklassen

Sicherheits- und Warnhinweise sind in verschiedene Gefahrenklassen (nach ANSI Z535) abgestuft. Die Gefahrenklasse definiert das potentielle Schadensrisiko bei Nichtbeachten des Sicherheitshinweises und ist durch ein einzelnes Signalwort beschrieben. Das Signalwort wird von einem Warnsymbol (ISO 3864, DIN EN ISO 7010) begleitet. In Übereinstimmung mit ANSI Z535 werden folgende Signalworte zur Einstufung der Gefahrenklasse verwendet:

Warnsymbol und Signalwort	Gefahrenklasse und Bedeutung
⚠ GEFAHR	GEFAHR kennzeichnet eine Gefährdung, die Tod oder schwere Körperverletzungen zur Folge **hat**, wenn der Sicherheitshinweis nicht beachtet wird.
⚠ WARNUNG	WARNUNG kennzeichnet eine Gefährdung, die Tod oder schwere Körperverletzungen zur Folge **haben kann**, wenn der Sicherheitshinweis nicht beachtet wird.
⚠ VORSICHT	VORSICHT kennzeichnet eine Gefährdung, die leichte oder mittlere Körperverletzungen zur Folge haben kann, wenn der Sicherheitshinweis nicht beachtet wird.
HINWEIS	HINWEIS kennzeichnet mögliche Sachschäden, wenn der Hinweis nicht beachtet wird.

2.4 Bestimmungsgemäße Verwendung

Die AMK Software 'AIPEX PRO V3' unterstützt die Inbetriebnahme von AMK Antriebssystemen. Angeschlossene AMK Antriebssysteme mit elektronischem Typenschild werden automatisch erkannt und identifiziert. Dabei werden alle wichtigen Daten der Geräte angezeigt. Mit der AMK Software 'AIPEX PRO V3' und der integrierten 'CODESYS V3' Software können die AMK Antriebssysteme konfiguriert, optimiert, diagnostiziert, programmiert und in betrieb genommen werden.

3 AIPEX PRO

3.1 Installationsanleitung

3.1.1 Voraussetzungen

Empfohlene PC Voraussetzungen

- Prozessor: Pentium V, Centrino > 3,0 GHz, Pentium M > 1,5 GHz
- Festplatten-Speicher: mind. 1 GB
- Ram-Speicher: mind. 1024 MB

Unterstützte Betriebssysteme

- Windows 10
- Windows 8 und 8.1
- Windows 7
- Windows XP mit SP3 (SP1 muss vor SP3 installiert werden)

Benötigte .NET Versionen

SafePMT benötigt Microsoft .NET Framework 3.5
CODESYS V3 benötigt Microsoft .NET Framework 4.0

Für die Installation werden Administratorrechte benötigt.

3.1.2 Wichtige Hinweise

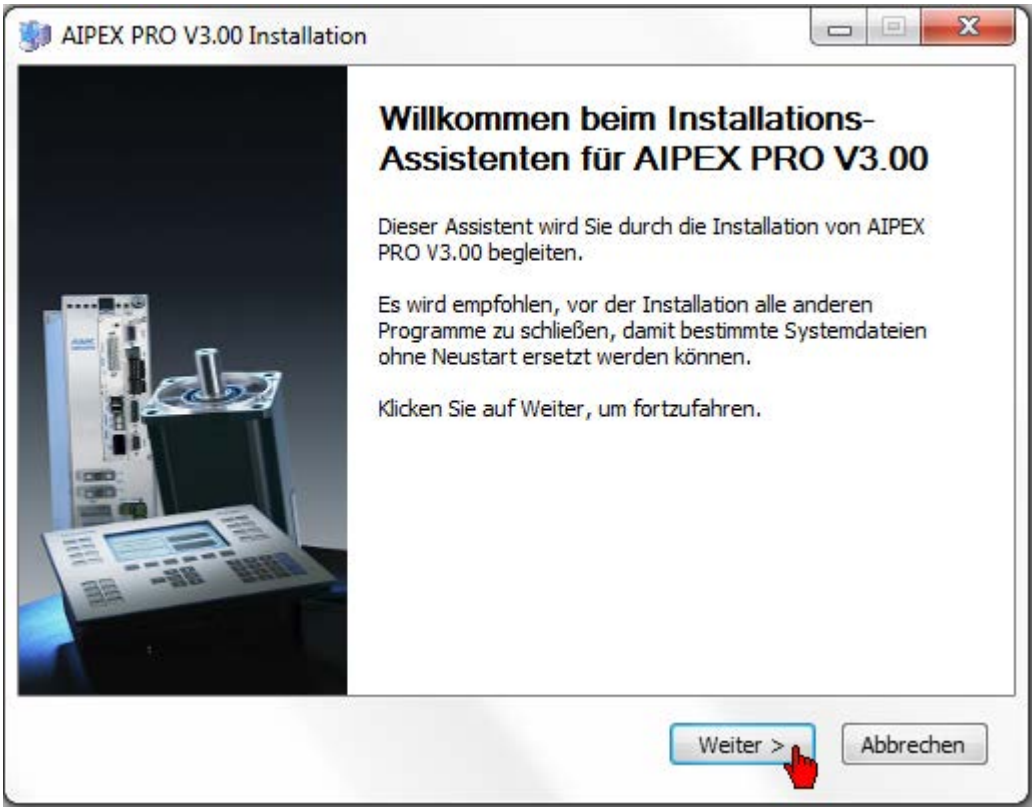
Vor der Installation dieser SafePMT Version müssen ältere Versionen manuell deinstalliert werden.

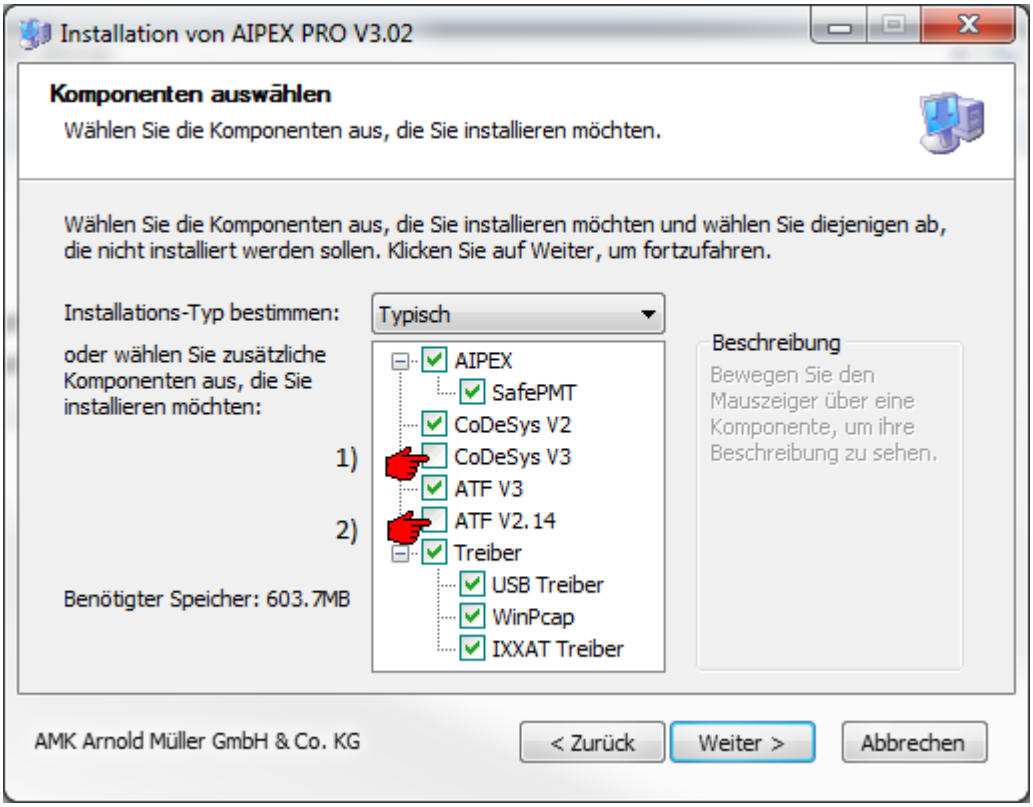
Bei der Deinstallation von AIPEX PRO werden alle installierten Treiber (WinPcap, USBCOM Treiber und IXXAT Treiber) vom PC entfernt.

3.1.3 Installation

Nach dem Einlegen der CD startet der 'AMK AIPEX PRO Installer' automatisch.
Ist der automatische Start deaktiviert, muss die Datei 'SetupAipexPro.exe' aufgerufen werden.

Folgen Sie den Anweisungen des 'AMK AIPEX PRO Installers'


AIPEX PRO V3.00 Installation
Willkommen beim Installations-Assistenten für AIPEX PRO V3.00
Dieser Assistent wird Sie durch die Installation von AIPEX PRO V3.00 begleiten.
Es wird empfohlen, vor der Installation alle anderen Programme zu schließen, damit bestimmte Systemdateien ohne Neustart ersetzt werden können.
Klicken Sie auf Weiter, um fortzufahren.
Weiter >
Abbrechen


Installation von AIPEX PRO V3.02
Komponenten auswählen
Wählen Sie die Komponenten aus, die Sie installieren möchten.
Wählen Sie die Komponenten aus, die Sie installieren möchten und wählen Sie diejenigen ab, die nicht installiert werden sollen. Klicken Sie auf Weiter, um fortzufahren.
Installations-Typ bestimmen:
Typisch
oder wählen Sie zusätzliche Komponenten aus, die Sie installieren möchten:
AIPEX
SafePMT
CoDeSys V2
1)
CoDeSys V3
ATF V3
2)
ATF V2.14
Treiber
USB Treiber
WinPcap
IXXAT Treiber
Beschreibung
Bewegen Sie den Mauszeiger über eine Komponente, um ihre Beschreibung zu sehen.
Benötigter Speicher: 603.7MB
AMK Arnold Müller GmbH & Co. KG
< Zurück
Weiter >
Abbrechen

1) Ab AIPEX PRO V3.02

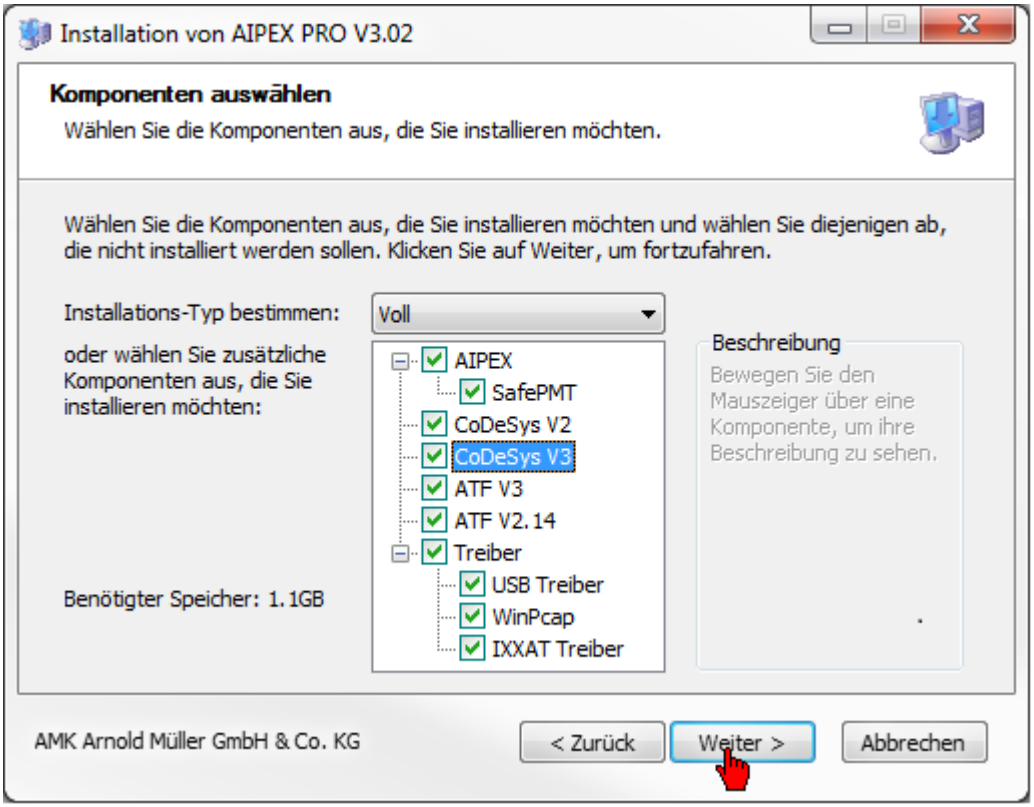
CODESYS V3 wird nicht automatisch installiert. Wählen Sie CODESYS V3 explizit an, wenn es installiert werden soll.

2) Software ATF Version 2.14

Für die Geräte IDT4 und KWZ muss die Software ATF V2.14 verwendet werden. Die Version V2.14 ist im Installationspaket von AIPEX PRO enthalten, muss bei der Installation aber explizit angewählt werden.

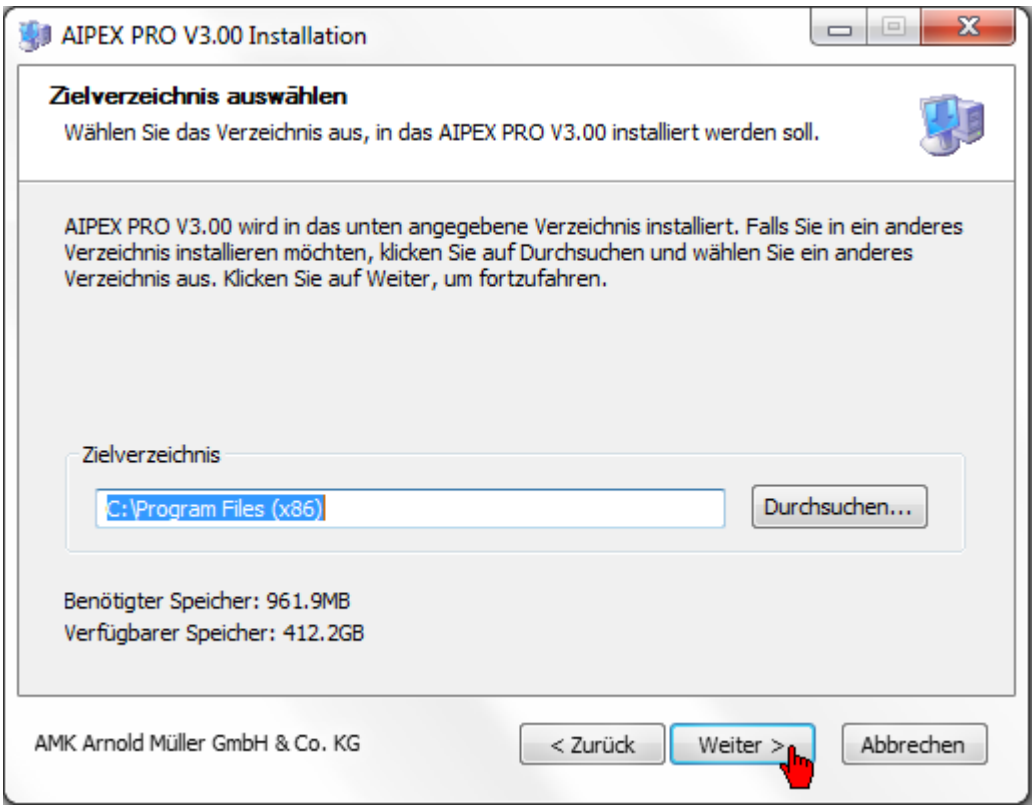
ATF V2.14 und 3.x können parallel installiert werden. Wenn ATF V2.14 gestartet ist, können nicht ATF ab V3.0 und andere Bestandteile von AIPEX PRO 3.x zur gleichen Zeit gestartet werden.

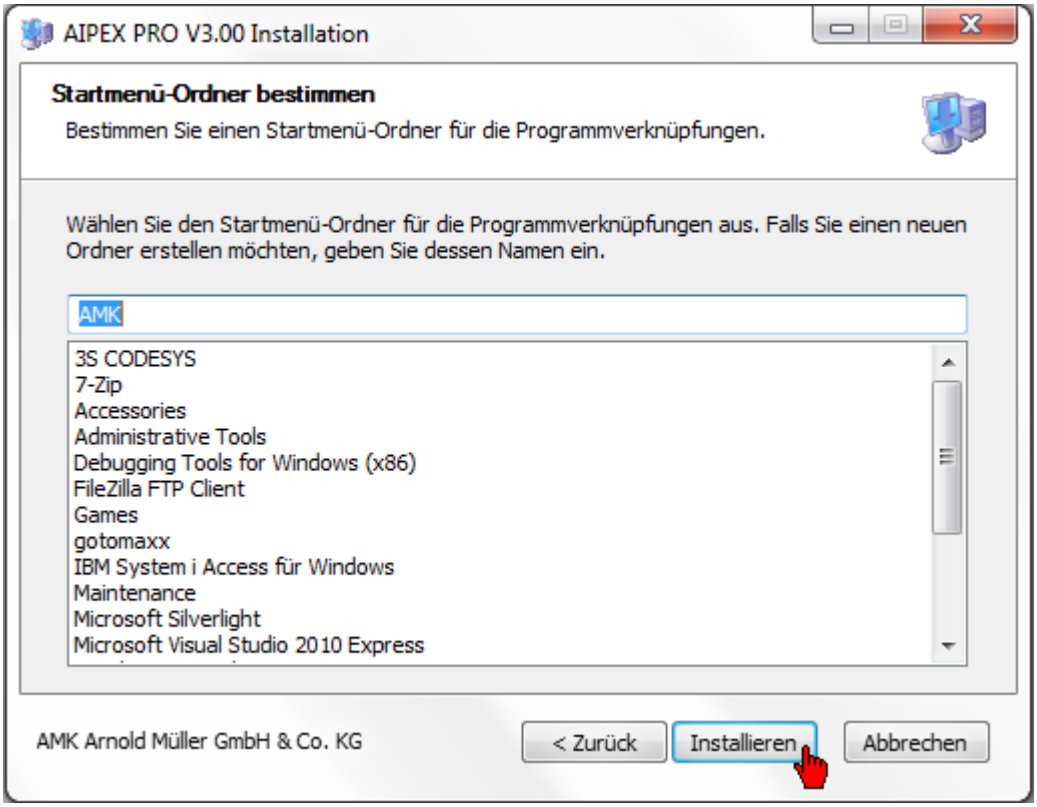
Alle Bestandteile von ATF V2.14 werden nicht in das Windows Verzeichnis (C:\Windows\System32), sondern in das lokale Verzeichnis des Programms (C:\Program Files (x86)\ATF V2.14) gespeichert.

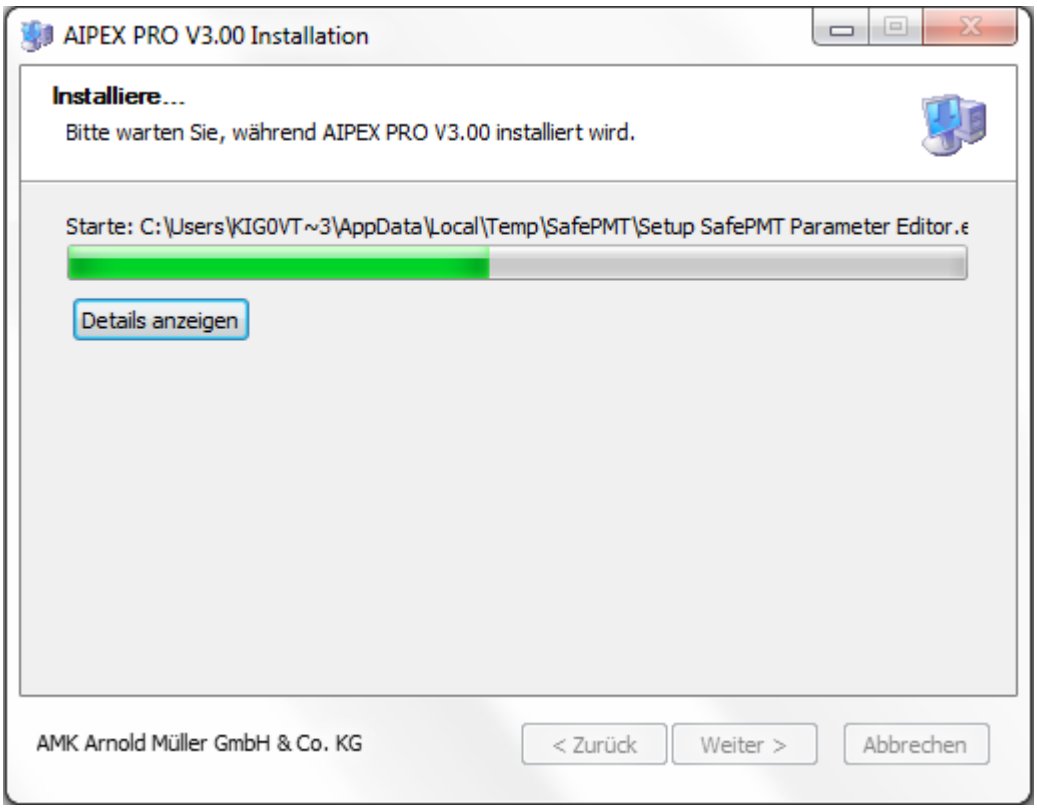


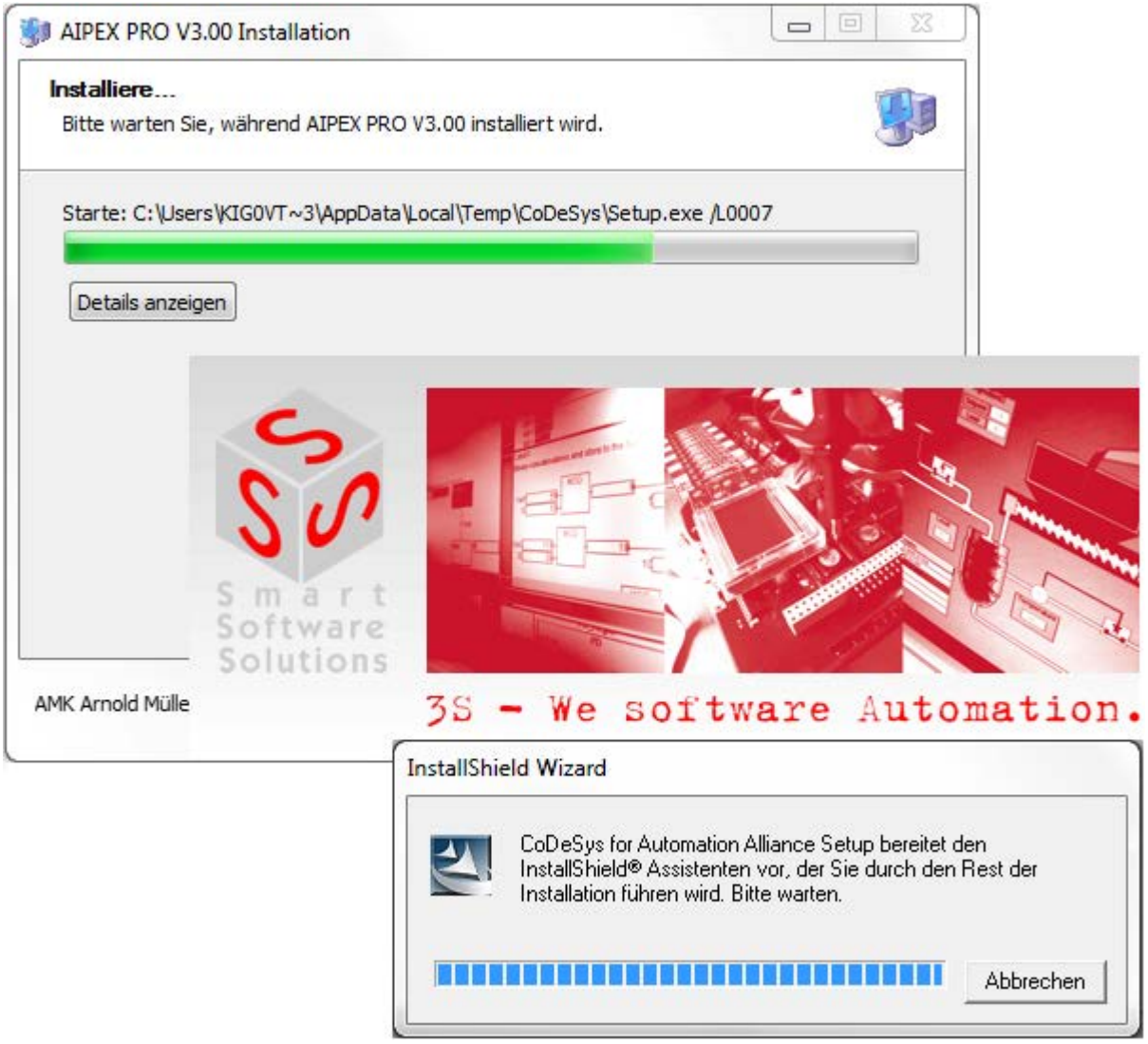
AIPEX: Inbetriebnahme- und Parameter Explorer
SafePMT: Safety Parametereditor für Geräte mit Safetyfunktionen
CODESYS V2: IEC 61131-3 Programmiersystem für PLC (CODESYS V2)
CODESYS V3: IEC 61131-3 Programmiersystem für PLC (CODESYS V3)
ATF: Werkzeug zur Firmware-Aktualisierung
USBCOM: Treiber für die serielle Kommunikation über USB
WinPcap: Windows Bibliothek packet capture für Zugriff auf EtherCAT
IXXAT: Treiber für USB-to-CAN Adapter von IXXAT

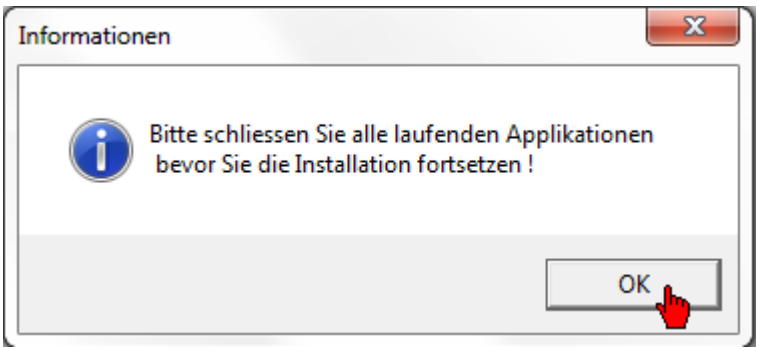
Bei der Deinstallation von AIPEX PRO werden alle installierten Treiber (WinPcap, USBCOM Treiber und IXXAT Treiber) vom PC entfernt.


AIPEX PRO V3.00 Installation
Zielverzeichnis auswählen
Wählen Sie das Verzeichnis aus, in das AIPEX PRO V3.00 installiert werden soll.
AIPEX PRO V3.00 wird in das unten angegebene Verzeichnis installiert. Falls Sie in ein anderes Verzeichnis installieren möchten, klicken Sie auf Durchsuchen und wählen Sie ein anderes Verzeichnis aus. Klicken Sie auf Weiter, um fortzufahren.
Zielverzeichnis
C:\Program Files (x86)
Durchsuchen...
Benötigter Speicher: 961.9MB
Verfügbarer Speicher: 412.2GB
AMK Arnold Müller GmbH & Co. KG
< Zurück
Weiter >
Abbrechen

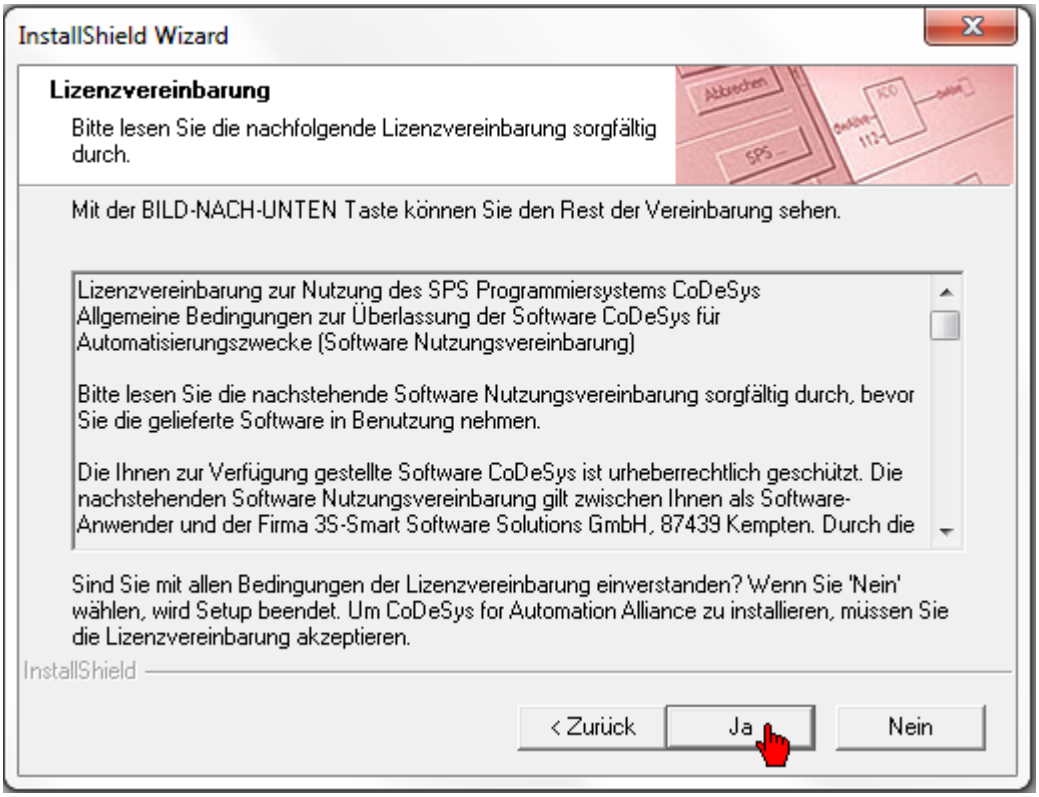

AIPEX PRO V3.00 Installation
Startmenü-Ordner bestimmen
Bestimmen Sie einen Startmenü-Ordner für die Programmverknüpfungen.
Wählen Sie den Startmenü-Ordner für die Programmverknüpfungen aus. Falls Sie einen neuen Ordner erstellen möchten, geben Sie dessen Namen ein.
AMK
3S CODESYS
7-Zip
Accessories
Administrative Tools
Debugging Tools for Windows (x86)
FileZilla FTP Client
Games
gotomaxx
IBM System i Access für Windows
Maintenance
Microsoft Silverlight
Microsoft Visual Studio 2010 Express
AMK Arnold Müller GmbH & Co. KG
< Zurück
Installieren
Abbrechen


AIPEX PRO V3.00 Installation
Installiere...
Bitte warten Sie, während AIPEX PRO V3.00 installiert wird.
Starte: C:\Users\KIG0VT~3\AppData\Local\Temp\SafePMT\Setup SafePMT Parameter Editor.e
Details anzeigen
AMK Arnold Müller GmbH & Co. KG
< Zurück
Weiter >
Abbrechen

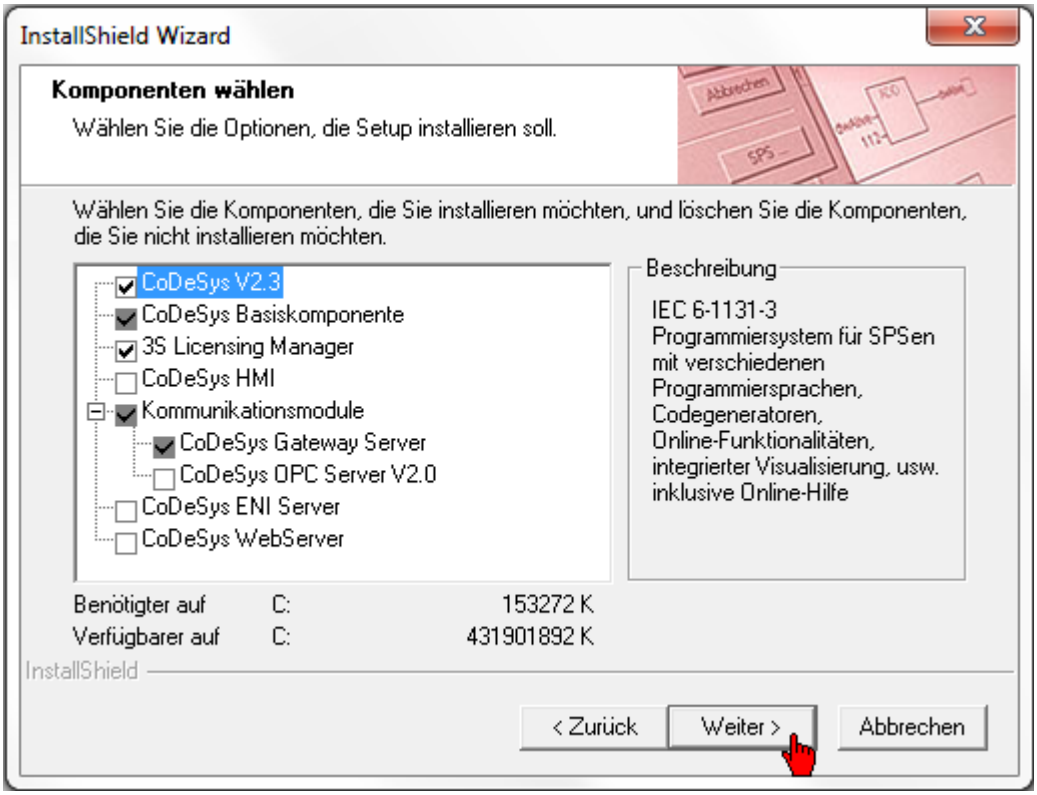

AIPEX PRO V3.00 Installation
Installiere...
Bitte warten Sie, während AIPEX PRO V3.00 installiert wird.
Starte: C:\Users\KIG0VT~3\AppData\Local\Temp\CoDeSys\Setup.exe /L0007
Details anzeigen
Smart Software Solutions
3S - We software Automation.
AMK Arnold Mülle
InstallShield Wizard
CoDeSys for Automation Alliance Setup bereitet den InstallShield® Assistenten vor, der Sie durch den Rest der Installation führen wird. Bitte warten.
Abbrechen

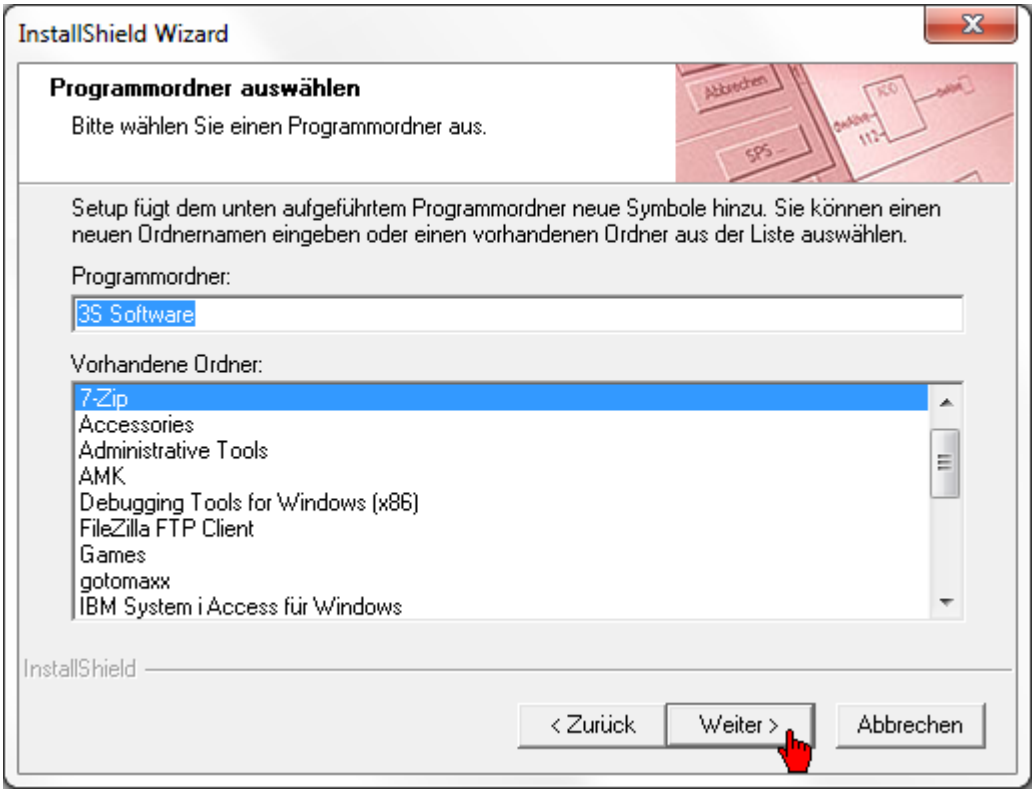

Informationen
Bitte schliessen Sie alle laufenden Applikationen
bevor Sie die Installation fortsetzen !
OK

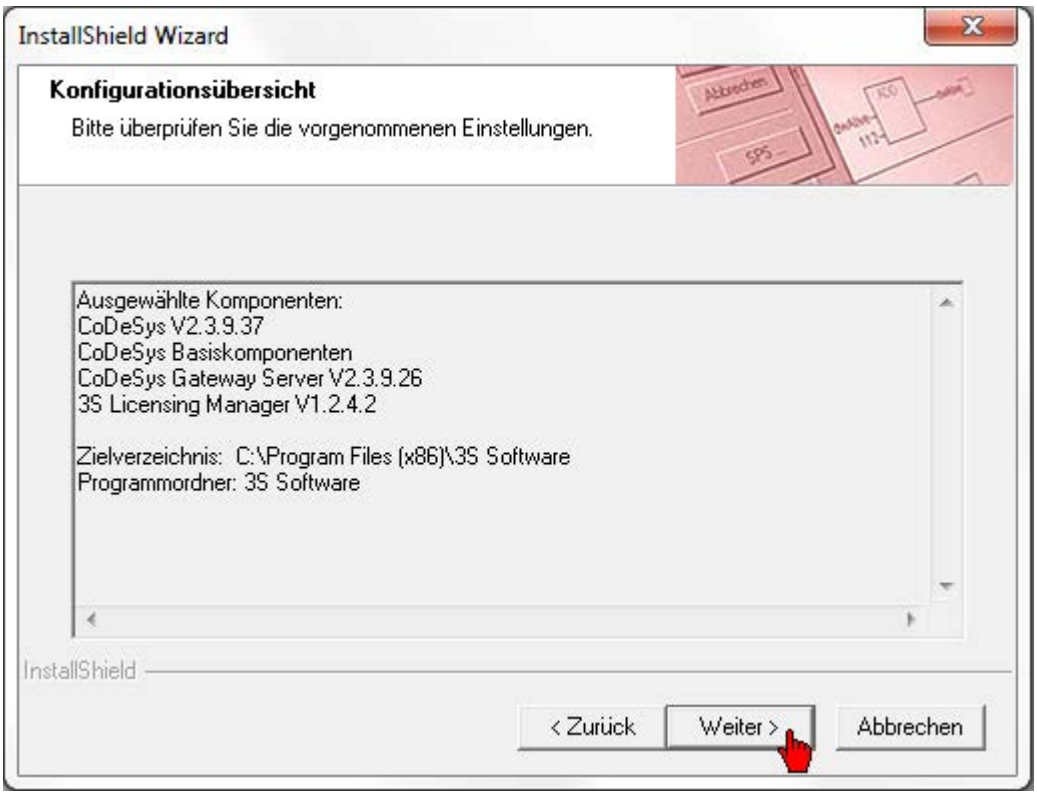
InstallShield Wizard
Smart
Software
Solutions
Willkommen bei InstallShield Wizard für
CoDeSys for Automation Alliance
InstallShield(r) Wizard installiert CoDeSys for Automation
Alliance auf Ihrem Computer. Klicken Sie auf 'Weiter', um
fortzufahren.
< Zurück
Weiter >
Abbrechen

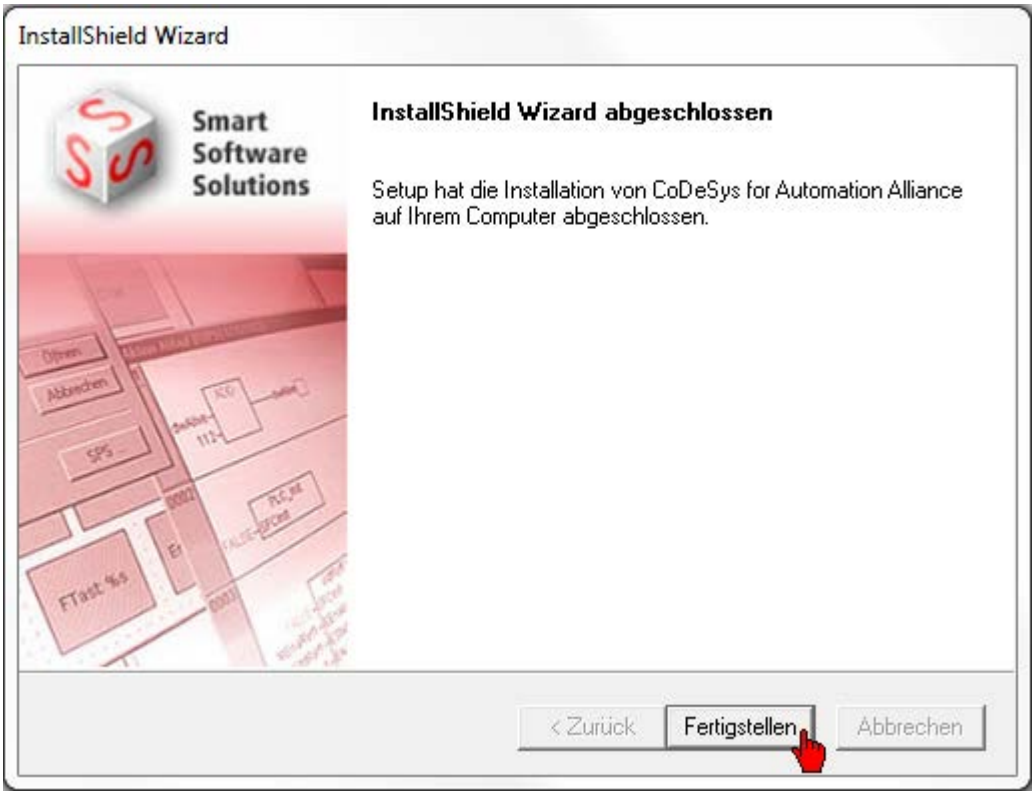

InstallShield Wizard
Lizenzvereinbarung
Bitte lesen Sie die nachfolgende Lizenzvereinbarung sorgfältig durch.
Mit der BILD-NACH-UNTEN Taste können Sie den Rest der Vereinbarung sehen.
Lizenzvereinbarung zur Nutzung des SPS Programmiersystems CoDeSys
Allgemeine Bedingungen zur Überlassung der Software CoDeSys für Automatisierungszwecke (Software Nutzungsvereinbarung)
Bitte lesen Sie die nachstehende Software Nutzungsvereinbarung sorgfältig durch, bevor Sie die gelieferte Software in Benutzung nehmen.
Die Ihnen zur Verfügung gestellte Software CoDeSys ist urheberrechtlich geschützt. Die nachstehenden Software Nutzungsvereinbarung gilt zwischen Ihnen als Software-Anwender und der Firma 3S-Smart Software Solutions GmbH, 87439 Kempten. Durch die
Sind Sie mit allen Bedingungen der Lizenzvereinbarung einverstanden? Wenn Sie 'Nein' wählen, wird Setup beendet. Um CoDeSys for Automation Alliance zu installieren, müssen Sie die Lizenzvereinbarung akzeptieren.
InstallShield
< Zurück
Ja
Nein

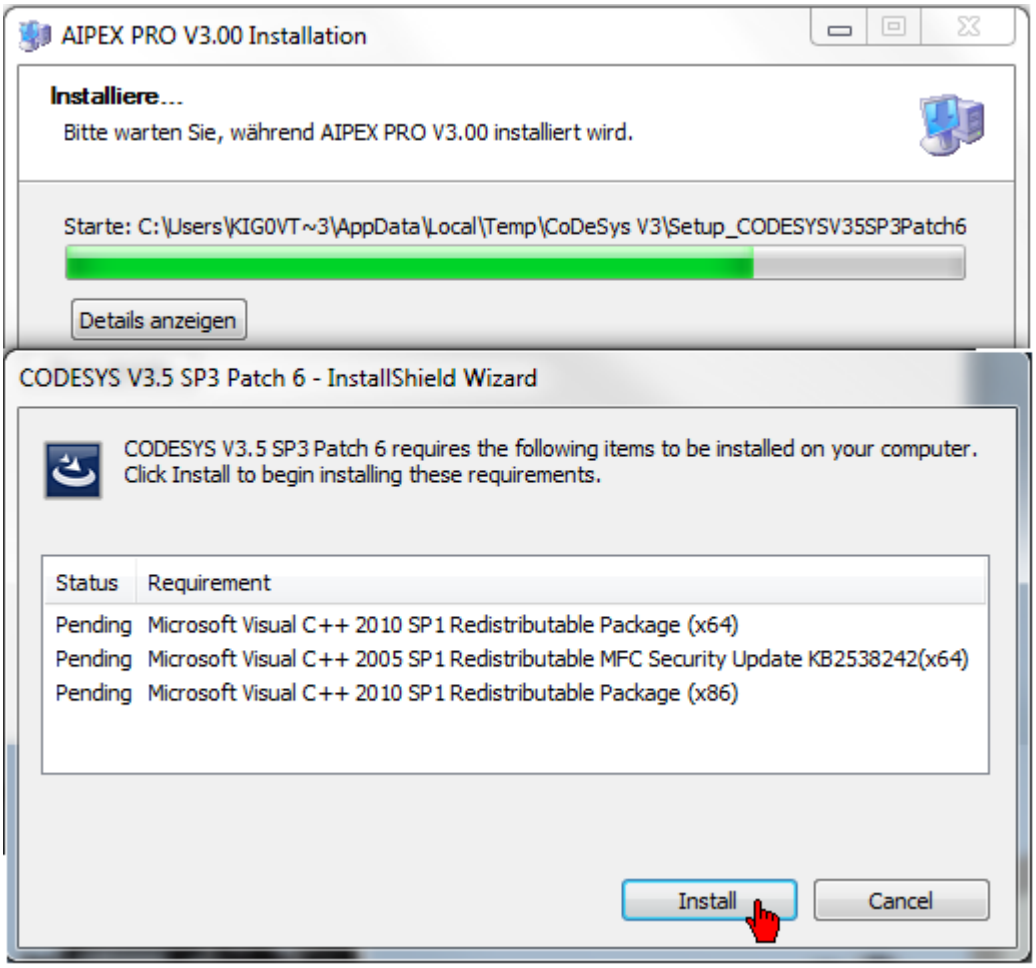
InstallShield Wizard
Zielpfad wählen
Wählen Sie den Zielordner aus, in den das Programm installiert werden soll.
Setup installiert CoDeSys for Automation Alliance in den folgenden Ordner.
Die einzelnen Komponenten werden in einen jeweiligen Unterordner installiert.
Klicken Sie auf 'Weiter', um in diesen Ordner zu installieren. Um in einen anderen Ordner zu installieren, klicken Sie auf 'Durchsuchen' und wählen Sie einen anderen Ordner.
Zielordner
C:\Program Files (x86)\3S Software
Durchsuchen...
InstallShield
< Zurück
Weiter >
Abbrechen

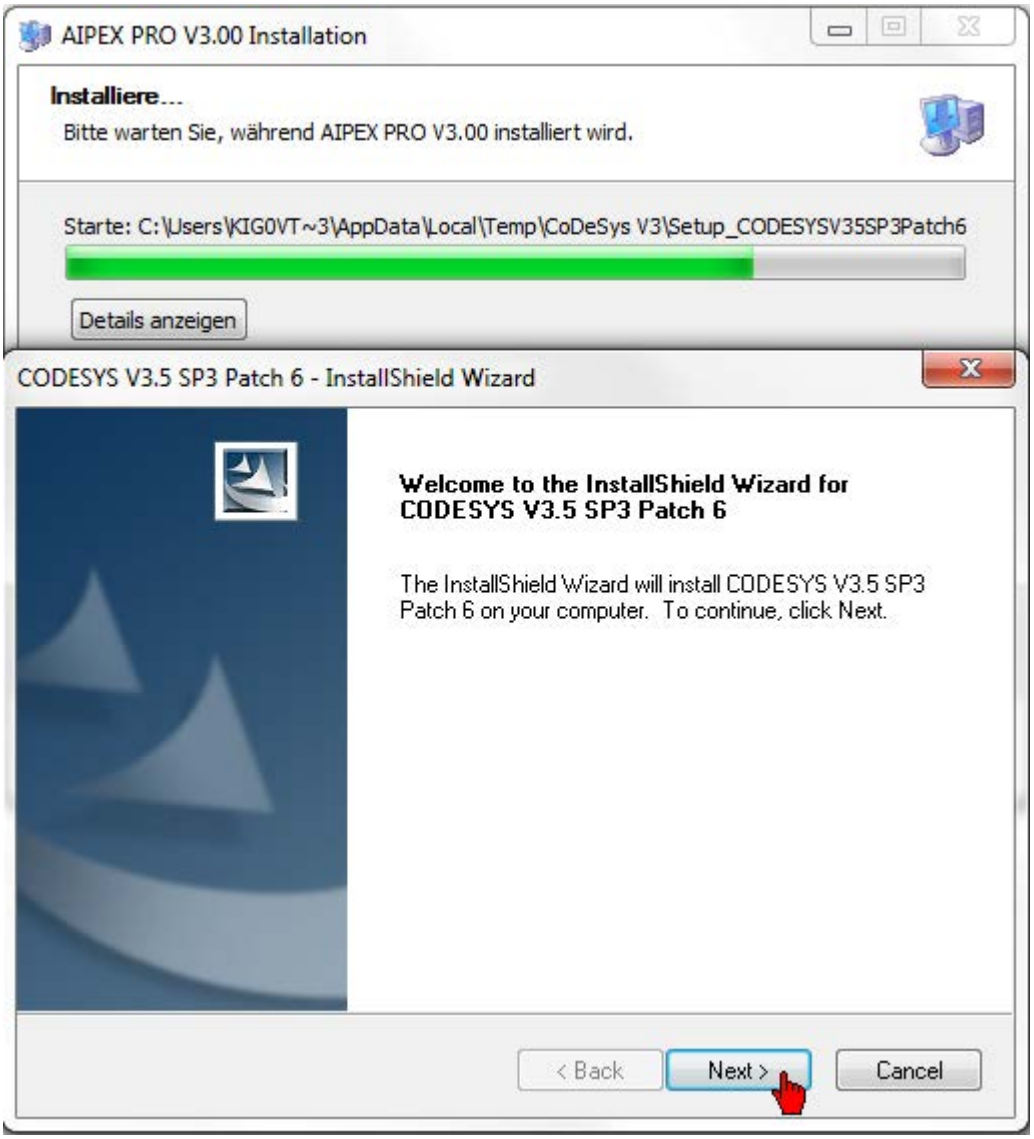

InstallShield Wizard
Komponenten wählen
Wählen Sie die Optionen, die Setup installieren soll.
Wählen Sie die Komponenten, die Sie installieren möchten, und löschen Sie die Komponenten, die Sie nicht installieren möchten.
CoDeSys V2.3
CoDeSys Basiskomponente
3S Licensing Manager
CoDeSys HMI
Kommunikationsmodule
CoDeSys Gateway Server
CoDeSys OPC Server V2.0
CoDeSys ENI Server
CoDeSys WebServer
Beschreibung
IEC 6-1131-3 Programmiersystem für SPSen mit verschiedenen Programmiersprachen, Codegeneratoren, Online-Funktionalitäten, integrierter Visualisierung, usw. inklusive Online-Hilfe
Benötigter auf C: 153272 K
Verfügbarer auf C: 431901892 K
InstallShield
< Zurück
Weiter >
Abbrechen

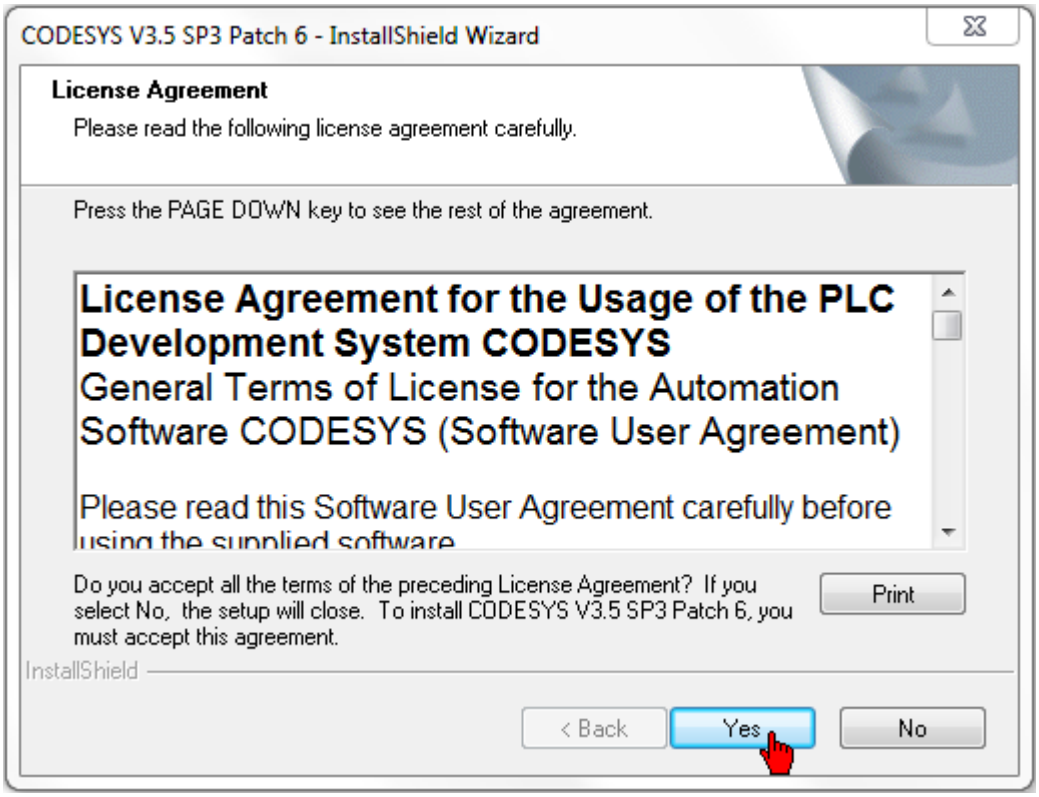

InstallShield Wizard
Programmordner auswählen
Bitte wählen Sie einen Programmordner aus.
Setup fügt dem unten aufgeführtem Programmordner neue Symbole hinzu. Sie können einen neuen Ordnernamen eingeben oder einen vorhandenen Ordner aus der Liste auswählen.
Programmordner:
3S Software
Vorhandene Ordner:
7-Zip
Accessories
Administrative Tools
AMK
Debugging Tools for Windows (x86)
FileZilla FTP Client
Games
gotomaxx
IBM System i Access für Windows
InstallShield
< Zurück
Weiter >
Abbrechen

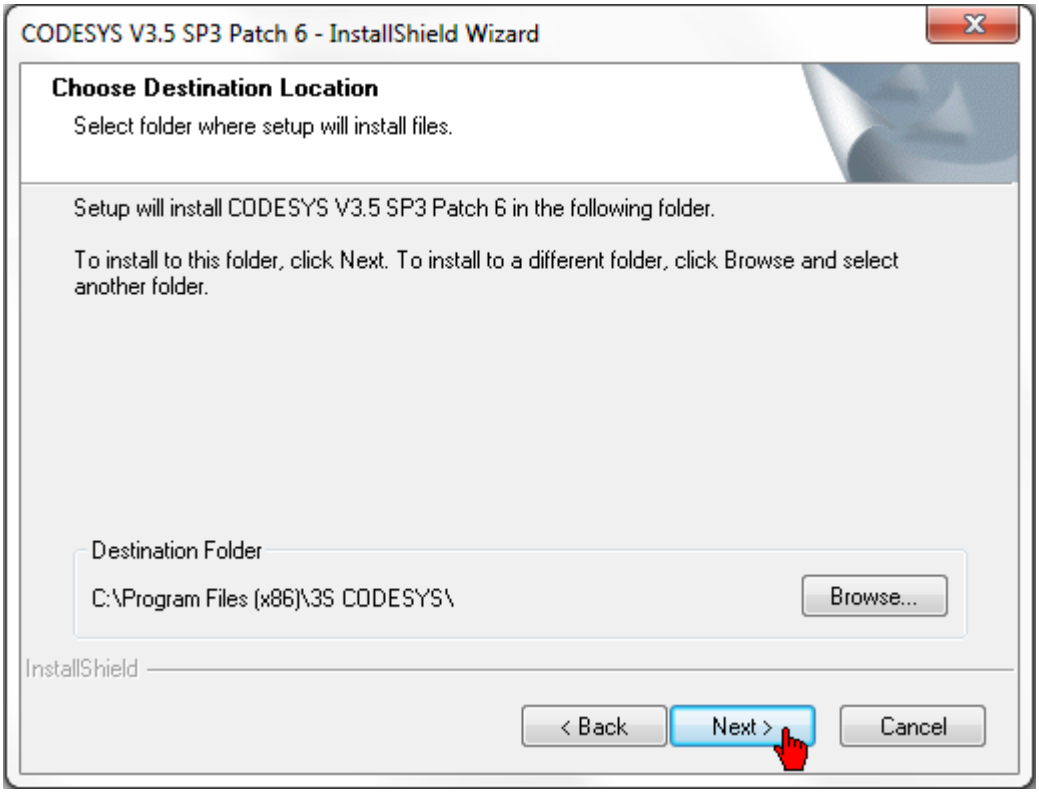

InstallShield Wizard
Konfigurationsübersicht
Bitte überprüfen Sie die vorgenommenen Einstellungen.
Ausgewählte Komponenten:
CoDeSys V2.3.9.37
CoDeSys Basiskomponenten
CoDeSys Gateway Server V2.3.9.26
3S Licensing Manager V1.2.4.2
Zielverzeichnis: C:\Program Files (x86)\3S Software
Programmordner: 3S Software
InstallShield
< Zurück
Weiter >
Abbrechen

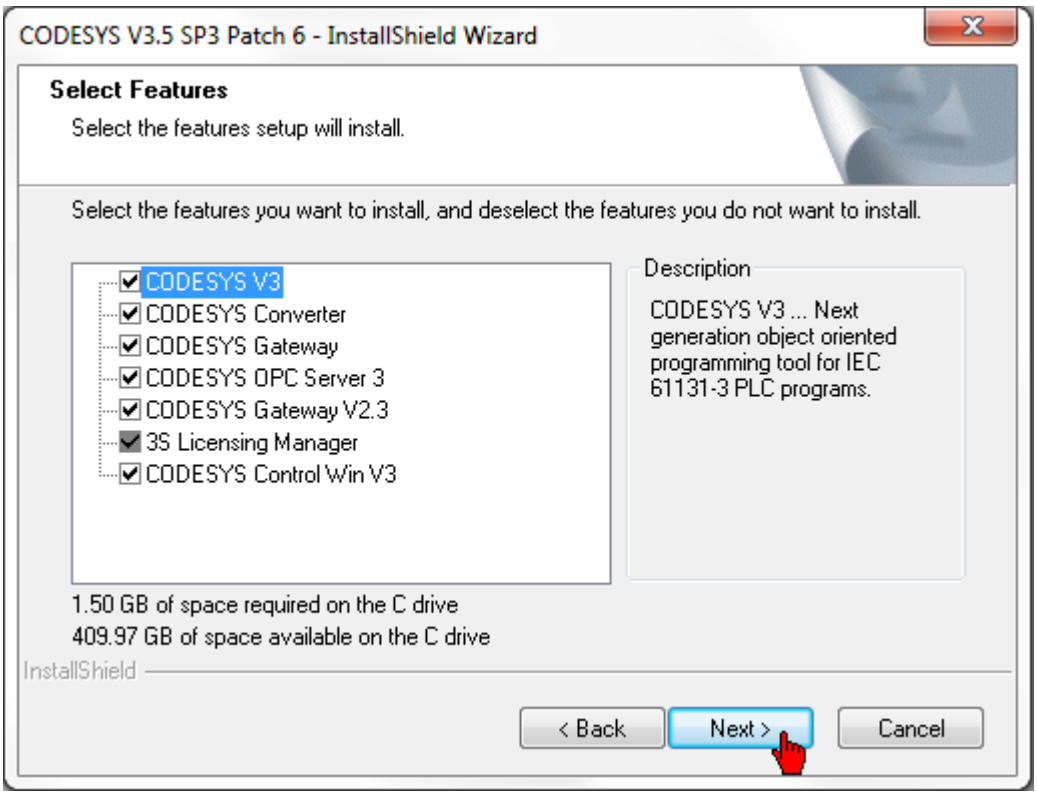

InstallShield Wizard
Smart
Software
Solutions
InstallShield Wizard abgeschlossen
Setup hat die Installation von CoDeSys for Automation Alliance auf Ihrem Computer abgeschlossen.
< Zurück
Fertigstellen
Abbrechen

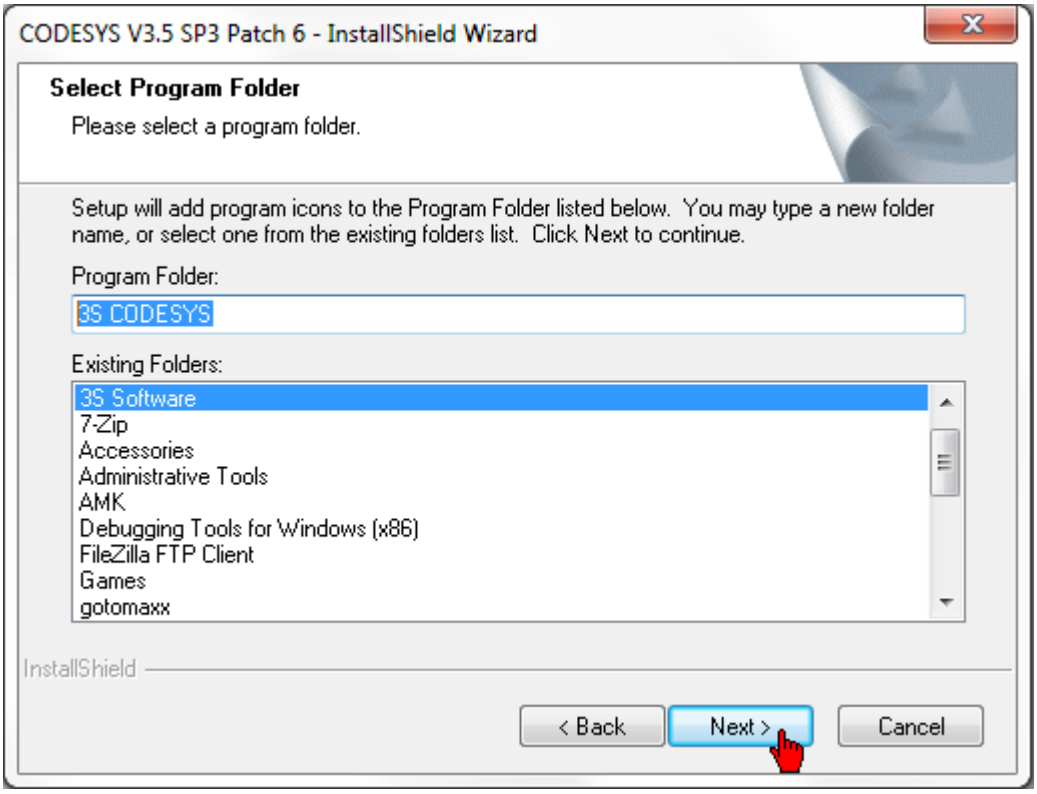

AIPEX PRO V3.00 Installation
Installiere...
Bitte warten Sie, während AIPEX PRO V3.00 installiert wird.
Starte: C:\Users\KIG0VT~3\AppData\Local\Temp\CoDeSys V3\Setup_CODESYSV35SP3Patch6
Details anzeigen
CODESYS V3.5 SP3 Patch 6 - InstallShield Wizard
CODESYS V3.5 SP3 Patch 6 requires the following items to be installed on your computer.
Click Install to begin installing these requirements.
Status
Requirement
Pending
Microsoft Visual C++ 2010 SP1 Redistributable Package (x64)
Pending
Microsoft Visual C++ 2005 SP1 Redistributable MFC Security Update KB2538242(x64)
Pending
Microsoft Visual C++ 2010 SP1 Redistributable Package (x86)
Install
Cancel

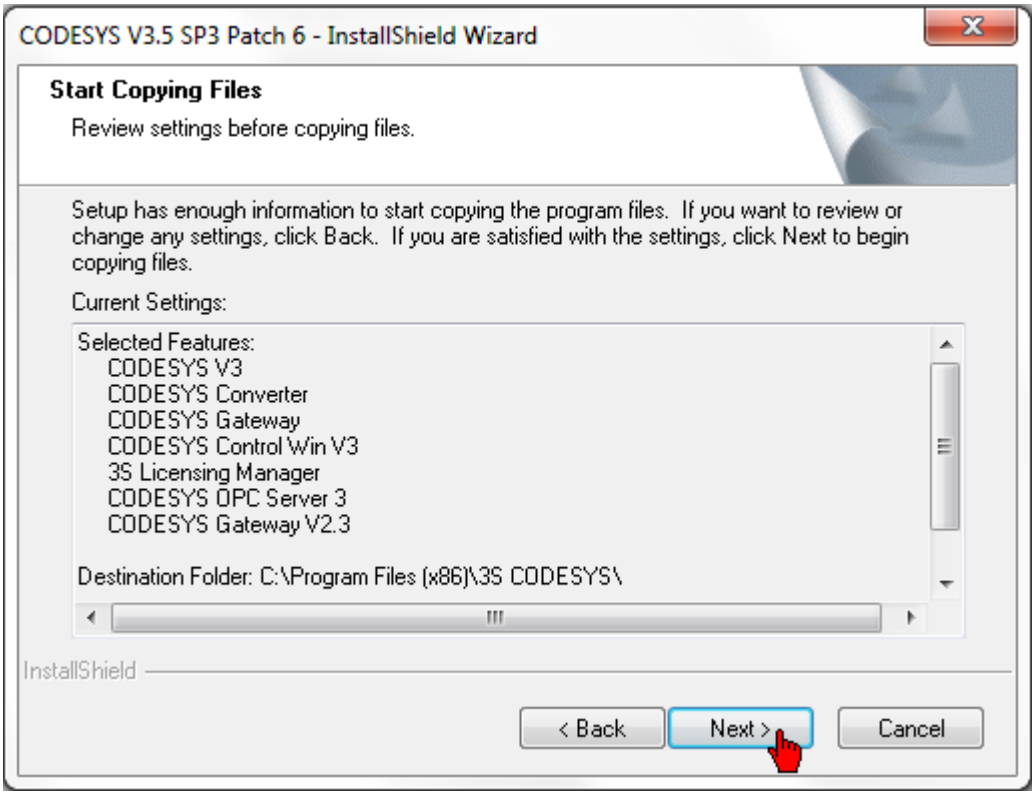

AIPEX PRO V3.00 Installation
Installiere...
Bitte warten Sie, während AIPEX PRO V3.00 installiert wird.
Starte: C:\Users\KIG0VT~3\AppData\Local\Temp\CoDeSys V3\Setup_CODESYSV35SP3Patch6
Details anzeigen
CODESYS V3.5 SP3 Patch 6 - InstallShield Wizard
Welcome to the InstallShield Wizard for CODESYS V3.5 SP3 Patch 6
The InstallShield Wizard will install CODESYS V3.5 SP3 Patch 6 on your computer. To continue, click Next.
< Back
Next >
Cancel

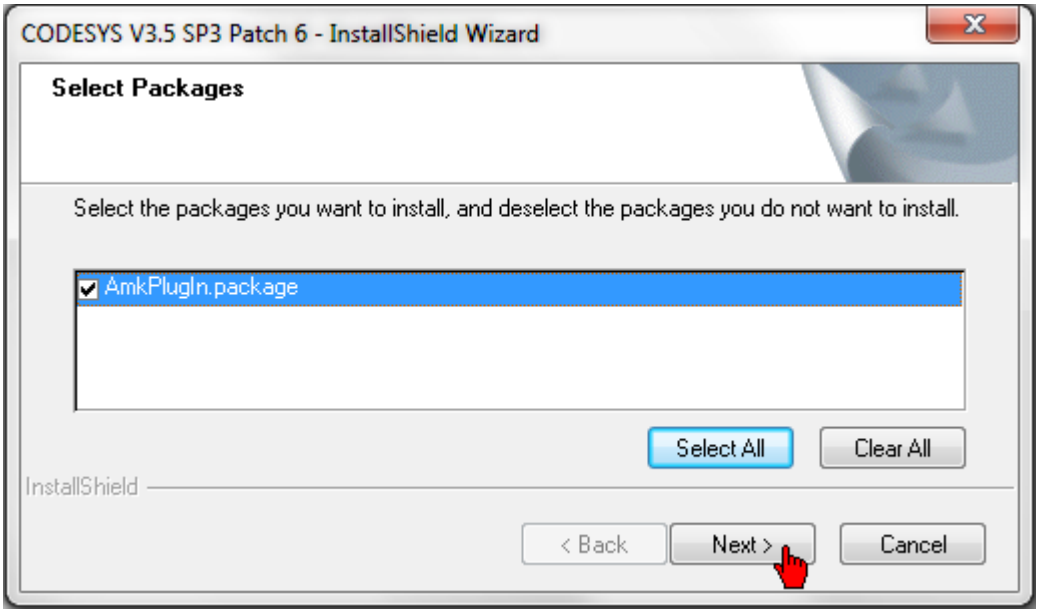

CODESYS V3.5 SP3 Patch 6 - InstallShield Wizard
License Agreement
Please read the following license agreement carefully.
Press the PAGE DOWN key to see the rest of the agreement.
License Agreement for the Usage of the PLC Development System CODESYS
General Terms of License for the Automation Software CODESYS (Software User Agreement)
Please read this Software User Agreement carefully before using the supplied software
Do you accept all the terms of the preceding License Agreement? If you select No, the setup will close. To install CODESYS V3.5 SP3 Patch 6, you must accept this agreement.
Print
InstallShield
< Back
Yes
No

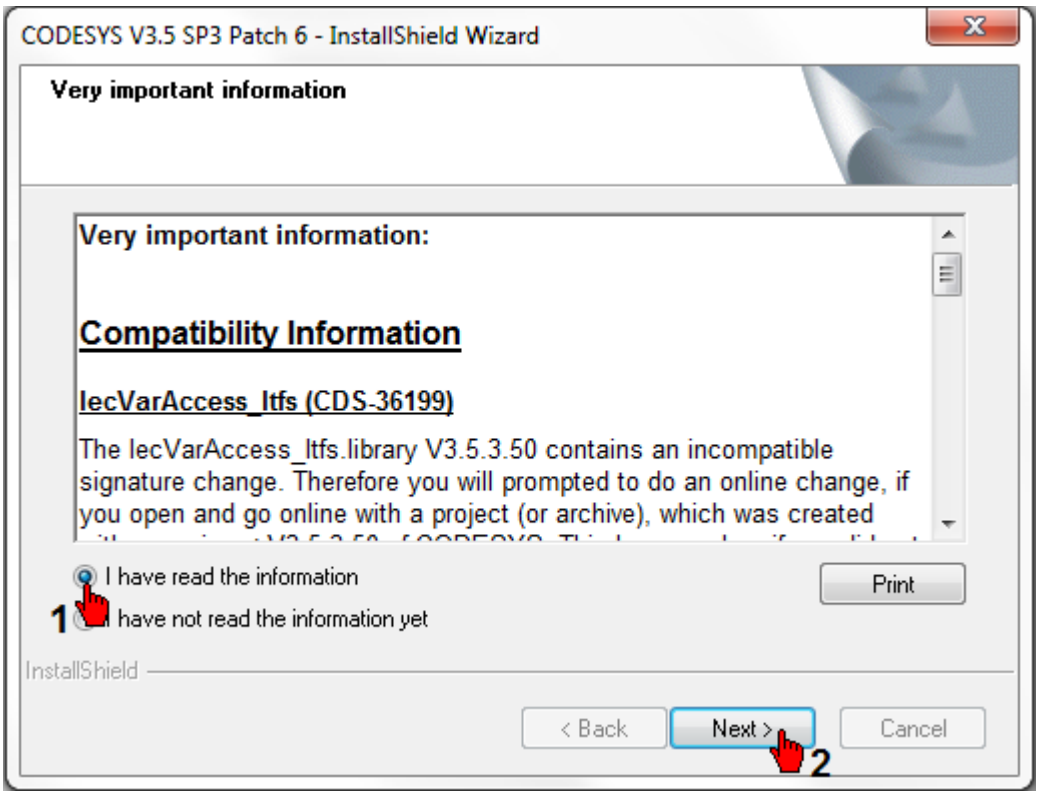

CODESYS V3.5 SP3 Patch 6 - InstallShield Wizard
Choose Destination Location
Select folder where setup will install files.
Setup will install CODESYS V3.5 SP3 Patch 6 in the following folder.
To install to this folder, click Next. To install to a different folder, click Browse and select another folder.
Destination Folder
C:\Program Files (x86)\3S CODESYS\
Browse...
InstallShield
< Back
Next >
Cancel

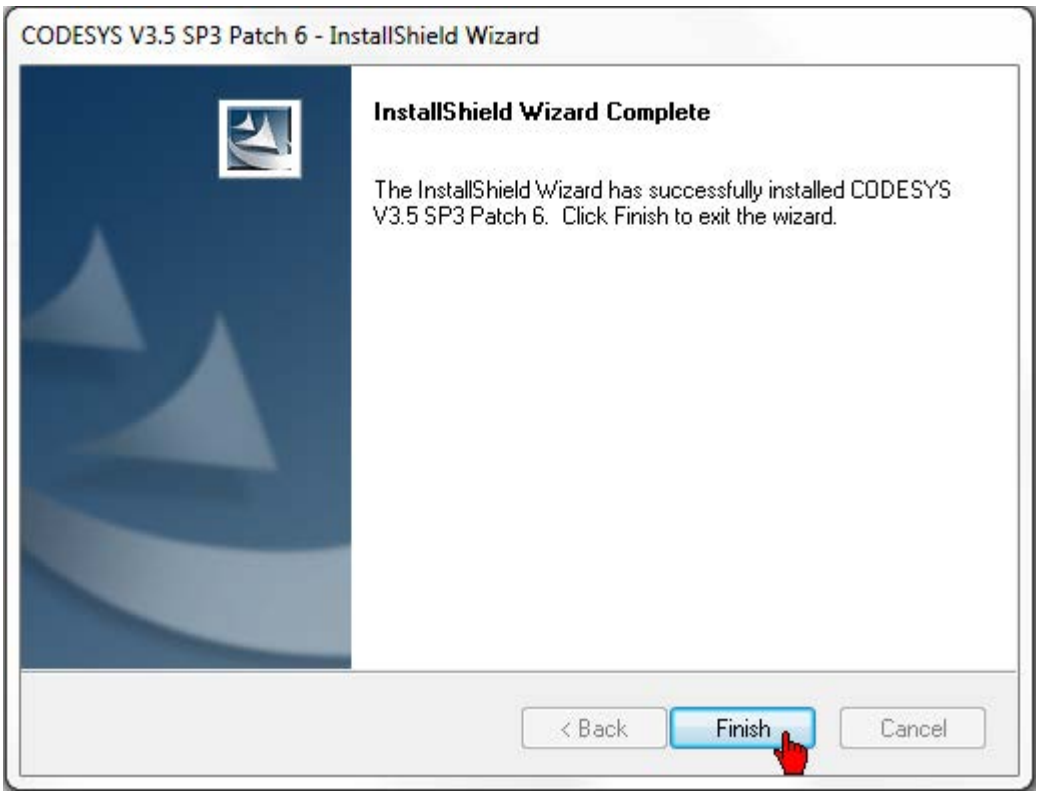

CODESYS V3.5 SP3 Patch 6 - InstallShield Wizard
Select Features
Select the features setup will install.
Select the features you want to install, and deselect the features you do not want to install.
CODESYS V3
CODESYS Converter
CODESYS Gateway
CODESYS OPC Server 3
CODESYS Gateway V2.3
3S Licensing Manager
CODESYS Control Win V3
Description
CODESYS V3 ... Next generation object oriented programming tool for IEC 61131-3 PLC programs.
1.50 GB of space required on the C drive
409.97 GB of space available on the C drive
InstallShield
< Back
Next >
Cancel

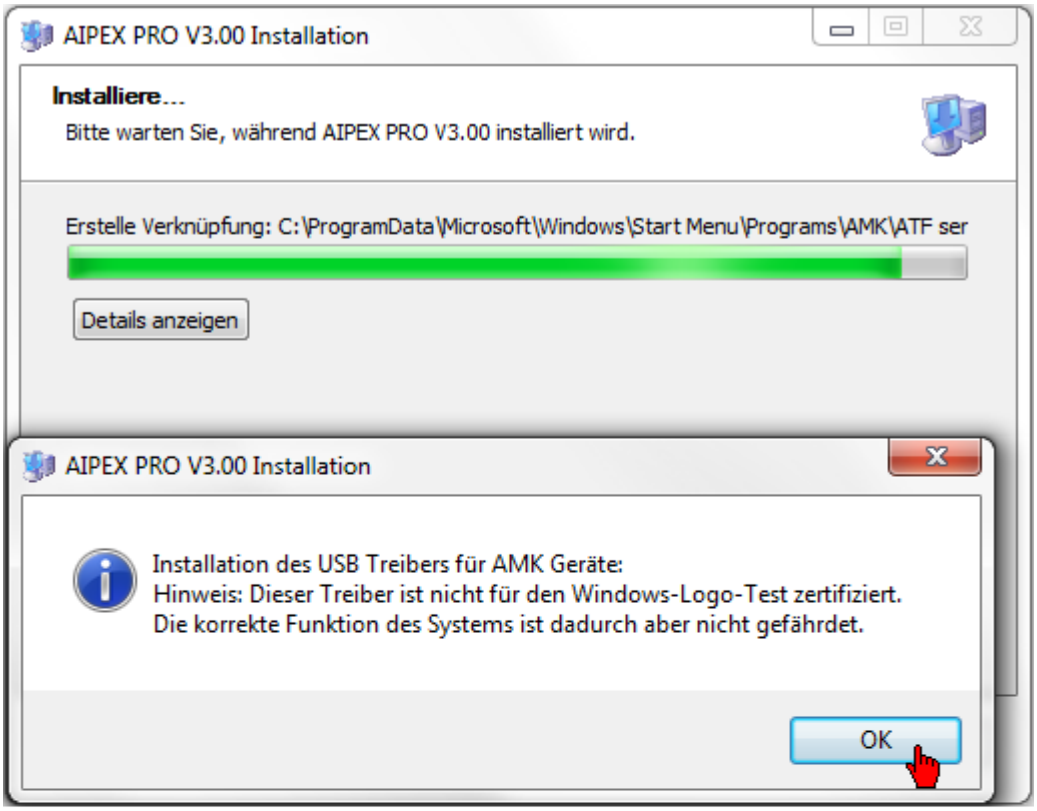

CODESYS V3.5 SP3 Patch 6 - InstallShield Wizard
Select Program Folder
Please select a program folder.
Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing folders list. Click Next to continue.
Program Folder:
3S CODESYS
Existing Folders:
3S Software
7-Zip
Accessories
Administrative Tools
AMK
Debugging Tools for Windows (x86)
FileZilla FTP Client
Games
gotomaxx
InstallShield
< Back
Next >
Cancel

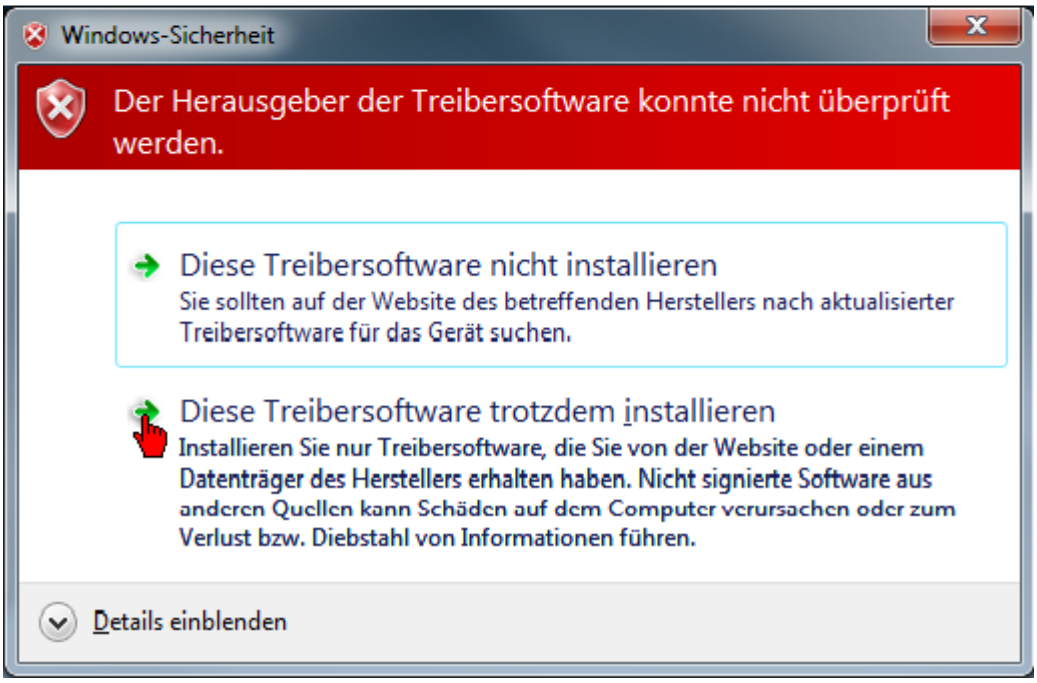

CODESYS V3.5 SP3 Patch 6 - InstallShield Wizard
Start Copying Files
Review settings before copying files.
Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.
Current Settings:
Selected Features:
CODESYS V3
CODESYS Converter
CODESYS Gateway
CODESYS Control Win V3
3S Licensing Manager
CODESYS OPC Server 3
CODESYS Gateway V2.3
Destination Folder: C:\Program Files (x86)\3S CODESYS\
InstallShield
< Back
Next >
Cancel


CODESYS V3.5 SP3 Patch 6 - InstallShield Wizard
Select Packages
Select the packages you want to install, and deselect the packages you do not want to install.
AmkPlugIn.package
Select All
Clear All
InstallShield
< Back
Next >
Cancel

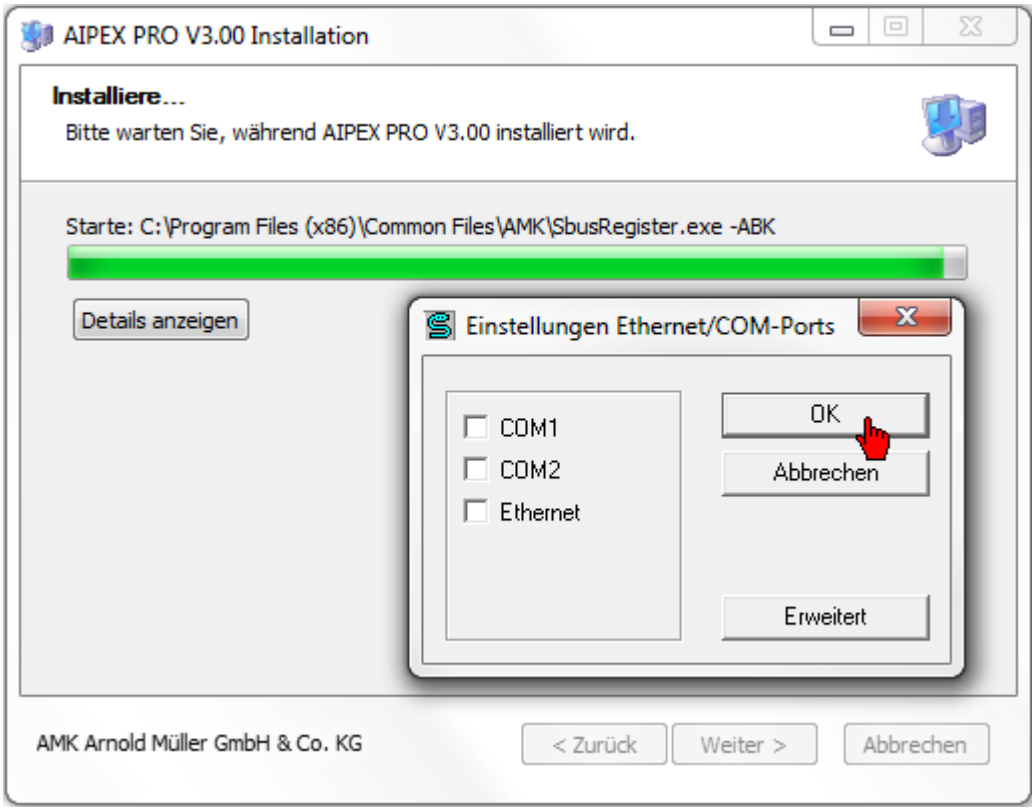

CODESYS V3.5 SP3 Patch 6 - InstallShield Wizard
Very important information
Very important information:
Compatibility Information
IecVarAccess_Itfs (CDS-36199)
The IecVarAccess_Itfs.library V3.5.3.50 contains an incompatible signature change. Therefore you will prompted to do an online change, if you open and go online with a project (or archive), which was created
I have read the information
have not read the information yet
Print
1
InstallShield
< Back
Next >
Cancel
2


CODESYS V3.5 SP3 Patch 6 - InstallShield Wizard
InstallShield Wizard Complete
The InstallShield Wizard has successfully installed CODESYS V3.5 SP3 Patch 6. Click Finish to exit the wizard.
< Back
Finish
Cancel

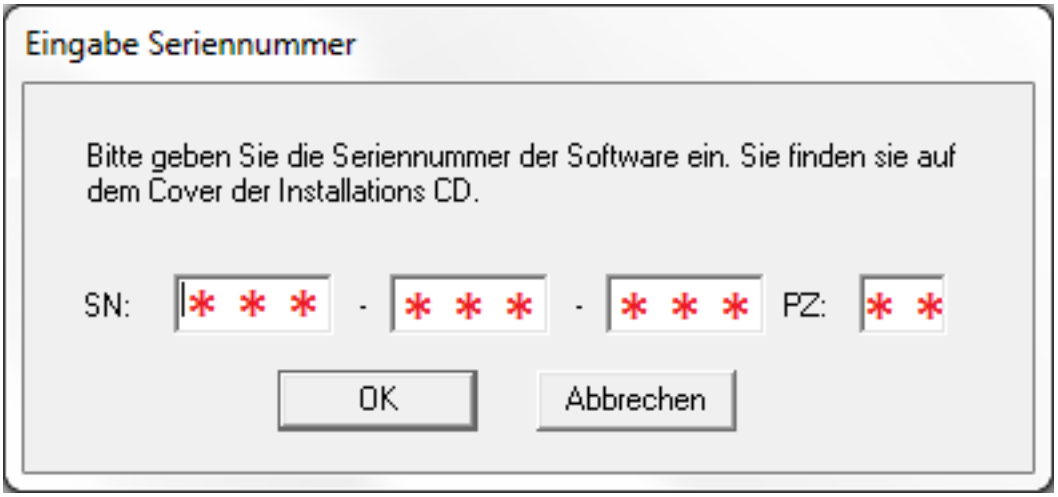

AIPEX PRO V3.00 Installation
Installiere...
Bitte warten Sie, während AIPEX PRO V3.00 installiert wird.
Erstelle Verknüpfung: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\AMK\ATF ser
Details anzeigen
AIPEX PRO V3.00 Installation
Installation des USB Treibers für AMK Geräte:
Hinweis: Dieser Treiber ist nicht für den Windows-Logo-Test zertifiziert.
Die korrekte Funktion des Systems ist dadurch aber nicht gefährdet.
OK

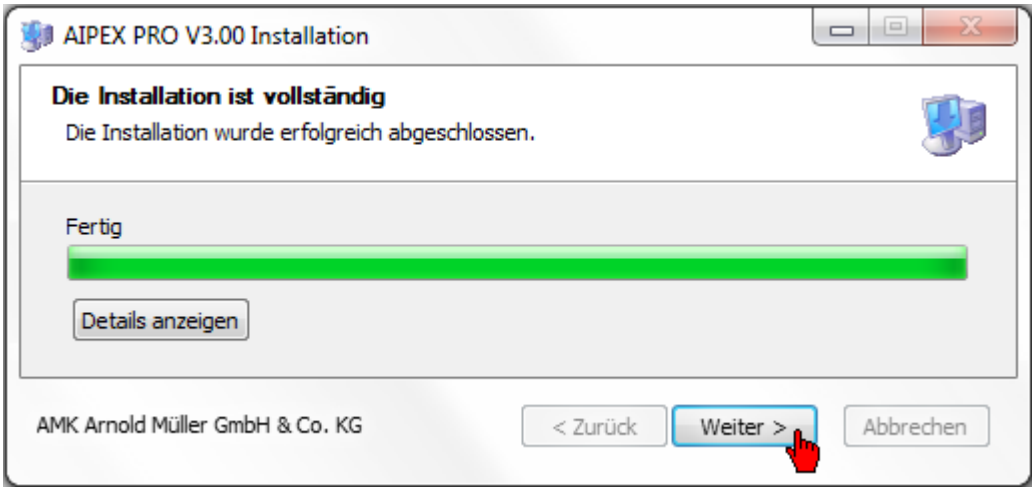

Windows-Sicherheit
Der Herausgeber der Treibersoftware konnte nicht überprüft werden.
Diese Treibersoftware nicht installieren
Sie sollten auf der Website des betreffenden Herstellers nach aktualisierter Treibersoftware für das Gerät suchen.
Diese Treibersoftware trotzdem installieren
Installieren Sie nur Treibersoftware, die Sie von der Website oder einem Datenträger des Herstellers erhalten haben. Nicht signierte Software aus anderen Quellen kann Schäden auf dem Computer verursachen oder zum Verlust bzw. Diebstahl von Informationen führen.
Details einblenden

Kundenspezifische Eingabe der Schnittstelle zwischen PC und AMK Gerät.
Die Schnittstelle kann später im laufenden Programm gewechselt werden.
Siehe 'Kommunikation PC - AMK Gerät' auf Seite 43.



Kundenspezifische Eingabe der AIPEX PRO Seriennummer





3.2 Servicepack

Ein AIPEX PRO Servicepack wird benötigt, wenn nach einer AIPEX PRO Freigabe „neue“, das bedeutet für AIPEX PRO noch unbekannte Geräte bzw. Geräteeigenschaften von AMK freigegeben wurden.

Die Installation erfolgt nach Aufruf des gelieferten Setup.exe automatisch.

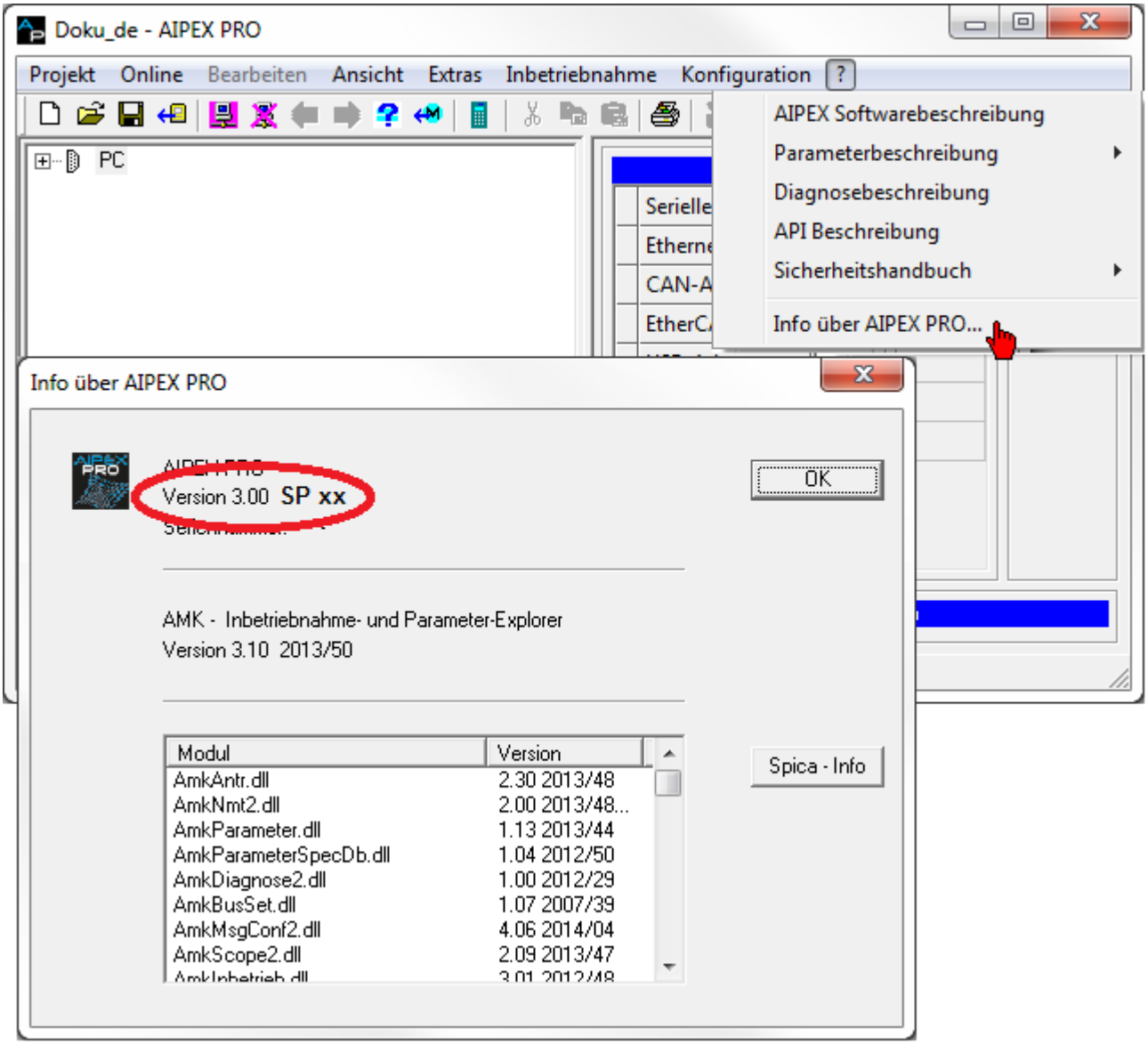
Die Installation eines Servicepacks ist nur für die dazugehörige AIPEX PRO Version möglich. Beispiel: Das Servicepack AipexPro_105_SP04_1051_203405.zip kann nur für eine AIPEX PRO Version V1.05 nachinstalliert werden.

Jedes Servicepack enthält alle Eigenschaften der vorhergehenden Servicepacks. Deshalb ist es ausreichend wenn immer nur das neuste Servicepack installiert wird.

Info über AIPEX PRO

Anzeige der AIPEX PRO Version und des installierten Servicepacks.

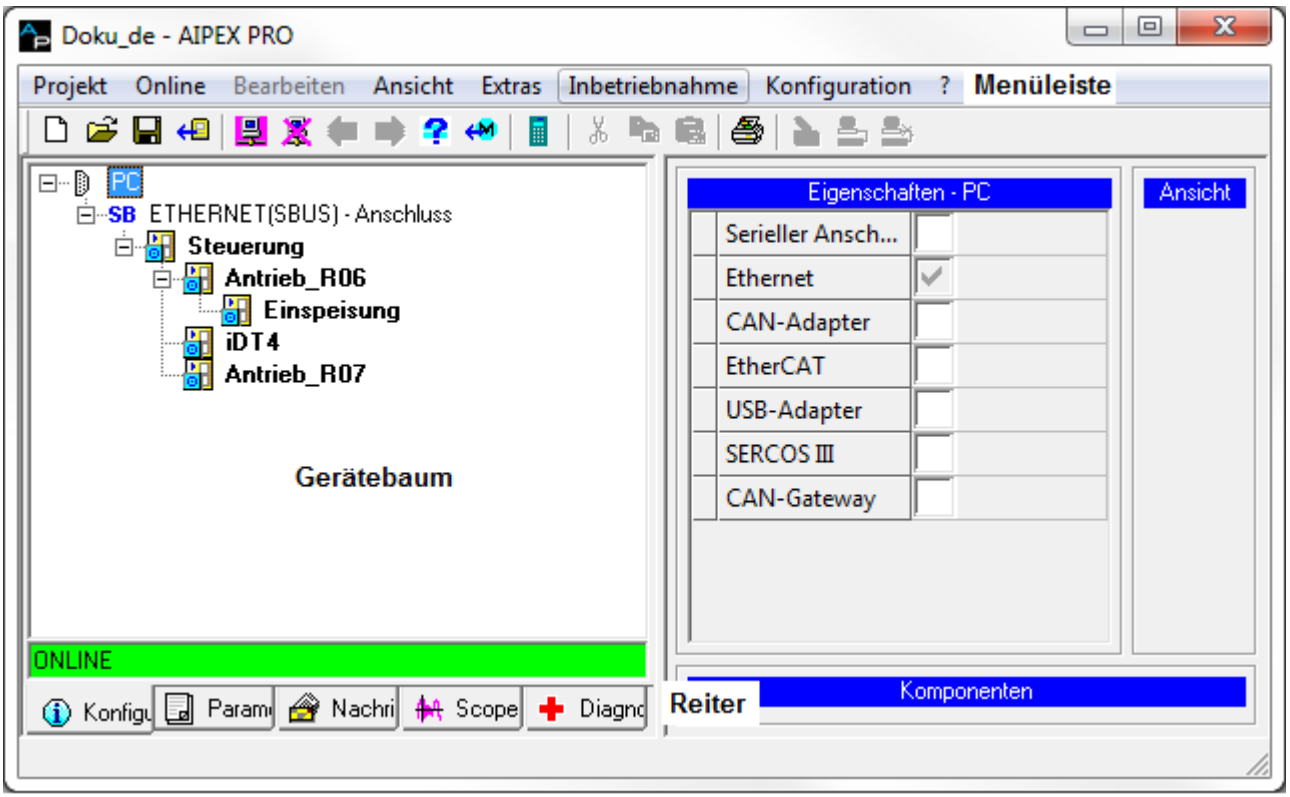
In der Tabelle werden die Versionen der zugehörigen Komponenten und Datenbanken aufgelistet.



3.3 Programmübersicht

AIPEX PRO integriert alle Engineering Tools, die während des Lifecycles einer Maschine benötigt werden, wie z.B. Programmierung, Parametrierung Inbetriebnahme, Optimierung und Diagnose. Dies erspart Ihnen aufwendige Abstimmungen beispielsweise zwischen Ihrem PLC-Programm mit den Antriebsparametern und der Konfiguration des Nutzdatenaustausches über den Feldbus.

AIPEX PRO arbeitet hier vollautomatisch. Das Engineering Tool verfügt ebenfalls über einen Visualisierungseditor mit vorgefertigten Bausteinen.



Reiter	Funktion
Konfiguration	Anzeige und Eingabemöglichkeit für Geräteeigenschaften Siehe 'Konfiguration' auf Seite 70.
Parameter	Anzeige und Eingabemöglichkeit für Parameterwerte Siehe 'Parameter' auf Seite 75.
Nachrichten	Anzeige und Eingabemöglichkeit für Netzwerkvariablen Siehe 'Nachrichten' auf Seite 84.
Scope	Oszilloskopefunktion um Gerätesignale aufzuzeichnen Siehe 'Scope' auf Seite 92.
Diagnose	Diagnosemodul Siehe 'Diagnose' auf Seite 96.

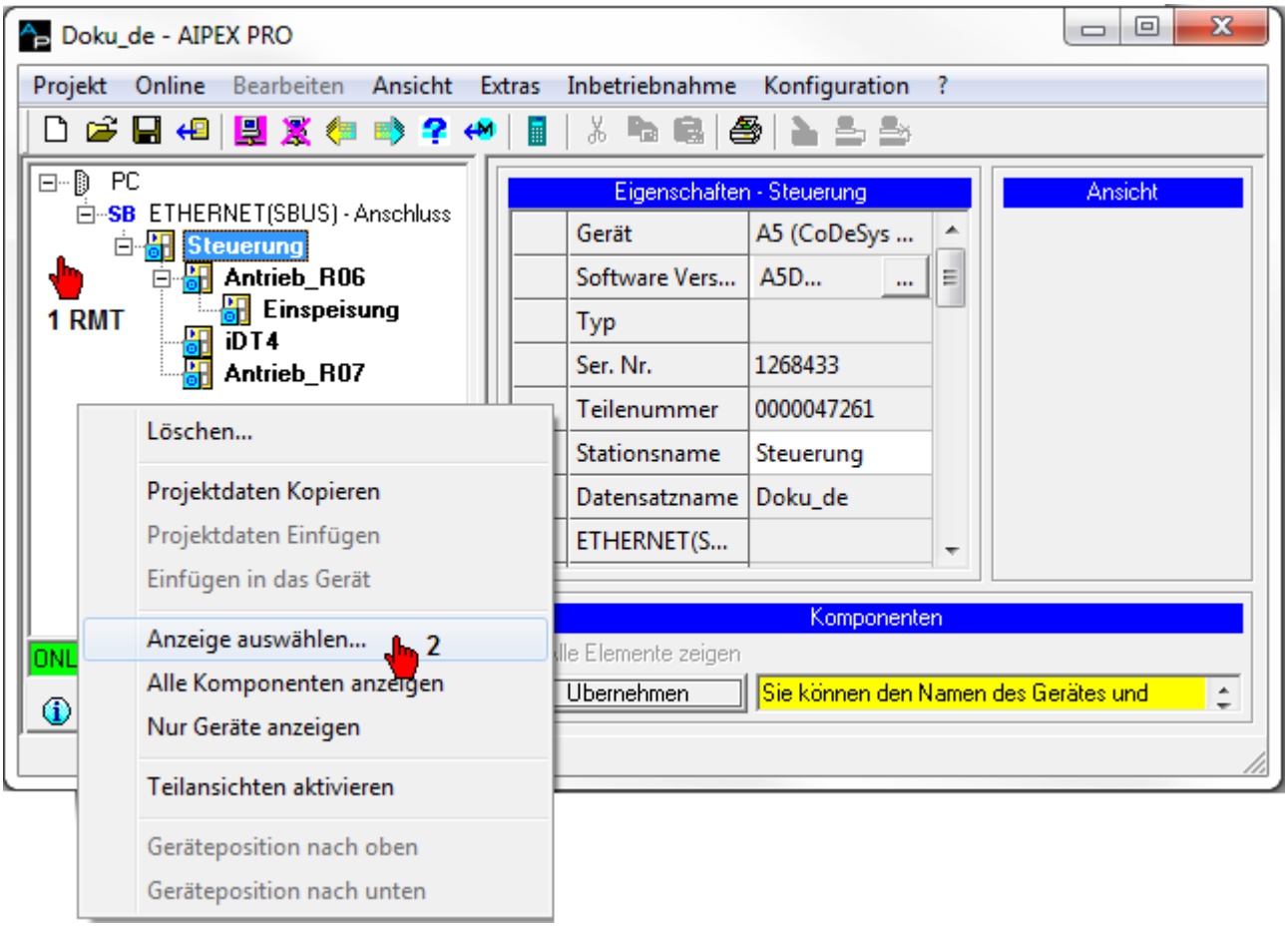
3.3.1 Programmübersicht - Symbole und Icons

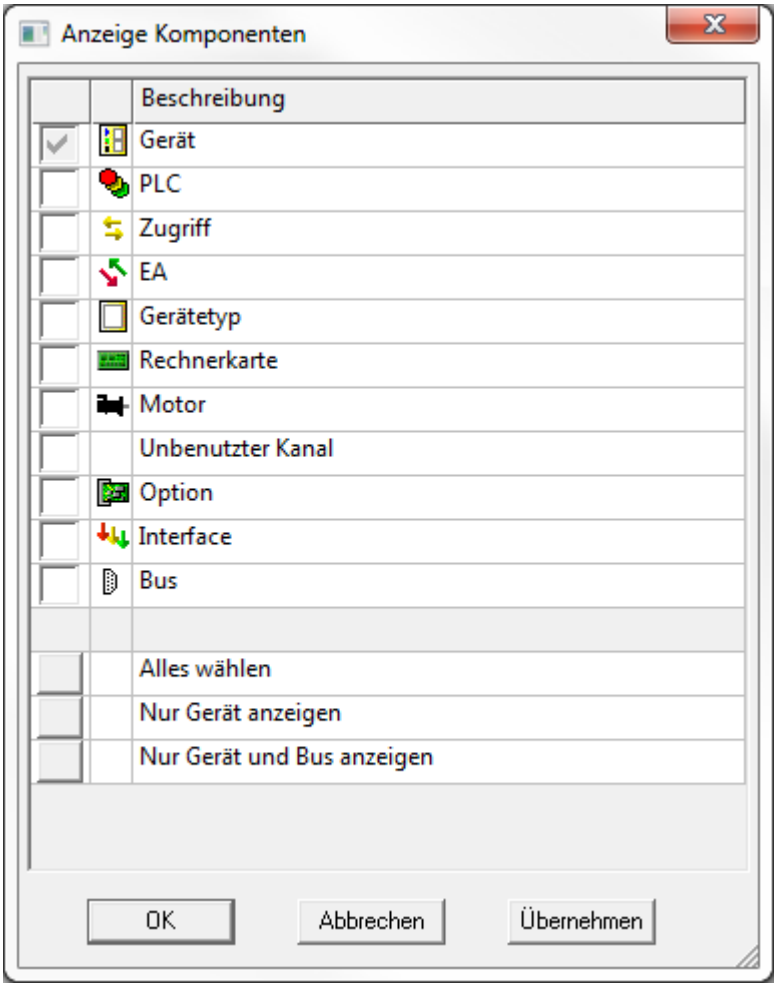
Symbol Icon	Bedeutung
	Mit dem Hand-Symbol wird in Beispielen oder Screenshots angezeigt, an welche Stelle Sie mit der Maus KLICKEN müssen. RMT: Rechte Maus Taste
	Kommunikationsschnittstelle zwischen PC und Gerät aktiviert, physikalisch vorhanden und betriebsbereit
	Kommunikationsschnittstelle zwischen PC und Gerät aktiviert aber nicht betriebsbereit
	Das geöffnete Projekt wird geschlossen und ein neues Projekt wird angelegt Siehe 'Projekt' auf Seite 97.
	AIPEX PRO und AIPEX/AIPAR Datensätze können geöffnet werden Siehe 'Projekt' auf Seite 97.
	Speichern des AIPEX PRO Datensatzes Siehe 'Projekt' auf Seite 97.
	Drucken Siehe 'Projekt' auf Seite 97.

Symbol Icon	Bedeutung
	Offline Gerät Keine Verbindung zu einem physikalisch vorhandenen Gerät
	Online Gerät Mit Verbindung zu einem physikalisch vorhandenen Gerät
	Siehe 'Projekt' auf Seite 97.
	Siehe 'Online' auf Seite 100.
	Siehe 'Online' auf Seite 100.
	Die Parameter vom angewählten Gerät, werden in den Projektdatensatz übernommen
	Der Projektdatensatz wird in das angewählte Gerät überspielt
	Siehe 'Online' auf Seite 100.
	Siehe 'Online' auf Seite 100.
	Siehe 'Direktmode' auf Seite 127.
	Undo / Redo im Nachrichtenkonfigurator (gilt nicht für die automatische Konfiguration)
	Siehe 'Konfiguration' auf Seite 125.
	Siehe 'Konfiguration' auf Seite 125.
	Siehe 'Konfiguration' auf Seite 125.
	CODESYS Version 2
	CODESYS Version 3

3.3.2 Programmübersicht - Anzeigefilter

Gestalten Sie den 'Gerätebaum' übersichtlicher, indem Sie den Anzeigefilter auf Ihre Applikation anpassen. Klicken Sie dazu mit der rechten Maustaste in den Gerätebaum.





Für jeden Reiter **'Konfiguration'**, **'Parameter'**, **'Nachrichten'**, **'Scope'** und **'Diagnose'** kann ein separater Anzeigefilter eingestellt werden.

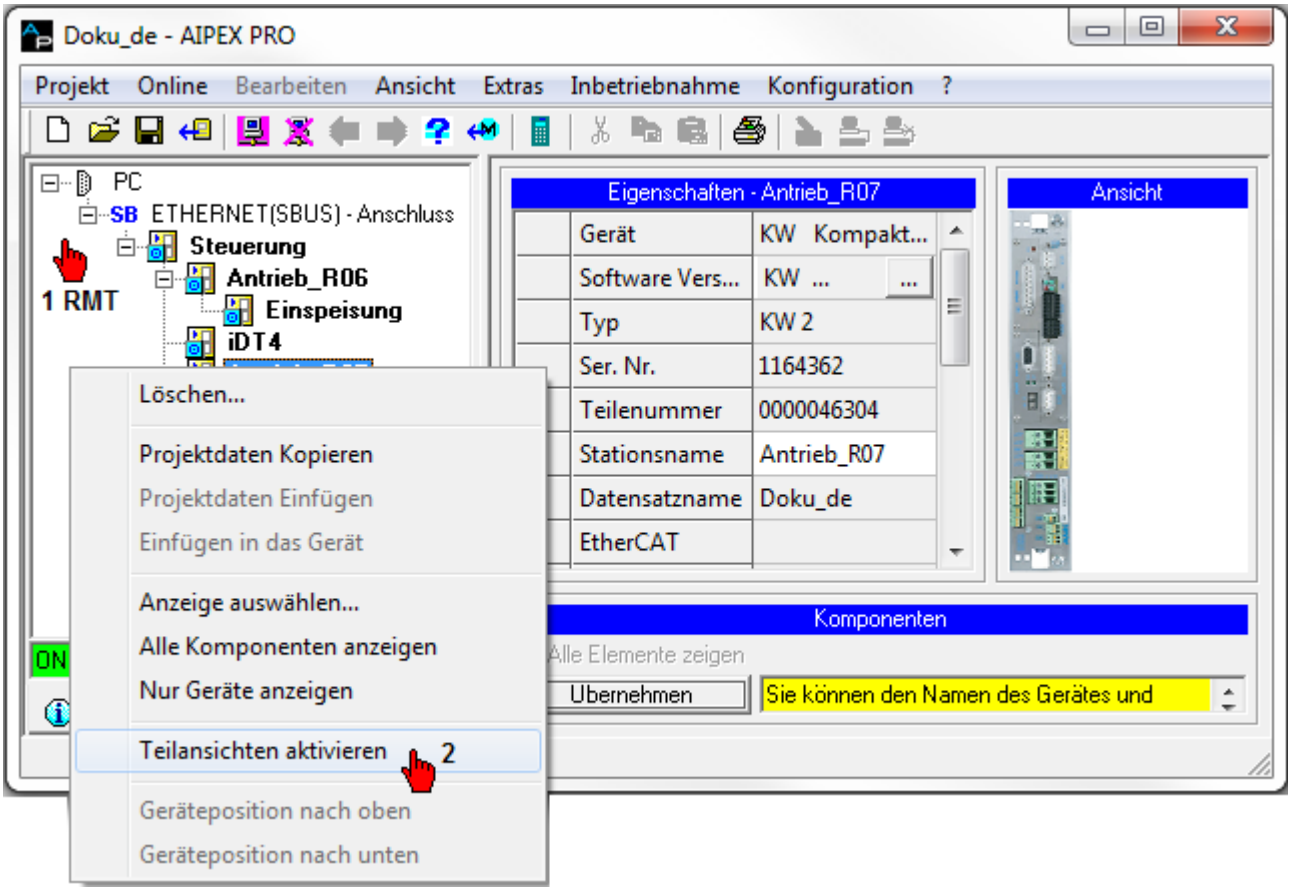
Name	Bemerkung
Gerät	Anzeige des Stationsnamen
PLC	Anzeige der PLC Baugruppe
EA	Anzeige der binären Ein- und Ausgänge
Gerätetyp	Anzeige des Gerätetyps
Rechnerkarte	Anzeige der Rechnerkarte
Motor	Eigenschaften des Motors
Unbenutzter Kanal	Anzeige von ungenutzten Schnittstellen
Option	Anzeige von Optionskarten
Interface	Schnittstelle zwischen AIPEX PRO und CoDeSys
Bus	AMK Bus (z.B. ACC-Bus, CAN-S, EtherCAT...)

3.3.3 Programmübersicht - Teilansichten

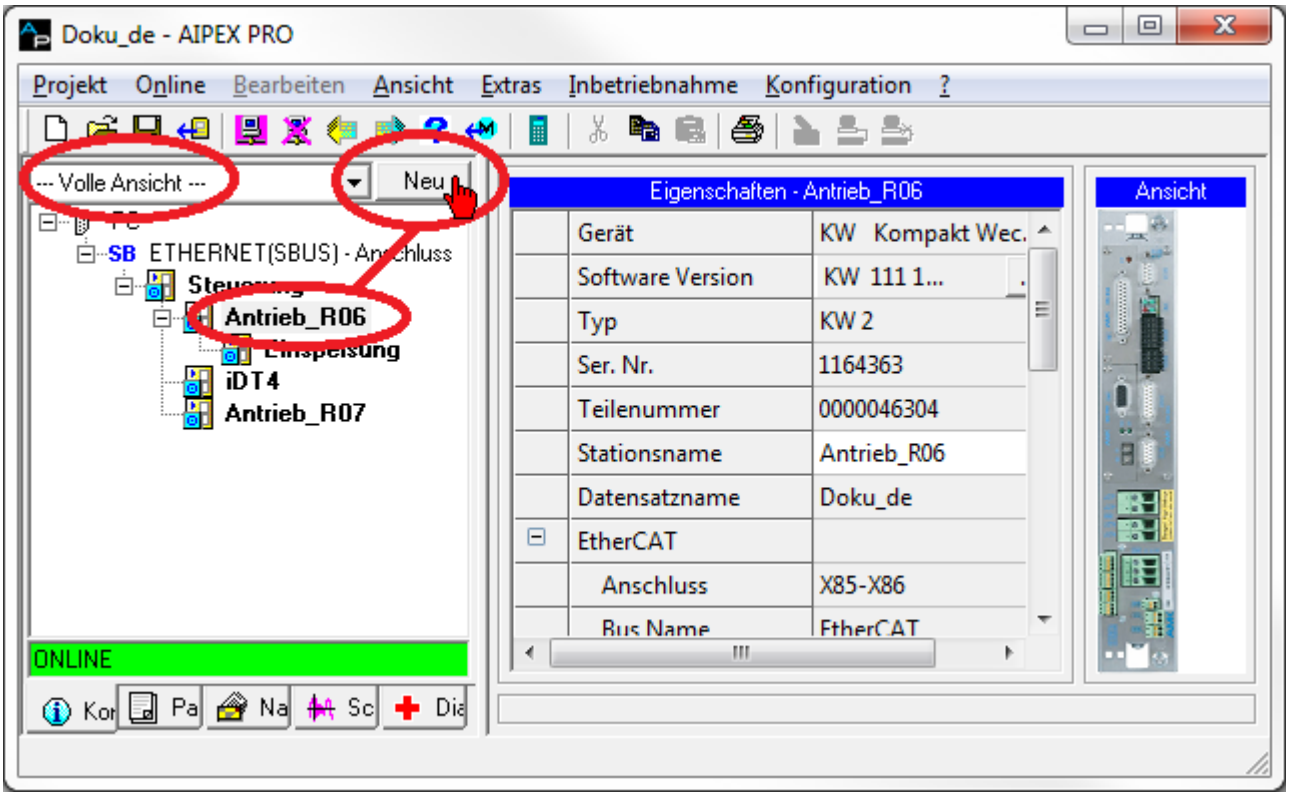
Mit der Funktion **'Teilansicht aktivieren'** können Teile des kompletten Gerätebaums ausgeblendet werden. Wählen Sie aus, welche Module angezeigt werden. Durch das Ausblenden einzelner Module steigt die Übersichtlichkeit. Zudem wird der Daten- und Gerätezugriff durch die ausgeblendeten Module beschleunigt.

Die Teilansicht befindet sich zwischen Startpunkt und Endpunkt. Der Startpunkt ist das angewählte Modul bei Betätigen der Taste **'Neu'**. Der Endpunkt wird im sich öffnenden Dialogfeld **'Teilansicht erstellen'** gesetzt.

Sie aktivieren die Teilansicht, indem Sie mit der rechten Maustaste in den Gerätebaum klicken.
Klicken Sie anschließend im Dialogfeld auf **'Teilansicht aktivieren'**.

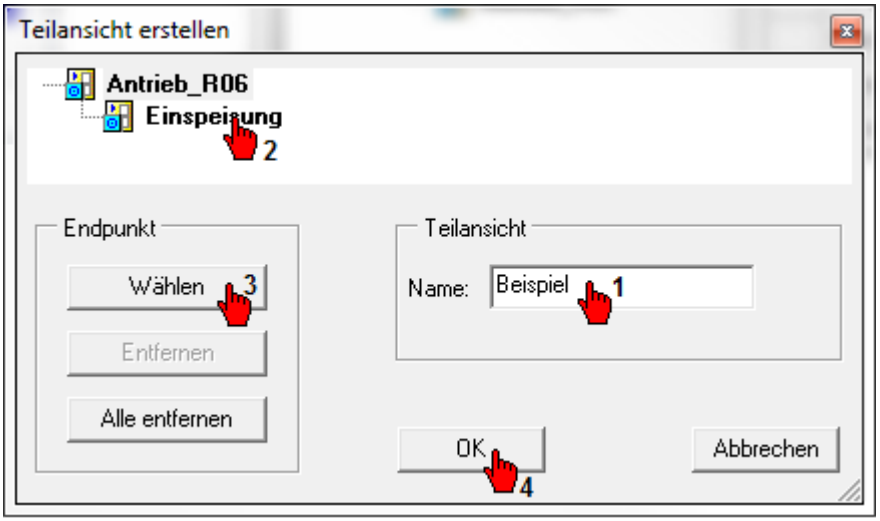


In der Auswahlbox können Sie zwischen der **'Vollen Ansicht'** und (sofern vorhanden) den anwenderspezifischen Teilansichten wählen. Anwenderspezifische Teilansichten werden durch anwählen des Startpunktes und anschließendem anklicken der Taste **'Neu'** erstellt.

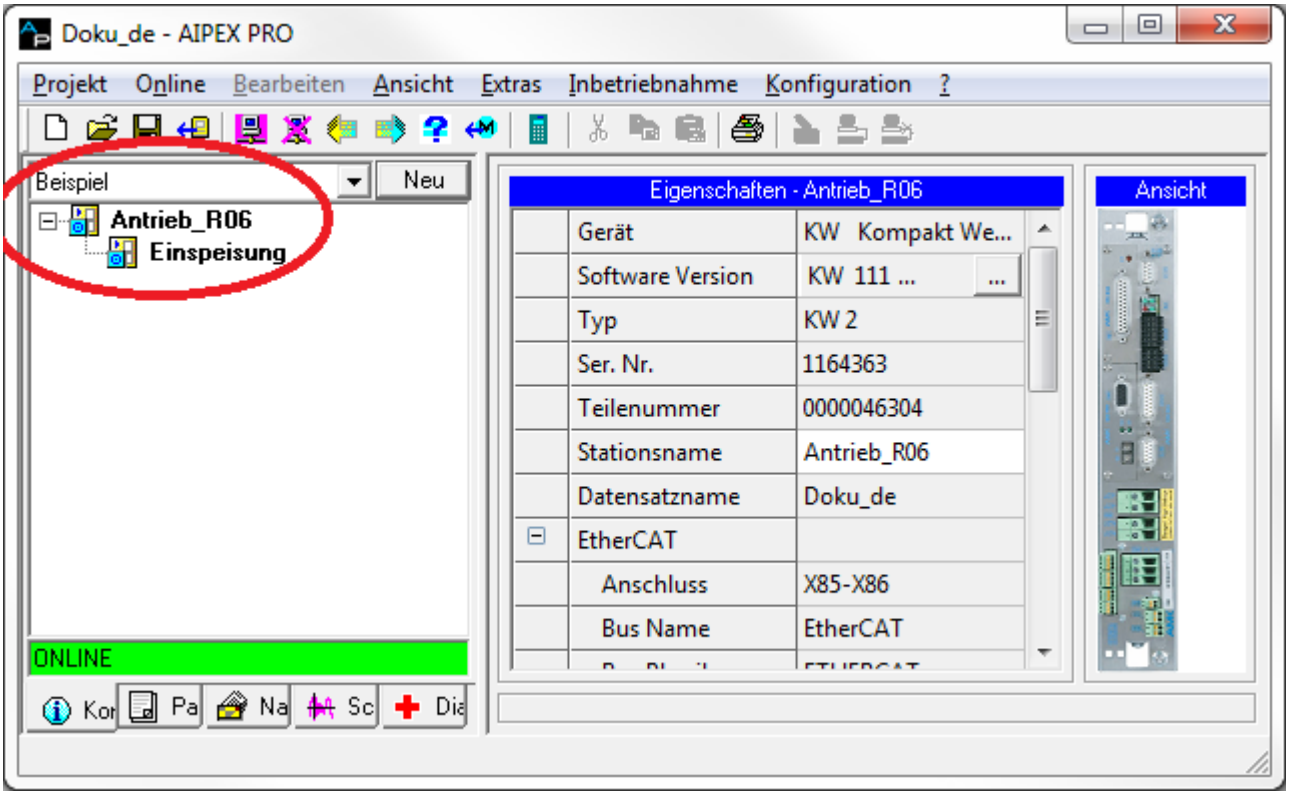


Teilansicht erstellen:

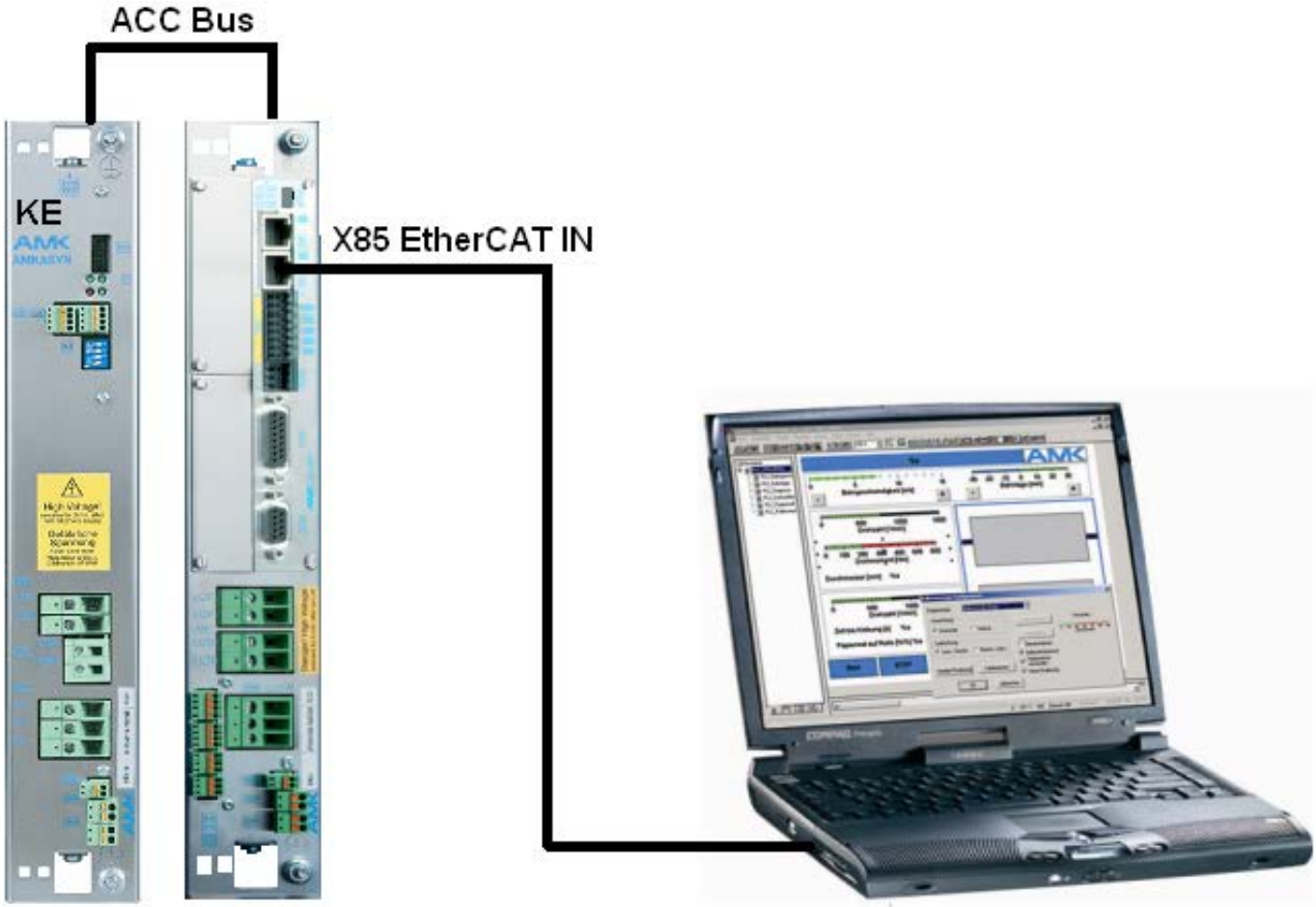
Vergeben Sie einen Namen, wählen Sie anschließend ihren Endpunkt (Modul) an. Durch Drücken der Taste Setzen wird der Endpunkt übernommen.



Teilansicht Beispiel: Startpunkt Modul Antrieb_R06, Endpunkt Modul Einspeisung.



3.4 Kommunikation PC - AMK Gerät



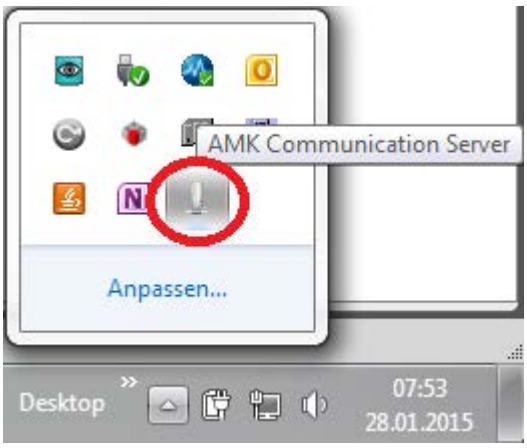
Siehe 'CAN Schnittstelle' auf Seite 46.
Siehe 'COM Schnittstelle' auf Seite 48.
Siehe 'EtherCAT Schnittstelle' auf Seite 54.
Siehe 'Ethernet Schnittstelle' auf Seite 57.
Siehe 'SERCOS III Schnittstelle' auf Seite 66.
Siehe 'USB Schnittstelle' auf Seite 69.
Siehe 'AIPEX PRO add in gateway for TwinCAT V2' auf Seite 69.
Siehe 'Kommunikation testen' auf Seite 69.

3.4.1 Kommunikationsserver

Der Kommunikationsserver (AMK Communication Server) wird dazu verwendet, eine Verbindung (Ethernet, EtherCAT, CANCLIENT und USB) zu den Geräten herzustellen. Mit dem Kommunikationsserver kann von mehreren Programmen (AIPEX PRO, AIPEX LITE, ATF,...) gleichzeitig auf die Geräte zugegriffen werden. Dadurch kann parallel zu AIPEX PRO z. B. auch das Erstinbetriebnahmetool 'AIPEX Startup' ausgeführt werden.

Es ist nicht möglich mehrere Instanzen der gleichen Software gleichzeitig zu betreiben!

Das Programm wird automatisch bei Aufruf von AIPEX PRO, AIPEX LITE, ATF,.. gestartet und auch wieder beendet.
Der gestartete 'AMK Communication Server' ist erkennbar in der Windows Taskleiste.

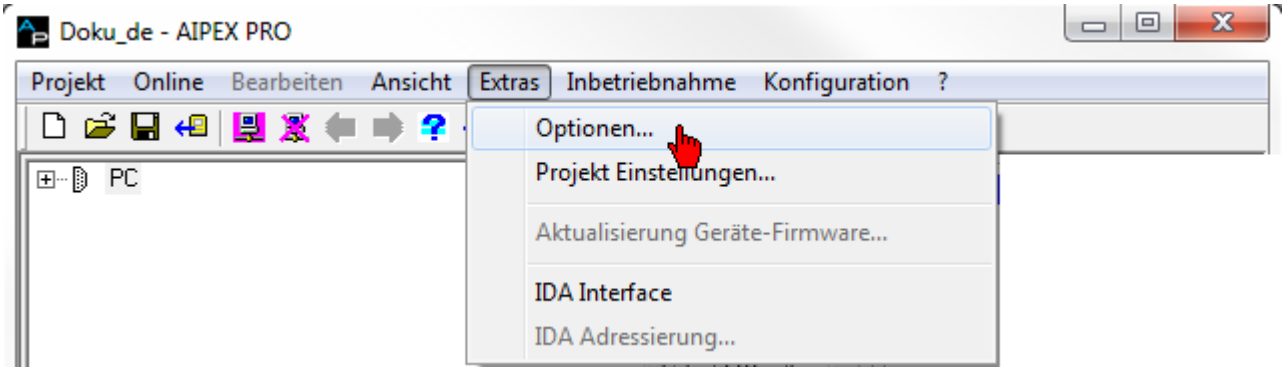


3.4.1.1 Kommunikationsserver Einstellungen

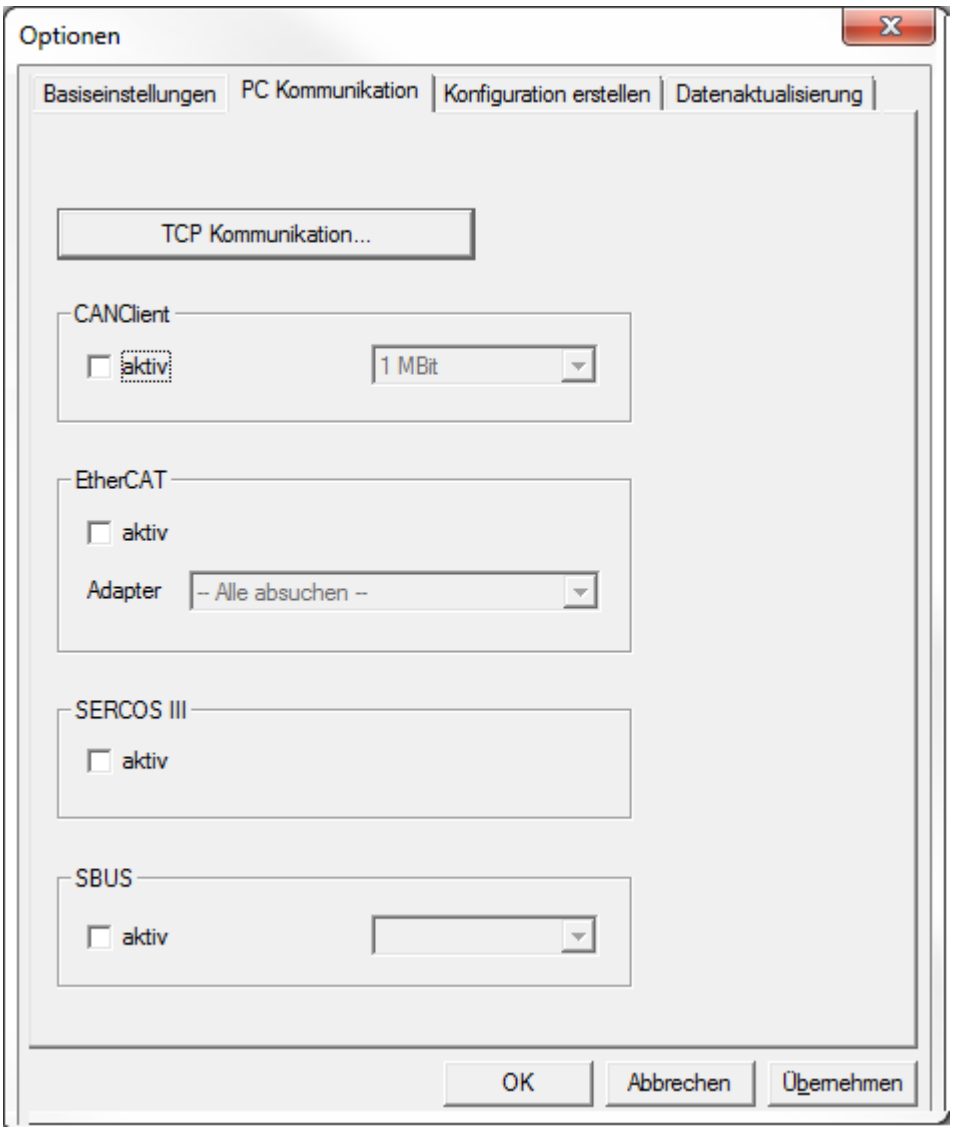
Folgende Bussysteme werden vom Kommunikationsserver unterstützt (Hardwarekomponenten und Treiber müssen installiert sein):

- Ethernet ist immer aktiv
- USB ist immer aktiv
- CANCLIENT muss in AIPEX PRO aktiviert werden
- EtherCAT muss in AIPEX PRO aktiviert werden

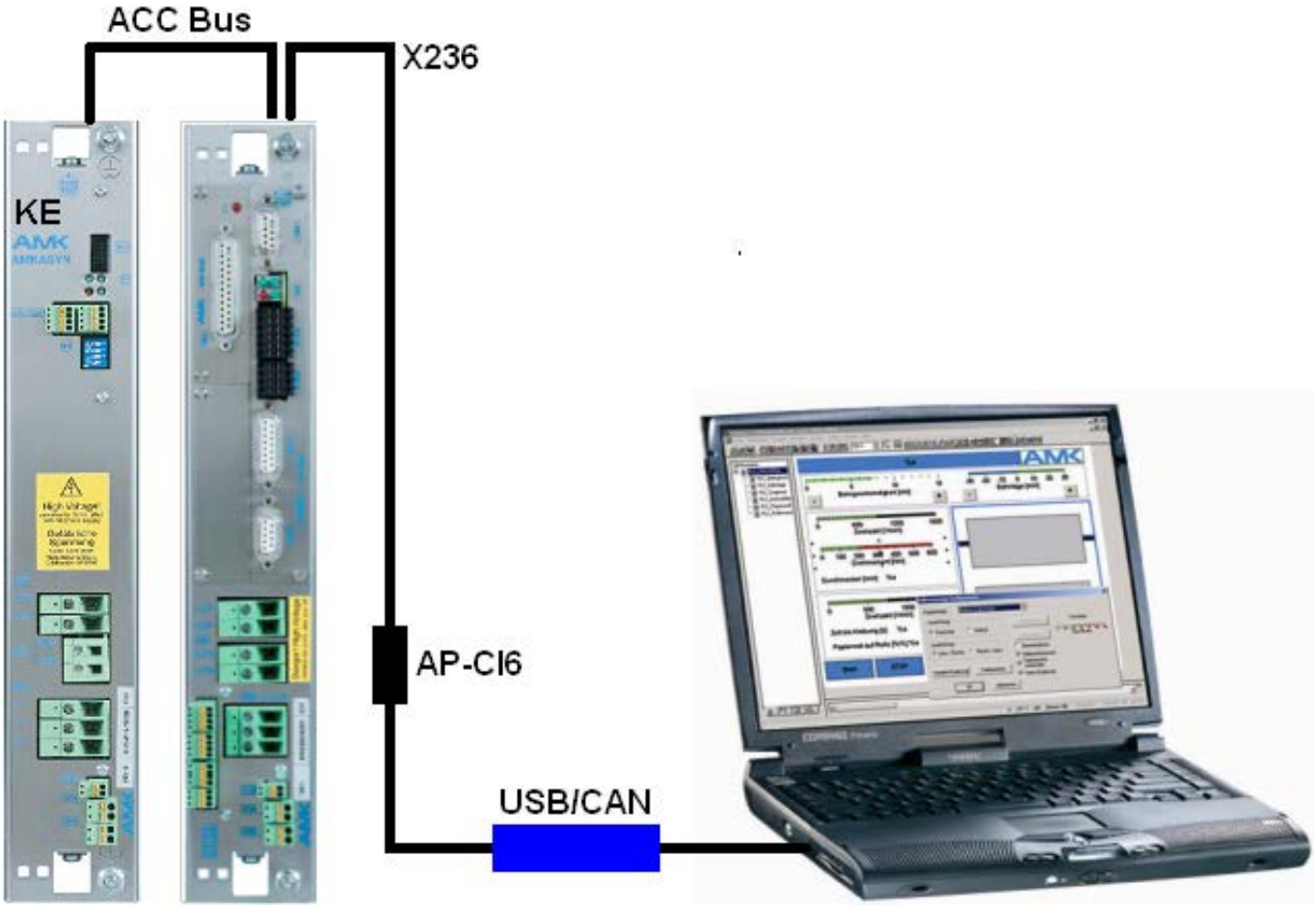
In AIPEX PRO muss die Schnittstelle aktiviert werden:



Im Reiter 'PC Kommunikation' kann die Schnittstelle aktiviert oder deaktiviert werden.


Optionen
Basiseinstellungen
PC Kommunikation
Konfiguration erstellen
Datenaktualisierung
TCP Kommunikation...
CANClient
aktiv
1 MBit
EtherCAT
aktiv
Adapter
-- Alle absuchen --
SERCOS III
aktiv
SBUS
aktiv
OK
Abbrechen
Übernehmen

3.4.2 CAN Schnittstelle

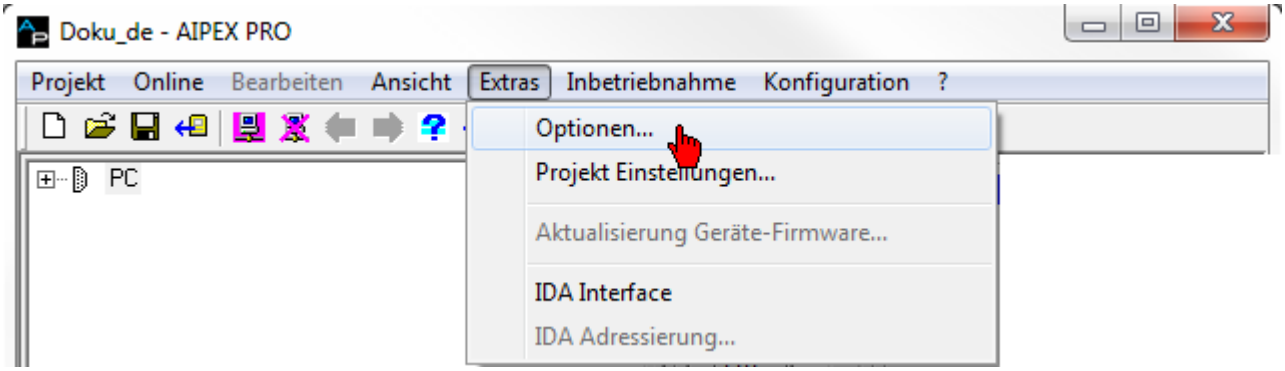


Beispiel:
KE mit ACC Bus und KW mit KW-R03 (kompatibel mit KU-/KW-R03(P) und KU-/KW-R04).

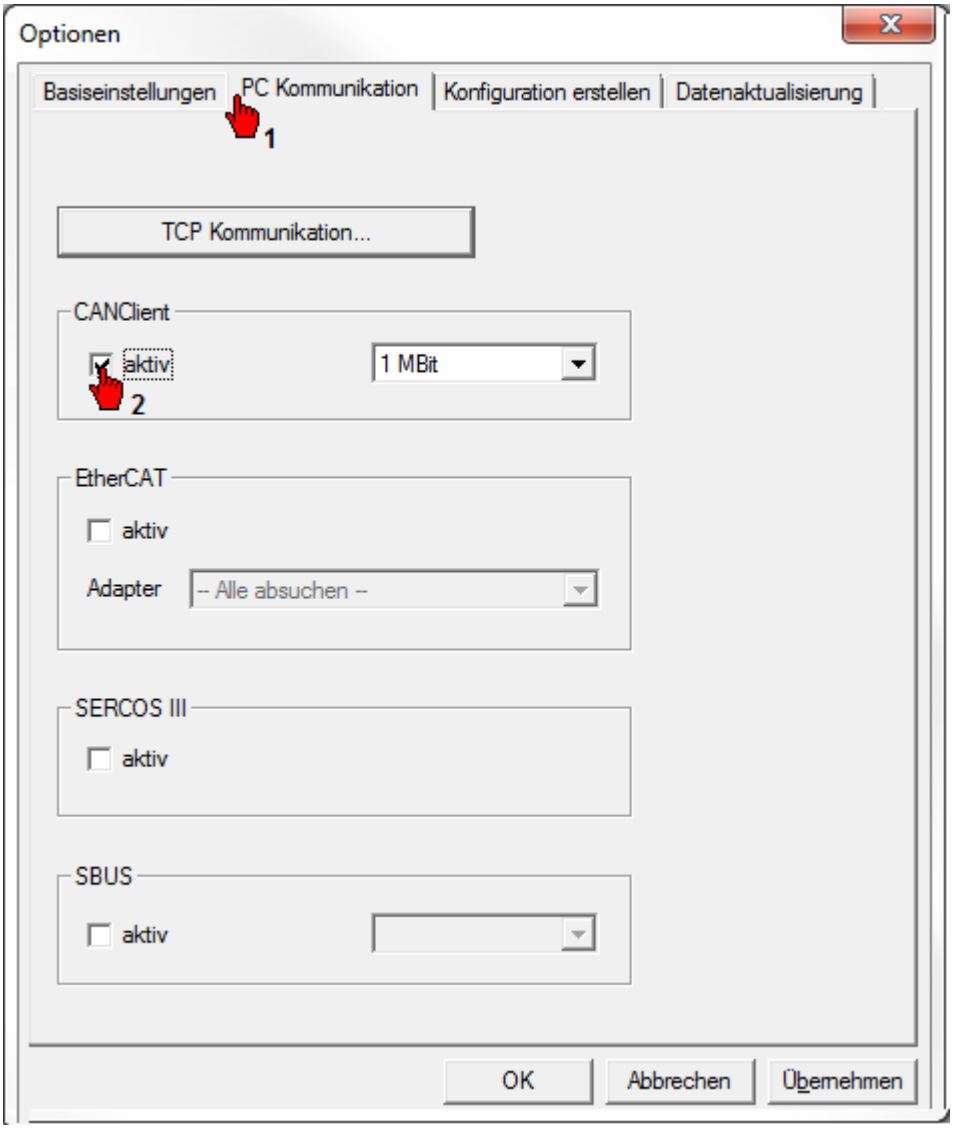
Konverter:
AMK USB-CAN Konverter AMK Teile-Nr. O755. (Enthält Anschlusskabel und Abschlusswiderstände für die 'KE/KW' Baureihe sowie für die 'Dezentrale Antriebstechnik').

Gültig für die ACC Bus Geräte
Bei einem CAN Netzwerk ohne NMT Master (CAN/ACC Bus Master) muss folgendes Bit gesetzt sein:
ID34026 'BUS Modusattribut' Bit 11 = 1

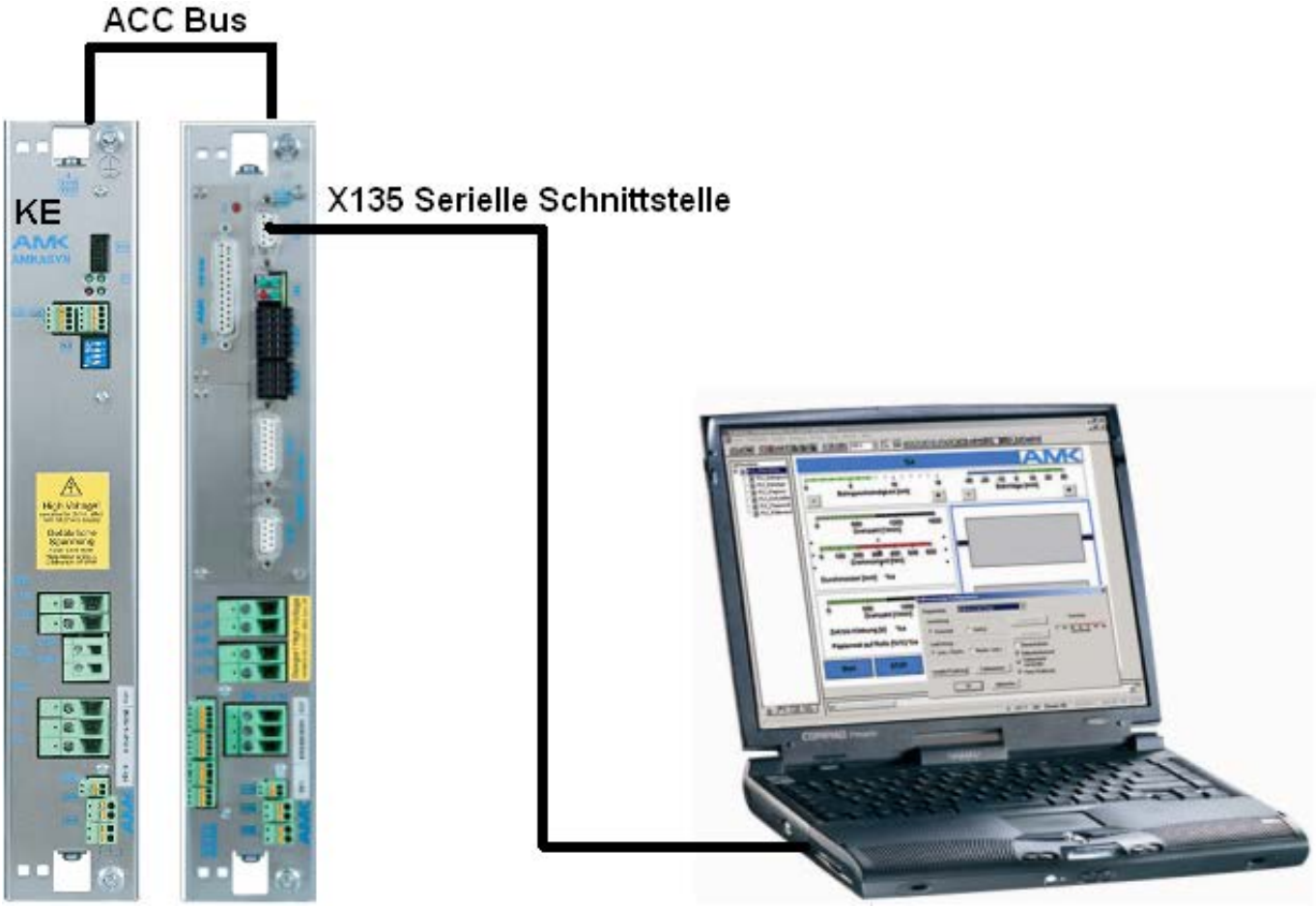
3.4.2.1 CAN Kommunikationseinstellungen



Der IXXAT Treiber muss installiert sein.
Der USB-CAN Konverter (IXXAT Gerät) muss mit dem PC verbunden sein.



3.4.3 COM Schnittstelle



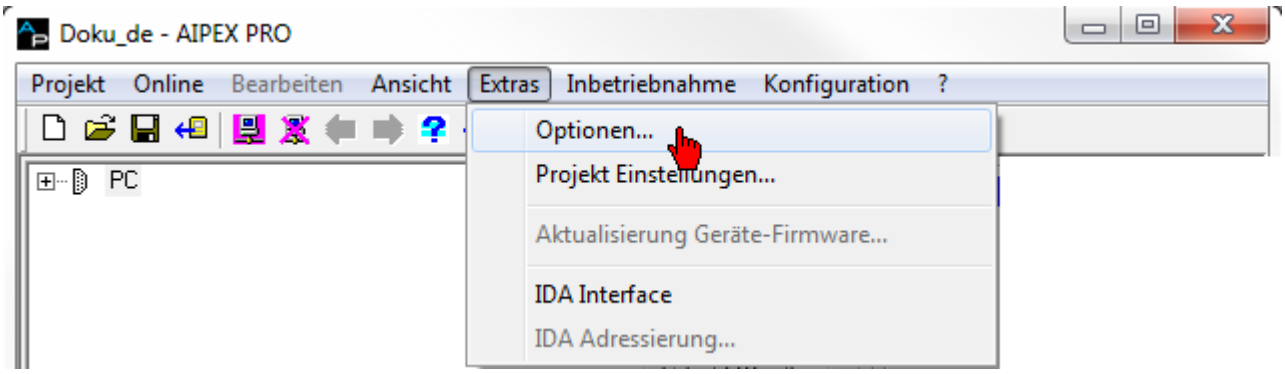
Beispiel:

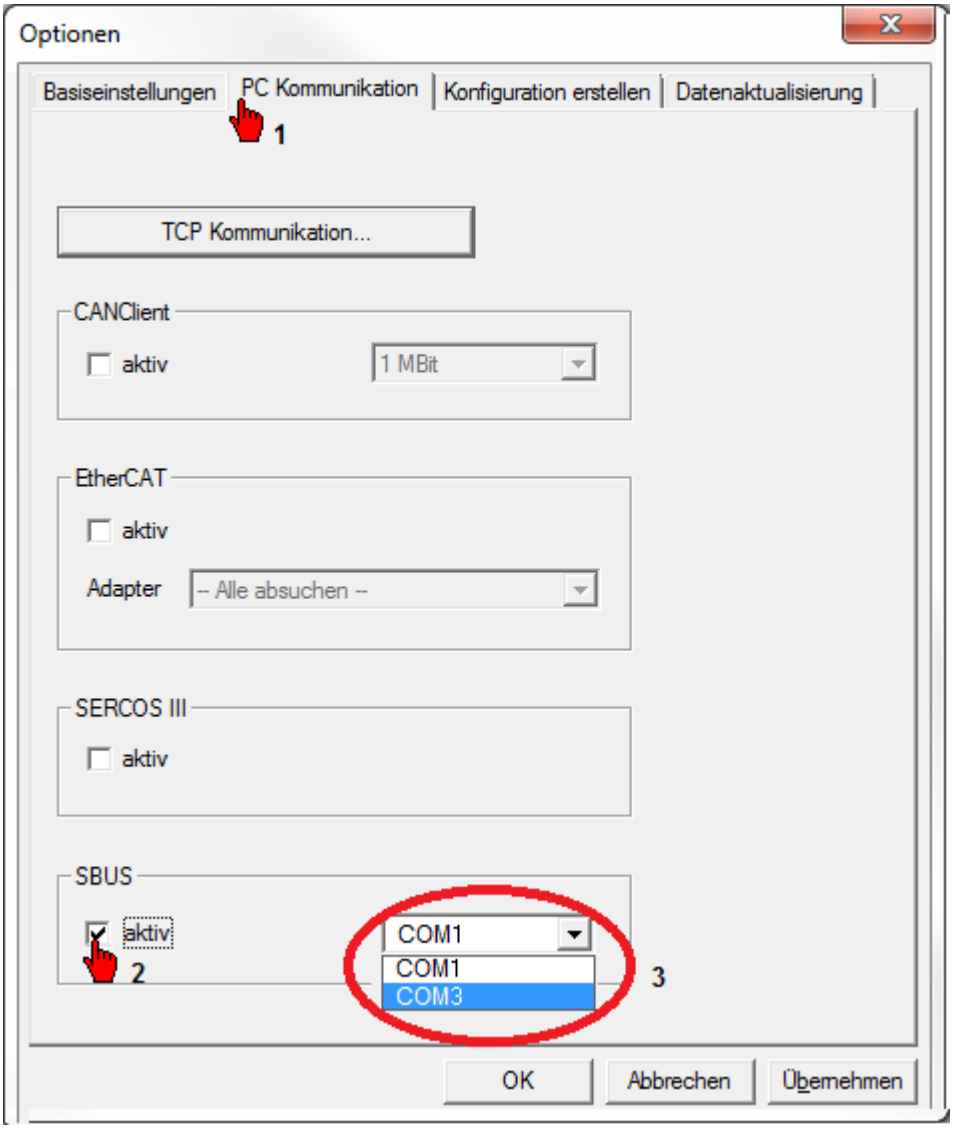
KE mit ACC Bus und KW mit KW-R03 (kompatibel mit KU-/KW-R03(P) und KU-/KW-R04). Es besteht die Möglichkeit über ACC Bus die angeschlossenen Geräte zu erreichen. In diesem Fall muss die Reglerkarte als ACC Bus Master definiert sein.

Kabel:

- AMK Serielles Datenkabel AMK Teile-Nr. O576
- AMK USB-RS232 Konverter AMK Teile-Nr. 200770

3.4.3.1 COM Kommunikationseinstellungen

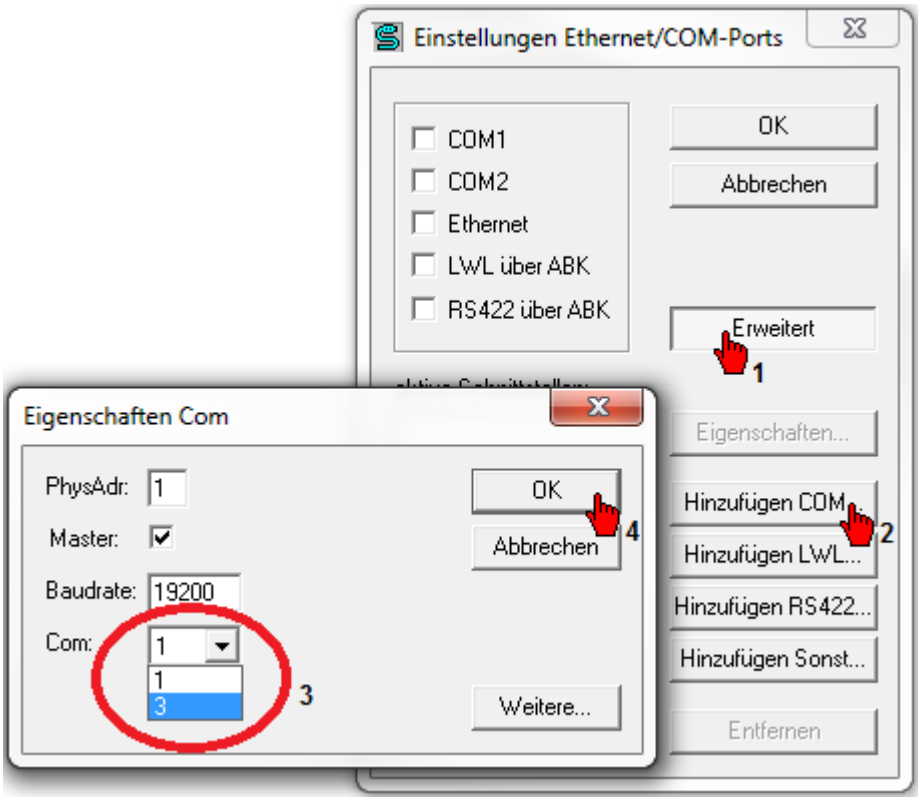




Stellen Sie nur die gewünschte Schnittstelle ein.

Wenn Sie eine andere Schnittstelle benötigen als COM1 oder COM2, können Sie diese im 'SbusRegister' ('Start' → 'Alle Programme' → 'AMK' → 'SbusRegister')durch klicken auf den Taster **'Erweitert'** → **'Hinzufügen COM'** einstellen.

Die vorgegebenen Eigenschaften dürfen nur in Verbindung mit der AZ Optionskarte CNC oder bei einer Modemverbindung geändert werden.



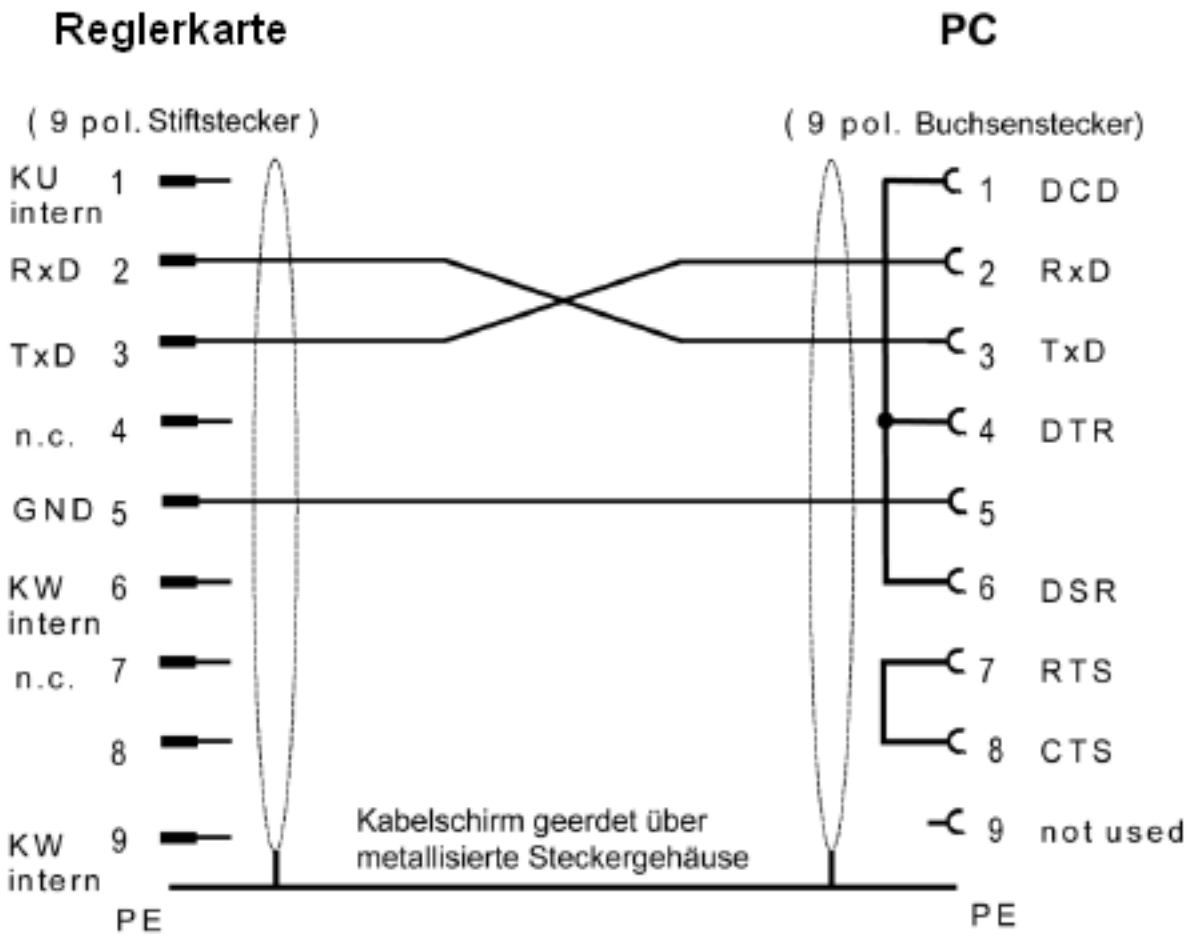
Um die Kommunikation zu testen, verbindet man ein Gerät mit dem PC und startet AIPEX PRO erneut.

Der Wechsel der Farbe des Symbols von rot nach grün am Rand rechts unten zeigt an, dass die Kommunikation störungsfrei läuft.

Ein grün/rotes Symbol steht für mehr als eine aktive Schnittstelle.

3.4.3.2 Serielles AMK Verbindungskabel RS232 (PC-AMK)

Das serielle AMK Verbindungskabel RS232, AMK Teile-Nr. O576 verbindet die serielle Schnittstelle der AMK KU-/ KW-Geräte mit der seriellen Schnittstelle des PCs.

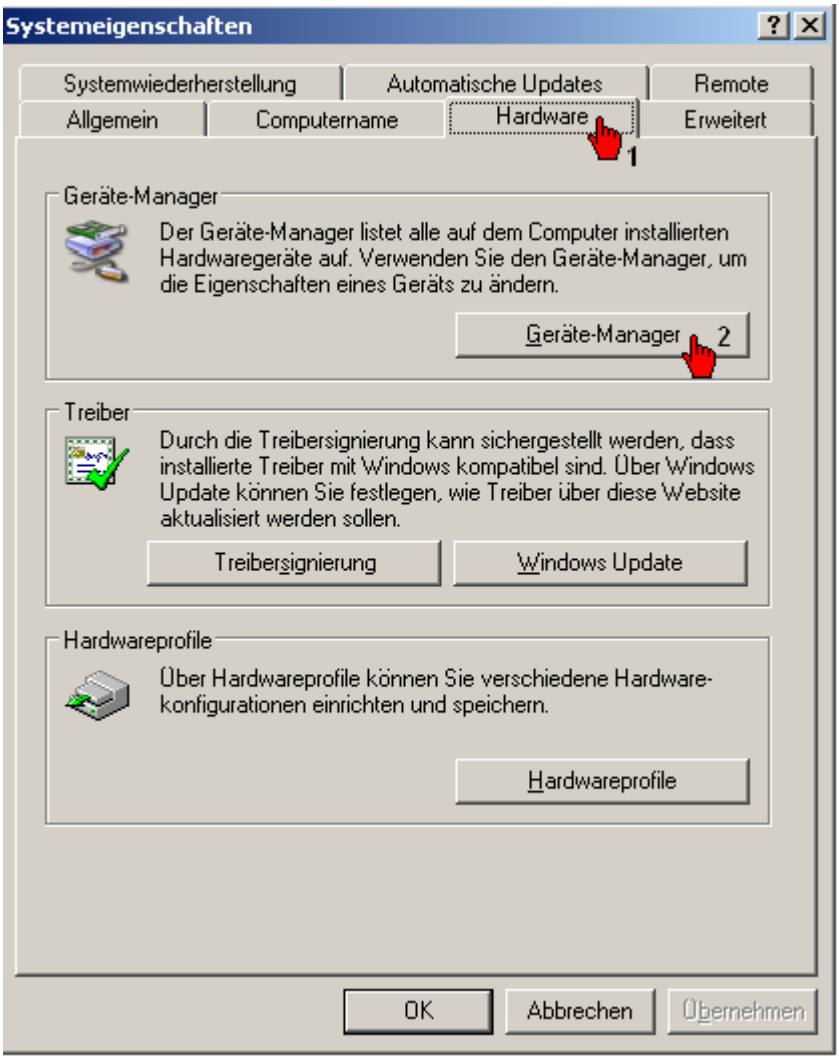


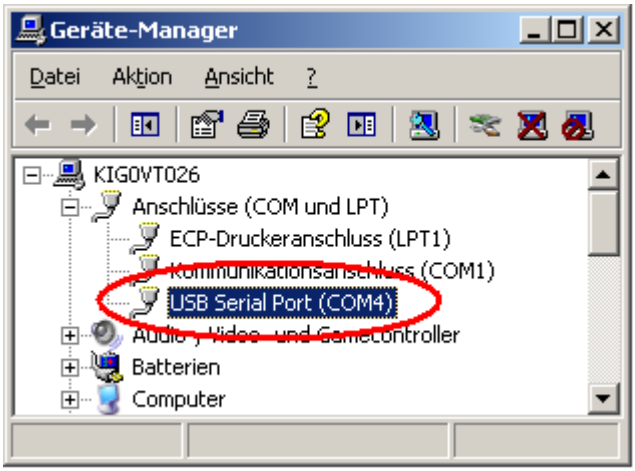
3.4.3.3 USB-RS232 Konverter

Der AMK USB-RS232 Konverter AMK Teile-Nr. 200770 verbindet die serielle Schnittstelle der AMK KU-/ KW-Geräte mit der USB Schnittstelle des PCs.

Beispiel Windows XP

Bei der Installation eines USB-RS232 Konverter wird eine zusätzliche virtuelle COM Schnittstellen vom PC generiert. Die Schnittstellennummer kann unter 'Windows' → 'Systemeigenschaften' → 'Hardware' → 'Geräte-Manager' → 'Anschlüsse (COM und LPT)' ausgelesen werden.

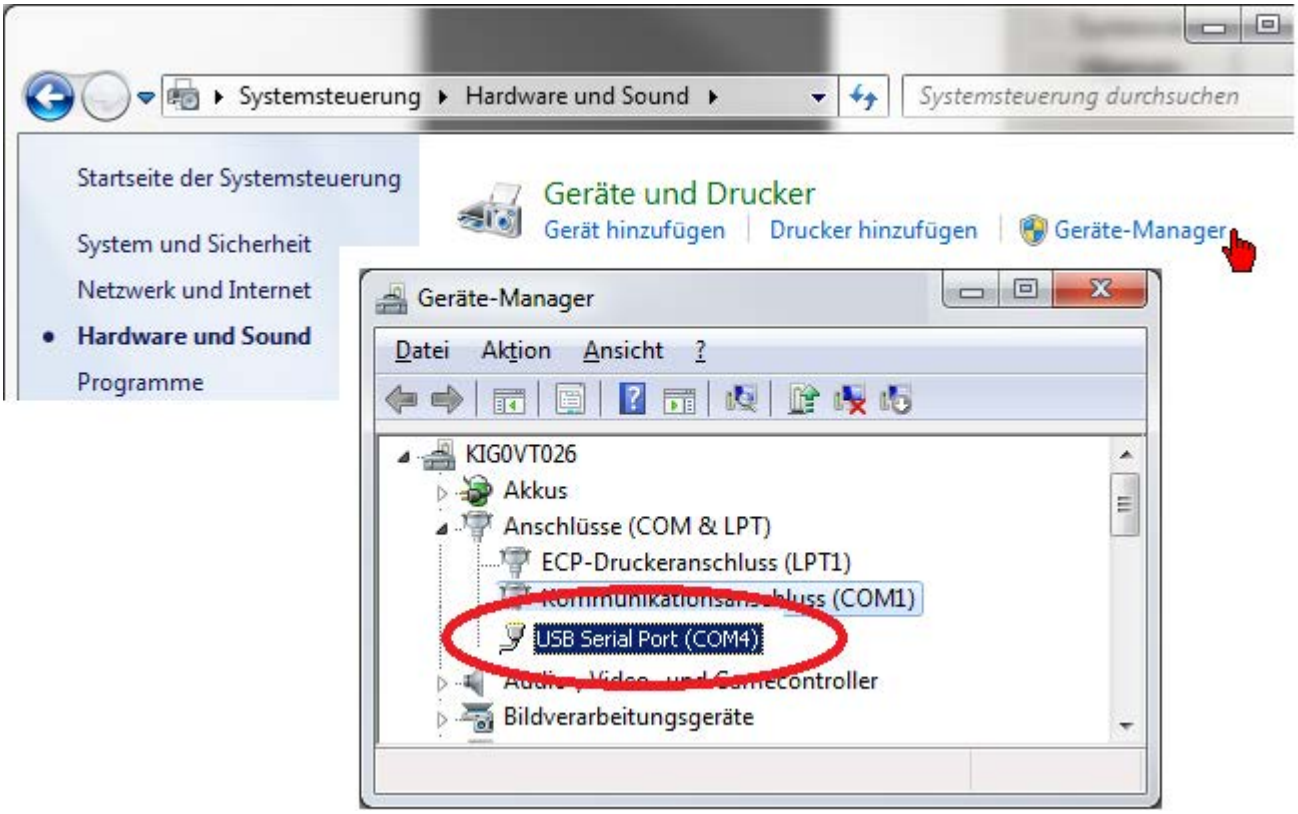




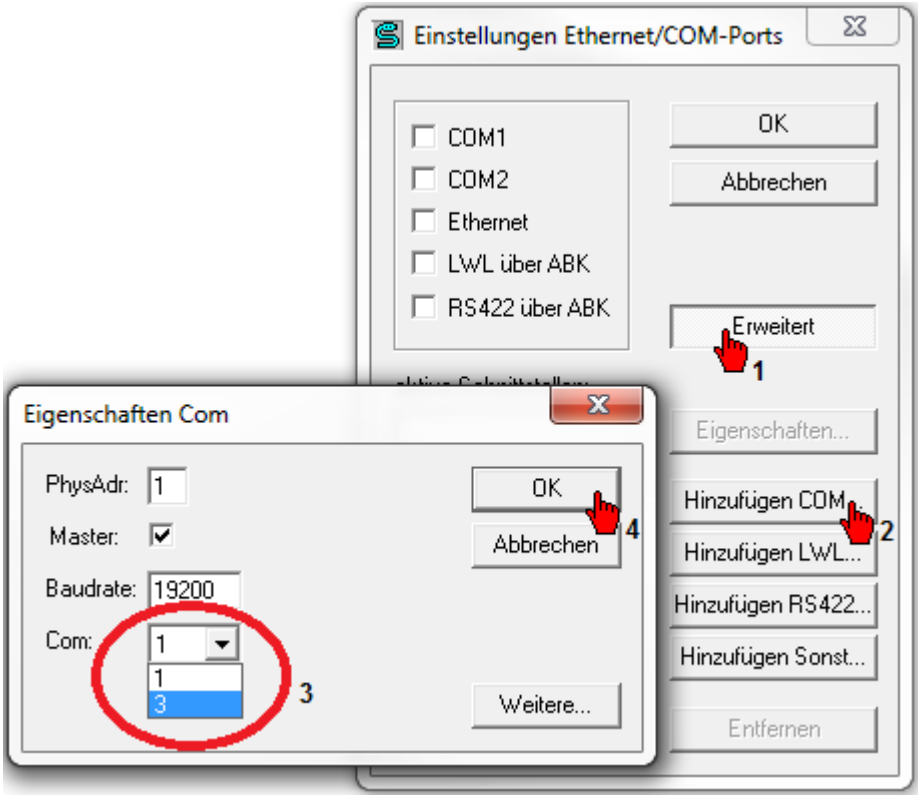
Beispiel Windows 7

Öffnen Sie 'Systemsteuerung' → 'Hardware und Sound'.

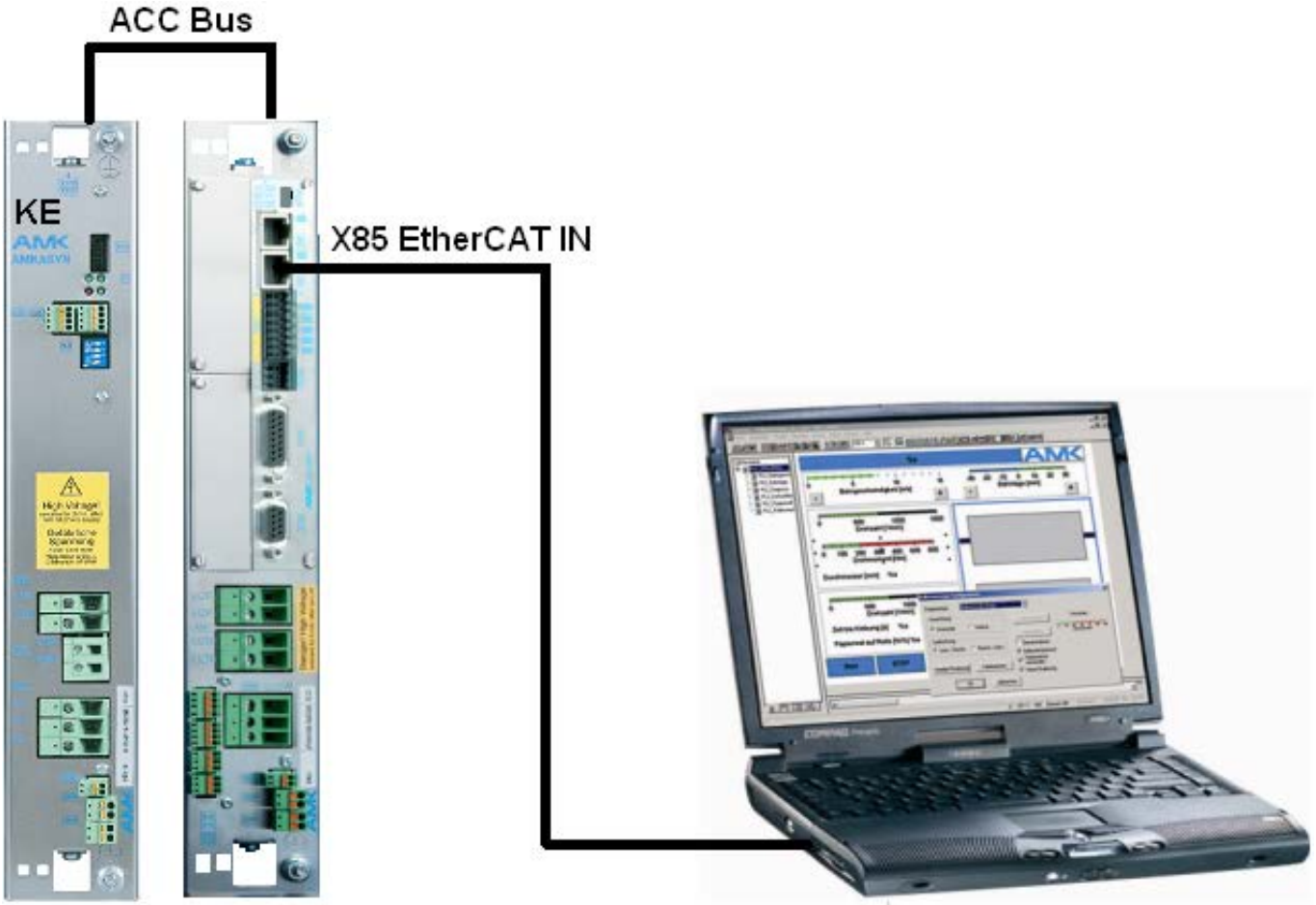
Öffnen Sie den 'Geräte-Manager'.



Tragen Sie die Schnittstellennummer (in diesem Beispiel COM3) im 'SbusRegister' ein ('Start' → 'Alle Programme' → 'AMK' → 'SbusRegister'). Verwenden Sie dazu die Taste 'Erweitert' und die Taste 'Hinzufügen COM'.



3.4.4 EtherCAT Schnittstelle



Beispiel:

KE mit ACC Bus und KW mit KW-R06 (kompatibel mit KW-R07, KW-R16 und KW-R17).

Über die EtherCAT Verbindung kann auf alle angeschlossenen Teilnehmer zugegriffen werden. AIPEX PRO kann zudem über die ACC-Bus Verbindung der Reglerkarte auf ACC-Bus Slave Teilnehmer zugreifen.

Kabel:

Ethernet Standard RJ45 Twisted Pair Patch Kabel.

Für die Verbindung zu der AMK Geräteserie 'Dezentrale Antriebstechnik' wird ein Ethernet-Kabel mit M12 Stecker benötigt:

Siehe Gerätebeschreibung AMK Teile-Nr. 203445 Dezentrale Antriebstechnik iC / iX / iDT5, Kapitel: Kabel für EtherCAT Anschluss [X85] und [X86].

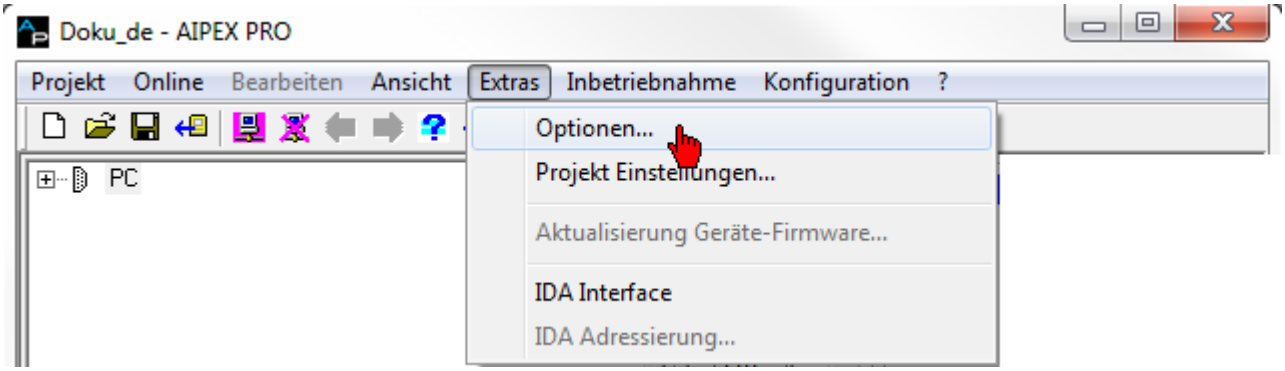
3.4.4.1 EtherCAT Kommunikationseinstellungen

EtherCAT Master:

Bei der Punkt-zu-Punkt Verbindung über EtherCAT übernimmt der PC die Aufgabe des EtherCAT Masters.
An den Ethernet Netzwerkeinstellungen des PCs müssen keine Änderungen vorgenommen werden.

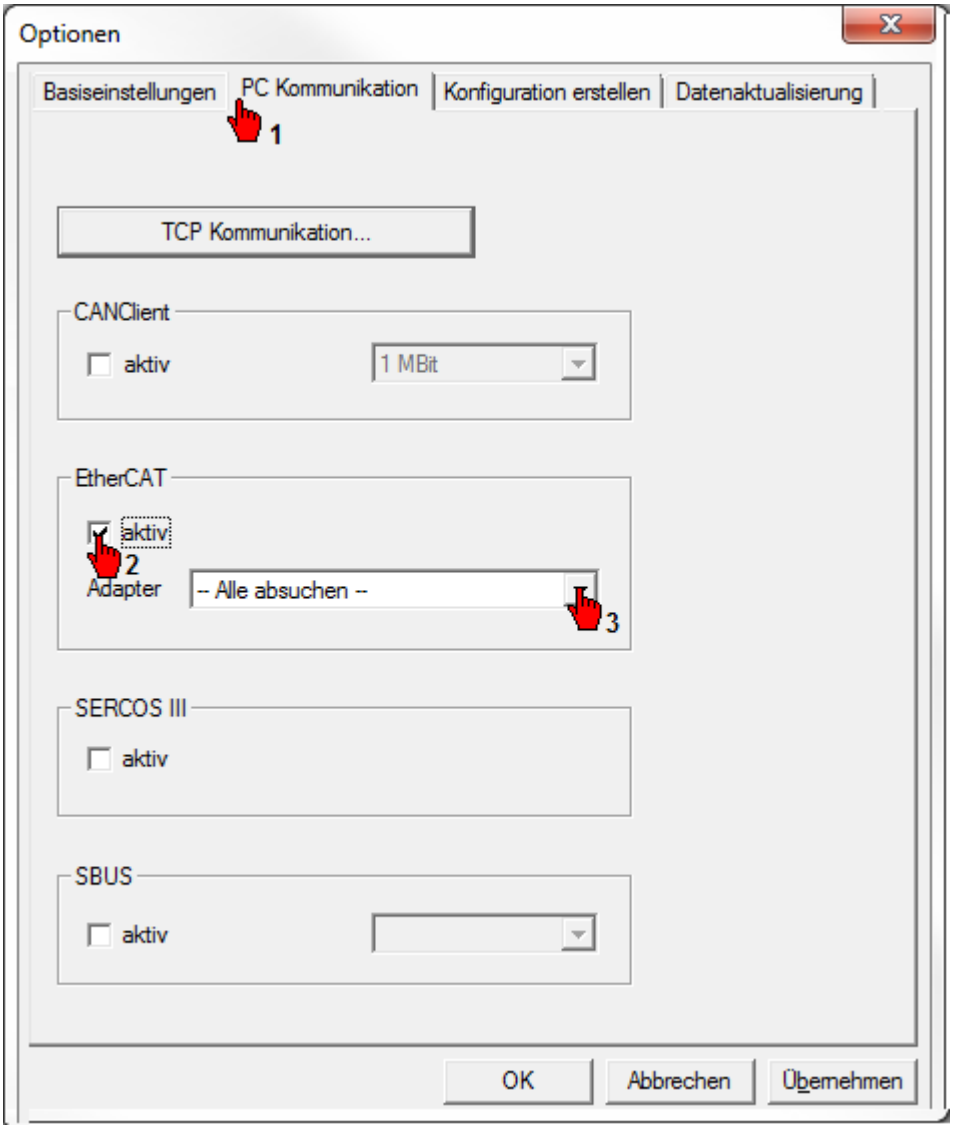
EtherCAT Slave:

Der angeschlossene Antrieb (Kompakteinspeisung oder Kompaktwechselrichter) ist der EtherCAT Slave.
In AIPEX PRO muss die EtherCAT Schnittstelle aktiviert werden:



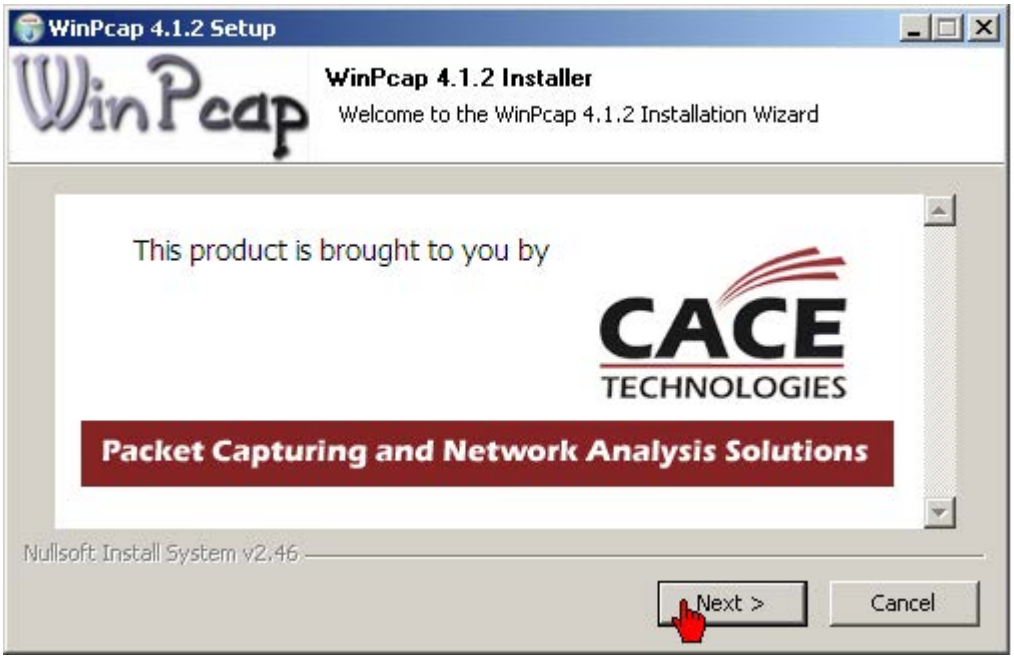
Im Reiter 'PC Kommunikation' kann die EtherCAT Kommunikation aktiviert und deaktiviert werden.

Sie können die Initialisierungsphase verkürzen, indem Sie den Ethernet Treiber ihrer verwendeten Ethernet Schnittstelle angeben.

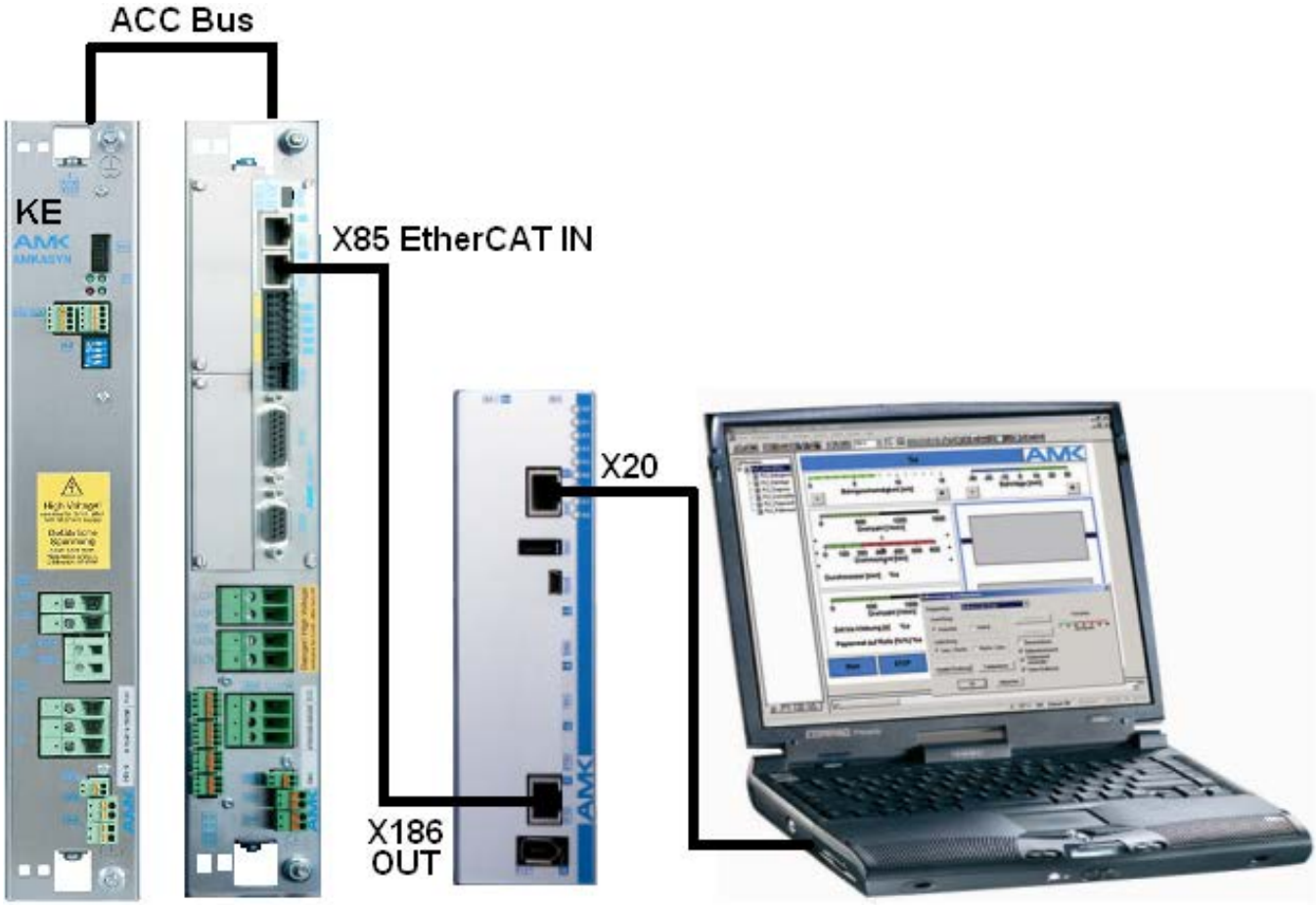


Voraussetzung:

Auf dem verwendeten PC muss das Programm WinPcap installiert sein.



3.4.5 Ethernet Schnittstelle



Beispiel:

Schaltschranksteuerung, KE mit ACC Bus und KW mit KW-R06 (kompatibel mit KW-R07, KW-R16 und KW-R17).

Über die Ethernet Verbindung kann auf alle angeschlossenen Module zugegriffen werden. AIPEX PRO kann zudem über die ACC-Bus Verbindung der Reglerkarte auf ACC-Bus Slave Teilnehmer zugreifen.

Kabel:

Ethernet Standard RJ45 Twisted Pair Patch Kabel.

Verwenden Sie ein RJ45 Crosskabel für die Verbindung, wenn Ihr PC nicht automatisch ein Twisted Pair Kabel kreuzt.

Bei einer aktiven PC Firewall sind folgende Freigaben erforderlich

- TCP Port 700
- TCP 50.001
- UDP Port 40.000
- Broadcast on

3.4.5.1 Ethernet Kommunikationseinstellungen

Die Ethernet Schnittstelle ist immer aktiv.

PC Einstellungen bei einer Direktverbindung

Bei einer Direktverbindung zwischen PC und Steuerung muss der PC eine feste IP Adresse in der gleichen Netzwerkklasse wie die Steuerung besitzen.

In der AMKAMAC Kompaktsteuerung ist die Standardadresse 192.168.0.1 hinterlegt. Diese Adresse gehört zum Netzadressbereich der Subnetzmaske 255.255.255.0 (Netzwerkklasse C).

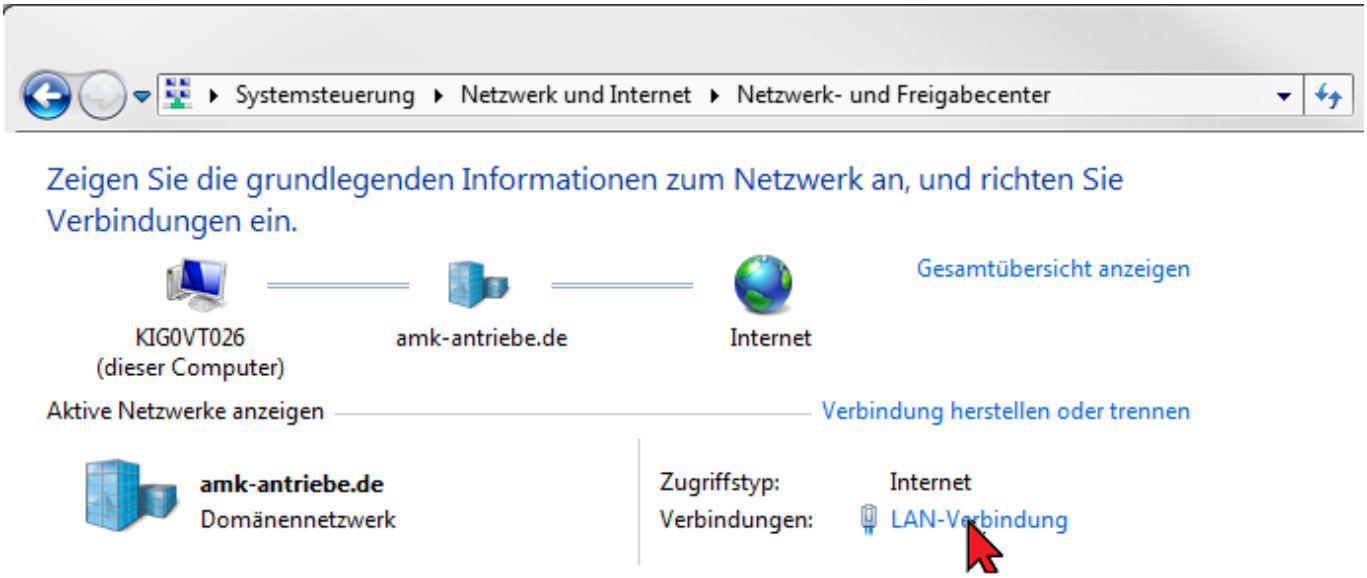
Nachfolgend wird beschrieben, wie Sie in Ihrem PC eine feste IP Adresse und die dazugehörige Subnetzmaske hinterlegen.

Wird die Steuerung und der PC an ein (Firmen-) Netzwerk angeschlossen, muss die IP Adresse der Steuerung an das Netzwerk angepasst werden.

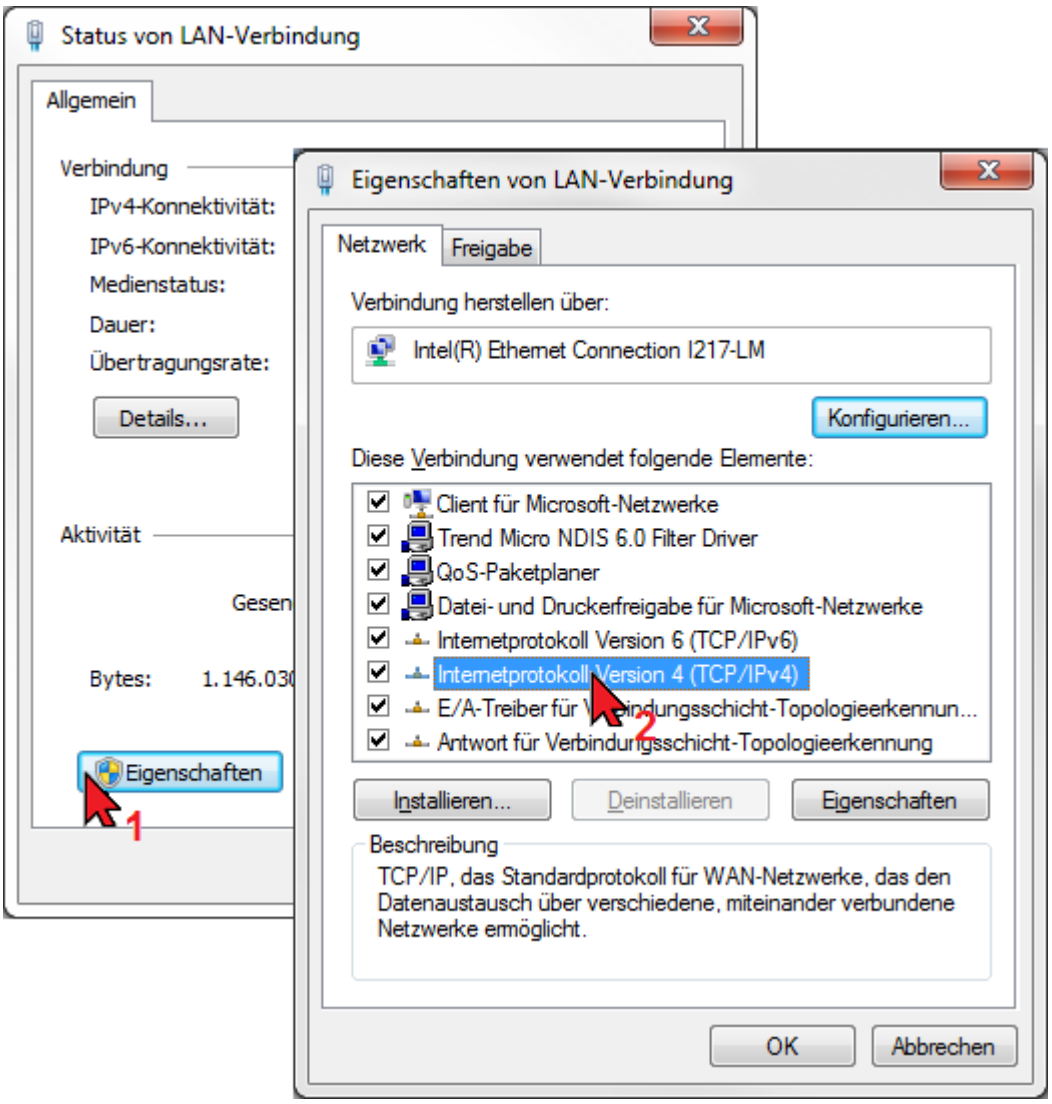
Beispiel Windows 7

Öffnen Sie **'Systemsteuerung'** → **'Netzwerk und Internet'** → **'Netzwerk- und Freigabecenter'**. Öffnen Sie durch Anklicken Ihre aktive LAN Verbindung.

Keine LAN Verbindung aktiv: Rufen Sie das Menü **'Adaptereinstellungen ändern'** auf.



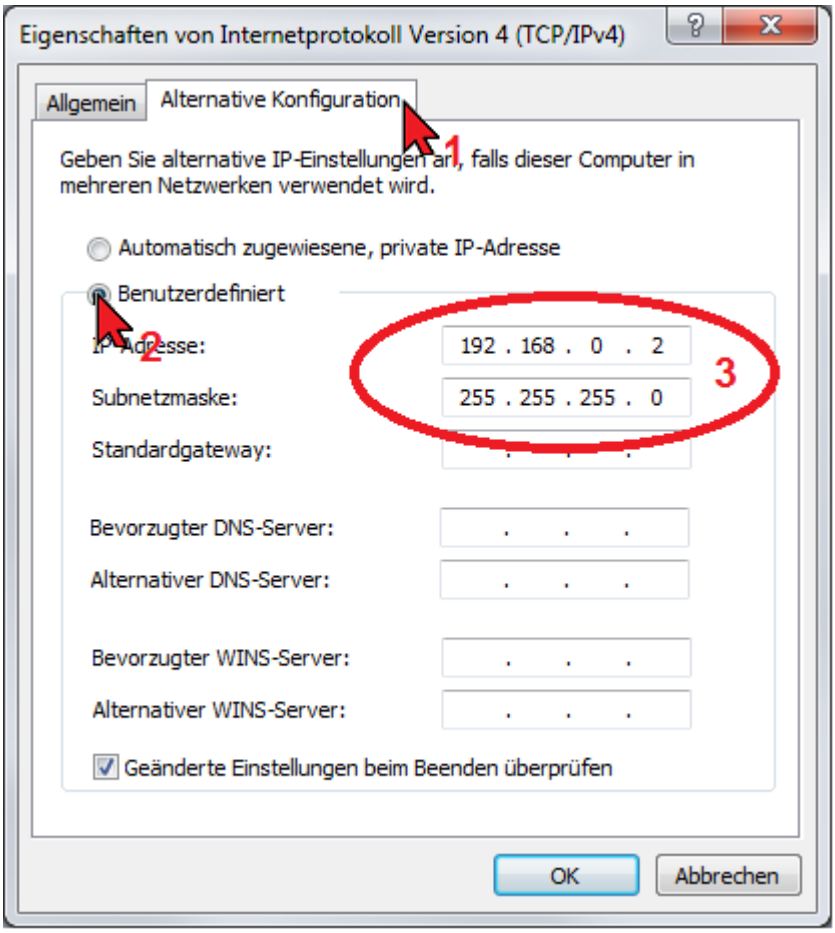
Öffnen Sie das **'Internetprotokoll Version 4 (TCP/IPv4)'**



Tragen Sie im Reiter **'Alternative Konfiguration'** unter **'Benutzerdefiniert'** die 'IP Adresse 192.168.0.2' und die 'Subnetzmaske 255.255.255.0' ein.

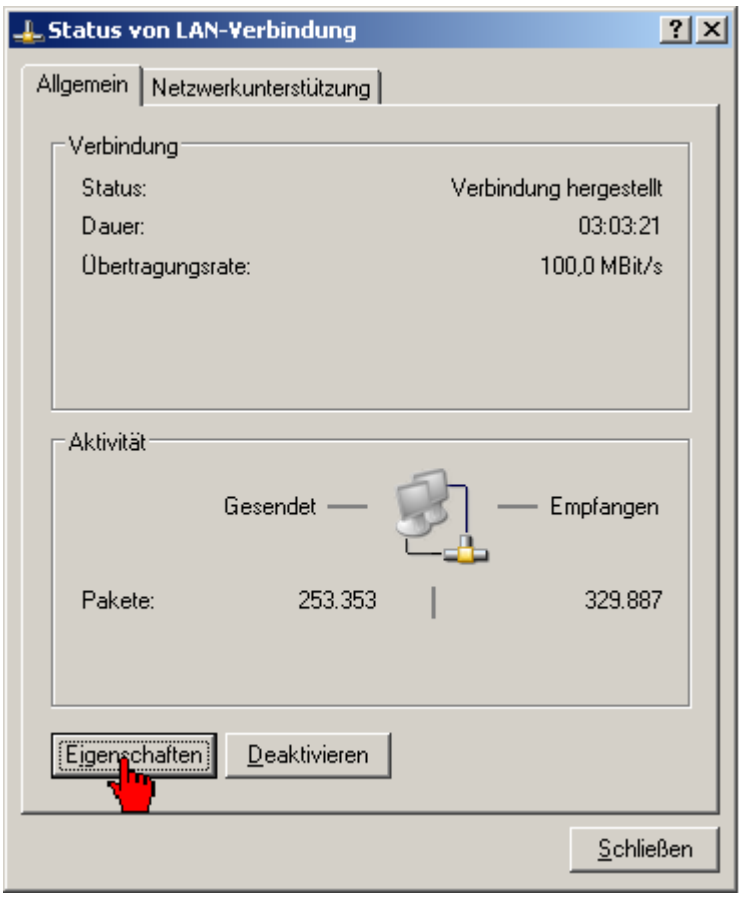
Bestätigen Sie mit dem Taster **'OK'**.

Der Verbindungsaufbau zwischen PC und Steuerung geht schneller, wenn die IP Adresse im Reiter **'Allgemein'** eingetragen wird. Nachteil, die IP Adresse muss jedes mal manuell geändert werden, wenn Sie zwischen Firmennetzwerk und Steuerung wechseln.

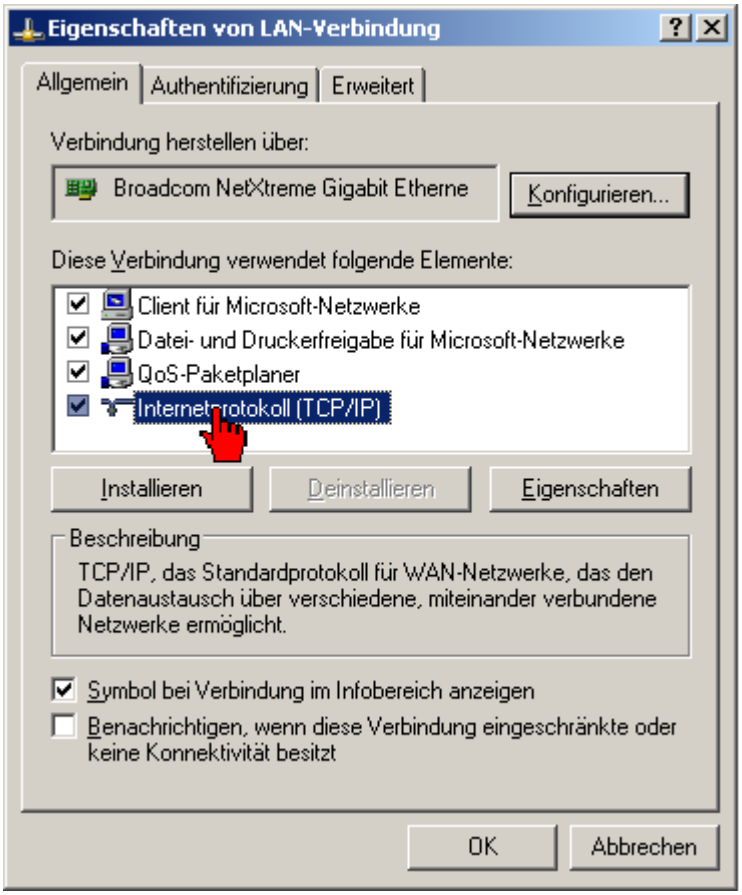


Beispiel Windows XP

Öffnen Sie die 'Netzwerkverbindungen'. Öffnen Sie durch Anklicken Ihre aktive LAN Verbindung.
Wählen Sie den Taster 'Eigenschaften' an.



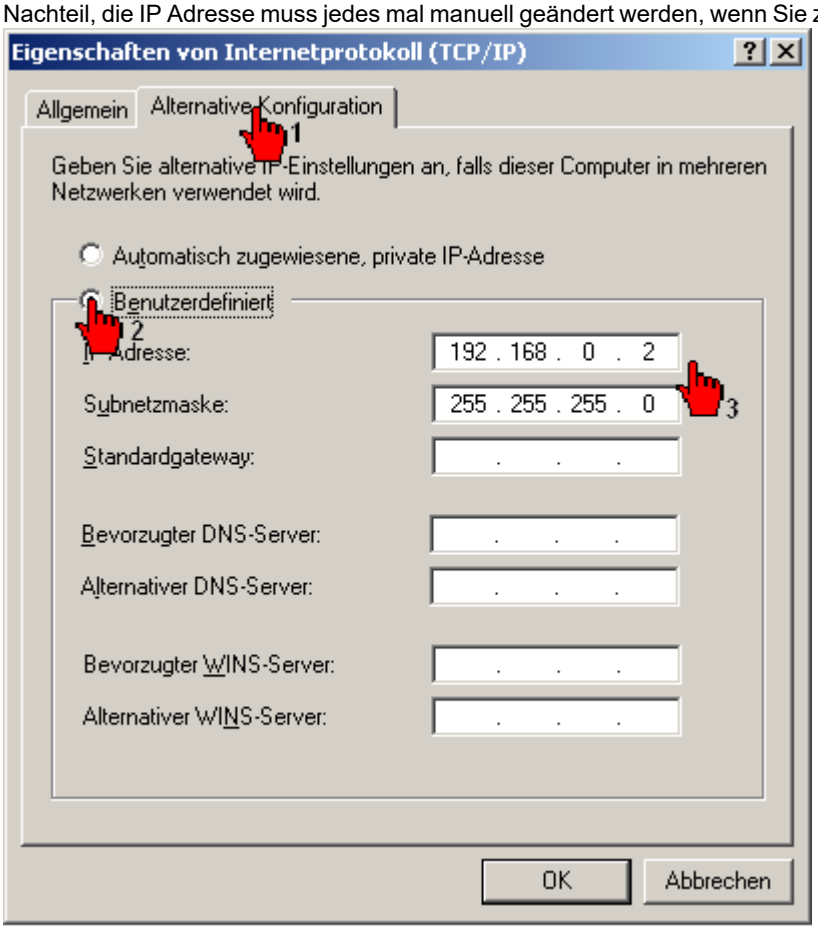
Öffnen Sie durch Anklicken die Eigenschaften des 'Internetprotokolls TCP/IP'.



Tragen Sie im Reiter **'Alternative Konfiguration'** unter **'Benutzerdefiniert'** die 'IP Adresse 192.168.0.2' und die 'Subnetzmaske 255.255.255.0' ein.

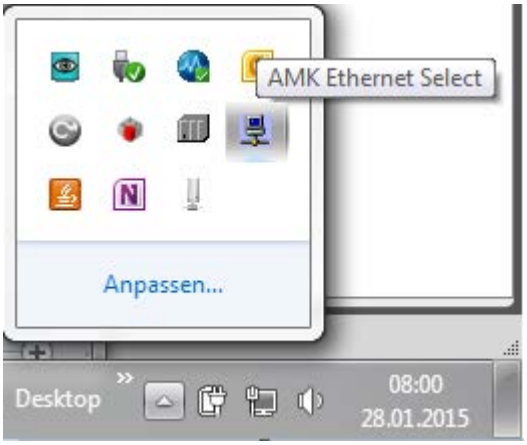
Bestätigen Sie mit dem Taster **'OK'**.

Der Verbindungsaufbau zwischen PC und Steuerung geht schneller, wenn die IP Adresse im Reiter **'Allgemein'** eingetragen wird. Nachteil, die IP Adresse muss jedes mal manuell geändert werden, wenn Sie zwischen Firmennetzwerk und Steuerung wechseln.



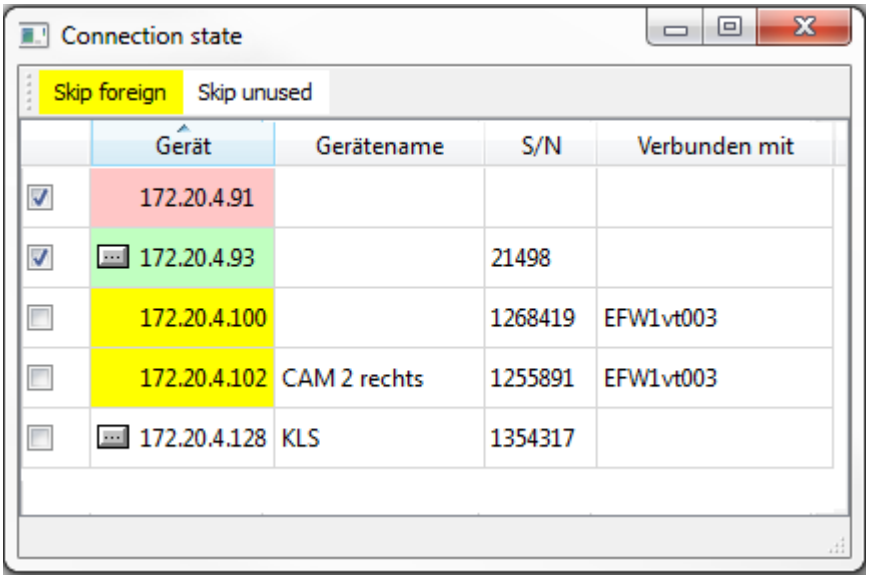
AMK Ethernet Select

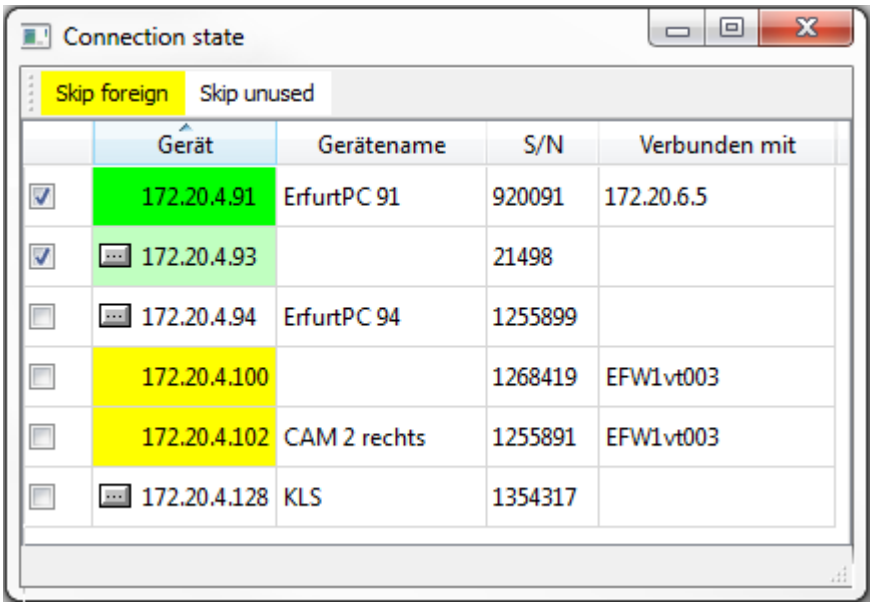
Bei aktiver Ethernet Kommunikation finden Sie **'AMK Ethernet Select'** in der Windows Taskleiste. Wählen Sie das Icon **'AMK Ethernet Select'** an um das Dialogfeld 'Verbindungsstatus' zu öffnen.



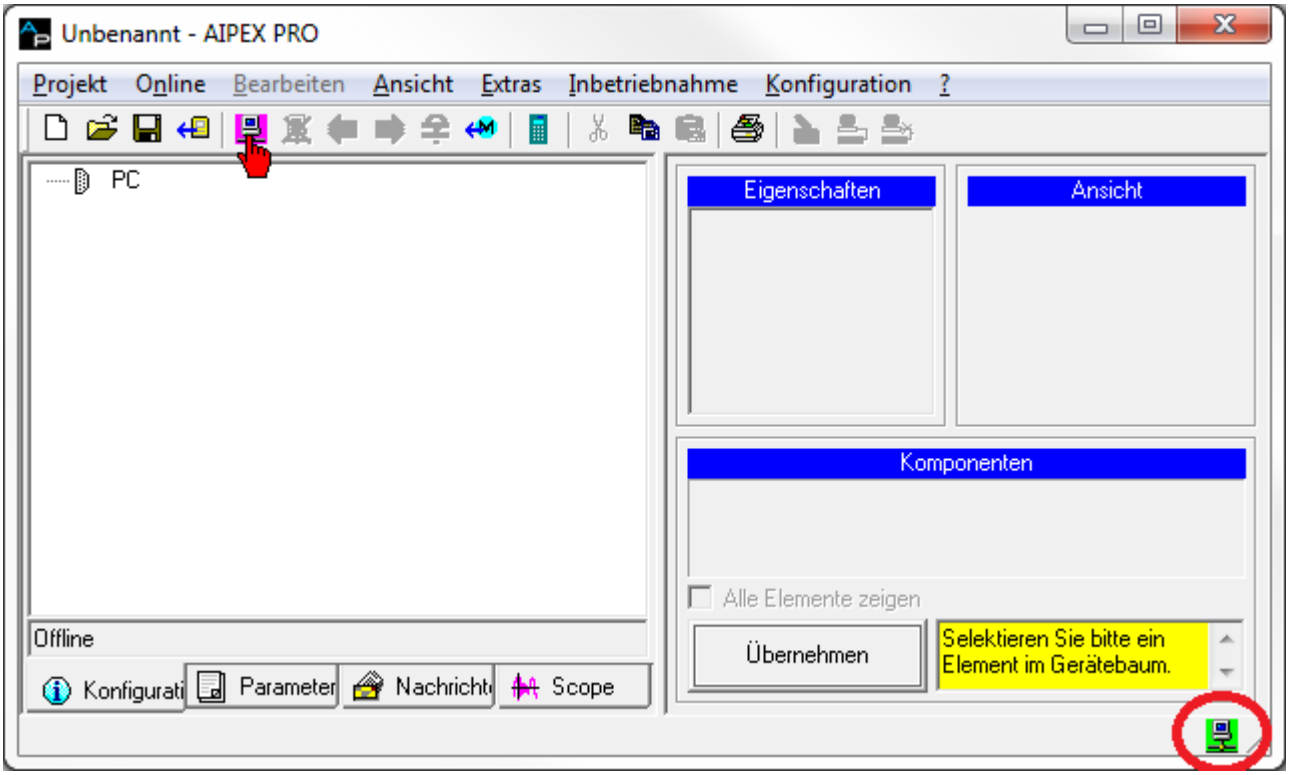
Es werden alle aktiven und die manuell über das Eingabefeld in der Titelleiste **'Gerät'** angelegten AMK Ethernet Geräte angezeigt. Wählen Sie das Gerät, mit dem Sie Verbindung aufnehmen wollen an.

Farbstatus	Bedeutung
Rot	Gerät ist über Ethernet nicht erreichbar
Gelb	Gerät ist mit einem anderen PC verbunden
Hellgrün	Gerät ist mit Ihrem PC verbunden aber nicht aktiv
Grün	Gerät ist mit Ihrem PC verbunden und es wird aktiv auf das Gerät zugegriffen
Weiß	Gerät ist nicht verbunden

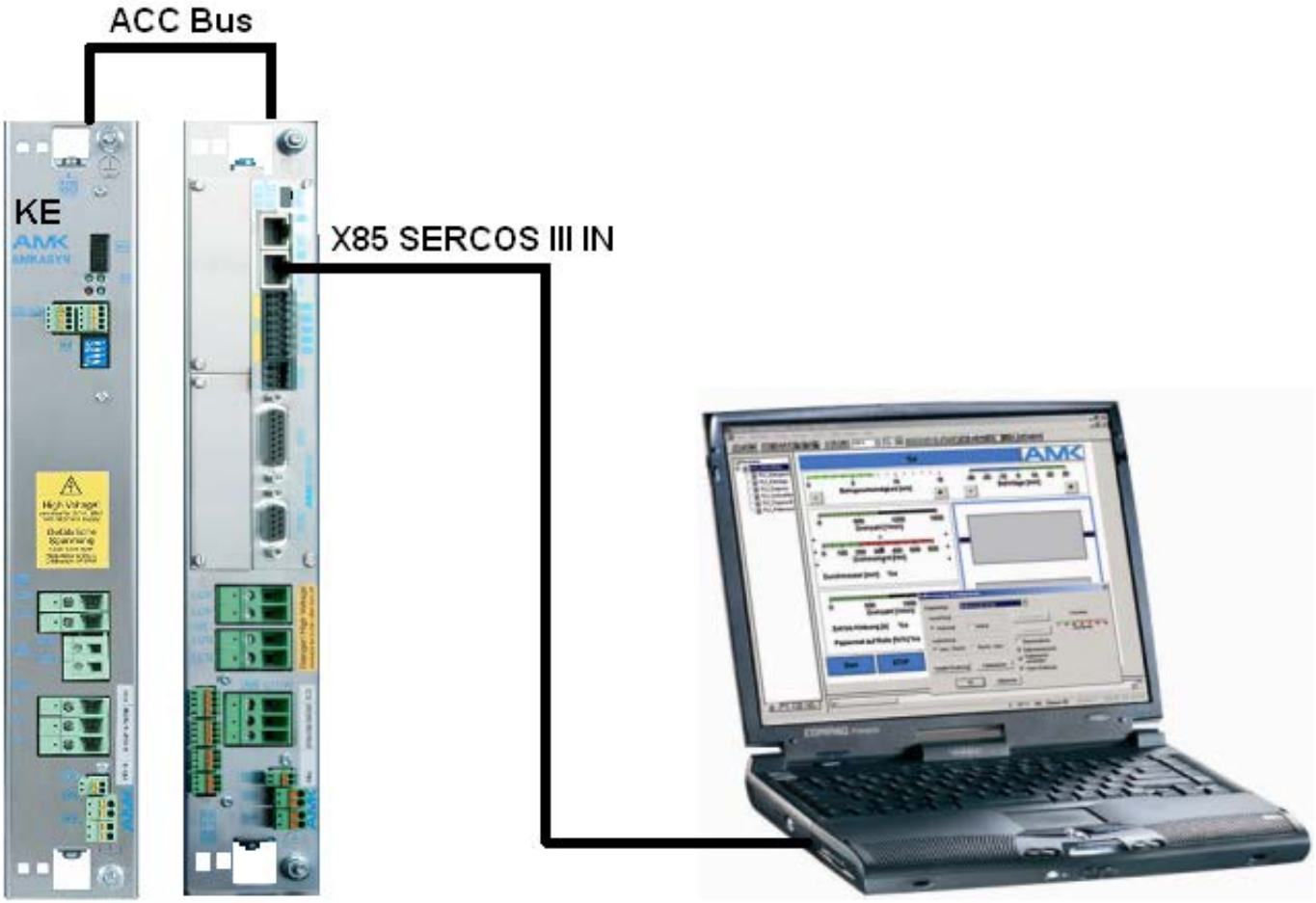




Schließen Sie **'AMK Ethernet Select'**. Sobald das Status 'Kommunikations-Icon' grün/gelb wird können Sie die Taste **'Einloggen'** drücken.



3.4.6 SERCOS III Schnittstelle

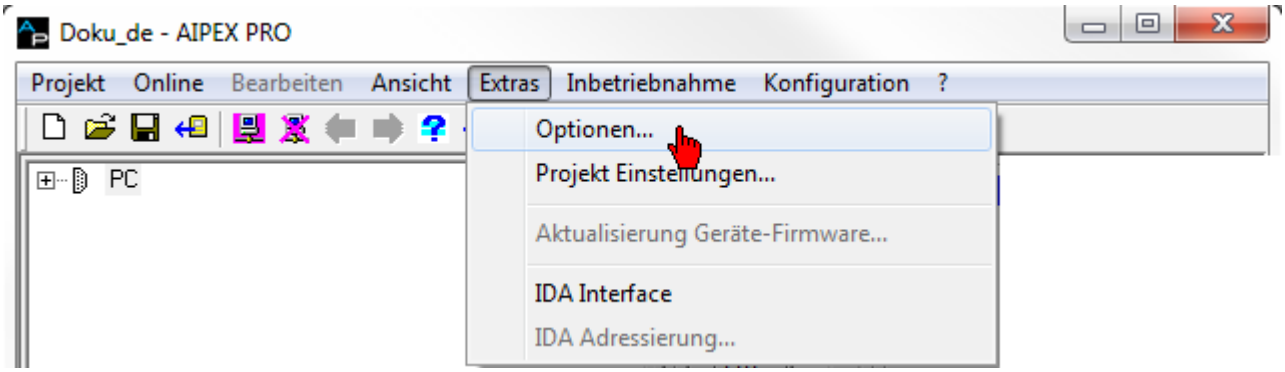


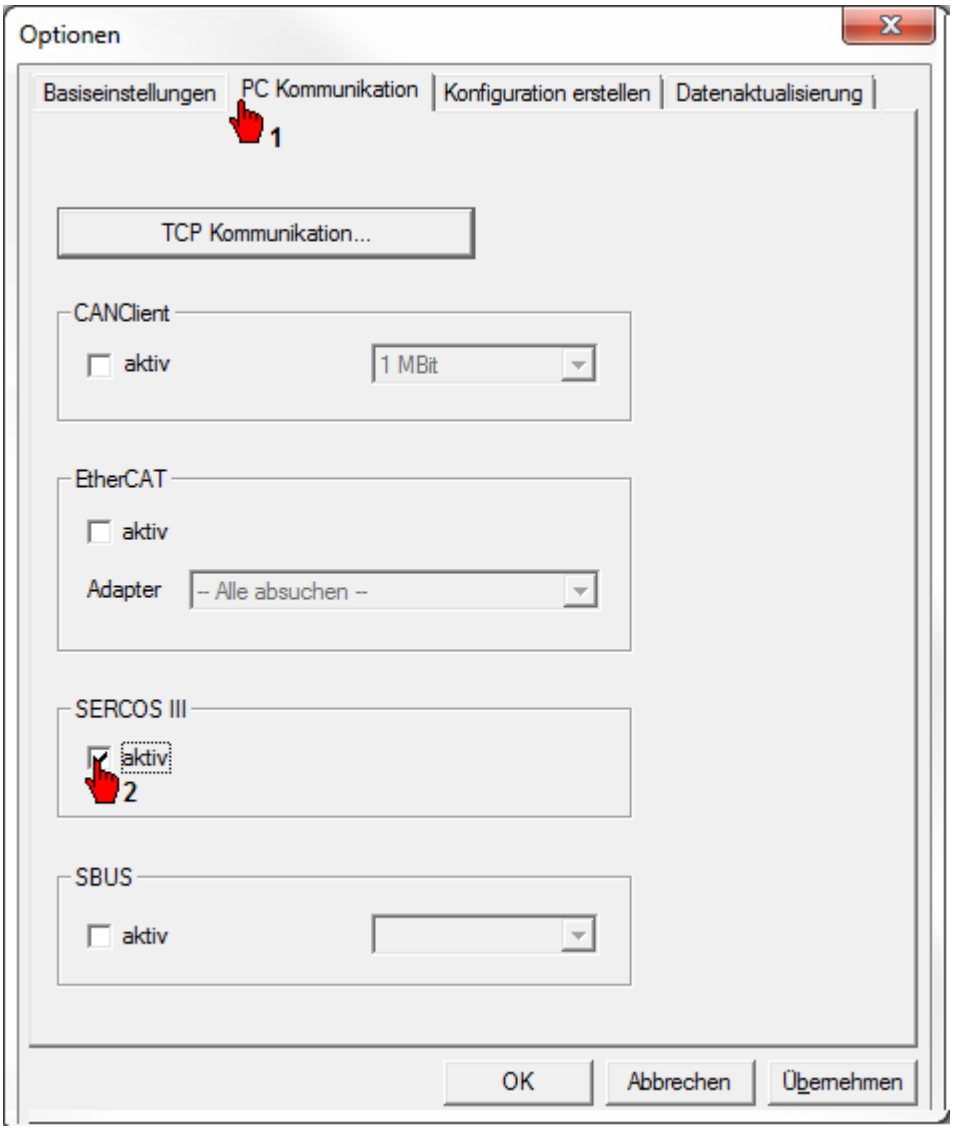
Beispiel:

KE mit ACC Bus und KW mit KW-R06 (kompatibel mit KW-R07, KW-R16 und KW-R17).
Über die SERCOS III Verbindung kann auf alle angeschlossenen Module zugegriffen werden. AIPEX PRO kann zudem über die ACC-Bus Verbindung der Reglerkarte auf ACC-Bus Slave Teilnehmer zugreifen.

Kabel:
Ethernet Standard RJ45 Twisted Pair Patch Kabel.

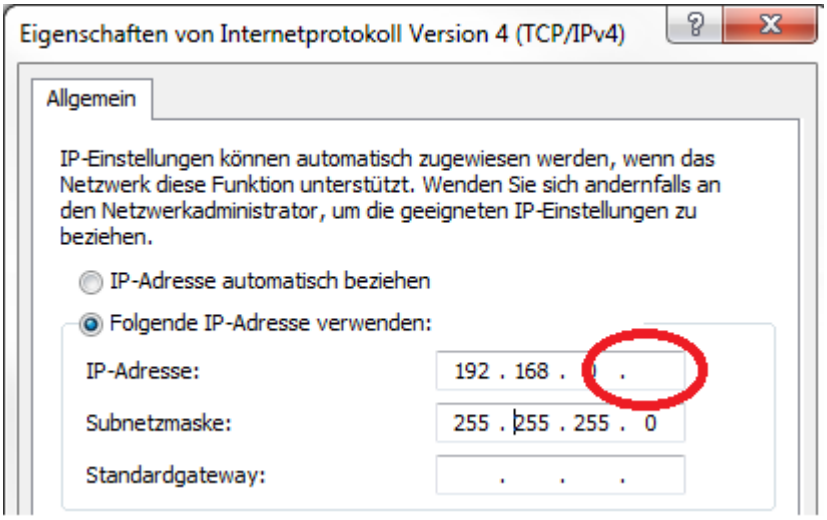
3.4.6.1 SERCOS III Kommunikationseinstellungen



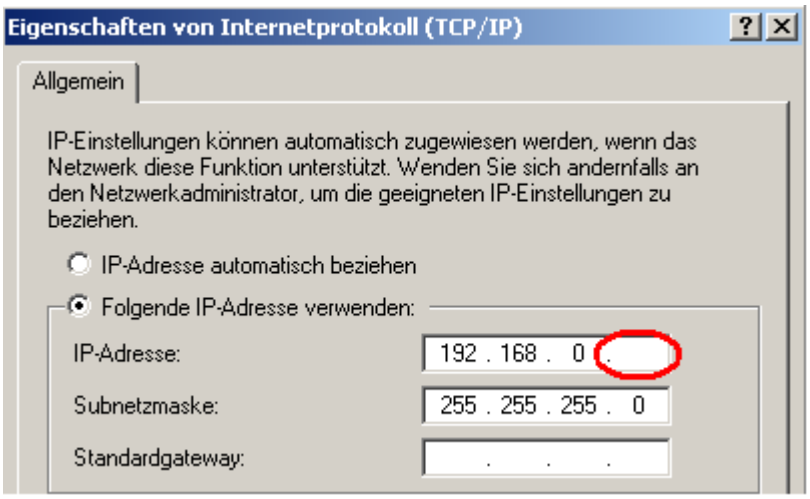


PC Einstellung: Geben Sie eine feste IP Adresse ein, die nicht von einem SERCOS III Teilnehmer belegt ist.
Die SERCOS III Teilnehmer müssen in dem vorgegebenen Adressbereich 192.168.0.x liegen.

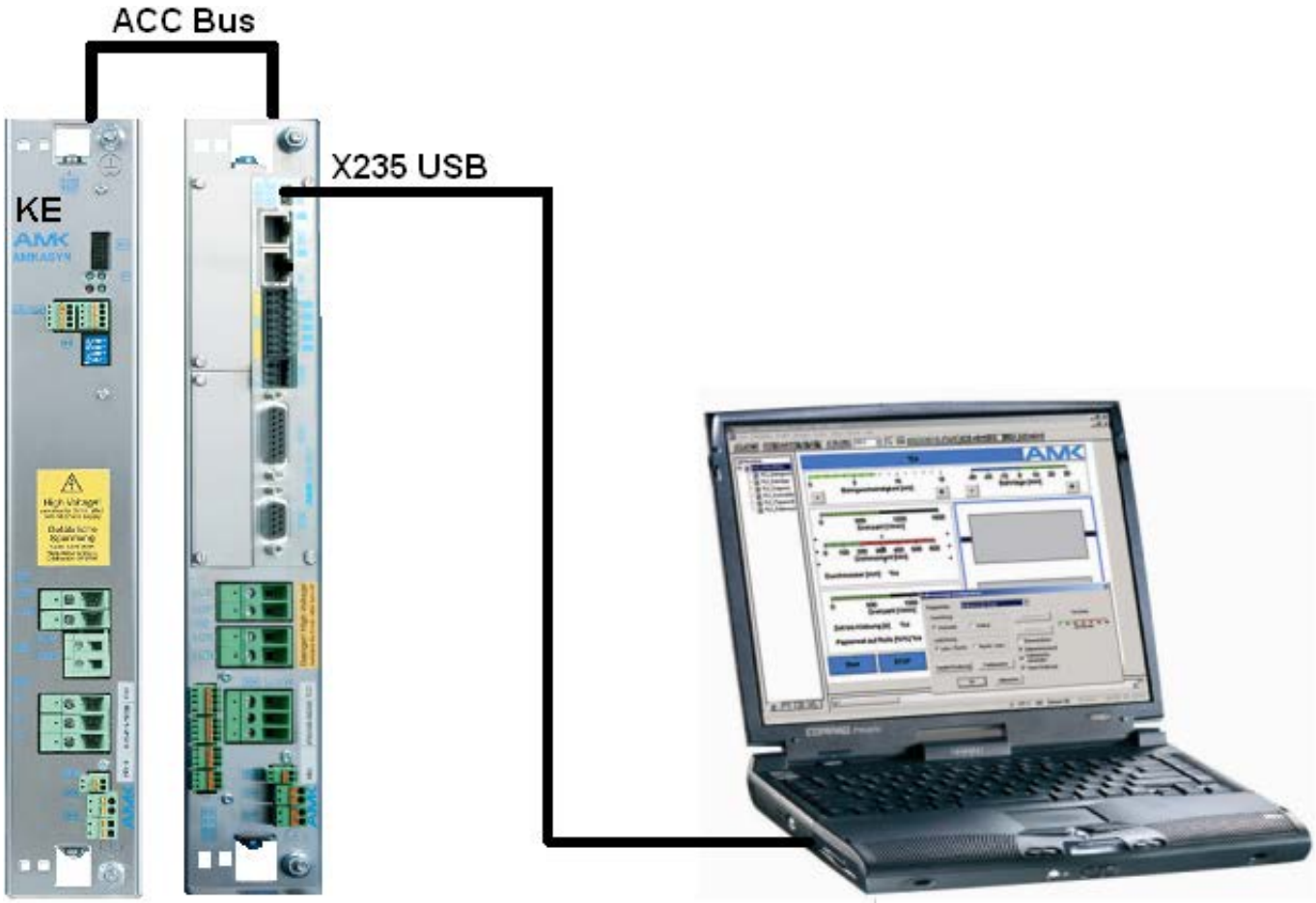
Beispiel Windows 7:



Beispiel Windows XP:



3.4.7 USB Schnittstelle



Beispiel:
KE mit ACC Bus und KW mit KW-R06 (kompatibel mit KW-R07, KW-R16 und KW-R17).
AIPEX PRO kann über die ACC-Bus Verbindung der Reglerkarte auf ACC-Bus Slave Teilnehmer zugreifen.
Kabel:
AMK USB Kabel mit Ferritkern AMK Teile-Nr. 47058

3.4.7.1 USB Kommunikationseinstellungen

Bei Verwendung der USB Schnittstelle müssen keine Kommunikationseinstellungen verändert werden.

3.4.8 AIPEX PRO add in gateway for TwinCAT V2

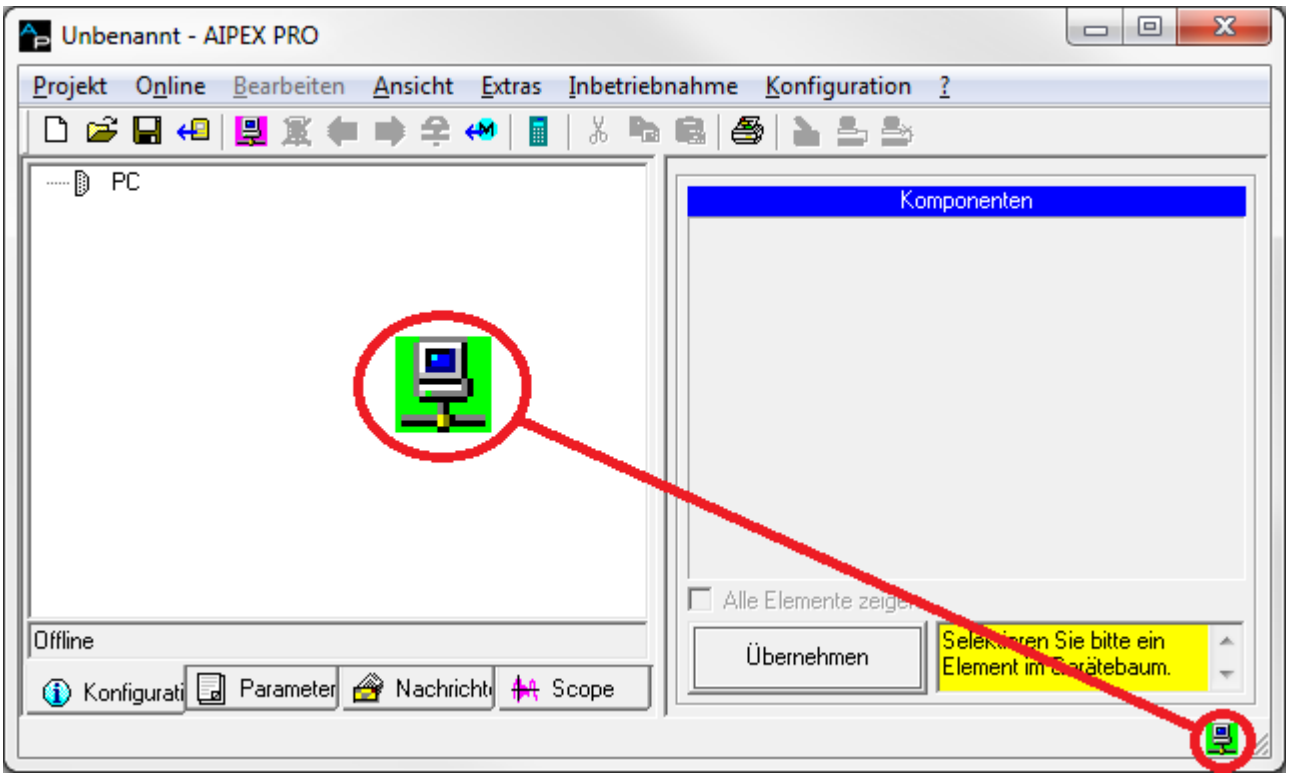
Siehe Dokument PDK_204072_AIPEXPRO_Add_In_TwinCAT_Gateway

3.4.9 Kommunikation testen

1. Legen Sie die 24 VDC Versorgungsspannung am Gerät an und verbinden Sie es mit dem PC.
2. Starten Sie AIPEX PRO.
3. Nach der Initialisierungsphase zeigt das grüne 'Kommunikations-Icon' in der PC Statusleiste eine aktive Verbindung zwischen PC (mit AIPEX PRO) und dem AMK Gerät an.

Voraussetzung:

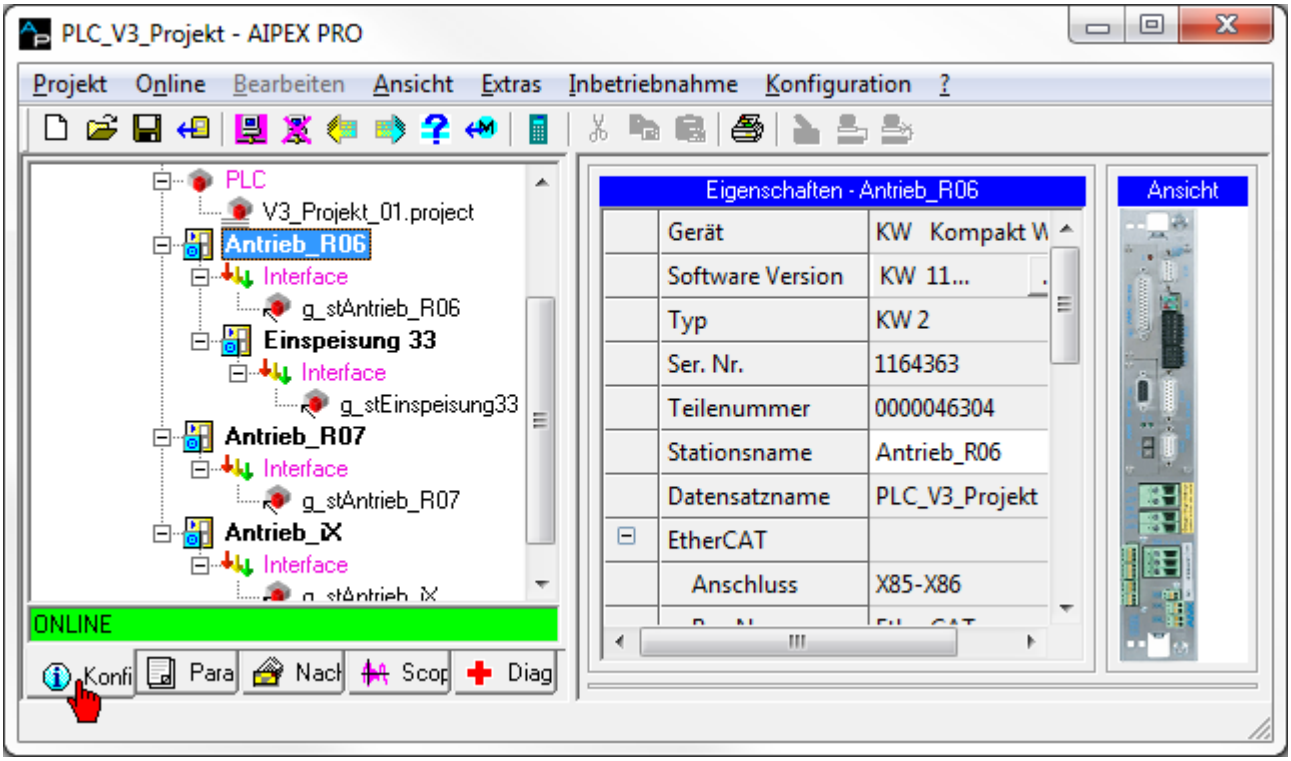
- Kommunikationsverbindung in AIPEX PRO definiert
- Verbindungskabel zwischen PC und Antrieb eingesteckt
- Ansicht AIPEX PRO Fenster auf 'Maximiert' (Vollbildschirm)



Grünes Symbol	Verbindung zwischen PC und AMK Gerät ist aktiv
Kein bzw. rotes Symbol	• keine Verbindung vorhanden, • keine Schnittstelle definiert • AIPEX PRO Fenster wird 'Minimiert' dargestellt.
Symbol zweifarbig (grün / rot)	mehrere Kommunikationsschnittstellen in AIPEX PRO aktiviert.

3.5 Reiter

3.5.1 Konfiguration



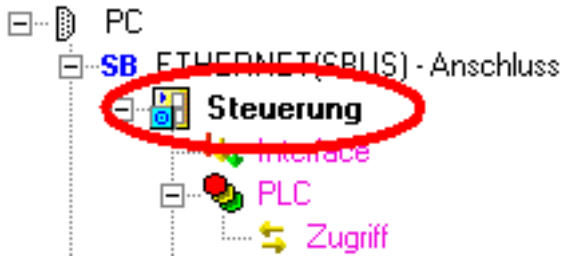
Im 'Gerätebaum' finden Sie die Geräte, die offline und online vorhanden sind.

Die Ansicht des 'Gerätebaumes' kann für jeden Reiter spezifisch eingestellt werden.Siehe 'Programmübersicht - Teilansichten' auf Seite 40. und siehe 'Programmübersicht - Anzeigefilter' auf Seite 39

Auf der rechten Seite finden Sie die spezifischen Gerätemerkmale. Über den Reiter 'Konfiguration' können (offline) Projekte komplette offline erstellt und anschließend auf die Maschine übertragen werden.

3.5.1.1 Konfiguration - Anleitung

3.5.1.1.1 Eigenschaften - Gerät

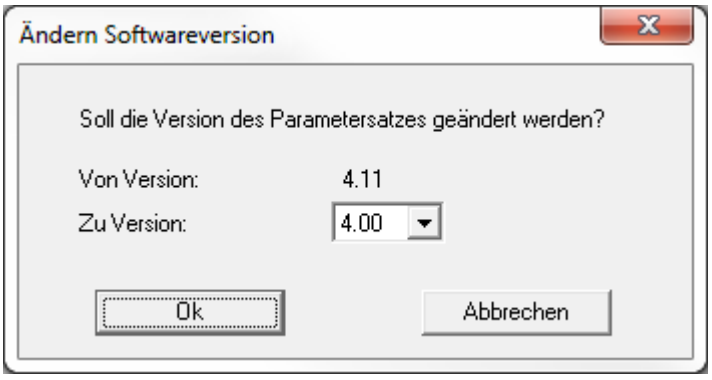


Geräte Eigenschaften

AIPEX PRO	Beschreibung	Beispiel
Gerät	AMK Gerätebezeichnung	KW Kompakt Wechselrichter
Software Version	Installierte Software Version auf dem Gerät.	KW 111 1320 204533
Typ	AMK Typbezeichnung mit Leistungsangabe	KW 2
Ser. Nr.	Seriennummer	123456
Teilenummer	Teilenummer	46304
Stationsname	Der Stationsname kann anwendungsspezifisch vom Benutzer eingegeben werden.	Antrieb_R06
Datensatzname	Name unter dem das aktuell geöffnete AIPEX PRO Projekt gespeichert ist.	Demo

Taster **'Software Version'**

Bei einem Offline Projekt kann der Geräte Softwarestand eingestellt werden. (Beispiel: Projekt wird offline erstellt und anschließend auf eine bestehende Applikation mit definiertem Geräte Softwarestand gespielt.)



Feldbus Eigenschaften

AIPEX PRO	Beschreibung	Beispiel
Anschluss	Schnittstelle auf dem Gerät	X85-X86
Bus Name	AMK Busbezeichnung	EtherCAT
Bus Physik	Generelle Busbezeichnung	ETHERCAT
Instanz	Jeder Bus wird in einer eigenen Instanz parametriert.	1
IP Adresse *1	Netzwerkeinstellung für Steuerungen	192.168.0.1
Netzwerk Maske *1		255.255.255.0
Gateway Adresse *1		0.0.0.0

AIPEX PRO	Beschreibung	Beispiel
Adresse *12	Geräteadresse	1
Feste Adresse *1	Automatisch vom Busmaster zugewiesene oder feste Adresse	
Optional *1	Das Gerät muss bei der Initialisierung des Bussystems nicht anwesend sein. Voraussetzung: 'Fest Adresse' ist aktiviert	
Master *1	Das Gerät ist Busmaster oder Busslave	
Knoten Konfigurationsdatei *1	Gerätebeschreibungsdatei (*.eds oder *.xml)	
Feste Adresse der Slaves *1	Alle Slaves im Bus werden auf 'Feste Adresse' gesetzt bzw. rückgesetzt.	

*1 bus- oder gerätespezifisch

***2 Unterschiede Offline / Online Projekt bei EtherCAT**

Offline Projekt

1. **Automatische Adressierung durch den Busmaster**
   Wenn Sie ein Offline Projekt erzeugen, muss die zukünftige Busposition als Adresse eingetragen werden.
   Voraussetzung: ID34023 'BUS Teilnehmer Adresse' = 0.
2. **Feste Adressen**
   Geben Sie die Adresse ein, setzen Sie anschließend den Haken bei **'Feste Adresse'**.

Online Projekt

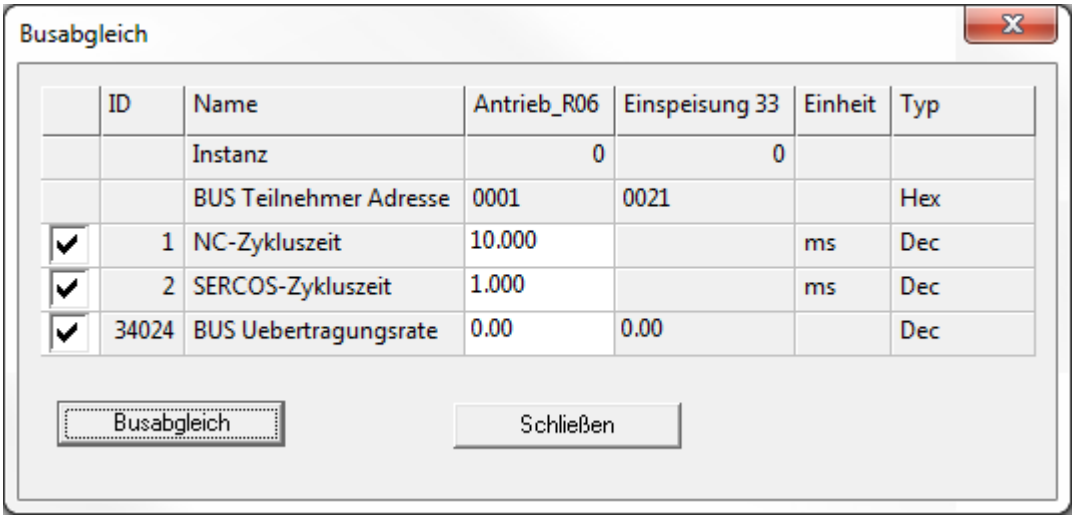
AIPEX PRO liest die aktuelle (Fest-) Adresse aus dem Gerät und zeigt diese an.

Nach dem Ändern der Adresse bzw. der Checkbox **'Feste Adresse'** müssen Sie die Busteilnehmer neu starten.

Die (Fest-) Adresse wird durch den Aufruf des Menüpunktes **'Konfiguration'** → **'Konfiguration erstellen...'** in die XML Konfigurationsdatei übernommen.

Taster **'Busabgleich...'**

Die angewählten Parameter werden in allen Busteilnehmern auf den gleichen Wert gesetzt.



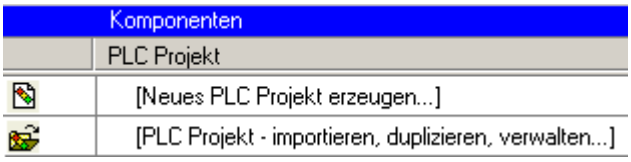
3.5.1.1.2 Interface

Das 'Interface' ist die Schnittstelle zwischen den physikalisch vorhandenen Geräten und den symbolischen Gerätenamen (Variablen) aus dem PLC Editor.

Die Zuweisung kann automatisch erfolgen (Siehe 'Automatische Buskonfiguration' auf Seite 743.) , oder manuell per 'Drag and Drop'.

3.5.1.1.3 PLC

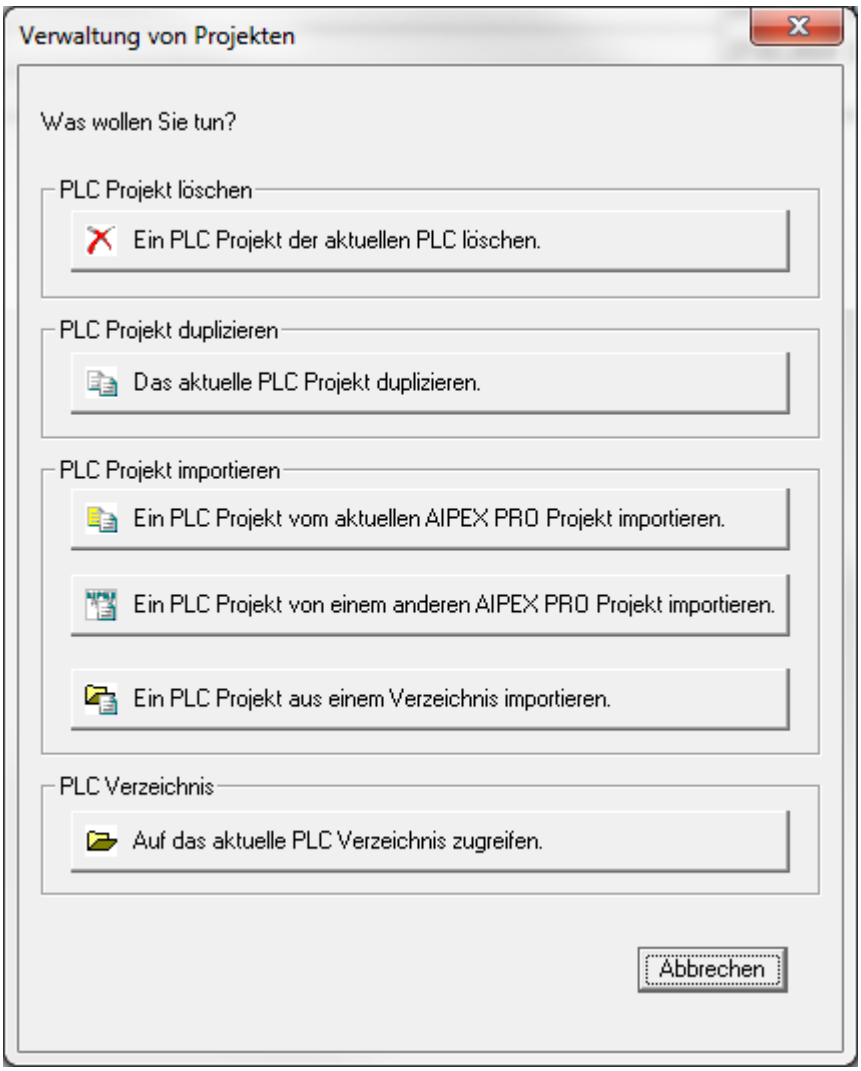
Klicken Sie auf **'PLC'** um das PLC Projekt zu verwalten.



Neues PLC Projekt erzeugen

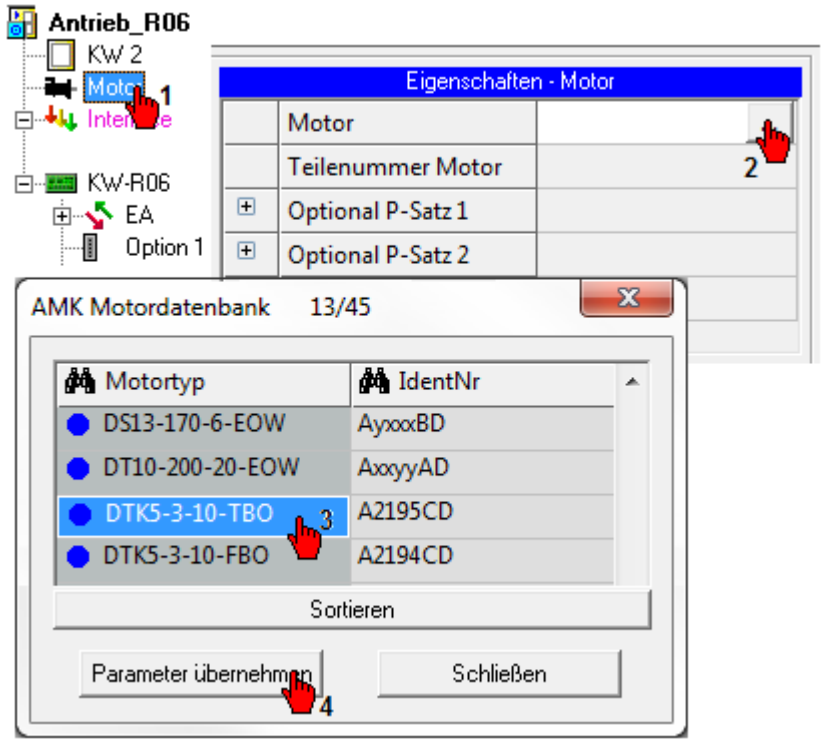
Öffnet den PLC Editor und legt ein neues PLC Programm an.

PLC Projekt - importieren, duplizieren, verwalten ...

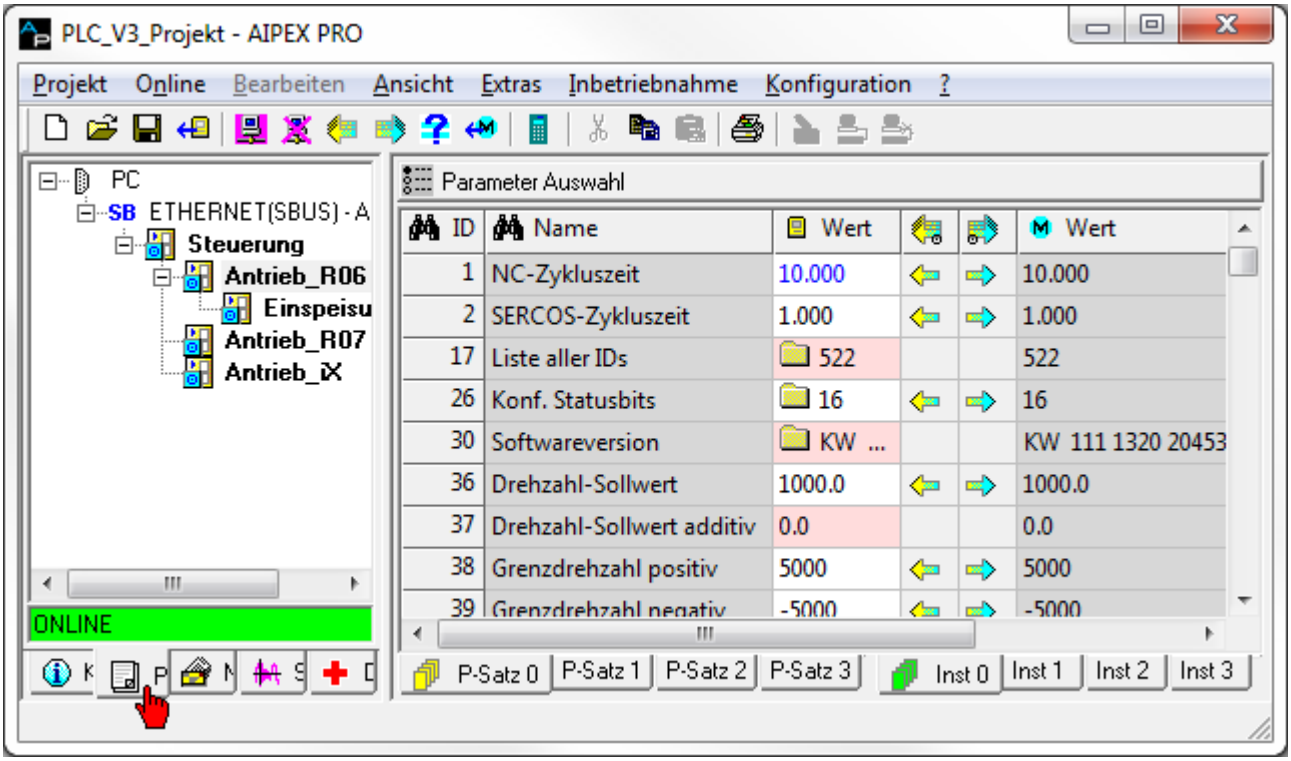


3.5.1.1.4 Motor

Manuelle Auswahl der Motorparameter.



3.5.2 Parameter



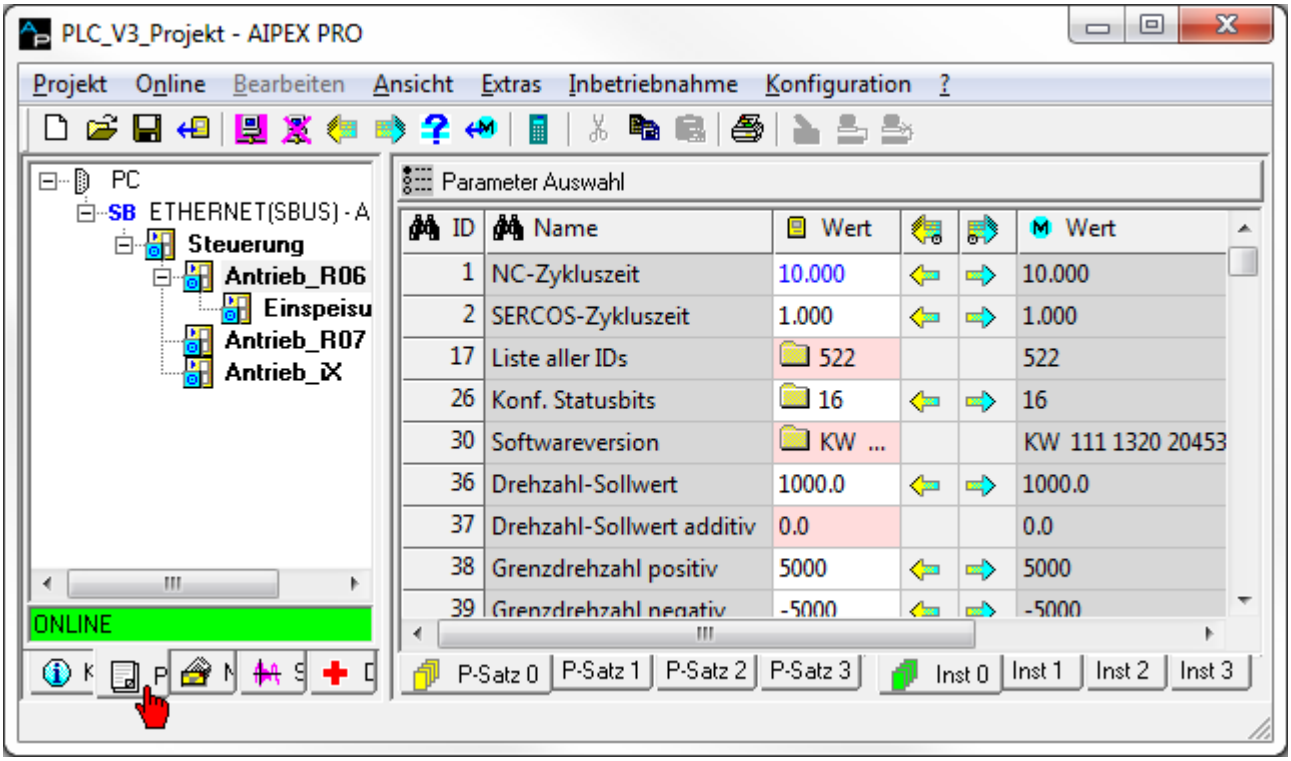
'Parameter' zeigt vorhandene offline und online Werte des angewählten Gerätes an.

3.5.2.1 Parameter - Anleitung

Änderungen im Reiter 'Parameter' werden im Gerät nicht sofort aktiv. Die Aktivierung der Änderung ist von der Parametergruppe abhängig.

- Gruppe 'Global' und 'Instanz': Logikspannung 24 VDC aus/an
- Gruppe 'Antriebsspezifisch': Reglerfreigabe aus/an

3.5.2.1.1 Übersicht

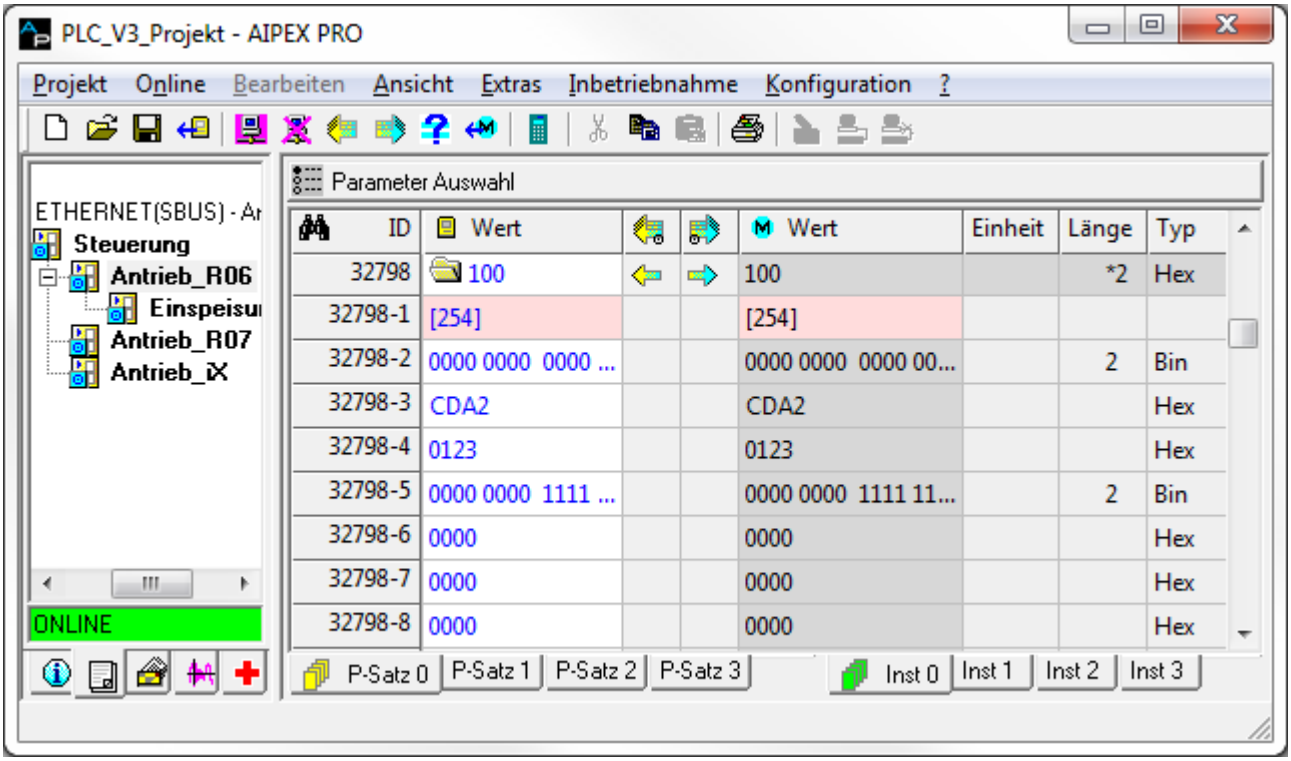


Icon	Bedeutung
Parameter Auswahl	Öffnet Fenster 'Parameter Auswahl' und 'Systeminterne Parameter' Siehe 'Parameter - Auswahl' auf Seite 81.
ID	Suchen nach ID Nummer
Name	Suchen nach ID (Parameter) Name
Wert	Online Wert (gespeicherter Wert im Gerät)
Wert	Offline Wert (gespeicherter Wert im PC)
	Upload Parameterwert (vom Gerät zum PC) Download Parameterwert (vom PC zum Gerät) des ausgewählten Parameters
	Upload (vom Gerät zum PC) Download (vom PC zum Gerät) des kompletten ausgewählten Parametersatzes
P-Set 0	Parametersatz
Inst 0	Instanz (benutzt für Busparameter)
	Auswahlfeld für Biteingaben
Berlin	Listenparameter

Icon	Bedeutung
0.27	Rosa Hintergrund: Read only Parameter

Feld	Bedeutung
ID	Parameternummer
Name	Bezeichnung des Parameters
Wert	Aktueller Wert des Parameters Icon Ordner = Parameterliste
Einheit	Maßeinheit des Parameters
Länge	Datenlänge des Parameters in Byte • 1 Byte • 2 Byte • 4 Byte
Typ	Darstellung des Parameters • Dec: dezimal • ±Dec: dezimal vorzeichenbehaftet • Bin: binär • Hex: hexadezimal • Ascii: Ascii-String
Anmerkung	Frei durch den Anwender nutzbares Textfeld
Hintergrundfarbe in der Spalte „Wert“	Weiß: Wert kann geändert werden Rosa: Wert ist nicht änderbar (schreibgeschützt oder formal)
Textfarbe in der Spalte „Wert“	Blau: Wert wurde geändert aber noch nicht gespeichert Schwarz: Wert ist seit der letzten Speicherung unverändert Rot: Es handelt sich um einen Systemparameter (nur nach Freischaltung änderbar)
Icon in der Spalte „Länge“	Gelb: Parameter ist parametersatzspezifisch Grün: Parameter ist instanzspezifisch

3.5.2.1.2 Darstellung und Strukturierung von Listen



Listen

- Listen werden durch das Ordner Icon in der Spalte 'Wert' gekennzeichnet.
- Bei nicht ASCII-Listen steht neben dem Icon die Anzahl der Listenelemente. Bei ASCII-Listen wird hier der Textinhalt angezeigt.
- Alle Listen können durch Doppelklick in die Spalte 'Wert' expandiert dargestellt werden. In der ersten Unterzeile steht die Anzahl der maximal möglichen Listenelemente. Alle weiteren Zeilen zeigen den Listeninhalt Elementweise.
- Die Anzeige der Listenelemente kann durch Eingabe eigener Werte in den Spalten Länge (1, 2, 4) und Typ (Dec, Hex, Bin, Ascii) strukturiert werden.

Eingabe von Werten

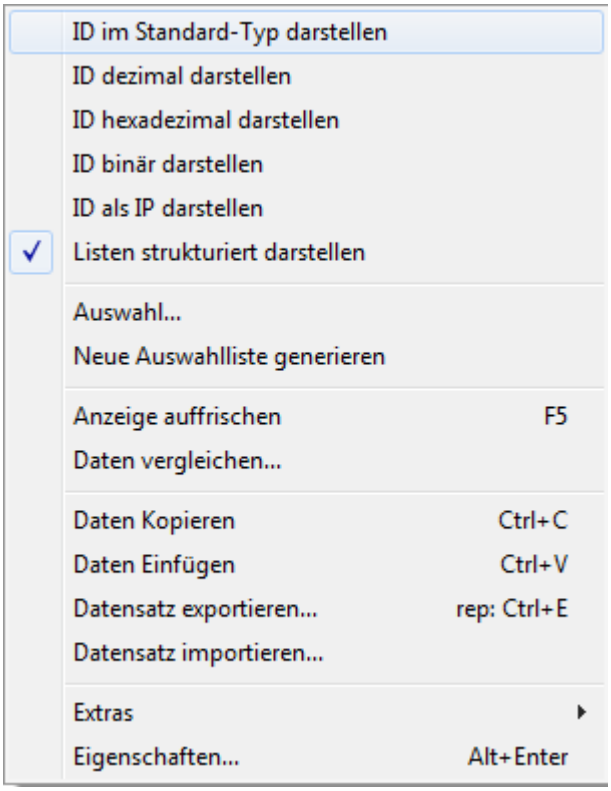
- Der Wert änderbarer Parameter kann direkt in der Tabelle editiert werden. Jede Eingabe wird durch die **'Enter'** Taste abgeschlossen.
- Die Eingabe muss dem angezeigten Datentyp entsprechen.
- Einige Parameter haben minimale und maximale Begrenzungen des Wertes. Wird dieser bei der Eingabe unter- bzw. überschritten wird eine Fehlermeldung angezeigt.

Veränderung von Listen

- Bei nicht ASCII-Listen wird durch die Eingabe eines Wertes neben dem Ordner Icon die aktuelle Listenlänge verändert. Der Wert darf nicht größer als die maximale Listenlänge sein.
- Bei ASCII-Listen wird eine Eingabe neben dem Ordner Icon direkt als String interpretiert und die Liste geändert.
- Bei expandierter Liste können die Listenelemente direkt verändert werden. Die Eingabe muss dem angezeigten Datentyp des Elementes oder, wenn keines existiert, dem der Liste entsprechen.

3.5.2.1.3 Kontextmenü Parameter

Wählen Sie den Reiter **'Parameter'** an. Durch Anwählen eines beliebigen Parameters mit der rechten Maustaste öffnet sich das Kontextmenü.

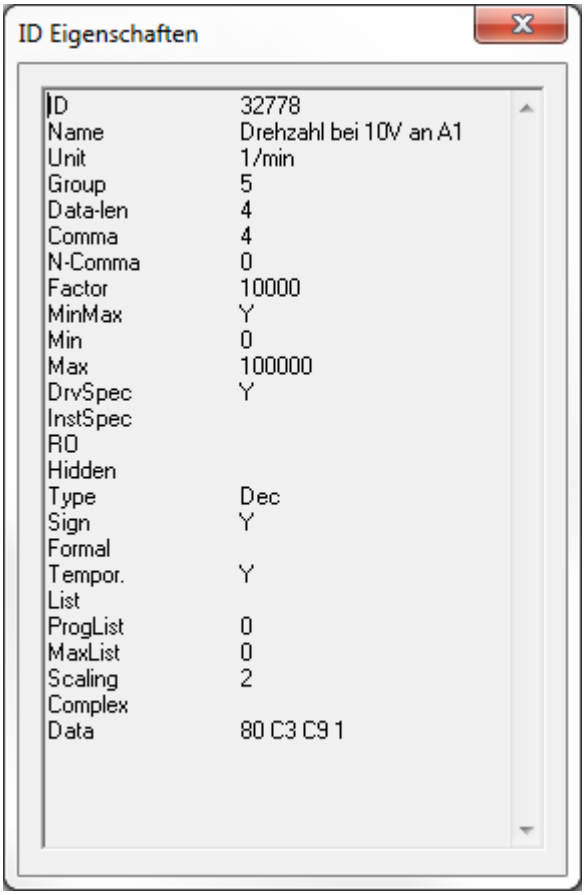


Feld	Bedeutung
ID im Standard-Typ darstellen	Der angewählte Parameter wird dargestellt, wie gekennzeichnet.

Feld	Bedeutung
ID dezimal darstellen	Der angewählte Parameter wird dezimal dargestellt.*
ID hexadezimal darstellen	Der angewählte Parameter wird hexadezimal dargestellt.*
ID binär darstellen	Der angewählte Parameter wird binär dargestellt.*
Listen strukturiert darstellen	Ist im Kontextmenü 'Listen strukturiert darstellen' aktiviert, kann der Anwender 'Name', 'Wert', 'Einheit Länge' festlegen (z. B. Anwenderlisten).
Neue Auswahlliste generieren	Öffnet das Fenster Siehe 'Parameter - Auswahl' auf Seite 81.
ID in Auswahlliste übernehmen	Die Nummer des angewählten Parameters wird in die aktuell eingestellte 'eigene Liste' aufgenommen. Es ist dabei nicht von Bedeutung, ob die 'eigene Liste' aktiv ist.
Anzeige auffrischen	Aktualisierung der Online Werte. Alle Werte werden verworfen und erneut aus dem angeschlossenen Antrieb ausgelesen.
Daten vergleichen	Der Inhalt des aktuell angezeigten Parametersatz kann mit dem eines gespeicherten verglichen werden.
Daten Kopieren	Die Inhalte der aktuell markierten Teile der Parametertabelle werden in die Windows Zwischenablage kopiert und stehen als einfügbarer Text für viele Anwendungen, z.B. Textprogramme, zur Verfügung. Sind komplette Zeilen in der Tabelle markiert, wird zusätzlich eine interne Kopie dieser Parameter angelegt.
Daten Einfügen	Existiert eine interne Kopie von Parametern, werden deren Inhalte in die entsprechenden Parameter eingefügt. Existiert keine interne Kopie, wird der Inhalte der Windows Zwischenablage ungeprüft als Text ab der aktuell selektierten Stelle in die Tabelle eingefügt.
Daten exportieren	Durch den Kontext-Menüpunkt **'Datensatz exportieren'** der Parametertabelle werden nur die Parameter der aktuellen Auswahl (z.B. Auswahl Motorparameter) gesichert.
Daten importieren	Mit der Funktion **'Datensatz importieren'** werden alle in der Datei (externes XML File) vorhandenen und schreibbaren Parameter importiert.
Eigenschaften	Anzeige der Eigenschaften und Attribute des angewählten Parameters.

* Ausgenommen hiervon sind ASCII Listen

3.5.2.1.4 ID Eigenschaften

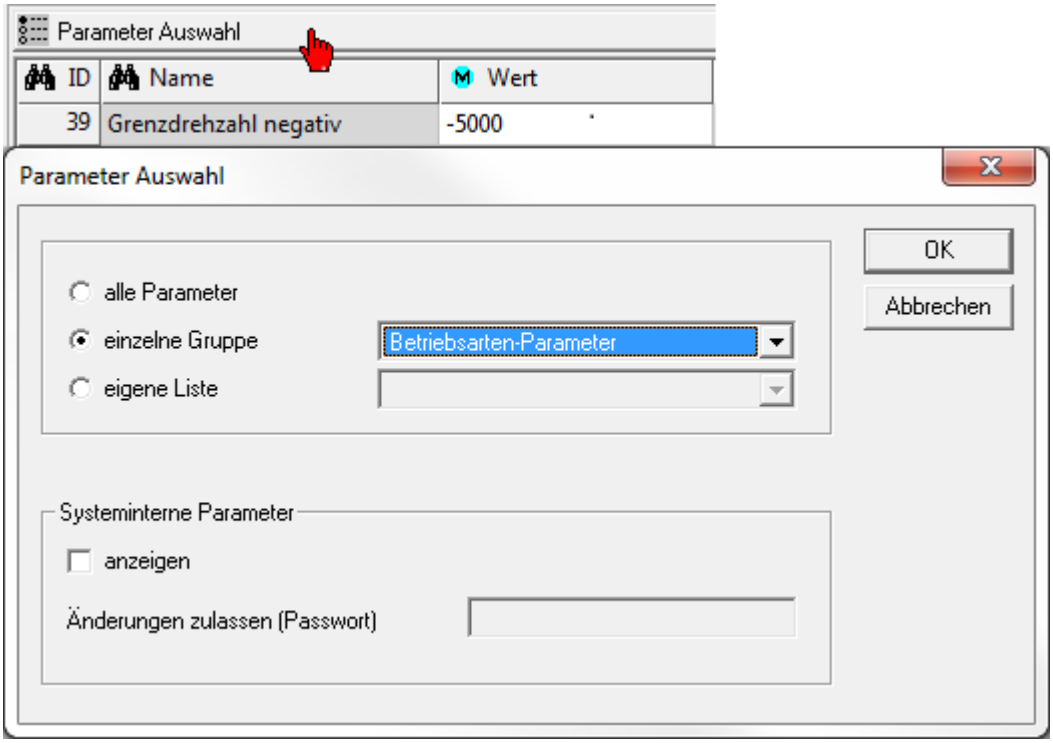


ID Eigenschaften	
ID	Parameternummer
Name	Parametername
Unit	Einheit
Group	Parametergruppe
Data-len	Datenlänge des Parameter
Comma	10er Potenz Skalierungsfaktor X (für 10 hoch X)
N-Comma	Nachkommastellen für Parmeterwertanzeige
Factor	Skalierungsfaktor in 10 hoch X
MinMax	Minimumwert / Maximalwert vorhanden
Min	Minimaler Eingabewert
Max	Maximaler Eingabewert
DrvSpec	Antriebsspezifischer Parameter
InstSpec	Instanzspezifischer Parameter
	[Nicht DrvSpec und nicht InstSpec = Globaler Parameter]
RO	Read Only
Hidden	Systeminterner Parameter
Bin	Eingabeformat Binär
Hex	Eingabeformat Hexadezimal
Ascii	Eingabeformat ASCII
Sign	Eingabeformat Vorzeichenbehaftet
Formal	Prozessdatum, eingetragene Werte werden nicht remanent gespeichert sondern nur zur Laufzeit berechnet
Tempor.	Temporär änderbar
List	Parameter ist vom Typ Liste

ID Eigenschaften	
ProgList	Aktuelle Listenlänge
MaxList	Maximale Listenlänge
Scaling	Wichtungsart
Complex	Typ komplexe Liste
Data	Parameterwert

3.5.2.2 Parameter - Auswahl

Öffnen Sie das Dialogfeld mit dem Taster **'Parameter Auswahl'**

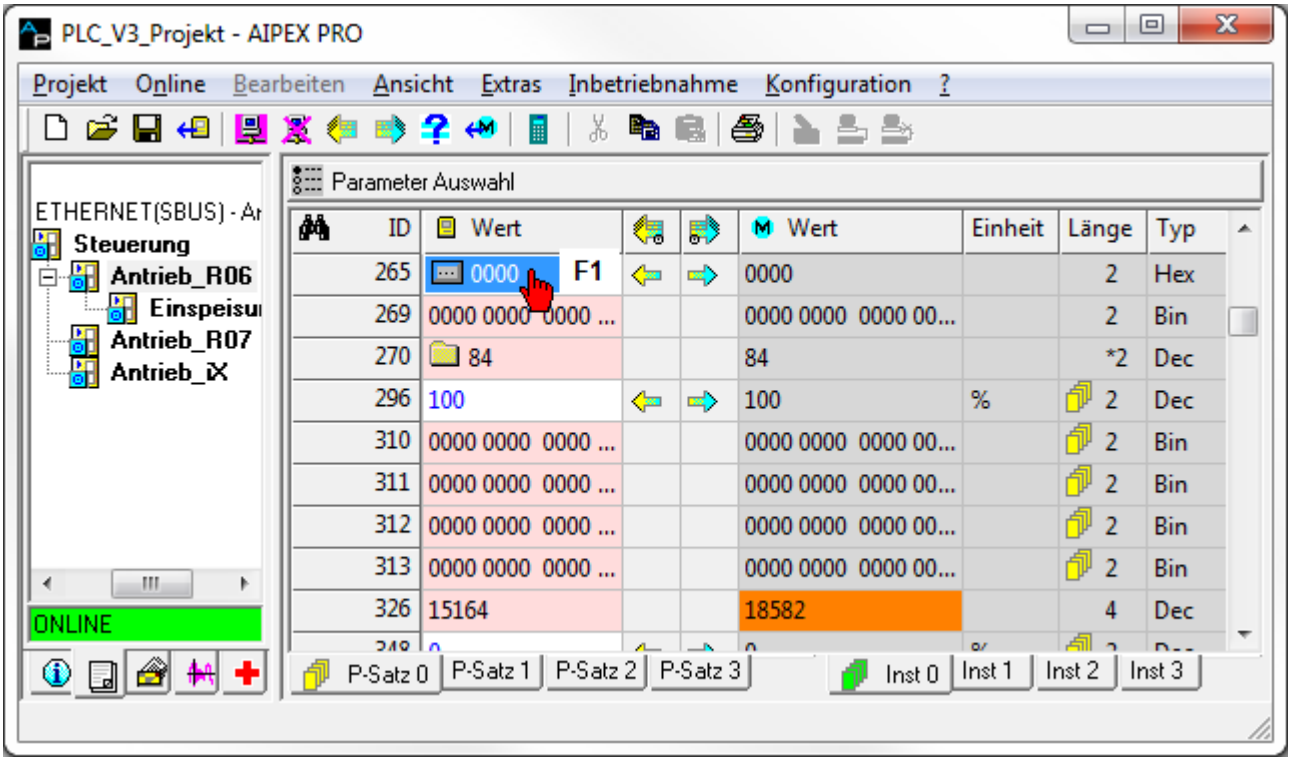


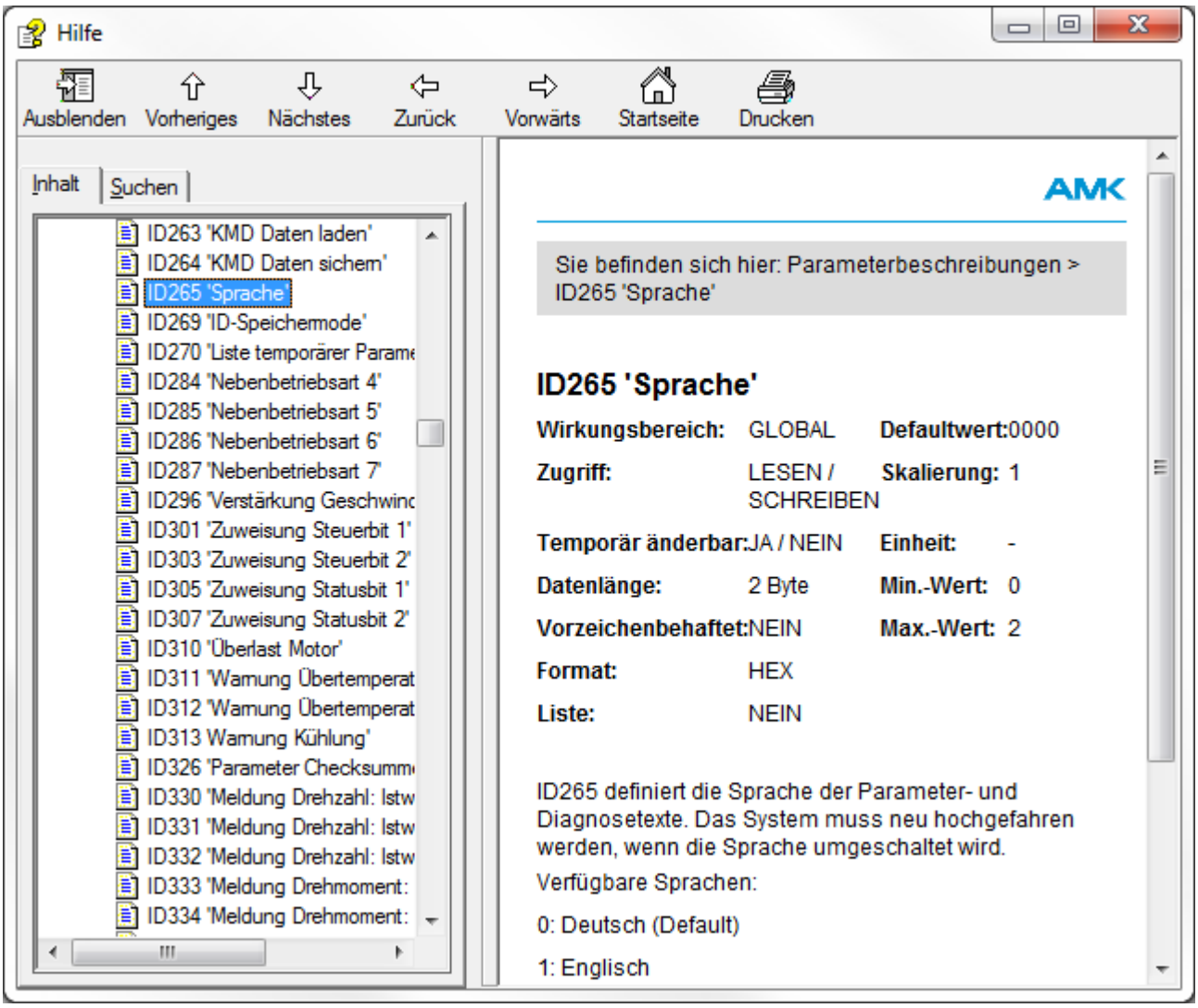
Auswahl	Beschreibung
alle Parameter	Alle Parameter werden angezeigt. (Systeminterne Parameter müssen zusätzlich angewählt werden)
einzelne Gruppe	Alle Parameter der angewählten Parametergruppe werden angezeigt.
eigene Liste	Alle Parameter der ausgewählten eigenen Liste werden angezeigt. Die eigenen Listen können beliebig erweitert werden. Jede Liste besteht aus einer durch Kommas getrennte Folge von ID-Nummern oder Nummerbereiche und kann durch eine frei wählbare Bezeichnung gekennzeichnet werden. Bsp.: Meine Liste 1, 2, 5, 90, 32000-32100 Jede Liste wird durch vollständiges Entfernen des Inhaltes wieder gelöscht.
Systeminterne Parameter	Systeminterne Parameter werden in der Parameterliste angezeigt (Farbe rot)
Änderungen zulassen	Nach dem Freischalten können systeminterne Parameter geändert werden.

3.5.2.3 Parameter - Online Hilfe

Aufruf der Parameter – Online Hilfe

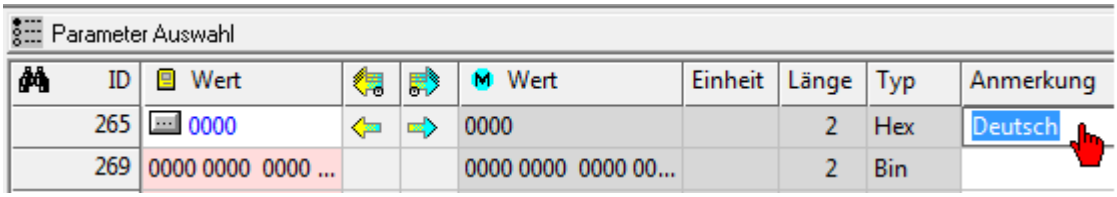
Markieren Sie einen Parameter. Drücken Sie die **'F1'** Taste.





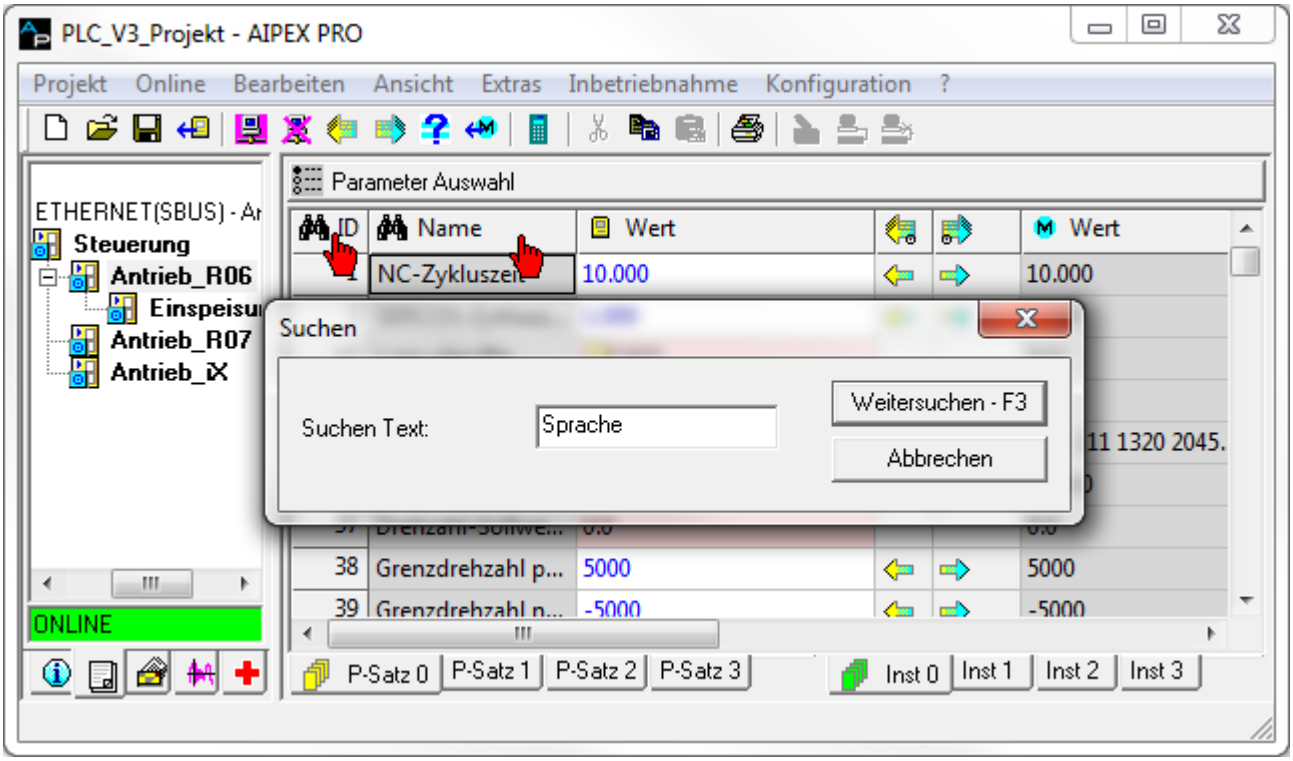
3.5.2.4 Parameter - Kommentare

In die letzte Spalte der Tabelle können beliebige Kommentare eingegeben werden.
Dies ist jedoch nur dann möglich, wenn ein 'Offline' Datensatz zur Verfügung steht, also nicht bei reiner 'Online' Arbeit.
Kommentare sind nicht nur für jede ID, sondern auch für jedes Listenelement in einer expandierten Liste möglich.

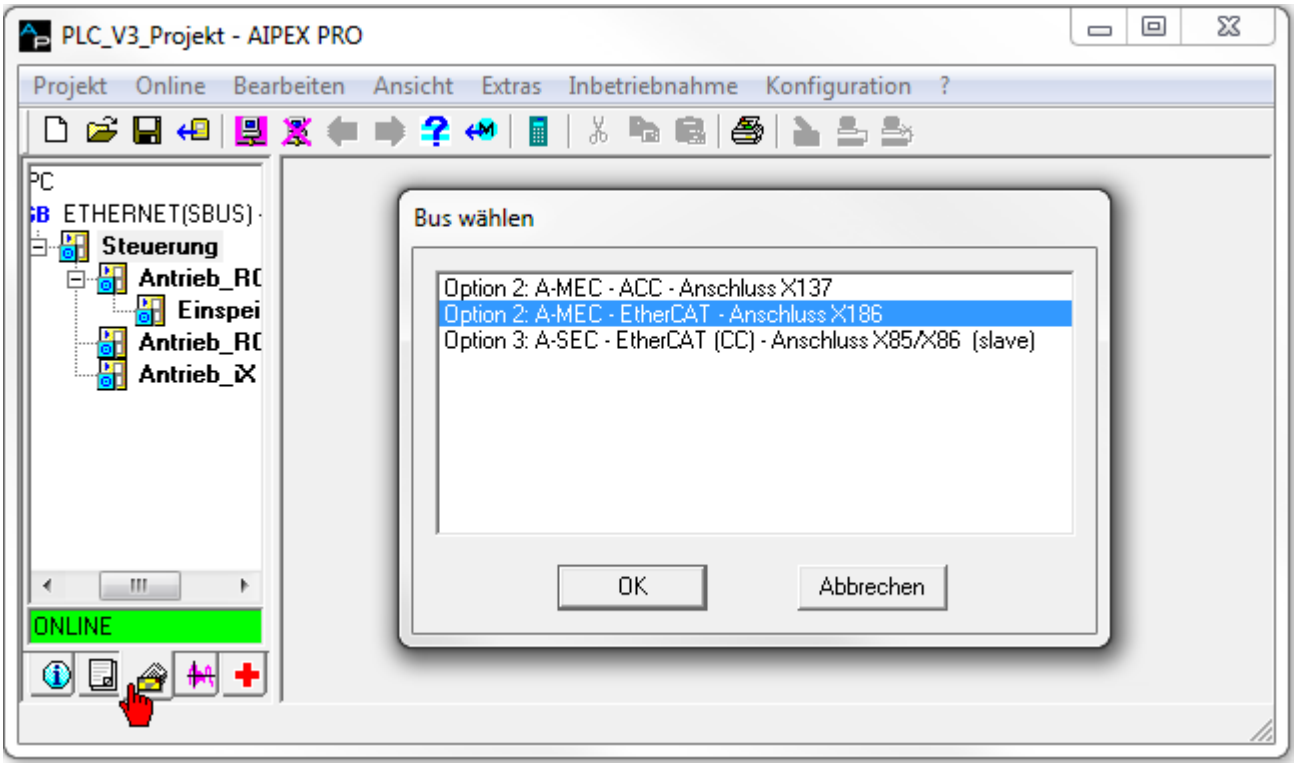


3.5.2.5 Parameter - Suchfunktion

Nutzen Sie die Suchfunktion (Symbol Fernglas), um in den Spalten nach ID-Nummern oder Texten zu suchen.



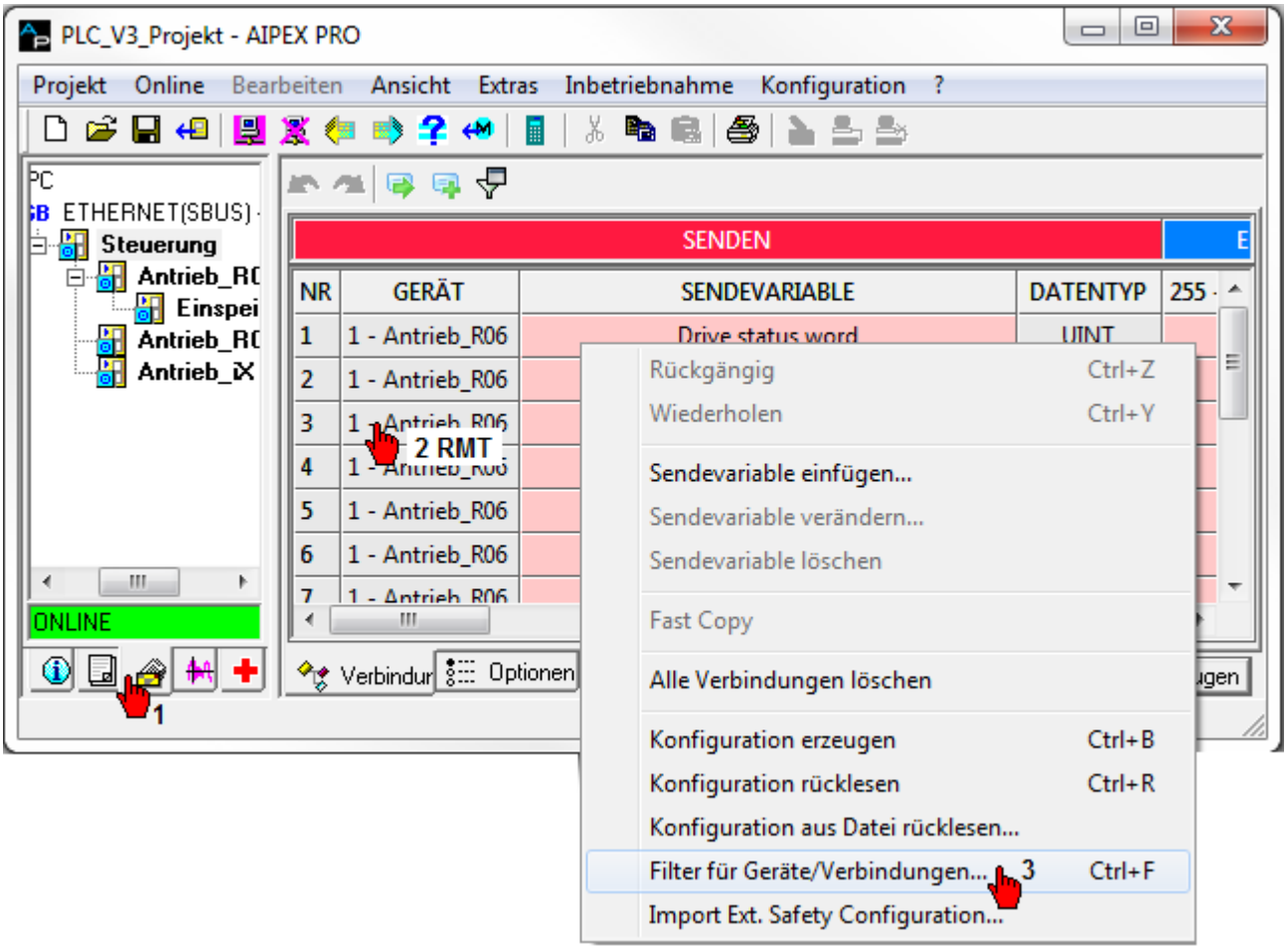
3.5.3 Nachrichten

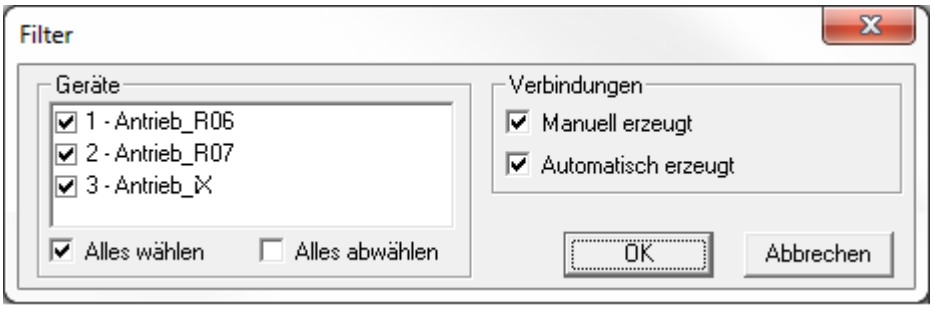


Durch die Kopplung von Komponentenkonfiguration und Programmierumgebung ist AIPEX PRO in der Lage die automatische Feldbuskonfiguration durchzuführen. Dabei werden Daten, je nach ihrem Verwendungszweck automatisch im PLC Programm, synchron oder asynchron bereitgestellt. Antriebsdaten als auch E/A-Daten werden automatisch konfiguriert.

3.5.3.1 Nachrichten - Anzeigefilter

Gestalten Sie den Nachrichtenkonfigurator übersichtlicher, indem Sie den Anzeigefilter auf Ihre Applikation anpassen. Klicken Sie dazu mit der rechten Maustaste in den Nachrichtenkonfigurator.





Geräte
De- und aktivieren Sie einzelne Geräte.

'Alles wählen'
Alle Geräte werden im Nachrichtenkonfigurator angezeigt.

'Alles abwählen'
Alle Geräte werden abgewählt

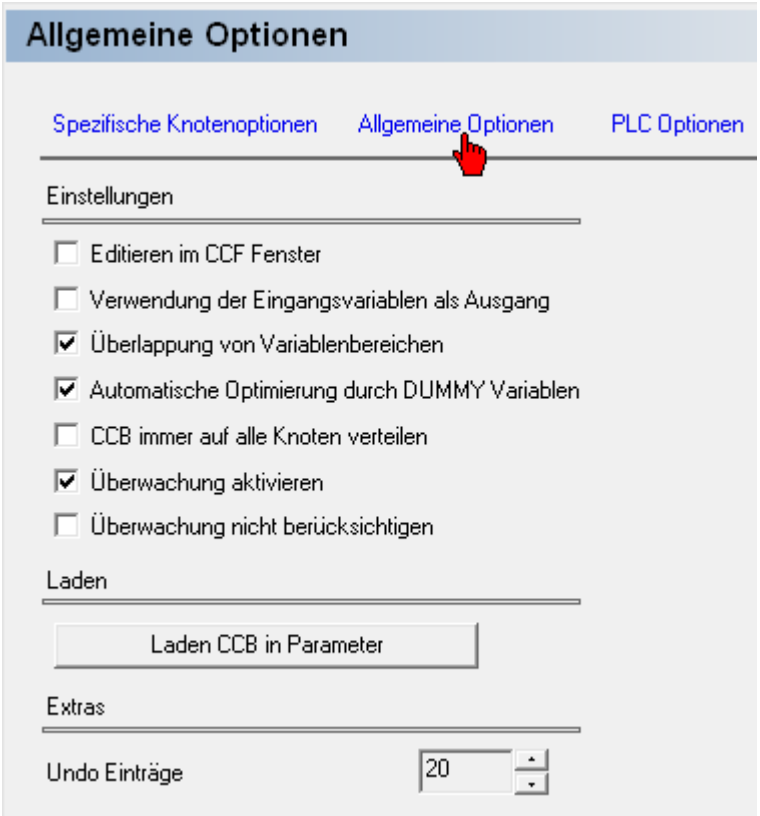
Verbindungen

'Manuell erzeugt'
Nur manuell vom Anwender erzeugte Verbindungen werden im Nachrichtenkonfigurator angezeigt.

'Automatisch erzeugt'
Nur automatisch von AIPEX PRO erzeugte Verbindungen werden im Nachrichtenkonfigurator angezeigt.

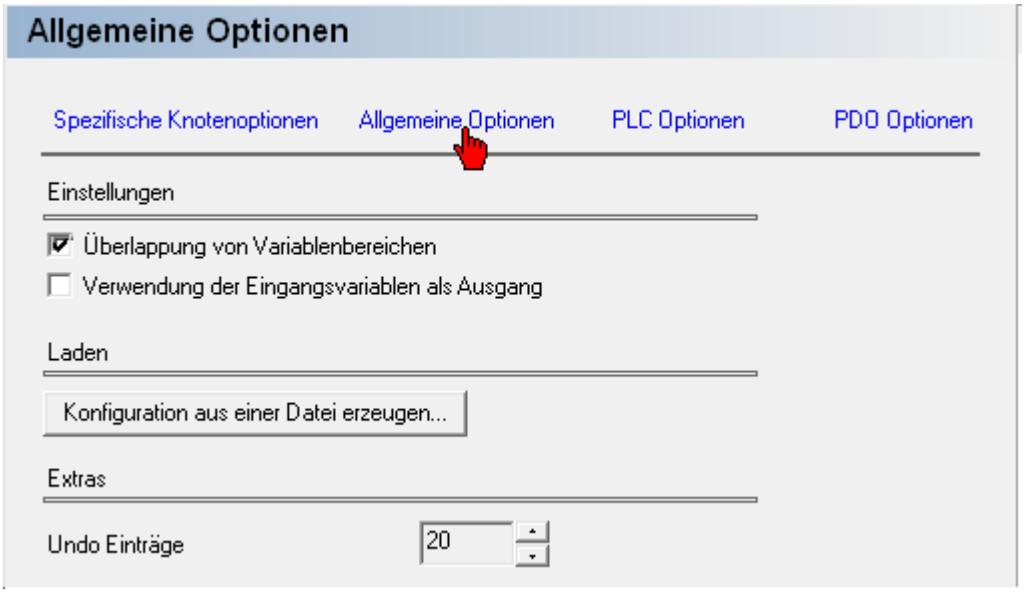
3.5.3.2 Nachrichten - Allgemeine Optionen

3.5.3.2.1 ACC-Bus



- **Editieren im CCF Fenster**. Ermöglicht eine direkte Bearbeitung der CCF Datei im Reiter 'CCF Ausgabe'
- **Verwendung der Eingangsvariablen als Ausgang**. Eingangsvariablen die in der EDS Datei beschrieben wurden, können Rückgelesen werden.
- **Überlappung von Variablenbereichen**. In der Registerkarte 'Verbindungen' werden überlappende Variablen nicht gefiltert.
- Automatische Optimierung durch DUMMY Variablen
- **CCB immer auf alle Knoten verteilen**. Aktivieren Sie diese Option wenn Sie eine CANclient Struktur ohne Master realisieren möchten.
- **Überwachung aktivieren**. Node guarding (Master) und Life guarding (Slave). Die Anwesenheit von Master und Slave am Bus wird gegenseitig geprüft. Bei einem fehlenden Knoten (Master oder Slave) wird eine Fehlermeldung generiert. Die benötigten Werte „Guard Time“ und „Life Time Faktor“ werden von AIPEX PRO automatisch generiert.
- **Überwachung nicht berücksichtigen**. Die Überwachungsfunktionen Node guarding (Master) und Life guarding (Slave) werden komplett deaktiviert. (Nutzbar bei externen Klemmen!)
- **Laden CCB in Parameter**. Bestehende *.eds Dateien können direkt in das Projekt geladen werden.

3.5.3.2.2 EtherCAT

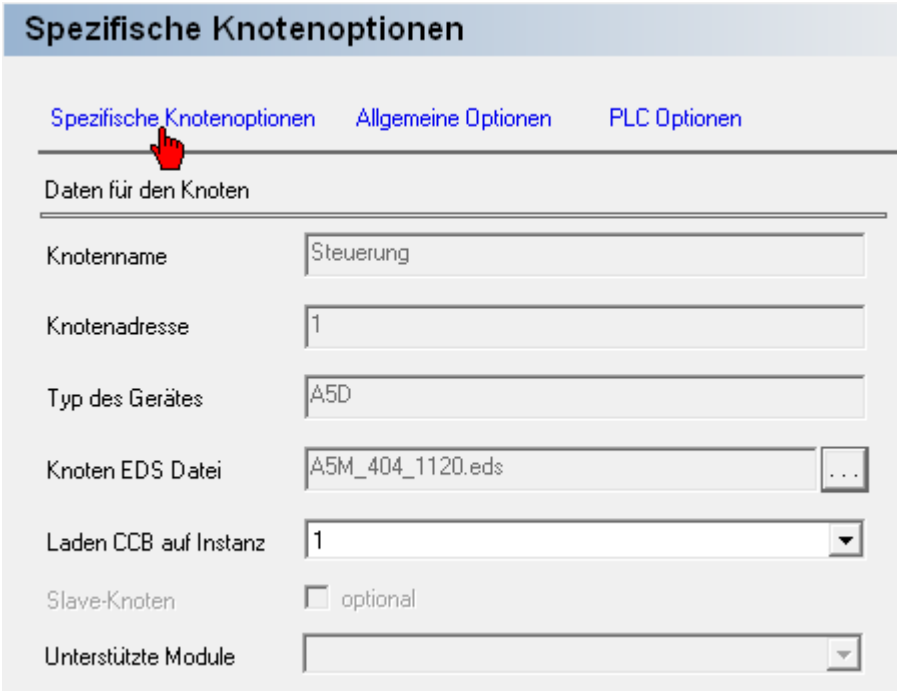


- **Überlappung von Variablenbereichen**. In der Registerkarte 'Verbindungen' werden überlappende Variablen nicht gefiltert.
- **Verwendung der Eingangsvariablen als Ausgang**. Eingangsvariablen die in der EDS Datei beschrieben wurden, können Rückgelesen werden.

Konfiguration aus einer Datei erzeugen. Bestehende *.XML Dateien können direkt in das Projekt geladen werden.

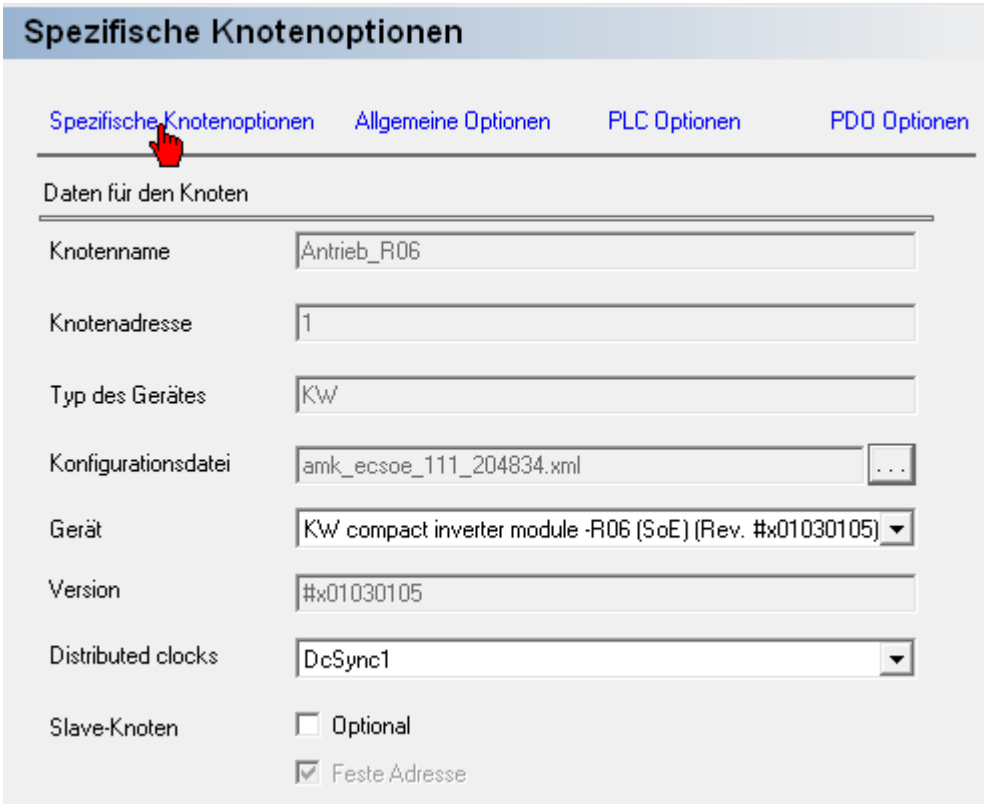
3.5.3.3 Nachrichten - Spezifische Knotenoptionen

3.5.3.3.1 ACC-Bus



- **Knotenname** . Name des selektierten Gerätes (nur Anzeige).
- **Knotenadresse** . Bus-Adresse des selektierten Gerätes (nur Anzeige).
- **Typ des Gerätes**. Art des selektierten Gerätes (nur Anzeige).
- **Knoten EDS Datei**. Name der Konfigurationsdatei des selektierten Gerätes. Dieser Eintrag kann durch Auswahl einer anderen Datei verändert werden.
- **Laden CCB auf Instanz**. Das Feld “Laden auf Instanz” gibt an, auf welche Instanz die erzeugte CCB– Datei geladen wird (in ID34036). Dieses Feld ist nur für das CAN Hauptgerät aktiv, da es die Konfiguration für das gesamte Netzwerk lädt. Für KW ist der voreingestellte Wert 0 und für AMKAMAC AS-PL Steuerungen ist er 1.
- **Slave Knoten - optional**. Slaves die als optional gekennzeichnet sind müssen bei der Initialisierung des Systems nicht anwesend sein. Es wird keine Fehlermeldung ausgegeben.

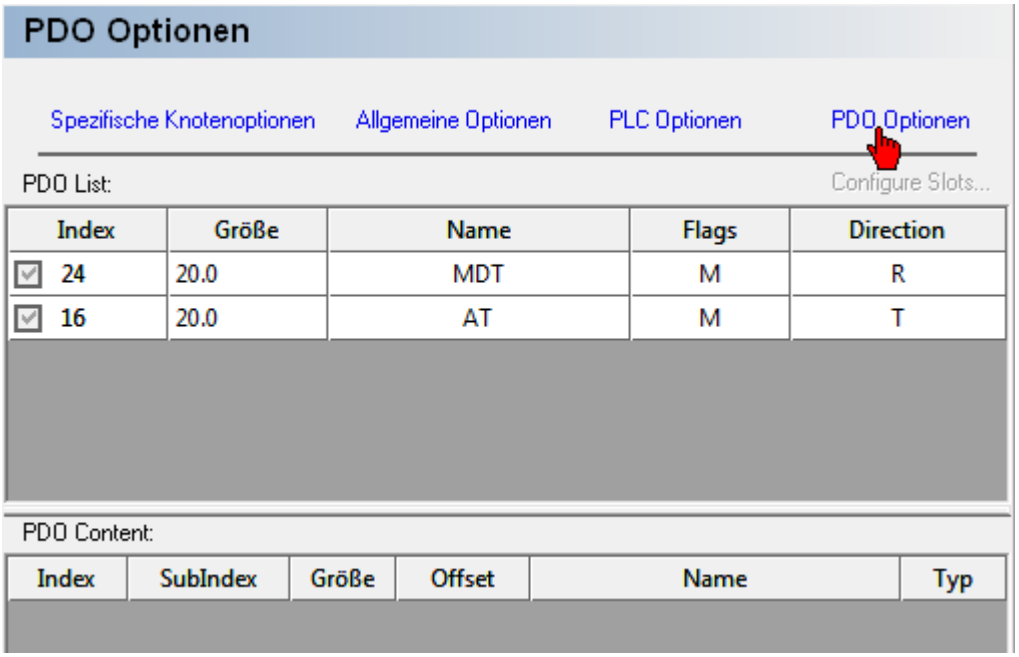
3.5.3.3.2 EtherCAT



- **Knotenname** . Name des selektierten Gerätes (nur Anzeige).
- **Knotenadresse**. Bus-Adresse des selektierten Gerätes (nur Anzeige).
- **Typ des Gerätes**. Art des selektierten Gerätes (nur Anzeige).
- **Konfigurationsdatei**. Name der Konfigurationsdatei des selektierten Gerätes. Dieser Eintrag kann durch Auswahl einer anderen Datei verändert werden.
- **Gerät**. Das Gerät muss zur Konfigurationsdatei passen. Wird eine neue Konfigurationsdatei gewählt, muss jeweils das dazugehörige Gerät ausgewählt werden.
- **Slave Knoten - optional**. Slaves die als optional gekennzeichnet sind müssen bei der Initialisierung des Systems nicht anwesend sein. Es wird keine Fehlermeldung ausgegeben. Voraussetzung ist, dass für diesen Knoten die Option „Feste Adresse“ gesetzt ist.
- **Slave Knoten - Feste Adresse**. Ein damit gekennzeichneter Knoten bekommt durch den Master prinzipiell die vorgegebene feste Adresse. Diese Option ist bei AMK Geräten immer aktiv.

3.5.3.4 Nachrichten - PDO Optionen

EtherCAT



Es werden alle in der Beschreibungsdatei des Gerätes enthaltenen PDOs aufgelistet.

- „Index“ - eindeutige numerische Kennzeichnung
- „Größe“ - Größe der PDO in Bytes.Bits
- „Name“ - Bezeichnung der PDO (freier Text)
- „Flags“ - F:fix, V:virtuell, M:obligatorisch (mandatory)
- “Direction” - T: transmit, R: receive

Die verwendeten PDOs können mit folgenden Einschränkungen gewählt / abgewählt werden:

- Obligatorische PDOs können nicht abgewählt werden
- Excluded PDOs können nur gewählt werden, wenn die Gegenstücke abgewählt sind

Für die in der PDO Liste selektierte PDO wird in der unteren Tabelle der PDO Inhalt angezeigt:

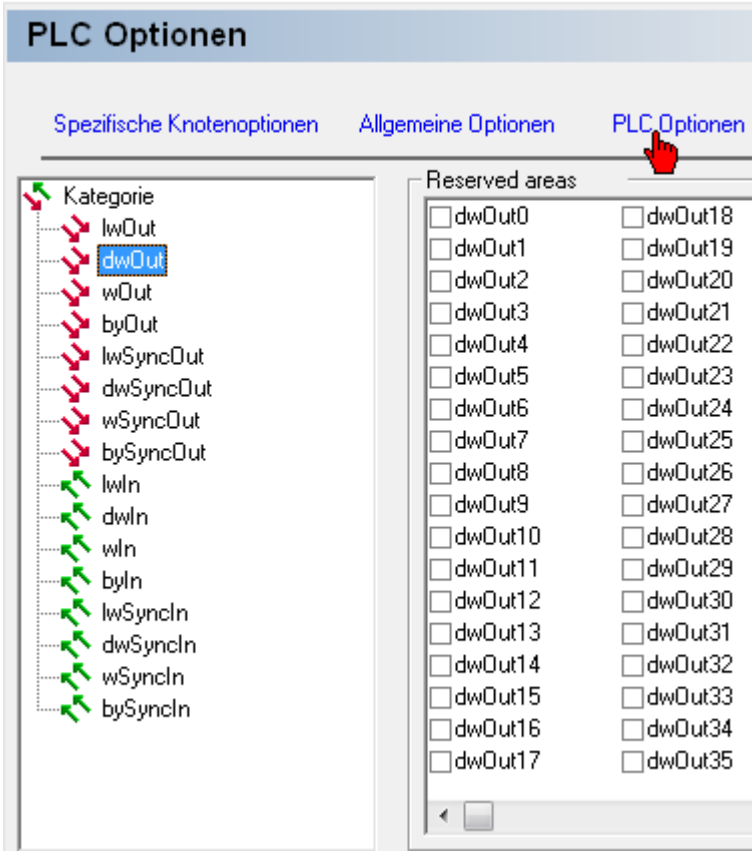
- „Index“/“SubIndex“ - eindeutige numerische Kennzeichnung der Variablen
- „Größe“ - Größe der Variablen in Bytes.Bits
- „Offset“ - Lage der Variablen in der PDO
- „Name“ - Bezeichnung der Variablen (freier Text)
- „Typ“ - Datentyp der Variablen

Bei nicht fixen PDOs kann der PDO Inhalt über das Kontextmenü verändert werden. Es können z. B. neue Elemente hinzugefügt werden.

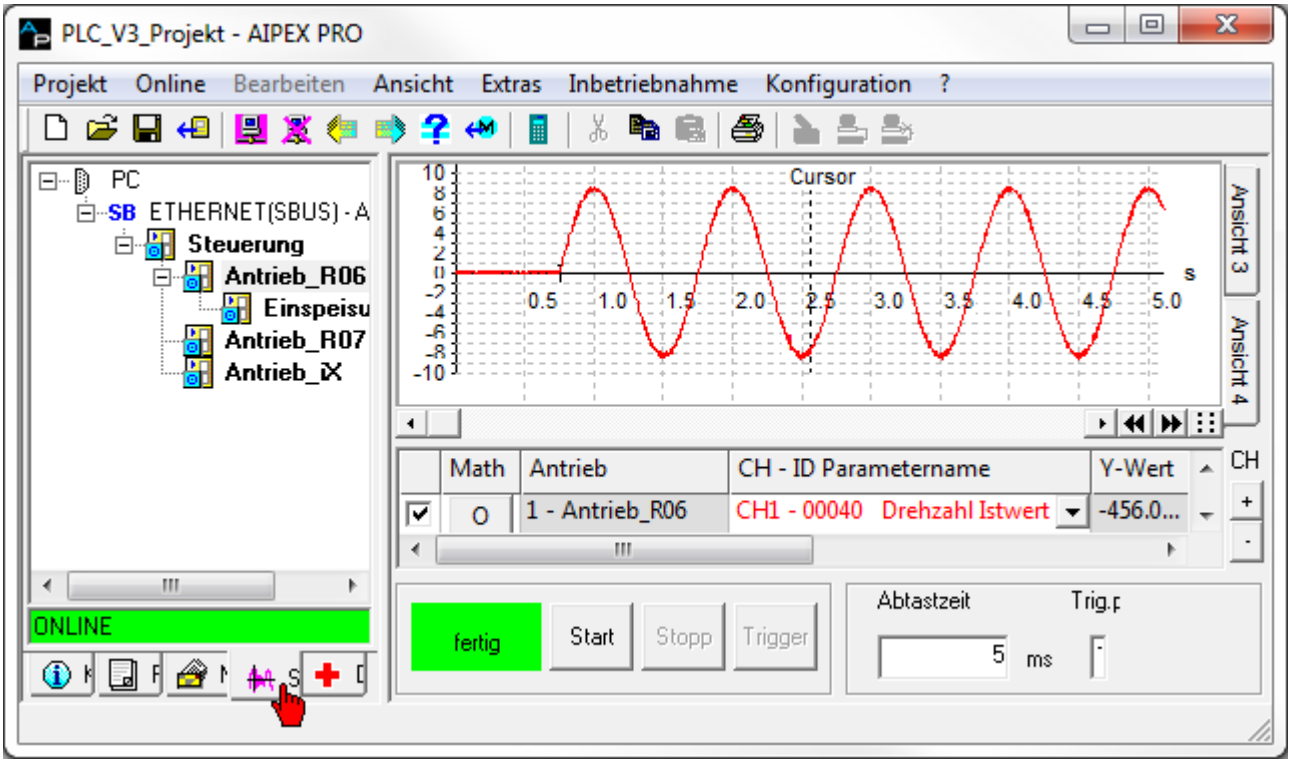
3.5.3.5 Nachrichten - PLC Optionen

Angewählte (reservierte) Ausgangsvariablen werden durch den automatischen Nachrichtenkonfigurator nicht belegt.
Die Einstellungen gelten nur für das im Gerätebaum angewählte Gerät.

Beispiel:
Profibus in Verbindung mit einer AMK PB Optionskarte verwendet feste Adressbereiche. Diese Adressbereiche dürfen daher nicht von der automatischen Netzwerkkonfiguration belegt werden.



3.5.4 Scope



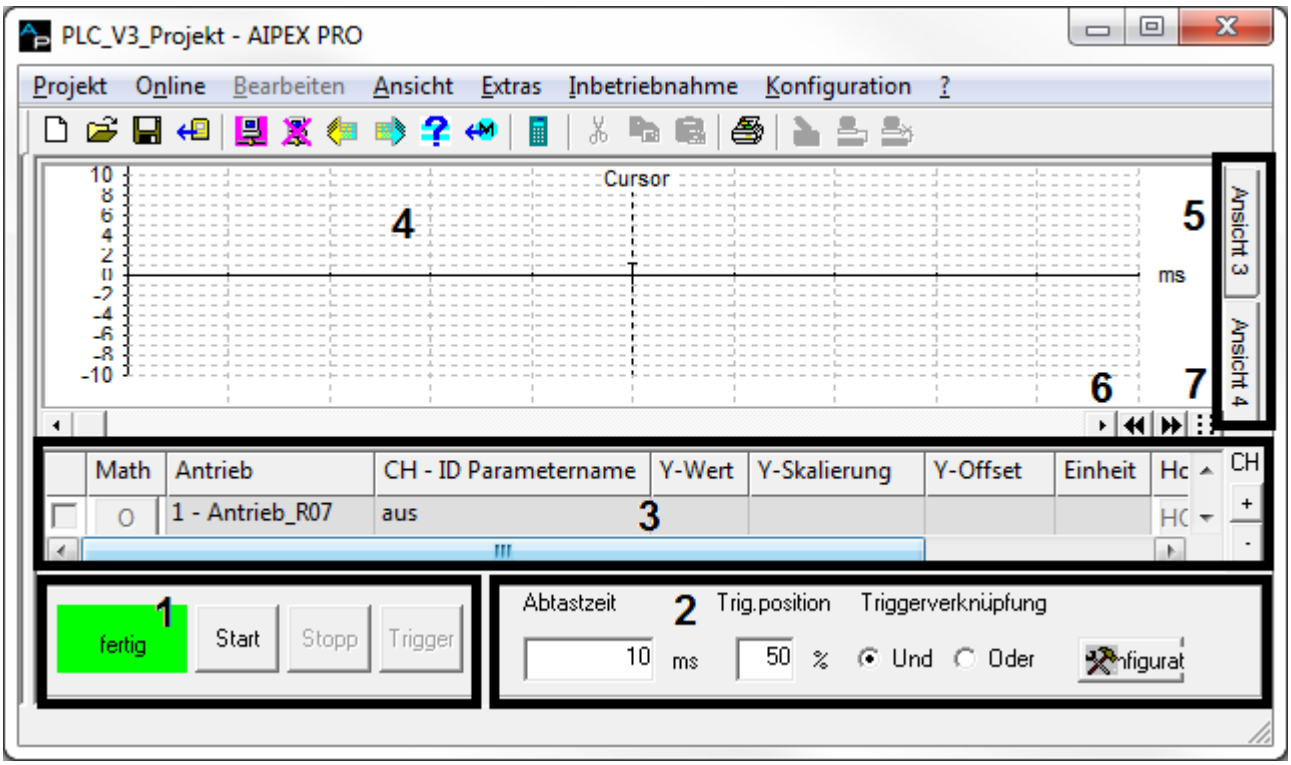
Jeder AMK Antrieb verfügt über eine interne Oszilloskopfunktion. Mit AIPEX PRO können Sie Signale konfigurieren, die anschließend in Echtzeit im Antrieb aufgezeichnet werden. Sobald der interne Datenspeicher im Antrieb voll ist, werden die Messdaten übertragen und grafisch mit AIPEX PRO dargestellt.

3.5.4.1 Scope - Anleitung

Sie können antriebsspezifische oder geräteübergreifende Messungen durchführen (nur ACC-Bus).

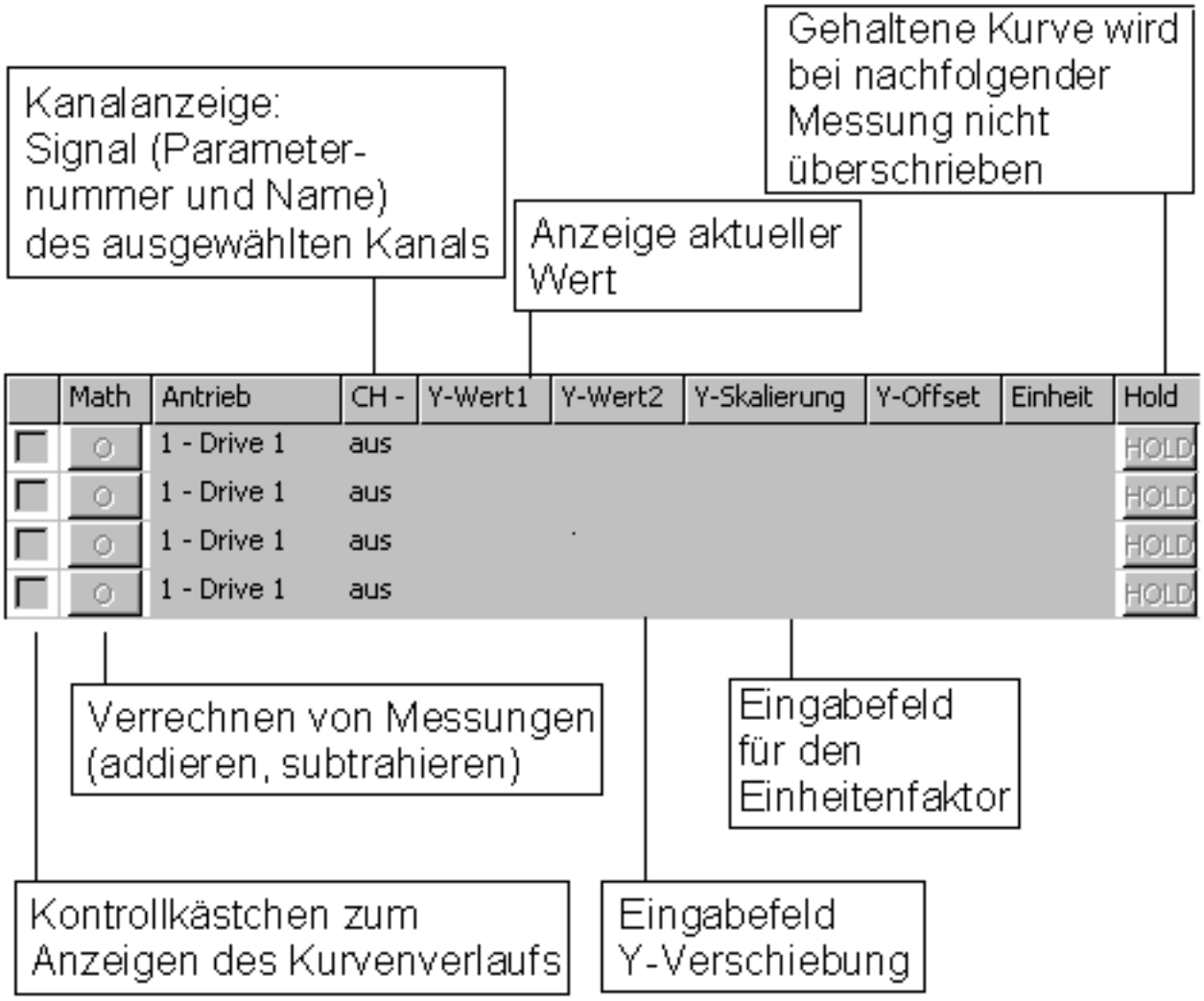
Markieren Sie bei antriebsspezifischen Messungen ein Antrieb im Gerätebaum.

Bei der geräteübergreifenden Messung markieren Sie den Feldbus (z. B. ACC-Bus) danach können Sie Signale von mehreren Geräten aufzeichnen und darstellen. Die Messung wird mit einem gemeinsamen Triggersignal in allen Antrieben gestartet.



Nummer	Funktion
1	Bedienung Oszilloskop (Start, Stop und Trigger Button) Darstellung Status Offline (grau) Bereit (grün) Gestartet (gelb) Getriggert (gelb) Fertig (grün) Fehler (rot)
2	Konfigurationsfeld
3	Feld für Antriebsauswahl und Signalparameter
4	Display
5	Jede Ansicht hat eine eigene Kanalauswahl
6	Zeitachse skalieren
7	Gitternetz im Display (an / aus)

Nummer	Funktion
1	Eingabefeld für die Abtastzeit der nachfolgenden Messungen (0.5ms bis 20ms)
2	Triggerposition. (Kurvenverlauf vor dem Trigger-Ereignis („pre trigger“) kann angezeigt werden.
3	Bei Verwendung mehrerer Triggersignalen kann zwischen einer 'Und' oder 'Oder' - Verknüpfung der Triggersignale gewählt werden
4	Button zum öffnen des Konfigurationsfeldes
5	Anzeige des Datums und der Uhrzeit nach einer Messung
6	Button zum Öffnen und Eingeben einer Information zur Messung (Button gelb hinterlegt bedeutet Text Information vorhanden)



Statusanzeigen

Statusanzeige 'Grau' - es ist kein gültiges Gerät ausgewählt oder es wird Offline gearbeitet

Status 'bereit' - Oszilloskop ist bereit zum Start

Status 'gestartet' - Konfiguration der Kanäle ist gültig, Oszilloskop gestartet. Die aktuellen Werte werden angezeigt.

Status 'getriggert' - Trigger erfüllt, oder manuell mit der Taste 'Trigger' ausgelöst. Die Daten werden aus dem Antrieb geholt.

Status 'fertig' - Messung ist beendet, und die Kurven werden aufgezeichnet.

Status 'INVALID' - Konfiguration der Kanäle ist ungültig.
Das Oszilloskop unterstützt ein Signal auf einem konfigurierten Kanal nicht, d. h. dieses Signal ist zu korrigieren bzw. zu deaktivieren.

Status 'FEHLER' - kann sowohl durch einen Fehler im Oszilloskop oder auch durch einen anderen Fehler ausgelöst werden. Die genaue Ursache kann über die AIPEX Registerkarte „Diagnose“ ermittelt werden.

3.5.4.2 Scope - Maximale Aufzeichnungszeit

Die maximale Aufzeichnungszeit wird durch die variable Größe des Datenspeichers, sowie die konfigurierten Messsignale und der Abtastzeit beeinflusst.

Der Datenspeicher im Antrieb kann mit der ID34284 'OSC Container Länge' konfiguriert werden.
Der Standardwert beträgt 4096 Byte, der maximale Wert 32600 Byte.

Die maximale Aufzeichnungszeit wird mit folgender Formel bestimmt:

Beispiel zur Bestimmung der maximalen Aufzeichnungszeit:

Parameter: (AIPEX PRO 'Parameter')	ID34284 'OSC Container Länge'	4096 Byte
Messsignale: (AIPEX PRO 'Scope')	ID40 'Drehzahl Istwert', Länge 4 Byte* ID84 'Moment Istwert', Länge 2 Byte* ID380 'Zwischenkreisspannung', Länge 2 Byte*	(Summe) 8 Byte
	Abtastzeit	10 ms

* Die Länge in Byte kann in der Spalte 'Länge' (AIPEX PRO Reiter 'Parameter') des jeweiligen Parameters entnommen werden.

Maximale Aufzeichnungszeit =

Die Initialisierungszeit der Oszilloskopfunktion im Antrieb, ausgelöst durch das betätigen des Tasters 'Start' enspricht ungefähr der maximalen Aufzeichnungszeit.

3.5.4.3 Interpretation der Lagesollwerte und Lageistwerte

Oszilloskop zum EtherCAT Bus synchronisieren

Für eine synchrone Aufzeichnung muss die Abtastzeit des Oszilloskop gleich der ID2 'SERCOS-Zykluszeit' gewählt werden.
Bei Verwendung des Interpolators (z. B. bei einer Referenzpunktfahrt) muss für die Abtastzeit 1 ms gewählt werden.

Interpretation der im Oszilloskop angezeigten Lagewerte

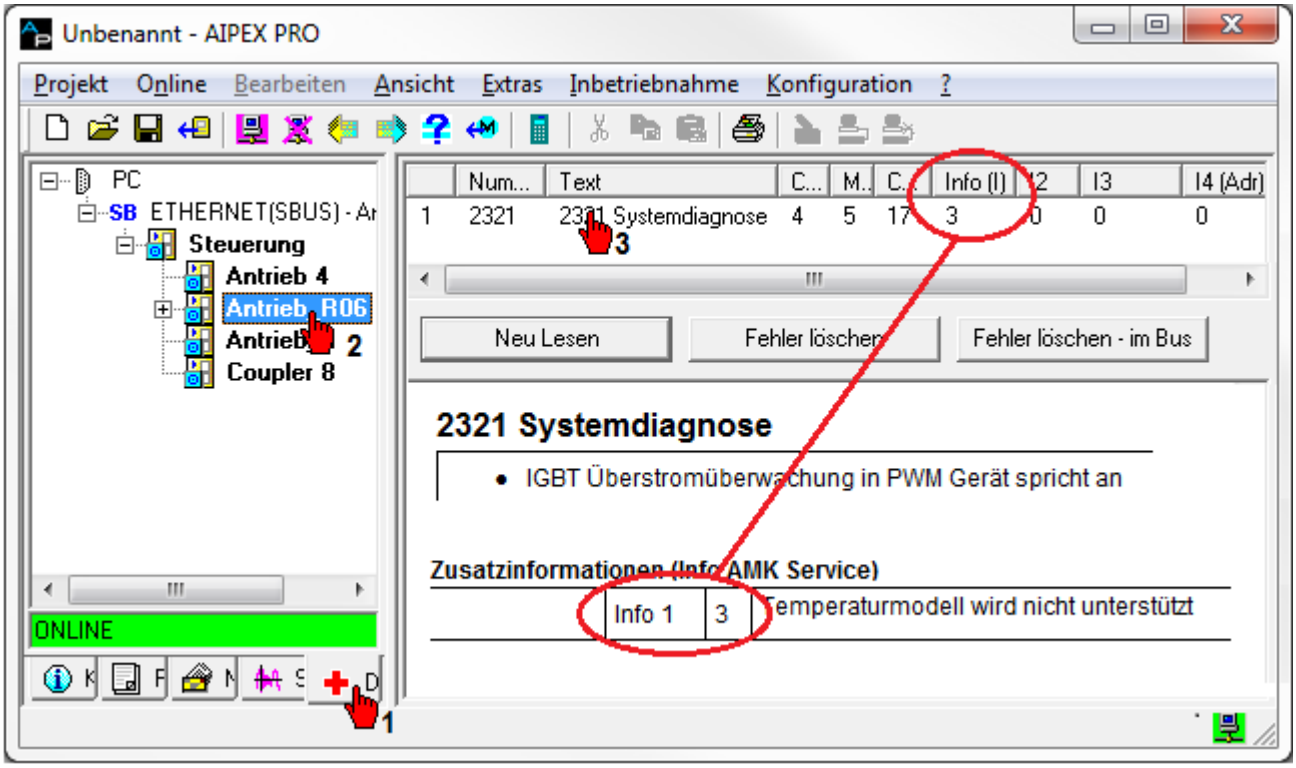
Prinzipbedingt eilt der angezeigte Lagesollwert (ID47) dem angezeigten Lageistwert (ID51) um 2 Messzyklen voraus.

Eine Berechnung vom Schleppabstand im gleichen Messzyklus führt zu einem verfälschten Wert und unterscheidet sich von dem angezeigten Wert ID189 'Schleppabstand'.

Interne AMK Berechnung für den Schleppabstand (ID189)

Schleppabstand(t_x) = Lagesollwert (t_{x-2}) - Lageistwert (t_x)

3.5.5 Diagnose



Mit 'Diagnose' können die Diagnosemeldungen im angewählten Gerät ausgelesen werden.

Klicken Sie eine Diagnosemeldung an, um die Erklärung einzublenden.
Weiterführende Informationen zur Diagnosemeldung erhalten Sie über die Info(I), I2 und I3
Die erste Diagnosemeldung in der angezeigten Liste ist der Hauptauslöser der Störung, weitere Meldungen können Folgefehler sein, die nach Beseitigung der Ursache für die erste Diagnosemeldung nicht mehr erscheinen.

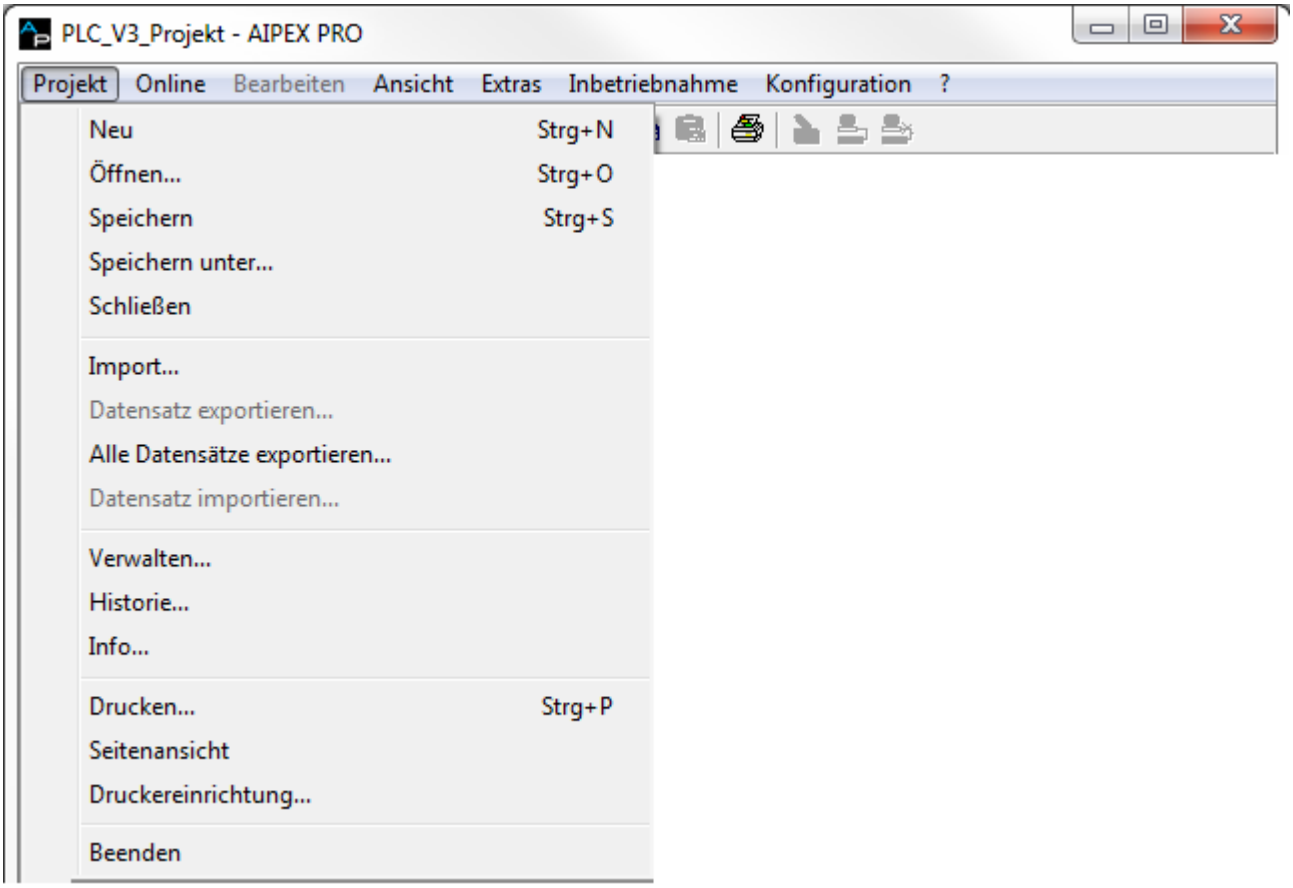
Taster **'Neu Lesen'**
Diagnosemeldungen werden aus dem angewählten Gerät ausgelesen.

Taster **'Fehler löschen'**
'Fehler löschen' wird im angewählten Gerät ausgeführt.

Taster **'Fehler löschen - im BUS'**
Alle Fehler werden im angewählten Bus gelöscht. Wählen Sie dazu das Bussystem im Gerätebaum an.

3.6 Menüleiste

3.6.1 Projekt



Neu [Strg + N]

Icon
Das geöffnete Projekt wird geschlossen und ein neues Projekt wird angelegt.

Öffnen... [Strg + O]

Icon
AIPEX PRO und AIPEX/AIPAR Datensätze können geöffnet werden.

Speichern [Strg + S]

Icon
Wenn Sie eine Projekt erstmalig speichern, wird automatisch das Dialogfeld **'Speichern unter...'** geöffnet.

Speichern unter...
Wählen Sie einen Speicherort. Geben Sie Ihren Projektnamen ein.

Schließen
Das geöffnete Projekt wird geschlossen. Wählen Sie anschließend **'Projekt'** → **'Neu'** aus um fortzufahren.

Import

Icon

Einzelne Gerätedaten aus einem Projektdatensatz können importiert werden.
Wählen Sie Ihren Projektdatensatz an.

Datensatz exportieren...
Mit der Funktion **'Datensatz exportieren...'**, wird vom aktuell angewählten Gerät eine XML Datei erzeugt und exportiert.

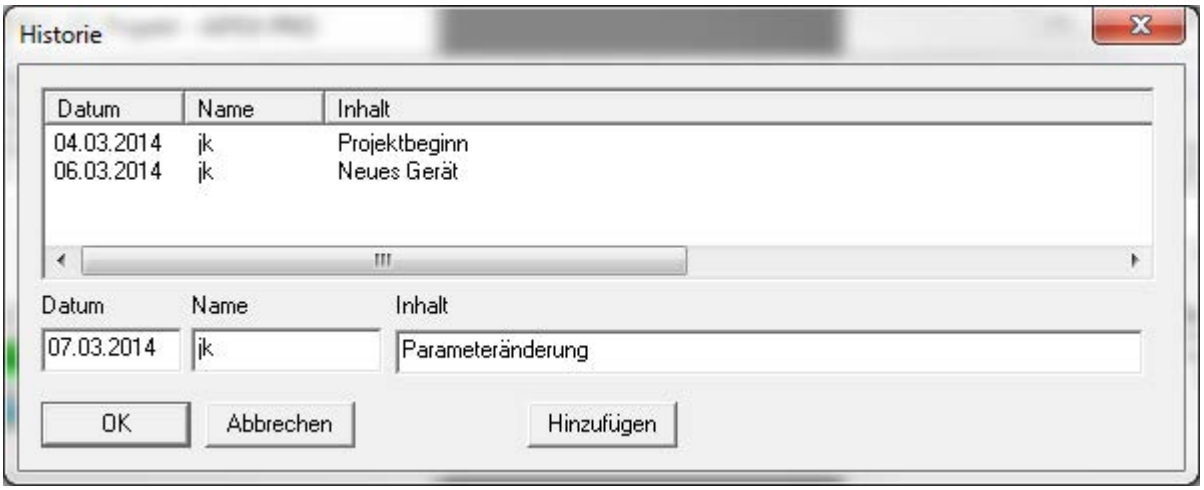
Alle Datensätze exportieren...
Mit der Funktion **'Alle Datensätze exportieren...'**, wird eine XML Datei erzeugt und exportiert, dass alle im Projekt enthaltenen Geräten enthält.

Datensatz importieren...
Mit der **'Funktion Datensatz importieren...'** importieren werden alle in der Datei (externe XML Datei) vorhandenen und schreibbaren Parameter importiert.

Verwalten...
Mit **'Verwalten...'** können Sie komplette AIPEX PRO Projektdatensätze löschen.

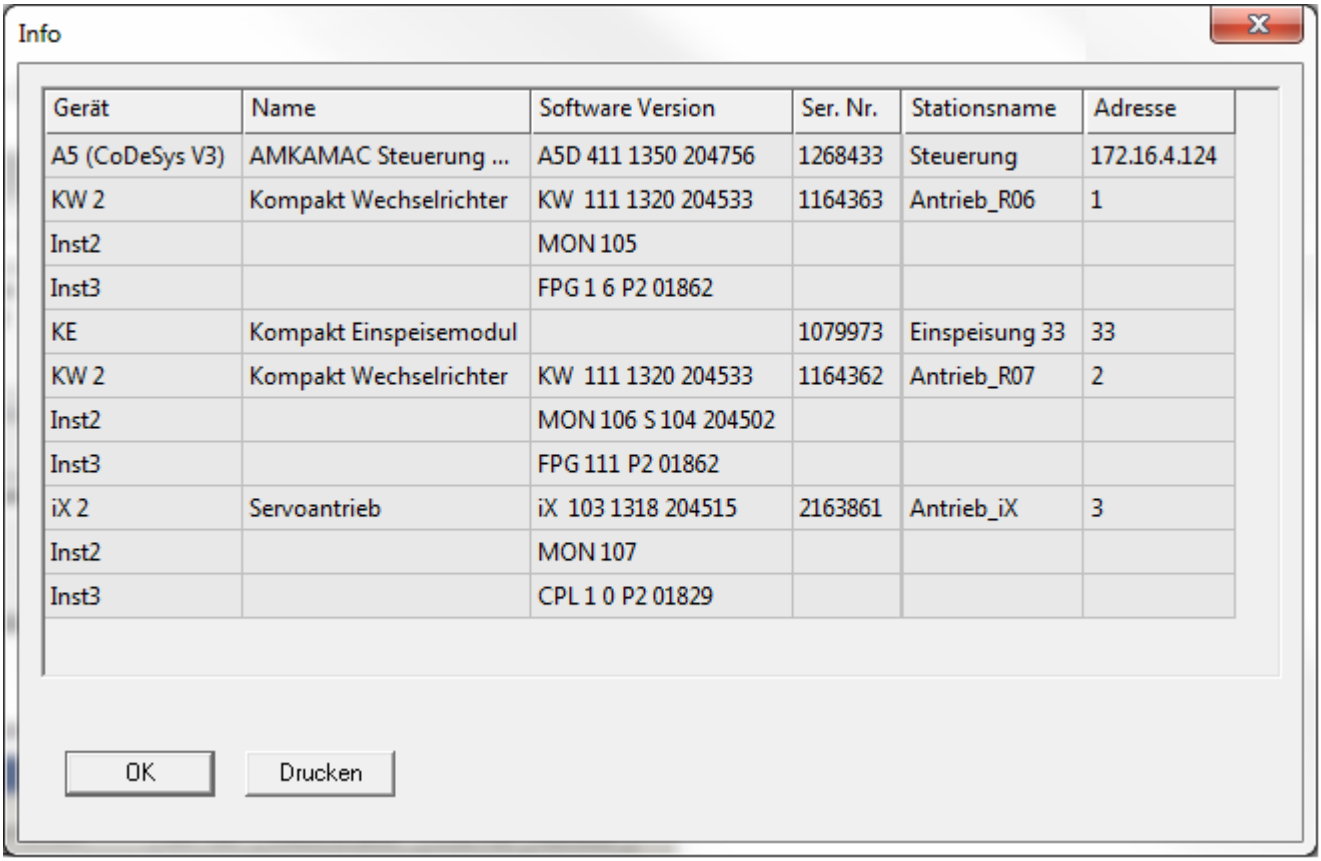
Historie...
Erstellen Sie für Ihr Projekt eine Historie.

Hinzufügen von Einträgen:
Tragen Sie 'Datum', 'Name' und 'Inhalt' ein. Drücken Sie zur Bestätigung den Taster **'Hinzufügen'**.



Info...

Das Fenster Info gibt detaillierte Informationen über alle Geräte wieder, die sich im Projekt befinden.



Drucken... [Strg + P]

Icon

Mit der Funktion **'Drucken...'** können die angezeigten Daten im Fenster **'Parameter'**, **'Nachrichtenkonfiguration'** und **'Scope'** gedruckt werden.

Seitenansicht

Mit der Funktion **'Seitenansicht'** werden die Daten im Fenster **'Parameter'**, **'Nachrichtenkonfiguration'** und **'Scope'** als Seitenansicht angezeigt. Anschließend kann die Seitenansicht gedruckt werden.

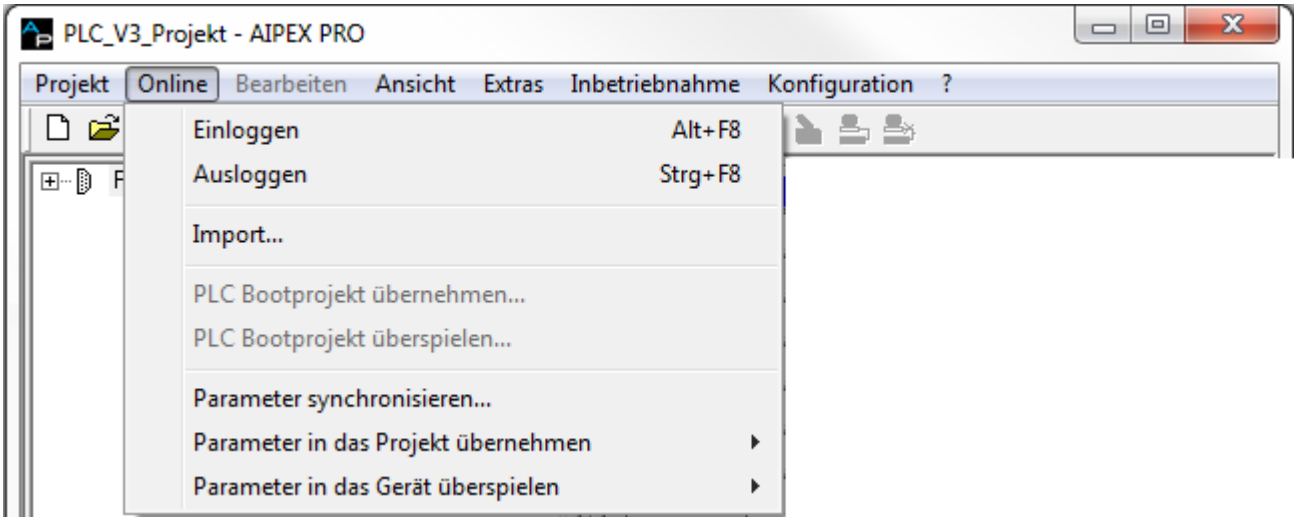
Druckereinrichtungen...

Mit der Funktion **'Druckereinrichtungen...'** kann ein Drucker eingerichtet werden.

Beenden

Das Projekt wird geschlossen und AIPEX PRO beendet.

3.6.2 Online



Einloggen [Alt + F8]

Icon

AIPEX PRO nimmt bei einer aktiven Schnittstelle Verbindung zu den AMK Geräten auf.
Alle Geräte die beim Abtasten erkannt werden, erscheinen im Fenster 'Import'.

Ausloggen [Strg + F8]

Icon

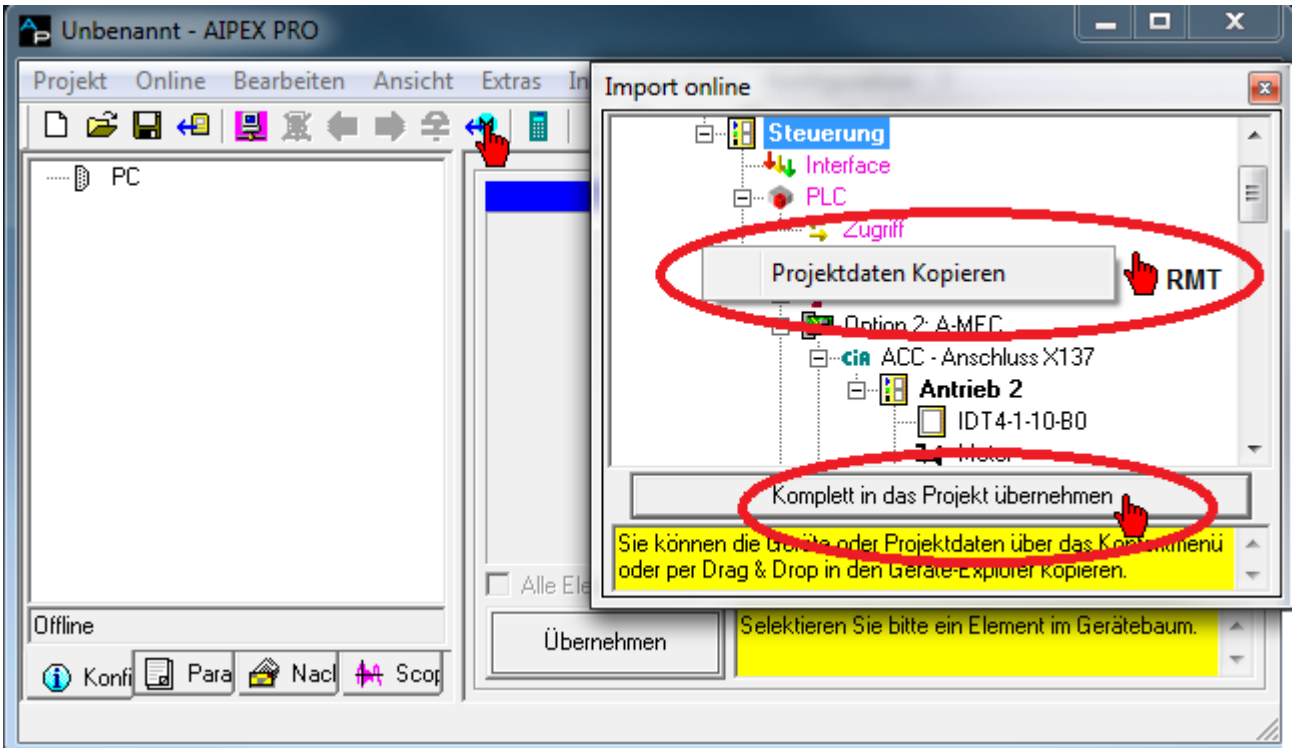
Schließt die Online Verbindung zu den AMK Geräten.

Import

Icon

Klicken Sie mit der rechten Maustaste in das geöffnete Fenster 'Import online'. Mit der Auswahl **'Komplett in das Projekt übernehmen'** werden alle Gerätedaten übertragen.
Mit der Maus (Drag and Drop) können einzelne Online Geräte aus dem Importfenster in das aktuelle Projekt importiert werden.

Die Busstruktur muss immer eingehalten werden. (PC Anschluss - Busmaster - Antriebsbus - Antriebe). Bevor Sie z.B. einen einzelnen Antrieb in ein leeres Projekt importieren können, müssen Sie zuerst den PC Anschluss und ein Busmaster manuell einfügen.



PLC Bootprojekt übernehmen...

Das PLC Bootprojekt wird von der angewählten Steuerung in den AIPEX PRO Projektdatensatz übernommen.

Das PLC Bootprojekt wird im Parameter ID34159 'PLC Dateien' gespeichert.

PLC Bootprojekt überspielen...

Das PLC Bootprojekt wird vom AIPEX PRO Projektdatensatz in die angewählte Steuerung überspielt.

Das PLC Bootprojekt wird im Parameter ID34159 'PLC Dateien' gespeichert.

Parameter synchronisieren...

Icon

Die Funktion **'Parameter synchronisieren...'** zeigt an, ob der Offline Datensatz (PC) mit dem Online Datensatz (Gerät) identisch ist.

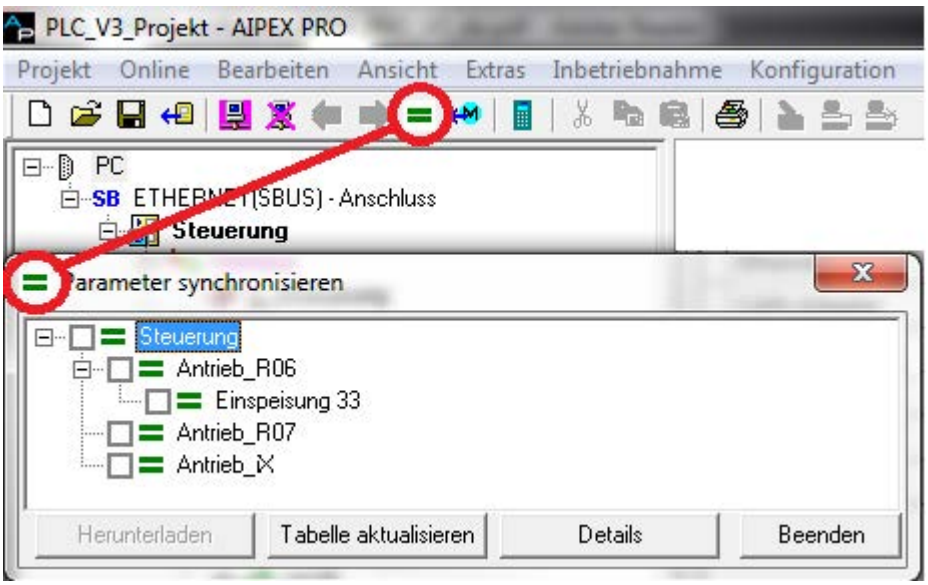
Mit dem Taster **'Herunterladen'** werden die Offline Daten auf die angewählten Geräte gespielt.

(Funktion ist identisch mit **'Online'** → **'Parameter in das Gerät überspielen'**)

Mit dem Taster **'Tabelle aktualisieren'** werden die Online Daten gelesen und mit den Offline Daten verglichen.

Mit dem Taster **'Details'** werden die Unterschiede zwischen Online und Offline Datensatz angezeigt.

Der Aufruftaster der Funktion **'Parameter Synchronisieren'** zeigt (Und-verknüpft) den Gerätezustand an.



Bedeutung der Symbole

Ungleich (Control): Online und Offline Datensatz sind nicht identisch.

Fragezeichen (Drive1): Online Daten stehen noch nicht zu Verfügung.(Taste **'Fertig lesen'** betätigen)

Gleich (Drive2): Online und offline Datensatz sind identisch.

Parameter in das Projekt übernehmen

Die Funktion kann nur aufgerufen werden, wenn zuvor **'Einloggen'** erfolgreich ausgeführt wurde.

Die aktuellen Gerätedaten eines physikalisch vorhandenen Geräts (Online Gerät) werden in den Projektdatensatz (Offline Geräte) übernommen. Physikalisch vorhandene Geräte die im Projektdatensatz nicht vorhanden sind werden nicht als Offline Gerät ergänzt.

'Parameter in das Projekt übernehmen' → **'vom aktuellen Gerät'**: Es wird nur das im Gerätebaum selektierte Gerät berücksichtigt

'Parameter in das Projekt übernehmen' → **'von allen Geräten'**: Es werden alle im Gerätebaum existierenden Geräte berücksichtigt

Parameter in das Gerät überspielen

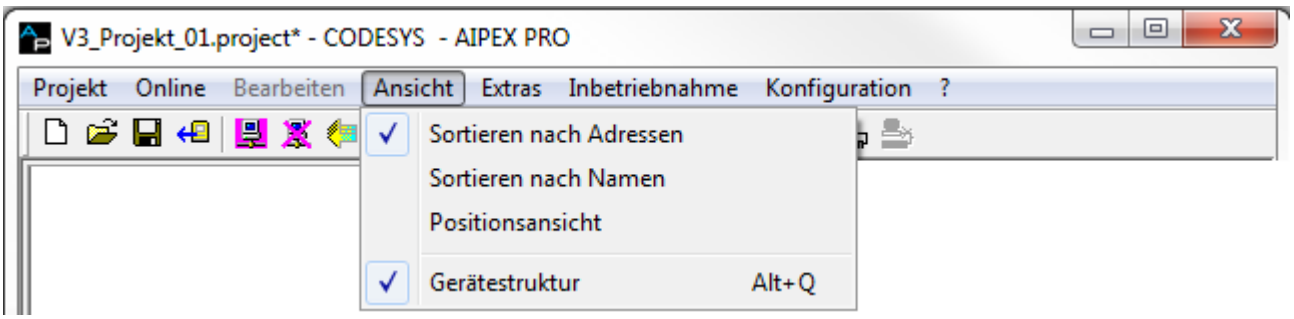
Die Funktion kann nur aufgerufen werden, wenn zuvor **'Einloggen'** erfolgreich ausgeführt wurde.

Die Daten des Projektdatensatzes (offline Geräte) werden auf physikalische vorhandene (online Geräte) übertragen.

'Parameter in das Gerät überspielen' → **auf aktuelles Gerät**: Es wird nur das im Gerätebaum selektierte Gerät berücksichtigt

'Parameter in das Gerät überspielen' → **auf alle Geräte**: Es werden alle im Gerätebaum existierenden Geräte berücksichtigt

3.6.3 Ansicht



Sortieren nach Adressen
Die Geräte im Gerätebaum werden nach aufsteigenden Busadressen sortiert.

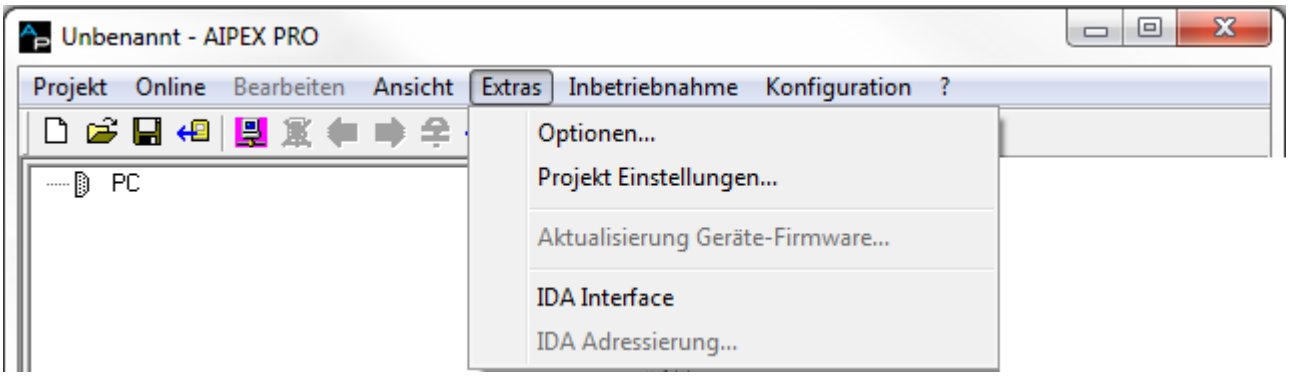
Sortieren nach Namen
Die Geräte im Gerätebaum werden alphabetisch nach ihren Stationsnamen sortiert.

Positionsansicht
EtherCAT:
Die Geräte werden in der Reihenfolge angezeigt, wie sie physikalisch im Bus vorhanden sind.
Die Geräte können im Gerätebaum verschoben werden.
Beispiel: Siehe 'Geräteposition im EtherCAT Bus verschieben' auf Seite 154.

ACC-Bus:
Unsortiert

Gerätestruktur [Alt + Q]
Die Gerätestruktur im Gerätebaum kann angezeigt oder ausgeblendet werden.

3.6.4 Extras



Optionen...
Menü für 'Basiseinstellungen', 'PC Kommunikation', 'Konfiguration' und 'Datenaktualisierung'.
Siehe 'Extras - Optionen' auf Seite 104.

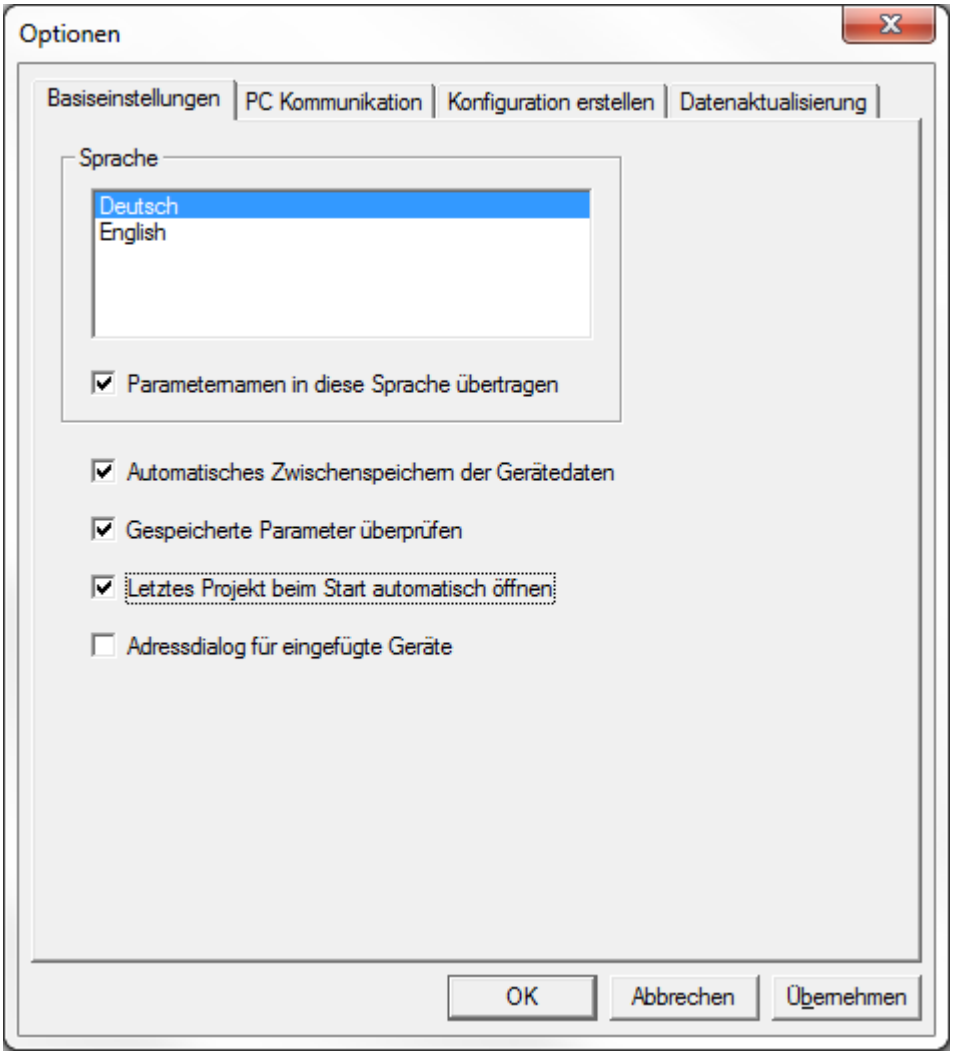
Projekt Einstellungen...
Menü für 'Basiseinstellungen' und 'Konfiguration'. Gültig für das geöffnete Projekt.
Siehe 'Extras - Projekt Einstellungen' auf Seite 112.

Aktualisierung Geräte-Firmware...
Siehe 'Aktualisierung der Geräte-Firmware' auf Seite 141.

IDA Interface (AMK Service)
IDA ist ein "Hightorque" Außenläufer Synchronmotor mit integriertem Positionsgeber und Servoregler als anschlußfertige kompakte Einheit.

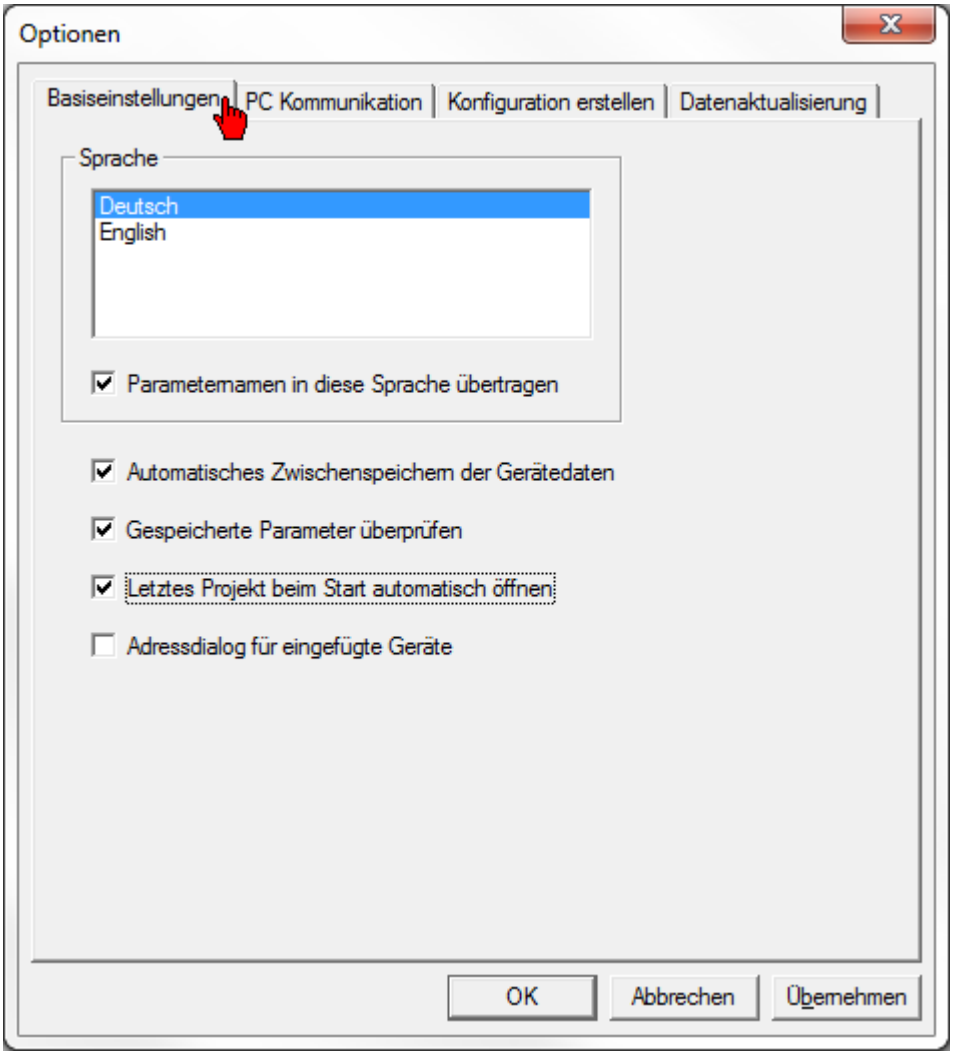
IDA Adressierung (AMK Service)
IDA ist ein "Hightorque" Außenläufer Synchronmotor mit integriertem Positionsgeber und Servoregler als anschlußfertige kompakte Einheit.

3.6.4.1 Extras - Optionen



Siehe 'Basiseinstellungen' auf Seite 105.
Siehe 'PC Kommunikation' auf Seite 106.
Siehe 'Konfiguration erstellen' auf Seite 107.
Siehe 'Datenaktualisierung' auf Seite 110.

3.6.4.1.1 Basiseinstellungen



Sprache

Wählen Sie, in welcher Landessprache die Menü- und Dialogtexte, sowie die Online Hilfe angezeigt werden soll.

Die Gerätesprache wird mit dem Parameter ID265 'Sprache' umgestellt.

Define here, in which language the menu and dialog texts should be displayed.

The device language will be change with the parameter ID265 'Language'.

Nach dem Ändern der Spracheinstellung, muss AIPEX PRO neu gestartet werden.

Restart AIPEX PRO after changing the language.

Parametername in diese Sprache übertragen

Ist diese Option aktiviert, werden die Namen und Einheiten der Parameter nicht in der im Modul eingestellten Sprache (siehe ID265 Sprache), sondern in der für AIPEX PRO gewählten Sprache dargestellt. Dies ist jedoch nur möglich, wenn die dazugehörige ADB Datei die entsprechenden Texte enthält.

Automatisches Zwischenspeichern der Gerätedaten

Ist diese Option gesetzt, werden beim ersten Zugriff auf ein Gerät alle Parameter komplett gelesen und zwischengespeichert. (Empfohlen)

Ist diese Option nicht gesetzt, werden immer nur aktuell benötigten Parameter gelesen und zwischengespeichert.

Gespeicherte Parameter überprüfen

Die Parameterwerte werden nach dem Download auf ein Gerät von AIPEX PRO rückgelesen. Anschließend wird der AIPEX PRO Projektdatensatz mit dem Online Datensatz auf Übereinstimmung überprüft.

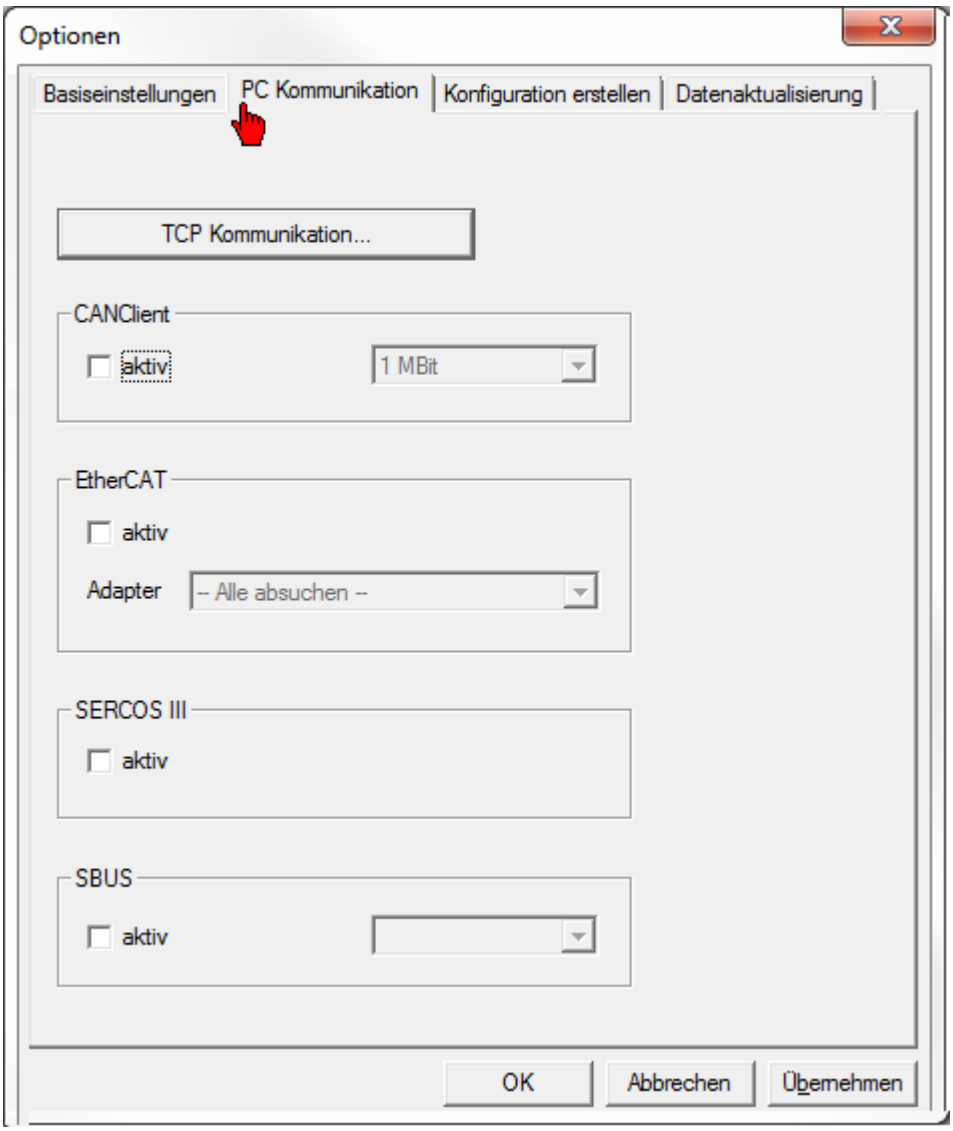
Letztes Projekt beim Start automatisch öffnen

Ist diese Option gesetzt, wird beim Start von AIPEX PRO das letzte gespeicherte Projekt automatisch geöffnet.

Adressdialog für eingefügte Geräte

Ist diese Option gesetzt, kann beim Einfügen von Geräten im Gerätekonfigurator eine freie Geräteadresse eingegeben werden, ansonsten wird die nächste freie Adresse automatisch vergeben.

3.6.4.1.2 PC Kommunikation

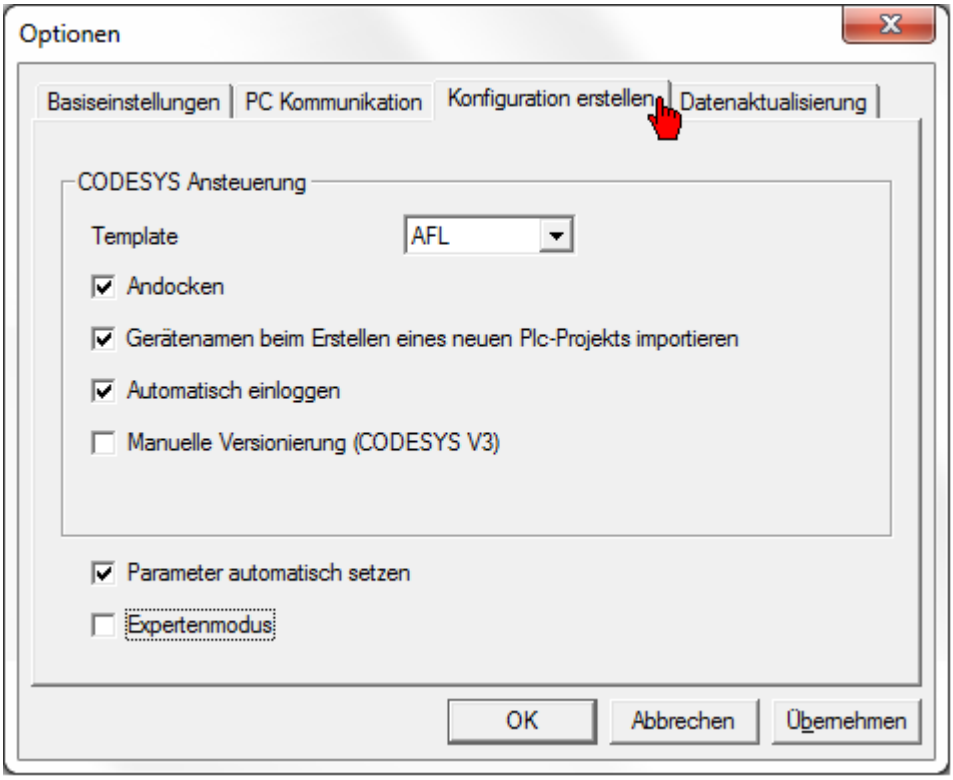


Im Fenster **'PC Kommunikation'** wird die Kommunikationsschnittstelle zwischen PC und AMK Gerät angewählt.
Abhängig vom Zielgerät stehen folgende Schnittstellen zu Verfügung:

- COM
- Ethernet (immer aktiv)
- CAN
- EtherCAT
- SERCOS III
- USB (immer aktiv)
- TwinCAT Gateway

Siehe 'Kommunikation PC - AMK Gerät' auf Seite 43.

3.6.4.1.3 Konfiguration erstellen



Template Sprache

Über das Template wird das Starttemplate bei einem neu erstellten CODESYS Projekt eingestellt.

Als Starttemplate steht 'ST' Strukturierter Text (textuelle Sprache), FUP Funktionsplan (grafische Sprache) oder die AFL Bibliotheken (sofern zusätzlich Installiert) zur Verfügung. Strukturierter Text ist die Standard Einstellung. Das 'Template' muss eingestellt sein, bevor das PLC Projekt erzeugt wird.

Andocken

Das PLC Programmiersystem CODESYS wird in die AIPEX PRO Oberfläche direkt integriert. Ist diese Option nicht gesetzt, wird der PLC Programmiersystem CODESYS als eigenständiges Softwareprogramm gestartet.

Gerätenamen beim Erstellen eines neuen PLC Projekts importieren

Geräte die physikalisch vorhanden sind werden automatisch durch den Aufruf „Neues PLC Projekt erzeugen“ in das PLC Projekt übernommen.

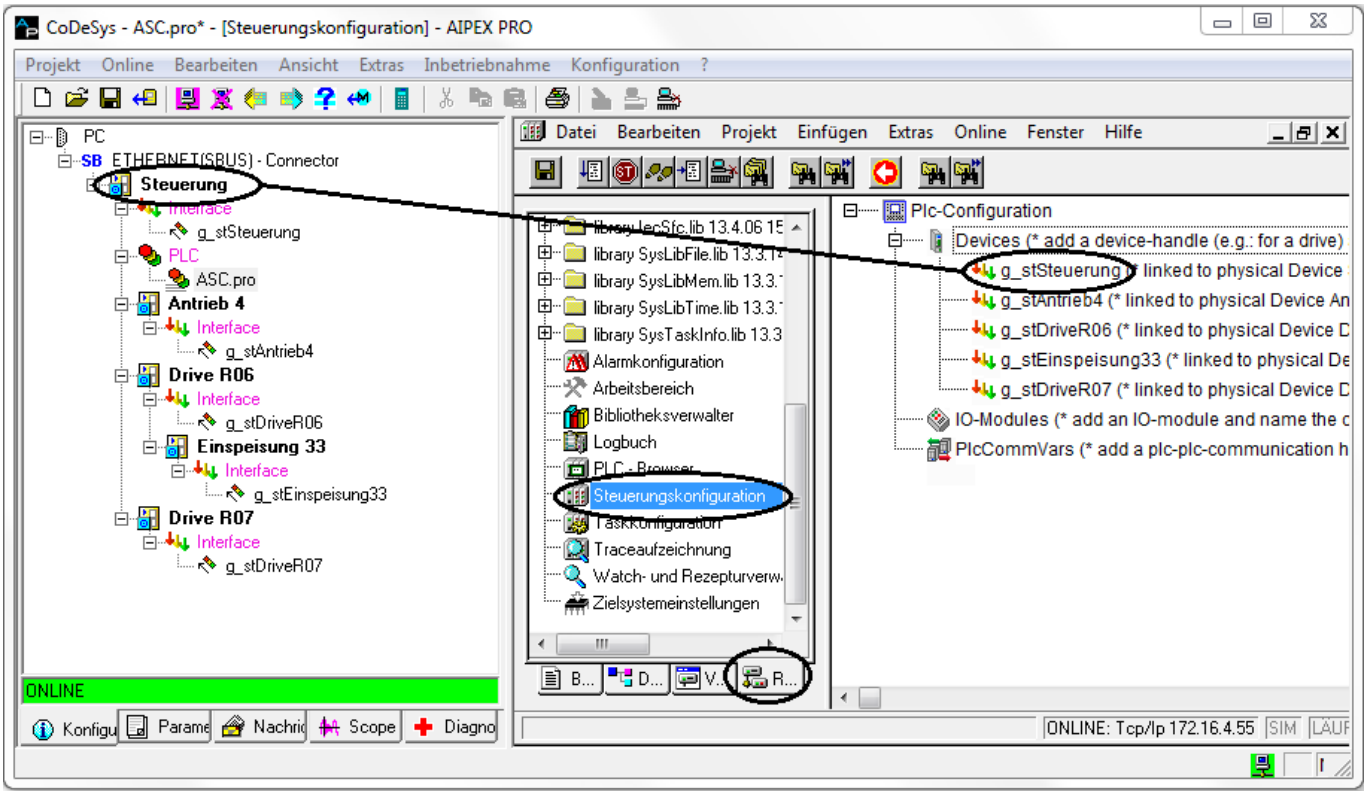
Es ist nicht zulässig die Gerätenamen nachträglich zu ändern.

Beispiel CODESYS V2

Sie finden die symbolischen Gerätenamen in CODESYS unter **Ressourcen → Steuerungskonfiguration → PLC-Configuration**.

Im AIPEX PRO Gerätebaum wird der 'PLC-Handel Gerätenamen' automatisch mit dem dazugehörigen Geräteicon 'Interface' verknüpft.

Der automatische Nachrichtenkonfigurator nutzt diese Information um das Nachrichtenkonfigurationsfile zu erstellen.

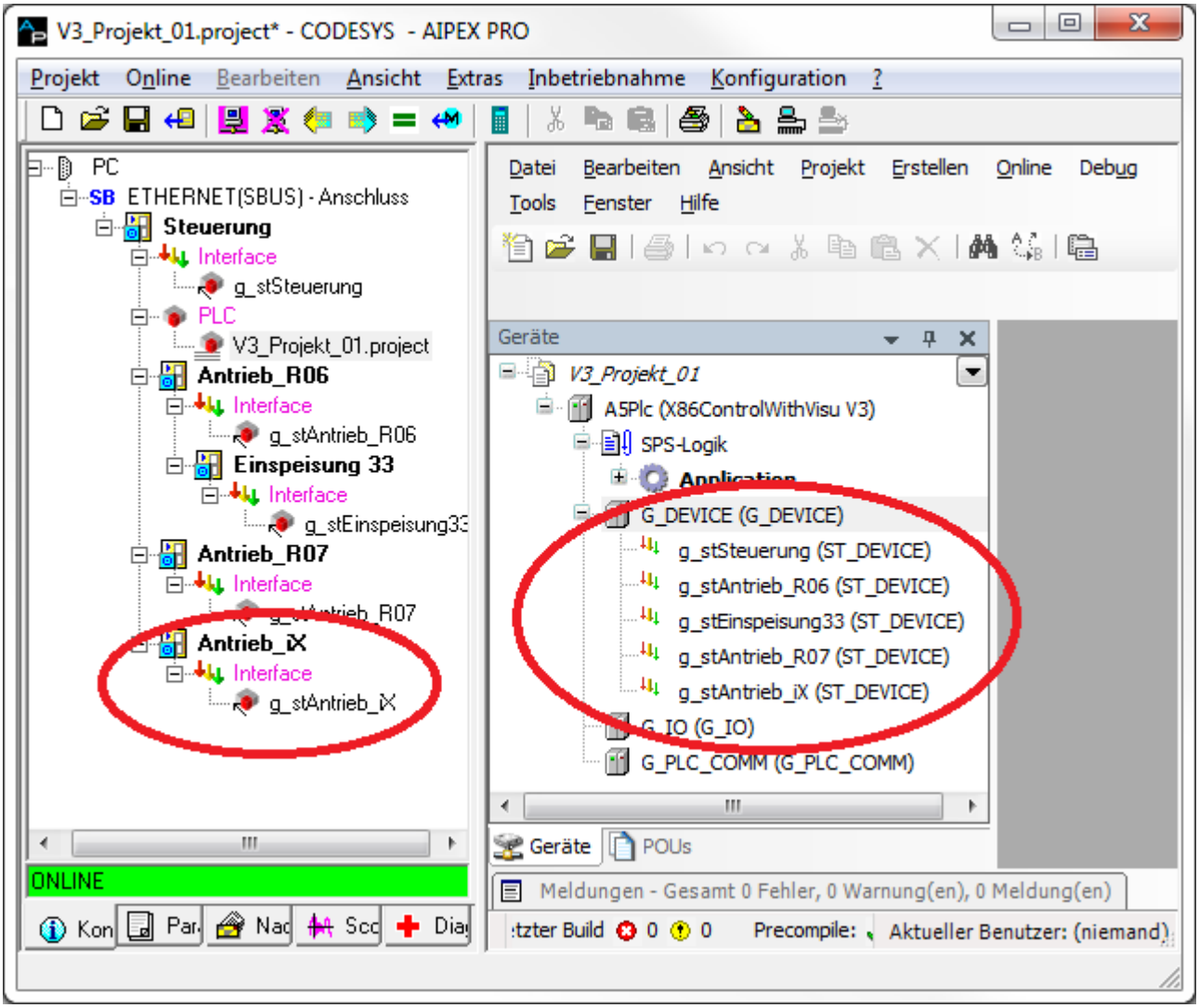


Beispiel CODESYS V3

Die symbolischen Gerätenamen finden Sie auf der CODESYS Oberfläche im Gerätebaum unter 'G_DEVICE (G_DEVICE)' wieder.

Im AIPEX PRO Gerätebaum wird der symbolischen Gerätenamen automatisch mit dem dazugehörigen Geräteicon 'Interface' verknüpft.

Der automatische Nachrichtenkonfigurator nutzt diese Information um das Nachrichtenkonfigurationsfile zu erstellen.



Automatisch einloggen

Nachdem eine Konfiguration fehlerfrei erstellt ist, wird das PLC Programm automatisch eingeloggt.

Ab AIPEX PRO Version 3.03 wird eine manuelle bzw. automatische CODESYS V3 Versionsauswahl angeboten.

Manuelle Versionierung (CODESYS V3) 'deaktiviert' (Standardeinstellung)

- Neue CODESYS V3 Projekte werden immer mit der aktuell installierten CODESYS V3 Version erstellt.
- Ab AIPEX PRO Version 3.03 erkennt die Software automatisch mit welcher CODESYS V3 Version ein bestehendes CODESYS V3 Projekt erstellt wurde und öffnet automatisch die dazu passende CODESYS V3 Version. Ist die passende CODESYS V3 Version nicht installiert, wird das Projekt mit der aktuellsten CODESYS V3 Version geöffnet.

Manuelle Versionierung (CODESYS V3) 'aktiviert'

Vor jedem öffnen von CODESYS V3 durch AIPEX PRO Version ≥ 3.03 wird nachgefragt, welche installierte Version von CODESYS V3 man verwenden möchte.

- Bestehendes CODESYS V3 Projekt öffnen → Anwender muss eine installierte CODESYS V3 Version auswählen, mit der das bestehende Projekt geöffnet werden soll.
- Neues CODESYS V3 Projekt erstellen → Anwender muss eine installierte CODESYS V3 Version auswählen, mit der das neue Projekt erstellt werden soll.

Siehe 'Versionierung CODESYS V3' auf Seite 741.

Parameter automatisch setzen

Für auserwählte AMK Funktionsbausteine werden die benötigten Geräteparameter automatisch gesetzt. Diese globale Einstellung kann individuell an die Applikation angepasst werden.

Funktionsbeschreibung und Anleitung zur individuellen Anpassung:

Siehe 'Konfiguration erstellen (projektspezifisch)' auf Seite 113.

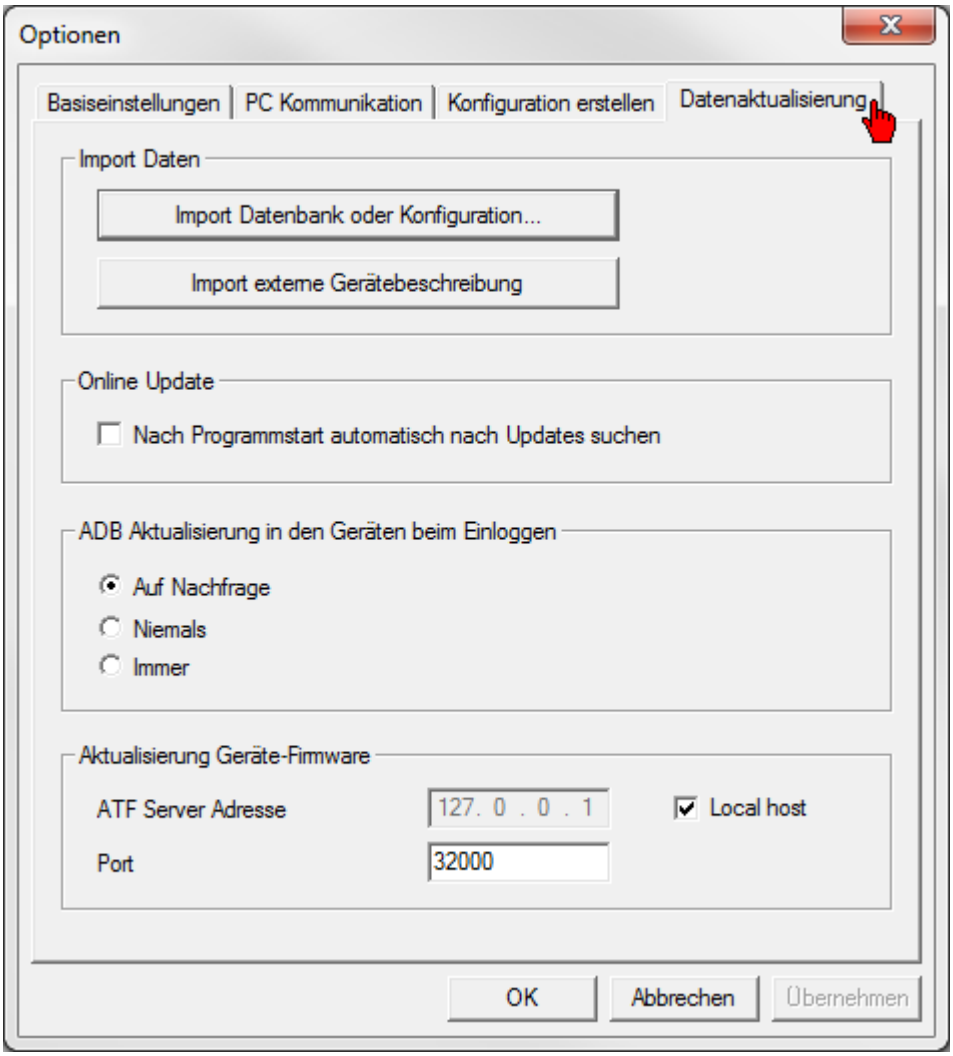
Expertenmodus

Es wird unter **Konfiguration→ CODESYS Einloggen** keine neue 'Buskonfiguration' erstellt.

Ignoriert werden Änderungen im PLC Programm die einen neue 'Buskonfiguration' zur Folge hätten.

Das **Einloggen** über CODESYS **Online** ist freigeschaltet (betrifft CODESYS V2).

3.6.4.1.4 Datenaktualisierung



Import Datenbank oder Konfiguration

Beliebige Konfigurationsdateien (z. B. ADB Dateien) und komplette Updates können über die Funktion **Import Datenbank oder Konfiguration** eingespielt werden.

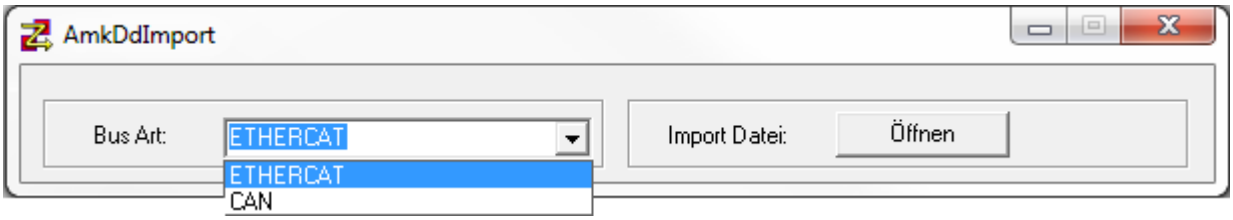
Die Ausgewählte Datei wird wenn nötig automatisch entpackt und in das richtige AIPEX PRO Verzeichnis kopiert.

Import externe Gerätebeschreibung

Die Funktionalität von externen Geräten, wie z. B. EA Klemmen ist in den Gerätebeschreibungsdateien enthalten. Bei EtherCAT sind es XML Dateien bei CAN (ACC-Bus) EDS Dateien. Mit der Funktion **'Import externe Gerätebeschreibung'** können Gerätebeschreibungsdateien importiert werden, die standardmäßig nicht in AIPEX PRO integriert sind.

Für die Funktionsfähigkeit von zusätzlich importierten Geräten kann keine Garantie übernommen werden.

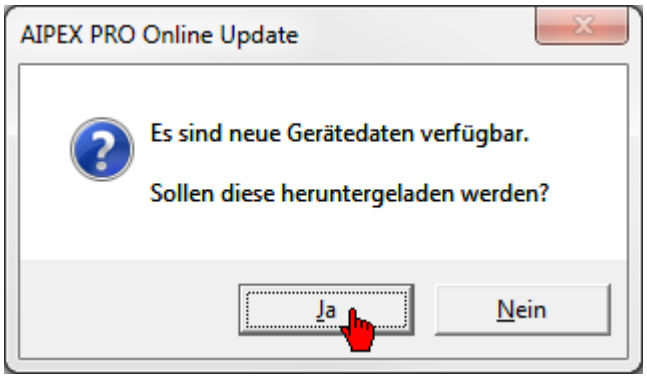
Nach dem importieren muss AIPEX PRO neu gestartet werden.



Online Update

Nach Programmstart wird automatisch nach Updates auf der AMK Webseite gesucht.

Ist ein Update vorhanden, muss der Download bestätigt werden.



Die Funktion 'Online Update' aktualisiert folgende Dateien:

- ADB: AMK Datenbank enthält Informationen über alle AMK Parameter
- MDB: Motordatenbank Datei, enthält Informationen über alle AMK Motorparameter
- Gerätedateien (z. B. für EtherCAT Geräte)
- Dokumentationen

Voraussetzungen:

- Internetzugang
- Administratorrechte

Für das 'Online Update' sind Administratorrechte notwendig. Wird AIPEX PRO ohne diese gestartet, wird nach Updates gesucht, jedoch können vorhandene Dateien nicht geladen werden.

Das 'Online Update' kann manuell über das Menü '?' gestartet werden. Siehe "'?' - Dokumentationen' auf Seite 126.

ADB Aktualisierung in den Geräten beim Einloggen

ADB (AMK Datenbank). Die ADB Datei enthält eine modulspezifische Parameterliste, samt Parametereigenschaften und Defaultwerten.

Beim Einloggen wird die ADB Datei in den im Projekt (d. h. im Gerätebaum) vorhandenen Steuerungen bei Bedarf aktualisiert. Das ADB Update ist notwendig, wenn Geräte mit neuerer Firmware als Bus-Slaves an die Steuerung angeschlossen werden. Die Steuerung kann nur dann fehlerfrei auf Slaves routen, wenn deren Firmwareversion im internen ADB bekannt ist.

Aktualisierung Geräte-Firmware

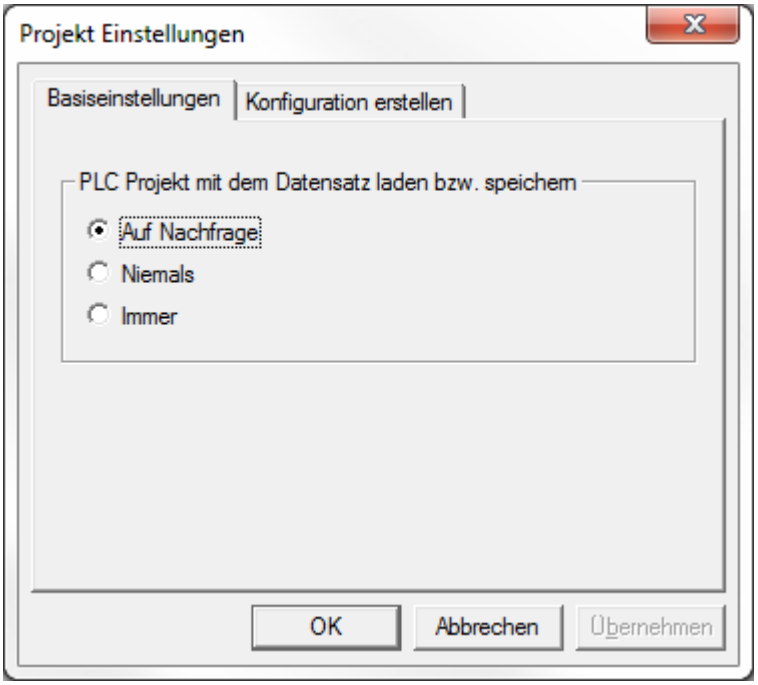
Adresse des Rechners auf dem die Software ATF (bzw. Serverteil der Software) (AMKASYN TOOL FLASHER) installiert ist.

Standardeinstellung ATF Port 32000.

Falls der Port von einer anderen Applikation belegt ist, muss der Port in AIPEX PRO und in der ATF Software geändert werden.

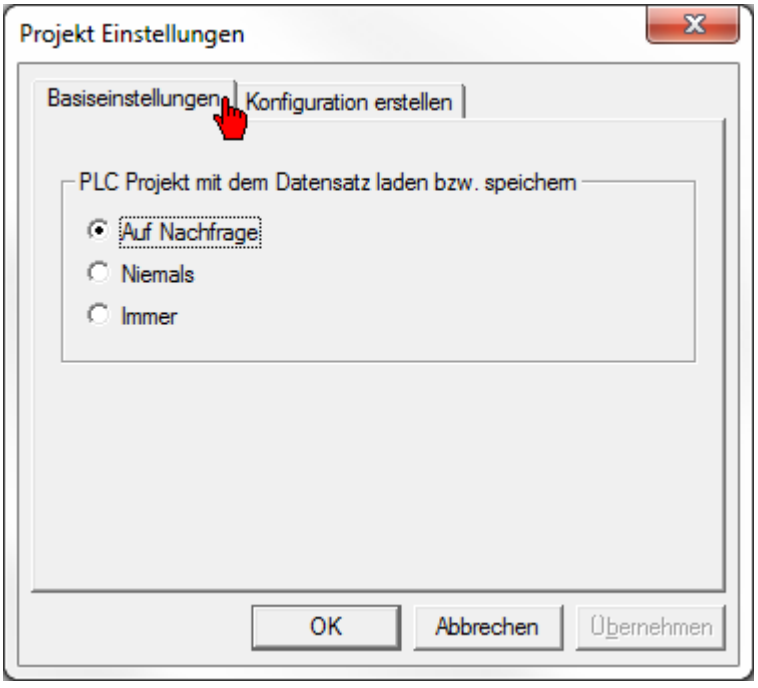
Beispiel: Siehe 'Aktualisierung der Geräte-Firmware' auf Seite 141.

3.6.4.2 Extras - Projekt Einstellungen



Siehe 'Basiseinstellungen' auf Seite 112.
Siehe 'Konfiguration erstellen (projektspezifisch)' auf Seite 113.

3.6.4.2.1 Basiseinstellungen

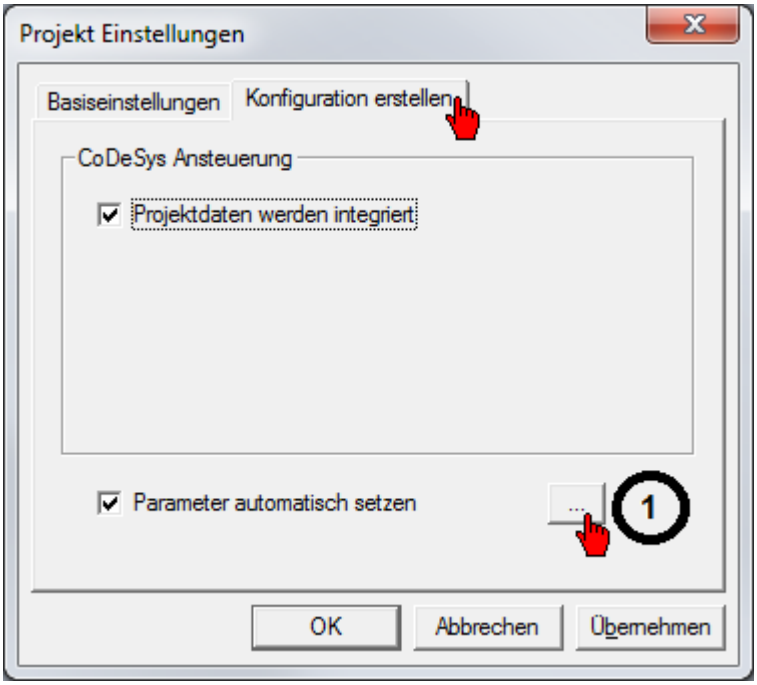


Ein vorhandenes PLC Projekt wird in der ID34159 'PLC Dateien'gespeichert.

Die Basiseinstellungen beeinflussen die Menüpunkte
'Online' → 'Parameter in das Projekt übernehmen
'Online' → 'Parameter in das Gerät überspielen'

3.6.4.2.2 Konfiguration erstellen (projektspezifisch)

Die Projekt Einstellungen können vor jedem Generieren (AIPEX PRO Funktion Menü 'Konfiguration' → 'Konfiguration erstellen ... ') geändert werden.



Projektdaten werden integriert

Das CODESYS Projekte wird in die AIPEX Projektdatei integriert.

Ist diese Option nicht gesetzt, liegen die CODESYS Projekte in einem separaten Verzeichnis neben dem AIPEX PRO Projekt.

Parameter automatisch setzen

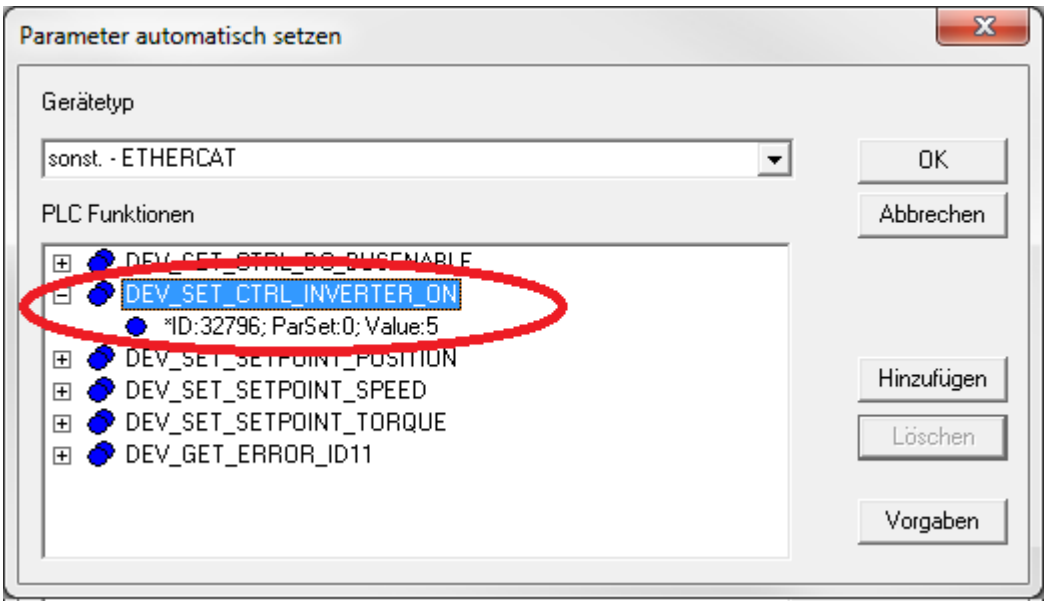
Für auserwählte AMK Funktionsbausteine werden die benötigten Geräteparameter automatisch gesetzt.

Die Funktionsbausteine für die eine automatische Parametrierung hinterlegt ist, kann dem Dialogfeld 'Parameter automatisch setzen' (siehe Taster Nr. 1) entnommen werden.

Beispiel:

Wird im PLC Programm der AMK Funktionsbaustein *SET_CTRL_INVERTER_ON__RF* aufgerufen, setzt AIPEX PRO den Wert der ID32796 'Quelle Reglerfreigabe' auf 5 (Bedeutung: Signal Reglerfreigabe wird über die PLC gesetzt).

In der Parameterbeschreibung werden die IDs und deren Bedeutung ausführlich beschrieben.



Auswahl - Gerätetyp	Bezeichnung
EC1-100 - ETHERCAT	AMKASYN Option KU/KW-EC1
IDT - 100 - CAN	AMKASMART iDT4
ihX - 100 - ETHERCAT	AMKASMART ihX
KE - 207 - CAN	AMKASYN KE Komapkteinspeisung mit ACC Bus
KE - 400 - ETHERCAT	AMKASYN KE Komapkteinspeisung mit EtherCAT
KWF - 103 - CAN	AMKASYN KWF Kompaktfrequenzumrichter mit ACC Bus
KW-R2x - 200 - ETHERCAT	AMKASYN Reglerkarte KW-R24(-R), KW-R25, KW-R26, KW-R27
sonst. - CAN	AMKASYN Reglerkarte KW-R03(P), KW-R04
sonst. - ETHERCAT	AMKASYN Reglerkarte KW-R06, KW-R07, KW-R16, KW-R17
sonst. - Lokal	Lokale IOs auf der Option KU-/KW-PLC2

Gibt es keinen Datensatz für einen konkreten Gerätetyp, wird der allgemeine 'sonst.- x' Datensatz verwendet.

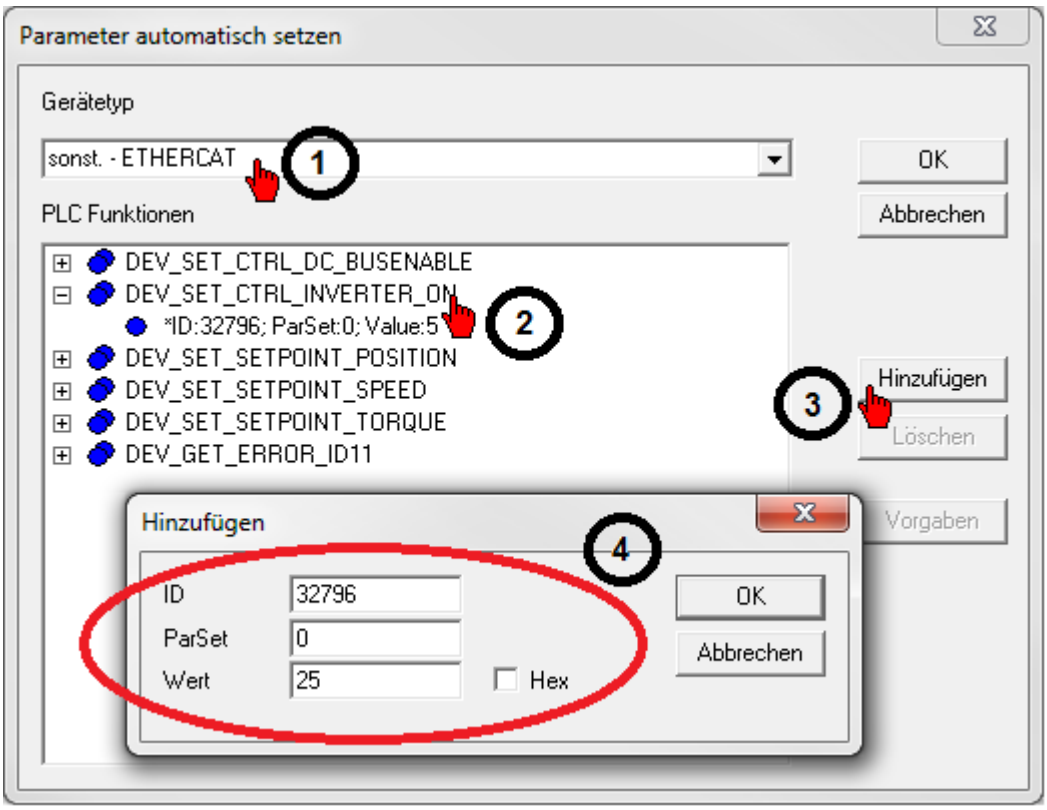
Mit der Taste **'Hinzufügen'** können Sie den PLC Funktionen eigene Geräteparametrierungen zuweisen.
Mit der Taste **'Löschen'** können Sie eigene Geräteparametrierungen löschen.
Mit der Taste **'Vorgabe'** können Sie den ursprünglichen Defaultzustand wiederherstellen.

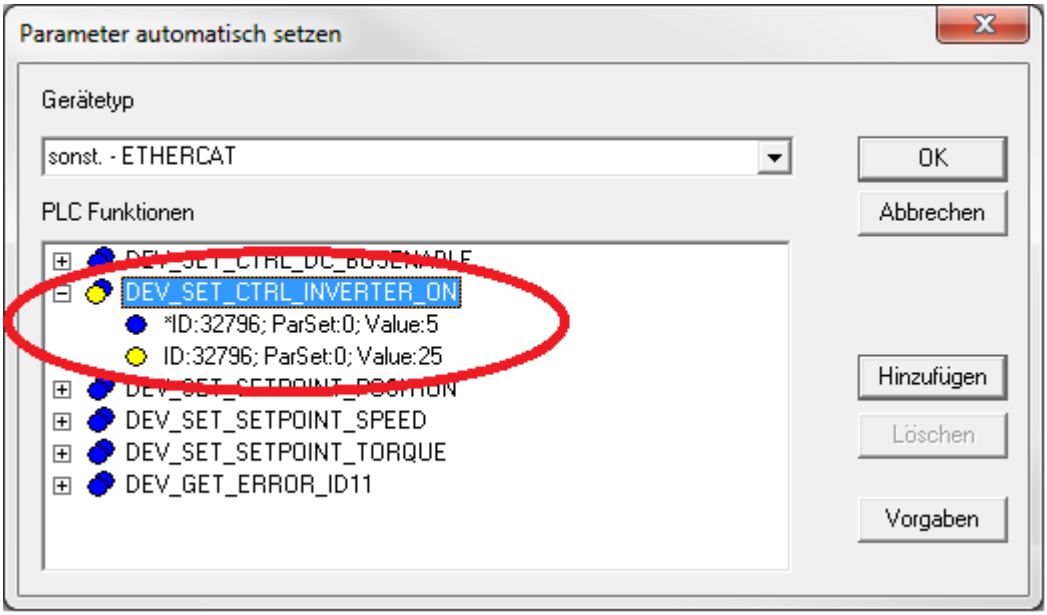
Manuell geänderte Parameterwerte werden beim Aufruf der AIPEX PRO Funktion Menü 'Konfiguration' → 'Konfiguration erstellen ... ' mit dem Standardwert überschrieben.

Lösungsmöglichkeiten:

- Automatische Parametrierung um den geforderten Wert ergänzen (siehe Beispiel)
- Automatische Parametrierung deaktivieren

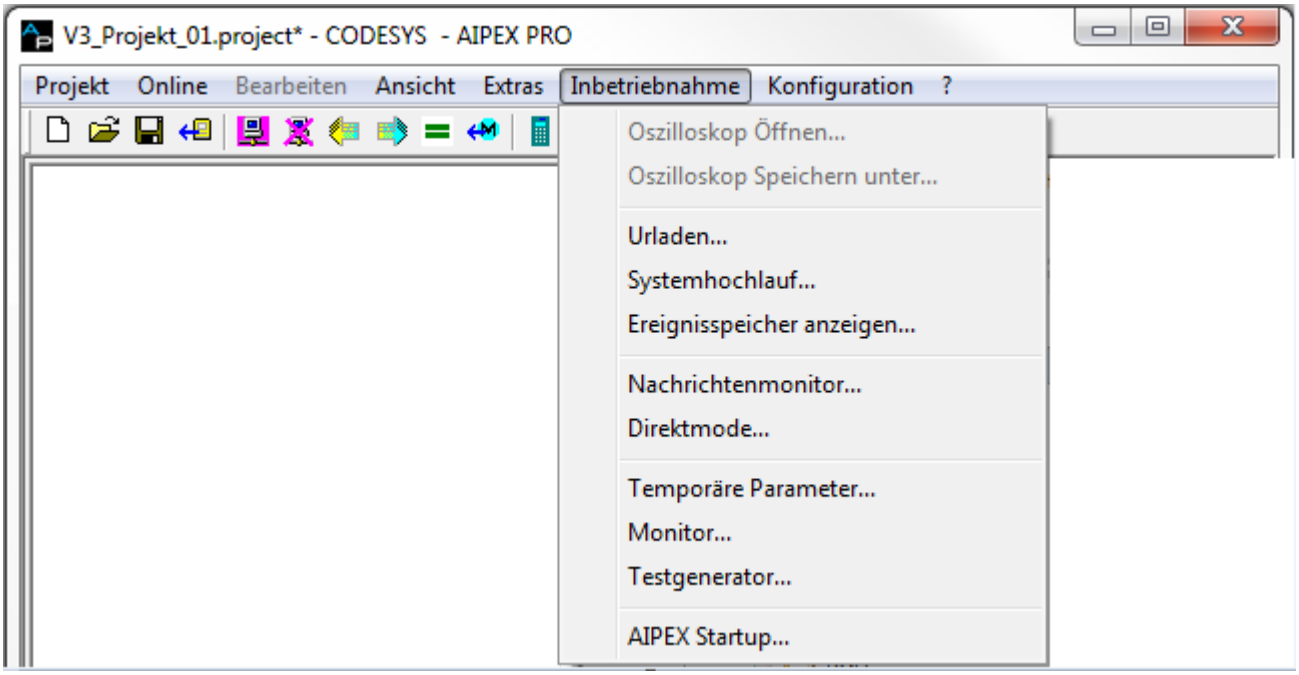
Beispiel Reglerkarte KW-R06:
Die Defaultkonfiguration ID32796 'Quelle Reglerfreigabe' wird von Code 5 'RF über EtherCAT' auf Code 25 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' geändert.





Zur Übernahme muss das AIPEX PRO Menü 'Konfiguration' → 'Konfiguration erstellen ... ' ausgeführt werden. Anschließend 24 VDC AUS/EIN.

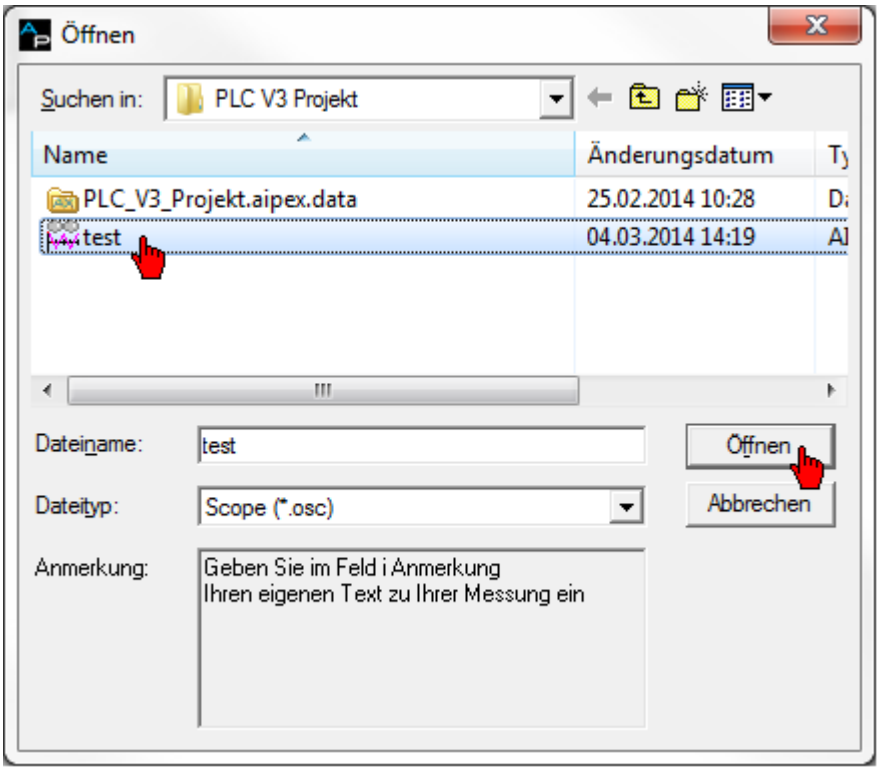
3.6.5 Inbetriebnahme



Oszilloskop Öffnen...

Der Menüpunkt **'Oszilloskop Öffnen...'** ist nur im Reiter **'Scope'** aktiv.

Eine gespeicherte Oszilloskope Datei (*.osc) kann geöffnet werden.



Oszilloskop Speichern unter...

Der Menüpunkt **'Oszilloskop Speichern unter ...'** ist nur im Reiter **'Scope'** aktiv.

Die aktuellen Einstellungen, Messwerte und die dazugehörige Beschreibung (Feld Anmerkung) werden als Oszilloskope Datei *.osc gespeichert.

Urladen...

Die Funktion 'Urladen' setzt AMK Geräte in den Defaultzustand (Auslieferungszustand) zurück.

Nach erfolgreichem 'Urladen' muss ein Systemneustart durchgeführt werden.

Voraussetzungen zum 'Urladen': Eine direkte Verbindung zwischen AIPEX PRO und AMK Gerät, Passworteingabe.

Systemhochlauf...

Auf dem angewählten Gerät wird ein 'Systemhochlauf' ausgeführt.

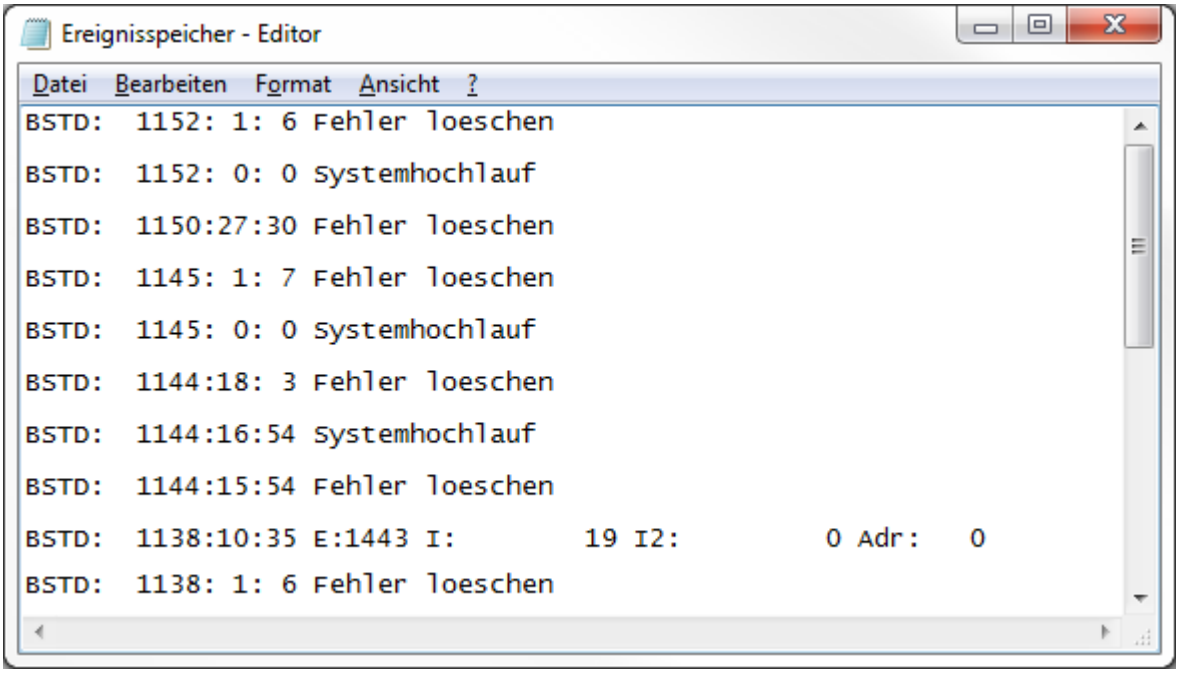
Die Funktion 'Systemhochlauf' bewirkt eine Neuberechnung der Datenhaltung. (Istwerte bleiben bestehen, Antriebsbus z. B. EtherCAT läuft weiter...)

Ereignisspeicher anzeigen...

Mit der Funktion **'Ereignisspeicher anzeigen...'** wird der Inhalt der ID34088 'Ereignisspeicher' ausgelesen und als Text im Texteditor angezeigt. Diese Funktion kann auch bei einem vorhandenen Projektdatensatz im Offlinemodus aufgerufen werden.

Der Parameter dient zur zentralen Ereignis- und Fehlerauswertung für ein AMKASYN System. Fehler im System werden über den ACC-Bus in den Parameter ID34088 'Ereignisspeicher' des Masters eingetragen und können z.B. von AIPEX PRO oder der AMK PLC ausgelesen werden.

Der Parameter liegt im Servicebereich und speichert die letzten Fehler (gemäß der Struktur ERROR_STRUCT) remanent im SEEP des Parametermoduls. Die Anzahl der gespeicherten Fehler ist abhängig von der eingesetzten Hardware.



Die Liste wird im LIFO Modus aufgebaut (Last in first out), so dass zuerst der letzte Eintrag zu sehen ist.

BSTD: Betriebsstunden

E: Fehlernummer

I: Zusatzinformation

Adr: Geräteadresse

Nachrichtenmonitor...

Der 'Nachrichtenmonitor' ist eine dynamische Anzeige von gemappten Variablen. Es erfolgt eine zyklische, aber zeitlich undefinierte Anzeige der Werte.

Bei **EtherCAT** werden nicht direkt die Werte aus dem selektierten Gerät, sondern deren Gegenstück im zugeordneten Master ausgelesen und angezeigt.

Es können nur die zum Master gemappten Variablen angezeigt werden (keine Slave – Slave Verbindungen).

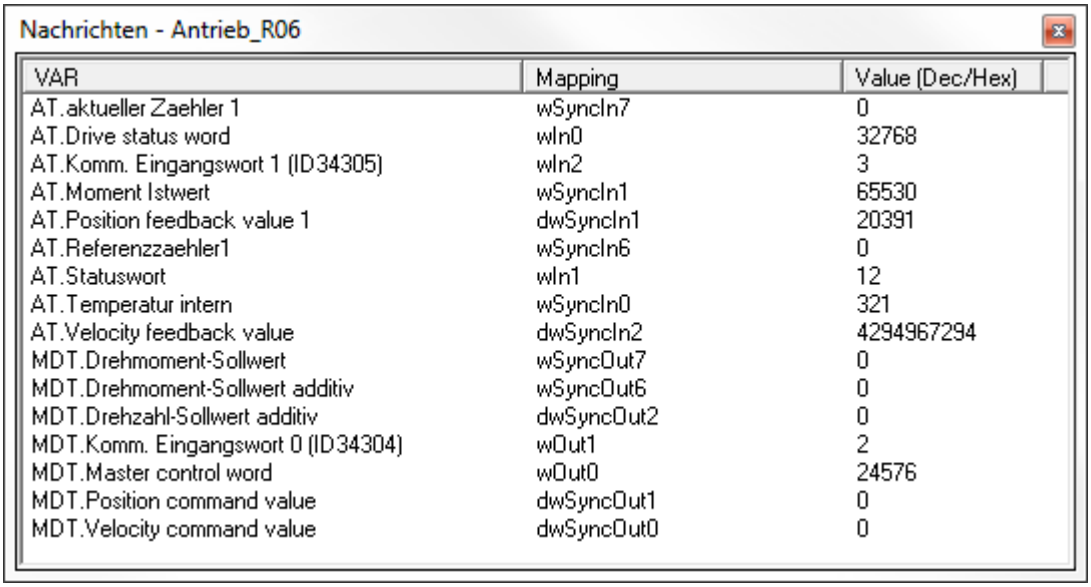
Es können nur die Werte angezeigt werden, die im Master in eine ID abgebildet werden.

Das Zielgerät muss vor Aufruf dieses Menüpunktes im Gerätebaum selektiert sein.

Bei **ACC-Bus** muss ein IXXAT CAN – Adapter in den ACC-Bus eingebunden werden. Alle im selektierten ACC-Bus konfigurierten Nachrichten werden mit Sender und Empfänger sowie dem aktuellen Wert dynamisch angezeigt.

Der Menüpunkt kann nur aufgerufen werden, wenn der CAN – Adapter nicht für die Online Verbindung aktiviert wurde.

Der ACC-Bus Master muss vor Aufruf dieses Menüpunktes im Gerätebaum selektiert sein.



VAR	Mapping	Value (Dec/Hex)
AT.aktueller Zaehler 1	wSyncIn7	0
AT.Drive status word	wIn0	32768
AT.Komm. Eingangswort 1 (ID34305)	wIn2	3
AT.Moment Istwert	wSyncIn1	65530
AT.Position feedback value 1	dwSyncIn1	20391
AT.Referenzzaehler1	wSyncIn6	0
AT.Statuswort	wIn1	12
AT.Temperatur intern	wSyncIn0	321
AT.Velocity feedback value	dwSyncIn2	4294967294
MDT.Drehmoment-Sollwert	wSyncOut7	0
MDT.Drehmoment-Sollwert additiv	wSyncOut6	0
MDT.Drehzahl-Sollwert additiv	dwSyncOut2	0
MDT.Komm. Eingangswort 0 (ID34304)	wOut1	2
MDT.Master control word	wOut0	24576
MDT.Position command value	dwSyncOut1	0
MDT.Velocity command value	dwSyncOut0	0

Direktmode...

Im **'Direktmode'** können Sie online auf AMK Geräte zugreifen.

Änderungen werden sofort in das Gerät übertragen. Wann die Änderung aktiv wird, hängt von den Parametereigenschaften ab.

Um die Änderungen anschließend in den AIPEX PRO Projektdatensatz zu übernehmen, muss das Menü **'Online'** → **'Parameter in das Projekt übernehmen'** aufgerufen werden. Siehe 'Online' auf Seite 100.

Der 'Direktmode' benötigt eine aktive Kommunikationsverbindung zwischen PC und AMK Gerät.
Siehe 'Kommunikation PC - AMK Gerät' auf Seite 43.

Siehe 'Direktmode' auf Seite 127.

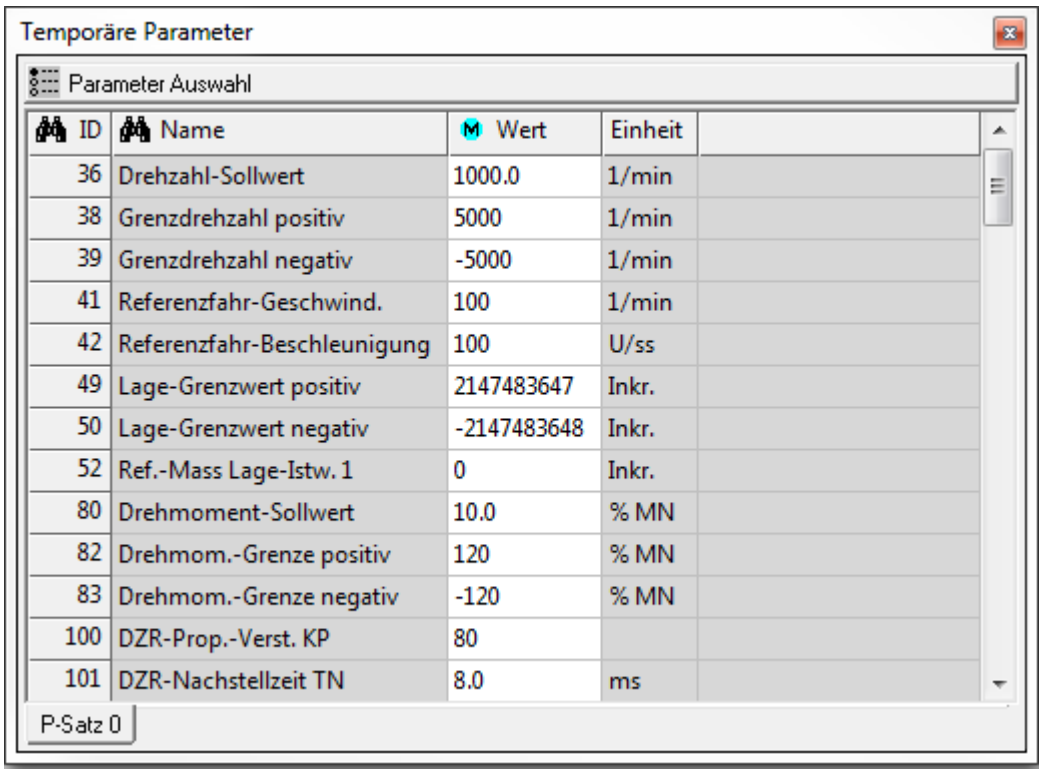
Temporäre Parameter...

Änderungen in den 'Temporären Parametern' werden sofort im Antrieb wirksam.

In der Auswahlliste werden nur Parameter angeboten, die temporär änderbar sind.

Alle geänderten Werte werden in grüner Textfarbe dargestellt.

Beim Schließen des Temporären Fensters können Sie festlegen ob die Änderungen permanent gespeichert werden sollen.



ID	Name	Wert	Einheit
36	Drehzahl-Sollwert	1000.0	1/min
38	Grenzdrehzahl positiv	5000	1/min
39	Grenzdrehzahl negativ	-5000	1/min
41	Referenzfahr-Geschwind.	100	1/min
42	Referenzfahr-Beschleunigung	100	U/ss
49	Lage-Grenzwert positiv	2147483647	Inkr.
50	Lage-Grenzwert negativ	-2147483648	Inkr.
52	Ref.-Mass Lage-Istw. 1	0	Inkr.
80	Drehmoment-Sollwert	10.0	% MN
82	Drehmom.-Grenze positiv	120	% MN
83	Drehmom.-Grenze negativ	-120	% MN
100	DZR-Prop.-Verst. KP	80	
101	DZR-Nachstellzeit TN	8.0	ms

Monitor...

Der **'Monitor'** ist eine dynamische Anzeige von Gerätedaten und -zuständen. Es erfolgt eine zyklische, aber zeitlich undefinierte Anzeige der Werte.

Siehe 'Inbetriebnahme - Monitor' auf Seite 120.

Testgenerator...

Der Testgenerator in AIPEX PRO unterstützt Sie bei der Erstinbetriebnahme und der Regleroptimierung. Die Inbetriebnahmefunktion ist ein Sollwertgenerator, der verschiedene Kurvenformen (Trapezsignal, Rechtecksignal, Dreiecksignal und Sinussignal) in den verschiedenen Betriebsarten (Momentsteuerung, Drehzahlregelung und Lageregelung) generieren kann.

Siehe 'Testgenerator' auf Seite 165.

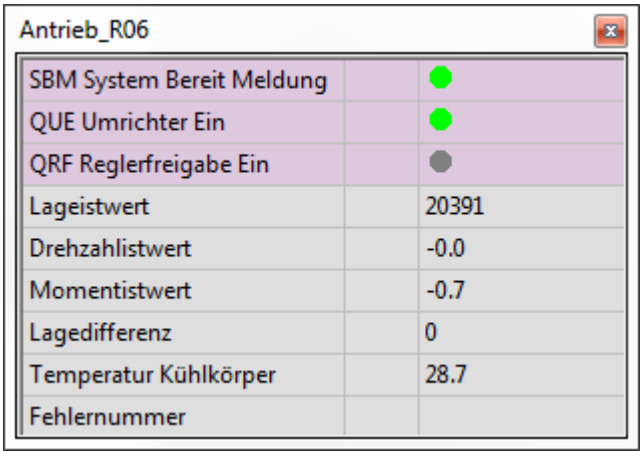
AIPEX Startup...

Die AMK Software 'AIPEX PRO Startup' unterstützt die Inbetriebnahme von AMK Antriebssystemen. Angeschlossene AMK Antriebssysteme mit elektronischem Typenschild werden automatisch erkannt und identifiziert. Dabei werden alle wichtigen Größen zum Motor, Motorgeber und Getriebe angezeigt. Betriebsbereite Parameterwerte für Stromregler, Drehzahlregler und Lageregler kann die Software in kürzester Zeit selbständig bestimmen und im Antrieb einstellen. Das gemessene Regelverhalten auf einen Sollwertsprung wird automatisch aufgezeichnet und grafisch dargestellt.
Der Anwender kann die Reglereinstellungen beeinflussen und für seine Anwendung optimieren.

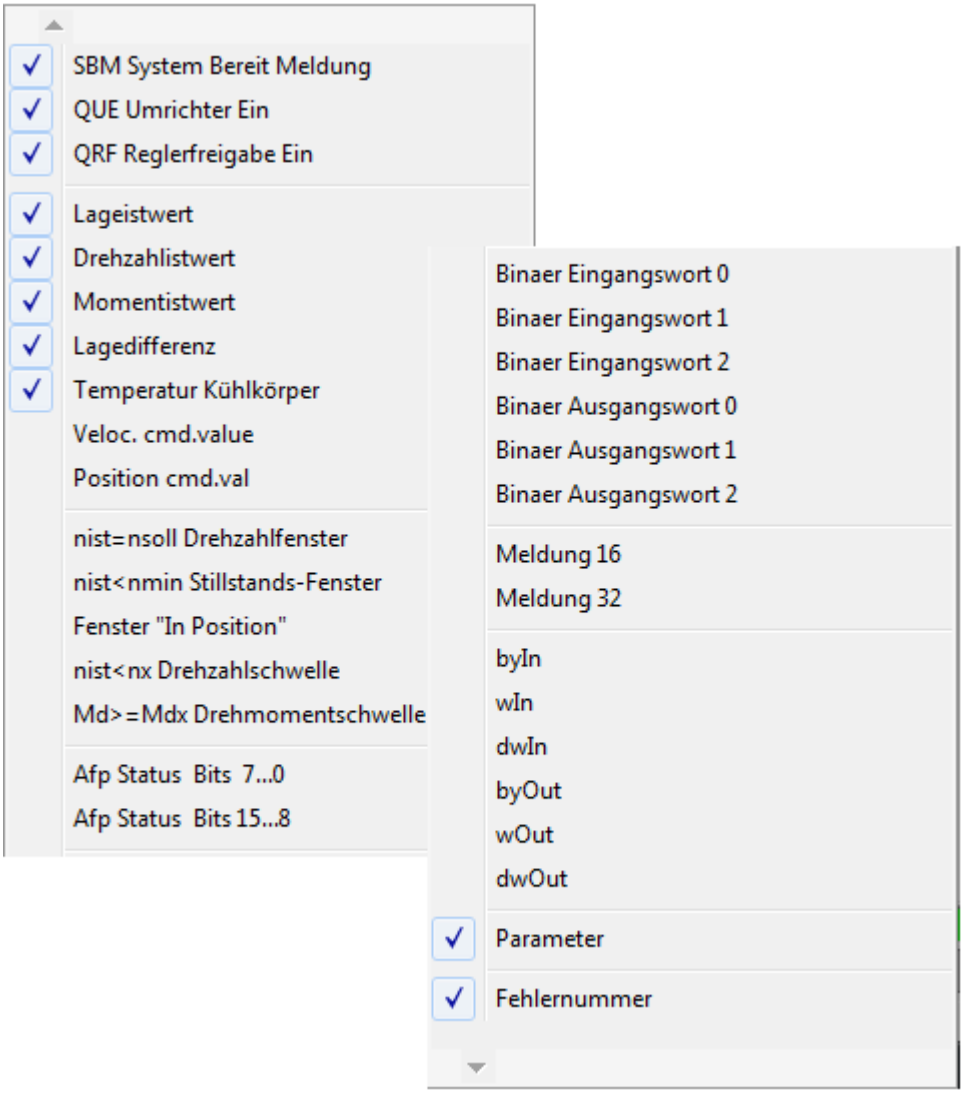
Die Software 'AIPEX Startup' ist ein eigenständiges Produkt und muss für einen Aufruf über das AIPEX PRO Menü auf dem PC installiert sein.
Siehe Dokument Softwarebeschreibung AIPEX PRO Inbetriebnahme, (Teile-Nr. 205171).

3.6.5.1 Inbetriebnahme - Monitor

Der **Online Monitor** ist eine dynamische Anzeige von Gerätedaten und -zuständen. Es erfolgt eine zyklische, aber zeitlich undefinierte Anzeige der Werte.



Mit Klicken der rechten Maustaste auf den **Online Monitor** öffnet sich ein Kontext-Menü, über das weitere Anzeigewerte in den **Online Monitor** übernommen werden können.



Mit dem Feld **'Parameter'** können Sie einen beliebigen Parameter dynamisch anzeigen lassen.
Die Parameternummer können Sie direkt in den Online Monitor eingeben.

Dynamische Anzeigewerte speziell für ACC-Bus

Der Inhalt der dynamischen Anzeige der 'Meldung 16' ist über die ID32785 'Meldung 16' änderbar.

Der Inhalt der dynamischen Anzeige der 'Meldung 32' ist über die ID32786 'Meldung 32' änderbar.

Beispiel: ID32786 = 40 (Drehzahlistwert); Der Drehzahlistwert wird dynamisch durch anwählen der Meldung 32 angezeigt.

ByIn/Out, wIn/Out, dwIn/Out sind PLC Variablen. Geben Sie in der zweiten Spalte den Index Ihrer PLC Variable an, die dargestellt werden soll.

Beispiel: In der PLC (Steuerungskonfiguration) ist das Word wOut4 konfiguriert. Geben Sie als Index die 4 an.

3.6.5.2 Testgenerator

⚠GEFAHR	
	Bewegungen der Motorwelle! Gefährliche Bewegungen entstehen durch unkontrollierte oder ungewollte Bewegungen der Motorwelle. **Gegenmaßnahmen:** • Entkoppeln Sie den Motor von der Last damit die Motorwelle frei drehen kann. • Legen Sie die maximal zulässige Prozessdrehzahl fest und parametrieren Sie ID113 entsprechend. (ID113 = maximale Prozessdrehzahl/1,25)

Der Testgenerator in AIPEX PRO unterstützt Sie bei der Erstinbetriebnahme und der Regleroptimierung. Die Inbetriebnahmefunktion ist ein Sollwertgenerator, der verschiedene Kurvenformen (Trapezsignal, Rechtecksignal, Dreiecksignal und Sinussignal) in den verschiedenen Betriebsarten (Momentsteuerung, Drehzahlregelung und Lageregelung) generieren kann.

Überprüfen Sie bei der Erstinbetriebnahme die Drehrichtung des Motors.

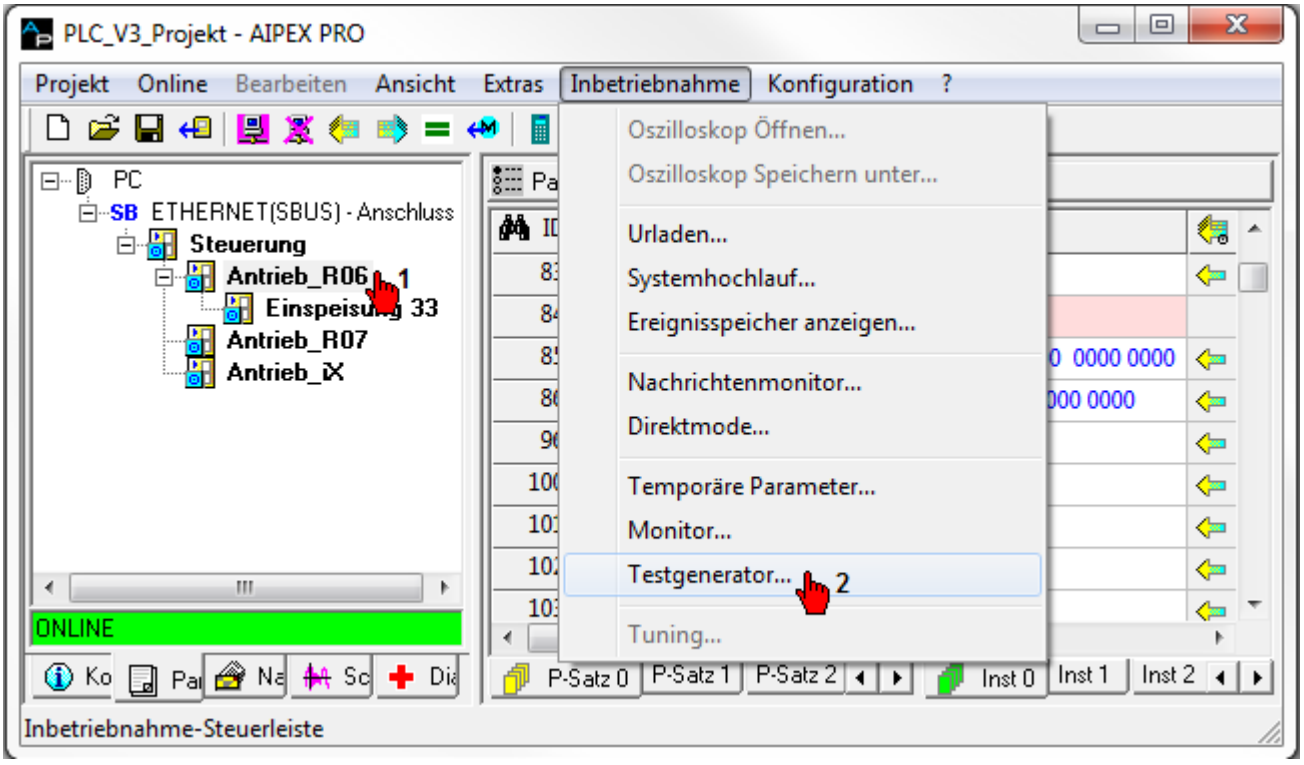
Das Beispiel zeigt, wie Sie einen positiven Drehzahlsollwert mit Beschleunigungs- und Bremsrampe vorgeben können. Mit der AIPEX PRO 'Monitor- und Oszilloskopfunktion' können Sie die Istwerte des Motors anzeigen und aufzeichnen.

Die Drehrichtung des Motors kann in ID32773 Bit 16 = 1 negiert werden.

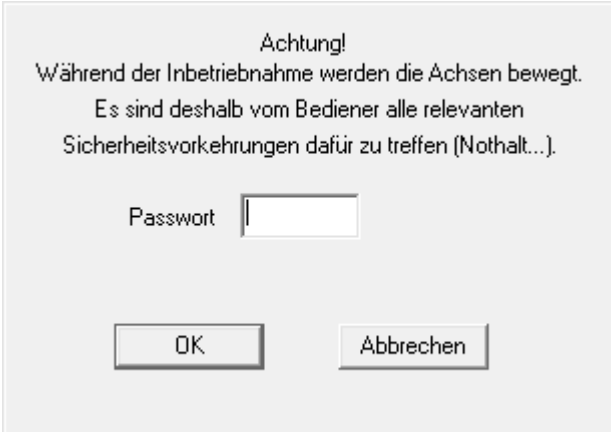
Starten des Testgenerators

Markieren Sie im Gerätebaum den Antrieb, für den Sie mit dem Testgenerator einen Sollwert vorgeben möchten.

Klicken Sie auf das Menü 'Inbetriebnahme' und anschließend auf die Funktion 'Testgenerator...'

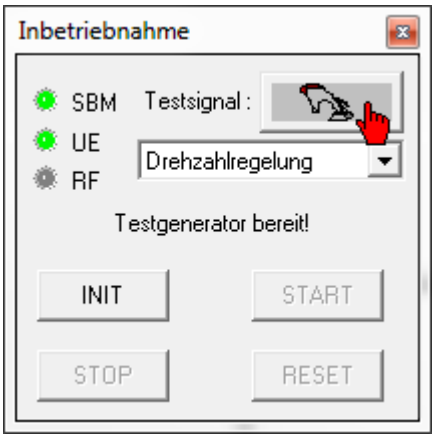


Der Testgenerator wird mit dem AMK Servicepasswort (500591) freigegeben.



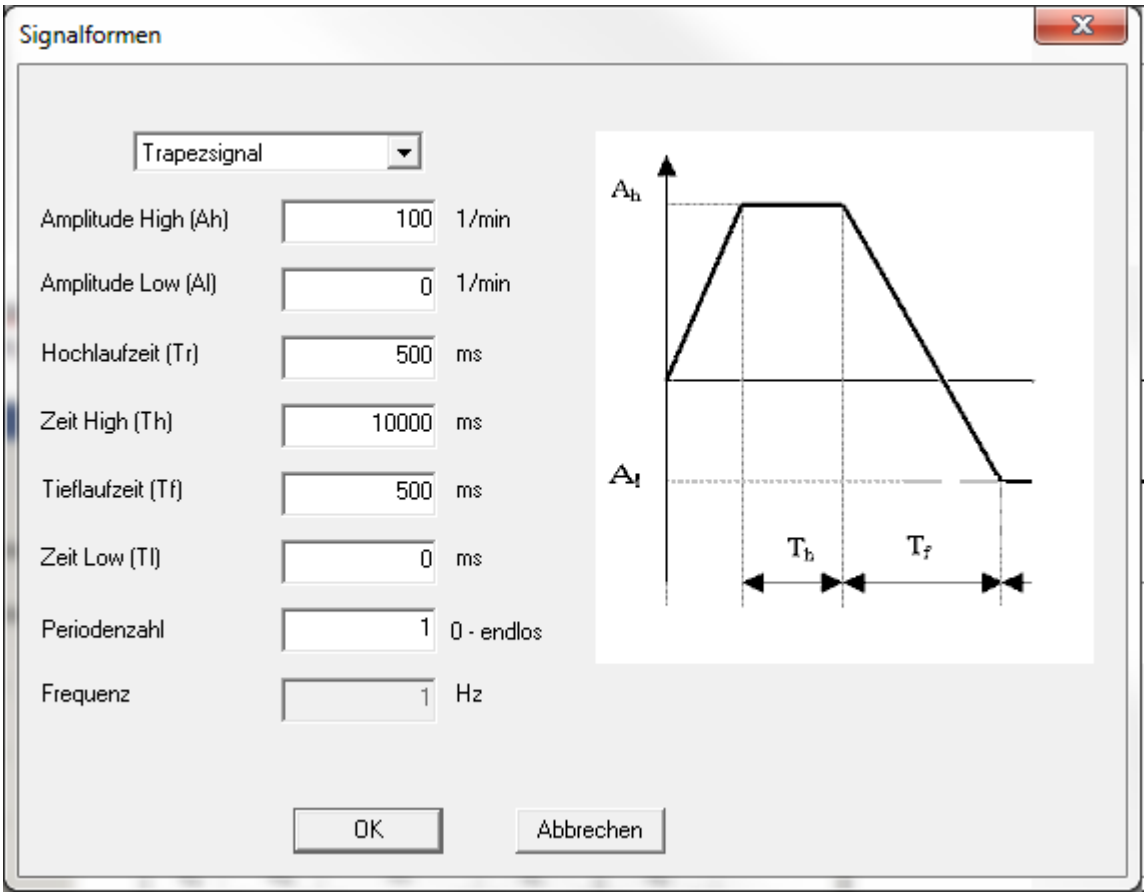
Signal Konfigurationsfenster 'Signalformen' öffnen.

Die Reglerfreigabe muss zur Konfiguration deaktiviert sein.

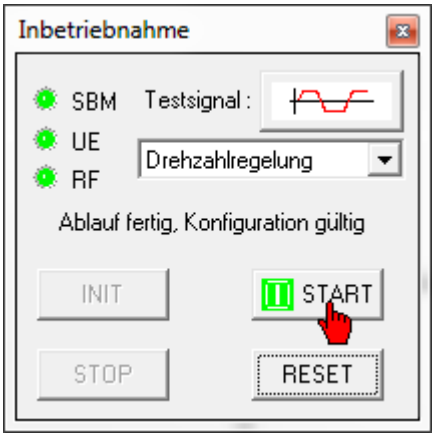


Signalkonfiguration am Beispiel 'Trapezsignal'.

Mit dem 'Trapezsignal' können Sie einen positiven und negativen Drehzahlsollwert mit Beschleunigungs- und Bremsrampe vorgeben. Im Beispiel ist eine positive Bewegung konfiguriert.



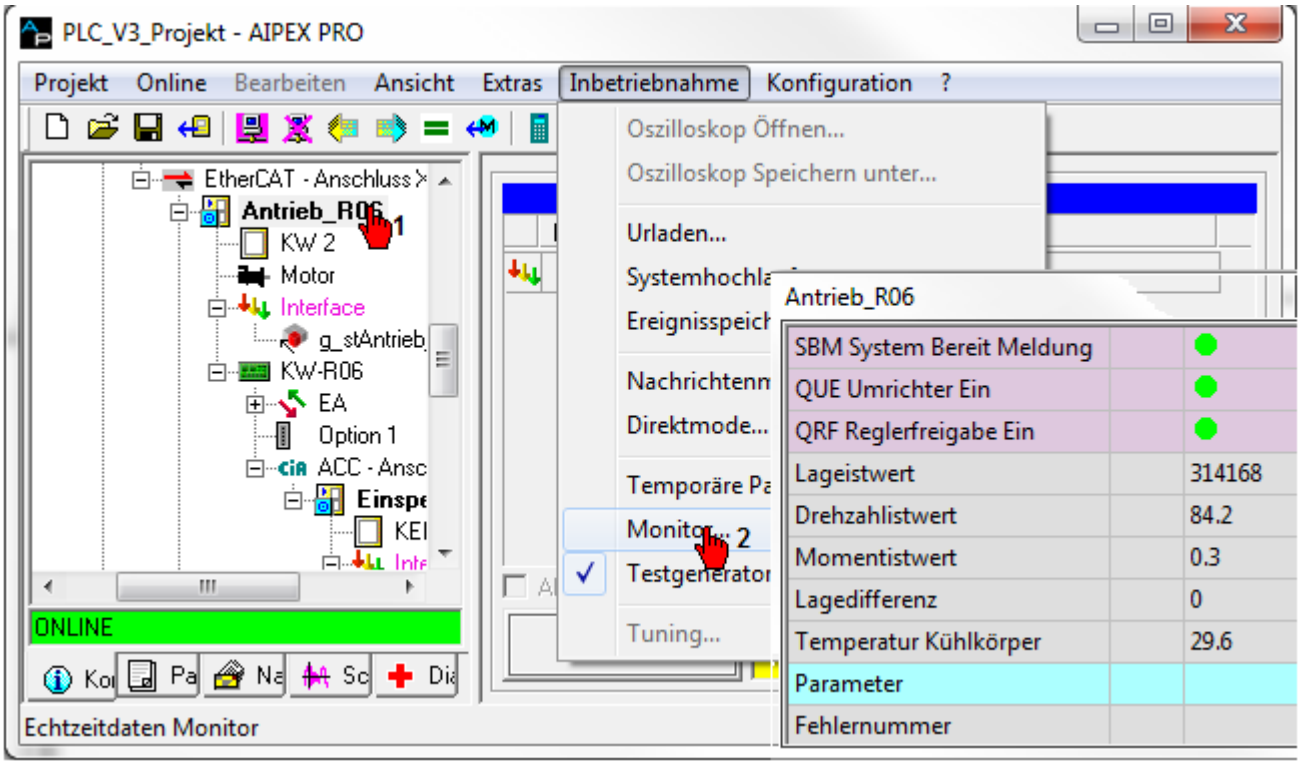
Aktivieren Sie jetzt die Motorregelung.
Die LED's zeigen den Status von SBM, UE und RF des Antriebsystems an. Grün bedeutet aktiv.
Der konfigurierte Bewegungsablauf wird mit der Taste **'START'** gestartet.



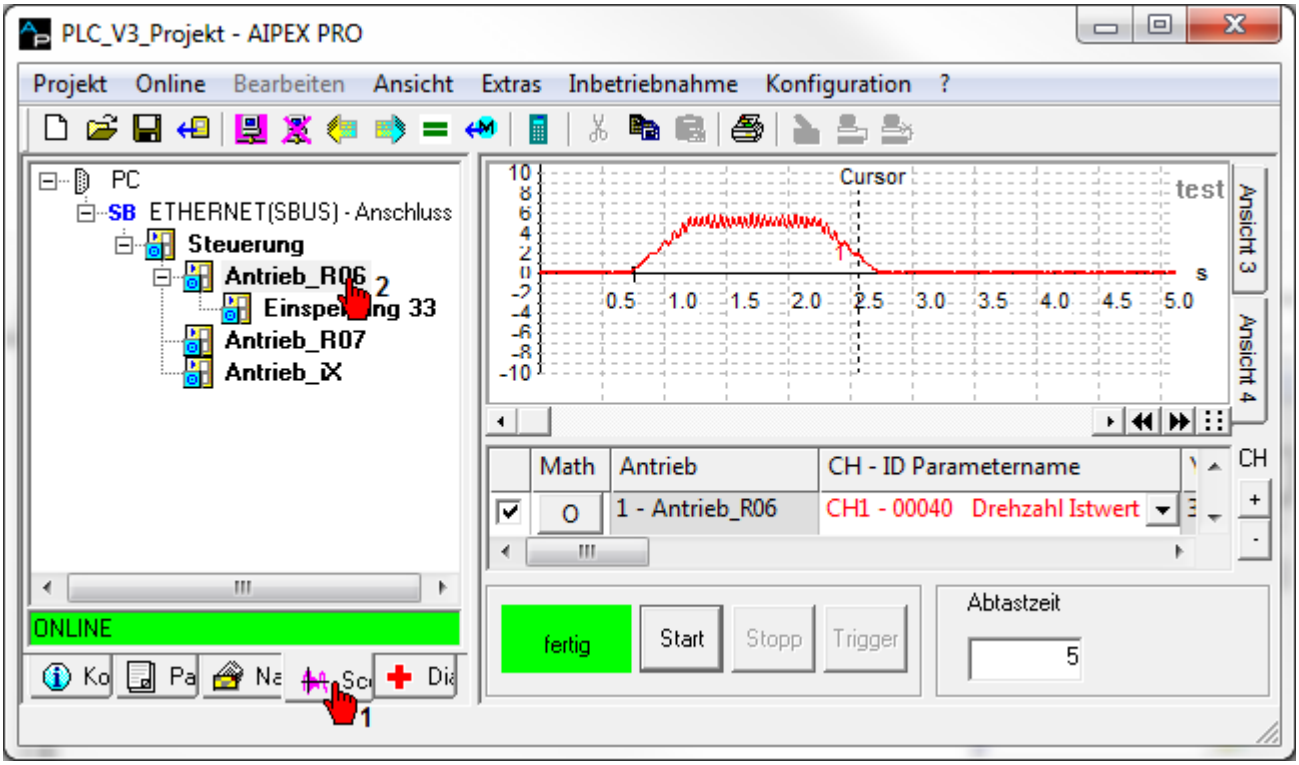
Die Bewegung kann mit den Tasten **'STOP'** angehalten und mit **'START'** weitergeführt werden.
Ein Klick auf den Taster **'STOP'** und anschließend auf den Taster **'RESET'** bricht den kompletten Bewegungsablauf ab.

Istwertanzeigen

Mit der AIPEX PRO 'Monitorfunktion' können Sie Status und Istwerte anzeigen.
Prüfen Sie, ob der Lageistwert ansteigt und die Drehrichtung positiv ist, wenn Sie positive Sollwerte vorgeben.
Eine positive Drehrichtung bedeutet, dass sich die Motorwelle im Uhrzeigersinn dreht, wenn Sie auf die Motorwelle blicken.

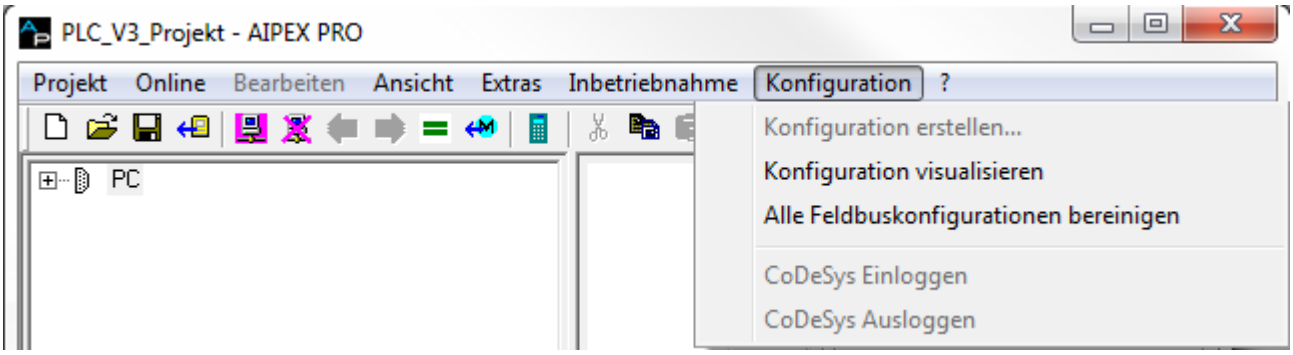


Mit dem AIPEX PRO Oszilloskop kann die Bewegung aufgezeichnet und ausgewertet werden.



Konfigurationshilfe Oszilloskop:
Siehe 'Scope - Anleitung' auf Seite 92.

3.6.6 Konfiguration



Konfiguration erstellen...

Icon

Konfiguration erstellen übersetzt das PLC Programm und erzeugt automatisch die Nachrichtenkonfiguration zwischen den Geräten. Zusätzlich werden Geräteparameter automatisch an verwendete Funktionsblöcke im CODESYS Programmiersystem angepasst. (Siehe 'Konfiguration erstellen (projektspezifisch)' auf Seite 113.)

Die Verknüpfung zwischen den symbolischen Gerätenamen (Variablen) aus dem CODESYS Programmiersystem und den physikalischen vorhanden Geräten müssen manuell per 'Drag and Drop' (Siehe 'Konfiguration erstellen' auf Seite 107.) oder automatisch durch die AIPEX PRO Funktion 'Gerätenamen beim Erstellen eines neuen PLC Projekts importieren' erstellt werden Siehe 'Automatische Buskonfiguration' auf Seite 743.).

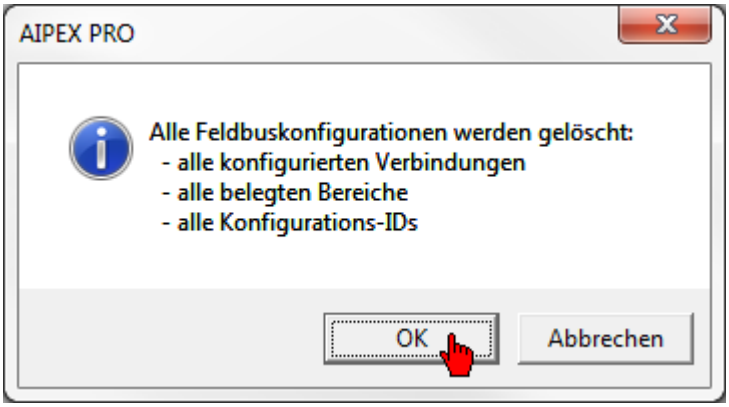
Konfiguration visualisieren

Die Funktion **'Konfiguration visualisieren'** zeigt die Netzwerkkonfiguration mit den PLC Variablen an.

Alle Feldbuskonfigurationen bereinigen

Die Funktion **'Alle Feldbuskonfigurationen bereinigen'** löscht sämtliche Verbindungen und Speicherplatzbelegungen aus allen Buskonfigurationen sowie die Konfigurations-IDs (34036 CCB-File, 1204 XML-File...) aller Geräte im Projekt.

Starten Sie anschließend die Geräte neu.



CODESYS Einloggen

Icon

'CODESYS Einloggen' verbindet CODESYS mit der PLC Steuerung und wechselt in den Onlinemodus.

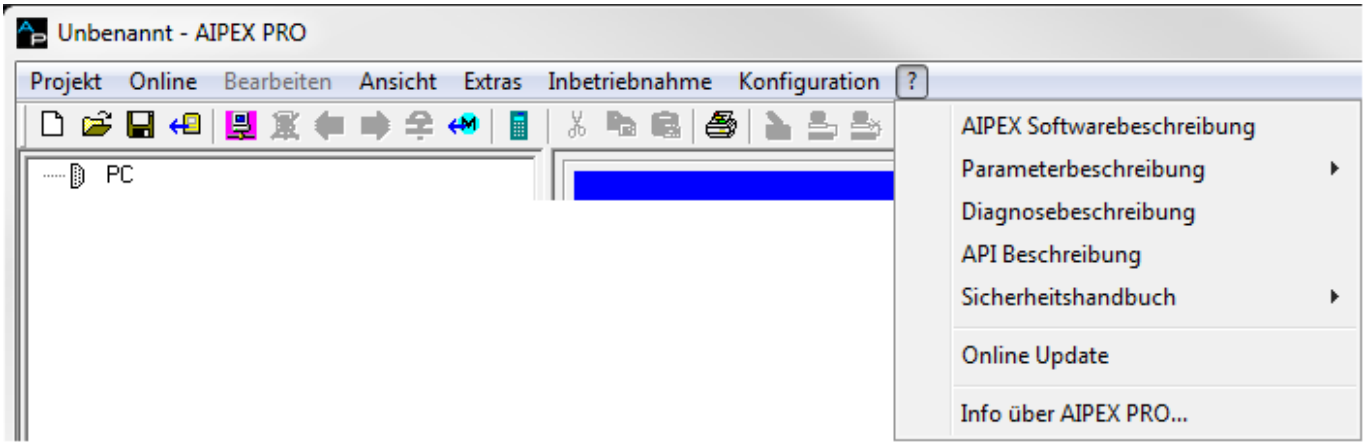
Bei veralteter Konfiguration wird automatisch das Menü **'Konfiguration erstellen...'** aufgerufen.

CODESYS Ausloggen

Icon

Mit **'CODESYS Ausloggen'** wird die Verbindung zur Steuerung getrennt und in den Offlinemodus gewechselt.

3.6.7 '?' - Dokumentationen



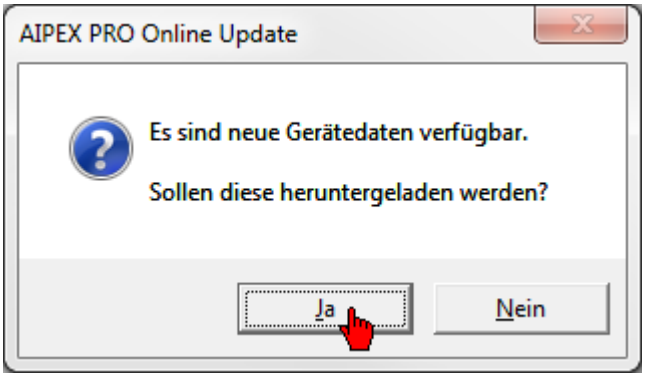
Dokumentationen werden mit einer Installation von 'AIPEX PRO Service Packs' oder einem 'Online Update' automatisch aktualisiert.

(In Service Packs sind die Daten mehrere 'Online Updates' enthalten).

Online Update

Ist ein Update vorhanden, muss der Download bestätigt werden.

Nach dem Download muss AIPEX PRO neu gestartet werden.



Die Funktion 'Online Update' aktualisiert folgende Dateien:

- ADB: AMK Datenbank enthält Informationen über alle AMK Parameter
- MDB: Motordatenbank Datei, enthält Informationen über alle AMK Motorparameter
- Gerätedateien (z. B. für EtherCAT Geräte)
- Dokumentationen

Voraussetzungen:

- Internetzugang
- Administratorrechte

Für das 'Online Update' sind Administratorrechte notwendig. Wird AIPEX PRO ohne Administratorrechte gestartet, wird nach Updates gesucht, jedoch können vorhandene Dateien nicht installiert werden.

Das 'Online Update' kann bei Programmstart automatisch gestartet werden. Siehe 'Datenaktualisierung' auf Seite 110.

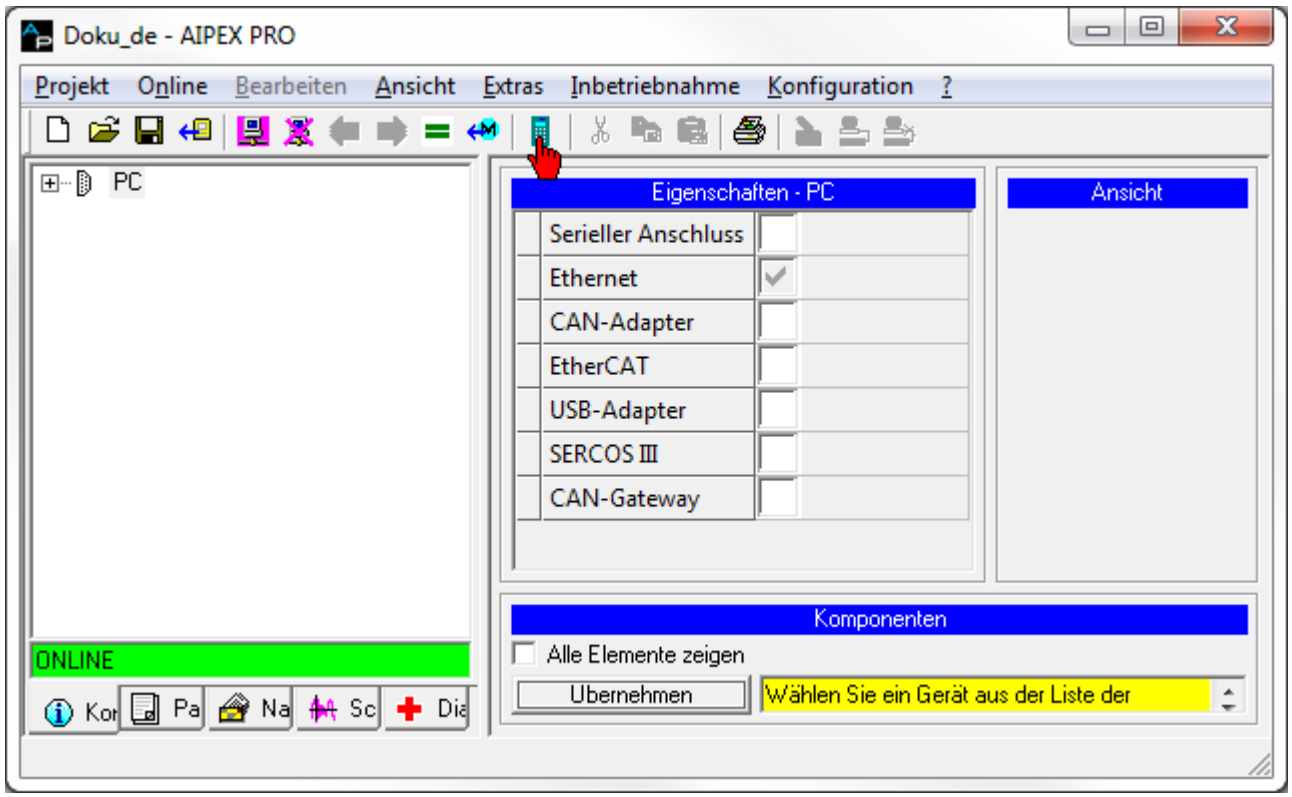
3.6.8 Direktmode

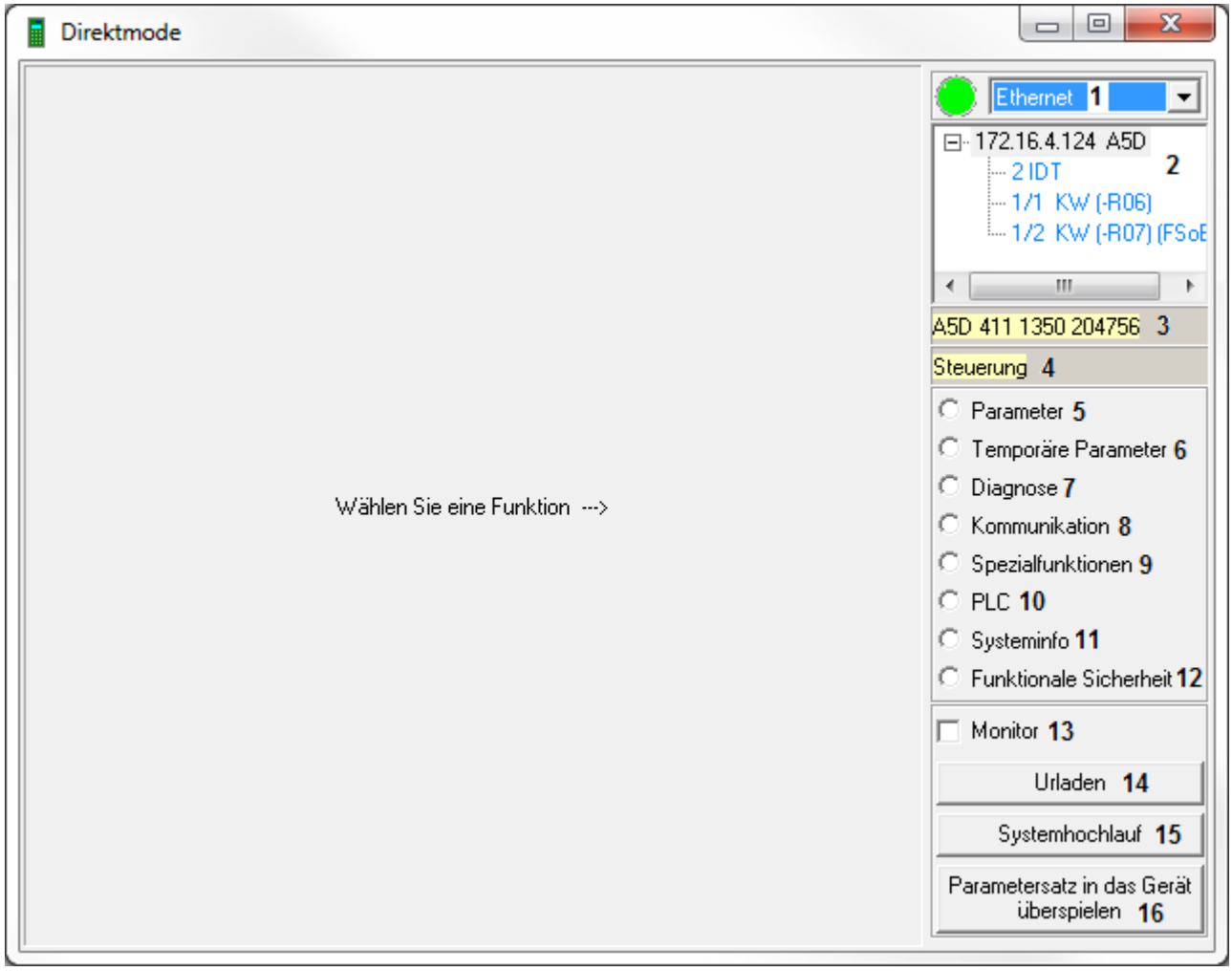
Im **'Direktmode'** können Sie online auf AMK Geräte zugreifen.

Änderungen werden sofort in das Gerät übertragen. Wann die Änderung aktiv wird, hängt von den Parametereigenschaften ab.

Um die Änderungen anschließend in den AIPEX PRO Projektdatensatz zu übernehmen, muss das Menü **'Online'** → **'Parameter in das Projekt übernehmen'** aufgerufen werden. Siehe 'Online' auf Seite 100.

Der 'Direktmode' benötigt eine aktive Kommunikationsverbindung zwischen PC und AMK Gerät.
Siehe 'Kommunikation PC - AMK Gerät' auf Seite 43.

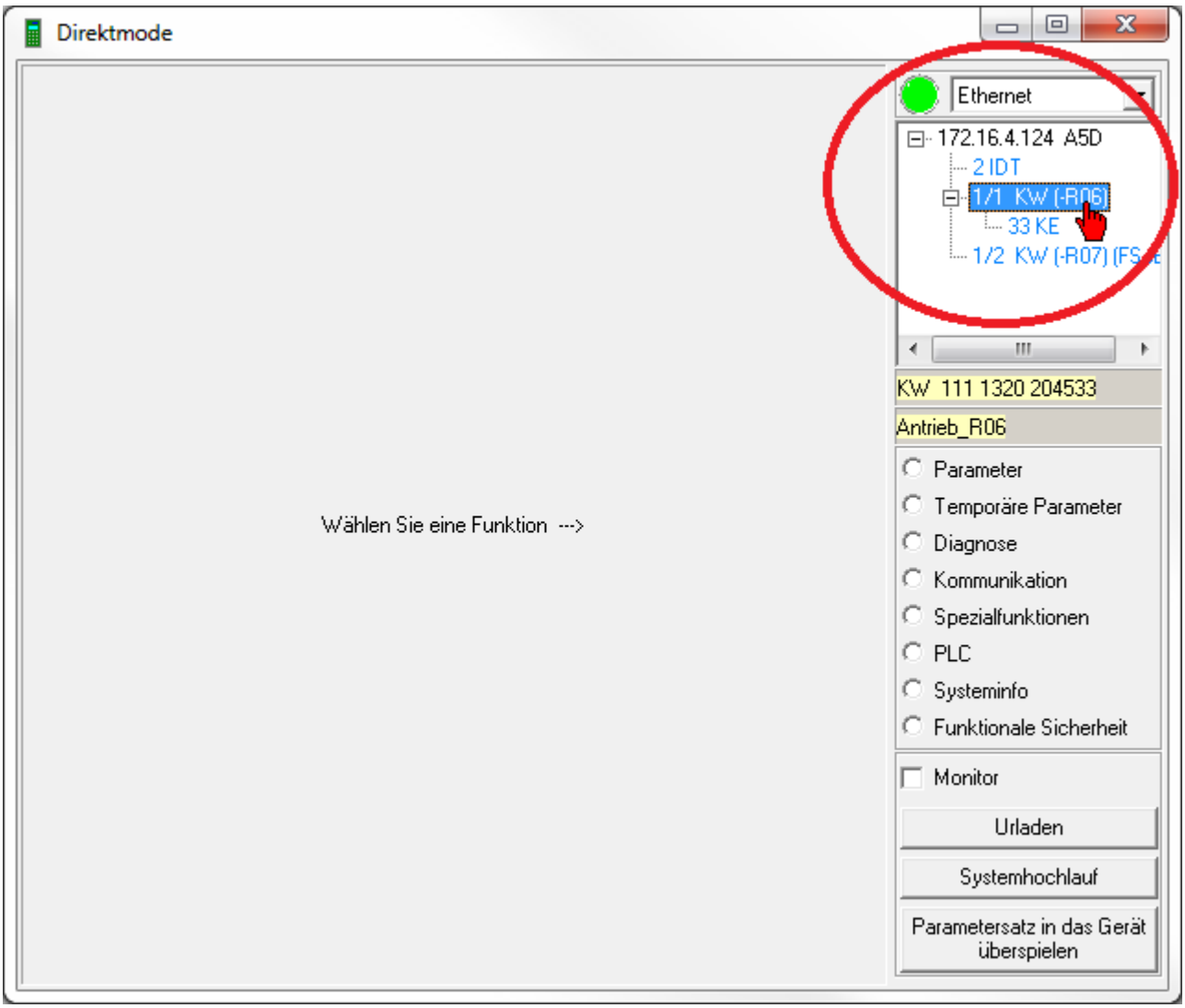

Doku_de - AIPEX PRO
Projekt
Online
Bearbeiten
Ansicht
Extras
Inbetriebnahme
Konfiguration
?
PC
Eigenschaften - PC
Serieller Anschluss
Ethernet
CAN-Adapter
EtherCAT
USB-Adapter
SERCOS III
CAN-Gateway
Ansicht
ONLINE
Komponenten
Alle Elemente zeigen
Übernehmen
Wählen Sie ein Gerät aus der Liste der


Direktmode
Wählen Sie eine Funktion --->
Ethernet 1
172.16.4.124 A5D
2 IDT 2
1/1 KW (-R06)
1/2 KW (-R07) (FSoE
A5D 411 1350 204756 3
Steuerung 4
Parameter 5
Temporäre Parameter 6
Diagnose 7
Kommunikation 8
Spezialfunktionen 9
PLC 10
Systeminfo 11
Funktionale Sicherheit 12
Monitor 13
Urladen 14
Systemhochlauf 15
Parametersatz in das Gerät überspielen 16

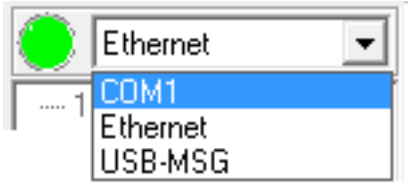
Nr.	Funktion
1	Aktivierte Schnittstelle zwischen PC und Gerät Grün: Verbindung verfügbar Rot: Keine Verbindung verfügbar
2	Anzeige der online erreichbaren Geräte
3	Firmware Version
4	System Name
5	Anzeige und Eingabemöglichkeit für Parameterwerte
6	Anzeige und Eingabemöglichkeit für temporäre Parameterwerte
7	Diagnosemodul
8	Anzeige und Eingabemöglichkeit für Busparameter
9	Inbetriebnahmefunktionen
10	Anzeige von PLC Eigenschaften
11	Anzeige von Systeminformationen
12	Option Funktionale Sicherheit
13	Funktion 'Monitor': Siehe 'Inbetriebnahme - Monitor' auf Seite 120.
14	Funktion 'Urladen' Die Funktion 'Urladen' setzt AMK Geräte in den Defaultzustand (Auslieferungszustand) zurück. Nach erfolgreichem 'Urladen' muss ein Systemneustart durchgeführt werden. Voraussetzungen zum 'Urladen': Eine direkte Verbindung zwischen AIPEX PRO und AMK Gerät, Passworteingabe.
15	Funktion 'Systemhochlauf' Auf dem angewählten Gerät wird ein 'Systemhochlauf' ausgeführt. Die Funktion 'Systemhochlauf' bewirkt eine Neuberechnung der Datenhaltung. (Istwerte bleiben bestehen, Antriebsbus z. B. EtherCAT läuft weiter...)
16	Funktion 'Parameter in das Gerät überspielen'

3.6.8.1 Verbindung im Direktmode aktivieren

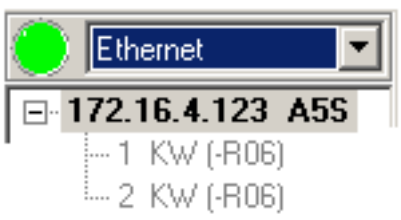
Nachfolgend wird beschreiben, wie Sie mit dem **Direktmode** auf angeschlossene AMK Geräte zugreifen.



Alle aktiven Schnittstellen (Verbindungen zwischen PC und AMK Geräten) werden zur Auswahl angeboten.
Wählen Sie die gewünschte Schnittstelle aus.
Statusanzeige:
Grün = Verbindung vorhanden
Rot = Verbindung nicht vorhanden bzw. unterbrochen



EtherCAT Geräte werden sofort angezeigt.
Bei anderen Antriebsbussen wie z.B. dem ACC-Bus, muss auf den jeweiligen Bus Master geklickt werden.

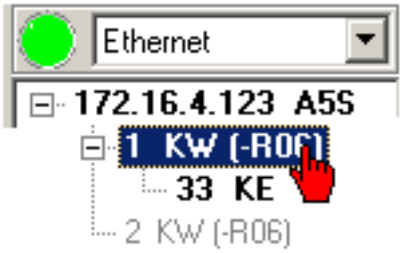


Status Grau: Gerätedaten wurden noch nicht ausgelesen.

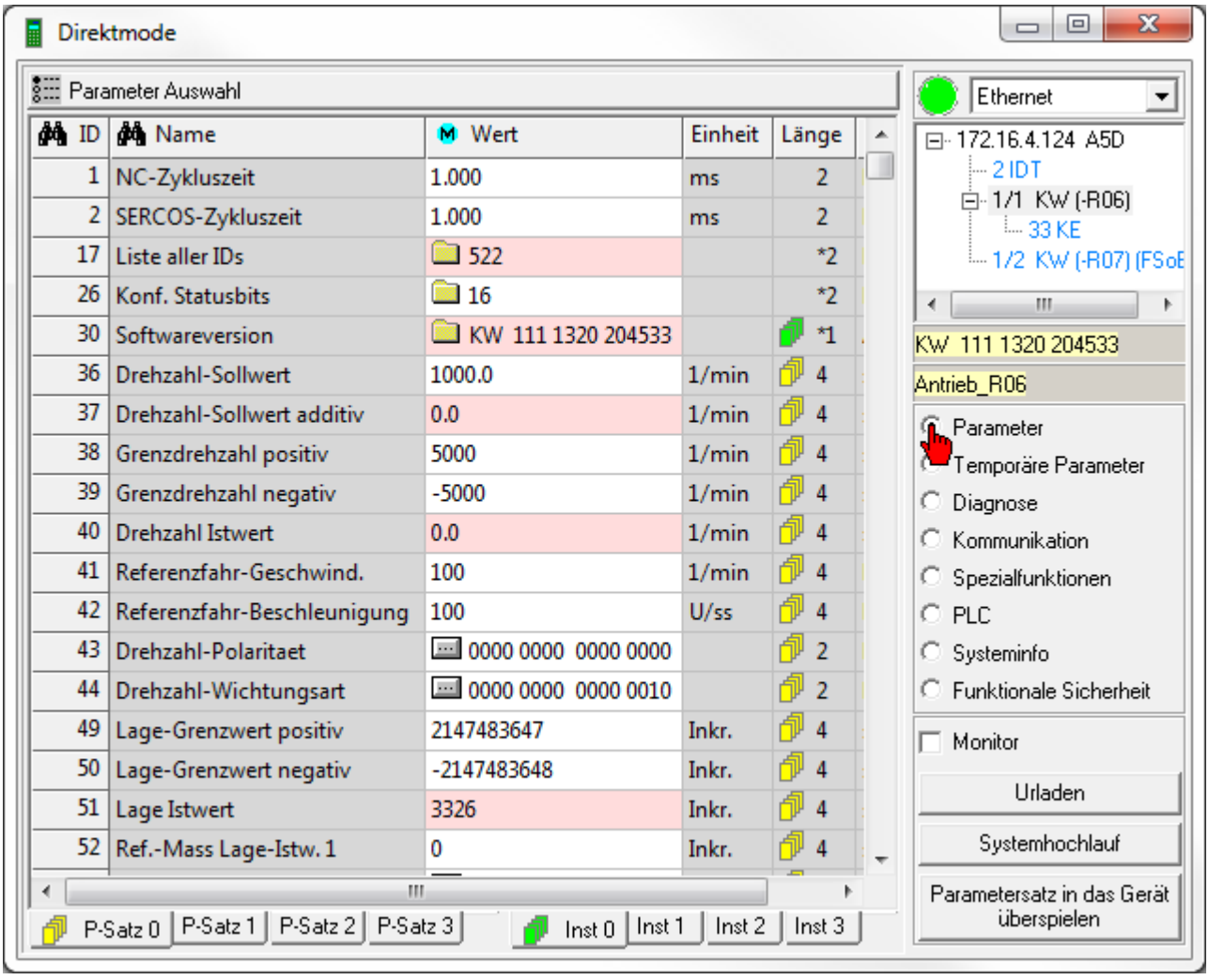
Status Schwarz: Gerätedaten wurden ausgelesen und auf dem PC gespeichert. Durch Doppelklicken eines schwarzen Gerätes werden die Daten neu ausgelesen.

Beispiel:

Das Einspeisemodul KE 33 ist über ACC-Bus verbunden.

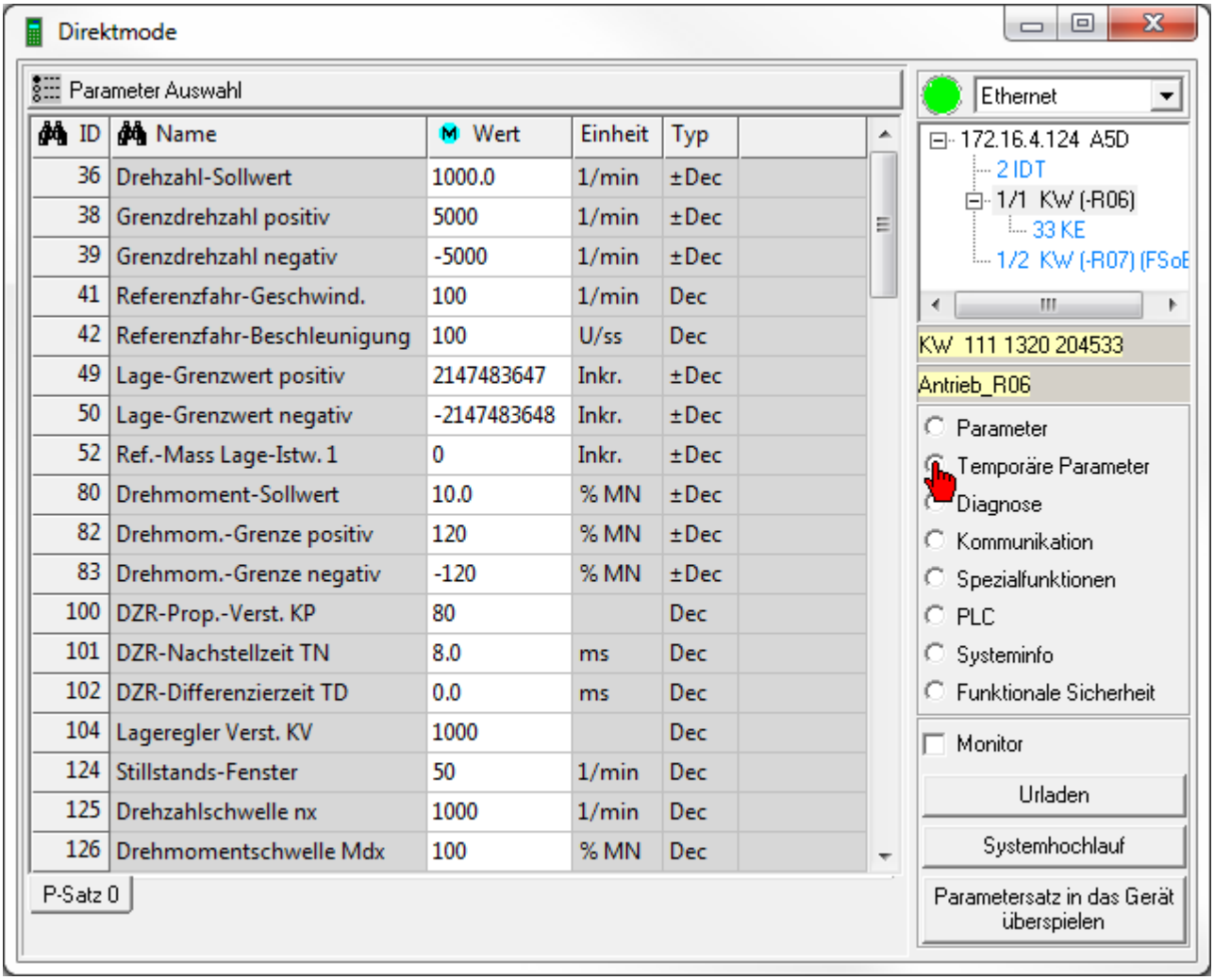


3.6.8.2 Parameter



'Parameter' zeigt vorhandene offline und online Werte des angewählten Gerätes an.

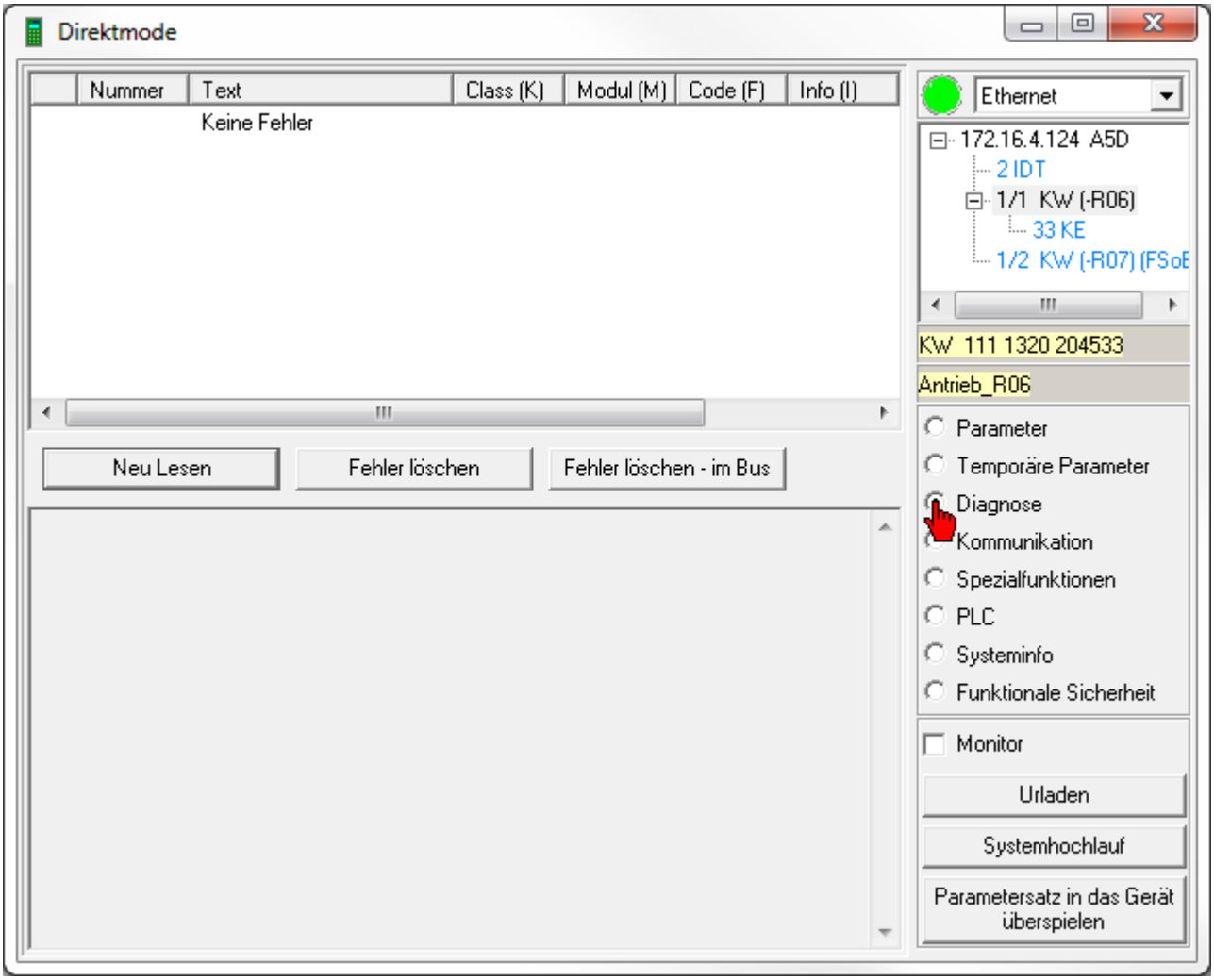
3.6.8.3 Temporäre Parameter



Änderungen in den 'Temporären Parametern' werden sofort im Antrieb wirksam.
In der Auswahlliste werden nur Parameter angeboten, die temporär änderbar sind.
Alle geänderten Werte werden in grüner Textfarbe dargestellt.

Beim Schließen des Temporären Fensters können Sie festlegen ob die Änderungen permanent gespeichert werden sollen.

3.6.8.4 Diagnose



Mit 'Diagnose' können die Diagnosemeldungen im angewählten Gerät ausgelesen werden.

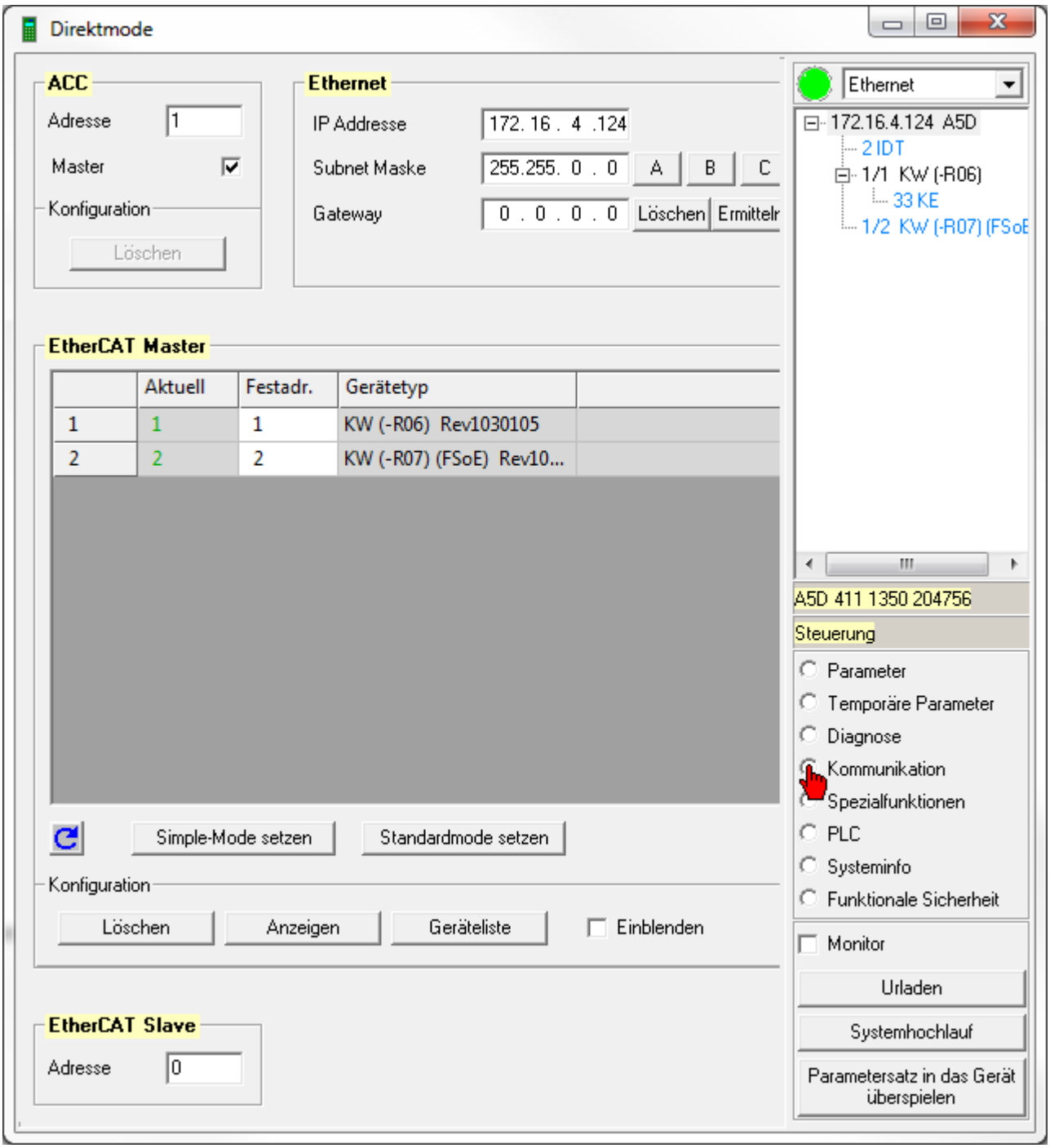
Klicken Sie eine Diagnosemeldung an, um die Erklärung einzublenden.
Weiterführende Informationen zur Diagnosemeldung erhalten Sie über die Info(I), I2 und I3
Die erste Diagnosemeldung in der angezeigten Liste ist der Hauptauslöser der Störung, weitere Meldungen können Folgefehler sein, die nach Beseitigung der Ursache für die erste Diagnosemeldung nicht mehr erscheinen.

Taster **'Neu Lesen'**
Diagnosemeldungen werden aus dem angewählten Gerät ausgelesen.

Taster **'Fehler löschen'**
'Fehler löschen' wird im angewählten Gerät ausgeführt.

Taster **'Fehler löschen - im BUS'**
Alle Fehler werden im angewählten Bus gelöscht. Wählen Sie dazu das Bussystem im Gerätebaum an.

3.6.8.5 Kommunikation



ACC-Bus
Adresse: ACC-Bus Teilnehmeradresse (ID34023).
Master: Klicken Sie in die Checkbox um das Gerät als ACC-Bus Master zu deklarieren. (ID34025).
Konfiguration: Taster **'Löschen'** löscht die ACC Bus Konfiguration (ID34036).

Ethernet
IP Adresse: Anzeige und Eingabefeld für die Geräte IP Adresse.

Subnet Maske:
Taster **'A'** / Class A Netz / 255.0.0.0
Taster **'B'** / Class B Netz / 255.255.0.0
Taster **'C'** / Class C Netz / 255.255.255.0

Gateway:

Taste **'Löschen'** deaktiviert die aktuelle Gateway Einstellung.

Taste **'Ermitteln'** trägt die Adresse des aktuellen Gateways des PC in die Gateway Einstellung des selektierten Gerätes ein (Diese Adresse ist häufig, aber nicht immer, für beide Teilnehmer identisch).

EtherCAT Master

Spalte 1: Physikalische Position nach dem EtherCAT Master.

Symbol EtherCAT Busstatus Operational (XML Konfiguration gültig).

Symbol EtherCAT Busstatus Pre-Operational (XML Konfiguration ungültig).

Spalte Aktuell: Aktuelle EtherCAT Adresse.

Spalte Festadresse: Manuelle Eingabe der Adresse durch den Benutzer (Festadresse).

Gerätetyp: Gerätetyp und EtherCAT Revisionsstand.

Vorgabe: Adressinformation aus dem automatisch generierten XML Nachrichtenkonfigurationsfile.

Taste **'Richtungspfeil'** liest die Geräteliste neu aus.

Taste **'Simple-Mode setzen'** löscht alle Festadressen in den Slaves.

Taste **'Standardmode setzen'** setzt in den Slaves alle aktuell vergebenen Adressen als Festadressen.

Konfiguration

Mit der Taste **'Löschen'** wird die aktuelle EtherCAT Konfiguration gelöscht.

Mit der Taste **'Anzeigen'** wird die aktuelle Gerätekonfiguration wird aus den Parametern (IDs) ausgelesen und angezeigt.

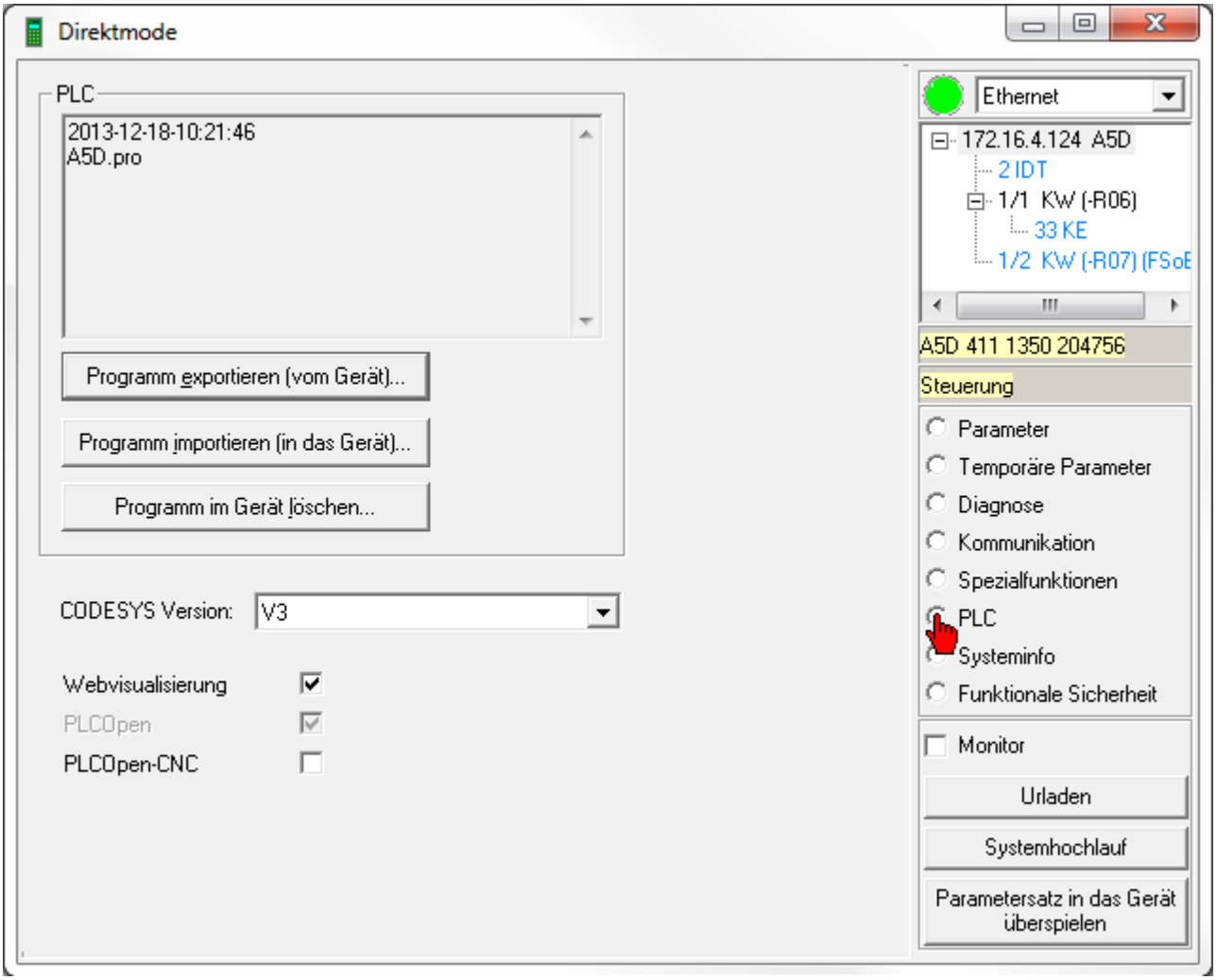
Mit der Taste **'Geräteliste'** wird aus der aktuellen Gerätekonfiguration die Geräteliste extrahiert und im Editor angezeigt.

Checkbox **'Einblenden'**: In die EtherCAT Adresstabelle können zu Vergleichszwecken die Vorgaben aus der Bus-Konfiguration eingeblendet werden. Differenzen zum Istzustand werden dabei orange markiert.

EtherCAT Slave

Adresse: Anzeige und Eingabefeld für EtherCAT Slave Adresse.

3.6.8.6 PLC



PLC
Wiedergabe der ID34172 'PLC Projektinfo'. Folgende Daten könne über den PLC Editor (Menüpunkt **Projekt → Projektinformation**) eingegeben und angezeigt werden.

- Datum
- Projektname
- Titel
- Version
- Autor
- Kommentar

Die Funktionen **'Programm importieren'** und **'Programm im Gerät löschen'** sind mit dem AMK Servicepasswort geschützt.
Das PLC Programm ist in der ID34159 'PLC Dateien' hinterlegt.

Taster **'Programm exportieren'**
Das PLC Programm wird vom angewählten Gerät gelesen und in einem frei wählbaren Ordner auf der PC- Festplatte gespeichert.
(Erstellt wird eine Datei mit der Endung *.bin)

Taster **'Programm importieren'**
Das PLC Programm wird aus einer Datei (PC Festplatte) gelesen und in das angewählte Gerät geschrieben.
Nur von AIPEX PRO exportierte *.bin PLC Dateien können importiert werden.

Taster **'Programm im Gerät löschen'**
Das PLC Programm wird im angewählten Gerät gelöscht.

CODESYS Version:
Auswahl der CODESYS Version mit der anschließend auf der Steuerung programmiert wird.

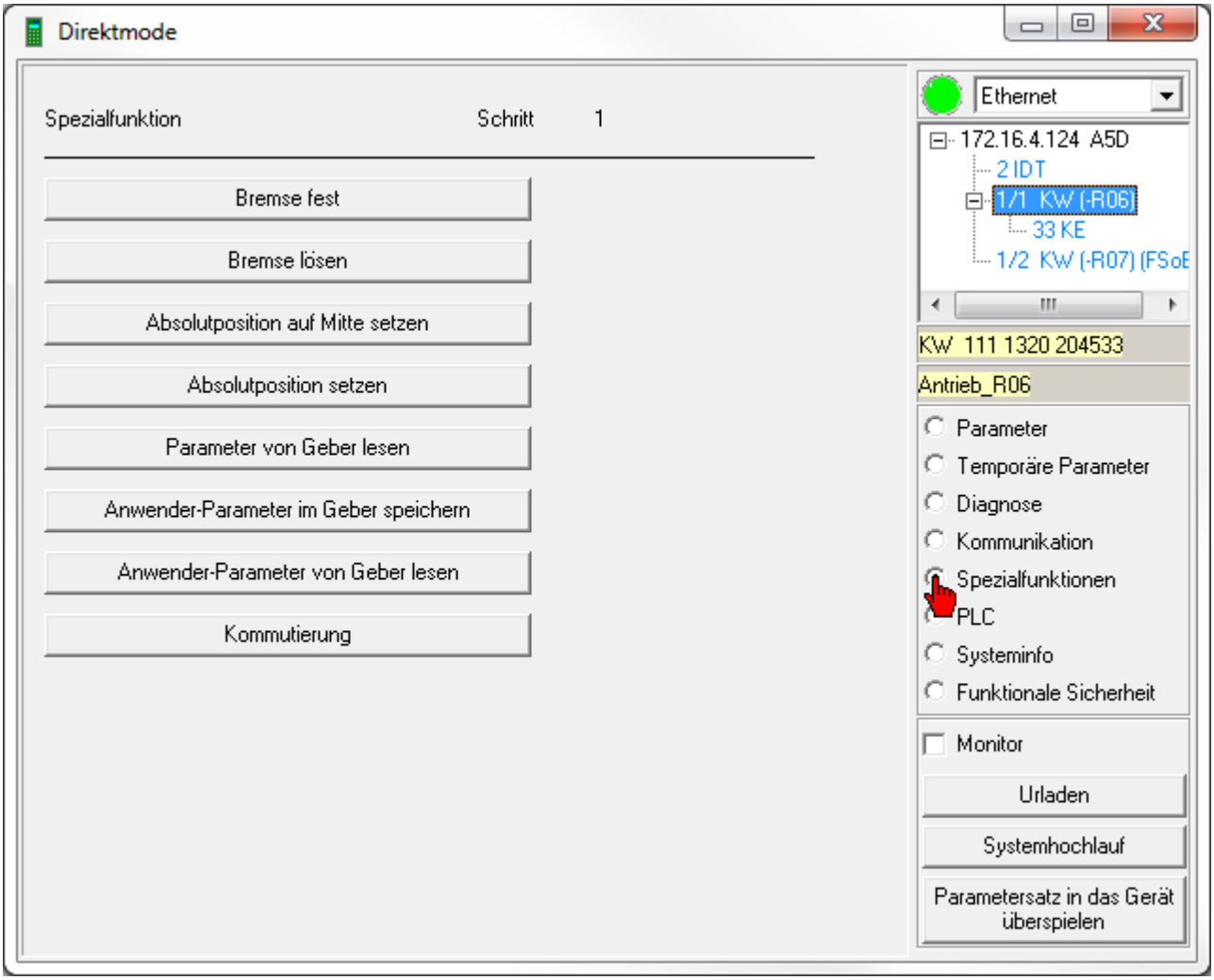
Die Standardeinstellung einer A4 und A5 Steuerung ist CODESYS V2.
Die Standardeinstellung einer A6 und iSA Steuerung ist CODESYS V3.

Steuerungsoptionen
Gültig für CODESYS V3.
Mit Hilfe der Checkboxen können die ab Werk freigeschalteten CODESYS Optionen deaktiviert und bei Bedarf wieder aktiviert werden.

Nach dem de- oder aktivieren muss die Steuerung neu gestartet werden (24 VDC AUS/EIN).

Beispiel:
Aus einer iSA Softmotion Steuerung mit Zusatzoption PCO (PLCopen) und Option VIS (Visualisierung), Zielsystem: ArmPLCopenControlWithVisu V3
wird nach Abwahl der Ckeckbox 'PLCopen' eine iSA Steuerung mit Zusatzoption VIS (Visualisierung), Zielsystem: ArmControlWithVisu V3.

3.6.8.7 Spezialfunktionen



Taster **'Bremse fest'**

Motorhaltebremse wird geschlossen.

Bei Umrichtern mit der Option "Endstufenfreigabe EF" oder "STO" kann die Bremse erst angesteuert werden, wenn die Endstufenfreigabe gesetzt ist.

Taster **'Bremse lösen'**

⚠ GEFAHR	
	Lebensgefahr durch hängende Achsen! Die optional im Motor vorhandene Motorhaltebremse ist eine Haltebremse und ist als Personenschutz NICHT ausreichend. Hängende Achsen können herabfallen und zu schweren Verletzungen führen. **Gegenmaßnahmen:** • Hängende Achsen müssen mechanisch gegen Herunterfallen gesichert werden, z. B. mit einer Fangeinrichtung oder einer zusätzlichen externen Bremse. • Personen dürfen sich nicht unter hängenden Lasten aufhalten

Motorhaltebremse wird geöffnet.

Bei Umrichtern mit der Option "Endstufenfreigabe EF" oder "STO" kann die Bremse erst angesteuert werden, wenn die Endstufenfreigabe gesetzt ist.

Taster **'Absolutposition auf Mitte setzen'**

Reserviert für AMK interne Nutzung!

Taster **'Absolutposition setzen'**

Reserviert für AMK interne Nutzung!

Taste **'Parameter von Geber lesen'**

Antrieb darf nicht bestromt sein!
Das Service Kommando darf nicht ausgeführt werden, wenn der Antrieb in Regelung ist. Reglerfreigabe muss AUS sein (RF = 0).

Die über ID32841 'Geberliste Motor' festgelegten Parameterwerte werden aus dem Geberspeicher eines Absolutwertgebers gelesen und im aktuellen Parametersatz gespeichert.

Taste **'Anwender-Parameter im Geber speichern'**

Antrieb darf nicht bestromt sein!
Das Service Kommando darf nicht ausgeführt werden, wenn der Antrieb in Regelung ist. Reglerfreigabe muss AUS sein (RF = 0).

Die aktuellen Werte der in ID32842 'Geberliste Anwender' eingetragenen Parameter werden in den Geberspeicher eines Absolutwertgebers geschrieben.

Nachdem die Parameter in den Geber geschrieben wurden, muss Netz AUS/EIN durchgeführt werden.

Taste **'Anwender-Parameter von Geber lesen'**

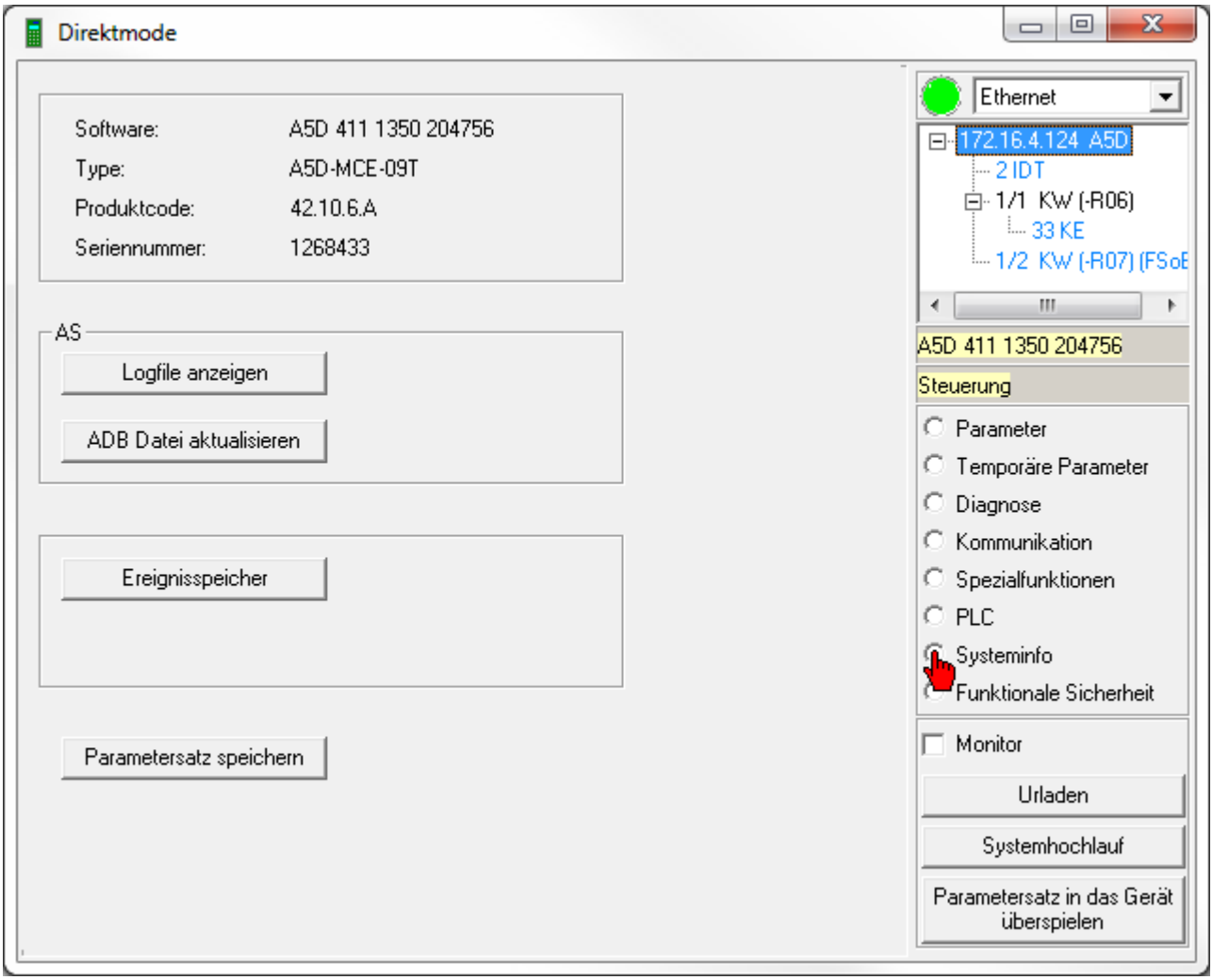
Antrieb muss stillstehen!
Das Service Kommando darf nur ausgeführt werden, wenn der Antrieb im Stillstand ist. Bei Achsen auf die Kräfte von außen wirken, z. B. hängende Achsen, muss der Anwender sicherstellen, dass sich die Achse nicht bewegen kann, während das Service Kommando aktiv ist.

Die über ID32842 'Geberliste Anwender' festgelegten Parameterwerte werden aus dem Geberspeicher eines Absolutwertgebers gelesen und im aktuellen Parametersatz gespeichert.

Taste **'Kommutierung'**

Reserviert für AMK interne Nutzung!

3.6.8.8 Systeminfo



Systeminfo

Folgende Informationen werden vom ausgewählten Gerät angezeigt:

- Software
- Type
- Produkt Code
- Seriennummer

AS - AMKAMAC Steuerung

Taster **'Logfile Anzeigen'**zeigt das Logfile der aktuell angewählten Steuerung an.

Taster **'ADB Datei aktualisieren'** aktuallisiert die ADB Datei in der aktuell angewählten Steuerung.

Taster **'Ereignisspeicher'**

ID34088 'Ereignisspeicher' des angewählten Gerätes wird angezeigt.

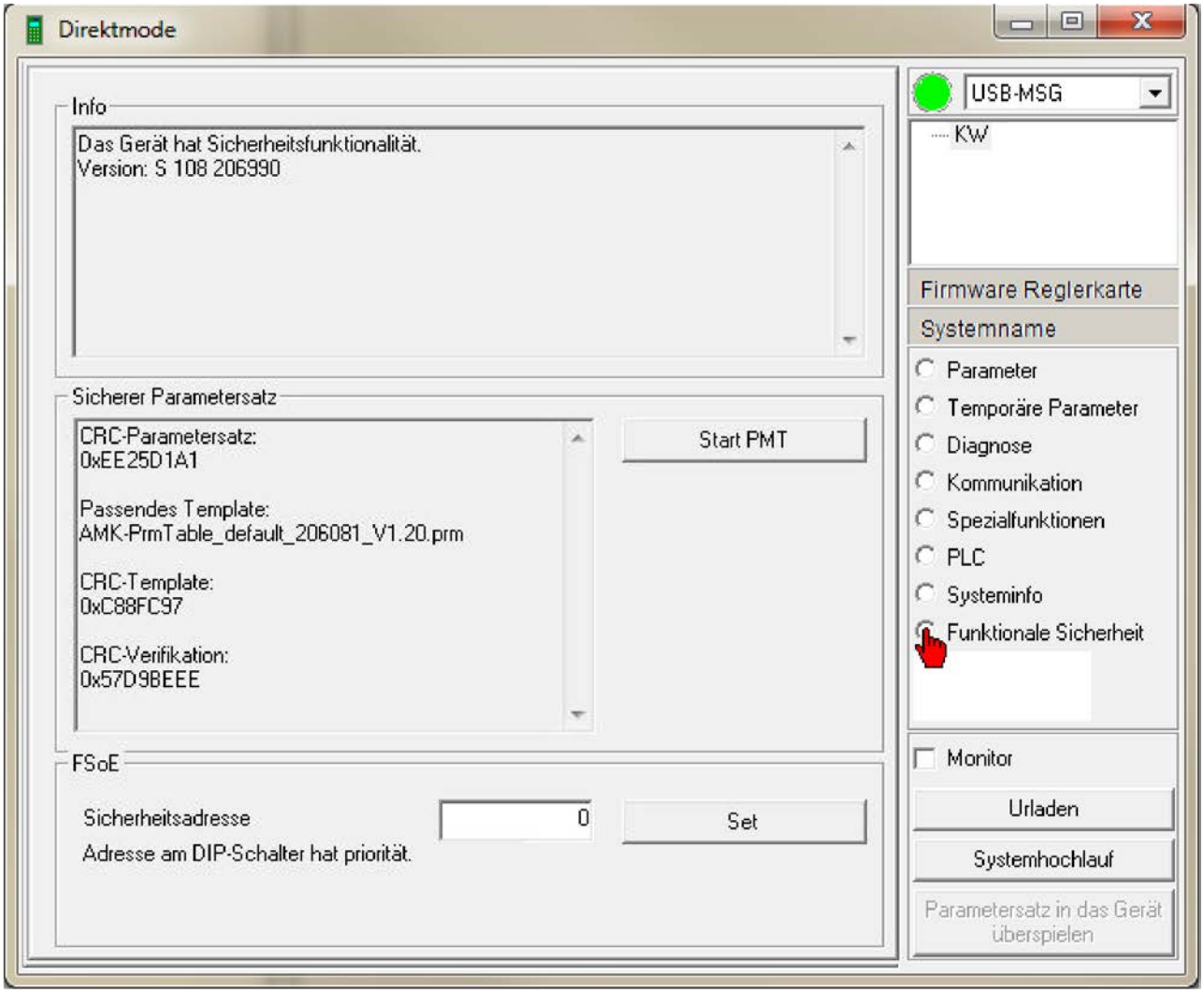
Weitere Informationen: Siehe 'Inbetriebnahme' auf Seite 116.

Taster **'Parametersatz speichern'**

Der Parametersatz des aktuell angewählten Gerätes wird auf den PC gespeichert.

Die erstellte Datei *.apu können Sie mit AIPEX PRO wieder öffnen und anschauen. Eine *.apu Datei kann nicht in ein Gerät überspielt werden. Erstellen Sie dazu ein AIPEX PRO Projekt.

3.6.8.9 Funktionale Sicherheit



Fensterinhalt 'Info'

Statusanzeige Verbindung, bzw. ob das verbundene Gerät eine Sicherheitsbaugruppe enthält.

Version: Firmware der Sicherheitsbaugruppe

Passendes Template zur Geräteversion (Firmwareversion der Sicherheitsbaugruppe)

Fensterinhalt 'Sicherer Parametersatz'

CRC-Parametersatz: Checksumme anwendungsspezifischer Sicherer Parametersatz

Passendes Template: Passendes Template zur Geräteversion (Firmwareversion der Sicherheitsbaugruppe)

CRC-Template: Checksumme Template

CRC-Verifikation: Checksumme anwendungsspezifischer Sicherer Parametersatz + Seriennummer Gerät

Der Taster **'Start PMT'** startet die Software SafePMT Parametereditor. (Parametereditor für die 'Sichere Parametrierung')

FSoE

Die eingegebene FSoE Adresse wird mit dem Taster **'Set'** in die ID33201 'Sichere Adresse' übernommen.

Eine eingestellte Adresse am Geräte Dip-Schalter hat Priorität und kann nicht überschreiben werden.

Bei einem Firmware-Update oder mit dem Urladen des Parametersatzes wird ID33201 auf 0 zurückgesetzt.

3.7 Funktionen

3.7.1 Aktualisierung der Geräte-Firmware

Die Firmware ist das Betriebssystem der Regler- und Optionskarten bzw. der PLC Steuerungen.

Der aktuelle Firmwarestand kann z. B. aus der ID30 'Softwareversion' ausgelesen werden.

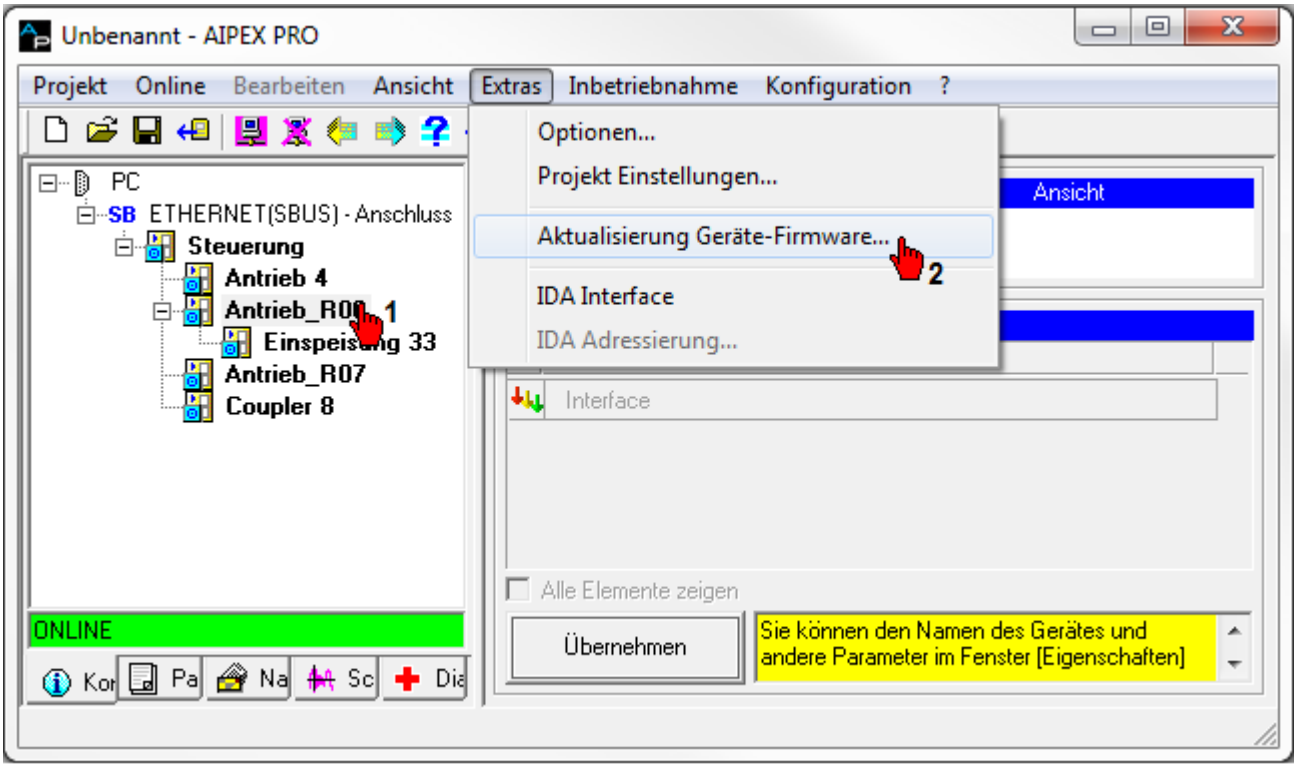
Die Funktion **'Aktualisierung Geräte Firmware'** ist vorrangig dazu bestimmt, um aus einem bestehenden Projekt, Geräte mit einer neuen Firmware zu aktualisieren.

Ohne Projekt kann die Funktion nicht genutzt werden. Erzeugen Sie in diesem Fall ein neues AIPEX PRO Projekt, indem Sie sich einloggen und die Gerätedaten übernehmen.

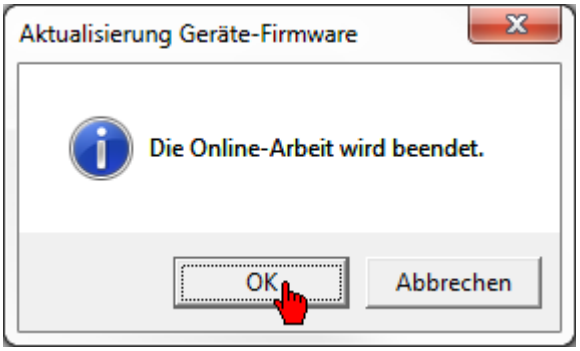
Die Firmware Dateien sind nicht in AIPEX PRO integriert. Sie müssen einmalig importiert werden. Sie erhalten die Firmware Dateien von AMK.

Wählen Sie im Gerätebaum das Gerät aus, welches aktualisiert werden soll.

Starten Sie die Funktion über **'Extras'** → **'Aktualisierung Geräte-Firmware'**



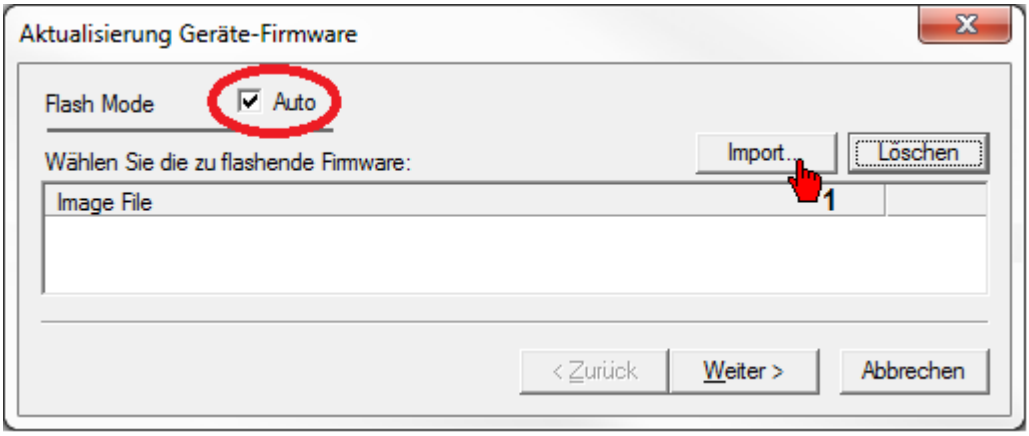
Falls Sie online sind, bestätigen Sie mit **'OK'**.



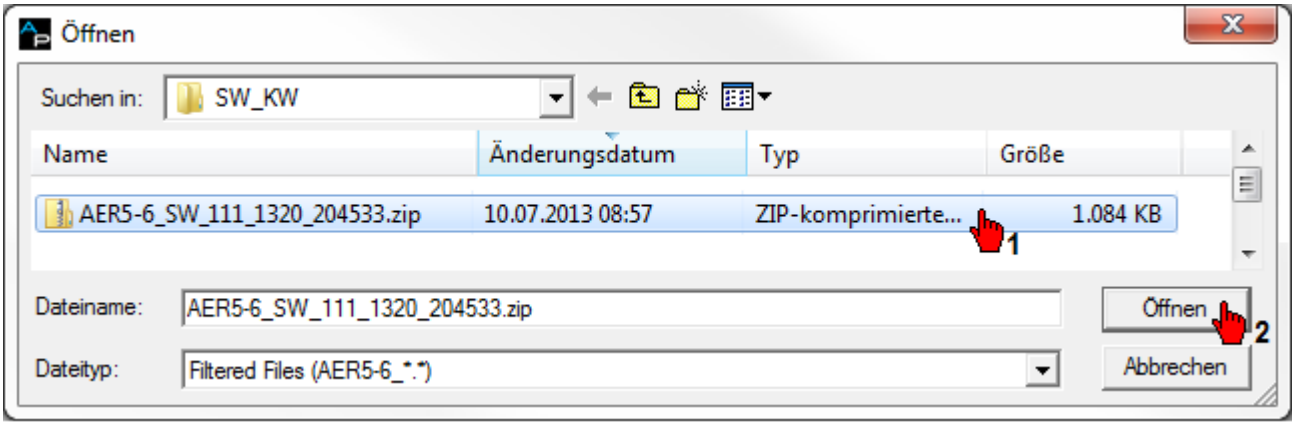
Die Funktion ist Passwort geschützt.
Autorisierte Personen erhalten das Passwort von AMK.

'Flash Mode Auto' aktiv. Das Gerät wird automatisch in den Flash Mode geschalten.
'Flash Mode Auto' inaktiv. Das Gerät muss manuell in den Flash Mode geschalten werden.

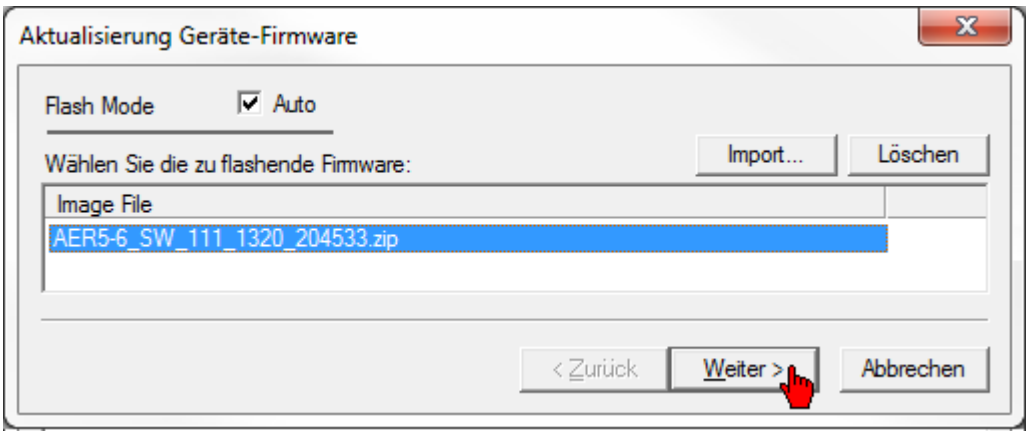
Über den Taster **'Import'** können Sie neue Firmware Dateien importieren.



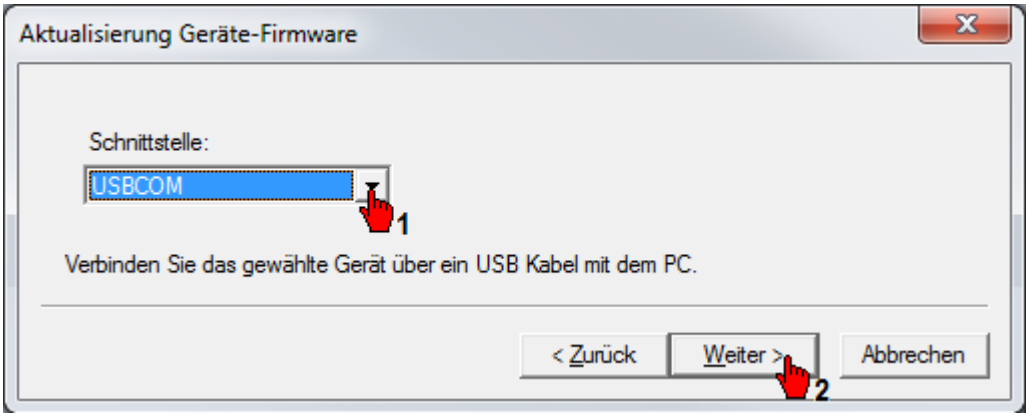
Eine einmal importierte Datei bleibt unter 'Image File' gespeichert und kann sofort angewählt werden.
Wählen Sie die Firmware Datei an. Bestätigen Sie mit **'Öffnen'**.



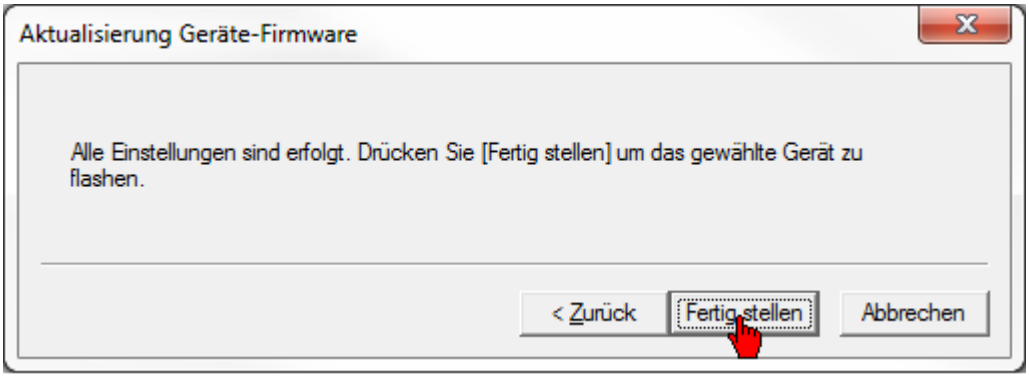
Bestätigen Sie mit **'Weiter'**.



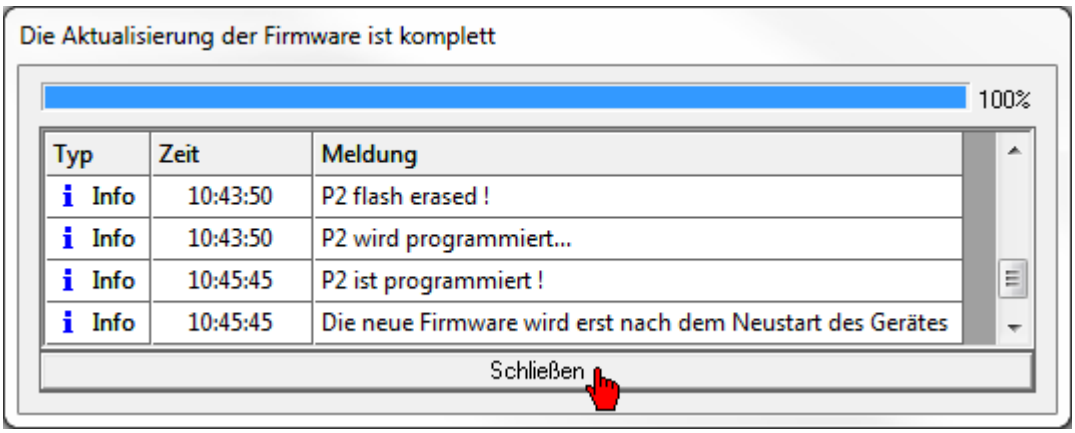
Stellen Sie eine Direktverbindung zwischen PC und Gerät her. Z. B. über eine USB oder EtherCAT Verbindung.
Wählen Sie die verwendete Schnittstelle an.



Mit **'Fertig stellen'** starten Sie die Aktualisierung der Firmware.



Fortschrittsanzeige



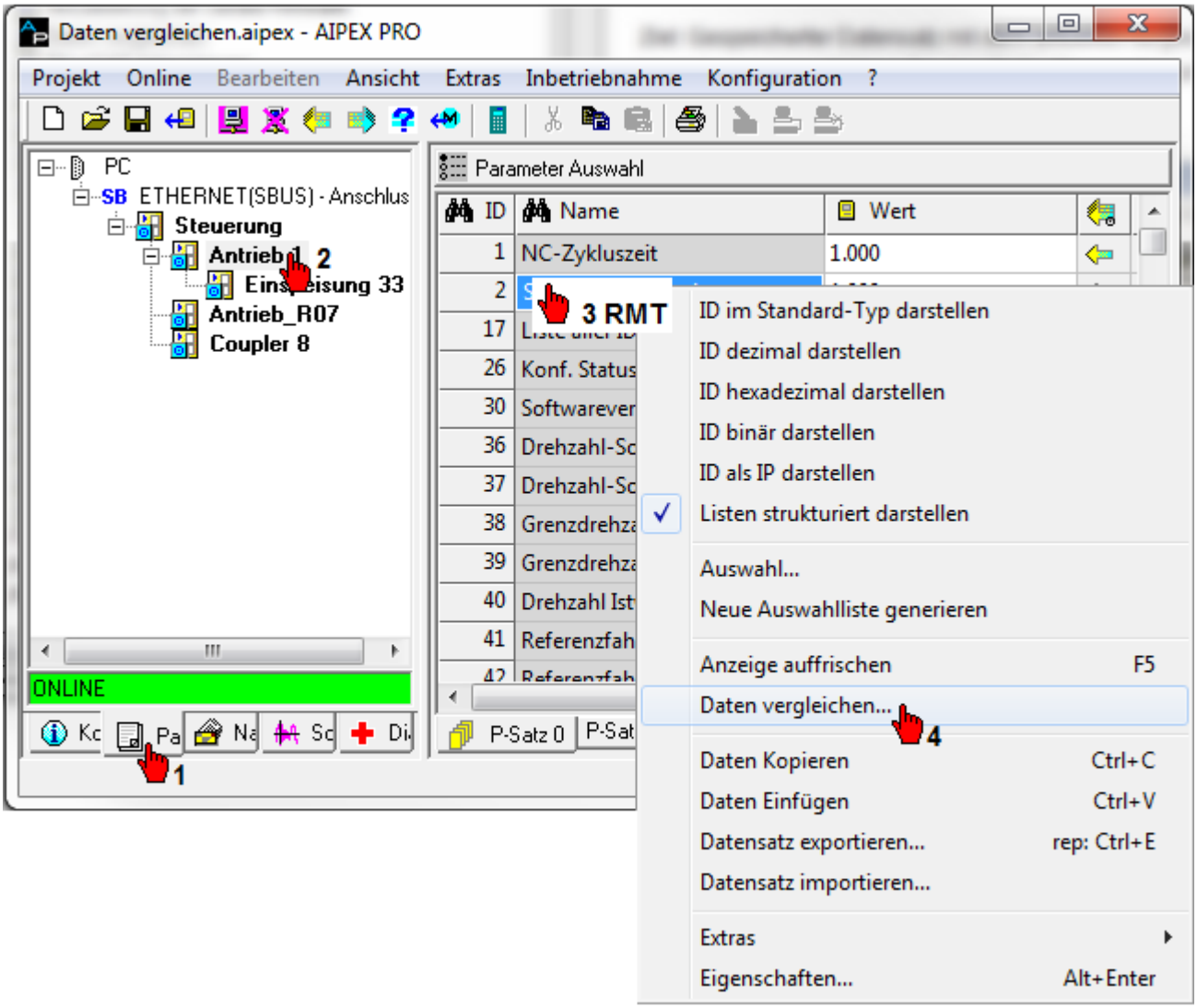
Die neue Firmware wird erst nach dem Neustart des Gerätes aktiviert!

Abhängig von der neuen Firmwareversion ist ein anschließendes 'Urladen' nötig.

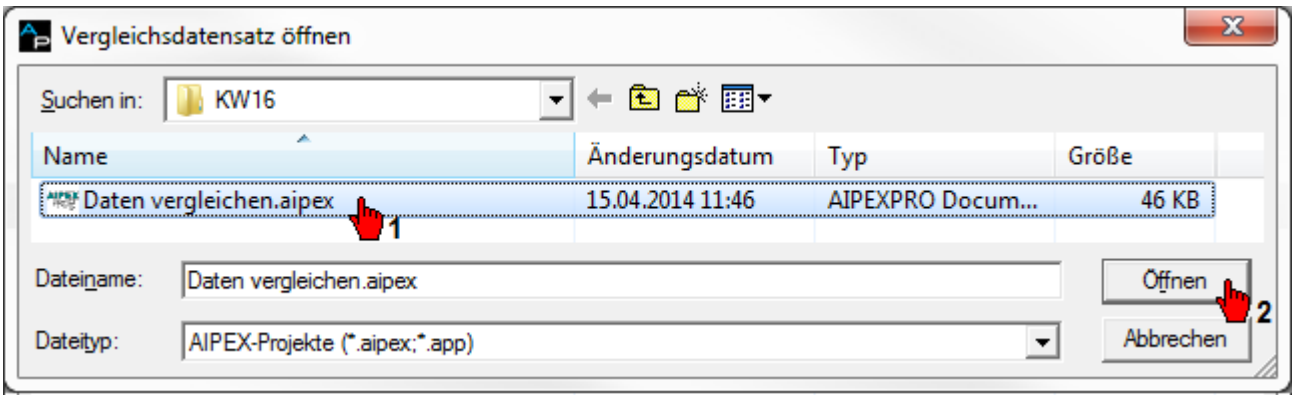
3.7.2 Daten vergleichen

Mit der Funktion 'Daten vergleichen' können die Inhalte des aktuell angezeigten Parametersatz mit dem Inhalt eines gespeicherten Datensatzes verglichen werden.

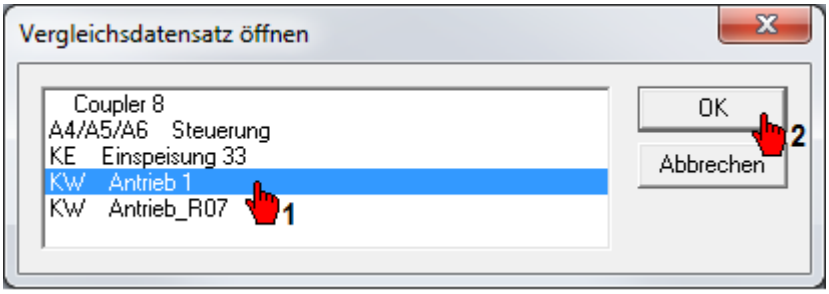
Wählen Sie den Reiter **'Parameter'** an. Durch anwählen eines beliebigen Parameters mit der rechten Maustaste öffnet sich das Kontextmenü. Wählen Sie **'Daten vergleichen'** an.



Wählen Sie den Ordner und den Vergleichsdatensatz an. Bestätigen Sie die Auswahl mit **'Öffnen'**.

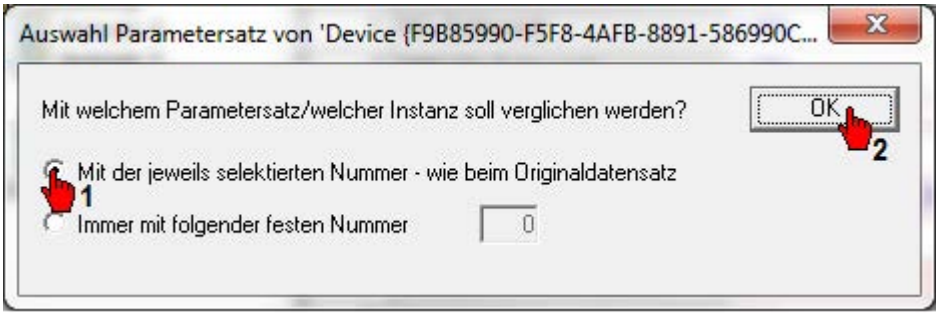


Enthält der Vergleichsdatensatz mehrere Geräte müssen Sie das Vergleichsgerät anhand des Stationsnamen auswählen.



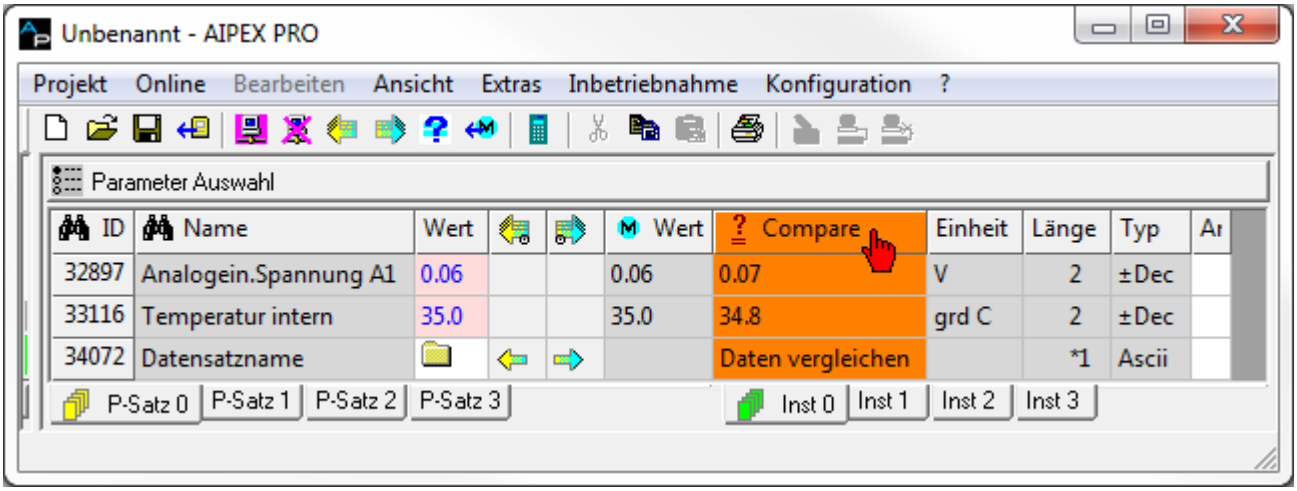
Wird die obere Auswahlmöglichkeit gewählt, wird immer der gleiche Parametersatz bzw. die gleiche Instanz der Vergleichsdatei wie beim Originaldatensatz verwendet, d. h. die Kartenreiter ('P-Satz 0', 'P-Satz 1'... 'Inst 0', 'Inst 2'..)) unterhalb der Tabelle gelten auch für den Vergleichsdatensatz.

Mit der unteren Auswahlmöglichkeit wird ein fester Wert für den Parametersatz und die Instanz der Vergleichsdatei gewählt. Die Kartenreiter ('P-Satz 0', 'P-Satz 1'... 'Inst 0', 'Inst 2'...) unterhalb der Tabelle haben dadurch keine Auswirkung auf die Vergleichsdatei.

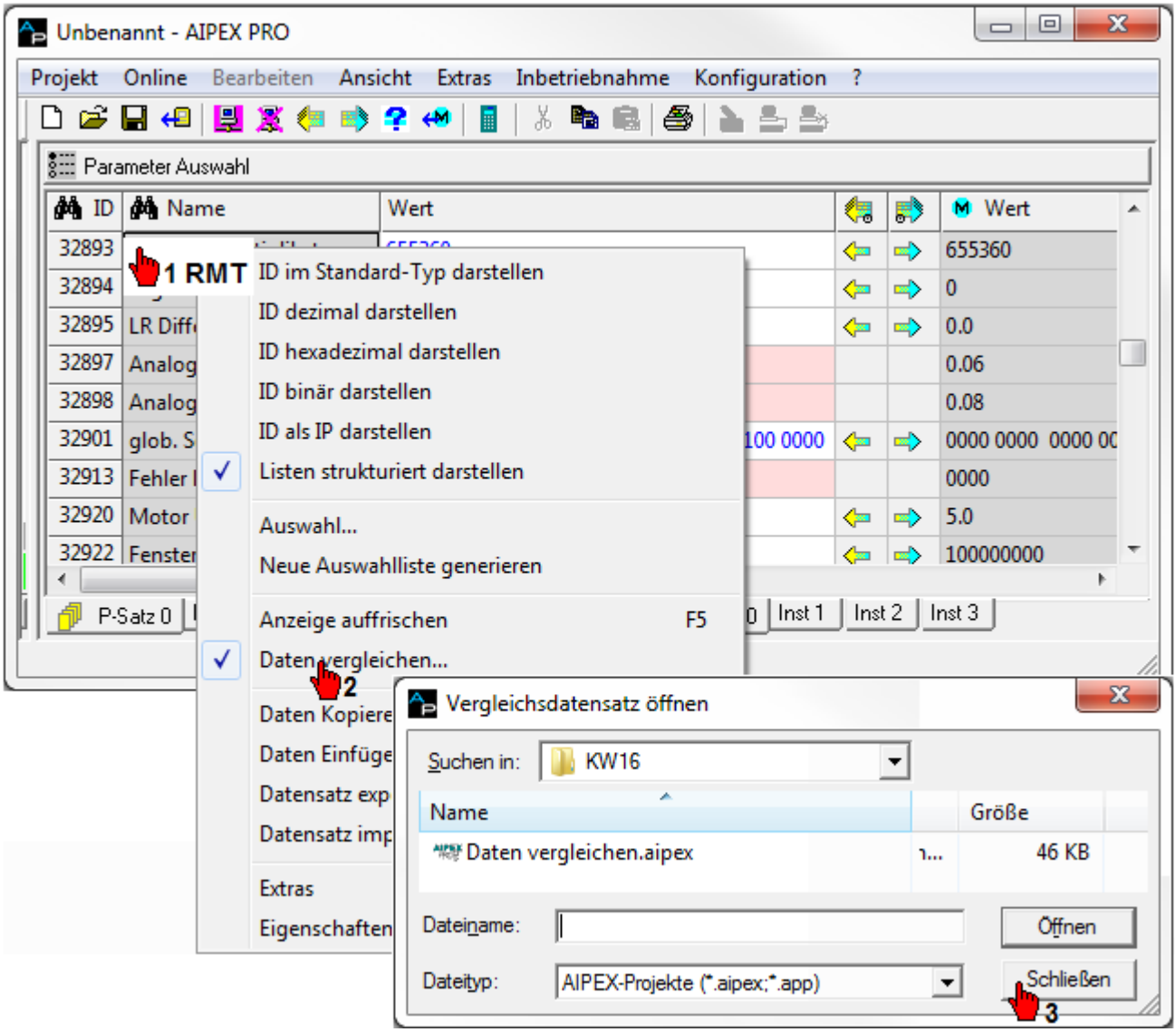


Die Daten der Vergleichsdatei werden in einer zusätzlichen Spalte angezeigt, sie können jedoch nicht verändert werden. Alle Daten der Vergleichsdatei, die sich vom Originaldatensatz unterscheiden, werden orange hinterlegt.

Durch Klicken in das Titelfeld der Vergleichsdatei wird zwischen der Anzeige aller Parameter und einer Anzeige, bei der nur die sich unterscheidenden Parameter dargestellt werden, gewechselt.



Öffnen Sie erneut das Kontextmenü mit der rechten Maustaste. Durch Anwählen des Menüpunktes 'Datensatz vergleichen' kann im darauffolgenden Dialogfenster der Vergleich mit **'Schließen'** beendet werden.



3.7.3 Diagnose mit AIPEX PRO

Mit **'Diagnose'** können die Diagnosemeldungen im angewählten Gerät ausgelesen werden.

Klicken Sie eine Diagnosemeldung an, um die Erklärung einzublenden.

Weiterführende Informationen zur Diagnosemeldung erhalten Sie über die Info(I), I2 und I3.

Die erste Diagnosemeldung in der angezeigten Liste ist der Hauptauslöser der Störung, weitere Meldungen können Folgefehler sein, die nach Beseitigung der Ursache für die erste Diagnosemeldung nicht mehr erscheinen.

Taster **'Neu Lesen'**

Diagnosemeldungen werden aus dem angewählten Gerät ausgelesen.

Taster **'Fehler löschen'**

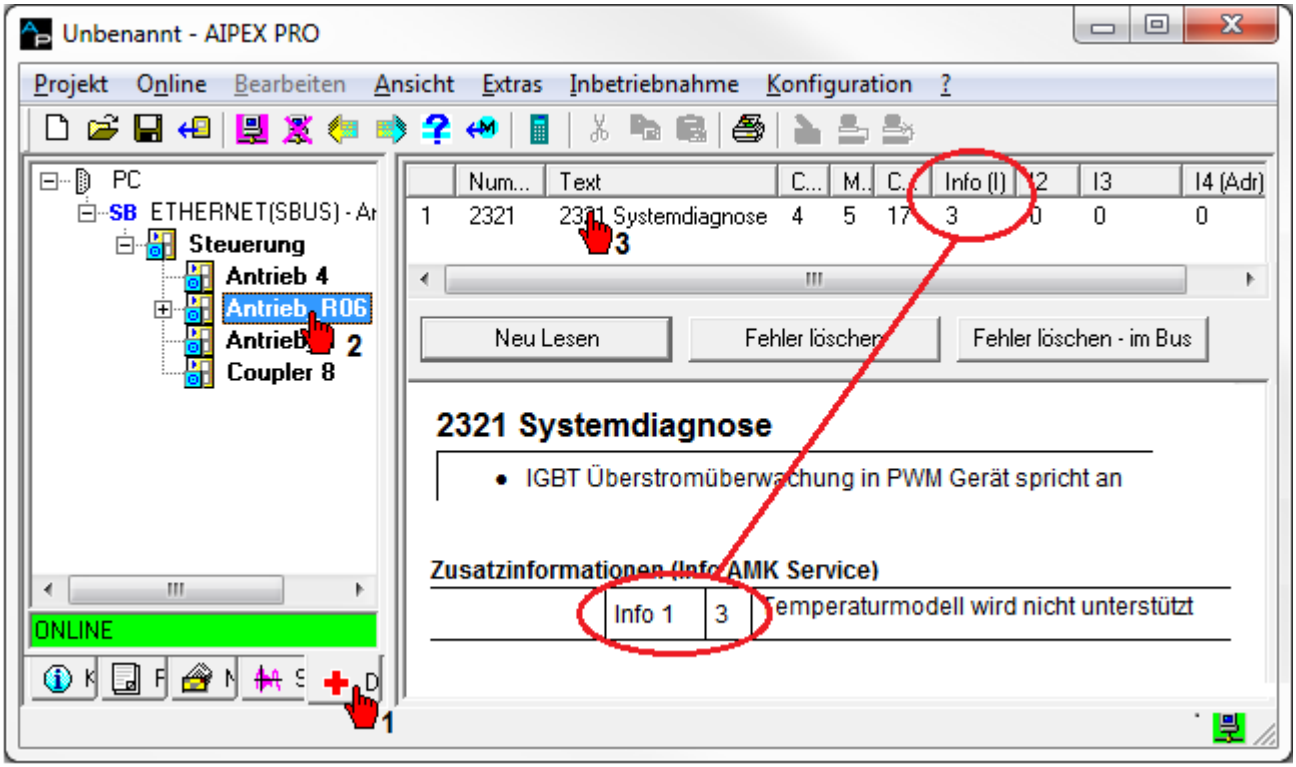
'Fehler löschen' wird im angewählten Gerät ausgeführt.

Taster **'Fehler löschen - im BUS'**

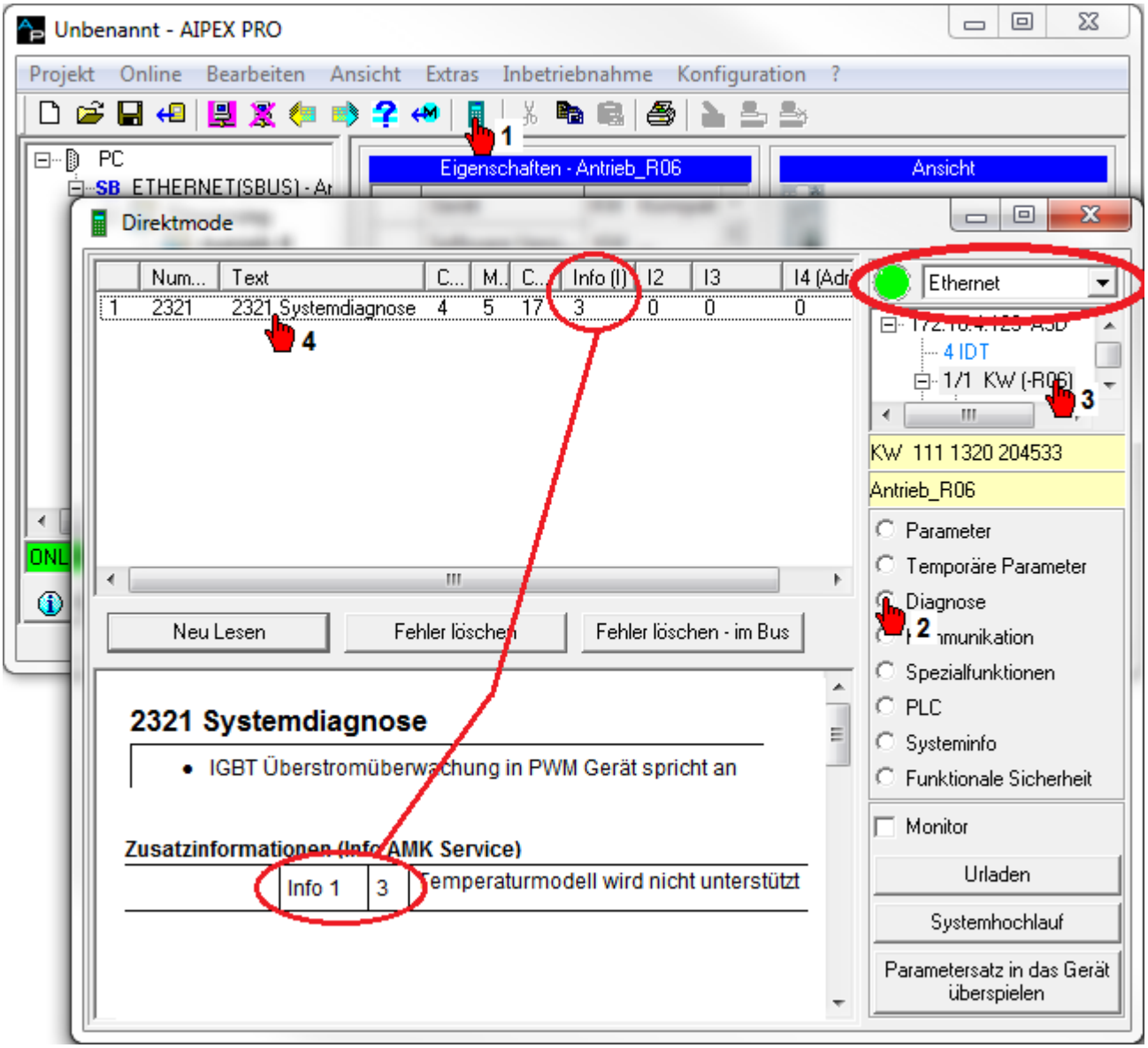
Alle Fehler werden im angewählten Bus gelöscht. Wählen Sie dazu das Bussystem im Gerätebaum an.

Nach erfolgreicher Fehlerbeseitigung bewirkt 'Fehler Löschen' einen erneuten Systemhochlauf und SBM wird gesetzt.

Diagnose im AIPEX PRO Projekt (Reiter 'Diagnose')



Diagnose im Direktmode

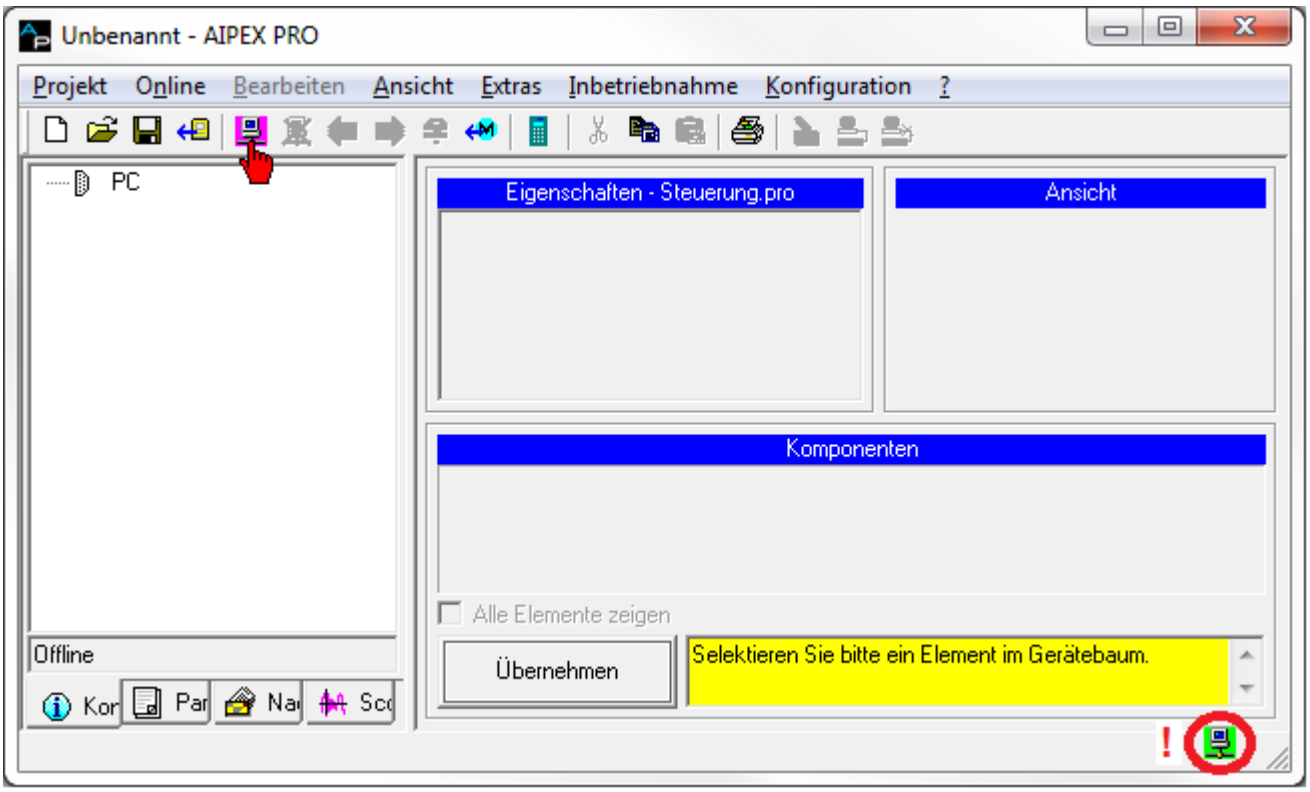


3.7.4 Einloggen

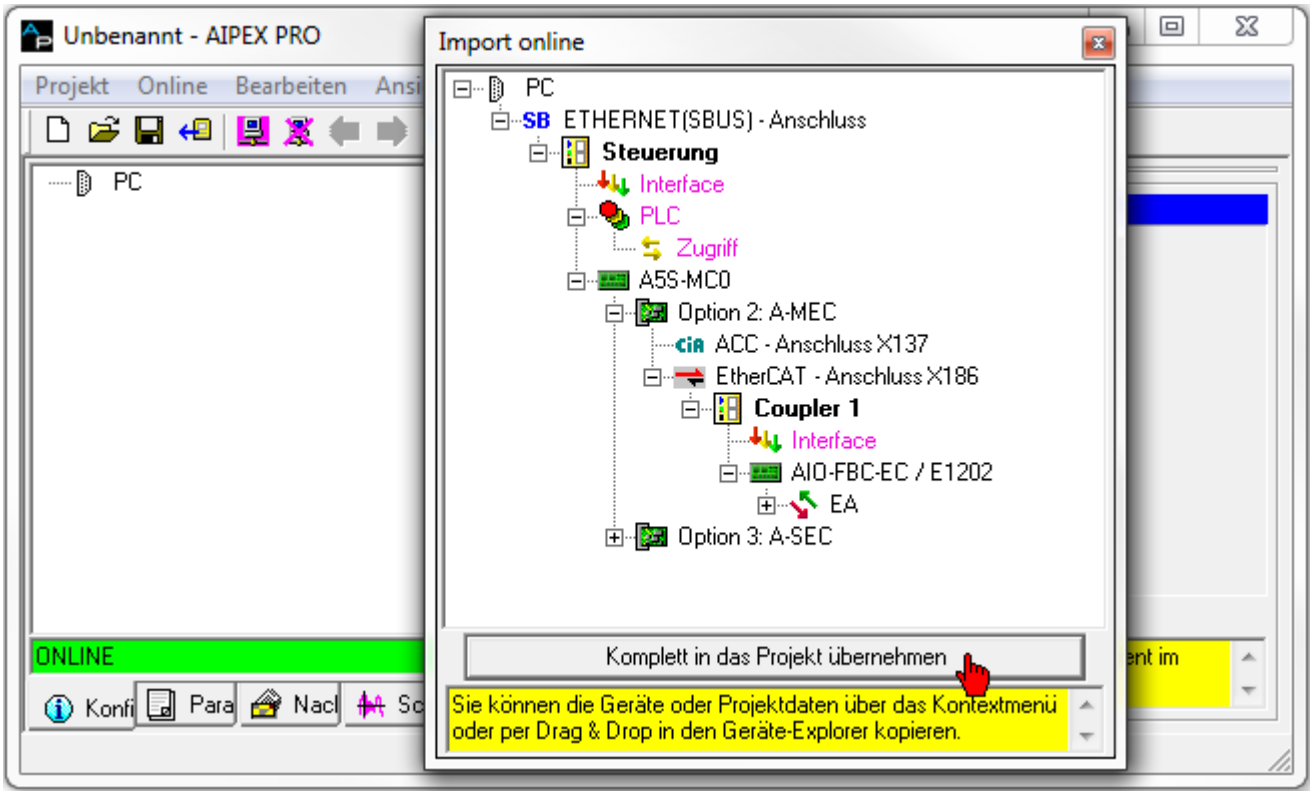
Einloggen und Projekt erstellen

Nutzen Sie diese Variante, wenn Sie anschließend die Gerätedaten als AIPEX PRO Projekt auf Ihrem PC speichern wollen.

Drücken Sie **'Einloggen'**

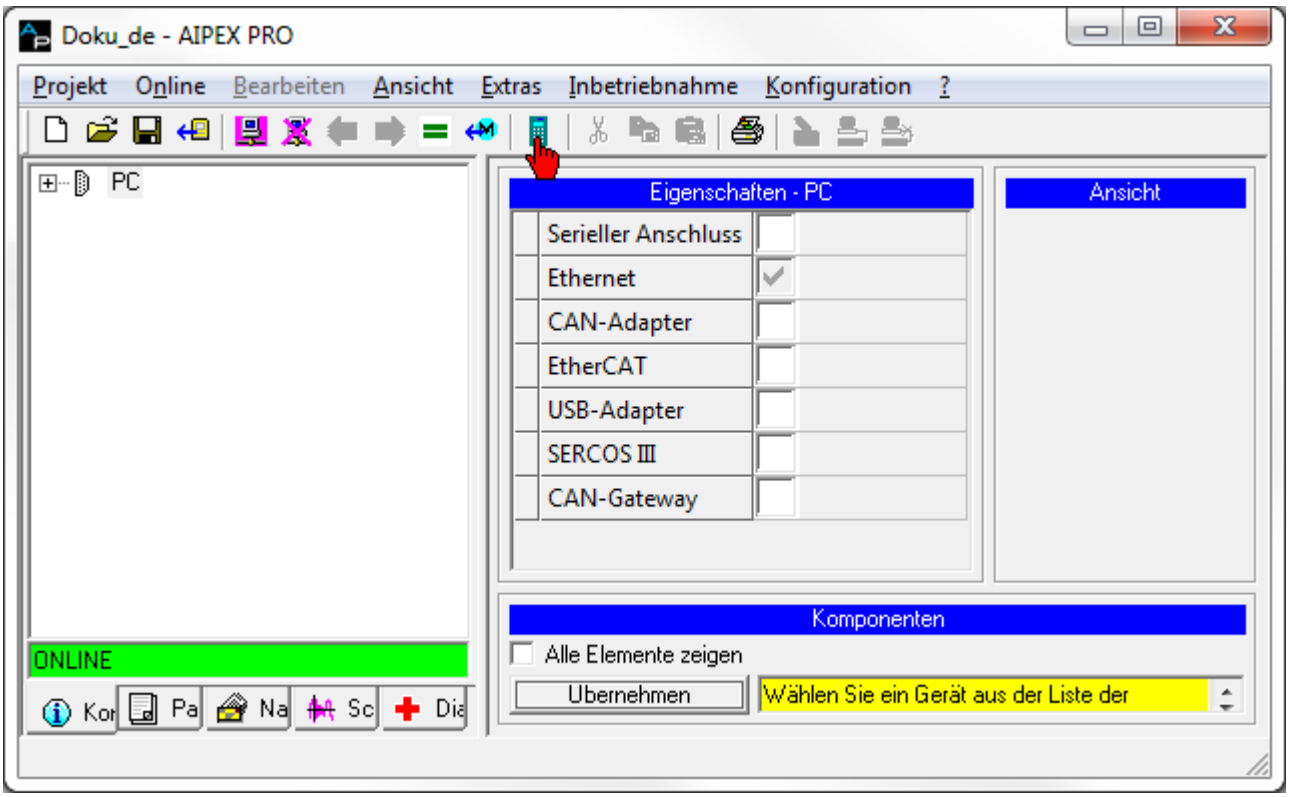


Übernehmen Sie die Gerätedaten in Ihr Projekt. Drücken Sie **'Komplett in das Projekt übernehmen'**



Einloggen über den Direktmode

Im **'Direktmode'** können Sie online auf AMK Geräte zugreifen.
Änderungen werden sofort in das Gerät übertragen. Wann die Änderung aktiv wird, hängt von den Parametereigenschaften ab.

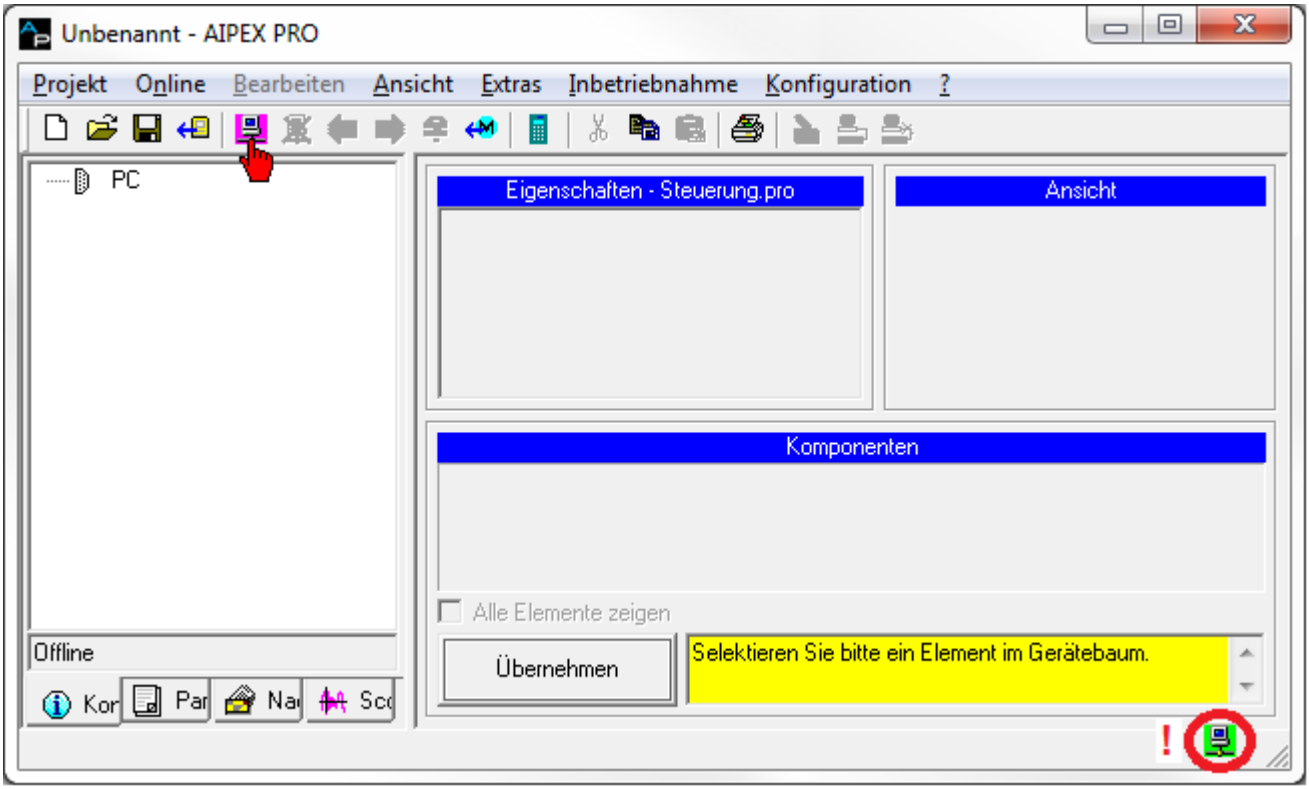


Siehe 'Direktmode' auf Seite 127.

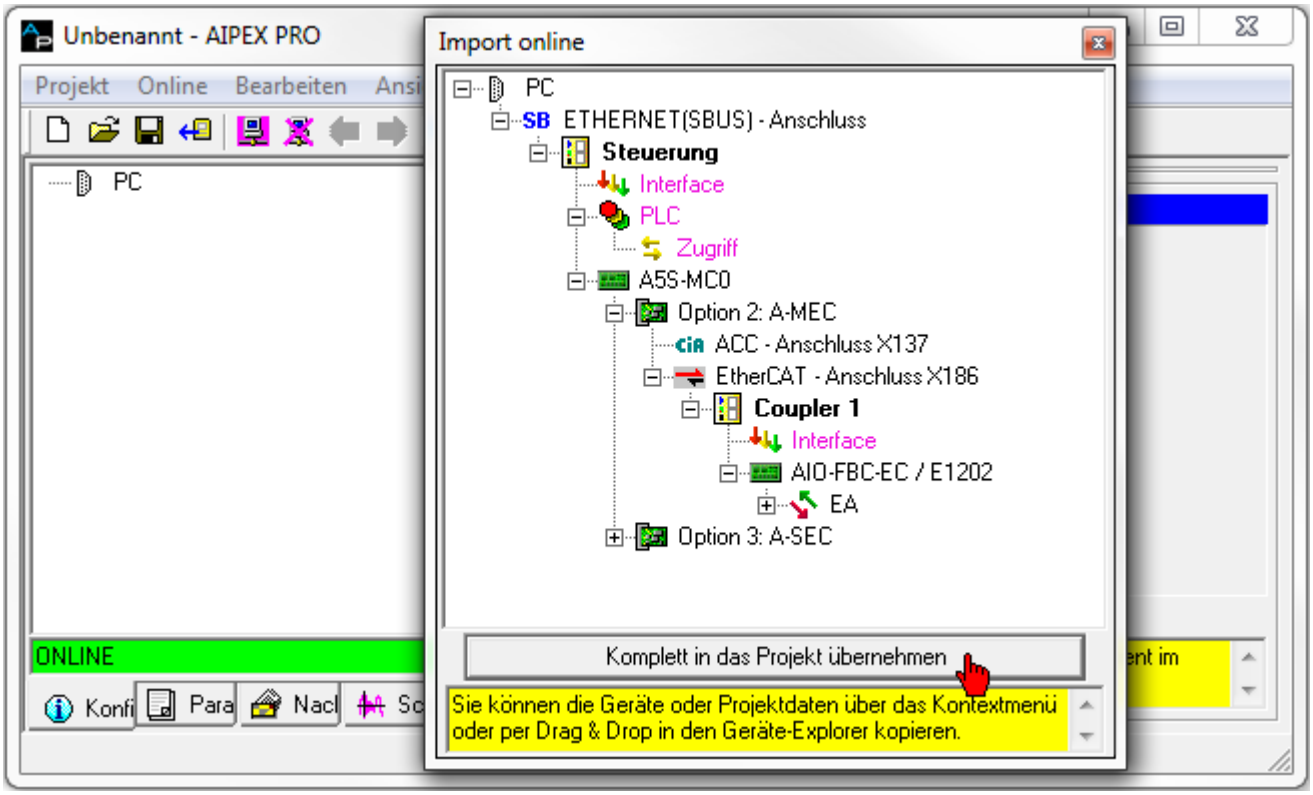
3.7.5 Gerätedaten auslesen und speichern

Im nachfolgenden Beispiel wird beschrieben wie Sie Gerätedaten auslesen und auf dem PC speichern.

Drücken Sie **'Einloggen'**

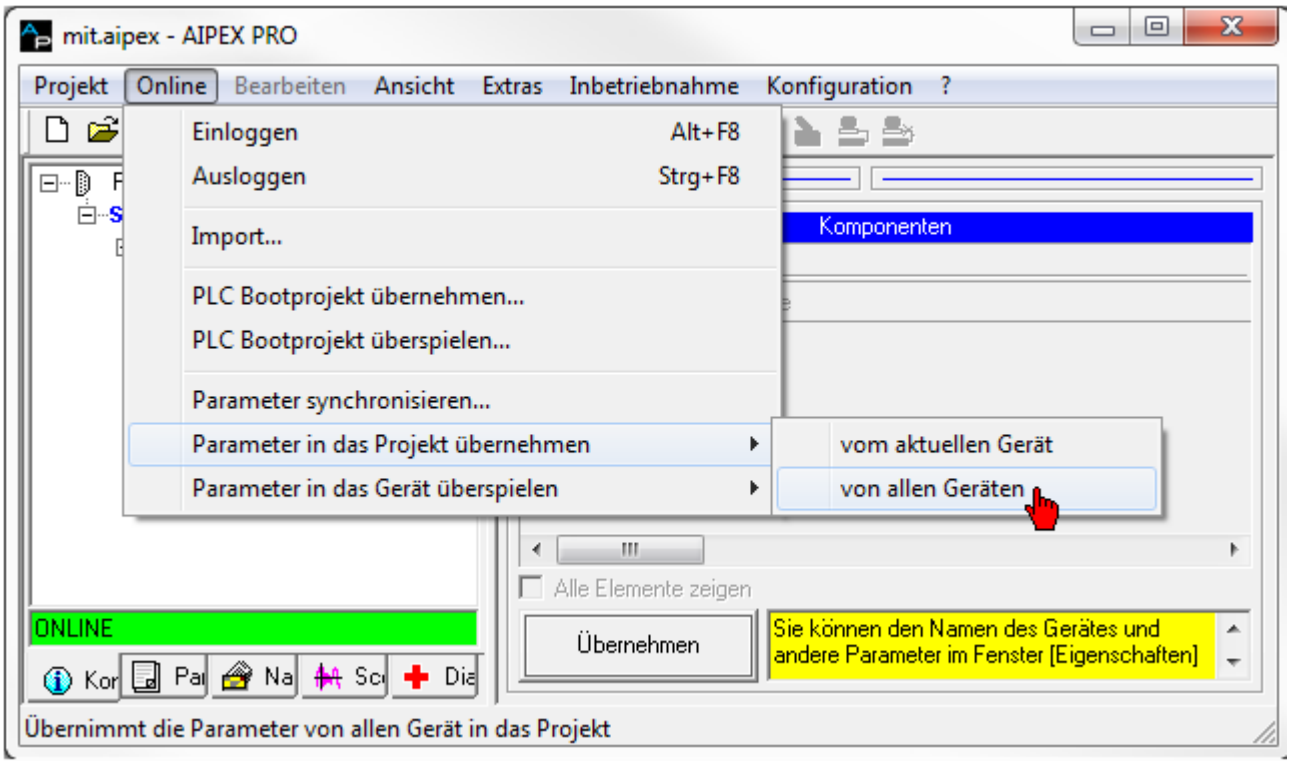


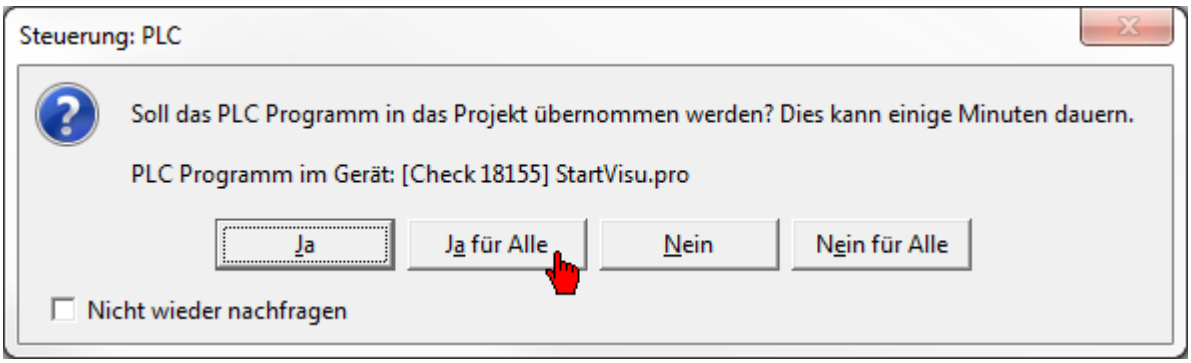
Übernehmen Sie die Gerätedaten in Ihr Projekt. Drücken Sie **'Komplett in das Projekt übernehmen'**



Die Ladezeit ist Abhängig von der eingesetzten Verbindung (Ethernet, SBUS, CAN ...) und der Anzahl der angeschlossenen Geräte.

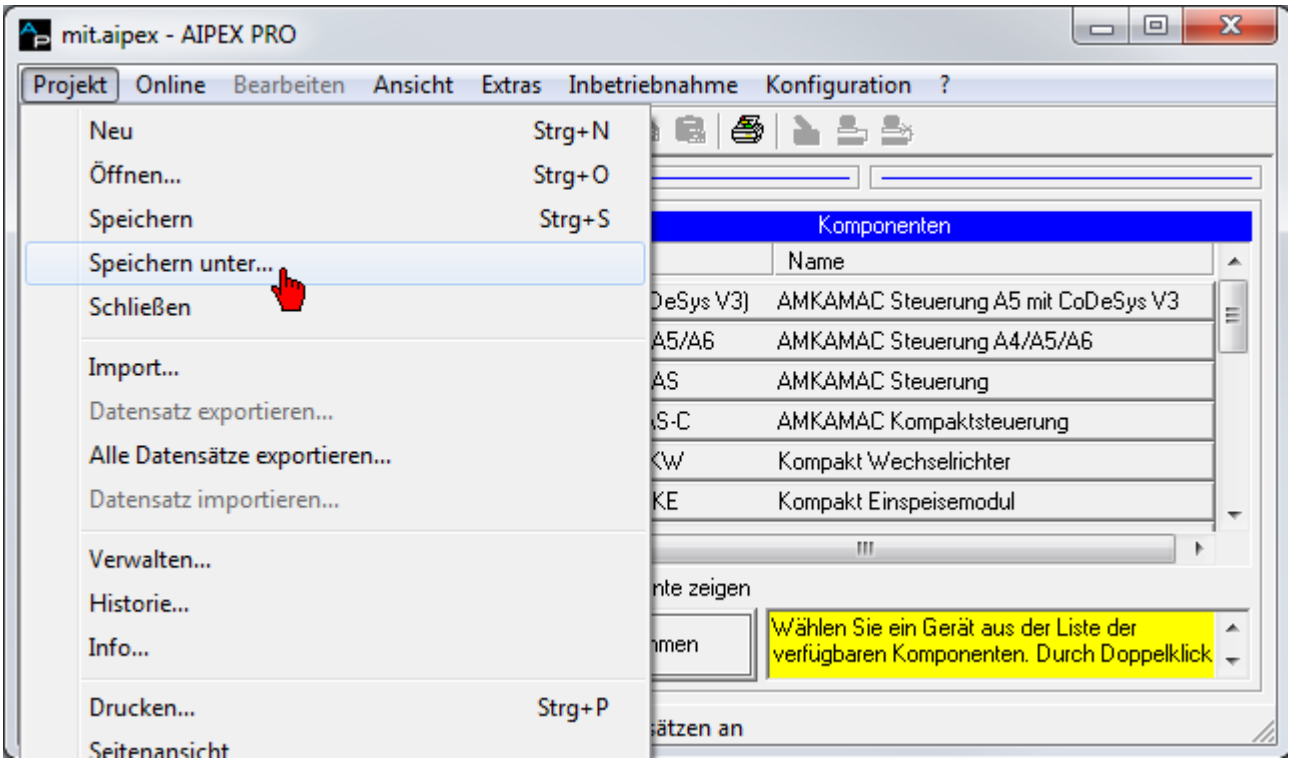
Das PLC Programm wird nur in das AIPEX PRO Projekt übernommen, wenn Sie die Funktion **'Online'** → **'Parameter in das Projekt übernehmen'** → **'von allen Geräten'** ausführen und bestätigen.





Einstellmöglichkeiten: Siehe 'Basiseinstellungen' auf Seite 112.

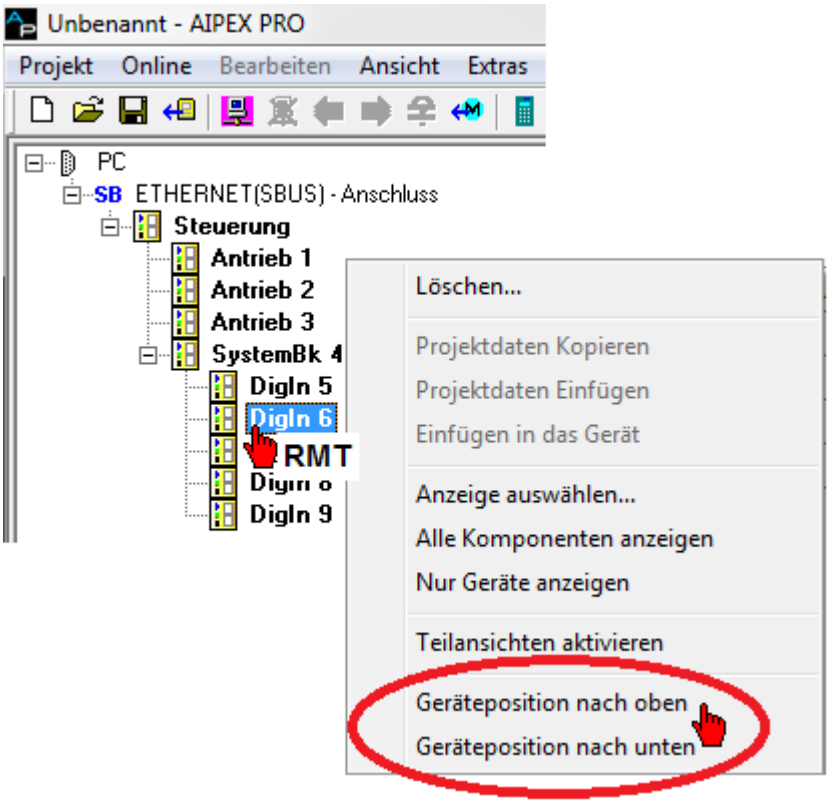
Speichern Sie das Projekt. Im Dialogfeld können Sie anschließend das Verzeichnis und den Dateinamen angeben.



3.7.6 Gerätepositon im EtherCAT Bus verschieben

Voraussetzung:
Aktivieren Sie die Positionsansicht (Menü 'Ansicht') im Gerätebaum.
Siehe 'Ansicht' auf Seite 102.

Klicken Sie mit der 'RMT' auf das Gerät das Sie verschieben wollen. Über die Menüpunkte 'Geräteposition nach oben' bzw. 'Geräteposition nach unten' können Sie die Position schrittweise verschieben.



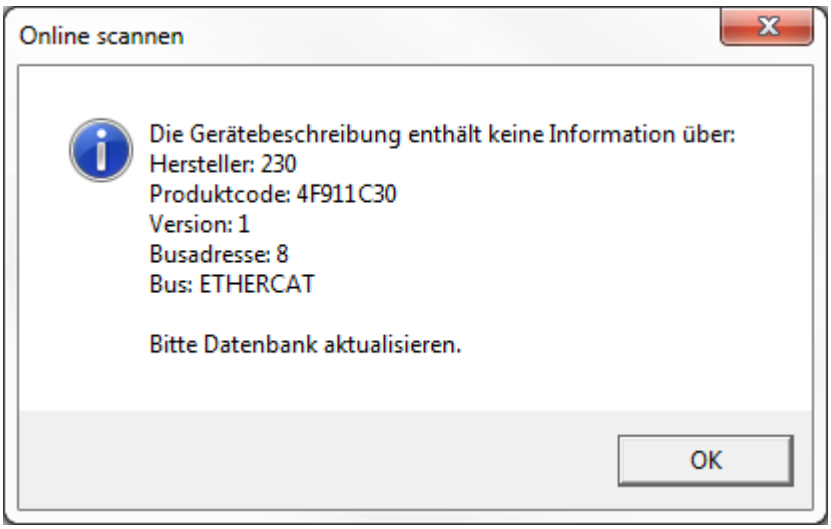
3.7.7 Importieren einer externen Gerätebeschreibung

Die Funktionalität von externen Geräten, wie z. B. EA Klemmen ist in den Gerätebeschreibungsdateien enthalten. Bei EtherCAT sind es XML Dateien bei CAN (ACC-Bus) EDS Dateien. Mit der Funktion **'Import externe Gerätebeschreibung'** können Gerätebeschreibungsdateien importiert werden, die standardmäßig nicht in AIPEX PRO integriert sind.

Für die Funktionsfähigkeit von zusätzlich importierten Geräten kann keine Garantie übernommen werden.

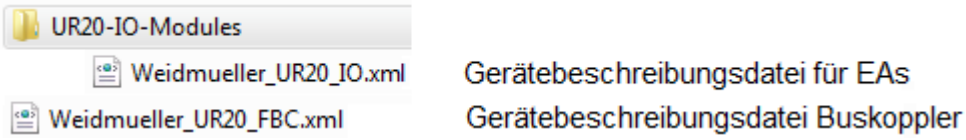
Nach dem importieren muss AIPEX PRO neu gestartet werden.

Erkennt AIPEX PRO beim 'Einloggen' ein externes Gerät ohne Gerätebeschreibungsdatei wird folgende Diagnosemeldung angezeigt:
Die Gerätebeschreibung enthält keine Informationen über:
.....
Bitte Datenbank aktualisieren

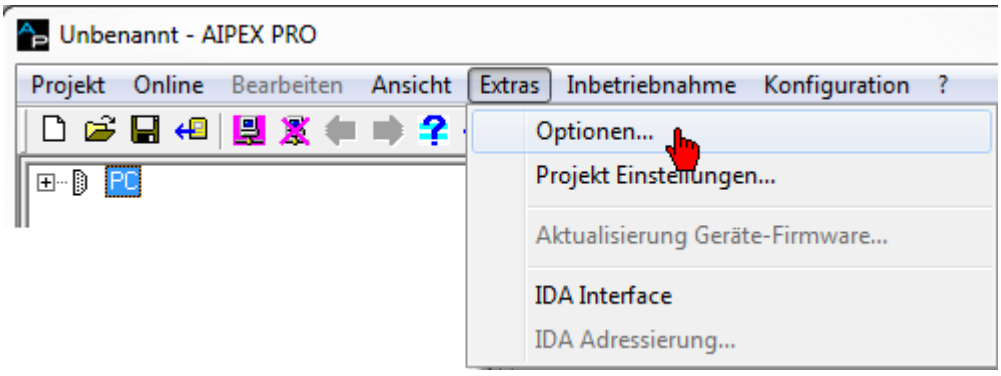


Gerätebeschreibungsdateien werden wie folgt importiert:
Speichern Sie die benötigten Gerätebeschreibungen vom Klemmenhersteller in den AIPEX PRO Ordner:
Windows 7: C:\Program Files (x86)\Common Files\AMK\EtherCAT
Windows XP: C:\Programme\Gemeinsame Dateien\AMK\EtherCAT

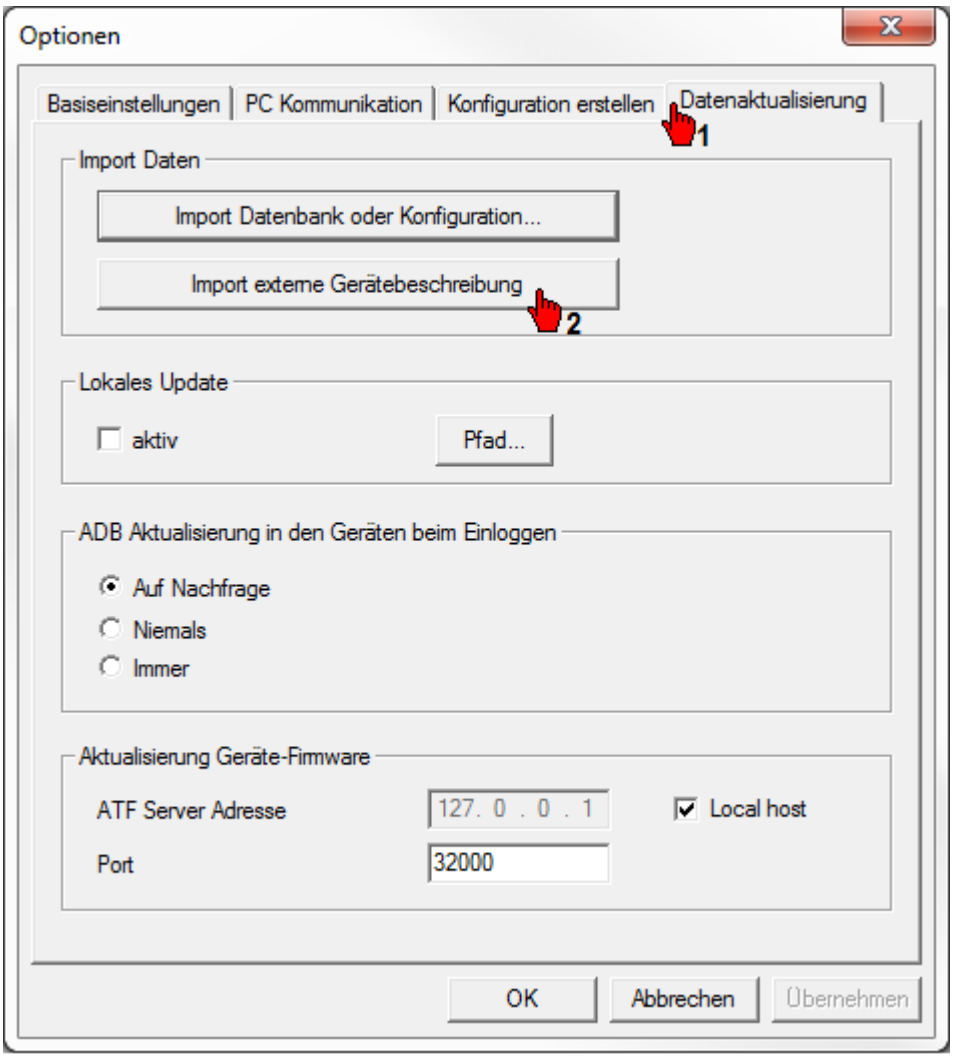
Beispiel Weidmüllerklemme:



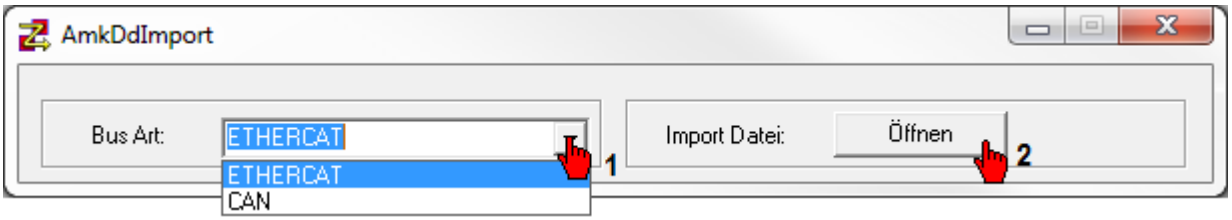
Öffnen Sie das Fenster 'Optionen'.



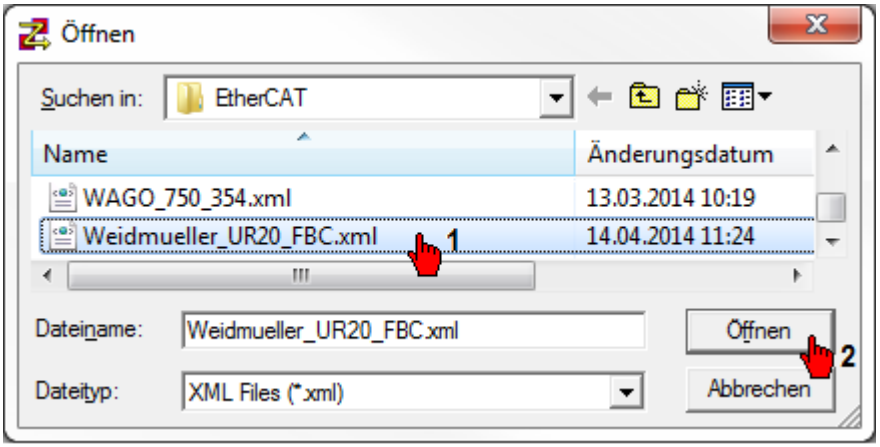
Wählen Sie den Reiter **'Datenaktualisierung'** an. Drücken Sie den Taster **'Import externe Gerätebeschreibung'**.



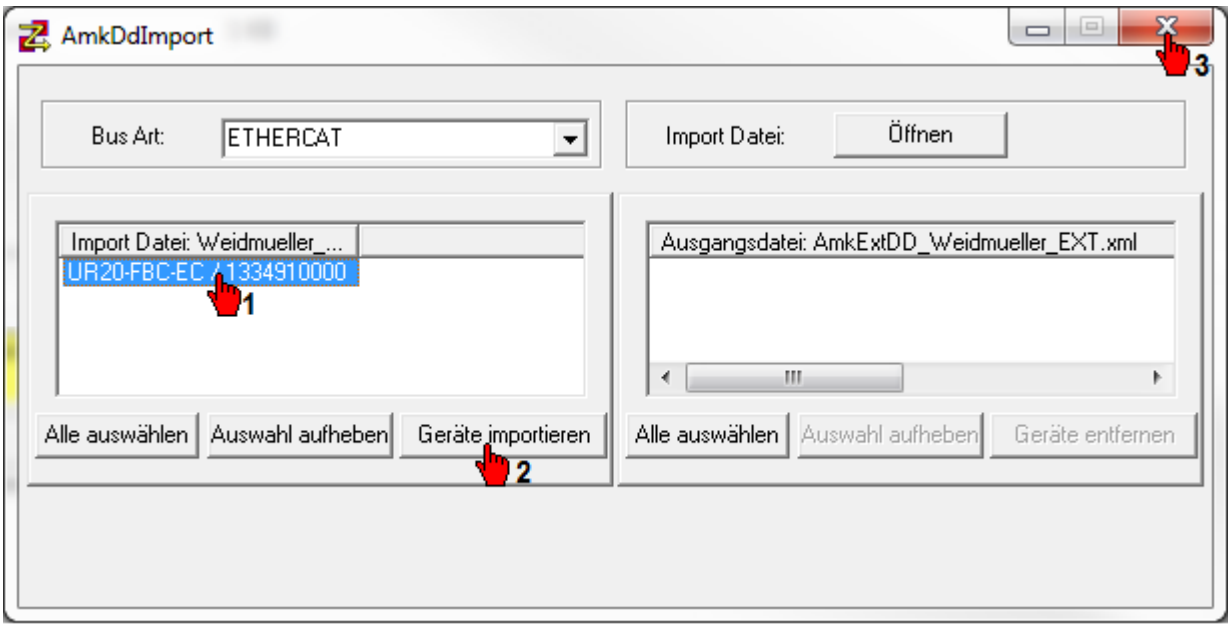
Wählen Sie den Feldbus an.



Wählen Sie die zu importierende Gerätebeschreibungsdatei des Buskopplers aus.



Importieren Sie die benötigte Datei mit dem Taster **'Geräte importieren'**.



Starten Sie AIPEX PRO neu.

3.7.8 Modular Device konfigurieren

Ein Modular Device ist ein Gerät das eine Teilnehmeradresse am EtherCAT Bus belegt.

Intern besteht das Modular Device aus mehreren Teilnehmern, die über einen internen Bus miteinander kommunizieren. Der EtherCAT Bus sieht bei einem Modular Device nur 1 Teilnehmeradresse.

Im nachfolgenden Beispiel wird beschrieben wie Sie ein 'Modular Device' (im Beispiel eine AIO-Klemme) konfigurieren.

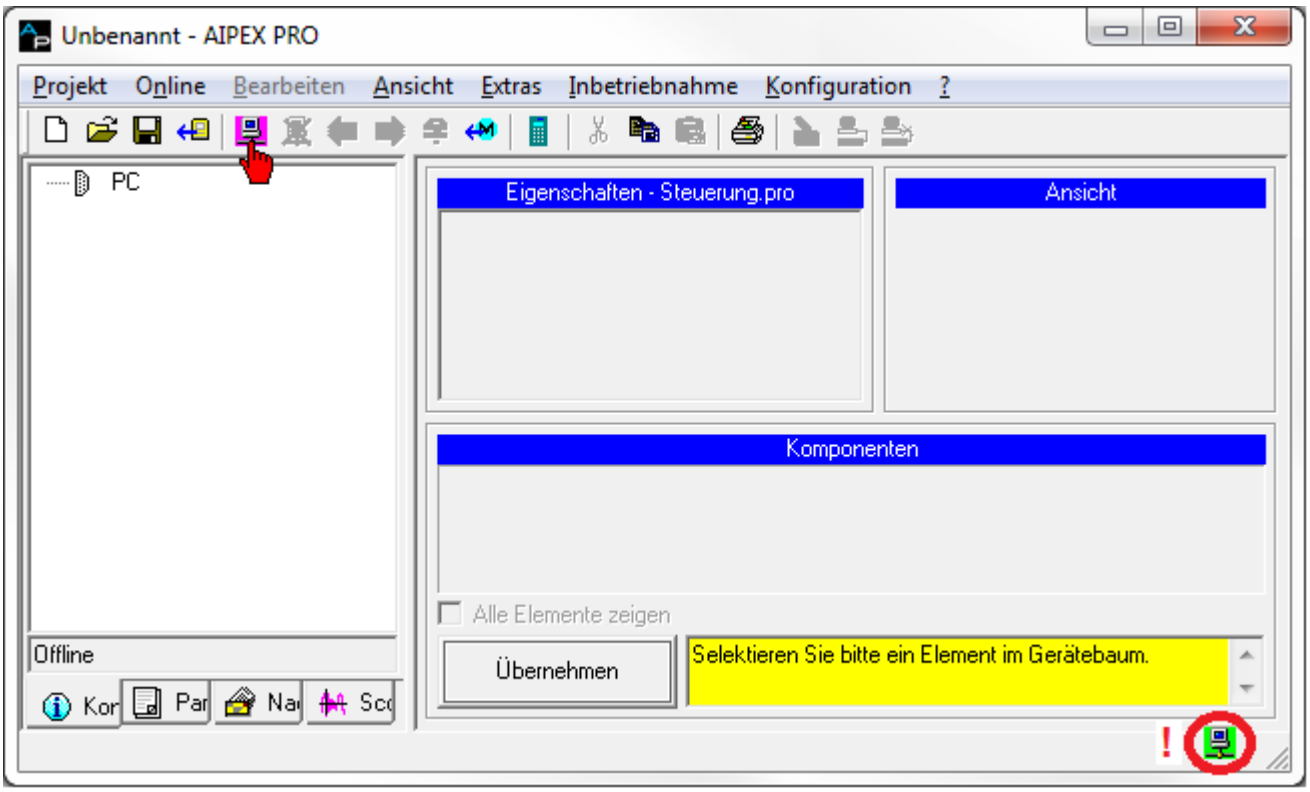
Voraussetzung:

Die Gerätebeschreibungsdatei des Buskoppler ist importiert.

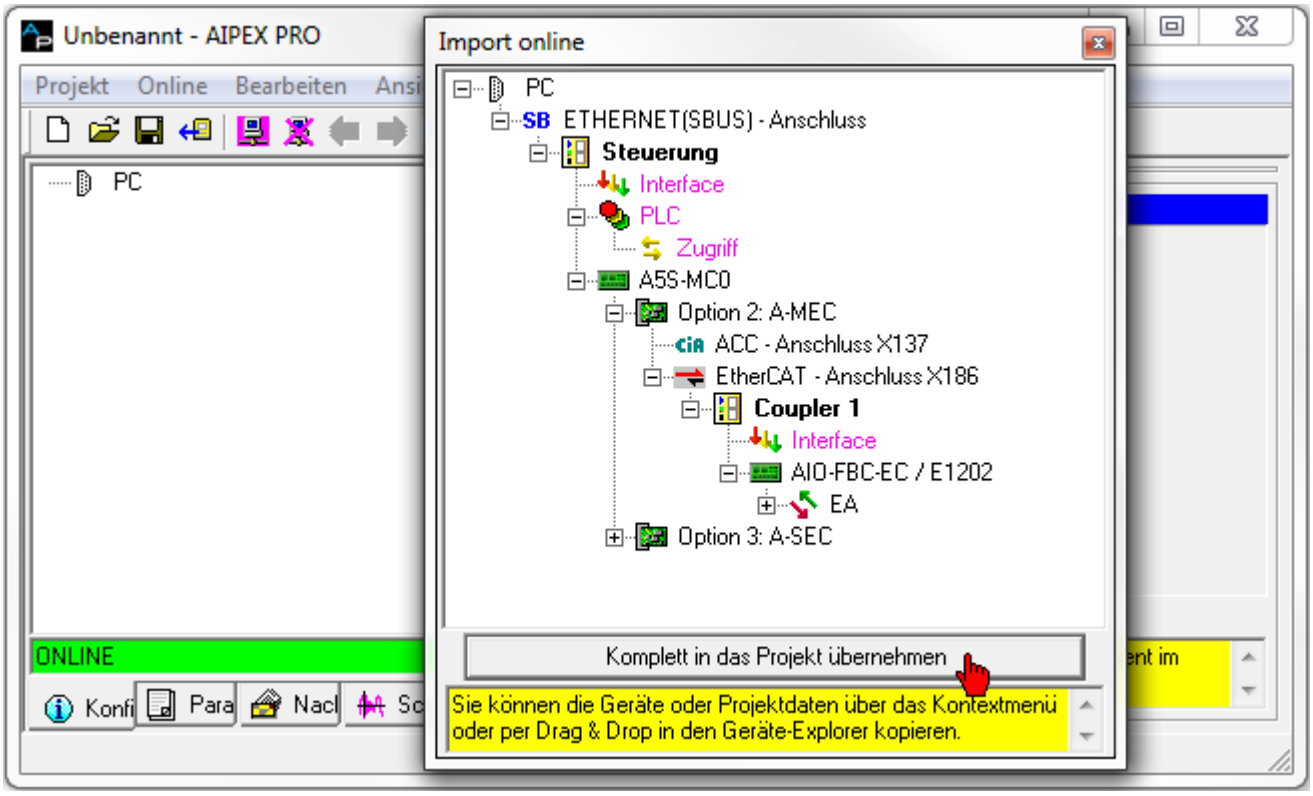
Siehe 'Datenaktualisierung' auf Seite 110. Funktion 'Import externe Gerätebeschreibung'

Starten Sie AIPEX PRO.

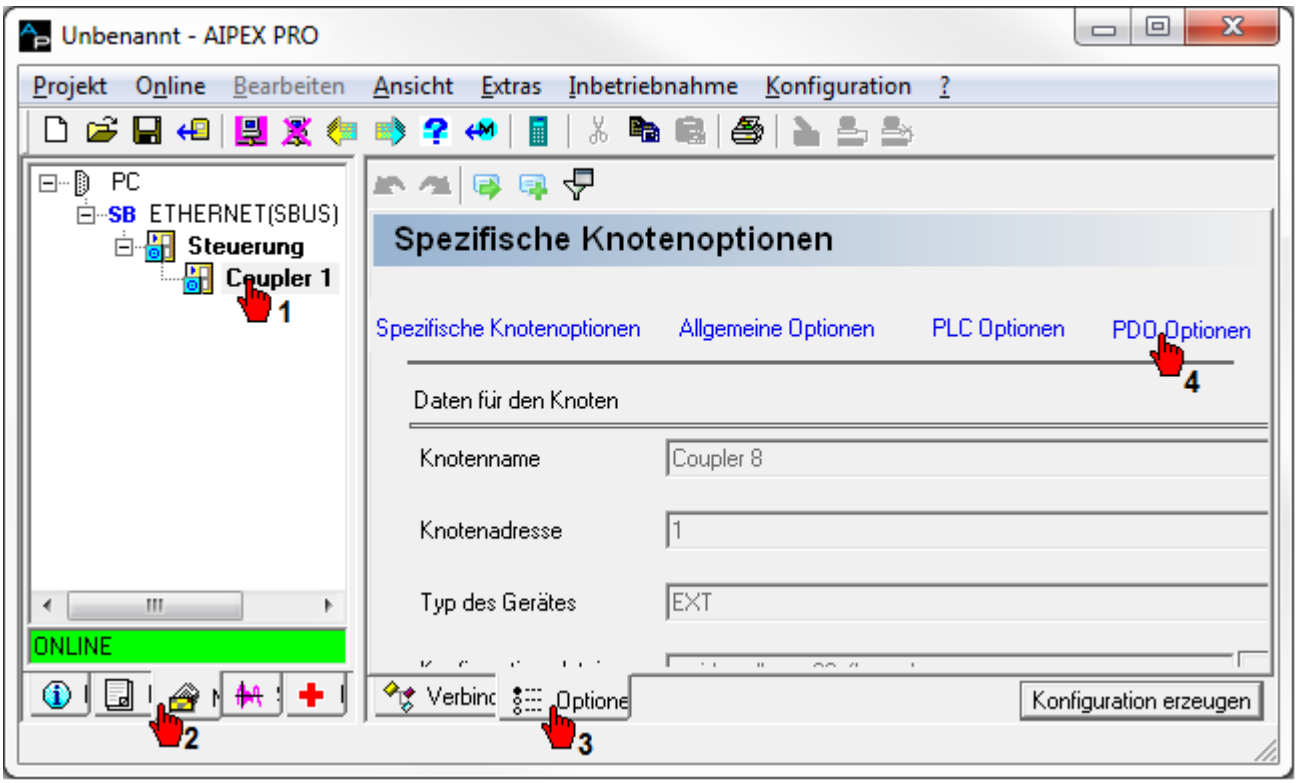
Drücken Sie **'Einloggen'**



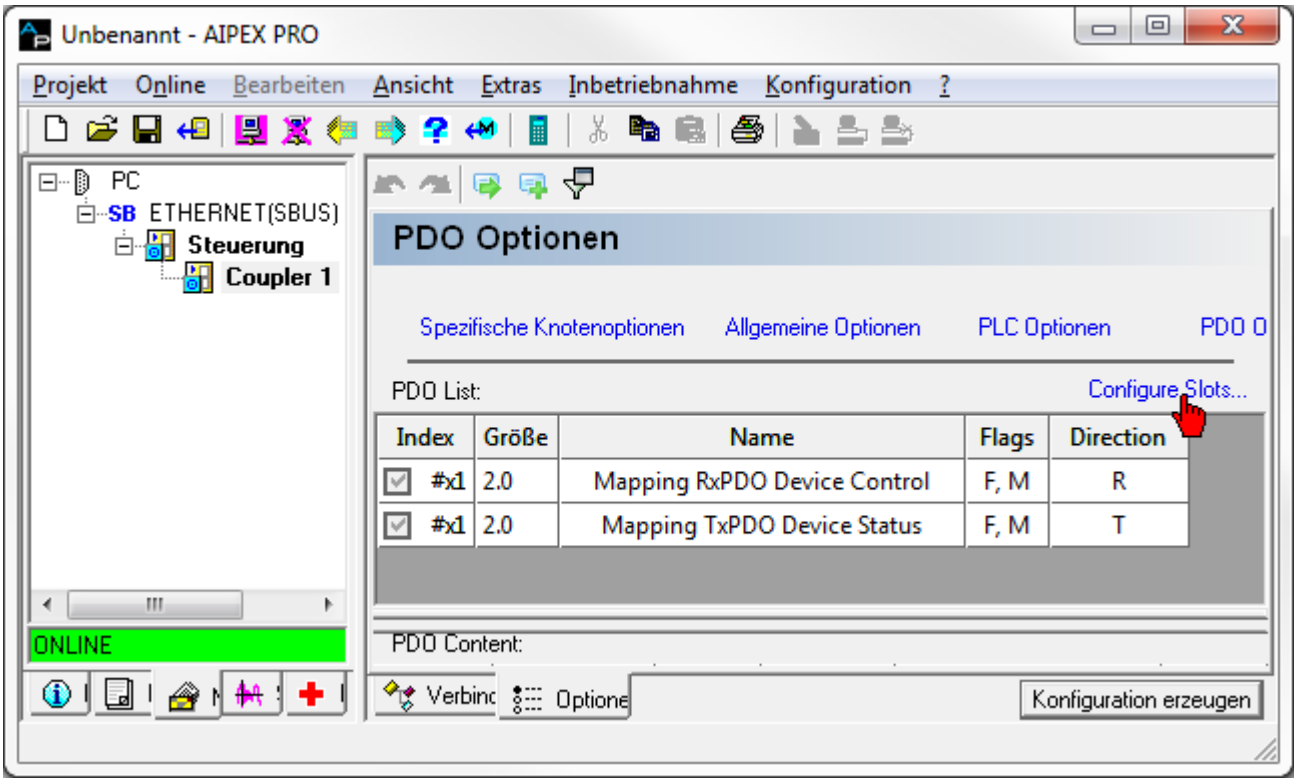
Übernehmen Sie die Gerätedaten in Ihr Projekt. Drücken Sie **'Komplett in das Projekt übernehmen'**



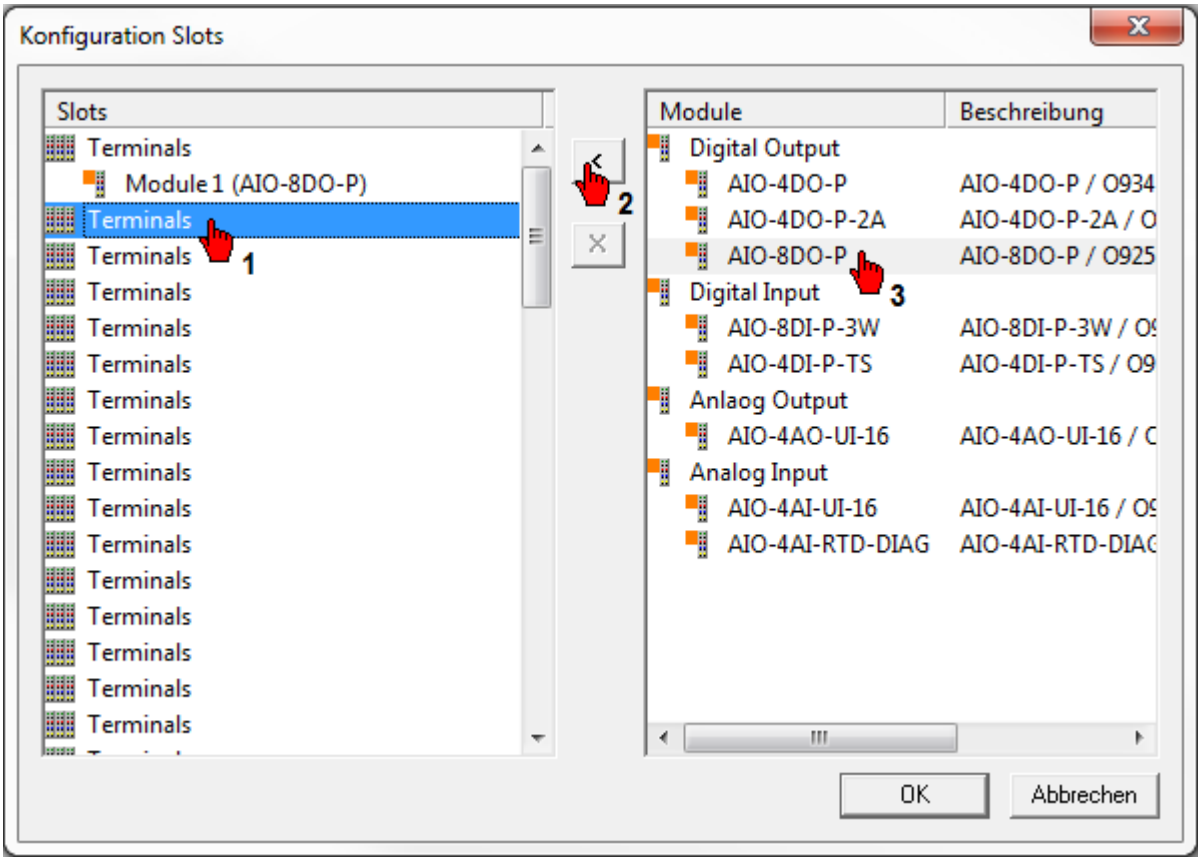
Wählen Sie die **'PDO Optionen'** unter dem Reiter **'Nachrichten'** → **'Optionen'** an.



Wählen Sie **'Configure Slots...'** an.



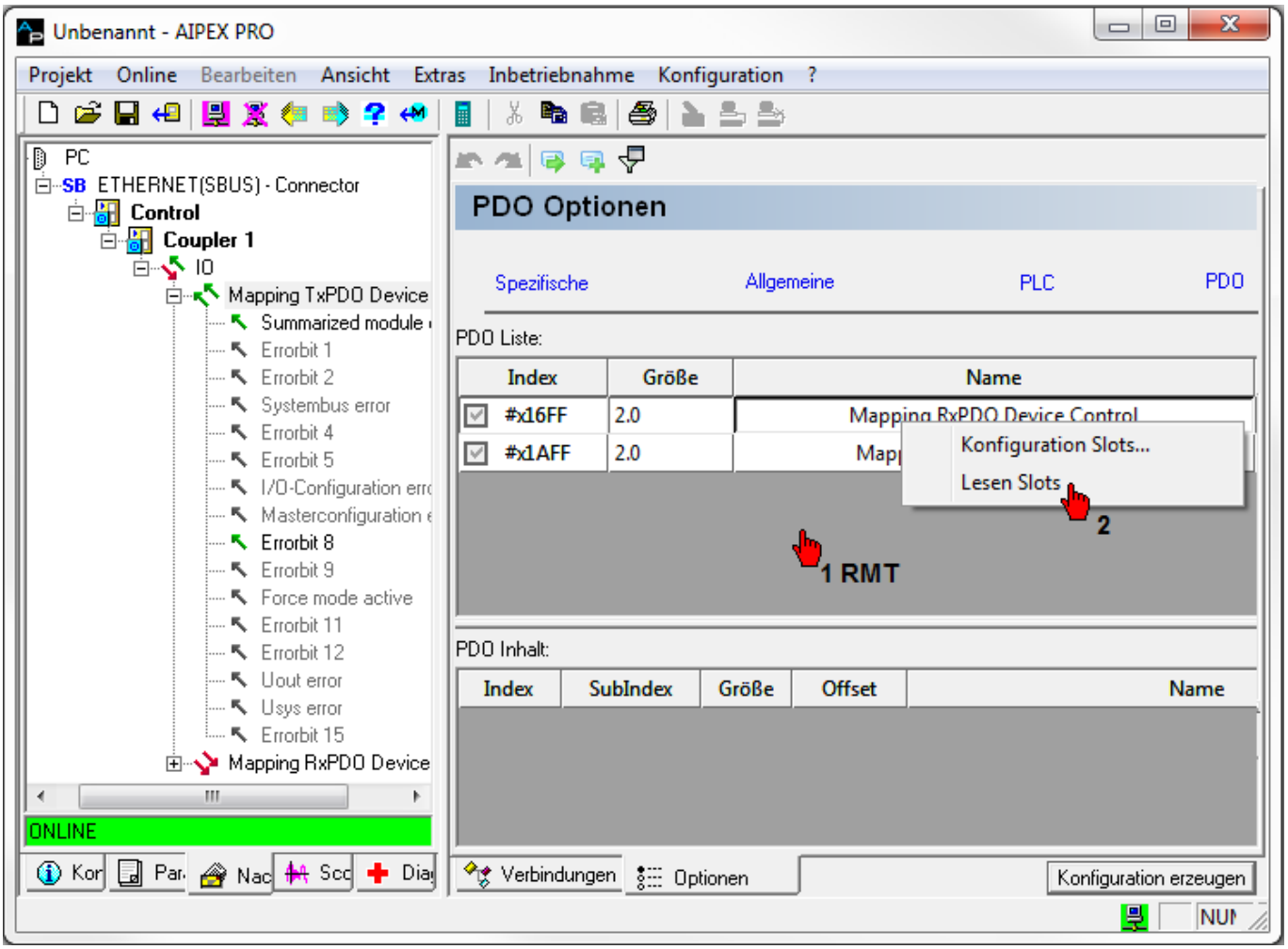
Das erste Terminal entspricht physikalisch dem ersten Klemmenblock nach dem Buskoppler (usw.)
Weisen Sie den Terminals die physikalisch vorhandenen Klemmenblöcke zu.



Beispiel I/O-Klemme mit CODESYS lesen und schreiben.
Siehe 'Externe I/O-Klemme (asynchron)' auf Seite 706.
Siehe 'Externe I/O-Klemme (zyklisch)' auf Seite 710.

Als Alternative zur manuellen Eingabe der in 'Modular Device' belegten Slots kann die aktuelle Konfiguration direkt aus dem angeschlossenem Gerät ausgelesen werden.

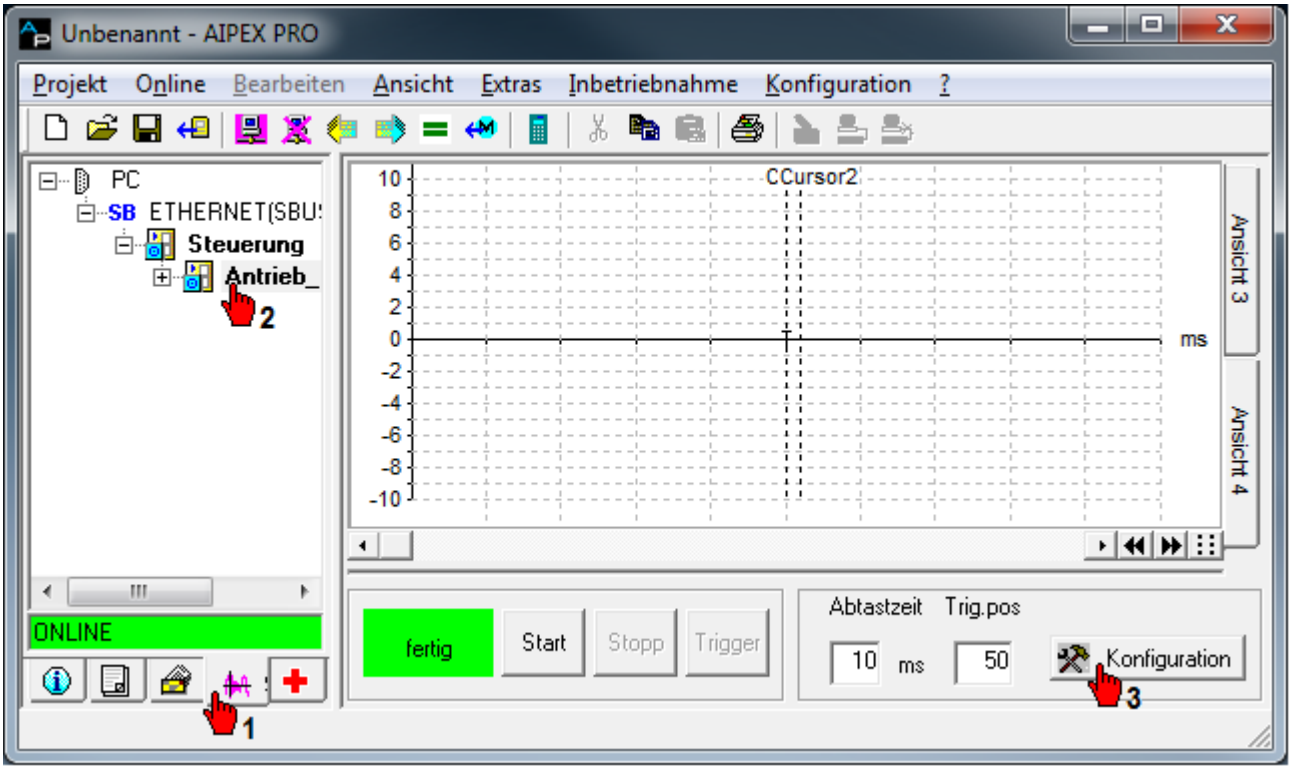
Voraussetzung:
Online Mode aktiv.



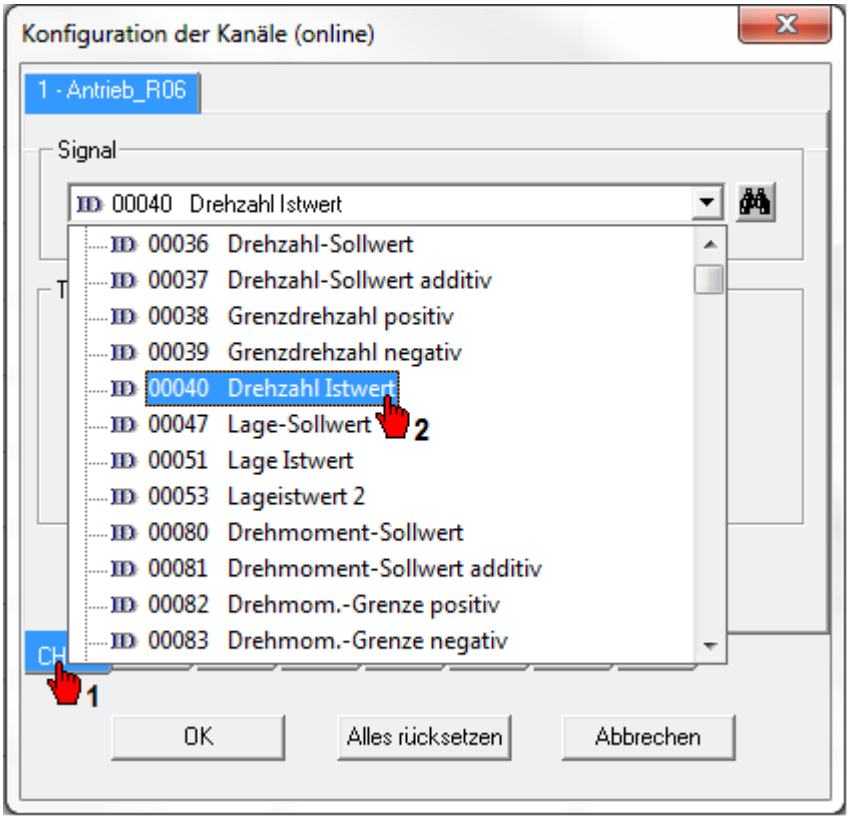
3.7.9 Oszilloskop konfigurieren

Das Beispiel zeigt, wie Sie mit dem AIPEX PRO Oszilloskop einen Drehzahl Istwert aufzeichnen.

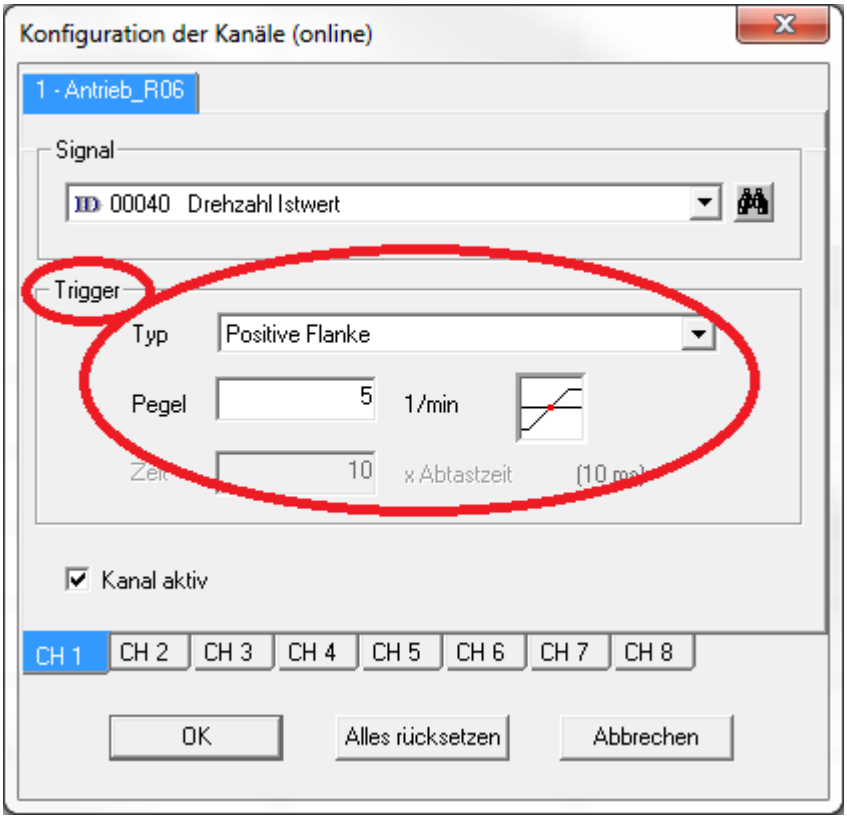
Öffnen Sie das 'Signal Konfigurationsfenster'.



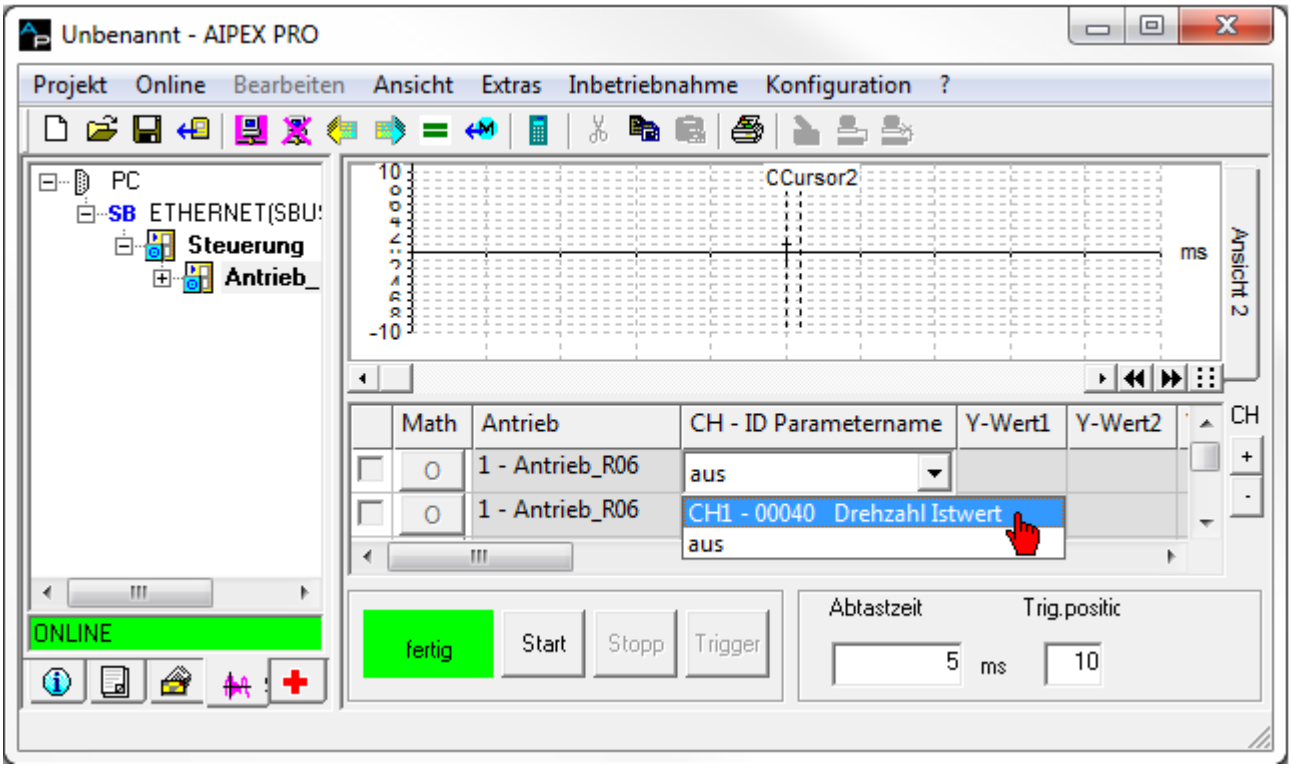
Tragen Sie im Kanal 1 (CH1) die ID40 'Drehzahl Istwert' ein.



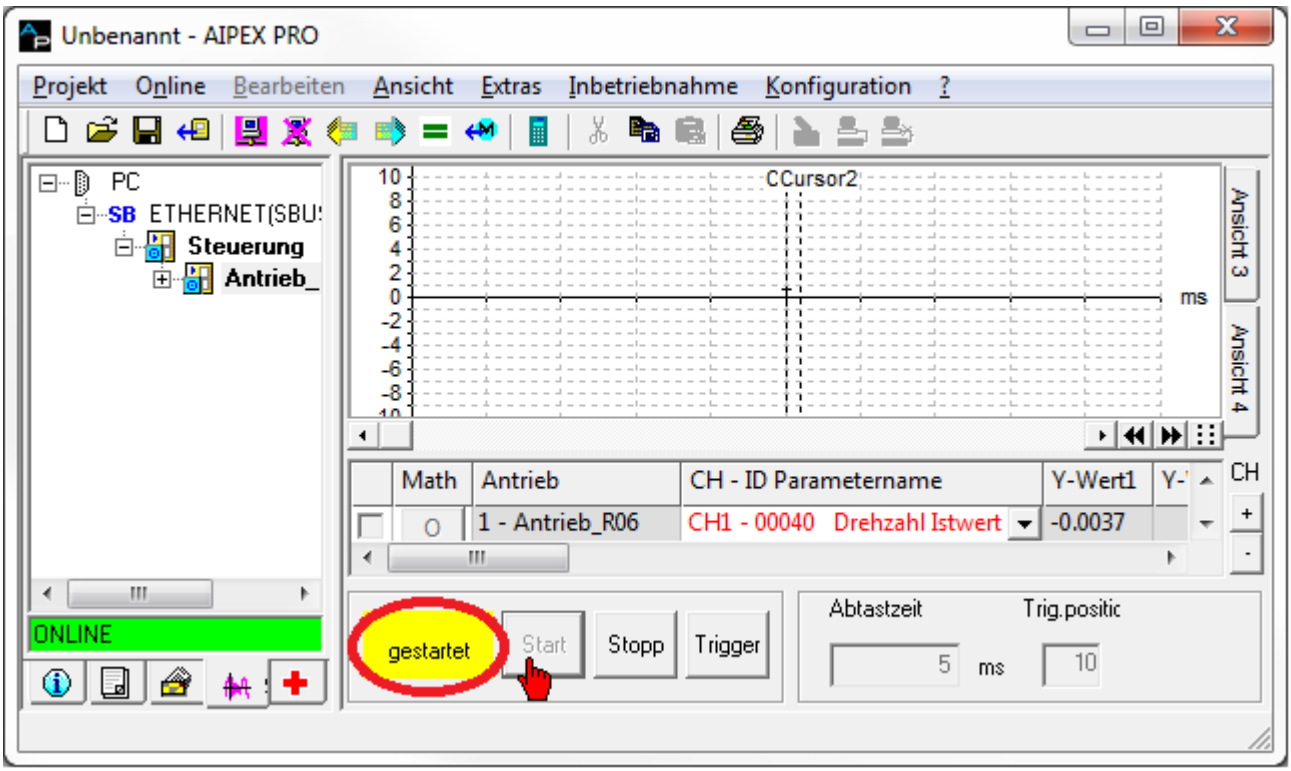
Triggereinstellungen:
Trigger Typ: Positive Flanke, Pegel 5 1/min. Die Messung wird gestartet sobald der Drehzahlistwert 5 1/min überschreitet.



Über das Feld 'CH – ID Parameternamen' werden die konfigurierten Signale ein- und ausgeblendet.
Passen Sie die 'Abtastzeit' und die 'Triggerposition' an Ihre Messung an.

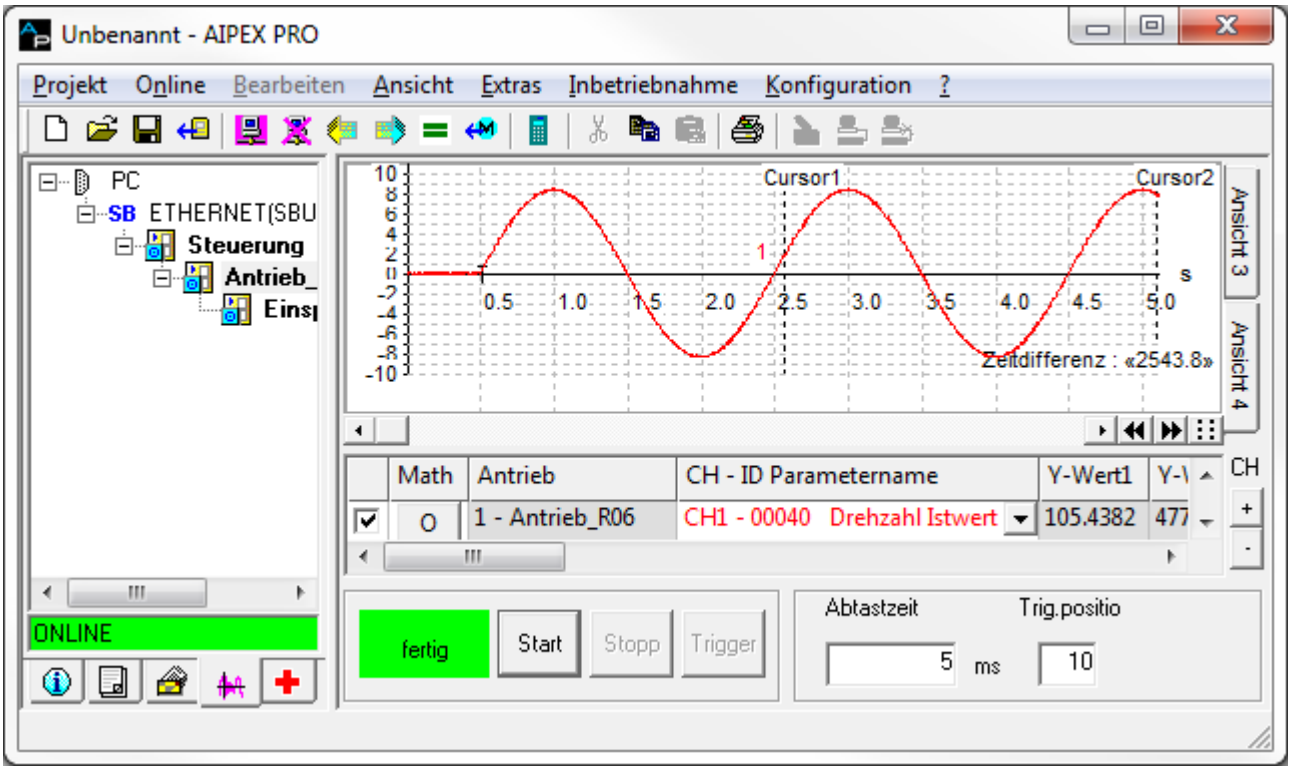


Klicken Sie auf **'Start'**. Der Status muss auf 'gestartet' wechseln.



Das Scope wartet auf das konfigurierte Triggersignal. Anschließend werden die aufgezeichneten Messdaten übertragen und grafisch dargestellt.

(Im Beispiel wird ein Sollwert durch den Testgenerator vorgegeben).



3.7.10 Testgenerator

⚠GEFAHR	
	Bewegungen der Motorwelle! Gefährliche Bewegungen entstehen durch unkontrollierte oder ungewollte Bewegungen der Motorwelle. **Gegenmaßnahmen:** • Entkoppeln Sie den Motor von der Last damit die Motorwelle frei drehen kann. • Legen Sie die maximal zulässige Prozessdrehzahl fest und parametrieren Sie ID113 entsprechend. (ID113 = maximale Prozessdrehzahl/1,25)

Der Testgenerator in AIPEX PRO unterstützt Sie bei der Erstinbetriebnahme und der Regleroptimierung. Die Inbetriebnahmefunktion ist ein Sollwertgenerator, der verschiedene Kurvenformen (Trapezsignal, Rechtecksignal, Dreiecksignal und Sinussignal) in den verschiedenen Betriebsarten (Momentsteuerung, Drehzahlregelung und Lageregelung) generieren kann.

Überprüfen Sie bei der Erstinbetriebnahme die Drehrichtung des Motors.

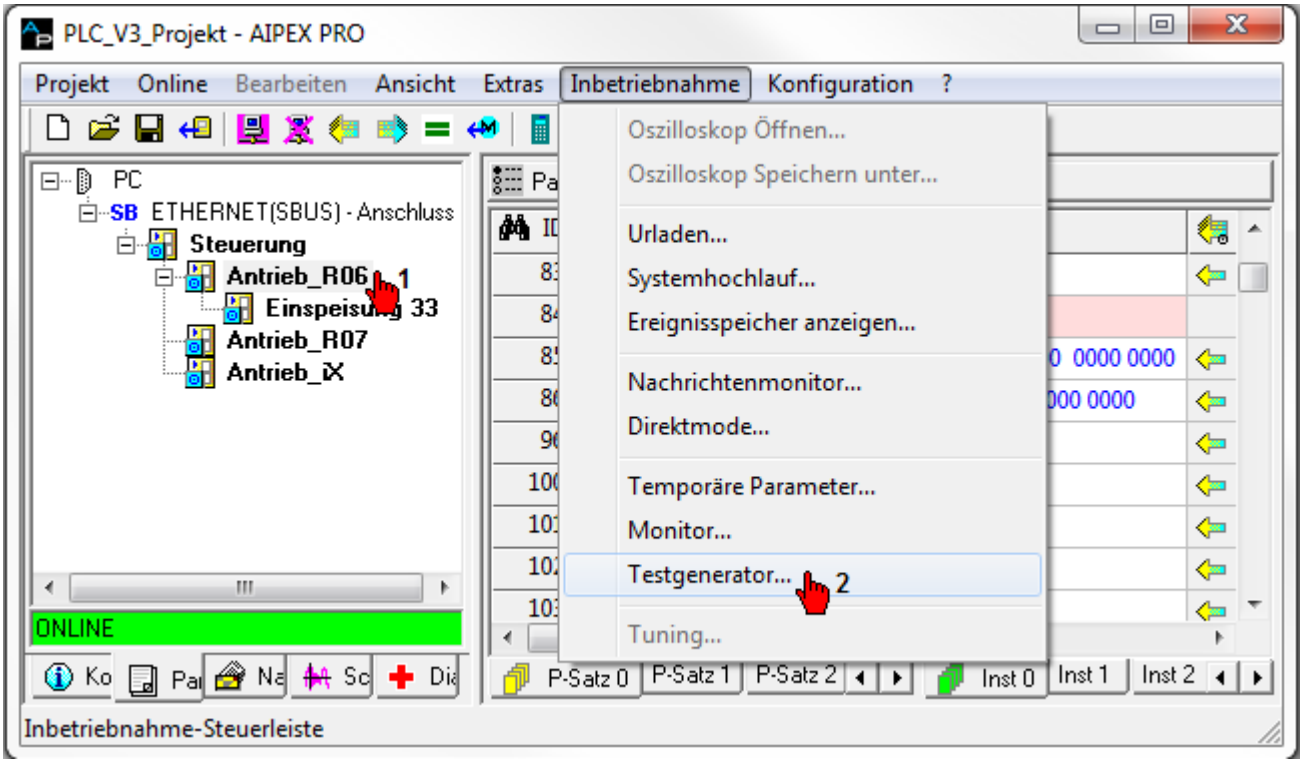
Das Beispiel zeigt, wie Sie einen positiven Drehzahlsollwert mit Beschleunigungs- und Bremsrampe vorgeben können. Mit der AIPEX PRO 'Monitor- und Oszilloskopfunktion' können Sie die Istwerte des Motors anzeigen und aufzeichnen.

Die Drehrichtung des Motors kann in ID32773 Bit 16 = 1 negiert werden.

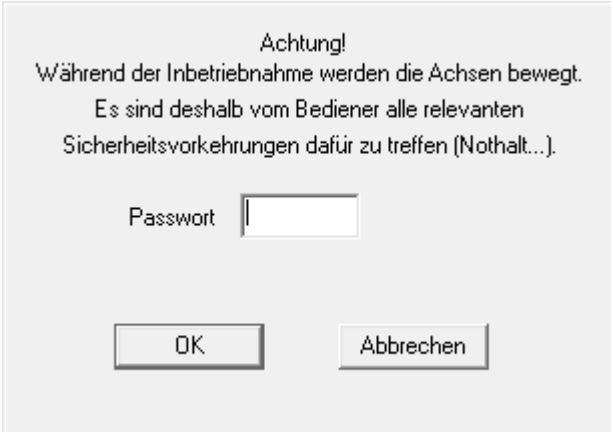
Starten des Testgenerators

Markieren Sie im Gerätebaum den Antrieb, für den Sie mit dem Testgenerator einen Sollwert vorgeben möchten.

Klicken Sie auf das Menü 'Inbetriebnahme' und anschließend auf die Funktion 'Testgenerator...'



Der Testgenerator wird mit dem AMK Servicepasswort (500591) freigegeben.

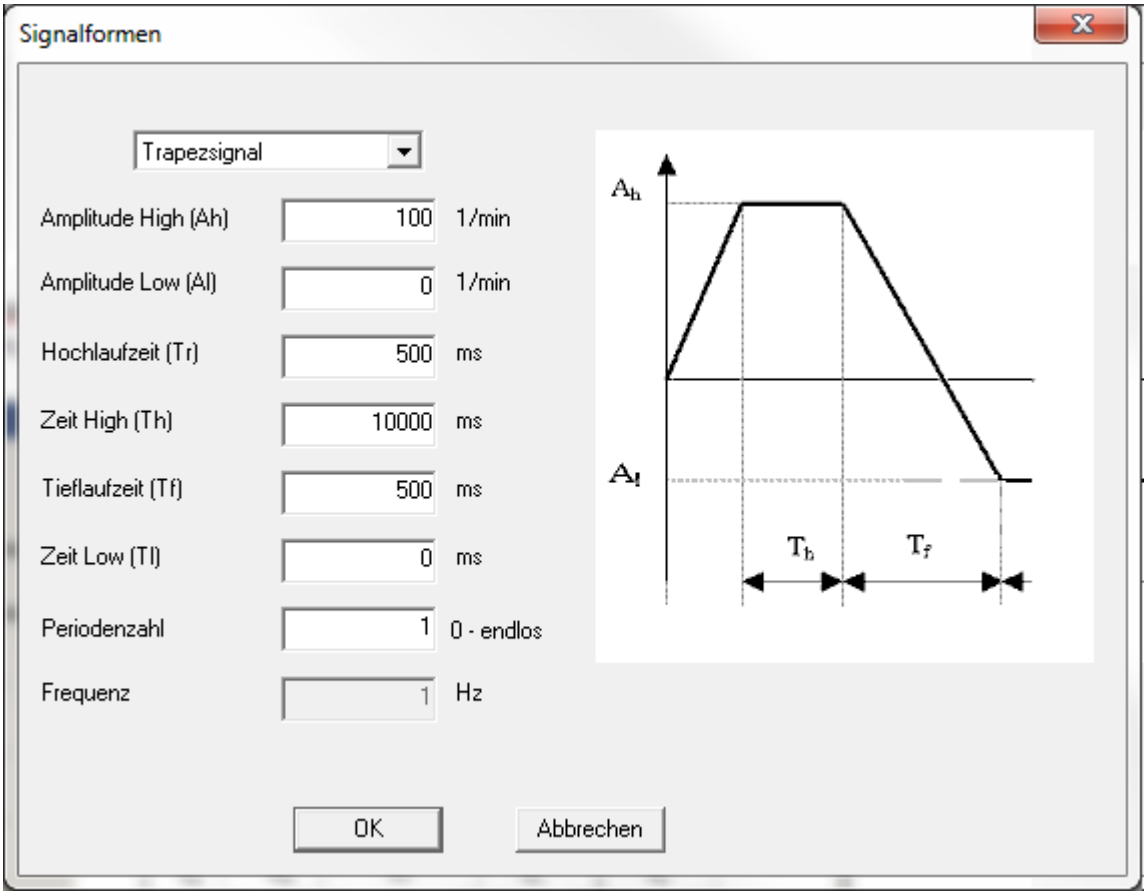


Signal Konfigurationsfenster 'Signalformen' öffnen.

Die Reglerfreigabe muss zur Konfiguration deaktiviert sein.

Signalkonfiguration am Beispiel 'Trapezsignal'.

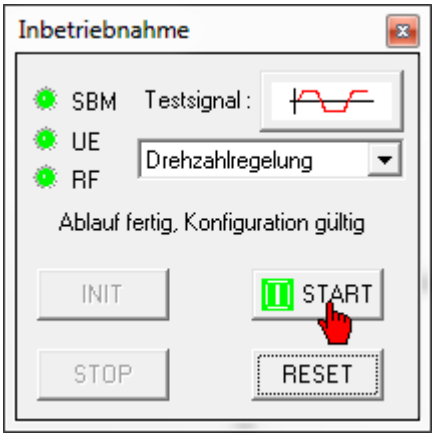
Mit dem 'Trapezsignal' können Sie einen positiven und negativen Drehzahlsollwert mit Beschleunigungs- und Bremsrampe vorgeben. Im Beispiel ist eine positive Bewegung konfiguriert.



Aktivieren Sie jetzt die Motorregelung.

Die LED's zeigen den Status von SBM, UE und RF des Antriebsystems an. Grün bedeutet aktiv.

Der konfigurierte Bewegungsablauf wird mit der Taste **'START'** gestartet.

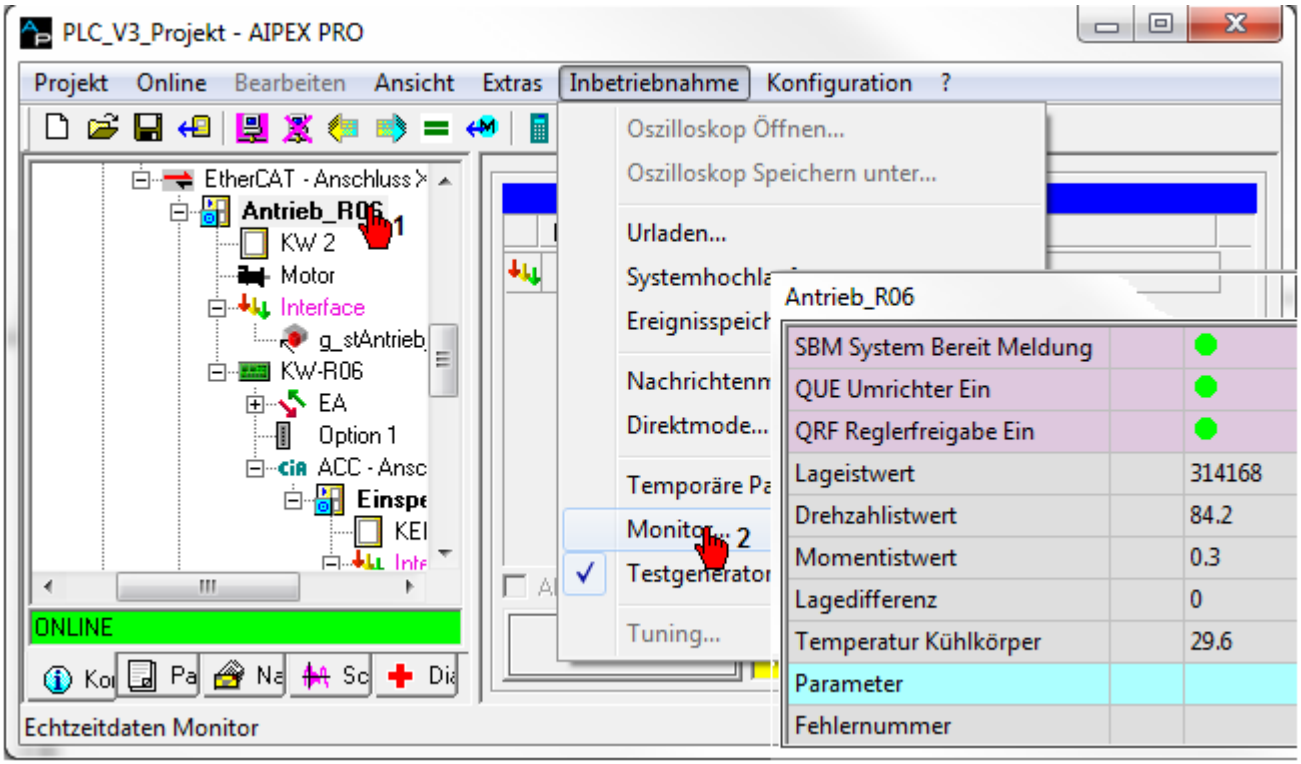


Die Bewegung kann mit den Tasten **'STOP'** angehalten und mit **'START'** weitergeführt werden.

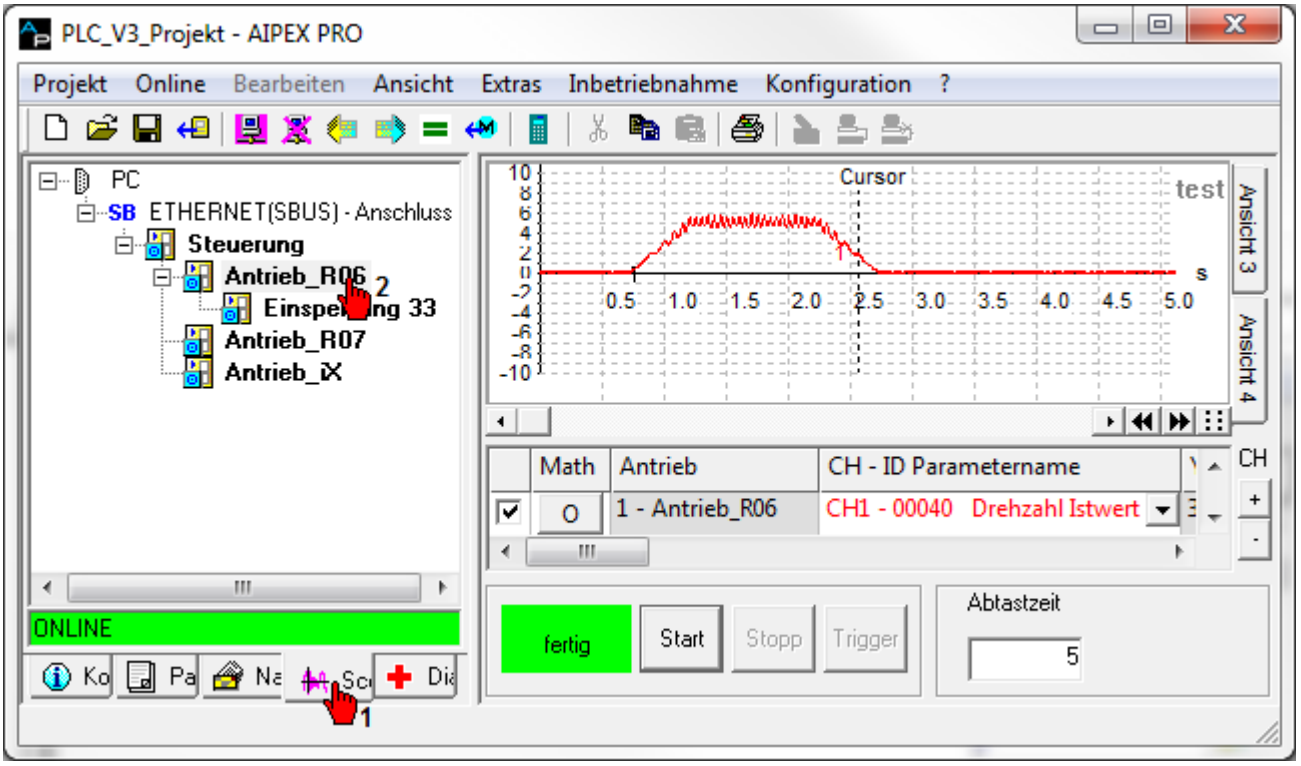
Ein Klick auf den Taster **'STOP'** und anschließend auf den Taster **'RESET'** bricht den kompletten Bewegungsablauf ab.

Istwertanzeigen

Mit der AIPEX PRO 'Monitorfunktion' können Sie Status und Istwerte anzeigen.
Prüfen Sie, ob der Lageistwert ansteigt und die Drehrichtung positiv ist, wenn Sie positive Sollwerte vorgeben.
Eine positive Drehrichtung bedeutet, dass sich die Motorwelle im Uhrzeigersinn dreht, wenn Sie auf die Motorwelle blicken.



Mit dem AIPEX PRO Oszilloskop kann die Bewegung aufgezeichnet und ausgewertet werden.

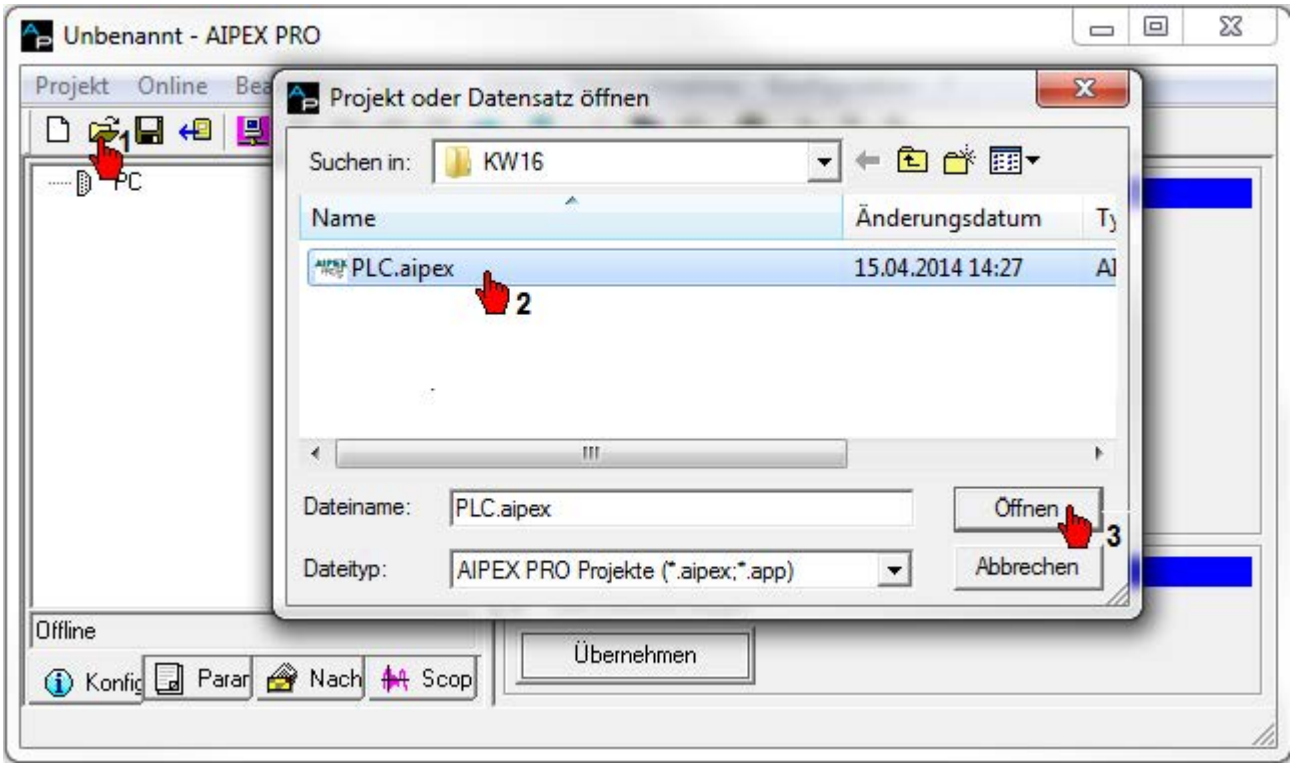


Konfigurationshilfe Oszilloskop:
Siehe 'Scope - Anleitung' auf Seite 92.

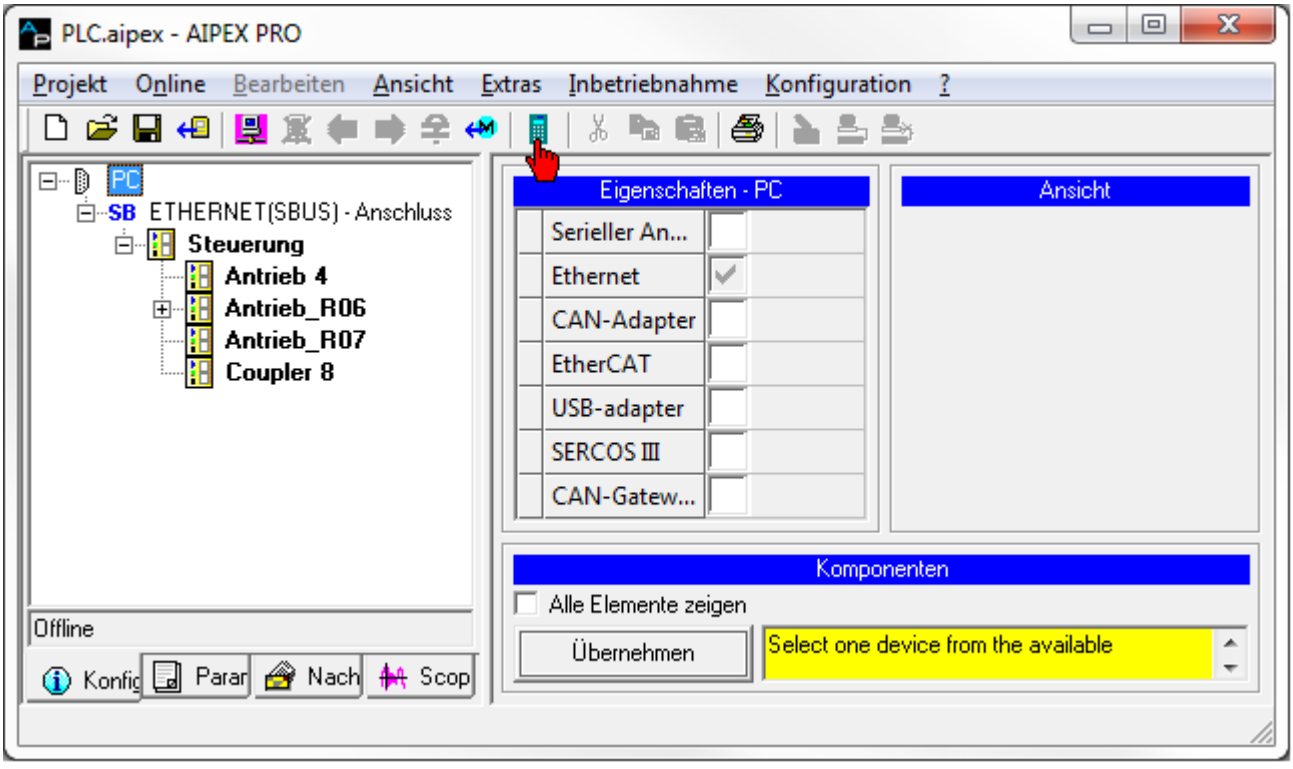
3.7.11 Überspielen eines Offline Projekts in ein Gerät

Im nachfolgenden Beispiel wird beschrieben, wie Sie ein Parametersatz öffnen und den Datensatz in ein AMK Gerät übertragen.

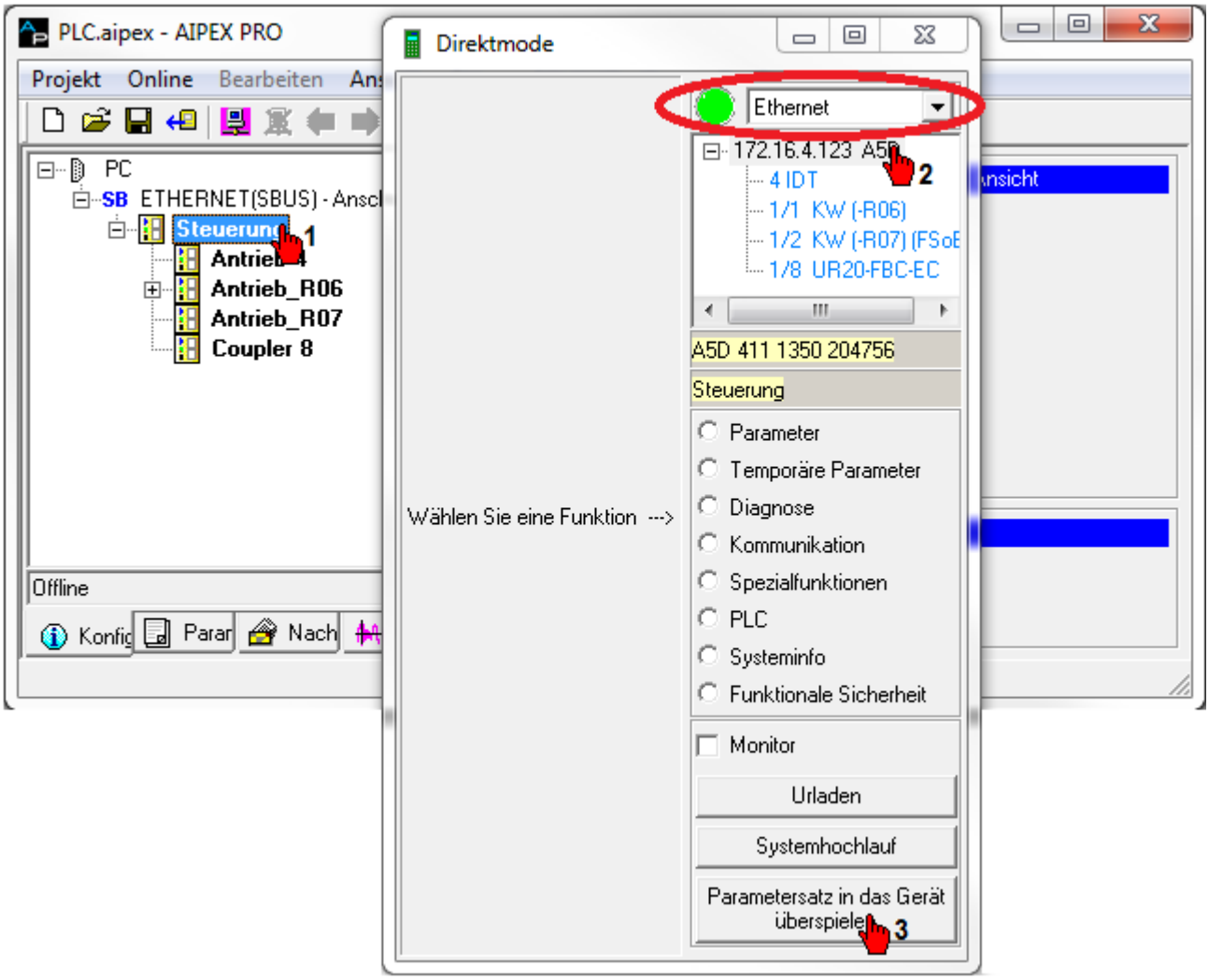
Klicken Sie auf den Taster **'Öffnen'** um das Dialogfeld 'Projekt oder Datensatz öffnen' zu öffnen. Wählen Sie den gewünschten Datensatz an.



Öffnen Sie den 'Direktmode'.



Der Datensatz des im Gerätebaum markierten Gerätes wird auf das markierte Gerät im Direktmode übertragen.
Dazu muss zwischen dem PC und dem Gerät eine physikalische Verbindung (Direktverbindung oder über Feldbus) bestehen.
Wählen Sie die verwendete Schnittstelle an. Eine aktive Schnittstelle erkennen Sie am Status grün.
Mit **'Parametersatz in das Gerät überspielen'** wird der Transfer gestartet.



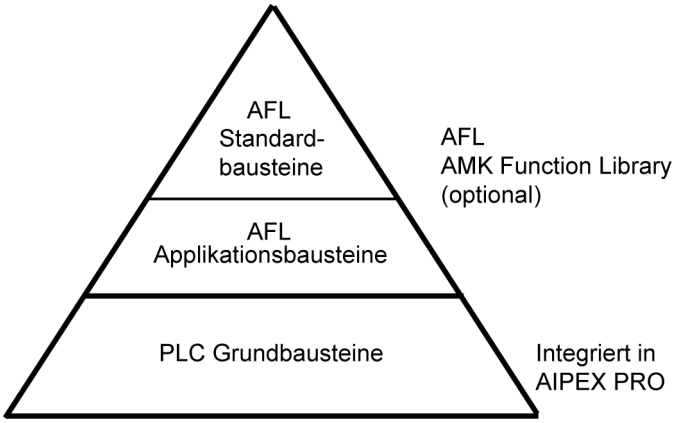
4 AFL (AMK Function Libraries)

4.1 AmkAfl Standardbausteine

4.1.1 Einführung AFL Standardbausteine für CODESYS V3

Mit AFL Standardbausteinen wird die Funktionalität der AMK Geräte in ein PLC Projekt eingebunden. Der Programmieraufwand durch die Verwendung der AFL Standardbausteine ist deutlich reduziert. Die AFL Standardbausteine bestehen aus einer Vielzahl von Funktionsbausteinen aus verschiedenen AMK Bibliotheken.

4.1.1.1 Übersicht der AMK Bibliotheken



Die AFL Standardbausteine bestehen aus AFL Applikationsbausteinen und sind Bestandteil der AFL Function Library.

Siehe 'AMK AFL Applikationsbausteine' auf Seite 366.

Die AFL Applikationsbausteine bestehen aus den PLC Grundbausteinen. (Siehe Dokument Softwarebeschreibung AmkLibraries (Teile-Nr.205210))

Beispiel anhand dem Standardbaustein 'STANDARD_AXIS':

AFL Standardbausteine

Der Standardbaustein 'STANDARD_AXIS' (FB) besteht aus den AFL Applikationsbausteinen 'MANUAL_JOG_VAJ', 'POSITION_ABSOLUT_VAJ', 'MANUAL_VELOCITY', 'POSITION_HOMING_FIXED_STOP' und weiteren aus der AMK AFL Library (AmkAfl.lib). Der Quellcode des Standardbausteins ist vom Anwender editierbar.

AFL Applikationsbausteine

AMK stellt mit den AFL Applikationsbausteinen komplexere Funktionen zur Verfügung. Sie bestehen aus einzelnen PLC Grundbausteinen. Die Funktionalität der AFL Applikationsbausteine ist fest definiert und kann vom Anwender nicht editiert werden.

Beispiel: MANUAL_JOG_VAJ (FB) aus der AMK AFL Library (AmkAfl.lib)

Der AFL Applikationsbaustein 'MANUAL_JOG_VAJ' realisiert den Tippbetrieb (Plus / Minus) in Lageregelung. Dabei kann der Anwender neben Position, Geschwindigkeit und Beschleunigung auch den Ruck vorgeben.

PLC Grundbausteinen

Der AFL Applikationsbaustein 'MANUAL_JOG_VAJ' besteht aus den PLC Grundbausteinen 'VGEN_AJ' und 'RATIO_INC' aus der Bibliothek AmkBase.lib. Die Funktionalität der PLC Grundbausteine ist fest definiert und kann vom Anwender nicht editiert werden.

Übersicht AFL Function Librarys

Die AMK AFL Library muss zusätzlich zu AIPEX PRO installiert werden. Die Version der AFL Function Library ist abhängig von der verwendeten CODESYS Version .

Versionsübersicht

AIPEX PRO + integrierte CODESYS V3 Version + kompatible AFL Version

AIPEX PRO Version	CODESYS V3 Version	CODESYS Profile	Kompatible AFL Version
3.04	3.5.10.4	CODESYS V3.5 **SP10** Patch 4 AIPEX PRO	AFL V4 Version 3.5.5.0 2015/41 (Teile-Nr. 206004)
	3.5.5.5	CODESYS V3.5 **SP5** Patch 5 AIPEX PRO	
	3.5.3.6	CODESYS V3.5 **SP3** Patch 6 AIPEX PRO	AFL V4 Version 3.5.3.0 2014/06 (Teile-Nr. 204786)
3.03	3.5.5.5	CODESYS V3.5 **SP5** Patch 5 AIPEX PRO	AFL V4 Version 3.5.5.0 2015/41 (Teile-Nr. 206004)
3.02	3.5.3.6	CODESYS V3.5 **SP3** Patch 6 AIPEX PRO	AFL V4 Version 3.5.3.0 2014/06 (Teile-Nr. 204786)
3.01			
3.00			

Steuerung	Firmware Version	CODESYS V3 Profil
iSA / A4	≥ 4.22	CODESYS V3.5 SP10 Patch 4 AIPEX PRO
iSA / A4	≤ 4.21	CODESYS V3.5 SP5 Patch 5 AIPEX PRO
A5 / A6	alle	CODESYS V3.5 SP10 Patch 4 AIPEX PRO (empfohlen) (mit Einschränkungen[1)])
		CODESYS V3.5 SP5 Patch 5 AIPEX PRO (mit Einschränkungen[1)])
		CODESYS V3.5 SP3 Patch 6 AIPEX PRO

1) Neue Features dieser CODESYS Version, die das Laufzeitsystem betreffen, werden nicht unterstützt.

Entsprechende CODESYS Version muss bei der Installation von AIPEX PRO mit installiert werden.

4.1.1.2 Verwendung der AFL Standardbausteine

Die AFL Standardbausteine ermöglichen dem Anwender den einfachen Zugriff auf alle Standard Funktionen der Steuerungen, Einspeisungen und Achsen (Antriebe).

Vordefinierte Strukturen ermöglichen den Zugriff auf Soll- und Istwerte, Steuer- und Statusbits, Parameter und Diagnosemeldungen.

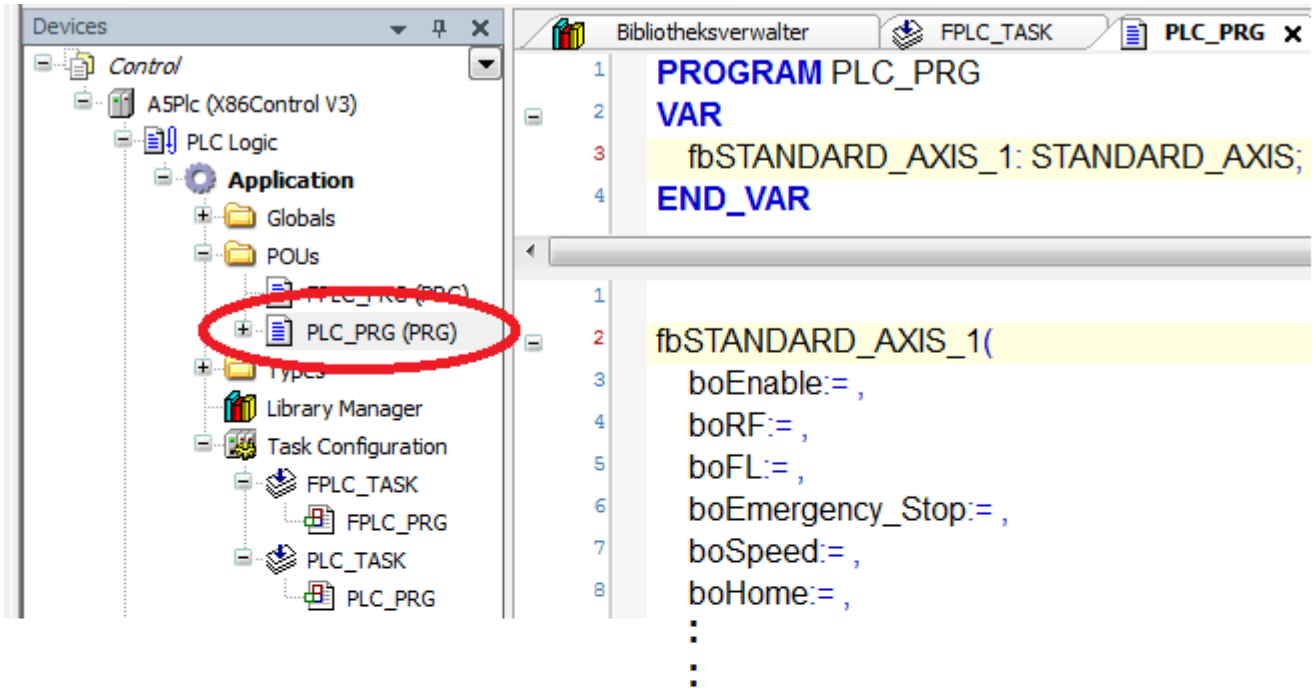
Zusätzlich stehen Funktionen wie Folgeachse, Getriebeverhältnisse, Wickler, Druckmarkenregelung und Tabellenbearbeitung zu Verfügung. Die Funktionalität wird kontinuierlich erweitert.

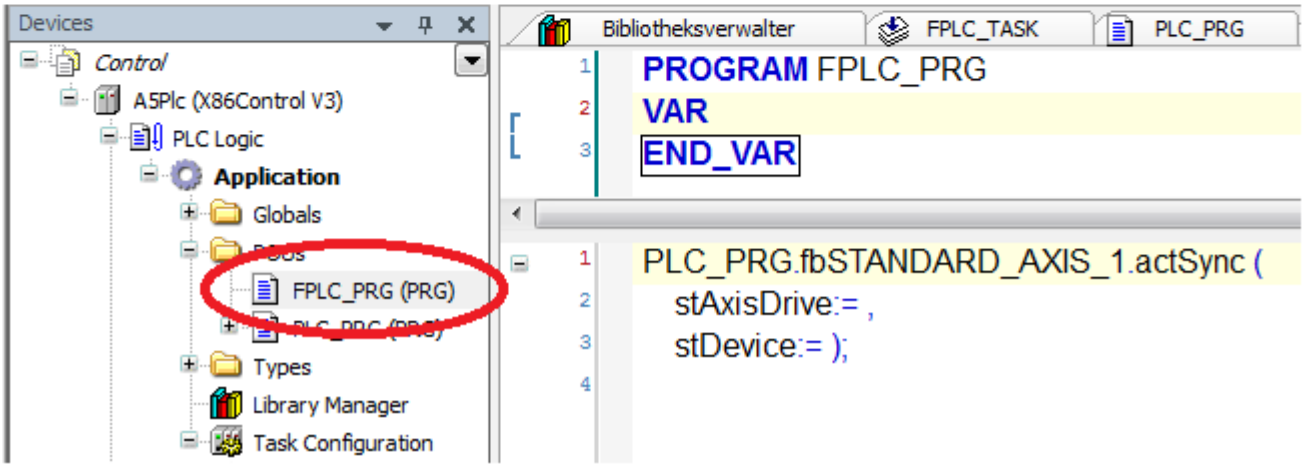
Die für die Applikation benötigten AFL Standardbausteine werden in das PLC Projekt importiert. Der Quelltext liegt offen und kann vom Anwender applikationsspezifisch editiert werden.

Die Standardbausteine werden im CODESYS Programmbaustein 'PLC_PRG' aufgerufen. Achsbausteine verfügen zusätzlich über eine synchrone Aktion (actSync). Die synchrone Aktion wird im 'FPLC_PRG' aufgerufen.

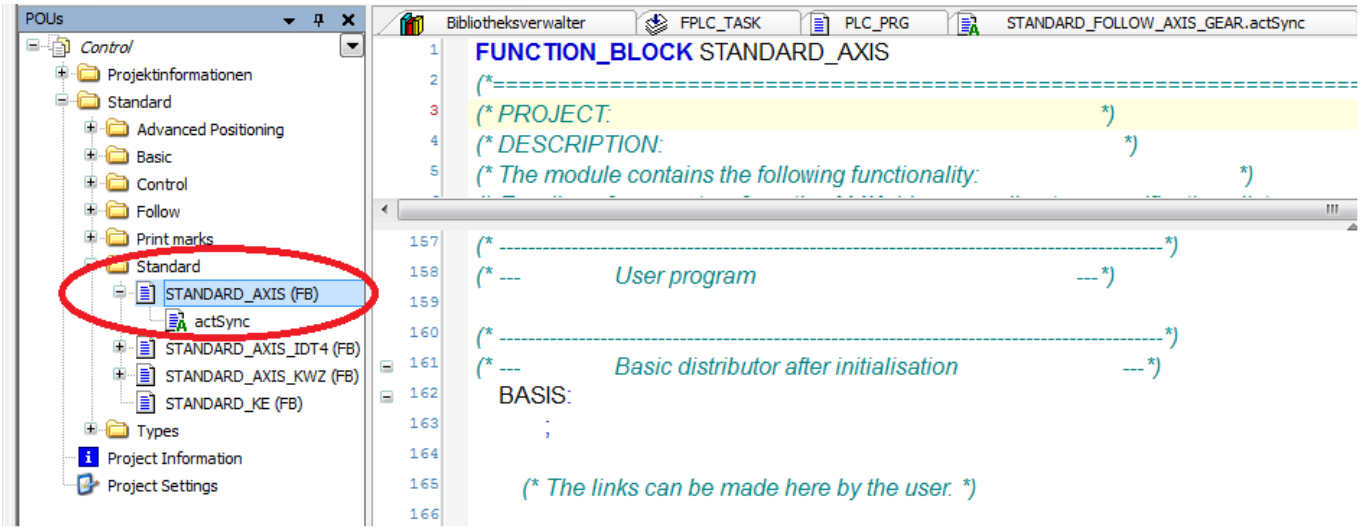
Beispiel: Instanzierter Standardbaustein Typ 'STANDARD_AXIS'

Programmbaustein 'PLC_PRG':	Freilaufende Task für asynchrone Funktionalität / Aufruf und Deklaration 'STANDARD_AXIS'
Programmbaustein 'FPLC_PRG':	Extern ereignisgesteuerte Task, synchronisiert zur ID2 'SERCOS-Zykluszeit' für synchrone Funktionalität / Aufruf synchrone Aktion (actSync)





Editierbarer Quelltext 'STANDARD_AXIS'



Ein PLC Projekt mit AFL Standardbausteinen kann mit AFL Applikationsbausteinen und PLC Grundbausteinen erweitert werden.

Achten Sie darauf, dass sich die Funktionalität der verwendeten Bausteine nicht gegenseitig widerspricht, z. B. mehrere Bausteine in einem Projekt die das Steuersignal 'RF Reglerfreigabe' setzen können.

4.1.1.3 Programmaufbau mit CODESYS V3 und AFL Standardbausteinen

Legen Sie für jedes Geräte (Einspeisung, Steuerung, Antriebe ...), auf das vom PLC Programm zugegriffen werden soll, einen Standardbaustein an.

Die Standardbausteine werden im CODESYS Programmbaustein 'PLC_PRG' aufgerufen. Achsbausteine verfügen zusätzlich über eine synchrone Aktion (actSync). Die synchrone Aktion wird im 'FPLC_PRG' aufgerufen. Der FB BASIC_MOTION ist in jeder Standard Achse integriert und muss somit nicht separat im PLC Programm aufgerufen werden.

Die Schnittstelle zwischen Gerät und AFL Funktionsbaustein wird mit der Variable 'ST_DEVICE' realisiert. 'ST_CONTROL' fasst die steuerungsspezifischen und 'ST_AXIS_DRIVE' die achsspezifischen Daten zusammen.

Die AMK Datei 'V3_Import.export' enthält die AFL Standardbausteine und die dazugehörigen Strukturen. Die Inhalte der Ordner 'Basic', 'Control' und 'Types' müssen zwingend für jedes CODESYS V3 Projekt importiert werden.

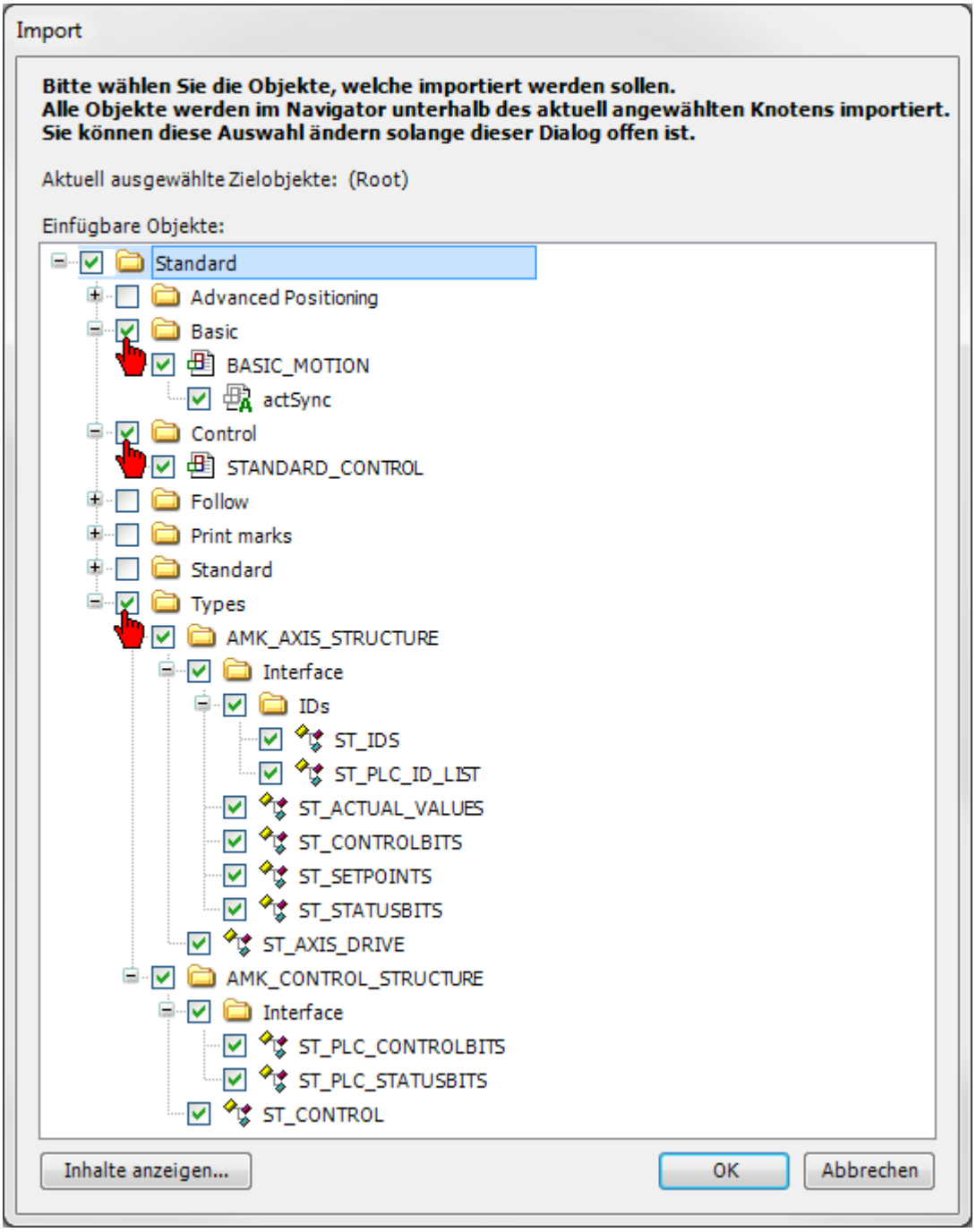
Die Standardbausteine werden mit Hilfe des CODESYS V3 Menüs 'Projekt' → 'Import' in das aktuelle PLC Projekt importiert.

Die Visualisierungselemente der AFL Standardbausteine sind in einer zusätzlichen Bibliothek enthalten.

Voraussetzungen:

- AMK Software AFL Function Library für CODESYS V3 AMK Teile-Nr. O913 installiert.
- AFL Template angewählt. Siehe 'Grund Einstellungen AIPEX PRO' auf Seite 670.

CODESYS V3 Menü 'Projekt' → 'Import'
(C:\Program Files (x86)\3S CODESYS\AmkAddition\Imports\Application\V3_import\V3_Import.export)



Übersicht der Standardbausteine und der dazugehörigen Makros:

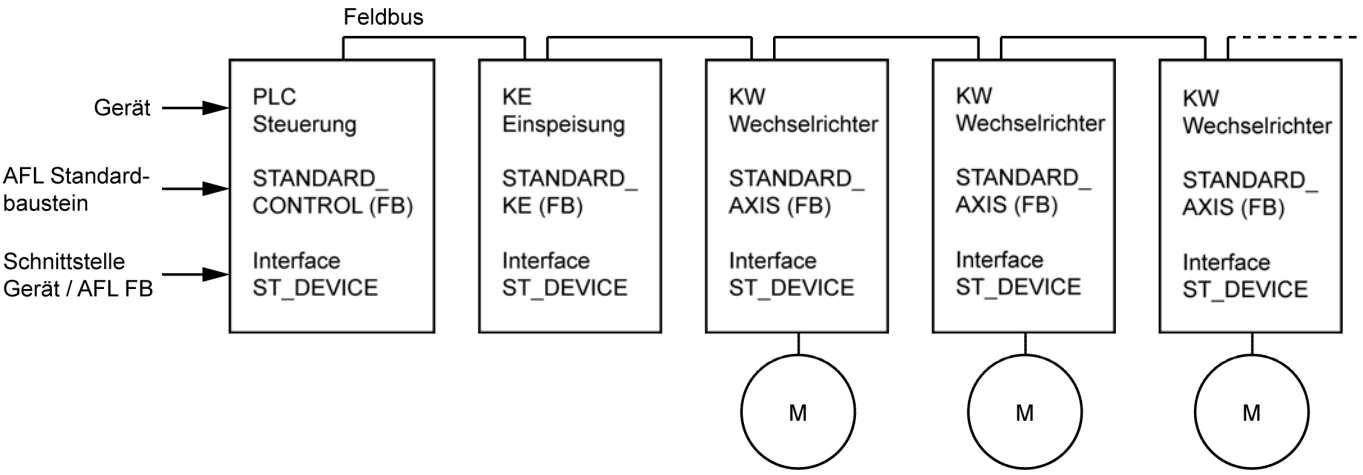
Standard Funktionsbausteine	Makro Ordner	Makro	Für Gerät
STANDARD_CONTROL (FB) [1]	AflImportStandard	AflImportStandardControl	AMKAMAC A4, A5, A6
STANDARD_CONTROL _iSA (FB) [1]	AflImportStandard	AflImportStandardControl_iSA	AMKASMART iSA
STANDARD_KE (FB)	AflImportStandard	AflImportStandardKE	AMKASYN KE/KEN/KES
STANDARD_AXIS_IDT4 (FB) [2]	AflImportStandard	AflImportStandardAxisIDT4	AMKASMART iDT4
STANDARD_AXIS_KWZ (FB) [2]	AflImportStandard	AflImportStandardAxisKWZ	AMKASYN KWZ
STANDARD_AXIS_ihX (FB) [2]	AflImportStandard	AflImportStandardAxis_ihX	AMKASMART ihX
STANDARD_AXIS (FB) [2]	AflImportStandard	AflImportStandardAxis	AMKASYN KW/KWD (KW-R06, KW-R07, KW-R16, KW-R17, KW-R24-R, KW-R25, KW-R26, KW-R27) AMKASMART iDT5 AMKASMART ihX [3]
STANDARD_AXIS_ POSITION_INTERPOSED (FB) [2]	AflImportAdvanced-Positioning	AflImportAdvancedPositioning	
STANDARD_AXIS_ POSITION_TO_THE_MARK (FB) [2]	AflImportPrintMarkAxis	AflImportStandardAxis-PrintMarkPositionToTheMark	
STANDARD_AXIS_ PRINTMARK_INSETTER_ INTERVAL (FB) [2]	AflImportPrintMarkAxis	AflImportStandardAxis-PrintMarkInsetterInterval	
STANDARD_AXIS_ PRINTMARK_INSETTER_ CONT (FB) [2]	AflImportPrintMarkAxis	AflImportStandardAxis-PrintMarkInsetterCont	
STANDARD_AXIS_ PRINTMARK_REGISTER_ CONTROL_CONT (FB) [2]	AflImportPrintMarkAxis	AflImportStandardAxis-PrintMarkRegisterControlCont	
STANDARD_AXIS_ PRINTMARK_REGISTER_ CONTROL_DISCONT (FB) [2]	AflImportPrintMarkAxis	AflImportStandardAxis-PrintMarkRegisterControlDiscont	
STANDARD_FOLLOW_AXIS_GEAR (FB) [2]	AflImportFollowAxis	AflImportFollowAxisGear	
STANDARD_FOLLOW_AXIS_ CONTINUOUS_TABLE1_TABLE2 (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-ContinuousTable1Table2	
STANDARD_FOLLOW_AXIS_ SWITCHING_TABLE_WITH_ OUTGEAR (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-SwitchingTableWithOutgear	
STANDARD_FOLLOW_AXIS_ SWITCHING_TABLE1_TABLE2 (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-SwitchingTable1Table2	
STANDARD_FOLLOW_AXIS_ TABLE_CONTINUOUS (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-ContinuousTable	
STANDARD_FOLLOW_TABLE_ ENGAGE_DISENG_AXIS (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-TableEngageDiseng	
STANDARD_FOLLOW_TABLE_ START_AUTOSTOP_AXIS (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-TableStartAutoStop	
STANDARD_WINDER_DANCER_ AXIS (FB) [2]	AflImportStandardWinder	AflImportStandardWinderDancer	
STANDARD_WINDER_TORQUE_ AXIS (FB) [2]	AflImportStandardWinder	AflImportStandardWinderTorque	

[1] Alle Standardbausteine vom Typ CONTROL benötigen die 'AMK CONTROL STRUCTURE' 'ST_CONTROL (STRUCT)'. 'ST_CONTROL (STRUCT)' wird mit dem Makro AflImportBasicControl aus dem Ordner AflImportBasic importiert. 'ST_CONTROL (STRUCT)' muss 1 x pro PLC Projekt importiert werden.

[2] Alle Standardbausteine vom Typ AXIS rufen den FB 'BASIC_MOTION' auf. Der FB 'BASIC_MOTION' wird mit dem Makro AflImportBasicDrive aus dem Ordner AflImportBasic importiert. FB 'BASIC_MOTION' muss 1 x pro PLC Projekt importiert werden.

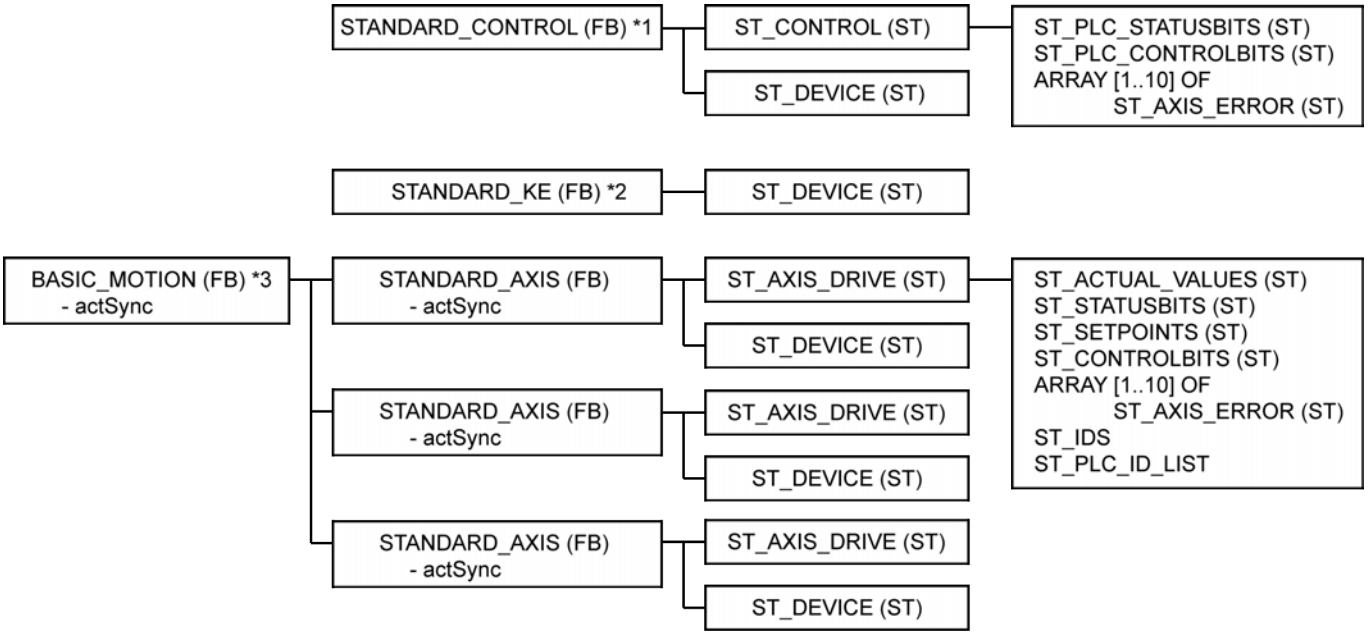
[3] Alle Standard Funktionsbausteine mit erweiterter Funktionalität können mit einer Modifikation für den ihX verwendet werden.
Siehe 'Modifizierte 'Standard Achsen' für ihX Antriebe' auf Seite 209.

Programmaufbau mit CODESYS V3 und AFL Standard Funktionsbausteinen



Aufbau dazugehöriges PLC Programm:

Die Makros 'AflImport...' importieren die Standardbausteine und Strukturen in das aktuelle PLC Projekt.



*1 AflImportBasicControl importiert die Struktur 'ST_CONTROL' und muss pro Projekt nur einmal eingefügt werden.

*2 Wird nur beim Einsatz der Einspeisemodule KE, KEN oder KES benötigt. (Ausnahme: KEN 05-xx).
Die Geräte AMKASMART iC und AMKASYN KEN 05-xx laden automatisch den Zwischenkreis für die angeschlossenen Wechselrichter.

*3 AflImportBasicDrive importiert den 'BASIC_MOTION' (FB) und muss pro Projekt nur einmal eingefügt werden. Der 'BASIC_MOTION' muss nicht vom Anwender instanziert werden. Er wird im PLC Projekt automatisch über die 'STANDARD_AXIS' Funktionsbausteine aufgerufen.

Eine Instanz des Funktionsbausteins 'STANDARD_CONTROL' (bei A-Serie) bzw.'STANDARD_CONTROL'_iSA (bei iSA) muss zwingend in jedem PLC Projekt verwendet werden.

EtherCAT

Die Kombination mit einem der AFL Funktionsbausteine 'STANDARD_AXIS' und ID32796 'Quelle Reglerfreigabe' Code 0: 'RF über Binäreingang' ist nicht zulässig.

Das Steuersignal RF muss zwingend über den Funktionsbaustein gesetzt werden.

Zulässige Varianten:

Code 5: 'RF über EtherCAT' bzw.

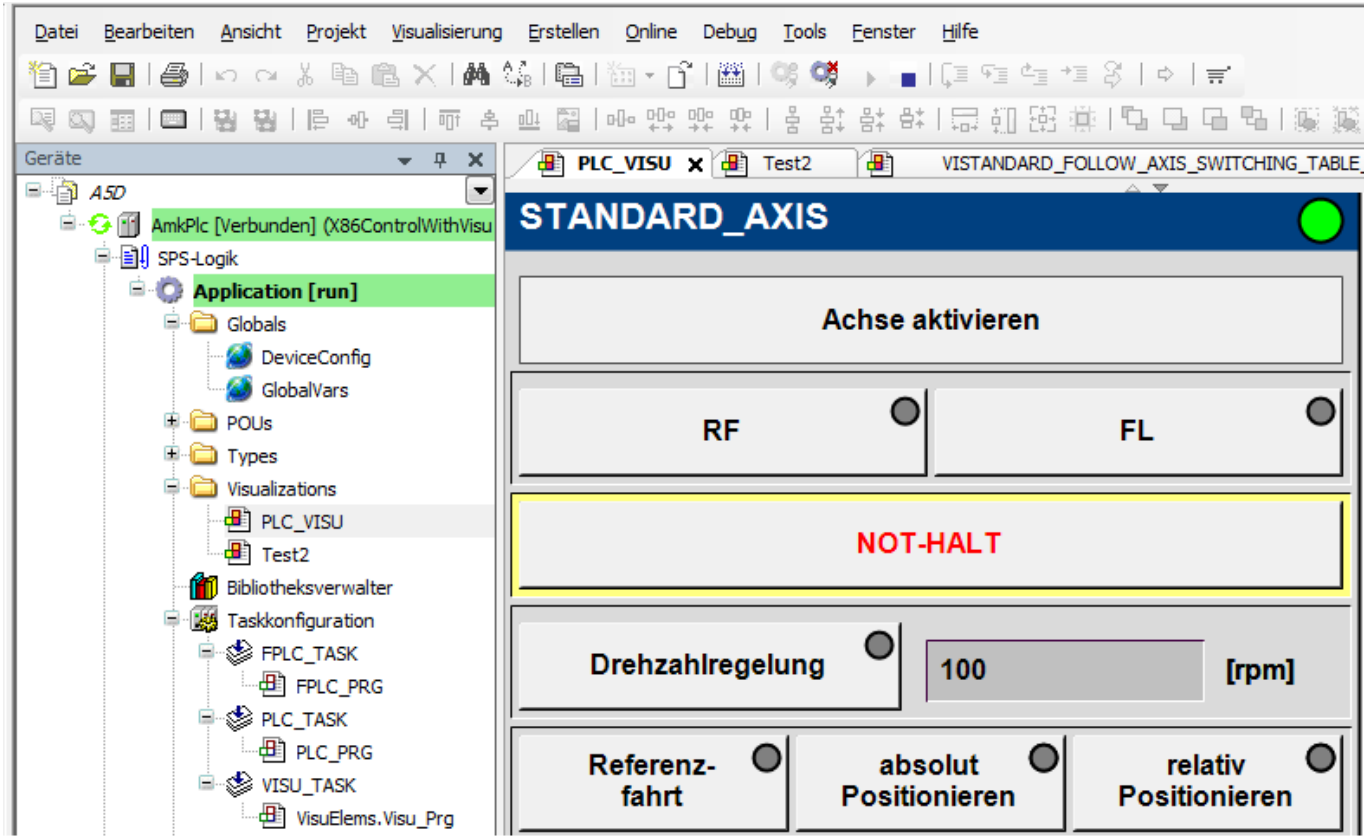
Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF'

4.1.1.4 Visualisierung Standardbausteine

Für alle Standardbausteine gibt es fertige Visualisierungselemente.

Siehe 'Visualisierung' auf Seite 701.

Beispiel: Anzeige im 'RUN Modus'



4.1.1.5 Übersicht Standardbausteine und Achsstrukturen

Die Standardbausteine werden bei CODESYS V3 mit Hilfe der Import Funktion in das aktuelle PLC Projekt eingefügt. Siehe CODESYS Menü 'Projekt' → 'Importieren'.

STANDARD_CONTROL (FB)

(Import Ordner: POUs → Control)

Folgende Grundfunktionalität für die Steuerung steht zu Verfügung:

- Statusbits (SBM Systembereit Meldung, QFL Quitt. Fehler gelöscht, Netzwerk und Bus Informationen)
- Steuern von FL Fehler löschen
- Auslesen von Diagnosemeldungen

Siehe 'STANDARD_CONTROL (FB)' auf Seite 197.

STANDARD_CONTROL _iSA (FB)

(Import Ordner: POUs → Control)

Folgende Grundfunktionalität für die Steuerung mit Einspeisung steht zu Verfügung:

- auslesen der Statusbitinformationen (Netzwerk bereit, Bus bereit, Bus Warnung, Bus Fehler, SBM, QUE und QFL)
- das Steuern von FL
- das Auslesen von Diagnosemeldungen

Siehe 'STANDARD_CONTROL_iSA (FB)' auf Seite 199.

STANDARD_KE (FB)
(Import Ordner: POUs → Standard)
Folgende Grundfunktionalität für die Einspeisung steht zu Verfügung:

- Statusbits (QUE Quitt. Umrichter Ein, SBM Systembereit Meldung)
- Steuerbits (UE Umrichter Ein, FL Fehler Löschen)
- Ausgabe Diagnosenummer

Siehe 'STANDARD_KE (FB)' auf Seite 201.

STANDARD_AXIS (FB)
STANDARD_IDT4 (FB)
STANDARD_ihX (FB)
STANDARD_KWZ (FB)
(Import Ordner: POUs → Standard)
Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Siehe 'STANDARD_AXIS (FB) (STANDARD_AXIS_KWZ, STANDARD_AXIS_IDT4, STANDARD_AXIS_ihX)' auf Seite 202.

4.1.1.5.1 Erweiterte Standard Achsen

Die erweiterten Achsen verfügen über dieselben Grundfunktionalitäten wie der Funktionsbaustein 'STANDARD_AXIS'.

Mit den Achsen vom Typ 'Winder' kann ein Tänzerwickler oder ein sensorloser Auf- und Abwickler realisiert werden.
Mit der Achse vom Typ 'Position' können Sie eine Druckmarkenregelung oder ablösende Positionierung realisieren.
Mit den Achsen vom Typ 'Printmark' können Sie eine Druckmarkenregelung realisieren.
Es stehen Achsen vom Typ 'Follow' zu Verfügung. Diese folgen synchron einem Sollwert (Master).
Zusätzlich können Tabellen für z. B. Kurvenscheiben–Funktionen verrechnet werden.

STANDARD_AXIS_POSITION_INTERPOSED (FB)
(Import Ordner: POUs → Advanced Positioning)
Erweiterte Funktionalität:

- Ablösende Positionierung

Siehe 'STANDARD_AXIS_POSITION_INTERPOSED (FB)' auf Seite 210.

STANDARD_AXIS_POSITION_TO_THE_MARK (FB)
(Import Ordner: POUs → Print marks)
Erweiterte Funktionalität:

- Druckmarkenregelung

Siehe 'STANDARD_AXIS_POSITION_TO_THE_MARK (FB)' auf Seite 219.

STANDARD_AXIS_PRINTMARK_INSETTER_CONT (FB)
(Import Ordner: POUs → Print marks)
Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des zu bearbeitenden Bandes wird zu einem Masterformat geregelt (Insetting)

Siehe 'STANDARD_AXIS_PRINTMARK_INSETTER_CONT (FB)' auf Seite 228.

STANDARD_AXIS_PRINTMARK_INSETTER_INTERVAL (FB)
(Import Ordner: POUs → Print marks)
Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des zu bearbeitenden Bandes wird zu einem Masterformat geregelt (Insetting)

Siehe 'STANDARD_AXIS_PRINTMARK_INSETTER_INTERVAL (FB)' auf Seite 240.

STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_CONT (FB)
(Import Ordner: POUs → Print marks)
Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des bearbeitenden Aggregats (z.B. Messer, Magnetstanze) wird zu einem Masterformat geregelt (Registerregler)

Siehe 'STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_CONT (FB)' auf Seite 252.

STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_DISCONT (FB)
(Import Ordner: POUs → Print marks)
Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des bearbeitenden Aggregats (z.B. Messer, Magnetstanze) wird zu einem Masterformat geregelt (Registerregler)

Siehe 'STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_DISCONT (FB)' auf Seite 264.

STANDARD_FOLLOW_AXIS_GEAR (FB)
(Import Ordner: POUs → Follow)
Erweiterte Funktionalität:

- Die Slave Achse folgt kontinuierlich einem Master.
- Ein Getriebeverhältnis kann vorgegeben werden.

Siehe 'STANDARD_FOLLOW_AXIS_GEAR (FB)' auf Seite 276.

STANDARD_FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2 (FB)
(Import Ordner: POUs → Follow)
Erweiterte Funktionalität:

- Die Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle.
- Am Tabellenende kann eine Tabellenumschaltung durchgeführt werden.

Siehe 'STANDARD_FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2 (FB)' auf Seite 284.

STANDARD_FOLLOW_AXIS_SWITCHING_TABLE_WITH_OUTGEAR (FB)
(Import Ordner: POUs → Follow)
Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle.
- Am Tabellenende kann eine Tabellenumschaltung durchgeführt werden.
- Die Tabelle kann über ein Getriebeverhältnis angepasste werden (Stauchen und Stecken).

Siehe 'STANDARD_FOLLOW_AXIS_SWITCHING_TABLE_WITH_OUTGEAR (FB)' auf Seite 293.

STANDARD_FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2 (FB)
(Import Ordner: POUs → Follow)
Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle mit Ein- und Auskuppeltabelle.
- Am Tabellenende kann eine Tabellenumschaltung durchgeführt werden.

Siehe 'STANDARD_FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2 (FB)' auf Seite 301.

STANDARD_FOLLOW_AXIS_TABLE_CONTINUOUS (FB)
(Import Ordner: POUs → Follow)
Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle.

Siehe 'STANDARD_FOLLOW_AXIS_TABLE_CONTINUOUS (FB)' auf Seite 311.

STANDARD_FOLLOW_TABLE_ENGAGE_DISENG_AXIS (FB)
(Import Ordner: POUs → Follow)
Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle mit Ein- und Auskuppeltabelle.

Siehe 'STANDARD_FOLLOW_TABLE_ENGAGE_DISENG_AXIS (FB)' auf Seite 319.

STANDARD_FOLLOW_TABLE_START_AUTOSTOP_AXIS (FB)
(Import Ordner: POUs → Follow)
Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle mit Ein- und Auskuppeltabelle.

Siehe 'STANDARD_FOLLOW_TABLE_START_AUTOSTOP_AXIS (FB)' auf Seite 328.

STANDARD_WINDER_DANCER_AXIS (FB)
(Import Ordner: POUs → Winder)
Erweiterte Funktionalität:

- Der Funktionbaustein 'STANDARD_WINDER_TORQUE_AXIS' realisiert einen sensorlosen Auf- und Abwickler und konfigurierbarer Bahnspannung mit folgenden Merkmalen:
- Momentgesteuerter Zentrumswickler
- Betrieb als Auf- und Abwickler
- Durchmesserberechnung
- Durchmessersuche beim Start des Wicklers
- Ermittlung des Rollenträgheitsmoments bei unbekanntem Durchmesser
- Synchronisieren auf eine laufende Bahn

Siehe 'STANDARD_WINDER_DANCER_AXIS (FB)' auf Seite 337.

STANDARD_WINDER_TORQUE_AXIS (FB)
(Import Ordner: POUs → Winder)
Erweiterte Funktionalität:

- Der Funktionsbaustein 'STANDARD_WINDER_DANCER_AXIS' realisiert einen Tänzerwickler mit folgenden Merkmalen:
- PID geregelter Tänzerwickler
- Betrieb als Auf- und Abwickler
- Durchmesserberechnung
- Durchmessersuche bei Start des Wicklers

Siehe 'STANDARD_WINDER_TORQUE_AXIS (FB)' auf Seite 347.

4.1.1.5.2 Strukturen

Folgende Zugriffsmöglichkeiten stehen zu Verfügung:

PLC Steuerung (ST_CONTROL):

(Import Ordner: POUs → Data typs → AMK_CONTROL_STRUCTURE)

- Statusbits (SBM Systembereitmeldung, Netzwerkstatus)
- Steuerbits (FL Fehler Löschen, Diagnosemeldung Lesen)
- Diagnosearray (Speicher für 10 Diagnosemeldungen)

Siehe 'ST_CONTROL (ST)' auf Seite 191.

Achse (ST_AXIS_DRIVE):

(Import Ordner: POUs → Data typs → AMK_AXIS_STRUCTURE)

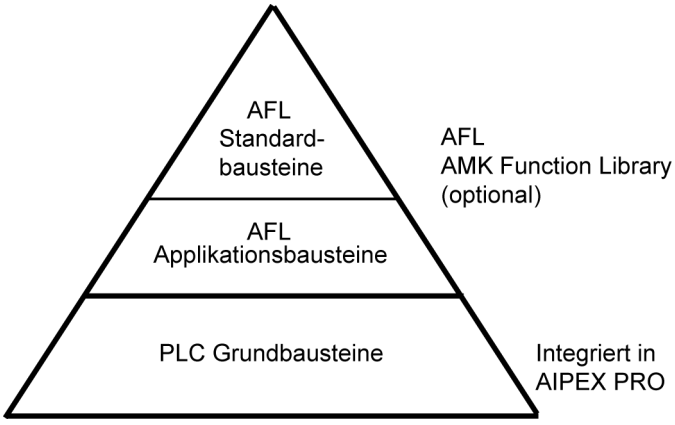
- Sollwerte (Drehzahl-, Lage-, Momentsollwerte, Vorsteuerwerte)
- Istwerte (Drehzahl-, Lage-, Momentistwert, Messwerte ...)
- Steuerbits (RF Reglerfreigabe, FL Fehler Löschen, Diagnosemeldung Lesen)
- Statusbits (Drehzahlfenster, Stillstandsfenster...)
- Diagnosearray (Speicher für 10 Diagnosemeldungen)
- Parameterliste Lesen
- Konfiguration (PLC Funktionen)

Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.

4.1.2 Einführung AFL Standardbausteine für CODESYS V2

Mit AFL Standardbausteinen wird die Funktionalität der AMK Geräte in ein PLC Projekt eingebunden. Der Programmieraufwand durch die Verwendung der AFL Standardbausteine ist deutlich reduziert. Die AFL Standardbausteine bestehen aus einer Vielzahl von Funktionsbausteinen aus verschiedenen AMK Bibliotheken.

4.1.2.1 Übersicht der AMK Bibliotheken



Die AFL Standardbausteine bestehen aus AFL Applikationsbausteinen und sind Bestandteil der AFL Function Library.

Siehe 'AMK AFL Applikationsbausteine' auf Seite 366.

Die AFL Applikationsbausteine bestehen aus den PLC Grundbausteinen. (Siehe DokumentSoftwarebeschreibung AIPEX PRO V3 (Teile-Nr.205210))

Beispiel anhand dem Standardbaustein 'STANDARD_AXIS':

AFL Standardbausteine

Der Standardbaustein 'STANDARD_AXIS' (FB) besteht aus den AFL Applikationsbausteinen 'MANUAL_JOG_VAJ', 'POSITION_ABSOLUT_VAJ', 'MANUAL_VELOCITY', 'POSITION_HOMING_FIXED_STOP' und weiteren aus der AMK AFL Library (AmkAfl.lib). Der Quellcode des Standardbausteins ist vom Anwender editierbar.

AFL Applikationsbausteine

AMK stellt mit den AFL Applikationsbausteinen komplexere Funktionen zur Verfügung. Sie bestehen aus einzelnen PLC Grundbausteinen. Die Funktionalität der AFL Applikationsbausteine ist fest definiert und kann vom Anwender nicht editiert werden.

Beispiel: MANUAL_JOG_VAJ (FB) aus der AMK AFL Library (AmkAfl.lib)

Der AFL Applikationsbaustein 'MANUAL_JOG_VAJ' realisiert den Tippbetrieb (Plus / Minus) in Lageregelung. Dabei kann der Anwender neben Position, Geschwindigkeit und Beschleunigung auch den Ruck vorgeben.

PLC Grundbausteinen

Der AFL Applikationsbaustein 'MANUAL_JOG_VAJ' besteht aus den PLC Grundbausteinen 'VGEN_AJ' und 'RATIO_INC' aus der Bibliothek AmkBase.lib. Die Funktionalität der PLC Grundbausteine ist fest definiert und kann vom Anwender nicht editiert werden.

4.1.2.2 Verwendung der AFL Standardbausteine

Die AFL Standardbausteine ermöglichen dem Anwender den einfachen Zugriff auf alle Standard Funktionen der Steuerungen, Einspeisungen und Achsen (Antriebe).

Vordefinierte Strukturen ermöglichen den Zugriff auf Soll- und Istwerte, Steuer- und Statusbits, Parameter und Diagnosemeldungen.

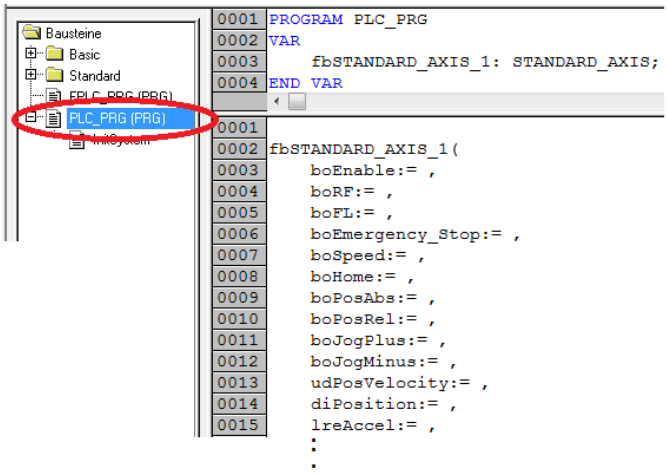
Zusätzlich stehen Funktionen wie Folgeachse, Getriebeverhältnisse, Wickler, Druckmarkenregelung und Tabellenbearbeitung zu Verfügung. Die Funktionalität wird kontinuierlich erweitert.

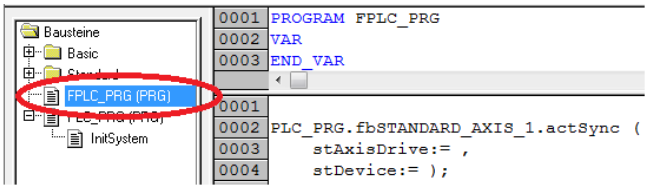
Die für die Applikation benötigten AFL Standardbausteine werden in das PLC Projekt importiert. Der Quelltext liegt offen und kann vom Anwender applikationsspezifisch editiert werden.

Die Standardbausteine werden im CODESYS Programmbaustein 'PLC_PRG' aufgerufen. Achsbausteine verfügen zusätzlich über eine synchrone Aktion (actSync). Die synchrone Aktion wird im 'FPLC_PRG' aufgerufen.

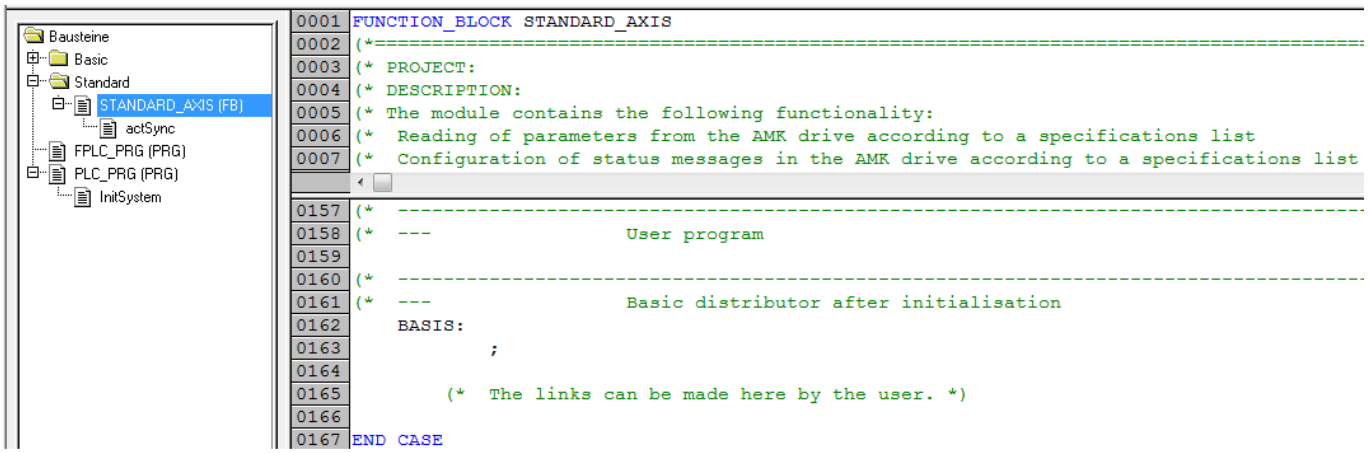
Beispiel: Instanzierter Standardbaustein Typ 'STANDARD_AXIS'

Programmbaustein 'PLC_PRG':	Freilaufende Task für asynchrone Funktionalität / Aufruf und Deklaration 'STANDARD_AXIS'
Programmbaustein 'FPLC_PRG':	Extern ereignisgesteuerte Task, synchronisiert zur ID2 'SERCOS-Zykluszeit' für synchrone Funktionalität / Aufruf synchrone Aktion (actSync)





Editierbarer Quelltext 'STANDARD_AXIS'



Ein PLC Projekt mit AFL Standardbausteinen kann mit AFL Applikationsbausteinen und PLC Grundbausteinen erweitert werden.

Achten Sie darauf, dass sich die Funktionalität der verwendeten Bausteine nicht gegenseitig widerspricht, z. B. mehrere Bausteine in einem Projekt die das Steuersignal 'RF Reglerfreigabe' setzen können.

4.1.2.3 Programmaufbau mit CODESYS V2 und AFL Standardbausteinen

Legen Sie für jedes Geräte (Einspeisung, Steuerung, Antriebe ...), auf das vom PLC Programm zugegriffen werden soll, einen Standardbaustein an.

Die Standardbausteine werden im CODESYS Programmbaustein 'PLC_PRG' aufgerufen. Achsbausteine verfügen zusätzlich über eine synchrone Aktion (actSync). Die synchrone Aktion wird im 'FPLC_PRG' aufgerufen. Der FB BASIC_MOTION ist in jeder Standard Achse integriert und muss somit nicht separat im PLC Programm aufgerufen werden.

Die Schnittstelle zwischen Gerät und AFL Funktionsbaustein wird mit der Variable 'ST_DEVICE' realisiert. 'ST_CONTROL' fasst die steuerungsspezifischen und 'ST_AXIS_DRIVE' die achsspezifischen Daten zusammen.

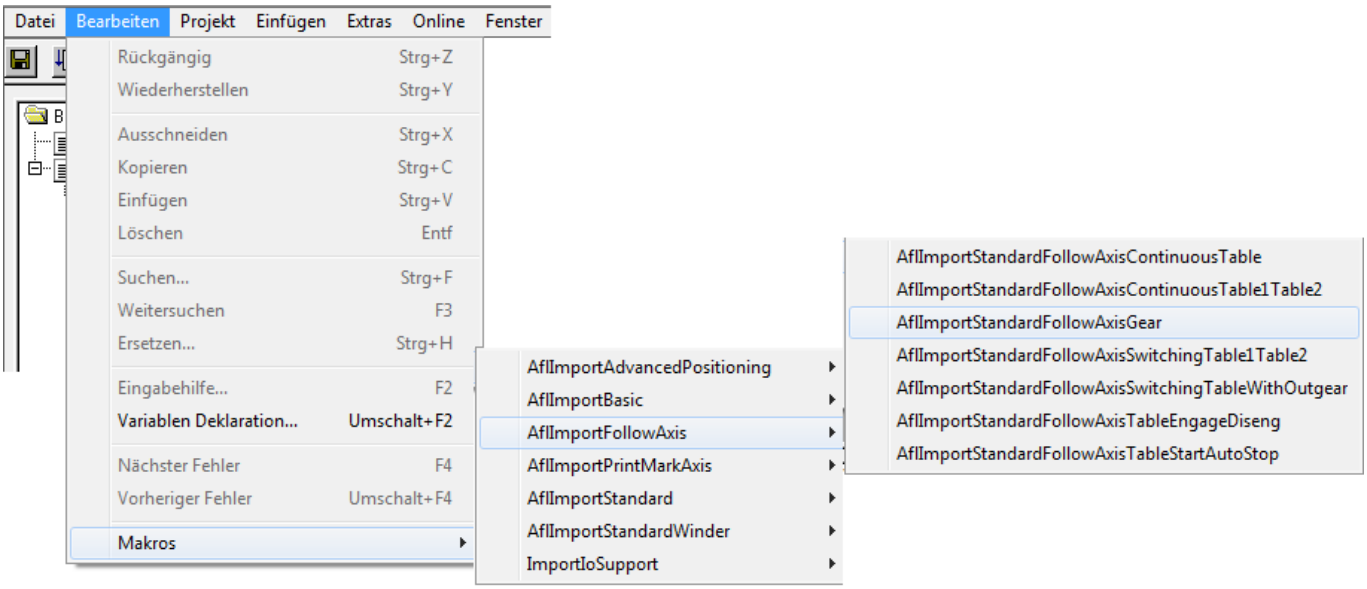
Zu jedem Standardbaustein wird eine dazu passende Visualisierung importiert.

Die Standardbausteine werden mit Hilfe von Makros in das aktuelle PLC Projekt importiert. Das Makro enthält alle Anweisungen, um einen exportierten Standardbaustein in das aktuelle Projekt zu importieren.

Voraussetzungen:

- AMK Software AFL Function Library für CODESYS V2 AMK Teile-Nr. O877 installiert.
- AFL Template angewählt. Siehe 'Konfiguration erstellen (allgemein)' auf Seite 766.

Die Makros werden unter CODESYS V2 mit dem Menü 'Bearbeiten' → 'Makros' ausgeführt.



Übersicht der Standardbausteine und der dazugehörigen Makros:

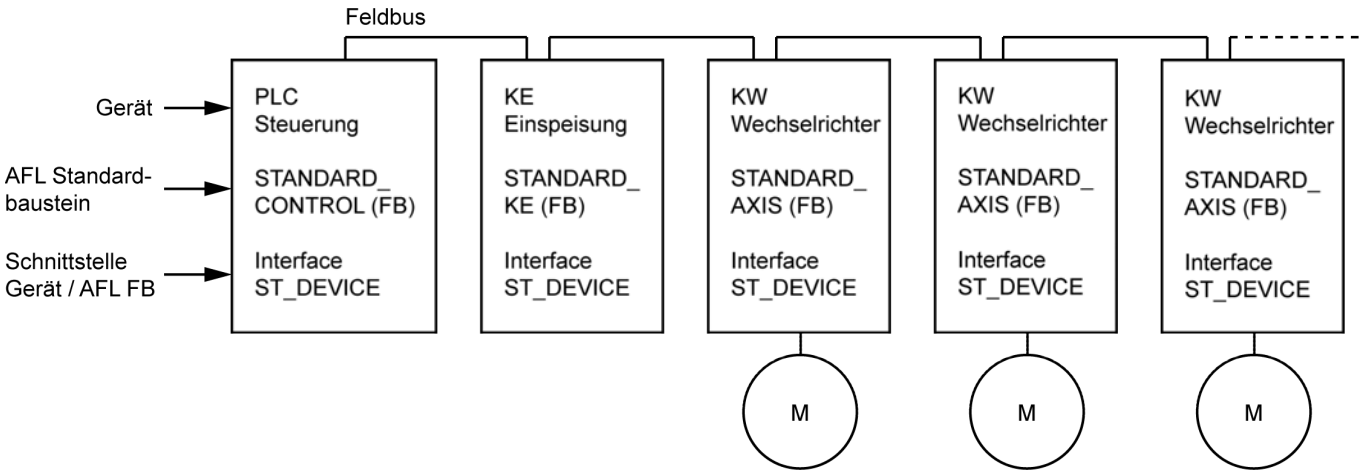
Standard Funktionsbausteine	Makro Ordner	Makro	Für Gerät
STANDARD_CONTROL (FB) [1]	AflImportStandard	AflImportStandardControl	AMKAMAC A4, A5, A6
STANDARD_CONTROL _iSA (FB) [1]	AflImportStandard	AflImportStandardControl_iSA	AMKASMART iSA
STANDARD_KE (FB)	AflImportStandard	AflImportStandardKE	AMKASYN KE/KEN/KES
STANDARD_AXIS_IDT4 (FB) [2]	AflImportStandard	AflImportStandardAxisIDT4	AMKASMART iDT4
STANDARD_AXIS_KWZ (FB) [2]	AflImportStandard	AflImportStandardAxisKWZ	AMKASYN KWZ
STANDARD_AXIS_ihX (FB) [2]	AflImportStandard	AflImportStandardAxis_ihX	AMKASMART ihX
STANDARD_AXIS (FB) [2]	AflImportStandard	AflImportStandardAxis	AMKASYN KW/KWD (KW-R06, KW-R07, KW-R16, KW-R17, KW-R24-R, KW-R25, KW-R26, KW-R27) AMKASMART iDT5 AMKASMART ihX [3]
STANDARD_AXIS_ POSITION_INTERPOSED (FB) [2]	AflImportAdvanced-Positioning	AflImportAdvancedPositioning	
STANDARD_AXIS_ POSITION_TO_THE_MARK (FB) [2]	AflImportPrintMarkAxis	AflImportStandardAxis-PrintMarkPositionToTheMark	
STANDARD_AXIS_ PRINTMARK_INSETTER_ INTERVAL (FB) [2]	AflImportPrintMarkAxis	AflImportStandardAxis-PrintMarkInsetterInterval	
STANDARD_AXIS_ PRINTMARK_INSETTER_ CONT (FB) [2]	AflImportPrintMarkAxis	AflImportStandardAxis-PrintMarkInsetterCont	
STANDARD_AXIS_ PRINTMARK_REGISTER_ CONTROL_CONT (FB) [2]	AflImportPrintMarkAxis	AflImportStandardAxis-PrintMarkRegisterControlCont	
STANDARD_AXIS_ PRINTMARK_REGISTER_ CONTROL_DISCONT (FB) [2]	AflImportPrintMarkAxis	AflImportStandardAxis-PrintMarkRegisterControlDiscont	
STANDARD_FOLLOW_AXIS_GEAR (FB) [2]	AflImportFollowAxis	AflImportFollowAxisGear	
STANDARD_FOLLOW_AXIS_ CONTINUOUS_TABLE1_TABLE2 (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-ContinuousTable1Table2	
STANDARD_FOLLOW_AXIS_ SWITCHING_TABLE_WITH_ OUTGEAR (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-SwitchingTableWithOutgear	
STANDARD_FOLLOW_AXIS_ SWITCHING_TABLE1_TABLE2 (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-SwitchingTable1Table2	
STANDARD_FOLLOW_AXIS_ TABLE_CONTINUOUS (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-ContinuousTable	
STANDARD_FOLLOW_TABLE_ ENGAGE_DISENG_AXIS (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-TableEngageDiseng	
STANDARD_FOLLOW_TABLE_ START_AUTOSTOP_AXIS (FB) [2]	AflImportFollowAxis	AflImportFollowAxis-TableStartAutoStop	
STANDARD_WINDER_DANCER_ AXIS (FB) [2]	AflImportStandardWinder	AflImportStandardWinderDancer	
STANDARD_WINDER_TORQUE_ AXIS (FB) [2]	AflImportStandardWinder	AflImportStandardWinderTorque	

[1] Alle Standardbausteine vom Typ CONTROL benötigen die 'AMK CONTROL STRUCTURE' 'ST_CONTROL (STRUCT)'. 'ST_CONTROL (STRUCT)' wird mit dem Makro AflImportBasicControl aus dem Ordner AflImportBasic importiert. 'ST_CONTROL (STRUCT)' muss 1 x pro PLC Projekt importiert werden.

[2] Alle Standardbausteine vom Typ AXIS rufen den FB 'BASIC_MOTION' auf. Der FB 'BASIC_MOTION' wird mit dem Makro AflImportBasicDrive aus dem Ordner AflImportBasic importiert. FB 'BASIC_MOTION' muss 1 x pro PLC Projekt importiert werden.

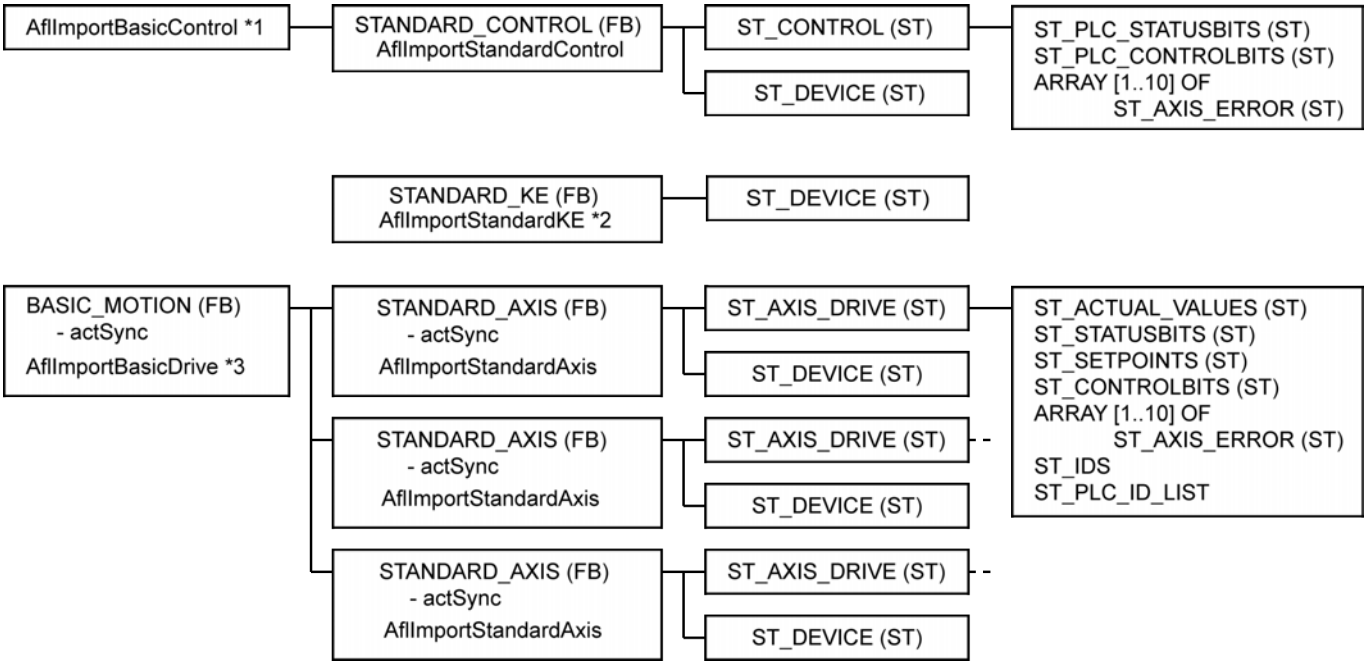
[3] Alle Standard Funktionsbausteine mit erweiterter Funktionalität können mit einer Modifikation für den ihX verwendet werden.
Siehe 'Modifizierte 'Standard Achsen' für ihX Antriebe' auf Seite 209.

Beispielapplikation:



Aufbau dazugehöriges PLC Programm:

Die Makros 'AflImport...' importieren die Standardbausteine und Strukturen in das aktuelle PLC Projekt.



*1 AflImportBasicControl importiert die Struktur 'ST_CONTROL' und muss pro Projekt nur einmal eingefügt werden.

*2 Wird nur beim Einsatz der Einspeisemodule KE, KEN oder KES benötigt. (Ausnahme: KEN 05-xx).
Die Geräte AMKASMART iC und AMKASYN KEN 05-xx laden automatisch den Zwischenkreis für die angeschlossenen Wechselrichter.

*3 AflImportBasicDrive importiert den 'BASIC_MOTION' (FB) und muss pro Projekt nur einmal eingefügt werden. Der 'BASIC_MOTION' muss nicht vom Anwender instanziert werden. Er wird im PLC Projekt automatisch über die 'STANDARD_AXIS' Funktionsbausteine aufgerufen.

Eine Instanz des Funktionsbausteins 'STANDARD_CONTROL' (bei A-Serie) bzw.'STANDARD_CONTROL'_iSA (bei iSA) muss zwingend in jedem PLC Projekt verwendet werden.

EtherCAT

Die Kombination mit einem der AFL Funktionsbausteine 'STANDARD_AXIS' und ID32796 'Quelle Reglerfreigabe' Code 0: 'RF über Binäreingang' ist nicht zulässig.

Das Steuersignal RF muss zwingend über den Funktionsbaustein gesetzt werden.

Zulässige Varianten:

Code 5: 'RF über EtherCAT' bzw.

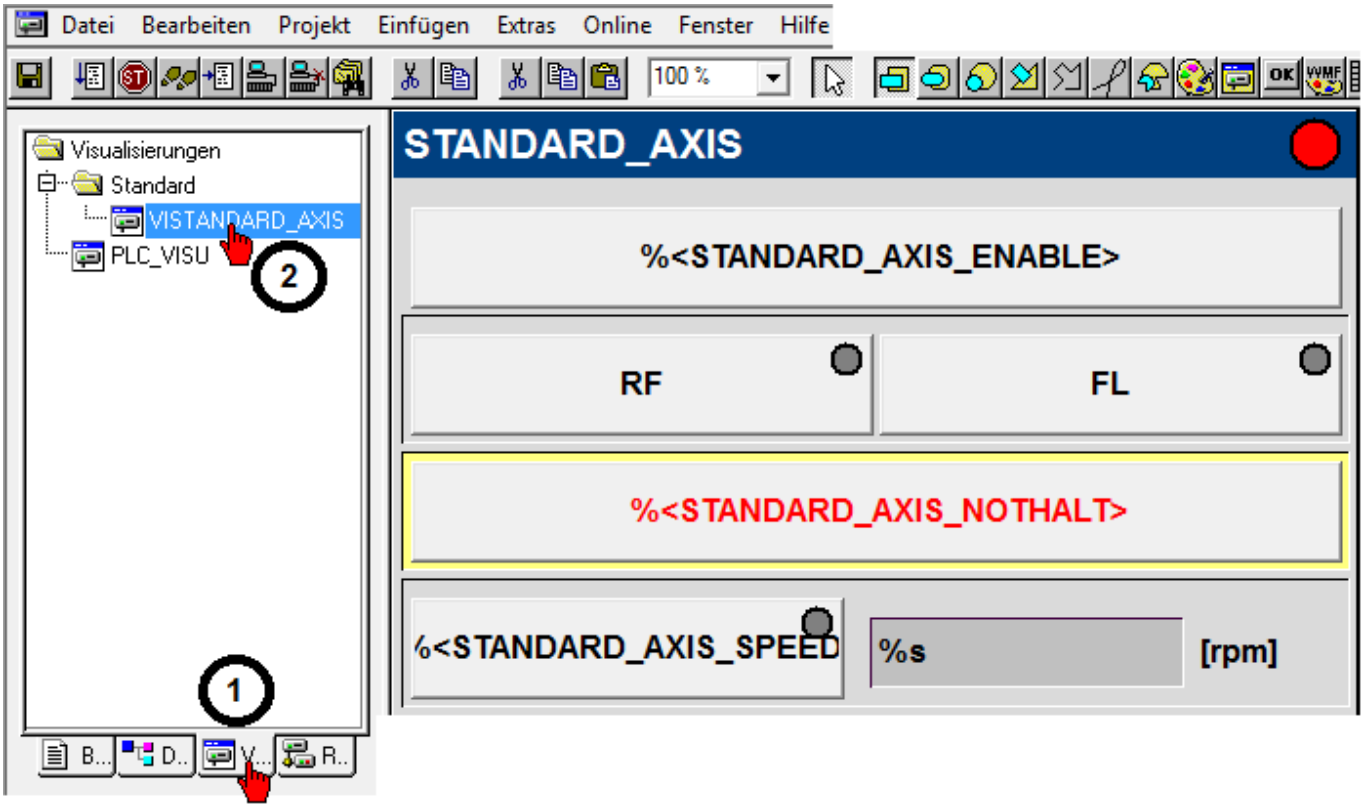
Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF'

4.1.2.4 Visualisierung Standardbausteine

Die Visualisierungen der Standardbaustein werden in den Tap/Ordner 'Visualisierungen' importiert.

Für jeden Standardbaustein muss eine Instanz der Visualisierung gebildet werden.

Siehe 'Visualisierung' auf Seite 796.



4.1.2.5 Übersicht Standardbausteine und Achsstrukturen

Die Standardbausteine werden bei CODESYS V2 mit Hilfe von Makros in das aktuelle PLC Projekt eingefügt. Siehe CODESYS Menü 'Bearbeiten' → 'Makros'.

STANDARD_CONTROL (FB)

(Makro: AflImportBasic - AflImportBasicControl + AflImportStandard - AflImportStandardControl)

Folgende Grundfunktionalität für die Steuerung steht zu Verfügung:

- Statusbits (SBM Systembereit Meldung, QFL Quitt. Fehler gelöscht, Netzwerk und Bus Informationen)
- Steuern von FL Fehler löschen
- Auslesen von Diagnosemeldungen

Siehe 'STANDARD_CONTROL (FB)' auf Seite 197.

STANDARD_CONTROL _iSA (FB)
(Makro: AflImportBasic - AflImportBasicControl + AflImportStandard - AflImportStandardControl_iSA)
Folgende Grundfunktionalität für die Steuerung mit Einspeisung steht zu Verfügung:

- auslesen der Statusbitinformationen (Netzwerk bereit, Bus bereit, Bus Warnung, Bus Fehler, SBM, QUE und QFL)
- das Steuern von FL
- das Auslesen von Diagnosemeldungen

Siehe 'STANDARD_CONTROL_iSA (FB)' auf Seite 199.

STANDARD_KE (FB)
(Makro: AflImportStandard - AflImportStandardKE)
Folgende Grundfunktionalität für die Einspeisung steht zu Verfügung:

- Statusbits (QUE Quitt. Umrichter Ein, SBM Systembereit Meldung)
- Steuerbits (UE Umrichter Ein, FL Fehler Löschen)
- Ausgabe Diagnosenummer

Siehe 'STANDARD_KE (FB)' auf Seite 201.

STANDARD_AXIS (FB)
STANDARD_IDT4 (FB)
STANDARD_ihX (FB)
STANDARD_KWZ (FB)
(Makro: AflImportBasic - AflImportBasicDrive + AflImportStandard - AflImportStandardAxis)
Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Siehe 'STANDARD_AXIS (FB) (STANDARD_AXIS_KWZ, STANDARD_AXIS_IDT4, STANDARD_AXIS_ihX)' auf Seite 202.

4.1.2.5.1 Erweiterte Standard Achsen

Die erweiterten Achsen verfügen über dieselben Grundfunktionalitäten wie der Funktionsbaustein 'STANDARD_AXIS'.

Mit den Achsen vom Typ 'Winder' kann ein Tänzerwickler oder ein sensorloser Auf- und Abwickler realisiert werden.
Mit der Achse vom Typ 'Position' können Sie eine Druckmarkenregelung oder ablösende Positionierung realisieren.
Mit den Achsen vom Typ 'Printmark' können Sie eine Druckmarkenregelung realisieren.
Es stehen Achsen vom Typ 'Follow' zu Verfügung. Diese folgen synchron einem Sollwert (Master).
Zusätzlich können Tabellen für z. B. Kurvenscheiben–Funktionen verrechnet werden.

STANDARD_AXIS_POSITION_INTERPOSED (FB)
(Makro: AflImportAdvancedPositioning - AflImportAdvancedPositioning)
Erweiterte Funktionalität:

- Ablösende Positionierung

Siehe 'STANDARD_AXIS_POSITION_INTERPOSED (FB)' auf Seite 210.

STANDARD_AXIS_POSITION_TO_THE_MARK (FB)

(Makro: AflImportPrintMarkAxis - AflImportStandardAxisPrintMarkPositionToTheMark)

Erweiterte Funktionalität:

- Druckmarkenregelung

Siehe 'STANDARD_AXIS_POSITION_TO_THE_MARK (FB)' auf Seite 219.

STANDARD_AXIS_PRINTMARK_INSETTER_CONT (FB)

(Makro: AflImportPrintMarkAxis - AflImportStandardAxisPrintMarkInsetterCont)

Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des zu bearbeitenden Bandes wird zu einem Masterformat geregelt (Insetting)

Siehe 'STANDARD_AXIS_PRINTMARK_INSETTER_CONT (FB)' auf Seite 228.

STANDARD_AXIS_PRINTMARK_INSETTER_INTERVAL (FB)

(Makro: AflImportPrintMarkAxis - AflImportStandardAxisPrintMarkInsetterInterval)

Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des zu bearbeitenden Bandes wird zu einem Masterformat geregelt (Insetting)

Siehe 'STANDARD_AXIS_PRINTMARK_INSETTER_INTERVAL (FB)' auf Seite 240.

STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_CONT (FB)

(Makro: AflImportPrintMarkAxis - AflImportStandardAxisPrintMarkRegisterControlCont)

Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des bearbeitenden Aggregats (z.B. Messer, Magnetstanze) wird zu einem Masterformat geregelt (Registerregler)

Siehe 'STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_CONT (FB)' auf Seite 252.

STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_DISCONT (FB)

(Makro: AflImportPrintMarkAxis - AflImportStandardAxisPrintMarkRegisterControlDiscont)

Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des bearbeitenden Aggregats (z.B. Messer, Magnetstanze) wird zu einem Masterformat geregelt (Registerregler)

Siehe 'STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_DISCONT (FB)' auf Seite 264.

STANDARD_FOLLOW_AXIS_GEAR (FB)

(Makro: AflImportFollowAxis - AflImportFollowAxisGear)

Erweiterte Funktionalität:

- Die Slave Achse folgt kontinuierlich einem Master.
- Ein Getriebeverhältnis kann vorgegeben werden.

Siehe 'STANDARD_FOLLOW_AXIS_GEAR (FB)' auf Seite 276.

STANDARD_FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2 (FB)

(Makro: AflImportFollowAxis - AflImportFollowAxisContinuousTable1Table2)

Erweiterte Funktionalität:

- Die Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle.
- Am Tabellenende kann eine Tabellenumschaltung durchgeführt werden.

Siehe 'STANDARD_FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2 (FB)' auf Seite 284.

STANDARD_FOLLOW_AXIS_SWITCHING_TABLE_WITH_OUTGEAR (FB)

(Makro: AflImportFollowAxis - AflImportFollowAxisSwitchingTableWithOutgear)

Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle.
- Am Tabellenende kann eine Tabellenumschaltung durchgeführt werden.
- Die Tabelle kann über ein Getriebeverhältnis angepasste werden (Stauchen und Stecken).

Siehe 'STANDARD_FOLLOW_AXIS_SWITCHING_TABLE_WITH_OUTGEAR (FB)' auf Seite 293.

STANDARD_FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2 (FB)
(Makro: AflImportFollowAxis - AflImportFollowAxisSwitchingTable1Table2)
Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle mit Ein- und Auskuppeltabelle.
- Am Tabellenende kann eine Tabellenumschaltung durchgeführt werden.

Siehe 'STANDARD_FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2 (FB)' auf Seite 301.

STANDARD_FOLLOW_AXIS_TABLE_CONTINUOUS (FB)
(Makro: AflImportFollowAxis - AflImportFollowAxisContinuousTable)
Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle.

Siehe 'STANDARD_FOLLOW_AXIS_TABLE_CONTINUOUS (FB)' auf Seite 311.

STANDARD_FOLLOW_TABLE_ENGAGE_DISENG_AXIS (FB)
(Makro: AflImportFollowAxis - AflImportFollowAxisTableEngageDiseng)
Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle mit Ein- und Auskuppeltabelle.

Siehe 'STANDARD_FOLLOW_TABLE_ENGAGE_DISENG_AXIS (FB)' auf Seite 319.

STANDARD_FOLLOW_TABLE_START_AUTOSTOP_AXIS (FB)
(Makro: AflImportFollowAxis - AflImportFollowAxisTableStartAutoStop)
Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle mit Ein- und Auskuppeltabelle.

Siehe 'STANDARD_FOLLOW_TABLE_START_AUTOSTOP_AXIS (FB)' auf Seite 328.

STANDARD_WINDER_DANCER_AXIS (FB)
(Makro: AflImportStandardWinder - AflImportStandardWinderDancer)
Erweiterte Funktionalität:

- Der Funktionbaustein 'STANDARD_WINDER_TORQUE_AXIS' realisiert einen sensorlosen Auf- und Abwickler und konfigurierbarer Bahnspannung mit folgenden Merkmalen:
- Momentgesteuerter Zentrumswickler
- Betrieb als Auf- und Abwickler
- Durchmesserberechnung
- Durchmessersuche beim Start des Wicklers
- Ermittlung des Rollenträgheitsmoments bei unbekanntem Durchmesser
- Synchronisieren auf eine laufende Bahn

Siehe 'STANDARD_WINDER_DANCER_AXIS (FB)' auf Seite 337.

STANDARD_WINDER_TORQUE_AXIS (FB)
(Macro: AflImportStandardWinder - AflImportStandardWinderTorque)
Erweiterte Funktionalität:

- Der Funktionsbaustein 'STANDARD_WINDER_DANCER_AXIS' realisiert einen Tänzerwickler mit folgenden Merkmalen:
- PID geregelter Tänzerwickler
- Betrieb als Auf- und Abwickler
- Durchmesserberechnung
- Durchmessersuche bei Start des Wicklers

Siehe 'STANDARD_WINDER_TORQUE_AXIS (FB)' auf Seite 347.

4.1.2.5.2 Strukturen

Folgende Zugriffsmöglichkeiten stehen zu Verfügung:

PLC Steuerung (ST_CONTROL):

- Statusbits (SBM Systembereitmeldung, Netzwerkstatus)
- Steuerbits (FL Fehler Löschen, Diagnosemeldung Lesen)
- Diagnosearray (Speicher für 10 Diagnosemeldungen)

Siehe 'ST_CONTROL (ST)' auf Seite 191.

Achse (ST_AXIS_DRIVE):

- Sollwerte (Drehzahl-, Lage-, Momentsollwerte, Vorsteuerwerte)
- Istwerte (Drehzahl-, Lage-, Momentistwert, Messwerte ...)
- Steuerbits (RF Reglerfreigabe, FL Fehler Löschen, Diagnosemeldung Lesen)
- Statusbits (Drehzahlfenster, Stillstandsfenster...)
- Diagnosearray (Speicher für 10 Diagnosemeldungen)
- Parameterliste Lesen
- Konfiguration (PLC Funktionen)

Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.

4.1.3 Beschreibung der Strukturen (ST)

4.1.3.1 ST_CONTROL (ST)

Die Steuerungsstruktur 'ST_CONTROL' fasst die steuerungsspezifischen Daten zusammen.
Sie untergliedert sich in folgende Unterstrukturen:

- ST_PLC_STATUSBITS Siehe 'ST_PLC_STATUSBITS (ST)' auf Seite 191.)
- ST_PLC_CONTROLBITS (Siehe 'ST_PLC_CONTROLBITS (ST)' auf Seite 192.
- ARRAY [1..10] OF ST_AXIS_ERROR Siehe 'arstDiagnosis (ARR_ST)' auf Seite 192.

Speicherort:
CODESYS V2
'Data types' → **'AMK_CONTROL_STRUCTURE'** → **'Interface'**

CODESYS V3
'POUs' → **'Control'** → **'Standard'** → **'Types'** → **'AMK_CONTROL_STRUCTURE'**

4.1.3.1.1 ST_PLC_STATUSBITS (ST)

Die Struktur 'ST_PLC_STATUSBITS' ist eine Unterstruktur der 'ST_CONTROL'.
Über 'ST_CONTROL.ST_PLC_STATUSBITS' kann auf die Statusbits der Steuerung zugegriffen werden.
Inhalt:

Name	Beschreibung	Einheit
boID33029_SBM	Systembereitmeldung	-
boQFL	Quittierung Fehler Löschen	-
boNetworkReady	Netzwerk Bereit	-
boBusReady	Bus Bereit	-
boBusWarning	Buswarnung	-
boBusError	Busfehler	-
boDiagOk	Diagnose abgeschlossen	-

Speicherort:
CODESYS V2
'Data types' → **'AMK_CONTROL_STRUCTURE'**

CODESYS V3
'POUs' → **'Control'** → **'Standard'** → **'Types'** → **'AMK_CONTROL_STRUCTURE'** → **'Interface'**

4.1.3.1.2 ST_PLC_CONTROLBITS (ST)

Die Struktur 'ST_PLC_CONTROLBITS' ist eine Unterstruktur der 'ST_CONTROL'.
Über 'ST_CONTROL.ST_PLC_CONTROLBITS' können die Steuerbits der Steuerung gesetzt werden.
Inhalt:

Name	Beschreibung	Einheit
boID32913_FL	Fehler löschen	-
boDiagnosis	Lesen der Diagnosemeldungen	-

Speicherort:
CODESYS V2
'Data types' → 'AMK_CONTROL_STRUCTURE'

CODESYS V3
'POUs' → 'Control' → 'Standard' → 'Types' → 'AMK_CONTROL_STRUCTURE' → 'Interface'

4.1.3.1.3 arstDiagnosis (ARR_ST)

Das Array 'arstDiagnosis' ist ein Bestandteil der Struktur der 'ST_AXIS_DRIVE' und 'ST_CONTROL'.
Über z.B. 'ST_AXIS_DRIVE.arstDiagnosis [1-10]' kann auf die Diagnosemeldungen zugegriffen werden.
Das Diagnosearray beinhaltet bis zu 10 Diagnosemeldungen die wie folgt aufgebaut sind:

Name	Beschreibung	Einheit
arstDiagnosis [1]	boErr: Fehlerbit für Fehler 1 iDiagNo: Diagnosenummer für Fehler 1 diInfo1: Zusatzinfo 1 für Fehler 1 iAdr: Geräteadresse*	-
arstDiagnosis [2]	boErr: Fehlerbit für Fehler 2 iDiagNo: Diagnosenummer für Fehler 2 diInfo1: Zusatzinfo 1 für Fehler 2 iAdr: Geräteadresse*	-
...	...	
arstDiagnosis [10]	boErr: Fehlerbit für Fehler 10 iDiagNo: Diagnosenummer für Fehler 10 diInfo1: Zusatzinfo 1 für Fehler 10 iAdr: Geräteadresse*	-

* iAdr: Geräteadresse
Eine ACC-Bus Steuerung zeigt die Geräteadresse bei einem Slavefehler an, ansonsten 0 (lokaler Fehler). EtherCAT immer 0.

4.1.3.2 ST_AXIS_DRIVE (ST)

Die Achsstruktur 'ST_AXIS_DRIVE' fasst die achsspezifischen Daten zusammen.
Sie untergliedert sich in folgende Unterstrukturen:

- 'ST_ACTUAL_VALUES' Siehe 'ST_ACTUAL_VALUES (ST)' auf Seite 193.
- 'ST_STATUSBITS' Siehe 'ST_STATUSBITS (ST)' auf Seite 194.
- 'ST_SETPOINTS' Siehe 'ST_SETPOINTS (ST)' auf Seite 195.
- 'ST_CONTROLBITS' Siehe 'ST_CONTROLBITS (ST)' auf Seite 195.
- 'ARRAY [1..10] OF ST_AXIS_ERROR' (Siehe 'arstDiagnosis (ARR_ST)' auf Seite 196.)
- 'ST_IDS' Siehe 'ST_IDS (ST)' auf Seite 196.
- 'ST_PLC_ID_LIST' (vorläufig) Siehe 'ST_PLC_ID_LIST (ST) (vorläufig)' auf Seite 196.

Speicherort:
CODESYS V2
'Data types' → 'AMK_AXIS_STRUCTURE' → 'Interface'

CODESYS V3
'POUs' → 'Control' → 'Standard' → 'Types' → 'AMK_AXIS_STRUCTURE'

Die Struktur 'ST_AXIS_DRIVE' wird nach Freigabe der dazugehörigen STANDARD_ACHSE aktualisiert.

4.1.3.2.1 ST_ACTUAL_VALUES (ST)

Die Struktur 'ST_ACTUAL_VALUES' ist eine Unterstruktur der 'ST_AXIS_DRIVE'.
Über 'ST_AXIS_DRIVE.ST_ACTUAL_VALUES' kann auf die Ist-Werte der Achse zugegriffen werden.
Inhalt:

Name	Beschreibung	Einheit
diID00040_Velocity_feedback_value	aktueller Drehzahlistwert	0,0001 U/min
diID00051_Position_feedback_value	aktueller Lageistwert	Inkremente
diID00084_Torque_feedb_val	aktueller Momentistwert	0,1 % Mn
diID00130_Probe1_val_p_edge	Messwert 1 positive Flanke [1)]	Inkremente
diID00131_Probe1_val_n_edge	Messwert 1 negative Flanke [1)]	Inkremente
diID00189_Following_dist	Schleppfehler [1)]	Inkremente
diRectangularPulsInput_X132	Eingangsimpulse des Rechteckgebereingangs	Inkremente

1) Wirkt nur bei EtherCAT!
Die Messwerte und der Schleppfehler sind im PLC Programmcode der Standard Achse auskommentiert und werden dadurch in der Struktur nicht aktualisiert.
Auskommentierter Teile im PLC Programmcode siehe:
- Deklarationseditor Standard Achse: fbGetStatusActVal
- Programmcode Editor: Auswertung der Variablen 'boErr'
- Programmcode Editor: Auswertung der Variablen 'boErrID'
- Aktion Standard Achse actSync: PLC Programmcode Editor , Abschnitt 'Read status'

Speicherort:
CODESYS V2
'Data types' → 'AMK_AXIS_STRUCTURE' → 'Interface'

CODESYS V3
'POUs' → 'Control' → 'Standard' → 'Types' → 'AMK_AXIS_STRUCTURE' → 'Interface'

4.1.3.2.2 ST_STATUSBITS (ST)

Die Struktur 'ST_STATUSBITS' ist eine Unterstruktur der 'ST_AXIS_DRIVE'.

Über 'ST_AXIS_DRIVE.ST_STATUSBITS' kann auf die Statusbits der Achse zugegriffen werden.

Inhalt:

Name	Beschreibung	Einheit
boID330_VelocityWindow	Code 330 nIst = nSoll ID157 Drehzahlfenster ("Auf Drehzahl")	-
boID331_ZeroVelocityWindow	Code 331 nist < nmin ID124 Stillstandsfenster	-
boID332_VelocityWindowLimit	Code 332 nIst < nx ID125 Drehzahlschwelle nx	-
boID333_TorqueLimit	Code 333 Md>=Mdx ID126 Drehmomentschwelle Mdx	-
boID334_NegTorqueLimit	Code 334 MSoll >= MGrenz Sollmoment Grenzmoment (ID82/ID83)	-
boID335_NegVelocityLimit	335 nSoll >= nGrenz Solldrehzahl . Grenzdrehzahl (ID38/ID39)	-
boID336_InPosition	In Position [1)]	-
boID33029_SBM	Systembereitmeldung	-
boID33030_QUE	Quittierung Umrichter Ein	-
boID33031_QRF	Quittierung Reglerfreigabe	-
boID33036_RFP_known	Code 33036 Referenzpunkt bekannt	-
boID33074_Warning_active	Code 33074 Warnung aktiv Sammelwarnung	-
boDiagOk	Diagnose abgeschlossen	-
boSetPosEnabAck	Nebenbetriebsart1 Freigabe Positionsvorgabe	-
boSetVelocityEnabAck	Nebenbetriebsart2 Freigabe Drehzahl	-
boSetTorqueEnabAck	Nebenbetriebsart3 Freigabe Momentvorgabe	-
boResidDistDone	Restweg gelöscht	-

1) Das 'In Positions' Bit wird nur bei Reglerkarten mit internem Interpolator ausgewertet. Bei Ethernet Reglerkarten wirkt das Bit nur bei den Kommandos absolute und relative Positionierung (boPosAbs / boPosRel)

Speicherort:

CODESYS V2

'Data types' → 'AMK_AXIS_STRUCTURE' → 'Interface'

CODESYS V3

'POUs' → 'Control' → 'Standard' → 'Types' → 'AMK_AXIS_STRUCTURE' → 'Interface'

4.1.3.2.3 ST_SETPOINTS (ST)

Die Struktur 'ST_SETPOINTS' ist eine Unterstruktur der 'ST_AXIS_DRIVE'.

Über 'ST_AXIS_DRIVE.ST_SETPOINTS' können Sollwerte in die Achse geschrieben werden.

Inhalt:

Name	Beschreibung	Einheit
diID00036_Velocity_command_value	Drehzahlsollwert (ID36)	0,0001 U/min
diID00037_Velocity_command_value_addit	Drehzahlvorsteuerwert (ID37) 1)	0,0001 U/min
diID00047_Position_command_value	Lagesollwert (ID47)	Inkremente
diID00080_Torque_command_value	Momentsollwert (ID80)	0,1 % Mn
diID00081_Torque_command_value_additiv	Momentvorsteuerwert (ID81) 1)	0,1 % Mn

1) Die Vorsteuerwerte sind im PLC Programmcode der Standard Achse auskommentiert und werden dadurch in der Struktur nicht aktualisiert.

   Auskommentierter Teile im PLC Programmcode siehe:

   - Deklarationseditor Standard Achse: fbMotion
   - Aktion Standard Achse actSync: PLC Programmcode Editor , Abschnitt 'Writing the setpoint values'

Speicherort:

CODESYS V2

'Data types' → **'AMK_AXIS_STRUCTURE'** → **'Interface'**

CODESYS V3

'POUs' → **'Control'** → **'Standard'** → **'Types'** → **'AMK_AXIS_STRUCTURE'** → **'Interface'**

4.1.3.2.4 ST_CONTROLBITS (ST)

Die Struktur 'ST_CONTROLBITS' ist eine Unterstruktur der 'ST_AXIS_DRIVE'.

Über 'ST_AXIS_DRIVE.ST_CONTROLBITS' können die Steuerbits der Achse gesetzt werden.

Inhalt:

Name	Beschreibung	Einheit
boID32904_RF	Reglerfreigabe ein	-
boID32913_FL	Fehler löschen	-
enSetPointMode	Mode der Sollwertvorgabe Typ EN_SETPOINT AMK_SETPOINT_VELOCITY Drehzahlvorgabe AMK_SETPOINT_POSITION Positionsvorgabe AMK_SETPOINT_TORQUE Momentvorgabe	-
boDiagnosis	Lesen der Diagnosemeldungen	

Speicherort:

CODESYS V2

'Data types' → **'AMK_AXIS_STRUCTURE'** → **'Interface'**

CODESYS V3

'POUs' → **'Control'** → **'Standard'** → **'Types'** → **'AMK_AXIS_STRUCTURE'** → **'Interface'**

4.1.3.2.5 ST_IDS (ST)

Die Struktur 'ST_IDS' ist eine Unterstruktur der 'ST_AXIS_DRIVE'.
Über 'ST_AXIS_DRIVE.ST_IDS' kann auf Parameterwerte zugegriffen werden.
Inhalt:

Name	Beschreibung	Einheit
diID00002_SERCOS_cycle	SERCOS Zykluszeit	0,001ms
diID00038_Pos_velocity_limit	pos.Drehzahlgrenze	0,0001 U/min
diID00039_Neg_velocity_limit	neg.Drehzahlgrenze	0,0001 U/min
diID00049_Positive_position_limit	Lagegrenzwert positiv	Inkremente
diID00050_Negative_position_limit	Lagegrenzwert negativ	Inkremente
diID00057_In_posit_window	Positionsfenster	Inkremente
diID00116_Resol_mot_encod	Auflösung Motorgeber	Inkremente
diID00228_Synchr_pos_window	Winkelsynchron Fenster	Inkremente
diID32771_Nom_torque	Nenndrehmoment Mn	0,1 % Mn

Speicherort:
CODESYS V2
'Data types' → **'AMK_AXIS_STRUCTURE'** → **'IDs'**

CODESYS V3
'POUs' → **'Control'** → **'Standard'** → **'Types'** → **'AMK_AXIS_STRUCTURE'** → **'IDs'**

4.1.3.2.6 ST_PLC_ID_LIST (ST) (vorläufig)

Die Struktur 'ST_PLC_ID_LIST' ist eine Unterstruktur der 'ST_AXIS_DRIVE'.
Über 'ST_AXIS_DRIVE.ST_PLC_ID_LIST' werden die benötigten PLC Parameter für die Standard Achsen gesetzt.
Inhalt:

Name	Beschreibung	Einheit
diID32919_00_Plc_List	PLC-Parameterliste	-
diID32919_01_maximum	Parameterlistenlänge	-
diID32919_02_Resid_dist_wind	neg. Fenster Restweg Löschen Standardwert: 20000	Inkremente
diID32919_03_Sync_set_pulses_multip	Sollwert Multiplikator Dient der Anpassung an die mechanischen Verhältnisse der Anlage Standardwert: 10000	Inkremente
diID32919_04_Sync_set_pulses_divider	Sollwert Teiler Dient der Anpassung an die mechanischen Verhältnisse der Anlage Standardwert: 10000	Inkremente
diID32919_05_diEmergency_StopRamp	Notstopprampe	ms

Speicherort:
CODESYS V2
'Data types' → **'AMK_AXIS_STRUCTURE'** → **'IDs'**

CODESYS V3
'POUs' → **'Control'** → **'Standard'** → **'Types'** → **'AMK_AXIS_STRUCTURE'** → **'IDs'**

4.1.3.2.7 arstDiagnosis (ARR_ST)

Das Array 'arstDiagnosis' ist ein Bestandteil der Struktur der 'ST_AXIS_DRIVE' und 'ST_CONTROL'.
Über z.B. 'ST_AXIS_DRIVE.arstDiagnosis [1-10]' kann auf die Diagnosemeldungen zugegriffen werden.
Das Diagnosearray beinhaltet bis zu 10 Diagnosemeldungen die wie folgt aufgebaut sind:

Name	Beschreibung	Einheit
arstDiagnosis [1]	boErr: Fehlerbit für Fehler 1 iDiagNo: Diagnosenummer für Fehler 1 diInfo1: Zusatzinfo 1 für Fehler 1 iAdr: Geräteadresse*	-
arstDiagnosis [2]	boErr: Fehlerbit für Fehler 2 iDiagNo: Diagnosenummer für Fehler 2 diInfo1: Zusatzinfo 1 für Fehler 2 iAdr: Geräteadresse*	-
...	...	
arstDiagnosis [10]	boErr: Fehlerbit für Fehler 10 iDiagNo: Diagnosenummer für Fehler 10 diInfo1: Zusatzinfo 1 für Fehler 10 iAdr: Geräteadresse*	-

* iAdr: Geräteadresse
Eine ACC-Bus Steuerung zeigt die Geräteadresse bei einem Slavefehler an, ansonsten 0 (lokaler Fehler). EtherCAT immer 0.

4.1.4 Beschreibung der Standardbausteine (FB)

4.1.4.1 Einspeisung und PLC

4.1.4.1.1 STANDARD_CONTROL (FB)

Folgende Grundfunktionalität für die Steuerung steht zu Verfügung:

- auslesen der Statusbitinformationen (Netzwerk bereit, Bus bereit, Bus Warnung, Bus Fehler, SBM und QFL) aus der AMK-Steuerung
- das Steuern von FL
- das Auslesen von Diagnosemeldungen

Anwender Interface

		STANDARD_CONTROL			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Fehler löschen	-	boErrorReset	boSBM	-	Sammelbereitmeldung
Bus Instanz	-	uiBusInstance	boNetworkReady	-	Netzwerk bereit
			boBusReady	-	Bus bereit
			boBusWarning	-	Bus Warnung
			boBusError	-	Bus Fehler
			boErr	-	Fehlersignal
			iErrID	-	Fehler Identnummer
			enErrName	-	Name fehlerh Baustein
Steuerungsstruktur	-	stControl	stControl	-	Steuerungsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boErrorReset	BOOL	Error Reset (FL = Fehler löschen)		x
uiBusInstance	UINT	Bus Instanz; Standardwert: 5		x

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boNetworkReady	BOOL	Netzwerk bereit		x
boBusReady	BOOL	Bus bereit (Modul, Netzwerk bereit, keine Fehler und Warnungen, Bus Operational Mode)		x
boBusWarning	BOOL	Bus Warnung		x
boBusError	BOOL	Bus Fehler		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stControl	STRUCT	ST_CONTROL Siehe 'ST_CONTROL (ST)' auf Seite 191.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_CONTROL	-

4.1.4.1.2 STANDARD_CONTROL_iSA (FB)

Folgende Grundfunktionalität für die Steuerung mit Einspeisung steht zu Verfügung:

- auslesen der Statusbitinformationen (Netzwerk bereit, Bus bereit, Bus Warnung, Bus Fehler, SBM, QUE und QFL)
- das Steuern von FL
- das Auslesen von Diagnosemeldungen

Fehlermeldung bei der automatischen Konfigurationserstellung: DEV_GET_ERROR_ID11
Ursache: iSA Beschreibungsdatei enthält nicht die erforderlichen Einträgt.
Ab AIPEX PRO Version ≥ 3.03 2015/41 Teile-Nr.205890 integriert.

Anwender Interface

		STANDARD_CONTROL_iSA			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Fehler löschen	-	boErrorReset	boSBM	-	Sammelbereitmeldung
Bus Instanz	-	uiBusInstance	boQUE	-	Quittierung "Umrichter Ein"
			wErrorID11	-	Status ID11 'Zustandsklasse 1-Fehler' *
			boErrTemperature	-	Fehler Temperatur Gerät *
			boErrDC_overvoltage	-	Überspg. Zwischenkreis *
			boNetworkReady	-	Netzwerk bereit
			boBusReady	-	Bus bereit
			boBusWarning	-	Bus Warnung
			boBusError	-	Bus Fehler
			boErr	-	Fehlersignal
			iErrID	-	Fehler Identnummer
			enErrName	-	Name fehlerh Baustein
Steuerungsstruktur	-	stControl	stControl	-	Steuerungsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boErrorReset	BOOL	Error Reset (FL = Fehler löschen)		x
uiBusInstance	UINT	Bus Instanz; Standardwert: 5		x

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boQUE	BOOL	Quittierung Umrichter Ein (QUE)		x
wErrorID11	WORD	Status ID11 'Zustandsklasse 1-Fehler' *		x
boErrTemperature	BOOL	Fehler Temperatur Gerät *		x
boErrDC_overvoltage	BOOL	Überspannung Zwischenkreis *		x
boNetworkReady	BOOL	Netzwerk bereit		x
boBusReady	BOOL	Bus bereit (Modul, Netzwerk bereit, keine Fehler und Warnungen, Bus Operational Mode)		x
boBusWarning	BOOL	Bus Warnung		x
boBusError	BOOL	Bus Fehler		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stControl	STRUCT	ST_CONTROL Siehe 'ST_CONTROL (ST)' auf Seite 191.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_CONTROL_iSA	-

*

Verhalten iSA:

Keine selbständige Reaktion auf einen Temperatur- oder Überspannungsfehler.

Der Anwender muss über das PLC-Programm geeignete Maßnahmen einleiten, z. B. Motoren Momentlos schalten (STO).

4.1.4.1.3 STANDARD_KE (FB)

Folgende Grundfunktionalität für die Einspeisung steht zu Verfügung:

- auslesen der Statusbitinformationen (QUE, SBM) aus der AMK-Einspeisung
- das Steuern von UE und FL
- das Auslesen der Diagnosenummer

Der Funktionsblock 'STANDARD_KE' beinhaltet eine Sperrzeit (AUS / AN) für das Steuersignal 'boUE' Umrichter Ein.
Baustein 'fbTof', Standardwert: 15 Sekunden.

Anwender Interface

		STANDARD_KE			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Umrichter Ein	-	boUE	uiDiagnosticNr	-	Diagnosenummer
Fehler löschen	-	boFL	boQUE	-	Quittierung "Umrichter Ein"
			boSBM	-	Sammelbereitmeldung
			boErr	-	Fehlersignal
			iErrID	-	Fehler Identnummer
			enErrName	-	Name fehlerh Baustein
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boUE	BOOL	Bit zum Einschalten des Umrichters (Zwischenkreisspannung Ein). Die entsprechende Rückmeldung QUE ist Vorbedingung für RF. Relevante Parameter: ID32795 'Quelle Umrichter Ein' Standardwert: 0 (UE über binären Eingang) Einstellung: z.B. Code 5 (UE über Feldbus)		x
boFL	BOOL	Bit zum Fehlerlöschen		x

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
uiDiagnosticNr	UINT	ID390 'Diagnosenummer'		x
boQUE	BOOL	Quittierung Umrichter Ein (QUE)		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_KE	-

4.1.4.2 Standard Achse

4.1.4.2.1 STANDARD_AXIS (FB) (STANDARD_AXIS_KWZ, STANDARD_AXIS_IDT4, STANDARD_AXIS_ihX)

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Anwender Interface

		STANDARD_AXIS			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Positioniergeschwindigkeit	-	udPosVelocity			
Zielposition	-	diPosition			
Beschleunigung	-	lreAccel			
Verzögerung	-	lreDecel			
Ruckbegrenzung beim Start	-	lreJerkStart			
Ruckbegrenzung beim Stopp	-	lreJerkEnd			
Tippgeschwindigkeit	-	udVelocityJog			
Drehzahl für Drehzahlregelung	-	diSpeedVelocity			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Achsstruktur	-	stAxisDrive	stAxisDrive	-	Achsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

		actSync		
	-	synchrone Aktion - Aufruf im synchronen Programmteil (FPLC_PRG)	-	
	-		-	

1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x

Name	Typ	Beschreibung	Sync.	Async.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_ Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung Busfehler wenn 'boErr' = TRUE: EtherCAT oder CAN nicht in Operation Mode.	x	
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS	STANDARD_AXIS.actSync

4.1.4.2.2 Modifizierte 'Standard Achsen' für ihX Antriebe

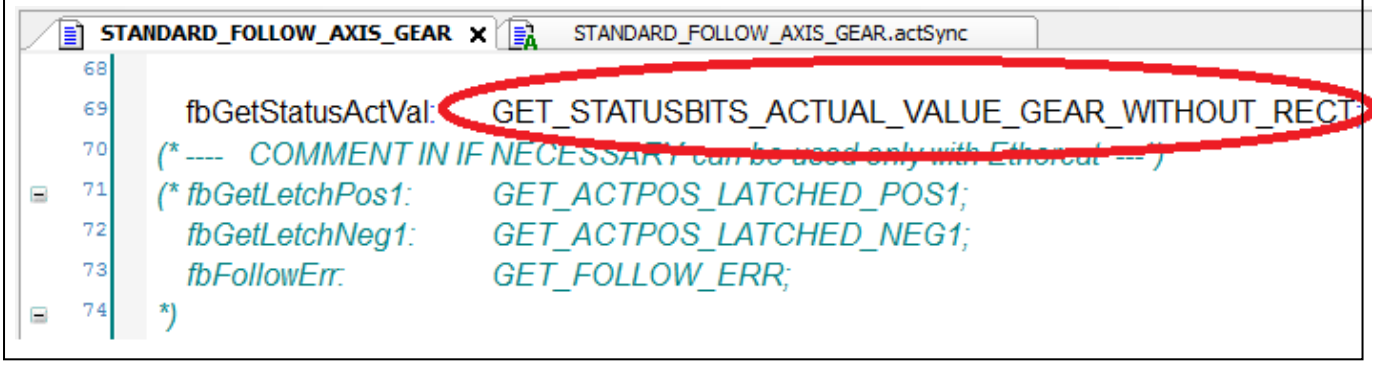
Alle erweiterten Standard Achsen können für die ihX Antriebe verwendet werden.

Der ihX verfügt über keinen Rechteckimpulsgebereingangs. Deshalb muss die Auswertung des Rechteckimpulsgebereingangs deaktiviert werden.

Über die Deklaration des Funktionsbausteines 'fbGetStatusActVal' kann der Anwender entscheiden ob der Rechteckimpulsgebereingang ausgewertet wird oder nicht.

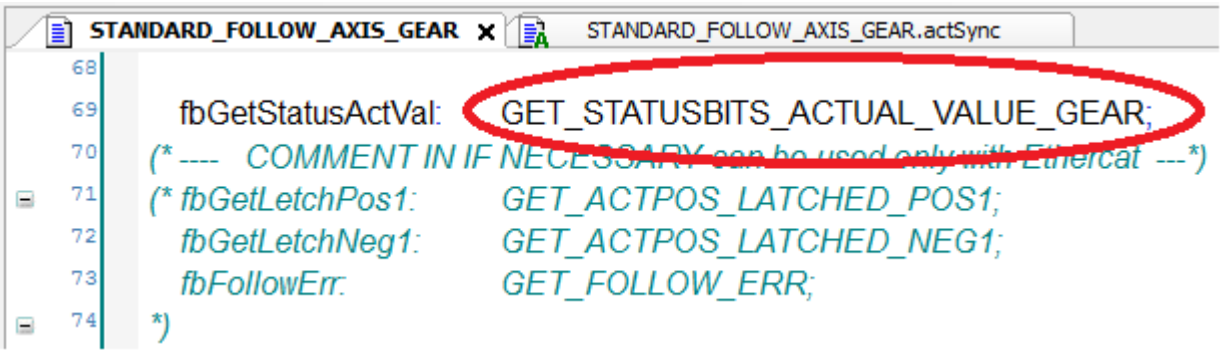
Ohne Rechteckimpulsgebereingang
Deklaration für fbGetStatusActVal:
GET_STATUSBITS_ACTUAL_VALUE_GEAR**_WITHOUT_RECT**

Beispiel:
FB STANDARD_FOLLOW_AXIS_GEAR (manuell abgewandelt für ihX)



Mit Rechteckimpulsgebereingang
Deklaration für fbGetStatusActVal:
GET_STATUSBITS_ACTUAL_VALUE_GEAR

Beispiel:
FB STANDARD_FOLLOW_AXIS_GEAR (original)



4.1.4.2.3 STANDARD_AXIS_POSITION_INTERPOSED (FB)

Erweiterte Funktionalität:

- Ablösende Positionierung

Siehe 'POSITION_RELATIV_INTERPOSED (FB)' auf Seite 386.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Anwender Interface

		STANDARD_AXIS_POSITION_INTERPOSED			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Positioniergeschwindigkeit	-	udPosVelocity	**boInPos_Interpose**	-	In Positions Bit
Zielposition	-	diPosition	**boDone_Interpose**	-	Positionierung beendet
Beschleunigung	-	lreAccel	**boBusy_Interpose**	-	Bewegung läuft
Beschleunigung	-	lreDecel			
Ruckbegrenzung Start	-	lreJerkStart			
Ruckbegrenzung Stopp	-	lreJerkEnd			
Tippgeschwindigkeit	-	udVelocityJog			
Drehzahl für Drehzahlr.	-	diSpeedVelocity			
Start Bewegung	-	**boExec_Interpose**			
Start ablösenden Pos.	-	**boInterpose_Interpose**			
Relativer Positionssollwert	-	**diPosition_Interpose**			
Korrektur-Offset	-	**diOffset_Interpose**			
Beschleunigung	-	**udAccel_Interpose**			
Verzögerung	-	**udDecel_Interpose**			
Geschwindigkeit	-	**udVelocity_Interpose**			
Positionsfenster	-	**diPosWindow_Interpose**			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Achsstruktur	-	stAxisDrive	stAxisDrive	-	Achsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

	actSync	
-	synchrone Aktion - Aufruf im synchronen Programmteil (FPLC_PRG)	-
-		-

1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x

Name	Typ	Beschreibung	Sync.	Async.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_ Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x

Name	Typ	Beschreibung	Sync.	Async.
boExec_ Interpose	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die kontinuierliche Bewegung der Achse bis mit einer positiven Flanke am Eingang 'boInterpose_Interpose' die ablösende Positionierung gestartet wird.	x	
boInterpose_ Interpose	BOOL	Startet die ablösende Positionierung (Sollwert diPosition_Interpose)	x	
diPosition_ Interpose	DINT	Relative Zielposition (nach boInterpose) Einheit: Inkr/s Standardwert: 60000	x	
diOffset_ Interpose	DINT	Korrektur der Sollposition [Inkremente] Einheit: Inkr/s Standardwert: 0	x	
udAccel_ Interpose	UDINT	Beschleunigung, mit der auf die Sollgeschwindigkeit gefahren wird. Einheit: Inkr/s2 Standardwert: 1000000	x	
udDecel_ Interpose	UDINT	Verzögerung, mit der eine niedrigere Sollgeschwindigkeit angefahren wird] Einheit: Inkr/s^2 Standardwert: 1000000	x	
udVelocity_ Interpose	UDINT	Geschwindigkeit, mit der die Positionierung ausgeführt wird Einheit: Inkr/s Standardwert: 34133	x	
diPosWindow_ Interpose	DINT	Fenster für die Meldung 'In Position' Einheit: Inkr/s Standardwert: 1000	x	

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boInPos_Interpose	BOOL	Meldung „InPosition“: Aktuelle Position = Zielposition innerhalb des Positionsfensters (diPosWindow_Interpose)	x	
boDone_Interpose	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde	x	
boBusy_Interpose	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.	x	

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS_POSITION_INTERPOSED	STANDARD_AXIS_POSITION_INTERPOSED.actSync

4.1.4.2.4 STANDARD_AXIS_POSITION_TO_THE_MARK (FB)

Erweiterte Funktionalität:

- Druckmarkenregelung

Siehe 'PRINT_MARK_POSITION_TO_THE_MARK (FB)' auf Seite 529.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Funktionsbeschreibung DM
Abkürzung: DM Druckmarke engl. PM printmark

Der Funktionsbaustein 'STANDARD_AXIS_POSITION_TO_THE_MARK' führt eine relative diskontinuierliche Druckmarkenregelung aus.

Der FB berechnet aus

- den Beschleunigungs- und Verzögerungswerten **a** (udAccelPM und udDecelPM [s^2])
- dem Verfahrweg **s** (diFormatPM [inkremente]) + Offset [Inkremente]
- und der Zeit **t**, in der die Positionierung abgeschlossen sein muss (udTimeForPosPM [ms])

ein Bewegungsprofil.

Anwender Interface

		STANDARD_AXIS_POSITION_TO_THE_MARK			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	boPosBusy	-	Positionierung aktiv
Absolute Pos. starten	-	boPosAbs	boJogBusy	-	Tippbetrieb aktiv
Relative Pos. starten	-	boPosRel	boHomeBusy	-	Referenzfahrt aktiv
Tippen Plus	-	boJogPlus	boSpeedBusy	-	Drehzahlregelung aktiv
Tippen Minus	-	boJogMinus	**boRefPuls**	-	Druckmarkenimpulse
Positioniergeschwindigkeit	-	udPosVelocity	**diDeviationPM**	-	DM Versatz
Zielposition	-	diPosition	**usiNoPmLost**	-	Anzahl verlorene DM
Beschleunigung	-	lreAccel			
Verzögerung	-	lreDecel			
Tippgeschwindigkeit	-	udVelocityJog			
Drehzahl für Drehzahlr.	-	diSpeedVelocity			
Freigabe DM	-	**boEnablePM**			
Start DM	-	**boStartPM**			
Referenzfahrt (DM suchen)	-	**boHomePM**			
Abstand zweier DM	-	**diFormatPM**			
Offsetverschiebung der DM	-	**diOffsetPM**			
Geschw. für Marke suchen	-	**diSpeedHomePM**			
DM-Fenster +/-	-	**diWindowPM**			
Beschleunigungswert	-	**udAccelPM**			
Verzögerungswert	-	**udDecelPM**			
Max. Korrekturwert	-	**diMaxCorrection**			
Max. Positionierzeit	-	**udTimeForPosPM**			
Anz. verl. Marken bis Fehler	-	**usiNoLostPM**			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Achsstruktur	-	stAxisDrive	stAxisDrive	-	Achsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

		actSync		
	-	synchrone Aktion - Aufruf im synchronen Programmteil (FPLC_PRG)	-	
	-		-	

1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x

Name	Typ	Beschreibung	Sync.	Async.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList.diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x

Name	Typ	Beschreibung	Sync.	Async.
boEnablePM	BOOL	Aktiviert den BS. Die eingegebenen Werte werden überprüft, das Bewegungsprofil wird berechnet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] ID32980 'Port 3 Bit 2'Code 401 ID32948 'Meldung 4x32' Code 24h ID169 'Messzyklus Parameter'Code 1h Relevante Variablen: Sollwertvorgabe: diInVal Tabellenauswahl: boEnabOpTab2 Pointer auf Tabelle 1: pstOpTab1 Pointer auf Tabelle 2: pstOpTab2		x
boStartPM	BOOL	Start der DM - Positionierung Relevante Parameter: Abstand zwischen zwei DM: diFormatPM Weg nach DM : diOffsetPM Fenster DM: diWindowPM Beschleunigung: udAccelPM Verzögerung: udDecelPM Max. Positionierzeit: iudTimeForPosPM Max. Korrekturwert: diMaxCorrection Max. verlorene DM: usiNoLostPM	x	
boHomePM	BOOL	Start der Referenzfahrt, die Achse fährt auf DM + Offset Relevante Variablen: Referenzfahrgeschwindigkeit: diSpeedHomePM Offset: diOffsetPM	x	
diFormatPM	DINT	Abstand zwischen zwei DM Einheit: Inkremente Standardwert: 20480	x	
diOffsetPM	DINT	Weg nach DM (Abstand DM-Sensor und Stoppposition) Einheit: Inkremente Standardwert: 10240	x	
diSpeedHomePM	DINT	Geschwindigkeit der DM-Referenzfahrt Einheit: Inkremente/s Standardwert: 10000	x	
diWindowPM	DINT	Fenster in der die Marken zum Erkennen liegen muss (+/- Zielposition) Einheit: Inkremente Standardwert: 20480	x	
udAccelPM	UDINT	DM-Beschleunigung Einheit: Inkremente/s² Standardwert: 1000000	x	

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>udDecelPM</td><td>UDINT</td><td>DM-Verzögerung
Einheit: Inkremente/s²
Standardwert: 1000000</td><td>x</td><td></td></tr>
<tr><td>udTimeForPosPM</td><td>UDINT</td><td>Zeit in der Positionierung abgeschlossen sein muss.
Einheit: ms
Standardwert: 2000</td><td>x</td><td></td></tr>
<tr><td>diMaxCorrection</td><td>DINT</td><td>Maximaler Korrekturwert
Vorgabe 0 = Der ermittelter Korrekturwert wird immer ohne Begrenzung verrechnet.
Einheit: Inkremente
Standardwert: 20480</td><td>x</td><td></td></tr>
<tr><td>usiNoLostPM</td><td>USINT</td><td>Sobald die vorgegebene Anzahl an Marken verloren geht wird eine Fehlermeldung generiert.
Vorgabe 0 = Fehlermeldung wird nie generiert
Einheit: Inkremente
Standardwert: 10</td><td>x</td><td></td></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde		x
boRefPuls	BOOL	Für einen Zyklus lang boRefPuls = TRUE, wenn sich der Latchwert geändert hat.	x	
diDeviationPM	DINT	Abweichung zwischen Ist- und Soll-Position der zuletzt gefundenen DM. Einheit: Inkremente	x	
usiNoPmLost	USINT	Anzahl der verlorenen Druckmarken	x	

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS	STANDARD_AXIS.actSync

4.1.4.2.5 STANDARD_AXIS_PRINTMARK_INSETTER_CONT (FB)

Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des zu bearbeitenden Bandes wird zu einem Masterformat geregelt (Insetting)

Siehe 'PRINT_MARK_INSETTER_CONT (FB)' auf Seite 515.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

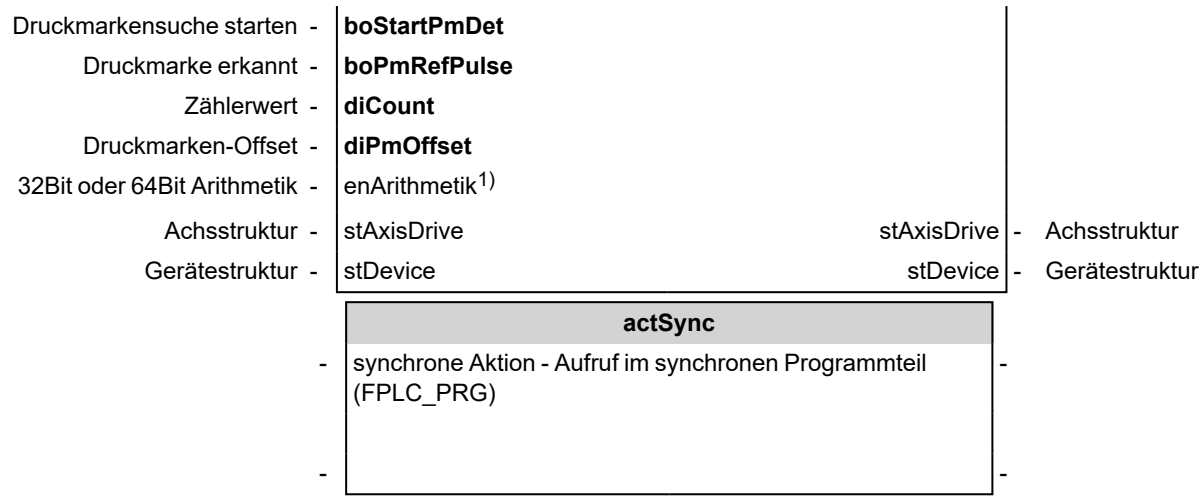
- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Anwender Interface

	STANDARD_AXIS_PRINTMARK_INSETTER_CONT		
Freigabesignal -	boEnable	boEnabAck	- Quittierung "Freigabe"
Reglerfreigabe -	boRF	boSBM	- Sammelbereitmeldung
Fehler löschen -	boFL	boDone	- Quittierung "Ausgeführt"
Not-Halt -	boEmergency_Stop	boErr	- Fehlersignal
Drehzahlregelung starten -	boSpeed	iErrID	- Fehler Identnummer
Referenzfahrt starten -	boHome	enErrName	- Name fehlerh Baustein
Absolute Pos. starten -	boPosAbs	boPosBusy	- Positionierung aktiv
Relative Pos. starten -	boPosRel	boJogBusy	- Tippbetrieb aktiv
Tippen Plus -	boJogPlus	boHomeBusy	- Referenzfahrt aktiv
Tippen Minus -	boJogMinus	boSpeedBusy	- Drehzahlregelung aktiv
Positioniergeschwindigkeit -	udPosVelocity	**boEnabAckPmInsetterCont**	- Druckmarkenmaustein initialisiert und aktiviert
Zielposition -	diPosition	**boParameterSetupDone**	- Parametrierung abgeschlossen
Beschleunigung -	lreAccel	**boPmDetected**	- Marke gefunden
Verzögerung -	lreDecel	**boPmLost**	- Marke verloren
Ruckbegrenzung beim Start -	lreJerkStart	**boPmWnd**	- Markenfenster aktiv
Ruckbegrenzung beim Stopp -	lreJerkEnd	**diPmDeviation**	- Abweichung der Soll-Ist Position
Tippgeschwindigkeit -	udVelocityJog	**diOpOffsetOut**	- Ausgabe Operatoroffset
Drehzahl für Drehzahlr. -	diSpeedVelocity	**iActPmLost**	- Anzahl verlorener Druckmarken
Freigabesignal Druckmarkenregelung -	**boEnablePmInsetterCont**		
Ausgang 'boPmLost' rücksetzen -	**boResetPmLost**		
Ausgabe Operatoroffset starten -	**boSetupOffset**		
Druckmarkensollposition starten -	**boSetupPmPos**		
Master zählt negativ -	**boMasterNegDir**		
Motorrichtung negativ -	**boMotorNegDir**		
Freigabe Korrekturausgabe -	**boCorrOutControlOn**		
Offset Operator -	**diOpOffset**		
Formatlänge -	**diFormatLength**		
Fensterlänge -	**diWindowLength**		
Anzahl Marken bis Meldung "Marke verloren" -	**iNoPmLost**		
Korrekturbereich -	**diCorrectionRange**		
Sollposition DM -	**diPmSetupPos**		
Maximaler Korrekturwert -	**diCorrectionLimit**		
Korrekturbereich 1 starten -	**diCorrRng1Start**		
Korrekturbereich 1 stoppen -	**diCorrRng1End**		
Offset der Korrektursteuerung 1 -	**diCorrOutControl1PreSet**		
Korrekturbereich 2 starten -	**diCorrRng2Start**		
Korrekturbereich 2 stoppen -	**diCorrRng2End**		
Offset der Korrektursteuerung 2 -	**diCorrOutControl2PreSet**		



1) Gilt nur für CODESYS V3

Eingangsvariablen

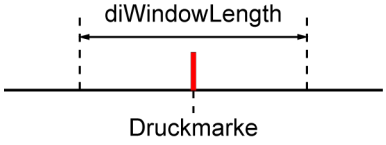
Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_ PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

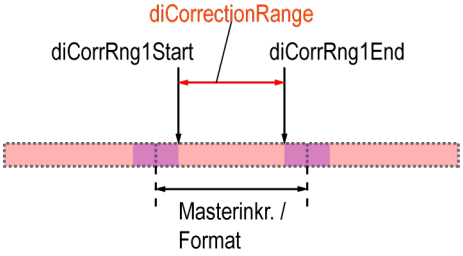
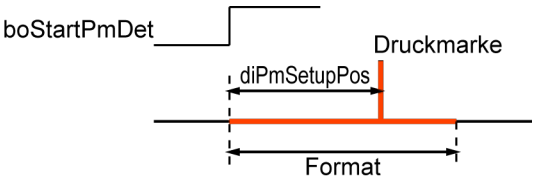
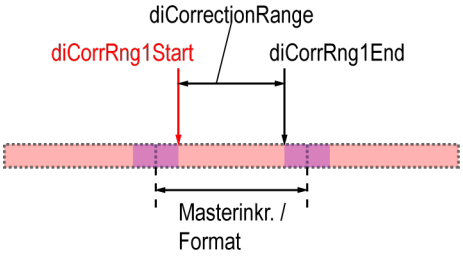
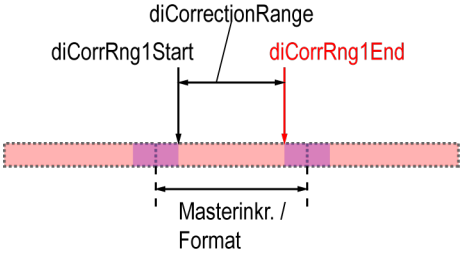
Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x

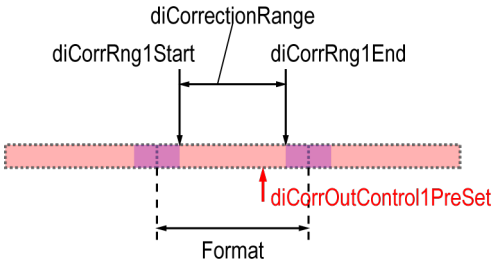
Name	Typ	Beschreibung	Sync.	Async.
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x

Name	Typ	Beschreibung	Sync.	Async.
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x

Name	Typ	Beschreibung	Sync.	Async.
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x
boEnablePmInsetterCont	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' = TRUE ist, bleibt der Baustein freigegeben und wird von der PLC bearbeitet. Die Eingangsinkrementerfassung ist aktiviert. Inkrementänderungen am Eingang 'diInVal' führen zu einer Änderung des Masterbezugspunkts. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boResetPmLost	BOOL	Rücksetzen des Ausgangs 'boPmLost'		x
boSetupOffset	BOOL	Ausgabestart des angegebenen Werts von 'diOpOffset'. Dieser Wert dient als Offset zur Verschiebung der Druckmarken-Sollposition		x
boSetupPmPos	BOOL	Steuerung der Sollpositionsbestimmung der Druckmarke bei positiver Flanke von 'boStartPmDet' 'boPmSetupPos' = FALSE: Es wird die Position der nächsten Druckmarke als Soll-Position erfasst und alle nachfolgenden Marken auf diese Position geregelt. 'boPmSetupPos' = TRUE: Der Parameter 'diPmSetupPos' gibt die Soll-Position der Druckmarke relativ zur aktuellen Masterposition an. Die folgenden Druckmarken werden auf diese Position geregelt.		x

Name	Typ	Beschreibung	Sync.	Async.
boMasterNegDir	BOOL	Negation Master-Impuls-Richtung 'boMasterNegDir' = FALSE: keine Negation 'boMasterNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.		x
boMotorNegDir	BOOL	Negation Motor-Drehrichtung 'boMotorNegDir' = FALSE: keine Negation 'boMotorNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.		x
boCorrOutControlOn	BOOL	Korrekturausgabe-Steuerung aktiv. 'boCorrOutControlOn' = FALSE: Korrekturausgabe immer aktiv 'boCorrOutControlOn' = TRUE: Korrekturausgabe erfolgt nur in dem durch die Variablen 'diCorrRng1Start' und 'diCorrRng1End' bzw. 'diCorrRng2Start' und 'diCorrRng2End' festgelegten Bereich. Beide Bereiche sind mit ODER-Funktion verknüpft. Die Bereichszählung (Modulozählung) wird mit der 0→1 Flanke an 'boCorrOutControlOn' gestartet und lässt sich durch einen Wert an 'diCorrOutControl1PreSet' bzw. 'diCorrOutControl2PreSet' entsprechend einstellen.		x
diOpOffset	DINT	Operator-Offset. Verschiebt die Sollposition der Druckmarke. Die Verschiebung erfolgt im Korrekturbereich und wird mit einer 0→1 Flanke an 'boSetupOffset' gestartet. Sie ist unabhängig von Wertänderungen an 'diCount'. Die Verschiebegeschwindigkeit berechnet sich aus		x
diFormatLength	DINT	Formatlänge Soll-Abstand zweier Druckmarken		x
diWindowLength	DINT	Fensterlänge Länge vor und nach der Soll-Druckmarkenposition. In diesem Bereich ist der Druckmarkensensor aktiviert. Eine Druckmarke wird akzeptiert.  		x
iNoPmLost	INT	Anzahl Druckmarken, die nacheinander fehlen dürfen, bevor der Ausgang 'boPMLost' (Marke verloren) gesetzt wird.		x

Name	Typ	Beschreibung	Sync.	Async.
diCorrectionRange	DINT	Korrekturbereich Die Druckmarkenregelung versucht die Korrektur im angegebenen Bereich auszugeben. [% diFormatLength] 		x
diPmSetupPos	DINT	Der Parameter gibt die Sollposition der Druckmarke zur aktuellen Masterposition bei positiver Flanke von 'boStartPmDet' an. Die erkannten Druckmarken werden auf diese Position geregelt. [Inkremente] 		x
diCorrectionLimit	DINT	Maximale Korrekturausgabe pro Format [Inkremente] 'diCorrectionLimit' = 0: keine Begrenzung 'diCorrectionLimit' > 0: maximaler Wert		x
diCorrRng1Start **diCorrRng2Start**	DINT	Korrekturbereich 1 / 2 Startwert bezogen auf Masterinkremente / Format [Inkremente] 		x
diCorrRng1End **diCorrRng2End**	DINT	Korrekturbereich 1 / 2 Endwert bezogen auf Masterinkremente / Format [Inkremente] 		x

Name	Typ	Beschreibung	Sync.	Async.
diCorrOutControl1PreSet **diCorrOutControl2PreSet**	DINT	Positionswert (Nullverschiebung) bei Start der Korrektursteuerung bezogen auf Masterinkremente / Format [Inkremente] Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn' >Format/2< einzustellen. 		x
boStartPmDet	BOOL	Start der Druckmarkenregelung und Bestimmung der Druckmarken-Sollposition durch Parameter 'diPmSetupPos' wenn 'boSetupPmPos' = TRUE	x	
boPmRefPulse	BOOL	Markenimpuls (Referenzpuls) erkannt (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT	x	
diCount	DINT	Zählerwert Masterimpulse [Inkremente] (Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT	x	
diPmOffset	DINT	Druckmarkenoffset der aktuellen Masterposition (Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT	x	

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar, bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung Busfehler wenn 'boErr' = TRUE: EtherCAT oder CAN nicht in Operation Mode.		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boEnabAckPmInsetterCont	BOOL	Druckmarkenbaustein initialisiert und aktiviert		x
boParameterSetupDone	BOOL	Parametrierung beendet		x
boPmDetected	BOOL	Marke erkannt, Regelung aktiv		x
boPmLost	BOOL	Meldung "Marke verloren", wenn die an Parameter 'iNoPmLost' eingestellte Anzahl von aufeinanderfolgenden Marken nicht erkannt wurde.		x
boPmWnd	BOOL	Markenfenster aktiviert den Druckmarkensensor		x
diPmDeviation	DINT	Abweichung Soll- und Istposition der aktuellen Druckmarke		x
diOpOffsetOut	DINT	Ausgabewert beinhaltet den aktuell ausgegebenen Operatoroffset		x
iActPmLost	INT	Anzahl verlorener Druckmarken		x

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS_PRINTMARK_INSETTER_CONT	STANDARD_AXIS_PRINTMARK_INSETTER_ CONT.actSync

4.1.4.2.6 STANDARD_AXIS_PRINTMARK_INSETTER_INTERVAL (FB)

Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des zu bearbeitenden Bandes wird zu einem Masterformat geregelt (Insetting)

Siehe 'PRINT_MARK_INSETTER_INTERVAL (FB)' auf Seite 521.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

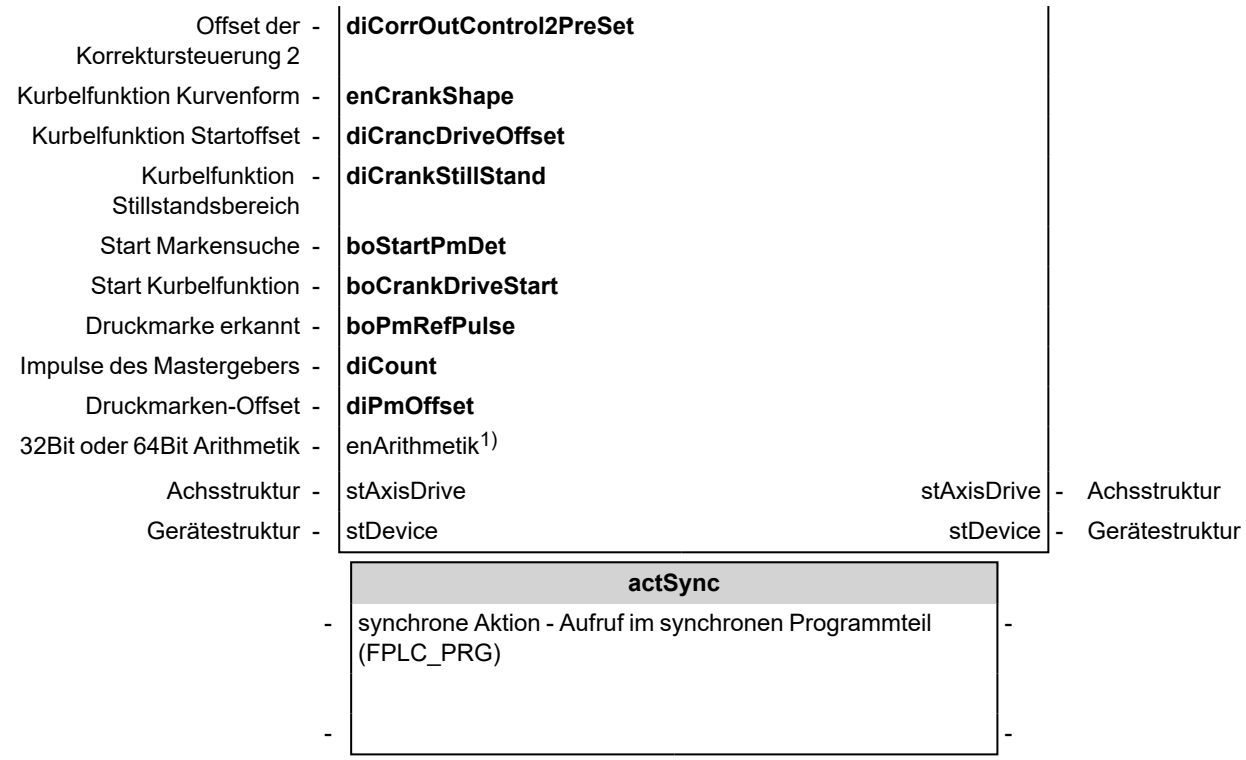
- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Anwender Interface

		STANDARD_AXIS_PRINTMARK_INSETTER_INTERVAL			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Positioniergeschwindigkeit	-	udPosVelocity	**boEnabAckPmInsetterInterval**	-	Druckmarkenbaustein initialisiert und aktiviert
Zielposition	-	diPosition	**boParameterSetupDone**	-	Parametrierung beendet
Beschleunigung	-	lreAccel	**boPmDetected**	-	Marke gefunden
Verzögerung	-	lreDecel	**boPmLost**	-	Marke verloren
Ruckbegrenzung beim Start	-	lreJerkStart	**boPmWnd**	-	Fenster ein
Ruckbegrenzung beim Stopp	-	lreJerkEnd	**diPmDeviation**	-	Abweichung Soll-Ist-Position Druckmarke
Tippgeschwindigkeit	-	udVelocityJog	**diOpOffsetOut**	-	Ausgegebener Operator-Offset
Drehzahl für Drehzahlr.	-	diSpeedVelocity	**iActPmLost**	-	Anzahl verlorener Druckmarken
Freigabesignal DM Regelung	-	**boEnablePmInsetterInterval**			
Rücksetzen Ausgang boPmLost	-	**boResetPmLost**			
Start Ausgabe Operator-Offset	-	**boSetupOffset**			
Start mit Druckmarken-Sollposition	-	**boSetupPmPos**			
Master zählt negativ	-	**boMasterNegDir**			
Motorrichtung negativ	-	**boMotorNegDir**			
Freigabe Korrekturausgabe	-	**boCorrOutControlOn**			
Operator Offset	-	**diOpOffset**			
Formatlänge	-	**diFormatLength**			
Fensterlänge	-	**diWindowLength**			
Anzahl Marken bis Meldung "Marke verloren"	-	**iNoPmLost**			
Korrekturbereich	-	**diCorrectionRange**			
Druckmarken-Sollposition	-	**diPmSetupPos**			
Maximaler Korrekturwert pro Format	-	**diCorrectionLimit**			
Start der Korrekturbereich 1	-	**diCorrRng1Start**			
Ende der Korrekturbereich 1	-	**diCorrRng1End**			
Offset der Korrektursteuerung 1	-	**diCorrOutControl1PreSet**			
Start der Korrekturbereich 2	-	**diCorrRng2Start**			
Ende der Korrekturbereich 2	-	**diCorrRng2End**			



1) Gilt nur für CODESYS V3

Eingangsvariablen

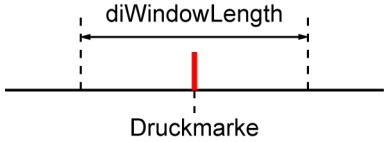
Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig		x
boFL	BOOL	Bit zum Fehlerlöschen		x

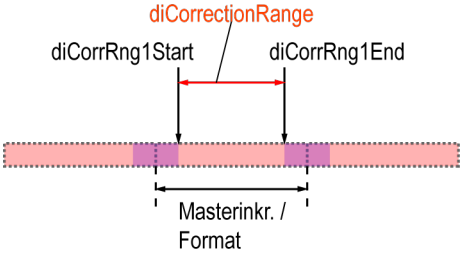
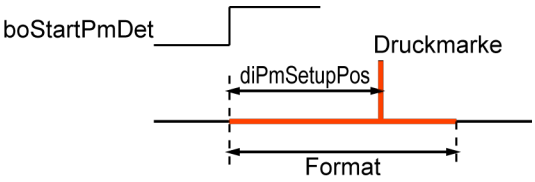
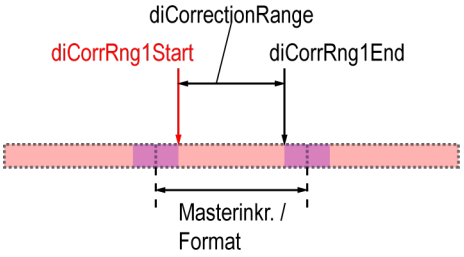
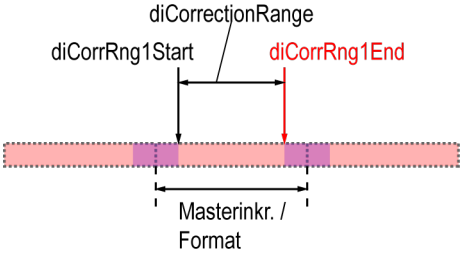
Name	Typ	Beschreibung	Sync.	Async.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_ PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x

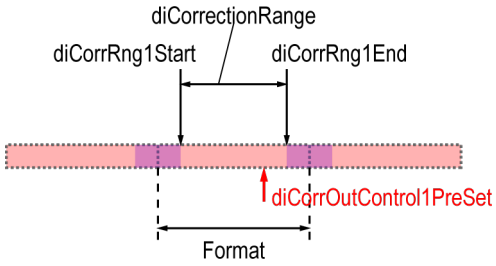
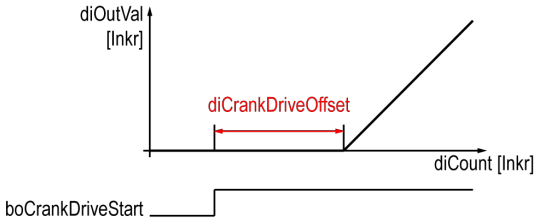
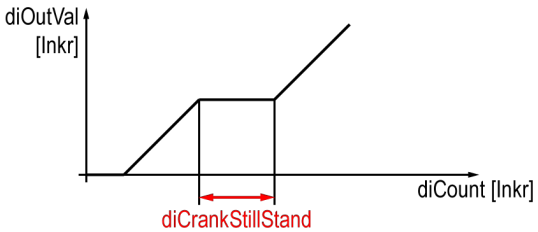
Name	Typ	Beschreibung	Sync.	Async.
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x

Name	Typ	Beschreibung	Sync.	Async.
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x

Name	Typ	Beschreibung	Sync.	Async.
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x
boEnablePmInsetterInterval	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' = TRUE ist, bleibt der Baustein freigegeben und wird von der PLC bearbeitet. Die Eingangsinkrementerfassung ist aktiviert. Inkrementänderungen am Eingang 'diInVal' führen zu einer Änderung des Masterbezugspunkts. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boResetPmLost	BOOL	Rücksetzen des Ausgangs 'boPmLost'		x
boSetupOffset	BOOL	Ausgabestart des angegebenen Werts von 'diOpOffset'. Dieser Wert dient als Offset zur Verschiebung der Druckmarken-Sollposition		x
boSetupPmPos	BOOL	Steuerung der Sollpositionsbestimmung der Druckmarke bei positiver Flanke von 'boStartPmDet' 'boPmSetupPos' = FALSE: Es wird die Position der nächsten Druckmarke als Soll-Position erfasst und alle nachfolgenden Marken auf diese Position geregelt. 'boPmSetupPos' = TRUE: Der Parameter 'diPmSetupPos' gibt die Soll-Position der Druckmarke relativ zur aktuellen Masterposition an. Die folgenden Druckmarken werden auf diese Position geregelt.		x

Name	Typ	Beschreibung	Sync.	Async.
boMasterNegDir	BOOL	Negation Master-Impuls-Richtung 'boMasterNegDir' = FALSE: keine Negation 'boMasterNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.		x
boMotorNegDir	BOOL	Negation Motor-Drehrichtung 'boMotorNegDir' = FALSE: keine Negation 'boMotorNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.		x
boCorrOutControlOn	BOOL	Korrekturausgabe-Steuerung aktiv. 'boCorrOutControlOn' = FALSE: Korrekturausgabe immer aktiv 'boCorrOutControlOn' = TRUE: Korrekturausgabe erfolgt nur in dem durch die Variablen 'diCorrRng1Start' und 'diCorrRng1End' bzw. 'diCorrRng2Start' und 'diCorrRng2End' festgelegten Bereich. Beide Bereiche sind mit ODER-Funktion verknüpft. Die Bereichszählung (Modulozählung) wird mit der 0→1 Flanke an 'boCorrOutControlOn' gestartet und lässt sich durch einen Wert an 'diCorrOutControl1PreSet' bzw. 'diCorrOutControl2PreSet' entsprechend einstellen.		x
diOpOffset	DINT	Operator-Offset. Verschiebt die Sollposition der Druckmarke. Die Verschiebung erfolgt im Korrekturbereich und wird mit einer 0→1 Flanke an 'boSetupOffset' gestartet. Sie ist unabhängig von Wertänderungen an 'diCount'. Die Verschiebegeschwindigkeit berechnet sich aus		x
diFormatLength	DINT	Formatlänge Soll-Abstand zweier Druckmarken		x
diWindowLength	DINT	Fensterlänge Länge vor und nach der Soll-Druckmarkenposition. In diesem Bereich ist der Druckmarkensensor aktiviert. Eine Druckmarke wird akzeptiert. 		x
iNoPmLost	INT	Anzahl Druckmarken, die nacheinander fehlen dürfen, bevor der Ausgang 'boPMLost' (Marke verloren) gesetzt wird.		x

Name	Typ	Beschreibung	Sync.	Async.
diCorrectionRange	DINT	Korrekturbereich Die Druckmarkenregelung versucht die Korrektur im angegebenen Bereich auszugeben. [% diFormatLength] 		x
diPmSetupPos	DINT	Der Parameter gibt die Sollposition der Druckmarke zur aktuellen Masterposition bei positiver Flanke von 'boStartPmDet' an. Die erkannten Druckmarken werden auf diese Position geregelt. [Inkremente] 		x
diCorrectionLimit	DINT	Maximale Korrekturausgabe pro Format [Inkremente] 'diCorrectionLimit' = 0: keine Begrenzung 'diCorrectionLimit' > 0: maximaler Wert		x
diCorrRng1Start **diCorrRng2Start**	DINT	Korrekturbereich 1 / 2 Startwert bezogen auf Masterinkremente / Format [Inkremente] 		x
diCorrRng1End **diCorrRng2End**	DINT	Korrekturbereich 1 / 2 Endwert bezogen auf Masterinkremente / Format [Inkremente] 		x

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>diCorrOutControl1PreSet
diCorrOutControl2PreSet</td><td>DINT</td><td>Positionswert (Nullverschiebung) bei Start der Korrektursteuerung bezogen auf Masterinkremente / Format [Inkremente]
Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn' >Format/2< einzustellen.
 
</td><td></td><td>x</td></tr>
<tr><td>enCrankShape</td><td>ENUM</td><td>EN_CRANK_DRIVE_SHAPE
Kurvenform der Kurbelfunktion
<table><tr><td>enCrankShapeLinear</td><td>linearer Verlauf 45°</td></tr><tr><td>enCrankShapeSine</td><td>SINUS-Funktion</td></tr><tr><td>enCrankShapeSquareSine</td><td>SINUS^2-Funktion</td></tr></table></td><td></td><td>x</td></tr>
<tr><td>diCrancDriveOffset</td><td>DINT</td><td>Kurbelfunktion Start-Offset.
Nach Start der Kurbelfunktion wartet die Funktion, bis der Master den Offset zurückgelegt hat und beginnt dann
 
</td><td></td><td>x</td></tr>
<tr><td>diCrankStillStand</td><td>DINT</td><td>Kurbelfunktion Stillstandsbereich
 
</td><td></td><td>x</td></tr>
<tr><td>boStartPmDet</td><td>BOOL</td><td>Start der Druckmarkenregelung und Bestimmung der Druckmarken-Sollposition durch Parameter 'diPmSetupPos' wenn 'boSetupPmPos' = TRUE</td><td>x</td><td></td></tr>
<tr><td>boCrankDriveStart</td><td>BOOL</td><td>Start der Kurbelfunktion
Wertausgabe am Ausgang 'diOutVal' entsprechend der am Eingang 'enCrankShape' gewählten Kurvenform. Die Wertausgabe erfolgt bei Wertänderung an 'diCount'</td><td>x</td><td></td></tr>
<tr><td>boPmRefPulse</td><td>BOOL</td><td>Markenimpuls (Referenzpuls) erkannt
(Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT</td><td>x</td><td></td></tr>
<tr><td>diCount</td><td>DINT</td><td>Zählerwert
Masterimpulse [Inkremente]
(Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT</td><td>x</td><td></td></tr>
</table>

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>diPmOffset</td><td>DINT</td><td>Druckmarkenoffset der aktuellen Masterposition
(Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT</td><td>x</td><td></td></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar, bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung Busfehler wenn 'boErr' = TRUE: EtherCAT oder CAN nicht in Operation Mode.		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boEnabAckPmInsetterInterval	BOOL	Druckmarkenbaustein initialisiert und aktiviert		x
boParameterSetupDone	BOOL	Parametrierung beendet		x
boPmDetected	BOOL	Marke erkannt, Regelung aktiv		x
boPmLost	BOOL	Meldung "Marke verloren", wenn die an Parameter 'iNoPmLost' eingestellte Anzahl von aufeinanderfolgenden Marken nicht erkannt wurde.		x
boPmWnd	BOOL	Markenfenster aktiviert den Druckmarkensensor		x
diPmDeviation	DINT	Abweichung Soll- und Istposition der aktuellen Druckmarke		x
diOpOffsetOut	DINT	Ausgabewert beinhaltet den aktuell ausgegebenen Operatoroffset		x
iActPmLost	INT	Anzahl verlorener Druckmarken		x

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS_PRINTMARK_INSETTER_INTERVAL	STANDARD_AXIS_PRINTMARK_INSETTER_ INTERVAL.actSync

4.1.4.2.7 STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_CONT (FB)

Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des bearbeitenden Aggregats (z.B. Messer, Magnetstanze) wird zu einem Masterformat geregelt (Registerregler)

Siehe 'PRINT_MARK_REGISTER_CONTROL_CONTINUOUS (FB)' auf Seite 533.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

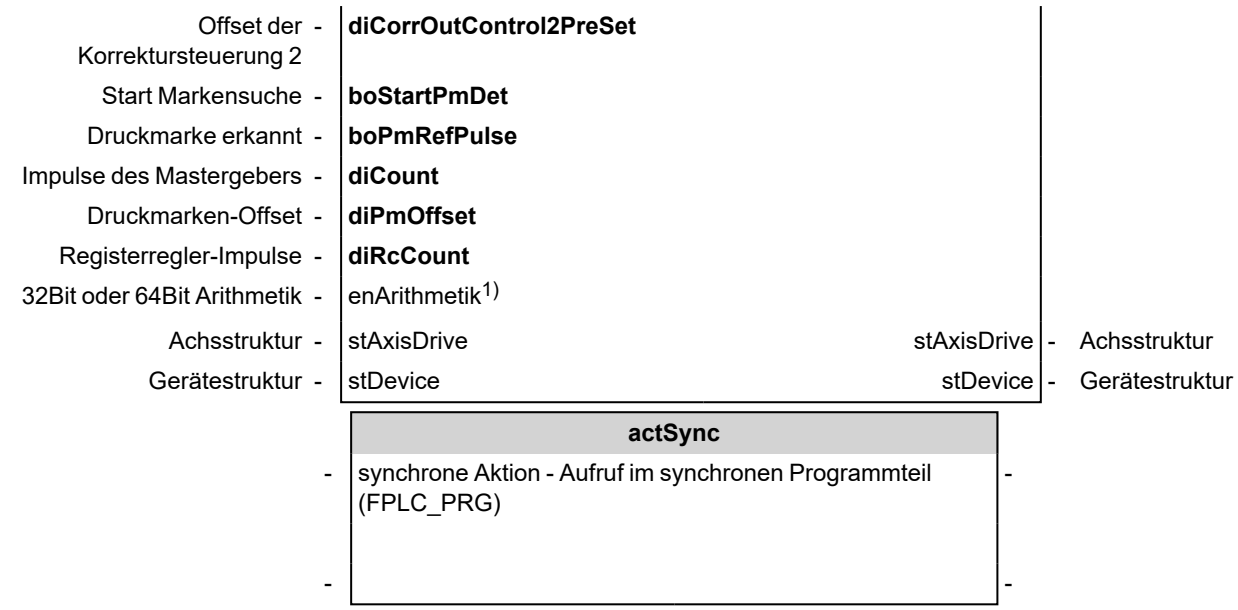
- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Anwender Interface

		STD_AXIS_PRINTMARK_REGISTER_CONTROL_CONT			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Positioniergeschwindigkeit	-	udPosVelocity	**boEnabAckPmRegCtrlCont**	-	Druckmarkenbaustein initialisiert und aktiviert
Zielposition	-	diPosition	**boParameterSetupDone**	-	Parametrierung beendet
Beschleunigung	-	lreAccel	**boPmDetected**	-	Marke gefunden
Verzögerung	-	lreDecel	**boPmLost**	-	Marke verloren
Ruckbegrenzung beim Start	-	lreJerkStart	**boPmWnd**	-	Fenster ein
Ruckbegrenzung beim Stopp	-	lreJerkEnd	**diPmDeviation**	-	Abweichung Soll-Ist-Position Druckmarke
Tippgeschwindigkeit	-	udVelocityJog	**diOpOffsetOut**	-	Ausgegebener Operator-Offset
Drehzahl für Drehzahlr.	-	diSpeedVelocity	**iActPmLost**	-	Anzahl verlorener Druckmarken
Freigabesignal DM Regelung	-	**boEnablePmRegisterCtrlCont**			
Rücksetzen Ausgang boPmLost	-	**boResetPmLost**			
Start Ausgabe Operator-Offset	-	**boSetupOffset**			
Start mit Druckmarken-Sollposition	-	**boSetupPmPos**			
Master zählt negativ	-	**boMasterNegDir**			
Motorrichtung negativ	-	**boMotorNegDir**			
Freigabe Korrekturausgabe	-	**boCorrOutControlOn**			
Operator Offset	-	**diOpOffset**			
Formatlänge	-	**diFormatLength**			
Fensterlänge	-	**diWindowLength**			
Anzahl Marken bis Meldung "Marke verloren"	-	**iNoPmLost**			
Korrekturbereich	-	**diCorrectionRange**			
Druckmarken-Sollposition	-	**diPmSetupPos**			
Maximaler Korrekturwert pro Format	-	**diCorrectionLimit**			
Start der Korrekturbereich 1	-	**diCorrRng1Start**			
Ende der Korrekturbereich 1	-	**diCorrRng1End**			
Offset der Korrektursteuerung 1	-	**diCorrOutControl1PreSet**			
Start der Korrekturbereich 2	-	**diCorrRng2Start**			
Ende der Korrekturbereich 2	-	**diCorrRng2End**			



1) Gilt nur für CODESYS V3

Eingangsvariablen

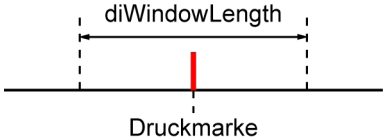
Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_ PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

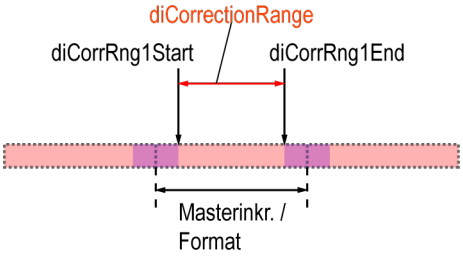
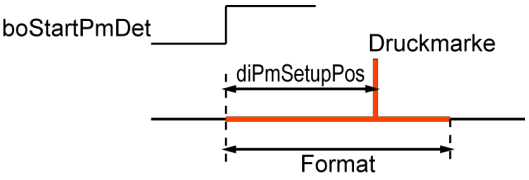
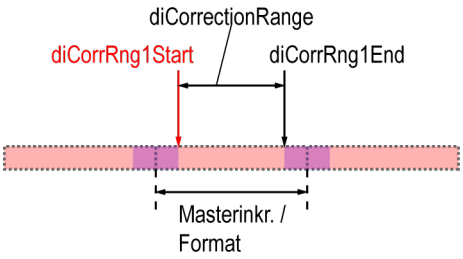
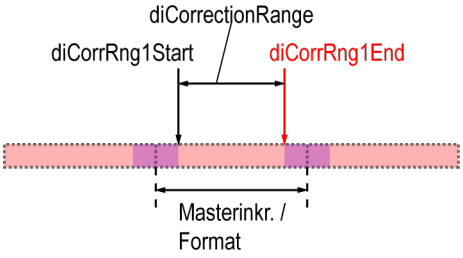
Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x

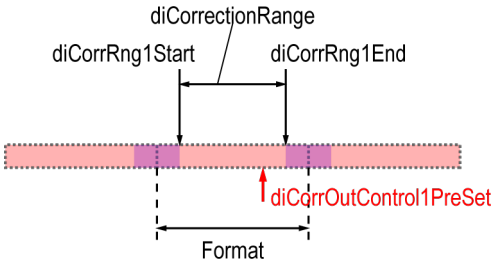
Name	Typ	Beschreibung	Sync.	Async.
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x

Name	Typ	Beschreibung	Sync.	Async.
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x

Name	Typ	Beschreibung	Sync.	Async.
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x
boEnablePmRegisterCtrlCont	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' = TRUE ist, bleibt der Baustein freigegeben und wird von der PLC bearbeitet. Die Eingangsinkrementerfassung ist aktiviert. Inkrementänderungen am Eingang 'diInVal' führen zu einer Änderung des Masterbezugspunkts. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boResetPmLost	BOOL	Rücksetzen des Ausgangs 'boPmLost'		x
boSetupOffset	BOOL	Ausgabestart des angegebenen Werts von 'diOpOffset'. Dieser Wert dient als Offset zur Verschiebung der Druckmarken-Sollposition		x

Name	Typ	Beschreibung	Sync.	Async.
boSetupPmPos	BOOL	Steuerung der Sollpositionsbestimmung der Druckmarke bei positiver Flanke von 'boStartPmDet' 'boPmSetupPos' = FALSE: Es wird die Position der nächsten Druckmarke als Soll-Position erfasst und alle nachfolgenden Marken auf diese Position geregelt. 'boPmSetupPos' = TRUE: Der Parameter 'diPmSetupPos' gibt die Soll-Position der Druckmarke relativ zur aktuellen Masterposition an. Die folgenden Druckmarken werden auf diese Position geregelt.		x
boMasterNegDir	BOOL	Negation Master-Impuls-Richtung 'boMasterNegDir' = FALSE: keine Negation 'boMasterNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.		x
boMotorNegDir	BOOL	Negation Motor-Drehrichtung 'boMotorNegDir' = FALSE: keine Negation 'boMotorNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.		x
boCorrOutControlOn	BOOL	Korrekturausgabe-Steuerung aktiv. 'boCorrOutControlOn' = FALSE: Korrekturausgabe immer aktiv 'boCorrOutControlOn' = TRUE: Korrekturausgabe erfolgt nur in dem durch die Variablen 'diCorrRng1Start' und 'diCorrRng1End' bzw. 'diCorrRng2Start' und 'diCorrRng2End' festgelegten Bereich. Beide Bereiche sind mit ODER-Funktion verknüpft. Die Bereichszählung (Modulozählung) wird mit der 0→1 Flanke an 'boCorrOutControlOn' gestartet und lässt sich durch einen Wert an 'diCorrOutControl1PreSet' bzw. 'diCorrOutControl2PreSet' entsprechend einstellen.		x
diOpOffset	DINT	Operator-Offset. Verschiebt die Sollposition der Druckmarke. Die Verschiebung erfolgt im Korrekturbereich und wird mit einer 0→1 Flanke an 'boSetupOffset' gestartet. Sie ist unabhängig von Wertänderungen an 'diCount'. Die Verschiebegeschwindigkeit berechnet sich aus		x
diFormatLength	DINT	Formatlänge Soll-Abstand zweier Druckmarken		x
diWindowLength	DINT	Fensterlänge Länge vor und nach der Soll-Druckmarkenposition. In diesem Bereich ist der Druckmarkensensor aktiviert. Eine Druckmarke wird akzeptiert. 		x

Name	Typ	Beschreibung	Sync.	Async.
iNoPmLost	INT	Anzahl Druckmarken, die nacheinander fehlen dürfen, bevor der Ausgang 'boPMLost' (Marke verloren) gesetzt wird.		x
diCorrectionRange	DINT	Korrekturbereich Die Druckmarkenregelung versucht die Korrektur im angegebenen Bereich auszugeben. [% diFormatLength] 		x
diPmSetupPos	DINT	Der Parameter gibt die Sollposition der Druckmarke zur aktuellen Masterposition bei positiver Flanke von 'boStartPmDet' an. Die erkannten Druckmarken werden auf diese Position geregelt. [Inkremente] 		x
diCorrectionLimit	DINT	Maximale Korrekturausgabe pro Format [Inkremente] 'diCorrectionLimit' = 0: keine Begrenzung 'diCorrectionLimit' > 0: maximaler Wert		x
diCorrRng1Start **diCorrRng2Start**	DINT	Korrekturbereich 1 / 2 Startwert bezogen auf Masterinkremente / Format [Inkremente] 		x
diCorrRng1End **diCorrRng2End**	DINT	Korrekturbereich 1 / 2 Endwert bezogen auf Masterinkremente / Format [Inkremente] 		x

Name	Typ	Beschreibung	Sync.	Async.
diCorrOutControl1PreSet **diCorrOutControl2PreSet**	DINT	Positionswert (Nullverschiebung) bei Start der Korrektursteuerung bezogen auf Masterinkremente / Format [Inkremente] Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn' >Format/2< einzustellen. 		x
boStartPmDet	BOOL	Start der Druckmarkenregelung und Bestimmung der Druckmarken-Sollposition durch Parameter 'diPmSetupPos' wenn 'boSetupPmPos' = TRUE	x	
boPmRefPulse	BOOL	Markenimpuls (Referenzpuls) erkannt (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT	x	
diCount	DINT	Zählerwert Masterimpulse [Inkremente] (Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT	x	
diPmOffset	DINT	Druckmarkenoffset der aktuellen Masterposition (Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT	x	
diRcCount	DINT	Registerregler-Impulse Impulse des Werkzeugs [Inkr]	x	

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar, bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:
<table>
<tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr>
<tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr>
<tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr>
</table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.
<table>
<tr><td>iErrID</td><td>5</td></tr>
</table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung Busfehler wenn 'boErr' = TRUE: EtherCAT oder CAN nicht in Operation Mode.		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boEnabAckPmRegCtrlCont	BOOL	Druckmarkenbaustein initialisiert und aktiviert		x
boParameterSetupDone	BOOL	Parametrierung beendet		x
boPmDetected	BOOL	Marke erkannt, Regelung aktiv		x
boPmLost	BOOL	Meldung "Marke verloren", wenn die an Parameter 'iNoPmLost' eingestellte Anzahl von aufeinanderfolgenden Marken nicht erkannt wurde.		x
boPmWnd	BOOL	Markenfenster aktiviert den Druckmarkensensor		x
diPmDeviation	DINT	Abweichung Soll- und Istposition der aktuellen Druckmarke		x
diOpOffsetOut	DINT	Ausgabewert beinhaltet den aktuell ausgegebenen Operatoroffset		x
iActPmLost	INT	Anzahl verlorener Druckmarken		x

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_CONT	STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_CONT.actSync

4.1.4.2.8 STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_DISCONT (FB)

Erweiterte Funktionalität:

- Druckmarkenregelung: Die Position des bearbeitenden Aggregats (z.B. Messer, Magnetstanze) wird zu einem Masterformat geregelt (Registerregler)

Siehe 'PRINT_MARK_REGISTER_CONTROL_DISCONT (FB)' auf Seite 539.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

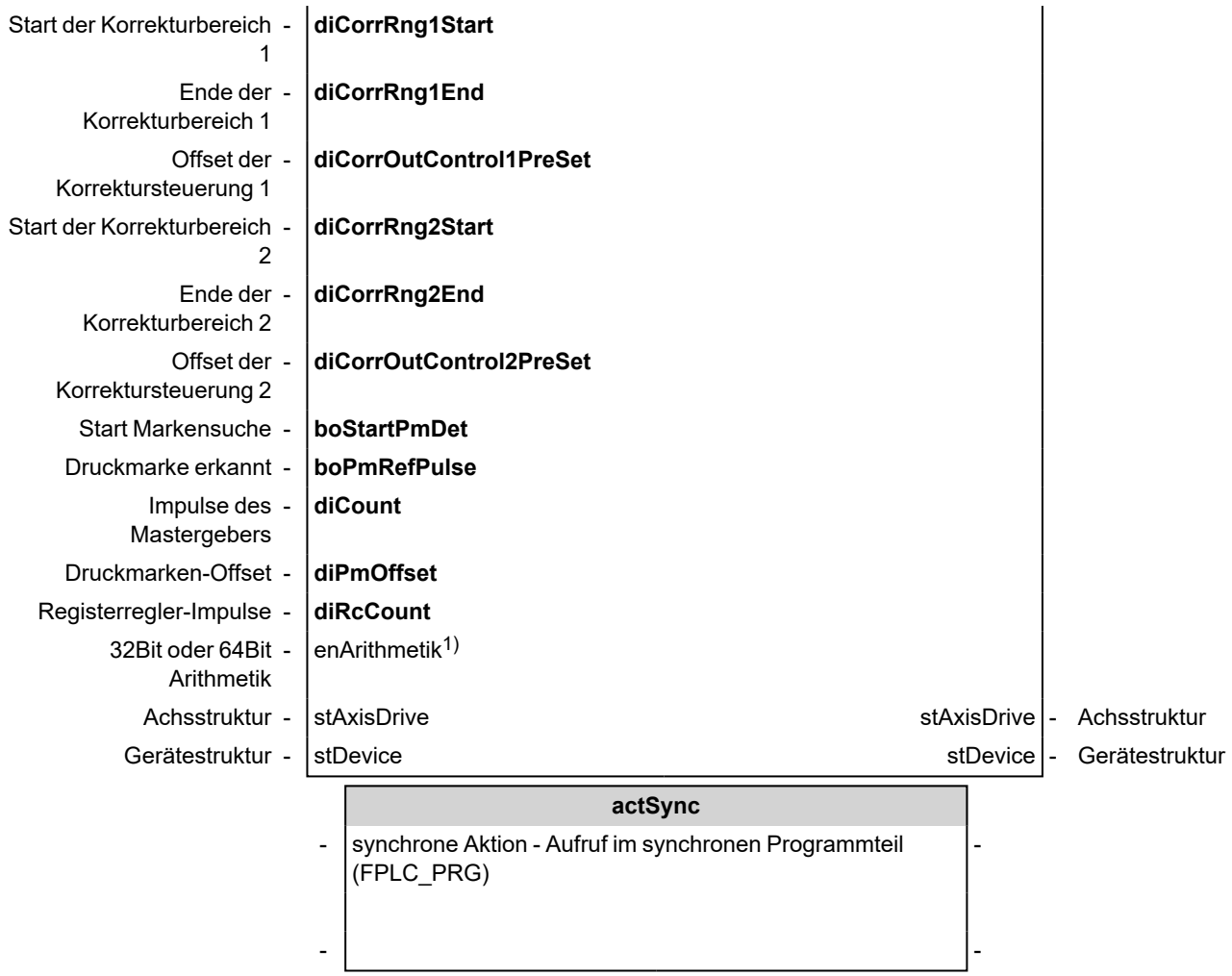
- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Anwender Interface

		STD_AXIS_PRINTMARK_REGISTER_CONTROL_DISCONT			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Positioniergeschwindigkeit	-	udPosVelocity	**boEnabAckPmRegisterCtrlDiscont**	-	Druckmarkenbaustein initialisiert und aktiviert
Zielposition	-	diPosition	**boParameterSetupDone**	-	Parametrierung beendet
Beschleunigung	-	lreAccel	**boPmDetected**	-	Marke gefunden
Verzögerung	-	lreDecel	**boPmLost**	-	Marke verloren
Ruckbegrenzung beim Start	-	lreJerkStart	**boPmWnd**	-	Fenster ein
Ruckbegrenzung beim Stopp	-	lreJerkEnd	**diPmDeviation**	-	Abweichung Soll-Ist-Position Druckmarke
Tippgeschwindigkeit	-	udVelocityJog	**diOpOffsetOut**	-	Ausgegebener Operator-Offset
Drehzahl für Drehzahlr.	-	diSpeedVelocity	**iActPmLost**	-	Anzahl verlorener Druckmarken
Freigabesignal DM Regelung	-	**boEnablePmRegisterCtrlDiscont**			
Rücksetzen Ausgang boPmLost	-	**boResetPmLost**			
Start Ausgabe Operator-Offset	-	**boSetupOffset**			
Start mit Druckmarken-Sollposition	-	**boSetupPmPos**			
Master zählt negativ	-	**boMasterNegDir**			
Motorrichtung negativ	-	**boMotorNegDir**			
Freigabe Korrekturausgabe	-	**boCorrOutControlOn**			
Operator Offset	-	**diOpOffset**			
Formatlänge	-	**diFormatLength**			
Fensterlänge	-	**diWindowLength**			
Anzahl Marken bis Meldung "Marke verloren"	-	**iNoPmLost**			
Korrekturbereich	-	**diCorrectionRange**			
Druckmarken-Sollposition	-	**diPmSetupPos**			
Maximaler Korrekturwert pro Format	-	**diCorrectionLimit**			



1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig		x
boFL	BOOL	Bit zum Fehlerlöschen		x

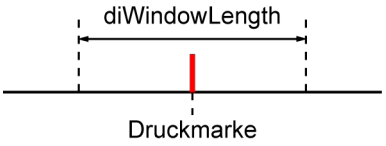
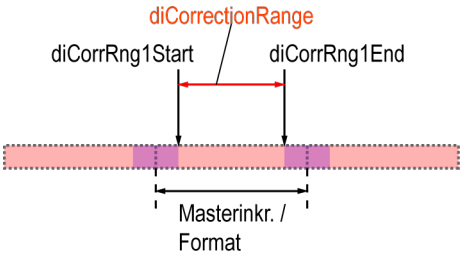
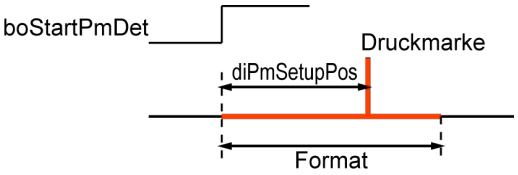
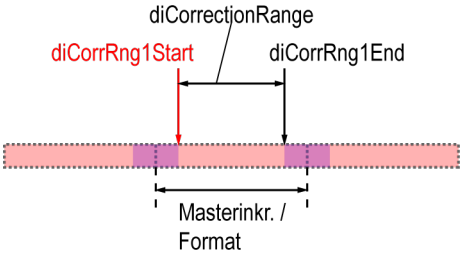
Name	Typ	Beschreibung	Sync.	Async.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_ PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x

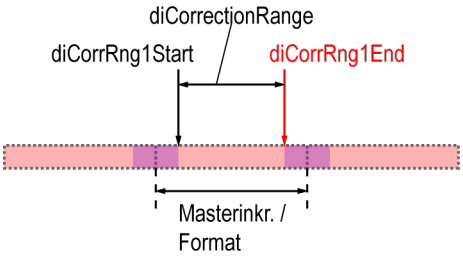
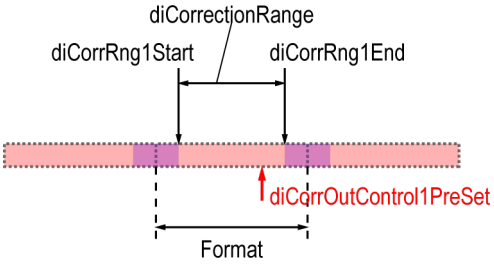
Name	Typ	Beschreibung	Sync.	Async.
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x

Name	Typ	Beschreibung	Sync.	Async.
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x

Name	Typ	Beschreibung	Sync.	Async.
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x
boEnablePmRegisterCtrlDiscont	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' = TRUE ist, bleibt der Baustein freigegeben und wird von der PLC bearbeitet. Die Eingangsinkrementerfassung ist aktiviert. Inkrementänderungen am Eingang 'diInVal' führen zu einer Änderung des Masterbezugspunkts. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boResetPmLost	BOOL	Rücksetzen des Ausgangs 'boPmLost'		x
boSetupOffset	BOOL	Ausgabestart des angegebenen Werts von 'diOpOffset'. Dieser Wert dient als Offset zur Verschiebung der Druckmarken-Sollposition		x

Name	Typ	Beschreibung	Sync.	Async.
boSetupPmPos	BOOL	Steuerung der Sollpositionsbestimmung der Druckmarke bei positiver Flanke von 'boStartPmDet' 'boPmSetupPos' = FALSE: Es wird die Position der nächsten Druckmarke als Soll-Position erfasst und alle nachfolgenden Marken auf diese Position geregelt. 'boPmSetupPos' = TRUE: Der Parameter 'diPmSetupPos' gibt die Soll-Position der Druckmarke relativ zur aktuellen Masterposition an. Die folgenden Druckmarken werden auf diese Position geregelt.		x
boMasterNegDir	BOOL	Negation Master-Impuls-Richtung 'boMasterNegDir' = FALSE: keine Negation 'boMasterNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.		x
boMotorNegDir	BOOL	Negation Motor-Drehrichtung 'boMotorNegDir' = FALSE: keine Negation 'boMotorNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.		x
boCorrOutControlOn	BOOL	Korrekturausgabe-Steuerung aktiv. 'boCorrOutControlOn' = FALSE: Korrekturausgabe immer aktiv 'boCorrOutControlOn' = TRUE: Korrekturausgabe erfolgt nur in dem durch die Variablen 'diCorrRng1Start' und 'diCorrRng1End' bzw. 'diCorrRng2Start' und 'diCorrRng2End' festgelegten Bereich. Beide Bereiche sind mit ODER-Funktion verknüpft. Die Bereichszählung (Modulozählung) wird mit der 0→1 Flanke an 'boCorrOutControlOn' gestartet und lässt sich durch einen Wert an 'diCorrOutControl1PreSet' bzw. 'diCorrOutControl2PreSet' entsprechend einstellen.		x
diOpOffset	DINT	Operator-Offset. Verschiebt die Sollposition der Druckmarke. Die Verschiebung erfolgt im Korrekturbereich und wird mit einer 0→1 Flanke an 'boSetupOffset' gestartet. Sie ist unabhängig von Wertänderungen an 'diCount'. Die Verschiebegeschwindigkeit berechnet sich aus		x
diFormatLength	DINT	Formatlänge Soll-Abstand zweier Druckmarken		x

Name	Typ	Beschreibung	Sync.	Async.
diWindowLength	DINT	Fensterlänge Länge vor und nach der Soll-Druckmarkenposition. In diesem Bereich ist der Druckmarkensensor aktiviert. Eine Druckmarke wird akzeptiert. 		x
iNoPmLost	INT	Anzahl Druckmarken, die nacheinander fehlen dürfen, bevor der Ausgang 'boPMLost' (Marke verloren) gesetzt wird.		x
diCorrectionRange	DINT	Korrekturbereich Die Druckmarkenregelung versucht die Korrektur im angegebenen Bereich auszugeben. [% diFormatLength] 		x
diPmSetupPos	DINT	Der Parameter gibt die Sollposition der Druckmarke zur aktuellen Masterposition bei positiver Flanke von 'boStartPmDet' an. Die erkannten Druckmarken werden auf diese Position geregelt. [Inkremente] 		x
diCorrectionLimit	DINT	Maximale Korrekturausgabe pro Format [Inkremente] 'diCorrectionLimit' = 0: keine Begrenzung 'diCorrectionLimit' > 0: maximaler Wert		x
diCorrRng1Start **diCorrRng2Start**	DINT	Korrekturbereich 1 / 2 Startwert bezogen auf Masterinkremente / Format [Inkremente] 		x

Name	Typ	Beschreibung	Sync.	Async.
diCorrRng1End **diCorrRng2End**	DINT	Korrekturbereich 1 / 2 Endwert bezogen auf Masterinkremente / Format [Inkremente] 		x
diCorrOutControl1PreSet **diCorrOutControl2PreSet**	DINT	Positionswert (Nullverschiebung) bei Start der Korrektursteuerung bezogen auf Masterinkremente / Format [Inkremente] Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn' >Format/2< einzustellen. 		x
boStartPmDet	BOOL	Start der Druckmarkenregelung und Bestimmung der Druckmarken-Sollposition durch Parameter 'diPmSetupPos' wenn 'boSetupPmPos' = TRUE	x	
boPmRefPulse	BOOL	Markenimpuls (Referenzpuls) erkannt (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT	x	
diCount	DINT	Zählerwert Masterimpulse [Inkremente] (Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT	x	
diPmOffset	DINT	Druckmarkenoffset der aktuellen Masterposition (Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT	x	
diRcCount	DINT	Registerregler-Impulse Impulse des Werkzeugs [Inkr]	x	

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar, bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung Busfehler wenn 'boErr' = TRUE: EtherCAT oder CAN nicht in Operation Mode.		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boEnabAckPmRegisterCtrlDiscont	BOOL	Druckmarkenbaustein initialisiert und aktiviert		x
boParameterSetupDone	BOOL	Parametrierung beendet		x
boPmDetected	BOOL	Marke erkannt, Regelung aktiv		x
boPmLost	BOOL	Meldung "Marke verloren", wenn die an Parameter 'iNoPmLost' eingestellte Anzahl von aufeinanderfolgenden Marken nicht erkannt wurde.		x
boPmWnd	BOOL	Markenfenster aktiviert den Druckmarkensensor		x
diPmDeviation	DINT	Abweichung Soll- und Istposition der aktuellen Druckmarke		x
diOpOffsetOut	DINT	Ausgabewert beinhaltet den aktuell ausgegebenen Operatoroffset		x
iActPmLost	INT	Anzahl verlorener Druckmarken		x

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_DISCONT	STANDARD_AXIS_PRINTMARK_REGISTER_CONTROL_DISCONT.actSync

4.1.4.2.9 STANDARD_FOLLOW_AXIS_GEAR (FB)

Erweiterte Funktionalität:

- Die Slave Achse folgt kontinuierlich einem Master.
- Ein Getriebeverhältnis kann vorgegeben werden.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

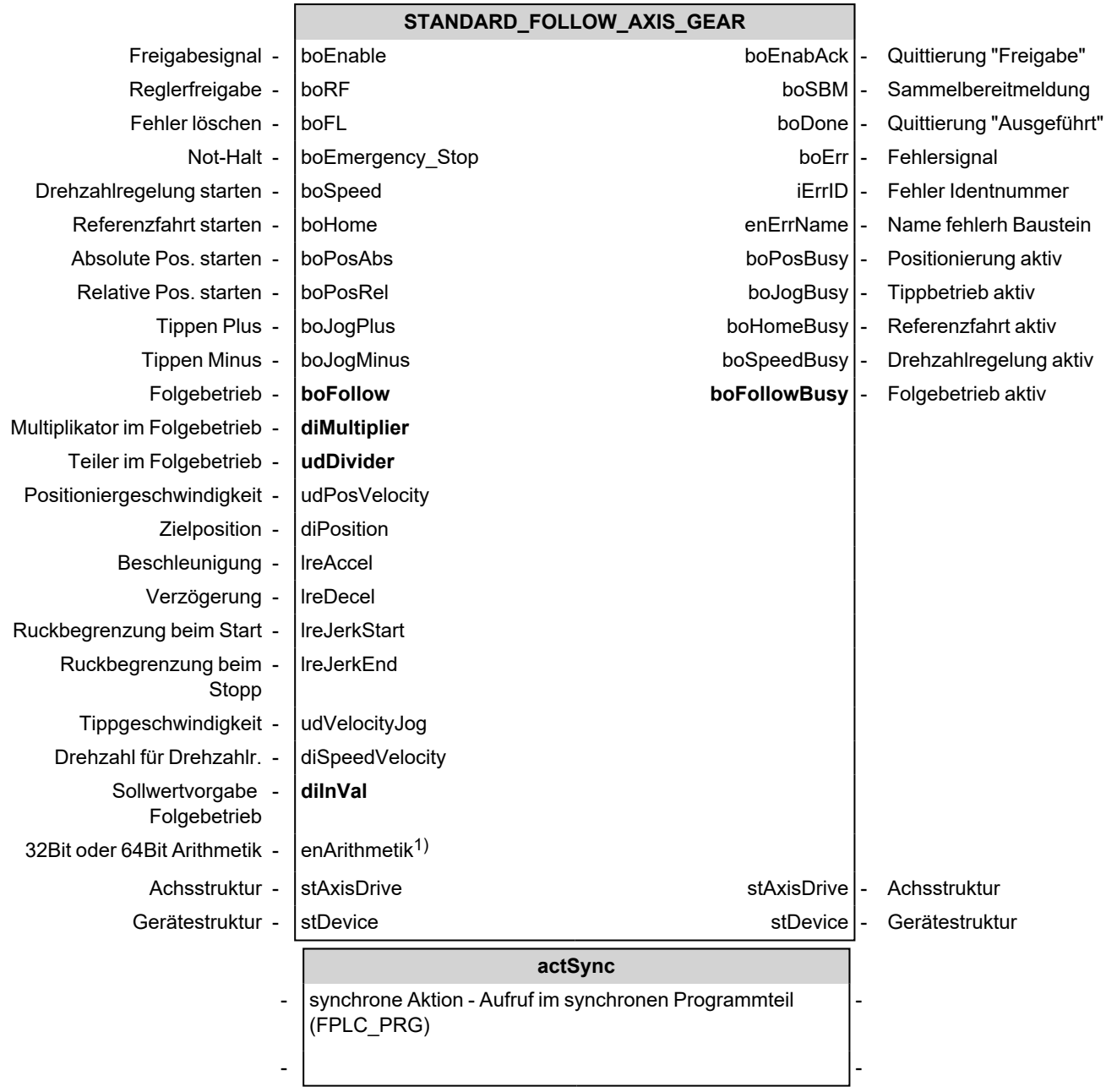
Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Anwender Interface



1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x

Name	Typ	Beschreibung	Sync.	Async.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig Verhalten bei RF Entzug in der Betriebsart Lageregelung mit Folgebetrieb (boFollow = TRUE) und aktiver SAK (ID32801 Bit 9 = TRUE) In der Betriebsart Lageregelung mit aktiver SAK (Schleppabstandskompensation) kann die Funktion Tieflauframpe nach RF Entzug (siehe ID32782 'Tieflaufzeit RF inaktiv') nicht genutzt werden. Bei Entzug der Reglerfreigabe (boRF = FALSE) wird die Lagesollwertausgabe sofort gestoppt und dadurch die Geschwindigkeit des Motors maximal verzögert. Abhilfe für Applikationen mit Tieflauframpe: Generieren Sie mit der PLC die Lagesollwerte unter Berücksichtigung der Tieflauframpe. Entziehen Sie erst bei Stillstand die Reglerfreigabe.		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_ Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
boFollow	BOOL	Folgebetrieb Freigabesignal: Mit einer positiven Flanke startet die Funktion Folgebetrieb. Solange 'boFollow' = TRUE ist, wird der Folgebetrieb ausgeführt. Mit einer negativen Flanke 'boFollow' = FALSE wird der laufende Folgebetrieb abgebrochen. Die Achse folgt den Eingangsinkrementen über das eingestellte Getriebeverhältnis. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' 0x00430404 Relevante Variablen: Sollwertvorgabe: diInVal Multiplikator: diMultiplier Teiler: udDivider Folgebetrieb aktiv: boFollowBusy		x
diMultiplier	DINT	Über diMultiplier' und 'udDivider' kann jedes beliebige Verhältnis vorgegeben werden. Sollwert Folgebetrieb = diInVal * (diMultiplier / udDivider)	x	
udDivider	UDINT		x	
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>lreDecel</td><td>LREAL</td><td>Verzögerung für Positionierung und Tippbetrieb<table><tr><td>Bereich</td><td>1,43 10^{-13} < |'lreDecel'| < 1,43 10^{+16}</td></tr><tr><td>Einheit</td><td>Inkr/s^2</td></tr><tr><td>Standardwert</td><td>1000000</td></tr></table></td><td></td><td>x</td></tr>
<tr><td>lreJerkStart</td><td>LREAL</td><td>Ruckbegrenzung (Start) für Positionierung und Tippbetrieb<table><tr><td>Einheit</td><td>Inkr/s^3</td></tr><tr><td>Standardwert</td><td>10000000</td></tr></table></td><td></td><td>x</td></tr>
<tr><td>lreJerkEnd</td><td>LREAL</td><td>Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb<table><tr><td>Einheit</td><td>Inkr/s^3</td></tr><tr><td>Standardwert</td><td>10000000</td></tr></table></td><td></td><td>x</td></tr>
<tr><td>udVelocityJog</td><td>UDINT</td><td>Tippgeschwindigkeit
Einheit: Inkr/s
Standardwert: 34133</td><td></td><td>x</td></tr>
<tr><td>diSpeedVelocity</td><td>DINT</td><td>Drehzahlsollwert für Drehzahlregelung
Einheit: U/min
Standardwert: 100</td><td></td><td>x</td></tr>
<tr><td>diInVal</td><td>DINT</td><td>Sollwertvorgabe Folgebetrieb
(z. B. durch Masterachse, FB VGEN)
Einheit: Inkremente</td><td>x</td><td></td></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boFollowBusy	BOOL	Folgebetrieb aktiv		x

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS	STANDARD_AXIS.actSync

4.1.4.2.10 STANDARD_FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2 (FB)

Erweiterte Funktionalität:

- Die Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle.
- Am Tabellenende kann eine Tabellenumschaltung durchgeführt werden.

Siehe 'FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2 (FB)' auf Seite 410.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Anwender Interface

		STD_FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Aktivierung der Tabelle 2	-	**boFollowEnable**	**boFollowEnableAck**	-	Folgebetrieb Freigabe
Folgebetrieb	-	**boEnableOpTab2**	**boTab1Busy**	-	Tabelle 1 aktiv
Positioniergeschwindigkeit	-	udPosVelocity	**boTab2Busy**	-	Tabelle 2 aktiv
Zielposition	-	diPosition			
Beschleunigung	-	lreAccel			
Verzögerung	-	lreDecel			
Ruckbegrenzung beim Start	-	lreJerkStart			
Ruckbegrenzung beim Stopp	-	lreJerkEnd			
Tippgeschwindigkeit	-	udVelocityJog			
Drehzahl für Drehzahlr.	-	diSpeedVelocity			
Sollwertvorgabe Folgebetrieb	-	**diInVal**			
Pointeradresse auf Tabelle 1	-	**pstOpTab1**			
Pointeradresse auf Tabelle 2	-	**pstOpTab2**			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Achsstruktur	-	stAxisDrive	stAxisDrive	-	Achsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

	actSync	
-	synchrone Aktion - Aufruf im synchronen Programmteil (FPLC_PRG)	-
-		-

1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x

Name	Typ	Beschreibung	Sync.	Async.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig Verhalten bei RF Entzug in der Betriebsart Lageregelung mit Folgebetrieb (boFollow = TRUE) und aktiver SAK (ID32801 Bit 9 = TRUE) In der Betriebsart Lageregelung mit aktiver SAK (Schleppabstandskompensation) kann die Funktion Tieflauframpe nach RF Entzug (siehe ID32782 'Tieflaufzeit RF inaktiv') nicht genutzt werden. Bei Entzug der Reglerfreigabe (boRF = FALSE) wird die Lagesollwertausgabe sofort gestoppt und dadurch die Geschwindigkeit des Motors maximal verzögert. Abhilfe für Applikationen mit Tieflauframpe: Generieren Sie mit der PLC die Lagesollwerte unter Berücksichtigung der Tieflauframpe. Entziehen Sie erst bei Stillstand die Reglerfreigabe.		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_ Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
boFollowEnable	BOOL	Folgebetrieb Freigabesignal: Mit einer positiven Flanke startet die Funktion Folgebetrieb. Solange 'boFollowEnable' = TRUE ist, wird der Folgebetrieb ausgeführt. Mit einer negativen Flanke 'boFollowEnable' = FALSE wird der laufende Folgebetrieb abgebrochen. Die Achse folgt den Eingangsinkrementen und der übergebenen Tabelle. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Sollwertvorgabe: diInVal Tabellenauswahl: boEnabOpTab2 Pointer auf Tabelle 1: pstOpTab1 Pointer auf Tabelle 2: pstOpTab2		x
boEnabOpTab2	BOOL	• Bei 'boEnabOpTab2' = TRUE wird die zweite Arbeitstabelle freigegeben, die Umschaltung erfolgt im Koordinatenursprung. • Mit 'boEnabOpTab2' = FALSE wird in die Arbeitstabelle1 zurückgeschaltet. Die Umschaltung erfolgt ebenfalls im Koordinatenursprung.	x	
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x

Name	Typ	Beschreibung	Sync.	Async.
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: Inkr/s^2 Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: Inkr/s^2 Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: Inkr/s^3 Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: Inkr/s^3 Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x
diInVal	DINT	Sollwertvorgabe Folgebetrieb (z. B. durch Masterachse, FB VGEN) Einheit: Inkremente	x	
pstOpTab1	POINTER	POINTER TO ST_PROF_TAB Pointer auf die Arbeitstabelle (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)	x	
pstOpTab2	POINTER	POINTER TO ST_PROF_TAB Pointer auf die Arbeitstabelle (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)	x	

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boFollowEnableAck	BOOL	Folgebetrieb Freigabe		x
boTab1Busy	BOOL	pstOpTab1: Tabelle 1 (Kurvenscheibe 1) aktiv	x	
boTab2Busy	BOOL	pstOpTab2: Tabelle 2 (Kurvenscheibe 2) aktiv	x	

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS	STANDARD_AXIS.actSync

4.1.4.2.11 STANDARD_FOLLOW_AXIS_SWITCHING_TABLE_WITH_OUTGEAR (FB)

Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle.
- Am Tabellenende kann eine Tabellenumschaltung durchgeführt werden.
- Die Tabelle kann über ein Getriebeverhältnis angepasste werden (Stauchen und Stecken).

Siehe 'FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2 (FB)' auf Seite 414.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Anwender Interface

		STD_...._WITH_OUTGEAR			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Aktivierung der Tabelle 2	-	**boFollowEnable**	**boFollowEnableAck**	-	Folgebetrieb Freigabe
Steuerung der Tab.funktion	-	**boEnableOpTab2**	**boTab1Busy**	-	Tabelle 1 aktiv
Positioniergeschwindigkeit	-	udPosVelocity	**boTab2Busy**	-	Tabelle 2 aktiv
Zielposition	-	diPosition			
Beschleunigung	-	lreAccel			
Verzögerung	-	lreDecel			
Ruckbegrenzung beim Start	-	lreJerkStart			
Ruckbegrenzung beim Stopp	-	lreJerkEnd			
Tippgeschwindigkeit	-	udVelocityJog			
Drehzahl für Drehzahlr.	-	diSpeedVelocity			
Sollwertvorgabe Folgebetrieb	-	**diInVal**			
Tabellenanp. -multiplikator	-	**iGearMultiplier**			
Tabellenanp. -teiler	-	**uiGearDivider**			
Pointeradr. auf Tabelle 1	-	**pstOpTab1**			
Pointeradr. auf Tabelle 2	-	**pstOpTab2**			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Achsstruktur	-	stAxisDrive	stAxisDrive	-	Achsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

	actSync	
-	synchrone Aktion - Aufruf im synchronen Programmteil (FPLC_PRG)	-
-		-

1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x

Name	Typ	Beschreibung	Sync.	Async.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig Verhalten bei RF Entzug in der Betriebsart Lageregelung mit Folgebetrieb (boFollow = TRUE) und aktiver SAK (ID32801 Bit 9 = TRUE) In der Betriebsart Lageregelung mit aktiver SAK (Schleppabstandskompensation) kann die Funktion Tieflauframpe nach RF Entzug (siehe ID32782 'Tieflaufzeit RF inaktiv') nicht genutzt werden. Bei Entzug der Reglerfreigabe (boRF = FALSE) wird die Lagesollwertausgabe sofort gestoppt und dadurch die Geschwindigkeit des Motors maximal verzögert. Abhilfe für Applikationen mit Tieflauframpe: Generieren Sie mit der PLC die Lagesollwerte unter Berücksichtigung der Tieflauframpe. Entziehen Sie erst bei Stillstand die Reglerfreigabe.		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_ Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
boFollowEnable	BOOL	Folgebetrieb Freigabesignal: Mit einer positiven Flanke startet die Funktion Folgebetrieb. Solange 'boFollowEnable' = TRUE ist, wird der Folgebetrieb ausgeführt. Mit einer negativen Flanke 'boFollowEnable' = FALSE wird der laufende Folgebetrieb abgebrochen. Die Achse folgt den Eingangsinkrementen und der übergebenen Tabelle. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Sollwertvorgabe: diInVal Tabellenauswahl: boEnabOpTab2 Pointer auf Tabelle 1: pstOpTab1 Pointer auf Tabelle 2: pstOpTab2 Tabellenanpassungsmultiplikator: iGearMultiplier Tabellenanpassungsteiler: uiGearDivider	x	
boEnabOpTab2	BOOL	• Bei 'boEnabOpTab2' = TRUE wird die zweite Arbeitstabelle freigegeben, die Umschaltung erfolgt im Koordinatenursprung. • Mit 'boEnabOpTab2' = FALSE wird in die Arbeitstabelle1 zurückgeschaltet. Die Umschaltung erfolgt ebenfalls im Koordinatenursprung.	x	
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x

Name	Typ	Beschreibung	Sync.	Async.
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x
diInVal	DINT	Sollwertvorgabe Folgebetrieb (z. B. durch Masterachse, FB VGEN) Einheit: Inkremente	x	
iGearMultiplier	INT	Tabellenanpassungsmultiplikator zum Stauchen/Strecken der Kurvenscheibenfunktion	x	
uiGearDivider	UINT	Tabellenanpassungsteiler zum Stauchen/Strecken der Kurvenscheibenfunktion	x	
pstOpTab1	POINTER	POINTER TO ST_PROF_TAB Pointer auf die Arbeitstabelle (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)	x	
pstOpTab2	POINTER	POINTER TO ST_PROF_TAB Pointer auf die Arbeitstabelle (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)	x	

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boFollowEnableAck	BOOL	Folgebetrieb Freigabe		x
boTab1Busy	BOOL	pstOpTab1: Tabelle 1 (Kurvenscheibe 1) aktiv	x	
boTab2Busy	BOOL	pstOpTab2: Tabelle 2 (Kurvenscheibe 2) aktiv	x	

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS	STANDARD_AXIS.actSync

4.1.4.2.12 STANDARD_FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2 (FB)

Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle mit Ein- und Auskuppeltabelle.
- Am Tabellenende kann eine Tabellenumschaltung durchgeführt werden.

Siehe 'FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2 (FB)' auf Seite 414.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Anwender Interface

		STD_FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Aktivierung der Tabelle 2	-	**boFollowEnable**	**boFollowEnableAck**	-	Folgebetrieb Freigabe
Folgebetrieb	-	**boTableControl**	**boTab1Busy**	-	Tabelle 1 aktiv
Steuerung der Tab.funktion	-	**boEnableOpTab2**	**boTab2Busy**	-	Tabelle 2 aktiv
Einkuppelbereich	-	**udInAngle**			
Auskuppelbereich	-	**udOutAngle**			
Positioniergeschwindigkeit	-	udPosVelocity			
Zielposition	-	diPosition			
Beschleunigung	-	lreAccel			
Verzögerung	-	lreDecel			
Ruckbegrenzung beim Start	-	lreJerkStart			
Ruckbegrenzung beim Stopp	-	lreJerkEnd			
Tippgeschwindigkeit	-	udVelocityJog			
Drehzahl für Drehzahlr.	-	diSpeedVelocity			
Sollwertvorgabe Folgebetrieb	-	**diInVal**			
Pointeradr. auf Einkuppeltab.	-	**pstInTab**			
Pointeradr. auf Tabelle 1	-	**pstOpTab1**			
Pointeradr. auf Tabelle 2	-	**pstOpTab2**			
Pointeradr. auf Auskuppeltab.	-	**pstOutTab**			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Achsstruktur	-	stAxisDrive	stAxisDrive	-	Achsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

	actSync	
-	synchrone Aktion - Aufruf im synchronen Programmteil (FPLC_PRG)	-
-		-

1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig Verhalten bei RF Entzug in der Betriebsart Lageregelung mit Folgebetrieb (boFollow = TRUE) und aktiver SAK (ID32801 Bit 9 = TRUE) In der Betriebsart Lageregelung mit aktiver SAK (Schleppabstandskompensation) kann die Funktion Tieflauframpe nach RF Entzug (siehe ID32782 'Tieflaufzeit RF inaktiv') nicht genutzt werden. Bei Entzug der Reglerfreigabe (boRF = FALSE) wird die Lagesollwertausgabe sofort gestoppt und dadurch die Geschwindigkeit des Motors maximal verzögert. Abhilfe für Applikationen mit Tieflauframpe: Generieren Sie mit der PLC die Lagesollwerte unter Berücksichtigung der Tieflauframpe. Entziehen Sie erst bei Stillstand die Reglerfreigabe.		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_ Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
boFollowEnable	BOOL	Folgebetrieb Freigabesignal: Mit einer positiven Flanke startet die Funktion Folgebetrieb. Solange 'boFollowEnable' = TRUE ist, wird der Folgebetrieb ausgeführt. Mit einer negativen Flanke 'boFollowEnable' = FALSE wird der laufende Folgebetrieb abgebrochen. Die Achse folgt den Eingangsinkrementen und der übergebenen Tabelle. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Sollwertvorgabe: diInVal Einkuppelbereich: udInVal Auskuppelbereich: udOutVal Tabellenauswahl: boEnabOpTab2 Steuerung der Tabellenfunktion: boTableControl Pointer auf Einkuppeltabelle: pstInTab Pointer auf Auskuppeltabelle: pstOutTab Pointer auf Tabelle 1: pstOpTab1 Pointer auf Tabelle 2: pstOpTab2		x

Name	Typ	Beschreibung	Sync.	Async.
boTableControl	BOOL	Steuerung der Tabellenfunktion Mit 'boTableControl' wird die Abarbeitung der Arbeitstabellen gesteuert. (Voraussetzung: Der Folgebetrieb ist aktiviert 'boFollowEnable' = TRUE und der Masterbezug ist hergestellt). Freigabesignal: Mit einer positiven Flanke an 'boTableControl' startet die Funktion und die Einkuppeltabelle wird abgearbeitet (Voraussetzung: diInVal liegt im Einkuppelbereich) Am Ende der Einkuppeltabelle erfolgt ein automatischer Übergang in die Arbeitstabelle. Solange 'boTableControl' = TRUE ist, wird die Arbeitstabelle abgearbeitet. Mit 'boTableControl' = FALSE wird am Ende der Arbeitstabelle (Voraussetzung: diOutVal liegt im Auskuppelbereich) in die Auskuppeltabelle umgeschaltet. Am Ende der Auskuppeltabelle stoppt die Tabellenabarbeitung automatisch. Der Masterbezug geht dadurch nicht verloren.	x	
boEnabOpTab2	BOOL	• Bei 'boEnabOpTab2' = TRUE wird die zweite Arbeitstabelle freigegeben, die Umschaltung erfolgt im Koordinatenursprung. • Mit 'boEnabOpTab2' = FALSE wird in die Arbeitstabelle1 zurückgeschaltet. Die Umschaltung erfolgt ebenfalls im Koordinatenursprung.	x	
udInAngle	UDINT	Einkuppelwinkel Maximal zulässige Position auf der x-Achse, bis zu der ein Einkuppelvorgang zulässig ist Einheit: Inkremente Standardwert: 1000	x	
udOutAngle	UDINT	Auskuppelwinkel Maximal zulässige Position auf der x-Achse, bis zu der ein Auskuppelvorgang zulässig ist Einheit: Inkremente Standardwert: 1000	x	
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x

Name	Typ	Beschreibung	Sync.	Async.
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: Inkr/s^3 Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: Inkr/s^3 Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x
diInVal	DINT	Sollwertvorgabe Folgebetrieb (z. B. durch Masterachse, FB VGEN) Einheit: Inkremente	x	
pstInTab	POINTER	POINTER TO ST_PROF_TAB Pointer auf die Einkuppeltabelle. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)	x	
pstOpTab1	POINTER	POINTER TO ST_PROF_TAB Pointer auf die Arbeitstabelle (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)	x	
pstOpTab2	POINTER	POINTER TO ST_PROF_TAB Pointer auf die Arbeitstabelle (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)	x	
pstOutTab	POINTER	POINTER TO ST_PROF_TAB Pointer auf die Auskuppeltabelle. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)	x	

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boFollowEnableAck	BOOL	Folgebetrieb Freigabe		x
boTab1Busy	BOOL	pstOpTab1: Tabelle 1 (Kurvenscheibe 1) aktiv	x	
boTab2Busy	BOOL	pstOpTab2: Tabelle 2 (Kurvenscheibe 2) aktiv	x	

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS	STANDARD_AXIS.actSync

4.1.4.2.13 STANDARD_FOLLOW_AXIS_TABLE_CONTINUOUS (FB)

Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle.

Siehe 'FOLLOW_AXIS_TABLE_CONTINUOUS (FB)' auf Seite 416.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Anwender Interface

		STANDARD_FOLLOW_AXIS_TABLE_CONTINUOUS			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Folgebetrieb	-	**boFollowCont**	**boFollowBusy**	-	Folgebetrieb aktiv
Positioniergeschwindigkeit	-	udPosVelocity			
Zielposition	-	diPosition			
Beschleunigung	-	lreAccel			
Verzögerung	-	lreDecel			
Ruckbegrenzung beim Start	-	lreJerkStart			
Ruckbegrenzung beim Stopp	-	lreJerkEnd			
Tippgeschwindigkeit	-	udVelocityJog			
Drehzahl für Drehzahlr.	-	diSpeedVelocity			
Sollwertvorgabe Folgebetrieb	-	**diInVal**			
Pointeradresse auf Tabelle	-	**pstOpTab**			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Achsstruktur	-	stAxisDrive	stAxisDrive	-	Achsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

	actSync	
-	synchrone Aktion - Aufruf im synchronen Programmteil (FPLC_PRG)	-
-		-

1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig Verhalten bei RF Entzug in der Betriebsart Lageregelung mit Folgebetrieb (boFollow = TRUE) und aktiver SAK (ID32801 Bit 9 = TRUE) In der Betriebsart Lageregelung mit aktiver SAK (Schleppabstandskompensation) kann die Funktion Tieflauframpe nach RF Entzug (siehe ID32782 'Tieflaufzeit RF inaktiv') nicht genutzt werden. Bei Entzug der Reglerfreigabe (boRF = FALSE) wird die Lagesollwertausgabe sofort gestoppt und dadurch die Geschwindigkeit des Motors maximal verzögert. Abhilfe für Applikationen mit Tieflauframpe: Generieren Sie mit der PLC die Lagesollwerte unter Berücksichtigung der Tieflauframpe. Entziehen Sie erst bei Stillstand die Reglerfreigabe.		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_ Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
boFollowCont	BOOL	Folgebetrieb Freigabesignal: Mit einer positiven Flanke startet die Funktion Folgebetrieb. Solange 'boFollowCont' = TRUE ist, wird der Folgebetrieb ausgeführt. Mit einer negativen Flanke 'boFollowCont' = FALSE wird der laufende Folgebetrieb abgebrochen. Die Achse folgt den Eingangsinkrementen und der übergebenen Tabelle. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Sollwertvorgabe: diInVal Pointer auf Tabelle: pstOpTab		x
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x

Name	Typ	Beschreibung	Sync.	Async.
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: Inkr/s^2 Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: Inkr/s^3 Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: Inkr/s^3 Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x
diInVal	DINT	Sollwertvorgabe Folgebetrieb (z. B. durch Masterachse, FB VGEN) Einheit: Inkremente	x	
pstOpTab	POINTER	POINTER TO ST_PROF_TAB Pointer auf die Arbeitstabelle (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)	x	

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boFollowBusy	BOOL	Folgebetrieb aktiv		x

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS	STANDARD_AXIS.actSync

4.1.4.2.14 STANDARD_FOLLOW_TABLE_ENGAGE_DISENG_AXIS (FB)

Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle mit Ein- und Auskuppeltabelle.

Siehe 'FOLLOW_AXIS_TABLE_ENGAGE_DISENGAGE (FB)' auf Seite 418.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Anwender Interface

		STD_FOLLOW_TABLE_ENGAGE_DISENG_AXIS			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Folgebetrieb	-	**boFollowEnable**	**boFollowEnabAck**	-	Folgebetrieb Freigabe
Steuerung der Tab.funktion	-	**boTableControl**	**boTableBusy**	-	Tabellenbearbeitung aktiv
Einkuppelbereich	-	**udInAngle**			
Auskuppelbereich	-	**udOutAngle**			
Positioniergeschwindigkeit	-	udPosVelocity			
Zielposition	-	diPosition			
Beschleunigung	-	lreAccel			
Verzögerung	-	lreDecel			
Ruckbegrenzung beim Start	-	lreJerkStart			
Ruckbegrenzung beim Stopp	-	lreJerkEnd			
Tippgeschwindigkeit	-	udVelocityJog			
Drehzahl für Drehzahlr.	-	diSpeedVelocity			
Sollwertvorgabe Folgebetrieb	-	**diInVal**			
Pointeradr. auf Einkuppeltab.	-	**pstInTab**			
Pointeradr. auf Tabelle	-	**pstOpTab**			
Pointeradr. auf Einkuppeltab.	-	**pstOutTab**			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Achsstruktur	-	stAxisDrive	stAxisDrive	-	Achsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

	actSync	
-	synchrone Aktion - Aufruf im synchronen Programmteil (FPLC_PRG)	-
-		-

1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x

Name	Typ	Beschreibung	Sync.	Async.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig Verhalten bei RF Entzug in der Betriebsart Lageregelung mit Folgebetrieb (boFollow = TRUE) und aktiver SAK (ID32801 Bit 9 = TRUE) In der Betriebsart Lageregelung mit aktiver SAK (Schleppabstandskompensation) kann die Funktion Tieflauframpe nach RF Entzug (siehe ID32782 'Tieflaufzeit RF inaktiv') nicht genutzt werden. Bei Entzug der Reglerfreigabe (boRF = FALSE) wird die Lagesollwertausgabe sofort gestoppt und dadurch die Geschwindigkeit des Motors maximal verzögert. Abhilfe für Applikationen mit Tieflauframpe: Generieren Sie mit der PLC die Lagesollwerte unter Berücksichtigung der Tieflauframpe. Entziehen Sie erst bei Stillstand die Reglerfreigabe.		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_ Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
boFollowEnable	BOOL	Folgebetrieb Freigabesignal: Mit einer positiven Flanke startet die Funktion Folgebetrieb. Solange 'boFollowEnable' = TRUE ist, wird der Folgebetrieb ausgeführt. Mit einer negativen Flanke 'boFollowEnable' = FALSE wird der laufende Folgebetrieb abgebrochen. Die Achse folgt den Eingangsinkrementen und der übergebenen Tabelle. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Sollwertvorgabe: diInVal Einkuppelbereich: udInVal Auskuppelbereich: udOutVal Steuerung der Tabellenfunktion: boTableControl Pointer auf Einkuppeltabelle: pstInTab Pointer auf Auskuppeltabelle: pstOutTab Pointer auf Tabelle: pstOpTab		x

Name	Typ	Beschreibung	Sync.	Async.
boTableControl	BOOL	Steuerung der Tabellenfunktion Mit 'boTableControl' wird die Abarbeitung der Arbeitstabellen gesteuert. (Voraussetzung: Der Folgebetrieb ist aktiviert 'boFollowEnable' = TRUE und der Masterbezug ist hergestellt). Freigabesignal: Mit einer positiven Flanke an 'boTableControl' startet die Funktion und die Einkuppeltabelle wird abgearbeitet (Voraussetzung: diInVal liegt im Einkuppelbereich) Am Ende der Einkuppeltabelle erfolgt ein automatischer Übergang in die Arbeitstabelle. Solange 'boTableControl' = TRUE ist, wird die Arbeitstabelle abgearbeitet. Mit 'boTableControl' = FALSE wird am Ende der Arbeitstabelle (Voraussetzung: diOutVal liegt im Auskuppelbereich) in die Auskuppeltabelle umgeschaltet. Am Ende der Auskuppeltabelle stoppt die Tabellenabarbeitung automatisch. Der Masterbezug geht dadurch nicht verloren.	x	
udInAngle	UDINT	Einkuppelwinkel Maximal zulässige Position auf der x-Achse, bis zu der ein Einkuppelvorgang zulässig ist Einheit: Inkremente Standardwert: 1000	x	
udOutAngle	UDINT	Auskuppelwinkel Maximal zulässige Position auf der x-Achse, bis zu der ein Auskuppelvorgang zulässig ist Einheit: Inkremente Standardwert: 1000	x	
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>udVelocityJog</td><td>UDINT</td><td>Tippgeschwindigkeit
Einheit: Inkr/s
Standardwert: 34133</td><td></td><td>x</td></tr>
<tr><td>diSpeedVelocity</td><td>DINT</td><td>Drehzahlsollwert für Drehzahlregelung
Einheit: U/min
Standardwert: 100</td><td></td><td>x</td></tr>
<tr><td>diInVal</td><td>DINT</td><td>Sollwertvorgabe Folgebetrieb
(z. B. durch Masterachse, FB VGEN)
Einheit: Inkremente</td><td>x</td><td></td></tr>
<tr><td>pstInTab</td><td>POINTER</td><td>POINTER TO ST_PROF_TAB
Pointer auf die Einkuppeltabelle.
(Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)</td><td>x</td><td></td></tr>
<tr><td>pstOpTab</td><td>POINTER</td><td>POINTER TO ST_PROF_TAB
Pointer auf die Arbeitstabelle
(Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)</td><td>x</td><td></td></tr>
<tr><td>pstOutTab</td><td>POINTER</td><td>POINTER TO ST_PROF_TAB
Pointer auf die Auskuppeltabelle.
(Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)</td><td>x</td><td></td></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boFollowEnableAck	BOOL	Folgebetrieb Freigabe		x
boTableBusy	BOOL	Tabellenbearbeitung aktiv	x	

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_AXIS	STANDARD_AXIS.actSync

4.1.4.2.15 STANDARD_FOLLOW_TABLE_START_AUTOSTOP_AXIS (FB)

Erweiterte Funktionalität:

- Slave Achse folgt kontinuierlich einer frei wählbaren Tabelle mit Ein- und Auskuppeltabelle.

Siehe 'FOLLOW_AXIS_TABLE_START_AUTOSTOP (FB)' auf Seite 420.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Anwender Interface

		STD_FOLLOW_TABLE_START_AUTOSTOP_AXIS			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Folgebetrieb	-	**boFollowEnable**	**boFollowEnabAck**	-	Folgebetrieb Freigabe
Steuerung der Tab.funktion	-	**boTableControl**	**boTableBusy**	-	Tabellenabarbeitung aktiv
Einkuppelbereich	-	**udInAngle**			
Tabellendurchläufe	-	**uiOpNo**			
Positioniergeschwindigkeit	-	udPosVelocity			
Zielposition	-	diPosition			
Beschleunigung	-	lreAccel			
Verzögerung	-	lreDecel			
Ruckbegrenzung beim Start	-	lreJerkStart			
Ruckbegrenzung beim Stopp	-	lreJerkEnd			
Tippgeschwindigkeit	-	udVelocityJog			
Drehzahl für Drehzahlr.	-	diSpeedVelocity			
Sollwertvorgabe Folgebetrieb	-	**diInVal**			
Pointeradr. auf Einkuppeltab.	-	**pstInTab**			
Pointeradr. auf Tabelle	-	**pstOpTab**			
Pointeradr. auf Einkuppeltab.	-	**pstOutTab**			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Achsstruktur	-	stAxisDrive	stAxisDrive	-	Achsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

	actSync	
-	synchrone Aktion - Aufruf im synchronen Programmteil (FPLC_PRG)	-
-		-

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x

Name	Typ	Beschreibung	Sync.	Async.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig Verhalten bei RF Entzug in der Betriebsart Lageregelung mit Folgebetrieb (boFollow = TRUE) und aktiver SAK (ID32801 Bit 9 = TRUE) In der Betriebsart Lageregelung mit aktiver SAK (Schleppabstandskompensation) kann die Funktion Tieflauframpe nach RF Entzug (siehe ID32782 'Tieflaufzeit RF inaktiv') nicht genutzt werden. Bei Entzug der Reglerfreigabe (boRF = FALSE) wird die Lagesollwertausgabe sofort gestoppt und dadurch die Geschwindigkeit des Motors maximal verzögert. Abhilfe für Applikationen mit Tieflauframpe: Generieren Sie mit der PLC die Lagesollwerte unter Berücksichtigung der Tieflauframpe. Entziehen Sie erst bei Stillstand die Reglerfreigabe.		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_ Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
boFollowEnable	BOOL	Folgebetrieb Freigabesignal: Mit einer positiven Flanke startet die Funktion Folgebetrieb. Solange 'boFollowEnable' = TRUE ist, wird der Folgebetrieb ausgeführt. Mit einer negativen Flanke 'boFollowEnable' = FALSE wird der laufende Folgebetrieb abgebrochen. Die Achse folgt den Eingangsinkrementen und der übergebenen Tabelle. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Sollwertvorgabe: diInVal Einkuppelbereich: udInVal Auskuppelbereich: udOutVal Steuerung der Tabellenfunktion: boTableControl Pointer auf Einkuppeltabelle: pstInTab Pointer auf Auskuppeltabelle: pstOutTab Pointer auf Tabelle: pstOpTab		x

Name	Typ	Beschreibung	Sync.	Async.
boTableControl	BOOL	Steuerung der Tabellenfunktion Mit 'boTableControl' wird die Abarbeitung der Arbeitstabellen gesteuert. (Voraussetzung: Der Folgebetrieb ist aktiviert 'boFollowEnable' = TRUE und der Masterbezug ist hergestellt). Freigabesignal: Mit einer positiven Flanke an 'boTableControl' startet die Funktion und die Einkuppeltabelle wird abgearbeitet (Voraussetzung: diInVal liegt im Einkuppelbereich) Am Ende der Einkuppeltabelle erfolgt ein automatischer Übergang in die Arbeitstabelle. Solange 'boTableControl' = TRUE ist, wird die Arbeitstabelle abgearbeitet. Mit 'boTableControl' = FALSE wird am Ende der Arbeitstabelle (Voraussetzung: diOutVal liegt im Auskuppelbereich) in die Auskuppeltabelle umgeschaltet. Am Ende der Auskuppeltabelle stoppt die Tabellenabarbeitung automatisch. Der Masterbezug geht dadurch nicht verloren.	x	
udInAngle	UDINT	Einkuppelwinkel Maximal zulässige Position auf der x-Achse, bis zu der ein Einkuppelvorgang zulässig ist Einheit: Inkremente Standardwert: 1000		x
uiOpNo	UINT	Anzahl der abzuarbeitenden Arbeitstabellendurchläufe [1]		x
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert lreAccel \rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert lreDecel \rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>diSpeedVelocity</td><td>DINT</td><td>Drehzahlsollwert für Drehzahlregelung
Einheit: U/min
Standardwert: 100</td><td></td><td>x</td></tr>
<tr><td>diInVal</td><td>DINT</td><td>Sollwertvorgabe Folgebetrieb
(z. B. durch Masterachse, FB VGEN)
Einheit: Inkremente</td><td>x</td><td></td></tr>
<tr><td>pstInTab</td><td>POINTER</td><td>POINTER TO ST_PROF_TAB
Pointer auf die Einkuppeltabelle.
(Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)</td><td></td><td>x</td></tr>
<tr><td>pstOpTab</td><td>POINTER</td><td>POINTER TO ST_PROF_TAB
Pointer auf die Arbeitstabelle
(Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)</td><td></td><td>x</td></tr>
<tr><td>pstOutTab</td><td>POINTER</td><td>POINTER TO ST_PROF_TAB
Pointer auf die Auskuppeltabelle.
(Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)</td><td></td><td>x</td></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung		x
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boFollowEnableAck	BOOL	Folgebetrieb Freigabe	x	
boTableBusy	BOOL	Tabellenbearbeitung aktiv	x	

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_FOLLOW_TABLE_START_AUTOSTOP_AXIS	STANDARD_FOLLOW_TABLE_START_AUTOSTOP_AXIS.actSync

4.1.4.2.16 STANDARD_WINDER_DANCER_AXIS (FB)

Erweiterte Funktionalität:

- Der Funktionsbaustein 'STANDARD_WINDER_DANCER_AXIS' realisiert einen Tänzerwickler mit folgenden Merkmalen:
- PID geregelter Tänzerwickler
- Betrieb als Auf- und Abwickler

- Durchmesserberechnung
- Durchmessersuche bei Start des Wicklers

Siehe 'WINDER_DANCER_CONTROL (FB)' auf Seite 568.

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

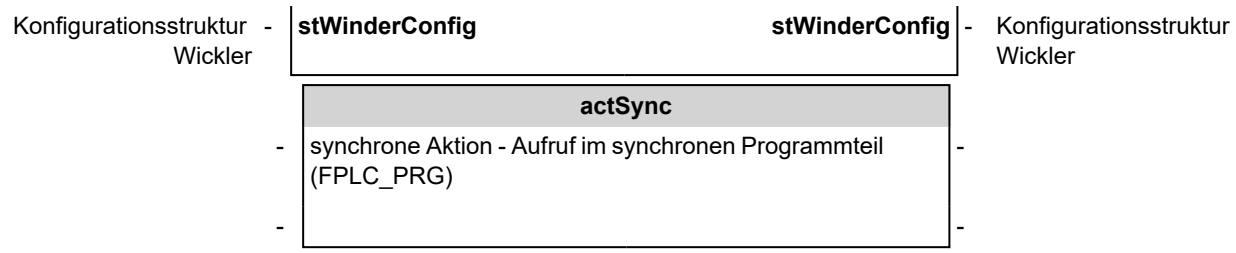
Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Anwender Interface

		STANDARD_WINDER_DANCER_AXIS			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Drehzahlregelung starten	-	boSpeed	iErrID	-	Fehler Identnummer
Referenzfahrt starten	-	boHome	enErrName	-	Name fehlerh Baustein
Absolute Pos. starten	-	boPosAbs	boPosBusy	-	Positionierung aktiv
Relative Pos. starten	-	boPosRel	boJogBusy	-	Tippbetrieb aktiv
Tippen Plus	-	boJogPlus	boHomeBusy	-	Referenzfahrt aktiv
Tippen Minus	-	boJogMinus	boSpeedBusy	-	Drehzahlregelung aktiv
Positioniergeschwindigkeit	-	udPosVelocity	**boEnabAcknWinder**	-	Wicklerbaustein initialisiert und aktiviert
Zielposition	-	diPosition	**boDmOk**	-	Durchmesser ok, Status "bekannt"
Beschleunigung	-	lreAccel	**boDmLimit**	-	Durchmesserwert wurde begrenzt
Verzögerung	-	lreDecel	**iActDm**	-	Istwert Durchmesser
Ruckbegrenzung beim Start	-	lreJerkStart	**diVWebSpeed**	-	Istwert Bahngeschwindigkeit
Ruckbegrenzung beim Stopp	-	lreJerkEnd	**diSetValMotorSpeed**	-	Drehzahlsollwert Motor
Tippgeschwindigkeit	-	udVelocityJog			
Drehzahl für Drehzahlr.	-	diSpeedVelocity			
Freigabesignal Wickler	-	**boEnableWinder**			
Aktivierung der Durchmesserrechnung	-	**boDmCalc**			
Tänzer in Mittenposition	-	**boDncMidPos**			
Tänzermodus "Istwert folgen"	-	**boDncTraceMod**			
Aktivieren Verstärkungsrampe	-	**boRaiseGain**			
Aktivieren Ausgabe Bahngeschwindigkeit	-	**boOutputSpeed**			
Durchmesser setzen, Status "bekannt"	-	**boDmOkPreset**			
Durchmesser setzen, Status "unbekannt"	-	**boDmNokPreset**			
Durchmesser Status von "Bekannt" → "Unbekannt" setzen	-	**boDmReset**			
Ausgang 'boDmOk' setzen	-	**boSetDmOk**			
Tänzerregler I Anteil löschen	-	**boClearI**			
Durchmesservorgabewert	-	**iDmPreset**			
Istwert Rollenposition	-	**diActReelPos**			
Istwert Bahnposition	-	**diActWebPos**			
Istwert Tänzerposition	-	**iActDncPos**			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Achsstruktur	-	stAxisDrive	stAxisDrive	-	Achsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur



1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_ Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x

Name	Typ	Beschreibung	Sync.	Async.
boEnableWinder	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Wicklerbausteins. Solange 'boEnableWinder' = TRUE ist, bleibt der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnableWinder' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boDmCalc	BOOL	Aktivierung der Durchmesserberechnung		x
boDncMidPos	BOOL	Tänzer in die Mittenposition bewegen		x
boDncTraceMod	BOOL	Tänzermodus Istwert folgen, der Tänzeristwert wird als Sollwert am PID Regler übernommen. Nur Aktivierbar bei 'boDncMidPos' = FALSE		x
boRaiseGain	BOOL	Lageabhängige Tänzer-Verstärkungsrampe aktiv		x
boOutputSpeed	BOOL	Ausgabe Bahngeschwindigkeit als Drehzahlsollwert für Motor		x
boDmOkPreset	BOOL	Durchmesser auf Wert der Eingangsvariable 'iDmPreset' setzen. Das Eingangsbit 'boDmPreset' wird nach setzen automatisch rückgesetzt. Durchmesserstatus "Bekannt": Der Durchmesserwert wird als gültiger Durchmesser gewertet, das bedeutet, Beschleunigungs- und Reibmomente werden berechnet.		x
boDmNokPreset	BOOL	Durchmesser auf Wert der Eingangsvariable 'iDmPreset' setzen. Das Eingangsbit 'boDmPreset' wird nach setzen automatisch rückgesetzt. Durchmesserstatus "Unbekannt": Der Durchmesserwert wird als ungültiger Durchmesser gewertet, das bedeutet, Beschleunigungs- und Reibmomente werden berechnet.		x
boDmReset	BOOL	Ausgang 'boDmOk' rücksetzen, → Durchmesserstatus "Unbekannt"		x
boSetDmOk	BOOL	Ausgang 'boDmOk' setzen, → Durchmesserstatus "Bekannt"		x
boClearI	BOOL	Tänzerregler: I-Anteil löschen		x
iDmPreset	INT	Durchmesservorgabewert Wert auf den der Durchmesser mit 'boDmNokPreset' = TRUE oder 'boDmOkPreset' = TRUE gesetzt wird. [mm]		x
diActReelPos	DINT	Istposition Rolle [Inkremente]	x	
diActWebPos	DINT	Istposition Bahn [Inkremente]	x	
iActDncPos	INT	Istwert Tänzerposition [mV]	x	

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_ Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung Busfehler wenn 'boErr' = TRUE: EtherCAT oder CAN nicht in Operation Mode.	x	
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boEnabAcknWinder	BOOL	Quittierung: Wickler ist initialisiert und freigegeben		x
boDmOk	BOOL	Durchmesser bekannt		x
boDmLimit	BOOL	Durchmesser wurde berechnet und das Ergebnis lag außerhalb von 'ST_WINDER_DANCER_CONTROL_CONFIG.iDmMin' und 'ST_WINDER_DANCER_CONTROL_CONFIG.iDmMax', der Wert wurde entsprechend begrenzt		x
iActDm	INT	Aktueller Durchmesser [mm]		x
diVWebSpeed	DINT	Istwert Bahngeschwindigkeit [mm/s]	x	
diSetValMotorSpeed	DINT	Drehzahlsollwert Motor [0,0001 U/min]	x	

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x
stWinderConfig	STRUCT	ST_WINDER_DANCER_CONTROL_CONFIG Konfigurationsstruktur des Wicklers (Siehe Dokument Softwarebeschreibung AFL - AMK Funktionsbibliothek Teil 2, Teile-Nr. 203694)		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_WINDER_DANCER_AXIS	STANDARD_WINDER_DANCER_AXIS.actSync

4.1.4.2.17 STANDARD_WINDER_TORQUE_AXIS (FB)

Gültig für AFL-CODESYS > V3_3.5.5.0

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt
- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Erweiterte Funktionalität:

- Der Funktionbaustein 'STANDARD_WINDER_TORQUE_AXIS' realisiert einen sensorlosen Auf- und Abwickler und konfigurierbarer Bahnspannung mit folgenden Merkmalen:
- Momentgesteuerter Zentrumswickler
- Betrieb als Auf- und Abwickler
- Durchmesserberechnung
- Durchmessersuche beim Start des Wicklers
- Ermittlung des Rollenträgheitsmoments bei unbekanntem Durchmesser
- Synchronisieren auf eine laufende Bahn

Beschreibung

Der STANDARD_WINDER_TORQUE_AXIS (FB) ist ein Momentwickler.

Mit dem Wickelbaustein kann ein sensorloser Auf- oder Abwickler mit konfigurierbarer Bahnspannung realisiert werden. In vielen Wickelapplikationen muss ein prozess- und materialabhängige Bahnspannung eingehalten werden. Die Bahnspannung wird durch die Bahnspannungsregelung unabhängig von der aktuellen Bahngeschwindigkeit konstant gehalten. Die Bahngeschwindigkeit wird durch den Vorzugsantrieb bestimmt. Der Drehmomentsollwert für den Wickelantrieb setzt sich aus den Faktoren Bahngeschwindigkeit, Reib- und Trägheitsmoment sowie dem geforderten Bahnspannung zusammen.

Die Anforderung an die Bahnspannungsregelung ist abhängig von folgenden Punkten:

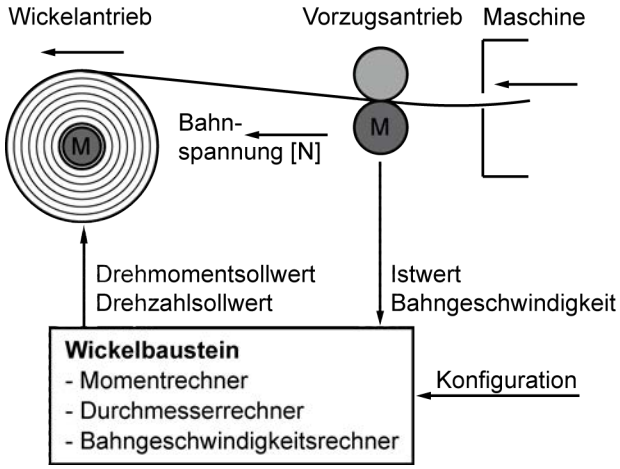
- Fertigungsprozess
- Physikalische Eigenschaften des Produktes
- Bahngeschwindigkeit

Die Bahnspannung muss an den Bearbeitungsprozess genau angepasst werden. Eine zu geringe Bahnspannung führt zu schlechten Wicklungslagen oder zum Abgleiten des Wickelgutes von den Führungsrollen. Eine zu hohe Bahnspannung dehnt das Wickelgut und kann dadurch die physikalischen Eigenschaften verändern bzw. das Wickelgut abreißen.

Während des Wickelvorgangs verändert sich der Wicklungsdurchmesser und mit ihm das Gewicht und die Drehzahl. Die Bahnspannungsregelung muss unabhängig von den veränderlichen Größen die Bahnspannung im zulässigen Bereich halten.

Eine weitere Anforderung wird durch die Bahngeschwindigkeit bestimmt. Je schneller die Bahngeschwindigkeit wird, umso dynamischer muss auch die Bahnspannungsregelung arbeiten.

Funktionsweise Momentwickler, Betriebsart Aufwickeln



Beim Wickeln wird vom Wickelbaustein ein Drehzahlsollwert und ein Drehmomentsollwert an den Wickelantrieb ausgegeben. Unabhängig von der aktuellen Bahngeschwindigkeit, wird die Bahn immer mit dem Sollwert für die Bahnspannung gestrafft.

Der Wickelbaustein erhält als Eingangssignal den Istwert Bahngeschwindigkeit und die parametrierten Werte aus der Konfiguration (ST_WINDER_TORQUE_CONTROL_CONFIG (ST)).

Bestandteile des Wickelbausteins:

Momentrechner

Der Momentrechner berechnet das Grenzmoment des Wickelantriebs. Dieses besteht aus mehreren Anteilen:

- Moment zur Überwindung des Trägheitsmoments der Rolle
- Moment zur Reibungskompensation
- Moment zur Erzeugung der Bahnspannung

Diese Momente werden addiert und als Drehmomentsollwert an den Wickelantrieb ausgegeben.

Bahngeschwindigkeitsrechner

Der Bahngeschwindigkeitsrechner berechnet aus dem Istwert Bahngeschwindigkeit die Bahngeschwindigkeit. Mit Hilfe des Rollendurchmessers wird aus der Bahngeschwindigkeit der Drehzahlsollwert des Wickelantriebs berechnet. Zu diesem Drehzahlsollwert wird ein Offset addiert, so dass der Wickelantrieb die Bahn immer strafft.

Durchmesserrechner

Der Durchmesserrechner berechnet während des Wickelvorgangs den aktuellen Rollendurchmesser.

Der Bahngeschwindigkeitsrechner benötigt den Rollendurchmesser zur Berechnung des richtigen Drehzahlsollwerts für den Wickelantrieb.

Damit der Rollendurchmesser richtig berechnet wird, muss die Bahn zwischen Rolle und Vorzugsantrieb gespannt sein.

Der Rollendurchmesser wird nach einem parametrierbaren Intervall berechnet. Die Rolle muss mindestens dieses Intervall gedreht haben, bevor eine Durchmesserberechnung durchgeführt wird. Die Größe des Intervalls stellt ein Kompromiss zwischen Genauigkeit und Frequenz der Berechnung dar.

Geeignete Anwendungen für den Momentwickler

Durch die in der Konfigurationsstruktur (ST_WINDER_TORQUE_CONTROL_CONFIG (ST)) enthaltenen Parameter werden die Einflussfaktoren (z. B. Reibung und Masseträgheit) in die Berechnung des Motormoments einbezogen. Je genauer die Parameter vom Anwender vorgegeben werden, umso kleiner wird die Toleranz bei der Bahnspannung ausfallen.

Einsatzbereich Momentwickler:

- Weite Toleranzen der Bahnspannung sind zulässig
- Wickelgut besitzt eine hohe Zugfestigkeit
- Geringe Änderung des Umfangs beim Wickeln

Die besten Wickelergebnisse sind bei folgenden Anwendungen zu erwarten:

- Die Bahnspannung ist größer als die Summe von Reibungs- und Massenträgheitsmoment
- Durchmesser der Wickelspule ist bekannt
- Konstante Bahngeschwindigkeit ohne dynamische Beschleunigungsrampen

Der Momentwickler ist kostengünstiger als ein Tänzerwickler zu realisieren. Da er sensorlos und ohne Tänzer arbeitet entstehen gewisse Einschränkungen.

Ein Materialpuffer (Tänzer) für externe Störeinflüsse ist nicht vorhanden.

Die Bahnspannungsregelung kann dynamischen Beschleunigungsvorgängen nur schwer folgen. Zudem wirkt sich eine Regelabweichung des Antriebs direkt auf die Bahnspannung aus. Daher sind Momentwickler für sehr genaue oder sehr kleine Bahnspannungsanforderungen ungeeignet.

In folgenden Fällen erreicht ein Tänzerwickler bessere Wickelergebnisse:

- wenn die Bahnspannung kleiner ist als die Summe von Reibungs- und Massenträgheitsmoment
- wenn der Bearbeitungsprozess eine konstante Bahnspannung benötigt

Funktionsbeschreibung Momentwickler

Folgende Funktionen sind im Momentwickler vorhanden:

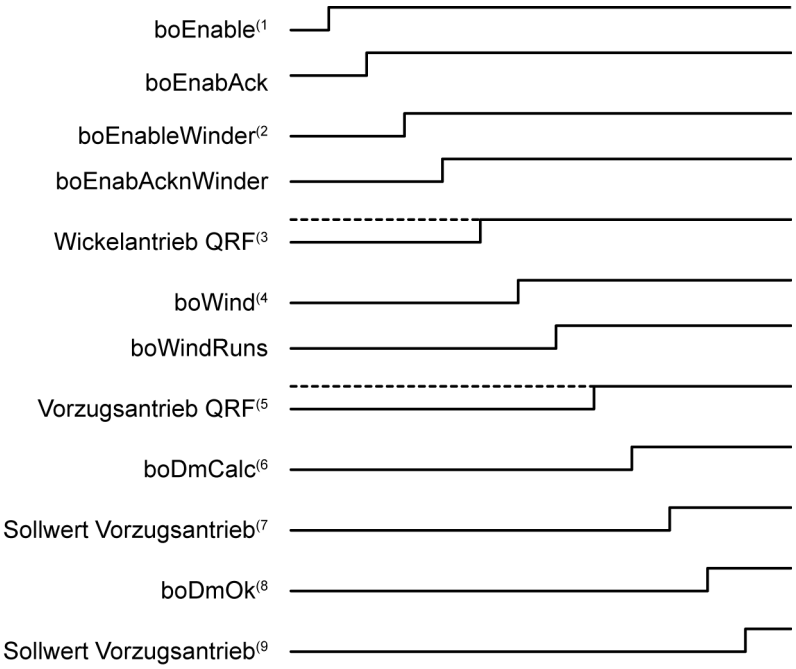
- Modus 1: Wickeln ohne Anfangsdurchmesser und Bahnspannung > Reibungs- und Massenträgheitsmoment
- Modus 2: Wickeln ohne Anfangsdurchmesser und Bahnspannung < Reibungs- und Massenträgheitsmoment
- Modus 3: Wickeln mit bekanntem Anfangsdurchmesser
- Modus 4: Wickeln mit geschätztem Anfangsdurchmesser
- Modus 5: Wickeln mit Durchmessersensor
- Modus 6 : Synchronisieren eines Wickels (z. B. zum Rollenwechsel) auf eine laufende Bahn

Modus 1: Wickeln ohne Anfangsdurchmesser und Bahnspannung > Reibungs- und Massenträgheitsmoment

Der Anfangsdurchmesser ist nicht bekannt. Der Wicklerbaustein ermittelt den Durchmesser.

(Schafft es der Wickler nicht die Bahn mit der eingestellten Bahnspannung zu straffen, kann das Reibungs- und Massenträgheitsmoment ermittelt werden. Siehe Modus 2)

Ablauf:



1) STANDARD_WINDER_TORQUE_AXIS (FB) aktivieren und Quittierung abwarten
2) Wickelfunktion aktivieren und Quittierung abwarten
3) Voraussetzung QRF Wickelantrieb = TRUE
4) **Rückkopplungsgefahr**
   Die Bahn darf den Vorzugsantrieb nicht bewegen.

   Wickeln Ein und Quittierung abwarten

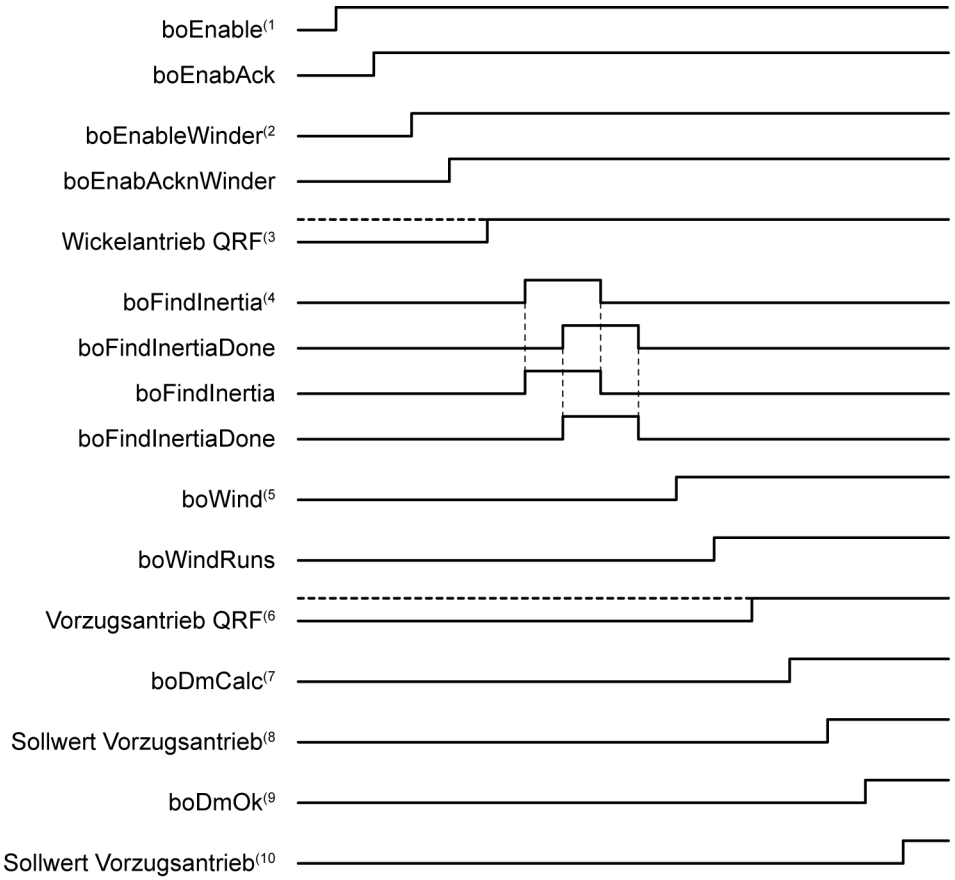
5 Voraussetzung: QRF Wickelantrieb = TRUE
6 Durchmesserberechnung aktivieren
7 Vorzugsantrieb langsam beschleunigen und mit niedriger Drehzahl die Quittierung 'boDmOk' = TRUE (Durchmesser gefunden) abwarten.
   Der Wickler strafft die Bahn unter Berücksichtigung der Variablen 'iDmMin' minimal zulässiger Durchmesserwert [mm] und 'diWebTension' Sollwert Bahnspannung.
8 'boDmOk' = TRUE; Normalbetrieb
9) Vorzugsantrieb langsam auf Prozessdrehzahl beschleunigen

Die Konfigurationsstruktur des Wicklers muss an die Anwendung angepasst werden.
ST_WINDER_TORQUE_CONTROL_CONFIG
(Siehe DokumentSoftwarebeschreibungAFL - AMK Funktionsbibliothek Teil 2, Teile-Nr. 203694)

Modus 2: Wickeln ohne Anfangsdurchmesser und Bahnspannung < Reibungs- und Massenträgheitsmoment

Wenn die eingestellte Bahnspannung kleiner ist als das Reibungs- und Massenträgheitsmoment des Wicklers kann die Bahn nicht gestrafft werde. In diesem Fall kann mit Modus 2 das Reibungs- und Massenträgheitsmoment ermittelt werden.

Ablauf:



1) STANDARD_WINDER_TORQUE_AXIS (FB) aktivieren und Quittierung abwarten
2) Wickelfunktion aktivieren und Quittierung abwarten
3) Voraussetzung QRF Wickelantrieb = TRUE
4) Massenträgheitsmoment ermitteln

Das Massenträgheitsmoment wird bestimmt, indem der Wickel eine Umdrehung zurückdreht und dadurch eine Schlaufe bildet. Danach wird schrittweise das Drehmoment von Null aus erhöht, bis sich die Rolle wieder dreht und die Bahn straff zieht.

Die Funktion ist mit 'boFindInertiaDone' = TRUE abgeschlossen. Der Eingang 'boFindInertia' muss anschließend zurückgesetzt werden.

Die Funktion wird abgebrochen, wenn der Eingang vor der Rückmeldung zurückgesetzt wird.

Der Vorzugsantrieb muss bei der Ermittlung des Masseträgheitsmoments auf der aktuellen Position gehalten werden. Die Bahn darf durch den Zug des Wicklers nicht den Vorzug bewegen bzw. zwischen dem Vorzug und der Andruckwalze durchrutschen.

Der Vorzugsantrieb kann durch folgende Maßnahmen auf der aktuellen Position gehalten werden:

- Motor im Stillstand bestromen (QRF)
- Motorhaltebremse (optional) schließen

5)

Rückkopplungsgefahr

Die Bahn darf den Vorzugsantrieb nicht bewegen.

Wickeln Ein und Quittierung abwarten

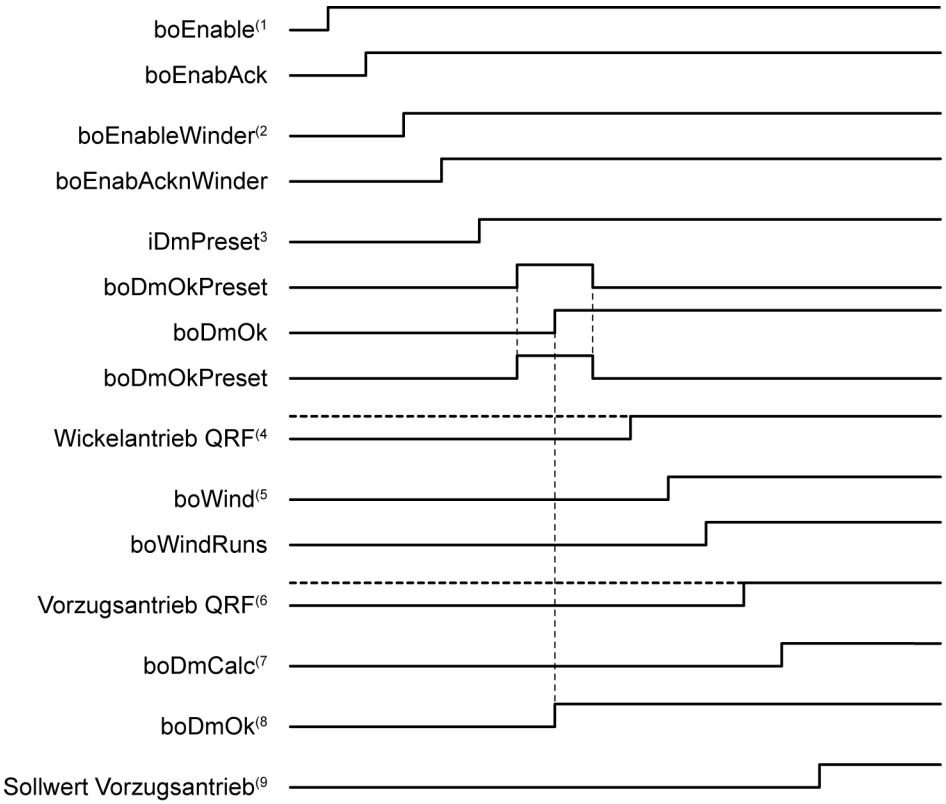
6 Voraussetzung: QRF Wickelantrieb = TRUE
7 Durchmesserberechnung aktivieren

8 Vorzugsantrieb langsam beschleunigen und mit niedriger Drehzahl die Quittierung 'boDmOk' = TRUE (Durchmesser gefunden) abwarten.
Der Wickler strafft die Bahn unter Berücksichtigung der Variablen 'iDmMin' minimal zulässiger Durchmesserwert [mm] und 'diWebTension' Sollwert Bahnspannung.
9) 'boDmOk' = TRUE; Normalbetrieb
10) Vorzugsantrieb langsam auf Prozessdrehzahl beschleunigen

Die Konfigurationsstruktur des Wicklers muss an die Anwendung angepasst werden.
ST_WINDER_TORQUE_CONTROL_CONFIG
(Siehe DokumentSoftwarebeschreibungAFL - AMK Funktionsbibliothek Teil 2, Teile-Nr. 203694)

Modus 3: Wickeln mit bekanntem Anfangsdurchmesser
Der Wickler strafft die Bahn mit dem für den Durchmesser berechneten Moment.

Ablauf:



1) STANDARD_WINDER_TORQUE_AXIS (FB) aktivieren und Quittierung abwarten
2) Wickelfunktion aktivieren und Quittierung abwarten
3) Wickeldurchmesser auf bekannten Wert setzen und Quittierung abwarten.
Vorgabe: 'iDmPreset' = Wickeldurchmesser [mm]
Übernahme: 'boDmOkPreset' = TRUE → Quittierung 'boDmOk' = TRUE
Nach der Quittierung muss der Eingang 'boDmOkPreset' zurückgesetzt werden.
3) Voraussetzung QRF Wickelantrieb = TRUE
5)

Rückkopplungsgefahr
Die Bahn darf den Vorzugsantrieb nicht bewegen.

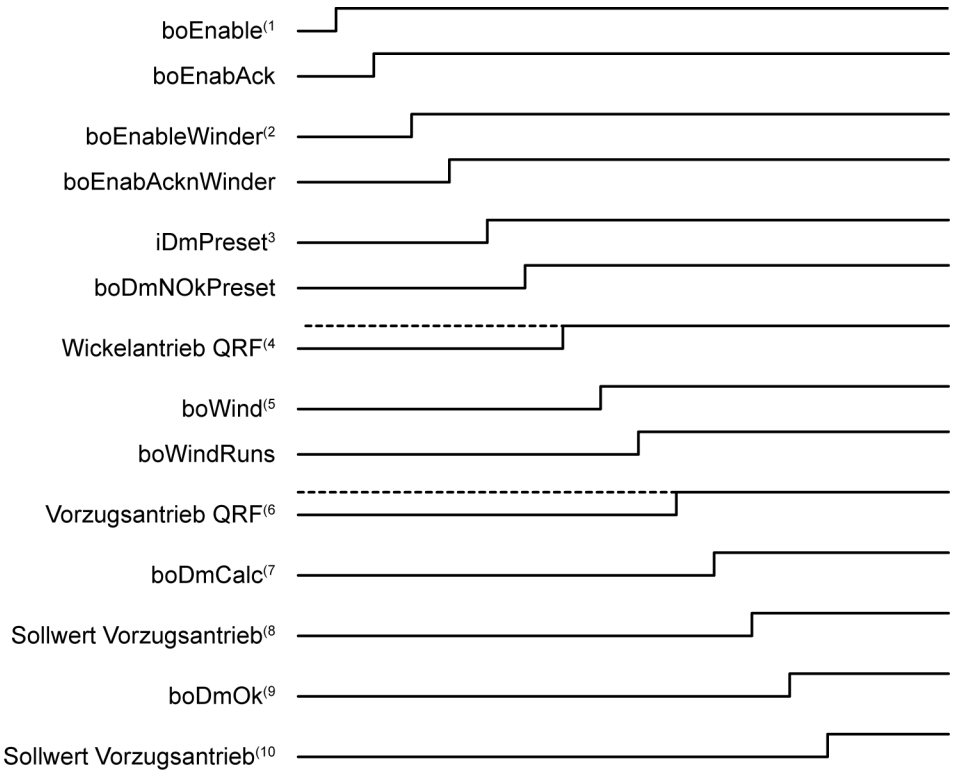
Wickeln Ein und Quittierung abwarten

6 Voraussetzung: QRF Wickelantrieb = TRUE
Der Wickler strafft die Bahn mit dem für den Durchmesser berechneten Drehmoment.
7 Durchmesserberechnung aktivieren
8 'boDmOk' = TRUE; Normalbetrieb
9) Vorzugsantrieb langsam auf Prozessdrehzahl beschleunigen

Die Konfigurationsstruktur des Wicklers muss an die Anwendung angepasst werden.
ST_WINDER_TORQUE_CONTROL_CONFIG
(Siehe DokumentSoftwarebeschreibungAFL - AMK Funktionsbibliothek Teil 2, Teile-Nr. 203694)

Modus 4: Wickeln mit geschätztem Anfangsdurchmesser
Die Sollwerte für die Durchmessersuche basieren auf dem geschätztem Anfangsdurchmesser

Ablauf:



1) STANDARD_WINDER_TORQUE_AXIS (FB) aktivieren und Quittierung abwarten
2) Wickelfunktion aktivieren und Quittierung abwarten
3) Wickeldurchmesser auf geschätzten Wert setzen
   Vorgabe: 'iDmPreset' = Wickeldurchmesser [mm]
   Übernahme: 'boDmNOkPreset' = TRUE
4) Voraussetzung QRF Wickelantrieb = TRUE
5) **Rückkopplungsgefahr**
   Die Bahn darf den Vorzugsantrieb nicht bewegen.

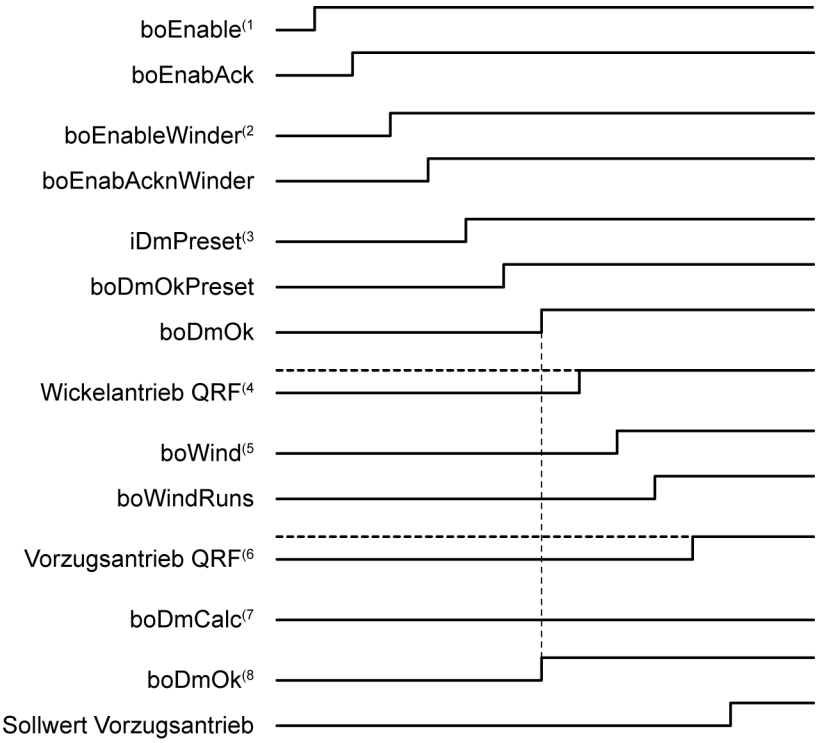
   Wickeln Ein und Quittierung abwarten

   Aus Sicherheitsgründen strafft der Wickler die Bahn mit dem für min DM geltenden Drehmoment.
   Für die Drehzahlausgabe wird der geschätzte Durchmesser als Grundlage verwendet.
6 Voraussetzung: QRF Wickelantrieb = TRUE
7 Durchmesserberechnung aktivieren
8 Durchmesser suchen und Quittierung abwarten
   Vorzugsantrieb langsam beschleunigen und mit niedriger Drehzahl die Quittierung 'boDmOk' = TRUE (Durchmesser gefunden) abwarten.
9) 'boDmOk' = TRUE; Normalbetrieb
10) Vorzugsantrieb langsam auf Prozessdrehzahl beschleunigen

Die Konfigurationsstruktur des Wicklers muss an die Anwendung angepasst werden.
ST_WINDER_TORQUE_CONTROL_CONFIG
(Siehe DokumentSoftwarebeschreibungAFL - AMK Funktionsbibliothek Teil 2, Teile-Nr. 203694)

Modus 5: Wickeln mit Durchmessersensor
Der Wickeldurchmesser wird mit einem Durchmessersensor gemessen.

Ablauf:



1) STANDARD_WINDER_TORQUE_AXIS (FB) aktivieren und Quittierung abwarten
2) Wickelfunktion aktivieren und Quittierung abwarten
3) Messwert Durchmessersensor ständig übergeben
   Übergabe: 'iDmPreset' = Messwert Durchmessersensor [mm]
   Übernahme: 'boDmOkPreset' = TRUE → Quittierung 'boDmOk' = TRUE
4) Voraussetzung QRF Wickelantrieb = TRUE
5) **Rückkopplungsgefahr**
   Die Bahn darf den Vorzugsantrieb nicht bewegen.

   Wickeln Ein und Quittierung abwarten

   Der Wickler strafft die Bahn mit dem für den Durchmesser berechneten Drehmoment.

6 Voraussetzung: QRF Wickelantrieb = TRUE
7 Die Durchmesserberechnung muss deaktiviert sein, da der Durchmesser gemessen und nicht berechnet wird.

   Durchmesserberechnung deaktivieren
   'boDmCalc' = FALSE

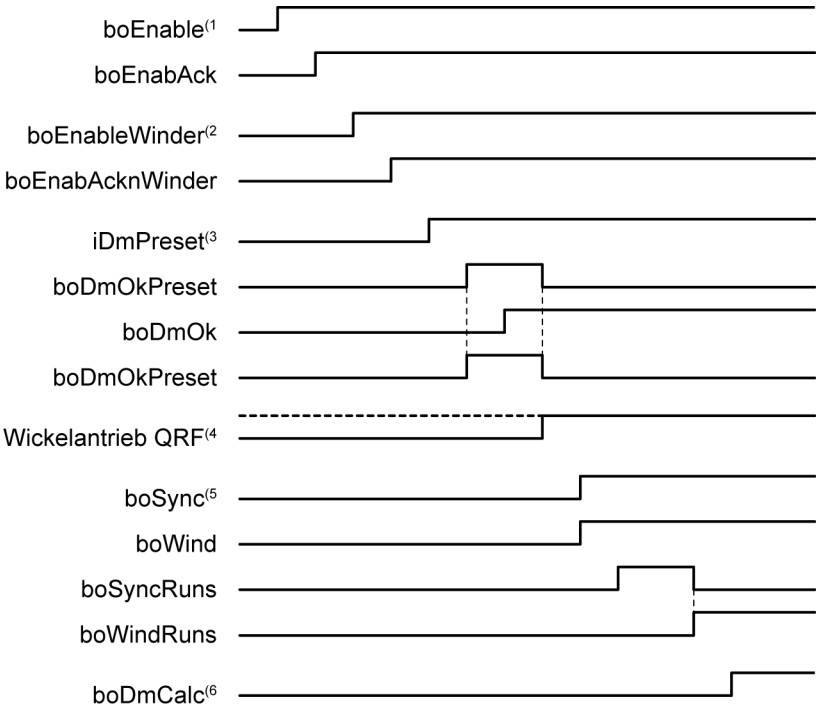
8) 'boDmOk' = TRUE; Normalbetrieb
   Vorzugsantrieb langsam auf Prozessdrehzahl beschleunigen

Die Konfigurationsstruktur des Wicklers muss an die Anwendung angepasst werden.
ST_WINDER_TORQUE_CONTROL_CONFIG
(Siehe DokumentSoftwarebeschreibungAFL - AMK Funktionsbibliothek Teil 2, Teile-Nr. 203694)

Modus 6: Synchronisieren eines Wickels (z. B. zum Rollenwechsel) auf eine laufende Bahn

Der Wickelantrieb (leer) beschleunigt auf die Bahngeschwindigkeit. Nachdem die Synchronisierung abgeschlossen ist, kann ein Bahnwechsel erfolgen.

Ablauf:



1) STANDARD_WINDER_TORQUE_AXIS (FB) aktivieren und Quittierung abwarten
2) Wickelfunktion aktivieren und Quittierung abwarten
3) Wickeldurchmesser auf bekannten Wert setzen und Quittierung abwarten.
   Vorgabe: 'iDmPreset' = Wickeldurchmesser [mm]
   Übernahme: 'boDmOkPreset' = TRUE → Quittierung 'boDmOk' = TRUE
   Nach der Quittierung muss der Eingang 'boDmOkPreset' zurückgesetzt werden.
3) Voraussetzung QRF Wickelantrieb = TRUE
5) **Rückkopplungsgefahr**
   Die Bahn darf den Vorzugsantrieb nicht bewegen.

   Wickeln Ein und Quittierung abwarten

   Voraussetzung: QRF Wickelantrieb = TRUE
   Synchronisierung starten
   'boSync' = TRUE
   'boWind' = TRUE

   Der Wickelantrieb (leer) beschleunigt auf die Bahngeschwindigkeit. Quittierung 'boSyncRuns' = TRUE
   Synchronisierung abgeschlossen, Bahnwechsel möglich.
   Nachdem der Wickelantrieb (leer) auf die Bahngeschwindigkeit aufsynchronisiert hat schaltet er automatisch in den Wickelbetrieb
   Quittierung Wickelbetrieb aktiv, Bahnwechsel vom vollen auf den leeren Wickel möglich
   boSyncRuns' = FALSE
   boWindRuns = TRUE

6 Durchmesserberechnung aktivieren
,nachdem die Bahn am neuen Wickel angeklebt ist.

Die Konfigurationsstruktur des Wicklers muss an die Anwendung angepasst werden.
ST_WINDER_TORQUE_CONTROL_CONFIG
(Siehe DokumentSoftwarebeschreibungAFL - AMK Funktionsbibliothek Teil 2, Teile-Nr. 203694)

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Anwender Interface

		STANDARD_WINDER_TORQUE_AXIS			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Reglerfreigabe	-	boRF	boSBM	-	Sammelbereitmeldung
Fehler löschen	-	boFL	boQRF		Quittierung "Reglerfreigabe"
Not-Halt	-	boEmergency_Stop	boDone	-	Quittierung "Ausgeführt"
Drehzahlregelung starten	-	boSpeed	boErr	-	Fehlersignal
Referenzfahrt starten	-	boHome	iErrID	-	Fehler Identnummer
Absolute Pos. starten	-	boPosAbs	enErrFbName	-	Name fehlerh Baustein
Relative Pos. starten	-	boPosRel	strErrName	-	Fehler Baustein
Tippen Plus	-	boJogPlus	strFbInstance	-	Instanznamen
Tippen Minus	-	boJogMinus	boPosBusy	-	Positionierung aktiv
Positioniergeschwindigkeit	-	udPosVelocity	boJogBusy	-	Tippbetrieb aktiv
Zielposition	-	diPosition	boHomeBusy	-	Referenzfahrt aktiv
Beschleunigung	-	lreAccel	boSpeedBusy	-	Drehzahlregelung aktiv
Verzögerung	-	lreDecel	**boEnabAcknWinder**	-	Wicklerbaustein initialisiert und aktiviert
Ruckbegrenzung beim Start	-	lreJerkStart	**boDmOk**	-	Durchmesser ok, Status "bekannt"
Ruckbegrenzung beim Stopp	-	lreJerkEnd	**boFindInertiaDone**	-	Trägheitsmoment ermittelt
Tippgeschwindigkeit	-	udVelocityJog	**boWindRuns**	-	Wickeln läuft
Drehzahl für Drehzahlr.	-	diSpeedVelocity	**boSyncRuns**	-	Synchronisation läuft
Freigabesignal Wickler	-	**boEnableWinder**	**boDmLimit**	-	Durchmesserwert wurde begrenzt
Aktivierung der Durchmesserrechnung	-	**boDmCalc**	**iActDm**	-	Istwert Rollendurchmesser
Ermittlung Rollenträgheitsmoment ein	-	**boFindInertia**	**diActValWebSpeed**	-	Istwert Bahngeschwindigkeit
Wickeln ein	-	**boWind**			
Synchronisieren ein	-	**boSync**			
Durchmesser setzen, Status "bekannt"	-	**boDmOkPreset**			
Durchmesser setzen, Status "unbekannt"	-	**boDmNOkPreset**			
Sollwert Bahnspannung	-	**diWebTension**			
Durchmesservorgabewert	-	**iDmPreset**			
Eingangsinkremente Bahntransport (Vorzug)	-	**diInVal**			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Achsstruktur	-	stAxisDrive	stAxisDrive	-	Achsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur
Konfigurationsstruktur Wickler	-	**stWinderConfig**	**stWinderConfig**	-	Konfigurationsstruktur Wickler

actSync	
synchrone Aktion - Aufruf im synchronen Programmteil (FPLC_PRG)	-
	-

Eingangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe Relevante Parameter: ID32796 'Quelle Reglerfreigabe' Einstellung: Code 5: 'RF über Feldbus' Code 25: 'RF über EtherCAT UND-verknüpft mit dem Binäreingang RF' Code 0: 'RF über Binäreingang' ist nicht zulässig		x
boFL	BOOL	Bit zum Fehlerlöschen		x
boEmergency_ Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden. Relevante Variablen: NOT-HALT-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms] NOT-HALT-Rampe: ST_AXIS_DRIVE.ST_IDS.ST_ID32919_PlcList. diID32919_05_diEmergency_StopRamp Einheit ms Standardwert: 100 ms		x

Name	Typ	Beschreibung	Sync.	Async.
boSpeed	BOOL	Drehzahlregelung Freigabesignal: Mit einer positiven Flanke startet die Funktion Drehzahlregelung. Solange 'boSpeed' = TRUE ist, wird die Drehzahlregelung (Drehzahlsollwert) ausgeführt. Mit einer negativen Flanke 'boSpeed' = FALSE wird die laufende Drehzahlregelung (mit Sollwert = 0) abgebrochen. Hoch- / Tieflauframpen ID32780 'Hochlaufzeit' und ID32781'Tieflaufzeit' bezieht sich auf die ID113 'Maximaldrehzahl'. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 1 und STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp muss auf 1 ms gestellt werden. Die PLC Variable STANDARD_AXIS.fbMotion.fbVelocity.diVelocityRamp ist die Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf den neuen Sollwert beschleunigt oder abgebremst wird. Voraussetzung: ID32802 'AMK-Nebenbetriebsart 2' Bit 6 = 0. (Hinweis, Bit 6 wird bei aktiver automatischer Parametrierung mit 1 überschrieben!) Relevante Parameter: ID32802 'AMK-Nebenbetriebsart 2' [Betriebsart: Drehzahlregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] [Sollwertrampe: aktiv / inaktiv (Bit 6)] ID32780 'Hochlaufzeit' ID32781'Tieflaufzeit' Relevante Variablen: Drehzahlsollwert: diSpeedVelocity Geschwindigkeitsrampe: STANDARD_ AXIS.fbMotion.fbVelocity.diVelocityRamp [Standardwert: 100 ms] Drehzahlregelung aktiv: boSpeedBusy		x
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Relevante Parameter: ID41 'Referenzfahr-Geschwindigkeit' ID147 'Referenzfahr-Parameter' ID150 'Referenzmaß Offset 1' ID32926 'AMK-Referenzfahr-Parameter' Relevante Variablen: Referenzfahrt aktiv: boHomeBusy Referenzfahrt beendet: boDone Referenzpunkt bekannt: stAxisDrive.stStatus.boID33036_RFP_known		x

Name	Typ	Beschreibung	Sync.	Async.
boPosAbs	BOOL	Absolute Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion absolute Positionierung. Solange 'boPosAbs' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosAbs' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Voraussetzung: Referenzpunkt muss bekannt sein, boID33036_RFP_known = TRUE [Relevante Parameter: Siehe 'boPosRel']		x
boPosRel	BOOL	Relative Positionierung Freigabesignal: Mit einer positiven Flanke startet die Funktion relative Positionierung. Solange 'boPosRel' = TRUE ist, wird die Positionierung ausgeführt. Mit einer negativen Flanke 'boPosRel' = FALSE wird die laufende Positionierung abgebrochen oder die abgeschlossene Positionierung beendet. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Positioniergeschwindigkeit: udPosVelocity Zielposition: diPosition Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Positionierung aktiv: boPosBusy Positionierung beendet: boDone		x
boJogPlus	BOOL	Tippen positive Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (positive Richtung). Solange 'boJogPlus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogPlus' = FALSE wird der laufende Tippbetrieb abgebrochen. [Relevante Parameter: Siehe 'boJogMinus']		x

Name	Typ	Beschreibung	Sync.	Async.
boJogMinus	BOOL	Tippen negative Richtung Freigabesignal: Mit einer positiven Flanke startet die Funktion Tippen (negative Richtung). Solange 'boJogMinus' = TRUE ist, wird der Tippbetrieb ausgeführt. Mit einer negativen Flanke 'boJogMinus' = FALSE wird der laufende Tippbetrieb abgebrochen. Relevante Parameter: ID32801 'AMK-Nebenbetriebsart 1' [Betriebsart: Lageregelung] [Sollwertquelle: diMainSetpoint (Code 41h)] Relevante Variablen: Tippgeschwindigkeit: udVelocityJog Beschleunigung: lreAccel Verzögerung: lreDecel Ruckbegrenzung (Start): lreJerkStart Ruckbegrenzung (Ende): lreJerkEnd Tippbetrieb aktiv: boJogBusy		x
udPosVelocity	UDINT	Positioniergeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diPosition	DINT	Zielposition Über die Eingänge 'boPosAbs' und 'boPosRel' wird bestimmt ob die Zielposition absolut oder relativ angefahren wird. Einheit: Inkremente Standardwert: 30720		x
lreAccel	LREAL	Beschleunigung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreAccel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreDecel	LREAL	Verzögerung für Positionierung und Tippbetrieb Bereich: $1{,}43\ 10^{-13} < \lvert\text{'lreDecel'}\rvert < 1{,}43\ 10^{+16}$ Einheit: $Inkr/s^2$ Standardwert: 1000000		x
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: $Inkr/s^3$ Standardwert: 10000000		x
udVelocityJog	UDINT	Tippgeschwindigkeit Einheit: Inkr/s Standardwert: 34133		x
diSpeedVelocity	DINT	Drehzahlsollwert für Drehzahlregelung Einheit: U/min Standardwert: 100		x

Name	Typ	Beschreibung	Sync.	Async.
boEnableWinder	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Wicklerbausteins. Solange 'boEnableWinder' = TRUE ist, bleibt der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnableWinder' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.		x
boDmCalc	BOOL	Aktivierung der Durchmesserberechnung		x
boFindInertia	BOOL	Ermittlung des Trägheitsmoments der Rolle und Bahnstraffung ein		x
boWind	BOOL	Wickeln ein		x
boSync	BOOL	Synchronisieren ein		x
boDmOkPreset	BOOL	Durchmesser auf Wert der Eingangsvariable 'iDmPreset' setzen. Das Eingangsbit 'boDmPreset' wird nach setzen automatisch rückgesetzt. Durchmesserstatus "Bekannt": Der Durchmesserwert wird als gültiger Durchmesser gewertet, das bedeutet, Beschleunigungs- und Reibmomente werden berechnet.		x
boDmNOkPreset	BOOL	Durchmesser auf Wert der Eingangsvariable 'iDmPreset' setzen. Das Eingangsbit 'boDmPreset' wird nach setzen automatisch rückgesetzt. Durchmesserstatus "Unbekannt": Der Durchmesserwert wird als ungültiger Durchmesser gewertet, das bedeutet, Beschleunigungs- und Reibmomente werden berechnet.		x
diWebTension	BOOL	Absolutes Bahnspannungsmoment [0,1 Nm]		x
iDmPreset	INT	Durchmesservorgabewert Wert auf den der Durchmesser mit 'boDmNokPreset' = TRUE oder 'boDmOkPreset' = TRUE gesetzt wird. [mm]		x
diInVal	DINT	Eingangsinkremente vom Bahntransport z. B. • Lagesollwert Vorzug (empfohlene Variante) • Externer Geber (Läuferrad), der auf der Bahn mitläuft • Lageistwert Vorzugsantrieb	x	

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th><th>Sync.</th><th>Async.</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td><td></td><td>x</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben		x
boDone	BOOL	Positionierung abgeschlossen. Gültig für 'boHome', 'boPosAbs' und 'boPosRel'		x
boSBM	BOOL	Systembereitmeldung (SBM)		x
boQRF	BOOL	Quittierung "Reglerfreigabe"		x
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler		x
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung Busfehler wenn 'boErr' = TRUE: EtherCAT oder CAN nicht in Operation Mode.	x	
enErrFbName	ENUM	CoDeSys V2 EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. CODESYS V3 gibt in der Visualisierung nur die Bausteinnummer aus. 'strErrName' gibt den Bausteinnamen aus.		x
strErrName	STRING	Bausteinname des Fehler verursachenden Bausteins (Information kann in der CODESYS V3 Visualisierung genutzt werden)		x
strFbInstanz	STRING	Gibt im Fehlerfall den Instanznamen des Bausteins an, welcher den Fehler verursacht hat.		x
boPosBusy	BOOL	Positionierung aktiv		x
boJogBusy	BOOL	Tippbetrieb aktiv		x
boHomeBusy	BOOL	Referenzierung aktiv		x
boSpeedBusy	BOOL	Drehzahlregelung aktiv		x
boEnabAcknWinder	BOOL	Quittierung: Wickler ist initialisiert und freigegeben		x
boDmOk	BOOL	Durchmesser bekannt		x
boFindInertiaDone	BOOL	Wickelmoment ermittelt, Bahn gestrafft		x
boWindRuns	BOOL	Wickeln läuft		x
boSyncRuns	BOOL	Synchronisation läuft		x
boDmLimit	BOOL	Durchmesser wurde berechnet 'ST_WINDER_TORQUE_CONTROL_CONFIG.iDmMin' und 'ST_WINDER_TORQUE_CONTROL_CONFIG.iDmMax', der Wert wurde begrenzt		x
iActDm	INT	Aktueller Durchmesser [mm]		x

Name	Typ	Beschreibung	Sync.	Async.
diActValWebSpeed	DINT	Istwert Bahngeschwindigkeit [mm/s]		x

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung	Sync.	Async.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)		x
stAxisDrive	STRUCT	ST_AXIS_DRIVE Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.		x
stWinderConfig	STRUCT	ST_WINDER_TORQUE_CONTROL_CONFIG Konfigurationsstruktur des Wicklers (Siehe Dokument Softwarebeschreibung AFL - AMK Funktionsbibliothek Teil 2, Teile-Nr. 203694)		x

Verwendungshinweis im CODESYS-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
STANDARD_WINDER_TORQUE_AXIS	STANDARD_WINDER_TORQUE_AXIS.actSync

Hinweise Inbetriebnahme

Durch die in der Konfigurationsstruktur (ST_WINDER_TORQUE_CONTROL_CONFIG (ST)) enthaltenen Parameter werden die Einflussfaktoren (z. B. Reibung und Masseträgheit) in die Berechnung des Motormoments einbezogen. Je genauer die Parameter vom Anwender vorgegeben werden, umso kleiner wird die Toleranz bei der Bahnspannung ausfallen.

Die Konfigurationsstruktur des Wicklers muss an die Anwendung angepasst werden.
ST_WINDER_TORQUE_CONTROL_CONFIG
(Siehe DokumentSoftwarebeschreibungAFL - AMK Funktionsbibliothek Teil 2, Teile-Nr. 203694)

Rückkopplungsgefahr

Bevor der Wickler aktiviert wird (boWind = True) muss der Vorzug auf der aktuellen Position gehalten werden. Die Bahn darf durch den Zug des Wicklers nicht den Vorzug bewegen bzw. zwischen dem Vorzug und der Andruckwalze durchrutschen.

Der Vorzug kann durch folgende Maßnahmen auf der aktuellen Position gehalten werden:

- Motor im Stillstand bestromen (QRF)
- Motorhaltebremse (optional) schließen

Bei Nichtbeachtung besteht die Gefahr einer Rückkopplung.

Der Wicklelantrieb wickelt die Bahn auf und der Istwertgeber gibt eine Bahngeschwindigkeit aus. Die Bahngeschwindigkeit geht in die Berechnung des Sollwerts für den Wickler ein. Der Wickler beschleunigt.

Folgende Konfigurationsvariablen müssen zwingend parametriert werden, damit der Funktionsbaustein fehlerfrei aktiviert werden kann:
Bahnbreite [mm] ST_WINDER_TORQUE_CONTROL_CONFIG.iWidth
Inkremente pro Motorumdrehung (ID116'Auflösung Motorgeber') ST_WINDER_TORQUE_CONTROL_CONFIG.diMotIncPerRev

Folgende Konfigurationsvariablen müssen zwingend parametriert werden, damit der Funktionsbaustein fehlerfrei aktiviert werden kann:
Anzahl der gezählten Inkremente am Eingang 'diInVal' bei zurücklegen der Strecke 'diWebEncLen' [Inkremente] ST_WINDER_TORQUE_CONTROL_CONFIG.diWebEncIncr

4.2 AMK AFL Applikationsbausteine

4.2.1 Bibliotheksbeschreibung AmkAfl.lib

4.2.1.1 05_StandardLevel

4.2.1.1.1 01_Comparison Functions

4.2.1.1.1.1 COMPARE_2VALUES (FB)

Der Funktionsbaustein 'COMPARE_2VALUES' setzt ein Ausgangsbit, wenn die Bedingung Wert1 = Wert2 innerhalb eines Fensters erfüllt ist.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	COMPARE_2VALUES			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Fenstergröße -	diWindow	boEquality	-	Gleichheit
Vergleichswert 1 -	diVal1			
Vergleichswert 2 -	diVal2			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diWindow	DINT	Parameter zur Eingabe der Fenstergröße [Gleiche Einheit wie die Vergleichswerte]
diVal1	DINT	Vergleichswert 1 bzw. 2 [Gleiche Einheit wie der Vergleichswert]
diVal2	DINT	Vergleichswert 1 bzw. 2 [Gleiche Einheit wie der Vergleichswert]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boEquality	BOOL	Vergleichswert 1 = Vergleichswert 2 innerhalb des angegebenen Fensters

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.1.2 fboCompare_ActVal_GT_Threshold (FB)

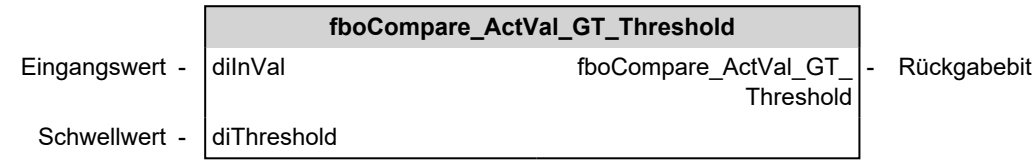
Der Funktionsbaustein 'fboCompare_ActVal_GT_Threshold' vergleicht den Eingangswert mit dem Schwellwert.

Der Rückgabewert ist TRUE, wenn der Eingangswert > Schwellwert, bei Schwellwert >0.

Der Rückgabewert ist TRUE, wenn der Eingangswert < Schwellwert, bei Schwellwert <0.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
diInVal	DINT	Eingangswert
diThreshold	DINT	Schwellwert

Ausgangsvariablen

Name	Typ	Beschreibung
fboCompare_ActVal_GT_Threshold	BOOL	Rückgabebit

4.2.1.1.2 03_Position

4.2.1.1.2.1 Support

POSITION_1_A4 (FB)

Der Funktionsbaustein 'POSITION_1_A4' ist ein Grundbaustein zur relativen Positionierung. Er positioniert eine AMK-Achse relativ bzw. relativ und nachtriggerbar.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

'Support'-Funktionen und -Funktionsbausteine werden nicht direkt angewendet, sondern von anderen Funktionen und Funktionsbausteinen aufgerufen.

Anwender Interface

		POSITION_1_A4			
Auswahlmodus	-	enMode	boDone	-	Quittierung "Ausgeführt"
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Ausführungssignal	-	boStart	bo0Vel	-	Keine Sollwertausgabe
Stoppsignal	-	boStop	boSetVel	-	Sollgeschwindigkeit erreicht
Not-Halt	-	boNotHalt	boSetPos	-	Position erreicht (Nachtrigger)
Zielposition	-	diPosition	boErr	-	Fehlersignal
Beschleunigung	-	lreAccel	iErrId	-	Fehler Identnummer
Verzögerung	-	lreDecel	diOutOffs	-	Offset
Not-Halt Rampe	-	lreDecelNotHalt	lreActVelocity	-	Aktuelle Geschwindigkeit
Geschwindigkeit	-	lreVelocity	lreActAccel	-	Aktuelle Beschleunigung
Ruck beim Start	-	lreJerkStart	diOutVal	-	Ausgangswert
Ruck beim Stop	-	lreJerkEnd			
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
enMode	ENUM	EN_POS_J_MODE Auswahlmodus POS_REL: relative Positionierung POS_REL_RETRIG_EXT: relative Positionierung mit Nachtrigger-Modus und erweiterter Funktionalität
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boStart	BOOL	Mit einer positiven Flanke startet die Ausführung des Bausteins.
boStop	BOOL	Stopp aller Bewegungen (Übergang in Drehzahlregelung mit Solldrehzahl = 0).
boNotHalt	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diPosition	DINT	Modulo Position
lreAccel	LREAL	Beschleunigung, mit der auf die Sollgeschwindigkeit gefahren wird
lreDecel	LREAL	Verzögerung, mit der eine niedrigere Sollgeschwindigkeit angefahren wird
lreDecelNotHalt	LREAL	NOT-HALT-Rampe: Verzögerung, in der der Antrieb beim NOT-HALT von der aktuellen Geschwindigkeit auf Null abgebremst wird. Einheit: Inkr/s^2
lreVelocity	LREAL	Geschwindigkeit, mit der die Positionierung ausgeführt wird. [Inkremente/s]
lreJerkStart	LREAL	Ruckbegrenzung (Start) für Positionierung und Tippbetrieb Einheit: Inkr/s^3 Standardwert: 10000000

Name	Typ	Beschreibung
lreJerkEnd	LREAL	Ruckbegrenzung (Ende) für Positionierung und Tippbetrieb Einheit: Inkr/s^3 Standardwert: 10000000

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
bo0Vel	BOOL	Wenn 'bo0Vel' = TRUE ist, wird kein Sollwert mehr ausgegeben.
boSetVel	BOOL	Wenn 'boSetVel' = TRUE ist, ist die Sollgeschwindigkeit erreicht.
boSetPos	BOOL	Letzte Sollposition erreicht Das Signal ist nur einen Zyklus lang aktiv, 'boSetPos' = TRUE, und kennzeichnet das Ende der ursprünglichen Positionierung, bzw. den Beginn des Retrigger-Vorgangs
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrId	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
diOutOffs	DINT	Offsetwert, am Ende der ursprünglichen Positionierung von der tatsächlich vorgegebenen Sollposition; gültig, wenn 'boSetPos'=TRUE
lreActVelocity	LREAL	Aktuelle Geschwindigkeit [Inkremente/s]
lreActAccel	LREAL	Aktueller Beschleunigungswert [Inkremente/s^2]
diOutVal	DINT	Ausgangswert [Inkremente]

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	POSITION_1_A4

POSITION_JERK_1_A5 (FB)

Der Funktionsbaustein 'POSITION_JERK_1_A5' ist ein Grundbaustein zur relativen Positionierung mit Ruckbegrenzung. Er positioniert eine AMK-Achse relativ bzw. relativ und nachtriggerbar.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

'Support'-Funktionen und -Funktionsbausteine werden nicht direkt angewendet, sondern von anderen Funktionen und Funktionsbausteinen aufgerufen.

Anwender Interface

		POSITION_JERK_1_A5			
Auswahlmodus	-	enMode	boDone	-	Quittierung "Ausgeführt"
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Ausführungssignal	-	boStart	bo0Vel	-	Keine Sollwertausgabe
Stoppsignal	-	boStop	boSetVel	-	Sollgeschwindigkeit erreicht
Not-Halt	-	boNotHalt	boSetPos	-	Position erreicht (Nachtrigger)
Zielposition	-	diPosition	boErr	-	Fehlersignal
Positionierbeschleunigung	-	lreAccel	iErrID	-	Fehler Identnummer
Positionierverzögerung	-	lreDecel	diOutOffs	-	Offset
Not-Halt Rampe	-	lreDecelNotHalt	lreActVelocity	-	Geschwindigkeitsistwert
Positiongeschwindigkeit	-	lreVelocity	lreActAccel	-	Aktuelle Beschleunigung
Ruck beim Start	-	lreJerkStart	diOutVal		Ausgangswert
Ruck beim Stop	-	lreJerkEnd			
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
enMode	ENUM	EN_POS_J_MODE Auswahlmodus POS_REL: Relative Positionierung POS_REL_RETRIG_EXT: Relative Positionierung nachtriggerbar, erweiterte Funktionalität POS_J_REL_RETRIG: Relative Positionierung nachtriggerbar mit Ruckbegrenzung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boStart	BOOL	Mit einer positiven Flanke startet die Ausführung des Bausteins.
boStop	BOOL	Mit einer positiven Flanke wird die Ausführung des Bausteins abgebrochen oder beendet.
boNotHalt	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diPosition	DINT	Ziel der Positionierung [Inkremente]
lreAccel / lreDecel	LREAL	Beschleunigungs- / Bremsrampe [Inkremente/s^2]
lreDecelNotHalt	LREAL	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
lreVelocity	LREAL	Geschwindigkeit, mit der die Positionierung ausgeführt wird. [Inkremente/s]
lreJerkStart / lreJerkEnd	LREAL	Ruck bei der Beschleunigung / Verzögerung [Inkremente/s^3]

Ausgangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben

Name	Typ	Beschreibung
bo0Vel	BOOL	Wenn 'bo0Vel' = TRUE ist, wird kein Sollwert mehr ausgegeben.
boSetVel	BOOL	Wenn 'boSetVel' = TRUE ist, ist die Sollgeschwindigkeit erreicht.
boSetPos	BOOL	Letzte Sollposition erreicht Das Signal ist nur einen Zyklus lang aktiv, 'boSetPos' = TRUE, und kennzeichnet das Ende der ursprünglichen Positionierung, bzw. den Beginn des Retrigger-Vorgangs
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung 1: Eingabeparameter sind nicht sinnvoll 2: Positioniergeschwindigkeit wurde automatisch angepasst 3: Endgeschwindigkeit wurde automatisch angepasst 4: Ruckparameter wurde automatisch angepasst
diOutOffs	DINT	Offsetwert, am Ende der ursprünglichen Positionierung von der tatsächlich vorgegebenen Sollposition; gültig, wenn 'boSetPos'=TRUE
lreActVelocity	LREAL	Aktuelle Geschwindigkeit [Inkremente/s] (Vorsteuerwert)
lreActAccel	LREAL	Aktueller Beschleunigungswert [Inkremente/s^2] (Vorsteuerwert)
diOutVal	DINT	Ausgabeposition

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	POSITION_JERK_1_A5

4.2.1.1.2.2 POSITION_ABSOLUT (FB)

Der Funktionsbaustein 'POSITION_ABSOLUT' positioniert eine AMK-Achse absolut.
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		POSITION_ABSOLUT			
Funktionsausführung	-	boExec	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boBusy	-	Ausführung läuft
Not-Halt Rampe	-	diEmergency_StopRamp	boInPos	-	In Position
Zielposition	-	diPosition	boErr	-	Fehlersignal
Positioniergeschwindigkeit	-	lreVelocity	iErrID	-	Fehler Identnummer
Positionierbeschleunigung	-	lreAccel	enErrName	-	Name fehlerh Baustein
Positionsfenster	-	diPosWindow			
Positionsistwert	-	diActPosition			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

1) Gilt nur für CODESYS V3

Eingangsvariablen

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>boExec</td><td>BOOL</td><td>Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins.
Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet.
Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.</td></tr>
<tr><td>boEmergency_Stop</td><td>BOOL</td><td>Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.</td></tr>
<tr><td>diEmergency_StopRamp</td><td>DINT</td><td>Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]</td></tr>
<tr><td>diPosition</td><td>DINT</td><td>Ziel der Positionierung [Inkremente]</td></tr>
<tr><td>lreAccel</td><td>LREAL</td><td>Beschleunigungs- / Bremsrampe [Inkremente/s^2]</td></tr>
<tr><td>lreVelocity</td><td>LREAL</td><td>Geschwindigkeit, mit der die Positionierung ausgeführt wird. [Inkremente/s]</td></tr>
<tr><td>diPosWindow</td><td>DINT</td><td>Positionsfenster: Fenster, innerhalb dessen das InPos–Bit gesetzt wird. [Inkremente]
Standardwert: 1000 Inkremente</td></tr>
<tr><td>diActPosition</td><td>DINT</td><td>Aktueller Lageistwert [Inkremente]</td></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:
<table>
<tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr>
<tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr>
<tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr>
</table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.
<table>
<tr><td>iErrID</td><td>5</td></tr>
</table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boInPos	BOOL	Meldung „InPosition“: Aktuelle Position = Zielposition innerhalb des Positionsfensters
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: POS_1 – Bibliothek: AmkBase.lib
strFbInstance	STRING	Aufrufpfad des Bausteins

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	POSITION_ABSOLUT

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.2.3 POSITION_ABSOLUT_VAJ (FB)

Der Funktionsbaustein 'POSITION_ABSOLUT_VAJ' positioniert eine AMK-Achse absolut. Dabei kann der Anwender neben Position, Geschwindigkeit und Beschleunigung auch den Ruck vorgeben.

$$v = \frac{\Delta s}{\Delta t}$$ Geschwindigkeit

$$a = \frac{\Delta v}{\Delta t} = \frac{\Delta s}{\Delta t^2}$$ Beschleunigung

$$j = \frac{\Delta a}{\Delta t} = \frac{\Delta v}{\Delta t^2} = \frac{\Delta s}{\Delta t^3}$$ Ruck

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

In der AMKAMAC A4 Steuerung wird die Ruckbegrenzung nicht unterstützt.

Anwender Interface

		POSITION_ABSOLUT_VAJ			
Funktionsausführung	-	boExec	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boBusy	-	Ausführung läuft
Not-Halt Rampe	-	diEmergency_StopRamp	boInPos	-	In Position
Zielposition	-	diPosition	lreActVelocity	-	Positioniergeschwindigkeit
Positionierbeschleunigung	-	lreAccel	lreActAccel	-	Positionierbeschleunigung
Positionierverzögerung	-	lreDecel	boErr	-	Fehlersignal
Positioniergeschwindigkeit	-	lreVelocity	iErrID	-	Fehler Identnummer
Ruck beim Start	-	lreJerkStart	enErrName	-	Name fehlerh Baustein
Ruck beim Stop	-	lreJerkEnd			
Positionsfenster	-	diPosWindow			
Positionsistwert	-	diActPosition			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diPosition	DINT	Ziel der Positionierung [Inkremente]
lreAccel / lreDecel	LREAL	Beschleunigungs- / Bremsrampe [Inkremente/s^2]
lreVelocity	LREAL	Geschwindigkeit, mit der die Positionierung ausgeführt wird. [Inkremente/s]
lreJerkStart / lreJerkEnd	LREAL	Ruck bei der Beschleunigung / Verzögerung [Inkremente/s^3]

Name	Typ	Beschreibung
diPosWindow	DINT	Positionsfenster: Fenster, innerhalb dessen das InPos–Bit gesetzt wird. [Inkremente] Standardwert: 1000 Inkremente
diActPosition	DINT	Aktueller Lageistwert [Inkremente]
enArithmetik[1)]	ENUM	EN_POS_J_ARITHMETIK (Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar, bei CODESYS V2 wird die Auswahl immer automatisch angepasst) Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt. Auswahlmöglichkeiten:

enAutomaticSelection	Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung) Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.
en64BitArithmetik	64 Bit Arithmetik
en32BitArithmetik	32 Bit Arithmetik

Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.

iErrID	5

Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boInPos	BOOL	Meldung „InPosition“: Aktuelle Position = Zielposition innerhalb des Positionsfensters
lreActVelocity	LREAL	Geschwindigkeit, mit der die Positionierung ausgeführt wird. [Inkremente/s]
lreActAccel	LREAL	Aktueller Beschleunigungswert [Inkremente/s^2]
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler

FALSE	Kein Fehler (Kommandierung zulässig oder Warnung)
TRUE	Fehler

iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben

iErrID = 0		Kein Fehler
iErrID ≠ 0	boErr = TRUE	Fehler
iErrID ≠ 0	boErr = FALSE	Warnung

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein** / **Bibliothek** POSITION_JERK_1 / AmkAfl.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	POSITION_ABSOLUT_VAJ

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.2.4 POSITION_CHECK_LIMIT_SWITCH (FB)

Der Funktionsbaustein 'POSITION_CHECK_LIMIT_SWITCH' prüft eine Zielposition auf Endlagengrenzen.
Liegt die Zielposition innerhalb der angegebenen Grenzen, wird dies durch das Ausgangsbit 'boLimitOk' angezeigt. Diese Meldung kann z.B. dazu benutzt werden, eine Absolutpositionierung zu starten. Die entsprechende Zielposition steht am Ausgang 'diTargetPosition' zur Verfügung.
Durch das Setzten des Eingangsbits 'boAdaptTargetPosition' wird eine Anpassung der Zielposition auf die Grenzen vorgenommen. D.h. liegt das Ziel außerhalb der angegebenen Grenzen, wird die ausgegebene Zielposition am Ausgang 'diTargetPosition' auf die entsprechende Grenze angepasst. Ist das Eingangsbit 'boAdaptTargetPosition' nicht gesetzt, wird nur die Endlagenprüfung durchgeführt und die Zielposition durchgeschleift.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	POSITION_CHECK_LIMIT_SWITCH			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Anpassung Zielposition -	boAdaptTargetPosition	boLimitOk	-	Zielpos in Grenzen
Endlagengrenze -	diLimit1	diTargetPosition	-	Ausgabe Zielpos
Endlagengrenze -	diLimit2			
Ziel Positionierung -	diTarget			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boAdaptTargetPosition	BOOL	Anpassung der Zielposition entsprechend der Endlagen bei Endlagenüberschreitung
diLimit1	DINT	Endlage 1: kleinste Position [Inkremente]
diLimit2	DINT	Endlage 2: größte Position [Inkremente]
diTarget	DINT	Auf Endlagen zu prüfendes Ziel für die Positionierung

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boLimitOk	BOOL	Die Ausgabe-Zielposition liegt in den Endlagegrenzen. Dieses Bit kann zum Auslösen der Positionierung verwendet werden.
diTargetPosition	DINT	Ausgabe der geprüften Zielposition. diese Ausgabe kann als Zielposition für die Positionierung verwendet werden.

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.2.5 POSITION_HOMING_FIXED_STOP (FB)

Der Funktionsbaustein 'POSITION_HOMING_FIXED_STOP' erkennt einen Festanschlag.

Die Bewegung zum Festanschlag (Drehzahlregelung) muss außerhalb des Funktionsbausteins realisiert werden, z. B. mit 'STANDARD_AXIS'. Zum Ansteuern der Drehzahlregelung kann der Ausgang 'boHomeBusy' benutzt werden.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	POSITION_HOMING_FIXED_STOP			
Funktionsausführung -	boExec	boDone	-	Quittierung "Ausgeführt"
Rücksetzen -	boReset	boRpfKnown	-	Referenzpunkt bekannt
Wartezeit -	tWait	boHomeBusy	-	Ausführung läuft
Quittierung "Reglerfreigabe" -	boQRF	boSetHomePos	-	Festanschlag erreicht
Stillstand -	boZeroVelocityWindow			
Drehmomentschwelle -	boTorqueLimit			

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet. Mit der positiven Flanke an diesem Eingang wird der Ausgang 'boHomeBusy' gesetzt.
boReset	BOOL	Löscht die Ausgangsmeldung 'boRpfKnown' (Referenzpunkt bekannt)
tWait	TIME	Verweilzeit am Festanschlag, bis die Ausgänge 'boRpfKnown' und 'boSetHomePos' gesetzt werden. [ms] Standardwert: 500 ms
boQRF	BOOL	Quittierung Reglerfreigabe (QRF)
boTorqueLimit	BOOL	Drehmomentschwelle erreicht (ID126 'Drehmomentschwelle', Code 333 Md ≥ Mdx)
boZeroVelocityWindow	BOOL	Drehzahl im Stillstandsfenster (ID124 'Stillstands-Fenster', Code 331 nact < nmin)

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boRpfKnown	BOOL	Referenzpunkt bekannt Wird der Festanschlag erreicht (ID333 'Meldung Drehmoment: Istwert ≥ Schwelle' und ID331 'Meldung Drehzahl: Istwert < Minimum' liegen für die an 'tWait' angegebene Zeit an), wird 'boRpfKnown' gesetzt und bleibt solange gesetzt, bis er durch einen positive Flanke an 'boReset' gelöscht wird.
boHomeBusy	BOOL	Bearbeitung ist aktiv Dieser Ausgang kann benutzt werden, um den Antrieb in die Betriebsart Drehzahlregelung umzuschalten. Z. B. boHomeBusy -> STANDARD_AXIS.boSpeed
boSetHomePos	BOOL	Festanschlag erreicht -> Referenzposition setzen Wird der Festanschlag erreicht (ID333 'Meldung Drehmoment: Istwert ≥ Schwelle' und ID331 'Meldung Drehzahl: Istwert < Minimum' liegen für die an 'tWait' angegebene Zeit an), wird der Ausgang 'boSetHomePos' für einen Zyklus gesetzt.

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	POSITION_HOMING_FIXED_STOP

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.2.6 POSITION_JERK_1 (FB)

Der Funktionsbaustein 'POSITION_JERK_1' ist ein Grundbaustein zur relativen Positionierung mit Ruckbegrenzung. Er positioniert eine AMK-Achse relativ bzw. relativ und nachtriggerbar.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	POSITION_JERK_1			
Auswahlmodus -	enMode	boDone	-	Quittierung "Ausgeführt"
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Ausführungssignal -	boStart	bo0Vel	-	Keine Sollwertausgabe
Stoppsignal -	boStop	boSetVel	-	Sollgeschwindigkeit erreicht
Not-Halt -	boNotHalt	boSetPos	-	Position erreicht (Nachtrigger)
Zielposition -	diPosition	boErr	-	Fehlersignal
Positionierbeschleunigung -	lreAccel	iErrID	-	Fehler Identnummer
Positionierverzögerung -	lreDecel	diOutOffs	-	Offset
Not-Halt Rampe -	lreDecelNotHalt	lreActVelocity	-	Geschwindigkeitsistwert
Positiongeschwindigkeit -	lreVelocity	lreActAccel	-	Aktuelle Beschleunigung
Ruck beim Start -	lreJerkStart	diOutVal	-	Ausgangswert
Ruck beim Stop -	lreJerkEnd			
32Bit oder 64Bit Arithmetik -	enArithmetik[1)]			
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung
enMode	ENUM	EN_POS_J_MODE Auswahlmodus POS_J_REL: Relative Positionierung POS_J_REL_RETRIG_EXT: Relative Positionierung nachtriggerbar
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boStart	BOOL	Mit einer positiven Flanke startet die Ausführung des Bausteins.
boStop	BOOL	Mit einer positiven Flanke wird die Ausführung des Bausteins abgebrochen oder beendet.
boNotHalt	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diPosition	DINT	Ziel der Positionierung [Inkremente]
lreAccel / lreDecel	LREAL	Beschleunigungs- / Bremsrampe [Inkremente/s^2]
lreDecelNotHalt	LREAL	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
lreVelocity	LREAL	Geschwindigkeit, mit der die Positionierung ausgeführt wird. [Inkremente/s]
lreJerkStart / lreJerkEnd	LREAL	Ruck bei der Beschleunigung / Verzögerung [Inkremente/s^3]

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>boDone</td><td>BOOL</td><td>Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde</td></tr>
<tr><td>boEnabAck</td><td>BOOL</td><td>Quittierung: Funktionsbaustein ist initialisiert und freigegeben</td></tr>
<tr><td>bo0Vel</td><td>BOOL</td><td>Wenn 'bo0Vel' = TRUE ist, wird kein Sollwert mehr ausgegeben.</td></tr>
<tr><td>boSetVel</td><td>BOOL</td><td>Wenn 'boSetVel' = TRUE ist, ist die Sollgeschwindigkeit erreicht.</td></tr>
<tr><td>boSetPos</td><td>BOOL</td><td>Letzte Sollposition erreicht
Das Signal ist nur einen Zyklus lang aktiv, 'boSetPos' = TRUE, und kennzeichnet das Ende der ursprünglichen Positionierung, bzw. den Beginn des Retrigger-Vorgangs</td></tr>
<tr><td>boErr</td><td>BOOL</td><td>Der Funktionsbaustein befindet sich im Zustand Fehler<table><tr><td>FALSE</td><td>Kein Fehler (Kommandierung zulässig oder Warnung)</td></tr><tr><td>TRUE</td><td>Fehler</td></tr></table></td></tr>
</table>

Name	Typ	Beschreibung
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0 — Kein Fehler iErrID ≠ 0 — boErr = TRUE — Fehler iErrID ≠ 0 — boErr = FALSE — Warnung 1 — Eingabeparameter sind nicht sinnvoll 2 — Positioniergeschwindigkeit wurde automatisch angepasst 3 — Endgeschwindigkeit wurde automatisch angepasst 4 — Ruckparameter wurde automatisch angepasst 5 — Steuerungstyp konnte nicht ermittelt werden. Bitte 'ST_DEVICE' mit der Steuerung im AIPEX PRO verbinden und Autokonfiguration starten.
diOutOffs	DINT	Offsetwert, am Ende der ursprünglichen Positionierung von der tatsächlich vorgegebenen Sollposition; gültig, wenn 'boSetPos'=TRUE
lreActVelocity	LREAL	Aktuelle Geschwindigkeit [Inkremente/s] (Vorsteuerwert)
lreActAccel	LREAL	Aktueller Beschleunigungswert [Inkremente/s^2] (Vorsteuerwert)
diOutVal	DINT	Ausgabeposition

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	POSITION_JERK_1

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.2.7 POSITION_RELATIV (FB)

Der Funktionsbaustein 'POSITION_RELATIV' positioniert eine AMK-Achse relativ.
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		POSITION_RELATIV			
Funktionsausführung	-	boExec	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boBusy	-	Ausführung läuft
Not-Halt Rampe	-	diEmergency_StopRamp	boInPos	-	In Position
Zielposition	-	diPosition	boErr	-	Fehlersignal
Positioniergeschwindigkeit	-	lreVelocity	iErrID	-	Fehler Identnummer
Positionierbeschleunigung	-	lreAccel	enErrName	-	Name fehlerh Baustein
Positionsfenster	-	diPosWindow			
Positionsistwert	-	diActPosition			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

1) Gilt nur für CODESYS V3

Eingangsvariablen

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>boExec</td><td>BOOL</td><td>Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins.
Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet.
Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.</td></tr>
<tr><td>boEmergency_Stop</td><td>BOOL</td><td>Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.</td></tr>
<tr><td>diEmergency_StopRamp</td><td>DINT</td><td>Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]</td></tr>
<tr><td>diPosition</td><td>DINT</td><td>Ziel der Positionierung [Inkremente]</td></tr>
<tr><td>lreAccel</td><td>LREAL</td><td>Beschleunigungs- / Bremsrampe [Inkremente/s^2]</td></tr>
<tr><td>lreVelocity</td><td>LREAL</td><td>Geschwindigkeit, mit der die Positionierung ausgeführt wird. [Inkremente/s]</td></tr>
<tr><td>diPosWindow</td><td>DINT</td><td>Positionsfenster: Fenster, innerhalb dessen das InPos–Bit gesetzt wird. [Inkremente]
Standardwert: 1000 Inkremente</td></tr>
<tr><td>diActPosition</td><td>DINT</td><td>Aktueller Lageistwert [Inkremente]</td></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:
<table>
<tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr>
<tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr>
<tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr>
</table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.
<table>
<tr><td>iErrID</td><td>5</td></tr>
</table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boInPos	BOOL	Meldung „InPosition“: Aktuelle Position = Zielposition innerhalb des Positionsfensters
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: POS_1 – Bibliothek: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	POSITION_RELATIV

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

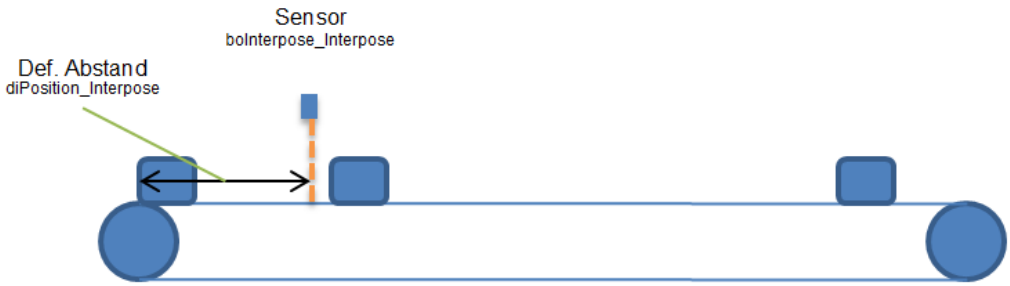
4.2.1.1.2.8 POSITION_RELATIV_INTERPOSED (FB)

Der Funktionsbaustein 'POSITION_RELATIV_INTERPOSED' bewegt nach einer positiven Flanke an 'boExec' die Achse kontinuierlich, bis durch eine positive Flanke an 'boInterpose' die ablösende Positionierung gestartet wird. Ist die relative Position erreicht, stoppt der Baustein.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Beispiel:

Bei einem Transportband werden unregelmäßig Teile abgelegt. Nach auslösen des Sensors, werden die Teile bei einer definierten Position gestoppt und vom Band genommen.



Anwender Interface

	POSITION_RELATIV_INTERPOSED			
Funktionsausführung -	boExec	boDone	-	Quittierung "Ausgeführt"
Positionierung ablösen -	boInterpose	boBusy	-	Ausführung läuft
Not-Halt -	boEmergency_Stop	boErr	-	Fehlersignal
Not-Halt Rampe -	diEmergency_StopRamp	iErrID	-	Fehler Identnummer
Zielposition (relativ) -	diPosition	enErrName	-	Name fehlerh Baustein
Korrektur Sollposition -	diOffset	boInPos	-	In Position
Beschleunigung -	udAccel	bo0Vel	-	Geschw. = 0
Verzögerung -	udDecel	boSetVel	-	Sollgeschwindigkeit erreicht
Geschwindigkeit -	udVelocity			
Positionsfenster -	diPosWindow			
Positionsistwert -	diActPosition			
Lagesollwert -	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
boInterpose	BOOL	Ablösende Positionierung starten
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diPosition	DINT	Relative Zielposition (nach boInterpose) Einheit: Inkr/s Standardwert: 60000

Name	Typ	Beschreibung
diOffset	DINT	Korrektur der Sollposition [Inkremente] Einheit: Inkr/s Standardwert: 0
udAccel	UDINT	Beschleunigung, mit der auf die Sollgeschwindigkeit gefahren wird
udDecel	UDINT	Verzögerung, mit der eine niedrigere Sollgeschwindigkeit angefahren wird
udVelocity	UDINT	Geschwindigkeit, mit der die Positionierung ausgeführt wird. [Inkremente/s]
diPosWindow	DINT	Fenster für die Meldung "In Position" [Inkremente]
diActPosition	DINT	Aktueller Lageistwert [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: POS_1 – Bibliothek: AmkBase.lib
boInPos	BOOL	Meldung „InPosition“: Aktuelle Position = Zielposition innerhalb des Positionsfensters
bo0Vel	BOOL	Wenn 'bo0Vel' = TRUE ist, wird kein Sollwert mehr ausgegeben.
boSetVel	BOOL	Wenn 'boSetVel' = TRUE ist, ist die Sollgeschwindigkeit erreicht.

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	POSITION_RELATIV_INTERPOSED

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.2.9 POSITION_RELATIV_RETRIGGER_VAJ (FB)

Der Funktionsbaustein 'POSITION_RELATIV_RETRIGGER_VAJ' positioniert einen AMK-Antrieb relativ und nachtriggerbar.
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

In der AMKAMAC A4 Steuerung wird die Ruckbegrenzung nicht unterstützt.

Anwender Interface

		POSITION_RELATIV_RETRIGGER_VAJ			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Ausführungssignal	-	boStart	boDone	-	Quittierung "Ausgeführt"
Stoppsignal	-	boStop	boBusy	-	Ausführung läuft
Not-Halt	-	boEmergency_Stop	boInPos	-	In Position
Not-Halt Rampe	-	diEmergency_StopRamp	lreActVelocity	-	Geschwindigkeitsistwert
Zielposition	-	diPosition	lreActAccel	-	Aktuelle Beschleunigung
Beschleunigung	-	lreAccel	bo0Vel	-	Keine Sollwertausgabe
Verzögerung	-	lreDecel	boSetVel	-	Sollgeschwindigkeit erreicht
Positioniergeschwindigkeit	-	lreVelocity	boSetPos	-	Position erreicht (Nachtrigger)
Ruck beim Start	-	lreJerkStart	diOutOffs	-	Offset
Ruck beim Stop	-	lreJerkEnd	boErr	-	Fehlersignal
Positionsfenster	-	diPosWindow	iErrID	-	Fehler Identnummer
Positionsistwert	-	diActPosition	enErrName	-	Name fehlerh Baustein
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boStart	BOOL	Mit einer positiven Flanke startet die Ausführung des Bausteins.
boStop	BOOL	Mit einer positiven Flanke wird die Ausführung des Bausteins abgebrochen oder beendet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diPosition	DINT	Ziel der Positionierung [Inkremente]
lreAccel / lreDecel	LREAL	Beschleunigungs- / Bremsrampe [Inkremente/s^2]

Name	Typ	Beschreibung
lreVelocity	LREAL	Geschwindigkeit, mit der die Positionierung ausgeführt wird. [Inkremente/s]
lreJerkStart / lreJerkEnd	LREAL	Ruck bei der Beschleunigung / Verzögerung [Inkremente/s^3]
diPosWindow	DINT	Fenster für die Meldung "In Position" [Inkremente]
diActPosition	DINT	Aktueller Lageistwert [Inkremente]
enArithmetik[1)]	ENUM	EN_POS_J_ARITHMETIK (Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar, bei CODESYS V2 wird die Auswahl immer automatisch angepasst) Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt. Auswahlmöglichkeiten: enAutomaticSelection: Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung) Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden. en64BitArithmetik: 64 Bit Arithmetik en32BitArithmetik: 32 Bit Arithmetik Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden. iErrID: 5 Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl: Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.

1) Gilt nur für CODESYS V3

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boInPos	BOOL	Meldung „InPosition“: Aktuelle Position = Zielposition innerhalb des Positionsfensters
lreActVelocity	LREAL	Aktuelle Geschwindigkeit [Inkremente/s]
lreActAccel	LREAL	Aktueller Beschleunigungswert [Inkremente/s^2]
bo0Vel	BOOL	Wenn 'bo0Vel' = TRUE ist, wird kein Sollwert mehr ausgegeben.
boSetVel	BOOL	Wenn 'boSetVel' = TRUE ist, ist die Sollgeschwindigkeit erreicht.
boSetPos	BOOL	Letzte Sollposition erreicht Das Signal ist nur einen Zyklus lang aktiv, 'boSetPos' = TRUE, und kennzeichnet das Ende der ursprünglichen Positionierung, bzw. den Beginn des Retrigger-Vorgangs

Name	Typ	Beschreibung
diOutOffs	DINT	Offsetwert, am Ende der ursprünglichen Positionierung von der tatsächlich vorgegebenen Sollposition; gültig, wenn 'boSetPos'=TRUE
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: POSITION_JERK_1 – Bibliothek: AmkAfl.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	POSITION_RELATIV_RETRIGGER_VAJ

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.2.10 POSITION_RELATIV_VAJ (FB)

Der Funktionsbaustein 'POSITION_RELATIV_VAJ' positioniert eine AMK-Achse absolut. Dabei kann der Anwender neben Position, Geschwindigkeit und Beschleunigung auch den Ruck vorgeben.

$v = \frac{\Delta s}{\Delta t}$ Geschwindigkeit

$a = \frac{\Delta v}{\Delta t} = \frac{\Delta s}{\Delta t^2}$ Beschleunigung

$j = \frac{\Delta a}{\Delta t} = \frac{\Delta v}{\Delta t^2} = \frac{\Delta s}{\Delta t^3}$ Ruck

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

In der AMKAMAC A4 Steuerung wird die Ruckbegrenzung nicht unterstützt.

Anwender Interface

		POSITION_RELATIV_VAJ			
Funktionsausführung	-	boExec	boDone	-	Quittierung "Ausgeführt"
Not-Halt	-	boEmergency_Stop	boBusy	-	Ausführung läuft
Not-Halt Rampe	-	diEmergency_StopRamp	boInPos	-	In Position
Zielposition	-	diPosition	lreActVelocity	-	Positioniergeschwindigkeit
Positionierbeschleunigung	-	lreAccel	lreActAccel	-	Positionierbeschleunigung
Positionierverzögerung	-	lreDecel	boErr	-	Fehlersignal
Positioniergeschwindigkeit	-	lreVelocity	iErrID	-	Fehler Identnummer
Ruck beim Start	-	lreJerkStart	enErrName	-	Name fehlerh Baustein
Ruck beim Stop	-	lreJerkEnd			
Positionsfenster	-	diPosWindow			
Positionsistwert	-	diActPosition			
32Bit oder 64Bit Arithmetik	-	enArithmetik[1)]			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

1) Gilt nur für CODESYS V3

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diPosition	DINT	Ziel der Positionierung [Inkremente]
lreAccel / lreDecel	LREAL	Beschleunigungs- / Bremsrampe [Inkremente/s^2]
lreVelocity	LREAL	Geschwindigkeit, mit der die Positionierung ausgeführt wird. [Inkremente/s]
lreJerkStart / lreJerkEnd	LREAL	Ruck bei der Beschleunigung / Verzögerung [Inkremente/s^3]
diPosWindow	DINT	Positionsfenster: Fenster, innerhalb dessen das InPos–Bit gesetzt wird. [Inkremente] Standardwert: 1000 Inkremente
diActPosition	DINT	Aktueller Lageistwert [Inkremente]

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>enArithmetik[1)]</td><td>ENUM</td><td>EN_POS_J_ARITHMETIK
(Dieser Eingang ist nur für CODESYS V3 Bausteine verfügbar,
bei CODESYS V2 wird die Auswahl immer automatisch angepasst)

Mit 'EN_POS_J_ARITHMETIK' wird die Auswahl 32 Bit oder 64 Bit Arithmetik für Positionierbausteine in der Betriebsart Lageregelung eingestellt.
Auswahlmöglichkeiten:
<table><tr><td>enAutomaticSelection</td><td>Automatische Bestimmung ob 32 Bit oder 64 Bit Arithmetik (Standardeinstellung)

Die Standardeinstellung 'enAutomaticSelection' kann nur genutzt werden, wenn sich die Steuerung auf dem ersten Platz in der CODESYS 'Device_Config' befindet. Ist das nicht der Fall muss die Auswahl durch den Anwender manuell getroffen werden.</td></tr><tr><td>en64BitArithmetik</td><td>64 Bit Arithmetik</td></tr><tr><td>en32BitArithmetik</td><td>32 Bit Arithmetik</td></tr></table>
Funktioniert die automatische Bestimmung nicht, wird folgende Fehlermeldung ausgegeben und die Auswahl muss manuell getroffen werden.<table><tr><td>iErrID</td><td>5</td></tr></table>
Weitere Informationen zu 'EN_POS_J_ARITHMETIK' und manuellen Auswahl:
Siehe 'EN_POS_J_ARITHMETIK' auf Seite 631.</td></tr>
</table>

1) Gilt nur für CODESYS V3

Ausgangsvariablen

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>boDone</td><td>BOOL</td><td>Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde</td></tr>
<tr><td>boBusy</td><td>BOOL</td><td>Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.</td></tr>
<tr><td>boInPos</td><td>BOOL</td><td>Meldung „InPosition“: Aktuelle Position = Zielposition innerhalb des Positionsfensters</td></tr>
<tr><td>lreActVelocity</td><td>LREAL</td><td>Geschwindigkeit, mit der die Positionierung ausgeführt wird. [Inkremente/s]</td></tr>
<tr><td>lreActAccel</td><td>LREAL</td><td>Aktueller Beschleunigungswert [Inkremente/s^2]</td></tr>
<tr><td>boErr</td><td>BOOL</td><td>Der Funktionsbaustein befindet sich im Zustand Fehler<table><tr><td>FALSE</td><td>Kein Fehler (Kommandierung zulässig oder Warnung)</td></tr><tr><td>TRUE</td><td>Fehler</td></tr></table></td></tr>
<tr><td>iErrID</td><td>INT</td><td>Fehler Identnummer: Diagnosenummer wird ausgegeben<table><tr><td colspan="2">iErrID = 0</td><td>Kein Fehler</td></tr><tr><td>iErrID ≠ 0</td><td>boErr = TRUE</td><td>Fehler</td></tr><tr><td>iErrID ≠ 0</td><td>boErr = FALSE</td><td>Warnung</td></tr></table></td></tr>
</table>

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein** / **Bibliothek** POSITION_JERK_1 / AmkAfl.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	POSITION_RELATIV_VAJ

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.3 04_Manual

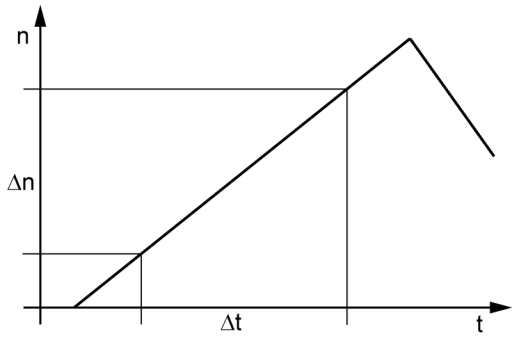
4.2.1.1.3.1 MANUAL_DETERMINE_INERTIA (FB)

Der Funktionsbaustein 'MANUAL_DETERMINE_INERTIA' ermittelt das Trägheitsmoment des Antriebs und der verbundenen Mechanik. Dazu beschleunigt der Antrieb an der angegebenen Momentgrenze auf die vorgegebene Maximalgeschwindigkeit.

$J = \frac{M}{\alpha}$ Trägheitsmoment

$\alpha = \frac{\Delta\omega}{\Delta t}$ Winkelbeschleunigung

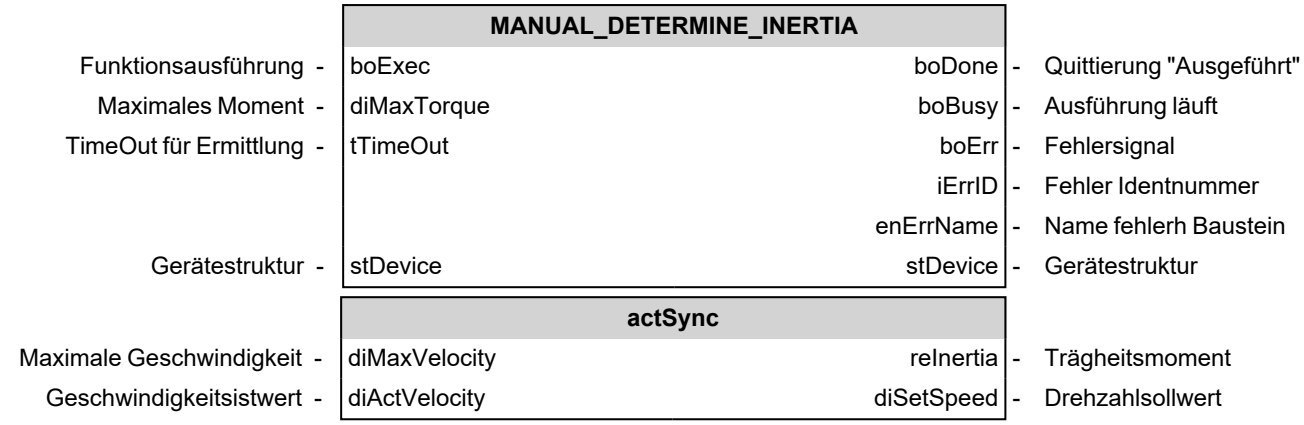
$\omega = 2 * \pi * n$ Winkelgeschwindigkeit
n – Drehzahl / 1/s



Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Anwender Interface



Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
diMaxTorque	DINT	Maximales Moment [0,1% Mn]
tTimeOut	TIME	Timeout Zeit Zeit, innerhalb der die Maximalgeschwindigkeit erreicht sein muss. Andernfalls wird die Ausführung des Funktionsbausteins abgebrochen. [ms]

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diMaxVelocity	DINT	Maximale Geschwindigkeit [0,0001 U/min]
diActVelocity	DINT	Aktuelle Geschwindigkeit [0,0001 U/min]]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>iErrID</td><td>INT</td><td>Fehler Identnummer: Diagnosenummer wird ausgegeben
<table>
<tr><td colspan="2">iErrID = 0</td><td>Kein Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = TRUE</td><td>Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = FALSE</td><td>Warnung</td></tr>
</table>
enErrName = NONE
<table>
<tr><th>iErrID</th><th>Bedeutung</th></tr>
<tr><td>1</td><td>Fehleingabe: Benötigte Eingabevariablen enthalten Werte ≤ 0
=> Eingabevariablen überprüfen</td></tr>
<tr><td>2</td><td>Timeout: Antrieb hat es innerhalb 'tTimeOut' nicht geschafft, auf 'diMaxVelocity' zu beschleunigen
=> eventuell 'diMaxTorque' erhöhen</td></tr>
</table>
enErrName ≠ NONE
Fehlernummer gemäß 'enErrName'</td></tr>
<tr><td>enErrName</td><td>ENUM</td><td>EN_FB_NAME
Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird.
Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.
<table>
<tr><th>Baustein</th><th>Bibliothek</th></tr>
<tr><td>READ_ID_DINT</td><td>AmkSystem.lib</td></tr>
<tr><td>WRITE_ID_DINT_TMP</td><td>AmkSystem.lib</td></tr>
</table></td></tr>
<tr><td>strFbInstance</td><td>STRING</td><td>Aufrufpfad des Bausteins</td></tr>
</table>

Ausgangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
reInertia	REAL	Ermitteltes Trägheitsmoment [kg x m^2]
diSetSpeed	DINT	Geschwindigkeitssollwert [0,0001 U/min]

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
MANUAL_DETERMINE_INERTIA	MANUAL_DETERMINE_INERTIA.actSync

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.3.2 MANUAL_JOG (FB)

Der Funktionsbaustein 'MANUAL_JOG' realisiert den Tippbetrieb (Plus / Minus) in Lageregelung.
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	MANUAL_JOG			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Not-Halt -	boEmergency_Stop	boBusy	-	Ausführung läuft
Not-Halt Rampe -	diEmergency_StopRamp	boErr	-	Fehlersignal
Tippen positive Richtg -	boPlus	iErrID	-	Name fehlerh Baustein
Tippen negative Richtg -	boMinus	enErrName	-	Fehler Identnummer
Tippgeschwindigkeit -	udVelocity			
Tippbeschleunigung -	udAccel			
Lagesollwert -	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
boPlus / boMinus	BOOL	Tippen in positive bzw. negative Richtung.
udVelocity	UDINT	Geschwindigkeit für den Tippbetrieb [Inkremente/s]
udAccel	UDINT	Beschleunigungs- / Bremsrampe [Inkremente/s^2]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.

Baustein	Bibliothek
VGEN_A	AmkBase.lib
RATIO_INC	AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	MANUAL_JOG

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.3.3 MANUAL_JOG_VAJ (FB)

Der Funktionsbaustein 'MANUAL_JOG_VAJ' realisiert den Tippbetrieb (Plus / Minus) in Lageregelung. Dabei kann der Anwender neben Position, Geschwindigkeit und Beschleunigung auch den Ruck vorgeben.

$v = \frac{\Delta s}{\Delta t}$ Geschwindigkeit

$a = \frac{\Delta v}{\Delta t} = \frac{\Delta s}{\Delta t^2}$ Beschleunigung

$j = \frac{\Delta a}{\Delta t} = \frac{\Delta v}{\Delta t^2} = \frac{\Delta s}{\Delta t^3}$ Ruck

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		MANUAL_JOG_VAJ			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Not-Halt	-	boEmergency_Stop	boBusy	-	Ausführung läuft
Not-Halt Rampe	-	diEmergency_StopRamp	boErr	-	Fehlersignal
Tippen positive Richtg	-	boPlus	iErrID	-	Fehler Identnummer
Tippen negative Richtg	-	boMinus	enErrName	-	Name fehlerh Baustein
Tippgeschwindigkeit	-	udVelocity			
Tippbeschleunigung	-	udAccel			
Tippverzögerung	-	udDecel			
Ruck beim Start	-	udAccJerk			
Ruck beim Stop	-	udDecJerk			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
boPlus / boMinus	BOOL	Tippen in positive bzw. negative Richtung.
udVelocity	UDINT	Geschwindigkeit für den Tippbetrieb [Inkremente/s]
udAccel / udDecel	UDINT	Beschleunigungs- / Bremsrampe [Inkremente/s^2]
udAccJerk / udDecJerk	UDINT	Ruck bei der Beschleunigung / Verzögerung [Inkremente/s^3]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben

Name	Typ	Beschreibung
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: VGEN_AJ – Bibliothek: AmkBase.lib Baustein: RATIO_INC – Bibliothek: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	MANUAL_JOG_VAJ

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.3.4 MANUAL_STEP (FB)

Der Funktionsbaustein 'MANUAL_STEP' realisiert den Schrittbetrieb (Plus / Minus) in Lageregelung.
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		MANUAL_STEP			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Not-Halt Rampe	-	diEmergency_StopRamp	iErrID	-	Fehler Identnummer
Tippen positive Richtg	-	boPlus	enErrName	-	Name fehlerh Baustein
Tippen negative Richtg	-	boMinus			
Schrittweite	-	diStep			
Schrittgeschwindigkeit	-	udVelocity			
Schrittbeschleunigung	-	udAccel			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
boPlus / boMinus	BOOL	Tippen in positive bzw. negative Richtung. Der Antrieb stoppt die Positionierung, wenn das Signal vor Erreichen der Endposition abfällt.
diStep	DINT	Schrittweite: Weg, der bei Dauerbetätigung der Tipptaste maximal abgefahren wird
udVelocity	UDINT	Geschwindigkeit für den Schrittbetrieb [Inkremente/s]
udAccel	UDINT	Beschleunigungs- / Bremsrampe [Inkremente/s^2]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein** / **Bibliothek** POS_1 / AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	MANUAL_STEP

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.3.5 MANUAL_VELOCITY (FB)

Der Funktionsbaustein 'MANUAL_VELOCITY' realisiert die Drehzahlvorgabe mit Not-Halt-Funktion. Die Rampenzeit 'diVelocityRamp' ist konstant. Dadurch werden die unterschiedlichen Sollgeschwindigkeiten immer in der gleichen Zeit erreicht.

Beispiel 1: Sprung am Eingang 'diInVal' 0→100 [U/min]
Rampenzeit 1000 [ms]
Daraus folgt: Der Ausgangswert 'diOutVal' rampt innerhalb 1s von 0→ 100.0000 [0,0001 U/min],

Beispiel 2: Sprung am Eingang 'diInVal' 0→1000 [U/min]
Rampenzeit 1000 [ms]
Daraus folgt: Der Ausgangswert 'diOutVal' rampt innerhalb 1s von 0→ 1000.0000 [0,0001 U/min],

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		MANUAL_VELOCITY			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert Geschw		diInVal	bo0Velocity	-	Keine Sollwertausgabe
Not-Halt	-	boEmergency_Stop	diOutVal	-	Ausgangswert Geschw
Not-Halt Rampe	-	diEmergency_StopRamp			
Geschwindigkeitsrampe	-	diVelocityRamp			
SERCOS Zykluszeit	-	diID2_SERCOS_cycle			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Vorgabe des Geschwindigkeitssollwertes [U/min] Eingabe des Geschwindigkeitsollwertes in U/min!
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diVelocityRamp	DINT	Geschwindigkeitsrampe [ms]
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
bo0Velocity	BOOL	Wenn 'bo0Vel' = TRUE ist, wird kein Sollwert mehr ausgegeben.
diOutVal	DINT	Geschwindigkeitsollwert zur Weiterleitung an den Antrieb [0,0001 U/min] Ausgabe des Geschwindigkeitsollwertes in 0,0001 U/min!

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	MANUAL_VELOCITY

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.3.6 MANUAL_VELOCITY_1 (FB)

Der Funktionsbaustein 'MANUAL_VELOCITY_1' realisiert die Drehzahlvorgabe mit Not-Halt-Funktion. Die Steigung der vorgegebenen Rampe in 'diVelocityRamp' ist konstant. Dadurch werden die unterschiedlichen Sollgeschwindigkeiten in unterschiedlichen Zeiten erreicht.

Beispiel 1: Sprung am Eingang 'diInVal' 0→100 [U/min]
Rampenanstieg 100.0000 [0,0001 U/s]
Daraus folgt: Der Ausgangswert 'diOutVal' rampt innerhalb 1s von 0→ 100.0000 [0,0001 U/min].

Beispiel 2: Sprung am Eingang 'diInVal' 0→1000 [U/min]
Rampenanstieg 100.0000 [0,0001 U/s]
Daraus folgt: Der Ausgangswert 'diOutVal' rampt innerhalb 10s von 0→ 1000.0000 [0,0001 U/min].

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		MANUAL_VELOCITY_1			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Not-Halt	-	boEmergency_Stop	bo0Velocity	-	Keine Sollwertausgabe
Not-Halt Rampe	-	diEmergency_StopRamp	diOutVal	-	Ausgangswert Geschwindigkeit
Eingangswert Geschwindigkeit	-	diInVal			
Geschwindigkeitsrampe	-	diVelocityRamp			
SERCOS Zykluszeit	-	diID2_SERCOS_cycle			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Vorgabe des Geschwindigkeitssollwertes [U/min] Eingabe des Geschwindigkeitsollwertes in U/min!
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diVelocityRamp	DINT	Rampenanstieg [0,0001 U/s]
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
bo0Velocity	BOOL	Wenn 'bo0Vel' = TRUE ist, wird kein Sollwert mehr ausgegeben.

Name	Typ	Beschreibung
diOutVal	DINT	Geschwindigkeitsollwert zur Weiterleitung an den Antrieb [0,0001 U/min] Ausgabe des Geschwindigkeitsollwertes in 0,0001 U/min!

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	MANUAL_VELOCITY_1

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.4 06_Follow

4.2.1.1.4.1 FOLLOW_AXIS_CAM_CONT (FB)

Der Funktionsbaustein 'FOLLOW_AXIS_CAM_CONT' ermöglicht die konfigurierbare Ausgabe eines Binärsignals als Funktion der Eingangsvariablen 'diInVal' (z.B. Weginformation) in Form eines Nockenschaltwerks. Das heißt, die Signalzustände des Binärausgangs können mittels Nockenvorgabe, z.B. in Abhängigkeit von Positionswerten festgelegt werden.

Die Schaltpunkte der Spur können durch die Vorgabe der Nocken Ein- / Ausschaltposition ('di..CamPositionOn' / 'di..CamPositionOff') als Eingangswerte definiert werden. Maximal 16 Nocken können beliebig auf die Spur verteilt werden. Die Schaltpunkte (Nocken) können "online", d.h. während der Funktionsblock aktiviert ist ('boEnable' = TRUE), geändert werden.

Weiter lassen sich über die entsprechenden Eingangsvariablen folgende Merkmale beeinflusst werden:

- enMode: Betriebsart
- tFilter: Filterzeitkonstante
- tDelay Totzeit
- uiHyst: Hysterese

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Betriebsart

Bei der Betriebsart wird unterschieden zwischen:

- enMode = CAM_CONT_INC: Inkrementelle Eingangswertvorgabe
  Die Eingangsvariable 'diInVal' wird als 32-Bit vorzeichenbehaftete Festpunktzahl behandelt (32-Bit Integer Wert). Durch den Funktionsbaustein werden die Eingangswertdifferenzen zweier zeitlich aufeinanderfolgender Eingangsinformationen zu einem positiven 32-Bit Wert aufsummiert. Bei Überschreitung eines konfigurierbaren Modulowertes (z.B. Inkremente pro Umdrehung) wird erneut bei Wert 0 begonnen (Modulozählung).
- enMode = CAM_CONT_ABS: Absolute Eingangswertvorgabe
  Die Eingangsvariable 'diInVal' wird als 32-Bit vorzeichenbehaftete Festpunktzahl behandelt (32-Bit Integer Wert). Ein Überschwingen am Verfahrbereichsende wird begrenzt.

Filterung im Rahmen der Totzeitkompensation

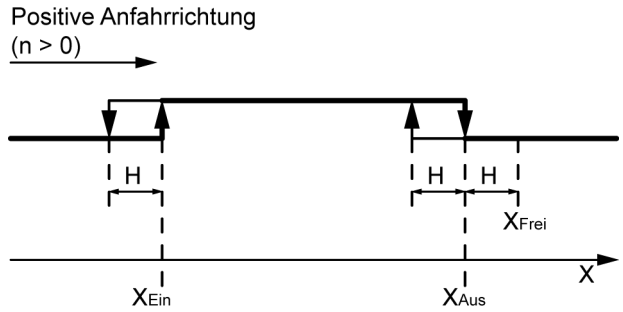
Zur Dämpfung des Einflusses von Geschwindigkeitsänderungen im Rahmen der Totzeitkompensation wird eine Mittelwertbildung über mehrere Geschwindigkeitswerte durchgeführt. Die Filterzeitkonstante 'tFilter' bestimmt die Anzahl der Geschwindigkeitswerte, über die gemittelt wird

uiCycleTime = ID2 'SERCOS-Zykluszeit'

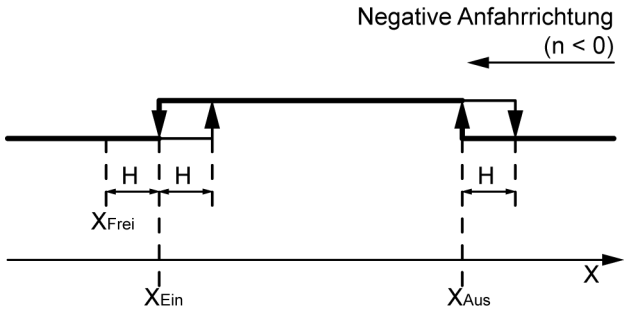
Hysterese

Die Hysteresebildung (X_{Ein}, X_{Aus}) wird durch die folgenden Abbildungen verdeutlicht:

Positive Anfahrrichtung: n > 0; X größer werdend



Negative Anfahrrichtung: n < 0; X kleiner werdend



Für eine **positive Anfahrrichtung** ergibt sich folgendes Verhalten:

1. Ausgehend von einer Position $X < X_{Ein}$ => Nockeninformation = 0
2. Bei $X_{Ein} \leq X < X_{Aus}$ => Nockeninformation = 1
3. Bei Zurückdrehen bis zur Position $X \geq (X_{Ein} – H)$ => Nockeninformation = 1
   Bei weiterem Zurückdrehen $X < (X_{Ein} – H)$ => Nockeninformation = 0
4. Beim Vorwärtsdrehen ab $X \geq X_{Aus}$ => Nockeninformation = 0
5. Zurückdrehen bei $X \leq X_{Frei}$ (X_{Frei} = XAus + H) => Nockeninformation = 0, bis $X \leq (X_{Aus} - H)$
   (Bei weiterem Zurückdrehen $X \leq (X_{Aus} - H)$ => Nockeninformation = 1)
6. Wird nach Erreichen der Position $X \geq X_{Frei}$ zurückgedreht, so wird das Nockensignal gemäß der 'negativen Anfahrrichtung' gebildet.

Für eine **negative Anfahrrichtung** ergibt sich folgendes Verhalten:

1. Ausgehend von einer Position $X \geq X_{Aus}$ => Nockeninformation = 0
2. Bei $X_{Ein} \leq X < X_{Aus}$ => Nockeninformation = 1
3. Bei Vordrehen bis zur Position $X < (X_{Aus} + H)$ => Nockeninformation = 1
   Bei weiterem Vordrehen $X \geq (X_{Aus} + H)$ => Nockeninformation = 0
4. Beim Zurückdrehen ab $X < X_{Ein}$ => Nockeninformation = 0
5. Zurückdrehen bei $X > X_{Frei}$ ($X_{Frei} = X_{Ein} – H$) => Nockeninformation = 0 bis $X < (X_{Ein} + H)$
   (Bei weiterem Zurückdrehen $X \geq (X_{Ein} + H)$ => Nockeninformation = 1).
6. Wird nach Erreichen der Position $X < X_{Frei}$ zurückgedreht, so wird das Nockensignal gemäß der 'positiven Anfahrrichtung' gebildet.

Die Umschaltung zwischen der Hysteresebildung für positive bzw. negative Anfahrrichtung erfolgt somit, wenn der Nocken um mindestens den Hysteresebetrag (bzw. den Hysteresebetrag + 1) überfahren wurde. Das heißt, die Position X_{Frei} wurde erreicht bzw. überschritten in der Richtung, in der der Nocken angefahren wurde.

Anwender Interface

		FOLLOW_AXIS_CAM_CONT			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Auswahlmodus Betriebsart	-	enMode	boErr	-	Fehlersignal
Eingangswert	-	diInVal	iErrID	-	Fehler Identnummer
Modulowert	-	udModulo	enErrName	-	Name fehlerh Baustein
Filterzeit	-	tFilter	boOutVal	-	Ausgabebit
Totzeitkonstante	-	tDelay			
Hysterese	-	uiHyst			
Anzahl der Nocken	-	uiNumberOfCams			
Einschaltpos. Nocken 1	-	di1CamPositionOn			
Ausschaltpos. Nocken 1	-	di1CamPositionOff			
	-	:			
	-	:			
Einschaltpos. Nocken 16	-	di16CamPositionOn			
Ausschaltpos. Nocken 16	-	di16CamPositionOff			
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>boEnable</td><td>BOOL</td><td>Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins.
Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet.
Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.</td></tr>
<tr><td>enMode</td><td>ENUM</td><td>EN_CAM_CONT_MODE
Auswahlmodus Betriebsart (s. o.)<table><tr><td>CAM_CONT_INC</td><td>inkrementelle Eingangswertauswertung (Standardeinstellung)</td></tr><tr><td>CAM_CONT_ABS</td><td>absolute Eingangswertauswertung</td></tr></table></td></tr>
<tr><td>diInVal</td><td>DINT</td><td>Eingangswert (Wegbezug) [Inkremente]</td></tr>
<tr><td>udModulo</td><td>UDINT</td><td>Modulowert
bei diesem Wert beginnt in der Betriebsart 'enMode' = CAM_CONT_INC die Nockentabellenauswertung erneut mit dem Wert "0"<table><tr><td>Bereich</td><td>0 … $+2^{31}$-1</td></tr><tr><td>Standardwert</td><td>20000</td></tr></table></td></tr>
<tr><td>tFilter</td><td>TIME</td><td>Filterzeitkonstante
Dämpfung des Einflusses von Geschwindigkeitsänderungen im Rahmen der Totzeitkompensation. (Standardwert = 1 ms)</td></tr>
<tr><td>tDelay</td><td>TIME</td><td>Totzeitkonstante zur Berechnung des von der aktuellen Geschwindigkeit (Eingangswertdifferenz) abhängigen Versatzes (Voreilen) der Binärinformation, im Rahmen der Totzeitkompensation. [ms]
(Standardwert = 0 = inaktiv)</td></tr>
</table>

Name	Typ	Beschreibung
uiHyst	UINT	Hysterese (s. o.) Hysteresewert , mit dem die Ein- und Ausschaltflanke (X_{Ein}, X_{Aus}) eines Nockensignals behaftet ist. [inc] (Standardwert = 0 = inaktiv) In Zusammenhang mit einer Totzeitkompensation muss die Hysterese größer als die Totzeitkompensationsstrecke X_{tot} gewählt werden. Dabei gilt: X_{tot} = Totzeitkompensationsstrecke [Inkremente] T_{tot} = Totzeit [ms] n = Drehzahl [U/min] G = Geberauflösung [Inkr/U] Bei 'enMode' = CAM_CONT_INC muss gelten: $X_{Aus} > X_{Ein}$: H < Modulowert – ($X_{Aus} – X_{Ein}$) $X_{Aus} < X_{Ein}$: H < $X_{Ein} – X_{Aus}$
uiNumberOfCams	UINT	Anzahl der aktiven Nocken (Standardwert = 1)
di1CamPositionOn ... di16CamPositionOn	DINT	Einschaltpositionen für Nocken 1 ... 16 [Inkremente]
di1CamPositionOff ... di16CamPositionOff	DINT	Ausschaltpositionen für Nocken 1 ... 16 [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boOutVal	BOOL	Ausgangssignal der Nockenspuren
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: CAM_CONT – Bibliothek: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FOLLOW_AXIS_CAM_CONT

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.4.2 FOLLOW_AXIS_CONTINUOUS_TAB1_TAB2_OUTGEAR32 (FB)

Der Funktionsbaustein 'FOLLOW_AXIS_CONTINUOUS_TAB1_TAB2_OUTGEAR32' realisiert den kontinuierlichen Folgebetrieb über eine Tabellenfunktionen (Kurvenscheibe).

- Dabei können zwei verschiedene Arbeitstabellen übergeben werden.
- Die Auswahl der aktiven Arbeitstabelle erfolgt über ein Binärsignal.
- Durch das Eingangsbit 'boEnabOpTab2' = TRUE wird die Arbeitstabelle2 abgearbeitet.
- Die Tabellenumschaltung erfolgt immer im Koordinatenursprung um sprunghafte Ausgleichsbewegungen zu vermeiden.
- Wird keine zweite Arbeitstabelle angegeben oder das Eingangsbit 'boEnabOpTab2' = FALSE, wird die Arbeitstabelle1 abgearbeitet.
- Minimal erforderlich ist die Arbeitstabelle 1.

Dem Baustein können auch Tabellen mit mehr als 360 Stützpunkten übergeben werden.

Zusätzlich besitzt der Baustein eine Ausganggetriebestufe. Dadurch können die normierten Kurvenscheiben an unterschiedliche Gegebenheiten angepasst werden.

Die Berechnung der Getriebestufe erfolgt im 32 Bit-Modus, somit können für dem Multiplikator / Divisor maximal 16 Bit-Werte übergeben werden.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		FOLLOW_AXIS_CONTINUOUS_TAB1_TAB2_OUTGEAR32			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Freigabe Arbeitstabelle	-	boEnabOpTab2	boTab1Busy	-	Arbeittabelle läuft
Not-Halt	-	boEmergency_Stop	boTab2Busy	-	Arbeittabelle läuft
Not-Halt Rampe	-	diEmergency_StopRamp	boDone	-	Tabelle fertig
SERCOS Zykluszeit	-	diID2_SERCOS_cycle	boErr	-	Fehlersignal
Getriebe Multiplikator	-	iMultiplier	iErrID	-	Fehler Identnummer
Getriebe Divisor	-	uiDivider	enErrName	-	Name fehlerh Baustein
Eingangswert	-	diInVal			
Arbeitstabelle 1	-	pstOpTab1			
Arbeitstabelle 2	-	pstOpTab2			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.

Name	Typ	Beschreibung
boEnabOpTab2	BOOL	• Bei 'boEnabOpTab2' = TRUE wird die zweite Arbeitstabelle freigegeben, die Umschaltung erfolgt im Koordinatenursprung. • Mit 'boEnabOpTab2' = FALSE wird in die Arbeitstabelle1 zurückgeschaltet. Die Umschaltung erfolgt ebenfalls im Koordinatenursprung.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]
iMultiplier	INT	Getriebe Multiplikator / Faktor [1]
uiDivider	UINT	Getriebe Divisor / Teiler [1]
diInVal	DINT	Eingangsinkremente vom Mastergeber (Inkrementenquelle, der gefolgt werden soll) [Inkremente]
pstOpTab1 / pstOpTab2	POINTER	POINTER TO ST_PROF_TAB Verweis auf Arbeitstabelle (Tabellengestützte Kurvenscheibe) (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boTab1Busy / boTab2Busy	BOOL	Tabellenabarbeitung läuft.
boDone	BOOL	Tabellenabarbeitung ist fertig.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein**: CAM_PROF **Bibliothek**: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FOLLOW_AXIS_CONTINUOUS_TAB1_TAB2_OUTGEAR32

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.4.3 FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2 (FB)

Der Funktionsbaustein 'FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2' realisiert den kontinuierlichen Folgebetrieb über eine Tabellenfunktionen (Kurvenscheibe).

- Dabei können zwei verschiedene Arbeitstabellen übergeben werden.
- Die Auswahl der aktiven Arbeitstabelle erfolgt über ein Binärsignal.
- Durch das Eingangsbit 'boEnabOpTab2' = TRUE wird die Arbeitstabelle2 abgearbeitet.
- Die Tabellenumschaltung erfolgt immer im Koordinatenursprung, um sprunghafte Ausgleichsbewegungen zu vermeiden.
- Wird keine zweite Arbeitstabelle angegeben oder das Eingangsbit 'boEnabOpTab2' = FALSE, wird die Arbeitstabelle1 abgearbeitet.
- Minimal erforderlich ist die Arbeitstabelle1.

Dem Baustein können auch Tabellen mit mehr als 360 Stützpunkten übergeben werden.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Freigabe Arbeitstabelle -	boEnabOpTab2	boTab1Busy	-	Arbeittabelle läuft
Not-Halt -	boEmergency_Stop	boTab2Busy	-	Arbeittabelle läuft
Not-Halt Rampe -	diEmergency_StopRamp	boDone	-	Tabelle fertig
SERCOS Zykluszeit -	diID2_SERCOS_cycle	boErr	-	Fehlersignal
Eingangswert -	diInVal	iErrID	-	Fehler Identnummer
Arbeitstabelle -	pstOpTab1	enErrName	-	Name fehlerh Baustein
Arbeitstabelle -	pstOpTab2			
Lagesollwert -	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boEnabOpTab2	BOOL	• Bei 'boEnabOpTab2' = TRUE wird die zweite Arbeitstabelle freigegeben, die Umschaltung erfolgt im Koordinatenursprung. • Mit 'boEnabOpTab2' = FALSE wird in die Arbeitstabelle1 zurückgeschaltet. Die Umschaltung erfolgt ebenfalls im Koordinatenursprung.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]
diInVal	DINT	Eingangsinkremente vom Mastergeber (Inkrementenquelle, der gefolgt werden soll) [Inkremente]

Name	Typ	Beschreibung
pstOpTab1 / pstOpTab2	POINTER	POINTER TO ST_PROF_TAB Verweis auf Arbeitstabelle (Tabellengestützte Kurvenscheibe) (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boTab1Busy / boTab2Busy	BOOL	Tabellenabarbeitung läuft.
boDone	BOOL	Tabellenabarbeitung ist fertig.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: CAM_PROF — Bibliothek: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FOLLOW_AXIS_CONTINUOUS_TABLE1_TABLE2

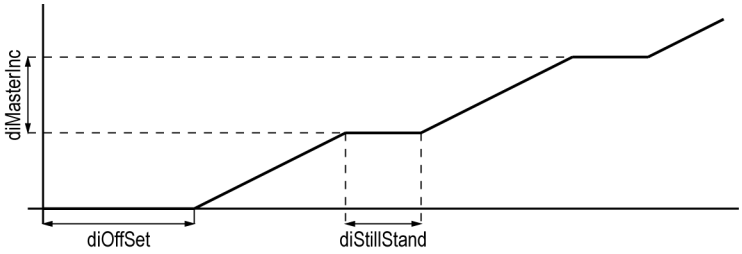
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.4.4 FOLLOW_AXIS_CRANK_DRIVE_EMULATION (FB)

Der Funktionsbaustein 'FOLLOW_AXIS_CRANK_DRIVE_EMULATION' realisiert ein Kurbelgetriebe.
Der Baustein stellt eine Weg-Weg-Funktion mit Stillstandsbereich zur Verfügung. Anwendungsgebiet sind u.a. Folientransport und Schnitt mit stehendem Messer.

- Die Kurve startet immer am Ende des Stillstandsbereichs.
- Mittels Parametrierung können verschiedene Übergangsfunktionen gewählt werden:
  - 45°-Kurve
  - Sinus-Kurve
  - Sinus²-Kurve

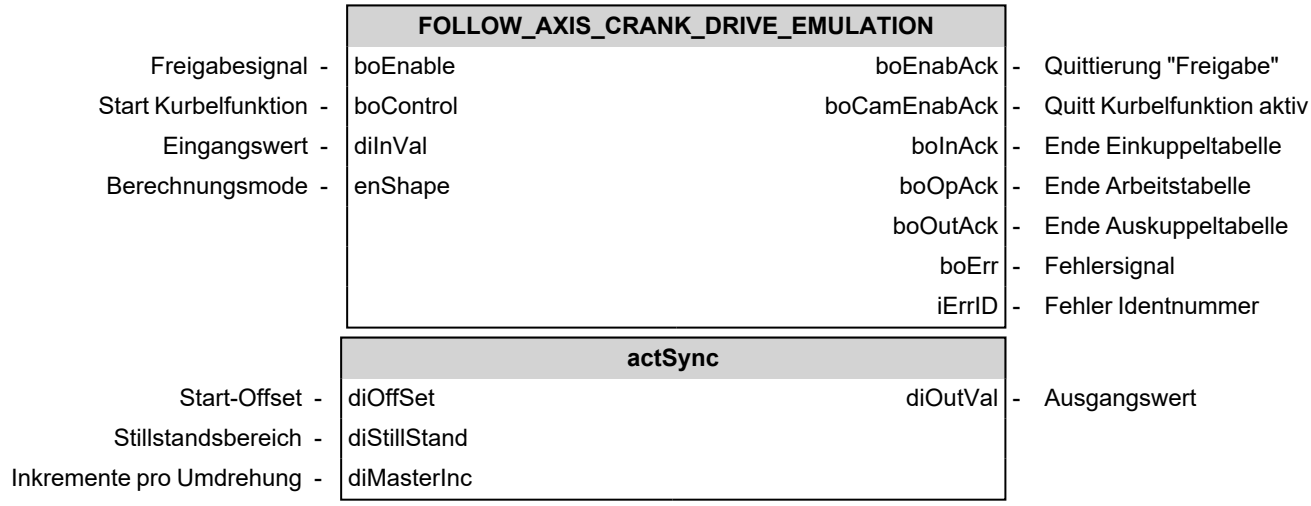
Am Anfang der Bewegung kann ein Offset definiert werden, mit dem die Kurve auf einen externen Startpunkt synchronisiert werden kann.



Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Anwender Interface



Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boControl	BOOL	Start / Stopp der Kurbelfunktion
diInVal	DINT	Eingangswert-Impulse [Inkr]

Name	Typ	Beschreibung
enShape	ENUM	EN_CRANK_DRIVE_SHAPE Kurvenform der Kurbelfunktion enCrankShapeLinear: linearer Verlauf 45° enCrankShapeSine: SINUS-Funktion enCrankShapeSquareSine: $SINUS^2$-Funktion

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diOffSet	DINT	Offset am Start der ersten Umdrehung. [Inkr] Der Offset verschiebt die Bewegungsfunktion nach hinten.
diStillStand	DINT	Stillstandsbereich bei einer Umdrehung am Ende der Umdrehung [Inkr]
diMasterInc	DINT	Anzahl Inkremente pro Umdrehung (ohne Start- Offset) [Inkr]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boCamEnabAck	BOOL	Quittierung: Kurbelfunktion aktiv
boInAck	BOOL	Quittierung Einkuppeltabelle Ende Puls am Ende der Einkuppeltabelle; 'boInAck' = TRUE für 2 Abtastzeitpunkte
boOpAck	BOOL	Quittierung Arbeitstabelle Ende Puls am Ende der Arbeitstabelle; 'boOpAck' = TRUE für 2 Abtastzeitpunkte
boOutAck	BOOL	Quittierung Auskuppeltabelle Ende Puls am Ende der Auskuppeltabelle; 'boOutAck' = TRUE für 2 Abtastzeitpunkte
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	BOOL	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung 1 - 99: Fehlercode entsprechend 'CAM_PROF' (AmkBase.lib) 100: Falscher Wert in Variable 'enShape' 101: Parameter 'diMasterInc' ≤ 0 102: Parameter 'diStillStand' zu groß ('diStillStand' ≤ 'diMasterInc' / 2) 103: Parameter 'diOffSet' < 0

Ausgangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diOutVal	DINT	Ausgangswert - Impulse [Inkr] (können z.B. mit 'SET_VAL_I' an den Antrieb ausgegeben werden)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
FOLLOW_AXIS_CRANK_DRIVE_EMULATION	FOLLOW_AXIS_CRANK_DRIVE_EMULATION.actSync

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.4.5 FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2 (FB)

Der Funktionsbaustein 'FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2' realisiert den kontinuierlichen Folgebetrieb über eine Tabellenfunktionen (Kurvenscheibe).

- Über eine Ein-/Auskuppeltabelle wird die Arbeitstabelle ein- oder ausgekuppelt.
- Ist keine Einkuppeltabelle übergeben, wird mit der angewählten Arbeitstabelle eingekuppelt.
- Gleiches gilt für den Auskuppelvorgang: ohne übergebene Auskuppeltabelle wird mit der aktiven Arbeitstabelle ausgekuppelt, wenn das Ausführungssignal 'boExec' entzogen wird.
- Es können zwei verschiedene Arbeitstabellen übergeben werden.
- Durch das Eingangsbit 'boEnabOpTab2' = TRUE wird die Arbeitstabelle2 aktiviert. Die Tabellenumschaltung erfolgt immer im Koordinatenursprung, um sprunghafte Ausgleichsbewegungen zu vermeiden.
- Wird keine zweite Arbeitstabelle angegeben oder ist das Eingangsbit 'boEnabOpTab2' = FALSE, wird die Arbeitstabelle1 abgearbeitet.
- Mit der positiven Flanke von 'boEnable', wird die Erfassung der Eingangsinkremente aktiviert. Das heißt, ab diesem Zeitpunkt führen Inkrementänderungen an der Eingangsvariable 'diInVal' zu einer Änderung des Masterbezugspunkts (X).
- Die per 'pstInTab', 'pstOpTab' und 'pstOutTab' zugeordneten Einkuppel-, Arbeits- und Auskuppeltabellen werden ausgewertet. Minimal erforderlich ist die Arbeitstabelle.
- Die Steuerung des Ein-/Auskuppelvorgangs erfolgt mit dem Eingang 'boExec'.
- Dabei gilt: bei positiver Flanke an 'boExec' und X ≤ 'udInAngle' wird der Tabelleninterpolator aktiviert. Das heißt, es werden an der Sollwertsenke 'diSetPosition' Inkrementänderungen gemäß der Einkuppeltabelle ausgegeben.
- Am Ende der Einkuppeltabelle erfolgt ein Übergang auf die Arbeitstabelle.
- Die Arbeitstabelle wird so lange abgearbeitet, bis eine negative Flanke an 'boExec' und X ≤ 'udOutAngle' erkannt wird. Dann erfolgt ein Übergang in die Auskuppeltabelle mit automatischem Stopp der Interpolation am Ende der Auskuppeltabelle.

Dem Baustein können auch Tabellen mit mehr als 360 Stützpunkten übergeben werden.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Freigabe Arbeitstabelle	-	boEnabOpTab2	boTab1Busy	-	Arbeittabelle läuft
Funktionsausführung	-	boExec	boTab2Busy	-	Arbeittabelle läuft
Not-Halt	-	boEmergency_Stop	boDone	-	Tabelle fertig
Not-Halt Rampe	-	diEmergency_StopRamp	boErr	-	Fehlersignal
SERCOS Zykluszeit	-	diID2_SERCOS_cycle	iErrID	-	Fehler Identnummer
Eingangswert	-	diInVal	enErrName	-	Name fehlerh Baustein
Einkuppelbereich	-	udInAngle			
Auskuppelbereich	-	udOutAngle			
Einkuppeltabelle	-	pstInTab			
Arbeitstabelle	-	pstOpTab1			
Arbeitstabelle	-	pstOpTab2			
Auskuppeltabelle	-	pstOutTab			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boEnabOpTab2	BOOL	• Bei 'boEnabOpTab2' = TRUE wird die zweite Arbeitstabelle freigegeben, die Umschaltung erfolgt im Koordinatenursprung. • Mit 'boEnabOpTab2' = FALSE wird in die Arbeitstabelle1 zurückgeschaltet. Die Umschaltung erfolgt ebenfalls im Koordinatenursprung.
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]
diInVal	DINT	Eingangsinkremente vom Mastergeber (Inkrementenquelle, der gefolgt werden soll) [Inkremente]
udInAngle / udOutAngle	UDINT	Ein- bzw. Auskuppelbereich auf der Tabellen X-Achse. Dieser Wert gibt an, bis zu welcher Tabellen-X-Position, vom Koordinatenursprung (0), ein Ein- bzw. Auskuppeln zulässig ist. (0... udInAngle)
pstInTab / pstOutTab	POINTER	POINTER TO ST_PROF_TAB Verweis auf Ein- bzw. Auskuppeltabelle (Tabellengestützte Kurvenscheibe) (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)
pstOpTab1 / pstOpTab2	POINTER	POINTER TO ST_PROF_TAB Verweis auf Arbeitstabelle (Tabellengestützte Kurvenscheibe) (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boTab1Busy / boTab2Busy	BOOL	Tabellenabarbeitung läuft.
boDone	BOOL	Tabellenabarbeitung ist fertig.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein** / **Bibliothek** CAM_PROF / AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FOLLOW_AXIS_SWITCHING_TABLE1_TABLE2

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.4.6 FOLLOW_AXIS_TABLE_CONTINUOUS (FB)

Der Funktionsbaustein 'FOLLOW_AXIS_TABLE_CONTINUOUS' realisiert den kontinuierlichen Folgebetrieb über eine Tabellenfunktionen (Kurvenscheibe).

Dabei sind die Inkremente des Masters (des Gebers, dem zu folgen ist) am Eingang 'diInVal' und die abzuarbeitende Folgetabelle ist als Zeiger am Eingang 'pstOpTab' zu übergeben. Dem Baustein können auch Tabellen mit mehr als 360 Stützpunkten übergeben werden.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		FOLLOW_AXIS_TABLE_CONTINUOUS			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert	-	diInVal	boErr	-	Fehlersignal
Not-Halt	-	boEmergency_Stop	iErrID	-	Fehler Identnummer
Not-Halt Rampe	-	diEmergency_StopRamp	enErrName	-	Name fehlerh Baustein
SERCOS Zykluszeit	-	diID2_SERCOS_cycle			
Arbeitstabelle	-	pstOpTab			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Eingangsinkremente vom Mastergeber (Inkrementenquelle, der gefolgt werden soll) [Inkremente]

Name	Typ	Beschreibung
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]
pstOpTab	POINTER	POINTER TO ST_PROF_TAB Verweis auf Arbeitstabelle (Tabellengestützte Kurvenscheibe) (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: CAM_PROF; Bibliothek: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FOLLOW_AXIS_TABLE_CONTINUOUS

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.4.7 FOLLOW_AXIS_TABLE_ENGAGE_DISENGAGE (FB)

Der Funktionsbaustein 'FOLLOW_AXIS_TABLE_ENGAGE_DISENGAGE' realisiert den kontinuierlichen Folgebetrieb über eine Tabellenfunktionen (Kurvenscheibe).

- Über eine Ein-/Auskuppeltabelle kann die Arbeitstabelle ein- oder ausgekuppelt werden.
- Mit der positiven Flanke von 'boEnable' wird die Erfassung derEingangsinkremente aktiviert. Das heißt, ab diesem Zeitpunkt führen Inkrementänderungen an der Eingangsvariablen diInVal zu einer Änderung des Masterbezugspunkts (X).
- Die per 'pstInTab', 'pstOpTab' und 'pstOutTab' zugeordneten Einkuppel-, Arbeits- und Auskuppeltabellen werden ausgewertet. Minimal erforderlich ist die Arbeitstabelle.
- Die Steuerung des Ein-/Auskuppelvorgangs erfolgt mit dem Eingang 'boExec' oder 'boTableControl'
- Dabei gilt: bei positiver Flanke an 'boExec' (bzw. 'boTableControl') und „X ≤ 'udInAngle“ wird der Tabelleninterpolator aktiviert. Das heißt, es werden an der Sollwertsenke 'diSetPosition' Inkrementänderungen gemäß der Einkuppeltabelle ausgegeben.
- Am Ende der Einkuppeltabelle erfolgt ein Übergang auf die Arbeitstabelle. Die Arbeitstabelle wird so lange abgearbeitet, bis eine negative Flanke an 'boExec' und "X ≤ udOutAngle" erkannt wird oder die unter 'uiOpNo' angegebene Anzahl abgearbeitet ist.
- Dann erfolgt ein Übergang in die Auskuppeltabelle mit automatischem Stopp der Interpolation am Ende der Auskuppeltabelle.

Dem Baustein können auch Tabellen mit mehr als 360 Stützpunkten übergeben werden.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		FOLLOW_AXIS_TABLE_ENGAGE_DISENGAGE			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Funktionsausführung	-	boExec	boBusy	-	Ausführung läuft
Not-Halt	-	boEmergency_Stop	boDone	-	Tabelle fertig
Not-Halt Rampe	-	diEmergency_StopRamp	boErr	-	Fehlersignal
SERCOS Zykluszeit	-	diID2_SERCOS_cycle	iErrID	-	Fehler Identnummer
Eingangswert	-	diInVal	enErrName	-	Name fehlerh Baustein
Einkuppelbereich	-	udInAngle			
Auskuppelbereich	-	udOutAngle			
Einkuppeltabelle	-	pstInTab			
Arbeitstabelle	-	pstOpTab			
Auskuppeltabelle	-	pstOutTab			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.

Name	Typ	Beschreibung
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]
diInVal	DINT	Eingangsinkremente vom Mastergeber (Inkrementenquelle, der gefolgt werden soll) [Inkremente]
udInAngle / udOutAngle	UDINT	Ein- bzw. Auskuppelbereich auf der Tabellen X-Achse. Dieser Wert gibt an, bis zu welcher Tabellen-X-Position, vom Koordinatenursprung (0), ein Ein- bzw. Auskuppeln zulässig ist. (0... udInAngle)
pstInTab / pstOutTab	POINTER	POINTER TO ST_PROF_TAB Verweis auf Ein- bzw. Auskuppeltabelle (Tabellengestützte Kurvenscheibe) (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)
pstOpTab	POINTER	POINTER TO ST_PROF_TAB Verweis auf Arbeitstabelle (Tabellengestützte Kurvenscheibe) (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boBusy	BOOL	Tabellenabarbeitung läuft.
boDone	BOOL	Tabellenabarbeitung ist fertig.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: CAM_PROF; Bibliothek: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FOLLOW_AXIS_TABLE_ENGAGE_DISENGAGE

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.4.8 FOLLOW_AXIS_TABLE_START_AUTOSTOP (FB)

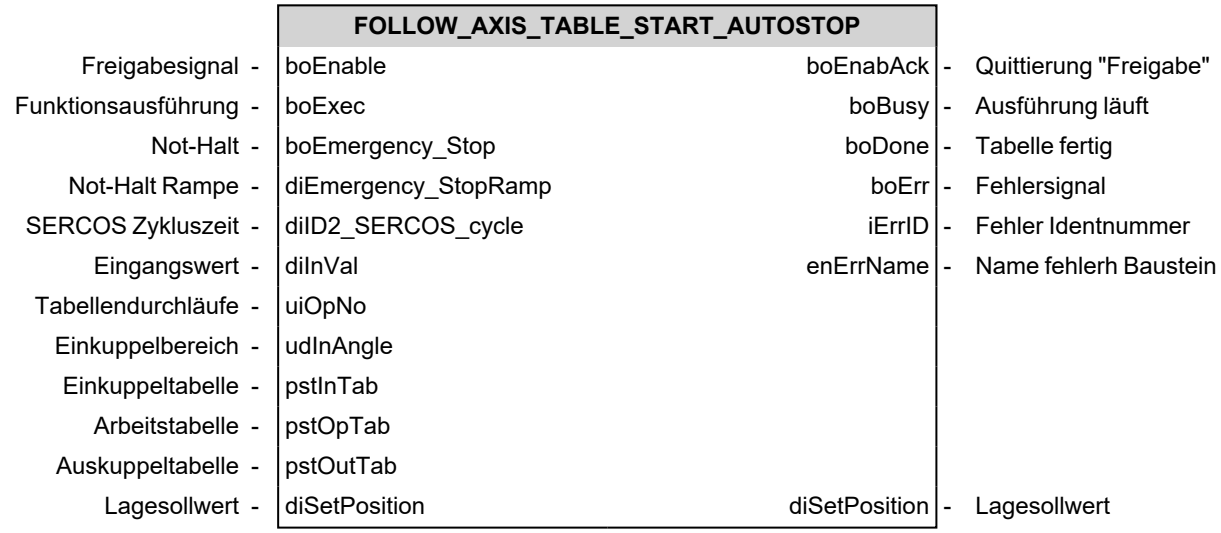
Der Funktionsbaustein 'FOLLOW_AXIS_TABLE_START_AUTOSTOP' realisiert den kontinuierlichen Folgebetrieb über eine Tabellenfunktionen (Kurvenscheibe)

- Über eine Ein-/Auskuppeltabelle kann die Arbeitstabelle eingekuppelt und nach Ablauf der vorgegebenen Tabellendurchläufe automatisch ausgekuppelt werden.
- Mit der positiven Flanke von 'boEnable', wird die Erfassung der Eingangsinkremente aktiviert. Das heißt, ab diesem Zeitpunkt führen Inkrementänderungen an der Eingangsvariable 'diInVal' zu einer Änderung des Masterbezugspunkts (X).
- Die per 'pstInTab', 'pstOpTab' und 'pstOutTab' zugeordnete Einkuppel-, Arbeits- und Auskuppeltabelle werden ausgewertet. Minimal erforderlich ist die Arbeitstabelle.
- Die Steuerung des Ein-/Auskuppelvorgangs erfolgt mit dem Eingang 'boExec'.
- Dabei gilt: bei positiver Flanke an 'boExec' und „X ≤ udInAngle“ wird der Tabelleninterpolator aktiviert. Das heißt, es werden an der Sollwertsenke 'diSetPosition' Inkrementänderungen gemäß der Einkuppeltabelle ausgegeben.
- Am Ende der Einkuppeltabelle erfolgt ein Übergang auf die Arbeitstabelle. Die Arbeitstabelle wird so lange abgearbeitet, bis die unter 'uiOpNo' angegebene Anzahl der Durchläufe erreicht ist.
- Dann erfolgt ein Übergang in die Auskuppeltabelle, mit automatischem Stopp der Interpolation am Ende der Auskuppeltabelle.

Dem Baustein können auch Tabellen mit mehr als 360 Stützpunkten übergeben werden.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]

Name	Typ	Beschreibung
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]
diInVal	DINT	Eingangsinkremente vom Mastergeber (Inkrementenquelle, der gefolgt werden soll) [Inkremente]
uiOpNo	UINT	Anzahl der abzuarbeitenden Arbeitstabellendurchläufe [1]
udInAngle	UDINT	Ein- bzw. Auskuppelbereich auf der Tabellen X-Achse. Dieser Wert gibt an, bis zu welcher Tabellen-X-Position, vom Koordinatenursprung (0), ein Ein- bzw. Auskuppeln zulässig ist. (0... udInAngle)
pstInTab / pstOutTab	POINTER	POINTER TO ST_PROF_TAB Verweis auf Ein- bzw. Auskuppeltabelle (Tabellengestützte Kurvenscheibe) (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)
pstOpTab	POINTER	POINTER TO ST_PROF_TAB Verweis auf Arbeitstabelle (Tabellengestützte Kurvenscheibe) (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boBusy	BOOL	Tabellenabarbeitung läuft.
boDone	BOOL	Tabellenabarbeitung ist fertig.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: CAM_PROF; Bibliothek: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FOLLOW_AXIS_TABLE_START_AUTOSTOP

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.4.9 FOLLOW_AXIS_TABLE_START_AUTOSTOP_1 (FB)

Der Funktionsbaustein 'FOLLOW_AXIS_TABLE_START_AUTOSTOP_1' realisiert den kontinuierlichen Folgebetrieb über eine Tabellenfunktionen (Kurvenscheibe)

- Über eine Ein-/Auskuppeltabelle kann die Arbeitstabelle eingekuppelt und nach Ablauf der vorgegebenen Tabellendurchläufe automatisch ausgekuppelt werden.
- Mit der positiven Flanke von 'boEnable', wird die Erfassung der Eingangsinkremente aktiviert. Das heißt, ab diesem Zeitpunkt führen Inkrementänderungen an der Eingangsvariable 'diInVal' zu einer Änderung des Masterbezugspunkts (X).
- Die per 'pstInTab', 'pstOpTab' und 'pstOutTab' zugeordnete Einkuppel-, Arbeits- und Auskuppeltabelle werden ausgewertet. Minimal erforderlich ist die Arbeitstabelle.
- Die Steuerung des Ein-/Auskuppelvorgangs erfolgt mit dem Eingang 'boExec'.
- Dabei gilt: bei positiver Flanke an 'boExec' und „X ≤ udInAngle“ wird der Tabelleninterpolator aktiviert. Das heißt, es werden an der Sollwertsenke 'diSetPosition' Inkrementänderungen gemäß der Einkuppeltabelle ausgegeben.
- Am Ende der Einkuppeltabelle erfolgt ein Übergang auf die Arbeitstabelle. Die Arbeitstabelle wird so lange abgearbeitet, bis die unter 'uiOpNo' angegebene Anzahl der Durchläufe erreicht ist.
- Dann erfolgt ein Übergang in die Auskuppeltabelle, mit automatischem Stopp der Interpolation am Ende der Auskuppeltabelle.

Der Baustein 'FOLLOW_AXIS_TABLE_START_AUTOSTOP_1' ist im Wesentlichen gleich dem Baustein 'FOLLOW_AXIS_TABLE_START_AUTOSTOP'.

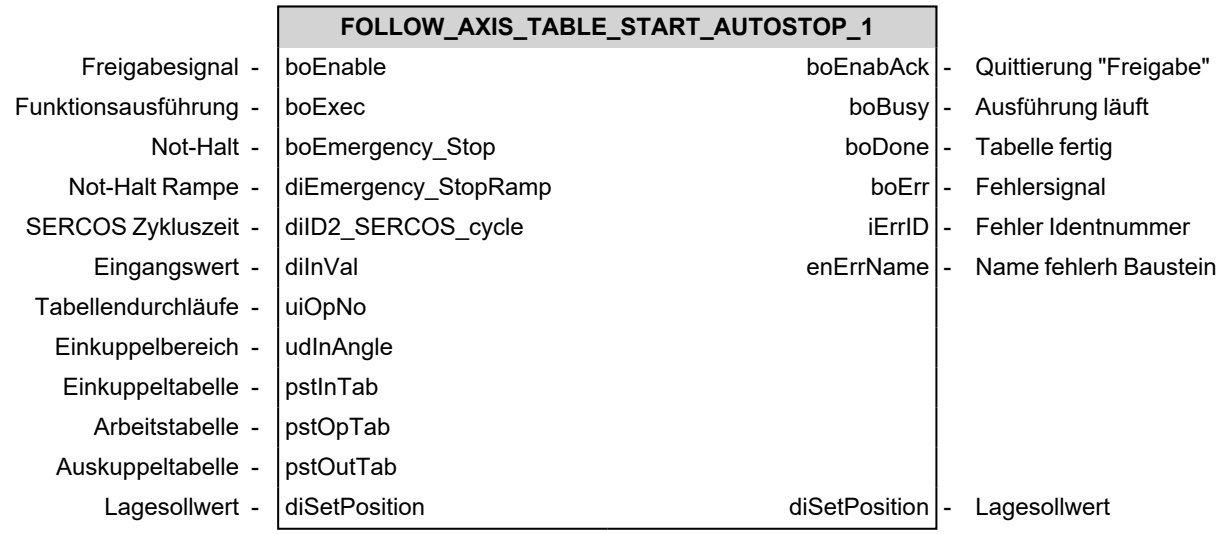
Die beiden Bausteine unterscheiden sich in den nachfolgend aufgeführten Punkten:

- Die Zeiger auf die Folgetabellen werden bei 'boEnable' = TRUE ständig übernommen. Ein 0→1 Flankenwechsel an 'boEnable' ist nicht mehr notwendig.
- Wird nur eine Arbeitstabelle übergeben, stoppt die Wertausgabe automatisch am Ende der Arbeitstabelle. Die Tabelle wird also mindestens einmal durchlaufen. Der Ein- und Auskuppelvorgang entfällt. Beim Baustein 'FOLLOW_AXIS_TABLE_START_AUTOSTOP' wird in diesem Fall die Arbeitstabelle auch für den Ein- und Auskuppelvorgang verwendet. D.h. der automatische Stopp ist minimal nach 3 Durchläufen möglich.

Dem Baustein können auch Tabellen mit mehr als 360 Stützpunkten übergeben werden.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]
diInVal	DINT	Eingangsinkremente vom Mastergeber (Inkrementenquelle, der gefolgt werden soll) [Inkremente]
uiOpNo	UINT	Anzahl der abzuarbeitenden Arbeitstabellendurchläufe [1]
udInAngle	UDINT	Ein- bzw. Auskuppelbereich auf der Tabellen X-Achse. Dieser Wert gibt an, bis zu welcher Tabellen-X-Position, vom Koordinatenursprung (0), ein Ein- bzw. Auskuppeln zulässig ist. (0... udInAngle)
pstInTab / pstOutTab	POINTER	POINTER TO ST_PROF_TAB Verweis auf Ein- bzw. Auskuppeltabelle (Tabellengestützte Kurvenscheibe) (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)
pstOpTab	POINTER	POINTER TO ST_PROF_TAB Verweis auf Arbeitstabelle (Tabellengestützte Kurvenscheibe) (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boBusy	BOOL	Tabellenabarbeitung läuft.
boDone	BOOL	Tabellenabarbeitung ist fertig.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. CAM_PROF: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FOLLOW_AXIS_TABLE_START_AUTOSTOP_1

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.4.10 FOLLOW_RATIO_INC_64 (FB)

Der Funktionsbaustein 'FOLLOW_RATIO_INC_64' realisiert ein elektronisches Getriebe.

Die Eingangsinkremente werden über Multiplikator und Divisor so angepasst, dass sie entsprechend des Übersetzungsverhältnisses am Ausgang zur Verfügung stehen. Die Berechnung des Übersetzungsverhältnisses erfolgt mit 64 Bit, dadurch lassen sich 32 Bit-Größen für Multiplikator und Divisor vorgeben.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	FOLLOW_RATIO_INC_64			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Not-Halt -	boEmergency_Stop	boErr	-	Fehlersignal
Not-Halt Rampe -	diEmergency_StopRamp	iErrID	-	Fehler Identnummer
SERCOS Zykluszeit -	diID2_SERCOS_cycle	enErrName	-	Name fehlerh Baustein
Eingangswert -	diInVal			
Getriebe Multiplikator -	diMultiplier			
Getriebe Divisor -	udDivider			
Lagesollwert -	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]
diInVal	DINT	Eingangsinkremente vom Mastergeber (Inkrementenquelle, der gefolgt werden soll) [Inkremente]
diMultiplier	DINT	Getriebe Multiplikator / Faktor [1]
udDivider	UDINT	Getriebe Divisor / Teiler [1]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben

Name	Typ	Beschreibung
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: RATIO_INC_1 – Bibliothek: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FOLLOW_RATIO_INC_64

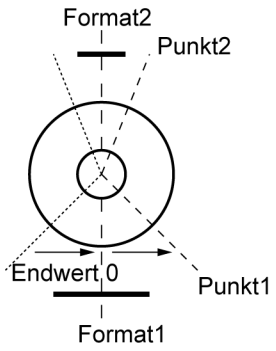
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

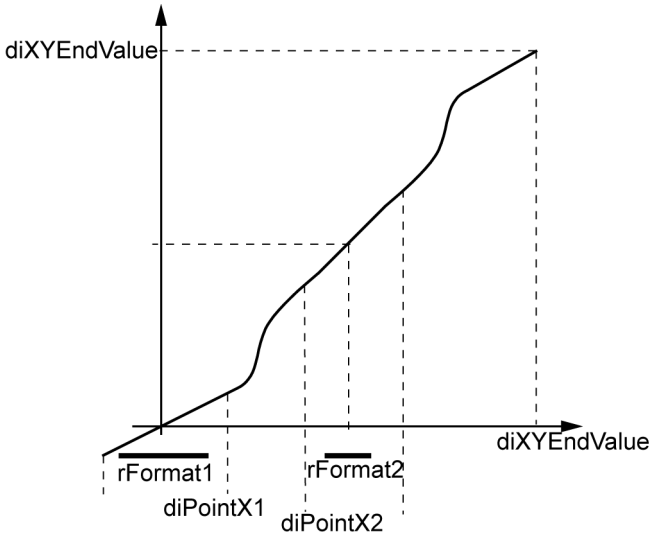
4.2.1.1.5 07_Convert/Calc

4.2.1.1.5.1 CALC_PROF_2_FORMAT_POLY (FB)

Der Funktionsbaustein 'CALC_PROF_2_FORMAT_POLY' dient der Berechnung eines Tabellenprofils.

Zwei Formate, die sich gegenüberliegen, werden mit einer Ausgleichsfahrt nach Polynom 5. Ordnung miteinander verbunden. Die Berechnungsergebnisse werden in einem ARRAY abgelegt.





Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	CALC_PROF_2_FORMAT_POLY			
Funktionsausführung -	boExec	boDone	-	Quittierung "Ausgeführt"
Endwert X- und Y-Achse -	diXYEndValue			
Format -	rFormat1			
Format -	rFormat2			
Umfang der Walze -	rPerimeter			
Punkt auf der X-Achse -	diPointX1			
Punkt auf der X-Achse -	diPointX2			
Tabellenstützpunkte -	arTab_A	arTab_A	-	Tabellenstützpunkte

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
diXYEndValue	DINT	Endwert der X- und Y-Achse der Tabelle
rFormat1	REAL	Formatlänge 1 [mm]
rFormat2	REAL	Formatlänge 2 [mm]
rPerimeter	REAL	Umfang der Walze [mm]

Name	Typ	Beschreibung
diPointX1	DINT	Punkt 1 auf der X-Achse [Inkremente] Dieser Punkt beschreibt das Ende der synchronen Bewegung und den Startpunkt der Ausgleichsbewegung
diPointX2	DINT	Punkt 2 auf der X-Achse [Inkremente] Dieser Punkt beschreibt das Ende der Ausgleichsbewegung und den Startpunkt der zweiten synchronen Bewegung

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
arTab_A	ARRAY	ARRAY[0..5] OF SMC_CAMXYVA Stützpunktarray (Hinterlegung der errechneten 6 Stützstellen)

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.5.2 CONVERT_ANALOG_TO_DIGITAL (FB)

Der Funktionsbaustein 'CONVERT_ANALOG_TO_DIGITAL' berechnet aus einem analogen Wert den zugehörigen Digitalwert. Die Umrechnung erfolgt linear. Die zugehörige Gerade wird aus den Eingabeparametern berechnet.

$y = m * x + n$ Geradengleichung

$m = \frac{\Delta y}{\Delta x} = \frac{y_1 - y_0}{x_1 - x_0}$ Steigung

$n = y_0 - m * x_0$ Versatz

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	CONVERT_ANALOG_TO_DIGITAL			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Analoger Eingangswert -	reAnalogInput	wDigitalOutput	-	Digitaler Ausgangswert
Digitaler Startwert -	wDigitalStartPoint			
Analoger Startwert -	reAnalogStartPoint			
Digitaler Endwert -	wDigitalEndPoint			
Analoger Endwert -	reAnalogEndPoint			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
reAnalogInput	REAL	Analoger Eingangswert, der in einen digitalen Wert umgerechnet werden soll
wDigitalStartPoint	WORD	Digitaler Wert des 1. Wertepaares zur Bestimmung der Umrechnungs-Geraden; y_0
reAnalogStartPoint	REAL	Analoger Wert des 1. Wertepaares zur Bestimmung der Umrechnungs-Geraden; x_0
wDigitalEndPoint	WORD	Digitaler Wert des 2. Wertepaares zur Bestimmung der Umrechnungs-Geraden; y_1
reAnalogEndPoint	REAL	Analoger Wert des 2. Wertepaares zur Bestimmung der Umrechnungs-Geraden; x_1

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
wDigitalOutput	WORD	Berechneter Ausgangswert boEnable=FALSE => wDigitalOutput=0

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.5.3 CONVERT_DIGITAL_TO_ANALOG (FB)

Der Funktionsbaustein 'CONVERT_DIGITAL_TO_ANALOG' berechnet aus einem digitalen Wert den zugehörigen Analogwert. Die Umrechnung erfolgt linear. Die zugehörige Gerade wird aus den Eingabeparametern berechnet.

$y = m * x + n$ Geradengleichung

$m = \frac{\Delta y}{\Delta x} = \frac{y_1 - y_0}{x_1 - x_0}$ Steigung

$n = y_0 - m * x_0$ Versatz

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	CONVERT_DIGITAL_TO_ANALOG			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Digitaler Eingangswert -	wDigitalInput	reAnalogOutput	-	Analoger Ausgangswert
Digitaler Startwert -	wDigitalStartPoint			
Analoger Startwert -	reAnalogStartPoint			
Digitaler Endwert -	wDigitalEndPoint			
Analoger Endwert -	reAnalogEndPoint			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
wDigitalInput	WORD	Digitaler Eingangswert, der in einen analogen Wert umgerechnet werden soll
wDigitalStartPoint	WORD	Digitaler Wert des 1. Wertepaares zur Bestimmung der Umrechnungs-Geraden; x_0
reAnalogStartPoint	REAL	Analoger Wert des 1. Wertepaares zur Bestimmung der Umrechnungs-Geraden; y_0
wDigitalEndPoint	WORD	Digitaler Wert des 2. Wertepaares zur Bestimmung der Umrechnungs-Geraden; x_1
reAnalogEndPoint	REAL	Analoger Wert des 2. Wertepaares zur Bestimmung der Umrechnungs-Geraden; y_1

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
reAnalogOutput	REAL	Berechneter Ausgangswert boEnable=FALSE => reAnalogOutput=0

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.5.4 CONVERT_VELOCITY (FB)

Der Funktionsbaustein 'CONVERT_VELOCITY' dient der Umrechnung von Geschwindigkeiten. Die Auswahl der Geschwindigkeitsumrechnung erfolgt über Modi.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	CONVERT_VELOCITY			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Auswahlmodus -	enMode	reOutVal	-	Ausgangswert
Eingangswert -	diInVal			
SERCOS Zykluszeit -	diID2_SERCOS_cycle			
Motorgeberauflösung -	diID116_Resol_mot_encod			
Weg einer Motordrehung -	diDistancePerMotorRevolution			

Eingangsvariablen

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>boEnable</td><td>BOOL</td><td>Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins.
Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet.
Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.</td></tr>
<tr><td>enMode</td><td>ENUM</td><td>EN_VELOCITY_CONVERT_MODE
Auswahlmodus
<table><tr><td>V_CONVERT_RPM_TO_INC_ID2</td><td>U/min => Inkr/ID2
Eingangsvariablen:
- diID2_SERCOS_cycle
- diID116_Resol_mot_encod
- diInVal</td></tr><tr><td>V_CONVERT_INC_ID2_TO_RPM</td><td>Inkr/ID2 => U/min
Eingangsvariablen:
- diID2_SERCOS_cycle
- diID116_Resol_mot_encod
- diInVal</td></tr><tr><td>V_CONVERT_U_MIN_TO_INC_S</td><td>U/min => Inkr/s
Eingangsvariablen:
- diID116_Resol_mot_encod
- diInVal</td></tr><tr><td>V_CONVERT_INC_S_TO_U_MIN</td><td>Inkr/s => U/min
Eingangsvariablen:
- diID116_Resol_mot_encod
- diInVal</td></tr><tr><td>V_CONVERT_MM_S_TO_U_MIN</td><td>mm/s => U/min
Eingangsvariablen:
- diDistancePerMotorRevolution
- diInVal</td></tr><tr><td>V_CONVERT_U_MIN_TO_MM_S</td><td>U/min => mm/s
Eingangsvariablen:
- diDistancePerMotorRevolution
- diInVal</td></tr></table></td></tr>
<tr><td>diInVal</td><td>DINT</td><td>Eingangswert</td></tr>
<tr><td>diID2_SERCOS_cycle</td><td>DINT</td><td>ID2 'SERCOS-Zykluszeit' [µs]</td></tr>
<tr><td>diID116_Resol_mot_encod</td><td>DINT</td><td>ID116 'Auflösung Motorgeber' [Inkremente]</td></tr>
<tr><td>diDistancePerMotor Revolution</td><td>DINT</td><td>Weg bei einer Motorumdrehung [mm]</td></tr>
</table>

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
reOutVal	REAL	Ausgangswert

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.5.5 fbrInertiaHollowCylinder (F)

Die Funktion 'fbrInertiaHollowCylinder' berechnet das Massenträgheitsmoment eines Hohlzylinders.

$J_{Hz} = J_a - J_i$ Trägheitsmoment Hohlzylinder

$J_{Hz} = \frac{1}{32} \pi h \rho (d_a^4 - d_i^4)$

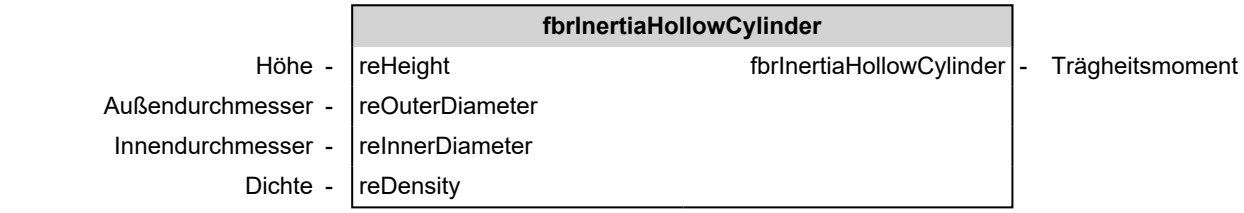
$J = \frac{1}{8} m * d^2$ Trägheitsmoment

$m = V * \rho$ Masse

$V = \frac{1}{4} \pi d^2 * h$ Volumen

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
reHeight	REAL	Höhe h [m]
reOuterDiameter	REAL	Außendurchmesser d_a [m]
reInnerDiameter	REAL	Innendurchmesser d_i [m]
reDensity	REAL	Spez. Dichte ρ [kg/m^3]

Ausgangsvariable

Name	Typ	Beschreibung
fbrInertiaHollowCylinder	REAL	Trägheitsmoment J_{Hz} [kg x m^2]

4.2.1.1.5.6 fbrInertiaPlainCylinder (F)

Die Funktion 'fbrInertiaPlainCylinder' berechnet das Massenträgheitsmoment eines Vollzylinders.

$J_{Vz} = \frac{1}{32} \pi h \rho d^4$ Trägheitsmoment Vollzylinder

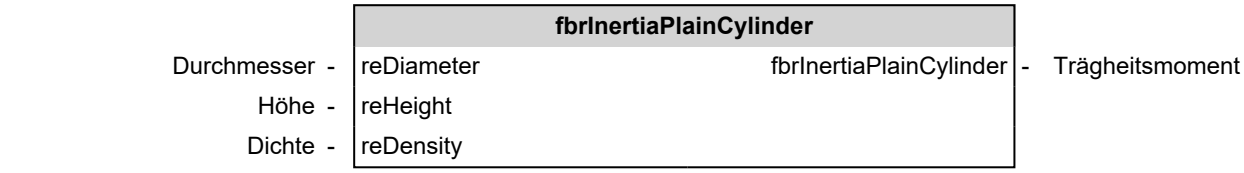
$J = \frac{1}{8} m * d^2$ Trägheitsmoment

$m = V * \rho$ Masse

$V = \frac{1}{4} \pi d^2 * h$ Volumen

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
reDiameter	REAL	Durchmesser d [m]
reHeight	REAL	Höhe h [m]
reDensity	REAL	Spez. Dichte ρ [kg/m^3]

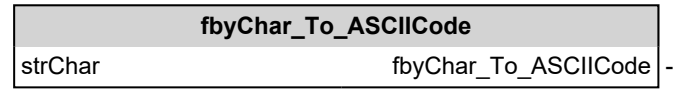
Ausgangsvariable

Name	Typ	Beschreibung
fbrInertiaPlainCylinder	REAL	Trägheitsmoment J [kg x m^2]

4.2.1.1.5.7 fbyChar_To_ASCIICode (F)

Die Funktion 'fbyChar_To_ASCIICode' erzeugt für ein Zeichen den entsprechenden ASCII-Code.

Anwender Interface



Eingangsvariable

Name	Typ	Beschreibung
strChar	STRING (1)	String; z.B. 'a'

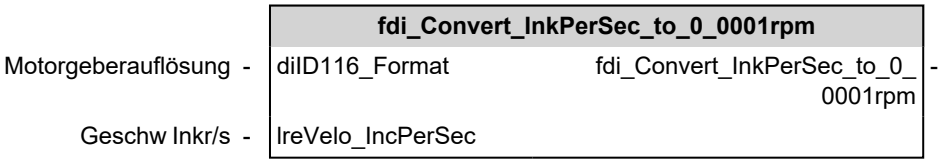
Ausgangsvariable

Name	Typ	Beschreibung
fbyChar_To_ASCIICode	BYTE	ASCII-Code, z.B. 97

4.2.1.1.5.8 fdi_Convert_InkPerSec_to_0_0001rpm (F)

Die Funktion 'fdi_Convert_InkPerSec_to_0_0001rpm' rechnet einen Geschwindigkeitswert von [Inkremente/s] in [0,0001 Umdrehungen/m] um.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
diID116_Format	DINT	ID116 'Auflösung Motorgeber' [Inkremente]
lreVelo_IncPerSec	LREAL	Geschwindigkeitswert, der umgerechnet werden soll [Inkremente/s]

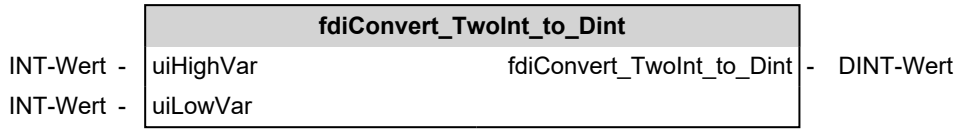
Ausgangsvariable

Name	Typ	Beschreibung
fdi_Convert_InkPerSec_to_0_0001rpm	DINT	Umgerechnete Geschwindigkeit [0,0001 U/min]

4.2.1.1.5.9 fdiConvert_TwoInt_to_Dint (F)

Die Funktion 'fdiConvert_TwoInt_to_Dint' verknüpft zwei INT-Werte zu einem DINT-Wert.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
uiHighVar	INT	INT-Wert
uiLowVar	INT	INT-Wert

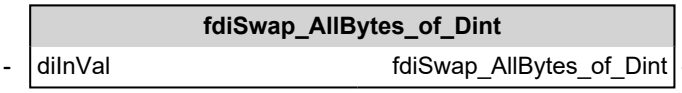
Ausgangsvariable

Name	Typ	Beschreibung
fdiConvert_TwoInt_to_Dint	DINT	DINT-Wert: <uiHighVar><uiLowVar>

4.2.1.1.5.10 fdiSwap_AllBytes_of_Dint (F)

Die Funktion 'fdiSwap_AllBytes_of_Dint' kehrt die Reihenfolge aller Bytes eines DINT-Wertes um.

Anwender Interface



Eingangsvariable

Name	Typ	Beschreibung
diInVal	DINT	Byte3 | Byte2 | Byte1 | Byte0 HighWord | LowWord

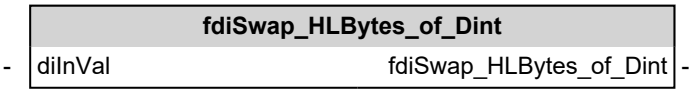
Ausgangsvariable

Name	Typ	Beschreibung
fdiSwap_AllBytes_of_Dint	DINT	Byte0 | Byte1 | Byte2 | Byte3 HighWord | LowWord

4.2.1.1.5.11 fdiSwap_HLBytes_of_Dint (F)

Die Funktion 'fdiSwap_HLBytes_of_Dint' tauscht High- und Low-Byte innerhalb der Wörter eines DINT-Wertes

Anwender Interface



Eingangsvariable

Name	Typ	Beschreibung
diInVal	DINT	HighByte | LowByte | HighByte | LowByte HighWord | LowWord

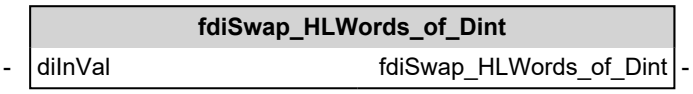
Ausgangsvariable

Name	Typ	Beschreibung
fdiSwap_HLBytes_of_Dint	DINT	LowByte | HighByte | LowByte | HighByte HighWord | LowWord

4.2.1.1.5.12 fdiSwap_HLWords_of_Dint (F)

Die Funktion 'fdiSwap_HLWords_of_Dint' tauscht die zwei Worte eines DINT-Wertes

Anwender Interface



Eingangsvariable

Name	Typ	Beschreibung
diInVal	DINT	HighWord | LowWord

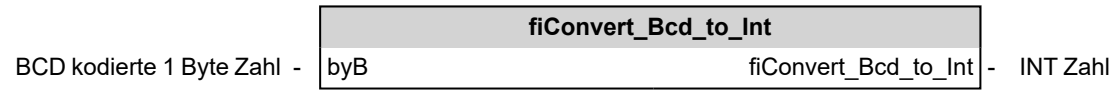
Ausgangsvariable

Name	Typ	Beschreibung
fdiSwap_HLWords_of_Dint	DINT	LowWord | HighWord

4.2.1.1.5.13 fiConvert_Bcd_to_Int (F)

Die Funktion 'fiConvert_Bcd_to_Int' wandelt eine BCD-kodierte 1-Byte Zahl in eine INT-Zahl.

Anwender Interface



Eingangsvariable

Name	Typ	Beschreibung
byB	BYTE	BCD-kodierte 1-Byte Zahl

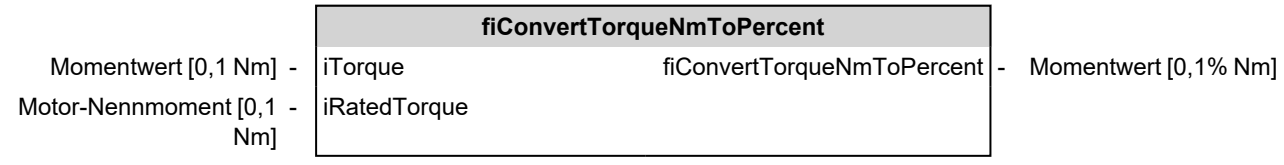
Ausgangsvariable

Name	Typ	Beschreibung
fiConvert_Bcd_to_Int	INT	Kein BCD-Code: fiConvert_Bcd_to_Int = -1 sonst: fiConvert_Bcd_to_Int = 0...10

4.2.1.1.5.14 fiConvertTorqueNmToPercent (F)

Die Funktion 'fiConvertTorqueNmToPercent' wandelt ein Momentwert [0,1 Nm] in einen prozentualen Ausgabewert des Motornennmoments [% Mn].

Anwender Interface



Eingangsvariable

Name	Typ	Beschreibung
iTorque	INT	Momentwert [0,1 Nm]
iRatedTorque	INT	Motor Nennmoment [0,1 Nm]

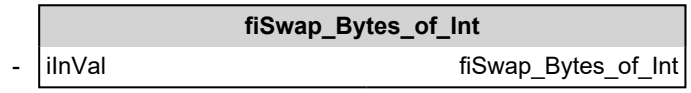
Ausgangsvariable

Name	Typ	Beschreibung
fiConvertTorqueNmToPercent	INT	Momentwert [0,1% Nm]

4.2.1.1.5.15 fiSwap_Bytes_of_Int (F)

Die Funktion 'fiSwap_Bytes_of_Int' tauscht die beiden Bytes eines INT-Wertes

Anwender Interface



Eingangsvariable

Name	Typ	Beschreibung
iInVal	INT	HighByte | LowByte

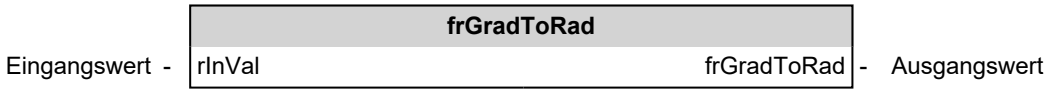
Ausgangsvariable

Name	Typ	Beschreibung
fiSwap_Bytes_of_Int	INT	LowByte | HighByte

4.2.1.1.5.16 frGradToRad (F)

Die Funktion 'frGradToRad' rechnet den Grad-Wert eines Winkels in einen Radiant-Wert um.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
rInVal	REAL	Eingangswert [°]

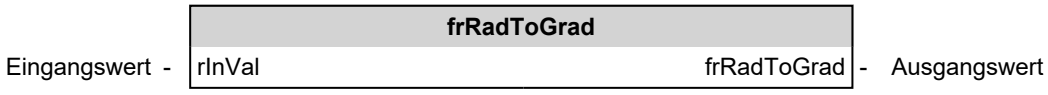
Ausgangsvariable

Name	Typ	Beschreibung
frGradToRad	REAL	Ausgangswert [rad]

4.2.1.1.5.17 frRadToGrad (F)

Die Funktion 'frRadToGrad' rechnet den Radiant-Wert eines Winkels in einen Grad-Wert um.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
rInVal	REAL	Eingangswert [rad]

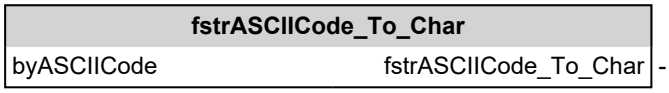
Ausgangsvariable

Name	Typ	Beschreibung
frRadToGrad	REAL	Ausgangswert [°]

4.2.1.1.5.18 fstrASCIICode_To_Char (F)

Die Funktion 'fstrASCIICode_To_Char' erzeugt aus einem ASCII-Code das entsprechende Zeichen

Anwender Interface



Eingangsvariable

Name	Typ	Beschreibung
byASCIICode	BYTE	ASCII-Code, z.B. 097

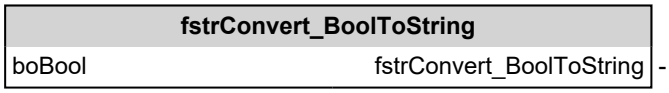
Ausgangsvariable

Name	Typ	Beschreibung
fstrASCIICode_To_Char	STRING (1)	String, z.B. 'a'

4.2.1.1.5.19 fstrConvert_BoolToString (F)

Die Funktion 'fstrConvert_BoolToString' wandelt eine Boolsche Variable in einen String um.

Anwender Interface



Eingangsvariable

Name	Typ	Beschreibung
boBool	BOOL	z.B. TRUE

Ausgangsvariable

Name	Typ	Beschreibung
fstrConvert_BoolToString	STRING (1)	z.B. '1'

4.2.1.1.5.20 fstrConvert_DintToString (F)

Die Funktion 'fstrConvert_DintToString' wandelt eine DINT-Variable in einen String um.

Anwender Interface

Eingangsvariable

Name	Typ	Beschreibung
diDint	DINT	z.B. 12345

Ausgangsvariable

Name	Typ	Beschreibung
fstrConvert_DintToString	STRING	z.B. '12345'

4.2.1.1.5.21 fudCalc_Ramp_Acceleration (F)

Die Funktion 'fudCalc_Ramp_Acceleration' berechnet die Beschleunigungsrampe.

$$a\left[\frac{Inkr}{s^2}\right] = \frac{v\left[\frac{Inkr}{s}\right]}{t\,[s]}$$ Beschleunigung

$$v\left[\frac{Inkr}{s}\right] = \frac{udVelocity\left[\frac{U}{min}\right]}{10000 * 60\left[\frac{s}{min}\right]} * ID116\left[\frac{Inkr}{U}\right]$$ Geschwindigkeit

$$t\,[s] = \frac{udAt\_ms\,[ms]}{1000\left[\frac{ms}{s}\right]}$$ Beschleunigungszeit

Anwender Interface

	fudCalc_Ramp_Acceleration			
Beschleunigungszeit -	udAt_ms	fudCalc_Ramp_Acceleration	-	Beschleunigung
Maximalgeschwindigkeit -	udVelocity			
Motorgeberauflösung -	diID116			

Eingangsvariablen

Name	Typ	Beschreibung
udAt_ms	UDINT	Beschleunigungszeit [ms]
udVelocity	UDINT	Maximale Geschwindigkeit [0,0001 U/min]
diID116	DINT	ID116 'Auflösung Motorgeber' [Inkremente]

Ausgangsvariable

Name	Typ	Beschreibung
fudCalc_Ramp_Acceleration	UDINT	Beschleunigung [Inkremente/s^2]

4.2.1.1.5.22 fudiConvert_Bcd_to_Udint (F)

Die Funktion 'fudiConvert_Bcd_to_Udint' wandelt eine 8-stellige BCD-kodierte positive Zahl in einen UDINT-Wert.

Anwender Interface

	fudiConvert_Bcd_to_Udint		
-	udiInVal	fudiConvert_Bcd_to_Udint	-

Eingangsvariable

Name	Typ	Beschreibung
udiInVal	UDINT	8-stellige BCD-kodierte positive Zahl

Ausgangsvariable

Name	Typ	Beschreibung
fudiConvert_Bcd_to_Udint	UDINT	Kein BCD-Code: fudiConvert_Bcd_to_Udint = -1 sonst: fudiConvert_Bcd_to_Udint = binärer Wert

4.2.1.1.5.23 fuiConvert_Bcd_to_Uint (F)

Die Funktion 'fuiConvert_Bcd_to_Uint' wandelt eine 8-stellige BCD-kodierte positive Zahl in einen UINT-Wert.

Anwender Interface

fuiConvert_Bcd_to_Uint		
- uiInVal	fuiConvert_Bcd_to_Uint	-

Eingangsvariable

Name	Typ	Beschreibung
uiInVal	UINT	8-stellige BCD-kodierte positive Zahl

Ausgangsvariable

Name	Typ	Beschreibung
fuiConvert_Bcd_to_Uint	UINT	Kein BCD-Code: fuiConvert_Bcd_to_Uint = -1 sonst: fuiConvert_Bcd_to_Uint = binärer Wert

4.2.1.1.5.24 udfConvert_Acceleration (F)

Die Funktion 'udfConvert_Acceleration' wird vollständig durch 'fudCalc_Ramp_Acceleration' ersetzt.
Für neue Anwendungen ist ausschließlich 'fudCalc_Ramp_Acceleration' zu verwenden!

Die Funktion 'udfConvert_Acceleration' berechnet die Beschleunigungsrampe.

$$a\left[\frac{Inkr}{s^2}\right] = \frac{v\left[\frac{Inkr}{s}\right]}{t\,[s]}$$ Beschleunigung

$$v\left[\frac{Inkr}{s}\right] = \frac{udVelocity\left[\frac{U}{min}\right]}{10000 * 60\left[\frac{s}{min}\right]} * ID116\left[\frac{Inkr}{U}\right]$$ Geschwindigkeit

$$t\,[s] = \frac{udAt\_01s\,[s]}{10}$$ Beschleunigungszeit

Anwender Interface

	udfConvert_Acceleration			
Beschleunigungszeit -	iAt_01s	udfConvert_Acceleration	-	Beschleunigung
Maximalgeschwindigkeit -	udVelocity			
Motorgeberauflösung -	diID116			

Eingangsvariablen

Name	Typ	Beschreibung
iAt_01s	INT	Beschleunigungszeit [0,1 s]
diVelocity	DINT	Maximale Geschwindigkeit [0,0001 U/min]
diID116	DINT	ID116 'Auflösung Motorgeber' [Inkremente]

Ausgangsvariable

Name	Typ	Beschreibung
udfConvert_Acceleration	UDINT	Beschleunigung [Inkremente/s^2]

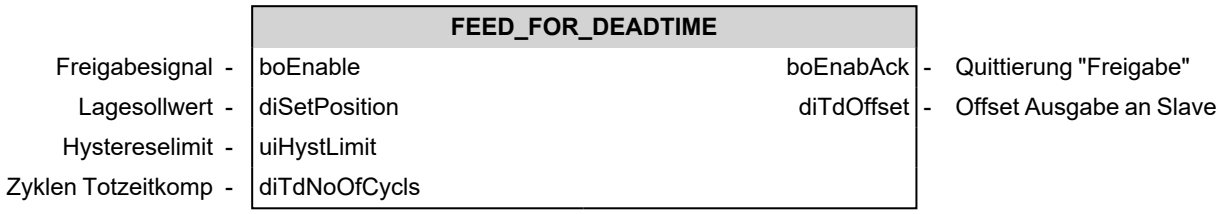
4.2.1.1.6 08_Feedforward

4.2.1.1.6.1 FEED_FOR_DEADTIME (FB)

Der Funktionsbaustein 'FEED_FOR_DEADTIME' realisiert eine Totzeitkompensation. Der Baustein misst, mit welcher Geschwindigkeit sich der Lagesollwert ändert, und ermittelt in Abhängigkeit der Totzeit die Verschiebung des Slaves.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Format	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
uiHystLimit	UINT	Lagesollwert, der zum KW ausgegeben wird (Baustein SET_SETPOINT_ POSITION)
diTdNoOfCycls	DINT	Anzahl der Zyklen ID2 für die Totzeitkompensation [0,1 * ID2]

Ausgangsvariablen

Name	Format	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diTdOffset	DINT	Offset für Ausgabe an Slave [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FEED_FOR_DEADTIME

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

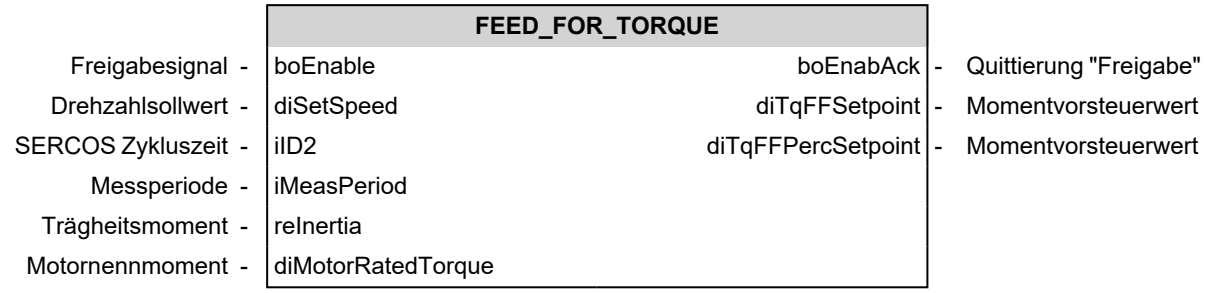
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.6.2 FEED_FOR_TORQUE (FB)

Der Funktionsbaustein 'FEED_FORWARD_TORQUE' realisiert eine Momentvorsteuerung. Der Baustein misst die Geschwindigkeit, mit der sich der Drehzahlsollwert ändert, und ermittelt das entsprechende Moment.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diSetSpeed	DINT	Geschwindigkeitssollwert [0,0001 U/min] Baustein SET_SETPOINT_SPEED
iID2	INT	ID2 'SERCOS-Zykluszeit' [µs]
iMeasPeriod	INT	Periode der Geschwindigkeitsmessung [Vielfache von ID2]
reInertia	REAL	Ermitteltes Trägheitsmoment [kg x m^2]
diMotorRatedTorque	DINT	Motornennmoment Mn [0,1 Nm]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diTqFFSetpoint	DINT	Momentvorsteuerungswert; muss mit Baustein 'SET_PRE_SETPOINT_ TORQUE' an den Antrieb ausgegeben werden. [0,1 Nm]
diTqFFPercSetpoint	DINT	Momentvorsteuerungswert; muss mit Baustein 'SET_PRE_SETPOINT_ TORQUE' an den Antrieb ausgegeben werden. [0,1% Mn]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FEED_FOR_TORQUE

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

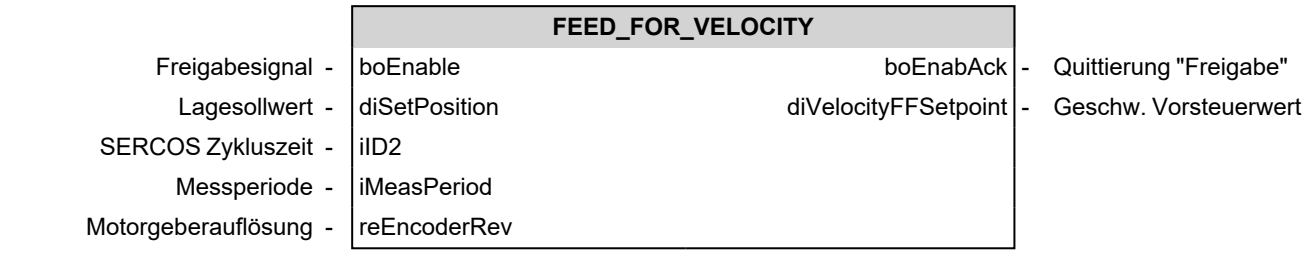
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.6.3 FEED_FOR_VELOCITY (FB)

Der Funktionsbaustein 'FEED_FOR_VELOCITY' realisiert eine Drehzahlvorsteuerung. Der Baustein ermittelt, mit welcher Geschwindigkeit sich der Lagesollwert ändert, und berechnet die entsprechende Motordrehzahl.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente] Baustein SET_SETPOINT_POSITION
iID2	INT	ID2 'SERCOS-Zykluszeit' [µs]
iMeasPeriod	INT	Periode der Geschwindigkeitsmessung [Vielfache von ID2]
reEncoderRev	REAL	Motorgeberauflösung

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diVelocityFFSetpoint	DINT	Drehzahlvorsteuerungswert; muss mit Baustein 'SET_PRE_SETPOINT_SPEED' an den Antrieb ausgegeben werden.

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	FEED_FOR_VELOCITY

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.7 09_Filter

4.2.1.1.7.1 FILTER_BLOCKING_NEGATIVE_COUNT (FB)

Der Funktionsbaustein 'FILTER_BLOCKING_NEGATIVE_COUNT' sperrt negative Zählimpulse.

Dabei wird am Filterausgang erst wieder positiv gezählt, wenn der negative „Zählweg“ z.B. eines Rechteckgebers abgebaut ist. Vorteil: Eine Lageregelung um eine Position (z.B. ±1 Inkrement) führt nicht zum Hochzählen des Filterausgangs. Somit bleibt der Bezug zum Eingang (z.B. Masterrechteckgeber) erhalten.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	FILTER_BLOCKING_NEGATIVE_COUNT		
Freigabesignal -	boEnable	boEnabAck	- Quittierung "Freigabe"
Eingangswert -	diInVal	diOutVal	- Ausgangswert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Eingangswert [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diOutVal	DINT	Ausgangswert [Inkremente]

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.7.2 FILTER_BLOCKING_POSITIVE_COUNT (FB)

Der Funktionsbaustein 'FILTER_BLOCKING_POSITIVE_COUNT' sperrt positive Zählimpulse. Dabei wird am Filterausgang erst wieder negativ gezählt, wenn der positive „Zählweg“ z.B. eines Rechteckgebers abgebaut ist. Vorteil: Eine Lageregelung um eine Position (z.B. +/- 1 Inkrement) führt nicht zum Zählen des Filterausgangs. Somit bleibt der Bezug zum Eingang (z.B. Masterrechteckgeber) erhalten.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	FILTER_BLOCKING_POSITIVE_COUNT		
Freigabesignal -	boEnable	boEnabAck	- Quittierung "Freigabe"
Eingangswert -	diInVal	diOutVal	- Ausgangswert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Eingangswert [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diOutVal	DINT	Ausgangswert [Inkremente]

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.7.3 FILTER_HYSTERESIS (FB)

Der Funktionsbaustein 'FILTER_HYSTERESIS' realisiert eine Hysterese.

Der Baustein steuert die Ausgangsgröße in Abhängigkeit der Änderung der Eingangsgröße und der Grenzen .

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	FILTER_HYSTERESIS			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert -	diInVal	diOutVal	-	Ausgangswert
Positive Grenze -	uiLimitPos			
Negative Grenze -	uiLimitNeg			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Eingangswert
uiLimitPos	UINT	Positive Grenze der Hysterese
uiLimitNeg	UINT	Negative Grenze der Hysterese

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben

Name	Typ	Beschreibung
diOutVal	DINT	Ausgangswert mit Hysterese

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.7.4 FILTER_MEAN_VALUE_DINT_MAX20 (FB)

Der Funktionsbaustein 'FILTER_MEAN_VALUE_DINT_MAX20' bildet über die Eingangswerte den arithmetischen Mittelwert mittels Schieberegister über maximal 20 Werte.

Bei der Mittelwertbildung wird der neue Eingangswert in die Variable 'diInVal' geschrieben. Nach Aufruf des Bausteins kann der neue Mittelwert in der Variablen 'diMeanVal' ausgelesen werden.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	FILTER_MEAN_VALUE_DINT_MAX20			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert -	diInVal	diMeanVal	-	Mittelwert
Anzahl Werte -	iNoOfValues			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Eingangswert
iNoOfValues	INT	Anzahl von Werten, über die der Mittelwert gebildet wird. 1 ≤ iNoOfValues ≤ 20

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diMeanVal	DINT	Arithmetischer Mittelwert

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.7.5 FILTER_MEAN_VALUE_MAX20 (FB)

Der Funktionsbaustein 'FILTER_MEAN_VALUE_MAX20' bildet über die Eingangswerte den arithmetischen Mittelwert mittels Schieberegister über maximal 20 Werte.

Bei der Mittelwertbildung wird der neue Eingangswert in die Variable 'iInVal' geschrieben. Nach Aufruf des Bausteins kann der neue Mittelwert in der Variablen 'iMeanVal' ausgelesen werden.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	FILTER_MEAN_VALUE_MAX20			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert -	iInVal	iMeanVal	-	Mittelwert
Anzahl Werte -	iNoOfValues			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
iInVal	INT	Eingangswert
iNoOfValues	INT	Anzahl von Werten, über die der Mittelwert gebildet wird. 1 ≤ iNoOfValues ≤ 20

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
iMeanVal	INT	Arithmetischer Mittelwert

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.7.6 FILTER_PT1 (FB)

Der Funktionsbaustein 'FILTER_PT1' realisiert ein Tiefpass-Filter 1. Ordnung

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	FILTER_PT1			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert -	diInVal	diOutVal	-	Ausgangswert
Filterzeit -	diTime			
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Eingangswert
diTime	DINT	Filterzeitkonstante [ms]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diOutVal	DINT	Gefilterter Ausgangswert

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.7.7 FILTER_RAMP_INCRDECR_TIMECONST (FB)

Der Funktionsbaustein 'FILTER_RAMP_INCRDECR_TIMECONST' liefert bei sprungartig verlaufender Eingangsgröße einen gerampten Ausgangswert.

- Der Baustein begrenzt den Ausgangswert 'diOutVal' ≤ 'diInVal'.
- Ist das Eingangssignal 'boEnable' = FALSE, wird der Eingangswert ohne Rampe an den Ausgang kopiert.
- 'diIncreasPos' und 'diIncreasNeg' legen unabhängig voneinander die Steilheiten der Rampe beim Beschleunigen bzw. Verzögern fest.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

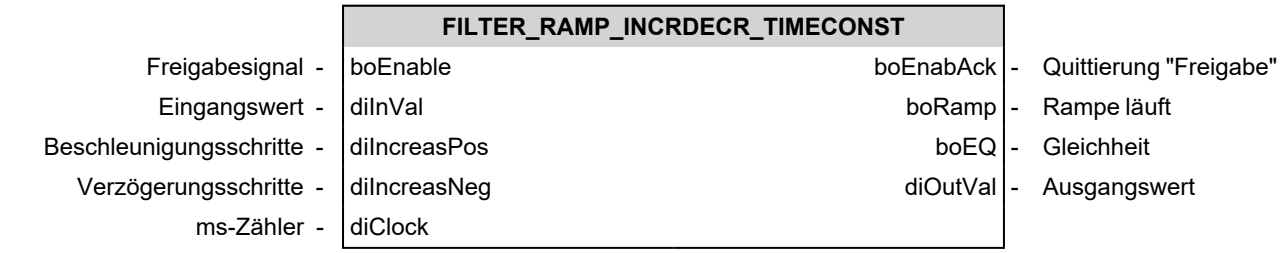
Beispiel:

Sprung der Eingangsgröße „diInVal“ von 0 -> 1000

Rampenanstieg für Beschleunigung 'diIncreasPos' = 100 [1/s]

Daraus folgt: Der Ausgangswert 'diOutVal' rampt innerhalb von 10 s von 0 -> 1000

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Rampenfreigabesignal: Mit einer positiven Flanke startet die Rampenfunktion des Bausteins. Solange 'boEnable' = TRUE ist, bleibt der Baustein freigegeben und der Ausgang wird gerampt. Im Zustand 'boEnable' = FALSE wird dem Baustein die Rampenfreigabe entzogen und der Eingangswert wird direkt auf den Ausgangswert kopiert
diInVal	DINT	Eingangswert
diIncreasPos	DINT	Beschleunigungsschritte: Anzahl der Schritte, um die sich der Ausgangswert pro Sekunde erhöht [1/s]
diIncreasNeg	DINT	Verzögerungsschritte: Anzahl der Schritte, um die sich der Ausgangswert pro Sekunde verringert [1/s]
diClock	DINT	ms-Zähler [ms]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Rampenfunktion ist aktiv
boRamp	BOOL	Rampe wird durchlaufen
boEQ	BOOL	Eingangswert = Ausgangswert
diOutVal	DINT	Ausgangswert

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

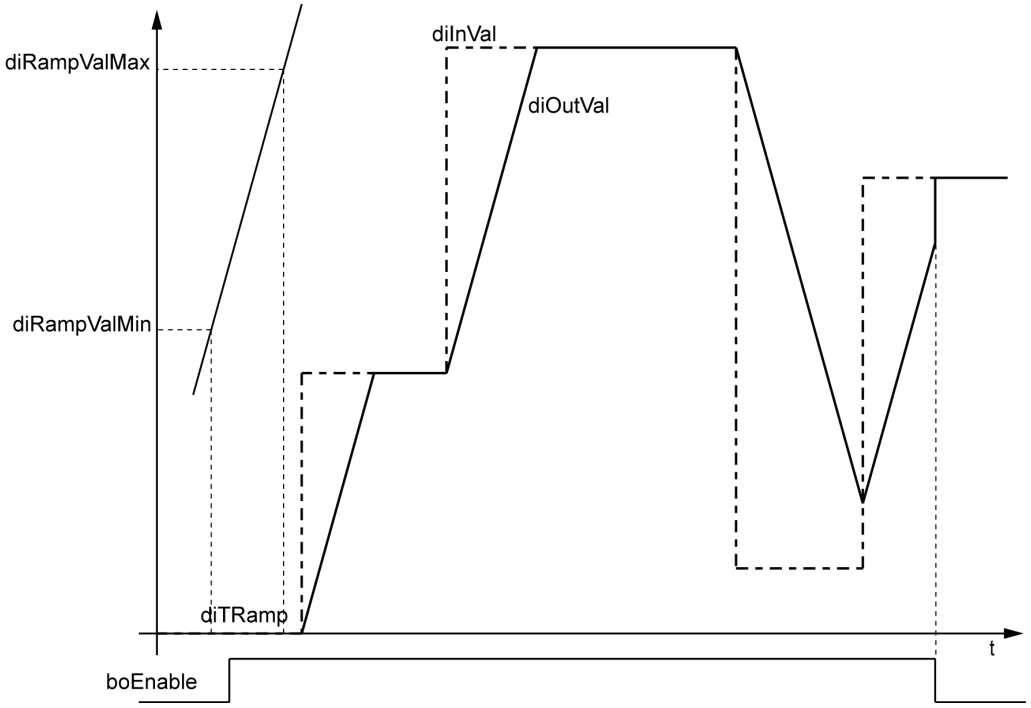
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.7.8 FILTER_RAMP_MINMAX_TIME (FB)

Der Funktionsbaustein 'FILTER_RAMP_MINMAX_TIME' liefert bei sprungartig verlaufender Eingangsgröße einen gerampten Ausgangswert.

- Der Baustein begrenzt den Ausgangswert 'diOutVal' ≤ 'diInVal'.
- Ist das Eingangssignal 'boEnable' = FALSE, wird der Eingangswert ohne Rampe an den Ausgang kopiert.
- Die Rampenzeit legt die Zeit fest, in der der Ausgangswert bei Sprung der Eingangsgröße von 'diRampValMin' auf 'diRampValMax' hochgefahren wird.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.



Anwender Interface

FILTER_RAMP_MINMAX_TIME				
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert -	diInVal	boRamp	-	Rampe läuft
Min-Wert Rampe -	diRampValMin	boEQ	-	Gleichheit
Max-Wert Rampe -	diRampValMax	diOutVal	-	Ausgangswert
Rampenzeit -	diTRamp			
ms-Zähler -	diClock			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Rampenfreigabesignal: Mit einer positiven Flanke startet die Rampenfunktion des Bausteins. Solange 'boEnable' = TRUE ist, bleibt der Baustein freigegeben und der Ausgang wird gerampt. Im Zustand 'boEnable' = FALSE wird dem Baustein die Rampenfreigabe entzogen und der Eingangswert wird direkt auf den Ausgangswert kopiert
diInVal	DINT	Eingangswert
diRampValMin	DINT	Minimalwert der Rampe

Name	Typ	Beschreibung
diRampValMax	DINT	Maximalwert der Rampe
diTRamp	DINT	Rampenzeit [ms]
diClock	DINT	ms-Zähler [ms]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Rampenfunktion ist aktiv
boRamp	BOOL	Rampe wird durchlaufen
boEQ	BOOL	Eingangswert = Ausgangswert
diOutVal	DINT	Ausgangswert

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.7.9 FILTER_SETPOINT (FB)

Der Funktionsbaustein 'FILTER_SETPOINT' realisiert ein Sollwertfilter

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

		FILTER_SETPOINT			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Filterzeit	-	lreFilterTime	diOutVal	-	Ausgangswert
Eingangswert	-	diPosition	lreVelocity	-	Aktuelle Geschwindigkeit
Modulo Eingang	-	diModuloIn	lreAccel	-	Aktuelle Beschleunigung
Modulo Ausgang	-	diModuloOut			
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
lreFilterTime	LREAL	Filterzeit [s]
diPosition	DINT	Eingangswert [Inkremente]
diModuloIn	DINT	Modulo für den Eingangswert
diModuloOut	DINT	Modulo für den Ausgangswert

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben

Name	Typ	Beschreibung
diOutVal	DINT	Ausgangswert [Inkremente]
lreVelocity	LREAL	Aktuelle Geschwindigkeit [Inkremente/s]
lreAccel	LREAL	Aktueller Beschleunigungswert [Inkremente/s^2]

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.8 10_Buffer

4.2.1.1.8.1 BUFFER_FIFO_DINT_MAX_50 (FB)

Der Funktionsbaustein 'BUFFER_FIFO_DINT_MAX_50' stellt einen FIFO-Speicher für Doppel-Integer-Werte zur Verfügung.
Es gibt drei Aktionen:

- actInPut - Einspeichern eines Wertes in den FIFO
- actOutPut - Ausspeichern eines Wertes aus dem FIFO
- actReset - Rücksetzen des FIFO

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	BUFFER_FIFO_DINT_MAX_50			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert -	diInVal	boEmpty	-	Puffer leer
		boOverflow	-	Pufferüberlauf
		diOutVal	-	Ausgangswert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Einspeicherwert

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boEmpty	BOOL	Der FIFO-Puffer ist leer

Name	Typ	Beschreibung
boOverflow	BOOL	Pufferüberlauf: Ein neuer Wert soll in den Puffer gespeichert werden, es ist aber kein freier Speicherplatz vorhanden.
diOutVal	DINT	Wert, der beim letzten Aufruf der Aktion 'actOutPut' ausgelesen wurde

Aktionen

Name	Beschreibung
actInput	Aktion speichert den Wert aus 'diInVal' in den FIFO ein <BUFFER_FIFO_DINT_MAX_50>.actInput(diInVal:=...);
actOutPut	Aktion liest den nächsten Wert aus dem FIFO aus und schreibt den Wert nach 'diOutVal' <BUFFER_FIFO_DINT_MAX_50>.actOutPut(diOutVal=>...);
actReset	Aktion setzt den FIFO zurück, alle gespeicherten Werte gehen verloren <BUFFER_FIFO_DINT_MAX_50>.actReset();

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.8.2 BUFFER_FIFO_INT_MAX50 (FB)

Der Funktionsbaustein 'BUFFER_FIFO_INT_MAX50' stellt einen FIFO-Speicher für Integer-Werte zur Verfügung.

Es gibt drei Aktionen:

- actInPut - Einspeichern eines Wertes in den FIFO
- actOutPut - Ausspeichern eines Wertes aus dem FIFO
- actReset - Rücksetzen des FIFO

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	BUFFER_FIFO_INT_MAX50			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert -	iInVal	boEmpty	-	Puffer leer
		boOverflow	-	Pufferüberlauf
		iOutVal	-	Ausgangswert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
iInVal	INT	Einspeicherwert

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boEmpty	BOOL	Der FIFO-Puffer ist leer

Name	Typ	Beschreibung
boOverflow	BOOL	Pufferüberlauf: Ein neuer Wert soll in den Puffer gespeichert werden, es ist aber kein freier Speicherplatz vorhanden.
iOutVal	INT	Wert, der beim letzten Aufruf der Aktion 'actOutPut' ausgelesen wurde

Aktionen

Name	Beschreibung
actInput	Aktion speichert den Wert aus 'diInVal' in den FIFO ein <BUFFER_FIFO_INT_MAX50>.actInput(diInVal:=...);
actOutPut	Aktion liest den nächsten Wert aus dem FIFO aus und schreibt den Wert nach 'diOutVal' <BUFFER_FIFO_INT_MAX50>.actOutPut(diOutVal=>...);
actReset	Aktion setzt den FIFO zurück, alle gespeicherten Werte gehen verloren <BUFFER_FIFO_INT_MAX50>.actReset();

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.8.3 BUFFER_SHIFT_REG_INT_MAX10 (FB)

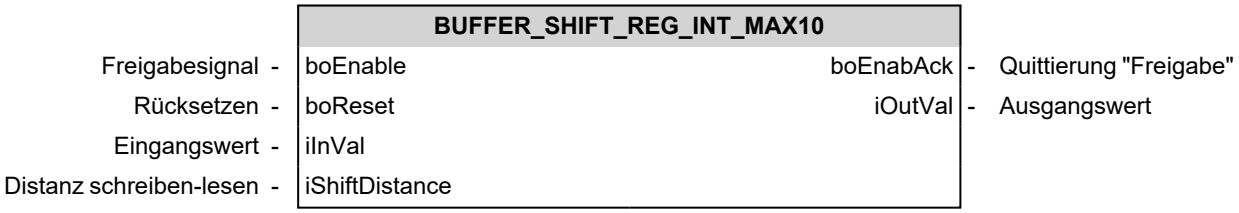
Der Funktionsbaustein 'BUFFER_SHIFT_REG_INT_MAX10' stellt ein Schieberegister mit 10 Plätzen zu Verfügung.

Es gibt zwei Aktionen:

- actClock - Weiterschieben des Schieberegisters um einen Platz
- actReset - Rücksetzen des Schieberegisters

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boReset	BOOL	Schieberegister zurücksetzen
iInVal	INT	Einspeicherwert
iShiftDistance	INT	Anzahl der Takte, bis der Einspeicherwert am Ausgang erscheint

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
iOutVal	INT	Nach der Aktion 'actClock' hat die Variable den Wert des Schieberegisters an der Auslesepositon

Aktionen

Name	Beschreibung
actClock	Die Aktion speichert den Wert von 'iInVal' in das Schieberegister ein, verschiebt das Register und setzt die Ausgangsvariable 'diOutVal' <BUFFER_SHIFT_REG_INT_MAX10>.actClock();
actReset	Aktion leert das gesamte Schieberegister <BUFFER_SHIFT_REG_INT_MAX10>.actReset();

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.9 11_Modulo

4.2.1.1.9.1 fdiAddModulo (F)

Die Funktion 'fdiAddModulo' addiert zwei Werte mit Modulo.

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	fdiAddModulo		
Summand 1 -	diSummand1	fdiAddModulo	- Ausgangswert
Summand 2 -	diSummand2		
Modulowert -	diModulo		

Eingangsvariablen

Name	Typ	Beschreibung
diSummand1	DINT	Summand 1 [Inkremente]
diSummand2	DINT	Summand 2 [Inkremente]
diModulo	DINT	Modulowert [Inkremente]

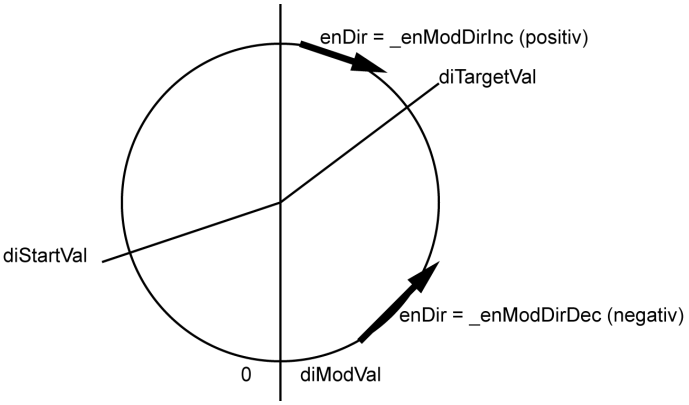
Ausgangsvariable

Name	Typ	Beschreibung
fdiAddModulo	DINT	Ausgangswert [Inkremente]

4.2.1.1.9.2 fdiModulo_DiffModulo (F)

Die Funktion 'fdiModulo_DiffModulo' liefert die Modulo-Differenz zwischen zwei Werten.

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.



Anwender Interface

	fdiModulo_DiffModulo			
Startwert -	diStartVal	fdiModulo_DiffModulo	-	Differenz
Zielwert -	diTargetVal			
Modulo-Wert -	diModVal			
Richtung Differenz -	enDir			
Richtungsnegation -	boDirNeg			

Eingangsvariablen

Name	Typ	Beschreibung
diStartVal	DINT	Start-Wert der Differenz; 0 < diStartVal < diModVal [Inkremente]
diTargetVal	DINT	Ziel-Wert der Differenz; 0 < diTargetVal < diModVal [Inkremente]
diModVal	DINT	Modulo Endwert des Modulozählers [Inkremente] diModVal > 0
enDir	ENUM	EN_FILTER_MODULO_DIR Richtung der Differenzbildung: _enModDirShortDist: kürzester Weg _enModDirInc: Differenz in Richtung aufsteigender Werte _enModDirDec: Differenz in Richtung fallender Werte
boDirNeg	BOOL	Berechnung der Differenz in negative Richtung

Ausgangsvariable

Name	Typ	Beschreibung
fdiModulo_DiffModulo	DINT	Differenz zwischen Start- und Zielwert [Inkremente]

4.2.1.1.9.3 fdiModulo_Modulo (F)

Die Funktion 'fdiModulo_Modulo' berechnet den Modulo-Wert des fortlaufenden Eingangswertes.

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	fdiModulo_Modulo			
Eingangswert -	diInVal	fdiModulo_Modulo	-	Modulo-Wert
Modulo Endwert des Modulozählers -	diModVal			

Eingangsvariablen

Name	Typ	Beschreibung
diInVal	DINT	Eingangswert [Inkremente]
diModVal	DINT	Modulo Endwert des Modulozählers [Inkremente]

Ausgangsvariable

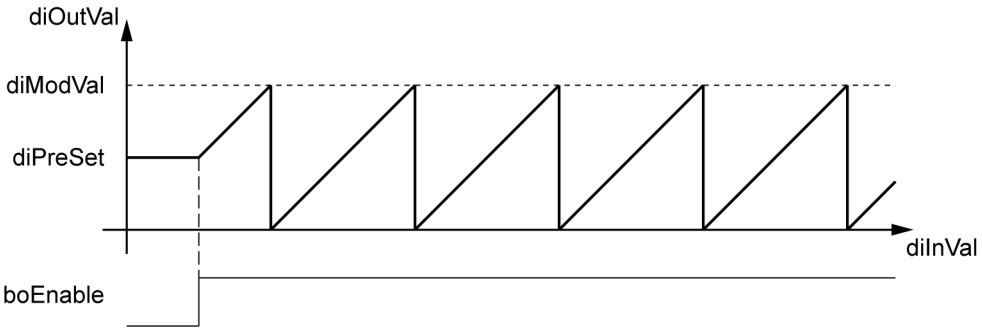
Name	Typ	Beschreibung
fdiModulo_Modulo	DINT	Modulo-Wert von 'diInVal' [Inkremente]

4.2.1.1.9.4 MODULO_COUNT (FB)

Der Funktionsbaustein 'MODULO_COUNT' realisiert einen Modulozähler.

Der Zähler kann auf einen Wert voreingestellt werden.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.



Anwender Interface

	MODULO_COUNT			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert -	diInVal	boOverFlow	-	Modulo-Umschlag
Modulo-Wert -	diModVal	diOutVal	-	Ausgangswert
Preset-Wert -	diPreSet			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Eingangswert [Inkremente]

Name	Typ	Beschreibung
diModVal	DINT	Modulo Endwert des Modulozählers [Inkremente] Ausgang 'diOutVal' zählt von 0 bis 'diModVal'
diPreSet	DINT	Voreinstellwert, auf den der Ausgang gesetzt wird. Der Wert wird mit der positiven Flanke von 'boEnable' übernommen. [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boOverFlow	BOOL	Markierung des Modulo-Umschlags; bei Über- und Unterlauf ist 'boOverFlow' für einen Zyklus TRUE.
diOutVal	DINT	Ausgangswert [Inkremente]

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

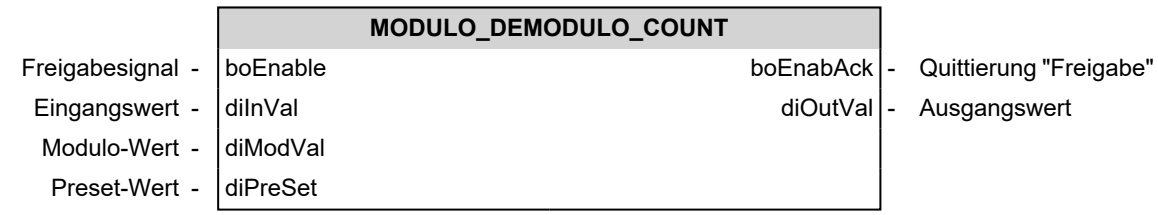
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.9.5 MODULO_DEMODULO_COUNT (FB)

Der Funktionsbaustein 'MODULO_DEMODULO_COUNT' realisiert einen Zähler, der einen Modulozählwert in einen fortlaufenden Zählwert umwandelt.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Eingangswert [Inkremente]
diModVal	DINT	Modulo Endwert des Modulozählers [Inkremente]
diPreSet	DINT	Voreinstellwert, auf den der Ausgang gesetzt wird. Der Wert wird mit der positiven Flanke von 'boEnable' übernommen. [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diOutVal	DINT	Fortlaufender Ausgangswert [Inkremente]

Verwendungshinweis im CoDeSys-Programm

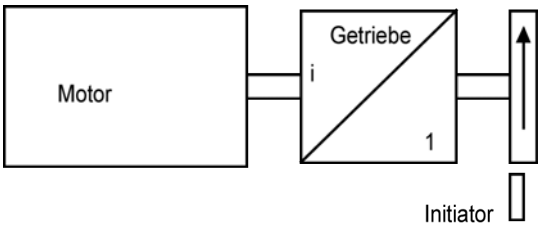
Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

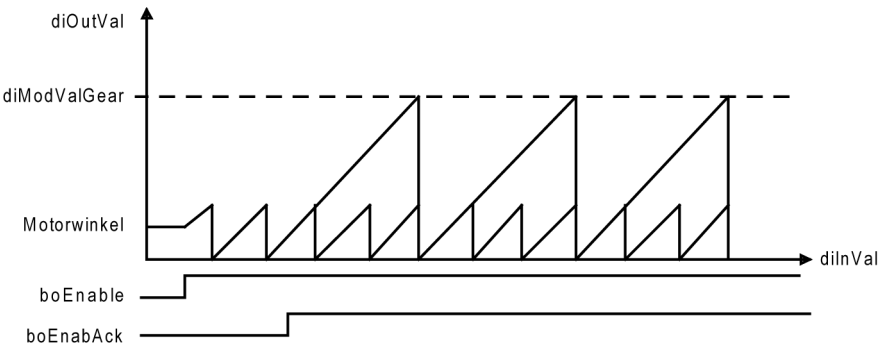
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.9.6 MODULO_MODCOUNT_GEAR (FB)

Der Funktionsbaustein 'MODULO_MODCOUNT_GEAR' realisiert die Voraussetzungen für eine genaue Positionierung von endlos rotierenden Achsen mit Getriebe über das Aus- und Einschalten der Anlage hinaus, wie zum Beispiel Druckzylinder. Die Positionierung erfolgt modulo, d. h. innerhalb einer Umdrehung des Getriebe-Abtriebs. Dazu muss der Motor mit einem Singleturn-Absolutwertgeber ausgerüstet sein (z. B. S-Geber) und ein Initiator am Getriebe-Abtrieb muss die Umdrehung 1 des Motors innerhalb einer Umdrehung des Getriebeabtriebs markieren.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.





Anwender Interface

		MODULO_MODCOUNT_GEAR			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Initiator Eingang	-	boInitiator	boOverFlow	-	Modulo-Umschlag erfolgt
Eingangsimpulse	-	diInVal	diOutVal	-	Modulo Ausgangswert
Modulo-Wert	-	diModInVal	diIniPos	-	Initiatorposition
Getriebeübersetzung	-	iGearRatio	boErr	-	Fehlersignal
			iErrID	-	Fehler Identnummer

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boInitiator	BOOL	Initiator Eingang

Name	Typ	Beschreibung
diInVal	DINT	Eingangswert, Zähler Eingang, Lageistwert Motor
diModInVal	DINT	Modulo-Wert; legt fest, wie weit der Ausgangswert 'diOutVal' zählt (von 0 bis 'diModInVal') [Inkremente]
iGearRatio	INT	Getriebeübersetzung zwischen Eingang und Ausgang

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben Der Funktionsbaustein hat die Initiatorflanke erkannt.
boOverFlow	BOOL	Markierung des Modulo-Umschlags; bei Über- und Unterlauf ist 'boOverFlow' für einen Zyklus TRUE.
diOutVal	DINT	Ausgangswert; der Ausgangswert 'diOutVal' zählt von 0 bis 'diModInVal' [Inkremente]
diIniPos	DINT	Position, an der die Low->High Flanke des Initiators erkannt wurde
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 1: diModInVal ≤ 1 iErrID = 2: iGearRatio < 1

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	MODULO_MODCOUNT_GEAR

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.9.7 MODULO_MODULO1_TO_MODULO2_COUNT (FB)

Der Funktionsbaustein 'MODULO_MODULO1_TO_MODULO2_COUNT' realisiert einen Umwandlungszähler, der einen Modulozählwert in einen anderen Modulozählwert umwandelt.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	MODULO_MODULO1_TO_MODULO2_COUNT		
Freigabesignal -	boEnable	boEnabAck	- Quittierung "Freigabe"
Eingangswert -	diInVal	boOverFlow	- Überlaufmeldung
Modulo-Wert -	diModVal1	diOutVal	- Ausgangswert
Preset-Wert -	diPreSet1		
Modulo-Wert -	diModVal2		
Preset-Wert -	diPreSet2		

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Eingangswert [Inkremente]
diModVal1	DINT	Eingang: Modulo Endwert des Modulozählers [Inkremente]
diPreSet1	DINT	Voreinstellwert, auf den der Eingangswert bei einer positiven Flanke an 'boEnable' gesetzt wird [Inkremente]
diModVal2	DINT	Ausgang: Modulo Endwert des Modulozählers [Inkremente]
diPreSet2	DINT	Voreinstellwert, auf den der Ausgangswert bei einer positiven Flanke an 'boEnable' gesetzt wird [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boOverFlow	BOOL	Markierung des Modulo-Umschlags; bei Über- und Unterlauf ist 'boOverFlow' für einen Zyklus TRUE.
diOutVal	DINT	Ausgangswert [Inkremente]

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

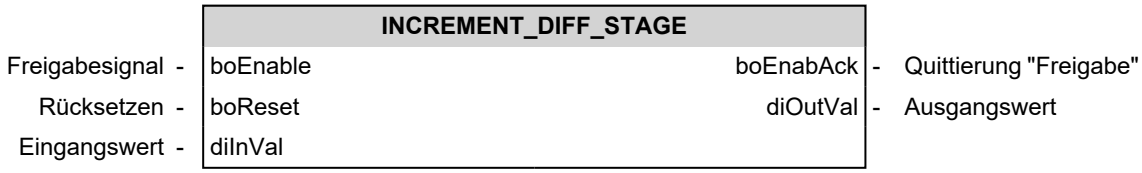
4.2.1.1.10 12_Increment

4.2.1.1.10.1 INCREMENT_DIFF_STAGE (FB)

Der Funktionsbaustein 'INCREMENT_DIFF_STAGE' realisiert eine Differenzstufe, die Bausteine entkoppelt und Positionen verarbeitet. Den Ausgang der Differenzstufe bildet ein 32-bit-Zähler.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boReset	BOOL	Reset; Ausgangswert = 0, anschließend wird der Eingang 'boReset' vom Baustein selbständig zurückgesetzt
diInVal	DINT	Eingangswert [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diOutVal	DINT	Ausgangswert [Inkremente] 32-Bit-Zähler

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

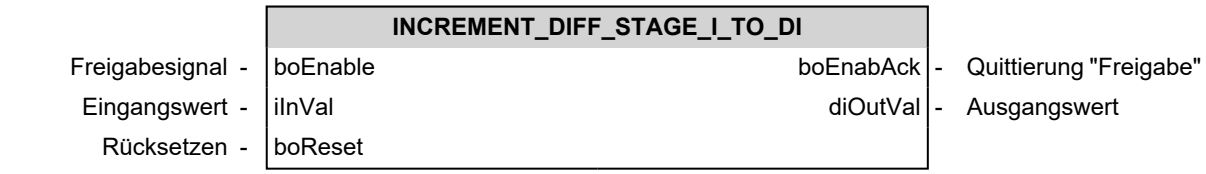
4.2.1.1.10.2 INCREMENT_DIFF_STAGE_I_TO_DI (FB)

Der Funktionsbaustein 'INCREMENT_DIFF_STAGE_I_TO_DI' realisiert eine Differenzstufe, die Bausteine entkoppelt und Positionen verarbeitet.

Das Format wird vom INT-Eingang in einen DINT-Ausgang umgewandelt.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
iInVal	INT	Eingangswert [Inkremente]
boReset	BOOL	Reset; Ausgangswert = 0, anschließend wird der Eingang 'boReset' vom Baustein selbständig zurückgesetzt

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diOutVal	DINT	Ausgangswert [Inkremente] 32-Bit-Zähler

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.10.3 INCREMENT_MULTIPLEX_2_INT_TO_1_DINT (FB)

Differenzstufe mit 2 INT-Eingängen. Diese werden addiert zum DINT-Ausgang.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	INCREMENT_MULTIPLEX_2_INT_TO_1_DINT			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Rücksetzen -	boReset	diOutVal	-	Ausgangswert
Eingang aktivieren -	boInput1Enable			
Eingangswert -	iInput1			
Eingang aktivieren -	boInput2Enable			
Eingangswert -	iInput2			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boReset	BOOL	Reset; Ausgangswert = 0, anschließend wird der Eingang 'boReset' vom Baustein selbständig zurückgesetzt
boInput1Enable, boInput2Enable	BOOL	Eingang aktivieren Die Differenz wird auf den Ausgang addiert
iInput1, iInput2	INT	Eingangswert [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diOutVal	DINT	Ausgangswert [Inkremente] 32-Bit-Zähler

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

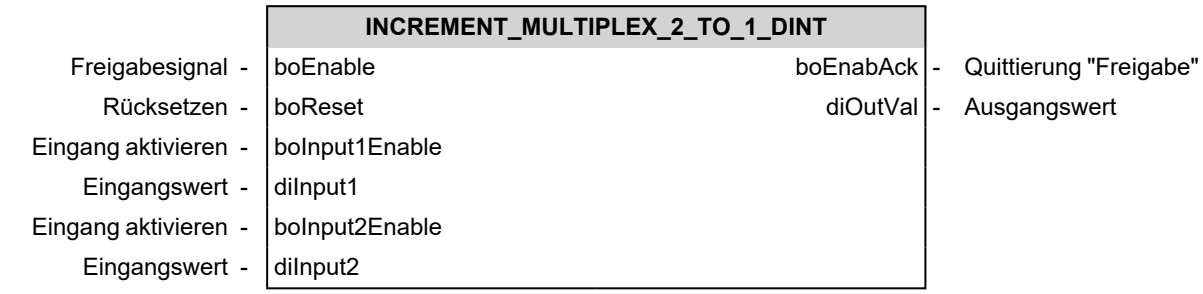
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.10.4 INCREMENT_MULTIPLEX_2_TO_1_DINT (FB)

Multiplexer mit 2 DINT-Eingängen, die zum Ausgang addiert werden.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boReset	BOOL	Reset; Ausgangswert = 0, anschließend wird der Eingang 'boReset' vom Baustein selbständig zurückgesetzt
boInput1Enable, boInput2Enable	BOOL	Eingang aktivieren Die Differenz wird auf den Ausgang addiert
diInput1, diInput2	DINT	Eingangswert [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diOutVal	DINT	Ausgangswert [Inkremente] 32-Bit-Zähler

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

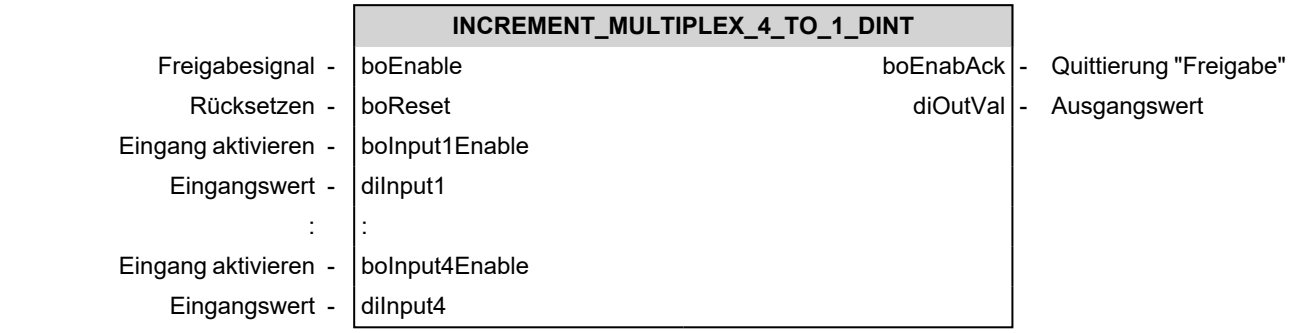
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.10.5 INCREMENT_MULTIPLEX_4_TO_1_DINT (FB)

Multiplexer mit 4 DINT-Eingängen. Diese werden addiert zum Ausgang.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boReset	BOOL	Reset; Ausgangswert = 0, anschließend wird der Eingang 'boReset' vom Baustein selbständig zurückgesetzt
boInput1Enable ... boInput4Enable	BOOL	Eingang aktivieren Die Differenz wird auf den Ausgang addiert
diInput1 ... diInput4	DINT	Eingangswert [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diOutVal	DINT	Ausgangswert [Inkremente] 32-Bit-Zähler

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.10.6 INCREMENT_OFFSET (FB)

Der Funktionsbaustein 'INCREMENT_OFFSET' realisiert die Ausgabe von Lagewerten zur Offsetverschiebung. Dabei lassen sich die Schrittgröße und der Verschiebewert übergeben.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	INCREMENT_OFFSET			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Verschiebewert -	diOffset	boBusy	-	Ausführung läuft
Schrittweite -	uiStep	boDone	-	Quittierung "Ausgeführt"
		diOutVal	-	Ausgangswert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diOffset	DINT	Verschiebewert [Inkremente]
uiStep	UINT	Schrittgröße [Inkremente/ID2 'SERCOS-Zykluszeit']

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
diOutVal	DINT	Ausgangswert [Inkremente]

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.10.7 INCREMENT_SUMMATION (FB)

Der Funktionsbaustein 'INCREMENT_SUMMATION' realisiert einen Summationspunkt für die Lagevorgabe.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	INCREMENT_SUMMATION			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert -	diInVal			
Lagesollwert -	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Eingangswert [Inkremente] Inkrementenquelle, die in das Lagesollwertsystem geschrieben werden soll.

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	INCREMENT_SUMMATION

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.11 13_Timing control

4.2.1.1.11.1 TIMING_CONTROL_BLINK (FB)

Der Funktionsbaustein 'TIMING_CONTROL_BLINK' liefert einen Blinktakt.

Die Zeiten für Signal ein und Signal aus können unabhängig voneinander parametriert werden.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	TIMING_CONTROL_BLINK			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Zeit für Ein -	uiBlinkOn	boOutBlink	-	Blinksignal
Zeit für Aus -	uiBlinkOff			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
uiBlinkOn	UINT	Zeit für Blinksignal EIN [ms]
uiBlinkOff	UINT	Zeit für Blinksignal AUS [ms]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boOutBlink	BOOL	Blinksignal

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.11.2 TIMING_CONTROL_CLOCK (FB)

Der Funktionsbaustein 'TIMING_CONTROL_CLOCK' stellt Takte zur zeitgesteuerten Ausführung von Programm-Aktionen zur Verfügung (z.B. Gleitkomma-Berechnungen und Istwert-Ausgaben).

Die Takte sind nicht miteinander synchronisiert. Das bedeutet, dass die Rechenlast gleichmäßiger verteilt wird als bei synchronisierten Takten.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	TIMING_CONTROL_CLOCK			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Start-Zeit setzen -	boSetRealTimeClock	boClock50ms	-	Pulse 50 ms
Start-Zeit -	dtSetDateTime	boClock100ms	-	Pulse 100 ms
ms-Zähler -	diTime	boClock250ms	-	Pulse 250 ms
		boClock500ms	-	Pulse 500 ms
		boClock1s	-	Pulse 1 s
		dtDateTime	-	Zeit und Datum

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boSetRealTimeClock	BOOL	Echtzeituhr und Datum auf Startwert 'dtSetDateTime' setzen
dtSetDateTime	DATE_ AND_ TIME	Startwert Zeit und Datum
diTime	DINT	ms-Zähler [ms]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boClock50ms boClock100ms boClock250ms boClock500ms	BOOL	Puls alle n ms . boClocknms bleibt einen PLC-Zyklus lang TRUE.

Name	Typ	Beschreibung
boClock1s	BOOL	Pulse alle 1 s. boClock1s bleibt einen PLC-Zyklus lang TRUE.
dtDateTime	DATE_ AND_ TIME	Zeit und Datum

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	TIMING_CONTROL_CLOCK

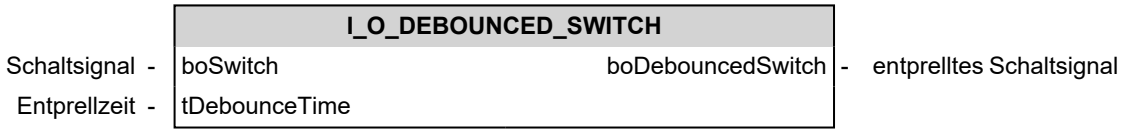
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.12 14_I_O_Handling

4.2.1.1.12.1 I_O_DEBOUNCED_SWITCH (FB)

Der Funktionsbaustein 'I_O_DEBOUNCED_SWITCH' realisiert einen entprellten Schalter.
Wird der Eingang 'boSwitch' geschaltet, wird der Ausgang 'boDebouncedSwitch' nach Ablauf der Entprellzeit 'tDebounceTime' gesetzt. Mit 'boSwitch' = FALSE wird auch der Ausgang 'boDebouncedSwitch' auf FALSE geschaltet.
Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boSwitch	BOOL	Schaltsignal, das entprellt werden soll
tDebounceTime	TIME	Entprellzeit [ms] (Standardwert = 50 ms)

Ausgangsvariablen

Name	Typ	Beschreibung
boDebouncedSwitch	BOOL	Entprelltes Schaltsignal

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
I_O_DEBOUNCED_SWITCH	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

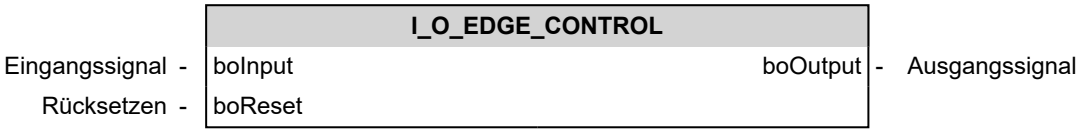
4.2.1.1.12.2 I_O_EDGE_CONTROL (FB)

Der Funktionsbaustein 'I_O_EDGE_CONTROL' steuert einen Ausgang 'boOutput' so, dass er durch eine positive Flanke am Eingang 'boInput' gesetzt und durch eine negative Flanke rückgesetzt wird.

Durch den Eingang 'boReset' wird der Ausgang gelöscht. Nach dem Löschen ist immer eine 0 -> 1 Flanke am Eingang 'boInput' nötig, damit der Ausgang wieder gesetzt wird.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boInput	BOOL	Eingangssignal
boReset	BOOL	Rücksetzen; durch den Eingang 'boReset' wird der Ausgang gelöscht.

Ausgangsvariablen

Name	Typ	Beschreibung
boOutput	BOOL	Ausgangssignal

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

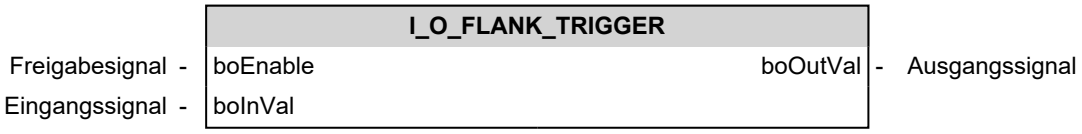
4.2.1.1.12.3 I_O_FLANK_TRIGGER (FB)

Der Funktionsbaustein 'I_O_FLANK_TRIGGER' realisiert eine Flankenauswertung des Eingangs 'boInVal', um den Ausgang 'boOutVal' zu schalten.

Ist die Flankenauswertung durch den Eingang 'boEnable' = TRUE freigegeben, schaltet der Eingang den Ausgang flankengesteuert. Wird die Freigabe entzogen, wird der Ausgang zurückgesetzt.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet. Wird die Freigabe entzogen, so wird der Ausgang gelöscht.
boInVal	BOOL	Eingangssignal, schaltet flankengesteuert den Ausgang

Ausgangsvariablen

Name	Typ	Beschreibung
boOutVal	BOOL	Ausgangssignal

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.12.4 I_O_FLANK_TRIGGER_ACKNOWLEDGEMENT (FB)

Der Funktionsbaustein 'I_O_FLANK_TRIGGER_ACKNOWLEDGEMENT' realisiert eine Flankenauswertung des Eingangs 'boInVal', wenn der Eingang 'boInValAck' = FALSE ist, um den Ausgang 'boOutVal' zu schalten.

Wird an 'boInValAck' das Quittierungssignal erkannt, führt ein FALSE Signal am Eingang 'boInVal' oder 'boInValAck' zum zurücksetzen des Ausgangs 'boOutVal'. Der Eingang 'tTimeOut' legt fest, in welcher Zeit das Quittierungssignal am Eingang 'boInValAck' erwartet wird. Liegt das Quittierungssignal nicht in der angegebenen Zeit an, dann wird der Ausgang 'boErrTimeOut' gesetzt.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	I_O_FLANK_TRIGGER_ACKNOWLEDGEMENT			
Eingangsbit -	boInVal	boOutVal	-	Ausgangssignal
Quittierungsbit -	boInValAck	boErr	-	Fehlersignal
Timeout -	tTimeOut	iErrID	-	Fehler Identnummer

Eingangsvariablen

Name	Typ	Beschreibung
boInVal	BOOL	Mit einer positiven Flanke wird bei 'boInValAck' = FALSE der Ausgang 'boOutVal' gesetzt.
boInValAck	BOOL	Quittierungssignal
tTimeOut	TIME	Timeout Zeit

Ausgangsvariablen

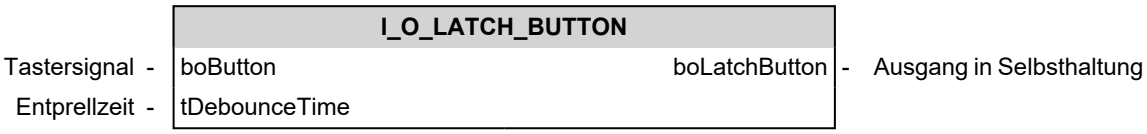
Name	Typ	Beschreibung
boOutVal	BOOL	Ausgangssignal
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung Wert 1: Timeout (Quittierungssignal nicht in der angegebenen Zeit an 'boInValAck' erkannt)

4.2.1.1.12.5 I_O_LATCH_BUTTON (FB)

Der Funktionsbaustein 'I_O_LATCH_BUTTON' realisiert einen Taster mit Selbsthaltung und Entprellung.
Bei einer 0 -> 1 Flanke am Eingang 'boButton' wird nach der Entprellzeit der Ausgang 'boLatchButton' von 0 -> 1 geschaltet.
Bei einer erneuten 0 -> 1 Flanke am Eingang schaltet der Ausgang wieder von 1 -> 0.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boButton	BOOL	Taster, der entprellt und selbst gehalten werden soll.
tDebounceTime	TIME	Entprellzeit [ms] (Standardwert = 50 ms)

Ausgangsvariablen

Name	Typ	Beschreibung
boLatchButton	BOOL	Selbstgehaltener Taster

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
I_O_LATCH_BUTTON	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.12.6 I_O_PLUS_MINUS_BUTTON (FB)

Mit dem Funktionsbaustein 'I_O_PLUS_MINUS_BUTTON' wird der Ausgangswert erhöht und vermindert.

Wird eine Taste, z. B. 'boButtonPlus', einmalig betätigt, wird der Ausgangswert 'diActValue' um die Schrittweite von 'diStep' nach oben gezählt.
Wird die Taste länger als die an 'tByWaitingPeriod' eingestellte Zeit betätigt, zählt der Ausgang selbständig hoch. Die Zählgeschwindigkeit kann dabei über den Eingang 'tStepTime' angepasst werden. Durch die Eingänge 'diMinValue' und 'diMaxValue' wird der Ausgang begrenzt.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	I_O_PLUS_MINUS_BUTTON			
Tastersignal für Pluszählung -	boButtonPlus	diActValue	-	Ausgangswert
Tastersignal für Minuszählung -	boButtonMinus			
Presetwert -	diPreset			
Schrittweite der Zählung -	diStep			
obere Grenze -	diMaxValue			
untere Grenze -	diMinValue			
Entprellzeit -	tDebounceTime			
Wartezeit bis Dauerzählung -	tByWaitingPeriod			
Schrittdauer -	tStepTime			

Eingangsvariablen

Name	Typ	Beschreibung
boButtonPlus	BOOL	Taster für Pluszählung
boButtonMinus	BOOL	Taster für Minuszählung
diPreset	DINT	Presetwert für 'diActValue' bei Aufruf der Aktion(actPreset)
diStep	DINT	Schrittweite: Weg, der bei Dauerbetätigung der Tipptaste maximal abgefahren wird
diMaxValue	DINT	obere Grenze der Ausgabe an 'diActValue'
diMinValue	DINT	untere Grenze der Ausgabe an 'diActValue'
tDebounceTime	TIME	Entprellzeit [ms] (Standardwert 200 ms)
tByWaitingPeriod	TIME	Wartezeit bis Dauerzählung [ms] (Standardwert 1000 ms)
tStepTime	TIME	Schrittdauer [ms] (Standardwert 400 ms)

Ausgangsvariablen

Name	Typ	Beschreibung
diActValue	DINT	Ausgangswert

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
I_O_PLUS_MINUS_BUTTON	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

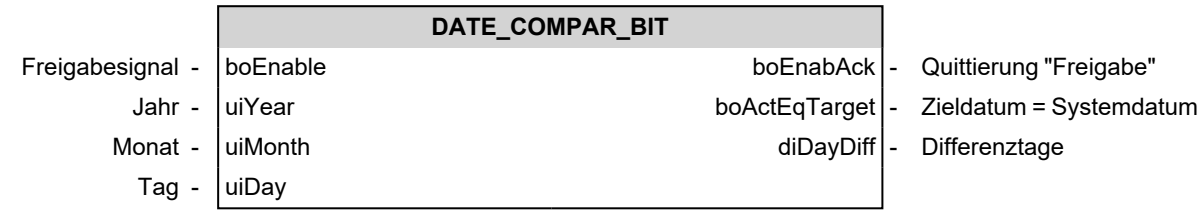
4.2.1.1.13 17_Date

4.2.1.1.13.1 DATE_COMPAR_BIT (FB)

Der Funktionsbaustein 'DATE_COMPAR_BIT' setzt eine Meldung ab, wenn das aktuelle Systemdatum mit dem vorgegebenen Zieldatum übereinstimmt.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
uiYear	UINT	Jahreszahl des Datums
uiMonth	UINT	Monat des Datums
uiDay	UINT	Tag des Datums

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boActEqTarget	BOOL	Zieldatum = Systemdatum
diDayDiff	DINT	Anzahl der Tage, die zwischen Zieldatum und aktuellem Systemdatum liegen

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
DATE_COMPAR_BIT	

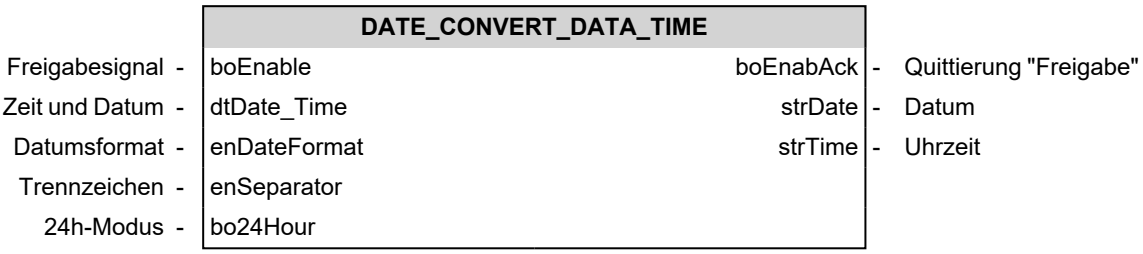
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.13.2 DATE_CONVERT_DATA_TIME (FB)

Der Funktionsbaustein 'DATE_CONVERT_DATA_TIME' wandelt eine Datumseingabe des Typs DT in eine formatierte Datums- und Zeitanzeige.
(Beispiel: DT#2010-03-30-10:00:52)

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
dtDate_Time	DATE_ AND_ TIME	Datum und Zeit, die konvertiert werden sollen.
enDateFormat	ENUM	Datumsformat einstellen: enDateFormat_YYYY_MM_DD: Reihenfolge Jahr - Monat - Tag enDateFormat_DD_MM_YYYY: Reihenfolge Tag - Monat - Jahr enDateFormat_MM_DD_YYYY: Reihenfolge Monat - Tag - Jahr
enSeparator	ENUM	Trennzeichen auswählen enSeparator_Point: Trennzeichen Punkt z.B. DD.MM.YYYY: 30.03.2010 enSeparator_Minus: Trennzeichen Minus z.B. YYYY-MM-DD: 2010-03-30 enSeparator_Slash: Trennzeichen Slash z.B. MM/DD/YYYY: 03/30/2010
bo24Hour	BOOL	Schaltet zwischen 24- und 12-Stunden Modus um bo24Hour = TRUE => 24-h-Modus: 23:24:53 bo24Hour = FALSE => 12-h-Modus: 11:24:53 pm

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
strDate	STRING	Formatierte Datumsanzeige
strTime	STRING	Formatierte Zeitanzeige

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
DATE_CONVERT_DATA_TIME	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.13.3 DATE_PAYDAY (FB)

Der Funktionsbaustein 'DATE_PAYDAY' vergleicht das Datum aus 'stDate_Payday' mit dem Systemdatum.

Ist das Datum erreicht oder überschritten, dann wird der Ausgang 'boPayDay' auf TRUE gesetzt. Der Benutzer muss ein Passwort eingeben, um die Maschine wieder freizuschalten.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

		DATE_PAYDAY			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Passwort	-	strPassword_Input	boPayDay	-	Zahltag, Maschine Stop
			boWrong_Password	-	Falsches Passwort
			boCorrect_Password	-	Richtiges Passwort
Bezahlt	-	boPaid	boPaid	-	Bezahlt
Datum des Zahltages	-	stDate_Payday	stDate_Payday	-	Datum des Zahltages
Feld Datum-Passwort	-	ar_stPassword	ar_stPassword	-	Feld Datum-Passwort

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
strPassword_Input	STRING	Eingabe des Passwortes, mit dem die Maschine wieder freigeschaltet werden soll.

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boPayDay	BOOL	'boPayDay' = TRUE: Zahltag-Datum erreicht, Maschine stoppen und Passwort anfordern.
boWrong_Password	BOOL	Einen Zyklus lang TRUE, wenn ein falsches Passwort eingegeben wurde.
boCorrect_Password	BOOL	Einen Zyklus lang TRUE, wenn das richtige Passwort eingegeben wurde

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
boPaid	BOOL	Gibt an ob, schon bezahlt wurde. D.h. dass ein Passwort eingegeben wurde, das mit einem Zahltagsdatum „0“ hinterlegt wurde.
stDate_Payday	STRUCT	ST_DATE_PAYDAY Struktur mit dem Datum des nächsten Zahltags.
ar_stPassword	ARRAY	ARRAY [1.. MAX_LEN_ST_PAYDAY] OF ST_PAYDAY Feld, in dem die Daten und Passwörter hinterlegt sind. Es sollte mindestens ein Feldelement mit einem Passwort und dem Datum 0 übergeben werden. Dieses schaltet die Maschine dann dauerhaft frei.

Die Ein-/Ausgangsvariablen
- boPaid
- stDate_Payday
- ar_stPassword

sollten remanent global deklariert werden

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
DATE_PAYDAY	

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.13.4 fboDate_Leap_Year (F)

Die Funktion 'fboDate_Leap_Year' überprüft, ob es sich beim eingegebenen Jahr um ein Schaltjahr handelt.
Die Regel lautet:
- Ist es durch 400 teilbar, dann ist es ein Schaltjahr;
- Sonst: Ist es durch 100 teilbar, dann ist es kein Schaltjahr;
- Sonst: Ist es durch 4 teilbar, dann ist es ein Schaltjahr.

(Beispiel: 1996 = Schaltjahr; 2000 = Schaltjahr; 2100 = kein Schaltjahr)
Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	fboDate_Leap_Year		
Jahr -	uiYear	fboDate_Leap_Year	- Schaltjahr

Eingangsvariablen

Name	Typ	Beschreibung
uiYear	UINT	Jahreszahl des Datums, die überprüft werden soll

Ausgangsvariable

Name	Typ	Beschreibung
fboDate_Leap_Year	BOOL	TRUE: Schaltjahr FALSE: kein Schaltjahr

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fboDate_Leap_Year	

4.2.1.1.13.5 fdiDate_Diff_Time (F)

Die Funktion 'fdiDate_Diff_Time' bestimmt die Differenz zweier Daten in Tagen.
Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	fdiDate_Diff_Time		
Datum -	stDate1	fdiDate_Diff_Time	- Anzahl Tage
Datum -	stDate2		

Eingangsvariablen

Name	Typ	Beschreibung
stDate1	STRUCT	ST_DATE Späteres Datum
stDate2	STRUCT	ST_DATE Früheres Datum

Ausgangsvariable

Name	Typ	Beschreibung
fdiDate_Diff_Time	DINT	Differenz der beiden Daten [day] "fdiDate_Diff_Time" < 0, wenn "stDate1" < "stDate2"

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fdiDate_Diff_Time	

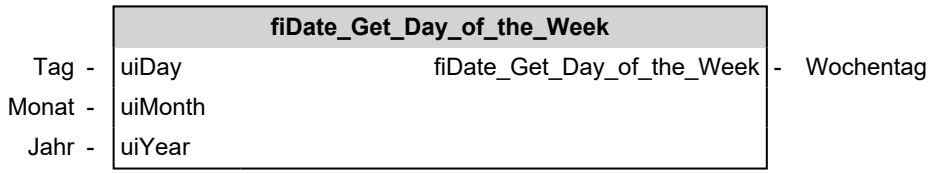
4.2.1.1.13.6 fiDate_Get_Day_of_the_Week (F)

Die Funktion 'fiDate_Get_Day_of_the_Week' bestimmt den Wochentag des übergebenen Datums.

Der Wochentag wird nach der Gauß'schen Wochentagsformel berechnet. Diese Formel gilt seit der Einführung des Gregorianischen Kalenders am 15.10.1582 und prinzipiell so lange, bis von diesen Kalenderregeln wieder abgewichen wird, also bis zur nächsten Kalenderreform.

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
uiDay	UINT	Tag des Datums, für das der Wochentag gesucht wird
uiMonth	UINT	Monat des Datums, für das der Wochentag gesucht wird
uiYear	UINT	Jahreszahl des Datums, für das der Wochentag gesucht wird

Ausgangsvariable

Name	Typ	Beschreibung
fiDate_Get_Day_of_the_Week	INT	Ermittelter Wochentag: 0 = Sonntag 1 = Montag 2 = Dienstag 3 = Mittwoch 4 = Donnerstag 5 = Freitag 6 = Samstag

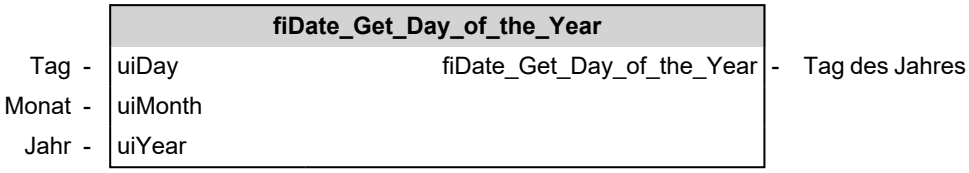
Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fiDate_Get_Day_of_the_Week	

4.2.1.1.13.7 fiDate_Get_Day_of_the_Year (F)

Die Funktion 'fiDate_Get_Day_of_the_Year' bestimmt den Tag des Jahres.
Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
uiDay	UINT	Tag des Datums, für das der Tag des Jahres gesucht wird
uiMonth	UINT	Monat des Datums, für das der Tag des Jahres gesucht wird
uiYear	UINT	Jahreszahl des Datums, für das der Tag des Jahres gesucht wird

Ausgangsvariable

Name	Typ	Beschreibung
fiDate_Get_Day_of_the_Year	INT	Tag des Jahres

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fiDate_Get_Day_of_the_Year	

4.2.1.1.13.8 fusiDate_Get_Calendar_Week_EU (F)

Die Funktion 'fusiDate_Get_Calendar_Week_EU' bestimmt die Kalenderwoche des vorgegebenen Datums.
Die Berechnung erfolgt gemäß DIN 1355, welche u.a. besagt:

- Jeden Montag und nur Montags beginnt eine neue Kalenderwoche.
- Die 1. Kalenderwoche ist diejenige, die mindestens 4 Tage des neuen Jahres enthält.

Aus diesen Punkten ergeben sich folgende Eigenschaften:

- Es gibt keine unvollständigen Kalenderwochen, ausnahmslos jede KW enthält genau 7 Tage.
- Jedes Jahr hat entweder 52 oder 53 Kalenderwochen.
- Ein Jahr hat genau dann 53 Kalenderwochen, wenn es mit einem Donnerstag beginnt oder endet:
  - Ein Normaljahr mit 53 Wochen beginnt und endet an einem Donnerstag.
  - Ein Schaltjahr mit 53 Wochen beginnt entweder an einem Mittwoch und endet somit mit Donnerstag oder beginnt an einem Donnerstag und endet an einem Freitag.
- Der 4. Januar liegt immer in Kalenderwoche 1.
- Der 29., 30. und 31. Dezember können schon zur ersten Kalenderwoche des Folgejahres gehören.
- Der 1., 2. und 3. Januar können noch in der letzten Kalenderwoche des Vorjahres liegen.
- Der Donnerstag ist ausschlaggebend, zu welchem Jahr die Woche gezählt wird. Liegt er im neuen Jahr, ist es die Kalenderwoche 1.

(aus Wikipedia.org)

Normalfall:

$$KW = \frac{TagDesJahres - 1 + Wochentag(1.Januar)}{7} + 1$$

Sonderfälle:

- Der 1. Januar des entsprechenden Jahres ist ein Freitag oder Samstag:
  $$KW = \frac{TagDesJahres - 1 + Wochentag(1.Januar) - 7}{7} + 1$$
- Jahresanfang 01. - 03.01.:
  TagDesJahres - 1 + Wochentag(1.Januar) - 7 ≤ 0:
  KW ist die letzte KW des Vorjahres
- Jahresende 29. - 31.12.:
  Ergibt die obige KW-Berechnung KW=53, dann ist es KW 53, wenn der 1. Januar dieses Jahres ein Donnerstag oder, sofern es ein Schaltjahr ist, ein Mittwoch ist.
  Andernfalls ist es KW1 des Folgejahres.

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	fusiDate_Get_Calendar_Week_EU			
Tag -	uiDay	fusiDate_Get_Calendar_Week_EU	-	Kalenderwoche EU
Monat -	uiMonth			
Jahr -	uiYear			

Eingangsvariablen

Name	Typ	Beschreibung
uiDay	UINT	Tag des Datums, für das die Kalenderwoche gesucht wird
uiMonth	UINT	Monat des Datums, für das die Kalenderwoche gesucht wird
uiYear	UINT	Jahreszahl des Datums, für das die Kalenderwoche gesucht wird

Ausgangsvariable

Name	Typ	Beschreibung
fusiDate_Get_Calendar_ Week_EU	USINT	Kalenderwoche nach DIN 1355

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fusiDate_Get_Calendar_Week_EU	

4.2.1.1.13.9 fusiDate_Get_Calendar_Week_USA (F)

Die Funktion 'fusiDate_Get_Calendar_Week_USA' bestimmt die Kalenderwoche des Datums auf Basis der amerikanischen Regeln.

- Woche = So Mo .... Sa
- 1. Januar ist in KW1

$$KW = \frac{TagDesJahres + Wochentag(1.Januar)}{7} + 1$$

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	fusiDate_Get_Calendar_Week_USA			
Tag -	uiDay	fusiDate_Get_Calendar_Week_USA	-	Kalenderwoche USA
Monat -	uiMonth			
Jahr -	uiYear			

Eingangsvariablen

Name	Typ	Beschreibung
uiDay	UINT	Tag des Datums, für das die Kalenderwoche gesucht wird
uiMonth	UINT	Monat des Datums, für das die Kalenderwoche gesucht wird
uiYear	UINT	Jahreszahl des Datums, für das die Kalenderwoche gesucht wird

Ausgangsvariable

Name	Typ	Beschreibung
fusiDate_Get_Calendar_ Week_USA	USINT	Kalenderwoche nach amerikanischer Regel

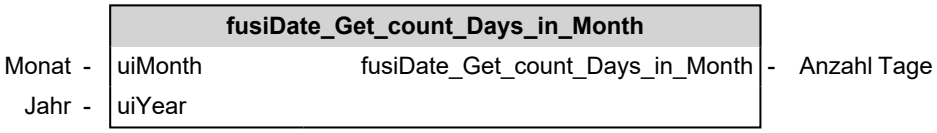
Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fusiDate_Get_Calendar_Week_USA	

4.2.1.1.13.10 fusiDate_Get_count_Days_in_Month (F)

Die Funktion 'fusiDate_Get_count_Days_in_Month' berechnet die Gesamtzahl der Tage eines Monats.
Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
uiMonth	UINT	Monat, für den die Anzahl der Tage bestimmt werden soll.
uiYear	UINT	Jahreszahl des Monats, für den die Anzahl der Tage bestimmt werden soll.

Ausgangsvariable

Name	Typ	Beschreibung
fusiDate_Get_count_Days_ in_Month	USINT	Anzahl der Tage

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fusiDate_Get_count_Days_in_Month	

4.2.1.1.13.11 fusiDate_Get_Eastern (F)

Die Funktion 'fusiDate_Get_Eastern' berechnet den Tag im Jahr, an dem Ostern ist.

Ostern ist ein beweglicher Feiertag. Seit dem Konzil von Nicäa im Jahre 325 nach Christus wird am ersten Sonntag (dem Ostersonntag) nach dem ersten Vollmond nach Frühlingsanfang das Osterfest gefeiert. Damit ist der 22. März der früheste Termin, der 25. April der letzte, auf den Ostern fallen kann.

Das Datum wird nach der Gauß'schen Osterformel in der Korrektur von 1816 berechnet.

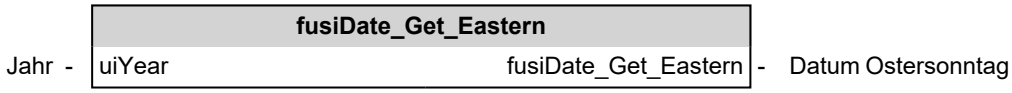
Von diesem Termin hängen viele andere bewegliche christliche Feiertage ab, wie beispielsweise Pfingsten.

Offset für andere (Feier)Tage

Tag	Offset [Tage]
Schwerdonnerstag / Weiberfastnacht	-52
Rosenmontag	-48
Fastnachtsdienstag	-47
Aschermittwoch	-46
Gründonnerstag	-3
Karfreitag	-2
Ostersonntag	0
Ostermontag	+1
Christi Himmelfahrt	+39
Pfingstsonntag	+49
Pfingstmontag	+50
Fronleichnam	+60

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
uiYear	UINT	Jahreszahl, für das der Ostertermin bestimmt werden soll

Ausgangsvariable

Name	Typ	Beschreibung
fusiDate_Get_Eastern	USINT	Tag des Osterfests (Format: 2#M00D DDDD) • Bit 0..4: Tag des Datums • Bit 7 = 0: März Bit 7 = 1: April Beispiele: 2007: 1000 1000 => 08.04. 2008: 0001 0111 => 23.03.

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (sasynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fusiDate_Get_Eastern	

4.2.1.1.14 18_File

4.2.1.1.14.1 Support

FILE_SORT_LIST_ALPHABETIC (FB)

Der Funktionsbaustein 'FILE_SORT_LIST_ALPHABETIC' sortiert eine Dateiliste. Gefüllt wird diese Liste durch den Funktionsbaustein 'FILE_FIND'.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

'Support'-Funktionen und -Funktionsbausteine werden nicht direkt angewendet, sondern von anderen Funktionen und Funktionsbausteinen aufgerufen.

Anwender Interface

		FILE_SORT_LIST_ALPHABETIC			
Funktionsausführung	-	boExec	boDone	-	Quittierung "Ausgeführt"
Anzahl Dateien	-	iFilesFound			
Datei-Liste	-	arystFileName	arystFileName	-	Datei-Liste

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
iFilesFound	INT	Anzahl der gefundenen Dateien in 'arystFileName'

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
arystFileName	ARRAY	ARRAY[0..MAX_LEN_ST_FILE] OF ST_FILE Liste der Dateinamen, die sortiert werden sollen

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
FILE_SORT_LIST_ALPHABETIC	

Es empfiehlt sich, Bausteine, die mit Dateizugriff arbeiten, in einer separaten Task in der asynchronen Programmebene PLC_PRG (Typ "Freilaufend") aufzurufen.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

FILE_STRING_FILTER (FB)

Der Funktionsbaustein 'FILE_STRING_FILTER' entfernt bzw. ersetzt verbotene Zeichen aus einem String.
Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

'Support'-Funktionen und -Funktionsbausteine werden nicht direkt angewendet, sondern von anderen Funktionen und Funktionsbausteinen aufgerufen.

Anwender Interface

		FILE_STRING_FILTER			
Funktionsausführung	-	boExec	strFilterString	-	Gefilterter String
Auszufilternde Zeichen	-	strForbiddenChar	iForbiddenCharCount	-	Anzahl verbotene Zeichen
Ersatzzeichen	-	strReplaceChar	boDone	-	Quittierung "Ausgeführt"
Ersetzen / löschen	-	boReplace	boErr	-	Fehlersignal
String, der geprüft wird	-	strCheckString	iErrID	-	Fehler Identnummer
			strErr	-	String mit gef. Zeichen

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
strForbiddenChar	STRING (15)	Dem String können bis zu 15 Zeichen übergeben werden, die aussortiert werden sollen. Vorbelegung \ : * ? " < > |
strReplaceChar	STRING (1)	Ersatzzeichen Vorbelegung _
boReplace	BOOL	boReplace = TRUE: Gefundene Zeichen werden durch das Ersatzzeichen ersetzt. boReplace = FALSE: Gefundene Zeichen werden aus dem String gelöscht
strCheckString	STRING (64)	String, der überprüft werden soll

Ausgangsvariablen

Name	Typ	Beschreibung
strFilterString	STRING (64)	Ausgabe-String ohne die verbotenen Zeichen
iForbiddenCharCount	INT	Anzahl der in 'strCheckString' gefundenen verbotenen Zeichen
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler

Name	Typ	Beschreibung
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0 — Kein Fehler iErrID ≠ 0 — boErr = TRUE — Fehler iErrID ≠ 0 — boErr = FALSE — Warnung 1 — 'strForbiddenChar' ist nicht definiert oder hat die Länge 0 2 — 'strReplaceChar' ist nicht definiert oder hat die Länge 0 3 — 'strCheckString' ist nicht definiert oder hat die Länge 0
strErr	STRING (64)	Rückmeldung der verbotenen Zeichen, die gefunden wurden.

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
FILE_STRING_FILTER	

Es empfiehlt sich, Bausteine, die mit Dateizugriff arbeiten, in einer separaten Task in der asynchronen Programmebene PLC_PRG (Typ "Freilaufend") aufzurufen.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.14.2 FILE_COPY (FB)

Der Funktionsbaustein 'FILE_ COPY' kopiert Dateien.

Soll auf oder zwischen externen Speichermedien, wie z.B. USB-Speichermedium oder Netzlaufwerken kopiert werden, muss der Zugriff vorher mit der Supportfunktion 'FiFileConnect' (AmkFile.lib) organisiert werden.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

		FILE_COPY			
Funktionsausführung	-	boExec	udSize	-	Dateigröße
Dateiname Quelle	-	strSourceFile	boErr	-	Fehlersignal
Dateiname Ziel	-	strDestinationFile	iErrID	-	Fehler Identnummer
Pointer auf Datenpuffer	-	ptbyWorkBuffer	enErrName	-	Name fehlerh Baustein
Größe des Datenpuffers	-	udWorkBufferSize	boDone	-	Quittierung "Ausgeführt"

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
strSourceFile	STRING	Dateiname der Datei incl. -endung Es sind maximal 64 Zeichen zulässig.
strDestinationFile	STRING	Dateiname der Datei incl. -endung Es sind maximal 64 Zeichen zulässig.
ptbyWorkBuffer	POINTER	POINTER TO BYTE Zeiger auf den Puffer, auf dem die Daten der Quelle zwischengespeichert werden können.

Name	Typ	Beschreibung
udWorkBufferSize	UDINT	Größe des Zwischenspeichers [Byte]

Ausgangsvariablen

Name	Typ	Beschreibung
udSize	UDINT	Größe der Quelldatei [Byte]
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung 1: In 'strSourceFile' ist kein Dateiname vorgegeben 2: LEN(strSourceFile) > 64: Dateiname ist zu lang 3: In 'strDestinationFile' ist kein Dateiname vorgegeben 4: LEN(strDestinationFile) > 64: Dateiname ist zu lang 5: ptbyWorkBuffer: Zeiger auf Speicherbereich ist ungültig
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. READ_FILE_1: AmkFile.lib WRITE_FILE_1: AmkFile.lib SIZE_FILE_1: AmkFile.lib
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
FILE_COPY	

Es empfiehlt sich, Bausteine, die mit Dateizugriff arbeiten, in einer separaten Task in der asynchronen Programmebene PLC_PRG (Typ "Freilaufend") aufzurufen.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.14.3 FILE_DELETE (FB)

Der Funktionsbaustein 'FILE_DELETE' löscht die vorgegebene Datei.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	FILE_DELETE			
Funktionsausführung -	boExec	boErr	-	Fehlersignal
Dateiname -	strFileName	iErrID	-	Fehler Identnummer
		enErrName	-	Name fehlerh Baustein
		boDone	-	Quittierung "Ausgeführt"

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
strFileName	STRING	Dateiname der Datei incl. -endung Es sind maximal 64 Zeichen zulässig.

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung 1: In 'strFileName' ist kein Dateiname vorgegeben 2: LEN(strFileName) > 64: Dateiname ist zu lang
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: REMOVE_FILE_1 – Bibliothek: AmkFile.lib

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
FILE_DELETE	

Es empfiehlt sich, Bausteine, die mit Dateizugriff arbeiten, in einer separaten Task in der asynchronen Programmebene PLC_PRG (Typ "Freilaufend") aufzurufen.

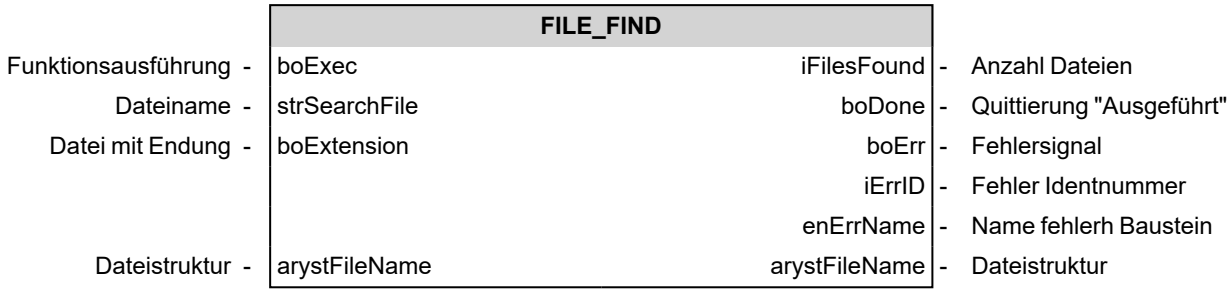
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.14.4 FILE_FIND (FB)

Der Funktionsbaustein 'FILE_FIND' findet eine oder mehrere Dateien und legt deren Namen und Größe in einem Array ab.
Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
strSearchFile	STRING	STRING(64) Dateiname, nach dem gesucht werden soll. Es sind auch Wildcards möglich, z.B. '*.*'
boExtension	BOOL	Dateien mit Endung

Ausgangsvariablen

Name	Typ	Beschreibung
iFilesFound	INT	Anzahl der gefundenen Dateien
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung 1: In 'strSearchFile' sind keine Suchkriterien vorgegeben 2: LEN(strSearchFile) > 64: Suchstring ist zu lang
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. FIND_FILE_1: AmkFile.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
arystFileName	ARRAY	ARRAY[0..MAX_LEN_ST_FILE] OF ST_FILE Dateistruktur

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
FILE_FIND	

Es empfiehlt sich, Bausteine, die mit Dateizugriff arbeiten, in einer separaten Task in der asynchronen Programmebene PLC_PRG (Typ "Freilaufend") aufzurufen.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.14.5 FILE_READ (FB)

Der Funktionsbaustein 'FILE_READ' dient dem Lesen einer Datei und Schreiben des Inhalts in einem Speicherbereich.
Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

		FILE_READ			
Funktionsausführung	-	boExec	udNoByte	-	Gelesene Bytes
Dateiname	-	strFileName	udSizeOfFile	-	Dateigröße
Größe des Datenpuffers	-	udSize	boDone	-	Quittierung "Ausgeführt"
Pointer auf Datenpuffer	-	pbyFileBuff	boErr	-	Fehlersignal
Lesemodus	-	enReadMode	iErrID	-	Fehler Identnummer
			enErrName	-	Name fehlerh Baustein

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
strFileName	STRING	STRING(64) Dateiname der Datei incl. -endung Es sind maximal 64 Zeichen zulässig.
udSize	UDINT	Gibt die Größe des Datenspeichers an [Byte] Ist der Wert größer als der eigentliche Datenbereich, dann wird auf einen unbestimmten Datenbereich zugegriffen.
pbyFileBuff	POINTER	POINTER TO BYTE Zeiger auf den Datenspeicher, auf den die Daten aus der Datei geschrieben werden sollen

Name	Typ	Beschreibung
enReadMode	ENUM	EN_READ_MODE Auswahl des Lese-Modus READ_BEGIN: Der Speicherbereich wird mit den Daten aus der Datei vom Anfang an überschrieben. READ_CURRENT: Der Speicherbereich wird mit den Daten aus der Datei von der aktuellen Position an überschrieben.

Ausgangsvariablen

Name	Typ	Beschreibung
udNoByte	UDINT	Anzahl der gelesenen Bytes aus der Datei [Byte]
udSizeOfFile	UDINT	Größe der gelesenen Datei [Byte] udSize = 0 => udSizeOfFile = Größe der Datei sonst: udSizeOfFile = udSize
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung 1: strFileName: Dateiname nicht definiert 2: LEN(strFileName) > 64: Dateiname zu lang 3: pbyFileBuff: Zeiger auf Speicherbereich ist nicht definiert 4: udSizeOfFile = 0: Dateilänge = 0
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. READ_FILE_1: AmkFile.lib SIZE_FILE_1: AmkFile.lib

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
FILE_READ	

Es empfiehlt sich, Bausteine, die mit Dateizugriff arbeiten, in einer separaten Task in der asynchronen Programmebene PLC_PRG (Typ "Freilaufend") aufzurufen.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.14.6 FILE_WRITE (FB)

Der Funktionsbaustein 'FILE_WRITE' dient dem Schreiben eines Speicherinhaltes in eine Datei.

Nachdem der Schreibvorgang abgeschlossen wurde, wird überprüft, ob die Größen der Datei und des Speichers zusammenpassen.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

		FILE_WRITE			
Funktionsausführung	-	boExec	boDone	-	Quittierung "Ausgeführt"
Dateiname	-	strFileName	boErr	-	Fehlersignal
Pointer auf Datenpuffer	-	pbyFileBuff	iErrID	-	Fehler Identnummer
Größe des Datenpuffers	-	udSize	enErrName	-	Name fehlerh Baustein
Schreibmodus	-	enWriteMode			

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
strFileName	STRING	Dateiname der Datei incl. -endung Es sind maximal 64 Zeichen zulässig.
pbyFileBuff	POINTER	POINTER TO BYTE Zeiger auf den Datenspeicher, der die Daten enthält, die in die Datei geschrieben werden sollen
udSize	UDINT	Gibt die Größe des Datenspeichers an [Byte] Ist der Wert größer als der eigentliche Datenbereich, dann wird auf einen unbestimmten Datenbereich zugegriffen.
enWriteMode	ENUM	EN_WRITE_MODE Auswahl des Schreibmodus WRITE_NEW: Der Inhalt des Speichers überschreibt den bisherigen Inhalt der Datei. WRITE_APPEND: Der Inhalt des Speichers wird an den bisherigen Inhalt der Datei angehängt.

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0 — Kein Fehler iErrID ≠ 0 — boErr = TRUE — Fehler iErrID ≠ 0 — boErr = FALSE — Warnung 1 — strFileName: Dateiname nicht definiert 2 — LEN(strFileName) > 64: Dateiname zu lang 3 — pbyFileBuff: Zeiger auf Speicherbereich ist undefiniert 4 — udSize: Anzahl der zu schreibenden Bytes ist ungültig

Name	Typ	Beschreibung
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: WRITE_FILE_1 – Bibliothek: AmkFile.lib Baustein: SIZE_FILE_1 – Bibliothek: AmkFile.lib

Verwandlung im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
FILE_WRITE	

Es empfiehlt sich, Bausteine, die mit Dateizugriff arbeiten, in einer separaten Task in der asynchronen Programmebene PLC_PRG (Typ "Freilaufend") aufzurufen.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

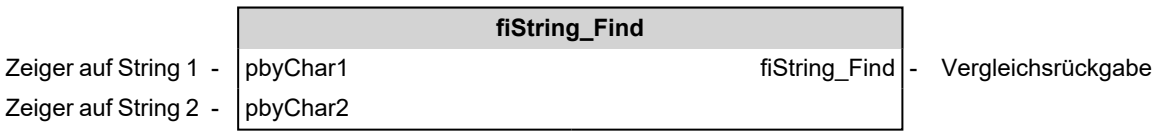
4.2.1.1.15 19_Stringfunctions

4.2.1.1.15.1 fiString_Find (F)

Die Funktion 'fiString_Find' sucht einen Teilstring in einem String.

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
pbyChar1	POINTER	POINTER TO BYTE Pointer auf String 1
pbyChar2	POINTER	POINTER TO BYTE Pointer auf String 2

Ausgangsvariable

Name	Typ	Beschreibung
fiString_Find	INT	Vergleichsrückgabe Position des ersten Zeichens des ersten Vorkommens von 'pbyChar2' in 'pbyChar1'. Wenn 'pbyChar2' in 'pbyChar1' nicht vorkommt, dann gilt fiString_Find := 0.

Verwendungshinweis im CoDeSys-Programm

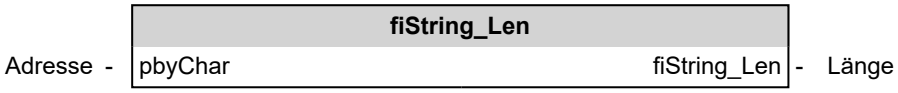
Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

4.2.1.1.15.2 fiString_Len (F)

Die Funktion 'fiString_Len' ermittelt die Länge eines Strings (Anzahl der Zeichen). Der String darf auch mehr als 255 Zeichen enthalten.

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
pbyChar	POINTER	POINTER TO BYTE Adresse des Strings

Ausgangsvariable

Name	Typ	Beschreibung
fiString_Len	INT	Länge des Strings, Anzahl der Zeichen

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

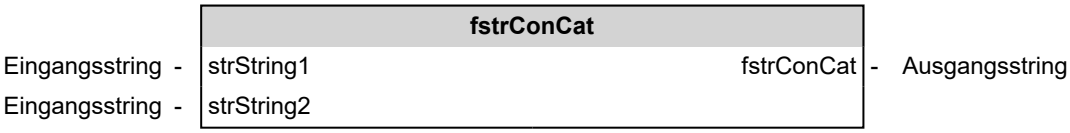
4.2.1.1.15.3 fstrConCat (F)

Die Funktion 'fstrConCat' verbindet zwei STRING-Variablen miteinander.

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

In der AMKAMAC A4 Steuerung ist diese Funktion zu verwenden an Stelle der CONCAT aus der Bibliothek Standard.lib.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
strString1	STRING	Eingangsstring 1, maximale Länge 80 Zeichen
strString2	STRING	Eingangsstring 2, maximale Länge 80 Zeichen

Ausgangsvariable

Name	Typ	Beschreibung
fstrConCat	STRING	Ausgangsstring, maximale Länge 80 Zeichen

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fstrConCat	

4.2.1.1.15.4 fstrString_Append

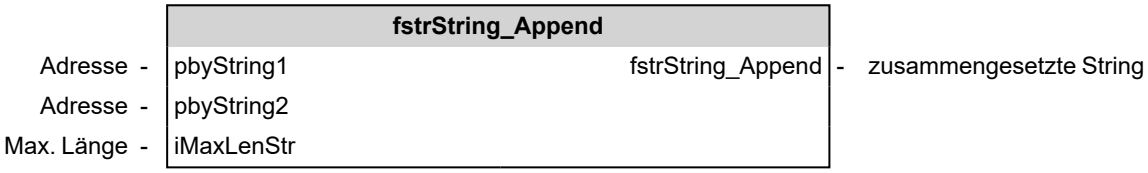
Die Funktion 'fstrString_Append' setzt zwei Strings zusammen. Dabei wird der String 2 an den String 1 angehängt, und zwar so oft, bis die Zeichenlänge von 'iMaxLenStr' erreicht ist.

Beispiele:

iMaxLenStr	String 1	String 2	String 1 neu
0	Hallo	Welt	
3	Hallo	Welt	Hal
7	Hallo	Welt	HalloWe
9	Hallo	Welt	HalloWelt
15	Hallo	Welt	HalloWeltWeltWe

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
pbyString1	POINTER	POINTER TO BYTE Adresse des String 1
pbyString2	POINTER	POINTER TO BYTE Adresse des String 2
iMaxLenStr	INT	Maximale Stringlänge für neuen String1. Gibt durch die Länge des neuen String1, an wie oft der String2 an den String1 angehängt werden kann.

Ausgangsvariable

Name	Typ	Beschreibung
fstrString_Append	STRING	Keine Rückgabe

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

4.2.1.1.15.5 fstrString_Append1 (F)

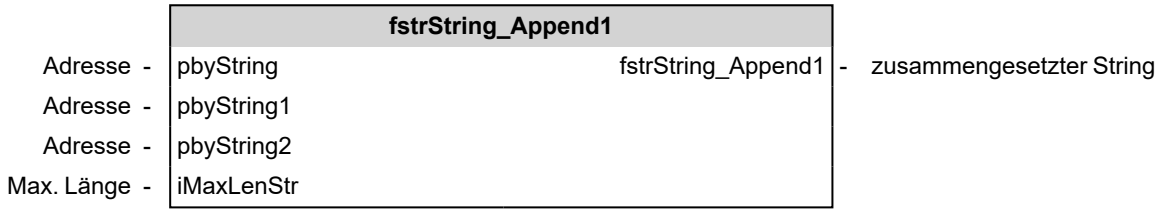
Die Funktion 'fstrString_Append1' setzt zwei Strings in einem neuen String zusammen. Dabei wird der String 2 so oft an den String 1 angehängt, bis die Zeichenlänge von 'iMaxLenStr' erreicht ist.

Beispiele:

iMaxLenStr	String 1	String 2	String
0	Hallo	Welt	
3	Hallo	Welt	Hal
7	Hallo	Welt	HalloWe
9	Hallo	Welt	HalloWelt
15	Hallo	Welt	HalloWeltWeltWe

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
pbyString	POINTER	POINTER TO BYTE Adresse des Strings
pbyString1	POINTER	POINTER TO BYTE Adresse des String 1
pbyString2	POINTER	POINTER TO BYTE Adresse des String 2
iMaxLenStr	INT	Maximale Stringlänge

Ausgangsvariable

Name	Typ	Beschreibung
fstrString_Append1	STRING	Zusammengesetzter String

Verwendungshinweis im CoDeSys-Programm

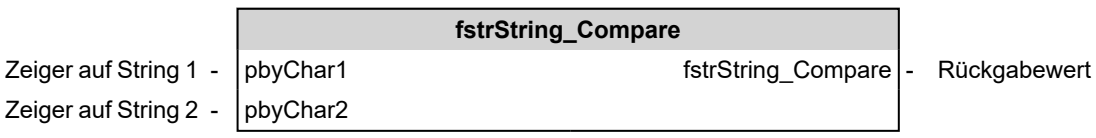
Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

4.2.1.1.15.6 fstrString_Compare (F)

Die Funktion 'fstrString_Compare' vergleicht die Länge zweier Strings

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
pbyChar1	POINTER	POINTER TO BYTE Pointer auf String 1
pbyChar2	POINTER	POINTER TO BYTE Pointer auf String 2

Ausgangsvariable

Name	Typ	Beschreibung
fstrString_Compare	STRING	Rückgabewert 0 Länge (pbyChar1) = Länge (pbyChar2) 1 Länge (pbyChar1) > Länge (pbyChar2) -1 Länge (pbyChar1) < Länge (pbyChar2)

Verwendungshinweis im CoDeSys-Programm

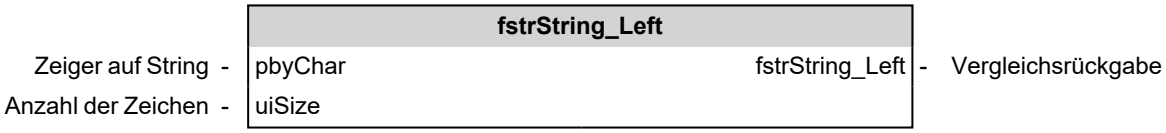
Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

4.2.1.1.15.7 fstrString_Left (F)

Die Funktion 'fstrString_Left' liefert die ersten 'uiSize' Zeichen eines Strings von links.

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
pbyChar	POINTER	POINTER TO BYTE Pointer auf String
uiSize	UINT	Anzahl der Zeichen

Ausgangsvariable

Name	Typ	Beschreibung
fstrString_Left	STRING	Vergleichsrückgabe fstrString_Left(pbyChar, uiSize) bedeutet: Nehme die ersten 'uiSize' Zeichen von links im String 'pbyChar'. Beispiel: fstrString_Left(SUSI,3) = SUS

Verwendungshinweis im CoDeSys-Programm

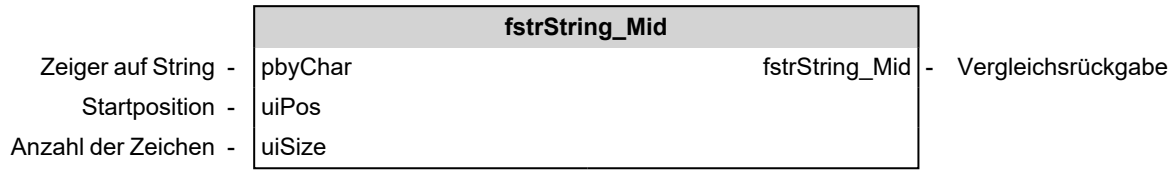
Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

4.2.1.1.15.8 fstrString_Mid (F)

Die Funktion 'fstrString_Mid' liefert einen Teilstring eines Strings.

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
pbyChar	POINTER	POINTER TO BYTE Pointer auf String
uiPos	UINT	Startposition
uiSize	UINT	Anzahl der Zeichen

Ausgangsvariable

Name	Typ	Beschreibung
fstrString_Mid	STRING	Vergleichsrückgabe fstrString_Mid(pbyChar, uiPos, uiSize) bedeutet: Nehme die 'uiSize' Zeichen im String 'pbyChar', beginnend mit dem Zeichen an der Stelle 'uiPos'. Beispiel: fstrString_Mid(SUSI,2,2) = US

Verwendungshinweis im CoDeSys-Programm

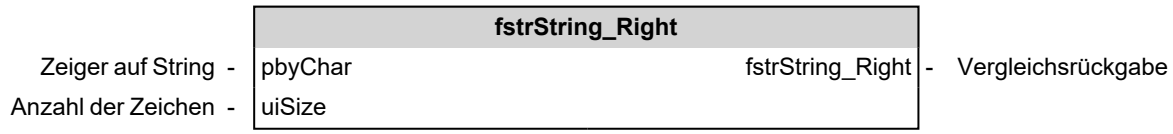
Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

4.2.1.1.15.9 fstrString_Right (F)

Die Funktion 'fstrString_Right' liefert die ersten 'uiSize' Zeichen eines Strings von rechts.

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
pbyChar	POINTER	POINTER TO BYTE Pointer auf String
uiSize	UINT	Anzahl der Zeichen

Ausgangsvariable

Name	Typ	Beschreibung
fstrString_Right	STRING	Vergleichsrückgabe fstrString_Right(pbyChar, uiSize) bedeutet: Nehme die ersten 'uiSize' Zeichen von rechts im String 'pbyChar'. Beispiel: fstrString_Right(SUSI,3) = USI

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

4.2.1.1.16 20_Virtual Master

4.2.1.1.16.1 VIRTUAL_RFGEN (FB)

Der Funktionsbaustein 'VIRTUAL_RFGEN' (Ramp Function Generator) realisiert einen Hochlaufgeber. Mit ihm lässt sich der Ausgang 'diOutVal' über Rampenzeiten entsprechend anpassen, damit sprunghafte Sollwertänderungen vermieden werden.

Wird der Baustein durch den Eingang 'boEnable' aktiviert, wird der Ausgang auf den Wert von 'diMinVal' gesetzt. Ist der Wert an 'diPreset' größer als 'diMinVal', wird der Ausgang über die eingestellte Rampenzeit auf den Presetwert hochgezählt.

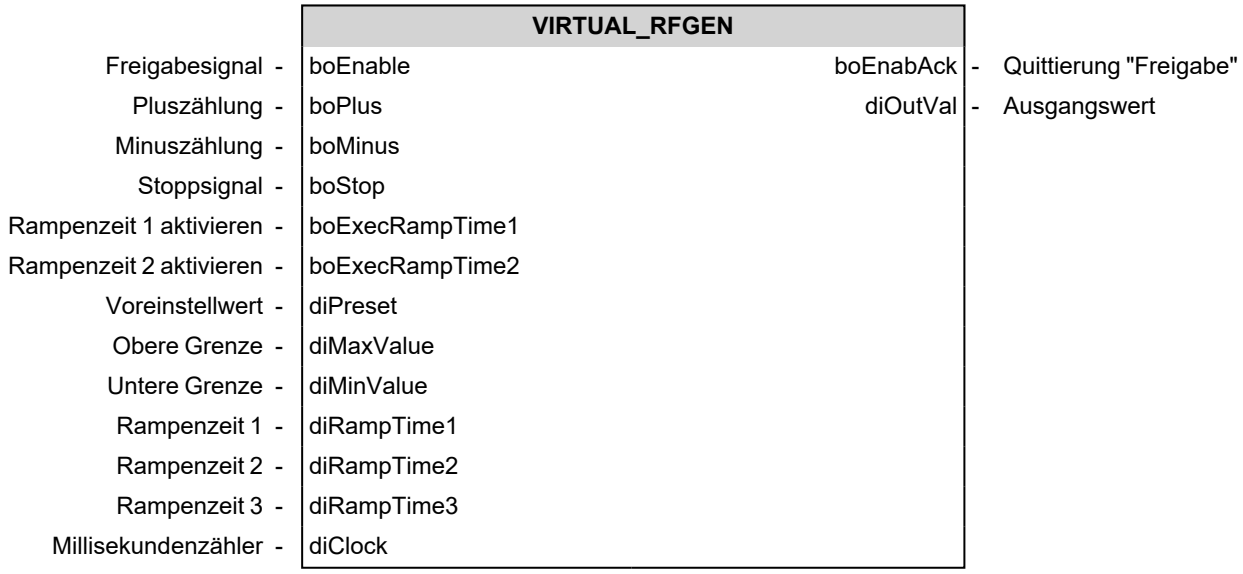
Welche Rampenzeit aktiviert wird, ist abhängig von den Eingängen 'boExecRampTime1' und 'boExecRampTime2'. Mit 'boExecRampTime1' wird die Rampenzeit an 'diRampTime1' aktiviert und mit 'boExecRampTime2' die Rampenzeit an 'diRampTime2'. Ist kein Eingang zur Aktivierung der Rampe gesetzt, ist die Zeit an 'diRampTime3' aktiv.

Durch den Eingang 'boStop' wird der Ausgang über die eingestellte Rampenzeit auf den Minimalwert 'diMinVal' zurückgestellt. Mit den Eingängen 'boPlus' und 'boMinus' kann der Ausgang über die aktivierte Rampe entsprechend erhöht oder vermindert werden.

Der Ausgang wird durch die Eingangsgrößen 'diMaxVal' und 'diMinVal' begrenzt. Dabei müssen beide Eingangsgrößen positiv sein.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boPlus	BOOL	Pluszählung

Name	Typ	Beschreibung
boMinus	BOOL	Minuszählung
boStop	BOOL	Stopp (Minuszählung)
boExecRampTime1	BOOL	Rampenzeit 'diRampTime1' ist aktiv
boExecRampTime2	BOOL	Rampenzeit 'diRampTime2' ist aktiv
diPreset	DINT	Voreinstellwert
diMaxValue	DINT	obere Grenze
diMinValue	DINT	untere Grenze
diRampTime1	DINT	Rampenzeit 1 [s] (Standardwert: 5s)
diRampTime2	DINT	Rampenzeit 2 [s] (Standardwert: 10s)
diRampTime3	DINT	Rampenzeit 3 [s] (Standardwert: 20s)
diClock	DINT	ms-Zähler [ms]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diOutVal	DINT	Ausgangswert

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
VIRTUAL_RFGEN	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.16.2 VIRTUAL_VGEN (FB)

Der Funktionsbaustein 'VIRTUAL_VGEN' realisiert einen virtuellen Geschwindigkeitsgenerator mit der Vorgabe einer Not-Halt-Rampe.
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		VIRTUAL_VGEN			
Funktionsausführung	-	boExec	boBusy	-	Ausführung läuft
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Not-Halt Rampe	-	diEmergency_StopRamp	iErrID	-	Fehler Identnummer
Geschwindigkeitssollwert	-	diVelocity	enErrName	-	Name fehlerh Baustein
Geschwindigkeit in %	-	siOverride			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.

Name	Typ	Beschreibung
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diVelocity	DINT	Geschwindigkeitssollwert [Inkremente/s]
siOverride	SINT	Geschwindigkeitsausgabefaktor

Ausgangsvariablen

Name	Typ	Beschreibung
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: VGEN_AJ; Bibliothek: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	VIRTUAL_VGEN

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.16.3 VIRTUAL_VGEN_A (FB)

Der Funktionsbaustein 'VIRTUAL_VGEN_A' realisiert einen virtuellen Geschwindigkeitsgenerator mit der Vorgabe
- einer Beschleunigungsrampe
- einer Not-Halt-Rampe.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		VIRTUAL_VGEN_A			
Funktionsausführung	-	boExec	boBusy	-	Ausführung läuft
Not-Halt	-	boEmergency_Stop	boErr	-	Fehlersignal
Not-Halt Rampe	-	diEmergency_StopRamp	iErrID	-	Fehler Identnummer
Geschwindigkeitssollwert	-	diVelocity	enErrName	-	Name fehlerh Baustein
Beschleunigung	-	udAccel			
Geschwindigkeit in %	-	siOverride			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diVelocity	DINT	Geschwindigkeitssollwert [Inkremente/s]
udAccel	UDINT	Beschleunigung, mit der auf die Sollgeschwindigkeit gefahren wird
siOverride	SINT	Geschwindigkeitsausgabefaktor

Ausgangsvariablen

Name	Typ	Beschreibung
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein** / **Bibliothek** VGEN_AJ / AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	VIRTUAL_VGEN_A

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.16.4 VIRTUAL_VGEN_AJ (FB)

Der Funktionsbaustein 'VIRTUAL_VGEN_AJ' realisiert einen virtuellen Geschwindigkeitsgenerator mit der Vorgabe
- einer Beschleunigungsrampe mit Ruckbegrenzung
- einer Verzögerungsrampe mit Ruckbegrenzung
- einer Not-Halt-Rampe.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	VIRTUAL_VGEN_AJ			
Funktionsausführung -	boExec	boBusy	-	Ausführung läuft
Not-Halt -	boEmergency_Stop	boErr	-	Fehlersignal
Not-Halt Rampe -	diEmergency_StopRamp	iErrID	-	Fehler Identnummer
Geschwindigkeitssollwert -	diVelocity	enErrName	-	Name fehlerh Baustein
Beschleunigung -	udAccel			
Verzögerung -	udDecel			
Beschleunigungsruck -	udAccJerk			
Verzögerungsruck -	udDecJerk			
Geschwindigkeit in % -	siOverride			
Lagesollwert -	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
diVelocity	DINT	Geschwindigkeitssollwert [Inkremente/s]
udAccel	UDINT	Beschleunigung, mit der auf die Sollgeschwindigkeit gefahren wird
udDecel	UDINT	Verzögerung, mit der eine niedrigere Sollgeschwindigkeit angefahren wird
udAccJerk / udDecJerk	UDINT	Ruck bei der Beschleunigung / Verzögerung [Inkremente/s^3]
siOverride	SINT	Geschwindigkeitsausgabefaktor

Ausgangsvariablen

Name	Typ	Beschreibung
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein**: VGEN_AJ — **Bibliothek**: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	VIRTUAL_VGEN_AJ

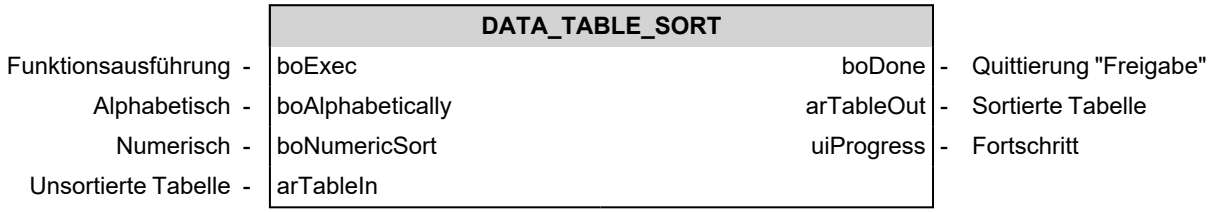
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.1.17 21_Data

4.2.1.1.17.1 DATA_TABLE_SORT (FB)

Der Funktionsbaustein 'DATA_TABLE_SORT' sortiert das übergebene Datenfeld alphabetisch oder numerisch.
Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
boAlphabetically	BOOL	Alphabetische Sortierung
boNumericSort	BOOL	Numerische Sortierung
arTableIn	ARRAY	ARRAY [1..g_cuiDataTableLength] OF DATA_TABLE Tabelle, die sortiert werden soll

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
arTableOut	ARRAY	ARRAY [1..g_cuiDataTableLength] OF DATA_TABLE Sortierte Tabelle
uiProgress	UINT	Fortschritt: Zeigt den Fortschritt prozentual an

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
DATA_TABLE_SORT	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2 06_TechnologyLevel

4.2.1.2.1 01_Emergency stop

4.2.1.2.1.1 EMERGENCY_BREAK_POSITION (FB)

Der Funktionsbaustein 'EMERGENCY_BREAK_POSITION' entkoppelt bei Not-Halt die Positionsvorgabe und fährt diese in der am Eingang 'diTRamp' festgelegten Zeit zum Stillstand herunter.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	EMERGENCY_BREAK_POSITION			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Not-Halt -	boEmergency_Stop	boBusy	-	Ausführung läuft
Eingangswert -	diInVal	boDone	-	Quittierung "Ausgeführt"
Rampenzeit -	diTRamp	diOutVal	-	Ausgangswert
SERCOS Zykluszeit -	diID2_SERCOS_cycle			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diInVal	DINT	Eingangswert [Inkremente]
diTRamp	DINT	Rampenzeit [ms]
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
diOutVal	DINT	Ausgangswert [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	EMERGENCY_BREAK_POSITION

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.1.2 EMERGENCY_BREAK_VELOCITY (FB)

Der Funktionsbaustein 'EMERGENCY_BREAK_VELOCITY' entkoppelt bei Not-Halt die Geschwindigkeitsvorgabe und fährt diese in der am Eingang 'diTRamp' festgelegten Zeit auf Null herunter.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	EMERGENCY_BREAK_VELOCITY			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Not-Halt -	boEmergency_Stop	boBusy	-	Ausführung läuft
Eingangswert -	diInVal	boDone	-	Quittierung "Ausgeführt"
Rampenzeit -	diTRamp	diOutVal	-	Ausgangswert
SERCOS Zykluszeit -	diID2_SERCOS_cycle			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diInVal	DINT	Eingangswert [Inkremente]
diTRamp	DINT	Rampenzeit [ms]
diID2_SERCOS_cycle	DINT	ID2 'SERCOS-Zykluszeit' [µs]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
diOutVal	DINT	Ausgangswert [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	EMERGENCY_BREAK_VELOCITY

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.2 02_Print marks

4.2.1.2.2.1 PRINT_MARK_CONTROL (FB)

Der Funktionsbaustein 'PRINT_MARK_CONTROL' ist ein Basisbaustein zur Realisierung einer Druckmarkenregelung.
Er muss vom Anwender nicht direkt benutzt werden und kommt in den folgenden Bausteinen zu Einsatz:

- PRINT_MARK_INSETTER_CONT
- PRINT_MARK_INSETTER_INTERVAL
- PRINT_MARK_REGCONT_CONTINUOUS
- PRINT_MARK_REGISTER_CONTROL_DISCONT

Folgender Funktionsumfang wird durch den Baustein abgedeckt:

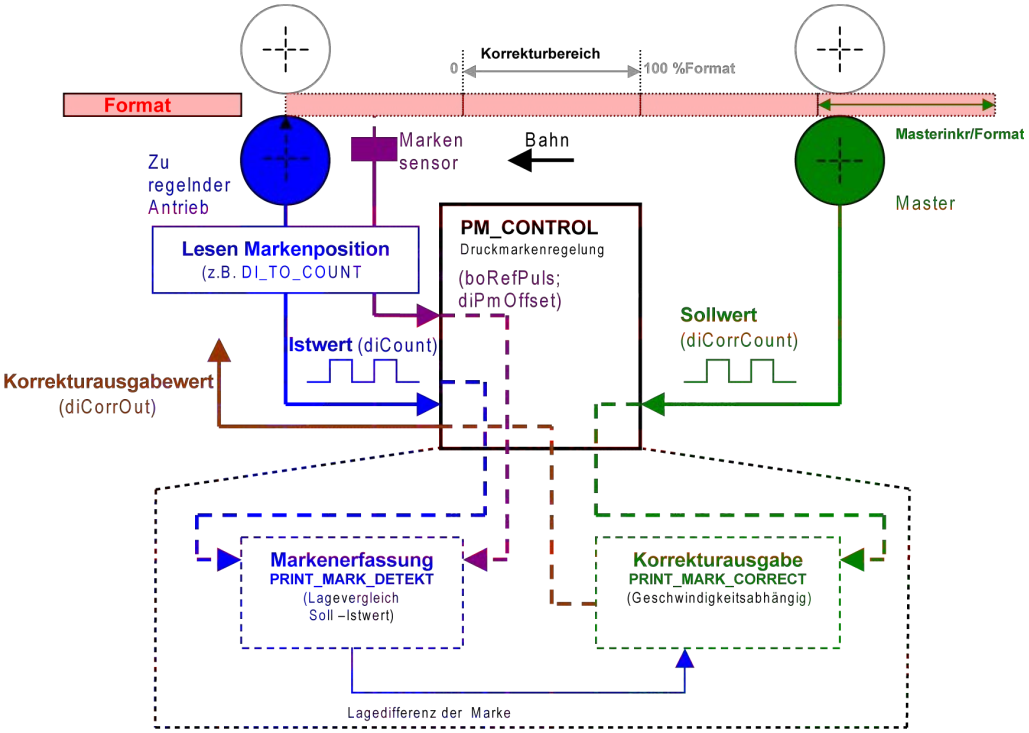
- Die Abweichung der Druckmarken-Soll-Ist-Position wird berechnet und als Korrekturwert ausgegeben.
- Der Baustein realisiert zwei Arten der Druckmarken-Sollpositions-Erfassung:
  - Die Druckmarken-Sollposition kann auf eine absolute Position innerhalb des Druckformates festgelegt werden
  - Die erste erkannte Druckmarke legt die Sollposition fest.
- Steuerung des Druckmarkensensors in Abhängigkeit eines Fensters
- Überwachung der Druckmarken
- Simulation fehlender Druckmarken
- Ausgabe des Korrekturweges als gleichmäßige Verteilung der Inkremente über einen vorgebbaren Korrekturweg
- Begrenzung der Korrekturgeschwindigkeit
- Ausgabe eines Offsets zur Verschiebung der Druckmarken-Sollposition

Der Baustein arbeitet mit positiver Zählrichtung der Eingangsimpulse und liefert positive Ausgangsimpulse.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

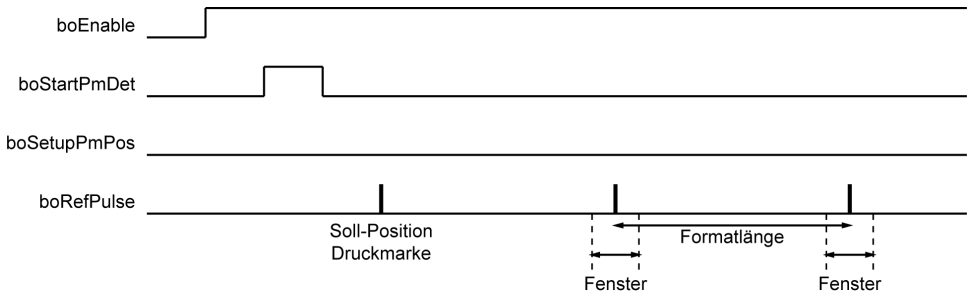
Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Prinzipskizze: PRINT_MARK_CONTROL Verwendung als Registerregler

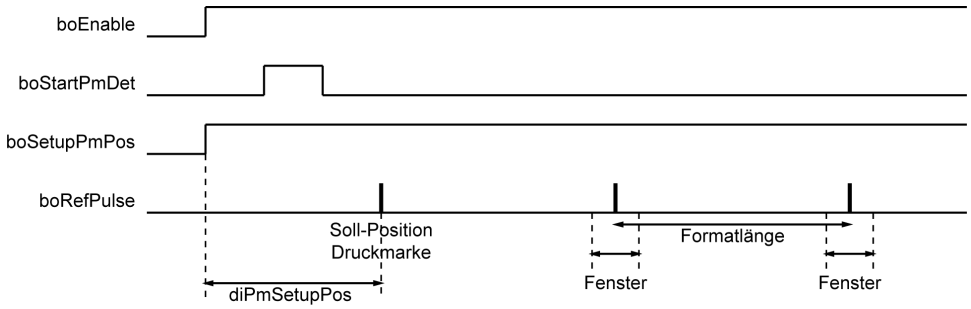


Festlegung der Druckmarkensollposition

- **Start der Druckmarkenregelung mit Suche der Sollposition:**
  Nach positiver Flanke von 'boStartPmDet' wird der 1. Impuls als Soll-Position der Druckmarke übernommen.
  Alle weiteren Druckmarken werden auf diese Position geregelt.



- **Start der Druckmarkenregelung mit Vorgabe der Sollposition**:
  Bei positiver Flanke von 'boStartPmDet' wird zur aktuellen Masterposition der Wert 'diPmSetupPos' addiert.
  Dieser Wert gibt die Soll-Position der nächsten Druckmarke an. Alle Druckmarken werden auf diese Position geregelt.



Markensimulation

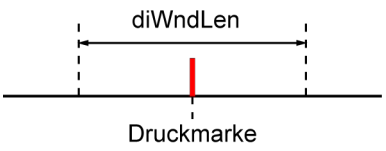
Der Wert 'iNoPmLost' gibt an, wie viele Marken simuliert werden sollen, bis der Ausgang 'boPmLost' = TRUE wird (Marke verloren). Der Ausgangswert 'iActPmLost' zeigt an, wie viele Druckmarken seit der letzten erkannten Druckmarke simuliert wurden.
Wird bis zum Ende des Markenfensters kein Markenimpuls erkannt, wird eine Marke simuliert. Es wird angenommen, dass die Marke an der Soll-Position erkannt worden ist. Der Ausgang 'iActPmLost' wird um 1 erhöht.
Der Ausgang 'boPmLost' hat keinen Einfluss auf die Regelung.

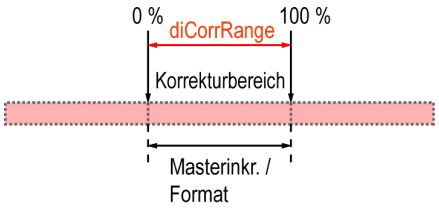
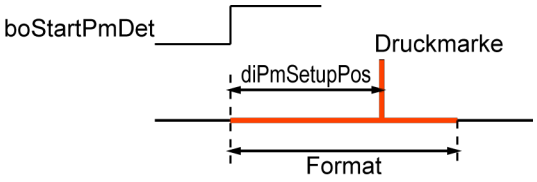
Anwender Interface

		PRINT_MARK_CONTROL			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Formatlänge	-	diFormat			
Fensterlänge	-	diWndLen			
Anzahl Marken bis Meldung 'Marke verloren'	-	iNoPmLost			
Korrekturbereich	-	diCorrRange			
Druckmarken-Sollposition	-	diPmSetupPos			
Max. Korrekturwert pro Format	-	diCorrectionLimit			
Modus Insetter / Registerregler	-	enPmcMode			

		actSync			
Rücksetzen Ausgang boPmLost	-	boResetPmLost	boPmDetected	-	Marke gefunden
Start Markensuche	-	boStartPmDet	boPmLost	-	Marke verloren
Freigabe Korrekturausgabe	-	boCorrEnable	boPmWnd	-	Fenster ein
Start Ausgabe Operator-Offset	-	boSetupOffset	diPmDeviation	-	Abweichung Soll-Ist-Position
Start mit Druckmarken-Sollposition	-	boSetupPmPos	diOpOffsetOut	-	Ausgegebener Operator-Offset
Druckmarke erkannt	-	boRefPulse	diCorrOut	-	Ausgabewert
Impulse des zu regelnden Aggregates	-	diCount	iActPmLost	-	Anzahl verlorener Marken
Druckmarken-Offset	-	diPmOffset			
Operator Offset	-	diOpOffset			
Impulse des Geschwindigkeitsmasters	-	diCorrCount			

Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diFormat	DINT	Formatlänge Soll-Abstand zweier Druckmarken
diWndLen	DINT	Fensterlänge: Länge vor und hinter der Soll-Druckmarkenposition, auf der der Druckmarkensensor aktiviert ist und eine Druckmarke akzeptiert wird. 
iNoPmLost	INT	Anzahl Druckmarken, die nacheinander fehlen dürfen, bevor der Ausgang 'boPMLost' (Marke verloren) gesetzt wird.

Name	Typ	Beschreibung
diCorrRange	DINT	Korrekturbereich, die Druckmarkenregelung versucht im angegebenen Bereich die Korrektur auszugeben [% der Formatlänge] 
diPmSetupPos	DINT	Der Parameter gibt die Sollposition der Druckmarke zur aktuellen Masterposition bei positiver Flanke von 'boStartPmDet' an. Die erkannten Druckmarken werden auf diese Position geregelt. [Inkremente] 
diCorrectionLimit	DINT	Maximale Korrekturausgabe pro Format [Inkremente] 'diCorrectionLimit' = 0: keine Begrenzung 'diCorrectionLimit' > 0: maximaler Wert
enPmcMode	ENUM	EN_PMC_MODE Druckmarkenregelungsmode: _enPmcInsetter: Regelung eines Vorzugs (Insetting) _enPmcRegisterController: Regelung eines Bearbeitungsaggregats (Registerregler)

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
boResetPmLost	BOOL	Rücksetzen des Ausgangs 'boPmLost'
boStartPmDet	BOOL	Start der Druckmarkenregelung 'boStartPmDet' = TRUE: Druckmarkenregelung ist aktiviert Durch den Eingang 'boSetupPmPos' wird festgelegt, welche Druckmarkenposition beim Start der Druckmarkenregelung berücksichtigt wird.
boCorrEnable	BOOL	Freigabe der Korrekturausgabe. (Kann mit Baustein 'PRINT_MARK_CORROUT_CONTROL' angesteuert werden, um Korrekturausgaben nur in bestimmten Bereichen zuzulassen)
boSetupOffset	BOOL	Ausgabestart des angegebenen Werts von 'diOpOffset'. Dieser Wert dient als Offset zur Verschiebung der Druckmarken-Sollposition
boSetupPmPos	BOOL	Steuerung der Sollpositionsbestimmung der Druckmarke bei positiver Flanke von 'boStartPmDet' 'boPmSetupPos' = FALSE: Es wird die Position der nächsten Druckmarke als Soll-Position erfasst und alle nachfolgenden Marken auf diese Position geregelt. 'boPmSetupPos' = TRUE: Der Parameter 'diPmSetupPos' gibt die Soll-Position der Druckmarke relativ zur aktuellen Masterposition an. Die folgenden Druckmarken werden auf diese Position geregelt.

Name	Typ	Beschreibung
boRefPulse	BOOL	Referenzimpuls (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport DI_TO_COUNT
diCount	DINT	Zählerwert Impulse des zu regelnden Aggregates zur Bestimmung der Druckmarkenabweichung
diPmOffset	DINT	Druckmarkenoffset der aktuellen Position (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport DI_TO_COUNT
diOpOffset	DINT	Operator-Offset. Verschiebt die Sollposition der Druckmarke. Die Verschiebung erfolgt im Korrekturbereich und wird mit einer 0→1 Flanke an 'boSetupOffset' gestartet. Sie ist unabhängig von Wertänderungen an 'diCount'. Die Verschiebegeschwindigkeit berechnet sich aus
diCorrCount	DINT	Impulse des Geschwindigkeitsmasters (Bahngeschwingkeitsmaster)

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben

Ausgangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
boPmDetected	BOOL	Marke erkannt, Regelung aktiv
boPmLost	BOOL	Meldung "Marke verloren", wenn die an Parameter 'iNoPmLost' eingestellte Anzahl von aufeinanderfolgenden Marken nicht erkannt wurde.
boPmWnd	BOOL	Markenfenster aktiviert den Druckmarkensensor
diPmDeviation	DINT	Abweichung Soll- und Istposition der aktuellen Druckmarke
diOpOffsetOut	DINT	Ausgabewert beinhaltet den aktuell ausgegebenen Operatoroffset
diCorrOut	DINT	Korrekturwert
iActPmLost	INT	Anzahl verlorener Druckmarken

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
PRINT_MARK_CONTROL	PRINT_MARK_CONTROL.actSync

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.2.2 PRINT_MARK_CORRECT (FB)

Der Funktionsbaustein 'PRINT_MARK_CORRECT' gibt einen Weg aus in Abhängigkeit von der Geschwindigkeit eines Masters.

Die Korrekturgeschwindigkeit ist abhängig von der Größe des Korrekturwertes und dem Bereich, in dem die Korrektur ausgeführt werden soll.

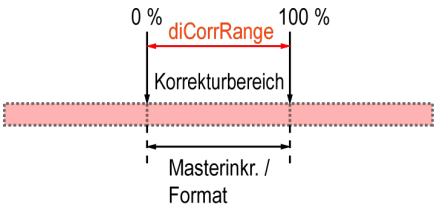
Der Ausgabewert kann direkt an den zu regelnden Antrieb ausgegeben werden.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	PRINT_MARK_CORRECT		
Freigabesignal -	boEnable	boEnabAck	- Quittierung "Freigabe"
Ausgabe freigeben -	boCorrEnable	diOutVal	- Ausgangswert
Korrekturwert -	diCorrVal		
Masterimpulse -	diCount		
Korrekturbereich -	diCorrRange		

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boCorrEnable	BOOL	Freigabe der Korrekturausgabe. Durch diesen Eingang lässt sich die Ausgabe steuern. Mit 'boCorrEnable' = TRUE ist die Wertausgabe am Ausgang 'diOutVal' freigegeben. Mit 'boCorrEnable' = FALSE ist die Ausgabe am Ausgang 'diOutVal' gesperrt. Dadurch ist es möglich, eine Korrekturausgabe nur dann auszugeben, wenn es erwünscht ist. (z.B. in bestimmten Bereichen)
diCorrVal	DINT	Länge der Wegstrecke (Korrekturwert) [Inkremente]
diCount	DINT	Masterimpulse [Inkremente]
diCorrRange	DINT	Korrekturbereich, die Druckmarkenregelung versucht im angegebenen Bereich die Korrektur auszugeben [% der Formatlänge]  Korrekturbereich, sollte kleiner als ein Format sein [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben

Name	Typ	Beschreibung
diOutVal	DINT	Ausgangswert [Inkremente] Der alte Korrekturwert am Eingang 'diCorrVal' wird mit dem neuen Korrekturwert am Eingang 'diCorrVal' verrechnet. Beispiel: Alter Wert an 'diCorrVal' = 500, neuer Wert an 'diCorrVal' = 700 => am Ausgang 'diOutVal' werden 200 Inkremente in Abhängigkeit der Mastergeschwindigkeit ('diCount') und dem Korrekturbereich ('diCorrRange') ausgegeben.

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	PRINT_MARK_CORRECT

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.2.3 PRINT_MARK_CORROUT_CONTROL (FB)

Der Baustein 'PRINT_MARK_CORROUT_CONTROL' realisiert folgend Funktionen:

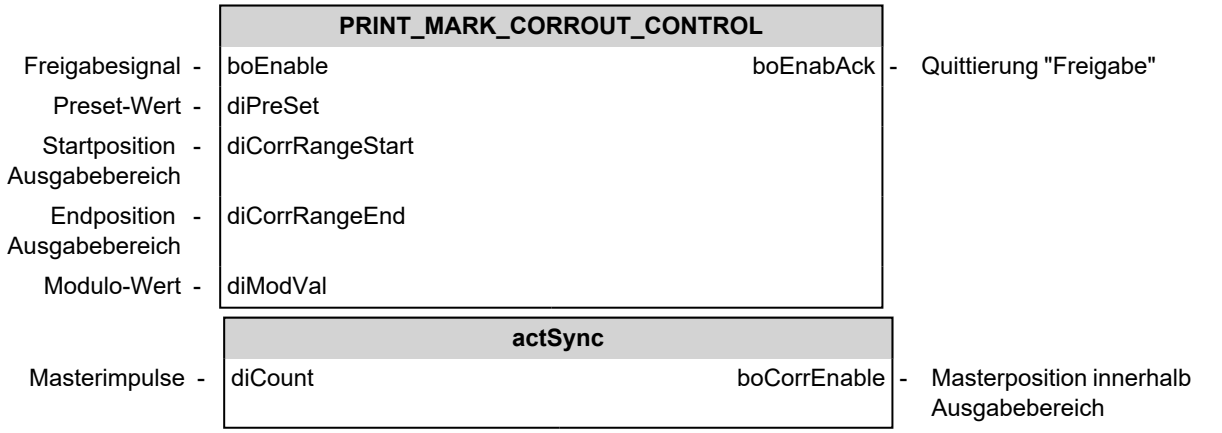
- Er detektiert, ob sich das Eingangssignal in einem definierten Bereich befindet.
  Der Bereich erstreckt sich zwischen 0 und einem Modulo-Wert.
- Der Ausgabebereich wird durch untere und obere Grenze definiert.
- Der Baustein kann durch einen Preset-Wert auf eine bestimmte Startposition gesetzt werden.
- Der Eingang des Bausteins ist durch eine Differenzstufe entkoppelt.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Wird der Baustein zur Realisierung einer Druckmarkenregelung verwendet, muss er vollständig in der synchronen Ebene aufgerufen werden.

Anwender Interface



Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diPreSet	DINT	Voreinstellwert, auf den der Ausgang gesetzt wird. Der Wert wird mit der positiven Flanke von 'boEnable' übernommen. [Inkremente]

Name	Typ	Beschreibung
diCorrRangeStart	DINT	Startposition des Bereichs [Inkremente] mit Aktion actParameterSetup bei 'boEnable' = TRUE änderbar
diCorrRangeEnd	DINT	Endposition des Bereichs [Inkremente] mit Aktion actParameterSetup bei 'boEnable' = TRUE änderbar
diModVal	DINT	Modulowert (Formatlänge) [Inkremente]

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diCount	DINT	Masterimpulse [Inkremente]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben

Ausgangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
boCorrEnable	BOOL	Masterposition im Ausgabebereich

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
PRINT_MARK_CORROUT_CONTROL	PRINT_MARK_CORROUT_CONTROL.actSync

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.2.4 PRINT_MARK_DETECT (FB)

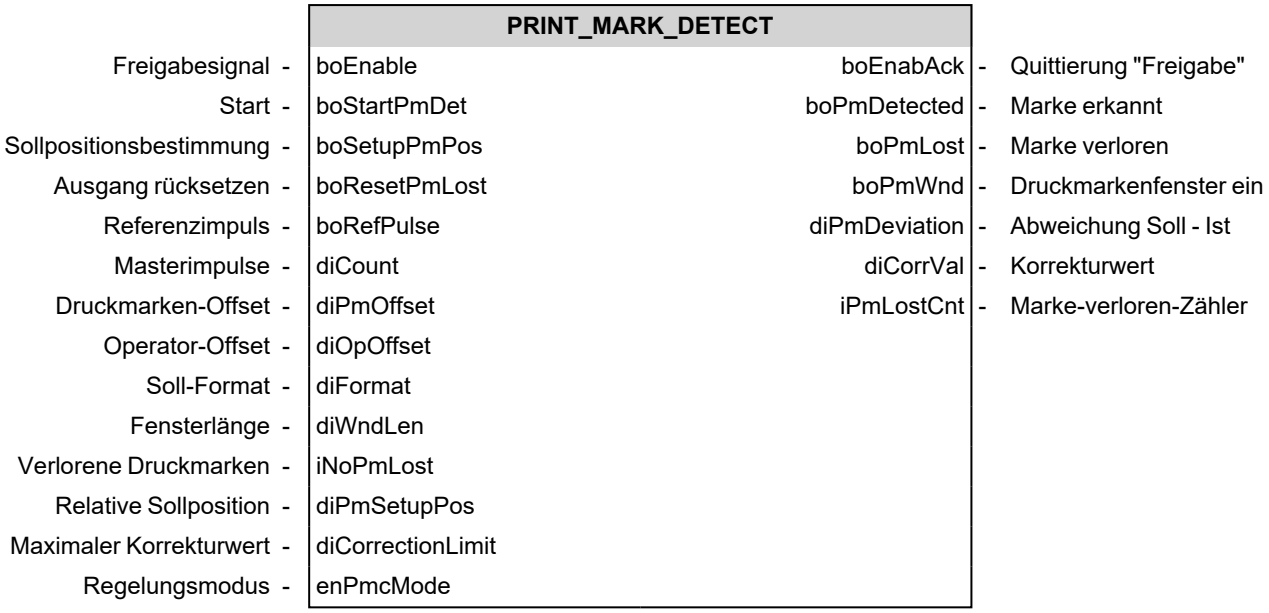
Der Funktionsbaustein 'PRINT_MARK_DETECT' realisiert folgend Funktionen:

- Markenerkennung und Markenvermessung mit Berechnung der Abweichung von Soll- und Ist-Position der Marke.
- Steuerung des Druckmarkensensors in Abhängigkeit eines Fensters um die Druckmarke.
- Überwachung auf fehlende Druckmarken.
- Simulation fehlender Druckmarken.
- Begrenzung der Korrekturausgabe pro Format.
- Die Differenz von Ist- und Sollmarkenposition wird am Ausgang bereitgestellt und kann für eine Nachregelung verwendet werden. Die Differenzen werden addiert.
- Es wird ein Offset verarbeitet, der die Marken-Sollposition verschiebt,

Der Baustein arbeitet mit positiver Zählrichtung der Eingangsimpulse und liefert positive Ausgangsimpulse.

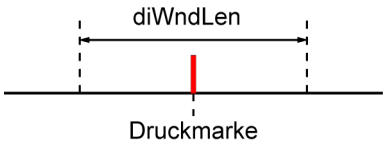
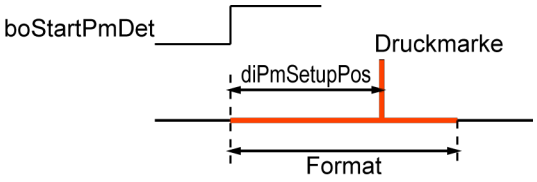
Wird der Baustein zur Realisierung einer Druckmarkenregelung verwendet, muss er in der synchronen Ebene aufgerufen werden.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boStartPmDet	BOOL	Start der Druckmarkenregelung und Bestimmung der Druckmarken-Sollposition durch Parameter 'diPmSetupPos' wenn 'boSetupPmPos' = TRUE Bestimmung der Druckmarken-Sollposition und Start der Druckmarkenregelung
boSetupPmPos	BOOL	Steuerung der Sollpositionsbestimmung der Druckmarke bei positiver Flanke von 'boStartPmDet' 'boPmSetupPos' = FALSE: Es wird die Position der nächsten Druckmarke als Soll-Position erfasst und alle nachfolgenden Marken auf diese Position geregelt. 'boPmSetupPos' = TRUE: Der Parameter 'diPmSetupPos' gibt die Soll-Position der Druckmarke relativ zur aktuellen Masterposition an. Die folgenden Druckmarken werden auf diese Position geregelt.
boResetPmLost	BOOL	Rücksetzen des Ausgangs 'boPmLost'
boRefPulse	BOOL	Referenzimpuls erkannt (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport DI_TO_COUNT
diCount	DINT	Masterimpulse [Inkremente] (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport DI_TO_COUNT
diPmOffset	DINT	Druckmarkenoffset der aktuellen Position (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport DI_TO_COUNT

Name	Typ	Beschreibung
diOpOffset	DINT	Operator-Offset. Verschiebt die Sollposition der Druckmarke. Die Verschiebung erfolgt im Korrekturbereich und wird mit einer 0→1 Flanke an 'boSetupOffset' gestartet. Sie ist unabhängig von Wertänderungen an 'diCount'. Die Verschiebegeschwindigkeit berechnet sich aus
diFormat	DINT	Formatlänge Soll-Abstand zweier Druckmarken
diWndLen	DINT	Fensterlänge: Länge vor und hinter der Soll-Druckmarkenposition, auf der der Druckmarkensensor aktiviert ist und eine Druckmarke akzeptiert wird. 
iNoPmLost	INT	Anzahl Druckmarken, die nacheinander fehlen dürfen, bevor der Ausgang 'boPMLost' (Marke verloren) gesetzt wird.
diPmSetupPos	DINT	Der Parameter gibt die Sollposition der Druckmarke zur aktuellen Masterposition bei positiver Flanke von 'boStartPmDet' an. Die erkannten Druckmarken werden auf diese Position geregelt. [Inkremente] 
diCorrectionLimit	DINT	Maximale Korrekturausgabe pro Format [Inkremente] 'diCorrectionLimit' = 0: keine Begrenzung 'diCorrectionLimit' > 0: maximaler Wert
enPmcMode	ENUM	EN_PMC_MODE Druckmarkenregelungsmode: Vorzugregler (Insetting): enPmcInsetter Registerregler: enPmcRegisterController

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boPmDetected	BOOL	Marke erkannt, Regelung aktiv
boPmLost	BOOL	Meldung "Marke verloren", wenn die an Parameter 'iNoPmLost' eingestellte Anzahl von aufeinanderfolgenden Marken nicht erkannt wurde.
boPmWnd	BOOL	Markenfenster aktiviert den Druckmarkensensor
diPmDeviation	DINT	Abweichung Soll- und Istposition der aktuellen Druckmarke
diCorrVal	DINT	Korrekturwert Summe aller bisherigen Abweichungen
iPmLostCnt	INT	Markenverloren-Zähler

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	PRINT_MARK_DETECT

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.2.5 PRINT_MARK_INSETTER_CONT (FB)

Der Funktionsbaustein 'PRINT_MARK_INSETTER_CONT' realisiert eine Druckmarkenregelung, wobei die Position des zu bearbeitende Bandes zu einem Masterformat geregelt wird (Insetting).

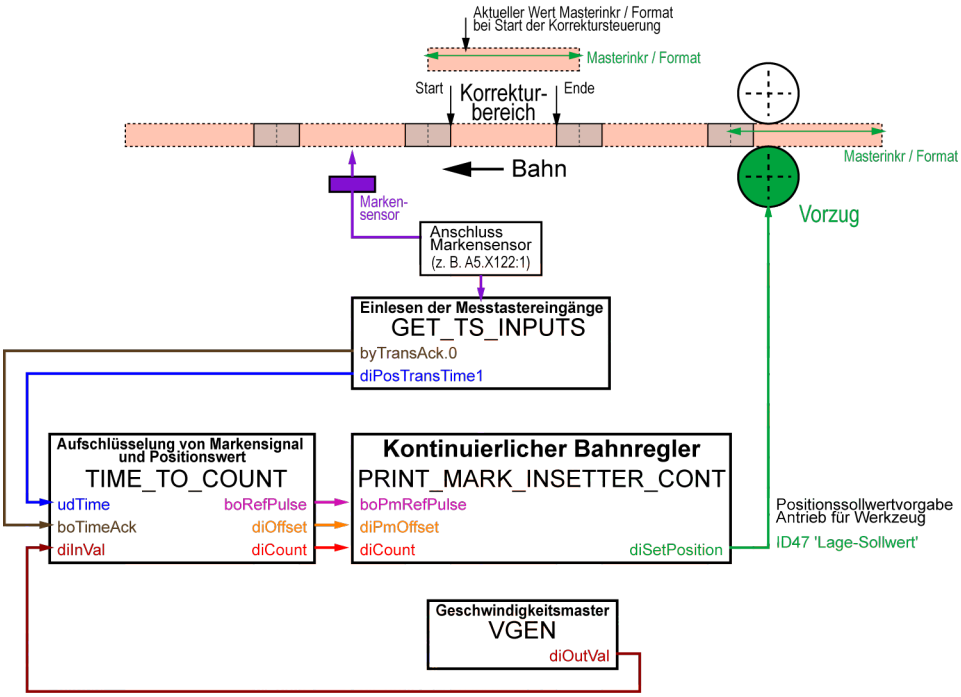
- Das zu regelnde Band bewegt sich kontinuierlich
- Es wird eine Trendabweichung der Druckmarken-Ist-Position von der Soll-Position ausgeregelt
- Der Baustein unterstützt zwei Arten der Druckmarken-Sollpositions-Erfassung.
  - Die Druckmarken-Sollposition kann auf eine absolute Position innerhalb des Druckformates festgelegt werden
  - Die erste erkannte Druckmarke legt die Sollposition fest (Siehe 'PRINT_MARK_CONTROL (FB)' auf Seite 505.)
- Steuerung des Druckmarkensensors in Abhängigkeit eines Fensters
- Überwachung der Druckmarkenerkennung
- Simulation fehlender Druckmarken
- Ausgabe des Korrekturweges als gleichmäßige Verteilung der Inkremente über einen vorgebbaren Korrekturweg
- Begrenzung der Korrekturgeschwindigkeit
- Ausgabe eines Offsets zur Verschiebung der Druckmarken-Sollposition

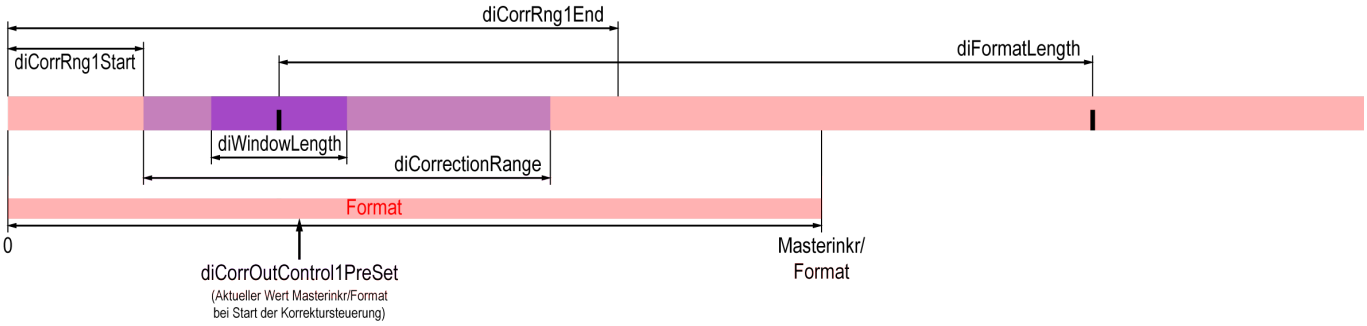
Der Baustein arbeitet intern mit positiver Zählrichtung der Eingangsimpulse und liefert positive Ausgangsimpulse.
Mit den Variablen 'boMasterNegDir' und 'boMotorNegDir' kann die Zähl- bzw. Drehrichtung angepasst werden.
Alle anderen Ausgangswerte bleiben davon unbeeinflusst.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Prinzip-Skizze





Beispielwerte

diFormatLength	10000 Inkr
diWindowLength	2500 Inkr
diCorrectionRange	70 %
diCorrRng1Start	1000 Inkr
diCorrRng1End	9000 Inkr
diCorrOutControl1PreSet	0 Inkr (keine Verschiebung)

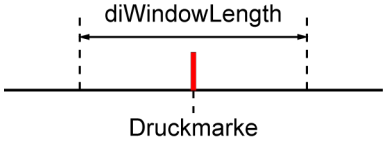
Anwender Interface

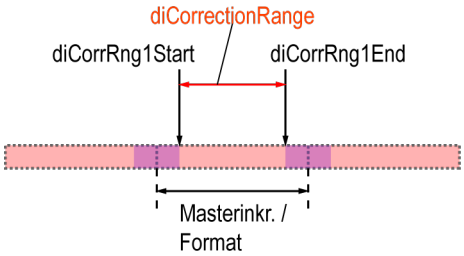
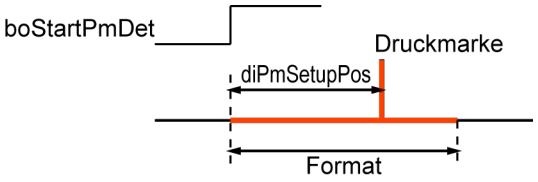
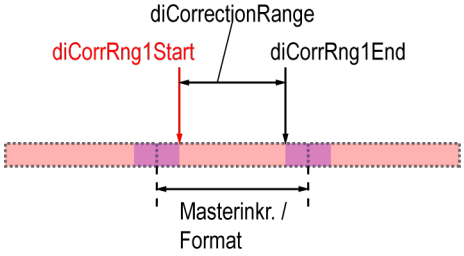
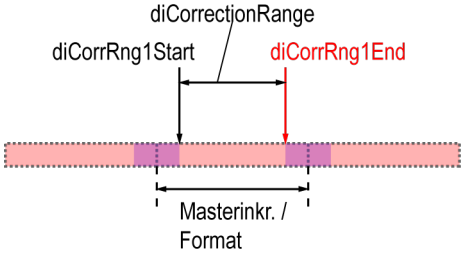
		PRINT_MARK_INSETTER_CONT			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Rücksetzen Ausgang boPmLost	-	boResetPmLost	boParameterSetupDone	-	Parametrierung beendet
Start Ausgabe Operator-Offset	-	boSetupOffset	boPmDetected	-	Marke gefunden
Start mit Druckmarken-Sollposition	-	boSetupPmPos	boPmLost	-	Marke verloren
Master zählt negativ	-	boMasterNegDir	boPmWnd	-	Fenster ein
Motorrichtung negativ	-	boMotorNegDir	diPmDeviation	-	Abweichung Soll-Ist-Position Druckmarke
Freigabe Korrekturausgabe	-	boCorrOutControlOn	diOpOffsetOut	-	Ausgegebener Operator-Offset
Operator Offset	-	diOpOffset	iActPmLost	-	Anzahl verlorener Druckmarken
Formatlänge	-	diFormatLength	boErr	-	Fehlersignal
Fensterlänge	-	diWindowLength	iErrID	-	Fehler Identnummer
Anzahl Marken bis Meldung "Marke verloren"	-	iNoPmLost			
Korrekturbereich	-	diCorrectionRange			
Druckmarken-Sollposition	-	diPmSetupPos			
Maximaler Korrekturwert pro Format	-	diCorrectionLimit			
Start der Korrekturbereich 1	-	diCorrRng1Start			
Ende der Korrekturbereich 1	-	diCorrRng1End			
Offset der Korrektursteuerung 1	-	diCorrOutControl1PreSet			
Start der Korrekturbereich 2	-	diCorrRng2Start			
Ende der Korrekturbereich 2	-	diCorrRng2End			
Offset der Korrektursteuerung 2	-	diCorrOutControl2PreSet			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

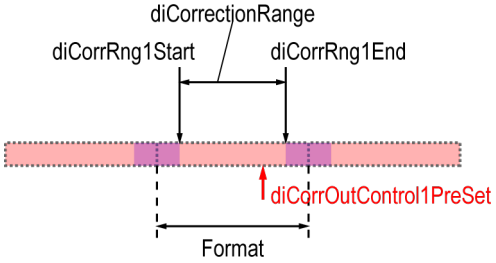
		actSync			
Start Markensuche	-	boStartPmDet			
Druckmarke erkannt	-	boPmRefPulse			
Impulse des Mastergebers	-	diCount			
Druckmarken-Offset	-	diPmOffset			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boResetPmLost	BOOL	Rücksetzen des Ausgangs 'boPmLost'
boSetupOffset	BOOL	Ausgabestart des angegebenen Werts von 'diOpOffset'. Dieser Wert dient als Offset zur Verschiebung der Druckmarken-Sollposition

Name	Typ	Beschreibung
boSetupPmPos	BOOL	Steuerung der Sollpositionsbestimmung der Druckmarke bei positiver Flanke von 'boStartPmDet' 'boPmSetupPos' = FALSE: Es wird die Position der nächsten Druckmarke als Soll-Position erfasst und alle nachfolgenden Marken auf diese Position geregelt. 'boPmSetupPos' = TRUE: Der Parameter 'diPmSetupPos' gibt die Soll-Position der Druckmarke relativ zur aktuellen Masterposition an. Die folgenden Druckmarken werden auf diese Position geregelt.
boMasterNegDir	BOOL	Negation Master-Impuls-Richtung 'boMasterNegDir' = FALSE: keine Negation 'boMasterNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.
boMotorNegDir	BOOL	Negation Motor-Drehrichtung 'boMotorNegDir' = FALSE: keine Negation 'boMotorNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.
boCorrOutControlOn	BOOL	Korrekturausgabe-Steuerung aktiv. 'boCorrOutControlOn' = FALSE: Korrekturausgabe immer aktiv 'boCorrOutControlOn' = TRUE: Korrekturausgabe erfolgt nur in dem durch die Variablen 'diCorrRng1Start' und 'diCorrRng1End' bzw. 'diCorrRng2Start' und 'diCorrRng2End' festgelegten Bereich. Beide Bereiche sind mit ODER-Funktion verknüpft. Die Bereichszählung (Modulozählung) wird mit der 0→1 Flanke an 'boCorrOutControlOn' gestartet und lässt sich durch einen Wert an 'diCorrOutControl1PreSet' bzw. 'diCorrOutControl2PreSet' entsprechend einstellen. Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn', Format/2 an 'diCorrOutControl1PreSet' einzustellen.
diOpOffset	DINT	Operator-Offset. Verschiebt die Sollposition der Druckmarke. Die Verschiebung erfolgt im Korrekturbereich und wird mit einer 0→1 Flanke an 'boSetupOffset' gestartet. Sie ist unabhängig von Wertänderungen an 'diCount'. Die Verschiebegeschwindigkeit berechnet sich aus
diFormatLength	DINT	Formatlänge Soll-Abstand zweier Druckmarken
diWindowLength	DINT	Fensterlänge Länge vor und nach der Soll-Druckmarkenposition. In diesem Bereich ist der Druckmarkensensor aktiviert. Eine Druckmarke wird akzeptiert. 

Name	Typ	Beschreibung
iNoPmLost	INT	Anzahl Druckmarken, die nacheinander fehlen dürfen, bevor der Ausgang 'boPMLost' (Marke verloren) gesetzt wird.
diCorrectionRange	DINT	Korrekturbereich Die Druckmarkenregelung versucht die Korrektur im angegebenen Bereich auszugeben. [% diFormatLength] 
diPmSetupPos	DINT	Der Parameter gibt die Sollposition der Druckmarke zur aktuellen Masterposition bei positiver Flanke von 'boStartPmDet' an. Die erkannten Druckmarken werden auf diese Position geregelt. [Inkremente] 
diCorrectionLimit	DINT	Maximale Korrekturausgabe pro Format [Inkremente] 'diCorrectionLimit' = 0: keine Begrenzung 'diCorrectionLimit' > 0: maximaler Wert
diCorrRng1Start diCorrRng2Start	DINT	Korrekturbereich 1 / 2 Startwert bezogen auf Masterinkremente / Format [Inkremente] 
diCorrRng1End diCorrRng2End	DINT	Korrekturbereich 1 / 2 Endwert bezogen auf Masterinkremente / Format [Inkremente] 

Name	Typ	Beschreibung
diCorrOutControl1PreSet diCorrOutControl2PreSet	DINT	Positionswert (Nullverschiebung) bei Start der Korrektursteuerung bezogen auf Masterinkremente / Format [Inkremente] Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn' >Format/2< einzustellen. 

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
boStartPmDet	BOOL	Start der Druckmarkenregelung und Bestimmung der Druckmarken-Sollposition durch Parameter 'diPmSetupPos' wenn 'boSetupPmPos' = TRUE
boPmRefPulse	BOOL	Markenimpuls (Referenzpuls) erkannt (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT
diCount	DINT	Masterimpulse [Inkremente] (Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT
diPmOffset	DINT	Druckmarkenoffset der aktuellen Masterposition (Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boParameterSetupDone	BOOL	Parametrierung beendet
boPmDetected	BOOL	Marke erkannt, Regelung aktiv
boPmLost	BOOL	Meldung "Marke verloren", wenn die an Parameter 'iNoPmLost' eingestellte Anzahl von aufeinanderfolgenden Marken nicht erkannt wurde.
boPmWnd	BOOL	Markenfenster aktiviert den Druckmarkensensor
diPmDeviation	DINT	Abweichung Soll- und Istposition der aktuellen Druckmarke
diOpOffsetOut	DINT	Ausgabewert beinhaltet den aktuell ausgegebenen Operatoroffset
iActPmLost	INT	Anzahl verlorener Druckmarken
boErr	DINT	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Name	Typ	Beschreibung
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben 101 diFormatLength ≤ 0 104 diCorrectionRange ≤ 0

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
PRINT_MARK_INSETTER_CONT	PRINT_MARK_INSETTER_CONT.actSync

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.2.6 PRINT_MARK_INSETTER_INTERVAL (FB)

Der Funktionsbaustein 'PRINT_MARK_INSETTER_INTERVAL' realisiert eine Druckmarkenregelung, wobei die Position des zu bearbeitende Bandes zu einem Masterformat geregelt wird (Insetting).

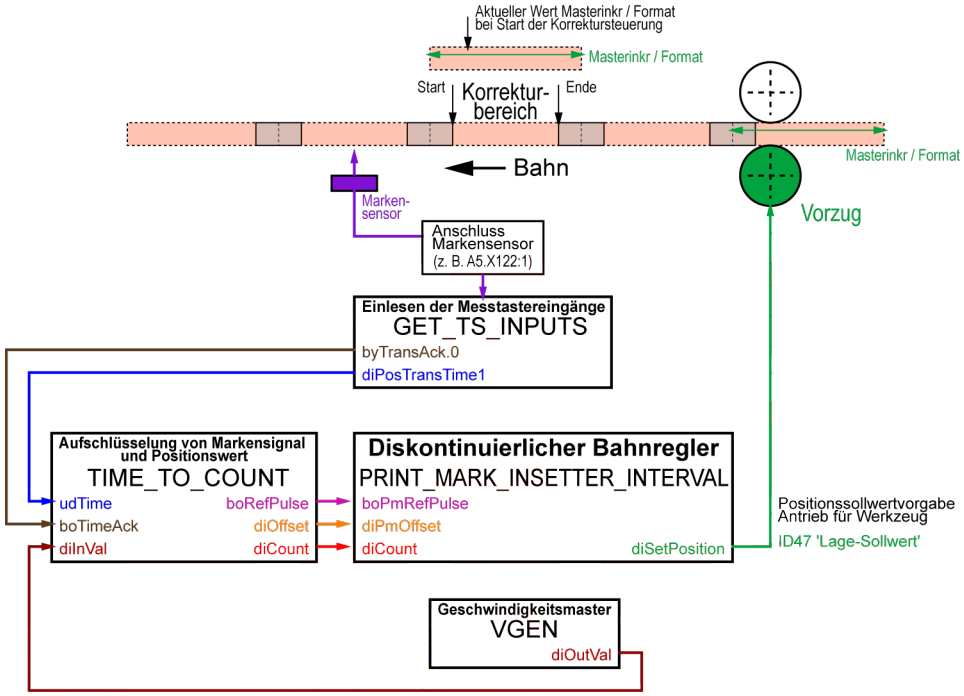
- Das zu regelnde Band bewegt sich getaktet
- Bewegungsform wählbar:
  - 45°-Gerade
  - SIN-Kurve
  - SIN²-Kurve
- Die Länge des Stillstandsbereiches ist parametrierbar
- Es wird eine Trendabweichung der Druckmarken-Ist-Position von der Soll-Position ausgeregelt
- Der Baustein realisiert zwei Arten der Druckmarken-Sollpositions-Erfassung:
  - Die Druckmarken-Sollposition kann auf eine absolute Position innerhalb des Druckformates festgelegt werden
  - Die erste erkannte Druckmarke legt die Sollposition fest (Siehe 'PRINT_MARK_CONTROL (FB)' auf Seite 505.)
- Steuerung des Druckmarkensensors in Abhängigkeit eines Fensters
- Überwachung der Druckmarkenerkennung
- Simulation fehlender Druckmarken
- Ausgabe des Korrekturweges als gleichmäßige Verteilung der Inkremente über einen vorgebbaren Korrekturweg
- Begrenzung der Korrekturgeschwindigkeit
- Ausgabe eines Offsets zur Verschiebung der Druckmarken-Sollposition

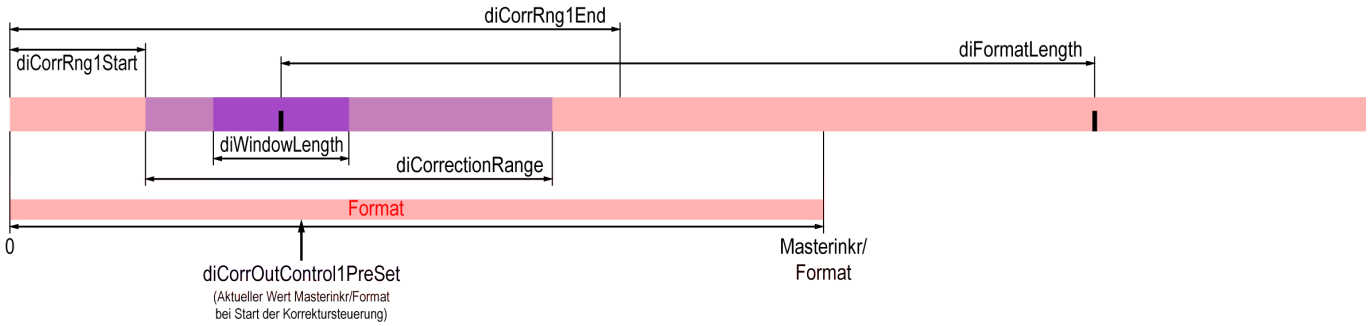
Der Baustein arbeitet intern mit positiver Zählrichtung der Eingangsimpulse und liefert positive Ausgangsimpulse.
Mit den Variablen 'boMasterNegDir' und 'boMotorNegDir' kann die Zähl- bzw. Drehrichtung angepasst werden. Alle anderen Ausgangswerte bleiben davon unbeeinflusst.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Prinzipskizze





Beispielwerte

diFormatLength	10000 Inkr
diWindowLength	2500 Inkr
diCorrectionRange	70 %
diCorrRng1Start	1000 Inkr
diCorrRng1End	9000 Inkr
diCorrOutControl1PreSet	0 Inkr (keine Verschiebung)

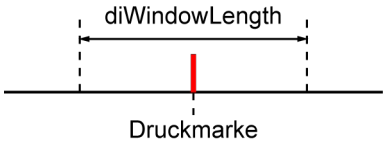
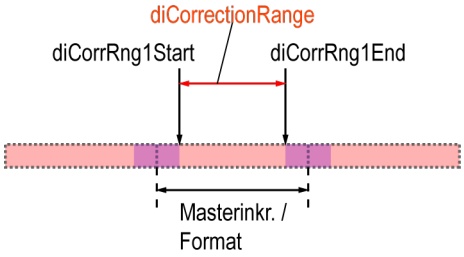
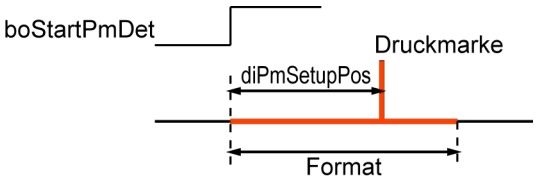
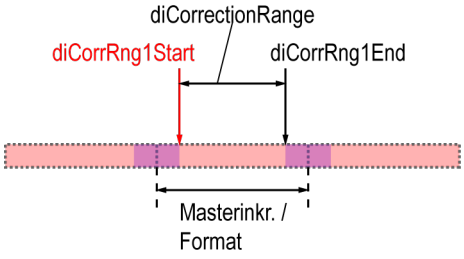
Anwender Interface

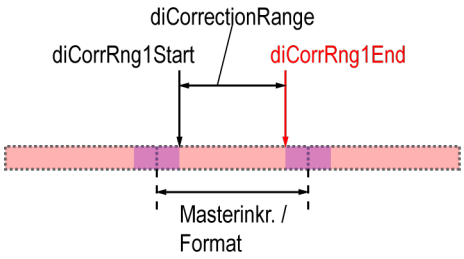
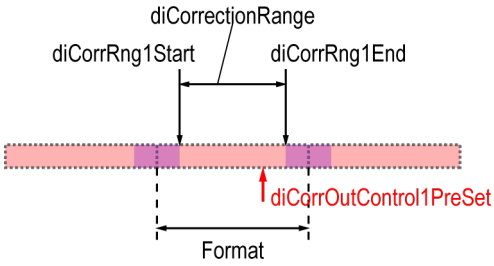
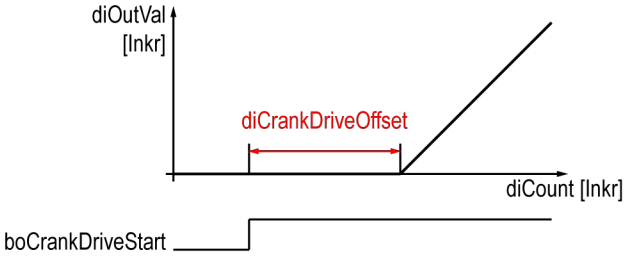
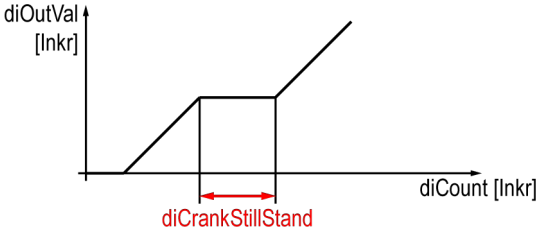
		PRINT_MARK_INSETTER_INTERVAL			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Rücksetzen Ausgang boPmLost	-	boResetPmLost	boParameterSetupDone	-	Parametrierung beendet
Start Ausgabe Operator-Offset	-	boSetupOffset	boPmDetected	-	Marke gefunden
Start mit Druckmarken-Sollposition	-	boSetupPmPos	boPmLost	-	Marke verloren
Master zählt negativ	-	boMasterNegDir	boPmWnd	-	Fenster ein
Motorrichtung negativ	-	boMotorNegDir	diPmDeviation	-	Abweichung Soll-Ist-Position Druckmarke
Freigabe Korrekturausgabe	-	boCorrOutControlOn	diOpOffsetOut	-	Ausgegebener Operator-Offset
Kurbelfunktion Start am Nullpunkt	-	boCrankDriveStart	iActPmLost		Anzahl verlorener Druckmarken
Operator Offset	-	diOpOffset	boErr	-	Fehlersignal
Formatlänge	-	diFormatLength	iErrID	-	Fehler Identnummer
Fensterlänge	-	diWindowLength			
Anzahl Marken bis Meldung "Marke verloren"	-	iNoPmLost			
Korrekturbereich	-	diCorrectionRange			
Druckmarken-Sollposition	-	diPmSetupPos			
Maximaler Korrekturwert pro Format	-	diCorrectionLimit			
Start der Korrekturbereich 1	-	diCorrRng1Start			
Ende der Korrekturbereich 1	-	diCorrRng1End			
Offset der Korrektursteuerung 1	-	diCorrOutControl1PreSet			
Start der Korrekturbereich 2	-	diCorrRng2Start			
Ende der Korrekturbereich 2	-	diCorrRng2End			
Offset der Korrektursteuerung 2	-	diCorrOutControl2PreSet			
Kurbelfunktion Kurvenform	-	enCrankShape			
Kurbelfunktion Start-Offset	-	diCrancDriveOffset			
Kurbelfunktion Stillstandsbereich	-	diCrankStillStand			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

		actSync			
Start Markensuche	-	boStartPmDet			
Druckmarke erkannt	-	boPmRefPulse			
Impulse des Mastergebers	-	diCount			
Druckmarken-Offset	-	diPmOffset			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boResetPmLost	BOOL	Rücksetzen des Ausgangs 'boPmLost'
boSetupOffset	BOOL	Ausgabestart des angegebenen Werts von 'diOpOffset'. Dieser Wert dient als Offset zur Verschiebung der Druckmarken-Sollposition
boSetupPmPos	BOOL	Steuerung der Sollpositionsbestimmung der Druckmarke bei positiver Flanke von 'boStartPmDet' 'boPmSetupPos' = FALSE: Es wird die Position der nächsten Druckmarke als Soll-Position erfasst und alle nachfolgenden Marken auf diese Position geregelt. 'boPmSetupPos' = TRUE: Der Parameter 'diPmSetupPos' gibt die Soll-Position der Druckmarke relativ zur aktuellen Masterposition an. Die folgenden Druckmarken werden auf diese Position geregelt.
boMasterNegDir	BOOL	Negation Master-Impuls-Richtung 'boMasterNegDir' = FALSE: keine Negation 'boMasterNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.
boMotorNegDir	BOOL	Negation Motor-Drehrichtung 'boMotorNegDir' = FALSE: keine Negation 'boMotorNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.
boCorrOutControlOn	BOOL	Korrekturausgabe-Steuerung aktiv. 'boCorrOutControlOn' = FALSE: Korrekturausgabe immer aktiv 'boCorrOutControlOn' = TRUE: Korrekturausgabe erfolgt nur in dem durch die Variablen 'diCorrRng1Start' und 'diCorrRng1End' bzw. 'diCorrRng2Start' und 'diCorrRng2End' festgelegten Bereich. Beide Bereiche sind mit ODER-Funktion verknüpft. Die Bereichszählung (Modulozählung) wird mit der 0→1 Flanke an 'boCorrOutControlOn' gestartet und lässt sich durch einen Wert an 'diCorrOutControl1PreSet' bzw. 'diCorrOutControl2PreSet' entsprechend einstellen. Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn', Format/2 an 'diCorrOutControl1PreSet' einzustellen.
diOpOffset	DINT	Operator-Offset. Verschiebt die Sollposition der Druckmarke. Die Verschiebung erfolgt im Korrekturbereich und wird mit einer 0→1 Flanke an 'boSetupOffset' gestartet. Sie ist unabhängig von Wertänderungen an 'diCount'. Die Verschiebegeschwindigkeit berechnet sich aus

Name	Typ	Beschreibung
diFormatLength	DINT	Formatlänge Soll-Abstand zweier Druckmarken
diWindowLength	DINT	Fensterlänge Länge vor und nach der Soll-Druckmarkenposition. In diesem Bereich ist der Druckmarkensensor aktiviert. Eine Druckmarke wird akzeptiert. 
iNoPmLost	INT	Anzahl Druckmarken, die nacheinander fehlen dürfen, bevor der Ausgang 'boPMLost' (Marke verloren) gesetzt wird.
diCorrectionRange	DINT	Korrekturbereich Die Druckmarkenregelung versucht die Korrektur im angegebenen Bereich auszugeben. [% diFormatLength] 
diPmSetupPos	DINT	Der Parameter gibt die Sollposition der Druckmarke zur aktuellen Masterposition bei positiver Flanke von 'boStartPmDet' an. Die erkannten Druckmarken werden auf diese Position geregelt. [Inkremente] 
diCorrectionLimit	DINT	Maximale Korrekturausgabe pro Format [Inkremente] 'diCorrectionLimit' = 0: keine Begrenzung 'diCorrectionLimit' > 0: maximaler Wert
diCorrRng1Start diCorrRng2Start	DINT	Korrekturbereich 1 / 2 Startwert bezogen auf Masterinkremente / Format [Inkremente] 

Name	Typ	Beschreibung
diCorrRng1End diCorrRng2End	DINT	Korrekturbereich 1 / 2 Endwert bezogen auf Masterinkremente / Format [Inkremente] 
diCorrOutControl1PreSet diCorrOutControl2PreSet	DINT	Positionswert (Nullverschiebung) bei Start der Korrektursteuerung bezogen auf Masterinkremente / Format [Inkremente] Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn' >Format/2< einzustellen. 
enCrankShape	ENUM	EN_CRANK_DRIVE_SHAPE Kurvenform der Kurbelfunktion enCrankShapeLinear: linearer Verlauf 45° enCrankShapeSine: SINUS-Funktion enCrankShapeSquareSine: SINUS^2-Funktion
diCrancDriveOffset	DINT	Kurbelfunktion Start-Offset. Nach Start der Kurbelfunktion wartet die Funktion, bis der Master den Offset zurückgelegt hat und beginnt dann 
diCrankStillStand	DINT	Kurbelfunktion Stillstandsbereich 

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
boStartPmDet	BOOL	Start der Druckmarkenregelung und Bestimmung der Druckmarken-Sollposition durch Parameter 'diPmSetupPos' wenn 'boSetupPmPos' = TRUE

Name	Typ	Beschreibung
boCrankDriveStart	BOOL	Start der Kurbelfunktion Wertausgabe am Ausgang 'diOutVal' entsprechend der am Eingang 'enCrankShape' gewählten Kurvenform. Die Wertausgabe erfolgt bei Wertänderung an 'diCount'
boPmRefPulse	BOOL	Markenimpuls (Referenzpuls) erkannt (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT
diCount	DINT	Masterimpulse [Inkremente] (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport DI_TO_COUNT
diPmOffset	DINT	Druckmarkenoffset der aktuellen Masterposition (Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boParameterSetupDone	BOOL	Parametrierung beendet
boPmDetected	BOOL	Marke erkannt, Regelung aktiv
boPmLost	BOOL	Meldung "Marke verloren", wenn die an Parameter 'iNoPmLost' eingestellte Anzahl von aufeinanderfolgenden Marken nicht erkannt wurde.
boPmWnd	BOOL	Markenfenster aktiviert den Druckmarkensensor
diPmDeviation	DINT	Abweichung Soll- und Istposition der aktuellen Druckmarke
diOpOffsetOut	DINT	Ausgabewert beinhaltet den aktuell ausgegebenen Operatoroffset
iActPmLost	INT	Anzahl verlorener Druckmarken
boErr	DINT	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben 1 ... 99 Fehler siehe Baustein 'CAM_PROF' (AmkBase.lib) 100 Falscher Mode in Parameter 'enCrankShape' 101 diFormatLength ≤ 0 102 diCrankStillStand zu groß 103 diCrancDriveOffset < 0 104 diCorrectionRange ≤ 0

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
PRINT_MARK_INSETTER_INTERVAL	PRINT_MARK_INSETTER_INTERVAL.actSync

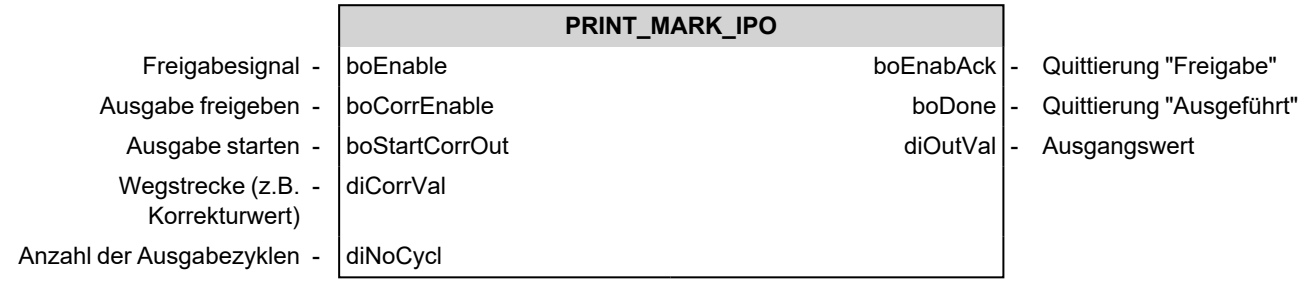
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.2.7 PRINT_MARK_IPO (FB)

Der Funktionsbaustein 'PRINT_MARK_IPO' realisiert die Ausgabe einer Wegstrecke in einzelnen Schritten. Die Schrittweite pro Zyklus lässt sich dabei vorgeben.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boCorrEnable	BOOL	Freigabe der Korrekturausgabe. Durch diesen Eingang lässt sich die Ausgabe steuern. Mit 'boCorrEnable' = TRUE ist die Wertausgabe am Ausgang 'diOutVal' freigegeben. Mit 'boCorrEnable' = FALSE ist die Ausgabe am Ausgang 'diOutVal' gesperrt. Dadurch ist es möglich, eine Korrekturausgabe nur dann auszugeben, wenn es erwünscht ist. (z.B. in bestimmten Bereichen)
boStartCorrOut	BOOL	Start der Wertausgabe (Korrekturausgabe)
diCorrVal	DINT	Länge der Wegstrecke (Korrekturwert) [Inkremente]
diNoCycl	DINT	Anzahl der Ausgabezyklen

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde Ausgabe abgeschlossen
diOutVal	DINT	Ausgangswert [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	PRINT_MARK_IPO

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

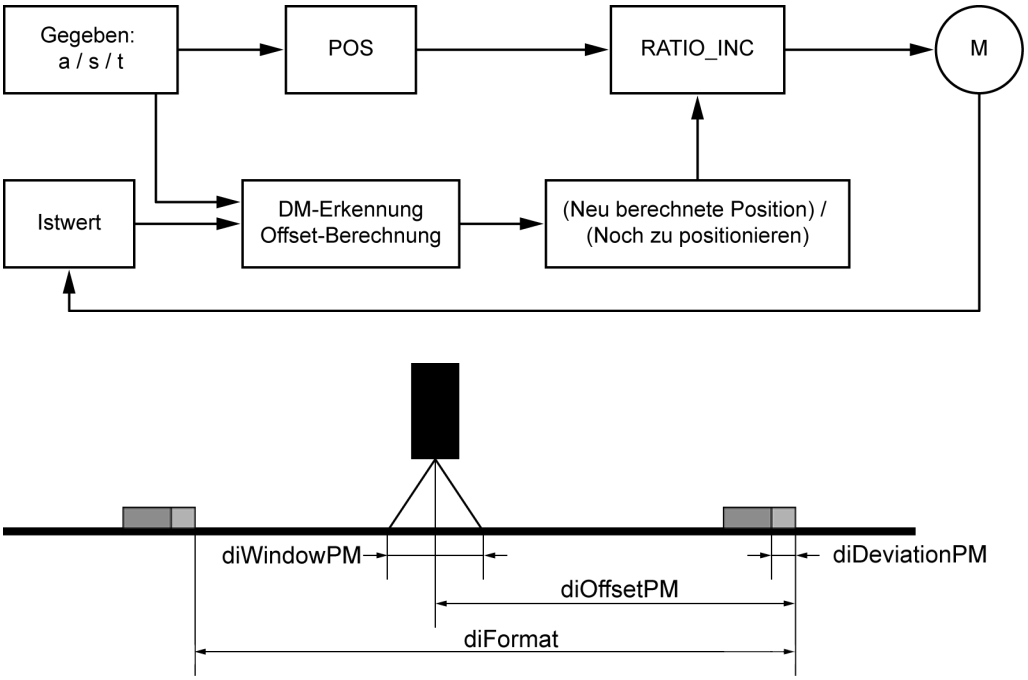
4.2.1.2.2.8 PRINT_MARK_POSITION_TO_THE_MARK (FB)

Der Funktionsbaustein 'PRINT_MARK_POSITION_TO_THE_MARK' positioniert eine AMK-Achse relativ.

Während der Positionierung wird nach einer Druckmarke innerhalb des Fensters geschaut. Wird die Druckmarke gefunden, regelt sich die Positionierung danach aus.

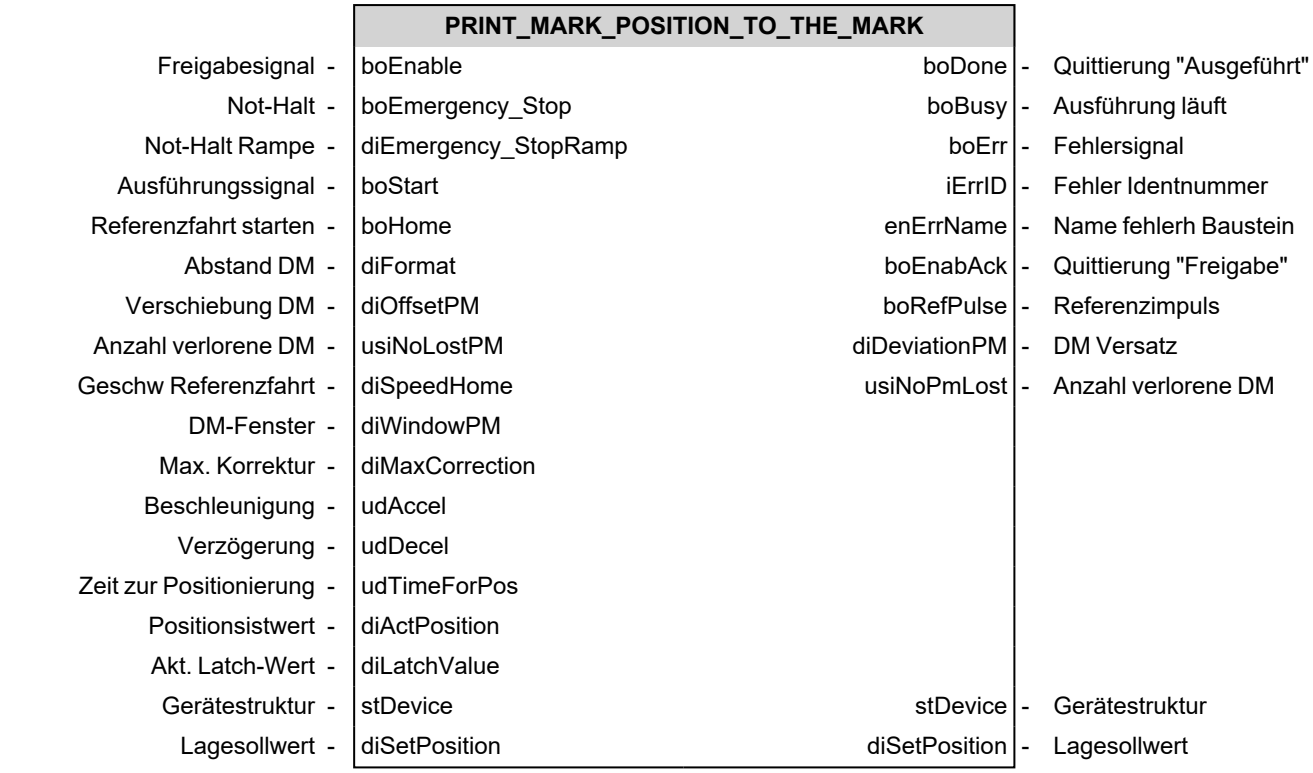
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Schema



- Der Funktionsbaustein 'RATIO_INC' streckt oder staucht entsprechend der DM-Position die eigentliche Positionierung
- So ändert sich für den Funktionsbaustein POS nichts.
- Die Positionierung wird in vorgegebener Zeit beendet
- Fehlt die DM, wird normal die Formatlänge abgefahren

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boEmergency_Stop	BOOL	Not-Halt: Der Sollwert der Geschwindigkeit wird an der Not-Halt-Rampe zu Null heruntergefahren. Ein einmal eingeleiteter Not-Halt kann nicht abgebrochen werden.
diEmergency_StopRamp	DINT	Not-Halt-Rampe: Rampenzeit, in der der Antrieb von der aktuellen Geschwindigkeit auf Null abgebremst wird. [ms]
boStart	BOOL	Mit einer positiven Flanke startet die Ausführung des Bausteins.
boHome	BOOL	Referenzfahrt Freigabesignal: Mit einer positiven Flanke startet die Funktion Referenzpunktfahrt. Solange 'boHome' = TRUE ist, wird die Referenzfahrt ausgeführt. Mit einer negativen Flanke 'boHome' = FALSE wird die laufende Referenzierung abgebrochen oder die abgeschlossene Referenzierung beendet. Achse fährt auf DM + Offset
diFormat	DINT	Formatlänge Soll-Abstand zweier Druckmarken
diOffsetPM	DINT	Weg nach DM (Abstand DM-Sensor und Stoppposition) Einheit: Inkremente Standardwert: 10240

Name	Typ	Beschreibung
usiNoLostPM	USINT	Sobald die vorgegebene Anzahl an Marken verloren geht wird eine Fehlermeldung generiert. Vorgabe 0 = Fehlermeldung wird nie generiert Einheit: Inkremente Standardwert: 10
diSpeedHome	DINT	Geschwindigkeit der Referenzfahrt [Inkremente/s]
diWindowPM	DINT	Fenster in der die Marken zum Erkennen liegen muss (+/- Zielposition) Einheit: Inkremente Standardwert: 20480
diMaxCorrection	DINT	Maximaler Korrekturwert Vorgabe 0 = Der ermittelter Korrekturwert wird immer ohne Begrenzung verrechnet. Einheit: Inkremente Standardwert: 20480
udAccel	UDINT	Beschleunigung, mit der auf die Sollgeschwindigkeit gefahren wird
udDecel	UDINT	Verzögerung, mit der eine niedrigere Sollgeschwindigkeit angefahren wird
udTimeForPos	UDINT	Zeit in der Positionierung abgeschlossen sein muss. Einheit: ms Standardwert: 2000
diActPosition	DINT	Aktueller Lageistwert [Inkremente] ID51 'Lage Istwert'
diLatchValue	DINT	Aktueller Latchwert [Inkremente] ID130 'Messwert 1 positive Flanke'

Ausgangsvariablen

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>boEnabAck</td><td>BOOL</td><td>Quittierung: Funktionsbaustein ist initialisiert und freigegeben</td></tr>
<tr><td>boBusy</td><td>BOOL</td><td>Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.</td></tr>
<tr><td>boErr</td><td>BOOL</td><td>Der Funktionsbaustein befindet sich im Zustand Fehler
<table>
<tr><td>FALSE</td><td>Kein Fehler (Kommandierung zulässig oder Warnung)</td></tr>
<tr><td>TRUE</td><td>Fehler</td></tr>
</table></td></tr>
<tr><td>iErrID</td><td>INT</td><td>Fehler Identnummer: Diagnosenummer wird ausgegeben
<table>
<tr><td colspan="2">iErrID = 0</td><td>Kein Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = TRUE</td><td>Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = FALSE</td><td>Warnung</td></tr>
</table>
enErrName = 0:
<table>
<tr><td>0</td><td>Kein Fehler</td></tr>
<tr><td>1</td><td>Nulleingabe, benötigte Werte haben „0“ als Übergabewert</td></tr>
<tr><td>2</td><td>Rampe (Fahrprofil) mathematisch nicht möglich (z. B. a,d zu klein für Zeit)</td></tr>
<tr><td>3</td><td>Druckmarke verloren</td></tr>
<tr><td>4</td><td>diOffsetPM> diFormat</td></tr>
<tr><td>5</td><td>Beim Referenzieren keine DM gefunden, Suche abgebrochen</td></tr>
</table>
enErrName ≠ 0:
Fehlernummer siehe Baustein (enErrName)</td></tr>
</table>

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein** / **Bibliothek** RATIO_INC_1 / AmkBase.lib POS_1 / AmkBase.lib
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boRefPulse	BOOL	Für einen Zyklus lang boRefPuls = TRUE, wenn sich der Latchwert geändert hat.
diDeviationPM	DINT	Abweichung zwischen Ist- und Soll-Position der zuletzt gefundenen DM. Einheit: Inkremente
usiNoPmLost	USINT	Anzahl der verlorenen Druckmarken

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	PRINT_MARK_POSITION_TO_THE_MARK

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.2.9 PRINT_MARK_REGISTER_CONTROL_CONTINUOUS (FB)

Der Funktionsbaustein 'PRINT_MARK_REGISTER_CONTROL_CONTINUOUS' realisiert eine Druckmarkenregelung, wobei die Position des bearbeitenden Aggregats (z.B. Messer, Magnetstanze) zu einem Masterformat geregelt wird (Registerregler).

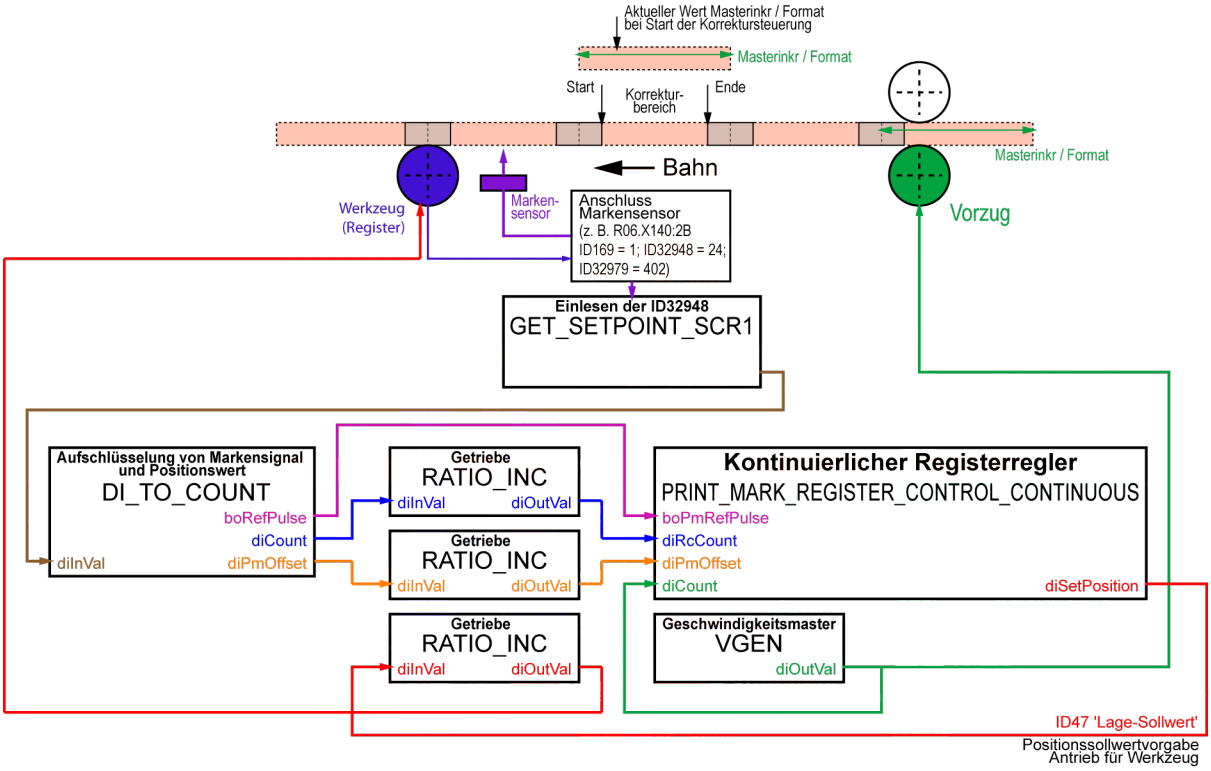
- Das Bearbeitungs-Aggregat bewegt sich kontinuierlich
- Masterformat und Umfang des bearbeitenden Aggregats müssen gleich sein
- Es wird eine Trendabweichung der Druckmarken-Ist-Position von der Soll-Position ausgeregelt
- Der Baustein realisiert zwei Arten der Druckmarken-Sollpositions-Erfassung:
  - Die Druckmarken-Sollposition kann auf eine absolute Position innerhalb des Druckformates festgelegt werden
  - Die erste erkannte Druckmarke legt die Sollposition fest (Siehe 'PRINT_MARK_CONTROL (FB)' auf Seite 505.)
- Steuerung des Druckmarkensensors in Abhängigkeit eines Fensters
- Überwachung der Druckmarkenerkennung
- Simulation fehlender Druckmarken
- Ausgabe des Korrekturweges als gleichmäßige Verteilung der Inkremente über einen vorgebbaren Korrekturweg
- Begrenzung der Korrekturgeschwindigkeit
- Ausgabe eines Offsets zur Verschiebung der Druckmarken-Sollposition

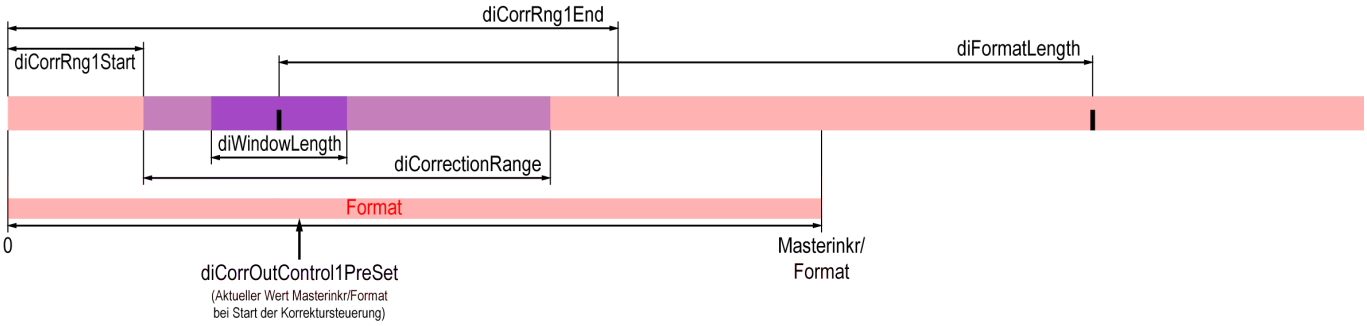
Der Baustein arbeitet intern mit positiver Zählrichtung der Eingangsimpulse und liefert positive Ausgangsimpulse. Mit den Variablen 'boMasterNegDir' und 'boMotorNegDir' kann die Zähl- bzw. Drehrichtung angepasst werden. Alle anderen Ausgangswerte bleiben davon unbeeinflusst.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Prinzipskizze





Beispielwerte

diFormatLength	10000 Inkr
diWindowLength	2500 Inkr
diCorrectionRange	70 %
diCorrRng1Start	1000 Inkr
diCorrRng1End	9000 Inkr
diCorrOutControl1PreSet	0 Inkr (keine Verschiebung)

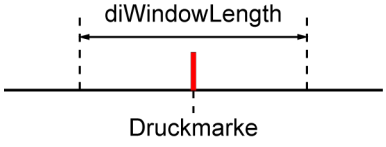
Anwender Interface

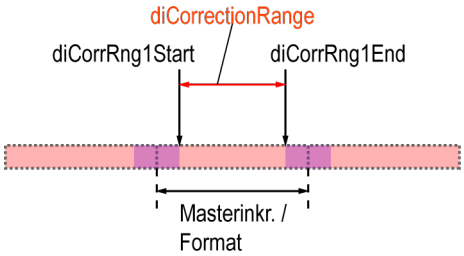
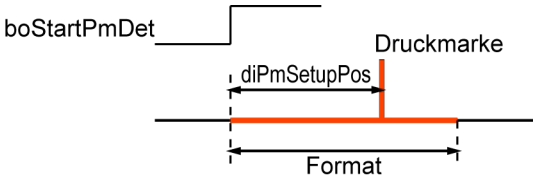
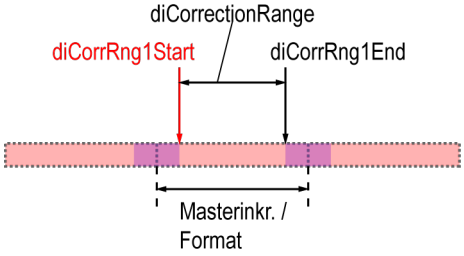
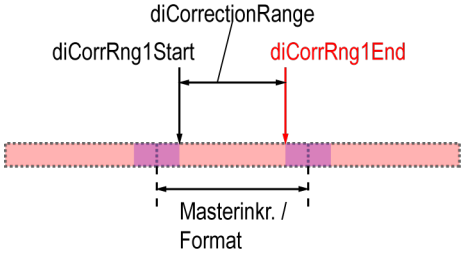
		PRINT_MARK_REGISTER_CONTROL_CONTINUOUS			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Rücksetzen Ausgang boPmLost	-	boResetPmLost	boParameterSetupDone	-	Parametrierung beendet
Start Ausgabe Operator-Offset	-	boSetupOffset	boPmDetected	-	Marke gefunden
Start mit Druckmarken-Sollposition	-	boSetupPmPos	boPmLost	-	Marke verloren
Master zählt negativ	-	boMasterNegDir	boPmWnd	-	Fenster ein
Motorrichtung negativ	-	boMotorNegDir	diPmDeviation	-	Abweichung Soll-Ist-Position Druckmarke
Freigabe Korrekturausgabe	-	boCorrOutControlOn	diOpOffsetOut	-	Ausgegebener Operator-Offset
Operator Offset	-	diOpOffset	iActPmLost	-	Anzahl verlorener Druckmarken
Formatlänge	-	diFormatLength	boErr	-	Fehlersignal
Fensterlänge	-	diWindowLength	iErrID	-	Fehler Identnummer
Anzahl Marken bis Meldung "Marke verloren"	-	iNoPmLost			
Korrekturbereich	-	diCorrectionRange			
Druckmarken-Sollposition	-	diPmSetupPos			
Maximaler Korrekturwert pro Format	-	diCorrectionLimit			
Start der Korrekturbereich 1	-	diCorrRng1Start			
Ende der Korrekturbereich 1	-	diCorrRng1End			
Offset der Korrektursteuerung 1	-	diCorrOutControl1PreSet			
Start der Korrekturbereich 2	-	diCorrRng2Start			
Ende der Korrekturbereich 2	-	diCorrRng2End			
Offset der Korrektursteuerung 2	-	diCorrOutControl2PreSet			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

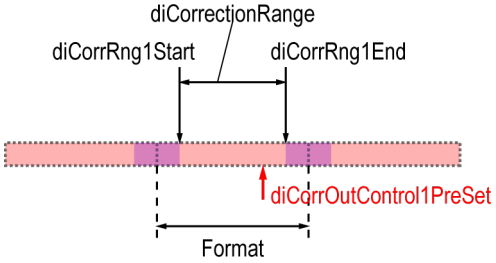
		actSync			
Start Markensuche	-	boStartPmDet			
Druckmarke erkannt	-	boPmRefPulse			
Impulse des Mastergebers	-	diCount			
Druckmarken-Offset	-	diPmOffset			
Registerregler-Impulse	-	diRcCount			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boResetPmLost	BOOL	Rücksetzen des Ausgangs 'boPmLost'
boSetupOffset	BOOL	Ausgabestart des angegebenen Werts von 'diOpOffset'. Dieser Wert dient als Offset zur Verschiebung der Druckmarken-Sollposition

Name	Typ	Beschreibung
boSetupPmPos	BOOL	Steuerung der Sollpositionsbestimmung der Druckmarke bei positiver Flanke von 'boStartPmDet' 'boPmSetupPos' = FALSE: Es wird die Position der nächsten Druckmarke als Soll-Position erfasst und alle nachfolgenden Marken auf diese Position geregelt. 'boPmSetupPos' = TRUE: Der Parameter 'diPmSetupPos' gibt die Soll-Position der Druckmarke relativ zur aktuellen Masterposition an. Die folgenden Druckmarken werden auf diese Position geregelt.
boMasterNegDir	BOOL	Negation Master-Impuls-Richtung 'boMasterNegDir' = FALSE: keine Negation 'boMasterNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.
boMotorNegDir	BOOL	Negation Motor-Drehrichtung 'boMotorNegDir' = FALSE: keine Negation 'boMotorNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.
boCorrOutControlOn	BOOL	Korrekturausgabe-Steuerung aktiv. 'boCorrOutControlOn' = FALSE: Korrekturausgabe immer aktiv 'boCorrOutControlOn' = TRUE: Korrekturausgabe erfolgt nur in dem durch die Variablen 'diCorrRng1Start' und 'diCorrRng1End' bzw. 'diCorrRng2Start' und 'diCorrRng2End' festgelegten Bereich. Beide Bereiche sind mit ODER-Funktion verknüpft. Die Bereichszählung (Modulozählung) wird mit der 0→1 Flanke an 'boCorrOutControlOn' gestartet und lässt sich durch einen Wert an 'diCorrOutControl1PreSet' bzw. 'diCorrOutControl2PreSet' entsprechend einstellen. Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn', Format/2 an 'diCorrOutControl1PreSet' einzustellen.
diOpOffset	DINT	Operator-Offset. Verschiebt die Sollposition der Druckmarke. Die Verschiebung erfolgt im Korrekturbereich und wird mit einer 0→1 Flanke an 'boSetupOffset' gestartet. Sie ist unabhängig von Wertänderungen an 'diCount'. Die Verschiebegeschwindigkeit berechnet sich aus
diFormatLength	DINT	Formatlänge Soll-Abstand zweier Druckmarken
diWindowLength	DINT	Fensterlänge Länge vor und nach der Soll-Druckmarkenposition. In diesem Bereich ist der Druckmarkensensor aktiviert. Eine Druckmarke wird akzeptiert. 

Name	Typ	Beschreibung
iNoPmLost	INT	Anzahl Druckmarken, die nacheinander fehlen dürfen, bevor der Ausgang 'boPMLost' (Marke verloren) gesetzt wird.
diCorrectionRange	DINT	Korrekturbereich Die Druckmarkenregelung versucht die Korrektur im angegebenen Bereich auszugeben. [% diFormatLength] 
diPmSetupPos	DINT	Der Parameter gibt die Sollposition der Druckmarke zur aktuellen Masterposition bei positiver Flanke von 'boStartPmDet' an. Die erkannten Druckmarken werden auf diese Position geregelt. [Inkremente] 
diCorrectionLimit	DINT	Maximale Korrekturausgabe pro Format [Inkremente] 'diCorrectionLimit' = 0: keine Begrenzung 'diCorrectionLimit' > 0: maximaler Wert
diCorrRng1Start diCorrRng2Start	DINT	Korrekturbereich 1 / 2 Startwert bezogen auf Masterinkremente / Format [Inkremente] 
diCorrRng1End diCorrRng2End	DINT	Korrekturbereich 1 / 2 Endwert bezogen auf Masterinkremente / Format [Inkremente] 

Name	Typ	Beschreibung
diCorrOutControl1PreSet diCorrOutControl2PreSet	DINT	Positionswert (Nullverschiebung) bei Start der Korrektursteuerung bezogen auf Masterinkremente / Format [Inkremente] Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn' >Format/2< einzustellen. 

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
boStartPmDet	BOOL	Start der Druckmarkenregelung und Bestimmung der Druckmarken-Sollposition durch Parameter 'diPmSetupPos' wenn 'boSetupPmPos' = TRUE
boPmRefPulse	BOOL	Markenimpuls (Referenzpuls) erkannt (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT
diCount	DINT	Masterimpulse [Inkremente] (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport DI_TO_COUNT
diPmOffset	DINT	Druckmarkenoffset der aktuellen Masterposition (Siehe DokumentSoftwarebeschreibungAmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT
diRcCount	DINT	Registerregler-Impulse Impulse des Werkzeugs [Inkr]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boParameterSetupDone	BOOL	Parametrierung beendet
boPmDetected	BOOL	Marke erkannt, Regelung aktiv
boPmLost	BOOL	Meldung "Marke verloren", wenn die an Parameter 'iNoPmLost' eingestellte Anzahl von aufeinanderfolgenden Marken nicht erkannt wurde.
boPmWnd	BOOL	Markenfenster aktiviert den Druckmarkensensor
diPmDeviation	DINT	Abweichung Soll- und Istposition der aktuellen Druckmarke
diOpOffsetOut	DINT	Ausgabewert beinhaltet den aktuell ausgegebenen Operatoroffset
iActPmLost	INT	Anzahl verlorener Druckmarken
boErr	DINT	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Name	Typ	Beschreibung
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben 101 diFormatLength ≤ 0 104 diCorrectionRange ≤ 0

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
PRINT_MARK_REGISTER_CONTROL_CONTINUOUS	PRINT_MARK_REGISTER_CONTROL_CONTINUOUS.actSync

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.2.10 PRINT_MARK_REGISTER_CONTROL_DISCONT (FB)

Der Funktionsbaustein 'PRINT_MARK_REGISTER_CONTROL_DISCONT' realisiert eine Druckmarkenregelung, wobei die Position des bearbeitenden Aggregats (z.B. Messer, Magnetstanze) zu einem Masterformat (z.B. Beutellänge) geregelt wird (Registerregler).

Masterformat und Umfang des bearbeitenden Aggregats brauchen nicht gleich sein. Im Materialeingriff (z.B.Schnitt) bewegt sich das Aggregat synchron zur Bahn. Nach der Bearbeitung erfolgt eine entsprechende Ausgleichsbewegung des Aggregates. Die Bewegung erfolgt diskontinuierlich.

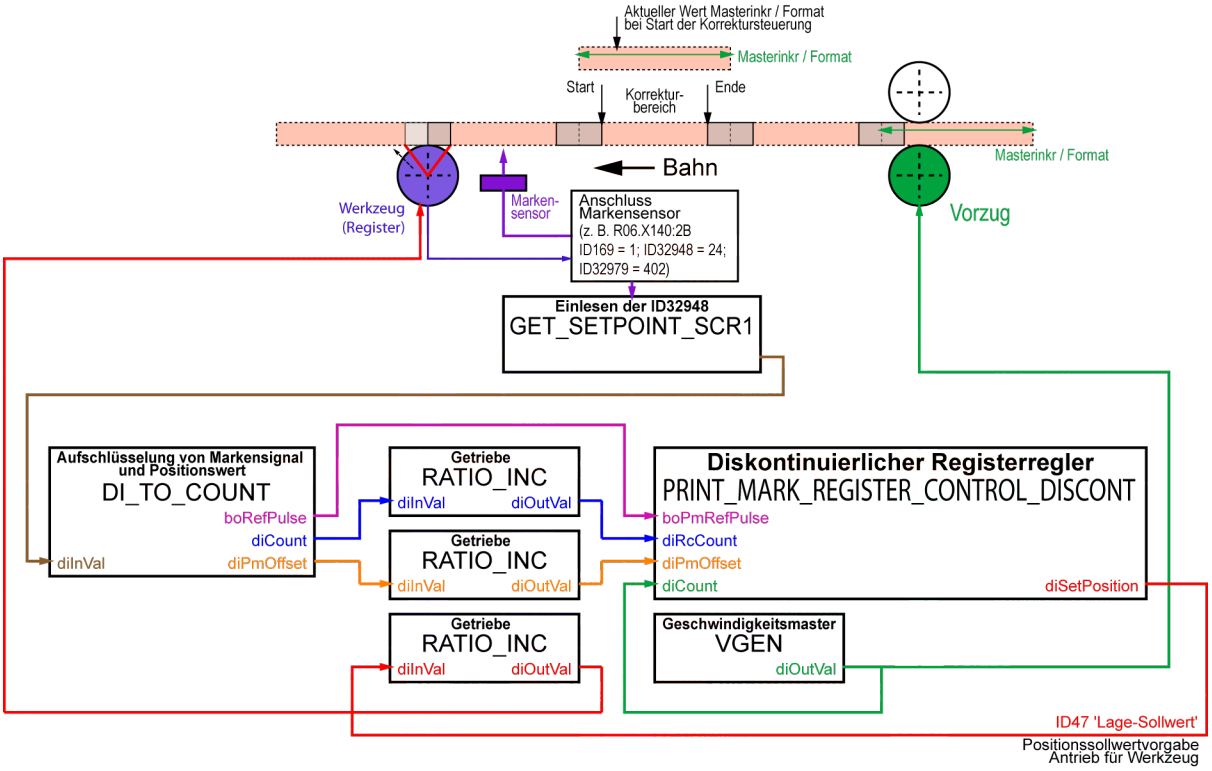
- Es wird eine Trendabweichung der Druckmarken-Ist-Position von der Soll-Position ausgeregelt.
- Der Baustein realisiert zwei Arten der Druckmarken-Sollpositions-Erfassung.
  - Die Druckmarken-Sollposition kann auf eine absolute Position innerhalb des Druckformates festgelegt werden
  - Die erste erkannte Druckmarke legt die Sollposition fest (Siehe 'PRINT_MARK_CONTROL (FB)' auf Seite 505.).
- Steuerung Druckmarkensensor in Abhängigkeit eines Fensters.
- Überwachung der Druckmarkenerkennung.
- Simulation fehlender Druckmarken.
- Ausgabe des Korrekturweges als gleichmäßige Verteilung der Inkremente über einen vorgebbaren Korrekturweg.
- Begrenzung der Korrekturgeschwindigkeit
- Ausgabe eines Offsets zur Verschiebung der Druckmarken-Sollposition.

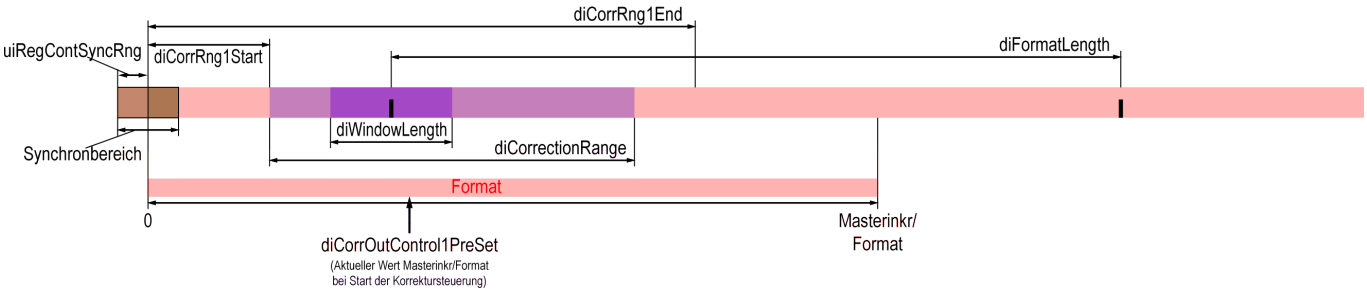
Der Baustein arbeitet intern mit positiver Zählrichtung der Eingangsimpulse und liefert positive Ausgangsimpulse. Mit den Variablen 'boMasterNegDir' und 'boMotorNegDir' kann die Zähl- bzw. Drehrichtung angepasst werden. Alle anderen Ausgangswerte bleiben davon unbeeinflusst.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Prinzipskizze





Beispielwerte

diFormatLength	10000 Inkr
diWindowLength	2500 Inkr
diCorrectionRange	70 %
diCorrRng1Start	1000 Inkr
diCorrRng1End	9000 Inkr
diCorrOutControl1PreSet	0 Inkr (keine Verschiebung)
uiRegContSyncRng	18 °

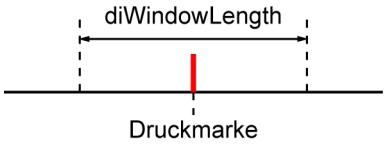
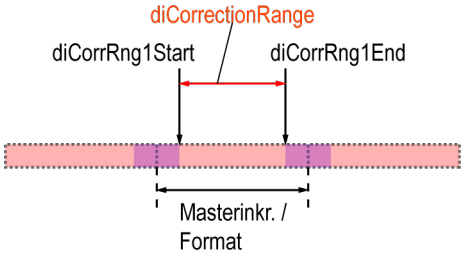
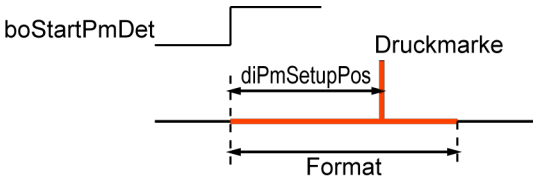
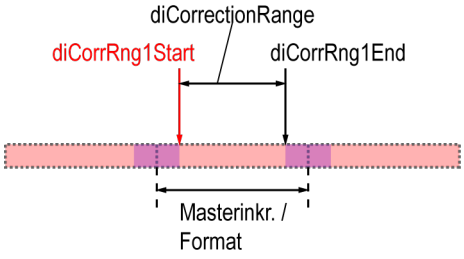
Anwender Interface

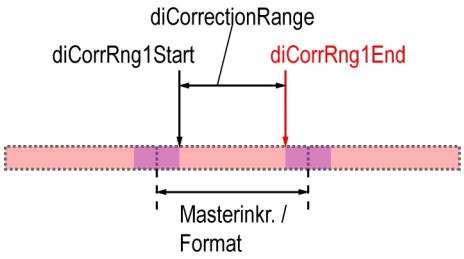
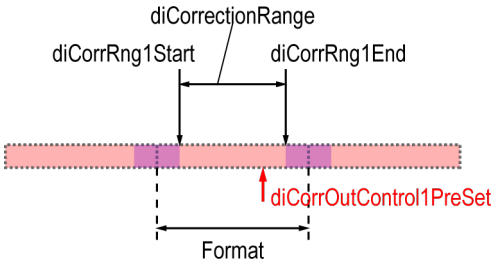
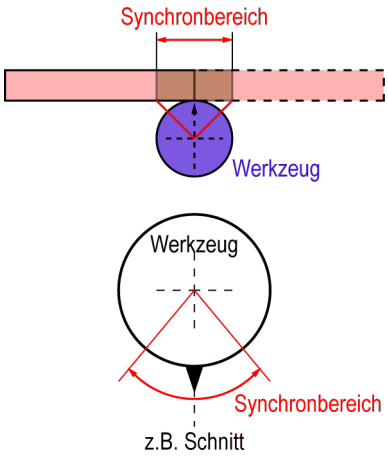
		PRINT_MARK_REGISTER_CONTROL_DISCONT			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Rücksetzen Ausgang boPmLost	-	boResetPmLost	boParameterSetupDone	-	Parametrierung beendet
Start Ausgabe Operator-Offset	-	boSetupOffset	boPmDetected	-	Marke gefunden
Start mit Druckmarken-Sollposition	-	boSetupPmPos	boPmLost	-	Marke verloren
Master zählt negativ	-	boMasterNegDir	boPmWnd	-	Fenster ein
Motorrichtung negativ	-	boMotorNegDir	diPmDeviation	-	Abweichung Soll-Ist-Position Druckmarke
Freigabe Korrekturausgabe	-	boCorrOutControlOn	diOpOffsetOut	-	Ausgegebener Operator-Offset
Operator Offset	-	diOpOffset	iActPmLost	-	Anzahl verlorener Druckmarken
Formatlänge	-	diFormatLength	boErr	-	Fehlersignal
Fensterlänge	-	diWindowLength	iErrID	-	Fehler Identnummer
Anzahl Marken bis Meldung "Marke verloren"	-	iNoPmLost			
Korrekturbereich	-	diCorrectionRange			
Druckmarken-Sollposition	-	diPmSetupPos			
Maximaler Korrekturwert pro Format	-	diCorrectionLimit			
Start der Korrekturbereich 1	-	diCorrRng1Start			
Ende der Korrekturbereich 1	-	diCorrRng1End			
Offset der Korrektursteuerung 1	-	diCorrOutControl1PreSet			
Start der Korrekturbereich 2	-	diCorrRng2Start			
Ende der Korrekturbereich 2	-	diCorrRng2End			
Offset der Korrektursteuerung 2	-	diCorrOutControl2PreSet			
Registerregler Format	-	diRegContFormat			
Verhältnis Masterinkremente	-	diRegControllerRatio			
Registerregler Synchronbereich	-	uiRegContSyncRng			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

		actSync			
Start Markensuche	-	boStartPmDet			
Druckmarke erkannt	-	boPmRefPulse			
Impulse des Mastergebers	-	diCount			
Druckmarken-Offset	-	diPmOffset			
Registerregler-Impulse	-	diRcCount			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boResetPmLost	BOOL	Rücksetzen des Ausgangs 'boPmLost'
boSetupOffset	BOOL	Ausgabestart des angegebenen Werts von 'diOpOffset'. Dieser Wert dient als Offset zur Verschiebung der Druckmarken-Sollposition
boSetupPmPos	BOOL	Steuerung der Sollpositionsbestimmung der Druckmarke bei positiver Flanke von 'boStartPmDet' 'boPmSetupPos' = FALSE: Es wird die Position der nächsten Druckmarke als Soll-Position erfasst und alle nachfolgenden Marken auf diese Position geregelt. 'boPmSetupPos' = TRUE: Der Parameter 'diPmSetupPos' gibt die Soll-Position der Druckmarke relativ zur aktuellen Masterposition an. Die folgenden Druckmarken werden auf diese Position geregelt.
boMasterNegDir	BOOL	Negation Master-Impuls-Richtung 'boMasterNegDir' = FALSE: keine Negation 'boMasterNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.
boMotorNegDir	BOOL	Negation Motor-Drehrichtung 'boMotorNegDir' = FALSE: keine Negation 'boMotorNegDir' = TRUE: Negation Dieser Eingang darf nur bei Programmstart gesetzt werden.
boCorrOutControlOn	BOOL	Korrekturausgabe-Steuerung aktiv. 'boCorrOutControlOn' = FALSE: Korrekturausgabe immer aktiv 'boCorrOutControlOn' = TRUE: Korrekturausgabe erfolgt nur in dem durch die Variablen 'diCorrRng1Start' und 'diCorrRng1End' bzw. 'diCorrRng2Start' und 'diCorrRng2End' festgelegten Bereich. Beide Bereiche sind mit ODER-Funktion verknüpft. Die Bereichszählung (Modulozählung) wird mit der 0→1 Flanke an 'boCorrOutControlOn' gestartet und lässt sich durch einen Wert an 'diCorrOutControl1PreSet' bzw. 'diCorrOutControl2PreSet' entsprechend einstellen. Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn', Format/2 an 'diCorrOutControl1PreSet' einzustellen.
diOpOffset	DINT	Operator-Offset. Verschiebt die Sollposition der Druckmarke. Die Verschiebung erfolgt im Korrekturbereich und wird mit einer 0→1 Flanke an 'boSetupOffset' gestartet. Sie ist unabhängig von Wertänderungen an 'diCount'. Die Verschiebegeschwindigkeit berechnet sich aus

Name	Typ	Beschreibung
diFormatLength	DINT	Formatlänge Soll-Abstand zweier Druckmarken
diWindowLength	DINT	Fensterlänge Länge vor und nach der Soll-Druckmarkenposition. In diesem Bereich ist der Druckmarkensensor aktiviert. Eine Druckmarke wird akzeptiert. 
iNoPmLost	INT	Anzahl Druckmarken, die nacheinander fehlen dürfen, bevor der Ausgang 'boPMLost' (Marke verloren) gesetzt wird.
diCorrectionRange	DINT	Korrekturbereich Die Druckmarkenregelung versucht die Korrektur im angegebenen Bereich auszugeben. [% diFormatLength] 
diPmSetupPos	DINT	Der Parameter gibt die Sollposition der Druckmarke zur aktuellen Masterposition bei positiver Flanke von 'boStartPmDet' an. Die erkannten Druckmarken werden auf diese Position geregelt. [Inkremente] 
diCorrectionLimit	DINT	Maximale Korrekturausgabe pro Format [Inkremente] 'diCorrectionLimit' = 0: keine Begrenzung 'diCorrectionLimit' > 0: maximaler Wert
diCorrRng1Start diCorrRng2Start	DINT	Korrekturbereich 1 / 2 Startwert bezogen auf Masterinkremente / Format [Inkremente] 

Name	Typ	Beschreibung
diCorrRng1End diCorrRng2End	DINT	Korrekturbereich 1 / 2 Endwert bezogen auf Masterinkremente / Format [Inkremente] 
diCorrOutControl1PreSet diCorrOutControl2PreSet	DINT	Positionswert (Nullverschiebung) bei Start der Korrektursteuerung bezogen auf Masterinkremente / Format [Inkremente] Beispiel: Steht der zu regelnde Antrieb beim Einschalten der Korrektursteuerung in der Mitte des Formates, so ist bei der 0->1 Flanke an 'boCorrOutControlOn' >Format/2< einzustellen. 
diRegContFormat	DINT	Anzahl Inkremente pro Register Umdrehung [Inkremente]
diRegControllerRatio	DINT	Synchronverhältnis [%]:
uiRegContSyncRng	UINT	Bereich, in dem der Registerregler synchron zum bearbeitenden Band laufen soll [Grad] 

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
boStartPmDet	BOOL	Start der Druckmarkenregelung und Bestimmung der Druckmarken-Sollposition durch Parameter 'diPmSetupPos' wenn 'boSetupPmPos' = TRUE
boPmRefPulse	BOOL	Markenimpuls (Referenzpuls) erkannt (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport TIME_TO_COUNT
diCount	DINT	Masterimpulse [Inkremente] (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport DI_TO_COUNT

Name	Typ	Beschreibung
diPmOffset	DINT	Druckmarkenoffset der aktuellen Position (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986) BasicSupport DI_TO_COUNT
diRcCount	DINT	Registerregler-Impulse Impulse des Werkzeugs [Inkr]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boParameterSetupDone	BOOL	Parametrierung beendet
boPmDetected	BOOL	Marke erkannt, Regelung aktiv
boPmLost	BOOL	Meldung "Marke verloren", wenn die an Parameter 'iNoPmLost' eingestellte Anzahl von aufeinanderfolgenden Marken nicht erkannt wurde.
boPmWnd	BOOL	Markenfenster aktiviert den Druckmarkensensor
diPmDeviation	DINT	Abweichung Soll- und Istposition der aktuellen Druckmarke
diOpOffsetOut	DINT	Ausgabewert beinhaltet den aktuell ausgegebenen Operatoroffset
iActPmLost	INT	Anzahl verlorener Druckmarken
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben 101 diFormatLength ≤ 0 104 diCorrectionRange ≤ 0 105 diRegContFormat ≤ 0 106 AND 107 uiRegContSyncRng < 0 OR uiRegContSyncRng > 180 > 1000 Fehlercode siehe AmkBase.lib Baustein 'TAB_CALC' iErrID – 1000

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
PRINT_MARK_REGISTER_CONTROL_DISCONT	PRINT_MARK_REGISTER_CONTROL_DISCONT.actSync

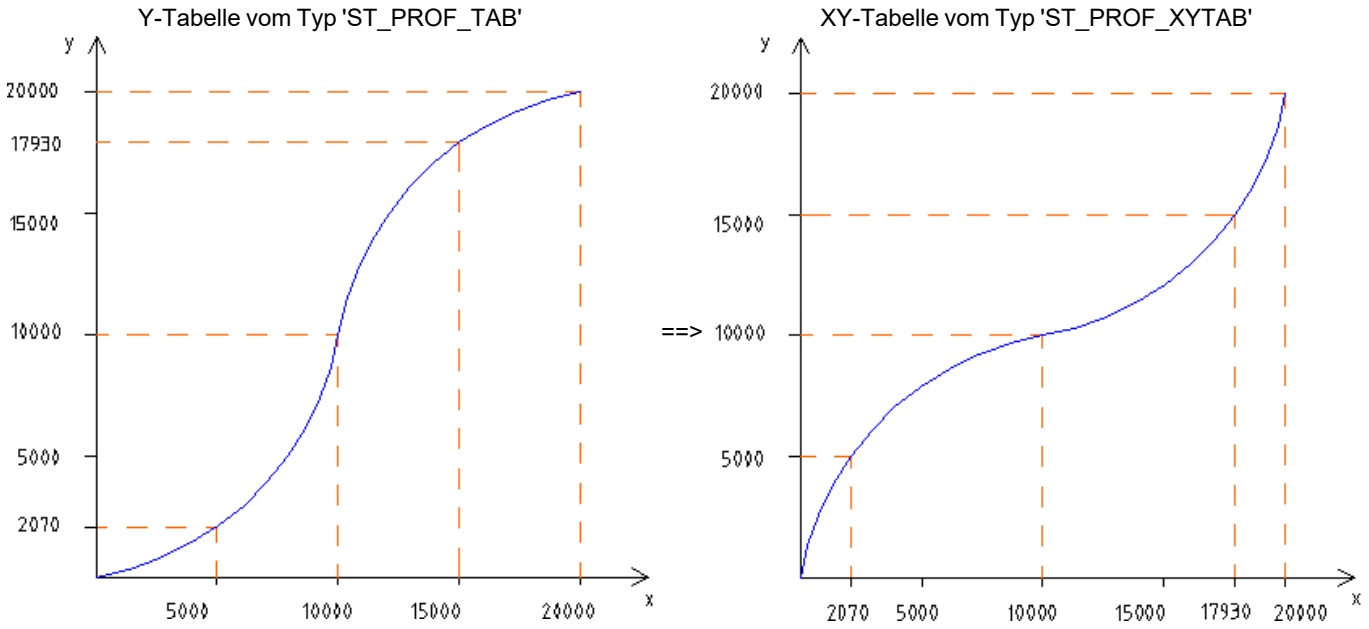
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.2.11 PRINT_MARK_TABLE_INVERTER (FB)

Der Funktionsbaustein 'PRINT_MARK_TABLE_INVERTER' wandelt und invertiert eine Y-Tabelle vom Typ 'ST_PROF_TAB' in eine XY-Tabelle vom Typ 'ST_PROF_XYTAB'

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Wirkungsweise des Funktionsbausteins PRINT_MARK_TABLE_INVERTER



Anwender Interface

		PRINT_MARK_TABLE_INVERTER			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Funktionsausführung	-	boExec	boDone	-	Quittierung "Ausgeführt"
Y-Tabelle	-	stYTab	stYTab	-	Y-Tabelle
XY-Tabelle	-	stXYInversTab	stXYInversTab	-	XY-Tabelle

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet. Invertierung ausführen

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde Ausgabe abgeschlossen

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stYTab	STRUCT	ST_PROF_TAB Tabellenstruktur (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)
stXYInversTab	STRUCT	ST_PROF_XYTAB Tabellenstruktur XY-Tabelle (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
PRINT_MARK_TABLE_INVERTER	

- Wird die Berechnung der Tabelle mit dem Baustein 'TAB_CALC' (AmkBase.lib) oder einem Baustein aus der AmkTabC.lib durchgeführt, kann die Tabellenstruktur unverändert am Eingang 'stYTab' verwendet werden.
- Wird die Tabelle mit dem Editor erstellt, ist folgende Anpassung nötig:

  Deklaration:

```
VAR
    pYTab:   POINTER TO ST_PROF_TAB;
END_VAR
```

  Programmaufruf:

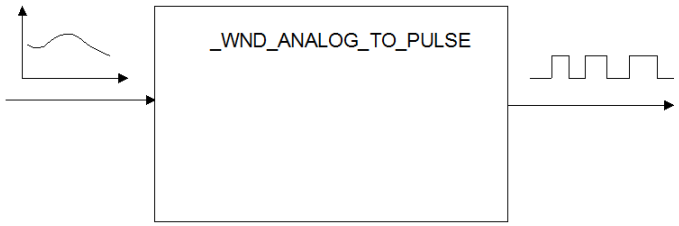
```
pYTab:=ADR(g_TestCamY);

fbPrintMarkTableInvert(
        boEnable:= ,
        boExec:= ,
        stYTab:=          pYTab^,
        stXYInversTab:= g_TestXY,
        boEnabAck=> ,
        boDone=> );
```

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.3 03_Winder

4.2.1.2.3.1 WINDER_ANALOG_TO_PULSE (FB)

Der Funktionsbaustein 'WINDER_ANALOG_TO_PULSE' ermöglicht die Umwandlung eines Analogwerts in Impulse.
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.



Anwender Interface

		WINDER_ANALOG_TO_PULSE			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Ausgabewert löschen	-	boResetOutput	diPulsOutVal	-	Impulsausgang
Analogeingang	-	iAnalogInVal	boErr	-	Fehlersignal
Anzahl Impulse bei 10 V	-	diPulsesAt10V	iErrID	-	Fehler Identnummer
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boResetOutput	BOOL	Löscht die Ausgabewerte
iAnalogInVal	INT	Analogeingang , aktueller Analogwert [mV]
diPulsesAt10V	DINT	Anzahl der Impulse pro Sekunde bei 10 V [Inkremente/s] Umrechnung in U/min des Antriebs:

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diPulsOutVal	DINT	Impulsausgang, die Impulse können zur Realisierung einer analogen Geschwindigkeitsvorgabe zu Baustein 'WINDER_WINDER' Eingang 'diActWebPos' kopiert werden.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

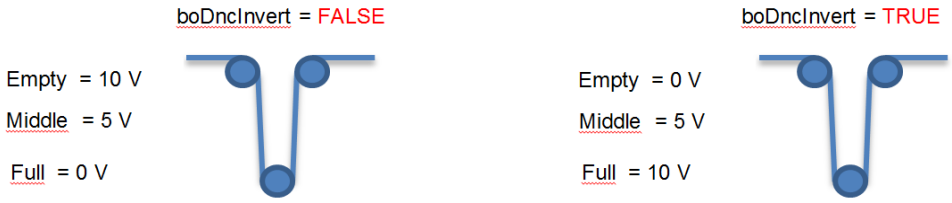
Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	WINDER_ANALOG_TO_PULSE

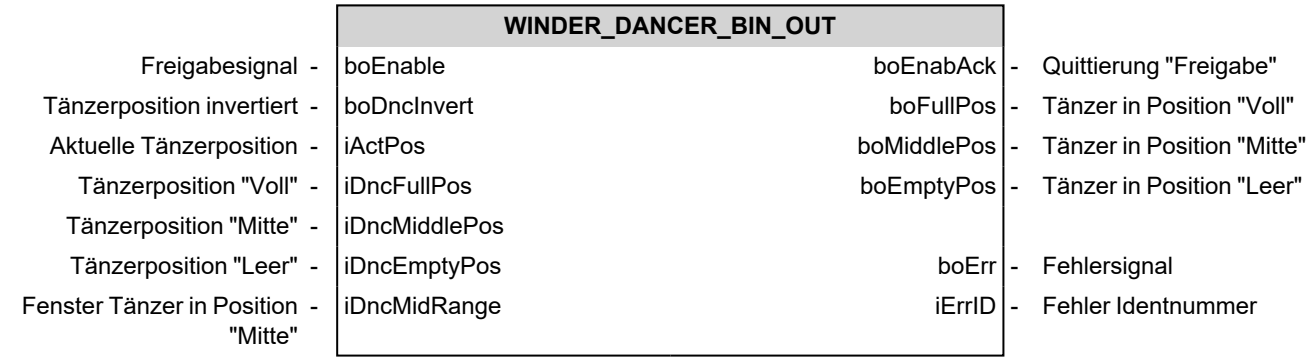
4.2.1.2.3.2 WINDER_DANCER_BIN_OUT (FB)

Der Funktionsbaustein 'WINDER_DANCER_BIN_OUT' ermöglicht eine Bewertung der Tänzerposition. Die Bewertung wird über Binärausgänge ausgegeben und kann im weiteren Programm zur Überwachung oder Statusanzeige weiter verarbeitet werden.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.



Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boDncInvert	BOOL	FALSE: keine Tänzerinvertierung, Position "Voll" des Tänzers bei kleinster Spannung TRUE: Tänzerinvertierung aktiv, Position "Voll" des Tänzers bei größter Spannung
iActPos	INT	Aktuelle Tänzerposition [mV]
iDncFullPos	INT	Tänzer in Position "Voll" [mV] Standardwert: 1500 mV
iDncMiddlePos	INT	Tänzer in Position "Mitte" [mV] Standardwert: 5000 mV

Name	Typ	Beschreibung
iDncEmptyPos	INT	Tänzer in Position "Leer" [mV] Standardwert: 9500 mV
iDncMidRange	INT	Bereich um 'iDncMiddlePos' für Tänzer in mittlerer Position [mV] Standardwert: 1000 mV

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boFullPos	BOOL	Tänzer in Position "Voll"
boMiddlePos	BOOL	Tänzer in Position "Mitte"
boEmptyPos	BOOL	Tänzer in Position "Leer"
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

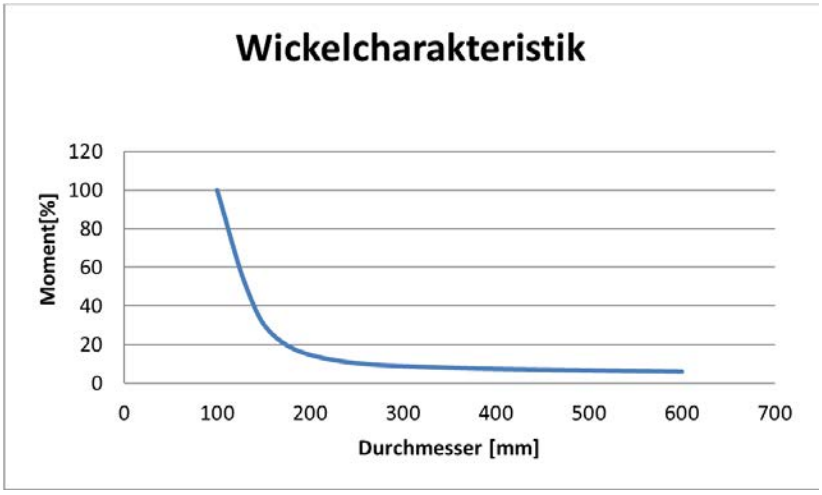
Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
WINDER_DANCER_BIN_OUT	

4.2.1.2.3.3 WINDER_CHARACTERISTICS (FB)

Der Funktionsbaustein 'WINDER_CHARACTERISTICS' realisiert die Anpassung der Bahnspannung in Abhängigkeit des Rollendurchmessers anhand einer Wickelcharakteristik. Der Verlauf der Charakteristik wird in einer Wertetabelle vorgegeben. Es können maximal 6 Wertepaare vorgegeben werden.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.



Anwender Interface

		WINDER_CHARACTERISTICS			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Aktueller Durchmesser	-	iActDm	iForce	-	Wickelkraft
Wertetabelle	-	arstFunction	boErr	-	Fehlersignal
			iErrID	-	Fehler Identnummer

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
iActDm	INT	Aktueller Durchmesser [mm]
arstFunction	ARRAY	ARRAY OF ST_WINDER_CHARACTERISTICS_XY_POINT Wertetabelle der Charakteristik als Punkte-Array iDiameter - Durchmessergrößen iTorque - Wickelkraft in %

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
iForce	INT	Wickelkraft [%]
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>iErrID</td><td>INT</td><td>Fehler Identnummer: Diagnosenummer wird ausgegeben
<table>
<tr><td colspan="2">iErrID = 0</td><td>Kein Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = TRUE</td><td>Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = FALSE</td><td>Warnung</td></tr>
</table></td></tr>
</table>

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	WINDER_CHARACTERISTICS

Programmbeispiel für Wickelcharakteristik

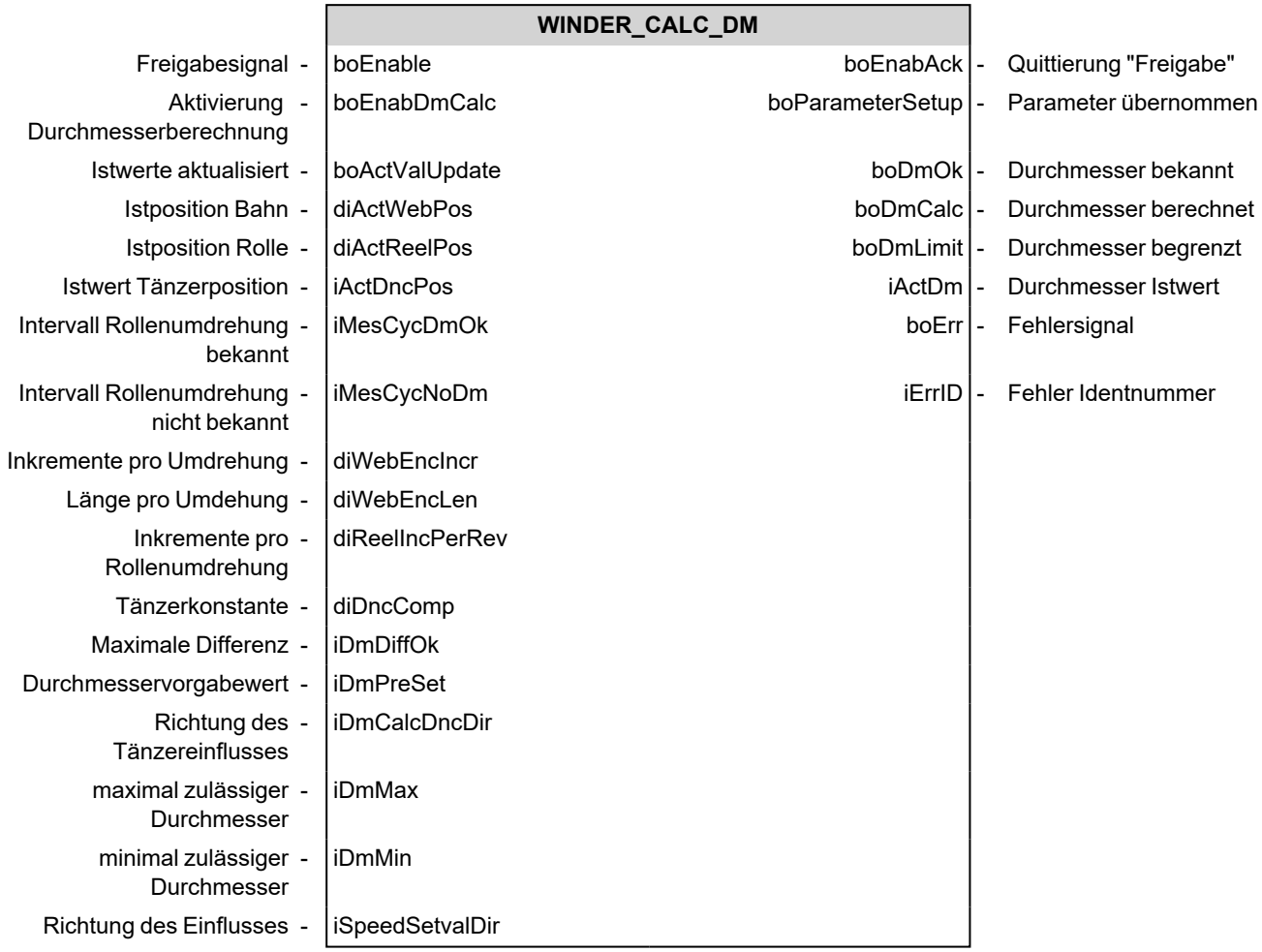
```
iDmDiff:=iDmMax - iDmMin;
fbWinderChar.arstFunction[0].iDiameter:= iDmMin;
fbWinderChar.arstFunction[0].iTorque:= 100;

fbWinderChar.arstFunction[1].iDiameter:= iDmMin+iDmDiff/16;
fbWinderChar.arstFunction[1].iTorque:= 50;

fbWinderChar.arstFunction[2].iDiameter:= iDmMin+iDmDiff/8;
fbWinderChar.arstFunction[2].iTorque:= 24;

fbWinderChar.arstFunction[3].iDiameter:= iDmMin+iDmDiff/4;
fbWinderChar.arstFunction[3].iTorque:= 12;

fbWinderChar.arstFunction[4].iDiameter:= iDmMin+iDmDiff/2;
fbWinderChar.arstFunction[4].iTorque:= 8;

fbWinderChar.arstFunction[5].iDiameter:= iDmMin+iDmDiff;
fbWinderChar.arstFunction[5].iTorque:= 6;
```

4.2.1.2.3.4 WINDER_CALC_DM (FB)

Der Funktionsbaustein 'WINDER_CALC_DM' realisiert die Durchmesserberechnung. Die Tänzerbewegung wird bei der Durchmesserberechnung kompensiert.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boEnabDmCalc	BOOL	Aktivierung der Durchmesserberechnung
boActValUpdate	BOOL	Istwerte wurden durch schnelle Ebene aktualisiert
diActWebPos	DINT	Istposition Bahn [Inkremente]
diActReelPos	DINT	Istposition Rolle [Inkremente]
iActDncPos	INT	Istwert Tänzerposition [mV]
iMesCycDmOk	INT	Rollenumdrehung für Zyklus Durchmesserberechnung wenn Durchmesser bekannt [0,1 Umdrehungen]
iMesCycNoDm	INT	Rollenumdrehung für Zyklus Durchmesserberechnung wenn Durchmesser nicht bekannt [0,1 Umdrehungen]

Name	Typ	Beschreibung
diWebEncIncr	DINT	Inkremente pro Umdrehung Bahn - Positionsgeber [Inkremente]
diWebEncLen	DINT	Länge pro Umdrehung Bahn - Positionsgeber [Inkremente]
diReelIncPerRev	DINT	Inkremente pro Rollenumdrehung [Inkremente]
diDncComp	DINT	Tänzerkonstante für Durchmesserberechnung
iDmDiffOk	INT	Maximale Differenz zwischen zwei aufeinanderfolgend berechneten Durchmessern, damit die Bewertung "Durchmesser gültig" übernommen wird [mm]
iDmPreSet	INT	Durchmesservorgabewert
iDmCalcDncDir	INT	Richtung des Tänzereinflusses auf die Durchmesserberechnung [1,-1]
iDmMax	INT	Maximal zulässiger Durchmesserwert [mm]
iDmMin	INT	Minimal zulässiger Durchmesserwert [mm]
iSpeedSetvalDir	INT	Richtung des Einflusses der Bahngeschwindigkeit auf die Durchmesserberechnung [1,-1]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boParameterSetup	BOOL	Parameter übernommen
boDmOk	BOOL	Durchmesser bekannt
boDmCalc	BOOL	Durchmesser wurde berechnet und Wert lag außerhalb von 'iDmMin' und 'iDmMax', der Wert wurde entsprechend begrenzt
boDmLimit	BOOL	Neuer Durchmesser - Istwert berechnet
iActDm	INT	Aktueller Durchmesser [mm]
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Aktionen

Name	Beschreibung
actParameterSetup	Parametriert den Baustein < WINDER_CALC_DM >. actParameterSetup(iMesCycDmOk:=..., iMesCycNoDm:=..., diWebEncIncr:=..., diWebEncLen:=..., diReelIncPerRev:=..., iSpeedSetvalDir:=..., iMesCycDmOk:=..., iMesCycNoDm:=..., iDmDiffOk:=..., iDmMax:=..., iDmMin:=...); Wird bei Enable des Bausteins aufgerufen.
actPresetDm	Setzt den Durchmesser auf den Wert 'iDmPreSet' <_WND_CALC_DM >. actPresetDm (iDmPreSet:=...));
actResetDmOk	Setzt "Durchmesser bekannt" auf FALSE <_WND_CALC_DM >. actResetDmOk();
actSetDmOk	Setzt den Durchmesser auf "bekannt" <_WND_CALC_DM >. actPresetDmOk();

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	WINDER_CALC_DM

4.2.1.2.3.5 WINDER_CALC_WEB_SPEED (FB)

Der Funktionsbaustein 'WINDER_CALC_WEB_SPEED' berechnet den Bahngeschwindigkeitsistwert und den Bahnbeschleunigungsistwert.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		WINDER_CALC_WEB_SPEED			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Bahnimpulse	-	diInVal	boWebRuns	-	Bahngeschwindigkeit <>0
ID2	-	diID2	iWebSpeed	-	Bahngeschwindigkeit
Messperiode	-	iPeriod	iWebAcc	-	Beschleunigung
Bahngeber Zähleingang	-	diWebEncIncr	boErr	-	Fehlersignal
Bahngeberweg	-	diWebEncLen	iErrID	-	Fehler Identnummer

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Bahngeber Zähleingang, Impulseingang [Inkremente]

Name	Typ	Beschreibung
diID2	INT	PGT Takt: ID2 'SERCOS-Zykluszeit' [µs]
iPeriod	DINT	Messperiode in Vielfache von ID2
diWebEncIncr	DINT	Anzahl der gezählten Inkremente am Eingang 'diInVal' bei zurücklegen der Strecke 'diWebEncLen' [Inkremente]
diWebEncLen	DINT	Zurückgelegte Bahnstrecke bei Änderung des Eingangs 'diInVal' um 'diWebEncIncr' Inkremente [mm]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boWebRuns	BOOL	Bahngeschwindigkeit <>0
iWebSpeed	INT	Bahngeschwindigkeit [mm/s]
iWebAcc	INT	Bahnbeschleunigung [mm/s^2]
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Verwendungshinweis im CoDeSys-Programm

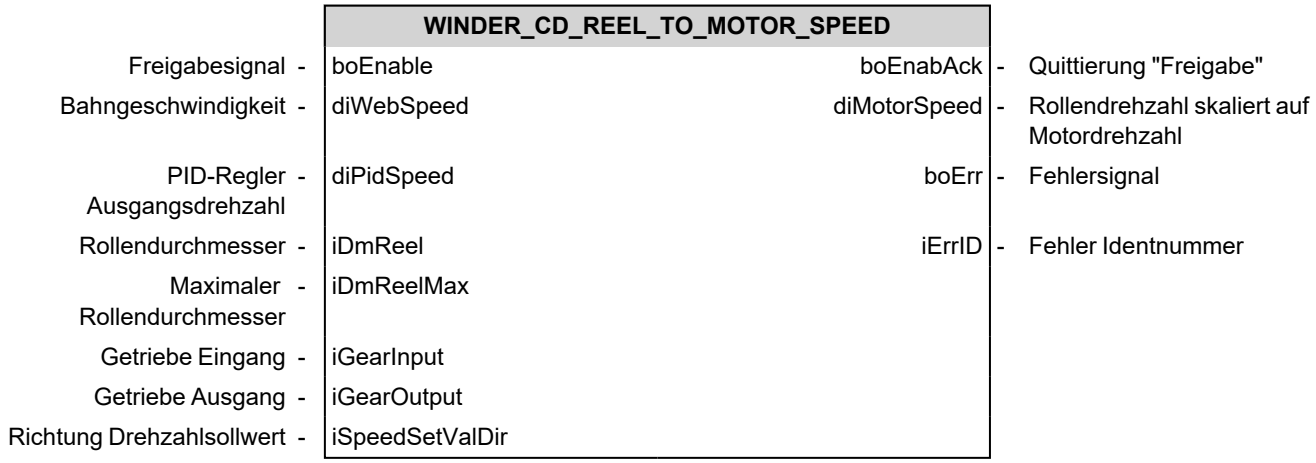
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	WINDER_CALC_WEB_SPEED

4.2.1.2.3.6 WINDER_CD_REEL_TO_MOTOR_SPEED (FB)

Der Funktionsbaustein 'WINDER_CD_REEL_TO_MOTOR_SPEED' berechnet die Motordrehzahl für die Rolle mit dem in 'iDmReel' angegebenen Durchmesser aus der Bahngeschwindigkeit und der Ausgangsdrehzahl des PID-Reglers, im Bezug auf den maximalen Rollendurchmesser 'iDmReelMax'.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diWebSpeed	DINT	Bahngeschwindigkeit [mm/s]
diPidSpeed	DINT	Ausgang PID-Regler [0,1 U/min]
iDmReel	INT	Rollendurchmesser [mm]
iDmReelMax	INT	maximaler Rollendurchmesser [mm]
iGearInput	INT	Getriebe Eingang Motor → Rolle
iGearOutput	INT	Getriebe Ausgang Motor → Rolle
iSpeedSetValDir	INT	Richtung Drehzahlsollwert [1,-1]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diMotorSpeed	DINT	Skalierte Rollendrehzahl, Drehzahlsollwert Motor [0,0001 U/min]
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Verwendungshinweis im CoDeSys-Programm

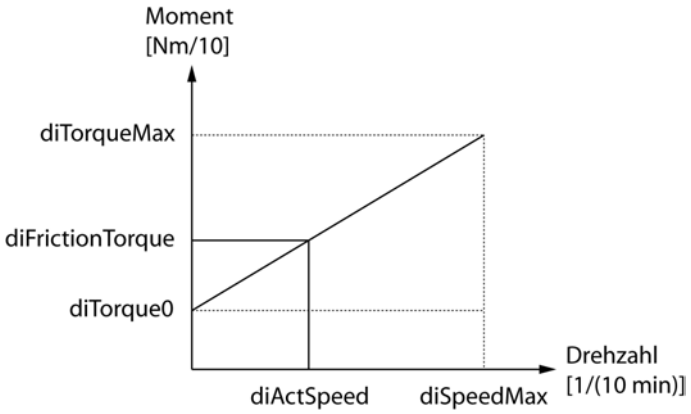
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	WINDER_CD_REEL_TO_MOTOR_SPEED

4.2.1.2.3.7 WINDER_CALC_FRICTION (FB)

Der Funktionsbaustein 'WINDER_CALC_FRICTION' berechnet das Reibmoment eines Wickelantriebs in Abhängigkeit der Aktuellen Drehzahl

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Berechnungsprinzip: Funktion des Reibmoments



Anwender Interface

	WINDER_CALC_FRICTION			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Grundreibmoment -	diTorque0	boErr	-	Fehlersignal
Drehzahl bei diTorqueMax -	diSpeedMax	iErrID	-	Fehler Identnummer
Reibmoment bei diSpeedMax -	diTorqueMax			

	actSync			
aktuelle Drehzahl Motor -	diActSpeed	diFrictionTorqueAbs	-	Reibmoment

Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diTorque0	DINT	Grundreibmoment [0,1 Nm]
diSpeedMax	DINT	Drehzahl bei der 'diTorqueMax' wirkt [0,1 Nm]
diTorqueMax	DINT	Reibmoment bei Drehzahl 'diSpeedMax' [0,1 Nm]

Eingangsvariable der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diActSpeed	DINT	aktuelle Drehzahl [0,1 U/min]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben

Name	Typ	Beschreibung
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Ausgangsvariable der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diFrictionTorqueAbs	DINT	wirksames Reibmoment absolut [0,1 Nm]

Aktionen

Name	Beschreibung
actParameterSetup	Parametriert den Baustein < WINDER_CALC_DM >. actParameterSetup(diTorque0:=..., diSpeedMax:=..., diTorqueMax:=...,); Wird bei Enable des Bausteins aufgerufen.

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
WINDER_CALC_FRICTION	WINDER_CALC_FRICTION.actSync

4.2.1.2.3.8 WINDER_CALC_INERTIA (FB)

Der Funktionsbaustein 'WINDER_CALC_INERTIA' berechnet das Trägheitsmoment einer Rolle. Für die Berechnung sind Wickler als homogene Vollzylinder (J_{Wickel}= 0,5 x m x r^2) angenommen. Um das Trägheitsmoment zu bestimmen, sind die Berechnungsparameter Durchmesser, Dichte und Bahnbreite nötig.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	WINDER_CALC_INERTIA			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Durchmesser -	iDm	diInertia	-	Trägheitsmoment
Bahnbreite -	iWidth	boErr	-	Fehlersignal
Materialdichte -	iDensity	iErrID	-	Fehler Identnummer

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
iDm	INT	Rollendurchmesser [mm]
iWidth	INT	Rollenbreite [mm]
iDensity	INT	Dichte des Rollenmaterials [kg/m^3]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
diInertia	DINT	Trägheitsmoment des Wicklers [kgm^{2}x1000]
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
WINDER_CALC_INERTIA	

4.2.1.2.3.9 WINDER_CALC_TORQUE (FB)

Der Funktionsbaustein 'WINDER_CALC_TORQUE' berechnet das Reibmoment der Rolle.
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	WINDER_CALC_TORQUE			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Durchmesser gültig -	boDmOk	iSetTorque	-	Momentsollwert
absolute Reibekomponente -	diFrictionAbs	boErr	-	Fehlersignal
absolute Bahnspannung -	diWebTension	iErrID	-	Fehler Identnummer
Trägheitsmoment der Rolle -	diReelInertia			
Rollendurchmesser -	iDmReel			
Bahnbeschleunigung -	diWebAcc			
Getriebe Eingang -	iGearInput			
Getriebe Ausgang -	iGearOutput			
Motor Nennmoment -	iRatedTorque			
Momentsollwert wenn Durchmesser bekannt -	diSetupReelTorqueAbs			
Richtung der Bahnspannung zur Motordrehrichtung -	iWebTensionDir			
Richtung der Reibung zur Motordrehrichtung -	iFrictionDir			
Richtung der Bahngeschwindigkeit zur Motordrehrichtung -	iSpeedDir			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boDmOk	BOOL	Durchmesser bekannt [mm] Rollendurchmesser 'iDmReel' gültig
diFrictionAbs	DINT	Absolutes Reibemoment [0,1% Nm]
diWebTension	DINT	Absolutes Bahnspannungsmoment [0,1 Nm]
diReelInertia	DINT	Trägheitsmoment der Rolle [0,001 kgm^2]
iDmReel	INT	Rollendurchmesser [mm]
diWebAcc	DINT	Bahnbeschleunigung [mm/s^2]
iGearInput	INT	Getriebe Eingangsdrehzahl
iGearOutput	INT	Getriebe Ausgang sdrehzahl
iRatedTorque	INT	Motor Nennmoment [0,1 Nm]
diSetupReelTorqueAbs	DINT	Momentsollwert der Rolle, wenn der Durchmesser 'iDmReel' unbekannt ist [0,1% Mn]
iWebTensionDir	INT	Richtung der Bahnspannung im Bezug auf die positive Motordrehrichtung [1,-1]
iFrictionDir	INT	Richtung des Reibemoments in Bezug auf die positive Motordrehrichtung [1,-1]
iSpeedDir	INT	Richtung der Bahngeschwindigkeit in Bezug auf die positive Motordrehrichtung [1,-1]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
iSetTorque	INT	Momentsollwert [0,1% Mn]
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Verwendungshinweis im CoDeSys-Programm

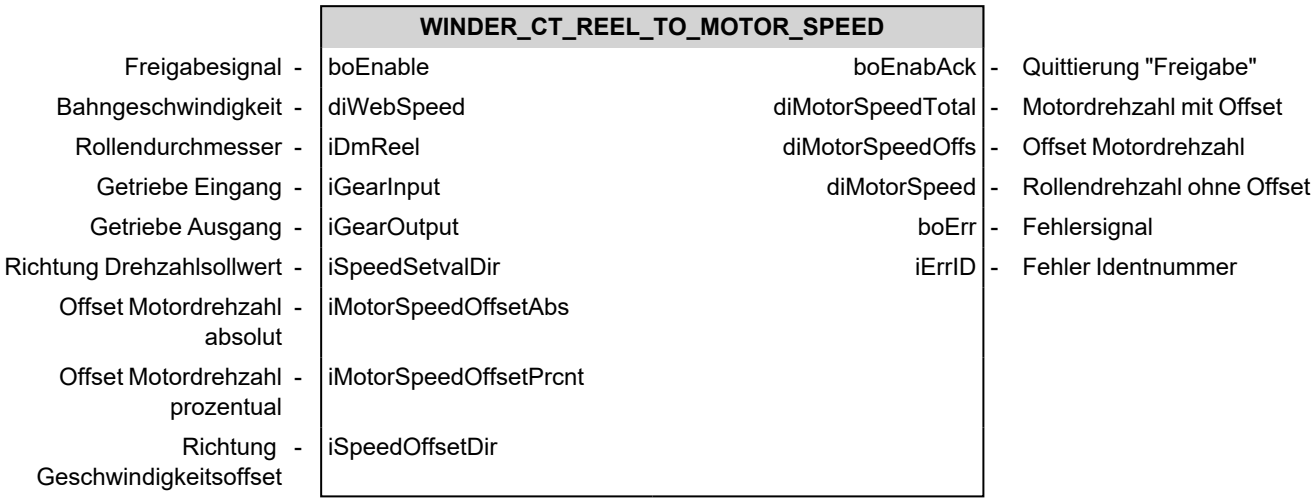
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	WINDER_CALC_TORQUE

4.2.1.2.3.10 WINDER_CT_REEL_TO_MOTOR_SPEED (FB)

Der Funktionsbaustein 'WINDER_CT_REEL_TO_MOTOR_SPEED' berechnet aus der Drehzahl der Rolle die Drehzahl des Motors. Es wird die Richtung der Drehzahl aufgrund der Bahngeschwindigkeit, sowie die Richtung des Drehzahloffsets aufgrund der Richtung des Bahnzuges berücksichtigt.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diWebSpeed	DINT	Bahngeschwindigkeit [mm/s]
iDmReel	INT	Rollendurchmesser [mm]
iGearInput	INT	Getriebe Eingang Motor → Rolle
iGearOutput	INT	Getriebe Ausgang Motor → Rolle
iSpeedSetvalDir	INT	Richtung Drehzahlsollwert [1,-1]
iMotorSpeedOffsetAbs	INT	Offset Drehzahl absolut [0,1 U/min]
iMotorSpeedOffsetPrcnt	INT	Offset Drehzahl prozentual [%]
iSpeedOffsetDir	INT	Offset Drehzahl Richtung [1,-1]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boParamSetupDone	BOOL	Parametrierung beendet
diMotorSpeedTotal	DINT	Motordrehzahl Sollwert [0,0001 U/mni]
diMotorSpeedOffs	DINT	Offset Motordrehzahl [0,1 U/min]
diMotorSpeed	DINT	Rollendrehzahl Sollwert [0,1 U/min]

Name	Typ	Beschreibung
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Verwendungshinweis im CoDeSys-Programm

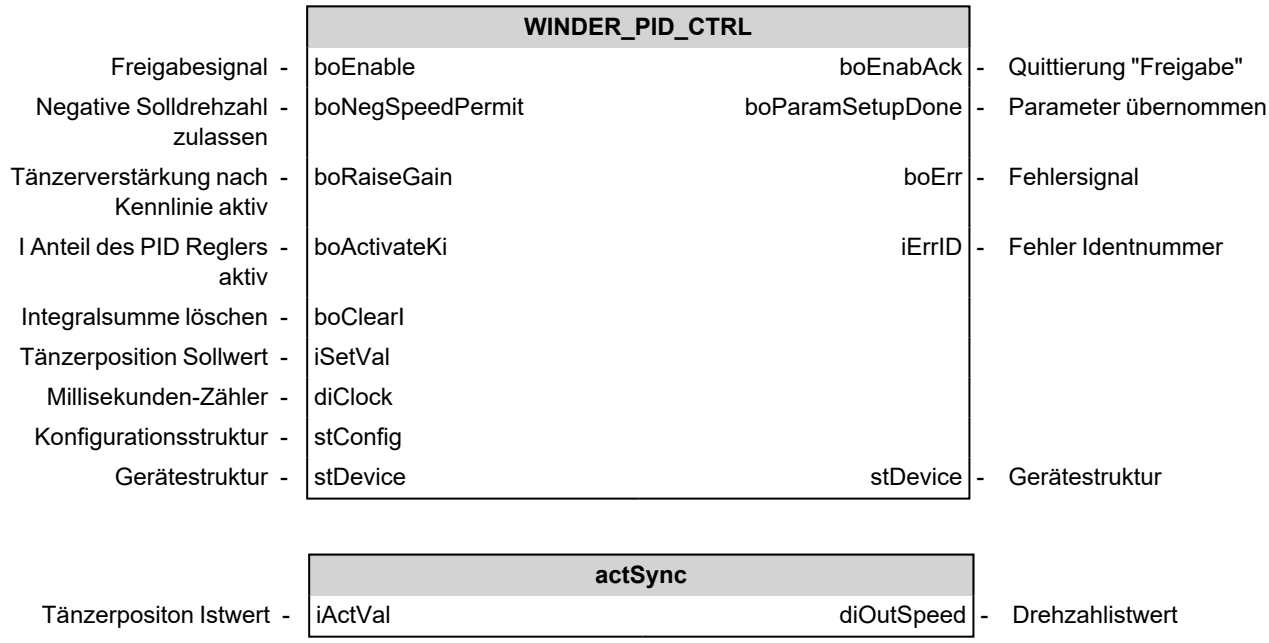
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	WINDER_CT_REEL_TO_MOTOR_SPEED

4.2.1.2.3.11 WINDER_PID_CTRL (FB)

Der Funktionsbaustein 'WINDER_PID_CTRL' realisiert die Tänzerregelung der Rolle.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Anwender Interface



Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boNegSpeedPermit	BOOL	Negativer Drehzahlsollwert zulässig
boRaiseGain	BOOL	FALSE: Tänzerverstärkung gleich Parameter iKp TRUE: Tänzerverstärkung nach Kennlinie aktiv
boActivateKi	BOOL	I Anteil im PID Regler aktiv
boClearI	BOOL	Integralsumme löschen des PID Reglers

Name	Typ	Beschreibung
iSetVal	INT	Tänzerposition Sollwert [mV]
stConfig	STRUCT	ST_WINDER_PID_CONFIG Konfigurationsstruktur

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
iActVal	INT	Tänzerposition Istwert [mV]
diClock	DINT	ms-Zähler [ms]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boParamSetupDone	BOOL	Parameter übernommen
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Ausgangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diOutSpeed	DINT	Drehzahlsollwert (Regelgröße) Die Einheit ist abhängig von 'ST_WINDER_PID_CONFIG' → 'diOutputScale' diOutputScale = 1 → [0,0001 U/min] diOutputScale = 10 → [0,001 U/min] diOutputScale = 100 → [0,01 U/min] diOutputScale = 1000 → [0,1 U/min]

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Aktionen

Name	Beschreibung
actParameterSetup	Parametriert den Baustein < WINDER_PID_CTRL >. actParameterSetup(stConfig:=...,); Wird bei Enable des Bausteins aufgerufen.

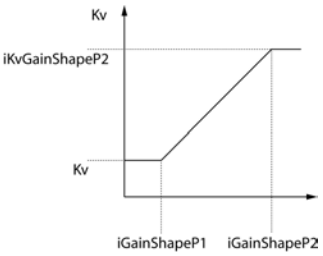
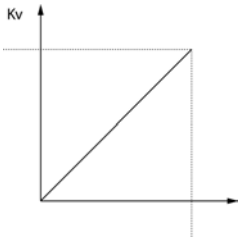
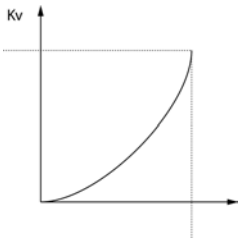
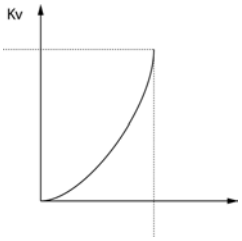
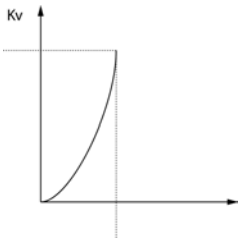
Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
WINDER_PID_CTRL	WINDER_PID_CTRL.actSync

ST_WINDER_PID_CONFIG (ST)

Konfigurationsstruktur PID-Regler

Name	Typ	Bsp. Wert	Beschreibung
iPidTick	INT	10	Aufruf Takt des PID-Regler [iPidTick*ID2]
iKp	INT	5000	Kp für PID-Regler [-]
iKi	INT	10	Ki für PID-Regler [-]
iKd	INT	0	Kd für PID-Regler [-]
iKv	INT	25	Gesamtverstärkung des PID-Reglers: Bei aktivierter tänzerlage abhängiger Verstärkung ist Kv die Verstärkung bei (Tänzersollwert - Tänzeristwert) < iGainShapeP1 [-]
diOutputScale	DINT	1	Maßstab Drehzahlausgang [-]
iPosKiActivate	INT	2500	Tänzerposition für die Aktivierung von Ki [mV]
diImax	DINT	20000	Maximal zulässiger Integralanteil [U/10000 min]
iDmPidSet	INT	600	Rollendurchmesser, bei dem die Parameter 'iKp', 'iKi', 'iKd' und 'iKv' des PID-Reglers eingestellt wurden [mm]

Name	Typ	Bsp. Wert	Beschreibung
iGainShapeType	INT		Typ der Verstärkeranpassung Wert = 0:  Tänzerspannungsdifferenz [mV] = (Tänzeristposition - Tänzersollposition) Wert = 1: lineare Verstärkungskennlinie  Tänzerspannungsdifferenz [mV] = (Tänzeristposition - Tänzersollposition) Wert =2: quadratische Verstärkungskennlinie  Tänzerspannungsdifferenz [mV] = (Tänzeristposition - Tänzersollposition) Wert =3: Verstärkungskennlinie[3]  Tänzerspannungsdifferenz [mV] = (Tänzeristposition - Tänzersollposition) Wert = 4: Verstärkungskennlinie [4]  Tänzerspannungsdifferenz [mV] = (Tänzeristposition - Tänzersollposition)
iGainShapeP1	INT	1000	(Tänzeristwert - Tänzersollwert) = iGainShapeP1 Startpunkt der Verstärkungsrampe [mV]
iGainShapeP2	INT	3000	(Tänzeristwert - Tänzersollwert) = iGainShapeP1 Endpunkt der Verstärkungsrampe [mV]

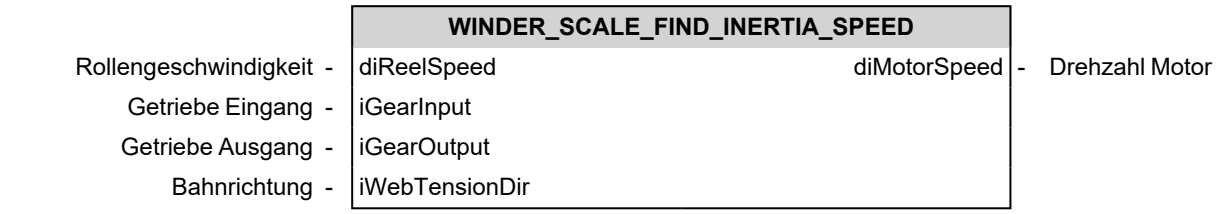
Name	Typ	Bsp. Wert	Beschreibung
iKvGainShapeP2	INT	50	Verstärkung bei (Tänzeristwert - Tänzersollwert) > iGainShapeP2

4.2.1.2.3.12 WINDER_SCALE_FIND_INERTIA_SPEED (FB)

Der Funktionsbaustein 'WINDER_SCALE_FIND_INERTIA_SPEED' berechnet die Geschwindigkeit, um das Trägheitsmoment der Rolle festzustellen.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
diReelSpeed	DINT	Rollengeschwindigkeit [0,1 U/min]
iGearInput	INT	Getriebe Eingang
iGearOutput	INT	Getriebe Ausgang
iWebTensionDir	INT	Richtung der Bahnspannung im Bezug auf die positive Motordrehrichtung [1,-1]

Ausgangsvariablen

Name	Typ	Beschreibung
diMotorSpeed	DINT	Motordrehzahl

Verwendungshinweis im CoDeSys-Programm

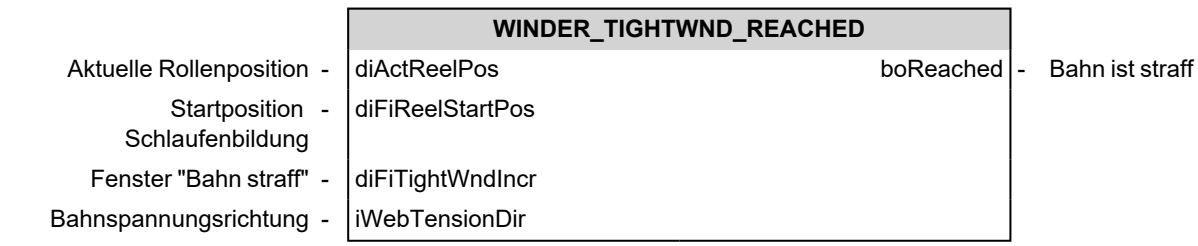
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
WINDER_SCALE_FIND_INERTIA_SPEED	

4.2.1.2.3.13 WINDER_TIGHTWND_REACHED (FB)

Der Funktionsbaustein 'WINDER_TIGHTWND_REACHED' überwacht die Straffheit der Bahn.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
diActReelPos	DINT	Istposition Rolle [Inkremente]
diFiReelStartPos	DINT	Lageistwert der Rolle an dem die Bahnschlaufe gebildet wurde [Inkremente]
diFiTightWndIncr	DINT	Fenster, in dem die Bahn auf Straffheit überwacht wird [Inkremente]
iWebTensionDir	INT	Richtung der Bahnspannung im Bezug auf die positive Motordrehrichtung [1,-1]

Ausgangsvariablen

Name	Typ	Beschreibung
boReached	BOOL	Bahn ist straff

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
WINDER_TIGHTWND_REACHED	

4.2.1.2.3.14 WINDER_DANCER_CONTROL (FB)

Der Funktionsbaustein 'WINDER_DANCER_CONTROL' realisiert einen Tänzerwickler mit folgenden Merkmalen:

- PID geregelter Tänzerwickler
- Betrieb als Auf- und Abwickler
- Durchmesserberechnung
- Durchmessersuche bei Start des Wicklers

‚Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Anwender Interface

		WINDER_DANCER_CONTROL			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Aktivierung der Durchmesserrechnung	-	boDmCalc	boDmOk	-	Durchmesser ok, Status "bekannt"
Tänzer in Mittenposition	-	boDncMidPos	boDmLimit	-	Durchmesserwert wurde begrenzt
Tänzermodus "Istwert folgen"	-	boDncTraceMode	iActDm	-	Istwert Durchmesser
Aktivieren Verstärkungsrampe	-	boRaiseGain	boErr	-	Fehlersignal
Aktivieren Ausgabe Bahngeschwindigkeit	-	boOutputSpeed	iErrID	-	Fehler Identnummer
Durchmesser setzen	-	boDmPreset			
Durchmesser setzen, Status "bekannt"	-	boDmOkPreset			
Durchmesser setzen, Status "unbekannt"	-	boDmNokPreset			
Durchmesser Status von "Bekannt" → "Unbekannt" setzen	-	boDmReset			
Ausgang 'boDmOk' setzen	-	boSetDmOk			
Tänzerregler I Anteil löschen	-	boClearI			
Durchmesservorgabewert	-	iDmPreset			
PGT-Takt	-	uiID2			
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

		actSync			
Istwert Rollenposition	-	diActReelPos	diVWebSpeed	-	Istwert Bahngeschwindigkeit
Istwert Bahnposition	-	diActWebPos	diSetValMotorSpeed	-	Drehzahlsollwert Motor
Istwert Tänzerposition	-	iActDncPos			

Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boDmCalc	BOOL	Aktivierung der Durchmesserberechnung
boDncMidPos	BOOL	Tänzer in die Mittenposition bewegen
boDncTraceMode	BOOL	Tänzermodus Istwert folgen, der Tänzeristwert wird als Sollwert am PID Regler übernommen. Nur Aktivierbar bei 'boDncMidPos' = FALSE
boRaiseGain	BOOL	Lageabhängige Tänzer-Verstärkungsrampe aktiv
boOutputSpeed	BOOL	Ausgabe Bahngeschwindigkeit als Drehzahlsollwert für Motor
boDmPreset	BOOL	Durchmesser auf Wert des Eingangs 'iDmPreset' setzen. Das Eingangsbit wird nach setzen automatisch rückgesetzt.
boDmOkPreset	BOOL	Durchmesser auf Wert der Eingangsvariable 'iDmPreset' setzen. Das Eingangsbit 'boDmPreset' wird nach setzen automatisch rückgesetzt. Durchmesserstatus "Bekannt": Der Durchmesserwert wird als gültiger Durchmesser gewertet, das bedeutet, Beschleunigungs- und Reibmomente werden berechnet.
boDmNokPreset	BOOL	Durchmesser auf Wert der Eingangsvariable 'iDmPreset' setzen. Das Eingangsbit 'boDmPreset' wird nach setzen automatisch rückgesetzt. Durchmesserstatus "Unbekannt": Der Durchmesserwert wird als ungültiger Durchmesser gewertet, das bedeutet, Beschleunigungs- und Reibmomente werden berechnet.
boDmReset	BOOL	Ausgang 'boDmOk' rücksetzen, → Durchmesserstatus "Unbekannt"
boSetDmOk	BOOL	Ausgang 'boDmOk' setzen, → Durchmesserstatus "Bekannt"
boClearI	BOOL	Tänzerregler: I-Anteil löschen
iDmPreset	INT	Durchmesservorgabewert Wert auf den der Durchmesser mit 'boDmNokPreset' = TRUE oder 'boDmOkPreset' = TRUE gesetzt wird. [mm]
uiID2	UINT	PGT-Takt [ms]

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diActReelPos	DINT	Istposition Rolle [Inkremente]
diActWebPos	DINT	Istposition Bahn [Inkremente]
iActDncPos	INT	Istwert Tänzerposition [mV]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boDmOk	BOOL	Durchmesser bekannt

Name	Typ	Beschreibung
boDmLimit	BOOL	Durchmesser wurde berechnet und das Ergebnis lag außerhalb von 'ST_WINDER_DANCER_CONTROL_CONFIG.iDmMin' und 'ST_WINDER_DANCER_CONTROL_CONFIG.iDmMax', der Wert wurde entsprechend begrenzt
iActDm	INT	Aktueller Durchmesser [mm]
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>iErrID</td><td>INT</td><td>Fehler Identnummer: Diagnosenummer wird ausgegeben
<table>
<tr><td colspan="2">iErrID = 0</td><td>Kein Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = TRUE</td><td>Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = FALSE</td><td>Warnung</td></tr>
</table>
<table>
<tr><th>Wert</th><th>Bedeutung</th></tr>
<tr><td>1</td><td>Der Wert am Eingang uiID2 = 0</td></tr>
<tr><td>2</td><td>Einer der folgenden Parameter hat einen Wert <=0:
ST_WINDER_DANCER_CONTROL_CONFIG.iGearInput,
ST_WINDER_DANCER_CONTROL_
CONFIG.iGearOutput</td></tr>
<tr><td>3</td><td>Einer der folgenden Parameter hat einen Wert <=0:
ST_WINDER_DANCER_CONTROL_CONFIG.iDmMin,
ST_WINDER_DANCER_CONTROL_CONFIG.iDmMax
oder
ST_WINDER_DANCER_CONTROL_CONFIG.iDmMin >=
ST_WINDER_DANCER_CONTROL_CONFIG.iDmMax</td></tr>
<tr><td>4</td><td>Der Parameter hat einen Wert <=0:
ST_WINDER_DANCER_CONTROL_
CONFIG.diMotIncPerRev</td></tr>
<tr><td>5</td><td>Einer der folgenden Parameter hat einen Wert <=0:
ST_WINDER_DANCER_CONTROL_
CONFIG.iMesCycDmOk,
ST_WINDER_DANCER_CONTROL_
CONFIG.iMesCycNoDm,
ST_WINDER_DANCER_CONTROL_CONFIG.iDmDiffOk</td></tr>
<tr><td>9</td><td>Einer der folgenden Parameter hat einen Wert <=0:
ST_WINDER_DANCER_CONTROL_
CONFIG.diWebSpeedMax,
ST_WINDER_DANCER_CONTROL_
CONFIG.diWebEncLen,
ST_WINDER_DANCER_CONTROL_
CONFIG.diWebEncIncr,
ST_WINDER_DANCER_CONTROL_
CONFIG.iWebSpdPeriod,
ST_WINDER_DANCER_CONTROL_
CONFIG.iMotorSpeedOffsetAbs,
ST_WINDER_DANCER_CONTROL_
CONFIG.iMotorSpeedOffsetPrcnt</td></tr>
<tr><td>16</td><td>Einer der folgenden Parameter hat einen Wert <=0:
ST_WINDER_DANCER_CONTROL_
CONFIG.stPidConfig.iPidTick,
ST_WINDER_DANCER_CONTROL_
CONFIG.stPidConfig.iKp,
ST_WINDER_DANCER_CONTROL_
CONFIG.stPidConfig.iKv,
ST_WINDER_DANCER_CONTROL_
CONFIG.stPidConfig.diOutputScale,
ST_WINDER_DANCER_CONTROL_
CONFIG.stPidConfig.diImax,
ST_WINDER_DANCER_CONTROL_
CONFIG.stPidConfig.iDmPidSet</td></tr>
</table></td></tr>
</table>

Name	Typ	Beschreibung
		<table><tr><th>Wert</th><th>Bedeutung</th></tr><tr><td>17</td><td>Einer der folgenden Parameter hat einen Wert <=0: ST_WINDER_DANCER_CONTROL_ CONFIG.stDancerConfig.iDancerFullPosition, ST_WINDER_DANCER_CONTROL_ CONFIG.stDancerConfig.iDancerMidPosition, ST_WINDER_DANCER_CONTROL_ CONFIG.stDancerConfig.iDancerEmptyPosition, ST_WINDER_DANCER_CONTROL_ CONFIG.stDancerConfig.diTUpRamp, ST_WINDER_DANCER_CONTROL_ CONFIG.stDancerconfig.diTDownRamp</td></tr></table>

Ausgangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diVWebSpeed	DINT	Istwert Bahngeschwindigkeit [mm/s]
diSetValMotorSpeed	DINT	Drehzahlsollwert Motor [0,0001 U/min]

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)
stConfig	STRUCT	ST_WINDER_DANCER_CONTROL_CONFIG Konfigurationsstruktur des Wicklers (Siehe Dokument Softwarebeschreibung AFL - AMK Funktionsbibliothek Teil 2, Teile-Nr. 203694)

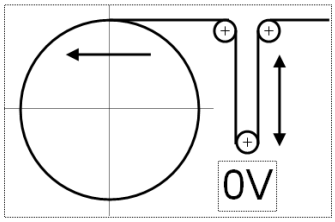
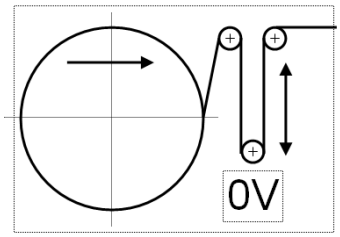
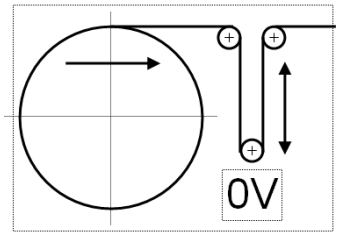
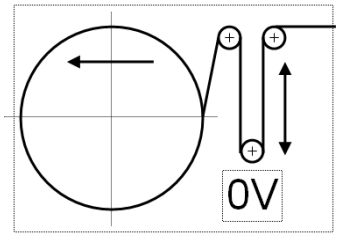
Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
WINDER_DANCER_CONTROL	WINDER_DANCER_CONTROL.actSync

ST_WINDER_DANCER_CONTROL_CONFIG (ST)

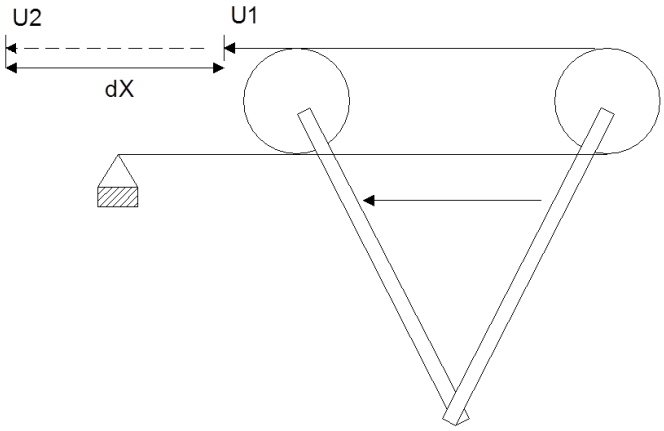
Konfigurationsstruktur

Name	Typ	Bsp. Wert	Beschreibung
boWindCWise	BOOL	FALSE	FALSE: Rolle wird aufgewickelt wenn Drehrichtung des Motors entgegen dem Uhrzeigersinn. TRUE: Rolle wird aufgewickelt wenn Drehrichtung des Motors im Uhrzeigersinn.
boUnwinder	BOOL	FALSE	FALSE: Wickler lauft als Aufwickler. Bei positiver Bahngeschwindigkeit wird aufgewickelt. TRUE: Wickler läuft als Abwickler. Bei positiver Bahngeschwindigkeit wird abgewickelt.

Anordnung	Typ	boWindCWise	boUnwinder
	Aufwickler	FALSE	FALSE
	Aufwickler	TRUE	FALSE
	Abwickler	FALSE	TRUE
	Abwickler	TRUE	TRUE

Name	Typ	Bsp. Wert	Beschreibung
boMotCcWise	BOOL	FALSE	Einbaulage des Motor entgegengesetzt zur Rolle
iGearInput	INT	1	Getriebe Eingang (vom Motor aus gesehen)
iGearOutput	INT	1	Getriebe Ausgang (vom Motor aus gesehen)
iDmMin	INT	100	Minimal zulässiger Durchmesserwert [mm]
iDmMax	INT	600	Maximal zulässiger Durchmesserwert [mm]
diMotIncPerRev	DINT	20000	Inkremente pro Motorumdrehung (ID116'Auflösung Motorgeber')
diDncComp	DINT		Tänzerkonstante → Spannungsdifferenz bei 1m Tänzerweg [mV/m]

Die Tänzerkonstante 'diDncComp' kann messtechnisch folgendermaßen ermittelt oder überprüft werden:



Durch den Tänzer wird ein Seil oder ein Draht gezogen.

Der Tänzer wird an eine untere Position gestellt. Jetzt wird die Position des Seilendes markiert und der Spannungswert des Tänzers als U1 notiert.

Danach wird der Tänzer an eine obere Position gestellt. Auch hier wird die Position des Seilendes markiert und der Spannungswert des Tänzers als U2 notiert (U2 > U1).

Der Abstand zwischen den Seilenden der beiden Tänzerpositionen wird gemessen und ergibt ΔX. Der Wert ΔX wird zusammen mit den zwei Spannungswerten U1 und U2 in die folgende Formel eingesetzt und 'diDncComp' berechnet:

Parameter	Einheit	Bedeutung
diDncComp	[mV/1000mm]	Tänzerkompensationskonstante
U1	[mV]	Spannung U1 an einer Tänzerposition X1
U2	[mV]	Spannung U2 an einer Tänzerposition X2
ΔX	[mm]	Differenz der Positionen X2 - X1

Beispiel:

Parameter	gemessener Wert	Einheit	Bedeutung
U1	3625	[mV]	Spannung U1 an einer Tänzerposition X1
U2	7453	[mV]	Spannung U2 an einer Tänzerposition X2
ΔX	800	[mm]	Differenz der Positionen X2 - X1

Bei der Messung ist darauf zu achten, dass der Tänzer nicht bis in die Endlagen bewegt wird, da dort meist ein nichtlinearer Zusammenhang zwischen aufgenommener Bahn und Spannungsdifferenz besteht.

Name	Typ	Bsp. Wert	Beschreibung
iMesCycDmOk	INT	10	Rollenumdrehung für Zyklus Durchmesserberechnung wenn Durchmesser bekannt [0,1 Umdrehungen]
iMesCycNoDm	INT	1	Rollenumdrehung für Zyklus Durchmesserberechnung wenn Durchmesser nicht bekannt [0,1 Umdrehungen]
iDmDiffOk	INT	10	Maximale Differenz zwischen zwei aufeinanderfolgend berechneten Durchmessern, damit die Bewertung "Durchmesser gültig" übernommen wird [mm]

Geschwindigkeitsparameter

Name	Typ	Bsp. Wert	Beschreibung
diWebEncIncr	DINT	20000	Anzahl der gezählten Inkremente am Eingang 'diInVal' bei zurücklegen der Strecke 'diWebEncLen' [Inkremente]

Name	Typ	Bsp. Wert	Beschreibung
diWebEncLen	DINT	1000	Zurückgelegte Bahnstrecke bei Änderung des Eingangs 'diInVal' um 'diWebEncIncr' Inkremente [mm]
iWebSpdPeriod	INT	5	'iWebSpdPeriod' gibt an, nach wie vielen ID2 'SERCOS-Zykluszeit' Zyklen die Bahngeschwindigkeit berechnet wird. Umso größer dieser Wert ist, umso ruhiger, aber auch träger wird das Wickelverhalten. Hier ist ein Kompromiss zu suchen (z. B. jede 50. Zyklus)

Tänzer- und Reglerparameter

Name	Typ	Bsp. Wert	Beschreibung
stDancerConfig	STRUCT		ST_WINDER_DANCER_CONFIG
stPidConfig	STRUCT		ST_WINDER_PID_CONFIG

Abstimmungsparameter

Name	Typ	Bsp. Wert	Beschreibung
diTimeDmRamp	DINT	2000	Rampenzeit für Durchmesserausgabe [ms] Bei sprunghafter Durchmesseränderung am Eingang (z.B. Setzen auf einen Presetwert) wird der Durchmesser linear in der eingestellten Rampenzeit angepasst.
diWebSpeedMax	DINT	20000	Maximale Bahngeschwindigkeit [mm/s]
diTimeWebSpeedRamp	DINT	5000	Bahngeschwindigkeitsrampe [ms] Die Bahngeschwindigkeit wird gerampt an die Drehzahlberechnung der Rolle übergeben

ST_WINDER_DANCER_CONFIG (ST)

Konfigurationsstruktur Tänzer

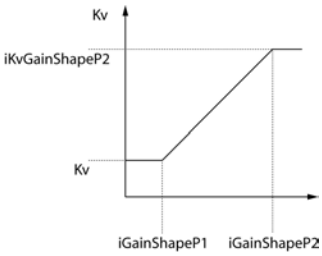
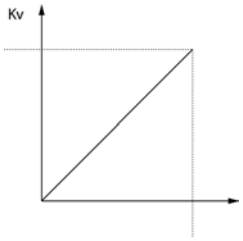
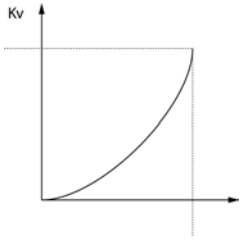
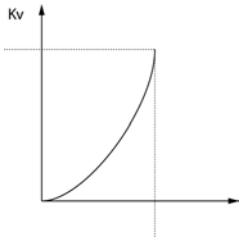
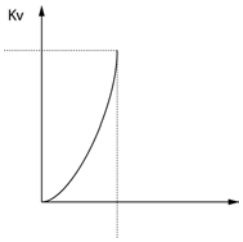
Name	Typ	Bsp. Wert	Beschreibung
iDancerFullPosition	INT	1000	Tänzerspannung, wenn der Tänzer mit Material gefüllt ist [mV]
iDancerMidPosition	INT	5000	Tänzerspannung in der Mittelstellung des Tänzers (Arbeitspunkt) [mV]
iDancerEmptyPosition	INT	9000	Tänzerspannung, wenn der Tänzer leer ist [mV]
diTUpRamp	DINT	1000	Rampenzeit für die Bewegung des Tänzers in die Arbeitspunktposition [ms]
diTDownRamp	DINT	3000	Rampenzeit für die Bewegung des Tänzers in Ablageposition (Tänzer gefüllt) [ms]

ST_WINDER_PID_CONFIG (ST)

Konfigurationsstruktur PID-Regler

Name	Typ	Bsp. Wert	Beschreibung
iPidTick	INT	10	Aufruf Takt des PID-Regler [iPidTick*ID2]
iKp	INT	5000	Kp für PID-Regler [-]
iKi	INT	10	Ki für PID-Regler [-]
iKd	INT	0	Kd für PID-Regler [-]

Name	Typ	Bsp. Wert	Beschreibung
iKv	INT	25	Gesamtverstärkung des PID-Reglers: Bei aktivierter tänzerlage abhängiger Verstärkung ist Kv die Verstärkung bei (Tänzersollwert - Tänzeristwert) < iGainShapeP1 [-]
diOutputScale	DINT	1	Maßstab Drehzahlausgang [-]
iPosKiActivate	INT	2500	Tänzerposition für die Aktivierung von Ki [mV]
diImax	DINT	20000	Maximal zulässiger Integralanteil [U/10000 min]
iDmPidSet	INT	600	Rollendurchmesser, bei dem die Parameter 'iKp', 'iKi', 'iKd' und 'iKv' des PID-Reglers eingestellt wurden [mm]

Name	Typ	Bsp. Wert	Beschreibung
iGainShapeType	INT		Typ der Verstärkeranpassung Wert = 0:  Tänzerspannungsdifferenz [mV] = (Tänzeristposition - Tänzersollposition) Wert = 1: lineare Verstärkungskennlinie  Tänzerspannungsdifferenz [mV] = (Tänzeristposition - Tänzersollposition) Wert =2: quadratische Verstärkungskennlinie  Tänzerspannungsdifferenz [mV] = (Tänzeristposition - Tänzersollposition) Wert =3: Verstärkungskennlinie[3]  Tänzerspannungsdifferenz [mV] = (Tänzeristposition - Tänzersollposition) Wert = 4: Verstärkungskennlinie [4]  Tänzerspannungsdifferenz [mV] = (Tänzeristposition - Tänzersollposition)
iGainShapeP1	INT	1000	(Tänzeristwert - Tänzersollwert) = iGainShapeP1 Startpunkt der Verstärkungsrampe [mV]
iGainShapeP2	INT	3000	(Tänzeristwert - Tänzersollwert) = iGainShapeP1 Endpunkt der Verstärkungsrampe [mV]

Name	Typ	Bsp. Wert	Beschreibung
iKvGainShapeP2	INT	50	Verstärkung bei (Tänzeristwert - Tänzersollwert) > iGainShapeP2

4.2.1.2.3.15 WINDER_TORQUE_CONTROL (FB)

Der Funktionbaustein 'WINDER_TORQUE_CONTROL' realisiert einen sensorlosen Auf- und Abwickler.

Der Wickler hat folgende Merkmale:

- Momentgesteuerter Zentrumswickler
- Betrieb als Auf- und Abwickler
- Durchmesserberechnung
- Durchmessersuche bei Start des Wicklers
- Ermittlung des Rollenträgheitsmoments bei unbekanntem Durchmesser
- Synchronisieren auf eine laufende Bahn

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Anwender Interface

	WINDER_TORQUE_CONTROL			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Aktivierung der Durchmesserrechnung -	boDmCalc	boDmOk	-	Durchmesser ok
Ermittlung Rollenträgheitsmoment ein -	boFindInertia	boFindInertiaDone	-	Trägheitsmoment ermittelt
Wickeln ein -	boWind	boWindRuns	-	Wickeln läuft
Synchronisieren ein -	boSync	boSyncRuns	-	Synchronisation läuft
Durchmesser setzen -	boDmPreset	boDmLimit	-	Durchmesserwert wurde begrenzt
Durchmesser setzen, Status "Bekannt" -	boDmOkPreset	iActDm	-	Istwert Rollendurchmesser
Durchmesser setzen, Status "unbekannt" -	boDmNOkPreset	diActValWebSpeed	-	Istwert Bahngeschwindigkeit
Durchmesser Status von "Bekannt" → "Unbekannt" setzen -	boDmReset	boErr	-	Fehlersignal
Ausgang 'boDmOk' setzen -	boSetDmOk	iErrID	-	Fehler Identnummer
Sollwert Bahnspannung -	diWebTension			
Durchmesservorgabewert -	iDmPreset			
PGT Takt -	uiID2			
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

	actSync			
Istwert Motordrehzahl -	diActMotorSpeed	diSetValMotorSpeed	-	Drehzahlsollwert Motor
Istwert Geberposition Motor -	diActReelPos	iSetValTorqueAbs	-	Grenzmomentsollwert Motor
Aktuelle Bahnposition -	diActWebPos	diActMotorSpeedAbs	-	Aktuelle Motordrehzahl absolut
		diActReelPosAbs	-	Aktuelle Rollenposition absolut

Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boDmCalc	BOOL	Aktivierung der Durchmesserberechnung
boFindInertia	BOOL	Ermittlung des Trägheitsmoments der Rolle und Bahnstraffung ein
boWind	BOOL	Wickeln ein
boSync	BOOL	Synchronisieren ein
boDmPreset	BOOL	Durchmesser auf Wert des Eingangs 'iDmPreset' setzen. Das Eingangsbit wird nach setzen automatisch rückgesetzt.
boDmOkPreset	BOOL	Durchmesser auf Wert der Eingangsvariable 'iDmPreset' setzen. Das Eingangsbit 'boDmPreset' wird nach setzen automatisch rückgesetzt. Durchmesserstatus "Bekannt": Der Durchmesserwert wird als gültiger Durchmesser gewertet, das bedeutet, Beschleunigungs- und Reibmomente werden berechnet.
boDmNOkPreset	BOOL	Durchmesser auf Wert der Eingangsvariable 'iDmPreset' setzen. Das Eingangsbit 'boDmPreset' wird nach setzen automatisch rückgesetzt. Durchmesserstatus "Unbekannt": Der Durchmesserwert wird als ungültiger Durchmesser gewertet, das bedeutet, Beschleunigungs- und Reibmomente werden berechnet.
boDmReset	BOOL	Ausgang 'boDmOk' rücksetzen
boSetDmOk	BOOL	Ausgang 'boDmOk' setzen
diWebTension	DINT	Absolutes Bahnspannungsmoment [0,1 Nm]
iDmPreset	INT	Durchmesservorgabewert Wert auf den der Durchmesser bei 'boDmNokPreset' = TRUE oder 'boDmOkPreset' = TRUE gesetzt wird [mm]
uiID2	UINT	PGT-Takt [ms]

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diActMotorSpeed	DINT	Drehzahlistwert Motor [0,0001 U/min]
diActReelPos	DINT	Istposition Rolle [Inkremente]
diActWebPos	DINT	Istposition Bahn [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boDmOk	BOOL	Durchmesser bekannt
boFindInertiaDone	BOOL	Wickelmoment ermittelt, Bahn gestrafft
boWindRuns	BOOL	Wickeln läuft
boSyncRuns	BOOL	Synchronisation läuft

Name	Typ	Beschreibung
boDmLimit	BOOL	Durchmesser wurde berechnet 'ST_WINDER_TORQUE_CONTROL_CONFIG.iDmMin' und 'ST_WINDER_TORQUE_CONTROL_CONFIG.iDmMax', der Wert wurde begrenzt
iActDm	INT	Aktueller Durchmesser [mm]
diActValWebSpeed	DINT	Istwert Bahngeschwindigkeit [mm/s]
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>iErrID</td><td>INT</td><td>Fehler Identnummer: Diagnosenummer wird ausgegeben
<table>
<tr><td colspan="2">iErrID = 0</td><td>Kein Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = TRUE</td><td>Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = FALSE</td><td>Warnung</td></tr>
</table>
<table>
<tr><th>Wert</th><th>Bedeutung</th></tr>
<tr><td>1</td><td>Der Wert am Eingang uiID2 = 0</td></tr>
<tr><td>2</td><td>Einer der folgenden Parameter hat einen Wert <=0:
ST_WINDER_TORQUE_CONTROL_CONFIG.iGearInput,
ST_WINDER_TORQUE_CONTROL_
CONFIG.iGearOutput</td></tr>
<tr><td>3</td><td>Einer der folgenden Parameter hat einen Wert <=0:
ST_WINDER_TORQUE_CONTROL_CONFIG.iDmMin,
ST_WINDER_TORQUE_CONTROL_CONFIG.iDmMax
oder
ST_WINDER_TORQUE_CONTROL_CONFIG.iDmMin >=
ST_WINDER_TORQUE_CONTROL_CONFIG.iDmMax</td></tr>
<tr><td>4</td><td>Der Parameter hat einen Wert <=0:
ST_WINDER_TORQUE_CONTROL_
CONFIG.diMotIncPerRev</td></tr>
<tr><td>5</td><td>Einer der folgenden Parameter hat einen Wert <=0:
ST_WINDER_TORQUE_CONTROL_
CONFIG.iMesCycDmOk,
ST_WINDER_TORQUE_CONTROL_
CONFIG.iMesCycNoDm,
ST_WINDER_TORQUE_CONTROL_CONFIG.iDmDiffOk</td></tr>
<tr><td>6</td><td>Der Parameter hat einen Wert <=0:
ST_WINDER_TORQUE_CONTROL_
CONFIG.iSyncTorque</td></tr>
<tr><td>7</td><td>Einer der folgenden Parameter hat einen Wert <=0:
ST_WINDER_TORQUE_CONTROL_CONFIG.iWidth,
ST_WINDER_TORQUE_CONTROL_CONFIG.iDensity</td></tr>
<tr><td>8</td><td>Der Parameter hat einen Wert <=0:
ST_WINDER_TORQUE_CONTROL_
CONFIG.diFricitonSpeedMax</td></tr>
<tr><td>9</td><td>Einer der folgenden Parameter hat einen Wert <=0:
ST_WINDER_TORQUE_CONTROL_
CONFIG.diWebSpeedMax,
ST_WINDER_TORQUE_CONTROL_
CONFIG.diWebEncLen,
ST_WINDER_TORQUE_CONTROL_
CONFIG.diWebEncIncr,
ST_WINDER_TORQUE_CONTROL_
CONFIG.iWebSpdPeriod,
ST_WINDER_TORQUE_CONTROL_
CONFIG.iMotorSpeedOffsetAbs,
ST_WINDER_TORQUE_CONTROL_
CONFIG.iMotorSpeedOffsetPrcnt</td></tr>
<tr><td>10</td><td>Der Parameter hat einen Wert <=0:
ST_WINDER_TORQUE_CONTROL_
CONFIG.iRatedTorque</td></tr>
</table>
</td></tr>
</table>

Name	Typ	Beschreibung
		<table><tr><th>Wert</th><th>Bedeutung</th></tr><tr><td>11</td><td>Baustein WINDER_CALC_INERTIA: Fehler bei Berechnung des Rollenträgheitsmoments. Eine der Variablen 'iDensity' oder 'iWidth' hat einen Wert <=0</td></tr><tr><td>12</td><td>Baustein WINDER_FRICTION_CALC: Eine der Eingangsvariablen 'iRatedTorque' oder 'diSpeedMax' hat einen Wert <=0</td></tr><tr><td>13</td><td>Baustein WINDER_WINDER_CT: Bei der Funktion "Ermittlung des Trägheitsmoments der Rolle und Bahnstraffung" wurde das maximal zulässige Moment des Parameters 'ST_WINDER_TORQUE_CONTROL_CONFIG.iFiMaxTorque', erreicht.</td></tr><tr><td>14</td><td>Baustein WINDER_TORQUE_CONTROL: Bei der Funktion "Ermittlung des Trägheitsmoments der Rolle und Bahnstraffung" wurde die maximal zulässige Zeit des Parameters ST_WINDER_TORQUE_CONTROL_CONFIG.tFiTightTout überschritten. Die Rolle konnte die Bahn nicht straffen. Wahrscheinlich ist die Bahn gerissen.</td></tr><tr><td>15</td><td>Einer der folgenden Parameter hat einen Wert <=0: ST_WINDER_TORQUE_CONTROL_CONFIG.udFiSlackTorque, ST_WINDER_TORQUE_CONTROL_CONFIG.stConfig.udFiSlackVelocity, ST_WINDER_TORQUE_CONTROL_CONFIG.stConfig.udFiTramp, ST_WINDER_TORQUE_CONTROL_CONFIG.stConfig.iFiMaxTorque, ST_WINDER_TORQUE_CONTROL_CONFIG.stConfig.iFiTightSpeed, ST_WINDER_TORQUE_CONTROL_CONFIG.stConfig.iFiSpeedThresholdPerc ST_WINDER_TORQUE_CONTROL_CONFIG.stConfig.iFiSpeed0Perc ST_WINDER_TORQUE_CONTROL_CONFIG.stConfig.iFiTightWnd</td></tr><tr><td>18</td><td>Baustein WINDER_CHARACTERISTICS: Einer der Durchmesserwerte ist kleiner als sein Vorgängerwert</td></tr></table>

Ausgangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diSetValMotorSpeed	DINT	Drehzahlsollwert Motor [0,0001 U/min]
iSetValTorqueAbs	INT	Grenzmomentsollwert Motor [0,1% Mn]
diActMotorSpeedAbs	DINT	Aktuelle Motordrehzahl absolut [0,0001 U/min]
diActReelPosAbs	DINT	Aktuelle Rollenposition absolut [Inkremente]

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Name	Typ	Beschreibung
stConfig	STRUCT	ST_WINDER_TORQUE_CONTROL_CONFIG Konfigurationsstruktur des Wicklers (Siehe Dokument Softwarebeschreibung AFL - AMK Funktionsbibliothek Teil 2, Teile-Nr. 203694)

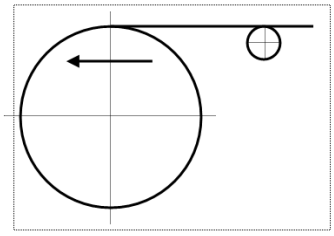
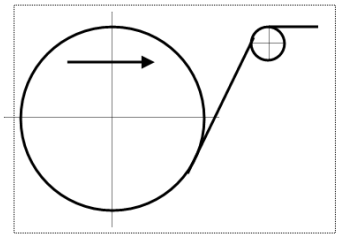
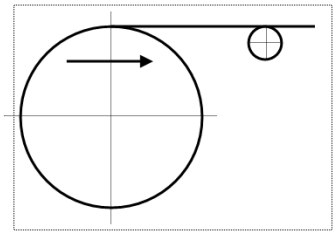
Verwendungshinweis im CoDeSys-Programm

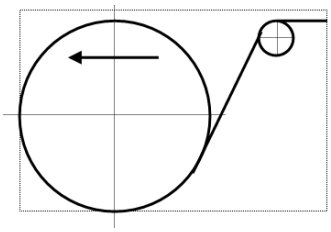
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
WINDER_TORQUE_CONTROL	WINDER_TORQUE_CONTROL.actSync

ST_WINDER_TORQUE_CONTROL_CONFIG (ST)

Konfigurationsstruktur

Name	Typ	Bsp. Wert	Beschreibung
boWndCWise	BOOL	FALSE	FALSE: Rolle wird aufgewickelt wenn Drehrichtung des Motors entgegen dem Uhrzeigersinn. TRUE: Rolle wird aufgewickelt wenn Drehrichtung des Motors im Uhrzeigersinn.
boUnwinder	BOOL	FALSE	FALSE: Wickler lauft als Aufwickler. Bei positiver Bahngeschwindigkeit wird aufgewickelt. TRUE: Wickler läuft als Abwickler. Bei positiver Bahngeschwindigkeit wird abgewickelt.

Anordnung	Typ	boWndCWise	boUnwinder
	Aufwickler	FALSE	FALSE
	Aufwickler	TRUE	FALSE
	Abwickler	FALSE	TRUE

Anordnung	Typ	boWndCWise	boUnwinder
	Abwickler	TRUE	TRUE

Im Betrieb wirkt sich eine Drehrichtungsumkehr am Vorzug direkt auf den Wickler aus. Die Drehrichtung am Wickler dreht automatisch. Es ist keine Änderung der Konfiguration nötig.

Name	Typ	Bsp. Wert	Beschreibung
boMotCcWise	BOOL	FALSE	Einbaulage des Motor entgegen dem Uhrzeigersinn
iGearInput	INT	1	Getriebe Eingang multiplikator Wicklerantrieb
iGearOutput	INT	1	Getriebe Ausgang teiler Wicklerantrieb
iDmMin	INT	100	Minimal zulässiger Durchmesserwert [mm]
iDmMax	INT	600	Maximal zulässiger Durchmesserwert [mm]
diMotIncPerRev	DINT		Inkremente pro Motorumdrehung (ID116'Auflösung Motorgeber')
iMesCycDmOk	INT	10	Rollenumdrehung für Zyklus Durchmesserberechnung wenn Durchmesser bekannt [0,1 Umdrehungen]
iMesCycNoDm	INT	1	Rollenumdrehung für Zyklus Durchmesserberechnung wenn Durchmesser nicht bekannt [0,1 Umdrehungen]
iDmDiffOk	INT	20	Maximale Differenz zwischen zwei aufeinanderfolgend berechneten Durchmessern, damit die Bewertung "Durchmesser gültig" übernommen wird [mm]

Rollensynchronisation

Name	Typ	Bsp. Wert	Beschreibung
udSyncTramp	UDINT	40000	Steigung der Synchronisationsrampe [ms]
iSyncTorque	INT	1000	Moment während der Synchronsation [0,1% Mn]

Parameter für das berechnen des Trägheitsmoments der Rolle

Name	Typ	Bsp. Wert	Beschreibung
iWidth	INT		Bahnbreite [mm]
iDensity	INT	800	Bahndichte [kg/m^3]

Parameter für die Reibungskompensation

Name	Typ	Bsp. Wert	Beschreibung
diFrictionTorque0	DINT		Reibmoment bei Drehzahl 0 [0,1 Nm]
diFrictionSpeedMax	DINT		Drehzahl bei dem Reibmoment 'diFrictionTorqueMax' wurde ermittelt [0,1 U/min]
diFrictionTorqueMax	DINT		Max. Reibmoment bei Drehzahl 'diFrictionSpeedMax' [0,1 Nm]

Name	Typ	Bsp. Wert	Beschreibung
uiFrcSetSpeedLimitPos	UINT	50	Positive Hysteresegrenze für die Berechnung des Reibmoments [0,1 U/min]
uiFrcSetSpeedLimitNeg	UINT	50	Negative Hysteresegrenze für die Berechnung des Reibmoments [0,1 U/min]

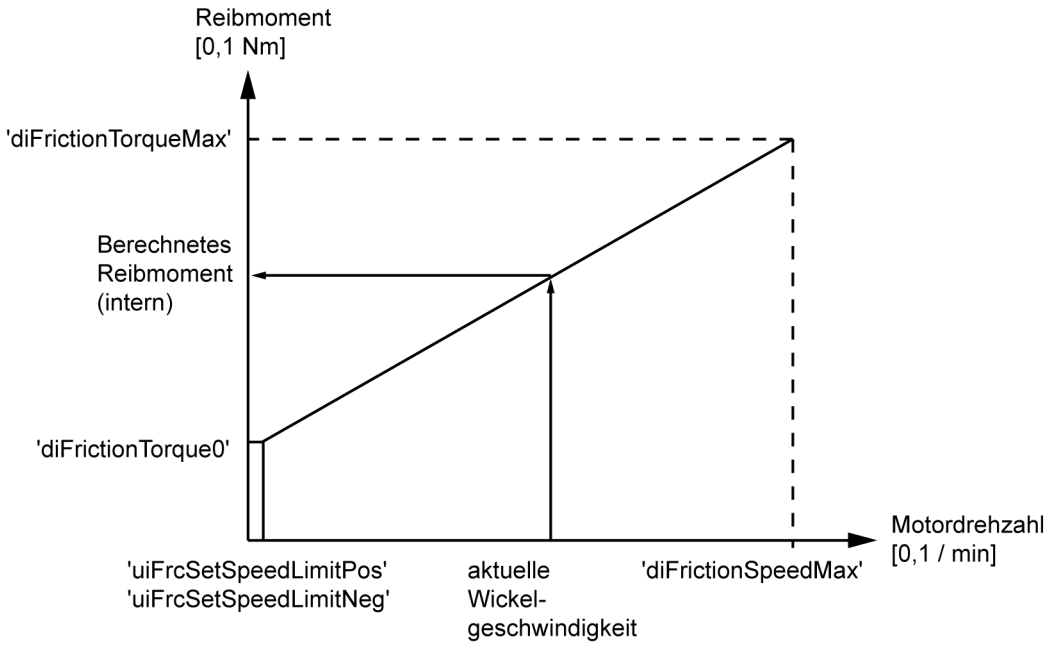
Reibungskompensation

Der Wickelbaustein kompensiert die Reibung während des Auf- und Abwickelns. Die Kompensation wird linear als Funktion der Motordrehzahl durchgeführt.

Reibmoment ermitteln

Das Reibmoment kann experimentell ermittelt werden, indem der Motor mit einer konstanten Drehzahl eine kleine Rolle dreht und das dazu erforderliche Moment (ID84 'Moment Istwert') ausgelesen wird. Der Wert des Parameters 'diFrictionTorqueMax' ist das Moment bei maximaler Drehzahl 'diFrictionSpeedMax'.

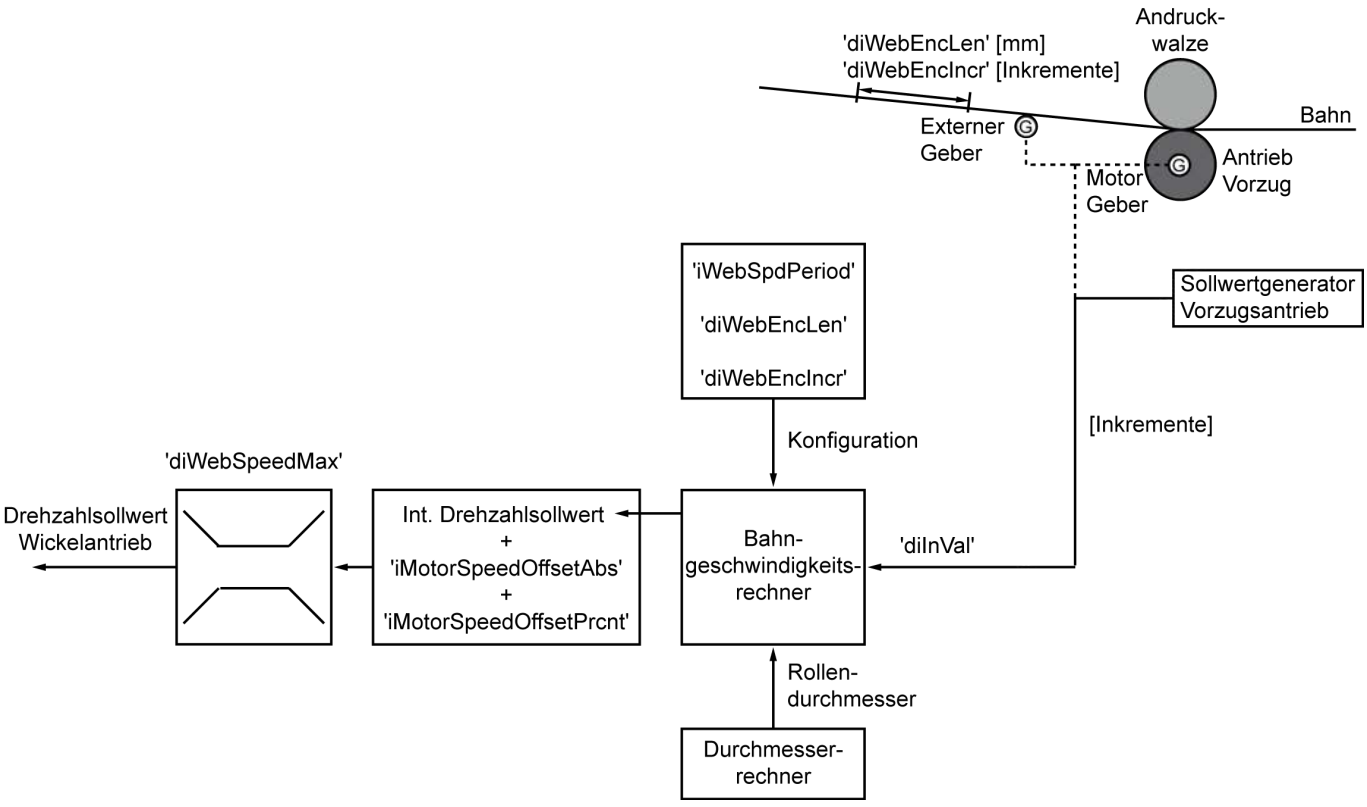
Im Stillstand kann es den Effekt geben, dass die Bahngeschwindigkeit Werte liefert, die ständig um den 0-Punkt schwanken. Die Ausgabe des Reibmoments würde dadurch zu einem Schwingen der Bahn führen. Um das zu vermeiden wird die Bahngeschwindigkeit zur Berechnung des Reibmoments mit einer Hysterese bewertet. Die Hysteresegrenzen sind mit 5 U/min voreingestellt.



Geschwindigkeitsparameter

Name	Typ	Bsp. Wert	Beschreibung
diWebSpeedMax	DINT	20000	Maximale Bahngeschwindigkeit [mm/s] Der Bahngeschwindigkeitsrechner limitiert den Drehzahlsollwert auf diesen Wert. Z. B. maximale Prozessgeschwindigkeit.
diWebEncLen	DINT	1000	Zurückgelegte Bahnstrecke bei Änderung des Eingangs 'diInVal' um 'diWebEncIncr' Inkremente [mm] z. B. 1000 mm Bahn 'diWebEncLen' entsprechen 500.000 Inkr. 'diWebEncIncr' am Eingang 'diInVal'

Name	Typ	Bsp. Wert	Beschreibung
diWebEncIncr	DINT		Anzahl der gezählten Inkremente am Eingang 'diInVal' bei zurücklegen der Strecke 'diWebEncLen' [Inkremente] z. B. 500.000 Inkr. 'diWebEncIncr' am Eingang 'diInVal' entsprechen 1000 mm Bahn 'diWebEncLen'
iWebSpdPeriod	INT	50	'iWebSpdPeriod' gibt an, nach wie vielen ID2 'SERCOS-Zykluszeit' Zyklen die Bahngeschwindigkeit berechnet wird. Umso größer dieser Wert ist, umso ruhiger, aber auch träger wird das Wickelverhalten. Hier ist ein Kompromiss zu suchen (z. B. jede 50. Zyklus)
iMotorSpeedOffsetAbs	INT	200	Offset Drehzahl absolut [0,1 U/min] Drehzahlsollwert um die Bahn zu spannen. Bei der Funktion Bahnspannen wirkt 'iMotorSpeedOffsetPrcnt' nicht, da der Wert 'diInVal' unverändert bleibt, weil der Vorzugsantrieb steht.
iMotorSpeedOffsetPrcnt	INT	20	Offset Drehzahl prozentual [%] Offset Motordrehzahl, prozentual bezogen auf den Internen Drehzahlsollwert [%] Der Drehzahlsollwert Wickelantrieb muss höher sein als der berechnete interne Drehzahlsollwert, damit die Bahnspannung aufgebaut werden kann.
iRatedTorque	INT		Motor Nennmoment [0,1 Nm] ID32771 'Nenndrehmoment'



Parameter für die Ermittlung des Rollenträgheitsmoments

Folgende Parameter können zu Optimierung des Ablaufes der Ermittlung des Rollenträgheitsmoments bei unbekanntem Rollendurchmesser geändert werden:

Name	Typ	Bsp. Wert	Beschreibung
udFiSlackTorque	UDINT	1000	Sollwert Motormoment während der Ermittlung des Rollenträgheitmoments [0,1% Mn]
udFiSlackVelocity	UDINT	50	Drehzahlsollwert zum zurückdrehen der Rolle zur Ermittlung des Rollenträgheitsmoments [0,1 U/min]
udFiTramp	UDINT	1000	Beschleunigungsrampe für das Zurückdrehen der Rolle [ms]
tFiTimeStep	TIME	500	Zeitintervall bis Erhöhung des Moments während der Ermittlung des Rollenträgheitsmoments [ms]
iFiTorqueStep	INT	50	Momentschritterhöhung während der Ermittlung des Rollenträgheitsmoments [0,1% Mn]
iFiMaxTorque	INT	1000	Maximales Moment während der Ermittlung des Rollenträgheitsmoments [0,1% Mn]
iFiTightSpeed	INT	50	Drehzahlsollwert zum Straffen der Bahn [0,1 U/min]
iFiSpeedThresholdPerc	INT	90	Drehzahlschwelle zum Straffen der Bahn [%iFiTightSpeed]
iFiSpeed0Perc	INT	60	Schwelle Drehzahl = 0 zum Straffen der Bahn [%iFiTightSpeed]
tFiTightTout	TIME	120	Timeout bei Straffen der Bahn [s]
iFiTightWnd	INT	20	Positionsfenster für Straffen der Bahn, wenn Rolle steht und Rollenposition ist im Fenster, gilt die Bahn als gestrafft [1/U]

Abstimmungsparameter

Name	Typ	Bsp. Wert	Beschreibung
diTimeDmRamp	DINT	2000	Rampenzeit für Durchmesserausgabe [ms] Bei sprunghafter Durchmesseränderung am Eingang (z.B. Setzen auf einen Presetwert) wird der Durchmesser linear in der eingestellten Rampenzeit angepasst.

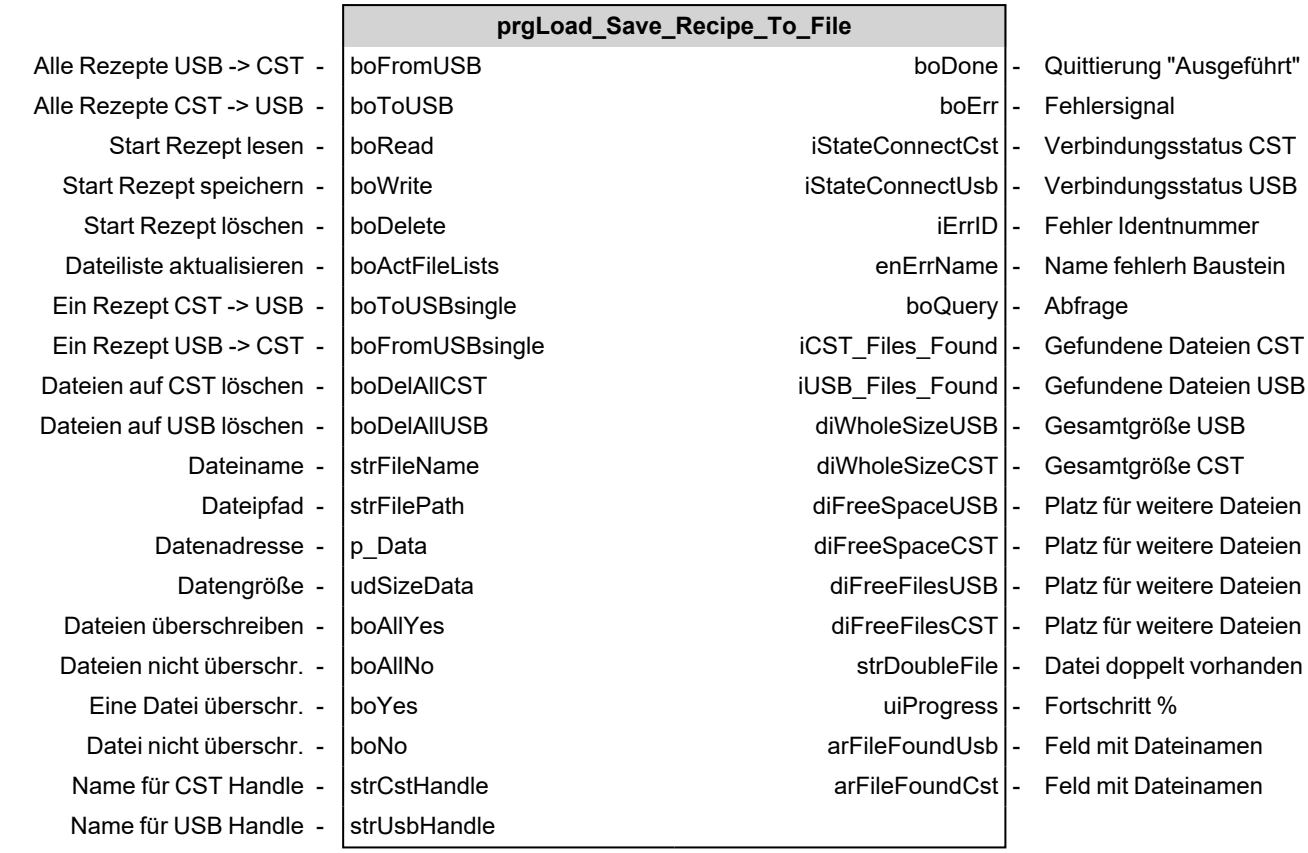
4.2.1.2.4 04_Recipe

4.2.1.2.4.1 prgLoad_Save_Recipe_To_File (PRG)

Über die Eingänge des Programms 'prgLoad_Save_Recipe_To_File' können Rezeptdaten abgespeichert, gelesen und/oder auf und von USB kopiert werden. Außerdem können die gespeicherten Rezeptdaten wieder gelöscht werden.

Dieses Programm sollte in einer eigenen freilaufenden Task aufgerufen werden.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boFromUSB	BOOL	Alle Rezeptdateien von USB auf CST kopieren
boToUSB	BOOL	Alle Rezeptdateien von CST auf USB kopieren
boRead	BOOL	Ausgewählte Datei ('strFileName') und Rezept auslesen (nur CST)
boWrite	BOOL	Rezept in ausgewählte Datei ('strFileName') abspeichern (nur CST)
boDelete	BOOL	Ausgewählte Datei ('strFileName') löschen (nur CST)
boActFileLists	BOOL	Dateilisten aktualisieren ('arFileFoundUsb' und 'arFileFoundCst')
boToUSBsingle	BOOL	Ausgewählte ('strFileName') Rezeptdatei auf USB kopieren
boFromUSBsingle	BOOL	Ausgewählte ('strFileName') Rezeptdatei auf CST kopieren
boDelAllCST	BOOL	Alle Rezeptdateien auf CST löschen
boDelAllUSB	BOOL	Alle Rezeptdateien auf USB löschen
strFileName	STRING	Dateiname der Datei ohne -endung.
strFilePath	STRING	Pfad (USB und CST), in der die Rezeptdateien gespeichert werden sollen (‘/’ als Trennzeichen)
p_Data	POINTER	POINTER TO BYTE Pointer auf die Rezeptstruktur; Daten, die abgespeichert oder geladen werden sollen
udSizeData	UDINT	Datengröße (p_Data) [Byte]
boAllYes	BOOL	Wenn über 'boQuery' abgefragt wird: alle doppelten Dateien überschreiben
boAllNo	BOOL	Wenn über 'boQuery' abgefragt wird: alle doppelten Dateien nicht überschreiben

Name	Typ	Beschreibung
boYes	BOOL	Wenn über 'boQuery' abgefragt wird: die doppelte Datei ('strDoubleFile') überschreiben
boNo	BOOL	Wenn über 'boQuery' abgefragt wird: die doppelte Datei ('strDoubleFile') nicht überschreiben
strCstHandle	STRING	Name für CST-Handle Standardeinstellung: strCstHandle = 'CST_1'
strUsbHandle	STRING	Name für USB-Handle **Steuerung** / **SW-Version** / **strUsbHandle** A4 / ≥ A4_406_1151_203984 / 'USB11_1' A5 / < A5_410_ / 'USB21_2' A5 / ≥ A5_410_ / 'USB11_1' ASC / ≥ AS_CP_200_0812_202053 / 'USB11_2' Standardeinstellung: strUsbHandle = 'USB21_2'

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iStateConnectCst	INT	Verbindungszustand zum CST 0: Verbindung erfolgreich hergestellt 1: Falsches logisches Laufwerk 2: Das Laufwerk fehlt (z.B. USB-Stick ist nicht gesteckt) 3: Falsche physikalische Adresse (z.B. für EXT-Laufwerk ohne physikalische Adresse) 4: Kein erfolgreiches Herstellen der Verbindung 5: Verbindung existiert bereits; beim erneuten Versuch konnte die Verbindung nicht wieder aufgebaut werden
iStateConnectUsb	INT	Verbindungszustand zum USB-Stick 0: Verbindung erfolgreich hergestellt 1: Falsches logisches Laufwerk 2: Das Laufwerk fehlt (z.B. USB-Stick ist nicht gesteckt) 3: Falsche physikalische Adresse (z.B. für EXT-Laufwerk ohne physikalische Adresse) 4: Kein erfolgreiches Herstellen der Verbindung 5: Verbindung existiert bereits; beim erneuten Versuch konnte die Verbindung nicht wieder aufgebaut werden
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0 / boErr = TRUE: Fehler iErrID ≠ 0 / boErr = FALSE: Warnung enErrName = NONE: 1: Speicherplatz nicht ausreichend enErrName ≠ NONE: Fehlernummer s. Baustein

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein** / **Bibliothek** WRITE_FILE_1 / AmkFile.lib READ_FILE_1 / AmkFile.lib FIND_FILE_1 / AmkFile.lib REMOVE_FILE_1 / AmkFile.lib CREATE_DIR_1 / AmkFile.lib FILE_FIND / AmkAfl.Lib
boQuery	BOOL	boQUERY = TRUE: Abfrage, ob die Datei ('strDoubleFile') überschrieben werden soll. Muss mittels 'boYes', 'boNo', 'boAllYes' oder 'boAllNo' beantwortet werden.
iCST_Files_Found	INT	Anzahl der gefundenen Rezeptdateien auf CST
iUSB_Files_Found	INT	Anzahl der gefundenen Rezeptdateien auf USB
diWholeSizeUSB	DINT	Gesamtgröße der Rezeptdateien auf USB [Byte]
diWholeSizeCST	DINT	Gesamtgröße der Rezeptdateien auf CST [Byte]
diFreeSpaceUSB	DINT	Freier Speicher auf USB [kByte]
diFreeSpaceCST	DINT	Freier Speicher auf CST [kByte]
diFreeFilesUSB	DINT	Anzahl der Dateien, die noch auf USB passen (bei 1kByte Clusters)
diFreeFilesCST	DINT	Anzahl der Dateien, die noch auf CST passen (bei 1kByte Clusters)
strDoubleFile	STRING	Name der Rezeptdatei, die schon vorhanden ist und überspielt werden soll
uiProgress	UINT	Fortschritt: Zeigt den Fortschritt prozentual an
arFileFoundUsb	ARRAY	ARRAY[0..MAX_LEN_ST_FILE] OF ST_FILE Feld mit allen gefundenen Rezeptdateien auf USB
arFileFoundCst	ARRAY	ARRAY[0..MAX_LEN_ST_FILE] OF ST_FILE Feld mit allen gefundenen Rezeptdateien auf CST

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
prgLoad_Save_Recipe_To_File	

Es empfiehlt sich, Bausteine, die mit Dateizugriff arbeiten, in einer separaten Task in der asynchronen Programmebene PLC_PRG (Typ "Freilaufend") aufzurufen.

4.2.1.2.5 05_Alarm

4.2.1.2.5.1 Alarmtabelle

Die Bausteine und Visualisierungen dienen der Alarmbehandlung auf AMK-Steuerungen.

Um die Funktionalität zu realisieren, muss das Programm 'PRG_ERR_HANDLER' in einer eigenen freilaufenden Task aufgerufen werden.

Der Funktionsbaustein 'FB_READ_ERR_HISTORIE' dient dem Auslesen von Historiedateien.

Die Visualisierung 'VI_ALARMTAB' zeigt die aktuellen Fehler die durch 'fboSetErr' hinzugefügt wurden.

Die vergangenen Fehler kann man in der Visualisierung 'VI_HISTORIE' nachschauen.

4.2.1.2.5.2 FILE_MANAGER

fbyReadFile (F)

Die Funktion 'fbyReadFile' liest den Inhalt einer Datei aus. Sie meldet ihren Zustand zurück.

Anders als die beiden Funktionsbausteine 'READ_FILE' und 'READ_FILE_1' der AmkFile.lib wird hier auf die SysLibFile.lib zurückgegriffen.

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	fbyReadFile		
Dateiname -	strFileName	fbyReadFile -	Zustand
Pointer auf Datenpuffer -	pbyDest		
Größe des Datenpuffers -	dwSize		

Eingangsvariablen

Name	Typ	Beschreibung
strFileName	STRING	Dateiname der Datei incl. -endung, die ausgelesen werden soll
pbyDest	POINTER	POINTER TO BYTE Pointer auf den Buffer, in dem das Gelesene gespeichert werden soll
dwSize	DWORD	Buffergröße [Byte]

Ausgangsvariable

Name	Typ	Beschreibung
fbyReadFile	BYTE	Zustand 0 – Undefinierter Zustand 1 – Datei wurde erfolgreich gelesen 10 – Falsche Eingabe Parameter 11 – Fehler beim Öffnen der Datei 14 – Fehler beim Schließen der Datei 15 – Fehler beim Lesen

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fbyReadFile	

Es empfiehlt sich, Bausteine, die mit Dateizugriff arbeiten, in einer separaten Task in der asynchronen Programmebene PLC_PRG (Typ "Freilaufend") aufzurufen.

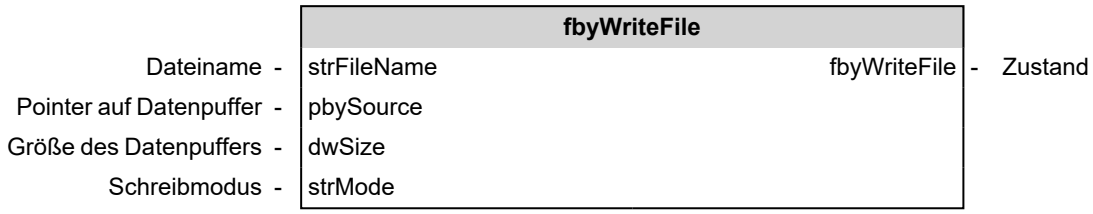
fbyWriteFile (F)

Die Funktion 'fbyWriteFile' schreibt in eine Datei und gibt ihren Zustand zurück.

Anders als die beiden Funktionsbausteine 'WRITE_FILE' und 'WRITE_FILE_1' der AmkFile.lib wird hier auf die SysLibFile.lib zurückgegriffen.

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
strFileName	STRING	Dateiname der Datei incl. -endung, die geschrieben werden soll
pbySource	POINTER	POINTER TO BYTE Pointer auf den Buffer, dessen Inhalt in die Datei geschrieben werden soll
dwSize	DWORD	Gibt die Größe des Datenspeichers an [Byte]
strMode	STRING(2)	Zugriffsmodus der Datei Modus, in dem die Datei bearbeitet werden soll: w – write – Datei wird überschrieben oder neu angelegt r – read – Datei wird nur zum Lesen geöffnet; wenn die Datei nicht existiert, wird ein Fehler zurückgemeldet. rw – read and write – Datei wird überschrieben; wenn die Datei nicht existiert, wird ein Fehler zurückgemeldet a – append – Datei wird wie mit 'w' geöffnet, aber beim Schreiben wird am Dateiende angehängt

Ausgangsvariable

Name	Typ	Beschreibung
fbyWriteFile	BYTE	Zustand 0 – Undefinierter Zustand 1 – Datei wurde erfolgreich geschrieben 10 – Falsche Eingabe Parameter 11 – Fehler beim Öffnen der Datei 13 – Fehler beim Schreiben der Datei 14 – Fehler beim Schließen der Datei

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fbyWriteFile	

Es empfiehlt sich, Bausteine, die mit Dateizugriff arbeiten, in einer separaten Task in der asynchronen Programmebene PLC_PRG (Typ "Freilaufend") aufzurufen.

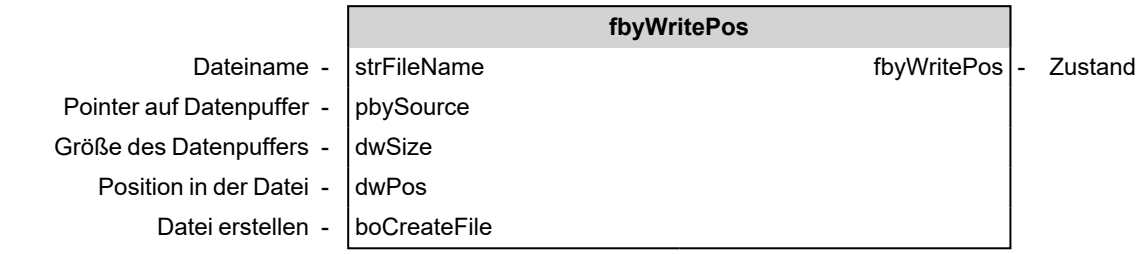
fbyWritePos (F)

Diese Funktion schreibt in eine Datei an die gewünschte Stelle und gibt ihren Zustand zurück.

Anders als die beiden Funktionsbausteine 'WRITE_FILE' und 'WRITE_FILE_1' aus der AmkFile.lib wird hier auf die SysLibFile.lib zurückgegriffen.

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
strFileName	STRING	Dateiname der Datei incl. -endung, die geschrieben werden soll
pbySource	POINTER	POINTER TO BYTE Pointer auf den Buffer, dessen Inhalt in die Datei geschrieben werden soll
dwSize	DWORD	Gibt die Größe des Datenspeichers an [Byte]
dwPos	DWORD	Position in der Datei, an der angefangen werden soll zu schreiben
boCreateFile	BOOL	TRUE: Datei wird neu erstellt FALSE: Es wird in die bestehende Datei geschrieben

Ausgangsvariable

Name	Typ	Beschreibung
fbyWritePos	BYTE	Zustand 0 – Undefinierter Zustand 1 – Datei wurde erfolgreich geschrieben 10 – Falsche Eingabe Parameter 11 – Fehler beim Öffnen der Datei 12 – Fehler bei der Änderung des Offset für den Dateizugriff 13 – Fehler beim Schreiben der Datei 14 – Fehler beim Schließen der Datei 16 – Fehler beim Setzen der Position: Aktueller Wert ≠ Vorgabe

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fbyWritePos	

Es empfiehlt sich, Bausteine, die mit Dateizugriff arbeiten, in einer separaten Task in der asynchronen Programmebene PLC_PRG (Typ "Freilaufend") aufzurufen.

4.2.1.2.5.3 Support

FB_LIST (FB)

Der Funkionsbaustein 'FB_LIST' wird im Programm 'PRG_ERR_HANDLER' aufgerufen. Der Baustein wird für eigens erstellte Tabellen dieser Bibliothek benötigt. Er realisiert Indexumschaltung und das entsprechende Umkopieren der Daten.
Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

'Support'-Funktionen und -Funktionsbausteine werden nicht direkt angewendet, sondern von anderen Funktionen und Funktionsbausteinen aufgerufen.

Anwender Interface

	FB_LIST			
Globale Alarmstruktur -	stGlobalAlarmData	stGlobalAlarmData	-	Globale Alarmstruktur
Datenablage -	arst_AlarmTab_ErrHist	arst_AlarmTab_ErrHist	-	Datenablage

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stGlobalAlarmData	STRUCT	ST_GLOBAL_ALARM_DATA Globale Alarm-Struktur
arst_AlarmTab_ErrHist	ARRAY	ST_ERR_ANZ Struktur, in die der Dateiinhalt geschrieben wird (g_arst_AlarmTab_ErrHist)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
FB_LIST	

fboClrStructur (F)

Die Funktion 'fboClrStructur' überschreibt den angegebenen Speicherbereich komplett mit 0, d. h. der Bereich wird gelöscht.
Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

'Support'-Funktionen und -Funktionsbausteine werden nicht direkt angewendet, sondern von anderen Funktionen und Funktionsbausteinen aufgerufen.

Anwender Interface

	fboClrStructur			
Startadresse -	pPointerToSt	fboClrStructur	-	Zustand
Bereichsgröße -	uiSize			

Eingangsvariablen

Name	Typ	Beschreibung
pPointerToSt	POINTER	POINTER TO BYTE Startadresse des zu löschenden Bereichs.
uiSize	UINT	Größe des zu löschenden Bereichs [Byte]

Ausgangsvariable

Name	Typ	Beschreibung
fboClrStructur	BOOL	Zustand

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fboClrStructur	

fstrGetDateStr (F)

Die Funktion 'fstrGetDateStr' konvertiert eine ganze Zahl in eine String-Variable

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

'Support'-Funktionen und -Funktionsbausteine werden nicht direkt angewendet, sondern von anderen Funktionen und Funktionsbausteinen aufgerufen.

Anwender Interface

	fstrGetDateStr		
Eingangswert -	siZahl	fstrGetDateStr	- Ausgangswert

Eingangsvariablen

Name	Typ	Beschreibung
siZahl	SINT	SINT(-99..99) Eingangsvariable ganze Zahl, z. B. 37

Ausgangsvariable

Name	Typ	Beschreibung
fstrGetDateStr	STRING (2)	Ausgangsvariable String, z. B. '37'

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fstrGetDateStr	

4.2.1.2.5.4 FB_READ_ERR_HISTORIE (FB)

Mit dem Funktionsbaustein 'FB_READ_ERR_HISTORIE' lassen sich die abgespeicherten Historie-Daten wieder auslesen. Diese werden auf der globalen Struktur 'g_arst_AlarmTab_ErrHist' abgelegt.

Durch die Visualisierung 'VI_HISTORIE' werden Daten entsprechend dargestellt.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	FB_READ_ERR_HISTORIE		
Funktionsausführung -	boExec	boErr	- Fehlersignal
Tag -	usiDay	iErrID	- Fehler Identnummer
Monat -	usiMonth	enErrName	- Name fehlerh Baustein
Jahr -	uiYear	boDone	- Quittierung "Ausgeführt"
Dateinummer -	usiFileNr	uiCount	- Anz. Ausgelesene Fehler
Dateiname -	strFileNameIn		
Datenablage -	arstErr	arstErr	- Datenablage
Globale Alarmstruktur -	stGlobalAlarmData	stGlobalAlarmData	- Globale Alarmstruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
usiDay	USINT	Tag, dient der Dateiauswahl (strFileNameIn='0' und PRG_ERR_HANDLER.enSaveMode ≠ enSaveErrRingBuffer)
usiMonth	USINT	Monat, dient der Dateiauswahl (strFileNameIn='0' und PRG_ERR_HANDLER.enSaveMode ≠ enSaveErrRingBuffer)
uiYear	UINT	Jahreszahl, dient der Dateiauswahl (strFileNameIn='0' und PRG_ERR_HANDLER.enSaveMode ≠ enSaveErrRingBuffer)
usiFileNr	USINT	Dateinummer, dient der Dateiauswahl (strFileNameIn='0' und PRG_ERR_HANDLER.enSaveMode ≠ enSaveErrRingBuffer und PRG_ERR_HANDLER.enSaveMode= enSaveErrSequential)
strFileNameIn	STRING (40)	String, der den Dateinamen enthält

- Entweder man übergibt dem Funktionsbaustein die entsprechenden Daten (Monat, Jahr, Tag und Dateinummer), um eine Historie Datei (*.err) auszulesen. In diesen Fall sollte man der Eingangsvariablen strFileNameIn:='0' übergeben.
  Oder man übergibt gleich den gesamten Dateinamen der Variable 'strFileNameIn' (z.B. '07042008_0.err'). Dabei werden die übergebenen Datumswerte ignoriert.
- Wenn PRG_ERR_HANDLER.enSaveMode = enSaveErrRingBuffer, dann darf an 'strFileNameIn' kein Wert bzw. sollte '0' übergeben werden.

Ausgangsvariablen

Name	Typ	Beschreibung
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. FIND_FILE_1: AmkFile.lib READ_FILE_1: AmkFile.lib
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
uiCount	UINT	Anzahl der ausgelesenen Fehler

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
arstErr	ARRAY	ST_ERR_ANZ Struktur, in die der Dateiinhalt geschrieben wird (g_arst_AlarmTab_ErrHist)
stGlobalAlarmData	STRUCT	ST_GLOBAL_ALARM_DATA Globale Alarm-Struktur

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
FB_READ_ERR_HISTORIE	

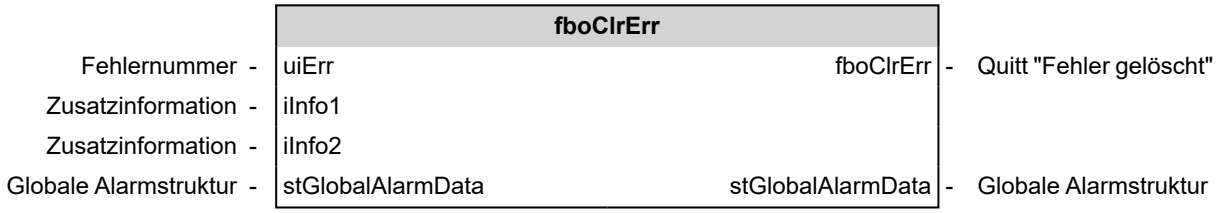
4.2.1.2.5.5 fboClrErr (F)

Der Fehler, der der Funktion 'fboClrErr' mit seinen Zusatzinfos übergeben wird, wird aus der aktuellen Alarmtabelle gelöscht, jedoch nicht aus der Historie.

Dazu wird das ganze Feld nach dem Fehler durchsucht. Gibt es den Fehler nicht, wird auch nichts gelöscht, aber trotzdem TRUE zurückgemeldet. Wird der zu löschende Fehler gefunden, dann wird er aus der Tabelle gelöscht und die in der Tabelle folgenden Fehler rücken auf. Danach wird die Ansicht aktualisiert.

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
uiErr	UINT	Fehlernummer
iInfo1	INT	Fehler Zusatzinformation 1
iInfo1	INT	Fehler Zusatzinformation 2

Ausgangsvariable

Name	Typ	Beschreibung
fboClrErr	BOOL	Quittierung: Fehler gelöscht TRUE: Fehler aus Tabelle gelöscht FALSE: Buffer gesperrt

Ein- und Ausgangsvariable

Name	Typ	Beschreibung
stGlobalAlarmData	STRUCT	ST_GLOBAL_ALARM_DATA Globale Alarm-Struktur

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fboClrErr	

4.2.1.2.5.6 fboSetErr (F)

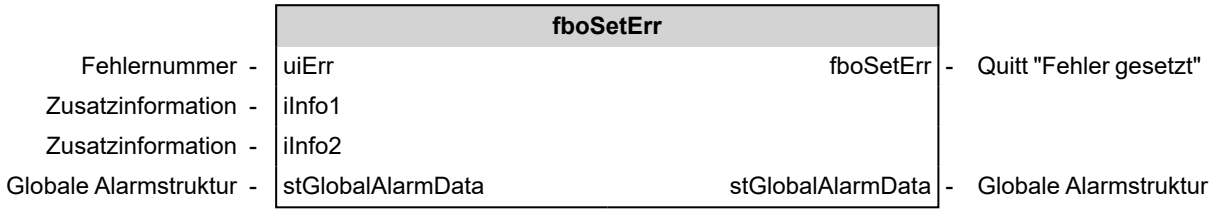
Der Fehler, der Funktion 'fboSetErr' übergeben wird, wird gesetzt und in der globalen Struktur g_stGlobalAlarmData an die nächste freie Stelle geschrieben.

Konnte der Fehler nicht gesetzt werden, liefert die Funktion FALSE zurück. In diesem Fall sollte die Funktion so lange wiederholt aufgerufen werden, bis sie TRUE zurückliefert.

TRUE liefert die Funktion zurück, wenn der Fehler gesetzt und in den FIFO geschrieben wurde. Enthält die Alarmtabelle den aktuellen Fehler (mit Zusatzinfo 1+2) schon, wird der Fehler nicht erneut gesetzt, aber die Funktion liefert den Wert TRUE zurück.

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
uiErr	UINT	Fehlernummer
iInfo1	INT	Fehler Zusatzinformation 1
iInfo1	INT	Fehler Zusatzinformation 2

Ausgangsvariable

Name	Typ	Beschreibung
fboSetErr	BOOL	Quittierung: Fehler gesetzt TRUE: Fehler in Tabelle geschrieben FALSE: Buffer gesperrt oder voll

Ein- und Ausgangsvariable

Name	Typ	Beschreibung
stGlobalAlarmData	STRUCT	ST_GLOBAL_ALARM_DATA Globale Alarm-Struktur

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fboSetErr	

4.2.1.2.5.7 fusiSetErrTabRow (F)

Die Funktion 'fusiSetErrTabRow' legt fest, wie viele Zeilen die Alarmtabellen in der Visualisierung enthält.

Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

Eingangsvariablen

Name	Typ	Beschreibung
usiNoRows	USINT	USINT(3..15)Anzahl der Zeilen in der 'VI_ALARMTAB'

Ausgangsvariable

Name	Typ	Beschreibung
fusiSetErrTabRow	USINT	Anzahl der Zeilen in der 'VI_ALARMTAB'

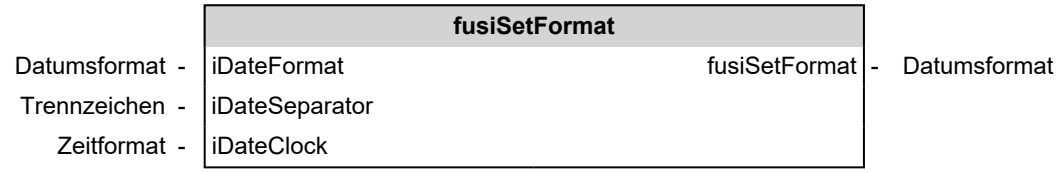
Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fusiSetErrTabRow	

4.2.1.2.5.8 fusiSetFormat (F)

Die Funktion 'fusiSetFormat' stellt das Datum- und Uhrzeitformat ein.
Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
iDateFormat	INT	INT(0..2) 0 enDD_MM_YYYY 1 enYYYY_MM_DD 2 enMM_DD_YYYY
iDateSeparator	INT	INT(3..5) 3 enSeperatot_Point [.] 4 enSeperatot_Hyphen [-] 5 enSeperatot_Slash [/])
iDateClock	INT	INT(6..7) 6 enClock_24_Hours 7 enClock_12_Hours

Ausgangsvariable

Name	Typ	Beschreibung
fusiSetFormat	USINT	Datumsformat

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fusiSetFormat	

4.2.1.2.5.9 fusiSetHistTabRow (F)

Die Funktion 'fusiSetHistTabRow' legt fest, wie viele Zeilen die Historientabelle in der Visualisierung enthält.
Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

Eingangsvariablen

Name	Typ	Beschreibung
usiNoRows	USINT	USINT(3..15)Anzahl der Zeilen in der 'VI_HISTORIE'

Ausgangsvariable

Name	Typ	Beschreibung
fusiSetHistTabRow	USINT	Anzahl der Zeilen in der 'VI_HISTORIE'

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fusiSetHistTabRow	

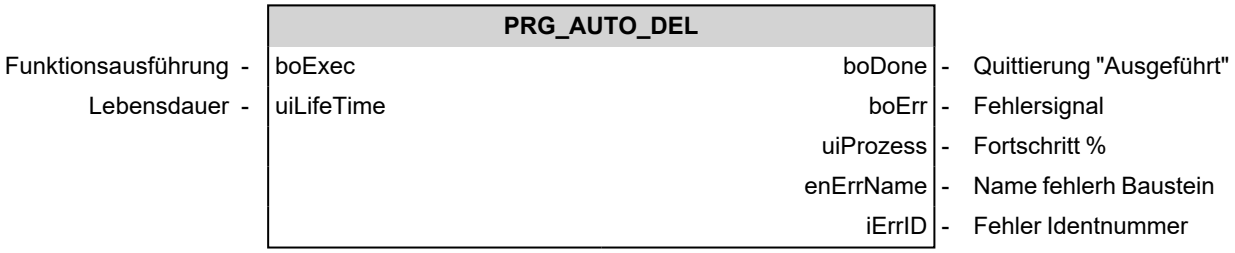
4.2.1.2.5.10 PRG_AUTO_DEL (PRG)

Das Programm 'PRG_AUTO_DEL' wird im 'FB_ERR_HANDLER' aufgerufen. Es dient dem automatischen Löschen der Historiedateien, die älter als die eingestellte Zeit sind. Dazu wird das aktuelle Datum mit dem Datum der vorhandenen Historiedateien (außer die Ringbufferdatei) verglichen.

Der Anwender braucht diesen Funktionsbaustein nicht explizit aufzurufen oder zu steuern, da dies im 'FB_ERR_HANDLER' geschieht. Die Lebensdauer der Historie Dateien lässt sich über die Eingangsvariable 'uiLifeTimeHist' des 'PRG_ERR_HANDLER' einstellen.

Das Programm wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
uiLifeTime	UINT	Lebensdauer der Historiedateien [Tage]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
uiProzess	UINT	Fortschritt (relativ zur Anzahl der gefunden *.err Dateien) [%]

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein** / **Bibliothek** FILE_FIND / AmkAfl.lib FILE_DELETE / AmkAfl.lib
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0 — Kein Fehler iErrID ≠ 0 — boErr = TRUE — Fehler iErrID ≠ 0 — boErr = FALSE — Warnung

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
PRG_AUTO_DEL	

4.2.1.2.5.11 PRG_ERR_HANDLER (PRG)

Das Programm verarbeitet alles Notwendige für die Alarmtabelle und deren Historie.

Dieses Programm überwacht den globalen Puffer, der durch die Funktion 'fboSetErr' gefüllt wird. Enthält die erste Stelle des FIFO ein Fehlerelement, so wird dieses an die aktuelle Historiedatei angehängt. Des Weiteren ruft dieses Programm das Programm 'PRG_AUTO_DEL' auf.

Das Programm wird in der asynchronen Programmebene PLC_PRG aufgerufen.
Das Programm 'PRG_ERR_HANDLER' muss in einer separaten Task aufgerufen werden.

Anwender Interface

	PRG_ERR_HANDLER			
Alarmtabelle löschen -	bo_AlarmTab_ClrAns	boErr	-	Fehlersignal
Scrollbar in Visu -	boScrollbar	iErrID	-	Fehler Identnummer
"?" Dokubutton -	boDoku	enErrName	-	Name fehlerh Baustein
Lebensdauer History -	uiLifeTimeHist	boDone	-	Quittierung "Ausgeführt"
History-Speichermodus -	enSaveMode	boClrAnsAckn	-	Quitt "Alarmtab. gelöscht"

Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
bo_AlarmTab_ClrAns	BOOL	Mit einer positiven Flanke wird die Alarmtabelle gelöscht
boScrollbar	BOOL	Mit dieser Variablen kann der Scrollbalken ein- bzw. ausgeschaltet werden (auch während der Laufzeit)
boDoku	BOOL	Mit Hilfe dieses Eingangs kann der Dokubutton "?" ein- bzw. ausgeschaltet werden
uiLifeTimeHist	UINT	Lebensdauer der Historiedateien (wenn enSaveMode ≠ enSaveErrRingBuffer)

Name	Typ	Beschreibung
enSaveMode	ENUM	EN_SAVE_MODE Gibt vor wie die Fehler abgespeichert werden sollen 0: enSaveErrSequential • Fehler werden der Reihe nach in Datei gespeichert. Ist diese Datei voll, wird eine neue Datei erstellt . • Für jeden Tag wird min. eine Datei erstellt (wenn ein Fehler auftritt) 1: enSaveErrDayRingBuffer • Fehler werden der Reihe nach in Datei gespeichert. Ist diese Datei voll, wird sie von vorne überschrieben. -> Ringpuffer • Für jeden Tag wird eine Datei erstellt (wenn ein Fehler auftritt) 2: enSaveErrRingBuffer • Fehler werden der Reihe nach in Datei gespeichert. Ist diese Datei voll, wird sie von vorne überschrieben -> Ringpuffer • Es gibt nur diese eine Datei die nicht gelöscht wird.

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: FIND_FILE_1 – Bibliothek: AmkFile.lib Baustein: WRITE_FILE_1 – Bibliothek: AmkFile.lib
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boClrAnsAckn	BOOL	Rückmeldung, dass die Alarmtabelle gelöscht wurde

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
PRG_ERR_HANDLER	

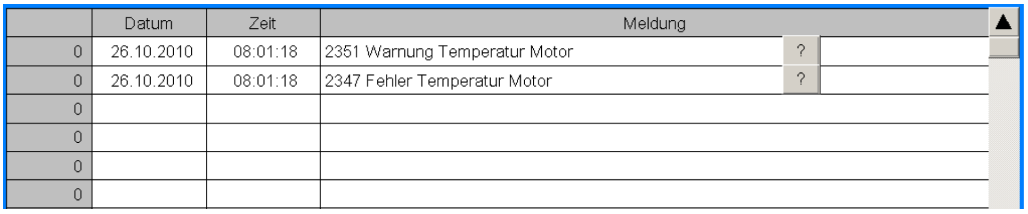
4.2.1.2.5.12 Visualisierung

VI_ALARMTAB (VI)

Diese Visualisierung kann in jede beliebige Maske integriert werden. Sie zeigt alle aktuellen Fehler an.

- Die Anzahl ihrer Zeilen (3..15) kann über 'fusiSetErrTabRow' angepasst werden.
- Mit Hilfe der Funktion 'fusiSetFormat' können das Datums- und Zeitformat angepasst werden.

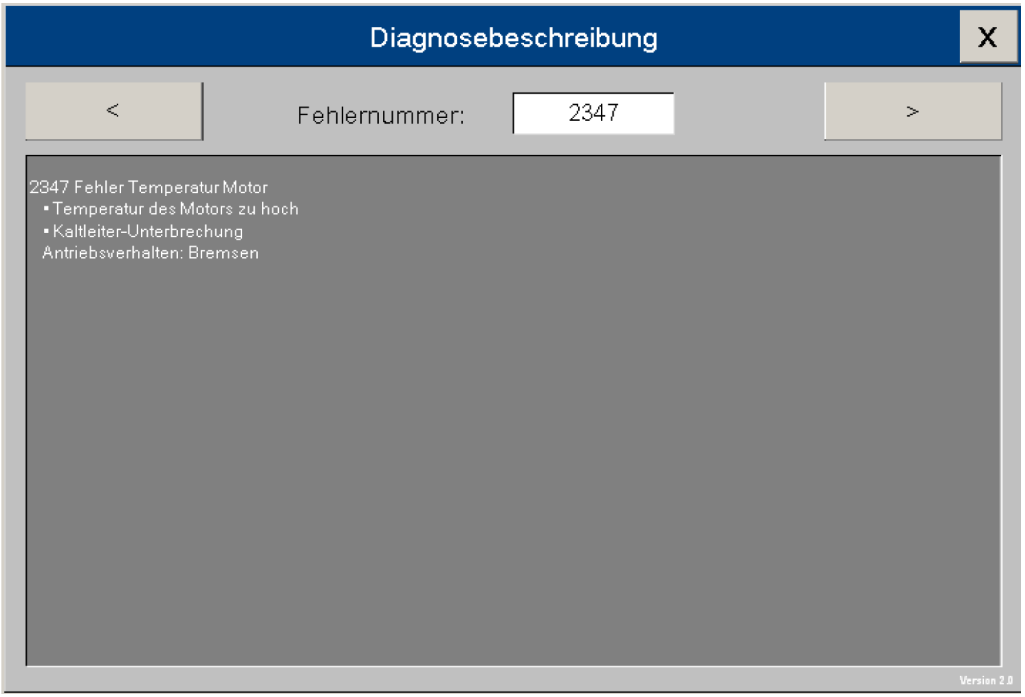
- Durch die globale Variable 'g_stVisuAlarmTab' von der Struktur 'VisualObjectType' kann die Größe der Tabelle angepasst werden.
- Mittels der Variable 'g_stGlobalAlarmData.iAlarmFontHeight' kann die Schriftgröße der Tabelle verändert werden.



- Mit Hilfe des ? -Buttons gelangt der Anwender zur Maske VI_DOKU_ERR.
  Über PRG_ERR_HANDLER.boDoku wird eingestellt, ob der Button bei einem Fehler erscheinen soll oder nicht
- Der Scrollbalken lässt sich über PRG_ERR_HANDLER.boScrollbar ein- / ausschalten
- Mittels dynamischer Sprachumschaltung können Fehlertexte in der Tabelle dargestellt werden
- Der Prefix lautet "Error", über die Fehlernummer wird die ID festgelegt
  Beispiel:
  <text prefix="Error" id="2347">
  <deu><![CDATA[2347 Fehler Temperatur Motor]] ></deu><text>
- Mit der Konstanten 'gc_uiElementeAlarmTab' kann die Tabellenlänge eingestellt werden
  Der Wert dieser Konstanten sollte nur zwischen 15 und 4900 überschattet werden

VI_DOKU_ERR (VI)

Diese Visualisierung ist als eine eigene Maske zu verstehen, die angezeigt wird, wenn der Benutzer den ?-Button in der Alarm- oder Historietabelle betätigt.



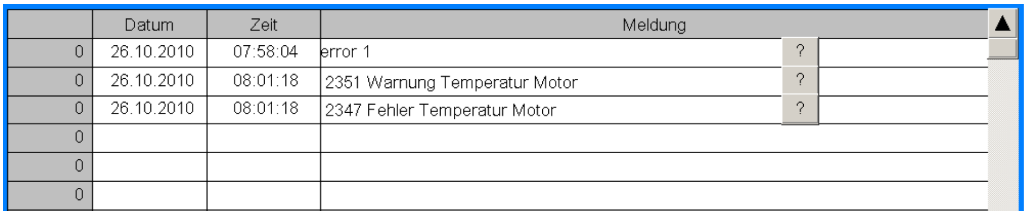
- Mit Hilfe des X -Buttons gelangt der Anwender zur vorherigen Maske zurück.
- Mittels dynamischer Sprachumschaltung können Fehlertexte in der Tabelle dargestellt werden

- Der Prefix lautet AntrErr, über die Fehlernummer wird die ID festgelegt
  Beispiel:
  <text prefix="AntrErr" id="2347">
  <deu><![CDATA[2347 Fehler Temperatur Motor
  - Temperatur des Motors zu hoch
  - Kaltleiter-Unterbrechung
  - Antriebsverhalten: Bremsen]] ></deu></text>

VI_HISTORIE (VI)

Diese Visualisierung kann in jede beliebige Maske integriert werden. Sie zeigt die mittels 'FB_READ_ERR_HISTORIE' ausgelesenen Fehler der Historie an.

- Ihre Anzahl der Zeilen (3..15) kann über 'fusiSetHistTabRow' angepasst werden.
- Mit Hilfe der Funktion 'fusiSetFormat' können das Datums- und Zeitformat angepasst werden.
- Durch die globale Variable 'g_stVisuHistoryTab' von der Struktur 'VisualObjectType' kann die Größe der Tabelle angepasst werden.
- Mittels der Variable 'g_stGlobalAlarmData.iHistoryFontHeight' kann die Schriftgröße der Tabelle verändert werden.



	Datum	Zeit	Meldung
0	26.10.2010	07:58:04	error 1 ?
0	26.10.2010	08:01:18	2351 Warnung Temperatur Motor ?
0	26.10.2010	08:01:18	2347 Fehler Temperatur Motor ?
0			
0			
0			

- Mit Hilfe des ?-Buttons gelangt der Anwender zur Maske 'VI_DOKU_ERR'
  Über 'PRG_ERR_HANDLER.boDoku' wird eingestellt, ob der Button bei einem Fehler erscheinen soll oder nicht
- Der Scrollbalken lässt sich über 'PRG_ERR_HANDLER.boScrollbar' ein- / ausschalten
- Mittels dynamischer Sprachumschaltung können Fehlertexte in der Tabelle dargestellt werden
- Der Prefix lautet "Error", über die Fehlernummer wird die ID festgelegt
  Beispiel:
  <text prefix="Error" id="2347">
  <deu><![CDATA[2347 Fehler Temperatur Motor]] ></deu><text>

4.2.1.2.6 06_Commands

4.2.1.2.6.1 CMD_HOME (FB)

Der Funktionsbaustein 'CMD_HOME' kommandiert eine Referenzfahrt eines Amk-Antriebes aus.

Dabei sind die Parameter aus dem Parametersatz der Achse gültig. Die Referenzfahrparameter sind in der Parameterbeschreibung, Kapitel Positionierparameter, erläutert.

Die Referenzfahrt kann durch Entzug des Ausführungsbits abgebrochen werden.

Nach erfolgreicher Referenzpunktfahrt werden das Lageistwertsystem des Antriebs und das Lagesollwertsystem der Steuerung gleichgesetzt.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	CMD_HOME			
Funktionsausführung -	boExec	boDone	-	Quittierung "Ausgeführt"
Positionsistwert -	diActPosition	boBusy	-	Ausführung läuft
		boErr	-	Fehlersignal
		iErrID	-	Fehler Identnummer
		enErrName	-	Name fehlerh Baustein
Lagesollwert -	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
diActPosition	DINT	Aktueller Lageistwert [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boBusy	BOOL	Ausführungsmeldung: Das Bit bleibt gesetzt, solange der Baustein bearbeitet wird.
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: DO_CMD_ONCE — Bibliothek: AmkDevAccess.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
CMD_HOME	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.7 07_NC Motion

4.2.1.2.7.1 NC_MOTION

- Geradeninterpolation mit bis zu 16 Achsen
- Kreisinterpolation cw/ccw mit den Achsen 1/2 ,1/3, 2/3, programmierbar in 3 Ebenen
- Möglichkeit der Kreisinterpolation (Achsen 1-3) und gleichzeitig Geradeninterpolation

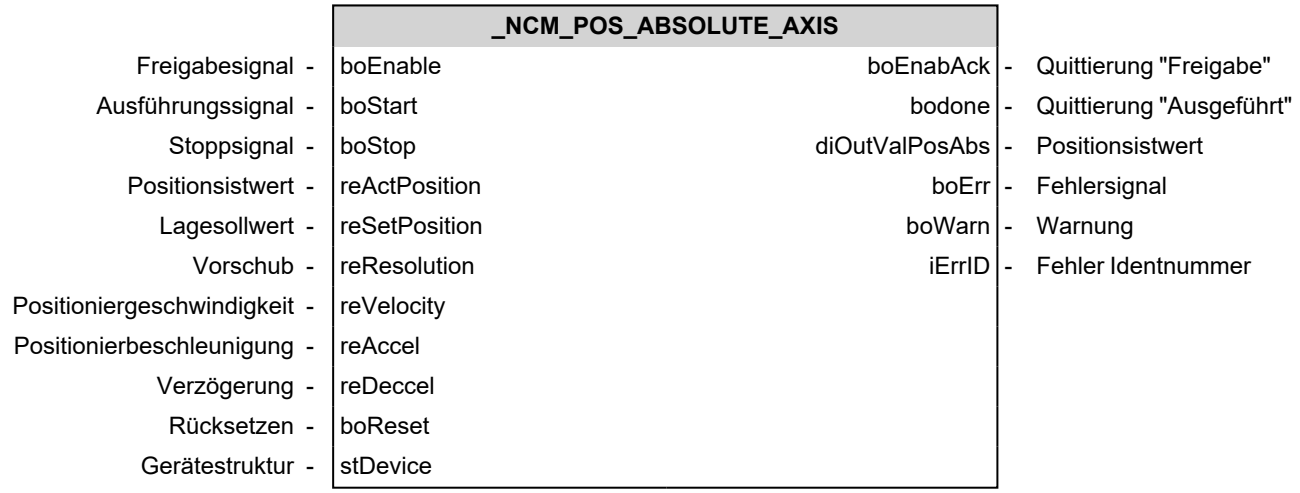
- Programmierbare Verzögerungszeit zwischen zwei NC-Sätzen
- Programmierbarer Halt (nach NC-Satz)
- Unverzüglicher Stop (während des NC-Satzes)
- Modus Einzelsatz
- 32 M-Funktionen programmierbar als synchronisiert und unsynchronisiert
- Bis zu 3 M-Funktionen in einem NC-Satz programmierbar
- Programmierbare Bahngeschwindigkeit
- Programmierbare Beschleunigungs-, Verzögerungsrampen
- Korrektur der Bahngeschwindigkeit aufgrund der maximalen Drehzahl des Motors
- Override 0-200%
- Tangentieller Übergang zwischen NC-Sätzen
  (Wahlweise automatische Berechnung oder manuelle Eingabe des tangentiellem Übergangs)
- Bis zu 16 programmierbare Prozessdaten pro NC-Satz
- Genauhalt mittels Positionsfenster möglich
- Anzeige aktuelle NC-Satznummer
- Jede Achse kann als "Mitschleppachse" konfiguriert werden
- Jede Achse kann als Rundachse mit zugehöriger Moduloposition konfiguriert werden
- Plausibilitätsüberprüfung der Eingangsdaten
- Ausgabe von Warn -, und Fehlermeldungen

4.2.1.2.7.2 FAST Modul

_NCM_POS_ABSOLUTE_AXIS (FB)

Mit dem Funktionsbaustein '_NCM_POS_ABSOLUTE_AXIS' wird eine einzelne Achse absolut positioniert.
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boStart	BOOL	Mit einer positiven Flanke startet die Ausführung des Bausteins.
boStop	BOOL	Mit einer positiven Flanke wird die Ausführung des Bausteins abgebrochen oder beendet.
reActPosition	REAL	Aktueller Lageistwert [mm]

Name	Typ	Beschreibung
reSetPosition	REAL	Vorgabe des Lagesollwertes [mm]
reResolution	REAL	Vorschubkonstante: Umsetzung der rotatorischen in eine lineare Bewegung [Inkremente/mm]
reVelocity	REAL	Geschwindigkeitssollwert [Inkremente/s]
reAccel / reDeccel	REAL	Beschleunigung / Verzögerung [mm/s^2]
boReset	BOOL	Reset; Ausgangswert = 0, anschließend wird der Eingang 'boReset' vom Baustein selbständig zurückgesetzt
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
bodone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
diOutValPosAbs	DINT	Aktueller Lageistwert [mm]
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
boWarn	BOOL	Der Funktionsbaustein befindet sich im Warn-Zustand
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben

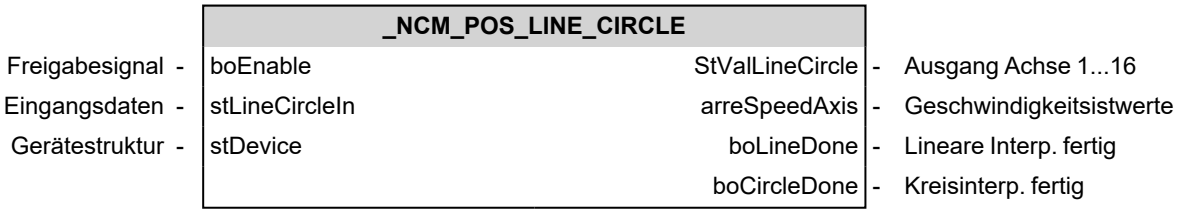
Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	_NCM_POS_ABSOLUTE_AXIS

_NCM_POS_LINE_CIRCLE (FB)

Der Funktionsbaustein '_NCM_POS_LINE_CIRCLE' positioniert die Achsen relativ mit Ruckbegrenzung.
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
stLineCircleIn	STRUCT	_NCM_ST_LINE_CIRCLE_IN Eingangsdaten

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Ausgangsvariablen

Name	Typ	Beschreibung
StValLineCircle	STRUCT	_NCM_ST_VALOUT_LINE_CIRCLE Ausgangsdaten der Achsen 1 ... 16
arreSpeedAxis	ARRAY	ARRAY[1..16]OF REAL Geschwindigkeitistwerte der Achsen
boLineDone	BOOL	Lineare Interpolation beendet
boCircleDone	BOOL	Kreisinterpolation beendet

Verwendungshinweis im CoDeSys-Programm

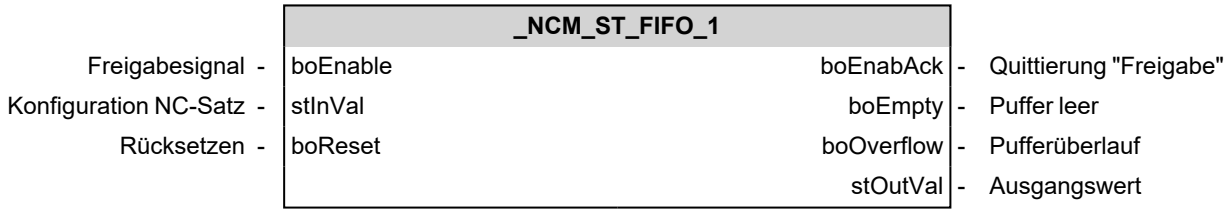
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	_NCM_POS_LINE_CIRCLE

4.2.1.2.7.3 FIFO

_NCM_ST_FIFO_1 (FB)

Der Funktionsbaustein '_NCM_ST_FIFO_1' verwaltet den FIFO-Speicher in der asynchronen Programmebene PLC_PRG.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
stInVal	STRUCT	_NCM_ST_GEO_IN Datenstruktur mit den Daten des NC-Satzes, der gespeichert werden soll.
boReset	BOOL	FIFO Speicher zurücksetzen

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boEmpty	BOOL	Der FIFO-Puffer ist leer
boOverflow	BOOL	Pufferüberlauf: Ein neuer Wert soll in den Puffer gespeichert werden, es ist aber kein freier Speicherplatz vorhanden.

Name	Typ	Beschreibung
stOutVal	STRUCT	_NCM_ST_GEO_IN Datenstruktur mit den Daten des NC-Satzes, der gespeichert werden soll.

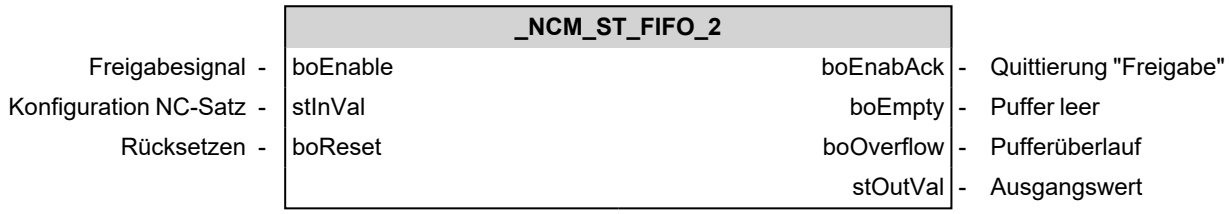
Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
_NCM_ST_FIFO_1	

_NCM_ST_FIFO_2 (FB)

Der Funktionsbaustein '_NCM_ST_FIFO_2' verwaltet den FIFO-Speicher in der synchronen Programmebene FPLC_PRG.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
stInVal	STRUCT	_NCM_ST_GEO Datenstruktur mit den Daten des aktuellen NC-Satzes, der bearbeitet werden soll.
boReset	BOOL	FIFO Speicher zurücksetzen

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boEmpty	BOOL	Der FIFO-Puffer ist leer
boOverflow	BOOL	Pufferüberlauf: Ein neuer Wert soll in den Puffer gespeichert werden, es ist aber kein freier Speicherplatz vorhanden.
stOutVal	STRUCT	_NCM_ST_GEO Datenstruktur mit den Daten des aktuellen NC-Satzes, der bearbeitet werden soll.

Verwendungshinweis im CoDeSys-Programm

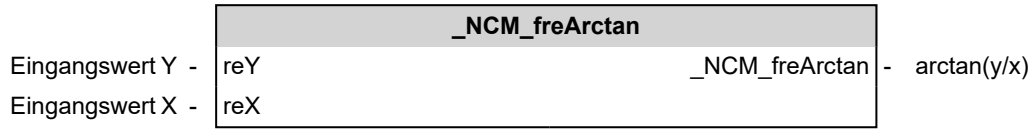
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	_NCM_ST_FIFO_2

4.2.1.2.7.4 Funktionen

_NCM_freArctan (F)

Die Funktion '_NCM_freArctan' berechnet den arctan(y/x).
Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
reY	REAL	Eingangswert Y: Gegenkathete
reX	REAL	Eingangswert X: Ankathete

Ausgangsvariable

Name	Typ	Beschreibung
_NCM_freArctan	REAL	arctan (y/x) [rad]

Verwendungshinweis im CoDeSys-Programm

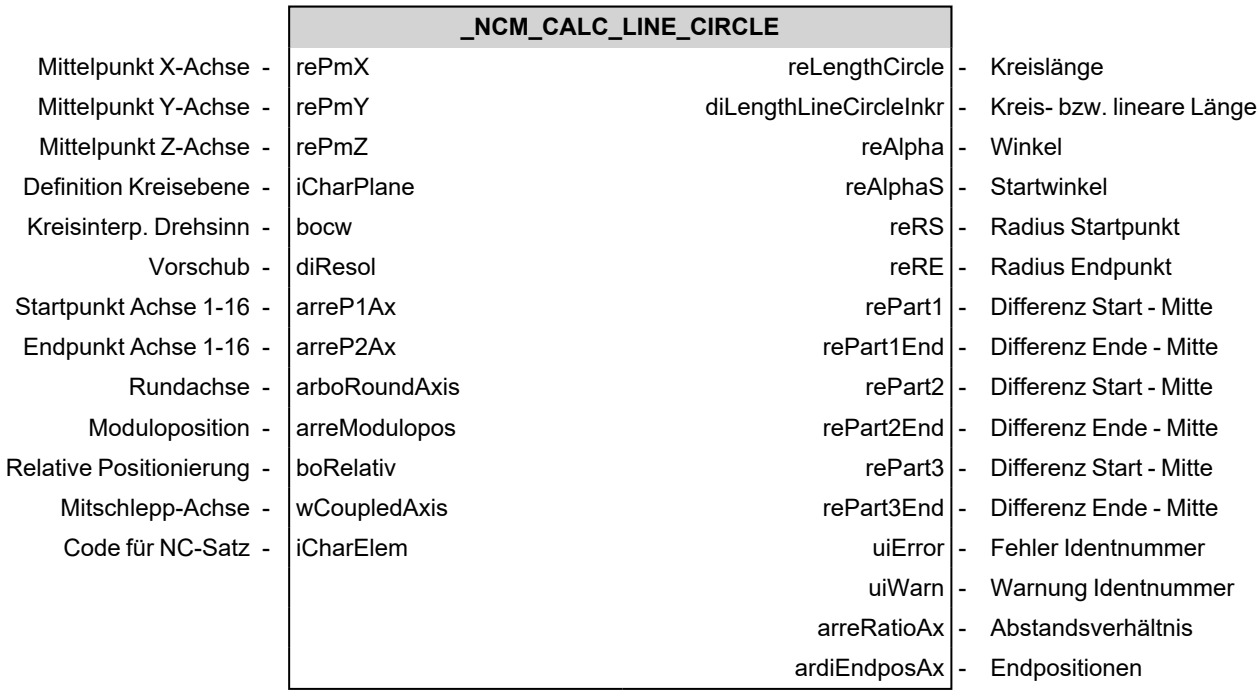
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
_NCM_freArctan	

4.2.1.2.7.5 Module

_NCM_CALC_LINE_CIRCLE (FB)

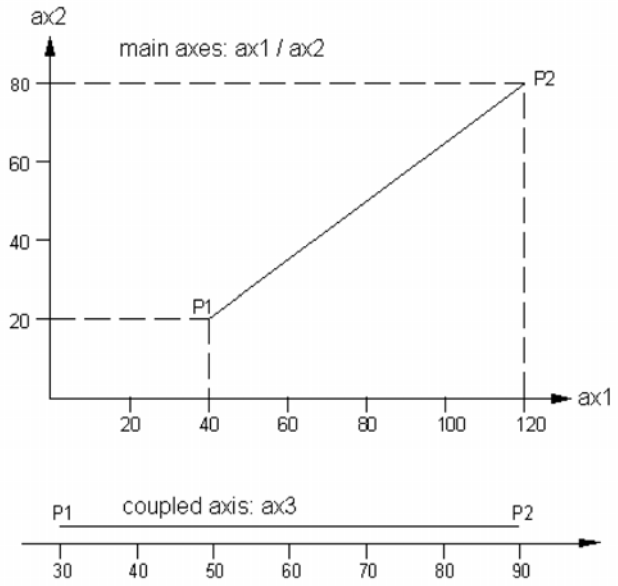
Der Funktionsbaustein '_NCM_CALC_LINE_CIRCLE' berechnet die lineare bzw. Kreisinterpolation für einen NC-Datensatz.
Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>rePmX, rePmY, rePmZ</td><td>REAL</td><td>Position des Kreismittelpunktes, Achse X / Y / Z [mm]</td></tr>
<tr><td>iCharPlane</td><td>INT</td><td>Code zum Definieren der Kreisebene
<table><tr><th>iCharPlane</th><th>Kreisebene</th></tr><tr><td>17</td><td>X/Y (Achse 1 / Achse 2)</td></tr><tr><td>18</td><td>X/Z (Achse 1 / Achse 3)</td></tr><tr><td>19</td><td>Y/Z (Achse 2 / Achse 3)</td></tr></table></td></tr>
<tr><td>bocw</td><td>BOOL</td><td>Kreisinterpolation
TRUE: clockwise (cw)
FALSE: counter clockwise (ccw)</td></tr>
<tr><td>diResol</td><td>DINT</td><td>Vorschubkonstante: Umsetzung der rotatorischen in eine lineare Bewegung [Inkremente/mm]</td></tr>
<tr><td>arreP1Ax, arreP2Ax</td><td>ARRAY</td><td>ARRAY[1..16]OF REAL
Start- bzw. Endpunkte der Achsen 1 ... 16</td></tr>
<tr><td>arboRoundAxis</td><td>ARRAY</td><td>ARRAY[1..16]OF BOOL
Bestimmung der Achse als Rundachse
TRUE: Rundachse</td></tr>
<tr><td>arreModulopos</td><td>ARRAY</td><td>ARRAY[1..16]OF REAL
Mdouloposition</td></tr>
<tr><td>boRelativ</td><td>BOOL</td><td>Relative Positionierung
Soll ein NC-Fahrsatz nicht absolut, sondern relativ ausgeführt werden, muss 'boRelativ' auf true gesetzt werden.
Beispiel:
Relative Geradeninterpolation der Achse1 um 100mm.</td></tr>
</table>

Name	Typ	Beschreibung
wCoupledAxis	WORD	Mit der Bitleiste 'wCoupledAxis' kann jede Achse als MitschleppAchse konfiguriert werden. Die Koordinatenangaben einer Mitschleppachse gehen nicht in die Berechnung der Bahngeschwindigkeit aller anderen Achsen mit ein. Beispiel: Die Bahngeschwindigkeit wird für den Weg von P1 nach P2 für Achse1 und Achse2 bestimmt. Die Bahngeschwindigkeit für Achse3 wird so berechnet, dass sie abhängig vom ihrem zurückzulegenden Weg gleichzeitig wie Achse1 und Achse2 am Zielpunkt ankommt. Achse3 als Mitschleppachse: wCoupledAxis := 16#0004 
iCharElem	INT	Code für den NC-Satz 1 – Geradeninterpolation 2 – Kreisinterpolation clockwise (cw) 3 – Kreisinterpolation counter clockwise (ccw) 4 – Verzögerungszeit 9 – Halt 30 – Programm Ende

Ausgangsvariablen

Name	Typ	Beschreibung
reLengthCircle	REAL	Kreislänge [mm]
diLengthLineCircleInkr	DINT	Kreis- bzw. lineare Länge [Inkremente]
reAlpha	REAL	Winkel des Kreisbogens [rad]
reAlphaS	REAL	Startwinkel [rad]
reRS, reRE	REAL	Länge des Radius am Start- / Endpunkt [mm]
rePart1, rePart2, rePart3	REAL	Differenz Startpunkt - Mittelpunkt X / Y / Z [mm]
rePart1End, rePart2End, rePart3End	REAL	Differenz Endpunkt - Mittelpunkt X / Y / Z [mm]

Name	Typ	Beschreibung
uiError	UINT	Fehler Identnummer: Diagnosenummer wird ausgegeben 0 Kein Fehler 5 Kreisradius ist Null 6 Auswahl der gekoppelten Achsen ist fehlerhaft
uiWarn	UINT	Warnung Identnummer: Diagnosenummer wird ausgegeben 0 Keine Warnung 1 Länge der Geraden ist Null 6 Radius am Startpunkt des Kreises ist unterschiedlich zum Radius am Endpunkt des Kreises.
arreRatioAx	ARRAY	ARRAY[1..16]OF REAL Feld mit Verhältnissen der Distanzen der Achsen 1 ... 16 zur Gesamtdistanz
ardiEndposAx	ARRAY	ARRAY[1..16]OF DINT Feld mit Endpositionen der Achsen 1 ... 16 [Inkremente]

Verwendungshinweis im CoDeSys-Programm

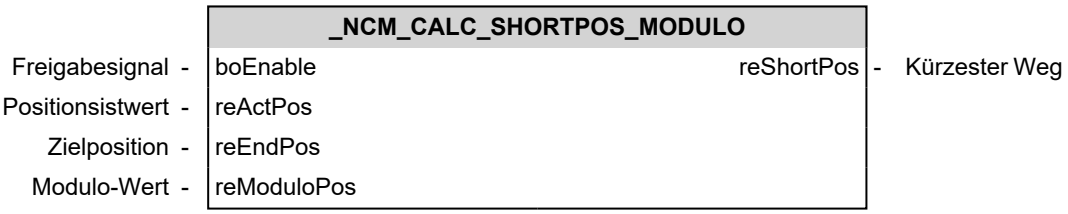
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
_NCM_CALC_LINE_CIRCLE	

_NCM_CALC_SHORTPOS_MODULO (FB)

Der Funktionsbaustein '_NCM_CALC_SHORTPOS_MODULO' berechnet den kürzesten Weg zur Modulo Positionierung einer rotierenden Achse.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
reActPos	REAL	Aktuelle Position der Modulo-Achse [°]
reEndPos	REAL	Endposition der Modulo-Achse [°]
reModuloPos	REAL	Modulo-Wert der Achse [°]

Ausgangsvariablen

Name	Typ	Beschreibung
reShortPos	REAL	Kürzester Weg zwischen 'reActPos' und 'reEndPos' [°]

Verwendungshinweis im CoDeSys-Programm

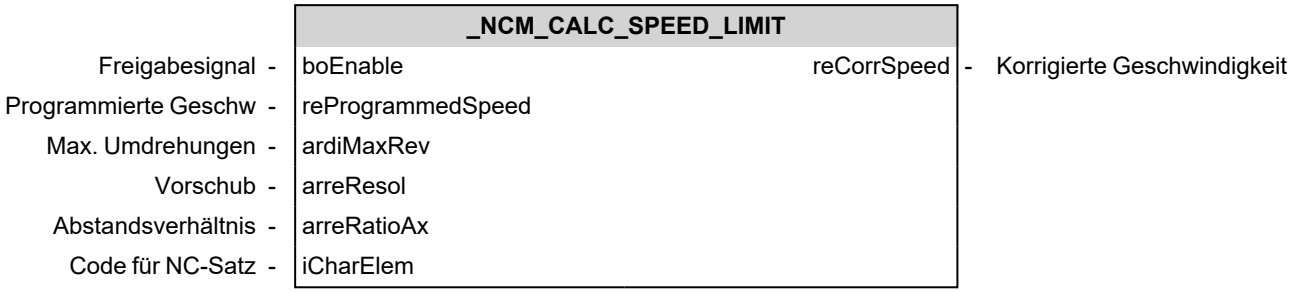
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
_NCM_CALC_SHORTPOS_MODULO	

_NCM_CALC_SPEED_LIMIT (FB)

Der Funktionsbaustein '_NCM_CALC_SPEED_LIMIT' berechnet die maximale Geschwindigkeit der Achse, basierend auf den maximal zugelassenen Umdrehungen.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
reProgrammedSpeed	REAL	Programmierte Geschwindigkeit [mm/min]
ardiMaxRev	ARRAY	ARRAY[1..16] OF DINT Maximale Umdrehungen der Achse [Umdrehungen]
arreResol	ARRAY	ARRAY[1..16] OF REAL Vorschubkonstante: Umsetzung der rotatorischen in eine lineare Bewegung [mm/Umdrehung]
arreRatioAx	ARRAY	ARRAY[1..16]OF REAL Feld mit Verhältnissen der Distanzen der Achsen 1 ... 16 zur Gesamtdistanz
iCharElem	INT	Code für den NC-Satz 1 – Geradeninterpolation 2 – Kreisinterpolation clockwise (cw) 3 – Kreisinterpolation counter clockwise (ccw) 4 – Verzögerungszeit 9 – Halt 30 – Programm Ende

Ausgangsvariablen

Name	Typ	Beschreibung
reCorrSpeed	REAL	Korrigierte Geschwindigkeit [mm/min]

Verwendungshinweis im CoDeSys-Programm

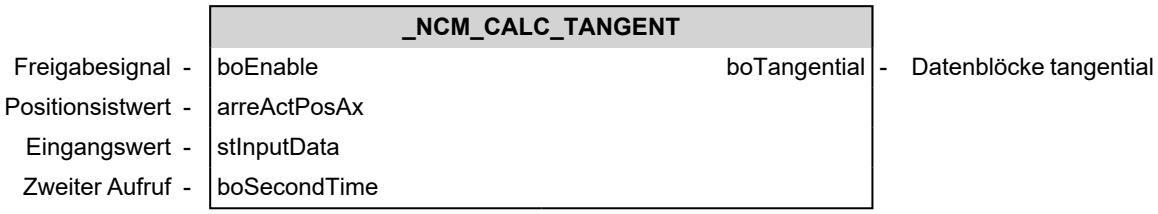
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
_NCM_CALC_SPEED_LIMIT	

_NCM_CALC_TANGENT (FB)

Der Funktionsbaustein '_NCM_CALC_TANGENT' ermittelt, ob der Übergang zwischen zwei NC-Datensätzen tangential ist, so dass die Bahngeschwindigkeit beim Übergang von NC-Fahrsatz (n) zu NC-Fahrsatz (n+1) nicht auf Null abgebremst werden muss.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
arreActPosAx	ARRAY	ARRAY[1..16]OF REAL Feld der aktuellen Positionen
stInputData	STRUCT	_NCM_ST_GEO_IN Datenstruktur mit den Daten des NC-Satzes, der bearbeitet werden soll.
boSecondTime	BOOL	Der Funktionsbaustein startet erst, wenn er zum zweiten Mal aufgerufen wird

Ausgangsvariablen

Name	Typ	Beschreibung
boTangential	BOOL	Datenblöcke sind tangential, Winkel = 0

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
_NCM_CALC_TANGENT	

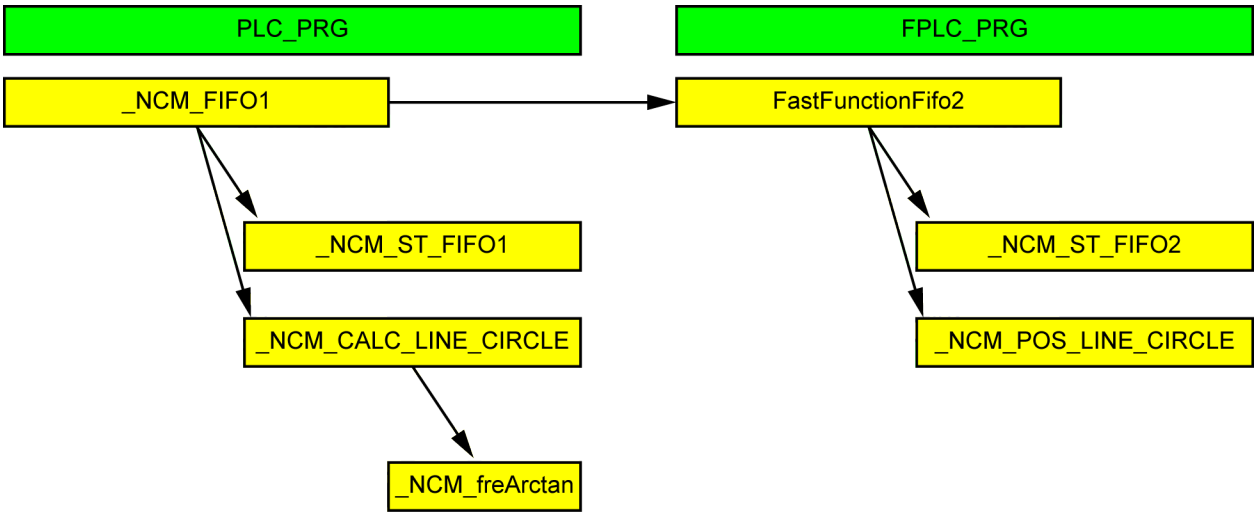
_NCM_FIFO1 (FB)

Der Funktionsbaustein '_NCM_FIFO1' ist der übergeordnete Baustein der NC-Motion Funktionen.

- Die notwendigen Parameter und Randbedingungen werden eingegeben und alle Ergebnisse und Informationen rückgemeldet.
- Die Eingangsdaten werden in den FIFO-Speicher geschrieben.
- Lineare bzw. Kreisinterpolation werden vorberechnet.
- Über die synchrone Aktion 'FastFunctionFifo2' werden die vorberechneten Daten in den

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

_NCM_FIFO1 ruft die nachgeordneten Funktiosbausteine der asynchronen und der synchronen Programmebene auf.



Anwender Interface

	_NCM_FIFO1			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Konfiguration NC-Satz -	stInVal	boStoreDataDone	-	Quitt "NC-Satz gespeichert"
Maximaldrehzahl 1-16 -	ardiConfigMaxRev	boLineDone	-	Quitt "NC-Satz mit Gerade"
Auflösung 1-16 -	arreConfigResol	bocircleDone	-	Quitt "NC-Satz mit Kreis"
Aktuelle Position 1-16 -	arreActPosAx	boStop	-	Quitt "NC-Satz mit Halt"
Rundachse 1-16 -	arboRoundAxis	boMessageFin	-	Quitt "NC-Satz mit Ende"
Moduloposition 1-16 -	arreModulopos	dwMFunctOut	-	Bitleiste M-Funktionen
Daten in FIFO -	boStoreData	boFifoOverflow	-	Pufferüberlauf
Rücksetzen -	boReset	arreprocessdataOut	-	Prozessdaten 1-16
NC-Sätze ausgeben -	boEnRestoreData	arreSpeedAxis	-	Geschwindigkeit 1-16
Modus Einzelsatz -	boMSingleBlock	uiCountDataSet	-	Ausgegebene NC-Sätze
Einzelsatz ausgeben -	bostep	boError	-	Fehlersignal
Stop-Funktion rücksetzen -	boResStop	uiError	-	Fehler Identnummer
Quitt synchr. M-Funktionen -	boConfMFunct	uiWarn	-	Warnung Identnummer
Bewegung sofort stop -	boStopMotion	parFollErr	-	Schleppabstand 1-16
Einfluss Bahngeschw -	reOverride	StValOutLineCircle	-	Ausgangswert 1-16
Konfiguration M-Funktion -	dwSyncMFunction			
Aktuelle Position einlesen -	boReadinActPos			
Berechnung max. Geschw -	boEnCalcSpeedLimit			
Berechnung tang. Überg. -	boEnCalcTangCon			
Gerätestruktur -	stDevice			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
stInVal	STRUCT	_NCM_ST_GEO_IN Datenstruktur mit den Daten des NC-Satzes, der gespeichert werden soll.
ardiConfigMaxRev	ARRAY	ARRAY{1..16]OF DINT Maximaldrehzahl des Motors der Achsen 1-16 [U/min]
arreConfigResol	ARRAY	ARRAY{1..16]OF REAL Auflösung der Achsen 1-16 [mm/rev.] / [°/rev.]
arreActPosAx	ARRAY	ARRAY[1..16]OF REAL Feld der aktuellen Positionen
arboRoundAxis	ARRAY	ARRAY[1..16]OF BOOL Bestimmung der Achse als Rundachse TRUE: Rundachse Bei Rundachsen erfolgt die Positionierung immer auf dem kürzestem Weg. **Beispiel:** arreModuloposition[1]=360°; aktuelle Position=10°; Zielposition=350° arboRoundAxis[1] = true --> Achse[1] verfährt -20° zur Zielposition (kürzester Weg). arboRoundAxis[1] = false --> Achse[1] verfährt +340° zur Zielposition.
arreModulopos	ARRAY	ARRAY[1..16]OF REALMdouloposition der als Rundachsen konfigurierten Achsen [°]
boStoreData	BOOL	Mit einem Flankenwechsel FALSE->TRUE wird ein an 'stInVal' anliegender NC-Satz in Fifo1 gespeichert Das Signal muss zurückgesetzt werden, wenn 'boStoreDataDone' = TRUE wird (handshake).
boReset	BOOL	• Stoppt alle Bewegungen der Achsen. • Setzt alle Fehlermeldungen zurück. • Setzt alle M-Funktionen zurück. • Setzt alle internen Zustände zurück.
boEnRestoreData	BOOL	Feigabesignal zur Ausgabe der NC-Sätze. Nur wenn 'boEnRestoreData' = TRUE ist, können NC-Sätze ausgegeben werden.
boMSingleBlock	BOOL	Das Signal aktiviert den Modus Einzelsatz, d.h. es wird jeweils ein NC-Datensatz bearbeitet. Der nachfolgende Datensatz wird jeweils mit 'bostep' gestartet.
bostep	BOOL	Startet die Ausgabe eines NC-Satzes (Nur in Verbindung mit 'boMSingleBlock' = TRUE).
boResStop	BOOL	Quittierungssignal für einen programmierten Halt 'iCharElem'=9 (Halt). Bei einem Flankenwechsel FALSE -> TRUE wird der nächste NC-Satz ausgegeben.
boConfMFunct	BOOL	Quittierungssignal für alle synchronen M-Funktionen.
boStopMotion	BOOL	Signal stoppt alle Achsbewegungen unverzüglich.
reOverride	REAL	Faktor auf die programmierte Bahngeschwindigkeit (0-200%)

Name	Typ	Beschreibung
dwSyncMFunction	DWORD	Bitleiste zum Konfigurieren der M-Funktionen 1..32 als synchrone oder unsynchrone M-Funktion. • unsynchron bedeutet: M-Funktion wird mit Start Ausgabe des jeweiligen NC-Satzes TRUE. Mit Ausgabe des nächsten NC-Satzes wird die M-Funktion FALSE • synchron bedeutet: M-Funktion wird mit Start Ausgabe des jeweiligen NC-Satzes TRUE. Die Ausgabe des nächsten NC-Satzes ist gesperrt. Wird 'boConfMFunct' = TRUE, wird die M-Funktion FALSE und der nächste NC-Satz wird ausgegeben. Beispiel zum Konfigurieren von synchronen M-Funktionen: dwSyncMFunct:=16#FFFF0000 bedeutet: M-function 1..16 unsynchron M-function 17..32 synchron
boReadinActPos	BOOL	Um eine absolute Positionierung zu ermöglichen, ist es notwendig die aktuelle Position der Achsen einzulesen. Im Normalfall wird 'boReadInActPos' beim Abspeichern des 1.NC-Satzes auf TRUE gesetzt. Damit ist der Startpunkt für die interne Berechnung die aktuelle Position der Achsen zum Zeitpunkt des Abspeicherns. Bei allen weiteren NC-Sätzen ist 'boReadInActPos' = FALSE. Damit ist der Startpunkt für die interne Berechnung immer der Endpunkt des vorherigen Datensatzes.
boEnCalcSpeedLimit	BOOL	Freigabesignal zur internen Speedlimit Berechnung aufgrund der maximalen Drehzahl ('ardiConfigMaxRev'). TRUE: Berechnung wird ausgeführt. FALSE: Berechnung wird nicht ausgeführt.
boEnCalcTangCon	BOOL	Freigabesignal zur automatischen Berechnung eines tangentiellen Übergangs zwischen zwei NC-Fahrsätzen. TRUE: Die Berechnung erfolgt automatisch. Wird ein tangentieller Übergang erkannt, werden beide Fahrsätze mit der programmierten Geschwindigkeit gefahren. Es erfolgt kein Herunterrampen auf V=0 nach dem 1.Fahrsatz. FALSE: Es wird keine Berechnung durchgeführt. Ein tangetieller Übergang kann jedoch über die Eingangsstruktur '_NCM_ST_GEO_IN' boTangential „manuell“ vorgegeben werden.
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boStoreDataDone	BOOL	FALSE -> TRUE: Ein NC-Satz wurde erfolgreich abgespeichert. Das Eingangssignal 'boStoreData' wird FALSE (handshake): TRUE -> FALSE
boLineDone	BOOL	FALSE -> TRUE: Ein NC-Satz mit Geradeninterpolation wurde erfolgreich ausgegeben. Der nächste NC-Satz wird ausgegeben: TRUE -> FALSE
bocircleDone	BOOL	FALSE -> TRUE: Ein NC-Satz mit Kreisinterpolation wurde erfolgreich ausgegeben. Der nächste NC-Satz wird ausgegeben: TRUE -> FALSE

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>boStop</td><td>BOOL</td><td>FALSE -> TRUE: Ein NC-Satz mit programmiertem Halt wurde erfolgreich ausgegeben.
Der nächste NC-Satz wird ausgegeben(boResStop = TRUE): TRUE -> FALSE.</td></tr>
<tr><td>boMessageFin</td><td>BOOL</td><td>FALSE -> TRUE: Ein NC-Satz mit Programmende wurde erfolgreich ausgegeben.
Ein neues Programm wird gestartet oder boReset wird TRUE: TRUE -> FALSE</td></tr>
<tr><td>dwMFunctOut</td><td>DWORD</td><td>Bitleiste mit den gesetzten M-Funktionen.
Beispiel:
NC-Satz mit den M-Funktionen
uiFirstMFunct = 5
uiSecondMFunct = 9
uiThirdMFunct=13
hat dwMFunctOut den Wert 16#00001110.</td></tr>
<tr><td>boFifoOverflow</td><td>BOOL</td><td>FALSE -> TRUE: FiFo ist voll.
Es ist nicht mehr möglich, weitere Datensätze abzuspeichern (boStoreDataDone = FALSE).</td></tr>
<tr><td>arreprocessdataOut</td><td>ARRAY</td><td>ARRAY[1..16]OF REAL
Ausgabe der im Datensatz programmierten Prozessdaten (Achse 1..16).</td></tr>
<tr><td>arreSpeedAxis</td><td>ARRAY</td><td>ARRAY[1..16]OF REAL
Geschwindigkeitistwerte der Achsen</td></tr>
<tr><td>uiCountDataSet</td><td>UINT</td><td>Zähler der ausgegebenen NC-Sätze.</td></tr>
<tr><td>boError</td><td>BOOL</td><td>FALSE -> TRUE: Die Plausibilitätsüberprüfung hat einen Fehler festgestellt.
'boReset' wird TRUE: TRUE -> FALSE</td></tr>
<tr><td>uiError</td><td>UINT</td><td>Fehler Identnummer: Diagnosenummer wird ausgegeben
<table>
<tr><td>0</td><td>Kein Fehler</td></tr>
<tr><td>5</td><td>Kreisradius ist Null</td></tr>
<tr><td>10</td><td>Falsch programmierte M-Funktion</td></tr>
<tr><td>11</td><td>Falsch programmierter Code (iCharElem)</td></tr>
<tr><td>12</td><td>Falsch programmierte Kreisebene (iCharPlane)</td></tr>
<tr><td>13</td><td>Programmierte Bahngeschwindigkeit ist Null</td></tr>
<tr><td>14</td><td>Programmierte Beschleunigung ist Null</td></tr>
<tr><td>15</td><td>Programmierte Verzögerung ist Null</td></tr>
</table></td></tr>
<tr><td>uiWarn</td><td>UINT</td><td>Warnung Identnummer: Diagnosenummer wird ausgegeben
<table>
<tr><td>0</td><td>Keine Warnung</td></tr>
<tr><td>1</td><td>Länge der Geraden ist Null</td></tr>
<tr><td>6</td><td>Radius am Startpunkt des Kreises ist unterschiedlich zum Radius am Endpunkt des Kreises.</td></tr>
</table>
'uiWarn' wird zurückgesetzt, wenn der nächste NC-Satz ausgegeben wird.</td></tr>
</table>

Name	Typ	Beschreibung
parFollErr	POINTER	POINTER TO ARRAY[1..16]OF DINT Um einen genauen Halt der Achsen zu ermöglichen, ist es notwendig, den aktuellen Schleppabstand jeder Achse mit einem "In Position"-Fenster 'ardiVerifyFollErr' zu vergleichen. Erst wenn der Schleppabstand innerhalb des "In Position"-Fensters ist, kann der nächste NC-Satz ausgegeben werden. In der synchronen Task (FPLC_PRG) können die aktuellen Schleppabstände der Achsen 1-16 in das Array 'ardiFollErrAx [1..16]' geschrieben werden: PLC_PRG.fbFifo1.parFollErr^[1] = Schleppabstand Achse1 : : PLC_PRG.fbFifo1.parFollErr^[16] = Schleppabstand Achse16 Werden die aktuellen Schleppabstände nicht in das Array geschrieben, wird intern mit Schleppabstand=0 gerechnet und somit ist die Bedingung "In Position" erreicht.
StValOutLineCircle	STRUCT	_NCM_ST_VALOUT_LINE_CIRCLE Datenstruktur, die die Ausgangsinkremente der Achsen 1..16 enthält. Beispiel: PLC_PRG.fbFifo1.StValOutLineCircle.diOutVal[1] enthält die Ausgangsinkremente für Achse1

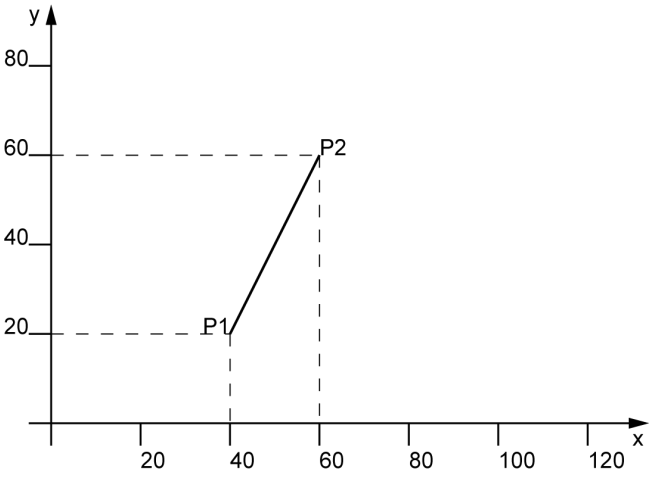
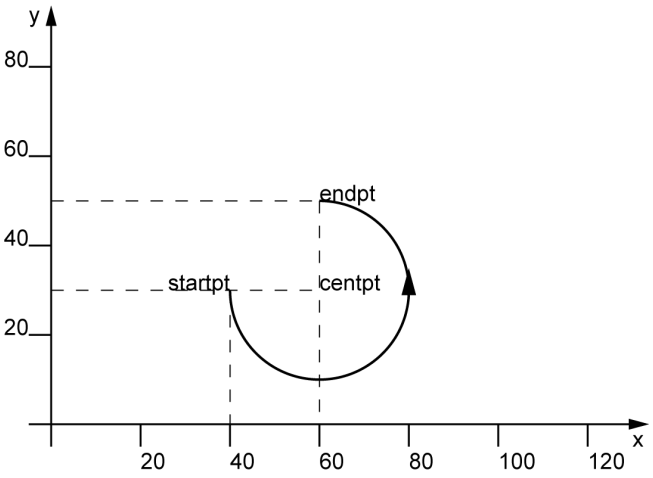
Verwendungshinweis im CoDeSys-Programm

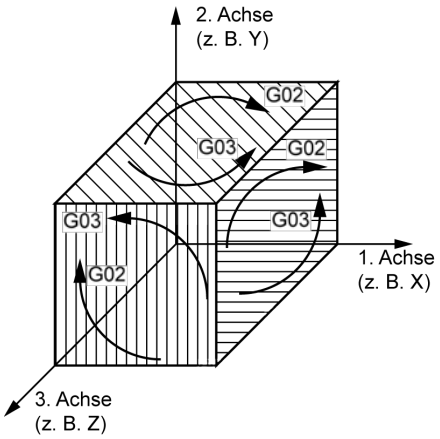
PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
_NCM_FIFO1	_NCM_FIFO1.FastFunctionFifo2

_NCM_ST_GEO_IN (ST)

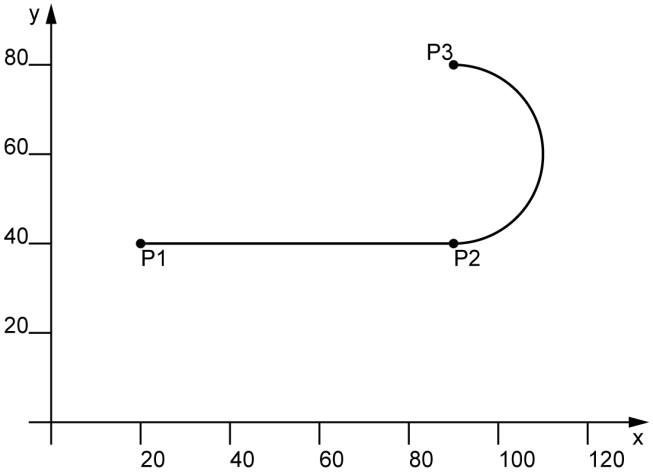
Die Struktur '_NCM_ST_GEO_IN' enthält alle relevanten Daten, um einen NC-Satz zu konfigurieren.

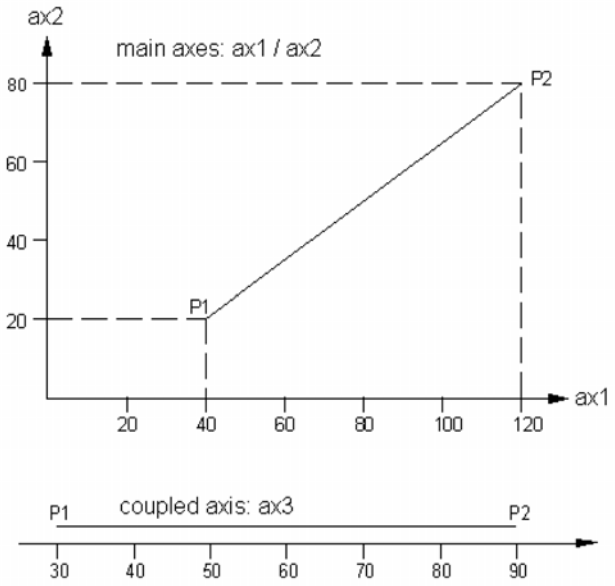
Name	Typ	Beschreibung
iCharElem	INT	Code für den NC-Satz 1 – Geradeninterpolation 2 – Kreisinterpolation clockwise (cw) 3 – Kreisinterpolation counter clockwise (ccw) 4 – Verzögerungszeit 9 – Halt 30 – Programm Ende

Name	Typ	Beschreibung
arreEndPosAx	ARRAY	ARRAY[1..16] OF REAL Legt den Endpunkt (Bsp: P2) der jeweiligen Achse im NC-Satz fest [mm]. Beispiel: NC-Satz für Geradeninterpolation iCharElem := 1 arreEndPosAx[1] := 60 mm arreEndPosAx[2] := 60 mm  P1: Endpunkt NC-Satz (n) P2: Endpunkt NC-Satz (n+1)
rePMAx1; rePMAx2; rePMAx3	REAL	Position Kreismittelpunkt [mm] Achse 1 (X), Achse 2 (Y), Achse 3 (Z) Beispiel: NC-Satz für Kreisinterpolation cw in Ebene X/Y iCharElem := 3 iCharPlane := 17 rePMAx1 := 60 mm rePMAx2 := 30 mm arreEndPosAx[1] := 60 mm arreEndPosAx[2] := 50 mm 

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>iCharPlane</td><td>INT</td><td>Code zum Definieren der Kreisebene
<table>
<tr><th>iCharPlane</th><th>Kreisebene</th></tr>
<tr><td>17</td><td>X/Y (Achse 1 / Achse 2)</td></tr>
<tr><td>18</td><td>X/Z (Achse 1 / Achse 3)</td></tr>
<tr><td>19</td><td>Y/Z (Achse 2 / Achse 3)</td></tr>
</table>

</td></tr>
<tr><td>uiFirstMFunct;
uiSecondMFunct;
uiThirdMFunct</td><td>UINT</td><td>M-Funktionen

Insgesamt sind 32 M-Funktionen verfügbar.
Pro NC-Satz sind bis zu 3 M-Funktionen programmierbar.
Jede M-Funktion kann als synchrone oder unsynchrone M-Funktion konfiguriert werden.
uiFirstMFunct (1. M-Funktion im NC-Satz)
uiSecondMFunct (2. M-Funktion im NC-Satz)
uiThirdMFunct (3. M-Funktion im NC-Satz)
Beispiel:
Ein NC-Satz mit 3 M-Funktionen
Die 1. M-Funktion soll M-Funktion Nummer 5 sein.
Die 2. M-Funktion soll M-Funktion Nummer 9 sein.
Die 3. M-Funktion soll M-Funktion Nummer 13 sein.
uiFirstMFunct := 5
uiSecondMFunct := 9
uiThirdMFunct := 13</td></tr>
<tr><td>reDelaytime</td><td>REAL</td><td>Verzögerungszeit bis zur Ausgabe des nächsten NC-Satzes [s].
Beispiel:
Verzögerung 3.5 s
iCharElem := 4
reDelaytime := 3.5 s</td></tr>
<tr><td>ardiVerifyFollErr</td><td>ARRAY</td><td>ARRAY[1..16] OF DINT
Vergleichsfenster für Genauhalt (siehe 'parFollError')
Beispiel:
Vergleichsfenster für Achse1 soll 100 Inkr sein.
ardiVerifyFollErr[1] := 100</td></tr>
<tr><td>reVelocity</td><td>REAL</td><td>Angabe der Bahngeschwindigkeit im NC-Satz
Linear: [mm/s]; Kreis: [°/s].</td></tr>
<tr><td>reAcceleration</td><td>REAL</td><td>Angabe der Beschleunigung im NC-Satz
Linear: [mm/s^2]; Kreis: [°/s^2].</td></tr>
</table>

Name	Typ	Beschreibung
reDecceleration	REAL	Angabe der Verzögerung im NC-Satz Linear: [mm/s^2]; Kreis: [°/s^2].
reACCJerk	REAL	Angabe der Ruckbegrenzung bei der Beschleunigung Linear: [mm/s^3]; Kreis: [°/s^3]. Bei reAccJerk=0 ist der Ruck unendlich.
reDecJerk	REAL	Angabe der Ruckbegrenzung bei der Verzögerung Linear: [mm/s^3]; Kreis: [°/s^3]. Bei reDecJerk=0 ist der Ruck unendlich.
boTangential	BOOL	In manchen Applikationen ist es ungünstig, wenn die Bahngeschwindigkeit beim Übergang von NC-Fahrsatz (n) zu NC-Fahrsatz (n+1) auf Null abgebremst wird. (Verbindung zwischen Gerade P1/P2 und Kreis P2/P3) boTangential := TRUE im NC-Satz des Kreises P2/P3 bedeutet: Die Bewegung der Achsen beschleunigt an P1, bis die programmierte Bahngeschwindigkeit erreicht ist, und verzögert erst zum Stillstand an P3. Eine tangentielle Bewegung zwischen zwei NC-Fahrsätzen ist erst dann möglich, wenn mindestens 2 NC-Sätze abgespeichert sind. 
boRelativ	BOOL	Relative Positionierung Soll ein NC-Fahrsatz nicht absolut, sondern relativ ausgeführt werden, muss 'boRelativ' auf true gesetzt werden. Beispiel: Relative Geradeninterpolation der Achse1 um 100mm. iCharElem := 1 arreEndPosAx[1] := 100 mm boRelativ := TRUE
arreProzessdata	ARRAY	ARRAY[1..16] OF REAL Es sind insgesamt 16 programmierbare Prozessdaten pro NC-Satz verfügbar. Beim Ausgeben des NC-Satzes werden die programmierten Prozessdaten in die Ausgangsvariable 'arreProcessdataOut' geschrieben. Beispiel: An einer NC-Fräsmaschine soll die Schnittgeschwindigkeit des Fräsers von NC-Satz zu NC-Satz eingestellt werden können.

Name	Typ	Beschreibung
wCoupledAxis	WORD	Mit der Bitleiste 'wCoupledAxis' kann jede Achse als MitschleppAchse konfiguriert werden. Die Koordinatenangaben einer Mitschleppachse gehen nicht in die Berechnung der Bahngeschwindigkeit aller anderen Achsen mit ein. Beispiel: Die Bahngeschwindigkeit wird für den Weg von P1 nach P2 für Achse1 und Achse2 bestimmt. Die Bahngeschwindigkeit für Achse3 wird so berechnet, dass sie abhängig vom ihrem zurückzulegenden Weg gleichzeitig wie Achse1 und Achse2 am Zielpunkt ankommt. Achse3 als Mitschleppachse: wCoupledAxis := 16#0004 

4.2.1.2.8 08_Sequence

4.2.1.2.8.1 SEQUENCE_STATE_CHECK (FB)

Der Funktionsbaustein 'SEQUENCE_STATE_CHECK' ermöglicht die Überprüfung einer Schrittkette.

An den Ausgängen 'uiStateTm..' wird der Zustand der Schrittkette zum entsprechenden Zeitpunkt angezeigt. D. h. man kann bis zu 10 Zustandswechsel der Schrittkette nachverfolgen.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

		SEQUENCE_STATE_CHECK			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Zustand der Schrittkette	-	uiState	uiStateTm1	-	uiState(t-1)
			uiStateTm2	-	uiState(t-2)
			uiStateTm3	-	uiState(t-3)
			uiStateTm4	-	uiState(t-4)
			uiStateTm5	-	uiState(t-5)
			uiStateTm6	-	uiState(t-6)
			uiStateTm7	-	uiState(t-7)
			uiStateTm8	-	uiState(t-8)
			uiStateTm9	-	uiState(t-9)
			uiStateTm10	-	uiState(t-10)

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
uiState	UINT	Zustand der Schrittkette

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
uiStateTm1	UINT	Zustand der Schrittkette zum Zeitpunkt (t-1)
uiStateTm2	UINT	Zustand der Schrittkette zum Zeitpunkt (t-2)
uiStateTm3	UINT	Zustand der Schrittkette zum Zeitpunkt (t-3)
uiStateTm4	UINT	Zustand der Schrittkette zum Zeitpunkt (t-4)
uiStateTm5	UINT	Zustand der Schrittkette zum Zeitpunkt (t-5)
uiStateTm6	UINT	Zustand der Schrittkette zum Zeitpunkt (t-6)
uiStateTm7	UINT	Zustand der Schrittkette zum Zeitpunkt (t-7)
uiStateTm8	UINT	Zustand der Schrittkette zum Zeitpunkt (t-8)
uiStateTm9	UINT	Zustand der Schrittkette zum Zeitpunkt (t-9)
uiStateTm10	UINT	Zustand der Schrittkette zum Zeitpunkt (t-10)

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.1.2.8.2 SEQUENCE_STATE_TIMEOUT (FB)

Der Funktionsbaustein 'SEQUENCE_STATE_TIMEOUT' dient der Zeitüberwachung.
Wird der aktuelle Zustand (Schritt) nicht in der angegebenen Zeit verlassen, wird der Ausgang 'boTimeOut' gesetzt.

Der Aufruf dieser Funktion ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	SEQUENCE_STATE_TIMEOUT			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Aktueller Zustand -	uiActualState	boTimeOut	-	Timeout überschritten
Timeout-Zeit -	tTimeOut			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
uiActualState	UINT	Aktueller Zustand (Schritt)
tTimeOut	TIME	Timeout Zeit [ms] Standardwert: tTimeOut = 10 sec

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boTimeOut	BOOL	Timeout-Zeit überschritten

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

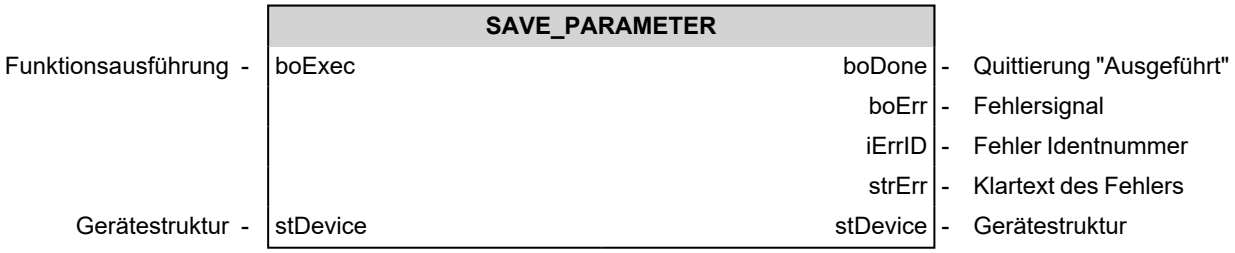
4.2.1.2.9 09_Parameter

4.2.1.2.9.1 SAVE_PARAMETER (FB)

Der Funktionsbaustein 'SAVE_PARAMETER' sichert den Parametersatz eines Antriebs. Nach einem Austausch der Hardware, kann der Parametersatz durch den Baustein 'RESTORE_PARAMETER' wiederhergestellt werden. Die Parameter werden als FILE auf der Steuerung hinterlegt.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde

Name	Typ	Beschreibung
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
strErr	STRING	Klartext des Fehlers
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung Wert 1: Fehler beim Auslesen der ID Wert 2: Timeout ID auslesen Wert 3: Fehler Datei schreiben Wert 4: Timeout Datei schreiben Wert 10: Es wurde kein Dateiname 'strFileName' übergeben

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
SAVE_PARAMETER	

4.2.1.2.9.2 RESTORE_PARAMETER (FB)

Mit dem Funktionsbaustein 'RESTORE_PARAMETER' können gespeicherte Parameter auf einen Antrieb geschrieben werden. Die Parameter müssen vorher durch den Baustein 'SAVE_PARAMETER' gesichert worden sein.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	RESTORE_PARAMETER			
Funktionsausführung -	boExec	boDone	-	Quittierung "Ausgeführt"
		boErr	-	Fehlersignal
		iErrID	-	Fehler Identnummer
		strErr	-	Klartext des Fehlers
		usiPercent	-	Prozessfortschritt
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.

Ausgangsvariablen

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>boDone</td><td>BOOL</td><td>Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde</td></tr>
<tr><td>boErr</td><td>BOOL</td><td>Der Funktionsbaustein befindet sich im Zustand Fehler
<table><tr><td>FALSE</td><td>Kein Fehler (Kommandierung zulässig oder Warnung)</td></tr><tr><td>TRUE</td><td>Fehler</td></tr></table></td></tr>
<tr><td>iErrID</td><td>INT</td><td>Fehler Identnummer: Diagnosenummer wird ausgegeben
<table><tr><td colspan="2">iErrID = 0</td><td>Kein Fehler</td></tr><tr><td>iErrID ≠ 0</td><td>boErr = TRUE</td><td>Fehler</td></tr><tr><td>iErrID ≠ 0</td><td>boErr = FALSE</td><td>Warnung</td></tr></table><table><tr><th>Wert</th><th>Bedeutung</th></tr><tr><td>1</td><td>Fehler beim lesen der Datei</td></tr><tr><td>2</td><td>Timeout Datei lesen</td></tr><tr><td>3</td><td>Timeout ID schreiben</td></tr><tr><td>4</td><td>Listenlänge ID größer als 4096 Byte</td></tr><tr><td>6</td><td>Fehler 'WRITE_ID_LIST'</td></tr><tr><td>7</td><td>Unbekannte Datenlänge der ID</td></tr><tr><td>10</td><td>Es wurde kein Dateiname 'strFileName' übergeben</td></tr></table></td></tr>
<tr><td>strErr</td><td>STRING</td><td>Klartext des Fehlers</td></tr>
<tr><td>usiPercent</td><td>USINT</td><td>Prozessfortschritt</td></tr>
</table>

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
RESTORE_PARAMETER	

4.2.1.3 07_Visualization

4.2.1.3.1 NUMPAD (FB)

Der Funktionsbaustein 'NUMPAD' ist die Logik hinter der Visualisierung 'ViNUMPAD'. Er realisiert die Eingaben und vergleicht den aktuellen Wert mit den vorgegebenen Grenzen.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

		NUMPAD			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Ganzzahlig	-	boInteger	boDone	-	Quittierung "Ausgeführt"
Wert vor der Eingabe	-	reStartValue	reDisplay	-	Eingeg. Wert als REAL
Untere Grenze	-	reMinimalValue	diDisplay	-	Eingeg. Wert als DINT
Obere Grenze	-	reMaximalValue	strDisplay	-	Eingeg. Wert als STRING
Button Farbe	-	dwButtonColor			
Button Alarmfarbe	-	dwButtonAlarmColor			
Button Textfarbe	-	dwButtonTextColor			
Button Textgröße	-	wButtonTextHeight			
Anzeige Textgröße	-	wDisplayTextHeight			
Anzeige Textfarbe	-	dwDisplayTextColor			
Anzeige Farbe	-	dwDisplayColor			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boInteger	BOOL	TRUE: nur ganzzahlige Eingabe möglich
reStartValue	REAL	Wert vor der Eingabe

Name	Typ	Beschreibung
reMinimalValue	REAL	Unterer Grenzwert der Eingabe (reMinimalValue = reMaximalValue: keine Überwachung)
reMaximalValue	REAL	Oberer Grenzwert der Eingabe (reMinimalValue = reMaximalValue: keine Überwachung)
dwButtonColor	DWORD	Farbe der Buttons (16#00BBGGRR) Vorbelegung: Blau (16#00FF0000)
dwButtonAlarmColor	DWORD	Farbe der Buttons bei der Betätigung (16#00BBGGRR) Vorbelegung: Grün (16#0000FF00)
dwButtonTextColor	DWORD	Textfarbe der Buttons (16#00BBGGRR) Vorbelegung: Weiß (16#00FFFFFF)
wButtonTextHeight	WORD	Textgröße der Buttons
wDisplayTextHeight	WORD	Textgröße der Anzeige
dwDisplayTextColor	DWORD	Textfarbe der Anzeige (16#00BBGGRR) Vorbelegung: Rot (16#000000FF)
dwDisplayColor	DWORD	Farbe der Anzeige (16#00BBGGRR) Vorbelegung: Dunkelgrau (16#00555555)

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
reDisplay	REAL	Ausgabe des eingegebenen Wertes als Gleitkommazahl
diDisplay	DINT	Ausgabe des eingegebenen Wertes als ganze Zahl
strDisplay	STRING	Ausgabe des eingegebenen Wertes als String

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
NUMPAD	

4.2.1.4 08_Devices

4.2.1.4.1 fstrGetControllerTyp (F)

Die Funktion 'fstrGetControllerTyp' liefert den Steuerungstyp zurück.
Die Funktion wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

fstrGetControllerTyp		
fstrGetControllerTyp	-	Rückgabewert

Ausgangsvariable

Name	Typ	Beschreibung
fstrGetControllerTyp	STRING	Steuerungstyp

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
fstrGetControllerTyp	

4.2.1.5 Anhang

EN_POS_J_ARITHMETIK

Zusatzinformation zum Eingang 'enArithmetik', verwendet z. B. beim AFL Applikationsbaustein POSITION_JERK_1 (FB). Siehe 'POSITION_JERK_1 (FB)' auf Seite 380.

In der Betriebsart Lageregelung stehen 64 Bit und 32 Bit Positionierbausteine zu Verfügung.

64 Bit Positionierbausteine erreichen eine höhere Genauigkeit durch die höhere Auflösung als 32 Bit Positionierbausteine.

Mit dem Eingang 'enArithmetik' wird die Auswahl zwischen 64 Bit und 32 Bit Positionierbausteinen eingestellt.

Bei der Standardeinstellung 'enAutomaticSelection' wird in Abhängigkeit der eingesetzten Steuerung und der verwendeten CODESYS Version, die Auswahl automatisch eingestellt.

Mit 'en64BitArithmetik' bzw. 'en32BitArithmetik' kann die Auswahl vom Anwender manuell eingestellt werden.

Die maximale Größe der von der AMK Steuerung unterstützten Datentypen ist abhängig von der eingesetzten CODESYS Version 2 oder 3.

Der Zusammenhang zwischen Steuerung und CODESYS Version wird in der nachfolgenden Tabelle dargestellt.

Steuerung	CODESYS V2	CODESYS V3
iSA	REAL (32 Bit)	LREAL (64 Bit)
A4	LREAL (32 Bit)	LREAL (64 Bit)
A5	LREAL (64 Bit)	LREAL (64 Bit)
A6	REAL (64 Bit)	LREAL (64 Bit)

64 Bit Positionierbausteine (en64BitArithmetik) dürfen nicht mit einer 32 Bit CODESYS Version verwendet werden. Eine Konvertierung von 64 Bit in 32 Bit ist ohne Verluste nicht möglich. Die Positionierung wird ungenau und die Zielposition verfehlt. Folge Statusbits boDone/boInPos werden nicht TRUE.

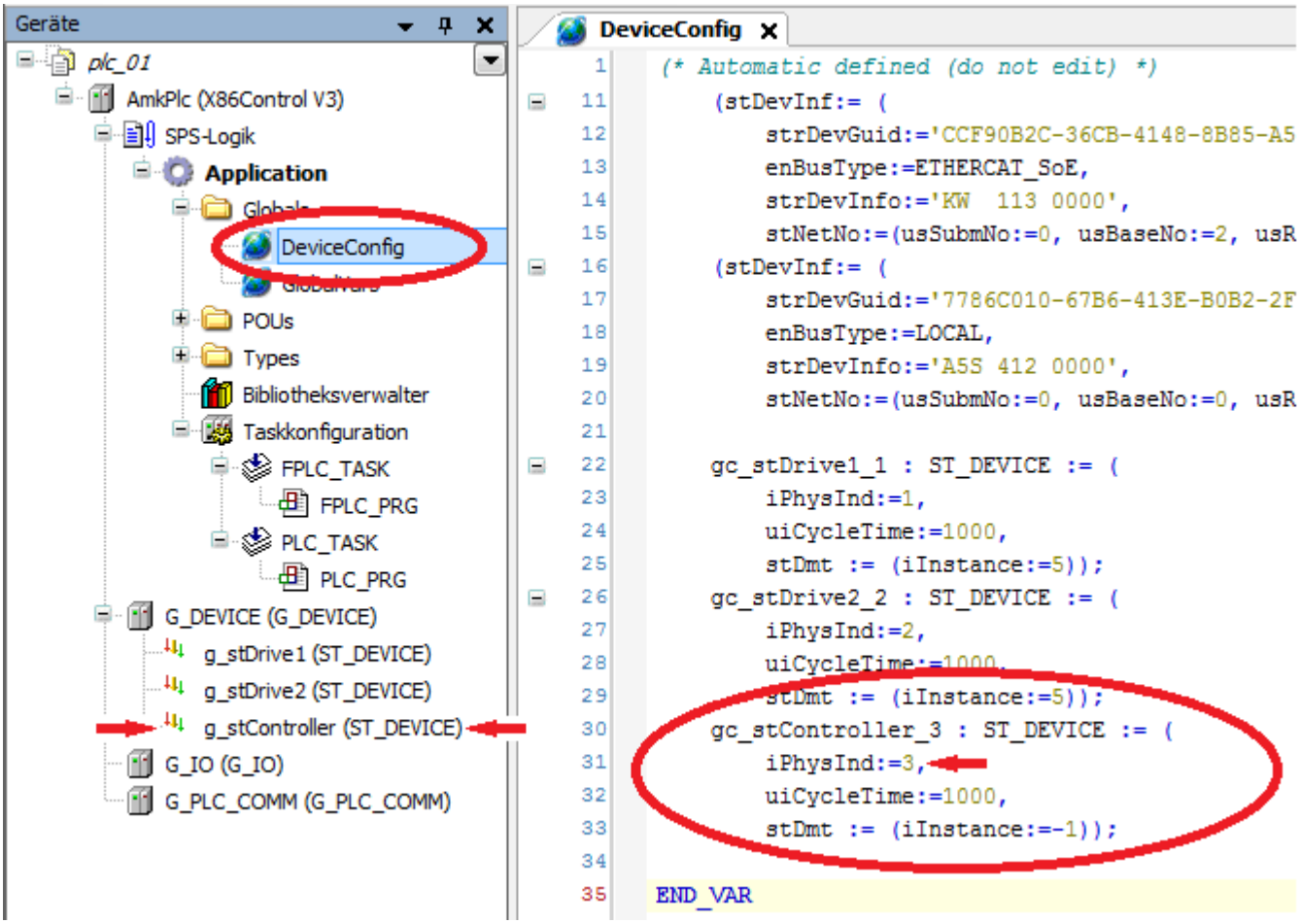
32 Bit Positionierbausteine (en32BitArithmetik) können auf einer 64 Bit CODESYS Version verwendet werden. Die Zielposition wird erreicht, jedoch wird durch die nicht genutzt Auflösung Genauigkeit verschenkt.

Beispiel automatische Bestimmung

Beispiel in dem die automatische Bestimmung, ob 32 Bit oder 64 Bit Arithmetik, nicht funktioniert.

Die symbolischen Gerätevariablen (Typ ST_DEVICE) wurden manuell in der Reihenfolge g_stDevice1, g_stDevice2, g_stController eingefügt.

Die Steuerung (stController) wird deshalb auf Platz 3 gelistet (iPhysInd:=3). Die automatische Bestimmung funktioniert nur, wenn die Steuerung als an Platz 1 iPhysInd:=1 (gelistet) wird.



4.2.2 Bibliotheksbeschreibung AmkAflConfig.lib

4.2.2.1 02_Parameter Access

4.2.2.1.1 ID_READ_10IDS_DINT (FB)

Der Funktionsbaustein 'ID_READ_10IDS_DINT' liest Parameter, jedoch keine Listenparameter, aus AMK-Geräten. Es können bis zu 10 ID-Nummern im Format Double Integer (DINT) ausgelesen werden. Jedem Eingang für eine ID-Nummer ist ein Ausgang für den ID-Wert zugeordnet.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

		ID_READ_10IDS_DINT			
Funktionsausführung	-	boExec	boDone	-	Quittierung "Ausgeführt"
auszulesende ID-Nr	-	uiIDNo1	diOutValNo1	-	gelesener ID-Wert
auszulesende ID-Nr	-	uiIDNo2	diOutValNo2	-	gelesener ID-Wert
:	-	:	:	-	:
:	-	:	:	-	:
auszulesende ID-Nr	-	uiIDNo10	diOutValNo10	-	gelesener ID-Wert
Parametersatz	-	uiParInst	boErr	-	Fehlersignal
			iErrID	-	Fehler Identnummer
			enErrName	-	Name fehlerh Baustein
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
uiIDNo1 ... uiIDNo10	UINT	Nummer des Parameters, der gelesen wird
uiParInst	UINT	Parametersatz- oder Instanz-Nummer

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
diOutValNo1 ... diOutValNo10	DINT	Ausgelesener ID-Wert für die am Eingang n angegebene ID-Nummer
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: READ_ID_DINT; Bibliothek: AmkSystem.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
ID_READ_10IDS_DINT	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.1.2 ID_READ_5IDS_DINT (FB)

Der Funktionsbaustein 'ID_READ_5IDS_DINT' liest Parameter, jedoch keine Listenparameter, aus AMK-Gräten. Es können bis zu 5 ID-Nummern im Format Double Integer (DINT) ausgelesen werden. Jedem Eingang für eine ID-Nummer ist ein Ausgang für den ID-Wert zugeordnet.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	ID_READ_5IDS_DINT			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
auszulesende ID-Nr -	uiIDNo1	diOutValNo1	-	gelesener ID-Wert
auszulesende ID-Nr -	uiIDNo2	diOutValNo2	-	gelesener ID-Wert
-	:	:	-	
-	:	:	-	
auszulesende ID-Nr -	uiIDNo5	diOutValNo5	-	gelesener ID-Wert
Parametersatz -	uiParInst	boErr	-	Fehlersignal
		iErrID	-	Fehler Identnummer
		enErrName	-	Name fehlerh Baustein
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
uiIDNo1 / 2 / 3 / 4 / 5	UINT	Nummer des Parameters, der gelesen wird
uiParInst	UINT	Parametersatz- oder Instanz-Nummer

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
diOutValNo1 / 2 / 3 / 4 / 5	DINT	Ausgelesener ID-Wert für die am Eingang n angegebene ID-Nummer
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: READ_ID_DINT; Bibliothek: AmkSystem.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
ID_READ_5IDS_DINT	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.1.3 ID_READ_NODE_LIST (FB)

Der Funktionsbaustein 'ID_READ_NODE_LIST' liest die Knotenliste (ID34211 'Knotenliste 2') des Gerätes 'stDevice'. Die gefundenen Knoten und die zugehörigen Informationen werden in der Struktur 'stList' abgelegt.
Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

		ID_READ_NODE_LIST			
Funktionsausführung	-	boExec	boDone	-	Quittierung "Ausgeführt"
Instanz	-	uiParInst	boErr	-	Fehlersignal
			iErrID	-	Fehler Identnummer
Knotenliste	-	stList	stList	-	Knotenliste
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
uiParInst	UINT	Parametersatz- oder Instanz-Nummer, aus der die Knotenliste gelesen werden soll (z.B. Instanz 5 für EtherCAT bei der A5)

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben Baustein: ID_READ_1 — Bibliothek: AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stList	STRUCT	ST_NODE_LIST Die Struktur enthält die maximale und aktuelle Größe der Liste, einen Header sowie die jeweiligen Knoten mit ihren Inhalten
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
ID_READ_NODE_LIST	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.1.4 ID_WRITE_10IDS_DINT (FB)

Der Funktionsbaustein 'ID_WRITE_10IDS_DINT' schreibt Parameter, jedoch keine Listenparameter, in AMK-Geräte. Es können bis zu 10 ID-Nummern im Format Double Integer (DINT) geschrieben werden. Jedem Eingang für eine ID-Nummer ist ein Eingang für den ID-Wert zugeordnet.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

		ID_WRITE_10IDS_DINT			
Funktionsausführung	-	boExec	boDone	-	Quittierung "Ausgeführt"
zu schreibende ID-Nr	-	uiIDNo1	boErr	-	Fehlersignal
Wert für ID	-	diValNo1	iErrID	-	Fehler Identnummer
zu schreibende ID-Nr	-	uiIDNo2	enErrName	-	Name fehlerh Baustein
Wert für ID	-	diValNo2			
:	-	:			
:	-	:			
zu schreibende ID-Nr	-	uiIDNo10			
Wert für ID	-	diValNo10			
Parametersatz	-	uiParInst			
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
uiIDNo1 ... uiIDNo10	UINT	Nummer des Parameters, der gelesen wird
diValNo1 ... diValNo10	DINT	Zu schreibender ID-Wert für die am Eingang n angegebene ID-Nummer
uiParInst	UINT	Parametersatz- oder Instanz-Nummer

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein**: WRITE_ID_DINT **Bibliothek**: AmkSystem.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
ID_WRITE_10IDS_DINT	

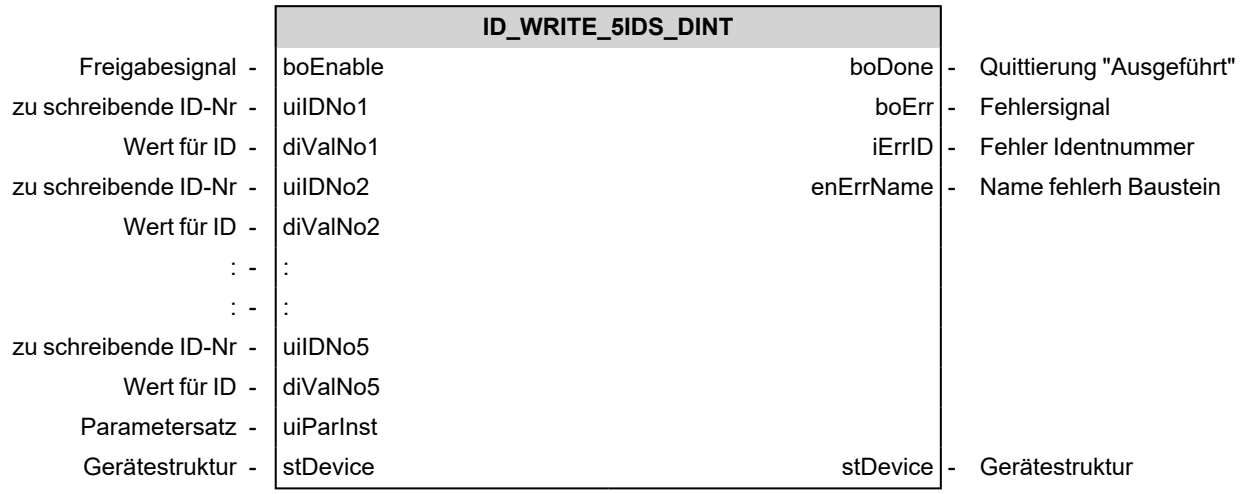
Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.1.5 ID_WRITE_5IDS_DINT (FB)

Der Funktionsbaustein 'ID_WRITE_5IDS_DINT' schreibt Parameter, jedoch keine Listenparameter, in AMK-Geräte. Es können bis zu 5 ID-Nummern im Format Double Integer (DINT) geschrieben werden. Jedem Eingang für eine ID-Nummer ist ein Eingang für den ID-Wert zugeordnet.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
uiIDNo1 ... uiIDNo5	UINT	Nummer des Parameters, der gelesen wird
diValNo1 ... diValNo5	DINT	Zu schreibender ID-Wert für die am Eingang n angegebene ID-Nummer
uiParInst	UINT	Parametersatz- oder Instanz-Nummer

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: WRITE_ID_DINT; Bibliothek: AmkSystem.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
ID_WRITE_5IDS_DINT	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.1.6 ID_WRITE_5IDS_TEMP_DINT (FB)

Der Funktionsbaustein 'ID_WRITE_5IDS_TEMP_DINT' schreibt temporäre Parameter, jedoch keine Listenparameter, in AMK-Geräte. Es können bis zu 5 ID-Nummern im Format Double Integer (DINT) geschrieben werden. Jedem Eingang für eine ID-Nummer ist ein Eingang für den ID-Wert zugeordnet.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

		ID_WRITE_5IDS_TEMP_DINT			
Funktionsausführung	-	boExec	boDone	-	Quittierung "Ausgeführt"
zu schreibende ID-Nr	-	uiIDNo1	boErr	-	Fehlersignal
Wert für ID	-	diValNo1	iErrID	-	Fehler Identnummer
zu schreibende ID-Nr	-	uiIDNo2	enErrName	-	Name fehlerh Baustein
Wert für ID	-	diValNo2			
:	-	:			
:	-	:			
zu schreibende ID-Nr	-	uiIDNo5			
Wert für ID	-	diValNo5			
Parametersatz	-	uiParInst			
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
uiIDNo1 ... uiIDNo5	UINT	Nummer des Parameters, der geschrieben wird
diValNo1 ... diValNo5	DINT	Zu schreibender ID-Wert für die am Eingang n angegebene ID-Nummer
uiParInst	UINT	Parametersatz- oder Instanz-Nummer

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>boErr</td><td>BOOL</td><td>Der Funktionsbaustein befindet sich im Zustand Fehler
<table>
<tr><td>FALSE</td><td>Kein Fehler (Kommandierung zulässig oder Warnung)</td></tr>
<tr><td>TRUE</td><td>Fehler</td></tr>
</table></td></tr>
<tr><td>iErrID</td><td>INT</td><td>Fehler Identnummer: Diagnosenummer wird ausgegeben
<table>
<tr><td colspan="2">iErrID = 0</td><td>Kein Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = TRUE</td><td>Fehler</td></tr>
<tr><td>iErrID ≠ 0</td><td>boErr = FALSE</td><td>Warnung</td></tr>
</table></td></tr>
<tr><td>enErrName</td><td>ENUM</td><td>EN_FB_NAME
Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird.
Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.
<table>
<tr><th>Baustein</th><th>Bibliothek</th></tr>
<tr><td>WRITE_ID_DINT_TMP</td><td>AmkSystem.lib</td></tr>
</table></td></tr>
</table>

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
ID_WRITE_5IDS_TEMP_DINT	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

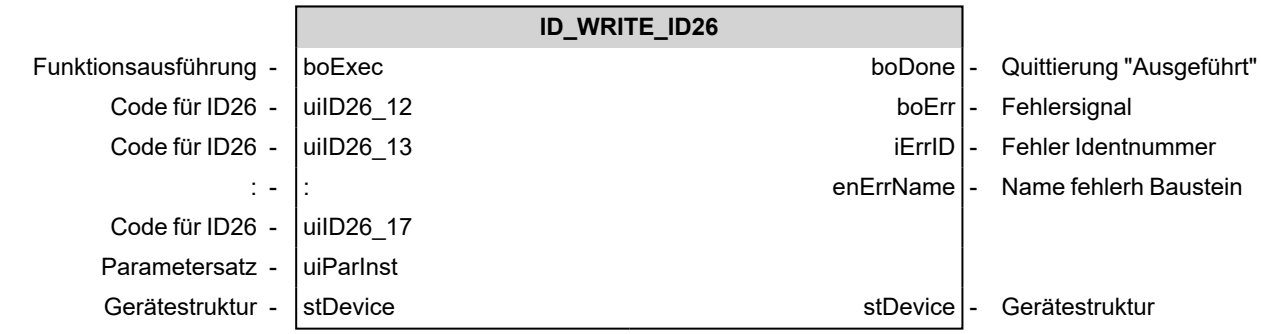
4.2.2.1.7 ID_WRITE_ID26 (FB)

Der Funktionsbaustein 'ID_WRITE_ID26' schreibt die einzelnen Elemente der ID26, 'Konfiguration Statusbits' in AMK-Geräte.
Die Listenelemente 2 – 11 sind wie folgt vorbelegt:

ID26 Element Nr.	Code	Bedeutung	
2	33036	Referenzpunkt bekannt	
3	330	$n_{ist} = n_{soll}$	innerhalb ID157 'Drehzahlfenster'
4	331	$n_{ist} < n_{min}$	innerhalb ID124 'Stillstands-Fenster'
5	332	$n_{ist} < n_x$	gem. ID125 'Drehzahlschwelle'
6	333	$M_d \geq M_{dx}$	gem. ID126 'Drehmomentschwelle'
7	334	$M_{soll} \geq M_{Grenz}$	ID83 'Drehmoment-Grenze negativ' ≤ Sollmoment ≤ ID82 'Drehmoment-Grenze positiv'
8	335	$n_{soll} \geq n_{Grenz}$	ID39 'Grenzdrehzahl negativ' ≤ Solldrehzahl ≤ ID38 'Grenzdrehzahl positiv'
9	33074	Warnung aktiv Sammelwarnung	
10		Reserviert	
11		Reserviert	

Für die Listenelemente 12 – 17 können weitere Codes frei gewählt werden. Die Codes sind der Parameterbeschreibung, Kapitel Binärausgänge zu entnehmen.
Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.
uiID26_12 ... uiID26_17	UINT	Übergabe eines frei wählbaren Statuscodes für das Element n der ID26
uiParInst	UINT	Parametersatz- oder Instanz-Nummer

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde

Name	Typ	Beschreibung
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: WRITE_ID_LIST; Bibliothek: AmkSystem.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
ID_WRITE_ID26	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.2 03_Actual Level

4.2.2.2.1 01_Status

4.2.2.2.1.1 GET_STATUSBITS (FB)

Der Funktionsbaustein 'GET_STATUSBITS' liest die Gerätestatus QUE, QRF, SBM und zusätzlich 16 frei konfigurierbare Bitmeldungen aus den verbundenen AMK-Geräten aus. Die gelesenen Statusinformationen QUE, QRF, SBM weist der Baustein seinen Ausgängen zu, die im Anwenderprogramm ausgewertet werden können. Die 16 frei konfigurierbaren Statusmeldungen werden in einer Struktur hinterlegt. Diese Struktur ist als IN_OUT-Variable zu übergeben.

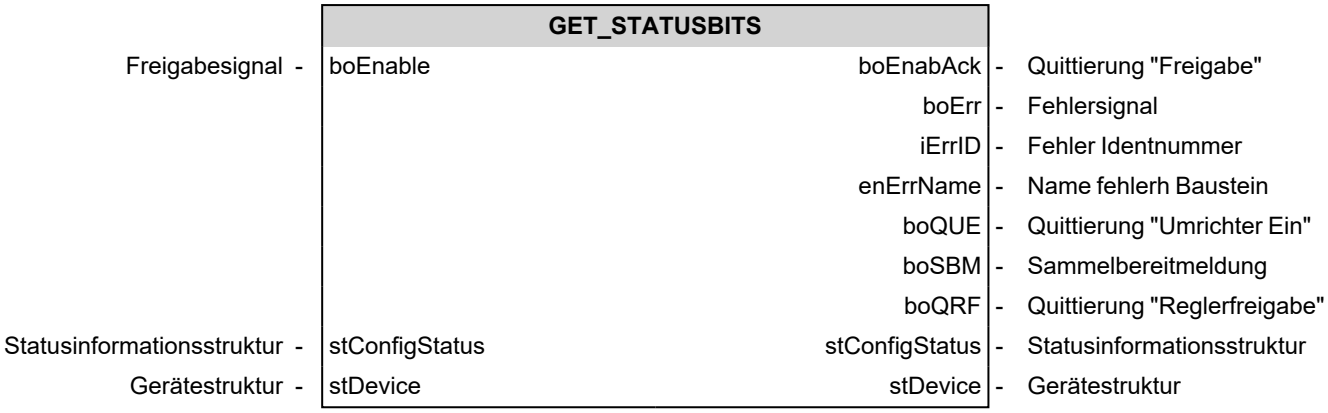
Die frei konfigurierbaren Statusbits werden im Parameter ID26, 'Konfiguration Statusbits' festgelegt. Zur Konfiguration wird der Code der gewünschten Statusmeldung in ein Element des Parameters ID26 eingetragen. Ausgelesen werden die Statusinformationen über ID144, 'Statuswort'.

Die Auswahlliste der Statusmeldungen finden Sie in der Parameterbeschreibung, Kapitel Binärausgänge.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Dieser Funktionsbaustein wird ab AFL Version 3.10 vollständig durch
'GET_STATUSBITS_ACTUAL_VALUE_GEAR'
ersetzt und ist daher für neue Anwendungen nicht mehr einzusetzen.

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. GET_STAT_DC_BUSENABLE_ACK_x_QUE – AmkDevAccess.lib GET_STAT_INVERTER_ON_ACK_x_QRF – AmkDevAccess.lib GET_STAT_SYSTEM_READY_x_SBM – AmkDevAccess.lib GET_STATUS_ID144 – AmkDevAccess.lib
boQUE	BOOL	Quittierung Umrichter Ein (QUE)
boSBM	BOOL	Systembereitmeldung (SBM)
boQRF	BOOL	Quittierung Reglerfreigabe (QRF)

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stConfigStatus	STRUCT	ST_CONFIG_STATUS_ID26 Statusinformationsstruktur boStatusID26_n Status nach ID26 Element n (n = 2 ... 17)
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.2.2 02_Actual_Value

4.2.2.2.2.1 GET_RECTANGULAR_PULSE_INPUT (FB)

Der Funktionsbaustein 'GET_RECTANGULAR_PULSE_INPUT' liest den Rechteckimpulseingang von angeschlossenen AMK-Geräten ein. ID32948, 'Meldung 4x32' muss entsprechend parametriert werden.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	GET_RECTANGULAR_PULSE_INPUT			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
		boErr	-	Fehlersignal
		iErrID	-	Fehler Identnummer
		enErrName	-	Name fehlerh Baustein
		boRefPulse	-	Referenzimpuls
		diOutVal	-	Impulsgebereingang
		diOffset	-	Verschiebung Ref.Impuls
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler

Name	Typ	Beschreibung
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0 – Kein Fehler iErrID ≠ 0 – boErr = TRUE – Fehler iErrID ≠ 0 – boErr = FALSE – Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: GeT_SETPOINT_SCR1 – Bibliothek: AmkDevAccess.lib Baustein: DI_TO_COUNT – Bibliothek: AmkBase.lib
boRefPulse	BOOL	Referenzimpuls: Impuls bei dem der aktuelle Zählerwert in den Referenzzähler eingetragen wird
diOutVal	DINT	Aktueller Zählerwert des Impulsgebereingangs [Inkremente]
diOffset	DINT	Verschiebewert [Inkremente] des Zählerwertes zum Referenzimpuls

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	GET_RECTANGULAR_PULSE_INPUT

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.2.2.2 GET_STATUSBITS_ACTUAL_VALUE_GEAR (FB)

Der Funktionsbaustein 'GET_STATUSBITS_ACTUAL_VALUE_GEAR' liest Status- und Istwertinformationen aus den angeschlossenen AMK-Geräten. Dabei wird das eingestellte Getriebeverhältnis berücksichtigt.

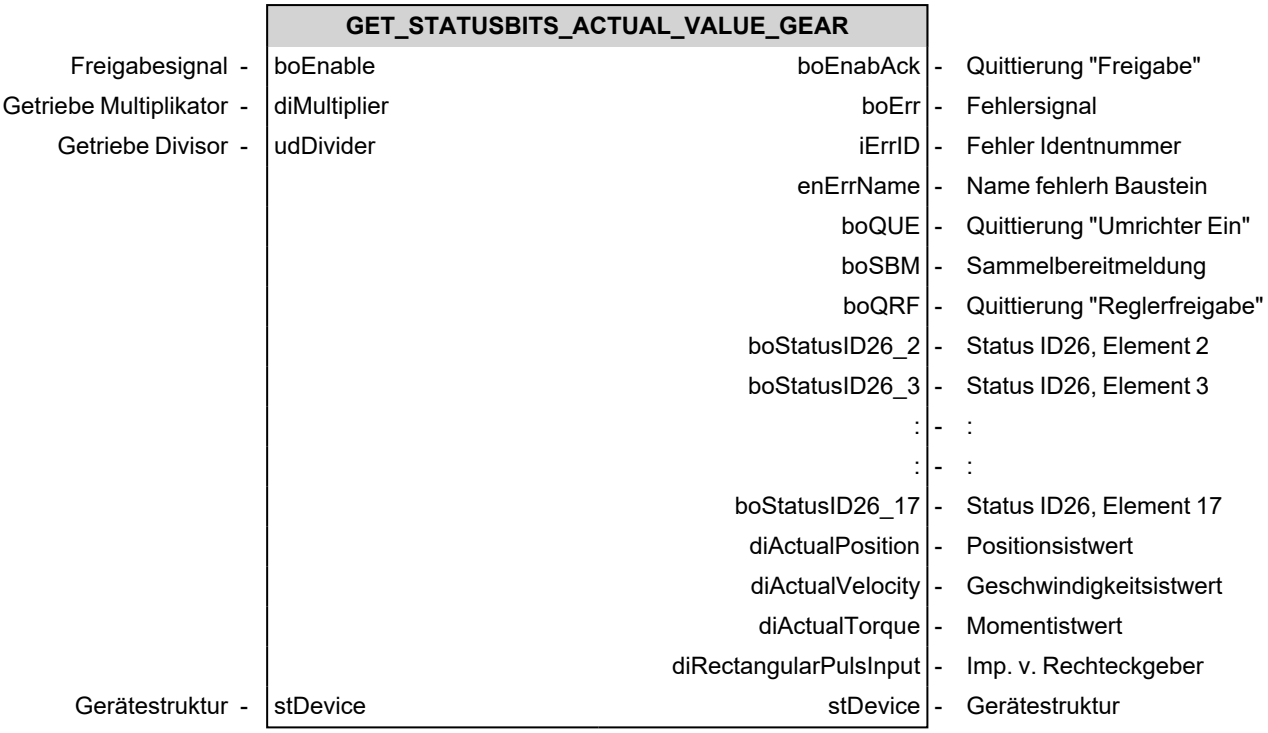
Die Statusinformation, sowie die aktuellen Werte werden vom Baustein direkt als Ausgang ausgegeben. Zusätzlich werden die 16 konfigurierbaren Statuswerte aus der Statusliste ausgelesen und stehen entsprechend ihrer Konfigurationsreihenfolge (Reihenfolge der Codes in ID26) als Ausgang zu Verfügung. Die Statuscodes sind in der Parameterbeschreibung, Kapitel Binärausgänge, beschrieben.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Dieser Funktionsbaustein ersetzt die folgenden Bausteine vollständig:

- GET_STATUSBITS
- GET_ACTUAL_VALUE
- GET_STATUSBITS_ACTUAL_VALUE

Anwender Interface



Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diMultiplier	DINT	Getriebe Multiplikator / Faktor [1]
udDivider	UDINT	Getriebe Divisor / Teiler [1]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. **Baustein** / **Bibliothek** GET_STAT_DC_BUSENABLE_ACK_x_QUE / AmkDevAccess.lib GET_STAT_INVERTER_ON_ACK_x_QRF / AmkDevAccess.lib GET_STAT_SYSTEM_READY_x_SBM / AmkDevAccess.lib GET_STATUS_ID144 / AmkDevAccess.lib GET_ACTUAL_POSITION / AmkDevAccess.lib GET_ACTUAL_SPEED / AmkDevAccess.lib GET_ACTUAL_TORQUE / AmkDevAccess.lib
boQUE	BOOL	Quittierung Umrichter Ein (QUE)
boSBM	BOOL	Systembereitmeldung (SBM)
boQRF	BOOL	Quittierung Reglerfreigabe (QRF)
boStatusID26_n	BOOL	n = 2 ... 17;
diActualPosition	DINT	Aktueller Positionsistwert [Inkremente]
diActualVelocity	DINT	Aktueller Geschwindigkeitswert [0,0001 U/min]
diActualTorque	DINT	Aktueller Momentistwert [0,1% Mn]
diOutRectangularPulsInput_X132	DINT	Aktueller Wert des Impulsgebereingangs X132 [Inkremente]

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	GET_STATUSBITS_ACTUAL_VALUE_GEAR

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.2.2.3 GET_STATUSBITS_ACTUAL_VALUE_GEAR_WITHOUT_RECT (FB)

Der Funktionsbaustein 'GET_STATUSBITS_ACTUAL_VALUE_GEAR_WITHOUT_RECT' liest Status- und Istwertinformationen aus den angeschlossenen AMK-Geräten. Dabei wird das eingestellte Getriebeverhältnis berücksichtigt.

Die Statusinformation, sowie die aktuellen Werte werden vom Baustein direkt als Ausgang ausgegeben. Zusätzlich werden die 16 konfigurierbaren Statuswerte aus der Statusliste ausgelesen und stehen entsprechend ihrer Konfigurationsreihenfolge (Reihenfolge der Codes in ID26) als Ausgang zu Verfügung. Die Statuscodes sind in der Parameterbeschreibung, Kapitel Binärausgänge, beschrieben.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

	GET_STATUSBITS_ACTUAL_VALUE_GEAR_WITHOUT_RECT			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Getriebe Multiplikator -	diMultiplier	boErr	-	Fehlersignal
Getriebe Divisor -	udDivider	iErrID	-	Fehler Identnummer
		enErrName	-	Name fehlerh Baustein
		boQUE	-	Quittierung "Umrichter Ein"
		boSBM	-	Sammelbereitmeldung
		boQRF	-	Quittierung "Reglerfreigabe"
		boStatusID26_2	-	Status ID26, Element 2
		boStatusID26_3	-	Status ID26, Element 3
		:	-	:
		:	-	:
		boStatusID26_17	-	Status ID26, Element 17
		diActualPosition	-	Positionsistwert
		diActualVelocity	-	Geschwindigkeitsistwert
		diActualTorque	-	Momentistwert
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diMultiplier	DINT	Getriebe Multiplikator / Faktor [1]
udDivider	UDINT	Getriebe Divisor / Teiler [1]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.
boQUE	BOOL	Quittierung Umrichter Ein (QUE)
boSBM	BOOL	Systembereitmeldung (SBM)
boQRF	BOOL	Quittierung Reglerfreigabe (QRF)
boStatusID26_n	BOOL	n = 2 ... 17;
diActualPosition	DINT	Aktueller Positionsistwert [Inkremente]
diActualVelocity	DINT	Aktueller Geschwindigkeitswert [0,0001 U/min]
diActualTorque	DINT	Aktueller Momentistwert [0,1% Mn]

Baustein	Bibliothek
GET_STAT_DC_BUSENABLE_ACK_x_QUE	AmkDevAccess.lib
GET_STAT_INVERTER_ON_ACK_x_QRF	AmkDevAccess.lib
GET_STAT_SYSTEM_READY_x_SBM	AmkDevAccess.lib
GET_STATUS_ID144	AmkDevAccess.lib
GET_ACTUAL_POSITION	AmkDevAccess.lib
GET_ACTUAL_SPEED	AmkDevAccess.lib
GET_ACTUAL_TORQUE	AmkDevAccess.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	GET_STATUSBITS_ACTUAL_VALUE_GEAR_WITHOUT_RECT

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.3 04_Setpoint_Level

4.2.2.3.1 01_Setpoint

4.2.2.3.1.1 SET_SET_POINTS_AND_FEED_FORWARD_GEAR (FB)

Der Funktionsbaustein 'SET_SET_POINTS_AND_FEED_FORWARD_GEAR' schreibt Steuerbits, Sollwerte und Vorsteuerwerte in AMK-Geräte. Zusätzlich wird das Lagesollwertsystem über eine Getriebestufe angepasst.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Dieser Funktionsbaustein ersetzt den folgenden Baustein vollständig:
SET_SET_POINTS_AND_FEED_FORWARD

Anwender Interface

	SET_SET_POINTS_AND_FEED_FORWARD_GEAR			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Umrichter Ein -	boUE	boErr	-	Fehlersignal
Reglerfreigabe -	boRF	iErrID	-	Fehler Identnummer
Fehler löschen -	boFL	enErrName	-	Name fehlerh Baustein
Auswahlmodus -	enMode	boSetPosEnabAck	-	Quittierung "Lagevorgabe"
Restwegfenster -	diResidDistWind	boSetVelocityEnabAck	-	Quittierung "Geschwindigkeitvorgabe"
Getriebe Multiplikator -	diMultiplier	boSetFeedForwardVelocityEnabAck	-	Quittierung "Geschwindigkeitsvorsteuerung"
Getriebe Divisor -	udDivider	boSetTorqueEnabAck	-	Quittierung "Momentvorgabe"
Geschwindigkeitssollwert -	diSetVelocity	boSetFeedForwardTorqueEnabAck	-	Quittierung "Momentvorsteuerung"
Geschw. Vorsteuerwert -	diSetFeedForwardVelocity	boResidDistDone	-	Restweg gelöscht
Momentsollwert -	diSetTorque			
Moment Vorsteuerwert -	diSetFeedForwardTorque			
Positionsistwert -	diActPosition			
Lagesollwertsystem -	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boUE	BOOL	Bit zum Einschalten des Umrichters (Zwischenkreisspannung Ein). Die entsprechende Rückmeldung QUE ist Vorbedingung für RF.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe
boFL	BOOL	Bit zum Fehlerlöschen
enMode	ENUM	EN_SETPOINT Auswahlmodus AMK_SETPOINT_NONE: Sollwertsenken deaktiviert AMK_SETPOINT_POSITION: Lagesollwertsenke aktiv AMK_SETPOINT_VELOCITY: Drehzahlsollwertsenke aktiv AMK_SETPOINT_TORQUE: Momentsollwertsenke aktiv
diResidDistWind	DINT	Das Restwegfenster gibt an, innerhalb welchen Bereiches die Achse bei RF-Wiederkehr in die Ausgangsstellung zurückzieht. Diese Einstellung ist nur bei Lagesollwertvorgabe relevant.
diMultiplier	DINT	Getriebe Multiplikator / Faktor [1]
udDivider	UDINT	Getriebe Divisor / Teiler [1]
diSetVelocity	DINT	Geschwindigkeitssollwert [0,0001 U/min]
diSetFeedForwardVelocity	DINT	Geschwindigkeitsvorsteuerwert [0,0001 U/min]
diSetTorque	DINT	Vorgabe des Momentsollwerts [0,1% Mn]

Name	Typ	Beschreibung
diSetFeedForwardTorque	DINT	Momentvorsteuerwert [0,1% Mn]
diActPosition	DINT	Aktueller Positionsistwert [Inkremente]
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.
boSetPosEnabAck	BOOL	Quittierung Freigabe der Lagevorgabe.
boSetVelocityEnabAck	BOOL	Quittierung Freigabe der Geschwindigkeitsvorgabe.
boSetFeedForwardVelocity EnabAck	BOOL	Quittierung Freigabe der Geschwindigkeitsvorsteuerung
boSetTorqueEnabAck	BOOL	Quittierung Freigabe der Momentvorgabe.
boSetFeedForwardTorque EnabAck	BOOL	Quittierung Freigabe der Momentvorsteuerung
boResidDistDone	BOOL	Restweg gelöscht.

Baustein	Bibliothek
SET_CTRL_DC_BUSENABLE_x_UE	AmkDevAccess.lib
SET_CTRL_ERR_RESET_x_FL	AmkDevAccess.lib
GET_STAT_ERR_RESET_ACK_x_QFL	AmkDevAccess.lib
SET_CTRL_INVERTER_ON_x_RF	AmkDevAccess.lib
SET_SETPOINT_POSITION	AmkDevAccess.lib
SET_SETPOINT_SPEED	AmkDevAccess.lib
SET_PRE_SETPOINT_SPEED	AmkDevAccess.lib
SET_SETPOINT_TORQUE	AmkDevAccess.lib
SET_PRE_SETPOINT_TORQUE	AmkDevAccess.lib
RATIO_INC_1	AmkBase.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	SET_SET_POINTS_AND_FEED_FORWARD_GEAR

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.3.1.2 SET_SET_POINTS_GEAR (FB)

Der Funktionsbaustein 'SET_SET_POINTS_GEAR' schreibt Steuerbits und Sollwerte in AMK-Geräte. Das Lagesollwertsystem wird über eine Getriebestufe angepasst.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Anwender Interface

		SET_SET_POINTS_GEAR			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Umrichter Ein	-	boUE	boErr	-	Fehlersignal
Reglerfreigabe	-	boRF	iErrID	-	Fehler Identnummer
Fehler löschen	-	boFL	enErrName	-	Name fehlerh Baustein
Auswahlmodus	-	enMode	boSetPosEnabAck	-	Quittierung "Lagevorgabe"
Restwegfenster	-	diResidDistWind	boSetVelocityEnabAck	-	Quittierung "Geschwindigkeitvorgabe"
Getriebe Multiplikator	-	diMultiplier	boSetTorqueEnabAck	-	Quittierung "Momentvorgabe"
Getriebe Divisor	-	udDivider	boResidDistDone	-	Restweg gelöscht
Geschwindigkeitssollwert	-	diSetVelocity			
Momentsollwert	-	diSetTorque			
Positionsistwert	-	diActPosition			
Lagesollwert	-	diSetPosition	diSetPosition	-	Lagesollwert
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boUE	BOOL	Bit zum Einschalten des Umrichters (Zwischenkreisspannung Ein). Die entsprechende Rückmeldung QUE ist Vorbedingung für RF.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe
boFL	BOOL	Bit zum Fehlerlöschen
enMode	ENUM	EN_SETPOINT Auswahlmodus AMK_SETPOINT_NONE: Sollwertsenken deaktiviert AMK_SETPOINT_VELOCITY: Drehzahlsollwertsenke aktiv AMK_SETPOINT_POSITION: Lagesollwertsenke aktiv AMK_SETPOINT_TORQUE: Momentsollwertsenke aktiv

Name	Typ	Beschreibung
diResidDistWind	DINT	Das Restwegfenster gibt an, innerhalb welchen Bereiches die Achse bei RF-Wiederkehr in die Ausgangsstellung zurückzieht. Diese Einstellung ist nur bei Lagesollwertvorgabe relevant.
diMultiplier	DINT	Getriebe Multiplikator / Faktor [1]
udDivider	UDINT	Getriebe Divisor / Teiler [1]
diSetVelocity	DINT	Geschwindigkeitssollwert [0,0001 U/min]
diSetTorque	DINT	Vorgabe des Momentsollwerts [0,1% Mn]
diActPosition	DINT	Aktueller Lageistwert [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. SET_CTRL_DC_BUSENABLE_x_UE: AmkDevAccess.lib SET_CTRL_ERR_RESET_x_FL: AmkDevAccess.lib GET_STAT_ERR_RESET_ACK_x_QFL: AmkDevAccess.lib SET_CTRL_INVERTER_ON_x_RF: AmkDevAccess.lib SET_SETPOINT_POSITION: AmkDevAccess.lib SET_SETPOINT_SPEED: AmkDevAccess.lib SET_SETPOINT_TORQUE: AmkDevAccess.lib RATIO_INC_1: AmkBase.lib
boSetPosEnabAck	BOOL	Quittierung Freigabe der Lagevorgabe.
boSetVelocityEnabAck	BOOL	Quittierung Freigabe der Geschwindigkeitsvorgabe.
boSetTorqueEnabAck	BOOL	Quittierung Freigabe der Momentvorgabe.
boResidDistDone	BOOL	Restweg gelöscht.

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	SET_SET_POINTS_GEAR

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.4 06_Diagnose

4.2.2.4.1 DIAGNOSE_ERROR (FB)

Der Funktionsbaustein 'DIAGNOSE_ERROR' liest die Fehlerinformationen aus AMK-Geräten (ID32840 'Diagnoseliste').
Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Anwender Interface

	DIAGNOSE_ERROR			
Funktionsausführung -	boExec	boDone	-	Quittierung "Ausgeführt"
		boErr	-	Fehlersignal
		iErrID	-	Fehler Identnummer
		enErrName	-	Name fehlerh Baustein
Gerätestruktur -	stDevice	stDevice	-	Gerätestruktur
Diagnosemeldung -	arstDiagnosis	arstDiagnosis	-	Diagnosemeldung

Eingangsvariablen

Name	Typ	Beschreibung
boExec	BOOL	Funktionsausführung: Mit einer positiven Flanke startet die Ausführung des Bausteins. Solange 'boExec' = TRUE ist, wird der Baustein von der PLC bearbeitet. Im Zustand 'boExec' = FALSE wird die Ausführung des Bausteins beendet.

Ausgangsvariablen

Name	Typ	Beschreibung
boDone	BOOL	Rückmeldung, dass der Funktionsbaustein vollständig ausgeführt wurde
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. Baustein: READ_ID_LIST; Bibliothek: AmkSystem.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)
arstDiagnosis	ARRAY	ARRAY [1..10] OF ST_AXIS_ERROR Diagnosemeldungen

lfd. Nr.	Diagnosemeldung	Info 1
1	Diagnose-Nr. n	
:	:	
10	Diagnose-Nr. m	

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
DIAGNOSE_ERROR	

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.4.2 DIAGNOSE_TRACE_CSV (FB)

Der Funktionsbaustein 'DIAGNOSE_TRACE_CSV' schreibt einen Wert mit 40 Einträgen in einen FIFO und speichert die Daten als CSV-Datei in der AMK-Steuerungen.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

		DIAGNOSE_TRACE_CSV			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert	-	stInVal	boEmpty	-	Puffer leer
			boOverflow	-	Pufferüberlauf
			stOutVal	-	Ausgangswert
			boWriteFileDone	-	Datei schreiben fertig
			boWriteFileErr	-	Fehler bei Datei schreiben
			iWriteFileErrID	-	Fehler-ID Datei schreiben

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
stInVal	STRUCT	ST_TRACE_CSV Eingangswert

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boEmpty	BOOL	Der FIFO-Puffer ist leer

Name	Typ	Beschreibung
boOverflow	BOOL	Pufferüberlauf: Ein neuer Wert soll in den Puffer gespeichert werden, es ist aber kein freier Speicherplatz vorhanden.
stOutVal	STRUCT	ST_TRACE_CSV Ausgangswert
boWriteFileDone	BOOL	Quittierung „Datei wurde in die Steuerung geschrieben“
boWriteFileErr	BOOL	Meldung „Fehler beim Datei schreiben“
iWriteFileErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.

Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.4.3 DIAGNOSE_TRACE_VAL1_TO_VAL10 (FB)

Der Funktionsbaustein 'DIAGNOSE_TRACE_VAL1_TO_VAL10' schreibt bis zu 10 verschiedene Werte mit der Länge von 40 Einträgen in einen FIFO und speichert die Daten als CSV-Datei in AMK-Steuerungen.

Er basiert auf dem Baustein 'DIAGNOSE_TRACE_CSV'.

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Anwender Interface

	DIAGNOSE_TRACE_VAL1_TO_VAL10			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
In CSV-Datei schreiben -	boWriteToFile	boWriteFileDone	-	Datei schreiben fertig
Eingangswert -	diValue1	boWriteFileErr	-	Fehler bei Datei schreiben
: -	:	iWriteFileErrID	-	Fehler-ID Datei schreiben
: -	:			
Eingangswert -	diValue10			

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boWriteToFile	BOOL	Mit einer positiven Flanke wird der aufgezeichnete FIFO-Inhalt in eine CSV-Datei auf die Steuerung geschrieben.
diValue1 ... diValue10	DINT	Werte, die in einem FIFO aufgezeichnet werden. Bei aktiviertem Baustein 'boEnabAck' = TRUE werden alle Eingangswerte 'diValue1' ... 'diValue10' bei Änderung eines oder mehrerer Eingangswerte in den FIFO geschrieben.

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boWriteFileDone	BOOL	Quittierung „Datei wurde in die Steuerung geschrieben“

Name	Typ	Beschreibung
boWriteFileErr	BOOL	Meldung „Fehler beim Datei schreiben“
iWriteFileErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben

Verwendungshinweis im CoDeSys-Programm

Der Aufruf dieses Funktionsbausteins ist sowohl im asynchronen Programmteil PLC_PRG als auch im synchronen Programmteil FPLC_PRG zulässig.

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.5 old version

4.2.2.5.1 GET_ACTUAL_VALUE (FB)

Der Funktionsbaustein 'GET_ACTUAL_VALUE' liest die aktuellen Istwert-Informationen

- Position
- Geschwindigkeit
- Moment

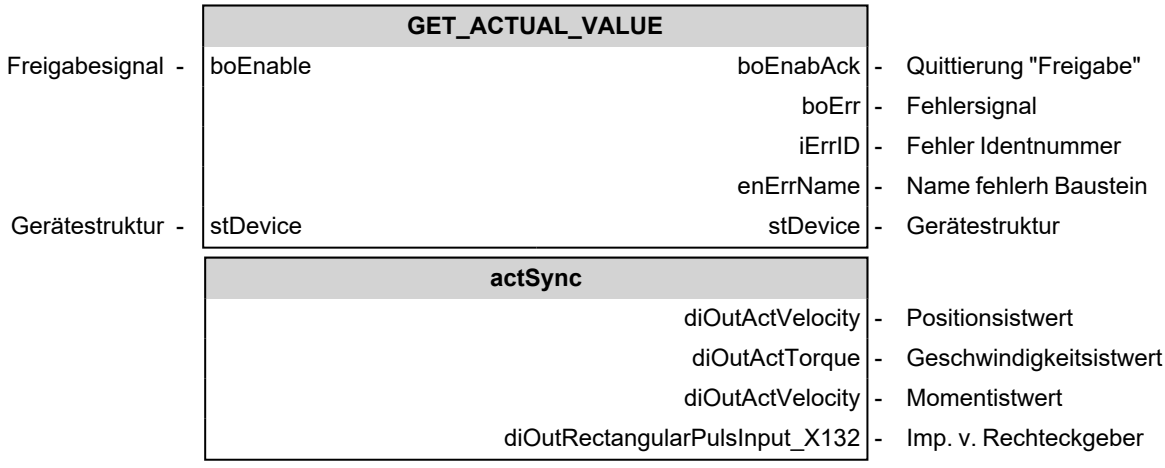
aus den angeschlossenen AMK-Geräten. Diese Werte werden den Ausgangsvariablen zugewiesen.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Dieser Funktionsbaustein wird ab AFL Version 3.10 vollständig durch
'GET_STATUSBITS_ACTUAL_VALUE_GEAR'
ersetzt und ist daher für neue Anwendungen nicht mehr einzusetzen.

Anwender Interface



Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. GET_ACTUAL_POSITION: AmkDevAccess.lib GET_ACTUAL_SPEED: AmkDevAccess.lib GET_ACTUAL_TORQUE: AmkDevAccess.lib

Ausgangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diOutActPos	DINT	Aktueller Positionsistwert [Inkremente]
diOutActVelocity	DINT	Aktueller Geschwindigkeitswert [0,0001 U/min]
diOutActTorque	DINT	Aktueller Momentistwert [0,1% Mn]
diOutRectangularPulsInput_X132	DINT	Aktueller Wert des Impulsgebereingangs X132 [Inkremente]

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
GET_ACTUAL_VALUE	GET_ACTUAL_VALUE.actSync

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.5.2 GET_STATUSBITS_ACTUAL_VALUE (FB)

Der Funktionsbaustein 'GET_STATUSBITS_ACTUAL_VALUE' liest Status- und Istwertinformationen aus den angeschlossenen AMK-Geräten.

Er fasst die Applikationsbausteine 'GET_STATUSBITS' und 'GET_ACTUAL_VALUE' zusammen.

Die Statusinformationen QUE, QRF und SBM werden vom Baustein direkt als Ausgang ausgegeben. Zusätzlich werden die 16 konfigurierbaren Statuswerte aus ID144, 'Statuswort' ausgelesen und stehen entsprechend ihrer Konfigurationsreihenfolge in einer Struktur zu Verfügung. Die Statuscodes sind in der Parameterbeschreibung, Kapitel Binärausgänge, beschrieben.

Die Istwertinformation aktuelle Position, aktuelle Geschwindigkeit und aktuelles Moment stehen ebenfalls als Ausgang zu Verfügung.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Dieser Funktionsbaustein wird ab AFL Version 3.10 vollständig durch 'GET_STATUSBITS_ACTUAL_VALUE_GEAR' ersetzt und ist daher für neue Anwendungen nicht mehr einzusetzen.

Anwender Interface

		GET_STATUSBITS_ACTUAL_VALUE			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Referenzpunkt bekannt	-	boID33036_RFP_known	boErr	-	Fehlersignal
Getriebe Multiplikator	-	diMultiplier	iErrID	-	Fehler Identnummer
Getriebe Divisor	-	udDivider	enErrName	-	Name fehlerh Baustein
			boQUE	-	Quittierung "Umrichter Ein"
			boSBM	-	Sammelbereitmeldung
			boQRF	-	Quittierung "Reglerfreigabe"
Statusinformationsstruktur	-	stConfigStatus	stConfigStatus	-	Statusinformationsstruktur
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

actSync			
diOutActPos	-	Positionsistwert	
diOutActVelocity	-	Geschwindigkeitsistwert	
diOutActTorque	-	Momentistwert	
diOutRectangularPulsInput_X132	-	Imp. v. Rechteckgeber	

Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boID33036	BOOL	Referenzpunkt bekannt
diMultiplier	DINT	Getriebe Multiplikator / Faktor [1]
udDivider	UDINT	Getriebe Divisor / Teiler [1]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben

<table>
<tr><th>Name</th><th>Typ</th><th>Beschreibung</th></tr>
<tr><td>boErr</td><td>BOOL</td><td>Der Funktionsbaustein befindet sich im Zustand Fehler
<table><tr><td>FALSE</td><td>Kein Fehler (Kommandierung zulässig oder Warnung)</td></tr><tr><td>TRUE</td><td>Fehler</td></tr></table></td></tr>
<tr><td>iErrID</td><td>INT</td><td>Fehler Identnummer: Diagnosenummer wird ausgegeben
<table><tr><td colspan="2">iErrID = 0</td><td>Kein Fehler</td></tr><tr><td>iErrID ≠ 0</td><td>boErr = TRUE</td><td>Fehler</td></tr><tr><td>iErrID ≠ 0</td><td>boErr = FALSE</td><td>Warnung</td></tr></table></td></tr>
<tr><td>enErrName</td><td>ENUM</td><td>EN_FB_NAME
Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird.
Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.
<table><tr><th>Baustein</th><th>Bibliothek</th></tr><tr><td>GET_STAT_DC_BUSENABLE_ACK_x_QUE</td><td>AmkDevAccess.lib</td></tr><tr><td>GET_STAT_INVERTER_ON_ACK_x_QRF</td><td>AmkDevAccess.lib</td></tr><tr><td>GET_STAT_SYSTEM_READY_x_SBM</td><td>AmkDevAccess.lib</td></tr><tr><td>GET_STATUS_ID144</td><td>AmkDevAccess.lib</td></tr><tr><td>GET_ACTUAL_POSITION</td><td>AmkDevAccess.lib</td></tr><tr><td>GET_ACTUAL_SPEED</td><td>AmkDevAccess.lib</td></tr><tr><td>GET_ACTUAL_TORQUE</td><td>AmkDevAccess.lib</td></tr></table></td></tr>
</table>

Ausgangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diOutActPos	DINT	Aktueller Positionsistwert [Inkremente]
diOutActVelocity	DINT	Aktueller Geschwindigkeitswert [0,0001 U/min]
diOutActTorque	DINT	Aktueller Momentistwert [0,1% Mn]
diOutRectangularPulsInput_X132	DINT	Aktueller Wert des Impulsgebereingangs X132 [Inkremente]

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stConfigStatus	STRUCT	ST_CONFIG_STATUS_ID26 Statusinformationsstruktur boStatusID26_n Status nach ID26 Element n (n = 2 ... 17)
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
GET_STATUSBITS_ACTUAL_VALUE	GET_STATUSBITS_ACTUAL_VALUE.actSync

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.5.3 SET_CONTROLBITS_SETPOINT_SETVALUES (FB)

Der Funktionsbaustein 'SET_CONTROLBITS_SETPOINT_SETVALUES' schreibt Steuerbits und Sollwerte in AMK-Geräte. Er bildet das Lagesollwertsystem in der Steuerung.

- Es können Positions-, Drehzahl- und Momentsollwerte durch diesen Baustein in den Antrieb geschrieben werden.
- Die Auswahl, welcher Sollwert in den Antrieb geschrieben wird, erfolgt am Modus-Eingang des Bausteins.
  Dabei sind die einzelnen Sollwerte bausteinintern gegeneinander verriegelt. Das heißt: Positions-, Drehzahl-, und Momentsollwerte können nur getrennt und unabhängig voneinander in den Antrieb geschrieben werden.
- Wird kein Sollwertmodus übergeben, sind die Sollwertsenken der SPS deaktiviert. Der aktive Baustein schaltet den Antrieb in die Nebenbetriebsart 2 (Drehzahlregelung), somit ist sichergestellt, dass beim Schalten der Reglerfreigabe keine Ausgleichsbewegung erfolgt.
- Für das Einschalten der Reglerfreigabe RF ist zuvor das Setzen des Steuerbits Umrichter Ein UE erforderlich. Ist UE gesetzt, kann durch das Steuerbit RF die Reglerfreigabe eingeschaltet werden.
- Bevor der Baustein das RF-Steuerbit an den Antrieb schreibt, wird die Lage aufgesetzt. Damit wird eine sprungartige Bewegung der Achse verhindert.
- Soll die Achse beim Einschalten von RF in Ihre Ausgangsposition zurückziehen, kann dies durch die Festlegung eines Fensters definiert werden. Diese Funktionalität ist nur in Lageregelung möglich.
- Liegt die Einschaltposition der Achse außerhalb des definierten Fensters, wird der Restweg gelöscht und die Achse wird ohne Bewegung bestromt. Die Löschung des Restweges wird mit einer Ausgangsmeldung angezeigt.
- Durch setzen des Fehlerlöschbits FL werden Antriebsfehler gelöscht.
- Als Ausgangsbits liefert der Baustein Rückmeldungen über die aktive Sollwertsenke.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Dieser Funktionsbaustein wird ab AFL Version 3.10 vollständig durch
'SET_SET_POINTS_GEAR'
ersetzt und ist daher für neue Anwendungen nicht mehr einzusetzen.

Anwender Interface

		SET_CONTROLBITS_SETPOINT_SETVALUES			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Umrichter Ein	-	boUE	boErr	-	Fehlersignal
Reglerfreigabe	-	boRF	iErrID	-	Fehler Identnummer
Quittierung "Reglerfreigabe"	-	boQRF	enErrName	-	Name fehlerh Baustein
Fehler löschen	-	boFL	boSetPosEnabAck	-	Quittierung "Lagevorgabe"
Auswahlmodus	-	enMode	boSetVelocityEnabAck	-	Quittierung "Geschwindigkeitvorgabe"
Restwegfenster	-	diResidDistWind	boSetTorqueEnabAck	-	Quittierung "Momentvorgabe"
			boResidDistDone	-	Restweg gelöscht
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

		actSync			
Geschwindigkeitssollwert	-	diSetVelocity			
Momentsollwert	-	diSetTorque			
Positionsistwert	-	diActPosition			
Lagesollwertsystem	-	diSetPosition	diSetPosition	-	Lagesollwert

Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boUE	BOOL	Bit zum Einschalten des Umrichters (Zwischenkreisspannung Ein). Die entsprechende Rückmeldung QUE ist Vorbedingung für RF.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe
boQRF	BOOL	Quittierung Reglerfreigabe (QRF)
boFL	BOOL	Bit zum Fehlerlöschen
enMode	ENUM	EN_SETPOINT Auswahlmodus AMK_SETPOINT_NONE: Sollwertsenken deaktiviert AMK_SETPOINT_POSITION: Lagesollwertsenke aktiv AMK_SETPOINT_VELOCITY: Drehzahlsollwertsenke aktiv AMK_SETPOINT_TORQUE: Momentsollwertsenke aktiv
diResidDistWind	DINT	Das Restwegfenster gibt an, innerhalb welchen Bereiches die Achse bei RF-Wiederkehr in die Ausgangsstellung zurückzieht. Diese Einstellung ist nur bei Lagesollwertvorgabe relevant.

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diSetVelocity	DINT	Geschwindigkeitssollwert [0,0001 U/min]
diSetTorque	DINT	Vorgabe des Momentsollwerts [0,1% Mn]
diActPosition	DINT	Aktueller Positionsistwert [Inkremente]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung

Name	Typ	Beschreibung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt.
boSetPosEnabAck	BOOL	Quittierung Freigabe der Lagevorgabe.
boSetVelocityEnabAck	BOOL	Quittierung Freigabe der Geschwindigkeitsvorgabe.
boSetTorqueEnabAck	BOOL	Quittierung Freigabe der Momentvorgabe.
boResidDistDone	BOOL	Restweg gelöscht.

Baustein	Bibliothek
SET_CTRL_DC_BUSENABLE_x_UE	AmkDevAccess.lib
SET_CTRL_ERR_RESET_x_FL	AmkDevAccess.lib
GET_STAT_ERR_RESET_ACK_x_QFL	AmkDevAccess.lib
SET_CTRL_INVERTER_ON_x_RF	AmkDevAccess.lib
SET_SETPOINT_POSITION	AmkDevAccess.lib
SET_SETPOINT_SPEED	AmkDevAccess.lib
SET_SETPOINT_TORQUE	AmkDevAccess.lib

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
SET_CONTROLBITS_SETPOINT_SETVALUES	SET_CONTROLBITS_SETPOINT_SETVALUES.actSync

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.5.4 SET_SET_POINTS_AND_FEED_FORWARD (FB)

Der Funktionsbaustein 'SET_SET_POINTS_AND_FEED_FORWARD' schreibt Steuerbits, Sollwerte und Vorsteuerwerte in AMK-Geräte. Er bildet die Funktionalität des FB 'SET_CONTROLBITS_SETPOINT_SETVALUES' ab und erweitert diese um die Vorgabe von Drehzahl- und Momentvorsteuerwerten.

Die Momentvorsteuerwerte sind in allen Betriebsarten (Lageregelung, Drehzahlregelung, Momentsteuerung) vorgebbar.

Drehzahlvorsteuerwerte könne in den Betriebsarten Lage- und Drehzahlregelung vorgegeben werden.

Dabei sind die Vorsteuerkanäle immer aktiv, d.h. wird ein Wert > 0 an den entsprechenden Vorsteuereingang geschrieben, wird dieser direkt an den Antrieb weitergegeben. Ein separates Aktivieren der Vorsteuervorgabe entfällt.

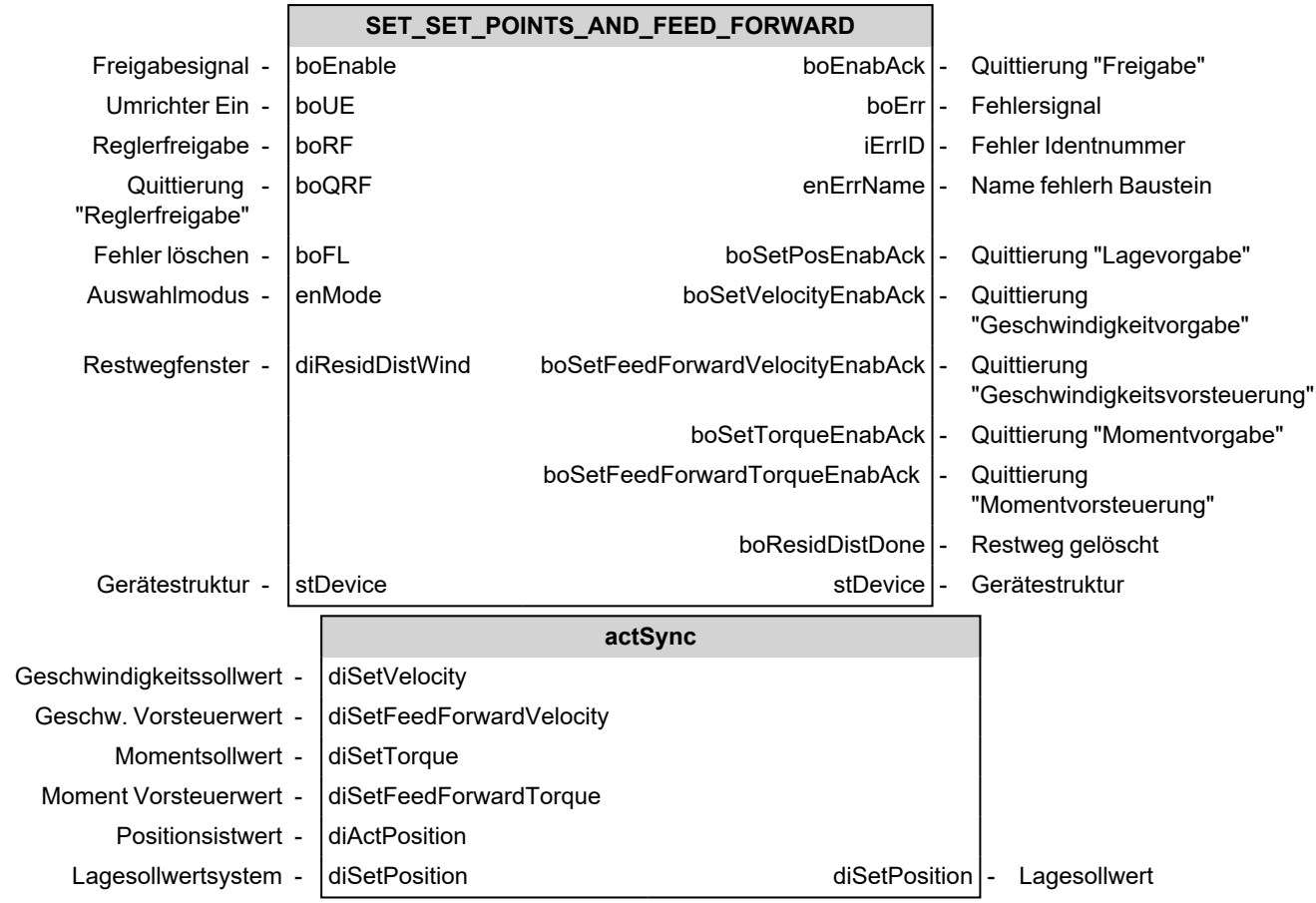
Durch Ausgangsvariablen wird die Aktivierung der Vorsteuerung angezeigt.

Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

Soll- und Istwerte werden in einer synchronen Aktion übergeben. Die synchrone Aktion muss in der synchronen Programmebene FPLC_ PRG aufgerufen werden.

Dieser Funktionsbaustein wird ab AFL Version 3.10 vollständig durch
'SET_SET_POINTS_AND_FEED_FORWARD_GEAR'
ersetzt und ist daher für neue Anwendungen nicht mehr einzusetzen.

Anwender Interface



Eingangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
boUE	BOOL	Bit zum Einschalten des Umrichters (Zwischenkreisspannung Ein). Die entsprechende Rückmeldung QUE ist Vorbedingung für RF.
boRF	BOOL	Bit zum Einschalten der Reglerfreigabe
boQRF	BOOL	Quittierung Reglerfreigabe (QRF)
boFL	BOOL	Bit zum Fehlerlöschen
enMode	ENUM	EN_SETPOINT Auswahlmodus AMK_SETPOINT_NONE: Sollwertsenken deaktiviert AMK_SETPOINT_POSITION: Lagesollwertsenke aktiv AMK_SETPOINT_VELOCITY: Drehzahlsollwertsenke aktiv AMK_SETPOINT_TORQUE: Momentsollwertsenke aktiv
diResidDistWind	DINT	Das Restwegfenster gibt an, innerhalb welchen Bereiches die Achse bei RF-Wiederkehr in die Ausgangsstellung zurückzieht. Diese Einstellung ist nur bei Lagesollwertvorgabe relevant.

Eingangsvariablen der synchronen Programmebene (FPLC_PRG)

Name	Typ	Beschreibung
diSetVelocity	DINT	Geschwindigkeitssollwert [0,0001 U/min]
diSetFeedForwardVelocity	DINT	Geschwindigkeitsvorsteuerwert [0,0001 U/min]
diSetTorque	DINT	Vorgabe des Momentsollwerts [0,1% Mn]
diSetFeedForwardTorque	DINT	Momentvorsteuerwert [0,1% Mn]
diActPosition	DINT	Aktueller Positionsistwert [Inkremente]

Ausgangsvariablen der asynchronen Programmebene (PLC_PRG)

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
enErrName	ENUM	EN_FB_NAME Name des fehlerhaften Bausteins, für den die Diagnosenummer 'iErrID' ausgegeben wird. Die Diagnosenummer ist in der Beschreibung der entsprechenden Bibliothek erklärt. SET_CTRL_DC_BUSENABLE_x_UE – AmkDevAccess.lib SET_CTRL_ERR_RESET_x_FL – AmkDevAccess.lib GET_STAT_ERR_RESET_ACK_x_QFL – AmkDevAccess.lib SET_CTRL_INVERTER_ON_x_RF – AmkDevAccess.lib SET_SETPOINT_POSITION – AmkDevAccess.lib SET_SETPOINT_SPEED – AmkDevAccess.lib SET_PRE_SETPOINT_SPEED – AmkDevAccess.lib SET_SETPOINT_TORQUE – AmkDevAccess.lib SET_PRE_SETPOINT_TORQUE – AmkDevAccess.lib
boSetPosEnabAck	BOOL	Quittierung Freigabe der Lagevorgabe.
boSetVelocityEnabAck	BOOL	Quittierung Freigabe der Geschwindigkeitsvorgabe.
boSetFeedForwardVelocity EnabAck	BOOL	Quittierung Freigabe der Geschwindigkeitsvorsteuerung
boSetTorqueEnabAck	BOOL	Quittierung Freigabe der Momentvorgabe.
boSetFeedForwardTorque EnabAck	BOOL	Quittierung Freigabe der Momentvorsteuerung
boResidDistDone	BOOL	Restweg gelöscht.

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
SET_SET_POINTS_AND_FEED_FORWARD	SET_SET_POINTS_AND_FEED_FORWARD.actSync

Zu diesem Funktionsbaustein ist im CoDeSys eine Visualisierung vorbereitet.
Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

4.2.2.5.5 SET_SETPOINT_POSITION_ABSOLUTE (FB)

Der Funktionsbaustein 'SET_SETPOINT_POSITION_ABSOLUTE' schreibt einen absoluten Positionssollwert in die ID47, 'Lage-Sollwert' des Antriebs.
Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

Dieser Funktionsbaustein wird ab AFL Version 3.10 vollständig durch
'SET_SETPOINT_POSITION'
ersetzt und ist daher für neue Anwendungen nicht mehr einzusetzen.

Anwender Interface

		SET_SETPOINT_POSITION_ABSOLUTE			
Freigabesignal	-	boEnable	boEnabAck	-	Quittierung "Freigabe"
Lagesollwert	-	diSetPosition	boErr	-	Fehlersignal
			iErrID	-	Fehler Identnummer
Gerätestruktur	-	stDevice	stDevice	-	Gerätestruktur

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diSetPosition	DINT	Vorgabe des Lagesollwertes (Lagesollwertsystem) [Inkremente]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler

Name	Typ	Beschreibung
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0 – Kein Fehler iErrID ≠ 0 – boErr = TRUE – Fehler iErrID ≠ 0 – boErr = FALSE – Warnung

Ein- und Ausgangsvariablen

Name	Typ	Beschreibung
stDevice	STRUCT	ST_DEVICE Die Gerätebeschreibungsstruktur weist den Baustein einem Gerät zu. (Siehe Dokument Softwarebeschreibung AmkBase Bibliothek, Teile-Nr. 204986)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	SET_SETPOINT_POSITION_ABSOLUTE

4.2.2.6 Support

4.2.2.6.1 CONTROL_UE_RF (FB)

Der Funktionsbaustein 'CONTROL_UE_RF' schaltet die Ausgänge für UE und RF entsprechend den 4 Eingängen.
Der Funktionsbaustein wird in der asynchronen Programmebene PLC_PRG aufgerufen.

'Support'-Funktionen und -Funktionsbausteine werden nicht direkt angewendet, sondern von anderen Funktionen und Funktionsbausteinen aufgerufen.

Anwender Interface

	CONTROL_UE_RF		
Umrichter EIN -	boInput_UE	boQutput_UE	- Ausgang Umrichter EIN
Reglerfreigabe -	boInput_RF	boOutput_RF	- Ausgang Reglerfreigabe
Quittierung Umrichter EIN -	boInput_QRF		
Quittierung Reglerfreigabe -	boInput_QUE		

Eingangsvariablen

Name	Typ	Beschreibung
boInput_UE	BOOL	Eingang für 'Umrichter EIN'
boInput_RF	BOOL	Eingang für 'Reglerfreigabe'
boInput_QRF	BOOL	Eingang für 'Quittierung Reglerfreigabe'
boInput_QUE	BOOL	Eingang für 'Quittierung Umrichter EIN'

Ausgangsvariablen

Name	Typ	Beschreibung
boOutput_UE	BOOL	Ausgang Umrichter EIN (boInput_UE = TRUE + Wartezeit 150 ms => boOutput_UE = TRUE)
boOutput_RF	BOOL	Ausgang Reglerfreigabe (boInput_UE = TRUE AND boInput_QUE = TRUE AND boInput_RF = TRUE => boOutput_RF = TRUE)

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
CONTROL_UE_RF	

4.2.2.6.2 RATIO_INC_ABS (FB)

Der Funktionsbaustein 'RATIO_INC_ABS' ist ein absolut zählendes Getriebe, bei dem das Verhältnis zwischen Eingangs- und Ausgangsinkrementen beliebig eingestellt werden kann.

Der Funktionsbaustein wird in der synchronen Programmebene FPLC_PRG aufgerufen.

'Support'-Funktionen und -Funktionsbausteine werden nicht direkt angewendet, sondern von anderen Funktionen und Funktionsbausteinen aufgerufen.

Anwender Interface

	RATIO_INC_ABS			
Freigabesignal -	boEnable	boEnabAck	-	Quittierung "Freigabe"
Eingangswert -	diInVal	boErr	-	Fehlersignal
Getriebe Multiplikator -	diMultiplier	iErrID	-	Fehler Identnummer
Getriebe Divisor -	udDivider	diOutVal	-	Ausgangswert

Eingangsvariablen

Name	Typ	Beschreibung
boEnable	BOOL	Freigabesignal: Mit einer positiven Flanke startet die Initialisierung des Bausteins. Solange 'boEnable' und der Ausgang 'boEnabAck' = TRUE ist, ist der Baustein freigegeben und wird von der PLC bearbeitet. Im Zustand 'boEnable' = FALSE wird dem Baustein die Freigabe entzogen und er wird nicht mehr abgearbeitet.
diInVal	DINT	Eingangswert
diMultiplier	DINT	Getriebe Multiplikator / Faktor [1]
udDivider	UDINT	Getriebe Divisor / Teiler [1]

Ausgangsvariablen

Name	Typ	Beschreibung
boEnabAck	BOOL	Quittierung: Funktionsbaustein ist initialisiert und freigegeben
boErr	BOOL	Der Funktionsbaustein befindet sich im Zustand Fehler FALSE: Kein Fehler (Kommandierung zulässig oder Warnung) TRUE: Fehler
iErrID	INT	Fehler Identnummer: Diagnosenummer wird ausgegeben iErrID = 0: Kein Fehler iErrID ≠ 0, boErr = TRUE: Fehler iErrID ≠ 0, boErr = FALSE: Warnung
diOutVal	DINT	Ausgangswert

Verwendungshinweis im CoDeSys-Programm

PLC_PRG (asynchroner Programmteil)	FPLC_PRG (synchroner Programmteil)
	RATIO_INC_ABS

4.3 FIRST STEPS AFL Standardbausteine

4.3.1 FIRST STEPS mit CODESYS V3 und AFL Standard Bausteine

'FIRST STEPS' beschreibt, wie Sie:

- AIPEX PRO 'Basis Einstellungen' vornehmen
- ein neues PLC Projekt mit AIPEX PRO erzeugen
- AFL Standard Bausteine für Steuerung, Einspeisung und Achsen importieren und instanzieren
- der Zugriff auf die Achsstukturen (Ist- und Sollwerte)
- und das PLC Programm übertragen

Das Kapitel 'Funktionen' enthält weitere Beispiele mit den Themen:

- EA Zugriff
- Erweiterte Variablenzugriffe
- Kommunikation zwischen PLC und PLC
- Kurvenscheiben
- PLC Projekt importieren
- Steuerungsvarianten
- Visualisierung

Das Kapitel 'Allgemein' enthält grundsätzliche Informationen wie:

- Automatische Buskonfiguration
- Bibliotheksverwalter
- Default Taskkonfiguration
- Diagnoseinformationen
- Gerätebeschreibungen
- Steuerungskonfiguration

4.3.1.1 Voraussetzungen

Hardware / Firmware	Software / AFL 4 Version
Hardware: AMKAMAC A4 Steuerung Firmware: ≥ V4.21 2017/08 Teile-Nr. 206755	Software: ≥ AIPEX PRO Version 3.03 2015/41 Teile-Nr.205890 AFL Library: ≥ AFL V4 Version 3.5.5.0 2015/41 Teile-Nr. 206004
Hardware: AMKAMAC A5 Steuerung Firmware: ≥ A5_411_1350 Teile-Nr. 204756	Software: ≥ AIPEX PRO Version 3.00 2013/50 Teile-Nr. 204905 AFL Library: ≥ AFL V4 Version 3.5.3.0 2014/06 Teile-Nr. 204786
Hardware: AMKAMAC A6 Steuerung Firmware: ≥ A6_V4.11_2014/24 Teile-Nr. 204760	
Hardware: AMKASMART iSA Steuerung Firmware: ≥ V4.20 2015/26 Teile-Nr. 205729	Software: ≥ AIPEX PRO Version 3.03 2015/41 Teile-Nr.205890 AFL Library: ≥ AFL V4 Version 3.5.5.0 2015/41 Teile-Nr. 206004

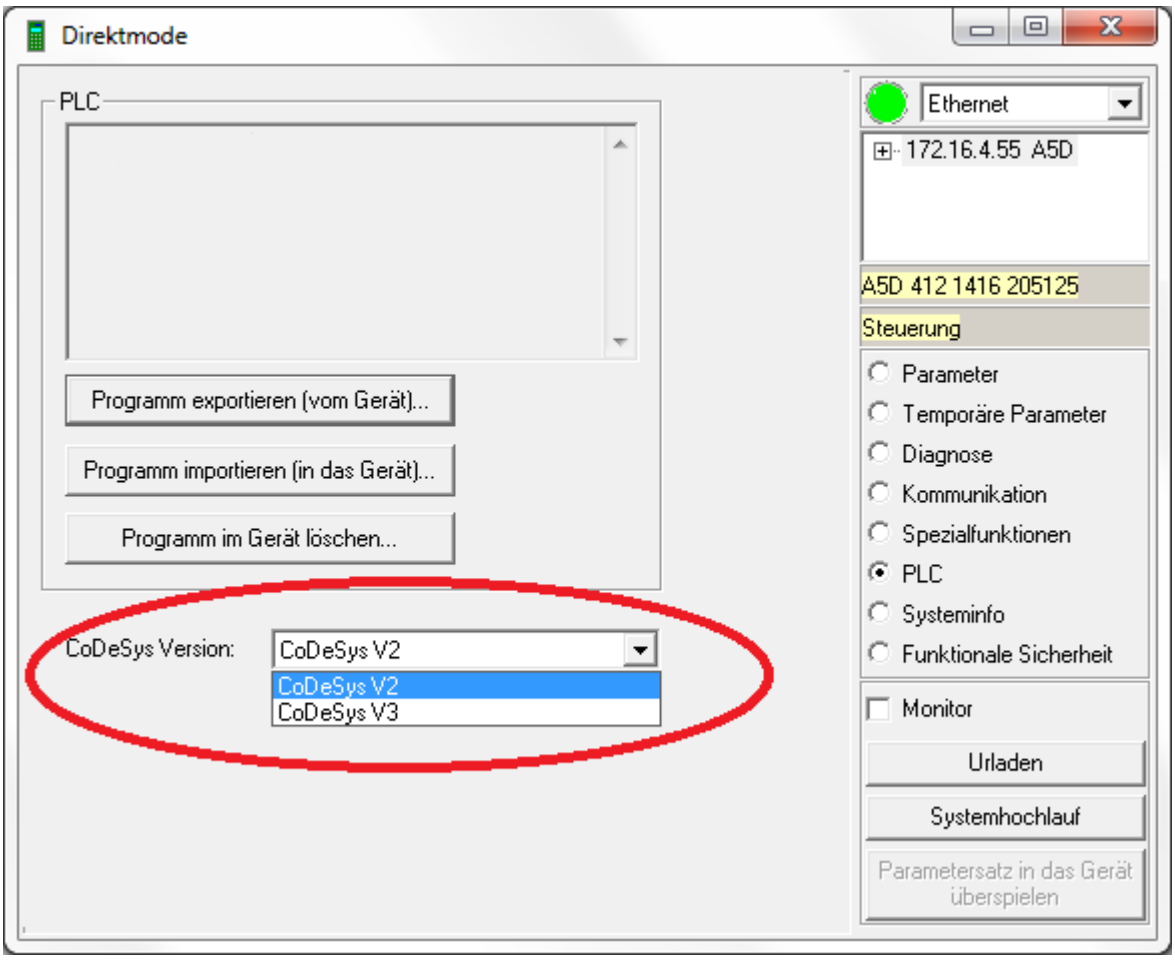
AMK A5 Steuerungen mit der Firmware ≥ A5 V4.11 2013/50 AMK Teile-Nr. 204756 können mit CODESYS V3 oder CODESYS V2 programmiert werden. Die Standardeinstellung ist CODESYS V2. Umstellung auf V3:
Siehe 'CODESYS Versionsauswahl' auf Seite 669.

4.3.1.2 CODESYS Versionsauswahl

Das PLC Anwenderprogramm wird mit dem AMK Engineeringtool AIPEX PRO erstellt. AIPEX PRO enthält die IEC 61131-3 Programmierplattform CODESYS der Firma 3S Smart Software Solutions GmbH. Die Steuerungen können mit CODESYS V2 bzw. mit CODESYS V3 programmiert werden. Für beide Versionen bietet AMK leistungsstarke Motion Control und Technologiefunktionen an.

Die Festlegung, ob eine Steuerung mit CODESYS V3 oder CODESYS V2 betrieben wird, kann mit dem AIPEX PRO 'Direktmode' eingestellt werden.

Öffnen Sie den Direktmode. Siehe 'Direktmode' auf Seite 127.



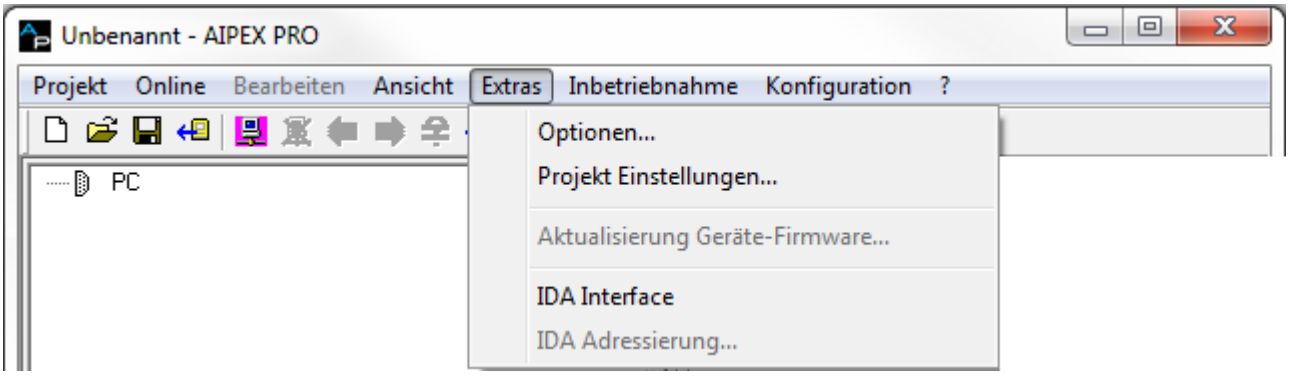
Führen Sie einen Neustart durch um die Änderung zu aktivieren.

Direkter Zugriff über Parameter
ID34175 'Steuerungseinstellungen' Bit 4 = 1 CODESYS V3
ID34175 'Steuerungseinstellungen' Bit 4 = 0 CODESYS V2

Statusleiste Displaysteuerung
Die aktive CODESYS Version wird bei Displaysteuerungen während des Systemhochlaufs in der Statusleiste angezeigt.

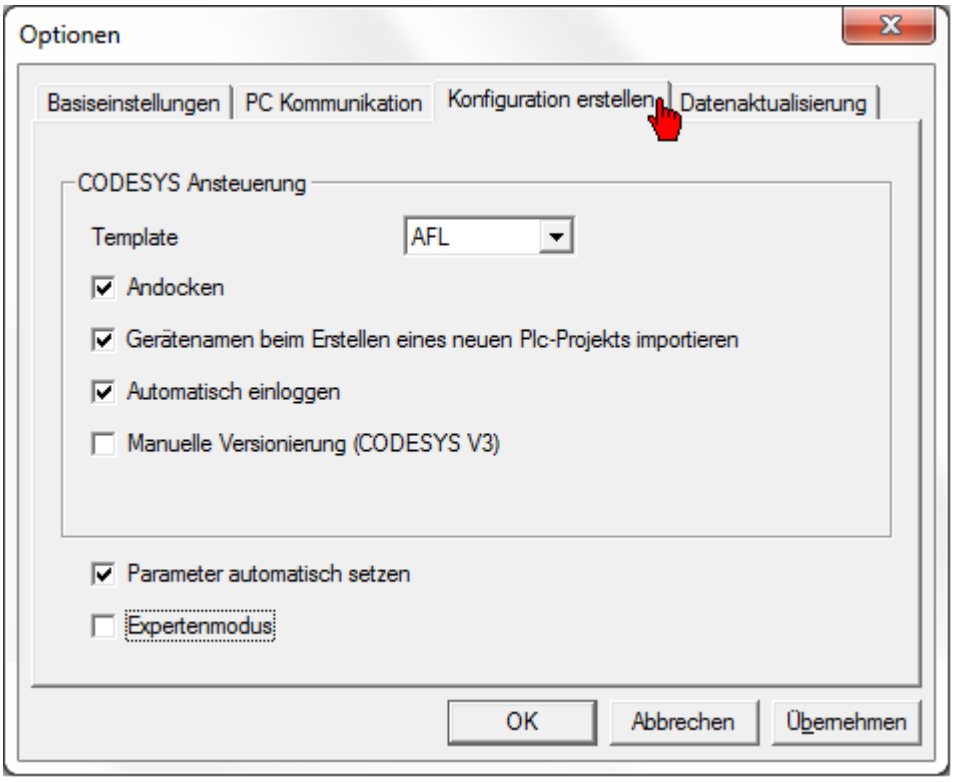
4.3.1.3 Grund Einstellungen AIPEX PRO

Öffnen Sie das Menü **'Extras'** → **'Optionen'**



Nehmen Sie folgende Grund Einstellungen vor

Das Template muss auf 'AFL' stehen, bevor das CODESYS Projekt erzeugt wird.

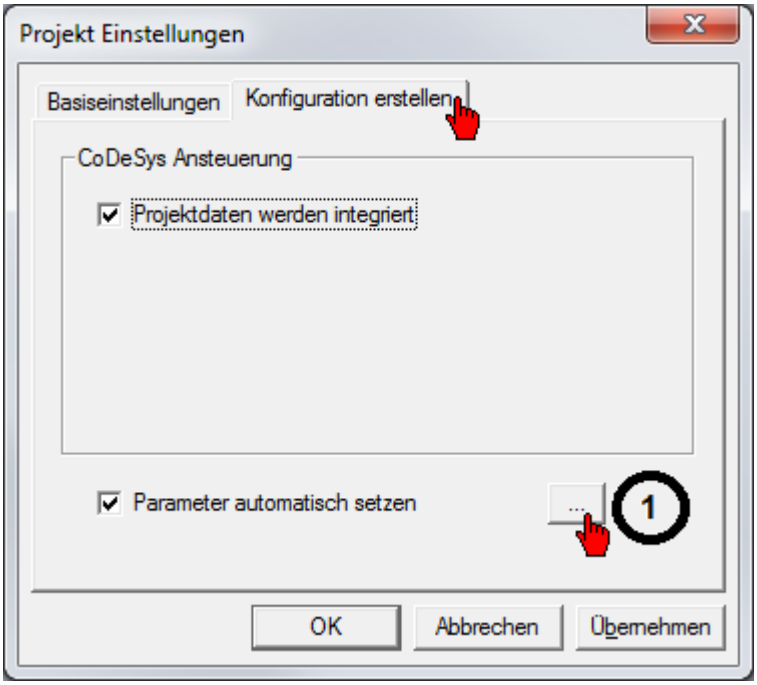


Beschreibung der Menüpunkte:

Siehe 'Konfiguration erstellen' auf Seite 107.

Öffnen Sie das Menü **'Extras'** → **'Projekt Einstellungen'** → **'Konfiguration erstellen'**

Nehmen Sie folgende Projekt Einstellungen vor. Die Projekt Einstellungen können vor jedem generieren geändert werden.



Beschreibung der Menüpunkte:

Siehe 'Konfiguration erstellen (projektspezifisch)' auf Seite 113.

4.3.1.4 PLC Projekt erzeugen (Online Projekt)

Das Kapitel beschreibt, wie Sie ein neues PLC Projekt erstellen. Im Beispiel ist AIPEX PRO mit einem Demomodell verbunden. Die Geräteeigenschaften werden direkt aus dem Demomodel übernommen.

Starten Sie AIPEX PRO.

Voraussetzung:

'Kommunikations-/Icon': Status 'Grün' (Schnittstelle breit)

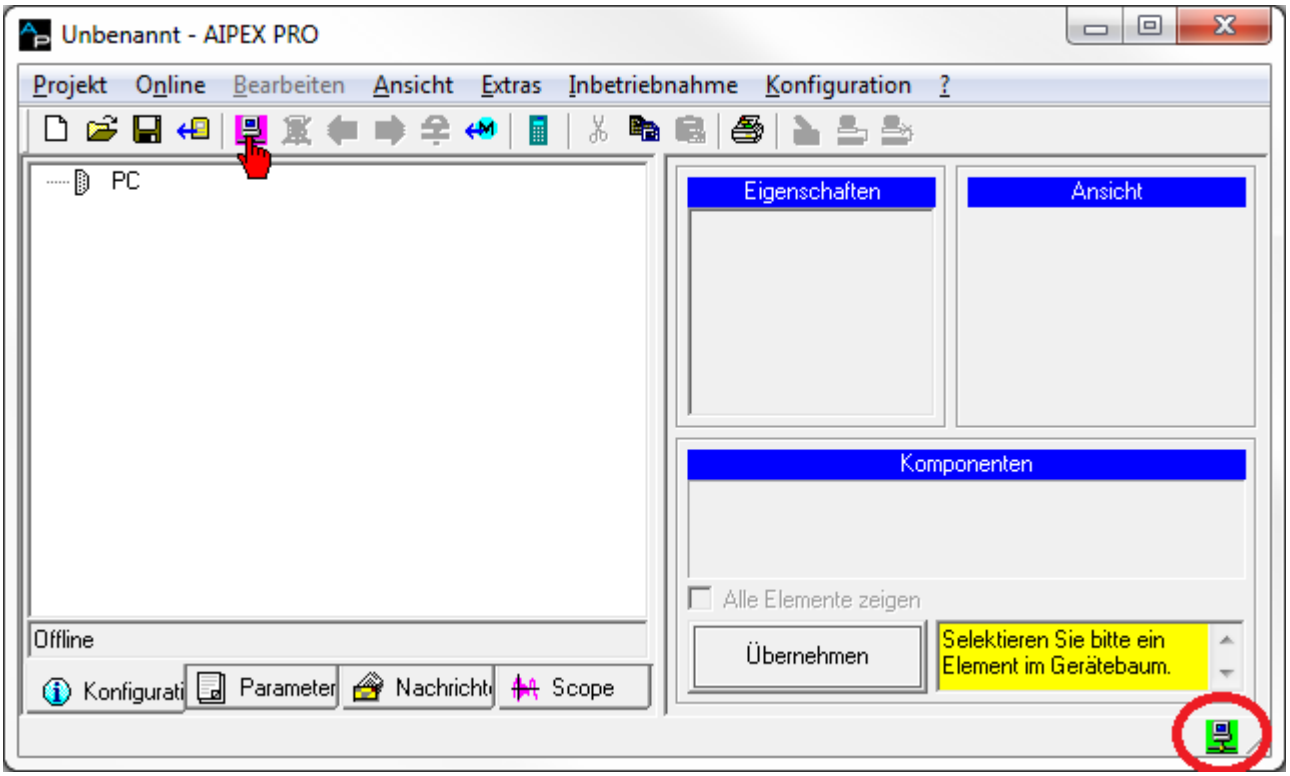
Icon: Status 'Rot' bzw. kein Icon vorhanden, prüfen Sie folgende Punkte:

Erstinbetriebnahme Feldbus abgeschlossen:
Siehe Dokument Erstinbetriebnahme KE/KW (AMK Teile-Nr. 204539) Erstinbetriebnahme Feldbus bzw. Dokument Erstinbetriebnahme iC/iX/iDT5 (AMK Teile-Nr. 204737) Erstinbetriebnahme Feldbus

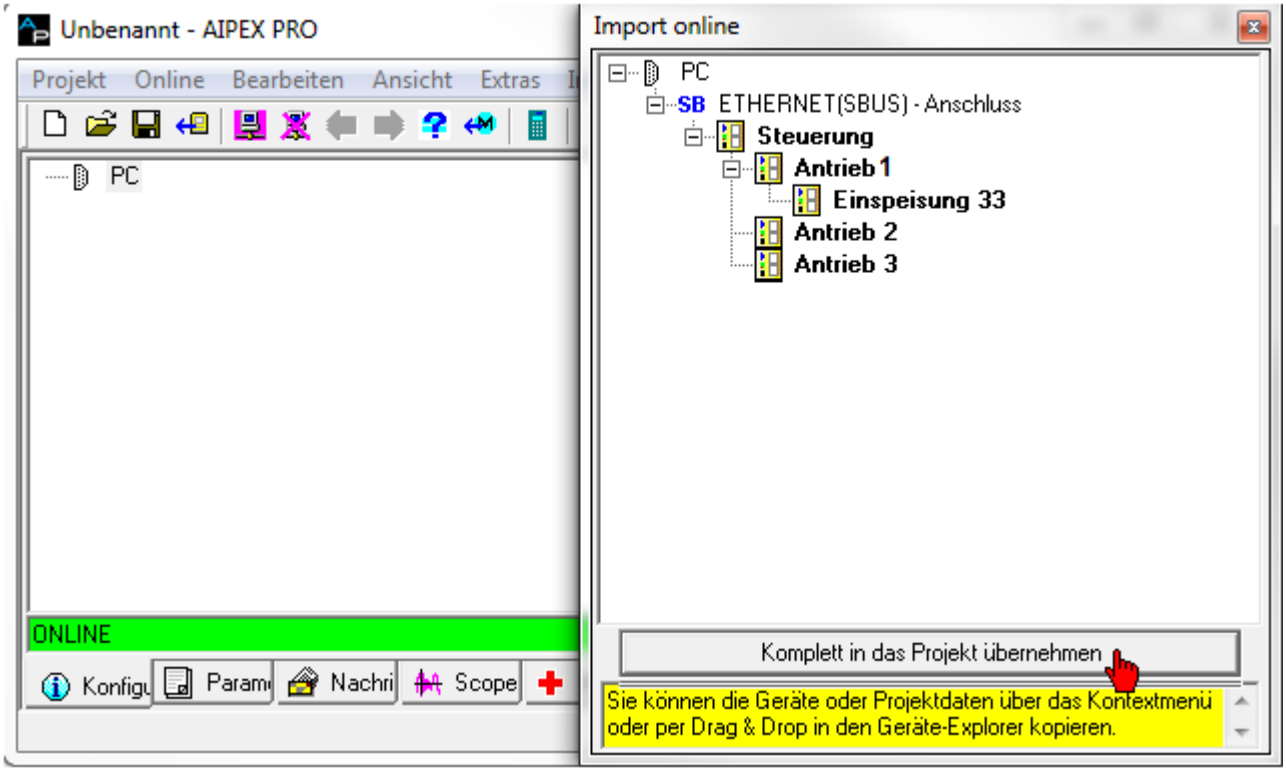
Direktverbindung über Ethernet aktiv?:
Siehe Dokument Gerätebeschreibung Steuerungen A-Serie (AMK Teile-Nr. 202975) Kapitel Direktverbindung über Ethernet.

Bevor das AIPEX PRO Projekt erstellt wird, muss die CODESYS Version eingestellt werden.
CODESYS Versionsumstellung: Siehe 'CODESYS Versionsauswahl' auf Seite 669.

Drücken Sie den Taster **'Einloggen'** um eine Verbindung mit den verbundenen Geräten herzustellen.

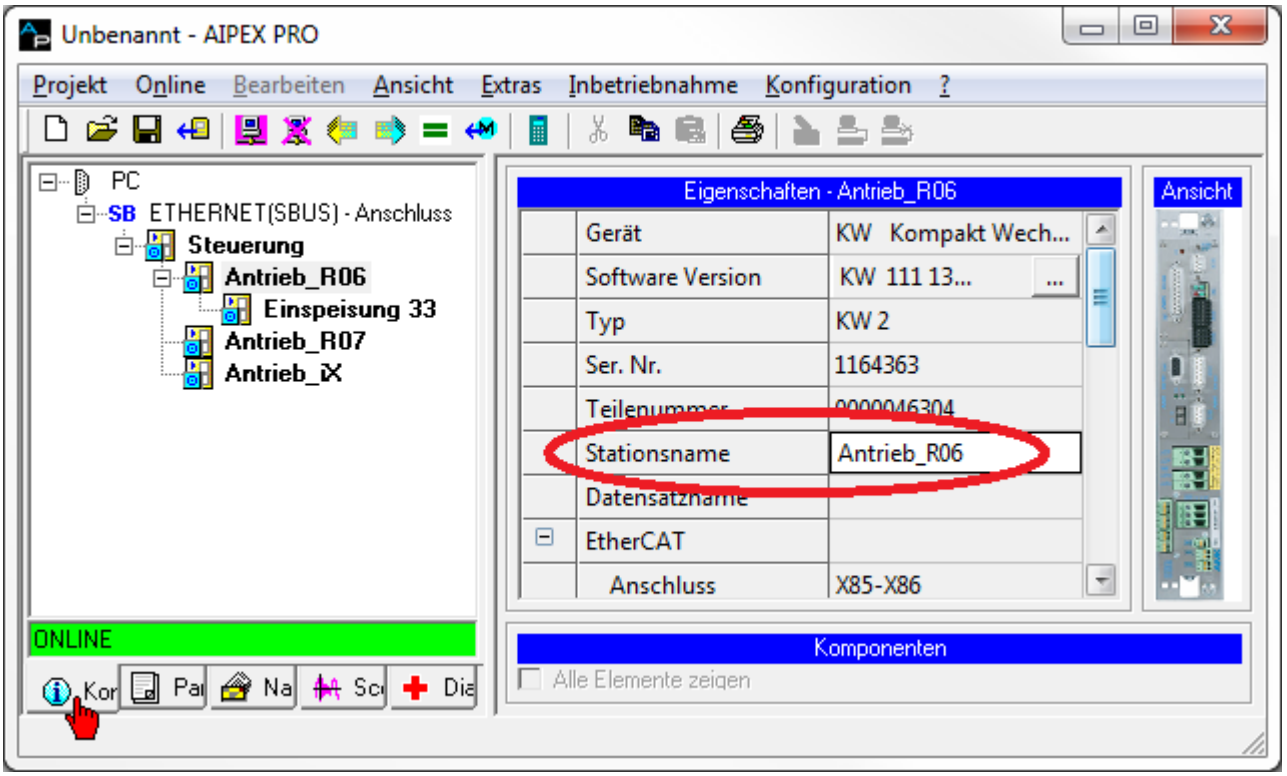


Prüfen Sie im Fenster 'Import Online' ob alle physikalisch vorhandenen Geräte erfasst sind.
Drücken Sie den Taster **'Komplett in das Projekt übernehmen'** um die Gerätedaten einzulesen.

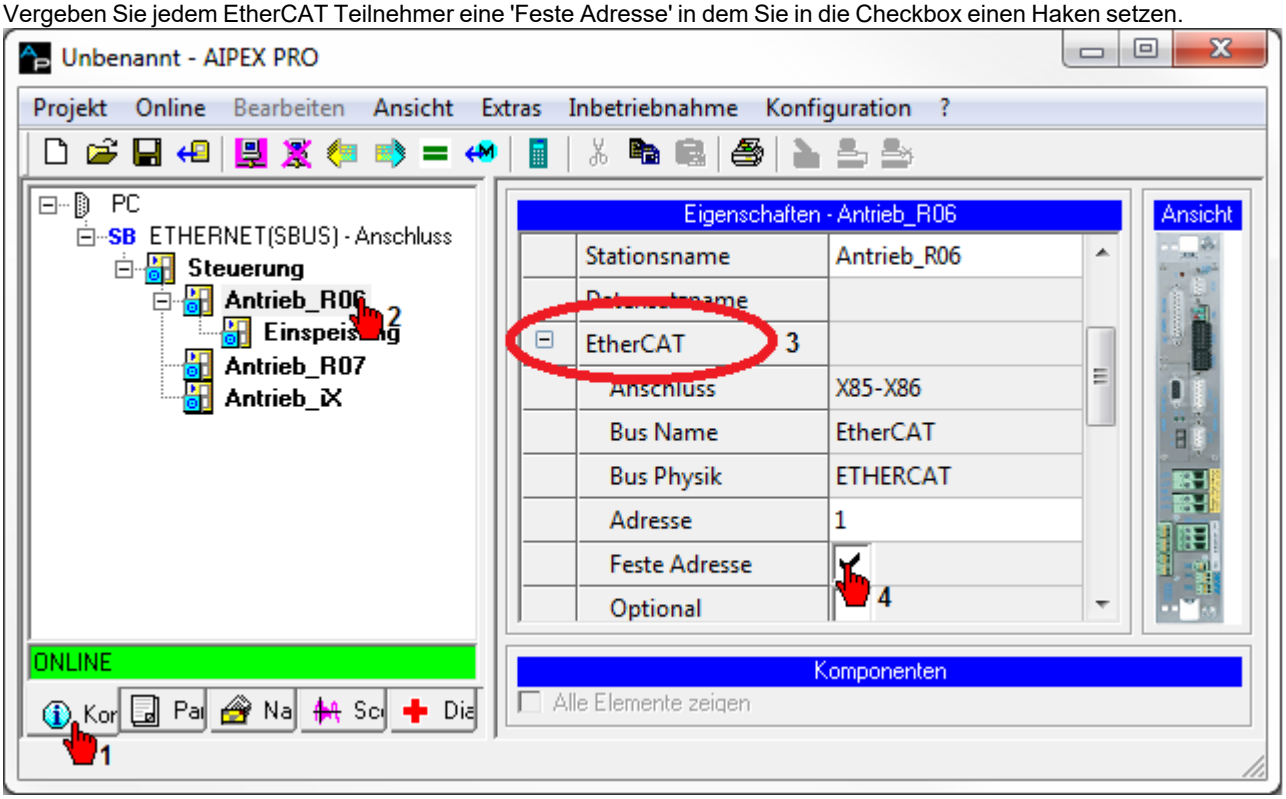


Prüfen Sie die Verkabelung und Systemzustände falls Geräte nicht angezeigt werden.

Sie können die Gerätenamen (Stationsnamen) an Ihre Applikation anpassen. Die Gerätenamen werden automatisch in das PLC Projekt als symbolische Gerätenamen (PLC Variablen) übernommen.

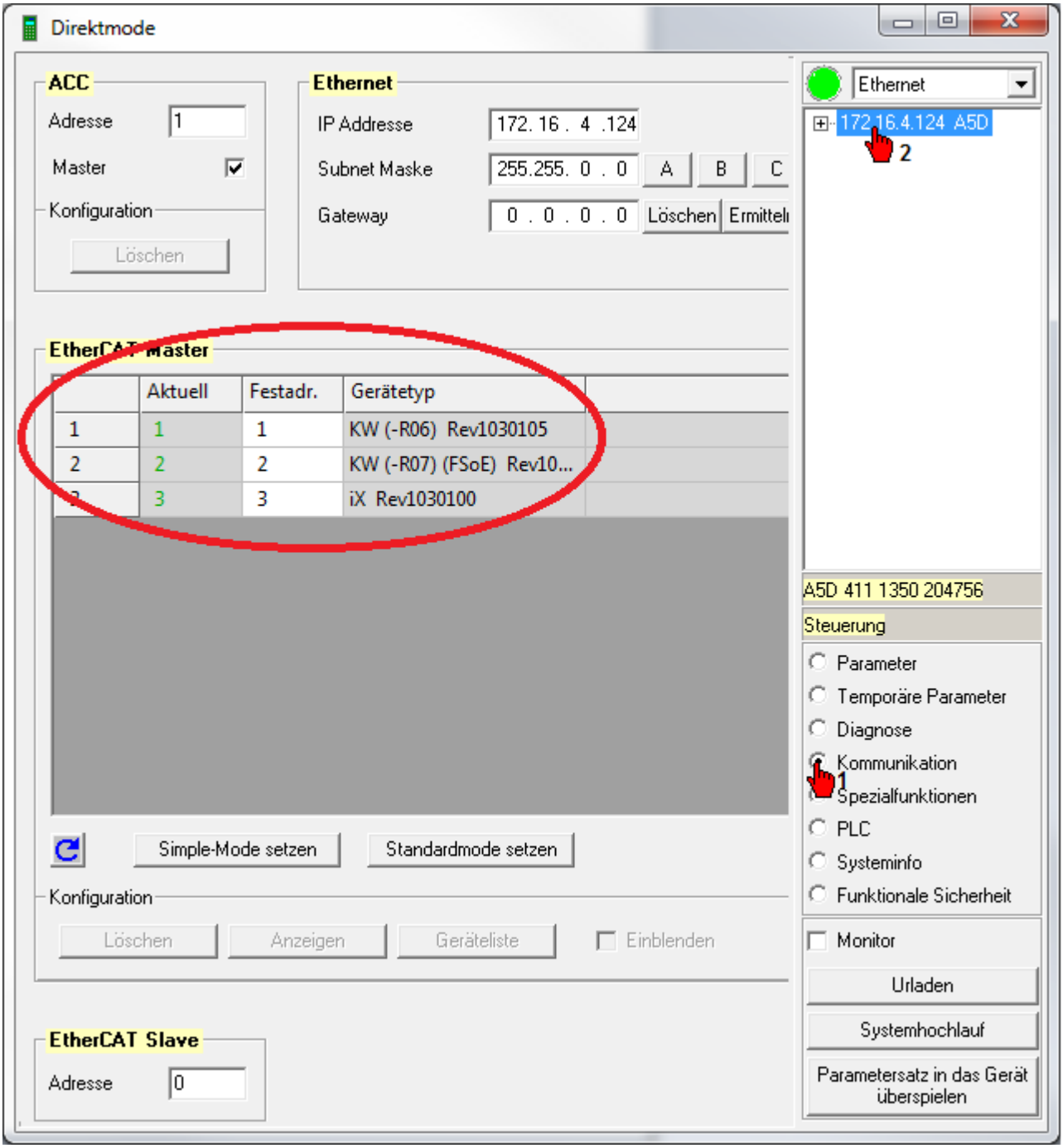


Vergeben Sie jedem EtherCAT Teilnehmer eine 'Feste Adresse' in dem Sie in die Checkbox einen Haken setzen.

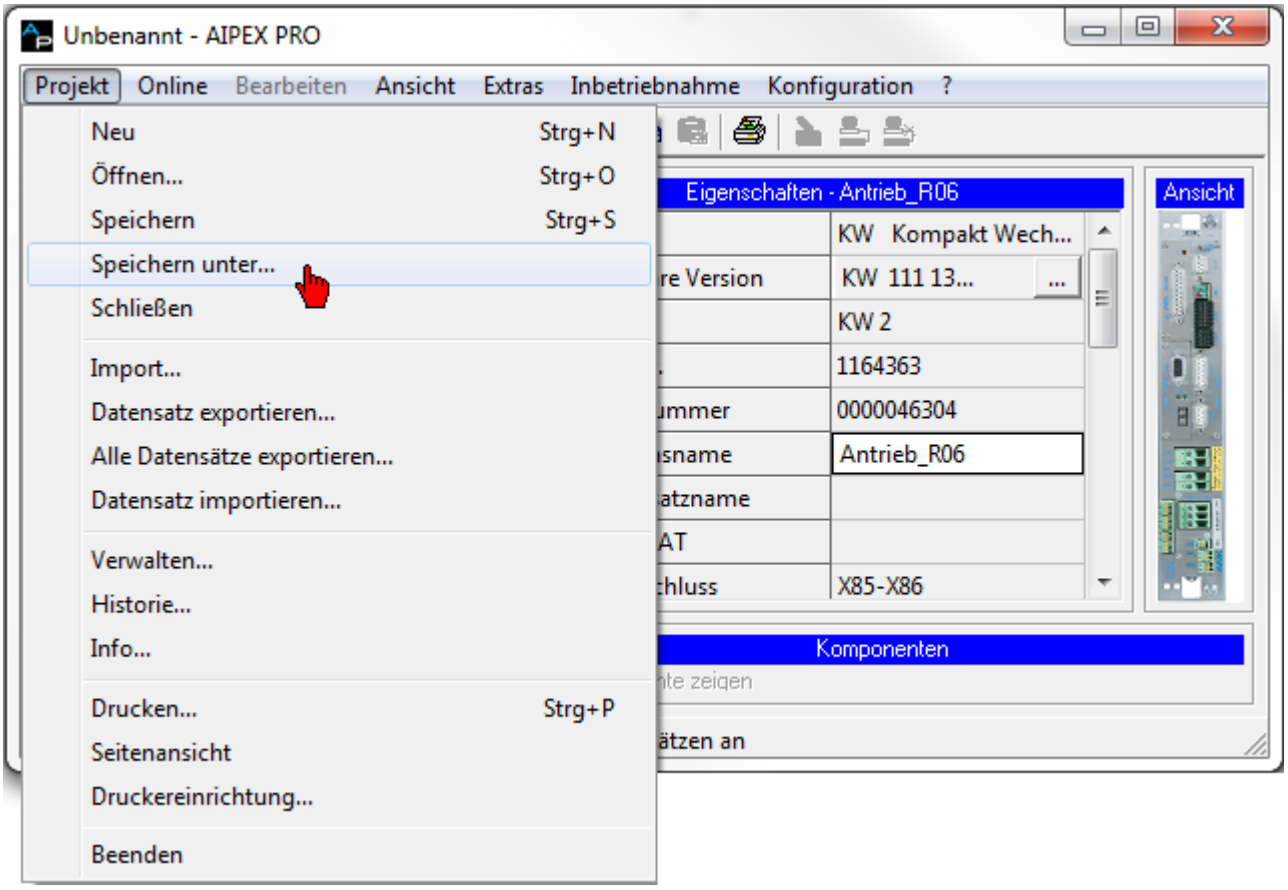


Führen Sie einen Neustart durch um die Änderung zu aktivieren.

Nachdem der Systemhochlauf vollständig abgeschlossen ist, können Sie im Menü 'Direktmode' → 'Kommunikation' die Adressierung überprüfen.



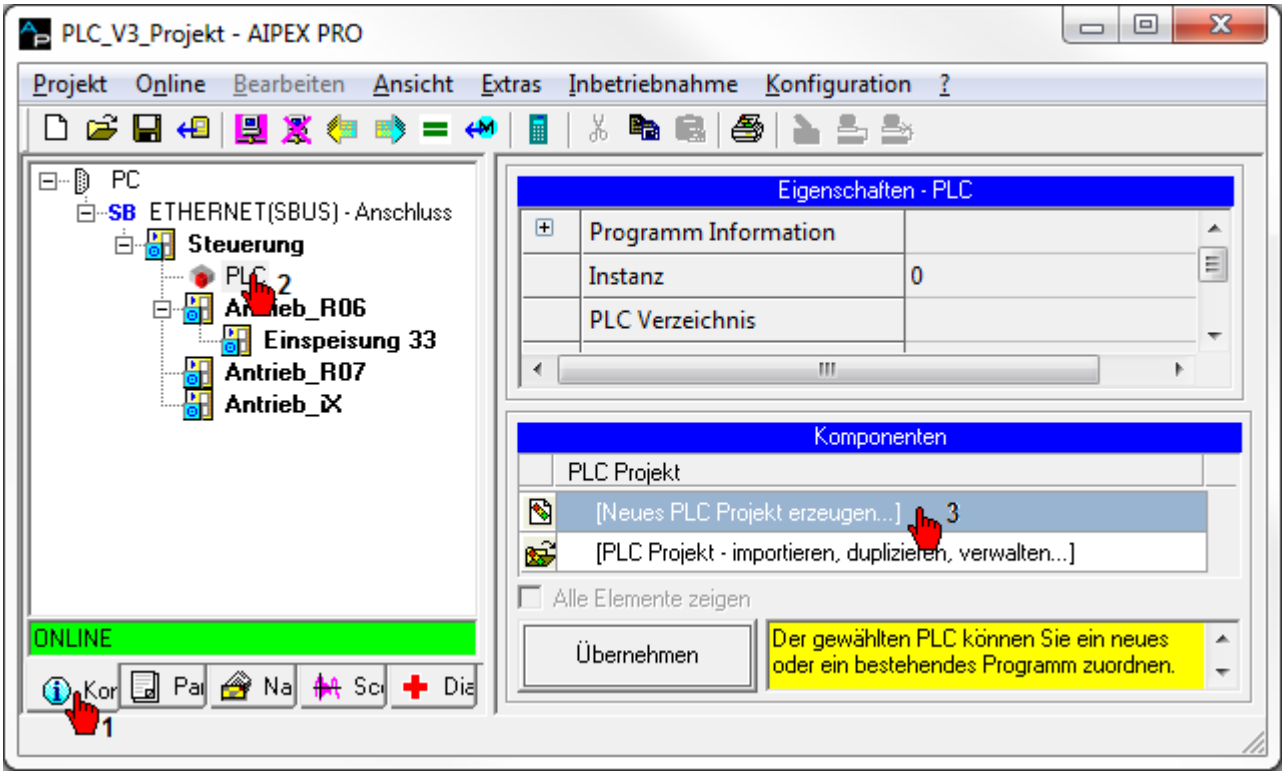
Speichern Sie das eingelesene Projekt unter 'Speichern unter...' ab. Dem PLC Projekt können Sie später einen separaten Namen vergeben.



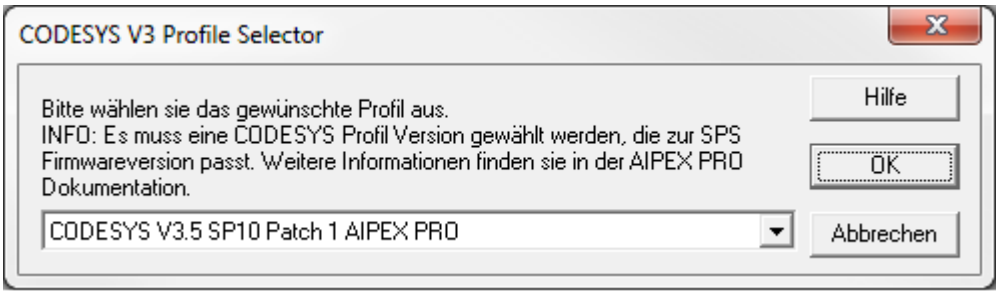
Welche CODESYS Version geöffnet wird, ist abhängig von der eingestellten CODESYS Version auf der Steuerung. CODESYS Versionsumstellung: Siehe 'CODESYS Versionsauswahl' auf Seite 669.

Wenn Sie die CODESYS Version erst jetzt ändern, müssen Sie das AIPEX PRO Projekt nochmals neu erstellen. Vorgehensweise Menü 'Projekt'→ 'Neu'. Siehe Anfang Kapitel.

Wählen Sie das Icon 'PLC' im Gerätebaum an.
Anschließend können Sie über **'Neues PLC Projekt erzeugen...'** das PLC Programmiersystem CODESYS öffnen.

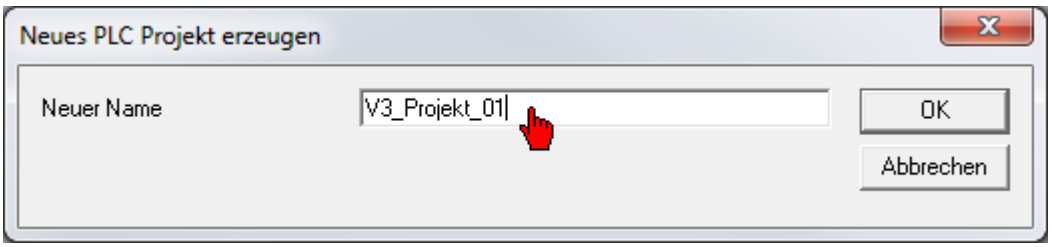


Icon '**PLC**' fehlt:
Klicken Sie mit der RMT in den Gerätebaum. Unter 'Anzeige auswählen' können Sie die Anzeige anpassen.
Siehe 'Programmübersicht - Anzeigefilter' auf Seite 39.



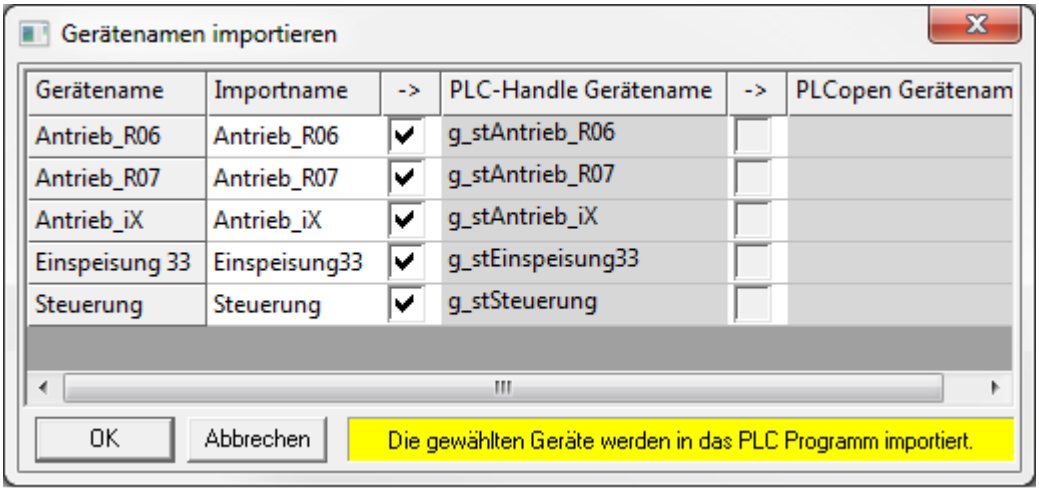
Siehe 'Versionierung CODESYS V3' auf Seite 741.

Vergeben Sie dem PLC Projekt einen Namen



CODESYS V3 startet

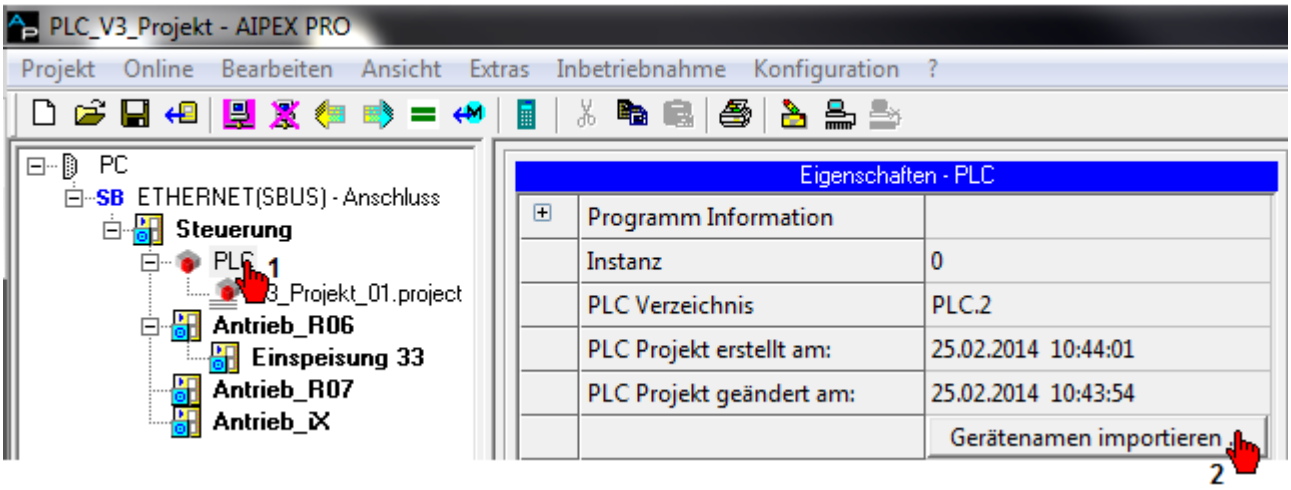
Die physikalisch vorhandenen Gerätenamen (aus dem AIPEX PRO Gerätebaum) werden automatisch als 'PLC-Handle Gerätenamen' (in der Dokumentation als symbolischer Gerätenamen bezeichnet) in das PLC Projekt importiert. Der Zusatz 'g_st' steht für eine globale Strukturvariable.



Diese Funktion wird über das Menü **'Extras'** → **'Optionen'** → **'Konfiguration erstellen'** → **'Gerätenamen beim Erstellen eines PLC-Projektes importieren'** aktiviert und deaktiviert.

Zusätzliche Information:

Manuell können die Gerätenamen mit dem Taster **'Gerätenamen importieren'** eingelesen werden.



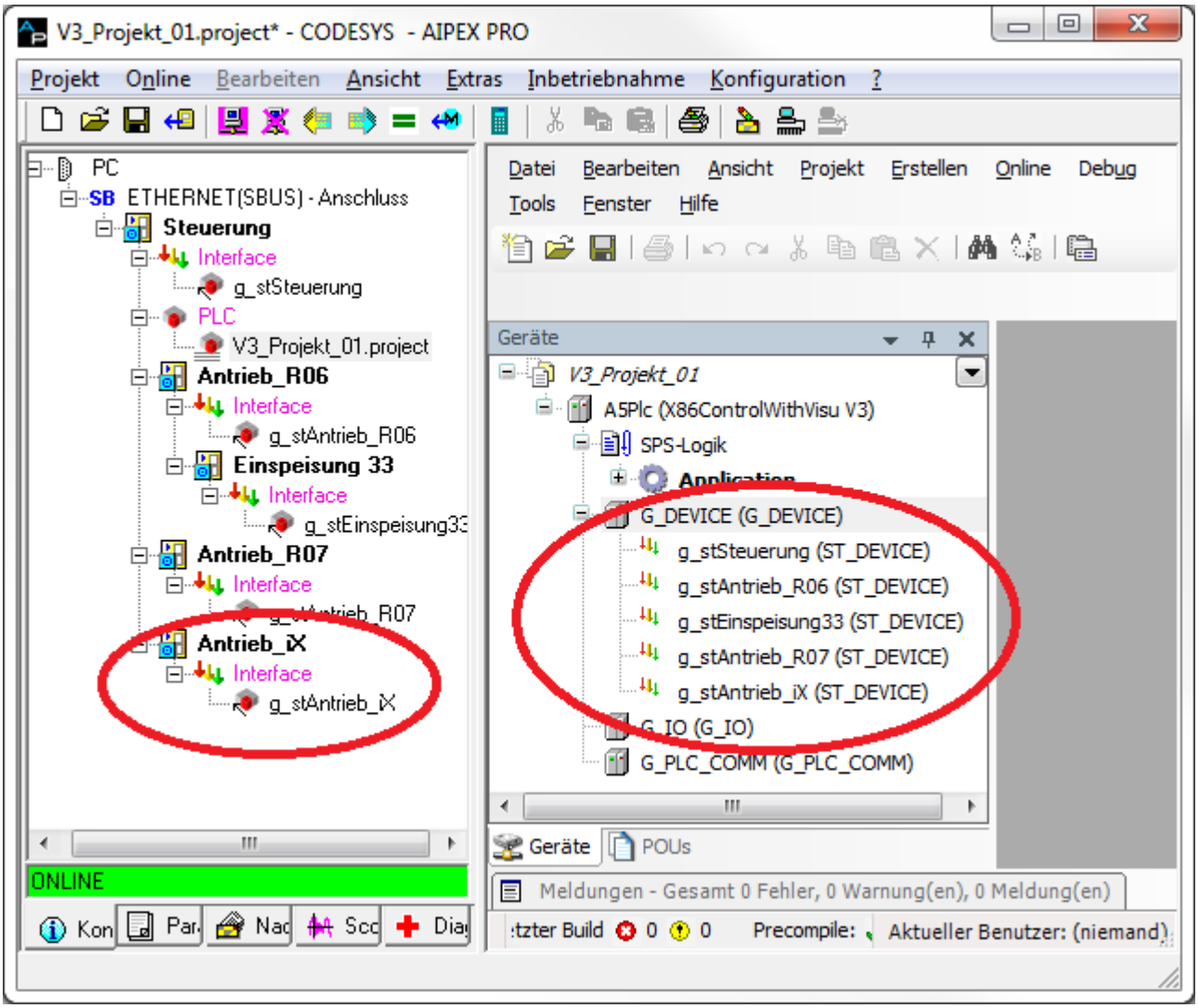
Wechsel zwischen der AIPEX und CODESYS Oberfläche

Klicken Sie auf ein beliebiges AIPEX Icon um in die AIPEX Oberfläche zu wechseln.
Klicken Sie auf den CODESYS PLC Projektnamen um in die CODESYS Oberfläche zu wechseln.

Die symbolischen Gerätenamen finden Sie auf der CODESYS Oberfläche im Gerätebaum unter 'G_DEVICE (G_DEVICE)' wieder.

Im AIPEX PRO Gerätebaum wird der symbolischen Gerätenamen automatisch mit dem dazugehörigen Geräteicon 'Interface' verknüpft.

Der automatische Nachrichtenkonfigurator nutzt diese Information um das Nachrichtenkonfigurationsfile zu erstellen.

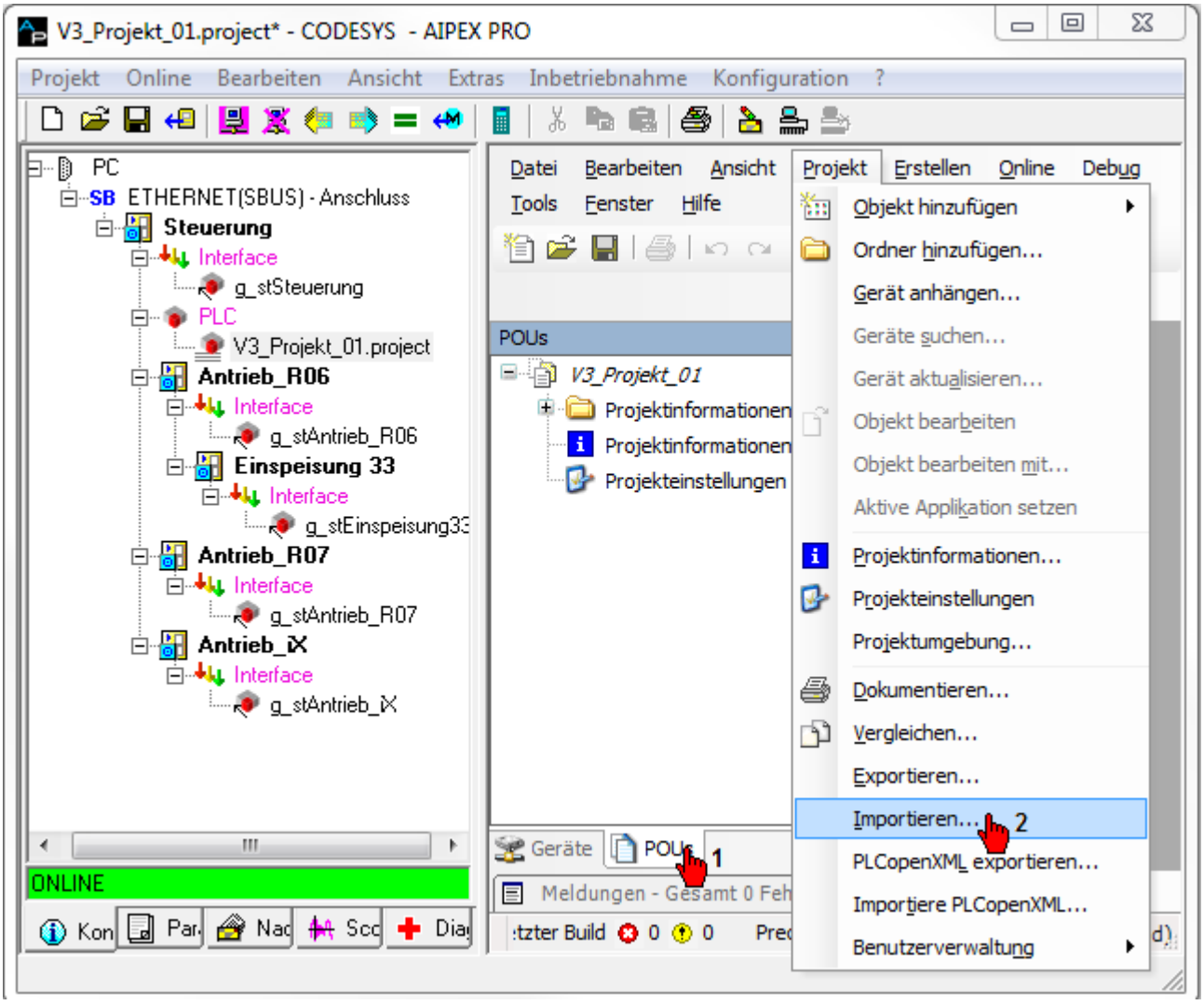


Zusätzliche Information:

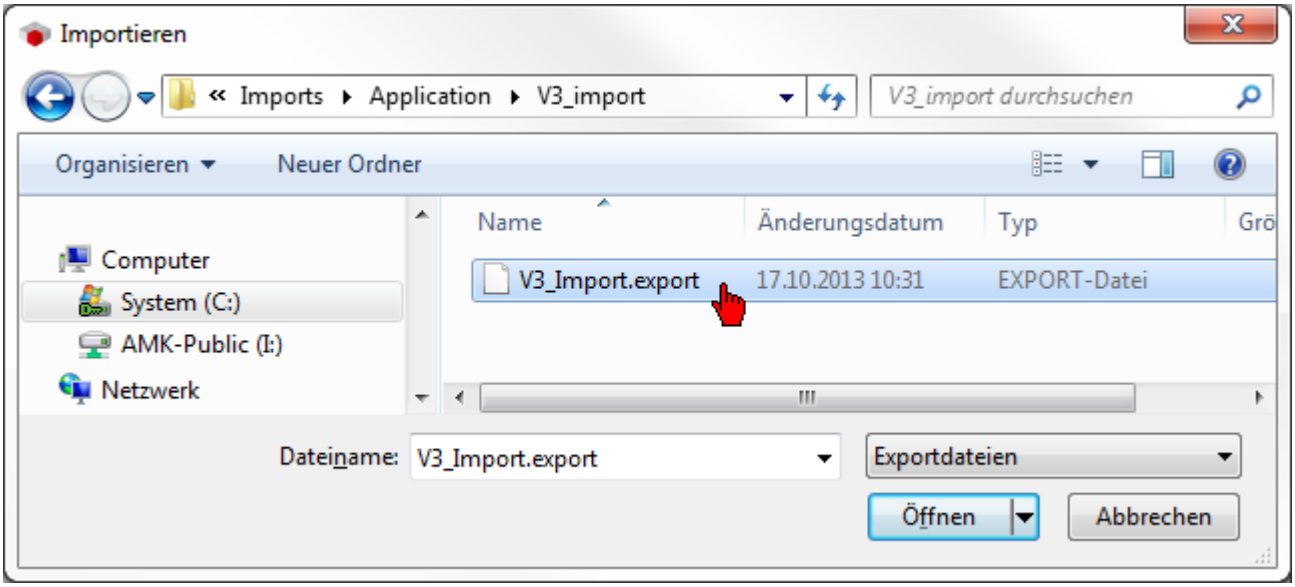
Die symbolischen Gerätenamen können auch manuell erstellt werden. In diesem Fall muss die Verknüpfung zwischen Geräte Interface und symbolischen Gerätenamen manuell zugewiesen werden. Dies findet statt, nachdem Sie das AIPEX Menü '**Konfiguration**' → '**Konfiguration erstellen**' aufgerufen haben. Siehe 'Automatische Buskonfiguration' auf Seite 743.

4.3.1.5 Standard Funktionsbausteine importieren

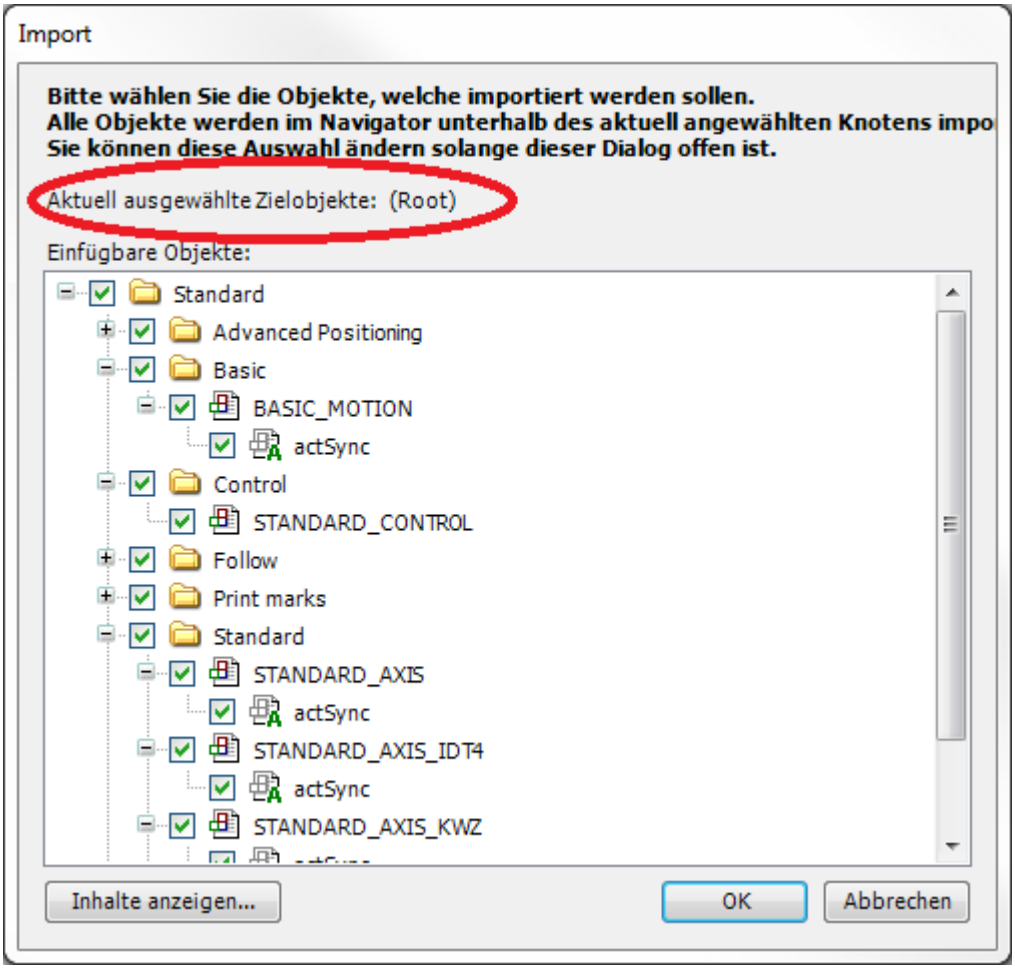
Wechseln Sie auf den Reiter 'POUs', bevor Sie die Importfunktion aufrufen.



Öffnen Sie die Datei 'V3_Import.export'. Sie finden die Datei unter dem Windows 7 Pfad:
C:\Programme (x86)\3S CODESYS\AmkAddition\Imports\Application\V3_import

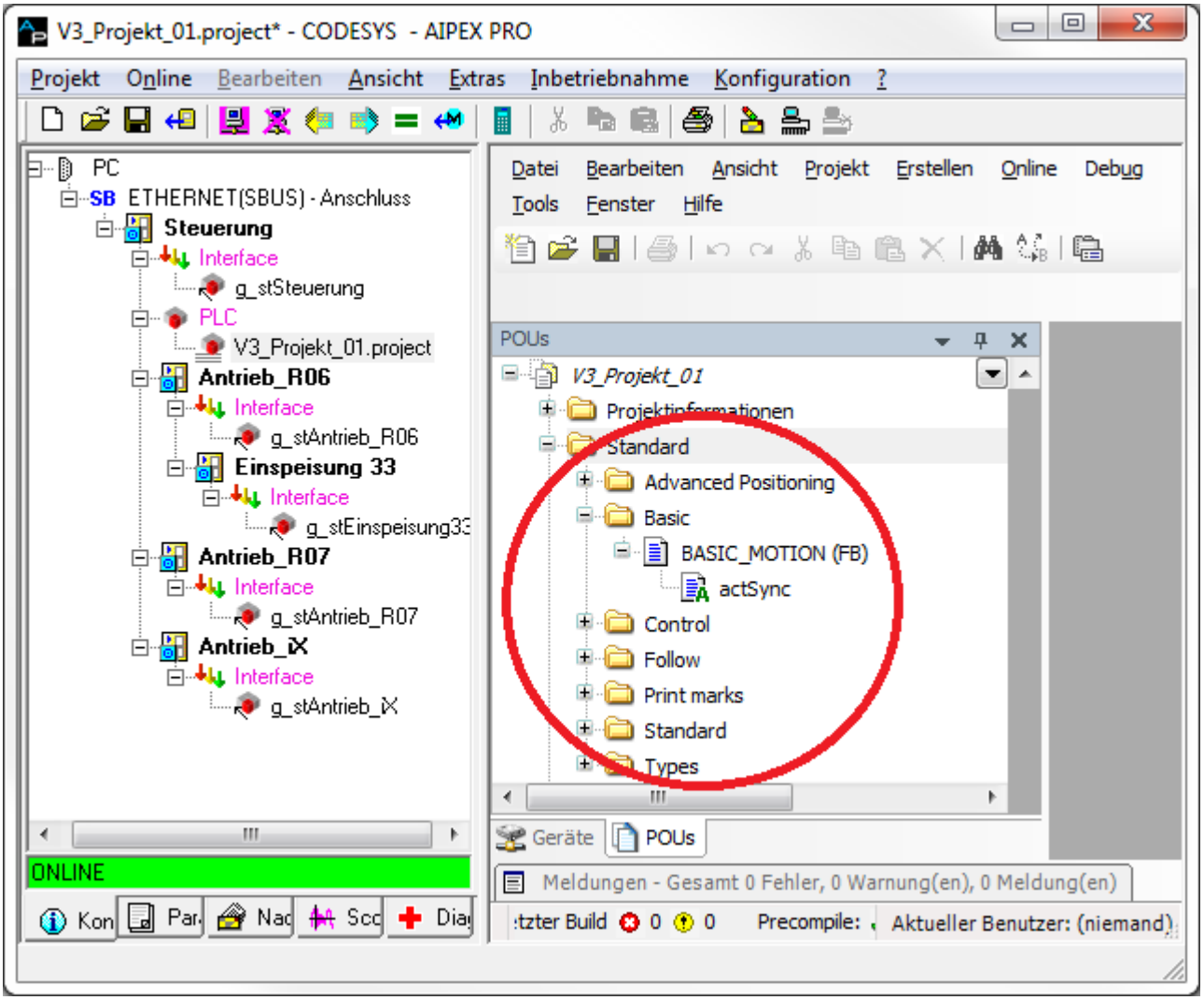


Importieren Sie alle Objekte.
Erfahrene Benutzer können die Objekte projektspezifisch auswählen.



Das Fenster 'Import' zeigt nichts an? Wechseln Sie auf den CODESYS Reiter 'POUs'.

Für die importierten Objekte wird unter 'POUs' ein Ordner mit dem Namen 'Standard' angelegt.



4.3.1.6 Standard Control

Folgende Grundfunktionalität für die Steuerung steht zu Verfügung:

- Statusbits (SBM Systembereit Meldung, QFL Quitt. Fehler gelöscht, Netzwerk und Bus Informationen)
- Steuern von FL Fehler löschen
- Auslesen von Diagnosemeldungen

Zusätzliche Information:

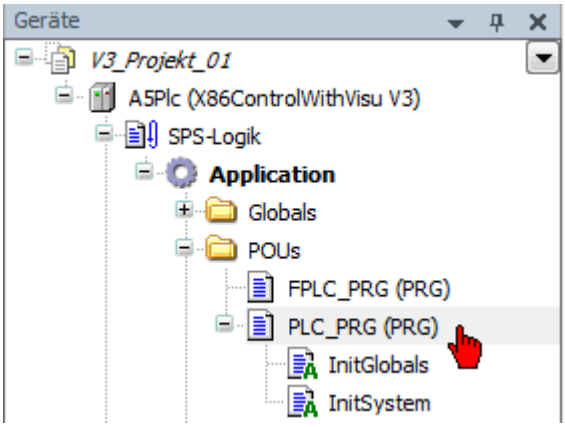
Beschreibung Funktionsbaustein: Siehe 'STANDARD_CONTROL (FB)' auf Seite 197.

Der Funktionsbaustein 'STANDARD_CONTROL' benötigt die Struktur 'ST_CONTROL'.

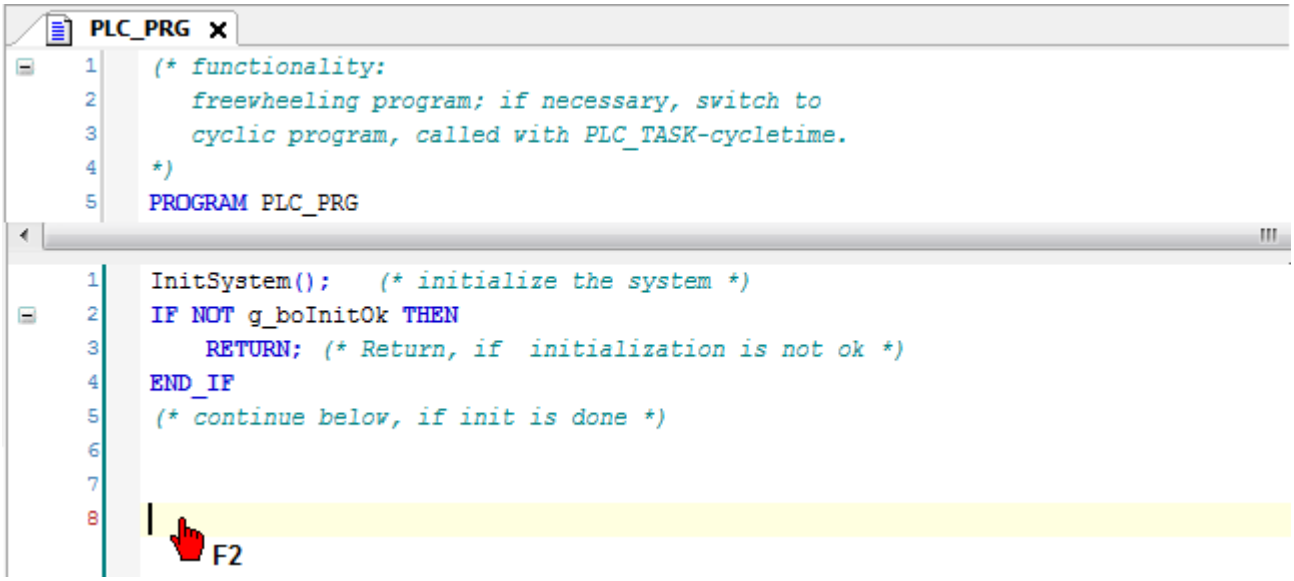
Beschreibung Struktur: Siehe 'ST_CONTROL (ST)' auf Seite 191.

Wechseln Sie auf den Reiter 'Geräte'

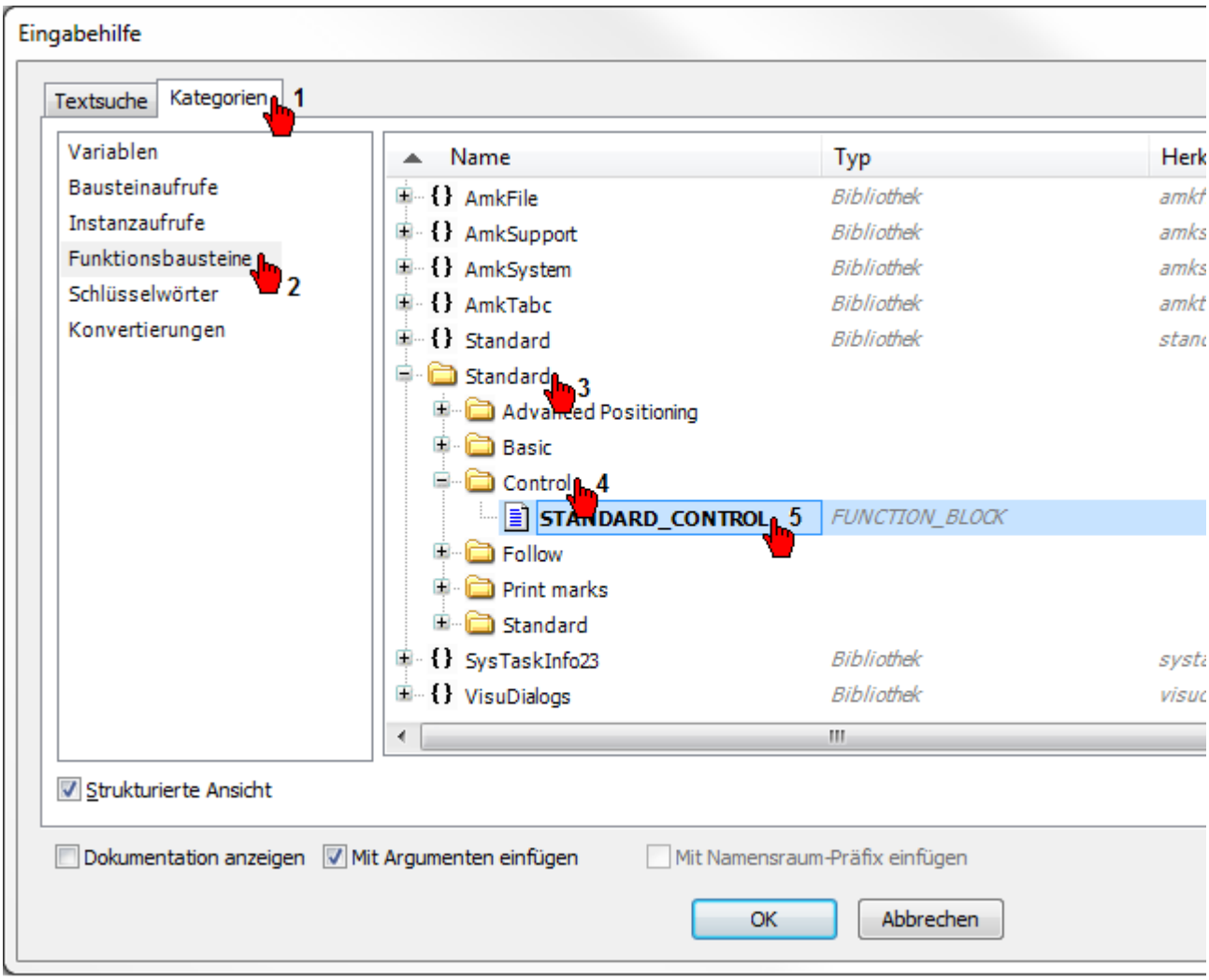
Öffnen Sie den Programmbaustein 'PLC_PRG (PRG)'



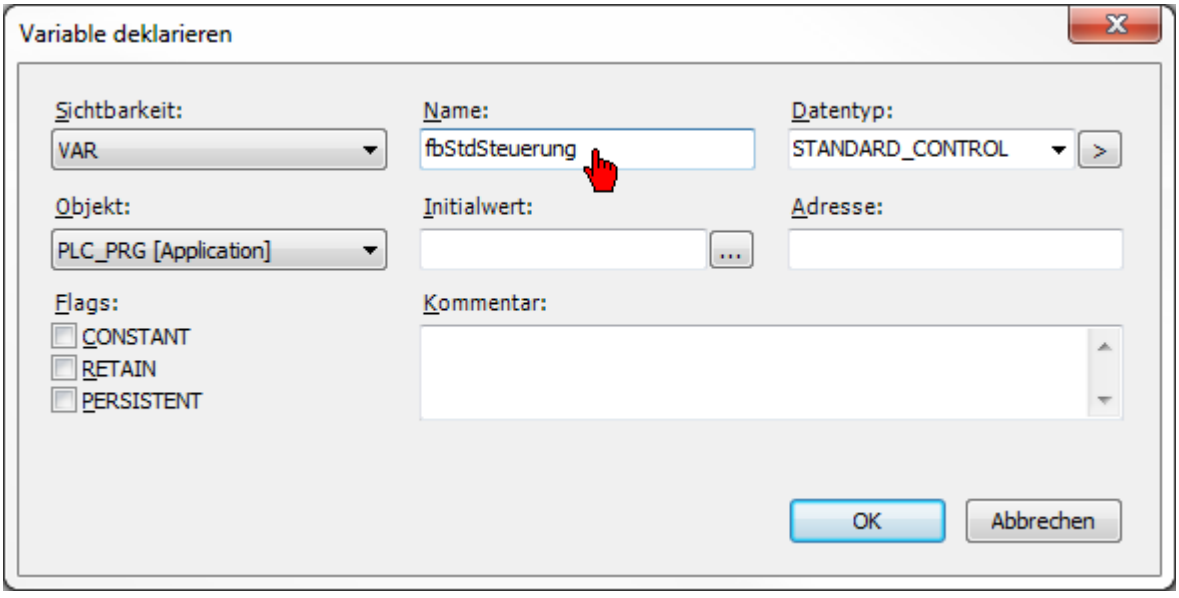
Klicken Sie im Programmeditor in eine leere Zeile. Mit der **'F2'** Taste öffnet sich die 'Eingabehilfe'.



Fügen Sie den Funktionsbaustein 'STANDARD_CONTROL' ein.



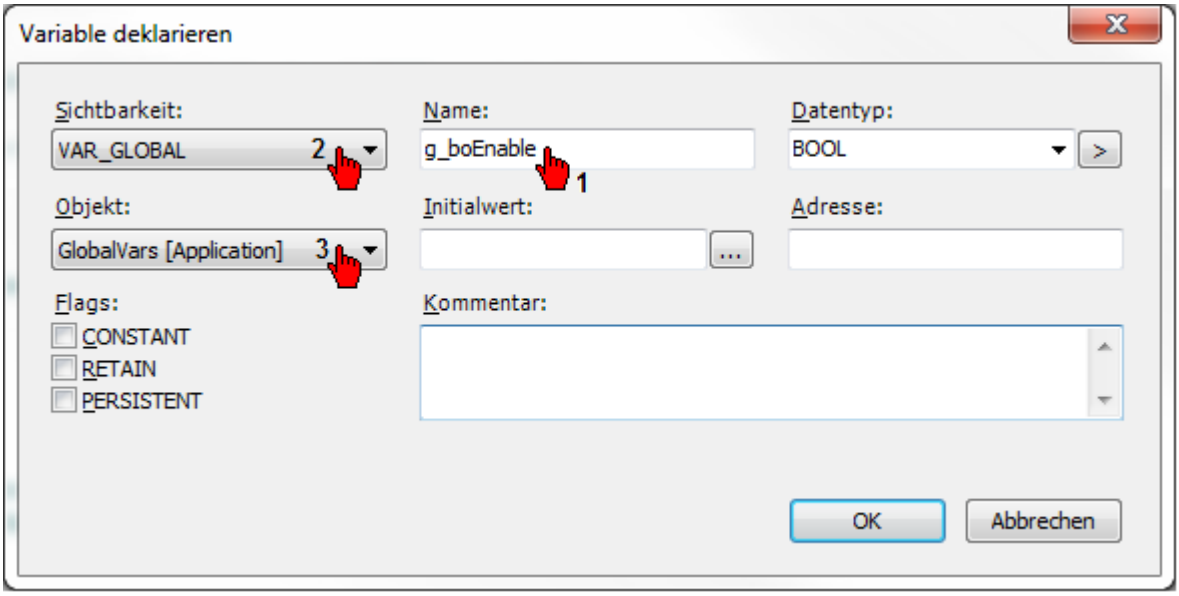
Instanzieren Sie den Funktionsbaustein indem Sie ihm einen eigenen Namen zuweisen.



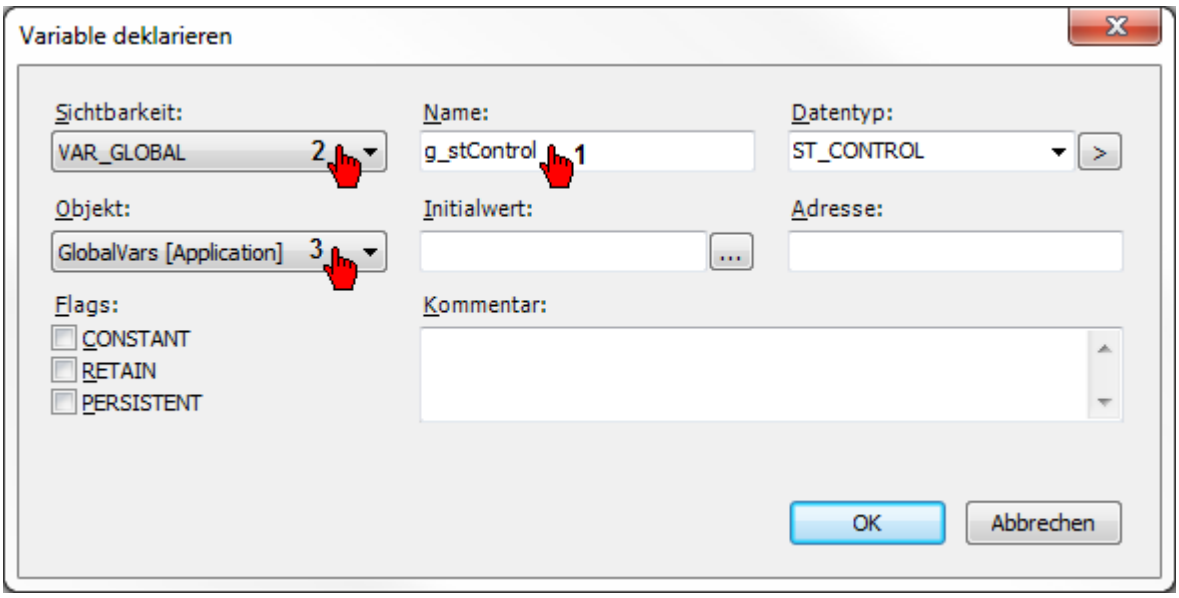
Erstellen Sie die benötigten Variablen,
tragen Sie bei der Variablen 'stDevice' den symbolischen Gerätenamen aus der 'Steuerungskonfiguration' ein.

```
fbStdSteuerung(
    boEnable:= TRUE, (* Bausteinfreigabe wird bei PLC Start auf TRUE gesetzt *)
    boErrorReset:= ,
    uiBusInstance:= 5, (* Instanz EtherCAT *)
    boEnabAck=> ,
    boSBM=> ,
    boNetworkReady=> ,
    boBusReady=> g_boEnable, (* Globales Freigabesignal für weitere Funktionsbausteine *)
    boBusWarning=> ,
    boBusError=> ,
    boErr=> ,
    iErrID=> ,
    enErrName=> ,
    stControl:= g_stControl, (* AMK Struktur für steuerungsspezifische Daten *)
    stDevice:= g_stSteuerung); (* Symbolischer Gerätenamen *)
```

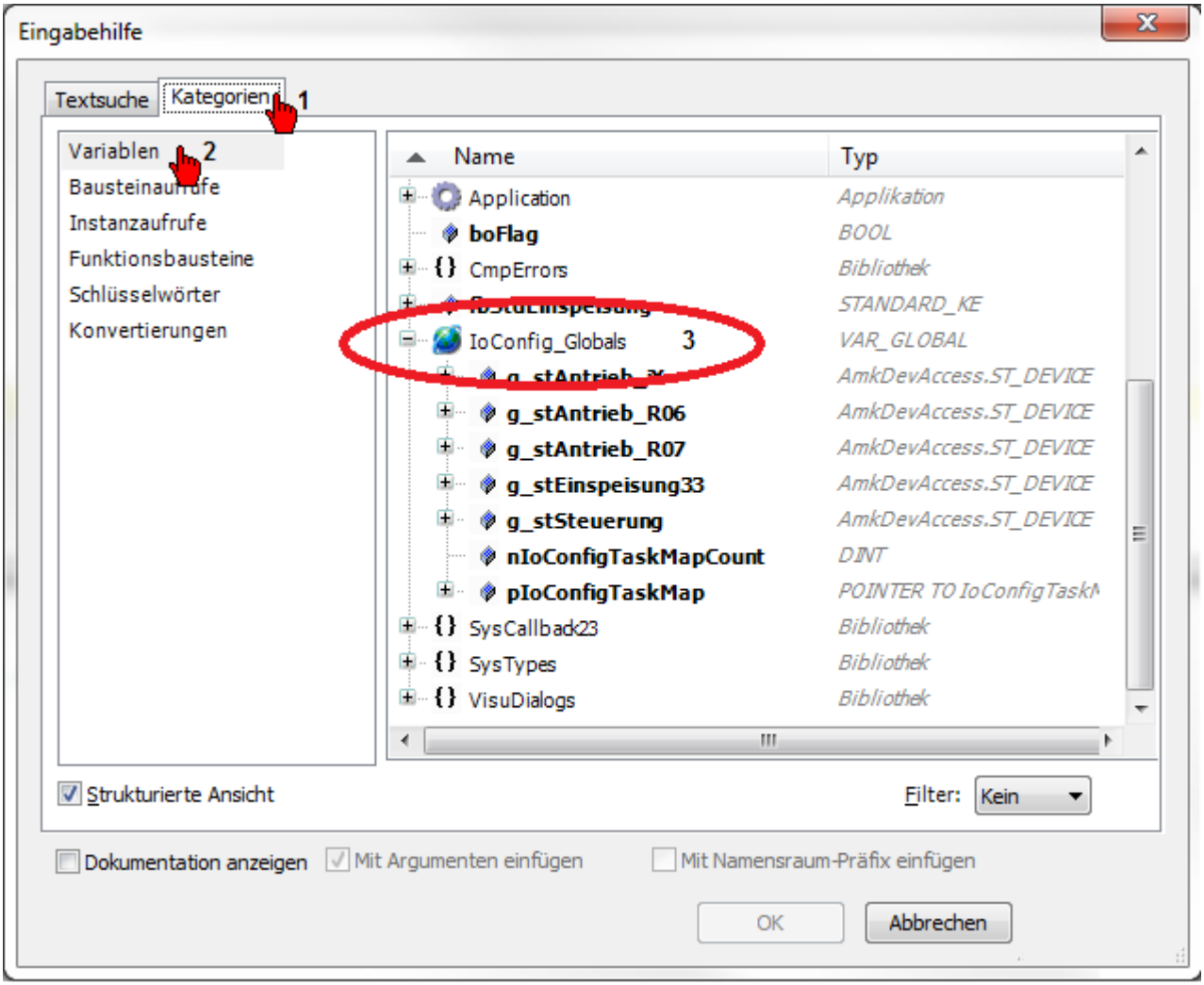
Globale Variable 'g_boEnable' deklarieren.



Globale Strukturvariable 'g_stControl' deklarieren.



Sie finden die symbolischen Gerätenamen unter 'IoConfig_Globals'.



4.3.1.7 Standard Einspeisung

Folgende Grundfunktionalität für die Einspeisung steht zu Verfügung:

- Statusbits (QUE Quitt. Umrichter Ein, SBM Systembereit Meldung)
- Steuerbits (UE Umrichter Ein, FL Fehler Löschen)

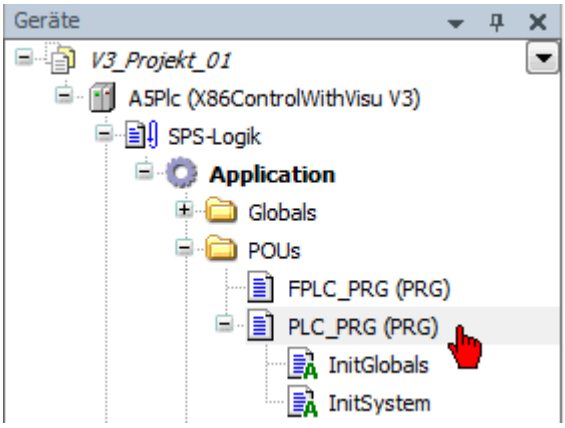
- Ausgabe Diagnosenummer

Zusätzliche Information:

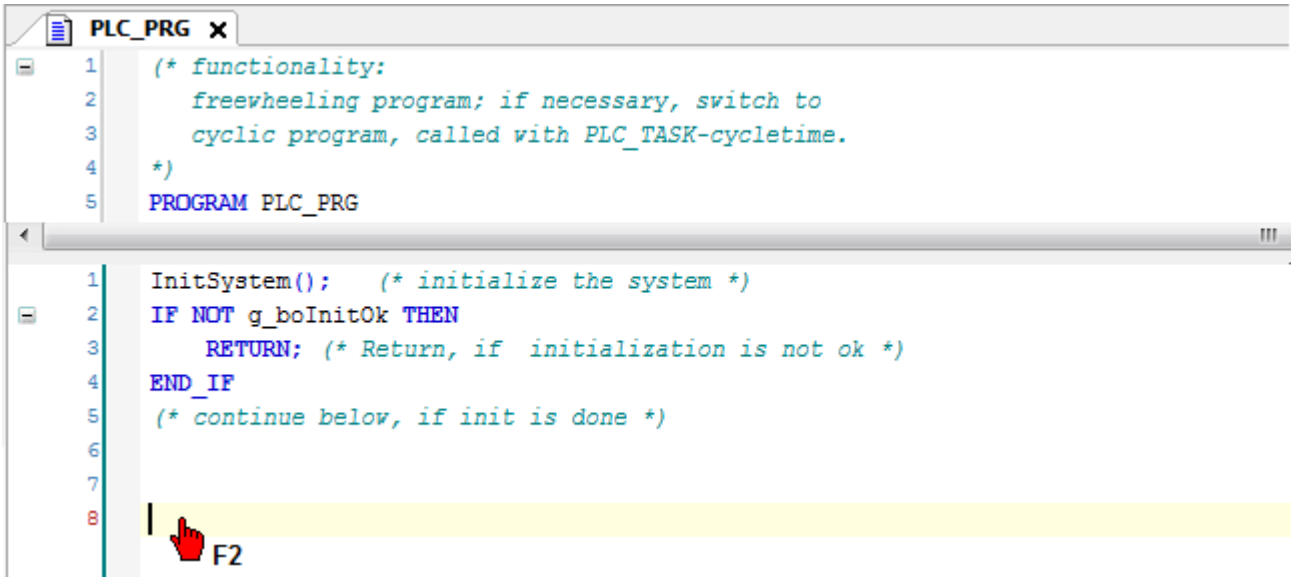
Beschreibung Funktionsbaustein: Siehe 'STANDARD_CONTROL (FB)' auf Seite 197.

Wechseln Sie auf den Reiter 'Geräte'

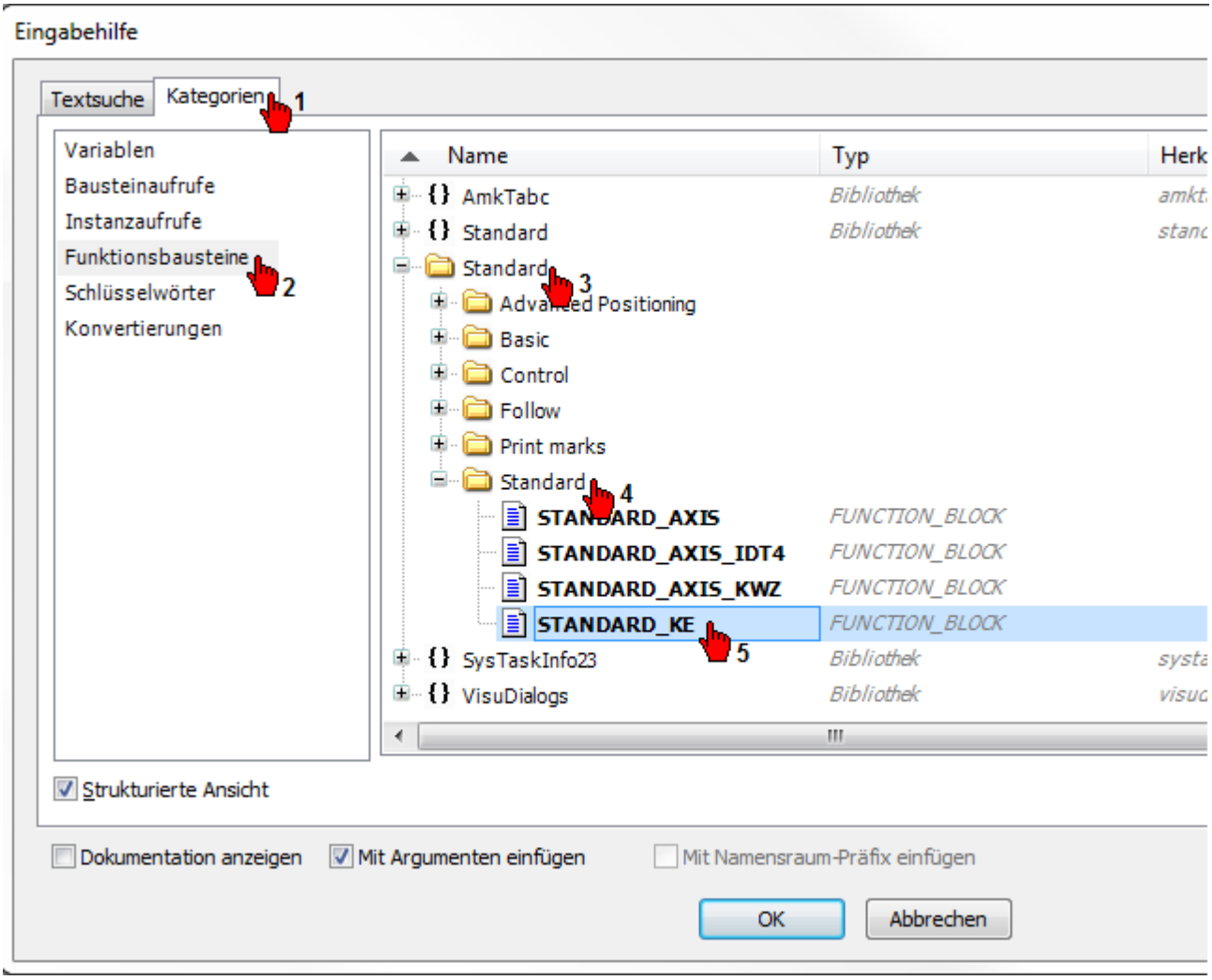
Öffnen Sie den Programmbaustein 'PLC_PRG (PRG)'



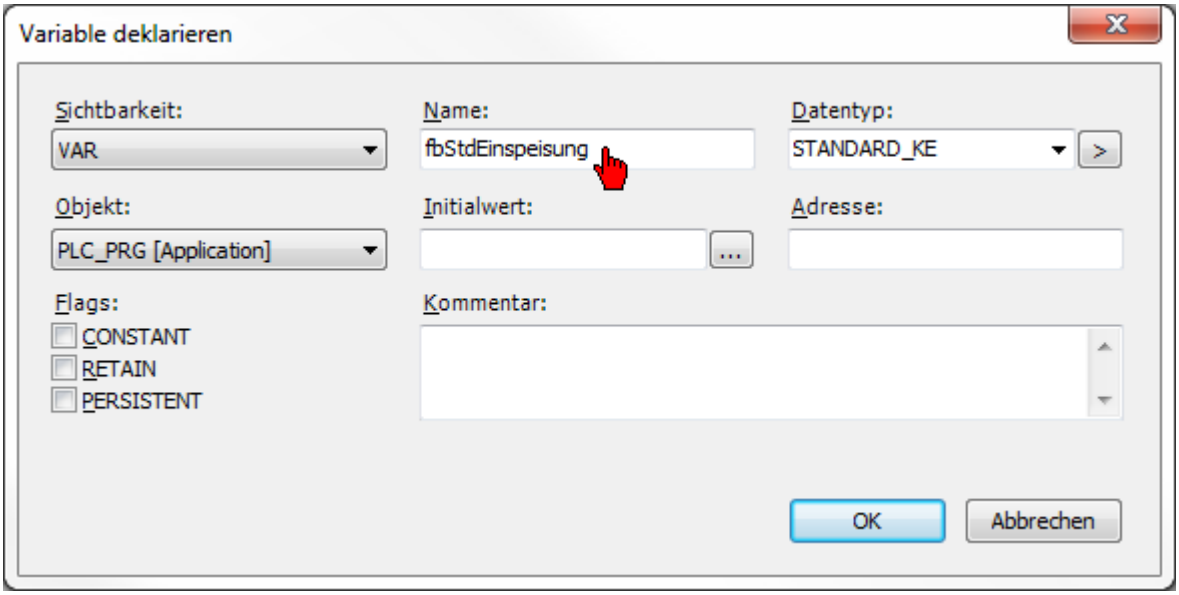
Klicken Sie im Programmeditor in eine leere Zeile. Mit der **'F2'** Taste öffnet sich die 'Eingabehilfe'.



Fügen Sie den Funktionsbaustein 'STANDARD_KE' ein.



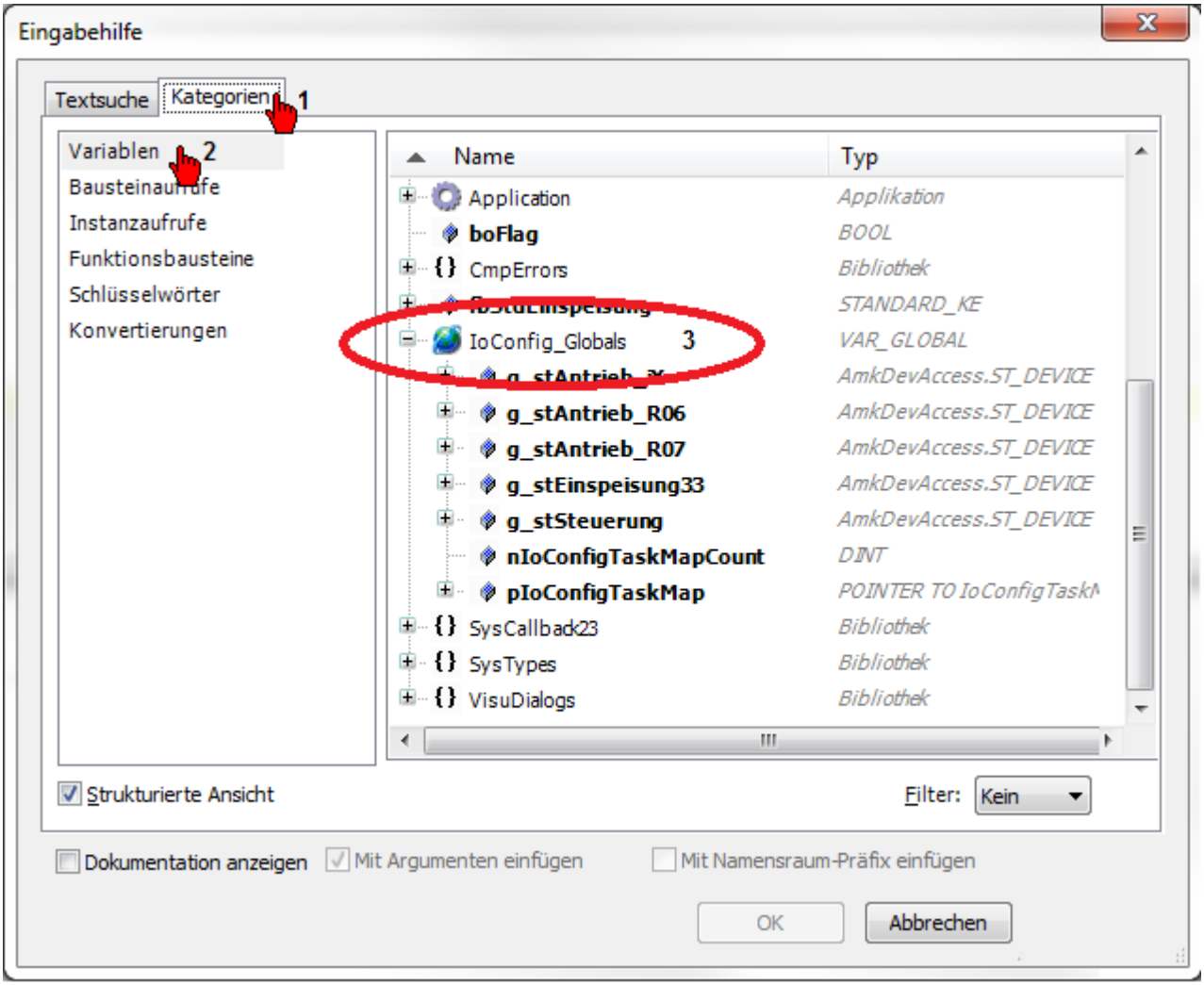
Instanzieren Sie den Funktionsbaustein indem Sie ihm einen eigenen Namen zuweisen.



Erstellen Sie die benötigten Variablen,
tragen Sie bei der Variablen 'stDevice' den symbolischen Gerätenamen aus der 'Steuerungskonfiguration' ein.

```
fbStdEinspeisung(
    boEnable:= g_boEnable, (* Globales Freigabesignal *)
    boUE:= ,
    boFL:= ,
    stDevice:= g_stEinspeisung33, (* Symbolischer Gerätenamen*)
    boEnabAck=> ,
    uiDiagnosticNr=> ,
    boQUE=> ,
    boSBM=> ,
    boErr=> ,
    iErrID=> ,
    enErrName=> );
```

Sie finden die symbolischen Gerätenamen unter 'IoConfig_Globals'.



4.3.1.8 Standard AXIS

Folgende Grundfunktionalität für die Achse steht zu Verfügung:

Ein und Ausgänge am Funktionsblock

- Steuersignale (RF Reglerfreigabe, FL Fehler Löschen)
- Not-Halt
- Drehzahlregelung
- Referenzpunktfahrt

- Positionierung (absolut / relativ)
- Tippbetrieb (Minus / Plus)
- Statussignale der Achsfunktionen

Zugriff über Programmcode

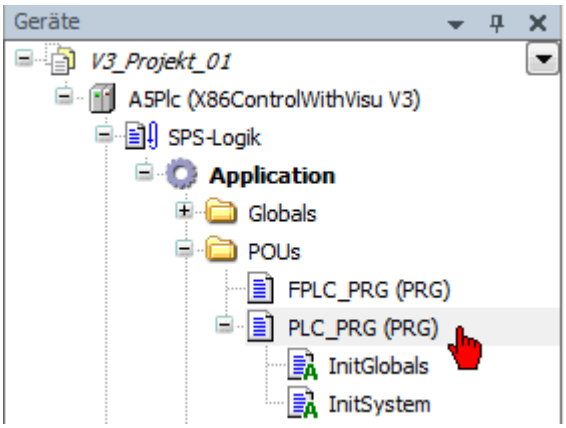
- Konfiguration von Statusmeldungen (ID26 'Konfiguration Statusbits')
- Lesen von Parametern entsprechend einer Vorgabeliste
- Positionsistwerte gelached
- Vorsteuerung (Moment und Drehzahl)

Der Funktionsbaustein 'STANDARD_AXIS' benötigt den Funktionsbaustein 'BASIC_MOTION' und die Struktur 'ST_AXIS_DRIVE'.
'STANDARD_AXIS' besteht aus einem asynchronen Funktionsteil der im PLC_PRG aufgerufen wird und einem synchronen Funktionsteil der im FPLC_PRG aufgerufen wird.

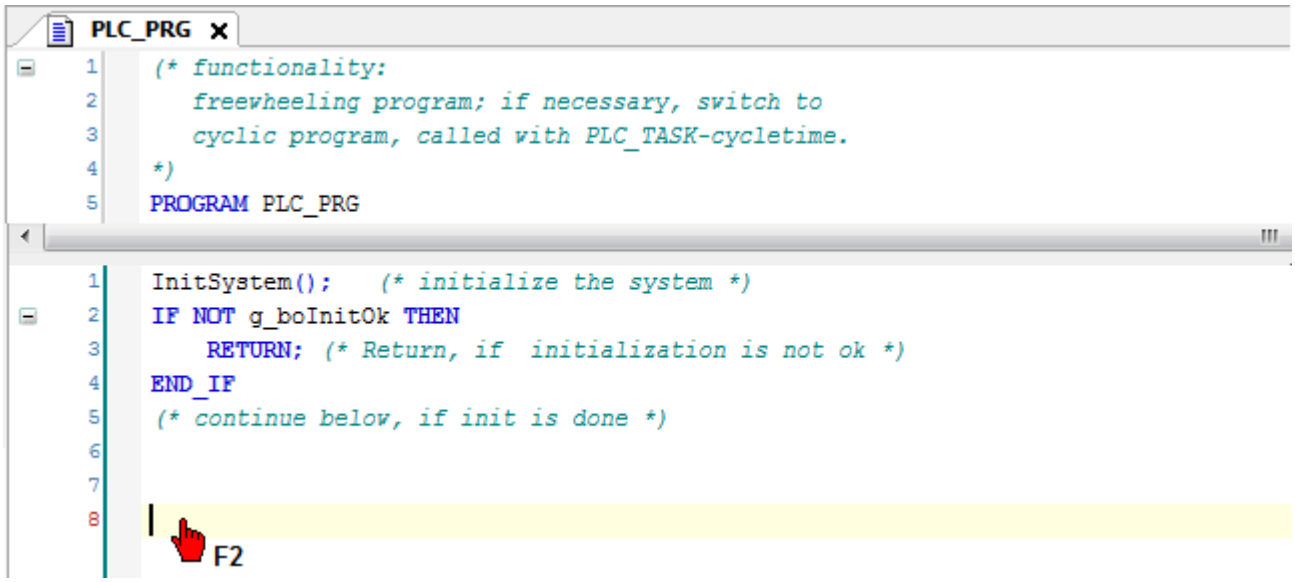
Zusätzliche Information:

Beschreibung Funktionsbaustein: Siehe 'STANDARD_AXIS (FB) (STANDARD_AXIS_KWZ, STANDARD_AXIS_IDT4, STANDARD_AXIS_ihX)' auf Seite 202.
Beschreibung Struktur: Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.

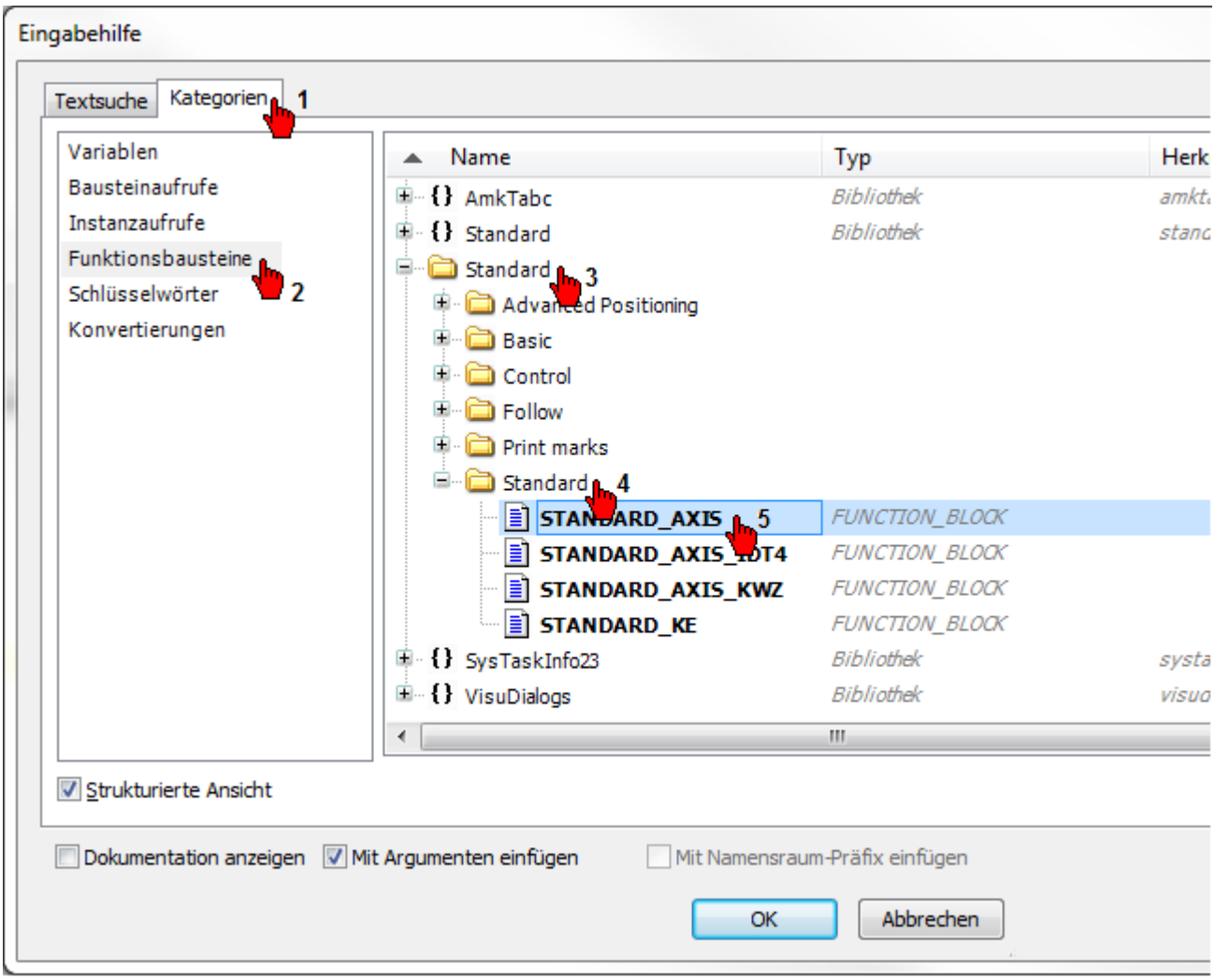
Wechseln Sie auf den Reiter 'Geräte'
Öffnen Sie den Programmbaustein 'PLC_PRG (PRG)'



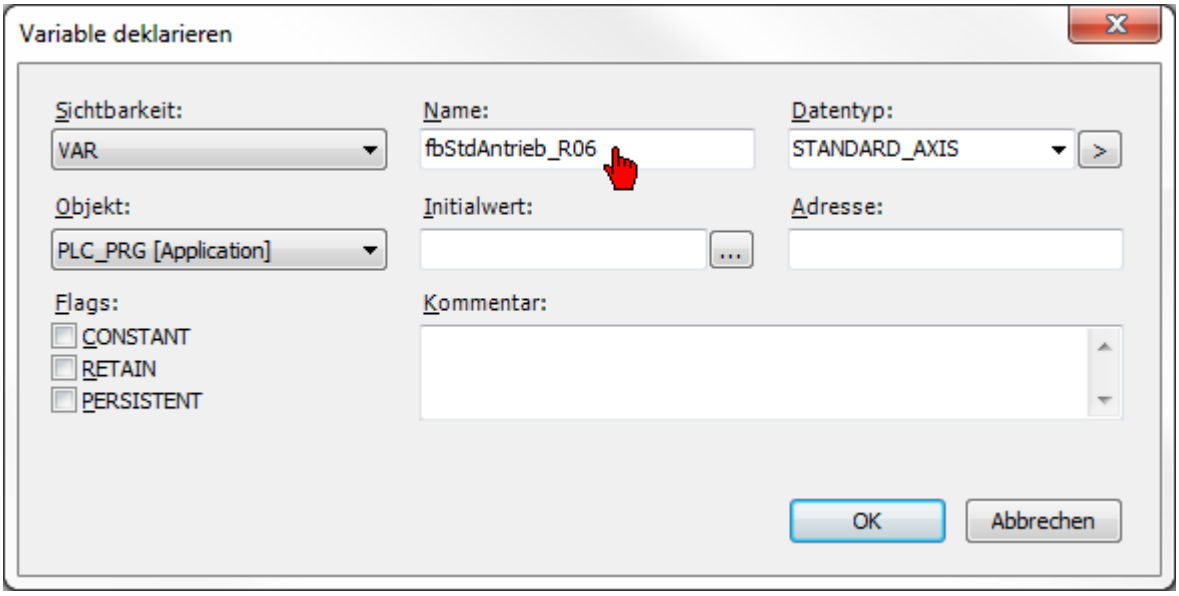
Klicken Sie im Programmeditor in eine leere Zeile. Mit der **'F2'** Taste öffnet sich die 'Eingabehilfe'.



Fügen Sie den Funktionsbaustein 'STANDARD_AXIS' ein.



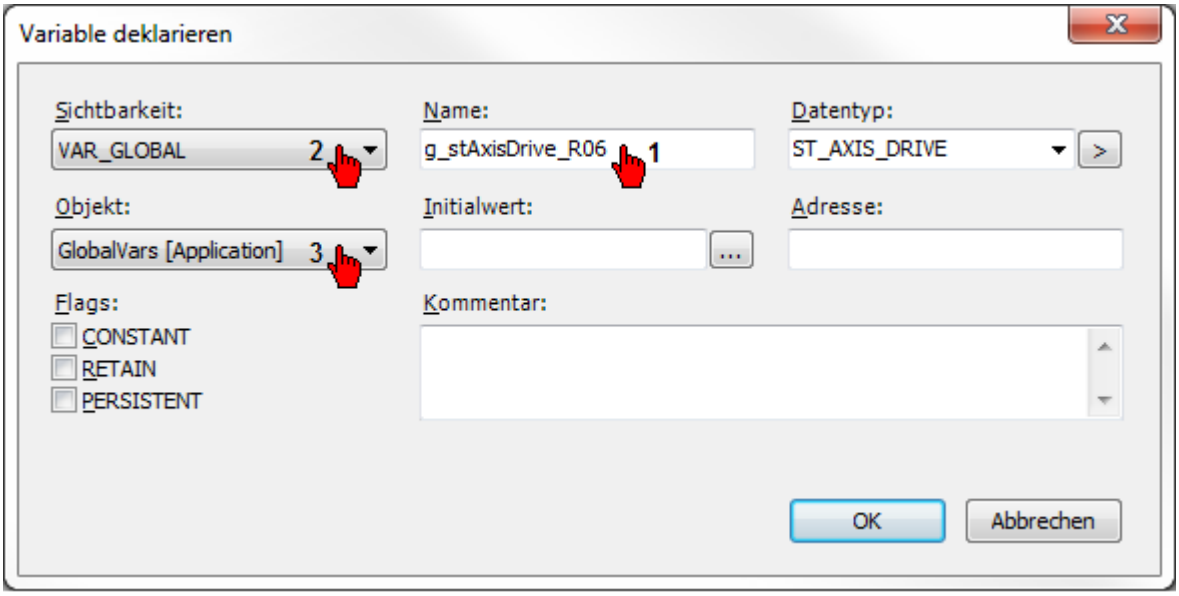
Instanzieren Sie den Funktionsbaustein indem Sie ihm einen eigenen Namen zuweisen.



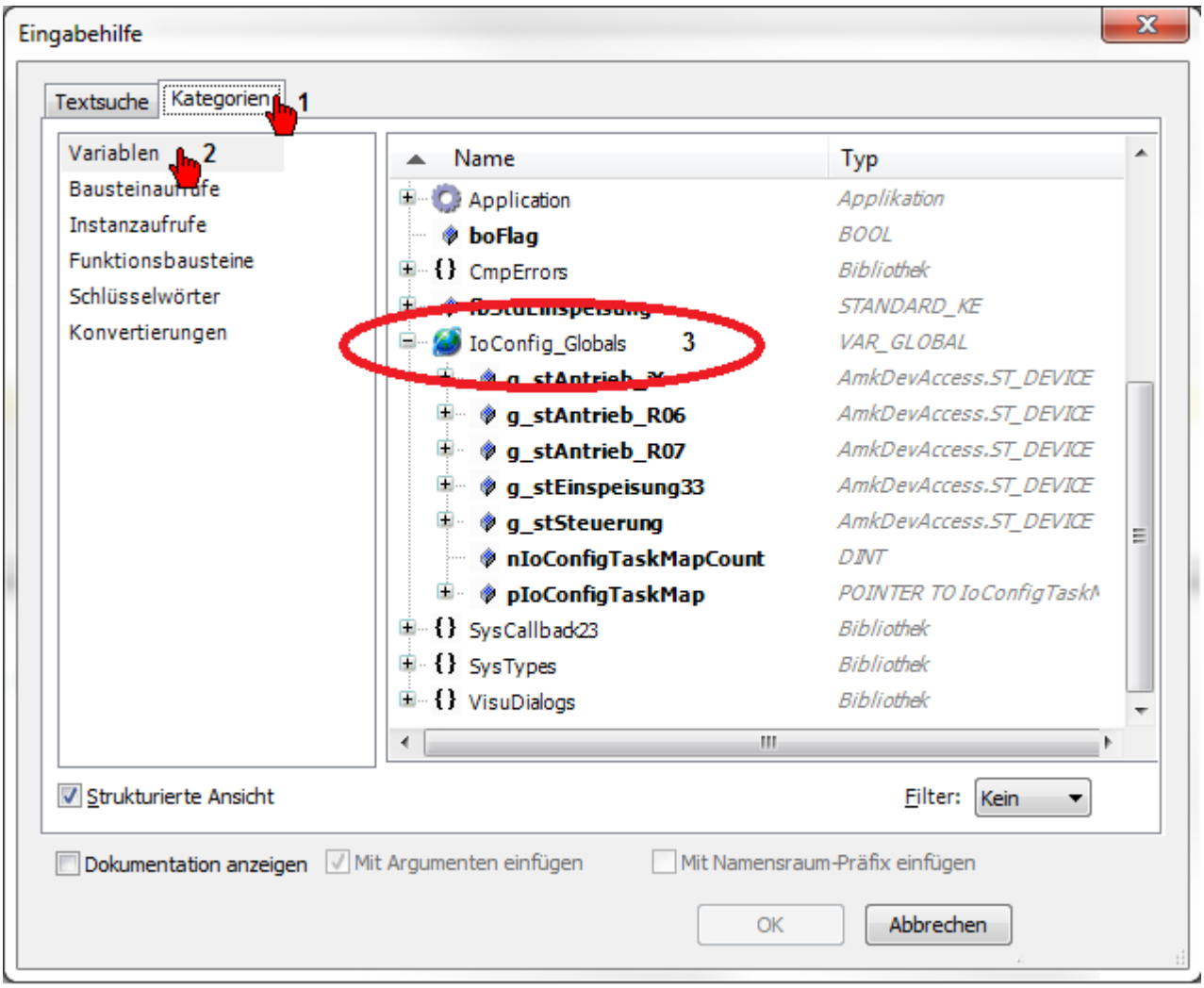
Erstellen Sie die benötigten Variablen,
tragen Sie bei der Variablen 'stDevice' den symbolischen Gerätenamen aus der 'Steuerungskonfiguration' ein.

```
fbStdAntrieb_R06(
    boEnable:= g_boEnable, (* Globales Freigabesignal *)
    boRF:= ,
    boFL:= ,
    boEmergency_Stop:= ,
    boSpeed:= ,
    boHome:= ,
    boPosAbs:= ,
    boPosRel:= ,
    boJogPlus:= ,
    boJogMinus:= ,
    udPosVelocity:= ,
    diPosition:= ,
    lreAccel:= ,
    lreDecel:= ,
    lreJerkStart:= ,
    lreJerkEnd:= ,
    udVelocityJog:= ,
    diSpeedVelocity:= ,
    boEnabAck=> ,
    boSBM=> ,
    boQRF=> ,
    boDone=> ,
    boErr=> ,
    iErrID=> ,
    enErrName=> ,
    boPosBusy=> ,
    boJogBusy=> ,
    boHomeBusy=> ,
    boSpeedBusy=> ,
    stAxisDrive:= g_stAxisDrive_R06, (* AMK Struktur für antriebsspezifische Daten *)
    stDevice:= g_stAntrieb_R06);(* Symbolischer Gerätenamen*)
```

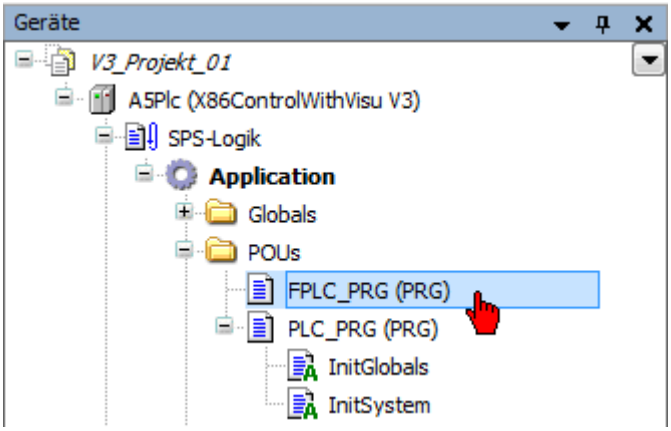
Globale Strukturvariable 'g_stAxisDrive_R06' deklarieren



Sie finden die symbolischen Gerätenamen unter 'IoConfig_Globals'.



Wechseln Sie auf den Reiter 'Geräte'
Öffnen Sie den Programmbaustein 'FPLC_PRG (PRG)'



Klicken Sie im Programmeditor in eine leere Zeile.

```
FPLC_PRG
1  (* functionality:
2     external event-program FPLC_PRG,
3     called by FPLC_TASK in PGT-cycletime (ID2).
4   *)
5  PROGRAM FPLC_PRG
```

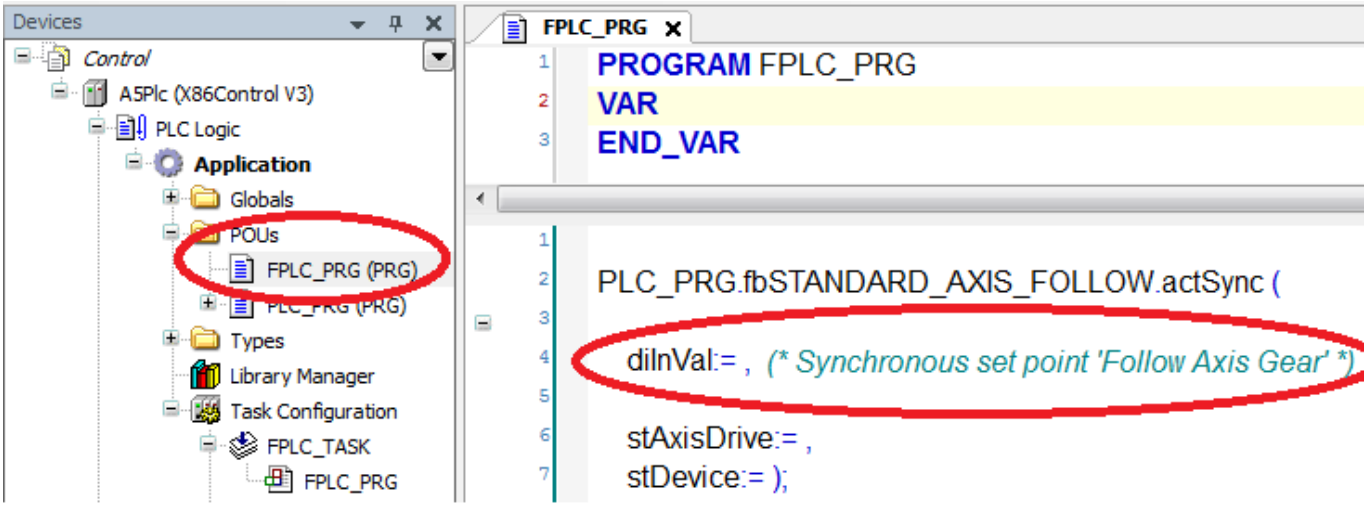
```
1  IF NOT g_boInitOk THEN
2      RETURN; (* Return, if  initialization is not ok *)
3  END_IF
4  (* continue below, if init is done *)
5
6
```

Der Funktionsbaustein 'STANDARD_AXIS' enthält eine synchrone Aktion die mit dem Antrieb synchronisiert ist.
Die synchrone Aktion muss im 'FPLC_PRG' aufgerufen werden. Ohne diesen Aufruf lässt sich der Funktionsbaustein nicht einschalten und die Daten werden nicht aktualisiert.

Erstellen Sie dazu folgenden Programmcode im 'FPLC_PRG'.
PLC_PRG.STANDARD_AXIS .actSync (* im Beispiel PLC_PRG.fbStdAntrieb_R06.actSync *)
(
stAxisDrive := Achstruktur (* im Beispiel g_stAxisDrive_R06 *),
stDevice := Symbolischer Gerätenamen (* im Beispiel g_stAntrieb_R06*)
);

```
PLC_PRG.fbStdAntrieb_R06.actSync
(
    stAxisDrive:= g_stAxisDrive_R06,  (* AMK Struktur für antriebsspezifische Daten *)
    stDevice:= g_stAntrieb_R06  (* Symbolischer Gerätenamen *)
);
```

Synchrone Sollwertvorgabe am Beispiel STANDARD_FOLLOW_AXIS_GEAR (FB)
Die synchronen Sollwertvorgabe erfolgt im FPLC_PRG / *.actSync Aufruf.
Fügen Sie die Input Variable 'diInVal' (Sollwertvorgabe Folgebetrieb) als Übergabevariable ein.

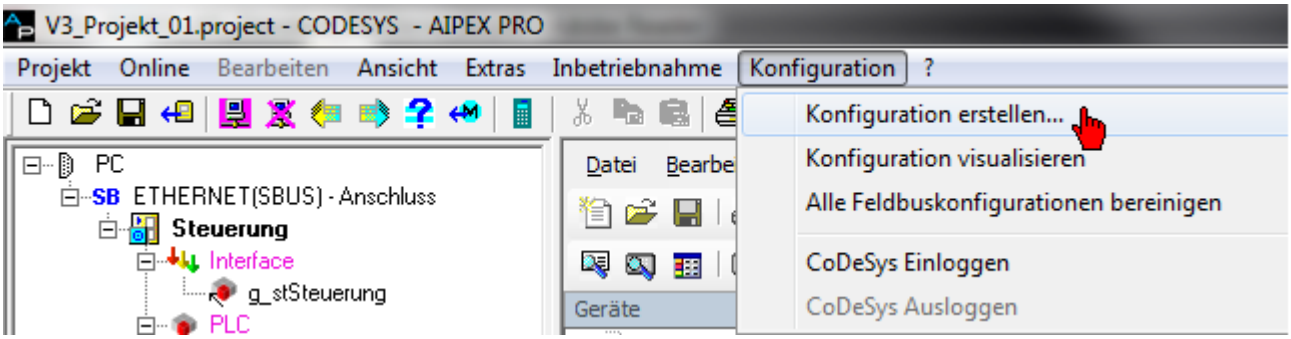


4.3.1.9 PLC Programm und Nachrichtenkonfiguration erstellen

Für den nachfolgenden Schritt muss AIPEX PRO 'Online' sein.

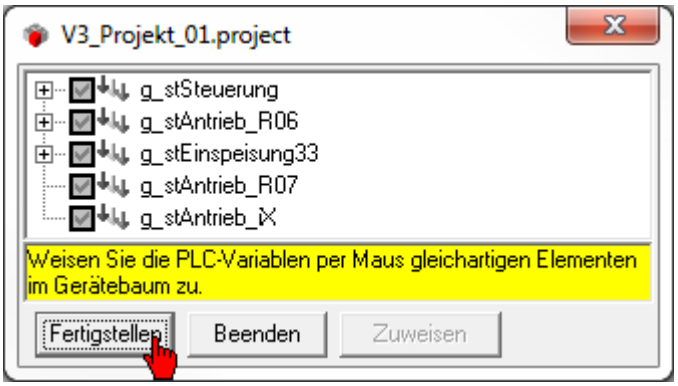
Bevor das PLC Programm auf die Steuerung übertragen wird, muss es fehlerfrei übersetzt und die Nachrichtenkonfiguration erstellt werden.

Starten Sie diesen Vorgang unter '**Konfiguration**' → '**Konfiguration erstellen**'.



g_stSteuerung [Pfeile grau] Der symbolische Gerätenamen ist einem Gerät (Interface) im AIPEX PRO Gerätebaum zugewiesen

g_stSteuerung [Pfeile farbig] Der symbolische Gerätenamen ist nicht zugewiesen. Siehe 'Automatische Buskonfiguration' auf Seite 743.

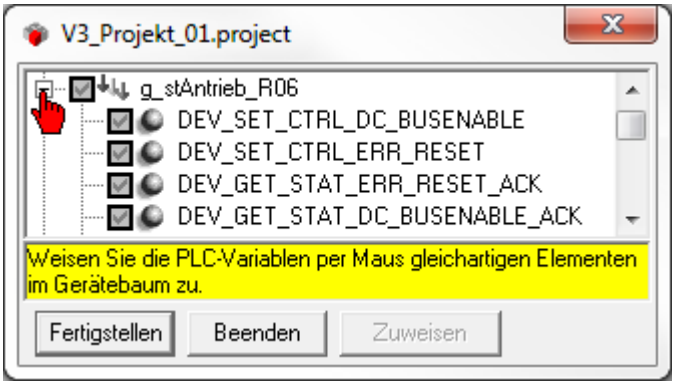


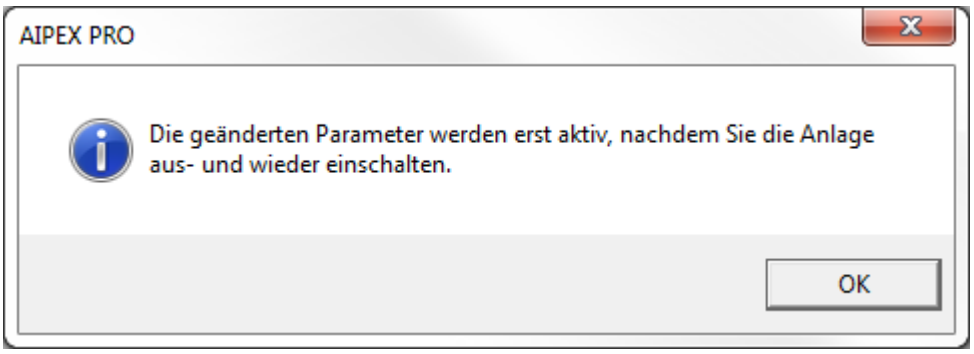
Zusätzliche Information:

Sofern aktiviert, ändert der Konfigurator automatisch die relevanten Parameter. Klicken Sie auf das + Symbol. Sie sehen welche Funktionsbausteine einem symbolischen Gerätenamen zugewiesen sind.

Aktivieren / deaktivieren bzw. Funktionsbaustein spezifische Änderungen:

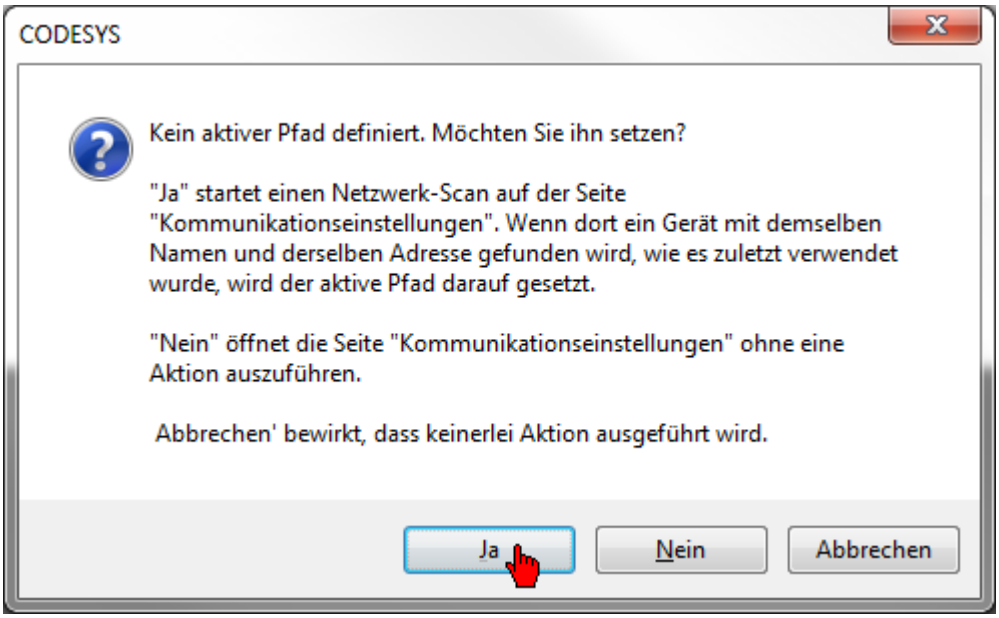
Aktivieren / deaktivieren bzw. Funktionsbaustein spezifische Änderungen: Siehe 'Konfiguration erstellen (projektspezifisch)' auf Seite 113.





Kommunikationseinstellungen

In CODESYS V3 müssen die Kommunikationseinstellungen zwischen PC und Steuerung separat eingestellt werden. Sie werden nicht wie in CODESYS V2 von AIPEX PRO übernommen.

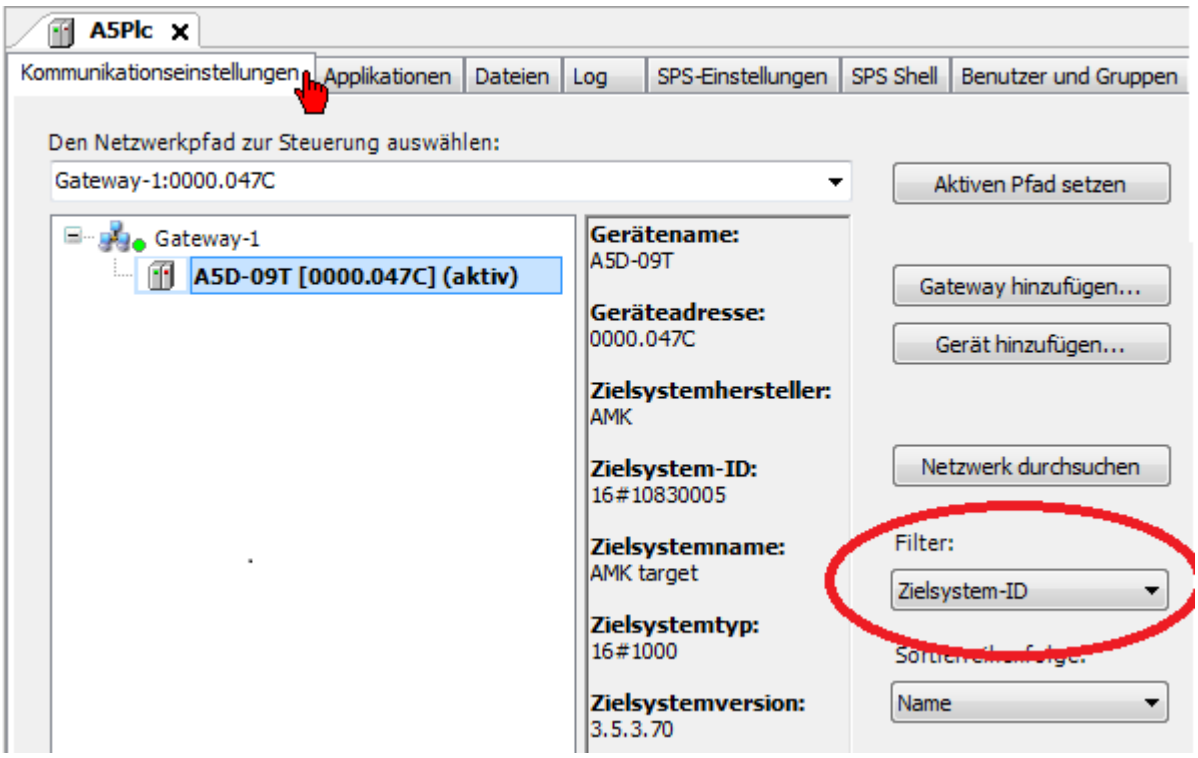


Mit aktivem Filter 'Zielsystem ID' werden automatisch nur diese Steuerungen angezeigt, die zu Ihrem Projekt passen. Treffen Sie anhand der 'Geräteadresse' ihre Auswahl.

Darstellung der TCP/IP Adresse der AMK Steuerung unter CODESYS V3. Nur die letzten 4 Ziffern sind entscheidend.

Beispiel, die Steuerung befindet sich in einem Netzwerk mit der TCP/IP Adresse 172.16.xxx.xxx.

CODESYS V3 **Geräteadresse**	**AMK Steuerung** **TCP/IP Adresse**
0000.047C (hex)	172.16.4.124 (dez)
0001.A000.0462 (hex)	172.16.4.098 (dez)



Erklärung Geräteeigenschaften: Siehe 'Geräteidentifizierung' auf Seite 752.

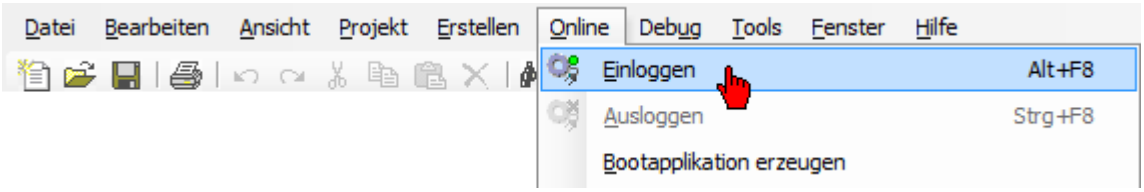
4.3.1.10 PLC Projekt testen

Ziel: Aktivierung der Motorregelung mit Hilfe des 'Deklarationseditor im Onlinemodus'.

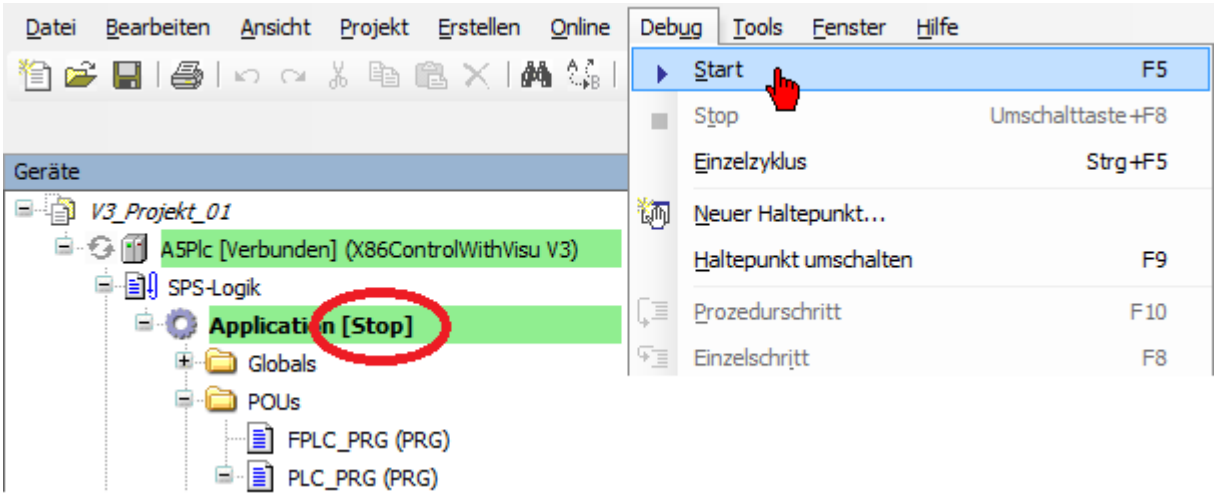
Voraussetzung:

- Erstinbetriebnahme Motor abgeschlossen
- EF/EF2 (STO) Hardwaresignale vorhanden
- Eventuell vorhandene Notauskreise geschlossen

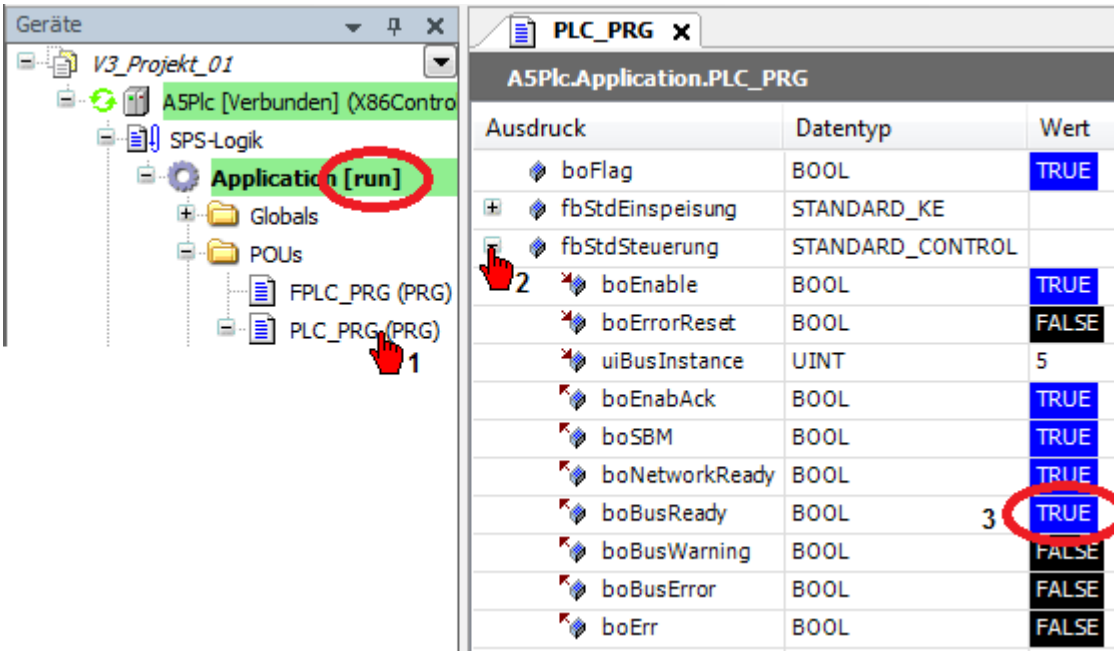
Einloggen



Starten Sie die PLC

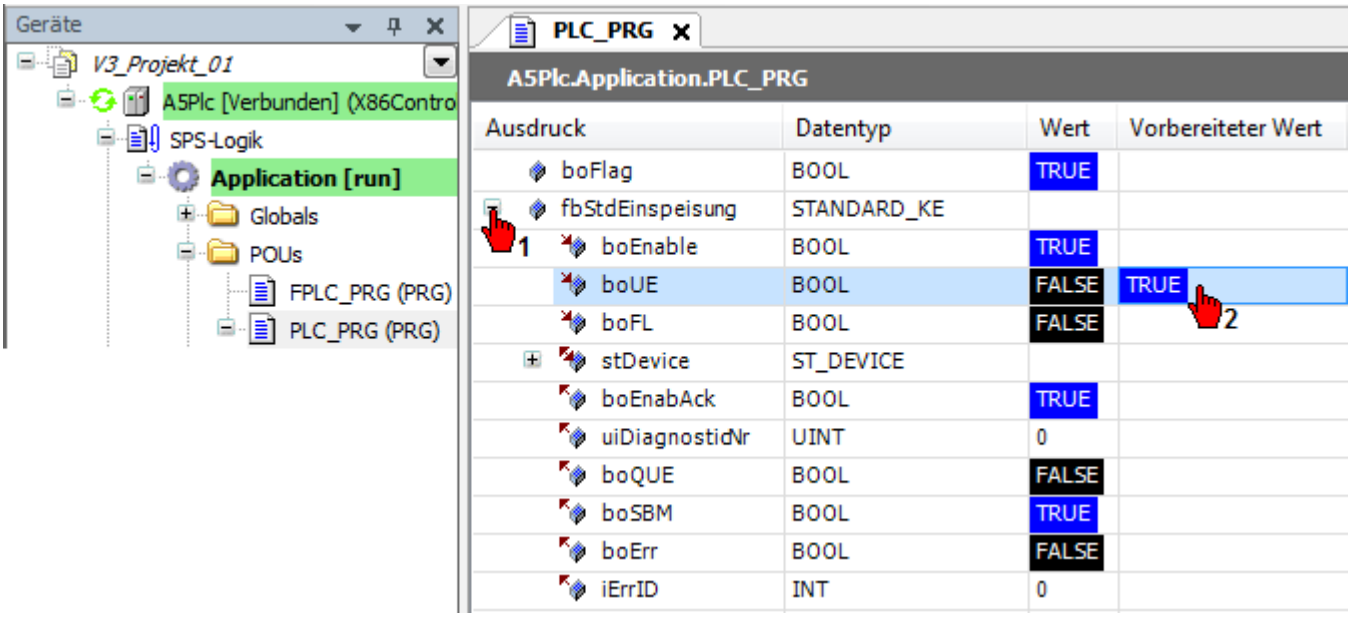


Der Status 'Application' wechselt auf [run]. Öffnen Sie den Programmbaustein 'PLC_PRG'. In den Eigenschaften des Funktionsbausteines 'fbStdSteuerung' sehen Sie jetzt (boBusReady = TRUE) ob die AMK Geräte fehlerfrei initialisiert sind und der EtherCAT Bus den 'Operational mode' erreicht hat.

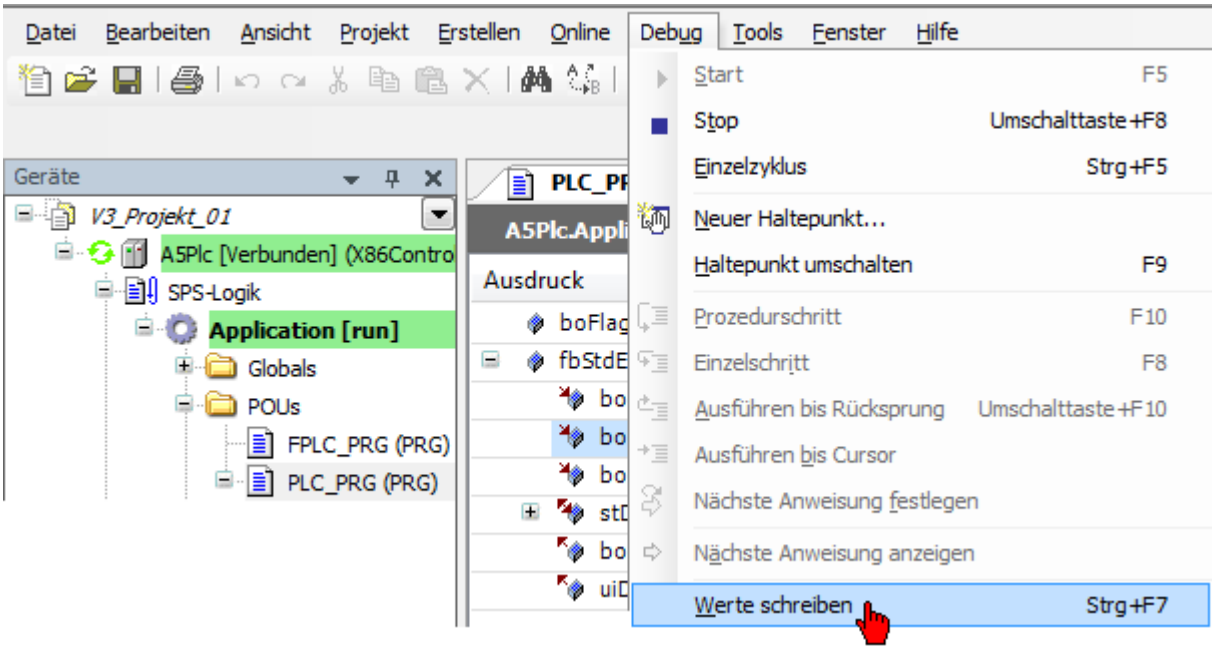


Laden Sie die Zwischenkreisspannung, indem Sie die Eingangsvariable 'boUE' des Funktionsbausteines 'fbStdEinspeisung' auf 'TRUE' setzen.

(Quittierungssignal für ein fehlerfreies Laden 'boQUE = TRUE')

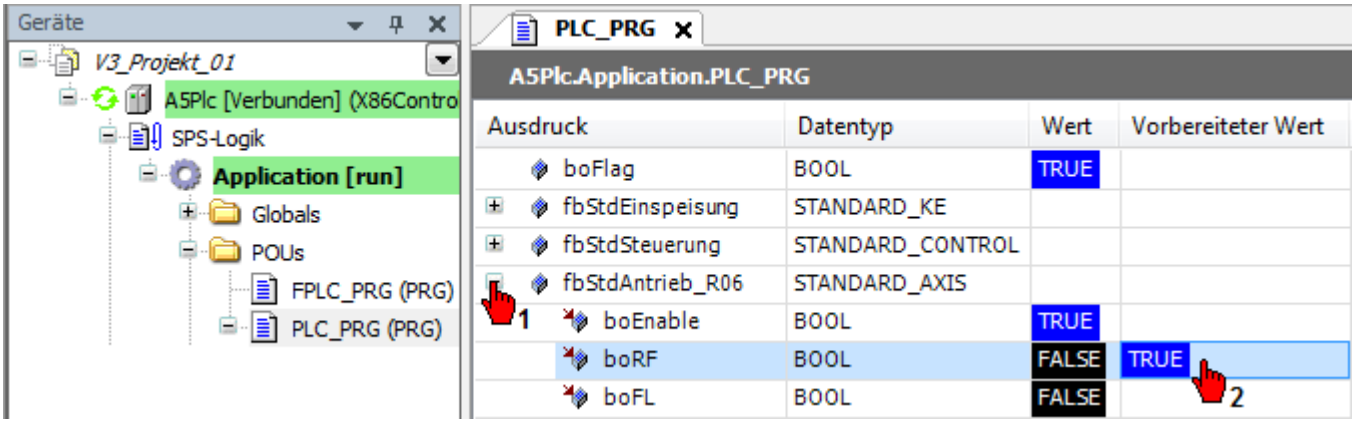


'boUE = TRUE' wird mit 'Werte schreiben [Strg + F7]' übernommen.



Aktivieren Sie die Motorregelung, indem Sie die Eingangsvariable 'boRF' des Funktionsbausteines 'fbStdAntrieb_R06' auf 'TRUE' setzen.

'boRF = TRUE' wird mit 'Werte schreiben [Strg + F7]' übernommen.

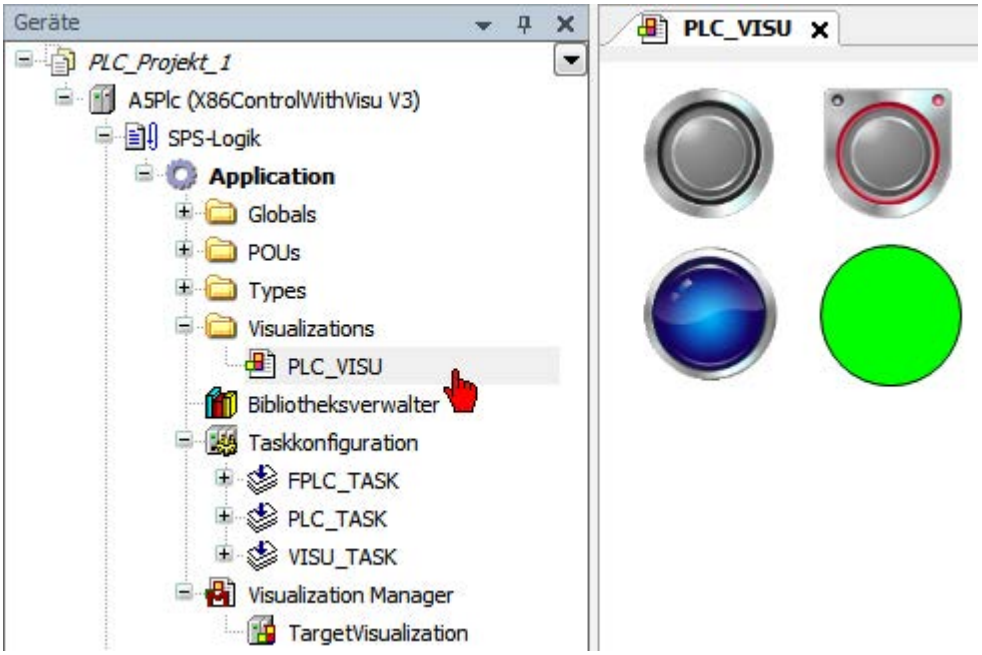


Quittierungssignal für ein fehlerfreies Aktivieren der Motorregelung 'boQRF = TRUE'

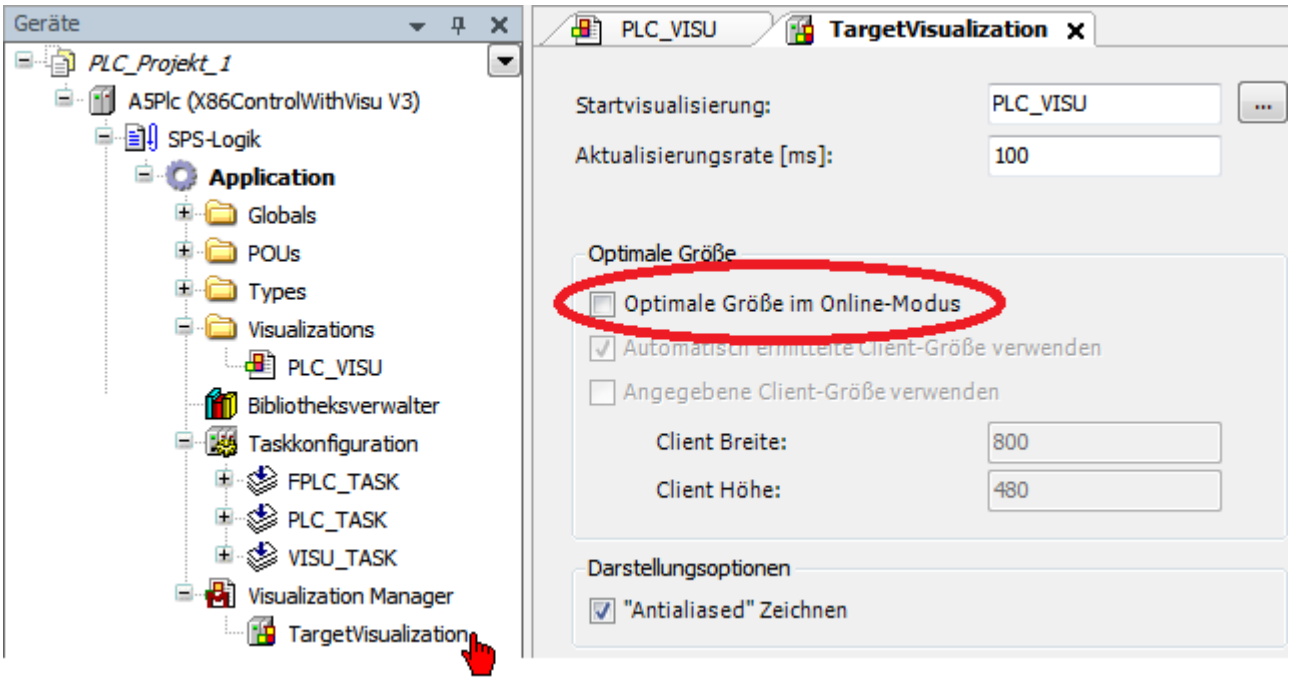
Der Antrieb ist bereit um Sollwerte zu verarbeiten.

4.3.1.11 Visualisierung

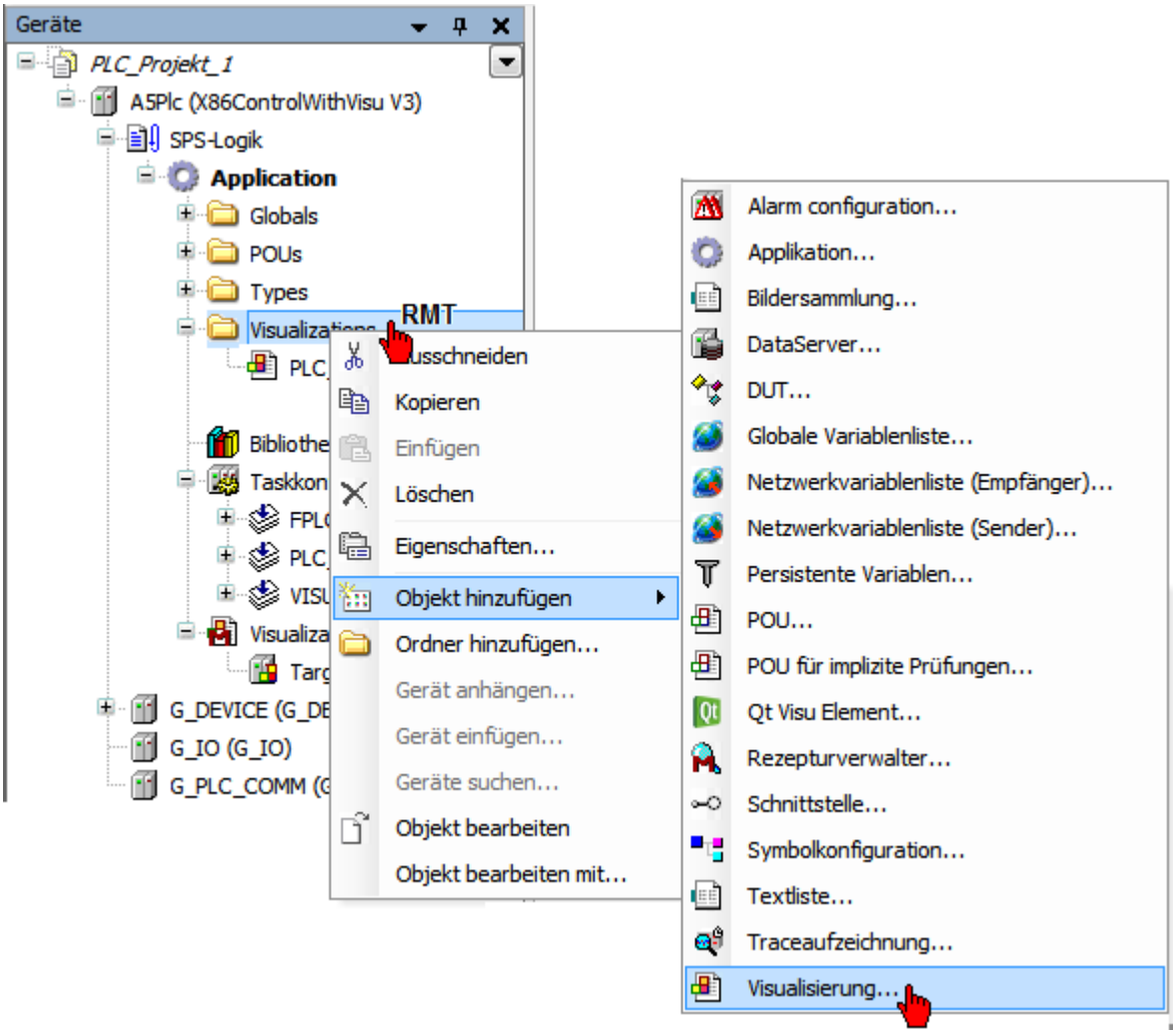
PLC_VISU ist immer die erste Visualisierungsseite die aufgerufen wird.



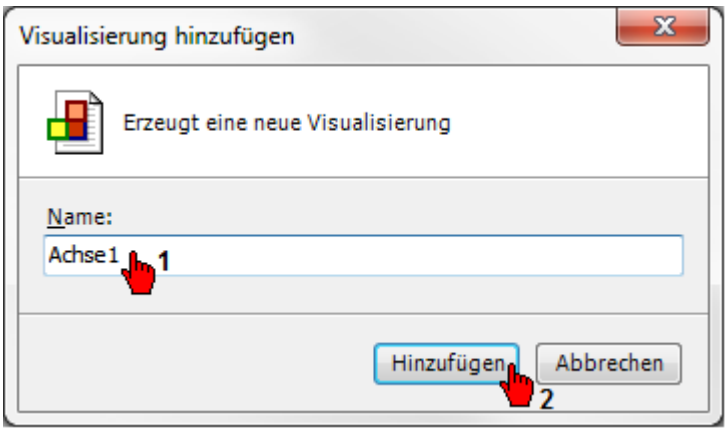
Die Grundeinstellungen werden in der Seite 'TargetVisualization' eingestellt.



Leere Seite anlegen



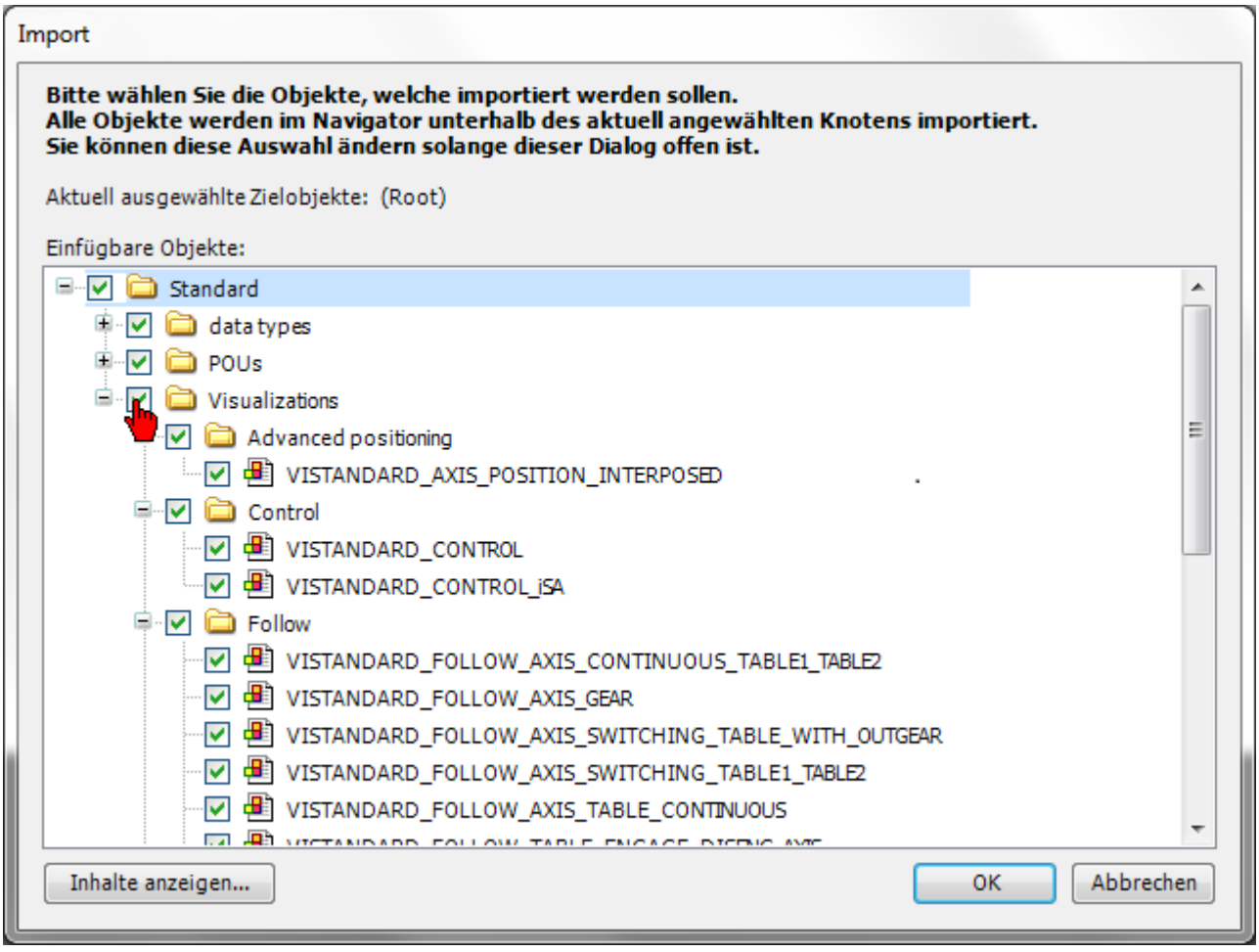
Vergeben Sie einen Namen



4.3.1.11.1 Visualisierung Standardbausteine

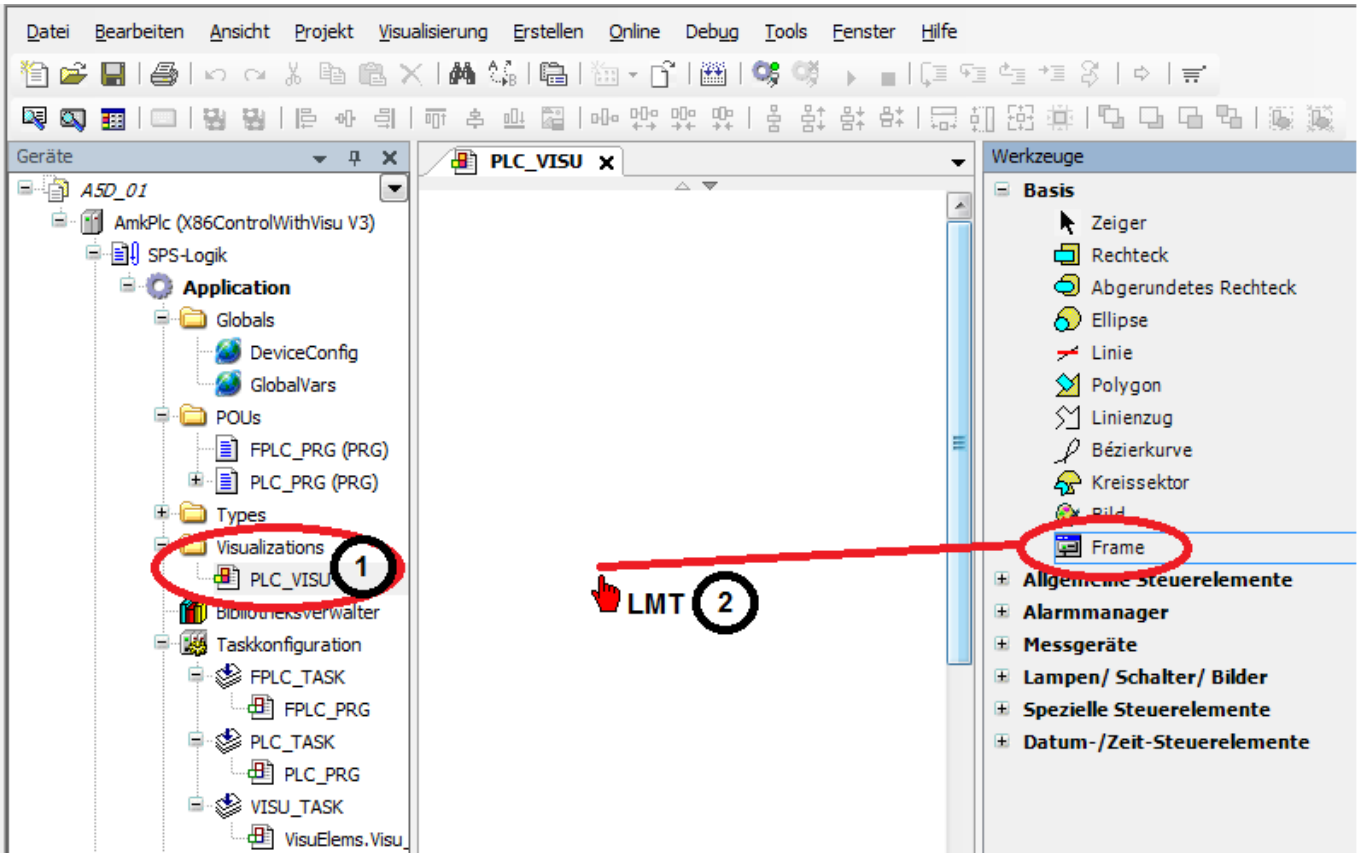
Die Visualisierungen der Standardbausteine werden bei CODESYS V3 mit Hilfe der Import Funktion in das aktuelle PLC Projekt eingefügt. CODESYS Menü 'Projekt' → 'Importieren'.

C:\Programme (x86)\3S CODESYS\AmkAddition\Imports\Application\V3_import.

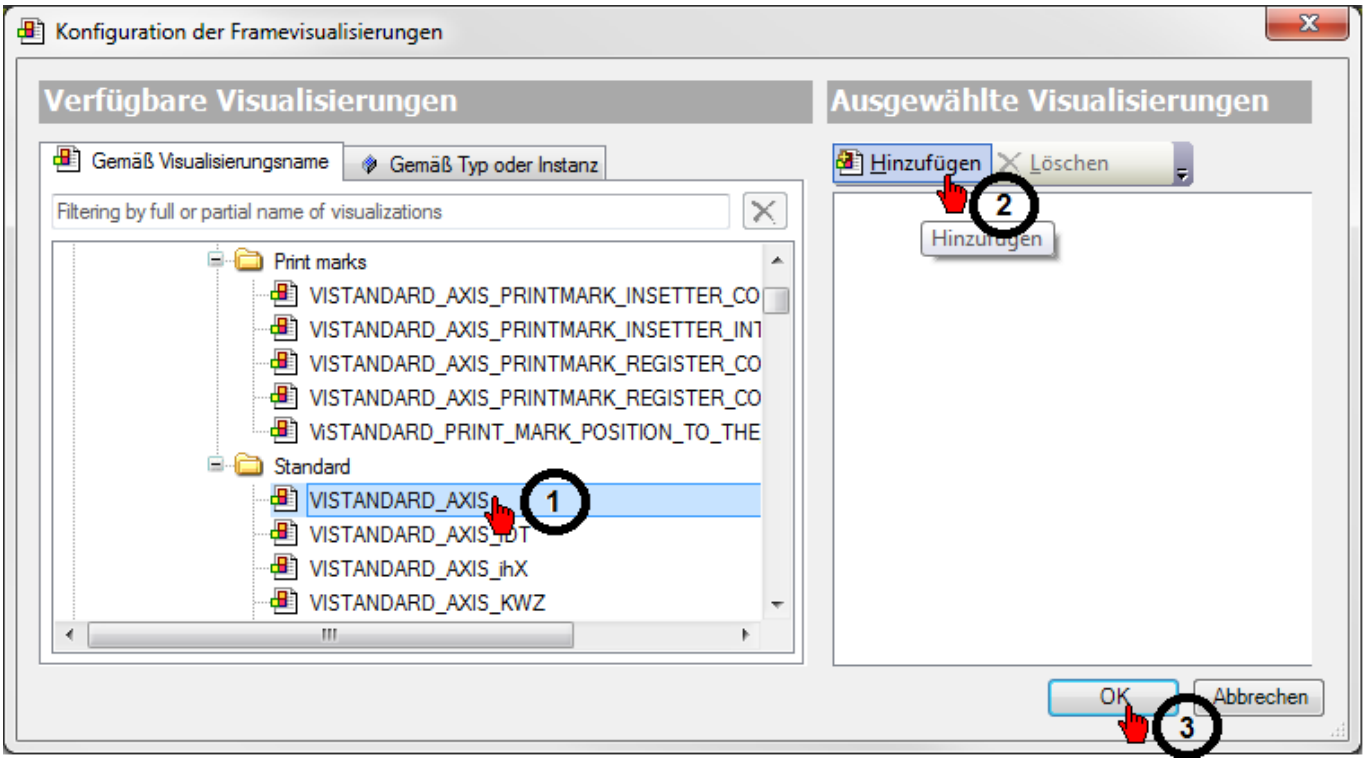


(Visualisierungen der AFL Applikationsbausteine sind in einer zusätzlichen Bibliothek enthalten. Siehe 'Visualisierung von AFL Bausteinen' auf Seite 843.

Die Visualisierungen werden mit Vis-Werkzeug 'Frames' integriert.

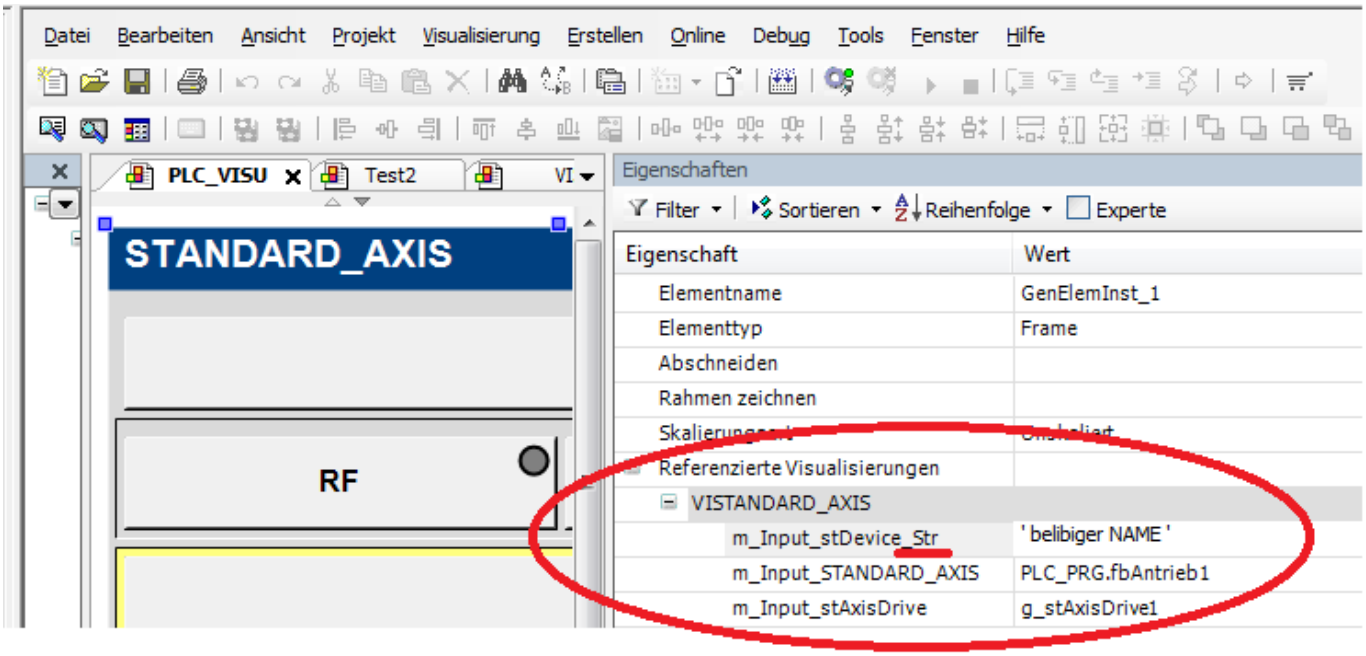


Wählen Sie durch die Taste 'Hinzufügen' die gewünschte Visualisierung aus.

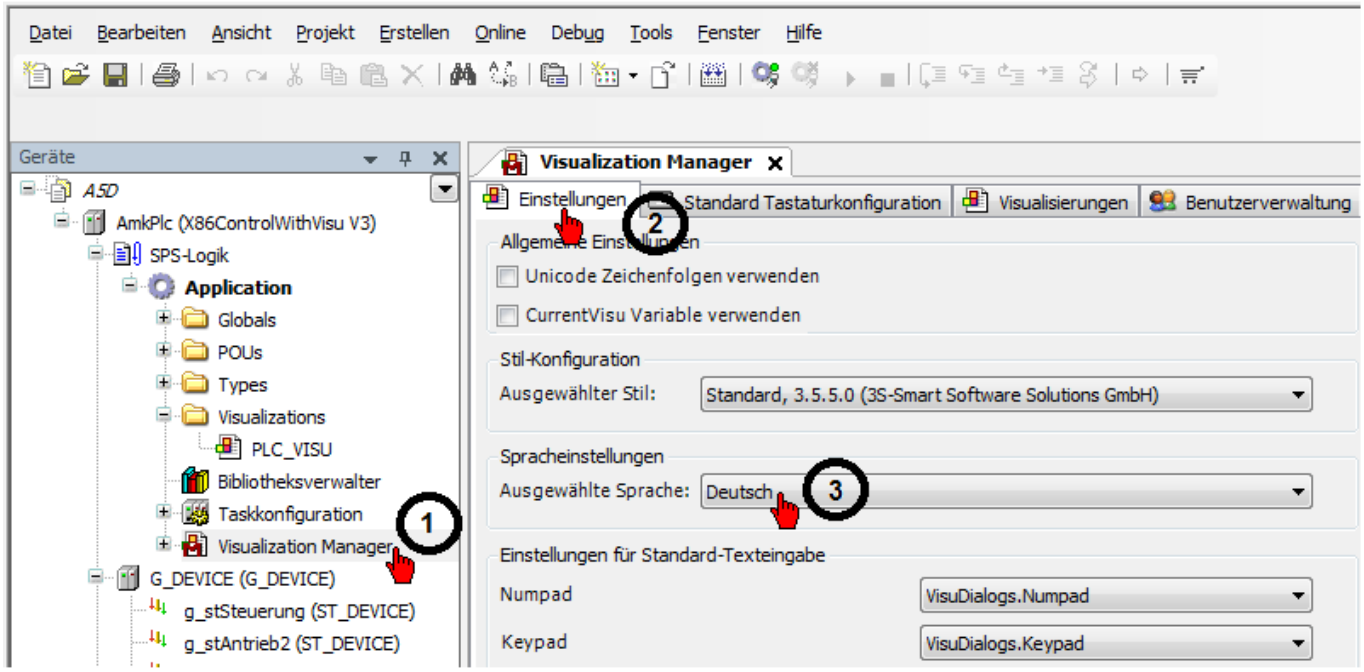


Die im Frame eingefügte Visualisierung muss mit dem instanzierten AFL Standardbaustein und der dazugehörigen Struktur ST_AXIS_DRIVE verknüpft werden.

Mit m_Input_stDevice_Str wird ein Name (STRING) angegeben der in der Visualisierung angezeigt wird.



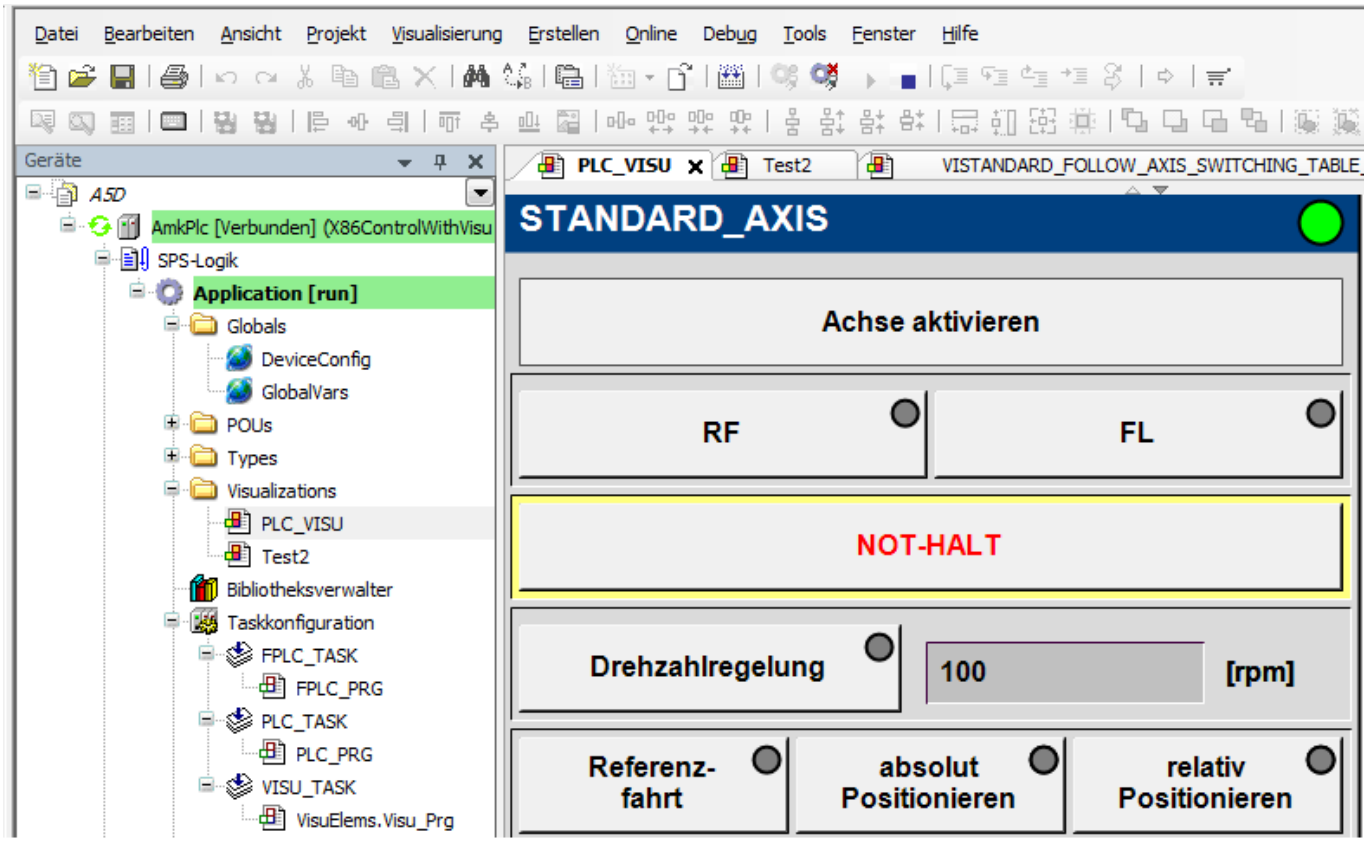
Spracheinstellung Visualisierung.



Sprachabhängig Texte werden erst im Online Modus angezeigt.

Die angezeigten Texte (Deutsch und Englisch) sind in einer Textliste hinterlegt: Ordner 'POUs' → 'Standard' → 'Visualizations' → 'Standard' → 'TextListStandard'.

Anzeige im 'RUN Modus'



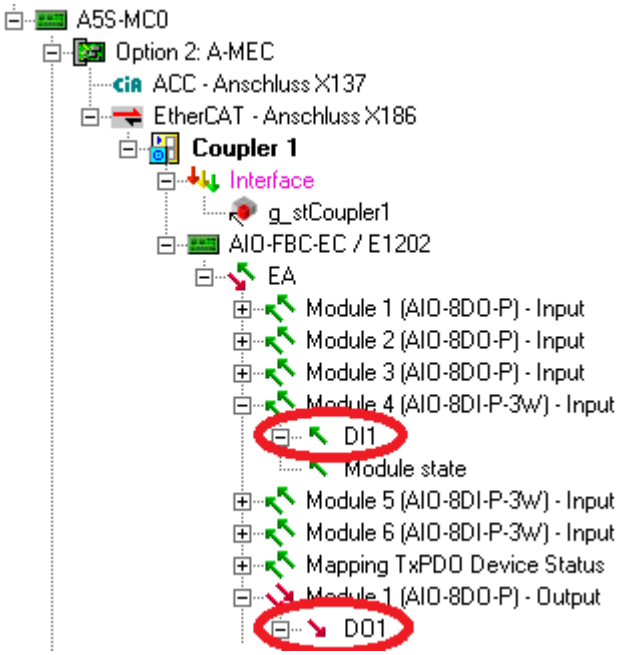
4.3.1.12 Funktionen

4.3.1.12.1 I/O Zugriff

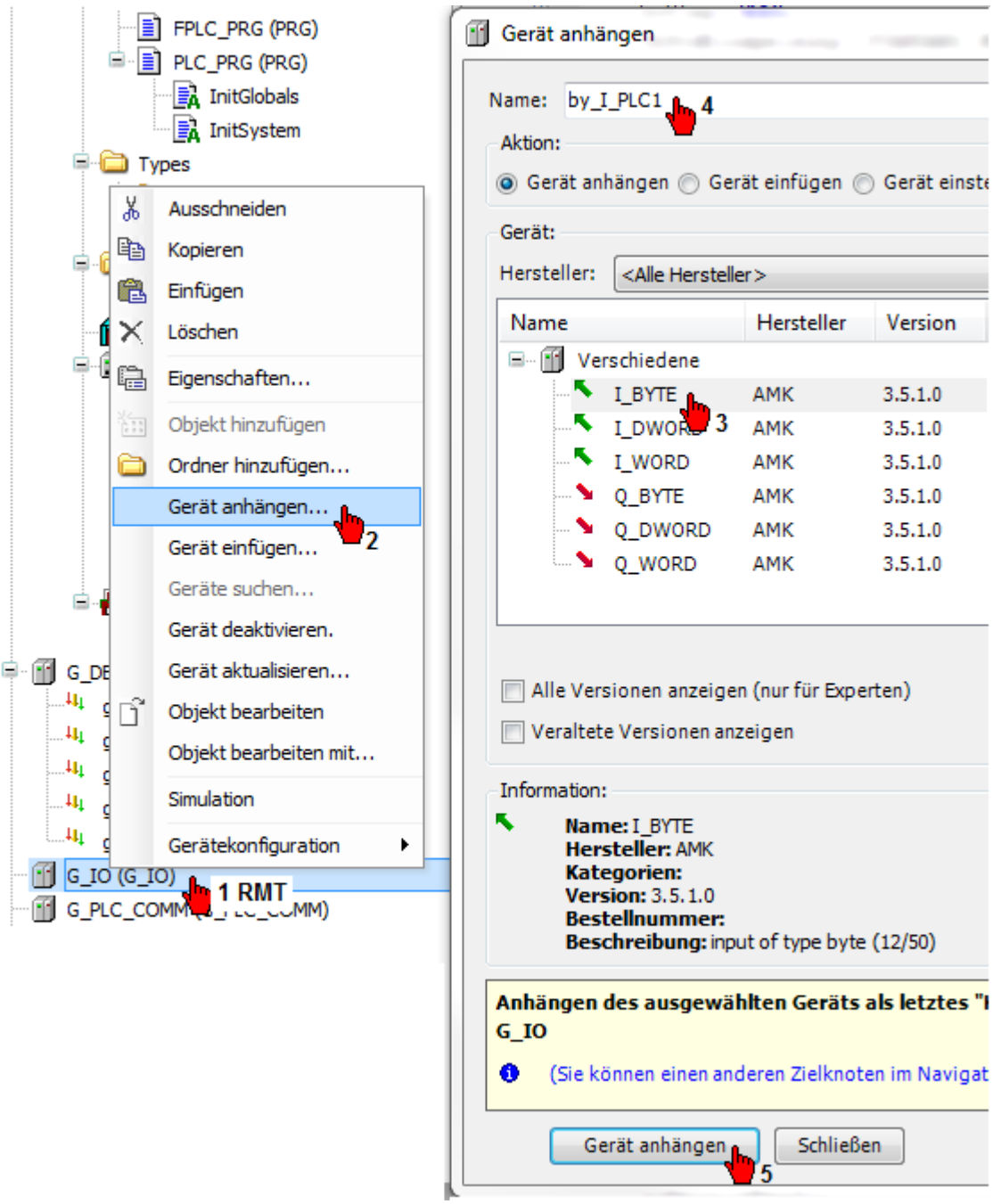
4.3.1.12.1.1 Externe I/O-Klemme (asynchron)

Einbinden einer externen I/O-Klemme. Das Beispiel zeigt einen AIO-FBC-EC Feldbuskoppler mit drei AIO-8DO-P und drei AIO-8DI-P-3W Modulen. Der Feldbuskoppler wird von AIPEX PRO automatisch erkannt.

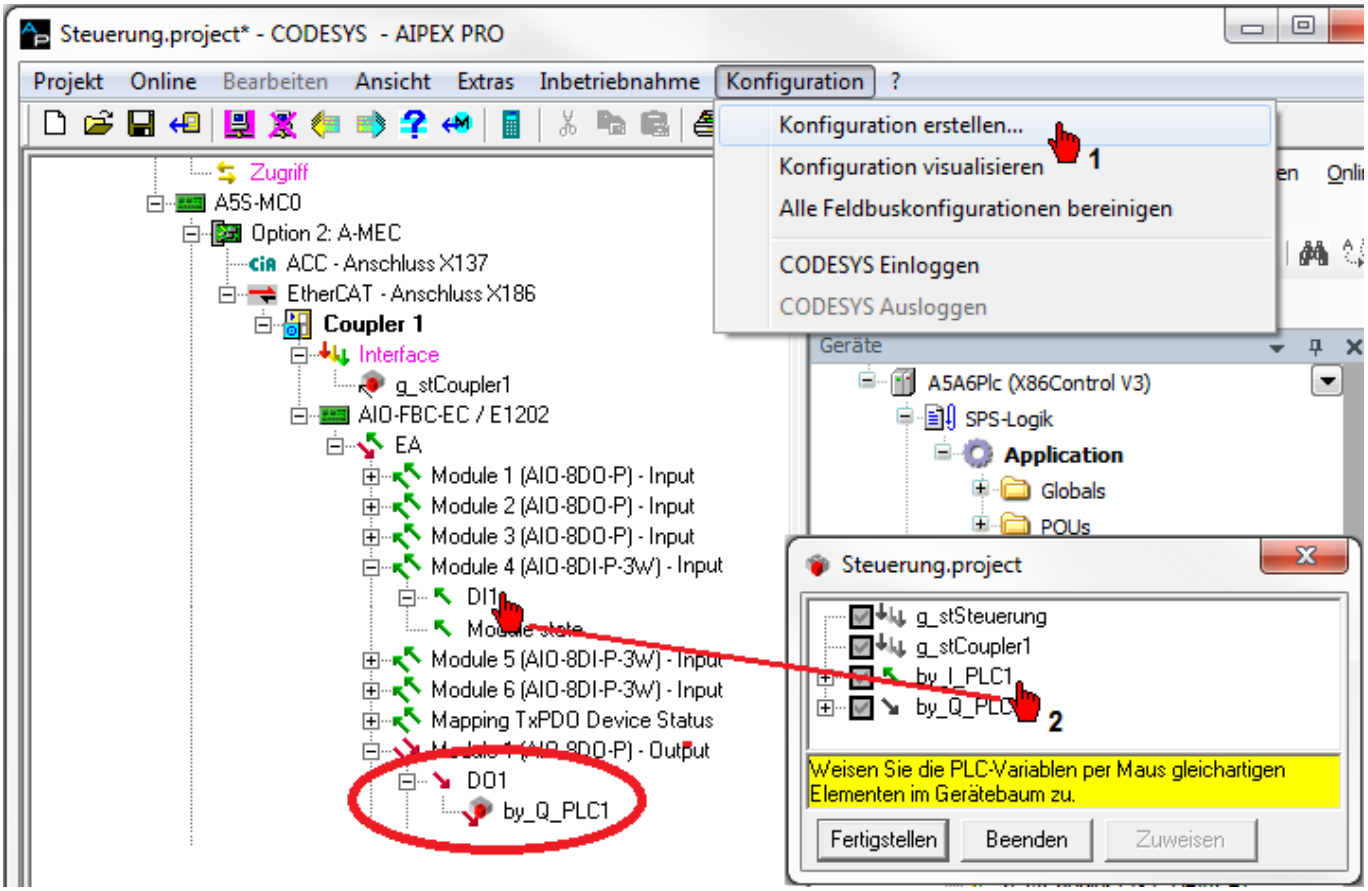
Die I/Os werden innerhalb eines Programmzyklus gelesen (am Programmanfang) und geschrieben (am Programmende).



Für jeden Ein- und Ausgangsblock muss in CODESYS eine globale I/O-Variable angelegt werden.
Im Beispiel wird ein Input Byte by_I_PLC1 und ein Output Byte by_Q_PLC1 angelegt.



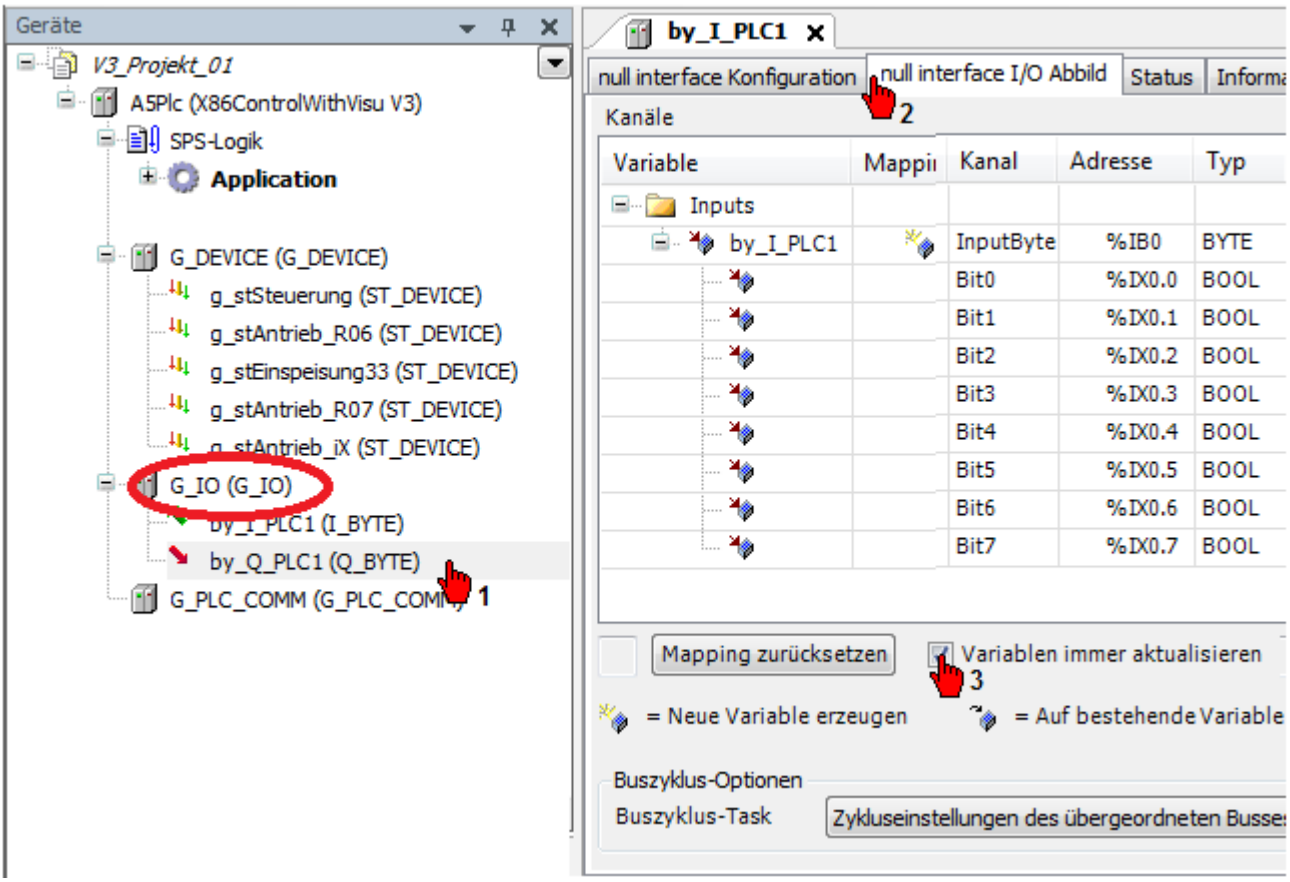
Rufen Sie **'Konfiguration erstellen'** auf. Weisen Sie die symbolischen Gerätenamen den I/O Schnittstellen im AIPEX PRO Gerätebaum zu.



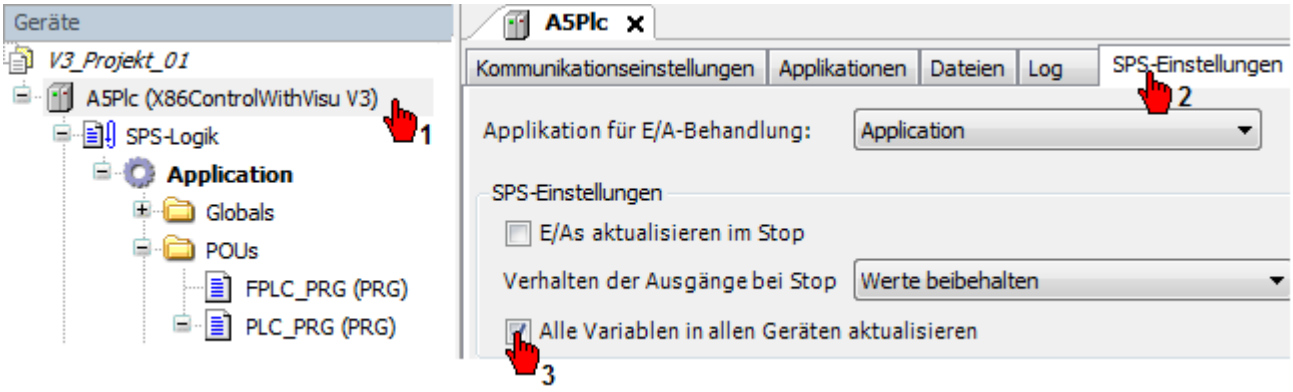
Testen

Damit die I/O Variablen im PLC Projekt aktualisiert werden, stehen folgende Möglichkeiten zu Verfügung:

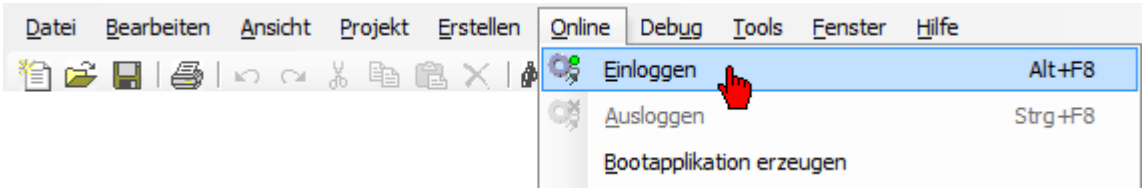
- Verknüpfen Sie die symbolische Gerätenamen mit PLC Variablen
- Setzen Sie die Option 'Variable immer aktualisieren' (siehe Bild)



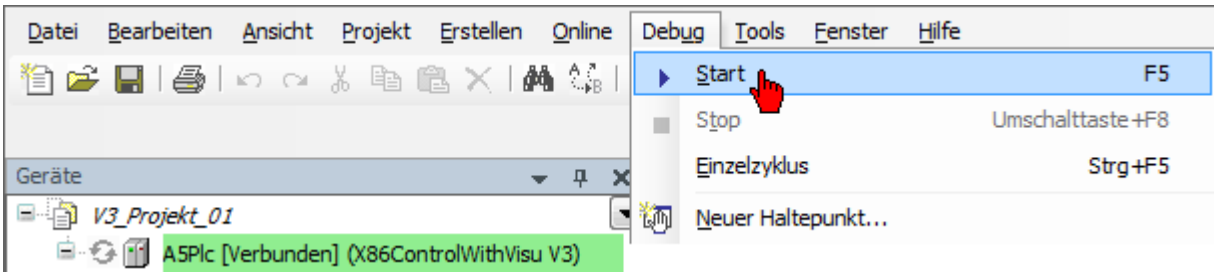
Alternativ können Sie die Option 'Alle Variablen in allen Geräten aktualisieren' aktivieren.



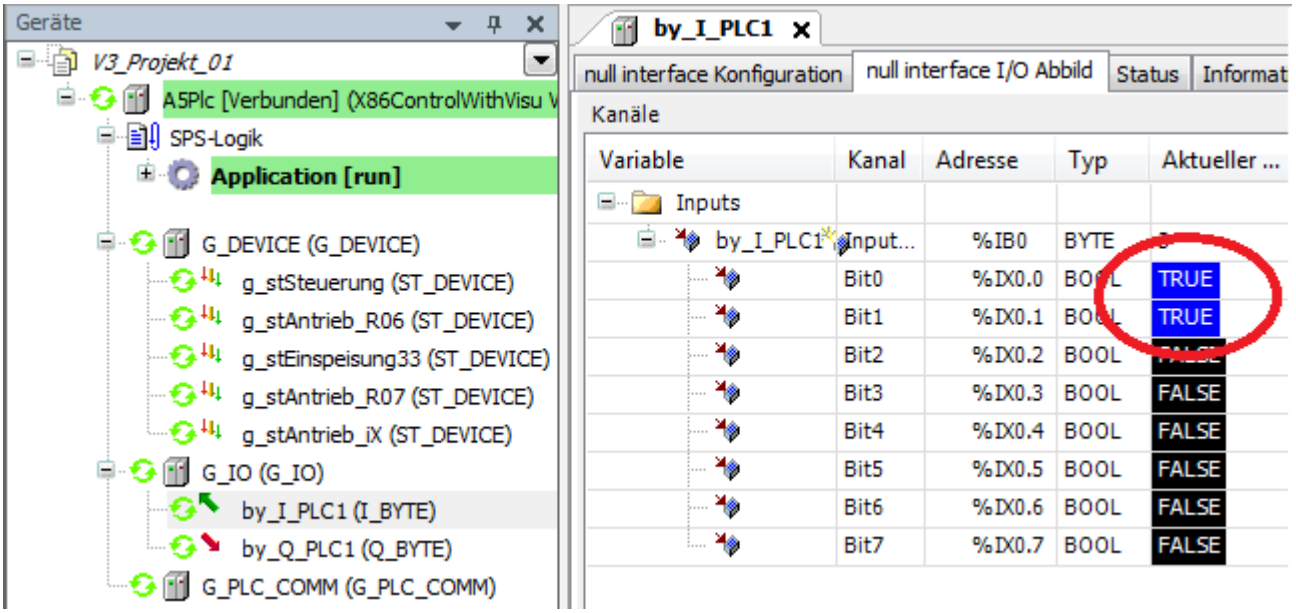
Rufen Sie die Funktion **'Einloggen'** auf (Code wird automatisch erzeugt).



Starten Sie die Applikation auf der Steuerung.



Setzen Sie zum Testen binäre Eingänge auf der Steuerung.

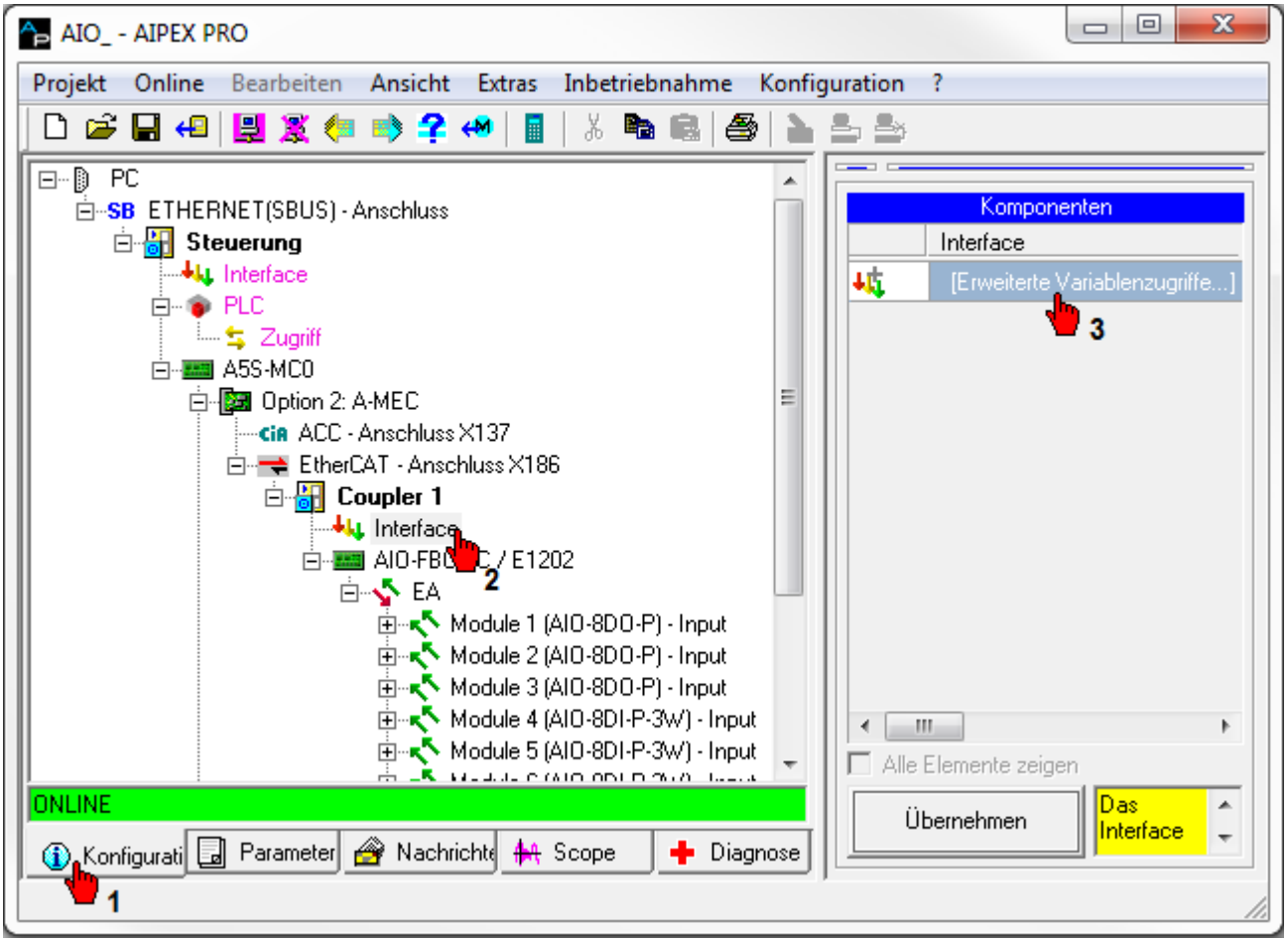


4.3.1.12.1.2 Externe I/O-Klemme (zyklisch)

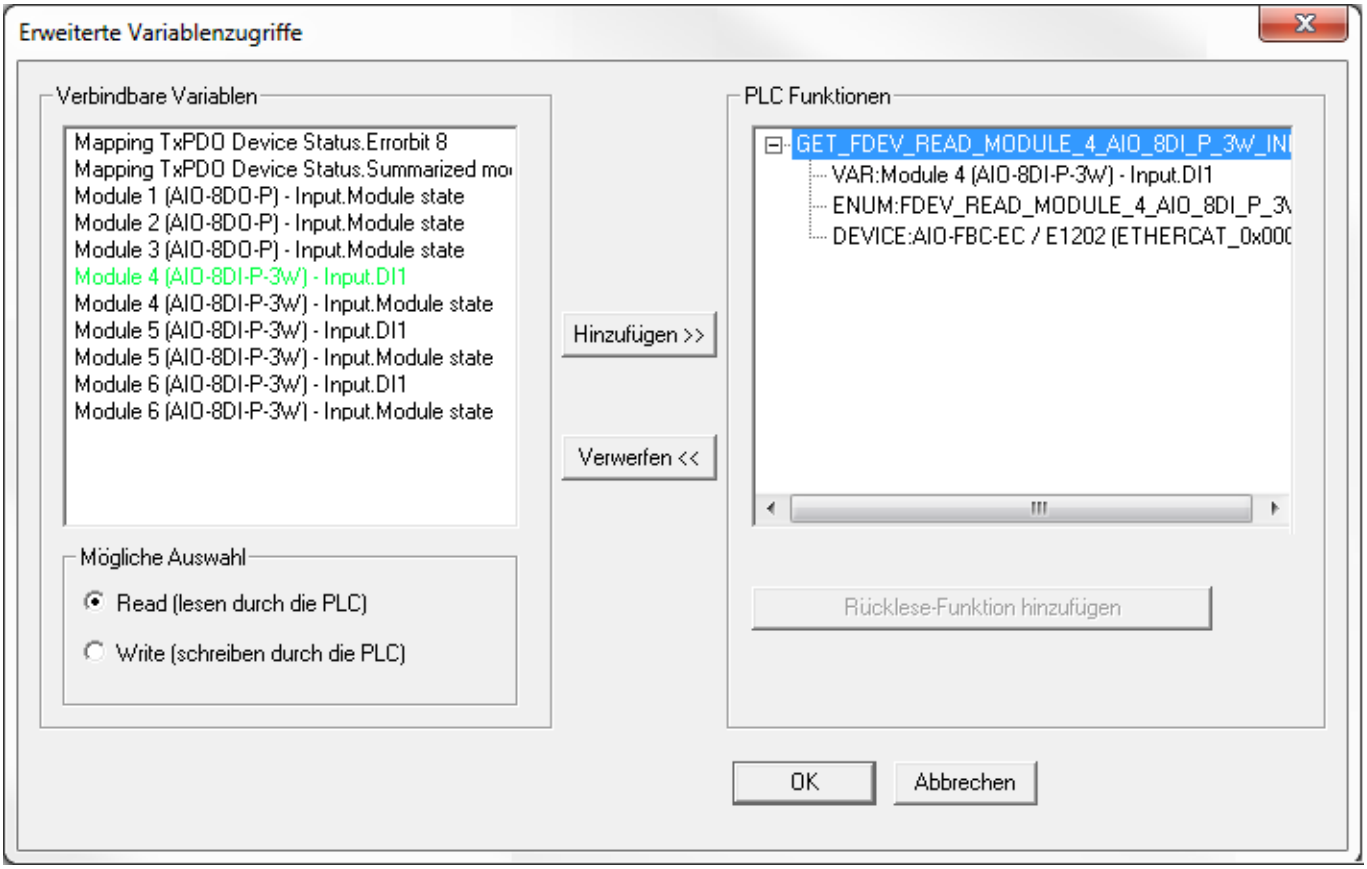
Einbinden einer externen I/O Klemme. Das Beispiel zeigt einen AIO-FBC-EC Feldbuskoppler mit drei AIO-8DO-P und drei AIO-8DI-P-3W Modulen. Der Feldbuskoppler wird von AIPEX PRO automatisch erkannt.

Die Vorgehensweise basiert auf dem Prinzip 'Erweiterte Variablenzugriffe'. Siehe 'Erweiterte Variablenzugriffe' auf Seite 719.

Das zyklische Lesen und Schreiben eines I/O Abbildes findet über das Geräteinterface statt.
Legen Sie dazu an ihre Klemme ein **'Interface'** an und Öffnen Sie das Fenster 'Erweiterte Variablenzugriffe'.

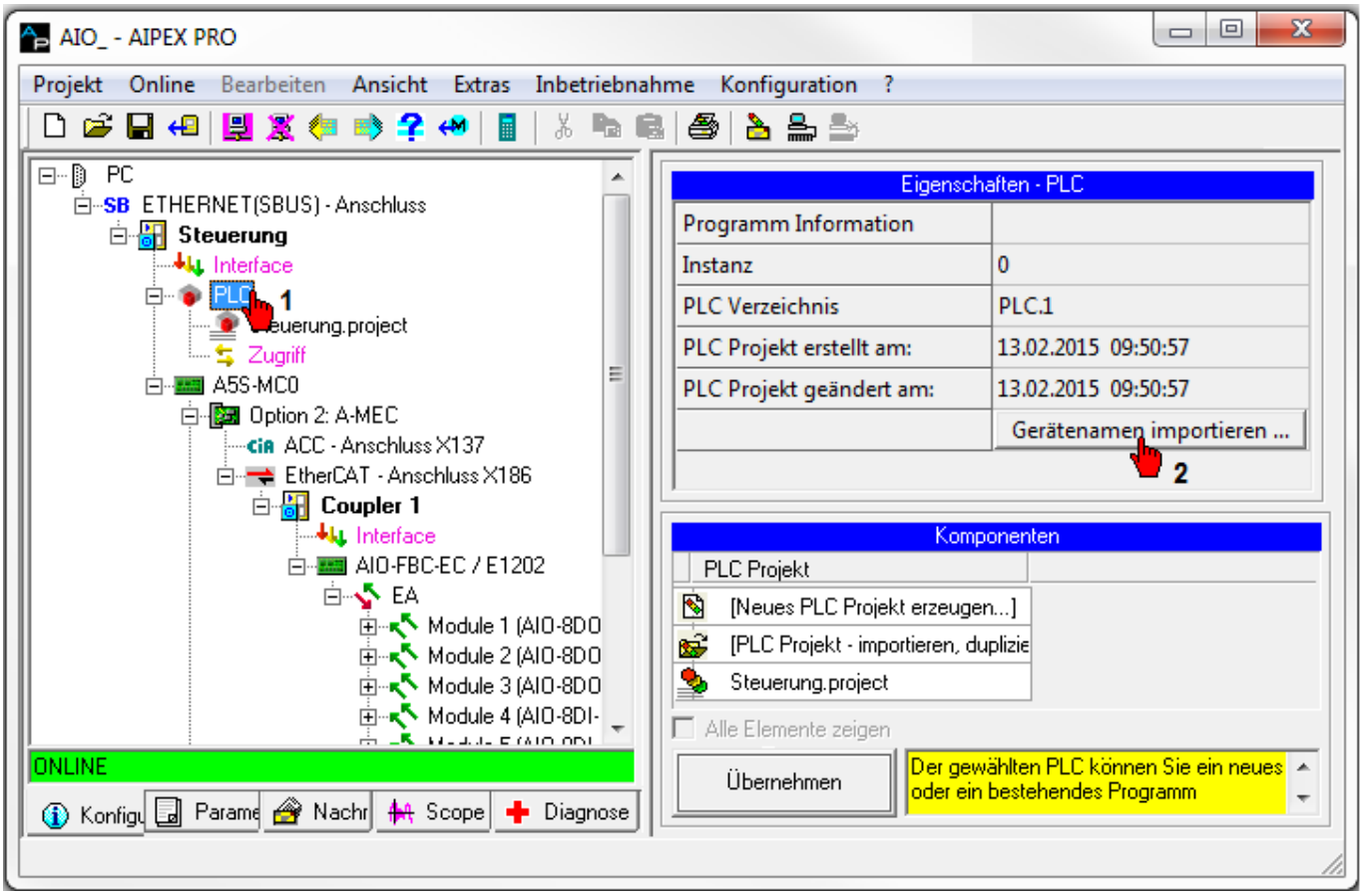


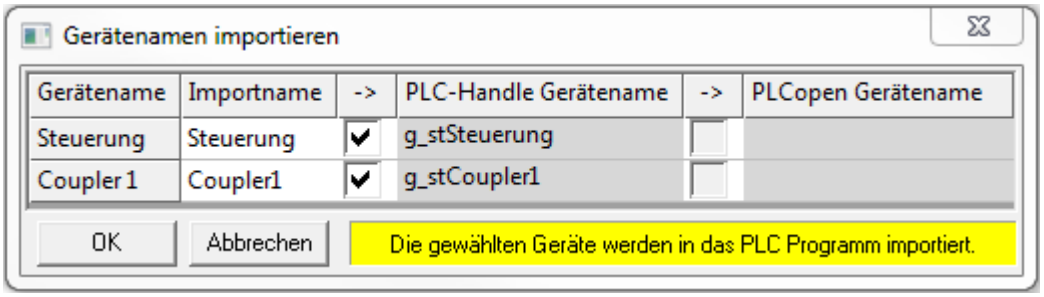
Die mögliche Auswahl der verbindbaren Variablen ist in der dazugehörigen Geräte -XML Datei hinterlegt.



Generieren Sie in gleicher weise ein 'Write' Funktionsbaustein für das Ausgangsmodul (AIO-8DO-P).

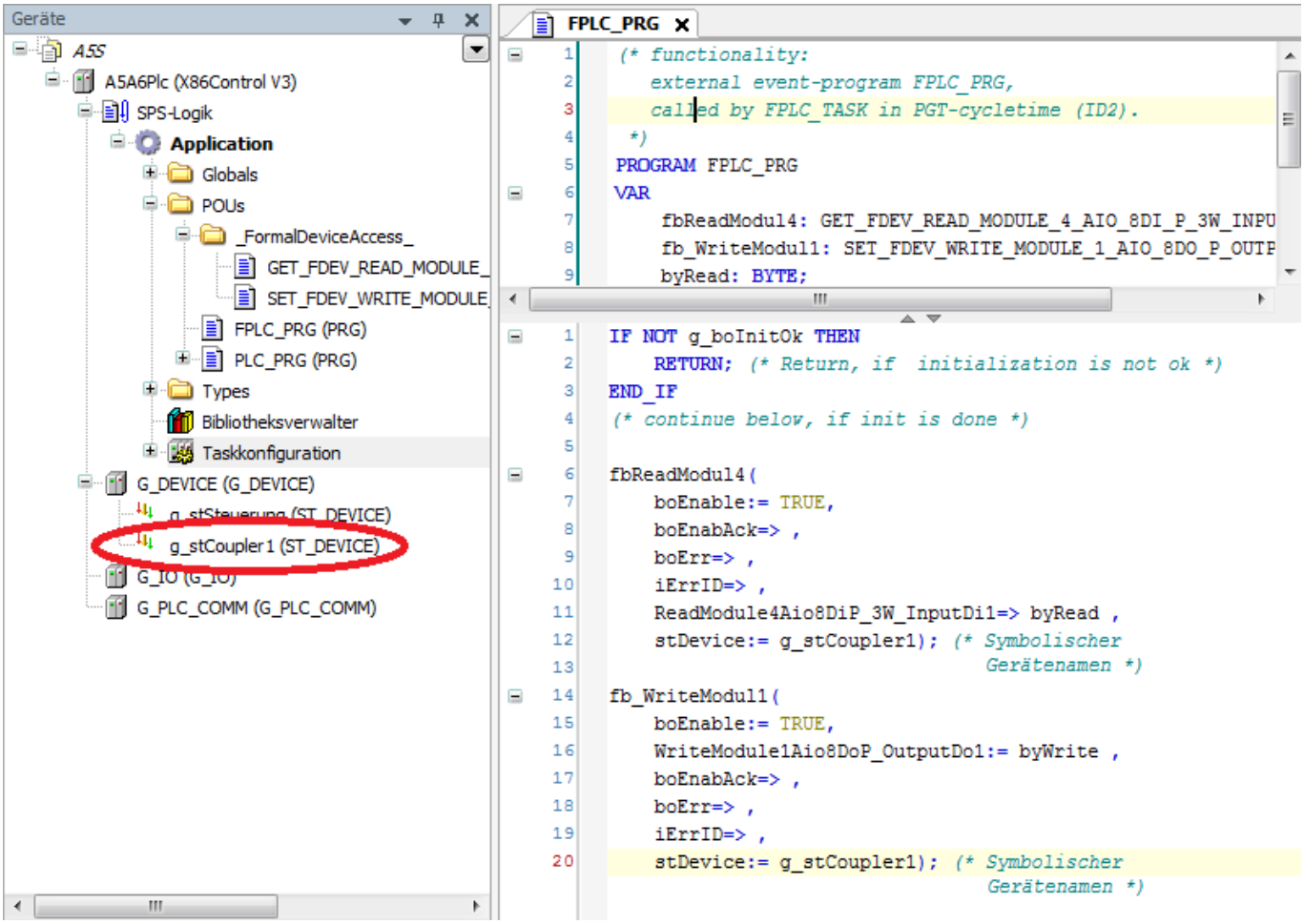
Übernehmen Sie die Gerätenamen in das PLC Projekt. Dafür können Sie die automatische Funktion **'Gerätenamen importieren'** verwenden. Alternative können Sie die Klemmen manuell in der PLC Steuerungskonfiguration (G_DEVICE (G_DEVICE) anlegen und anschließend dem Icon 'Interface' zuweisen.





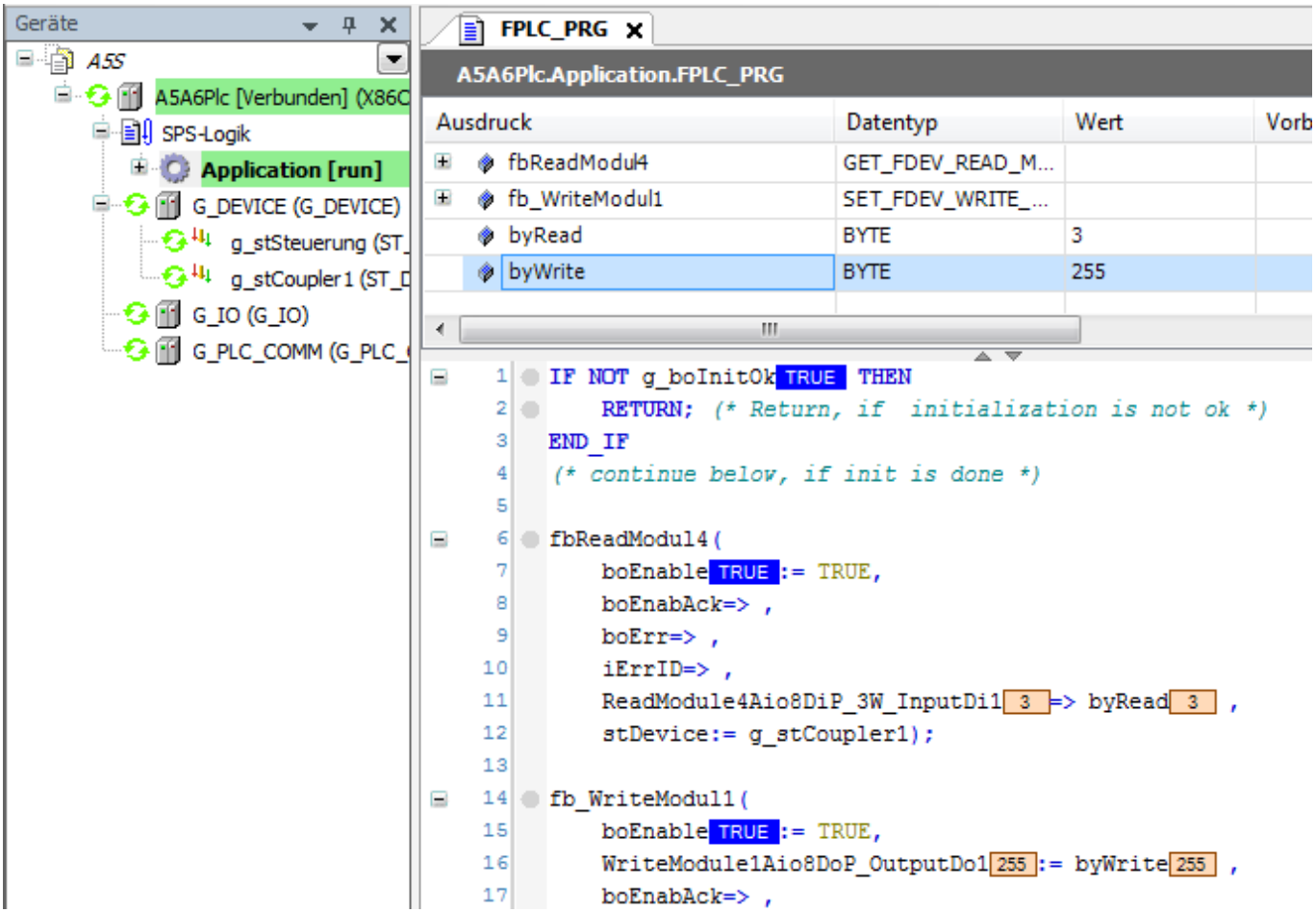
Öffnen Sie das CODESYS Projekt. (Steuerung.project)

Fügen Sie in der Echtzeit Ebene (FPLC_PRG) jeweils eine Instanz der generierten Bausteine 'GET_FDEV_READ_MODULE_4_AIO_8DI_P_3W_INPUT_DI1' und 'SET_FDEV_WRITE_MODULE_1_AIO_8DO_P_OUTPUT_DO1' an.
Verbinden Sie jeweils die Variable 'stDevice' mit dem jeweiligen symbolischen Gerätenamen.



Testen

Erstellen Sie das Projekt: Siehe 'PLC Programm und Nachrichtenkonfiguration erstellen' auf Seite 695.



4.3.1.12.1.3 Time Stamp EAs und XFC Klemmen

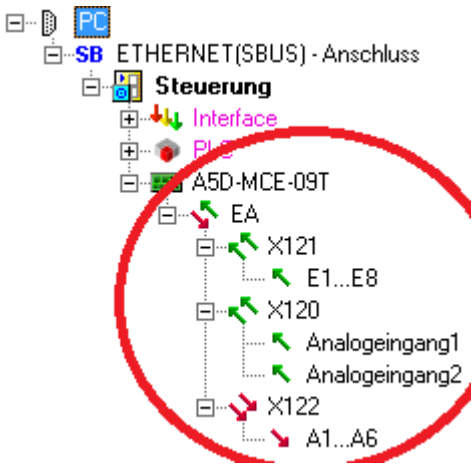
Das Kapitel beschreibt den Zugriff auf die Time Stamp EAs der A5 Steuerung mit der 'E/A Erweiterung'. Die angegebenen PLC Funktionsbausteine werden für die Time Stamp EAs der A5 Steuerung und für die XFC Klemmen EL 1252 und EL 2252 verwendet. Die Funktionalität der XFC Klemmen ist kompatibel mit den Time Stamp EAs der A5 Steuerung.

PLC Funktionsbausteine für den Time Stamp EA Zugriff werden mit der AMK Bibliothek 'AmkDevAccess' zu Verfügung gestellt.
Siehe: Ordner: 'POUs' → 'DeviceAccessSync' → 'TimeStamp'
Funktionsbaustein GET_TS_INPUTS (Time Stamp Eingänge)
Funktionsbaustein SET_TS_OUTPUTS (Time Stamp Ausgänge)

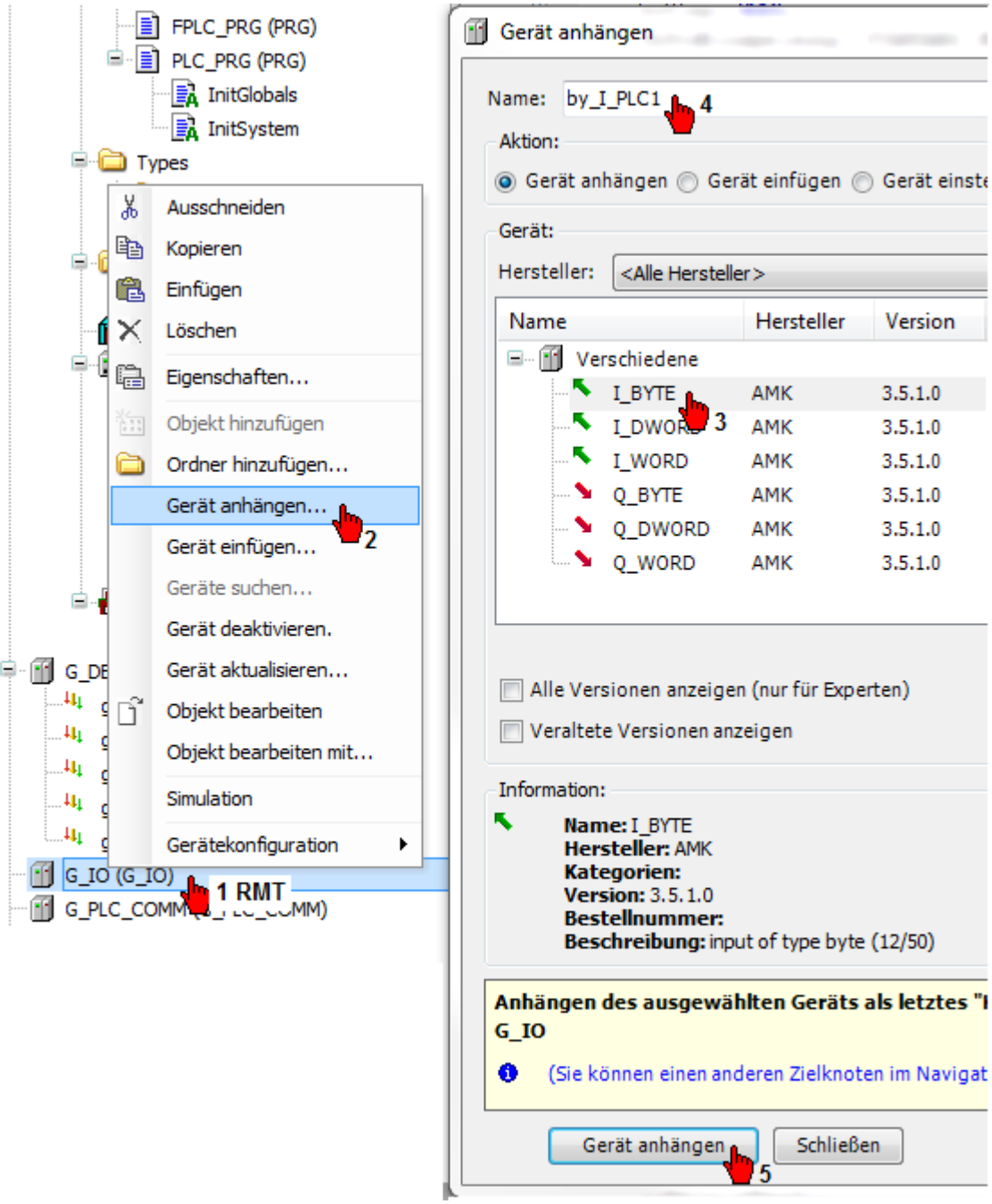
Beschreibung der Messeingänge und Time Stamp Ausgänge der A5 Steuerung:
Siehe Produktbeschreibung Steuerungen A4 / A5 / A6 Kapitel: 'Messeingänge' und Kapitel 'Time Stamp Ausgänge'

4.3.1.12.1.4 EA Klemme Steuerung (asynchon)

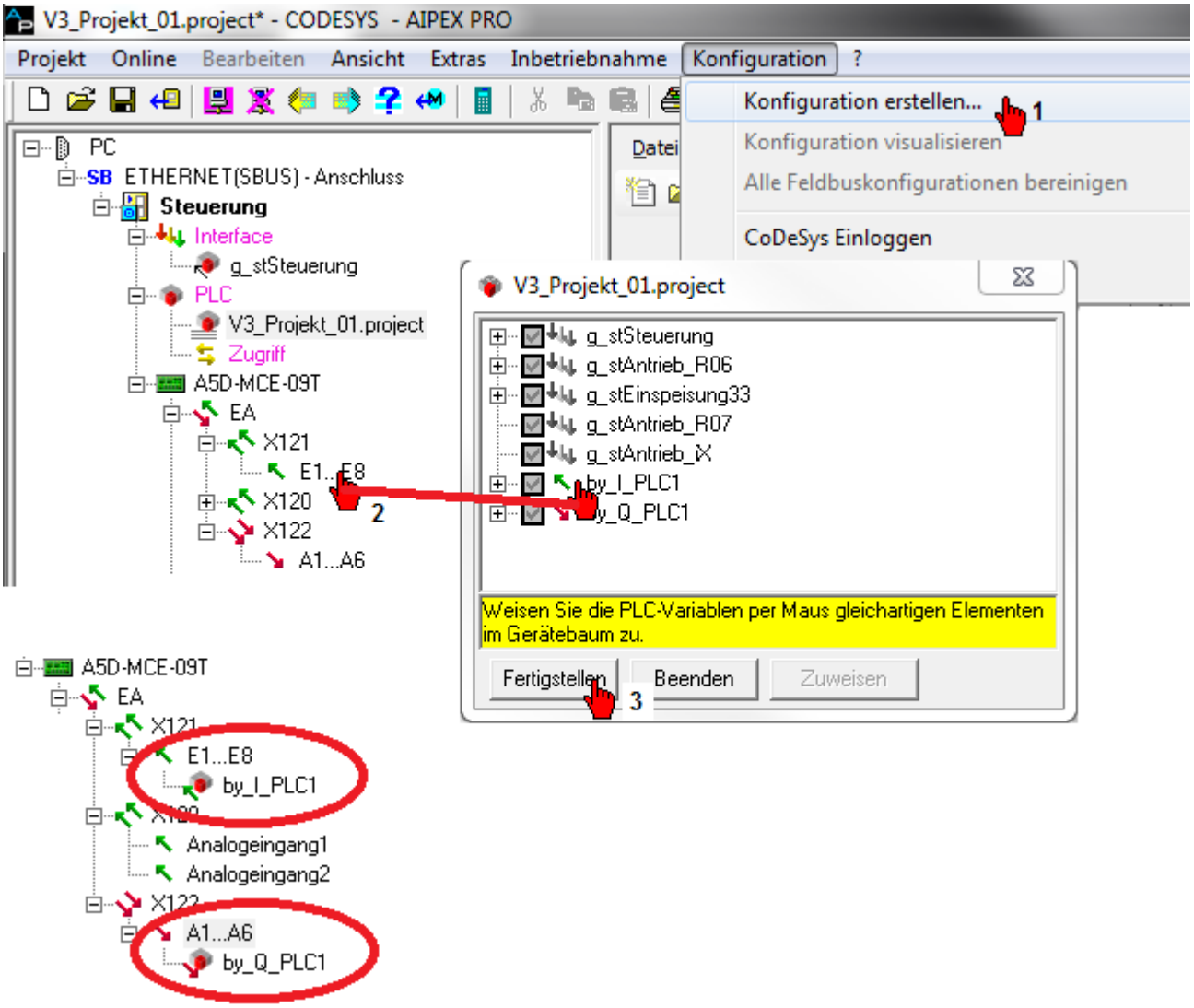
Die I/Os werden innerhalb eines Programmzyklus gelesen (am Programmanfang) und geschrieben (am Programmende).



Für jeden Ein- und Ausgangsblock muss in CODESYS eine globale I/O-Variable angelegt werden.
Im Beispiel wird ein Input Byte by_I_PLC1 und ein Output Byte by_Q_PLC1 angelegt.



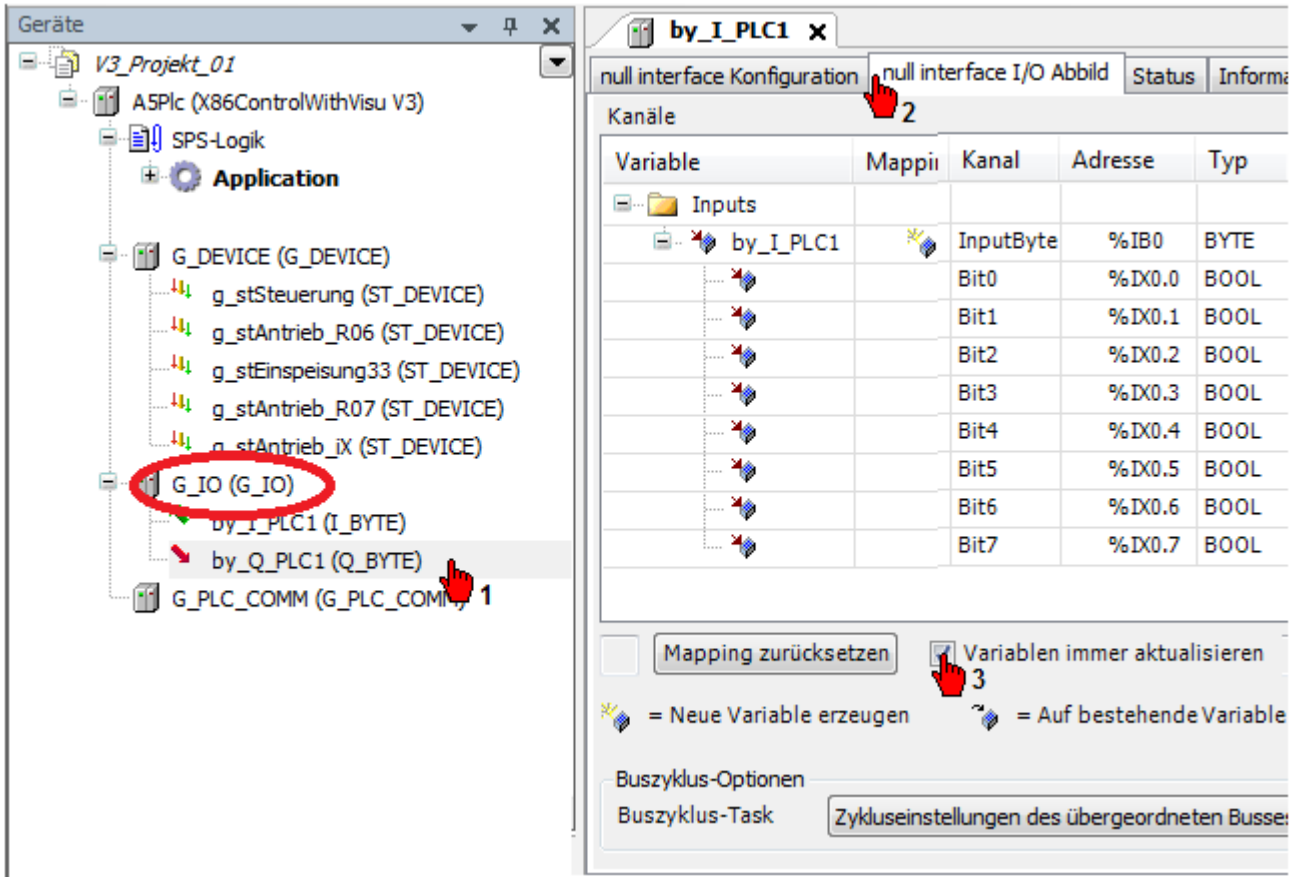
Rufen Sie **'Konfiguration erstellen'** auf. Weisen Sie die symbolischen Gerätenamen den I/O Schnittstellen im AIPEX PRO Gerätebaum zu.



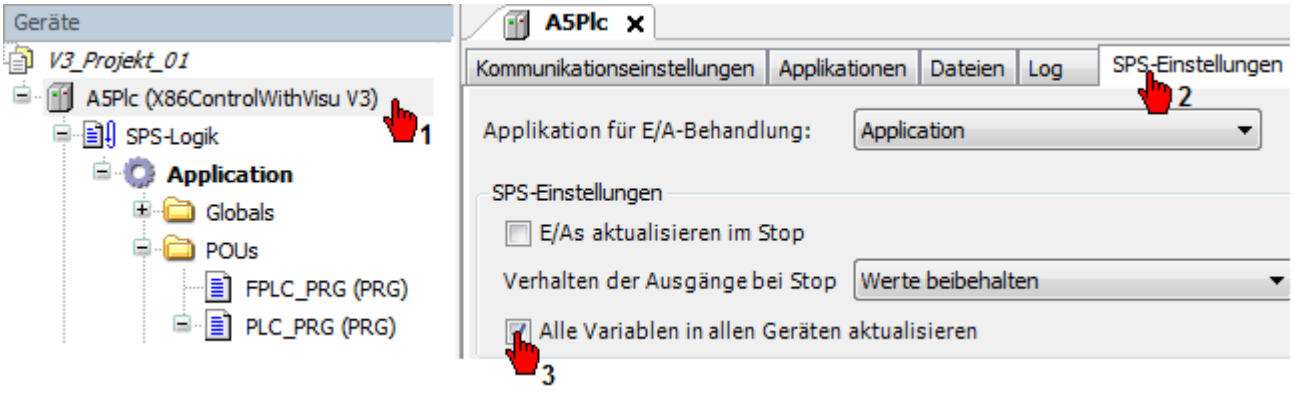
Testen

Damit die I/O Variablen im PLC Projekt aktualisiert werden, stehen folgende Möglichkeiten zu Verfügung:

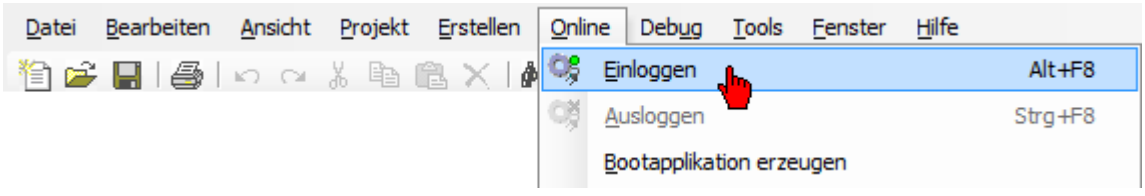
- Verknüpfen Sie die symbolische Gerätenamen mit PLC Variablen
- Setzen Sie die Option 'Variable immer aktualisieren' (siehe Bild)



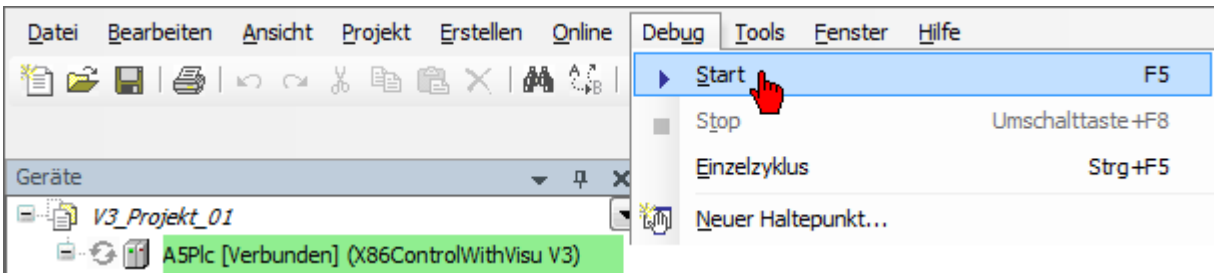
Alternativ können Sie die Option 'Alle Variablen in allen Geräten aktualisieren' aktivieren.



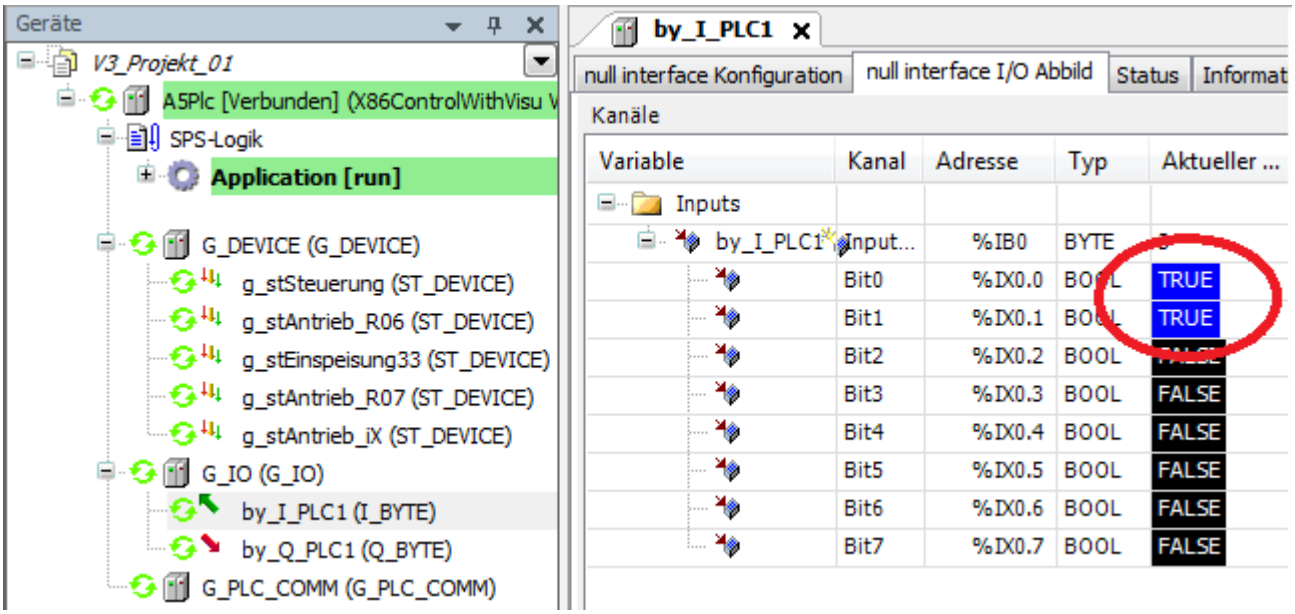
Rufen Sie die Funktion **'Einloggen'** auf (Code wird automatisch erzeugt).



Starten Sie die Applikation auf der Steuerung.



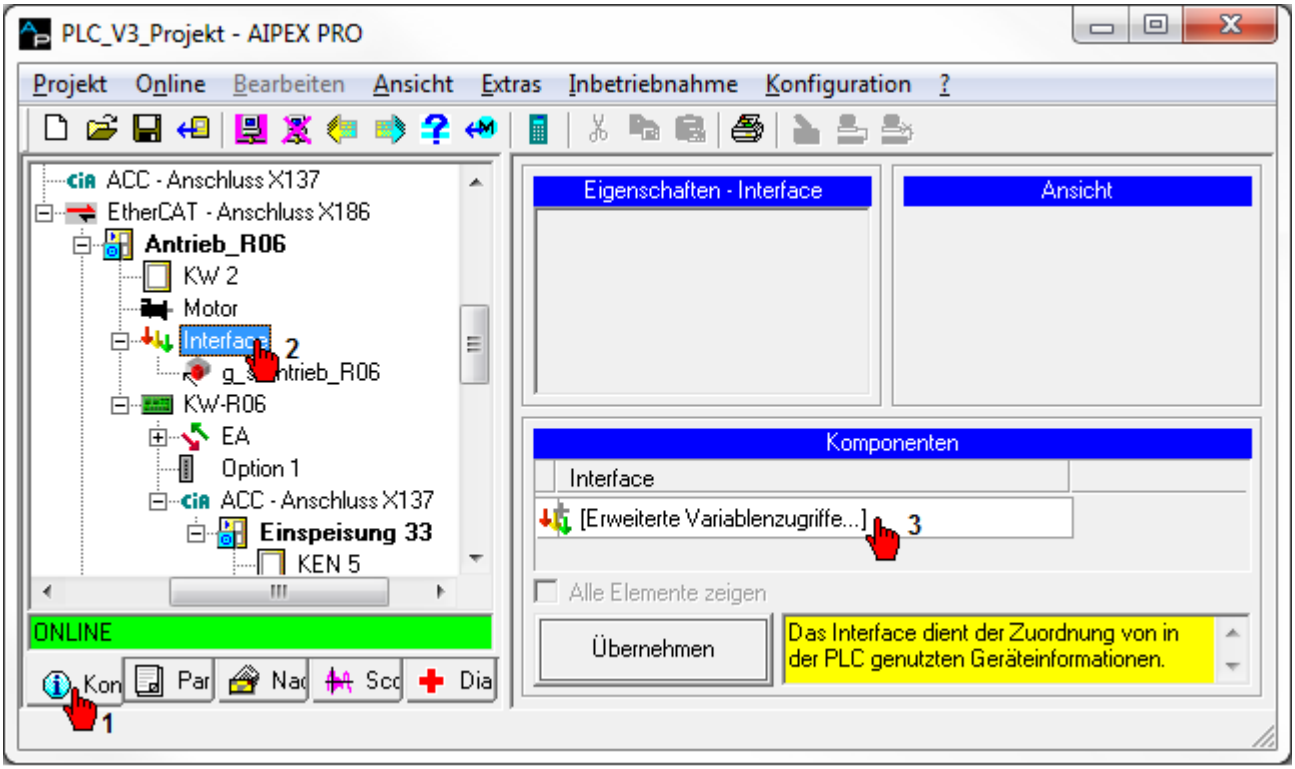
Setzen Sie zum Testen binäre Eingänge auf der Steuerung.



4.3.1.12.2 Erweiterte Variablenzugriffe

Die AMK Bibliotheken stellen eine Auswahl an Funktionsbausteinen zu Verfügung um Echtzeitbitmeldungen, Istwerte, Sollwerte usw. zwischen Antrieb und PLC auszutauschen. Nicht vorhandene (formale) Bausteine können über die Option 'Erweiterte Variablenzugriffe' erstellt werden.

Wählen Sie dazu den Antrieb (Interface) aus für den ein neuer Funktionsbaustein erstellt werden soll.

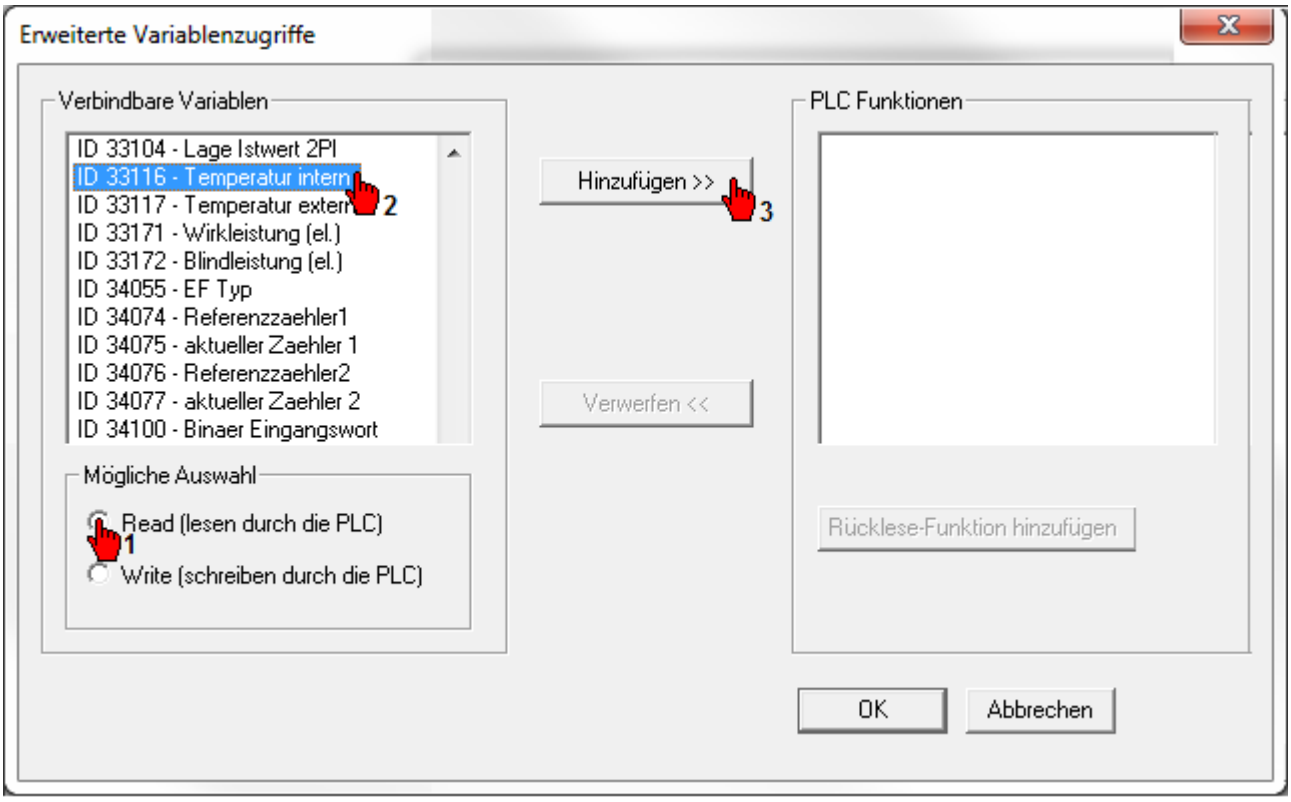


Wählen Sie im ersten Schritt aus ob von der PLC lesend oder schreibend auf den Antrieb zugegriffen werden soll.

In der Variablenliste finden Sie alle im Antrieb verfügbaren Werte.

Die Variablenauswahl wird mit der Taste **'Hinzufügen'** übernommen.

Mit der 'Rücklese Funktion' können 'GET_FDEV_...' Bausteine generiert werden, mit denen Sie die mit 'SET_FDEV_...' - Bausteinen geschriebenen Werten lesen können.

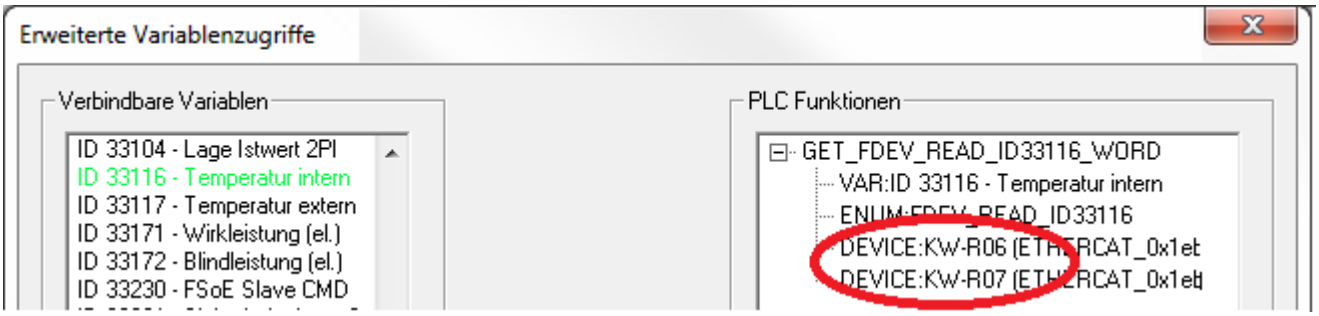


Zusätzliche Information:

Sie können den erstellten FDEV Baustein für alle Geräte im PLC Projekt instanzieren die den gleichen Gerätetyp besitzen. Z. B. DEVICE: KW-R06.

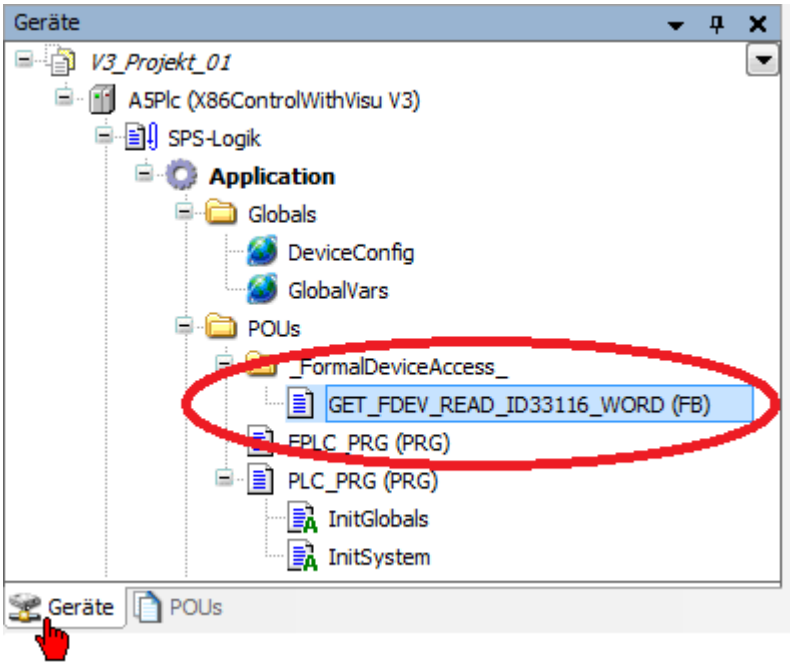
Wenn ihr PLC Projekt verschiedene Gerätetypen enthält (z. B. zusätzlich ein DEVICE: KW-R07), müssen Sie das 'Interface' dieses Gerätes anwählen und die Funktion 'Erweiterterte Variablenzugriffe' identisch ausführen.

FDEV Formaler Baustein, verknüpft mit den Geräten KW-R06 und KW-R07



Öffnen Sie den Programmiereditor.

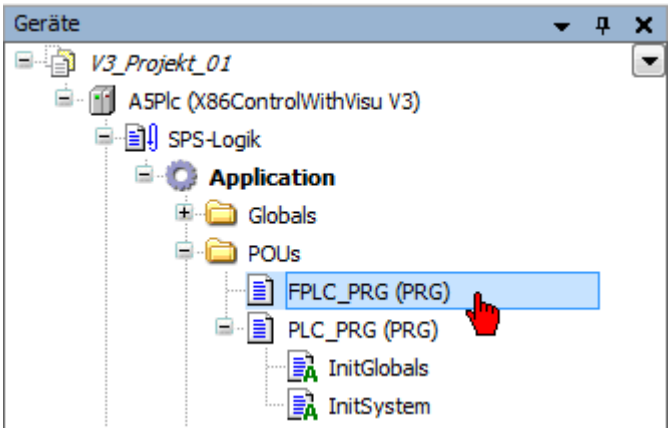
Sie finden den generierten Funktionsbaustein unter 'Geräte' → 'Application' → 'POUs' → _FormalDeviceAccesss_ wieder.



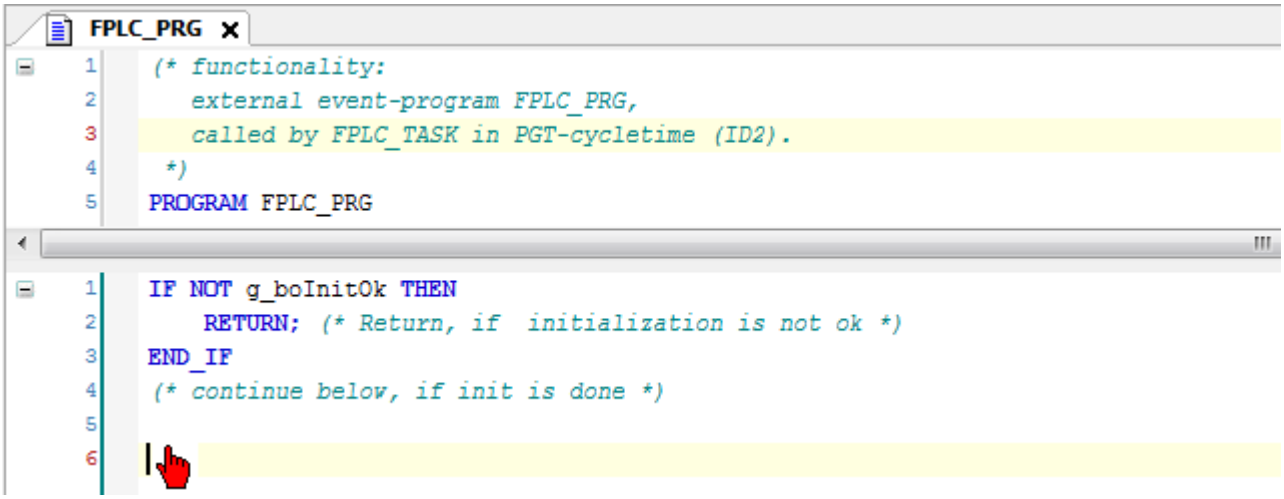
Eine konsistente Datenübertragung ist nur bei einem Instanzaufruf im synchronen Programmbaustein FPLC_PRG garantiert.

Wechseln Sie auf den Reiter 'Geräte'

Öffnen Sie den Programmbaustein 'FPLC_PRG (PRG)'



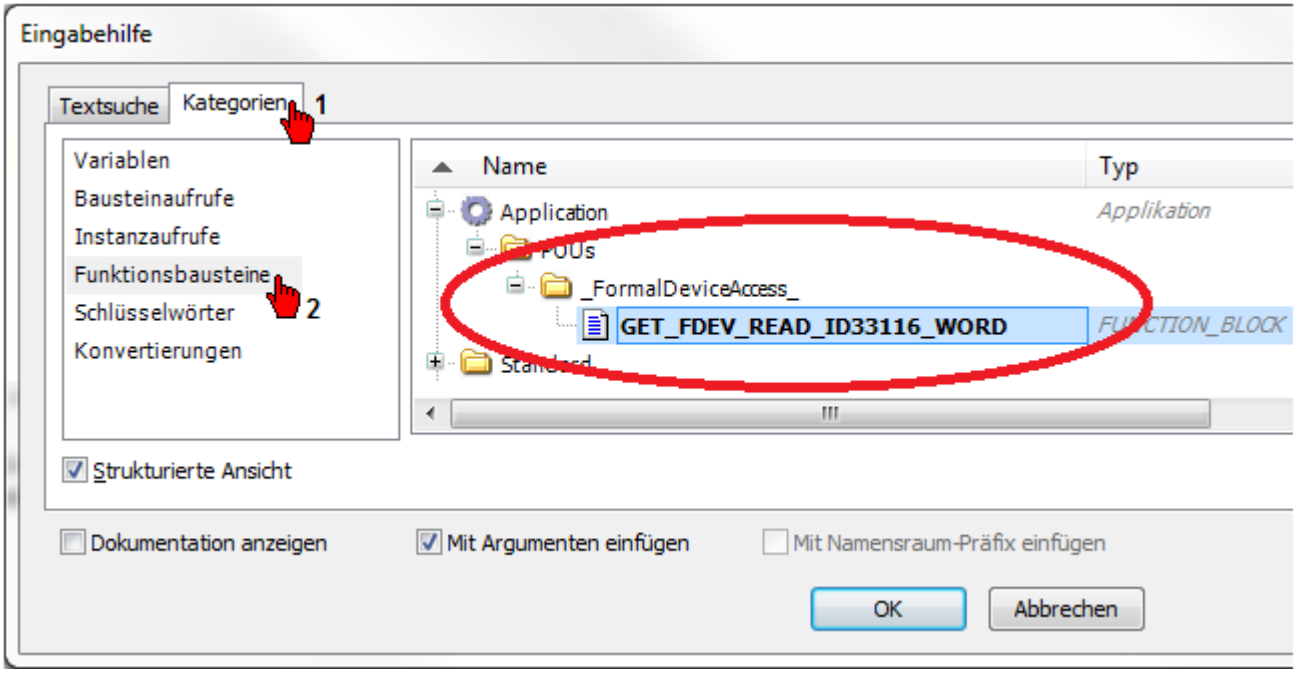
Klicken Sie im Programmeditor in eine leere Zeile.



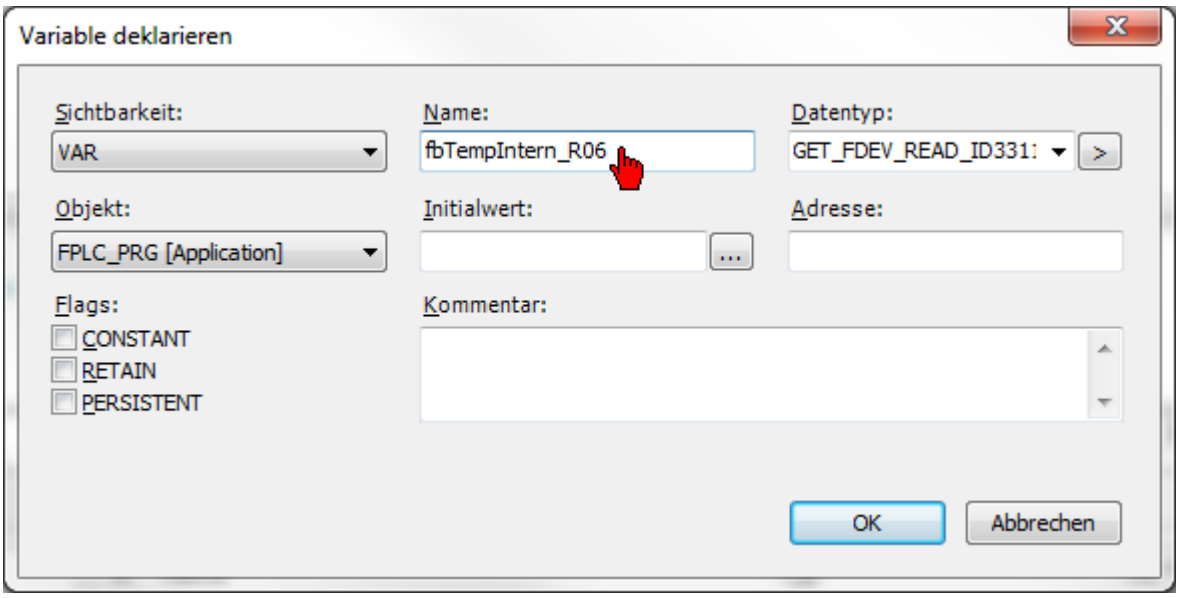
Öffnen Sie die Eingabehilfe durch drücken der **'F2'** Taste.

Sie finden den generierten Funktionsbaustein unter 'Kategorien' → 'Funktionsbausteine' → 'Application' -> 'POUs' -> _ FormalDeviceAccesss_.

Klicken Sie den Funktionsbaustein an, um ihn ins Projekt zu übernehmen.



Instanzieren Sie den Funktionsbaustein indem Sie ihm einen eigenen Namen zuweisen.



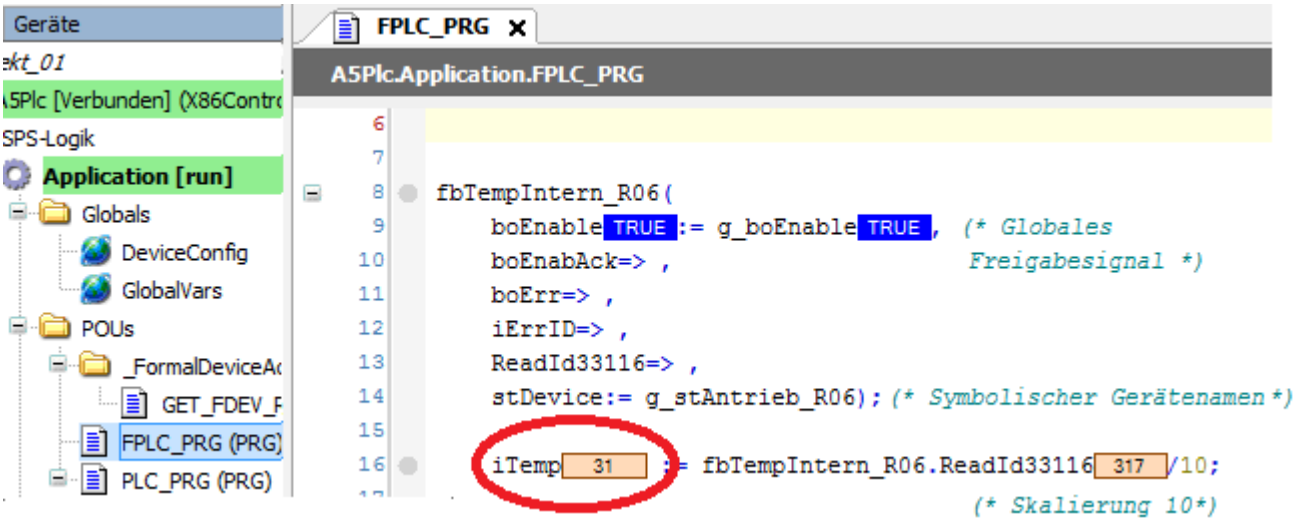
Beispiel:

```
fbTempIntern_R06(
    boEnable:= g_boEnable, (* Globales Freigabesignal *)
    boEnabAck=> ,
    boErr=> ,
    iErrID=> ,
    ReadId33116=> ,
    stDevice:= g_stAntrieb_R06); (* Symbolischer Gerätenamen *)

    iTemp := fbTempIntern_R06.ReadId33116/10; (* Skalierung 10*)
```

Testen

Erstellen Sie das Projekt: Siehe 'PLC Programm und Nachrichtenkonfiguration erstellen' auf Seite 695.

Starten Sie die PLC: Siehe 'PLC Projekt testen' auf Seite 698.



4.3.1.12.3 EtherCAT Querkommunikation zwischen zwei AMK Steuerungen

Das Beispiel zeigt, wie Sie zwischen AMK Steuerungen ein Datenaustausch (schreibend und lesend) realisieren.

Das PLC Programm der Steuerung PLC_Master sendet eine Integer Variable die von dem PLC Programm der Steuerung PLC_Slave_02 empfangen wird.

4.3.1.12.3.1 Vorbereitung

Vorausgesetzte Verdrahtung und Parametrierung der Steuerungen:
Siehe Produktbeschreibung Steuerungen A4 / A5 / A6 Kapitel: 'Querkommunikation'

Erzeugen Sie ein Online Projekt
Siehe Produktbeschreibung Steuerungen A4 / A5 / A6 Kapitel: 'Querkommunikation - Online Projekt mit AIPEX PRO einlesen und konfigurieren'.

Parametrierung der Steuerungen

Bei der Querkommunikation zwischen AMK Steuerungen müssen folgende Parameter konfiguriert werden:
Verwenden Sie dazu den AIPEX PRO 'Direktmode'

EtherCAT Master

Instanz	Parameter	Wert	Bedeutung
5	ID1204 'XML File'		Bus Konfigurationsfile
	ID1205 'XML File'		Bus Konfigurationsfile
	ID1206 'XML File'		Bus Konfigurationsfile
	ID1207 'XML File'		Bus Konfigurationsfile
	ID34023 'BUS Teilnehmer Adresse'	0xFF	Standardeinstellung, Adresse darf im Bussystem nur einmal vergeben werden
	ID34024 'BUS Übertragungsrate'	100.000	Standardeinstellung, entspricht 100 Mbaud
	ID34025 'BUS Modus'	0x2	Master
	ID34026 'BUS Modusattribut'	0x0	-
	ID34140 'AS Bus Protokoll'	0x41	EtherCAT Master / Slave
	ID34143 'Verwendung Port'	0x2	CC Bus (Querkommunikation zwischen Steuerungen)

EtherCAT Slave

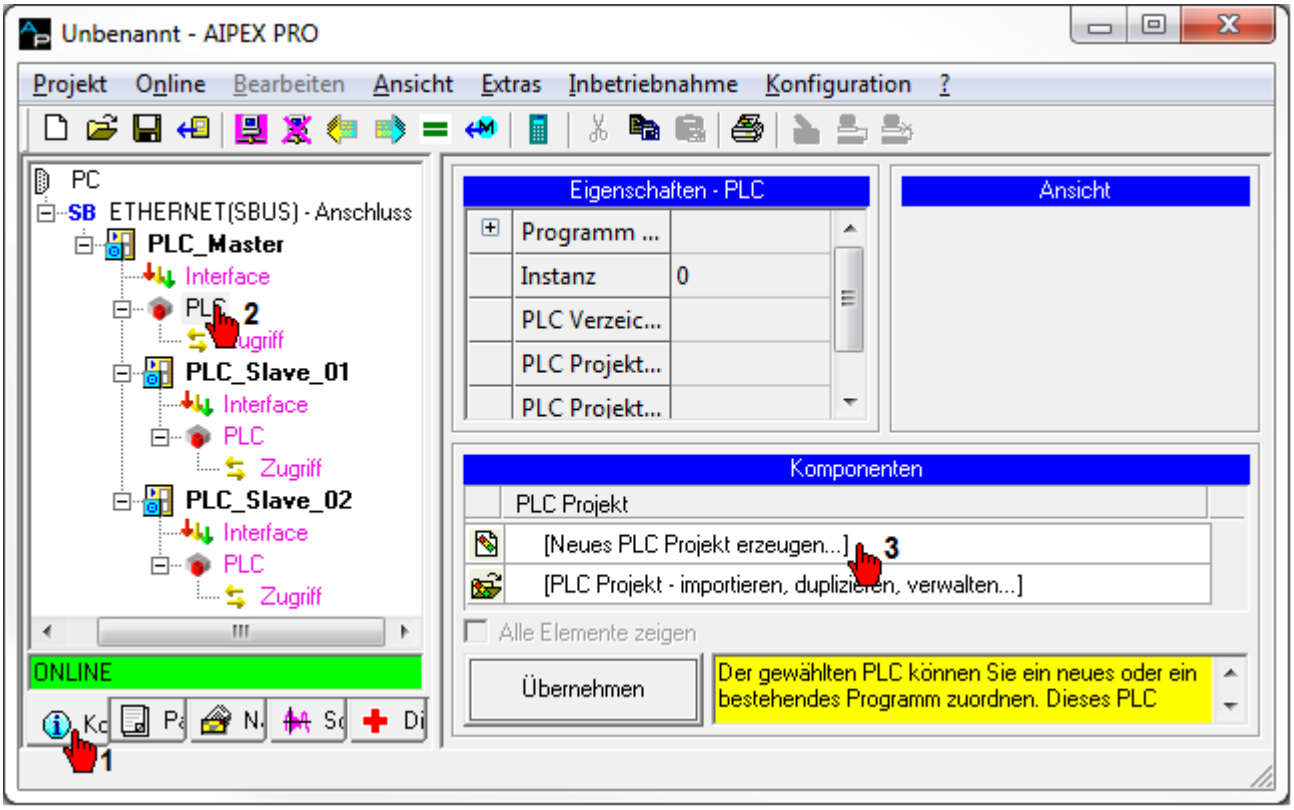
Instanz	Parameter	Wert	Bedeutung
2	ID34023 'BUS Teilnehmer Adresse'	0	Standardeinstellung, automatische Adressierung
	ID34024 'BUS Übertragungsrate'	100.000	Standardeinstellung, entspricht 100 Mbaud
	ID34025 'BUS Modus'	0	Slave
	ID34026 'BUS Modusattribut'	0	-
	ID34140 'AS Bus Protokoll'	0x41	EtherCAT Slave Option A-SEC
	ID34143 'Verwendung Port'	2	CC Bus (Querkommunikation zwischen Steuerungen)

Die Querkommunikation muss vor dem Antriebsbus initialisiert sein. Bei Bedarf kann beim Antriebsbus Master über die ID34026 'BUS Modusattribut' eine Hochlaufverzögerung eingestellt werden.

Nach dem Parametrieren müssen beide (alle) Steuerungen neu gestartet werden (24 VDC AUS/EIN).
Nach dem Neustart ist die Querkommunikation aktiv. Der EtherCAT Master ist über den EtherCAT (CC) Anschluss mit dem EtherCAT Slave verbunden.
Lesen Sie die Konfiguration ein und speichern Sie das Projekt: AIPEX PRO Menü 'Online' → 'Einloggen'
Projekt speichern: AIPEX PRO Menü 'Projekt' → 'Speichern unter ...'

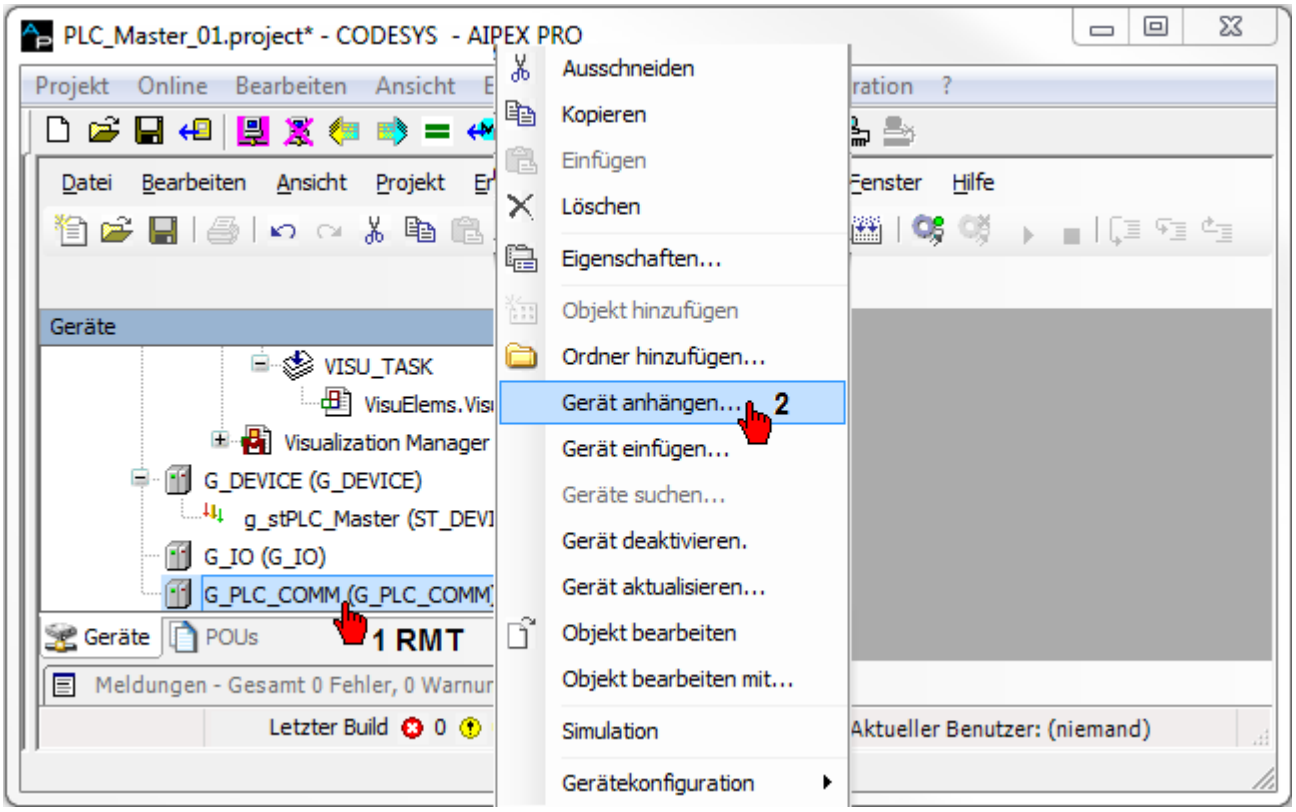
4.3.1.12.3.2 PLC Programm Master Steuerung

Erzeugen und speichern Sie das PLC Programm für die Steuerung PLC_Master.

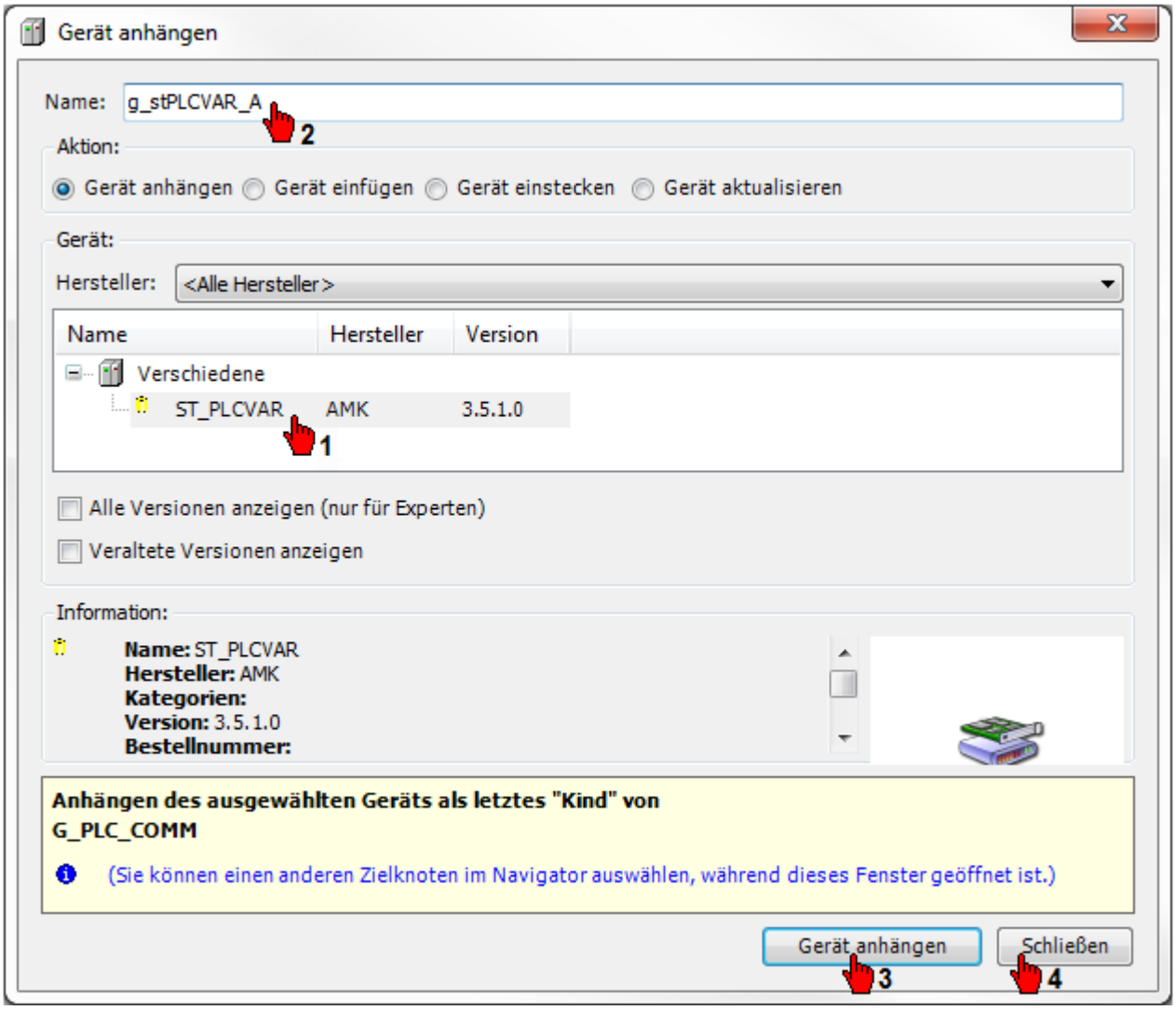


Übernehmen Sie den 'PLC-Handle Gerätenamen' (symbolischer Gerätenamen) der PLC_Master Steuerung in das PLC Programm.

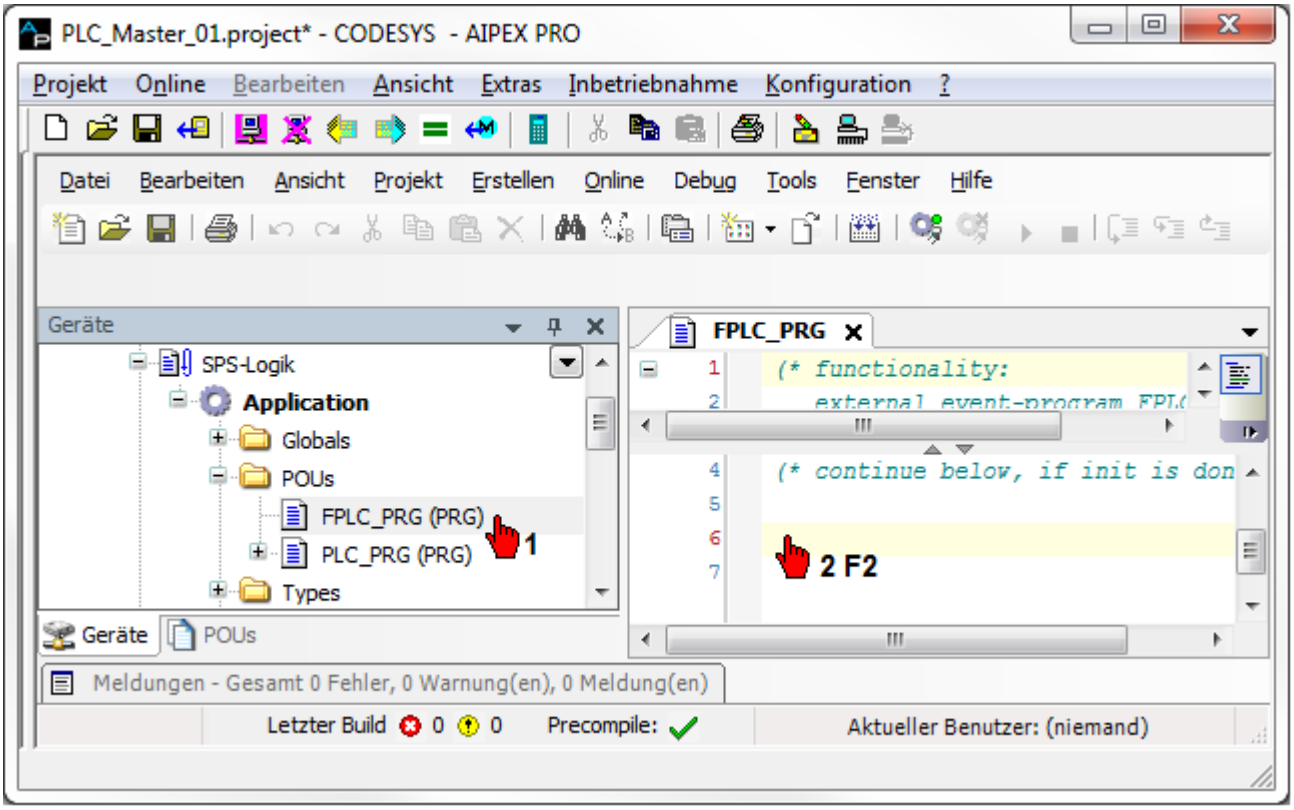
Wechseln Sie auf die 'Steuerungskonfiguration'.
Hängen Sie in der Steuerungskonfiguration G_PLC_COMM eine Kommunikationsvariable vom Typ ST_PLCVAR an.



Erstellen Sie eine Kommunikationsvariable (im Beispiel g_stPLCVAR_A).



Im Beispiel wird eine Integer Variable in der synchronen Ebene übertragen.
Instanzieren Sie in der Echtzeitebene FPLC_PRG einen Funktionsbaustein des Typs SET_PLCVAR_SYNC_INT .



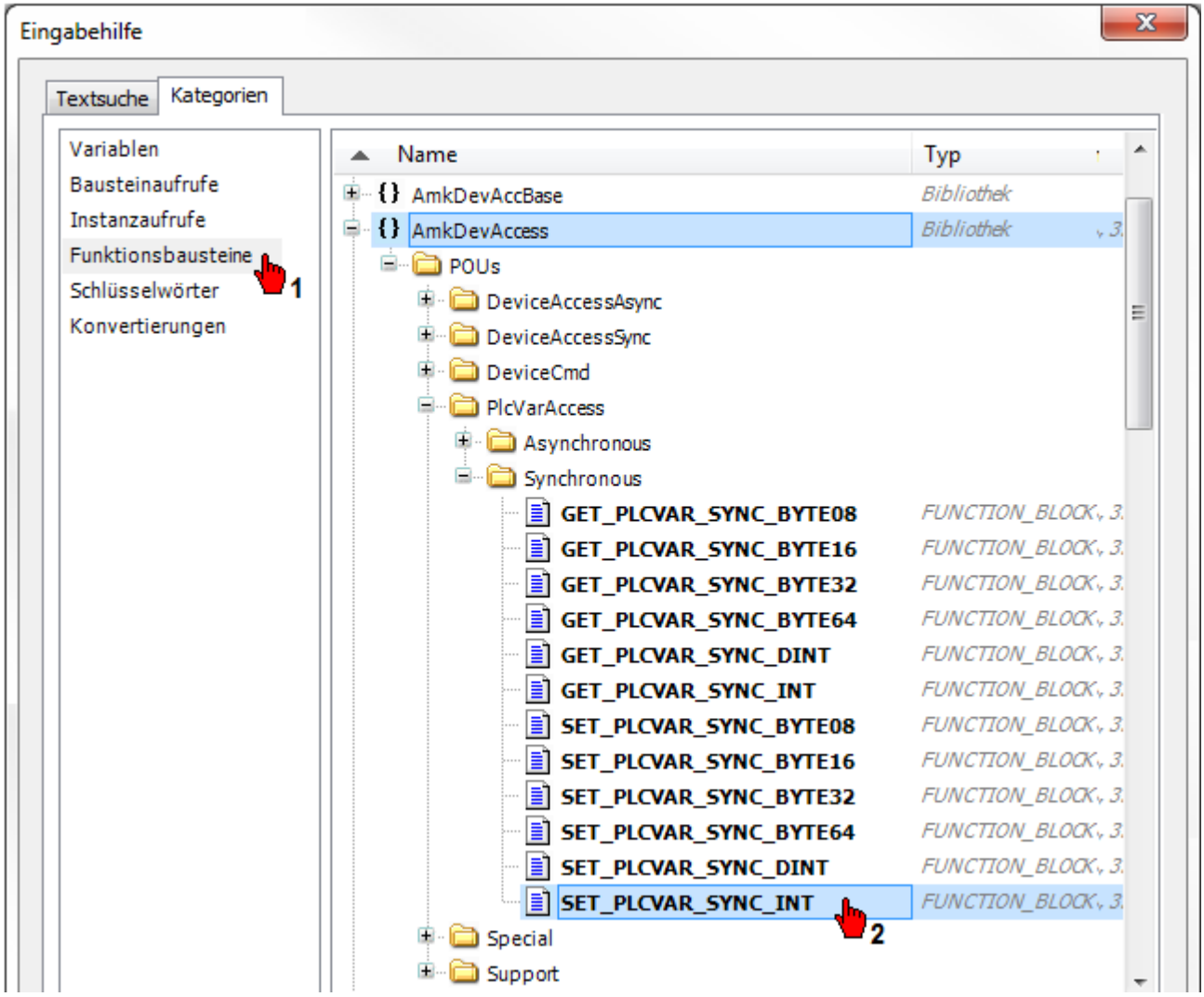
PLC Funktionsbausteine für den Kommunikationsaustausch zwischen den Steuerungen werden mit der AMK Bibliothek 'AmkDevAccess' zu Verfügung gestellt.

Asynchroner Datenaustausch siehe: Ordner: 'POUs' → 'PlcVarAccess' → 'Asynchronous'

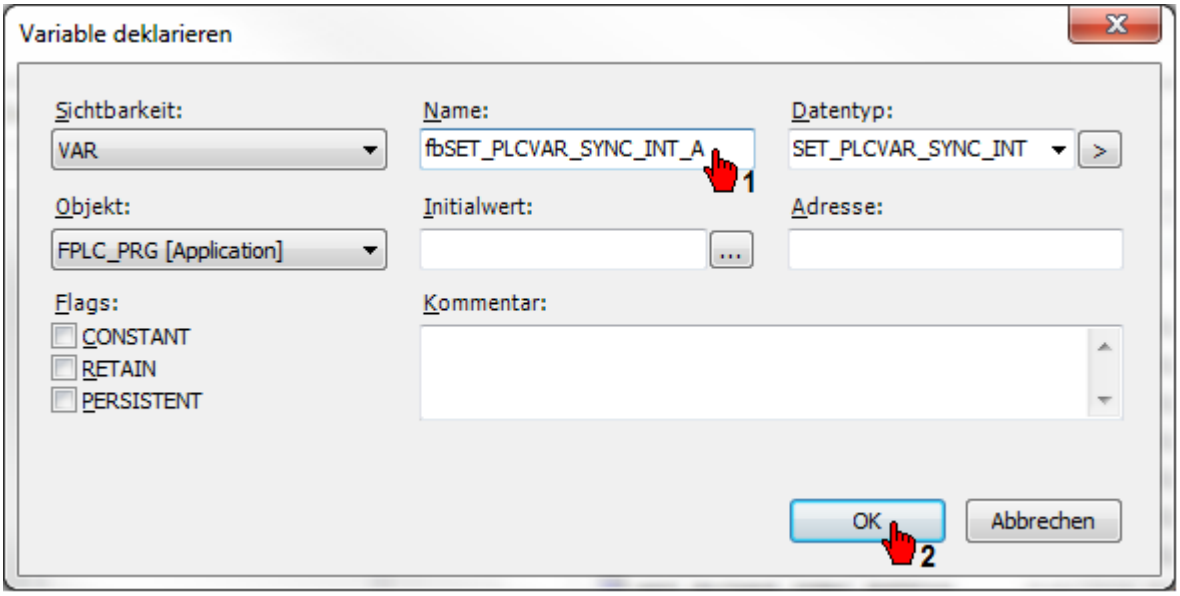
Synchroner Datenaustausch siehe: Ordner 'POUs' → 'PlcVarAccess' → 'Synchronous'

Funktionsbaustein SET_: schreiben

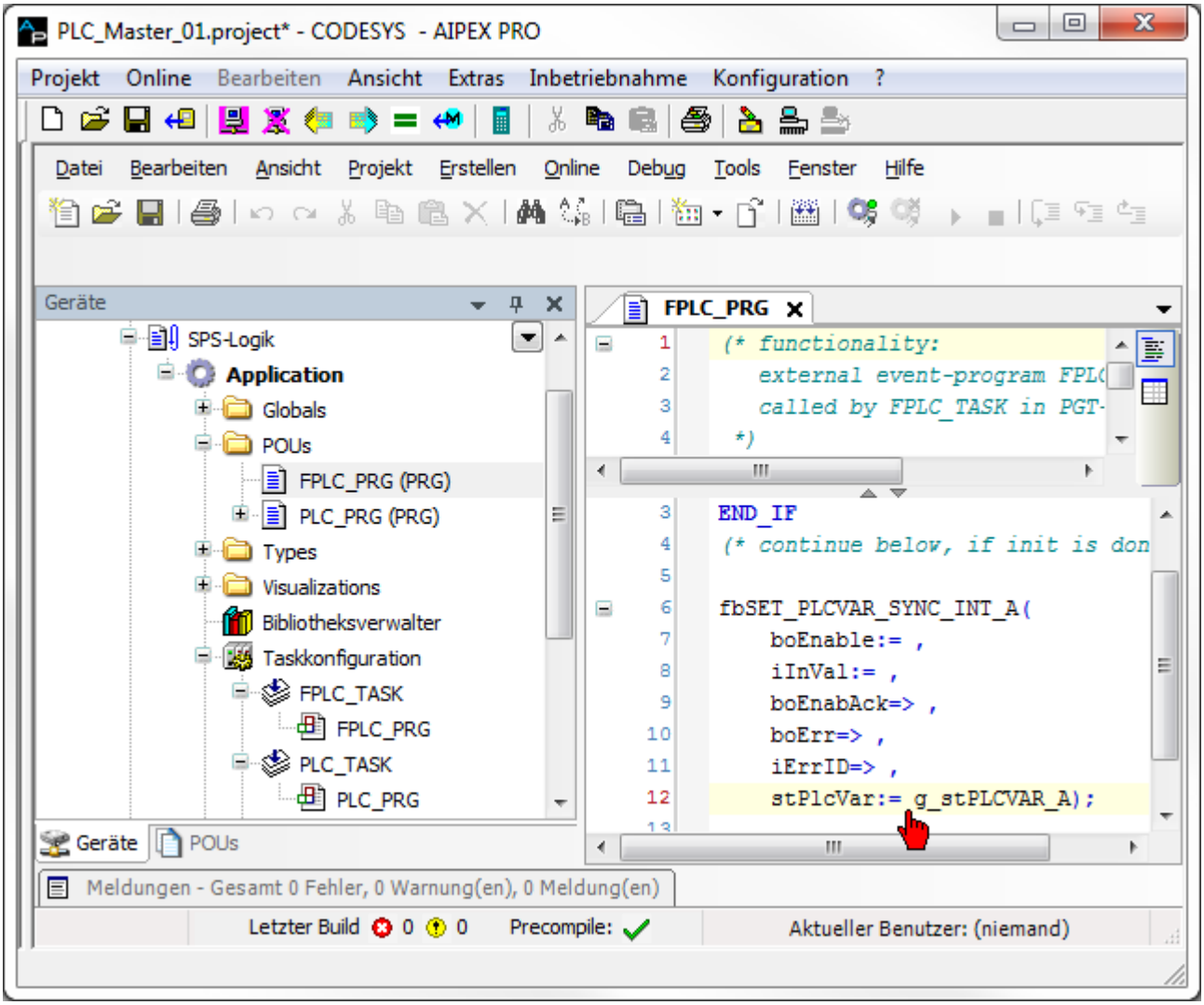
Funktionsbaustein GET_: lesen



Instanzieren Sie den Funktionsbaustein indem Sie ihm einen eigenen Namen zuweisen.



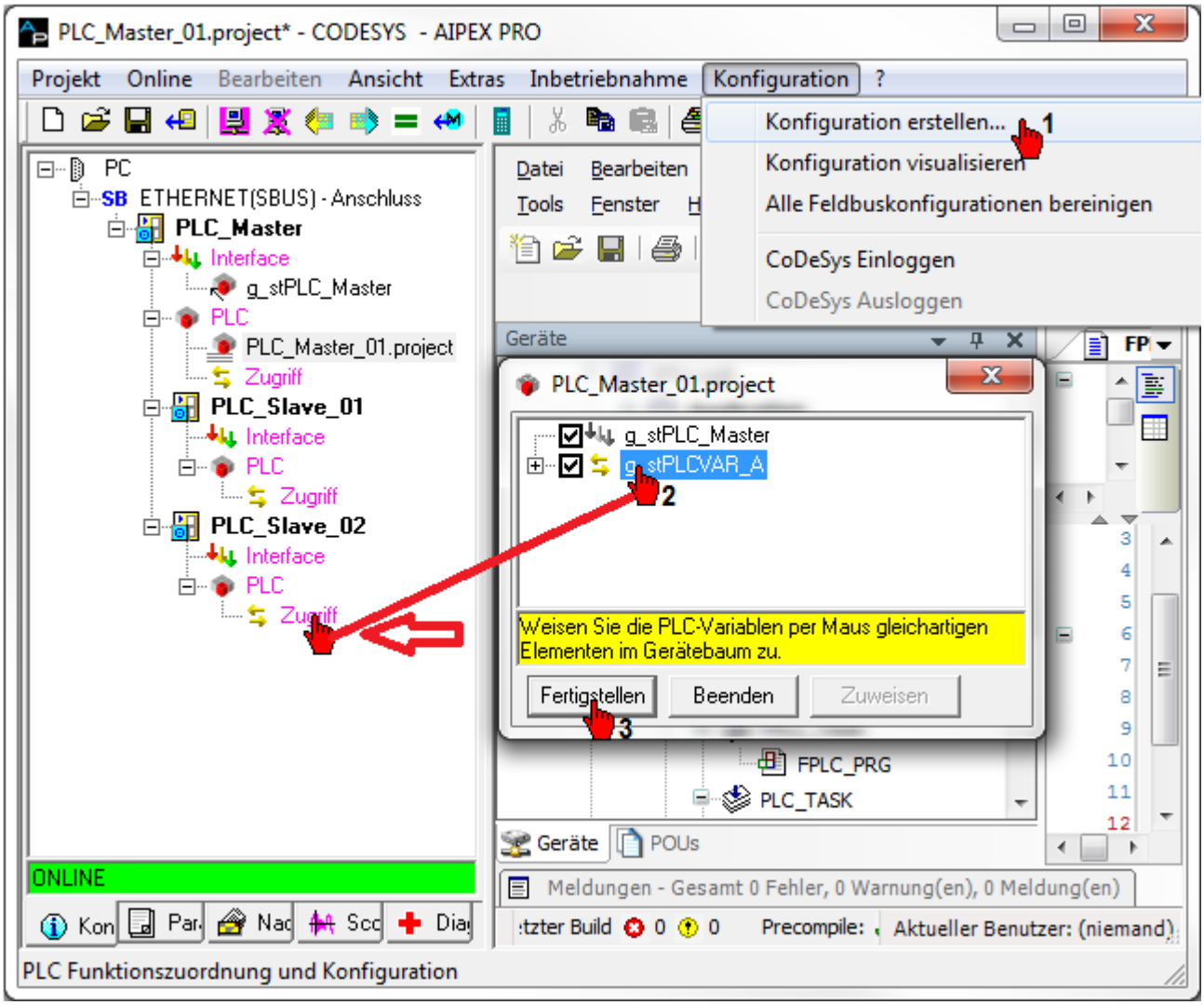
Verknüpfen Sie die Funktionsblockvariable 'stPlcVar' mit der Kommunikationsvariablen (g_stPLCVAR_A) .



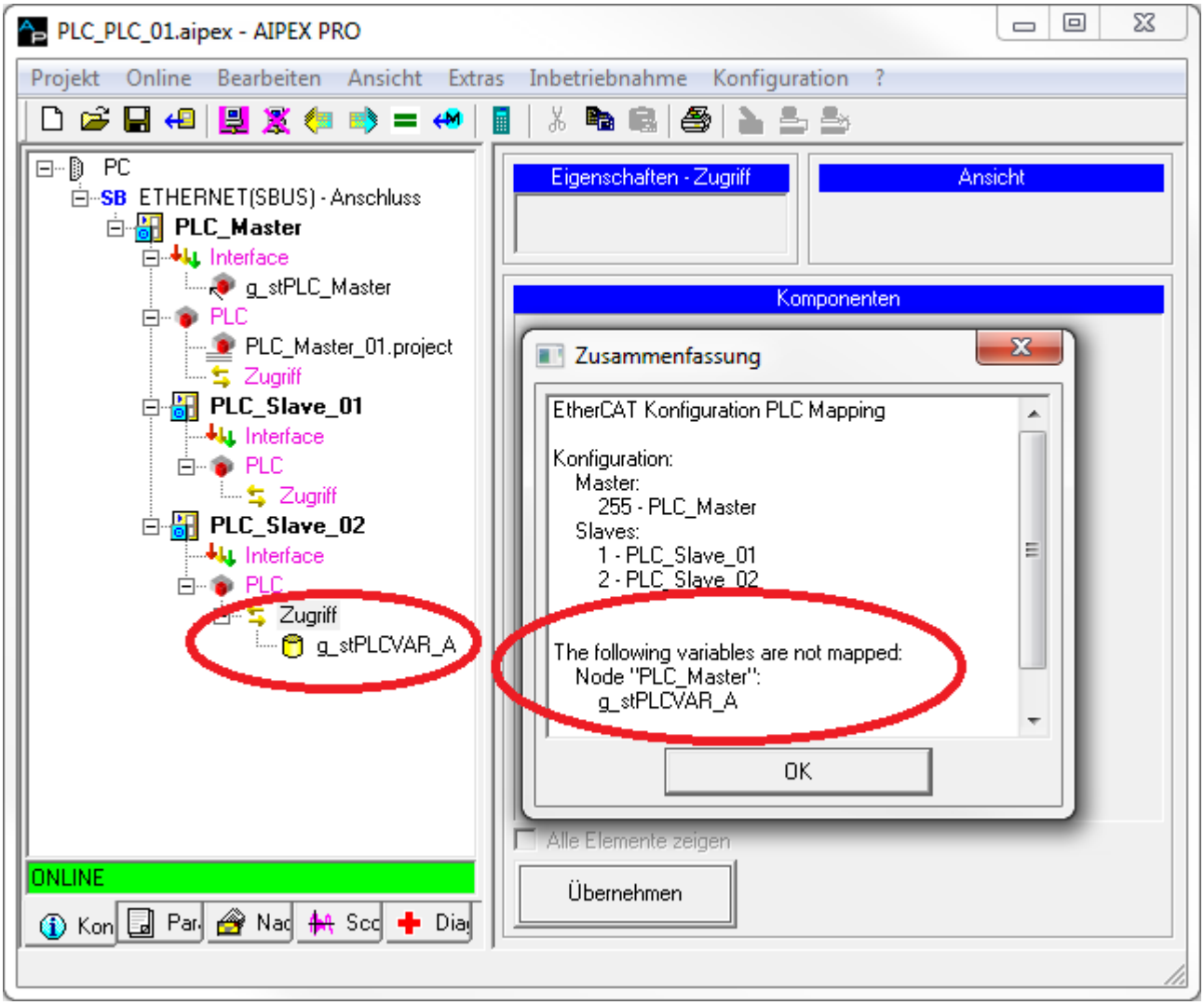
Nachrichtenkonfiguration für den EtherCAT Master erstellen

Klicken Sie auf das AIPEX PRO Menü 'Konfiguration' → 'Konfiguration erstellen ...'

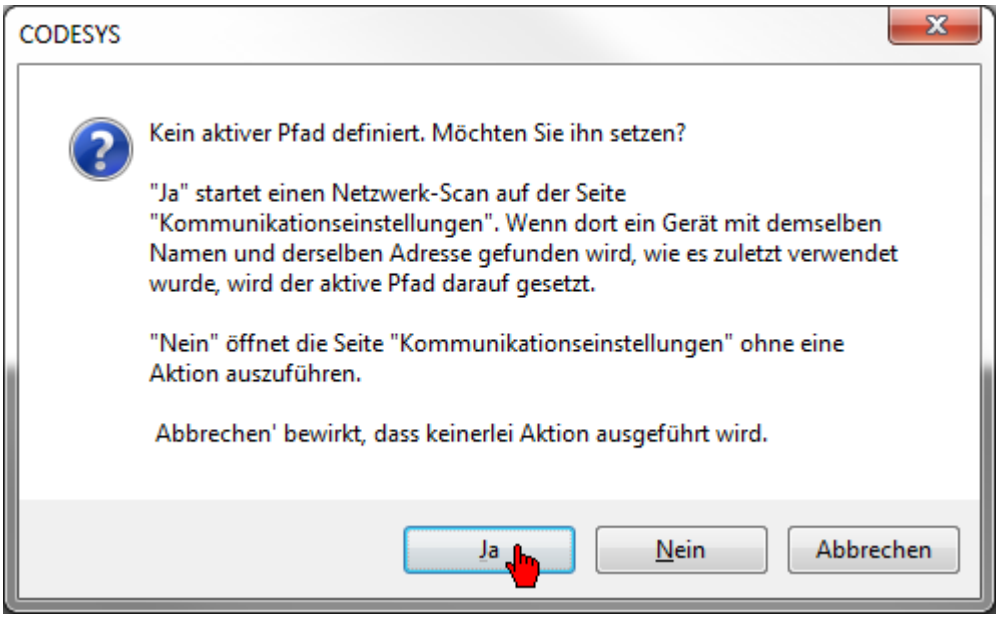
Weisen Sie die Kommunikationsvariable per 'Drag and Drop' dem PLC_Master zu (Schnittstelle 'Zugriff')



Das abschließende Variablen Mapping findet statt, nachdem die Slave PLC programmiert ist und die Funktion 'Konfiguration erstellen' aufgerufen wird.



Kommunikationseinstellungen

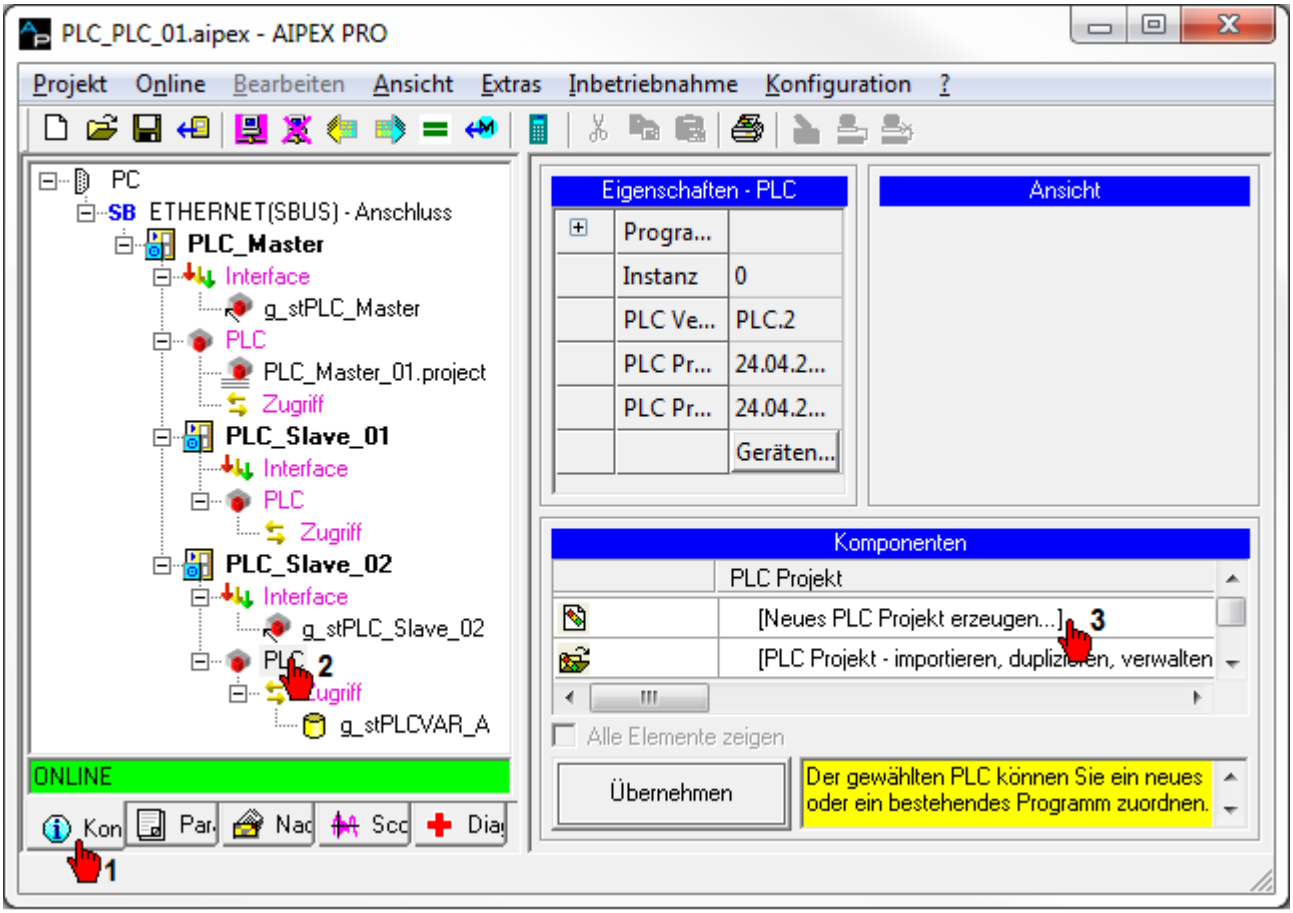


Siehe 'PLC Programm und Nachrichtenkonfiguration erstellen' auf Seite 695.

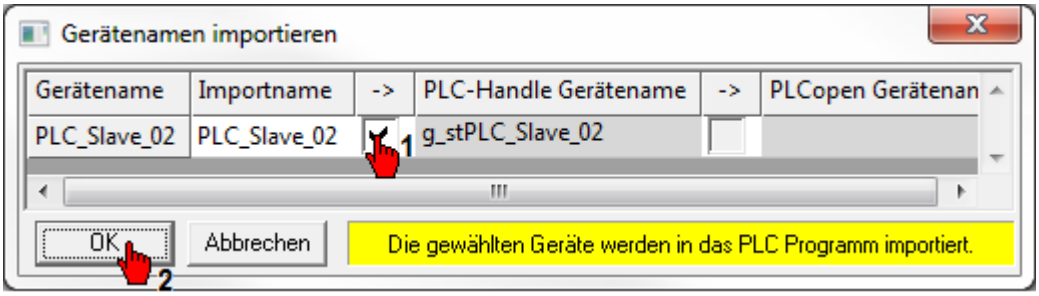
Abschnitt Kommunikationseinstellungen

4.3.1.12.3.3 PLC Programm Slave Steuerung

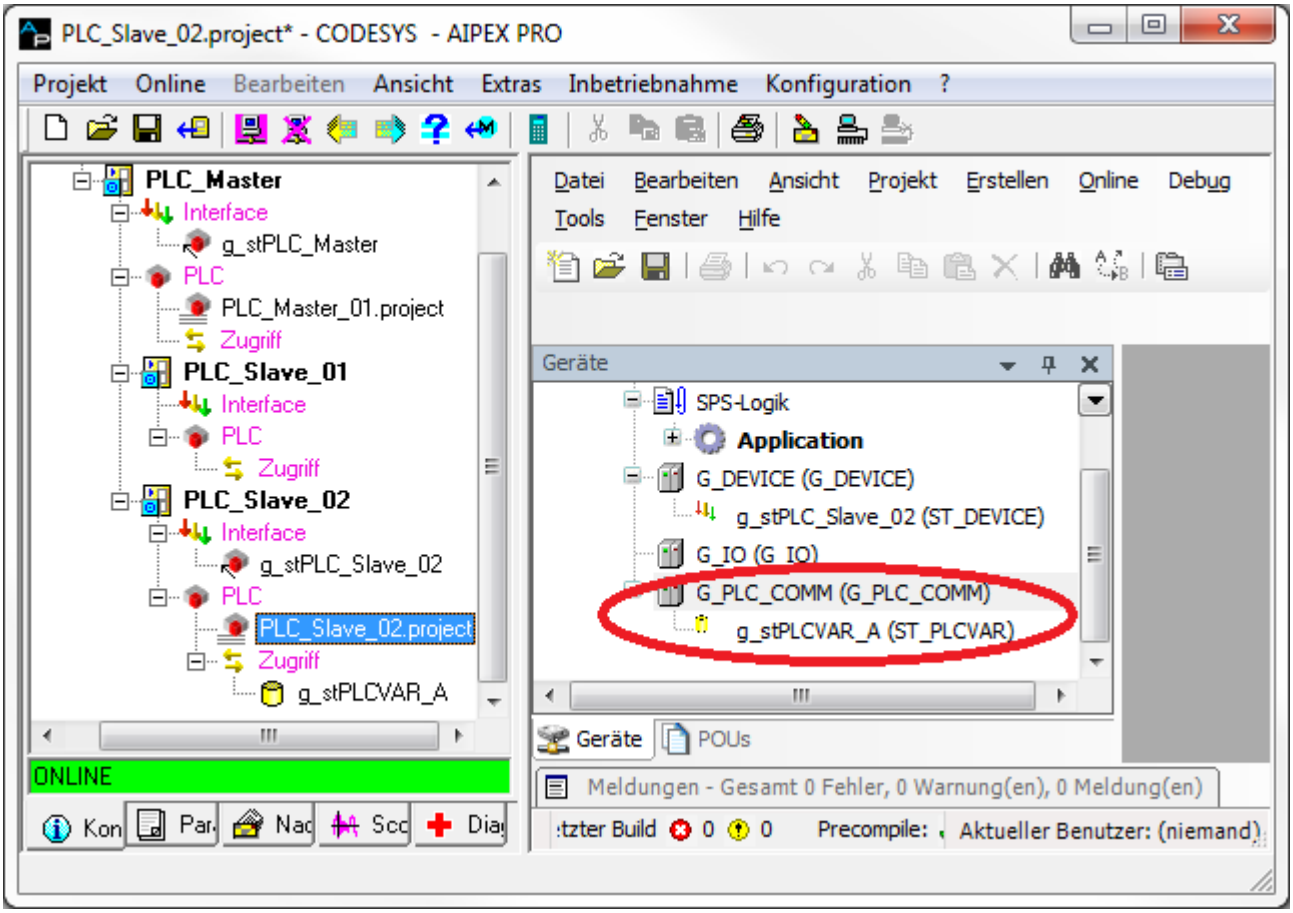
Erzeugen und speichern Sie das PLC Programm für die Steuerung PLC_Slave_02.



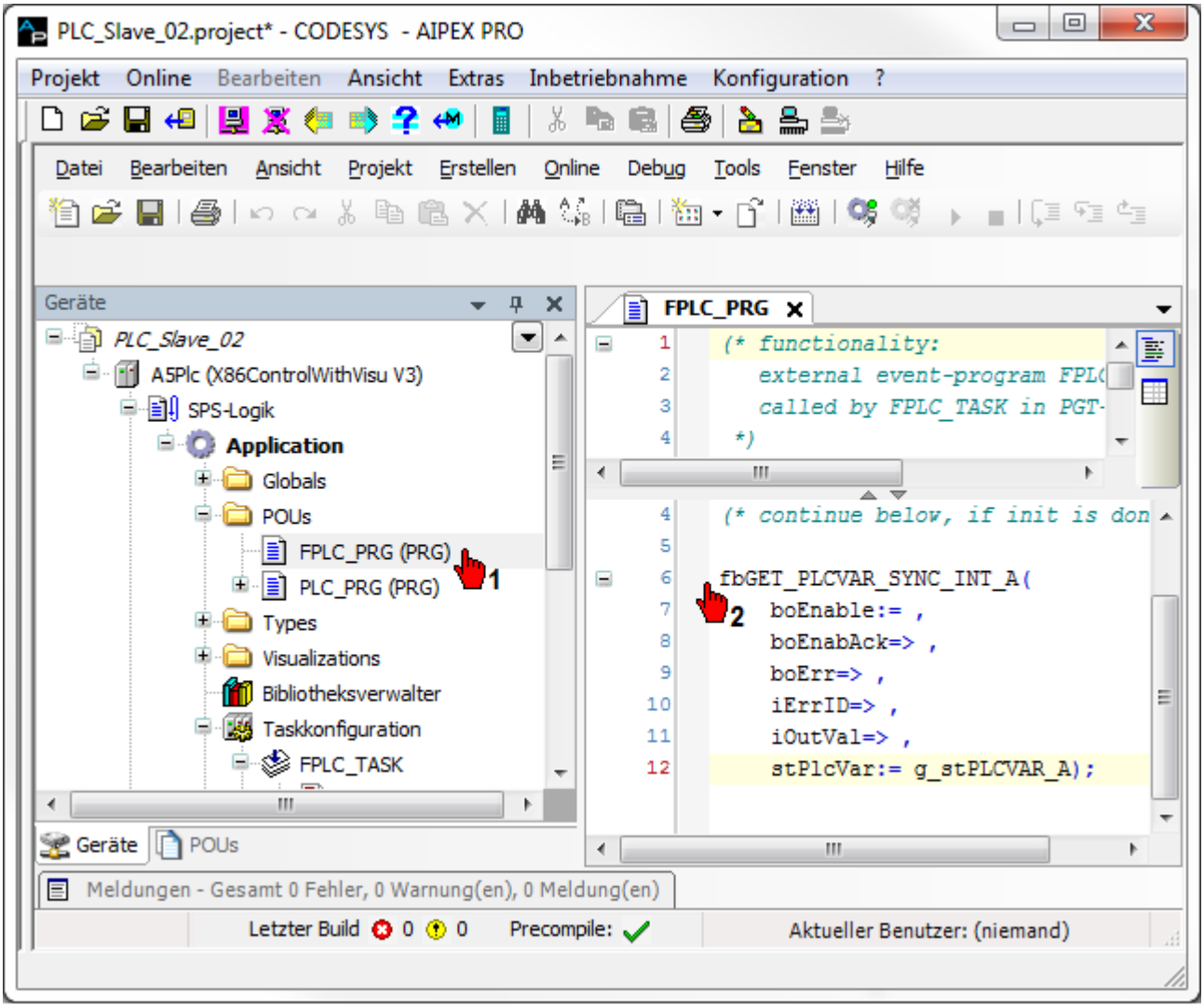
Übernehmen Sie den 'PLC-Handle Gerätenamen' (symbolischer Gerätenamen) der PLC_Slave_02 Steuerung in das PLC Programm.



Die zugewiesene PLC Kommunikationsvariable aus der Steuerung PLC_Master (im Beispiel g_stPLCVAR_A) wird automatisch in die Steuerungskonfiguration 'G_PLC_COMM' importiert.

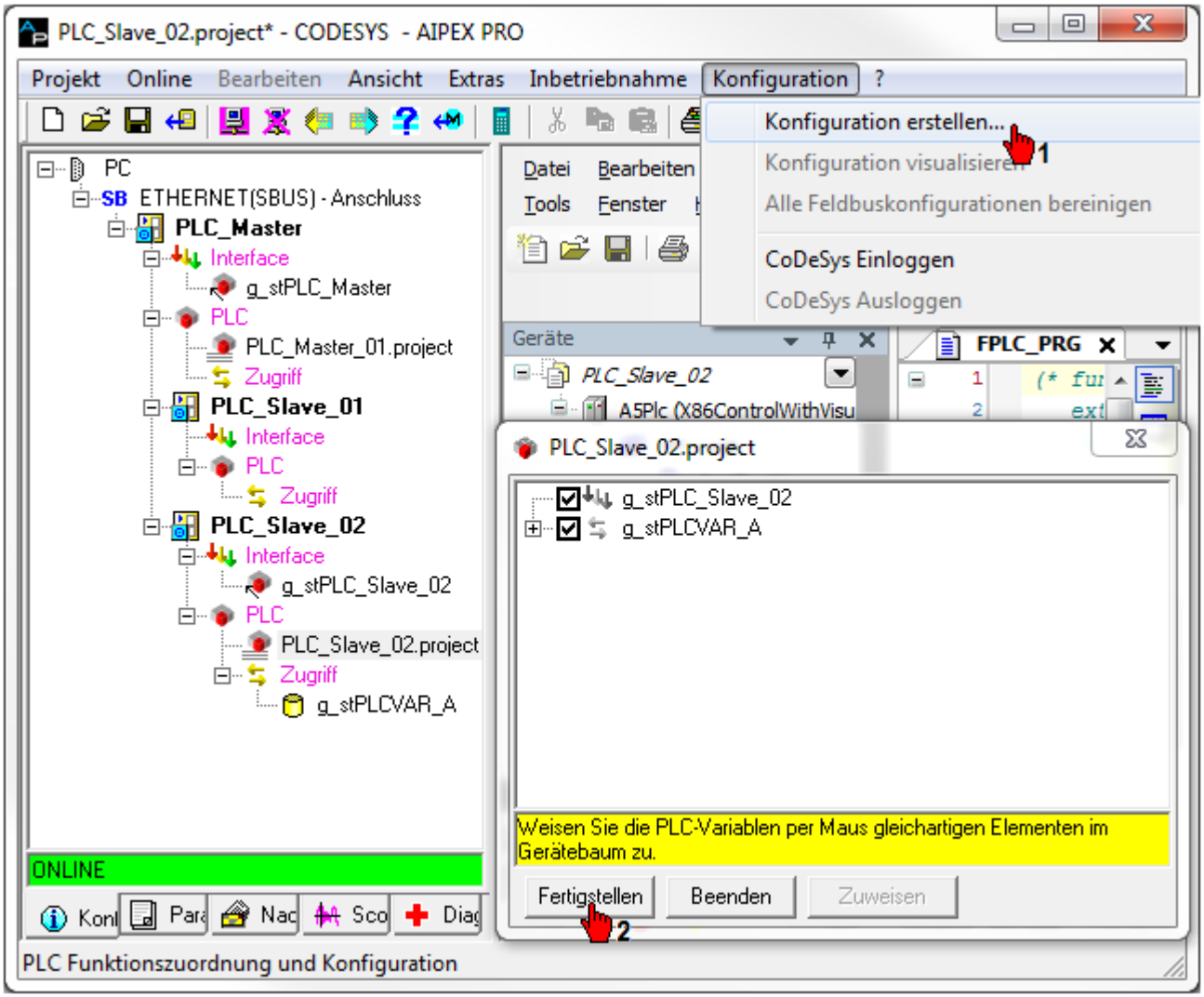


Instanzieren Sie in der Echtzeitebene FPLC_PRG einen Funktionsbaustein des Typs GET_PLCVAR_SYNC_INT .
Verknüpfen Sie die Funktionsblockvariable 'stPlcVar' mit der Kommunikationsvariablen (g_stPLCVAR).

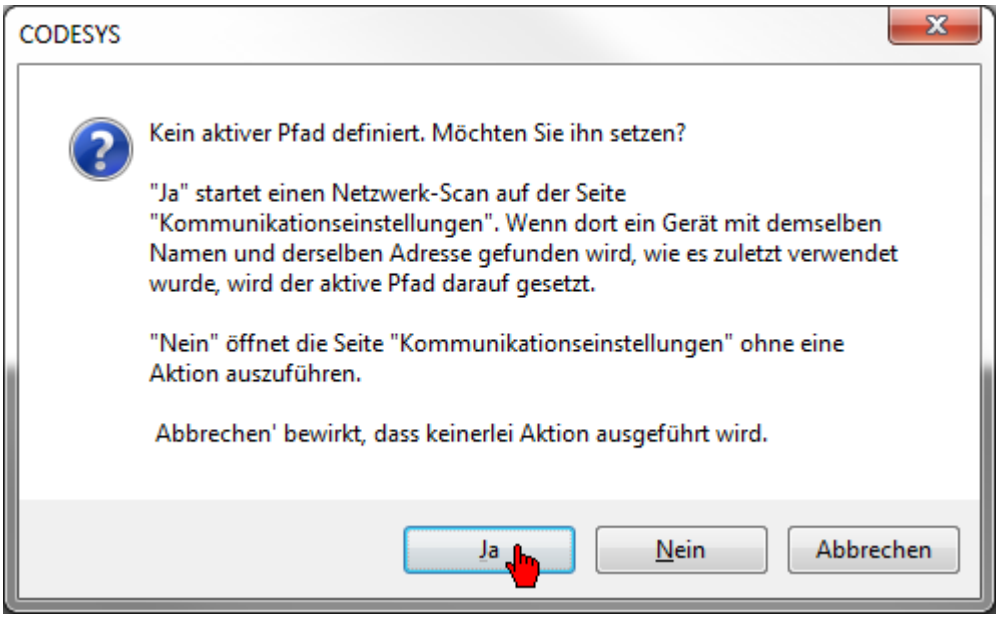


Nachrichtenkonfiguration für den EtherCAT Slave erstellen

AIPEX PRO: Menü Konfiguration 'Konfiguration erstellen ...'



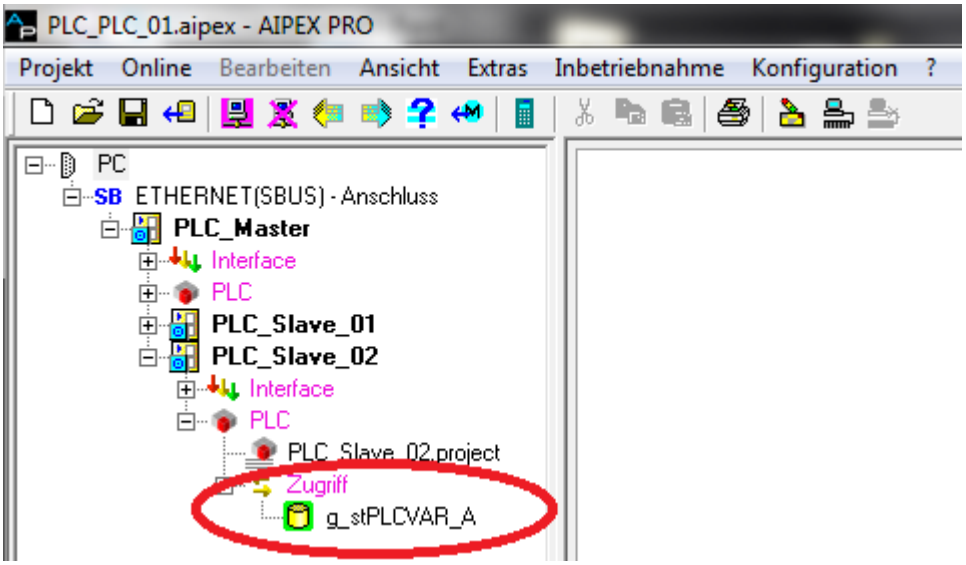
Kommunikationseinstellungen



Siehe 'PLC Programm und Nachrichtenkonfiguration erstellen' auf Seite 695.

Abschnitt Kommunikationseinstellungen

Das Icon einer gemappten PLC Kommunikationsvariable ist 'grün' hinterlegt.



Konfiguration abgeschlossen! Das Programm kann getestet werden.

- Für jeden Sendebaustein SET_ muss eine neue PLC Kommunikationsvariable angelegt werden.
- Eine PLC Kommunikationsvariable kann mit beliebig vielen PLC Slave Steuerungen verknüpft werden.
- Gehen Sie gleich vor, wenn Sie Daten zwischen zwei Slave PLCs austauschen wollen.

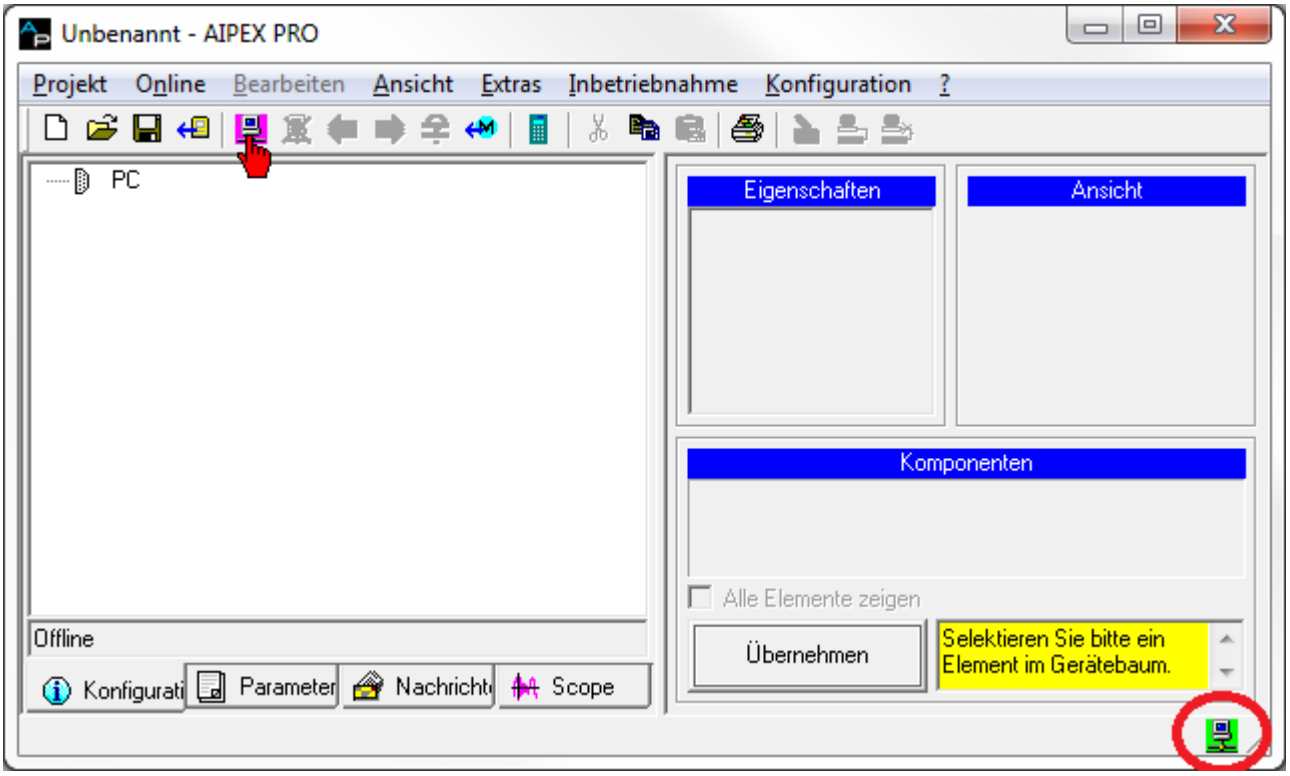
4.3.1.12.4 Kurvenscheibe

Voraussetzung Kurvenscheibe:
A5-Geräte mit Zusatzoption PCO (PLCopen) bzw. ein A5-Geräte mit Zusatzoption PNC (PLCopen CNC).

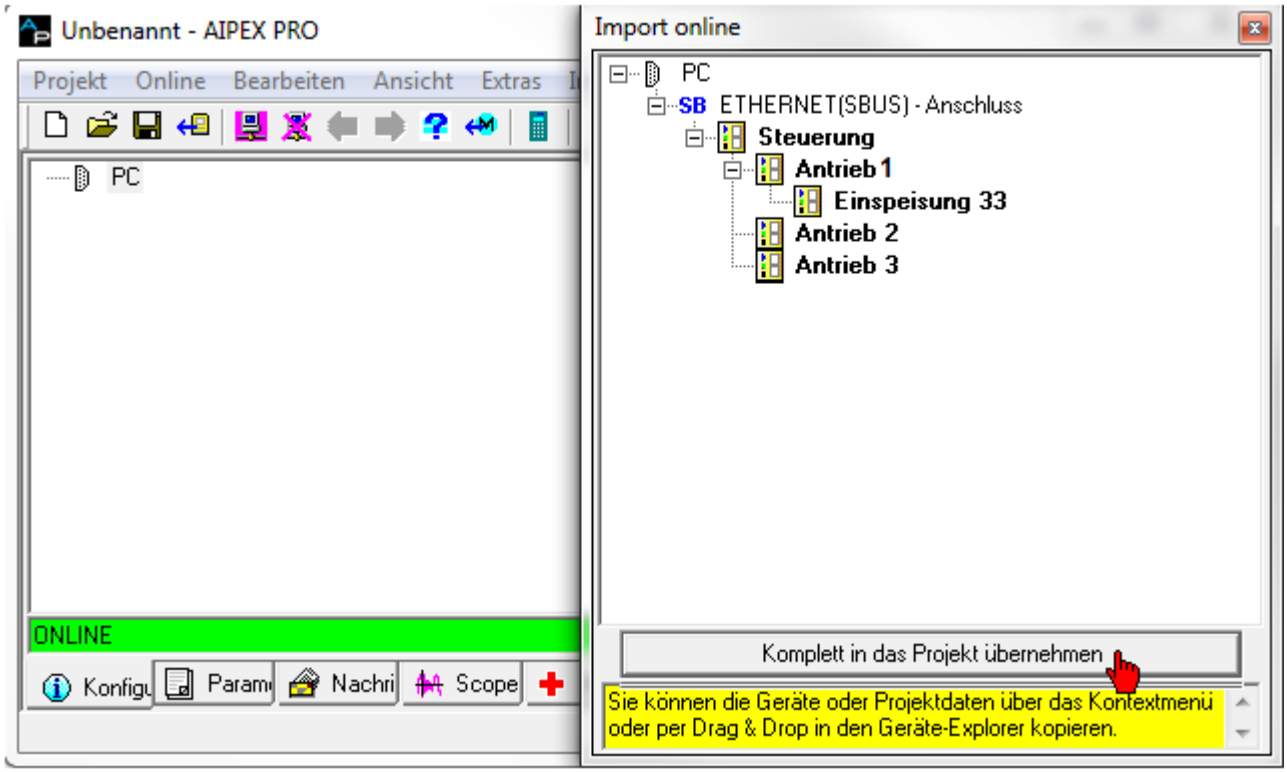
4.3.1.12.5 PLC Projekt importieren

Dieses Kapitel beschreibt, wie Sie ein PLC Projekt von einem anderen AIPEX PRO Projekt importieren und auf eine bestehende Applikation übertragen.

Drücken Sie den Taster **'Einloggen'** um eine Verbindung mit den verbundenen Geräten herzustellen.

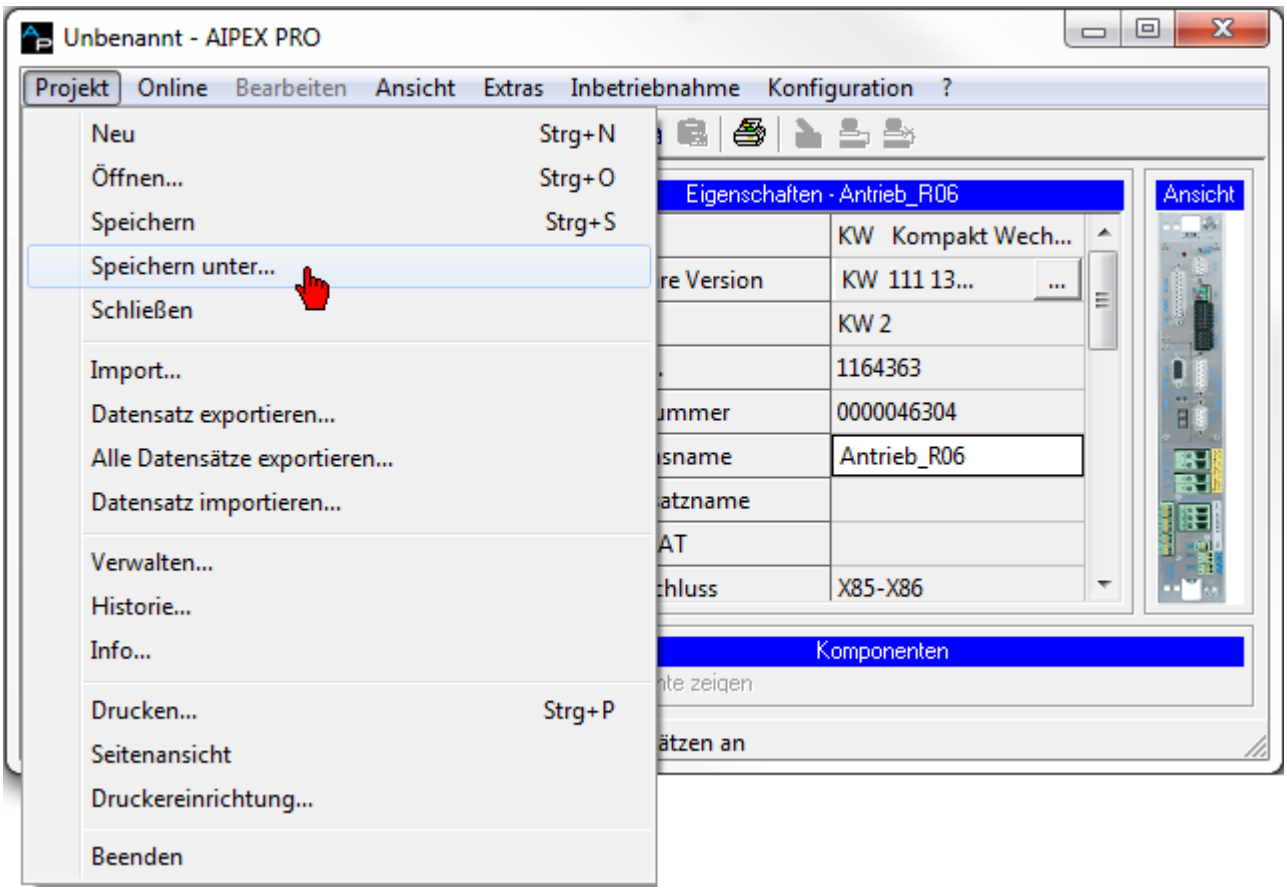


Prüfen Sie im Fenster 'Import Online' ob alle physikalisch vorhandenen Geräte erfasst sind.
Drücken Sie den Taster **'Komplett in das Projekt übernehmen'** um die Gerätedaten einzulesen.

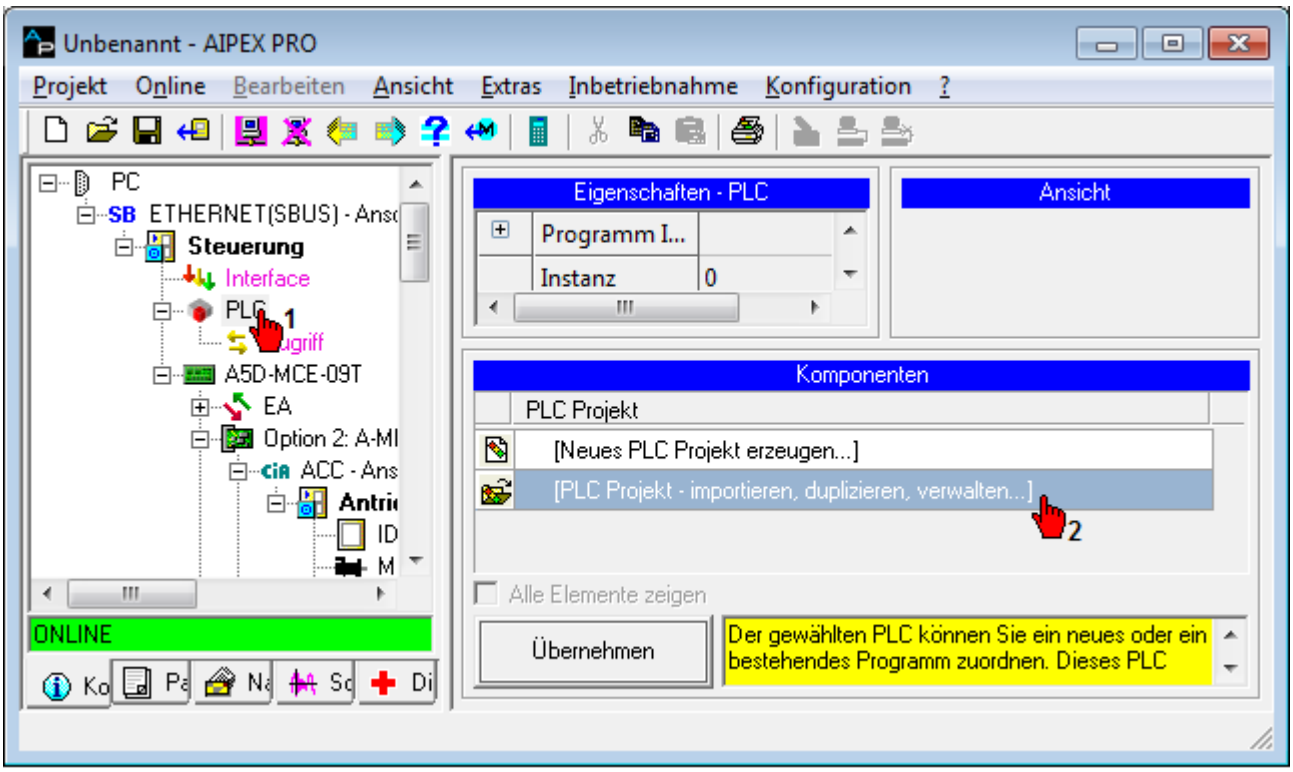


Prüfen Sie die Verkabelung und Systemzustände falls Geräte nicht angezeigt werden.

Speichern Sie das eingelesene Projekt unter **'Speichern unter...'** ab.

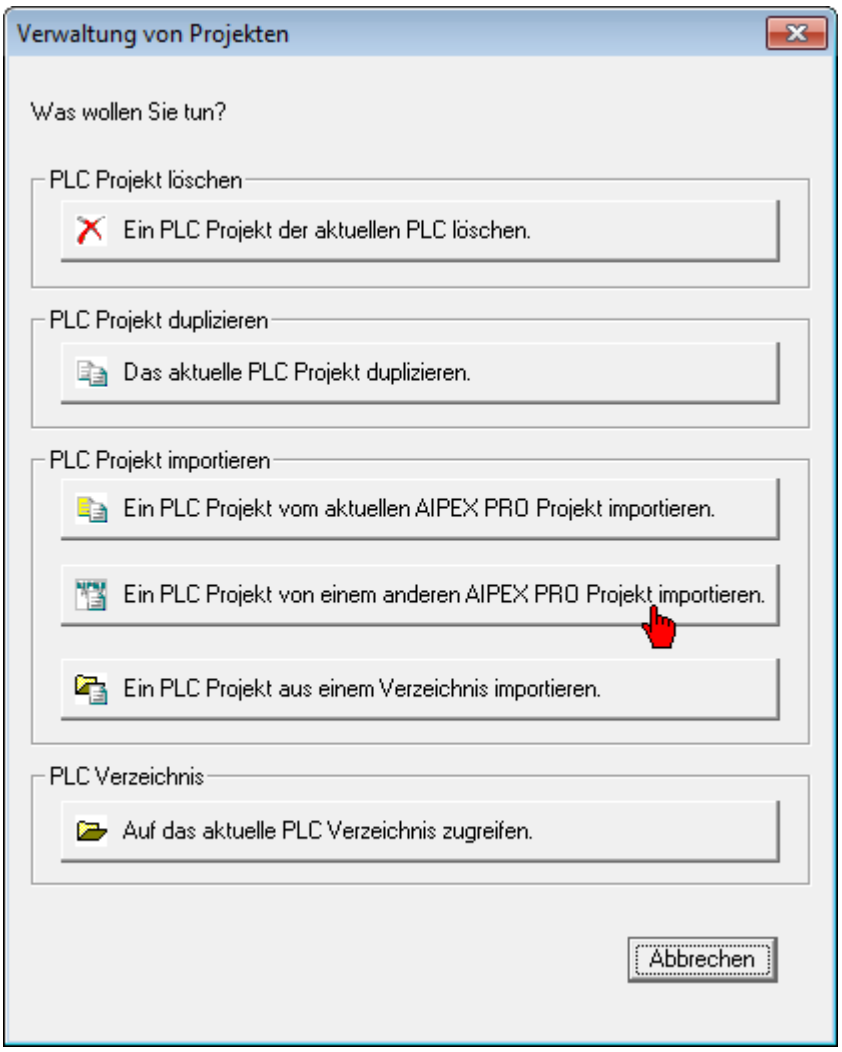


Klicken Sie auf das Icon **'PLC'** im Gerätebaum.
Klicken Sie auf **'PLC Projekt - importieren, dublizieren, verwalten ...'**

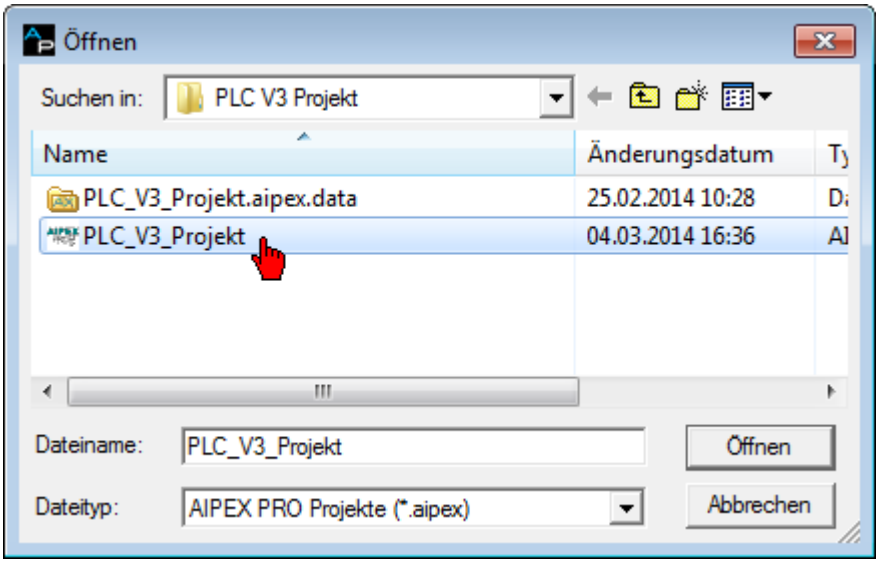


Icon '**PLC**' fehlt:
Klicken Sie mit der RMT in den Gerätebaum. Unter 'Anzeige auswählen' können Sie die Anzeige anpassen.
Siehe 'Programmübersicht - Anzeigefilter' auf Seite 39.

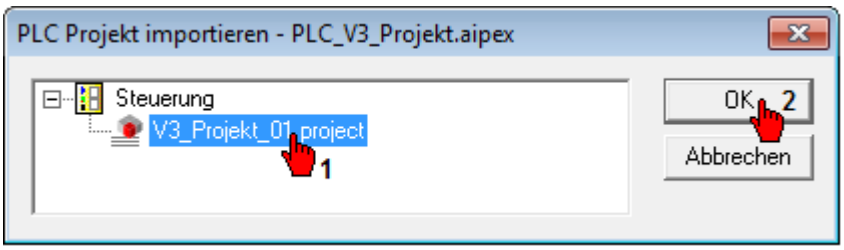
Klicken Sie auf **'Ein PLC Projekt von einem anderen AIPEX PRO Projekt importieren'** an.



Öffnen Sie das AIPEX PRO Projekt mit dem integrierten PLC Projekt.



Wählen Sie das PLC Projekt aus. Bestätigen Sie mit der Taste **'OK'** um das Programmiersystem CODESYS zu starten.



Bevor das PLC Programm auf die Steuerung übertragen wird, muss es fehlerfrei übersetzt und die Nachrichtenkonfiguration erstellt werden.

Starten Sie diesen Vorgang unter '**Konfiguration**' → '**Konfiguration erstellen**'.

Siehe 'PLC Programm und Nachrichtenkonfiguration erstellen' auf Seite 695.

4.3.1.13 Generelle Informationen

4.3.1.13.1 Versionierung CODESYS V3

Die in AIPEX PRO integrierten CODESYS V3 Versionen sind untereinander nur eingeschränkt kompatibel. Öffnen Sie deshalb bestehende CODESYS V3 Projekte immer mit der CODESYS V3 Version mit der das bestehende Projekt erstellt wurde.

Sind im bestehende CODESYS Projekt AFL Standardbausteine oder AFL Applikationsbausteine verbaut, muss zusätzlich die AFL AMK Function Library installiert sein. Die installierte AFL AMK Function Library muss kompatibel zur CODESYS V3 Version sein.

Für neue CODESYS V3 Projekt wird immer die aktuellste CODESYS V3 Version empfohlen.

Es dürfen nur von AMK freigegebene CODESYS V3 Versionen verwendet werden. Die von AMK freigegebenen CODESYS V3 Versionen sind in den AIPEX PRO Setups enthalten.

Versionsübersicht

AIPEX PRO + integrierte CODESYS V3 Version + kompatible AFL Version

AIPEX PRO Version	CODESYS V3 Version	CODESYS Profile	Kompatible AFL Version
3.04	3.5.10.4	CODESYS V3.5 **SP10** Patch 4 AIPEX PRO	AFL V4 Version 3.5.5.0 2015/41 (Teile-Nr. 206004)
	3.5.5.5	CODESYS V3.5 **SP5** Patch 5 AIPEX PRO	
	3.5.3.6	CODESYS V3.5 **SP3** Patch 6 AIPEX PRO	AFL V4 Version 3.5.3.0 2014/06 (Teile-Nr. 204786)
3.03	3.5.5.5	CODESYS V3.5 **SP5** Patch 5 AIPEX PRO	AFL V4 Version 3.5.5.0 2015/41 (Teile-Nr. 206004)
3.02	3.5.3.6	CODESYS V3.5 **SP3** Patch 6 AIPEX PRO	AFL V4 Version 3.5.3.0 2014/06 (Teile-Nr. 204786)
3.01			
3.00			

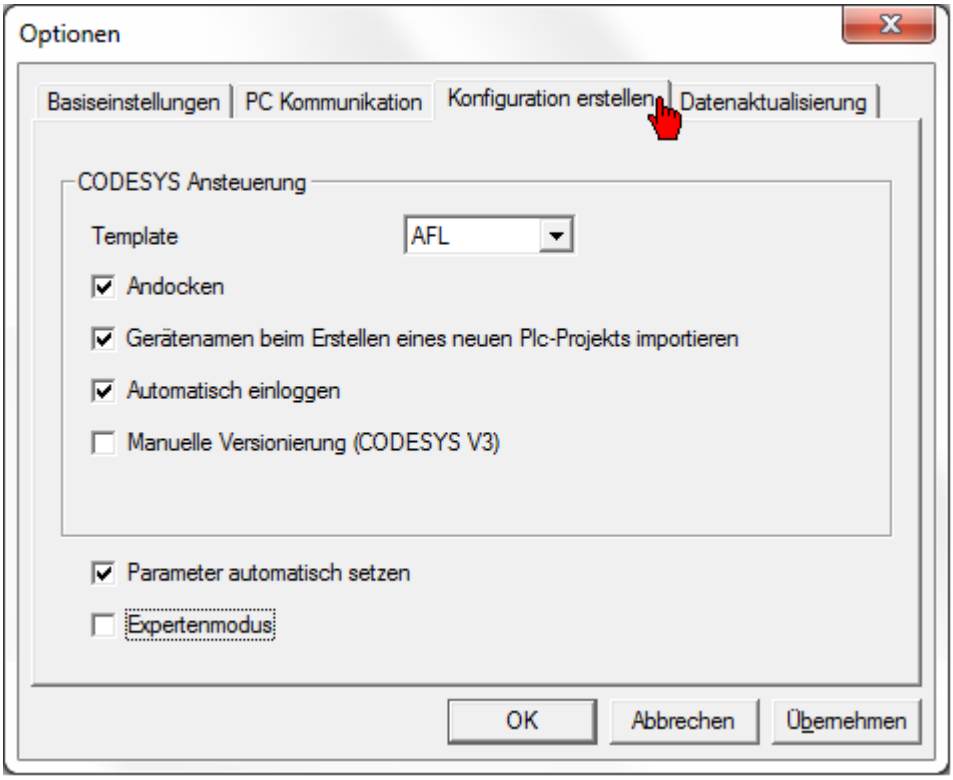
Steuerung	Firmware Version	CODESYS V3 Profil
iSA / A4	≥ 4.22	CODESYS V3.5 SP10 Patch 4 AIPEX PRO
iSA / A4	≤ 4.21	CODESYS V3.5 SP5 Patch 5 AIPEX PRO
A5 / A6	alle	CODESYS V3.5 SP10 Patch 4 AIPEX PRO (empfohlen) (mit Einschränkungen[1)])
		CODESYS V3.5 SP5 Patch 5 AIPEX PRO (mit Einschränkungen[1)])
		CODESYS V3.5 SP3 Patch 6 AIPEX PRO

1) Neue Features dieser CODESYS Version, die das Laufzeitsystem betreffen, werden nicht unterstützt.

Entsprechende CODESYS Version muss bei der Installation von AIPEX PRO mit installiert werden.

Ein AIPEX PRO Setup beinhaltet immer nur eine CODESYS V3 Version (siehe Tabelle: Versionsübersicht). Ältere CODESYS V3 Versionen müssen separat mit den dazugehörigen AIPEX PRO Setups installiert werden.
Die AFL AMK Function Library ist kein Bestandteil des AIPEX PRO Setups und muss separat installiert werden.

Ab AIPEX PRO Version 3.03 wird eine manuelle bzw. automatische CODESYS V3 Versionsauswahl angeboten
AIPEX PRO Menü 'Extras' → 'Optionen' → 'Konfiguration erstellen' → 'Manuelle Versionierung (CODESYS V3)'



Ab AIPEX PRO Version 3.03 wird eine manuelle bzw. automatische CODESYS V3 Versionsauswahl angeboten.
Manuelle Versionierung (CODESYS V3) 'deaktiviert' (Standardeinstellung)

- Neue CODESYS V3 Projekte werden immer mit der aktuell installierten CODESYS V3 Version erstellt.
- Ab AIPEX PRO Version 3.03 erkennt die Software automatisch mit welcher CODESYS V3 Version ein bestehendes CODESYS V3 Projekt erstellt wurde und öffnet automatisch die dazu passende CODESYS V3 Version. Ist die passende CODESYS V3 Version nicht installiert, wird das Projekt mit der aktuellsten CODESYS V3 Version geöffnet.

Manuelle Versionierung (CODESYS V3) 'aktiviert'
Vor jedem öffnen von CODESYS V3 durch AIPEX PRO Version ≥ 3.03 wird nachgefragt, welche installierte Version von CODESYS V3 man verwenden möchte.

- Bestehendes CODESYS V3 Projekt öffnen → Anwender muss eine installierte CODESYS V3 Version auswählen, mit der das bestehende Projekt geöffnet werden soll.
- Neues CODESYS V3 Projekt erstellen → Anwender muss eine installierte CODESYS V3 Version auswählen, mit der das neue Projekt erstellt werden soll.

Beispiel mit verschiedenen Versionen:

Sie benötigen die Funktionalität der AIPEX PRO Version 3.03. Zum Programmieren benötigen Sie aber die vorherige CODESYS V3 Version 3.5.3.6.

Vorgehensweise:

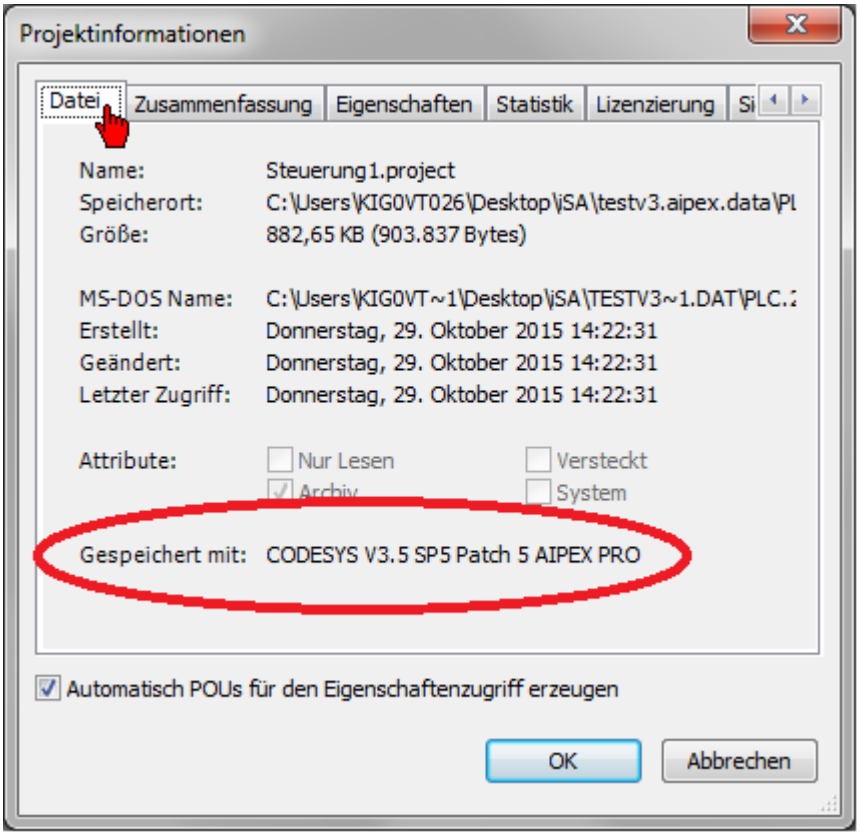
Installieren Sie zuerst das AIPEX PRO Setup 3.02 (enthält die CODESYS V3 Version 3.5.3.6). Anschließend muss das AIPEX PRO Setup 3.03 installiert werden.

Aktivieren Sie im AIPEX PRO Menü 'Extras' → 'Optionen' → 'Konfiguration erstellen' → 'Manuelle Versionierung (CODESYS V3)'

Beim öffnen des CODESYS V3 Projekts müssen Sie das Profile 'CODESYS V3.5 **SP3** Patch 6 AIPEX PRO' (CODESYS V3 Version 3.5.3.6) anwählen.

Im Projekt verwendete CODESYS Version bestimmen

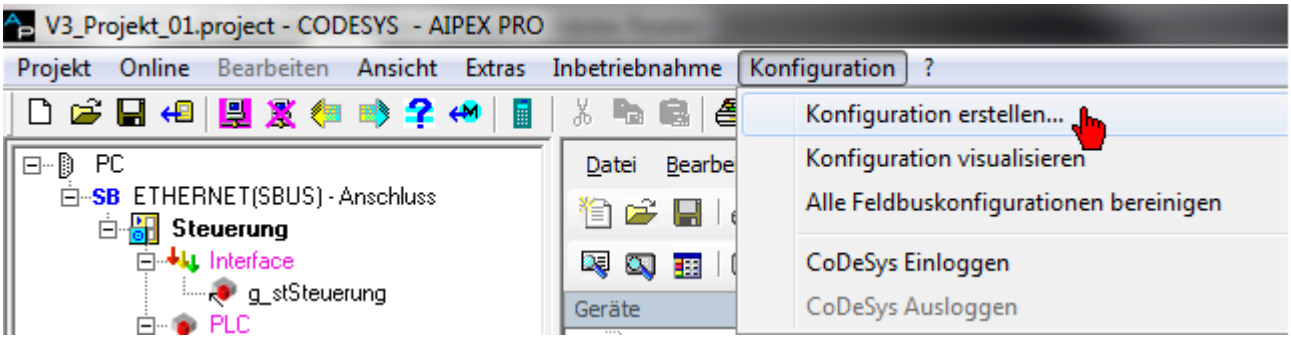
CODESYS V3 Menü 'Projekt' → 'Projekt information'



4.3.1.13.2 Automatische Buskonfiguration

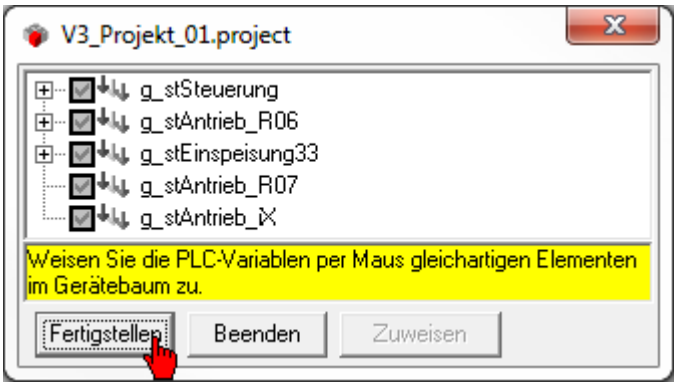
Die Zuordnung der physikalischen Geräten (AIPEX PRO) zu denen im PLC Programm verwendeten Devices, IO Variablen und PLC zu PLC Kommunikationsvariablen (in der Dokumentation als symbolischer Gerätenamen bezeichnet), erfolgt während des Buskonfigurationsvorganges. Die symbolischen Gerätenamen werden in der Steuerungskonfiguration angelegt.

Die automatische Buskonfiguration wird mit dem Menüpunkt 'Konfiguration erzeugen' gestartet.



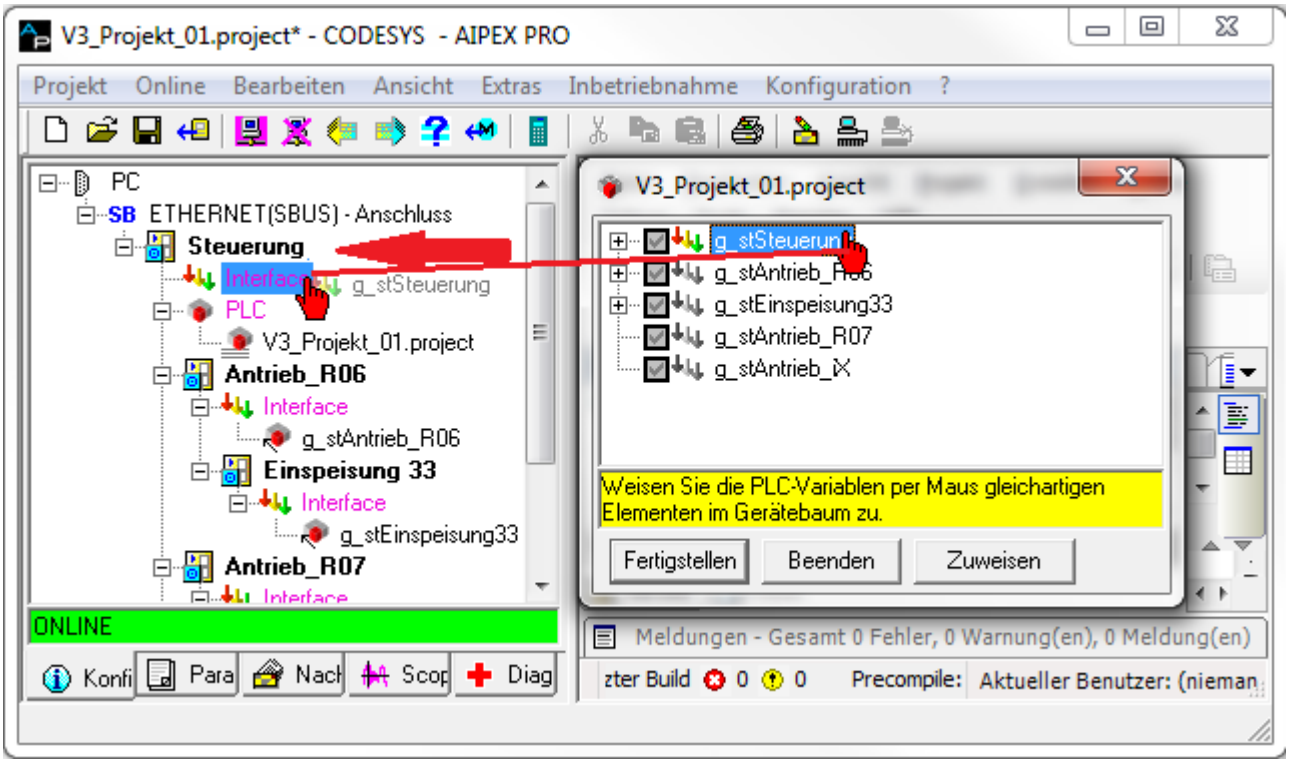
Hierbei werden die Variablen analysiert, die verknüpften symbolischen Gerätenamen ermittelt und im Zuordnungsfenster '*.project' dargestellt.

g_stSteuerung [Pfeile grau] Der Symbolische Gerätenamen ist einem Gerät im AIPEX PRO Gerätebaum zugewiesen



g_stSteuerung [Pfeile farbig] Der symbolische Gerätenamen ist nicht zugewiesen.

Die symbolischen Gerätenamen können per Maus (Drag & Drop), den entsprechenden Punkten des Gerätebaums (Interface, bzw. EA Unterelement) zugewiesen werden.



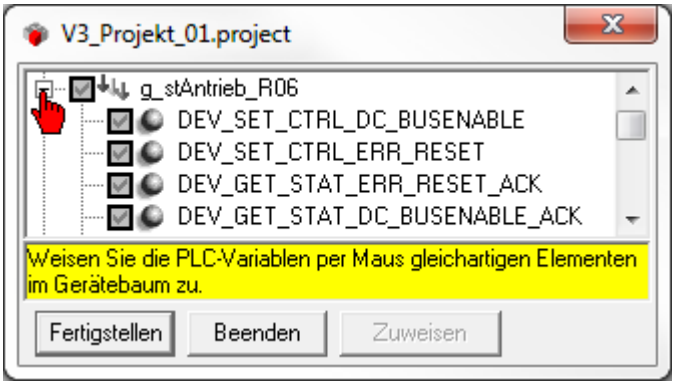
Zusätzliche Information:

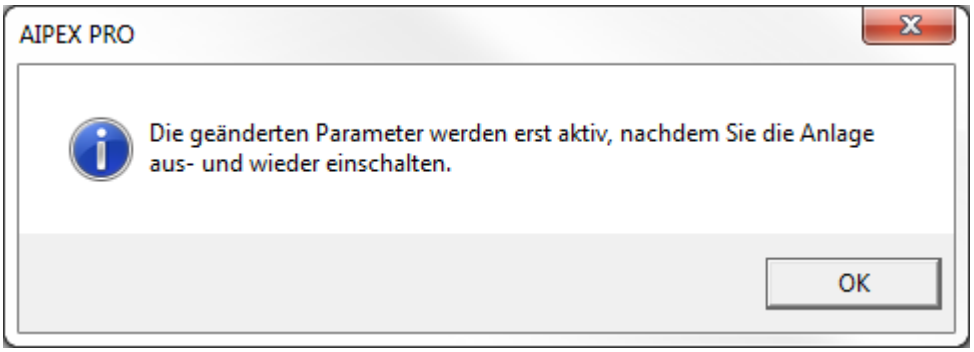
Klicken Sie auf das + Symbol. Sie sehen welche Funktionsbausteine einer Variablen zugewiesen sind.

Sofern aktiviert, ändert der Konfigurator automatisch die relevanten Parameter.

Aktivieren/deaktivieren bzw. Funktionsbaustein spezifische Änderungen:

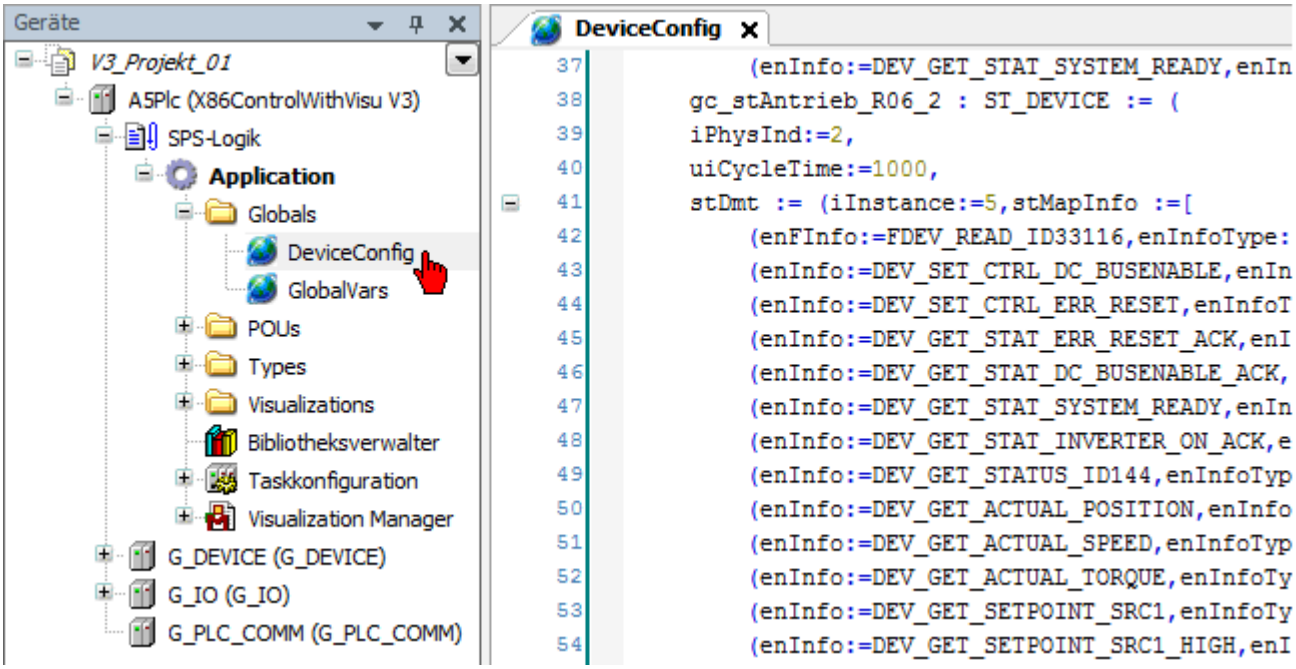
Siehe 'Konfiguration erstellen (projektspezifisch)' auf Seite 113.





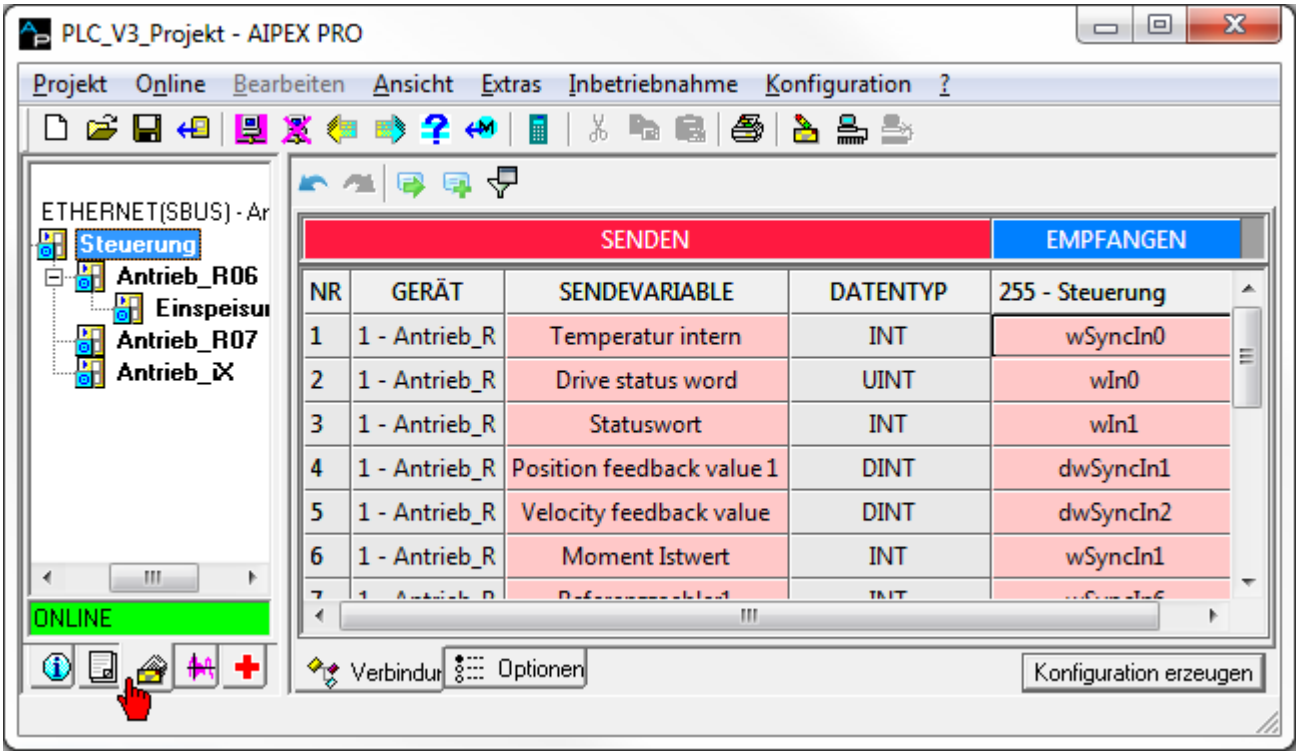
CODESYS V3

Die erstellte Buskonfiguration finden Sie unter 'Application' → 'Globals' → 'DeviceConfig'

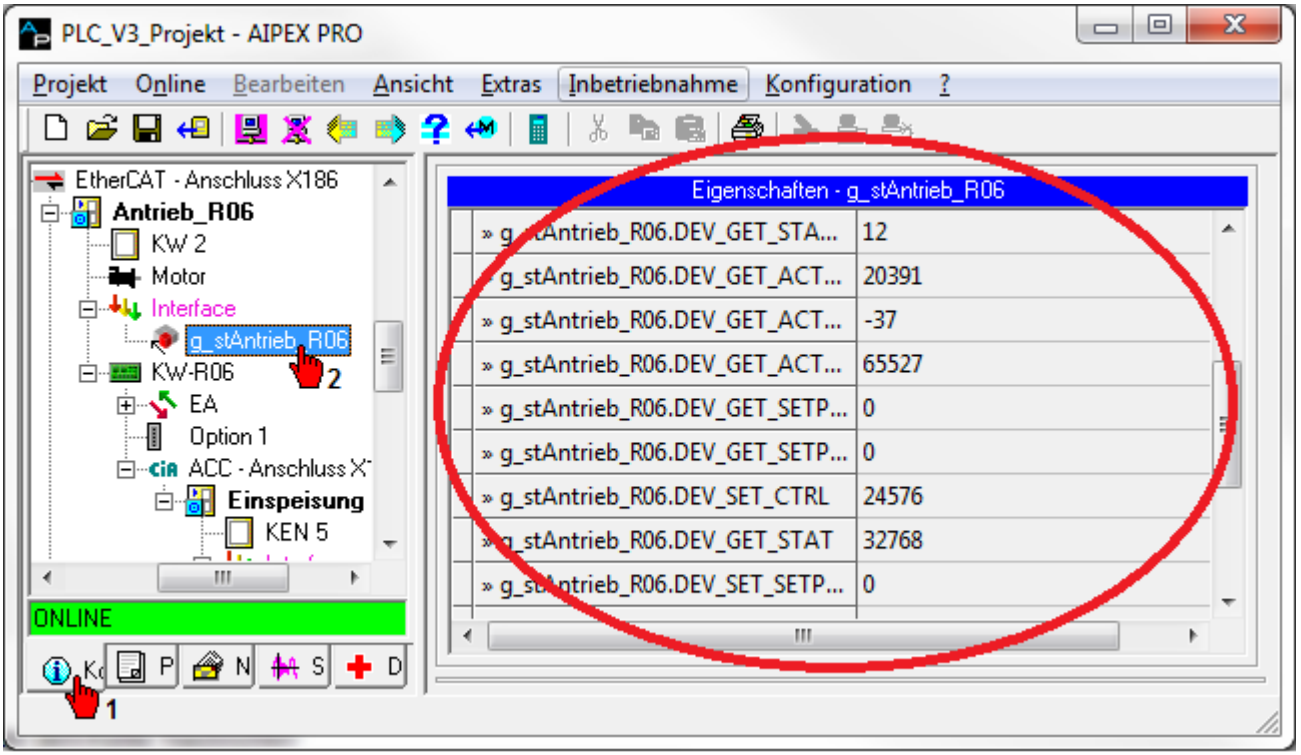


AIPEX PRO

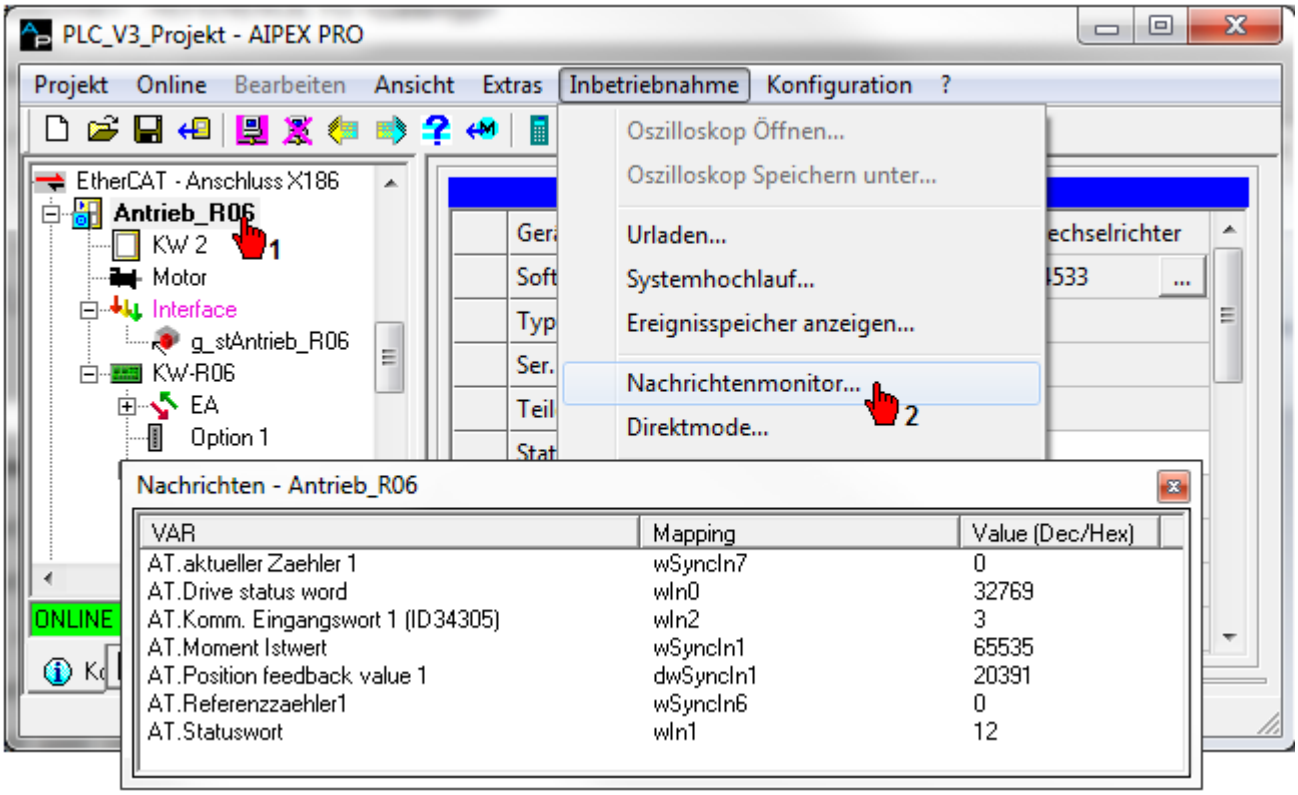
Das erstelle Nachrichtenkonfigurationsfile finden Sie unter dem Reiter 'Nachrichten'



Onlineanzeige der verwendeten Funktionsbausteine mit aktuellen Werten



Onlineanzeige der gemappten Variablen zwischen Slave und Master.



4.3.1.13.2.1 Einschränkung der automatischen Buskonfiguration

Um die Busnachrichten für die in CODESYS V3.5 verwendeten Funktionsbausteine zu ermitteln, wird eine statische Quell-Code-Analyse durchgeführt.

Bei dieser Analyse wird die Verknüpfung des symbolischen Gerätenamens mit einem Funktionsbaustein aufgelöst. Um diese Verknüpfung aufzulösen, muss (zur Entwurfszeit) der Weg, wie der symbolische Gerätename weitergereicht wird, eindeutig ermittelbar sein.

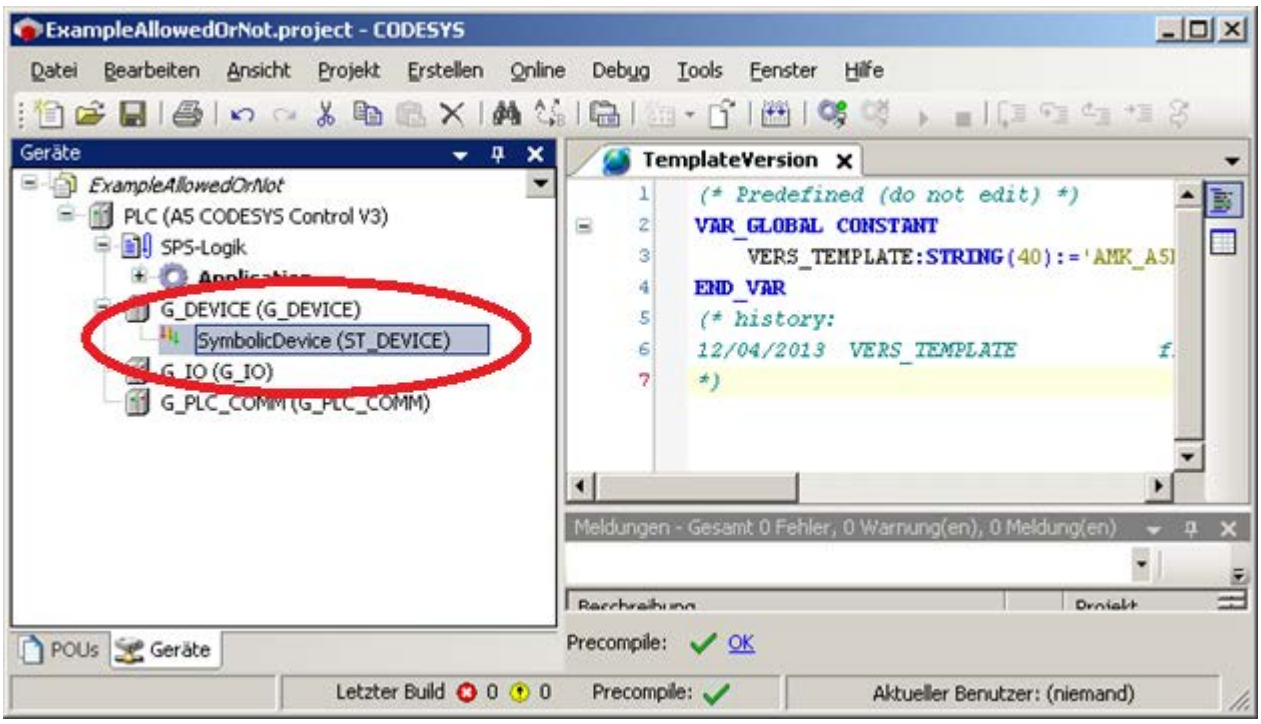
Für die automatische Buskonfiguration relevante Funktionsbausteine können in andere Bausteine integriert werden. Dabei ist zu beachten, dass die hierfür relevante 'stDevice' Variable direkt weitergereicht wird. Diese Variable muss im 'VAR_IN_OUT' Bereich des übergeordneten Bausteins deklariert werden und den Typ 'ST_DEVICE' besitzen.

Sollen Applikationsbausteine, die zur automatischen Buskonfiguration relevante Information enthalten, in einer eigenen 'Internen CODESYS V3.5-Bibliothek' bereitgestellt werden, so muss diese vom Typ 'library' sein. Der Typ 'compiled-library' ist nicht erlaubt!

Trotz korrekter IEC61131-3 Syntax sind manche Konstrukte für den Konfigurator nicht aufzulösen. Das heißt, es werden keine Busnachrichten für die beschriebenen Fälle generiert. Das PLC-Programm selbst ist jedoch funktionstüchtig. Lediglich der Zugriff auf die Busnachricht, im Rahmen des entsprechende Grundfunktionsbausteinaufrufs (mit 'boEnable := TRUE'), führt zu einer Fehlermeldung (iErrID := 1).

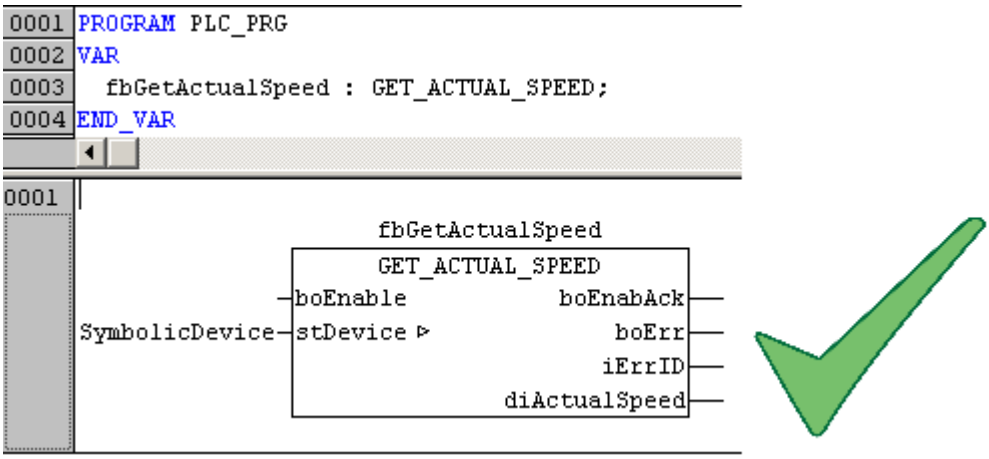
Im Folgenden werden diese Einschränkungen anhand von Beispielen aufgezeigt. Das symbolische Gerät für die folgenden Beispiele (Variable 'SymbolicDevice') ist in der nächsten Abbildung definiert.

Gerätedefinition in der Steuerungskonfiguration.

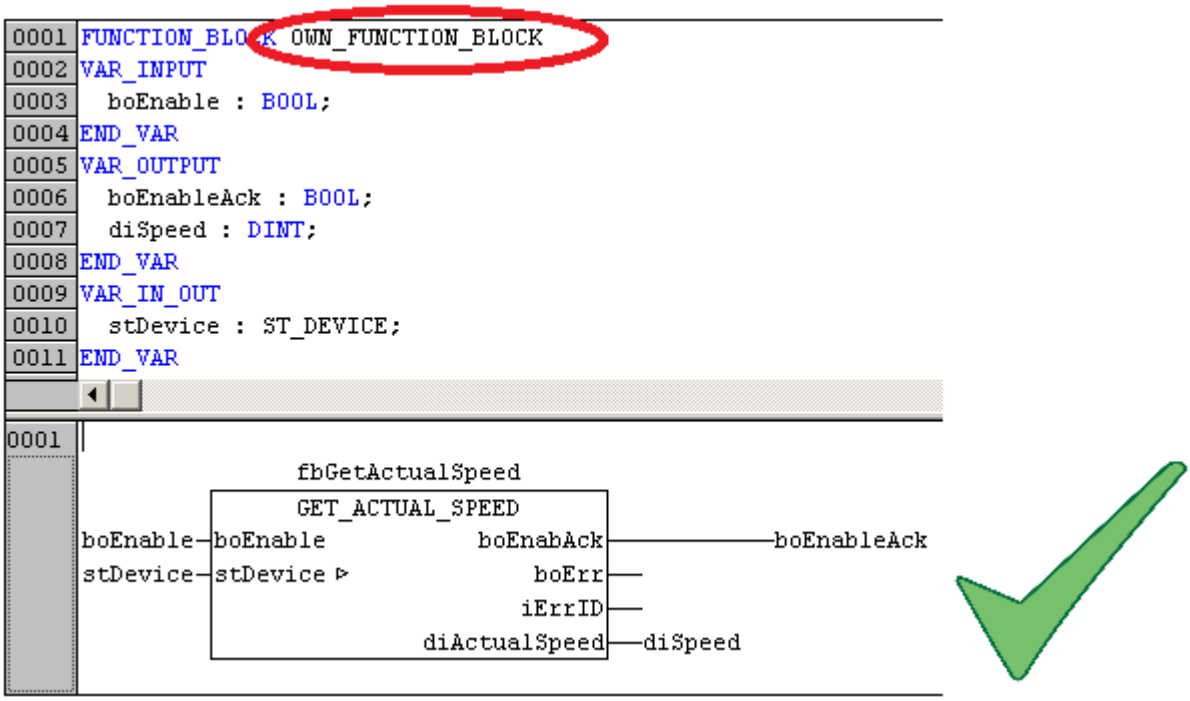


Bei folgenden Varianten ist eine korrekte Ermittlung von Busnachrichten möglich

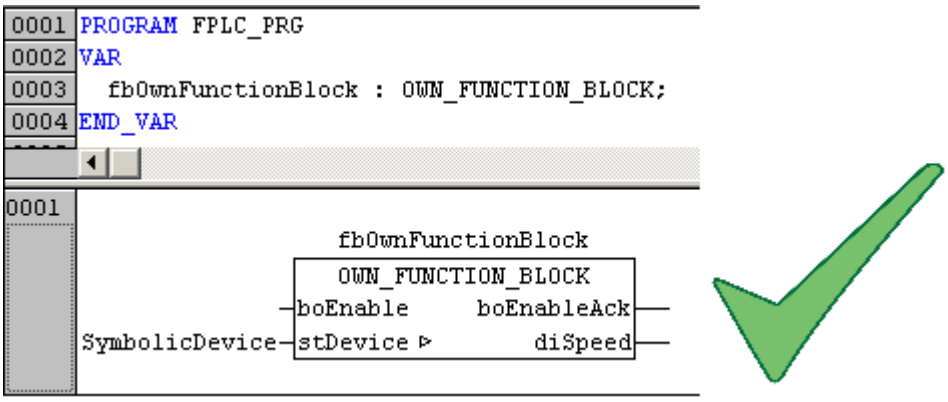
Variante 1: Aufruf eines Funktionsbausteins in dem zyklischen Programm 'PLC_PRG' bzw. 'FPLC_PRG' (Standard Bausteinaufruf)



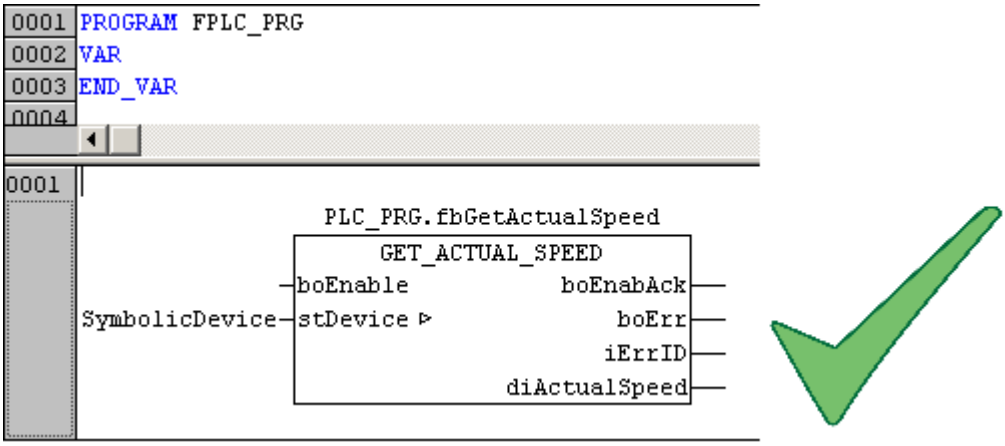
Variante 2: Aufruf gekapselt in eigenem Funktionsbaustein (Funktionsbausteindefinition)



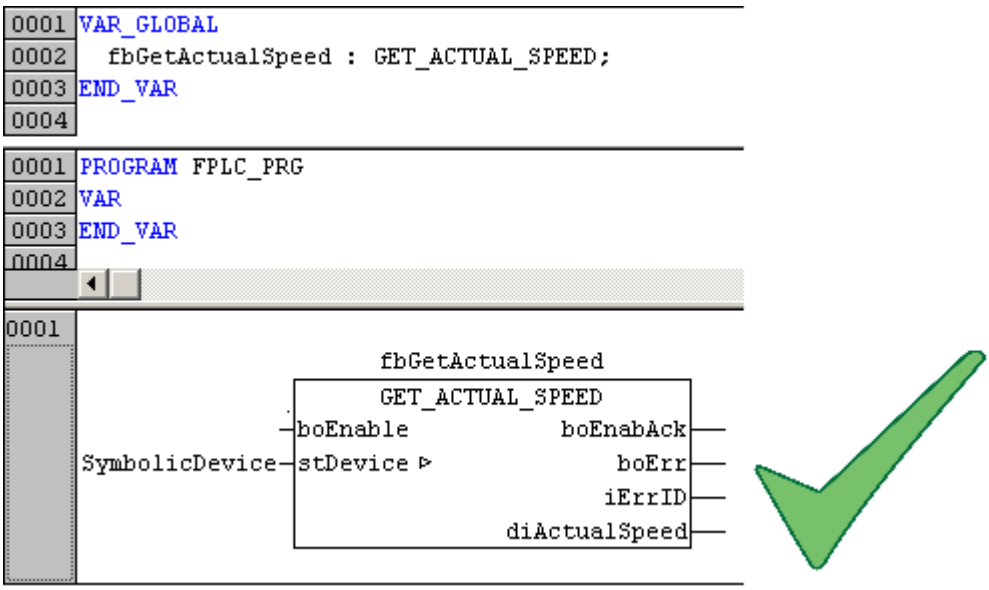
Aufruf des überlagerten Bausteins



Variante 3: Aufruf des Funktionsbausteins oder einer Aktion aus einem anderen Gültigkeitsbereich

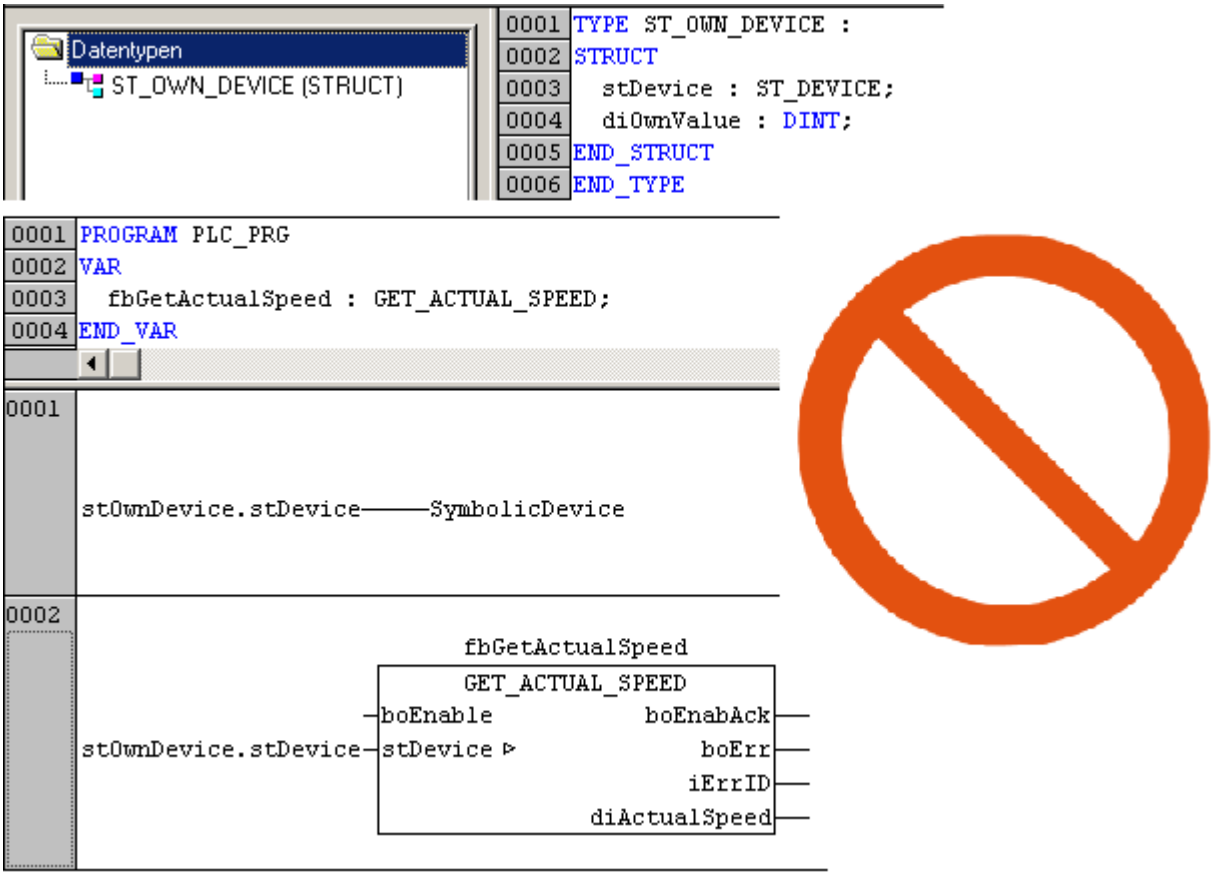


Variante 4: Funktionsblockinstanz ist global definiert



Bei folgender Variante ist keine Ermittlung von Busnachrichten möglich

Der symbolischen Gerätenamen ist in eine eigene Struktur eingebettet



4.3.1.13.3 Zielsystemauswahl für CODESYS V3

Für jede Steuerungsvariante gibt es ein passendes AMK Zielsystem (AMK-spezifische Gerätebeschreibung). Bei der Zielsystemauswahl muss zwischen dem in der Steuerung verbauten Prozessortyp und die zusätzlichen ab Werk freigeschalteten CODESYS Optionen unterschieden werden. Bei den Schnittstellen Optionen ist keine Unterscheidung erforderlich.

Beim Erstellen eines CODESYS Projekts wird in Abhängigkeit der ausgewählten Steuerungsvariante automatisch das dazugehörige AMK Zielsystem angewählt.

Das AMK Zielsystem muss mit der vorhandenen Steuerungsvariante (Hardware) exakt übereinstimmen. Das 'Einloggen' auf der Steuerung ist nicht möglich, wenn sich das eingestellte AMK Zielsystem und die Steuerungsvariante unterscheiden.

Beispiel:

Steuerungstyp:	Displaysteuerung A5 mit ATOM (X86 Intel) Prozessor	
Ab Werk freigeschaltete Optionen:	A5-VIS (Webvisualisierung) A5-PCO (PLCopen (CODESYS 'SM_PLCopen.lib'))	(Standard Displaysteuerung) (AMK Teile-Nr. O844)

Erforderliches AMK Zielsystem:
X86PLCopenControlWithVisu V3

4.3.1.13.3.1 Auswahl AMK Zielsysteme

Zusammenhang Steuerung, Prozessortyp und Zielsystem

Steuerung	Prozessor	AMK Zielsystem
A4	ARM Prozessor	ARM...
A5	ATOM (X86 Intel) Prozessor	X86...
A6	ATOM (X86 Intel) Prozessor	X86...
iSA	ARM Prozessor	ARM...

Verfügbare CODESYS Optionen

Option	Beschreibung
A4-VIS [1)] A5-VIS [1)] A6-VIS [1)] iSA-VIS	Webvisualisierung
A4-PCO A5-PCO A6-PCO iSA-PCO	PLCopen (CODESYS 'SM_PLCopen.lib')
A5-PNC A6-PNC iSA-PNC	Numerical Control Motion (A5-PCO integriert) für CODESYS V3

1) Ax-VIS ab Werk für die A5/A6 Displaysteuerungen freigeschaltet.

Weiterführende Informationen zu den CODESYS Optionen:
Für AMKAMAC A-Serie: Siehe Dokument Produktbeschreibung Steuerungen A4 / A5 / A6 (Teile-Nr. 202975)
Für AMKASMART iSA: Siehe Dokument Produktbeschreibung iSA (Teile-Nr. 205670)

A4 ARM Prozessor

Installierte Optionen	AMK Zielsystem für CODESYS V3	Beschreibung
-	X86Control V3	Steuerung A4 ohne Zusatzoption, ohne Option VIS (Visualisierung)
A4-VIS	X86ControlWithVisu V3	Steuerung A4 ohne Zusatzoption, mit Option VIS (Visualisierung)
A4-PCO	X86PLCopenControl V3	Steuerung A4 mit Zusatzoption PCO (PLCopen), ohne Option VIS (Visualisierung)
A4-VIS, A4-PCO	X86PLCopenControlWithVisu V3	Steuerung A4 mit Zusatzoption PCO (PLCopen), mit Option VIS (Visualisierung)
A4-PNC	X86PLCopenCncControl V3	Steuerung A4 mit Zusatzoption PNC (PLCopen CNC), ohne Option VIS (Visualisierung)
A4-VIS, A4-PNC	X86PLCopenCncControlWithVisu V3	Steuerung A4 mit Zusatzoption PNC (PLCopen CNC), mit Option VIS (Visualisierung)

A5/A6 mit ATOM (X86 Intel) Prozessor

Installierte Optionen	AMK Zielsystem für CODESYS V3	Beschreibung
-	X86Control V3	Steuerung A5/A6 ohne Zusatzoption, ohne Option VIS (Visualisierung)
A5/A6-VIS	X86ControlWithVisu V3	Steuerung A5/A6 ohne Zusatzoption, mit Option VIS (Visualisierung)
A5/A6-PCO	X86PLCopenControl V3	Steuerung A5/A6 mit Zusatzoption PCO (PLCopen), ohne Option VIS (Visualisierung)
A5/A6-VIS, A5/A6-PCO	X86PLCopenControlWithVisu V3	Steuerung A5/A6 mit Zusatzoption PCO (PLCopen), mit Option VIS (Visualisierung)
A5/A6-PNC	X86PLCopenCncControl V3	Steuerung A5/A6 mit Zusatzoption PNC (PLCopen CNC), ohne Option VIS (Visualisierung)
A5/A6-VIS, A5/A6-PNC	X86PLCopenCncControlWithVisu V3	Steuerung A5/A6 mit Zusatzoption PNC (PLCopen CNC), mit Option VIS (Visualisierung)

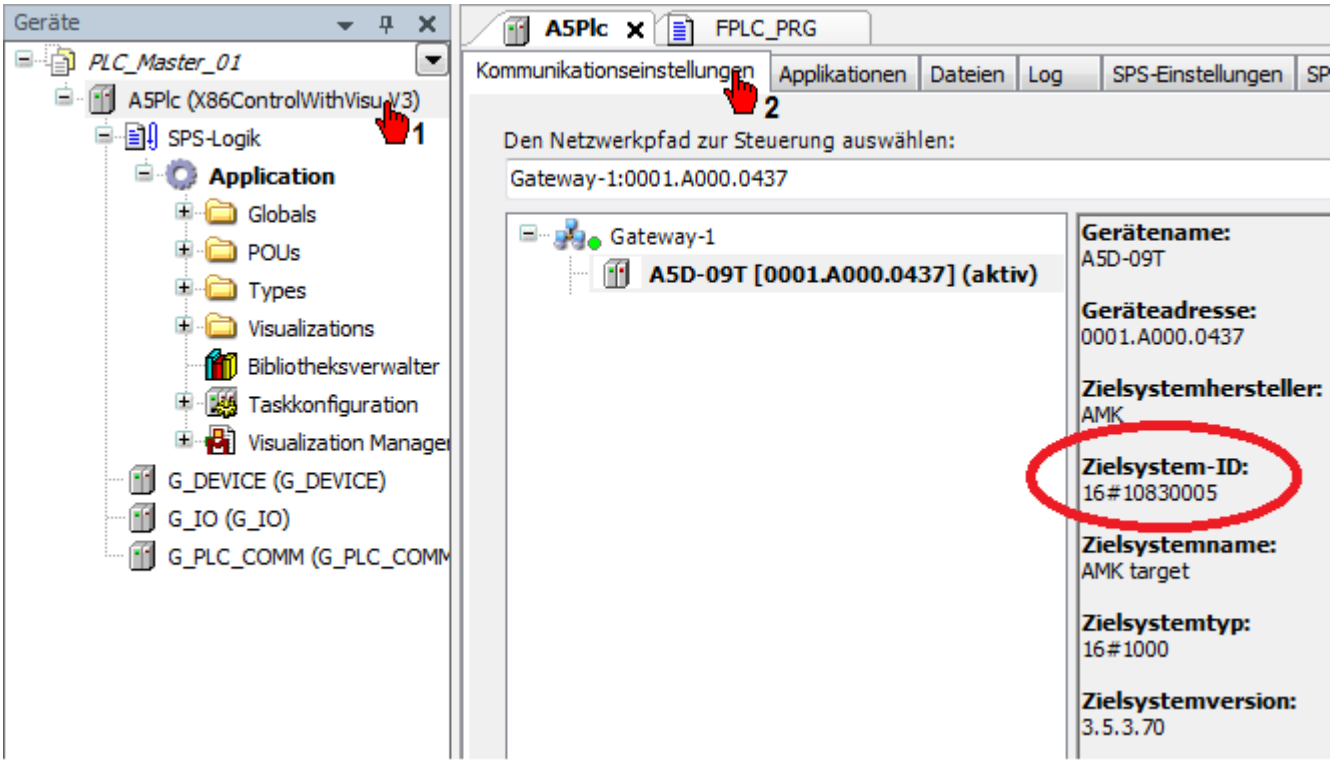
iSA mit ARM Prozessor

Installierte Optionen	AMK Zielsystem für CODESYS V3	Beschreibung
-	ArmControl V3	Steuerung iSA ohne Zusatzoption, ohne Option VIS (Visualisierung)
iSA-VIS	ArmControlWithVisu V3	Steuerung iSA ohne Zusatzoption, mit Option VIS (Visualisierung)
iSA-PCO	ArmPLCopenControl V3	Steuerung iSA mit Zusatzoption PCO (PLCopen), ohne Option VIS (Visualisierung)
iSA-VIS, iSA-PCO	ArmPLCopenControlWithVisu V3	Steuerung iSA mit Zusatzoption PCO (PLCopen), mit Option VIS (Visualisierung)
iSA-PNC	ArmPLCopenCncControl V3	Steuerung iSA mit Zusatzoption PNC (PLCopen CNC), ohne Option VIS (Visualisierung)
iSA-VIS, iSA-PNC	ArmPLCopenCncControlWithVisu V3	Steuerung iSA mit Zusatzoption PNC (PLCopen CNC), mit Option VIS (Visualisierung)

4.3.1.13.3.2 Geräteidentifizierung

Die Steuerungen werden basierend auf ihrer Zielsystem-ID und ihrem Zielsystemtyp identifiziert. Gemäß nachfolgender Tabelle besteht folgender Zusammenhang:

AMK Zielsystem	Zielsystem-ID	Zielsystemtyp	
X86Control V3 X86ControlWithVisu V3	16#10830006 16#10830005	16#1000	Steuerung
X86PLCopenControl V3 X86PLCopenControlWithVisu V3	16#10830004 16#10830002	16#1006	Softmotion Steuerung
X86PLCopenCncControl V3 X86PLCopenCncControlWithVisu V3	16#10830003 16#10830001	16#1006	Softmotion Steuerung
ArmControl V3 ArmControlWithVisu V3	16#10830016 16#10830015	16#1000	Steuerung
ArmPLCopenControl V3 ArmPLCopenControlWithVisu V3	16#10830014 16#10830012	16#1006	Softmotion Steuerung
ArmPLCopenCncControl V3 ArmPLCopenCncControlWithVisu V3	16#10830013 16#10830011	16#1006	Softmotion Steuerung



4.3.1.13.3.3 AIPEX PRO CODESYS Projekte mit CODESYS Optionen

Der Begriff CODESYS Optionen steht für ab Werk freigeschaltete Optionen wie PLCopen, Numerical Control Motion oder Webvisualisierung.

Die Steuerungsvariante im AIPEX PRO Gerätebaum ist die Basis für ein neues PLC Projekt. Das Template und das AMK Zielsystem wird automatisch an diesen Steuerungstyp und an aktive CODESYS Optionen angepasst.

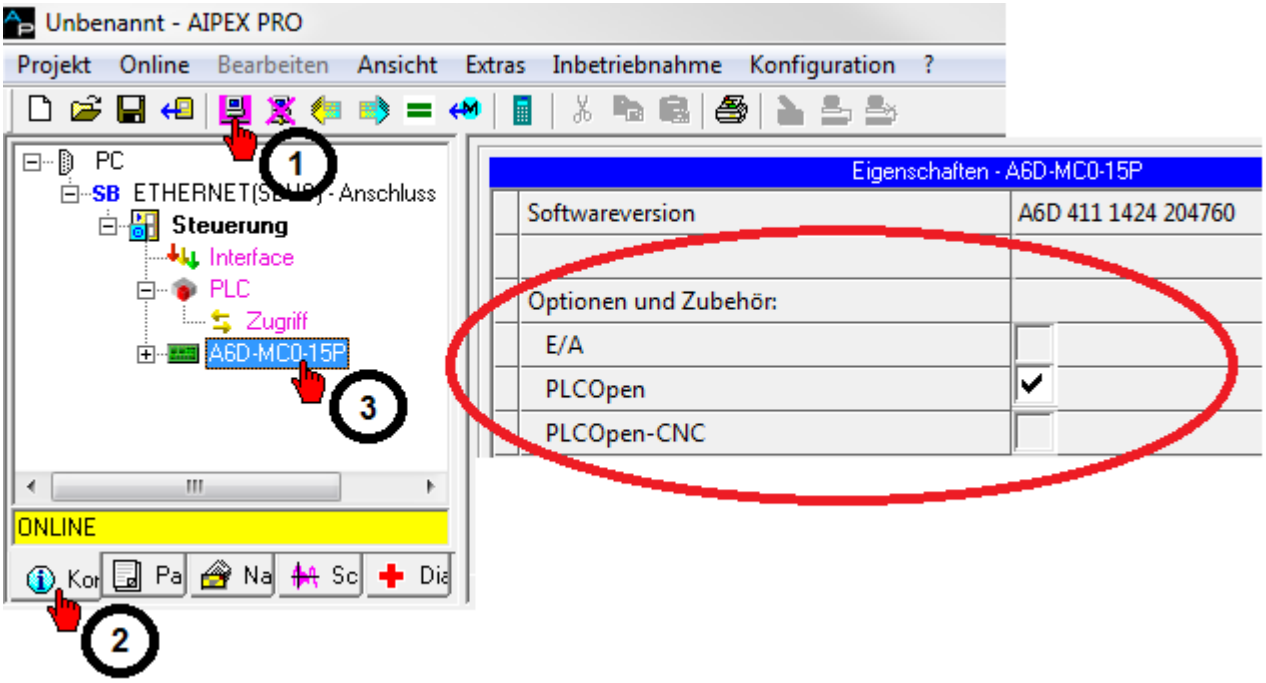
Eine nachträgliche Änderungen der Steuerungsvariante hat keine Auswirkung auf das CODESYS Projekt.

Erstellen Sie für eine neue Steuerungsvariante ein neues AIPEX PRO Projekt. Anschließend können Sie den bestehenden Programmcode in das neue CODESYS Projekt übernehmen.

Das AMK Zielsystem muss mit der vorhandenen Steuerungsvariante (Hardware) exakt übereinstimmen. Das 'Einloggen' auf der Steuerung ist nicht möglich, wenn sich das eingestellte AMK Zielsystem und die Steuerungsvariante unterscheiden.

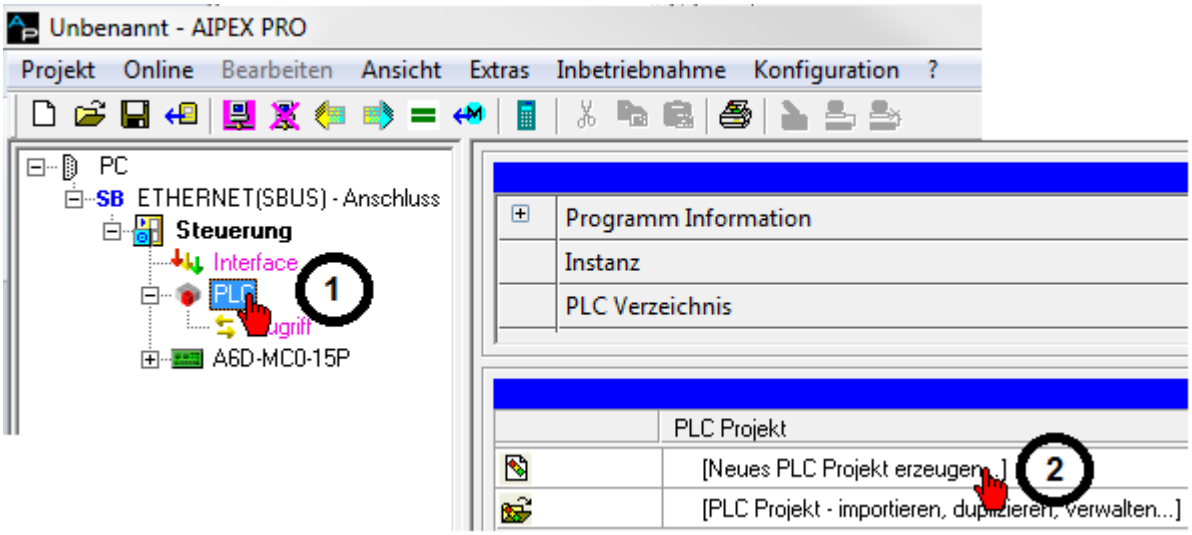
Neues Online - Projekt

Beim 'Online' - Scan mit AIPEX PRO werden die aktiven und ab Werk freigeschalteten CODESYS Optionen der Steuerung automatisch in das AIPEX PRO Projekt übernommen.



'Neues PLC Projekt erzeugen'

Templates und Zielsystemeinstellungen werden automatisch an die eingelesene Steuerungsvariante angepasst.

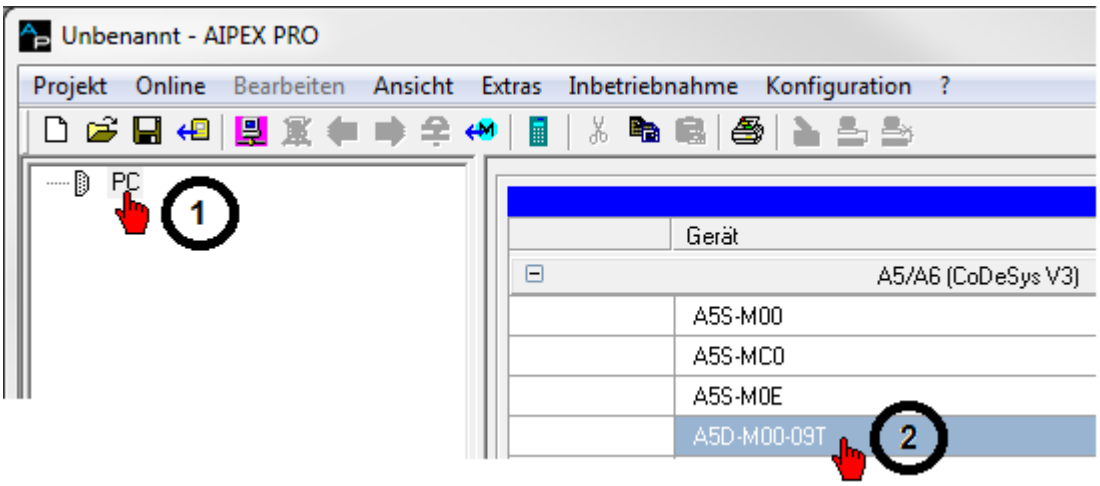


Neues Offline - Projekt

Bei einem Offline - Projekt muss der ausgewählte Steuerungstyp und die angewählten CODESYS Optionen zwingend mit der Steuerungsvariante (Hardware) übereinstimmen damit das Einloggen an der Steuerung und das Übertragen eines neu erstellten PLC Projektes möglich ist.

Vorgehensweise bei einem neuen Offline - Projekt:

Fügen Sie die PLC Steuerung ein



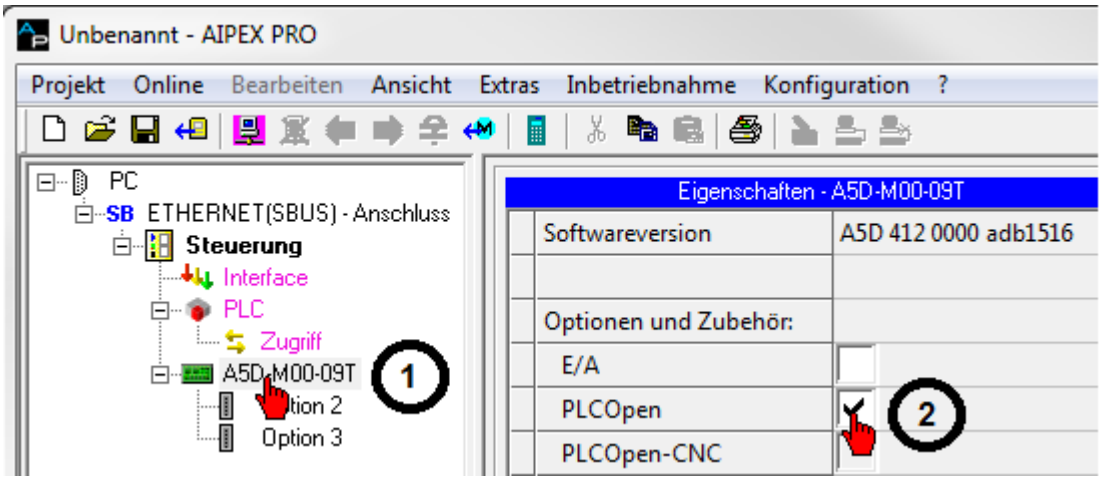
CODESYS Optionen auswählen

Wählen Sie alle CODESYS Optionen an, die in der Steuerungsvariante (Hardware) verfügbar sind.

Nachdem das PLC Projekt erzeugt ist (siehe nächster Schritt), haben nachträgliche Änderungen keinen Einfluss mehr auf das Template und das Zielsystem.

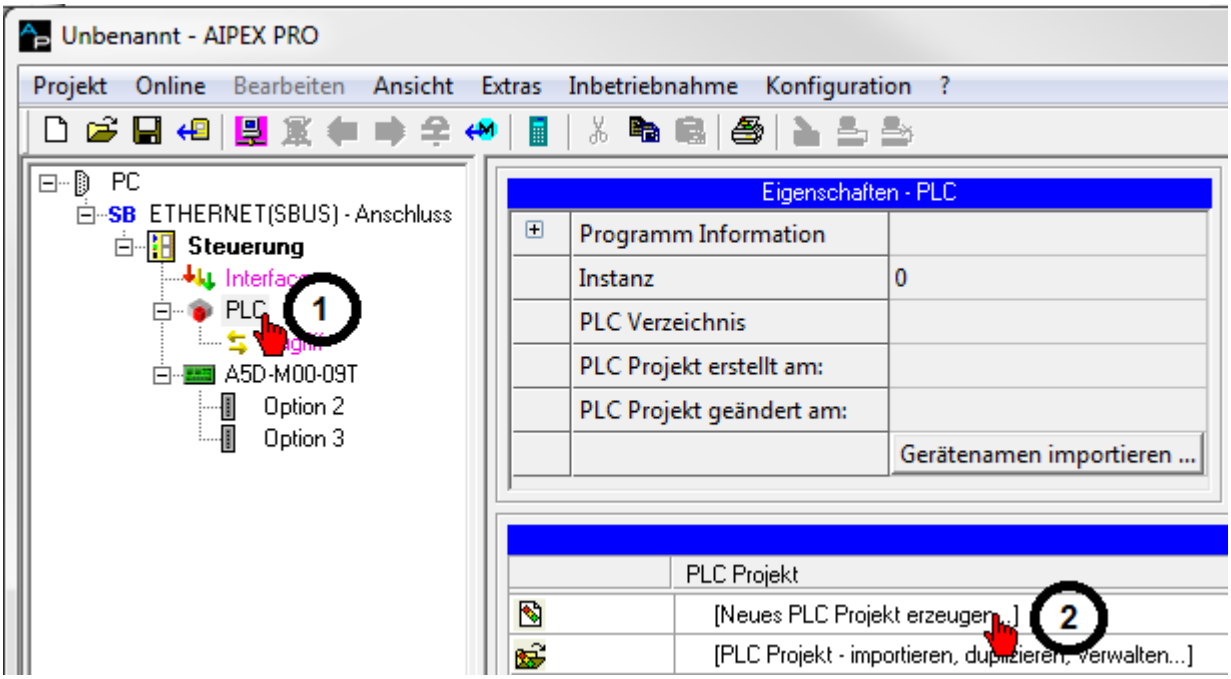
Es können nur CODESYS Optionen verwendet werden, die bei der Bestellung der Steuerung berücksichtigt wurden und ab Werk freigeschalteten sind.

Beispiel: Steuerung A5D-M00-09T mit CODESYS Option PLCopen (CODESYS 'SM_PLCopen.lib')



Neues PLC Projekt erzeugen

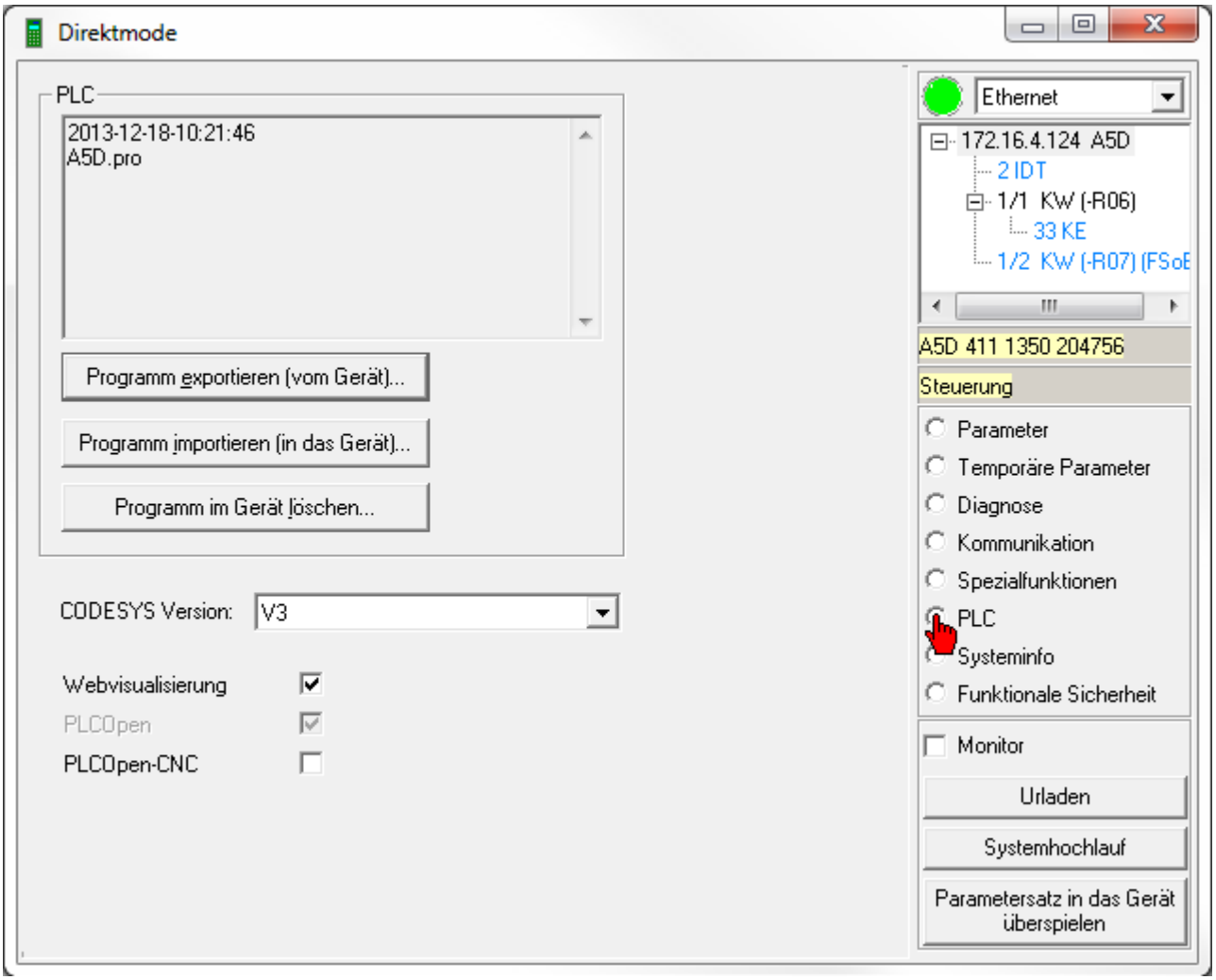
Templates und Zielsystemeinstellungen werden automatisch an die ausgewählte Steuerung und CODESYS Optionen angepasst.



4.3.1.13.3.4 CODESYS Optionen deaktivieren

Die ab Werk freigeschalteten CODESYS Optionen können vom Anwender deaktiviert und bei Bedarf wieder aktiviert werden.

Öffnen Sie dazu den AIPEX PRO 'Direktmode', Menü 'PLC'



Mit Hilfe der Checkboxen können die ab Werk freigeschalteten CODESYS Optionen deaktiviert und bei Bedarf wieder aktiviert werden.

Nach dem de- oder aktivieren muss die Steuerung neu gestartet werden (24 VDC AUS/EIN).

Beispiel:

Aus einer iSA Softmotion Steuerung mit Zusatzoption PCO (PLCopen) und Option VIS (Visualisierung), Zielsystem: ArmPLCopenControlWithVisu V3

wird nach Abwahl der Ckeckbox 'PLCopen' eine iSA Steuerung mit Zusatzoption VIS (Visualisierung), Zielsystem: ArmControlWithVisu V3.

Die CODESYS Optionsauswahl im AIPEX PRO 'Direktmode' Menü 'PLC' greift auf die ID34175 'Steuerungseinstellungen' Bits 8..10 zu.

Die nachfolgende Tabelle zeigt wie durch Setzen der Bits 8..10 einem ursprünglich Steuerungstyp und dem zugehörigen AMK Zielsystem ein neuer Steuerungstyp mit neuem AMK Zielsystem zugeordnet wird.

Beispiel:

Ab Werk freigeschaltete CODESYS Option PLCopen (CODESYS 'SM_PLCopen.lib')

Bit9 = 1 = Option PLCopen deaktiviert

Bit9 = 0 = Option PLCopen aktiviert

Installierte Optionen	AMK Zielsystem für CODESYS im Auslieferungszustand	A5/A6 -VIS	A5/A6 -PCO	A5/A6 -PNC	AMK Zielsystem für CODESYS nach Deaktivierung von Optionen
-	X86Control V3	-	-	-	X86Control V3
A5/A6-VIS	X86ControlWithVisu V3	Bit 8 = 1	-	-	X86Control V3
A5/A6-PCO	X86PLCopenControl V3	-	Bit 9 = 1	-	X86Control V3
A5/A6-VIS, A5/A6-PCO	X86PLCopenControlWithVisu V3	Bit 8 = 1	-	-	X86PLCopenControl V3
		-	Bit 9 = 1	-	X86ControlWithVisu V3
		Bit 8 = 1	Bit 9 = 1	-	X86Control V3
A5/A6-PNC	X86PLCopenCncControl V3	-	-	Bit 10 = 1	X86PLCopenControl V3
		-	Bit 9 = 1	Bit 10 = 1	X86Control V3
A5/A6-VIS, A5/A6-PNC	X86PLCopenCncControlWithVisu V3	Bit 8 = 1	-	-	X86PLCopenCncControl V3
		-	-	Bit 10 = 1	X86PLCopenControlWithVisu V3
		Bit 8 = 1	-	Bit 10 = 1	X86PLCopenControl V3
		-	Bit 9 = 1	Bit 10 = 1	X86ControlWithVisu V3
		Bit 8 = 1	Bit 9 = 1	Bit 10 = 1	X86Control V3

Installierte Optionen	CODESYS Zielsystem im Auslieferungszustand	iSA -VIS	iSA -PCO	iSA -PNC	Neues CODESYS Zielsystem nach Deaktivierung von Optionen
-	ArmControl V3	-	-	-	ArmControl V3
iSA-VIS	ArmControlWithVisu V3	Bit 8 = 1	-	-	ArmControl V3
iSA-PCO	ArmPLCopenControl V3	-	Bit 9 = 1	-	ArmControl V3
iSA-VIS, iSA-PCO	ArmPLCopenControlWithVisu V3	Bit 8 = 1	-	-	ArmPLCopenControl V3
		-	Bit 9 = 1	-	ArmControlWithVisu V3
		Bit 8 = 1	Bit 9 = 1	-	ArmControl V3
iSA-PNC	ArmPLCopenCncControl V3	-	-	Bit 10 = 1	ArmPLCopenControl V3
		-	Bit 9 = 1	Bit 10 = 1	ArmControl V3
iSA-VIS, iSA-PNC	ArmPLCopenCncControlWithVisu V3	Bit 8 = 1	-	-	ArmPLCopenCncControl V3
		-	-	Bit 10 = 1	ArmPLCopenControlWithVisu V3
		Bit 8 = 1	-	Bit 10 = 1	ArmPLCopenControl V3
		-	Bit 9 = 1	Bit 10 = 1	ArmControlWithVisu V3
		Bit 8 = 1	Bit 9 = 1	Bit 10 = 1	ArmControl V3

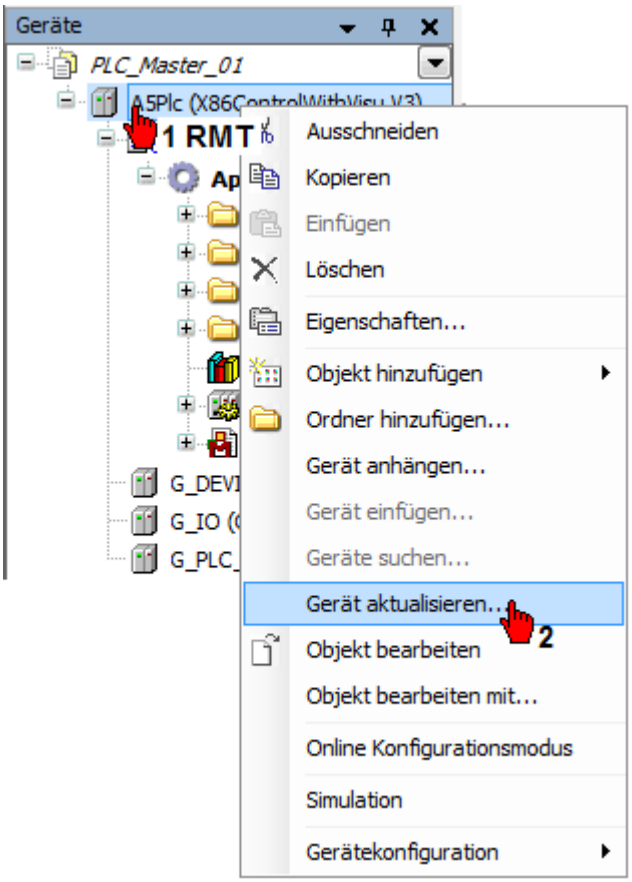
4.3.1.13.3.5 AMK Zielsystems Wechsel

Serviceinformation!

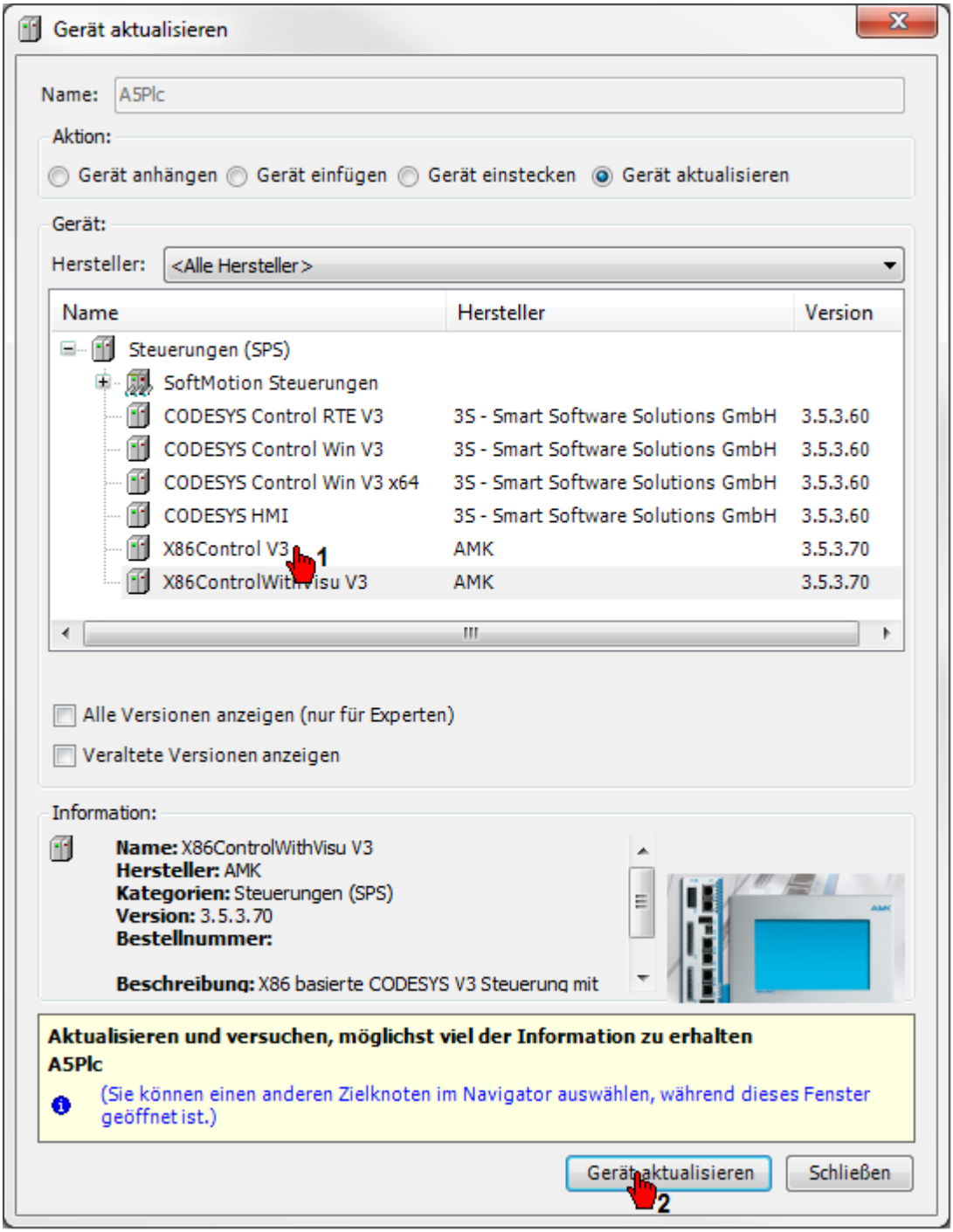
Nachfolgendes Beispiel zeigt, wie Sie ein bestehendes Projekt auf ein anderes Zielsystem umstellen.

Abhängig von der neuen Steuerungsvariante sind weitere Anpassungen durchzuführen. Achtung steuerungsspezifische Templates werden nicht aktualisiert.

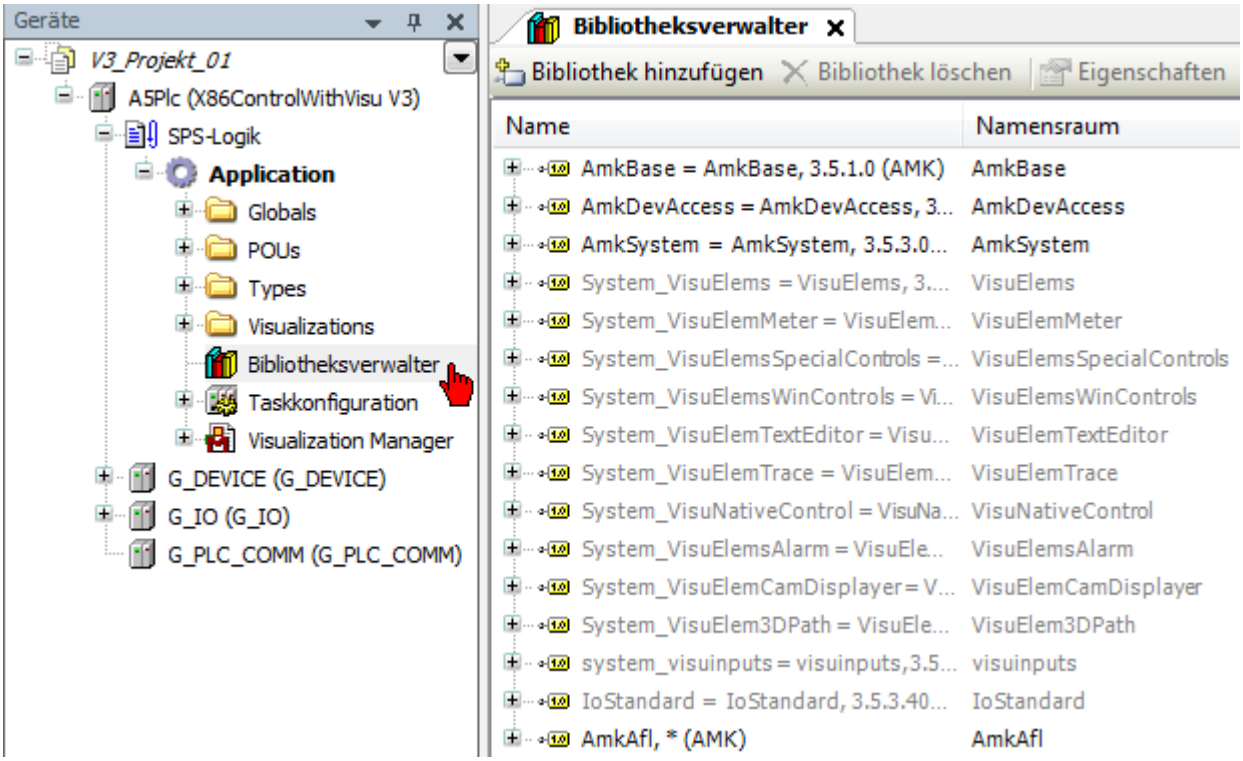
Eine nachträgliche Änderung des Zielsystems ist nicht angedacht. Erstellen Sie in diesem Fall ein neues AIPEX PRO Projekt für die neue Steuerungsvariante. Anschließend können Sie den bestehenden Programmcode in das neue CODESYS Projekt übernehmen.



Wählen Sie in der Auswahlliste den neuen Steuerungstyp an. Bestätigen Sie mit '**Gerät aktualisieren**'.



4.3.1.13.4 Bibliotheksverwalter



Neben dem vollen CODESYS V3 Programmiersystemumfang stehen dem Anwender verschiedene, auf die Antriebsfunktionalität abgestimmte, AMK-spezifische Bibliotheken zur Verfügung (siehe Tabelle Bibliotheksübersicht der AMK Grundbausteine). Der Funktionsumfang dieser Bibliotheken entspricht im Wesentlichen dem, der bereits in CODESYS V2.3 enthaltenen Bibliotheken zur Unterstützung der automatischen Buskonfiguration. Somit können bestehende CODESYS V2 Projekte mit vertretbarem Aufwand in CODESYS V3 Projekte konvertiert werden.

In der aktuellen Version von AIPEX PRO werden die CAN-basierte Buskonfiguration (ACC Amk Can Communication) sowie die EtherCAT-Buskonfiguration im Rahmen der automatischen Buskonfiguration unterstützt.

Basierend auf der jeweiligen Steuerung können mit AIPEX PRO zielsystemspezifische Musterprojekte (Templates) erzeugt werden, die Ausgangspunkt eines neuen CODESYS V3 Projekts und Basis der automatischen Buskonfiguration sind.

AIPEX PRO V3 ermöglicht:

- Die Konfiguration einer Geräteanordnung, mit allen über ACC oder EtherCAT erreichbaren Komponenten (Steuerungen, Antrieben, IO-Modulen, ..)
- Die Programmierung der Steuerungen mittels CODESYS V3 unter Verwendung der AMK-Bibliotheken
- Den funktionalen Zugriff (mittels AMK-Funktionsbausteine) auf alle über die Busse erreichbaren Komponenten
- Die Kommunikation (synchron/asynchron) zwischen AMK-PLC-Baugruppen (mittels AMK-Funktionsbausteinen)
- Die hierauf basierende, automatische Generierung der für die Buskommunikation erforderlichen Information

Bibliotheksübersicht der AMK Grundbausteine

Topic	<Name>.library	Version	Impl.[1)]	Bib.[2)]	Platz [3)]	Anmerkung
Basic	AmkBase	3.5.1.0	E	C	G	Basisfunktionalität
	AmkFile	3.5.1.0	E	C	G	File Funktionen
	AmkSystem	3.5.3.0	I	C	B	Systemfunktionalität
Communication	AmkCom	3.5.1.0	E	C	G	Kommunikationsfunktionalität
	AmkSockets	3.5.1.0	E	C	G	Ethernet Socket Funktionen
	AmkTcp	3.5.3.0	I	C	B	TCP Kommunikationsschnittstelle
	AmkUdp	3.5.3.0	I	C	B	UDP Kommunikationsschnittstelle

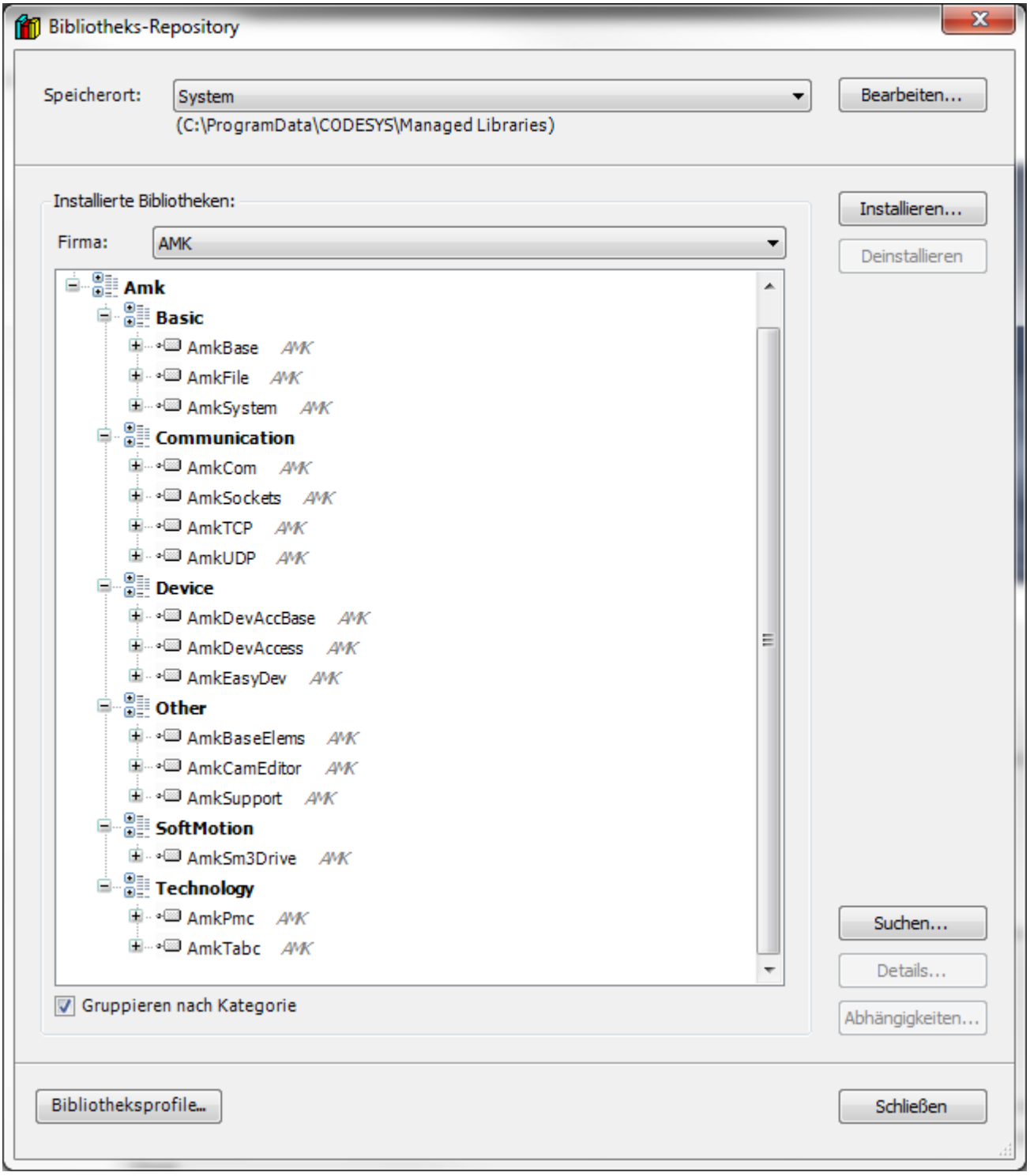
Topic	<Name>.library	Version	Impl.[1)]	Bib.[2)]	Platz [3)]	Anmerkung
Device	AmkDevAccBase	3.5.3.0	I	S	B	Basis Gerätezugriffsfunktionalität
	AmkDevAccess	3.5.3.0	I	S	B	Gerätezugriffsfunktionalität
	AmkEasyDev	3.5.3.0	I	S	B	Vereinfachtes AMK Geräteinterface
Other	AmkBaseElems	3.5.3.0	I	C	B	Basis Visualisierungselemente
	AmkCamEditor	3.5.3.0	I	S	B	CamEditor spezifische Type Definitionen
	AmkSupport	3.5.3.0	I	C	B	Unterstützung spezieller Hardware / Technologien
SoftMotion	AmkSm3Drive	3.5.3.0	I	S	B	AMK Softmotion Antriebsinterface
Technology	AmkPmc	3.5.3.0	I	C	B	Druckmarkenregelungsfunktionalität
	AmkTabc	3.5.3.0	I	C	B	Tabellenberechnungsbausteine

1) Implementierung: E = extern / I = intern
   - Extern: Realisiert in Form einer Systemsoftwarekomponente, programmiert in 'C'..
   - Intern: Realisiert in Form eines IEC-Programms.
2) Bibliotheksausführung: C = als 'compiled library'/ S = als 'source library'

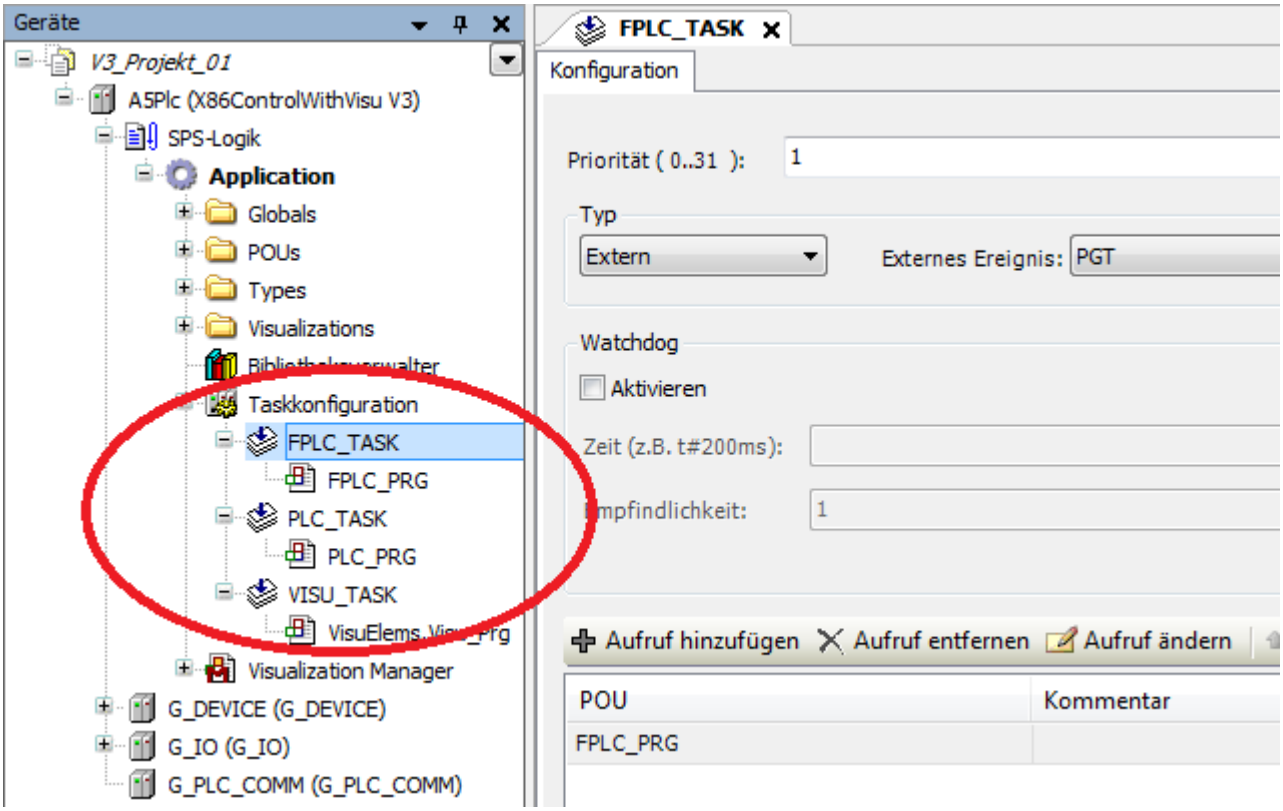
   Einer Bibliotheksausführung in Form einer 'kompilierten Bibliothek' (compiled library) bzw. einer Quelltext Bibliotheken (source library).
   - Kompilierte Bibliotheken sind Code effizienter, jedoch nicht im Quelltext analysierbar. Deshalb kann auch nicht für Testzwecke in die Bibliotheken 'gestept' werden.
   - Quelltext Bibliotheken können im Testbetrieb wie das Programm analysiert werden (Haltepunkte, Einzelschrittbetrieb,..).
3) Platzhalterrealisierung: G = mittels Gerätebeschreibung, B = mittels Bibliotheksprofil

   Einer Platzhalterrealisierung basierend auf einer Gerätebeschreibung bzw. mittels eines Bibliotheksprofils.
   - Bei der 'Platzhalterrealisierung basierend auf einer Gerätebeschreibung' wird die Versionsauflösung der Bibliothek im Rahmen der Gerätebeschreibung der Steuerung (Steuerungsversion) vorgenommen.
   - Bei der 'Platzhalterrealisierung basierend auf einem Bibliotheksprofil' wird die Versionsauflösung der Bibliothek in Abhängigkeit von der Compilerversion in CODESYS vorgenommen.

Die Auswahl der AMK-Bibliotheken erfolgt in CODESYS V3 mit dem Bibliotheksverwalters. Bei Reduktion der Ansicht auf Bibliotheken der Fa. AMK ergibt sich folgenden Übersicht:



4.3.1.13.5 Default Taskkonfiguration



POU	Typ	Priorität
FPLC_PRG	Extern, PGT Ereignisgesteuert PGT Zykluszeit = ID2 'SERCOS-Zykluszeit'	1
PLC_PRG	Freilaufend	5
VISU_TASK	Zyklisch, Intervall 100 ms	10

4.3.1.13.6 Diagnoseinformationen

Diagnoseinformationen werden mit den Variablen boErr (Fehlersignal) und iErrID (Fehleridentnummer) angezeigt.

```
fbSET_PLCVAR_SYNC_DINT_B(
    boEnable:= TRUE,
    diInVal:= 250,
    boEnabAck=> ,
    boErr=> ,
    iErrID=> ,
    stPlcVar:= g_stPLCVAR_B);
```

Hierbei gilt:

iErrID = 0 kein Fehler bzw. keine Warnung
iErrID <> 0 Warnung, falls boErr = FALSE
Fehler falls boErr = TRUE

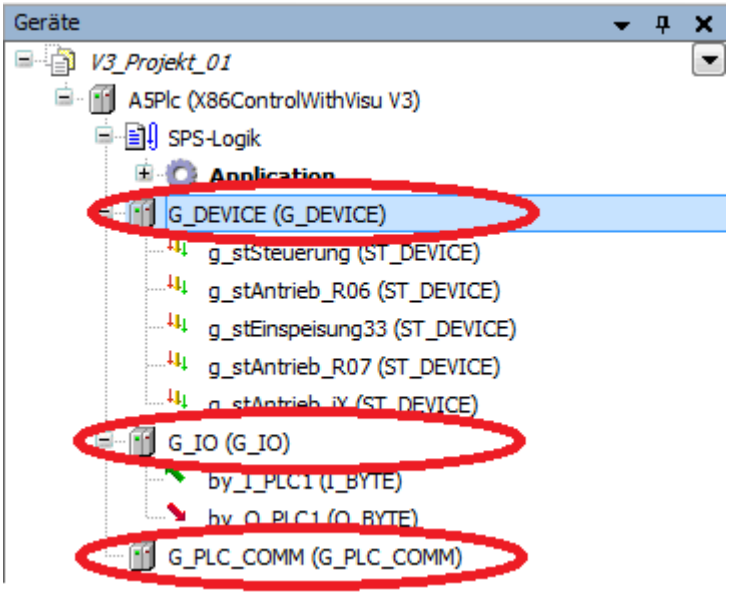
iErrID > 0 lokale Diagnoseinformation, gemäß der jeweiligen Bausteinspezifikation
iErrID < 0 globale Diagnoseinformation, gemäß der nachfolgenden

Globale Diagnoseinformation (bibliotheksübergreifend)

Fehleridentnummer (iErrID)	Bedeutung
- 8	enMode: Modus unzulässig
- 9	ID2: Zykluszeit unzulässig
- 10	Serielle Schnittstelle belegt
- 20	Funktion wird von diesem Zielsystem nicht unterstützt

4.3.1.13.7 Steuerungskonfiguration

Die 'Steuerungskonfiguration' bildet die Zielhardware im Programmiersystem ab.



Die Steuerungskonfiguration unterscheidet zwischen folgenden Konfigurationsmöglichkeiten:

G_DEVICE: globale Devices (zum Zugriff auf Geräteinformation von z.B. Antrieben, Einspeisemodulen oder Steuerungen). Die globalen Devices werden in der Informatik als 'Handle' bezeichnet.

G_IO: globale IO Variablen (zum Zugriff auf z. B. binäre Ein-/Ausgänge)

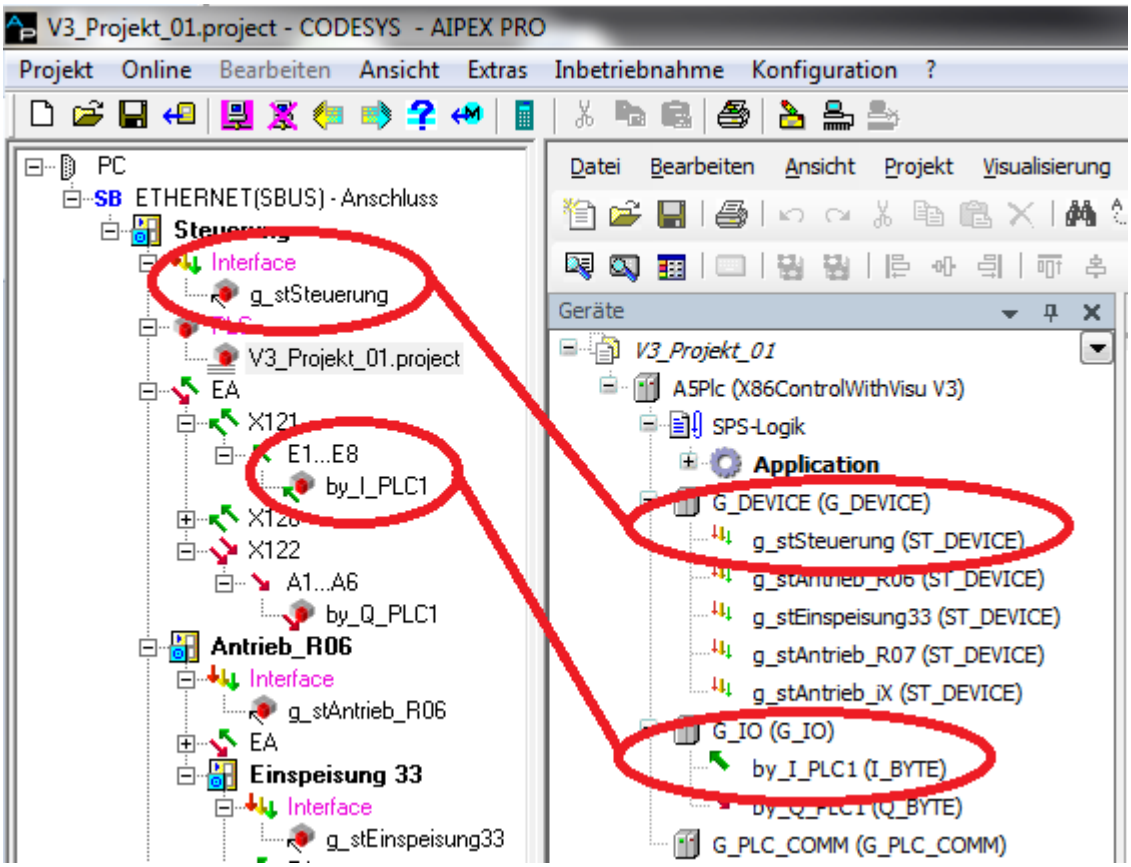
G_PLC_COMM:globale PLC zu PLC Kommunikationsvariable (zur Kommunikation zwischen PLCs). Die globalen PLC zu PLC Kommunikationsvariablen werden in der Informatik als 'Handle' bezeichnet.

Neue symbolische Gerätenamen erstellen:

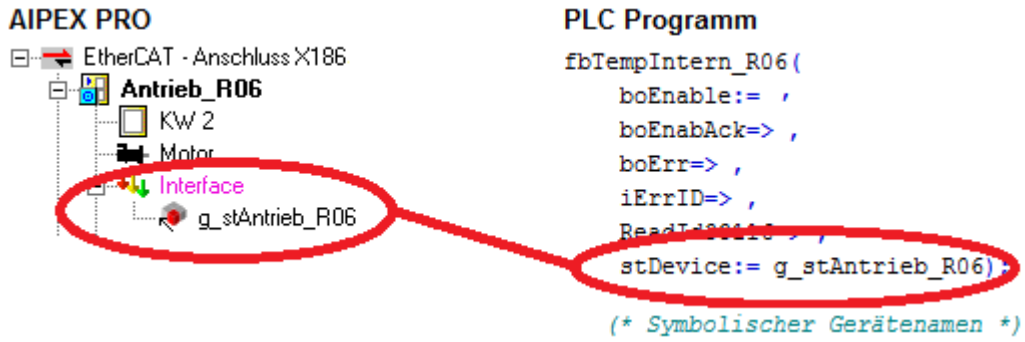
Klicken Sie mit der RMT auf **G_DEVICE**, **G_IO**, **G_PLC_COMM**. Wählen Sie anschließend den Menüpunkt 'Gerät anhängen' an.

Die in der Steuerungskonfiguration angelegten Geräte werden in der AMK Dokumentation als 'Symbolische Gerätenamen' bezeichnet.

Die Zuordnung der physikalischen Geräten zu den symbolischen Gerätenamen , erfolgt während des Buskonfigurationsvorganges.

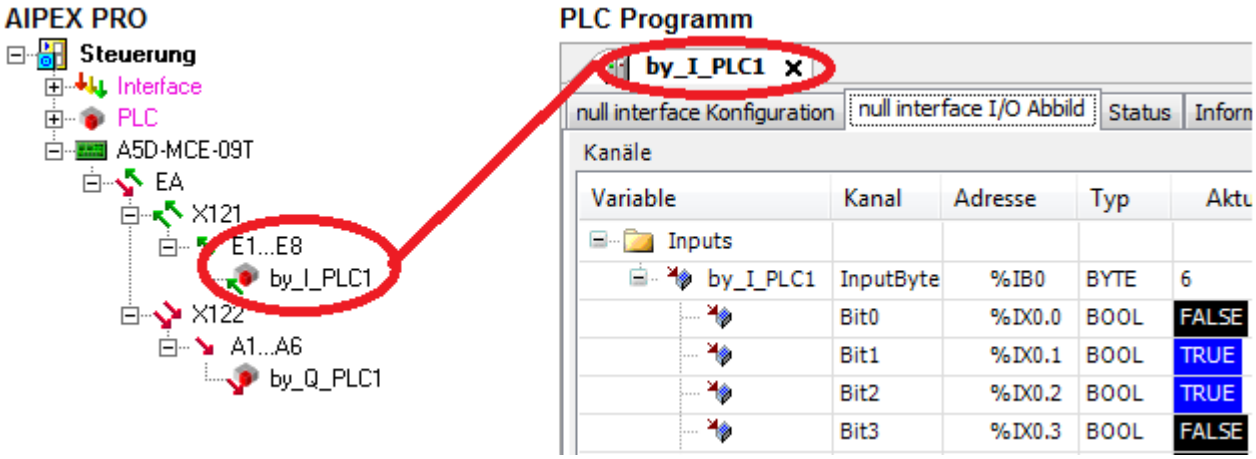


Die Zuordnung eines PLC Funktionsbausteines zu einem physikalischen vorhandenen Gerät findet über die Variable 'stDevice' statt. Verknüpfen Sie dazu die Variable 'stDevice' mit dem symbolischen Gerätenamen.



Example IOs:

Die Zuordnung der physikalischen Geräten zu den symbolischen Gerätename (IO Variablen), erfolgt während des Buskonfigurationsvorganges.



4.3.2 FIRST STEPS mit CODESYS V2 und AFL Standard Bausteine

'FIRST STEPS' beschreibt, wie Sie:

- AIPEX PRO 'Basis Einstellungen' vornehmen
- ein neues PLC Projekt mit AIPEX PRO erzeugen
- AFL Standard Bausteine für Steuerung, Einspeisung und Achsen importieren und instanzieren
- der Zugriff auf die Achsstukturen (Ist- und Sollwerte)
- und das PLC Programm übertragen

Das Kapitel 'Schnell zum Ziel' enthält weitere Beispiele mit den Themen:

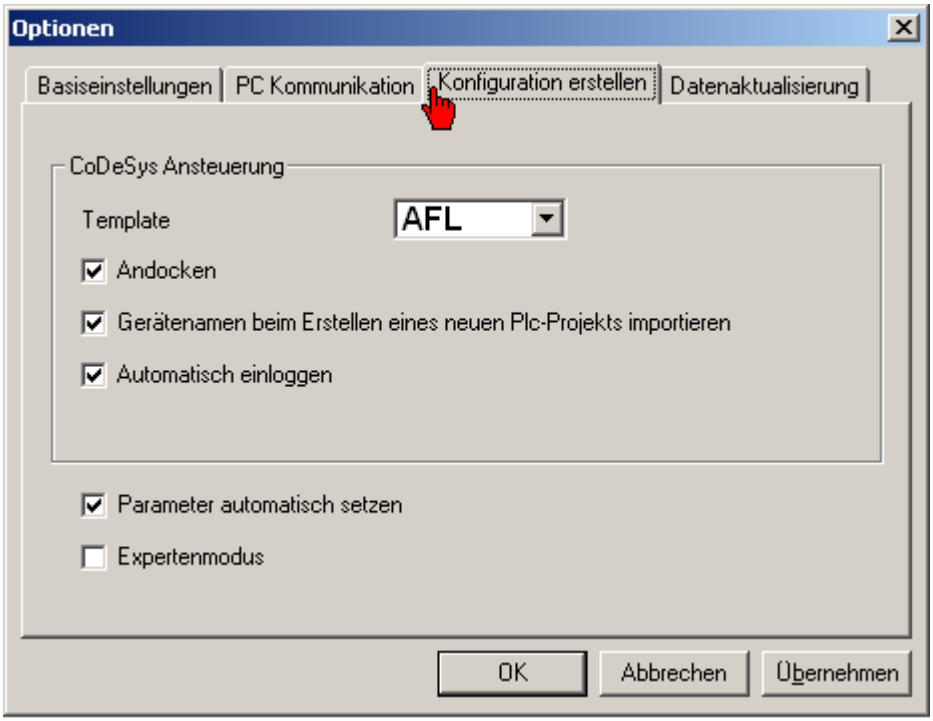
- Diagnose Array (arstDiagnosis) visualisieren
- EA Zugriff
- Erweiterte Variablenzugriffe
- Kurvenscheiben
- PLC Projekt importieren
- Statusbits erweitern
- Zugriff auf Steuerungs EA's

4.3.2.1 Basis Einstellungen

4.3.2.1.1 Konfiguration erstellen (allgemein)

Das Template muss auf 'AFL' stehen, bevor das CODESYS Projekt erzeugt wird.

'Extras' - 'Optionen'



Template Sprache

Über das Template wird das Starttemplate bei einem neu erstellten CoDeSys Projekt eingestellt.

Als Starttemplate steht 'ST' Strukturierter Text (textuelle Sprache), FUP Funktionsplan (grafische Sprache) oder die AFL Bibliotheken (sofern zusätzlich Installiert) zur Verfügung. Strukturierter Text ist die Standard Einstellung.

Andocken

Der PLC Editor CoDeSys wird in die AIPEX PRO Oberfläche direkt integriert. Ist diese Option nicht gesetzt, wird der PLC Editor CoDeSys als eigenständiges Softwareprogramm gestartet.

Automatisch einloggen

Bei einer fehlerfreier Erstellung einer Konfiguration wird direkt die CoDeSys Funktion ‚Einloggen' aufgerufen.

Gerätenamen beim Erstellen eines neuen Plc-Projekts importieren

Geräte die physikalisch vorhanden sind werden automatisch durch den Aufruf „Neues PLC Projekt erzeugen" in das PLC Projekt übernommen. Sie finden die symbolischen Gerätenamen unter **Ressourcen** -> **Steuerungskonfiguration** -> **PLC-Configuration**.

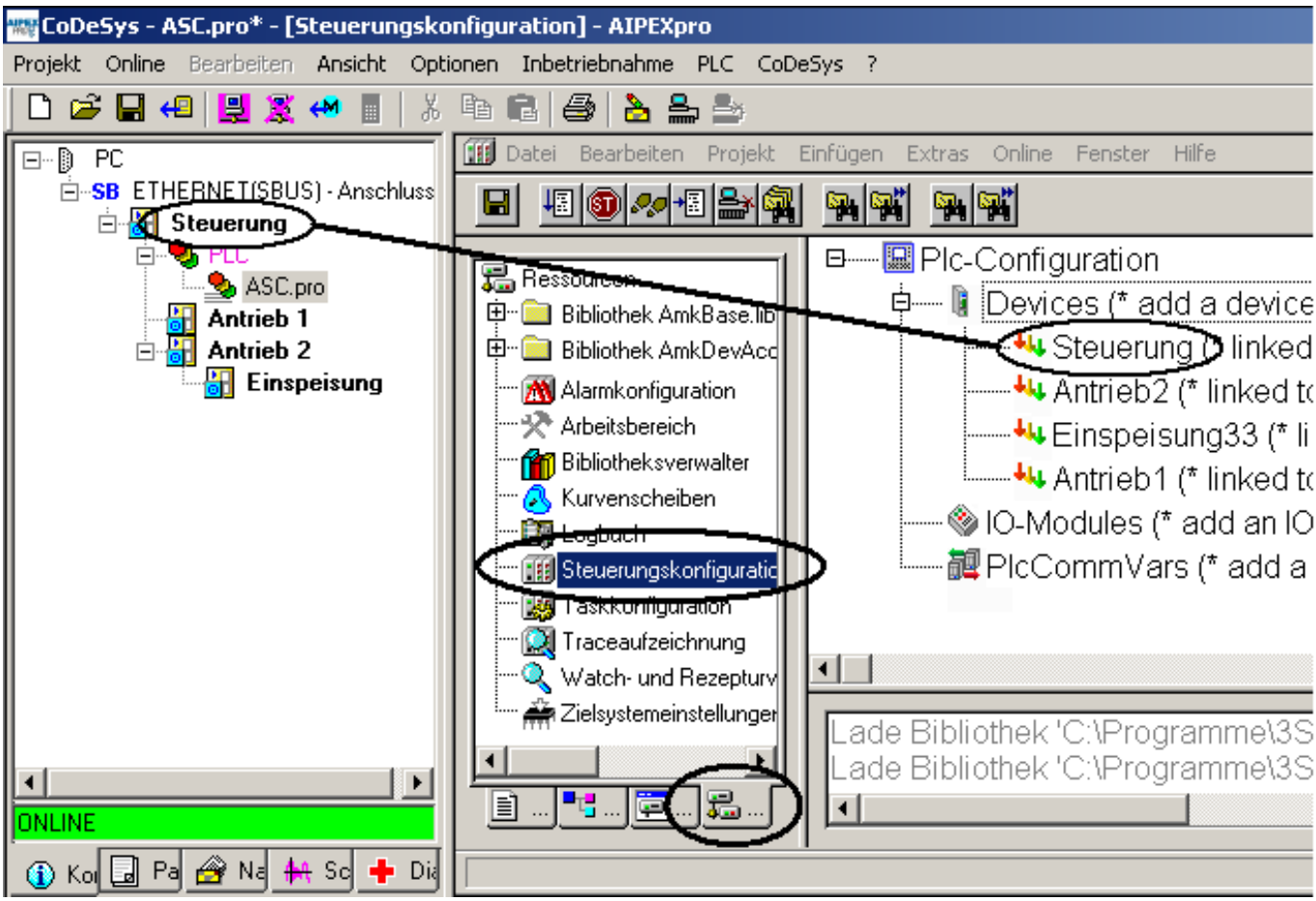
Ein nachträgliches Ändern der Gerätenamen ist nicht zulässig.

Beispiel CODESYS V2

Sie finden die symbolischen Gerätenamen in CODESYS unter **Ressourcen** -> **Steuerungskonfiguration** -> **PLC-Configuration**.

Im AIPEX PRO Gerätebaum wird der 'PLC-Handel Gerätenamen' automatisch mit dem dazugehörigen Geräteicon 'Interface' verknüpft.

Der automatische Nachrichtenkonfigurator nutzt diese Information um das Nachrichtenkonfigurationsfile zu erstellen.



Parameter automatisch setzen

Die für die CoDeSys Bibliotheksfunktionen benötigten Geräteparameter werden automatisch gesetzt. Diese **globale** Einstellung kann im Menü Extras -> Projekt Einstellungen -> Konfiguration erstellen --> Parameter automatisch setzen, individuell überschrieben werden.

Expertenmodus

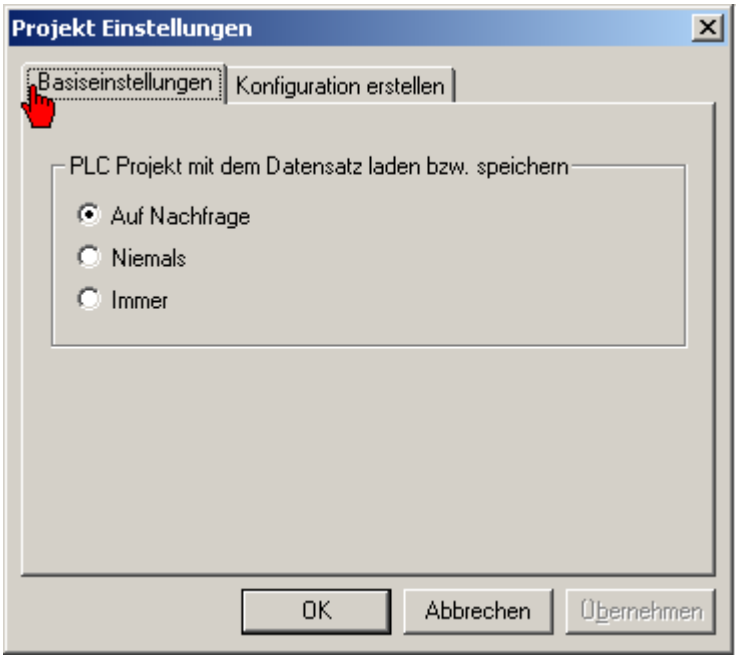
Es wird unter **CoDeSys** -> **Einloggen** keine neue „Buskonfiguration“ erstellt.

Ignoriert werden Änderungen im SPS Programm die einen neue Buskonfiguration zur Folge hätten.

Das **Einloggen** über CoDeSys **Online** ist freigeschalten.

4.3.2.1.2 Basiseinstellungen

'Extras' - 'Projekteinstellungen'

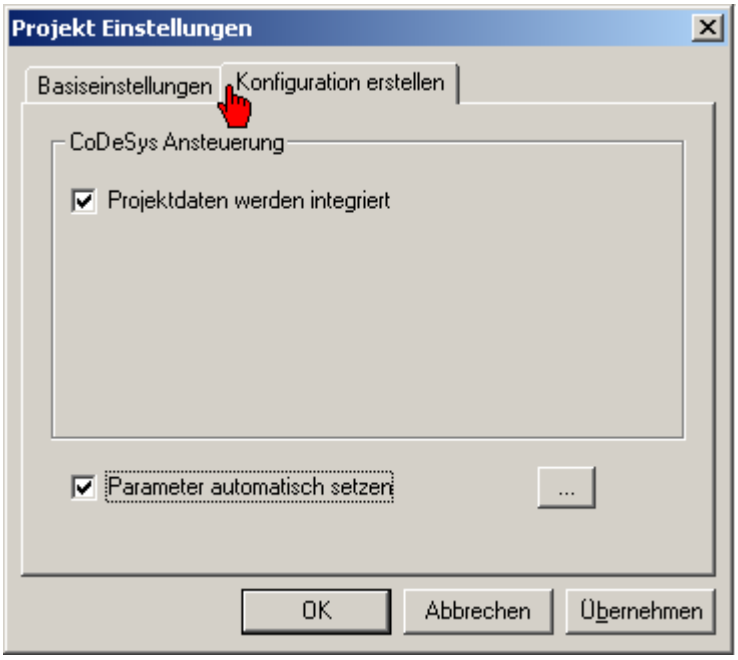


Ein vorhandenes PLC Projekt wird in der ID34159 gespeichert.

Die Basiseinstellungen beeinflussen die Menüpunkte
Online -> Parameter in das Projekt übernehmen
Online -> Parameter in das Gerät überspielen

4.3.2.1.3 Konfiguration erstellen (projektspezifisch)

'Extras' - 'Projekteinstellungen'



Projektdaten werden integriert
Das CoDeSys Projekte wird in die AIPEX Projektdatei integriert.
Ist diese Option nicht gesetzt, liegen die CoDeSys Projekte in einem separaten Verzeichnis neben dem AIPEX Projekt.

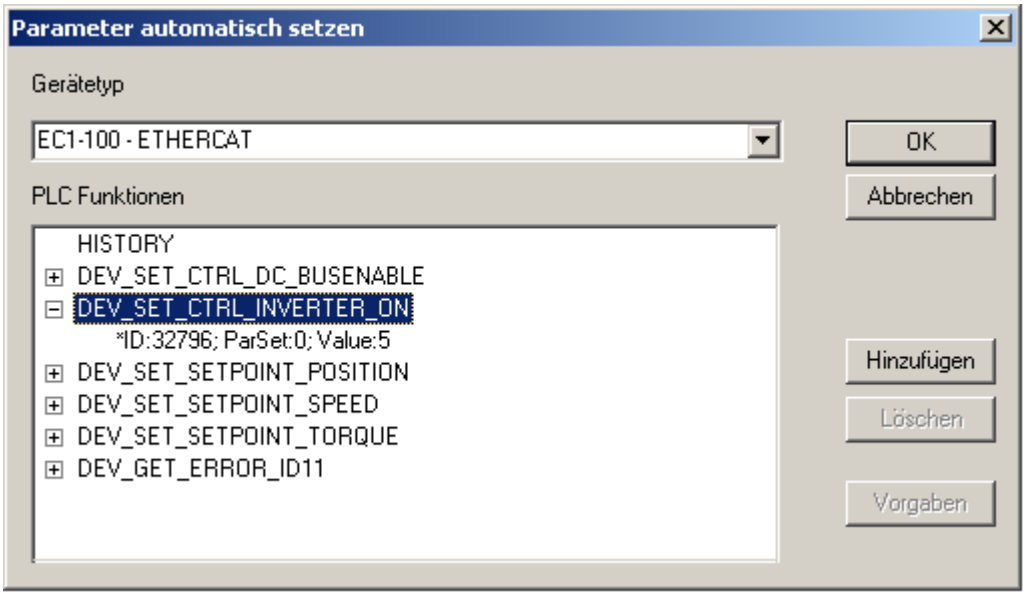
Parameter automatisch setzen

Die für die CoDeSys Bibliotheksfunktionen benötigten Geräteparameter werden automatisch gesetzt.

Beispiel:

Wird in CoDeSys der AMK Funktionsbaustein 'SET_CTRL_INVERTER_ON_x_RF' benutzt, setzt AIPEX PRO den Wert der ID32796 Quelle Reglerfreigabe auf 5 (Bedeutung: Signal Reglerfreigabe wird über die PLC gesetzt).

In der Parameterbeschreibung werden die ID's und deren Bedeutung ausführlich beschrieben.



Gibt es keinen Datensatz für einen konkreten Gerätetyp, wird der allgemeine („sonst.“) Datensatz verwendet.

Mit der Taste **Hinzufügen** können Sie den PLC Funktionen eigene Geräteparametrierungen zuweisen.
Mit der Taste **Löschen** können Sie eigene Geräteparametrierungen löschen.
Mit der Taste **Vorgabe** können Sie den ursprünglichen AMK Zustand wiederherstellen.

4.3.2.2 PLC Projekt erzeugen

Das Kapitel beschreibt, wie Sie ein neues PLC Projekt erstellen.

Die Screenshots sind in Englisch dargestellt. Unter **'Extras' 'Optionen' 'Basiseinstellungen '** kann die Programmsprache auf Englisch umgestellt werden.

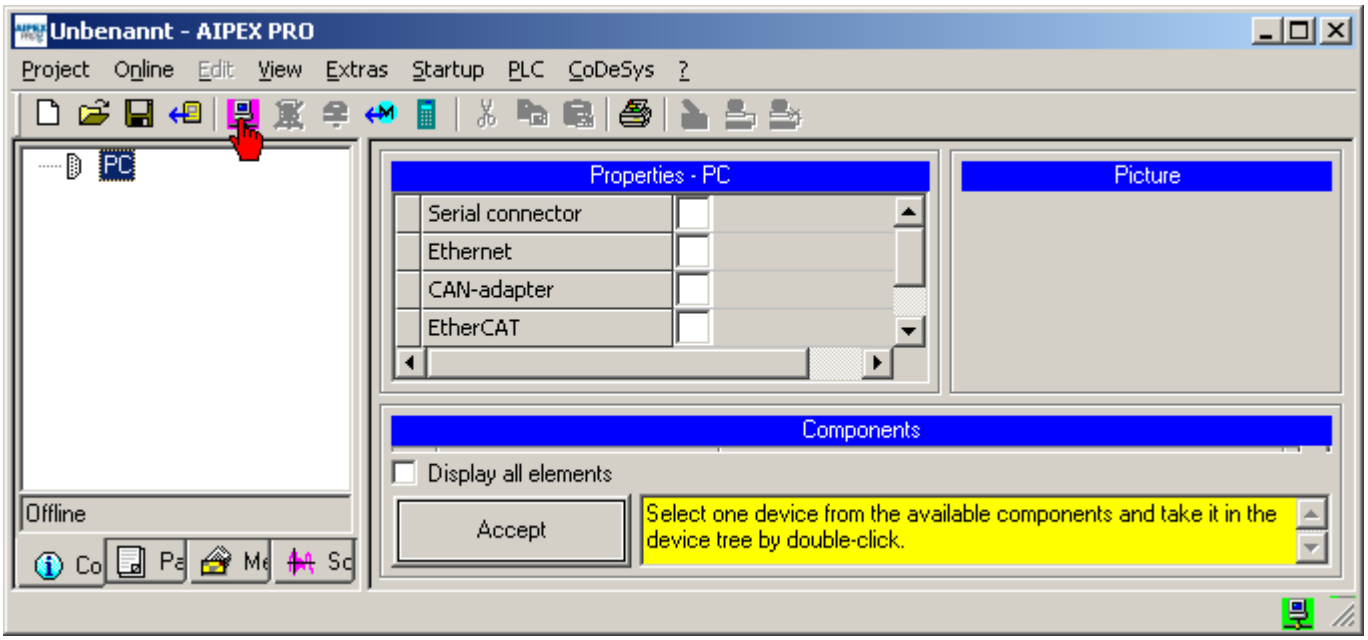
Voraussetzung:

Icon: Status 'Grün' (Schnittstelle breit)

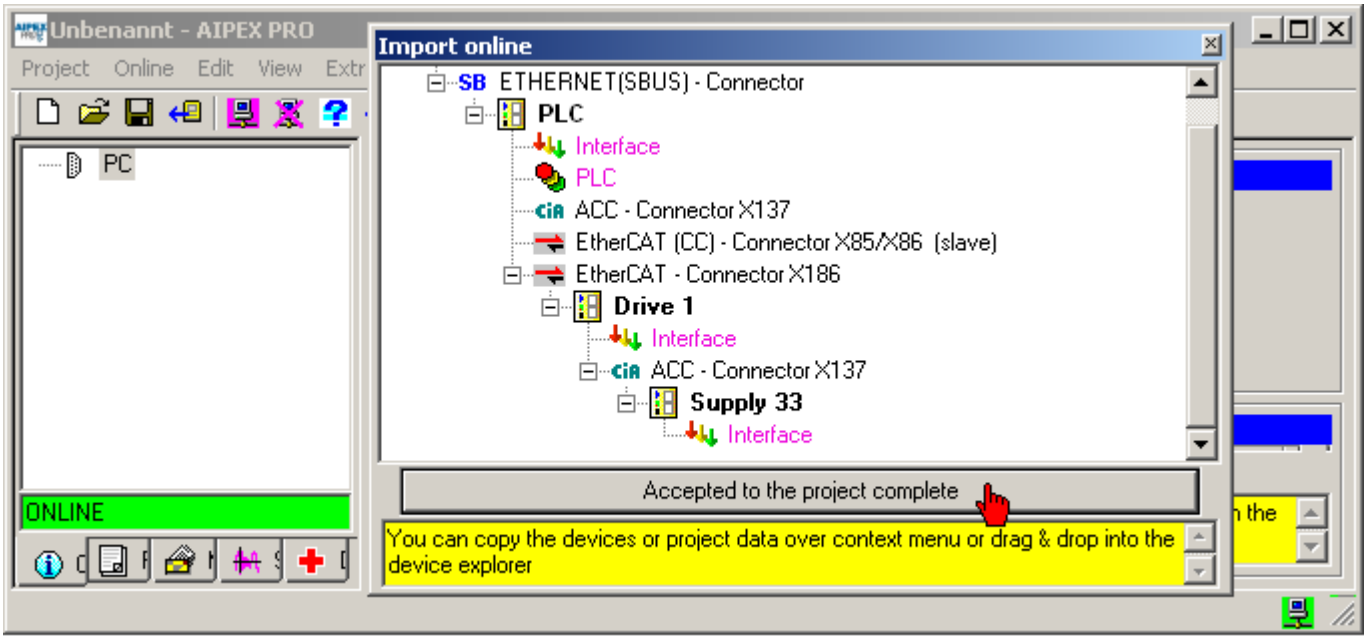
Icon: Status 'Rot' bzw. kein Icon vorhanden, prüfen Sie folgende Punkte:

- Erstinbetriebnahme Feldbus abgeschlossen: Siehe Dokument Erstinbetriebnahme KE/KW (AMK Teile-Nr. 204539) Erstinbetriebnahme Feldbus bzw. Dokument Erstinbetriebnahme iC/iX/iDT5 (AMK Teile-Nr. 204737) Erstinbetriebnahme Feldbus
- Direktverbindung über Ethernet aktiv: Siehe Dokument Gerätebeschreibung Steuerungen A-Serie (AMK Teile-Nr. 202975) Kapitel Direktverbindung über Ethernet.

Drücken Sie den Taster **'Logon'** um eine Online-Verbindung mit den verbundenen Modulen herzustellen.

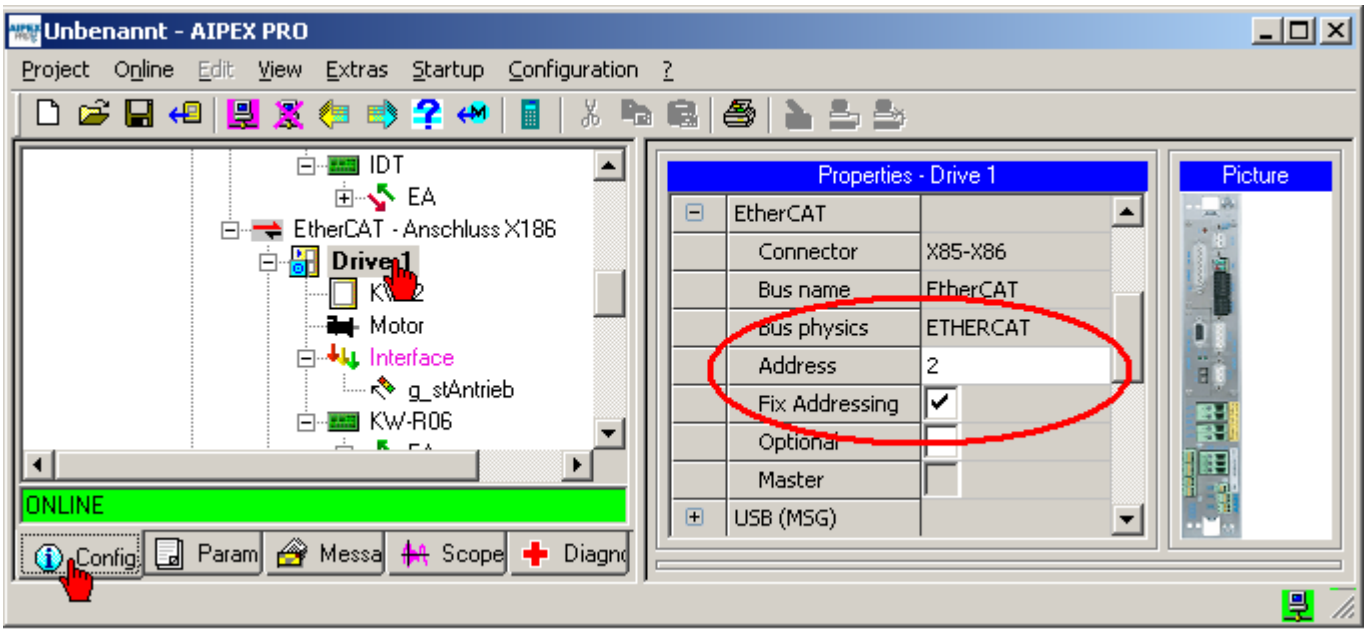


Prüfen Sie im Fenster 'Import Online' ob alle physikalisch vorhandenen Geräte erfasst sind.
Drücken Sie den Taster **'Accepted to the project complete'** um die Gerätedaten einzulesen.

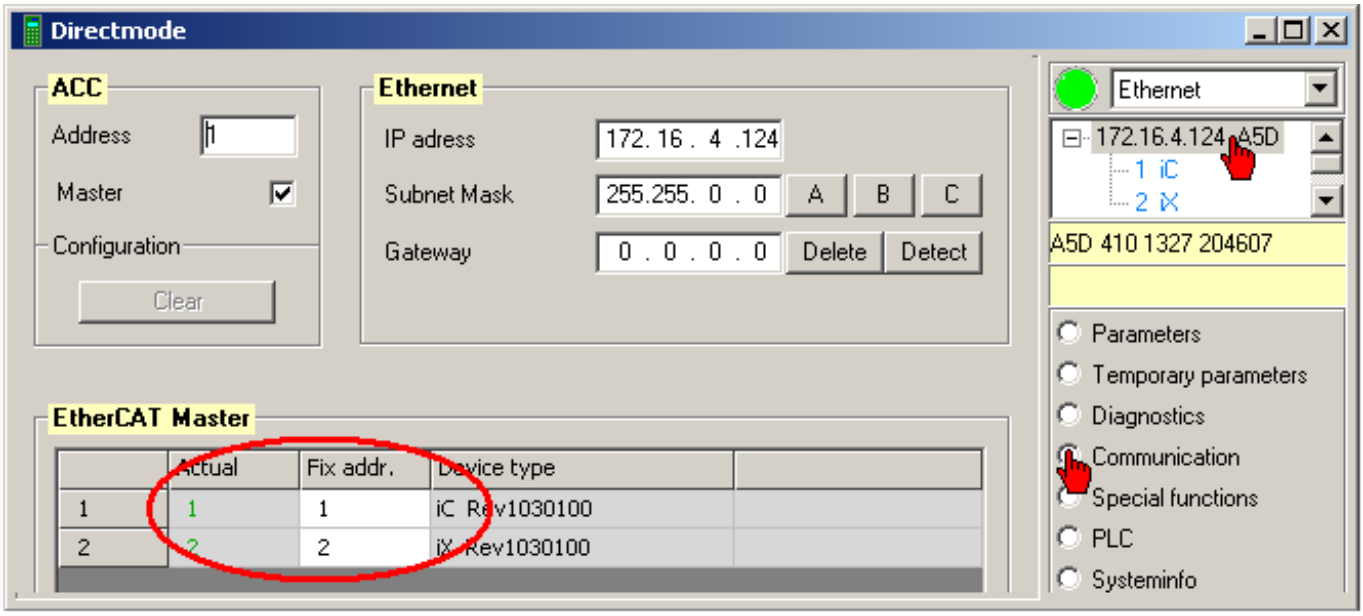


Prüfen Sie bitte die Verkabelung und Systemzustände falls Geräte nicht angezeigt werden.

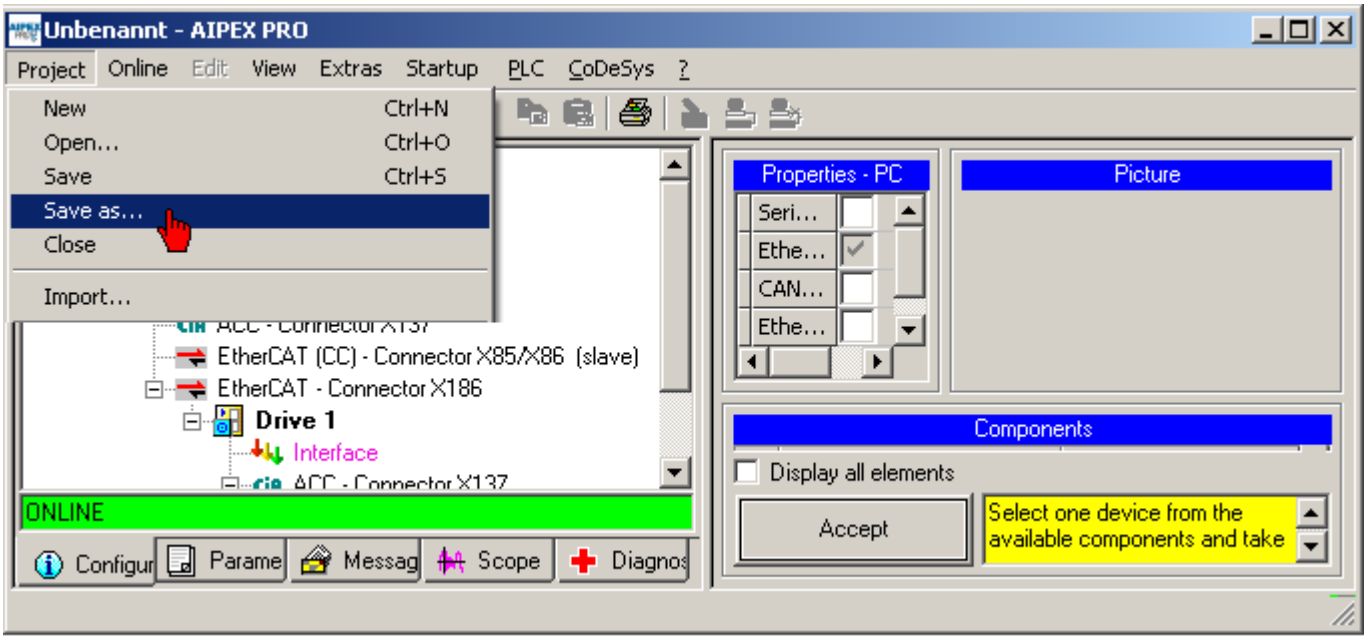
Vergeben Sie im Menü **'Configuration'** jedem EtherCAT Teilnehmer eine Fest-Adresse. Anschließend müssen Sie das System 'Neu Starten'.



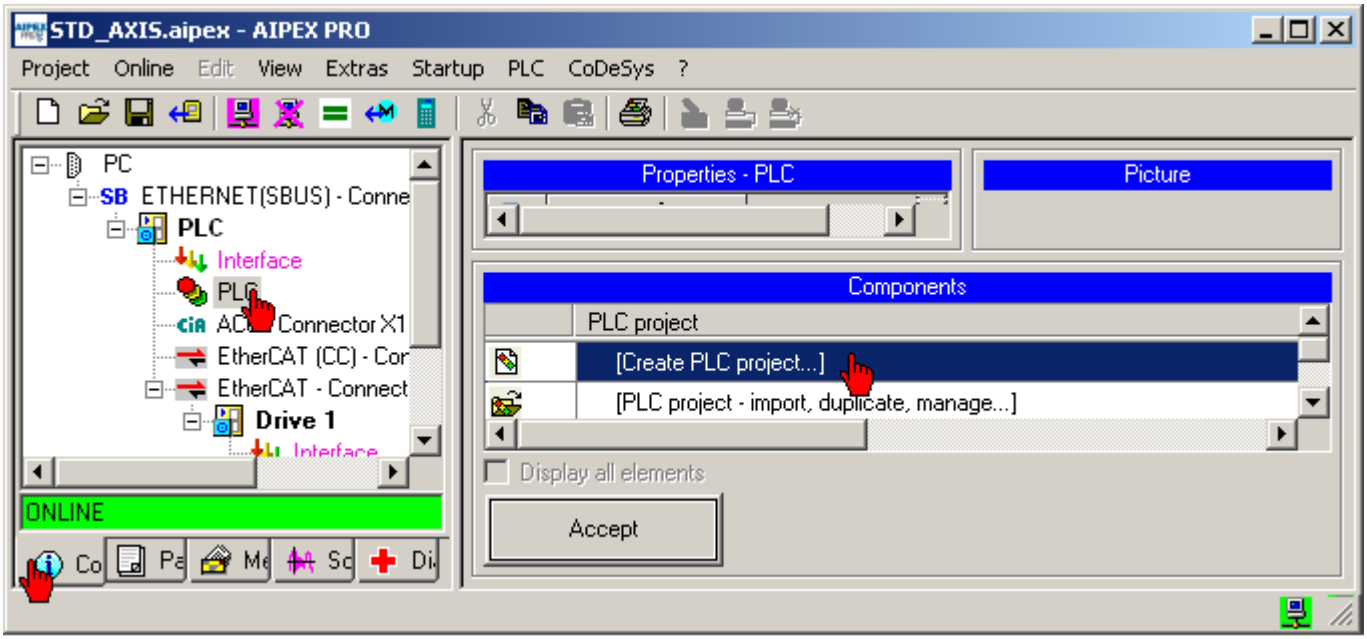
Kontrollieren Sie nach dem Neustart die aktuelle Position (Spalte: Actual) und die dazugehörige Festadresse (Spalte: Fix addr.). Öffnen sie dazu das AIPEX PRO Menü **'Directmode'** – **'Communication'**.



Speichern Sie das eingelesene Projekt unter **'Project'** **'Save as...'** ab. Dem PLC Projekt können Sie einen separaten Namen vergeben.

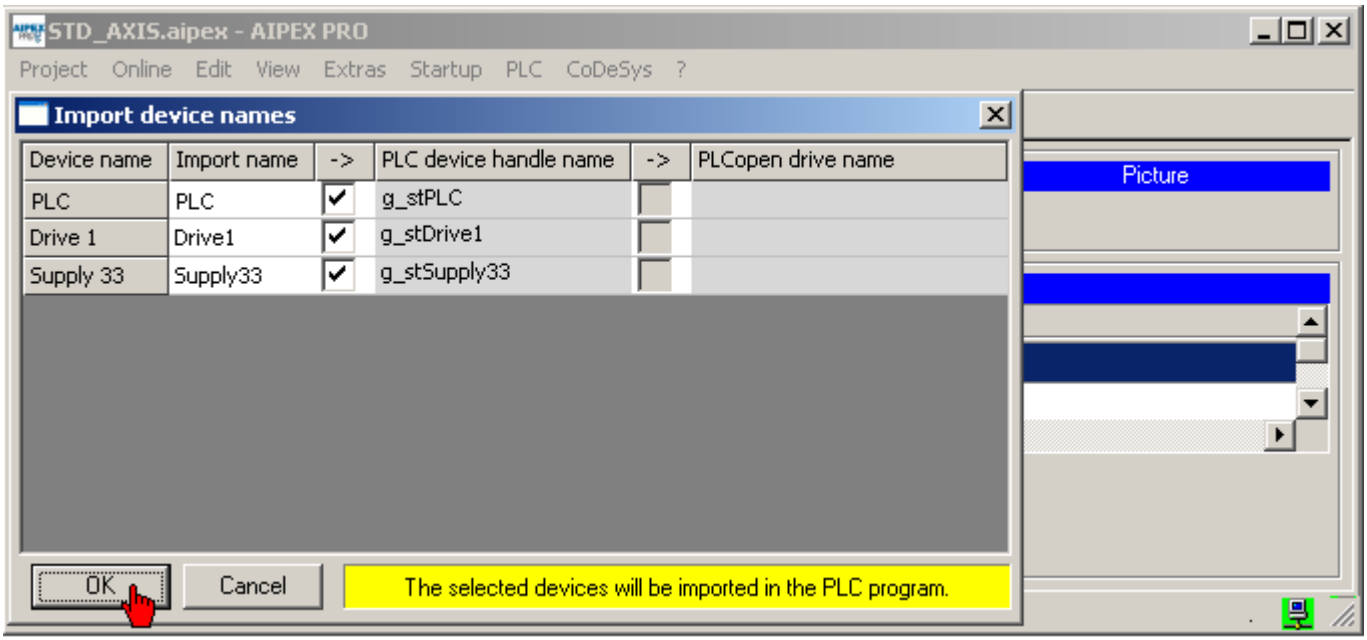


Wählen Sie die **'PLC'** im Gerätebaum an. Anschließend können Sie über **'Create PLC project..'** CODESYS starten.



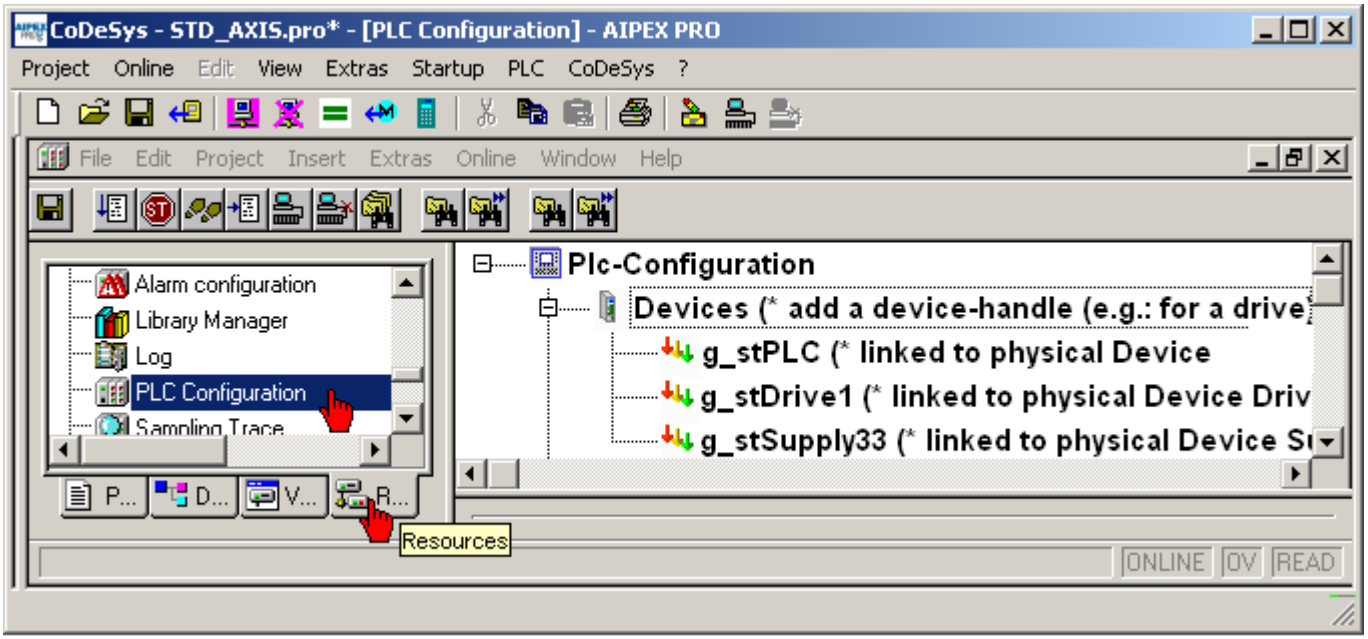
PLC fehlt:
Klicken Sie mit der RMT in den Gerätebaum. Unter **'Tree display set'** können Sie die Anzeige anpassen.

Mit dem Fenster **'Import device names'** werden die physikalisch vorhandenen Gerätenamen automatisch als symbolische Gerätenamen (Variablen) in das PLC Projekt importiert.



Das Fenster 'Import device names' kann manuell im Reiter **'Configuration (PLC marked)' 'Properties - PLC'** geöffnet werden.
Unter **'Extras' 'Options' 'Configuration create '** kann die Funktion an- und abgewählt werden.

Sie finden die symbolischen Gerätenamen (Variablen) unter **'Ressourcen' 'PLC Configuration'** wieder.

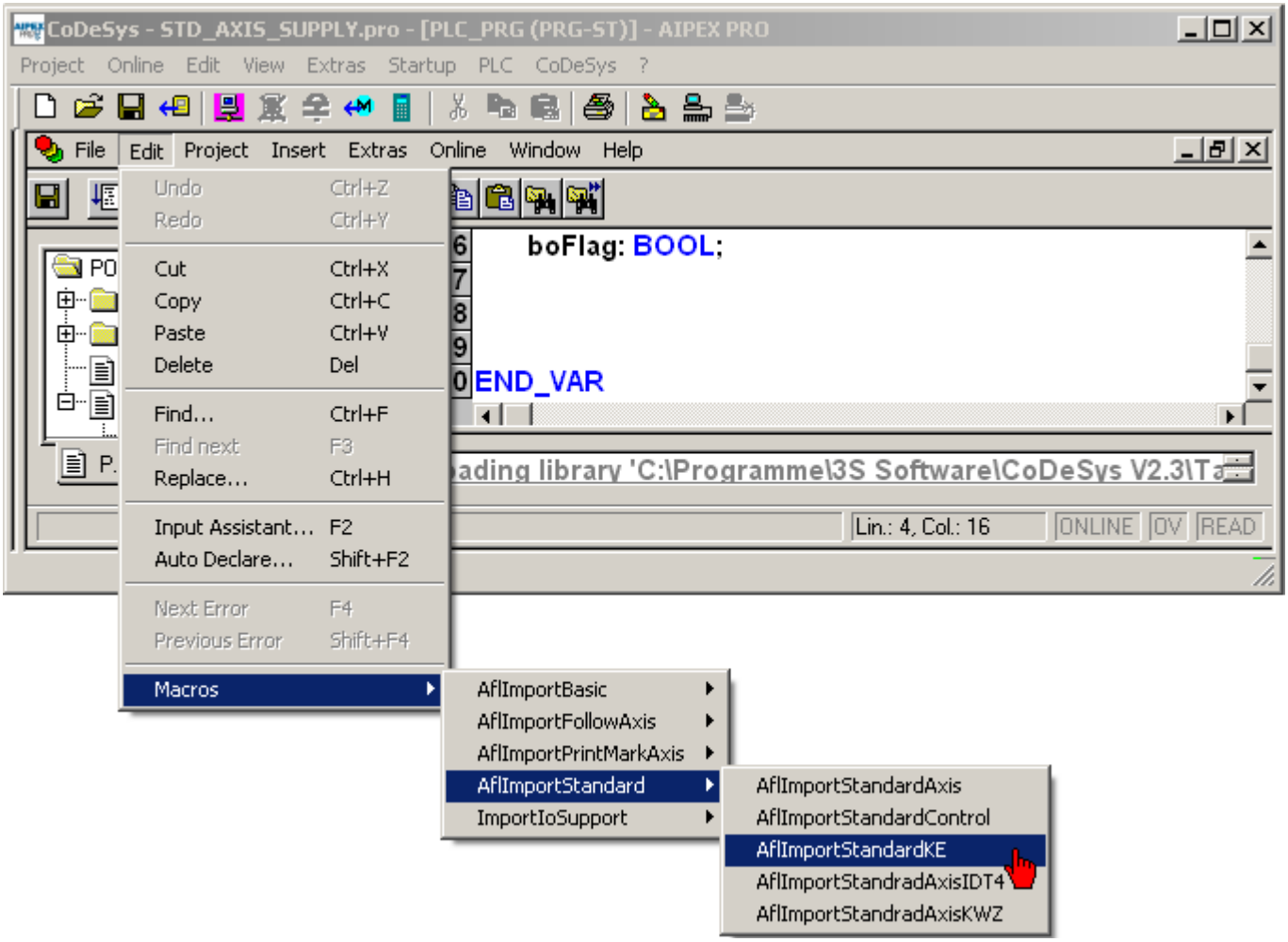


4.3.2.3 Standard Einspeisung importieren

Beschreibung der Standard Einspeisung: Siehe Funktionsbaustein 'STANDARD_KE' ab Seite 1

Wird nur bei Einsatz der Einspeisemodule KE/KEN/KES benötigt. (Ausnahme: KEN 05-xx).
Die Geräte AMKASMART iC und AMKASYN KEN 05-xx laden den, zur Versorgung der Wechselrichter benötigten, Gleichspannungszwischenkreis automatisch.

Importieren Sie die Funktion **'AflImportStandardKE'** über das Menü **'Edit' 'Macros' 'AflImportStandard'**.

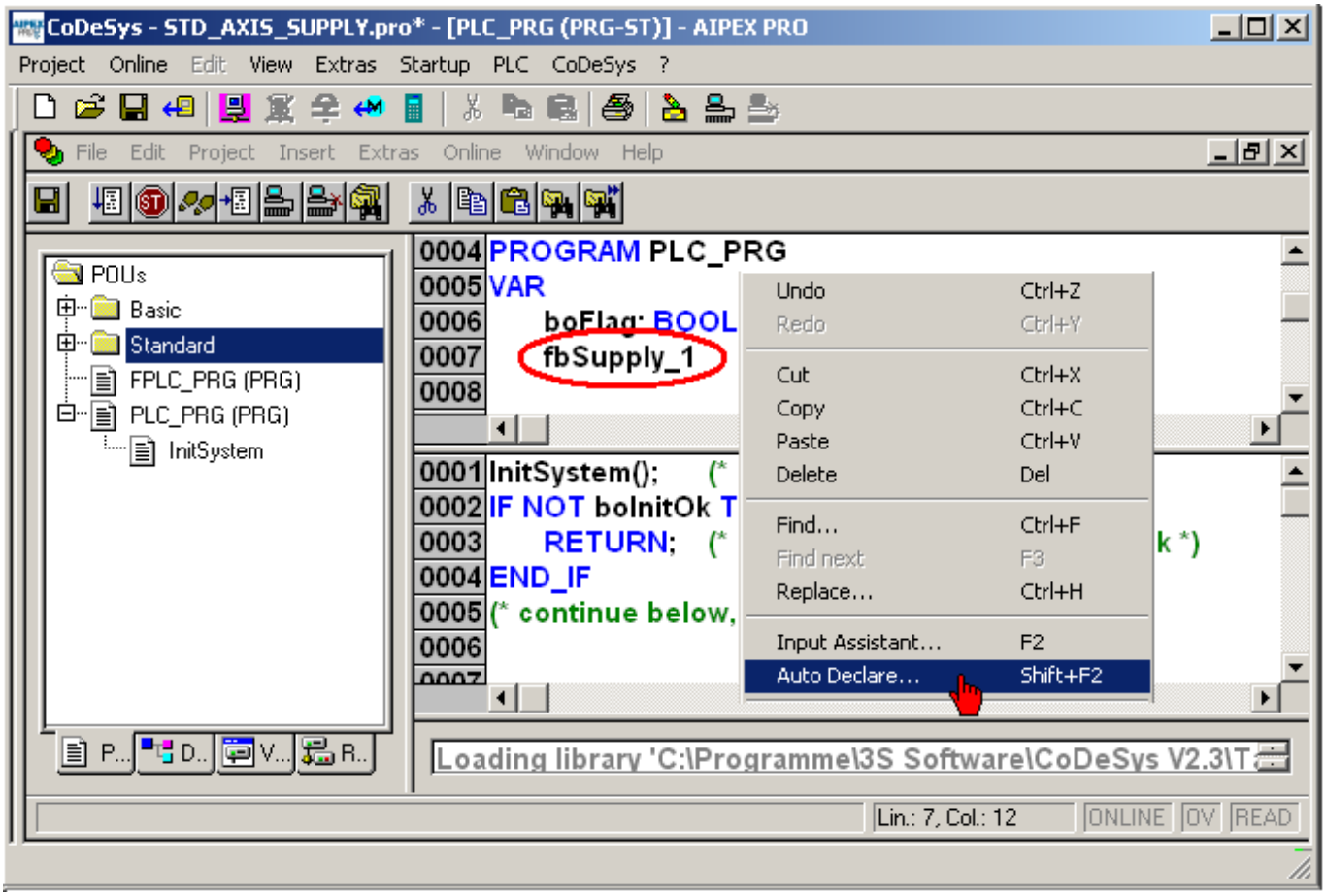


4.3.2.4 Standard Einspeisung instanzieren

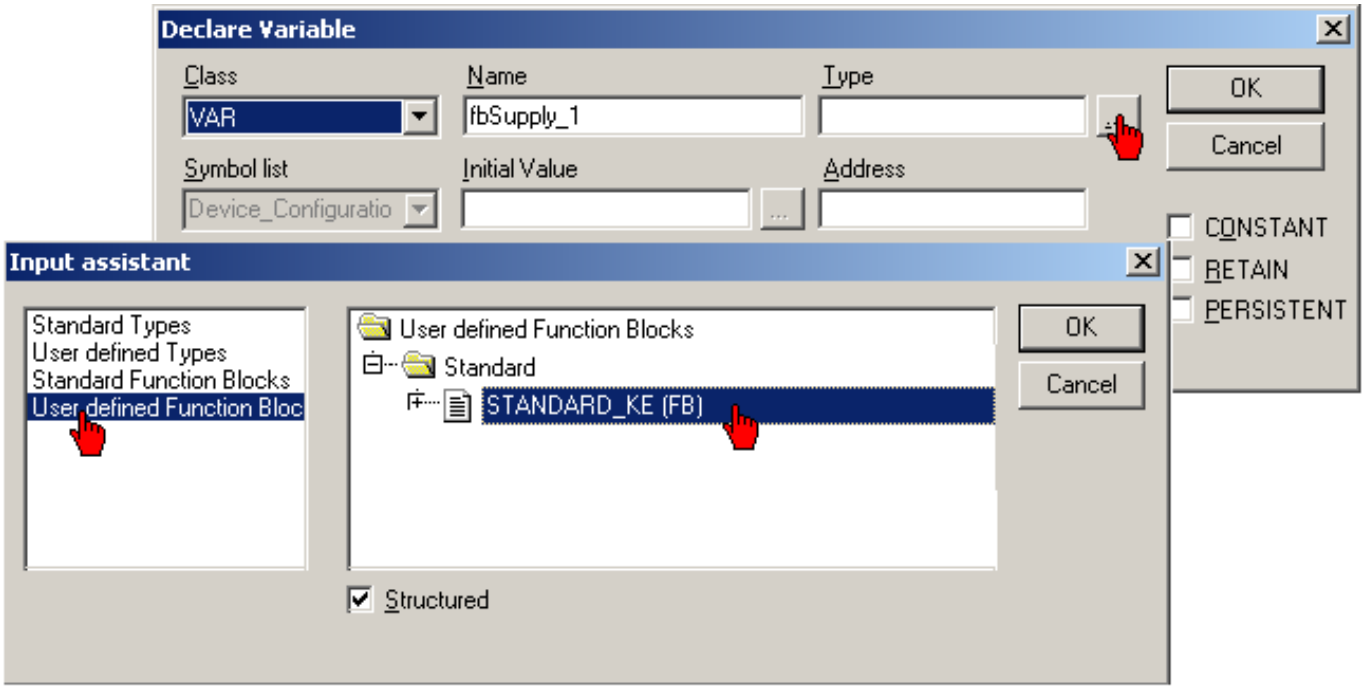
Wird nur bei Einsatz der Einspeisemodule KE/KEN/KES benötigt. (Ausnahme: KEN 05-xx).
Die Geräte AMKASMART iC und AMKASYN KEN 05-xx laden den, zur Versorgung der Wechselrichter benötigten, Gleichspannungszwischenkreis automatisch.

Legen Sie im 'PLC_PRG' Variablendeklarationseditor für jede physikalisch vorhandene Einspeisung (KE) eine Instanz vom Typ 'STANDARD_KE' an (im Beispiel fbSupply_1).

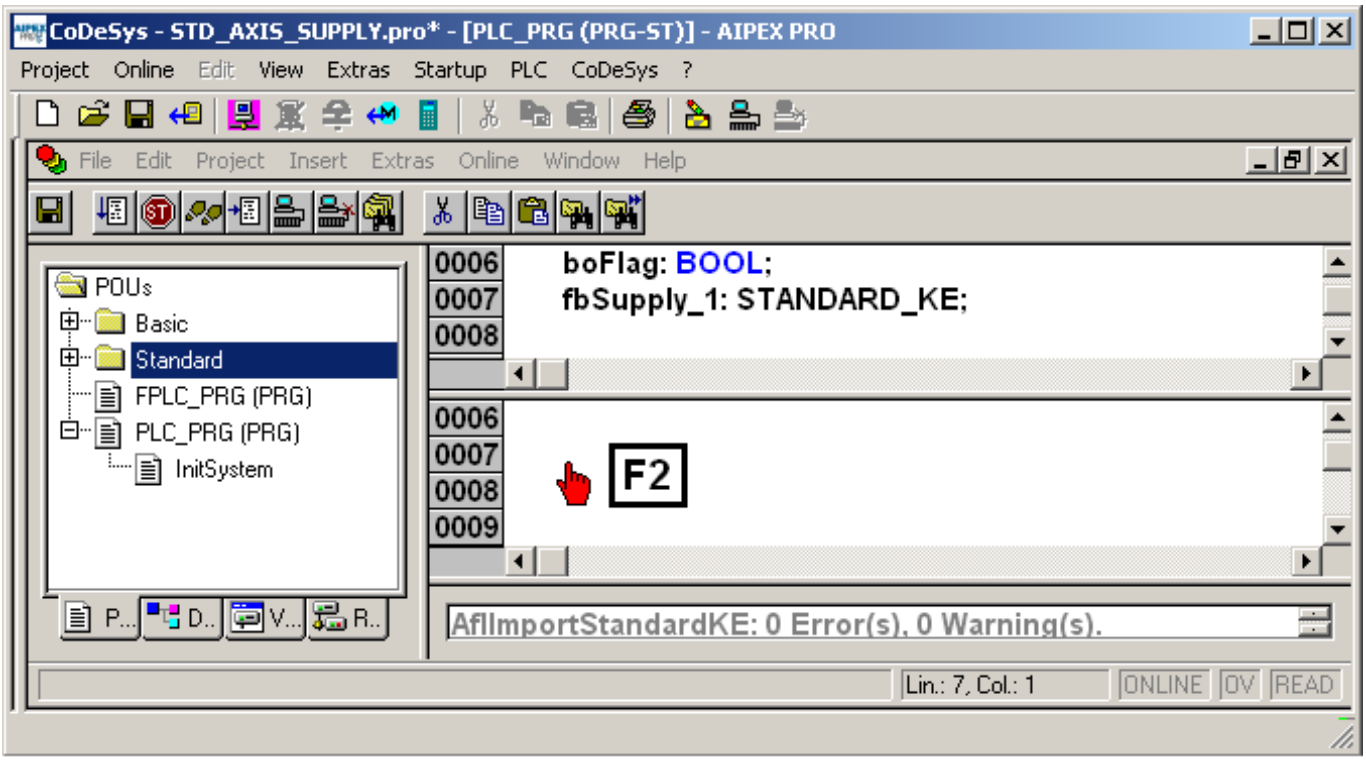
Klicken Sie mit der RMT auf die Variable. Anschließend können Sie mit **'Auto Declare'** das 'Variablen Deklarationsfenster (Declare Variable)' öffnen (bzw. Umschalt + F2).



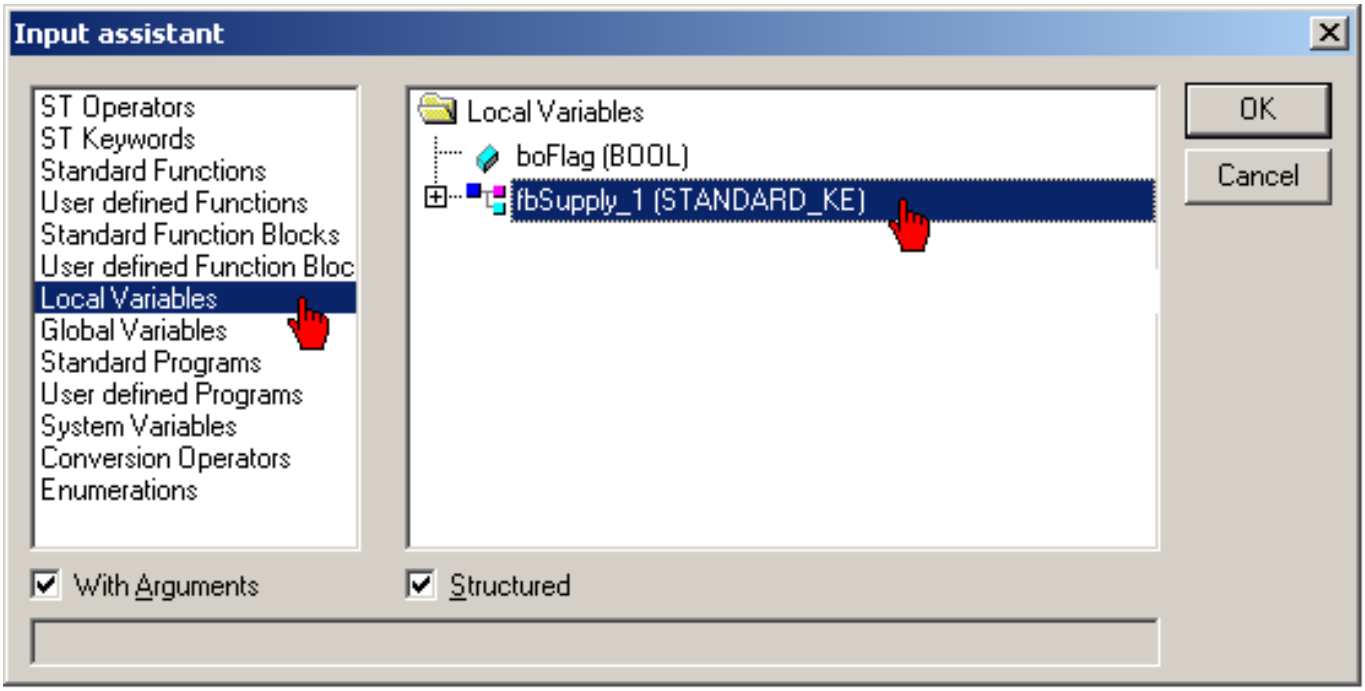
Sie finden den Funktionsblock 'STANDARD_KE' in der Eingabehilfe unter den **'User defined Function blocks' 'Standard'.**



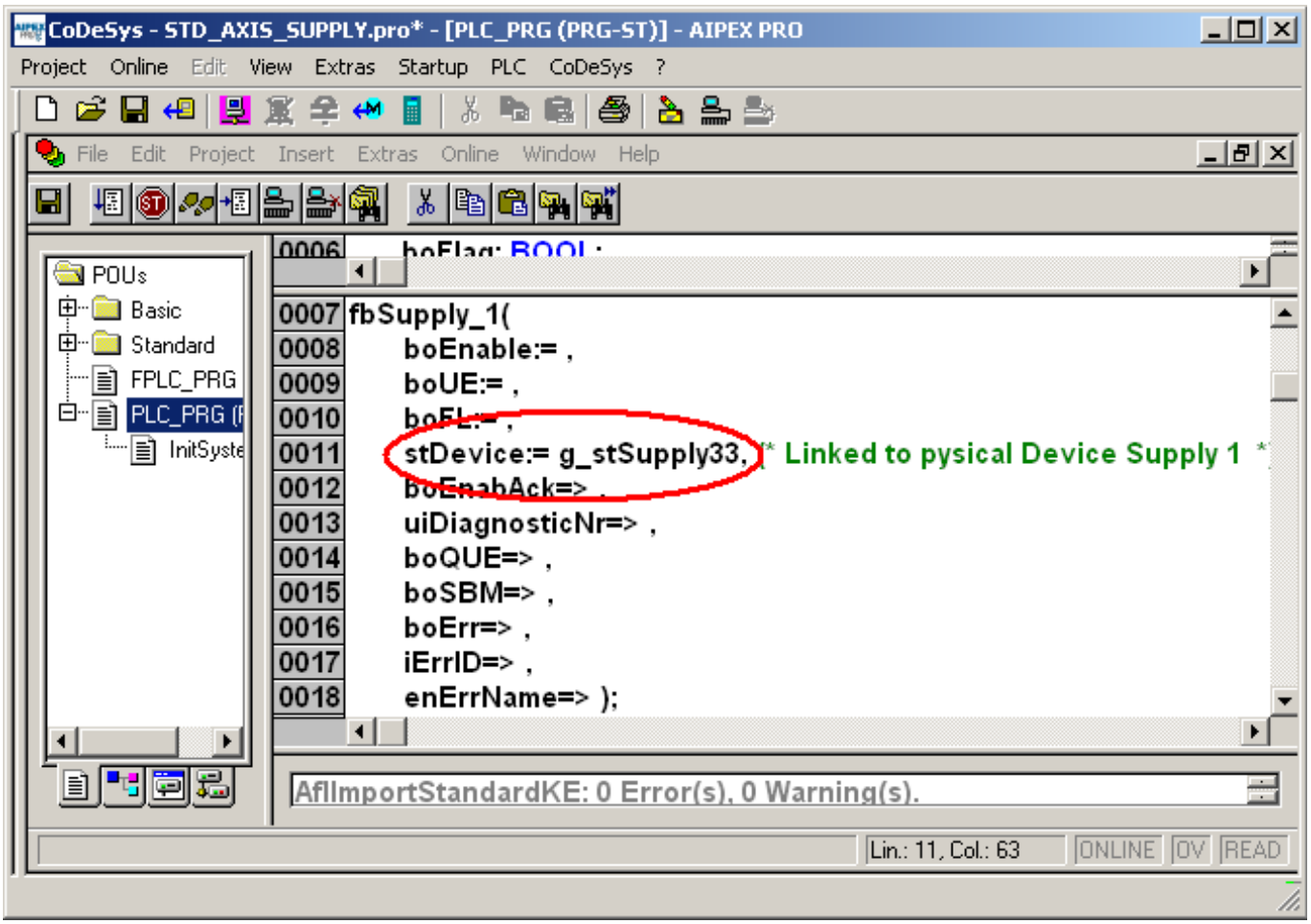
Fügen Sie ihren Funktionsbaustein 'fbSupply_1' in den Programmeditor ein.
Klicken Sie dazu in den Programmeditor. Mit der **'F2'** Taste öffnet sich der 'Input assistant'.



Sie finden den Funktionsbaustein 'fbSupply_1' in der Eingabehilfe unter den 'Local Variables'.



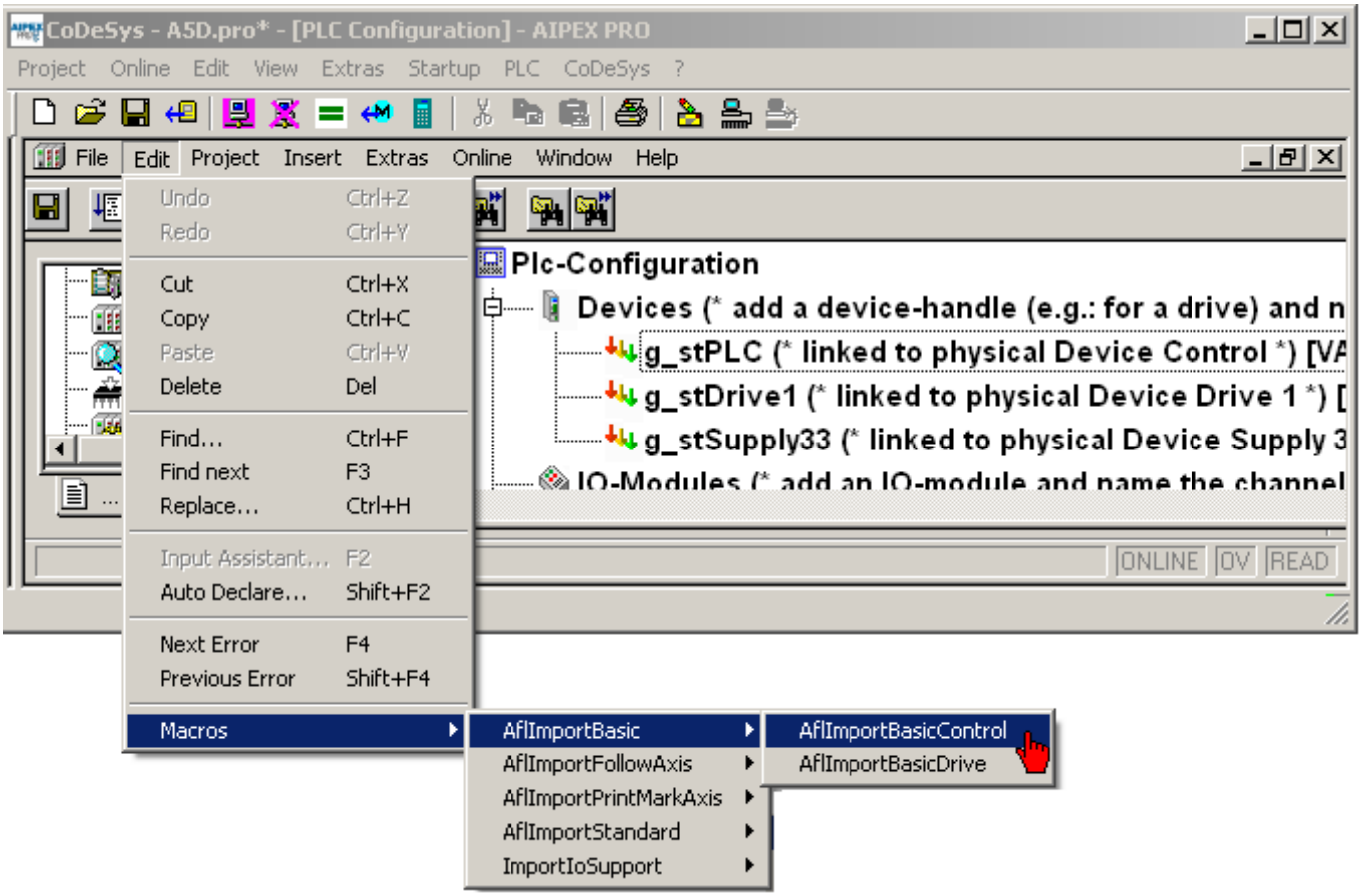
Die IN_OUT-Variable 'stDevice' muss immer zugewiesen werden.



4.3.2.5 Basis Steuerung importieren

'AflImportBasicControl' muss pro Projekt nur einmal eingefügt werden.

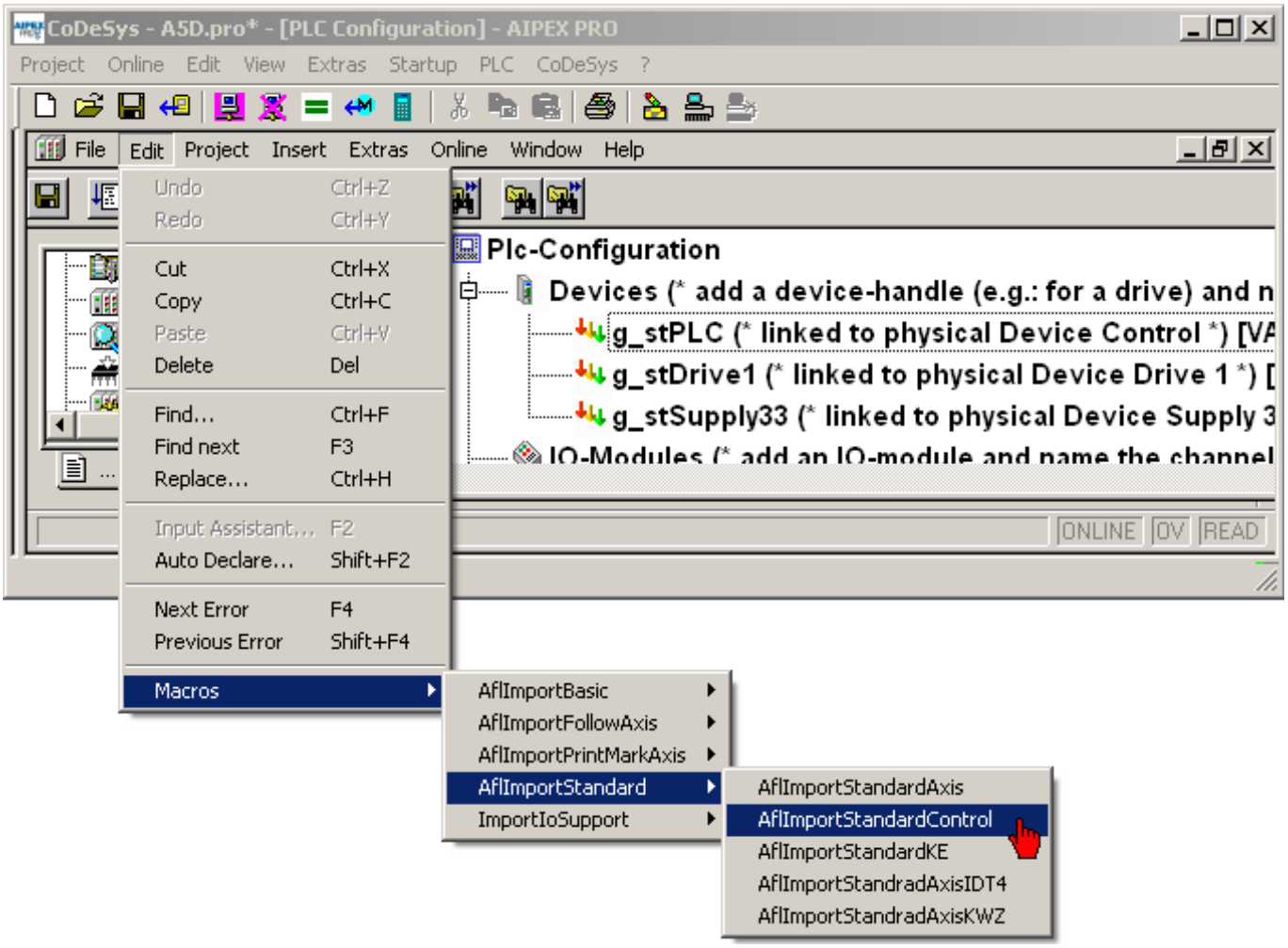
Importieren Sie die Funktion **'AflImportBasicControl'** über das Menü **'Edit' 'Macros' 'AflImportBasic'**.



4.3.2.6 Standard Steuerung importieren

Beschreibung der Standard Steuerung: Siehe 'STANDARD_CONTROL (FB)' auf Seite 197.

Importieren Sie die Funktion **'AflImportStandardControl'** über das Menü **'Edit' 'Macros' 'AflImportStandard'**.

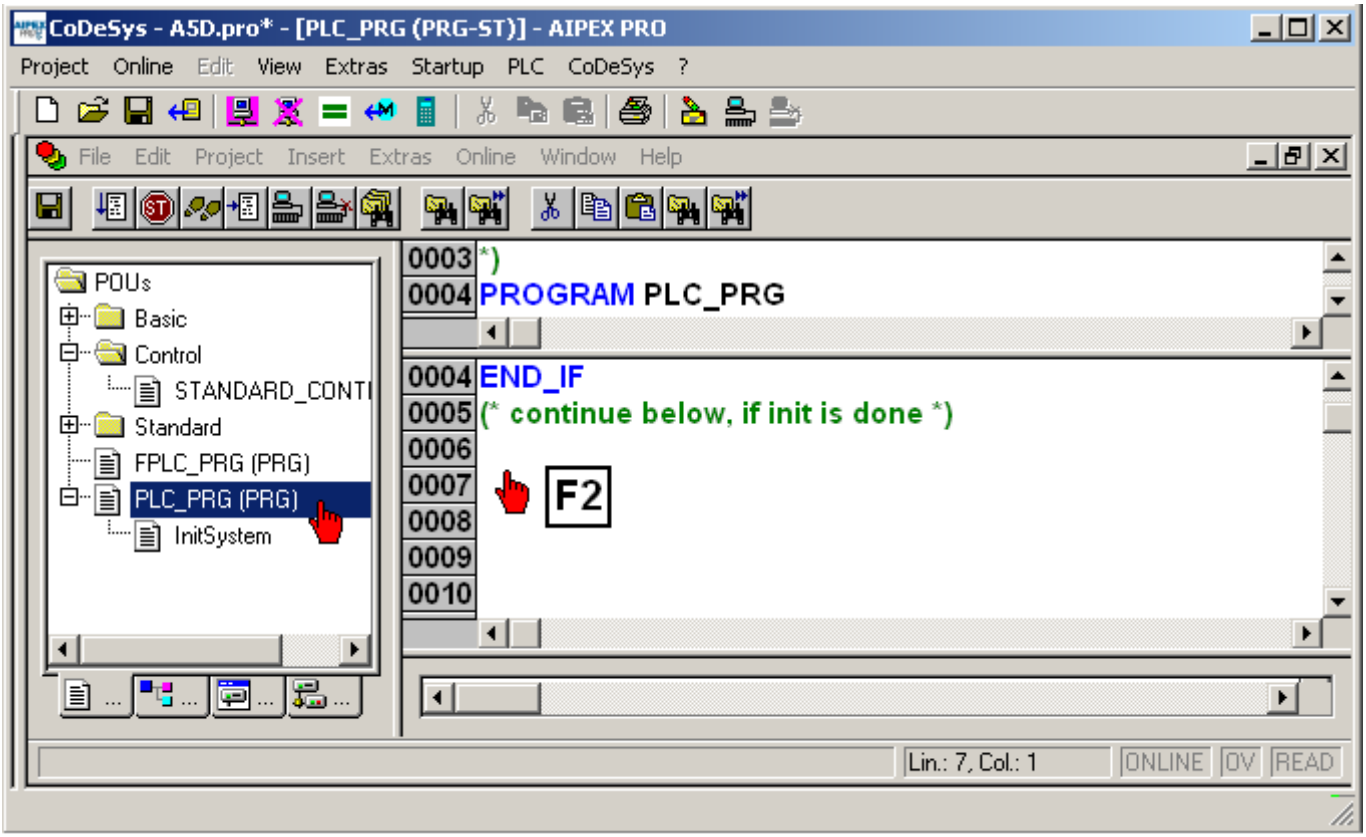


4.3.2.7 Standard Steuerung instanzieren

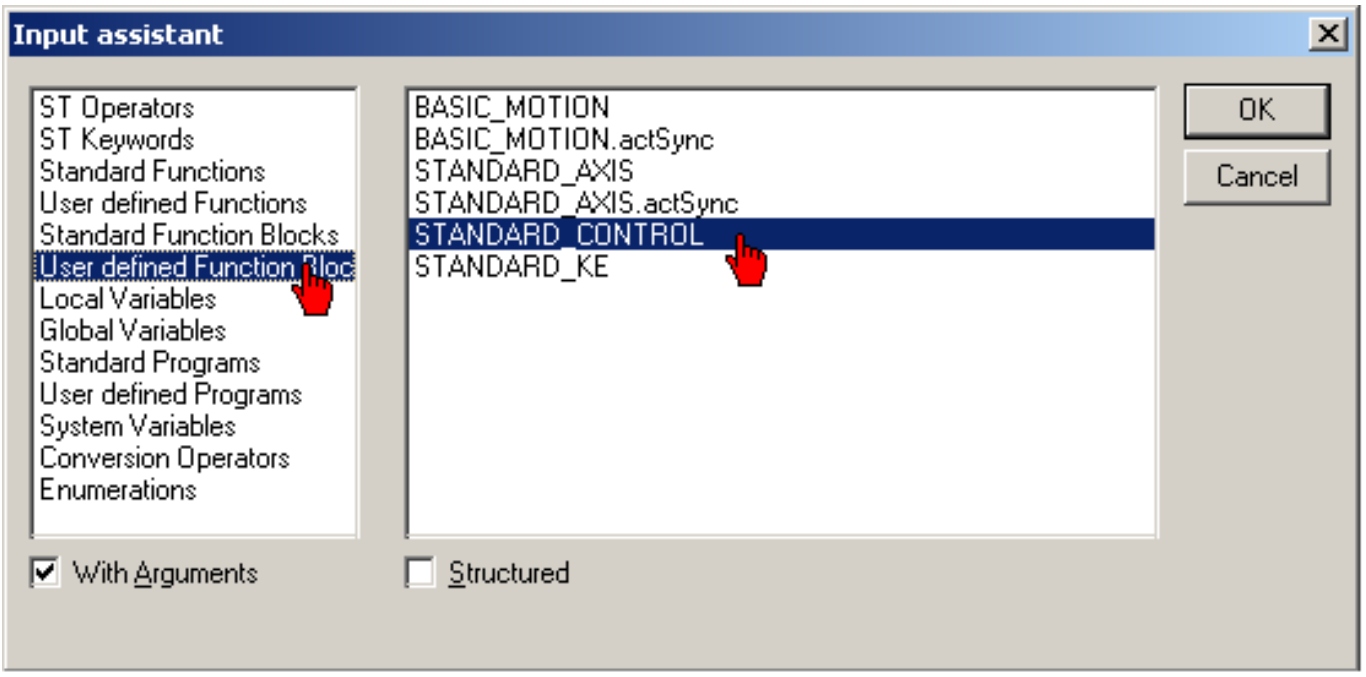
(Instanzieren von Bausteinen Variante 1)

Fügen Sie den Funktionsbaustein 'STANDARD_CONTROL' in den Programmeditor ein.

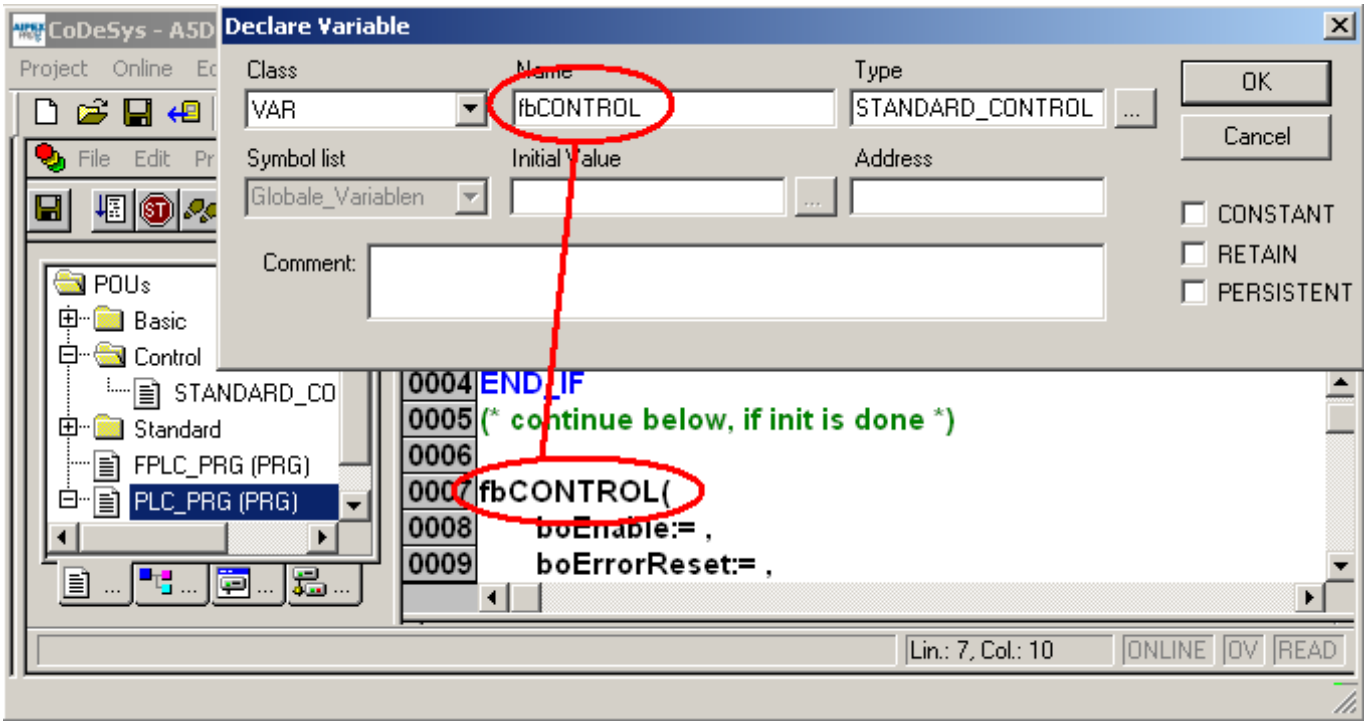
Klicken Sie dazu in den Programmeditor. Mit der **'F2'** Taste öffnet sich der 'Input Assistent'.



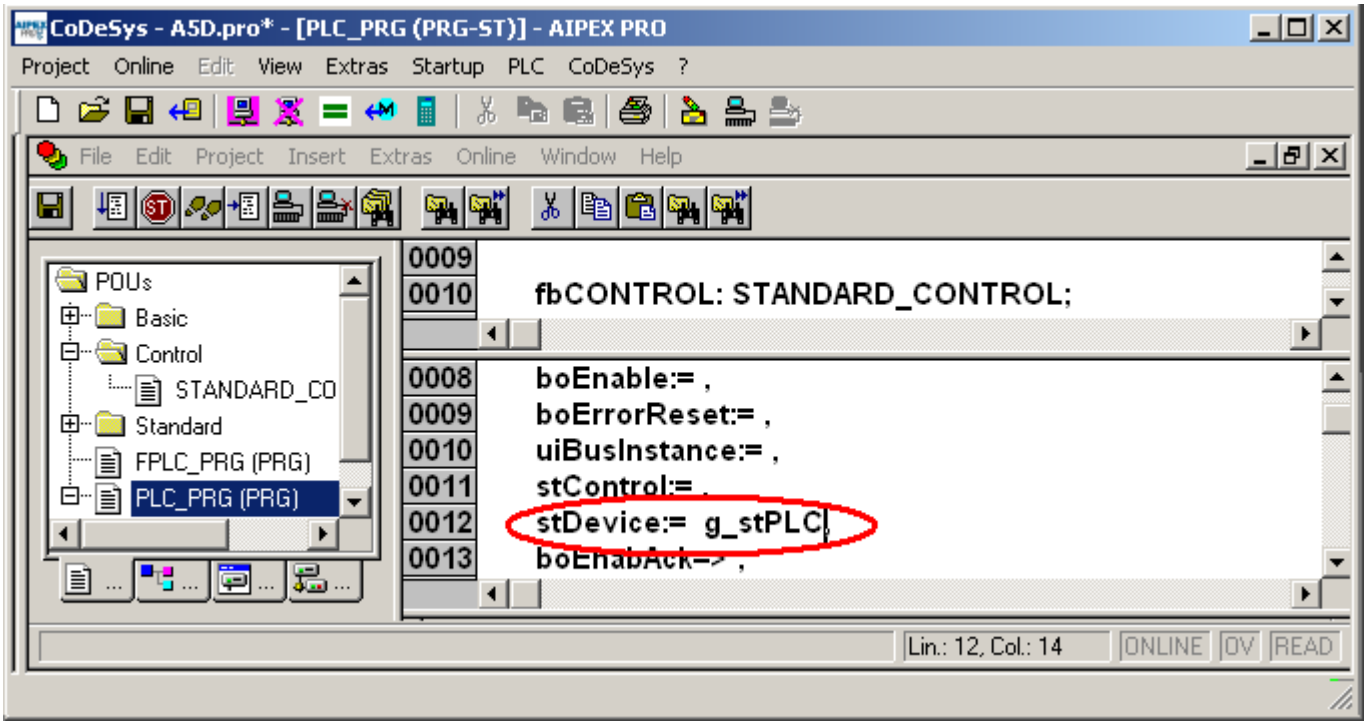
Sie finden den Funktionsblock 'STANDARD_CONTROL' in der Eingabehilfe unter den **'User defined function Blocks'**



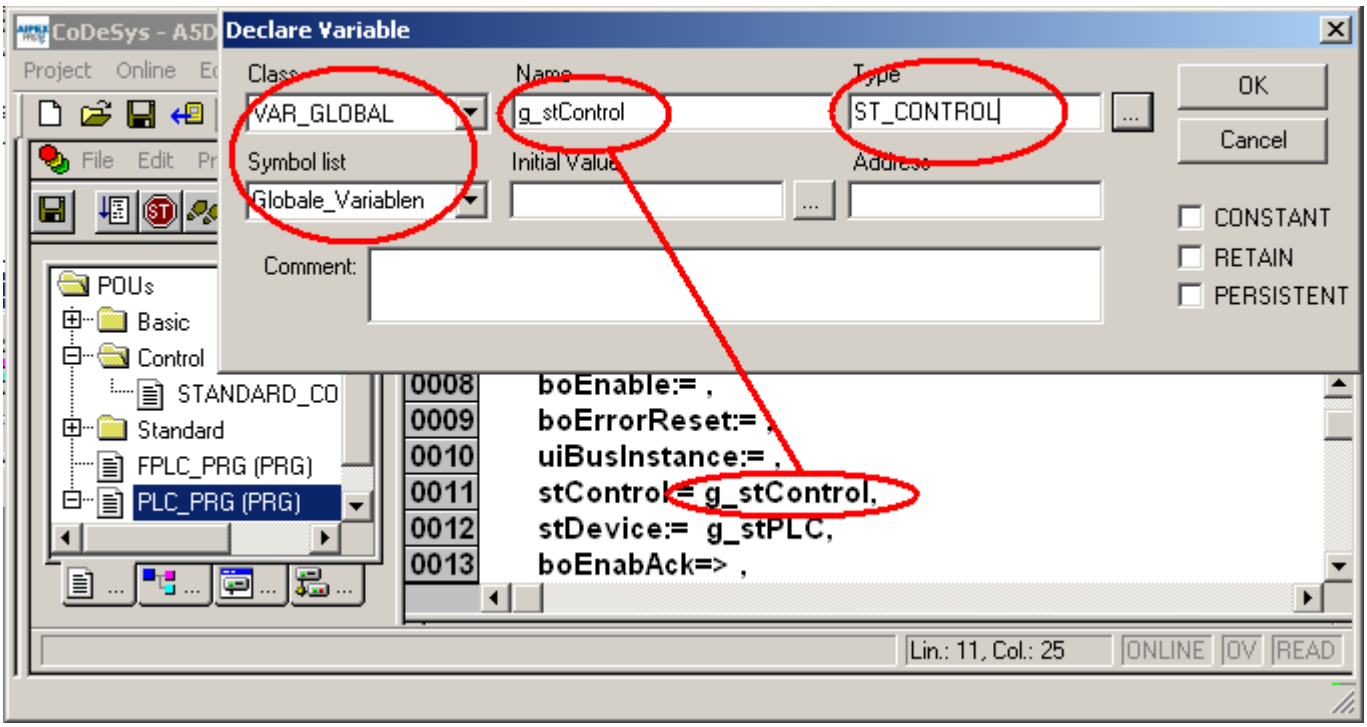
Legen Sie eine Instanz (Kopie) vom **FB** 'STANDARD_CONTROL' an, indem Sie einen eigenen Namen vergeben. Im Beispiel: fbControl.
Klicken Sie anschließend mit der Maus in den Programmeditor um das Variablendeklarationsfenster 'Declare Variable' zu öffnen.
Typ = Original Bausteinname 'STANDARD_CONTROL'



Die IN_OUT-Variablen 'stControl' und 'stDevice' müssen immer zugewiesen werden.
stDevice:= Link zum symbolischen Gerätenamen (im Beispiel: g_stPLC)



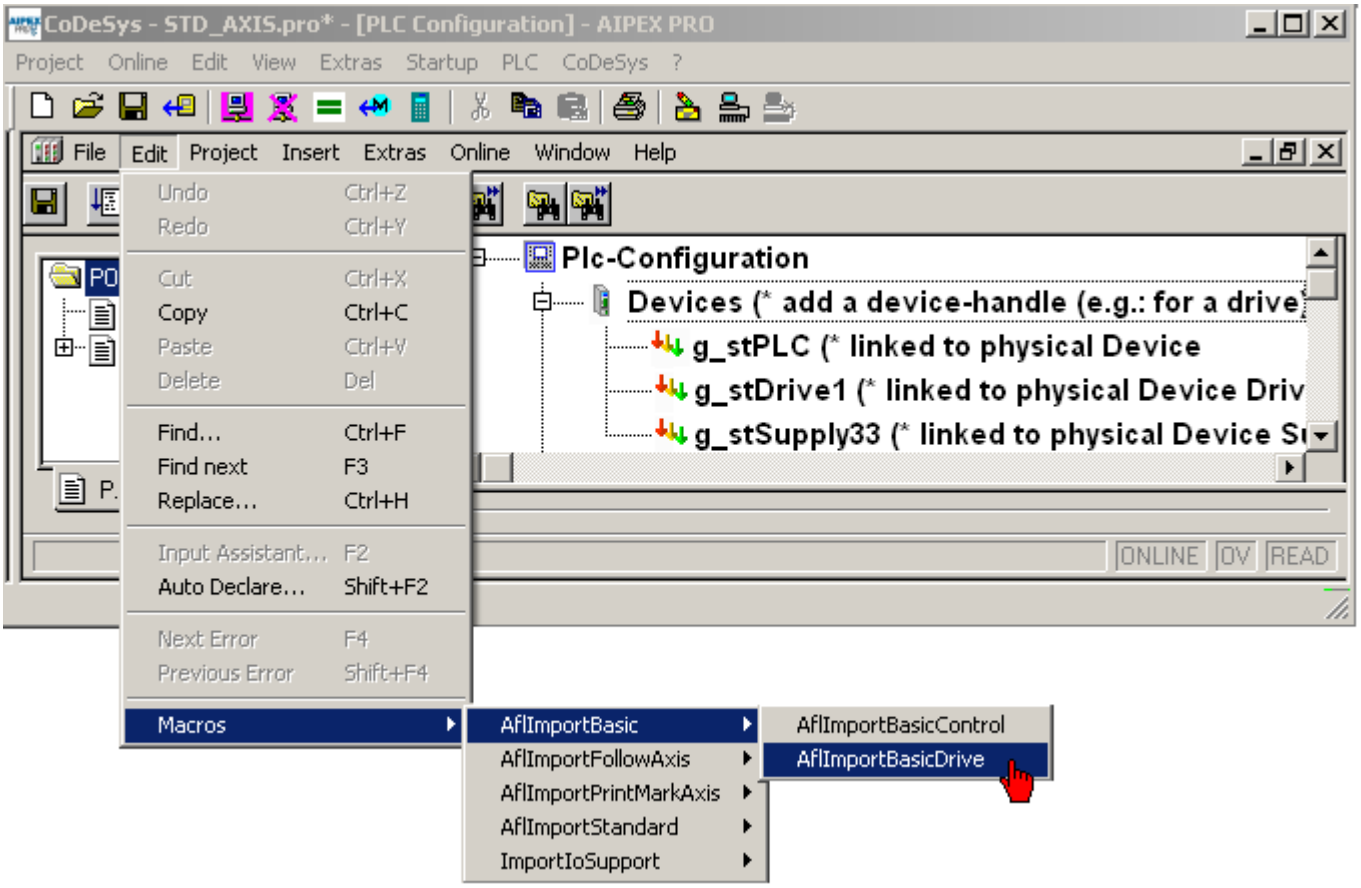
Legen Sie eine globale Stukturvariable an. Im Beispiel 'g_stControl'.
Klicken Sie anschließend mit der Maus in den Programmeditor um das Variablendeklerationsfenster 'Declare Variable' zu öffnen.
Klasse = VAR_GLOBAL
Symbol Liste = Globale_Variablen
Typ = Struktur 'ST_CONTROL'



4.3.2.8 Basis Achse importieren

Beschreibung der Basis Achse: Siehe 'Beschreibung Basis Achse' auf Seite 785.
'AflImportBasicDrive' muss pro Projekt nur einmal eingefügt werden.

Importieren Sie die Funktion **'AflImportBasicDrive'** über das Menü **'Edit' 'Macros' 'AflImportBasic'**.



4.3.2.8.1 Beschreibung Basis Achse

Die Basis Achse muss in jedes PLC Projekt einmal eingefügt werden.
Sie ist die Basis für alle nachfolgenden Standard, Print und Following –Achsen.

Mit der Basis Achse wird der Funktionsbaustein 'BASIC_MOTION' eingefügt.

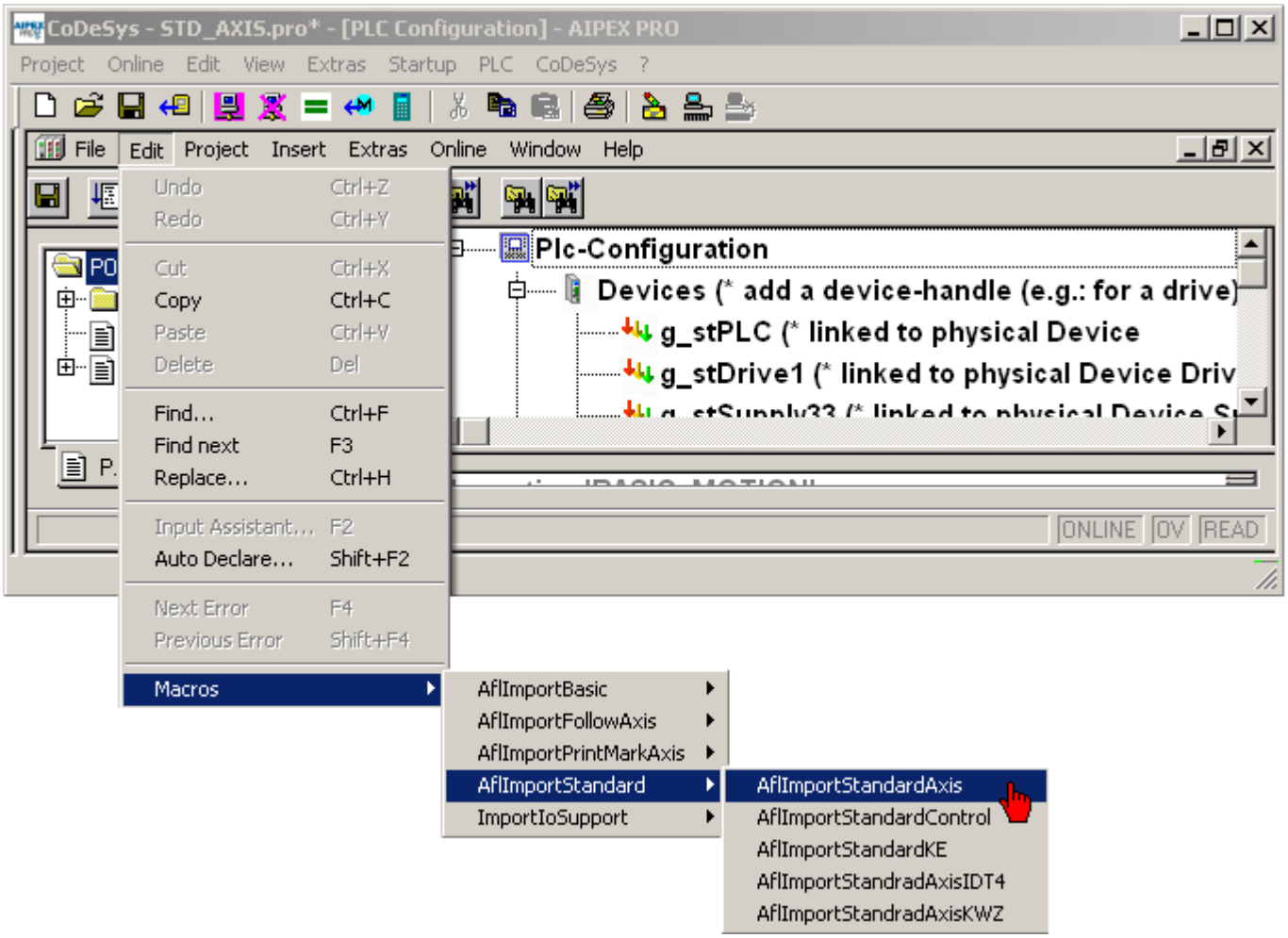
Funktionen des FB 'BASIC_MOTION'

- Referenzpunktfahrt
- Absolute Positionierung
- Relative Positionierung
- Drehzahlregelung
- Tippbetrieb

4.3.2.9 Standard Achse importieren

Beschreibung der Standard Achse: Siehe 'STANDARD_AXIS (FB) (STANDARD_AXIS_KWZ, STANDARD_AXIS_IDT4, STANDARD_AXIS_ihX)' auf Seite 202.

Importieren Sie die Funktion **'AflImportStandardAxis'** über das Menü **'Edit' 'Macros' 'AflImportStandard'**.



4.3.2.10 Achsstruktur deklarieren

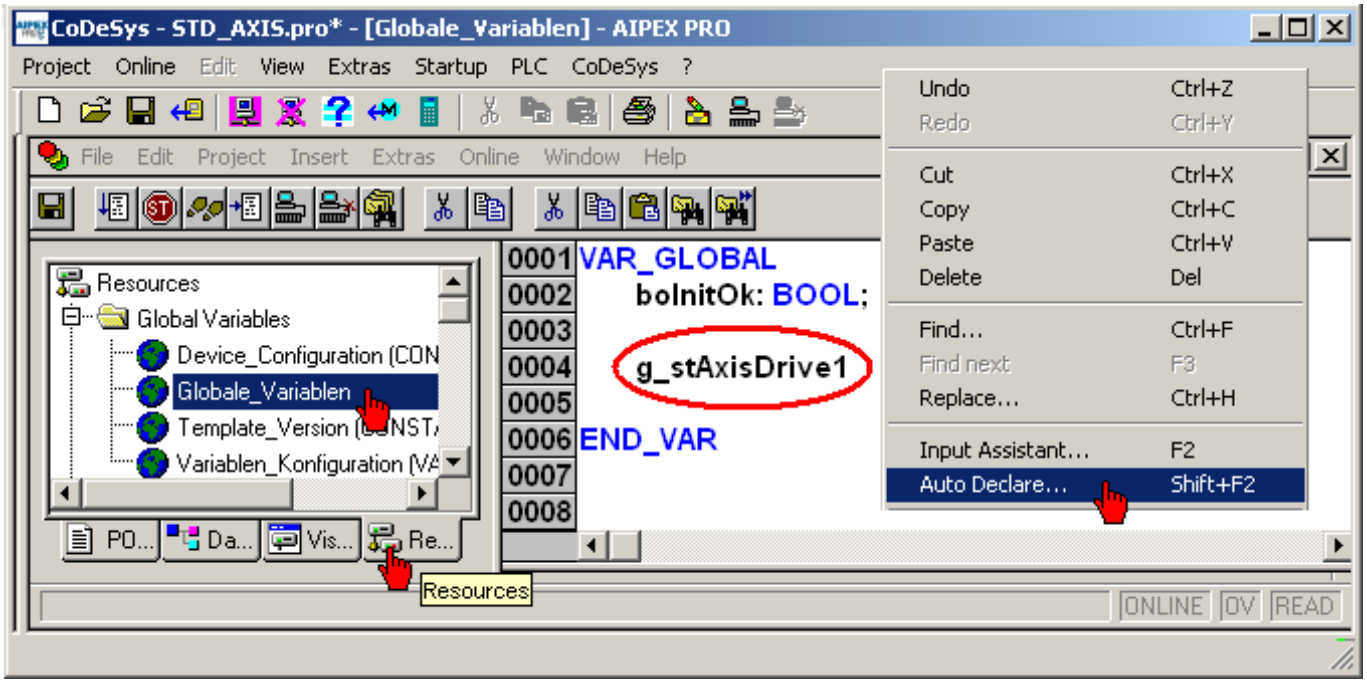
Beschreibung der Achstruktur: Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.

Legen Sie im Variablendeklerationseditor für jede physikalisch vorhandene Achse eine globale Strukturvariable vom Typ 'ST_AXIS_DRIVE' an (im Beispiel 'g_stAxisDrive1').

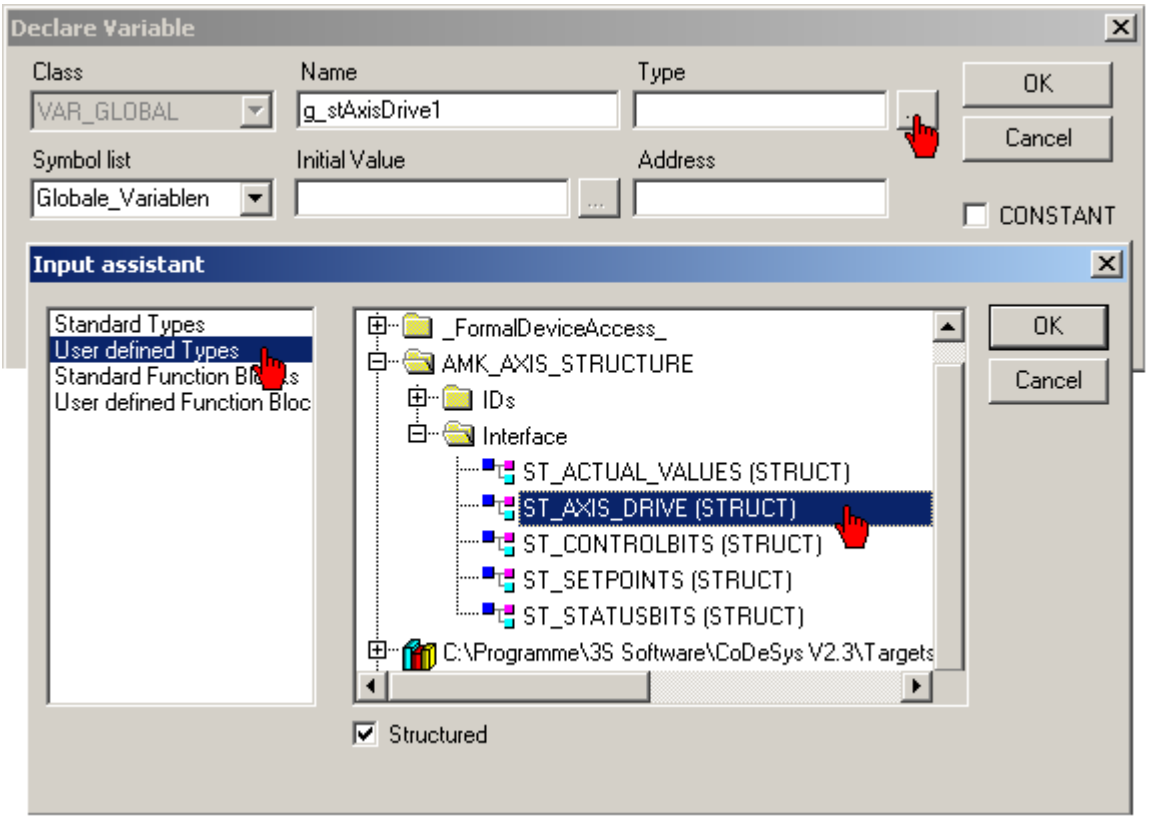
Den Variablendeklerationseditor für globale Variablen finden Sie unter **'Ressourcen' 'Global Variables' 'Globale_Variablen'**

Klicken Sie mit der RMT auf die Strukturvariable. Anschließend können Sie das Variablen Deklarationsfenster 'Declare Variable' öffnen (Umschalt + F2).

Die deklarierte Strukturvariable ermöglicht Ihnen den Zugriff auf alle achsspezifischen Daten. Siehe 'ST_AXIS_DRIVE (ST)' auf Seite 193.



Sie finden die Struktur 'ST_AXIS_DRIVE' in der Eingabehilfe unter den **'User defined Types' 'AMK_AXIS_STRUCTURE' 'Interface'.**

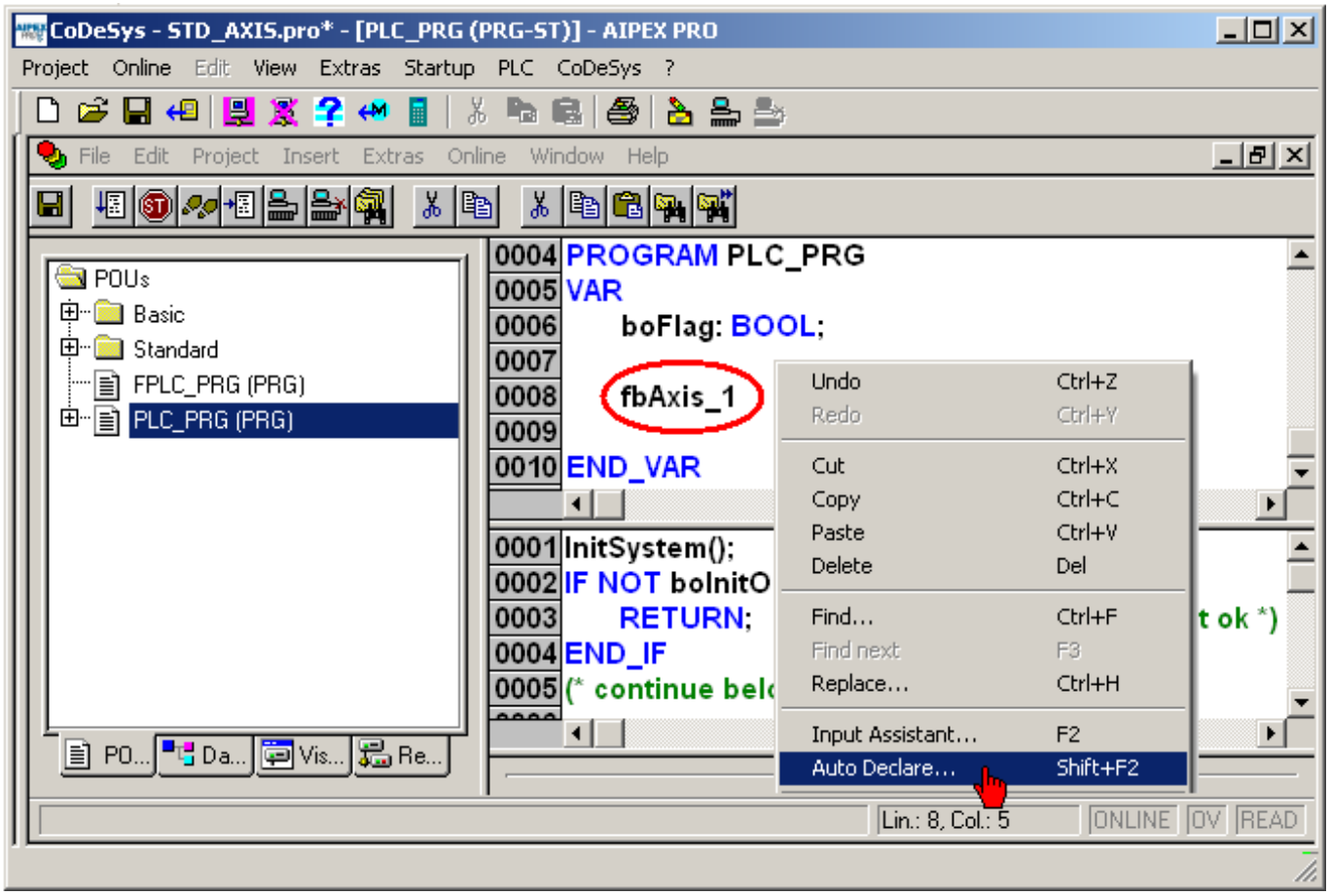


4.3.2.11 Standard Achse instanzieren

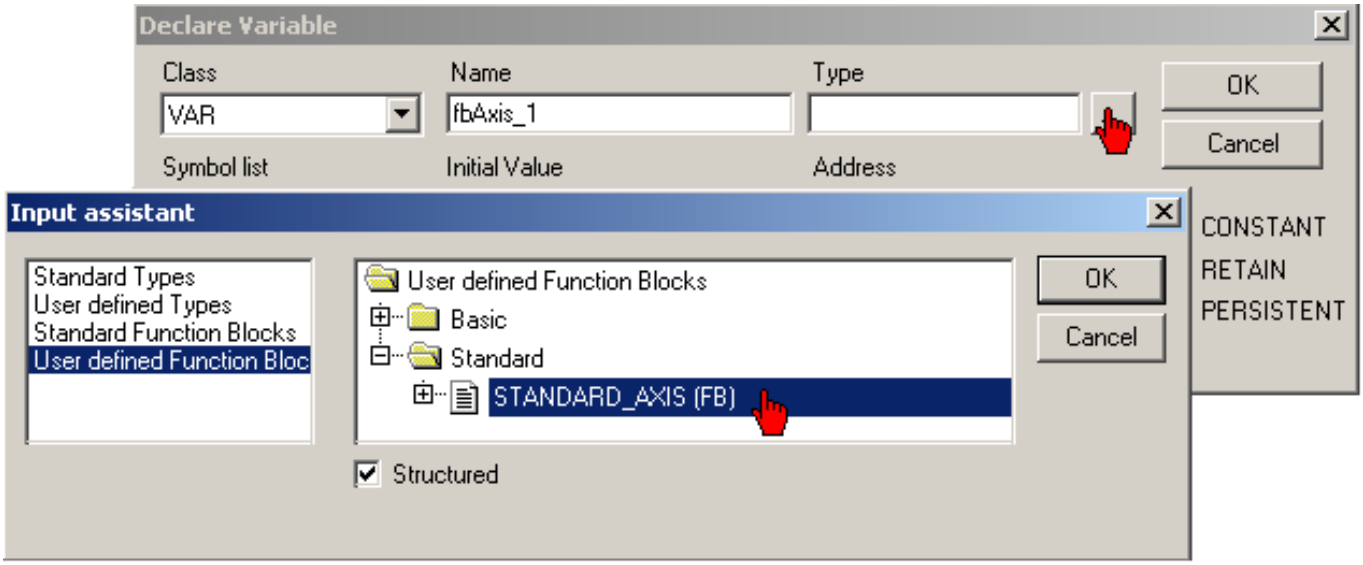
(Instanzieren von Bausteinen Variante 2)

Legen Sie im PLC_PRG Variablendeklerationseditor für jede physikalisch vorhandene Achse eine Instanz vom Typ 'STANDARD_AXIS' an (im Beispiel fbAxis_1).

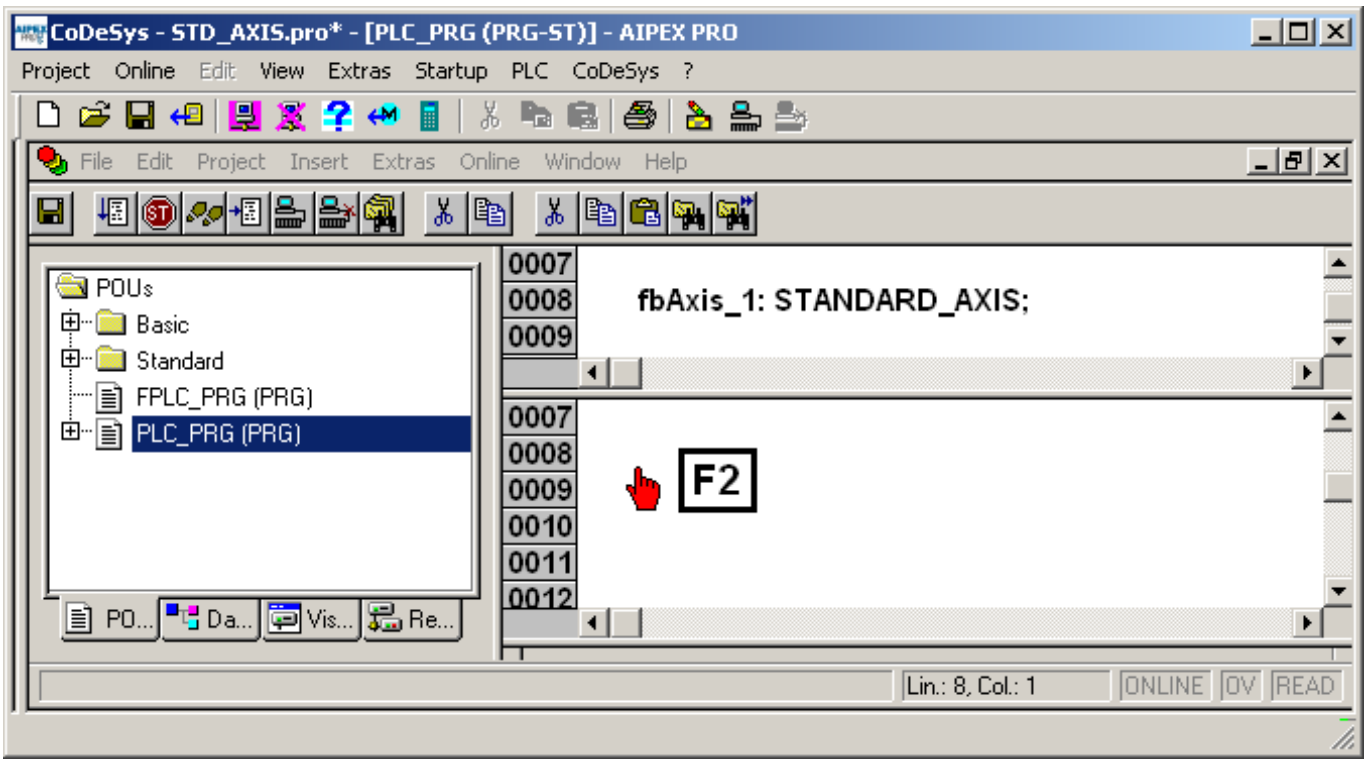
Klicken Sie mit der RMT auf die Variable. Anschließend können Sie das Variablen Deklarationsfenster 'Declare Variable' öffnen (Umschalt + F2).



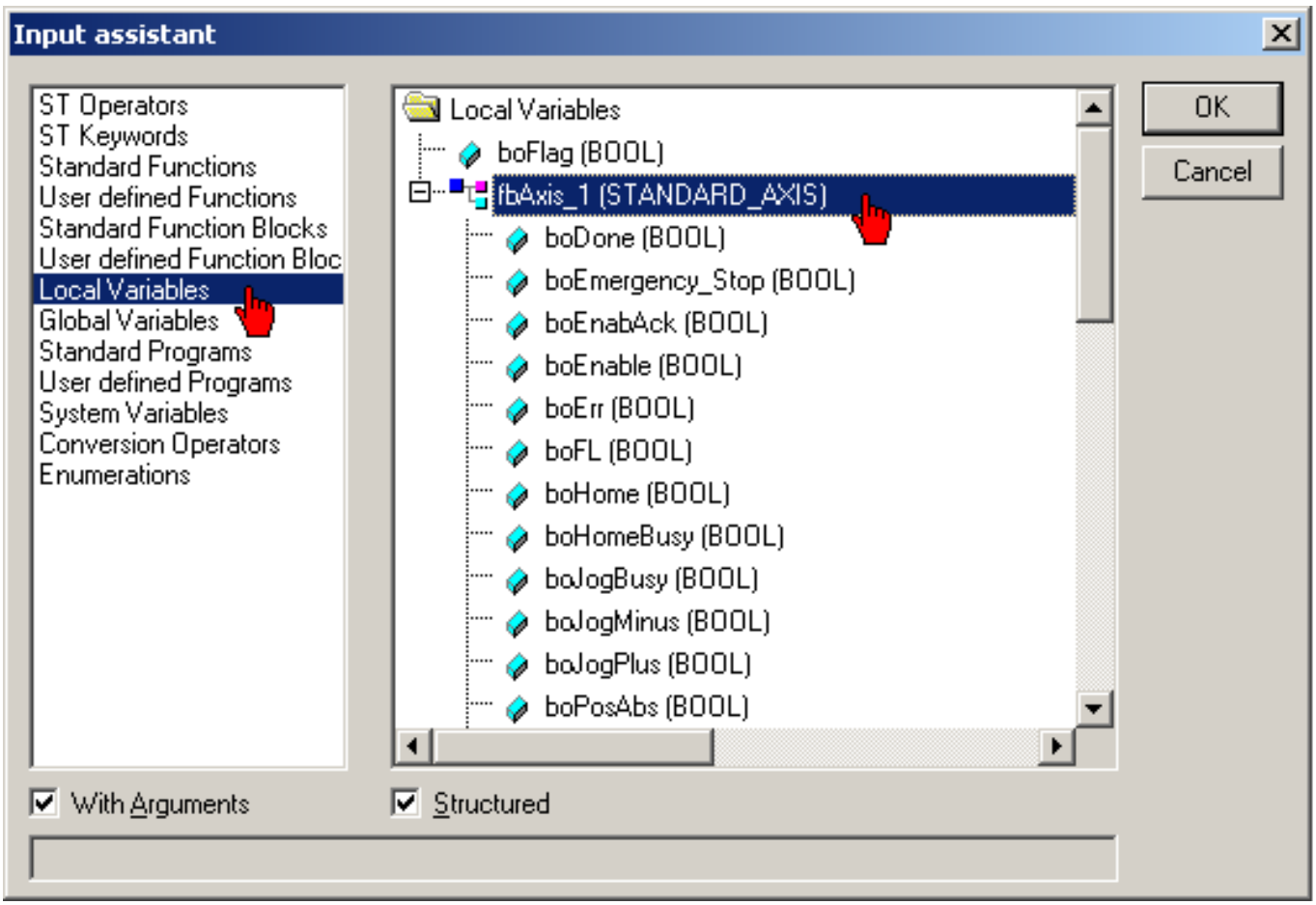
Sie finden den Funktionsblock 'STANDARD_AXIS' in der Eingabehilfe unter den **'User defined Function Blocks' 'Standard'.**

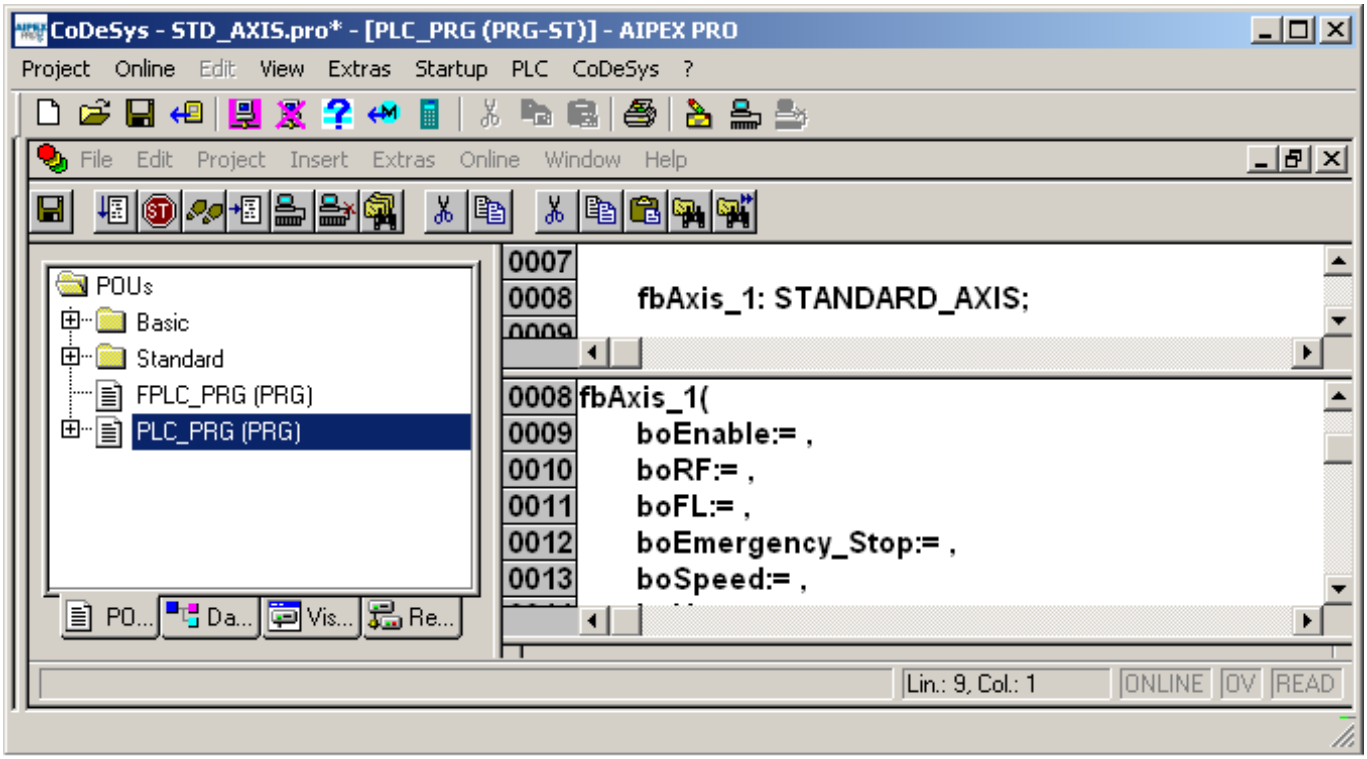


Fügen Sie ihren Funktionsbaustein 'fbAxis_1' in den Programmeditor ein.
Klicken Sie dazu in den Programmeditor. Mit der **'F2'** Taste öffnet sich der **'Input assistant'**.

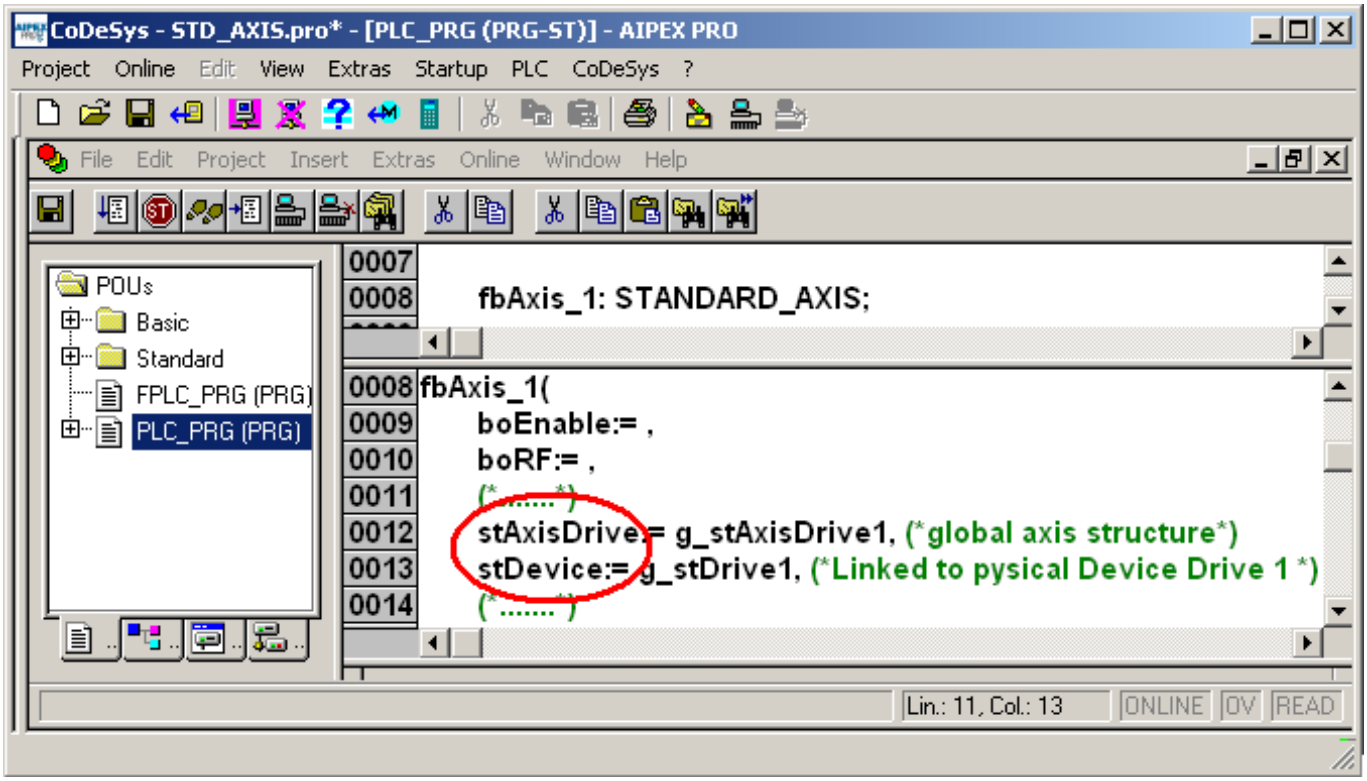


Sie finden den Funktionsbaustein **'fbAxis_1'** in der Eingabehilfe unter den **'Local Variables'**.





Die IN_OUT-Variablen 'stAxisDrive' und 'stDevice' müssen immer zugewiesen werden.



Standard Achse instanzieren

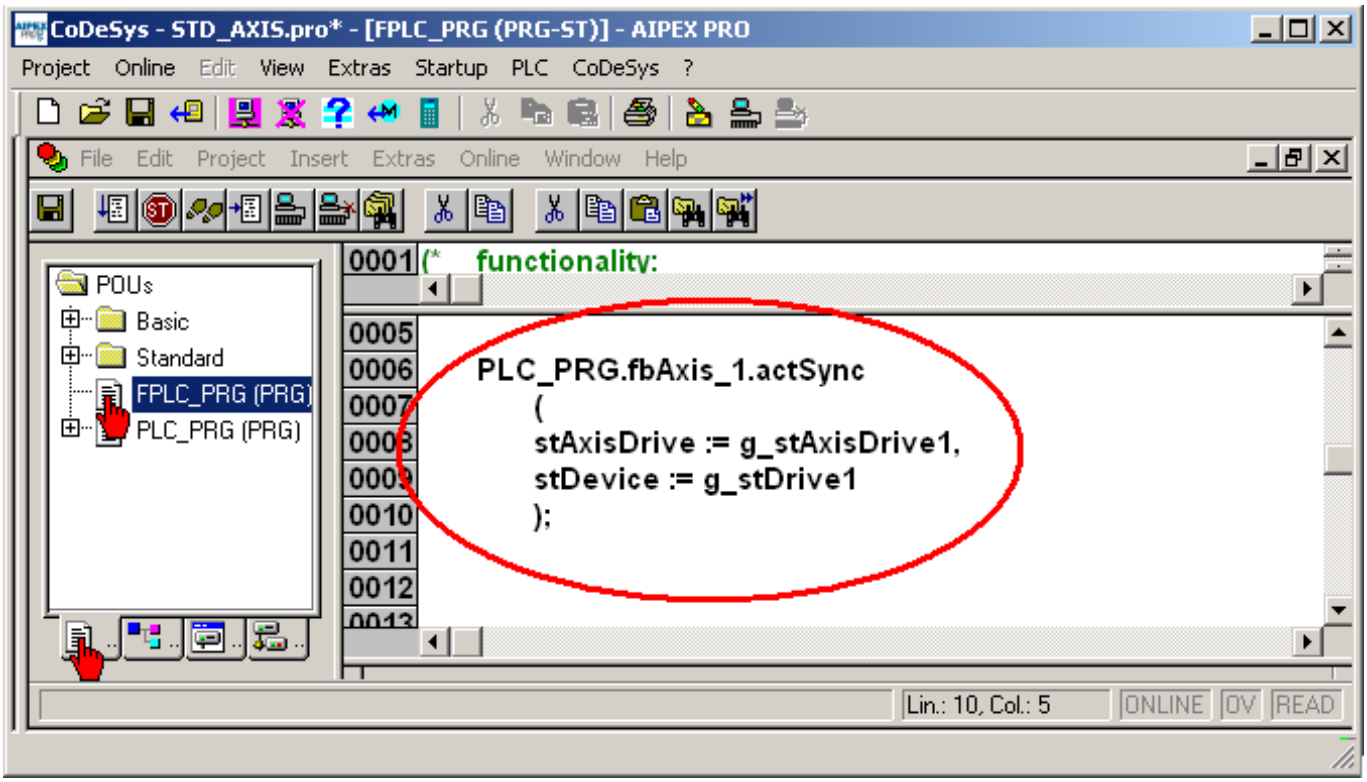
Alternativ können Sie auch einen eigenen Funktionsbaustein generieren in den Sie den AMK Funktionsbaustein 'STANDARD_AXIS' einfügen und instanzieren.

4.3.2.12 Synchrone Aktionen

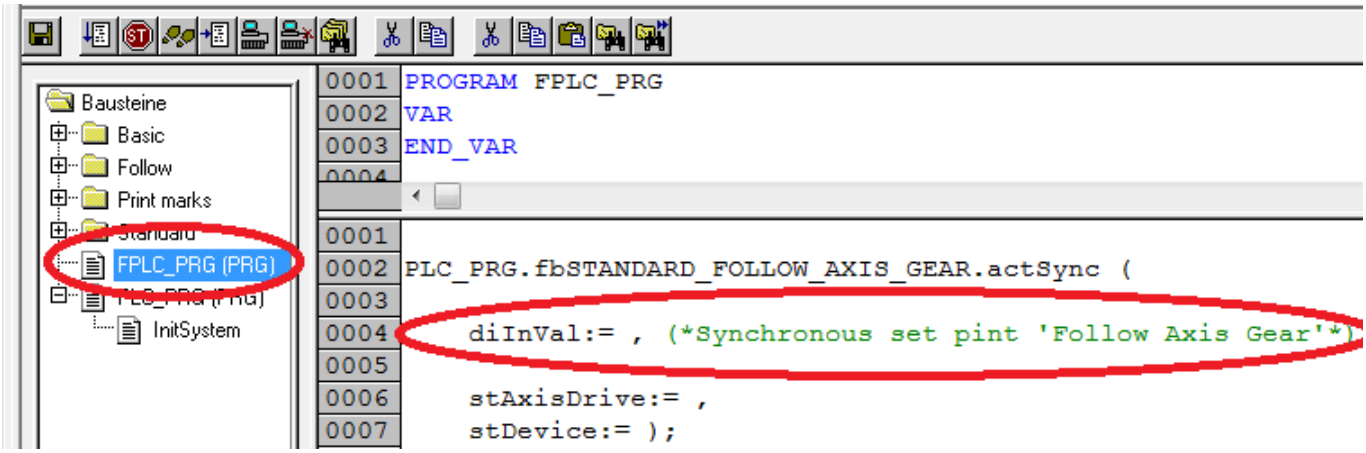
Der Funktionsbaustein 'STANDARD_AXIS' enthält synchrone Programmteile die mit dem Antrieb synchronisiert sind.
Die synchrone Aktion muss im 'FPLC_PRG' aufgerufen werden. Ohne diesen Aufruf lässt sich der FB nicht einschalten.

Erstellen Sie dazu folgenden Programmcode im 'FPLC_PRG'.

```
PLC_PRG.fbAxis_1.actSync
(
stAxisDrive := (* Achsstruktur im Beispiel g_stAxisDrive1 *),
stDevice := (* symbolischer Gerätename im Beispiel g_stDrive1*)
);
```



Synchrone Sollwertvorgabe am Beispiel STANDARD_FOLLOW_AXIS_GEAR (FB)
Die synchronen Sollwertvorgabe erfolgt im FPLC_PRG / *.actSync Aufruf.
Fügen Sie die Input Variable 'diInVal' (Sollwertvorgabe Folgebetrieb) als Übergabevariable ein.



4.3.2.13 Istwerte lesen

Die Struktur 'ST_ACTUAL_VALUES' ist eine Unterstruktur der 'ST_AXIS_DRIVE'.

Über 'ST_AXIS_DRIVE.ST_ACTUAL_VALUES' kann auf die Ist-Werte der Achse zugegriffen werden.

Inhalt:

Name	Beschreibung	Einheit
diID00040_Velocity_feedback_value	aktueller Drehzahlistwert	0,0001 U/min
diID00051_Position_feedback_value	aktueller Lageistwert	Inkremente
diID00084_Torque_feedb_val	aktueller Momentistwert	0,1 % Mn
diID00130_Probe1_val_p_edge	Messwert 1 positive Flanke 1)	Inkremente
diID00131_Probe1_val_n_edge	Messwert 1 negative Flanke 1)	Inkremente
diID00189_Following_dist	Schleppfehler 1)	Inkremente
diRectangularPulsInput_X132	Eingangsimpulse des Rechteckgebereingangs	Inkremente

1) Wirkt nur bei EtherCAT!

Die Messwerte und der Schleppfehler sind im PLC Programmcode der Standard Achse auskommentiert und werden dadurch in der Struktur nicht aktualisiert.

Auskommentierter Teile im PLC Programmcode siehe:

- Deklarationseditor Standard Achse: fbGetStatusActVal
- Programmcode Editor: Auswertung der Variablen 'boErr'
- Programmcode Editor: Auswertung der Variablen 'boErrID'
- Aktion Standard Achse actSync: PLC Programmcode Editor , Abschnitt 'Read status'

Speicherort:

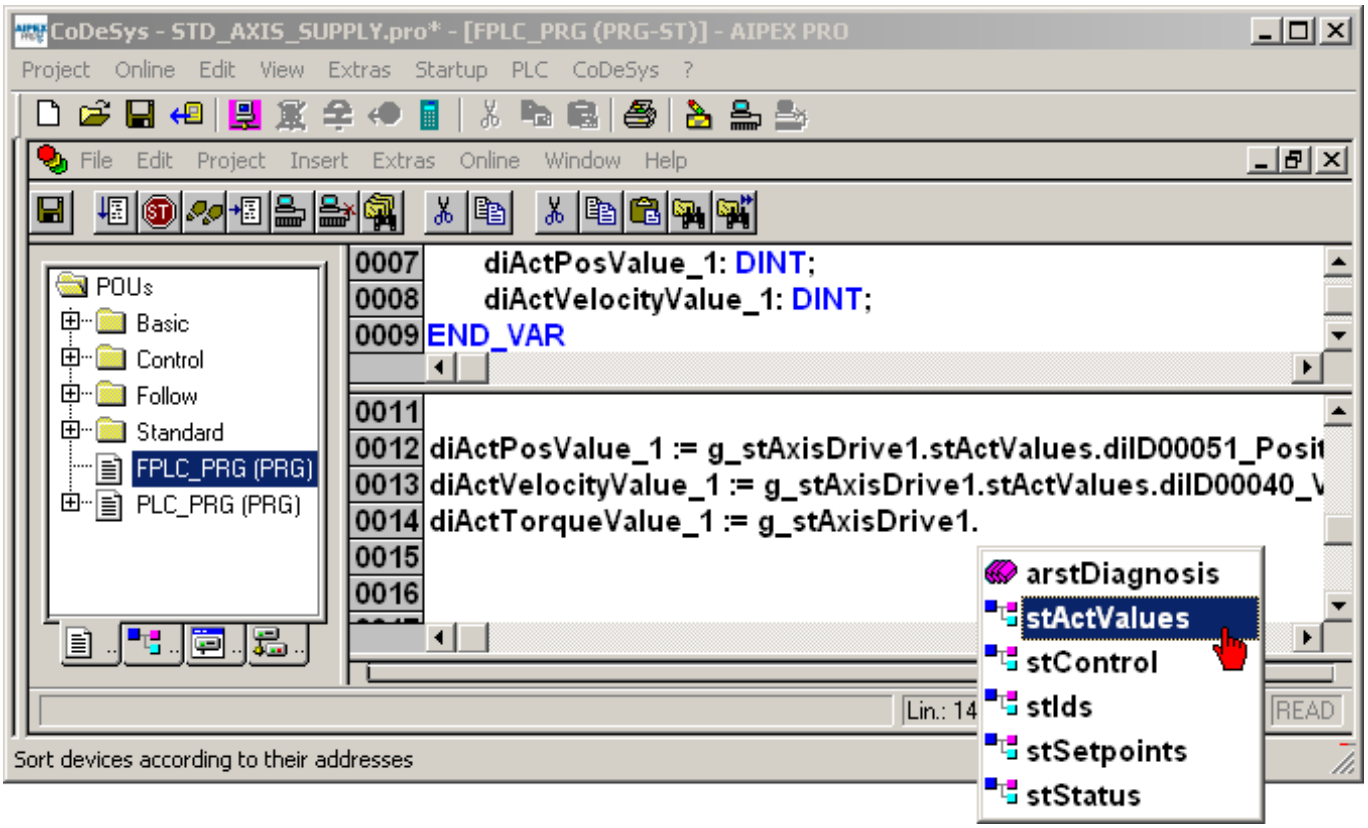
CODESYS V2

'Data types' → 'AMK_AXIS_STRUCTURE' → 'Interface'

CODESYS V3

'POUs' → 'Control' → 'Standard' → 'Types' → 'AMK_AXIS_STRUCTURE' → 'Interface'

Beispiel: Die Istwerte werden in lokale Variablen kopiert.



4.3.2.14 Statusbits lesen

Die Struktur 'ST_STATUSBITS' ist eine Unterstruktur der 'ST_AXIS_DRIVE'.

Über 'ST_AXIS_DRIVE.ST_STATUSBITS' kann auf die Statusbits der Achse zugegriffen werden.

Inhalt:

Name	Beschreibung	Einheit
boID330_VelocityWindow	Code 330 nIst = nSoll ID157 Drehzahlfenster ("Auf Drehzahl")	-
boID331_ZeroVelocityWindow	Code 331 nist < nmin ID124 Stillstandsfenster	-
boID332_VelocityWindowLimit	Code 332 nIst < nx ID125 Drehzahlschwelle nx	-
boID333_TorqueLimit	Code 333 Md>=Mdx ID126 Drehmomentschwelle Mdx	-
boID334_NegTorqueLimit	Code 334 MSoll >= MGrenz Sollmoment Grenzmoment (ID82/ID83)	-
boID335_NegVelocityLimit	335 nSoll >= nGrenz Solldrehzahl . Grenzdrehzahl (ID38/ID39)	-
boID336_InPosition	In Position [1)]	-
boID33029_SBM	Systembereitmeldung	-
boID33030_QUE	Quittierung Umrichter Ein	-
boID33031_QRF	Quittierung Reglerfreigabe	-
boID33036_RFP_known	Code 33036 Referenzpunkt bekannt	-
boID33074_Warning_active	Code 33074 Warnung aktiv Sammelwarnung	-
boDiagOk	Diagnose abgeschlossen	-
boSetPosEnabAck	Nebenbetriebsart1 Freigabe Positionsvorgabe	-
boSetVelocityEnabAck	Nebenbetriebsart2 Freigabe Drehzahl	-
boSetTorqueEnabAck	Nebenbetriebsart3 Freigabe Momentvorgabe	-
boResidDistDone	Restweg gelöscht	-

1) Das 'In Positions' Bit wird nur bei Reglerkarten mit internem Interpolator ausgewertet. Bei Ethernet Reglerkarten wirkt das Bit nur bei den Kommandos absolute und relative Positionierung (boPosAbs / boPosRel)

Speicherort:

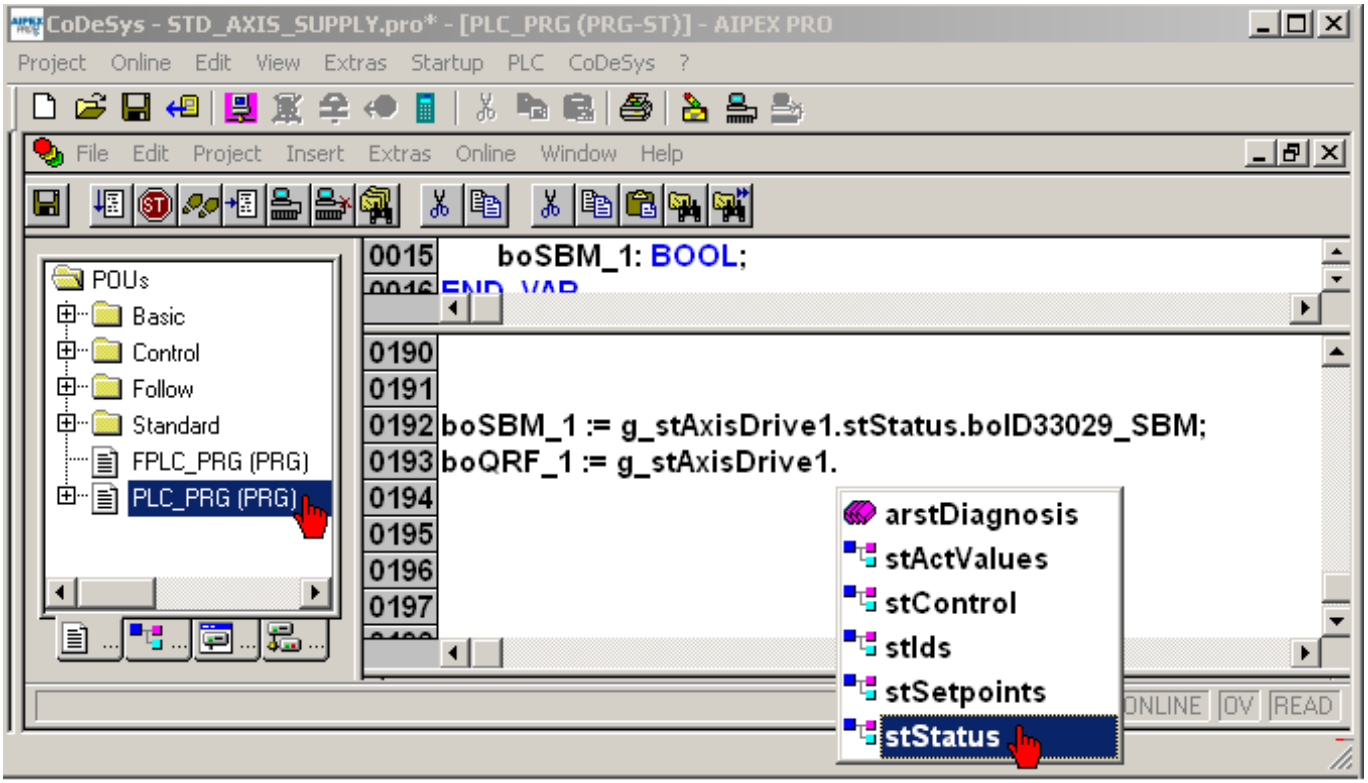
CODESYS V2

'Data types' → 'AMK_AXIS_STRUCTURE' → 'Interface'

CODESYS V3

'POUs' → 'Control' → 'Standard' → 'Types' → 'AMK_AXIS_STRUCTURE' → 'Interface'

Beispiel: Die Statusbits werden in lokale Variablen kopiert.



Beschreibung Statusbits erweitern: Siehe Statusbits erweitern auf Seite 836.

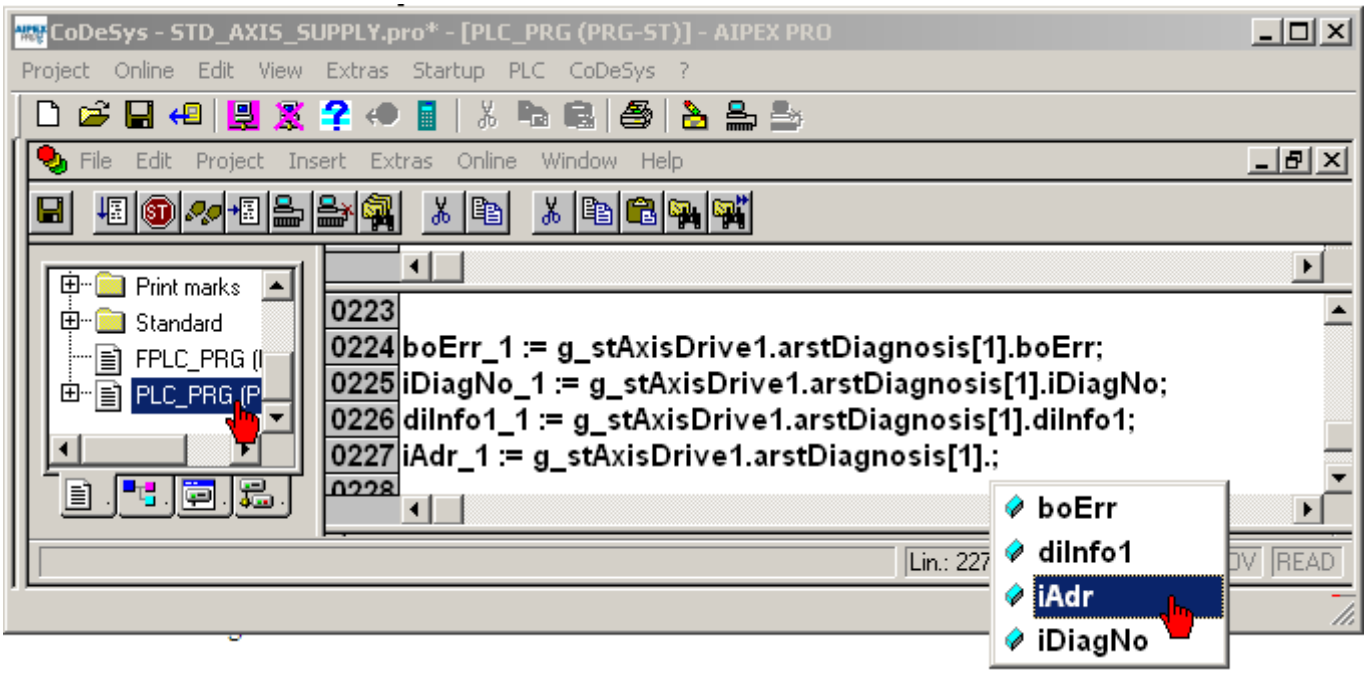
4.3.2.15 Diagnose lesen

Das Array 'arstDiagnosis' ist ein Bestandteil der Struktur der 'ST_AXIS_DRIVE' und 'ST_CONTROL'.
Über z.B. 'ST_AXIS_DRIVE.arstDiagnosis [1-10]' kann auf die Diagnosemeldungen zugegriffen werden.
Das Diagnosearray beinhaltet bis zu 10 Diagnosemeldungen die wie folgt aufgebaut sind:

Name	Beschreibung	Einheit
arstDiagnosis [1]	boErr: Fehlerbit für Fehler 1 iDiagNo: Diagnosenummer für Fehler 1 diInfo1: Zusatzinfo 1 für Fehler 1 iAdr: Geräteadresse*	-
arstDiagnosis [2]	boErr: Fehlerbit für Fehler 2 iDiagNo: Diagnosenummer für Fehler 2 diInfo1: Zusatzinfo 1 für Fehler 2 iAdr: Geräteadresse*	-
...	...	
arstDiagnosis [10]	boErr: Fehlerbit für Fehler 10 iDiagNo: Diagnosenummer für Fehler 10 diInfo1: Zusatzinfo 1 für Fehler 10 iAdr: Geräteadresse*	-

* iAdr: Geräteadresse
Eine ACC-Bus Steuerung zeigt die Geräteadresse bei einem Slavefehler an, ansonsten 0 (lokaler Fehler). EtherCAT immer 0.

Beispiel: Die Diagnosemeldung wird in lokale Variablen kopiert.



4.3.2.16 Visualisierung

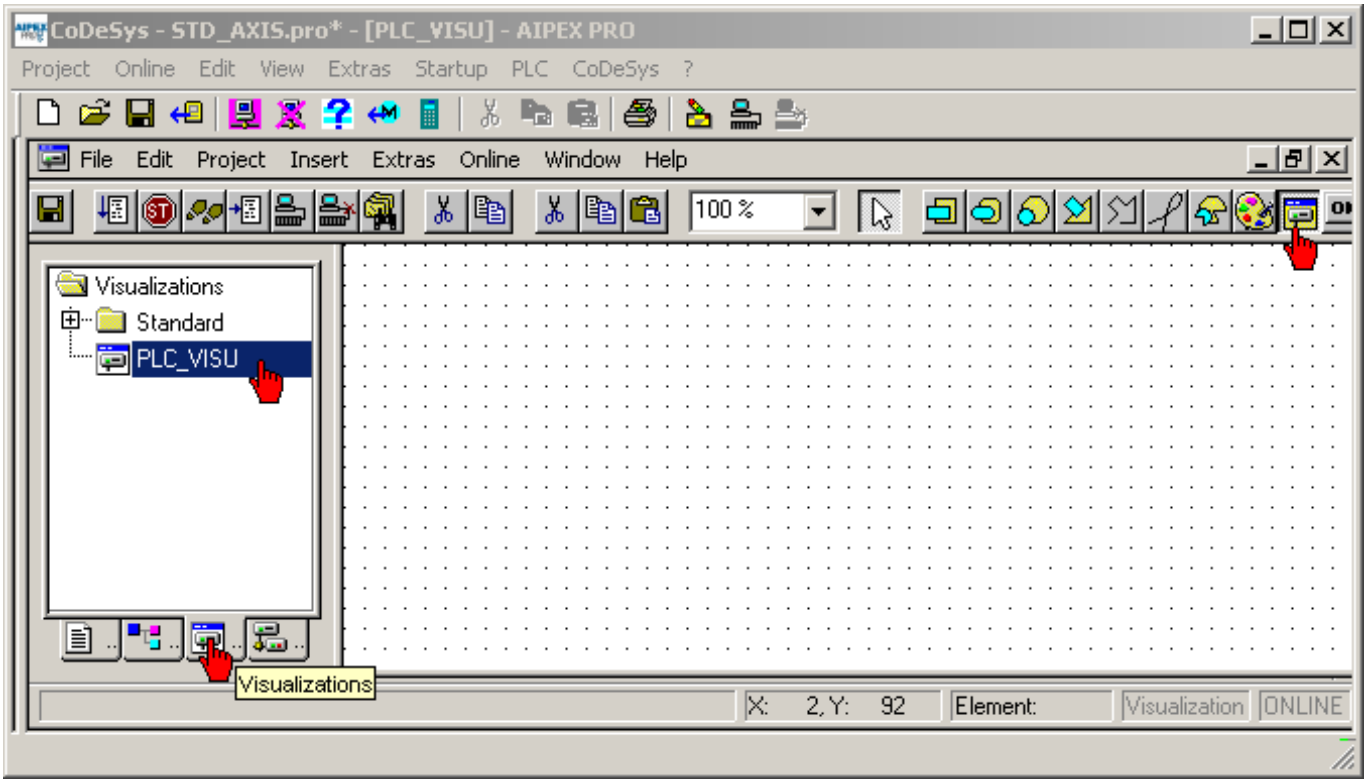
CODESYS enthält einen integrierten Visualisierungs-Editor. Mit dem Visualisierungs-Editor werden grafische Oberflächen erzeugt mit denen der Benutzer sein PLC Programm steuern und beobachten kann.

Für AMK-Funktionsblöcke gibt es vorgefertigte Visualisierungselemente.

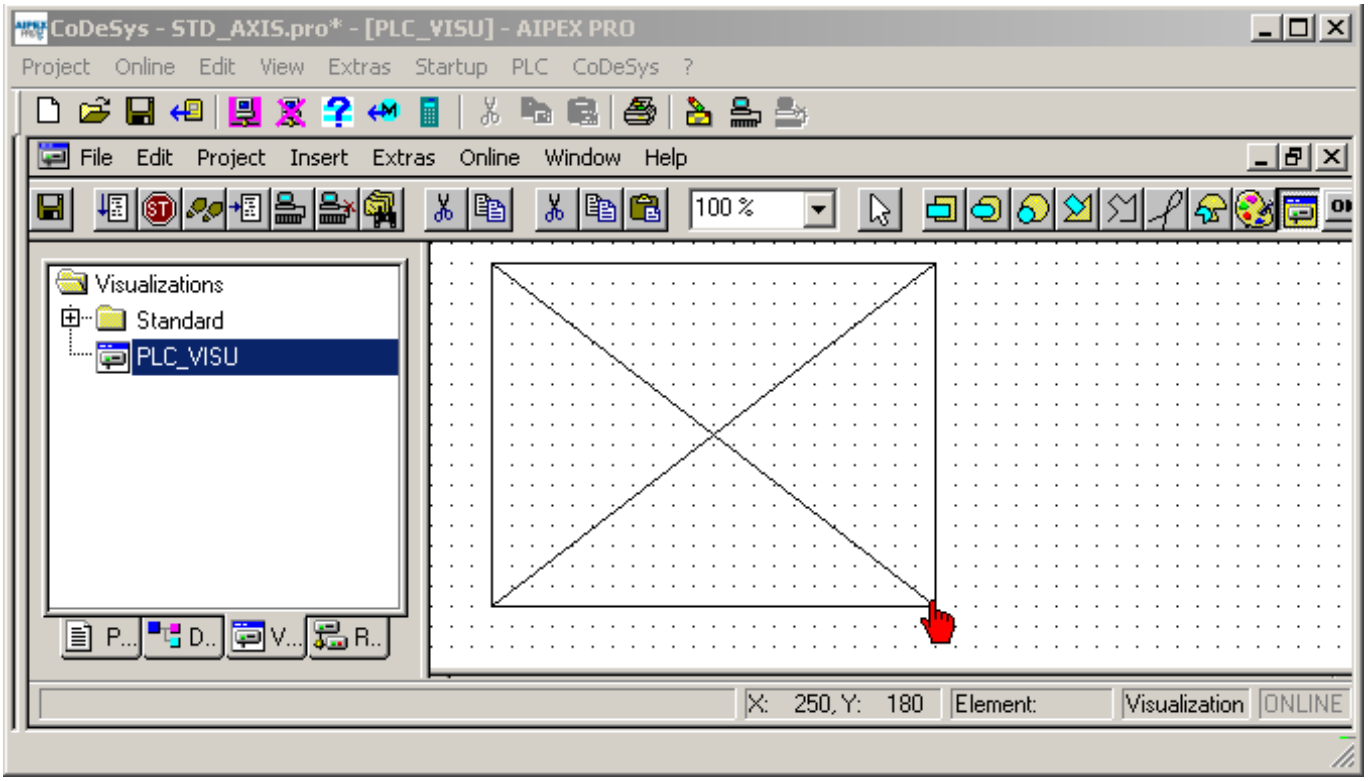
In den nächsten Schritten wird erklärt wie Sie eine Standard Achse mit einem Visualisierungselement verbinden.

'PLC_VISU' ist die Startseite und darf nicht umbenannt werden.

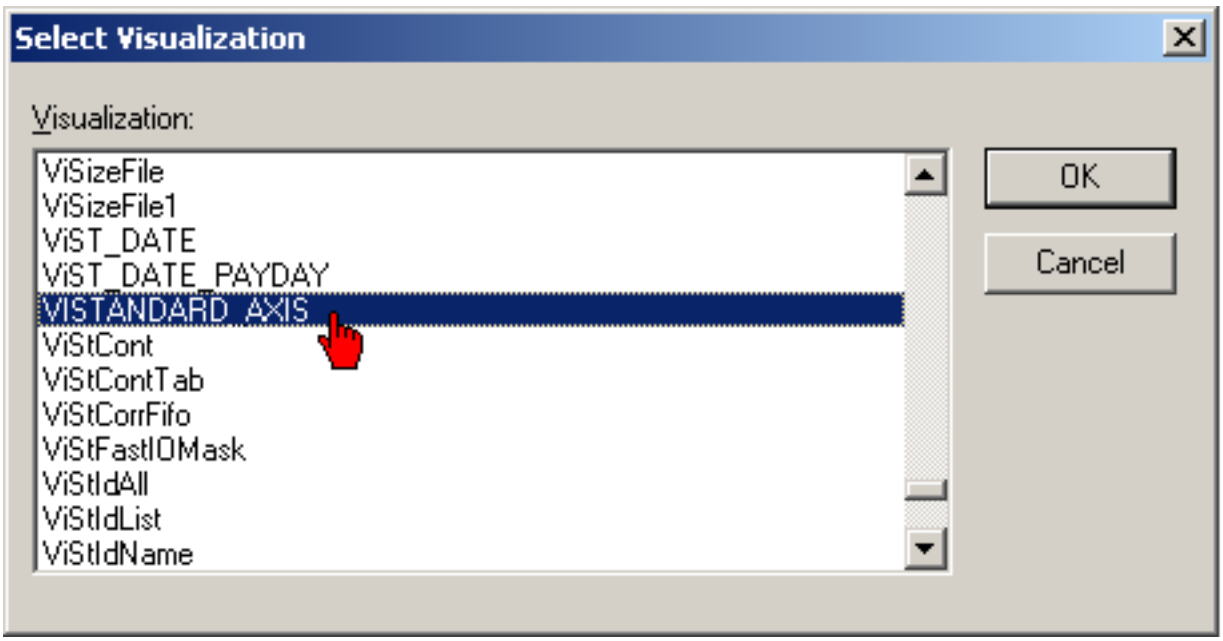
Wechseln Sie auf den Reiter **'Visualizations'**. Anschließend Doppelklick auf **'PLC_VISU'** um das Editorfenster zu öffnen.
Wählen Sie im Menü den Taster **'Visualizations'** an.



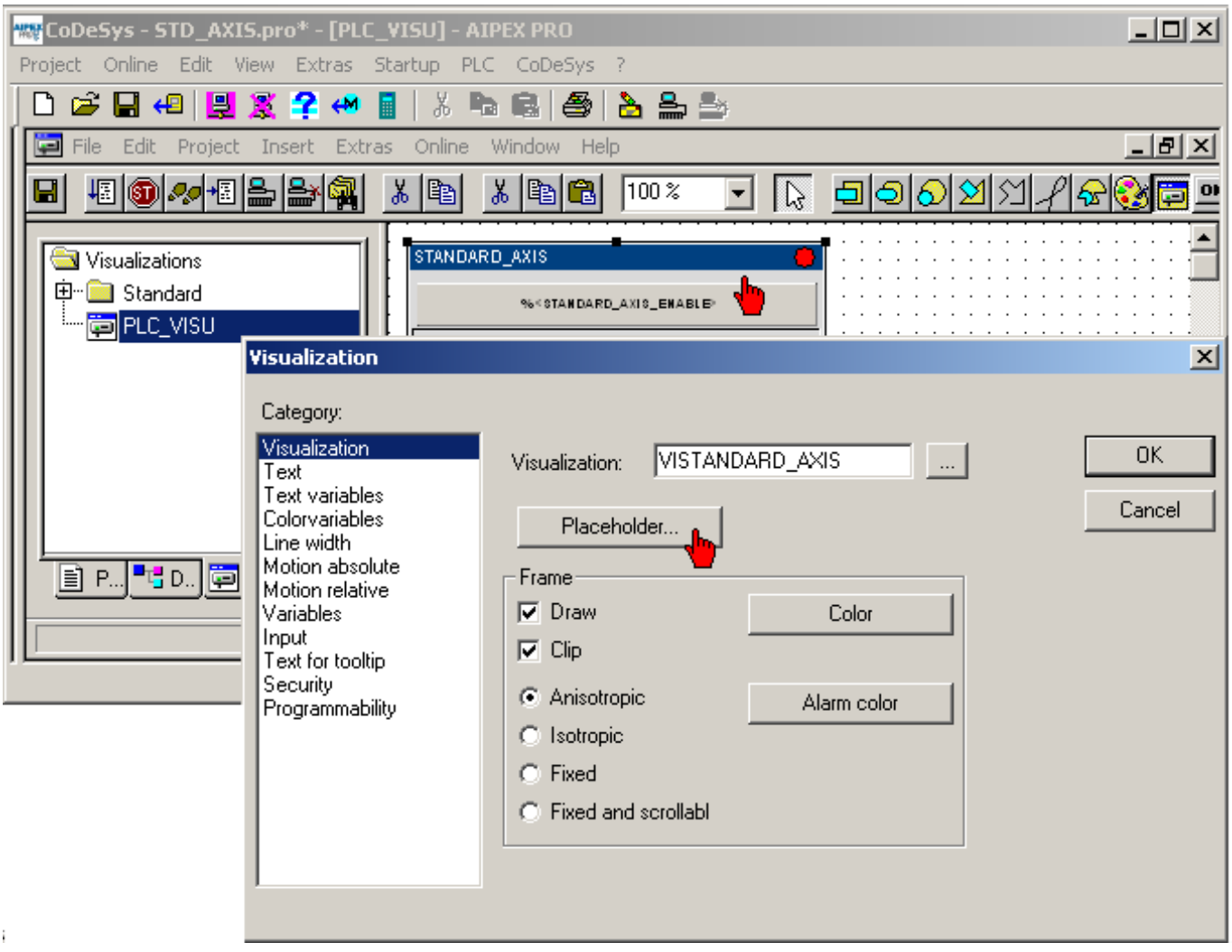
Mit gedrückter linker Maustaste können Sie an einer beliebigen Stelle ein Visualisierungselement aufziehen. Die genaue Größe lässt sich nachträglich anpassen.



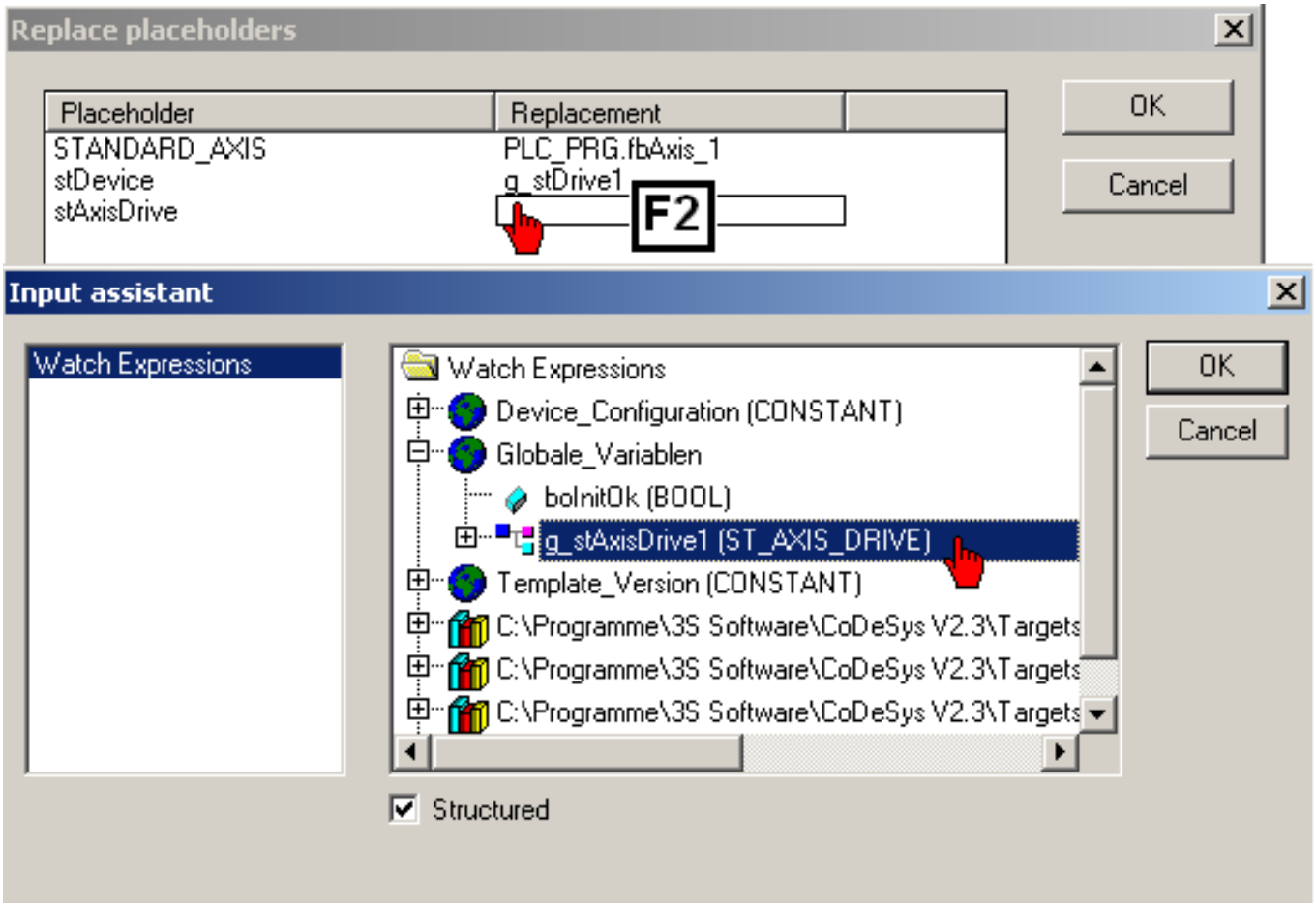
Wählen Sie im Dialogfenster das Element **'VISTANDARD_AXIS'** aus.



Klicken Sie dazu das Visualisierungselement doppelt an um das Dialogfeld 'Visualization' zu öffnen.
Drücken Sie den Taster **'Placeholder'.** Dort definieren Sie die Verbindungen zum Programmcode.

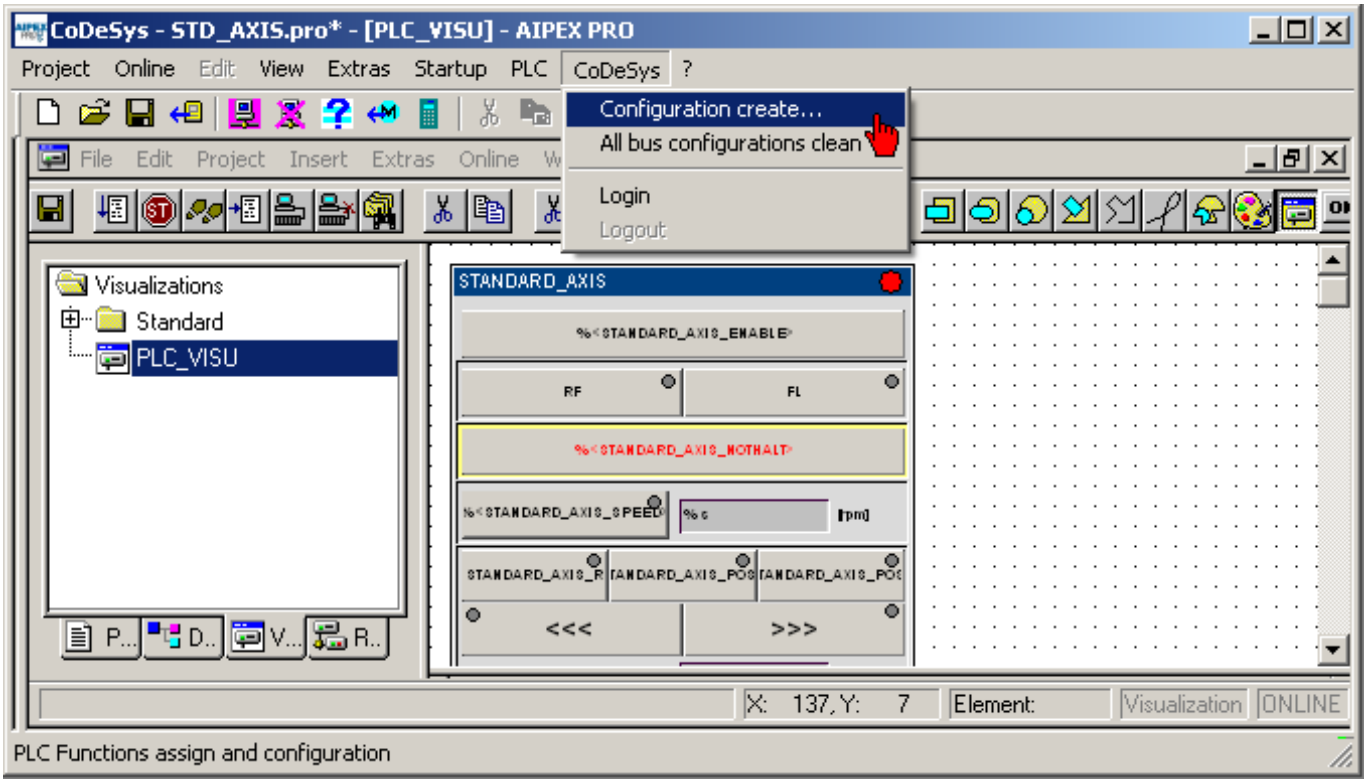


STANDARD_AXIS: Verbindung zum Funktionsbaustein im 'PLC_PRG'.
stDevice: Verbindung zur Achsstruktur
stAxisDrive: Verbindung zum symbolischen Gerätenamen

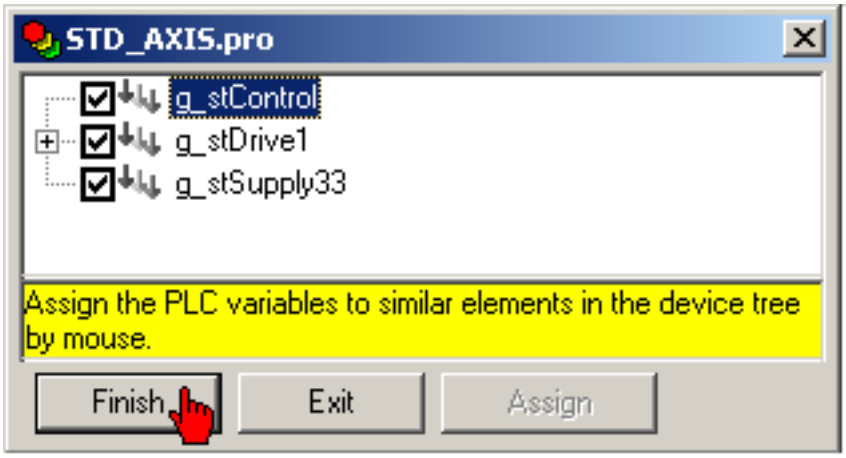


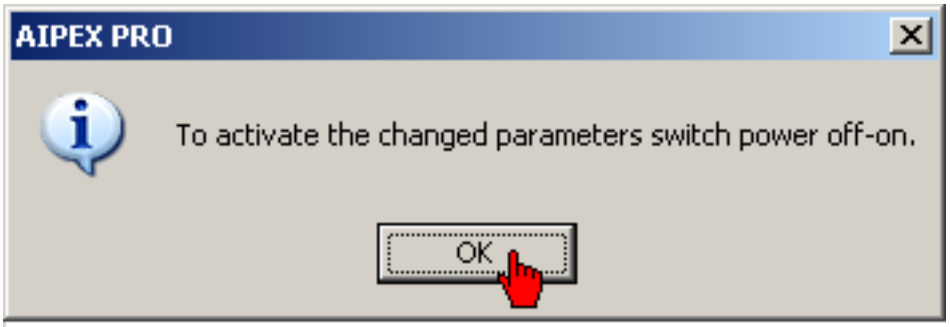
4.3.2.17 Programm übertragen

Bevor das Programm übertragen werden kann muss es fehlerfrei übersetzt und die Nachrichtenkonfiguration erstellt werden.
Starten Sie diesen Vorgang unter **'CODESYS' 'Configuration create'**.

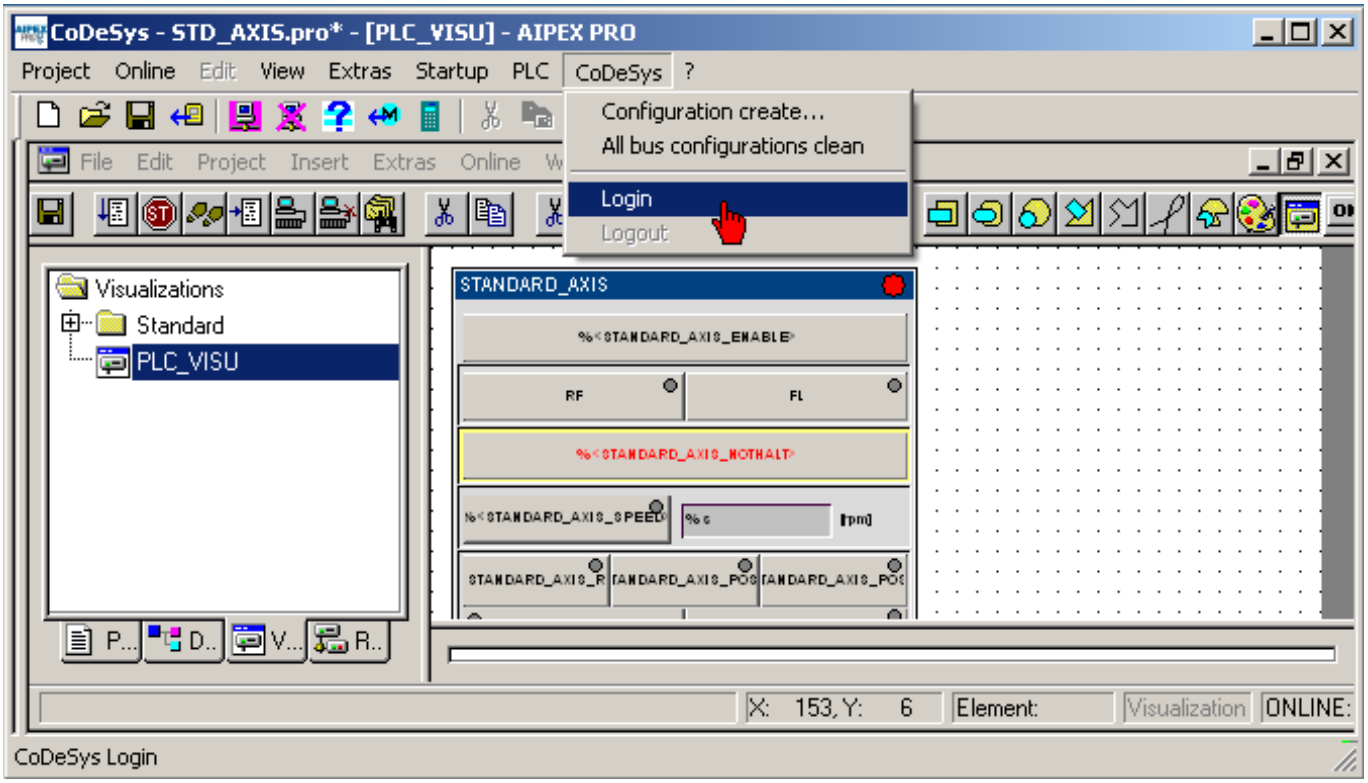


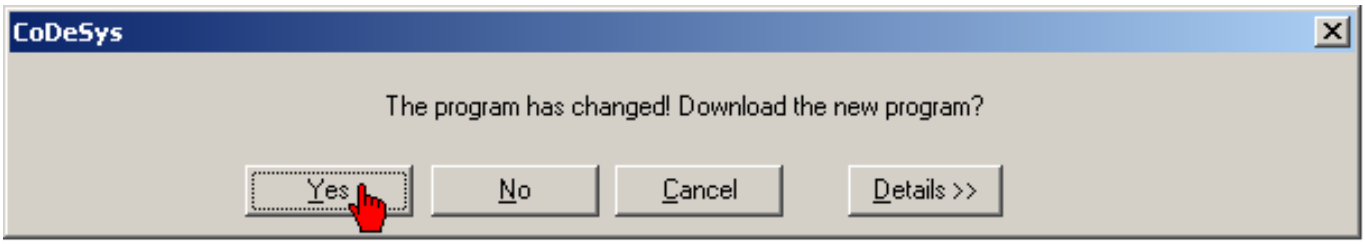
Drücken Sie die Taste **Finish**. AIPEX PRO erstellt automatisch die Buskonfiguration und ändert die relevanten Parameter.
(Automatische Parametrierung: Siehe AIPEX PRO **'Menue' 'Extras' 'Configuration create'**)



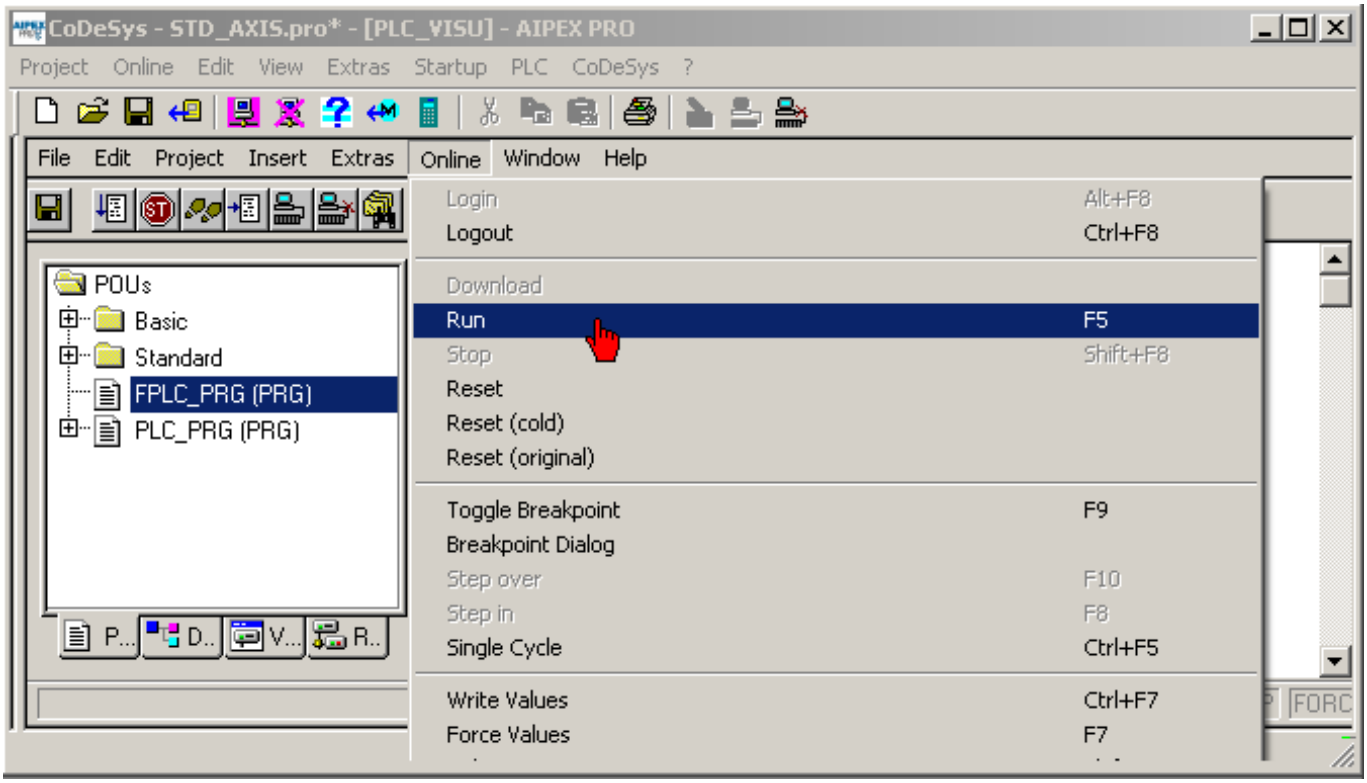


AIPEX PRO muss online sein bevor Sie sich unter **'CODESYS' 'Login'** an der PLC Einloggen können.

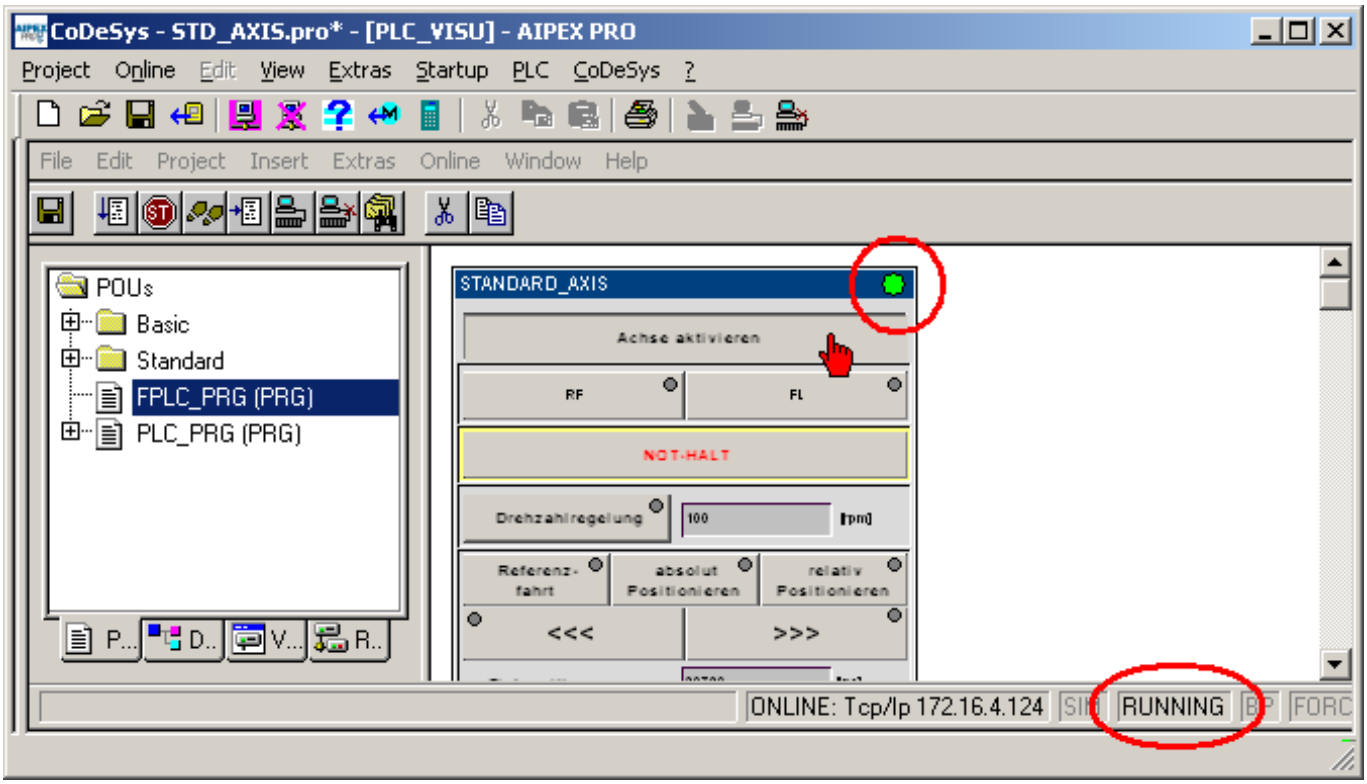




Die PLC wird unter **'Online' 'Run'** gestartet.



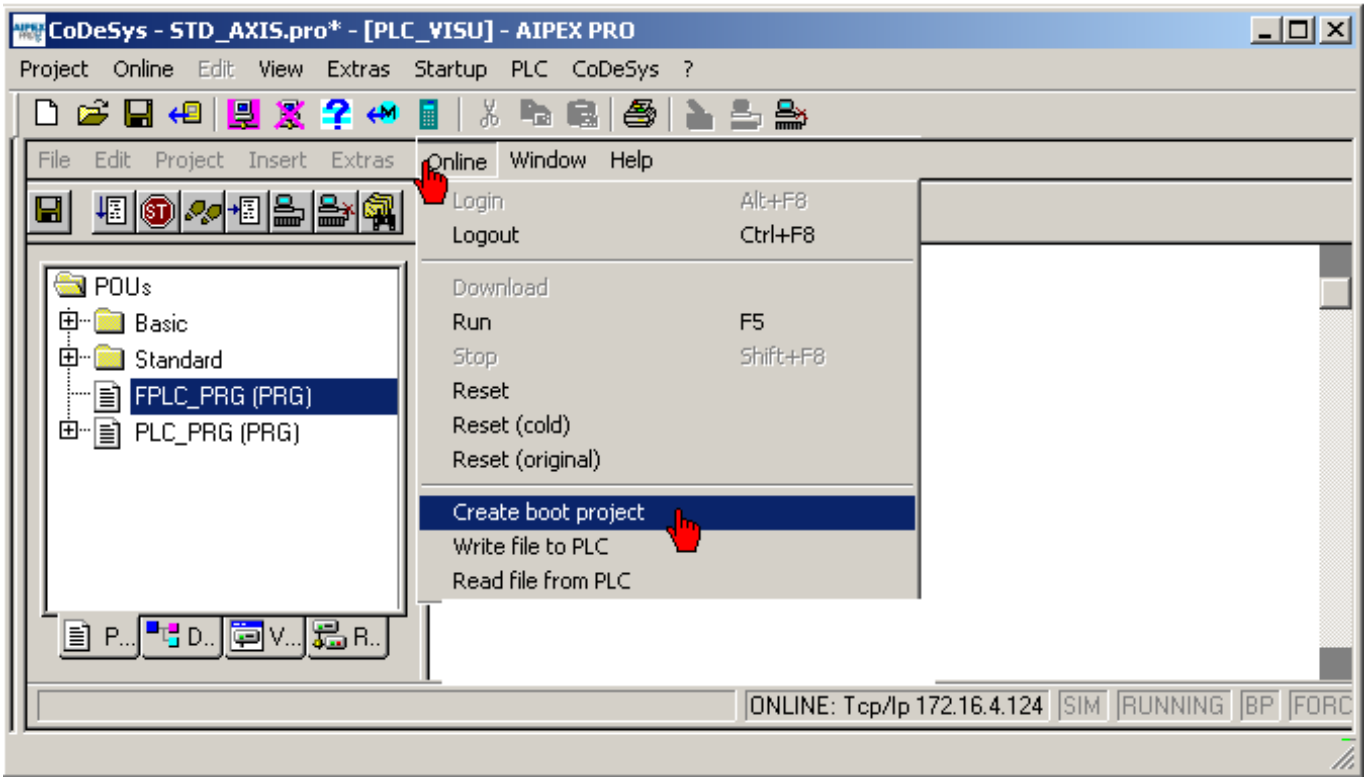
Sie können jetzt Ihr Antriebssystem mit der von Ihnen erstellten Visualisierung Steuern.



Ein- und Ausschaltdiagramm AMKASYN KE/KW bzw. Kombination AMKASYN KE/KW + AMKASMART iX/iDT5 siehe: Dokumentation Erstinbetriebnahme (AMK Teilenr. 204539) Kapitel: Ein- und Ausschaltdiagramm.

Ein- und Ausschaltdiagramm AMKASMART iC/iX/iDT5 siehe: Dokumentation Erstinbetriebnahme (AMK Teilenr. 204737) Kapitel: Ein- und Ausschaltdiagramm iX und iDT5 bzw. Ein- und Ausschaltdiagramm iC.

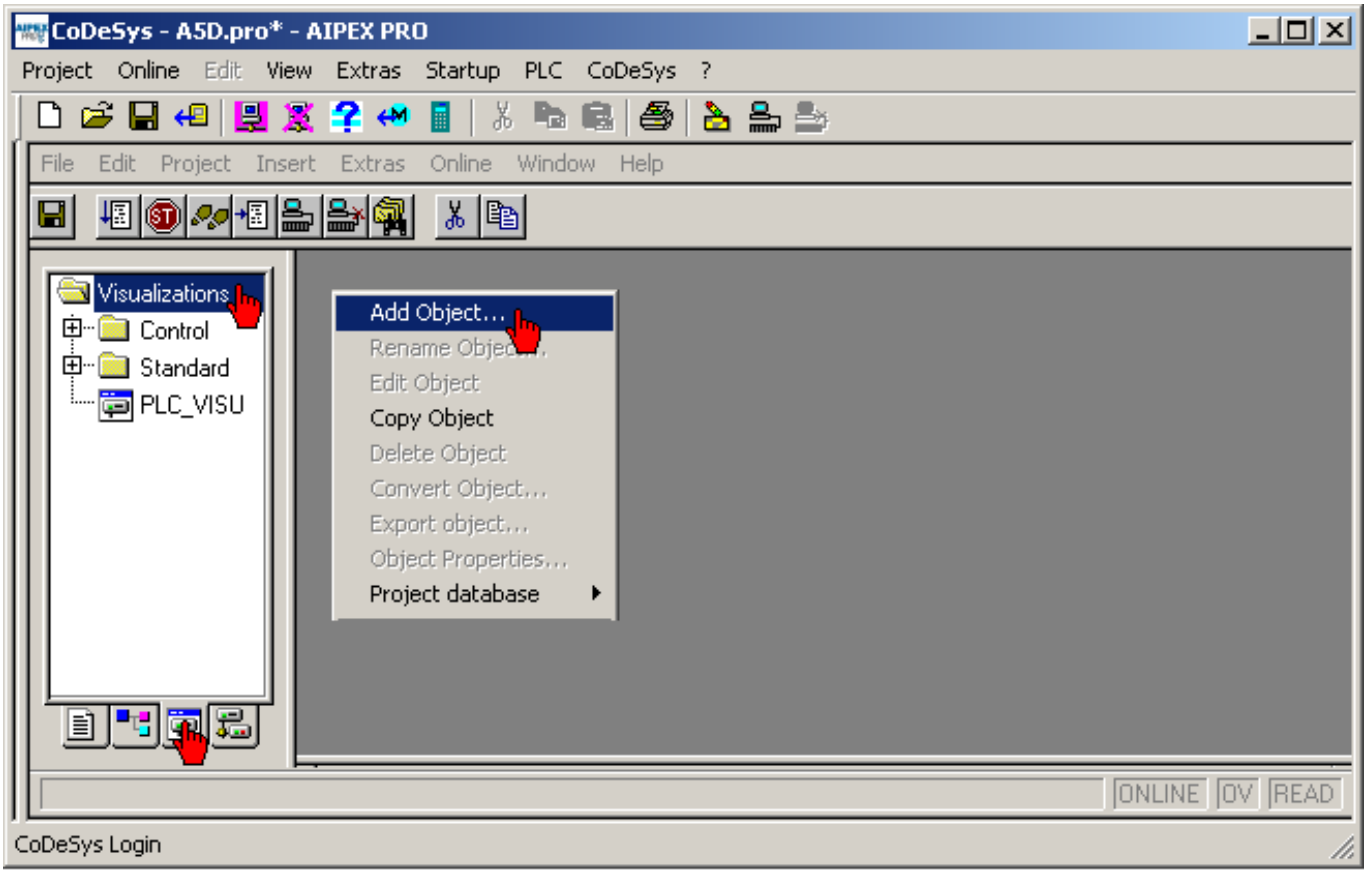
Soll das PLC Programm remanent gespeichert werden, müssen Sie ein Bootprojekt erstellen. PLC Editor - **'Online' 'Create boot project'**.



4.3.2.18 Schnell zum Ziel

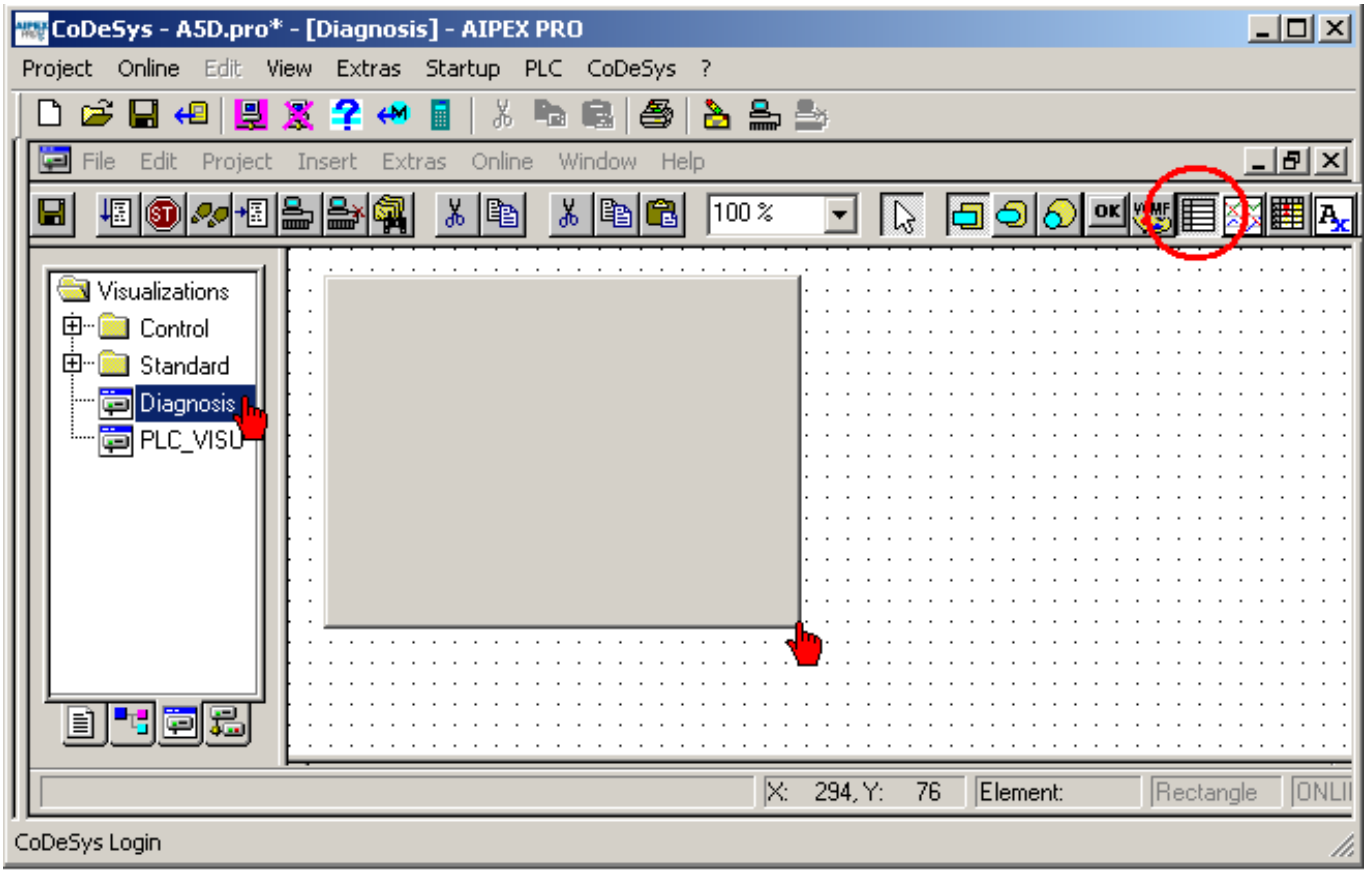
4.3.2.18.1 Diagnose Array (arstDiagnosis) visualisieren

Wechseln Sie auf den Reiter **'Visualizations'**. Fügen Sie mit der RMT eine neue Seite ein.

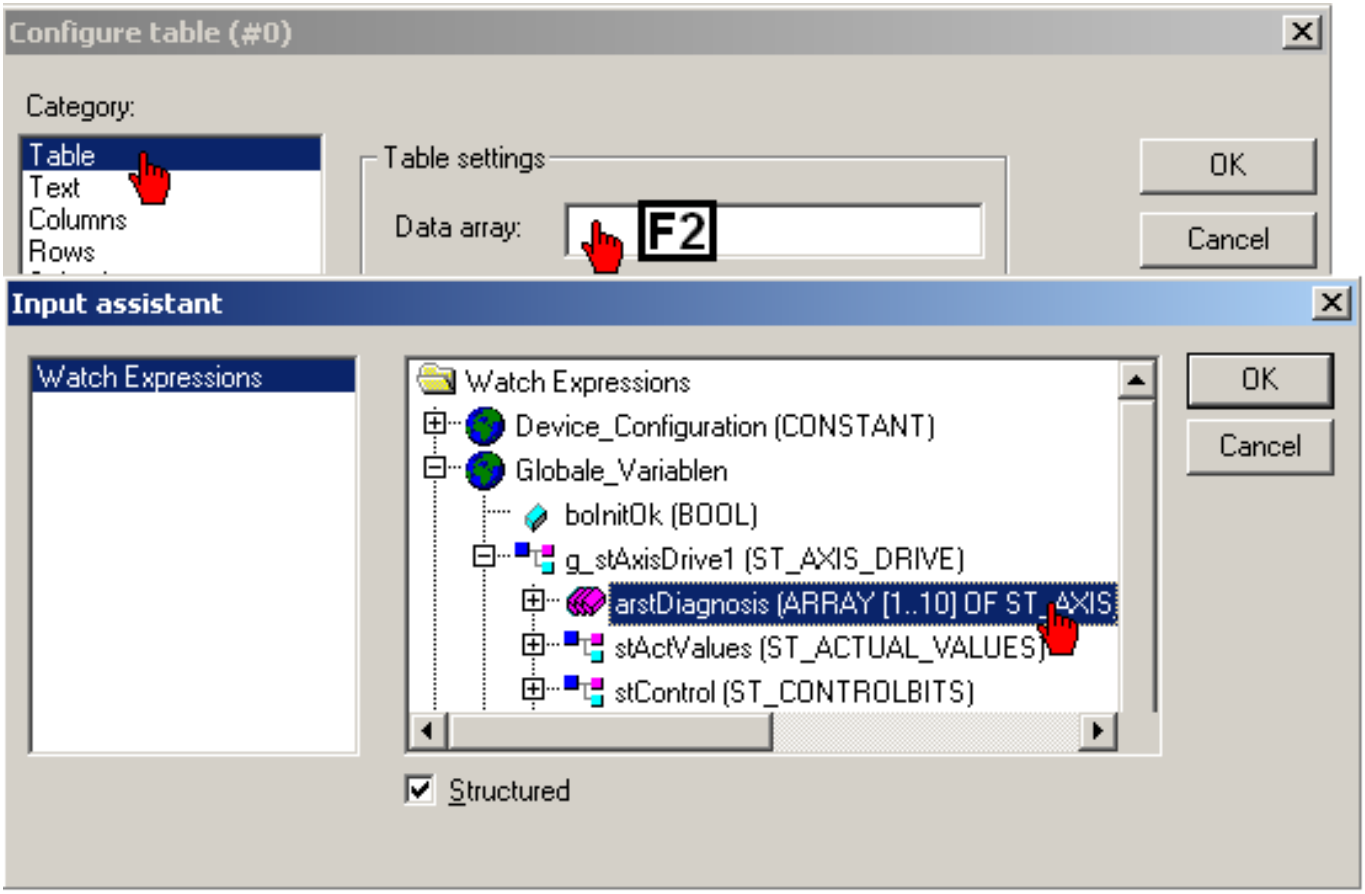


Wählen Sie im Menü den Taster **'Table'** an.

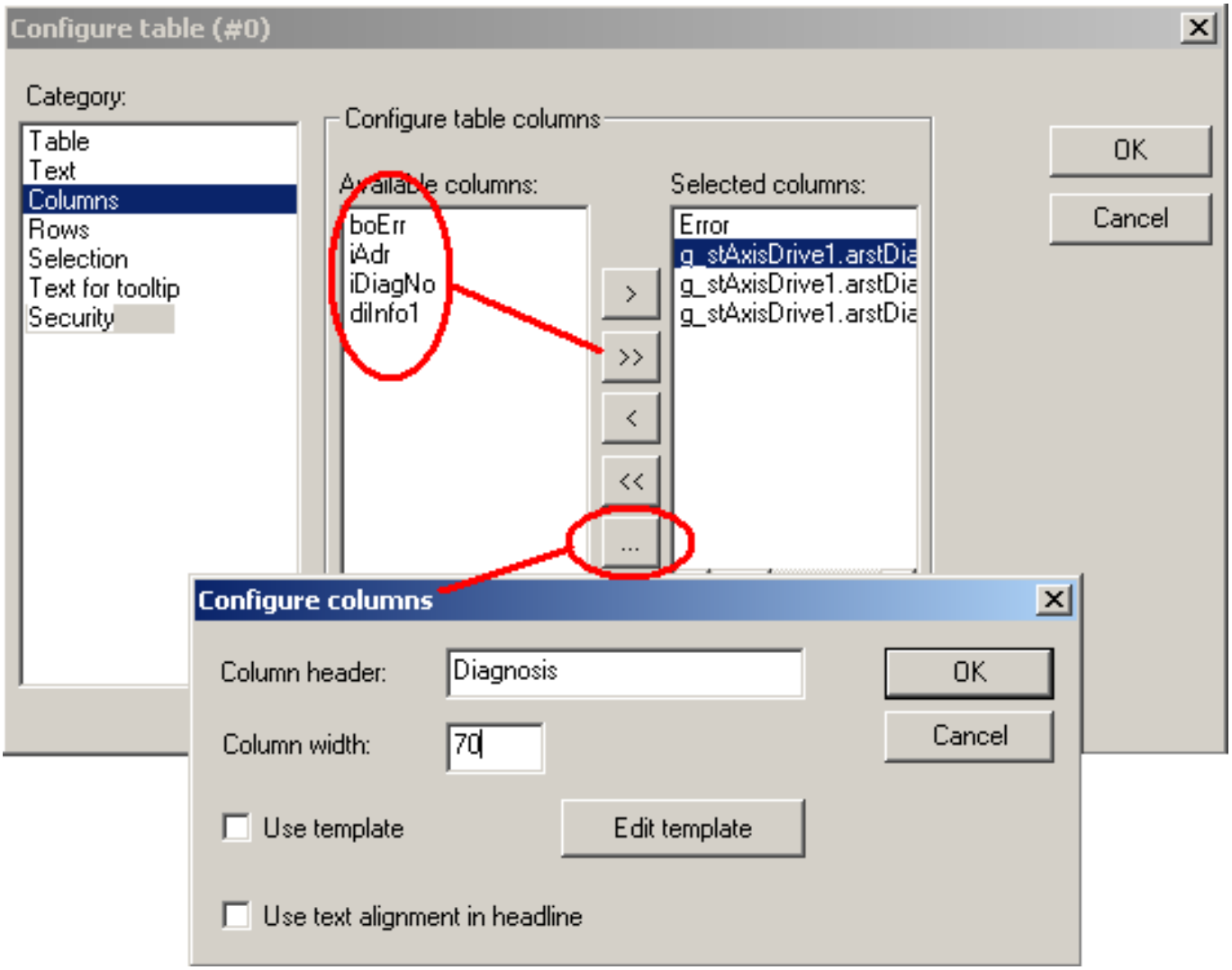
Mit gedrückter linker Maustaste können Sie an einer beliebigen Stelle die Tabelle aufziehen. Die genaue Größe lässt sich nachträglich anpassen.



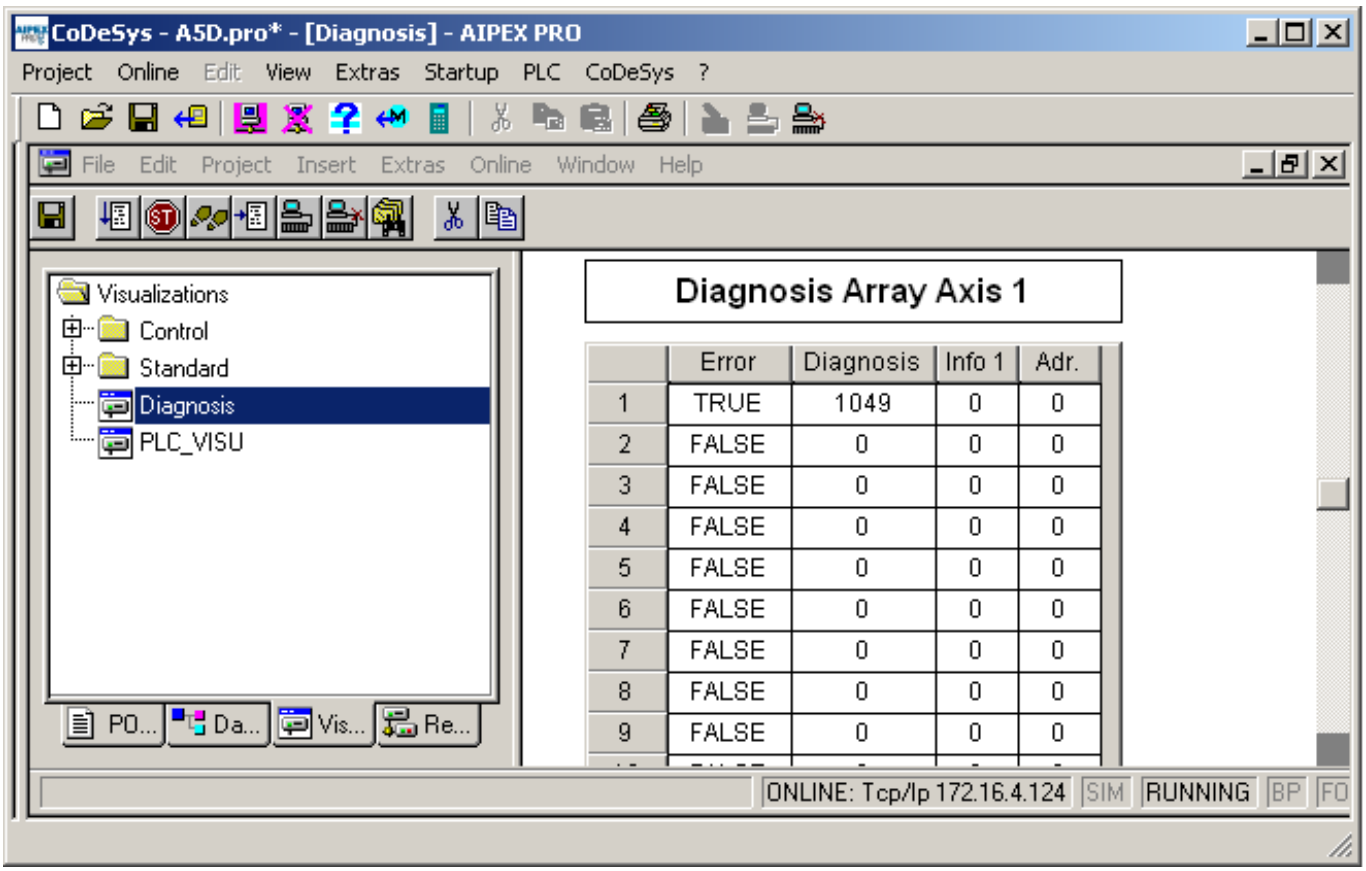
Über **'Data array'** und **'F2'** definieren Sie die Verbindung. Das Diagnose Array 'arstDiagnosis' finden Sie in der Struktur 'ST_AXIS_DRIVE'.



Übernehmen Sie unter **'Columns'** die Variablen die Sie in der Tabelle anzeigen wollen.
Wählen Sie anschließend die Variablen einzeln an, um Tabellenüberschrift und breite anzupassen.

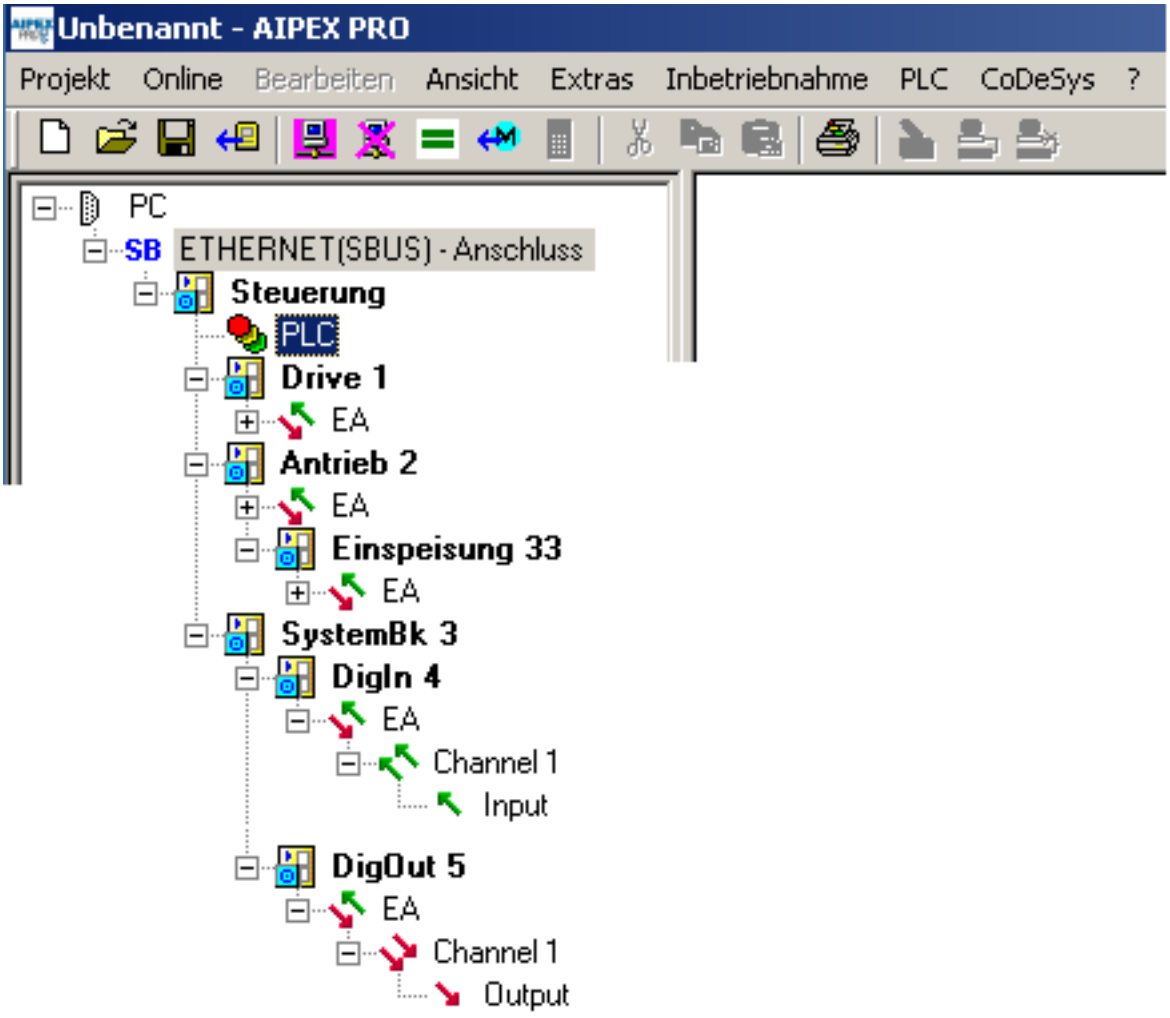


Diagnosenummer 1: Hauptmeldung
Diagnosenummer 2 – 10: Eventuell Folgefehler


CoDeSys - A5D.pro* - [Diagnosis] - AIPEX PRO
Project Online Edit View Extras Startup PLC CoDeSys ?
File Edit Project Insert Extras Online Window Help
Visualizations
Control
Standard
Diagnosis
PLC_VISU
PO...
Da...
Vis...
Re...
Diagnosis Array Axis 1
Error Diagnosis Info 1 Adr.
1 TRUE 1049 0 0
2 FALSE 0 0 0
3 FALSE 0 0 0
4 FALSE 0 0 0
5 FALSE 0 0 0
6 FALSE 0 0 0
7 FALSE 0 0 0
8 FALSE 0 0 0
9 FALSE 0 0 0
ONLINE: Tcp/Ip 172.16.4.124
SIM
RUNNING
BP
FO

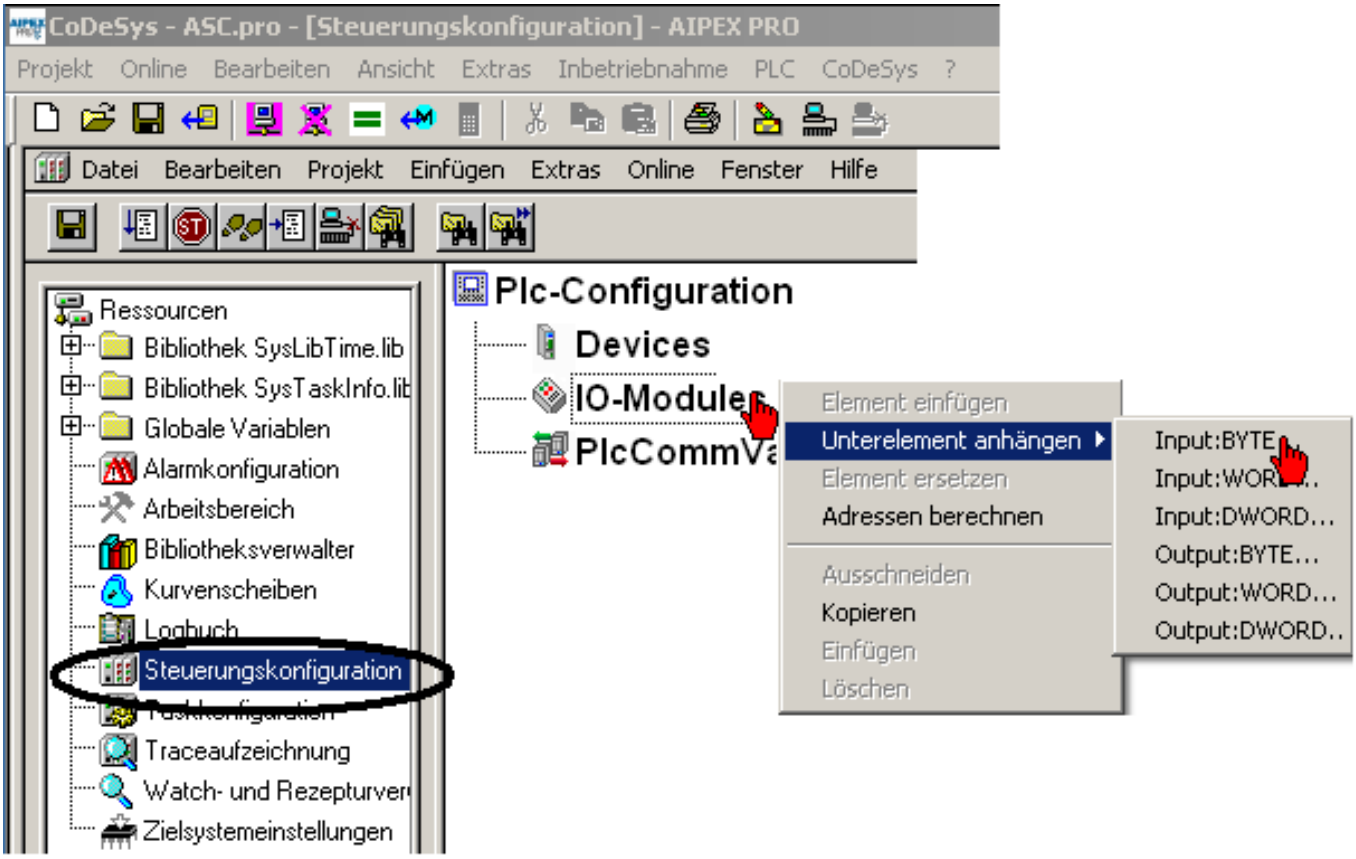
4.3.2.18.2 Externe asynchrone EA Klemme

Einbinden einer externen EA-Klemme. Das Beispiel zeigt eine EtherCAT EA Klemme (symbolischer Name SystemBK3) mit 8 Eingängen und 8 Ausgängen. Die EtherCAT Klemmen wird von AIPEX PRO automatisch erkannt.

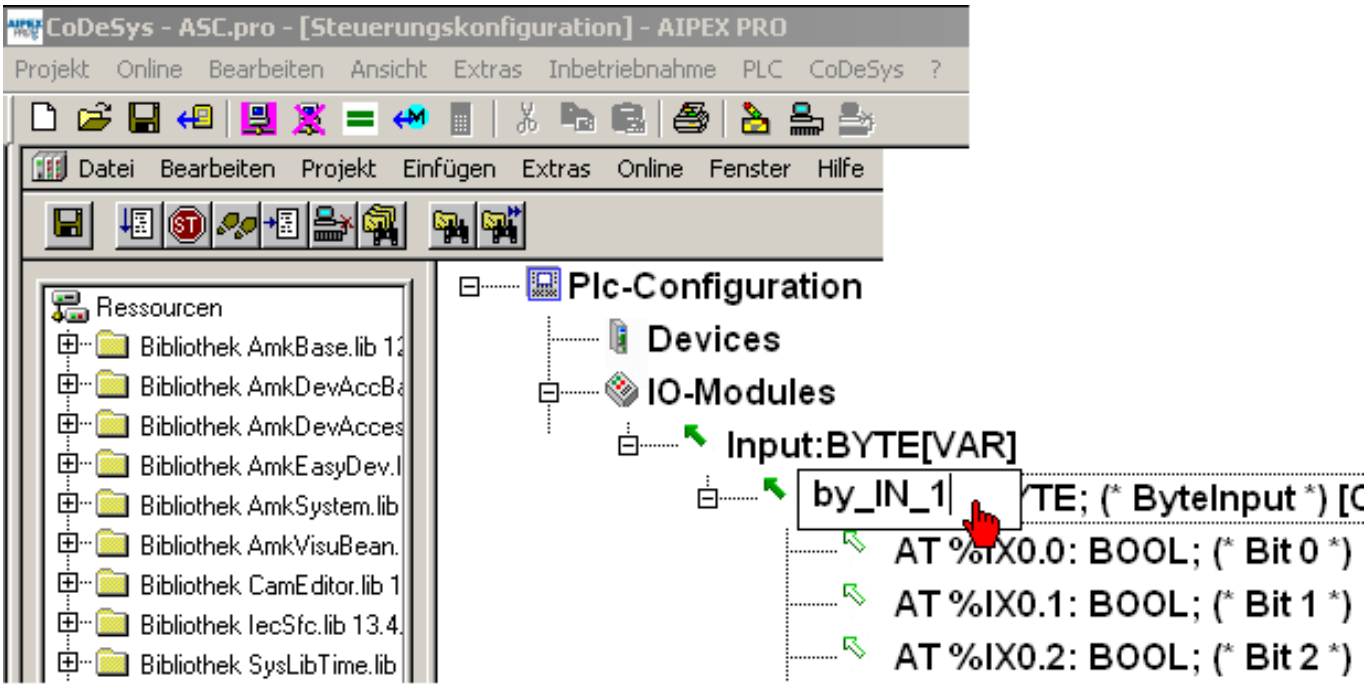


Für jeden Ein und Ausgangsblock muss in CoDeSys eine Variable angelegt werden.
Öffnen Sie dazu in ihrem CoDeSys Projekt unter **'Ressourcen'** die **'Steuerungskonfiguration'**.
Klicken Sie mit der rechten Maustaste auf die IO-Modules neue Variablen einzufügen.

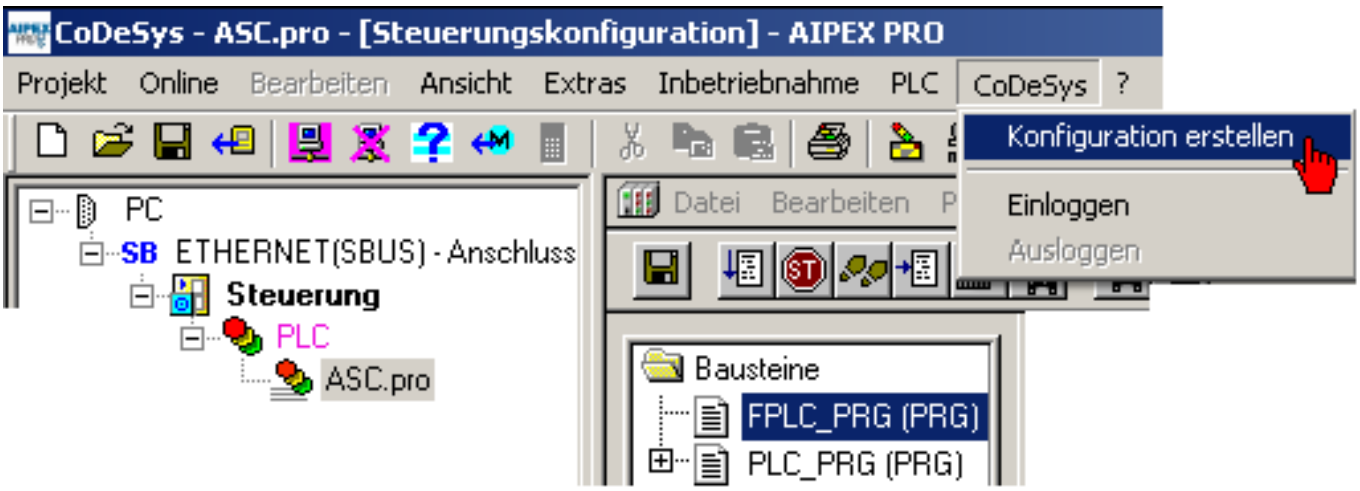
Input: Daten werden von den Antrieben in die PLC eingelesen
Output: Daten werden von der PLC auf die Antriebe geschrieben



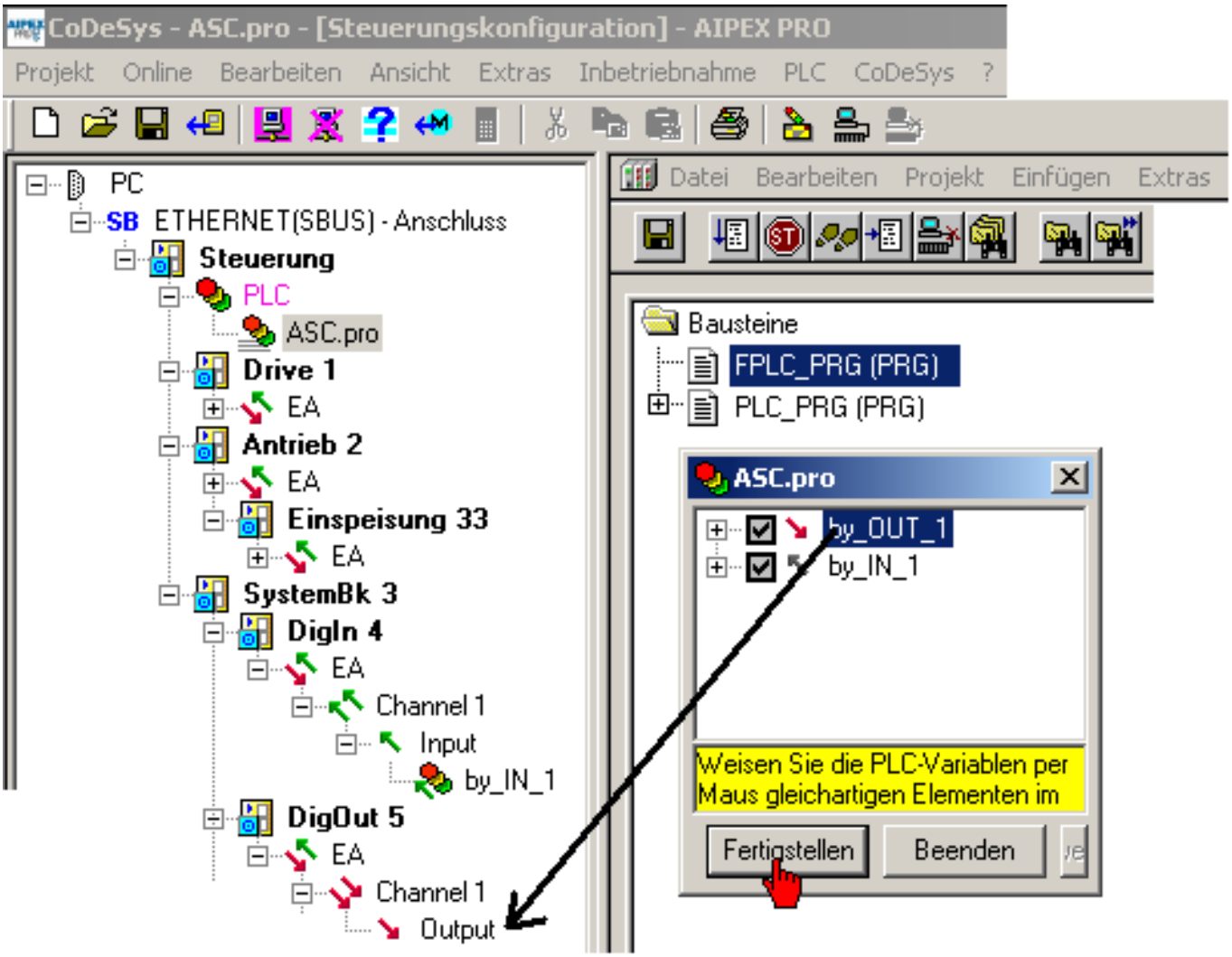
Für jedes physikalisch vorhandene Modul muss eine symbolische Variable angelegt werden.



Starten Sie anschließend die automatische Nachrichtenkonfigurationserstellung.

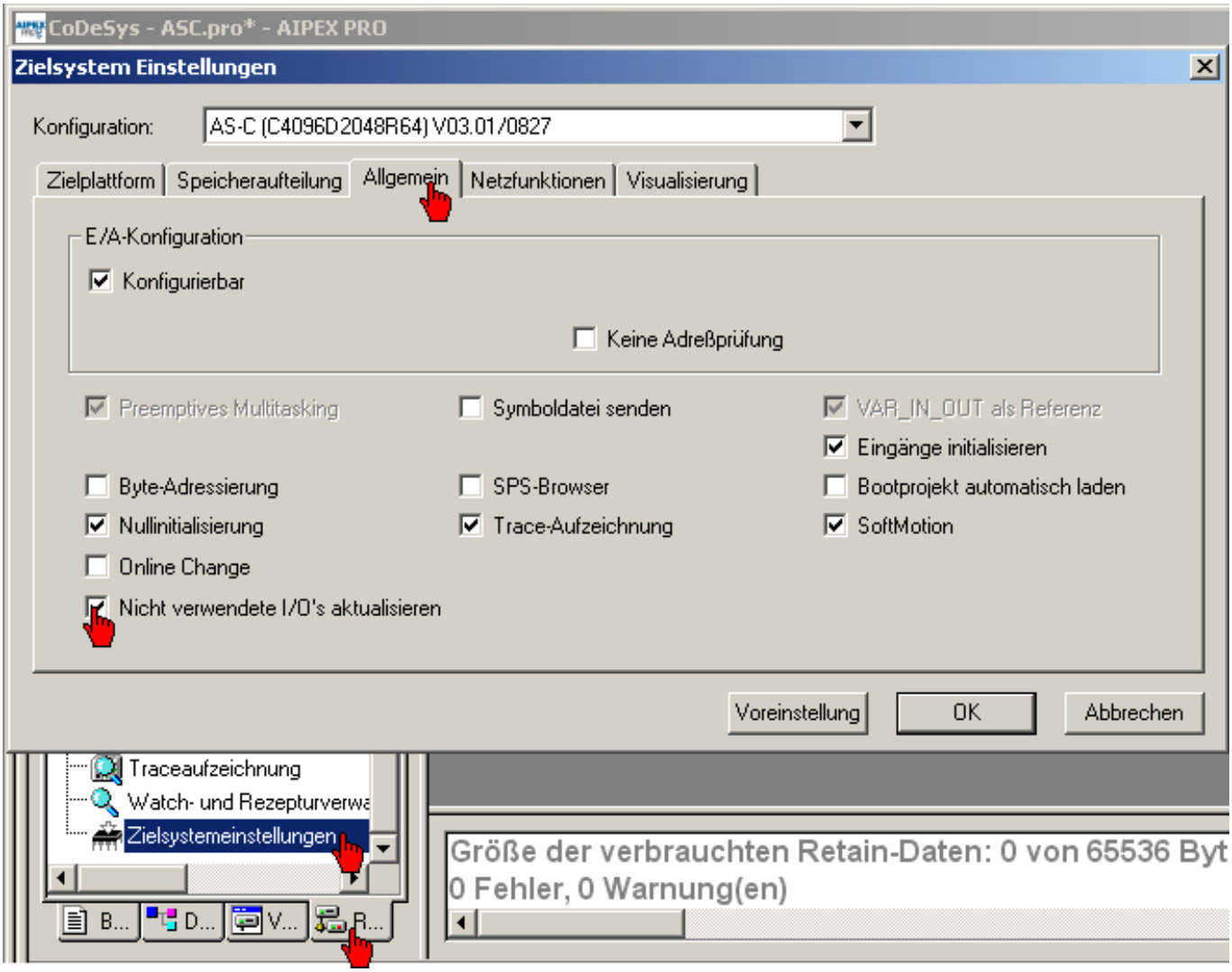


Ordnen Sie die symbolischen Variablen den physikalisch Vorhandenen Ein und Ausgängen zu.
Nach dem erstellen der Nachrichtenkonfiguration muss das System neu gestartet werden.

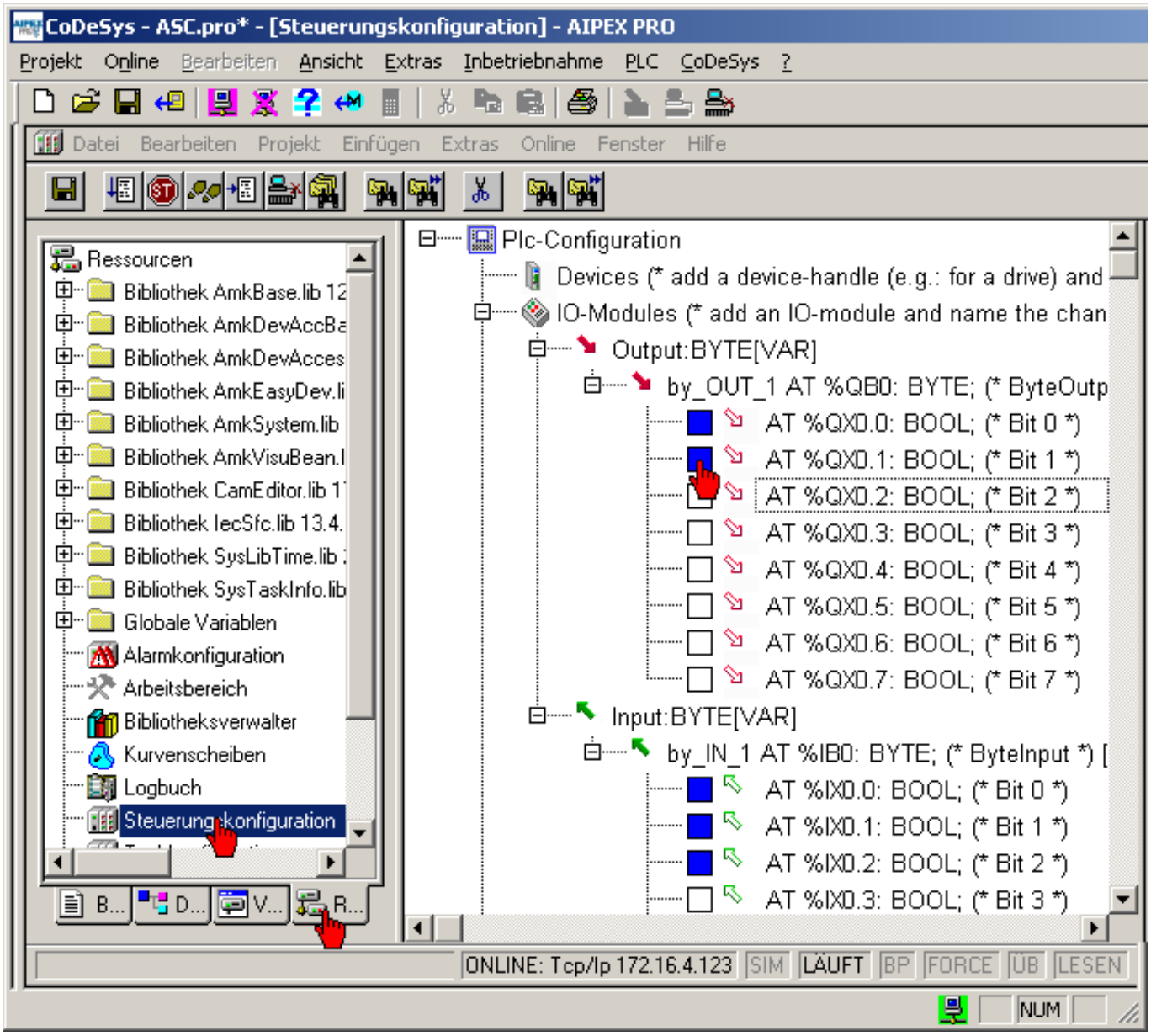


Funktionstest

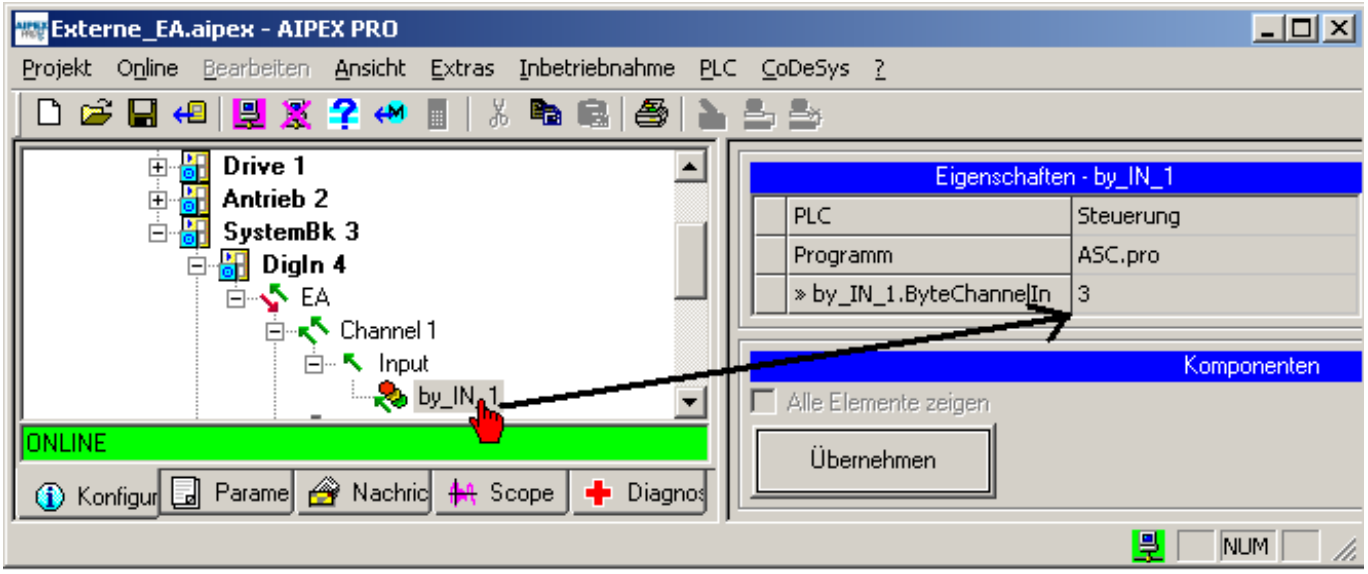
Aktivieren Sie die Checkbox **Nicht verwendete I/O's aktualisieren** unter **Ressourcen** -> **Zielsystem Einstellungen** -> **Allgemein**
oder weisen Sie jeder I/O ein symbolischer Namen zu.



Funktionstest: Voraussetzung: AIPEX PRO Eingeloggt, CoDeSys Eingeloggt, SPS läuft.
Output – Variablen können durch doppeltes anklicken gesetzt werden.

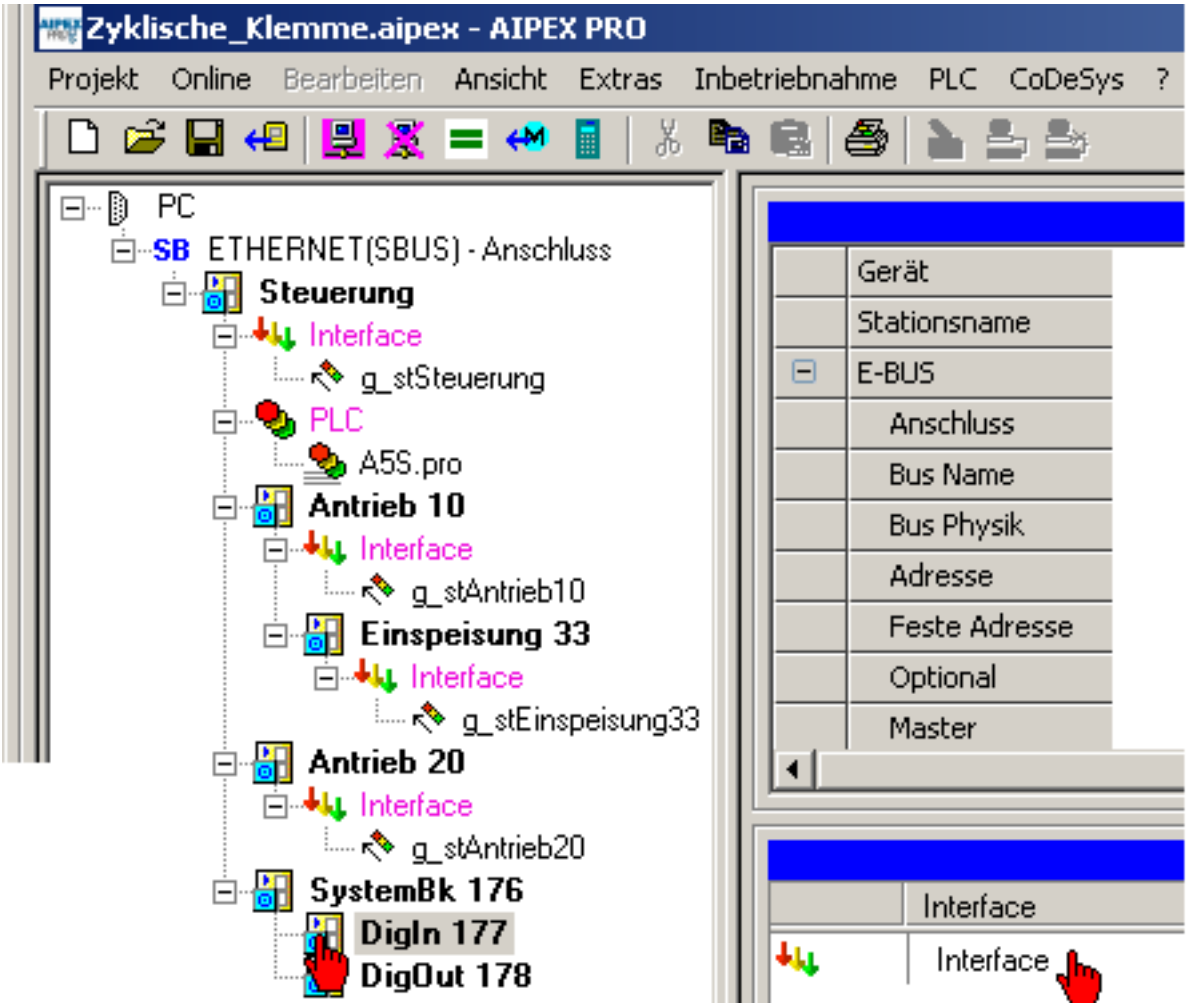


In der AIPEX PRO Oberfläche, unter Konfiguration wird der aktuelle Wert in dezimal angezeigt.

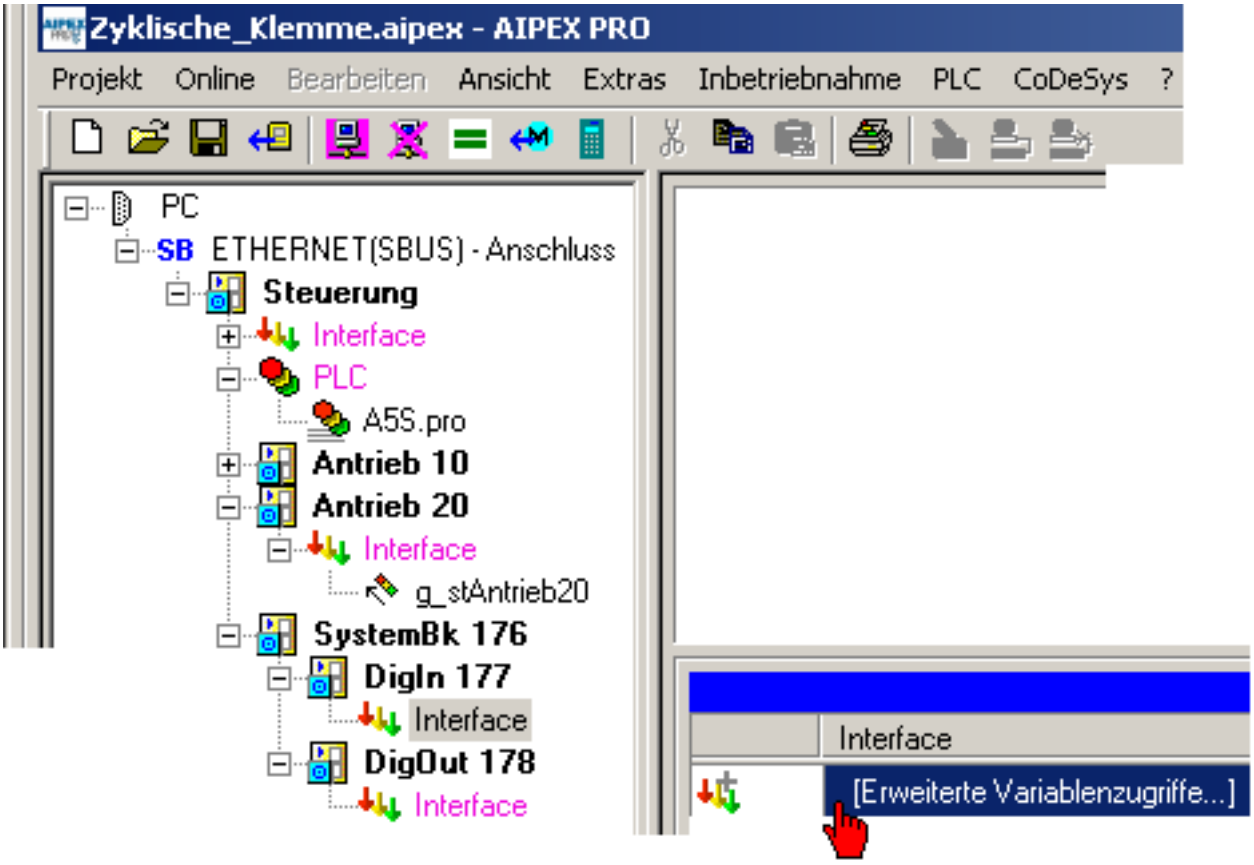


4.3.2.18.3 Externe zyklische EA Klemme

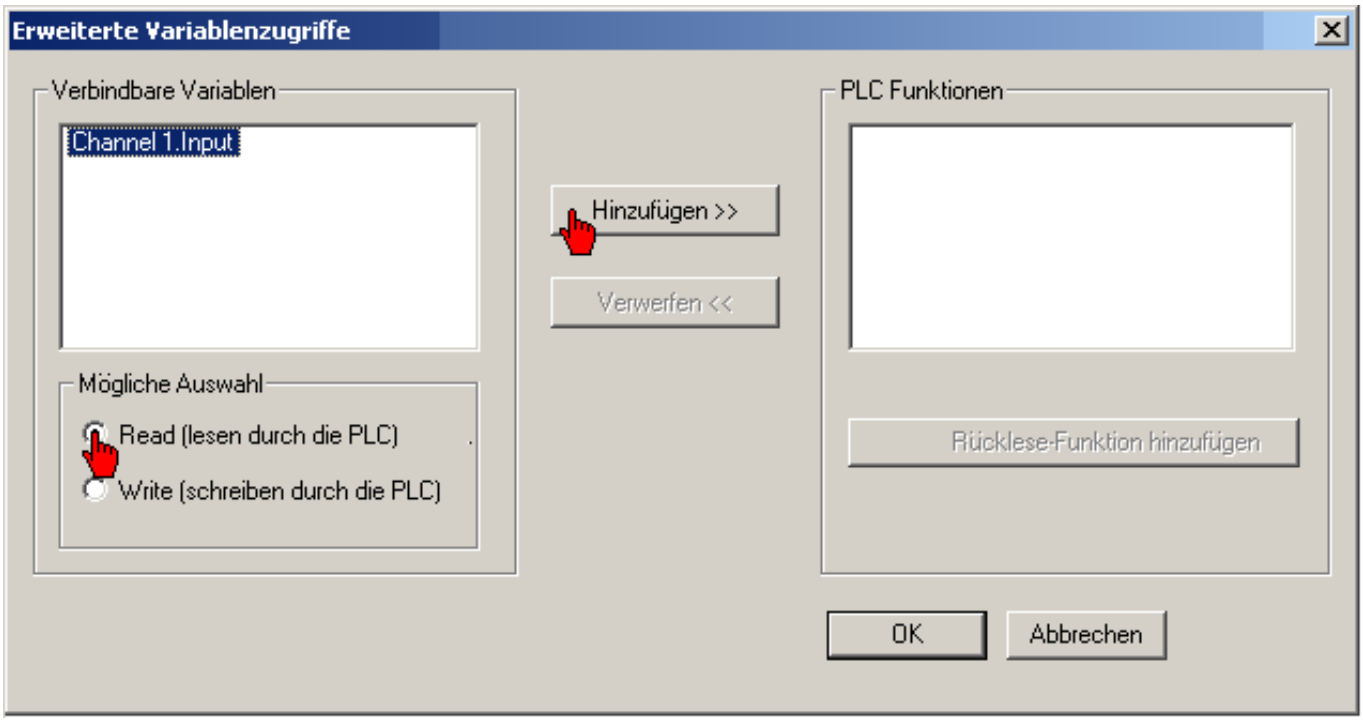
Das zyklische Lesen und Schreiben eines EA Abbildes findet über das Geräteinterface statt.
Legen Sie dazu an ihre Klemme ein Interface an.

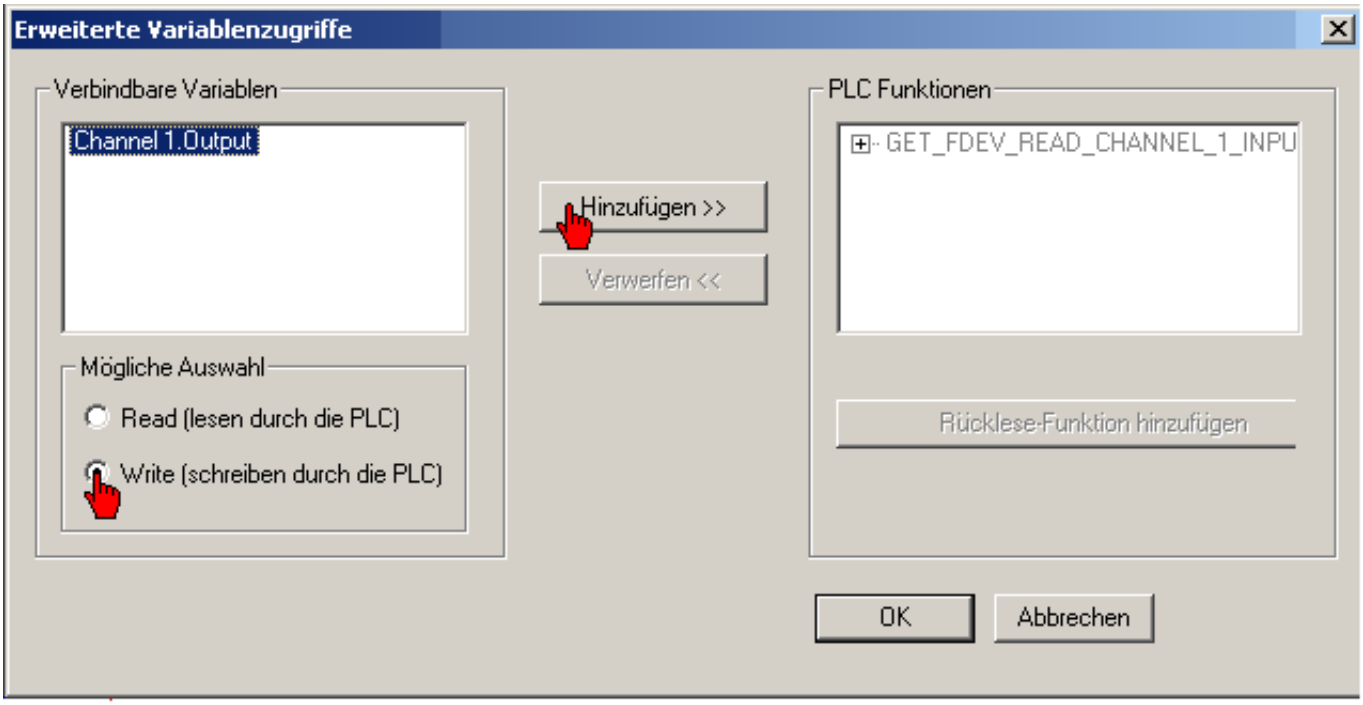


Verknüpfen Sie das Interface mit einem formalen Funktionsbaustein. Rufen Sie dazu den Menüpunkt **Erweiterte Variablenzugriffe** auf.

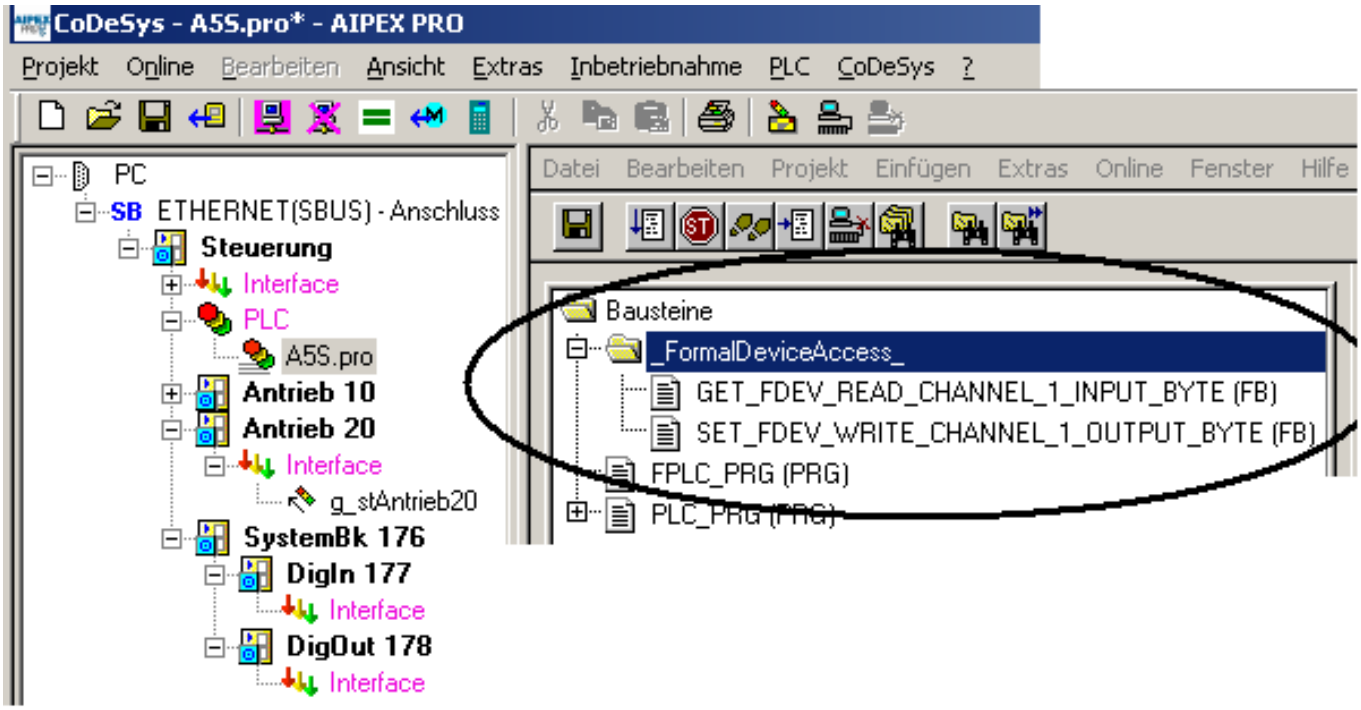


Wählen Sie bei Eingängen Read (lesen durch die PLC) an.
Wählen Sie bei Ausgängen Write (schreiben durch die PLC).
In der Variablenliste finden Sie alle verfügbaren Werte.
Die Variablenauswahl wird mit der Taste Hinzufügen übernommen

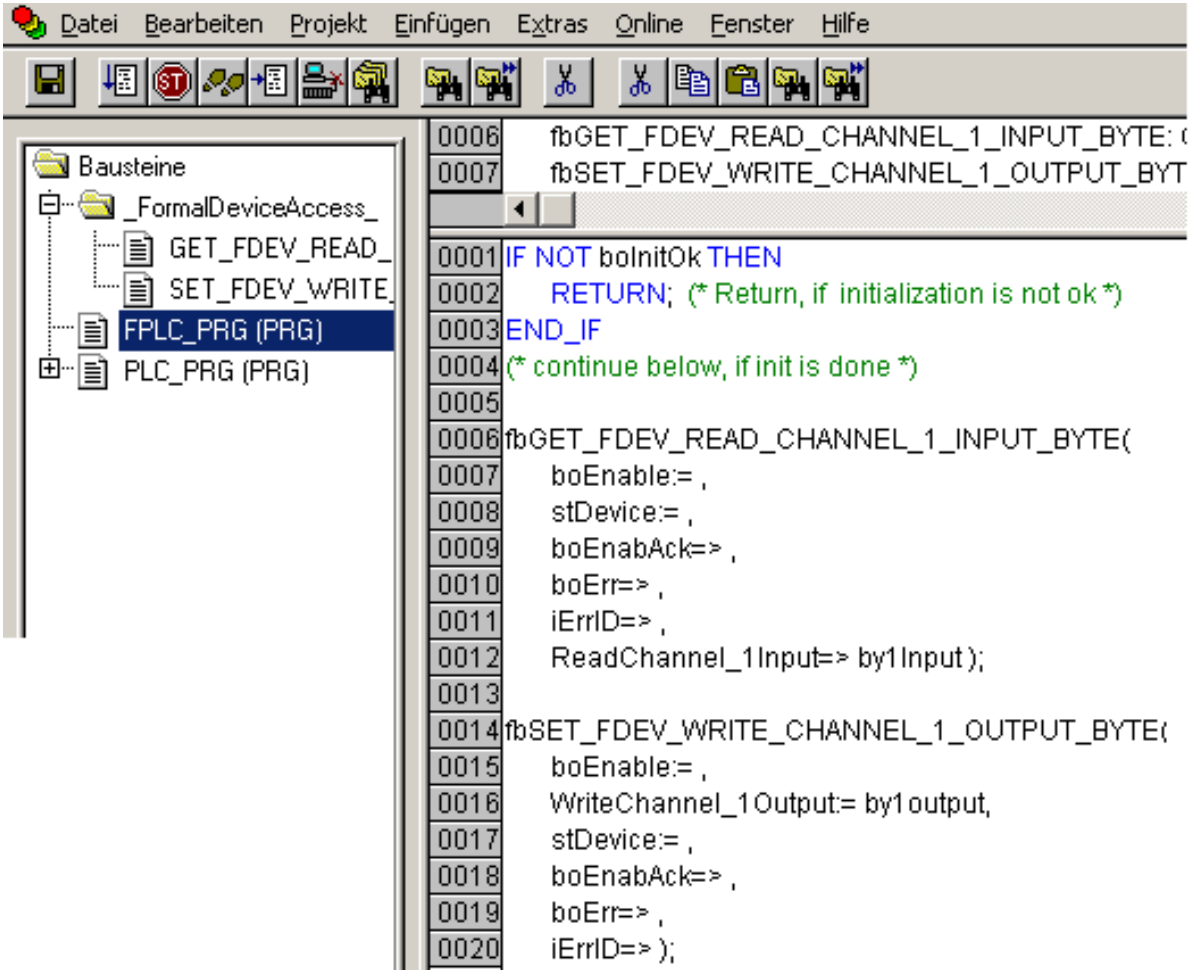




Öffnen Sie ihr CoDeSys Projekt.
Im Ordner **_FormalDeviceAccess_** finden Sie die neu angelegten Funktionsbausteine.

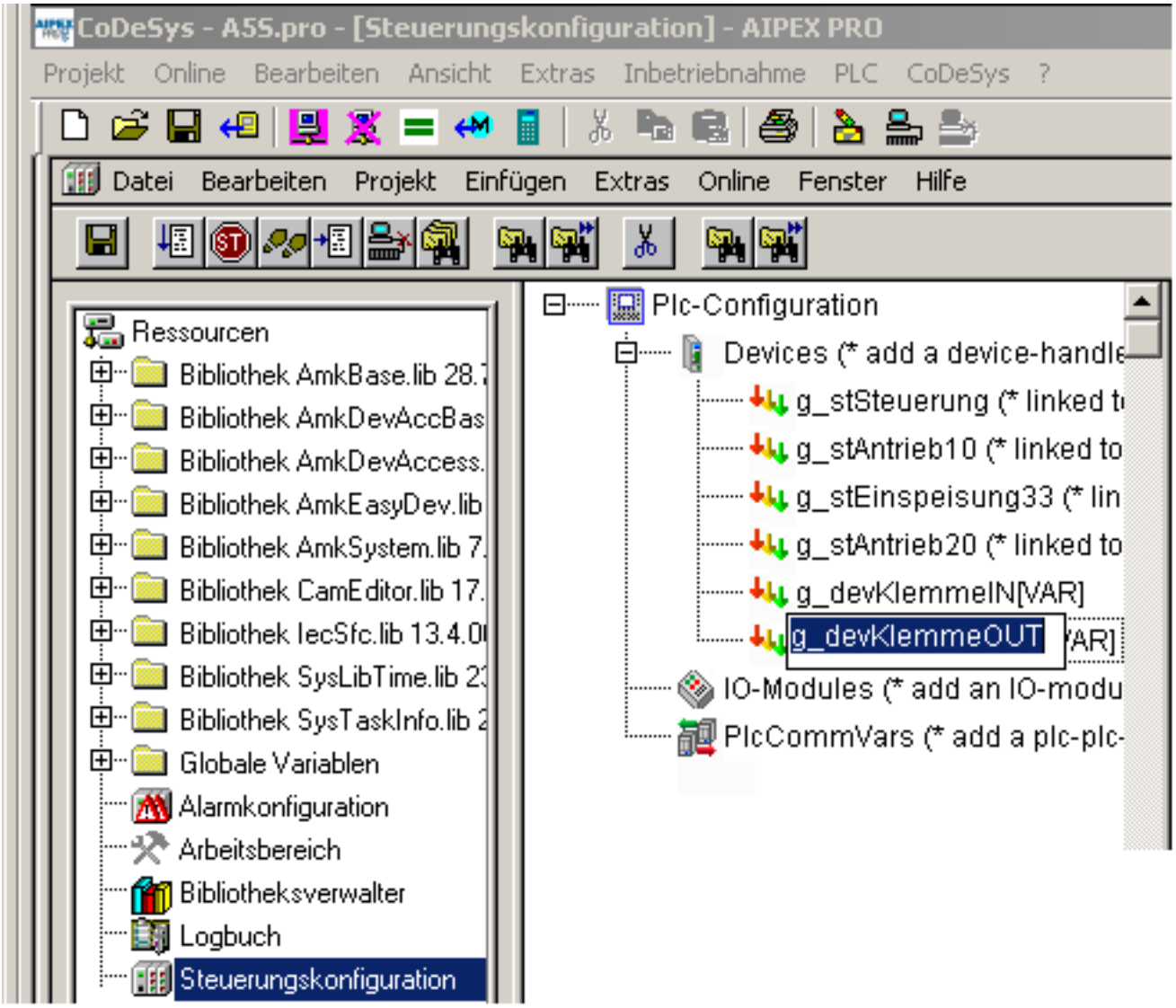


Fügen Sie die Funktionsbausteine in die Echtzeitebene FPLC_PRG ein.
Bilden Sie eine Instanz des Bausteines.

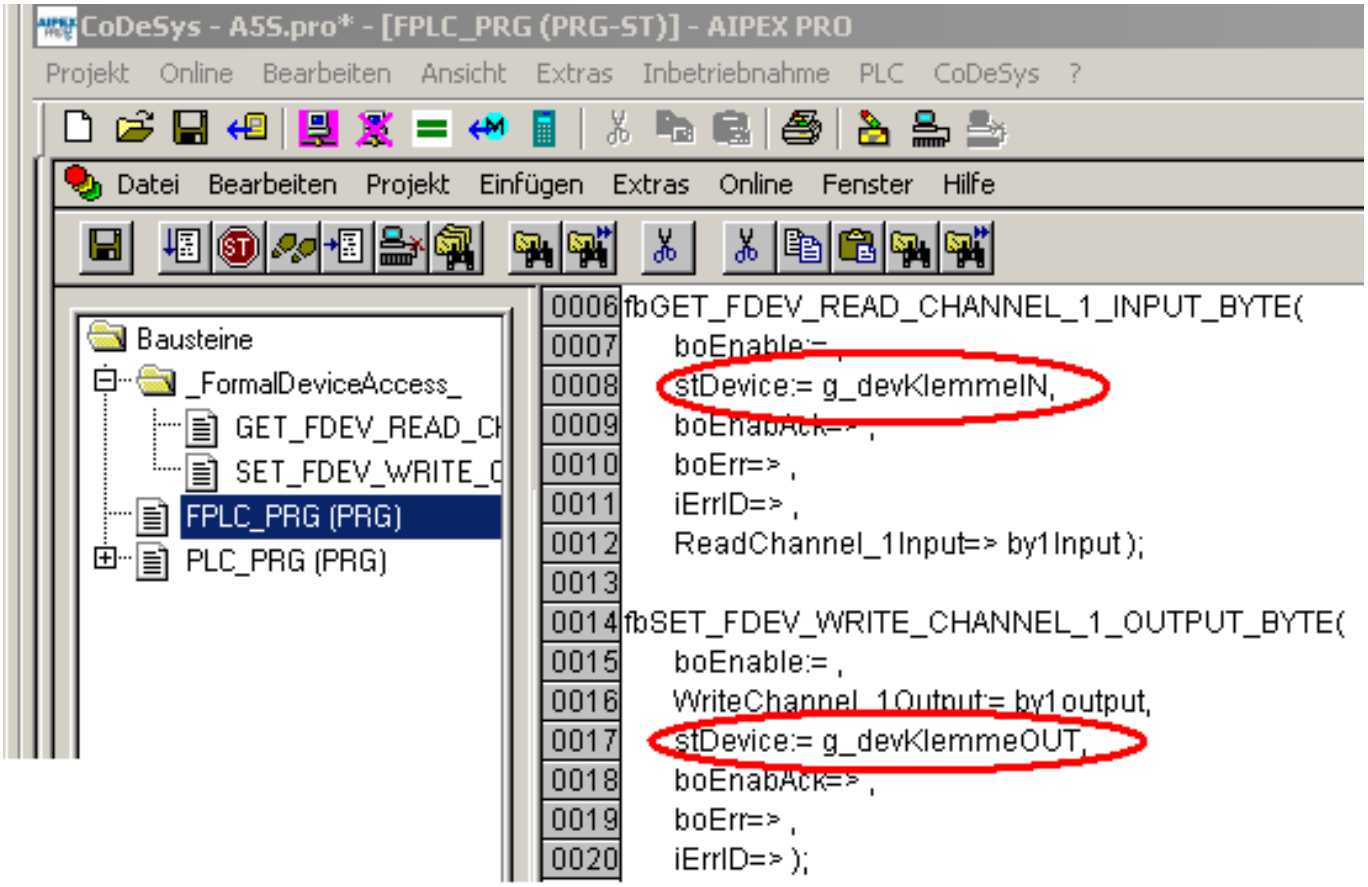


'stDevice' muss immer mit einem symbolischen Namen belegt sein. Dieser wird mit dem Geräteinterface der externen Klemme verknüpft.

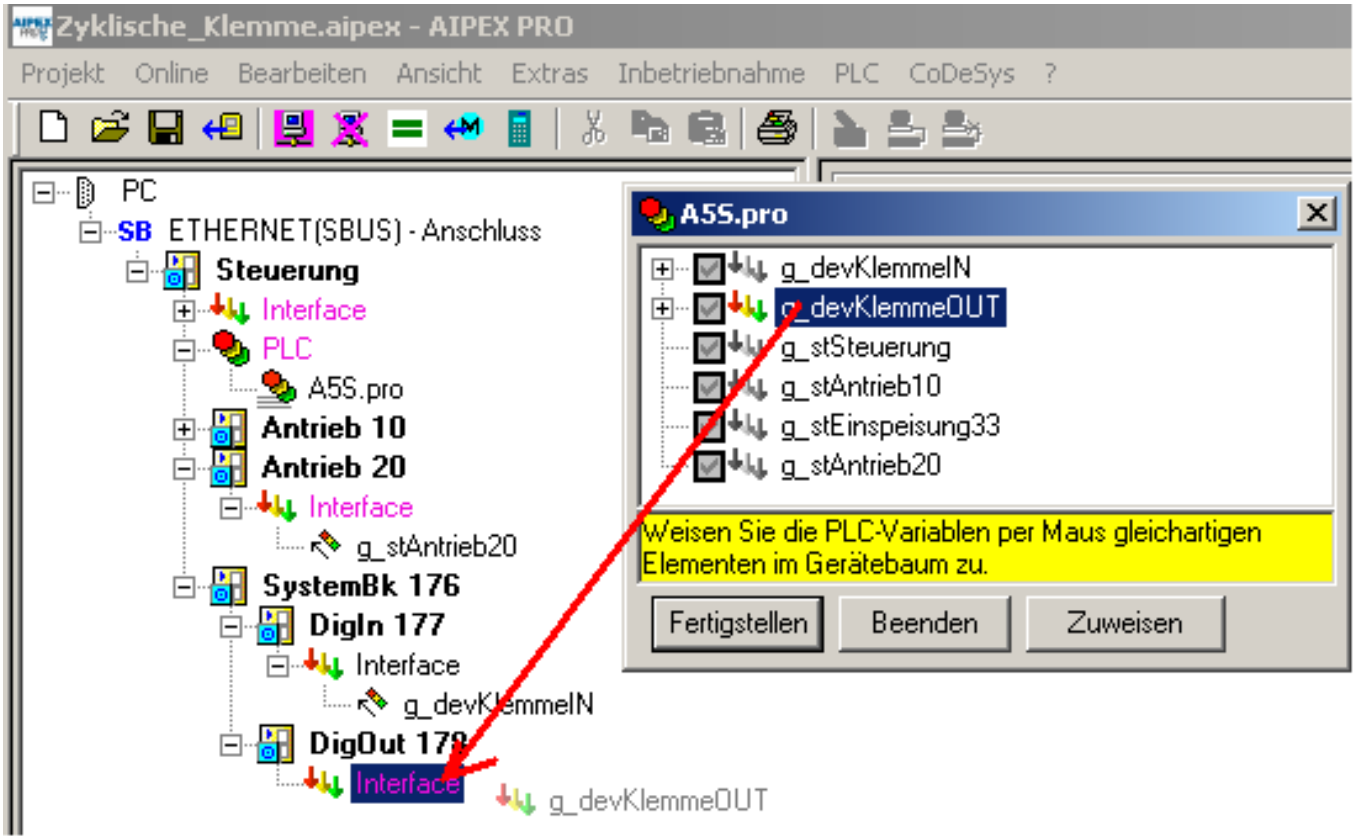
Legen Sie in der **Steuerungskonfiguration** unter **Devices** symbolische Namen für die Klemmen an.



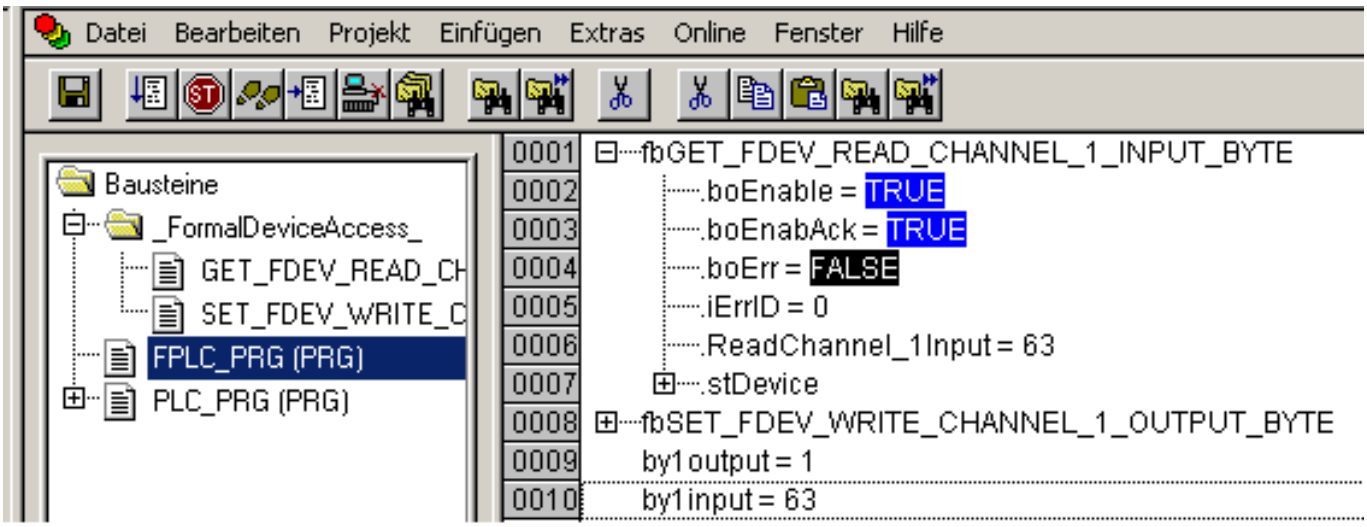
Verknüpfen Sie 'ST_DEVICE' mit dem symbolischen Gerätenamen.



Aktivieren Sie die automatische Nachrichtenkonfiguration, Menüpunkt **CoDeSys Nachrichtenkonfiguration erstellen**.
Ziehen Sie anschließend den symbolischen Gerätenamen auf das dazugehörige Geräteinterface.



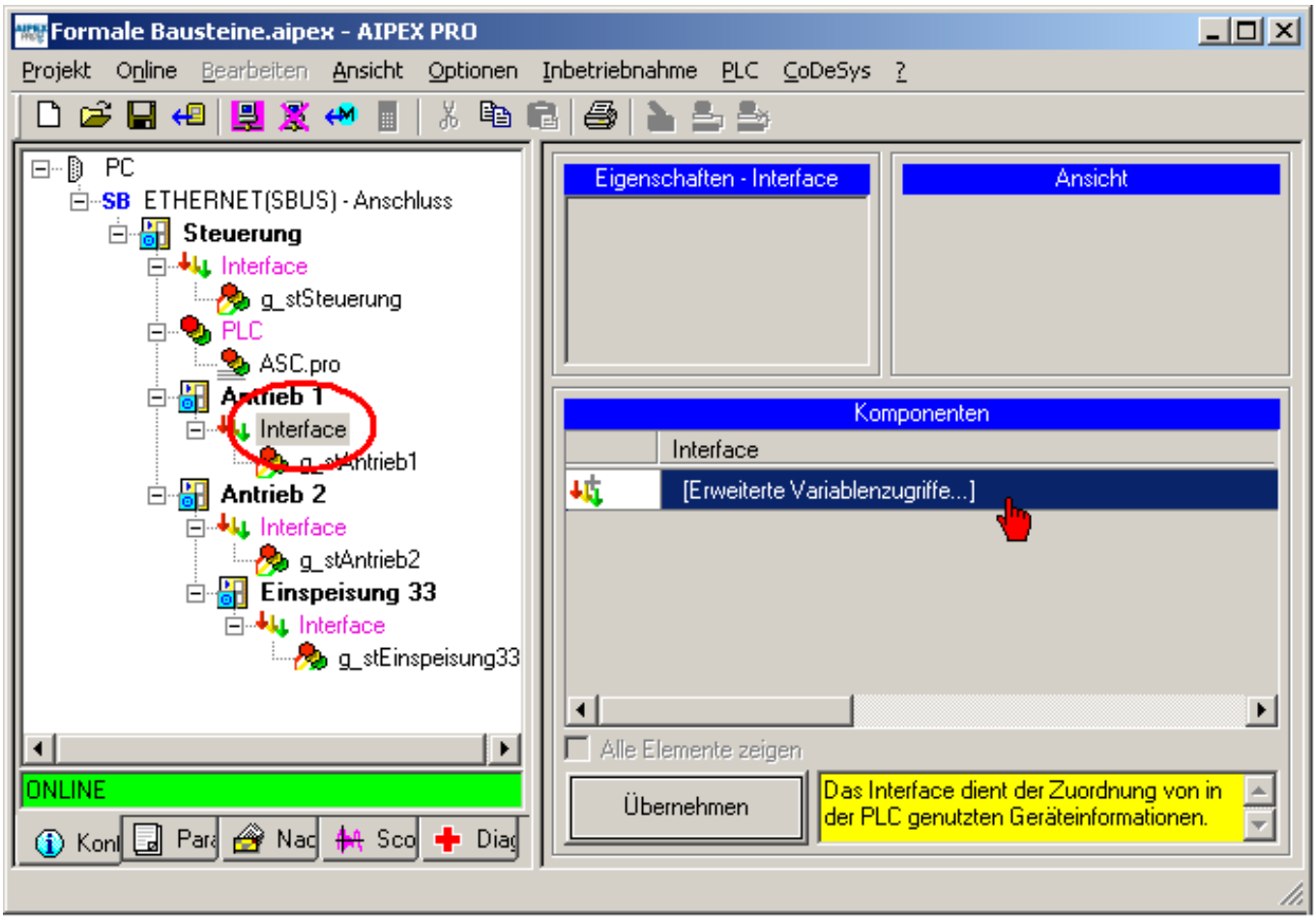
Setzen Sie 'boEnable' = TRUE um den Funktionsbaustein zu starten.



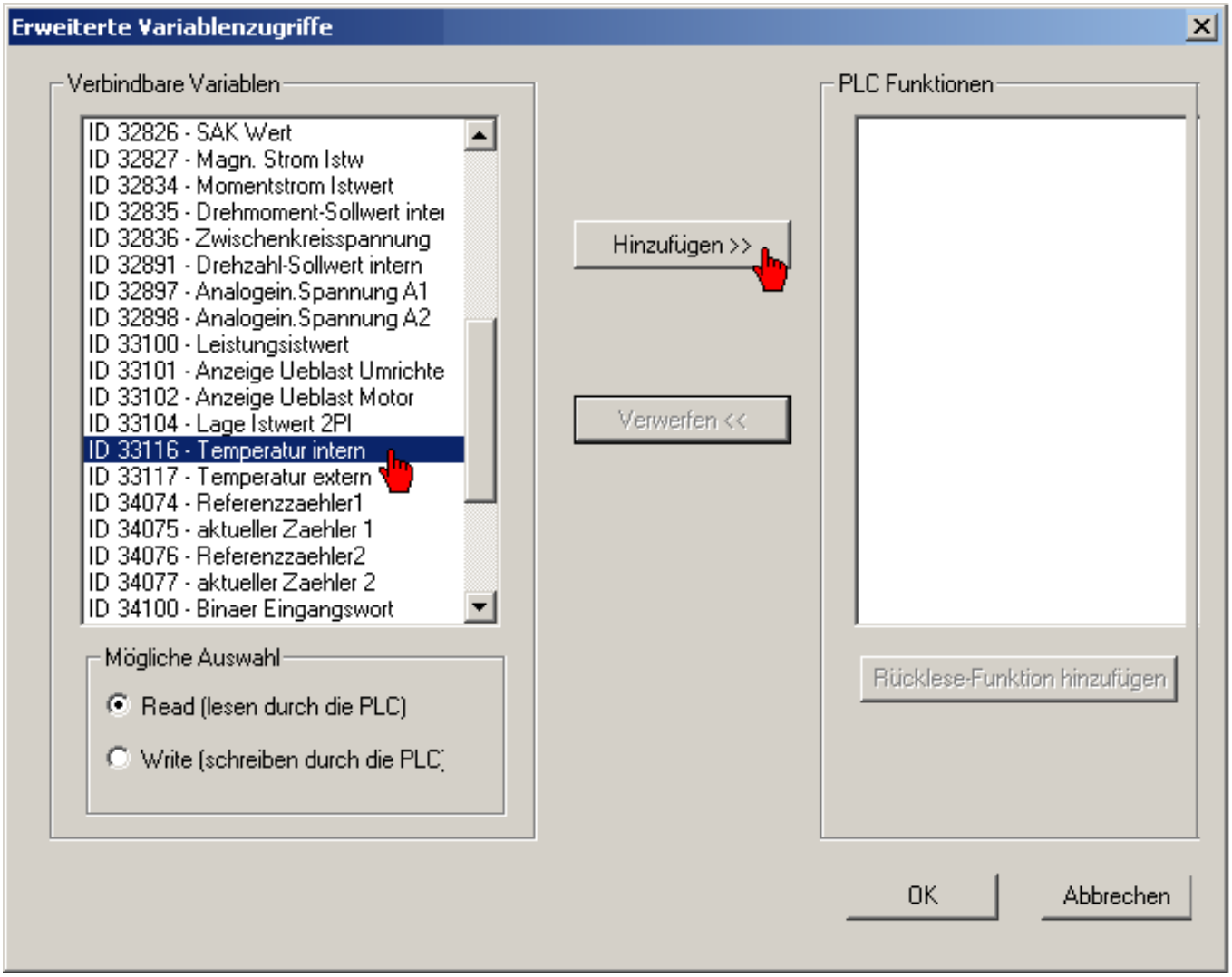
4.3.2.18.4 Erweiterte Variablenzugriffe

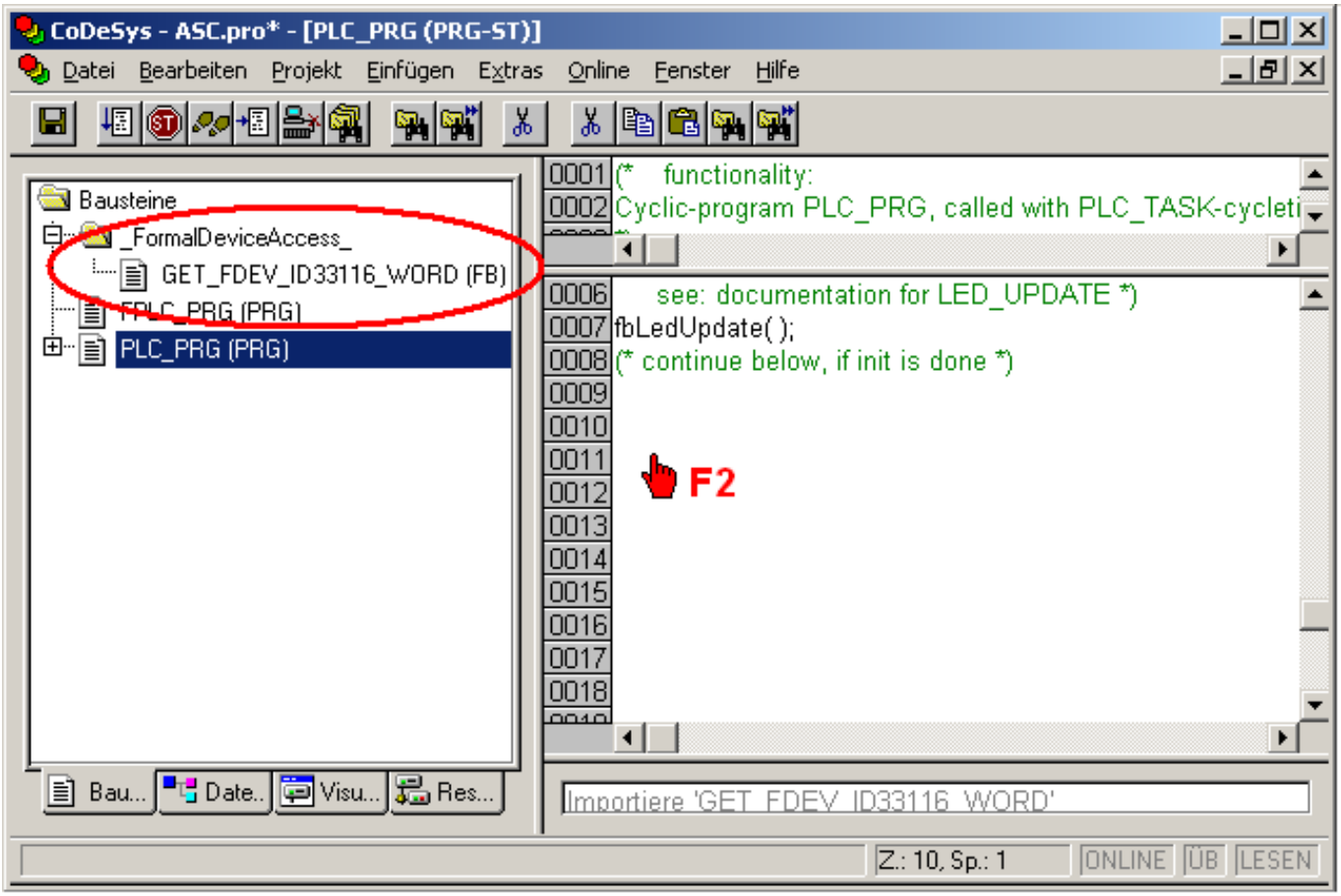
Die AMK CoDeSys Bibliotheken stellen eine Auswahl an Funktionsbausteinen zu Verfügung um Echtzeitbitmeldungen, Istwerte, Sollwerte usw. zwischen Antrieb und PLC auszutauschen. Nicht vorhandene (formale) Bausteine können über die Option Erweiterte Variablenzugriffe erstellt werden.

Wählen Sie dazu den Antrieb (Interface) aus für den ein neuer Funktionsbaustein erstellt werden soll.

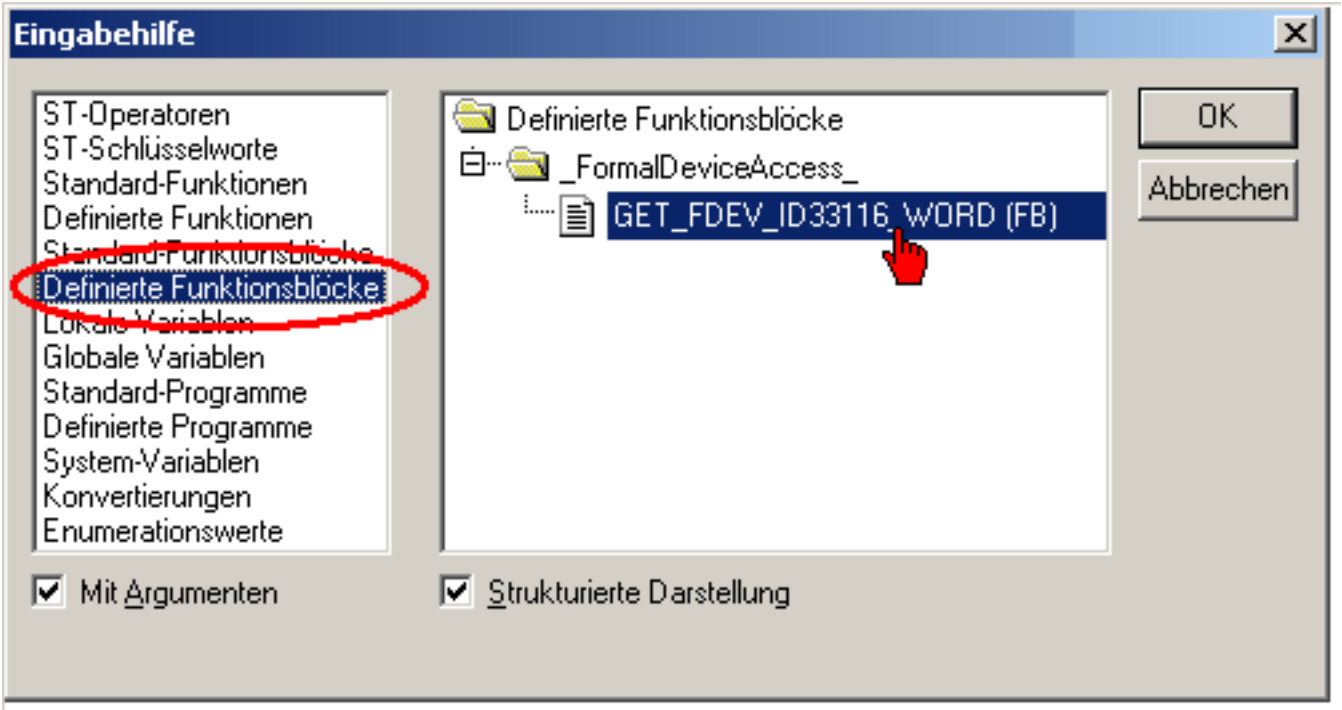


Wählen Sie im ersten Schritt aus ob von der PLC lesend oder schreibend auf den Antrieb zugegriffen werden soll.
In der Variablenliste finden Sie alle im Antrieb verfügbaren Werte.
Die Variablenauswahl wird mit der Taste **Hinzufügen** übernommen.

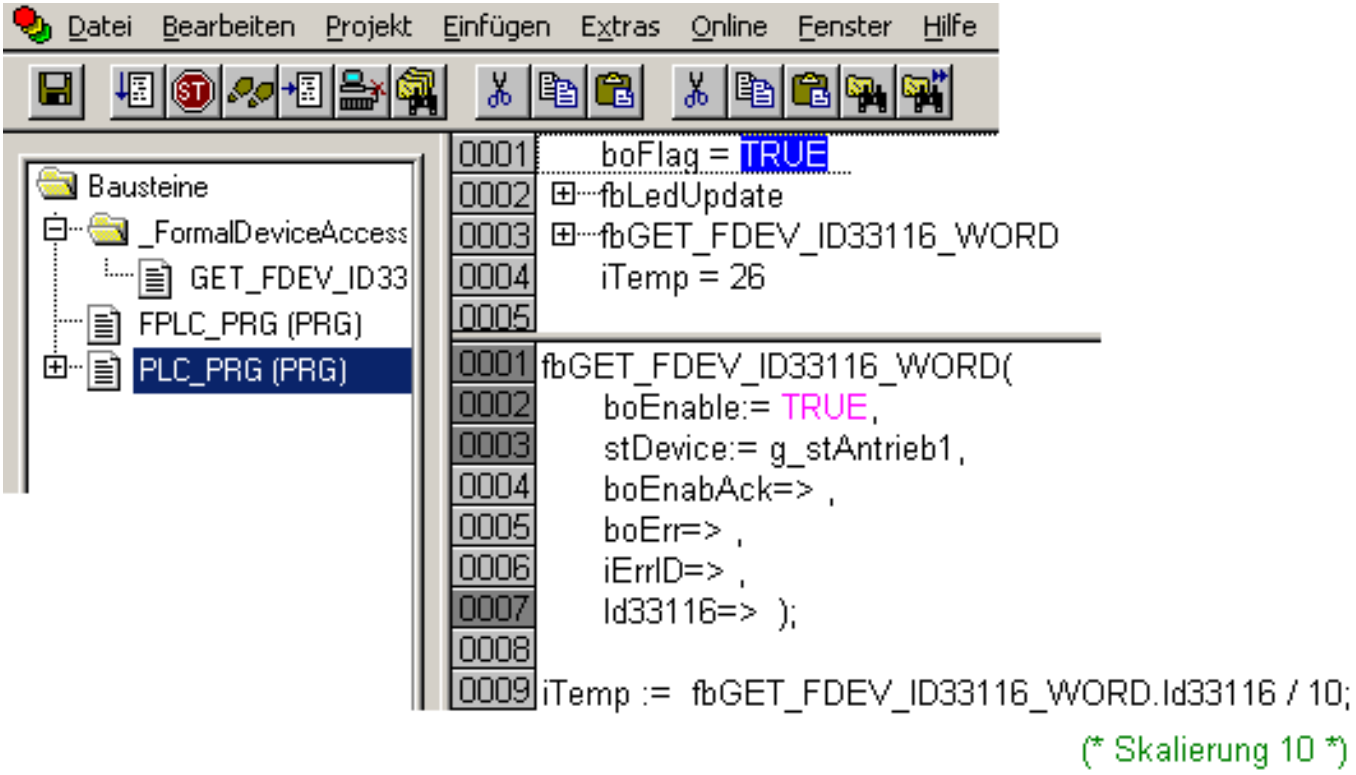




Über die Eingabehilfe F2 können die Funktionsblöcke in das PLC Projekt eingebunden und instanziert werden.



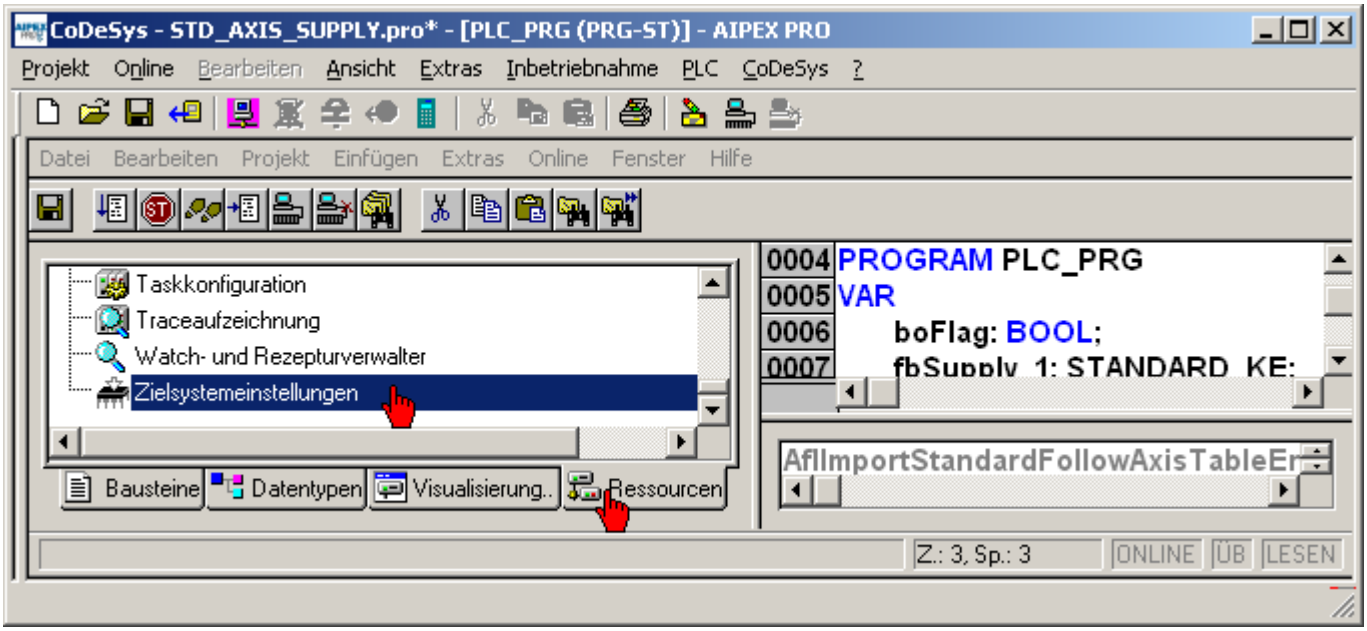
Beachten Sie die Skallierung der verschiedenen Einheiten. Im Beispiel Faktor 10.



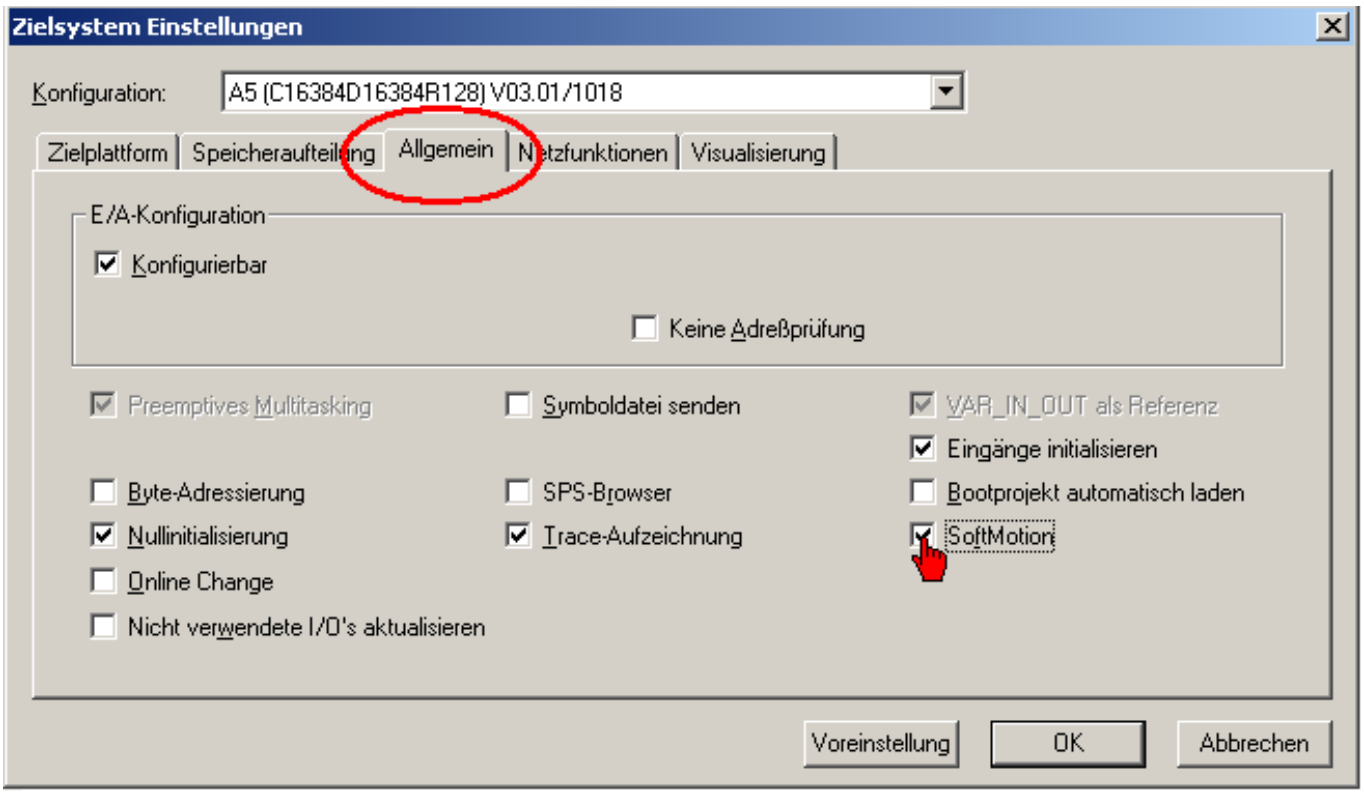
4.3.2.18.5 Kurvenscheibe erstellen

Das Beispiel zeigt, wie Sie eine Kurvenscheibe erstellen und diese mit einem FB der Standard Achsen verknüpfen.

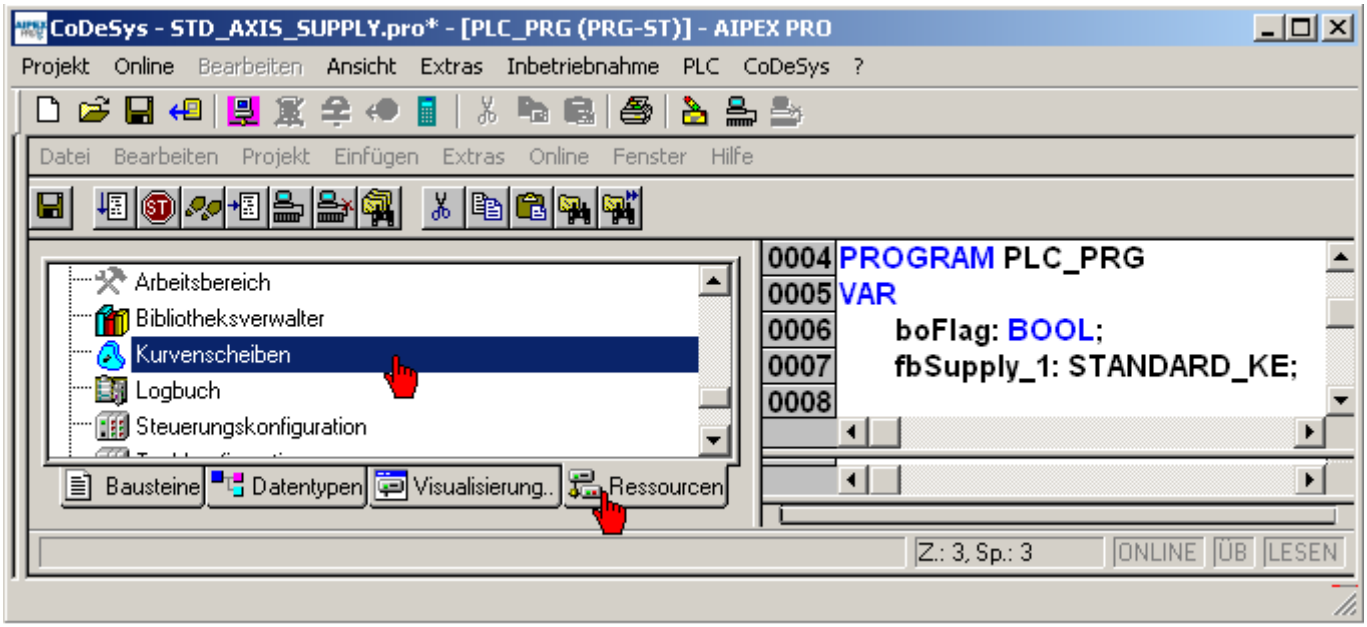
Öffnen Sie zuerst das Fenster **'Zielsystem Einstellungen'** unter **'Ressourcen'**.



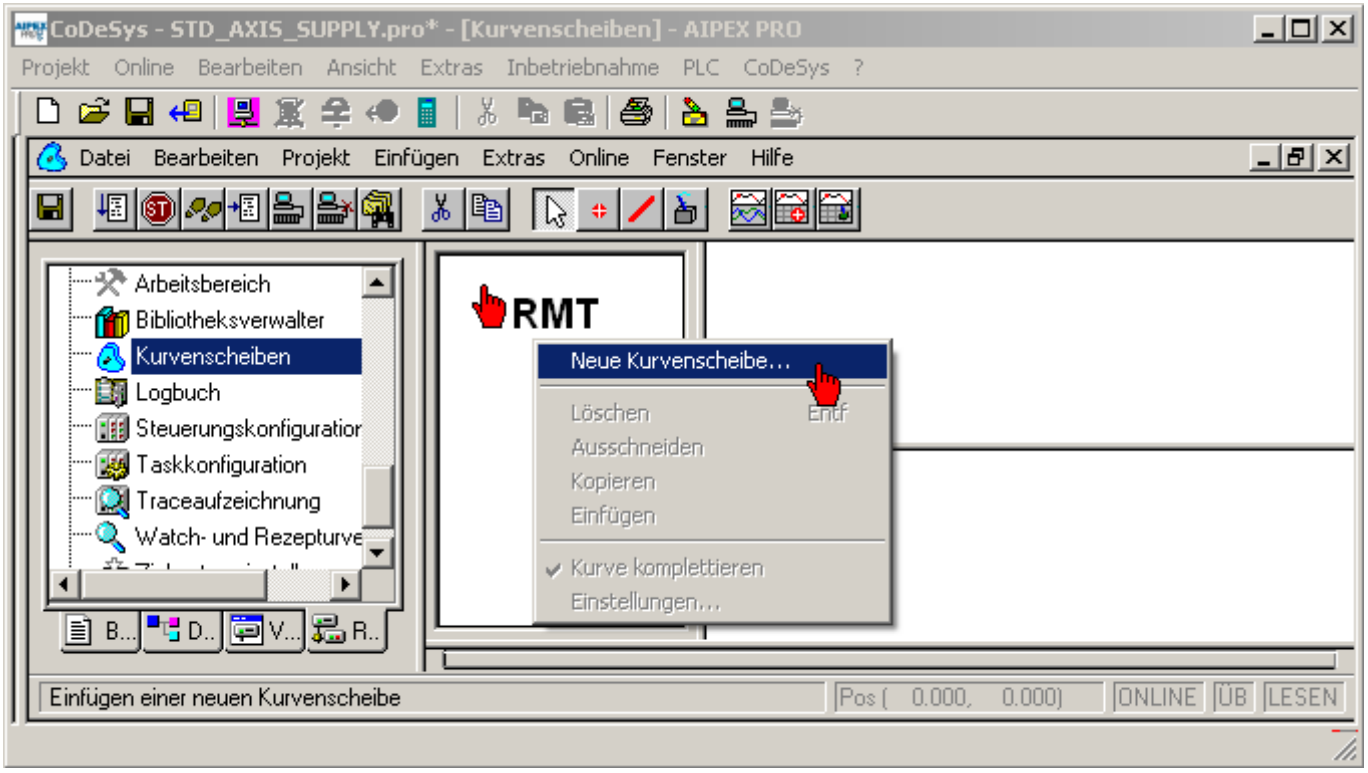
Aktivieren Sie **'SoftMotion'** unter **'Allgemein'**.



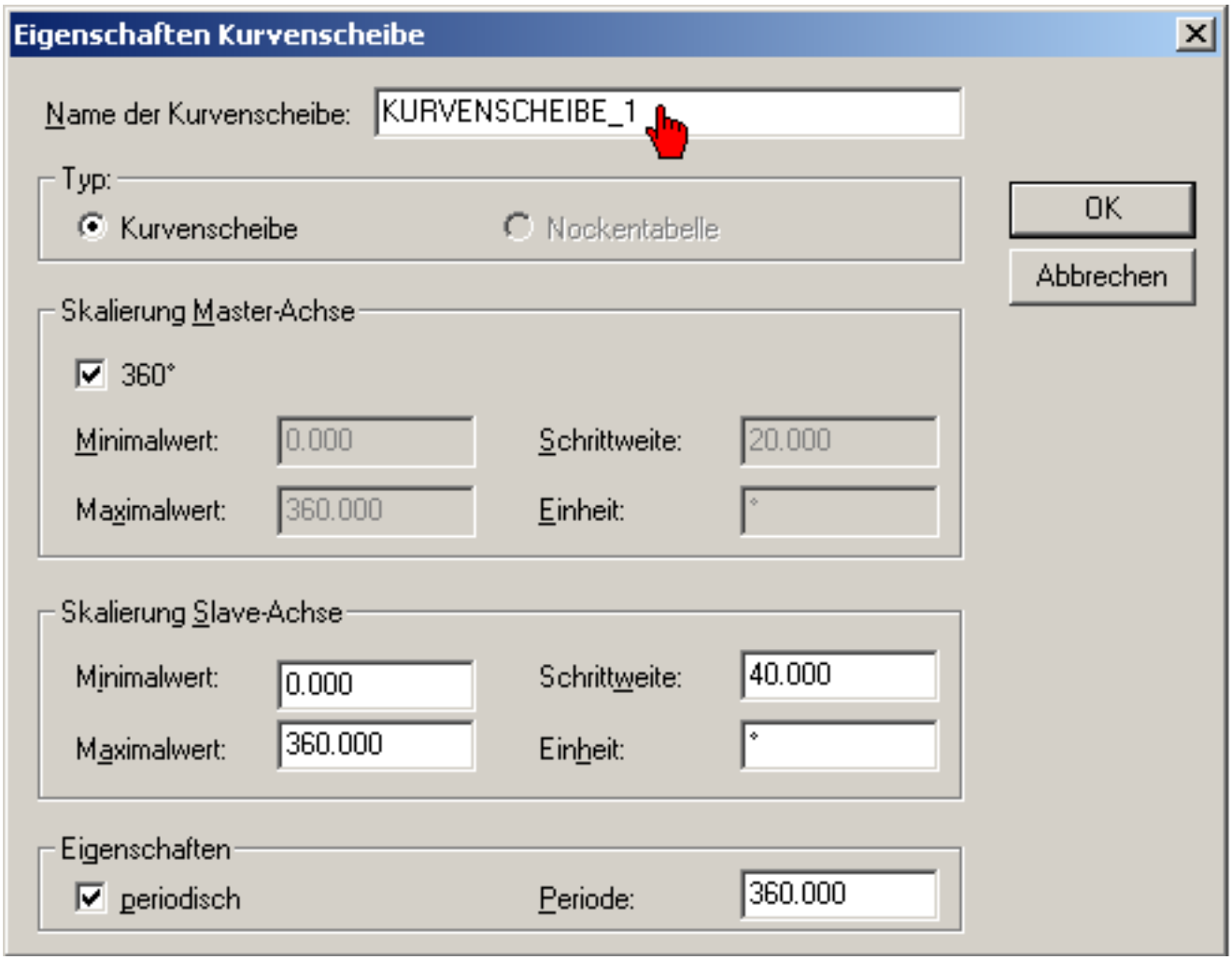
Jetzt finden Sie unter den **'Ressourcen'** das Menü **'Kurvenscheibe'**.

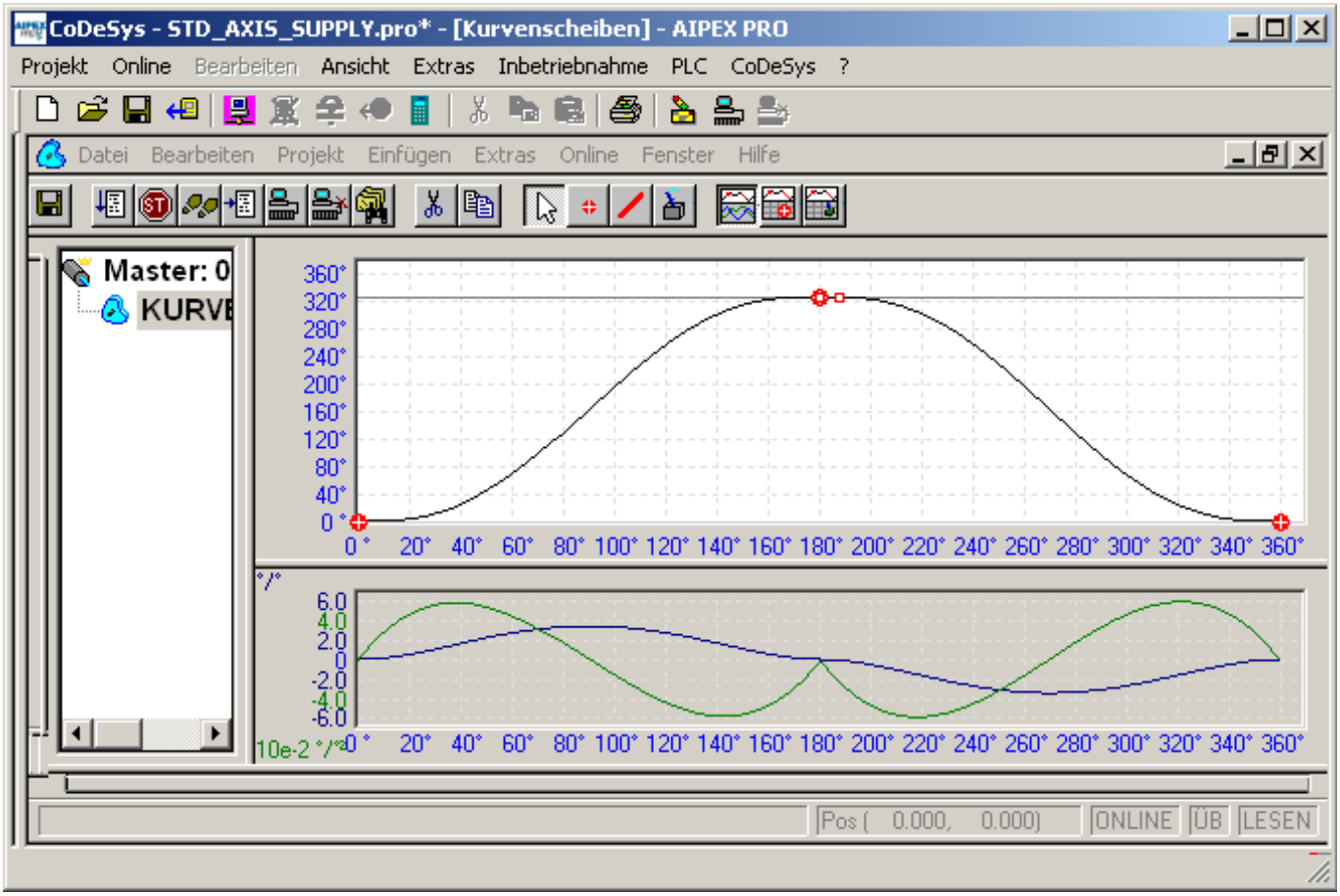


Klicken Sie mit der rechten Maustaste in das mittlere Feld um eine neue Kurvenscheibe einzufügen.

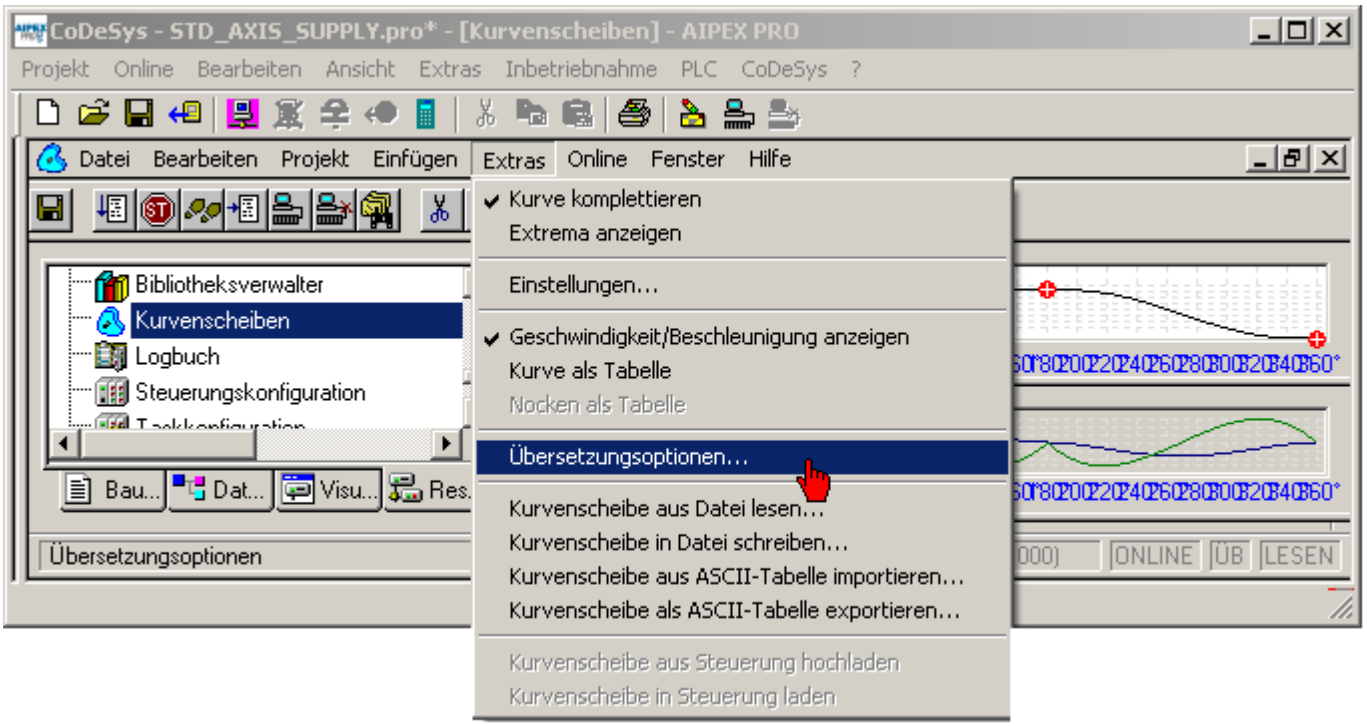


Sie finden in der CODESYS Hilfe detaillierte Informationen zur Kurvenscheibe und deren Einstellungen. Bitte verwenden Sie diese.



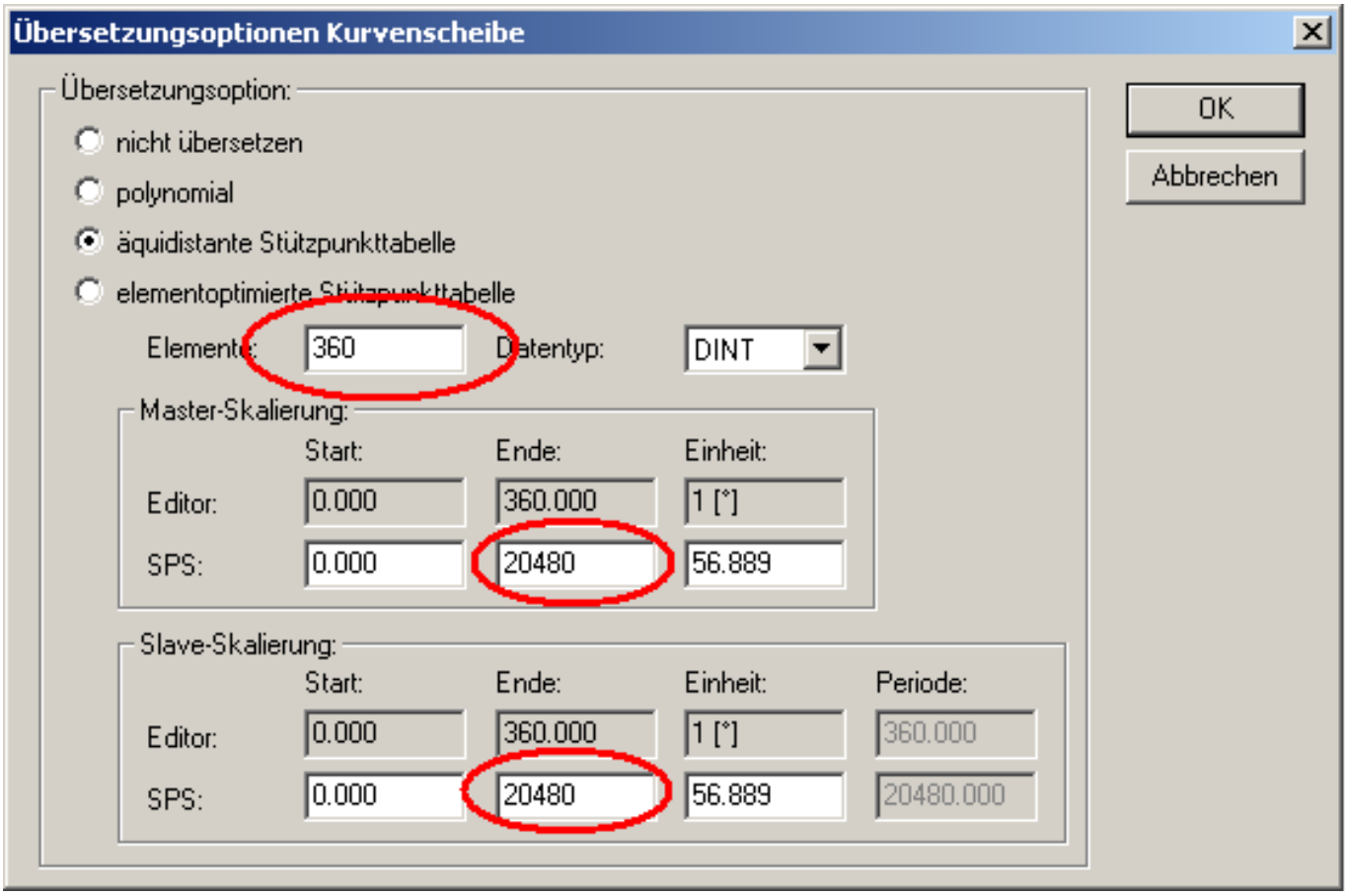


In den **'Übersetzungsoptionen'** können Sie die Anzahl der Stützpunkte und die Skalierung (Motorgeberauflösung) eingeben.

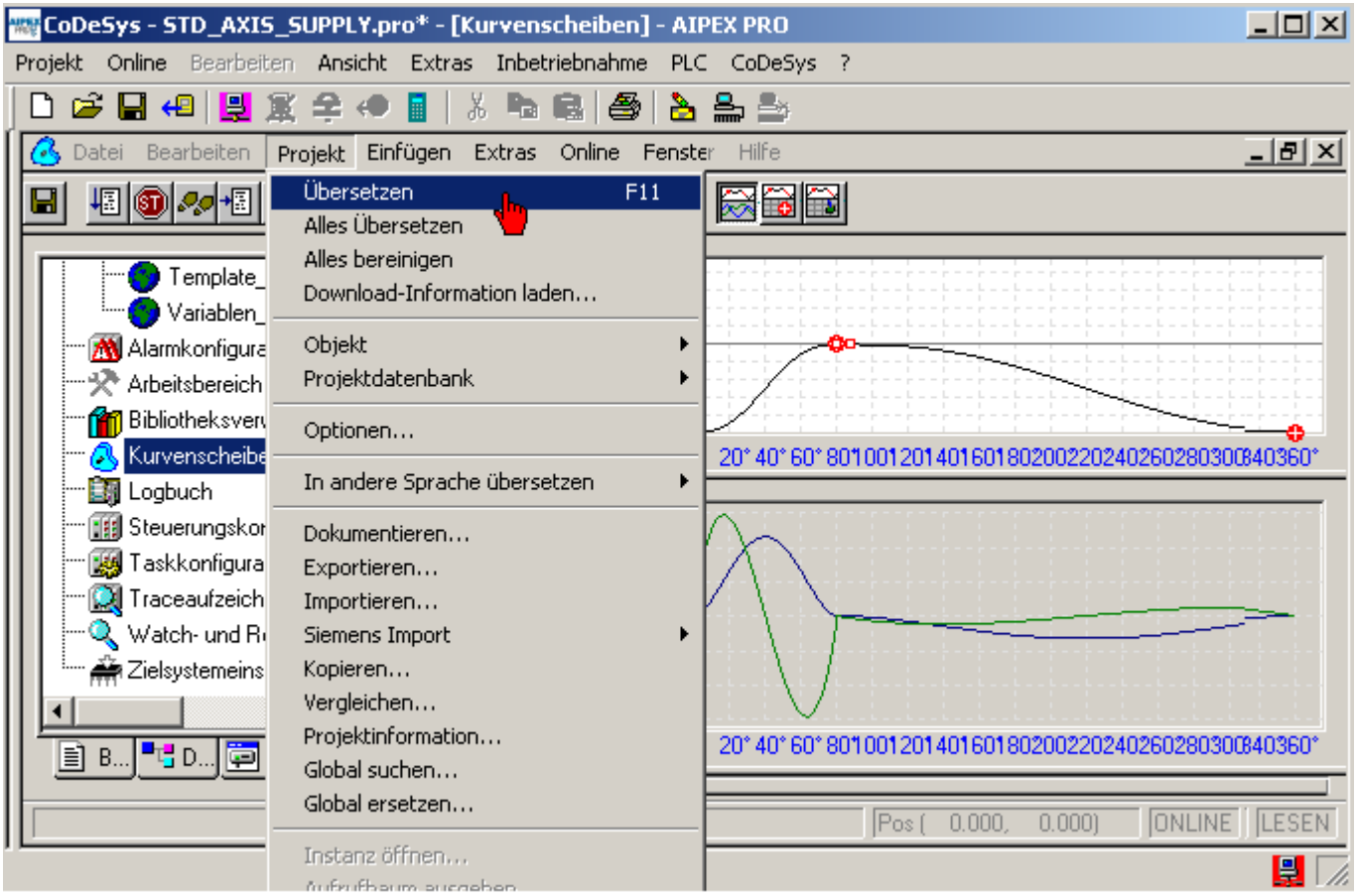


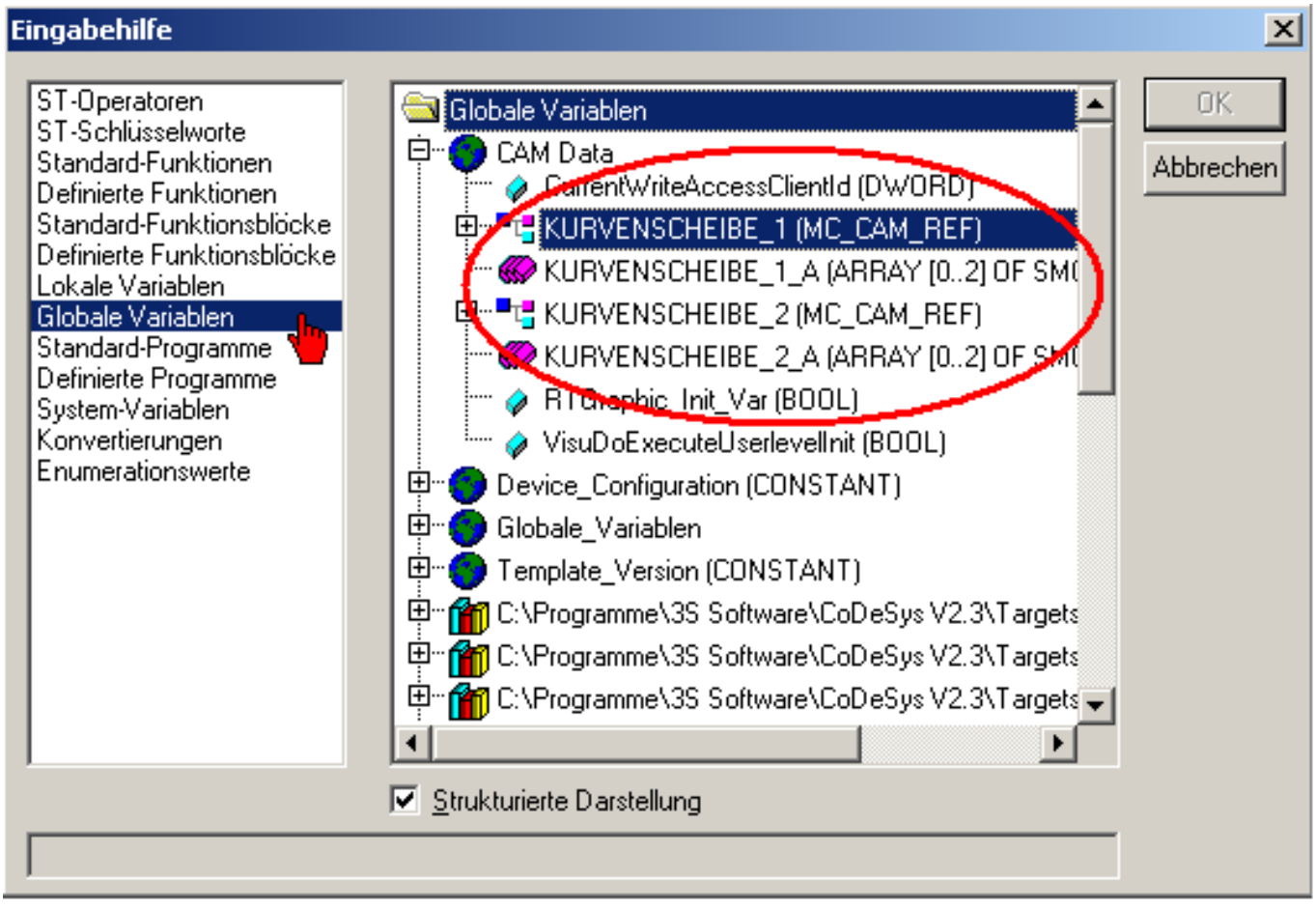
Äquidistante Stützpunkttabelle max. 361 DINT Werte
Elementoptimierte Stützpunkttabelle max. 181 DINT Werte

Skalierung: 360° entspricht der ID116 'Auflösung Motorgeber'

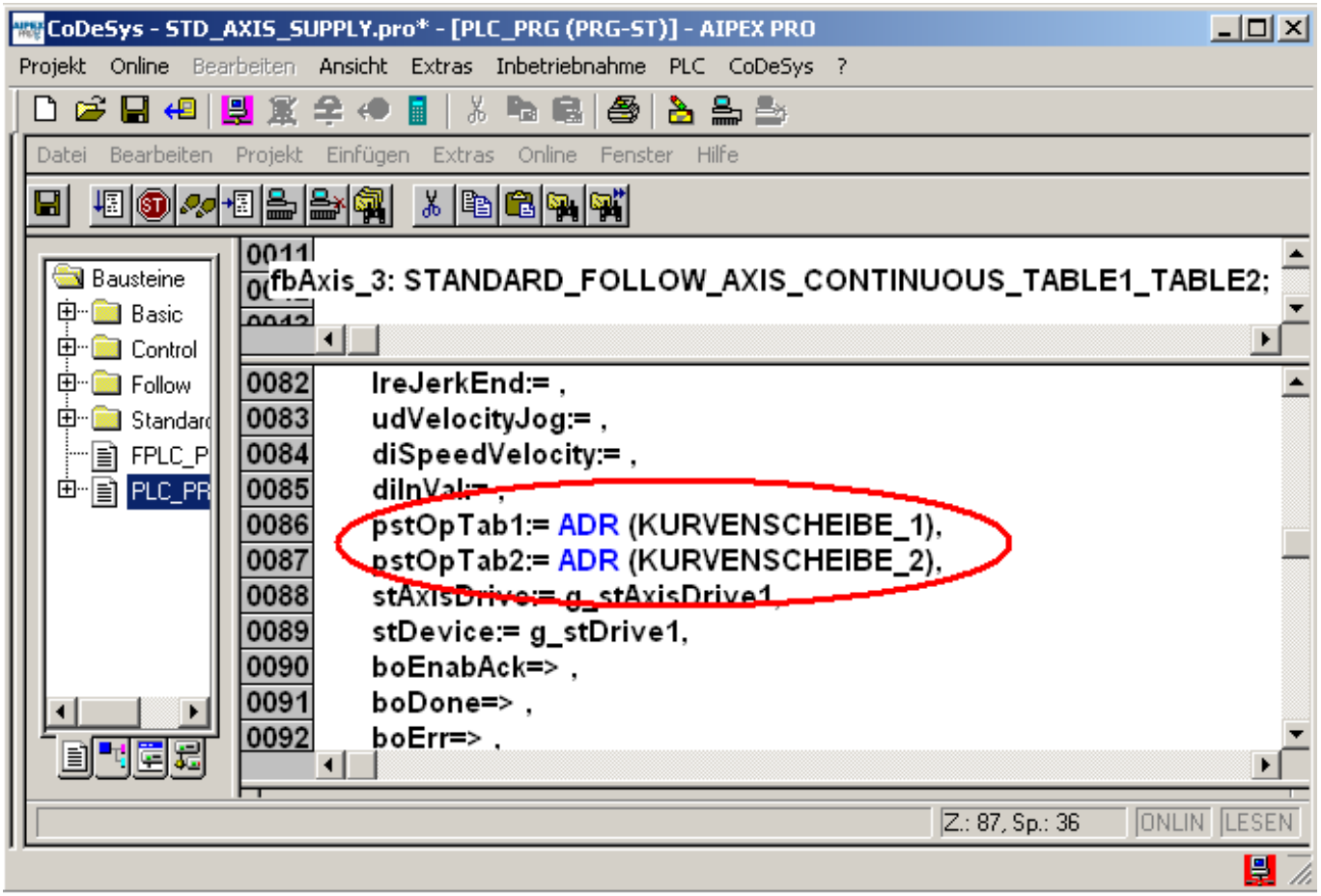


Nach dem Übersetzen finden Sie die Daten der Kurvenscheibe in den **'Globalen Variablen'**.





Verwenden Sie die Adressfunktion **'ADR'** um die Kurvenscheibe mit ihrem Baustein zu verknüpfen.

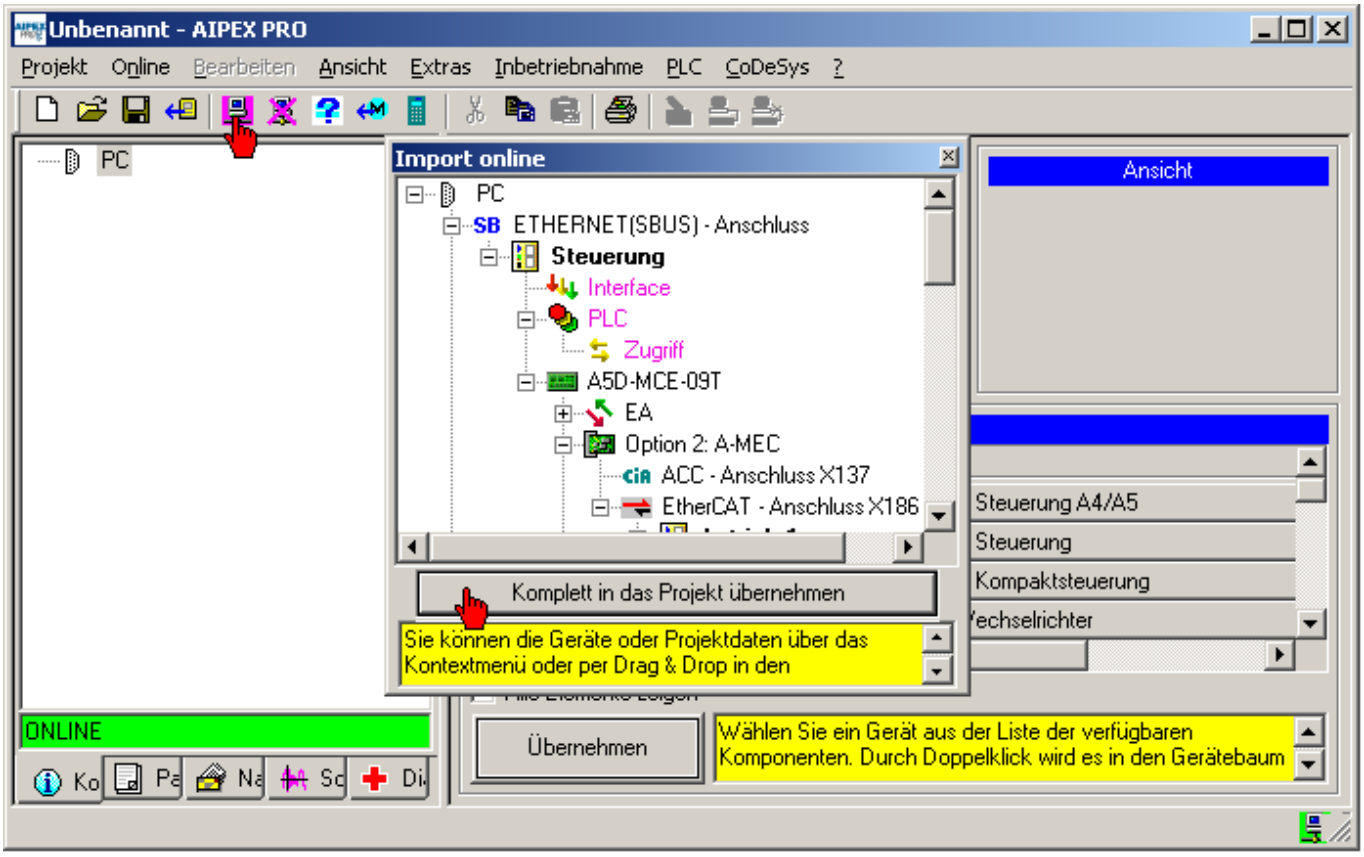


4.3.2.18.6 PLC Projekt von AIPEXPRO importieren

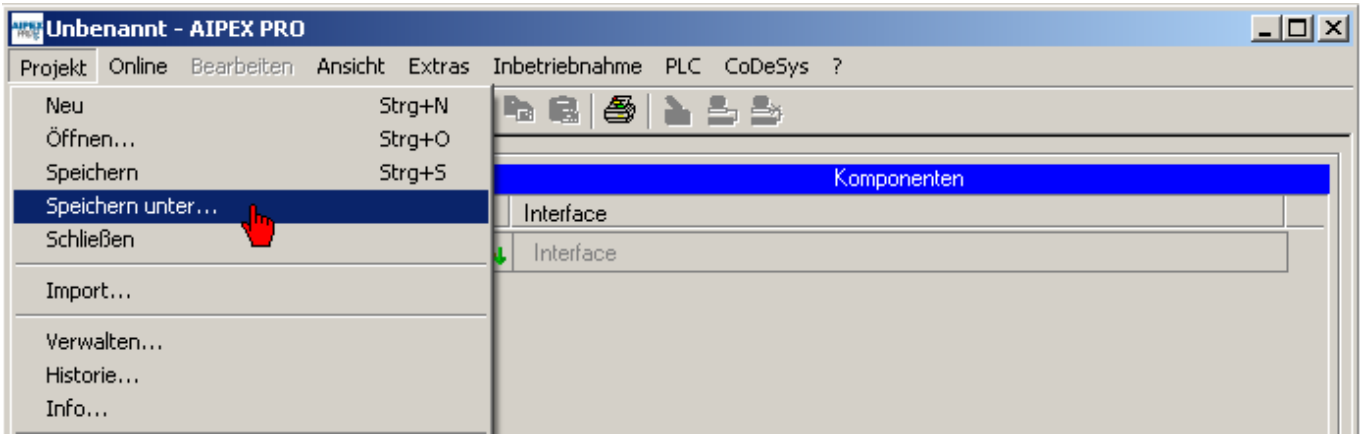
Öffnen Sie unter **Projekt - Neu** ein leeres AIPEX PRO Projekt.

Betätigen Sie den Taster **Einloggen**.

Übernehmen Sie die angeschlossenen Geräte in ihr AIPEX PRO File.

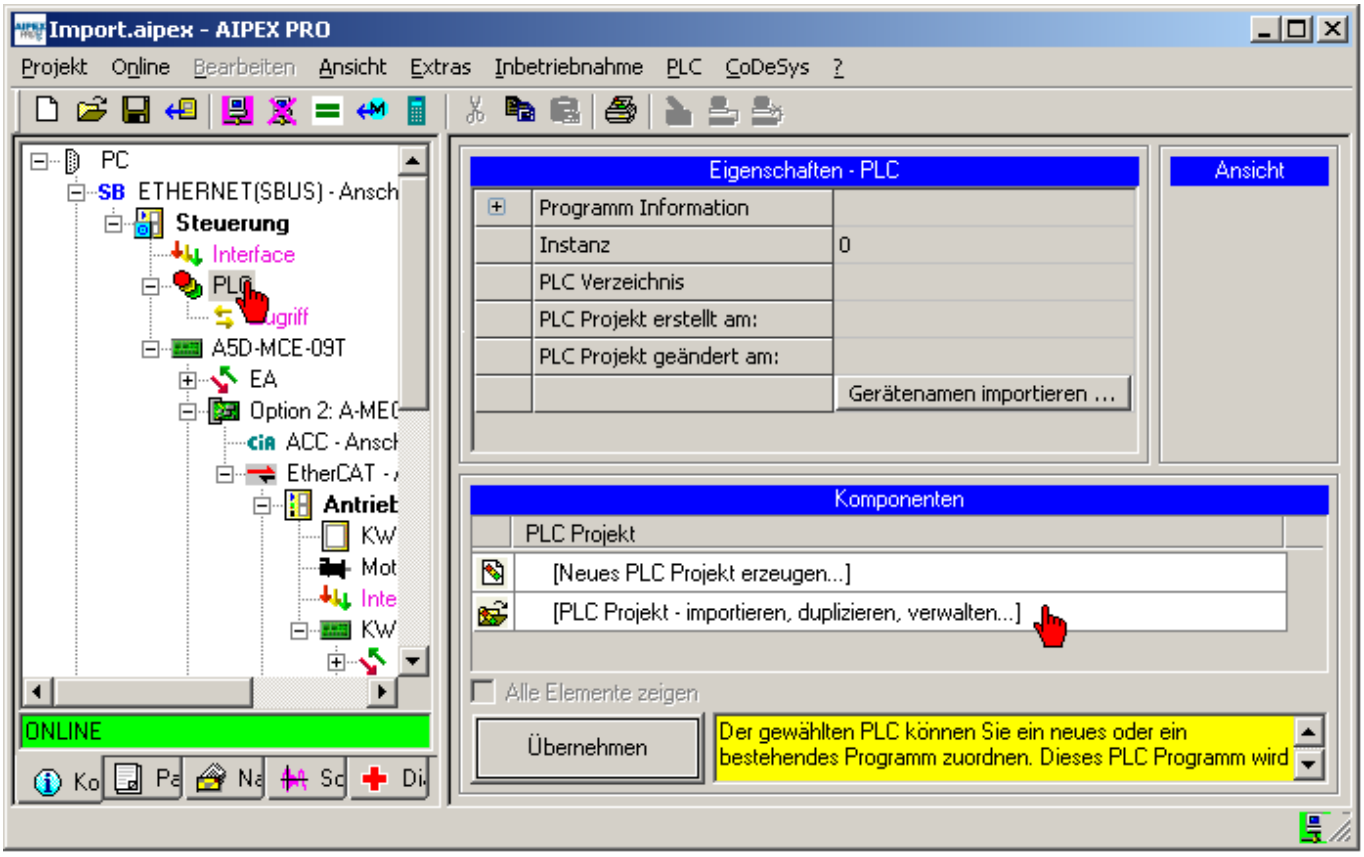


Speichern Sie die ausgelesenen Gerätedaten ab.

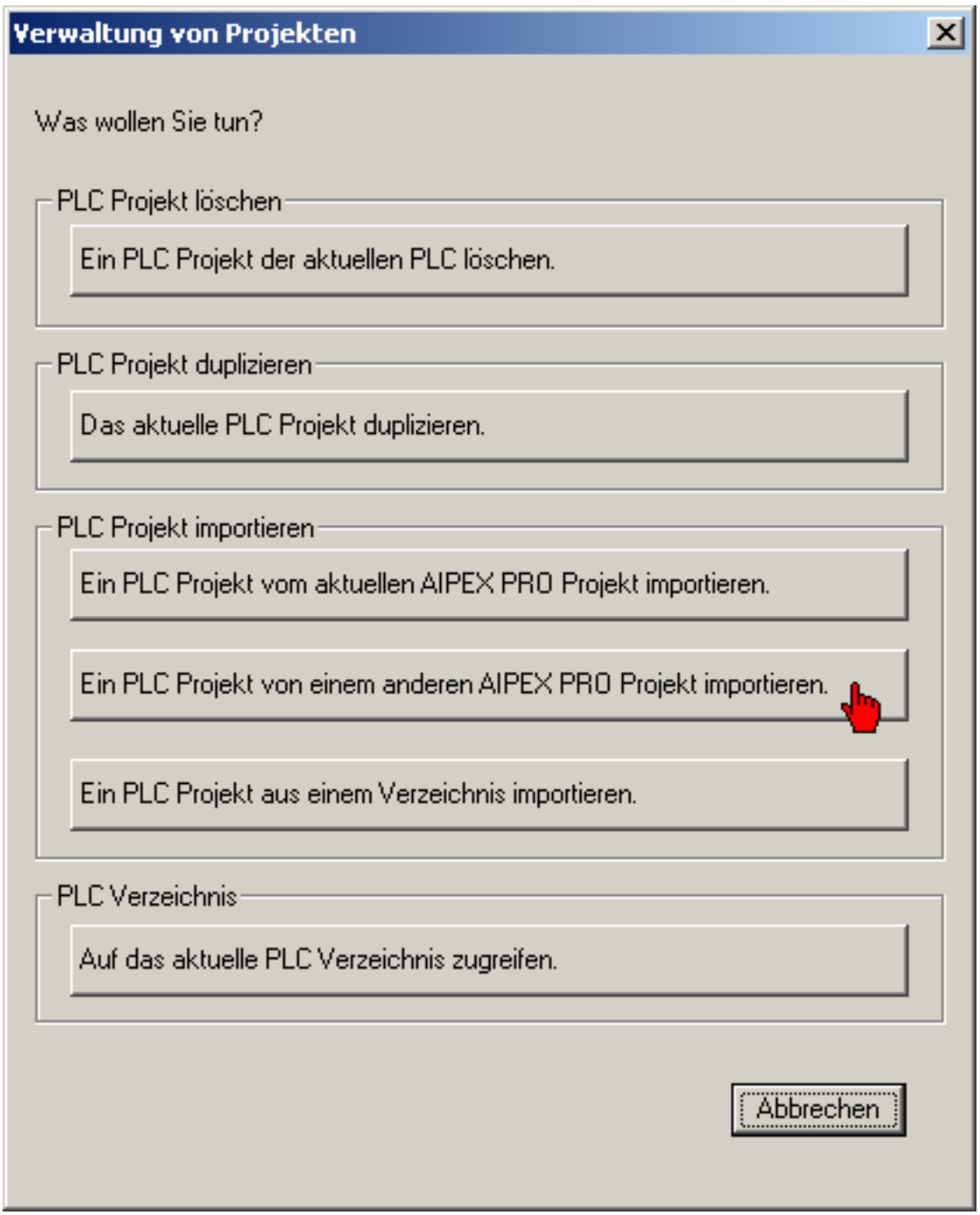


Wählen Sie die **PLC** im Gerätebaum an.

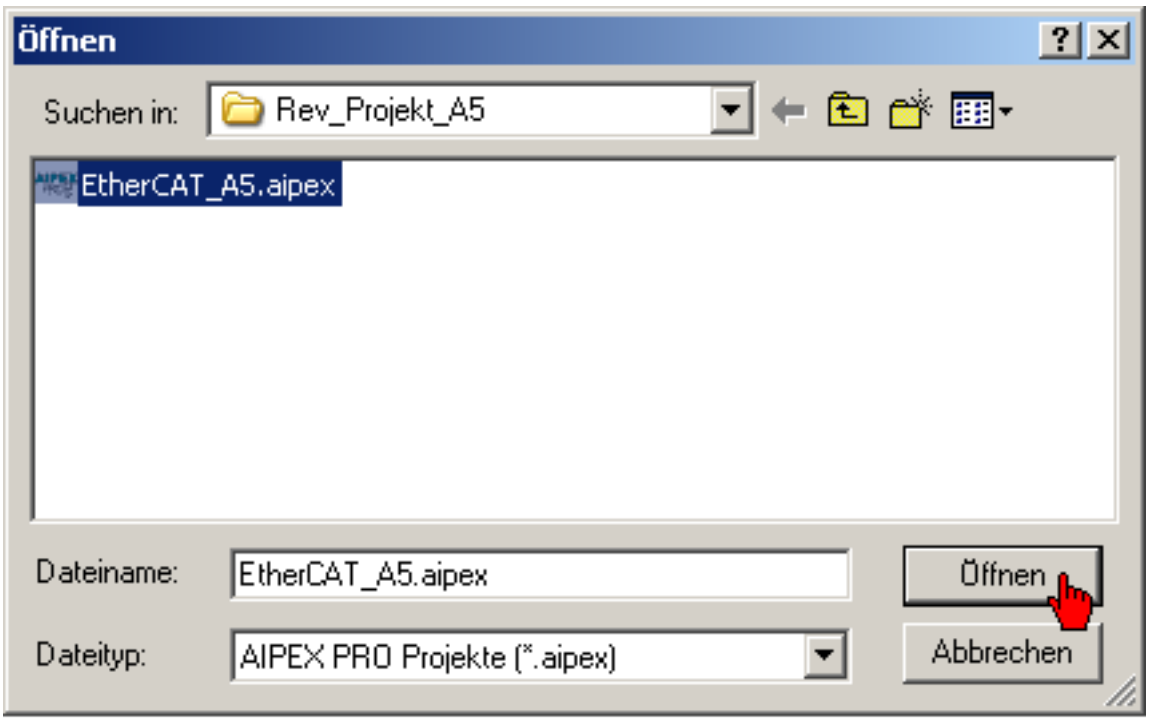
Anschließend die Auswahl **[PLC Projekt - importieren, dublizieren, verwalten ...]**



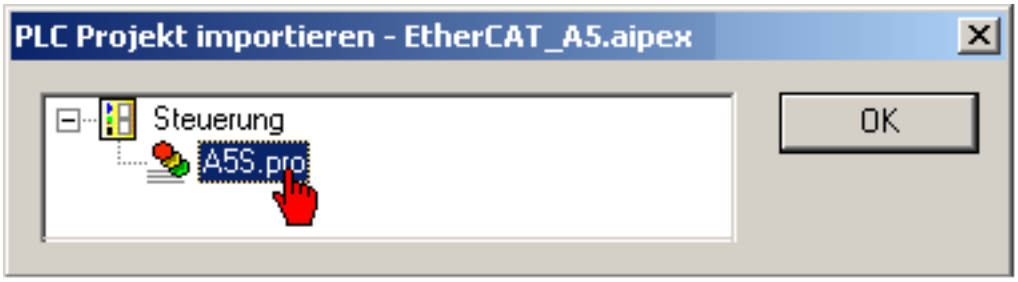
Wählen Sie **Ein PLC Projekt von einem anderen AIPEX PRO Projekt importieren** an.



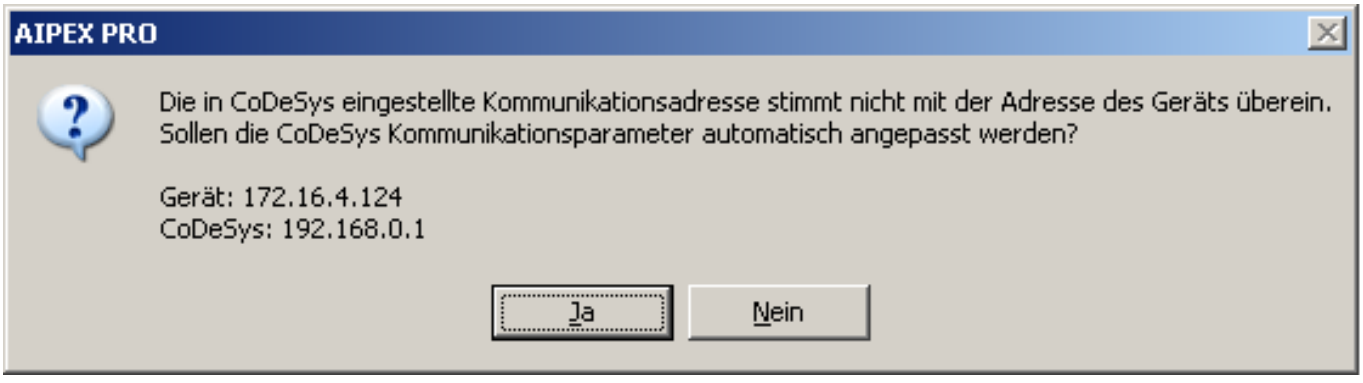
Öffnen Sie das gewünschte File, dass das SPS Programm enthält.



Öffnen Sie das SPS Programm [*.pro].

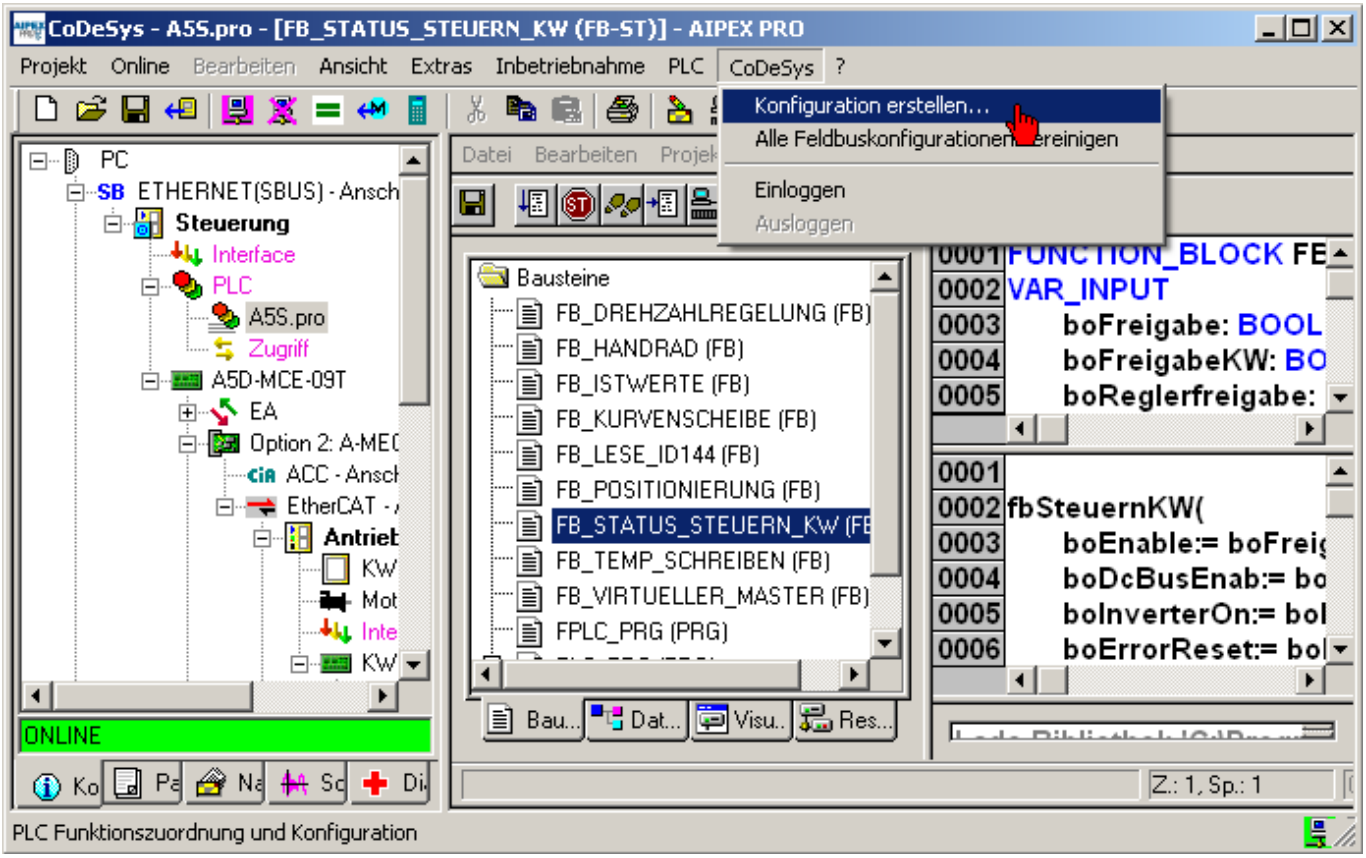


Gleichen Sie eventuell unterschiedliche Adressen ab.



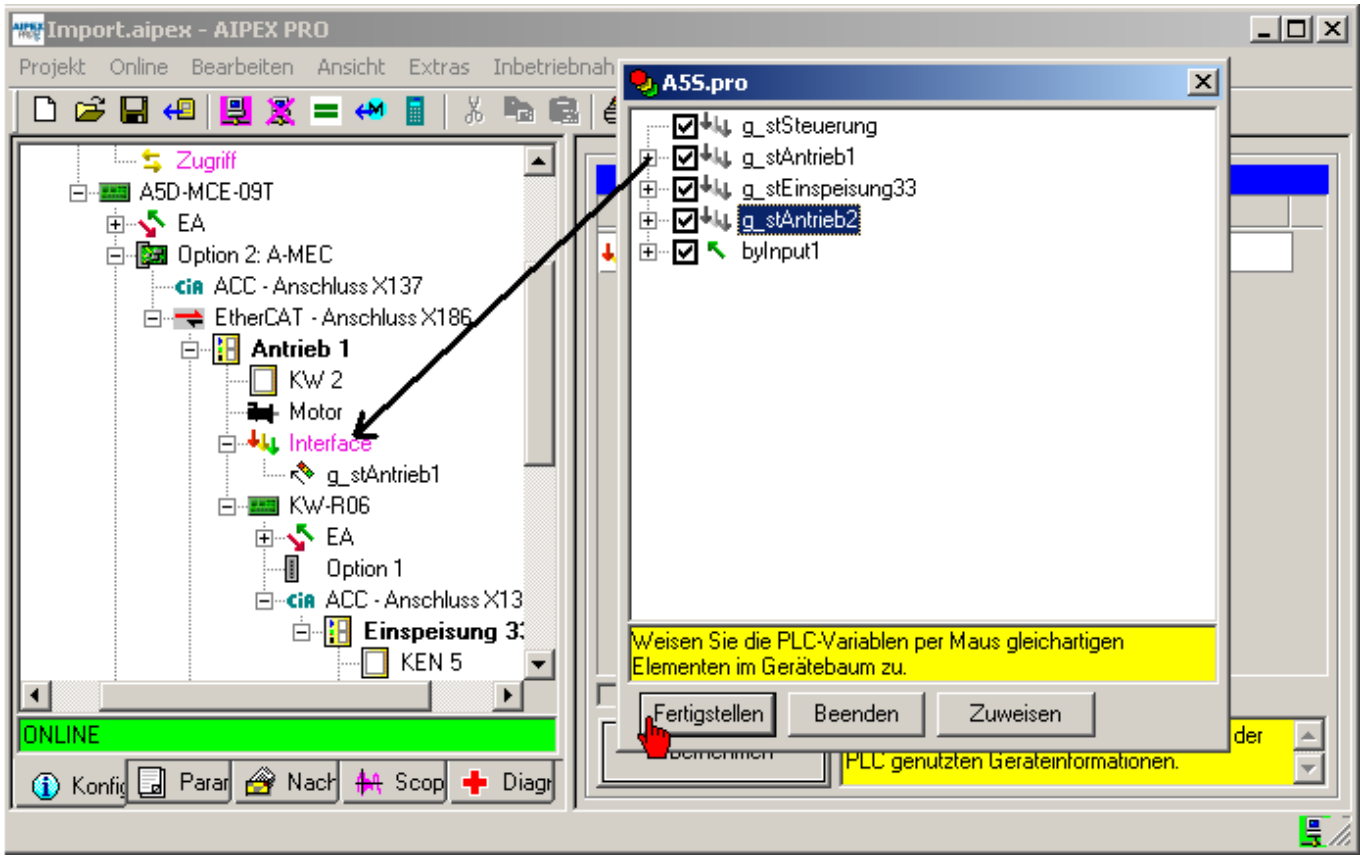
CoDeSys öffnet sich automatisch.

Erstellen Sie die neue Nachrichtenkonfiguration unter **CoDeSys** - **Konfiguration erstellen**.



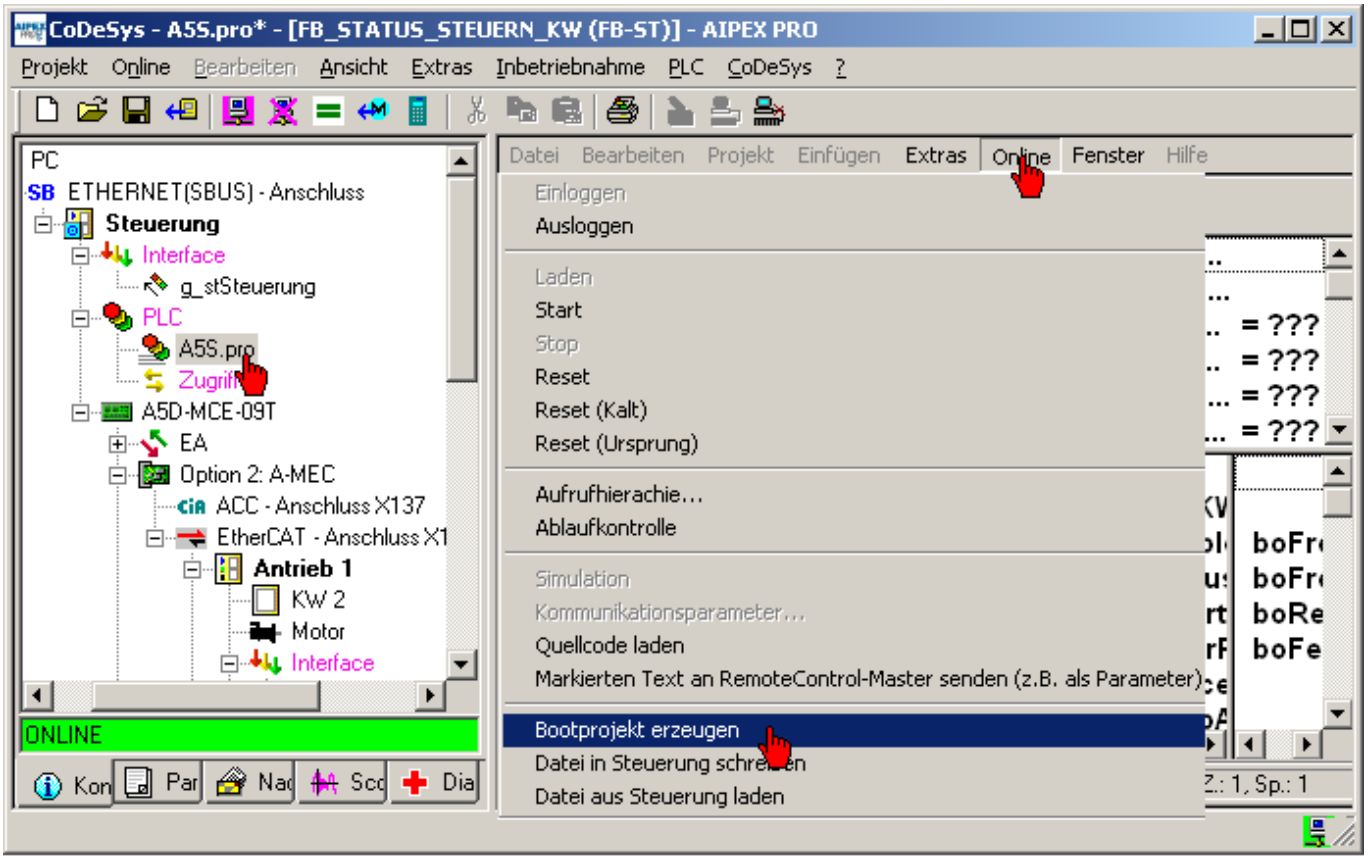
Klicken (und halten) Sie mit der linken Maustaste Ihren symbolischen Gerätenamen. Ziehen Sie Ihn nun auf Ihr physikalisch vorhandenes Gerät im AIPEX PRO Gerätebaum (Interface).

Drücken Sie anschließend den Taster **Fertigstellen**.



Laden Sie das SPS Programm auf die Steuerung.

Erstellen Sie ein Bootprojekt über **Online** - **Bootprojekt erzeugen**.
Starten Sie die Geräte anschließend neu.



4.3.2.18.7 Statusbits erweitern

Die Statusbits werden mit der ID26 'Konfiguration Statusbits' im Gerät konfiguriert.

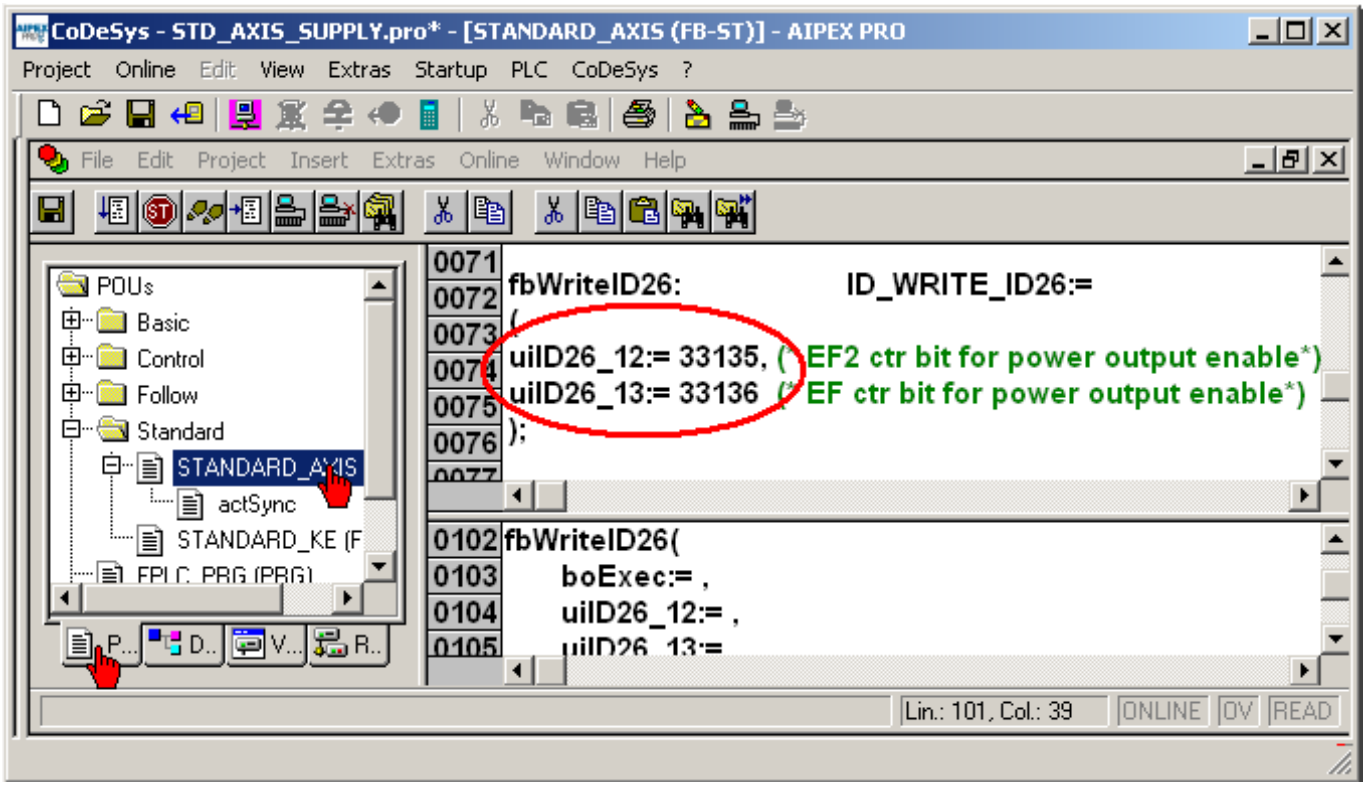
Die Standard Achse verwendet den Funktionsbaustein 'ID_WRITE_ID26' um eine Konfiguration von der PLC in das Gerät zu übertragen.

Die Listenelemente 2-11 sind vorbelegt. Siehe 'ST_STATUSBITS (ST)' auf Seite 194.

Für die Listenelemente 12 – 17 können weitere Codes frei gewählt werden. Die Codes sind der Gerätebeschreibung zu entnehmen.

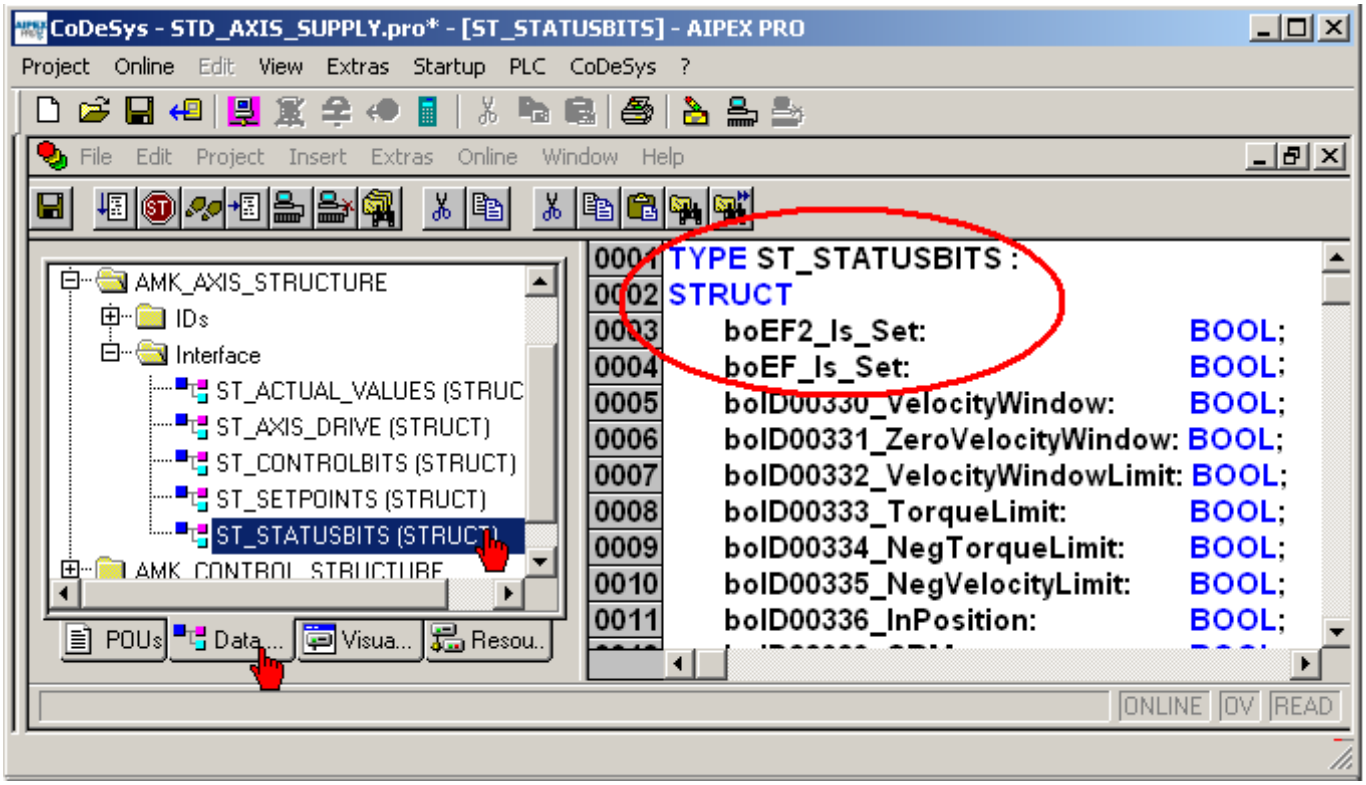
Das Beispiel zeigt wie Sie zusätzlich zu den vorhandenen Statusbits die Bitmeldungen Code 33135 'EF2 Zustand Steuersignal Endstufenfreigabe' und Code 33136 'Zustand EF Steuersignal Endstufenfreigabe' auslesen.

Öffnen Sie die den AMK Funktionsbaustein 'STANDARD_AXIS'. Erweitern Sie in der Variablendeklaration die Deklaration des 'fbWriteID26' mit den Initialwerten.

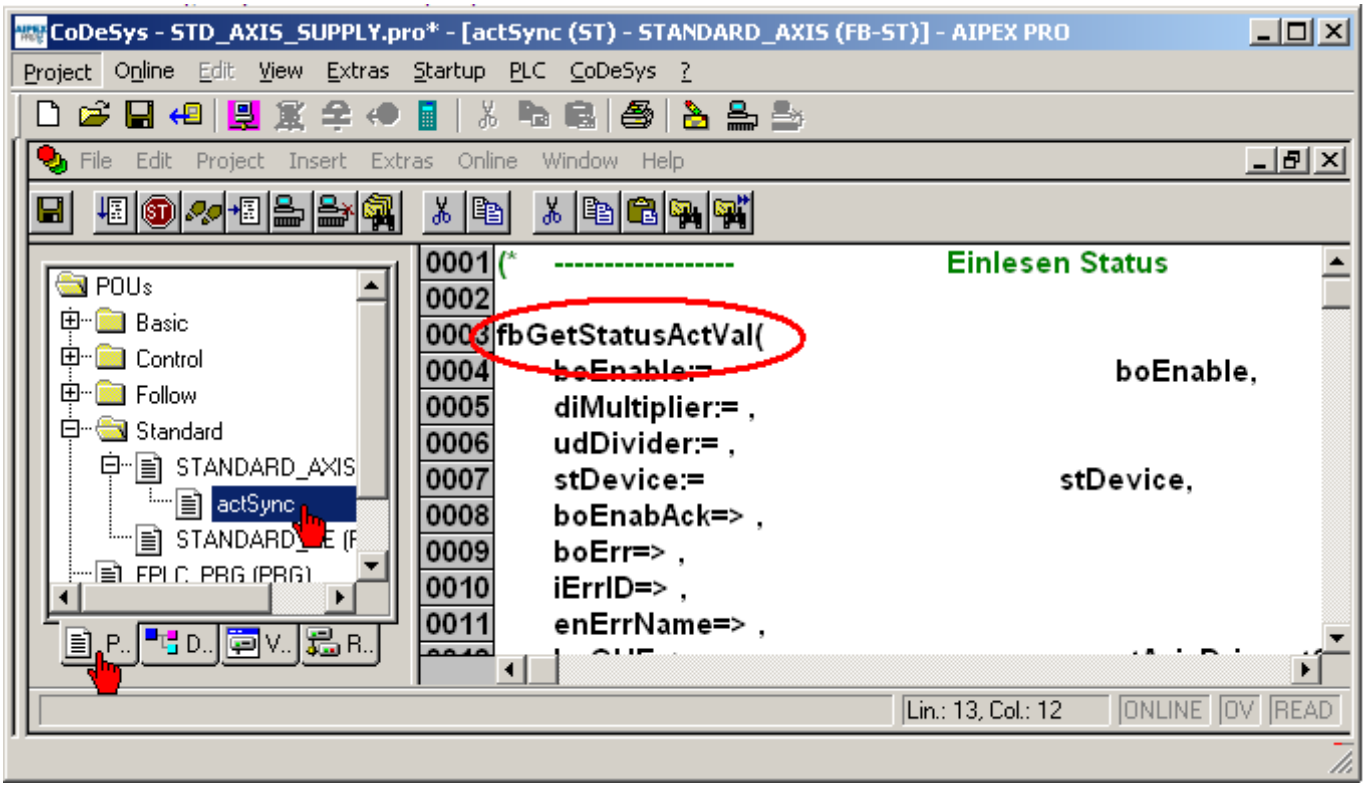


Die Struktur 'ST_STATUSBITS' können Sie beliebig erweitern.
Speicherort:
CODESYS V2
'Data types' → 'AMK_AXIS_STRUCTURE' → 'Interface'

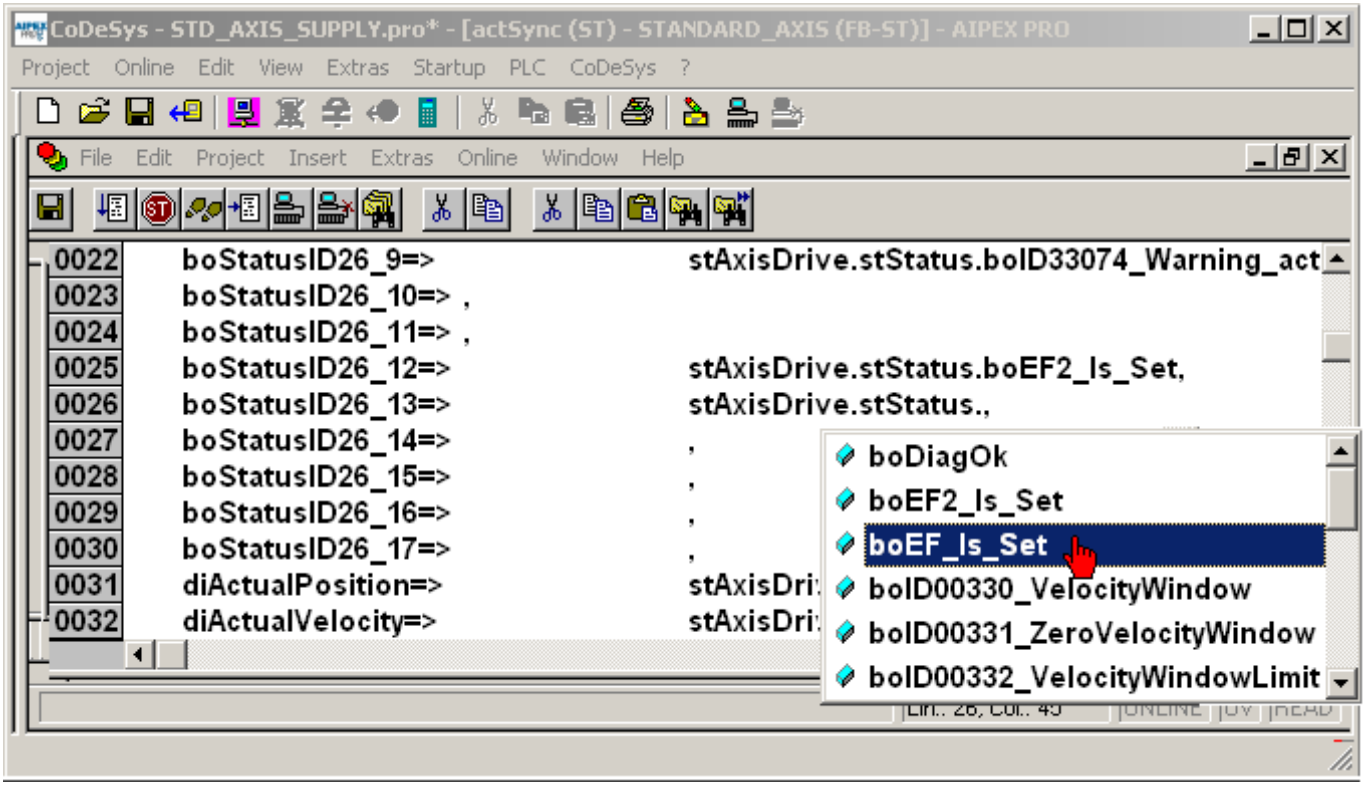
CODESYS V3
'POUs' → 'Control' → 'Standard' → 'Types' → 'AMK_AXIS_STRUCTURE' → 'Interface'



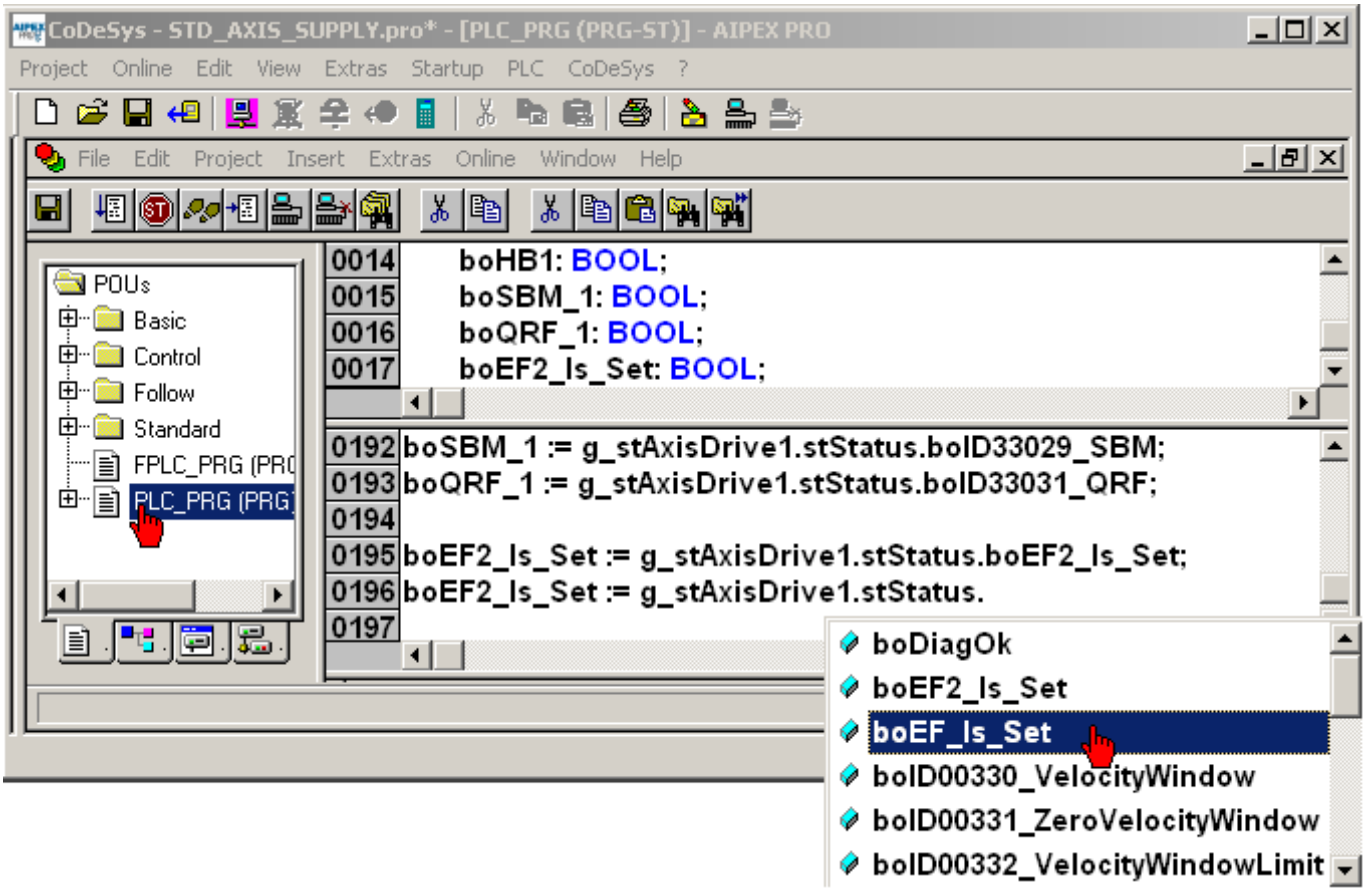
Wechseln Sie auf die synchrone Aktion des **FB '**STANDARD_AXIS**'**. Sie finden diesen unter: **'POUs' 'Standard' 'STANDARD_AXIS' 'actSync'**.
Der 'fbGetStatusActVal' kopiert unter anderem die ausgelesene Statusinformationen in die Struktur 'ST_STATUSBITS'.



Kopieren Sie das Listenelement 12 und 13 in die Struktur 'ST_STATUSBITS'.
Anschließend können Sie auf 'boEF2_Is_Set' und 'boEF_Is_Set' wie auf die bereits vorhandenen Statusbits zugreifen.



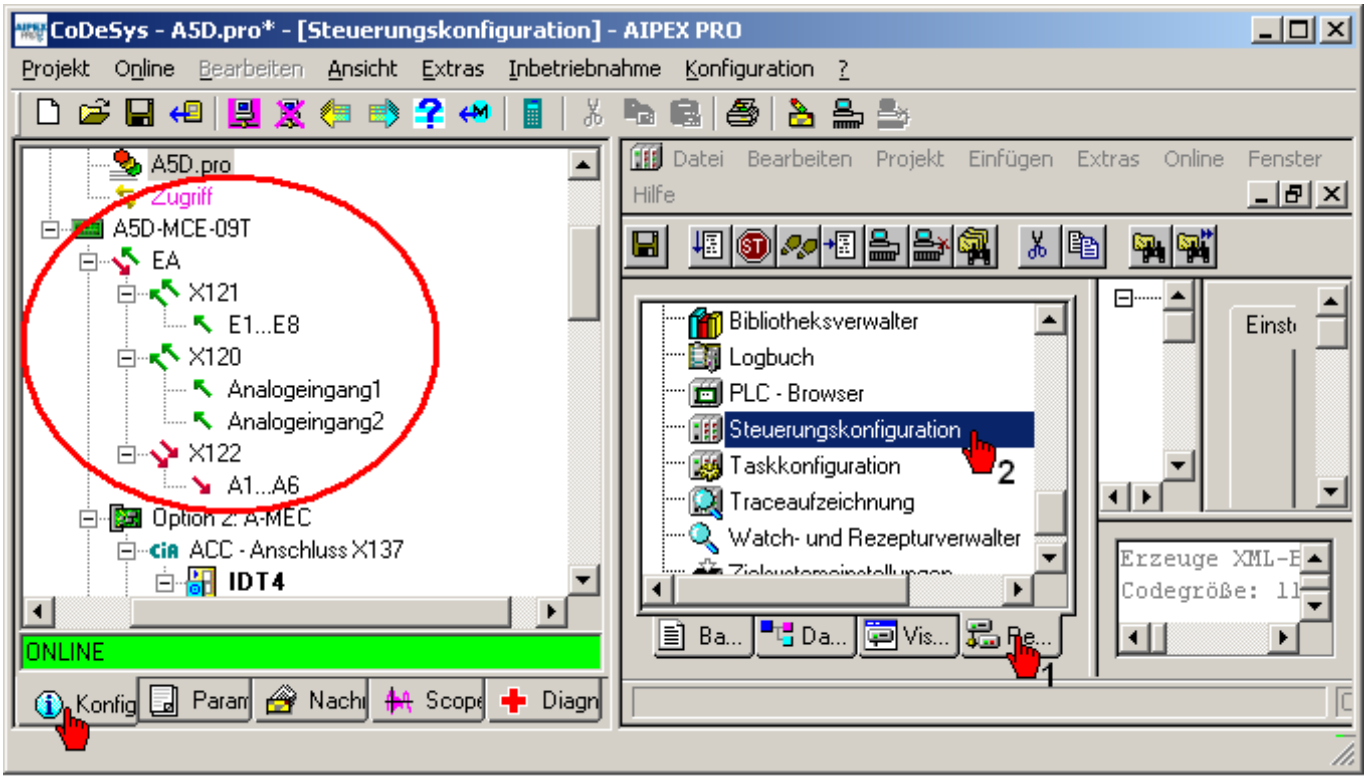
Erstellen Sie lokale Variablen im **'PLC_PRG'**. Übergeben Sie Statusbits aus der Struktur.



Nach dem ersten automatischen Schreiben (WriteID26) der Statusbits ist ein Geräteneustart erforderlich.

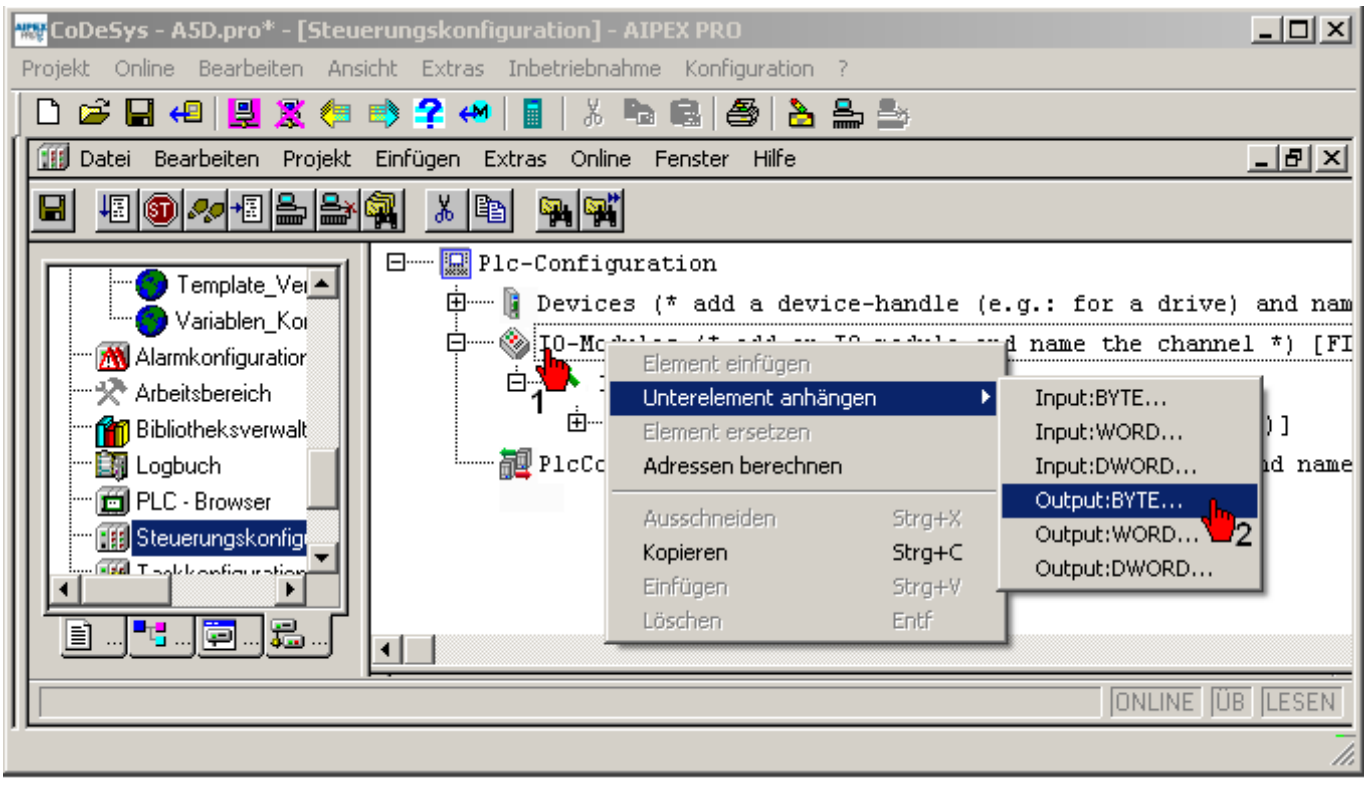
4.3.2.18.8 Zugriff auf Steuerungs EAs

Für jeden Ein- und Ausgangsblock muss in CODESYS eine Variable angelegt werden.
Öffnen Sie dazu in ihrem CODESYS Projekt unter **'Ressourcen'** (1) die **'Steuerungskonfiguration'** (2).

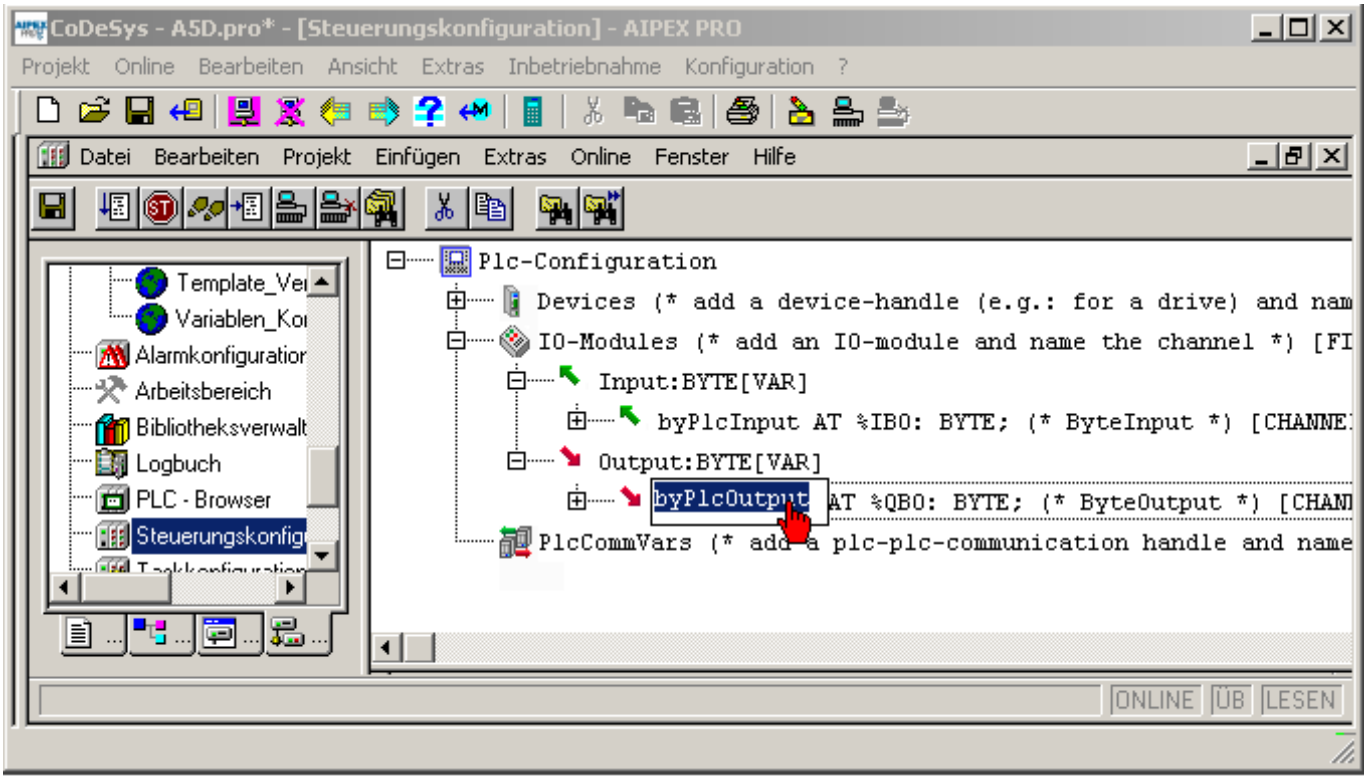


Klicken Sie mit der rechten Maustaste auf 'IO-Modules' (1), um neue Variablen einzufügen.

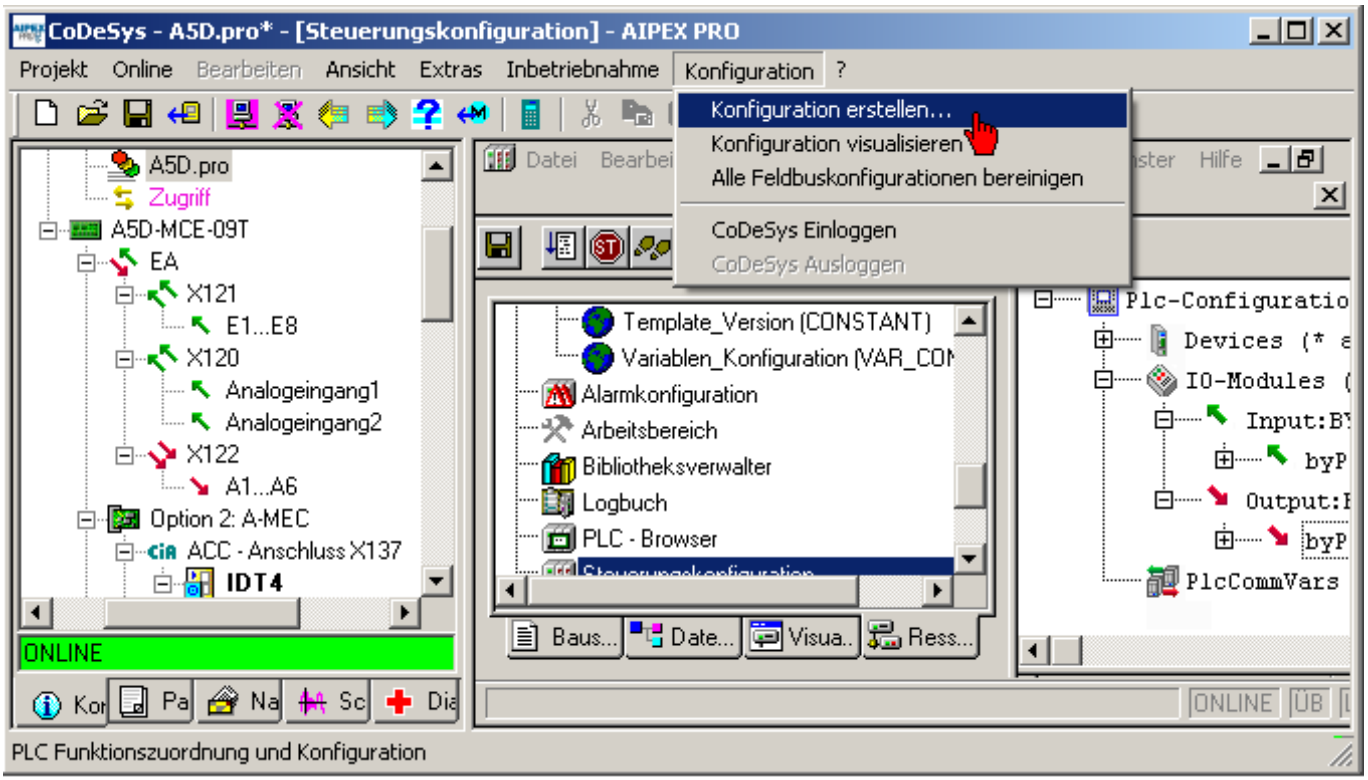
Input: Daten werden von den Antrieben in die PLC eingelesen
Output: Daten werden von der PLC auf die Antriebe geschrieben



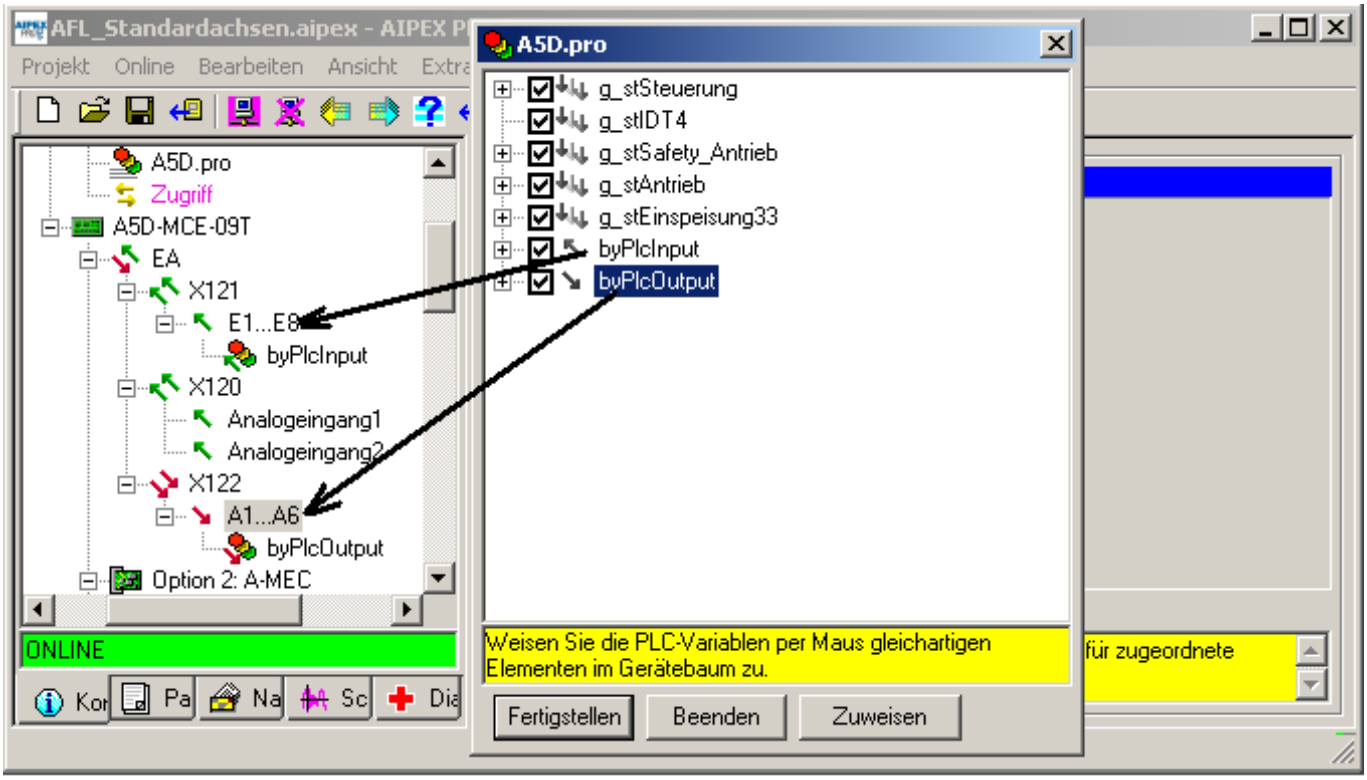
Für jedes physikalisch vorhandene Modul muss eine symbolische Variable angelegt werden.



Starten Sie anschließend die automatische Nachrichtenkonfigurationserstellung.

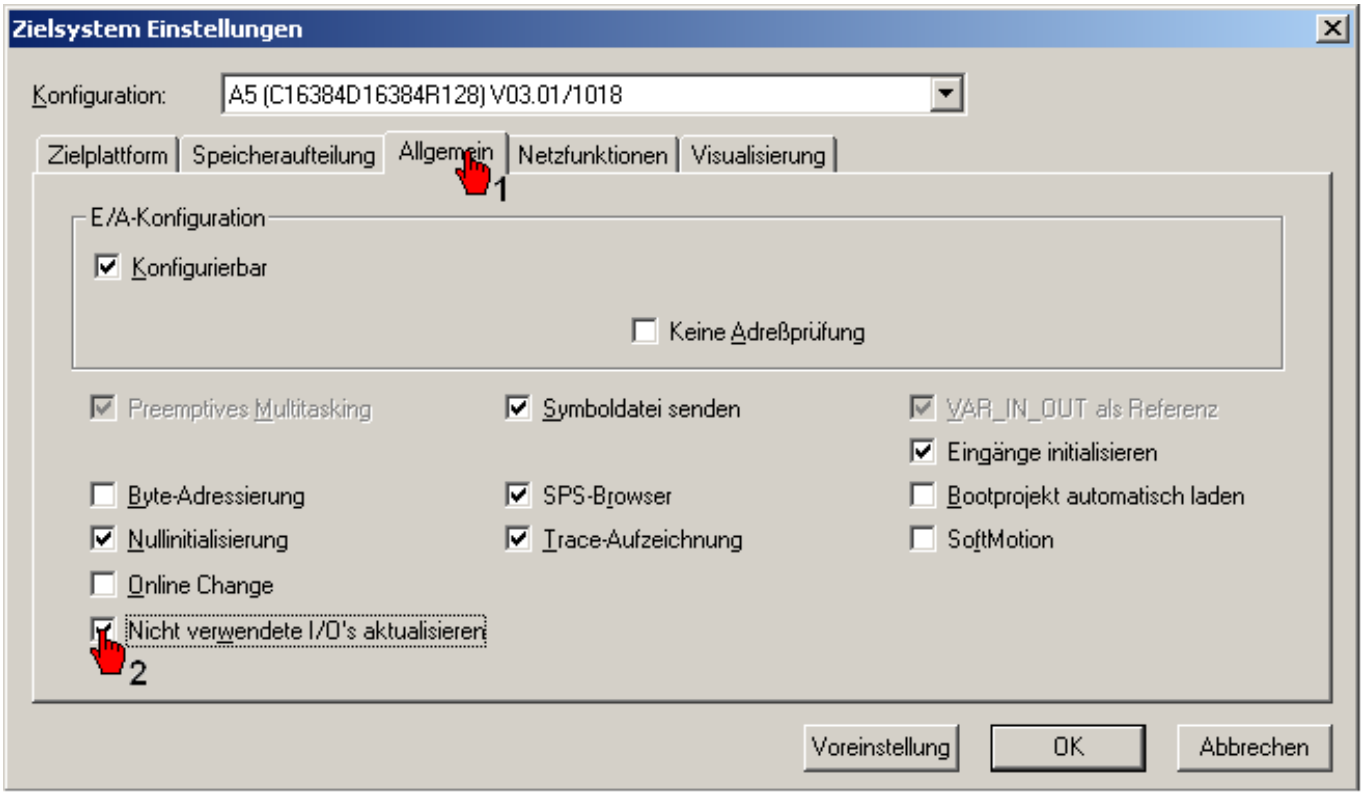


Ordnen Sie die symbolischen Variablen den physikalisch Vorhandenen Ein- und Ausgängen zu.
Nach dem Erstellen der Nachrichtenkonfiguration muss das System neu gestartet werden.



Funktionstest

Aktivieren Sie die Checkbox **'Nicht verwendete I/O's aktualisieren'** unter **'Ressourcen'** -> **'Zielsystem Einstellungen'** -> **'Allgemein'**
oder weisen Sie jeder I/O einen symbolischer Namen zu.



4.4 Visualisierung von AFL Bausteinen

Zu einem Großteil der Funktionsbausteine sind Visualisierungen vorbereitet, die die Ein- und Ausgabevariablen des Bausteins enthalten.
Damit ist eine schnelle Projektierung zum Testen der Funktionsbausteine möglich.
Die Namen der Visualisierungen sind analog zu denen der Funktionsbausteine gewählt, mit vorangestelltem 'Vi'.

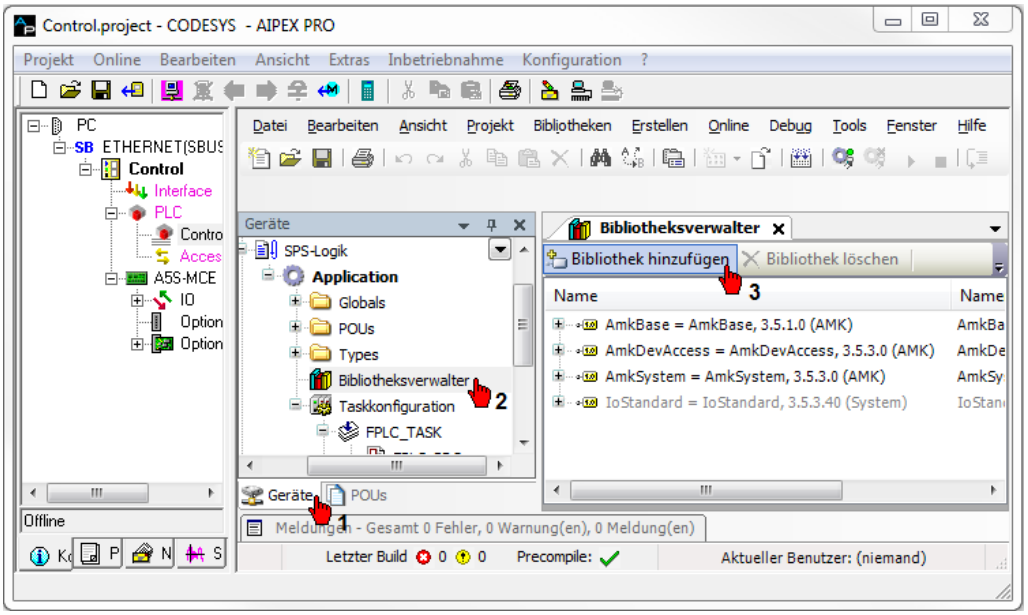
Beispiele:

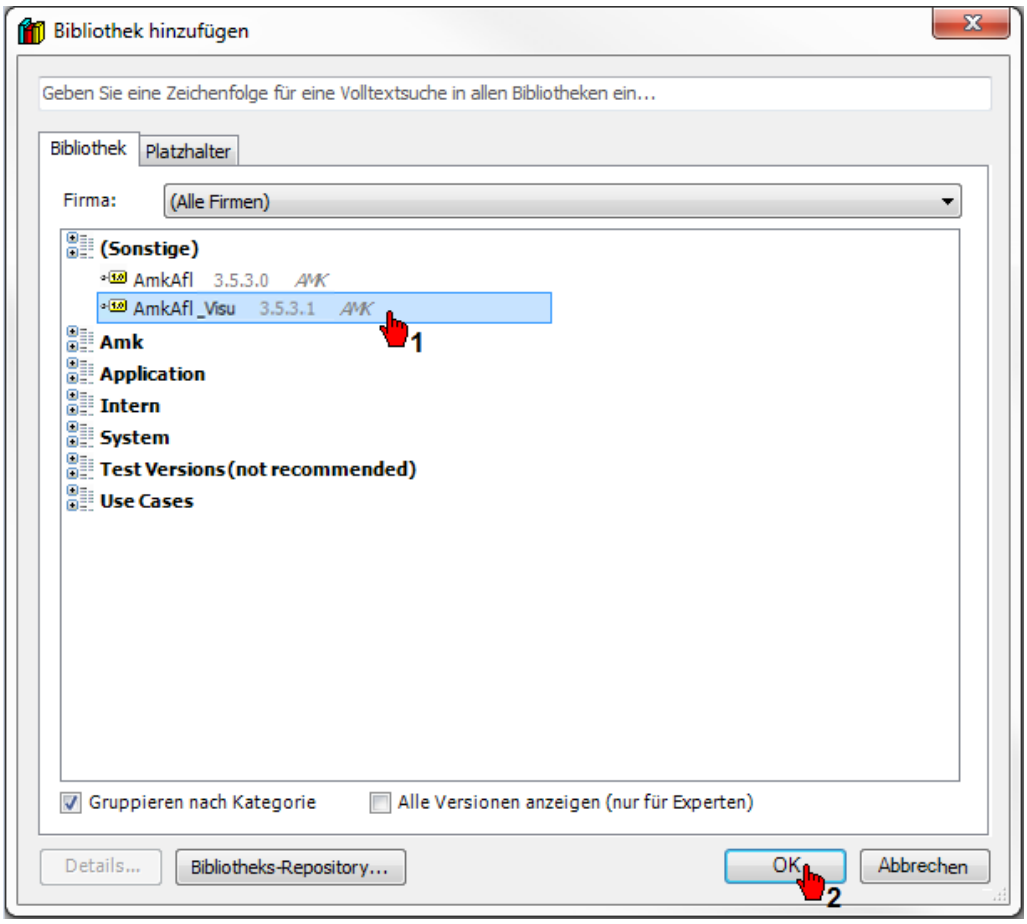
Funktionsbaustein	Visualisierung
COMPARE_2VALUES	ViCOMPARE_2VALUES
PRINT_MARK_CONTROL	ViPRINT_MARK_CONTROL
MODULO_COUNT	ViMODULO_COUNT

4.4.1 CODESYS V3

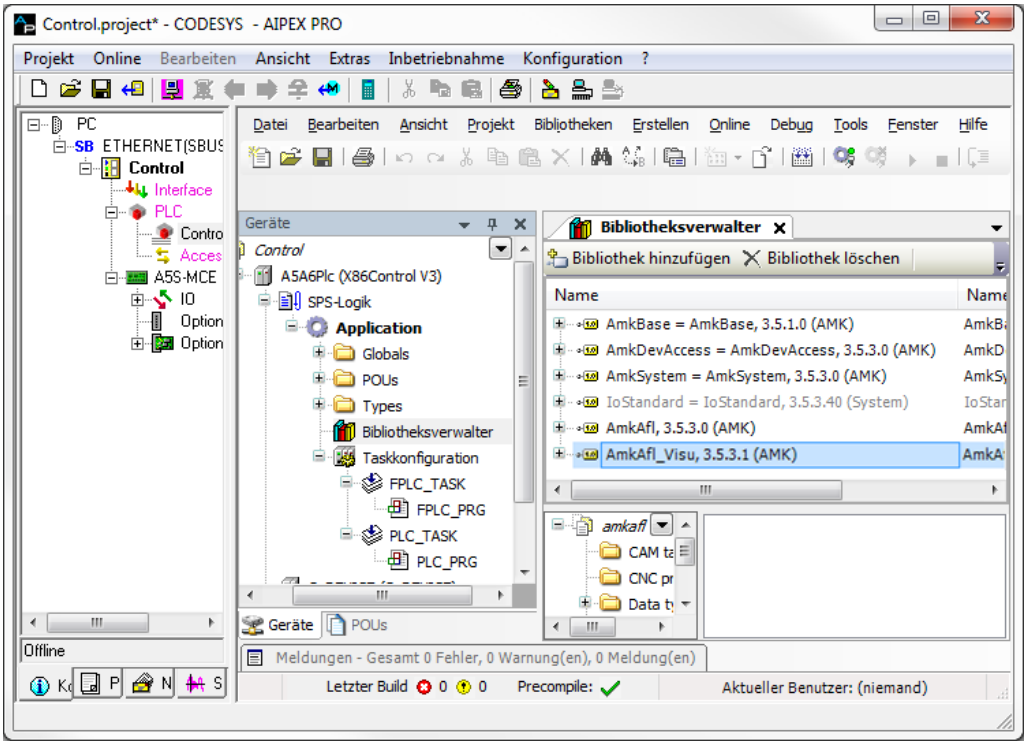
Ab der AFL Version 3.5.3.1 sind die Visualisierungen der Funktionsbausteine in einer zusätzlichen Bibliothek enthalten.

Die Bibliothek AmkAfl_Visu.library ist nicht in den Standard Templates eingebunden und muss manuell dem CODESYS Projekt hinzufügt werden.

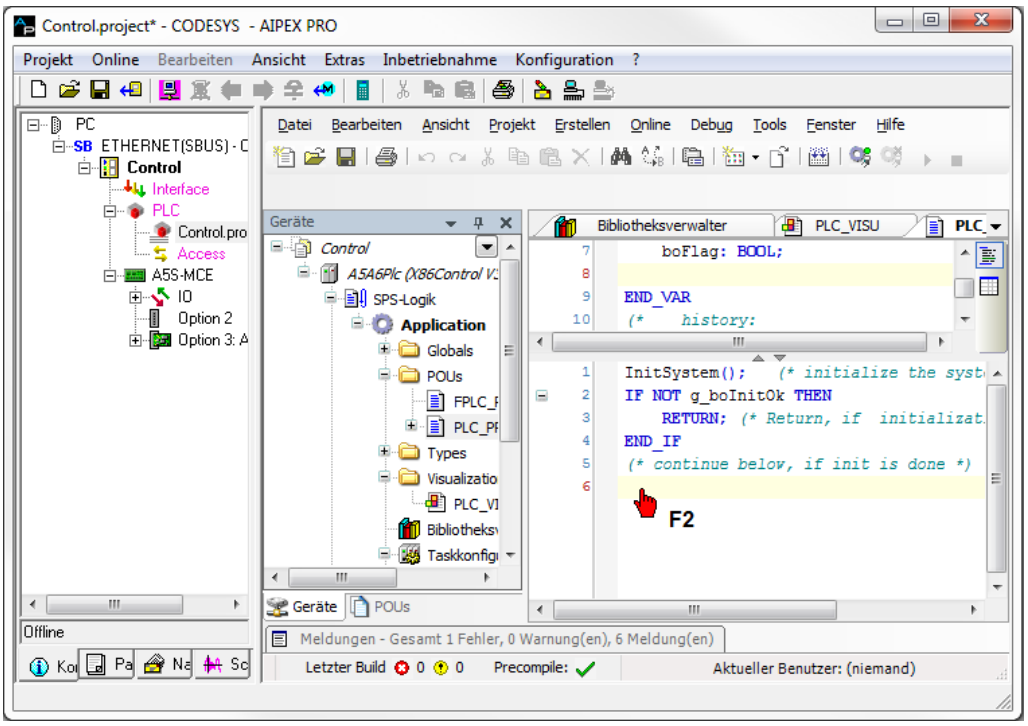


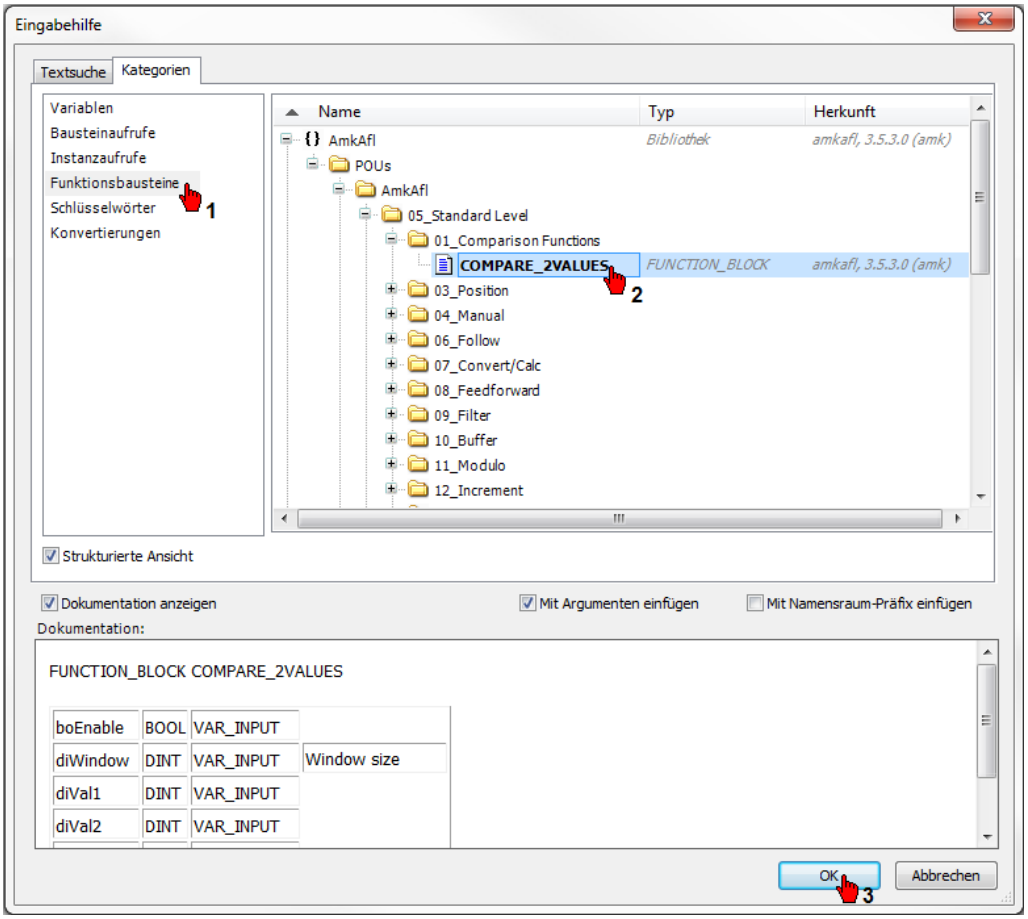


Die Bibliothek AmkAfl_Visu ist hinzugefügt, wenn sie im Bibliotheksverwalter aufgeführt ist.

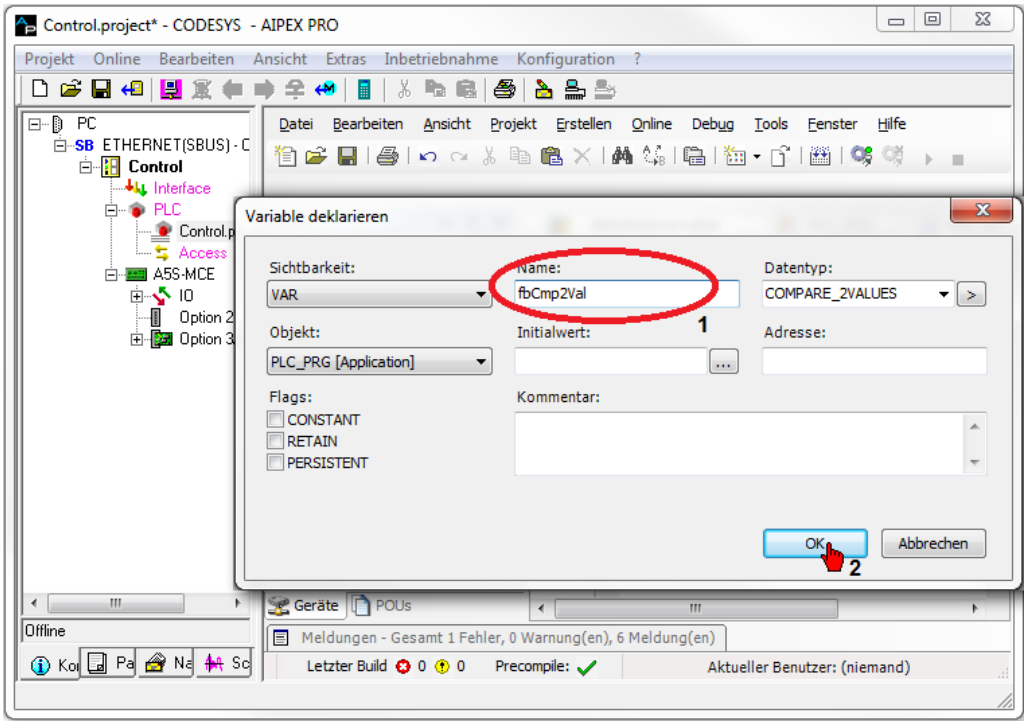


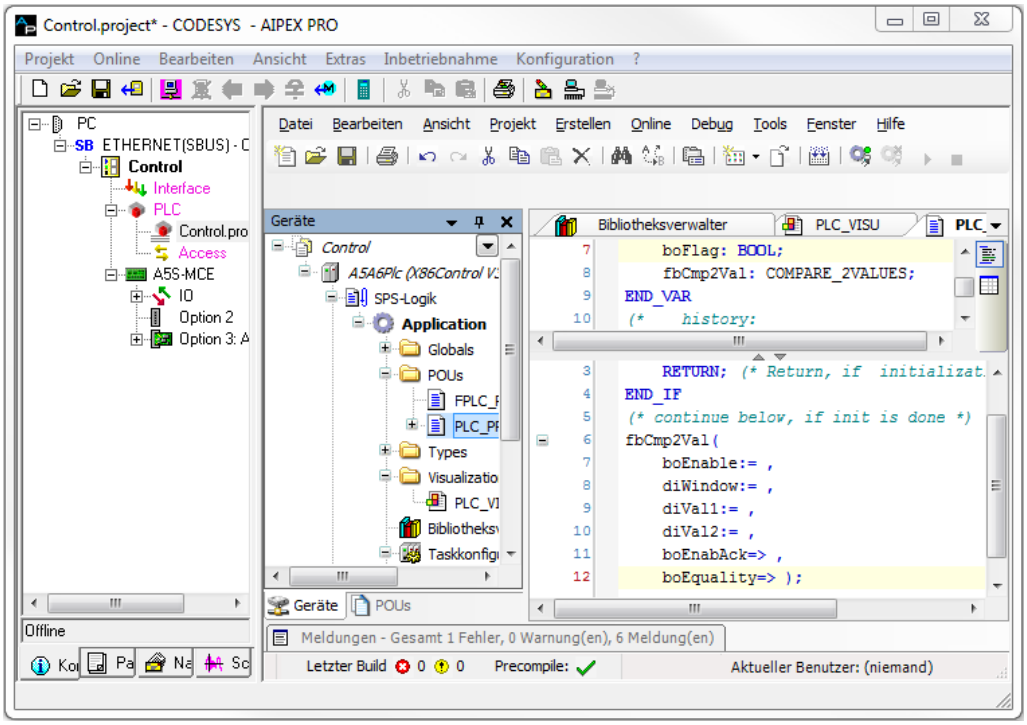
Funktionsblock einfügen (Beispiel: COMPARE_2VALUES)



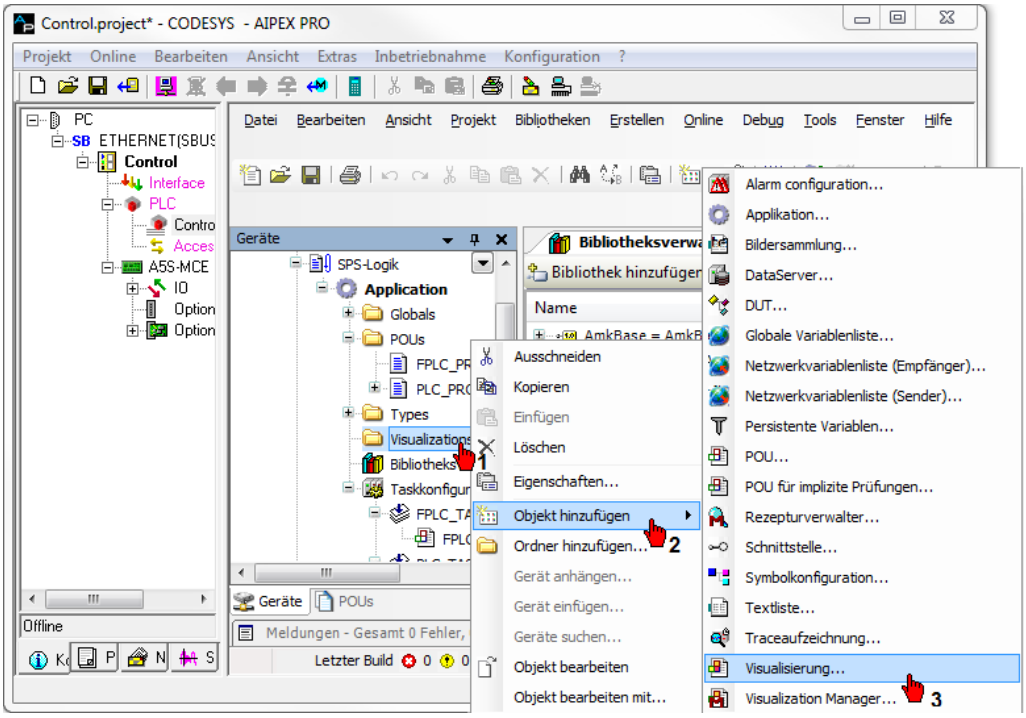


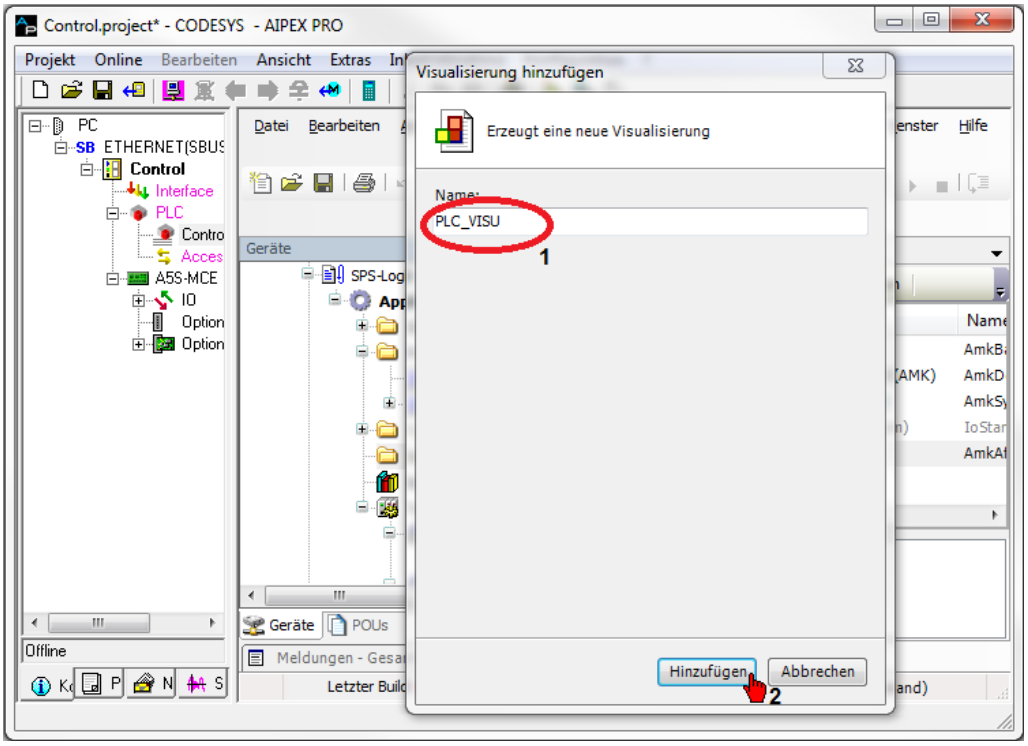
Funktionsblock deklarieren



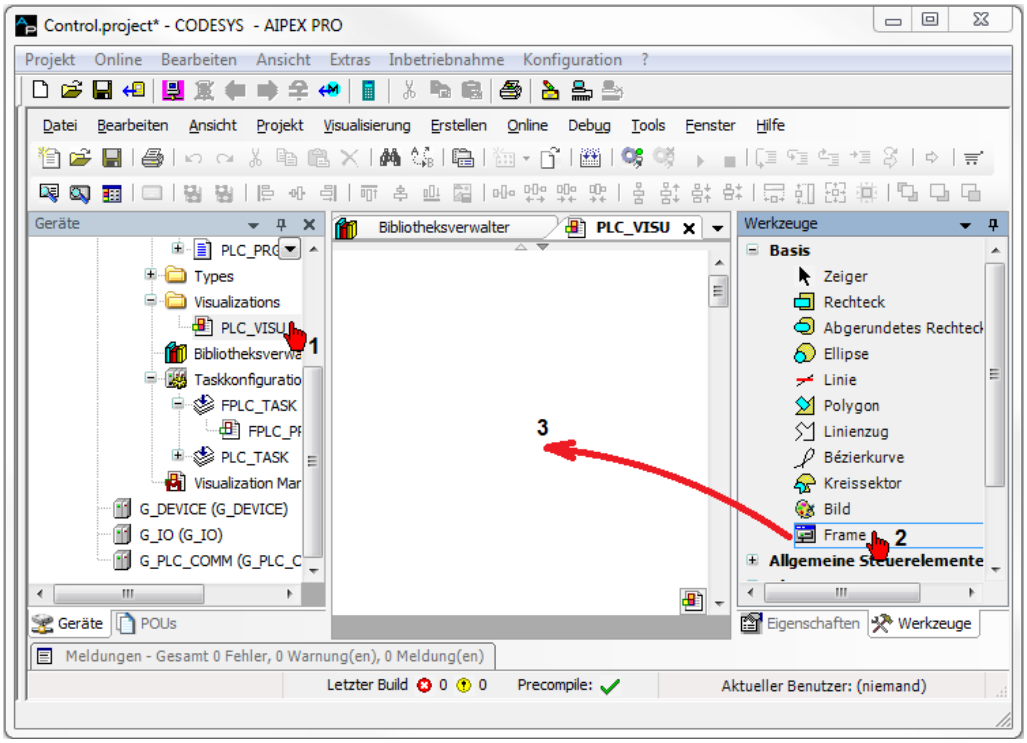


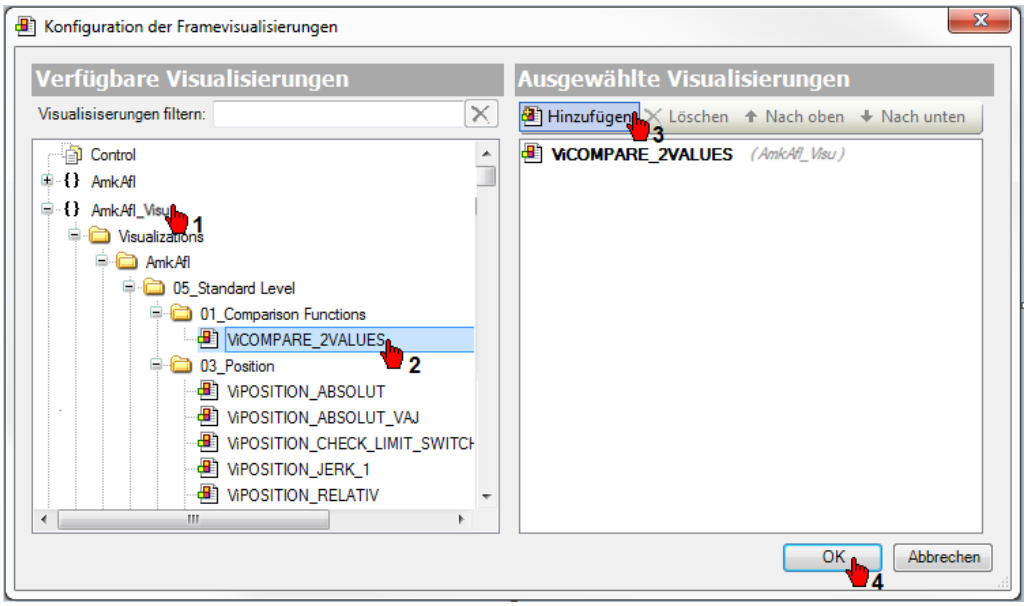
Visualisierung hinzufügen



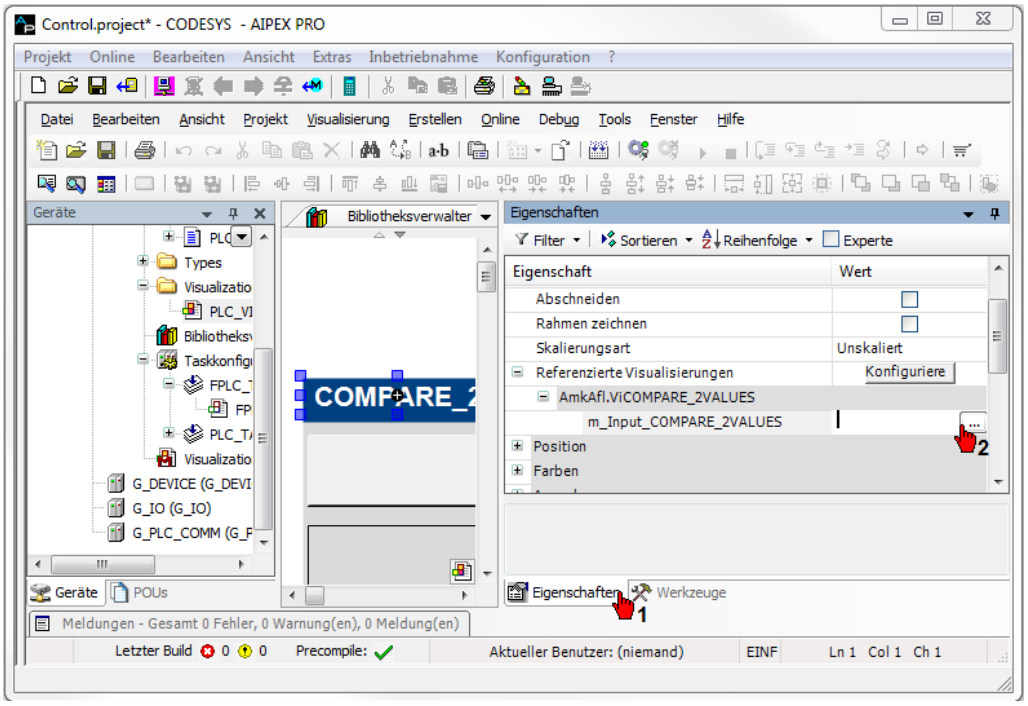


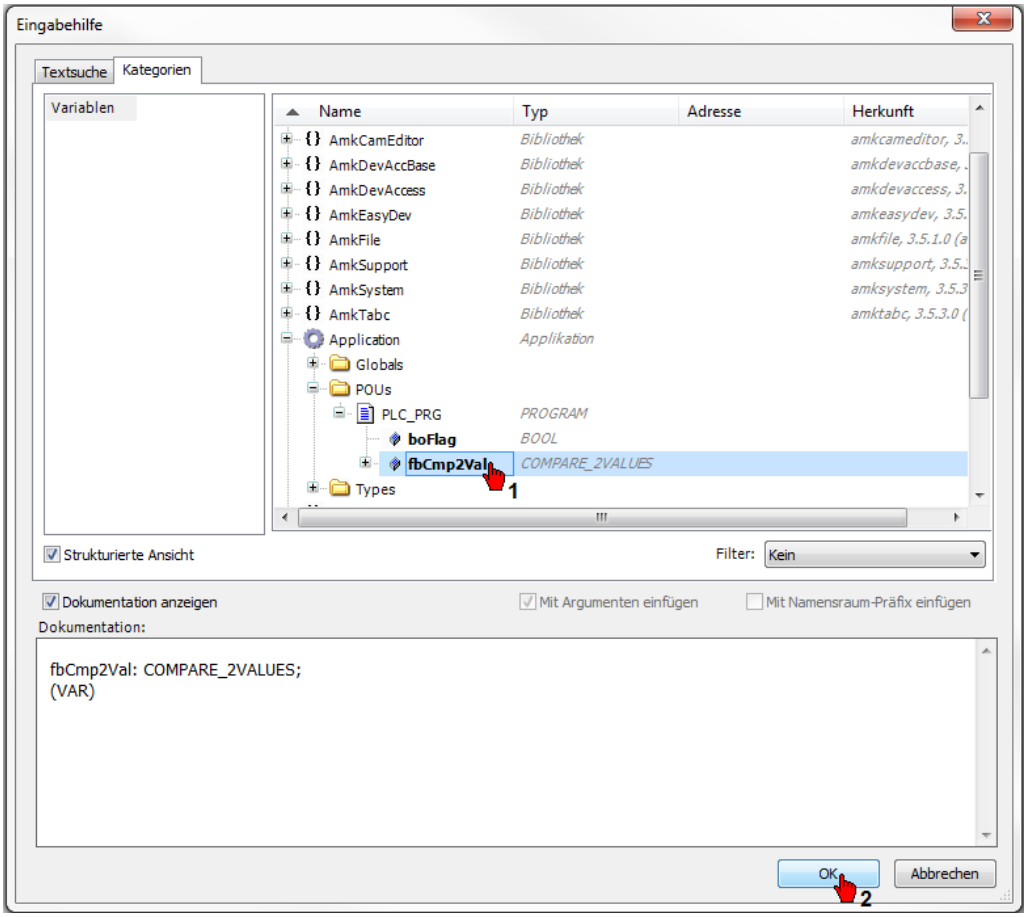
Fügen Sie den Frame ein und wählen Sie die Visualisierung aus (Beispiel: COMPARE_2VALUES)



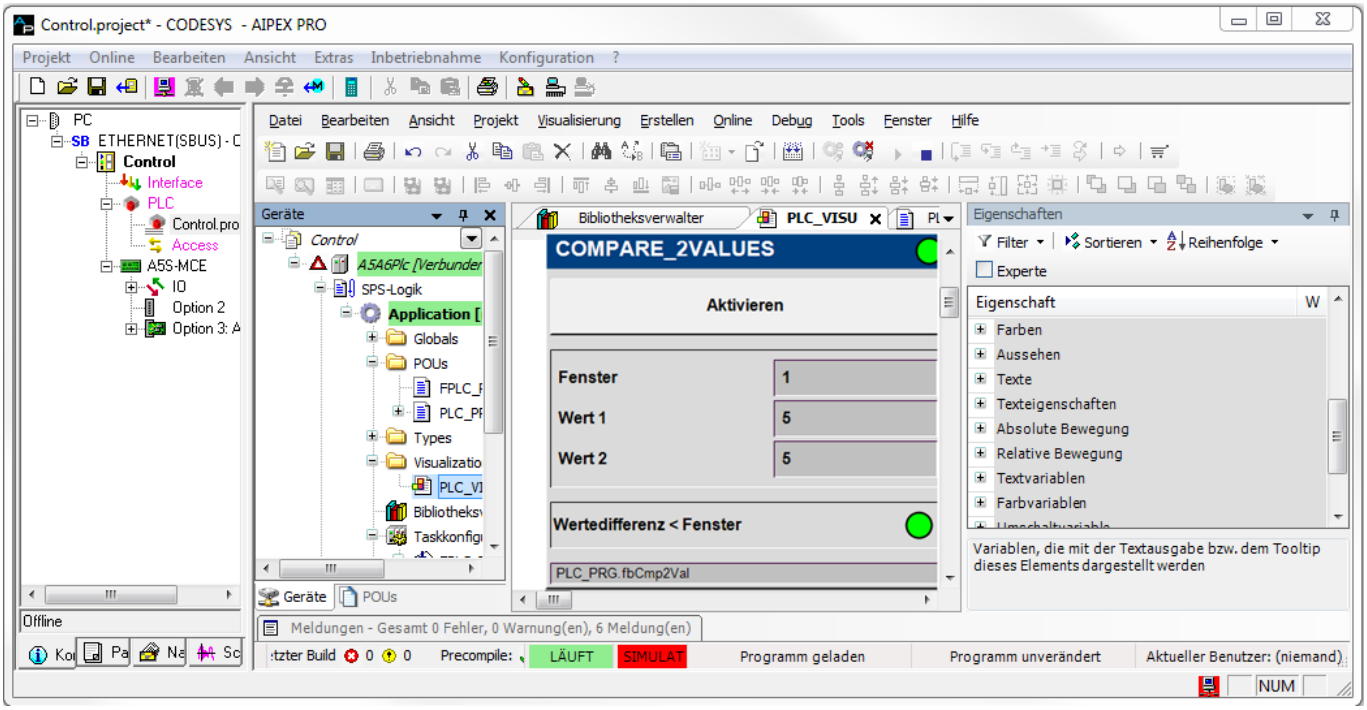


Wählen Sie den Funktionsbaustein aus, den Sie der Visualisierung zuordnen wollen.





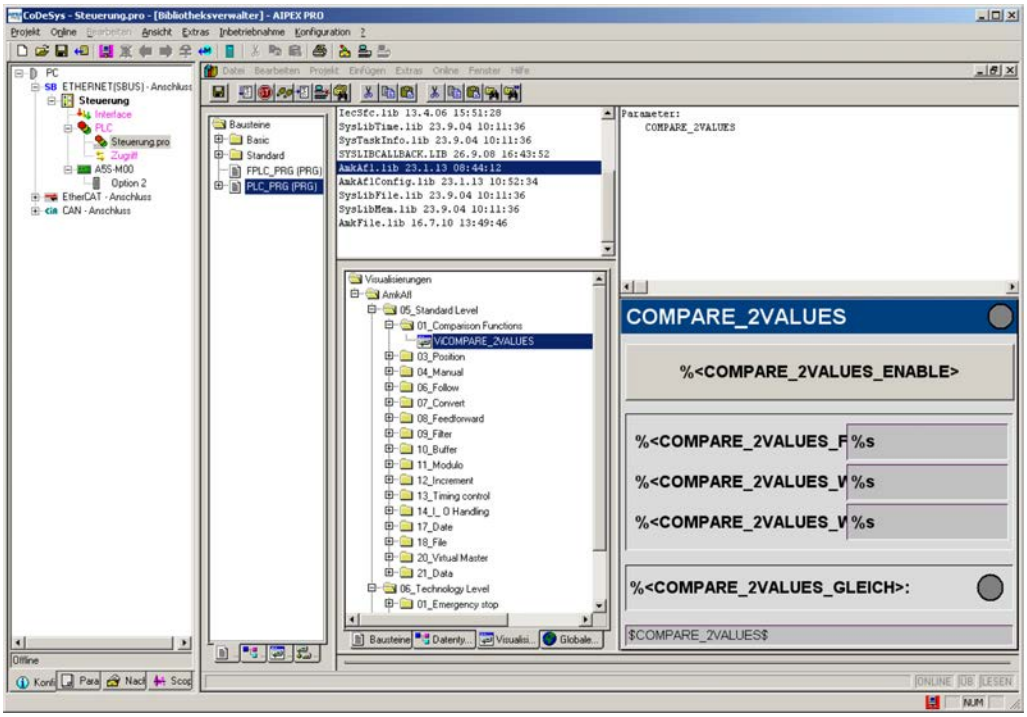
Die Werte werden erst angezeigt, wenn das CODESYS Programm gestartet wurde



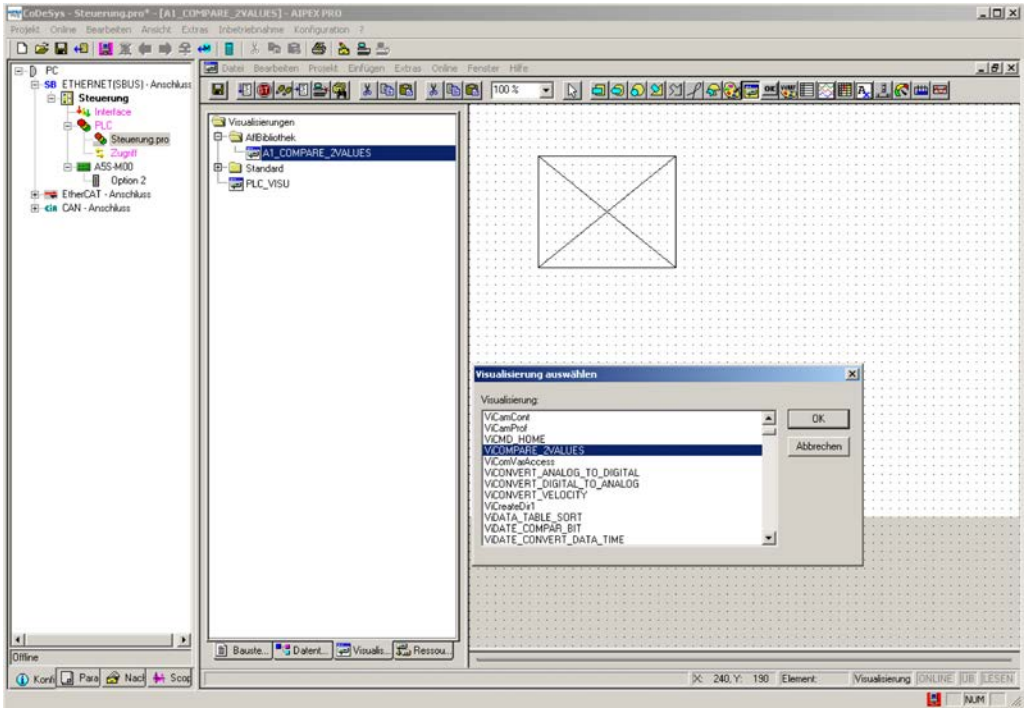
Weitere Informationen zur PLC-Programmierung unter CODESYS entnehmen Sie der Softwarebeschreibung AIPEX PRO V3 (AMK Teile-Nr. 204979).

4.4.2 CODESYS V2

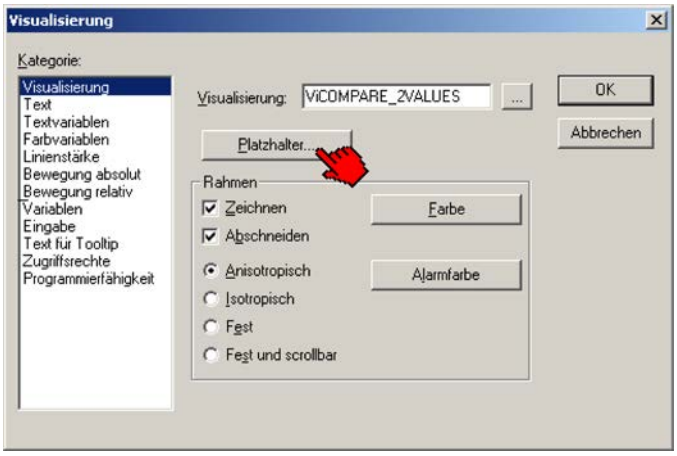
CODESYS Oberfläche Bibliotheksverwaltung: Visualisierung mit hinterlegten Variablen:



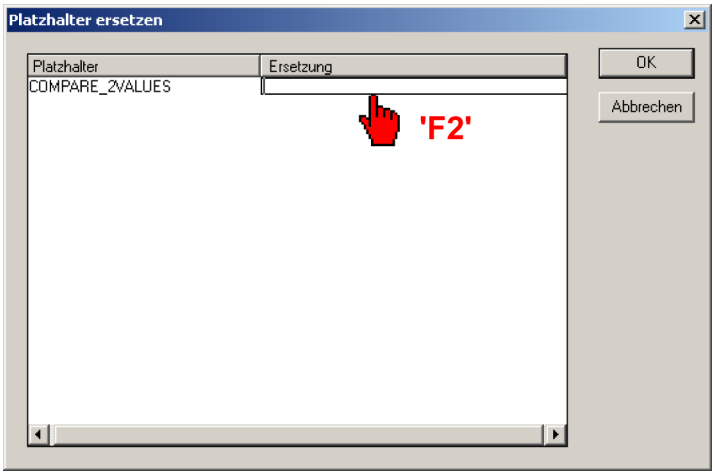
Im Visualisierungs-Editor werden die Visu-Bausteine mit dem Menü **'Einfügen'** -> **'Visualisierung'** aufgerufen.
Ziehen Sie eine Fläche auf und wählen Sie den benötigten Visu-Baustein aus.
Die endgültige Position und Größe der Visualisierung können Sie nachträglich anpassen.



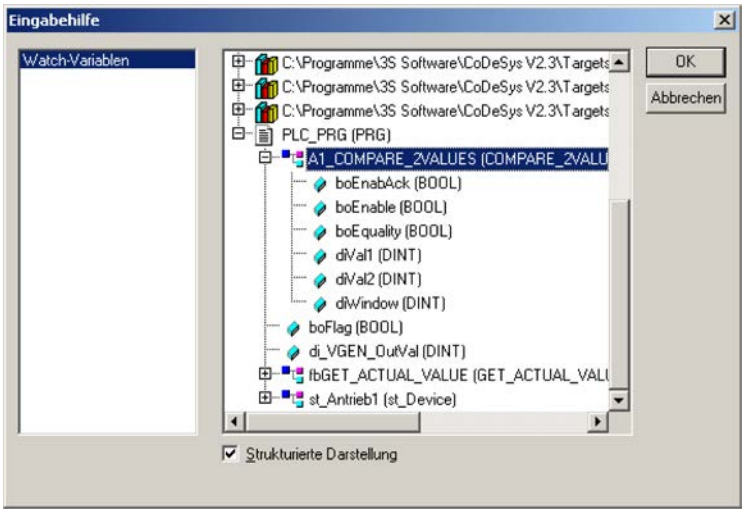
Doppelklick auf den eingefügten Visu-Baustein oder rechte Maustaste und **'Konfigurieren...'** führen zum Konfigurationsmenü.



Wählen Sie **'Platzhalter...'** an. Hier können Sie die Platzhalter ersetzen, die dem Visu-Baustein hinterlegt sind.
Klicken Sie in die Spalte 'Ersetzung' und Drücken Sie **'F2'**.



Wählen Sie den Funktionsbaustein aus, den Sie der Visualisierung zuordnen wollen.



Bestätigen Sie alle Eingabefenster mit **'OK'**.

Glossar

A

A1
Analogeingang 1

A-SIP
EtherNET/IP Slave Schnittstelle

A-SCN
CAN / ACC Bus Slave Schnittstelle

A-SPB
Profibus DP Slave Schnittstelle

A-SEC
EtherCAT Slave Schnittstelle

AIO
AMKAMAC I/O-Klemmen

A-MEC
EtherCAT Master Schnittstelle

Ax-VIS
Webvisualisierung

Ax-PNC
Numerical Control Motion

Ax-PCO
PLCopen

ATF
AMK Tool Flasher (PC Software um Firmware auf Geräte zu überspielen)

ASCII
American Standard Code for Information Interchange

A4 / A5 / A6
AMKAMAC Steuerung A4 / A5 / A6

ACC
AMK CAN Communication (CAN-Bus Schnittstelle mit Standard CANopen Protokoll DS301 und zusätzlichem Hardware Synchronisationssignal)

actSync
Synchrone Aktion eines asynchronen Funktionsbausteins

ADB
AMK Datenbank - Datei im XML-Format mit Informationen über alle AMK Parameter

AIPAR
AMKASYN Inbetriebnahme Parametriersoftware

AIPEX
AMK Parametrier- und Inbetriebnahmeexplorer (PC Software): Programmieren, Parametrieren, Konfigurieren, Diagnose, Oszilloskop, Statusinformationen

B

BIN
Binär

BE
Binäreingang

C

CAN
Controller Area Network

CRC
Cyclic redundancy check (Prüfsumme)

D

DEZ
Dezimal

E

EMV
Elektromagnetische Verträglichkeit

EnDat 2.2
Motorgeber Schnittstellenprotokoll der Firma Heidenhain

EnDat 2.1
Motorgeber Schnittstellenprotokoll der Firma Heidenhain

E-Geber
Absolutwertgeber singleturn, EnDAT 2.1 mit zusätzlicher Sinus- und Cosinusspur

EF2
Endstufenfreigabe

EF
Endstufenfreigabe

EDS
Electronic Data Sheet

EtherCAT
Echtzeit-Ethernet Bus

E/A
Ein- und Ausgänge

F

F-Geber
Absolutwertgeber multiturn, EnDAT 2.1 mit zusätzlicher Sinus- und Cosinusspur

FB
Funktionsbaustein

FIFO
First in - first out (Erster rein - Erster raus): Speicher, der die gespeicherten Werte in der Reihenfolge ausgibt, in der sie eingespeichert wurden

Firmware
Betriebssystem oder Betriebssoftware, die AMK werkseitig in das Gerät lädt

FL
Kommando Fehler löschen (Bewirkt einen erneuten Systemhochlauf)

FPLC_PRG
Echtzeit-Task; zum Gerätezyklus synchronisiert

freilaufende Task
z.B. PLC_PRG; ist nicht synchronisiert mit dem Reglertakt im Antrieb.

FSoE
Fail-Safe over EtherCAT (Ausfallsicherer Signalaustausch über EtherCAT)

G

g_yourDevice
Symbolischer Name eines Gerätes im SPS Projekt. Der Name wird in der CoDeSys-Steuerungskonfiguration: Devices angelegt

H

HEX
Hexadezimal, 0x...

I

iSA-PNC
Numerical Control Motion

iSA-VIS
Webvisualisierung

ihXT
AMASMART Servomotor mit integriertem Wechselrichter

iX
AMKASMART Servowechselrichter

I-Geber
Inkrementalgeber; Optischer Geber mit Sinus- und Cosinusspur und Nullimpuls

iDT
AMKASMART Servomotor mit integriertem Wechselrichter

ID
Parameter-Identnummern nach SERCOS Standard

iC
AMKASMART Servoumrichter mit Einspeisung

iSA-PCO
PLCopen

iSA
AMKASMART Steuerung mit Einspeisung

K

KWD
AMKASYN Doppelwechselrichter zur Regelung von 2 Motoren

KE
AMKASYN Kompakteinspeisung mit Blockrückspeisung

KE/KW
Modulares AMKASYN Antriebssystem (Besteht aus Kompakteinspeisung KE, Kompaktwechselrichtern KW mit Reglerkarte und ggf. Optionskarten)

KEN
AMKASYN Kompakteinspeisung ohne Rückspeisung

KU
AMKASYN Kompaktumrichter

KWZ
AMKASYN Zweiachswechselrichter zur Regelung von 2 Motoren

KW-Rxx
AMKASYN Reglerkarte, zum Einsatz in Kompaktwechselrichtern

L

LAN
Local Area Network

M

MDB
Motordatenbank Datei, enthält Informationen über alle AMK Motorparameter

Modulo
Modulo-Verarbeitung der Lagesoll- und -istwerte

MyTerm

N

NK
Nocken, Nockenschalter

P

Parameter
Identnummern nach SERCOS Standard

PDK_xxxxxx_abcdefgh
Produktdokumentation; xxxxxx - AMK Teile-Nr. , abcdefgh - Titel

PDO
Prozess Data Object

P-Geber
Absolutwertgeber singleturn, EnDAT 2.2 light

PGT
Peripherie Grund Takt; bildet den Aufrufzyklus im Grundgerät, auf den die Antriebsregler synchronisiert sind. (Die Zykluszeit entspricht der Identnummer ID2)

PLC_PRG
Freilaufende Task; nicht zum Gerätetezyklus synchronisiert

POU
Program Organisation Unit, Programm-Organisationseinheit, PLC Programmelemente der Typen Programm, Funktion oder Funktionsblock

PRG
Programm

Q

QRF
Quittierung Reglerfreigabe, Antrieb wird in der aktiven Betriebsart geregelt

QBR
Quittierung Motorhaltebremse

QUE
Quittierung Umrichter EIN: Statussignal zeigt an, dass der Zwischenkreis geladen wurde

QFL
Quittierung Fehler löschen, das Kommando Fehler Löschen wurde ausführt

Q-Geber
Absolutwertgeber multiturn, EnDAT 2.2 light

R

RO
Read Only - Lesezugriff

RF
Kommando Reglerfreigabe; der Antrieb wird bestromt und abhängig von der eingestellten Betriebsart geregelt (Die Reglerfreigabe kann nur gesetzt werden, wenn das Gerät fehlerfrei ist (SBM=TRUE) und die Quittierung Umrichter EIN (QUE) gesetzt ist. Ist die Reglerfreigabe gesetzt, wird die Quittierung Reglerfreigabe (QRF) ausgegeben)

S

Standard
Werkseinstellung, voreingestellt

SBM
System Bereit Meldung; zeigt an, dass das Gerät fehlerfrei ist. (Im Fehlerfall wird SBM rückgesetzt)

STO
Safe torque off (Sicher abgeschaltetes Moment). Sicherheitsfunktion nach DIN EN 61800-5-2

SERCOS
Genormte digitale Schnittstelle zur Kommunikation zwischen Steuerungen und Feldbusteilnehmern

SEEP
Geräteinterner Speicher, Serielles EEPROM

SBUS
AMK-spezifisches Protokoll für serielle Schnittstellen

SafePMT
SichererParametereditor

S-Geber
Absolutwertgeber singleturn, RS485 Hiperface mit Sinus- und Cosinusspur

T

TwinCAT
Automatisierungssoftware

T-Geber
Absolutwertgeber multiturn, RS485 Hiperface mit Sinus- und Cosinusspur

U

UE
Kommando Umrichter EIN ; Steuersignal mit dem der Zwischenkreis (z.B. im KE) geladen wird. Umrichter EIN kann nur gesetzt werden, wenn das Gerät fehlerfrei ist (SBM=TRUE). Ist der Zwischenkreis aufgeladen, wir die Quittierung Umrichter EIN (QUE) ausgegeben

Ihre Meinung zählt!

Mit unseren Dokumentationen möchten wir Sie im Umgang mit den AMKmotion Produkten bestmöglich unterstützen.
Daher sind wir ständig bestrebt, unsere Dokumentationen zu optimieren.
Ihre Kommentare oder Anregungen sind für uns immer interessant.
Nehmen Sie sich kurz Zeit und beantworten Sie unsere Fragen. Bitte schicken Sie anschließend eine Kopie dieser Seite an AMKmotion zurück.

E-Mail: Documentation@amk-motion.com
oder
Fax-Nr.: +49 7021/50 05-199

Vielen Dank für Ihre Mithilfe.
Ihr AMKmotion Dokumentationsteam

1. Wie sind Sie mit der Optik unserer Dokumentationen zufrieden?

   (1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

2. Ist der Inhalt gut gegliedert?

   (1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

3. Ist der Inhalt verständlich dokumentiert?

   (1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

4. Haben Sie Themen in der Dokumentation vermisst?

   (1) nein (2) ja, welche:

5. Fühlen Sie sich bei AMKmotion insgesamt gut betreut?

   (1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

AMKmotion GmbH + Co KG
Telefon: +49 7021/50 05-0, Telefax: +49 7021/50 05-199
E-Mail: info@amk-motion.com
Homepage: www.amk-motion.com