

Rexroth IndraMotion MLC 12VRS Technologie-Bibliotheken

R911333867
Ausgabe 02

Bibliothek



Titel	Rexroth IndraMotion MLC 12VRS Technologie-Bibliotheken											
Art der Dokumentation	Bibliothek											
Dokumentations-Type	DOK-MLC***-TF*LIB**V12-LI02-DE-P											
Interner Ablagevermerk	RS-ed9bd9627e6498430a6846a000f81475-2-de-DE-11											
Zweck der Dokumentation	Diese Dokumentation beschreibt die Funktionsbausteine, Funktionen und Datentypen der Bibliotheken • ML_TechBase.compiled-library • ML_TechMotion.compiled-library • RMB_TechCam.compiled-library sowie Bibliotheken zur Wicklerfunktionalität, Registerreglerfunktionalität und Querschneiderfunktionalität.											
Änderungsverlauf	<table><tr><th>Ausgabe</th><th>Stand</th><th>Bemerkung</th></tr><tr><td>Ausgabe 01</td><td>03.2011</td><td>Erstausgabe für 12VRS</td></tr><tr><td>Ausgabe 02</td><td>08.2011</td><td>Ergänzungen, Korrekturen</td></tr></table>			Ausgabe	Stand	Bemerkung	Ausgabe 01	03.2011	Erstausgabe für 12VRS	Ausgabe 02	08.2011	Ergänzungen, Korrekturen
Ausgabe	Stand	Bemerkung										
Ausgabe 01	03.2011	Erstausgabe für 12VRS										
Ausgabe 02	08.2011	Ergänzungen, Korrekturen										
Schutzvermerk	© Bosch Rexroth AG 2011 Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts wird nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zum Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten (DIN 34-1).											
Verbindlichkeit	Die angegebenen Daten dienen allein der Produktbeschreibung und sind nicht als zugesicherte Eigenschaften im Rechtssinne zu verstehen. Änderungen im Inhalt der Dokumentation und Liefermöglichkeiten der Produkte sind vorbehalten.											
Herausgeber	Bosch Rexroth AG Bgm.-Dr.-Nebel-Str. 2 ■ D-97816 Lohr a. Main Telefon +49 (0)93 52/ 40-0 ■ Fax +49 (0)93 52/ 40-48 85 http://www.boschrexroth.com/ Systementwicklung Automation Motion Logic Control, DT (KaWa)											
Hinweis	Diese Dokumentation ist auf chlorfrei gebleichtem Papier gedruckt.											

Inhaltsverzeichnis

	Seite
1 Zu dieser Dokumentation.....	9
1.1 Gültigkeit der Dokumentation.....	9
1.2 Gliederung der Dokumentation.....	9
1.3 Erforderliche und ergänzende Dokumentationen.....	10
1.4 Darstellung von Informationen.....	14
1.4.1 Sicherheitshinweise.....	14
1.4.2 Verwendete Symbole.....	14
1.4.3 Bezeichnungen und Abkürzungen.....	15
2 Wichtige Gebrauchshinweise.....	17
2.1 Bestimmungsgemäßer Gebrauch	17
2.1.1 Einführung.....	17
2.1.2 Einsatz- und Anwendungsbereiche.....	17
2.2 Nicht-bestimmungsgemäßer Gebrauch.....	18
3 Sicherheitshinweise für elektrische Antriebe und Steuerungen.....	19
3.1 Begriffsdefinitionen.....	19
3.2 Grundsätzliche Hinweise.....	20
3.2.1 Benutzung und Weitergabe der Sicherheitshinweise.....	20
3.2.2 Voraussetzungen für den sicheren Gebrauch.....	20
3.2.3 Gefahren durch falschen Gebrauch.....	22
3.3 Gefahrenbezogene Hinweise.....	23
3.3.1 Schutz gegen Berühren elektrischer Teile und von Gehäusen.....	23
3.3.2 Schutzkleinspannung als Schutz gegen elektrischen Schlag	24
3.3.3 Schutz vor gefährlichen Bewegungen.....	25
3.3.4 Schutz vor magnetischen und elektromagnetischen Feldern bei Betrieb und Montage.....	26
3.3.5 Schutz gegen Berühren heißer Teile.....	27
3.3.6 Schutz bei Handhabung und Montage.....	27
3.3.7 Schutz beim Umgang mit Batterien.....	27
3.3.8 Schutz vor unter Druck stehenden Leitungen.....	28
3.4 Erläuterung der Signalwörter und der Signalgrafik.....	28
4 Übersicht.....	31
4.1 Bibliotheken und Zielsysteme.....	31
4.2 Besonderheiten von IndraMotion MLD.....	31
4.2.1 Allgemeines.....	31
4.2.2 Antriebsfirmware, Funktionspakete.....	31
5 ML_TechBase.library.....	33
5.1 Einführung und Übersicht.....	33
5.2 Funktionsbausteine zur Anwendung "Vertippen von Variablen"	35
5.2.1 Einführung und Übersicht.....	35
5.2.2 MB_ContinuousAdjustType01.....	38

Inhaltsverzeichnis

	Seite
5.2.3 MB_ContinuousAdjustType02.....	42
5.2.4 MB_IncrementalAdjustType01.....	46
5.2.5 Parametrierung der Funktionsbausteine zum Vertippen von Variablen.....	48
5.3 Funktionsbaustein für Zweipunktregler	49
5.3.1 Einführung.....	49
5.3.2 ML_TwoPosCtrlType01.....	49
5.4 Funktionsbausteine für mathematische Funktionen.....	51
5.4.1 Einführung.....	51
5.4.2 ML_InterpolationLinear.....	51
5.4.3 ML_MaxValue.....	53
5.5 Funktionsbaustein zur Anwendung "Nockenschaltwerk".....	54
5.5.1 Einführung und Übersicht.....	54
5.5.2 MC_CAMSWITCH_REF mit ML_BASIC_CAMSWITCH_REF.....	54
5.5.3 MC_TRACK_REF mit ML_BASIC_TRACK_REF.....	55
5.5.4 MC_OUTPUT_REF.....	55
5.5.5 MC_DigitalCamSwitch.....	56
5.6 Funktionsbaustein Temperaturregler	71
5.6.1 IL_TempControlType01.....	71
5.6.2 Regler.....	79
5.6.3 Verwendung des Funktionsbausteins.....	81
5.6.4 Beschreibung der einzelnen Funktionen, Beispiele.....	84
5.7 Messtasterfunktionsbausteine.....	91
5.7.1 Einleitung und Überblick.....	91
5.7.2 Messtasterfunktionsbausteine – Komponenten und Parametrierung.....	92
Hardware.....	92
Firmware.....	92
Parametrierung der Messtasterfunktionsbausteine für IndraMotion MLC/IndraLogic XLC und IndraMotion MLD.....	93
5.7.3 MB_InitTouchProbe.....	102
5.7.4 MB_TouchProbe.....	109
5.7.5 MB_TouchProbeContinuous.....	112
5.7.6 MB_WriteExpectWindow.....	115
5.7.7 MB_AbortTrigger.....	118
5.7.8 Messtasterdatentypen.....	120
MB_PROBE_EXECUTION_MODE.....	120
MB_EXPECT_WINDOW_MODE.....	120
MB_PROBE_DATA.....	121
MB_POSITION_EVAL.....	121
MB_DIFF_EVAL.....	121
TOUCHPROBE_REF.....	122
5.8 Trigonometrische Funktionen.....	122
5.8.1 Einführung.....	122
5.8.2 ML_DegToInc.....	122
5.8.3 ML_DegToRad.....	123
5.8.4 ML_IncToDeg.....	123
5.8.5 ML_IncToRad.....	124

Inhaltsverzeichnis

	Seite
5.8.6 ML_RadToDeg.....	125
5.8.7 ML_RadToInc.....	125
5.9 Funktionsbausteine und Funktionen für die Verwaltung "zyklischer Datenkanäle".....	126
5.9.1 Einführung und Übersicht.....	126
5.9.2 MB_AllocCyclicParameter.....	128
5.9.3 MB_FreeCyclicParameter.....	130
5.9.4 MB_GetCyclicChannelConfig.....	132
5.9.5 MB_GetCyclicParameterHandle.....	133
5.9.6 MB_ReadCyclicParameter.....	135
5.9.7 MB_ReadCyclicRealParameter.....	136
5.9.8 MB_WriteCyclicParameter.....	136
5.9.9 MB_WriteCyclicRealParameter.....	137
5.9.10 MB_GetLastCyclicParameterError.....	138
5.9.11 MB_CYCLIC_PARAM_REF.....	139
5.9.12 MB_CYCLIC_TYPES.....	139
5.9.13 MB_CYCLIC_CHANNELS.....	140
5.9.14 MB_CYCLIC_CHANNEL_CONFIG.....	140
5.10 Funktionsbausteine zum Sichern und Wiederherstellen von Antriebsparametern.....	141
5.10.1 Einführung und Übersicht.....	141
5.10.2 ML_AxisBackup.....	141
5.10.3 ML_AxisRestore.....	144
5.10.4 ML_BACKUP_OPTIONS.....	146
5.10.5 ML_BACKUP_PARAMS.....	146
5.11 Bausteine für Logik Anwendungen.....	147
5.11.1 Einführung.....	147
5.11.2 IL_LinkedList.....	147
5.12 Funktionen zur Berechnung von Modulowerten.....	152
5.12.1 Einführung.....	152
5.12.2 MB_GetModuloType01.....	152
5.12.3 MB_GetModuloType02.....	153
5.12.4 MB_GetModuloType03.....	154
5.12.5 MB_GetModuloType04.....	155
5.13 Hilfsfunktionen.....	157
5.13.1 MB_GetSystemInfo.....	157
5.14 Funktionsbaustein zum Download von Kurvenscheiben.....	160
5.14.1 Einführung.....	160
5.14.2 MB_LoadCamData.....	160
5.14.3 MB_CAMPROF.....	163
5.14.4 MB_CAMTABLE.....	163
6 ML_TechMotion.library.....	165
6.1 Übersicht über die Funktionsbausteine.....	165
6.2 Funktionsbaustein zur Anwendung "Fliegende Säge".....	166
6.2.1 Einführung und Übersicht.....	166
6.2.2 MX(L)_FlyingShear	166
6.2.3 ML_FlyingShear - Besonderheiten für IndraMotion MLC.....	173

Inhaltsverzeichnis

	Seite
6.3 Funktionsbausteine zur Anwendung "Zugspannungsregler".....	175
6.3.1 Einführung und Übersicht.....	175
6.3.2 MB_TensionControlLoadCellType01.....	175
6.4 Funktionsbausteine zur Anwendung "Schlaufenregelung".....	187
6.4.1 Einführung und Übersicht.....	187
6.4.2 ML_SagControlC.....	188
6.4.3 ML_SagControlA.....	192
6.5 Funktionsbausteine zum taktsynchronen Ein- und Auskoppeln.....	196
6.5.1 Überblick.....	196
6.5.2 CamLock – Applikationsbeispiel.....	197
6.5.3 MB_PrepareCams.....	200
6.5.4 MB_CamLock.....	202
6.5.5 MB_CamLock Parametrierung.....	209
6.6 Bausteine zur allgemeinen Verwendung.....	210
6.6.1 MB_DistanceAccumulator.....	210
6.6.2 MB_AxisDistanceAccumulator.....	211
6.6.3 MB_ReachVelocityType02.....	212
6.7 Funktionsbausteine zur Anwendung "Positionsüberwachung".....	216
6.7.1 Einführung und Übersicht.....	216
6.7.2 MB_PositionMonitoring_GantryCant.....	216
6.7.3 MB_PositionMonitoring_PosCollision.....	218
6.8 Funktionsbaustein zur MagicBelt-Anwendung.....	221
6.8.1 Einführung und Überblick.....	221
6.8.2 MB_MagicBeltType01.....	223
6.9 Funktionsbaustein zur SmartBelt-Anwendung.....	256
6.9.1 MB_SmartBeltType01.....	256
6.9.2 Einführung in die SmartBelt-Verwendung.....	263
SmartBelt-Verwendung.....	263
Stauförderer.....	264
Abfolgegrad.....	265
Reibung.....	266
6.9.3 Projektplanung.....	267
Allgemeines.....	267
Mechanische Anforderungen an Band und Maschine.....	267
Antriebsanforderungen.....	268
Antriebsparametereinstellungen.....	269
SPS-Programmierung.....	271
Angabe des "SmartBeltPitch".....	273
Berechnung des "ProbeSetpoint".....	274
6.9.4 SmartBelt-Beschränkungen und -Verhalten.....	275
Beschränkungen.....	275
Zwei oder mehr Produkte auf einem Band.....	276
Mehrere Produkte pro Leitachszyklus.....	276
6.9.5 Tipps und Tricks.....	277
Korrekturabstand.....	277
Schlupf.....	277

Inhaltsverzeichnis

	Seite
6.9.6	Verbreitete Fehler und Missverständnisse..... 278
6.10	Funktionsbausteine zur Erstellung von Kurvenscheibentabellen für Technologiefunktionen..... 279
6.10.1	Einführung..... 279
6.10.2	MB_PilgrimStepCalcType01..... 280
6.10.3	MB_PrintLengthCorrCalcType01..... 290
7	ML_TechRegi.library/MX_TechRegi.lib..... 297
7.1	Einführung und Übersicht..... 297
7.2	Registerregler-FB..... 298
7.2.1	Überblick..... 298
7.2.2	MB_RegisterControllerType02..... 298
7.2.3	MB_RegisterControllerType04..... 308
7.2.4	Konfiguration des Registerreglers..... 321
	Überblick..... 321
	Einsatz der Registerregelung..... 322
7.2.5	Registerregler – Anwendungsbeispiele..... 326
	Stanzen bei Etikettendruck..... 326
	Insetter-Regelung..... 326
	Schlauchbeutelmaschine..... 327
7.3	Seitenregisterregler-FB..... 329
7.3.1	Überblick..... 329
7.3.2	MB_RegisterControllerSideType01..... 329
7.3.3	Konfiguration des Seitenregisters..... 336
7.4	Registerregelungs-, Mess- und Stellfunktionsbausteine..... 342
7.4.1	Überblick..... 342
7.4.2	MB_RegiMeasuringType01..... 342
7.4.3	MB_RegiMeasuringType02..... 344
7.4.4	MB_RegiMeasuringType03..... 348
7.4.5	MB_RegiRegulateType01..... 352
7.4.6	MB_RegiRegulateType02..... 356
7.4.7	MB_RegiRegulateType03..... 366
7.4.8	MB_RegiRegulateType04..... 372
7.5	Registersollwertübernahme-FB..... 380
7.5.1	Überblick..... 380
7.5.2	MB_SetpointLockType01..... 381
7.6	Registerqualitäts-FB..... 383
7.6.1	Überblick..... 383
7.6.2	IL_RegiQualityType01..... 383
8	ML_TechCrank.library..... 387
8.1	Einführung und Übersicht..... 387
8.2	Funktionen und Funktionsbausteine für die Kurbelkinematik..... 388
8.2.1	Einführung und Übersicht..... 388
8.2.2	Allgemeine Definitionen..... 390
	Definition der grundlegenden Variablen bei der Kurbelkinematik..... 390

Inhaltsverzeichnis

	Seite
Zählrichtung.....	392
Mechanische (Xmech) und virtuelle (Xvirt) translatorische Position.....	392
8.2.3 MB_CamTableCrank.....	393
8.2.4 MB_CamTableCrankSuperimposed.....	396
8.2.5 MB_PhiToXvirt.....	399
8.2.6 MB_MasterToPhi.....	401
8.2.7 MB_XvirtToXmech.....	401
8.3 Funktionen und Funktionsbausteine für die Kniehebelkinematik.....	402
8.3.1 Einführung und Übersicht.....	402
8.3.2 Allgemeine Definitionen.....	403
8.3.3 MB_BellCrankData.....	406
8.3.4 MB_CamTableBellCrank.....	408
8.3.5 MB_CamTableBellCrankSuperimposed.....	412
8.3.6 MB_PhiToYvirt.....	416
8.3.7 MB_YvirtToPhi.....	418
8.3.8 MB_YvirtToPhiPoly5.....	420
8.3.9 MB_YvirtToYmech.....	422
8.3.10 MB_YVirtToWorkRange.....	424
9 RMB_TechWinder.library/MX_TechWinder.lib.....	425
9.1 Einführung und Übersicht.....	425
9.2 Funktionsbaustein zur Tänzerlageregelung	426
9.2.1 Einführung und Übersicht.....	426
9.2.2 MB_DancerControlType03	428
9.2.3 Inbetriebnahme Tänzerlageregelung	435
9.3 Funktionsbaustein zur Durchmesserberechnung	436
9.3.1 Einführung und Übersicht.....	436
9.3.2 MB_DiameterCalculatorType03	436
9.3.3 Inbetriebnahme des Durchmesserrechners	444
9.4 Funktionsbaustein zur Erfassung eines extern gemessenen Durchmessers.....	445
9.4.1 Einführung und Übersicht.....	445
9.4.2 MB_DiameterMeasurementType01	445
9.5 Wickelrechner mit Tänzer	449
9.5.1 Einführung und Übersicht.....	449
9.5.2 MB_WinderDancerCtrlType01	450
9.5.3 Inbetriebnahme des Wickelrechners mit Tänzer	459
9.6 Funktionsbaustein zum Wickelrechner mit Zugkraftsteuerung bzw. Zugkraftregelung.....	460
9.6.1 Einführung und Übersicht.....	460
9.6.2 MB_WinderTensionCtrlType02	462
9.6.3 Inbetriebnahme des Wickelrechners mit Zugkraftsteuerung bzw. Zugkraftregelung.....	476
9.7 Funktionsbaustein zum Wickelrechner mit Zugkraftregelung und Drehzahlkorrektur.....	481
9.7.1 Einführung und Übersicht.....	481
9.7.2 MB_WinderTensionCtrlSpeedType01.....	481
9.7.3 Inbetriebnahme des Wicklers mit Zugregelung und Drehzahlkorrektur.....	493
9.8 Ergänzende Funktionsbausteine	497
9.8.1 Einführung und Übersicht.....	497

Inhaltsverzeichnis

	Seite
9.8.2 MB_CalcInertiaLimitType02	497
9.8.3 MB_UnwindMaterialType02	500
9.8.4 MB_WindTaperProfileType01.....	505
9.8.5 MB_WindSpeedControlAdaptionType01.....	508
9.9 Parameterübersicht	512
10 RMB_TechCrossCutCrossSeal.library.....	515
10.1 MB_CrossCutSealType01.....	515
10.2 MB_CrossCutterCalcType04.....	530
10.2.1 Allgemeines.....	530
10.2.2 Besonderheiten der IndraMotion MLC.....	538
10.2.3 MB_CAM_TABLE_DATA.....	539
10.2.4 MB_CORR_PROFILE.....	541
10.2.5 MB_RESOLUTION.....	542
10.2.6 MB_PUSH_OUT_CONFIG.....	543
11 RMB_TechCam.library.....	545
11.1 Einführung und Übersicht.....	545
11.2 MB_FeedMovementCalcType01.....	545
11.3 MB_FeedMovementCalcType02.....	551
11.4 MB_CoordinatePairCamType01.....	558
12 RIL_ParameterChannel.library.....	567
12.1 Einführung und Übersicht.....	567
12.2 IL_ParameterChannel.....	567
12.3 IL_ParameterChannel – Protokoll, Fehlerbehandlung und Inbetriebnahme.....	574
12.3.1 Beschreibung des unterlagerten Protokolls.....	574
Parameterkanal im zyklischen Datenkanal.....	574
Steuer-/Statuswort.....	575
Index und Subindex.....	576
Initialisierung der Kommunikation.....	579
Fehlertelegramme.....	579
12.3.2 Inbetriebnahme des Bausteins.....	582
12.3.3 Unterschiede zu Altsystemen.....	586
Allgemeines.....	586
Auf dem Feldbus-Slave.....	586
Auf dem Feldbus-Master.....	587
13 ML_TechTensionAdvanced.library.....	591
13.1 Einführung und Übersicht.....	591
13.2 Multiachs Zugspannungsregler MB_TensionControlLoadCellType02.....	591
13.3 Funktionsbausteine zur Anwendung "Entkopplungsnetzwerk".....	607
13.3.1 MB_TensionDecouplingNetworkType01.....	607

Inhaltsverzeichnis

	Seite
14 Service und Support.....	619
 Index.....	621

1 Zu dieser Dokumentation

1.1 Gültigkeit der Dokumentation

Zielgruppe Diese Dokumentation richtet sich an Anwender, die für Ihre Applikationen Funktionen und Funktionsbausteine aus den Technologie-Bibliotheken einsetzen.

Diese Dokumentation unterstützt den Anwender in den Phasen

- Engineering und
- Inbetriebnahme

1.2 Gliederung der Dokumentation

Der erste Teil des Dokuments enthält wichtige Gebrauchs- und Sicherheitshinweise ([Kap. 2 "Wichtige Gebrauchshinweise" auf Seite 17](#) und [Kap. 3 "Sicherheitshinweise für elektrische Antriebe und Steuerungen" auf Seite 19](#)).

[Kap. 4 "Übersicht" auf Seite 31](#) beinhaltet eine Auflistung der in dieser Dokumentation beschriebenen Bibliotheken.

[Kap. 5 "ML_TechBase.library" auf Seite 33](#) enthält die Beschreibungen der Technologie-Basisfunktionsbausteine.

[Kap. 6 "ML_TechMotion.library" auf Seite 165](#) enthält die Beschreibungen der TechMotion-Funktionsbausteine.

[Kap. 7 "ML_TechRegi.library/MX_TechRegi.lib" auf Seite 297](#) beschreibt die zur Registerregelung verfügbaren Funktionsbausteine.

[Kap. 8 "ML_TechCrank.library" auf Seite 387](#) beschreibt die Funktionen und Funktionsbausteine für Kurbel und Kniehebelkinematiken.

[Kap. 9 "RMB_TechWinder.library/MX_TechWinder.lib" auf Seite 425](#) enthält die Beschreibung der Funktionsbausteine für das Auf- und Abwickeln von Stoffbahnen (Wicklern).

[Kap. 10 "RMB_TechCrossCutCrossSeal.library" auf Seite 515](#) liefert Beschreibungen der Funktionsbausteine zur Quersiegel- / Querschneiderfunktionalität.

[Kap. 11 "RMB_TechCam.library" auf Seite 545](#) beschreibt die Funktionsbausteine zur Erstellung von Kurvenscheibentabellen.

[Kap. 12 "RIL_ParameterChannel.library" auf Seite 567](#) enthält eine Beschreibung des Funktionsbausteins zur azyklischen Kommunikation über den zyklischen Kanal einer Feldbusverbindung.

[Kap. 13 "ML_TechTensionAdvanced.library" auf Seite 591](#) beschreibt die **lizenspflichtigen** Funktionsbausteine für Multiachs Zugspannungsregler mit Entkopplungsnetzwerk.

[Kap. 14 "Service und Support" auf Seite 619](#) enthält Informationen zu unserem Kundendienst-Helpdesk.

Zu dieser Dokumentation

1.3 Erforderliche und ergänzende Dokumentationen

Dokumentationstitel mit Typenschlüssel und Materialnummer

IndraWorks		MLC	XLC
/36/	Rexroth IndraWorks 12VRS Software-Installation DOK-IWORKS-SOFTINS*V12-COxx-DE-P, R911334395 Die Dokumentation beschreibt den Installationsprozess von IndraWorks.	X	X
/5/	Rexroth IndraWorks 12VRS Engineering DOK-IWORKS-ENGINEE*V12-APxx-DE-P, R911334387 Diese Dokumentation beschreibt die Verwendung von IndraWorks, in das sich die Rexroth Engineering-Tools integrieren. Beschrieben werden das Arbeiten mit IndraWorks und die Bedienung der Oszilloskop-Funktion.	X	X
/20/	Rexroth IndraMotion MLC 12VRS Funktionsbeschreibung DOK-MLC***-FUNC****V12-APxx-DE-P, R911333847 Die Dokumentation dient der Beschreibung der Wizards, Kontextmenüs, Dialoge, Steuerungsinbetriebnahme, Gerätekonfiguration, Funktionalitäten der IndraMotion MLC.	X	
/20/	Rexroth IndraLogic XLC 12VRS Funktionsbeschreibung DOK-XLC***-FUNC****V12-APxx-DE-P, R911333877 Die Dokumentation dient der Beschreibung der Wizards, Kontextmenüs, Dialoge, Steuerungsinbetriebnahme, Gerätekonfiguration, Funktionalitäten der IndraLogic XLC.		X
/7/	Rexroth IndraWorks 12VRS CamBuilder DOK-IWORKS-CAMBUIL*V12-APxx-DE-P, R911333841 Diese Dokumentation dient zur Beschreibung der Grundlagen und der Bedienung des CamBuilder, des Werkzeugs zur Kurvenscheibenbearbeitung.	X	X
/37/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS Automation Interface DOK-XLCMLC-AUT*INT*V12-APxx-DE-P, R911334177 Diese Dokumentation beschreibt den skriptbasierten Zugriff auf IndraWorks-Projektdaten über die Schnittstelle des Automation Interfaces.	X	X
/38/	Rexroth IndraWorks 12VRS FDT-Container DOK-IWORKS-FDT*CON*V12-APxx-DE-P, R911334397 Diese Dokumentation beschreibt die FDT-Container-Funktionalität von IndraWorks. Beschrieben wird das Aktivieren der Funktionalität im Projekt und das Arbeiten mit DTMs.	X	X
/29/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS Projektkonvertierung DOK-XLCMLC-PROCONV*V12-APxx-DE-P, R911334186 Diese Dokumentation beschreibt die Projektkonvertierung von IndraLogic 04VRS und IndraMotion MLC04VRS auf IndraWorks Version 12 mit IndraLogic 2G. Es wird insbesondere auf die Änderungen im Bereich Motion und SPS eingegangen.	X	X
/28/	Rexroth IndraMotion MLC 12VRS Inbetriebnahme DOK-MLC***-STARTUP*V12-COxx-DE-P, R911333857 Das vorliegende Handbuch beschreibt die Inbetriebnahme- und Servicearbeitsschritte für das System IndraMotion MLC. Es beinhaltet Checklisten für häufig durchzuführende Arbeiten und eine ausführliche Beschreibung der Arbeitsschritte.	X	

Zu dieser Dokumentation

Motion		MLC	XLC
/23/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS PLCopen-Bibliotheken DOK-XLCMLC-FUNLIB**V12-LIxx-DE-P, R911334181 Diese Dokumentation beschreibt die Funktionsbausteine, Funktionen und Datentypen der Bibliotheken RIL_CommonTypes, ML_Base und ML_PLCopen für die IndraLogic XLC/IndraMotion MLC. Die Dokumentation enthält ferner die Fehlerreaktion der Bausteine.	X	X
/27/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS Generic Application Template DOK-XLCMLC-TF*GAT**V12-APxx-DE-P, R911334190 Dieses Handbuch stellt dem IndraLogic SPS-Programmierer ein strukturiertes Template, welches zum Hinzufügen und Bearbeiten des SPS-Programmiercodes verwendet werden kann, zur Verfügung. Beschrieben werden das Template selbst, der Template Wizard und Beispielapplikationen.	X	X
/31/	Rexroth IndraMotion MLC 12VRS RCL-Programmieranleitung DOK-MLC***-RCL*PRO*V12-APxx-DE-P, R911333851 Das vorliegende Handbuch informiert über die Robot Control. Die Programmiersprache RCL (Robot Control Language) steht dabei im Vordergrund. Beschrieben werden die Programmstruktur, Variablen, Funktionen, Bewegungsanweisungen sowie die notwendigen Parameter des Systems.	X	
/21/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS Parameter DOK-XLCMLC-PARAM***V12-RExx-DE-P, R911334185 Diese Dokumentation beschreibt die Parameter der Systeme XLC/MLC, sowie das Wechselspiel zwischen Parametrieren und Programmieren. Beschrieben werden die Achs-Parameter, die Steuerungs-Parameter, die Kinematik-Parameter, die Messtaster- und die Nockenschaltwerks-Parameter.	X	X
/10/	Rexroth IndraDrive Firmware für Antriebsregelgeräte MPH-, MPB-, MPD-, MPC-07 DOK-INDRV*-MP*-07VRS**-FKxx-DE-P, R911328669		
/11/	Rexroth IndraDrive MPx-16 Funktionen DOK-INDRV*-MP*-16VRS**-APxx-DE-P, R911326484		

Feldbusse		MLC	XLC
/39/	Rexroth IndraMotion MLC 11VRS PLCopen Feldbus DOK-IM*ML*-PLCFBUS*V11-APxx-DE-P, R911333895 Die Dokumentation beschreibt die Anlage von Feldbusantrieben innerhalb eines IndraWorks-Projekts, Funktionsbausteine, Funktionen und Datentypen der Bibliotheken RIL_CommonTypes.library (Auszug für Feldbusantriebe), RMB_PLCopenFieldBus.library, RIL_Uilities.library (Auszug für Feldbusantriebe). Die Dokumentation enthält ferner die Fehlerreaktion der Bausteine.	X	X
/4/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS Feldbusse DOK-XLCMLC-FB*****V12-APxx-DE-P, R911334393 Diese Dokumentation dient der Beschreibung der unterstützten Feldbusse und ihrer Diagnose-Funktionsbausteine.	X	X

HMI		MLC	XLC
/8/	Rexroth IndraWorks 12VRS HMI DOK-IWORKS-HMI*****V12-APxx-DE-P, R911334391 Diese Dokumentation beschreibt die Funktionen, die Konfiguration und die Bedienung der Bedienoberflächen IndraWorks HMI Engineering und IndraWorks HMI Operation.	X	X

Zu dieser Dokumentation

/6/	Rexroth IndraWorks 12VRS WinStudio DOK-IWORKS-WINSTUD*V12-APxx-DE-P, R911333843 Diese Dokumentation die Installation der Software, das Arbeiten mit WinStudio und das Erstellen und den Betrieb von Applikationen.	X	X
/50/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS HMI-Anbindung DOK-XLCMLC-HMI*****V12-APxx-DE-P, R911334183 Dieses Dokument beschreibt die von der IndraLogic XLC und IndraMotion MLC unterstützten Visualisierungssysteme sowie deren Anbindung.	X	X

SPS		MLC	XLC
/3/	Rexroth IndraWorks 12VRS IndraLogic 2G Programmieranleitung DOK-IWORKS-IL2GPRO*V12-APxx-DE-P, R911334389 Diese Dokumentation beschreibt das SPS-Programmierungswerkzeug IndraLogic 2G und seine Verwendung. Beschrieben werden die Basisverwendung, Erste Schritte, die Visualisierung, Menüpunkte und die Editoren.	X	X
/33/	Rexroth IndraWorks 12VRS Basis-Bibliotheken IndraLogic 2G DOK-IL*2G*-BASLIB**V12-LIxx-DE-P, R911333833 Das vorliegende Handbuch beschreibt die systemübergreifenden SPS-Bibliotheken.	X	X

Technologie			
/30/	Rexroth IndraMotion MLC 12VRS Technologie-Bibliotheken DOK-MLC***-TF*LIB**V12-LIxx-DE-P, R911333867 Diese Dokumentation beschreibt die Funktionsbausteine, Funktionen und Datentypen der Bibliotheken "ML_TechInterface.library", "ML_TechMotion.library", "RMB_TechCam.library" und "ML_TechBase.library" sowie Bibliotheken zur Wicklerfunktionalität, Registerreglerfunktionalität und Querschneiderfunktionalität.	X	
/60/	Rexroth IndraMotion MLC 12VRS RegisterControl (Bibliothek) DOK-MLC***-REGI*CO*V12-LIxx-DE-P, R911333855 Diese Dokumentation beschreibt die Ein- und Ausgänge der einzelnen Funktionsbausteine und gibt Hinweise für deren Verwendung.	X	
/62/	Rexroth IndraMotion MLC 12VRS RegisterControl (Anwendungsbeschreibung) DOK-MLC***-REGI*CO*V12-APxx-DE-P, R911333853 Dieses Dokument beschreibt die Anwendung der integrierten Registerregelung für eine Tiefdruckmaschine. Beschrieben werden die Komponenten des Markenstromsensors, die HMI-Anwendung und die Fehlerbehebung. Diese Anleitung dient dazu, die Registerregelung zu bedienen und auf Fehler zu reagieren sowie Diagnosen abzufragen. Die Zielgruppe für diese Dokumentation sind Maschineneinrichter und Maschinenbediener.	X	
/49/	Rexroth IndraMotion MLC 12VRS Anwendung Wicklerfunktionen DOK-MLC***-TF*WIND*V12-APxx-DE-P, R911333869 Diese anwendungsbezogene Systemdokumentation beschreibt die Anwendung der Wickler-Technologiefunktionen.	X	

Hardware		MLC	XLC
/1/	Rexroth IndraControl L45/L65 DOK-CONTRL-IC*L45*L65*-PRxx-DE-P, R911324660	X	X
/2/	Rexroth IndraControl L25 DOK-CONTRL-IC*L25*****-PRxx-DE-P, R911328473	X	X

Zu dieser Dokumentation

/24/	Rexroth IndraControl Lxx 12VRS Funktionsmodule DOK-CTRL-FM*LXX**V12-APxx-DE-P, R911333832 Diese Dokumentation dient der Beschreibung aller Funktionsmodule der Lxx Steuerungen inkl. Engineering und Diagnose.	X	X
/12/	Rexroth IndraDrive Antriebsregelgeräte MPx-02 bis MPx-07 DOK-INDRV*-GEN-**VRS**-PAxx-DE-P, R911297316		
/13/	Rexroth IndraDrive MPx-02 bis MPx-07 und HMV DOK-INDRV*-GEN-**VRS**-WAxx-DE-P, R911297318		
/35/	Rexroth IndraDrive Antriebsregelgeräte Steuerteile CSB01, CSH01, CDB01 DOK-INDRV*-CSH*****-PR08-DE-P, R911295011		

Diagnose und Service		MLC	XLC
/26/	Rexroth IndraMotion MLC/XLC 11VRS Service Tool DOK-IM*ML*-IMST***V11-RExx-DE-P, R911331939	X	X
/22/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS Diagnosen DOK-XLCMLC-DIAG***V12-RExx-DE-P, R911334179 Diese Dokumentation enthält alle in den Steuerungssystemen IndraLogic XLC und IndraMotion MLC implementierten Steuerungsparameter.	X	X

Systemübersicht		MLC	XLC
/48/	Rexroth IndraMotion for Printing 12VRS Systemübersicht DOK-IM*PR*-SYSTEM**V12-PRxx-DE-P, R911333839 Das vorliegende Handbuch beschreibt das Produkt IndraMotion for Printing. Es stellt die Steuerungs-, Antriebs- und IO-Systeme vor und gibt eine Einführung in die Inbetriebnahme und Programmierung.	X	
/48/	Rexroth IndraMotion for Packaging 12VRS Systemübersicht DOK-IM*PA*-SYSTEM**V12-PRxx-DE-P, R911333837 Das vorliegende Handbuch beschreibt das Produkt IndraMotion for Packaging. Es stellt die Steuerungs-, Antriebs- und IO-Systeme vor und gibt eine Einführung in die Inbetriebnahme und Programmierung.	X	
/9/	Rexroth IndraMotion MLC 12VRS Systemübersicht DOK-MLC***-SYSTEM**V12-PRxx-DE-P, R911333859 Diese Dokumentation stellt eine Übersicht der möglichen Hardware-/Softwarekomponenten des Automationssystems IndraMotion MLC in der genannten Version dar. Die Dokumentation dient als Auswahlhilfe zur Zusammenstellung eines Systems.	X	
/9/	Rexroth IndraLogic XLC 12VRS Systemübersicht DOK-XLC***-SYSTEM**V12-PRxx-DE-P, R911333879 Diese Dokumentation stellt eine Übersicht der möglichen Hardware-/Softwarekomponenten des Automationssystems IndraLogic XLC in der genannten Version dar. Die Dokumentation dient als Auswahlhilfe zur Zusammenstellung eines Systems.		X

Zu dieser Dokumentation

Erste Schritte		MLC	XLC
/25/	Rexroth IndraMotion MLC 12VRS Die ersten Schritte DOK-MLC***-F*STEP**V12-COxx-DE-P, R911333845 Diese Dokumentation beschreibt die "ersten Schritte" mit der IndraMotion MLC und RobotControl. Beschrieben werden die Hard- und Softwarevoraussetzungen und das Erstellen eines Projekts.	X	
/25/	Rexroth IndraLogic XLC 12VRS Die ersten Schritte DOK-XLC***-F*STEP**V12-COxx-DE-P, R911333875 Diese Dokumentation beschreibt die "ersten Schritte" mit der IndraLogic XLC. Beschrieben werden die Hard- und Softwarevoraussetzungen und das Erstellen eines Projekts.		X

xx entsprechender Ausgabestand
Abb. 1-1: XLC/MLC-Dokumentationsübersicht

1.4 Darstellung von Informationen

1.4.1 Sicherheitshinweise

Die Sicherheitshinweise, soweit sie in der vorliegenden Anwendungsdokumentation vorhanden sind, beinhalten bestimmte Signalwörter ("Gefahr", "Warnung", "Vorsicht", "Hinweis") und ggf. eine Signalgrafik (nach ANSI Z535.6-2006).

Das Signalwort soll die Aufmerksamkeit auf den Sicherheitshinweis lenken und bezeichnet die Schwere der Gefährdung.

Die Signalgrafik (Warndreieck mit Ausrufezeichen), welche den Signalwörtern "Gefahr", "Warnung" und "Vorsicht" vorangestellt wird, weist auf Gefährdungen für Personen hin.

Die Sicherheitshinweise in dieser Dokumentation werden wie folgt dargestellt:

GEFAHR

Bei Nichtbeachtung dieses Sicherheitshinweises **werden** Tod oder schwere Körperverletzung eintreten.

WARNUNG

Bei Nichtbeachtung dieses Sicherheitshinweises **können** Tod oder schwere Körperverletzung eintreten.

VORSICHT

Bei Nichtbeachtung dieses Sicherheitshinweises können mittelschwere oder leichte Körperverletzung eintreten.

HINWEIS

Bei Nichtbeachtung dieses Sicherheitshinweises können Sachschäden eintreten.

1.4.2 Verwendete Symbole

Darstellung Die Darstellung der unterschiedlichen Symbole finden Sie im Kapitel [3.4 Erläuterung der Signalwörter und der Signalgrafik, Seite 28](#).

Zu dieser Dokumentation

1.4.3 Bezeichnungen und Abkürzungen

Begriff	Erklärung
Bibliotheken	beinhalten Funktionen und Funktionsbausteine
Tech-FB	Technologie-Funktionsbaustein
PLS	Programmable Limit Switch (Nockenschaltwerk)
PLCOpen	ist eine 1992 gegründete, internationale hersteller- und produktunabhängige Interessensgemeinschaft von Steuerungsherstellern, Softwarehäusern und Instituten. Dem SPS-Standard IEC 61131-3 verpflichtend werden in verschiedenen Arbeitsgruppen (Technical Committes) Standards definiert, die eine erhebliche Effizienzsteigerung in der Entwicklung von Applikationssoftware ermöglichen
Zielsystem	Steuerung (z.B. IndraMotion MLC)
SYNTAX	Automatisierungssystem für Druck- und Verarbeitungsmaschinen
VisualMotion	programmierbares Mehrachsen-Bewegungssteuerungssystem
IndraMotion MLC	kompaktes Motion-Logic-System mit Motion-, Robot- und Logic-Control-Funktionalität
IndraLogic XLC	kompaktes SPS-System mit integrierter Motion-Funktionalität
IndraMotion MLD	Antriebsbasiertes Motion-Logic-System
IndraWorks	Projektierungs- und Inbetriebnahmetool von Bosch Rexroth
IndraDrive	Antriebs-Seuerggerät
MA	IndraMotion MLD Advanced
MLD-S	Motion Logic Drive (single drive)
MLD-M	Motion Logic Drive (multi drive)
D.h.	Das heißt
z.B.	zum Beispiel
n.def.	nicht definiert
vgl.	vergleiche
s	Sekunde
ms	Millisekunde
m	Meter
mm	Millimeter
%	Prozent

Abb. 1-2: Verwendete Bezeichnungen und Abkürzungen

2 Wichtige Gebrauchshinweise

2.1 Bestimmungsgemäßer Gebrauch

2.1.1 Einführung

Die Produkte von Bosch Rexroth werden nach dem jeweiligen Stand der Technik entwickelt und gefertigt. Vor ihrer Auslieferung werden sie auf ihren betriebssicheren Zustand hin überprüft.

WARNUNG

Personen- und Sachschäden durch falschen Gebrauch der Produkte!

Die Produkte sind für den Einsatz im industriellen Umfeld konzipiert und dürfen nur bestimmungsgemäß eingesetzt werden. Wenn sie nicht bestimmungsgemäß eingesetzt werden, dann können Situationen entstehen, die Sach- und Personenbeschädigung nach sich ziehen.



Für Schäden bei nicht-bestimmungsgemäßigem Gebrauch der Produkte leistet Bosch Rexroth als Hersteller keinerlei Gewährleistung, Haftung oder Schadensersatz; die Risiken bei nicht-bestimmungsgemäßigem Gebrauch der Produkte liegen allein beim Anwender.

Bevor Sie Bosch Rexroth Produkte einsetzen, müssen die folgenden Voraussetzungen erfüllt sein, um einen bestimmungsgemäßen Gebrauch der Produkte zu gewährleisten:

- Jeder, der in irgendeiner Weise mit einem unserer Produkte umgeht, muss die entsprechenden Sicherheitsvorschriften und den bestimmungsgemäßen Gebrauch lesen und verstehen
- Hardwareprodukte müssen in ihrem Originalzustand belassen werden; d. h. es dürfen keine baulichen Veränderungen an ihnen vorgenommen werden
- Softwareprodukte dürfen nicht dekompiert und ihre Quellcodes dürfen nicht verändert werden
- Beschädigte oder fehlerhafte Produkte dürfen nicht eingebaut oder betrieben werden
- Es muss gewährleistet sein, dass die Produkte entsprechend den in der Dokumentation genannten Vorschriften installiert sind

2.1.2 Einsatz- und Anwendungsbereiche

Die IndraMotion MLC/IndraLogic XLC von Bosch Rexroth ist für Motion-/Logic-Anwendungen vorgesehen.



Die IndraMotion MLC/IndraLogic XLC darf nur mit den in dieser Dokumentation angegebenen Zubehör- und Anbauteilen verwendet werden. Nicht ausdrücklich genannte Komponenten dürfen weder angebaut noch angeschlossen werden. Gleiches gilt für Kabel und Leitungen.

Der Betrieb darf nur in den ausdrücklich angegebenen Konfigurationen und Kombinationen der Komponenten und mit der in der jeweiligen Funktionsbeschreibung angegebenen und spezifizierten Soft- und Firmware erfolgen.

Wichtige Gebrauchshinweise

Die IndraMotion MLC/IndraLogic XLC ist für Logic-Anwendungen sowie den Einsatz in ein- und mehrachsigen Antriebs- und Steuerungsaufgaben entwickelt worden.

Für den applikationsspezifischen Einsatz der IndraMotion MLC/IndraLogic XLC stehen Gerätetypen mit unterschiedlicher Leistung und unterschiedlichen Schnittstellen zur Verfügung.

Typische Anwendungsbereiche der IndraMotion MLC/IndraLogic XLC sind:

- Handhabungs- und Montagesysteme
- Verpackungs- und Lebensmittelmaschinen
- Druck- und Papierverarbeitungsmaschinen
- Werkzeugmaschinen
- Pressen
- Hütten- und Walztechnik
- Fördertechnik
- Gebäudetechnik
- Sondermaschinen
- Holzbearbeitungsmaschinen

Die IndraMotion MLC/IndraLogic XLC darf nur unter den in dieser Dokumentation angegebenen Montage- und Installationsbedingungen, in der angegebenen Gebrauchslage und unter den angegebenen Umweltbedingungen (Temperatur, Schutzart, Feuchte, EMV u. a.) betrieben werden.

2.2 Nicht-bestimmungsgemäßer Gebrauch

Die Verwendung der IndraMotion MLC/IndraLogic XLC außerhalb der vorgenannten Anwendungsgebiete oder unter anderen als den in der Dokumentation beschriebenen Betriebsbedingungen und angegebenen technischen Daten gilt als "nicht bestimmungsgemäß".

Die IndraMotion MLC/IndraLogic XLC darf nicht eingesetzt werden wenn

- sie Betriebsbedingungen ausgesetzt werden, die die vorgeschriebenen Umgebungsbedingungen nicht erfüllen. Untersagt sind z. B. der Betrieb unter Wasser, unter extremen Temperaturschwankungen, extremen Maximaltemperaturen oder in explosionsgefährdeter Umgebung
- die beabsichtigten Anwendungen von Bosch Rexroth nicht freigegeben sind. Beachten Sie hierzu bitte unbedingt die Aussagen in den allgemeinen Sicherheitshinweisen!

Weiterhin ist der Einsatz als sicherheitsbezogenes Teil von Steuerungen im Sinne der DIN EN ISO 13849 nicht bestimmungsgemäß. Die funktionale Sicherheit muss durch zusätzliche sicherheitstechnisch zertifizierte Komponenten realisiert werden.

3 Sicherheitshinweise für elektrische Antriebe und Steuerungen

3.1 Begriffsdefinitionen

Anlage	Eine Anlage besteht aus mehreren zu einem bestimmten Zweck und an einem bestimmten Ort miteinander verbundenen Geräten oder Systemen, die jedoch nicht als eine einzige Funktionseinheit in Verkehr gebracht werden sollen.
Elektrisches Antriebssystem	Ein elektrisches Antriebssystem umfasst alle Bestandteile von der Netzeinspeisung bis zur Motorwelle; dazu zählen z. B. Elektromotor(en), Motorgeber, Versorgungs- und Antriebsregelgeräte, sowie Hilfs- und Zusatzkomponenten, wie Netzfilter, Netzdrossel und dazugehörige Leitungen und Kabel.
Anwender	Ein Anwender ist eine Person, die ein in Verkehr gebrachtes Produkt installiert, in Betrieb nimmt oder verwendet.
Anwendungsdokumentation	Eine Anwendungsdokumentation umfasst die gesamte Dokumentation, die dazu dient, den Anwender des Produkts über den Gebrauch und sicherheitsrelevante Inhalte für Projektierung, Einbau, Installation, Montage, Inbetriebnahme, Betrieb, Wartung, Reparatur, Außerbetriebnahme des Produkts zu informieren. Folgende Begriffe sind dafür ebenfalls üblich: Bedienungsanleitung, Betriebsanleitung, Inbetriebnahmeanleitung, Gebrauchsanleitung, Projektierungsanleitung, Anwendungsbeschreibung usw.
Elektrisches Betriebsmittel	Ein elektrisches Betriebsmittel ist ein Gegenstand, der zum Erzeugen, Umwandeln, Fortleiten, Verteilen oder Anwenden von elektrischer Energie benutzt wird, wie z. B. Elektromotoren, Transformatoren, Schaltgeräte, Kabel, Leitungen, Stromverbrauchsgeräte, bestückte Leiterplatten, Einschübe, Schaltschränke usw.
Gerät	Ein Gerät ist ein Endprodukt mit einer ihm eigenen Funktion, das für Anwender bestimmt ist und als eine einzelne Handelsware in Verkehr gebracht wird.
Hersteller	Unter Hersteller ist eine natürliche oder juristische Person zu verstehen, welche die Verantwortung für die Auslegung und die Herstellung eines Produktes trägt, das in seinem Namen in den Verkehr gebracht wird. Der Hersteller kann Fertigerzeugnisse, Fertigteile oder Fertigelemente verwenden oder Arbeiten an Subunternehmer vergeben. Er muss jedoch immer die Oberaufsicht behalten und die notwendigen Befugnisse besitzen, um die Verantwortung für das Produkt übernehmen zu können.
Komponente	Eine Komponente ist eine Kombination von Bauelementen mit vorgegebener Funktion, die Teil eines Betriebsmittels, Gerätes oder Systems sind. Komponenten des elektrischen Antriebs- und Steuerungssystems sind z. B. Versorgungsgeräte, Antriebsregelgeräte, Netzdrossel, Netzfilter, Motoren, Kabel, usw.
Maschine	Unter Maschine ist die Gesamtheit von miteinander verbundenen Teilen oder Baugruppen zu verstehen, von denen mindestens eine(s) beweglich ist. Eine Maschine besteht somit aus entsprechenden Maschinenantriebselementen sowie Steuer- und Energiekreisen, die für eine bestimmte Anwendung zusammengefügt sind. Eine Maschine ist z. B. für die Verarbeitung, Behandlung, Fortbewegung oder Verpackung eines Materials bestimmt. Der Ausdruck "Maschine" deckt auch eine Zusammenstellung von Maschinen ab, die so angeordnet und gesteuert werden, dass sie als einheitliches Ganzes funktionieren.
Produkt	Beispiele für ein Produkt: Gerät, Komponente, Bauteil, System, Software, Firmware u. a.

Sicherheitshinweise für elektrische Antriebe und Steuerungen

Projektierungsanleitung	Eine Projektierungsanleitung ist Teil der Anwendungsdokumentation zur Hilfestellung bei der Auslegung und Planung von Systemen, Maschinen oder Anlagen.
Qualifiziertes Personal	Im Sinne dieser Anwendungsdokumentation umfasst das qualifizierte Personal diejenigen Personen, die mit der Installation, Montage, Inbetriebnahme und Betrieb der Komponenten des elektrischen Antriebs- und Steuerungssystems sowie den damit verbundenen Gefahren vertraut sind und über die ihre Tätigkeit entsprechende Qualifikationen verfügen. Zu derartigen Qualifikationen gehören u. a.: 1) Eine Ausbildung oder Unterweisung bzw. Berechtigung, um Stromkreise und Geräte sicher ein- und auszuschalten, zu erden und zu kennzeichnen. 2) Eine Ausbildung oder Unterweisung für die Pflege und den Gebrauch angemessener Sicherheitsausrüstung 3) Eine Schulung in Erster Hilfe
Steuerungssystem	Ein Steuerungssystem umfasst mehrere miteinander verbundene Steuerungskomponenten, die als eine einzige Funktionseinheit in Verkehr gebracht werden.

3.2 Grundsätzliche Hinweise

3.2.1 Benutzung und Weitergabe der Sicherheitshinweise

Installieren und betreiben Sie keine Komponenten des elektrischen Antriebs- und Steuerungssystems, bevor Sie alle mitgelieferten Unterlagen sorgfältig durchgelesen haben. Diese Sicherheitshinweise und alle anderen Benutzerhinweise sind vor jeder Arbeit mit diesen Komponenten durchzulesen. Sollten Ihnen keine Benutzerhinweise für die Komponenten zur Verfügung stehen, wenden Sie sich an Ihren zuständigen Vertriebspartner von Bosch Rexroth. Verlangen Sie die unverzügliche Übersendung dieser Unterlagen an den oder die Verantwortlichen für den sicheren Betrieb der Komponenten.

Bei Verkauf, Verleih und/oder anderweitiger Weitergabe der Komponente sind diese Sicherheitshinweise ebenfalls in der Landessprache des Anwenders mitzugeben.

Unsachgemäßer Umgang mit diesen Komponenten und Nichtbeachten der hier angegebenen Sicherheitshinweise sowie unsachgemäße Eingriffe in die Sicherheitseinrichtung können zu Sachschäden, Körperverletzung, elektrischem Schlag oder im Extremfall zum Tod führen.

3.2.2 Voraussetzungen für den sicheren Gebrauch

Lesen Sie vor der ersten Inbetriebnahme der Komponenten des elektrischen Antriebs- und Steuerungssystems folgende Hinweise, damit Sie Körperverletzungen und/oder Sachschäden vermeiden können. Sie müssen diese Sicherheitshinweise einhalten.

- Bei Schäden infolge von Nichtbeachtung der Sicherheitshinweise übernimmt Bosch Rexroth keine Haftung.
- Vor der Inbetriebnahme sind die Betriebs-, Wartungs- und Sicherheitshinweise durchzulesen. Wenn die Anwendungsdokumentation in der vorliegenden Sprache nicht einwandfrei verstanden wird, bitte beim Lieferanten anfragen und diesen informieren.
- Der einwandfreie und sichere Betrieb der Komponente setzt sachgemäßen und fachgerechten Transport, Lagerung, Montage und Installation sowie sorgfältige Bedienung und Instandhaltung voraus.

Sicherheitshinweise für elektrische Antriebe und Steuerungen

- Nur qualifiziertes Personal darf an Komponenten des elektrischen Antriebs- und Steuerungssystems oder in dessen Nähe arbeiten.
- Nur von Bosch Rexroth zugelassene Zubehör- und Ersatzteile verwenden.
- Sicherheitsvorschriften und -bestimmungen des Landes beachten, in welchem die Komponenten des elektrischen Antriebs- und Steuerungssystems betrieben werden.
- Komponenten des elektrischen Antriebs- und Steuerungssystems nur bestimmungsgemäß verwenden. Siehe dazu Kapitel "Bestimmungsgemäßer Gebrauch".
- Die in der vorliegenden Anwendungsdokumentation angegebenen Umgebungs- und Einsatzbedingungen müssen eingehalten werden.
- Anwendungen zur funktionalen Sicherheit sind nur zugelassen, wenn sie ausdrücklich und eindeutig in der Anwendungsdokumentation "Integrierte Sicherheitstechnik" angegeben sind. Ist dies nicht der Fall, sind sie ausgeschlossen. Funktionale Sicherheit beinhaltet Teile der Gesamtsicherheit, bei der Maßnahmen zur Risikominderung zum Personenschutz von elektrischen, elektronischen oder programmierbaren Steuerungen abhängig sind.
- Die in der Anwendungsdokumentation gemachten Angaben zur Verwendung der gelieferten Komponenten stellen nur Anwendungsbeispiele und Vorschläge dar.

Der Maschinenhersteller und Anlagenerrichter muss für seine individuelle Anwendung die Eignung

- der gelieferten Komponenten und die in dieser Anwendungsdokumentation gemachten Angaben zu ihrer Verwendung selbst überprüfen,
- mit den für seine Anwendung geltenden Sicherheitsvorschriften und Normen abstimmen und die erforderlichen Maßnahmen, Änderungen, Ergänzungen durchführen.
- Die Inbetriebnahme der gelieferten Komponenten ist solange untersagt, bis festgestellt wurde, dass die Maschine oder Anlage, in der die Komponenten eingebaut sind, den länderspezifischen Bestimmungen, Sicherheitsvorschriften und Normen der Anwendung entspricht.
- Der Betrieb ist nur bei Einhaltung der nationalen EMV-Vorschriften für den vorliegenden Anwendungsfall erlaubt.
- Die Hinweise für eine EMV-gerechte Installation sind dem Abschnitt zur EMV in der zugehörigen Anwendungsdokumentation zu entnehmen.

Die Einhaltung der durch die nationalen Vorschriften geforderten Grenzwerte liegt in der Verantwortung des Herstellers der Anlage oder Maschine.
- Die technischen Daten, die Anschluss- und Installationsbedingungen der Komponenten sind den zugehörigen Anwendungsdokumentationen zu entnehmen und unbedingt einzuhalten.

Länderspezifische Vorschriften, die vom Anwender zu berücksichtigen sind

- Europäische Länder: entsprechend Euronormen EN
- Vereinigte Staaten von Amerika (USA):
 - Nationale Vorschriften für Elektrik (NEC)
 - Nationale Vereinigung der Hersteller von elektrischen Anlagen (NEMA) sowie regionale Bauvorschriften
 - Vorschriften der National Fire Protection Association (NFPA)

Sicherheitshinweise für elektrische Antriebe und Steuerungen

- Kanada: Canadian Standards Association (CSA)
- Andere Länder:
 - International Organization for Standardization (ISO)
 - International Electrotechnical Commission (IEC)

3.2.3 Gefahren durch falschen Gebrauch

- Hohe elektrische Spannung und hoher Arbeitsstrom! Lebensgefahr oder schwere Körperverletzung durch elektrischen Schlag!
- Hohe elektrische Spannung durch falschen Anschluss! Lebensgefahr oder Körperverletzung durch elektrischen Schlag!
- Gefahrbringende Bewegungen! Lebensgefahr, schwere Körperverletzung oder Sachschaden durch unbeabsichtigte Bewegungen der Motoren!
- Gesundheitsgefahr für Personen mit Herzschrittmachern, metallischen Implantaten und Hörgeräten in unmittelbarer Umgebung elektrischer Antriebssysteme!
- Verbrennungsgefahr durch heiße Gehäuseoberflächen!
- Verletzungsgefahr durch unsachgemäße Handhabung! Körperverletzung durch Quetschen, Scheren, Schneiden, Stoßen!
- Verletzungsgefahr durch unsachgemäße Handhabung von Batterien!
- Verletzungsgefahr durch unsachgemäße Handhabung von unter Druck stehenden Leitungen!

3.3 Gefahrenbezogene Hinweise

3.3.1 Schutz gegen Berühren elektrischer Teile und von Gehäusen



Dieser Abschnitt betrifft Komponenten des elektrischen Antriebs- und Steuerungssystems mit Spannungen **über 50 Volt**.

Werden Teile mit Spannungen größer 50 Volt berührt, können diese für Personen gefährlich werden und zu elektrischem Schlag führen. Beim Betrieb von Komponenten des elektrischen Antriebs- und Steuerungssystems stehen zwangsläufig bestimmte Teile dieser Komponenten unter gefährlicher Spannung.

Hohe elektrische Spannung! Lebensgefahr, Verletzungsgefahr durch elektrischen Schlag oder schwere Körperverletzung!

- Bedienung, Wartung und/oder Instandsetzung der Komponenten des elektrischen Antriebs- und Steuerungssystems darf nur durch qualifiziertes Personal erfolgen.
- Beachten Sie die allgemeinen Errichtungs- und Sicherheitsvorschriften zu Arbeiten an Starkstromanlagen.
- Stellen Sie vor dem Einschalten den festen Anschluss des Schutzleiters an allen elektrischen Komponenten entsprechend dem Anschlussplan her.
- Ein Betrieb, auch für kurzzeitige Mess- und Prüfzwecke, ist nur mit fest angeschlossenem Schutzleiter an den dafür vorgesehenen Punkten der Komponenten erlaubt.
- Trennen Sie elektrische Komponenten vom Netz oder von der Spannungsquelle, bevor Sie auf elektrische Teile mit Spannungen größer 50 V zugreifen. Sichern Sie die elektrische Komponente gegen Wiedereinschalten.
- Bei elektrischen Komponenten beachten:
Warten Sie nach dem Abschalten grundsätzlich **30 Minuten**, damit sich spannungsführende Kondensatoren entladen können, bevor Sie auf eine elektrische Komponente zugreifen. Messen Sie die elektrische Spannung von spannungsführenden Teilen vor Beginn der Arbeiten, um Gefährdungen durch Berührung auszuschließen.
- Bringen Sie vor dem Einschalten die dafür vorgesehenen Abdeckungen und Schutzvorrichtungen für den Berührschutz an.
- Berühren Sie elektrische Anschlussstellen der Komponenten im eingeschalteten Zustand nicht.
- Ziehen Sie Stecker nicht unter Spannung ab oder stecken Sie diese nicht unter Spannung auf.
- Elektrische Antriebssysteme können unter bestimmten Voraussetzungen an Netzen betrieben werden, die durch allstromsensitive FI-Schutzeinrichtungen (RCD/RCM) abgesichert sind.
- Für Einbaugeräte ist der Schutz gegen Eindringen von Fremdkörpern und Wasser sowie gegen direktes Berühren durch ein äußeres Gehäuse, z. B. Schaltschrank, sicherzustellen.

Hohe Gehäusespannung und hoher Ableitstrom! Lebensgefahr, Verletzungsgefahr durch elektrischen Schlag!

Sicherheitshinweise für elektrische Antriebe und Steuerungen

- Erden oder verbinden Sie vor dem Einschalten und der Inbetriebnahme die Komponenten des elektrischen Antriebs- und Steuerungssystems mit dem Schutzleiter an den Erdungspunkten.
- Schließen Sie den Schutzleiter der Komponenten des elektrischen Antriebs- und Steuerungssystems stets fest und dauerhaft an das Versorgungsnetz an. Der Ableitstrom ist größer als 3,5 mA.
- Stellen Sie eine Schutzleiterverbindung mit einem Mindestquerschnitt gemäß nachfolgender Tabelle her. Bei einem Außenleiterquerschnitt kleiner 10 mm² ist alternativ auch ein Anschluss von zwei Schutzleitern zulässig, welche jeweils den gleichen Querschnitt wie die Außenleiter aufweisen.

Querschnitt Außenleiter	Mindestquerschnitt Schutzleiter Ableitstrom $\geq 3,5$ mA	
	1 Schutzleiter	2 Schutzleiter
1,5 mm ² (AWG 16)	10 mm ² (AWG 8)	2 × 1,5 mm ² (AWG 16)
2,5 mm ² (AWG 14)		2 × 2,5 mm ² (AWG 14)
4 mm ² (AWG 12)		2 × 4 mm ² (AWG 12)
6 mm ² (AWG 10)		2 × 6 mm ² (AWG 10)
10 mm ² (AWG 8)		-
16 mm ² (AWG 6)	16 mm ² (AWG 6)	-
25 mm ² (AWG 4)		-
35 mm ² (AWG 2)		-
50 mm ² (AWG 1/0)	25 mm ² (AWG 4)	-
70 mm ² (AWG 2/0)	35 mm ² (AWG 2)	-
...	...	...

Abb.3-1: Mindestquerschnitt der Schutzleiterverbindung

3.3.2 Schutzkleinspannung als Schutz gegen elektrischen Schlag

Schutzkleinspannung dient dazu, Geräte mit Basisisolierung an Kleinspannungskreise anschließen zu können.

An den Komponenten des elektrischen Antriebs- und Steuerungssystems von Bosch Rexroth sind alle Anschlüsse und Klemmen, die Spannungen von 5 bis 50 Volt führen, in Schutzkleinspannung ("Protective Extra Low Voltage - PELV") ausgeführt. An diese Anschlüsse dürfen Geräte angeschlossen werden, die mit Basisisolierung ausgestattet sind, wie beispielsweise Programmiergeräte, PCs, Notebooks, Anzeigergeräte.

Lebensgefahr, Verletzungsgefahr durch elektrischen Schlag! Hohe elektrische Spannung durch falschen Anschluss!

Werden Kleinspannungskreise von Geräten, die auch Spannungen und Stromkreise über 50 Volt beinhalten (z. B. den Netzanschluss), an Produkten von Bosch Rexroth angeschlossen, dann müssen die angeschlossenen Kleinspannungskreise die Anforderungen für Schutzkleinspannung ("Protective Extra Low Voltage - PELV") erfüllen.

3.3.3 Schutz vor gefährlichen Bewegungen

Gefährliche Bewegungen können durch fehlerhafte Ansteuerung von angeschlossenen Motoren verursacht werden. Die Ursachen können verschiedenster Art sein:

- unsachgemäße oder fehlerhafte Verdrahtung oder Verkabelung
- Bedienungsfehler
- falsche Eingabe von Parametern vor der Inbetriebnahme
- Fehler in den Messwertgebern und Signalgebern
- defekte Komponenten
- Fehler in der Software oder in der Firmware

Diese Fehler können unmittelbar nach dem Einschalten oder nach einer unbestimmten Zeitdauer im Betrieb auftreten.

Die Überwachungen in den Komponenten des elektrischen Antriebs- und Steuerungssystems schließen eine Fehlfunktion in den angeschlossenen Antrieben weitestgehend aus. Im Hinblick auf den Personenschutz, insbesondere auf die Gefahr von Körperverletzung und/oder Sachschaden, darf auf diesen Sachverhalt nicht allein vertraut werden. Bis zum Wirksamwerden der eingebauten Überwachungen ist auf jeden Fall mit einer fehlerhaften Antriebsbewegung zu rechnen, deren Maß von der Art der Steuerung und des Betriebszustandes abhängt.

Gefahrbringende Bewegungen! Lebensgefahr, Verletzungsgefahr, schwere Körperverletzung oder Sachschaden!

Für die Anlage oder Maschine mit ihren spezifischen Gegebenheiten, in welche die Komponenten des elektrischen Antriebs- und Steuerungssystems eingebaut werden, ist eine **Risikobeurteilung** zu erstellen.

Aus der Risikobeurteilung sind vom Anwender Überwachungen und anlagenseitig übergeordnete Maßnahmen für den Personenschutz vorzusehen. Die für die Anlage oder Maschine geltenden Sicherheitsbestimmungen sind hierbei mit einzubeziehen. Durch Ausschalten, Umgehen oder fehlendes Aktivieren von Sicherheitseinrichtungen können willkürliche Bewegungen der Maschine oder andere Fehlfunktionen auftreten.

Vermeidung von Unfällen, Körperverletzung und/oder Sachschaden:

- Halten Sie sich nicht im Bewegungsbereich der Maschine und Maschinenteilen auf. Verhindern Sie den unbeabsichtigten Zugang für Personen, z. B. durch
 - Schutzzaun
 - Schutzgitter
 - Schutzabdeckung
 - Lichtschranke
- Stellen Sie eine ausreichende Festigkeit der Schutzzäune und Schutzabdeckungen gegen die maximal mögliche Bewegungsenergie sicher.
- Ordnen Sie NOT-HALT-Schalter leicht zugänglich und schnell erreichbar an. Prüfen Sie die Funktion der NOT-HALT-Einrichtung vor der Inbetriebnahme. Unterlassen Sie den Betrieb der Maschine bei Fehlfunktion des NOT-HALT-Schalters.

Sicherheitshinweise für elektrische Antriebe und Steuerungen

- Stellen Sie sicher, dass es nicht zu einem unbeabsichtigten Anlauf kommt. Schalten Sie den Leistungsanschluss der Antriebe über Ausschalter/-taster frei oder verwenden Sie eine sichere Anlaufsperre.
- Bringen Sie vor dem Zugriff oder Zutritt in den Gefahrenbereich die Antriebe sicher zum Stillstand.
- Sichern Sie zusätzlich vertikale Achsen gegen Herabfallen oder Absinken nach Abschalten des Motors, z. B. durch
 - mechanische Verriegelung der vertikalen Achse,
 - externe Brems-/ Fang-/ Klemmeinrichtung oder
 - ausreichenden Gewichtsausgleich der Achse.
- Die serienmäßig gelieferte **Motor-Haltebremse** oder eine externe, vom Antriebsregelgerät angesteuerte Haltebremse **alleine ist nicht für den Personenschutz geeignet!**
- Schalten Sie die Komponenten des elektrischen Antriebs- und Steuerungssystems über den Hauptschalter spannungsfrei und sichern Sie diese gegen Wiedereinschalten bei:
 - Wartungsarbeiten und Instandsetzung
 - Reinigungsarbeiten
 - langen Betriebsunterbrechungen
- Vermeiden Sie den Betrieb von Hochfrequenz-, Fernsteuer- und Funkgeräten in der Nähe von Komponenten des elektrischen Antriebs- und Steuerungssystems und deren Zuleitungen. Wenn ein Gebrauch dieser Geräte unvermeidlich ist, prüfen Sie bei Erstinbetriebnahme des elektrischen Antriebs- und Steuerungssystems die Maschine oder Anlage auf mögliche Fehlfunktionen bei Betrieb solcher Hochfrequenz-, Fernsteuer- oder Funkgeräte in deren möglichen Gebrauchslagen. Eventuell ist eine spezielle EMV-Prüfung notwendig.

3.3.4 Schutz vor magnetischen und elektromagnetischen Feldern bei Betrieb und Montage

Magnetische und elektromagnetische Felder, die in unmittelbarer Umgebung von stromführenden Leitern oder Permanentmagneten von Elektromotoren bestehen, können eine ernste Gefahr für Personen mit Herzschrittmachern, metallischen Implantaten und Hörgeräten darstellen.

Gesundheitsgefahr für Personen mit Herzschrittmachern, metallischen Implantaten und Hörgeräten in unmittelbarer Umgebung elektrischer Komponenten!

- Personen mit Herzschrittmachern und metallischen Implantaten ist der Zugang zu folgenden Bereichen untersagt:
 - Bereiche, in denen Komponenten der elektrischen Antriebs- und Steuerungssysteme montiert, in Betrieb genommen und betrieben werden
 - Bereiche, in denen Motorenteile mit Dauermagneten gelagert, repariert oder montiert werden
- Besteht die Notwendigkeit für Träger von Herzschrittmachern derartige Bereiche zu betreten, so ist das zuvor von einem Arzt zu entscheiden. Die Störfestigkeit von implantierten Herzschrittmachern ist sehr unterschiedlich, somit bestehen keine allgemein gültigen Regeln.
- Personen mit Metallimplantaten oder Metallsplintern sowie mit Hörgeräten haben vor dem Betreten derartiger Bereiche einen Arzt zu befragen.

3.3.5 Schutz gegen Berühren heißer Teile

Heiße Oberflächen von Komponenten des elektrischen Antriebs- und Steuerungssystems. Verbrennungsgefahr!

- Vermeiden Sie das Berühren von heißen Oberflächen von z. B. Bremswiderständen, Kühlkörpern, Versorgungs- und Antriebsregelgeräten, Motoren, Wicklungen und Blechpaketen!
- Temperaturen der Oberflächen können während oder nach dem Betrieb je nach Betriebsbedingungen **über 60 °C** (140 °F) liegen.
- Lassen Sie die Motoren nach dem Abschalten ausreichend lange abkühlen, bevor Sie diese berühren. Abkühlzeiten **bis 140 Minuten** können erforderlich sein! Die erforderliche Abkühlzeit ist ungefähr fünfmal so groß wie die in den technischen Daten angegebene thermische Zeitkonstante.
- Lassen Sie Drosseln, Versorgungs- und Antriebsregelgeräte **15 Minuten** lang nach dem Abschalten abkühlen, bevor Sie diese berühren.
- Tragen Sie Schutzhandschuhe oder arbeiten Sie nicht an heißen Oberflächen.
- Für bestimmte Anwendungen sind nach den Sicherheitsvorschriften Maßnahmen zur Verhinderung von Verbrennungsverletzungen in der Endanwendung vom Hersteller der Maschine oder Anlage vorzunehmen. Diese Maßnahmen können beispielsweise sein: Warnhinweise an der Maschine oder Anlage, trennende Schutzeinrichtung (Abschirmung oder Absperrung) oder Sicherheitshinweise in der Anwendungsdokumentation.

3.3.6 Schutz bei Handhabung und Montage

Verletzungsgefahr durch unsachgemäße Handhabung! Körperverschletzung durch Quetschen, Scheren, Schneiden, Stoßen!

- Beachten Sie die einschlägigen Vorschriften zur Verhütung von Unfällen (z. B. Unfallverhütungsvorschriften).
- Verwenden Sie geeignete Montage- und Transporteinrichtungen.
- Beugen Sie Einklemmungen und Quetschungen durch geeignete Vorkehrungen vor.
- Benutzen Sie nur geeignetes Werkzeug, sofern vorgeschrieben, Spezialwerkzeug.
- Setzen Sie Hebeeinrichtungen und Werkzeuge fachgerecht ein.
- Benutzen Sie geeignete Schutzausstattung (z. B. Schutzhelm, Schutzbrille, Sicherheitsschuhe, Schutzhandschuhe).
- Halten Sie sich nicht unter hängenden Lasten auf.
- Beseitigen Sie ausgelaufene Flüssigkeiten am Boden sofort, ansonsten besteht Sturzgefahr!

3.3.7 Schutz beim Umgang mit Batterien

Batterien bestehen aus aktiven Chemikalien in einem festen Gehäuse. Unsachgemäßer Umgang kann daher zu Verletzungen oder Sachschäden führen.

Verletzungsgefahr durch unsachgemäße Handhabung!

Sicherheitshinweise für elektrische Antriebe und Steuerungen

- Versuchen Sie nicht, leere Batterien durch Erhitzen oder andere Methoden zu reaktivieren (Explosions- und Ätzungsgefahr).
- Versuchen Sie nicht, Batterien aufzuladen, weil sie dabei auslaufen oder explodieren können.
- Werfen Sie Batterien nicht ins Feuer.
- Zerlegen Sie keine Batterie.
- Beschädigen Sie beim Wechsel der Batterie(n) nicht die elektrischen Bauteile in den Geräten.
- Verwenden Sie nur die für das Produkt vorgeschriebenen Batterietypen.



Umweltschutz und Entsorgung! Die im Produkt enthaltenen Batterien sind im Sinne der gesetzlichen Bestimmungen als Gefahrgut beim Transport im Land-, Luft- und Seeverkehr anzusehen (Explosionsgefahr). Entsorgen Sie Altbatterien getrennt von anderem Abfall. Beachten Sie die nationalen Bestimmungen Ihres Landes.

3.3.8 Schutz vor unter Druck stehenden Leitungen

Flüssigkeits- und druckluftgekühlte Motoren und Komponenten können entsprechend den Angaben in den Projektierungsanleitungen zum Teil mit extern zugeführten und unter Druck stehenden Medien wie Druckluft, Hydrauliköl, Kühlflüssigkeit und Kühlschmiermittel versorgt werden. Unsachgemäßer Umgang mit den angeschlossenen Versorgungssystemen, Versorgungsleitungen oder Anschlüssen kann zu Verletzungen oder Sachschäden führen.

Verletzungsgefahr durch unsachgemäße Handhabung von unter Druck stehenden Leitungen!

- Versuchen Sie nicht, unter Druck stehende Leitungen zu trennen, zu öffnen oder zu kappen (Explosionsgefahr).
- Beachten Sie die Betriebsvorschriften der jeweiligen Hersteller.
- Lassen Sie vor Demontage von Leitungen, Druck und Medium ab.
- Benutzen Sie geeignete Schutzausrüstung (z. B. Schutzbrillen, Sicherheitsschuhe, Schutzhandschuhe).
- Beseitigen Sie ausgelaufene Flüssigkeiten am Boden sofort, ansonsten besteht Sturzgefahr!



Umweltschutz und Entsorgung! Die für den Betrieb des Produktes verwendeten Medien können unter Umständen nicht umweltverträglich sein. Entsorgen Sie umweltschädliche Medien getrennt von anderem Abfall. Beachten Sie die nationalen Bestimmungen Ihres Landes.

3.4 Erläuterung der Signalwörter und der Signalgrafik

Die Sicherheitshinweise in der vorliegenden Anwendungsdokumentation beinhalten bestimmte Signalwörter (Gefahr, Warnung, Vorsicht, Hinweis) und ggf. eine Signalgrafik (nach ANSI Z535.6-2006).

Das Signalwort soll die Aufmerksamkeit auf den Sicherheitshinweis lenken und bezeichnet die Schwere der Gefährdung.

Sicherheitshinweise für elektrische Antriebe und Steuerungen

Die Signalgrafik (Warndreieck mit Ausrufezeichen), welche den Signalwörtern Gefahr, Warnung und Vorsicht vorangestellt wird, weist auf Gefährdungen für Personen hin.

⚠ GEFAHR

Bei Nichtbeachtung dieses Sicherheitshinweises **werden** Tod oder schwere Körperverletzung eintreten.

⚠ WARNUNG

Bei Nichtbeachtung dieses Sicherheitshinweises **können** Tod oder schwere Körperverletzung eintreten.

⚠ VORSICHT

Bei Nichtbeachtung dieses Sicherheitshinweises können mittelschwere oder leichte Körperverletzung eintreten.

HINWEIS

Bei Nichtbeachtung dieses Sicherheitshinweises können Sachschäden eintreten.

4 Übersicht

4.1 Bibliotheken und Zielsysteme

Die nachfolgende Tabelle zeigt die unterstützten Zielsysteme für die Technologie-Bibliotheken.

Bibliothek	Zielsystem IndraMotion MLC	Zielsystem IndraLogic XLC	Zielsystem IndraMotion MLD	Beschrieben in
ML_TechBase.compiled-library	X	X		(Kap. 5 "ML_TechBase.library" auf Seite 33)
ML_TechMotion.compiled-library	X			(Kap. 6 "ML_TechMotion.library" auf Seite 165)
ML_TechRegi.compiled-library MX_TechRegi.lib	X		X	(Kap. 7 "ML_TechRegi.library/MX_TechRegi.lib" auf Seite 297)
ML_TechCrank.compiled-library	X			(Kap. 8 "ML_TechCrank.library" auf Seite 387)
ML_TechTensionAdvanced.compiled-library	X			(Kap. 13 "ML_TechTensionAdvanced.library" auf Seite 591)
RMB_TechWinder.compiled-library MX_TechWinder.lib	X		X	(Kap. 9 "RMB_TechWinder.library/MX_TechWinder.lib" auf Seite 425)
RMB_CrosscutCrossseal.compiled-library	X			(Kap. 10 "RMB_TechCrossCutCrossSeal.library" auf Seite 515)
RMB_TechCam.compiled-library	X			(Kap. 11 "RMB_TechCam.library" auf Seite 545)
RIL_ParameterChannel.compiled-library	X	X		(Kap. 12 "RIL_ParameterChannel.library" auf Seite 567)

Abb.4-1: Von den Technologie-Bibliotheken unterstützte Zielsysteme

4.2 Besonderheiten von IndraMotion MLD

4.2.1 Allgemeines

IndraMotion MLD ist eine "OnBoard"-Steuerung die in der Hardware des IndraDrive-Steuerteils integriert ist. Die Firmware der SPS ist dann ein Teil der Antriebsfirmware. Die SPS-Entwicklungsumgebung ist im Unterschied zu IndraMotion MLC IndraLogic 1.X.

4.2.2 Antriebsfirmware, Funktionspakete

Die Antriebsfirmware gibt es ab 07VRS in folgenden Varianten:

Übersicht

Funktionspakete	• MPC-Firmware: Mehrachs-MotionControl IndraMotion MLD-M für bis zu 8 Achsen • MPH-Firmware: Einachs-Steuerung IndraMotion MLD-S für 1 Achse • MPD-Firmware: Keine IndraMotion MLD-Funktionalität möglich • MPB-Firmware: Einachs-Steuerung IndraMotion MLD-S für 1 Achse Die Funktionalität der Antriebsfirmware ist in mehrere Funktionspakete unterteilt. Die Funktionspakete müssen der Applikation entsprechend bestellt und lizenziert werden. Die benötigten Funktionspakete sind dann freigeschaltet.
	 Für IndraMotion MLD muss das Funktionspaket "IndraMotion MLD" freigeschaltet sein. Für erweiterte Technologiebausteine muss das Zusatzpaket "IndraMotion MLD Advanced" (abgekürzt "MA") freigeschaltet sein.
	 Das Zusatzpaket "IndraMotion MLD Advanced" ist nur für MPC-Firmware und MPH-Firmware verfügbar (nicht für MPB-Firmware).
Mehrachssteuerung IndraMotion MLD-M	Technologiebibliotheken die z.B. das Zusatzpaket "IndraMotion MLD Advanced" voraussetzen: • MX_TechWinder • MX_TechRegi Für die Mehrachs-Steuerung IndraMotion MLD-M wird ab der Antriebsfirmware 07VRS die Antriebsfirmware MPC benötigt. Für diese Firmware wird das Steuerteil CSH01.3 benötigt. Zusätzlich muss in der SPS-Konfiguration eingestellt werden, dass die SPS "permanente Kontrolle" (P-0-1367 "SPS-Konfiguration" Bit 4 gesetzt, bzw. Dialog für SPS-Konfiguration) über den Antrieb hat.  Für die MPC Firmware wird ab MPX07 das Steuerteil CSH01.3 benötigt.

5 ML_TechBase.library

5.1 Einführung und Übersicht

Allgemeines Technologie-Funktionsbausteine (Tech-FB) erweitern die Grundfunktionalität der Zielsysteme IndraMotion MLC/IndraLogic XLC und IndraMotion MLD und stellen applikationsspezifische Funktionalitäten zur Verfügung. Die hier beschriebene Bibliothek beinhaltet Basis-Funktionalitäten die für viele verschiedene Anwendungen zum Einsatz kommen, aber auch einzeln verwendet werden können.

Die hier beschriebenen Funktionsbausteine werden über die Bibliothek "ML_TechBase.compiled-library" für IndraMotion MLC/IndraLogic XLC zur Verfügung gestellt.

Diese Dokumentation beschreibt die Funktionalität, sowie die Ein- und Ausgänge der Technologie-Funktionsbausteine.

Voraussetzungen Technologie-Funktionsbausteine der Bibliothek "ML_TechBase.compiled-library" benötigen die Firmware-Unterstützung des Zielsystems. Spezifische Voraussetzungen der Technologie-Funktionsbausteine sind im Kapitel der entsprechenden Funktionsbausteine dokumentiert.

Alias In der Bibliothek befindet sich im Verzeichnis "Global variables" auch ein Eintrag "FP_Aliases". Dabei handelt es sich um Bezeichner, hinter denen Parameter stehen. Die Parameter sind dabei systemabhängig. Der Anwender sollte bevorzugt diese Bezeichner verwenden, da er damit unabhängig vom gewählten Zielsystem ist. So wird beispielsweise der Bezeichner "FP_MODULO" auf einer IndraMotion MLC/IndraLogic XLC auf den Achsparameter A-0-0045 gesetzt, auf einem IndraMotion MLD System auf den SERCOS-Parameter S-0-0103. Auf Applikationsebene kann man damit das verwendete System besser abstrahieren und allgemein gültigen Code schreiben.

Übersicht über die Funktionsbausteine

Funktionsbaustein	Beschreibung
Vertippen von Variablen	
MB_ContinuousAdjustType01	Kontinuierliche Verstellung einer REAL-Variablen über binäre Eingänge
MB_ContinuousAdjustType02	Kontinuierliche Verstellung einer zu beeinflussenden DINT-Variable über binäre Eingänge
MB_IncrementalAdjustType01	Inkrementelle Verstellung einer REAL-Variablen über binäre Eingänge
Zweipunktregler	
ML_TwoPosCtrlType01	realisiert einen Zweipunktregler mit Hysterese
Mathematische Funktionen	
ML_InterpolationLinear	errechnet gemäß Stützstellentabelle zu gegebenem X-Wert zyklisch den zugehörigen Y-Wert
ML_MaxValue	ermittelt aus dem zeitlichen Verlauf eines Eingangssignals den bisherigen Maximalwert
Anwendung "Nockenschaltwerk"	
MC_DigitalCamSwitch	implementiert ein frei programmierbares Nockenschaltwerk mit bis zu 32 Nocken auf 8 Ausgängen

ML_TechBase.library

Funktionsbaustein	Beschreibung
Temperaturregler	
IL_TempControlType01	dient zur Regelung von Temperaturen, inkl. Komfortfunktionen wie Überwachung, Pause, Handbetrieb
Messtasterfunktionsbausteine	
MB_InitTouchProbe	Initialisierung der Messtasterfunktionalität
MB_TouchProbe	Erfassen einer Achsposition, Leitachsposition oder Zeiterfassung, Einzelmessung
MB_TouchProbeContinuous	Erfassen einer Achsposition, Leitachsposition oder Zeiterfassung bei fortlaufender Messung
MB_WriteExpectWindow	konfiguriert das Erwartungsfenster für die Messtasterfunktionalität
MB_AbortTrigger	Abbruch des Funktionsbausteins MB_TouchProbe
Trigonometrische Funktionen	
ML_DegToInc	wandelt Winkel vom Gradmaß in entsprechenden Inkrementwert
ML_DegToRad	konvertiert Winkel im Gradmaß in Winkel in Bogenmaß inkl. Modulo
ML_IncToDeg	wandelt einen in Inkrementen vorgegebenen Winkel ins Gradmaß
ML_IncToRad	konvertiert einen in Inkrementen vorgegebenen Winkel in das entsprechende Bogenmaß
ML_RadToDeg	konvertiert einen Winkel im Bogenmaß in den entsprechenden Winkel im Gradmaß inkl. Modulo
ML_RadToInc	wandelt einen im Bogenmaß gegebenen Winkel in den entsprechenden Inkrementwert
Verwaltung "zyklischer Datenkanäle"	
MB_AllocCyclicParameter	Überprüfung und Überarbeitung der Konfiguration eines zyklisch zu lesenden oder beschreibenden Parameters im zyklischen Datenkanal
MB_FreeCyclicParameter	Freigabe eines bereits konfigurierten zyklisch zu lesenden oder beschreibenden Parameters im zyklischen Datenkanal
MB_GetCyclicChannelConfig	liest die Konfiguration zyklischer Lese- und Schreibdatenkanäle aus und speichert sie in einer Datenstruktur
MB_GetCyclicParameterHandle	sucht im zyklischen Datenkanal nach einem zyklisch zu lesenden oder beschreibenden Parameter und legt alle benötigten Informationen in einem Handle ab
MB_ReadCyclicParameter	liest einen zyklischen Parameter über den Handle und gibt den aktuellen Inhalt als DINT zurück
MB_ReadCyclicRealParameter	liest einen zyklischen Parameter über den Handle und gibt den aktuellen Inhalt als REAL zurück
MB_WriteCyclicParameter	schreibt einen zyklischen Parameter über den Handle, Nachkommastellen werden gemäß SERCOS gewandelt

Funktionsbaustein	Beschreibung
MB_WriteCyclicRealParameter	schreibt einen zyklischen Parameter über den Handle, Nachkommastellen werden gerundet
MB_GetLastCyclicParameterError	gibt den letzten aufgetretenen Fehler beim Lesen oder Schreiben eines zyklischen Werts für einen Handle zurück und löscht den Fehler
Achsen Sichern und Wiederherstellen	
ML_AxisBackup	Parametersicherung einer Achse auf der Kompakt-Flash-Karte
ML_AxisRestore	Wiederherstellung von Parametern einer Achse von der Kompakt-Flash-Karte
Logik-Anwendungen	
IL_LinkedList	Verwaltung einer doppelt verketteten Liste
Berechnung von Modulowerten	
MB_GetModuloType01	liefert Modulowert (Wertebereich zwischen 0 und +Modulowert) zurück (Datentyp REAL)
MB_GetModuloType02	liefert Modulowert (Wertebereich zwischen 0 und +Modulowert) zurück (Datentyp DINT)
MB_GetModuloType03	liefert Modulowert (Wertebereich zwischen -Modulowert und +Modulowert) zurück
MB_GetModuloType04	liefert Modulowert (Wertebereich zwischen -modulo/2 und +modulo/2) zurück
Hilfsfunktionen	
MB_GetSystemInfo	überprüft Funktionspakete und Systemoptionen
Download von Kurvenscheibentabellen	
MB_LoadCamData	übernimmt den Download von Kurvenscheiben auf den Antrieb und die Steuerung, um die vorhandenen Detailunterschiede zu kapseln

Abb.5-1: Funktionsbausteine in Bibliothek ML_TechBase

5.2 Funktionsbausteine zur Anwendung "Vertippen von Variablen"

5.2.1 Einführung und Übersicht

Mit den nachstehend genannten Funktionsbausteinen können SPS-Variable über binäre Eingänge kontinuierlich oder inkrementell geändert (vertippt) werden:

- [Kap. 5.2.2 "MB_ContinuousAdjustType01" auf Seite 38](#)
- [Kap. 5.2.3 "MB_ContinuousAdjustType02" auf Seite 42](#)
- [Kap. 5.2.4 "MB_IncrementalAdjustType01" auf Seite 46](#)

Bei Verwendung der oben aufgeführten Funktionsbausteine zum Vertippen von SPS-Variablen sind die folgenden Bedingungen zu berücksichtigen:

- Der Betrieb muss über einen binären Eingang ("Enable") gestoppt werden

ML_TechBase.library

- Mit den Funktionsbausteinen MB_ContinuousAdjustType01 und MB_ContinuousAdjustType02 kann die zu beeinflussende Variable kontinuierlich verändert werden (ähnlich dem langen Tippen bei SYNAX)
- Mit dem Funktionsbaustein MB_IncrementalAdjustType01 kann die zu beeinflussende Variable schrittweise verändert werden (ähnlich dem kurzen Tippen bei SYNAX)
- Die gewählte Variable kann innerhalb der Grenzwerte in aufsteigender oder absteigender Schrittfolge verändert werden
- Mit Erreichen eines Grenzwertes ("HighLimitActive" = TRUE oder "LowLimitActive" = TRUE) wird die kontinuierliche Verstellung deaktiviert, d.h., der Funktionsbaustein beeinflusst die Variable nicht weiter. In diesem Fall wird der entsprechende Grenzwert am Eingang "HighLimit" bzw. "LowLimit" ausgegeben

Allgemeine Grundlagen

Besitzen bei einem angegebenen Modulwert die Grenzen "HighLimit" und "LowLimit" den gleichen Wert, wird "LowLimit" gleich Null und "HighLimit" gleich dem Modulwert gesetzt. Somit ist es möglich, die Grenzwerte über den gesamten Modulobereich zu verstellen.

Die Änderungsgeschwindigkeit beschreibt die Rate, mit der sich die beeinflusste Variable während des Stellvorgangs ändert. Diese Rate ist abhängig von der Schrittweite und der Anzahl der Schritte pro Sekunde. Die Änderungsgeschwindigkeit errechnet sich z.B. beim Ändern einer Position wie folgt:

$$\text{Änderungsgeschwindigkeit} \left[\frac{\text{Degree}}{s} \right] = \text{Increments} [\text{Degree}] * \text{Increments per Second} \left[\frac{1}{s} \right]$$

Abb.5-2: Geschwindigkeitsänderung bei Positionsänderung

Wird z.B. die Drehzahl der virtuellen Leitachse verändert, errechnet sich die Änderungsgeschwindigkeit wie folgt:

$$\text{Änderungsgeschwindigkeit} \left[\frac{r.p.m.}{s} \right] = \text{Increments} [r.p.m.] * \text{Increments per Second} \left[\frac{1}{s} \right]$$

Abb.5-3: Geschwindigkeitsänderung beim Ändern einer Achsdrehzahl

Solange das Stellsignal anliegt, wird die beeinflusste Variable mit der definierten Geschwindigkeit verändert.

Stellgrenzen

Die Verstellung kann nur über die entsprechenden Eingänge innerhalb der festgelegten Grenzwerte erfolgen.

Ausnahme:

Befindet sich der Anfangswert der zu verändernden Größe außerhalb der vorgegebenen Grenzen, kann die beeinflusste Variable nur in Richtung des Zielbereiches verändert werden. Wird ein Modulwert angegeben, kann die beeinflusste Variable in beiden Richtungen verändert werden. Nach dem Erreichen des Zielbereichs ist es nicht mehr möglich, diesen zu verlassen.

Preset

Mit Hilfe der Eingänge "Preset" und "PresetValue" kann die Ausgangsgröße auf einen definierten Wert vorbesetzt werden. Der Ausgang "PresetAck" wird gesetzt, sobald der "Presetvalue" übernommen wurde. Die Ausgangsgröße wird auf dem "PresetValue" gehalten, solange der Eingang "Preset" gesetzt ist.

Grundregeln

- Sind beide Stellsignale ("Inc"/"Dec") gleichzeitig TRUE, werden beide als FALSE bewertet

- Ein Stellsignal ändert die Polarität und löst einen sofortigen Richtungswechsel aus. Ebenso wird die aktuelle Verstellung nicht beendet
- Wird die Stellgeschwindigkeit während einer Bewegung umgeschaltet, verhalten sich die Funktionsbausteine

MB_ContinuousAdjustType01/MB_ContinuousAdjustType02 und MB_IncrementalAdjustType01 unterschiedlich.

Bei MB_ContinuousAdjustType01 und MB_ContinuousAdjustType02 ist die Änderung sofort wirksam. Im Gegensatz dazu wird die Bewegung bei MB_IncrementalAdjustType01 beendet, bevor die neue Schrittgeschwindigkeit übernommen wird. Ein nachfolgender Stellvorgang wird dann mit den geänderten Größen durchgeführt.

Für den Funktionsbaustein MB_IncrementalAdjustType01 gelten ferner folgende Regeln:

- Die Eingänge "Inc" und "Dec" werden flankenbezogen ausgewertet. Eine steigende Flanke löst eine Anpassung um die vorgegebene Schrittweite ("StepWidth") aus
- Ein während einer laufenden Bewegung erkannt neues Stellsignal (positive Flanke) mit gleicher Polarität bewirkt eine weitere Bewegung im Anschluss an die Aktuelle

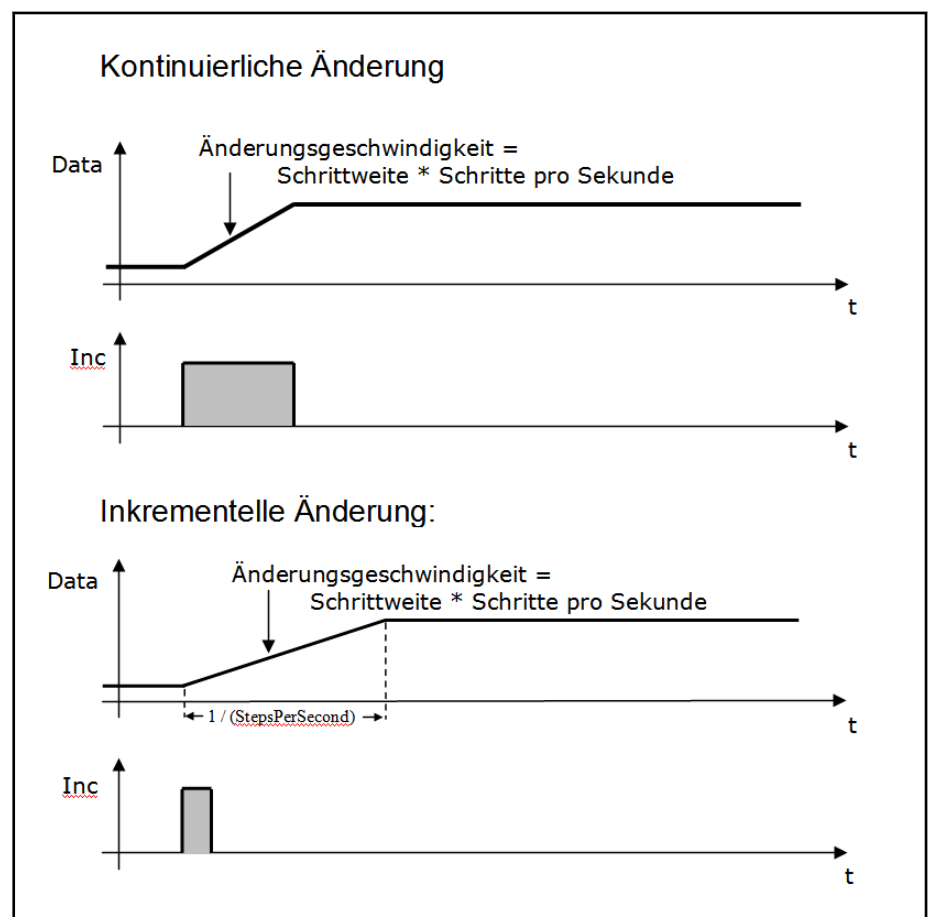


Abb.5-4: Vergleich des Verhaltens von kontinuierlicher und inkrementeller Änderung

Die Funktionsbausteine MB_ContinuousAdjustType01, MB_ContinuousAdjustType02 und MB_IncrementalAdjustType01 initialisieren bei der ersten Ak-

tivierung ("Enable" = TRUE) unabhängig von den anstehenden Eingängen die zur Bestimmung der Zykluszeit notwendigen Variablen. Im zweiten Zyklus steht dann eine berechnete Zykluszeit zur Verfügung, die notwendig ist, um die korrekte Änderungsgeschwindigkeit zu berechnen. Anschließend wird die Zykluszeit fortlaufend aktualisiert.

Durch die Angabe eines Modulowertes liefert der Funktionsbaustein eine Ausgangsgröße, die sich zwischen Null und dem angegebenen Modulowert bewegt. Bei Überschreiten des Modulowertes wird lediglich die Ausgangsgröße auf Null zurückgesetzt, der Verstellvorgang jedoch fortgesetzt. Der vorgegebene gültige Bereich kann dementsprechend zwischen Null und dem Modulowert liegen. Der Bereich kann um den Nullwert verstellt werden.

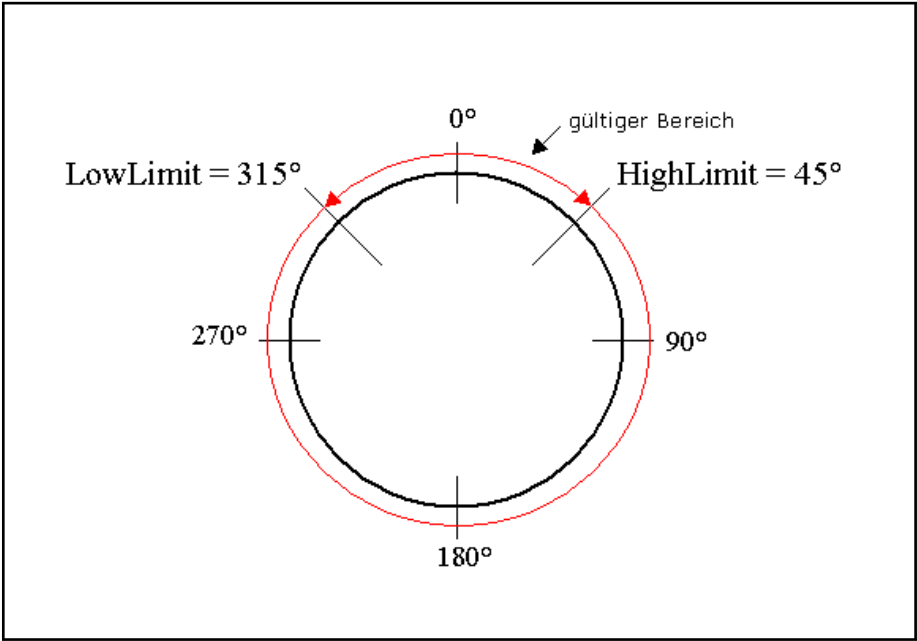


Abb.5-5: Mögliche Zuordnung eines gültigen Bereichs bei Angabe eines Modulowertes von 360°

Zusätzlich bietet der Funktionsbaustein MB_ContinuousAdjustType01 bzw. MB_ContinuousAdjustType02 die Möglichkeit, die Änderungsgeschwindigkeit durch die Angabe von Wichtungsfaktoren und Zeitintervallen zeitabhängig zu verändern.

5.2.2 MB_ContinuousAdjustType01

Kurzbeschreibung Der Funktionsbaustein MB_ContinuousAdjustType01 ermöglicht die kontinuierliche Verstellung einer REAL-Variablen über binäre Eingänge.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD ab MPx18 (in Vorbereitung)	in Vorbereitung

Abb.5-6: Referenztablette Funktionsbaustein MB_ContinuousAdjustType01

Schnittstellenbeschreibung

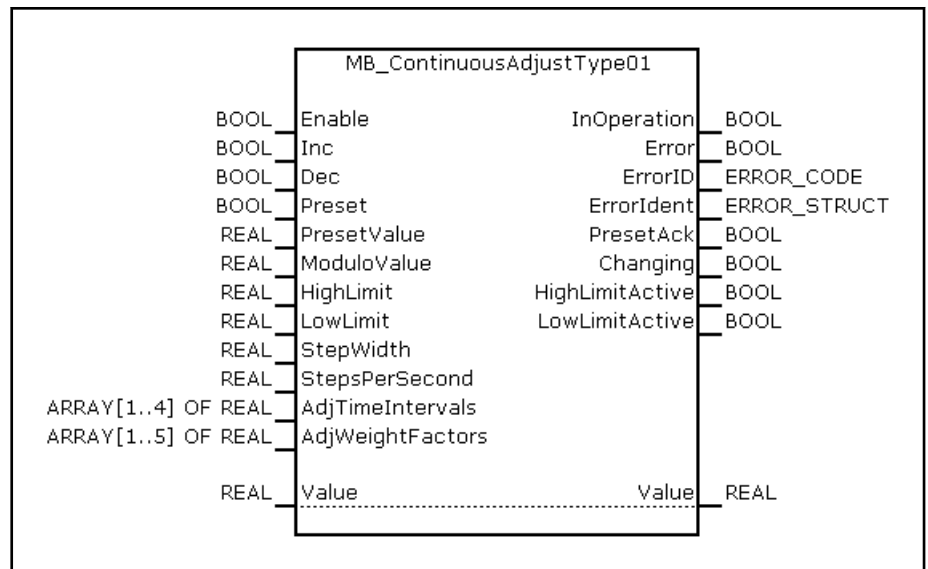


Abb.5-7: Funktionsbaustein MB_ContinuousAdjustType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Freigabe des Funktionsbausteins (zyklisch, zustands-gesteuert)
	Inc	BOOL	Beeinflusste Variable erhöhen
	Dec	BOOL	Beeinflusste Variable verringern
	Preset	BOOL	Voreinstellwert (PresetValue) setzen
	PresetValue	REAL	Vorgabewert für den Voreinstellwert
	ModuloValue	REAL	Modulowert (wenn 0, dann absolute Verarbeitung)
	HighLimit	REAL	Maximaler Ausgangswert der zu beeinflussenden Va-riablen
	LowLimit	REAL	Minimaler Ausgangswert der zu beeinflussenden Va-riablen
	StepWidth	REAL	Schrittweite
	StepsPerSecond	REAL	Anzahl der Schritte pro Sekunde
	AdjTimeIntervals	ARRAY [1..4] OF REAL	Zeitintervalle in Sekunden zur Wichtung der Ände-rungsgeschwindigkeit
	AdjWeightFactors	ARRAY [1..5] OF REAL	Wichtungsfaktoren der Änderungsgeschwindigkeit
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein aktiv. Die Ausgänge "HighLimitActi-ve" und "LowLimitActive" sind gültig
	Error	BOOL	Berechnung der zu beeinflussenden Variablen "Value" wurde mit Fehler abgeschlossen; die Ausgangsvariab-le "ErrorIdent" ist gültig
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detailbeschreibung der Diagnose im Fehlerfall
	PresetAck	BOOL	Der Wert des Eingangs "PresetValue" wurde über-nommen

ML_TechBase.library

E/A-Typ	Name	Datentyp	Beschreibung
	Changing	BOOL	Berechnung der zu beeinflussenden Variablen erfolgt. Ausgangsvariable "Value" ist gültig
	HighLimitActive	BOOL	Maximaler Ausgangswert der zu beeinflussenden Variablen wurde erreicht
	LowLimitActive	BOOL	Minimaler Ausgangswert der zu beeinflussenden Variablen wurde erreicht
VAR_IN_OUT	Value	REAL	Die zu beeinflussende Variable

Abb.5-8: Schnittstellenvariablen Funktionsbaustein MB_ContinuousAdjust-Type01

Signal-Zeit-Diagramm

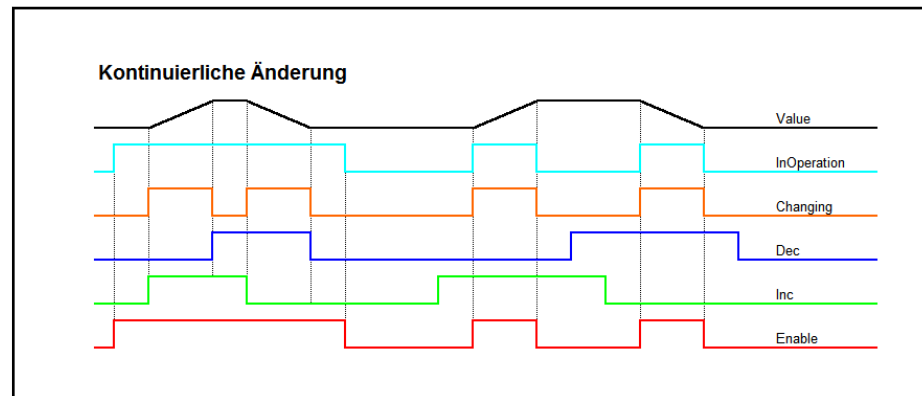


Abb.5-9: Kontinuierliche Parametereinstellung

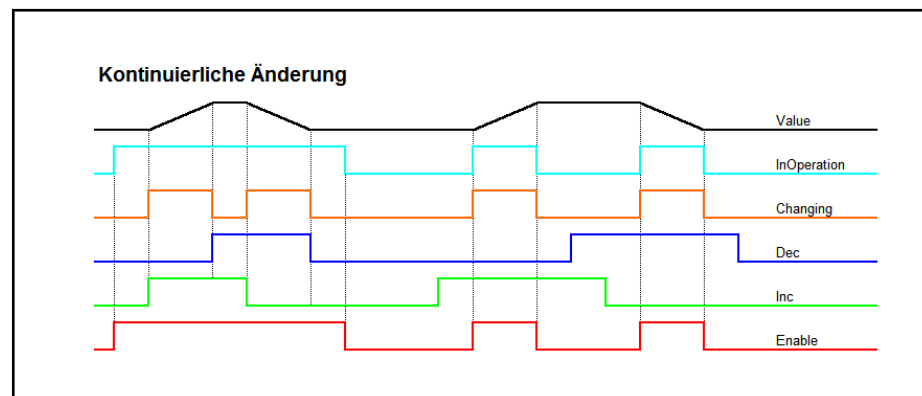


Abb.5-10: Kontinuierliche Parametereinstellung

Funktionsbeschreibung

Die Verstellung der beeinflussten Variablen "Value" kann sowohl in positiver als auch in negativer Richtung erfolgen. Durch die Grenzwerte wird ein Gültigkeitsbereich für die Beeinflussung der Variablen festgelegt. Die Vorgabe eines Modulo-Wertes erlaubt die Verstellung um innerhalb des Bezugssystems der Achse zu sein. Durch Festlegen der Schrittweite und der Anzahl der Schritte pro Sekunde wird die Änderungsgeschwindigkeit der beeinflussten Variablen bestimmt. Zudem besteht die Möglichkeit, einen vorgegebenen Wert ("PresetValue"), der sich im zulässigen Betriebsbereich befindet, direkt auf die beeinflusste Variable zu schreiben.

Nach der Bearbeitungsfreigabe über "Enable" kann die zu beeinflussende Variable über die Eingänge "Inc" (in positiver Richtung) und "Dec" (in negativer Richtung) verändert werden. Solange der Eingang "Inc" bzw. "Dec" gesetzt ist, wird die Größe der einstellbaren Variablen mit der vorgegebenen Geschwindigkeit kontinuierlich erhöht bzw. verringert. ("StepsPerSecond" * "StepWidth").

Bei der Initialisierung des Funktionsbausteines muss sichergestellt werden, dass die Eingangsgrößen "ModuloValue", "HighLimit" und "LowLimit" gültige Werte besitzen.

Durch die Festlegung von Wichtungsfaktoren ("AdjWeightFactors[]") und dazugehörigen Zeitintervallen ("AdjTimeIntervals[]") lässt sich die Änderungsgeschwindigkeit zeitabhängig verändern. Die Wichtungsfaktoren und die Zeitintervalle müssen zusätzlich zu den Eingängen "StepsPerSecond" und "StepWidth" gesetzt werden. Die Änderungsgeschwindigkeit wird dann, abhängig von dem aktuellen Zeitintervall, mit dem entsprechenden Wichtungsfaktor multipliziert. Die nachfolgende Abbildung stellt das Verhalten des Funktionsbausteins anhand einer Beispielkonfiguration dar: Die Eingangsgrößen von "AdjTimeIntervals" (T1; T2; T3; T4) werden in Sekunden angegeben. T5 bezeichnet dabei den Zeitraum nach Abschluss des Zeitintervalls T4 bis zum Abbruch des Verstellvorganges. Die Eingangsgrößen von "AdjWeightFactors" ([0,5; 1; 1,5; 2; 4]) bestimmen die Wichtung der Änderungsgeschwindigkeit. Der letzte Wert von "AdjWeightFactors" ("AdjWeightFactors [5]") wird dem Zeitintervall T5 zugeordnet.

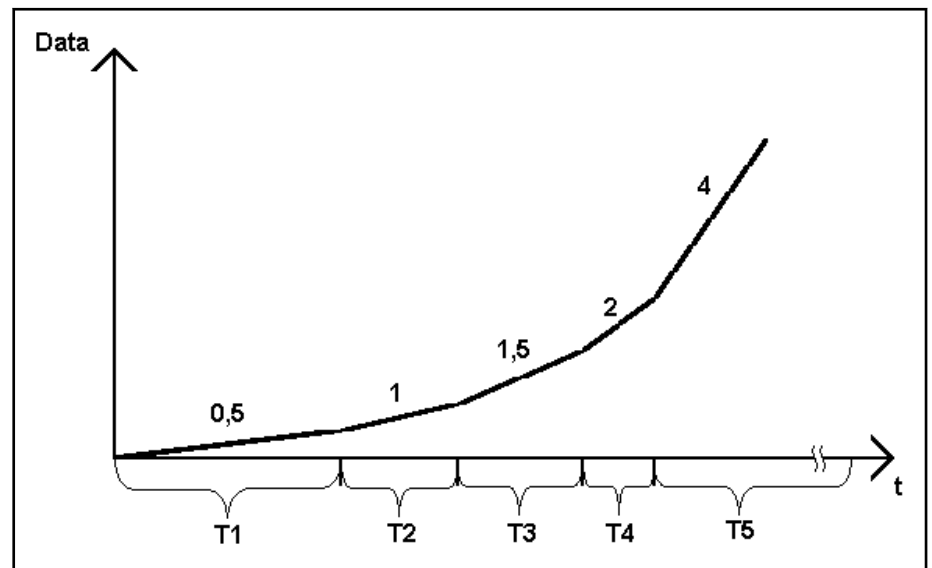


Abb.5-11: Gewichtete Verstellung



Wird ein Zeitintervall von Null gewählt, wird der darauffolgende Wichtungsfaktor bis zum Abbruch des Verstellvorganges beibehalten.

Beispiel:

Ist "AdjTimeIntervals[2]" = 0, wird der Wichtungsfaktor von "AdjWeightFactors[3]" bis zum Abbruch beibehalten.

Die Berechnung der abgelaufenen Zeit zur Bestimmung des korrekten Zeitintervalls der Wichtungsfaktoren wird mit jedem neu beginnenden Verstellvorgang rückgesetzt. Sind die Eingänge "AdjWeightFactors" und "AdjTimeIntervals" nicht gesetzt, wird eine kontinuierliche Verstellung mit der durch die Eingänge "StepsPerSecond" und "StepWidth" vorgegebenen Änderungsgeschwindigkeit durchgeführt.

ML_TechBase.library



Für die Eingangswerte der Zeitintervalle ("AdjTimeIntervals") müssen positive Werte angegeben werden. Die Wichtungsfaktoren können auch negative Werte erhalten.

Die zu beeinflussende Variable kann nur auf den vorgegebenen Voreinstellwert gesetzt werden, wenn sich dieser innerhalb des gültigen, durch "HighLimit" und "LowLimit" vorgegebenen Bereichs befindet.

Der Ausgang "InOperation" signalisiert, dass der Funktionsbaustein in Betrieb ist. Eine Verstellung findet jedoch nicht statt. Der Ausgang "Changing" zeigt an, dass der Verstellvorgang fehlerfrei abgearbeitet wurde. Der Wert der verstellten Variablen "Value" ist gültig.

Der Ausgang "Error" zeigt das Auftreten eines Fehlers während des Verstellvorgangs an. Detailangaben zu dem Fehler sind der Struktur "ErrorIdent" zu entnehmen.

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle „F_RELATED_TABLE“, 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Eingänge liegen außerhalb des zulässigen Bereichs
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Antriebs-Firmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins

Abb.5-12: Fehlercodes Funktionsbaustein MB_ContinuousAdjustType01

5.2.3 MB_ContinuousAdjustType02

Kurzbeschreibung Der Funktionsbaustein MB_ContinuousAdjustType02 ermöglicht die kontinuierliche Verstellung einer zu beeinflussenden DINT-Variablen über binäre Eingänge.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD ab MPx18 (in Vorbereitung)	in Vorbereitung

Abb.5-13: Referenztable Funktionsbaustein MB_ContinuousAdjustType02

Schnittstellenbeschreibung

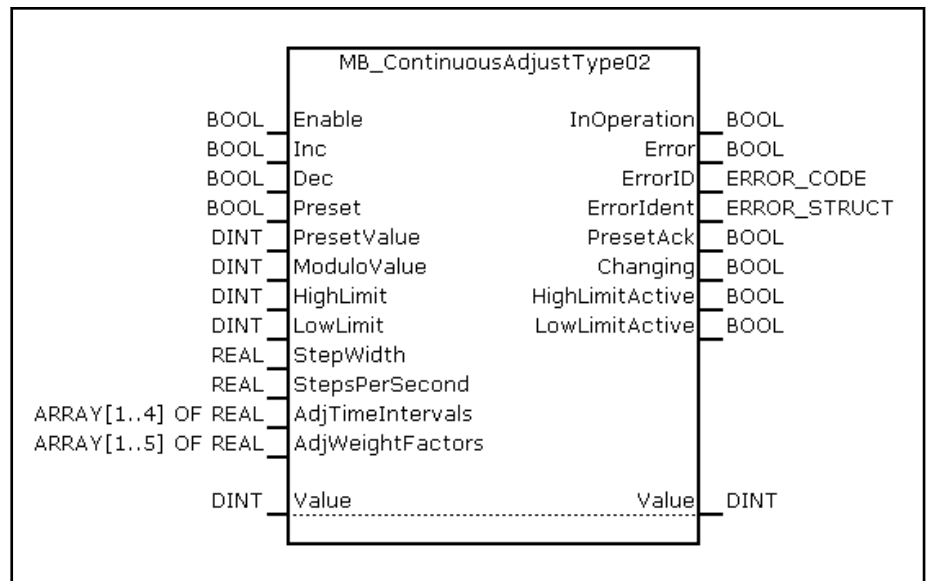


Abb.5-14: Funktionsbaustein MB_ContinuousAdjustType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Freigabe des Funktionsbausteins (zyklisch, zustands-gesteuert)
	Inc	BOOL	Beeinflusste Variable erhöhen
	Dec	BOOL	Beeinflusste Variable verringern
	Preset	BOOL	Voreinstellwert (PresetValue) setzen
	PresetValue	DINT	Vorgabewert für den Voreinstellwert
	ModuloValue	DINT	Modulowert (wenn 0, dann absolute Verarbeitung)
	HighLimit	DINT	Maximaler Ausgangswert der zu beeinflussenden Va-riablen
	LowLimit	DINT	Minimaler Ausgangswert der zu beeinflussenden Va-riablen
	StepWidth	REAL	Schrittweite
	StepsPerSecond	REAL	Schritte pro Sekunde
	AdjTimeIntervals	ARRAY [1..4] OF REAL	Zeitintervalle in Sekunden zur Wichtung der Ände-rungsgeschwindigkeit
	AdjWeightFactors	ARRAY [1.0,5] OF REAL	Wichtungsfaktoren der Änderungsgeschwindigkeit
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein aktiv. Die Ausgänge HighLimitActi-ve und LowLimitActive sind gültig
	Error	BOOL	Berechnung der zu beeinflussenden Variablen "Value" wurde mit Fehler abgeschlossen; die Ausgangsvariab-le „ErrorIdent“ ist gültig
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detailbeschreibung der Diagnose im Fehlerfall
	PresetAck	BOOL	Der Wert des Eingangs "PresetValue" wurde über-nommen

ML_TechBase.library

E/A-Typ	Name	Datentyp	Beschreibung
	Changing	BOOL	Berechnung der zu beeinflussenden Variablen erfolgt. Die Ausgangsvariable "Value" ist gültig
	HighLimitActive	BOOL	Maximaler Ausgangswert der zu beeinflussenden Variablen wurde erreicht
	LowLimitActive	BOOL	Minimaler Ausgangswert der zu beeinflussenden Variablen wurde erreicht
VAR_IN_OUT	Value	DINT	Die zu beeinflussende Variable

Abb.5-15: Schnittstellenvariablen Funktionsbaustein MB_ContinuousAdjust-Type02

Signal-Zeit-Diagramm

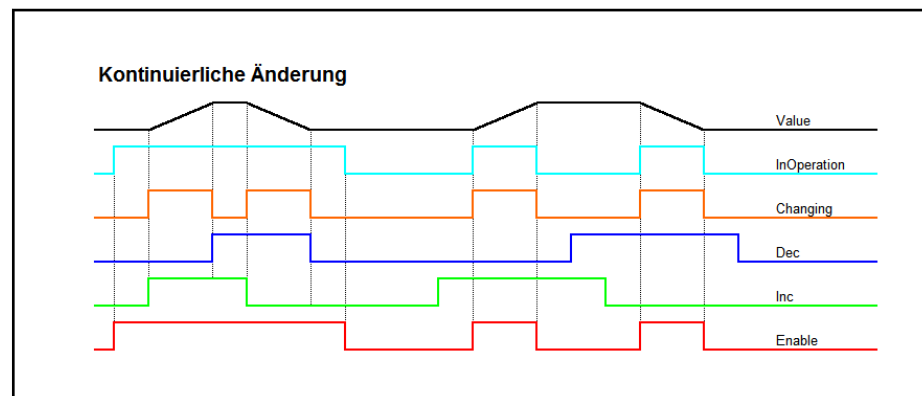


Abb.5-16: Kontinuierliche Parametereinstellung

Funktionsbeschreibung

Die Verstellung der beeinflussten Variablen "Value" kann sowohl in positiver als auch in negativer Richtung erfolgen. Durch die Grenzwerte wird ein Gültigkeitsbereich für die Beeinflussung der Variablen festgelegt. Die Vorgabe eines Modulo-Wertes erlaubt die Verstellung um innerhalb des Bezugssystems der Achse zu sein. Durch Festlegen der Schrittweite und der Anzahl der Schritte pro Sekunde wird die Änderungsgeschwindigkeit der beeinflussten Variablen bestimmt. Zudem besteht die Möglichkeit, einen vorgegebenen Wert (PresetValue), der sich im zulässigen Betriebsbereich befindet, direkt auf die beeinflusste Variable zu schreiben.

Nach der Bearbeitungs freigabe über "Enable" kann die zu beeinflussende Variable über die Eingänge "Inc" (in positiver Richtung) und "Dec" (in negativer Richtung) verändert werden. Solange der Eingang "Inc" bzw. "Dec" gesetzt ist, wird die Größe der einstellbaren Variablen mit der vorgegebenen Geschwindigkeit kontinuierlich erhöht bzw. verringert. ("StepsPerSecond" * "StepWidth").

Bei der Initialisierung des Funktionsbausteines muss sichergestellt werden, dass die Eingangsgrößen "ModuloValue", "HighLimit" und "LowLimit" gültige Werte besitzen.

Durch die Festlegung von Wichtungsfaktoren ("AdjWeightFactors[]") und dazugehörigen Zeitintervallen ("AdjTimeIntervals[]") lässt sich die Änderungsgeschwindigkeit zeitabhängig verändern. Die Wichtungsfaktoren und die Zeitintervalle müssen zusätzlich zu den Eingängen "StepsPerSecond" und "StepWidth" gesetzt werden. Die Änderungsgeschwindigkeit wird dann, abhängig von dem aktuellen Zeitintervall, mit dem entsprechenden Wichtungsfaktor multipliziert. Die nachstehende Abbildung stellt das Verhalten des Funktionsbausteines anhand einer Beispielkonfiguration dar: Die Eingangsgrößen von "AdjTimeIntervals" (T1; T2; T3; T4) werden in Sekunden angegeben. T5 bezeichnet dabei den Zeitraum nach Abschluss des Zeitintervalls T4 bis zum Abbruch des Verstellvorganges. Die Eingangsgrößen von "AdjWeightFac-

tors" ([0,5; 1; 1,5; 2; 4]) bestimmen die Wichtung der Änderungsgeschwindigkeit. Der letzte Wert von "AdjWeightFactors" ("AdjWeightFactors [5]") wird dem Zeitintervall T5 zugeordnet.

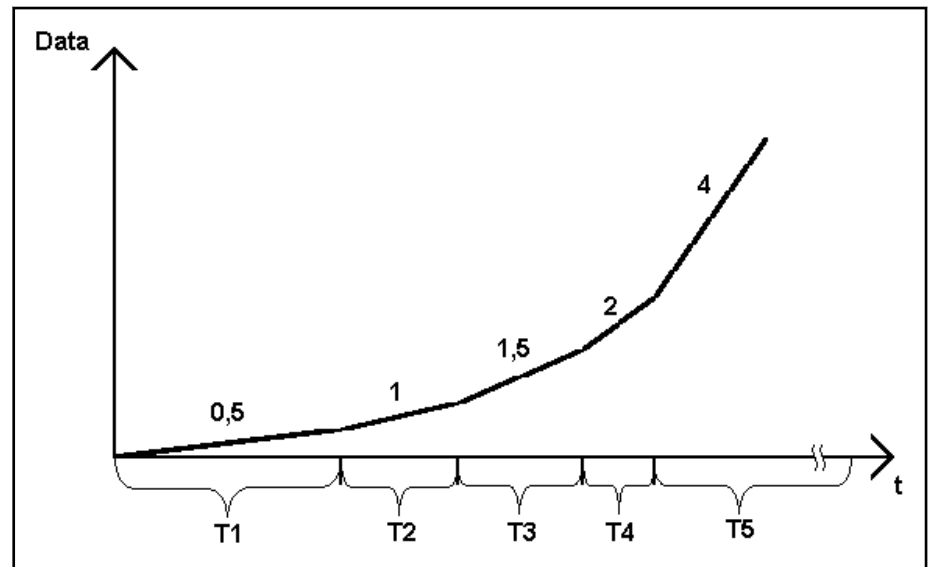


Abb.5-17: Gewichtete Verstellung



Wird ein Zeitintervall von Null gewählt, wird der darauffolgende Wichtungsfaktor bis zum Abbruch des Verstellvorganges beibehalten.

Beispiel:

Ist "AdjTimeIntervals[2]" = 0, wird der Wichtungsfaktor von "AdjWeightFactors[3]" bis zum Abbruch beibehalten.

Die Berechnung der abgelaufenen Zeit zur Bestimmung des korrekten Zeitintervalls der Wichtungsfaktoren wird mit jedem neu beginnenden Verstellvorgang rückgesetzt. Sind die Eingänge "AdjWeightFactors" und "AdjTimeIntervals" nicht gesetzt, wird eine kontinuierliche Verstellung mit der durch die Eingänge "StepsPerSecond" und "StepWidth" vorgegebenen Änderungsgeschwindigkeit durchgeführt.



Für die Eingangswerte der Zeitintervalle (AdjTimeIntervals) müssen positive Werte angegeben werden. Die Wichtungsfaktoren können auch negative Werte erhalten.

Die zu beeinflussende Variable kann nur auf den vorgegebenen Voreinstellwert gesetzt werden, wenn sich dieser innerhalb des gültigen, durch "HighLimit" und "LowLimit" vorgegebenen Bereichs befindet.

Der Ausgang "InOperation" signalisiert, dass der Funktionsbaustein in Betrieb ist. Eine Verstellung findet jedoch nicht statt. Der Ausgang "Changing" zeigt an, dass der Verstellvorgang fehlerfrei abgearbeitet wurde. Der Wert der verstellten Variablen "Value" ist gültig.

Der Ausgang "Error" zeigt das Auftreten eines Fehlers während des Verstellvorgangs an. Detailangaben zu dem Fehler sind der Struktur "ErrorIdent" zu entnehmen.

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ML_TechBase.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Eingänge liegen außerhalb des zulässigen Bereichs
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Antriebs-Firmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins

Abb.5-18: Fehlercodes Funktionsbaustein MB_ContinuousAdjustType02

5.2.4 MB_IncrementalAdjustType01

Kurzbeschreibung

Der Funktionsbaustein MB_IncrementalAdjustType01 ermöglicht die inkrementelle Verstellung einer REAL-Variablen über binäre Eingänge.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD ab MPx18 (in Vorbereitung)	in Vorbereitung

Abb.5-19: Referenztabelle Funktionsbaustein MB_IncrementalAdjustType01

Schnittstellenbeschreibung

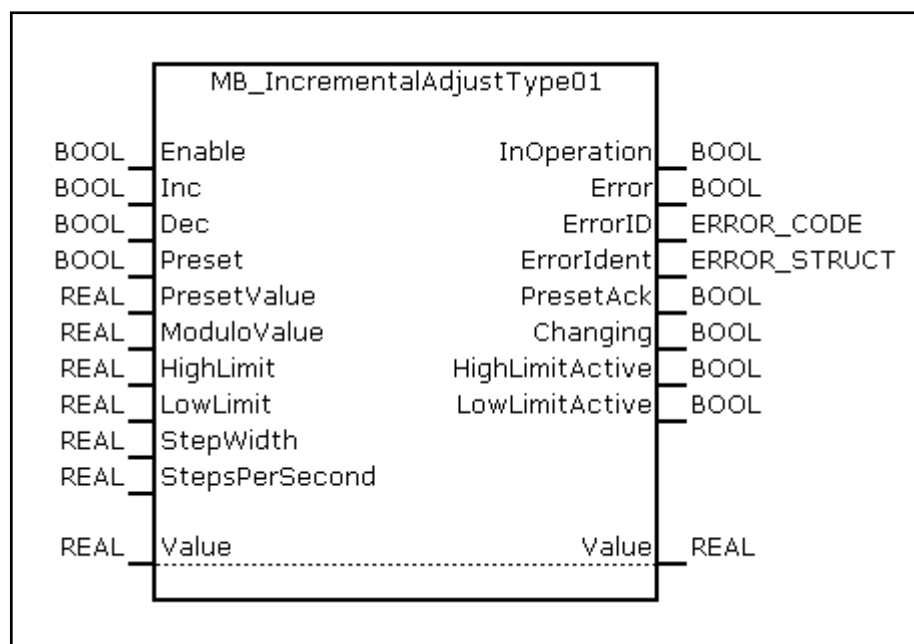


Abb.5-20: Funktionsbaustein MB_IncrementalAdjustType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Freigabe des Funktionsbausteins (zyklisch, zustands-gesteuert)
	Inc	BOOL	Beeinflusste Variable erhöhen
	Dec	BOOL	Beeinflusste Variable verringern
	Preset	BOOL	Voreinstellwert (PresetValue) setzen
	PresetValue	REAL	Vorgabewert für den Voreinstellwert
	ModuloValue	REAL	Modulowert zur rotatorischen Berechnung

E/A-Typ	Name	Datentyp	Beschreibung
	HighLimit	REAL	Maximaler Ausgangswert der zu beeinflussenden Variablen
	LowLimit	REAL	Minimaler Ausgangswert der zu beeinflussenden Variablen
	StepWidth	REAL	Schrittweite
	StepsPerSecond	REAL	Schritte pro Sekunde
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein aktiv. Die Ausgänge "HighLimitActive" und "LowLimitActive" sind gültig
	Error	BOOL	Berechnung der zu beeinflussenden Variablen "Value" wurde mit Fehler abgeschlossen; die Ausgangsvariable "ErrorIdent" ist gültig
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detailbeschreibung der Diagnose im Fehlerfall
	PresetAck	BOOL	Der Wert des Eingangs "PresetValue" wurde übernommen
	Changing	BOOL	Berechnung der zu beeinflussenden Variablen erfolgt. Ausgangsvariable "Value" ist gültig
	HighLimitActive	BOOL	Maximaler Ausgangswert der zu beeinflussenden Variablen wurde erreicht
	LowLimitActive	BOOL	Minimaler Ausgangswert der zu beeinflussenden Variablen wurde erreicht
VAR_IN_OUT	Value	REAL	Die zu beeinflussende Variable

Abb.5-21: Schnittstellenvariablen Funktionsbaustein MB_IncrementalAdjust-Type01

Signal-Zeit-Diagramm

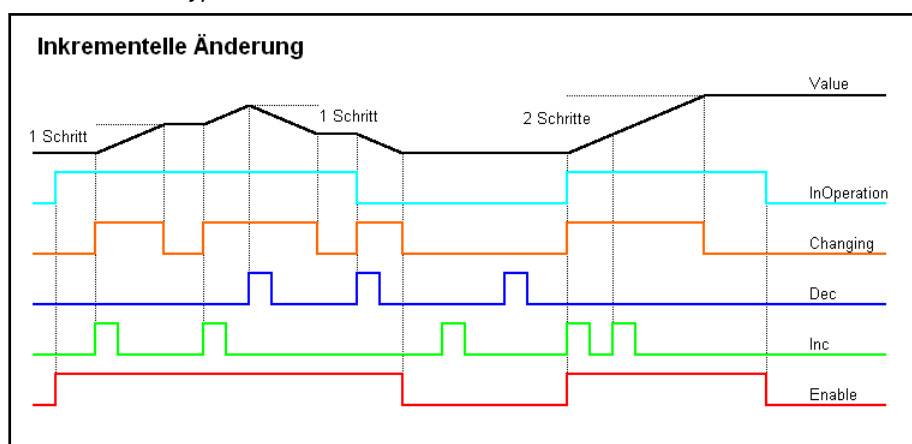


Abb.5-22: Inkrementelle Parameterverstellung

Funktionsbeschreibung

Die Verstellung der beeinflussten Variablen "Value" kann sowohl in positiver als auch in negativer Richtung erfolgen. Durch die Grenzwerte wird ein Gültigkeitsbereich für die Beeinflussung der Variablen festgelegt. Die Vorgabe eines Modulo-Wertes erlaubt die Verstellung um innerhalb des Bezugssystems der Achse zu sein. Durch Festlegen der Schrittweite und der Anzahl der Schritte pro Sekunde wird die Änderungsgeschwindigkeit der beeinflussten Variablen bestimmt. Zudem besteht die Möglichkeit, einen vorgegeben Wert

ML_TechBase.library

("PresetValue"), der sich im zulässigen Betriebsbereich befindet, direkt auf die beeinflusste Variable zu schreiben.

Nach der Bearbeitungsfreigabe über "Enable" kann die zu beeinflussende Variable über die Eingänge "Inc" (in positiver Richtung) und "Dec" (in negativer Richtung) verändert werden. Solange der Eingang "Inc" bzw. "Dec" gesetzt ist, wird die Größe der einstellbaren Variablen mit der vorgegebenen Geschwindigkeit kontinuierlich erhöht bzw. verringert. ("StepsPerSecond" * "StepWidth").



Die zu beeinflussende Variable kann nur auf den vorgegebenen Voreinstellwert gesetzt werden, wenn sich dieser innerhalb des gültigen, durch "HighLimit" und "LowLimit" vorgegebenen Bereichs befindet.

Der Ausgang "InOperation" signalisiert, dass der Funktionsbaustein in Betrieb ist. Eine Verstellung findet jedoch nicht statt. Der Ausgang "Changing" zeigt an, dass der Verstellvorgang fehlerfrei abgearbeitet wurde. Der Wert der verstellten Variablen "Value" ist gültig.

Ein während der Bearbeitung des Funktionsbausteins auftretender Fehler wird durch den Ausgang "Error" = TRUE signalisiert. Die Detailangaben zu dem Fehler sind der Ausgangsstruktur "ErrorIdent" zu entnehmen.

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Eingänge liegen außerhalb des zulässigen Bereichs
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Antriebs-Firmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins

Abb.5-23: Fehlercodes Funktionsbaustein MB_IncrementalAdjustType01

5.2.5 Parametrierung der Funktionsbausteine zum Vertippen von Variablen

Erforderliche Hardware

Die folgenden Rexroth-Hardware-Komponenten werden benötigt:

- IndraMotion MLC/IndraLogic XLC Hardware

Erforderliche Firmware

Mit den oben genannten Rexroth-Hardware-Komponenten ist folgende Steuerungs-Firmware zu verwenden:

- IndraMotion MLC: FWA-CMLXX*-MLC-12VRS
- IndraLogic XLC: FWA-CMLXX*-XLC-12VRS

XX - Gerätevariante, z.B. CML65

Erforderliche Software

Folgende PC-Software ist zu verwenden:

- IndraWorks IndraMotion MLC 12VRS
- IndraWorks IndraLogic XLC 12VRS

Parametrierung der Funktionsbausteine zum Vertippen von Variablen

Es ist keine besondere Parametrierung erforderlich.

5.3 Funktionsbaustein für Zweipunktregler

5.3.1 Einführung

Der Zweipunktregler besitzt eine Hysterese und ist für einfache Regelungen geeignet.

5.3.2 ML_TwoPosCtrlType01

Kurzbeschreibung

Der Funktionsbaustein ML_TwoPosCtrlType01 realisiert einen Zweipunktregler mit Hysterese.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-24: Referenztablette Funktionsbaustein ML_TwoPosCtrlType01

Schnittstellenbeschreibung

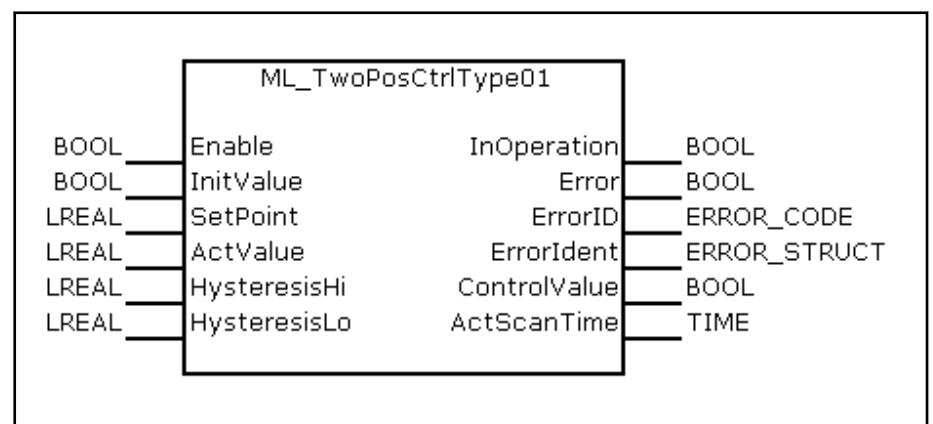


Abb.5-25: Funktionsbaustein ML_TwoPosCtrlType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (zyklisch, zustandsgesteuert)
	InitValue	BOOL	Vorgabewert für Stellgröße y
	SetPoint	LREAL	Führungsgröße w (Sollwert)
	ActValue	LREAL	Regelgröße x (Istwert)
	HysteresisHi	LREAL	Obere Ansprechschwelle der Hysterese
	HysteresisLo	LREAL	Untere Ansprechschwelle der Hysterese
VAR_OUTPUT	InOperation	BOOL	Berechnung der Stellgröße y ohne Fehler abgeschlossen, Ausgangsvariable "ControlValue" gültig
	Error	BOOL	Berechnung der Stellgröße "ControlValue" mit Fehler abgeschlossen, Ausgangsvariable "ErrorIdent" gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Beschreibung der Diagnose im Fehlerfall
	ControlValue	BOOL	Reglerausgabewert – Stellgröße y
	ActScanTime	TIME	Reglerzykluszeit [ms]

Abb.5-26: Schnittstellenvariablen Funktionsbaustein ML_TwoPosCtrlType01

Spezifikation

Der Funktionsbaustein ML_TwoPosCtrlType01 kann über folgende Eingangsvariablen in seinen Eigenschaften angepasst werden:

ML_TechBase.library

"HysteresisHi":

- Der Eingang muss mit einem positiven Wert bzw. Null vorbelegt werden, der Wert Null schaltet die positive Hysterese aus.

"HysteresisLo":

- Der Eingang muss mit einem negativen Wert bzw. Null vorbelegt werden, der Wert Null schaltet die negative Hysterese aus.



Bei der Berechnung der Stellgröße "ControlValue" ist "FALSE" dominant, d.h. bei "HysteresisLo" == "HysteresisHi" == Null ist bei einer Regeldifferenz von Null ("SetPoint" == "ActualValue") der Ausgang "ControlValue" stets "FALSE".

Funktionsbeschreibung

Der Funktionsbaustein ML_TwoPosCtrlType01 belegt nach der Bearbeitungsfreigabe des Funktionsbausteins mittels "Enable" im ersten Reglerzyklus den internen Reglerzustand sowie den Reglerausgang auf den am Eingang "InitValue" anliegenden Presetwert vor und berechnet in den Folgezyklen zyklisch die Stellgröße "ControlValue".

Im Fall der fehlerfreien Bearbeitung des Funktionsbausteins wird die Gültigkeit der Stellgröße durch den Ausgang "InOperation" signalisiert, die Elemente der Ausgangsstruktur "ErrorID" und "ErrorIdent" werden in diesem Fall nicht aktualisiert.

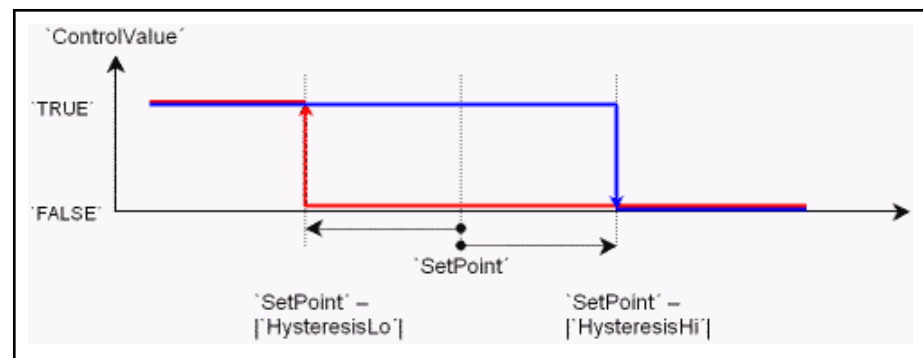


Abb.5-27: Hysterese Funktionsbaustein ML_TwoPosCtrlType01

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert, sowie durch die in diesem Fall aktualisierten Elemente der Ausgangsstruktur "ErrorID" und "ErrorIdent" spezifiziert. Die Ausgänge "ControlValue" und "ActScanTime" werden im Fehlerfall nicht aktualisiert.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **MLC_TABLE**, 16#0030. Er kann in Additional1 und Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#F0260030	0	Eingänge "HysteresisHi" oder "HysteresisLo" sind ungültig
STATE_MACHINE_ERROR (16#0005)	16#F0260031	xx	Ungültiger Zustand der Zustandsmaschine xx gibt den ungültigen Zustand an

Abb.5-28: Fehlercodes Funktionsbaustein ML_TwoPosCtrlType01

5.4 Funktionsbausteine für mathematische Funktionen

5.4.1 Einführung

Die nachfolgenden Bausteine stellen mathematische Funktionen zur Verfügung. Der Baustein [ML_InterpolationLinear](#) Seite, 51 dient zur Berechnung von Zwischenpunkten und kann z.B. für Linearisierungen verwendet werden.

Der Baustein [ML_MaxValue](#), Seite, 53 gibt an seinem Ausgang mit jedem Aufruf den Maximalwert aller bisher aufgetreten Wert an.

5.4.2 ML_InterpolationLinear

Kurzbeschreibung

Der Funktionsbaustein ML_InterpolationLinear berechnet auf der Grundlage einer definierten X-Y-Stützstellentabelle zu einem gegebenen X-Wert durch lineare Interpolation zyklisch den zugehörigen Y-Wert.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-29: Referenztable Funktionbaustein ML_InterpolationLinear

Schnittstellenbeschreibung

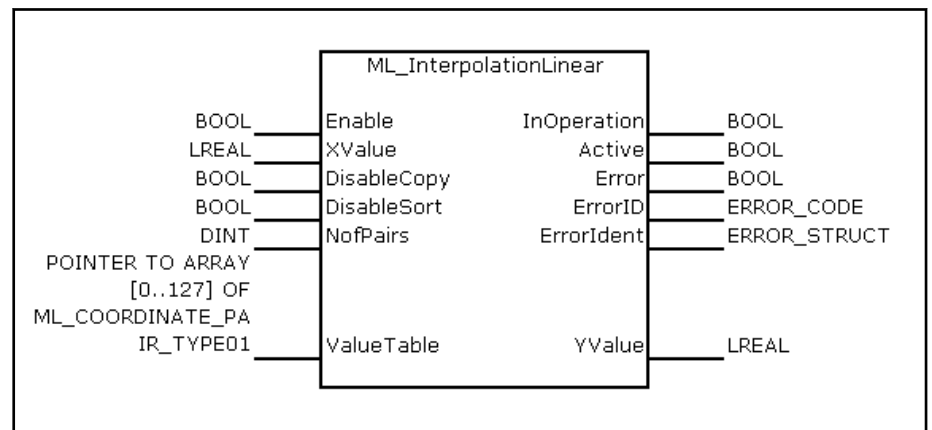


Abb.5-30: Funktionsbaustein ML_InterpolationLinear

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Freigabe des Funktionsbausteins (zyklisch, zustands-gesteuert)
	XValue	LREAL	X-Wert
	DisableCopy	BOOL	Deaktivierung des Kopierens der Stützstellentabelle
	DisableSort	BOOL	Deaktivierung des Sortierens der Stützstellentabelle
	NofPairs	DINT	Anzahl der zusammengehörigen Wertepaare in der Stützstellentabelle
	ValueTable	POINTER TO ARRAY [0..127] OF ML_COORDINATE_PAIR_TYPE01	Adresse der Stützstellentabelle
VAR_OUTPUT	InOperation	BOOL	Zyklische Berechnung eines Y-Wertes ohne Fehler abgeschlossen, Ausgangsvariable "YValue" gültig
	Active	BOOL	Berechnung eines Y-Wertes aktiv, Ausgangsvariablen ungültig

ML_TechBase.library

E/A-Typ	Name	Datentyp	Beschreibung
	Error	BOOL	Berechnung eines Y-Wertes mit Fehler abgeschlossen, Fehlerausgänge "ErrorID" und "ErrorIdent" gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang detaillierte Informationen über den Fehler
	YValue	LREAL	Y-Wert

Abb.5-31: Schnittstellenvariablen Funktionsbaustein ML_InterpolationLinear

Funktionsbeschreibung

Der Funktionsbaustein ML_InterpolationLinear berechnet nach der Bearbeitungsfreigabe des Funktionsbausteins mittels "Enable" zyklisch zum Eingangswert "XValue" den zugehörigen Y-Wert "YValue" auf Basis einer vorgegebenen Stützpunkttable.

Die Stützpunkttable wird abhängig von den Eingängen "DisableCopy" und "DisableSort" bei steigender Flanke auf ein internes Datenarray gesichert und anschließend aufsteigend sortiert. Die Zeit für die Berechnung eines zugehörigen Y-Wertes "YValue" kann minimiert werden, wenn bei Bearbeitungsfreigabe des Bausteins eine bereits aufsteigend sortierte Stützpunkttable übergeben und dies durch den Eingang "DisableSort" gekennzeichnet wird. In diesem Falle wird das Datenarray nicht sortiert. Soll bei einer erneuten Abarbeitung des Bausteins die intern vorhandene und bereits verwendete Stützpunkttable erneut angewendet werden, so muss sowohl der Eingang "DisableCopy" als auch der Eingang "DisableSort" auf aktiviert gesetzt werden.

Der zu berechnende Y-Wert wird im Falle der fehlerfreien Bearbeitung des Funktionsbausteins auf dem Ausgang "YValue" abgelegt, wobei die Gültigkeit des Wertes durch den Ausgang "InOperation" signalisiert wird.

Kann der zu berechnende Y-Wert nach Aktivierung des Funktionsbausteins nicht unmittelbar zur Verfügung gestellt werden, wird dies durch den Ausgang "Active" angezeigt.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Der Ausgang "YValue" wird im Fehlerfall nicht aktualisiert.

Einfluss der Eingänge "DisableCopy" und "DisableSort" auf das Laufzeitverhalten des FB

DisableCopy	DisableSort	Anwendungsfall
FALSE	FALSE	Eine unsortierte Stützpunkttable wird vorgegeben, die in dieser Art in der vorherigen Berechnung noch nicht verwendet worden ist. Die Stützpunkttable wird kopiert und sortiert
FALSE	TRUE	Eine bereits aufsteigend sortierte Stützpunkttable wird vorgegeben, die in dieser Art in der vorherigen Berechnung noch nicht verwendet worden ist. Die Stützpunkttable wird kopiert, aber intern nicht sortiert

DisableCopy	DisableSort	Anwendungsfall
TRUE	FALSE	Diese Kombination bildet keinen sinnvollen Zustand
TRUE	TRUE	Die gleiche Stützstellentabelle wurde in vorherigen Berechnungen bereits verwendet. Sie ist daher intern bereits bekannt und aufsteigend sortiert

Abb.5-32: Belegung der Eingänge DisableCopy und DisableSort



Die Sortierung der Stützstellentabelle nach Aktivierung des Funktionsbausteins bedingt unter Umständen, dass der Funktionsbaustein auf Grund der Einschränkung auf eine max. zulässigen Laufzeit den gewünschten Ausgangswert nicht im ersten bzw. Folgezyklus liefern kann. Diese Zeitdauer zu Beginn der Aktivierung wird durch "Active" signalisiert.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **MLC_TABLE**, 16#0030. Er kann in Additional1 und Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#F0260030	0	Eingänge ungültig (XValue außerhalb der Stützpunkte)
STATE_MACHINE_ERROR (16#0005)	16#F0260031	xx	ungültiger Zustand der Zustandsmaschine. xx gibt den ungültigen Zustand an

Abb.5-33: Fehlercodes Funktionsbaustein ML_InterpolationLinear

5.4.3 ML_MaxValue

Kurzbeschreibung

Der Funktionsbaustein ML_MaxValue ermittelt aus dem zeitlichen Verlauf des Eingangssignals dessen bisherigen Maximalwert.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-34: Referenztablette Funktionsbaustein ML_MaxValue

Schnittstellenbeschreibung

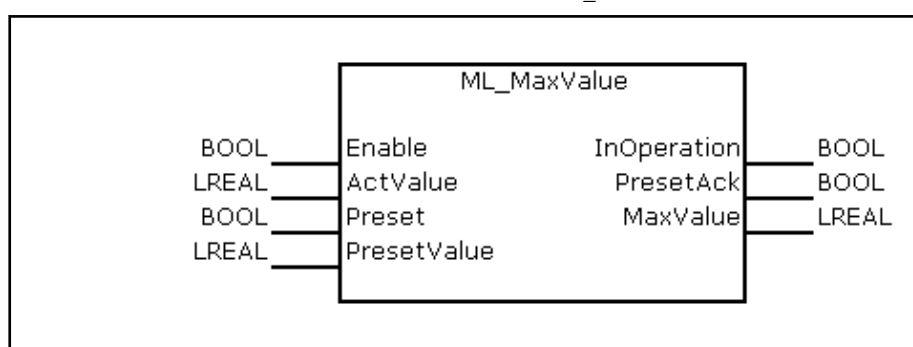


Abb.5-35: Funktionsbaustein ML_MaxValue

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (zyklisch, zustandsgesteuert)
	ActValue	LREAL	aktueller Wert

ML_TechBase.library

E/A-Typ	Name	Datentyp	Beschreibung
	Preset	BOOL	Aktivierung des Vorgabewertes
		LREAL	Vorgabewert
VAR_OUTPUT	InOperation	BOOL	Ermittlung des Maximalwertes abgeschlossen
	PresetAck	BOOL	Bestätigung für die Übernahme des "PresetValue"-Wertes
	MaxValue	LREAL	maximaler Wert

Abb.5-36: Schnittstellenvariablen Funktionsbaustein ML_MaxValue

Funktionsbeschreibung

Der Funktionsbaustein ML_MaxValue vergleicht nach der Freigabe des Funktionsbausteins mittels "Enable" den aktuellen Wert der Eingangsvariable "ActValue" mit dem Wert in "MaxValue". Er kopiert den jeweils größeren Wert auf die Ausgangsvariable "MaxValue".

Wird zur Laufzeit des Funktionsbausteins der Eingang "Preset" aktiviert, wird der Wert der Eingangsvariablen "PresetValue" auf die Ausgangsvariable "MaxValue" kopiert. Der Eingang ist damit zustandsgesteuert, d.h. solange der "Preset" Eingang aktiv ist, wird der "PresetValue" auf die Ausgangsvariable "MaxValue" kopiert.



Soll mit dem Aktivieren des Bausteins auch gleichzeitig ein "PresetValue"-Wert vorgegeben werden, muss der Eingang "Preset" gleichzeitig mit dem Eingang "Enable" gesetzt werden.

Fehlercodes

Der Funktionsbaustein ML_MaxValue generiert keinen Fehler.

5.5 Funktionsbaustein zur Anwendung "Nockenschaltwerk"

5.5.1 Einführung und Übersicht

Der Funktionsbaustein zur Anwendung "Nockenschaltwerk" [Kap. 5.5.5 "MC_DigitalCamSwitch" auf Seite 56](#) ist ein PLCopen-konformer Baustein. Damit können bis zu 8 Ausgänge mit bis zu 32 Nocken frei programmiert werden.

Die dabei verwendeten Datenstrukturen

- [Kap. 5.5.4 "MC_OUTPUT_REF" auf Seite 55](#)
- [Kap. 5.5.2 "MC_CAMSWITCH_REF mit ML_BASIC_CAMSWITCH_REF" auf Seite 54](#)
- [Kap. 5.5.3 "MC_TRACK_REF mit ML_BASIC_TRACK_REF" auf Seite 55](#)

entsprechen ebenso den Vorgaben der PLCopen Spezifikation.

5.5.2 MC_CAMSWITCH_REF mit ML_BASIC_CAMSWITCH_REF

Typdefinitionen

Die folgende Definition beschreibt Datentypen, die im Funktionsbaustein [Kap. 5.5.5 "MC_DigitalCamSwitch" auf Seite 56](#) benutzt werden.

Die Typ-Definitionen sind in der Bibliothek enthalten.

"MC_CAMSWITCH_REF" enthält ein Array von 32 Basis-Strukturen, die jede eine einzelne Nocke beschreibt.

Werte werden in Einheiten [u] angegeben. Diese beziehen sich auf die zugehörige Achse. Ist z. B. eine Einschaltposition bei 20u und die zugehörige Achse wird in Grad gewichtet, ist die Einschaltposition entsprechend 20°.

Die Basis-Struktur "ML_BASIC_CAMSWITCH_REF" sieht wie folgt aus:

Strukturelement	Typ	Beschreibung
TrackNumber	INT	"TrackNumber" ist die Referenz zur Nockenscheibe
FirstOnPosition [u]	REAL	Untere Grenze bei der der Ausgang AN ist (Einschaltposition)
LastOnPosition [u]	REAL	Obere Grenze bei der der Ausgang AN ist (Ausschaltposition)
AxisDirection	INT	Richtung in der das Nockenschaltwerk wirksam ist: beide (=0; Default); Positiv (1); Negativ (2)
CamSwitchMode	INT	Positionsbezogen (=0; Default); zeitbezogen (=1)
Duration	TIME	Gekoppelt mit zeitbezogenem "CamSwitchMode". D.h., der Ausgang bleibt für die angegebene Zeit an

Abb.5-37: Definition der Basiselemente der ML_BASIC_CAMSWITCH_REF-Struktur

5.5.3 MC_TRACK_REF mit ML_BASIC_TRACK_REF

Typdefinitionen Die folgende Definition beschreibt Datentypen, die im Funktionsbaustein [Kap. 5.5.5 "MC_DigitalCamSwitch" auf Seite 56](#) verwendet werden. Die Definitionen der Datentypen sind in der Bibliothek enthalten.

Eine Nockenscheibe ist ein Element, das die Eigenschaften des Nockenschaltwerkes bezogen auf einen Ausgang beschreibt.

Die "MC_TRACK_REF" enthält ein Array mit 8 Basis-Strukturen (d.h., 8 Nockenscheiben, die 8 Ausgängen zugeordnet sind). Die Basis-Struktur ("ML_BASIC_TRACK_REF") einer Nockenscheibe sieht wie folgt aus:

Strukturelement	Typ	Beschreibung
OnCompensation	TIME	Vorhaltezeit der Nockenscheibe bezogen auf den Einschaltzeitpunkt.
OffCompensation	TIME	Vorhaltezeit der Nockenscheibe bezogen auf den Ausschaltzeitpunkt.
Hysteresis [u]	REAL	Einstellbare Hysterese um mit Rauschen behaftete Eingangssignale zu filtern.
TimeDef	WORD	Zeitdefinition: Bit 0 gesetzt: "OnCompensation"-Zeit ist negativ; Bit 1 gesetzt: "OffCompensation"-Zeit ist negativ

Abb.5-38: Definition der Basiselemente der MC_TRACK_REF-Struktur

Jeder Nockenscheibe können mehrere Nocken zugeordnet sein.

5.5.4 MC_OUTPUT_REF

Typdefinitionen Die folgende Definition beschreibt Datentypen, die im Funktionsbaustein [Kap. 5.5.5 "MC_DigitalCamSwitch" auf Seite 56](#) verwendet werden. Die Definitionen der Datentypen sind in der Bibliothek enthalten.

Der Datentyp beschreibt die 8 Ausgänge des Nockenschaltwerkes.

"MC_OUTPUT" enthält ein ARRAY mit 8 Booleschen Variablen, die den Ausgängen zugeordnet sind und sieht wie folgt aus:

ML_TechBase.library

Strukturelement	Typ	Beschreibung
OUTPUT	ARRAY [0..7] OF BOOL	Ausgänge des Nockenschaltwerkes.

Abb.5-39: Definition der Basiselemente der MC_OUTPUT_REF-Struktur

5.5.5 MC_DigitalCamSwitch

Kurzbeschreibung Der Funktionsbaustein MC_DigitalCamSwitch implementiert ein frei programmierbares Nockenschaltwerk mit bis zu 32 Nocken auf 8 Ausgängen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-40: Referenztabelle Funktionsbaustein MC_DigitalCamSwitch

Schnittstellenbeschreibung

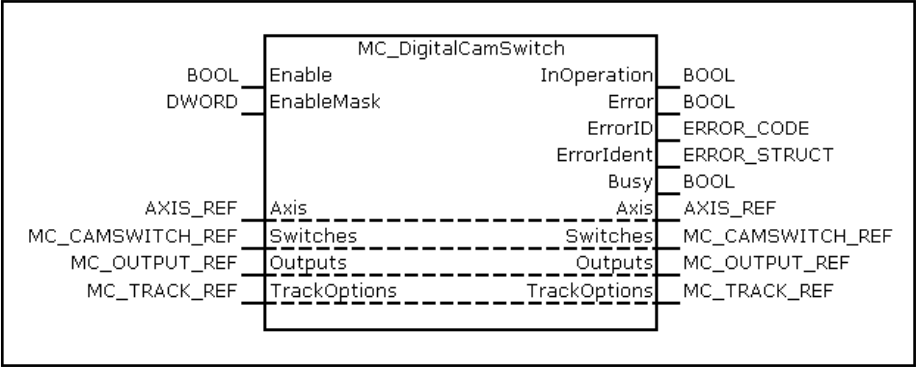


Abb.5-41: Funktionsbaustein MC_DigitalCamSwitch

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Axis	AXIS_REF	Referenz zur Achse, der das Nockenschaltwerk zugeordnet ist
	Switches	MC_CAMSWITCH_REF	Referenz zur Nockenstruktur, die alle Definitionen für alle Nocken enthält. Siehe auch Kap. 5.5.2 "MC_CAMSWITCH_REF mit ML_BASIC_CAMSWITCH_REF" auf Seite 54
	Outputs	MC_OUTPUT_REF	Referenz zu den Signalausgängen direkt bezogen auf die zugehörigen Nockenscheiben (max. 8 pro Funktionsbaustein, erster Ausgang = erste "TrackNum-ber"). Siehe auch Kap. 5.5.4 "MC_OUTPUT_REF" auf Seite 55
	TrackOptions	MC_TRACK_REF	Referenz zur Struktur, die die zugehörigen Nockenscheibeneigenschaften enthält, z.B. die Ein/Aus-Vorhaltezeit pro Ausgang/Nockenscheibe. Siehe auch Kap. 5.5.3 "MC_TRACK_REF mit ML_BASIC_TRACK_REF" auf Seite 55
VAR_INPUT	Enable	BOOL	aktiviert die Schaltausgänge

E/A-Typ	Name	Datentyp	Beschreibung
	EnableMask	DWORD	32 Bits vom Typ BOOL. Aktiviert die verschiedenen Nockenscheiben. Das niederwertigste Bit ist der niedrigsten "TrackNumber" zugeordnet. Wird das Datum auf 1, bzw. auf TRUE gesetzt, ist die zugeordnete "TrackNumber" aktiviert. Da nur die 8 niederwertigsten Bits zur Verfügung stehen, ist die Anzahl auf 8 Nockenscheiben pro Funktionsbaustein begrenzt. Ist das Bit nicht gesetzt, wird der entsprechende Ausgang auf 0 gesetzt
VAR_OUTPUT	InOperation	BOOL	Die aktivierten Nockenscheiben werden berechnet und die Ausgänge aktualisiert
	Error	BOOL	Zeigt an, dass ein Fehler innerhalb des Funktionsbausteins aufgetreten ist
	ErrorID	ERROR_CODE	Fehlerkennung
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler. Siehe auch "Fehlercodes" auf Seite 70
	Busy	BOOL	Wird gesetzt, wenn der FB aktiv ist (nicht im Idle-Mode)
Hinweis: Eine detaillierte Beschreibung der verschiedenen Ein-/Ausgänge folgt später in diesem Dokument.			

Abb.5-42: Schnittstellenvariablen Funktionsbaustein MC_DigitalCamSwitch

Funktionalität

Der folgende Funktionsbaustein MC_DigitalCamSwitch bietet eine PLCopen-spezifische (Teil 2) Schnittstelle für ein programmierbares Nockenschaltwerk. Der FB berechnet 8 binäre Nockenscheibenausgänge entsprechend der unten gezeigten Spezifikation.

Jeder binäre Ausgang ist einer Nockenscheibe zugeordnet. Eine Nockenscheibe kann mehrere Ein/Aus-Positionen entsprechend der folgenden Spezifikation enthalten. Es sind maximal 32 Nocken verfügbar die frei auf die 8 Nockenscheiben verteilt werden können.

Terminologie

Bezeichnung	Beschreibung
IndraMotion MLC/IndraLogic XLC	BRC Rack-basierende MotionLogic Control
Digital cam switch	Programmierbares Nockenschaltwerk (PLS)

Abb.5-43: Terminologie des Nockenschaltwerks

Einstellungen

Die Einheiten für Position und Geschwindigkeit müssen für die Achse, die dem Nockenschaltwerk zugeordnet sind, entsprechend eingestellt sein. Wenn z.B. die Positionen der Achse in Grad definiert sind, werden die Nockenschaltwerk-Positionen ebenfalls in Grad interpretiert.

Datentypen

Der Funktionsbaustein "MC_DigitalCamSwitch" nutzt folgende Datentypen:

Eine Nockenscheibe ist ein Element, das die Eigenschaften des Nockenschaltwerkes bezogen auf einen Ausgang beschreibt.

- [Kap. 5.5.3 "MC_TRACK_REF mit ML_BASIC_TRACK_REF" auf Seite 55](#) enthält ein Array mit 8 Basis-Strukturen (d.h. 8 Nockenscheiben, die 8 Ausgängen zugeordnet sind). Jeder Nockenscheibe können mehrere Nocken zugeordnet sein

ML_TechBase.library

- [Kap. 5.5.2 "MC_CAMSWITCH_REF mit ML_BASIC_CAMSWITCH_REF"](#) auf Seite 54 enthält ein Array mit 32 Basis-Strukturen, die jede eine einzelne Nocke beschreibt
- [Kap. 5.5.4 "MC_OUTPUT_REF"](#) auf Seite 55 enthält ein Array mit 8 booleschen Variablen, die den Ausgängen des Nockenschaltwerkes zugeordnet sind

Beispielprogramm

Ein kurzes Beispiel soll das Konzept erklären, wie der Funktionsbaustein programmiert wird. Die verschiedenen Strukturelemente werden später im Detail erklärt.

Angenommen, eine einzelne Nocke soll angelegt werden:

- mit Nummer 4 auf Nockenscheibe 3
- mit einer "FirstOnPosition" von 35 und
- einer "LastOnPosition" von 55
- die Nocke soll 10ms früher auslösen und die Nockenscheibe sollte eine Hysterese von 3 haben
- die Nocke soll positionsbezogen, nicht zeitbezogen, ein/ausschalten und nur in positiver Richtung wirksam sein

Die Variablendeklaration und der Funktionsbaustein-Aufruf sehen dann folgendermaßen aus:

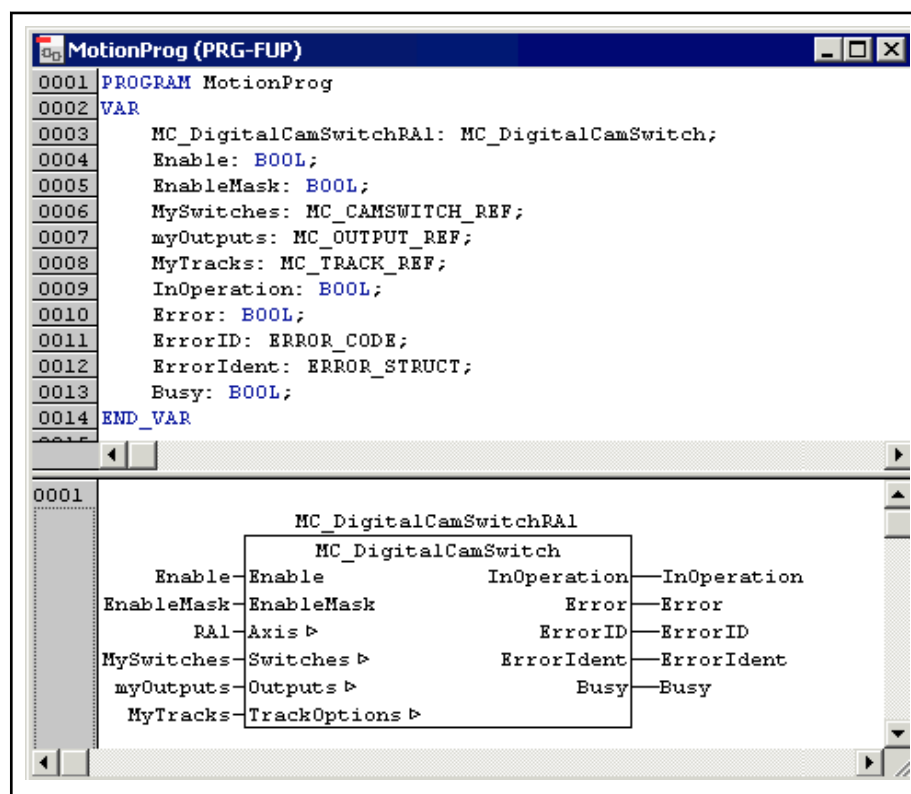


Abb.5-44: Deklaration der Variablen und Funktionsbaustein-Aufruf

Beachten Sie die drei Eingangs-Strukturen "MySwitches", "MyOutputs" und "MyTracks", die alle Eigenschaften der Nocken, Nockenscheiben und Ausgänge definieren. Um diese Strukturen besser verstehen zu können, ist eine nähere Betrachtung der Daten-Inhalte der Strukturen, welche die gewünschten Ergebnisse des obigen Beispiels liefern, notwendig:

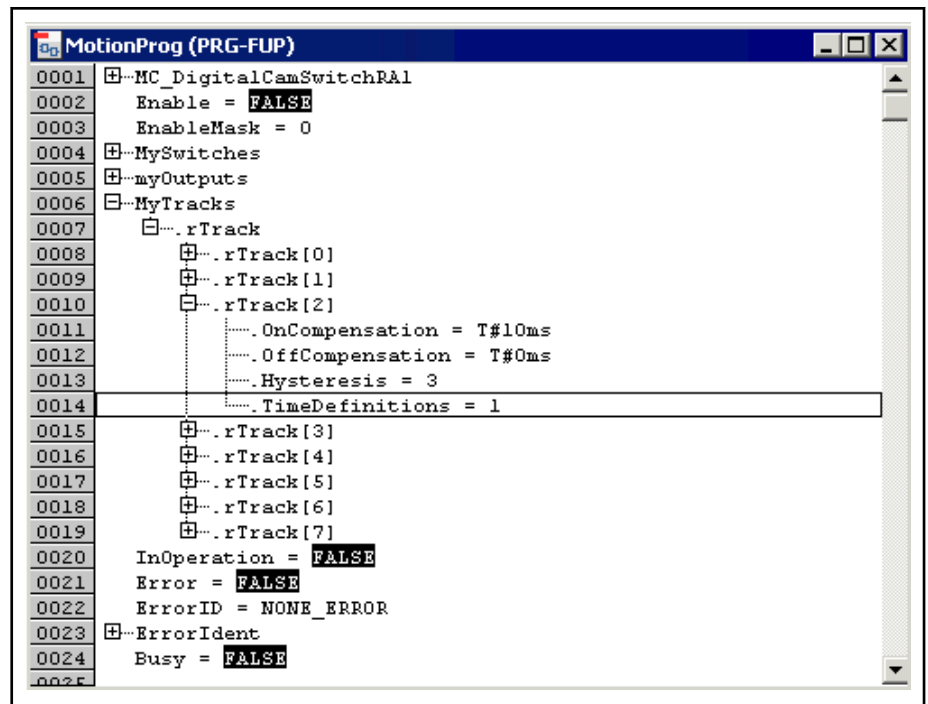


Abb.5-45: Inhalt der MyTracks-Struktur

Zuerst wird die Nockenscheibe definiert (Bild oben).

1. Da Nockenscheibe 3 benutzt werden soll, muss rTrack[2] initialisiert werden, da das Feld mit 0 beginnt.
2. Die OnCompensation-Zeit wird auf die gewünschten 10ms gesetzt. Damit die Nocke früher und nicht später schaltet, muss Bit 0 in "TimeDefinitions" gesetzt werden, welches die "OnCompensation"-Zeit als negativ definiert.
3. Zum Schluss muss noch die Hysteresis auf den gewünschten Wert 3 gesetzt werden.

ML_TechBase.library

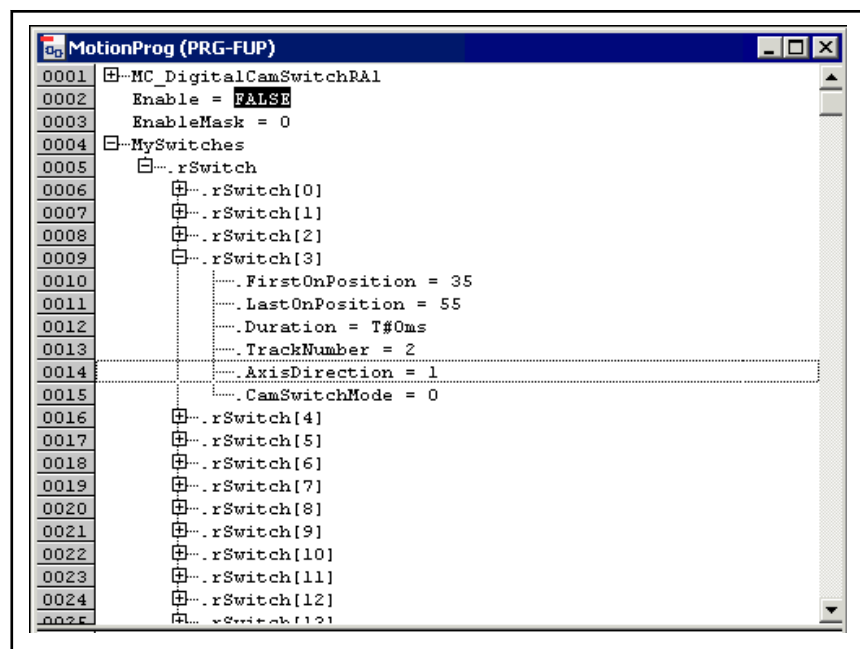


Abb.5-46: Inhalt der MySwitches-Struktur

Nun wird die Nocke definiert (Bild oben).

1. Die Initialisierung der Nocken-Struktur muss mit Index 3 erfolgen, um Nocke 4 zu initialisieren, da das Feld mit Index 0 beginnt.
2. "FirstOnPosition" wird auf den gewünschten Wert 35 gesetzt, "LastOnPosition" auf 55 und die "AxisDirection" wird auf 1 gesetzt, was bedeutet, dass die Nocke nur in positiver Richtung wirksam ist.
3. Um die Nocke einer Nockenscheibe zuzuordnen, wird TrackNumber auf 2 gesetzt; damit wird festgelegt, dass diese Nocke der Nockenscheibe mit Index 2 in der Nockenscheiben-Struktur zugeordnet ist.
4. Zum Schluss muss noch der "CamSwitchMode" auf 0 gesetzt werden, um den positionsbezogenen Mode einzuschalten.



Der Zugriff auf die einzelnen Elemente der REF-Strukturen erfolgt nach nachstehendem Muster:

- MyTracks.rTrack[2].TimeDefinitions
- MySwitches.rSwitches[3].AxisDirection
- MyOutputs.Output[4]

Der Start des Funktionsbausteines wird vorbereitet:

Damit der Funktionsbaustein gestartet werden kann, müssen die restlichen Variablen zum Funktionsbaustein hinzugefügt und Ihre Werte entsprechend gesetzt werden. Speziell der "Enable"-Eingang muss gesetzt werden und "EnableMask" muss das entsprechende Bit gesetzt haben:

4 oder anschaulicher 2#...00000100, Bit 2 für dritte Nockenscheibe "rTrack[2]".

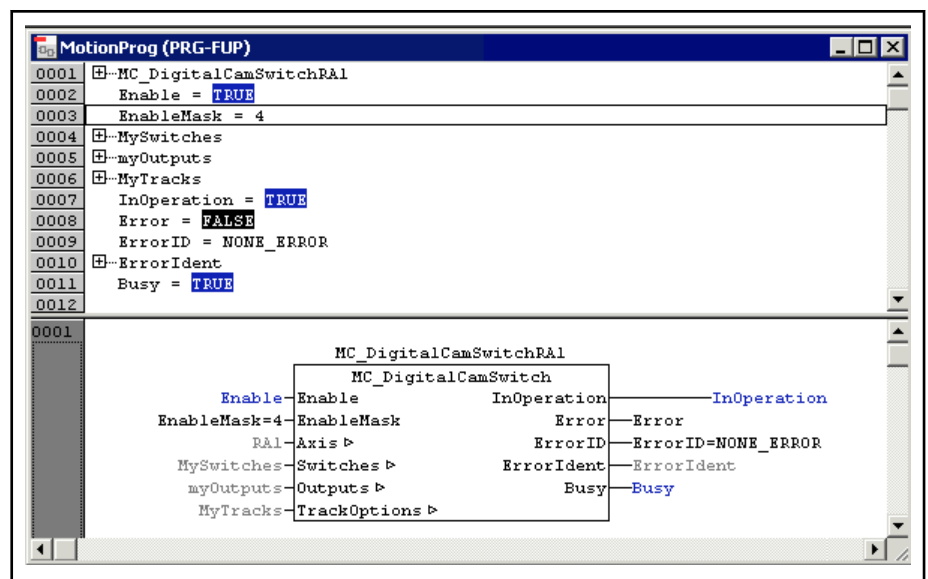


Abb.5-47: Laufender Funktionsbaustein

Funktionsbeschreibung

Der **Axis-Eingang** definiert die Position und Geschwindigkeit die für die Berechnungen benötigt wird. Wechselt der "Axis"-Eingang während der "Enable"-Eingang noch gesetzt ist, bleibt der "Busy"-Ausgang gesetzt, und der "InOperation"-Ausgang wird inaktiv. Dies geschieht aufgrund der Tatsache, dass einige interne Werte zuerst aktualisiert werden müssen, bevor die Berechnung fortgeführt wird. Dies benötigt einige SPS-Takte. Nach dieser Initialisierungssequenz wird der "InOperation"-Ausgang wieder aktiv.

Wird der **Enable**-Eingang das erste Mal gesetzt, wird auch der "Busy"-Ausgang gesetzt, der "InOperation"-Ausgang bleibt jedoch inaktiv. Dies geschieht ebenfalls aufgrund der Tatsache, dass einige interne Werte zuerst aktualisiert werden müssen, bevor die Berechnung fortgeführt wird. Dies benötigt einige SPS-Takte. Nach dieser Initialisierungssequenz wird der "InOperation"-Ausgang wieder aktiv.

Mit dem **EnableMask**-Eingang können die Ausgänge schnell freigegeben oder gesperrt werden. Das niederwertigste Bit ist der niedrigsten "TrackNumber" zugeordnet. Wird der Wert auf 1, bzw. TRUE gesetzt, dann ist die entsprechende "TrackNumber" freigegeben. Durch die Beschränkung auf die 8 niederwertigsten Bits stehen nur 8 Nockenscheiben pro Funktionsbaustein zur Verfügung. Ist das Bit nicht gesetzt, wird der entsprechende Ausgang auf 0 gesetzt, ungeachtet dessen, ob die aktuelle Position innerhalb des Bereiches einer Nocke liegt. Wenn "EnableMask" 0 ist, wird "InOperation" inaktiv, da keiner der Ausgänge freigegeben sind.

Jede Nocke hat eine Start- und Endposition. Dadurch kann der Benutzer für jede Nocke eine **FirstOnPosition** und eine **LastOnPosition** (oder Zeit) festlegen. Zusätzlich zu einer mechanischen Nocke besteht die Möglichkeit, die Nocke für eine bestimmte Zeit zu aktivieren und ihr für eine bestimmte Zeit eine Vorhaltezeit und Hysterese zuzuordnen. Wenn ("FirstOnPosition" > "LastOnPosition") ist, verhält sich die Nocke invers und ist nur ausgeschaltet innerhalb dieser Positionen.

Bereich/Modulo-Berechnungen

Das PLS wird höchstwahrscheinlich nur für rotatorische Abläufe eingesetzt. Dennoch könnte man das PLS auch für eine Linearachse, die vorwärts und rückwärts läuft, einsetzen. In beiden Fällen basieren die PLS-Einheiten auf die für die Achse definierten Einheiten.

ML_TechBase.library

AxisDirection gibt an, in welcher Bewegungsrichtung die Nocke aktiv wird. Ist "AxisDirection" 0 (default), wird die Nocke in beiden Bewegungsrichtungen aktiv. Ist "AxisDirection" 1, wird der Ausgang nur gesetzt, wenn in positiver Richtung verfahren wird; ist "AxisDirection" auf 2 gesetzt, wird der Ausgang nur gesetzt, wenn in negative Richtung verfahren wird.

CamSwitchMode wechselt zwischen Positions- oder Zeitangabe. "CamSwitchMode" = 0 bedeutet Positionsmodus, dabei schaltet der Ausgang an der "FirstOnPosition" ein und an der "LastOnPosition" aus. Ist "CamSwitchMode" = 1, wird der Ausgang an der "FirstOnPosition" eingeschaltet, und nach der angegebenen "Duration" (Dauer) wieder ausgeschaltet.

Duration: Zeitdauer, die eine zeitgesteuerte Nocke eingeschaltet ist. Wenn die Position nach Ablauf dieser Zeit noch innerhalb des Nockenbereichs ist, wird der Ausgang nicht neu gesetzt, bis die Position die Nocke ausgeschaltet hat. Die maximal mögliche Zeitdauer ist abhängig von der verwendeten Hardware. Bei zu langer Einschaltdauer wird der Fehlercode 16#F0260006 erzeugt.

Die Vorhaltezeit (**OnCompensation** und **OffCompensation**) kann positiv oder negativ sein. Negativ bedeutet, dass der Ausgang schaltet bevor die Schaltposition erreicht wird. Die Vorhaltezeit wird in ms angegeben (siehe auch "TimeDef").

Hysteresis: Dieser Parameter verhindert, dass der Ausgang zu oft schaltet, wenn die Achse nahe dem Schaltpunkt ist und die aktuelle Position um den Schaltpunkt schwankt. Die Hysteresis ist ein Teil von "MC_TRACK_REF", was bedeutet, dass für jede Nockenscheibe eine eigene Hysteresis einstellbar ist.

TimeDef: Dieser Parameter wurde hinzugefügt, um zwischen positiven und negativen Zeiten für die "On/OffCompensation" zu unterscheiden. Wenn Bit 0 gesetzt ist, dann ist die "OnCompensation"-Zeit negativ, ist Bit 1 gesetzt, dann ist die "OffCompensation"-Zeit negativ.

Beispiel:

"OffCompensation" ist auf 300 gesetzt, "TimeDef" auf 2.

Das bedeutet, dass die Vorhaltezeit "-300ms" ist.

Die folgende Abbildung zeigt die verschiedenen Variablen und Strukturen und ihre entsprechende Benutzung:

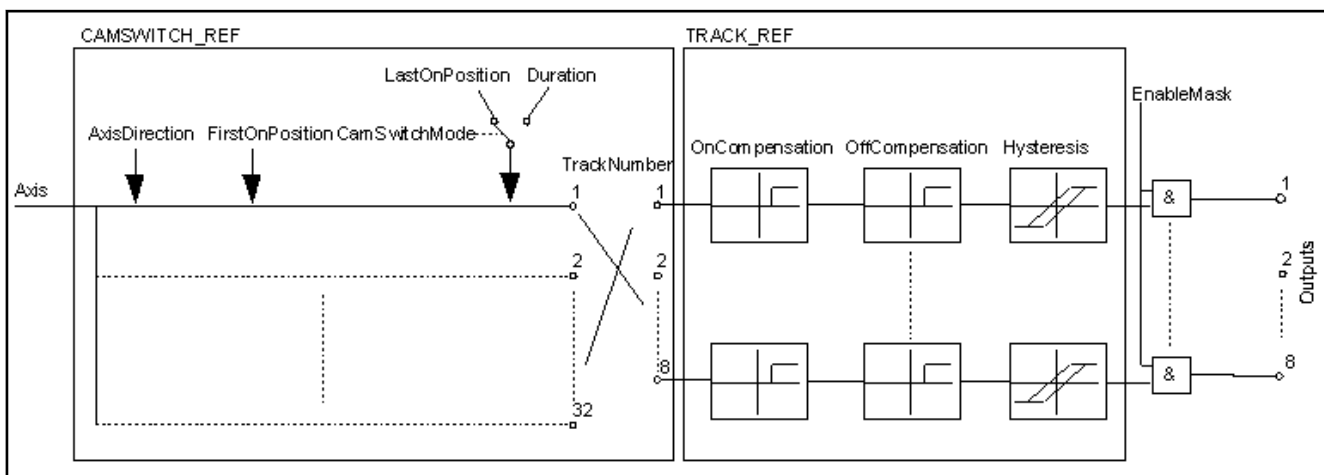


Abb.5-48: Beziehung der Strukturen und entsprechende Funktionalität des Bausteins MC_DigitalCamSwitch

Das folgende Bild zeigt das mechanische Äquivalent des digitalen Nockenschaltwerks.

Beachten Sie bitte, dass nicht alle Funktionen dargestellt wurden, da sie entweder nicht möglich, bzw. schwer darstellbar sind.

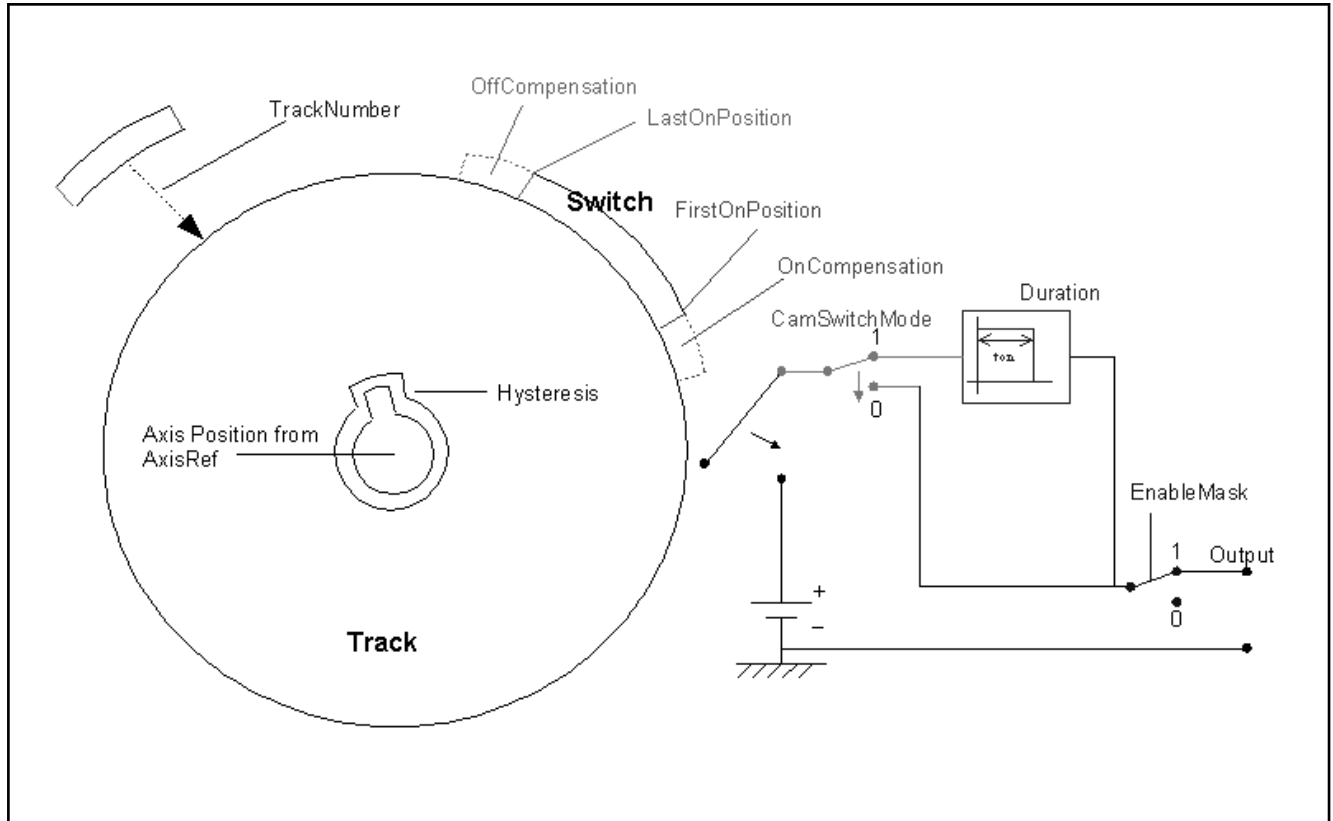


Abb.5-49: Mechanisches Äquivalent des digitalen Nockenschaltwerks

Zeitdiagramm

Das folgende Beispiel soll die Benutzung der "MC_CAMSWITCH_REF"-Struktur und der zeitbasierenden Nocken zeigen. In der folgenden Liste finden Sie eine vorgegebene Konfiguration von Werten:

Parameter	Typ	Switch01	Switch02	Switch03	Switch04	...	SwitchN
TrackNumber	INT	1	1	1	2		
FirstOnPosition [u]	REAL	2000	2500	4000	3000		
LastOnPosition [u]	REAL	3000	3000	1000	-		
AxisDirection	INT	1=Pos	2=Neg	0=Beide	0=Beide		
CamSwitchMode	INT	0=Position	0=Position	0=Position	1=Zeit		
Duration	TIME	-	-	-	1350		

Abb.5-50: Beispiel MC_CAMSWITCH_REF-Struktur

Dieses Beispiel benutzt die Werte aus dem obigen Beispiel "MC_CAMSWITCH_REF". Es benutzt weder "On/OffCompensation" noch "Hysteresis".

Dies ist das Verhalten der Ausgänge, wenn die Achse sich kontinuierlich in positive Richtung bewegt. Die Achse ist eine Modulo-Achse mit einer Modulolänge von 5000u (Einheiten).

ML_TechBase.library

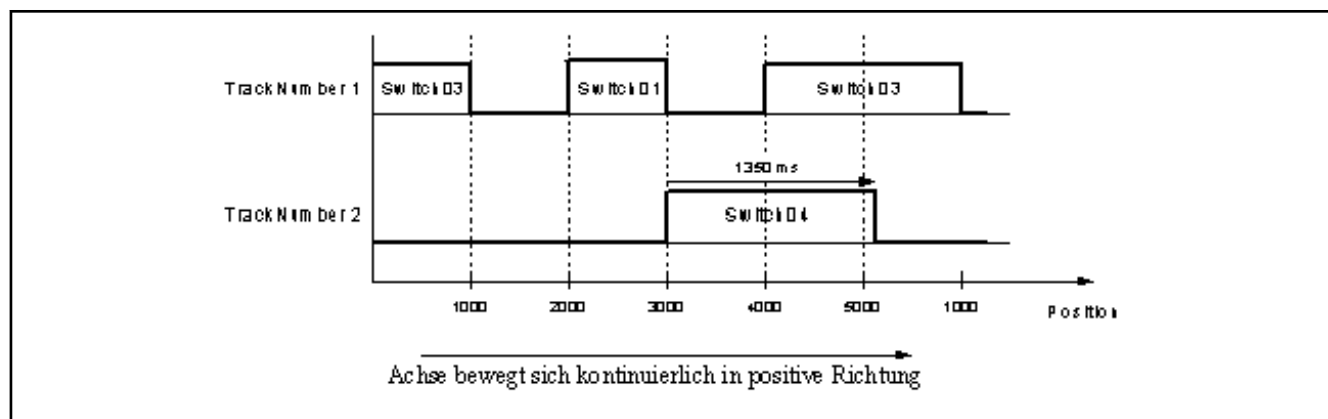


Abb.5-51: Verhalten der Ausgänge im Beispiel

Detaillierte Beschreibung der Nocke "Switch01"

Dieses Beispiel benutzt zusätzlich "OnCompensation" = -125ms und "OffCompensation" = +250ms.

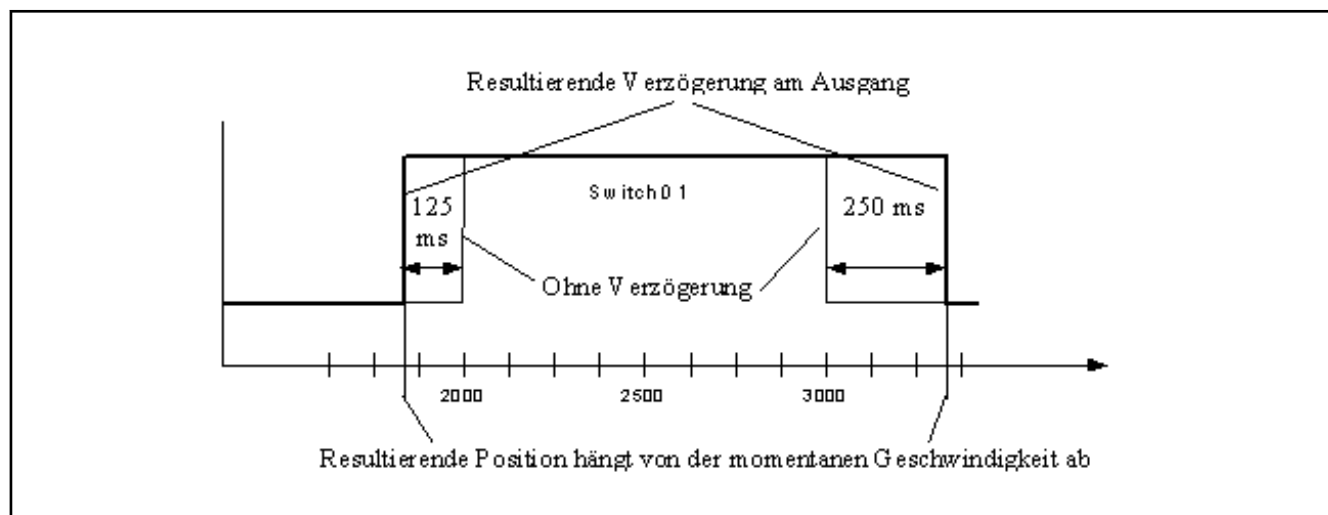


Abb.5-52: Detailliertes Verhalten von Switch01 beim Bewegen in positive Richtung

Dies ist das Verhalten der Ausgänge, wenn sich die Achse ohne "On-" oder "OffCompensation" und ohne "Hysteresis" kontinuierlich in negative Richtung bewegt:

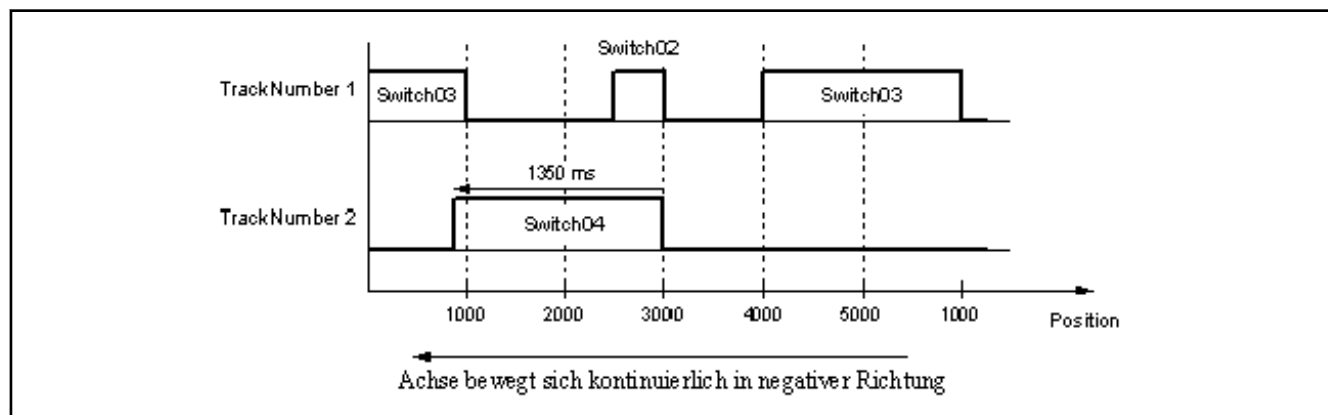


Abb.5-53: Detailliertes Verhalten der Nocken beim Bewegen in negative Richtung

Realisierungsdetails

Die PLCOpen-Erweiterung spezifiziert das Interface der PLS-Funktionalität. Jedoch sind einige Anwendungsdetails nicht spezifiziert. Deshalb wurden diese für eine Durchführung mit besserem Laufzeitverhalten definiert.

OnCompensation/OffCompensation

PLCOpen definiert die "On-/OffCompensation" nur für Bewegungen in positive Richtung. In diesem Fall bezieht sich die "OnCompensation" auf die "FirstOnPosition", während die "OffCompensation" sich auf die "LastOnPosition" bezieht. Wie ist dies zu definieren für Bewegungen in negative Richtung?

Basierend auf Applikationen bei denen die Vorhaltezeiten benutzt werden, um bei Aktuatoren die An- und Ausschaltverzögerungen zu kompensieren bedeutet dies, dass die "OnCompensation" für die steigende Flanke des Signals zuständig ist. Mit anderen Worten: fährt man in positive Richtung, dann bezieht sich die "OnCompensation" auf die "FirstOnPosition", fährt man in negative Richtung, dann bezieht sich die "OnCompensation" auf die "LastOnPosition". Die "OffCompensation" ist entsprechend definiert

Die folgende Abbildung zeigt das Verhalten für "OnCompensation"= -0.1s und "OffCompensation"= +0.2s:

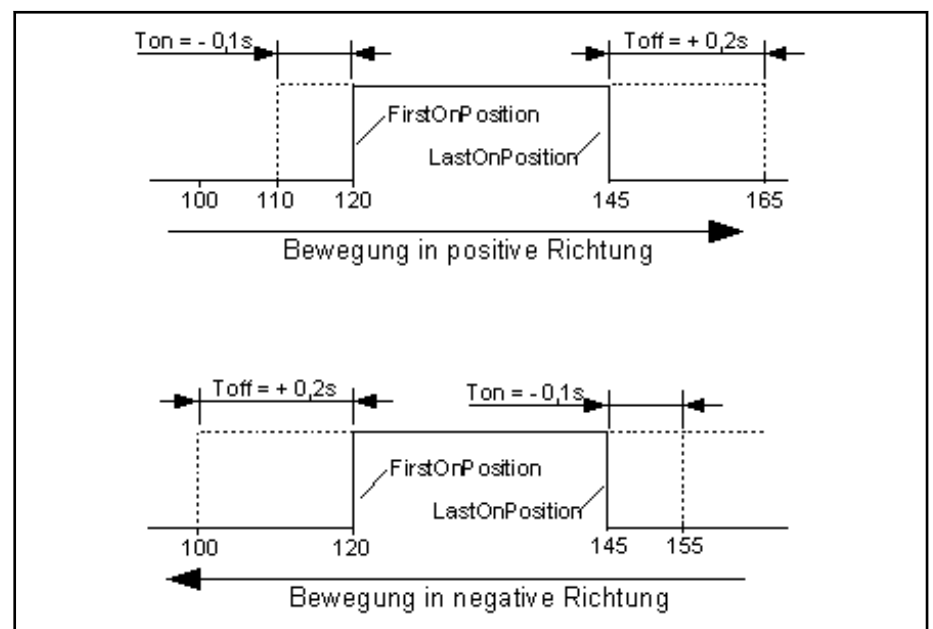


Abb.5-54: Definition der OnCompensation- und OffCompensation-Zeit

Duration

Die "Duration" (Dauer) ist definiert als die Zeit, die die Nocke eingeschaltet bleibt, nachdem sie eingeschaltet wurde. Die Frage stellt sich was passiert, wenn die Zeitdauer abgelaufen ist, aber die momentane Position noch innerhalb der "FirstOnPosition" und "LastOnPosition" ist. Soll der Ausgang sofort eingeschaltet, oder soll gewartet werden, bis die aktuelle Position sich aus der "FirstOnPosition" und "LastOnPosition" heraus bewegt hat?

Es wurde entschieden auf eine steigende Flanke des Nockenausgangs zu warten. Mit anderen Worten: die aktuelle Position muss ausserhalb der "FirstOnPosition" und "LastOnPosition" sein, bevor die "Duration" wieder neu angestoßen werden kann.

Die gleiche Überlegung gilt für die "OnCompensation"/"OffCompensation", wobei die "Duration" mit der steigenden Flanke einer Nocke beginnt, d.h. an der "LastOnPosition" beim Bewegen in negativer Richtung.

Der Sachverhalt wird in der folgenden Abbildung erläutert:

ML_TechBase.library

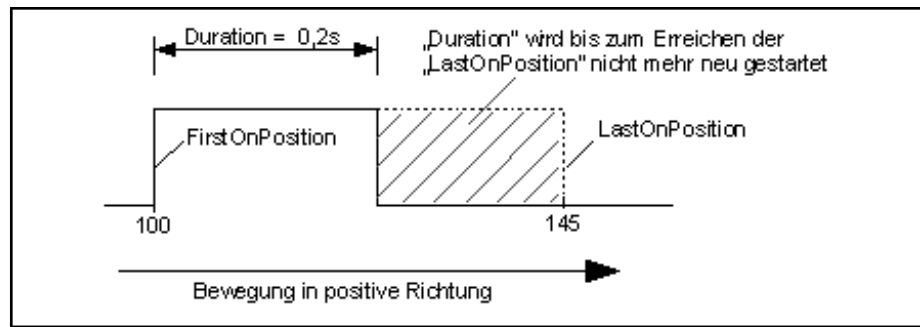


Abb.5-55: Definition der Zeitdauer

Kombination von OnCompensation/OffCompensation und Duration

Sollen bei der Benutzung einer zeit-basierenden Nocke mit "On-/OffCompensation"-Zeiten diese dann berücksichtigt werden oder nicht? Die "OnCompensation" wird berücksichtigt. Daraus resultiert eine frühere/spätere Aktivierung des Ausgangs. Die "OffCompensation"-Zeit ist allerdings schwieriger, da der Ausgang nach der Vorhaltezeit ("Duration") ausgeschaltet wird. Die "OffCompensation"-Zeit ist nur relevant, wenn man bestimmen muss, dass der Ausgang retriggered werden soll (schraffierter Bereich im oberen Bild).

In dieser Anwendung wurden beide, "OnCompensation" und "OffCompensation", in Betracht gezogen.

Wie eine Nocke nicht verwendet wird

Das PLS hat bis zu 32 Nocken. Um festzustellen, ob ein Nocken benutzt wird oder nicht gibt es das Element "TrackNumber" in der MC_CAMSWITCH_REF-Struktur.

Diese Variable beinhaltet die Nockenscheiben-Nummer. Gültige Werte dafür sind 0 bis 7, da es maximal 8 Nockenscheiben gibt. Die Default-Einstellung dieser Variable ist 8, sie wird als nicht benutzt markiert. Jede ungültige Nummer (8 bis 65535) wird interpretiert als eine nicht benutzte Nocke und wird deshalb auch nicht von der Firmware ausgewertet.

Einschränkungen des PLS

Die Einschränkungen des PLS werden in diesem Abschnitt diskutiert. Die Einschränkungen sind lediglich Definitionen, die jedoch zu keinem Fehler führen. Das Systemverhalten wird über diese Definitionen festgelegt, der Anwender muss deshalb mit unterschiedlichem Verhalten rechnen.

"Verlorene" Schaltvorgänge

Das Abtast-Theorem gibt an, dass die Abtast-Rate doppelt so hoch ist, wie die höchste Signalfrequenz die man braucht, um das Signal erkennen zu können. Dies gilt auch für das PLS. Das PLS tastet die Positionsdaten jedes Mal ab, wenn es von der SPS aufgerufen wird. Das ist normalerweise die Motion-synchrone Task, oder aber jede andere zyklische Task. Jede Nocke die kürzer eingeschaltet ist als die Abtast-Rate, wird nicht richtig erkannt. Die folgenden Gleichungen zeigen, wie man die Geschwindigkeit oder die Größe der Nocke berechnet, bei denen Schaltvorgänge verloren gehen.

Rotatorische Systeme:

$$1. \omega = 2\pi n$$

n ist die Umdrehung pro Sekunde.

$$2. \omega = \rho / t$$

ρ ist der Winkel in rad und t die Zeit in Sekunden.

3. Konvertierungsfaktor von Grad to rad:

$$\pi / 180$$

Nimmt man Gleichungen 1 und 2 erhält man

$$\rho[\text{rad}] = 2\pi n t$$

für den Winkel, oder

$$n \left[\frac{\text{Umdrehungen}}{\text{Sekunde}} \right] = \frac{\rho}{2\pi t}$$

Passt man die Faktoren für gemeinsame Einheiten an, so erhält man:

Gleichung A1.)

$$\rho[^\circ] = n[U/min] * t[ms] * 0,006$$

oder für die Geschwindigkeit:

Gleichung B1.)

$$n[U/min] = \frac{\rho[^\circ]}{t[ms]} * 166,6$$

A1.) und B1.) liefern dem Anwender die beiden notwendigen Gleichungen, um festzustellen, ob Schaltvorgänge verloren gehen können. Zum Beispiel, wenn die maximale Drehzahl von 1000U/min bekannt ist, und das PLS in einer zyklischen Task mit einer Zykluszeit von 2 ms aufgerufen wird. Was ist dann die minimale Grösse einer Nocke um sicherzustellen, dass sie von dem PLS erkannt wird? Nimmt man Gleichung A1.) ergibt das:

$$\rho[^\circ] = 1000U/min * 2ms * 0,006 = 12$$

D.h., eine Nocke muss mindestens 12° groß sein, um bei einer Drehzahl von max. 1000U/min noch sicher erkannt zu werden.

Lineare Systeme:

$$1. \ v = s / t ,$$

s ist die Distanz in m und t die Zeit in Sekunden.

Nimmt man die Zykluszeit der SPS als Abtastzeit für das PLS und addiert die entsprechenden Faktoren um die Einheiten anzugleichen, so erhält man die folgenden Gleichungen:

Gleichung A2.)

$$s[mm] = v \left[\frac{mm}{min} \right] * t[ms] * 1,66 * 10^{-5}$$

oder für die Geschwindigkeit:

Gleichung B2.)

$$v \left[\frac{mm}{min} \right] = \frac{s[mm]}{t[ms]} * 60000$$

Diese Gleichungen gelten auch für Systeme die Inch als Einheit für die Strecke haben:

Gleichung A2.)

ML_TechBase.library

$$s[\text{inch}] = v \left[\frac{\text{inch}}{\text{min}} \right] * t[\text{ms}] * 1,66 * 10^{-5}$$

oder für die Geschwindigkeit:

Gleichung B2.)

$$v \left[\frac{\text{inch}}{\text{min}} \right] = \frac{s[\text{inch}]}{t[\text{ms}]} * 60000$$

Beispiel: Eine Nocke in einem linearen System ist 0,02 Inches lang. Das PLS ist in einer SPS-Task mit 2ms Zykluszeit. Welche maximale Geschwindigkeit ist möglich, damit die Nocke noch sicher erkannt wird?

Nimmt man Gleichung B2.) erhält man das Ergebnis:

$$v \left[\frac{\text{inch}}{\text{min}} \right] = \frac{0,02\text{inch}}{2\text{ms}} * 60000 \frac{1}{\text{min}} = 600$$

Nocken haben scheinbar die falsche Größe

Das vorhergehende Kapitel handelte über verlorene Schaltvorgänge. Auch wenn die Bedingungen für verlorene Nocken nicht erfüllt sind, können die Schaltzeitpunkte ungenau sein. Das PLS kann nur dann den Zustand seiner Ausgänge wechseln, wenn es von der SPS aufgerufen wird. Wenn z.B. eine Nocke für 3,7 ms bei einer vorgegebenen Geschwindigkeit, geschlossen sein soll, das PLS aber nur alle 2 ms aufgerufen wird, wird der Ausgang entweder für 2 ms (Fall B) oder 4 ms (Fall A) auf "On" gesetzt, abhängig von dem Abtastzeitpunkt des PLS. Eine nähere Erläuterung hierzu sehen Sie in der folgende Abbildung

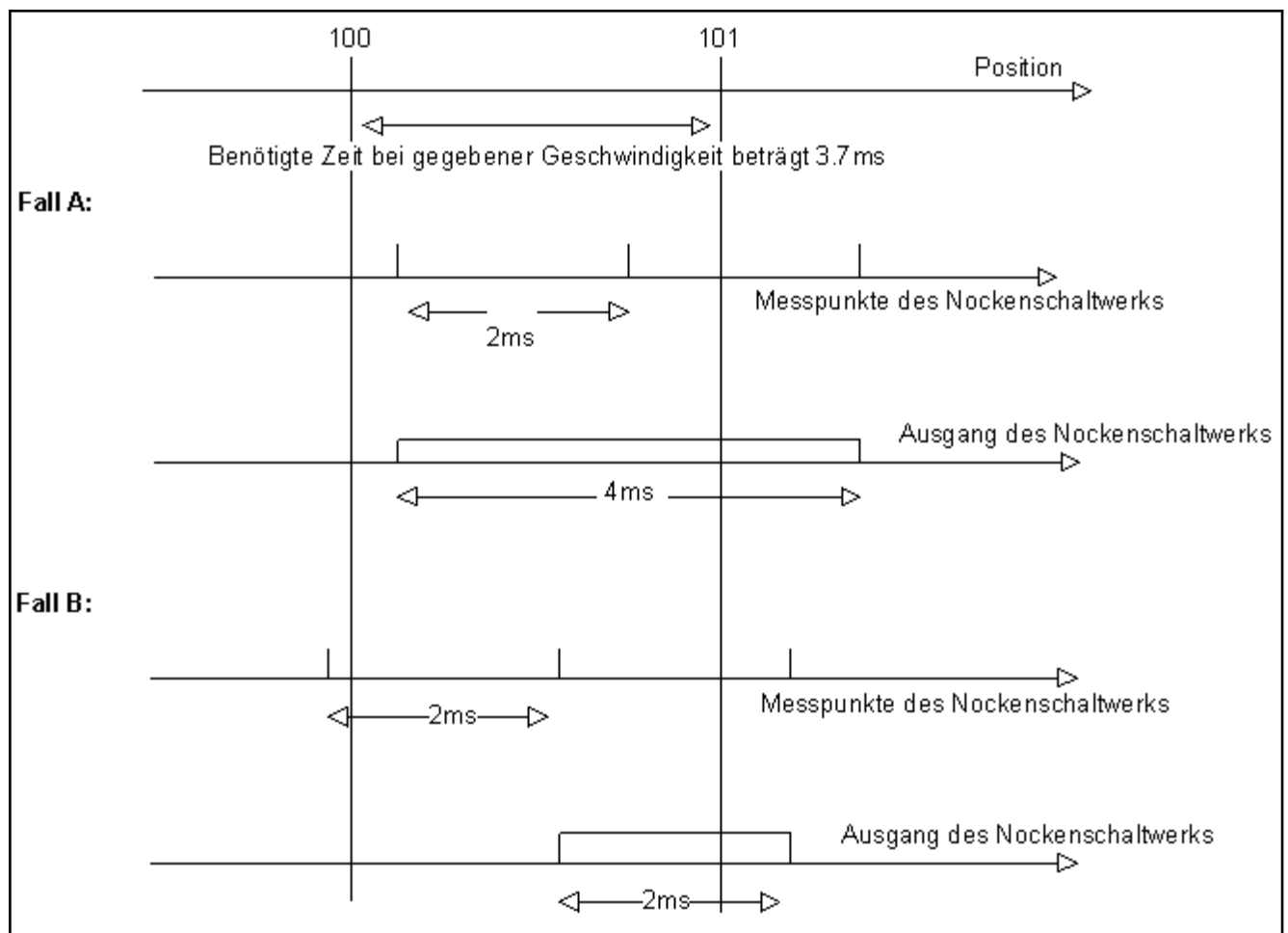


Abb.5-56: Ungenaue Größe der Nocken

Schaltvorgang erfolgt zweimal beim Beschleunigen/Verzögern

Werden "OnCompensation" und "OffCompensation" für eine Nocke benutzt, ist es möglich, dass sie mehrfach ein- und ausgeschaltet wird, wenn die PLS Masterposition beschleunigt oder abbremst. Dies passiert aufgrund der Tatsache, dass das PLS eine konstante Geschwindigkeit voraussetzt, was unter bestimmten Umständen jedoch nicht gegeben ist. Das größere Problem taucht beim Abbremsen auf. Einige Betrachtungen, wann eine Nocke zu mehreren Zeiten aktiviert werden kann, folgt.

Man kann die folgenden Gleichungen aufstellen:

$$1.) s_v = v_0 * t_{lead}$$

dabei ist s_v der Weg, der bei der vorgegebenen Geschwindigkeit v_0 in der "OnCompensation"-Zeit t_{lead} zurückgelegt wird.

$$2.) s_b = v_0 * t - \frac{1}{2} a t^2$$

s_b ist der Weg, der nötig ist zum Abbremsen bei einer vorgegebenen Startgeschwindigkeit v_0 und Verzögerung a in der Zeit t .

Außerdem die allgemeinen Gleichungen:

$$3.) v = a * t \text{ oder } t = v / a$$

ML_TechBase.library

Die "OnCompensation"-Zeit eines programmierbaren Nockenschaltwerks "bewegt" die Nocke in Richtung niedriger Positionen entsprechend Gleichung 1.). Der Weg, der benötigt wird, um bei einer vorgegebenen Geschwindigkeit zum Stillstand zu kommen, kann mit Gleichung 2.) ermittelt werden.

Die allgemeine Regel lautet deshalb:

Wenn die Position mit $v = v_0$ erreicht ist, damit der Ausgang eines programmierbaren Nockenschaltwerks mit "OnCompensation"-Zeit eingeschaltet werden soll – und die Eingangsposition in einer Weise zum Stoppen abgebremst wird, dass die Stopposition gleichzeitig auch die Einschaltposition des programmierbaren Nockenschaltwerks ohne "OnCompensation"-Zeit ($v = 0$) ist, dann wird der Ausgang dazwischen nicht ausgeschaltet.

Wenn wir Gleichung 1.) gleichsetzen mit Gleichung 2.), erhalten wir die Gleichung, mit der die Nocke nur einmal eingeschaltet wird. Wenn wir etwas schneller abbremsen, dann wird die Nocke mehrfach eingeschaltet.

$$s_b = s_v \Rightarrow v_0 t - \frac{1}{2} a t^2 = v_0 t_{\text{lead}} \Rightarrow v_0 \frac{v_0}{a} - \frac{1}{2} a \frac{v_0^2}{a^2} = v_0 t_{\text{lead}}$$

$$\Rightarrow \frac{v_0^2}{2a} = v_0 t_{\text{lead}}$$

$$4.) \Rightarrow v_0 = 2 t_{\text{lead}} a$$

In Gleichung 4.) kann man sehen, dass die Startgeschwindigkeit v_0 bei der es möglich ist, dass ein Nockenschalter mehrfach eingeschaltet wird, grösser wird durch Erhöhung der "OnCompensation"-Zeit t_{lead} oder Verzögerung a .

Es gibt keine Lösung für dieses Problem. Man kann lediglich die "OnCompensation"-Zeit und Verzögerung für die gewünschte Applikation so klein wie möglich halten.

Eine andere Möglichkeit ist es, die "Hysterese" zu benutzen, damit die Schwankungen aufgrund der Beschleunigung/Verzögerung darin berücksichtigt werden.

Die gleichen Überlegungen können angewandt werden für das Beschleunigen, und/oder negative Geschwindigkeiten.

Fehlercodes Der Funktionsbaustein nutzt die Fehlertabelle **MLC_TABLE**, 16#0030. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
SYSTEM_ERROR (16#7FFF)	16#F0260001	0	Fehler beim Initialisieren des Funktionsbausteins. Es konnte keine Referenz zum Parametersystem hergestellt werden
COMMUNICATION_ERROR (16#0002)	16#F0260004	xx	Laufzeitfehler im Funktionsbaustein: Parameter konnte nicht gelesen werden xx bezeichnet die Parameternummer (A-Parameter)

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR (16#0001)	16#F0260005	0	Laufzeitfehler im Funktionsbaustein: Ungültige Kombination von Einheiten für Geschwindigkeit, Zeit und Position
INPUT_INVALID_ERROR (16#0001)	16#F0260006	0	Laufzeitfehler im Funktionsbaustein: Der Wert "Duration" einer Nocke ist zu groß
STATE_MACHINE_ERROR (16#0005)	16#F0260007	0	Laufzeitfehler im Funktionsbaustein: Ungültiger Zustand der Zustandsmaschine
RESSOURCE_ERROR	16#F0260040	0	Referenz zum Basissystem konnte nicht gefunden werden
RESSOURCE_ERROR	16#F0260041	0	Der Baustein konnte nicht instantiiert werden
SYSTEM_ERROR	16#F0260042	0	Funktionsbaustein darf nicht im RETAIN Datenbereich liegen

Abb.5-57: Fehlercodes Funktionsbaustein MC_DigitalCamSwitch

5.6 Funktionsbaustein Temperaturregler

5.6.1 IL_TempControlType01

Kurzbeschreibung

Der Funktionsbaustein IL_TempControlType01 dient zur Regelung von Temperaturen. Er kann die Reglerparameter entweder selbst ermitteln oder mit Vorgaben arbeiten. Komfortfunktionen wie Überwachung, Pause, Handbetrieb können zusätzlich aktiviert werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx18 (in Vorbereitung)	in Vorbereitung

Abb.5-58: Referenztablette Funktionsbaustein IL_TempControlType01

ML_TechBase.library

Schnittstellenbeschreibung

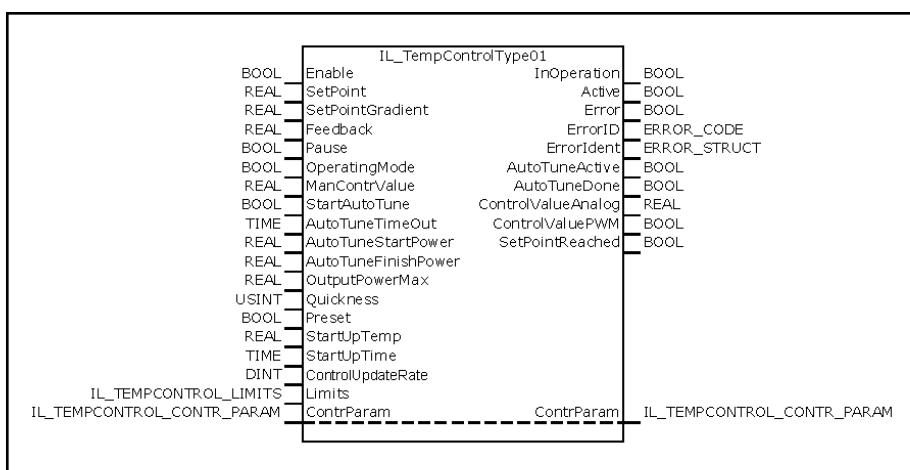


Abb.5-59: Funktionsbaustein IL_TempControlType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Der Funktionsbaustein ist aktiv, wenn dieser Eingang "TRUE" ist. Der FB wird mit einer fallenden Flanke auf seine Ausgangswerte zurückgesetzt. Der Eingang wird in jedem Zyklus gelesen
	SetPoint	REAL	Solltemperatur Einheit in [u] Der Eingang wird in jedem Zyklus gelesen
	SetPointGradient	REAL	Maximale Änderung des Parameters "SetPoint" in [u/s]. Der Eingang wird in jedem Zyklus gelesen. Voreingestellter Wert ist "0 [u/s]" (nicht aktiv)
	Feedback	REAL	Eingang für die Isttemperatur Einheit in [u] Der Eingang wird in jedem Zyklus gelesen
	Pause	BOOL	Dieser Eingang "friert" Stellgröße und Reglerberechnung ein. Der Eingang wird in jedem Zyklus gelesen. Voreingestellter Wert ist "FALSE" (nicht aktiv)
	OperatingMode	BOOL	Setzt den Reglermodus von Automatik auf Handbetrieb: AUTO:=FALSE, MANUAL:=TRUE. Voreingestellter Wert ist "FALSE" (AUTO). Der Eingang wird in jedem Zyklus gelesen

E/A-Typ	Name	Datentyp	Beschreibung
	ManContrValue	REAL	Gibt die Stellgröße in [%] bei Handbetrieb an, wird aber nur aktiv gelesen, wenn "OperatingMode:=1". Der Eingang wird in jedem Zyklus gelesen. Voreingestellter Wert ist "0%" (keine Stellgröße)
	StartAutoTune	BOOL	Eine steigende Flanke startet den Identifikationsprozess. Die PID-Reglerparameter werden im Anschluss berechnet, jedoch nicht aktiv bis eine fallende Flanke ansteht. Der Eingang wird in jedem Zyklus gelesen. Voreingestellter Wert ist "FALSE" (nicht aktiv)
	AutoTuneTimeOut	TIME	Watchdog, der die Zeitüberwachung während der Identifikation vornimmt. Voreingestellter Wert ist " $10000 \cdot T_A$ ". Der Eingang wird bei einer steigenden Flanke von "StartAutoTune" gelesen
	AutoTuneStartPower	REAL	Stellgröße in [%], die die untere Grenze des Identifikationsprozesses darstellt. Voreingestellter Wert ist "0%". Der Eingang wird bei einer steigenden Flanke von "StartAutoTune" gelesen
	AutoTuneFinishPower	REAL	Stellgröße in [%], die die obere Grenze des Identifikationsprozesses darstellt. Voreingestellter Wert ist "50%". Der Eingang wird bei einer steigenden Flanke von "StartAutoTune" gelesen
	OutputPowerMax	REAL	Maximal ausgegebene Stellgröße in [%]. Voreingestellter Wert ist "100%". Der Eingang wird in jedem Zyklus gelesen
	Quickness	USINT	Schnelligkeit des Reglers Voreingestellter Wert ist "3" (ganz leichtes Überschwingen). Der Eingang kann im Bereich von "1" (schnelle Regelung, starkes Überschwingen) bis "5" (langsame Regelung, kein Überschwingen) gewählt werden. Der Eingang wird bei steigender Flanke von "StartAutoTune" gelesen
	Preset	BOOL	Eine steigende Flanke übernimmt die Reglerparameter der "ContrParam"-Struktur. Voreingestellter Wert ist "FALSE" (nicht aktiv). Der Eingang wird in jedem Zyklus gelesen und wirkt nur wenn "Enable" = TRUE

ML_TechBase.library

E/A-Typ	Name	Datentyp	Beschreibung
	StartUpTemp	REAL	Anfahrtemperatur in [u]. Der Eingang wird bei steigender Flanke von "Enable" gelesen. Voreingestellter Wert ist "0u" (nicht aktiv). Wird nur aktiv, wenn Eingang "StartUpTime" mit einem Wert ungleich "0" belegt wird
	StartUpTime	TIME	Anfahrzeit, d.h., Zeit, in der die Anfahrtemperatur ("StartUpTemp") gehalten wird bis in die "normale" Regelung übergegangen wird. Voreingestellter Wert ist "T#0s" (nicht aktiv). Der Eingang wird bei steigender Flanke von "Enable" gelesen
	ControlUpdateRate	DINT	Dieser Faktor gibt an, um wieviel langsamer der Regler regelt, im Vergleich zur Task in der sich der Baustein befindet. Aus Multiplikation der Zykluszeit (= Task Intervallzeit) und "ControlUpdateRate" ergibt sich die Abtastzeit
	Limits	IL_TEMPCONTROL_LIMITS	Struktur, die die Alarmschranken in [u] (Einheiten) enthält. Voreingestellter Wert ist jeweils "0u" (nicht aktiv). Der Eingang wird in jedem Zyklus gelesen. Wird die Struktur mit Werten belegt, muss Limits.UpperAlarm > Limits.LowerAlarm und (Setpoint + Limits.UpperAtSetpoint) >= (Setpoint - Limits.LowerAtSetpoint) sein, ansonsten wird ein Fehler erzeugt
VAR_OUTPUT	InOperation	BOOL	Die Temperaturregelung ist bei gesetztem Ausgang aktiv
	Active	BOOL	Dieser Ausgang wird gesetzt, wenn "Enable" gesetzt und der Regler aktiv ist
	Error	BOOL	Ein Fehler ist während des Betriebes aufgetreten. Die Stellgröße wird auf "0%" gesetzt
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem Error-Ausgang enthält dieser Ausgang detaillierte Informationen über den Fehler
	AutoTuneActive	BOOL	Der Identifikationsprozess ist aktiv
	AutoTuneDone	BOOL	Der Identifikationsprozess wurde erfolgreich beendet
	ControlValueAnalog	REAL	Analoger Wert der Stellgröße in %
	ControlValuePWM	BOOL	Digitaler, pulsweitenmodulierter Stellgrößenwert, entweder "0" oder "1"

ML_TechBase.library

E/A-Typ	Name	Datentyp	Beschreibung
	SetPointReached	BOOL	Ein Statusausgang der anzeigt, ob sich der Istwert innerhalb der Schranken ("Setpoint" + "Limits.UpperAtSetpoint") $\geq$ Istwert $\geq$ (Setpoint - "Limits.LowerAtSetpoint") befindet
VAR_IN_OUT	ContrParam	IL_TEMPCONT-ROL_CONTR_PARAM	Diese Struktur enthält die aktuellen Parameter K_P , K_I und K_D des Reglers sowie die Totzeit T_t

Abb.5-60: Schnittstellenvariablen Funktionsbaustein IL_TempControlType01

Schnittstellenbeschreibung Datenstruktur

In der folgenden Tabelle sind die Datenstrukturen des Funktionsbausteins L_TempControlType01 aufgelistet.

Strukturname	Strukturelement	Typ
IL_TEMPCONTROL_LIMITS	UpperAlarm	REAL
	LowerAlarm	REAL
	UpperAtSetpoint	REAL
	LowerAtSetpoint	REAL
IL_TEMPCONTROL_CONTR_PARAM	KP	REAL
	KI	REAL
	KD	REAL
	Tt	TIME

Abb.5-61: Datenstrukturen

Zeitdiagramm

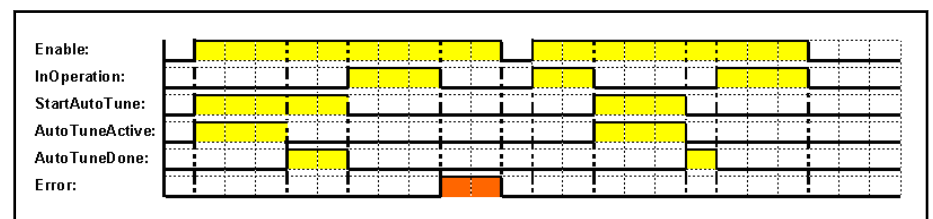


Abb.5-62: Zeitdiagramm

Fehlerbeschreibung

ML_TechBase.library

ErrorID	Table	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	F_RELATED_TABLE	16#1600	Aktueller State	Fehler in der State Machine, ungültiger Zustand
INPUT_RANGE_ERROR	F_RELATED_TABLE	16#1601	16#0001	Eingabewerte für maximalen Ausgabewert ("OutputPowerMax", "AutoTuneStartPower" oder "AutoTuneFinishPower") außerhalb des gültigen Bereichs. Wert muss größer gleich 0 [%] bis max. 100 [%] sein
			16#1602	Eingabewerte für Alarmschranken ungültig (Bedingung: Limits.UpperAlarm > Limits.LowerAlarm)
			16#1603	Eingabewert für manuelle Stellgröße ungültig. Wert muss im Bereich von 0 [%] und 100 [%] liegen
			16#0004	Reglerparameter ungültig
			16#000A	Start- und Endwert für Identifikation sind zu nah beieinander. Die Minstdifferenz ist 5 [%]
			16#000B	Der Eingang SetpointGradient ist ungültig. Wert muss größer 0 sein
			16#000C	Eingabewerte für Statusschranken ungültig (Bedingung: (Setpoint + Limits.UpperAtSetPoint) > (Setpoint - Limits.LowerAtSetPoint))
			16#000D	Zeit für Identifikation ist zu groß gewählt ("AutoTuneTimeOut")
			16#000E	Wert für "Quickness" ist nicht innerhalb gültiger Grenzen (1..5)
			16#000F	Wert für "ControlUpdateRate" ist nicht innerhalb gültiger Grenzen (1..1000)
SYSTEM_ERROR	F_RELATED_TABLE	16#1602	Abtastzeit	Jittern der Abtastzeit > 20 [%]. Fehler wird erst bei 2 aufeinander folgenden Zykluszeitüberschreitungen aktiv
			16#0005	Gradient des Istwertes zu groß, möglicherweise Sensorfehler (Diagnose)
			16#0006	Fehler im Regelkreis, Sensor möglicherweise verrutscht/zerstört

ErrorID	Table	Additional1	Additional2	Beschreibung
CALCULATION_ERROR	F_RELATED_TABLE	16#1603	16#0007	Identifizierte Streckenparameter ungültig, Streckenselbstidentifikation fehlgeschlagen
			16#0008	Das identifizierte Prozessmodell ist zu schlecht/ungültig
ACCESS_ERROR	F_RELATED_TABLE	16#1604	Feldnummer des Array	Fehlerhafter Zugriff auf interne Variable
OTHER_ERROR	F_RELATED_TABLE	16#1605	16#0009	Zeitüberlauf bei Identifikation (Watchdog)
DEVICE_ERROR	F_RELATED_TABLE	16#1606	Temperatur	Temperaturfehlerschranke überschritten, Temperatur außerhalb der Fehler-schranken

Abb.5-63: Fehlercodes Funktionsbaustein IL_TempControlType01

Hinweise zu bestimmten Fehlern

- Fehler**
"Gradient des Istwertes zu groß"
 (Additional1 = 16#1602; Additional2 = 16#0005):
 Hier wird der Gradient des Istwertes überwacht. Ist die Differenz zwischen zwei aufeinander folgenden Istwerten größer als 10u/ms, wird dieser Fehler generiert. Dies kann bei Einsatz eines Inline Temperaturmoduls der Fall sein, denn dort werden Fehlerdiagnosen als Messwert mit Werten größer gleich 32768 ausgegeben (Bit 15 gesetzt). Wird anwenderseitig diese Diagnose nicht ausgewertet, kann es zu dieser Fehlermeldung kommen.
- Fehler**
"Fehler im Regelkreis, Sensor möglicherweise verrutscht/zerstört"
 (Additional1 = 16#1602; Additional2 = 16#0006):
 Hier wird überwacht, ob bei maximaler Ausgangsleistung der Istwert sich dem Sollwert nähert. Dazu wird eine Zeitüberwachung gestartet, die bei aktivem I-Anteil

$$2000 / K_i + T_t \text{ und ohne I-Anteil}$$

$$2000 * \text{Abtastzeit} + T_t \text{ beträgt.}$$
 Bewegt sich der Istwert innerhalb dieser Zeit nicht in Richtung Sollwert, wird der Fehler generiert. Es besteht die Möglichkeit, dass der Sensor defekt/verrutscht ist, oder die Heizung/Kühlung nicht funktioniert. Auch kann der Fehler auftreten, wenn bei Vorgabe der Reglerparameter über die Struktur "IL_TEMPCONTROL_CONTR_PARAM" keine Totzeit (Tt) angegeben wird. In diesem Fall sollte für die Totzeit ein geschätzter Wert angegeben und die Reglerparameter neu gesetzt werden. Bei der Identifikation der Strecke erfolgt dies automatisch.
- Fehler**
"Identifizierte Streckenparameter ungültig, Streckenselbstidentifikation fehlgeschlagen"
 (Additional1 = 16#1603; Additional2 = 16#0007):
 Die Streckenparameter ergaben ungültige Werte. Dies kann sich durch falsche Wahl der Abtastzeit oder fehlerhaften Istwert (Sensor nicht an-

geschlossen) ergeben. Siehe auch ["Empfehlung für die Wahl der Abtastzeit " auf Seite 83.](#)

- Fehler

"Das identifizierte Prozessmodel ist zu schlecht/ungültig"

(Additional1 = 16#1603; Additional2 = 16#0008):

Bei der Streckenidentifizierung wird das errechnete Modell mit den gemessenen Werten verglichen. Der Vergleich ergab keine zufriedenstellende Übereinstimmung. Das Modell weicht um mehr als 20% von den Messwerten ab. Unter Umständen kann die Abtastzeit falsch gewählt sein, oder die Strecke weist kein aperiodisches Verhalten auf. Auch Störgrößen, die während der Identifikation auftreten, können dies verursachen.

- Fehler

"Zeitüberlauf bei Identifikation (Watchdog)"

(Additional1 = 16#1605; Additional2 = 16#0009):

Bei der Streckenidentifizierung wird gewartet bis der Istwert eingeschwungen ist, dann erfolgt ein Sollwertsprung und es wird erneut gewartet bis sich der Istwert eingeschwungen hat. Dauert dies zu lange, wird der Fehler generiert. Die Ursache liegt entweder in einem stark schwankenden Istwert, oder in einer falsch gewählten Abtastzeit (siehe auch nächster Absatz).

- Fehler **"Zeit für Identifikation ist zu groß gewählt ('AutoTuneTimeout)'"**

(Additional1 = 16#1601; Additional2 = 16#000D):

Bei der Streckenidentifizierung kann der Wert für die Zeitüberwachung eingestellt werden. Dieser Wert kann maximal so groß sein wie 10000 * Abtastzeit. In diesem Fall, entweder die Zeitüberwachung auf 0 setzen (maximal mögliche Zeit der Identifizierung bei gegebener Abtastzeit), oder die Abtastzeit entsprechend erhöhen.

Anwendung Der Baustein kann für IndraLogic XLC und IndraMotion MLC mit Hilfe von Inline Temperatur-Modulen (IL TEMP 2 RTD) betrieben werden. Die Inline-Module stehen zum gegenwärtigen Zeitpunkt noch nicht für die IndraMotion MLD zur Verfügung. Generell kann jedoch jeder beliebige Wert als Temperaturwert benutzt werden, da der Temperaturregler nicht direkt auf Hardware zugreift. Damit ist es auch möglich, Temperaturwerte über z.B. Feldbusverbindungen zu erhalten.

Identifikation Die automatische Identifikation, auch "AutoTune" genannt, dient der Findung der Parameter der Regelstrecke, um daraus die Parameter des Reglers festzulegen. Es werden Modellparameter berechnet, anschließend wird das Modell mit der realen Regelstrecke verglichen, um Fehlparametrierungen der Reglerparameter zu verhindern.

Bevor der Sprung aufgeschaltet wird, wird sichergestellt, dass sich die Regelstrecke in Ruhe befindet. Dazu wird eine Zeitüberwachung gestartet. Voreingestellter Zeitwert für diese ist 5000 multipliziert mit der Wurzel aus der Abtastzeit in [ms]. Vor Ablauf der Zeitüberwachung darf der Istwert die Toleranzgrenzen der definierten Toleranzschranken nicht überschreiten, ansonsten wird die Zeitüberwachung neu gestartet und wieder gewartet bis diese abgelaufen ist. Die Grenzschränke für das Temperaturfenster in dem der Istwert für die Zeit der Zeitüberwachung liegen muss, wird wie folgt berechnet:

$$|\text{IdentTemp}-\text{IstTemp}| < \sqrt{\left| \frac{\text{IdentTemp}}{10} \right|} + 0,5$$

Die Variable "IdentTemp" ist dabei der Wert um den das Toleranzfenster für die Isttemperatur aufgespannt wird. Das Warten auf den stationären Endwert erfolgt in gleicher Weise.

Die Güte des ermittelten Modells wird abgeschätzt. Wird dabei eine zu große Abweichung festgestellt, gibt der FB den Fehler

"16#1603" (CALCULATION_ERROR)

mit der Kennung "16#0008" (Prozessmodell ungültig) aus.

5.6.2 Regler

Einstellung der Reglerparameter

Gleichung der Strecke:

$$G_S(s) = \frac{K_S}{(1 + sT_{SUM})} e^{-sT_t} \approx \frac{K_S}{(1 + sT_t)(1 + sT_{SUM})}$$

Gleichung des Reglers:

$$G_R(p) = K_P \left(1 + \frac{1}{sT_N} + T_V s \right) = K_P + \frac{K_I}{s} + \frac{K_D s}{1 + sT_f}$$

Einstellung nach Betragsoptimum:

$$T_N = T_{SUM}$$

$$T_V = T_t$$

$$K_P = \frac{T_N}{a \cdot K_S \cdot T_t}$$

bzw.

$$K_I = \frac{K_P}{T_{SUM}}$$

$$K_D = T_t \cdot K_P$$

$$K_P = \frac{T_{SUM}}{\text{Quickness} \cdot K_S \cdot T_t}$$

Quickness = 1...5;

1 = sehr schnell mit großem Überschwingen,

5 = sehr langsam, kein Überschwingen;

Voreingestellter Wert ist Quickness = 3.

Einstellung der Filterzeit für den Differenzanteil:

$$T_f = T_V / 10$$

ML_TechBase.library

Dieser Wert wird fest eingestellt und kann nicht geändert werden. Die errechneten Reglerparameter werden in die dafür vorgesehene Struktur "IL_TEMPCONTROL_CONTR_PARAM" eingetragen.

Dabei entspricht der Eintrag:

KP dem proportionalem Verstärkungsfaktor
KI der Integrationskonstante
KD der Differentiationskonstante und
Tt der Totzeit

Werden diese Parameter durch den Anwender vorgegeben, sollte für die Totzeit zumindest ein Schätzwert eingetragen werden um den Fehler

"Fehler im Regelkreis, Sensor möglicherweise verrutscht/zerstört"

(Additional1 = 16#1602; Additional2 = 16#0006)

zu vermeiden.

Regelalgorithmus Es ist ein additiver $PIDT_1$ -Algorithmus mit Anti-Wind-Up des I-Anteils und PT_1 -Filterung des D-Anteils implementiert. Die additive Form ermöglicht das Abschalten einzelner Anteile des Reglers, so dass bei Bedarf beispielsweise auf die Struktur eines PI-Reglers umgeschaltet werden kann, indem der D-Anteil zu "Null" parametrisiert wird.

Die Begrenzung des I-Anteils erfolgt durch einen Anti-Wind-Up-Begrenzer, der intern auf einen Wert von 90% der Stellgröße voreingestellt ist.

Die Sättigungsgrenze des Reglers bildet der maximale Ausgangswert (VAR_IN OutputPowerMax) des Reglers. Voreingestellt ist ein Wert von 100%.

Reglerstruktur Dargestellt ist im Folgenden die $PIDT_1$ -Reglerstruktur mit Anti-Wind-Up-Begrenzung und Begrenzung der maximalen Stellgröße. Die additive Struktur ermöglicht ein selektives Abschalten einzelner Regleranteile. Somit können auch reine P-, PI- oder PDT_1 -Regler parametrisiert werden. Diese Abschaltung der einzelnen Anteile muss jedoch vom Anwender selbst durchgeführt werden. Dazu müssen in die Struktur "contrParam" die gewünschten Werte eingestellt werden. Mit einer steigenden Flanke am Eingang "Preset" werden diese neuen Parameter dann im Regler aktiv geschaltet. Standardmäßig arbeitet der Regler mit der kompletten $PIDT_1$ -Struktur.

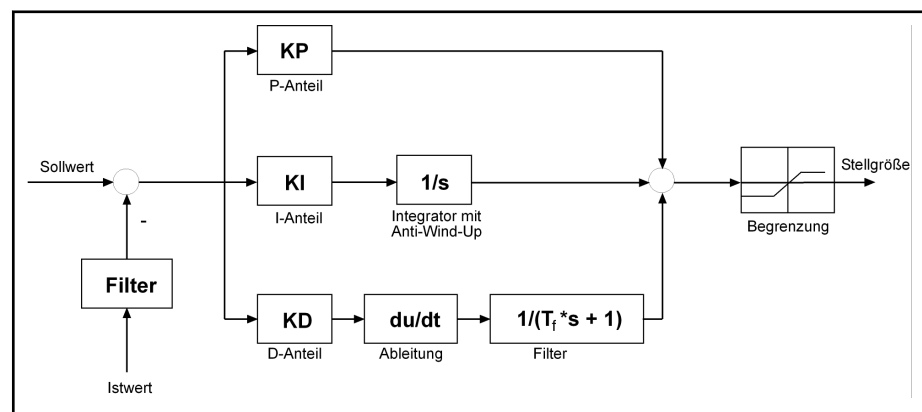


Abb.5-64: Reglerstruktur

Der Istwert der Temperatur wird sowohl bei der Identifikation als auch beim Regeln mit einem PT_1 -Filter geglättet um stochastische Störungen zu vermindern. Die Zeitkonstante des Filters ist fest auf das 20fache der Zykluszeit eingestellt.

Regelkreisüberwachung

Der Funktionsbaustein verfügt über eine Regelkreisüberwachung. Bei Ausgabe einer Stellgröße von "100%" (volle Leistung) muss sich die Temperatur in Richtung des in K_P eingestellten Vorzeichens ändern. Ansonsten wird ein Timer aktiv, der bei Ablauf einen Fehler ausgibt. Der Timer wird zurückgesetzt, sobald sich die Temperatur wieder in die richtige Richtung bewegt. Der Timer wird folgendermaßen mit Werten belegt:

Bei vorhandenem I-Anteil wird ein Wert von $(2000 / K_I) + \text{Totzeit}$ verwendet. Wird kein I-Anteil verwendet, wird lediglich der Wert $2000 \cdot \text{Abtastzeit}$ voreingestellt. Auch wenn die Stellgröße wieder kleiner "100%" wird, wird der Timer zurückgesetzt. Es ist wichtig, die Totzeit (T_t) in der Struktur für die Reglerparameter mit anzugeben, wenn die Reglerparameter durch den "Preset" Eingang vorgegeben werden. Bei großem K_I und großer Totzeit könnte sonst die Überwachung ansprechen. Durch diese Überwachung können Fehler am Sensor (z.B. Verrutschen) oder Fehler am Stellglied erkannt werden.

Zusätzlich verfügt der FB noch über eine Gradientenüberwachung des Istwertes, der aber nur bei sehr großen Abweichungen anschlägt. Solche können z.B. vorliegen, wenn der Sensor einen Fehlercode ausgibt und dieser Fehlercode vom Anwender nicht abgefangen/ausgewertet wird. Der Gradient ist fest auf 10 u/ms eingestellt. Es wird eine einmalige Überschreitung toleriert, ehe ein Fehler generiert wird.

5.6.3 Verwendung des Funktionsbausteins

Der Funktionsbaustein sollte in einer zyklischen Task betrieben werden. Der FB erkennt die Taskintervallzeit (= Zykluszeit) automatisch und passt den Regelalgorithmus daran an. Der Regler arbeitet mit der Abtastzeit, die sich aus der Taskintervallzeit multipliziert mit der Eingangsvariable "ControlUpdateRate" ergibt. Wenn "ControlUpdateRate" = 1, entspricht die Abtastzeit somit der Taskintervallzeit. Die Abtastzeit findet Verwendung bei der Identifizierung der Strecke, sowie bei eingeschaltetem Regler. Alle anderen Funktionen des Bausteins (Istwerte lesen, Gradientenüberwachung, PWM Ausgabe, Manuelle Regelung) arbeiten mit der Taskintervallzeit. Dadurch ist es möglich, den Regelungs-/Identifizierungsprozess zeitlich von anderen Funktionen des Bausteins zu entkoppeln. Dies ist besonders wichtig, um die Ausgabe der Stellgröße mittels PWM vom Regelungs-/Identifizierungsprozess abzukoppeln.

Die Taskintervallzeit bildet die Grundlage für die Pulsweitenmodulation (PWM), d.h. die Taskintervallzeit ist die kleinste mögliche Zeitscheibe, die bei der PWM auf "An" bzw. "Aus" gesetzt werden kann. Eine Periode der PWM dauert 20-mal länger als die Taskintervallzeit. Damit kann die Stellgröße zwischen 0% und 100% in 5% Schritten variiert werden.

Beispiel:

Die Taskintervallzeit wird für die zyklische Task auf 20 ms parametrieret. Das heißt, dass alle 400 ms ein neuer Wert für die Stellgröße an die PWM übergeben werden kann. Mögliche Ein-/Ausschaltzeiten für die PWM sind somit:

Stellgröße [%]	Einschaltdauer [ms]	Ausschaltdauerzeit [ms]
0	0	400
5	20	380
10	40	360
...		

ML_TechBase.library

Stellgröße [%]	Einschaltdauer [ms]	Ausschaltdauerzeit [ms]
50	200	200
...		
95	380	20
100	400	0

Abb.5-65: Beispiel Ein-/Ausschaltzeiten

Die Wahl der Taskintervallzeit und "ControlUpdateRate" muss stets in Abhängigkeit von Stellglied und Regelstrecke getroffen werden. Zu kleine Taskintervallzeiten verschleifen das Stellglied sehr schnell (Bsp. Relaismodule), zu große Taskintervallzeiten/Abtastzeiten vermindern die Genauigkeit der Regelung.

Weiterhin ist eine Überwachung der Abtastzeit vorhanden, die bei Über-/Unterschreiten der Abtastzeit um mehr als 20% einen Fehler erzeugt und den Ausgang "Error" auf TRUE setzt.

Empfehlung für die Wahl der Taskintervallzeit

Die Taskintervallzeit/Zykluszeit sollte so eingestellt werden, dass die PWM keine sichtbare Temperaturänderungen innerhalb einer PWM-Periode von $20 \cdot \text{Taskintervallzeit}$ hervorruft. Das heißt, wenn bei einer Stellgröße von z.B. 20% die Temperatur der Strecke während des Heizens deutlich zunimmt und während dem ausgeschalteten Zustand der Heizung die Temperatur wieder abnimmt, ist die Taskintervallzeit zu groß gewählt. Dies ist im folgenden Bild zu erkennen, wobei die obere Kurve den Temperaturverlauf angibt, die untere Kurve zeigt die Heizphasen an (TRUE). Es ist klar ersichtlich, dass mit jedem Heizen die Strecke sichtbar erwärmt wird und, während die Heizung ausgeschaltet ist, die Strecke schon wieder abkühlt.

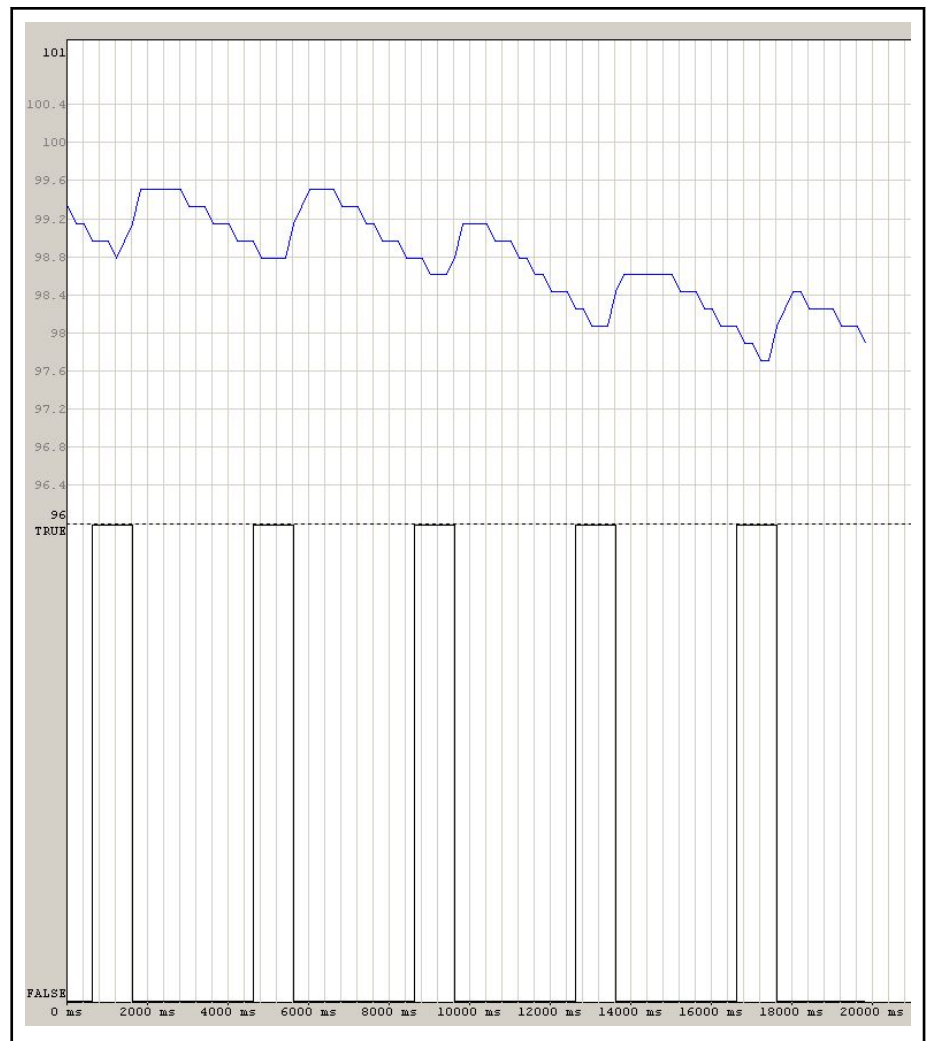


Abb.5-66: Taskintervallzeit

Umgekehrt ist es auch nicht sinnvoll die Taskintervallzeit zu klein zu wählen, da das Stellglied dann erhöhten Belastungen ausgesetzt ist (z.B. Relaismodul muss sehr oft schalten).

Eine sinnvolle Taskintervallzeit muss durch Tests ermittelt werden.

Empfehlung für die Wahl der Abtastzeit

Zunächst sollte die Wahl der Taskintervallzeit, wie im vorgehenden Absatz beschrieben, gewählt werden. Dann kann man mittels des Eingangs "ControlUpdateRate" die Abtastzeit auf den gewünschten Wert festlegen.

Eine Empfehlung für die Abtastzeit ist:

- 10 ms für schnelle/dynamische Strecken (Strecken Zeitkonstante von 200ms bis 2s)
- 100 ms für mittlere Strecken (Strecken Zeitkonstante von 2s bis 20s)
- 1s bis 30s für langsame/träge Strecken (Strecken Zeitkonstante von 20s bis 100min)

Die Abtastzeit sollte etwa 20 bis 200-mal schneller sein als die "dominierende Zeitkonstante" der Regelstrecke. Die "dominierende Zeitkonstante" ist in etwa die Zeit, die eine Regelstrecke benötigt um 63% der Änderung eines aufgeschalteten Sprunges zu durchlaufen. Prinzipiell ist die Abtastzeit jedoch vom Anwender komplett frei wählbar.

ML_TechBase.library

Bei zu kleiner Wahl der Abtastzeit kann bei der Identifikation der Fehler "Zeitüberlauf bei Identifikation (Watchdog)" (Additional1 = 16#1605; Additional2 = 16#0009) auftreten. In diesem Fall ist eine größere Abtastzeit zu wählen, da bei Ablauf der Zeitüberwachung die Regelstrecke noch nicht eingeschwungen war.

Bei zu großer Wahl der Abtastzeit kann bei der Identifikation der Fehler "Identifizierte Streckenparameter ungültig, Streckenselbstidentifikation fehlgeschlagen" (Additional1 = 16#1603; Additional2 = 16#0007) auftreten. In diesem Fall ist eine kleinere Abtastzeit zu wählen, da das Streckenverhalten in zu großen zeitlichen Abständen gemessen wurde.

Empfehlung für die Identifikation

Für die Identifikation sind einige Regeln notwendig um gute Ergebnisse zu erzielen. Diese sollen hier näher erläutert werden.

Temperaturstrecken sind in der Regel nichtlinear. Bei kleinen Temperaturen muss weniger Leistung für eine bestimmte Temperaturerhöhung aufgebracht werden als bei größerer Temperatur. Ein Regler benötigt bei niedrigen Temperaturen andere Einstellungen als bei hohen Temperaturen.

Deshalb ist es besonders wichtig, die Identifikation auch in dem Temperaturbereich durchzuführen, in dem der Regler später arbeiten soll. Dabei hat man zwei Möglichkeiten.

1. Bei sehr genauer Regelung in einem kleinen Temperaturbereich sollte zunächst die im Handbetrieb benötigte Leistung so vorgegeben werden, dass sich die gewünschte Temperatur ergibt. Anschließend kann man dann eine Identifikation durchführen, bei der der Wert für "AutoTuneStartPower" etwas niedriger als der ermittelte Wert liegt und der Wert für "AutoTuneFinishPower" etwas höher als der ermittelte Wert liegt. Hier werden Reglerparameter ermittelt, die genau für diesen Temperaturbereich bestimmt sind. Außerhalb des Temperaturbereiches wird die Regelung dagegen ungenauer sein.
2. Bei einem breiten Regelungsbereich sollte im Handbetrieb jeweils die Leistung für die niedrigste und höchste Temperatur ermittelt werden. Diese dienen dann als Werte für "AutoTuneStartPower" und "AutoTuneFinishPower". Die Regelung wird dabei über den gesamten Temperaturbereich gemittelt. Das bedeutet in der Regel bei kleinen Temperaturen eine etwas dynamischere Regelung und bei höheren Temperaturen eine etwas gedämpftere Regelung.

5.6.4 Beschreibung der einzelnen Funktionen, Beispiele

Für die im Folgenden vorgestellten Anwendungsbeispiele sind im Programm des Anwenders folgende Variablen im Deklarationsteil für die Belegung des Funktionsbausteines zu vereinbaren.

Programm:

```

VAR
fb_TempControl:IL_TempControlType01;      (* Instanz des FB's *)
strContrParam: IL_TEMPCONTROL_CONTR_PARAM; (* Struktur, welche die Reglerparameter enthält *)
strLimits: IL_TEMPCONTROL_LIMITS;          (* Alarm-/Statusschranken *)
rActualTemp:REAL;                          (* Isttemperatur *)
rActualPower:REAL;                        (* Stellgröße, analog *)
bActualPower:BOOL;                        (* Stellgröße, digital (PWM) *)
bAutoTuneDone:BOOL;                      (* Zustand der Identifikation *)
bSetPointReached:BOOL;                  (* Istwert liegt im Sollwert Fenster *)
bError:BOOL;                            (* Zustand des Fehlerausgangs *)
iState:INT;                             (* Wert in Schrittfolge *)
END_VAR

```

In den nachfolgenden Beispielen wurde der Übersicht halber auf eine Fehlerbetrachtung verzichtet.

Einfache Regelung mit Reglerparametervorgabe

Der Funktionsbaustein wird mit den gewünschten Reglerparametern in der Struktur "strContrParam" parametrisiert. Mit einer steigenden Flanke am Eingang "Preset" können neue Reglerparameter auch während der Regelung parametrisiert werden. Der Sollwert ("Setpoint") muss vorgegeben, der aktuelle Istwert ("Feedback") muss zurückgeführt werden. Die Stellgröße kann entweder am Analogausgang ("ControlValueAnalog") oder am Digitalausgang ("ControlValuePWM") abgeholt werden.

Programm:

```
(* Setzen der Reglerparameter *)
strContrParam.KP := 1.5;
strContrParam.KI := 2;
strContrParam.KD := 0.6;
strContrParam.Tt := T#1s;

(* Starten des Bausteins bei gleichzeitiger Übernahme *)
(* der Reglerparameter *)
fb_TempControl( Enable:=TRUE,
                Setpoint:=50,
                Feedback:=rActualTemp,
                Preset:=TRUE,
                ContrParam:=strContrParam,
                ControlValueAnalog=>rActualPower,
                ControlValuePWM=>bActualPower);
```

Damit werden die Reglerparameter einmalig übernommen und der Regler arbeitet mit den vorgegebenen Werten.

Einfache Identifikation einer Regelstrecke mit anschließendem Übergang in eine Regelung

Der Funktionsbaustein wird zuerst in den Identifikations-/AutoTune-Modus geschaltet. Nach erfolgreichem Abschluss der Identifikation wird die Regelung gestartet, indem am Eingang "StartAutoTune" eine fallende Flanke generiert wird.

Die berechneten Reglerparameter werden in der Struktur "strContrParam" abgespeichert und stehen somit dem Anwender zur Verfügung. Diese Reglerparameter können somit vom Anwender in den Retainbereich der Steuerung gesichert werden (siehe Beispiel ["Retainsicherung der Reglerparameter" auf Seite 89](#)). Damit ist bei einem erneuten Anlauf der Regelung keine Identifikation nötig, es kann sofort die einfache Regelung mit Parametervorgabe gestartet werden.

Mit dem optionalen Eingang "Quickness" wurde eine etwas schnellere Einstellung des Reglers vorgegeben (schnelleres Anregeln, dafür aber höhere Neigung zum Überschwingen).

Die Identifikation wird mit einem Sprung von 0% ("AutoTuneStartPower") auf 10% ("AutoTuneFinishPower") Ausgangsleistung vorgenommen.

Programm:

```
CASE iState OF
1: (* Identifikation *)
    fb_TempControl ( Enable:=TRUE,
                    StartAutoTune := TRUE
                    Feedback:=rActualTemp,
                    AutoTuneStartPower:= 0,
                    AutoTuneFinishPower:= 10,
                    Quickness:=2,
                    ContrParam:=strContrParam,
                    ControlValueAnalog=>rActualPower,
                    ControlValuePWM=>bActualPower,
                    AutoTuneDone=>bAutoTuneDone);
    IF (bAutoTuneDone = TRUE) THEN
        iState := 2;
    END_IF

2: (* Übergang in Regelung *)
    fb_TempControl( Enable:=TRUE,
```

ML_TechBase.library

```

StartAutoTune := FALSE,
Setpoint:=50,
Feedback:=rActualTemp,
ContrParam:=strContrParam,
ControlValueAnalog=>rActualPower,
ControlValuePWM=>bActualPower);

```

Identifikation der Regelstrecke im Arbeitspunkt

Während einer Prozessregelung kann eine erneute Identifikation im Arbeitspunkt gewünscht sein. Ausgehend vom vorgehenden Beispiel "Einfache Identifikation einer Regelstrecke mit anschließendem Übergang in eine Regelung" wird hier nun beschrieben, wie für diesen Fall vorzugehen ist. Mit einem Setzen des Eingangs "AutoTuneStart" wird die Identifikation gestartet, die Anfangs- und Endparameter der Stellgrößenparameter werden in "AutoTuneStartPower" und "AutoTuneFinishPower" übergeben. Die neuen Reglerparameter stehen dem Anwender nach erfolgreicher Identifikation wieder in der Struktur "strContrParam" zur Verfügung.

Im Beispiel wird der Arbeitsbereich, ausgehend von dem aktuellen Wert der Stellgröße (in %) berechnet. Voraussetzung ist natürlich, dass sich die Strecke im eingeschwungenen (stationären) Zustand befindet. Der Arbeitsbereich wird von "Stellgröße-10%" bis "Stellgröße+10%" im Beispiel gewählt. Die Identifikation im Arbeitspunkt startet mit Wechsel in den State "3".

Programm:

```

CASE iState OF
2: (* Regelung *)
    fb_TempControl( Enable:=TRUE,
                    StartAutoTune := FALSE,
                    Setpoint:=50,
                    Feedback:=rActualTemp,
                    ContrParam:=strContrParam,
                    ControlValueAnalog=>rActualPower);

    If (bIdent = TRUE) THEN
        iState := 3;
    End_IF;

3: (* Identifikation im Arbeitspunkt *)
    fb_TempControl( Enable:=TRUE,
                    StartAutoTune := TRUE,
                    AutoTuneStartPower:=rActualPower-10,
                    AutoTuneFinishPower:=rActualPower+10,
                    Feedback:=rActualTemp,
                    ContrParam:=strContrParam,
                    ControlValueAnalog=>rActualPower,
                    AutoTuneDone=>bAutoTuneDone);

    IF (bAutoTuneDone = TRUE) THEN
        iState := 2;
    END_IF

```

Definition einer Anlauframpe

Erfordert ein Prozess vor der eigentlichen Temperaturregelung das Verhalten an einer bestimmten Temperatur für eine gewisse Zeit (Anlauframpe), so ist diese Funktion über die Eingänge "StartUpTemp" und "StartUpTime" verfügbar.

Der Regler regelt zuerst bis zur Anlauftemperatur. Wenn diese erreicht wird, läuft die "StartUpTime" ab und der Regler hält die Anlauftemperatur konstant. Nach Ablauf der "StartUpTime" ist die Anlauframpe abgeschlossen und der Regler verhält sich in gewohnter Weise. Die Anlauframpe wird nach jedem Wechsel von "Enable" auf "TRUE" einmalig ausgeführt.

Im Beispiel wird zuerst auf eine Temperatur von "100" geregelt, dann 5 Minuten und 30 Sekunden gewartet und danach erst auf die gewünschte Temperatur von "150" geregelt (siehe [Abb. 5-68 "Verwendung eines Sollwertgradienten und Anlauframpe" auf Seite 89](#)).

Programm:

```
fb_TempControl(    Enable:=TRUE,
                  Setpoint:=150,
                  Feedback:=rActualTemp,
                  Preset:=TRUE,
                  StartUpTemp:=100,
                  StartUpTime:=t#5m30s,
                  ContrParam:=strContrParam,
                  ControlValueAnalog=>rActualPower);
```

Definition von Alarm- und Statusschranken

Zur Prozessüberwachung können Alarm- und Statusschranken angelegt werden.

Die "Limits" Struktur beinhaltet die Werte "UpperAlarm" und "LowerAlarm". Bei Verletzung der Alarmschranken wird ein Fehler generiert. Die Überwachung wird erst aktiv, wenn sich die aktuelle Prozesstemperatur im Bereich zwischen "UpperAlarm" und "LowerAlarm" befindet, um während des Prozessanlaufs keine Fehler zu erzeugen. Bei den Werten handelt es sich um absolute Werte.

Die Limits Struktur beinhaltet außerdem die Werte "UpperAtSetPoint" und "LowerAtSetPoint". Der Ausgang "SetPointReached" wird gesetzt sobald die Bedingung $(\text{Setpoint} - \text{Limits.LowerAtSetPoint}) \leq \text{Istwert} \leq (\text{Setpoint} + \text{Limits.UpperAtSetPoint})$ erfüllt ist. Die Werte sind relative Werte die sich auf den "Setpoint" beziehen. Somit kann der "Setpoint" geändert werden, ohne "Limits.LowerAtSetPoint" und "Limits.UpperAtSetPoint" ändern zu müssen. Anhand dieser Statusanzeige kann man erkennen, ob die gewünschte Temperatur bereits erreicht wurde.

Programm:

```
strLimits.UpperAlarm:=250;
strLimits.LowerAlarm:=70;
strLimits.UpperAtSetpoint:=20;
strLimits.LowerAtSetPoint:=20;

fb_TempControl(    Enable:=TRUE,
                  Setpoint:=150,
                  Feedback:=rActualTemp,
                  Limits:=strLimits,
                  ContrParam:=strContrParam,
                  ControlValueAnalog=>rActualPower,
                  SetPointReached=>bSetPointReached);
```

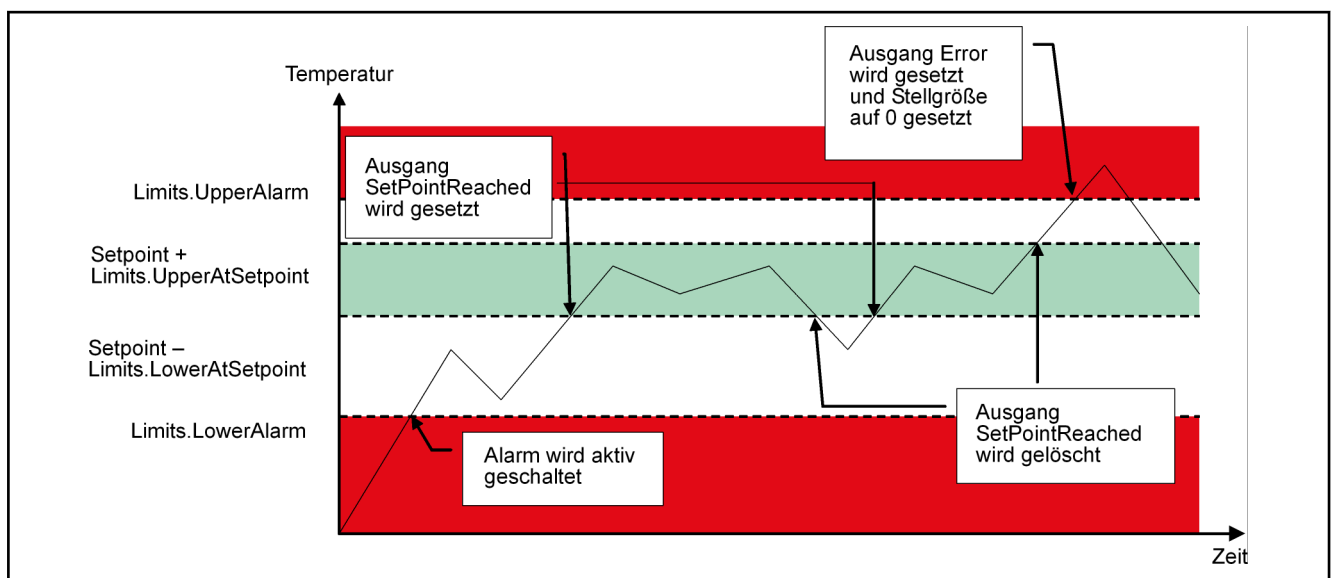


Abb.5-67: Wirkungsweise der Alarm- und Statusschranken

ML_TechBase.library

Anhalten der Regelung (PAUSE-Modus)

Ein Anhalten der Regelung kann über den Eingang "Pause" realisiert werden. Dabei wird die Stellgröße konstant gehalten und es wird der Regler mit all seinen Werten "eingefroren". Nach dem Rücksetzen des Eingangs "Pause" läuft die Regelung weiter.

Im Beispiel wird in Zustand "4" die Temperatur auf "150" geregelt. Mit Umschalten auf Zustand 5 wird die Regelung "eingefroren", es werden keine neuen Stellgrößen ausgegeben. Mit Rückkehr auf Zustand "4" läuft die Regelung weiter, neue Stellgrößen werden wieder ausgegeben. Es kann dabei zu einem Sprung kommen, wenn sich zwischenzeitlich der Istwert stark verändert hat.

Programm:

```

CASE iState OF
4:
fb_TempControl(      Enable:=TRUE,
                     Pause:=FALSE,
                     Setpoint:=150,
                     Feedback:=rActualTemp,
                     Preset:=TRUE,
                     ContrParam:=strContrParam,
                     Limits:=strLimits,
                     ControlValueAnalog=>rActualPower);

5:
fb_TempControl(      Enable:=TRUE,
                     Pause:=TRUE,
                     Setpoint:=150,
                     Feedback:=rActualTemp,
                     Preset:=TRUE,
                     ContrParam:=strContrParam,
                     Limits:=strLimits,
                     ControlValueAnalog=>rActualPower);

```

Analoger/Digitaler Stellgrößenausgang

Der Funktionsbaustein verfügt über zwei Stellgrößenausgänge, einen Analogen ("ControlValueAnalog") an welchem REAL-Werte zwischen "0%" (keine Leistung) und "100%" (volle Leistung) ausgegeben werden und einen Digitalen ("ControlValuePWM") an welchem die BOOL-Werte "FALSE" (keine Leistung) oder "TRUE" (volle Leistung) ausgegeben werden. Der digitale Ausgang kann bei Benutzung von Relaismodulen zur Pulsung der Leistung genutzt werden. Die Pulsung erfolgt über eine Pulsweitenmodulation, deren Zykluszeit 20-mal langsamer ist als die zyklische Task in der der FB läuft. Somit können Stellgrößen mit einer Auflösung von „0%“ bis „100%“ in „5%“ Schritten moduliert werden.

Der analoge Ausgang kann für eine Analogausgabe oder zur Überwachung der aktuellen Stellgröße verwendet werden.

Programm:

```

fb_TempControl(      Enable:=TRUE,
                     Pause:=FALSE,
                     Setpoint:=150,
                     Feedback:=rActualTemp,
                     Preset:=TRUE,
                     ContrParam:=strContrParam,
                     Limits:=strLimits,
                     ControlValueAnalog=>rActualPower,
                     ControlValuePWM=>bActualPower);

```

Vorgabe eines Sollwertgradienten

Der Funktionsbaustein bietet die Möglichkeit den Anstieg des Sollwertes zu begrenzen (zu verrampen). Dies vermindert die Überschwingweite und schont die Anlage. Dabei wird bei jeder neuen Vorgabe des Eingangs "Set-Point" der Sollwert intern rampenförmig (nicht sprungförmig) solange verändert, bis die gewünschte Sollgröße erreicht ist. Die Eingabe von "SetPoint-Gradient" erfolgt in [u/s].

Im Beispiel wird die Solltemperatur auf "150" gefahren mit einem Anstieg von "0,5 je Sekunde".

Programm:

```
fb_TempControl( Enable:=TRUE,
                Pause:=FALSE,
                Setpoint:=150,
                Setpointgradient:=0.5,
                Feedback:=rActualTemp,
                Preset:=TRUE,
                ContrParam:=strContrParam,
                Limits:=strLimits,
                ControlValueAnalog=>rActualPower);
```

Eine Verrampung mit Sollwertgradient und zusätzlicher Anlauframpe hätte dann den, wie folgt gezeigten, zeitlichen Verlauf von Sollwert und Istwert. Hier wurde auf eine Solltemperatur ("Setpoint:=200") von "200°C" gefahren mit einem Gradienten ("Setpointgradient:=1") von "1°C/s". Die Anlauftemperatur ("StartUpTemp:=100") beträgt "100°C", die Anlaufzeit ("StartUpTime:=T#50s") "50s".

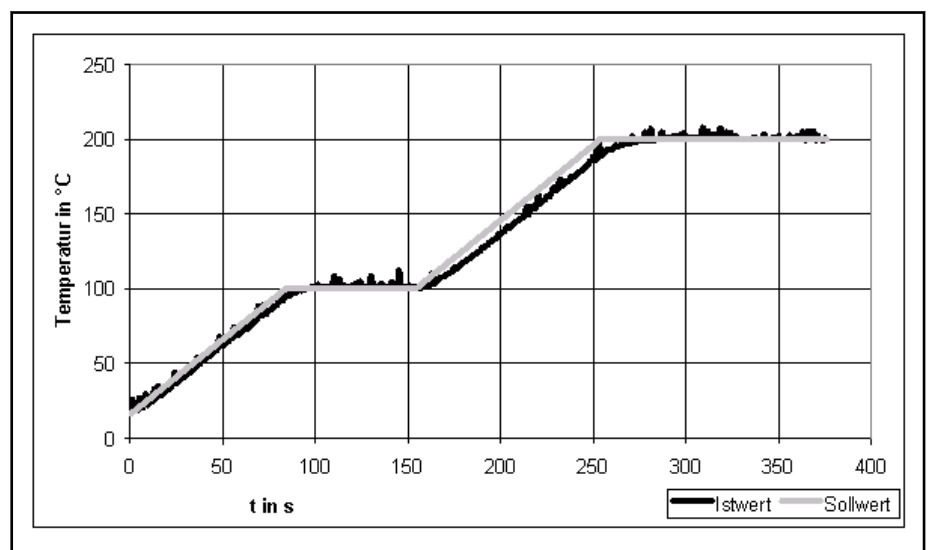


Abb.5-68: Verwendung eines Sollwertgradienten und Anlauframpe

Vorgabe einer manuellen Stellgröße

Um die Temperatur manuell zu steuern, lässt sich eine Handstellgröße festlegen. Diese stellt gleichzeitig die Stellgröße am Ausgang des Funktionsbausteins dar. Eine Handstellgröße kann benötigt werden, um manuelle Testfunktionen aufzuzeichnen oder bei Sensorfehler in einen Notbetrieb übergehen zu können.

Im Beispiel wird der FB manuell mit einer Stellgröße von "50%" angesteuert.

Programm:

```
fb_TempControl( Enable:=TRUE,
                Pause:=FALSE,
                OperatingMode:=TRUE,
                ManContrValue:=50,
                ContrParam:=strContrParam,
                Limits:=strLimits,
                ControlValueAnalog=>rActualPower);
```

Retainsicherung der Reglerparameter

Um die Reglerparameter spannungsausfallsicher zu speichern, bietet es sich an, diese in den Retainbereich zu sichern. Somit kann beim Wiedereinschalten des Regler die Identifikation eingespart und sofort mit der Regelung begonnen werden.

Dazu muss vom Anwender die Struktur "strContrParam" als Retainvariablen ("VAR RETAIN") angelegt werden, die dann die Werte der einzelnen Reglerparameter beinhaltet. Gleichzeitig muss ein Mechanismus geschaffen wer-

ML_TechBase.library

den, um beim Anlaufen zu unterscheiden, ob schon einmal eine Identifikation durchgeführt wurde.

Im Header müssen zusätzlich Retainvariablen angelegt werden.

Programm:

```

VAR RETAIN
  strContrParam : STRUCT_CONTR_PARAM;
  bIdentification: BOOL := FALSE;
END_VAR

If (bIdentification = FALSE) THEN
  fb_TempControl(
    Enable:=TRUE,
    StartAutoTune := TRUE,
    Feedback:=rActualTemp,
    Quickness:=2,
    ContrParam:=strContrParam,
    ControlValueAnalog=>rActualPower,
    ControlValuePWM=>bActualPower,
    AutoTuneDone=>bAutoTuneDone);

  IF (bAutoTuneDone = TRUE) THEN
    bIdentification:= TRUE;
  END_IF
ELSE
  fb_TempControl(
    Enable:=TRUE,
    StartAutoTune := FALSE
    Preset:=TRUE,
    Setpoint:=150,
    Feedback:=rActualTemp,
    ContrParam:=strContrParam,
    Limits:=strLimits,
    ControlValueAnalog=>rActualPower);
END_IF;

```

Die Werte des Reglers werden im obigen Beispiel in jedem Zyklus in die Retainvariablen gesichert. Nach einem Abschalten/Netzausfall prüft der Code anhand der booleschen Variable "bIdentification" ob bereits eine Identifizierung stattfand. Ist dies nicht der Fall wird eine Identifizierung durchgeführt, ansonsten kann der Regler mit den gesicherten (alten) Parametern neu gestartet werden. Der Eingang "Preset" muss dabei mit einer steigenden Flanke belegt werden.

Fehlerbehandlung mittels Ersatzwertstrategie

Im Fehlerfall wird vom Funktionsbaustein standardmäßig die Stellgröße "0%" (keine Leistung) ausgegeben. Dies wird aus Gründen der Anlagensicherheit getan. Wird vom Anwender eine Steuerung auf eine bestimmte Leistung gewünscht, um beispielsweise ein starkes Abkühlen einer Anlage zu verhindern, empfiehlt es sich, die im folgenden Beispiel gezeigt Ersatzwertstrategie zu verwenden.

Im Zustand "6" wird mit "bError" auf Fehler geprüft. Falls diese Bedingung "TRUE" wird, wird der Funktionsbaustein zurückgesetzt "Enable:=FALSE" und in Zustand "7" übergegangen.

Im Zustand "7" wird der Funktionsbaustein mit einer steigenden Flanke am Eingang "Enable" neu gestartet und im manuellen Modus (siehe ["Vorgabe einer manuellen Stellgröße" auf Seite 89](#)) betrieben.

Im Beispiel wird eine Stellgröße von "50%" ausgegeben.

Nähere Informationen zum Fehler werden an den Ausgängen "ErrorID" und "ErrorIdent" ausgegeben, die jedoch in diesem Beispiel nicht ausgewertet wurden.

Programm:

```

CASE iState OF
6:
  fb_TempControl(
    Enable:=TRUE,
    Setpoint:=50,
    Feedback:=rActualTemp,
    Preset:=TRUE,

```

```

ContrParam:=strContrParam,
Limits:=strLimits,
ControlValueAnalog=>rActualPower,
Error=>bError);

IF (bError = TRUE) THEN
  fb_TempControl( Enable:=FALSE,
                  ContrParam:=strContrParam,
                  Limits:=strLimits);

  iState:=7;
END_IF

7:
fb_TempControl( Enable:=TRUE,
                Pause:=FALSE,
                OperatingMode:=TRUE,
                ManContrValue:=50,
                ContrParam:=strContrParam,
                Limits:=strLimits,
                ControlValueAnalog=>rActualPower);

```

Anzumerken ist hierbei noch, dass der Stellgrößenausgang im obigen Beispiel für einen Zyklus auf 0 gesetzt wird (wenn der Fehler erkannt und Enable:=FALSE gesetzt wird). Im Zustand 7 wird dann der manuelle Wert als Stellgröße verwendet.

5.7 Messtasterfunktionsbausteine

5.7.1 Einleitung und Überblick

Die Funktionsbausteine MB_InitTouchProbe, MB_TouchProbe, MB_WriteExpectWindow und MB_TouchProbeContinuous steuern und verwalten die von den Bosch Rexroth IndraDrive Antrieben unterstützte Messtasterfunktionalität.

Die Messtasterfunktionalität der Steuerungshardware wird über die folgenden Funktionsbausteine der ML_PLCOpen-Bibliothek bereitgestellt:

Funktionsbaustein	Beschreibung
ML_InitTouchProbe	Aktiviert und konfiguriert den Messtaster in der Steuerung
ML_WriteExpectWindow	Konfiguriert das Erwartungsfenster für die Messtasterfunktionalität
ML_TouchProbe	Aktiviert den ausgewählten Messtaster, wertet den Status aus und stellt die Messwerte nach dem Triggerereignis zur Verfügung
ML_TouchProbeContinuous	Aktiviert den ausgewählten Messtaster, wertet den Status aus und stellt die Messwerte nach dem Triggerereignis zur Verfügung
ML_AbortTrigger	Bricht eine aktive Messung ab

Abb.5-69: Funktionsbausteine zur Messtasterfunktionalität der Steuerungshardware

Die Messtasterfunktionalität der Rexroth IndraDrive Antriebe wird über die folgenden Funktionsbausteine der ML_TechBase-Bibliothek bereitgestellt:

ML_TechBase.library

Funktionsbaustein	Beschreibung
MB_InitTouchProbe	Aktiviert und konfiguriert den Messtaster im Antrieb
MB_WriteExpectWindow	Konfiguriert das Erwartungsfenster für die Messtasterfunktionalität
MB_TouchProbe	Aktiviert den ausgewählten Messtaster, wertet den Status aus und stellt die Messwerte nach dem Triggerereignis einmalig zur Verfügung
MB_TouchProbeContinuous	Aktiviert den ausgewählten Messtaster, wertet den Status aus und stellt die Messwerte nach dem Triggerereignis fortlaufend zur Verfügung
MB_AbortTrigger	Bricht eine aktive Messung ab

Abb.5-70: Funktionsbausteine zur Messtasterfunktionalität des Rexroth IndraDrive

Die folgende Dokumentation befasst sich mit den Funktionsbausteinen zur Messtasterfunktionalität des Rexroth IndraDrive. Die Funktionsbausteine zur Messtasterfunktionalität der Steuerung werden in den Dokumentationen "Rexroth IndraMotion MLC 12VRS Funktionsbeschreibung", "Rexroth IndraLogic XLC 12VRS Funktionsbeschreibung" und "Rexroth IndraMotion MLC IndraMotion IndraLogic XLC PLCopen Funktionsbausteine und Datentypen" beschrieben.

Die Messtasterfunktionalität des IndraDrives erlaubt das Erfassen der Positionen oder der internen Systemzeit in [us] bei der Instanz des schnellen, digitalen Eingangssignals. Der Messtasterfunktionsbaustein verwendet diese Antriebsfunktionalität und stellt eine Benutzerschnittstelle bereit. Die folgenden Schritte sind notwendig um die Messtasterfunktionsbausteine zu verwenden:

5.7.2 Messtasterfunktionsbausteine – Komponenten und Parametrierung

Hardware

Die folgenden Bosch Rexroth Hardwarekomponenten werden benötigt:

- IndraMotion MLC/IndraLogic XLC
- IndraDrive C oder IndraDrive M mit Advanced oder Basic Performance

Die folgenden Steuerungen werden unterstützt:

- ADVANCED (Typenschlüssel: CSB01.XC-...)
- BASIC SERCOS (Einzelachse; Typenschlüssel: CSB01.XN-SE-...)
- BASIC PROFIBUS (Einzelachse; Typenschlüssel: CSB01.XN-PB-...)
- BASIC UNIVERSAL (Einzelachse; Typenschlüssel: CSB01.XC-...)

Firmware

Die folgende Firmware wird benötigt und mit den oben genannten Hardwarekomponenten von Bosch Rexroth verwendet:

- Steuerungsfirmware 12VRS
- Antriebsfirmware MPx07VRS oder höher
- Ein Antriebsfunktionspaket, das die Messtasterfunktion beinhaltet

Parametrierung der Messtasterfunktionsbausteine für IndraMotion MLC/IndraLogic XLC und IndraMotion MLD

IndraMotion MLC/IndraLogic XLC-Konfiguration

Die folgenden Antriebsparametrierschritte sind notwendig um die Technologiefunktionsbausteine MB_InitTouchProbe, MB_TouchProbe und MB_TouchProbeContinuous zu verwenden:

Die folgenden Antriebsparametereinstellungen sind notwendig, wenn die Messtasterfunktionalität in der IndraMotion MLC/IndraLogic XLC verwendet wird:

Öffnen Sie IndraWorks und gehen Sie mit dem entsprechenden Projekt Online.

- Nehmen Sie konfigurierte Messtastersteuerbits in das Signalsteuerwort mit auf.
- **IndraWorks ► Projekt ► Motion ► Reale Achsen ► Rechtsklick auf Achse: Kommunikation ► Signalsteuerwort**
 - Messtaster 1: S-0-0405, Bit 0
 - Messtaster 2: S-0-0406, Bit 0

	Status	Quellparameter	Bitnummer
Bit 0	1	S-0-0346: Positioniersteuerwort	0
Bit 1	0	S-0-0346: Positioniersteuerwort	3
Bit 2	0	S-0-0346: Positioniersteuerwort	4
Bit 3	0	S-0-0346: Positioniersteuerwort	5
Bit 4	0	S-0-0393: Sollwertmodus	0
Bit 5	0	S-0-0393: Sollwertmodus	1
Bit 6	0	P-0-0088: Steuerwort Synchronbetriebsarten	0
Bit 7	0	P-0-0088: Steuerwort Synchronbetriebsarten	1
Bit 8	0	P-0-0155: Aufsynchonisier-Modus	1
Bit 9	0	P-0-0154: Aufsynchonisier-Richtung	0
Bit 10	0	P-0-0154: Aufsynchonisier-Richtung	1
Bit 11	0	S-0-0000: <reserviert>	0
<Achse>.UserCmdDataBitA	0	S-0-0405: Messtaster 1 Freigabe	0
<Achse>.UserCmdDataBitB	0	S-0-0000: <leer>	0
<Achse>.UserCmdDataBitC	0	S-0-0000: <leer>	0
<Achse>.UserCmdDataBitD	0	S-0-0000: <leer>	0

Anwenden

Abb.5-71: Signalsteuerwort

- Nehmen Sie konfigurierte Messtasterstatusbits in das Signalstatuswort mit auf.
- IndraWorks ► Projekt ► Motion ► Reale Achsen ► Rechtsklick auf Achse: Kommunikation ► Signalstatuswort**
- Messtaster 1, positiv erfasst: S-0-0409, Bit 0
 - Messtaster 1, negativ erfasst: S-0-0410, Bit 0
 - Messtaster 2, positiv erfasst: S-0-0411, Bit 0
 - Messtaster 2, negativ erfasst: S-0-0412, Bit 0

ML_TechBase.library

	Status	Quellparameter	Bitnummer
Bit 0	1	S-0-0330: Meldung n_ist = n_soll	0
Bit 1	1	S-0-0331: Meldung n_ist = 0	0
Bit 2	0	S-0-0437: Positionierstatus	2
Bit 3	0	S-0-0403: Status Lageistwerte	1
Bit 4	0	S-0-0419: Positioniersollwert-Quittung	0
Bit 5	0	P-0-0089: Statuswort Synchronbetriebsarten	0
Bit 6	0	P-0-0089: Statuswort Synchronbetriebsarten	1
Bit 7	1	P-0-0089: Statuswort Synchronbetriebsarten	8
Bit 8	0	S-0-0824: Meldung Drehmoment-/Kraftsollwert erreicht	0
Bit 9	0	P-0-0089: Statuswort Synchronbetriebsarten	5
Bit 10	1	S-0-0336: Meldung In-Position	0
Bit 11	1	P-0-0089: Statuswort Synchronbetriebsarten	6
<Achse>.UserActualDataBitA	0	S-0-0409: Messtaster 1 positiv erfasst	0
<Achse>.UserActualDataBitB	0	S-0-0410: Messtaster 1 negativ erfasst	0
<Achse>.UserActualDataBitC	0	S-0-0000: <leer>	0
<Achse>.UserActualDataBitD	0	S-0-0000: <leer>	0

Anwenden

Abb.5-72: Signalstatuswort

- Nehmen Sie konfigurierte Messtasterwerte in das optionale, zyklische AT-Telegramm auf.

IndraWorks ▶ Projekt ▶ Motion ▶ Reale Achsen ▶ Rechtsklick auf Achse: Kommunikation ▶ Zyklischer SERCOS-Kanal

- Messtaster 1, positive Flanke: S-0-0130
- Messtaster 1, negative Flanke: S-0-0131
- Messtaster 2, positive Flanke: S-0-0132
- Messtaster 2, negative Flanke: S-0-0133
- Messtaster 1, Anzahl der Markenausfälle: P-0-0224
- Messtaster 2, Anzahl der Markenausfälle: P-0-0225
- Messtaster 1, Differenzwert des Messtasters: P-0-0202
- Messtaster 2, Differenzwert des Messtasters: P-0-0203

Konfiguration zyklischer Schreibzugriff (MDT)			
<Achse>.UserCmdDataA	S-0-0000: <leer>		
<Achse>.UserCmdDataB	S-0-0000: <leer>		
<Achse>.UserCmdDataC	S-0-0000: <leer>		
<Achse>.UserCmdDataD	S-0-0000: <leer>		

Konfiguration zyklischer Lesezugriff (AT)			
<Achse>.UserActualDataA	S-0-0130: Messwert 1 positive Flanke	224,9635	Grad
<Achse>.UserActualDataB	S-0-0131: Messwert 1 negative Flanke	0,0000	Grad
<Achse>.UserActualDataC	P-0-0202: Differenz Messwerte 1	135,0365	Grad
<Achse>.UserActualDataD	P-0-0224: Messtaster 1, Anzahl Markenausfälle	0	--

Abb.5-73: Konfigurieren eines zyklischen AT-Telegramms

**Konfiguration der IndraMotion
MLD-S bzw. IndraMotion MLD-M
(Leitachse)**

Die folgenden Antriebsparametereinstellungen sind notwendig, um die Messtasterfunktionalität innerhalb einer IndraMotion MLD-S bzw. IndraMotion MLD-M als Leitachse zu verwenden:

- Nehmen Sie konfigurierte Messtastersteuerbits in das Signalsteuerwort mit auf.

IndraWorks ► Projekt ► MLD ► Rechtsklick auf: AxisData

- Messtaster 1: S-0-0405, Bit 0
- Messtaster 2: S-0-0406, Bit 0

Achse #1 (Lokale Achse)

Sollwerte

wUserCmdDataBitA_q	S-0-0405 : Messtaster 1 Freigabe	0
wUserCmdDataBitB_q	S-0-0406 : Messtaster 2 Freigabe	0
wUserCmdDataBitC_q	S-0-0000 : Frei	0
wUserCmdDataBitD_q	S-0-0000 : Frei	0
dwUserCmdDataA_q	S-0-0000 : Frei	
dwUserCmdDataB_q	S-0-0000 : Frei	
dwUserCmdDataC_q	S-0-0000 : Frei	
dwUserCmdDataD_q	S-0-0000 : Frei	

Istwerte

wUserActualDataBitA_i	S-0-0409 : Messtaster 1 positiv erfasst	0
wUserActualDataBitB_i	S-0-0410 : Messtaster 1 negativ erfasst	0
wUserActualDataBitC_i	S-0-0000 : Frei	0
wUserActualDataBitD_i	S-0-0000 : Frei	0
dwUserActualDataA_i	S-0-0130 : Messwert 1 positive Flanke	
dwUserActualDataB_i	S-0-0131 : Messwert 1 negative Flanke	
dwUserActualDataC_i	P-0-0202 : Differenz Messwerte 1	
dwUserActualDataD_i	P-0-0224 : Messtaster 1, Anzahl Markenausfälle	

Konfiguration übernehmen

Abb.5-74: Sollwertdatenbits

- Nehmen Sie konfigurierte Messtasterstatusbits in das Signalstatuswort mit auf.

IndraWorks ► Rechtsklick auf SERCOS III ► Querkommunikation Antriebe ► Signalstatuswort

- Messtaster 1, positiv erfasst: S-0-0409, Bit 0
- Messtaster 1, negativ erfasst: S-0-0410, Bit 0
- Messtaster 2, positiv erfasst: S-0-0411, Bit 0
- Messtaster 2, negativ erfasst: S-0-0412, Bit 0

ML_TechBase.library

Bit	Status	Quellparameter	Quellbit
0	0	P-0-0115 : Gerätesteuerung: Statuswort	12
1	0	S-0-0331 : Meldung n_ist = 0	0
2	0	S-0-0403 : Status Lageistwerte	0
3	0	S-0-0419 : Positioniersollwert-Quittung	0
4	0	S-0-0521 : Lageregler-Statuswort	0
5	0	S-0-0521 : Lageregler-Statuswort	2
6	0	P-0-0152 : Aufsynchronisieren beendet	0
7	0	P-0-0115 : Gerätesteuerung: Statuswort	7
8	0	S-0-0409 : Messtaster 1 positiv erfasst	0
9	0	S-0-0410 : Messtaster 1 negativ erfasst	0
10	0	S-0-0411 : Messtaster 2 positiv erfasst	0
11	0	S-0-0412 : Messtaster 2 negativ erfasst	0
12	0	S-0-0409 : Messtaster 1 positiv erfasst	0
13	0	S-0-0410 : Messtaster 1 negativ erfasst	0
14	0	S-0-0411 : Messtaster 2 positiv erfasst	0
15	0	S-0-0412 : Messtaster 2 negativ erfasst	0

Die Bits 12 bis 15 entsprechen den Elementen
wUserActualDataBitA_i bis wUserActualDataBitD_i in der Struktur MX_AXISDATA

[Einstellungen](#) [CCD-Signalsteuerwort](#) [Konfiguration übernehmen](#)

Abb.5-75: Istwertdatenbits

- Nehmen Sie konfigurierte Messtasterwerte in das optionale, zyklische AT-Telegramm auf.

IndraWorks ► Projekt ► MLD ► Rechtsklick auf: AxisData

- Messtaster 1, positive Flanke: S-0-0130
- Messtaster 1, negative Flanke: S-0-0131
- Messtaster 2, positive Flanke: S-0-0132
- Messtaster 2, negative Flanke: S-0-0133
- Messtaster 1, Anzahl der Markenausfälle: P-0-0224
- Messtaster 2, Anzahl der Markenausfälle: P-0-0225
- Messtaster 1, Differenzwert des Messtasters: P-0-0202
- Messtaster 2, Differenzwert des Messtasters: P-0-0203
- Nehmen Sie die folgenden Parameter für jedes konfigurierte, gemessene Messtastersignal in den Istwertdatencontainer auf:
 - TP_ACT_POS_FEEDBACK1 => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_POS_FEEDBACK2 => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_POSIN_ACT_CYCLE => Keine zusätzlichen Daten werden benötigt
 - TP_SYNC_CMD_POS => P-0-0779
 - TP_PROBE_TIME => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_MEASURING_ENC_POS => P-0-0332
 - TP_RES_MASTER_AXIS_POS => P-0-0764

ML_TechBase.library

- TP_EFF_MASTER_AXIS_POS => P-0-0777
- TP_CAM_SHAFT_PROF_ACCESS_ANGLE => P-0-0777

Sollwerte	
wUserCmdDataBitA_q	S-0-0405 : Messtaster 1 Freigabe
wUserCmdDataBitB_q	S-0-0406 : Messtaster 2 Freigabe
wUserCmdDataBitC_q	S-0-0000 : Frei
wUserCmdDataBitD_q	S-0-0000 : Frei
dwUserCmdDataA_q	S-0-0000 : Frei
dwUserCmdDataB_q	S-0-0000 : Frei
dwUserCmdDataC_q	S-0-0000 : Frei
dwUserCmdDataD_q	S-0-0000 : Frei

Istwerte	
wUserActualDataBitA_i	S-0-0409 : Messtaster 1 positiv erfasst
wUserActualDataBitB_i	S-0-0410 : Messtaster 1 negativ erfasst
wUserActualDataBitC_i	S-0-0000 : Frei
wUserActualDataBitD_i	S-0-0000 : Frei
dwUserActualDataA_i	S-0-0130 : Messwert 1 positive Flanke
dwUserActualDataB_i	S-0-0131 : Messwert 1 negative Flanke
dwUserActualDataC_i	P-0-0202 : Differenz Messwerte 1
dwUserActualDataD_i	P-0-0224 : Messtaster 1, Anzahl Markenausfälle

Konfiguration übernehmen

Abb.5-76: Sollwertdatenbits

- Setzen Sie P-0-1367, Bit 6 (Update der verwendeten Achsdatenstruktur aktivieren, nur IndraMotion MLD-M)

Konfiguration der IndraMotion MLD-M (Folgeachse)

Die folgenden Antriebsparametereinstellungen sind notwendig, wenn die Messtasterfunktionalität in der IndraMotion MLD-M als Folgeachse verwendet wird:

- Nehmen Sie konfigurierte Messtastersteuerbits in das Signalsteuerwort mit auf.

IndraWorks ► Rechtsklick auf SERCOS III ► Querkommunikation Antriebe ► Signalsteuerwort

- Messtaster 1: S-0-0405, Bit 0
- Messtaster 2: S-0-0406, Bit 0

ML_TechBase.library

Bit	Status	Quellparameter	Quellbit
0	0	S-0-0520 : Achsregler-Steuerwort	0
1	0	S-0-0520 : Achsregler-Steuerwort	2
2	0	P-0-0155 : Aufsynchrisier-Modus	1
3	0	P-0-0154 : Aufsynchrisier-Richtung	0
4	0	P-0-0154 : Aufsynchrisier-Richtung	1
5	0	S-0-0000 : Frei	0
6	0	S-0-0000 : Frei	0
7	0	S-0-0000 : Frei	0
8	0	S-0-0000 : Frei	0
9	0	S-0-0000 : Frei	0
10	0	S-0-0000 : Frei	0
11	0	S-0-0000 : Frei	0
12	0	S-0-0405 : Messtaster 1 Freigabe	0
13	0	S-0-0406 : Messtaster 2 Freigabe	0
14	0	S-0-0000 : Frei	0
15	0	S-0-0000 : Frei	0

Die Bits 12 bis 15 entsprechen den Elementen
wUserCmdDataBitA_q bis wUserCmdDataBitD_q in der Struktur MX_AXISDATA.

[Einstellungen](#) [CCD-Signalstatuswort](#) [Konfiguration übernehmen](#)

Abb.5-77: Signalsteuerwort

- Nehmen Sie konfigurierte Messtasterstatusbits in das Signalstatuswort mit auf.

IndraWorks ► Rechtsklick auf SERCOS III ► Querkommunikation Antriebe ► Signalstatuswort

- Messtaster 1, positiv erfasst: S-0-0409, Bit 0
- Messtaster 1, negativ erfasst: S-0-0410, Bit 0
- Messtaster 2, positiv erfasst: S-0-0411, Bit 0
- Messtaster 2, negativ erfasst: S-0-0412, Bit 0

Bit	Status	Quellparameter	Quellbit
0	0	P-0-0115 : Gerätesteuerung: Statuswort	12
1	0	S-0-0331 : Meldung n_ist = 0	0
2	0	S-0-0403 : Status Lageistwerte	0
3	0	S-0-0419 : Positioniersollwert-Quittung	0
4	0	S-0-0521 : Lageregler-Statuswort	0
5	0	S-0-0521 : Lageregler-Statuswort	2
6	0	P-0-0152 : Aufsynchronisieren beendet	0
7	0	P-0-0115 : Gerätesteuerung: Statuswort	7
8	0	S-0-0409 : Messtaster 1 positiv erfasst	0
9	0	S-0-0410 : Messtaster 1 negativ erfasst	0
10	0	S-0-0411 : Messtaster 2 positiv erfasst	0
11	0	S-0-0412 : Messtaster 2 negativ erfasst	0
12	0	S-0-0409 : Messtaster 1 positiv erfasst	0
13	0	S-0-0410 : Messtaster 1 negativ erfasst	0
14	0	S-0-0411 : Messtaster 2 positiv erfasst	0
15	0	S-0-0412 : Messtaster 2 negativ erfasst	0

Die Bits 12 bis 15 entsprechen den Elementen
wUserActualDataBitA_i bis wUserActualDataBitD_i in der Struktur MX_AXISDATA

[Einstellungen](#) [CCD-Signalsteuerwort](#) [Konfiguration übernehmen](#)

Abb.5-78: Signalstatuswort

- Nehmen Sie konfigurierte Messtasterwerte in das optionale, zyklische AT-Telegramm auf.

IndraWorks ► Rechtsklick auf SERCOS III ► Querkommunikation Antriebe ► Zyklischer SERCOS-Kanal

- Messtaster 1, positive Flanke: S-0-0130
- Messtaster 1, negative Flanke: S-0-0131
- Messtaster 2, positive Flanke: S-0-0132
- Messtaster 2, negative Flanke: S-0-0133
- Messtaster 1, Anzahl der Markenausfälle: P-0-0224
- Messtaster 2, Anzahl der Markenausfälle: P-0-0225
- Messtaster 1, Differenzwert des Messtasters: P-0-0202
- Messtaster 2, Differenzwert des Messtasters: P-0-0203
- Nehmen Sie die folgenden Parameter für jedes konfigurierte, gemessene Messtastersignal in den optionalen, zyklischen ATs auf:
 - TP_ACT_POS_FEEDBACK1 => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_POS_FEEDBACK2 => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_POSIN_ACT_CYCLE => Keine zusätzlichen Daten werden benötigt
 - TP_SYNC_CMD_POS => P-0-0779
 - TP_PROBE_TIME => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_MEASURING_ENC_POS => P-0-0332

ML_TechBase.library

- TP_RES_MASTER_AXIS_POS => P-0-0764
- TP_EFF_MASTER_AXIS_POS => P-0-0777
- TP_CAM_SHAFT_PROF_ACCESS_ANGLE => P-0-0777

Abb.5-79: Konfigurieren eines zyklischen AT-Telegramms

- Setzen Sie P-0-1367, Bit 6 (Update der verwendeten Achsdatenstruktur aktivieren, nur IndraMotion MLD-M)

Konfiguration Probe1

Messtaster 1	AT-Container	Steuerbit	Statusbit
Positive Flanke	S-0-0130	S-0-0405.0	S-0-0409.0
Negative Flanke	S-0-0131	S-0-0405.0	S-0-0410.0
Positive und negative Flanke, Differenz, Markenausfall	S-0-0130, S-0-0131, P-0-0224 UND P-0-0202	S-0-0405.0	S-0-0409.0, S-0-0410.0

Abb.5-80: Konfiguration Probe1

Konfiguration Probe2

Probe2	AT-Container	Steuerbit	Statusbit
Positive Flanke	S-0-0132	S-0-0406.0	S-0-0411.0
Negative Flanke	S-0-0133	S-0-0406.0	S-0-0412.0
Positive und negative Flanke, Differenz, Markenausfall	S-0-0132, S-0-0133, P-0-0225 UND P-0-0203	S-0-0406.0	S-0-0411.0, S-0-0412.0

Abb.5-81: Konfiguration Probe2

Beispielkonfiguration des Signalstatuswortes, Signalsteuerwortes und zyklischen SERCOS-Kanals

- IndraDrive Probe1
 - Negative Flanke
 - Erwartungsfenster deaktiviert
 - Markenausfall deaktiviert
 - Differenzmessung deaktiviert
1. Konfiguration eines Signalsteuerwortes

	Status	Quellparameter	Bitnummer
Bit 0	1	S-0-0346: Positioniersteuerwort	0
Bit 1	0	S-0-0346: Positioniersteuerwort	3
Bit 2	0	S-0-0346: Positioniersteuerwort	4
Bit 3	0	S-0-0346: Positioniersteuerwort	5
Bit 4	0	S-0-0393: Sollwertmodus	0
Bit 5	0	S-0-0393: Sollwertmodus	1
Bit 6	0	P-0-0088: Steuerwort Synchronbetriebsarten	0
Bit 7	0	P-0-0088: Steuerwort Synchronbetriebsarten	1
Bit 8	0	P-0-0155: Aufsynchonisier-Modus	1
Bit 9	0	P-0-0154: Aufsynchonisier-Richtung	0
Bit 10	0	P-0-0154: Aufsynchonisier-Richtung	1
Bit 11	0	S-0-0000: <reserviert>	0
<Achse>.UserCmdDataBitA	1	S-0-0405: Messtaster 1 Freigabe	0
<Achse>.UserCmdDataBitB	0	S-0-0000: <leer>	0
<Achse>.UserCmdDataBitC	0	S-0-0000: <leer>	0
<Achse>.UserCmdDataBitD	0	S-0-0000: <leer>	0

Anwenden

Abb.5-82: Beispiel eines Signalsteuerwortes

2. Konfiguration eines Signalstatuswortes

	Status	Quellparameter	Bitnummer
Bit 0	1	S-0-0330: Meldung n_ist = n_soll	0
Bit 1	1	S-0-0331: Meldung n_ist = 0	0
Bit 2	0	S-0-0437: Positionierstatus	2
Bit 3	0	S-0-0403: Status Lageistwerte	1
Bit 4	0	S-0-0419: Positioniersollwert-Quittung	0
Bit 5	0	P-0-0089: Statuswort Synchronbetriebsarten	0
Bit 6	0	P-0-0089: Statuswort Synchronbetriebsarten	1
Bit 7	0	P-0-0089: Statuswort Synchronbetriebsarten	8
Bit 8	0	S-0-0824: Meldung Drehmoment-/Kraftsollwert erreicht	0
Bit 9	1	P-0-0089: Statuswort Synchronbetriebsarten	5
Bit 10	1	S-0-0336: Meldung In-Position	0
Bit 11	1	P-0-0089: Statuswort Synchronbetriebsarten	6
<Achse>.UserActualDataBitA	1	S-0-0410: Messtaster 1 negativ erfasst	0
<Achse>.UserActualDataBitB	0	S-0-0000: <leer>	0
<Achse>.UserActualDataBitC	0	S-0-0000: <leer>	0
<Achse>.UserActualDataBitD	0	S-0-0000: <leer>	0

Anwenden

Abb.5-83: Beispiel eines Signalstatuswortes

3. Konfiguration eines zyklischen SERCOS-Kanals

Konfiguration zyklischer Schreibzugriff (MDT)

<Achse>.UserCmdDataA	S-0-0000: <leer>	
<Achse>.UserCmdDataB	S-0-0000: <leer>	
<Achse>.UserCmdDataC	S-0-0000: <leer>	
<Achse>.UserCmdDataD	S-0-0000: <leer>	

Konfiguration zyklischer Lesezugriff (AT)

<Achse>.UserActualDataA	S-0-0131: Messwert 1 negative Flanke	0.0000	Deg
<Achse>.UserActualDataB	S-0-0000: <leer>		
<Achse>.UserActualDataC	S-0-0000: <leer>		
<Achse>.UserActualDataD	S-0-0000: <leer>		

Abb.5-84: Beispiel eines zyklischen SERCOS-Kanals

ML_TechBase.library

5.7.3 MB_InitTouchProbe

Kurzbeschreibung Der Funktionsbaustein MB_InitTouchProbe initialisiert die Messtasterfunktionalität.

Schnittstellenbeschreibung

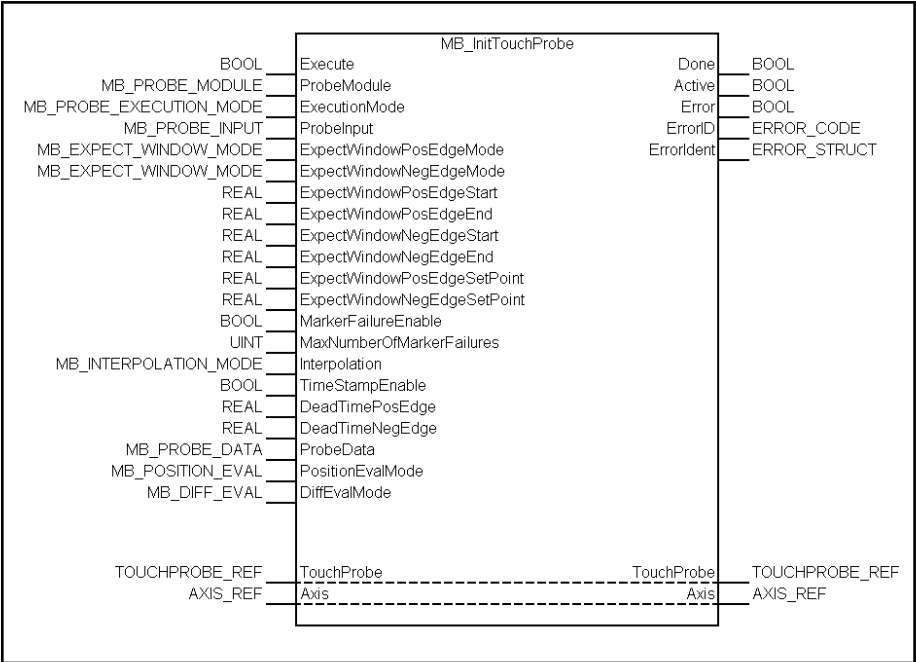


Abb.5-85: Funktionsbaustein MB_InitTouchProbe

E/A-Typ	Name	Datentyp	Kommentar
VAR_IN_OUT	Axis	AXIS_REF	Referenz zur Achse
	TouchProbe	TOUCHPROBE_REF	Referenz zur Signaltriggerquelle
VAR_INPUT	Execute	BOOL	Startet die Initialisierung der Messtasterfunktionalität bei steigender Flanke
	ProbeModule	MB_PROBE_MODULE	Position des Messtasters auf der MLC (onboard, Funktionsmodule)
	ExecutionMode	MB_EXECUTION_MODE	Auswahl, ob einzelne oder fortlaufende Messung
	ProbeInput	MB_PROBE_INPUT	Wählt den digitalen Eingang für den MLC-Messtaster aus
	ExpectWindowPosEdgeMode	MB_EXPECT_WINDOW_MODE	Erwartungsfenstermodus, wenn positive Flanke ausgewertet wird
	ExpectWindowNegEdgeMode	MB_EXPECT_WINDOW_MODE	Erwartungsfenstermodus, wenn negative Flanke ausgewertet wird
	ExpectWindowPosEdgeStart	REAL	Start des Erwartungsfensters, wenn positive Flanke ausgewertet wird
	ExpectWindowPosEdgeEnd	REAL	Ende des Erwartungsfensters, wenn positive Flanke ausgewertet wird
	ExpectWindowNegEdgeStart	REAL	Start des Erwartungsfensters, wenn negative Flanke ausgewertet wird

E/A-Typ	Name	Datentyp	Kommentar
	ExpectWindowNegEdgeEnd	REAL	Ende des Erwartungsfensters, wenn negative Flanke ausgewertet wird
	ExpectWindowPosEdgeSetPoint	REAL	Sollwert für das relative Erwartungsfenster, wenn positive Flanke ausgewertet wird
	ExpectWindowNegEdgeSetPoint	REAL	Sollwert für das relative Erwartungsfenster, wenn negative Flanke ausgewertet wird
	MarkerFailureEnable	BOOL	Markenausfall aktivieren
	MaxNumberOfMarkerFailure	UINT	Schwelle für die registrierte Anzahl der Markenausfälle
	Interpolation	MB_INTERPOLATION_MODE	Interpolation für den MLC-Messtaster
	TimeStampEnable	BOOL	Zeitstempel für den MLC-Messtaster aktivieren
	DeadTimePosEdge	REAL	Totzeitkompensation für die positive Flanke, wenn ausgewertet
	DeadTimeNegEdge	REAL	Totzeitkompensation für die negative Flanke, wenn ausgewertet
	ProbeData	MB_PROBE_DATA	Gemessenes Messtastersignal
	PositionEvalMode	MB_POSITION_EVAL	Wählt die zu auszuwertende Flanke aus
	DiffEvalMode	MB_DIFF_EVAL	Auswahl der Flanke zur Differenzmessung
VAR_OUTPUT	Done	BOOL	Initialisierung durchgeführt
	Active	BOOL	Der Funktionsbaustein ist momentan aktiv
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

Abb.5-86: Schnittstellenvariablen Funktionsbaustein MB_InitTouchProbe

Min-/Max- und Standard-Werte

In der folgenden Tabelle sind der Min-/Max- und der Defaultwert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
Execute	BOOL			FALSE	Kontinuierlich
ProbeModule	MB_PROBE_MODULE			ONBOARD_MODULE	Steigende Flanke an "Execute"
ExecutionMode	MB_PROBE_EXECUTION_MODE			CONTINUOUS	Steigende Flanke an "Execute"
ProbeInput	MB_PROBE_INPUT			D_INPUT_1	Steigende Flanke an "Execute"
ExpectWindowPosEdgeMode	MB_EXPECT_WINDOW_MODE			DISABLED_EXP	Steigende Flanke an "Execute"

ML_TechBase.library

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
ExpectWindow-NegEdgeMode	MB_EXPECT_WINDOW_MODE			DISABLED_EXP	Steigende Flanke an "Execute"
ExpectWindow-PosEdgeStart	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
ExpectWindow-PosEdgeEnd	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
ExpectWindow-NegEdgeStart	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
ExpectWindow-NegEdgeEnd	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
ExpectWindow-PosEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
ExpectWindow-NegEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
MarkerFailureEnable	BOOL			FALSE	Steigende Flanke an "Execute"
MaxNumberOfMarkerFailure	UINT	0		0	Steigende Flanke an "Execute"
TimeStampEnable	BOOL			FALSE	Steigende Flanke an "Execute"
DeadTimePosEdge	REAL	0.0		0.0	Steigende Flanke an "Execute"
DeadTimeNegEdge	REAL	0.0		0.0	Steigende Flanke an "Execute"
ProbeData	MB_PROBE_DATA			TP_ACT_POS_FEEDBACK	Steigende Flanke an "Execute"
PositionEvalMode	MB_POSITION_EVAL			DISABLED_EVAL	Steigende Flanke an "Execute"
DiffEvalMode	MB_DIFF_EVAL			DISABLED_DIFF	Steigende Flanke an "Execute"

Abb.5-87: Min-/Max- und Standard-Werte für die Eingänge des MB_InitTouchProbe

Funktionsbeschreibung des MB_InitTouchProbe

Die in diesem Abschnitt definierten Eingänge, müssen den Datentypen wie in [Kap. "MB_PROBE_EXECUTION_MODE" auf Seite 120](#) beschrieben, entsprechen.



Der Funktionsbaustein MB_InitTouchProbe ist in der SERCOS-Phase 4 auszuführen bevor die Funktionsbausteine MB_TouchProbe und MB_TouchProbeContinuous verwendet werden kann.

TouchProbe, ProbeModule, ProbeInput

Der Eingang "TouchProbe" definiert verwendete Steuerung und Messtaster. Das Element "CntrlNo" muss auf "LOCAL_CNTRL" gesetzt sein. Das Ele-

ment "TouchProbeNo" kann den Wert "TOUCHPROBE_DRIVE_1" oder "TOUCHPROBE_DRIVE_2" annehmen.

Die Eingänge "ProbeModule" und "ProbeInput" werden derzeit nicht ausgewertet.

ExecutionMode

Am Eingang "ExecutionMode", wird die Messtasterfunktion auf Einzelmessung oder fortlaufende Messung gesetzt. Wird die Einzelmessung verwendet, muss der Funktionsbaustein MB_TouchProbe verwendet werden. Für eine fortlaufende Messung ist der Funktionsbaustein MB_TouchProbeContinuous zu verwenden.

ExpectWindow, PositionEvalMode

ExpectWindow beschreibt das Erwartungsfenster des Messtasters. Momentan unterstützt das Erwartungsfenster des Antriebes kein separates Erwartungsfenster für die negativen und positiven Flanken gleichzeitig. Wird nur die positive Flanke ausgewertet ("PositionEvalMode"=POSITION_POS_EDGE), werden die Eingänge "ExpectWindowPosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeEnd" und "ExpectWindowPosEdgeSetPoint" ausgewertet.

Wird die negative Flanke ausgewertet ("PositionEvalMode"=POSITION_NEG_EDGE), werden "ExpectWindowNegEdgeMode", "ExpectWindowNegEdgeStart", "ExpectWindowNegEdgeEnd" und "ExpectWindowNegEdgeSetPoint" ausgewertet.

Werden beide Flanken ausgewertet ("PositionEvalMode"=POSITION_POS_NEG_EDGE), werden nur die Eingänge "ExpectWindowPosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeEnd", "ExpectWindowPosEdgeSetPoint" für das Erwartungsfenster des Antriebes ausgewertet. Das Erwartungsfenster gilt für beide ausgewerteten Flanken.

DeadTimePosEdge, DeadTimeNegEdge

Die Eingänge "DeadTimePosEdge" und "DeadTimeNegEdge" bestimmen die Totzeitkompensation der Flanken. Der Eingang "DeadTimePosEdge" ist gültig, wenn die positive Flanke ausgewertet wird. Der Eingang "DeadTimeNegEdge" ist gültig, wenn die negative Flanke ausgewertet wird.

ProbeData, DiffValMode

Der Eingang "ProbeData" definiert das gemessene Signal der Messtasterfunktion. Die folgenden Signale stehen zur Auswahl:

Signal	Beschreibung
TP_ACT_POS_FEEDBACK1	Messwert positive Flanke
TP_ACT_POS_FEEDBACK2	Messwert negative Flanke
TP_SYNC_CMD_POS	Synchroner Lagesollwert
TP_PROBE_TIME	Zeitstempel
TP_ACT_MEASURING_ENC_POS	Lageistwert Messgeber
TP_RES_MASTER_AXIS_POS	Resultierende Leitachssposition
TP_EFF_MASTER_AXIS_POS	Effektive Leitachssposition
TP_CAM_SHAFT_PROF_ACCESS_ANGLE	Zugriffswinkel der Kurvenscheibentabelle

Abb.5-88: Mögliche Werte des Eingangs ProbeData

Der Eingang "DiffValMode" wählt die Differenzmessung der Messtasterfunktion aus. Momentan wird nur "STANDARD_MODE" unterstützt.

ModuloValue

Der Ausgang "ModuloValue" zeigt den Modulowert des gemessenen Signals an.

Interpolation

Der Eingang "Interpolation" gibt an, wie Positionswerte Interpoliert werden sollen. der Funktionsbaustein MB_InitTouchProbe unterstützt zur Zeit nur die Werte "NO_INTERPOLATION" und "LINEAR_INTERPOLATION". Er wird jedoch nicht weiter ausgewertet.

Zeitdiagramm Zeitdiagramm gemäß der Spezifikation PLCopen.

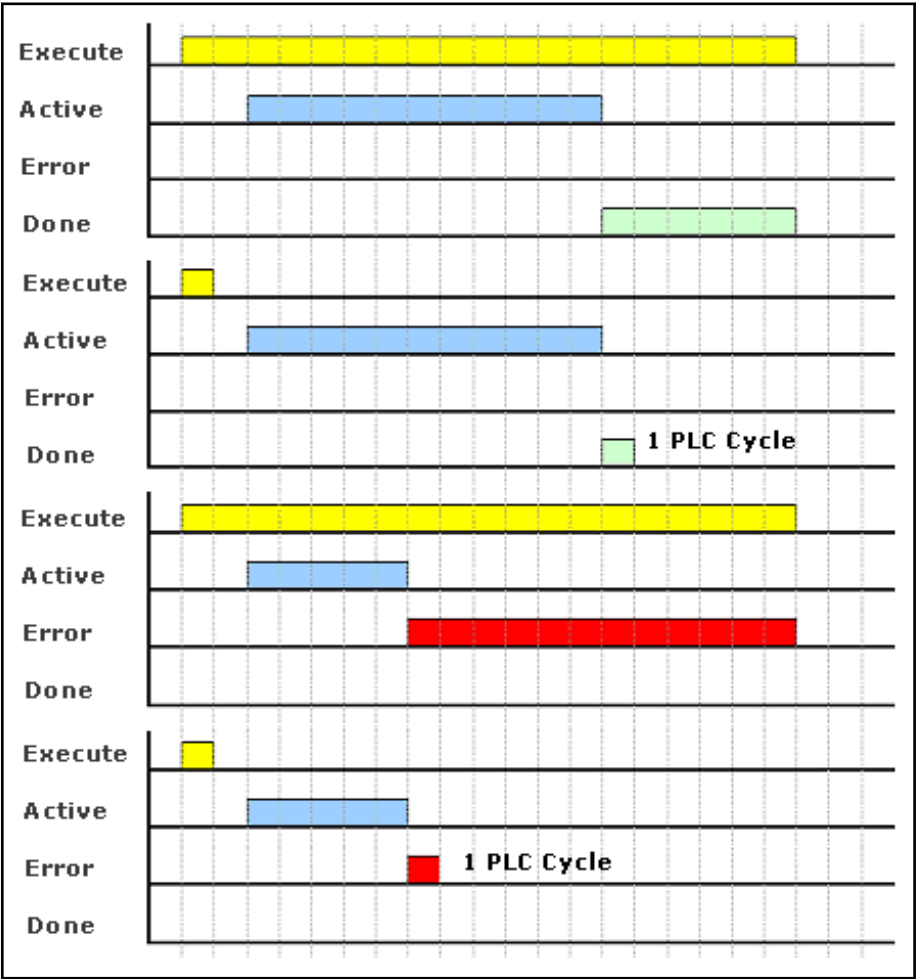


Abb.5-89: Zeitdiagramm Funktionsbaustein MB_InitTouchProbe

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für "F_RELATED_TABLE" 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins

ML_TechBase.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von „Axis“ ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde geändert während der FB in Betrieb ist
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0001	Der Eingang "ProbeModule" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0002	Der Eingang "ExecutionMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0003	Der Eingang "ProbeInput" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0004	Der Eingang "DeadTimePosEdge" ist ungültig (überprüfe, ob < 0)
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0005	Der Eingang "DeadTimeNegEdge" ist ungültig (überprüfe, ob < 0)
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0006	Der Eingang "ProbeData" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0007	Der Eingang "PositionEvalMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0008	Der Eingang "DiffEvalMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0009	Der Eingang "MaxNumberOfMarkerFailure" ist ungültig (überprüfe, ob < 0)
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0001	Der Eingang "ExpectWindowPosEdgeMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0002	Der Eingang "ExpectWindowNegEdgeMode" ist ungültig
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0003	Der Eingang "ExpectWindowPosEdgeStart" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0004	Der Eingang "ExpectWindowPosEdgeEnd" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0005	Der Eingang "ExpectWindowNegEdgeStart" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0006	Der Eingang "ExpectWindowNegEdgeEnd" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0007	Der Eingang "ExpectWindowPosEdgeSetPoint" ist außerhalb des gültigen Bereichs

ML_TechBase.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0008	Der Eingang "ExpectWindowNegEdgeSetPoint" ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0403	16#0001	Benötigtes Messtastersteuerbit (S-0-0405, Bit 0) ist nicht im Signalsteuerwort konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0002	Benötigtes Messtastersteuerbit (S-0-0406, Bit 0) ist nicht im Signalsteuerwort konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0003	Benötigtes Messtastersteuerbit (S-0-0405, Bit 0) ist nicht im Signalstatuswort konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0004	Benötigtes Messtastersteuerbit (S-0-0410, Bit 0) ist nicht im Signalstatuswort konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0005	Benötigtes Messtastersteuerbit (S-0-0411, Bit 0) ist nicht im Signalstatuswort konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0006	Benötigtes Messtastersteuerbit (S-0-0412, Bit 0) ist nicht im Signalstatuswort konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0007	Benötigter Messtasterwert S-0-0130 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0008	Benötigter Messtasterwert S-0-0131 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0009	Benötigter Messtasterwert S-0-0132 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#000A	Benötigter Messtasterwert S-0-0133 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#000B	Benötigter Messtasterparameter P-0-0224 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#000C	Benötigter Messtasterparameter P-0-0225 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#000D	Benötigter Messtasterparameter P-0-0202 ist nicht zyklisch im AT konfiguriert

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0403	16#000E	Benötigter Messtasterparameter P-0-0203 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#00XX	P-0-1367, Bit 6 der MLD-M-Masterachse ist nicht auf TRUE gesetzt. Dies ist jedoch für ein Update der AxisData-Struktur notwendig

Abb.5-90: Fehlercodes Funktionsbaustein MB_InitTouchProbe

Parametrierte Parameter

Der Funktionsbaustein MB_InitTouchProbe parametriert die folgenden Parameter:

- P-0-0226, Messtaster, erweitertes Steuerwort
- S-0-0169, Messtastersteuerparameter
- S-0-0170, Kommando Messtasterzyklus
- S-0-0426 Signalauswahl Messtaster1
- S-0-0427 Signalauswahl Messtaster2

5.7.4 MB_TouchProbe

Kurzbeschreibung

Der Funktionsbaustein MB_TouchProbe wird zum Erfassen einer Achsposition, Leitachseposition oder zur Zeiterfassung bei einem Triggerereignis verwendet. Dabei handelt es sich um einen Funktionsbaustein für die Einzelmessung. Der Funktionsbaustein muss erneut ausgeführt werden, um ein weiteres Signal zu erfassen. Das zu messende Signale wird am Eingang "TouchProbe" des Funktionsbausteins MB_InitTouchProbe definiert.

Schnittstellenbeschreibung

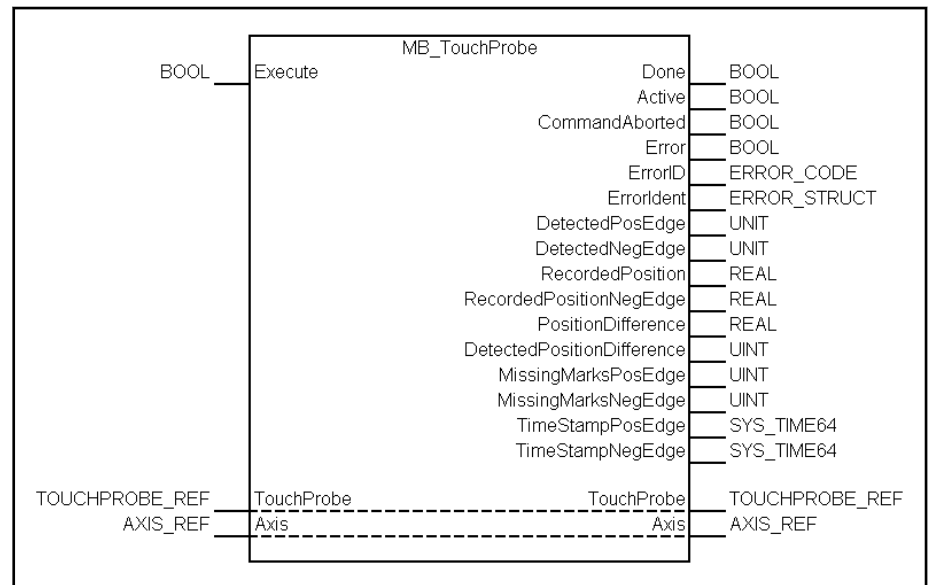


Abb.5-91: Funktionsbaustein MB_TouchProbe

E/A-Typ	Name	Datentyp	Kommentar
VAR_IN_OUT	Axis	AXIS_REF	Referenz zur Achse
	TouchProbe	TOUCHPROBE_REF	Referenz zur Signaltriggerquelle
VAR_INPUT	Execute	BOOL	Startet die Messtasteraufzeichnung bei steigender Flanke

ML_TechBase.library

E/A-Typ	Name	Datentyp	Kommentar
VAR_OUTPUT	Done	BOOL	Triggerereignis aufgezeichnet
	Active	BOOL	Der Funktionsbaustein ist momentan aktiv
	CommandAborted	BOOL	Befehl durch MB_AbortTrigger abgebrochen
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	DetectedPosEdge	UINT	Positive Flanke erkannt
	DetectedNegEdge	UINT	Negative Flanke erkannt
	RecordedPositionPosEdge	REAL	Position, an der die positive Flanke auftrat (in technischen Einheiten)
	RecordedPositionNegEdge	REAL	Position, an der die negative Flanke auftrat (in technischen Einheiten)
	PositionDifference	REAL	Zeigt die Differenz zwischen den Messtasterwerten in Abhängigkeit von "DiffEvalMode" an
	DetectedPositionDifference	UINT	Differenzmessung erkannt
	MissingMarksPosEdge	UINT	Zähler der fehlenden Marken, wenn die positive Flanke ausgewertet wird
	MissingMarksNegEdge	UINT	Zähler der fehlenden Marken, wenn die negative Flanke ausgewertet wird
	TimeStampPosEdge	SYS_TIME64	Zeitpunkt, wann die positive Flanke auftrat (nur MLC/XLC-Messtaster)
	TimeStampNegEdge	SYS_TIME64	Zeitpunkt, wann die negative Flanke auftrat (nur MLC/XLC-Messtaster)

Abb.5-92: Schnittstellenvariablen Funktionsbaustein MB_TouchProbe

Standardwert In der folgenden Tabelle ist der Standardwert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
Enable	BOOL	n/a	n/a	FALSE	Kontinuierlich

Abb.5-93: Standardwert für den Eingang des MB_TouchProbe

Funktionsbeschreibung des MB_TouchProbe

Der Funktionsbaustein MB_TouchProbe führt eine Einzelmessungen aus. Der Funktionsbaustein wird bei steigender Flanke am Eingang "Execute" aktiviert.

DetectedPosEdge, DetectedNegEdge

Ist die Messtasterfunktion eingestellt um eine positive/negative Flanke des Eingangssignals auszuwerten, erhöht sich der Ausgang "DetectedPosEdge"/"DetectedNegEdge", wenn eine positiven/negativen Flanke erkannt wird.

RecordedPositionPosEdge, RecordedPositionNegEdge

Die Ausgänge "RecordedPositionPosEdge" bzw. "RecordedPositionNegEdge" enthalten die Position, an der die positive bzw. negative Flanke auftrat.

PositionDifference, DetectedPositionDifference

Der Ausgang "PositionDifference" enthält im Falle einer aktivierten Differenzmessung die Differenz der beiden zuletzt gemessenen Werte des gleichen Messtasters. Bei jeder Messung wird der Ausgang "DetectedPositionDifference" erhöht.

Die folgende Berechnung gilt, wenn die Antriebsmesstasterfunktion für den Ausgang "PositionDifference" aktiviert wird:

$$\text{PositionDifference} = |(S-0-0130) - (S-0-0131)|$$

$$\text{PositionDifference} = |(S-0-0132) - (S-0-0133)|$$

Abb.5-94: Berechnung der Positionsdivferenz für Achsen

MissingMarksPosEdge, MissingMarksNegEdge

Die Ausgänge "MissingMarksPosEdge" bzw. "MissingMarksNegEdge" werden nur ausgewertet, wenn das Erwartungsfenster aktiv ist und der Markenausfall aktiviert wurde. Sie zählen die Marken außerhalb des Erwartungsfensters. Die Ausgänge werden jedesmal auf 0 gesetzt, wenn eine Marke innerhalb des Erwartungsfensters erkannt wird. Ist der Messtaster eingestellt um beide Flanken auszuwerten, ist nur der Ausgang "MissingMarksPosEdge" gültig. Momentan unterstützt die Antriebsmesstasterfunktion nicht gleichzeitig individuelle Markenausfälle für jede Flanke.

Zeitdiagramm

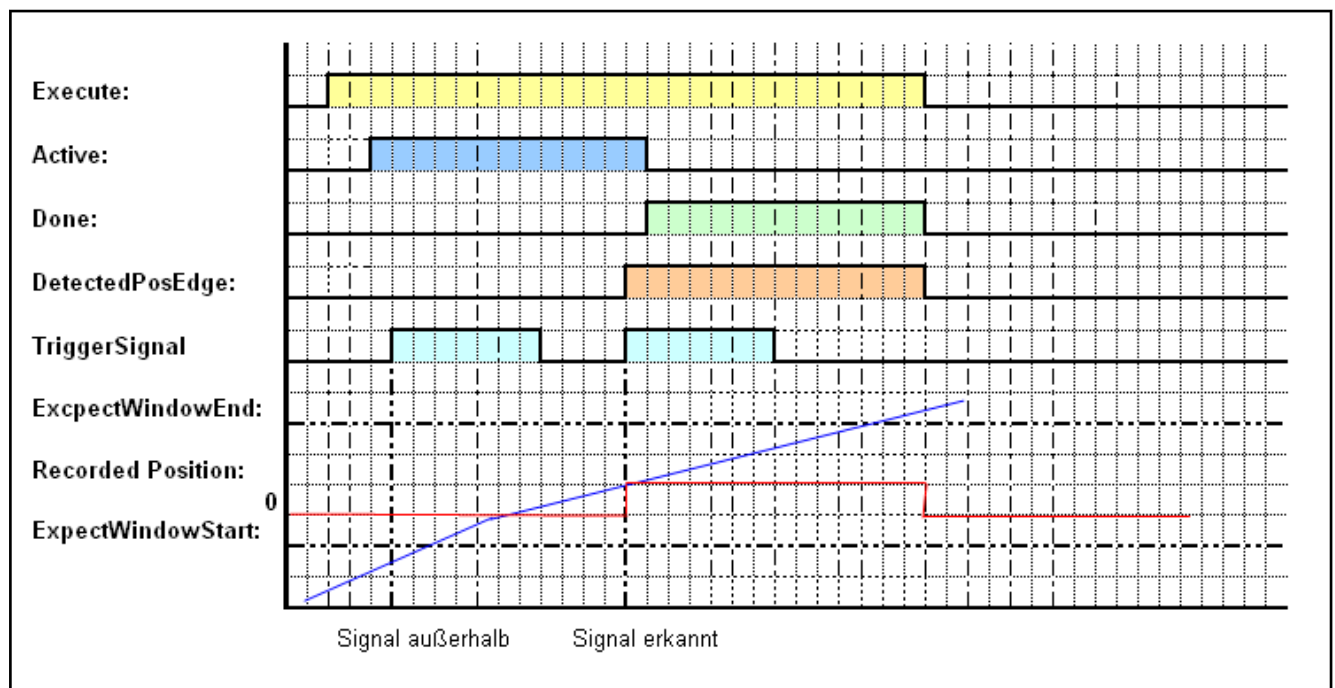


Abb.5-95: Zeitdiagramm Funktionsbaustein MB_TouchProbe

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für "F_RELATED_TABLE" 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins

ML_TechBase.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von "Axis" ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde geändert während der FB in Betrieb ist
ACCESS_ERROR (16#0003)	16#0406	16#0001	Leeres Handle für ATPosEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0002	Leeres Handle für ATNegEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0003	Leeres Handle für ATMarkerFailure
ACCESS_ERROR (16#0003)	16#0406	16#0004	Leeres Handle für ATDiffProbeVal
ACCESS_ERROR (16#0003)	16#0406	16#0005	Leeres Handle für ATDeadTime-Comp
ACCESS_ERROR (16#0003)	16#0406	16#0006	Leeres Handle für SCWEnableProbe
ACCESS_ERROR (16#0003)	16#0406	16#0007	Leeres Handle für SSWPosLatched
ACCESS_ERROR (16#0003)	16#0406	16#0008	Leeres Handle für SSWNegLatched

Abb.5-96: Fehlercodes Funktionsbaustein MB_TouchProbe

5.7.5 MB_TouchProbeContinuous

Kurzbeschreibung Der Funktionsbaustein MB_TouchProbeContinuous wird zum Erfassen einer Achsposition, Leitachspannung oder zur Zeiterfassung bei einem Triggerereignis verwendet. Er führt eine fortlaufende Messung aus, solange der Eingang "Enable" TRUE ist. Das zu erfassende Signale, wird am Eingang "TouchProbe" des Funktionsbausteins MB_InitTouchProbe definiert.

Schnittstellenbeschreibung

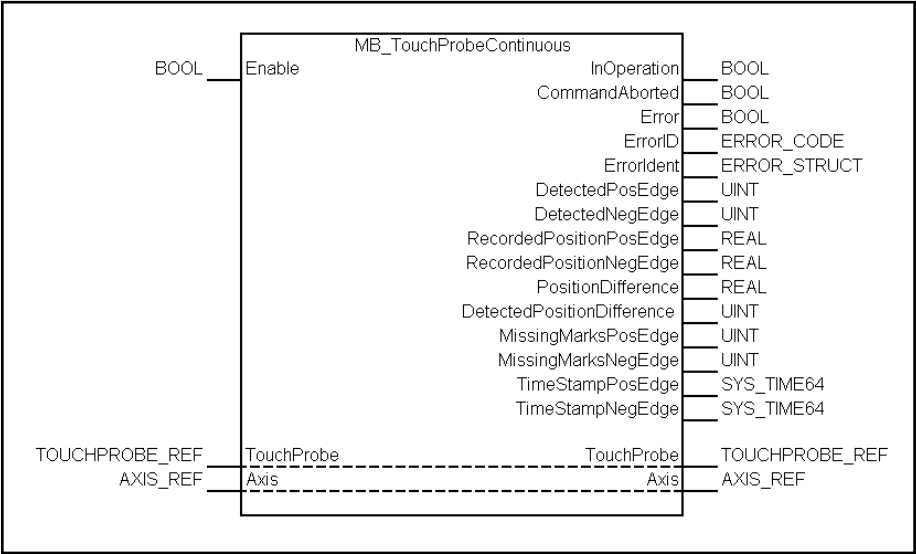


Abb.5-97: Funktionsbaustein MB_TouchProbeContinuous

E/A-Typ	Name	Datentyp	Kommentar
VAR_IN_OUT	Axis	AXIS_REF	Referenz zur Achse
	TouchProbe	TOUCHPROBE_REF	Referenz zur Signaltriggerquelle
VAR_INPUT	Enable	BOOL	Startet die Messtasteraufzeichnung bei TRUE
VAR_OUTPUT	InOperation	BOOL	Messtaster in Betrieb
	CommandAborted	BOOL	Befehl durch MB_AbortTrigger abgebrochen
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	DetectedPosEdge	UINT	Positive Flanke erkannt
	DetectedNegEdge	UINT	Negative Flanke erkannt
	RecordedPositionPosEdge	REAL	Position, an der die positive Flanke auftrat (in technischen Einheiten)
	RecordedPositionNegEdge	REAL	Position, an der die negative Flanke auftrat (in technischen Einheiten)
	PositionDifference	REAL	Zeigt die absolute Differenz zwischen den Messtasterwerten an
	DetectedPositionDifference	UINT	Differenzmessung erkannt
	MissingMarksPosEdge	UINT	Zähler der fehlenden Marken für die positive Flanke, wenn ausgewertet
	MissingMarksNegEdge	UINT	Zähler der fehlenden Marken für die negative Flanke, wenn ausgewertet
	TimeStampPosEdge	MB_TIME_STAMP	Zeitpunkt, wann die positive Flanke auftrat (nur MLC/XLC-Messtaster)
	TimeStampNegEdge	MB_TIME_STAMP	Zeitpunkt, wann die negative Flanke auftrat (nur MLC/XLC-Messtaster)

Abb.5-98: Schnittstellenvariablen Funktionsbaustein MB_TouchProbeContinuous

Standardwert In der folgenden Tabelle ist der Standardwert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
Enable	BOOL			FALSE	Kontinuierlich

Abb.5-99: Standardwert für den Eingang des MB_InitTouchProbe

Funktionsbeschreibung des MB_TouchProbeContinuous

Der Funktionsbaustein MB_TouchProbeContinuous wird ausschließlich für die fortlaufende Messung verwendet. Der Funktionsbaustein ist aktiviert, solange der Eingang "Enable" TRUE ist.

DetectedPosEdge, DetectedNegEdge

Ist die Messtasterfunktion eingestellt um eine positive/negative Flanke des Eingangssignals auszuwerten, erhöht sich der Ausgang "DetectedPosEdge"/"DetectedNegEdge", wenn eine positiven/negativen Flanke erkannt wird.

RecordedPositionPosEdge, RecordedPositionNegEdge

Die Ausgänge "RecordedPositionPosEdge" bzw. "RecordedPositionNegEdge" enthalten die Position, an der die positive bzw. negative Flanke auftrat.

PositionDifference, DetectedPositionDifference

Der Ausgang "PositionDifference" enthält im Falle einer aktivierten Differenzmessung die Differenz der beiden zuletzt gemessenen Werte des gleichen Messtasters. Bei jeder Messung wird der Ausgang "DetectedPositionDifference" erhöht.

Die folgende Berechnung gilt, wenn die Antriebsmesstasterfunktion für den Ausgang "PositionDifference" aktiviert wird:

$$\text{PositionDifference} = |(S-0-0130) - (S-0-0131)|$$

$$\text{PositionDifference} = |(S-0-0132) - (S-0-0133)|$$

Abb.5-100: Berechnung der Positionsdifferenz für Achsen

MissingMarksPosEdge, MissingMarksNegEdge

Die Ausgänge "MissingMarksPosEdge" bzw. "MissingMarksNegEdge" werden nur ausgewertet, wenn das Erwartungsfenster aktiv ist und der Markenausfall aktiviert wurde. Sie zählen die Marken außerhalb des Erwartungsfensters. Die Ausgänge werden jedesmal auf 0 gesetzt, wenn eine Marke innerhalb des Erwartungsfensters erkannt wird. Ist der Messtaster eingestellt um beide Flanken auszuwerten, ist nur der Ausgang "MissingMarksPosEdge" gültig. Momentan unterstützt die Antriebsmesstasterfunktion nicht gleichzeitig individuelle Markenausfälle für jede Flanke.

Zeitdiagramm

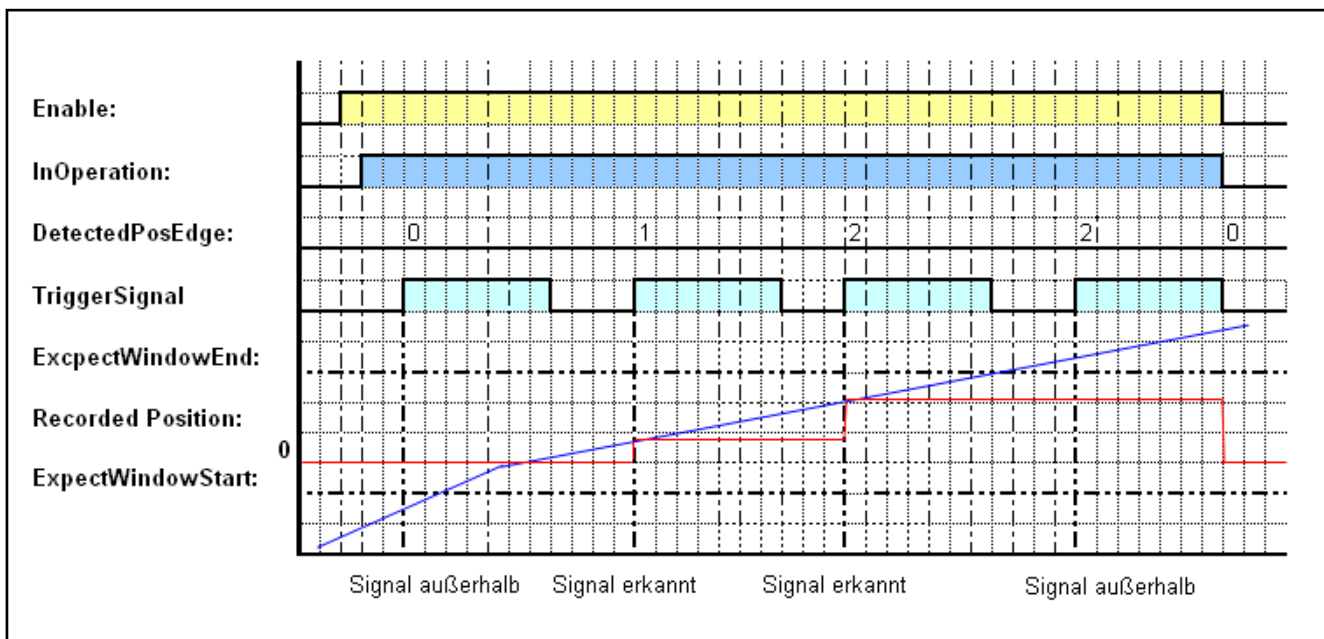


Abb.5-101: Zeitdiagramm Funktionsbaustein MB_TouchProbeContinuous

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für "F_RELATED_TABLE" 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von „Axis“ ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde geändert während der FB in Betrieb ist
ACCESS_ERROR (16#0003)	16#0406	16#0001	Leeres Handle für ATPosEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0002	Leeres Handle für ATNegEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0003	Leeres Handle für ATMarkerFailure
ACCESS_ERROR (16#0003)	16#0406	16#0004	Leeres Handle für ATDiffProbeVal
ACCESS_ERROR (16#0003)	16#0406	16#0005	Leeres Handle für ATDeadTime-Comp
ACCESS_ERROR (16#0003)	16#0406	16#0006	Leeres Handle für SCWEnableProbe
ACCESS_ERROR (16#0003)	16#0406	16#0007	Leeres Handle für SSWPosLatched
ACCESS_ERROR (16#0003)	16#0406	16#0008	Leeres Handle für SSWNegLatched

Abb.5-102: Fehlercodes Funktionsbaustein MB_TouchProbeContinuous

5.7.6 MB_WriteExpectWindow

Kurzbeschreibung

Der Funktionsbaustein MB_WriteExpectWindow konfiguriert das Erwartungsfenster für die Messtasterfunktionalität.

Schnittstellenbeschreibung

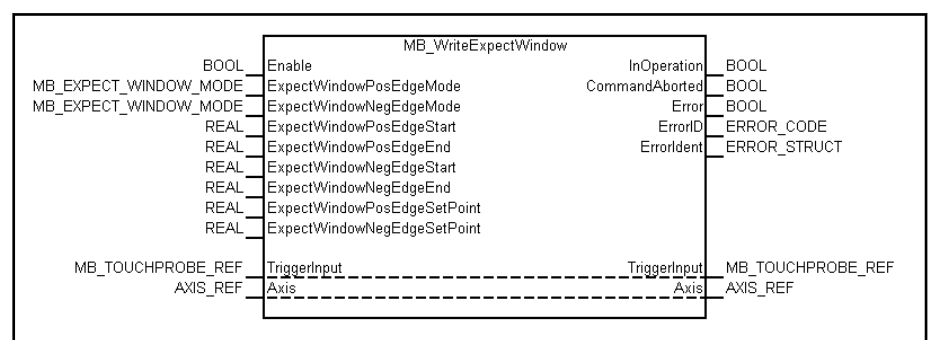


Abb.5-103: Funktionsbaustein MB_WriteExpectWindow

E/A-Typ	Name	Datentyp	Kommentar
VAR_IN_OUT	Axis	AXIS_REF	Referenz zur Achse
	TouchProbe	TOUCHPROBE_REF	Referenz zur Signaltriggerquelle
VAR_INPUT	Enable	BOOL	Aktiviert des Funktionsbaustein

ML_TechBase.library

E/A-Typ	Name	Datentyp	Kommentar
	ExpectWindowPosEdgeMode	MB_EXPECT_WINDOW_MODE	Erwartungsfenstermodus, wenn positive Flanke ausgewertet wird
	ExpectWindowNegEdgeMode	MB_EXPECT_WINDOW_MODE	Erwartungsfenstermodus, wenn negative Flanke ausgewertet wird
	ExpectWindowPosEdgeStart	REAL	Start des Erwartungsfensters, wenn positive Flanke ausgewertet wird
	ExpectWindowPosEdgeEnd	REAL	Ende des Erwartungsfensters, wenn positive Flanke ausgewertet wird
	ExpectWindowNegEdgeStart	REAL	Start des Erwartungsfensters, wenn negative Flanke ausgewertet wird
	ExpectWindowNegEdgeEnd	REAL	Ende des Erwartungsfensters, wenn negative Flanke ausgewertet wird
	ExpectWindowPosEdgeSetPoint	REAL	Sollwert für das relative Erwartungsfenster, wenn positive Flanke ausgewertet wird
	ExpectWindowNegEdgeSetPoint	REAL	Sollwert für das relative Erwartungsfenster, wenn negative Flanke ausgewertet wird
VAR_OUTPUT	InOperation	BOOL	FB in Betrieb
	CommandAborted	BOOL	FB abgebrochen
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

Abb.5-104: Schnittstellenvariablen Funktionsbaustein MB_WriteExpectWindow

Min-/Max- und Standard-Werte

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
Enable	BOOL			FALSE	Kontinuierlich
ExpectWindowPosEdgeMode	MB_EXPECT_WINDOW_MODE			DISABLED_EXP	Ansteigende Flanke bei „Enable“
ExpectWindowNegEdgeMode	MB_EXPECT_WINDOW_MODE			DISABLED_EXP	Ansteigende Flanke bei „Enable“
ExpectWindowPosEdgeStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich
ExpectWindowPosEdgeEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich
ExpectWindowNegEdgeStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich
ExpectWindowNegEdgeEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
ExpectWindow-PosEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Kontinuierlich
ExpectWindow-NegEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Kontinuierlich

Abb.5-105: Min-/Max- und Standard-Werte für die Eingänge des Funktionsbausteins MB_WriteExpectWindow

Funktionsbeschreibung des Antriebsmesstasters

Momentan unterstützt das Erwartungsfensters des Antriebes kein separates Erwartungsfenster für die negativen und positiven Flanken gleichzeitig. Wird nur die positive Flanke ausgewertet, werden die Eingänge "ExpectWindowPosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeStart" und "ExpectWindowPosEdgeSetPoint" ausgewertet.

Wird die negative Flanke ausgewertet, werden "ExpectWindowNegEdgeMode", "ExpectWindowNegEdgeStart", "ExpectWindowNegEdgeStart" und "ExpectWindowNegEdgeSetPoint" ausgewertet.

Werden beide Flanken ausgewertet, werden nur die Eingänge "ExpectWindowPosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeEnd", "ExpectWindowPosEdgeSetPoint" für das Erwartungsfenster des Antriebes ausgewertet. Das Erwartungsfenster gilt für beide ausgewerteten Flanken.

Zeitdiagramm

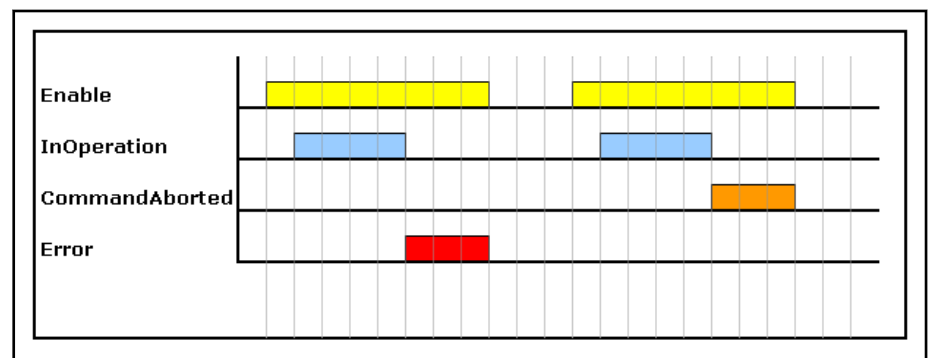


Abb.5-106: Zeitdiagramm Funktionsbaustein MB_WriteExpectWindow

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für "F_RELATED_TABLE" 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von „Axis“ ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde geändert während der FB in Betrieb ist
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0001	Der Eingang "ExpectWindowPosEdgeMode" ist ungültig

ML_TechBase.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0002	Der Eingang "ExpectWindowNegEdgeMode" ist ungültig
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0003	Der Eingang "ExpectWindowPosEdgeStart" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0004	Der Eingang "ExpectWindowPosEdgeEnd" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0005	Der Eingang "ExpectWindowNegEdgeStart" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0006	Der Eingang "ExpectWindowNegEdgeEnd" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0007	Der Eingang "ExpectWindowPosEdgeSetPoint" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0008	Der Eingang "ExpectWindowNegEdgeSetPoint" ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0403	16#000F	P-0-1367, Bit 6 der MLD-M-Masterachse ist nicht auf TRUE gesetzt. Dies ist jedoch für ein Update der AxisData-Struktur notwendig.

Abb.5-107: Fehlercodes Funktionsbaustein MB_WriteExpectWindow

5.7.7 MB_AbortTrigger

Kurzbeschreibung Der Funktionsbaustein MB_AbortTrigger wird zum Abbrechen des Funktionsbausteins MB_TouchProbe verwendet.

Schnittstellenbeschreibung

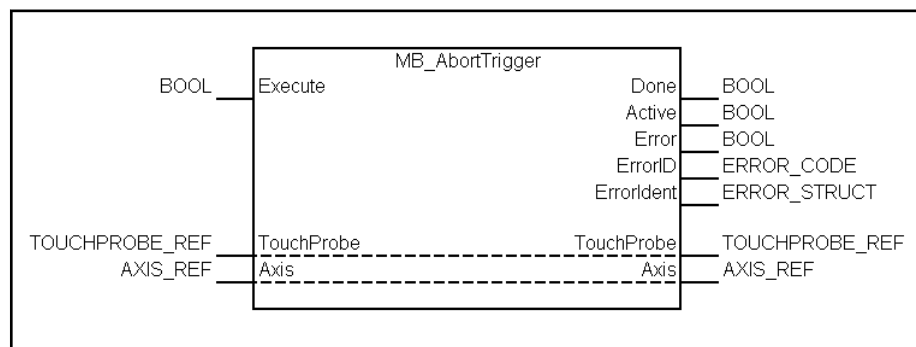


Abb.5-108: Funktionsbaustein MB_AbortTrigger

E/A-Typen	Name	Datentyp	Kommentar
VAR_IN_OUT	Axis	AXIS_REF	Referenz zur Achse
	TouchProbe	TOUCHPROBE_REF	Referenz zur Signaltriggerquelle
VAR_INPUT	Execute	BOOL	Startet den Funktionsbaustein bei steigender Flanke
VAR_OUTPUT	Done	BOOL	Messtaster abgebrochen

	Active	BOOL	Der Funktionsbaustein ist momentan aktiv
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

Abb.5-109: Schnittstellenvariablen Funktionsbaustein MB_AbortTrigger

Standardwert In der folgenden Tabelle ist der Standardwert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
Execute	BOOL			FALSE	Kontinuierlich

Abb.5-110: Standardwert für den Eingang des Funktionsbausteins MB_AbortTrigger

Funktionsbeschreibung Der Funktionsbaustein MB_AbortTrigger deaktiviert die Messtasterfunktion. Er kann nur in Verbindung mit dem Baustein MB_TouchProbe verwendet werden. Um die fortlaufende Messung zu deaktivieren, muss der Eingang "Enable" des Funktionsbausteins MB_TouchProbeContinuous auf FALSE gesetzt werden.

Zeitdiagramm Zeitdiagramm gemäß der PLCopen-Spezifikation.

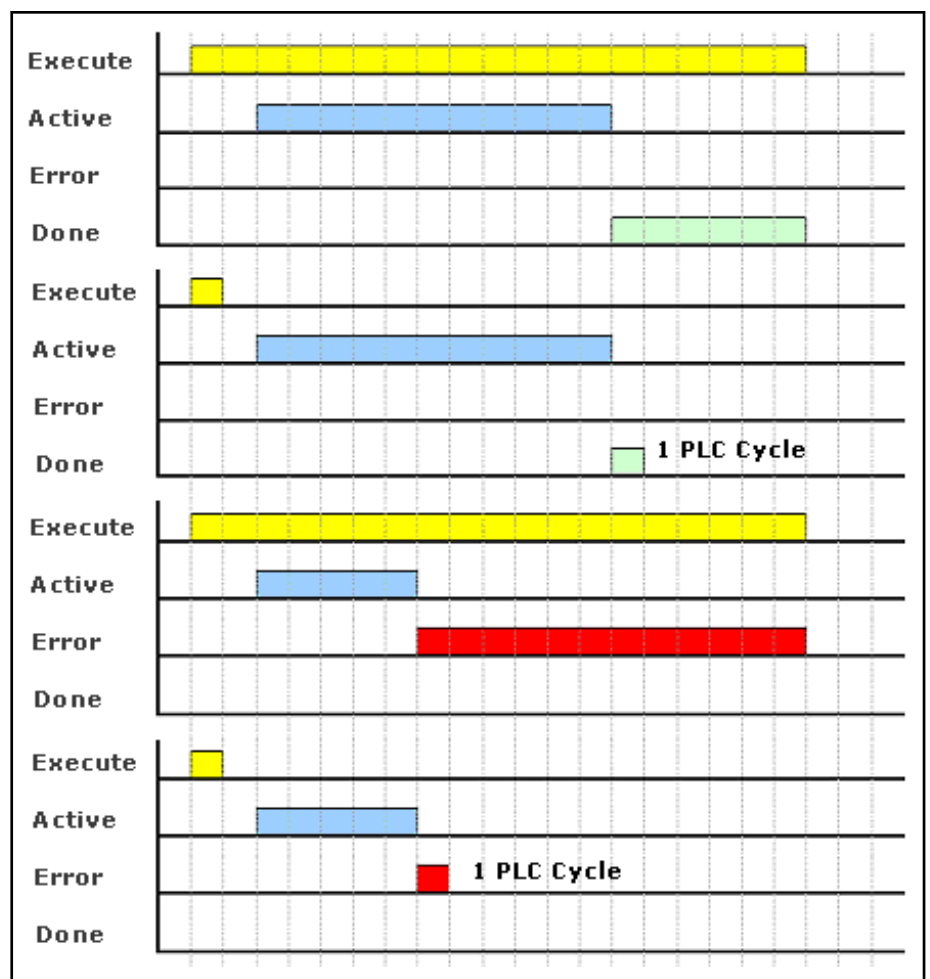


Abb.5-111: Zeitdiagramm Funktionsbaustein MB_AbortTrigger

ML_TechBase.library

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für "F_RELATED_TABLE" 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von „Axis“ ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde geändert während der FB in Betrieb ist
ACCESS_ERROR (16#0003)	16#0406	16#0006	Leeres Handle für SCWEnableProbe

Abb.5-112: Fehlercodes Funktionsbaustein MB_AbortTrigger

5.7.8 Messtasterdatentypen

MB_PROBE_EXECUTION_MODE

Die folgenden Datentypen definieren den Messtasterfunktionsmodus entweder als Einzelmessung oder als fortlaufende Messung:

Programm:

```

TYPE MB_PROBE_EXECUTION_MODE :
(
  CONTINUOUS := 0, (* probe function continuous measurement *)
  SINGLE_SHOT := 1 (* probe function single shot measurement *)
);
END_TYPE

```

MB_EXPECT_WINDOW_MODE

Ist der Eingang "ExpectWindowMode" auf DISABLED_EXP gesetzt, ist das Erwartungsfenster deaktiviert.

Ist der Eingang "ExpectWindowMode" auf EXPECT_WIN_ABS gesetzt, sind der Beginn und das Ende des Erwartungsfensters absolut.

Ist der Eingang "ExpectWindowMode" auf EXPECT_WIN_REL gesetzt, ist der Beginn und das Ende des Erwartungsfensters relativ zum Variablenwert des ExpectWindowPosEdgeSetPoint(ExpectWindowPosEdgeSetPoint).

Das relative Erwartungsfenster für die positive Flanke ist [ExpectWindowPosEdgeSetPoint – ExpectWindowPosEdgeStart .. ExpectWindowPosEdgeSetPoint + ExpectWindowPosEdgeEnd].

Für die negative Flanke ist es [ExpectWindowNegEdgeSetPoint – ExpectWindowNegEdgeStart .. ExpectWindowNegEdgeSetPoint + ExpectWindowNegEdgeEnd].

Programm:

```

TYPE MB_EXPECT_WINDOW_MODE :
(
  EXPECT_WINDOW_DISABLED := 0, (* expectation window disabled *)
  EXPECT_WINDOW_ABS      := 1, (* expectation window absolute *)
  EXPECT_WINDOW_REL      := 2 (* expectation window relative to "ExpectWindowSetPoint" *)
);
END_TYPE

```

MB_PROBE_DATA

Der folgende Datentyp definiert das gemessene Messtastersignal und konfiguriert die Parameter S-0-0426/S-0-0427 abhängig vom ausgewählten Messtaster:

Programm:

```
TYPE MB_PROBE_DATA:
(
  TP_ACT_POS_FEEDBACK1:=1, (* S-0-0051 A-0-0100 Position feedback 1 value *)
  TP_ACT_POS_FEEDBACK2:=2, (* S-0-0053 A-0-0100 Position feedback 2 value *)
  TP_ACT_POSIN_ACT_CYCLE:=3, (* P-0-0753 Position actual value in actual value cycle *)
  TP_SYNC_CMD_POS :=4, (* P-0-0778 A-0-2653 Synchronous position command value *)
  TP_PROBE_TIME:=5, (* time measurement *)
  TP_ACT_MEASURING_ENC_POS:=6, (* P-0-0052 Actual position value of measuring encoder *)
  TP_RES_MASTER_AXIS_POS:=7, (* P-0-0775 A-0-2650 Resulting master axis position *)
  TP_EFF_MASTER_AXIS_POS:=8, (* P-0-0776 A-0-2651 Effective master axis position *)
  TP_CAM_SHAFT_PROF_ACCESS_ANGLE:=9(*P-0-0227 A-0-2656 Cam shaft profile, access angle*)
);
END_TYPE
```

MB_POSITION_EVAL

Der folgende Datentyp wählt die zu bewertende Flanke aus:

Programm:

```
TYPE MB_POSITION_EVAL :
(
  POSITION_EVAL_DISABLED := 0, (* disabled *)
  POSITION_POS_EDGE := 1, (* positive edge is evaluated *)
  POSITION_NEG_EDGE := 2, (* negative edge is evaluated *)
  POSITION_POS_NEG_EDGE := 3 (* positive and negative edge is evaluated *)
);
END_TYPE
```

MB_DIFF_EVAL

Der folgende Datentyp wählt die Differenzmessung aus:

- DIFF_DISABLED: Differenzmessung deaktiviert
- DIFF_STANDARD_MODE: Die Differenz der beiden zuletzt gemessenen Werte desselben Messtasters (IndraMotion MLC/IndraLogic XLC/Antriebsmesstaster)
- DIFF_NEG_TO_NEG_EDGE: Die Differenz der beiden zuletzt ausgewerteten negativen Flankenwerte desselben Messtasters (IndraMotion MLC/IndraLogic XLC nur Messtaster)
- DIFF_POS_TO_NEG_EDGE: Die Differenz des zuletzt positiv und zuletzt negativ ausgewerteten Flankenwertes desselben Messtasters (IndraMotion MLC/IndraLogic XLC nur Messtaster)
- DIFF_NEG_TO_POS_EDGE: Die Differenz des zuletzt negativ und zuletzt positiv ausgewerteten Flankenwertes desselben Messtasters (IndraMotion MLC/IndraLogic XLC nur Messtaster)
- DIFF_POS_TO_POS_EDGE: Die Differenz der beiden zuletzt ausgewerteten positiven Flankenwerte desselben Messtasters (IndraMotion MLC/IndraLogic XLC nur Messtaster)

Programm:

```
TYPE MB_DIFF_EVAL :
(
  DIFF_DISABLED := 0, (* difference measurement is disabled *)
  DIFF_STANDARD_MODE := 1, (* difference of two measured values *)
  DIFF_NEG_TO_NEG_EDGE := 2, (* negative edge to negative edge difference *)
  DIFF_POS_TO_NEG_EDGE := 3, (* positive edge to negative edge difference *)
  DIFF_NEG_TO_POS_EDGE := 4, (* negative edge to positive edge difference *)
  DIFF_POS_TO_POS_EDGE := 5 (* positive edge to positive edge difference *)
);
END_TYPE
```

ML_TechBase.library

TOUCHPROBE_REF

Der folgende Datentyp definiert die für die Steuerung und den Messtaster verwendete Konfiguration:
Programm:

```
TYPE TOUCHPROBE_REF :  
  STRUCT CntrlNo:  
    CONTROLS:= LOCAL_CNTRL;      (* logical control address *)  
    TouchProbeNo: TP_OBJECTS:= NO_TOUCHPROBE;  (* probe number *)  
  END_STRUCT  
END_TYPE
```

5.8 Trigonometrische Funktionen

5.8.1 Einführung

Die nachfolgenden Funktionen erlauben dem Anwender Umrechnungen zwischen unterschiedlichen Einheiten vorzunehmen. Da es sich um Funktionen handelt, können keine Fehler ausgegeben werden. Werden ungültige Werte verwendet, so wird der Wert 0 zurückgeliefert.

5.8.2 ML_DegToInc

Kurzbeschreibung

Die Funktion ML_DegToInc wandelt den in Gradmaß vorgegebenen Winkel ($0^{\circ} \leq x < 360^{\circ}$) in den entsprechenden Inkrementwert ($0 \leq x < 2^{20}$).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-113: Referenztabelle Funktion ML_DegToInc

Schnittstellenbeschreibung

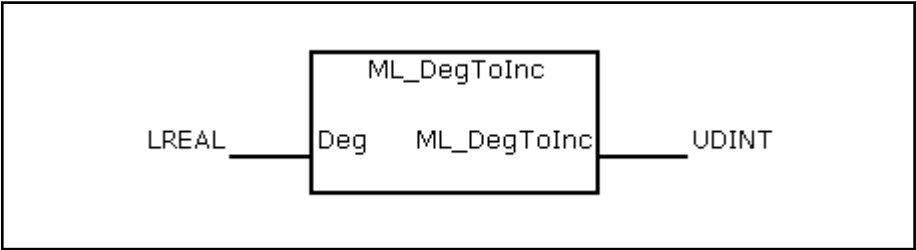


Abb.5-114: Funktion ML_DegToInc

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Deg	LREAL	Winkel in Gradmaß [DEG]
RETURN	ML_DegToInc	UDINT	Winkel in Inkrementen

Abb.5-115: Schnittstellenvariablen Funktion ML_DegToInc

Min.- Max.- und Default-Wert

In der folgenden Tabelle sind der Min-/Max- und Default-Wert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
Deg	LREAL	-2^32 * 360.0	2^32 * 360.0	0.0

Abb.5-116: Min-/Max- und Default-Wert für Funktion ML_DegToInc

Funktionsbeschreibung

Die Funktion ML_DegToInc wandelt den in Gradmaß vorgegebenen Winkel ($0^{\circ} \leq x < 360^{\circ}$) in den entsprechenden Inkrementwert ($0 \leq x < 2^{20}$). Wird ein Winkel größer gleich 360° vorgegeben, wird er auf den entsprechenden Winkel innerhalb des gültigen Wertebereichs umgerechnet (Modulo 360).

Wird ein Wert außerhalb des gültigen Wertebereiches vorgegeben, wird der Wert 0 zurückgeliefert.

Fehlercodes Die Funktion ML_DegToRad generiert keinen Fehler.

5.8.3 ML_DegToRad

Kurzbeschreibung Die Funktion ML_DegToRad konvertiert einen Winkelwert im Gradmaß in das entsprechende Bogenmaß inkl. Modulo.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-117: Referenztable Funktion ML_DegToRad

Schnittstellenbeschreibung

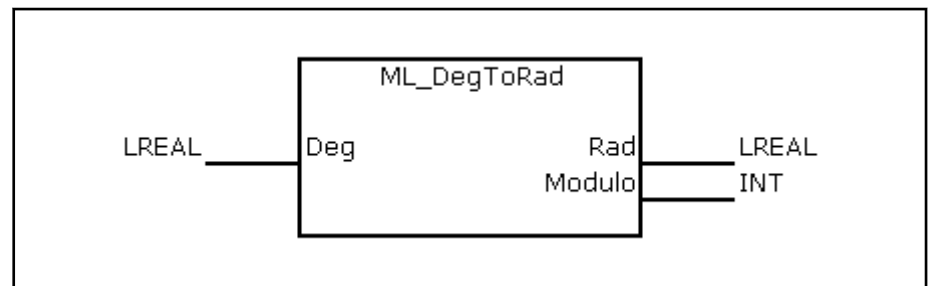


Abb.5-118: Funktion ML_DegToRad

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Deg	LREAL	Winkel [DEG]
VAR_OUTPUT	Rad	LREAL	Winkel [RAD]
	Modulo	INT	Modulo

Abb.5-119: Schnittstellenvariablen Funktion ML_DegToRad

Min.- Max.- und Default-Wert In der folgenden Tabelle sind der Min-/Max- und Default-Wert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
Deg	LREAL	$-2^{15} * 360.0$	$2^{15} * 360.0$	0.0

Abb.5-120: Min-/Max- und Default-Wert für Funktion ML_DegToRad

Funktionsbeschreibung Die Funktion ML_DegToRad konvertiert bei Aufruf der Instanz zyklisch den angelegten Winkel in Gradmaß in das entsprechende Bogenmaß inkl. Modulo.

Wird ein Wert außerhalb des gültigen Wertebereiches vorgegeben, wird der Wert 0 zurückgeliefert.

Fehlercodes Die Funktion ML_DegToRad generiert keinen Fehler.

5.8.4 ML_IncToDeg

Kurzbeschreibung Die Funktion ML_IncToDeg wandelt den in Inkrementen vorgegebenen Winkel ($0 \leq x < 2^{20}$) in den entsprechenden Winkel in Gradmaß ($0^\circ \leq x < 360^\circ$).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-121: Referenztable Funktion ML_IncToDeg

ML_TechBase.library

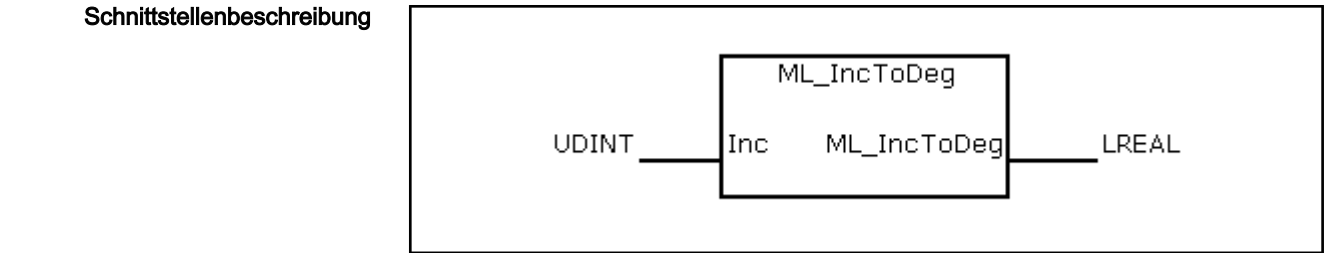


Abb.5-122: Funktion ML_IncToDeg

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Inc	UDINT	Winkel in Inkrementen
RETURN	ML_IncToDeg	LREAL	Winkel in Gradmaß [DEG]

Abb.5-123: Schnittstellenvariablen Funktion ML_IncToDeg

Funktionsbeschreibung

Die Funktion ML_IncToDeg wandelt den in Inkrementen vorgegebenen Winkel ($0 \leq x < 2^{20}$) in den entsprechenden Winkel in Gradmaß ($0^\circ \leq x < 360^\circ$). Wird ein Winkel größer gleich 2^{20} Inkrementen vorgegeben, so wird er auf den entsprechenden Winkel innerhalb des gültigen Wertebereichs umgerechnet (Modulo 2^{20}).

Fehlercodes

Die Funktion ML_IncToDeg generiert keinen Fehler.

5.8.5 ML_IncToRad

Kurzbeschreibung

Die Funktion ML_IncToRad wandelt den in Inkrementen vorgegebenen Winkel ($0 \leq x < 2^{20}$) in den entsprechenden Winkel in Bogenmaß ($0 \leq x < 2 \cdot \pi$).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-124: Referenztabelle Funktion ML_IncToRad

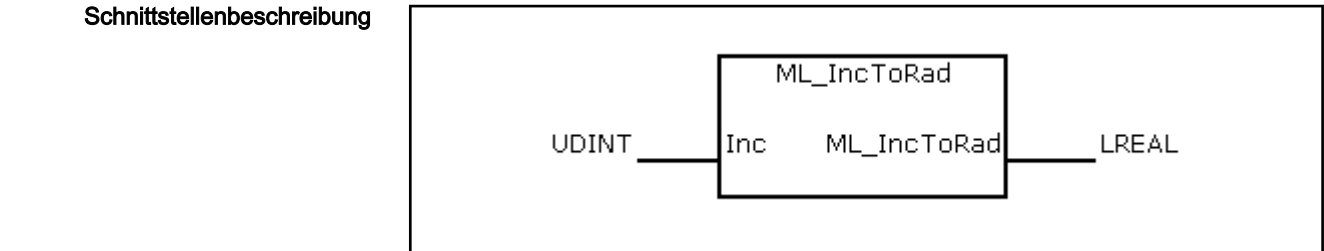


Abb.5-125: Funktion ML_IncToRad

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Inc	UDINT	Winkel in Inkrementen
RETURN	ML_IncToRad	LREAL	Winkel in Bogenmaß [RAD]

Abb.5-126: Schnittstellenvariablen Funktion ML_IncToRad

Funktionsbeschreibung

Die Funktion ML_IncToRad wandelt den in Inkrementen vorgegebenen Winkel ($0 \leq x < 2^{20}$) in den entsprechenden Winkel in Bogenmaß ($0 \leq x < 2 \cdot \pi$). Wird ein Winkel größer gleich 2^{20} Inkrementen vorgegeben, so wird er auf den entsprechenden Winkel innerhalb des gültigen Wertebereichs umgerechnet (Modulo 2^{20}).

Fehlercodes

Die Funktion ML_IncToRad generiert keinen Fehler.

5.8.6 ML_RadToDeg

Kurzbeschreibung

Die Funktion ML_RadToDeg konvertiert einen Winkelwert im Bogenmaß in das entsprechende Gradmaß inkl. Modulo.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-127: Referenztable Funktion ML_RadToDeg

Schnittstellenbeschreibung

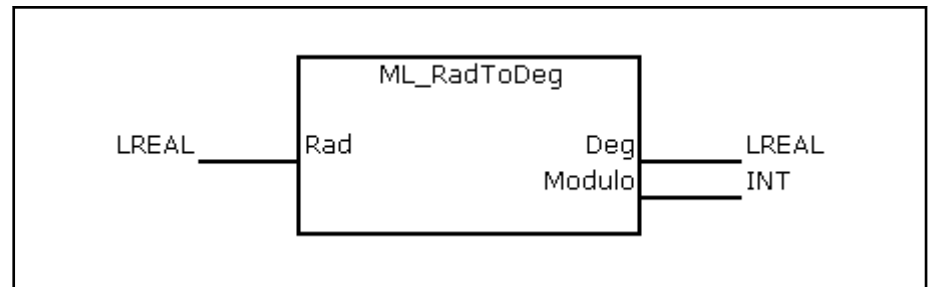


Abb.5-128: Funktion ML_RadToDeg

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Rad	LREAL	Winkel [RAD]
VAR_OUTPUT	Deg	LREAL	Winkel [DEG]
	Modulo	INT	Modulo

Abb.5-129: Schnittstellenvariablen Funktion ML_RadToDeg

Min.- Max.- und Default-Wert

In der folgenden Tabelle sind der Min-/Max- und Default-Wert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
Rad	LREAL	$-2 * \pi * 2^{15}$	$2 * \pi * 2^{15}$	0.0

Abb.5-130: Min-/Max- und Default-Wert für Funktion ML_RadToDeg

Funktionsbeschreibung

Die Funktion ML_RadToDeg konvertiert bei Aufruf der Instanz zyklisch den angelegten Winkel in Bogenmaß in das entsprechende Gradmaß inkl. Modulo.

Wird ein Wert außerhalb des gültigen Wertebereiches vorgegeben, wird der Wert 0 sowohl für "Deg", als auch für "Modulo" zurückgeliefert.

Fehlercodes

Die Funktion ML_RadToDeg generiert keinen Fehler.

5.8.7 ML_RadToInc

Kurzbeschreibung

Die Funktion ML_RadToInc wandelt den in Bogenmaß vorgegebenen Winkel ($0 \leq x < 2 * \pi$) in den entsprechenden Inkrementwert ($0 \leq x < 2^{20}$).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-131: Referenztable Funktion ML_RadToInc

ML_TechBase.library

Schnittstellenbeschreibung

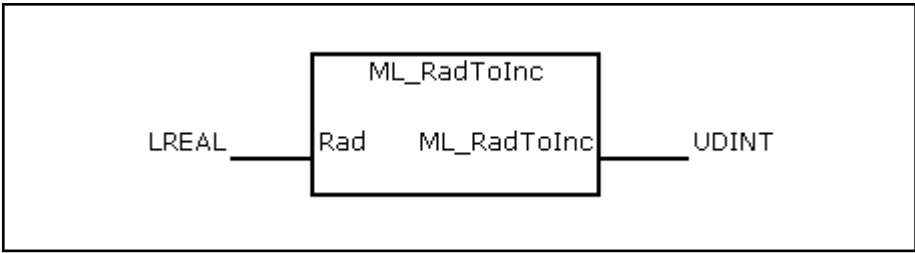


Abb.5-132: Funktion ML_RadToInc

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Rad	LREAL	Winkel in Bogenmaß [RAD]
RETURN	ML_RadToInc	UDINT	Winkel in Inkrementen

Abb.5-133: Schnittstellenvariablen Funktion ML_RadToInc

Min.- Max.- und Default-Wert In der folgenden Tabelle sind der Min-/Max- und der Defaultwert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
Rad	LREAL	$-2 * \pi * 2^{32}$	$2 * \pi * 2^{32}$	0.0

Abb.5-134: Min-/Max- und Default-Wert für Funktion ML_RadToInc

Funktionsbeschreibung Die Funktion ML_RadToInc wandelt den in Bogenmaß vorgegebenen Winkel ($0 \leq x < 2 * \pi$) in den entsprechenden Inkrementwert ($0 \leq x < 2^{20}$). Wird ein Winkel größer gleich $2 * \pi$ vorgegeben, wird er auf den entsprechenden Winkel innerhalb des gültigen Wertebereichs umgerechnet (Modulo $2 * \pi$).

Wird ein Wert außerhalb des gültigen Wertebereiches vorgegeben, wird der Wert 0 zurückgeliefert.

Fehlercodes Die Funktion ML_RadToInc generiert keinen Fehler.

5.9 Funktionsbausteine und Funktionen für die Verwaltung "zyklischer Datenkanäle"

5.9.1 Einführung und Übersicht

Das SERCOS-Interface erlaubt das Auflegen von frei konfigurierbaren zyklischen Daten. Diese werden zyklisch bei jedem SERCOS-Takt sowohl von der Steuerung in die Antriebe als auch von den Antrieben zur Steuerung übertragen. Pro Achse stehen jeweils vier Parameter in Lese- bzw. Schreibrichtung zur Verfügung. Zusätzlich können jeweils vier Bits gelesen oder geschrieben werden. Die Konfiguration der zyklischen Daten erfolgt im Parametriermodus (PM, SERCOS Phase 2) und ist im Betriebsmodus (BB, SERCOS Phase 4) nicht änderbar.

Im Parametriermodus (PM, SERCOS Phase 2) kann durch den Funktionsbaustein MB_AllocCyclicDataParameter ein zyklischer Datenkanal für einen zu übertragenden Parameter angefordert werden. Ist der Parameter bereits im zyklischen Datenkanal aufgelegt, so erfolgt keine Änderung der Konfiguration. Andernfalls wird ein freier Datenkanal gesucht und der entsprechende Parameter aufgelegt.

Bei Änderung der Konfiguration kann ein einmal zyklisch aufgelegter Parameter mit [Kap. 5.9.3 "MB_FreeCyclicParameter" auf Seite 130](#) auch wieder frei gegeben werden. Dadurch steht dieser Kanal wieder für eine neue Konfiguration zur Verfügung. Eine explizite Freigabe ist nur notwendig, wenn die Datenkanäle zur Laufzeit aus der SPS heraus neu konfiguriert werden.

Die Konfiguration kann auch in IndraWorks über den AxisData- bzw. den zyklischen Ist- und Sollwerte- Dialog erfolgen. Eine über IndraWorks festgelegte Konfiguration wird automatisch verwendet. Werden einige Parameter über IndraWorks konfiguriert und später zur Laufzeit weitere Parameter mit [Kap. 5.9.2 "MB_AllocCyclicParameter" auf Seite 128](#) angefordert, werden die neuen Parameter auf freie Kanäle verteilt. Mit [Kap. 5.9.4 "MB_GetCyclicChannelConfig" auf Seite 132](#) kann die aktuelle Konfiguration der zyklisch aufgelegten Parameter ausgelesen werden.

Im Betriebsmodus (BB) sucht der Funktionsbaustein [Kap. 5.9.5 "MB_GetCyclicParameterHandle" auf Seite 133](#) einen zyklisch aufgelegten Parameter in der Konfiguration einer Achse und schreibt alle benötigten Informationen in ein Handle. Mit diesem Handle kann dann in jedem SERCOS-Zyklus durch die [Kap. 5.9.6 "MB_ReadCyclicParameter" auf Seite 135](#) und [Kap. 5.9.8 "MB_WriteCyclicParameter" auf Seite 136](#) vom Parameter gelesen oder auf den Parameter geschrieben werden. Für nachkommastellenbehaftete zyklische Parameter stehen die Funktionen [Kap. 5.9.7 "MB_ReadCyclicRealParameter" auf Seite 136](#) und [Kap. 5.9.9 "MB_WriteCyclicRealParameter" auf Seite 137](#) zur Verfügung. Aufgetretene Fehler während des Zugriffs auf die zyklischen Datenkanäle können über die Funktion [Kap. 5.9.10 "MB_GetLastCyclicParameterError" auf Seite 138](#) abgefragt werden.

Die zyklisch aufgelegten Parameter stehen parallel auch in der AxisData-Struktur der jeweiligen Achse zur Verfügung. Die Istwerte stehen in den UserActualDataX-Einträgen, die Sollwerte werden über UserCmdDataX beschrieben. Hierbei muss jedoch je nach aufgelegtem Parameter der richtige Datentyp beschrieben werden. Dieser Aufwand entfällt bei Verwendung der Funktionsbausteine MB_ReadCyclicParameter und MB_WriteCyclicParameter.

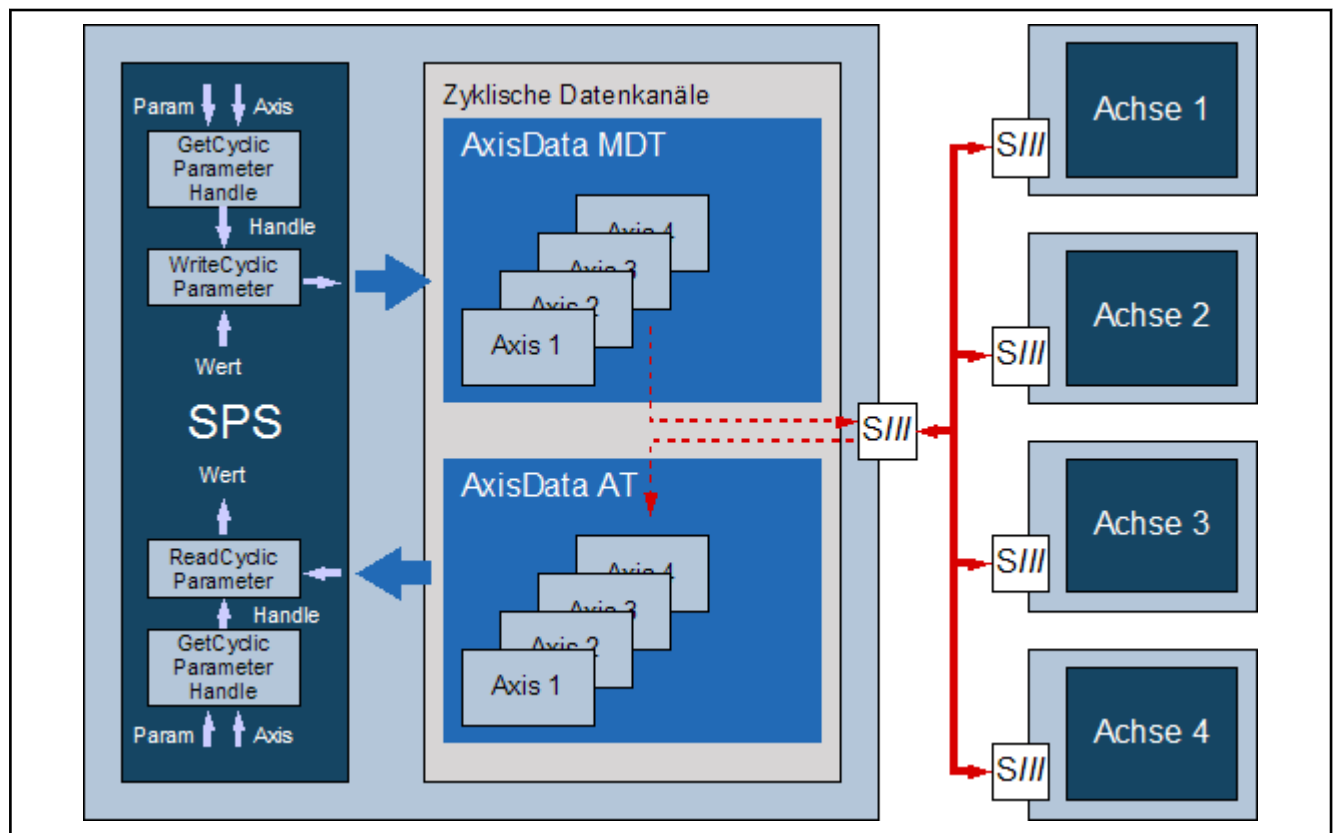


Abb.5-135: Zugriff auf zyklische Daten mit Hilfe der CyclicParameter Funktionsbausteine und Funktionen

ML_TechBase.library

5.9.2 MB_AllocCyclicParameter

Kurzbeschreibung

Dieser Funktionsbaustein überprüft die Konfiguration eines zyklisch zu lesen-
den oder zu beschreibenden Parameters im zyklischen Datenkanal einer
Achse und konfiguriert bei Bedarf den zyklischen Datenkanal. Belegte Para-
meter werden in einer globalen Datenstruktur vermerkt.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx07VRS ohne MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-136: Referenztabelle

Schnittstellenbeschreibung

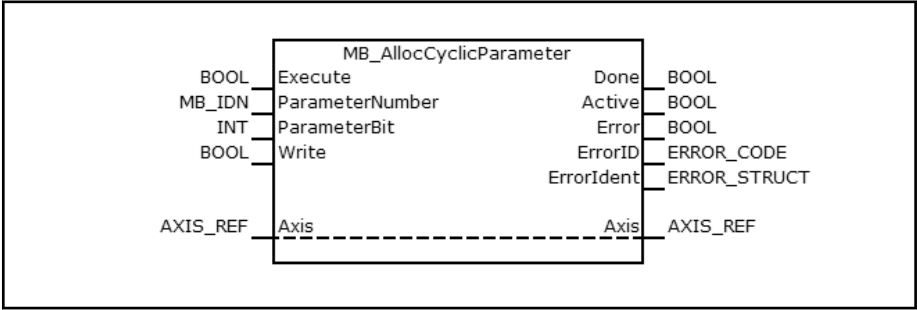


Abb.5-137: Funktionsbaustein MB_AllocCyclicParameter

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Axis	AXIS_REF	Achse des Parameters
VAR_INPUT	Execute	BOOL	Startet die Konfiguration des Parameter
	ParameterNumber	MB_IDN	Aufzulegender Parameter
	ParameterBit	INT	Bit innerhalb des Parameters; "CYCLIC_ALLBITS" für gesamten Parameter
	Write	BOOL	FALSE: Parameter lesen TRUE: Parameter schreiben
VAR_OUTPUT	Done	BOOL	Parameter wurde aufgelegt
	Active	BOOL	Der Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Grundsätzliche Fehlerursache
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe "Fehlercodes" auf Seite 129

Abb.5-138: Schnittstellenvariablen Funktionsbaustein MB_AllocCyclicParameter

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
ParameterNum- ber	DINT	0	n.def.	0	steigende Flanke an Execute
ParameterBit	INT	-1	31	CYCLIC_ALL-BITS	steigende Flanke an Execute
Write	BOOL			FALSE	steigende Flanke an Execute

Abb.5-139: Eingangsverhalten MB_AllocCyclicParameter

Funktionsbeschreibung

Dieser Funktionsbaustein arbeitet grundsätzlich nur in SERCOS-Phase 2. Bei steigender Flanke an "Execute" werden die Eingänge "ParameterNum-ber", "ParameterBit" und "Write" übernommen. Anhand des Eingangs "Write" wird zwischen den Istdaten (UserActualData) und den Sollwerten (UserCmd-Data) unterschieden. "ParameterBit" gibt die Bitnummer innerhalb des Parameters an. Liegt "ParameterBit" zwischen 0 und 31, so wird ein Bit-Datenkanal verwendet. Ist "ParameterBit" gleich "CYCLIC_ALLBITS", so wird der gesamte Parameter aufgelegt. "ParameterNumber" gibt den aufzulegenden Parameter an.

MB_AllocCyclicParameter überprüft im ersten Schritt, ob der Parameter sich eventuell schon in der aktuellen Konfiguration befindet. Ist dies nicht der Fall, wird ein freier zyklischer Kanal gesucht und der Parameter dort aufgelegt. Anschließend wird die Belegung in einer internen statischen Variable gespeichert. Wird in "ParameterBit" eine Bitnummer angegeben, wird vorzugsweise ein freier Bitkanal verwendet. Sind alle Bitkanäle belegt, wird ein Parameterkanal verwendet. Nach erfolgreichem Abschluss der Konfiguration des zyklischen Datenkanals wird der "Done"-Ausgang für mindestens einen Zyklus auf TRUE gesetzt. "Done" bleibt TRUE, solange der Eingang „Execute“ aktiv ist. Steht kein freier zyklischer Kanal zur Verfügung, wird eine entsprechende Fehlermeldung ausgegeben.

Fehlerverhalten

Kann der Parameter nicht im zyklischen Datenkanal aufgelegt werden oder tritt ein anderer Fehler auf, ist "Error" TRUE.

Fehlercodes

Der Funktionsblock generiert die folgenden Fehlermeldungen der "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#000D	AxisNo	AxisRef liegt außerhalb des gültigen Bereichs
ACCESS_ERROR	16#0010	16#0000	SERCOS-Phase 2 ist nicht aktiv
ACCESS_ERROR	16#1504	16#0000	AxisData-Struktur nicht aktiviert
INPUT_INVALID_ERROR	16#F212	16#0000	Parametertyp Struktur nicht unterstützt
INPUT_INVALID_ERROR	16#F213	16#0000	Parametertyp Funktion nicht unterstützt
INPUT_INVALID_ERROR	16#F214	16#0000	Parametertyp nicht unterstützt
INPUT_INVALID_ERROR	16#F215	16#0000	Parameter ist nicht beschreibbar

ML_TechBase.library

ErrorID	Additional1	Additional2	Beschreibung
RESSOURCE_ERROR	16#F216	16#0000	Kein freier zyklischer Kanal
INPUT_INVALID_ERROR	16#F217	16#0000	Parameter nicht im zyklischen Datenkanal konfigurierbar

Abb.5-140: Fehlercodes Funktionsbaustein MB_AllocCyclicParameter

5.9.3 MB_FreeCyclicParameter

Kurzbeschreibung Dieser Funktionsbaustein gibt einen bereits konfigurierten zyklisch zu lesen-
den oder zu beschreibenden Parameter im zyklischen Datenkanal einer Ach-
se wieder frei. Die Freigabe des Parameters wird in einer globalen Struktur
vermerkt.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx07VRS ohne MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-141: Referenztabelle

Schnittstellenbeschreibung

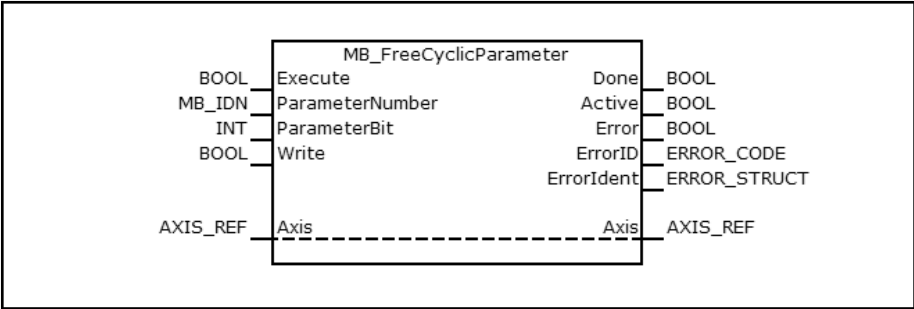


Abb.5-142: Funktionsbaustein MB_FreeCyclicParameter

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Axis	AXIS_REF	Achse des Parameters
VAR_INPUT	Execute	BOOL	Startet die Freigabe des Parameters
	ParameterNumber	MB_IDN	Nummer des freizugebenden Parameters
	ParameterBit	INT	Bit innerhalb des Parameters; "CYCLIC_ALLBITS" für gesamten Parameter
	Write	BOOL	FALSE: Parameter lesen TRUE: Parameter schreiben

E/A-Typ	Name	Datentyp	Beschreibung
VAR_OUTPUT	Done	BOOL	Parameter wurde aufgelegt
	Active	BOOL	Der Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Grundsätzliche Fehlerursache
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe "Fehlercodes" auf Seite 131

Abb.5-143: Schnittstellenvariablen Funktionsbaustein MB_FreeCyclicParameter

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
ParameterNum- ber	DINT	0	n.def.	0	steigende Flanke an Execute
ParameterBit	INT	-1	31	CYCLIC_ALL-BITS	steigende Flanke an Execute
Write	BOOL			FALSE	steigende Flanke an Execute

Abb.5-144: Eingangsverhalten Funktionsbaustein MB_FreeCyclicParameter

Funktionsbeschreibung

Dieser Funktionsbaustein arbeitet grundsätzlich nur in SERCOS-Phase 2. Bei steigender Flanke an "Execute" werden die Eingänge "ParameterNum-ber", "ParameterBit" und "Write" übernommen. Anhand des Eingangs "Write" wird zwischen den Istdaten (UserActualData) und den Sollwerten (UserCmd-Data) unterschieden. "ParameterBit" gibt die Bitnummer innerhalb des Parameters an. Liegt "ParameterBit" zwischen 0 und 31, wird ein Bit-Datenkanal verwendet. Ist "ParameterBit" gleich "CYCLIC_ALLBITS", wird der gesamte Parameter aufgelegt. "ParameterNumber" gibt den freizugebenden Parameter an.

MB_FreeCyclicParameter überprüft im ersten Schritt, ob der Parameter sich in der aktuellen Konfiguration befindet. Ist dies nicht der Fall, wird eine entsprechende Fehlermeldung ausgegeben. Anschließend wird die Freigabe in einer internen statischen Variable gespeichert. MB_FreeCyclicParameter ändert die eigentliche Konfiguration der zyklischen Datenkanäle nicht. Die Freigabe wird lediglich in der internen Kontrollstruktur vermerkt. Nach erfolgreichem Abschluss der Freigabe des zyklischen Datenkanals wird der "Done"-Ausgang für mindestens einen Zyklus auf TRUE gesetzt. "Done" bleibt TRUE, solange der Eingang "Execute" aktiv ist.

Fehlerverhalten

Ist der Parameter nicht im zyklischen Datenkanal aufgelegt oder tritt ein anderer Fehler auf, ist "Error" TRUE.

Fehlercodes

Der Funktionsblock generiert die folgenden Fehlermeldungen der "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#000D	16#0000	AxisRef liegt außerhalb des gültigen Bereichs
ACCESS_ERROR	16#0010	16#0000	SERCOS-Phase 2 ist nicht aktiv

ML_TechBase.library

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR	16#1504	16#0000	AxisData-Struktur nicht aktiviert
ACCESS_ERROR	16#F211	16#0000	Parameter nicht im zyklischen Datenkanal gefunden

Abb.5-145: Fehlercodes Funktionsbaustein MB_FreeCyclicParameter

5.9.4 MB_GetCyclicChannelConfig

Kurzbeschreibung

Der Funktionsbaustein MB_GetCyclicChannelConfig liest die Konfiguration der zyklischen Datenkanäle einer Achse. Es werden sowohl die Lese-, als auch die Schreibkanäle ausgelesen und in einer Datenstruktur abgelegt. Die Anzahl der Einträge ist geräteabhängig und wird in einem gesonderten Eintrag ausgegeben.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-146: Referenztabelle Funktionsbaustein MB_GetCyclicChannelConfig

Schnittstellenbeschreibung

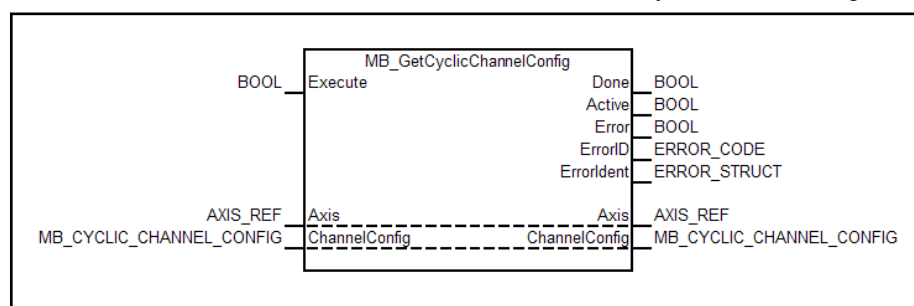


Abb.5-147: Funktionsbaustein MB_GetCyclicChannelConfig

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Axis	AXIS_REF	Achse des Parameters
	ChannelConfig	MB_CYCLIC_CHANNEL_CONFIG	Ausgelesene Konfiguration, siehe Kap. 5.9.14 "MB_CYCLIC_CHANNEL_CONFIG" auf Seite 140
VAR_INPUT	Execute	BOOL	Startet die Konfiguration des Parameter
VAR_OUTPUT	Done	BOOL	Parameter wurde aufgelegt
	Active	BOOL	Der Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Grundsätzliche Fehlerursache
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe "Fehlercodes" auf Seite 133

Abb.5-148: Schnittstellenvariablen Funktionsbaustein MB_GetCyclicChannelConfig

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich

Abb.5-149: Eingangsverhalten MB_GetCyclicChannelConfig

Funktionsbeschreibung

Dieser Funktionsbaustein arbeitet grundsätzlich nur in den SERCOS-Phasen 2 und 4. Zyklische Datenkanäle sind nur bei realen Achsen vorhanden. Demzufolge kann der Baustein nur für reale Achsen verwendet werden. Bei steig-

ender Flanke an Execute wird die angelegte Achse überprüft und die Konfiguration der zyklischen Datenkanäle ausgelesen. Diese Operation dauert mehrere Zyklen. Während dieser Zeit ist der Ausgang "Active" gesetzt. Nach erfolgreichem Abschluss des Auslesevorgangs wird "Active" zurückgesetzt und "Done" gesetzt. Die Konfiguration steht jetzt am Ausgang „ChannelConfig“ zur Verfügung. Der Ausgang "ChannelConfig" ist nur nach gesetztem "Done"-Ausgang gültig.

Die Anzahl der verfügbaren Datenkanäle ist von der Achse abhängig und ist in den "...Count" Elementen der "ChannelConfig"-Struktur hinterlegt. Die Elemente "ReadChannel", "WriteChannel", "ReadBitChannel" und "WriteBitChannel" enthalten die aufgelegten Parameternummern. Ist kein Parameter aufgelegt, so ist 0 eingetragen. Die Elemente "ReadBitChannelBit" und "WriteBitChannelBit" enthalten die aktive Bitnummer des Parameters.

Fehlerverhalten Tritt während der Abarbeitung ein Fehler auf, werden die Ausgänge "Active" und "Done" zurückgesetzt und "Error" gesetzt. "ErrorID" und "ErrorIdent" enthalten bei gesetztem "Error"-Ausgang nähere Informationen über den Fehler.

Fehlercodes Der Funktionsblock generiert die folgenden Fehlermeldungen der "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#000D	16#0000	AxisRef liegt außerhalb des gültigen Bereichs

Abb.5-150: Fehlercodes Funktionsbaustein MB_GetCyclicChannelConfig

5.9.5 MB_GetCyclicParameterHandle

Kurzbeschreibung Dieser Funktionsbaustein sucht einen zyklisch zu lesenden oder zu beschreibenden Parameter im zyklischen Datenkanal einer Achse und legt alle benötigten Informationen in einem Handle ab.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx07VRS ohne MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-151: Referenztablelle

Schnittstellenbeschreibung

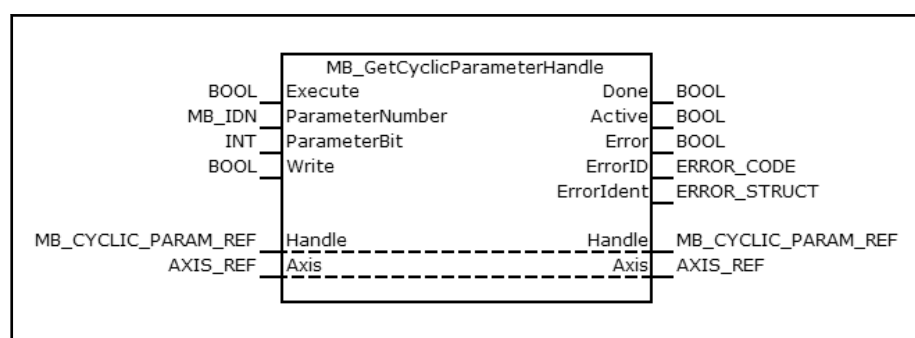


Abb.5-152: Funktionsbaustein MB_GetCyclicParameterHandle

ML_TechBase.library

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Handle	MB_CYCLIC_PA- RAM_REF	enthält Informationen für den Zugriff auf den zyklischen Parameter
	Axis	AXIS_REF	Achse des Parameters
VAR_INPUT	Execute	BOOL	startet die Suche nach dem Parameter
	ParameterNumber	MB_IDN	Nummer des zyklisch aufgelegten Parameters
	ParameterBit	INT	Bit innerhalb des Parameters; "CYCLIC_ALLBITS" für den gesamten Parameter
	Write	BOOL	FALSE: Parameter lesen TRUE: Parameter schreiben
VAR_OUTPUT	Done	BOOL	Parameter wurde gefunden und "Handle" ist gültig
	Active	BOOL	der Funktionsbaustein ist aktiv
	Error	BOOL	zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	grundsätzliche Fehlerursache
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe "Fehlercodes" auf Seite 134

Abb.5-153: Schnittstellenvariablen Funktionsbaustein MB_GetCyclicParameterHandle

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
ParameterNumber	DINT	0	n.def.	0	steigende Flanke an Execute
ParameterBit	INT	-1	31	CYCLIC_ALLBITS	steigende Flanke an Execute
Write	BOOL			FALSE	steigende Flanke an Execute

Abb.5-154: Eingangsverhalten MB_GetCyclicParameterHandle

Funktionsbeschreibung

Dieser Funktionsbaustein wird im Betriebsmodus (SERCOS Phase 4) verwendet. Nach einer steigenden Flanke an "Execute" werden die Eingänge "ParameterNumber", "ParameterBit" und "Write" gelesen. Der "Write"-Eingang entscheidet, ob Sollwerte des MDTs oder Istwerte des ATs gesucht werden. Ist "ParameterBit" "CYCLIC_ALLBITS", wird der komplette Parameter aufgelegt. Wird in "ParameterBit" eine Bitnummer angegeben, wird erst in dem Bitkanal gesucht. Wurde der Parameter nicht in den Bitkanälen gefunden, und handelt es sich um einen Istwert, werden die Parameterkanäle durchsucht. Nach erfolgreichem Abschluss der Bearbeitung ist "Done" aktiv und die Daten im angelegten "Handle" sind gültig.

Fehlerverhalten

Wird der Parameter nicht im zyklischen Datenkanal gefunden oder tritt ein anderer Fehler auf, ist "Error" TRUE und "Handle" wird nicht verändert.

Fehlercodes

Der Funktionsblock generiert die folgenden Fehlermeldungen der "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#000D	16#0000	AxisRef liegt außerhalb des gültigen Bereichs
ACCESS_ERROR	16#1504	16#0000	AxisData-Struktur nicht aktiviert
ACCESS_ERROR	16#F211	16#0000	Parameter nicht im zyklischen Datenkanal gefunden
INPUT_INVALID_ERROR	16#F212	16#0000	Parametertyp Struktur nicht unterstützt
INPUT_INVALID_ERROR	16#F213	16#0000	Parametertyp Funktion nicht unterstützt
INPUT_INVALID_ERROR	16#F214	16#0000	Parametertyp nicht unterstützt
INPUT_INVALID_ERROR	16#F215	16#0000	Parameter ist nicht beschreibbar

Abb.5-155: Fehlercodes Funktionsbaustein MB_GetCyclicParameterHandle

5.9.6 MB_ReadCyclicParameter

Kurzbeschreibung

Diese Funktion liest einen zyklischen Parameter über den "Handle" und gibt den aktuellen Inhalt als DINT zurück. Nachkommastellenbehaftete Werte werden entsprechend der im SERCOS-Attribut abgelegten Wichtung gewandelt. Diese Funktion kann auch für Bit-Werte verwendet werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx07VRS ohne MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-156: Referenztafel

Schnittstellenbeschreibung

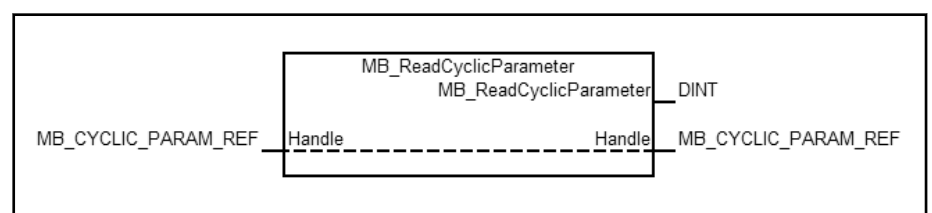


Abb.5-157: Funktion MB_ReadCyclicParameter

Eingangsvariable	Datentyp	Beschreibung
Handle	MB_CYCLIC_PARAM_REF	Enthält Informationen für den Zugriff auf den zyklischen Parameter
Rückgabewert	Datentyp	Beschreibung
MB_ReadCyclicParameter	DINT	Wert des Parameters

Abb.5-158: Schnittstellenvariablen Funktion MB_ReadCyclicParameter

ML_TechBase.library

5.9.7MB_ReadCyclicRealParameter

Kurzbeschreibung Diese Funktion liest einen zyklischen Parameter über den "Handle" und gibt den aktuellen Inhalt als REAL zurück. Ist der Datentyp des Parameters nicht REAL, findet eine Typwandlung statt. Diese Funktion kann nicht für Bit-Werte verwendet werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx07VRS ohne MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-159: Referenztabelle

Schnittstellenbeschreibung

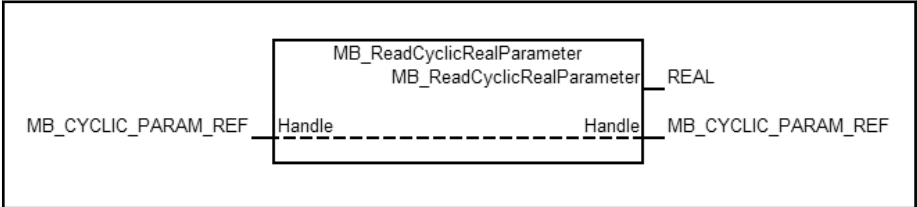


Abb.5-160: Funktion MB_ReadCyclicRealParameter

Eingangsvariable	Datentyp	Beschreibung
Handle	MB_CYCLIC_PARAM_REF	Enthält Informationen für den Zugriff auf den zyklischen Parameter
Rückgabewert	Datentyp	Beschreibung
MB_ReadCyclicRealParameter	REAL	Wert des Parameters

Abb.5-161: Schnittstellenvariablen Funktion MB_ReadCyclicRealParameter

5.9.8MB_WriteCyclicParameter

Kurzbeschreibung Diese Funktion schreibt einen zyklischen Parameter über den "Handle". Ist der Zieldatentyp nachkommastellenbehaftet, wird der Wert entsprechend der im SERCOS-Attribut angegebenen Nachkommastellen gewandelt. Diese Funktion kann auch für Bit-Werte verwendet werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx07VRS ohne MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-162: Referenztabelle

Schnittstellenbeschreibung

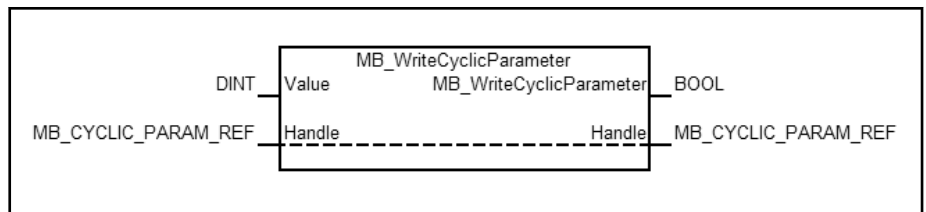


Abb.5-163: Funktion MB_WriteCyclicParameter

Eingangsvariable	Typ	Beschreibung
Handle	MB_CYCLIC_PARAM_REF	Enthält Informationen für den Zugriff auf den zyklischen Parameter
Value	DINT	Zu schreibender Wert
Rückgabewert	Typ	Beschreibung
MB_WriteCyclicParameter	BOOL	Dummy Ausgang ohne Bedeutung; ist immer TRUE

Abb.5-164: Schnittstellenvariablen Funktion MB_WriteCyclicParameter

5.9.9 MB_WriteCyclicRealParameter

Kurzbeschreibung

Diese Funktion schreibt einen zyklischen Parameter über den "Handle". Bei ganzzahligem Zieldatentyp ohne Nachkommastellen wird gerundet. Diese Funktion kann nicht für Bit-Werte verwendet werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx07VRS ohne MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-165: Referenztablelle

Schnittstellenbeschreibung

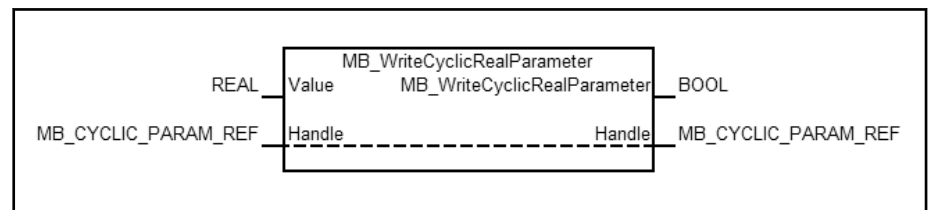


Abb.5-166: Funktion MB_WriteCyclicRealParameter

Eingangsvariable	Typ	Beschreibung
Handle	MB_CYCLIC_PARAM_REF	enthält Informationen für den Zugriff auf den zyklischen Parameter
Value	REAL	zu schreibender Wert

ML_TechBase.library

Eingangsvariable	Typ	Beschreibung
Rückgabewert	Typ	Beschreibung
MB_WriteCyclicRealParameter	BOOL	Dummy-Ausgang ohne Bedeutung; ist immer TRUE

Abb.5-167: Schnittstellenvariablen Funktion MB_WriteCyclicRealParameter

5.9.10 MB_GetLastCyclicParameterError

Kurzbeschreibung Diese Funktion gibt den letzten aufgetretenen Fehler beim Lesen oder Schreiben eines zyklischen Wertes für einen "Handle" zurück und löscht den Fehler.

Diese Funktion ist optional und muss nicht aufgerufen werden. Es wird jedoch empfohlen die zyklische Datenkommunikation in regelmäßigen Abständen auf aufgetretene Fehler zu prüfen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx07VRS ohne MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-168: Referenztabelle

Schnittstellenbeschreibung

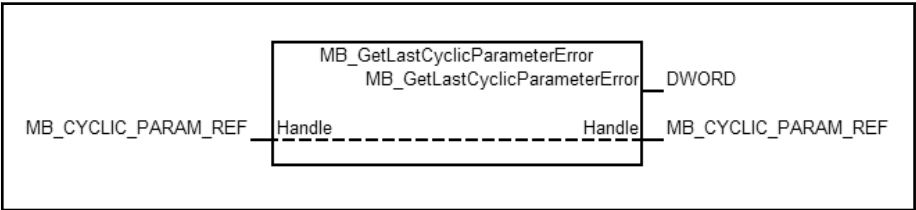


Abb.5-169: Funktion MB_GetLastCyclicParameterError

Eingangsvariable	Datentyp	Beschreibung
Handle	MB_CYCLIC_PARAM_REF	Enthält Informationen für den Zugriff auf den zyklischen Parameter
Rückgabewert	Datentyp	Beschreibung
MB_GetLastCyclicParameterError	DWORD	Fehlercode der letzten lesenden oder schreibenden Aktion

Abb.5-170: Schnittstellenvariablen Funktion MB_GetLastCyclicParameterError

Fehlercodes Die Funktion MB_GetLastCyclicParameterError generiert die folgenden Fehler:

ErrorID	Beschreibung
16#0001	kann Parameter nicht schreiben - Parameter nur zum Lesen konfiguriert
16#0002	falscher Typ für Bit-Zugriff
16#0003	falscher Typ für Parameter-Zugriff

ErrorID	Beschreibung
16#0004	ungültige Achsnummer
16#0005	ungültiger Datenkanal
16#0006	ungültiger Handle

Abb.5-171: Fehlercodes Funktion *GetLastCyclicParameterError*

5.9.11 MB_CYCLIC_PARAM_REF

Typdefinitionen Die folgende Definition beschreibt Datentypen, die im Funktionsbaustein [Kap. 5.9.5 "MB_GetCyclicParameterHandle"](#) auf Seite 133 verwendet werden.

Die Typ-Definitionen sind in der Bibliothek enthalten.

Die Struktur "MB_CYCLIC_PARAM_REF" darf gelesen, jedoch nicht beschrieben werden. Eine Veränderung des Inhalts kann zu Fehlfunktionen führen. Der Aufbau der Struktur kann sich in späteren Versionen ändern. Der Inhalt sollten nur zu Debug-Zwecken gelesen werden.

Die Struktur "MB_CYCLIC_PARAM_REF" sieht wie folgt aus:

Strukturelement	Typ	Beschreibung
ParameterNumber	MB_IDN	FP-Nummer des Parameters
AxisNo	OBJECTS	Logische Nummer der Achse
ParamType	MB_CYCLIC_TYPES	Datentyp des Parameters
Bit	INT	Bit-Nummer innerhalb des Parameters
AccessChannel	MB_CYCLIC_CHANNELS	Datenkanal in dem der Parameter gefunden wurde
ParamAdr	POINTER TO SM_TYPES	Zeiger auf den zyklischen Parameter
BitParamAdr	POINTER TO WORD	Zeiger auf den zyklischen Bit-Parameter
Factor	DINT	Wird zum Umrechnen von REAL zu DINT verwendet
ErrorCode	DWORD	Fehlercode des letzten aufgetretenen Fehlers, siehe "Fehlercodes" auf Seite 138

Abb.5-172: Definition der Basiselemente der MB_CYCLIC_PARAM_REF-Struktur

5.9.12 MB_CYCLIC_TYPES

Der Datentyp "MB_CYCLIC_TYPES" ist ein Aufzählungstyp und hat die folgenden Einträge:

Element	Wert
tINT	1
tUINT	2
tDINT	3
tUDINT	4
tREAL	5
tBIT	6

Abb.5-173: Definition des Aufzählungstyps MB_CYCLIC_TYPES

ML_TechBase.library

5.9.13 MB_CYCLIC_CHANNELS

Der Datentyp "MB_CYCLIC_CHANNELS" ist ein Aufzählungstyp und hat die folgenden Einträge:

Element	Wert
UserCmdDataA	0
UserCmdDataB	1
UserCmdDataC	2
UserCmdDataD	3
UserCmdDataBitA	32
UserCmdDataBitB	33
UserCmdDataBitC	34
UserCmdDataBitD	35
UserActualDataA	256
UserActualDataB	257
UserActualDataC	258
UserActualDataD	259
UserActualDataBitA	288
UserActualDataBitB	289
UserActualDataBitC	290
UserActualDataBitD	291

Abb.5-174: Definition des Aufzählungstyps MB_CYCLIC_CHANNELS

5.9.14 MB_CYCLIC_CHANNEL_CONFIG

Die Struktur "MB_CYCLIC_CHANNEL_CONFIG" wird vom Funktionsbaustein [Kap. 5.9.4 "MB_GetCyclicChannelConfig" auf Seite 132](#) beschrieben und enthält die aufgelegten Parameter im zyklischen Datenkanal der Achse.

Die Struktur "MB_CYCLIC_CHANNEL_CONFIG" sieht wie folgt aus:

Strukturelement	Typ	Min-Wert	Max-Wert	Default-Wert	Beschreibung
ReadChannelCount	UINT	0	10	0	Anzahl der Lesekanäle
ReadChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Konfigurierte Parameter der Lesekanäle
WriteChannelCount	UINT	0	10	0	Anzahl der Schreibkanäle
WriteChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Konfigurierte Parameter der Schreibkanäle
ReadBitChannelCount	UINT	0	10	0	Anzahl der Bit-Lesekanäle
ReadBitChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Konfigurierte Parameter der Bit-Lesekanäle
ReadBitChannelBit	ARRAY[1..10] OF UINT	0	63	0	Konfigurierte Bits der Bit-Lesekanäle

Strukturelement	Typ	Min-Wert	Max-Wert	Default-Wert	Beschreibung
WriteBitChannelCount	UINT	0	10	0	Anzahl der Bit-Schreibkanäle
WriteBitChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Konfigurierte Parameter der Bit-Schreibkanäle
WriteBitChannelBit	ARRAY[1..10] OF UINT	0	63	0	Konfigurierte Bits der Bit-Schreibkanäle

Abb.5-175: Definition der Struktur MB_CYCLIC_CHANNEL_CONFIG

5.10 Funktionsbausteine zum Sichern und Wiederherstellen von Antriebsparametern

5.10.1 Einführung und Übersicht

Antriebe werden normalerweise über IndraWorks parametrieren. An laufenden Anlagen besteht jedoch häufig die Notwendigkeit, Antriebe ohne die Verwendung von IndraWorks zu tauschen und danach neu parametrieren zu können.

Dieser Anwendungsfall wird durch zwei Funktionsbausteine abgedeckt. ML_AxisBackup erstellt eine Parametersicherung einer Achse und legt sie als textbasierte Datei auf der Kompakt-Flash-Karte der Steuerung ab. ML_AxisRestore spielt diese Parametersicherung wieder auf der Achse ein.

Die Wiederherstellung von Parametern erfordert, dass sich der Antrieb im Parametriermodus (P2) befindet. Dies wird auch beim Backup von Parametern empfohlen.

5.10.2 ML_AxisBackup

Kurzbeschreibung

Der Funktionsbaustein ML_AxisBackup sichert die Parameter einer Achse in einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-176: Referenztabelle Funktionsbaustein ML_AxisBackup

Schnittstellenbeschreibung

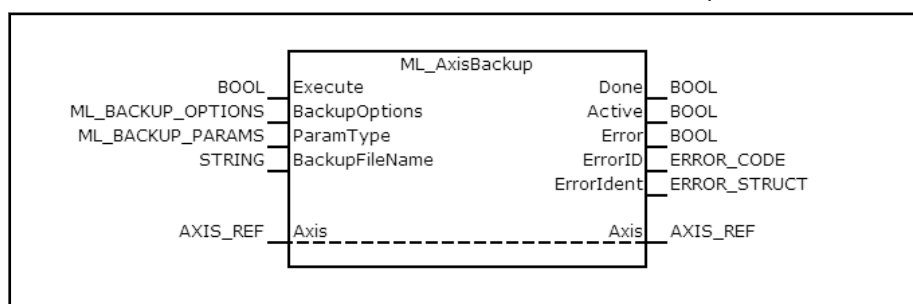


Abb.5-177: Funktionsbaustein ML_AxisBackup

ML_TechBase.library

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (flankengesteuert)
	BackupOptions	ML_BACKUP_OPTIONS	Umfang der Parametersicherung
	ParamType	ML_BACKUP_PARAMS	Auswahl der zu sichernden Parameter (alle, A, S und P)
	BackupFileName	STRING	Dateiname auf der Kompakt-Flash-Karte der Steuerung
VAR_OUTPUT	Done	BOOL	Bearbeitung abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Fehlerdiagnose (siehe nächste Tabelle Fehlercodes , Seite 143)
VAR_IN_OUT	Axis	AXIS_REF	Referenz auf die Achse

Abb.5-178: Schnittstellenvariablen Funktionsbaustein ML_AxisBackup

Min.- Max.- und Default-Werte der Eingänge

Die nachfolgenden Wertebereiche und Defaultwerte können an den Baustein angelegt werden:

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
BackupOptions	ML_BACKUP_OPTIONS			TO_SAVE_PARAMS	Steigende Flanke an Execute
ParamType	ML_BACKUP_PARAMS			ASP_PARAMS	Steigende Flanke an Execute
BackupFileName	STRING			"	Steigende Flanke an Execute

Abb.5-179: Min-/Max- und Defaultwerte Funktionsbaustein ML_AxisBackup

Funktionsbeschreibung

Der Funktionsbaustein ML_AxisBackup sichert die Parameter einer Achse in einer textbasierten Datei auf der User-Partition der Kompakt-Flash-Karte der Steuerung. Ist die Datei bereits vorhanden, so wird sie überschrieben. Die Datei kann vom Funktionsbaustein IL_AxisRestore und von IndraWorks wieder gelesen werden.

BackupOptions

Über den Eingang "BackupOptions" werden die zu sichernden Parameter ausgewählt. Die folgenden Auswahlmöglichkeiten sind vorhanden:

- ALL_PARAM: Es werden alle Parameter gesichert. Diese Option wird vor allem zum überprüfen der Konfiguration benötigt
- MODIFIED_PARAM: Es werden die von den Default-Werten abweichenden Parameter gesichert
- TO_SAVE_PARAM: Es werden die für eine Wiederherstellung notwendigen Parameter gesichert. Dies ist die Voreinstellung



Der Eingang "BackupOptions" wird derzeit nur für S- und P-Parameter ausgewertet. Beim Export von A-Parametern werden immer alle Parameter exportiert.

ParamType

Über den Eingang "ParamType" werden die zu sichernden Parametertypen ausgewählt. Die folgenden Optionen stehen zur Wahl:

- ASP_PARAMS: Parameter im Antrieb und steuerungsinterne Achsparameter
- A_PARAMS: steuerungsinterne Achsparameter
- SP_PARAMS: Parameter im Antrieb

BackupFileName

Über "BackupFileName" kann der Name der Datei auf der Kompakt-Flash-Karte der Steuerung angegeben werden. Es ist möglich, eine Pfadangabe zu verwenden. Der Pfad muss dann aber auf der Kompakt-Flash-Karte ansprechbar sein.

Beispiel:

- "Antrieb1.par" (legt die Datei "Antrieb1.par" auf der User-Partition der Kompakt-Flash-Karte ab)
- "Params/Antrieb1.par" (legt die Datei "Antrieb1.par" im Verzeichnis "Params" auf der User-Partition der Kompakt-Flash-Karte ab)
- "/ata0a/Antrieb2.par" (legt die Datei "Antrieb2.par" auf der OEM-Partition der Steuerung ab)

Zeitverhalten

Der Funktionsbaustein ML_AxisBackup arbeitet asynchron. Nach dem Start durch eine steigende Flanke am Eingang "Execute" liest er die Eingänge "BackupOptions", "ParamType" und "BackupFileName". Solange der Funktionsbaustein arbeitet, ist der Ausgang "Active" gesetzt. Nach erfolgreichem Abschluss der Operation ist der Ausgang "Done" gesetzt. Tritt ein Fehler auf, wird der Ausgang "Error" gesetzt.

Fehlercodes

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#0000000D	16#00000000	AxisRef liegt außerhalb des gültigen Bereiches
INPUT_RANGE_ERROR	16#00001E00	16#00000000	Ungültiger Eingang "BackupOptions"
INPUT_RANGE_ERROR	16#00001E01	16#00000000	Ungültiger Eingang "ParamType"
RESOURCE_ERROR	16#00001E02	16#00000000	Die in "ParamType" ausgewählte Parameterart wird von dieser Achse nicht unterstützt
RESOURCE_ERROR	16#00001E03	16#xxxxxxx	Fehler beim Lesen eines Parameters
ACCESS_ERROR	16#00001E10	16#00000000	Fehler beim Anlegen der Backup-Datei
ACCESS_ERROR	16#00001E12	16#00000000	Fehler beim Schreiben der Backup-Datei

ML_TechBase.library

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR	16#00001E14	16#00000000	Fehler beim Schließen der Backup-Datei
STATE_MACHINE_ERROR	16#00000006	16#0000FFFx	Unerlaubter Zustand des Funktionsbausteins

Abb.5-180: Fehlercodes Funktionsbaustein ML_AxisBackup

5.10.3 ML_AxisRestore

Kurzbeschreibung

Der Funktionsbaustein ML_AxisRestore lädt die Parameter aus einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung auf eine Achse.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-181: Referenztablelle Funktionsbaustein ML_AxisRestore

Schnittstellenbeschreibung

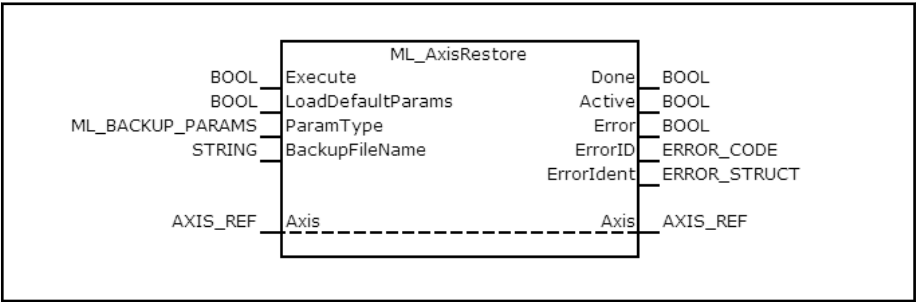


Abb.5-182: Funktionsbaustein ML_AxisRestore

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (flankengesteuert)
	LoadDefaultParams	BOOL	Default-Parameter laden, bevor die Parameter aus der Datei geladen werden
	ParamType	ML_BACKUP_PARAMS	Auswahl der zu wiederherzustellenden Parameter (alle, A, S und P)
	BackupFileName	STRING	Dateiname auf der Flash-Karte der Steuerung
VAR_OUTPUT	Done	BOOL	Bearbeitung abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Fehlerdiagnose (siehe Tabelle Fehlercodes, Seite 146).
VAR_IN_OUT	Axis	AXIS_REF	Referenz auf die Achse

Abb.5-183: Schnittstellenvariablen Funktionsbaustein ML_AxisRestore

Min.- Max.- und Default-Werte der Eingänge

Die nachfolgenden Wertebereiche und Defaultwerte können an den Baustein angelegt werden:

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
LoadDefaultParams	BOOL			FALSE	Steigende Flanke an Execute
ParamType	ML_BACK-UP_PARAMS			ASP_PARAMS	Steigende Flanke an Execute
BackupFileName	STRING			"	Steigende Flanke an Execute

Abb.5-184: Min-/Max- und Defaultwerte Funktionsbaustein ML_AxisRestore

Funktionsbeschreibung

Der Funktionsbaustein ML_AxisRestore lädt die Parameter aus einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung auf eine Achse. Nach erfolgreichem Laden der Parameterdatei wird die Konfiguration auf dem Antrieb oder der Achse gespeichert. Sie steht somit auch nach dem Ausschalten wieder zur Verfügung. Die Parameter-Datei kann durch den Funktionsbaustein ML_AxisBackup oder durch IndraWorks erzeugt werden.

LoadDefaultParams

Ist der Eingang "LoadDefaultParams" gesetzt, so wird vor dem Laden der Parameter ein Laden der Default-Parameter des Antriebs bzw. der Achse durchgeführt. Dies ist vor allem dann sinnvoll, wenn die Parameter-Datei mit den Optionen "MODIFIED_PARAM" oder "TO_SAVE_PARAM" erzeugt wurde.



Das Laden der Default-Parameter dauert pro Achse mehrere Sekunden.

ParamType

Über den Eingang "ParamType" werden die wiederherzustellenden Parametertypen ausgewählt. Die folgenden Optionen stehen zur Wahl:

- ASP_PARAMS: Parameter im Antrieb und steuerungsinterne Achsparameter
- A_PARAMS: steuerungsinterne Achsparameter
- SP_PARAMS: Parameter im Antrieb

BackupFileName

Über "BackupFileName" kann der Name der Datei auf der Kompakt-Flash-Karte der Steuerung angegeben werden. Es ist möglich, eine Pfadangabe zu verwenden. Der Pfad muss dann aber auf der Kompakt-Flash-Karte ansprechbar sein.

Beispiel:

- "Antrieb1.par" (liest die Datei "Antrieb1.par" von der User-Partition der Kompakt-Flash-Karte)
- "Params/Antrieb1.par" (liest die Datei "Antrieb1.par" vom Verzeichnis "Params" auf der User-Partition der Kompakt-Flash-Karte)
- "/ata0a/Antrieb2.par" (liest die Datei "Antrieb2.par" von der OEM-Partition der Steuerung)

Zeitverhalten

Der Funktionsbaustein ML_AxisRestore arbeitet asynchron. Nach dem Start durch eine steigende Flanke am Eingang "Execute" liest er die Eingänge "LoadDefaultParams", "ParamType" und "BackupFileName". Solange der Funk-

ML_TechBase.library

tionsbaustein arbeitet, ist der Ausgang "Active" gesetzt. Nach erfolgreichem Abschluss der Operation ist der Ausgang "Done" gesetzt. Tritt ein Fehler auf, wird der Ausgang "Error" gesetzt.

Fehlercodes Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#0000000D	16#00000000	AxisRef liegt außerhalb des gültigen Bereiches
INPUT_RANGE_ERROR	16#00001E01	16#00000000	Ungültiger Eingang "ParamType"
RESOURCE_ERROR	16#00001E02	16#00000000	Die in "ParamType" ausgewählte Parameterart wird von dieser Achse nicht unterstützt
RESOURCE_ERROR	16#00001E04	16#xxxxxxx	Fehler beim Schreiben eines Parameters
ACCESS_ERROR	16#00001E11	16#00000000	Fehler beim Öffnen der Backup-Datei
ACCESS_ERROR	16#00001E13	16#00000000	Fehler beim Lesen der Backup-Datei
ACCESS_ERROR	16#00001E14	16#00000000	Fehler beim Schließen der Backup-Datei
OTHER_ERROR	16#00001E15	16#xxxxxxx	Fehler beim Parsen der Backup-Datei
STATE_MACHINE_ERROR	16#00000006	16#0000FFFF	Unerlaubter Zustand des Funktionsbausteins

Abb.5-185: Fehlercodes Funktionsbaustein ML_AxisRestore

5.10.4 ML_BACKUP_OPTIONS

Typdefinitionen Die folgende Definition beschreibt den Aufzählungstyp ML_BACKUP_OPTIONS, der im Funktionsbaustein [Kap. 5.10.2 "ML_AxisBackup" auf Seite 141](#) verwendet wird.

Die Typ-Definitionen sind in der Bibliothek enthalten.

Der Datentyp "ML_BACKUP_OPTIONS" ist wie folgt definiert:

Aufzählungselement	Wert	Beschreibung
ALL_PARAM	1	Alle Parameter
MODIFIED_PARAM	2	Vom Default-Wert abweichende Parameter
TO_SAVE_PARAM	3	Zur Wiederherstellung notwendige Parameter

Abb.5-186: Definition der Aufzählungselemente des Datentyps ML_BACKUP_OPTIONS

5.10.5 ML_BACKUP_PARAMS

Typdefinitionen Die folgende Definition beschreibt den Aufzählungstyp ML_BACKUP_PARAMS, der in den Funktionsbausteinen [Kap. 5.10.2 "ML_AxisBackup" auf Seite 141](#) und [Kap. 5.10.3 "ML_AxisRestore" auf Seite 144](#) verwendet wird.

Die Typ-Definitionen sind in der Bibliothek enthalten.

Der Datentyp "ML_BACKUP_PARAMS" ist wie folgt definiert:

Aufzählungselement	Wert	Beschreibung
ASP_PARAMS	1	Parameter im Antrieb und steuerungsinterne Achsparameter
A_PARAMS	2	steuerungsinterne Achsparameter
SP_PARAMS	3	Parameter im Antrieb

Abb.5-187: Definition der Aufzählungselemente des Datentyps ML_BACKUP_PARAMS

5.11 Bausteine für Logik Anwendungen

5.11.1 Einführung

Der nachfolgende Baustein ist für Anwendungen gedacht, bei denen eine doppelt verkettete Liste benötigt wird. Dabei können Elemente beliebig hinzugefügt und gelöscht werden. Der Baustein verwaltet die Liste. Damit können Ringpuffer, FIFOs und LIFOs aufgebaut werden.

5.11.2 IL_LinkedList

Kurzbeschreibung

Der Funktionsbaustein verwaltet eine doppelt verkettete Liste. Der Listeninhalt kann vom Anwender selbst definiert werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx18 (in Vorbereitung)	in Vorbereitung

Abb.5-188: Referenztablelle Funktionsbaustein IL_LinkedList

Schnittstellenbeschreibung

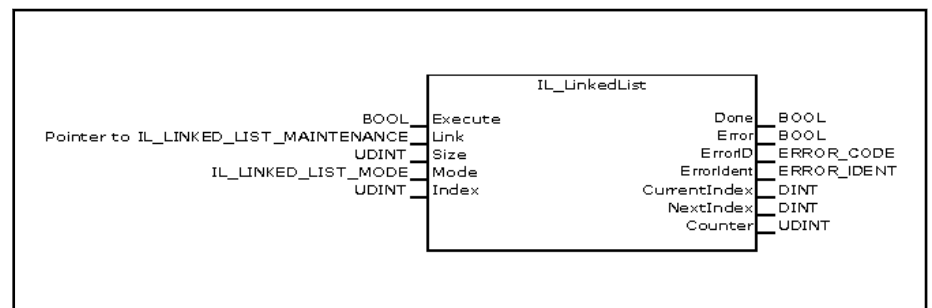


Abb.5-189: Funktionsbaustein IL_LinkedList

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bei steigender Flanke beginnt der Baustein mit seiner Funktionalität
	Link	Pointer to IL_LINKED_LIST_MAINTENANCE	Pointer zu einem Feld von IL_LINKED_LIST_MAINTENANCE-Strukturen, die für die Verwaltung der Liste benötigt wird
	Size	UDINT	Anzahl der Listenelemente, die zu verwalten sind
	Mode	IL_LINKED_LIST_MODE	Modus, den der Baustein ausführen soll, z.B. Element hinzufügen, oder entfernen
	Index	DINT	Index auf den die Operation angewendet werden soll

ML_TechBase.library

E/A-Typ	Name	Datentyp	Beschreibung
VAR_OUTPUT	Done	BOOL	Wenn dieser Ausgang TRUE wird, hat der Baustein seine Arbeit erfolgreich abgeschlossen
	Error	BOOL	Zeigt an, dass ein Fehler innerhalb des Funktionsbausteins aufgetreten ist. Die Ausgangsdaten werden alle ungültig
	ErrorID	ERROR_CODE	Fehlerkennung (siehe Tabelle Fehlercodes , Seite 151)
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehlerklassen
	CurrentIndex	DINT	gegenwärtiger Index
	NextIndex	DINT	nächster Index
	Counter	UDINT	Anzahl an Elementen, die sich in der Liste befinden

Abb.5-190: Schnittstellenvariablen Funktionsbaustein IL_LinkedList

Min.- Max.- und Default-Werte der Eingänge

Die nachfolgenden Wertebereiche und Defaultwerte können an den Baustein angelegt werden:

Name	Typ	Min-Wert	Max-Wert	Default-Wert
Link	0	0	16#FFFFFFFF	0
Size	UDINT	0	16#FFFFFFFF	0
Mode	IL_LINKED_LIST_MODE	ModeInitClear	ModeCheckIndex	ModeInitClear
Index	UDINT	0	16#FFFFFFFF	0

Abb.5-191: Min-/Max- und Defaultwerte Funktionsbaustein IL_LinkedList

Funktionsbeschreibung

Der Funktionsbaustein verwaltet eine doppelt verkettete Liste. Die Liste ist in einem Feld abgelegt, bei der lediglich die Indizes in diesem Feld verwaltet werden. Der Anwender muss für die Verwaltung der Liste ein Feld vom Typ IL_LINKED_LIST_MAINTENANCE anlegen (siehe Definition [Struktur IL_LINKED_LIST_MAINTENANCE](#), Seite 148), das genauso groß sein muss wie seine eigenen Daten.

Die Liste wird durch den Aufruf des Bausteins manipuliert. Die Art der Manipulation kann durch den "Mode"-Eingang festgelegt werden (siehe [Definitionen für Mode-Eingang](#), Seite 149).

Struktur IL_LINKED_LIST_MAINTENANCE

Die Datenstruktur IL_LINKED_LIST_MAINTENANCE wird durch den Baustein verwendet und darf nicht vom Anwender überschrieben werden. Da der Anwender die Länge der verketteten Liste selbst bestimmen kann, muss er auch die Verwaltungsstrukturen für die Liste selbst anlegen. Dies geschieht durch die Definition eines Feldes vom Typ IL_LINKED_LIST_MAINTENANCE. Die interne Struktur sieht folgendermaßen aus und kann für Debug-Zwecke verwendet werden. "Prev" ist dabei ein Pointer auf das vorhergehende Listenelement, "Next" auf das nachfolgende. Ist der Wert Null, so ist dies das Ende des Feldes, der Anfang oder ein unbenutztes Element.

Programm:

```

TYPE IL_LINKED_LIST_MAINTENANCE :
STRUCT
    Prev: POINTER TO DINT;
    Next: POINTER TO DINT;
END_STRUCT
END_TYPE

```

Definitionen für "Mode"-Eingang

Mit Hilfe des Mode-Eingangs bestimmt der Anwender welche Aktion durchgeführt werden soll. Folgende Enumerationen stehen zur Verfügung:

Enumeration	Bedeutung
ModelInitClear	Die verkettete Liste wird initialisiert. Sämtliche Einträge werden gelöscht
ModeAddTail	Es wird am Ende der Liste ein Element eingefügt. Ist die Liste leer, wird das erste Element angelegt
ModeAddHead	Es wird am Anfang der Liste ein Element eingefügt. Ist die Liste leer, wird das erste Element angelegt
ModeRemoveTail	Es wird das letzte Element in der Liste entfernt
ModeRemoveHead	Es wird das erste Element in der Liste entfernt
ModeRemoveIndex	Es wird das Element an Index "Index" entfernt
ModeGetHead	Liefert den Index des Listenanfangs zurück
ModeGetTail	Liefert den Index des Listendes zurück
ModeGetNext	Liefert das nachfolgende Element in der Liste zurück
ModeGetPrev	Liefert das vorhergehende Element in der Liste zurück
ModeGetEmptyIndex	Liefert den nächsten freien Index in der Liste zurück
ModeCheckIndex	Prüft, ob der angegebene Index in der verketteten Liste verwendet wird. Das Ergebnis wird in "CurrentIndex" zurückgegeben: 1: Index wird verwendet -1: Index wird nicht verwendet

Abb.5-192: Enumerationen und deren Bedeutung im Funktionsbaustein IL_LinkedList

Beispiel

Beispiel für die Verwendung des Bausteins:

Definition von Variablen:

Programm:

```
(* Maximale Größe des Feldes *)
uiMAX_ARRAY_SIZE: INT := 10;
(* Feld der eigenen Datenstrukturen *)
arMyData: ARRAY [1..uiMAX_ARRAY_SIZE] OF MY_STRUCTURE;
(* Feld für die Verwaltung der Liste*)
arLinkedList: ARRAY [1..uiMAX_ARRAY_SIZE] OF IL_LINKED_LIST_MAINTENANCE;
(* Baustein Instanz *)
fbMyList: IL_LinkedList;
(* Variable für den Index*)
udiIndex: DINT;
```

Aufruf zum Initialisieren:

Programm:

```
fbMyList(Execute:= TRUE,
  Link:= ADR(arLinkedList),
  Size:= uiMAX_ARRAY_SIZE,
  Mode:= ModeInitClear,
  Index:= ,
  Done =>,
  Error =>,
  ErrorId =>,
  ErrorIdent =>,
  CurrentIndex=> ,
  NextIndex=>);
```

Kopf in Liste einfügen:

ML_TechBase.library

Programm:

```

(* Zuerst Execute Eingang zurück setzen (Wert von Mode spielt keine Rolle) *)
fbMyList(Execute:= FALSE,
  Link:= ADR(arLinkedList),
  Size:= uiMAX_ARRAY_SIZE,
  Mode:= ModeInitClear,
  Index:= ,
  Done =>,
  Error =>,
  ErrorId =>,
  ErrorIdent =>,
  CurrentIndex=> ,
  NextIndex=>);

(* Dann Kopf hinzufügen *)
fbMyList(Execute:= TRUE,
  Link:= ADR(arLinkedList),
  Size:= uiMAX_ARRAY_SIZE,
  Mode:= ModeAddHead,
  Index:= ,
  Done =>,
  Error =>,
  ErrorId =>,
  ErrorIdent =>,
  CurrentIndex=> udiIndex,
  NextIndex=>);

(* Rückgabewert in CurrentIndex enthält den Index zum Kopf der Liste oder
-1 falls ein Fehler auftrat. *)
IF udiIndex <> -1 THEN
  arMyData[udiIndex].StructureMember := MyValue;
END_IF;

```

Durch eine Liste laufen:*Programm:*

```

(* Zuerst Execute Eingang zurück setzen (Wert von Mode spielt keine Rolle) *)
fbMyList(Execute:= FALSE,
  Link:= ADR(arLinkedList),
  Size:= uiMAX_ARRAY_SIZE,
  Mode:= ModeGetHead,
  Index:= ,
  Done =>,
  Error =>,
  ErrorId =>,
  ErrorIdent =>,
  CurrentIndex=> ,
  NextIndex=>);

(* Dann Kopf ermitteln *)
fbMyList(Execute:= TRUE,
  Link:= ADR(arLinkedList),
  Size:= uiMAX_ARRAY_SIZE,
  Mode:= ModeGetHead,
  Index:= ,
  Done =>,
  Error =>,
  ErrorId =>,
  ErrorIdent =>,
  CurrentIndex=> udiIndex,
  NextIndex=>);

(* Rückgabe Wert prüfen (-1 bedeutet Fehler) *)
WHILE (udiIndex > 0) DO
  (* Rückgabewert in CurrentIndex enthält den Index in die Liste *)
  arMyData[udiIndex].StructureMember := MyValue;

  (* Execute Eingang wieder zurück setzen *)
  fbMyList(Execute:= FALSE,
    Link:= ADR(arLinkedList),
    Size:= uiMAX_ARRAY_SIZE,
    Mode:= ModeGetNext,
    Index:= ,
    Done =>,
    Error =>,
    ErrorId =>,
    ErrorIdent =>,
    CurrentIndex=> ,

```

```

NextIndex=>);

(* Dann nächstes Element holen*)
fbMyList(Execute:= TRUE,
  Link:= ADR(arLinkedList),
  Size:= uiMAX_ARRAY_SIZE,
  Mode:= ModeGetNext,
  Index:= ,
  Done =>,
  Error =>,
  ErrorId =>,
  ErrorIdent =>,
  CurrentIndex=> udiIndex,
  NextIndex=>);

END_WHILE

```

Belegung der Ausgänge in Abhängigkeit des "Mode"-Eingangs

Je nach dem wie der Eingang "Mode" belegt ist, werden der Eingang "Index" und die Ausgänge "CurrentIndex" und "NextIndex" belegt. Die nachfolgende Tabelle gibt die entsprechenden Werte an. Ein X steht dabei für einen Wert der unverändert gegenüber dem vorhergehenden Aufruf ist. Tritt ein Fehler auf, wird -1 für "CurrentIndex" zurückgegeben.

Mode	Index	CurrentIndex	NextIndex
ModeInitClear	X	-1	nächster freier Index
ModeAddTail	X	Index des Schwanzes	nächster freier Index
ModeAddHead	X	Index des Kopfes	nächster freier Index
ModeRemoveTail	X	Index des neuen Schwanzes	nächster freier Index
ModeRemoveHead	X	Index des neuen Kopfes	nächster freier Index
ModeRemoveIndex	Index zum Entfernen	Index des vorhergehenden Elementes	nächster freier Index
ModeGetHead	X	Index des Kopfes	X
ModeGetTail	X	Index des Schwanzes	X
ModeGetNext	vorhergehender Index	Index des nachfolgenden Elementes	X
ModeGetPrev	nächster Index	Index des vorhergehenden Elementes	X
ModeGetEmptyIndex	X	X	nächster freier Index
ModeCheckIndex	Index für Prüfung	1: gefunden -1: nicht gefunden	-1

Abb.5-193: Bausteinausgänge in Abhängigkeit von "Mode"

Fehlercodes Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#1800	16#0	Eingang "Index" außerhalb gültiger Werte
INPUT_INVALID_ERROR	16#1801	16#0	Eingang "Link" ist ein Null Pointer
ACCESS_ERROR	16#1802	16#xxx	Ein Index ist immer noch in der Liste verlinkt, obwohl diese leer ist (interner Fehler)
ACCESS_ERROR	16#1803	16#0	Element in Liste soll freigegeben werden, aber Liste ist bereits leer

ML_TechBase.library

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#1804	16#0	Liste ist voll
STATE_MACHINE_ERROR	16#1805	16#0	Eingang "Mode" besitzt einen ungültigen Wert

Abb.5-194: Fehlercodes Funktionsbaustein IL_LinkedList

5.12 Funktionen zur Berechnung von Modulowerten

5.12.1 Einführung

Die nachfolgenden Funktionen berechnen einen Modulowert. Je nachdem welche Funktion gewählt wird, werden unterschiedliche Wertebereiche zugrunde gelegt.

5.12.2 MB_GetModuloType01

Kurzbeschreibung

Die Funktion liefert den Modulowert zurück. Der zurückgelieferte Wertebereich liegt zwischen - Modulowert und + Modulowert.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx18	MX_Technology07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-195: Referenztabelle Funktion MB_GetModuloType01

Schnittstellenbeschreibung

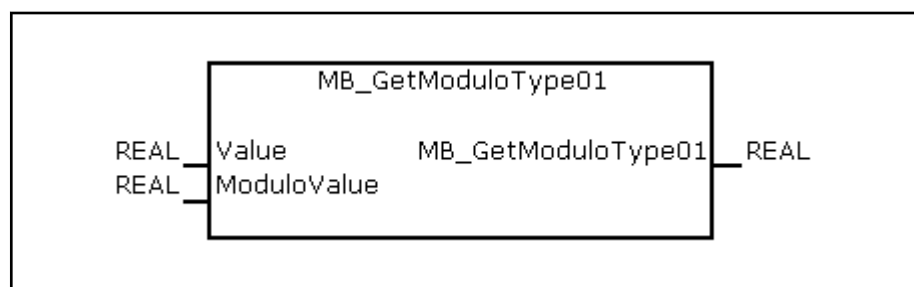


Abb.5-196: Funktion MB_GetModuloType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Value	REAL	Wert an dem die Modulofunktion angewendet werden soll
	ModuloValue	REAL	Modulowert der für die Moduloperation verwendet werden soll

Abb.5-197: Schnittstellenvariablen Funktion MB_GetModuloType01

Min.- Max.- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Value	REAL	- 1,175494e-38	3,402823e+38	0.0	mit jedem Aufruf
ModuloValue	REAL	- 1,175494e-38	3,402823e+38	360.0	mit jedem Aufruf

Abb.5-198: Min-/Max- und Default-Werte Funktion MB_GetModuloType01

Funktionsbeschreibung

Die Funktion liefert den Modulowert zurück. Der zurückgelieferte Wertebereich liegt zwischen 0.0 und + Modulowert. Positiver und negativer Grenzwert ist nicht Teil des Wertebereichs (siehe auch nachfolgende Grafik, in der auf der Abszisse der an die Funktion übergebene Wert und auf der Ordinate der von der Funktion zurückgelieferte Wert dargestellt sind).

Wird für "ModuloValue" ein Wert 0.0 übergeben, liefert die Funktion den Wert "Value" zurück.

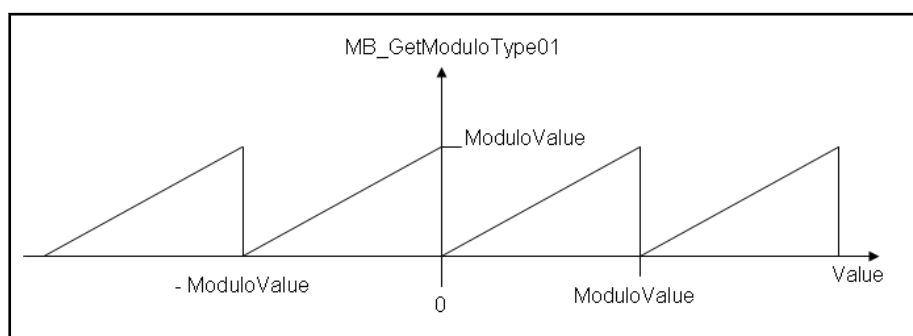


Abb.5-199: Grafische Darstellung der Modulo-Funktion

Fehlerbehandlung

Die Funktion generiert keine Fehler.

5.12.3 MB_GetModuloType02

Kurzbeschreibung

Die Funktion liefert den Modulowert zurück. Der zurückgelieferte Wertebereich liegt zwischen - Modulowert und + Modulowert.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx07VRS ohne MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-200: Referenztable

Schnittstellenbeschreibung

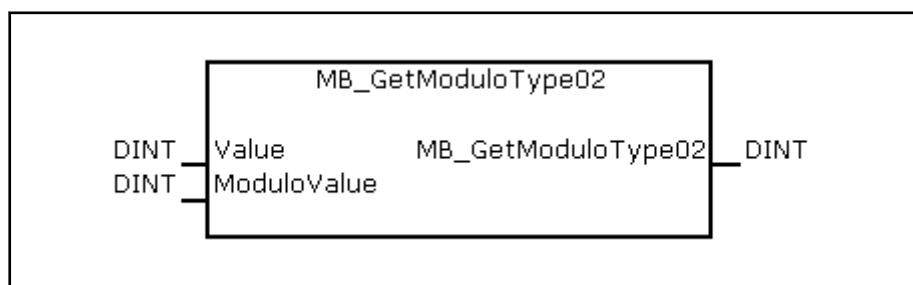


Abb.5-201: Funktion MB_GetModuloType02

ML_TechBase.library

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Value	DINT	Wert, an dem die Modulofunktion angewendet werden soll
	ModuloValue	DINT	Modulowert, der für die Modulooperation verwendet werden soll

Abb.5-202: Schnittstellenvariablen Funktion MB_GetModuloType02

Min.- Max.- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Value	DINT	- 2147483648	2147483647	0	mit jedem Aufruf
ModuloValue	DINT	- 2147483648	2147483647	360	mit jedem Aufruf

Abb.5-203: Min-/Max- und Default-Werte Funktion MB_GetModuloType02

Funktionsbeschreibung

Die Funktion liefert den Modulowert zurück. Der zurückgelieferte Wertebereich liegt zwischen 0 und + Modulowert. Positiver und negativer Grenzwert ist nicht Teil des Wertebereichs (siehe auch nachfolgende Grafik, in der auf der Abszisse der an die Funktion übergebene Wert und auf der Ordinate der von der Funktion zurückgelieferte Wert dargestellt sind).

Wird für "ModuloValue" ein Wert 0 übergeben, liefert die Funktion den Wert "Value" zurück.

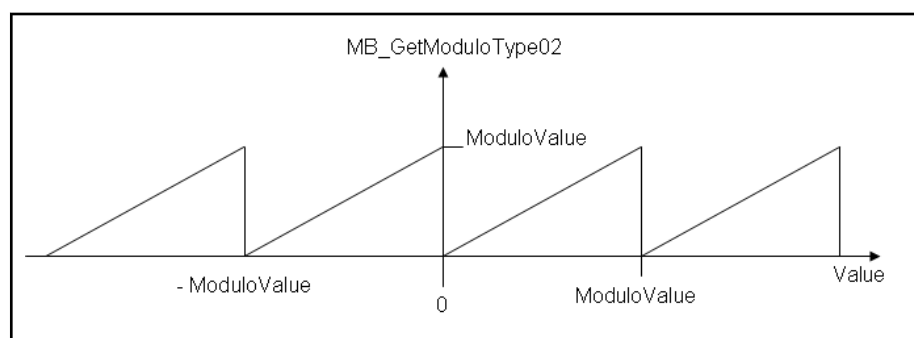


Abb.5-204: Grafische Darstellung der Modulo-Funktion

Fehlerbehandlung

Die Funktion generiert keine Fehler.

5.12.4 MB_GetModuloType03

Kurzbeschreibung

Die Funktion liefert den Modulowert zurück. Der zurückgelieferte Wertebereich liegt zwischen - Modulowert und + Modulowert.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx18	MX_Technology07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-205: Referenztable Funktion MB_GetModuloType03

Schnittstellenbeschreibung

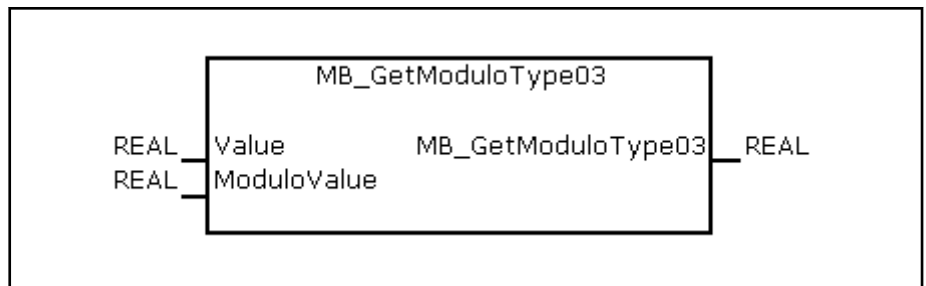


Abb.5-206: Funktion MB_GetModuloType03

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Value	REAL	Wert an dem die Modulofunktion angewendet werden soll
	ModuloValue	REAL	Modulowert der für die Modulooperation verwendet werden soll

Abb.5-207: Schnittstellenvariablen Funktion MB_GetModuloType03

Min.- Max.- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Value	REAL	- 1,175494e-38	3,402823e+38	0.0	mit jedem Aufruf
ModuloValue	REAL	- 1,175494e-38	3,402823e+38	360.0	mit jedem Aufruf

Abb.5-208: Min-/Max- und Default-Werte Funktion MB_GetModuloType03

Funktionsbeschreibung

Die Funktion liefert den Modulowert zurück. Der zurückgelieferte Wertebereich liegt zwischen - Modulowert und + Modulowert. Positiver und negativer Grenzwert ist nicht Teil des Wertebereichs (siehe auch nachfolgende Grafik in der auf der Abszisse der an die Funktion übergebene Wert und auf der Ordinate der von der Funktion zurückgelieferte Wert dargestellt sind).

Wird für "ModuloValue" ein Wert 0.0 übergeben, liefert die Funktion den Wert "Value" zurück.

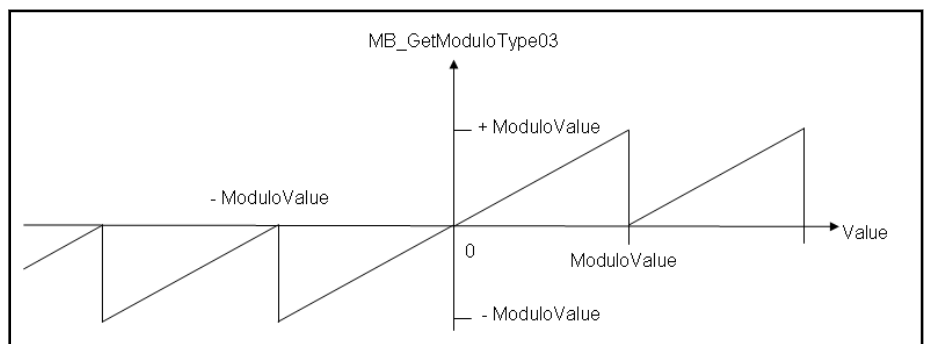


Abb.5-209: Grafische Darstellung der Modulo-Funktion

Fehlerbehandlung

Die Funktion generiert keine Fehler.

5.12.5 MB_GetModuloType04

Kurzbeschreibung

Die Funktion liefert den Modulowert zurück. Der zurückgelieferte Wertebereich liegt zwischen - modulo / 2 und + modulo / 2.

ML_TechBase.library

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx18	MX_Technology07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-210: Referenztabelle FunktionMB_GetModuloType04

Schnittstellenbeschreibung

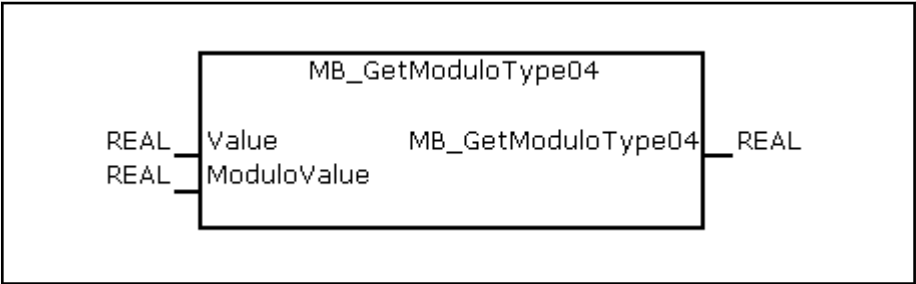


Abb.5-211: Funktion MB_GetModuloType04

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Value	REAL	Wert an dem die Modulofunktion angewendet werden soll
	ModuloValue	REAL	Modulowert der für die Modulooperation verwendet werden soll

Abb.5-212: Schnittstellenvariablen Funktion MB_GetModuloType04

Min.- Max.- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Value	REAL	- 1,175494e-38	3,402823e+38	0.0	mit jedem Aufruf
ModuloValue	REAL	- 1,175494e-38	3,402823e+38	360.0	mit jedem Aufruf

Abb.5-213: Min-/Max- und Default-Werte Funktion MB_GetModuloType04

Funktionsbeschreibung

Die Funktion liefert den Modulowert zurück. Der zurückgelieferte Wertebereich liegt zwischen $- 0.5 \cdot \text{Modulowert}$ und $+ 0.5 \cdot \text{Modulowert}$. Positiver und negativer Grenzwert sind Teil des Wertebereichs (siehe auch nachfolgende Grafik in der auf der Abszisse der an die Funktion übergebene Wert und auf der Ordinate der von der Funktion zurückgelieferte Wert dargestellt sind).
Wird für "ModuloValue" ein Wert 0.0 übergeben, liefert die Funktion den Wert "Value" zurück.

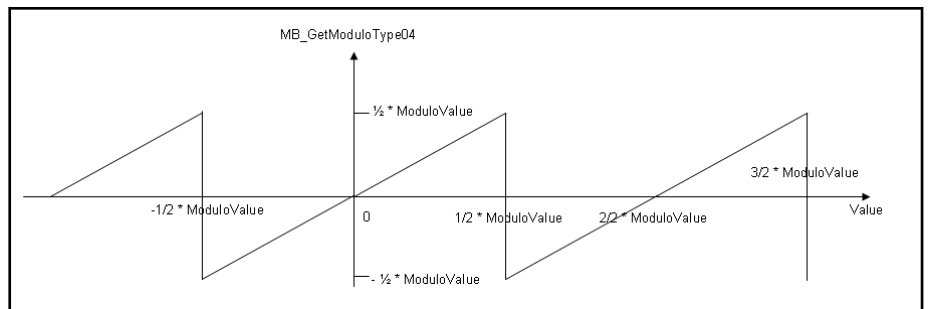


Abb.5-214: Graphische Darstellung der Modulo-Funktion

Fehlerbehandlung Die Funktion generiert keine Fehler.

5.13 Hilfsfunktionen

5.13.1 MB_GetSystemInfo

Kurzbeschreibung Dieser Funktionsbaustein überprüft Funktionspakete und Systemoptionen und gibt alle aktivierten Funktionspakete und Systemoptionen zusammen mit dem Systemtyp aus.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library
IndraMotion MLD/MPx07VRS ohne MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.5-215: Referenztablelle

Schnittstellenbeschreibung

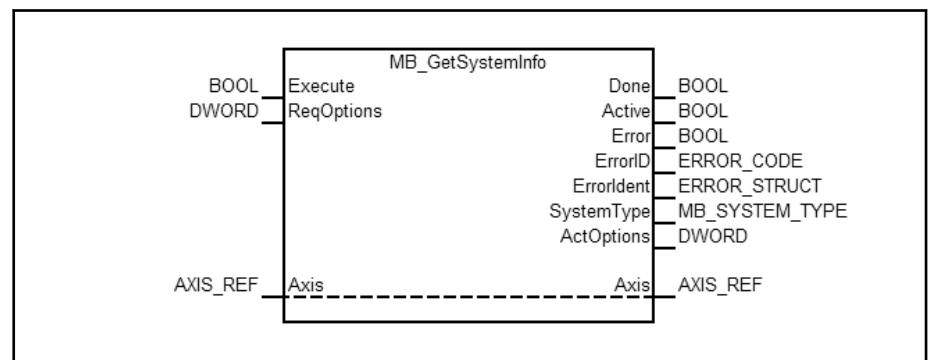


Abb.5-216: Funktionsbaustein MB_GetSystemInfo

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Axis	AXIS_REF	Zu prüfende Achse
VAR_INPUT	Execute	BOOL	Startet die Überprüfung
	ReqOptions	DWORD	Benötigte Systemoptionen

ML_TechBase.library

E/A-Typ	Name	Datentyp	Beschreibung
VAR_OUTPUT	Done	BOOL	Die Überprüfung verlief fehlerfrei
	Active	BOOL	Der Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Grundsätzliche Fehlerursache
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe "Fehlercodes" auf Seite 159
	SystemType	MB_SYSTEM_TYPE	Systemtyp
	ActOptions	DWORD	tatsächlich aktivierte Systemoptionen

Abb.5-217: Schnittstellenvariablen Funktionsbaustein MB_GetSystemInfo

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
ReqOptions	DWORD	0	n.def.	0	steigende Flanke an Execute

Abb.5-218: Eingangsverhalten MB_GetSystemInfo

Funktionsbeschreibung

Dieser Funktionsbaustein arbeitet in SERCOS-Phase 2 oder 4. Bei steigender Flanke an "Execute" wird der Eingang "ReqOptions" übernommen. Danach wird das System überprüft. Sind in "ReqOptions" Optionen oder Funktionspakete gesetzt, die nicht verfügbar sind, wird ein entsprechender Fehler (siehe ["Fehlercodes" auf Seite 159](#)) generiert.

Nach erfolgreichem Abschluss wird der Systemtyp in "SystemType" und alle aktivierten Optionen in "ActOptions" ausgegeben und "Done" auf TRUE gesetzt.

Option	IndraMotion XLC	MLC/IndraLogic	IndraMotion MLD	Wert	Beschreibung
MB_OPEN_LOOP	Axis		Axis	16#00000001	Grundpaket Open Loop
MB_CLOSED_LOOP	Axis		Axis	16#00000002	Grundpaket Closed-Loop
MB_SERVO	Axis		Axis	16#00000010	Funktionspaket Servo
MB_SYNCHRONIZATION	Axis		Axis	16#00000020	Funktionspaket Synchronization
MB_MAIN_SPINDLE	Axis		Axis	16#00000040	Funktionspaket Hauptspindel
MB_ML	immer gesetzt		erste Achse	16#00000200	Option Technologiefunktion
MB_TF	immer gesetzt		erste Achse	16#00000200	Option erweiterte Technologiefunktion
MB_MA	immer gesetzt		erste Achse	16#00000300	
MB_SAFETY				16#00010000	Sicherheitstechnik in der Achse aktiv
MB_AXISDATA	lokale Achsen		erste Achse	16#00100000	AxisData aktiviert
MB_REALAXIS	Reale Achse		immer gesetzt	16#01000000	
MB_VIRTUALAXIS	Virtuelle Achse			16#02000000	

Option	IndraMotion MLC/IndraLogic XLC	IndraMotion MLD	Wert	Beschreibung
MB_LINKAXIS	Achsen über Querverbund		16#04000000	
MB_LOCAL_INTERPOL	Interpolation in der Steuerung		16#10000000	
MB_INDRADRIIVE	Indradrive	immer gesetzt	16#20000000	
MB_INDRADRIIVE_1X	Indradrive 2. Generation	Indradrive 2. Generation	16#40000000	

Abb.5-219: Mögliche Grundpakete, Funktionspakete und Optionen in ReqOptions und ActOptions

Benötigte Optionen sollten in "ReqOptions" angegeben werden. Ist mindestens eine Option auf der angegebenen Achse nicht verfügbar, wird die entsprechende Fehlermeldung generiert. Sind alle in "ReqOptions" angeforderten Optionen vorhanden, wird "Done" aktiv. Es ist auch möglich keine benötigten Optionen anzugeben und die Auswertung selbst vorzunehmen.

Die Überprüfung der Optionen benötigt mehrere Funktionsbausteinanrufe. Während dieser Zeit ist der Ausgang "Active" gesetzt.

Systemtyp	Beschreibung
IL_SYSTEM_UNKNOWN	Unbekanntes System
IL_SYSTEM_MLC	IndraMotion MLC
IL_SYSTEM_XLC	IndraLogic XLC
IL_SYSTEM_MLD	IndraMotion MLD

Abb.5-220: Mögliche Systemtypen am Ausgang SystemType

Fehlerverhalten

Ist eine in "ReqOptions" angegebene Option nicht verfügbar oder ist ein anderer Fehler aufgetreten, so wird "Error" TRUE.

Fehlercodes

Der Funktionsblock generiert die folgenden Fehlermeldungen der "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR	16#1503	16#0000	AxisData-Struktur nicht aktiviert
RESOURCE_ERROR	16#000F	16#0001	Notwendige Option TF nicht freigeschaltet
RESOURCE_ERROR	16#000F	16#0002	Notwendige Option ML nicht freigeschaltet
RESOURCE_ERROR	16#000F	16#0003	Notwendige Option MA nicht freigeschaltet
RESOURCE_ERROR	16#000F	16#0010	Notwendiges Grundpaket Open-Loop nicht aktiv
RESOURCE_ERROR	16#000F	16#0011	Notwendiges Grundpaket Closed-Loop nicht aktiv
RESOURCE_ERROR	16#000F	16#0020	Notwendiges Funktionspaket Synchronisation nicht aktiv

ML_TechBase.library

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#000F	16#0021	Notwendiges Funktionspaket Servo nicht aktiv
RESOURCE_ERROR	16#000F	16#0022	Notwendiges Funktionspaket Hauptspindel nicht aktiv

Abb.5-221: Fehlercodes Funktionsbaustein MB_GetSystemInfo

5.14 Funktionsbaustein zum Download von Kurvenscheiben

5.14.1 Einführung

Der nachfolgende Baustein ist für den Download von Kurvenscheiben auf den Antrieb und die Steuerung gedacht, um die vorhandenen Detailunterschiede zu kapseln. Die folgende Enumeration "MB_CAMPROF" und die Struktur "MB_CAMTABLE" werden in den Funktionsbausteinen zur Erstellung von Kurvenscheibentabellen verwendet.

5.14.2 MB_LoadCamData

Kurzbeschreibung

Der MB_LoadCamData übernimmt den Download von Kurvenscheiben auf den Antrieb und die Steuerung, um die vorhandenen Detailunterschiede zu kapseln.

Zuordnung Zielsystem/Bibliothek	IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechBase.compiled-library

Abb.5-222: Referenztabelle MB_LoadCamData

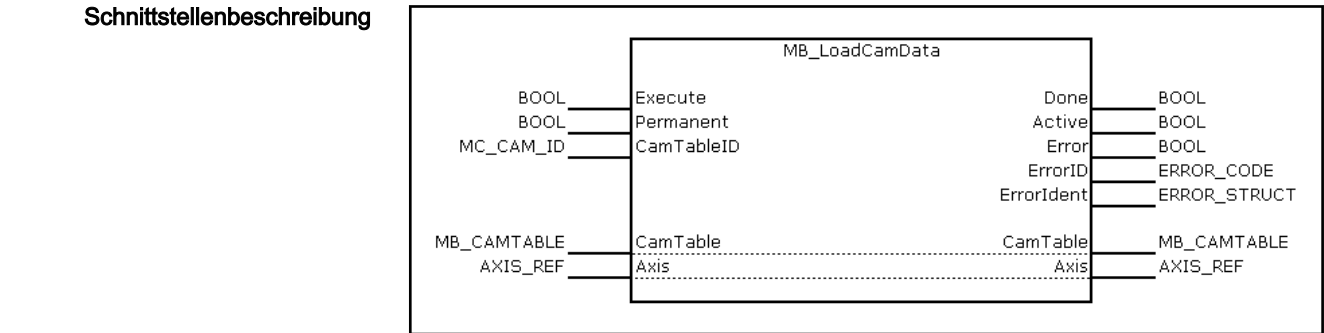


Abb.5-223: Funktionsbaustein MB_LoadCamData

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellen-Elementen und Information über die Anzahl der Elemente
	Axis	AXIS_REF	Auswahl der Referenz (Antrieb oder Steuerung)
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	Permanent	BOOL	Einstellmöglichkeit, ob Kurvenscheibe permanent in den Parameter geschrieben werden soll, oder nur temporär in den flüchtigen Systemspeicher
	CamTableID	MC_CAM_ID	Auswahl der gewünschten Kurvenscheibentabelle des Antriebs oder der Steuerung

E/A-Typ	Name	Typ	Kommentar
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe "Fehlerbehandlung" auf Seite 162)

Abb.5-224: Schnittstellenvariablen Funktionsbaustein MB_LoadCamData

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
Permanent	BOOL			FALSE	Steigende Flanke an Execute
CamTableID	MC_CAM_ID			CAM_TABLE_1	Steigende Flanke an Execute

Abb.5-225: Min-/Max- und Default-Werte Funktionsbaustein MB_LoadCamData

Die Enumeration "MC_CAM_ID" unterstützt die Auswahl der gewünschten Kurventabelle.

Element	Wert	Beschreibung
CAM_TABLE_1	1	Kurventabelle 1
CAM_TABLE_2	2	Kurventabelle 2
.....		
CAM_TABLE_255	255	Kurventabelle 255

Abb.5-226: Elemente des Aufzählungstyps MC_CAM_ID

Die Struktur [Kap. 5.14.4 "MB_CAMTABLE" auf Seite 163](#) besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Funktionsbeschreibung

Nach der Bearbeitungs freigabe mittels "Execute" prüft der Baustein, in welchem Format die über "CamTableID" ausgewählte Kurvenscheibentabelle beschrieben werden kann.

Die in der Struktur "MB_CAMTABLE" enthaltene Kurvenscheibentabelle wird dann in dieses Format gewandelt und mit Hilfe eines WriteParameter-Bausteins in die Steuerung oder den Antrieb geladen.

Die Auswahl des Speicherortes (Steuerung oder Antrieb) der Kurvenscheibentabelle wird durch den Eingang "Axis " festgelegt.

Durch den Eingang "Permanent" kann zwischen der permanenten Ablage der Kurvenscheibe oder der temporären Ablage im flüchtigen Speicher (RAM)

ML_TechBase.library

des Geräts entschieden werden. Die Realisierung der unterschiedlichen Ablage wird auf der Steuerung durch zwei verschiedene WriteParameter-Bausteine durchgeführt. Auf dem Antrieb wird die Unterscheidung der permanenten und temporären Ablage durch den Parameter S-0-0269 realisiert.

Über das Element "NumberOfElements" der "MB_CAMTABLE" Struktur ist ersichtlich, wie viele gültige Kurvenscheibepunkte das Array "CamTable" enthält. Ist über "CamTableID" eine Antriebskurvenscheibe mit <1024 Punkten ausgewählt, "NumberOfElements" signalisiert aber mehr Elemente, so wird ein Fehler generiert.

Die umgerechnete Kurvenscheibentabelle wird im Falle der fehlerfreien Bearbeitung des Funktionsbausteins an einen WriteParameter-Baustein übergeben und in die Steuerung oder den Antrieb geladen, wobei die Gültigkeit der Umrechnung und des Ladevorgangs durch den Ausgang "Done" signalisiert wird.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" sowie "Axis" werden im Fehlerfall nicht aktualisiert.

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins MB_LoadCamData weist folgende Eigenschaft auf:

- Der Algorithmus zur Umrechnung der Kurvenscheibenstützpunkte ist auf mehrere Zyklen, d. h. Aufrufe der Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden

	Kurvenscheibe in der Steuerung		Kurvenscheibe im Antrieb	
Axis	CntrlNo = LOCAL_CNTRL AxisNo = NO_OBJECT		CntrlNo = LOCAL_CNTRL AxisNo = AXIS_1..AXIS_99	
Permanent	FALSE (temporäre Kurvenscheibe)	TRUE (permanente Kurvenscheibe)	FALSE (temporäre Kurvenscheibe)	TRUE (permanente Kurvenscheibe)
CamTableID	1..99 (3-1024 Punkte)		1..4 (3-1024 Punkte); 5..8 (3-128 Punkte)	

Abb.5-227: Eingangsbelegungen des Funktionsbausteins MB_LoadCamData um die Kurvenscheibe in den Antrieb bzw. die Steuerung zu laden

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170x. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#00000006	16#00000000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR	16#00002200	16#00000001	Eingang NumberOfElements ist kleiner als Minimum oder größer als Maximum
INPUT_INVALID_ERROR	16#00002201	16#00000001	Adresse von Feld CamTable ist null

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#00002210	16#00000001	Eingang CamTableID ist zu groß für Auswahl vom Eingang Axis
INPUT_INVALID_ERROR	16#00002210	16#00000002	Eingang NumberOfElements ist zu groß für ausgewählte CamTableID

Abb.5-228: Fehlercodes Funktionsbaustein MB_LoadCamData

5.14.3 MB_CAMPROF

Typdefinitionen Die Enumeration wird verwendet um zwischen verschiedenen Kurvenscheibenprofilen zu wählen. Die Auswahl CAMPROF_STANDARD und CAMPROF_ADDITIVE beinhalten das alte Format, bei dem das letzte Element der Kurvenscheibentabelle der Stützpunkt $\varphi = 360^\circ - \Delta\varphi$ ist. Während die Auswahl CAMPROF_ADDITIVE_SEGMENTABLE das neue Format beinhaltet, bei dem das letzte Element der Kurvenscheibentabelle der Stützpunkt $\varphi = 360^\circ$ ist.

Das verwendete Format kann bei Kurventabellen im Antrieb über den Parameter P-0-0086 gewählt werden. Bei Kurventabellen in der Steuerung ist nur das neue Format verfügbar.

Element	Wert	Beschreibung
CAMPROF_STANDARD	0	Kurvenscheibenprofil ohne Untersetzung, nicht segmentierbar (altes Format)
CAMPROF_ADDITIVE	1	Überlagerungsprofil mit Untersetzung, nicht segmentierbar (altes Format)
CAMPROF_ADDITIVE_SEGMENTABLE	2	Überlagerungsprofil mit Untersetzung, segmentierbar (neues Format)

Abb.5-229: Elemente des Aufzählungstyps MB_CAMPROF

5.14.4 MB_CAMTABLE

Die Struktur besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
CamTable	ARRAY [0..1023] OF LREAL				Steigende Flanke an Execute	Feld mit den berechneten Kurvenscheibenpunkten
NumberOfElements	UINT	3	1024		Steigende Flanke an Execute	Anzahl der gültigen Kurvenscheibenpunkte des Arrays CamTable

Abb.5-230: Min-/Max-/Default-Werte der Struktur MB_CAMTABLE

6 ML_TechMotion.library

6.1 Übersicht über die Funktionsbausteine

Die Funktionsbausteine von ML_TechMotion

Funktionsbaustein	Beschreibung
Funktionsbausteine zur Anwendung "Fliegende Säge"	
ML_FlyingShear und MX_FlyingShear	Enthält den typischen Ablauf einer "Fliegenden Säge"
Funktionsbausteine zur Anwendung "Zugspannungsregler"	
MB_TensionControlLoadCellType01	Der Zugspannungsregler regelt, abhängig vom Messwert einer Kraftmessdose, die Zugspannung auf den gewünschten Sollwert
Funktionsbausteine zur Anwendung "Schlaufenregelung"	
ML_SagControlA	Enthält einen Schlaufenregler mit analoger Istwerterfassung der Schlaufe
ML_SagControlC	Enthält einen Schlaufenregler mit binären Min/Max- Kontakten
Funktionsbausteine zum taktsynchronen Ein- und Auskoppeln	
MB_CamLock	Wird verwendet um die durch den Funktionsbaustein MB_PrepareCams berechneten und gespeicherten Run-, LockOn- und LockOff-Kurvenscheibenprofile zu aktivieren
MB_PrepareCams	Verwendet ein Polynom 5. Ordnung um 3 Kurvenscheibenprofile zu berechnen, zu erzeugen und sie in Parameter, die durch die MC_CAM_ID-Eingänge festgelegt sind, zu laden. Die erzeugten Bewegungsprofile enthalten Randbedingungen für die Position und Geschwindigkeit
Funktionsbausteine zur allgemeinen Verwendung	
MB_DistanceAccumulator	Summiert den relativ gefahrenen Weg mittels eines Gebers auf und hält an, sobald der aufsummierte Weg größer als oder gleich des TotalDistance Eingangswerts ist. Die aktuelle Geberposition wird mittels des CurrentPosition Eingangs eingegeben
MB_AxisDistanceAccumulator	Summiert den relativ gefahrenen Weg mittels eines Gebers auf und hält an, sobald der aufsummierte Weg größer als oder gleich des TotalDistance Eingangswerts ist. Dieser Funktionsbaustein verwendet den Achsreferenzeingang (AXIS_REF) um die aktuelle Geberposition automatisch abzufragen
MB_ReachVelocityType02	Berechnet den entsprechenden Ruck, die Beschleunigung und die Zielgeschwindigkeit zum Abfahren bestimmter ruckbegrenzter Geschwindigkeitsverläufe (z.B. S-Kurve) und lässt hierbei bis zu 3 Änderungen der Zielgeschwindigkeiten zu
Funktionsbausteine zur Anwendung "Positionsüberwachung"	
MB_PositionMonitoring_GantryCant	Überwacht die relative Position zweier Achsen bezüglich Verkantung zueinander. Mittels der Toleranzband-Funktionalität ist die Rückzugsbewegung möglich, wobei die Bewegungsrichtung aus der Verkantung weiter überwacht wird
MB_PositionMonitoring_PosCollision	Überwacht die absolute und relative Position zweier Achsen bezüglich Auszahnung und Kollision
Funktionsbausteine zur Realisierung einer "Zuführmaschine"	

ML_TechMotion.library

Funktionsbaustein	Beschreibung
MB_SmartBeltType01	Führt am Produkt eine Phasenkorrektur mittels eines dreieckigen Geschwindigkeitsprofils durch
MB_MagicBeltType01	Enthält Funktionalitäten zur Ansteuerung einer Kette einer MagicBelt (=Mehrzugkette) Applikation
Funktionsbausteine zur Erstellung von Kurvenscheibentabellen für Technologiefunktionen	
MB_PilgrimStepCalcType01	Übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunkten sowie die Hübe des Vorwärts- und Rückwärtstransportes und die Stoppositionen für den Pilgerschrittbetrieb
MB_PrintLengthCorrCalcType01	Übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunkten für die Technologiefunktion Drucklängenkorrektur, um die Länge eines Druckes auf dem Druckgut (Warenbahn bzw. Bogen) zu verändern

Abb.6-1: ML_TechMotion.lib

6.2 Funktionsbaustein zur Anwendung "Fliegende Säge"

6.2.1 Einführung und Übersicht

Bei der Applikation "Fliegende Säge" wird eine Bearbeitungsstation auf ein kontinuierlich durchlaufendes Material (wie z.B. Metall, Kunststoff etc.) auf-synchronisiert und die Bearbeitung in dem Synchronlauf der Bearbeitungsstation ausgeführt.

Die Materialposition wird üblicherweise über ein Messrad erfasst und über einen optionalen Encodereingang an den Antriebsregler übertragen. Der Antriebsregler synchronisiert dann den angeschlossenen Servomotor auf das kontinuierlich durchlaufende Material, wenn die vorgegebene Schnittlänge durchlaufen wurde. Der Funktionsbaustein [Kap. 6.2.2 "MX\(L\)_FlyingShear "](#) auf Seite 166 bzw. [Kap. 6.2.3 "ML_FlyingShear - Besonderheiten für IndraMotion MLC"](#) auf Seite 173 übernimmt die Aufsynchronisierung und setzt den Ausgang "InSync", sobald die Bearbeitungsstation im Synchronlauf ist. In diesem Zustand kann die Bearbeitung durchgeführt werden, da sich nun die Bearbeitungsstation synchron zum Material bewegt.

Nach Abschluss der Bearbeitung kann der Synchronlauf über den Eingang "MoveReturn" unterbrochen werden, und die Bearbeitungsstation fährt wieder in die vorgegebene Ausgangslage zurück.

Für Testzwecke kann auch (anstelle des Messrades) eine virtuelle Achse verwendet werden. Über diese virtuelle Achse ist es dann möglich das Messrad zu simulieren.

6.2.2 MX(L)_FlyingShear

Kurzbeschreibung Der Funktionsbaustein MX(L)_FlyingShear enthält den typischen Ablauf einer fliegenden Säge und führt die folgenden Schritte aus, sobald der Eingang "Start" TRUE ist:

- Folgeachse in die Ausgangslage "ReturnPos" fahren und warten, bis die Achse in Position ist
- Folgeachse über ein Lock-On-Kurvenscheibenprofil auf die Leitachse aufsynchronisieren
- Ausgang "InSync" setzen, sobald die Folgeachse (Trennstation) auf das Material aufsynchronisiert ist

- Folgeachse wieder in Ausgangslage positionieren, wenn am Eingang "MoveReturn" das entsprechende Signal anliegt



Der Betrieb des Funktionsbausteines erfordert besondere Randbedingungen (z.B. Interpolation im Antrieb, zyklischer Aufruf des FB in der Motion-Task...).

Bitte beachten Sie daher die Inbetriebnahmehinweise [Kap. 6.2.3 "ML_FlyingShear - Besonderheiten für IndraMotion MLC"](#) auf Seite 173.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_Technology07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.6-2: Referenztabelle

Schnittstellenbeschreibung

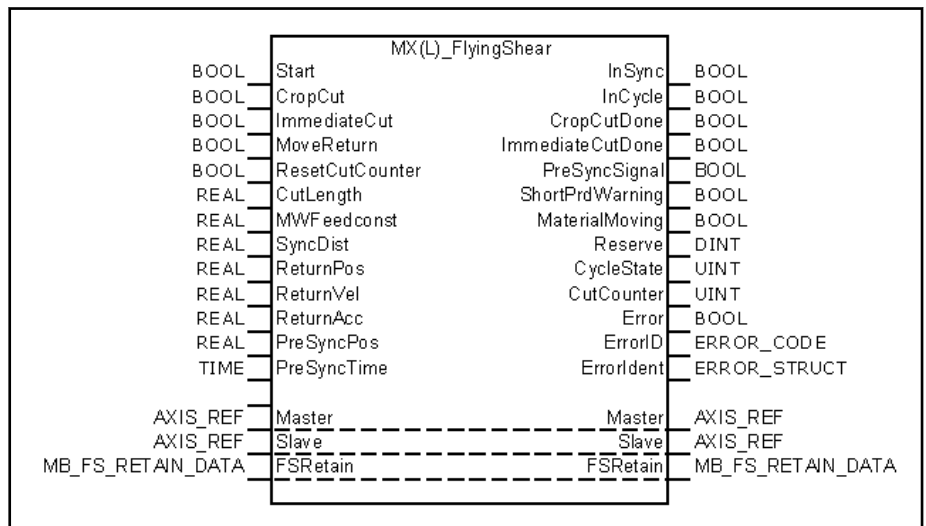


Abb.6-3: Funktionsbaustein MX(L)_FlyingShear

E/A-Typ	Name	Datentyp	Erläuterung
VAR_IN_OUT	Master	AXIS_REF	Referenz zur Leitachse
	Slave	AXIS_REF	Referenz zur Folgeachse (Trennstation)
	FSRetain	MB_FS_RETAIN_DATA*1	Referenz auf die benötigten Retain-Daten dieses Funktionsbausteins

ML_TechMotion.library

E/A-Typ	Name	Datentyp	Erläuterung
VAR_INPUT	Start	BOOL	Start=TRUE: Start des FlyingShear Funktionsbausteins. Je nach den Zuständen der Eingänge "ImmediateCut" und "CropCut" führt der Funktionsbaustein die folgenden Funktionen aus: Start=TRUE und positive Flanke am Eingang "ImmediateCut": Die Trennstation wird sofort aufsynchronisiert und führt die Schnittlänge (Cutlength) aus, sobald der Sofortschnitt (ImmediateCut) ausgeführt ist. Start=TRUE und positive Flanke am Eingang "CropCut": Die Trennstation wird aufsynchronisiert, nachdem die Schnittlänge (Cutlength) die Maschine passiert hat, und führt die Schnittlänge (Cutlength) aus, sobald der Kopfschnitt (CropCut) ausgeführt ist. Start=TRUE und kein Signal "ImmediateCut" und "CropCut": Die Trennstation wird aufsynchronisiert, nachdem die Schnittlänge (Cutlength), gemessen vom letzten Schnitt aus, die Maschine passiert hat
	CropCut	BOOL	Auf eine steigende Flanke hin beginnt die Folgeachse mit dem Aufsynchronisieren, nachdem die Schnittlänge die Maschine passiert hat. Befindet sich gerade ein Schneidezyklus in Ausführung (InCycle=TRUE), wird der Kopfschnitt (CropCut) im nächsten Schnittzyklus ausgeführt
	ImmediateCut	BOOL	Start = TRUE: Dieser Fall ist für einen Sofortschnitt bei laufendem Material vorgesehen. Der Sofortschnitt (ImmediateCut) wird durch eine positive Flanke ausgelöst. Die Folgeachse wird sofort aufsynchronisiert. Wird gerade ein Schnitt ausgeführt (InCycle = TRUE), erfolgt der Sofortschnitt im nächsten Schnittzyklus. Wenn (Start = FALSE) und ($V_{\text{measuring\_wheel}} < 5 \text{ U/min}$): Dieser Fall ist für einen Sofortschnitt bei Materialstillstand vorgesehen. Der Sofortschnitt (ImmediateCut) wird durch eine positive Flanke ausgelöst. Die Folgeachse bleibt im Stillstand und wird nicht synchronisiert. Der Ausgang "InSync" wird sofort gesetzt, und der Funktionsbaustein erwartet als Abschluss das Eingangssignal "MoveReturn". Nach Abschluss dieses Vorgangs ist die fliegende Säge auf das Material referenziert
	MoveReturn	BOOL	Beendet den Synchronlauf der Folgeachse mit dem Material und positioniert die Folgeachse wieder in der Ausgangslage "ReturnPos"
	ResetCutCounter	BOOL	Die positive Flanke setzt den Schnitzzähler (Cut Counter) zurück

E/A-Typ	Name	Datentyp	Erläuterung
	Cutlength	REAL	Die vorgegebene Materialschnittlänge ^{*2*3}
	MWFeedconst	REAL	Die Vorschubkonstante des Messrades pro Umdrehung ^{*2 *3}
	SyncDist	REAL	Aufsynchronisierweg, den die Folgeachse zum Aufsynchronisieren auf die Leitachse benötigt ^{*2*3} . Je kürzer der Aufsynchronisierweg, desto höher die Aufsynchronisierbeschleunigung.
	ReturnPos	REAL	Die Folgeachse verfährt in diese Ausgangslage zu Beginn sowie nachdem sie aufsynchronisiert wurde und der Eingang MoveReturn den Zustand TRUE annimmt ^{*2*3}
	ReturnVel	REAL	Die Folgeachse verfährt mit der Geschwindigkeit ReturnVel in die Ausgangslage "ReturnPos" ^{*2 *3}
	ReturnAcc	REAL	Die Folgeachse fährt mit der Beschleunigung ReturnAcc in die Ausgangslage "ReturnPos" ^{*2 *3}
	PreSyncPos	REAL	Das Ausgangssignal "PreSyncSignal" wird zur Vorbereitung der Synchronfahrt vor dem Synchronpunkt "PreSyncPos" mit der hier angegebenen Entfernung gesetzt (TRUE) ^{*2}
	PreSyncTime	TIME	Zeitdauer von "PreSyncSignal"
VAR_OUTPUT	InSync	BOOL	Die Folgeachse läuft synchron mit dem Material. Sie befindet sich im Synchronisationsfenster (siehe Antriebsparameter S-0-0228)
	InCycle	BOOL	Die Folgeachse führt gerade einen Schnittzyklus aus. Ein Sofort- oder Kopfschnitt wird erst im nächsten Zyklus ausgeführt (CutCycle = 0)
	CropCutDone	BOOL	Kopfschnitt ist ausgeführt
	ImmediateCutDone	BOOL	Sofortschnitt ist ausgeführt
	PreSyncSignal	BOOL	Vorsignal vor dem Synchronpunkt "PreSyncPos"
	ShortPrdWarning	BOOL	Die Folgeachse hat nicht genügend Zeit, um die Ausgangslage "ReturnPos" für den nächsten Schnitt zu erreichen. Die Rückfahrt wird unterbrochen, und die Aufsynchronisierrampe beginnt erneut. Dieser Fall sollte vermieden werden, da der Antrieb sonst überlastet wird und die Trennstation an das Wagenbetten- de fahren könnte. Gegenmaßnahmen: • Rückfahrgeschwindigkeit erhöhen (Funktionsbaustein-Eingang "ReturnVel") • Rückfahrbeschleunigung erhöhen (Funktionsbaustein-Eingang "ReturnAcc") • Aufsynchronisierweg verkürzen (Funktionsbaustein-Eingang "SyncDist") • Materialvorschubgeschwindigkeit verringern

ML_TechMotion.library

E/A-Typ	Name	Datentyp	Erläuterung
	MaterialMoving	BOOL	Die Drehzahl des Material-Gebers ist höher als 5 U/ min. Ein Sofortschnitt bei Materialstillstand ist nur zulässig, wenn MaterialMoving=FALSE
	Reserve	DINT	Reserve-Inkremente des Aufsynchronisiervorgangs zu Diagnosezwecken. Aufsynchronisierung nicht möglich, wenn Reserve ≤ 0 -> In diesem Fall gibt der FB eine Fehlermeldung aus
	CycleState	UINT	Aktueller Zustand des Schneidezyklus: 0: Stillstands- und Wartephase 1: Beschleunigungsphase 2: Synchronlaufphase 3: Rücklaufphase
	CutCounter	UINT	Mit jedem Schnitt wird der Schnitzzähler "CutCounter" erhöht. Durch (Start = FALSE) oder eine positive Flanke am Eingang "ResetCutCounter" wird der Zähler rückgesetzt
	Error	BOOL	Weist auf einen Fehler hin. Wird gelöscht wenn Start=FALSE
	ErrorID	INT (Enum)	ERROR_CODE: Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

- *1.:

STRUCT (bCutNotCompleted: BOOL, diMasterSyncPosition: DINT, iRevCounter:INT)
- *2.:

Maßeinheiten je nach Antriebswichtung, z. B. mm
- *3.:

Neue Eingangswerte werden beim Übergang von der Synchronlauf- zur Rücklaufphase übernommen. Deshalb muss eine neue Schnittlänge, die im nächsten Zyklus aktiviert werden soll, zum Funktionsbaustein-Eingang kopiert werden, bevor die Achse in die Rücklaufposition gefahren wird.

Abb.6-4: Schnittstellenvariablen Funktionsbaustein MX(L)_FlyingShear

Signal-Zeit-Diagramm

Das nachstehende Diagramm zeigt den Signalablauf bei Sofortschnitt und stillstehendem Material (Start = FALSE):

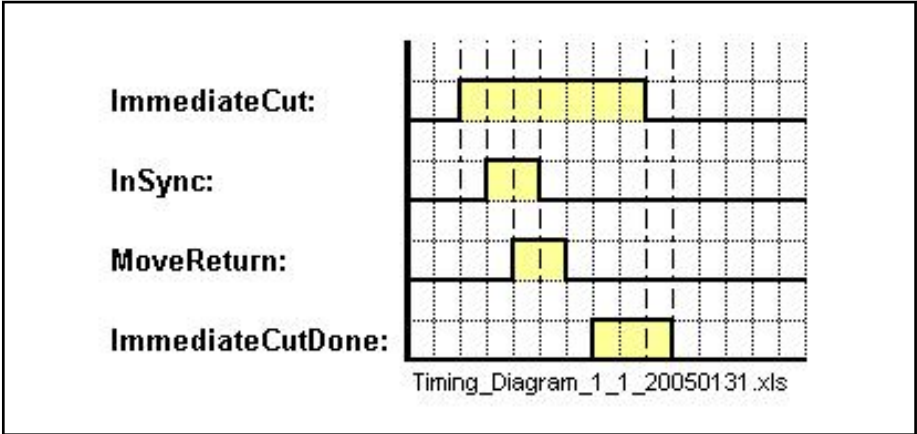


Abb.6-5: Signal-Zeit-Diagramm: Sofortschnitt bei Materialstillstand

Das folgende Diagramm zeigt den kompletten Signalablauf des Funktionsbausteins "Fliegende Säge" bei kontinuierlich durchlaufendem Material und Sofortschnitt:

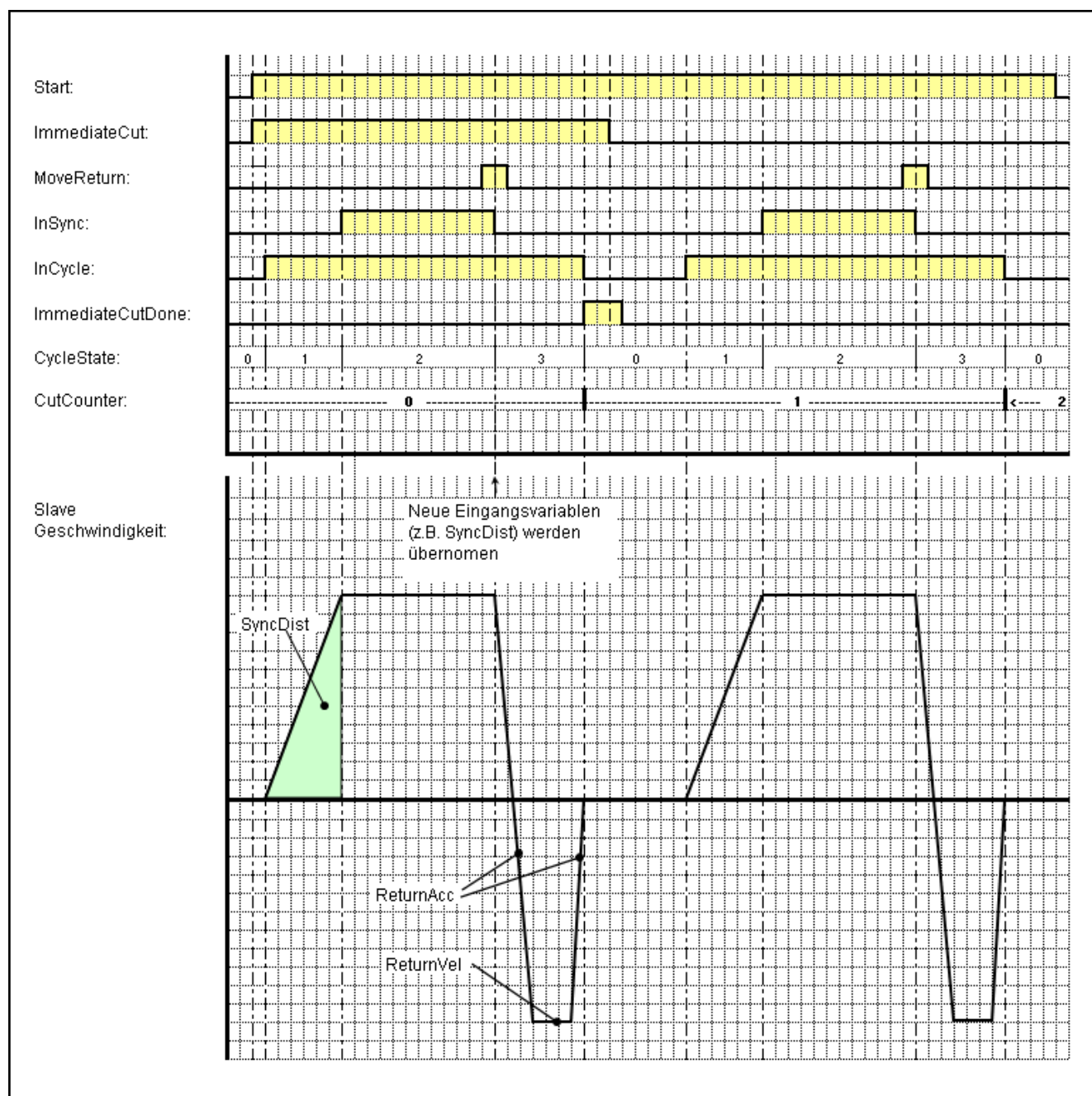


Abb.6-6: Kompletter Signalablauf des Funktionsbausteins "Fliegende Säge" bei Sofortschnitt

Fehlerbehandlung Der Funktionsbaustein "MX(L)_FlyingShear" erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb nicht freigeschaltet oder Antriebsfehler
ACCESS_ERROR (16#0004)	16#0003	16#0000	Funktionsbaustein wurde von einem anderen Funktionsbaustein unterbrochen

ML_TechMotion.library

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#0004	16#0000	Nicht unterstützte Antriebs-Firmware
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde während der Abarbeitung des Funktionsbausteins verändert
INPUT_RANGE_ERROR (16#0006)	16#0201	16#0001	CutLenght <= 0
INPUT_RANGE_ERROR (16#0006)	16#0201	16#0002	MWFeedconst <= 0
INPUT_RANGE_ERROR (16#0006)	16#0201	16#0003	SyncDist <= 0
INPUT_RANGE_ERROR (16#0006)	16#0201	16#0004	ReturnAcc <= 0
INPUT_RANGE_ERROR (16#0006)	16#0201	16#0005	ReturnVel <= 0
SYSTEM_ERROR (16#7FFF)	16#0202	16#0000	Synchronpunkt ist zu weit entfernt. Daher ist kein Aufsynchronisieren möglich. Fehler rücksetzen und Sofortschnitt veranlassen
ACCESS_ERROR (16#0004)	16#0203	16#0001	S-0-0048 ist nicht im optionalen zyklischen MDT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0002	P-0-0054 ist nicht im optionalen zyklischen MDT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0003	P-0-0691 ist nicht im optionalen zyklischen MDT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0004	P-0-0692 ist nicht im optionalen zyklischen MDT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0005	P-0-0052 ist nicht im optionalen zyklischen AT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0006	P-0-0053 ist nicht im optionalen zyklischen AT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0007	P-0-0761 ist nicht im optionalen zyklischen AT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0008	P-0-1367 Bit 6 ist nicht TRUE (Struktur AxisData wird nicht aktualisiert)
OTHER_ERROR (16#00FE)	16#0204	16#0001	Das ELS-Konfigurationswort (P-0-0086) der Folgeachse ist falsch konfiguriert. Die Konfiguration muss wie folgt lauten: P-0-0086 = 2#xxx01000

ErrorID	Additional1	Additional2	Beschreibung
OTHER_ERROR (16#00FE)	16#0204	16#0002	Parameter P-0-0060 der Folgeachse ist falsch konfiguriert. Der Wert muss lauten: P-0-0060=0
OTHER_ERROR (16#00FE)	16#0204	16#0003	Parameter P-0-0693 der Folgeachse ist falsch konfiguriert. Der Wert muss lauten: P-0-0693=0
OTHER_ERROR (16#00FE)	16#0204	16#0004	Parameter P-0-0750 der Folgeachse ist falsch konfiguriert. Der Wert muss lauten: P-0-0750=0
OTHER_ERROR (16#00FE)	16#0204	16#0005	Parameter P-0-0090 der Folgeachse ist falsch konfiguriert. Bit 2 von P-0-0090 muss FALSE sein
SYSTEM_ERROR (16#7FFF)	16#0205	16#0000	Sofortschnitt (im Stillstand) wurde angefordert, obwohl sich das Material weiterbewegte
OTHER_ERROR (16#00FE)	16#0206	16#0000	Der Parameter A-0-0200 der Slave Achse entspricht nicht der logischen Achsnummer der Masterachse

Abb.6-7: Fehlercodes Funktionsbaustein MX(L)_FlyingShear

6.2.3 ML_FlyingShear - Besonderheiten für IndraMotion MLC

Notwendige Hardware

- IndraMotion MLC Hardware CML65, CML45, CML25
- IndraDrive C oder M
- Zusätzliche Encoder-Interface-Karte zum Einlesen der Messradposition, falls Messgeber verwendet werden soll (kann entfallen, falls virtuelle oder reale Achse verwendet wird)
- Zusätzlicher Encoder zum Einlesen der Messradposition (gemäß IndraDrive Projektierung), falls Messgeber verwendet werden soll (kann entfallen, falls virtuelle oder reale Achse verwendet wird)



Ein hochauflösender Messgeber mit einer Auflösung von min. 4096 Inkremente/Umdrehung (möglichst Sinus-Encoder) wird empfohlen. Grob aufgelöste Gebersignale verursachen geringe Schnittgenauigkeit, sowie ggf. Laufgeräusche im Synchronbetrieb.

Notwendige Firmware

- IndraMotion MLC Firmware FWA-CMLXX*-MLC-12VRS
XX - Gerätevariante, z.B. CML65
- Antriebs-Firmware MPX07
- Die folgenden Funktionspakete müssen im Antrieb freigeschaltet sein
 - Closed Loop
 - Synchronization

Notwendige Software

IndraWorks 12VRS für IndraMotion MLC

Notwendige Parametrierung

Die nachfolgende Parametrierung muss am Antrieb vorgenommen werden, um eine ordnungsgemäße Funktion des Funktionsbausteins "MX(L)_FlyingShear" sicherzustellen.

ML_TechMotion.library

1. Anlegen der Slave-Achse in IndraWorks
Der Funktionsbaustein benötigt die Lagesollwertgenerierung im Antrieb. Daher darf beim Anlegen der Slave-Achse in IndraWorks die Checkbox "Interpolation im Antrieb" nicht abgewählt werden.
2. Laden des Parameterfiles "FlyingShearSettings.par"
Mit dem Laden des Parameterfiles "FlyingShearSettings.par" werden automatisch auf der Fliegenden Säge Achse wichtige Konfigurationen und die Kurvenscheiben geladen. Das Parameterfile wird über die Indraworks MLC Installations-CD 3 im Pfad "Add ons\ Tech FB_Parameterfile" bereitgestellt.
Das Parameterfile enthält folgende Informationen:
 - P-72 Lock On Cam Poly 5
 - P-92 Run Cam
 - P-750=1
 - P-693=0
 - P-60=0
 - P-86=0x0000.0000.0000.1000
 - P-90=0x0000.0000.0000.0000
 - A-500=P-691
 - A-502=P-692
3. Parametrierung Defaultleitachse
Im Parameter A-0-0200 der FlyingShear Slave Achse die logische Achsnummer der Masterachse eintragen
Über den IndraWorks Parametereditor den Parameter A-0-0200 der Slave Achse mit der logischen Achsnummer der Masterachse beschreiben.
4. Parametrierung der Fliegenden Säge Achse
Parametrierung der Fliegenden Säge gemäß Anwendung.
Wichtung sollte linear, absolut eingestellt werden.
5. Parametrieren des optionalen Encoders
IndraWorks ► Hinzufügen einer Geberachse in das Projekt
Die Wichtung sollte rotatorisch Modulo 360° eingestellt werden.
Einstellen der Encoder-Parameter gemäß Encoderdaten
6. Sicherstellen der Encoder Bewegungsrichtung
Der optionale Encoder (Messrad) muss sich im Normalbetrieb in positive Richtung bewegen. Die Bewegungsrichtung kann über den IndraWorks-Dialog der Geberachse beobachtet werden. Der Bewegungssinn muss invertiert werden, falls sich der Geber in negative Richtung bewegt.
7. Aufsynchonisierbeschleunigung (P-0-0142) und Aufsynchonisiergeschwindigkeit (P-0-0143) mit möglichst großen Werten parametrieren um einen dynamischen Aufsynchonisiervorgang sicherzustellen
IndraWorks ► Parametereditor.
8. Synchronlauffenster (S-0-0228) möglichst klein parametrieren, um eine möglichst hohe Schnittgenauigkeit zu erreichen
IndraWorks ► Parametereditor.

- Notwendige IndraLogic-Schritte**
- Der Funktionsbaustein "MX(L)_FlyingShear" enthält hochpriorie Motion-Befehle und sollte deshalb in einer hochpriorien zyklischen SPS-Task mit einer Zykluszeit $\leq 4\text{ms}$ zyklisch aufgerufen werden (MotionTask)
 - Vor Start des FB muss der "MC_Power" fehlerfrei abgearbeitet werden

6.3 Funktionsbausteine zur Anwendung "Zugspannungsregler"

6.3.1 Einführung und Übersicht

Der Zugspannungsregler (Kap. 6.3.2 "MB_TensionControlLoadCellType01" auf Seite 175) regelt, abhängig vom Messwert einer Kraftmessdose, die Zugspannung auf den gewünschten Sollwert. Der Istwert muss dem Regler zur Verfügung gestellt werden, und wird in der Regel über einen Analog- oder Feldbus-Eingang erfasst.

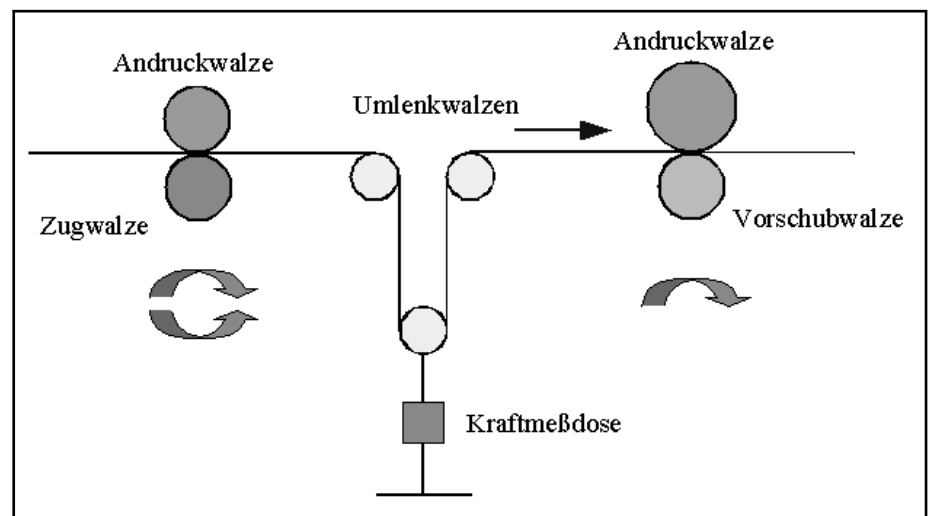


Abb.6-8: Prinzipdarstellung zur Zugkraftregelung

Der Zugregler ist als PI-Regler ausgeführt (wahlweise adaptiv). Der Reglerausgang wirkt auf den Getriebefeinabgleich der Folgeachse (Zugwalze). Zusätzlich kann der Regler auch bei Stillstand die Zugspannung halten bzw. die Bahn spannen.

Die Folgeachse kann sowohl in geschwindigkeitssynchroner oder winkelsynchroner Betriebsart betrieben werden.

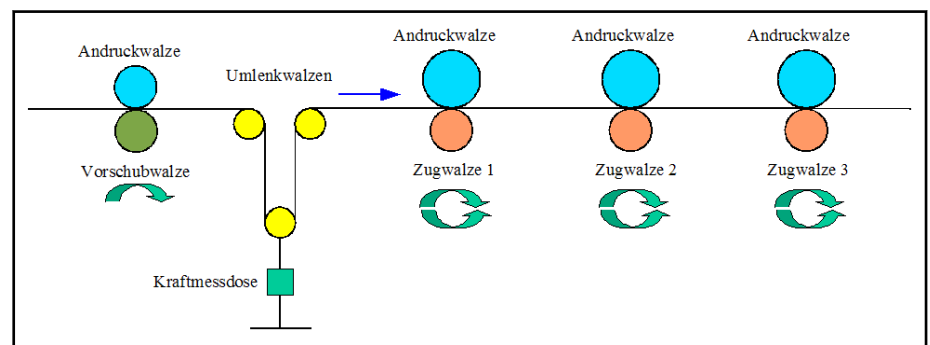


Abb.6-9: Multiachs-Zugspannungsregler

6.3.2 MB_TensionControlLoadCellType01

Kurzbeschreibung Der Zugspannungsregler regelt abhängig vom Messwert einer Kraftmessdose die Zugspannung auf den gewünschten Sollwert.

ML_TechMotion.library

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.6-10: Referenztabelle



Ab IndraMotion MLC 12VRS werden auch Slave-Achsen mit Interpolation auf der Steuerung unterstützt.

Schnittstellenbeschreibung

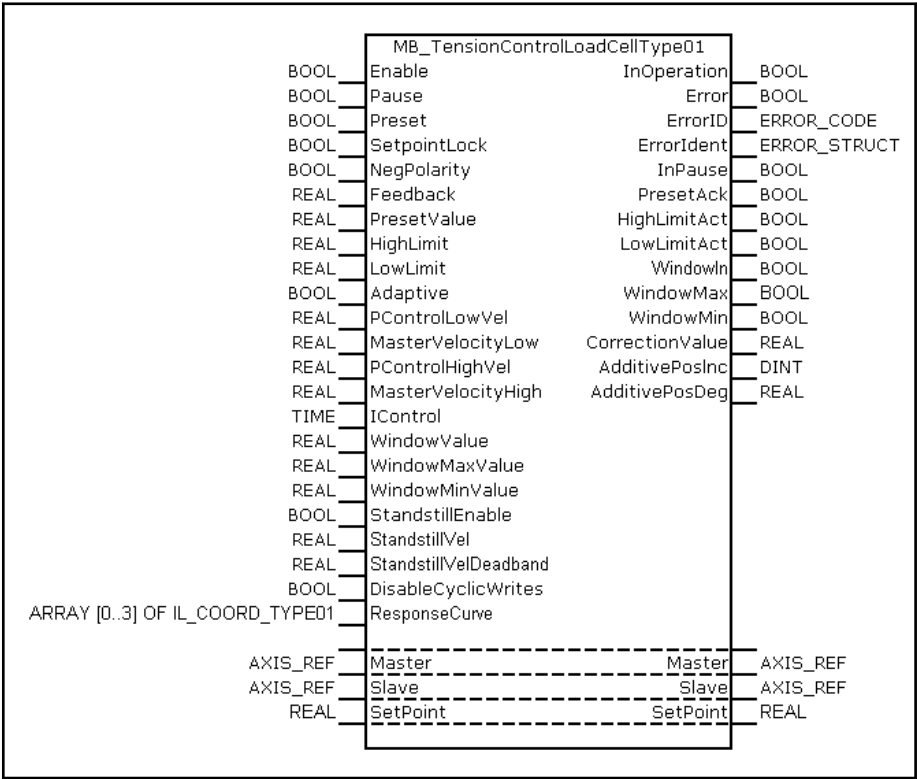


Abb.6-11: Funktionsbaustein MB_TensionControlLoadCellType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Master	AXIS_REF	Referenz zur Leitachse
	Slave	AXIS_REF	Referenz zur Folgeachse
	Setpoint	REAL	Sollwert der Zugspannung Diese wird in der Regel durch den Anwender gesetzt, kann aber auch vom Funktionsbaustein selbst geändert werden (siehe "SetpointLock")
VAR_INPUT	Enable	BOOL	Wird dieser Eingang gesetzt, arbeitet der Funktionsbaustein

E/A-Typ	Name	Datentyp	Beschreibung
	Pause	BOOL	Bei gesetztem Eingang wird der Zugspannungsregler angehalten und der I-Anteil des Ausgangswertes eingefroren. Ausgang "InPause" signalisiert ein aktives "Pause"
	Preset	BOOL	Bei der steigenden Flanke dieses Eingangs wird der Reglerausgang auf den in PresetValue parametrierbaren Wert gesetzt und bei Erfolg der Ausgang "PresetAck" gesetzt
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben. Der Wert wird nur bei steigender Flanke von "Enable" ausgewertet
	SetpointLock	BOOL	Bei steigender Flanke am Eingang wird der aktuelle Zugspannungswert an "Feedback" als neuer Sollwert für den Zugspannungsregler übernommen
	NegPolarity	BOOL	Bei gesetztem Eingang wird die Stellgröße "CorrectionValue" und "AdditivePosInc" mit einem negativen Vorzeichen versehen
	Feedback	REAL	aktueller Istwert der Zugspannung
	PresetValue	REAL	Wert auf den der Reglerausgang gesetzt wird, wenn der Eingang "Preset" gesetzt ist
	HighLimit	REAL	Maximalwert den der Reglerausgang annehmen kann. Wird die vom Regler berechnete Stellgröße größer als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	LowLimit	REAL	Minimalwert den der Reglerausgang annehmen kann. Wird die vom Regler berechnete Stellgröße kleiner als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	Adaptive	BOOL	Bei gesetztem Eingang wird eine adaptive P-Verstärkung für die Regelung verwendet
	PControlLowVel	REAL	P-Verstärkung des Reglers wenn "Adaptive" = FALSE Ansonsten entspricht der Wert der P-Verstärkung bei niedrigen Leitachsgeschwindigkeiten
	MasterVelocityLow	REAL	Nur wirksam wenn "Adaptive" = TRUE. Leitachsgeschwindigkeit bis zu der "PControlLowVel" verwendet wird
	PControlHighVel	REAL	P-Verstärkung des Reglers bei hohen Leitachsgeschwindigkeiten. Wenn "Adaptive" = FALSE, hat dieser Eingang keine Wirkung
	MasterVelocityHigh	REAL	Nur wirksam, wenn "Adaptive" = TRUE. Leitachsgeschwindigkeit ab der "PControlHighVel" verwendet wird

ML_TechMotion.library

E/A-Typ	Name	Datentyp	Beschreibung
	IControl	TIME	Nachstellzeit des I-Reglers Wird der Wert auf 0 sec. gesetzt, wird der I-Regler unwirksam
	WindowValue	REAL	Prozessgrößenfenster
	WindowMaxValue	REAL	Maximalwert der Prozessgröße
	WindowMinValue	REAL	Minimalwert der Prozessgröße
	StandstillEnable	BOOL	Bei gesetztem Eingang ist die in "StandstillVel" angegebenen additive Geschwindigkeit wirksam
	StandstillVel	REAL	Additive Geschwindigkeit für die Regelung im Stillstand
	StandstillVelDeadband	REAL	Dieser Wert bestimmt den Wert der Regelabweichung unterhalb dessen die Spanngeschwindigkeit bei der Regelung im Stillstand reduziert wird
VAR_OUTPUT	InOperation	BOOL	Der Zugspannungsregler arbeitet und die Ausgänge sind gültig
	Error	BOOL	Signalisiert einen Fehler Mit "Enable" = FALSE wird der Fehler gelöscht
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung gemäß Fehlertabelle (siehe Fehlerbehandlung auf Seite 186)
	InPause	BOOL	"Pause" ist aktiv
	PresetAck	BOOL	Ist dieser Ausgang aktiv, wurde der "Preset" Eingang ausgewertet und der Ausgang "CorrectionValue" hat den Wert "PresetValue" angenommen
	HighLimitAct	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße größer als der in "HighLimit" angegebene Wert ist
	LowLimitAct	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße kleiner als der in "LowLimit" angegebene Wert ist
	WindowIn	BOOL	Wenn TRUE, befindet sich die Prozessgröße innerhalb des parametrisierten Fensters
	WindowMax	BOOL	Wenn TRUE, befindet sich die Prozessgröße oberhalb des parametrisierten Maximalwertes
	WindowMin	BOOL	Wenn TRUE, befindet sich die Prozessgröße unterhalb des parametrisierten Minimalwertes
	CorrectionValue	REAL	Stellgröße des Zugspannungsreglers
	AdditivePosInc	DINT	Additive Geschwindigkeit in Inkrementen je SERCOS-Zyklus

Abb. 6-12: Schnittstellenvariablen Funktionsbaustein MB_TensionControlLoadCellType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Pause	BOOL			FALSE	kontinuierlich
Preset	BOOL			FALSE	kontinuierlich
DisableCyclicWrites	BOOL			FALSE	steigende Flanke an "Enable"
SetPointLock	BOOL			FALSE	kontinuierlich
NegPolarity	BOOL			FALSE	kontinuierlich
Feedback	REAL	n.def.	n.def.	0.0	kontinuierlich
PresetValue	REAL	n.def.	n.def.	0.0	steigende Flanke an "Preset"
HighLimit	REAL	>LowLimit	n.def.	200.0	kontinuierlich
LowLimit	REAL	-95.0	<HighLimit	-50	kontinuierlich
Adaptive	BOOL			FALSE	kontinuierlich
PControlLowVel	REAL	>0.0	n.def.	1.0	kontinuierlich
MasterVelocity-Low	REAL	>0.0	<=MasterVelocity-High	1.0	kontinuierlich
PControlHighVel	REAL	>0.0	n.def.	1.0	kontinuierlich
MasterVelocity-High	REAL	>=MasterVelocity-Low	n.def.	1.0	kontinuierlich
IControl	TIME	0s	n.def.	0s	kontinuierlich
WindowValue	REAL	0.0	100.0	0.0	kontinuierlich
WindowMaxValue	REAL	>=WindowMinValue	100.0	0.0	kontinuierlich
WindowMinValue	REAL	0.0	<=WindowMaxValue	0.0	kontinuierlich
StandstillEnable	BOOL			FALSE	kontinuierlich
StandstillVel	REAL	0.0	n.def.	0.0	kontinuierlich
StandstillDeadband	REAL	0.0	n.def.	1.0	kontinuierlich

Abb.6-13: Eingangsverhalten MB_TensionControlLoadCellType01

Funktionsbeschreibung

Im Folgenden werden die einzelnen Funktionen der Eingänge des Funktionsbausteins näher erläutert. Allgemein gilt für alle Eingänge, dass sie kontinuierlich bei gesetztem "Enable"-Eingang ausgewertet werden. Das heißt, Änderungen der Eingänge werden sofort ohne erneutes Toggeln des "Enable"-Eingangs wirksam.

Ausnahme dieser Regel bildet der "PresetValue"-Eingang, der nur bei steigender Flanke am "Preset"-Eingang übernommen wird. Ebenso wird der Eingang "DisableCyclicWrites" nur bei steigender Flanke am "Enable"-Eingang übernommen.

Die gesamte Wirkungsweise des Zugspannungsreglers ist in den folgenden Flussdiagrammen dargestellt. Je nach Eingang "StandstillEnable" ist entwe-

ML_TechMotion.library

der der Pfad "InVelocity" oder "Regelung im Stillstand" (Standstill Control) wirksam.

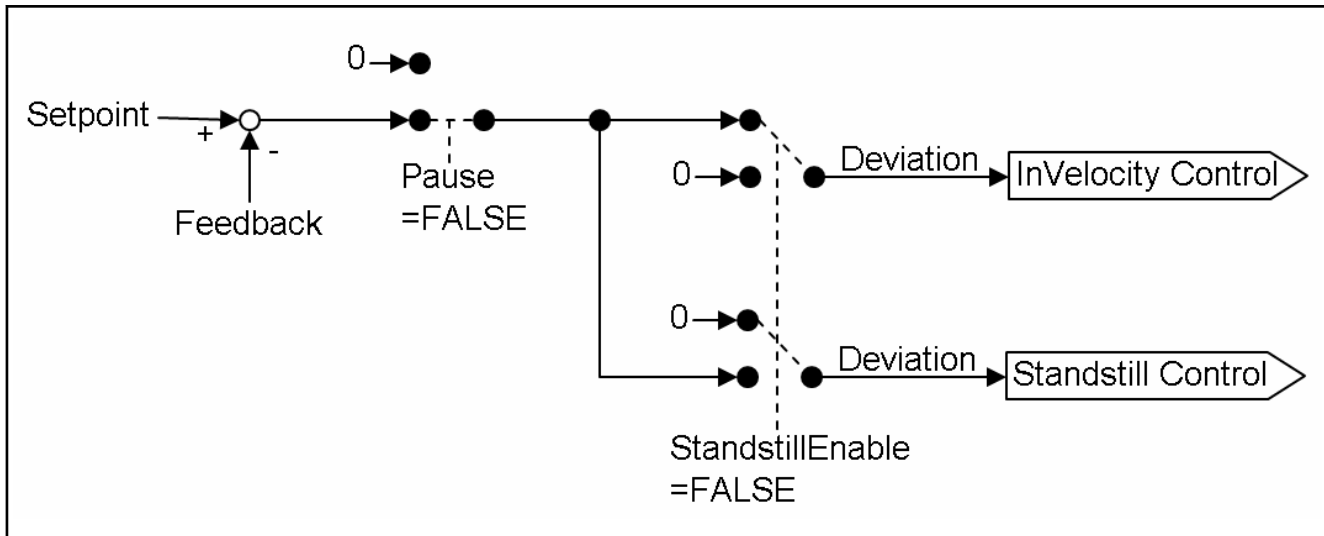


Abb.6-14: Flussdiagramm des Zugspannungsreglers mit Abzweig für Regelung im Stillstand / In Geschwindigkeit

Der Pfad "InVelocity Control" ist wirksam, wenn der Eingang "Standstill-Enable" nicht gesetzt ist. Die Zugspannung wird hierbei über den Parameter "Getriebefeinabgleich" P-0-0694 (Slave-Achse mit Interpolation auf dem Antrieb / IPO Drive) bzw. A-0-2605 (Slave-Achse mit Interpolation auf der Steuerung / IPO Control) geregelt. Es wird direkt auf die Parameter geschrieben, wenn der Eingang "DisableCyclicWrites" nicht gesetzt ist. Bei realen Achsen mit Interpolation auf dem Antrieb ist daher auch die Konfiguration des Parameters P-0-0694 in den optionalen zyklischen SERCOS-Daten (MDT) erforderlich.

Die Stellgröße wird zusätzlich auch über den Ausgang "CorrectionValue" ausgegeben.

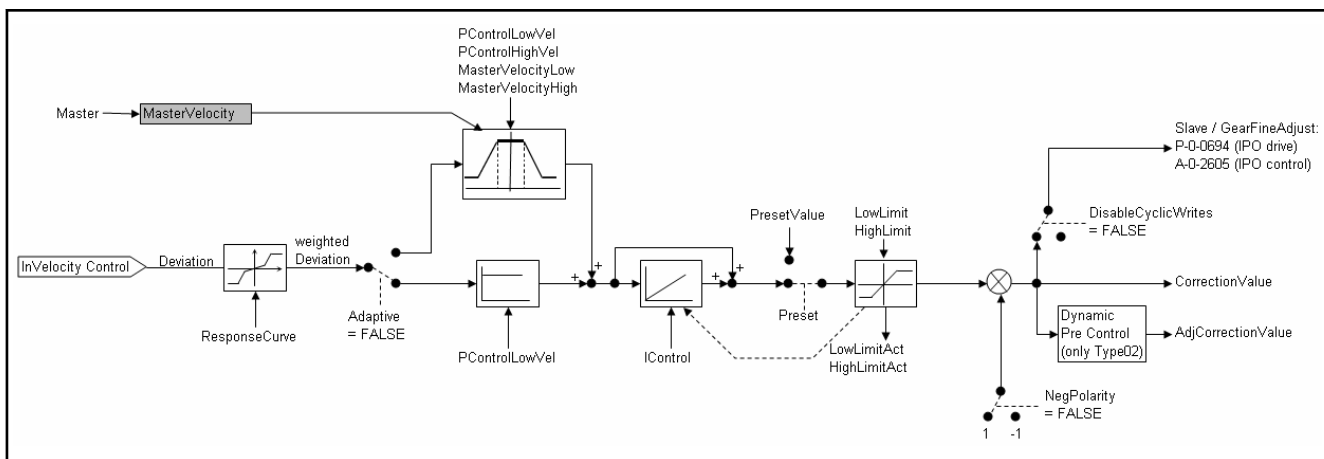


Abb.6-15: Flussdiagramm des Zugspannungsreglers für den Zweig "InVelocity Control"

Für die Regelung im Stillstand kann zusätzlich ein weiterer Pfad aktiv geschaltet werden, in dem auf die additive Leitachsposition (P-0-0692) geschrieben wird sofern "DisableCyclicWrites" inaktiv ist. Gleichzeitig steht der Wert immer über den Ausgang "AdditivePosInc" zur Verfügung.

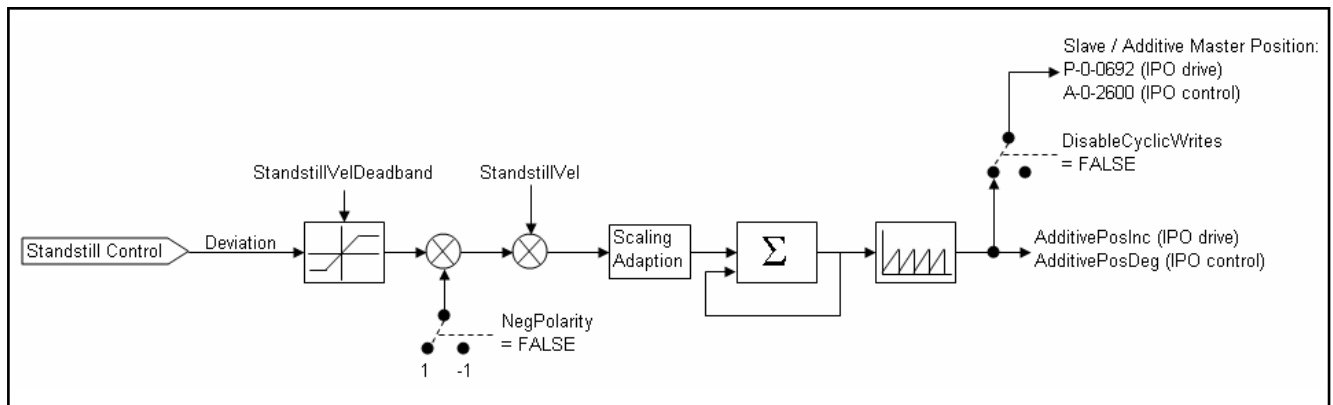


Abb.6-16: Flussdiagramm des Zugspannungsreglers für die Regelung im Stillstand (Standstill Control)

Das Flussdiagramm enthält aus Gründen der Übersichtlichkeit nicht die Begrenzungen "WindowValue", "WindowMaxValue", "WindowMinValue" sowie deren Statusanzeigen.

Inbetriebnahme

Ist "DisableCyclicWrites" inaktiv, müssen die beiden Parameter P-0-0694 (Getriebefeinabgleich) und P-0-0692 (additive Leitachposition) im zyklischen Sollwertkanal der Folgeachse aufgelegt werden. Bei steigender Flanke an "Enable" werden zyklische Datenkanäle überprüft. Wird einer der beiden Parameter nicht gefunden, erfolgt eine entsprechende Fehlermeldung an den Fehlerausgängen des Funktionsbausteins. Ist "DisableCyclicWrites" aktiv, wird die Konfiguration der zyklischen Sollwertdatenkanäle nicht überprüft.

Die verwendeten Achsen müssen vor dem Aktivieren des "Enable"-Eingangs in eine synchrone Betriebsart gebracht werden. Dies kann durch die PLCopen-Funktionsbausteine "GearIn" oder "GearInPos" geschehen.

Wird der Zugspannungsregler in der antriebsintegrierten Steuerung IndraMotion MLD verwendet, so muss die AxisData-Struktur im Antrieb aktiviert werden.

Preset/PresetValid

Durch eine positive Flanke am Eingang "Preset" wird ein Anfangswert über die Variable "PresetValue" für die Stellgröße vorgegeben. Gleichzeitig werden interne Variablen des Reglers so gesetzt, dass ein ruckfreies Anfahren des Reglers erfolgt. Der "Preset" wirkt nur auf den Getriebefeinabgleich im Pfad "InVelocity Control".

Der "Preset" Eingang wird nur ausgewertet, wenn "Enable" gesetzt ist. Bei steigender Flanke an "Preset" wird der neue Wert an "PresetValue" übernommen und "PresetAck" gesetzt. Eine fallende Flanke an "Preset" oder "Enable" löscht "PresetAck". Der "Preset"-Eingang ist dominant über "Pause", d.h. der Wert an "PresetValue" erscheint am Ausgang, auch wenn "Pause" gesetzt ist. Ist bei steigender Flanke an "Enable" "Preset" gesetzt, so wird "PresetValue" sofort als Stellgröße übernommen.



Die Preset Funktion sollte im Stillstand ausgeführt werden (StandstillEnable=TRUE), da ansonsten Geschwindigkeitssprünge durch das setzen des ControlValue auf den PresetValue entstehen können. Der Betrag der Geschwindigkeitssprünge ist proportional zu der Geschwindigkeit.

Der "Preset"-Eingang wirkt nur bei der steigenden Flanke.

ML_TechMotion.library

DisableCyclicWrites

Der Eingang "DisableCyclicWrites" bestimmt, ob die vom Funktionsbaustein berechneten Werte direkt über den zyklischen Kanal geschrieben werden (FALSE) oder nur am Bausteinausgang verfügbar sind (TRUE).

Damit ist es möglich mehrere Bausteine zu kaskadieren, indem man "DisableCyclicWrites" auf TRUE setzt. Dann kann man lediglich die Ausgänge ("CorrectionValue" und "AdditivePosInc") verwenden und weiter modifizieren. Allerdings ist es dann noch notwendig, dass der Anwender die Werte selbst in den optional zyklischen Kanal schreibt.

Ist "DisableCyclicWrites" auf TRUE gesetzt, wird außerdem die Prüfung, ob die optional zyklischen Parameter auch konfiguriert sind, nicht durchgeführt.

Adaptive, PControlLowVel, PControlHighVel, MasterVelocityLow, MasterVelocityHigh

Die P-Verstärkung kann wahlweise als konstante P-Verstärkung oder als adaptive P-Verstärkung vorgegeben werden. Über den Eingang "Adaptive" wird zwischen adaptiver oder konstanter P-Verstärkung gewählt. Durch Setzen von "Adaptive" auf TRUE wird die adaptive P-Verstärkung ausgewählt.

Ist eine konstante P-Verstärkung gewählt, ist nur der Wert in "PControlLowVel" für die P-Verstärkung wirksam.

Wird die adaptive P-Verstärkung gewählt, müssen die Eingänge "PControlLowVel", "PControlHighVel", "MasterVelocityLow" und "MasterVelocityHigh" beschaltet werden. Die Kennlinie sieht dann folgendermaßen aus:

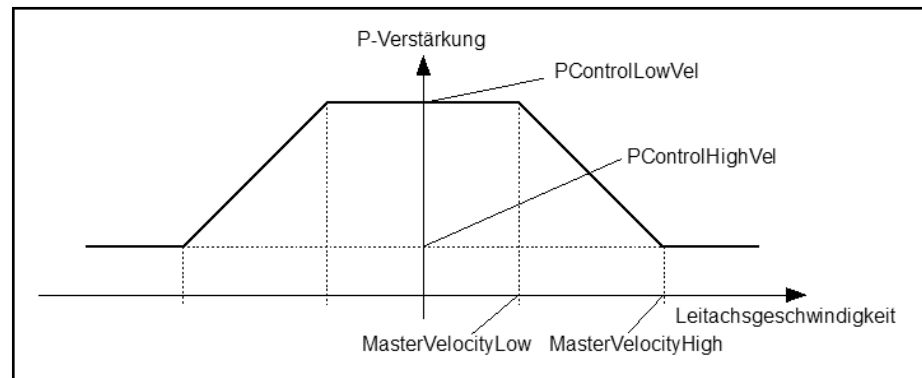


Abb.6-17: Kennlinie der adaptiven P-Verstärkung

Die einzelnen Eingänge beschreiben die Kennlinie für positive Leitachs-Geschwindigkeiten. Die Kennlinie für negative Leitachsgeschwindigkeiten ergibt sich aus der Spiegelung an der Achse der P-Verstärkung. Das heißt, es sind nur symmetrische Verläufe möglich. Die P-Verstärkung zwischen den Leitachsgeschwindigkeiten "MasterVelocityLow" und "MasterVelocityHigh" wird linear interpoliert. Dadurch ergeben sich einige Bedingungen für die Wertebereiche der Eingaben:

- Die Werte für "MasterVelocityLow" und "MasterVelocityHigh" müssen positiv sein
- Die Werte für "MasterVelocityLow" und "MasterVelocityHigh" dürfen nicht gleich sein
- Der Wert von "MasterVelocityLow" muss kleiner als der Wert von "MasterVelocityHigh" sein

Ist eine dieser Bedingungen nicht erfüllt, wird ein Fehler am Ausgang des Bausteines generiert.

SetpointLock

Durch Setzen des Einganges "SetpointLock" wird der aktuelle Zugspannungswert als neuer Sollwert für den Zugspannungsregler übernommen. Die Funktion erlaubt es, z.B. für den Sollwert der Zugspannung einen sinnvollen Startwert zu erhalten, der den momentanen Zustand der Maschine widerspiegelt.

Pause	Der Eingang "Pause" ermöglicht das Anhalten des Zugspannungsreglers. Wird der Eingang auf TRUE gesetzt, wird die Regelabweichung auf "0" gesetzt, d.h. der Reglerausgang wird auf dem aktuellen I-Anteil eingefroren. Ist "Pause" aktiv, wird der Ausgang "InPause" gesetzt.
NegPolarity	Der Eingang "NegPolarity" beeinflusst den Regelsinn. Abhängig von der Lage der Zugwalze, bezüglich der Durchlaufrichtung des Materials und der Lage der Vorschubwalze, muss der Regelsinn entweder positiv oder negativ sein. Ist dieser Eingang FALSE, wird der Regelsinn positiv und eine positive Regelabweichung führt zu einer positiven Stellbewegung. Der Eingang wirkt auch auf die Stillstandsregelung.
Feedback	Dieser Eingang stellt den Istwert der Zugspannung dar. Dieser kann z.B. durch einen Analogeingang am Antrieb oder den Inline I/O eingelesen werden. Der Anwender ist dabei für eine sinnvolle Skalierung verantwortlich.
PresetValue	Durch Setzen des Eingangs "Preset" wird ein Anfangswert über die Variable "PresetValue" für die Stellgröße vorgegeben. Gleichzeitig werden interne Variablen des Reglers so gesetzt, dass ein ruckfreies Anfahren des Reglers erfolgt.
HighLimit, HighLimitAct	Mit diesem Eingang wird die Stellgröße des Reglers auf einen Maximalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der maximale Wert wird durch den im Antrieb definierten Maximalwert für den Getriebefeinabgleich bestimmt. Tritt eine Begrenzung ein, wird der Ausgang "HighLimitAct" gesetzt.
LowLimit, LowLimitAct	Mit diesem Eingang wird die Stellgröße des Reglers auf einen Minimalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der minimale Wert liegt bei -95 unabhängig von ggf. kleineren Werten im Antrieb (siehe dazu auch Abschnitt "Getriebefeinabgleich" auf Seite 184). Tritt eine Begrenzung ein, wird der Ausgang "LowLimitAct" gesetzt.
PControlLowVel	Dieser Wert entspricht der Proportional-Verstärkung des Zugspannungsreglers, sofern der Eingang "Adaptive" nicht aktiv ist. Ansonsten stellt er die untere Verstärkung einer adaptiven P-Verstärkung dar.
WindowValue, WindowIn	Mit diesem Wert kann der Anwender ein symmetrisches Fenster definieren innerhalb dessen sich der aktuelle Zugspannungswert befinden soll. Das Fenster ist wie folgt definiert:

$$\left(SetPoint - |SetPoint| \cdot \frac{WindowValue}{100\%} \right) \leq Feedback \leq \left(SetPoint + |SetPoint| \cdot \frac{WindowValue}{100\%} \right)$$

Abb.6-18: WindowValue, WindowIn

Ist diese Bedingung erfüllt, wird der Ausgang "WindowIn" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt, es findet aber auch keinerlei Begrenzung statt.

WindowMaxValue, WindowMax	Mit diesem Wert kann der Anwender einen oberen Wert für den Zugspannungswert festlegen. Der Wert wird wie folgt ausgewertet:

$$Feedback > \left(SetPoint + |SetPoint| \cdot \frac{WindowMaxValue - 100}{100} \right)$$

Abb.6-19: WindowMaxValue, WindowMax

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMax" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt, es findet aber auch keinerlei Begrenzung statt.

ML_TechMotion.library

WindowMinValue, WindowMin

Mit diesem Wert kann der Anwender einen unteren Wert für den Zugspannungswert festlegen. Der Wert wird wie folgt ausgewertet:

$$Feedback < \left(SetPoint + |SetPoint| \cdot \frac{WindowMinValue - 100}{100} \right)$$

Abb.6-20: WindowMinValue, WindowMin

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMin" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt, es findet aber auch keinerlei Begrenzung statt.

StandstillEnable

Durch Setzen des Eingangs "StandstillEnable" wird die in "StandstillVel" angegebene Geschwindigkeit additiv auf die zu regelnde Achse aufgeschaltet. Damit ist eine Zugspannungsregelung auch im Stillstand möglich.



Der Getriebefeinabgleich verliert bei niedrigen Geschwindigkeiten immer mehr an Einfluss. Bei Stillstand der Leitachse besteht sogar ein offener Regelkreis, da der Getriebefeinabgleich dann keinen Einfluss mehr hat. Daher muss der Eingang "StandstillEnable" bei Erreichen des Stillstandsfensters gesetzt werden, damit im Pfad "InVelocity Control" der I-Anteil durch den offenen Regelkreis nicht bis zur Grenze ansteigt.



Bei Achsen mit Interpolation auf der Steuerung wird der Ausgang "AdditivePosInc" im Moduloformat 2²⁰ ausgegeben (ab Version 12).



Wird dieser Eingang verwendet, muss sichergestellt sein, dass der Zugspannungsregler nur in der SERCOS synchronen Task verwendet wird. Ist dies nicht der Fall, kann es zu unruhigem Lauf des Motors, oder Oszillieren um die Stillstandslage führen.

StandstillVel

Dieser Wert gibt die additive Geschwindigkeit an, die beim Setzen des Eingangs "StandstillEnable" an den Antrieb gesendet wird. Die Einheiten der Geschwindigkeit entsprechen dabei der im Antrieb eingestellten Wichtung.



Im Stillstand wird die Geschwindigkeit sprunghaft erzeugt. Da die Geschwindigkeit innerhalb eines SERCOS-Zyklus erreicht werden muss, können keine beliebig hohen Werte vorgegeben werden. Wird der Wert der Stillstandsgeschwindigkeit zu hoch gewählt, kann dies zur Oszillation des Antriebes führen. Der Maximalwert richtet sich dabei nach der Wichtung der Geschwindigkeit, sowie der eingestellten maximalen Beschleunigung und angeschlossenen Masse am Antrieb. Um zu große Geschwindigkeitssprünge zu vermeiden, kann eine Filterzeitkonstante im Antriebsparameter P-0-0693 definiert werden, wodurch der Sprung geglättet wird.

Hintergrundinformationen**Getriebefeinabgleich**

Der Zugspannungsregler arbeitet in der Regel mit Geschwindigkeiten größer 0. Dabei ist der Getriebefeinabgleich ein probates Mittel, um die Zugspannung aufrechtzuerhalten. Allerdings gibt es dabei auch einige Randbedingungen, die hier erläutert werden sollen.

Der Getriebefeinabgleich wird intern auf ein Minimum von -95% begrenzt. Wird ein kleinerer Wert eingegeben, erzeugt der Baustein einen Fehler.

Hintergrund dieser Begrenzung ist der errechnete Ausgangswert, der bei gegebenem Eingangswert wie folgt definiert ist:

$$\text{Ausgangswert} = \text{Eingangswert} * \left(1 + \frac{\text{Getriebefeinabgleich}}{100} \right)$$

Anhand der Gleichung kann man erkennen, dass - wenn der Getriebefeinabgleich gegen -100% geht - sich der Ausgangswert gegen 0 bewegt. Damit würde der Regelkreis praktisch geöffnet werden, da eine Änderung des Eingangswertes keinerlei Reaktion am Ausgang bewirkt. Schlimmer noch, bei einem Getriebefeinabgleich der kleiner als -100% wäre, kehrt sich das Vorzeichen des Ausgangswertes um. Im Regelkreis würde eine Mitkopplung entstehen, die das System instabil machen kann.

**StandstillEnable, StandstillVel,
StandstillVelDeadband**

Die Regelung im Stillstand kann durch Verwendung der Eingänge "StandstillEnable" und "StandstillVel" erfolgen. Dabei wird eine additive Geschwindigkeit aufgeschaltet. Um eine bessere Regelgüte zu erreichen, wird diese um den Sollwert herum verringert. Dies geschieht mit Hilfe einer abgewandelten Signum-Funktion, die nachfolgende Übertragungsfunktion besitzt:

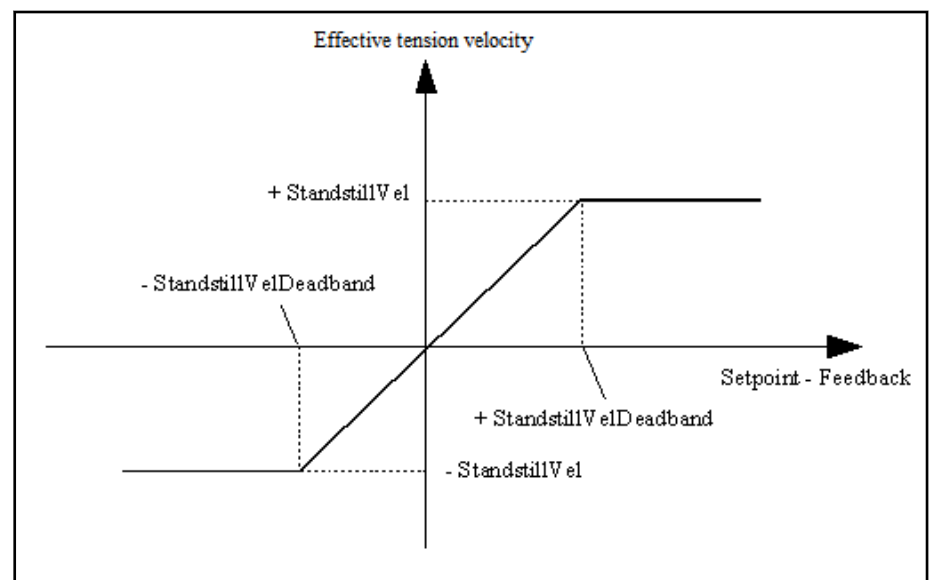


Abb.6-21: Abgewandelte Signum-Funktion für die Begrenzung der aufgeschalteten StandstillVel

Die Regelabweichung wird dadurch im Bereich des "Setpoint" relativiert. Nähert sich der "Feedback" dem "Setpoint", wird die Regelung "feinfühlicher" als ein reiner Zweipunktregler. Der Wert bei dem diese Begrenzung stattfindet, kann man am Eingang "StandstillVelDeadband" einstellen. Die Werte besitzen dabei dieselben Einheiten, wie der "Setpoint"-, bzw. "Feedback"- Eingang. Zwischen den Extremwerten "+StandstillVel" und "-StandstillVel" wird linear interpoliert, so dass sich obige Übertragungsfunktion für Regelabweichung und Spanngeschwindigkeit ergibt.

Anschließend wird der ermittelte Faktor mit der vorgegebenen "StandstillVel" multipliziert. Die Geschwindigkeit wird dabei als Drehzahl, mit der im Antrieb eingestellten Wichtung, verarbeitet. Möchte man anstatt der Drehzahl eine Bahngeschwindigkeit eingeben, ist diese vorher selbst zu berechnen.

Um eine umgekehrte Bewegungsrichtung für die Regelung im Stillstand zu erreichen, ist der Eingang "NegPolarity" zu setzen.

Nachfolgend wird noch mit einem Korrekturfaktor multipliziert. Dieser Korrekturfaktor setzt sich aus dem Getriebeübersetzungsverhältnis zusammen. Jedoch ist der Getriebefeinabgleich nicht berücksichtigt. In der Praxis ist der

ML_TechMotion.library

Getriebefeinabgleich in der Regel sehr klein, so dass dieser nicht ins Gewicht fällt. Das Getriebeübersetzungsverhältnis wird nur bei einer steigenden Flanke des "Enable"-Eingangs übernommen.

Im Flussdiagramm ist ersichtlich, dass die angegebene Geschwindigkeit in Inkremente umgerechnet wird. Dabei sind einige Regeln zu beachten:

- Die angegebene Geschwindigkeit kann nicht beliebig hoch gewählt werden, denn der Antrieb muss die Geschwindigkeit innerhalb eines SERCOS-Zyklus erreichen können. Damit sind nur sehr kleine Werte zulässig. Die Größe hängt dabei von der verwendeten Wichtung und der Masse am Motor ab. Die errechneten Inkremente werden über den Parameter P-0-0692 vorgegeben, dieser kann zudem über den im Antrieb verfügbaren Filter (P-0-0693) geglättet werden, so dass auch größere Sprünge möglich sind.
- Die errechneten Inkremente werden noch vor dem Getriebe aufaddiert. Das Getriebeübersetzungsverhältnis (A-0-2720 und A-0-2721 bzw. P-0-0156 und P-0-0157) wird dabei mit eingerechnet. Der Getriebefeinabgleich (P-0-0083 und P-0-0694) wird jedoch nicht berücksichtigt, da dieser in der Regel sehr klein ist. Das kann dazu führen, dass die Achse sich mit einer anderen, als der eingestellten Geschwindigkeit bewegt.
- Es wird eine Skalierung der Einheiten durchgeführt. Außerdem wird der eingegebene Geschwindigkeitswert nicht nach Maximum- und Minimumwerten überprüft. Dadurch können sich sehr große Werte für die Anzahl der Inkremente je SERCOS-Zyklus ergeben. Überschreitet der errechnete Zwischenwert die Maximalwerte von $\pm$ halber Modulowert, werden diese automatisch begrenzt. Die Folge ist eine niedrigere additive Geschwindigkeit, als am Eingang "StandstillVel" vorgegeben wird.
- Bei sehr kleinen Geschwindigkeiten ist außerdem zu beachten, dass die aktuelle Geschwindigkeit von der angegebenen Geschwindigkeit abweichen kann. Dies ist durch die Granulierung der Inkremente je SERCOS-Zyklus zu erklären: entspricht z.B. 1 Inkrement einer Geschwindigkeit von 0,03 U/min, so können nur Geschwindigkeiten die ein Vielfaches von 0,03 U/min sind, eingestellt werden. Ist dies nicht der Fall, wird zum nächstgelegenen Wert gerundet.
- Die errechneten Inkremente werden mit jedem SERCOS-Zyklus verarbeitet. Deshalb muss bei Verwendung der Regelung im Stillstand der Baustein in der SERCOS synchronen Task abgearbeitet werden. Ist dies nicht der Fall, wird sich der Antrieb einen SERCOS-Zyklus um die errechnete Anzahl an Inkrementen bewegen, und in den folgenden SERCOS-Zyklen wieder still stehen, was eine ruckende Bewegung zur Folge hat. Der Einsatz des Filters in P-0-0693 kann auch hier Abhilfe schaffen.

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle **F_RELATED_TABLE**, 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#00000A00	16#00000001	Der Parameter P-0-0694 ist nicht im optional zyklischen Telegramm der Folgeachse parametrier
ACCESS_ERROR (16#0004)	16#00000A00	16#00000002	Der Parameter P-0-692 ist nicht im optional zyklischen Telegramm der Folgeachse parametrier

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000001	Die Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") ist < 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000002	Die Leitachsgeschwindigkeit im oberen Betriebspunkt ("MasterVelocityHigh") ist < 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000003	Die Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") und oberen Betriebspunkt ("MasterVelocityHigh") besitzen die gleichen Werte
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000004	Der Wert der Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") ist größer als der Wert im oberen Betriebspunkt ("MasterVelocityHigh")
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000005	Der Wert des Eingangs "LowLimit" ist kleiner als der minimal zulässige (-95.0)
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000006	Der Wert der Eingänge "PGainLowVel" oder "PGainHighVel" ist kleiner oder gleich 0
DEVICE_ERROR (16#0008)	16#00000A02	16#00000007	Bei dem in "Slave" angegebenen Antrieb ist keine Leistung zugeschaltet. Eine Zugspannungsregelung ist daher nicht möglich
DEVICE_ERROR (16#0008)	16#00000A02	16#00000008	Der in "Slave" angegebene Antrieb befindet sich nicht in einer synchronen Betriebsart
RESOURCE_ERROR (16#0003)	16#00001503	16#00000000	AxisData ist in der Steuerung nicht aktiv. Dieser Fehler tritt nur bei der IndraMotion MLD auf

Abb.6-22: Fehlercodes Funktionsbaustein MB_TensionControlLoadCellType01

6.4 Funktionsbausteine zur Anwendung "Schlaufenregelung"

6.4.1 Einführung und Übersicht

Schlaufenregelungen werden zur Entkoppelung von Anlagenteilen eingesetzt.

Ein typischer Anwendungsfall ist nachstehende Bandanlage.

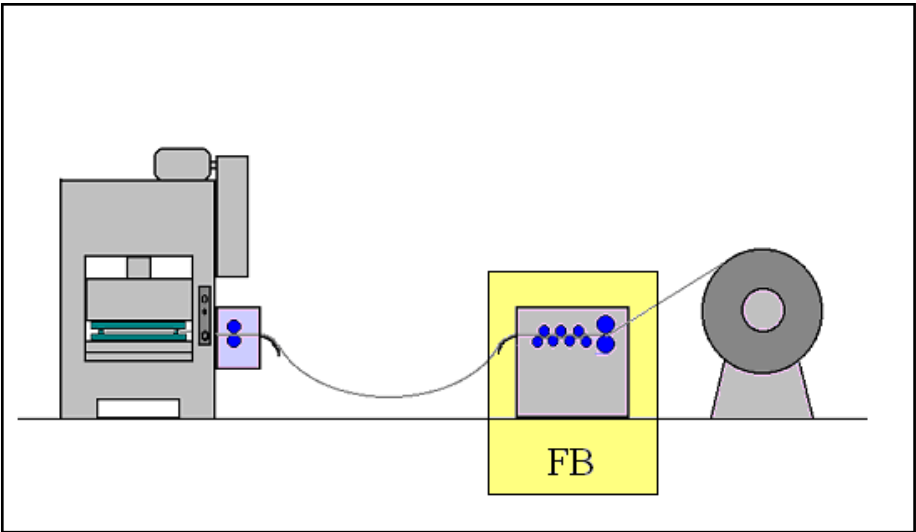


Abb.6-23: Bandanlage, Prinzipdarstellung

Die Schleife befindet sich zwischen den Vorschubwalzen der Stanzmaschine und dem vorgeschalteten Materialzufuhrsystem. Sie soll die beiden Bewegungssysteme (Start/Stop bzw. kontinuierlich) entkoppeln. Der Schlaufeninhalt hängt dabei von Walzenvorschubdaten ab.

Im Normalbetrieb läuft der eine Bereich der Bandanlage kontinuierlich mit einer vom Bediener vorgegebenen durchschnittlichen Geschwindigkeit v.

Der andere Anlagenbereich läuft im Start-Stop-Betrieb. Während der Blechbearbeitung geht dieser Teil der Anlage in Stopp und der Schlaufeninhalt wächst an. Wenn das Material für die nächste Bearbeitung nachgeführt wird, baut sich der Schlaufeninhalt wieder ab. In einem Arbeitstakt bewegt sich die Schleife immer zwischen den Min/Max-Kontakten.

Voraussetzung für den Einsatz des FB zur Schlaufenkontrolle:

Beim Einrichten der Bandanlage wird das neue Blechband vom Coil durch die Richtmaschine zur Stanzmaschine dergestalt geführt, dass bereits eine Schleife zwischen den Min/Max-Kontakten (oder Min/Max-Analogwerte) existiert.

- [ML_SagControlC](#), [Seite 188](#), regelt die Schleife mit Min/Max-Kontakten.
- [ML_SagControlA](#), [Seite 192](#), regelt die Schleife mit analogen Schaltsignalen.

6.4.2 ML_SagControlC

Kurzbeschreibung Der Funktionsbaustein "ML_SagControlC" enthält einen Schlaufenregler mit binären Min/Max-Kontakten.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library

Abb.6-24: Referenztablelle

Schnittstellenbeschreibung

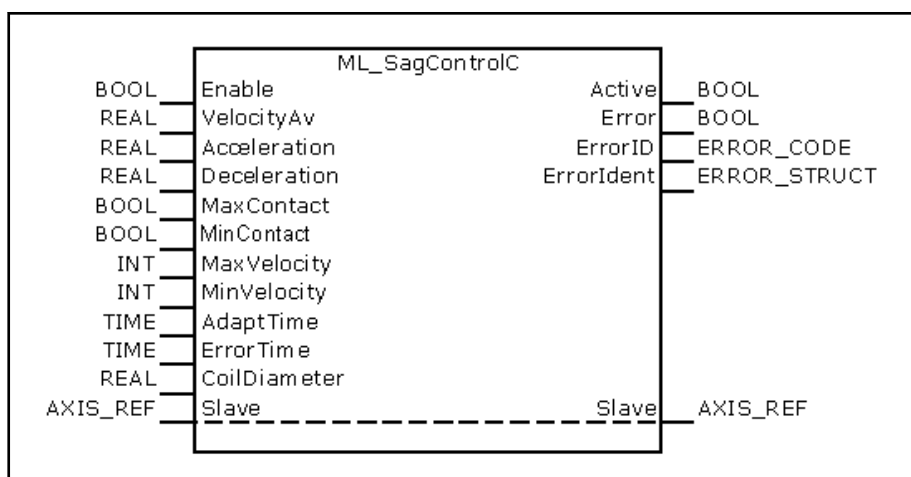


Abb.6-25: Funktionsbaustein ML_SagControlC

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Slave	AXIS_REF	Slave-Achse
VAR_INPUT	Enable	BOOL	Aktiviert/deaktiviert die Geschwindigkeitsregelung
	VelocityAv	REAL	Durchschnittsgeschwindigkeit der Bandanlage
	Acceleration	REAL	Beschleunigung für Geschwindigkeitsänderungen
	Deceleration	REAL	Verzögerung für Geschwindigkeitsänderungen
	MaxContact	BOOL	Max-Kontakt der Schleufe
	MinContact	BOOL	Min-Kontakt der Schleufe
	MaxVelocity	INT	Maximale Geschwindigkeit in % bezogen auf die Durchschnittsgeschwindigkeit der Bandanlage
	MinVelocity	INT	Minimale Geschwindigkeit in % bezogen auf die Durchschnittsgeschwindigkeit der Bandanlage
	AdaptTime	TIME	Adaptionszeit (Ta)
	ErrorTime	TIME	Fehlerzeit (Te)
	CoilDiameter	REAL	Coildurchmesser ⇒ Bewertung der Geschwindigkeit in %. Ist der Eingang nicht beschaltet, wird der Default-Wert 100% gesetzt
VAR_OUTPUT	Active	BOOL	Schlaufenregelung arbeitet
	Error	BOOL	Signalisiert einen Fehler. Mit "Enable"= FALSE wird Fehler gelöscht
	ErrorID	ERROR_CODE	INT (Enum): Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung in Abhängigkeit von der Fehlertabelle

Abb.6-26: Schnittstellenvariablen Funktionsbaustein ML_SagControlC

Funktionsbeschreibung

Der "Enable"-Eingang aktiviert/deaktiviert die Geschwindigkeitsregelung. Die erste aktive Geschwindigkeit ist die vom Bediener vorgegebene Durchschnittsgeschwindigkeit der Anlage.



Wird der "Enable"-Eingang zurückgesetzt, geht der zu regelnde Antrieb in Stop.

Der Funktionsbaustein für die Schleifenregelung wirkt ausschließlich auf den Antrieb für die Materialzufuhr. Hierbei wird versucht, die Schleife in einem definierten Fenster zu halten (Min/Max-Wert), indem je nach Abweichung die aktuelle Geschwindigkeit der Bandanlage angepasst wird, optional mit einem Sensor, der den Coildurchmesser erfasst.

Bei Erreichen des Min-Kontaktes wird die Geschwindigkeit des Materialzufuhrsystems auf die am Baustein anliegende Maximalgeschwindigkeit erhöht, um eine größere Schleife aufzubauen. Wird der Min-Kontakt innerhalb der Fehlerreaktionszeit " T_e " wieder in Richtung Max-Kontakt verlassen, so wird noch für die Nachhaltezeit " T_a " mit der erhöhten Geschwindigkeit gefahren, um dann wieder mit der vorgegebenen Durchschnittsgeschwindigkeit weiterzufahren. Vergrößert sich die Schleife so weit, dass der Max-Kontakt anspricht, so wird die Geschwindigkeit auf die am Baustein anliegende Minimalgeschwindigkeit reduziert. Bei Verlassen des Max-Kontaktes innerhalb der Fehlerreaktionszeit " T_e " fährt der Antrieb noch für die Zeit " T_a " mit reduzierter Geschwindigkeit weiter. Danach wird wieder mit der vorgegebenen Durchschnittsgeschwindigkeit weitergefahren (siehe v-t Diagramm).

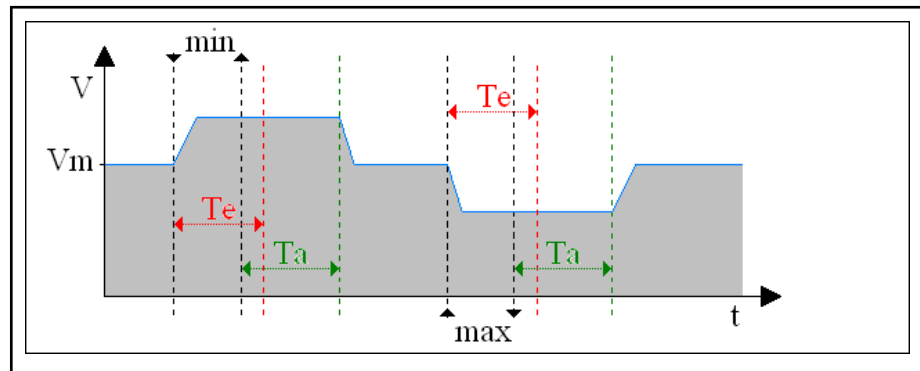


Abb.6-27: Bausteinverhalten, fehlerfrei

Fehlerbehandlung

Steht der Min-Kontakt für eine Zeit größer " T_e " immer noch an, so wird ein Fehler generiert und mit gleicher Geschwindigkeit weitergefahren. Dieser Zustand bleibt so lange bestehen, bis der Kontakt wieder verlassen wird. Hiernach fährt der Antrieb noch für die Nachhaltezeit " T_a " mit der erhöhten Geschwindigkeit und danach mit der vorgegebenen Durchschnittsgeschwindigkeit. Bleibt hingegen der Max-Kontakt für die Zeit " T_e " anstehen, so wird die Geschwindigkeit auf Null reduziert. Schaltet der Max-Kontakt wieder, so bleibt der Antrieb weiter für die Zeit " T_a " stehen, um dann wieder auf die vorgegebene Durchschnittsgeschwindigkeit " V_m " zu beschleunigen und mit dieser Geschwindigkeit weiterzufahren.

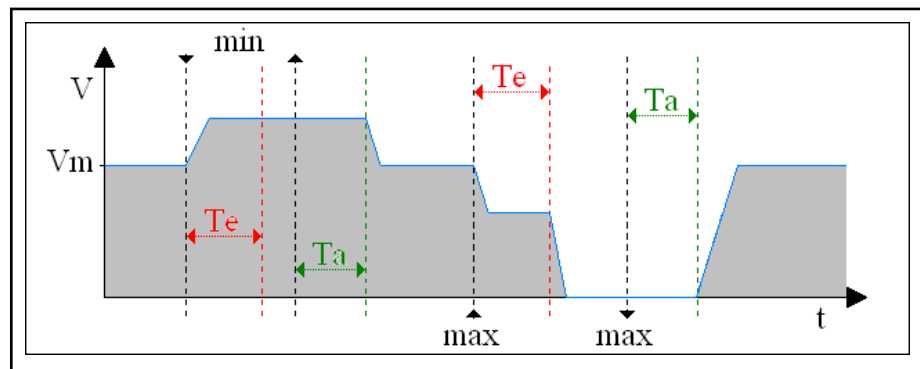


Abb.6-28: Bausteinverhalten, fehlerhaft

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle **F_RELATED_TABLE**, 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Eingänge außerhalb des Gültigkeitsbereichs: Geschwindigkeit <= 0 Beschleunigung <= 0 Verzögerung <= 0 Max. Geschw. <= Min. Geschw. Adaptionszeit <= 0 Fehlerreaktionszeit <= 0 Coildurchmesser <= 0
SYSTEM_ERROR (16#7FFF)	16#0501	16#0000	Geschwindigkeitsadaption erfolglos, Adaptionszeit und Fehlerzeit abgelaufen, Max-Kontakt steht noch an
SYSTEM_ERROR (16#7FFF)	16#0502	16#0000	Geschwindigkeitsadaption erfolglos, Adaptionszeit und Fehlerzeit abgelaufen, Min-Kontakt steht noch an
SYSTEM_ERROR (16#7FFF)	16#0503	16#0000	Max/Min-Kontakt stehen zur gleichen Zeit an. Kontaktproblem ?
SYSTEM_ERROR (16#7FFF)	16#0504	16#0000	Die vorgegebene Geschwindigkeit konnte nicht erreicht werden. Geschwindigkeitsvorgaben überprüfen

Abb.6-29: Fehlercodes Funktionsbaustein ML_SagControlC

Firm-, Soft- und Hardware-Anforderungen

Folgende Firm-, Soft- und Hardware wird benötigt:

- Hardware - IndraDrive C oder M
- Firmware - Antriebsfirmware (Funktionspaket Closed Loop)
- Software - IndraWorks Engineering 12VRS

Applikationsbeispiel 1

In der einfachsten Form sind es 2 Kontakte, ein Min- und ein Max-Kontakt, die als Regelparameter zur Verfügung stehen.

ML_TechMotion.library

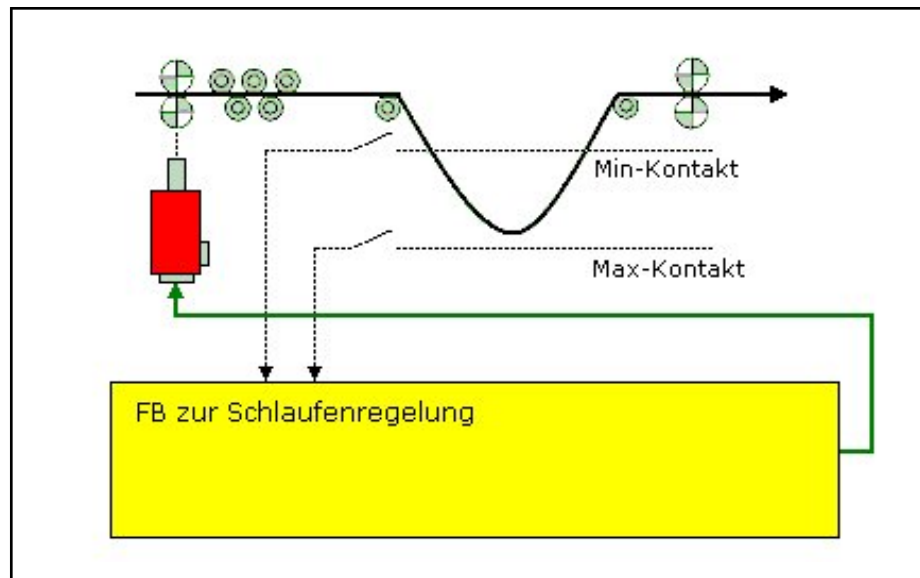


Abb.6-30: Anwendung 1, ML_SagControlC

Im Normalbetrieb läuft die Bandanlage mit einer durch den Bediener vorgegebenen durchschnittlichen Geschwindigkeit "Vm". Der FB für die Schlaufenregelung versucht nun, die Schlaufe in einem definierten Fenster zu halten (Min/Max-Kontakt), indem je nach Abweichung die aktuelle Geschwindigkeit der Bandanlage angepasst wird.

Applikationsbeispiel 2

Bei dieser Variante ist der zu regelnde Antrieb der Abwickler (Coil) selbst. Hierbei kommt neben den Min/Max-Kontakten auch der abnehmende Coildurchmesser als Regelgröße hinzu.

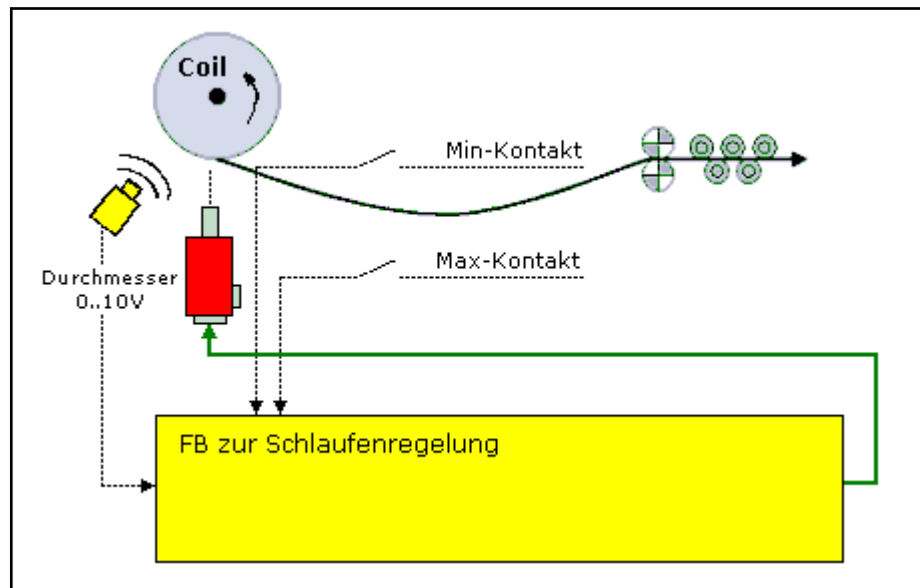


Abb.6-31: Anwendung 2, ML_SagControlC

6.4.3 ML_SagControlA

Kurzbeschreibung Der Funktionsbaustein "ML_SagControlA" enthält einen Schlaufenregler mit analoger Istwerterfassung der Schlaufe.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library

Abb.6-32: Referenztabelle

Schnittstellenbeschreibung

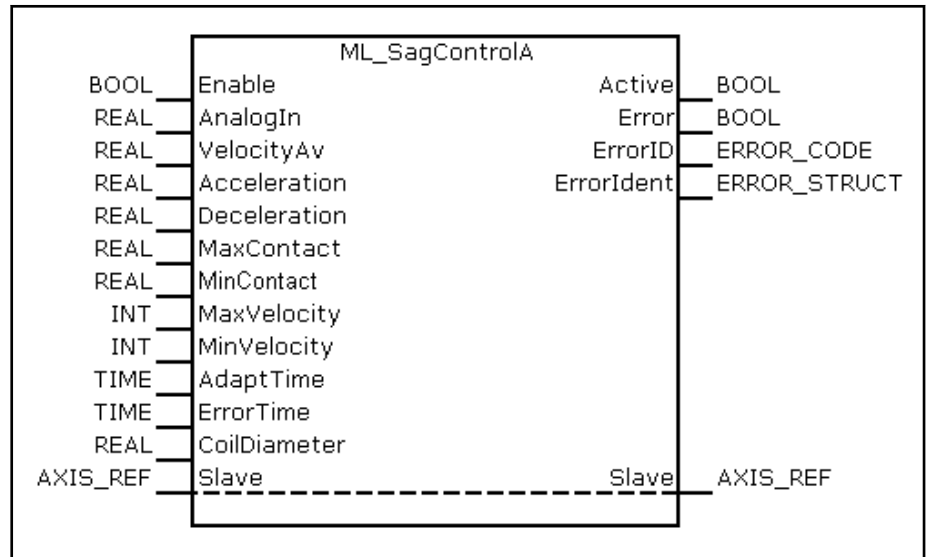


Abb.6-33: Funktionsbaustein ML_SagControlA

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Slave	AXIS_REF	Slave-Achse
VAR_INPUT	Enable	BOOL	Aktiviert / deaktiviert die Geschwindigkeitsregelung
	AnalogIn	REAL	Analogwert der Schlaufe (0..10)
	VelocityAv	REAL	Durchschnittsgeschwindigkeit der Bandanlage
	Acceleration	REAL	Beschleunigung für Geschwindigkeitsänderungen
	Deceleration	REAL	Verzögerung für Geschwindigkeitsänderungen
	MaxContact	REAL	Max-Kontakt Schlaufe
	MinContact	REAL	Min-Kontakt Schlaufe
	MaxVelocity	INT	Maximale Geschwindigkeit in % bezogen auf die Durchschnittsgeschwindigkeit der Bandanlage
	MinVelocity	INT	Minimale Geschwindigkeit in % bezogen auf die Durchschnittsgeschwindigkeit der Bandanlage
	AdaptTime	TIME	Adaptionszeit (Ta)
	ErrorTime	TIME	Fehlerzeit (Te)
	CoilDiameter	REAL	Coildurchmesser ⇒ Bewertung der Geschwindigkeit in %. Ist der Eingang nicht beschaltet, wird der Default-Wert 100% gesetzt
VAR_OUTPUT	Active	BOOL	Schlaufenregelung arbeitet
	Error	BOOL	Signalisiert einen Fehler. Mit "Enable"= FALSE wird Fehler gelöscht

ML_TechMotion.library

E/A-Typ	Name	Datentyp	Beschreibung
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung in Abhängigkeit von der Fehlertabelle

Abb.6-34: Schnittstellenvariablen Funktionsbaustein ML_SagControlA

Funktionsbeschreibung

Der Enable-Eingang aktiviert/deaktiviert die Geschwindigkeitsregelung. Die erste aktive Geschwindigkeit ist die vom Bediener vorgegebene Durchschnittsgeschwindigkeit der Anlage.



Wird der Enable-Eingang zurückgesetzt, geht der zu regelnde Antrieb in Stopp.

Der Funktionsbaustein für die Schleifenregelung wirkt ausschließlich auf den Antrieb für die Materialzufuhr. Hierbei wird versucht, die Schleife in einem definierten Fenster zu halten (Analogwert-Sensor), indem je nach Abweichung die aktuelle Geschwindigkeit der Bandanlage angepasst wird, optional mit einem Sensor, der den Coildurchmesser erfasst.

Bei Erreichen des Min-Kontaktes wird die Geschwindigkeit des Materialzufuhrsystems auf die am Baustein anliegende Maximalgeschwindigkeit erhöht, um eine größere Schleife aufzubauen. Wird der Min-Kontakt innerhalb der Fehlerreaktionszeit "Te" wieder in Richtung Max-Kontakt verlassen, so wird noch für die Nachhaltezeit "Ta" mit der erhöhten Geschwindigkeit gefahren, um dann wieder mit der vorgegebenen Durchschnittsgeschwindigkeit weiterzufahren. Vergrößert sich die Schleife so weit, dass der Max-Kontakt anspricht, so wird die Geschwindigkeit auf die am Baustein anliegende Minimalgeschwindigkeit reduziert. Bei Verlassen des Max-Kontaktes innerhalb der Fehlerreaktionszeit "Te" fährt der Antrieb noch für die Zeit "Ta" mit reduzierter Geschwindigkeit weiter. Danach wird wieder mit der vorgegebenen Durchschnittsgeschwindigkeit weitergefahren (siehe v-t Diagramm).

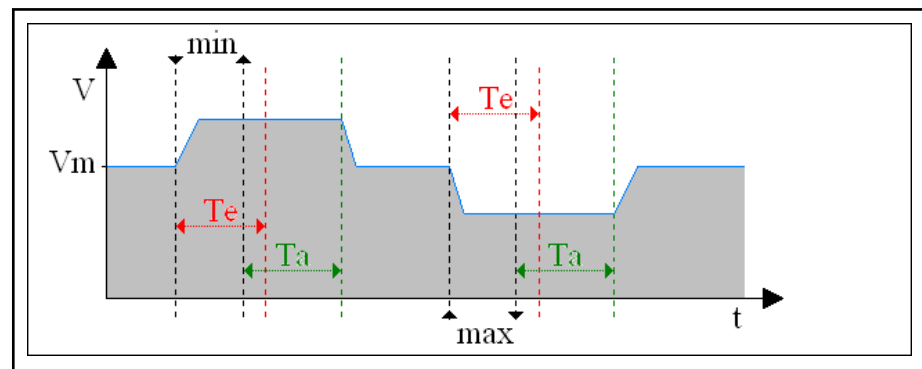


Abb.6-35: Bausteinverhalten, fehlerfrei

Fehlerbehandlung

Steht der Min-Kontakt für eine Zeit größer "Te" immer noch an, so wird ein Fehler generiert und mit gleicher Geschwindigkeit weitergefahren. Dieser Zustand bleibt so lange bestehen, bis der Kontakt wieder verlassen wird. Hier-nach fährt der Antrieb noch für die Nachhaltezeit "Ta" mit der erhöhten Geschwindigkeit und danach mit der vorgegebenen Durchschnittsgeschwindigkeit. Bleibt hingegen der Max-Kontakt für die Zeit "Te" anstehen, wird die Geschwindigkeit auf Null reduziert. Schaltet der Max-Kontakt wieder, bleibt der Antrieb weiter für die Zeit "Ta" stehen, um dann wieder auf die vorgegebenen Durchschnittsgeschwindigkeit "Vm" zu beschleunigen und mit dieser Geschwindigkeit weiterzufahren.

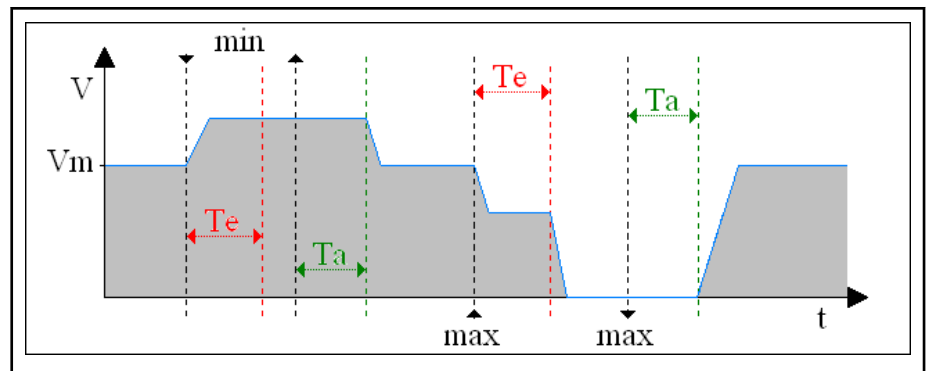


Abb.6-36: Bausteinverhalten, fehlerhaft

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle F_RELATED_TABLE, 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Eingänge außerhalb des Gültigkeitsbereichs: Geschwindigkeit ≤ 0 Beschleunigung ≤ 0 Verzögerung ≤ 0 Max. Geschw. $\leq$ Min. Geschw. Adaptionszeit ≤ 0 Fehlerreaktionszeit ≤ 0 Coil Durchmesser ≤ 0
SYSTEM_ERROR (16#7FFF)	16#0501	16#0000	Geschwindigkeitsadaption erfolglos, Adaptionszeit und Fehlerzeit abgelaufen, Max-Kontakt steht noch an
SYSTEM_ERROR (16#7FFF)	16#0502	16#0000	Geschwindigkeitsadaption erfolglos, Adaptionszeit und Fehlerzeit abgelaufen, Min-Kontakt steht noch an
SYSTEM_ERROR (16#7FFF)	16#0504	16#0000	Die vorgegebene Geschwindigkeit konnte nicht erreicht werden. Geschwindigkeitsvorgaben überprüfen

Abb.6-37: Fehlercodes Funktionsbaustein ML_SagControlA

Firm-, Soft- und Hardware-Anforderungen

Folgende Firm-, Soft- und Hardware wird benötigt:

- Hardware - IndraDrive C oder M
- Firmware - Antriebsfirmware (Funktionspaket Closed Loop)
- Software - IndraWorks Engineering 12VRS

Applikationsbeispiel 1

In dieser Variante bildet ein Sensor die Schlaufenlage als analogen Eingang (0-10V) ab, der zur Geschwindigkeitsregelung herangezogen wird.

ML_TechMotion.library

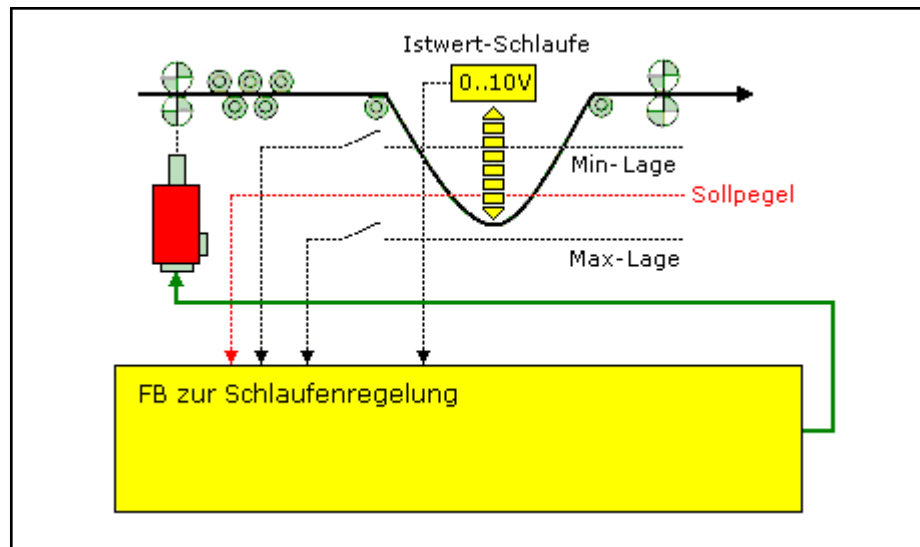


Abb.6-38: Anwendung 1, ML_SagControlA

Im Normalbetrieb läuft die Bandanlage mit einer durch den Bediener vorgegebenen durchschnittlichen Geschwindigkeit V_m . Der FB für die Schlaufenregelung versucht nun, die Schleife in einem definierten Fenster zu halten (Min/Max-Lage), indem je nach Abweichung die aktuelle Geschwindigkeit der Bandanlage angepasst wird.

Applikationsbeispiel 2

Bei dieser Variante ist der zu regelnde Antrieb der Abwickler (Coil) selbst. Hierbei kommt neben dem Analogwert-Eingang auch der abnehmende Coildurchmesser als Regelgröße hinzu.

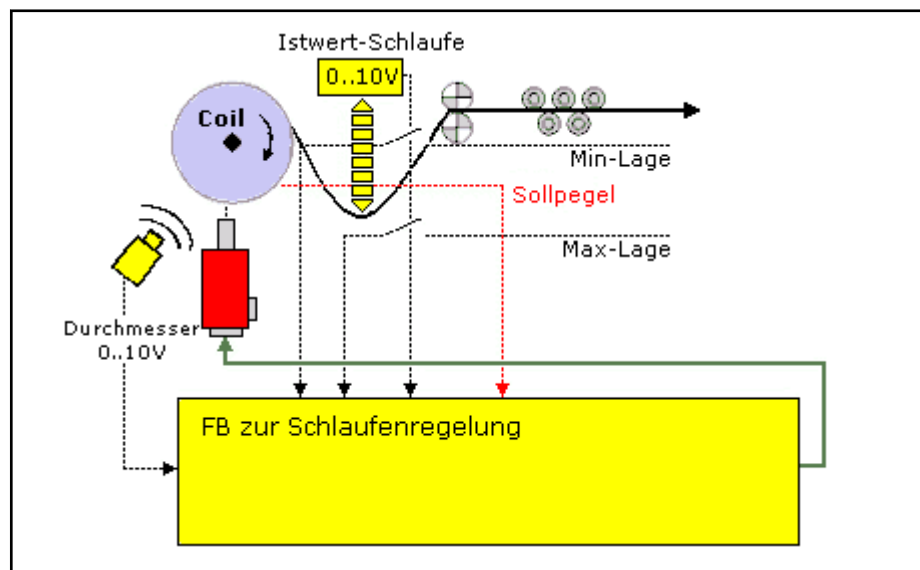


Abb.6-39: Anwendung 2, ML_SagControlA

6.5 Funktionsbausteine zum taktischen Ein- und Auskoppeln

6.5.1 Überblick

Die Bosch Rexroth IndraMotion MLC Steuerungen ab Firmware-Version 3 wurden durch Bereitstellung von CamLock-Technologie-Funktionsbausteinen um die "Lock On/Lock Off" Funktionalität erweitert.

Der **MB_PrepareCams** Funktionsbaustein verwendet ein Polynom 5. Ordnung um 3 Kurvenscheibenprofile zu berechnen, zu erzeugen und sie in eine Slave-Achse zu laden. Diese Kurvenscheibenprofile werden als Lock On, Lock Off und Eins-zu-Eins bezeichnet. Sie ermöglichen einer synchronisierten Slave-Achse sich vom Master auszukoppeln (Lock Off) und an einer vorher festgelegten Position zu stoppen, bis sie wieder mit dem Master synchronisiert wird (Lock On). Der Funktionsbaustein **MB_CamLock** wird verwendet um die Lock On/Lock Off Funktionalität zu aktivieren. Ein Anwender-Kurvenscheibenprofil wird ebenfalls unterstützt.

Solange eine Slave-Achse mit dem Master synchronisiert ist, folgt sie dem Master unter Nutzung der Eins-zu-Eins Kurvenscheibe. Ein Phasenoffset zwischen der Master- und der Slave-Position kann definiert werden. Wenn die Lock Off Kurvenscheibe aktiviert ist (über den MB_CamLock Funktionsbaustein), wechselt die Slave-Achse von der Master-Position zu einer vorher festgelegten Auskoppel-Position und kommt dann zum Stillstand. Ist die Lock Off Kurvenscheibe nicht aktiv (Lock On), synchronisiert sich die Slave-Achse von ihrer Stopp-Position aus wieder mit dem Master. Sobald sie sich aufsynchronisiert hat, schaltet die Slave-Achse wieder auf die Eins-zu-Eins Kurvenscheibe und folgt weiter der Position des Master-Eingangs.

6.5.2 CamLock – Applikationsbeispiel

Diese Funktionalität wird normalerweise in Verpackungsmaschinen verwendet, wobei vorausgesetzt wird, dass die ankommenden Produkte einen gleichmässigen Abstand aufweisen, bevor sie eingepackt werden können. Im Fall dass der Abstand zu groß ist, wird der Verpackungsprozess in der Maschine vom Master für einen oder mehrere Zyklen ausgekoppelt (Lock Off) bis das Produkt wieder erkannt wird. Diesen Zustand nennt man "No Product, No Seal". Sobald das Produkt erkannt ist, wird der Verpackungsprozess erneut auf den Master aufsynchronisiert (Lock On) und die Produkte werden weiter verpackt.

Diese nicht gleichmässig vorhandenen Abstände erfordern das Beschleunigen und Abbremsen der Slave-Achse um sie dadurch auf den Master zu synchronisieren. Die folgende Abbildung zeigt eine typische Verpackungsmaschine zum Füllen und Versiegeln:

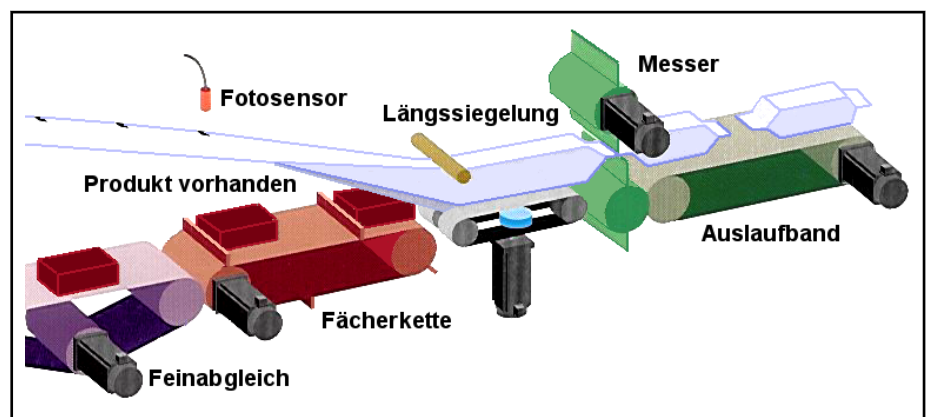


Abb.6-40: Horizontale Form einer Verpackungsmaschine zum Füllen und Versiegeln

Das folgende Diagramm zeigt den Bearbeitungs-, den Abbrems-, Stopp-, Beschleunigungs- und den erneuten Bearbeitungs-Prozess, dem eine Slave-Achse während des Lock On/Lock Off Prozesses folgt.

ML_TechMotion.library

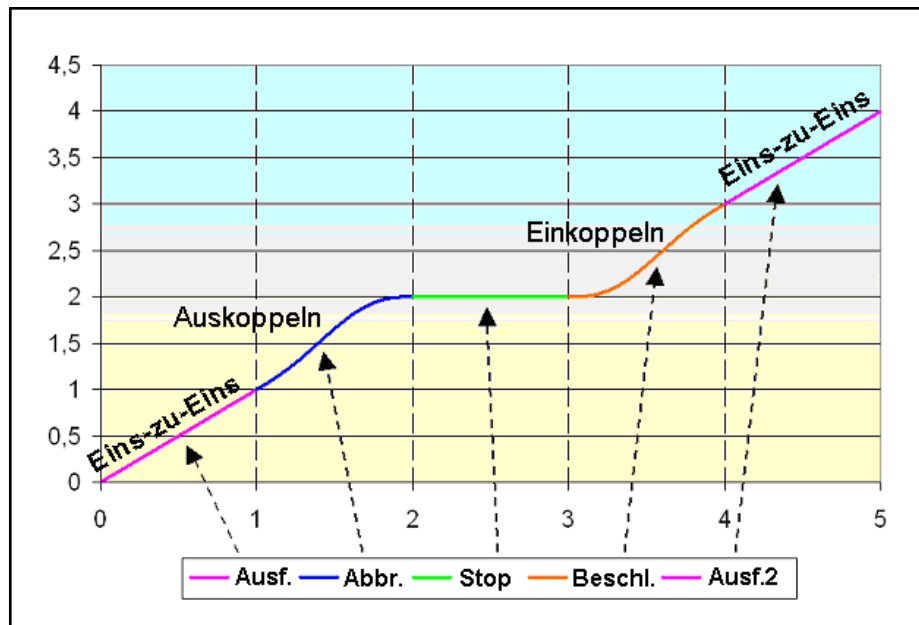


Abb.6-41: CamLock Kurvenscheibenprofil

Eins-zu-Eins Kurvenscheibenprofil "Synchronisiert auf Master"

Das Eins-zu-Eins Kurvenscheibenprofil ist aktiv und auf den Master-Eingang synchronisiert, wenn die Lock On/Lock Off Funktionalität nicht aktiv ist. Die Lock On/Lock Off Funktionalität wird aktiv, indem der Enable-Eingang des MB_CamLock Funktionsbausteins auf High gesetzt wird. Der LockOff-Eingang muss Low sein. Unter normalen Arbeitsbedingungen ist dieses Kurvenscheibenprofil aktiv und folgt den Vorgaben des Master-Eingangs.

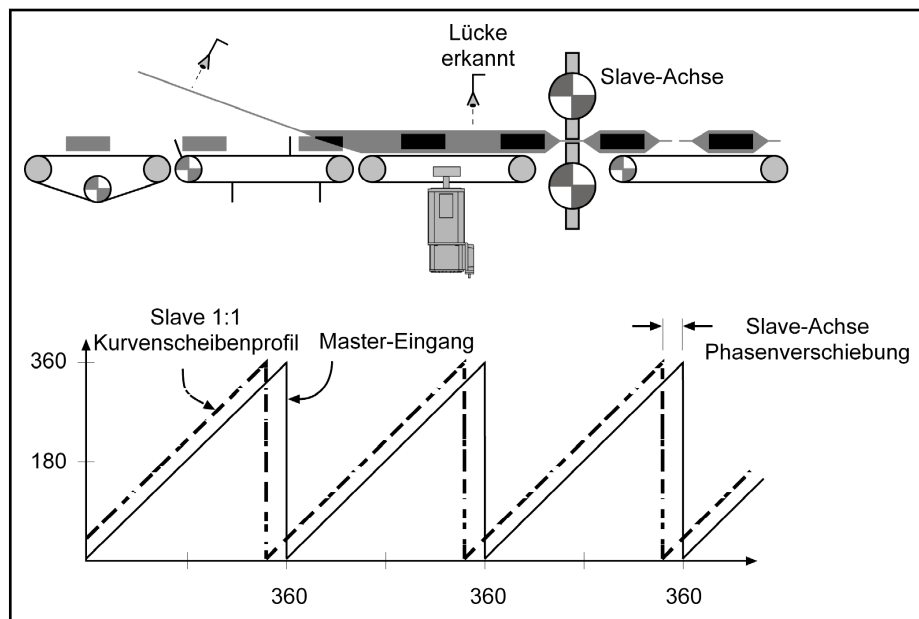


Abb.6-42: Bearbeitungs-Kurvenscheibe aktiv, Normalbetrieb der Verpackungs-Applikation

Lock Off Kurvenscheibenprofil

Das Lock Off Kurvenscheibenprofil bremst die Slave-Achse bis zum Stillstand über einen Zyklus des Masters. Die Lock Off Kurvenscheibe ist aktiv, solange die beiden Eingänge Enable und LockOff des MB_CamLock Funktionsbausteins auf High sind. Nach diesem Zyklus wird die Geschwindigkeit der Slave-Achse gestoppt und erst wieder neu gestartet, wenn das Lock On Kurvenscheibenprofil aktiv ist.



Einmal eingeschaltet, ist das Lock Off Kurvenscheibenprofil nach dem Nulldurchgang der Slave-Achse aktiv.

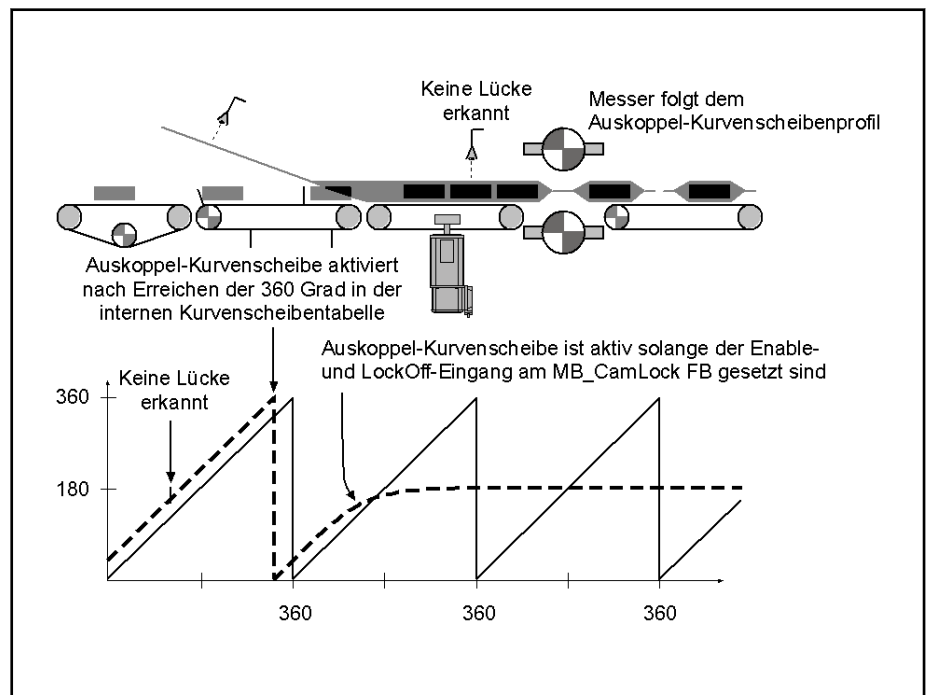


Abb.6-43: Lock Off Kurvenscheibe aktiv, No Product - No Seal

Lock On Kurvenscheibenprofil

Das Lock On Kurvenscheibenprofil beschleunigt von einer Halteposition bis die Geschwindigkeit des Master-Eingangs über einen Zyklus des Masters (360 Grad) erreicht ist. Nach diesem Zyklus stimmt die Geschwindigkeit von Slave-Achse und Master überein. Die Lock On Kurvenscheibe ist nur solange aktiv bis die Slave-Achse mit dem Master synchronisiert ist. Danach folgt die Slave-Achse der Eins-zu-Eins Kurvenscheibe während des Normalbetriebs. Das Lock On Kurvenscheibenprofil ist aktiv, solange am MB_CamLock Funktionsbaustein der Enable-Eingang High und der LockOff-Eingang Low ist.



Einmal eingeschaltet, ist das Lock On Kurvenscheibenprofil nach dem Nulldurchgang der Master-Achse aktiv.

ML_TechMotion.library

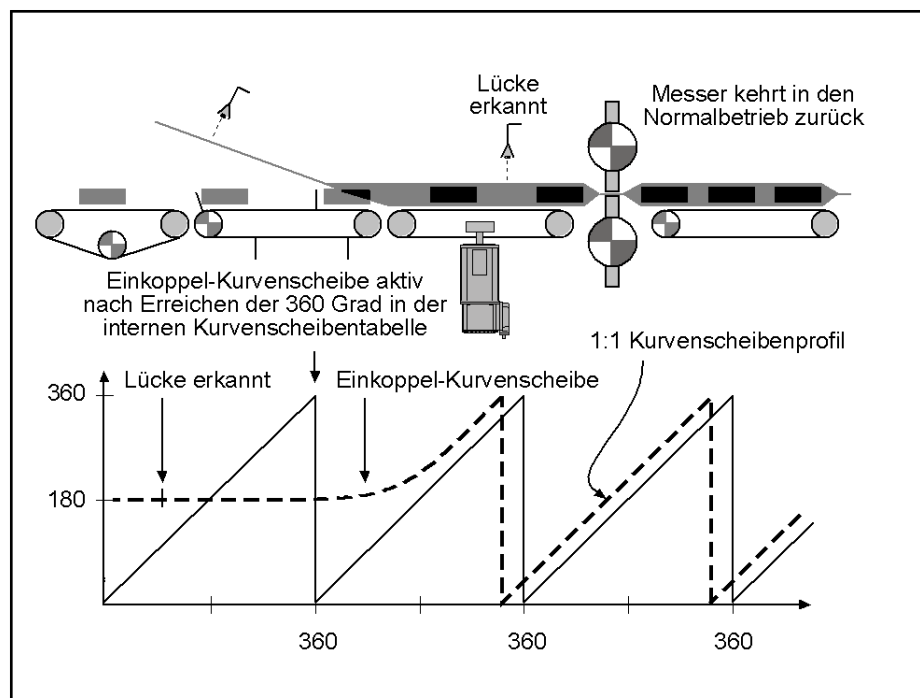


Abb.6-44: Lock On Kurvenscheibe aktiv, Produkt ist vorhanden

6.5.3 MB_PrepareCams

Kurzbeschreibung

Der Funktionsbaustein benutzt ein Polynom 5. Ordnung um 3 Kurvenscheibenprofile zu berechnen, zu erzeugen und sie in Parameter, die durch die MC_CAM_ID-Eingänge festgelegt sind, zu laden. Die erzeugten Bewegungsprofile enthalten Randbedingungen für die Position und Geschwindigkeit.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library

Abb.6-45: Referenztablette Funktionsbaustein MB_PrepareCams

Schnittstellenbeschreibung

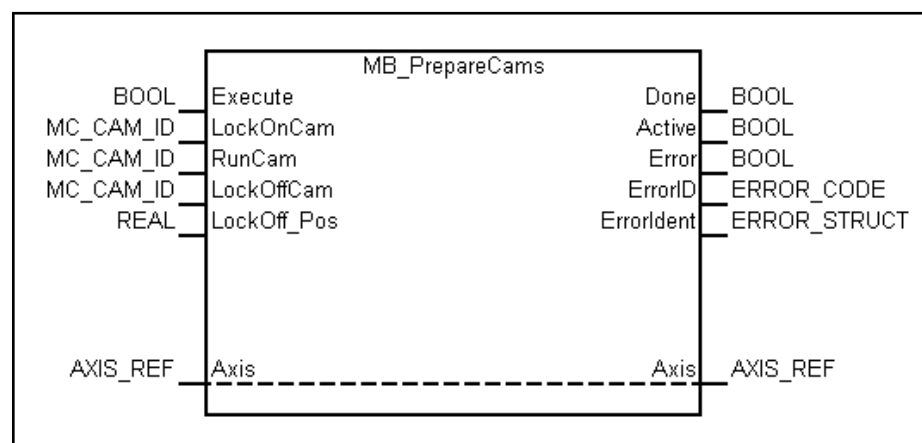


Abb.6-46: Funktionsbaustein MB_PrepareCams



Der MC_CAM_ID-Wert legt nur fest, welcher Parameter zum Speichern des berechneten Kurvenscheibenprofils benutzt wird. Die aktuelle Funktionalität (z.B. LockOnCam) wird durch den Eingang des Funktionsbausteins bestimmt.

Für ein IndraMotion MLC Steuerungssystem wird bei einem Wert von 1 das Kurvenscheibenprofil im Steuerungsparameter C-0-2001 abgespeichert. Um jedoch den MB_PrepareCams-Funktionsbaustein ohne Fehler ausführen zu können, ist nach wie vor ein Achs-Eingang für AXIS VAR_IN_OUT erforderlich. Die berechneten Kurvenscheibenprofile werden in Steuerungsparametern abgespeichert.

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Axis	AXIS_REF	Referenz zur Slave-Achse für die Kurvenscheibenprofile abgespeichert sind
VAR_INPUT	Execute	BOOL	Positive Flanke startet die Berechnung von drei Kurvenscheibenprofilen
	LockOnCam	MC_CAM_ID	Bestimmt die Zieladresse der Lock On-Kurvenscheibentabelle. Defaultwert ist 4
	RunCam	MC_CAM_ID	Bestimmt die Zieladresse der Bearbeitungs-Kurvenscheibentabelle. Defaultwert ist 3
	LockOffCam	MC_CAM_ID	Bestimmt die Zieladresse der Lock Off-Kurvenscheibentabelle. Defaultwert ist 2
	LockOff_Pos	REAL	Dieser Eingang wird zum Berechnen des entsprechenden Geschwindigkeitsprofils benutzt
VAR_OUTPUT	Done	BOOL	Drei Kurvenscheibenprofile wurden berechnet, erzeugt und geladen
	Active	BOOL	Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

Abb.6-47: Schnittstellenvariablen Funktionsbaustein MB_PrepareCams

IndraMotion MLC Steuerungssystem

Ein IndraMotion MLC Steuerungssystem unterstützt einen Block von 98 Steuerungsparametern zum Speichern von CamLock-Kurvenscheibenprofilen. Im Gegensatz zu den Antriebsparametern ist die Zuordnung zwischen MC_CAM_ID-Wert und Steuerungsparameter fortlaufend. Beginnend mit Steuerungsparameter C-0-2001 wird ein MC_CAM_ID-Wert von 1 in Steuerungsparameter C-0-2001, ein MC_CAM_ID-Wert von 98 in C-0-2098 gespeichert. Der Steuerungsparameter C-0-2099 kann nicht benutzt werden, um ein Kurvenscheibenprofil abzuspeichern; dieser Parameter ist reserviert für das "Stoppen der Slave-Achse".

Der Funktionsbaustein muss direkt nach dem Hochlauf der Steuerung gestartet werden, damit die drei Kurvenscheibenprofile berechnet und in die entsprechenden Parameter geladen werden können. Benutzen Sie den DONE-Ausgang, um zu überprüfen, dass dieser Vorgang beendet ist, bevor das SPS-Programm weiter läuft.

ML_TechMotion.library



Der MB_PrepareCams Funktionsbaustein wird nur einmal am Anfang des SPS-Programms und vor dem Start des MB_CamLock Funktionsbausteins ausgeführt.

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE" folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#0003	16#0000	FB wurde durch einen anderen FB beendet
RESOURCE_ERROR	16#0004	16#0000	Diese Antriebsfirmware-Version wird nicht unterstützt
INPUT_RANGE_ERROR	16#13A1	16#0001	Die kurvenscheiben-bezogenen Werte wurden nicht korrekt initialisiert
INPUT_RANGE_ERROR	16#13A1	16#0002	Slave Axis_Ref; die AxisNo liegt ausserhalb des zulässigen Bereichs
INPUT_RANGE_ERROR	16#13A1	16#0003	LockOff_Pos muss grösser als 0 und kleiner 360 sein, Default=180 Grad
CALCULATION_ERROR	16#13A2	16#0000	Berechnung der Schrittweite, Ergebnis = 0
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Zustand der State Machine

Abb.6-48: Fehlercodes Funktionsbaustein MB_PrepareCams

6.5.4 MB_CamLock

Kurzbeschreibung

Der MB_CamLock Funktionsbaustein wird verwendet um die durch den Funktionsbaustein MB_PrepareCams berechneten und gespeicherten Run-, LockOn- und LockOff-Kurvenscheibenprofile zu aktivieren. Zusätzlich zum Aktivieren von Kurvenscheibenprofilen bietet dieser Funktionsbaustein noch die folgenden Funktionalitäten:

- Elektronische Getriebeübersetzung
- Synchronisationsrichtung (SyncMode)
- MC_CamIn- und MC_CamOut-Funktionalität für Antriebs-Kurvenscheiben
- Master-Feinabgleich



Der Betrieb des Funktionsbausteines erfordert den zyklischen Aufruf in der Motion-Task. Bitte beachten Sie dazu die Parametrierung in [Kap. 6.5.5 "MB_CamLock Parametrierung" auf Seite 209](#)

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library

Abb.6-49: Referenztablelle

Schnittstellenbeschreibung

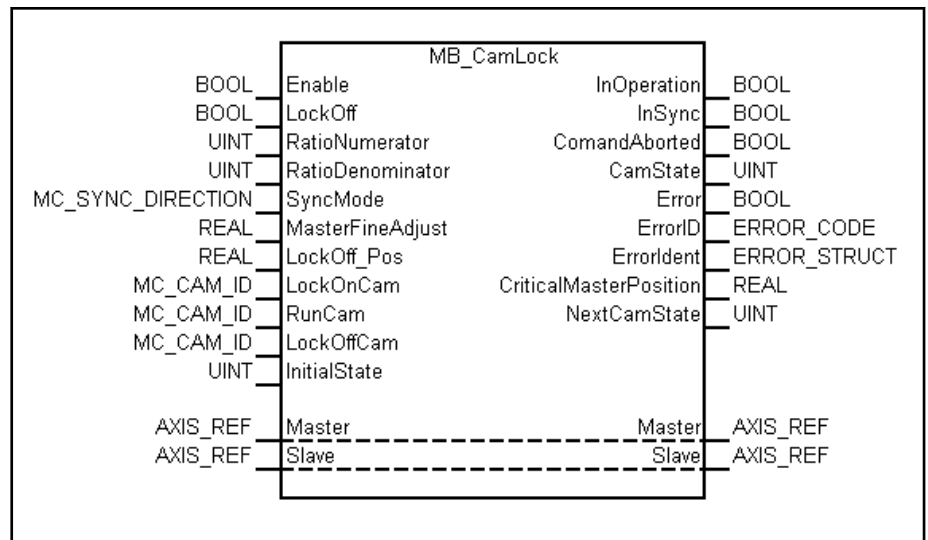


Abb.6-50: Funktionsbaustein MB_CamLock



Für das IndraMotion MLC Steuerungssystem können Steuerungskurvenscheiben verwendet werden, um dem Slave-Eingang eine virtuelle Achse zuzuweisen.

Die folgende Tabelle zeigt die verschiedenen Kurvenscheibenprofile die durch den MB_CamLock Funktionsbaustein gesteuert werden:

Kurvenscheibenprofil	Enable-Eingang	Lock Off-Eingang	Beschreibung
Run	High	Low	Normalbetrieb unter Nutzung eines 1:1 Kurvenscheibenprofils
LockOff	High	Low nach High	Die Slave-Achse schaltet von der Run-Kurvenscheibe zum LockOff-Kurvenscheibenprofil und kommt zum Stillstand an der durch den LockOff_Pos-Eingang festgelegten Position
LockOn	High	High nach Low	Die Slave-Achse beschleunigt von einer Halteposition unter Nutzung des LockOn-Kurvenscheibenprofils, synchronisiert sich mit dem Master und schaltet zur Run-Kurvenscheibe

Abb.6-51: Normaler CamLock-Betrieb

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Slave	AXIS_REF	Reale oder virtuelle Achse
	Master	AXIS_REF	Reale oder virtuelle Achse
VAR_INPUT	Enable	BOOL	Aktiviert den MB_CamLock FB. Durch eine steigende Flanke wird der Slave vor Eintritt in den Run-Zustand dynamisch aufsynchroisiert. Eine fallende Flanke bewirkt eine Getriebe- Auskopplung

ML_TechMotion.library

E/A-Typ	Name	Datentyp	Beschreibung
	LockOff	BOOL	TRUE: Locked off (ausgekoppelt) FALSE: Locked on (eingekoppelt) Ist dieser Eingang vor Setzen von Enable TRUE, dann synchronisiert sich die Achse dynamisch auf, geht in den Run-zustand über und koppelt sich dann aus (LockOff)
	RatioNumerator	UINT	Elektronische Getriebeübersetzung
	RatioDenominator	UINT	Elektronische Getriebeübersetzung
	SyncMode	MC_SYNC_DIRECTION	0: = kürzeste Wegstrecke 1: = positive Richtung 2: = negative Richtung
	MasterFineAdjust	REAL	Dieser Eingang wird für den MB_MotionProfile FB benötigt
	LockOff_Pos	REAL	Die Position in Grad an der die Slave-Achse stoppt wenn sie vom Master ausgekoppelt wird. Der Defaultwert ist 180 Grad
	LockOnCam	MC_CAM_ID	Bestimmt die Bezugsquelle der LockOn-Kurvenscheibentabelle. Defaultwert ist 4
	RunCam	MC_CAM_ID	Bestimmt die Bezugsquelle der Run-Kurvenscheibentabelle. Defaultwert ist 3
	LockOffCam	MC_CAM_ID	Bestimmt die Bezugsquelle der LockOff-Kurvenscheibentabelle. Defaultwert ist 2
	InitialState	UINT	Ausgangszustand der Zustandsmaschine
VAR_OUTPUT	InOperation	BOOL	Ist TRUE, solange der FB aktiviert ist
	InSync	BOOL	Der InSync-Ausgang ist TRUE, wenn sich die Run-Kurvenscheibe im synchronisierten Zustand befindet
	CommandAborted	BOOL	Kommando wurde durch einen anderen FB abgebrochen
	CamState	UINT	Aktueller Status des CamLock FB 1: in Dynamic Sync (dynamischer Synchronisation) 2: im Run (Bearbeitungszustand) 3: im LockOff (ausgekoppelten Zustand) 4: im Standstill (Stoppzustand) 5: im LockOn (eingekoppelten Zustand) 6: in Gear Out [Getriebe-Auskopplung (fortlaufende Bewegung)]
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

E/A-Typ	Name	Datentyp	Beschreibung
	CriticalMasterPosition	REAL	Stellt die notwendige kritische Master-Position für den Schaltzustandsübergang bereit
	NextCamState	UINT	Zeigt den nächsten zu verarbeitenden Ausgang an

Abb.6-52: Schnittstellenvariablen Funktionsbaustein MB_CamLock

Signal-Zeit-Diagramm

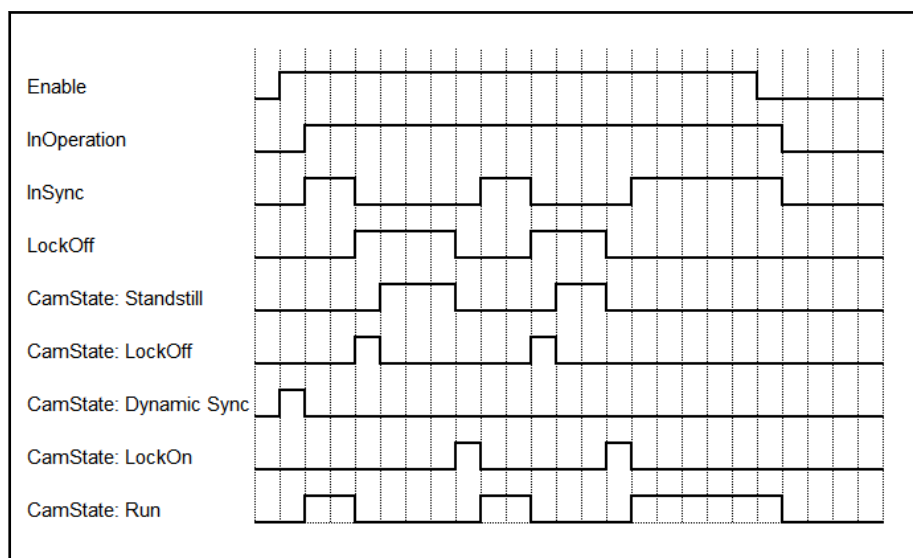


Abb.6-53: MB_CamLock Signal-Zeit-Diagramm

Funktionsbeschreibung

Ist der **Enable**-Eingang auf High und ist gleichzeitig der **LockOff**-Eingang auf Low gesetzt, synchronisiert sich die Slave-Achse dynamisch mit dem Master und schaltet sofort in den Run-Zustand. Der **CamState**-Ausgang wechselt von Dynamic Sync (1) in den Run-Zustand (2). Der Funktionsbaustein bleibt im Run-Zustand (bearbeitet eine 1:1 Kurvenscheibe) bis der **LockOff**-Eingang auf High geht.

Geht der **LockOff**-Eingang während der Bearbeitung auf High, führt der Funktionsbaustein eine LockOff-Kurvenscheibe aus und stoppt an der Position, die über den **LockOff_Pos**-Eingang vorgegeben ist. Während des Übergangs von Run zu Stop geht der Ausgang **CamState** von Run (2) über LockOff (3) zu Standstill (4). Der Defaultwert für die LockOff_Pos ist 180 Grad, welches die halbe Wegstrecke von LockOn zu Run und von Run zu LockOff ist.

Zu diesem Zeitpunkt bleibt der Funktionsbaustein im Stoppzustand (Standstill) bis LockOff auf Low geht. Dies beendet den Zyklus. Um den CamLock-Funktionsbaustein zu deaktivieren wird empfohlen, dass man sich im CamState=4 (Stoppzustand) befindet, und den Enable-Eingang auf Low setzt.

Ist der Eingang **Enable** während des Run-Zustands auf Low gesetzt, entsynchronisiert sich der Slave von der LockOn/LockOff-Kurvenscheibenfunktionalität und geht in die kontinuierliche Bewegung über.

ML_TechMotion.library

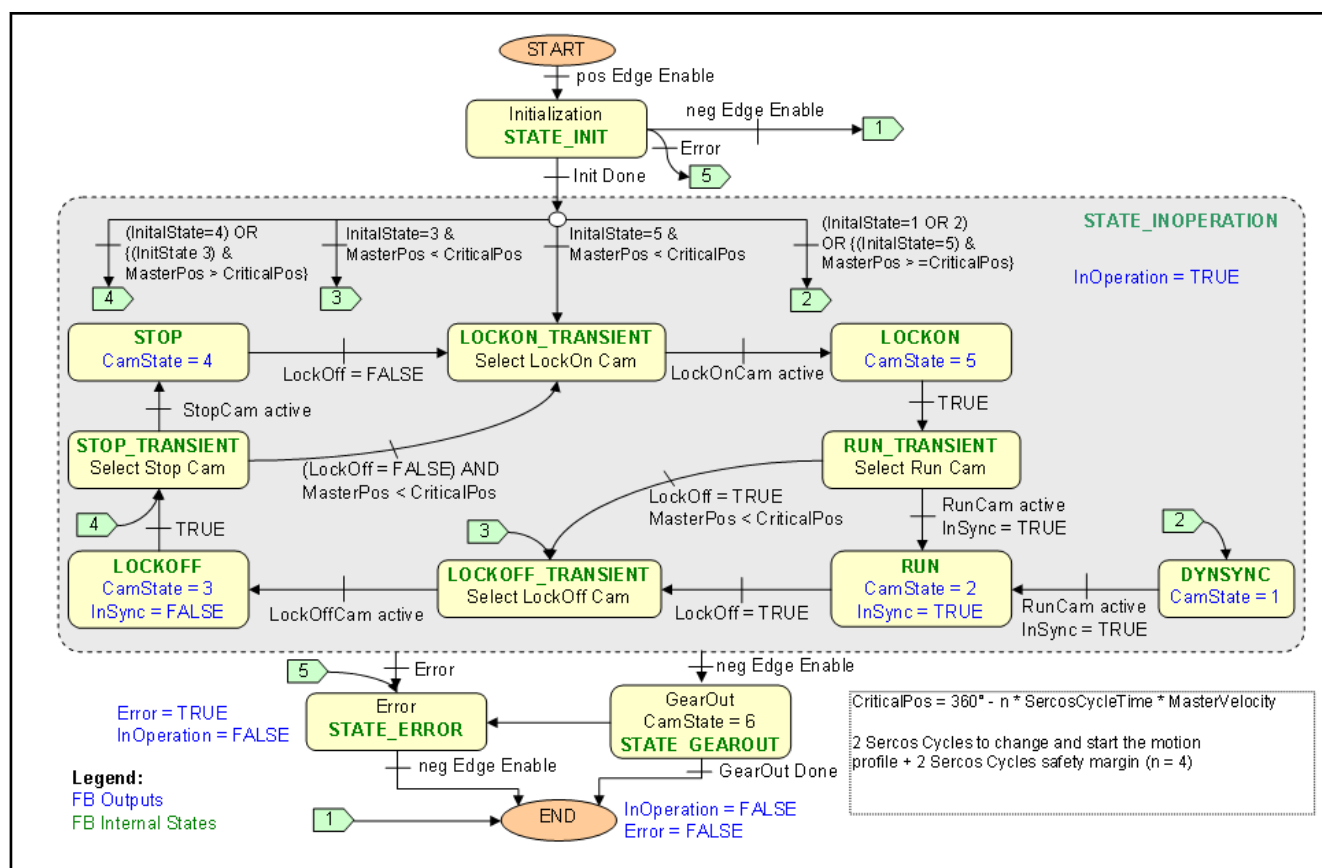
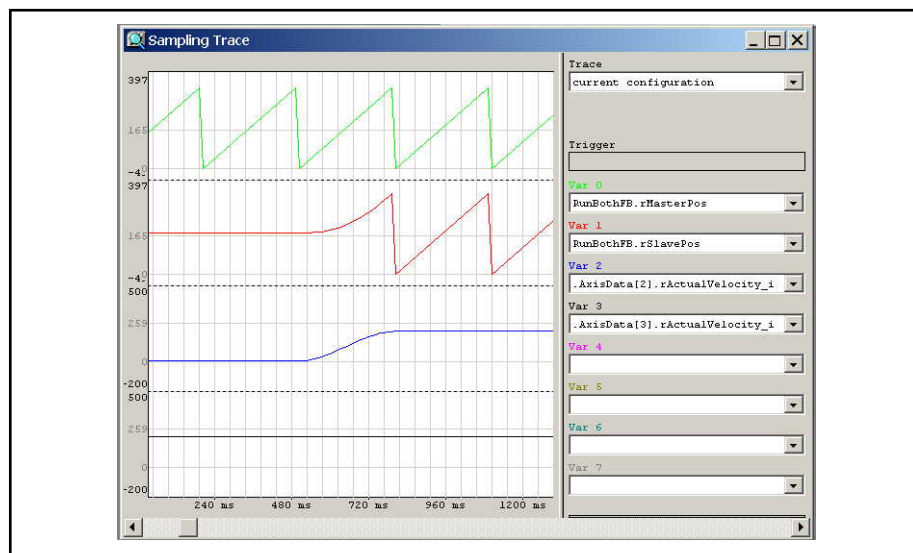


Abb. 6-54: CamLock Zustandsmaschine

LockOn/LockOff Ablaufbeispiele

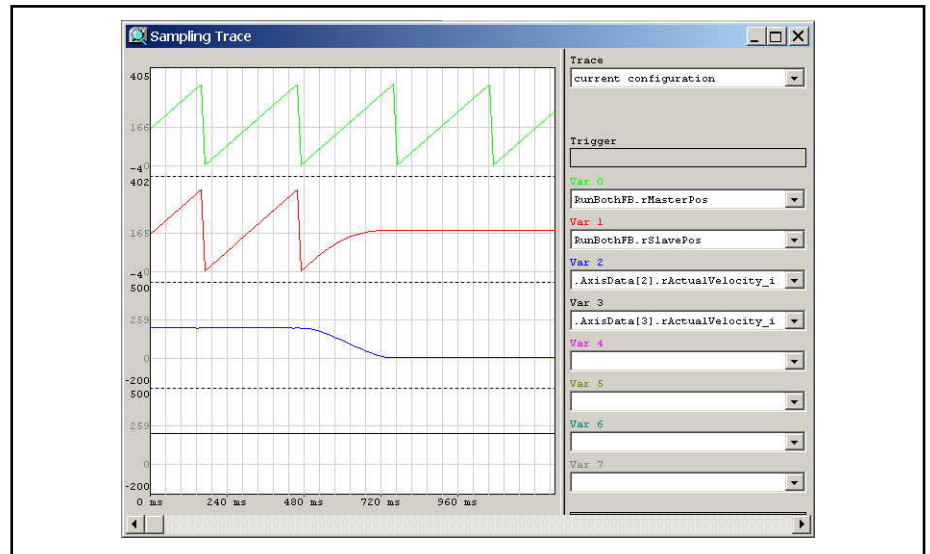


Kanal 0 (grün)= Master-Position
 Kanal 1 (rot) = Slave-Position
 Kanal 2 (blau)= Slave-Geschwindigkeit
 Master-Geschwindigkeit= 200 U/min

Abb. 6-55: LockOn-Profil mit einer LockOff_Pos von 180 Grad

Die obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während des LockOn-Profiles. Die LockOff_Pos beträgt 180 Grad (Default). Die obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während

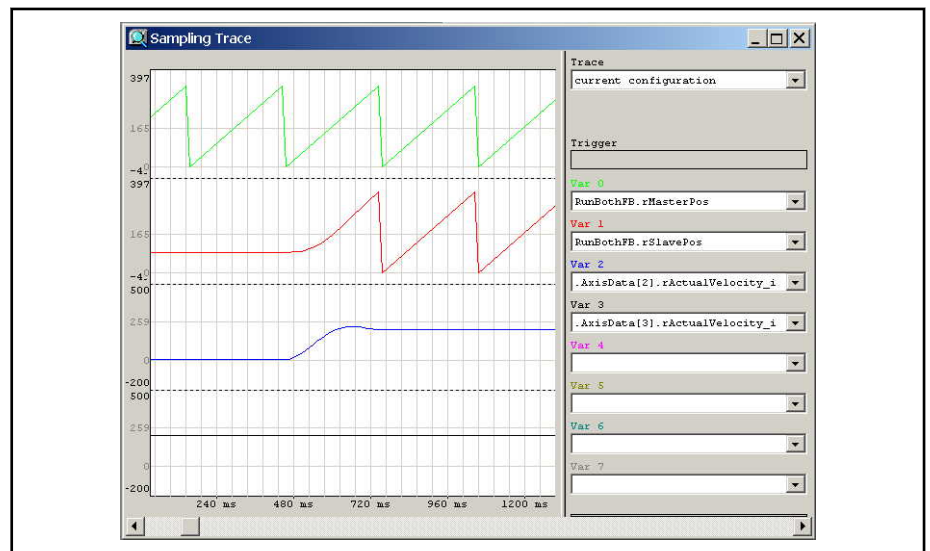
des LockOff-Profiles ohne Sprünge und Überschwüngen. Die LockOff-Position ist 180 Grad (Default).



Kanal 0 (grün)= Master-Position
Kanal 1 (rot) = Slave-Position
Kanal 2 (blau)= Slave-Geschwindigkeit
Master-Geschwindigkeit= 200 U/min

Abb.6-56: LockOff-Profil mit einer LockOff_Pos von 180 Grad

Die obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während des LockOff-Profiles. Die LockOff_Pos beträgt 180 Grad (Default). Die obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während des LockOff-Profiles ohne Sprünge und Überschwüngen. Die LockOff-Position ist 180 Grad (Default).



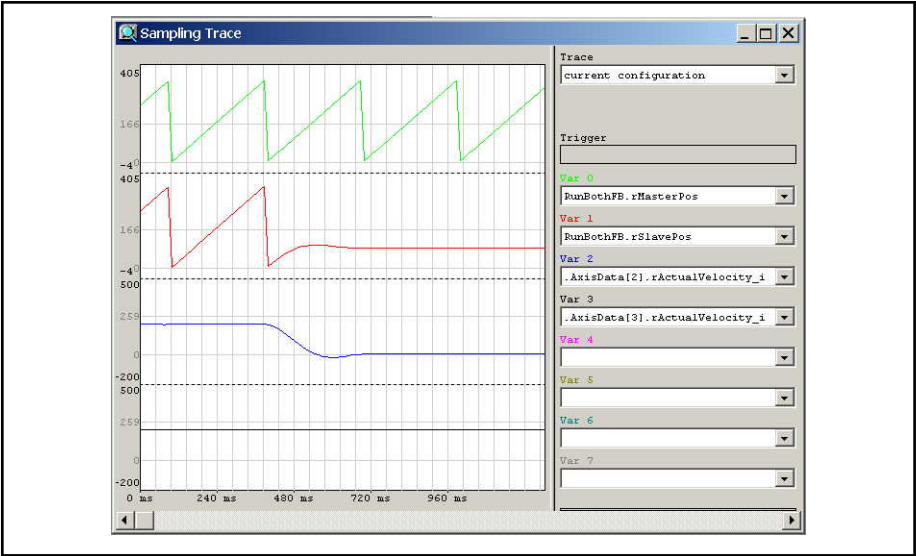
Kanal 0 (grün)= Master-Position
Kanal 1 (rot) = Slave-Position
Kanal 2 (blau)= Slave-Geschwindigkeit
Master-Geschwindigkeit= 200 U/min

Abb.6-57: LockOn-Profil mit einer LockOff_Pos von 90 Grad

Die obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während des LockOn-Profiles, wenn die LockOff_Pos 90 Grad ist. Zu sehen ist, dass bei einer LockOff-Position von 90 Grad die Positions- und Geschwindig-

ML_TechMotion.library

keits-Profile noch in Ordnung sind (es gibt eine Geschwindigkeits-Überschwingung, aber keine Rückwärtsbewegung).



Kanal 0 (grün)= Master-Position
Kanal 1 (rot) = Slave-Position
Kanal 2 (blau)= Slave-Geschwindigkeit
Master-Geschwindigkeit= 200 U/min

Abb.6-58: LockOff-Profil mit einer LockOff_Pos von 90 Grad

Die obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während des LockOff-Profils wenn die LockOff-Position 90 Grad ist. Zu sehen ist, dass bei einer LockOff-Position von 90 Grad die Positions- und Geschwindigkeits-Profile ein Überschwingen haben. Dieses Positionsüberschwingen verursacht eine Rückwärtsbewegung.



Es wird empfohlen für den LockOff_Pos-Eingang einen Wert von 180 Grad zu verwenden (der Wert dieses Eingangs muss sowohl für den MB_PrepareCams als auch für den MB_CamLock Funktionsbaustein übereinstimmen). Wird eine andere LockOff-Position als 180 Grad verwendet, kommt es, je weiter man von 180 Grad weg ist, zu einem Überschwingen in der Position und der Geschwindigkeit.

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle **F_RELATED_TABLE** folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#0001	16#0000	Spannung der Slave-Achse ist nicht auf AN
RESOURCE_ERROR	16#0003	16#0000	FB wurde durch einen anderen FB beendet
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Zustand der Zustandsmaschine
ACCESS_ERROR	16#0011	16#0000	Steuerung befindet sich nicht in der BB-Phase (SERCOS P4)

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR	16#1301	16#0001	LockOff_Pos muss größer als 0 und kleiner 360 sein, Default=180 Grad
INPUT_RANGE_ERROR	16#1301	16#0002	InitialState liegt außerhalb des Bereiches 1...5
INPUT_RANGE_ERROR	16#1301	16#0003	RatioNumerator <= 0
INPUT_RANGE_ERROR	16#1301	16#0004	RatioDenominator <= 0
INPUT_RANGE_ERROR	16#1301	16#0005	SyncMode < 0 ODER SyncMode > 2
INPUT_RANGE_ERROR	16#1301	16#0006	LockOnCam < 1 OR LockOnCam > 98
INPUT_RANGE_ERROR	16#1301	16#0007	RunCam < 1 ODER RunCam > 98
INPUT_RANGE_ERROR	16#1301	16#0008	LockOffCam < 1 ODER LockOffCam > 98
INPUT_RANGE_ERROR	16#1301	16#0009	UserCam_Profile < 1 ODER UserCam_Profile > 98
INPUT_RANGE_ERROR	16#1301	16#0010	Die CamTable-Eingänge sind gleich
ACCESS_ERROR	16#1302	16#0003	Der Achsenmodulowert des Slaves beträgt nicht 360°
ACCESS_ERROR	16#1302	16#0004	Der Mastermodulowert des Slaves beträgt nicht 360°

Abb.6-59: Fehlercodes Funktionsbaustein MB_CamLock

6.5.5 MB_CamLock Parametrierung

Der Funktionsbaustein CamLock benötigt die folgende Parametrierung:

1. Die Checkbox "Interpolation im Antrieb" muss bei Hinzufügen von einer realen Achse in IndraWorks abgewählt werden, wenn diese Achse als Slave Achse des MB_CamLock verwendet wird.
2. Leit- und Folgeachse sind ins Moduloformat 360° zu setzen.
3. Der Antrieb ist vor dem Gebrauch des FB zu referenzieren (absolute Position bevorzugt).
4. Der FB MB_PrepareLockCams muss einmal ausgeführt werden um die drei benötigten Kurvenscheiben zu berechnen und zu speichern.
 - LockOnCam, standardmäßig in CAM_TABLE_4 gespeichert
 - RunCam, standardmäßig in CAM_TABLE_3 gespeichert
 - LockOffCam, standardmäßig in CAM_TABLE_2 gespeichert
 - User_Cam_Profile, standardmäßig in CAM_TABLE_1 gespeichert
5. Die Leistung muss zum Antrieb zugeschaltet sein (Antrieb im Zustand "Enabled").
6. Der MB_CamLock enthält zeitkritische Motion -Funktionalität und sollte daher in der MotionTask aufgerufen werden.
7. Der FB arbeitet nur für eine Leitachse, die sich in positiver Richtung bewegt.
8. Anmerkung: CamID 99 wird immer als Stopp-Kurvenscheibe verwendet.

ML_TechMotion.library

6.6

Bausteine zur allgemeinen Verwendung

6.6.1

MB_DistanceAccumulator

Kurzbeschreibung

Der MB_DistanceAccumulator Funktionsbaustein summiert den relativ gefahrenen Weg mittels eines Gebers auf. Mit der steigenden Flanke des **Enable**-Eingangs wird der gefahrene Weg durch den FB aufsummiert bis der aufsummierte Weg größer als oder gleich des **TotalDistance**-Eingangswerts ist. Dabei wird der **DistanceReached**-Ausgang auf TRUE gesetzt und die Bearbeitung des Funktionsbausteines ist abgeschlossen. Die aktuelle Geberposition wird mittels des **CurrentPosition**-Eingangs vorgegeben.

Interface-Beschreibung

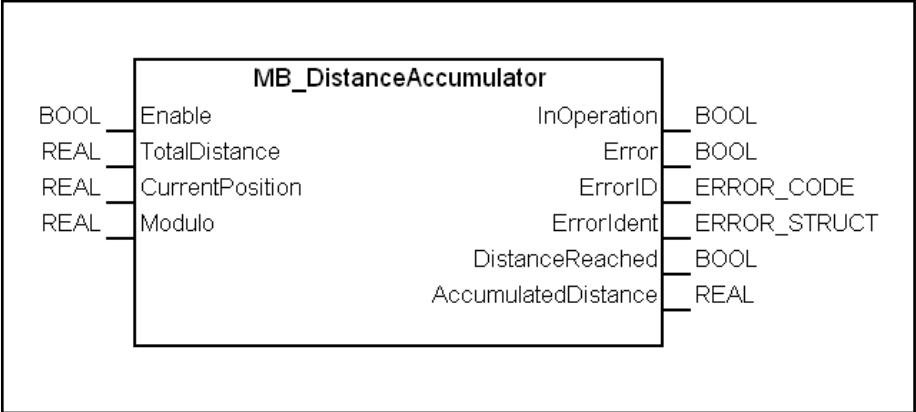


Abb.6-60: Funktionsbaustein MB_DistanceAccumulator

E/A-Typ	Name	Datentyp	Kommentar
VAR_INPUT	Enable	BOOL	Aktiviert Funktionsbaustein mit steigender Flanke
	TotalDistance	REAL	Zu fahrender Gesamtweg bevor Prozess abgeschlossen ist
	CurrentPosition	REAL	Aktuelle Geberposition
	Modulo	REAL	Modulo Wert: (360: rotatorisch 0: linear) Im Fall einer Servo-Achse kann dieser Wert aus A-0-0045 ausgelesen werden
VAR_OUTPUT	InOperation	BOOL	Der Funktionsbaustein ist zurzeit aktiv
	Error	BOOL	Prozess mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Kurze Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	DistanceReached	BOOL	TotalDistance wurde erreicht und die Bearbeitung des Funktionsbausteins ist abgeschlossen.
	AccumulatedDistance	REAL	Zeigt den aktuell gefahrenen Weg an

Abb.6-61: Schnittstellenvariablen Funktionsbaustein MB_DistanceAccumulator

Fehlerbehandlung

Der Funktionsbaustein generiert folgende Fehlermeldungen in Additional1/Additional2 für "F_RELATED_TABLE".

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#0003	16#0000	Der Funktionsbaustein wurde durch einen anderen Funktionsbaustein abgebrochen
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Status der Zustandmaschine

Abb.6-62: Fehlercodes Funktionsbaustein MB_DistanceAccumulator

6.6.2 MB_AxisDistanceAccumulator

Der MB_AxisDistanceAccumulator Funktionsbaustein summiert den relativ gefahrenen Weg mittels eines Gebers auf. Mit der steigenden Flanke des **Enable** Eingangs wird der gefahrene Weg durch den Geber aufsummiert, bis der aufsummierte Weg größer als oder gleich des **TotalDistance** Eingangswerts ist. Dabei wird der **DistanceReached** Ausgang auf TRUE gesetzt und der Funktionsbaustein hält an. Dieser Funktionsbaustein verwendet den Achsreferenzeingang (AXIS_REF) um die aktuelle Geberposition automatisch abzufragen.

Schnittstellenbeschreibung

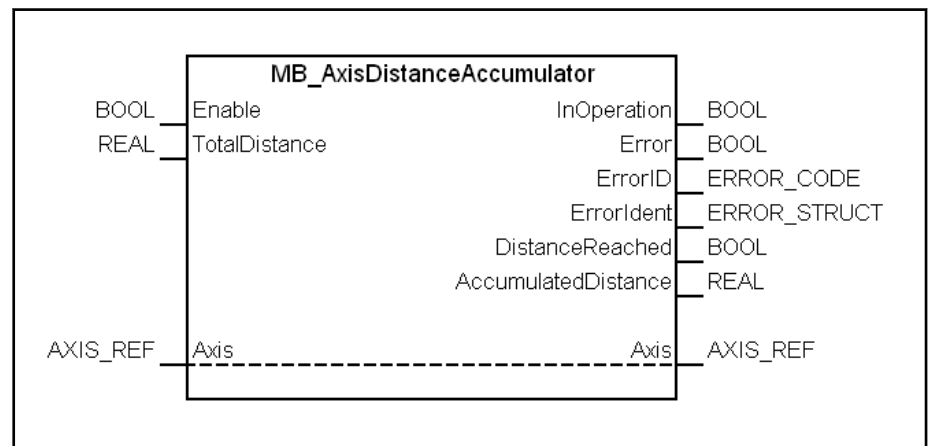


Abb.6-63: Funktionsbaustein MB_AxisDistanceAccumulator

E/A-Typ	Name	Datentyp	Kommentar
VAR_IN_OUT	Axis	AXIS_REF	Achsreferenz
VAR_INPUT	Enable	BOOL	Aktiviert Funktionsbaustein mit steigender Flanke
	TotalDistance	REAL	Zu fahrender Gesamtweg bevor Prozess abgeschlossen ist
VAR_OUTPUT	InOperation	BOOL	Der Funktionsbaustein ist zurzeit aktiv
	Error	BOOL	Prozess mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Kurze Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	DistanceReached	BOOL	TotalDistance wurde erreicht und die Bearbeitung des Funktionsbausteins ist abgeschlossen
	AccumulatedDistance	REAL	Zeigt den aktuell gefahrenen Weg an

Abb.6-64: Schnittstellenvariablen Funktionsbaustein MB_AxisDistanceAccumulator

ML_TechMotion.library

Fehlerbehandlung Der Funktionsbaustein generiert folgende Fehlermeldungen in Additional1/ Additional2 für "F_RELATED_TABLE":

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#0003	16#0000	Der Funktionsbaustein wurde durch einen anderen Funktionsbaustein abgebrochen
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Status der Zustandmaschine

Abb.6-65: Fehlercodes Funktionsbaustein MB_AxisDistanceAccumulator

6.6.3 MB_ReachVelocityType02

Kurzbeschreibung Der Funktionsbaustein berechnet den entsprechenden Ruck, die Beschleunigung und die Zielgeschwindigkeit zum Abfahren bestimmter ruckbegrenzter Geschwindigkeitsverläufe (z.B. S-Kurve) und lässt hierbei bis zu 3 Änderungen der Zielgeschwindigkeiten zu.

Es kann eine Ruckbegrenzung von 0-100% ausgewählt werden. Ist der Eingabewert 0, so ist der Verlauf der Geschwindigkeit linear, bei 100% S-kurvenförmig. Aus der Beschleunigungsdauer, der aktuellen Geschwindigkeit, der Zielgeschwindigkeit und dem JerkFactor werden die entsprechenden Sollwerte für Geschwindigkeit, Beschleunigung und Ruck berechnet. Die Beschleunigungsphase kann über StopAcc gestoppt werden.

Die Zielgeschwindigkeit kann während des Beschleunigungsvorgangs geändert werden. Neue Sollwerte für Geschwindigkeit, Beschleunigung und Ruck werden berechnet, um die neue Zielgeschwindigkeit zu erreichen. In der restlichen Zeit wirkt immer der neue Ruck, deswegen ist der angelegte JerkFactor nach einer Änderung der Zielgeschwindigkeit nicht mehr gültig. Die gesamte Beschleunigungsdauer ändert sich nur dann, wenn die neue Zielgeschwindigkeit sehr nahe an der aktuellen Geschwindigkeit liegt.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library

Abb.6-66: Referenztable

Schnittstellenbeschreibung

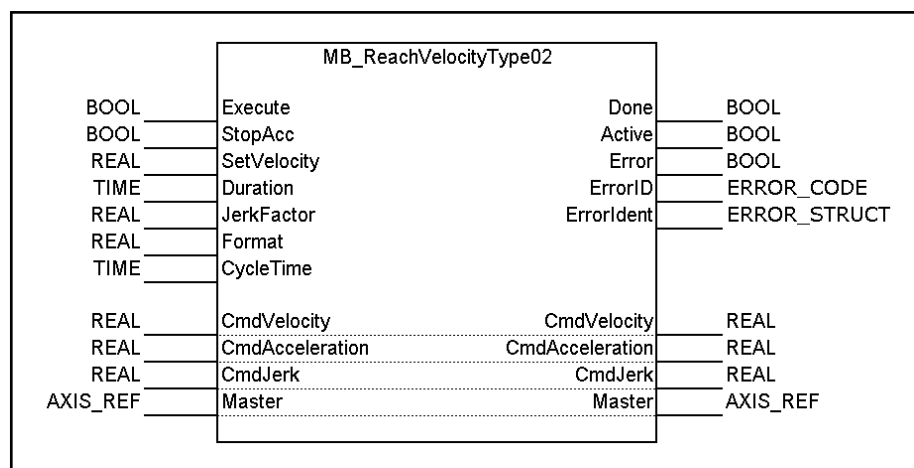


Abb.6-67: Funktionsbaustein MB_ReachVelocityType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	CmdVelocity	REAL	Sollgeschwindigkeit [U/min]
	CmdAcceleration	REAL	Sollbeschleunigung [rad/s²]

E/A-Typ	Name	Datentyp	Beschreibung
	CmdJerk	REAL	Sollruck [rad/s ³]
	Master	AXIS_REF	Referenz zur Leitachse Es wird empfohlen eine virtuelle Leitachse anzulegen
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins
	StopAcc	BOOL	Stoppt weitere Beschleunigung
	SetVelocity	REAL	Zielgeschwindigkeit [m/min]
	Duration	TIME	Beschleunigungszeit
	CycleTime	TIME	Zykluszeit der Task, in der der FB aufgerufen wird
	JerkFactor	REAL	Verrampung des Beschleunigungsprofils [%]
	Format	REAL	Format [mm]
VAR_OUTPUT	Done	BOOL	Bearbeitung abgeschlossen. Die Datenausgänge sind gültig
	Active	BOOL	Funktionsbaustein arbeitet und befindet sich in einem Zwischenzustand
	Error	BOOL	Signalisiert einen Fehler
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

Abb.6-68: Schnittstellenbeschreibung MB_ReachVelocityType02

Min-/Max- und Defaultwerte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
StopAcc	BOOL			FALSE	kontinuierlich
SetVelocity	REAL	n.def.	n.def.	0	kontinuierlich zwischen steigender Flanke an Execute und Done = TRUE
Duration	TIME	> 0 s	n.def.	30 s	steigende Flanke an Execute
JerkFactor	REAL	0	100	50	steigende Flanke an Execute
Format	REAL	> 0	n.def.	1000	steigende Flanke an Execute
CycleTime	TIME	> 1 ms	n.def.	10 ms	steigende Flanke an Execute

Abb.6-69: Eingangsverhalten MB_ReachVelocityType02

Funktionsbeschreibung

Die Beschleunigungsphase startet mit einer steigenden Flanke an Execute. Beschleunigungsdauer (Duration), die anzufahrende Bahngeschwindigkeit (SetVelocity) sowie das Beschleunigungsprofil (JerkFactor) werden übernommen. Eine Besonderheit ist, dass Active bei steigender Flanke an Execute gesetzt wird, die Beschleunigungsphase aber erst nach einer Sekunde be-

ginnt. D.h. nachfolgende Funktionsbausteine (beispielsweise der Registerregelung) können Active verwenden, um sich den Beginn einer Beschleunigungsphase anzeigen zu lassen. Nachdem die Zielgeschwindigkeit erreicht wurde, steht Active ebenfalls eine weitere Sekunde an, bis schließlich die Bearbeitung mit Done abgeschlossen wird. Die Beschleunigungsdauer (Duration), die anzufahrende Bahngeschwindigkeit (SetVelocity), sowie das Beschleunigungsprofil (JerkFactor) werden für eine virtuelle Leitachse vorgegeben, da die zu referenzierende Achse am Master (VAR_IN_OUT) eine virtuelle Achse sein muss, um die gewünschte Genauigkeit gewährleisten zu können. Um den Kurvenverlauf an einer realen Achse abzubilden, muss diese als Folgeachse synchron zur virtuellen Leitachse definiert werden.

S-Kurve Der Ruck ist die zeitliche Ableitung der Beschleunigung. Der JerkFactor gibt den prozentualen zeitlichen Anteil der Beschleunigungsphase an, in der der Ruck ungleich Null ist.

In nachfolgender Abbildung werden 3 Beschleunigungsphasen gezeigt:

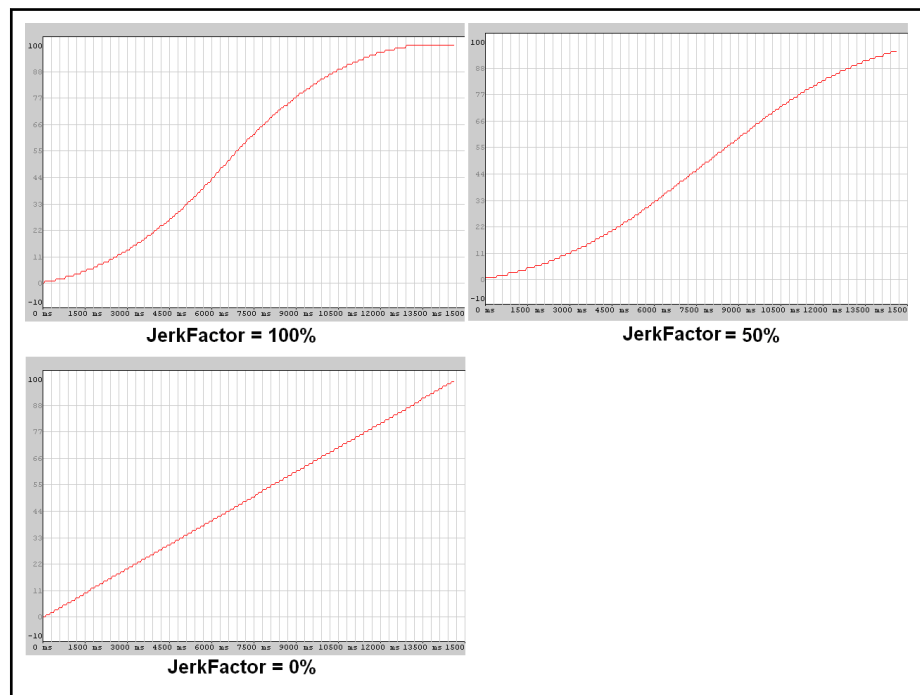


Abb.6-70: Beschleunigungsphasen mit unterschiedlichem JerkFactor

Änderung an SetVelocity

Nach der Zielgeschwindigkeitsänderung, vor dem Erreichen der Endgeschwindigkeit, werden CmdVelocity, CmdAcceleration und CmdJerk neu berechnet, indem die neue Zielgeschwindigkeit (SetVelocity) übernommen wird und die aktuelle Zeit, Geschwindigkeit und Beschleunigung intern gemessen werden. Der Funktionsbaustein erlaubt eine dreimalige Änderung der Zielgeschwindigkeit.

Abbildung Abb. 6-71 "Auswirkung der Änderung der Zielgeschwindigkeit" auf Seite 215 veranschaulicht die Leitachsdrehzahl in U/min einer Beschleunigungsphase von 0 auf 200 m/min (SetVelocity) in 30 Sekunden (Duration) und einem Format von 1 m mit JerkFactor 30%. Ist die Geschwindigkeit größer als 50 m/min, wird SetVelocity auf 300 m/min verändert. Nach der Änderung wird der neue Ruck berechnet und die verbleibende Beschleunigungsphase verläuft S-kurvenförmig. Der ursprüngliche JerkFactor wird nach der ersten Änderung nicht mehr berücksichtigt.

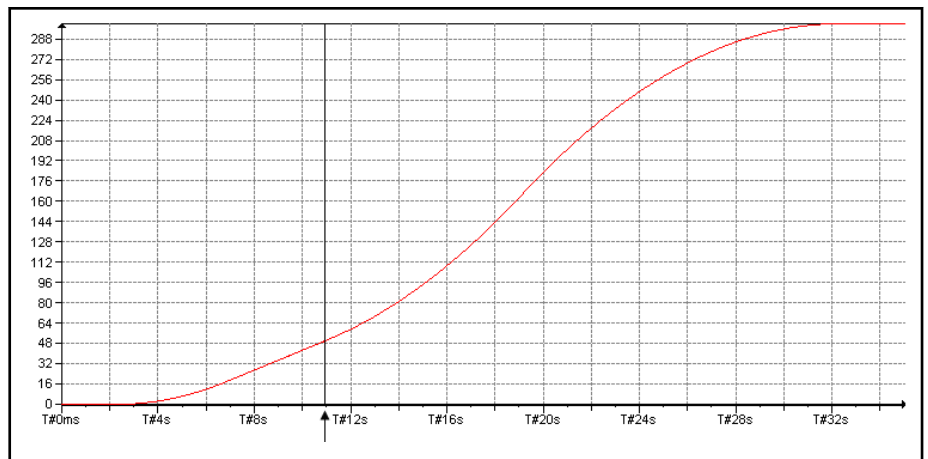


Abb.6-71: Auswirkung der Änderung der Zielgeschwindigkeit

Stoppen der Beschleunigung

Die mit Execute angestoßene Beschleunigungsphase kann mit StopAcc sofort angehalten werden. Der Funktionsbaustein beendet augenblicklich die weitere Beschleunigung. Der Ausgang Done wird gesetzt. Soll der gesetzte Geschwindigkeitswert SetVelocity dennoch angefahren werden, muss StopAcc deaktiviert und eine weitere steigende Flanke von Execute gesetzt werden. Das Erreichen des SetVelocity-Wertes wird bis zu einer fallenden Flanke an Execute durch den Ausgang Done signalisiert.

In nachfolgender Abbildung wird der Abbruch der linearen und der S-kurvenförmigen Beschleunigungsphase über StopAcc gezeigt:

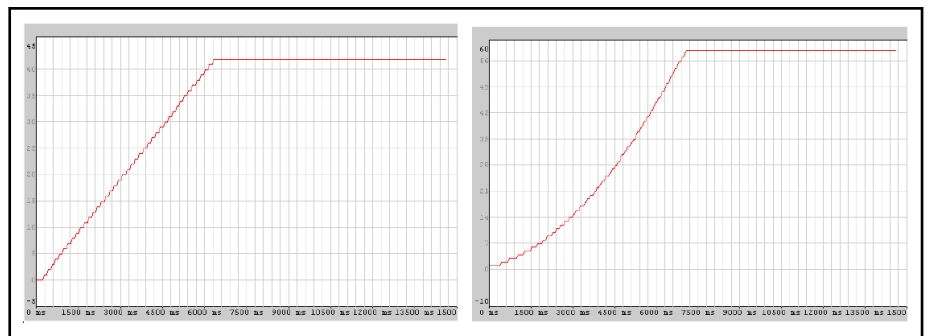


Abb.6-72: Abbruch der Beschleunigungsphase (links Abbruch bei 6300 ms, rechts Abbruch bei 7000 ms)

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle F_RELATED_TABLE, 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000000	Ungültiges Druckzylinderformat
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000023	Ungültige Zykluszeit
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000024	Ungültige Zeitdauer
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000025	Ungültiger JerkFactor
INPUT_INVALID_ERROR (16#0001)	16#0000000D	16#00000000	AxisNo von "Master" ist außerhalb des gültigen Bereichs

Abb.6-73: Fehlercodes von MB_ReachVelocityType02

ML_TechMotion.library

6.7 Funktionsbausteine zur Anwendung "Positionsüberwachung"

6.7.1 Einführung und Übersicht

Die PositionMonitoring-Bausteine dienen der Positionsüberwachung zweier Achsen.

Die Bausteine sind in mehreren Ausführungen verfügbar:

1. MB_PositionMonitoring_Gantrycant zur Überwachung der relativen Position zweier Achsen bezüglich Verkantung zueinander.
2. MB_PositionMonitoring_PosCollision zur Überwachung der absoluten und relativen Position zweier Achsen bezüglich Auszahnung und Kollision.

6.7.2 MB_PositionMonitoring_GantryCant

Kurzbeschreibung

Der Funktionsbaustein "MB_PositionMonitoring_GantryCant" überwacht die relative Position zweier Achsen bezüglich Verkantung zueinander. Mittels der Toleranzband-Funktionalität ist die Rückzugsbewegung möglich, wobei die Bewegungsrichtung aus der Verkantung weiter überwacht wird.

Der Funktionsbaustein eignet sich beispielsweise zur Überwachung von:

- Zustellachsen von Flexo-Druckwerken
- Gantry-Achsen mit mechanisch parallelen Verfahrenswegen u.a.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library

Abb.6-74: Referenztabelle Funktionsbaustein MB_PositionMonitoring_GantryCant

Schnittstellenbeschreibung

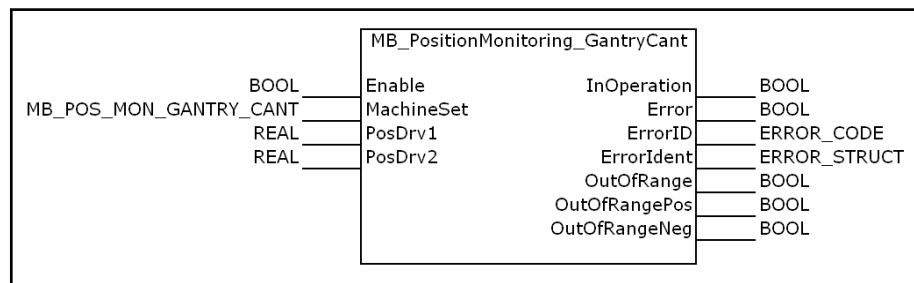


Abb.6-75: Funktionsbaustein MB_PositionMonitoring_GantryCant

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (zyklisch, zustandsgesteuert)
MachineSet	MB_POS_MON_GANTRY_CANT	maschinenspezifische Kenndaten	
PosDrv1	REAL	aktuelle Position der Achse 1	
PosDrv2	REAL	aktuelle Position der Achse 2	
VAR_OUTPUT	InOperation	BOOL	Überwachung der Achsen ohne Fehler abgeschlossen, Ausgangssignale gültig

E/A-Typ	Name	Datentyp	Beschreibung
Error	BOOL	Überwachung der Achsen mit Fehler abgeschlossen, Ausgangsvariable "ErrorIdent" gültig	
ErrorID	ERROR_ CODE	Beschreibung der Diagnose im Fehlerfall	
ErrorIdent	ERROR_ STRUCT	Beschreibung der Diagnose im Fehlerfall siehe "Fehlerbehandlung" auf Seite 218	
OutOfRange	BOOL	Überschreitung des zulässigen Bereiches	
OutOfRangePos	BOOL	Überschreitung des zulässigen Bereiches, Achsposition der Achse 1 größer bzw. gleich der Achsposition der Achse 2	
OutOfRangeNeg	BOOL	Überschreitung des zulässigen Bereiches, Achsposition der Achse 1 kleiner der Achsposition der Achse 2	

Abb.6-76: Schnittstellenvariablen Funktionsbaustein MB_PositionMonitoring_GantryCant

Datentypen Der Datentyp MB_POS_MON_GANTRY_CANT bildet den Maschinensatz des Funktionsbausteins "MB_PositionMonitoring_GantryCant" ab. Die Struktur enthält folgende Daten:

Strukturelement	Typ	Min-Wert	Max-Wert	Default-Wert	Beschreibung
Range	REAL	$\geq 2 \cdot \text{Tolerance}$	n.def.	10.0	maximal zulässige Positionsdivergenz beider Achsen
Tolerance	REAL	0.0	n.def.	5.0	Toleranz der Grenzen

Abb.6-77: Datenstruktur MB_POS_MON_GANTRY_CANT

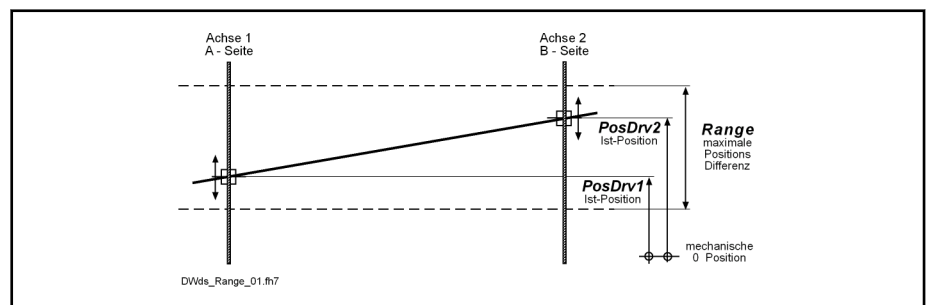


Abb.6-78: Druckwerk Draufsicht – maximale Positionsdivergenzbildung (Range)

Die angegebene Toleranz muss nachfolgender Bedingung genügen:

ML_TechMotion.library

$$\text{Range} > 2 * \text{Tolerance}$$

Abb.6-79: Bedingung Maschinensatz Type 01

Funktionsbeschreibung

Der Funktionsbaustein "MB_PositionMonitoring_GantryCant" überwacht nach Bearbeitungsfreigabe mittels "Enable" zyklisch die Achspositionen zweier Achsen bezüglich Verkantung zueinander. Überschreitet der Abstand der Achsen zueinander den maximal zulässigen Bereich "Range", wird dies über die Ausgänge "OutOfRange" und "OutOfRangePos" bzw. "OutOfRangeNeg" signalisiert.

Im Folgezyklus wird der Ausgang "OutOfRange" zurück gesetzt, die Ausgänge "OutOfRangePos" bzw. "OutOfRangeNeg" signalisieren weiterhin bis zum Erreichen des zulässigen Abstandes abzüglich Toleranz, dass eine Positionskorrektur notwendig ist.

Wird innerhalb des Zeitraumes dieser Positionskorrektur der maximal erreichte Abstand zuzüglich Toleranz erneut bzw. weiter überschritten, wird dies durch erneutes Setzen des Ausgangs "OutOfRange" signalisiert.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, wird dieser durch den Ausgang "Error" signalisiert sowie durch die, in diesem Fall aktualisierten Elemente der Ausgangsstruktur "ErrorIdent" spezifiziert. Die weiteren Ausgänge werden im Fehlerfall nicht aktualisiert.

Fehlercodes

Der Funktionsbaustein "MB_PositionMonitoring_GantryCant" generiert im Fehlerfall eine umfangreiche Diagnose, die lokal interpretiert werden kann.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170. Er kann in Additional1 und Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Eingänge liegen außerhalb des zulässigen Bereichs
STATE_MACHINE_ERROR (16#0005)	16#0006	16#????	ungültiger Zustand des Funktionsbausteins

Abb.6-80: Fehlercodes Funktionsbaustein MB_PositionMonitoring_GantryCant

6.7.3 MB_PositionMonitoring PosCollision

Kurzbeschreibung

Der Funktionsbaustein "MB_PositionMonitoring_PosCollision" überwacht die absolute und relative Position zweier Achsen bezüglich Auszahnung und Kollision.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library

Abb.6-81: Referenztable Funktionsbaustein MB_PositionMonitoring_PosCollision

Schnittstellenbeschreibung

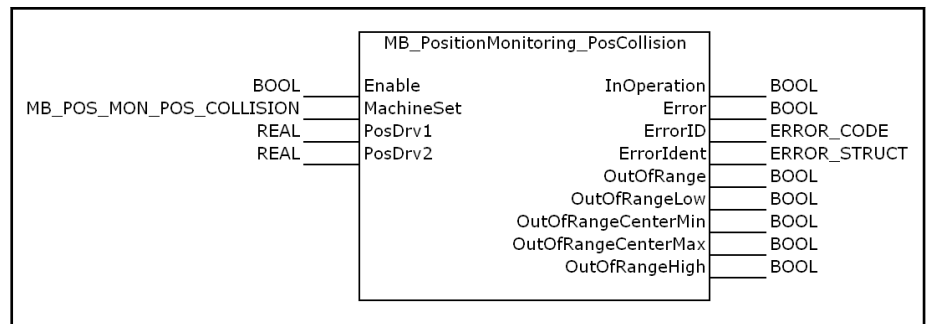


Abb.6-82: Funktionsbaustein MB_PositionMonitoring_PosCollision

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (zyklisch, zustandsgesteuert)
	MachineSet	MB_POS_MON_POS_COLLISION	maschinenspezifische Kenndaten
	PosDrv1	REAL	Aktuelle Position der Achse 1
	PosDrv2	REAL	Aktuelle Position der Achse 2
VAR_OUTPUT	InOperation	BOOL	Überwachung der Achsen ohne Fehler abgeschlossen, Ausgangssignale gültig
	Error	BOOL	Überwachung der Achsen mit Fehler abgeschlossen, Ausgangsvariable "ErrorIdent" gültig
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Beschreibung der Diagnose im Fehlerfall siehe "Fehlercodes" auf Seite 221
	OutOfRange	BOOL	Überschreitung des zulässigen Bereiches
	OutOfRangeLow	BOOL	Überschreitung des zulässigen Bereiches ("DeltaLow")
	OutOfRangeCenterMin	BOOL	Überschreitung des zulässigen Bereiches ("DeltaCenterMin")
	OutOfRangeCenterMax	BOOL	Überschreitung des zulässigen Bereiches ("DeltaCenterMax")
	OutOfRangeHigh	BOOL	Überschreitung des zulässigen Bereiches ("DeltaHigh")

Abb.6-83: Schnittstellenvariablen Funktionsbaustein MB_PositionMonitoring_PosCollision

Datentypen Der Datentyp MB_POS_MON_POS_COLLISION bildet den Maschinensatz des Funktionsbausteins "MB_PositionMonitoring_PosCollision" ab. Die Struktur enthält folgende Daten:

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Beschreibung
Position1	REAL	n.def.	n.def.	50.0	Sollposition der Achse 1
Diameter1	REAL	0.0	n.def.	20.0	Durchmesser der Achse 1
Diameter2	REAL	0.0	n.def.	20.0	Durchmesser der Achse 2

ML_TechMotion.library

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Beschreibung
DeltaLow	REAL	0.0	n.def.	5.0	Begrenzung des Positions- offsets der Achse 2
DeltaCenter- Min	REAL	0.0	$\leq \text{DeltaCenter-Max} - 2 * \text{Tolerance}$	2.0	Minimale Distanz der Ach- sen 1 und 2 zueinander
DeltaCenter- Max	REAL	$\geq 2 * \text{Tolerance}$	n.def.	5.0	Maximale Distanz der Ach- sen 1 und 2 zueinander
DeltaHigh	REAL	$\geq 2 * \text{Tolerance}$	n.def.	5.0	Begrenzung Achsposition 2
Tolerance	REAL	0.0	n.def.	1.0	Toleranz der Grenzen

Abb.6-84: Datenstruktur MB_POS_MON_POS_COLLISION

Die angegebene Toleranz muss nachfolgender Bedingung genügen:

$$\text{Kenngrößen} > 2 * \text{Tolerance}$$

Abb.6-85: Bedingung Maschinensatz PosCollision

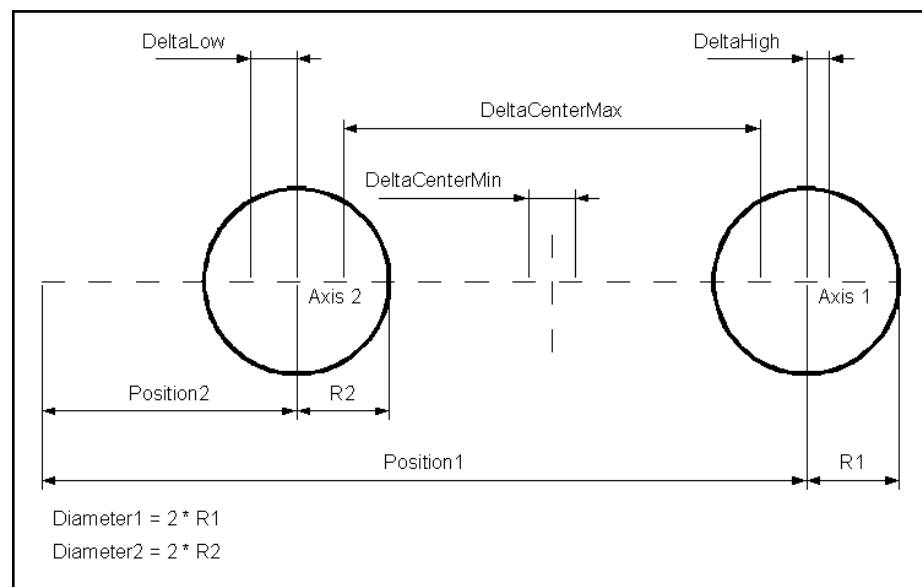


Abb.6-86: Maschinensatz des Funktionsbausteins MB_PositionMonitoring_Pos-Collision



Für den Funktionsbaustein "MB_PositionMonitoring_PosCollision" gelten nachfolgende Hinweise:

- Die Sollposition der Achse 2 ergibt sich aus den Kenngrößen des Maschinensatzes 2
- Die aktuelle Position der Achse 1 muss stets größer als die Summe der aktuellen Position der Achse 2 und der Radien beider Achsen sein
- Der Funktionsbaustein kann für drei oder mehr Achsen kaskadiert werden

Funktionsbeschreibung

Der Funktionsbaustein "MB_PositionMonitoring_PosCollision" überwacht nach Bearbeitungsfreigabe mittels "Enable" zyklisch die Achspositionen zwei-

er Achsen bezüglich Auszahnung und Kollision. Überschreitet der Abstand der Achsen zueinander einen maximal zulässigen Bereich, wird dies über die Ausgänge "OutOfRange" und die entsprechenden Bereiche "OutOfRangeLow", "OutOfRangeCenterMin", "OutOfRangeCenterMax" und/oder "OutOfRangeHigh" signalisiert.

Im Folgezyklus wird der Ausgang "OutOfRange" zurück gesetzt, die weiteren Ausgänge signalisieren weiterhin bis zum Erreichen des jeweilig zulässigen Abstandes abzüglich Toleranz, dass eine Positions Korrektur notwendig ist.

Wird innerhalb des Zeitraumes dieser Positions Korrektur der maximal erreichte Abstand zuzüglich Toleranz erneut bzw. weiter überschritten, wird dies durch erneutes Setzen des Ausgangs "OutOfRange" signalisiert.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, wird dieser durch den Ausgang "Error" signalisiert sowie durch die, in diesem Fall aktualisierten Elemente der Ausgangsstruktur "ErrorIdent" spezifiziert. Die weiteren Ausgänge werden im Fehlerfall nicht aktualisiert.

Fehlercodes Der Funktionsbaustein MB_PositionMonitoring_PosCollision generiert im Fehlerfall eine umfangreiche Diagnose, welche lokal interpretiert werden kann.

Der Funktionsbaustein nutzt die Fehlertabelle "F-RELATED-TABLE", 16#0170.

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Eingänge liegen außerhalb des zulässigen Bereichs
STATE_MACHINE_ERROR	16#0006	16#????	Eingänge liegen außerhalb des zulässigen Bereichs
CALCULATION_ERROR	16#000C	16#0000	Berechnungsfehler; falsche Achsposition

Abb.6-87: Fehlercodes Funktionsbaustein MB_PositionMonitoring_PosCollision

6.8 Funktionsbaustein zur MagicBelt-Anwendung

6.8.1 Einführung und Überblick

Bei der "MagicBelt"-Anwendung werden Produkte aus einem nahezu kontinuierlichem Produktstrom gruppiert und typischerweise einer Zweitverpackung (wie z.B. Karton) zugeführt. "MagicBelt"-Anwendungen sind auch unter den Begriffen Zweizugkette, DualBelt, MagicChain bekannt. Jede Kette wird bei dieser Anwendung über eine eigene Instanz des nachfolgend beschriebenen Funktionsbausteines MB_MagicBeltType01 angesteuert. Die Funktionsbausteine kommunizieren untereinander durch die angelegten Datenstrukturen.

ML_TechMotion.library

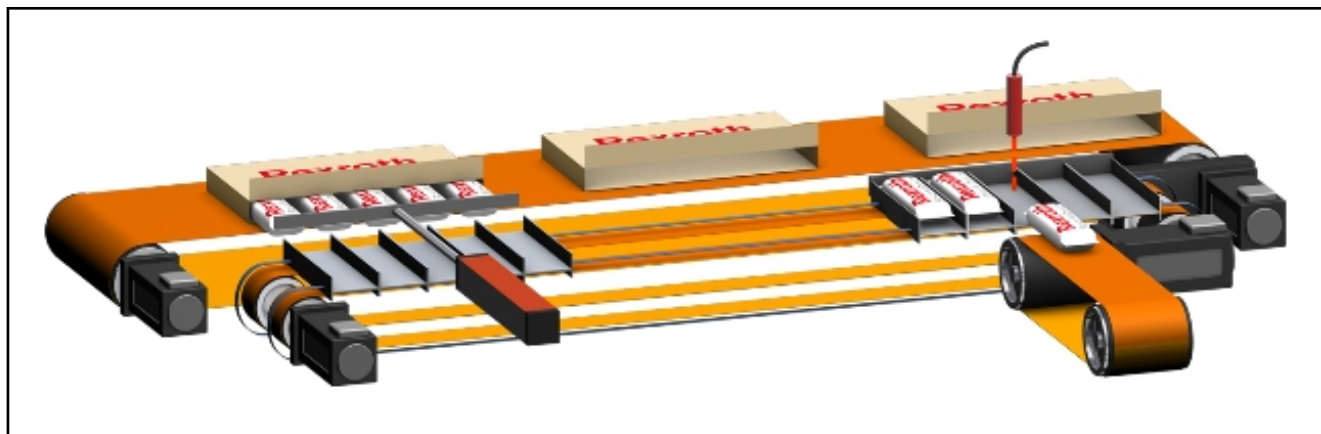


Abb.6-88: MagicBelt Applikation

Begriffsdefinitionen

Die nachfolgend verwendeten Begriffe sind wie folgt definiert:

- Kette: Die Kette (Chain) wird durch die Servomotoren angetrieben. Eine Kette kann bis zu 4 Züge enthalten
- Zug: Der Zug (Train) dient zum Gruppieren der Produkte und beinhaltet mehrere Fächer
- Fach: Ein Fach (Pocket) kann ein oder mehrere Produkte beinhalten und ist ein Teil eines Zuges
- InfeedMaster: Antriebs-Achse (oder Encoder) des Synchronrades bei Synchronradzuführung
- InfeedAxis: Intern genutzte virtuelle Achse um Infeed Bewegung auszuführen

Definitionen und Randbedingungen

In der Praxis werden oftmals mehrere Züge pro Kette eingesetzt, da dies zu kürzeren Verfahrbewegungen und Geschwindigkeiten führt. Folglich kann damit ein "kleinerer" Motor eingesetzt werden, und der mechanische Verschleiß von Kette und Motor wird reduziert.

Sind mehrere Züge auf einer Kette angebracht wird im Folgenden davon ausgegangen dass:

- Der Abstand von Zuanfang zum nächsten Zuanfang gleich ist (symmetrische Zugverteilung)
- Alle Züge einer Kette den gleichen Fächer Aufbau haben (z.B. Anzahl und Größe der Fächer)
- Alle Ketten mit der gleichen Anzahl von Zügen bestückt sind

Der Modulowert eines Ketten-Antriebes wird damit durch den Abstand von Zug-Vorderkante zu der nächsten Zug-Vorderkante definiert (translatorische Einheit). Befindet sich z.B. nur ein Zug auf der Kette, dann ist der Modulowert gleich der Kettenlänge.

Der Modulowert wird damit über die folgende Gleichung festgelegt:

$$\text{ModuloValue} = \frac{\text{Chain Length}}{\text{Trains Per Chain}}$$

Abb.6-89: Berechnung des Modulowerts

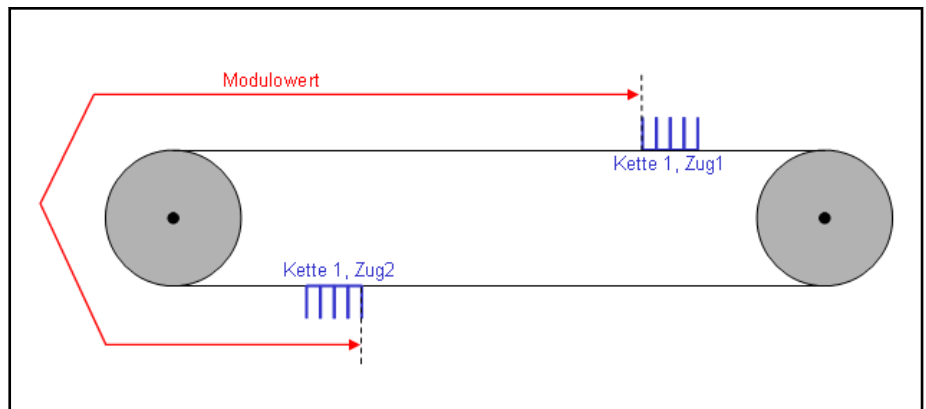


Abb.6-90: MagicBelt Definitionen Modulwert

6.8.2 MB_MagicBeltType01

Kurzbeschreibung

Der Funktionsbaustein enthält Funktionalitäten zur Ansteuerung einer Kette einer MagicBelt Applikation. Details zur Funktion sind in der nachfolgenden Funktionsbeschreibung enthalten. Je nach angewählter Betriebsart wird die jeweilige Kette entsprechend angesteuert:

- **Jog Mode:**
Die Züge können einzeln oder synchron wahlweise kontinuierlich oder inkrementell vertippt werden
- **Tune Mode:**
Um Dynamikgrenzwerte (z.B. maximale Beschleunigung) zu ermitteln werden die Züge synchronisiert, und fortlaufend um eine einstellbare Distanz verfahren
- **Automatik Mode:**
Im Wechselspiel fahren die Züge jeweils die Be- und Entladeposition(en) an



Der Betrieb des Funktionsbausteines erfordert besondere Randbedingungen (z.B. Modulo-Parametrierung, zyklischer Aufruf des FB in der Motion-Task...).

Bitte beachten Sie daher die Inbetriebnahmehinweise am Ende dieses Kapitels.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library

Abb.6-91: Referenztablelle Funktionsbaustein MB_MagicBeltType01

Leistungsmerkmale

- Maximal können jeweils 3 Ketten verwaltet werden (über 3 FB-Instanzen für Kette A, B, C)
- Auf einer Kette können maximal 4 Züge mit gleichen Abständen angeordnet sein
- Es können maximal 40 Fächer mit **unterschiedlichen** Fächerbreiten pro Zug verwendet werden. Bei mehreren **gleichlangen** Fächern vervielfacht sich die maximale Fächer-Anzahl entsprechend, siehe auch Strukturelement MB_MBELT_INFEED.InfeedStepData
- Getaktetes und synchronisiertes Beladen wird unterstützt
- Es können maximal 10 verschiedene Entladepositionen angefahren werden

ML_TechMotion.library

Schnittstellenbeschreibung

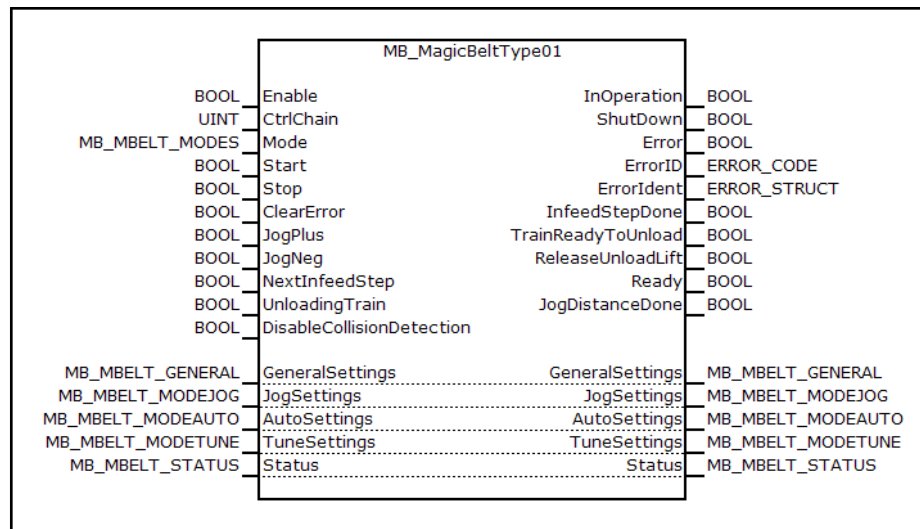


Abb.6-92: Funktionsbaustein MB_MagicBeltType01

Über die Funktionsbaustein-Eingänge (z.B. Start, Stopp...) werden die Ketten von Anwendungsprogramm angesteuert. Die Konfigurationsdaten für den Betrieb werden über die VAR_IN_OUT Datenstrukturen (z.B. AutoSettings...) angelegt. Über eine weitere "Status"-VAR_IN_OUT Datenstruktur stehen zusätzliche Diagnose-Informationen zur Verfügung.

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	GeneralSettings	MB_MBELT_GENE-RAL	Allgemeine Konfiguration der Magic Belt Anwendung
	JogSettings	MB_MBELT_MODE-JOG	Einstellungen für den Jog Mode
	AutoSettings	MB_MBELT_MODE-AUTO	Einstellungen für den Auto Mode
	TuneSettings	MB_MBELT_MODE-TUNE	Einstellungen für den Tune Mode
	Status	MB_MBELT_STATUS	Statusinformationen zu den Ketten
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins, zustandsgesteuert
	CtrlChain	UINT	Wählt die anzusteuernde Kettennummer aus: 1: Kette A, 2: Kette B, 3: Kette C
	Mode	MB_MBELT_MODES	Betriebsarten-Anwahl. Ein Wechsel zwischen den Betriebsarten wird nur in dem "Stopped"-Zustand ausgeführt
	Start	BOOL	Positive Flanke von Start startet die Achsbewegung (siehe auch StartType)
	Stop	BOOL	Positive Flanke von Stop stoppt die Achsbewegung (siehe auch StopType)
	ClearError	BOOL	Positive Flanke löscht einen anstehenden Fehler
	JogPlus	BOOL	Vertippen der Achsen in positive Richtung (nur bei aktivem Jog-Mode)

E/A-Typ	Name	Datentyp	Beschreibung
	JogNeg	BOOL	Vertippen der Achsen in negative Richtung (nur bei aktivem Jog-Mode)
	NextInfeedStep	BOOL	Bei Infeed Typ "FEED_INDEXED": Nach positiver Flanke und Ablauf der Zeit "StartDelay-Time" wird der Zug zum nächsten Fach weitergetaktet Bei Infeed Typ "FEED_SYNCHRONIZED" : Eine positive Flanke innerhalb des Erwartungsfensters ("ProductInExpectationWindow" = TRUE) signalisiert ein Produkt im Fach des Eintaktrades "Infeed-Master"
	UnloadingTrain	BOOL	Der Eingang muss beim Entladen des Zuges solange gesetzt bleiben bis der Entladevorgang abgeschlossen ist. Der Zug verbleibt solange an der Entladeposition wie dieses Signal gesetzt ist. Mit negativer Flanke wird der Zug zur nächsten Ent- bzw. Beladeposition weitergetaktet
	DisableCollisionDetection	BOOL	Schaltet die Kollisionsüberwachung ab, um z.B. nach einer Kollision die Ketten auseinander zu fahren
VAR_OUTPUT	InOperation	BOOL	Fehlerfreier Betrieb, Ausgangsdaten sind gültig
	ShutDown	BOOL	Definiertes Stillsetzen nach negativer Flanke an Enable
	Error	BOOL	Signalisiert fehlerhaften Zustand
	ErrorID	ERROR_CODE	Fehler Diagnose
	ErrorIdent	ERROR_STRUCT	Detail Fehler Diagnose
	InfeedStepDone	BOOL	Ausgabe nur bei FB der Kette A (CtrlChain = 1 ChainA). Ausgang ist nur für die Infeed Variante „FEED_INDEXED“ relevant. Der Ausgang wird gesetzt wenn der InfeedStep ausgeführt wurde. Mit positiver Flanke von NextInfeedStep wird der Ausgang zurückgesetzt
	TrainReadyToUnload	BOOL	Zug steht an der Entladeposition und ist bereit zur Entladung
	ReleaseUnloadLift	BOOL	Freigabe des Entlade-Rechens
	Ready	BOOL	Wird nur bei FB mit "CtrlChain" = 1 (ChainA) ausgegeben: Der Ausgang wird gesetzt wenn der einfließende Produktstrom vom MagicBelt aufgenommen werden kann, siehe auch Produktstauüberwachung
	JogDistanceDone	BOOL	Der Ausgang wird gesetzt wenn bei inkrementellem Joggen die vorgegebene Distanz zurückgelegt wurde. Der Ausgang wird zurückgesetzt wenn "Jog-Plus"/"JogNeg" zurückgesetzt wird

Abb.6-93: Schnittstellenvariablen Funktionsbaustein MB_MagicBeltType01

Min./Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

ML_TechMotion.library

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
CtrlChain	UINT	1	3	1	Pos. Flanke von "Enable"
Mode	MB_MBELT_MODES	MODE_AUTO	MODE_JOG	MODE_AUTO	Kontinuierlich
Start	BOOL			FALSE	Kontinuierlich im Auto- und Tune-Mode
Stop	BOOL			FALSE	Kontinuierlich im Auto- und Tune-Mode
ClearError	BOOL			FALSE	Kontinuierlich im Error-Mode
JogPlus	BOOL			FALSE	Kontinuierlich im Jog-Mode
JogNeg	BOOL			FALSE	Kontinuierlich im Jog-Mode
NextInfeedStep	BOOL			FALSE	Kontinuierlich im Auto-Mode
UnloadingTrain	BOOL			FALSE	Kontinuierlich im Auto-Mode
DisableCollision-Detection	BOOL			FALSE	Kontinuierlich

Abb.6-94: Min-/Max-/Default-Werte Funktionsbaustein MB_MagicBeltType01

MB_MBELT_MODES

Die Enumeration "MB_MBELT_MODES" enthält die auswählbaren Betriebsarten des Magic Belt Funktionsbausteines.

Element	Wert	Beschreibung
MODE_AUTO	16#100	Auto Mode
MODE_TUNE	16#300	Tune Mode
MODE_JOG	16#D00	Jog Mode

Abb.6-95: Elemente des Aufzählungstyps MB_MBELT_MODES

Verwendete Datenstrukturen/
Überblick

Die folgenden Datenstrukturen werden von dem Funktionsbaustein MB_MagicBeltType01 verwendet. Die Datenstrukturen sind nachfolgend dargestellt.

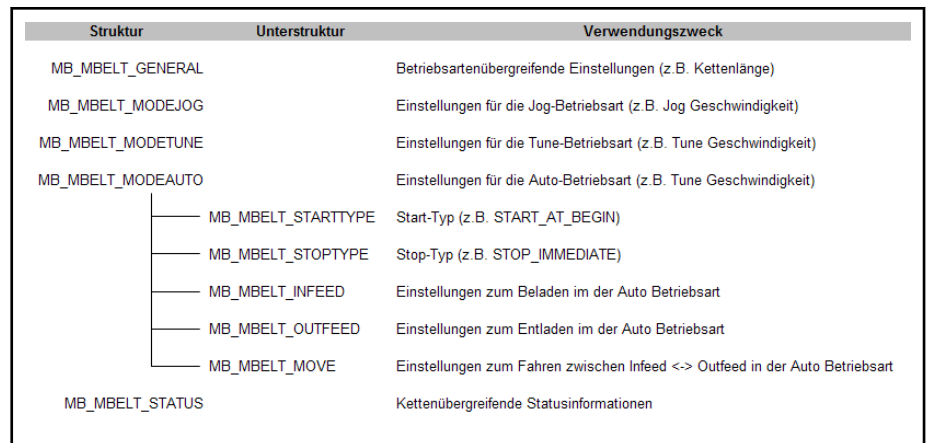


Abb.6-96: Datenstrukturen des MagicBelt Funktionsbausteines

MB_MBELT_GENERAL

Die Datenstruktur "MB_MBELT_GENERAL" enthält ketten- und betriebsartenübergreifende Einstellungen:

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
ChainLength	REAL	4000.0	pos. Flanke Enable	Kettenlänge [mm]; Minimum = 1 Max = undefiniert
NoOfChains	UINT	2	pos. Flanke Enable	Kettenanzahl Minimum = 1 Maximum = 3
TrainsPerChain	UINT	1	pos. Flanke Enable	Züge pro Kette Minimum = 1 Maximum = 4
DischargeBackwards	BOOL	FALSE	kontinuierlich	Bestimmt, durch welche Bewegungsrichtung die Züge entleert werden. FALSE: Fahrt über 2/3 Bandlänge in positive Richtung TRUE: Fahrt über 2/3 Bandlänge in negative Richtung
AllowNegDirection	BOOL	FALSE	kontinuierlich	Erlaubt die Bewegung der Züge in negative Richtung
TrainDistance	REAL	5.0	kontinuierlich	Ein nacheilender Zug wird mit diesem Abstand auf einen voreilenden Zug aufschließen (Verwendung im Automatik Mode) [mm]. Die TrainDistance muss größer als die SafetyDistance vorgegeben werden, da ansonsten die Kollisionsüberwachung auslöst. Minimum = 0.0mm Maximum = Modulowert/4
SafetyDistance	REAL	0.5	kontinuierlich	Sicherheitsabstand der Kollisions-Überwachung [mm] Minimum = -10mm Maximum = 100mm

ML_TechMotion.library

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
StopRamp	REAL	10000.0	kontinuierlich	Stopprampen-Verzögerung bei Fehlerreaktion oder Stopp mit der Stoppvariante "STOP_IMMEDIATE" . Die Stopp-Rampe muss größer oder mindestens gleich als der Maximalwert aller anderen Beschleunigungen (MaxMoveAcceleration; MaxInfeedAcceleration; MaxOutfeedAcceleration) sein. Minimum = siehe Beschreibung oben Maximum = undefiniert
StartAtLastStep-Window	REAL	100	kontinuierlich	Bei der Startoption "StartAtLastStep" werden die aktuellen Achspositionen mit den gespeicherten Positionen nach Erreichen des Stillstandes verglichen. Nur wenn die Positions Differenz aller Achsen kleiner als das hier vorgegebene Fenster ist, wird gemäß der Startoption "StartAtLastStep" weiter verfahren. Falls mindestens eine Achse außerhalb dieses Fensters liegt, so wird mit der Startoption "Start_At_Begin" weiter verfahren. Minimum: 0mm Maximum: ModuloValue
Feedrate	UINT	100	kontinuierlich	Alle Geschwindigkeiten im "Auto Mode" werden entsprechend reduziert, falls "FeedrateOverride" < 100 % vorgegeben wird. Minimum: 10% Maximum: 100%
InfeedMaster	AXIS_REF	NO_OBJECT	pos. Flanke Enable	Definiert die Master-Achse zum synchronisierten Beladen mit Radzuführung. Nur notwendig wenn synchronisiertes Beladen angewählt ist
InfeedAxis	AXIS_REF	NO_OBJECT	pos. Flanke Enable	Für die Infeed Bewegung genutzte virtuelle Achse "Infeed Achse"
DummyMaster	AXIS_REF	NO_OBJECT	pos. Flanke Enable	Zur internen "FlexProfile" -Positionierung muss hier eine Dummy-Achse angelegt werden. Der Achstyp und die Achseinstellungen sind nicht relevant. Die Achse wird nicht aktiv angesteuert und kann anwendungsseitig für andere Zwecke genutzt werden. Es wird nur eine gültige Achsreferenz benötigt
ChainAxes	ARRAY [1..3] OF AXIS_REF		pos. Flanke Enable	Achszuordnung der Ketten ChainAxes[1] = Chain A ChainAxes[2] = Chain B ChainAxes[3] = Chain C (optional)

Abb.6-97: Schnittstellenbeschreibung und Defaultwert der Datenstruktur MB_MBELT_GENERAL

MB_MBELT_MODEJOG

Diese Datenstruktur "MB_MBELT_MODEJOG" enthält die kettenübergreifenden Daten für den Jog Mode.

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
Incremental	BOOL	FALSE	Pos. Flanke Jog-Plus/JogNeg	TRUE: Mit einer positiven Flanke an den "Jog-Plus"/"JogNeg" wird die "JogDistance" in positive oder negative Richtung verfahren FALSE: Fahrt in positive bzw. negative Richtung solange "JogPlus"/"JogNeg" betätigt ist
SingleAxisJog	BOOL	FALSE	Pos. Flanke Jog-Plus/JogNeg	FALSE: Alle Züge werden synchron vertippt TRUE: Es wird nur der durch "JogAxisSelection" ausgewählte Zug vertippt
JogAxisSelection	UINT	1	Pos. Flanke Jog-Plus/JogNeg	Auswahl der Achse zum Einzelachstippen 1: Chain A, 2: Chain B, 3: Chain C
JogVelocity	REAL	100.0	Pos. Flanke Jog-Plus/JogNeg	Jog-Geschwindigkeit
JogAcceleration	REAL	1000.0	Pos. Flanke Jog-Plus/JogNeg	Jog-Beschleunigung
JogDistance	REAL	2.0	Pos. Flanke Jog-Plus/JogNeg	Jog-Distanz bei inkrementellen Joggen

Abb.6-98: Schnittstellenbeschreibung und Defaultwert der Datenstruktur MB_MBELT_MODEJOG

MB_MBELT_MODETUNE

Diese Datenstruktur "MB_MBELT_MODETUNE" enthält die kettenübergreifenden Daten für den Tune-Mode.

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
PosDirectionOnly	BOOL	FALSE	bei Bewegungsstart	TRUE: Es wird mit der "TuneDistance" relativ in positive Richtung gefahren FALSE: Von der Ausgangslage beim Start wird die "TuneDistance" im Wechselspiel Vor- und Rückwärts verfahren
MotionLaw	MC_STEP_MODE	X_FIT_SINE_TRAPEZE_ALIM	bei Bewegungsstart	Spezifiziert das Bewegungsgesetz für einen Tune Positioniervorgang. Verfügbare Bewegungsgesetze sind: • X_FIT_VEL_TRAPEZE_ALIM • X_FIT_SINE_TRAPEZE_ALIM • X_FIT_ACC_TRAPEZE_JLIM • REST_IN_REST_INCLINEDSINE • REST_IN_REST_MOD_TRAPEZE • X_MOTION_IN_MOTION_POLY5_VLIM Detailinformationen zu den Bewegungsgesetzen und dem Datentyp MC_STEP_MODE sind in der FlexProfile Dokumentation enthalten
TuneVelocity	REAL	1000.0	bei Bewegungsstart	Maximale Positionier Geschwindigkeit
TuneAcceleration	REAL	100.0	bei Bewegungsstart	Maximale Positionier Beschleunigung
TuneJerk	REAL	200.0	bei Bewegungsstart	Tune-Ruckbegrenzung (wirkt nur bei ruckbegrenzten Bewegungsgesetzen)

ML_TechMotion.library

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
TunePosTime	REAL	500	bei Bewegungs-start	Tune-Positionierzeit, wirkt bei allen Bewegungsgesetzen außer: • X_FIT_VEL_TRAPEZE ALIM • X_FIT_SINE_TRAPEZE ALIM • X_FIT_ACC_TRAPEZE J LIM
TuneDistance	REAL	10.0	bei Bewegungs-start	Verfahrweg pro Positioniervorgang

Abb.6-99: Schnittstellenbeschreibung und Defaultwert der Datenstruktur MB_MBELT_MODETUNE

MB_MBELT_MODEAUTO Die Datenstruktur "MB_MBELT_MODEAUTO" enthält kettenübergreifende Daten für den Automatik Mode.

Strukturelement	Typ	Default-Wert	Beschreibung
StartType	MB_BELT_START-TYPE	START_AT_BEGIN	Legt das Startverhalten für den Automatik Mode fest, siehe Beschreibung der Enumeration MB_MBELT_STARTTYPE
StopType	MB_BELT_STOPTYPE	STOP_CYCLE_COMPLETE	Legt das Stopverhalten für den Automatik Mode fest, siehe Beschreibung von "MB_MBELT_STOPTYPE". Übernahme bei pos. Stopp oder neg. Enable Flanke
Infeed	MB_BELT_INFEED		Enthält Daten für das Beladen der Züge
Outfeed	MB_BELT_OUTFEED		Enthält Daten für das Entladen der Züge
MoveTo	MB_BELT_MOVE		Enthält Daten für die Bewegung zwischen Be- und Entladen

Abb.6-100: Schnittstellenbeschreibung und Defaultwert der Datenstruktur MB_MBELT_MODEAUTO

MB_MBELT_STARTTYPE Die Enumeration "MB_MBELT_STARTTYPE" enthält die möglichen Startvarianten für den Automatik Mode:

Element	Wert	Beschreibung
START_AT_BEGIN	1	Definierter Start von Grundstellung
START_WITH_TRAIN_UNLOAD	2	Definierter Start von Grundstellung mit vorgeschalteter Entladung der Züge
START_AT_LAST_STEP	3	Start mit letztem (unterbrochenen) Zustand (z.B. nach Fehler oder Stopp)

Abb.6-101: Elemente des Aufzählungstyps MB_MBELT_STARTTYPE

MB_MBELT_STOPTYPE Die Enumeration "MB_MBELT_STOPTYPE" enthält die möglichen Stopvarianten für den Automatik Mode:

Element	Wert	Beschreibung
STOP_IMMEDIATE	1	Sofortiges Stoppen mit Rampe MB_MBELT_GENERAL.StopRamp
STOP_CYCLE_COMPLETE	2	Aktiver Produkt-Zyklus (Infeed Step) wird noch ausgeführt, danach wird angehalten (definiertes Stillsetzen). Alle Züge die nicht mit der InfeedAxis synchronisiert sind (z.B. bei Fahrten zwischen Outfeedpositionen) werden sofort gestoppt
STOP_IMMEDIATE_SYNC	3	Diese Stoppvariante ist nur für Anwendungen mit synchronisiertem Infeed sinnvoll. Die Stoppvariante verhält sich wie STOP_IMMEDIATE mit dem Unterschied dass die Achsen, die im Synchronverbund mit dem InfeedMaster stehen, weiter synchronisiert bleiben. Diese Achsen werden durch das Verzögern des externen InfeedMasters mit verzögert. Das Verzögern des InfeedMasters muss außerhalb des FB sichergestellt werden

Abb.6-102: Elemente des Aufzählungstyps MB_MBELT_STOCTYPE

MB_MBELT_INFEED

Die Datenstruktur "MB_MBELT_INFEED" enthält die kettenübergreifenden Daten für das Beladen im Automatik Mode:

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
InfeedType	MB_MBELT_FEEDTYPE	FEED_INDEXED	pos. Flanke Enable	Beladeverfahren (getaktet, synchronisiert, mit Geschwindigkeit, siehe Enumeration "MB_MBELT_FEEDTYPE")
StartDelayTime	TIME	T#50ms	pos. Flanke von NextInfeedStep	Nur für getaktetes Beladen relevant: Verzugszeit (pos. Flanke "NextInfeedStep"--> Start Bewegung) zum Weitertakten Minimum = T#0ms Maximum = T#1h
InfeedLoadTime	TIME	T#10ms	pos. Flanke von NextInfeedStep	Nur für getaktetes Beladen relevant: Zeitbedarf für die Zug Beladung. Innerhalb dieser Zeit verweilt der Zug im Stillstand um das Produkt beim Beladen nicht zu beschädigen. Eingang ist unwirksam bei "MotionLaw": X_FIT_ACC_TRAPEZE_JLIM X_FIT_SINE_TRAPEZE_ALIM X_FIT_VEL_TRAPEZE_ALIM Da in diesem Fall die vorgegeben a,v, j Maximalwerte verwendet werden
MaxProductRate	UINT	200	pos. Flanke von NextInfeedStep	Nur für getaktetes Beladen relevant: Maximale Anzahl von Produkten pro Minute die eingetaktet werden sollen [1/min] (Produktstrom). Eingang ist unwirksam bei "MotionLaw ": X_FIT_ACC_TRAPEZE_JLIM X_FIT_SINE_TRAPEZE_ALIM X_FIT_VEL_TRAPEZE_ALIM Da in diesem Fall die vorgegeben a,v, j Maximalwerte verwendet werden

ML_TechMotion.library

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
MotionLaw	MC_STEP_MODE	VELOCITY_IN_VELOCITY_MOD_SINE	pos. Flanke von NextInfeedStep	Nur für getaktetes Beladen relevant: Spezifiziert das Bewegungsgesetz für einen Infeed Schritt (Detail-Beschreibung der Bewegungsgesetze siehe FlexProfile Dokumentation). Folgende Bewegungsgesetze werden für MagicBelt Anwendungen empfohlen: • VELOCITY_IN_VELOCITY_MOD_SINE: Modifizierte Sinoide / ruckarm aber hohe Beschleunigungsspitze • X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM: Modifiziertes Beschleunigungstrapez / stark ruckbehaftet aber geringste Beschleunigungsspitze • X_FIT_ACC_TRAPEZE_JLIM: Ruckbegrenzte Bewegung. Bei diesem Bewegungsgesetz wirkt der Eingang "MaxProductRate" nicht, da die a,v, j Maximalwerte verwendet werden Achtung: Bei "REST_IN_REST_XXX"- Bewegungsgesetze muss der vorausgegangene Infeed Schritt bei Start eines neuen Infeed Schrittes abgeschlossen sein. Es wird der Fehler Add1 = 16#1A0F ausgegeben wenn diese Bedingung nicht erfüllt ist
MinDistance-ToOutfeed	REAL	400.0	kontinuierlich	Minimale Distanz zwischen Zug im "Infeed "-Zustand und einem weiteren Zug im "Outfeed "-Zustand. Wird die Distanz unterschritten wird der "Ready" Ausgang zurückgesetzt um einen Produktstau zu vermeiden, Details siehe Produktstau-Überwachung
MaxInfeedVelocity	REAL	300000.0	kontinuierlich	Infeed Schritt Maximalgeschwindigkeit: Nur wirksam bei "MotionLaw" : X_FIT_ACC_TRAPEZE_JLIM X_FIT_SINE_TRAPEZE_ALIM X_FIT_VEL_TRAPEZE_ALIM X_MOTION_IN_MOTION_POLY5_VLIM X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM
MaxInfeedAcceleration	REAL	5000.0	kontinuierlich	Infeed Schritt Maximalbeschleunigung: Nur wirksam bei "MotionLaw": X_FIT_ACC_TRAPEZE_JLIM X_FIT_SINE_TRAPEZE_ALIM X_FIT_VEL_TRAPEZE_ALIM X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM
MaxInfeedJerk	REAL	1000000.0	kontinuierlich	Infeed Schritt Maximalruck: Nur wirksam bei "MotionLaw": X_FIT_ACC_TRAPEZE_JLIM

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
ProductBuffer	UINT	3	pos. Flanke Enable	Nur für synchronisiertes Beladen: Anzahl der Produkte zwischen Produkterkennung und Produktauswurf Minimum: 0 Maximum: 64
ProdExpectationWindowStart	REAL	30.0	kontinuierlich	Nur für synchronisiertes Beladen: Start des Produkt-Erwartungsfensters. Innerhalb des Erwartungsfensters wird der Eingang "NextInfeedStep" ausgewertet. Minimum = 0° Maximum = ProdExpectationWindowEnd
ProdExpectationWindowEnd	REAL	180.0	kontinuierlich	Nur für synchronisiertes Beladen: Ende des Produkt-Erwartungsfensters. Innerhalb des Erwartungsfensters wird der Eingang "NextInfeedStep" ausgewertet. Minimum = ProdExpectationWindowStart Maximum = 360°
FirstInfeedPosition	REAL	500.0	kontinuierlich	Erste (absolute) Ladeposition Minimum = 0 Maximum = ModuloValue
AdditionalSteps	UINT	0	kontinuierlich	Der bereits vollständig beladene Zug kann mit "AdditionalSteps" noch einen oder mehrere Steps vor dem nächsten Zug hergeschoben werden, bevor er zur Outfeedposition fährt (z.B. bis Niederhalter erreicht wird)
InfeedStepData	ARRAY [1..3, 1..40] OF MB_BELT_INFEEDSTEPS		kontinuierlich	Liste mit der Größe der Fächer und Produktanzahl pro Fach für alle Ketten. "InfeedStepData"[1,x] = Kette A "InfeedStepData"[2,x] = Kette B "InfeedStepData"[3,x] = Kette C (optional) Bei dem synchronisierten Beladen wird immer von einem Produkt pro Fach ausgegangen

Abb.6-103: Schnittstellenbeschreibung und Defaultwert Datenstruktur MB_MBELT_INFEED

MB_MBELT_FEEDTYPE

Die Enumeration "MB_MBELT_FEEDTYPE" enthält die möglichen Be- und Entladeverfahren der Züge:

Element	Wert	Beschreibung
FEED_INDEXED	1	Getaktetes Beladen
FEED_SYNCHRONIZED	2	Synchronisiertes Beladen

Abb.6-104: Elemente des Aufzählungstyps MB_MBELT_FEEDTYPE

MB_MBELT_INFEEDSTEPS

Die Datenstruktur "MB_MBELT_INFEEDSTEPS" enthält kettenspezifische Daten für das Beladen im Automatik Mode:

ML_TechMotion.library

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
PocketWidth	REAL	50.0	kontinuierlich	Abstand von Vorderkante des aktuellen zur Vorderkante des nachfolgenden Faches. Sind mehrere Fächer gleich groß, so kann die Anzahl der gleich großen Fächer über das Element "PocketQuantity" vorgegeben werden
PocketQuantity	UINT	1	kontinuierlich	Anzahl der Fächer mit der gleichen Größe -1 = Kennung für Gruppenbildung (siehe Beispiel) 0 = Kennung für Zugende (siehe Beispiel)
ProductsPerPocket	UINT	0	kontinuierlich	Produktanzahl pro Fach (nur für getaktetes Beladen relevant)

Abb.6-105: Schnittstellenbeschreibung und Defaultwert der Datenstruktur MB_MBELT_INFEEDSTEPS

Beispiel:

Beispiel 1: Gleiche Fächer (siehe auch Abbildung unten):

Zug besteht aus 10 gleichen Fächern mit 50mm Fachbreite mit jeweils einem Produkt

ChainA_InfeedData[1].PocketWidth = 50mm

ChainA_InfeedData[1].PocketQuantity = 10

ChainA_InfeedData[1].ProductsPerPocket = 1

ChainA_InfeedData[2].PocketWidth = don't care

ChainA_InfeedData[2].PocketQuantity = 0 <- Zug Ende Kennung

ChainA_InfeedData[2].ProductsPerPocket = don't care

Beispiel:

Beispiel 2: Unterschiedliche Fächer (siehe auch Abbildung unten):

Der Zug besteht aus 3 unterschiedlichen Fächern (40mm, 80mm, 120mm) mit (1,2,3) Produkten pro Fach

ChainA_InfeedData[1].PocketWidth = 40mm

ChainA_InfeedData[1].PocketQuantity = 1

ChainA_InfeedData[1].ProductsPerPocket = 1

ChainA_InfeedData[2].PocketWidth = 80mm

ChainA_InfeedData[2].PocketQuantity = 1

ChainA_InfeedData[2].ProductsPerPocket = 2

ChainA_InfeedData[3].PocketWidth = 120mm

ChainA_InfeedData[3].PocketQuantity = 1

ChainA_InfeedData[3].ProductsPerPocket = 3

ChainA_InfeedData[4].PocketWidth = don't care

ChainA_InfeedData[4].PocketQuantity = 0 <- Zug Ende Kennung

ChainA_InfeedData[4].ProductsPerPocket = don't care

Beispiel:

Beispiel 3: Fächergruppe (siehe auch Abbildung unten):

Der Zug besteht aus einer Fächergruppe mit 5 * 3 unterschiedlichen Fächern (40mm, 80mm, 120mm) mit (1,2,3) Produkten pro Fach

ChainA_InfeedData[1].PocketWidth = 40mm

ChainA_InfeedData[1].PocketQuantity = -1 <-Gruppen Kennung

ChainA_InfeedData[1].ProductsPerPocket = 1

ChainA_InfeedData[2].PocketWidth = 80mm

ChainA_InfeedData[2].PocketQuantity = -1 <-Gruppen Kennung

ChainA_InfeedData[2].ProductsPerPocket = 2

ChainA_InfeedData[3].PocketWidth = 120mm

ChainA_InfeedData[3].PocketQuantity = 5 <-Gruppen Ende Kennung (Wert > 0) mit Anzahl der Gruppen

ChainA_InfeedData[3].ProductsPerPocket = 3

ChainA_InfeedData[4].PocketWidth = don't care

ChainA_InfeedData[4].PocketQuantity = 0 <-Zug Ende Kennung

ChainA_InfeedData[4].ProductsPerPocket = don't care

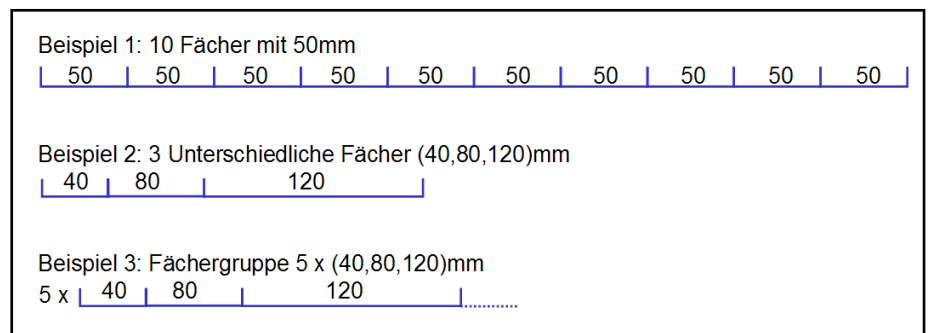


Abb.6-106: Zugkonfigurationen der oben gezeigten Beispiele

MB_MBELT_OUTFEED

Die Datenstruktur "MB_MBELT_OUTFEED" enthält die kettenübergreifenden Daten für das Entladen im Automatik Mode:

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
MotionLaw	MC_STEP_MO- DE	X_FIT_SI- NE_TRAPE- ZE_ALIM	pos. Flanke von UnloadingTrain	Spezifiziert das Bewegungsgesetz für die Be- wegung zur nächsten Outfeedposition (Detail- Beschreibung der Bewegungsgesetze siehe FlexProfile Dokumentation) Beschleunigungsbegrenzte Bewegung (X_FIT_VEL_TRAPEZE_ALIM) Ruckbegrenzte Bewegung (X_FIT_ACC_TRA- PEZE_JLIM) Beschleunigungsbegrenzte Sinoide (X_FIT_SI- NE_TRAPEZE_ALIM)
MaxOutfeedVe- locity	REAL	60000.0	kontinuierlich	Maximale Geschwindigkeit zum Weiterfahren auf die nächste Entladeposition
MaxOutfeedAc- celeration	REAL	5000.0	kontinuierlich	Maximale Beschleunigung zum Weiterfahren auf die nächste Entladeposition

ML_TechMotion.library

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
MaxOutfeedJerk	REAL	0.0	kontinuierlich	Maximaler Ruck zum Weiterfahren auf die nächste Entladeposition Nur wirksam bei "MotionLaw" = X_FIT_ACC_TRAPEZE_JLIM
FirstOutfeedPosition	REAL	2000.0	kontinuierlich	Erste (absolute) Entladeposition
EnableUnloadHandshake	BOOL	FALSE	kontinuierlich	FALSE: Entladehandshake abgeschaltet, für Anwendungen ohne Kollisionsgefahr der Entnahmeverrichtung mit dem Zug (z.B. Entnahmegeroboter) TRUE: Entladehandshake eingeschaltet, für Anwendungen mit Kollisionsgefahr der Entnahmeverrichtung mit dem Zug. "UnloadLiftLowPos" und "UnloadLiftHighPos" müssen entsprechend gesetzt werden (z.B. Entnahmerechen)
OutfeedDistances	ARRAY [1..3, 2..10] OF REAL	0.0	kontinuierlich	Diese Liste ist nur notwendig wenn mehr als eine Entladestelle angefahren werden soll. In diesem Falle können über diese Liste die Entfernungen von Entladestelle zur nachfolgenden Entladestelle angegeben werden. Mit jeder negativen Flanke des Eingangs UnloadingTrain wird zur nächsten Entladestelle weiter getaktet. Das Ende der Liste ist erreicht wenn der Wert 0 eingetragen ist. "OutfeedDistances" [1,x] = Kette A "OutfeedDistances" [2,x] = Kette B "OutfeedDistances" [3,x] = Kette C (optional)
UnloadLiftLowPos	ARRAY [1...10] OF BOOL	TRUE	kontinuierlich	Entladerechen Lage-Endschalter/TRUE = Entladerechen eingezogen
UnloadLiftHighPos	ARRAY [1...10] OF BOOL	TRUE	kontinuierlich	Entladerechen Lage-Endschalter/TRUE = Entladerechen ausgefahren

Abb.6-107: Schnittstellenbeschreibung und Defaultwert der Datenstruktur MB_MBELT_OUTFEED

MB_MBELT_MOVE

Die Datenstruktur "MB_MBELT_MOVE" enthält die kettenübergreifenden Daten für das Fahren zwischen der Be- und Entladeposition.

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
MaxMoveVelocity	REAL	300000.0	Bei Start der Move Bewegung	Maximale Geschwindigkeit zum Fahren zwischen Be- und Entladestation und zum Aufsynchronisieren
MaxMoveAcceleration	REAL	10000.0	Bei Start der Move Bewegung	Maximale Beschleunigung/Verzögerung zum Fahren zwischen Be- und Entladestation
MaxMoveJerk	REAL	0.0	Bei Start der Move Bewegung	Maximaler Ruck zum Fahren zwischen Be- und Entladestation

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
LoadTrainAcceleration	UINT	100	Bei Start der Move Bewegung	Prozentwert der genutzten max. Beschleunigung (MaxMoveAcceleration) bzw. Verzögerung bei der Bewegung zwischen der Beladestation und der ersten Entladestation. Ein voll beladener Zug kann damit mit reduzierter Beschleunigung verfahren werden
BeginReduced-Vel1	REAL	0.0	Bei Start der Move Bewegung	Ab dieser Position (der Zug Vorderkante) fährt der Zug mit reduzierter Geschwindigkeit. Damit können die entstehenden Fliehkräfte, welche durch die Umlenkung in den Kurven entstehen reduziert werden
EndReduced-Vel1	REAL	0.0	Bei Start der Move Bewegung	Ab dieser Position (der Zug Vorderkante) ist die Geschwindigkeitsbegrenzung wieder aufgehoben
BeginReduced-Vel2	REAL	0.0	Bei Start der Move Bewegung	Ab dieser Position (der Zug Vorderkante) fährt der Zug mit reduzierter Geschwindigkeit. Damit können die entstehenden Fliehkräfte, welche durch die Umlenkung in den Kurven entstehen reduziert werden
EndReduced-Vel2	REAL	0.0	Bei Start der Move Bewegung	Ab dieser Position (der Zug Vorderkante) ist die Geschwindigkeitsbegrenzung wieder aufgehoben
ReducedVelocity	REAL	100000.0	Bei Start der Move Bewegung	Geschwindigkeitsbegrenzung zur Reduzierung der Fliehkräfte in den Kurven
AdaptiveSync-Ramp	BOOL	TRUE	Bei Start der Move Bewegung	Aktiviert adaptive Synchronisations-Rampe

Abb.6-108: Schnittstellenbeschreibung und Defaultwert der Datenstruktur MB_MBELT_MOVE

MB_MBELT_STATUS Die Datenstruktur " MB_MBELT_STATUS" enthält Statusinformationen des Zuges:

Strukturelement	Typ	Beschreibung
ChainStates	ARRAY [1..3] OF MB_MBELT_MODES	Aktueller Ist-Zustand der Ketten
PocketsToBeLoaded	UINT	Ein Zug im Zustand "Auto-Infeed" gibt über dieses Strukturelement aus, wie viele Fächer noch in dem Zug beladen werden müssen. Damit kann z.B. übergeordnet berechnet werden wie lange der Ladevorgang voraussichtlich noch andauert
ActInfeedStep	UINT	Ein Zug im Zustand "Auto-Infeed" gibt über dieses Strukturelement aus welcher "Infeed" Schritt aktuell bearbeitet wird
ProductInExpectationWindow	BOOL	Im Infeed Modus "FEED_SYNCHRONIZED" wird dieser Ausgang gesetzt, wenn sich die Achse "InfeedMaster" in dem Erwartungsfenster befindet. Das Erwartungsfenster wird durch die Strukturelemente "MB_BELT_INFEED", "ProdExpectationWindowStart" und "MB_MBELT_INFEED" und "ProdExpectationWindowEnd" definiert
AllChainsHomed	BOOL	Alle Ketten sind referenziert

ML_TechMotion.library

Strukturelement	Typ	Beschreibung
Error	BOOL	Kettenübergreifendes Error Flag
ErrorSourceChain	UINT	Fehlerverursachende Kettennummer
MeasuredInfeedStepRate	REAL	Gemessene InfeedStepRate [min-1] (wird durch den zeitlichen Abstand der „NextInfeedStep“ Flanken ermittelt)
MaxProdExpectationWindowEnd	REAL	Maximal mögliche Endposition des Erwartungsfensters "ProdExpectationWindowEnd" berechnet aus der aktuellen Geschwindigkeit der InfeedMaster Achse [°]
ChainInAutoInfeed	UINT	Kettennummer, die sich im Zustand "AUTO_INFEED" befindet
SupressedInfeedSteps	UINT	Anzahl der unterdrückten InfeedSteps. Wird inkrementiert, bei der Beladevariante "FEED_INDEXED" wenn eine "NextInfeedStep" Flanke angelegt wird und der Infeed-Zug aber den Infeed Step nicht ausführen kann (für Diagnosezwecke)
MaxNoOfInfeedSteps	UINT	Maximale Infeed Step Anzahl für den aktuellen Zug an der Infeed Position
FirstMasterChain	UINT	Nummer der Kette, die zuerst die FirstInfeedPosition erreicht (nur für interne Zwecke)
OutfeedPosOccupied	ARRAY [1...10] OF BOOL	Wird für jede Outfeedposition gesetzt wenn diese durch einen Zug oder den Entlade-Rechen besetzt ist
OccupiedChain	ARRAY [1...10] OF UINT	Kettennummer welche die Outfeed Position belegt
ChainCmdStates	ARRAY [1..3] OF MB_MBELT_STATES	Berechneter Zielzustand (nur für interne Zwecke)
RestoreData	ARRAY [1..3] OF MB_MBELT_RESTORE	Gespeicherte Informationen zum Stopp-Zeitpunkt um bei dem "StartType" =" START_AT_LAST_STEP" den letzten Zustand wieder herzustellen

Abb.6-109: Schnittstellenbeschreibung und Defaultwert der Datenstruktur MB_MBELT_STATUS

Funktionsbeschreibung Mit "Enable"=TRUE startet der Funktionsbaustein und die am "Mode"-Eingang angewählte Betriebsart wird nach der Initialisierung aktiv. Entsprechend der angewählten Betriebsart wird die Kette angesteuert. Ein Wechsel zwischen den Betriebsarten ist nur im sicheren "Stopped"-Zustand erlaubt, wobei die Ketten im Stillstand sind. Die Betriebsarten und deren Funktionalität werden nachfolgend beschrieben.

Folgende Eingänge sind betriebsartenunabhängig und wirken immer:

- Enable
- Mode
- CtrlChain
- ClearError
- DisableCollisionDetection

Folgende Ausgänge sind betriebsartenunabhängig und wirken immer:

- InOperation
- Error
- ErrorID
- ErrorIdent

- Shutdown

Folgende Ein/Ausgänge sind betriebsartenunabhängig und wirken immer:

- GeneralSettings
- Status



Alle Chain-Achsen müssen bei Start des FB (über Enable=TRUE) referenziert sein. Ist diese Bedingung nicht erfüllt, meldet der FB einen Fehler.

Jog Betriebsart

In der Jog Betriebsart sind folgende Ein/Ausgänge aktiv:

Funktionsbaustein Eingänge:

- JogPlus
- JogNeg

Funktionsbaustein Ausgänge:

- JogDistanceDone

Funktionsbaustein Ein/Ausgänge:

- JogSettings

Mit der Jog Betriebsart können zum Einrichten der Maschine die Züge inkrementell oder kontinuierlich vertippt werden. Nach Aktivierung der Jog Betriebsart sind die Züge zunächst im Stillstand (Stopped State). Das Joggen der Achsen ist mit den optional enthaltenen Entlade-Rechen verriegelt um Kollisionen zu vermeiden. Daher ist das anschließende Joggen nur möglich wenn:

- Der FB-Eingang "UnloadingTrain" zurückgesetzt ist UND
- die optionalen Entlade-Rechen eingezogen sind d.h. die Entlade-Rechen Endschalte Signale sind: "MB_MBELT_OUTFEED. UnloadLift-LowPos[1..10]" = TRUE

Ist die o.g. Bedingung erfüllt, können über die Eingänge "JogPlus" oder "JogNeg" die Züge in positive bzw. negative Richtung vertippt werden. Die folgenden Optionen sind dabei möglich:

- Über das Strukturelement "MB_MBELT_MODEJOG.Incremental" kann zwischen inkrementellem und kontinuierlichem Joggen umgeschaltet werden (Defaulteinstellung = kontinuierliches Joggen)
- Über das Strukturelement "MB_MBELT_MODEJOG.SingleAxisJog" kann das Einzelachs-joggen aktiviert werden. Der zu vertippende Zug wird dann über das Strukturelement "MB_MBELT_MODEJOG.JogAxis-Selection" ausgewählt (Defaulteinstellung = synchronisiertes Joggen)

Tune Betriebsart

In der Tune Betriebsart sind folgende Ein/Ausgänge aktiv:

Funktionsbaustein Eingänge:

- Start
- Stop

Funktionsbaustein Ein/Ausgänge:

- TuneSettings

Mit der Tune Betriebsart können beim Einrichten der Maschine die optimalen Dynamikeinstellungen/Bewegungsgesetze ermittelt werden. Im permanenten Start/Stopp Betrieb werden die Züge hierbei gemäß des ausgewählten Bewegungsgesetzes wahlweise in eine Richtung oder bipolar bewegt. Wie bei der Jog-Betriebsart ist das Starten der Achsen mit den optionalen Entlade-Rechen verriegelt: Detailbeschreibung siehe Jog-Betriebsart.

ML_TechMotion.library

Nach positiver Flanke von Start werden die Ketten synchronisiert und fortlaufend zyklisch um die Distanz "MB_MBELT_MODETUNE.TuneDistance" verfahren. Je nach Einstellung von "MB_MBELT_MODETUNE.PosDirectionOnly" werden die Züge entweder fortlaufend in positive Richtung oder im Wechselspiel Vor- und Zurück verfahren. Während der Fahrt können die Dynamikeinstellungen (z.B. "MB_MBELT_MODETUNE.TuneAcceleration") verändert werden um z.B. Maximalwerte für die Beschleunigungen zu ermitteln.

Auto Betriebsart

In der Auto Betriebsart sind folgende Ein/Ausgänge aktiv:

Funktionsbaustein Eingänge:

- Start
- Stop
- NextInfeedStep
- UnloadingTrain

Funktionsbaustein Ausgänge:

- InfeedStepDone
- TrainReadyToUnload
- ReleaseUnloadLift
- Ready

Funktionsbaustein Ein/Ausgänge:

- AutoSettings
- Status

Im Automatik-Mode werden die Züge im Wechselspiel zwischen den Be- und Entladepositionen bewegt, um den nahezu kontinuierlichen Produktstrom an der Beladeposition aufzunehmen und zu gruppieren. Ist der Zug voll beladen werden die Entladepositionen angefahren, um die Produktgruppen über einen Entlade-Rechen oder einen Entladeroboter zu entnehmen. Nach der Entladung fährt der jeweilige Zug wieder die Beladeposition an.

Start

Die positive Start Flanke gibt die Bewegung der Züge frei. Je nach Startoption ("AutoSettings.StartType") wird wie folgt gestartet:

START_AT_BEGIN:

Der Zug mit der geringsten Entfernung zur "MB_MBELT_INFEED.FirstInfeedPosition" ist der führende Zug und damit der Master. Hierbei wird auch berücksichtigt ob sich die Züge nur in positive Richtung oder bipolar bewegen dürfen (einstellbar über "MB_MBELT_GENERAL.AllowNegDirection").

Alle Slave Züge schließen mit dem Zugabstand "MB_MBELT_GENERAL.ChainLength" auf den führenden Zug auf. Der führende Zug fährt zur "MB_MBELT_INFEED.FirstInfeedPosition", alle anderen Züge folgen synchron. Nach Erreichen der "FirstInfeedPosition" startet das Beladen des Zuges.

START_WITH_TRAIN_UNLOAD:

Vorgehensweise wie bei START_AT_BEGIN mit dem Unterschied dass eine zusätzliche Zug-Entladebewegung um $\frac{2}{3}$ * "MB_MBELT_GENERAL.ChainLength" vor Fahrt auf die "FirstInfeed"-Position ausgeführt wird, um die Züge sicher zu entladen. Die Richtung der Entladebewegung kann hierbei durch das Strukturelement "MB_MBELT_GENERAL.DischargeBackwards" beeinflusst werden. Die Entladebewegung wird mit der maximalen Geschwindigkeit "MB_MBELT_MOVE.MaxMoveVelocity" ausgeführt.

START_AT_LAST_STEP:

Mit dieser Startoption kann der letzte Zustand nach Fehler oder Stopp wiederhergestellt werden. Bei dieser Startoption wird zunächst geprüft ob alle Achs-Positionen noch innerhalb des einstellbaren Fensters "MB_MBELT_GENERAL.StartAtLastStepWindow" liegen (im Vergleich zur abgespeicherten Position nach Stopp). Falls mindestens eine der Achsen außerhalb dieses Fensters liegt, so wird nach Startoption "START_AT_BEGIN" verfahren. Liegen alle Achsen innerhalb dieses Fensters, so werden im nächsten Schritt alle Achsen auf die zuletzt aktive Position vor Stopp oder Fehler repositioniert. Anschließend wird der letzte Zustand vor Stopp- oder Fehler wiederhergestellt. Zur Wiederherstellung werden die Achsen - wenn notwendig - repositioniert. Diese Startoption kann daher nur ausgeführt werden, wenn folgende Randbedingungen erfüllt sind:

- Positions Differenz zwischen aktueller Ist-Position und letzter Position der Chain-Achsen im Stillstand liegt innerhalb des "MB_MBELT_GENERAL.StartAtLastStepWindow". Ist die Bedingung nicht erfüllt, wird START_AT_BEGIN ausgeführt
- Die abgespeicherten letzten Zustände vor der Unterbrechung von allen Chain Achsen entsprechen dem Auto-Mode. Ist die Bedingung nicht erfüllt wird START_AT_BEGIN ausgeführt
- Ist eine Entladevorrichtung vorhanden (EnableUnloadHandshake=TRUE) und ist mindestens ein Entladerechen ausgefahren (UnloadLiftHighPos[n] = TRUE) und ist eine Repositionierung von mehr als 2mm an mindestens einer Achse erforderlich, so kann kein "START_AT_LAST_STEP" ausgeführt werden, da ein mechanischer Schaden auftreten könnte. In diesem Fall wird daher ein Fehler ausgegeben

Das Repositionieren wird wie folgt durchgeführt:

- Ketten werden synchronisiert (relative Synchronisation)
- Nur wenn 3 Ketten konfiguriert sind: Kette C positioniert auf letzte gültige abgespeicherte Position. Alle weiteren Ketten folgen synchron. Weiter wenn Kette C die Zielposition erreicht hat
- Kette B positioniert auf letzte gültige abgespeicherte Position. Kette C (wenn vorhanden) bleibt im Stillstand. Kette A folgt synchron. Weiter wenn Kette B die Zielposition erreicht hat
- Kette A positioniert auf letzte gültige abgespeicherte Position. Alle anderen Ketten bleiben im Stillstand. Weiter wenn Kette A die Zielposition erreicht hat
- Der letzte State und alle notwendigen internen Variablen werden wieder hergestellt, um den Zustand vor der Unterbrechung wieder zu erreichen

Beladen Alle Einstellungen zu dem Beladevorgang werden über das Strukturelement "MB_MBELT_MODEAUTO.Infeed" vorgegeben.

Grundsätzlich wird zwischen getaktetem und synchronisiertem Beladen des Zuges unterschieden. Bei dem getaktetem Beladen wird der Zug an der Beladeposition über den FB Eingang "NextInfeedStep" weitergetaktet bis alle Fächer des Zuges gefüllt sind.

Bei dem synchronisierten Beladen ist der Zug beim Beladevorgang synchronisiert mit der Produktzuführung (Produktzuführung z.B. über Eintaktrad). Dieses Beladeverfahren ist im allgemeinen produktschonender für höhere Taktzeiten. Bei diesem Verfahren wird immer von einer gleichen Fächerbreite ausgegangen. Falls ein Fach der Produktzuführung leer ist, muss der Zug bei dem Beladen taktisynchron für eine Fächerbreite aussetzen (Leererteilungskompensation).

Das Zug-Beladeverfahren wird über das Strukturelement "MB_MBELT_MODEAUTO.Infeed.InfeedType" festgelegt.

Getaktetes Beladen

Bei dem getakteten Beladen wirken alle Elemente der Datenstruktur MB_MBELT_MODEAUTO.Infeed (Ausnahme: ProductBuffer und ProductExpectationWindow wirken nicht).

Bei dem getakteten Beladen (AutoSettings.Infeed.InfeedType = FEED_INDEXED) kommen die Produkte über Transportbänder mit mehr oder weniger gleichen Abständen aber mit einem Mindestabstand zueinander. Auch bei quasi gleichen Abständen können einzelne oder mehrere Produkte fehlen. Das Produkt löst bei dem Weg in das Fach ein Sensorsignal zur Weitertaktung des Zuges aus. Dieses Signal wird an den FB-Eingang "NextInfeedStep" angelegt.

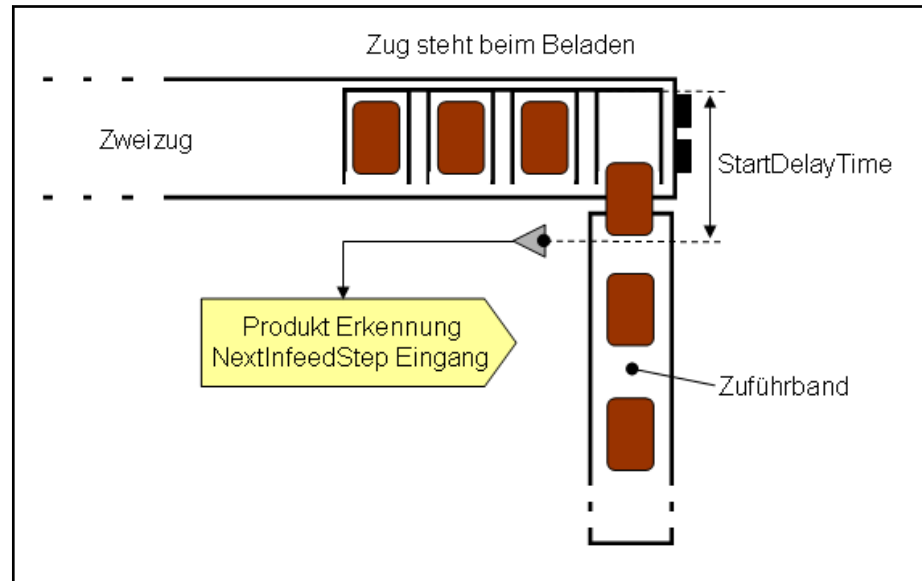


Abb.6-110: Getaktetes Produktzuführen/Beladen im Stillstand von Vorne

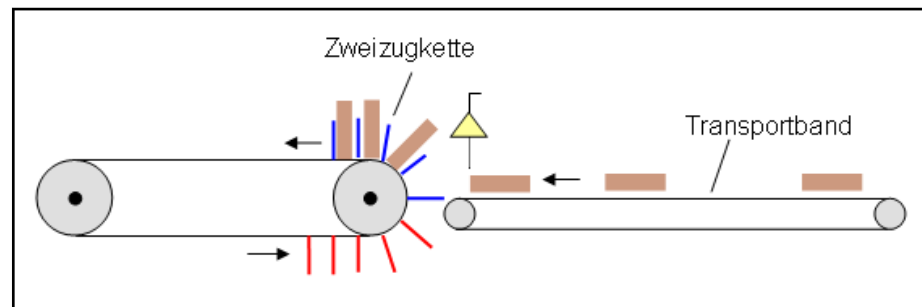


Abb.6-111: Getaktetes Produktzuführen/Beladen im Stillstand seitlich

Über die einstellbare Zeit "AutoSettings.Infeed.StartDelayTime" kann der Transportweg vom Erkennen des Sensorsignales bis zum Start der Weitertakt-Bewegung eingestellt werden. Nach Abschluss der Weitertaktbewegung wird der FB-Ausgang "InfeedStepDone" zur Quittierung gesetzt. Die Fächerbreite sowie die Produkte pro Tasche kann von Tasche zu Tasche variieren und wird über das Strukturelement "MB_MBELT_MODEAUTO.Infeed.InfeedStepData" vorgegeben. Die Eintaktbewegung wird hierbei durch folgenden Strukturelemente festgelegt (siehe auch nachfolgende Abbildung):

- "MB_MBELT_MODEAUTO.Infeed.MotionLaw" spezifiziert das genutzt Bewegungsgesetz

- "MB_MBELT_MODEAUTO.Infeed.StartDelayTime" spezifiziert die Zeitverzögerung zum Start der Infeed Bewegung
- "MB_MBELT_MODEAUTO.Infeed.InfeedLoadTime" spezifiziert die Zeit die der Zug im Stillstand stehen soll (und dabei beladen wird)
- "MB_MBELT_MODEAUTO.Infeed.MaxProductRate" spezifiziert die maximale Anzahl von Produkten pro Minute

Das ausgewählte Bewegungsgesetz "MB_MBELT_MODEAUTO.Infeed.MotionLaw" hat große Bedeutung auf die ausgeführte Infeed-Bewegung. Hierbei ist folgendes zu beachten:

- Grundsätzlich sollten möglichst "VELOCITY_IN_VELOCITY..."-Bewegungsprofile verwendet werden, da diese auch eine Zug-Weitertaktung unter Geschwindigkeit erlauben (wenn z.B. der vorausgegangene Infeed Step noch nicht vollständig abgeschlossen ist). In Sonderfällen können auch alle "Rast in Rast"-Flex Profile Bewegungsgesetze verwendet werden, wenn sichergestellt ist, dass der zu beladene Zug nach jedem Infeed Step den Stillstand erreicht. Empfohlene Infeed Bewegungsgesetze sind:
 - Modifizierte Sinoide (ruckarm aber hohe Beschleunigungsspitze) ("VELOCITY_IN_VELOCITY_MOD_SINE")
 - Modifiziertes Beschleunigungstrapez (stark ruckbehaftet aber geringste Beschleunigungsspitze) ("X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM")
 - Mit dem Bewegungsgesetz "X_FIT_ACC_TRAPEZE_JLIM" wird eine ruckbegrenzte Bewegung ausgeführt. Bei diesem Bewegungsgesetz wirkt der Eingang "MB_MBELT_MODEAUTO.Infeed.MaxProductRate" nicht, da die a, v, j Maximalwerte verwendet werden



Der Abstand zwischen dem Produktsensor und der Beladestation sollte kleiner als eine Produktlänge sein. Innerhalb der "StartDelayTime" werden alle weiteren positive Flanken an "NextInfeedStep" ausgeblendet.

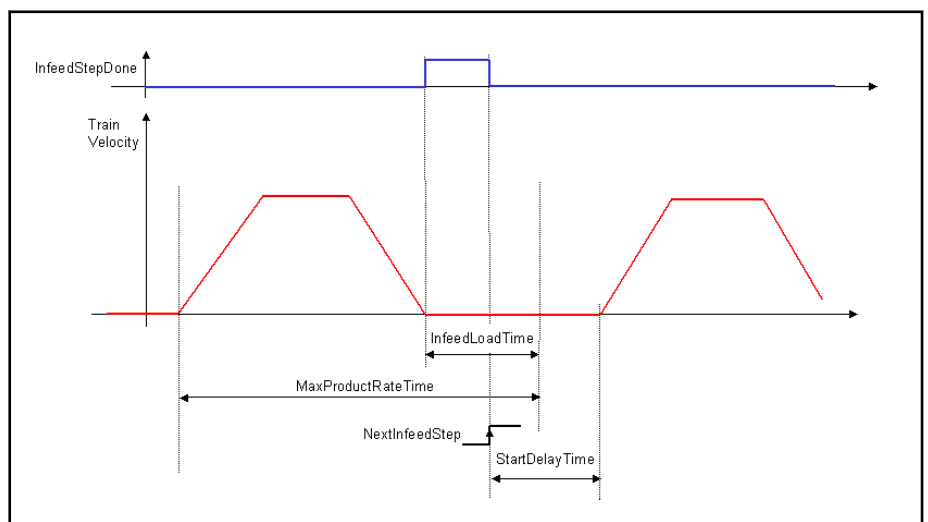


Abb.6-112: Zusammenhang und Wirkungsweise von InfeedLoadTime, StartDelayTime und MaxProductRate

Unter Vorgabe entsprechender Werte für "MB_MBELT_MODEAUTO.Infeed.StartDelayTime" kann auch erreicht werden, dass sich der Zug bei dem

Beladevorgang bewegt. Damit können auch Anwendungen umgesetzt werden bei denen der Zug unter Bewegung beladen wird (siehe nachfolgende Abbildung).

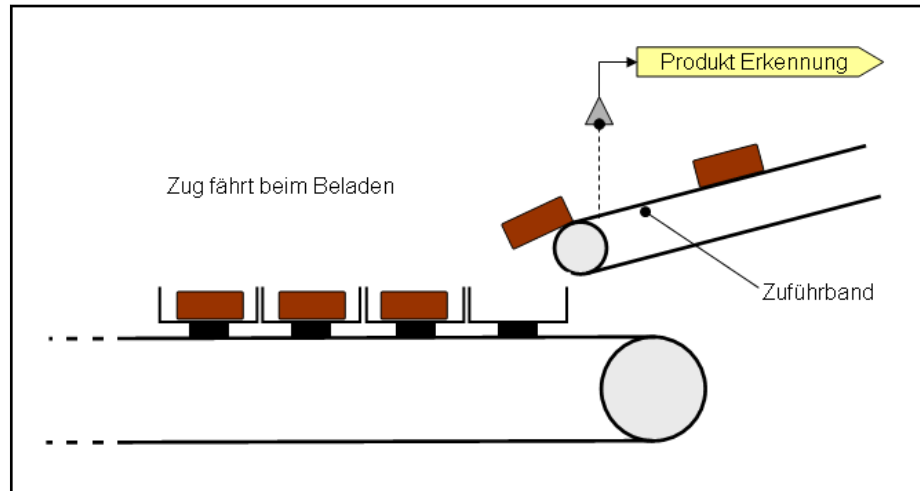


Abb.6-113: Getaktetes Produktzuführen/Beladen bei fahrendem Zug



Typischerweise wird bei der Beladevariante "FEED_INDEXED" der Produktstrom über den Eingang "NextInfeedStep" dem System vorgegeben. Der Zug muss dem vorgegebenen Produkt-Takt folgen. Bei diesen Anwendungen muss typischerweise immer ein Zug am Infeed zur Verfügung stehen um die Produkte aufzunehmen.

Es gibt jedoch auch Anwendungsfälle bei denen die Produkte aufgestaut zur Verfügung stehen, und der Zug die Produkte "anfordert" (Beispiel: Schachteltransport-Applikationen). Der MagicBelt FB gibt dabei die Taktrate über den Ausgang "InfeedStepDone" vor. Bei diesen Anwendungen muss nicht zwangsläufig ein Zug an der Beladeposition stehen. Auch diese Anwendungen können mit dem MagicBelt FB umgesetzt werden. Für diese Anwendungsfälle ist folgende Vorgehensweise zu empfehlen:

Ist die UND-Verknüpfung "InfeedStepDone" UND "Ready" = TRUE, so steht ein Zug am Infeed und fordert ein Produkt an. Das Produkt (z.B. Karton) sollte in das Zug-Fach eingetaktet werden. Ist die Beladung abgeschlossen, so kann mit einer positiven Flanke an "NextInfeedStep" der Zug weitergetaktet werden. Ist die Beladezeit nahezu konstant, so kann die InfeedLoadTime entsprechend eingestellt werden, und der Eingang NextInfeedStep wie folgt verdrahtet werden:

$\text{NextInfeedStep} := \text{Ready AND InfeedStepDone AND ProductsAvailable}$

Mit dieser Beschaltung taktet der Zug automatisch weiter, solange aufgestaute Produkte ($\text{ProductsAvailable}=\text{TRUE}$) zur Verfügung stehen. Das Signal "ProductsAvailable" wird im Anwendungsprogramm berechnet.

Infeed mit konstanter Geschwindigkeit

Grundsätzlich erlauben alle "VELOCITY_IN_VELOCITY..." Infeed Bewegungsgesetze (wie z.B. X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM) ein

Weitertakten in den nächsten "Infeed Step" wenn sich der Zug noch durch den vorausgegangenen "Infeed Step" bewegt (d.h. Geschwindigkeit > 0).

Werden die positiven Flanken an dem Eingang "NextInfeedStep" in kürzeren Zeitintervallen als der maximale Produktrate "MB_MBELT_INFEED.MaxProductRate" vorgegeben, so erreicht der Zug den Stillstand nicht mehr (Voraussetzung: MB_MBELT_INFEED.InfeedLoadTime = 0). Bei immer kürzeren Zeitintervallen des "NextInfeedStep" werden hierbei die Geschwindigkeitseinbrüche immer kleiner. Wenn die positiven Flanken an dem Eingang "NextInfeedStep" mit doppelter Produktrate "MB_MBELT_INFEED.MaxProductRate" (oder höher) angelegt werden, so fährt der Zug mit konstanter Geschwindigkeit.

Beispiel:

MB_MBELT_MODEAUTO.Infeed.MaxProductRate = 60 Produkte/Min --> 1 Produkt entspricht 1s --> Der Zug am Infeed bewegt sich mit konstanter Geschwindigkeit wenn alle 0,5s eine positive Flanke an "NextInfeedStep" angelegt wird.

Über das Status Strukturelement "MB_MBELT_STATUS.MeasuredInfeedStepRate" kann die Frequenz der positiven Flanken an dem Eingang "NextInfeedStep" für Diagnosezwecke überprüft werden. Im oben genannten Beispiel wird dabei NextInfeedStep = 120 angezeigt. Der zu beladene Zug bewegt sich dann mit einer konstanten Geschwindigkeit die 120 Produkten/Minute entspricht.

Synchrones Beladen

Bei dem synchronisieren Beladen wirken folgende Elemente der Datenstruktur MB_MBELT_MODEAUTO.Infeed:

- ProductBuffer
- ProductExpectationWindowStart
- ProductExpectationWindowEnd
- AdditionalSteps
- MinDistanceToOutfeed

Beim synchronen Beladen ist der Zug während dem Beladevorgang mit der Produktzuführung synchronisiert. Die Produktzuführung wird in diesem Fall über ein kontinuierlich drehendes Eintaktrad ("MB_MBELT_GENERAL.InfeedMaster") durchgeführt, welches sich mit jedem Produkt um einen Masterzyklus (= 360°) weiter bewegt. Ein Produktsensor ist auf den Eingang "NextInfeedStep" verdrahtet, und erkennt ob ein Produkt im Fach des Eintaktrades enthalten ist. Die Produkterkennung ist aktiviert, wenn sich das Produkt im aufgespannten Erwartungsfenster (ProdExpectationWindowStart ... ProdExpectationWindowEnd) im Bezug zur Position der InfeedMasters befindet. Das erkannte Produkt wird in ein Schieberegister eingetragen, welches mit jedem Masterzyklus weitertaktet. Der Zug fährt mit jedem Masterzyklus eine Fachlänge weiter, wenn im Eintaktrad-Fach ein Produkt enthalten ist. Falls kein Produkt enthalten ist, wird der Zug entsprechend aussetzen, damit kein Leerfach entsteht. Bei einem Leerfach kann der Zug nicht genau auf der "InfeedPosition" stehen bleiben, da noch der Bremsweg zurückgelegt werden muss. Der Zug bleibt dann ein halbes Fach später stehen (taktsynchrones Aussetzen). Zwischen Produkterkennung und Zugbeladung können mehrere Produkte enthalten sein. Die Anzahl dieser Produkte wird über "MB_MBELT_MODEAUTO.Infeed.ProductBuffer" dem FB vorgegeben. Auch beim synchronen Beladen sind unterschiedliche Fächergrößen einstellbar, es wird jedoch immer fest mit einem Produkt pro Fach ausgegangen. Der optional einstellbare Zugabstand wird mit dem letzten Infeed-Schritt auf die Fächerbreite aufaddiert um diesen zu kompensieren.



Im Sonderfall "MB_MBELT_MODEAUTO.INFEED.PRODUCT-BUFFER" = 0 befindet sich die Produkterkennung direkt vor der Zugbeladung. In diesem Fall wird im gleichem Masterzyklus das Produkt erfasst und das notwendige Bewegungsgesetz für den nächsten Masterzyklus vorbereitet. Die Vorbereitung des Bewegungsgesetzes für den nächsten Masterzyklus benötigt 3 SERCOS-Zyklen. Das Ende des Erwartungsfenster "MB_MBELT_INFFED.ProdExpectationWindowEnd" darf damit nicht größer als der berechnete Ausgangswert "MB_MBELT_STATUS.MaxProdExpectationWindowEnd" werden. Mit gegebener SERCOS-Zykluszeit und maximaler Geschwindigkeit der InfeedMaster-Achse kann dieser Wert auch extern mit der nachfolgenden Formel berechnet werden:

$$\varphi_{MaxProdWindowEnd} [^\circ] = 360^\circ - 3 \cdot t_{SercosCycle} [ms] \cdot v_{InfeedMaster} [min^{-1}] \cdot 0,006$$

Abb.6-114: Berechnung MB_MBELT_STATUS.MaxProdExpectationWindowEnd

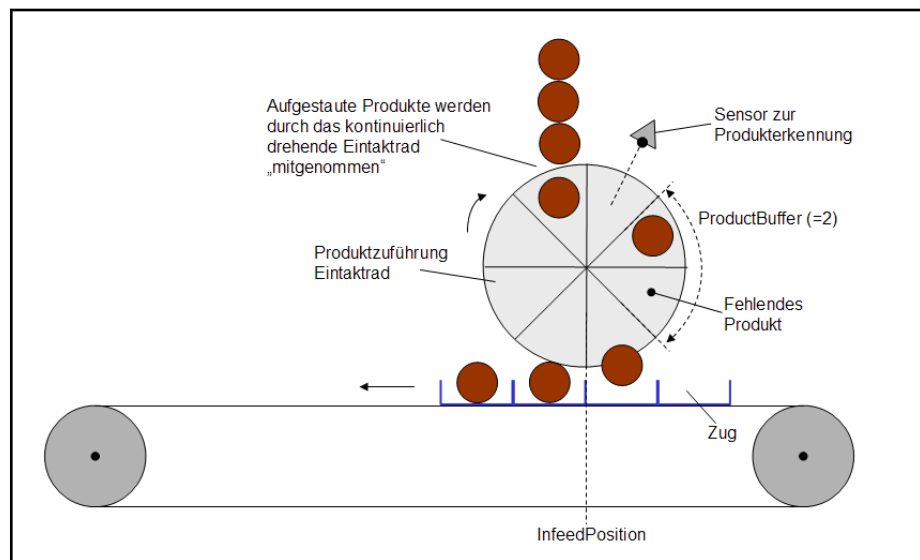
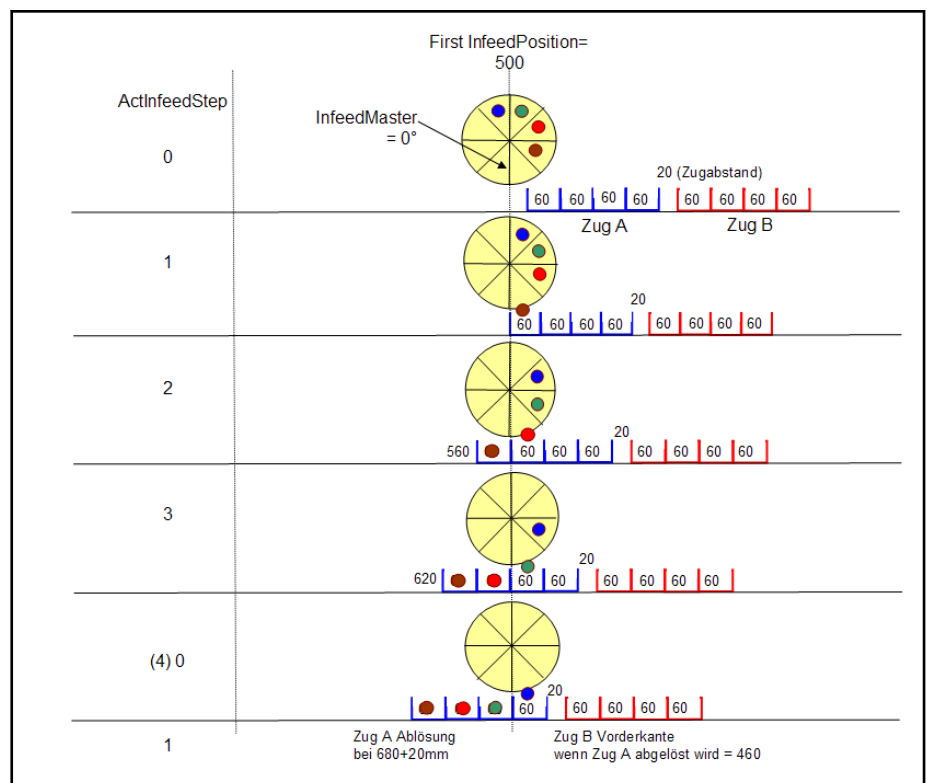


Abb.6-115: Synchrone Produktzuführung

Beispiel für die synchrone Produktzuführung (zeitlicher Ablauf):



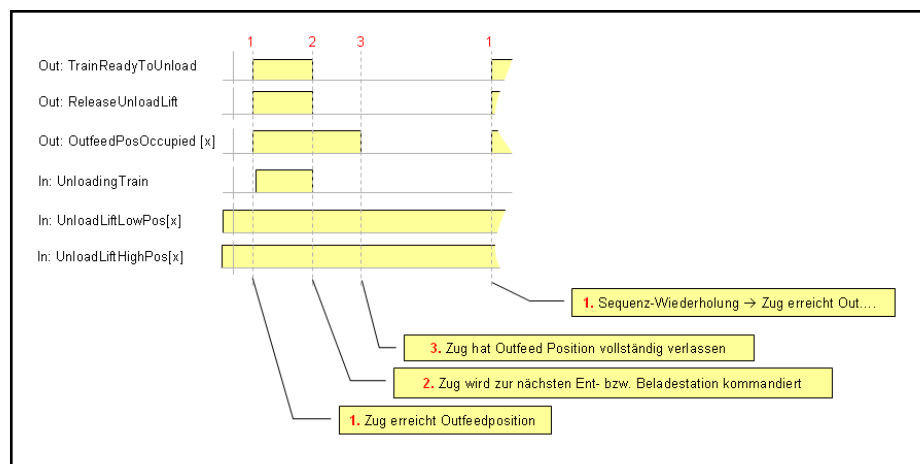


Abb.6-117: Signal Zeit Diagramm beim Entladen ohne Unload-Handshake

Wird der Zug über ein Entlade-Rechen entladen, so müssen die Bewegung des Zuges und des Entlade-Rechens verriegelt werden um Kollisionen zwischen Zug und Rechen zu vermeiden. Der Entlade-Handshake wird über das Strukturelement "MB_MBELT_MODEAUTO.Outfeed.EnableUnloadHandshake" aktiviert.

Im einfachsten Fall wird der Zug erst bei eingezogenem Entlade-Rechen zur nächsten Ent- bzw. Beladestation weitergetaktet. Der entsprechende Handshake und das Signal-Zeit Diagramm zu diesem Fall sind in der nachfolgenden Abbildung dargestellt.

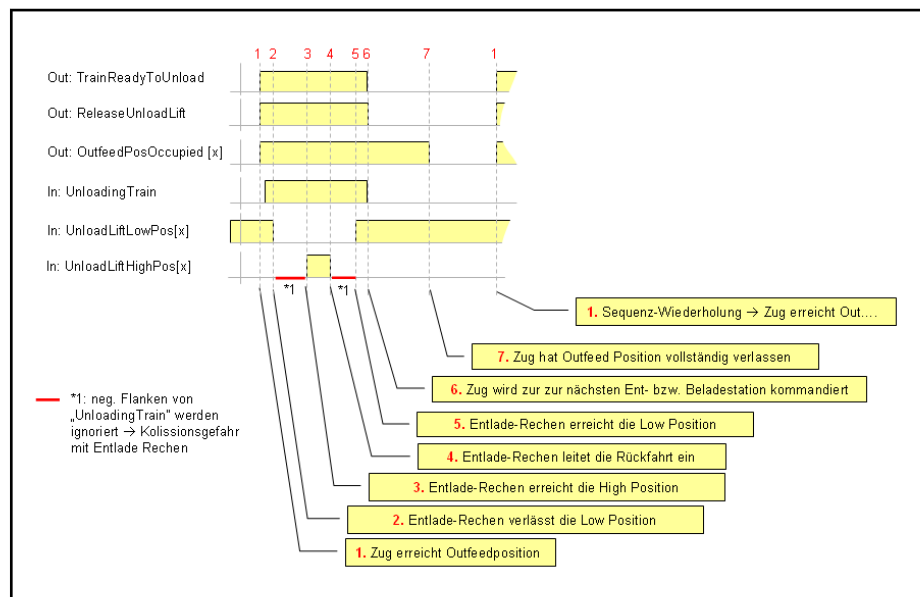


Abb.6-118: Signal Zeit Diagramm beim Entladen mit Handshake/Zug-Weiterfahrt bei eingefahrenem Entlade-Rechen

In einem weiteren Anwendungsfall wird der Zug schon bei vollständig ausgefahrenem Entlade-Rechen weitergetaktet. Der Entlade-Rechen darf dann erst wieder zurückfahren wenn der Zug vollständig aus dem Durchdringungsbe-
reich des Rechens gefahren ist. Die Entladeposition ist solange belegt bis der Entlade-Rechen wieder vollständig zurückgefahren ist. Der entsprechende Handshake und das Signal-Zeit-Diagramm zu diesem Fall sind in der nachfolgenden Abbildung dargestellt.

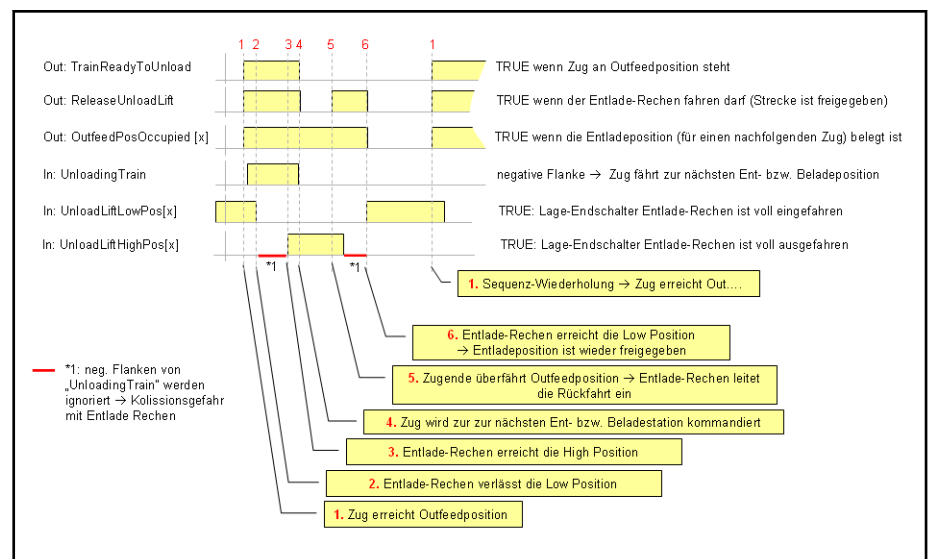


Abb.6-119: Signal-Zeit-Diagramm beim Entladen mit Handshake/Zug-Weiterfahrt bei ausgefahrenem Entlade-Rechen

Stopp Über eine positive Flanke an dem "Stop"-Eingang wird der Zug gemäß spezifizierter Stopp-Variante stillgesetzt ("MB_MBELT_MODEAUTO. StopType"). Bei der synchronen Infeed Variante wird empfohlen, das Anhalten der Ketten durch Verzögern des InfeedMasters durchzuführen und anschließend die Ketten über den StopType = STOP_IMMEDIATE stillzusetzen.

STOP_IMMEDIATE:

Bei dieser Stoppvariante werden alle Züge und der InfeedAxis sofort über die Rampe "MB_MBELT_GENERAL.StopRamp" verzögert (z.B. für schwerwiegende Störungen)

STOP_CYCLE_COMPLETE:

Bei dieser Stoppvariante wird der aktive Produkt-Zyklus (Infeed Step) noch ausgeführt. Danach wird angehalten (definiertes Stillsetzen). Alle Züge die nicht mit der InfeedAxis synchronisiert sind (z.B. bei Fahrten zwischen Outfeedpositionen) werden sofort gestoppt.

STOP_IMMEDIATE_SYNC:

Diese Stoppvariante ist nur für Anwendungen mit synchronisiertem Infeed sinnvoll. Die Stoppvariante verhält sich wie STOP_IMMEDIATE mit dem Unterschied dass die Achsen, die im Synchronverbund mit dem InfeedMaster stehen, weiter synchronisiert bleiben. Diese Achsen werden durch das Verzögern des externen InfeedMasters mit verzögert. Das Verzögern des InfeedMasters muss außerhalb des FB sichergestellt werden.

Fehlerreaktion

Im Fehlerfall werden alle Achsen über die "MB_MBELT_GENERAL.StopRamp" sofort angehalten, sofern sie noch den Sollwerten folgen können (Verhalten wie bei „STOP_IMMEDIATE“). Weiterhin wird der "Error"-Ausgang gesetzt, und Fehlercodes an den entsprechenden FB Ausgängen ausgegeben. Die Fehlerquittierung kann durch steigende Flanke an dem "ClearError"-Eingang erfolgen. Je nach gewähltem StartTyp kann nach Löschen des Fehlers der Automatik Betrieb an der unterbrochenen Stelle ("MB_MBELT_STARTTYPE" = START_AT_LAST_STEP), oder von der definierten Grundstellung ("MB_MBELT_STARTTYPE" = START_AT_BEGIN) fortgesetzt werden.

Produktstau Überwachung

Durch unplanmäßige Verzögerungen (z.B. beim Entladen) können Situationen entstehen bei denen der kontinuierliche Produktstrom nicht mehr aufgenommen werden kann, und folglich ein Produktstau entstehen kann. Solche

ML_TechMotion.library

Situationen müssen möglichst frühzeitig erkannt werden, um den Produktstrom zu drosseln oder zu unterbrechen.

Im ersten Zug (Zug A) ist daher eine Produktstauüberwachung aktiv, die den "Ready"-Ausgang setzt wenn folgende Bedingung eintritt:

- Es befindet sich ein Zug an der Beladestation

UND

Für den Fall **ohne** Entladevorrichtung ("MB_MBELT_INFEED.EnableUnload-Handshake" = FALSE):

- Distanz zwischen gerade beladenem Zug (Zug in "Infeed"-Zustand) und weiteren Zügen an den "Outfeed"-Positionen ist größer als "MB_MBELT_INFEED.MinDistanceToOutfeed"

Für den Fall **mit** Entladevorrichtung ("MB_MBELT_INFEED.EnableUnload-Handshake" = TRUE):

- Distanz zwischen gerade beladenem Zug (Zug in "Infeed"-Zustand) und als belegt gekennzeichneten "Outfeed"-Positionen (siehe "MB_MBELT_STATUS.OutfeedPosOccupied") ist größer als "MB_MBELT_INFEED.MinDistanceToOutfeed". Ist eine "Outfeed"-Position belegt, so wird eine feste Schranke an die Position ("Outfeed"-Position – Zuglänge) gelegt. Solange der Zug im "Infeed"-Zustand mit der Distanz "MB_MBELT_INFEED.MinDistanceToOutfeed" von dieser Schranke entfernt ist, wird der "Ready"-Ausgang gesetzt.

Sind die Bedingungen nicht erfüllt, so wird der "Ready"-Ausgang zurückgesetzt. Bei zurückgesetztem "Ready"-Ausgang sollte die vorgeschaltete Station den Produktstrom unterbrechen. (z.B. Produkte ausblasen wenn "Ready" = FALSE)

Bleibt der Produktstrom auch bei "Ready"=FALSE anstehen, so wird überwacht ob der Zug im "Infeed"-Zustand mit dem Zug an der/den belegten "Outfeed"-Position/Positionen kollidieren würde. Hierzu wird die "Stopp"-Rampe mit aktueller Zug-Geschwindigkeit und Verzögerung ("MB_MBELT_GENERAL.StopRamp") genutzt. Wird eine Kollision erkannt, so wird der betroffene Zug mit der "Stopp"-Rampe sofort stillgesetzt und ein Fehler ausgegeben. Bei der Ermittlung der Zug-Distanzen wird der Fall mit/ ohne Entladevorrichtung unterschieden:

- **Ohne** Entladevorrichtung ("MB_MBELT_INFEED.EnableUnloadHandshake" = FALSE) wird die Distanz zwischen gerade beladenem Zug (Zug in "Infeed"-Zustand) und weiteren Zügen an den "Outfeed"-Positionen überwacht
- **Mit** Entladevorrichtung wird die Distanz zwischen gerade beladenem Zug (Zug in "Infeed"-Zustand) und der als belegt gekennzeichneten "Outfeed"-Positionen überwacht. Ist eine "Outfeed"-Position belegt, so wird eine feste Schranke an die Position ("Outfeed"-Position – Zuglänge) gelegt. Würde der Infeed-Zug mit seiner Verzögerungsrampe "MB_MBELT_GENERAL.StopRamp" über diese Schranke hinausfahren, so wird eine Kollision erkannt und behandelt

Kollisionsüberwachung

Die Kollisionsüberwachung wird automatisch nach Start des FB aktiviert und läuft permanent zyklisch. Die Kollisionsüberwachung ist zweistufig aufgebaut:

In Betriebszuständen mit großen Abständen zum Vorderzug (wie z.B. Fahren auf freier Strecke) muss der Zugabstand größer als der Bremsweg auf den Vorderzug + Sicherheitsabstand MB_MBELT_GENERAL.SafetyDistance sein. Die Kollisionsüberwachung löst aus, wenn diese Bedingung nicht erfüllt ist.

In Betriebszuständen mit sehr geringen Abständen zum Vorderzug (wie z.B. das Aufschließen auf einen Vorderzug) muss der Zugabstand größer als der Sicherheitsabstand MB_MBELT_GENERAL.SafetyDistance sein. Die Kollisionsüberwachung löst aus, wenn diese Bedingung nicht erfüllt ist.



Die Kollisionsüberwachung basiert systembedingt auf Lagesollwerten. Ein Überspringen der Züge durch ungünstig parametrisierte Lageregler kann zu Kollisionen führen.

Die Kollisionsüberwachung kann durch den Eingang **DisableCollisionDetection** = TRUE deaktiviert werden. Diese Funktion sollte jedoch ausschließlich zum auseinander fahren der Züge nach einer Kollision verwendet werden. Sobald die Züge auseinander gefahren sind, ist die Kollisionsüberwachung wieder zu aktivieren.

Dynamiküberwachung

Die "MagicBelt" Applikation ist so einzustellen, dass die Bewegungsdynamik (z.B. Beschleunigung, Geschwindigkeit) beim Fahren zwischen "In- und Outfeed"-Positionen mit größeren oder gleichen Werten im Bezug zur "In- bzw. Outfeed"-Bewegung eingestellt ist. Nur so kann sichergestellt werden, dass ein Zug der eine "In- bzw. Outfeed"-Bewegung ausführt nicht mit einem langsameren Vorderzug kollidiert. Der FB generiert einen Fehler und verzögert alle Achsen mit der Rampe "MB_MBELT_GENERAL.StopRamp", falls die Dynamikdaten nicht den o.g. Regeln entsprechen.

Entladeüberwachung

Bei der Produktentnahme mit Greifer können im Fehlerfall Produkte in dem Zug verbleiben. Die Erkennung der unvollständigen Produktentnahme muss - bei Bedarf- außerhalb des "MagicBelt"-FB erfolgen. Wird ein solcher Fall extern erkannt, können über den Eingang "Stop" die Ketten stillgesetzt werden. Nach positiver Flanke von "Start" wird der Betrieb wieder fortgesetzt.

Fehlerbehandlung

Im Fehlerfall werden grundsätzlich alle Züge mit Rampe stillgesetzt. Es wird keine Unterscheidung getroffen welcher Zug den Fehler verursachte.

Wiederanlauf

Nach Beseitigung der Fehlerursache kann der Fehler durch positive Flanke an dem Eingang **ClearError** gelöscht werden. Über eine positive Flanke an "Start" starten die Züge gemäß vorgegebener Startoption. Mit der Startoption "START_AT_LAST_STEP" kann z.B. der Zustand vor der Unterbrechung wiederhergestellt werden, und z.B. ein noch beladener Zug wieder weiter auf die Entladeposition gefahren werden.

Fehlercodes

Die folgenden Fehlercodes der "F_RELATED_TABLE" werden von dem Funktionsbaustein generiert:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#0001	16#0000	Keine Leitung am Kettenantrieb zugeschaltet
RESOURCE_ERROR	16#0003	16#0000	FB wurde von einem anderen FB unterbrochen
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Zustand der internen Zustandsmaschine
RESOURCE_ERROR	16#0011	16#0000	Steuerung ist nicht betriebsbereit
INPUT_INVALID_ERROR	16#1A00	16#0001	Infeed positioning time <= 0ms
INPUT_INVALID_ERROR	16#1A00	16#0002	Products per pocket = 0
INPUT_INVALID_ERROR	16#1A00	16#0003	Feedrate override <= 0 OR > 100
INPUT_INVALID_ERROR	16#1A00	16#0004	CtrlChain <= 0 OR > 3
INPUT_INVALID_ERROR	16#1A00	16#0005	ChainLength <= 0

ML_TechMotion.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#1A00	16#0006	NoOfChains <= 0 OR > 3
INPUT_INVALID_ERROR	16#1A00	16#0007	TrainsPerChain <= 0 OR > 3
INPUT_INVALID_ERROR	16#1A00	16#0008	ProductBuffer <= 0
INPUT_INVALID_ERROR	16#1A00	16#0009	InfeedAxis AxisRef is invalid
INPUT_INVALID_ERROR	16#1A00	16#000A	Chain AxisRef is invalid
INPUT_INVALID_ERROR	16#1A00	16#000B	StopRamp zu klein. StopRamp muss größer oder gleich als alle anderen vorgegebenen Beschleunigungen sein
INPUT_INVALID_ERROR	16#1A00	16#000C	InfeedType is ungültig
INPUT_INVALID_ERROR	16#1A00	16#000D	FirstInfeedPosition ist außerhalb gültigen Bereich
INPUT_INVALID_ERROR	16#1A00	16#000E	BeginReducedVel1 ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#000F	EndReducedVel1 ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0010	BeginReducedVel2 ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0011	EndReducedVel2 ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0012	MaxProductRate ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0013	MinDistanceToOutfeed ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0014	MaxInfeedVelocity ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0015	MaxInfeedAcceleration ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0016	MaxInfeedJerk ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0017	StartDelayTime ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0018	InfeedLoadTime ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0019	Infeed motion law ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#001A	ProdExpectationWindowStart ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#001B	ProdExpectationWindowEnd ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#001C	AdditionalSteps ist außerhalb des gültigen Bereiches

ML_TechMotion.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#1A00	16#001D	InfeedStepData ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#001E	FirstOutfeedPosition ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#001F	MaxOutfeedVelocity ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0020	MaxOutfeedAcceleration ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0021	MaxOutfeedJerk ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0022	OutfeedDistances ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0023	MaxMoveVelocity ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0024	MaxMoveAcceleration ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0025	MaxMoveJerk ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0026	LoadTrainAcceleration ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0027	ReducedVelocity ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0028	JogAxisSelection ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0029	JogVelocity ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002A	JogAcceleration ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002B	JogDistance ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002C	TrainDistance ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002D	SafetyDistance ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002E	Mode Eingang außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002F	Erwartungsfenster Ende "ProdExpectationWindowEnd" ist zu groß
RESOURCE_ERROR	16#1A01	16#0000	Ein oder mehrere Kettenantriebe sind nicht referenziert
RESOURCE_ERROR	16#1A02	16#0000	Outfeed Position ist gesperrt
SYSTEM_ERROR	16#1A03	16#0000	Kollision mit Vorderzug erkannt

ML_TechMotion.library

ErrorID	Additional1	Additional2	Beschreibung
SYSTEM_ERROR	16#1A04	16#0000	Infeed Beschleunigung ist größer als Move Beschleunigung
RESOURCE_ERROR	16#1A05	16#0000	Fallende Flanke an TrainUnload aber Unload Lift ist nicht in eingezogen oder ausgefahren
RESOURCE_ERROR	16#1A06	16#0000	Modulowert des Kettenantriebes oder Infeed Masters sind nicht richtig eingestellt
RESOURCE_ERROR	16#1A07	16#0000	Folgefehler verursacht durch voreilende Kette
SYSTEM_ERROR	16#1A08	16#xxxx	Unerwarteter Zustand einer anderen Kette xxxx: Kettennummer
SYSTEM_ERROR	16#1A09	16#xxxx	Ungültiger interner Array Index xxxx: Array Index
RESOURCE_ERROR	16#1A0A	16#xxxx	UnloadLiftLowPos UND UnloadLiftHighPos sind zum gleichen Zeitpunkt gesetzt xxxx = Outfeed index
RESOURCE_ERROR	16#1A0B	16#xxxx	Resultierender Fehler verursacht durch eine andere Kette xxxx = Ketten Nummer mit der Fehler Ursache
RESOURCE_ERROR	16#1A0C	16#xxxx	Repositionierung ist erforderlich (START_AT_LAST_STEP) aber Entladevorrichtung ist blockiert xxxx = Ketten Nummer mit der Fehler Ursache
SYSTEM_ERROR	16#1A0D	16#0000	Ungültige interne Schrittweite
SYSTEM_ERROR	16#1A0E	16#0000	Intern berechneter Kollisions-Zug ist ungültig
SYSTEM_ERROR	16#1A0F	16#0000	Ein nächster Infeed Schritt wurde getriggert, der vorausgegangene Infeed Schritt war jedoch nicht abgeschlossen, und ein REST_IN_REST Infeed Bewegungsgesetzt wurde ausgewählt
RESOURCE_ERROR	16#1A10	16#0001	Positionswichtung des Kettenantriebes ist unzulässig
RESOURCE_ERROR	16#1A10	16#0002	Geschwindigkeitswichtung des Kettenantriebes ist unzulässig
RESOURCE_ERROR	16#1A10	16#0003	Beschleunigungswichtung des Kettenantriebes ist unzulässig
RESOURCE_ERROR	16#1A10	16#0004	Positionswichtung der InfeedAxis ist unzulässig

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#1A10	16#0005	Geschwindigkeitswichtung der InfeedAxis ist unzulässig
RESOURCE_ERROR	16#1A10	16#0006	Beschleunigungswichtung der InfeedAxis ist unzulässig
RESOURCE_ERROR	16#1A10	16#0007	Positionswichtung des InfeedMaster ist unzulässig
RESOURCE_ERROR	16#1A10	16#0008	Geschwindigkeitswichtung des InfeedMaster ist unzulässig
RESOURCE_ERROR	16#1A10	16#0009	Beschleunigungswichtung des InfeedMaster ist unzulässig

Abb.6-120: Fehlercodes Funktionsbaustein MB_MagicBeltType01

Inbetriebnahmehinweise

Die folgenden Punkte müssen zur Inbetriebnahme des Funktionsbausteines "MB_MagicBeltType01" berücksichtigt werden:

- Pro Kette muss jeweils eine eigene Instanz des Funktionsbausteines aufgerufen werden
- Der Funktionsbaustein "MB_MagicBeltType01" muss in einer **SERCOS-synchronen Task** zyklisch aufgerufen werden. Der Task Watchdog muss eingeschaltet sein (Empfindlichkeit = 1)
- Die Achsen des Kettenantriebs **Chain A, B, (C)** müssen mit IndraWorks angelegt werden, hierbei ist auf die folgende Parametrierung zu achten:
 - Achstyp: real oder virtuell (für Testzwecke)
 - "Interpolation auf Antrieb" bei Hinzufügen des Antriebes **muss ausgewählt werden** (Lagesollwertgenerierung erfolgt auf der Steuerung)
 - Wichtungsart "**Translatorisch, Modulo, Einheit [mm]**" muss eingestellt werden
 - Der Modulowert muss wie folgt eingestellt werden: **Modulowert = ChainLength/TrainsPerChain**
- Die **InfeedAxis** muss mit IndraWorks angelegt werden, hierbei ist auf folgende Parametrierung zu achten:
 - Achstyp: **virtuell**
 - Wichtungsart "**Translatorisch, Modulo, Einheit [mm]**" muss eingestellt werden
 - Der Modulowert muss wie folgt eingestellt werden: **Modulowert = ChainLength/TrainsPerChain**
- Die **DummyMaster** Achse muss in IndraWorks angelegt werden, falls keine andere zur Verfügung steht. Hierbei sind alle Achstypen zugelassen, und es ist keine besondere Parametrierung erforderlich, da diese Achse nur passiv genutzt wird
- Bei synchronem Beladen muss zusätzlich die **InfeedMaster** Achse mit IndraWorks angelegt werden, hierbei ist auf folgende Parametrierung zu achten:
 - Es können grundsätzlich alle Achstypen (Real, Virtuell, Encoder...) verwendet werden
 - Wichtungsart "**translatorisch, rotatorisch**" muss eingestellt werden
 - Modulowert = **360°**

ML_TechMotion.library

- Die in IndraWorks angelegten Achsen müssen dann über die Datenstruktur "MB_MBELT_GENERAL" den entsprechenden Achsen (wie z.B. "ChainAxes") zugeordnet werden (über AxisRef)

6.9 Funktionsbaustein zur SmartBelt-Anwendung

6.9.1 MB_SmartBeltType01

Kurzbeschreibung

Dieser Funktionsbaustein misst die Winkelverschiebung zwischen Produktposition und gewünschter Leitachseposition. Wird eine Winkelverschiebung erkannt, wird eine Winkelkorrektur an der Produktposition mittels eines dreieckigen Geschwindigkeitsprofils vorgenommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library
IndraMotion MLD ab 7VRS mit MA Funktionspaket	MX_Packaging_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Packaging_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Packaging_17.lib (in Vorbereitung)

Abb.6-121: Referenztabelle Funktionsbaustein MB_SmartBeltType01

Schnittstellenbeschreibung

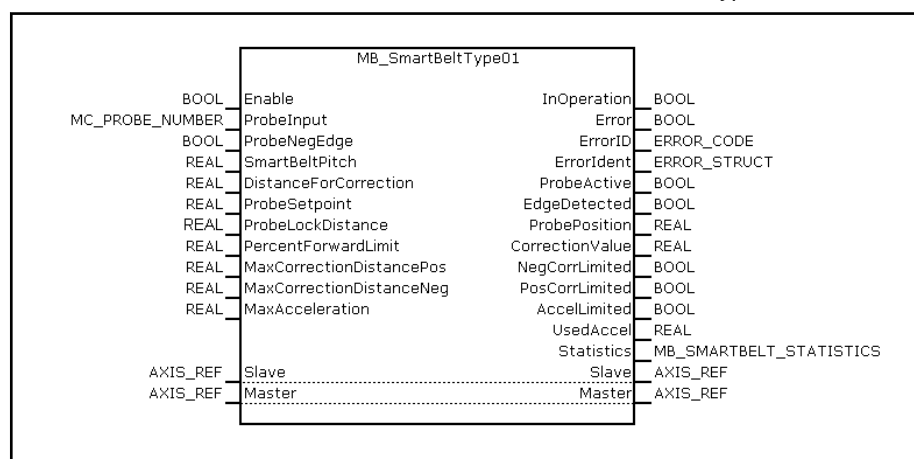


Abb.6-122: Funktionsbaustein MB_SmartBeltType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Slave	AXIS_REF	Adresse der SmartBelt-Achse
	Master	AXIS_REF	Adresse der Leitachse
VAR_INPUT	Enable	BOOL	Steigende Flanke aktiviert Funktionsbaustein
	ProbeInput	MC_PROBE_NUMBER	Auswahl des Messtastereingangs auf Antrieb. Wird bei steigender Flanke am Eingang "Enable" gelesen

E/A-Typ	Name	Datentyp	Beschreibung
	ProbeNegEdge	BOOL	FALSE: Positive Flanke des Messtasters wird ausgewertet TRUE: Negative Flanke des Messtasters wird ausgewertet. Wird bei steigender Flanke am Eingang "Enable" gelesen
	SmartBeltPitch	REAL	Wirksame SmartBelt-Länge (Inch bzw. mm). Wird bei steigender Flanke am Eingang "Enable" gelesen
	DistanceForCorrection	REAL	Verbleibender Abstand für die Korrektur auf dem Band. Dies bezieht sich normalerweise auf die Entfernung, von der der Messtaster am Ende des Bands angebracht ist. Wird bei steigender Flanke am Eingang "Enable" gelesen
	ProbeSetpoint	REAL	Sollwert des Messtasters (Inch oder mm) Wird zyklisch gelesen
	ProbeLockDistance	REAL	Entfernung innerhalb der Messtasterereignisse ignoriert werden (Inch oder mm). Bei Erkennung eines Messtasterereignisses wird das nächste Messtasterereignis erst berücksichtigt, nachdem das Band die Entfernung in "ProbeLockDistance" zurückgelegt hat. Wird zyklisch gelesen
	PercentForwardLimit	REAL	Der prozentuale Anteil des gesamten Produktzyklus, der die Verteilung zwischen Vorwärts- und Rückwärtskorrektur festlegt. Beispiel: Ein Wert von 70,0 würde bis zu 70 mm vorwärts und bis zu 30 mm rückwärts für einen Produktzyklus von 100 mm Länge korrigiert werden. Wird zyklisch gelesen
	MaxCorrectionDistPos	REAL	Maximaler Korrekturweg für positive Korrektur (Inch oder mm). Wird zyklisch gelesen
	MaxCorrectionDistNeg	REAL	Maximaler Korrekturweg für negative Korrektur (Inch oder mm). Wird zyklisch gelesen
	MaxAcceleration	REAL	Maximal zulässige Beschleunigung in Antriebseinheiten (in/s ² , mm/s ²). Wird zyklisch gelesen
VAR_OUTPUT	InOperation	BOOL	SmartBelt in Betrieb
	Error	BOOL	Weist auf einen Fehler hin. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung

ML_TechMotion.library

E/A-Typ	Name	Datentyp	Beschreibung
	ErrorStruct	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	ProbeActive	BOOL	SmartBelt-Messtasterfunktionalität aktiv. Wartet auf Ereignis
	EdgeDetected	BOOL	Messtasterereignis erkannt
	ProbePosition	REAL	Erfasste Position in Antriebseinheiten basierend auf "SmartBeltPitch"
	CorrectionValue	REAL	Korrekturwert in Antriebseinheiten basierend auf "SmartBeltPitch".
	NegCorrLimited	BOOL	Negative Korrektur ist auf den in "MaxCorrectionDistanceNeg" eingestellten Weg begrenzt
	PosCorrLimited	BOOL	Positive Korrektur ist auf den in "MaxCorrectionDistancePos" eingestellten Weg begrenzt
	AccelLimited	BOOL	Korrektur ist auf die in "MaxAcceleration" eingestellte Beschleunigung begrenzt
	UsedAccel	REAL	Berechnete Beschleunigung, die während der Winkelkorrektur verwendet wird (in/s ² oder mm/s ²)
	Statistik	MB_SMART-BELT_STATISTICS	Struktur, die verschiedene statistische Daten enthält

Abb.6-123: Schnittstellenvariablen Funktionsbaustein MB_SmartBeltType01

Struktur des MB_SMART-BELT_STATISTICS

Diese Struktur enthält statistische Daten, die für die Überwachung und Auswertung der Anwendung verwendet werden können. Die einzelnen Elemente sind in der nachfolgenden Tabelle aufgeführt:

Strukturelement	Typ	Default-Wert	Beschreibung
CurRate	REAL	0.0	Produktrate basierend auf dem Abstand zwischen aktuellem und vorherigem Produkt
NbrOnBelt	UINT	0	Aktuelle Produktanzahl auf dem Band. Ein Produkt ist "auf dem Band", wenn es weniger als eine ganze SmartBelt-Länge seit der Registrierung gefahren ist
MaxGap	REAL	0.0	Maximaler Abstand zwischen zwei aufeinanderfolgenden Produkten in Einheiten basierend auf dem "SmartBeltPitch"
MinGap	REAL	"SmartBeltPitch"	Minimaler Abstand zwischen zwei aufeinanderfolgenden Produkten in Einheiten basierend auf dem "SmartBeltPitch"
TotalNbr	UINT	0	Gesamtproduktanzahl, die vom SmartBelt registriert wurde

Abb.6-124: Struktur des MB_SMARTBELT_STATISTICS

Alle Elemente in der Struktur MB_SMARTBELT_STATISTICS werden auf ihren Standardwert bei steigender Flanke am Eingang "Enable" gesetzt.



Die Werte der Elemente können falsch sein, wenn die Produkte nicht richtig registriert wurden.

"CurRate" Wert wird auf der Basis von Zeitmessungen berechnet. Die Zeitauflösung beträgt 1ms. Für hohe Produktmengen pro Zeiteinheit können die berechneten Geschwindigkeiten im Bereich der 1ms-Auflösung variieren. Wird beispielsweise eine Produktrate von 400 Produkten pro Minute verwendet, beträgt die Zeit für ein Produkt $1 / (400 / 60) = 150$ ms. Bei einer Auflösung von 1ms können Werte im Bereich von 149ms, 150ms und 151ms gemessen werden, was in Produktraten von 403, 400 und 397 resultiert.

Die Werte für "MaxGap" und "MinGap" werden auf Basis der zurückgelegten Distanz von der Leitachse berechnet. Dies merkt man besonders auf dem ersten Band, wenn dieses beschleunigt oder abbremst. Der SmartBelt legt in diesem Fall wesentlich mehr/weniger Weg zurück als die Leitachse.

Min/Max- und Standard-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
Enable	BOOL			FALSE
ProbeInput	MC_PROBE_NUMBER	PROBE1	PROBE2	PROBE1
ProbeNegEdge	BOOL	-	-	FALSE
SmartBeltPitch	REAL	>0.0	10000.0	200.0
DistanceForCorrection	REAL	0.0	SmartBeltPitch	100.0
ProbeSetpoint	REAL	0.0	Modulowert der Leitachse	0.0
ProbeLockDistance	REAL	0.0	SmartBeltPitch	0.0
PercentForwardLimit	REAL	0.0	100.0	50.0
MaxCorrectionDistPos	REAL	0.0	SmartBeltPitch	100.0
MaxCorrectionDistNeg	REAL	-SmartBeltPitch	0.0	-100.0
MaxAcceleration	REAL	> 0.0	∞	100
Slave	AXIS_REF	AxisNo >= 1	<= MB_MAX_AXIS_NUMBER	-
Master	AXIS_REF	AxisNo >= 1	<= MB_MAX_AXIS_NUMBER	-

Abb.6-125: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Funktionsbeschreibung

Detaillierte Beschreibung der Eingangs- und Ausgangsvariablen

Die Maßeinheiten müssen für alle Eingänge gleich sein (Inch oder mm). Ist der "SmartBeltPitch" beispielsweise in mm angegeben, muss "DistanceForCorrection" in mm sein und auch "MaxAcceleration" in mm/s².

"PercentForwardLimit" wird verwendet um die Verteilung von Vorwärts- zu Rückwärts-Korrektur zu bestimmen. Wenn das Produkt z.B. nur in Vorwärtsrichtung korrigiert werden soll, muss als Wert 100.0 eingegeben werden. Damit werden all möglichen Positionsfehler durch eine Korrektur nach vorne ausgeglichen. Dieser Wert gibt die prozentuale Verteilung nach vorne, bzw. nach hinten an.

Der maximal gemessene Fehler für ein Produkt kann zwischen $(-0.5 * \text{Produktzyklus})$ und $(+0.5 * \text{Produktzyklus})$ liegen. Der Begriff Produktzyklus ist

ML_TechMotion.library

identisch mit "CamShaftDistance" oder entspricht der Produktlänge plus dem Abstand zwischen den Produkten. Die folgende Abbildung zeigt ein Beispiel:

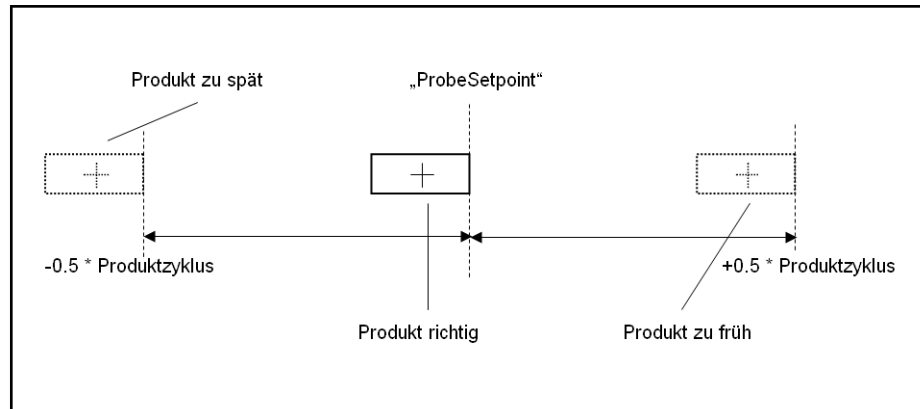


Abb.6-126: Bedeutung der Korrektur in positiver, bzw. negativer Richtung

Das Produkt kann jetzt entweder in positive oder negative Richtung korrigiert werden. Somit ist es möglich, die Länge eines gesamten Produktzyklus zu korrigieren. "PercentForwardLimit" gibt an, wie der gemessene Fehler in Bezug auf die gesamte Produktlänge zu korrigieren ist. Wird "PercentForwardLimit" auf 50% gesetzt, wird die Hälfte der Fehler vorwärts, die andere Hälfte rückwärts korrigiert.

Wird "PercentForwardLimit" auf 70% gesetzt, werden die Korrekturen wie folgt vorgenommen:

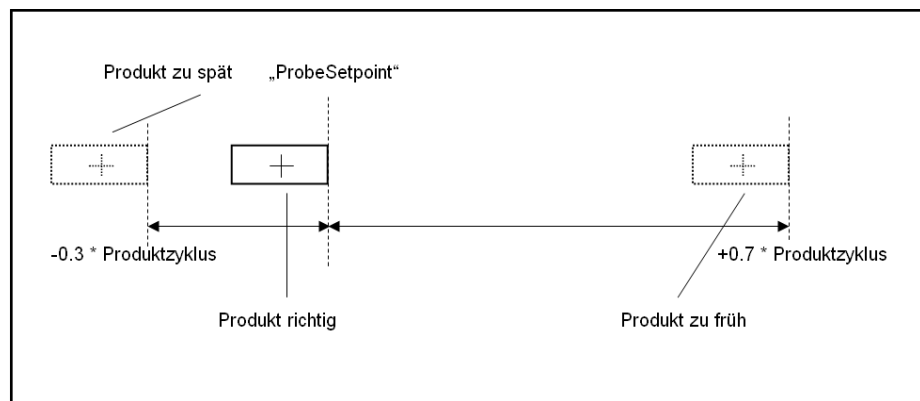


Abb.6-127: PercentForwardLimit

"MaxCorrectionDistancePos" und "MaxCorrectionDistanceNeg" begrenzen den absoluten Weg, der in beide Richtungen kompensiert wird. Mit "PercentForwardLimit" können Kombinationen erstellt werden, mittels derer alle Fehler oder nur Teilfehler in beide Richtungen kompensiert werden können. Wenn beispielsweise 80% aller Fehler in die positive Richtung korrigiert werden sollen (aber nicht mehr als 60 mm in die positive Richtung und nicht mehr als 20 mm in die negative Richtung der Länge eines gesamten Produktzyklus korrigiert werden sollen), werden die Eingänge wie folgt konfiguriert:

"PercentForwardLimit" = 80

"MaxCorrectionDistancePos" = 60

"MaxCorrectionDistanceNeg" = -20

Erläuterung: 80% (160 mm) werden vorwärts und 20% (40 mm) werden rückwärts für einen 200mm langen Produktzyklus mit "PercentForwardLimit" = 80 korrigiert. "MaxCorrectionDistancePos" = 60 begrenzt die vorhandenen 160

mm auf 60 mm und "MaxCorrectionDistanceNeg" = -20 begrenzt die zulässige Rückwärtskorrektur auf -20 mm.

Der Gesamtprozess wird im Folgenden beschrieben:

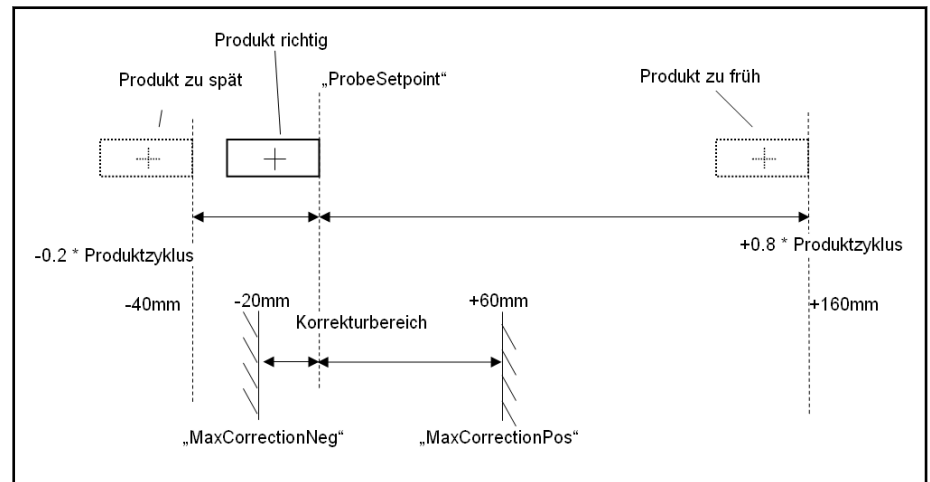


Abb.6-128: MaxCorrectionDistancePos und MaxCorrectionDistanceNeg

Ein Fehler, der sich außerhalb der "MaxCorrectionDistancePos"- und "MaxCorrectionDistanceNeg"-Begrenzungen befindet, wird auf "MaxCorrectionDistancePos" und "MaxCorrectionDistanceNeg" begrenzt. Der verbleibende Fehler muss auf dem nächsten Band korrigiert werden. Damit ist es möglich die Korrekturen auf alle Bänder zu verteilen, so dass das erste Band nicht die Hauptlast der Korrektur tragen muss.

Je länger der Korrekturweg, desto höher die Beschleunigung. Alternativ kann die maximale Beschleunigung auch mittels "MaxAcceleration" begrenzt werden. Die maximale Beschleunigung und der maximale Korrekturabstand sind gleichzeitig aktiv. Die niedrigere der beiden Beschränkungen gilt.

Zeitanforderungen

Der Funktionsbaustein benötigt bis zu 400 SPS-Zyklen um die Initialisierung abzuschließen. Die genaue Zyklenanzahl hängt von der Belastung der Steuerung ab.

Fehlerbehandlung

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0006)	16#0012	16#0000	Achse ohne Leistung
RESOURCE_ERROR (16#0006)	16#0001	x	Achse (Leit- oder Folgeachse) befindet sich nicht an der lokalen Steuerung. "x" gibt die Adresse der fehlerhafte Achse an
ACCESS_ERROR (16#0004)	16#0003	16#0000	Der intern benutzte Funktionsbaustein MB_TouchProbeContinuous wurde durch einen anderen Funktionsbaustein unterbrochen
ACCESS_ERROR (16#0004)	16#0003	16#0001	Der intern benutzte Funktionsbaustein MB_PhasingSlave wurde durch einen anderen Funktionsbaustein unterbrochen

ML_TechMotion.library

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Fehler in Zustandsmaschine
INPUT_INVALID_ERROR (16#0001)	16#000D	x	Achse (Leit- oder Folgeachse) besitzt nicht gültige Adresse. x gibt die Adresse der fehlerhaften Achse an
INPUT_INVALID_ERROR (16#0001)	16#0F02	16#0010	"SmartBeltPitch" außerhalb des gültigen Wertebereichs (0..10000)
INPUT_INVALID_ERROR	16#0F02	16#0011	"ProbeInput" ungültig (PROBE1, PROBE2)
INPUT_INVALID_ERROR	16#0F02	16#0012	"DistanceForCorrection" ungültig (0.." SmartBeltPitch")
INPUT_INVALID_ERROR	16#0F02	16#0013	"MaxCorrectionDistPos" ungültig (0.." SmartBeltPitch")
INPUT_INVALID_ERROR	16#0F02	16#0014	"MaxCorrectionDistNeg" ungültig (-"SmartBeltPitch"..0)
INPUT_INVALID_ERROR	16#0F02	16#0015	Achse ist keine IndraDrive-Achse (Packprofil nicht zulässig)
INPUT_INVALID_ERROR	16#0F02	16#0017	Folgeachswichtung ist nicht die bevorzugte Wichtung
INPUT_INVALID_ERROR	16#0F02	16#0018	Leitachswichtung ist nicht die bevorzugte Wichtung
INPUT_INVALID_ERROR	16#0F02	16#0019	Beschleunigungswert zu niedrig (>0)
INPUT_INVALID_ERROR	16#0F02	16#001A	Folgeachse hat keine Modulowichtung
INPUT_INVALID_ERROR	16#0F02	16#001B	Leitachse hat keine Modulowichtung
INPUT_INVALID_ERROR	16#0F02	16#001C	Einstellungen in P-0-0154 (IndraMotion MLD) / A-0-2793 (IndraMotion MLC) müssen auf "Shortest Path" gesetzt sein
INPUT_INVALID_ERROR	16#0F02	16#001D	"ProbeSetpoint"-Eingang ist außerhalb des gültigen Bereichs (0..Leitachsmodulo)
INPUT_INVALID_ERROR	16#0F02	16#001F	"ProbeLockDistance"-Eingang ist außerhalb des gültigen Bereichs (0.."SmartBeltPitch")
INPUT_INVALID_ERROR	16#0F02	16#0020	"PercentForwardLimit" ist außerhalb des gültigen Bereichs (0..100.0)
INPUT_INVALID_ERROR	16#0F02	16#0021	Modulowert der Folgeachse muss mindestens so groß wie MaxCorrectionDistPos + ABS(MaxCorrectionDistNeg)

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#0F02	16#0022	Modulwert der Folgeachse muss mindestens doppelt so groß wie der "SmartbeltPitch" sein
DEVICE_ERROR (16#0008)	16#0F06	16#0000	Geschwindigkeitswichtung und Wichtung der Lagedaten sind nicht gleich

Abb.6-129: Fehlercodes Funktionsbaustein MB_SmartBeltType01

Zusätzlich kann der Funktionsbaustein alle folgenden Fehler der Unterfunktionsbausteine ausgeben:

- MB_GetCyclicParameterHandle
- MB_PhasingSlave
- MB_ReadParameter
- MB_ReadRealParameter
- MB_ReadStringParameter
- MC_TouchProbeInit
- MC_TouchProbeContinuous

Siehe Dokumentation zu den Fehlercodes des jeweiligen Bausteins.

6.9.2 Einführung in die SmartBelt-Verwendung

SmartBelt-Verwendung

SmartBelts werden verwendet, um die Produkte vor ihrer Bearbeitung korrekt zu verteilen. Produkte werden durch eine Zuführeinheit den SmartBelts, wo der Winkel eingestellt wird, zur Verfügung gestellt. Dann übernimmt eine Bearbeitungseinheit die Produkte (z.B. Schlauchbeutelmaschine).

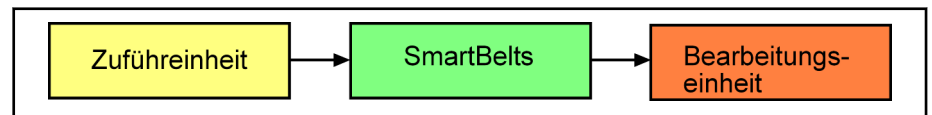


Abb.6-130: Beispiel einer Schlauchbeutelmaschine

Je nach Anwendung sind mehrere Situationen denkbar:

- Zuführeinheiten stellen Produkte mit richtigem Abstand zueinander bereit, aber die Bearbeitungseinheiten können nicht synchron zur Zuführeinheit starten bzw. stoppen. In diesem Fall können SmartBelts dazu benutzt werden, die Abstände zwischen den Produkten zu korrigieren
- Zuführeinheiten stellen Produkte aus dem Storage-on-Demand bereit. SmartBelts werden nicht benötigt
- Zuführeinheiten sorgen für einen zufälligen, aber fast immer gleichen Abstand zwischen den Produkten. SmartBelts können verwendet werden
- Zuführeinheiten sorgen für einen zufälligen Abstand zwischen den Produkten. SmartBelts können zusammen mit den folgenden Maßnahmen verwendet werden:
 - Die Geschwindigkeit der Zuführeinheiten kann gesteuert werden, um eine konstante Produktgeschwindigkeit zu erreichen
 - Die Geschwindigkeit der Bearbeitungseinheiten kann gesteuert werden, um eine konstante Produktgeschwindigkeit zu erreichen

ML_TechMotion.library

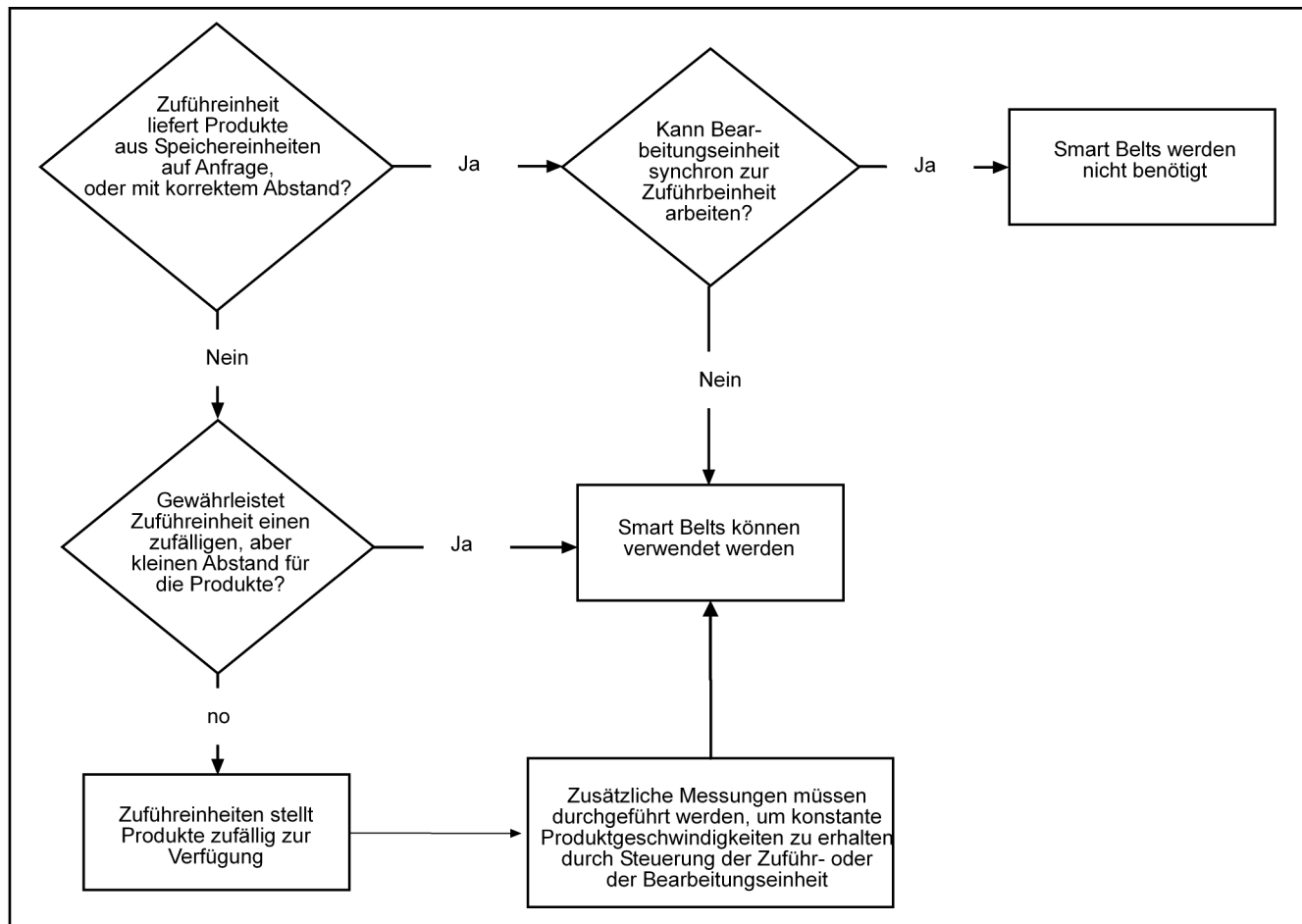


Abb.6-131: Verwendung des SmartBelts

Stauförderer

Die Verwendung eines Stauförderers erhöht die Produktabfolge. In der folgenden Abbildung ist Band 2 als Stauförderer dargestellt. Seine Länge (L2) muss länger als die Länge der Produkte auf dem vorhergehenden Band (L1) sein. Sonst schieben die Produkte von Band 1 die Produkte auf Band 2.

Damit sich die Produkte ansammeln können, muss die Geschwindigkeit des Stauförderers geringer sein als die Geschwindigkeit von Band 1.

Damit die Produkte einen vordefinierten Abstand voneinander erhalten, muss Band 3 mit einer höheren Geschwindigkeit laufen als der Stauförderer.

Abstände zwischen den Produkten auf dem Stauförderer resultieren in noch größere Abständen auf Band 3. Diese Abstände sind jedoch nicht zwingenderweise ein Vielfaches des gewünschten Abstands. Um diese Situation zu vermeiden, kann die Geschwindigkeit des Stauförderers gesteuert werden.

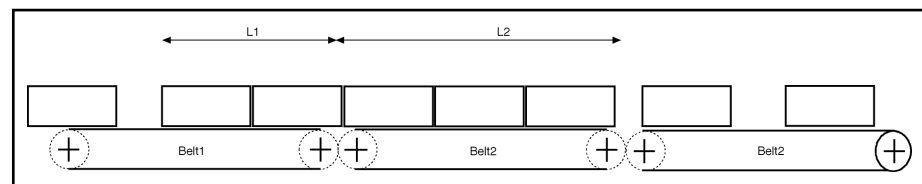


Abb.6-132: Stauförderer

Abfolgegrad

Auswertung mittels Produktzyklen und Abständen

Der Abfolgegrad ist entscheidend für die Anzahl der SmartBelts. Je höher der Grad, desto weniger SmartBelts werden benötigt. Der Abfolgegrad hat direkte Auswirkungen auf die SmartBelt-Betragskorrektur. Dieser wird mit der folgenden Formel berechnet:

Die folgende Abbildung erläutert die verschiedenen Parameter:

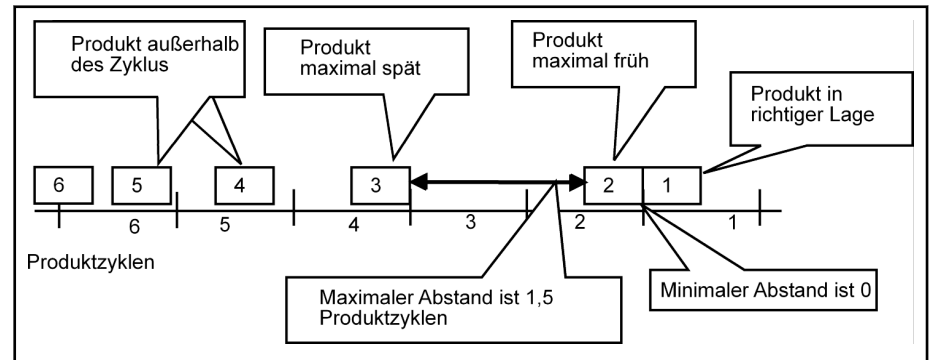


Abb.6-133: Auswertung mittels Produktzyklen und Abständen

Allgemein wird angenommen, dass die positive Produktflanke der Referenzpunkt ist. Solange immer derselbe Referenzpunkt verwendet wird, spielt es bei den folgenden Schlüssen keine Rolle, ob die positive/negative Flanke oder die Produktmitte betrachtet wird.

In der Abbildung oben, befindet sich Produkt 1 in der gewünschten Position und muss nicht korrigiert werden. Produkt 2 ist zu nahe an Produkt 1, aber immer noch innerhalb des Produktzyklus. Produkt 2 hat einen minimalen Abstand von 0 zum vorherigen Produkt. Produkt 3 ist zu spät, aber immer noch innerhalb des Produktzyklus. Die Produkte 4 und 5 befinden sich außerhalb ihres Produktzyklus, Produkt 6 hingegen ist innerhalb seines Produktzyklus.

Die Betragskorrektur wird mit Hilfe der folgenden Variablen berechnet:

- Produktlänge (L_{Prod})
- Nominaler Produktabstand bei vorgegebener Geschwindigkeit (L_{Abs})
- Nominaler Produktzyklus bei vorgegebener Geschwindigkeit (L_{Abs})
- Minimaler Produktabstand (L_{Absmin})
- Maximaler Produktabstand (L_{Absmax})
- Aufeinanderfolgende Produkte außerhalb des Produktzyklus (N_{Abs})

Die folgenden Werte gelten für die obenstehende Abbildung:

L_{Prod} : 40 mm (angenommen)

L_{Abs} : 50 mm (angenommen)

L_{zykl} : 90 mm ($= L_{\text{Prod}} + L_{\text{Abs}}$)

$L_{\text{Abs. min}}$: 0 mm (Abstand zwischen den Produkten 1 und 2)

$L_{\text{Abs. max}}$: 1.5 Produktzyklus = $1.5 * L_{\text{zykl}} = 135$ mm (Abstand zwischen den Produkten 2 und 3)

N_{Abs} : 2 (Abstand zwischen den Produkten 4 und 5)

Die benötigte Betragskorrektur für die SmartBelts wird wie folgt angegeben:

$$\text{Korrektur} = \min \left(L_{\text{max Abs}}, \left(1.5 * L_{\text{zykl}} - L_{\text{Prod}} + L_{\text{Abs}} * (N_{\text{Abs}} - 1) - (N_{\text{Abs}} * L_{\text{min Abs}}) \right) \right)$$

Der folgende Wert gilt für dieses Beispiel:

ML_TechMotion.library

Auswertung mittels Produktgeschwindigkeit und Dauer

$$\begin{aligned}
 \text{Korrektur} &= \min (135 \text{ mm}, (1.5 * 90 \text{ mm} - 40 \text{ mm} + 50 \text{ mm} * (2 - 1) - (2 * 0 \text{ mm}))) \\
 &= \min (135 \text{ mm}, 145 \text{ mm}) \\
 &= 135 \text{ mm}
 \end{aligned}$$

In diesem Beispiel sind 135 mm die Entfernung, die der SmartBelt durchfahren muss, um die Produktpositionen in beiden Richtungen zu korrigieren.

Auch mittels der Produktgeschwindigkeit kann geprüft werden, ob das Korrekturvermögen ausreicht. Die Geschwindigkeit sollte konstant sein, es gibt aber Abweichungen, die sich in der Nähe des Mittelwerts bewegen. Die Dauer und die Höhe dieser Abweichung kann auch dazu verwendet werden, um festzustellen, ob ein SmartBelt diese Anwendung unterstützen kann.

Die Formel zu Berechnung der Zeitabweichung ist folgende:

$$t = \frac{1}{\frac{R}{60} * \frac{\text{Abweichung}[\%]}{100}}$$

R ist die gewünschte Produktgeschwindigkeit in [1/min] und "Abweichung" ist der Prozentsatz um den die Produktgeschwindigkeit abweicht.

Beispiel:

Bei einer gewünschten Produktgeschwindigkeit von 200 Produkten pro Minute kommt es zu einer Abweichung von 10%. Das heißt, dass 180 bis 220 Produkte pro Minute über das Band laufen können. Die oben genannte Formel ergibt eine Zeit von:

$$t = \frac{1}{\frac{200}{60} * \frac{10\%}{100}} = 3s$$

Bei einer Dauer von weniger als 3 Sekunden in der die Produktrate von der nominalen Rate abweicht, kann der SmartBelt die Anwendung unterstützen. Ansonsten kommen Produkte ohne Korrektur vom Band oder es kommt zu leeren Plätzen.

Reibung

Der Zusammenhang zwischen Reibung und möglicher Beschleunigung lässt sich wie folgt erklären:

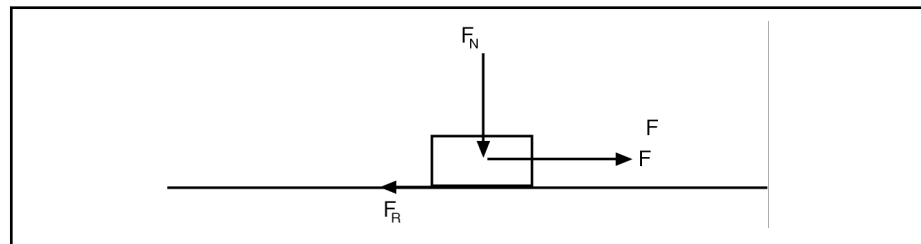


Abb.6-134: Reibung

Statische Reibung ist definiert als:

$$F_R = \mu_H F_N$$

mit

$$F_N = F_G = m * g$$

und ersetzt die zweite Formel durch die erste Formel

$$F_R = \mu_H * m * g$$

Benötigte Kraft für die Beschleunigung ist:

$$F = m \cdot a$$

Die dynamische Reibung beginnt bei $F = F_R$

oder ersetzt durch

$$\mu_H \cdot m \cdot g = m \cdot a_{\max}$$

gelöst für $a_{\max}$:

$$a_{\max} = \mu_H \cdot g$$

$a_{\max}$ ist die maximale Beschleunigung, μ_H ist der Reibungskoeffizient und g ist die Schwerkraftbeschleunigung.

μ_H kann auch anders visualisiert werden, wie die folgende Abbildung zeigt. μ_H ist das Äquivalent zu $\tan \phi$. In anderen Worten, ein Objekt mit einer Reibung von 1 würde bei einem Winkel von 45° anfangen zu rutschen.

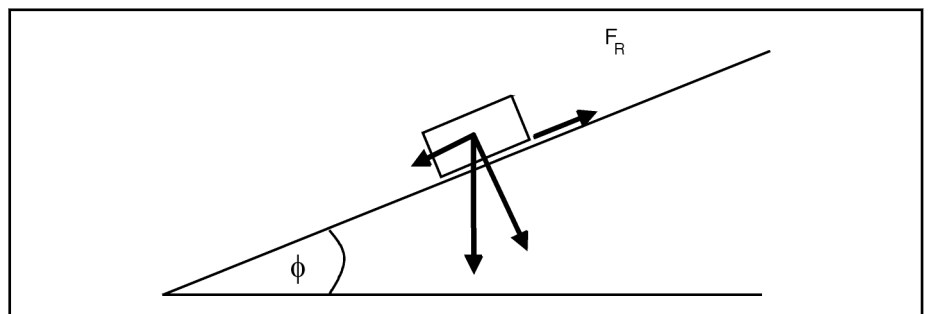


Abb.6-135: Zusammenhang zwischen Reibungskoeffizient und Winkel

6.9.3 Projektplanung

Allgemeines

Dieser Abschnitt beschreibt die durchzuführenden Einstellungen und Berechnungen, wenn der Funktionsbaustein MB_SmartBeltType01 verwendet wird.

Mechanische Anforderungen an Band und Maschine

Die Bandlänge sollte ungefähr die Produktgröße plus den gewünschten Abstand haben (Produktzyklus). Mit Bandlänge ist die physikalische Länge des gesamten Bands gemeint und nicht der Umfang. Es gibt keinen Vorteil, wenn das Band deutlich länger oder kürzer als der Produktzyklus ist. Bei einem längeren Band sind mehrere Produkte auf einem Band. Daher muss das letzte Produkt darauf warten, dass das erste Produkt das Band verlässt. Bei kürzeren Bändern werden weniger Zeit und schnellere Beschleunigungen benötigt um die gleichen Abstände zu erhalten.

Der Messtaster muss für jedes Band an den entsprechenden Antriebseingang angeschlossen sein. Merken Sie sich den verwendeten Messtastereingang. Der Messtaster muss so nah wie möglich an den Bandanfang platziert werden, wenn die positive Flanke erkannt wird. Das Produkt muss seinen Massenschwerpunkt auf dem Band haben. Bei symmetrischen Produkten ist dies die Hälfte der Produktlänge. Der Messtaster muss auf der Hälfte der Produktlänge plus die Hälfte des Achsdurchmessers des Bandes (wie in [Abb. 6-136 "Messtaster mit Erkennung der positiven Flanke" auf Seite 268](#) dargestellt) angebracht sein.

Wird die negative Flanke des Produkts verwendet, wird eine ähnliche Logik angewendet. Der Messtaster muss dann auf der Hälfte der Produktlänge minus der Hälfte des Achsdurchmessers des Bandes vor dem Band angebracht

ML_TechMotion.library

werden, wie in [Abb. 6-137 "Messtaster mit Erkennung der negativen Flanke"](#) auf Seite 268 dargestellt.

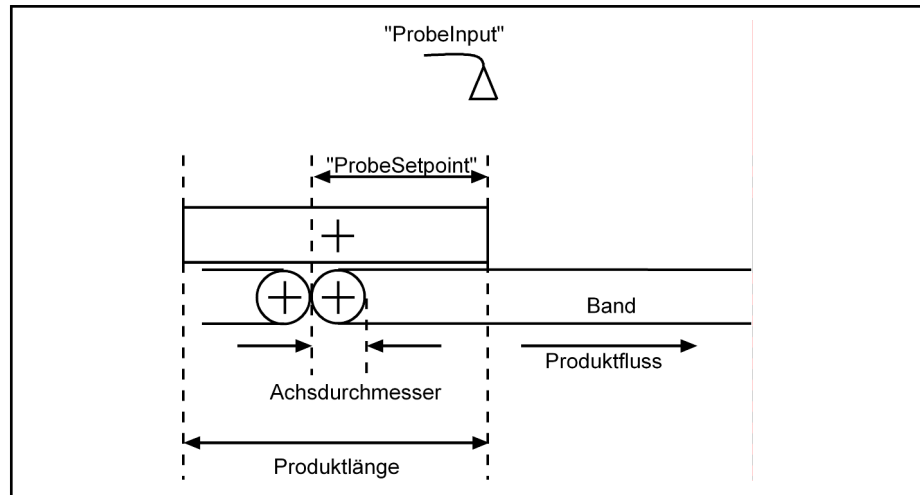


Abb.6-136: Messtaster mit Erkennung der positiven Flanke

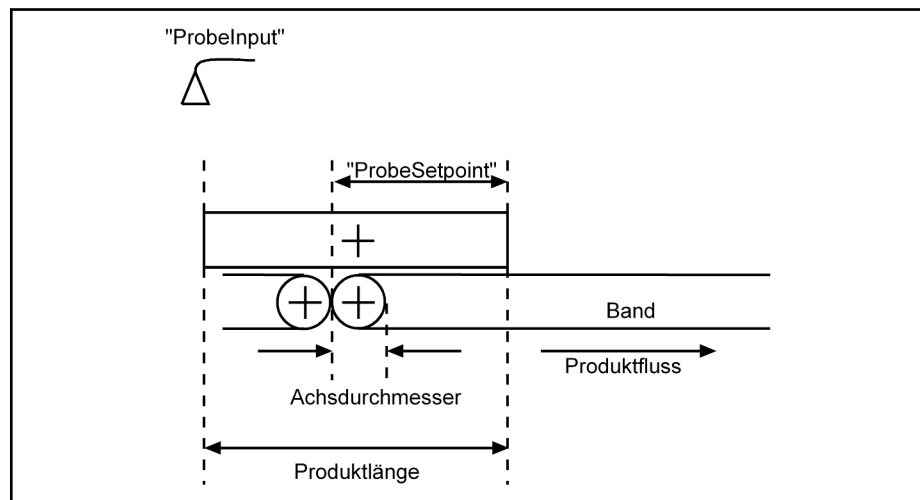


Abb.6-137: Messtaster mit Erkennung der negativen Flanke

Wenn mehrere Produkte mit den unterschiedlichen Formaten auf demselben SmartBelt verwendet werden sollen, muss der Messtaster für die Registrierung mit der negativen Flanke angebracht werden. In diesem Fall bestimmt das kleinste Produkt den Ort des Messtasters. In diesem Fall wird die Kompensationsfähigkeit größerer Produkte verringert.

Antriebsanforderungen

Der Antrieb muss mit einem Funktionspaket ausgestattet sein, das synchrone Betriebsarten und Messtastereingänge unterstützt.

Folgende Anforderungen gelten an die IndraMotion MLC:

- Nur IndraDrives mit Interpolation im Antrieb können verwendet werden. Die Antriebsfirmware muss mindestens MPx04V22 oder höher sein. MPx03 wird nicht unterstützt.

Folgende Anforderungen gelten für die IndraMotion MLD:

- Die Antriebsfirmware muss mindestens MPx05VRS oder höher sein. MPx03VRS und MPx04VRS werden nicht unterstützt.
- Das MA Paket muss aktiviert sein.

Antriebsparametereinstellungen

Es wird empfohlen, die Basisparameter vor der Antriebskonfiguration zu laden. In diesem Abschnitt werden die Unterschiede zu den Standardparametern aufgelistet.

Vor der Verwendung des Funktionsbausteins, müssen zusätzliche Einstellungen vorgenommen werden. Die Messtastereinstellungen müssen korrekt sein, da der Funktionsbaustein die Messtasterfunktionalität nutzt. Abhängig von dem verwendeten Messtastereingang und der Flanke zum Auslösen, werden die folgenden Einstellungen für den optionalen zyklischen Kanal benötigt. Um verschiedene Möglichkeiten zur Laufzeit zu haben, können mehrere Einstellungen konfiguriert werden. Setzen Sie beispielsweise S-0-0130 und S-0-0131, sowie S-0-0409 und S-0-0410 im zyklischen Lesekanal. Dies erlaubt die Verwendung von entweder positiven oder negativen Flanken zur Laufzeit.



Die folgenden Parameter werden in der Steuerung gesetzt, damit die zyklischen Parameter zum Antrieb übertragen werden können. Zusätzlich im Antrieb gesetzte Parameter sind für die Messtasterfunktionalität notwendig.

Der Antriebsparametereditor von IndraWorks kann für die Parametereinstellungen verwendet werden.

IndraMotion MLC Parametereinstellungen

Gewünschte Konfiguration	Zyklischer Lesezugriff	Signalsteuerwort
Messtaster 1, positive Flanke	S-0-0130, S-0-0409	S-0-0405, Bit 0
Messtaster 1, negative Flanke	S-0-0131, S-0-0410	S-0-0405, Bit 0
Messtaster 2, positive Flanke	S-0-0132, S-0-0411	S-0-0406, Bit 0
Messtaster 2, negative Flanke	S-0-0133, S-0-0412	S-0-0406, Bit 0

IndraMotion MLD Parametereinstellungen

Gewünschte Konfiguration	Istwerte	Sollwerte
Messtaster 1, positive Flanke	S-0-0130 S-0-0409	S-0-0405, Bit 0
Messtaster 1, negative Flanke	S-0-0131 S-0-0410	S-0-0405, Bit 0
Messtaster 2, positive Flanke	S-0-0132 S-0-0411	S-0-0406, Bit 0
Messtaster 2, negative Flanke	S-0-0133 S-0-0412	S-0-0406, Bit 0

Messtasterfunktionalität im Antrieb

Die nachfolgenden Parameter sind im Antrieb gesetzt um die Antriebsmesstasterfunktionalität zu aktivieren:

Parameter	Wert	Beschreibung
P-0-0300	S-0-0401 (Messtaster 1) bzw. S-0-0402 (Messtaster 2)	Digitale Eingangsbelegung der Messtasterfunktionalität
P-0-0301	0 (Bit 0)	Bitbelegung zur Messtasterfunktionalität

ML_TechMotion.library

S-0-0426	P-0-0775	Leitachspannung für die Messtastermessung
S-0-0169	Bit 0 durch 3	Aktivierung der positiven bzw. negativen Flankenerkennung für die Messtaster 1 und 2
P-0-0226	Bit 9 = TRUE	Messtasterfunktion wird automatisch bei Eintritt in Phase 4 im Antrieb aktiviert

Zusätzlich zu den anwendungsspezifischen Einstellungen, müssen die folgenden Parameter gesetzt sein.

Hardwareunabhängige Einstellungen

Parameter	Wert	Beschreibung
P-0-0072 P-0-0092 P-0-0780 P-0-0781 P-0-0782 P-0-0783 P-0-0784 P-0-0785	Lineare Kurvenscheibe mit dem Verhältnis 1:1 zwischen Leitachse und Folgeachse. Es muss nur ein Parameter (Kurvenscheibe) gesetzt werden. Die CamBuilder-Software kann verwendet werden um eine Kurvenscheibe zu erstellen und herunterzuladen	Die dazugehörige Kurvenscheibennummer (CamTableID) muss im MC_CamIn Baustein referenziert werden, wenn ein synchroner Betriebsmodus erstellt wird
S-0-0121	Eingangsumdrehungen des Lastgetriebes	Eingangsumdrehungen des mechanischen Getriebes
S-0-0122	Ausgangsumdrehungen des Lastgetriebes	Ausgangsumdrehungen des mechanischen Getriebes
S-0-0123	Lineare Bewegung pro Umdrehung	Vorschubkonstante für den Antrieb

Achspanparametereinstellungen der IndraMotion MLC

Parameter	Wert	Beschreibung
A-0-0045	Doppelte Bandlänge	Modulowert der Achse
A-0-0059	Bit 0 = TRUE, Bit 1 und 2 = FALSE, Bit 3 = FALSE, Bit 7 = TRUE	Auswahl der Linear-, Modulo-, und bevorzugten Wichtung
A-0-0059	Bit 4 = FALSE für Einheiten mm, = TRUE für Einheiten in Inch	Auswahl der Bandeinheiten
A-0-2790	Hoch genug für die Anwendung	Aufsynchronisierungsgeschwindigkeit
A-0-2791	Hoch genug für die Anwendung	Aufsynchronisierungsbeschleunigung
A-0-0032	Hoch genug für die Anwendung	Geschwindigkeitsgrenzwert positiv
A-0-0033	Hoch genug für die Anwendung	Geschwindigkeitsgrenzwert negativ
A-0-0034	Hoch genug für die Anwendung	Beschleunigungsgrenzwert

Die Leitachse für das Band muss auch eine Modulo- und bevorzugte Wichtung (A-0-0059: Bit 3 = FALSE, Bit 7 = TRUE) sowie eine rotatorische bzw. eine lineare Wichtung haben. Eine Leitachsumdrehung entspricht einem Produktzyklus (Produktlänge plus Abstand zwischen den Produkten) für eine rotatorische Leitachse. Bei einer linearen Leitachse, muss der Modulowert dem Produktzyklus entsprechen. Entspricht ein Leitachszyklus mehreren oder Teilen eines Produktzyklus, muss das Getriebeverhältnis entsprechend eingestellt werden. Entspricht beispielsweise eine Leitachsumdrehung 3 Produkten, sind die Eingänge des Getriebes wie folgt:

- "RatioNumerator"/"InputRevolution" muss auf 1 gesetzt sein
- "RatioDenominator"/"OutputRevolution" muss auf 3 gesetzt sein

IndraMotion MLD Parametereinstellungen

Parameter	Wert	Beschreibung
S-0-0103	Doppelte Bandlänge	Modulowert der Achse
S-0-0076	Bit 0 = TRUE, Bit 1 und 2 = FALSE, Bit 3 = FALSE, Bit 7 = TRUE	Auswahl der Linear-, Modulo-, und bevorzugten Wichtung
S-0-0076	Bit 4 = FALSE für Einheiten mm, = TRUE für Einheiten in Inch	Auswahl der Bandeinheiten
P-0-0143	Hoch genug für die Anwendung	Aufsynchronisierungsgeschwindigkeit
P-0-0142	Hoch genug für die Anwendung	Aufsynchronisierungsbeschleunigung
S-0-0038	Hoch genug für die Anwendung	Geschwindigkeitsgrenzwert positiv
S-0-0039	Hoch genug für die Anwendung	Geschwindigkeitsgrenzwert negativ
S-0-0138	Hoch genug für die Anwendung	Beschleunigungsgrenzwert

Die Leitachse für das Band muss auch über eine Modulo- und eine bevorzugte Wichtung (P-0-0756 bei einem Achsgenerator) sowie über rotatorische bzw. lineare Wichtung verfügen. Eine Leitachsumdrehung entspricht einem Produktzyklus (Produktlänge plus Abstand zwischen den Produkten) für eine rotatorische Leitachse. Bei einer linearen Leitachse, muss der Modulowert dem Produktzyklus entsprechen. Entspricht ein Leitachszyklus mehreren oder Teilen eines Produktzyklus, muss das Getriebeverhältnis entsprechend im SPS Programm eingestellt werden. Wenn beispielsweise eine Leitachsumdrehung drei Produkten entspricht, müssen die Eingänge "RatioNumerator"/"InputRevolution" auf 1 und "RatioDenominator"/"OutputRevolution" auf 3 gesetzt werden.

SPS-Programmierung

Die Achse mit der der SmartBelt betrieben werden soll, muss in einer synchronen Betriebsart sein. Wie diese einzustellen ist kann anhand der folgenden Code-Fragmente nachvollzogen werden.

Strukturierter Text:

Programm:

```
myMC_CamIn(Execute:= TRUE,
            RatioNumerator:= 1,
            RatioDenominator:= 1,
            MasterFineadjust:= 0,
            CamShaftDistance:= myProductCycle,
            SyncMode:= SHORTEST_WAY,
            StartMode:= RELATIVE,
            CamTableID:= myCamTableID,
            Master:= mySmartBeltMaster,
            Slave:= myBelt,
            InSync=> ,
            Active=> ,
            CommandAborted=> ,
            Error=> ,
            ErrorID=> ,
            ErrorIdent=> ,
            CamActiveID=>
);
```

Funktionsplan:

ML_TechMotion.library

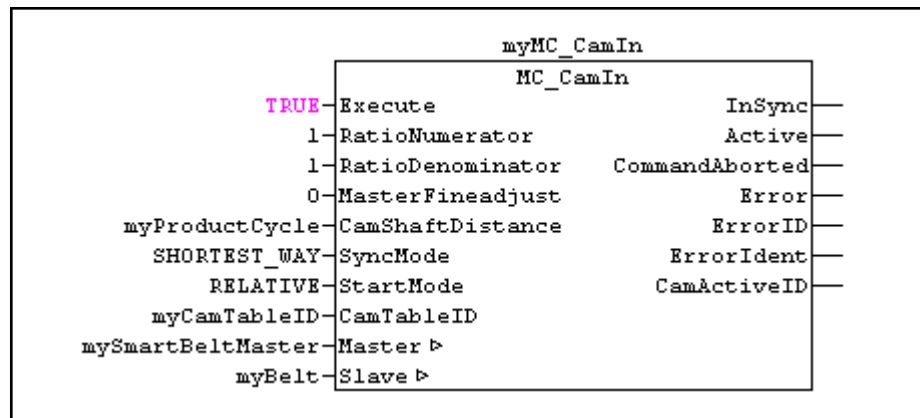


Abb.6-138: Einstellungen des MC_CamIn für den Einsatz mit dem SmartBelt

Wird das **AchsInterface** programmiert, muss der Betriebsmodus für den Antrieb auch auf "CamIn" gesetzt werden. Nachfolgend ein Beispiel:

Strukturierter Text:

Programm:

```
arAxisCtrl[myBelt.AxisNo].SyncMode.Master:= mySmartBeltMaster;
arAxisCtrl[myBelt.AxisNo].SyncMode.SyncDirection:= SHORTEST_WAY;
arAxisCtrl[myBelt.AxisNo].SyncMode.StartMode:= RELATIVE;
arAxisCtrl[myBelt.AxisNo].SyncMode.InputRevolution:= 1;
arAxisCtrl[myBelt.AxisNo].SyncMode.OutputRevolution:= 1;
arAxisCtrl[myBelt.AxisNo].SyncMode.FineAdjust:= 0;
arAxisCtrl[myBelt.AxisNo].SyncMode.CamShaftDistance:= myProductCycle;
arAxisCtrl[myBelt.AxisNo].SyncMode.CamTableID:= CAM_TABLE_1;
arAxisCtrl[myBelt.AxisNo].Admin._OpMode:= ModeSyncCam;
```

Der Synchronisationsabstand muss SHORTEST_WAY sein, um sicherzustellen, dass sich der SmartBelt vorwärts und rückwärts bewegt.

Der Startmodus muss auf RELATIVE gesetzt sein. Dadurch kann der Benutzer den SmartBelt mit einer Leitachse im Stillstand synchronisieren, ohne dass sich das Band bewegt.

Der Eingang "CamShaftDistance" muss auf den Produktzyklus der Anwendung gesetzt werden. Dies ist die Länge des Produkts plus der Abstand zwischen den Produkten. Bei dieser Länge muss der SmartBelt festlegen, ob die korrekte Bewegung vorwärts oder rückwärts ist. Auch wenn das Verhältnis zwischen Leitachse und SmartBelt nicht 1:1 ist, muss dieser Wert für den Produktzyklus gesetzt werden. In diesem Fall wird das Getriebe eingesetzt, um das Verhältnis zwischen Leitachse und SmartBelt zu wählen.

Die im Antrieb verwendete ID der Kurvenscheibe kann beliebig sein, muss aber entsprechend dem "CamTableID"-Element gesetzt werden. Vor seiner Verwendung muss die Kurvenscheibe auf den Antrieb heruntergeladen werden. Die Kurvenscheibe muss im Verhältnis 1:1 sein. Diese kann mit Hilfe der CamBuilder-Software erstellt werden. Um die Downloadzeit zu verringern, kann die Anzahl der Kurvenscheibenpunkte auf die minimal möglichen Punkte reduziert werden.

Einem Übersetzungsverhältnis von 1:1 folgt einem 1:1-Verhältnis von Leitachszzyklus zu Produktzyklus. Ist das Verhältnis zwischen den Zyklen anders, muss das Übersetzungsverhältnis entsprechend eingestellt werden.

Angabe des "SmartBeltPitch"

Der "SmartBeltPitch" gibt die physikalisch wirksame Länge des SmartBelt an. Die Länge des SmartBelts sollte ungefähr der Länge des Produktes + Lücke zwischen den Produkten (= Produktzyklus) entsprechen. Der Messtaster ist am Anfang des SmartBelts montiert und erfasst entweder den Produktanfang oder das Produktende. Der SmartBelt Funktionsbaustein nimmt dabei an, dass sich das Produkt nach der Erfassung durch den Messtaster um mindestens den "SmartBeltPitch" weiter bewegen muss, bevor es das Band verlässt und dann erst kann mit der Korrektur eines nachfolgenden Produktes begonnen werden kann. Dies ist in der nachfolgenden Zeichnung zu sehen. Hier ist der Produktzyklus etwas kleiner als die Länge des SmartBelts. Im gezeichneten Fall würde sich das Produkt 1 noch bis zur gestrichelt dargestellten Position weiter bewegen, bevor das nachfolgende Produkt (2) korrigiert werden kann.

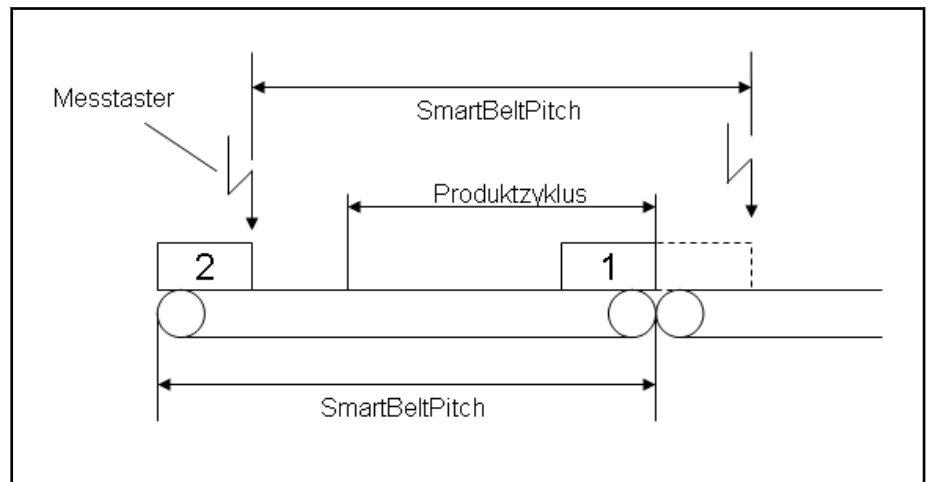


Abb.6-139: SmartBelt bei dem die physikalische Länge in etwa dem Produktzyklus entspricht

Diese Annahmen sind korrekt solange der Messtaster am Anfang des SmartBelts in dem Bereich montiert wird, die maximal der Länge des Produktes entspricht (siehe nachfolgendes Bild). In diesem Fall ist der SmartBelt etwas länger als zwei Produktzyklen. Ausgehend vom zweiten Band ist in dieser Zeichnung auch noch ein Produktzyklus A eingezeichnet, der ausgehend vom Messtaster des zweiten Bandes rückwärts eingezeichnet ist. Daraus ist ersichtlich, dass ein nachfolgendes Produkt sich immer noch auf dem ersten Band befindet, wenn das erste Produkt die Länge von "SmartBeltPitch" durchfahren hat. Damit ist dann eine Korrektur des zweiten Produktes noch möglich.

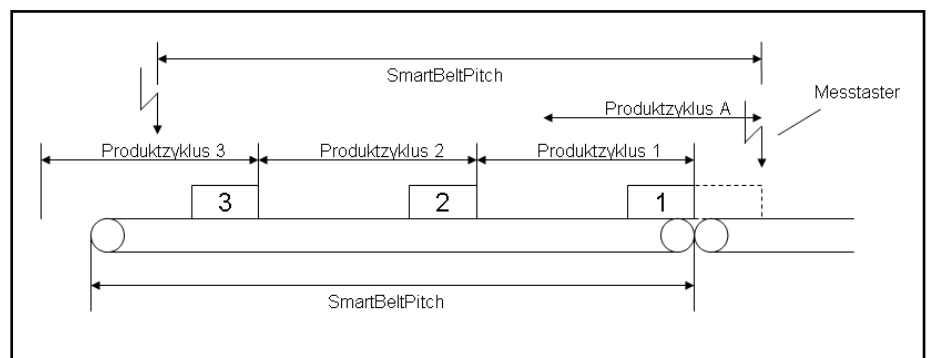


Abb.6-140: SmartBelt mit physikalischer Länge die ein mehrfaches des Produktzyklus entspricht

Wird der Messtaster allerdings nicht am Anfang des SmartBelts montiert, so arbeitet der SmartBelt nicht korrekt, da er nach dem Erkennen des Produktes dieses noch eine volle "SmartBeltPitch" Länge weiter bewegen möchte. In diesem Fall ist das Produkt bereits lange vom Band bevor das nachfolgende Produkt korrigiert wird. Im eingezeichneten Fall wäre auch das nachfolgende Produkt bereits vom Band, da Produktzyklus A vollständig auf dem nachfolgenden Band liegt. Eine Korrektur des nachfolgenden Produktes wäre damit nicht mehr möglich.

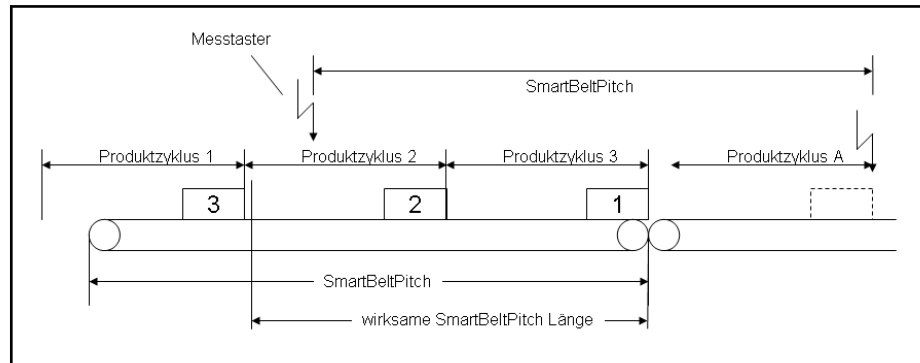


Abb.6-141: SmartBelt mit Messtaster der nicht am Anfang des Bandes montiert ist

Um dies sicherzustellen, darf nicht die physikalische Länge angegeben werden, sondern es muss die wirksame Länge des SmartBelts angegeben werden. Diese entspricht dem Abstand vom Ende des Bandes bis zum Messtaster + einer Produktlänge. Diese Strecke ist auch in dem Bild eingezeichnet. Die Differenz zwischen der wirksamen Länge und der physikalischen Länge ist der Bereich am Anfang des SmartBelts in dem das Produkt zwar auf dem Band ist, aber noch nicht vom Messtaster erfasst wurde. Damit wird es in diesem Bereich auch noch nicht korrigiert und ist damit unerheblich für den Funktionsbaustein.



Für die nachfolgende Berechnung des "ProbeSetpoint" muss die physikalische Länge des SmartBelts und nicht die wirksame Länge verwendet werden!

Berechnung des "ProbeSetpoint"

Der Eingang "ProbeSetpoint" erkennt die gewünschte Leitachsposition in dem Moment wenn das Produkt den Messtaster des SmartBelts passiert. Dieser Punkt kann berechnet werden, indem die Entfernung der Messtasterposition und das Ende aller SmartBelts, z.B. wo das Produkt ausgerichtet sein muss, benutzt werden. Die folgende Abbildung stellt dies dar:

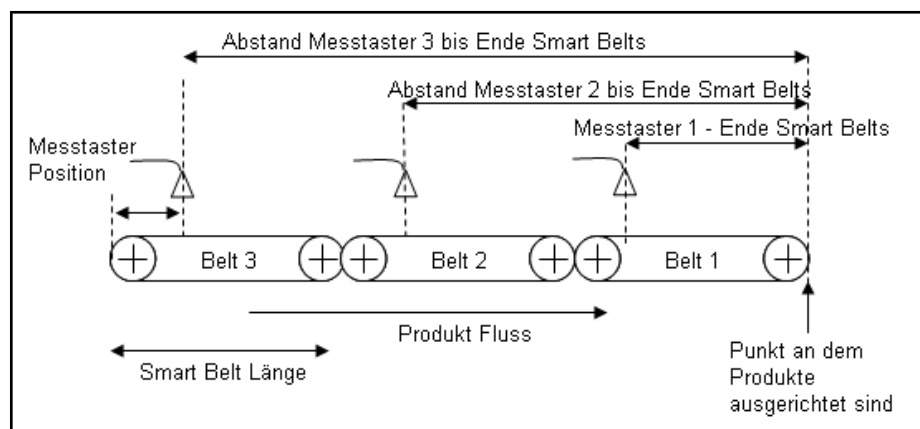


Abb.6-142: Messungen um den "ProbeSetpoint" zu berechnen

Zunächst müssen alle Abstände zwischen den Messtastern bis zum Ende aller SmartBelts, z.B. wo die Produkte ausgerichtet werden, gemessen werden. Anschließend kann die nachfolgende Formel benutzt werden, um den Wert für die **positive Flankenerkennung** auf dem Messtastereingang zu berechnen:

$$\text{ProbeSetpoint}_k = L_{\text{cycl}} - \left(\left(L_{\text{SmartBelt}_n} - L_{\text{ProbePos}_n} \right) + \sum_{n=1}^{k-1} L_{\text{SmartBelt}_n} \right) \text{MOD} (L_{\text{cycl}})$$

Abb.6-143: Formel für die positive Flankenerkennung

Die nachfolgende Formel gilt für die **negative Flankenerfassung** des Messtastereingangs:

$$\text{ProbeSetpoint}_k = L_{\text{cycl}} - \left(\left(L_{\text{SmartBelt}_n} - L_{\text{ProbePos}_n} - L_{\text{Prod}} \right) + \sum_{n=1}^{k-1} L_{\text{SmartBelt}_n} \right) \text{MOD} (L_{\text{cycl}})$$

Abb.6-144: Formel für die negative Flankenerkennung

Die nachfolgenden Variablen werden in den Berechnungen verwendet:

- L_{cycl} : Länge des Produktzyklus (Produktlänge (L_{Prod}) plus Länge des Abstands)
- k : Anzahl der verwendeten SmartBelts
- L_{ProbePos} : Messtasterposition an einem bestimmten SmartBelt (gemessen vom Anfang des jeweiligen SmartBelts)
- $L_{\text{SmartBelt}}$: Länge des SmartBelts

Angenommen 3 SmartBelts, alle 150 mm lang, mit einem Messtaster, der 30mm vom Anfang des SmartBelts angebracht ist und einem Produktzyklus von 120 mm sowie der Vorgabe die positive Produktflanke zu erkennen. Die folgenden "ProbeSetpoint"-Werte können dann wie folgt berechnet werden:

- SmartBelt 1: $k=1$; $\text{ProbeSetpoint}_1 = 120 - (((150-120) + 0) \text{MOD} (120)) = 120 - (30 \text{MOD} 120) = 90$
- SmartBelt 2: $k=2$; $\text{ProbeSetpoint}_2 = 120 - (((150-120) + 150) \text{MOD} (120)) = 120 - (180 \text{MOD} 120) = 60$
- SmartBelt 3: $k=3$; $\text{ProbeSetpoint}_3 = 120 - (((150-120) + 300) \text{MOD} (120)) = 120 - (330 \text{MOD} 120) = 30$

6.9.4 SmartBelt-Beschränkungen und -Verhalten

Beschränkungen

Der Funktionsbaustein wird für Winkelkorrekturen von Produkten auf einem Band verwendet. Die Produkte müssen mit einem Abstand auf dem Band platziert werden, damit jedes Produkt identifiziert werden kann. Ist die Distanz zu gering, wird das zweite Produkt nicht erkannt und führt zu einer falschen Korrektur.

Die Grenzwerte für Beschleunigung und/oder maximale und minimale Entfernungen müssen Werte sein, die erreicht werden können. Ansonsten rutschen die Produkte auf dem Band. Dies führt zu falschen Korrekturen.

Idealerweise ist auf dem Band immer nur ein Produkt. Sind zwei oder mehr Produkte auf dem Band, wird das zweite Produkt erst korrigiert, nachdem das erste Produkt das Band verlassen hat. Der verbleibende Weg für die Korrektur ist daher geringer. Zusätzlich erfährt auch das zweite Produkt die Korrekturen des ersten Produktes. Beide Werte werden beim Berechnen der Be-

tragskorrektur für das zweite Produkt berücksichtigt. Jedoch ist es nicht möglich, dass das zweite Produkt genauso korrigiert wird als wäre es alleine auf einem Band.

Zwei oder mehr Produkte auf einem Band

Sind zwei Produkte auf einem Band, wird das zweite Produkt nicht korrigiert bis das erste Produkt das Band verlassen hat. In diesem Fall heißt "das Band verlassen", dass das erste Produkt einen Weg "SmartBeltPitch" vom Registrierungspunkt zurückgelegt hat. In der nachfolgenden Abbildung hat das Produkt, beschrieben mittels einer gepunkteten Linie, "das Band verlassen", da es seit der Registrierung auf Band 1 (Blickwinkel: positive Flanke des Produkts) eine ganze "SmartBeltPitch" gefahren ist. Dieser Wert ist fest vorgegeben.

Das zweite Produkt hat weniger Zeit/Abstand, um Korrekturen vorzunehmen. Der verbleibende Abstand ist der "DistanceForCorrection" minus des bereits gefahrenen Weges. Mit diesem Wert können alle Variablen (Geschwindigkeit, Beschleunigung und Zeit) für das zweite Produkt berechnet werden.

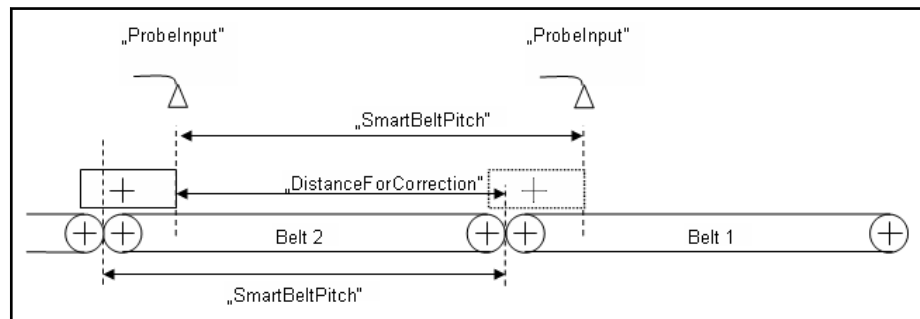


Abb.6-145: Zwei oder mehr Produkte auf einem Band

Ein zusätzliches Problem ist, dass das zweite Produkt stets rückwärts korrigiert werden muss. Die Korrektur wird aus den Einstellungen in "PercentForwardLimit", "MaxCorrectionDistancePos" und "MaxCorrectionDistanceNeg" berechnet. Wenn ein zweites Produkt auf dem Band ist und das erste Produkt gerade von Band genommen wurde, berechnet der Algorithmus einen neuen Korrekturwert für das zweite Produkt. Dieser könnte bewirken, dass das zweite Produkt nach vorne geschoben wird. Dies ist jedoch nicht möglich, da dort das erste Produkt ist. In diesem Fall wird ein ganzer Produktzyklus abgezogen, um das Produkt auf den nächsten freien Platz zu schieben. Jede Rückwärtsbewegung ist auf "MaxCorrectionDistanceNeg" begrenzt.

Mehrere Produkte pro Leitachszzyklus

Wenn mehrere Produkte oder Teile eines Produkts einem Leitachszzyklus entsprechen, müssen zusätzliche Einstellungen vorgenommen werden. Das Übersetzungsverhältnis bestimmt die Beziehung zwischen den Zyklen.

Wenn beispielsweise eine Leitachsumdrehung drei Produkten entspricht, müssen die Eingänge "RatioNumerator"/"InputRevolution" auf 1 und "RatioDenominator"/"OutputRevolution" auf 3 gesetzt werden (siehe auch [Kap. "SPS-Programmierung" auf Seite 271](#)).



Es ist wichtig zu vermerken, dass der Eingang "CamShaftDistance" nicht verwendet werden kann um das Verhältnis zwischen Leitachse und SmartBelt zu ändern. Ist "CamShaftDistance" nicht auf den Produktzyklus abgestimmt, kommt es zu falschen Korrekturen!

6.9.5 Tipps und Tricks

Korrekturabstand

Die vorhergehenden Abschnitte in diesem Kapitel beschreiben die Konfiguration der SmartBelt-Funktionalität. Dieser Abschnitt gibt zusätzliche Informationen, damit Abhängigkeiten deutlich werden und Optimierungen an dem bereits laufenden System durchgeführt werden können.

Der SmartBelt verwendet die Daten des Eingangs "DistanceForCorrection" um die zur Korrektur benötigten Abstände zwischen den Produkten zu berechnen. Dies ist normalerweise der Abstand zwischen der Messtasterposition und dem Bandende. Für Produkt A ist das in der Abbildung unten dargestellt. Dies ist sicherer, da das Produkt nur solange korrigiert wird, wie es sich noch vollständig auf dem Band befindet.

Produkt B verfügt über einen längeren Weg für die Korrektur. Der Massenschwerpunkt des Produktes ist an derselben Stelle wie die Mitte der SmartBelt-Achse. Hier ist die Möglichkeit des Schlupfes immer noch gering.

Bei Produkt C ist der Masseschwerpunkt zwischen den beiden angrenzenden SmartBelts. In diesem Fall kommt es bereits zu Schlupf.

Der Wert "DistanceToCorrect" kann von A bis C erhöht werden, um die Betragskorrektur zu erhöhen, während das Rutschen überwacht wird. Es hängt von der Anwendung, hauptsächlich von der Reibung und dem Durchmesser der SmartBelt-Achse ab, wie stark der Wert des "DistanceToCorrect" erhöht werden kann. Mit diesem Wissen kann die Betragskorrektur optimiert werden.

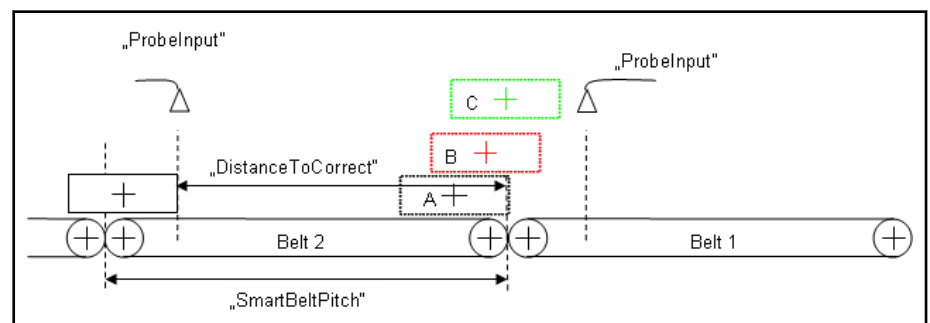


Abb. 6-146: Verschiedene Abstände für die Korrektur

Schlupf

Es kommt zu Schlupf, wenn das Produkt nicht so schnell beschleunigt werden kann wie der SmartBelt. In diesem Fall kommt es zu einem Fehler in der Betragskorrektur. Der Funktionsbaustein des SmartBelts benötigt die maximale Beschleunigung um die Höhe der möglichen Korrektur zu bestimmen und die Beschleunigung entsprechend zu begrenzen. Die mögliche Beschleunigung ist jedoch häufig unbekannt. Das Standardverhältnis zwischen einer möglichen Beschleunigung und der Reibung ist in [Kap. "Reibung" auf Seite 266](#) angegeben und erlaubt es einen möglichen Wert abzuschätzen.

Es gibt einen Weg um den Schlupf zu bestimmen. Jeder SmartBelt verfügt über den Ausgang "CorrectionValue", der die Betragskorrektur für den SmartBelt zeigt. Außerdem verfügt er über den Ausgang "ProbePosition", welcher die gemessene Produktposition erkennt. Jetzt muss der SmartBelt so konfiguriert werden, dass die Korrektur in Vorwärtsrichtung ("PercentForwardLimit" = 100% und "MaxCorrectionDistPos" = SmartBelt-Länge) durchgeführt wird.

ML_TechMotion.library

Anschließend werden mehrere Produkte dem SmartBelt so zugeführt, dass ihre Position die maximale Abweichung zur Leitachse, aufweist. In diesem Fall wird der Ausgang "AccelLimited" auf TRUE gesetzt, um anzuzeigen, dass die maximale Beschleunigung verwendet wurde (wird der Ausgang nicht TRUE, muss der Eingang "MaxAcceleration" für die maximale Beschleunigung solange verringert werden bis der Ausgang TRUE wird).

Nachdem die maximale Beschleunigung erreicht worden ist, sollten der Ausgang "ProbePosition" sowie "CorrectionValue" vermerkt werden. Auch der Ausgang "ProbePosition" des SmartBelts nach dem ersten SmartBelt sollte vermerkt werden. Die Grundregel ist, dass der "ProbePosition"-Wert des ersten SmartBelts minus "CorrectionValue" des ersten SmartBelts mit dem "ProbePosition"-Wert des zweiten SmartBelts übereinstimmt. Unterscheiden sich die Werte signifikant, liegt die Vermutung nahe, dass es zu Schlupf kam. In diesem Fall muss der Eingang "MaxAcceleration" weiter reduziert werden, bis es nicht mehr zu Schlupf kommt.

Stimmen die Werte überein, kann die Beschleunigung weiter erhöht werden um die Korrekturkapazität zu erhöhen. Dies kann so weit führen, dass der Maximalwert der Beschleunigung nie erreicht wird. Mit anderen Worten: Die gesamte Korrektur kann auf einem Band stattfinden.

6.9.6 Verbreitete Fehler und Missverständnisse

Frage	Antwort:
Der Antrieb synchronisiert nicht oder braucht sehr lange für die Synchronisation	Wenn der Antrieb in den linearen Modus geschaltet ist, ändern sich die Einheiten, aber die Standardwerte bleiben gleich. Die Beschleunigung und die Geschwindigkeit sind normalerweise sehr niedrig, so dass eine Synchronisation sehr langsam ist. Erhöhen Sie die Werte für A-0-2790 und A-0-2791 (IndraMotion MLC) bzw. P-0-0142 und P-0-0143 (IndraMotion MLD)
Fehler F2002028 (IndraMotion MLC) "Exzessive Regelabweichung" und F2028 (IndraMotion MLD) treten ständig auf	Setzen Sie die Parameter A-0-0032, A-0-0033 und A-0-0034 (IndraMotion MLC) bzw. S-0-0038, S-0-0039 und S-0-0138 (IndraMotion MLD) auf richtige Werte. Die Standardwerte für diese Parameter passen zu einer rotatorischen Wichtung, aber für die lineare Wichtung sind sie normalerweise zu klein und müssen angepasst werden

Frage	Antwort:
Der Funktionsbaustein ist so konfiguriert, dass er 100% korrigiert, aber das Produkt befindet sich nicht in der korrekten Position, nachdem es vom Band gekommen ist	Dafür kann es mehrere Gründe geben: Erstens, die Winkeldifferenz war wahrscheinlich zu groß, um innerhalb einer Bandlänge korrigiert zu werden. In diesem Fall werden die Ausgänge "NegCorrLimited", "PosCorrLimited" oder "AccelLimited" gesetzt. Zweitens, wenn die Werte "MaxCorrectionDistancePos", "MaxCorrectionDistanceNeg" oder "MaxAcceleration" zu hoch gesetzt werden, kann es sein, dass die Produkte physisch nicht dazu in der Lage sind, dem Band zu folgen und rutschen. Diese Bedingung wird nicht von den Statusausgängen angezeigt. Es können Maßnahmen ergriffen werden um die Reibung zu erhöhen (Vakuum Bänder) oder es müssen mehr Bänder verwendet werden, um die Fehler auf mehreren Bändern zu kompensieren. Drittens, sind zwei oder mehr Produkte auf dem Band, siehe nächste Frage
Befinden sich zwei oder mehr Produkte auf dem Band, so wird das zweite oder dritte Produkt nicht korrekt korrigiert	Mit der Korrektur des zweiten Produkts muss gewartet werden, bis das erste Produkt das Band verlassen hat. Dann wird es korrigiert. Ist der verbleibende Abstand zu klein, kann die notwendige Korrektur nicht erreicht werden
Befinden sich zwei Produkte Stoß auf Stoß auf dem Band, wird das zweite Produkt nicht korrigiert	Wenn die Produkte Stoß auf Stoß liegen, kann die Messtasterfunktion das zweite Produkt nicht erkennen. Der SmartBelt sieht nur ein Produkt. Es muss sichergestellt werden, dass der - den SmartBelts vorgelegte Prozess - die Produkte mit einem Minimalabstand voneinander trennt, damit die Messtasterfunktionalität das nächste Produkte erkennen kann
Alles ist konfiguriert, aber der SmartBelt nimmt keine Korrekturen vor und es liegt auch kein Fehler vor	Setzen Sie Bit 9 des Parameters P-0-0226 im Antrieb und schalten Sie in Phase 2 und dann wieder in Phase 4. Dies aktiviert die Messtasterfunktionalität automatisch. Sonst ist die Messtasterfunktionalität zwar konfiguriert aber nicht aktiviert. Dies führt dazu, dass keine Messtastermessungen durchgeführt werden und dadurch der Smart Belt auch kein Produkt erkennt

6.10 Funktionsbausteine zur Erstellung von Kurvenscheibentabellen für Technologiefunktionen

6.10.1 Einführung

Die Funktionsbausteine dienen der Erstellung von Kurvenscheibentabellen. Der MB_PilgrimStepCalcType01 übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunkten sowie die Hübe des Vorwärts- und Rückwärtstransportes und die Stopppositionen für den Pilgerschrittbetrieb. Der MB_PrintLengthCorrCalcType01 übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunkten für die Technologiefunktion Drucklängenkorrektur, um die Länge eines Druckes auf dem Druckgut (Warenbahn bzw. Bogen) zu verändern.

ML_TechMotion.library

6.10.2 MB_PilgrimStepCalcType01

Kurzbeschreibung Der Funktionsbaustein berechnet eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen sowie die Hübe des Vorwärts- und Rückwärtstransportes sowie die Stoppositionen für den Pilgerschrittbetrieb. Er berücksichtigt dabei, dass der synchrone Bereich des Druckzylinders größer sein kann, als die zu druckende Formatlänge. Beispielsweise ist bei Offsetdruckwerken die Plattenlänge durch die mechanische Einspannvorrichtung fest vorgegeben und damit immer konstant. Das auf der Druckplatte befindliche Druckformat kann jedoch beliebig kürzer sein.

Durch die vom Baustein berechnete Kurvenscheibe werden die Infeed- und Outfeed-Walze der Druckmaschine betrieben.

Zuordnung Zielsystem/Bibliothek	IndraMotion MLC/IndraLogic XLC 12VRS	ML_TechMotion.compiled-library

Abb.6-147: Referenztabelle MB_PilgrimStepCalcType01

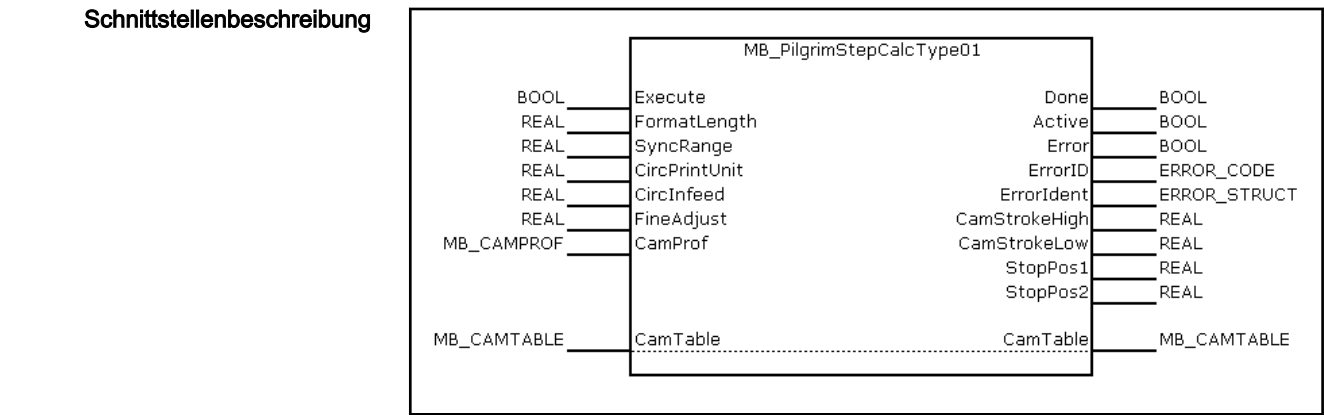


Abb.6-148: Funktionsbaustein MB_PilgrimStepCalcType01

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellen-Elementen und Information über die Anzahl der Elemente
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	FormatLength	REAL	Länge des zu druckenden Formates [mm]
	SyncRange	REAL	Länge des synchronen Bereichs [mm]
	CircPrintUnit	REAL	Umfang des Druckzylinders [mm]
	CircInfeed	REAL	Umfang der Zugwalzen [mm]
	FineAdjust	REAL	Feinabgleich [%] – definiert die gewünschte Bahnspannung
	CamProf	MB_CAMPROF	Auswahl des Kurvenscheibenprofils

E/A-Typ	Name	Typ	Kommentar
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe "Fehlerbehandlung" auf Seite 289)
	CamStrokeHigh	REAL	Hub P-0-0073 Outfeed / P-0-0093 Infeed Der P-0-0093 wird über den Funktionsbaustein MC_CamIn / Eingang CamShaftDistance vorgegeben Der P-0-0073 wird über den Funktionsbaustein MB_WriteRealParameter vorgegeben
	CamStrokeLow	REAL	Hub P-0-0073 Infeed / P-0-0093 Outfeed Der P-0-0093 wird über den Funktionsbaustein MC_CamIn / Eingang CamShaftDistance vorgegeben Der P-0-0073 wird über den Funktionsbaustein MB_WriteRealParameter vorgegeben Verwendete Beschleunigung in Relation zum Eingang 'MaxAcc' in [%]
	StopPos1	REAL	Umkehrpunkt 1 [°] mit Geschwindigkeit = 0
	StopPos2	REAL	Umkehrpunkt 2 [°] mit Geschwindigkeit = 0

Abb.6-149: Schnittstellenvariablen Funktionsbaustein MB_PilgrimStepCalcType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
FormatLength	REAL	>0.0	n.def.	100.0	Steigende Flanke an Execute
SyncRange	REAL	>0.0	n.def.	100.0	Steigende Flanke an Execute
CircPrintUnit	REAL	>0.0	n.def.	100.0	Steigende Flanke an Execute
CircInfeed	REAL	>0.0	n.def.	100.0	Steigende Flanke an Execute
FineAdjust	REAL	0.0	100.0	1.0	Steigende Flanke an Execute
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Steigende Flanke an Execute

Abb.6-150: Min-/Max- und Default-Werte Funktionsbaustein MB_PilgrimStepCalcType01

Die Enumeration [Kap. 5.14.3 "MB_CAMPROF" auf Seite 163](#) wird verwendet um zwischen verschiedenen Kurvenscheibenprofilen zu wählen.



Bei der Auswahl des Kurvenscheibenprofils "CAMPROF_STANDARD" muss der Linearpfad deaktiviert sein, da es sich um ein Profil ohne Untersetzung handelt. Während bei den Profilen "CAMPROF_ADDITIVE" und "CAMPROF_ADDITIVE_SEGMENT" der Linearpfad über ein Häkchen aktiviert werden muss und die Reduktion auf den Wert "1" zu stellen ist, da es sich um Profile mit Untersetzung handelt.

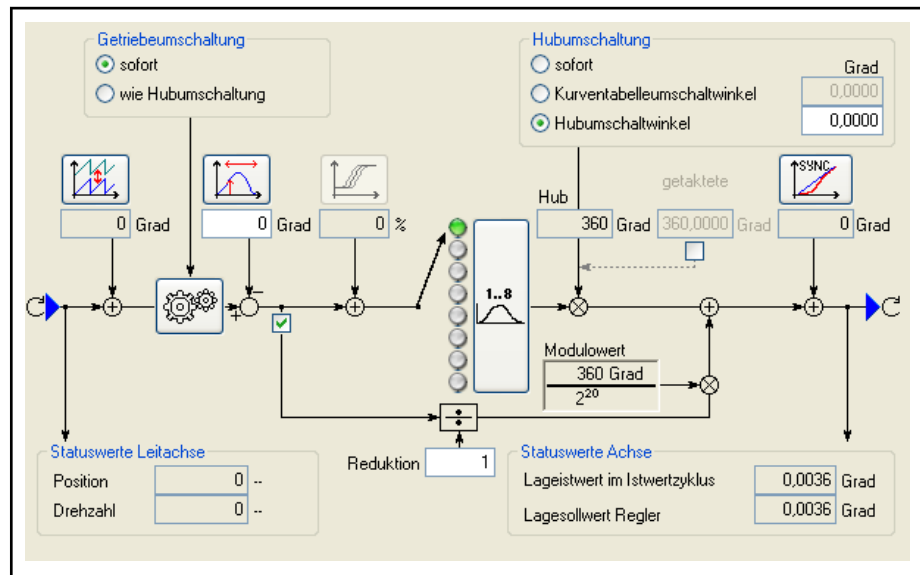


Abb.6-151: Aktivierung des Linearpfades

Die Struktur [Kap. 5.14.4 "MB_CAMTABLE" auf Seite 163](#) besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Funktionsbeschreibung

Wird nicht der gesamte Umfang eines Druckzylinders mit einer Druckplatte bzw. Druckklischee bedeckt, wechseln sich im kontinuierlichen Betrieb bedruckte mit nicht bedruckten Stellen ab. Um diese Druck-Lücken zu vermeiden wird die Materialbahn durch die Transportwalzen abgebremst, einen definierten Weg zurückgezogen und erneut auf synchrone Geschwindigkeit beschleunigt, so dass sich der neue Druck sofort nach dem alten anschließt.

Diese Ausgleichsbewegung wird durch den Funktionsbaustein berechnet und ist durch eine Stetigkeit bis zur Beschleunigung charakterisiert.

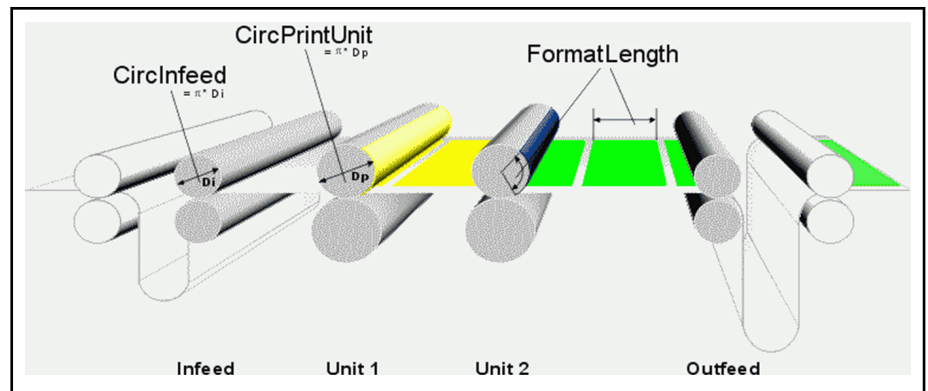


Abb.6-152: Prinzipdarstellung Pilgerschritt

Voraussetzungen zur Nutzung des Funktionsbausteins

- Der Funktionsbaustein ist nur für Achsen mit Interpolation auf Antrieb nutzbar
- Die Antriebsfunktion „Getaktete Zugwalze“ (siehe Antriebsdokumentation P-0-0086) muss vom Anwender eingeschaltet werden

Aus den mechanischen Angaben des Umfangs der Infeed- bzw. Outfeedwalze "CircInfeed", des Umfangs des Druckzylinders "CircPrintUnit", der Klischeelänge "FormatLength" auf dem Druckzylinder und der Gesamtlänge des Synchronbereichs "SyncRange" berechnet der Funktionsbaustein MB_PilgrimStepCalcType01 nach der Bearbeitungsfreigabe mittels "Execute" einmalig eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen, sowie die beiden Hubfaktoren "CamStrokeHigh" und "CamStrokeLow" und die Stoppositionen "StopPos1" und "StopPos2". Hierbei wird die Anzahl der Stützpunktelemente einer Kurvenscheibentabelle durch das Element "NumberOfElements" der Struktur "CamTable" eingestellt.

Es wird davon ausgegangen, dass der Umfang der Infeedwalze und der Umfang der Outfeedwalze identisch sind (CircInfeed).

Entsprechend der durch den Feinabgleich "FineAdjust" vorgegebenen Voreilung werden 2 Hübe berechnet, die dazu dienen, die Bahnspannung konstant zu halten. Der jeweils größere Hub ist für die in Transportrichtung vordere Walze bestimmt. Zur Berechnung der Hübe verwendet der Baustein folgende Formeln:

$$CamStrokeLow = 360^\circ * \frac{FormatLength}{CircInfeed}$$

Der Wert von CamStrokeLow gilt für den Parameter P-0-0073 der Einzugswalze und für den Parameter P-0-0093 der Auszugswalze

Abb.6-153: CamStrokeLow Formel

$$CamStrokeHigh = 360^\circ * \frac{FormatLength}{CircInfeed} * \left(1 + \frac{FineAdjust}{100} \right)$$

Der Wert von CamStrokeHigh gilt für den Parameter P-0-0073 der Auszugswalze und für den Parameter P-0-0093 der Einzugswalze

Abb.6-154: CamStrokeHigh Formel

Die Ausgänge "StopPos1" und "StopPos2" enthalten die Leitachspeditionen, bei denen die Kurvenscheibe einen Umkehrpunkt hat, um z.B. an einer dieser Positionen während einer Regelung einen Formatwechsel durchzuführen oder per Registerregler den Hub zu ändern.

Die zu berechnende Kurvenscheibentabelle wird in der Struktur "CamTable" mit Wichtung in Prozent abgelegt. Die Gültigkeit der Kurvenscheibenelemente wird durch den Ausgang "Done" signalisiert. Die Kurvenscheibentabelle kann mittels des FB [Kap. 5.14.2 "MB_LoadCamData" auf Seite 160](#) in eine Kurvenscheibe in der Steuerung oder in den Antrieb geladen werden.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" sowie die Ausgänge "CamStrokeHigh", "CamStrokeLow", "StopPos1" und "StopPos2" werden im Fehlerfall nicht aktualisiert.

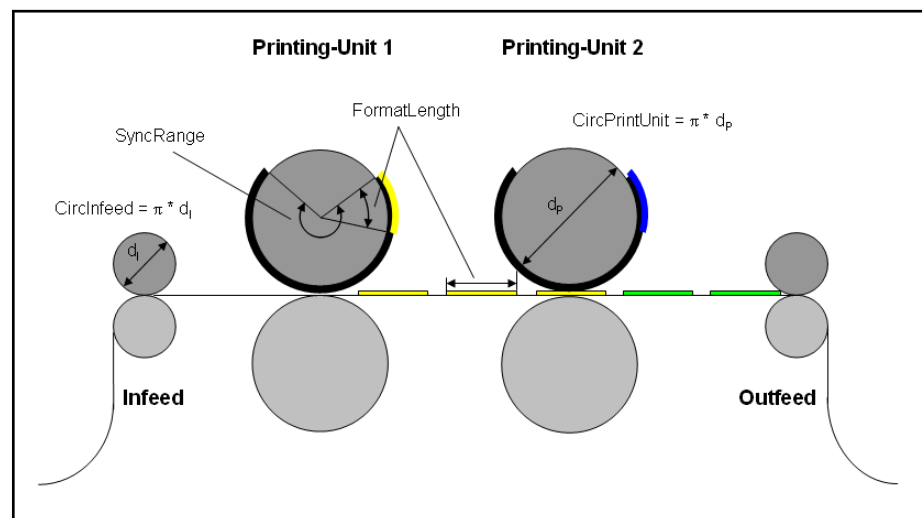


Abb.6-155: Prozess-Schema ($\text{FormatLength} < \text{SyncRange}$)

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins MB_PilgrimStepCalc-Typ01 weist folgende Eigenschaften auf:

- Der Funktionsbaustein basiert auf dem Funktionsbaustein MSV_CamTableType04 und deckt zusätzlich Fälle ab, bei denen der Synchronbereich des Druckzylinders größer sein muss, als der Druckbereich
- Die durch die Kurvenscheibentabelle charakterisierte Ausgleichsbewegung zum Rücktransport der Bahn ist stetig bis zur Beschleunigung (s , v , a) und weist einen Sprung im Ruck (j) auf
- Die Ausgleichsbewegung beginnt beim ersten Tabellenelement, nachfolgend schließt sich eine lineare Steigung winkelsynchron zum Druckzylinder an
- Der Algorithmus zur Berechnung der Kurvenscheibenstützpunkte ist auf mehrere Zyklen, d.h. Aufrufe der Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden
- Die berechneten Hübe des Funktionsbausteins weisen folgende Eigenschaften auf:
 - Die Hübe sind abhängig vom Druckformat und dem Umfang der Infeedwalze

- Der Hub CamStrokeHigh ist für den Parameter P-0-0093 der Infeedwalze und den Parameter P-0-0073 der Outfeedwalze, der Hub CamStrokeLow für den Parameter P-0-0073 der Infeedwalze und den Parameter P-0-0093 der Outfeedwalze vorgesehen

Zur Erzeugung der Bahnspannung sind der Hub für den Vortransport der Outfeedwalze und der Hub für den Rücktransport der Infeedwalze um den Feinabgleich größer als der Hub für den Vortransport der Infeedwalze und den Hub für den Rücktransport der Outfeedwalze.

Es werden drei Arten von Kurvenscheiben unterschieden:

- **FormatLength = SyncRange = CircPrintUnit**

Die Druckplatte bzw. das Druckklischee bedeckt die gesamte synchrone Zone des Zylinders und die synchrone Zone erstreckt sich über den kompletten Umfang des Druckzylinders. In diesem Fall findet keine Ausgleichsbewegung statt. SyncRange gleich CircPrintUnit, muss auf jeden Fall auch FormatLength diesen Wert haben, da keine Ausgleichsbewegung durchgeführt werden kann. Ist FormatLength in diesem Fall kleiner als SyncRange, bricht der Funktionsbaustein die Bearbeitung mit einem Fehler ab.

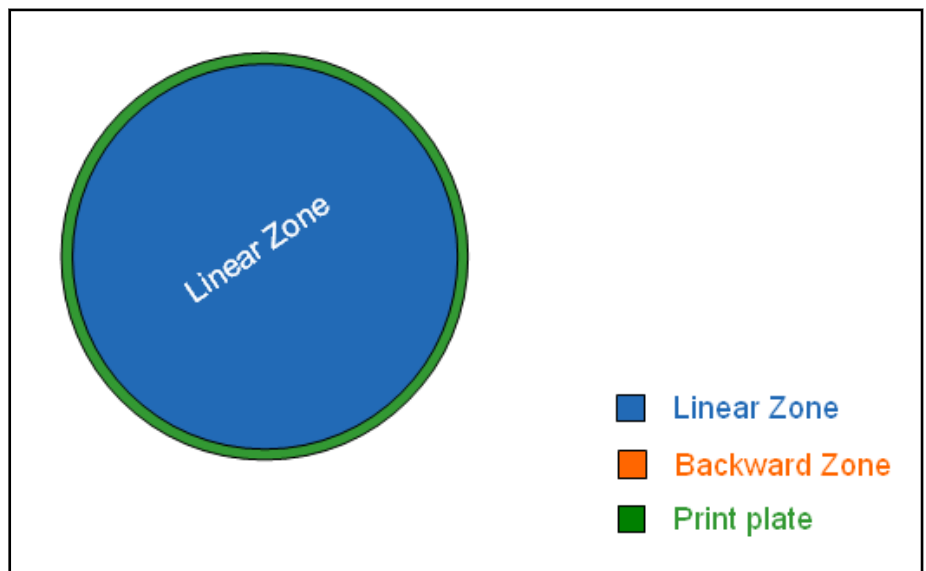
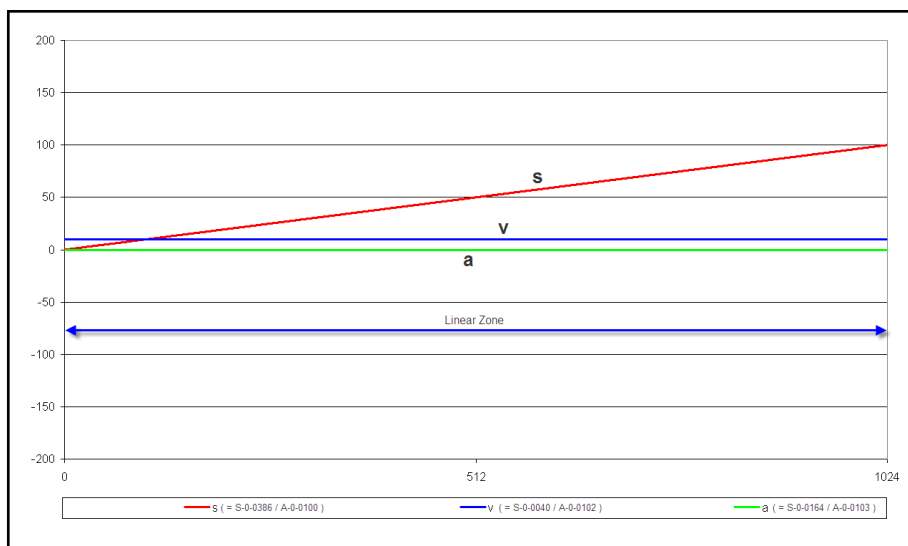


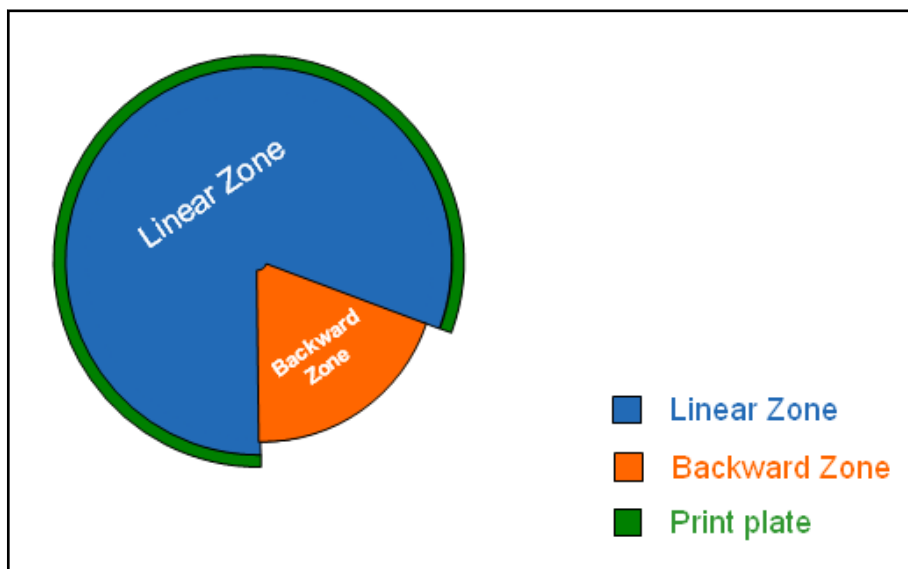
Abb.6-156: MB_PilgrimStepCalcType01 mit FormatLength = SyncRange = CircPrintUnit

ML_TechMotion.library



- **FormatLength = SyncRange**

Die Druckplatte bzw. das Druckklischee bedeckt die gesamte synchrone Zone des Zylinders. Der gesamte restliche Zylinderbereich kann für die Ausgleichsbewegung genutzt werden.



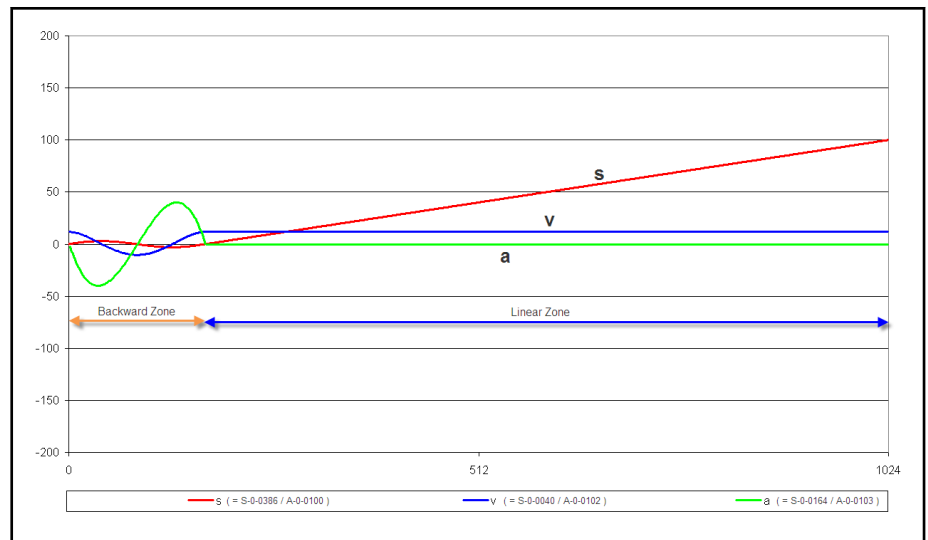


Abb. 6-159: MB_PilgrimStepCalcType01 Bewegungsprofil

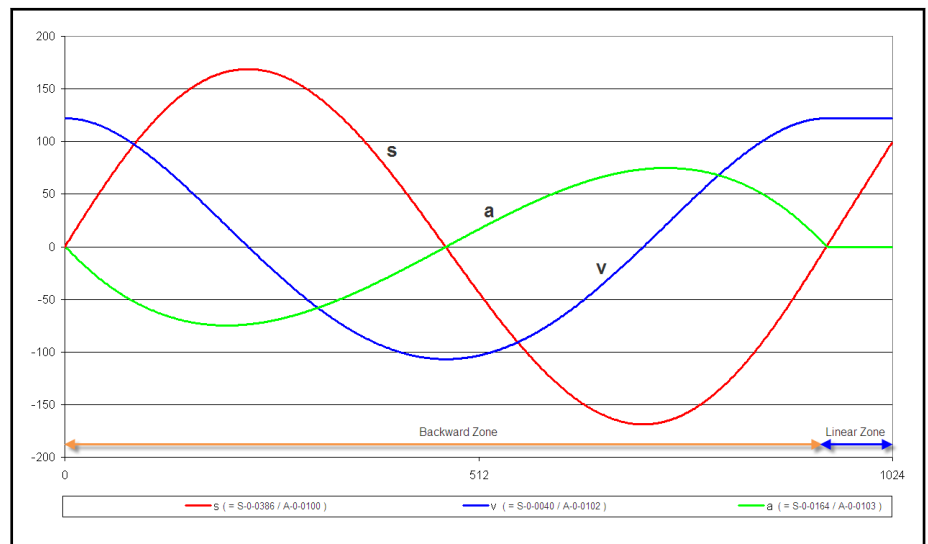


Abb. 6-160: MB_PilgrimStepCalcType01 Bewegungsprofil

- **FormatLength < SyncRange**

Die Druckplatte bzw. das Druckklischee bedeckt nur teilweise die synchrone Zone des Zylinders. Dieser muss sich bis zum Ende des synchronen Bereichs weiterhin mit synchroner Geschwindigkeit bewegen. Erst wenn der Zylinder keinen Bahnkontakt mehr hat, kann die Materialbahn abgebremst, zurückgezogen und erneut beschleunigt werden. Hier muss auch der zusätzliche Weg des nichtdruckenden synchronen Bereichs zurückgezogen werden.

ML_TechMotion.library

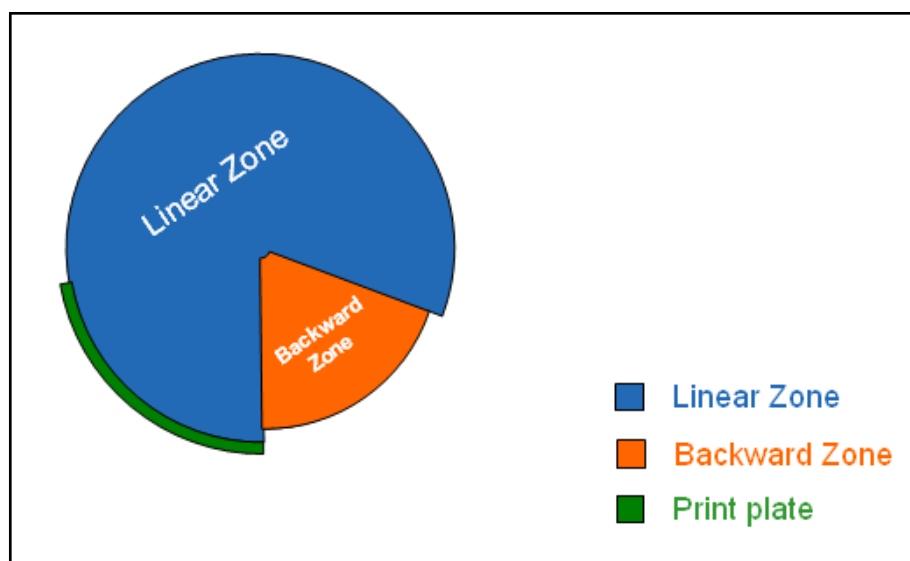
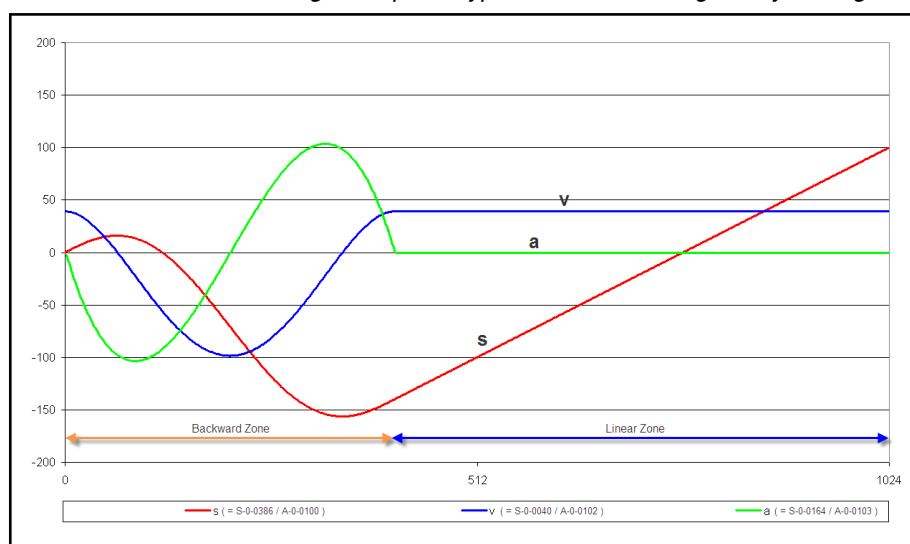
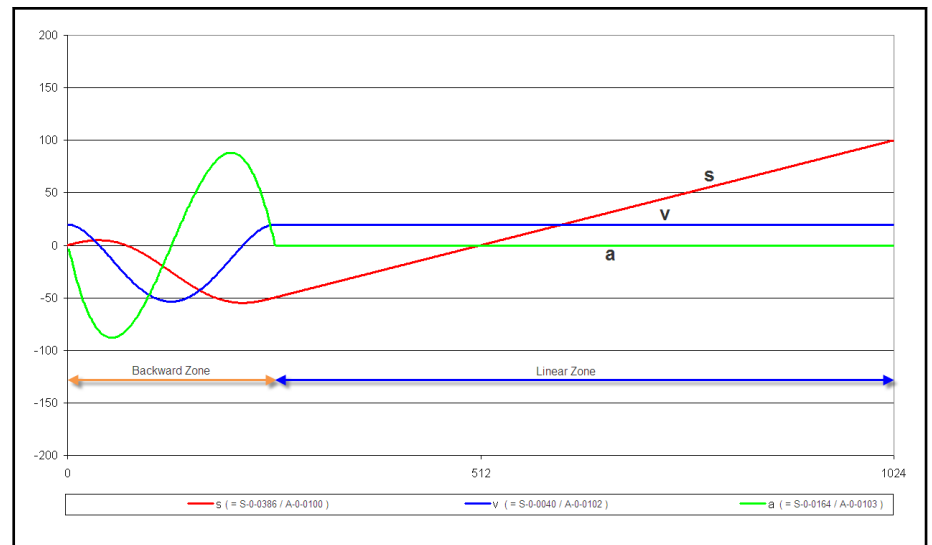


Abb.6-161: MB_PilgrimStepCalcType01 mit FormatLength < SyncRange



FormatLength = 500.0; SyncRange = 1200.0; CircPrintUnit
= 2000.0; CircInFeed = 3000.0

Abb.6-162: MB_PilgrimStepCalcType01 Bewegungsprofil



Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170x. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR	16#00002200	16#00000001	Eingang NumberOfElements ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR	16#00002260	16#00000001	Eingang FormatLength ist kleiner als Minimum
INPUT_RANGE_ERROR	16#00002260	16#00000002	Eingang SyncRange ist kleiner als Minimum
INPUT_RANGE_ERROR	16#00002260	16#00000003	Eingang CircPrintUnit ist kleiner als Minimum
INPUT_RANGE_ERROR	16#00002260	16#00000004	Eingang CircInfeed ist kleiner als Minimum
INPUT_RANGE_ERROR	16#00002260	16#00000005	Eingang FineAdjust ist kleiner als Minimum oder größer als Maximum
INPUT_INVALID_ERROR	16#00002261	16#00000001	Adresse von Feld CamTable ist null
INPUT_INVALID_ERROR	16#00002261	16#00000001	Eingang FormatLength ist größer als CircPrintUnit
INPUT_INVALID_ERROR	16#00002261	16#00000002	Eingang SyncRange ist größer als CircPrintUnit
INPUT_INVALID_ERROR	16#00002261	16#00000003	Eingang FormatLength ist größer als SyncRange
INPUT_INVALID_ERROR	16#00002261	16#00000004	SyncRange = CircPrintUnit aber FormatLength != SyncRange (kein Platz für Ausgleichsbewegung)

Abb.6-164: Fehlercodes Funktionsbaustein MB_PilgrimStepCalcType01

ML_TechMotion.library

6.10.3 MB_PrintLengthCorrCalcType01

Kurzbeschreibung	Der Funktionsbaustein berechnet eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen für die Technologiefunktion Drucklängenkorrektur.	
Zuordnung Zielsystem/Bibliothek	IndraMotion MLC 12VRS	ML_TechMotion.compiled-library

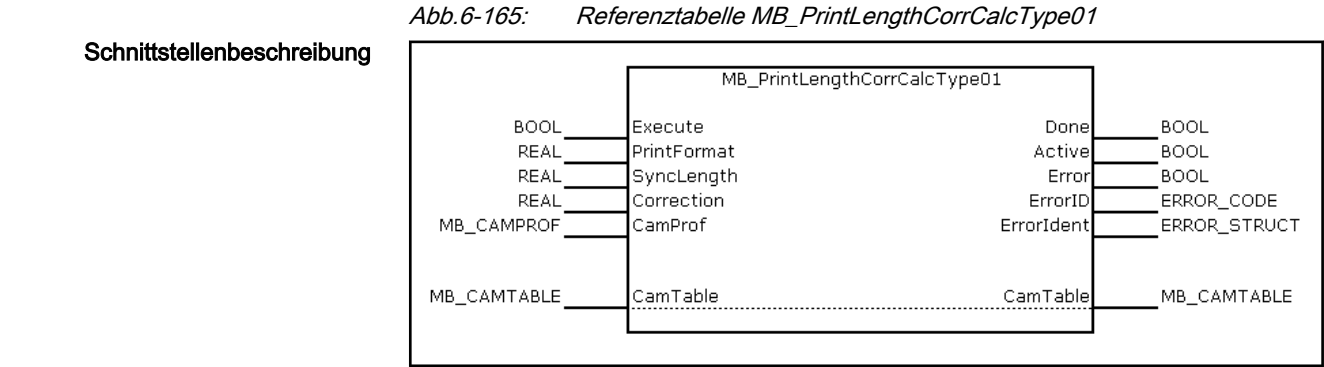


Abb.6-166: Funktionsbaustein MB_PrintLengthCorrCalcType01

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellen-Elementen und Information über die Anzahl der Elemente
	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	PrintFormat	REAL	Erwartetes, nominelles Druckformat [mm]
	SyncLength	REAL	Synchrone Formatlänge, d.h. Umfang des Druckzylinders [mm]
	Correction	REAL	Druckkorrektur zur Kompensation der Abweichung [mm]
VAR_OUTPUT	CamProf	MB_CAMPROF	Auswahl des Kurvenscheibenprofils
	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe "Fehlerbehandlung" auf Seite 294)

Abb.6-167: Schnittstellenvariablen Funktionsbaustein MB_PrintLengthCorrCalcType01

Min-/Max- und Default-Werte	In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
PrintFormat	REAL	>0.0	n.def	1000.0	Steigende Flanke an Execute
SyncLength	REAL	>0.0	n.def	1500.0	Steigende Flanke an Execute
Correction	REAL	-9999.9	9999.9	100.0	Steigende Flanke an Execute
CamProf	MB_CAMPF			CAM_PROF_STANDARD	Steigende Flanke an Execute

Abb.6-168: Min-/Max- und Default-Werte Funktionsbaustein MB_PrintLengthCorrection

Die Enumeration [Kap. 5.14.3 "MB_CAMPF"](#) auf Seite 163 wird verwendet um zwischen verschiedenen Kurvenscheibenprofilen zu wählen.



Bei der Auswahl des Kurvenscheibenprofils "CAMF_STANDARD" muss der Linearpfad deaktiviert sein, da es sich um ein Profil ohne Untersezung handelt. Während bei den Profilen "CAMF_ADDITIVE" und "CAMF_ADDITIVE_SEGMENT" der Linearpfad über ein Häkchen aktiviert werden muss und die Reduktion auf den Wert "1" zu stellen ist, da es sich um Profile mit Untersezung handelt.

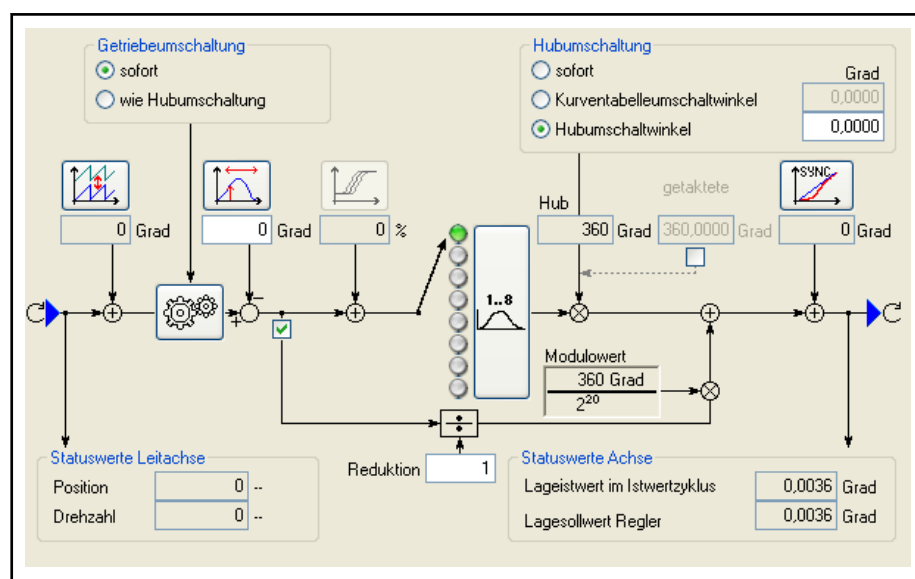


Abb.6-169: Aktivierung des Linearpfades

Die Struktur [Kap. 5.14.4 "MB_CAMTABLE"](#) auf Seite 163 besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

ML_TechMotion.library

Funktionsbeschreibung

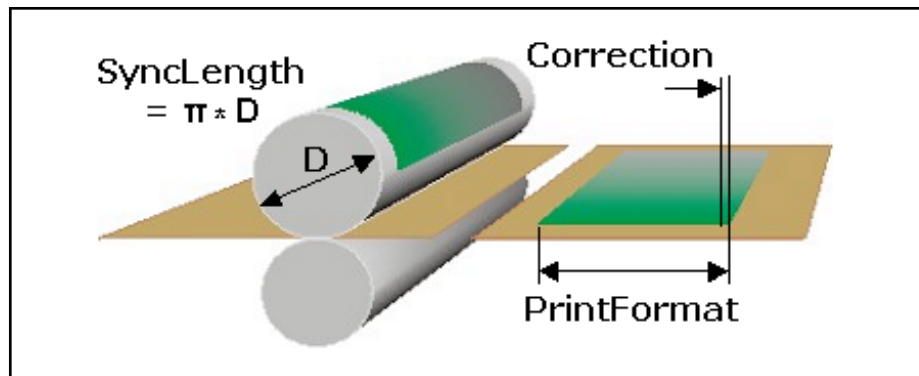


Abb.6-170: Prinzipdarstellung der Drucklängenkorrektur

Die Drucklängenkorrektur dient dazu, die Länge eines Druckes auf dem Druckgut (Warenbahn bzw. Bogen) zu verändern. Hierzu wird die Geschwindigkeit der Druckwalze zum Druckgut während des Druckkontaktes der Druckwalze vergrößert bzw. verringert. Innerhalb der darauffolgenden druckfreien Zone wird die Position des Druckzylinders derart korrigiert, dass der nächste Druck wieder registerhaltig begonnen wird.

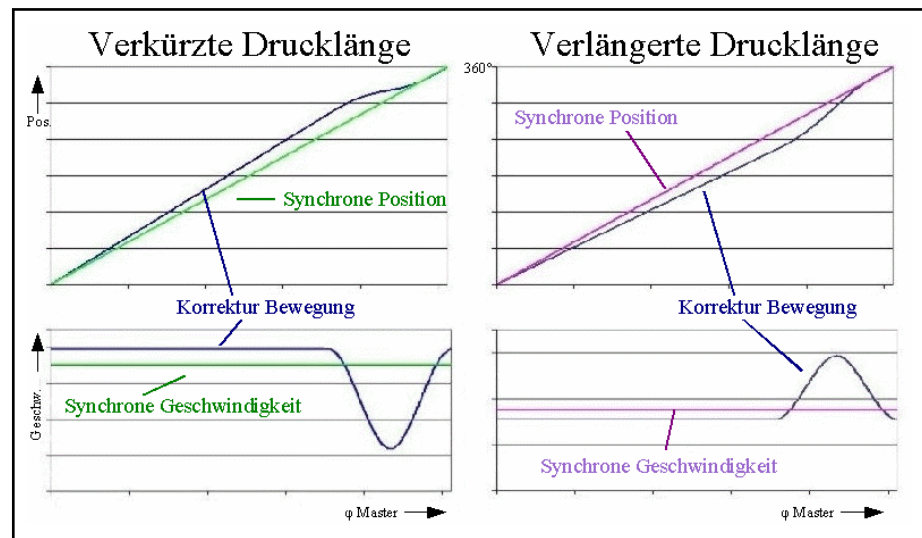


Abb.6-171: Profile der Drucklängenkorrektur

Mit der Vorgabe der Formate "PrintFormat" sowie "SyncLength" und der Korrektur "Correction" berechnet der Funktionsbaustein "MB_PrintLengthCorr- CalcType01" nach der Bearbeitungsfreigabe mittels "Execute" einmalig auf Basis der Eingangsgrößen eine Kurvenscheibentabelle mit bis zu 1024 Stützpunkten. Hierbei wird die Anzahl der Stützpunktelemente einer Kurvenscheibentabelle durch das Element "NumberOfElements" der Struktur "CamTable" eingestellt.

Die Summe aus Druckformat und Druckkorrektur darf den Umfang der Druckwalze nicht überschreiten. Die notwendigen positiven und negativen Beschleunigungswerte steigen mit sinkender druckfreien Zone.

- Ein positiver Korrekturwert vergrößert den vom Druckzylinder zurückgelegten Weg bei gleicher Leitachseposition und verringert somit das Druckformat

Die zu berechnende Kurvenscheibentabelle wird in der Struktur "CamTable" mit Wichtung in Prozent abgelegt. Die Gültigkeit der Kurvenscheibenelemente wird durch den Ausgang "Done" signalisiert. Die Kurvenscheibentabelle

kann mittels des FB [Kap. 5.14.2 "MB_LoadCamData" auf Seite 160](#) in eine Kurvenscheibe in der Steuerung oder in den Antrieb geladen werden.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" werden im Fehlerfall nicht aktualisiert.

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins MB_PrintLengthCorrCalcType01 weist folgende Eigenschaften auf:

- Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist stetig bis zur Beschleunigung (s,v,a) und weist einen Sprung im Ruck (j) auf
- Die Drucklängenkorrektur, d.h. die Verlängerung bzw. Verkürzung des Drucklängenformates basiert auf einer Vergrößerung bzw. Verringerung der Geschwindigkeit des Druckzylinders während des Druckes und abschließenden Ausgleichsbewegung
- Das Druckformat, d.h. das Klischee, darf inkl. der Druckkorrektur nur einen Teil der synchronen Formatlänge betragen, d.h. zwischen zwei Drucken muss eine druckfreie Zone bestehen, welche für die Ausgleichsbewegung genutzt wird
- Der Algorithmus zur Berechnung der Kurvenscheibenstützpunkte ist auf mehrere Zyklen, d.h. Aufrufe der FB-Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden

Bewegungsprofile mit verschiedenen Belegungen des Eingangs
Correction

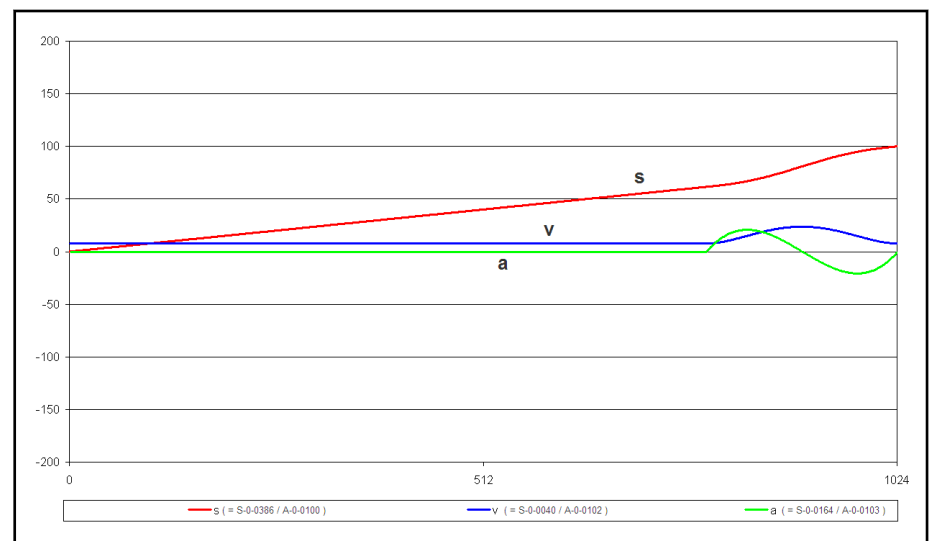
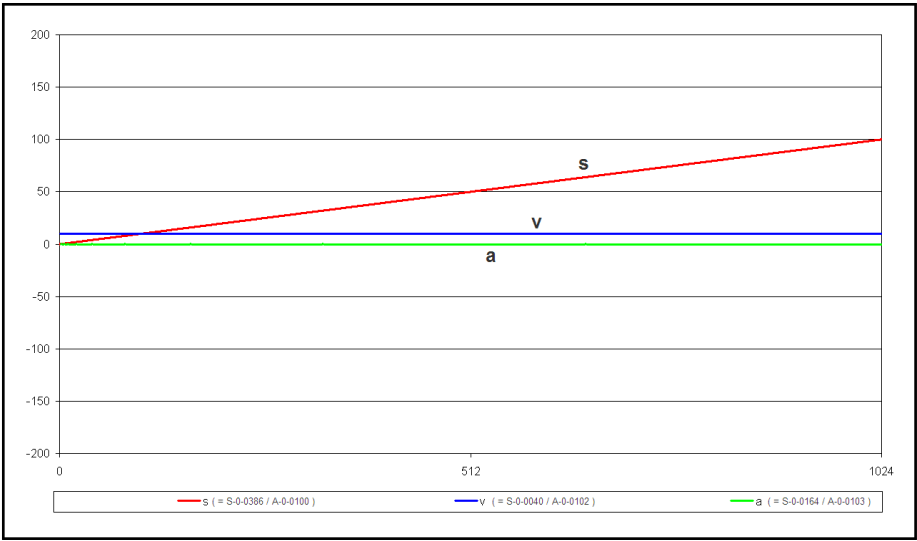
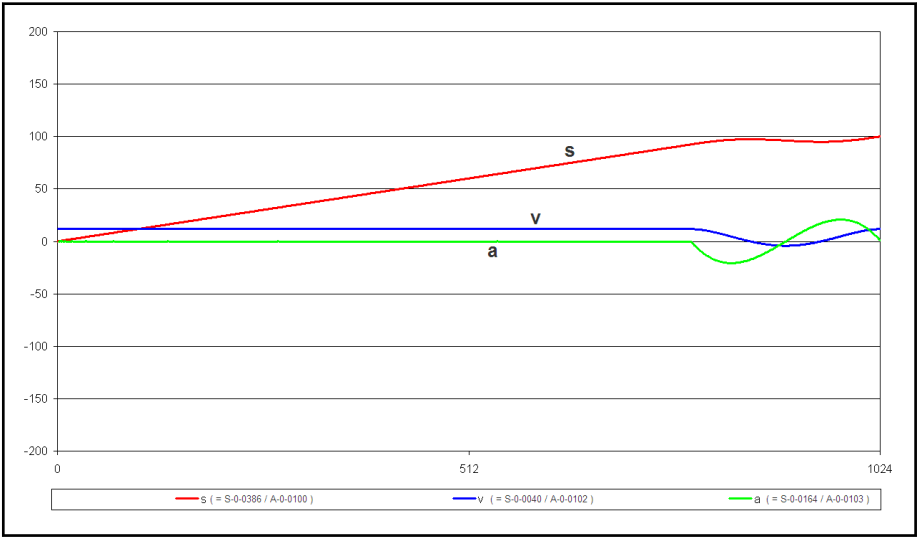


Abb.6-172: SyncLength: 1.3 * PrintLength; Correction: - 200
MB_PrintLengthCorrCalcType01 Bewegungsprofil

ML_TechMotion.library



SyncLength: 1.3 * PrintLength; Correction: 0
 Abb. 6-173: MB_PrintLengthCorrCalcType01 Bewegungsprofil



SyncLength: 1.3 * PrintLength; Correction: 200
 Abb. 6-174: MB_PrintLengthCorrCalcType01 Bewegungsprofil

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16x0170x. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR	16#00002200	16#00000001	Eingang NumberOfElements ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR	16#00002240	16#00000001	Eingang "PrintFormat" ist kleiner als Minimum
INPUT_RANGE_ERROR	16#00002240	16#00000002	Eingang "SyncLength" ist kleiner als Minimum
INPUT_RANGE_ERROR	16#00002240	16#00000003	Eingang "Correction" ist kleiner als Minimum oder größer als Maximum

ML_TechMotion.library

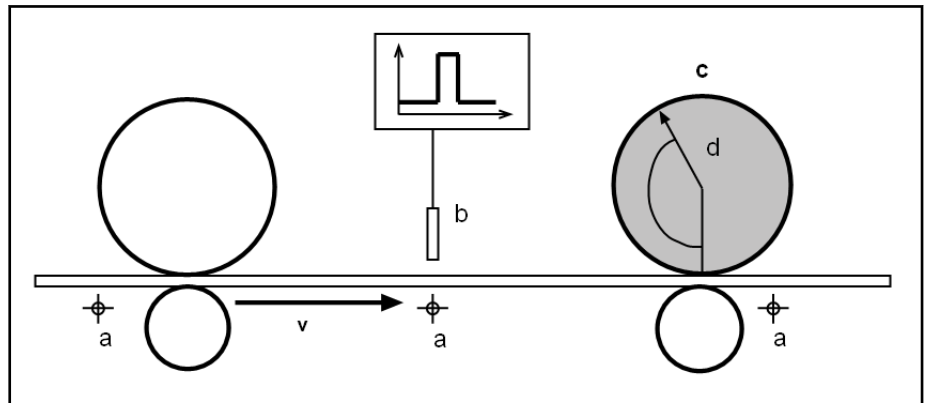
ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#00002201	16#00000001	Adresse von Feld "CamTable" ist null
INPUT_INVALID_ERROR	16#00002241	16#00000001	Ungültige Berechnung: (PrintFormat + Correction) > SyncLength oder SyncLength <= PrintFormat

Abb.6-175: Fehlercodes Funktionsbaustein MB_PrintLengthCorrCalcType01

7 ML_TechRegi.library/MX_TechRegi.lib

7.1 Einführung und Übersicht

Auf Papierbahnen oder Papierbogen gedruckte Registermarken werden durch einen digitalen Eingang erkannt und anschließend von einem Registerregler als Lagemesswert erkannt (Lagemessung). Die Positionsversatz, die Geschwindigkeit des Antriebs oder die Leitachsparameter werden durch die Registerregler-FB angepasst.



- a: Registermarke
- b: Registermesstaster
- c: Druckzylinder
- d: ϕ drucken

Abb. 7-1: Messprinzip mit dem Papierbahndrucken

Der Registerregler regelt die Lage des Materials in Bezug auf die Lage einer Bezugsachse. Diese kann die Lage des Bogens in Bezug auf den Druckzylinder sein. Die Regelung wird mit Hilfe eines elektrischen Eingangssignals an der Bezugsachse durchgeführt. Das elektrische Eingangssignal wird durch das optische Messen einer Registermarke auf dem Material zur Verfügung gestellt. Die folgenden Funktionsbausteine werden unterstützt:

Funktionsbaustein	ML_TechRegi für IndraMotion MLC	MX_TechRegi für IndraMotion MLD
MB_RegisterControllerType02	X	X
MB_RegisterControllerType04	X	X
MB_RegisterControllerSideType01	X	X
MB_RegiMeasuringType01	X	X
MB_RegiMeasuringType02	X	X
MB_RegiMeasuringType03	X	X
MB_RegiRegulateType01	X	X
MB_RegiRegulateType02	X	X
MB_RegiRegulateType03	X	X
MB_RegiRegulateType04	X	

ML_TechRegi.library/MX_TechRegi.lib

Funktionsbaustein	ML_TechRegi für IndraMotion MLC	MX_TechRegi für IndraMotion MLD
MB_RegiSetpointLockType01	X	X
IL_RegiQualityType01	X	X

Abb. 7-2: Überblick über Funktionsbausteinunterstützung durch die Bibliothek

7.2 Registerregler-FB

7.2.1 Überblick

Dieser Abschnitt beschreibt die Registerregler-FB für die Bibliotheken ML_TechRegi.compiled-library (IndraMotion MLC) und MX_TechRegi.lib (IndraMotion MLD).

7.2.2 MB_RegisterControllerType02

Kurzbeschreibung Der Registerregler-FB MB_RegisterControllerType02 unterstützt die folgende Funktionalität:

- Berechnung eines Korrekturwerts auf Grundlage eines gemessenen Werts und eines Sollwerts unter Verwendung eines P- oder PI-Regelkreises
- Preset-Funktion
- Pause-Funktion
- Min/Max-Begrenzung des berechneten Stellwerts
- Korrekturfenster-Funktion
- Sollwert übernehmen

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechRegi.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechRegi.lib

Abb. 7-3: Referenztable MB_RegisterControllerType02

Schnittstellenbeschreibung

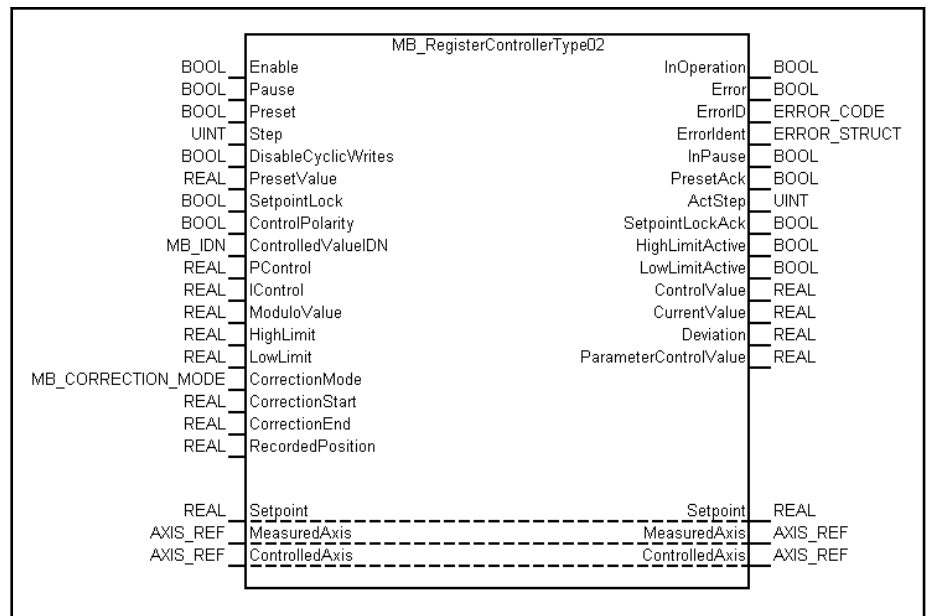


Abb.7-4: Funktionsbaustein MB_RegisterControllerType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Setpoint	REAL	Sollwert. Dieser Wert wird mit "RecordedPosition" verglichen
	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse
	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse
VAR_INPUT	Enable	BOOL	Aktiviert den Registerregler
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist. Die Ausgänge "ControlValue" und "ParameterControlValue" sind konstant, solange "Pause" aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt den Regelkreis zwangsweise auf "PresetValue". Zusätzlich setzt der "Preset"-Eingang "ParameterControlValue" auf den "PresetValue"-Wert
	Step	UINT	Verarbeitungsschritt
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlledValueIDN" wenn TRUE
	PresetValue	REAL	Wert an die Ausgänge "ControlValue" und "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	SetpointLock	BOOL	Aktiviert die Sollwertübernahme. Wenn TRUE, wird die nächste gemessene Marke gelatcht
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist

ML_TechRegi.library/MX_TechRegi.lib

E/A-Typ	Name	Datentyp	Beschreibung
	ControlledValueIDN	MB_IDN	SERCOS-IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: IndraMotion MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 IndraMotion MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	PControl	REAL	Proportionalverstärkung des PI-Reglers
	IControl	REAL	Nachstellzeit T_n des PI-Reglers
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
	HighLimit	REAL	Maximalwert für "ControlValue"
	LowLimit	REAL	Minimalwert für "ControlValue"
	CorrectionMode	MB_CORRECTION_MODE	CORR_ABS = absolutes Korrekturfenster [CorrectionStart .. CorrectionEnd]. CORR_REL = relatives Korrekturfenster [Anfang = "RecordedPosition" + "CorrectionStart" .. Ende = "RecordedPosition" + "CorrectionEnd"] CORR_REL_ABS = relatives Anfangs... absolutes Endkorrekturfenster [Anfang = "RecordedPosition" + "CorrectionStart" .. Ende = "CorrectionEnd"] CORR_ABS_REL = absolutes Anfangskorrekturfenster... relatives Endkorrekturfenster [Anfang = "CorrectionStart" .. Ende = "RecordedPosition" + "CorrectionEnd"] CORR_WIN_DISABLED = Korrekturfenster ist deaktiviert
	CorrectionStart	REAL	Anfang des Korrekturfensters
	CorrectionEnd	REAL	Ende des Korrekturfensters
	RecordedPosition	REAL	Gemessener Wert der Achsposition
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	InPause	BOOL	"Pause" ist aktiv. Die Ausgänge "ControlValue" und "ParameterControlValue" behalten ihre Werte
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in die Ausgänge "ControlValue" und "ParameterControlValue" kopiert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ControlValue" hoch
	SetpointLockAck	BOOL	"Setpoint" übernommen. Der aktuelle Wert in "RecordedPosition" wird nach "Setpoint" kopiert

E/A-Typ	Name	Datentyp	Beschreibung
	HighLimitActive	BOOL	Obergrenze ist aktiv. "ControlValue" wäre größer, ist aber auf "HighLimit" beschränkt
	LowLimitActive	BOOL	Untergrenze ist aktiv. "ControlValue" wäre niedriger, ist aber auf "LowLimit" beschränkt
	ControlValue	REAL	Vom Registerregler errechneter Stellwert
	CurrentValue	REAL	Die aktuell aufgezeichnete Lage, die im Registerregler verwendet wird
	Deviation	REAL	"Deviation" = ("Setpoint" - "CurrentValue") des Registerreglers
	ParameterControlValue	REAL	Der Registerregler berechnet einen Wert basierend auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parametereingang "ControlledValueIDN" geschrieben wenn der Eingang "DisableCyclicWrites" = FALSE ist

Abb. 7-5: Schnittstellenvariablen Funktionsbaustein MB_RegisterController-Type02

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	"Enable" TRUE (kontinuierlich)
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
DisableCyclicWrites	BOOL			FALSE	Kontinuierlich
SetpointLock	BOOL			FALSE	Kontinuierlich
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
Setpoint	REAL	n.def	n.def	0.0	Änderung von "Step"
PControl	REAL	0.0	n.def	1.0	Kontinuierlich
IControl	REAL	0.0	n.def	0.0	Kontinuierlich
ModuloValue	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"
HighLimit	REAL	LowLimit	n.def	0.0	Kontinuierlich
LowLimit	REAL	n.def	HighLimit	0.0	Kontinuierlich

ML_TechRegi.library/MX_TechRegi.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
CorrectionMode	MB_CORRECTI- ON_MODE			CORR_REL	Kontinuierlich
CorrectionStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich
RecordedPosition	REAL	n.def	n.def	0.0	Änderung von "Step"

Abb. 7-6: Übersicht über Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Das nachfolgende Diagramm zeigt den Signal-Zeit-Verlauf des FB MB_RegisterControllerType02 einschließlich der Funktionen "Pause" und "Preset".

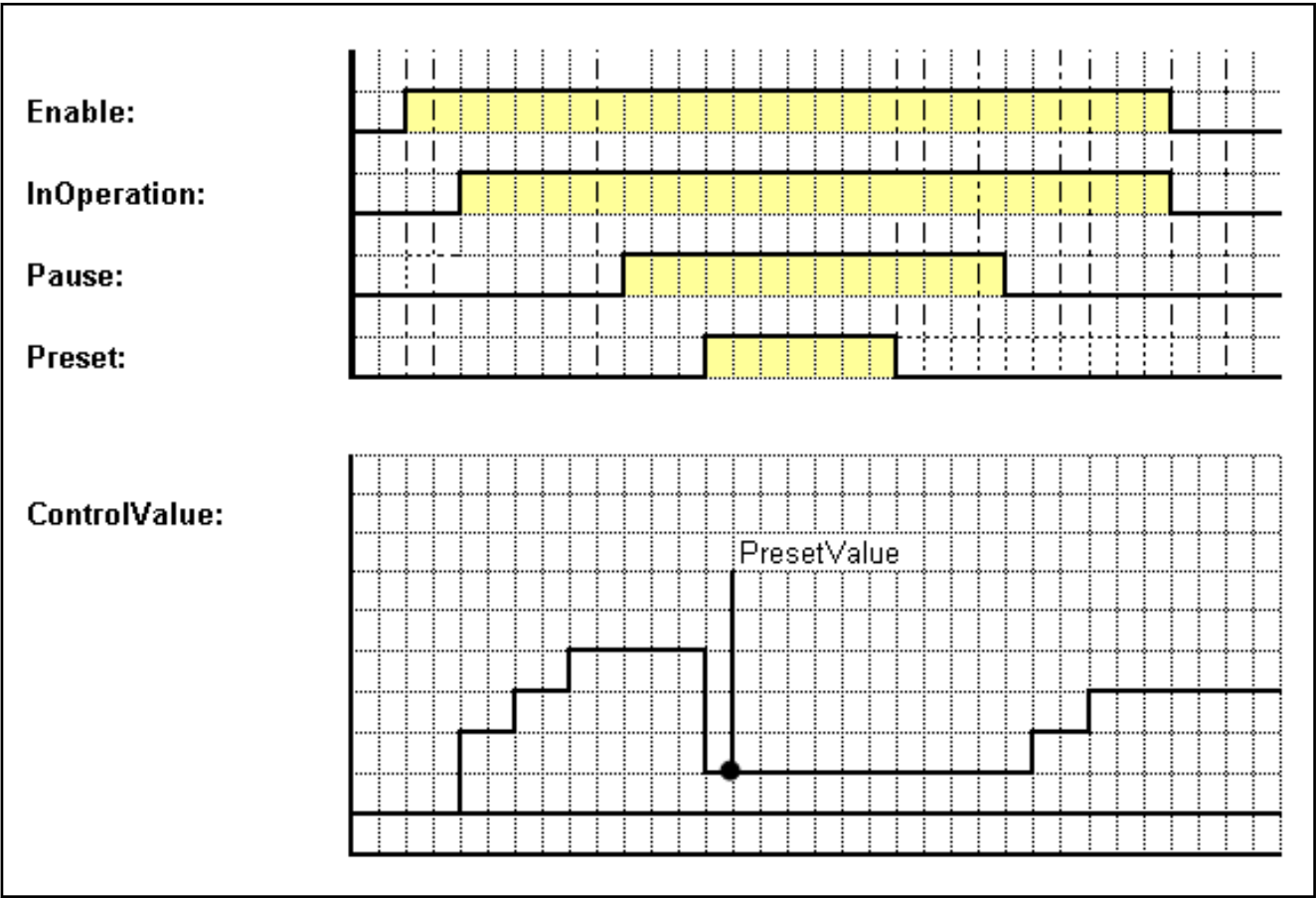


Abb. 7-7: Zeitdiagramm des FB MB_RegisterControllerType02 mit den Funktionen "Pause" und "Preset"

Funktionsbeschreibung Ein Sensor erkennt Marken, Perforationen, Schnitte oder einen geklebten Stoß auf dem Material und meldet dies dem Antrieb als (binäres) Signal. Die Messtasterfunktion des Antriebs bestimmt die Flanke des Sensorsignals und verriegelt, abhängig von der Signalauswahl des Messtasters, die folgenden Stellungen:

Antriebsistposition Parameter	S-0-0051, S-0-0053, P-0-0753
Leitachsisposition Parameter	P-0-0052, P-0-0227, P-0-0775, P-0-0776, P-0-0778

Die Messtasterfunktion zeichnet Positionsdaten mit einer Auflösung von 0,5µs auf. Der Sensor muss ein 24V-Signal mit einer Anstiegszeit im µs-Bereich bereitstellen. Die sensorspezifische Verzögerungszeit kann mithilfe der

Eingänge "DeadTimePosEdge" oder "DeadTimeNegEdge" des Funktionsbausteins MB_InitTouchProbe kompensiert werden.

Der Registerregler berechnet die Regelungsabweichung zwischen der gemessenen und der Sollwertposition und ermittelt den erforderlichen Korrekturwert, sobald eine neue Flanke des Sensorsignals ermittelt wird.

Änderungen des Korrekturwertes werden über das trapezförmiges Profil, die Geschwindigkeitsrampe, den PT1-Filter oder ohne Verzögerung aktiv, abhängig vom ausgewählten Regelungsparametereingang ("ControlledValueIDN") des Funktionsbausteins. Die nachstehende Tabelle führt die verfügbaren Regelungsparameter ("ControlledValueIDN") sowie das daraus resultierende MotionProfile aller verfügbaren Antriebsbetriebsarten mit Synchronisation auf.

Parametername	Controlled-ValueIDN	Bewegungsverhalten, wenn sich ein Parameterwert ändert	Verfügbar in den folgenden Betriebsmodus			
			Velocity-Synchronization	Phase-Synchronization	Cam Shaft	Electronic Motion Profile
Winkelverschiebung Anfang des Profils	P-0-0061 A-0-2730	Änderungen von P-0-0061 werden über Rampe mithilfe P-0-0158 (Geschwindigkeit) aktiv. Wenn P-0-0158=0, werden Änderungen an P-0-0061 sofort aktiv.	Nein	Nein	Ja	Ja
Geschwindigkeits-Sollwert additiv	P-0-0690 A-0-2615	Änderungen von P-0-0694 werden sofort aktiv	Ja	Nein	Nein	Nein
Lagesollwert additiv, Prozessregler	P-0-0691 A-0-2610	Wenn P-0-0155 Bit0 = 0: Änderungen von P-0-0691 werden über PT1-Filter mittels P-0-0060 (Filterzeitkonstante) aktiv. Wenn P-0-0060=0, werden Änderungen an P-0-0691 sofort aktiv. Wenn P-0-0155 Bit0 = 1: Änderungen von P-0-0691 werden sofort aktiv	Ja	Ja	Ja	Ja
Leitachspannung additiv, Prozessregler	P-0-0692 A-0-2600	Änderungen von P-0-0692 werden mit einem Filter 1. Ordnung geglättet bei Verwendung von P-0-0693 (Filterzeitkonstante) und dann zur Summe aus P-0-0054, Leitachspannung additiv, und P-0-0053, Leitachspannung, oder P-0-0052, Lageistwert Messgeber, addiert.	Ja	Ja	Ja	Ja
Feinabgleich Getriebeübersetzung, Prozessregler	P-0-0694 A-0-2605	Änderungen von P-0-0694 werden sofort aktiv	Ja	Ja	Ja	Ja
Winkelverschiebung Tabellenanfang, Prozessregler	P-0-0695 A-0-2620	Änderungen von P-0-0695 werden über PT1-Filter mittels P-0-0696 (Filterzeitkonstante) aktiv. Wenn P-0-0696=0, werden Änderungen an P-0-0695 sofort aktiv.	Nein	Nein	Ja	Ja

Abb. 7-8: Unterstützte Regelparameter und unterstütztes Verhalten

ML_TechRegi.library/MX_TechRegi.lib

Die folgenden "ControlValueIDN"-Werte werden integriert:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Initialisierung Während der Initialisierung müssen die Eingänge "Setpoint" und "Recorded-Position" über gültige Daten verfügen.

PresetValue Für einen fehlerfreien Start der Reglerausgänge "ControlValue" und "ParameterControlValue", muss der "Preset"-Eingang gesetzt sein, bevor der "Enable"-Eingang aktiviert wird. Während dieser Zeit müssen die Eingänge "Setpoint" und "RecordedPosition" über gültige Daten verfügen. Zudem werden die oberen und unteren Grenzwerte überwacht, um eine Begrenzung des Ausgangs "ControlValue" sicherzustellen, wenn der aktuelle Wert für "PresetValue" außerhalb des Gültigkeitsbereiches von "HighLimit" bzw. "Low-Limit" liegt.

Der "PresetAck"-Ausgang ist aktiv wenn der Wert am "PresetValue"-Eingang in die Ausgänge "ControlValue" und "ParameterControlValue" kopiert wird. Der "Preset"-Eingang hat eine höher Priorität als der "Pause"-Eingang. Der Wert in "PresetValue" wird in die entsprechenden Ausgänge bei steigender Flanke am "Preset"-Eingang kopiert wenn "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt.

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. In diesem Fall wird die Abweichung ("Setpoint" – "RecordedPosition") auf 0 gesetzt. Dadurch ändern sich die Ausgänge "ControlValue" und "ParameterControlValue" nicht mehr. Diese kann verwendet werden, um die Werte der Ausgänge "ControlValue" und "ParameterControlValue" beizubehalten. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Solange der Eingang "Pause" aktiv ist, ist der Eingang "InPause" TRUE.

Sollwertübernahme Einen Moduswechsel von Manuell zu Automatik wird durch das Einrasten beim nächsten erkannten Registermarker erreicht. Wenn zum Beispiel eine neue Papierbahn in Position gebracht und die Registermarke erkannt wird, wird der Eingang "SetpointLock" auf TRUE gesetzt. Der Wert im "Setpoint" wird mit dem Wert im "RecordedPosition" überschrieben sobald der "Step"-Eingang geändert wird.

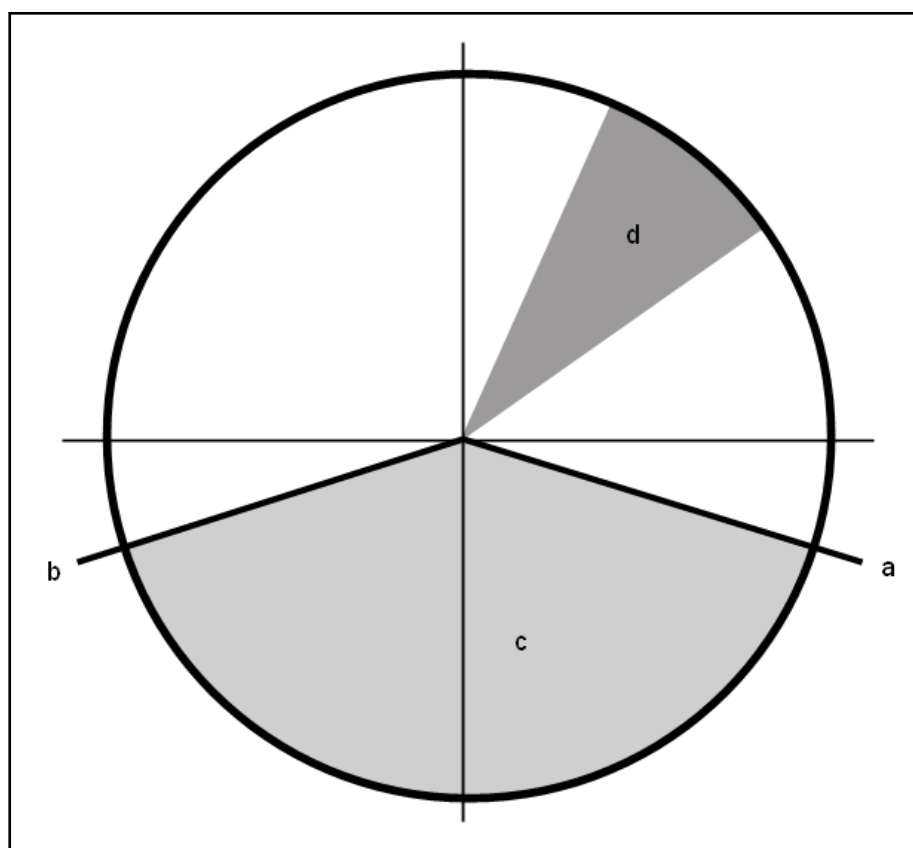
Korrekturfenster Das Korrekturfenster ist ein auf die Messachse bezogener Winkelbereich. Der Registerregler regelt die Regelungsabweichungen innerhalb dieses Fensters. Die Lage und Größe des Korrekturfensters werden mithilfe der folgenden Eingänge gesetzt:

Der Eingang "CorrectionMode" stellt das Korrekturfenster entweder auf absolut oder relativ ein. Wenn "CorrectionMode" auf CORR_WIN_DISABLED gesetzt wird, ist das Korrekturfenster inaktiv.

Der Eingang "CorrectionStart" legt den Anfang des Korrekturfensters fest.

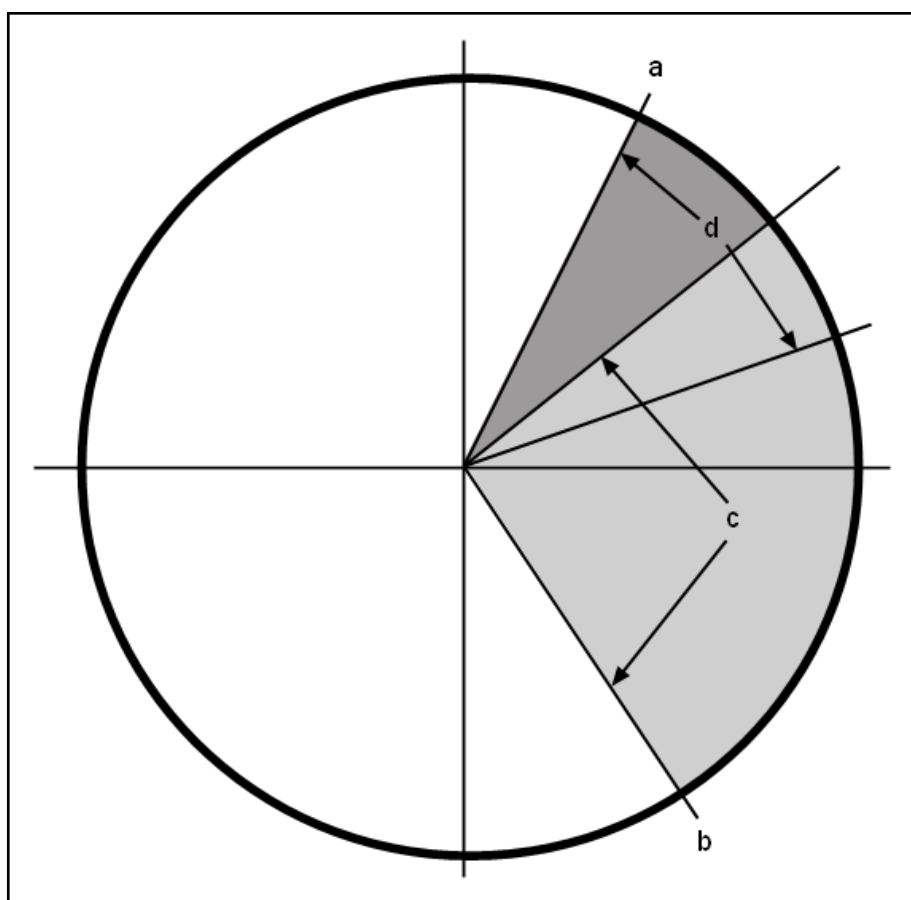
Der Eingang "CorrectionEnd" legt das Ende des Korrekturfensters fest.

Das Korrekturfenster unterstützt nur einen positiven Modulo-Bereich (z.B. 0...+360 Grad) der Messachse.



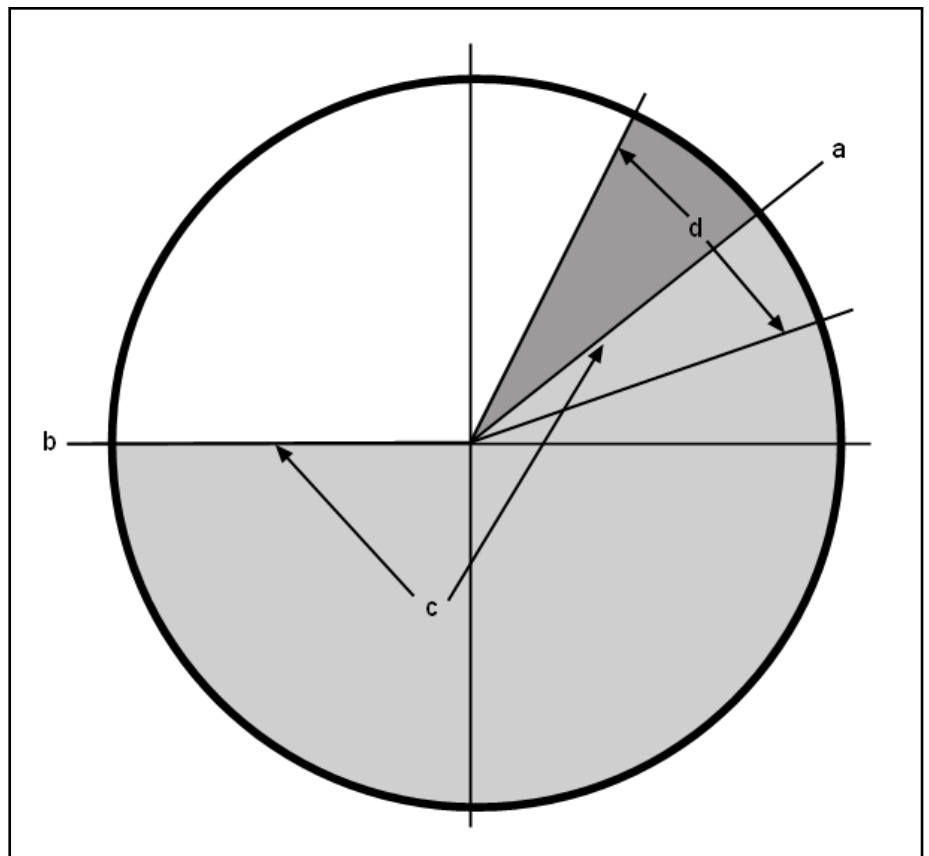
a: Beginn Korrekturfenster
b: Ende Korrekturfenster
c: Korrekturfenster
d: Erwartungsfenster
Abb. 7-9: Korrekturfenster absolut

ML_TechRegi.library/MX_TechRegi.lib



- a: Anfang Korrekturfenster = Messende
 b: Relatives Ende des Korrekturfensters bezogen auf das Messende
 c: Korrekturfenster
 d: Erwartungsfenster

Abb. 7-10: Korrekturfenster relativ



- a: Anfang Korrekturfenster = Messende
b: Ende des Korrekturfensters = perm. Winkel
c: Korrekturfenster
d: Erwartungsfenster

Abb. 7-11: Korrekturfenster, relativer Start und absolutes Ende

Normalbetrieb

Der RegisterController berechnet nur einen neuen Wert, wenn sich der Eingang "Step" verändert. Die Reglerparameter "PControl" und "IControl" können permanent geändert werden.

Wenn der Wert im "ControlValue"-Ausgang die Werte von "HighLimit" oder "LowLimit" über-, bzw. unterschreitet, dann wird "ControlValue" auf einen der folgenden Grenzwerte begrenzt.

- Ist der Wert in "ControlValue" positiv, wird der Wert in "HighLimit" verwendet.
- Ist der Wert in "ControlValue" negativ, wird der Wert in "LowLimit" verwendet.

Wenn der Registerregler eine Nachstellzeit hat, wird zusätzlich ein Anti-Reset-Windup durchgeführt und die Nachstellzeit wird nicht länger aufsummiert. In diesem Fall werden die Ausgänge "HighLimitActive" und "LowLimitActive" auf TRUE gesetzt.

Notwendige Parametrierung

IndraMotion MLC-Parametrierung

Wird eine reale Achse als geregelte Achse ("ControlledAxis"-Eingang) benutzt, muss der verwendete Parameter im "ControlValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695

ML_TechRegi.library/MX_TechRegi.lib



Zusätzlich muss dieser P-Parameter im MDT des zyklischen Datenkanals im Antrieb konfiguriert sein.

Wird eine virtuelle Achse als geregelte Achse benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten A-Parameter sein:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

IndraMotion MLD-Parametrierung

Der Parameter, der als geregelter Parameter für den "ControlledValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

Fehlerbehandlung

Der Funktionsbaustein gibt die Fehlermeldungen der folgenden unterlagerten Funktionsbausteine aus:

- MB_RegiMeasuringType01 ([Abb. 7-51 " Fehlercodes Funktionsbaustein MB_RegiMeasuringType01" auf Seite 344](#))
- IL_PIType02 (Bibliothek RIL_LoopControl, siehe Dokumentation "Rexroth IndraWorks 12VRS Basis-Bibliotheken")
- MB_RegiRegulateType02 ([Abb. 7-81 " Fehlercodes Funktionsbaustein MB_RegiRegulateType02 " auf Seite 364](#))
- MB_RegiSetpointLockType01 ([Abb. 7-104 " Fehlercodes Funktionsbaustein MB_RegiSetpointLockType01" auf Seite 382](#))

Er erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"Axis_Ref" liegt außerhalb des gültigen Wertebereichs (gemessen/geregelt)
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0001	"HighLimit" < "LowLimit"
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0002	"PControl" < 0

Abb. 7-12: Fehlercodes Funktionsbaustein MB_RegisterControllerType02

7.2.3 MB_RegisterControllerType04

Kurzbeschreibung

Der Registerregler-FB MB_RegisterControllerType04 unterstützt die folgende Funktionalität:

- Berechnung eines Korrekturwerts auf Grundlage eines gemessenen Werts und eines Sollwerts unter Verwendung eines P- oder PI-Regelkreises
- Preset-Funktion
- Pause-Funktion
- Min/Max-Begrenzung des berechneten Stellwerts
- Korrekturfenster-Funktion
- Schieberegister

ML_TechRegi.library/MX_TechRegi.lib

- Totzone-Kp-Kennlinie
- PT1-Element
- Sollwert übernehmen
- Markentakt-Funktion

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechRegi.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechRegi.lib

Abb.7-13: Referenztabelle MB_RegisterControllerType04

Schnittstellenbeschreibung

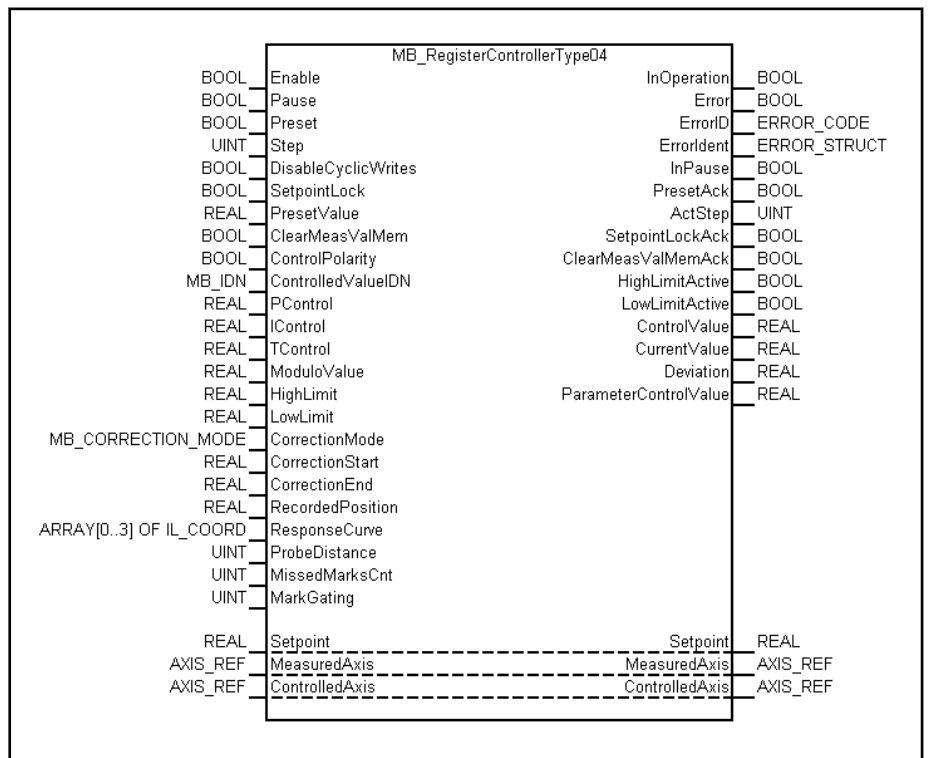


Abb.7-14: Funktionsbaustein MB_RegisterControllerType04

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Setpoint	REAL	Sollwert. Dieser Wert wird mit "RecordedPosition" verglichen
	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse
	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse
VAR_INPUT	Enable	BOOL	Aktiviert den Registerregler
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist. Die Ausgänge "ControlValue" und "ParameterControlValue" sind konstant, solange "Pause" aktiv ist

ML_TechRegi.library/MX_TechRegi.lib

E/A-Typ	Name	Datentyp	Beschreibung
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt den Regelkreis zwangsweise auf "Preset-Value". Zusätzlich setzt der "Preset"-Eingang "ParameterControlValue" auf den "PresetValue"-Wert
	Step	UINT	Verarbeitungsschritt
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlledValueIDN" wenn TRUE
	PresetValue	REAL	Wert an die Ausgänge "ControlValue" und "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	SetpointLock	BOOL	Aktiviert die Sollwertübernahme. Wenn TRUE, wird die nächste gemessene Marke gelatcht
	ClearMeasValMem	BOOL	Speicher für die Messwerte, die wird von der Funktion „Messtaster Abstand“ verwendet werden, löschen
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist
	ControlledValueIDN	MB_IDN	SERCOS-IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: IndraMotion MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 IndraMotion MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	PControl	REAL	Proportionalverstärkung des PI-Reglers
	IControl	REAL	Nachstellzeit T_n des PI-Reglers
	TControl	REAL	Filterkonstante für den Eingang "RecordedPosition"
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
	HighLimit	REAL	Maximalwert für "ControlValue"
	LowLimit	REAL	Minimalwert für "ControlValue"
	CorrectionMode	MB_CORRECTION_MODE	CORR_ABS = absolutes Korrekturfenster [CorrectionStart .. CorrectionEnd]. CORR_REL = relatives Korrekturfenster [Anfang = "RecordedPosition" + "CorrectionStart" .. Ende = "RecordedPosition" + "CorrectionEnd"] CORR_REL_ABS = relatives Anfangs... absolutes Endkorrekturfenster [Anfang = "RecordedPosition" + "CorrectionStart" .. Ende = "CorrectionEnd"] CORR_ABS_REL = absolutes Anfangskorrekturfenster... relatives Endkorrekturfenster [Anfang = "CorrectionStart" .. Ende = "RecordedPosition" + "CorrectionEnd"] CORR_WIN_DISABLED = Korrekturfenster ist deaktiviert
	CorrectionStart	REAL	Anfang des Korrekturfensters

E/A-Typ	Name	Datentyp	Beschreibung
	CorrectionEnd	REAL	Ende des Korrekturfensters
	RecordedPosition	REAL	Lage, an der die Flanke auftrat (in technischen Einheiten)
	ResponseCurve	ARRAY[0..3] OF IL_COORD	Kennlinie
	ProbeDistance	UINT	Der Abstand zwischen dem Markensensor und der geregelten Achse in Formatlängen (Leitachsenumdrehungen, Formate)
	MissedMarksCnt	UNIT	Zähler für fehlende Marken bei Verwendung eines Erwartungsfensters
	MarkGating	UINT	Jede n-te Marke wird für den Register verwendet
VAR_OUTPUT	InOperation	BOOL	Registerregler ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	InPause	BOOL	"Pause" ist aktiv. Die Ausgänge "ControlValue" und "ParameterControlValue" behalten ihre Werte
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in die Ausgänge "ControlValue" und "ParameterControlValue" kopiert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ControlValue" hoch
	SetpointLockAck	BOOL	"Setpoint" übernommen. Der aktuelle Wert in "RecordedPosition" wird nach "Setpoint" kopiert
	ClearMeasValMemAck	BOOL	Löschen des Messwertspeichers ist abgeschlossen
	HighLimitActive	BOOL	Obergrenze ist aktiv. "ControlValue" wäre größer, ist aber auf "HighLimit" beschränkt
	LowLimitActive	BOOL	Untergrenze ist aktiv. "ControlValue" wäre niedriger, ist aber auf "LowLimit" beschränkt
	ControlValue	REAL	Vom Registerregler errechneter Stellwert
	CurrentValue	REAL	Im Registerregler momentan verwendeter Wert. Er ist gleich dem "RecordedPosition"-Eingang wenn "ProbeDistance" und "MarkGating" deaktiviert sind

ML_TechRegi.library/MX_TechRegi.lib

E/A-Typ	Name	Datentyp	Beschreibung
	Deviation	REAL	"Deviation" = ("Setpoint" - "CurrentValue") des Registerreglers
	ParameterControlValue	REAL	Der Registerregler berechnet einen Wert basierend auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parametereingang "ControlledValueIDN" geschrieben wenn der Eingang "DisableCyclicWrites" = FALSE ist

Abb. 7-15: Schnittstellenvariablen Funktionsbaustein MB_RegisterController-Type04

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	"Enable" TRUE (kontinuierlich)
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
DisableCyclicWrites	BOOL			FALSE	Kontinuierlich
SetpointLock	BOOL			FALSE	Kontinuierlich
ClearMeasValMem	BOOL			FALSE	Kontinuierlich
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
Setpoint	REAL	n.def	n.def	0.0	Änderung von "Step"
PControl	REAL	0.0	n.def	1.0	Kontinuierlich
IControl	REAL	0.0	n.def	0.0	Kontinuierlich
TControl	REAL	0.0	n.def	0.0	Kontinuierlich
ModuloValue	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"
HighLimit	REAL	LowLimit	n.def	0.0	Kontinuierlich
LowLimit	REAL	n.def	HighLimit	0.0	Kontinuierlich
CorrectionMode	MB_CORRECTION_MODE			CORR_REL	Kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
CorrectionStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich
RecordedPosition	REAL	n.def	n.def	0.0	Änderung von "Step"
ResponseCurve	ARRAY[0..3] OF IL_COORD	n.def	n.def	0.0	Kontinuierlich
ProbeDistance	UINT	0	30	0	Kontinuierlich
MissedMarksCnt	UINT	0	65.535	0	Kontinuierlich. Initialisierung bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
MarkGating	UINT	0	65.535	1	Steigende Flanke an "Enable". Ein Wert von "0" deaktiviert den Markentakt (identisch mit dem Wert "1")

Abb. 7-16: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Das folgende Diagramm zeigt den Signal-Zeit-Verlauf von FB MB_RegisterControllerType04 einschließlich der Funktionen "Pause" und "Preset".

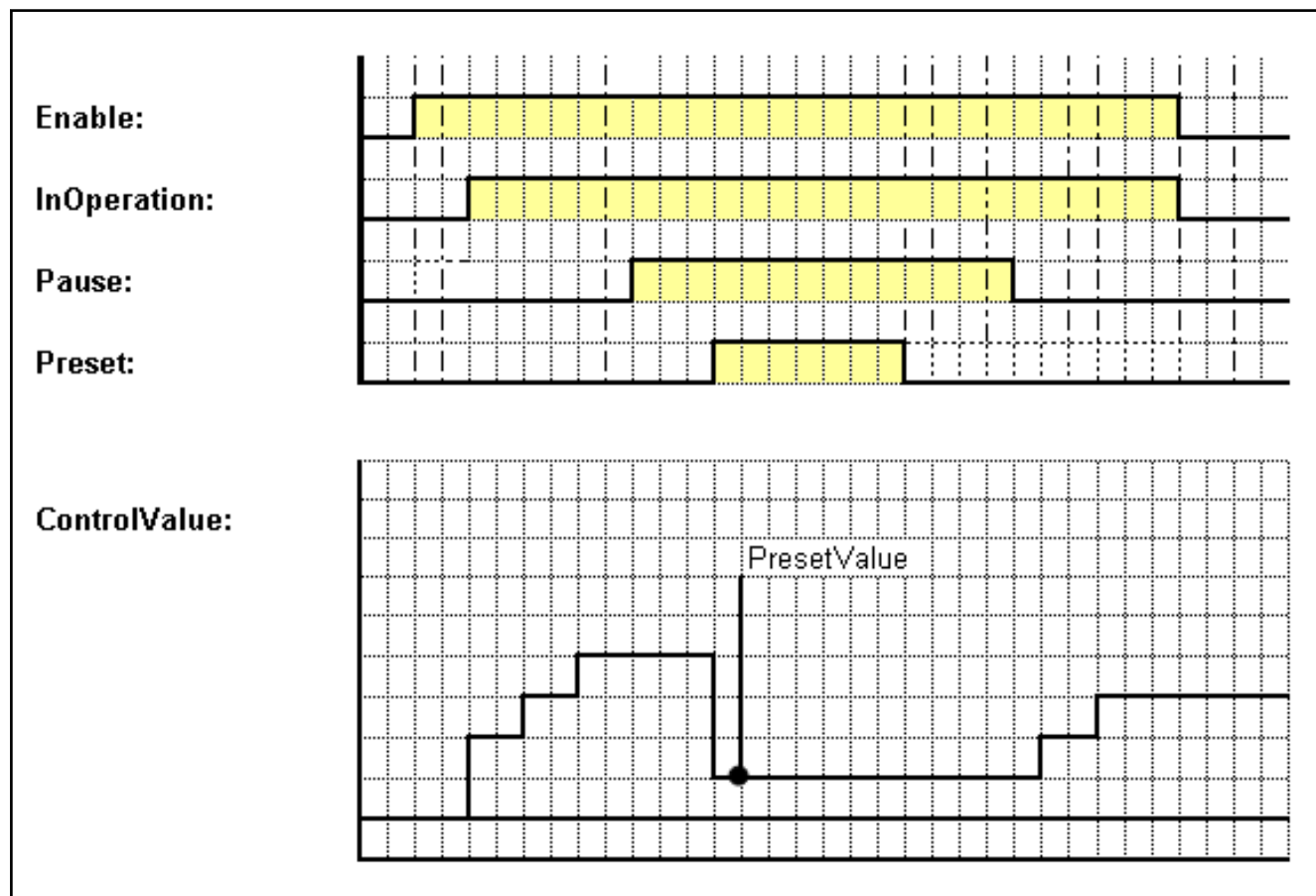


Abb. 7-17: Zeitdiagramm des FB MB_RegisterControllerType04 mit den Funktionen "Pause" und "Preset"

ML_TechRegi.library/MX_TechRegi.lib

Funktionsbeschreibung Ein Sensor erkennt Marken, Perforationen, Schnitte oder einen geklebten Stoß auf dem Material und meldet dies dem Antrieb als (binäres) Signal.

Die Messtasterfunktion des Antriebs bestimmt die Flanke des Sensorsignals und verriegelt, abhängig von der Signalauswahl des Messtasters, die folgenden Stellungen:

Antriebsistposition Parameter	S-0-0051, S-0-0053, P-0-0753
Leitachspanposition Parameter	P-0-0052, P-0-0227, P-0-0775, P-0-0776, P-0-0778

Die Messtasterfunktion zeichnet Positionsdaten mit einer Auflösung von 0,5µs auf. Der Sensor muss ein 24V-Signal mit einer Anstiegszeit im µs-Bereich bereitstellen. Die sensorspezifische Verzögerungszeit kann mithilfe der Eingänge "DeadTimePosEdge" oder "DeadTimeNegEdge" des Funktionsbausteins MB_InitTouchProbe kompensiert werden.

Der Registerregler berechnet die Regelungsabweichung zwischen der gemessenen und der Sollwertposition und ermittelt den erforderlichen Korrekturwert, sobald eine neue Flanke des Sensorsignals ermittelt wird.

Änderungen des Korrekturwertes werden über das trapezförmige Profil, die Geschwindigkeitsrampe, den PT1-Filter oder ohne Verzögerung aktiv, abhängig vom ausgewählten Regelungsparametereingang ("ControlledValueIDN") des Funktionsbausteins. Die nachstehende Tabelle führt die verfügbaren Regelungsparameter ("ControlledValueIDN") sowie das daraus resultierende MotionProfile aller verfügbaren Antriebsbetriebsarten mit Synchronisation auf.

Parametername	Controlled-ValueIDN	Bewegungsverhalten, wenn sich ein Parameterwert ändert	Verfügbar in den folgenden Betriebsmodus			
			Velocity-Synchronization	Phase-Synchronization	Cam Shaft	Electronic Motion Profile
Winkelverschiebung Anfang des Profils	P-0-0061 A-0-2730	Änderungen von P-0-0061 werden über Rampe mithilfe P-0-0158 (Geschwindigkeit) aktiv. Wenn P-0-0158=0, werden Änderungen an P-0-0061 sofort aktiv.	Nein	Nein	Ja	Ja
Geschwindigkeits-Sollwert additiv	P-0-0690 A-0-2615	Änderungen von P-0-0694 werden sofort aktiv	Ja	Nein	Nein	Nein
Lagesollwert additiv, Prozessregler	P-0-0691 A-0-2610	Wenn P-0-0155 Bit0 = 0: Änderungen von P-0-0691 werden über PT1-Filter mittels P-0-0060 (Filterzeitkonstante) aktiv. Wenn P-0-0060=0, werden Änderungen an P-0-0691 sofort aktiv. Wenn P-0-0155 Bit0 = 1: Änderungen von P-0-0691 werden sofort aktiv	Ja	Ja	Ja	Ja
Leitachspannung additiv, Prozessregler	P-0-0692 A-0-2600	Änderungen von P-0-0692 werden mit einem Filter 1. Ordnung geglättet bei Verwendung von P-0-0693 (Filterzeitkonstante) und dann zur Summe aus P-0-0054, Leitachspannung additiv, und P-0-0053, Leitachspannung, oder P-0-0052, Lageistwert Messgeber, addiert.	Ja	Ja	Ja	Ja
Feinabgleich Getriebeübersetzung, Prozessregler	P-0-0694 A-0-2605	Änderungen von P-0-0694 werden sofort aktiv	Ja	Ja	Ja	Ja
Winkelverschiebung Tabellenanfang, Prozessregler	P-0-0695 A-0-2620	Änderungen von P-0-0695 werden über PT1-Filter mittels P-0-0696 (Filterzeitkonstante) aktiv. Wenn P-0-0696=0, werden Änderungen an P-0-0695 sofort aktiv.	Nein	Nein	Ja	Ja

Abb. 7-18: Unterstützte Regelparameter und unterstütztes Verhalten

Die folgenden "ControlledValueIDN"-Werte werden integriert:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Initialisierung Während der Initialisierung müssen die Eingänge „Setpoint“ und „RecordedPosition“ gültige Daten enthalten.

PresetValue Für einen fehlerfreien Start der Reglerausgänge "ControlValue" und "ParameterControlValue", muss der "Preset"-Eingang gesetzt sein, bevor der "Enable"-Eingang aktiviert wird. Während dieser Zeit müssen die Eingänge

ML_TechRegi.library/MX_TechRegi.lib

"Setpoint" und "RecordedPosition" über gültige Daten verfügen. Zudem werden die oberen und unteren Grenzwerte überwacht, um eine Begrenzung des Ausgangs "ControlValue" sicherzustellen, wenn der aktuelle Wert für "PresetValue" außerhalb des Gültigkeitsbereiches von "HighLimit" bzw. "LowLimit" liegt.

Der "PresetAck"-Ausgang ist aktiv wenn der Wert am "PresetValue"-Eingang in die Ausgänge "ControlValue" und "ParameterControlValue" kopiert wird. Der "Preset"-Eingang hat eine höher Priorität als der "Pause"-Eingang. Der Wert in "PresetValue" wird in die entsprechenden Ausgänge bei steigender Flanke am "Preset"-Eingang kopiert wenn "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt.

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. In diesem Fall wird die Abweichung ("Setpoint" – "RecordedPosition") auf 0 gesetzt. Dadurch ändern sich die Ausgänge "ControlValue" und "ParameterControlValue" nicht mehr. Diese kann verwendet werden, um die Werte der Ausgänge "ControlValue" und "ParameterControlValue" beizubehalten. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Solange der Eingang "Pause" aktiv ist, ist der Eingang "InPause" TRUE.

Sollwertübernahme Einen Moduswechsel von Manuell zu Automatik wird durch das Einrasten beim nächsten erkannten Registermarker erreicht. Wenn zum Beispiel eine neue Papierbahn in Position gebracht und die Registermarke erkannt wird, wird der Eingang "SetpointLock" auf TRUE gesetzt. Der Wert im "Setpoint" wird mit dem Wert im "RecordedPosition" überschrieben sobald der "Step"-Eingang geändert wird.

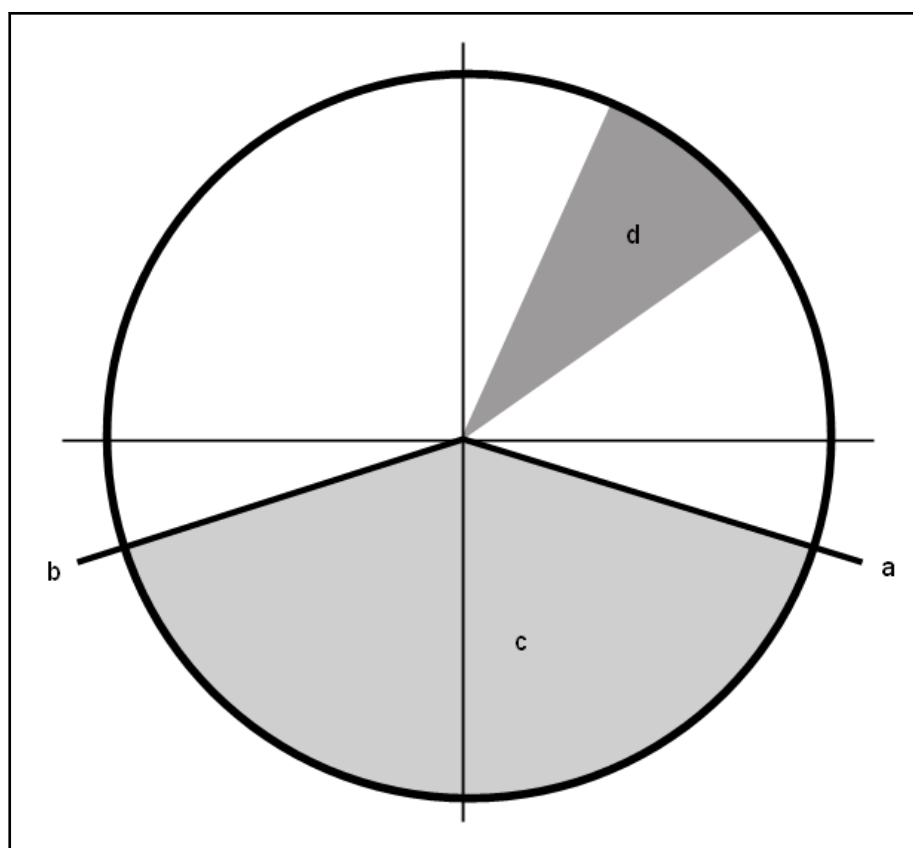
Korrekturfenster Das Korrekturfenster ist ein auf die Messachse bezogener Winkelbereich. Der Registerregler regelt die Regelungsabweichungen innerhalb dieses Fensters. Die Lage und Größe des Korrekturfensters werden mithilfe der folgenden Eingänge gesetzt:

Der Eingang "CorrectionMode" stellt das Korrekturfenster entweder auf absolut oder relativ ein. Wenn "CorrectionMode" auf CORR_WIN_DISABLED gesetzt wird, ist das Korrekturfenster inaktiv.

Der Eingang "CorrectionStart" legt den Anfang des Korrekturfensters fest.

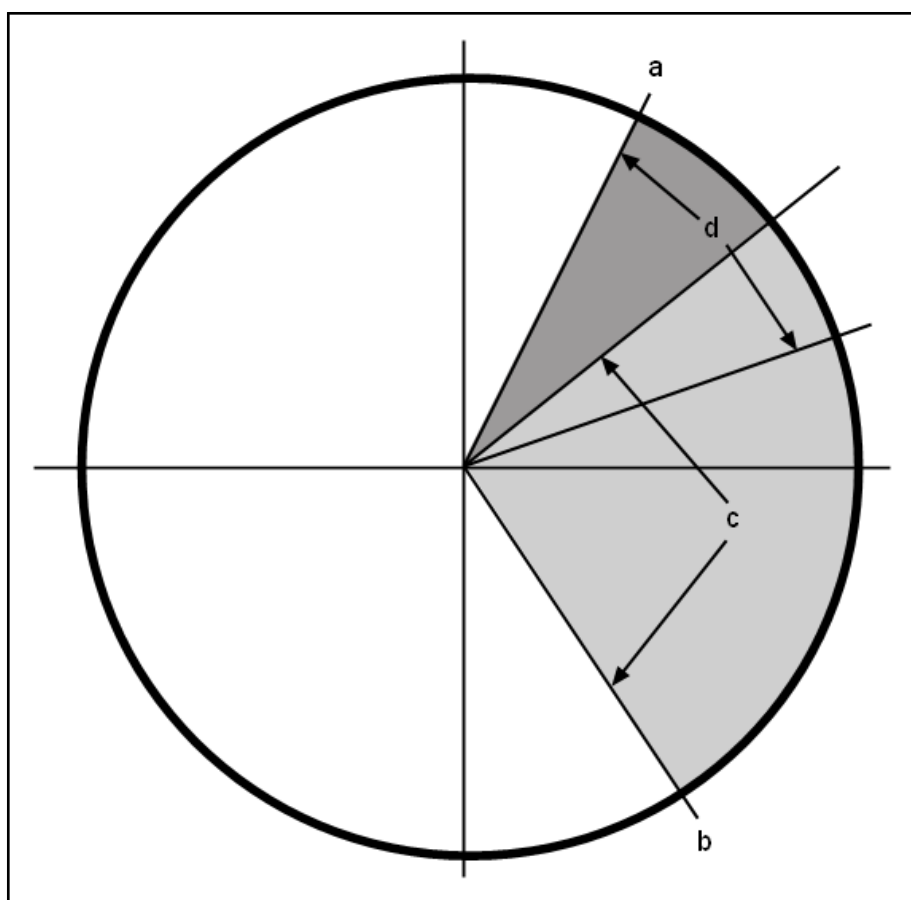
Der Eingang "CorrectionEnd" legt das Ende des Korrekturfensters fest.

Das Korrekturfenster unterstützt nur einen positiven Modulo-Bereich (z.B. 0...+360 Grad) der Messachse.



a: Beginn Korrekturfenster
b: Ende Korrekturfenster
c: Korrekturfenster
d: Erwartungsfenster
Abb. 7-19: *Korrekturfenster absolut*

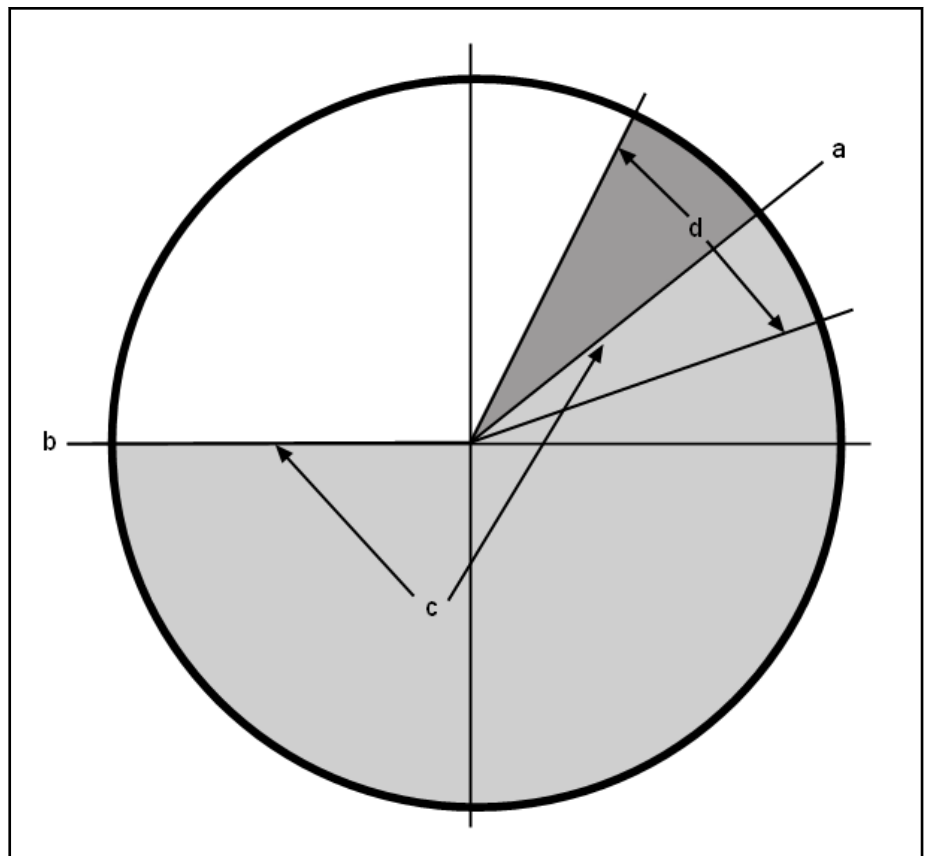
ML_TechRegi.library/MX_TechRegi.lib



- a: Anfang Korrekturfenster = Messende
 b: Relatives Ende des Korrekturfensters bezogen auf das Messende
 c: Korrekturfenster
 d: Erwartungsfenster

Abb. 7-20:

Korrekturfenster relativ



- a: Anfang Korrekturfenster = Messende
- b: Ende des Korrekturfensters = perm. Winkel
- c: Korrekturfenster
- d: Erwartungsfenster

Abb. 7-21: Korrekturfenster, relativer Start und absolutes Ende

Glättung Der Eingang "TControl" ist die Dämpfungskonstante für den Eingang "RecordedPosition". Wenn er auf 0 gesetzt wird, ist die Glättung inaktiv.

ProbeDistance Im einfachsten Fall ist die Messachse ein Formatzylinder (= geregelte Achse) und der Abstand des Sensors zur Messachse (= geregelten Achse) ist kleiner als die Formatlänge. Messung und Korrektur beziehen sich auf dasselbe Produkt. Oft muss der Markenleser (Sensor) weiter weg von der Messachse angebracht werden, z.B. bei Gefahrbereichen in Druckanlagen. Wenn der Abstand größer als die Formatlänge ist, kann der gemessene Registerfehler erst korrigiert werden wenn das Produkt den Formatzylinder erreicht hat. Die Korrektur muss um so viele Formate verzögert werden, wie es Formate zwischen der geregelten Achse und dem Sensor gibt. Der Eingang "ProbeDistance" ist der Abstand zwischen dem Markenleser (Sensor) und der geregelten Achse, in Formatlängen (Leitachsumdrehungen, Formate). Wenn er auf 0 gesetzt ist, ist der Abstand des Sensors zur Messachse kleiner als die Formatlänge.

ML_TechRegi.library/MX_TechRegi.lib

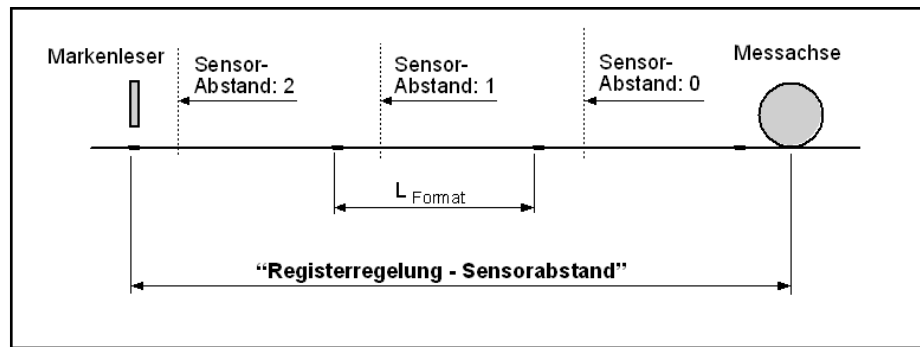


Abb. 7-22: Messtasterabstand

Löschen des internen Messwertspeichers

Die Funktion „Messtasterabstand“ enthält einen internen Speicher, der Messwerte (wie ein Schieberegister) speichert. Dieser interne Speicher kann mit dem Eingang "ClearMeasValMem" gelöscht (auf "0" gesetzt) werden.

DeadBand Das Differenzsignal des gefilterten Eingangs "RecordedPosition" und des Eingangs "Setpoint" wird durch den Eingang "ResponseCurve" gewichtet. Eine Kennlinie wird verwendet, um den Registerregler innerhalb kleiner Registerfehler anzuhalten.

MarkGating Mit diesem Feature kann der Anwender die Registrierung nur bei jeder n-ten Markierung regeln, indem er ungültige Markierungen ausfiltert. Man kann das z.B. verwenden, wenn die Markierungen auf dem Gewebe mit einem Klischee gedruckt wurden, auf dem das gedruckte Bild mehrfach erscheint. In dieser Situation sind die Markierungsabstände zwischen aufeinander folgenden Markierungen nicht konstant. Die Markierungsabstände beispielsweise zwischen jeder n-ten aufeinanderfolgenden Markierung sind jedoch konstant. Der Registerregler verwendet dann nur jede n-te Markierung für die Registerregelung.

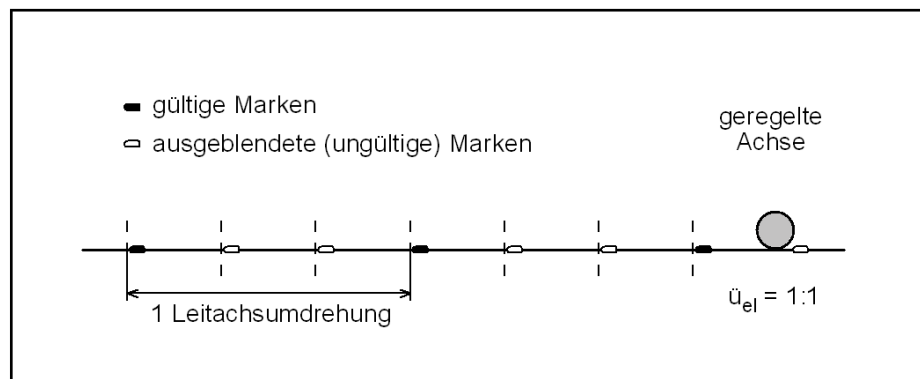


Abb. 7-23: Markierungsausblenden

Normalbetrieb Der RegisterController berechnet nur einen neuen Wert, wenn sich der Eingang "Step" verändert. Die Reglerparameter "PControl" und "IControl" können permanent geändert werden.

Wenn der Wert im "ControlValue"-Ausgang die Werte von "HighLimit" oder "LowLimit" über-, bzw. unterschreitet, dann wird "ControlValue" auf einen der folgenden Grenzwerte begrenzt.

- Ist der Wert in "ControlValue" positiv, wird der Wert in "HighLimit" verwendet.
- Ist der Wert in "ControlValue" negativ, wird der Wert in "LowLimit" verwendet.

Wenn der Registerregler eine Nachstellzeit hat, wird zusätzlich ein Anti-Reset-Windup durchgeführt und die Nachstellzeit wird nicht länger aufsummiert. In diesem Fall werden die Ausgänge "HighLimitActive" und "LowLimitActive" auf TRUE gesetzt.

Notwendige Parametrierung

IndraMotion MLC-Parametrierung

Wird eine reale Achse als geregelte Achse ("ControlledAxis"-Eingang) benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695



Zusätzlich muss dieser P-Parameter im MDT des zyklischen Datenkanals im Antrieb konfiguriert sein.

Wird eine virtuelle Achse als geregelte Achse benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten A-Parameter sein:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

IndraMotion MLD-Parametrierung

Der Parameter, der als geregelter Parameter für den "ControlledValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

Fehlerbehandlung

Der Funktionsbaustein gibt die Fehlermeldungen der folgenden unterlagerten Funktionsbausteine aus:

- MB_RegiMeasuringType03 ([Abb. 7-66 " Fehlercodes Funktionsbaustein MB_RegiMeasuringType03" auf Seite 351](#))
- IL_PIType02 (Bibliothek RIL_LoopControl, siehe Dokumentation "Rexroth IndraWorks 12VRS Basis-Bibliotheken")
- MB_RegiRegulateType02 ([Abb. 7-81 " Fehlercodes Funktionsbaustein MB_RegiRegulateType02 " auf Seite 364](#))
- MB_RegiSetpointLockType01 ([Abb. 7-104 " Fehlercodes Funktionsbaustein MB_RegiSetpointLockType01" auf Seite 382](#))

Er erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"Axis_Ref" liegt außerhalb des gültigen Wertebereichs (gemessen)
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0001	"HighLimit" < "LowLimit"
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0002	"PControl" < 0

Abb.7-24: Fehlercodes Funktionsbaustein MB_RegisterControllerType04

7.2.4 Konfiguration des Registerreglers

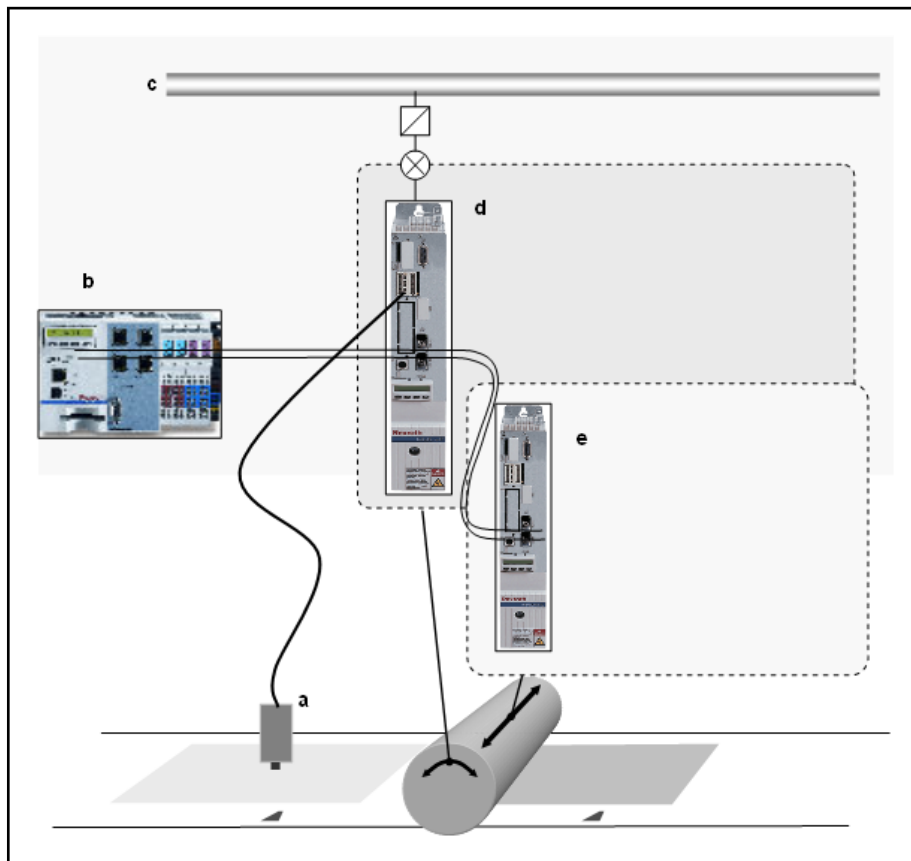
Überblick

Der Registerregler-FB MB_RegisterControllerType04 wird in Papier-, Druck-, Verpackungs- und Folienapplikationen eingesetzt, bei denen synchronisierte

ML_TechRegi.library/MX_TechRegi.lib

Antriebe (z.B. Winkel- oder Kurvenscheibensynchronisation), Material durch eine Maschine transportieren.

In diesen Applikationen beeinträchtigen Schwankungen in der Materialbeschaffenheit, rutschendes Material und der Produktionsprozess die Genauigkeit der Materialposition. Ein Sensor bestimmt die aktuelle Lage der Registermarken, die auf dem Material angebracht oder gedruckt sind und der Registerregler misst die Abweichung zum Sollwert. Er berechnet die notwendige Korrektur mithilfe eines Regelkreises. Durch diese Methode bleiben die Registermarken genau bei ihrem Sollwert und verschieben sich nicht. Die folgende Abbildung zeigt die Implementierung eines Registerreglers anhand eines Beispiels:



a Sensor
b: IndraMotion MLC-Regelung
c Leitachse (virtuelle Achse)
d Registerregelung, Messtasterfunktion, Winkelsynchronisation
Abb. 7-25: Beispiel eine Registerreglers bei einem IndraMotion MLC-basierten System

Einsatz der Registerregelung

Überblick

Die schnellste Möglichkeit, die Registerregelungsfunktionalität in einer Anwendung einzusetzen, ist die Kombination des Registerregler-FB mit einem Messtaster-FB. Der Messtaster-FB liefert den Istwert an den Registerregler. Der Registerregler berechnet den Korrekturwert auf Basis der aktuellen Abweichung zwischen dem Sollwert und dem Istwert.

Das Beispiel in diesem Abschnitt zeigt den Registerregler-FB MB_Register-ControllerType04 mit dem Funktionsbaustein MB_TouchProbeContinuous.

Konfiguration des Messtaster-FB

Die folgenden Messtastersignale müssen mithilfe des Funktionsbausteins MB_InitTouchProbe konfiguriert werden:

1. Antriebsmesstaster 1
2. Positiver Messtaster 1
3. Markenausfall aktiviert
4. Erwartungsfenster aktiviert



Die Beschreibung für den Messtaster-FB finden Sie in [Kap. 5.7.3 "MB_InitTouchProbe"](#) auf Seite 102.

Konfiguration des Messtaster-FB und des RegisterControllerType04-FB

Die folgende Abbildung zeigt die Verbindung der Funktionsbausteine MB_TouchProbeContinuous und MB_RegisterControllerType04, wenn die Messtasterfunktion durch eine positive Flanke ausgelöst wird. Teil der Konfiguration des Funktionsbausteins MB_InitTouchProbe sind der aktivierte Markenausfall und das aktivierte Erwartungsfenster. Daher muss der Zähler für Markenausfälle auch verbunden werden.

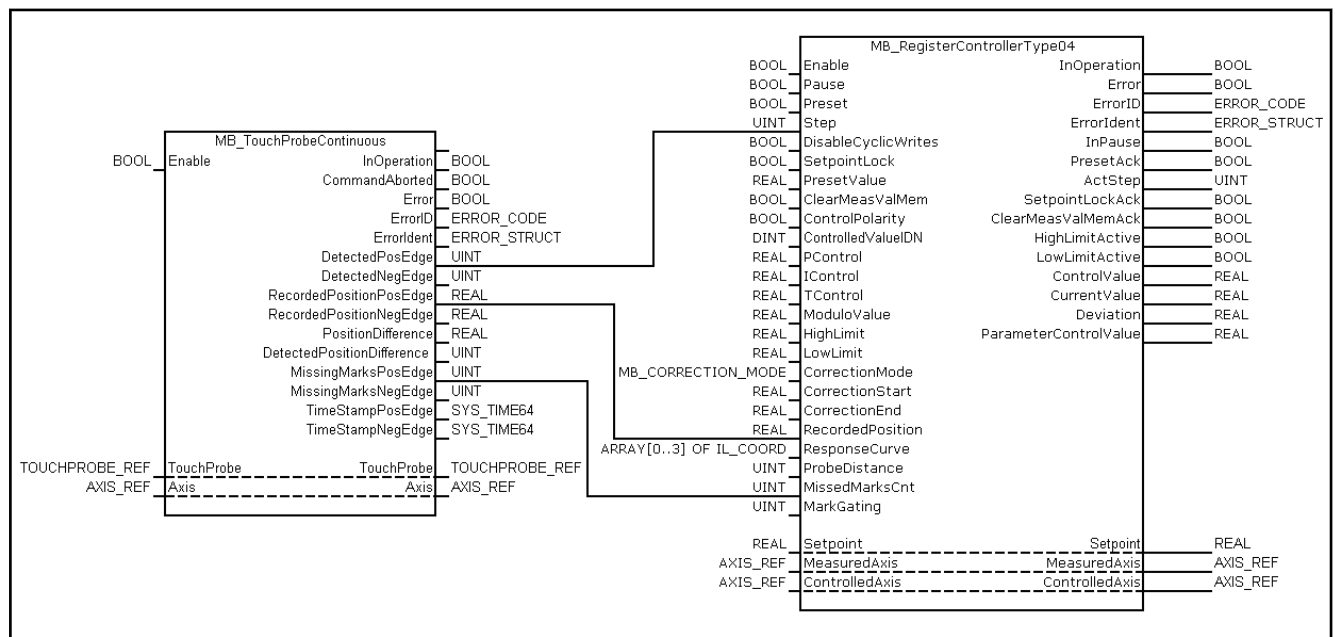


Abb. 7-26: Die Schnittstelle vom MB_TouchProbeContinuous und MB_RegisterControllerType04

Eine vollständige Beschreibung der Eingänge und Ausgänge finden Sie bei den einzelnen Beschreibungen der Funktionsbausteine.

In der obenstehenden Abbildung sind die folgenden Signalverbindungen beschrieben:

1. "DetectedPosEdge" zu "Step"
2. "RecordedPositionPosEdge" zu "RecordedPosition"
3. "MissingMarksPosEdge" zu "MissedMarksCnt"

Diese Verbindung ist nur notwendig wenn das Erwartungsfenster und der Zähler für Markenausfälle verwendet wird.

ML_TechRegi.library/MX_TechRegi.lib

Parametrierung des Registerreglers

Achswichtung

Um die Registerregelungsfunktionalität zu verwenden, müssen beide Achsen (gemessene Achse, geregelte Achse) ins rotatorische Modulo-Format gesetzt werden. Die Menüauswahl bei IndraWorks wie folgt:

- Für die IndraMotion MLD, klicken Sie auf **IndraWorks ▶ Achse ▶ Wichtung / Mechanisches System ▶ Wichtung / Maßeinheiten**.
- Für die IndraMotion MLC, klicken Sie auf **IndraWorks ▶ Motion ▶ Virtuelle Achsen / Reale Achsen ▶ Achse ▶ Wichtung / Mechanisches System ▶ Wichtung / Maßeinheiten**.

Siehe als Beispiel folgende Abbildung:

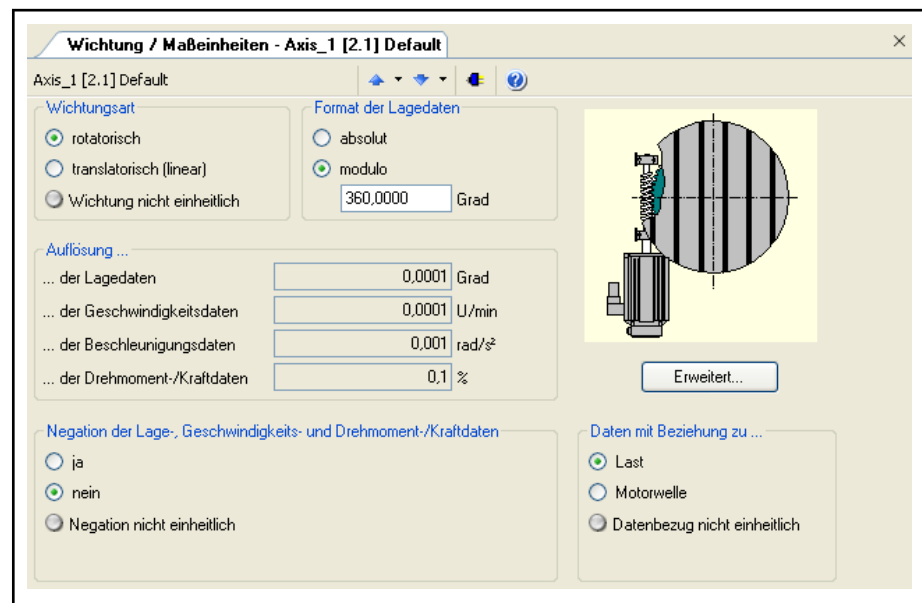


Abb.7-27: Einstellen der Achswichtung

Konfiguration des IndraMotion MLC Master-Datentelegramms (MDT)

Der verwendete Parameter für den "ControlledValueIDN"-Eingang muss im MDT der geregelten Achse konfiguriert werden. Wählen Sie aus dem entsprechenden IndraWorks-Projekt **IndraWorks ▶ Motion**, dann Rechtsklick auf der geregelten Achse und wählen Sie **Kommunikation ▶ Zyklischer SERCOS-Kanal**.

Siehe als Beispiel folgende Abbildung:

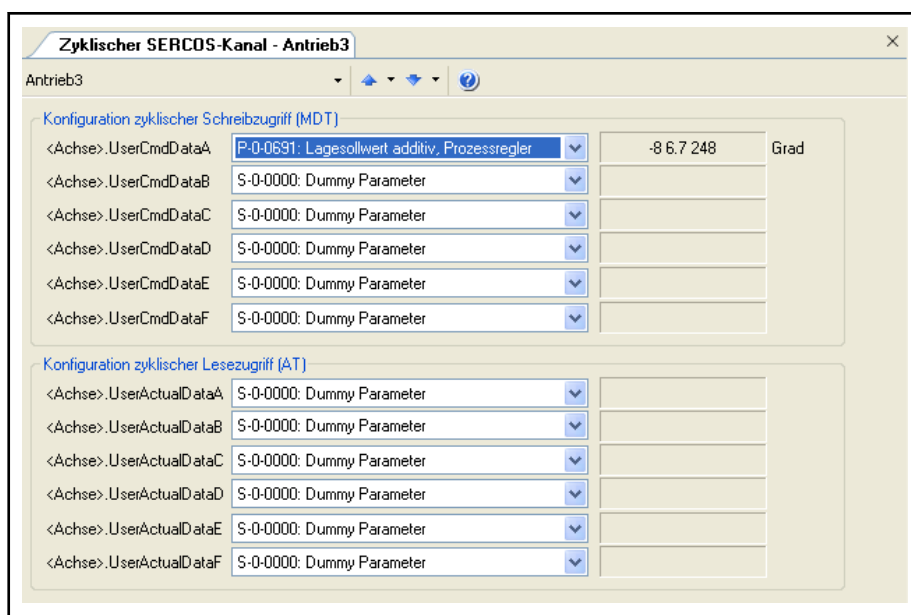


Abb.7-28: MDT-Konfiguration der geregelten Achse (IndraMotion MLC)

IndraMotion MLD-Achsdatenkonfiguration

Der verwendete Parameter für den "ControlledValueIDN"-Eingang muss in der AxisData-Struktur der geregelten Achse konfiguriert werden. Wählen Sie aus dem entsprechenden IndraWorks-Projekt **IndraWorks** ► **MLD** ► **Konfiguration** und klicken Sie auf **Unterstützung der AxisData-Struktur**.

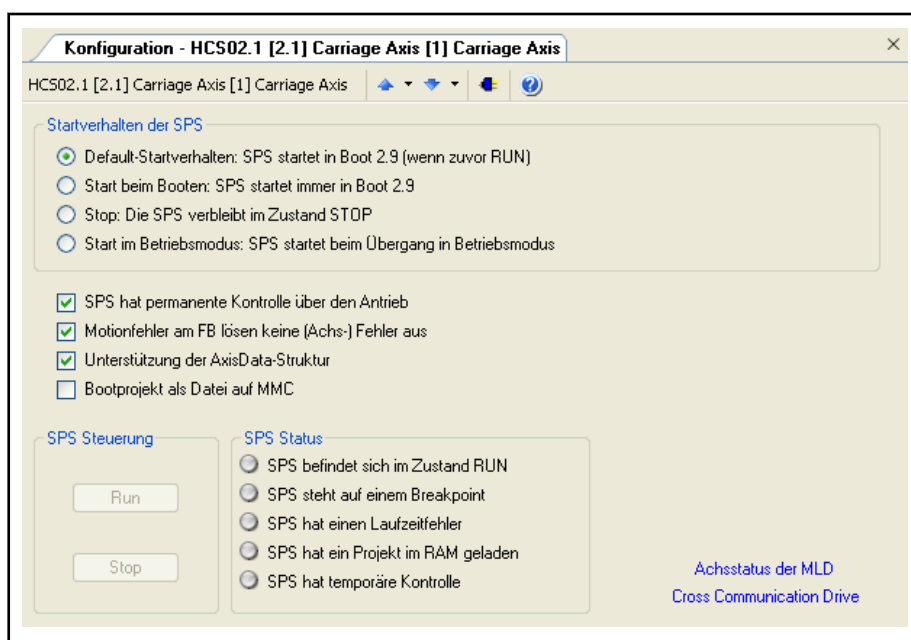


Abb.7-29: IndraMotion MLD-Konfiguration

Anschließend klicken Sie auf **IndraWorks** ► **MLD** ► **Konfiguration** ► **AxisData** und wählen Sie den verwendeten Parameter für den "ControlledValueIDN"-Eingang aus.

ML_TechRegi.library/MX_TechRegi.lib

dwUserCmdDataA_q	P-0-0691 : Lagesollwert additiv, Prozessregler	▼
dwUserCmdDataB_q	S-0-0000 : Frei	▼
dwUserCmdDataC_q	S-0-0000 : Frei	▼
dwUserCmdDataD_q	S-0-0000 : Frei	▼

Abb. 7-30: AxisData-Konfiguration

7.2.5 Registerregler – Anwendungsbeispiele

Stanzen bei Etikettendruck

Das folgende Beispiel stellt einen Stanzvorgang an einer Etikettendruckmaschine dar:

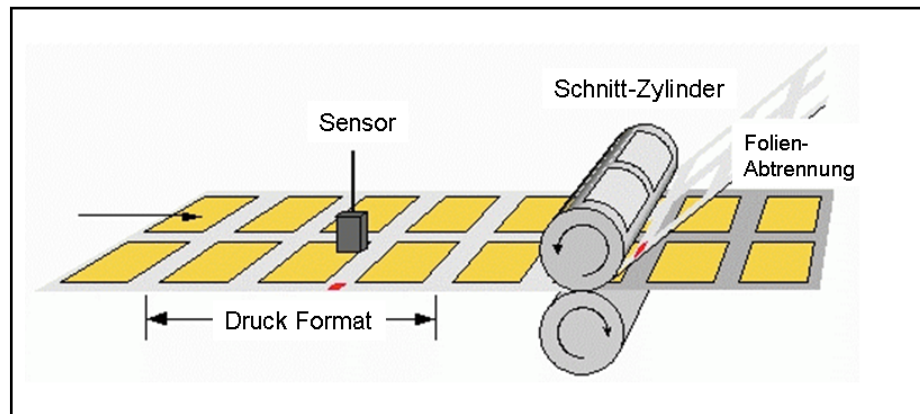


Abb. 7-31: Registerregelung einer Stanze

Etiketten werden auf einen Materialverbund einschließlich einer Verstärkungsfolie und einer selbstklebenden Decklage gedruckt. Das unbedruckte Raster wird eingeschlagen und umgewickelt.

Die Aufgabe des Registerreglers ist es, den Stanzzylinder auf die bedruckte Papierbahn einzustellen.

Der Stanzzylinder läuft im winkelsynchronen Betrieb. Die Ausrichtung auf die bedruckte Papierbahn erfolgt durch Anpassen des Winkelversatzes. Der Sensor ist am Antrieb des Stanzzylinders angeschlossen. Die Winkelposition (der Lageistwert) des Stanzzylinders wird über das Sensorsignal erfasst.

Der Positionierbefehl für den Registerregler entspricht der Zylinderposition, die exakt auf das Produkt ausgerichtet ist. Der Korrekturwert wird anhand der Differenz zwischen Referenz- und Istwert mit anschließender Addition zum aktuellen Winkelversatz berechnet.

Die wichtigsten Einstellungen für den Messtaster, den Registerregler und die Antriebsbetriebsart dieser Applikation sind:

- Gemessener Wert (Messtasterfunktion) = Lageistwert Geber 1 (S-0-0051)
- Geregelter Wert (ControlledValueIDN) = Lagesollwert additiv (P-0-0691)
- Antriebsbetriebsart = Winkelsynchronisation

Insetter-Regelung

Die folgende Abbildung stellt eine Insetter-Regelung in einer Materialzuführungsapplikation dar:

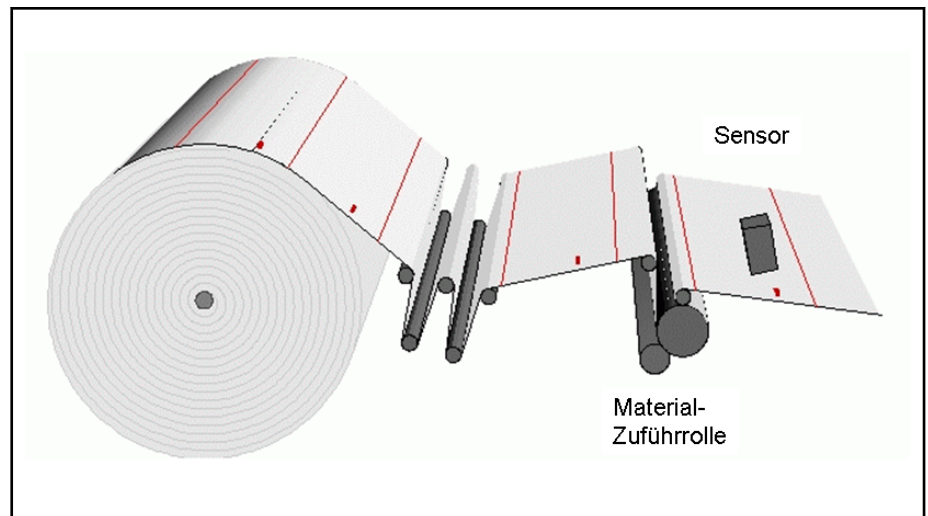


Abb.7-32: Insetter-Regelung der Materialzuführung

Die Materialzuführungseinheit transportiert vorgedrucktes Material zu einer Druckmaschine. Nach dem Vordrucken und Trocknen ist die Papierbahn geschrumpft, d. h. die Formatlänge stimmt nicht exakt mit dem Nennformat überein. Die Bahn muss zur weiteren Bearbeitung wieder ausgedehnt werden.

Die Aufgabe des Registerreglers besteht darin, die Geschwindigkeit der Materialzuführungsrolle so zu regeln, dass die vorgedruckte Papierbahn auf das nominale Papierformat gedehnt wird. Dies setzt voraus, dass bei jeder Umdrehung der Leitachse jeweils ein Papierformat weitertransportiert wird.

Der Positionierbefehl des Registerregler bezieht sich auf die Leitachse.

Die Materialzuführungsrolle läuft im Geschwindigkeitssynchronisationsmodus. Die Geschwindigkeit der Zuführungsrolle wird über den Getriebeübersetzungs-Feinabgleich geregelt. Die Papierbahnausdehnung ist dann richtig, wenn die Druckmarke immer in der gleichen Leitachsisposition unter dem Markenleser erkannt wird. Die Leitachsisposition wird über das Sensorsignal erfasst. Der Markenleser ist am Messtastereingang des Materialzuführungsrollen-Antriebs angeschlossen.

Die wichtigsten Einstellungen für den Messtaster, den Registerregler und die Antriebsbetriebsart dieser Applikation sind:

- Gemessener Wert (Messtasterfunktion) = resultierende Leitachsisposition (P-0-0775)
- Geregelter Wert (ControlledValueIDN) = Getriebeübersetzungs-Feinabgleich (P-0-0694)
- Antriebsbetriebsart = Drehzahlsynchronisation

Schlauchbeutelmaschine

Die folgende Abbildung stellt eine Sideseal-Achse (Folieneinzugsachse) einer Schlauchbeutel-Applikation dar:

ML_TechRegi.library/MX_TechRegi.lib

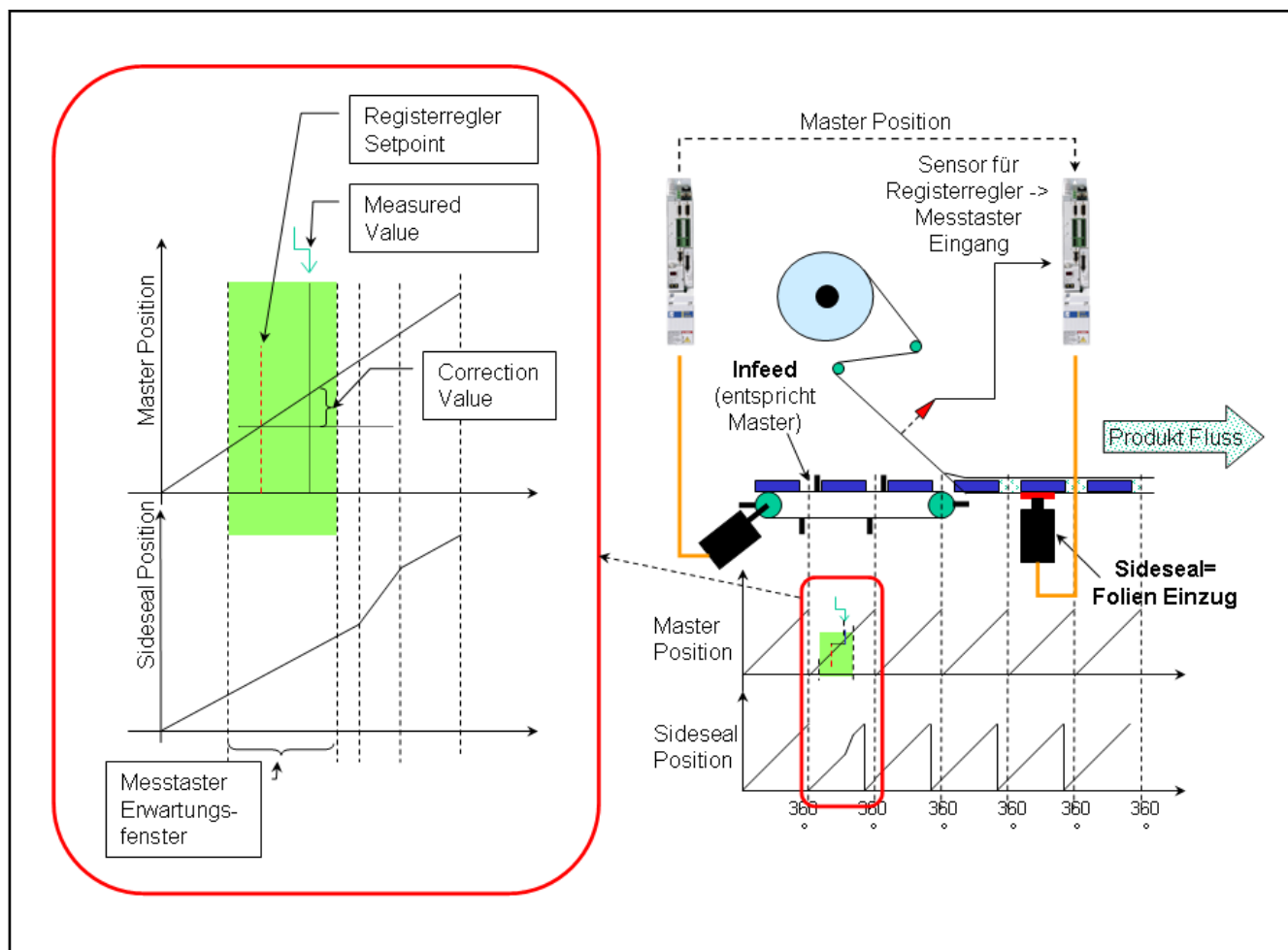


Abb.7-33: Folieneinzugsachse einer Schlauchbeutel-Applikation

In dieser Applikation wird das Produkt von einem Eintahtband in die Verpackungsmaschine eingeführt. Die Folieneinzugsachse (Sideséal-Achse) läuft synchron mit dem Eintahtband und zieht die Folie ein. Dabei muss die korrekte Ausrichtung der Folie auf das Produkt und eine gleichbleibende Produktlänge gewährleistet sein.

Schlupf zwischen dem Material und den Einzugsrollen sowie Ungenauigkeiten im Material können Positionierfehler zwischen dem Servomotor und der Folie hervorrufen. Daher sind Registermarken auf das Material aufgedruckt, die einen Ausgleich von Positionierfehlern ermöglichen.

Bedruckung, Bilder und Schmutz auf dem Produkt können Störsignale am Druckmarkensensoreingang hervorrufen. Deshalb ist ein Erwartungsfenster für das Druckmarkensignal erforderlich. Die Antriebsmesstasterfunktion ist nur innerhalb des Erwartungsfensters aktiv um die Druckmarkenposition zu erfassen. Des weiteren ist es notwendig, dass auch das Fehlen von Marken innerhalb des Erwartungsfensters erkannt wird.

Die Korrekturbewegung verhält sich proportional zur Differenz zwischen dem gemessenen und dem Sollwert. Aufgrund mechanischer Begrenzungen der Maschine muss die Korrekturbewegung auf einen nutzerdefinierten Wert begrenzt werden.

Der Registerregler muss die Folieneinzugsachse an die bedruckte Folie anpassen, um diese an die Produktposition anzupassen.

Die Folieneinzugsachse läuft im Winkelsynchronisationsmodus. Der Registerregler errechnet einen additiven Lagesollwert für die Folieneinzugsachse, durch den die Folie wieder korrekt ausgerichtet wird.

Die wichtigsten Einstellungen für den Messtaster, den Registerregler und die Antriebsbetriebsart dieser Applikation sind:

- Gemessener Wert (Messtasterfunktion) = resultierende Leitachsposition (P-0-0775)
- Geregelter Wert (ControlledValueIDN) = Lagesollwert additiv (P-0-0691)
- Antriebsbetriebsart = Winkelsynchronisation

7.3 Seitenregisterregler-FB

7.3.1 Überblick

Dieser Abschnitt beschreibt den Seitenregisterregler-FB, der Teil der Bibliothek ML(X)_TechRegi sind.

7.3.2 MB_RegisterControllerSideType01

Kurzbeschreibung Der Registerregler-FB MB_RegisterControllerSideType01 unterstützt die folgende Funktionalität:

- Berechnung eines Korrekturwerts auf der Grundlage sowohl gemessener Werte als auch Sollwerte unter Verwendung eines P- oder PI-Regelkreises
- Preset-Funktion
- Pause-Funktion
- Min/Max-Begrenzung des berechneten Stellwerts
- Sollwertübernahme-Funktion
- Totzone K_p Kennlinie
- PT1-Element

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechRegi.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechRegi.lib

Abb. 7-34: Referenztable MB_RegisterControllerSideType01

ML_TechRegi.library/MX_TechRegi.lib

Schnittstellenbeschreibung

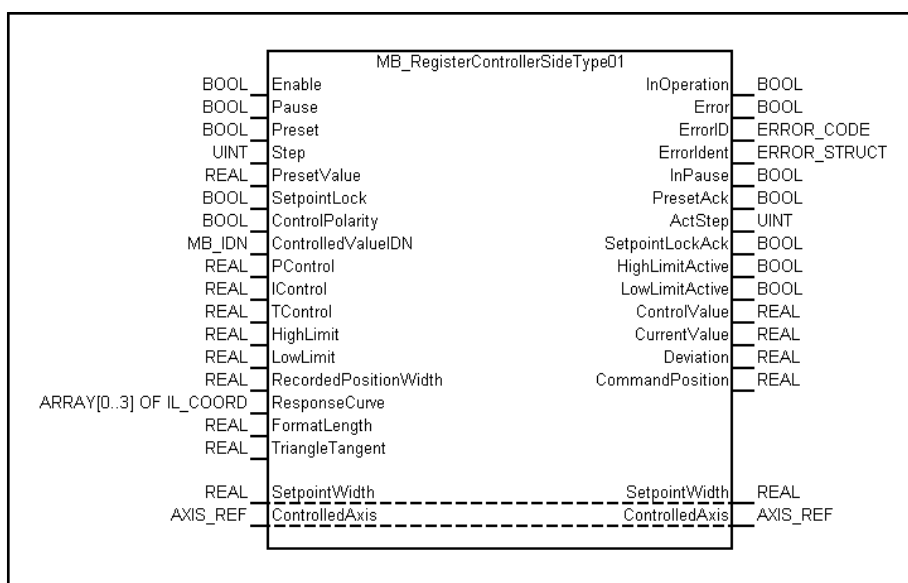


Abb. 7-35: Funktionsbaustein MB_RegisterControllerSideType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	SetpointWidth	REAL	Wert der Sollwertbreite (in technischen Einheiten)
	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse
VAR_INPUT	Enable	BOOL	Der Funktionsbaustein ist aktiv, solange der Eingang TRUE ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist. Die Ausgänge "ControlValue" und "CommandPosition" sind konstant, solange "Pause" aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt den Regelkreis zwangsweise auf "Preset-Value"
	Step	UINT	Verarbeitungsschritt
	PresetValue	REAL	Wert an den Ausgang "ControlValue" bei steigender Flanke am "Preset"-Eingang übergeben
	SetpointLock	BOOL	Aktiviert die Sollwertübernahme. Wenn TRUE, wird die nächste gemessene Marke gelatcht
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist
	ControlledValueIDN	MB_IDN	Der folgende IDN-Regelung wird unterstützt: S-0-0258
	PCControl	REAL	Proportionalverstärkung des PI-Reglers. Bei Einstellung 0 wird der proportionale Anteil des Reglers deaktiviert und die proportionale Verstärkung intern auf 1 gesetzt, sodass er als I-Regler arbeitet
	IControl	REAL	Nachstellzeit T_n des PI-Reglers. Bei der Einstellung auf 0 wird der Nachstellanteil des PI-Reglers deaktiviert

ML_TechRegi.library/MX_TechRegi.lib

E/A-Typ	Name	Datentyp	Beschreibung
	TControl	REAL	Filterkonstante für den Eingang "RecordedPosition-Width"
	HighLimit	REAL	Maximalwert für "ControlValue"
	LowLimit	REAL	Minimalwert für "ControlValue"
	RecordedPosition-Width	REAL	Breite der gemessenen Marke (in technischen Einheiten)
	ResponseCurve	ARRAY[0..3] OF IL_COORD	Kennlinie
	FormatLength	REAL	Formatlänge in mm
	TriangleTangent	REAL	Wert von $\tan \alpha$ = (Länge der Gegenkathete / Länge der Ankathete)
VAR_OUTPUT	InOperation	BOOL	Seitenregisterregler ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	InPause	BOOL	"Pause" ist aktiv. Die Ausgänge "ControlValue" und "Command" behalten ihre Werte
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in den Ausgang "ControlValue" kopiert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ControlValue" hoch. Hinweis: Der Prozessregler wird angehalten und der Ausgang "ActStep" zählt immer noch mit jedem neuen Wert am Eingang "Step" hoch
	SetpointLockAck	BOOL	Sollwert wird übernommen. Der aktuelle Wert in "RecordedPositionWidth" wird nach "SetpointWidth" kopiert
	HighLimitActive	BOOL	Obergrenze ist aktiv. "ControlValue" wäre größer, ist aber auf "HighLimit" beschränkt
	LowLimitActive	BOOL	Untergrenze ist aktiv. ControlValue wäre niedriger, ist aber auf "LowLimit" beschränkt
	ControlValue	REAL	Vom Registrierregler errechneter Stellwert
	CurrentValue	REAL	Aufgezeichnete Lage, die aktuell in der Registerreglerseite verwendet wird. Hinweis: Umgewandelter Wert „RecordedPosition-Width"
	Deviation	REAL	Differenz der Sollwertseite (umgewandelt "Setpoint-Width") - "CurrentValue" des Seitenregisterreglers (mm)
	CommandPosition	REAL	Berechnete Sollposition für den Antrieb

Abb.7-36: Schnittstellenvariablen Funktionsbaustein MB_RegisterControllerSideType01

ML_TechRegi.library/MX_TechRegi.lib

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	"Enable" TRUE (kontinuierlich)
Step	UINT	0	65335	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
SetpointLock	BOOL	n.def	n.def	FALSE	Kontinuierlich
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
PControl	REAL	0.0	n.def	1.0	Kontinuierlich
IControl	REAL	0.0	n.def	0.0	Kontinuierlich
TControl	REAL	0	n.def	0.0	Kontinuierlich
HighLimit	REAL	LowLimit	n.def	0.0	Kontinuierlich
LowLimit	REAL	n.def	HighLimit	0.0	Kontinuierlich
RecordedPositionWidth	REAL	n.def	n.def	0.0	Änderung von "Step"
ResponseCurve	ARRAY[0..3] OF IL_COORD	n.def	n.def	0.0	Kontinuierlich
FormatLength	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"
TriangleTangent	REAL	0.0	n.def	1.0	Steigende Flanke an "Enable"

Abb. 7-37: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm

Das folgende Diagramm zeigt den Signal-Zeit-Verlauf des Funktionsbausteins MB_RegisterControllerSideType01 einschließlich der Funktionen "Pause" und "Preset".

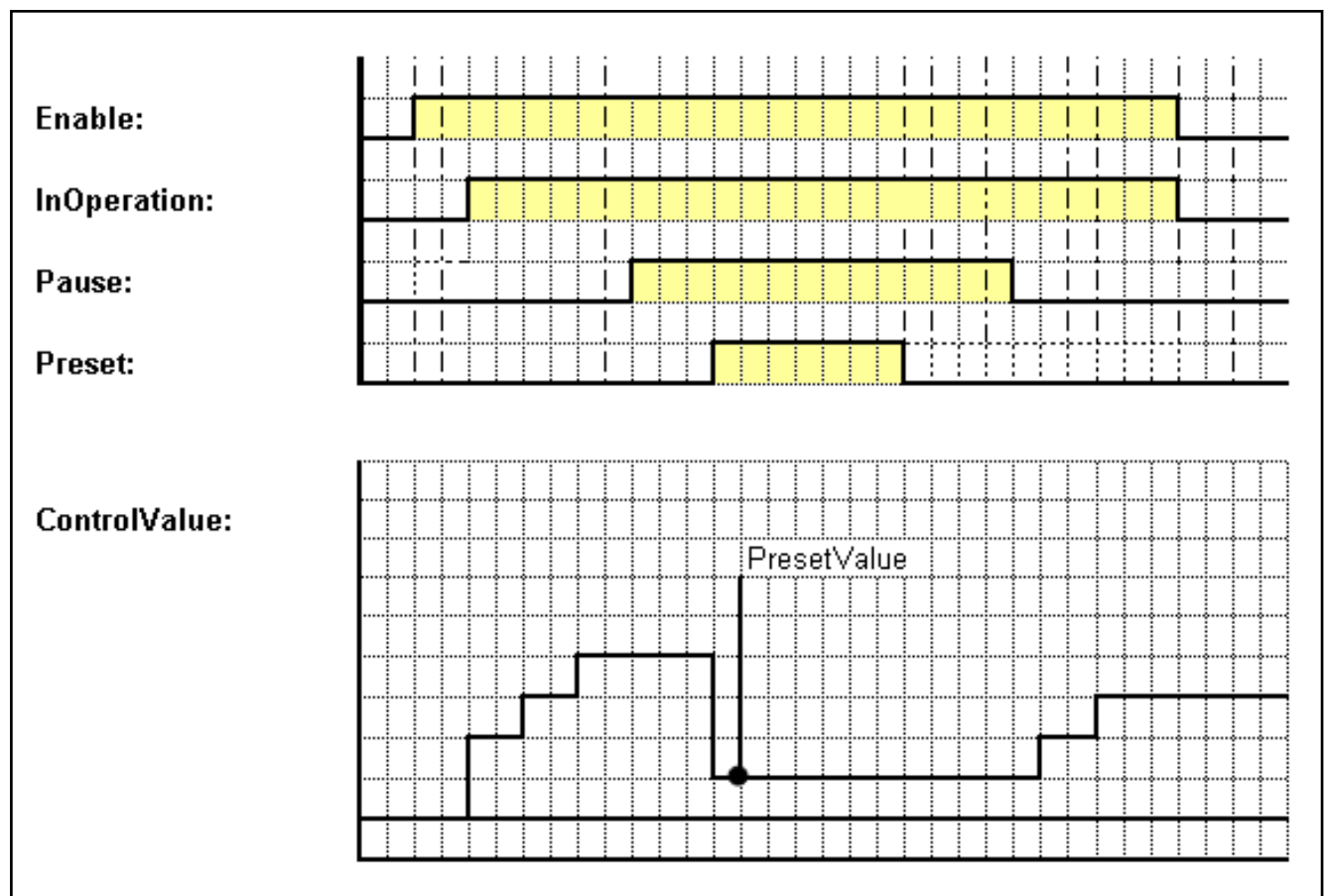


Abb.7-38: Zeitdiagramm des MB_RegisterControllerSideType01 mit den Funktionen "Pause" und "Preset"

Funktionsbeschreibung

- Initialisierung** Während der Initialisierung müssen die Eingänge "SetpointWidth" und "RecordedPositionWidth" über gültige Daten verfügen.
- PresetValue** Für einen fehlerfreien Start der Reglerausgänge "ControlValue" und "CommandPosition", muss der "Preset"-Eingang gesetzt sein, bevor der "Enable"-Eingang aktiviert wird. Während dieser Zeit müssen die Eingänge "SetpointWidth" und "RecordedPositionWidth" über gültige Daten verfügen. Zudem werden die oberen und unteren Grenzwerte überwacht, um eine Begrenzung des Ausgangs "ControlValue" sicherzustellen, wenn der aktuelle Wert für "PresetValue" außerhalb des Gültigkeitsbereiches von "HighLimit" bzw. "LowLimit" liegt.
- Der "PresetAck"-Ausgang ist aktiv wenn der Wert im Eingang "PresetValue" in den Ausgang "ControlValue" kopiert wird. Der "Preset"-Eingang hat eine höher Priorität als der "Pause"-Eingang. Der Wert in "PresetValue" wird in den entsprechenden Ausgang bei steigender Flanke am "Preset"-Eingang kopiert wenn "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt.
- Pause** Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. In diesem Fall wird die Abweichung ("SetpointWidth" – "RecordedPositionWidth") auf 0 gesetzt. Dadurch ändern sich die Ausgänge "ControlValue" und "CommandPosition" nicht mehr. Diese kann verwendet werden, um die Werte der Ausgänge "ControlValue" und "CommandPosition" beizubehalten. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Ein-

ML_TechRegi.library/MX_TechRegi.lib

gang "Pause" aktiv ist. Solange der Eingang "Pause" aktiv ist, ist der Eingang "InPause" TRUE.

Glättung Der Eingang "TControl" ist die Dämpfungskonstante für den Eingang "RecordedPositionWidth". Wenn er auf 0 gesetzt wird, ist die Glättung inaktiv.

DeadBand Das Differenzsignal des gefilterten Eingangs "RecordedPositionWidth" und des Eingangs "SetpointWidth" wird durch den Eingang "ResponseCurve" gewichtet. Eine Kennlinie wird verwendet, um den Registerregler innerhalb kleiner Registerfehler anzuhalten.

FormatLength Formatlänge (Leitachsenumdrehungen, Formate) in mm.

TriangleTangent Die Marken zwischen Gegenkathete und Ankathete ($\tan \alpha$).

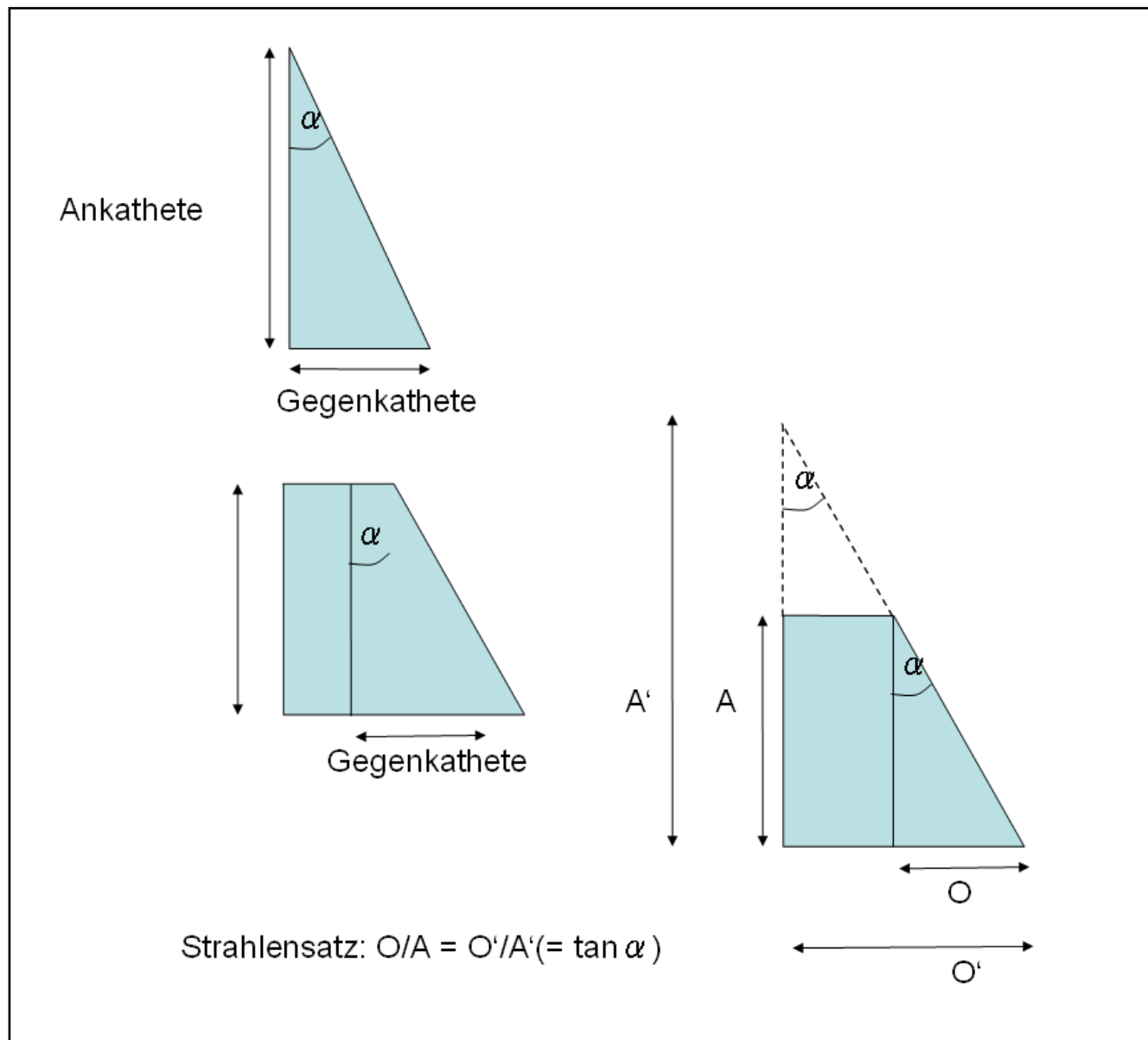


Abb. 7-39: Rechtwinklige Dreiecksmarke und konisch abgestumpfte Kegelmarke

Normalbetrieb Die Registerreglerseite berechnet nur einen neuen Wert wenn sich der Eingang "Step" verändert. Die Reglerparameter "PControl", "IControl" und "TControl" können permanent geändert werden.

Wenn der Wert im "ControlValue"-Ausgang die Werte von "HighLimit" oder "LowLimit" über-, bzw. unterschreitet, dann wird "ControlValue" auf einen der folgenden Grenzwerte begrenzt.

- Ist der Wert in "ControlValue" positiv, wird der Wert in "HighLimit" verwendet
- Ist der Wert in "ControlValue" negativ, wird der Wert in "LowLimit" verwendet

Wenn die Registerreglerseite eine Nachstellzeit hat, wird zusätzlich ein Anti-Reset-Windup durchgeführt und die Nachstellzeit wird nicht länger aufsummiert. In diesem Fall werden die Ausgänge "HighLimitActive" und "LowLimitActive" auf TRUE gesetzt.

Fehlerbehandlung

Der Funktionsbaustein gibt die Fehlermeldungen der folgenden unterlagerten Funktionsbausteine aus.

- MB_RegiMeasuringType01 ([Abb. 7-51 " Fehlercodes Funktionsbaustein MB_RegiMeasuringType01" auf Seite 344](#))
- IL_PIType02 (Bibliothek RIL_LoopControl, siehe Dokumentation "Rexroth IndraWorks 12VRS Basis-Bibliotheken")
- MB_RegiRegulateType01 ([Abb. 7-72 " Fehlercodes Funktionsbaustein MB_RegiRegulateType01" auf Seite 355](#))
- MB_RegiSetpointLockType01 ([Abb. 7-104 " Fehlercodes Funktionsbaustein MB_RegiSetpointLockType01" auf Seite 382](#))

Er erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von "ControlledAxis" ist außerhalb des gültigen Wertebereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Achse hat keine Stromversorgung "ControlledAxis"
RESOURCE_ERROR (16#0003)	16#0009	16#0000	"Axis_Ref" hat "ControlledAxis" geändert
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Ausgewählte "ControlledValueIDN" wird nicht unterstützt
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0001	"HighLimit" < "LowLimit"
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0002	"PControl" < 0
INPUT_RANGE_ERROR (16#0006)	16#0816	16#0001	Eingang "SetpointWidth" < 0
INPUT_RANGE_ERROR (16#0006)	16#0816	16#0002	Eingang "RecordedPositionWidth" < 0
INPUT_RANGE_ERROR (16#0006)	16#0816	16#0003	"TriangleTangent"-Eingang ist außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0816	16#0004	"FormatLength"-Eingang ist außerhalb des gültigen Wertebereichs
INPUT_INVALID_ERROR (16#0001)	16#0C01	16#0001	"LowLimit" > "HighLimit"

ML_TechRegi.library/MX_TechRegi.lib

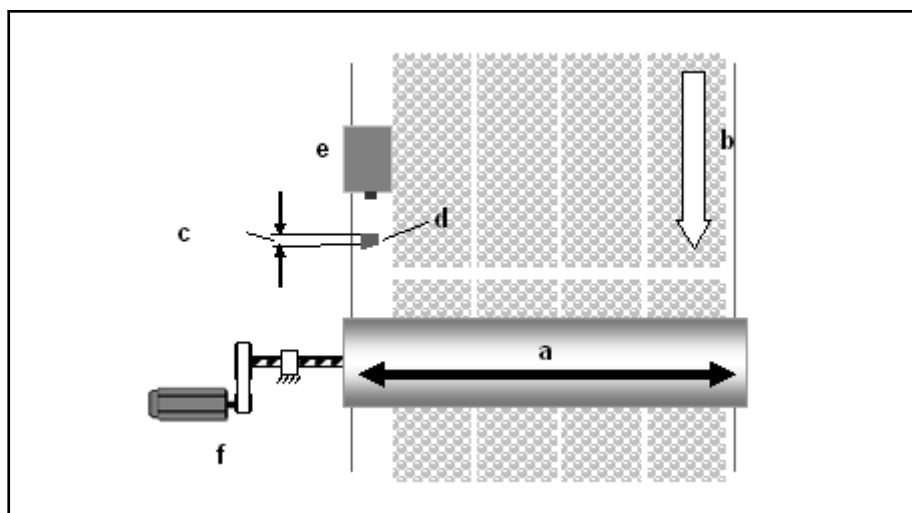
ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR (16#0001)	16#0C01	16#0002	Auswahl des Reglertyps wird nicht unterstützt. Eingänge "IControl" und "PControl" überprüfen
INPUT_RANGE_ERROR (16#0006)	16#0C02	16#0001	Eingang "TControl" ist ungültig
INPUT_RANGE_ERROR (16#0006)	16#0C02	16#0003	Eingang "IControl" ist ungültig
INPUT_RANGE_ERROR (16#0006)	16#0C02	16#0004	Eingang "PControl" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0C05	16#0001	X-Werte in "ResponseCurve" nicht aufsteigend
INPUT_INVALID_ERROR (16#0001)	16#0C05	16#0002	Mehr als einen Wert für eine X-Position in "ResponseCurve"
INPUT_INVALID_ERROR (16#0001)	16#0C05	16#0003	Negativer X-Wert in "ResponseCurve"
ACCESS_ERROR (16#0004)	16#0815	16#0008	Die geregelte Achse ist eine IndraMotion MLD-M Folgeachse, aber der Parameter P-0-1367, Bit 6 ist nicht TRUE

Abb. 7-40: Fehlercodes Funktionsbaustein MB_RegisterControllerSideType01

7.3.3 Konfiguration des Seitenregisters

Überblick Der Seitenregister ist eine Funktion zur axialen Anpassung der rotierenden Zylinder. Typisches Beispiel ist die Anpassung eines Druckzylinders für eine vorgedruckte Papierbahn.

Die Dreiecksmarke ist der axiale Bezug auf dem gedruckten Image. Ein optischer Sensor registriert die eingehende und ausgehende Flanke der Marke. Der Unterschied zwischen beiden Flanken ist die Breite der Marke. Der Wert ist der Istwert des Seitenregisters. Der Unterschied zwischen der Istbreite der Marke und der Sollbreite wird in die Zielposition des Servomotors überführt.



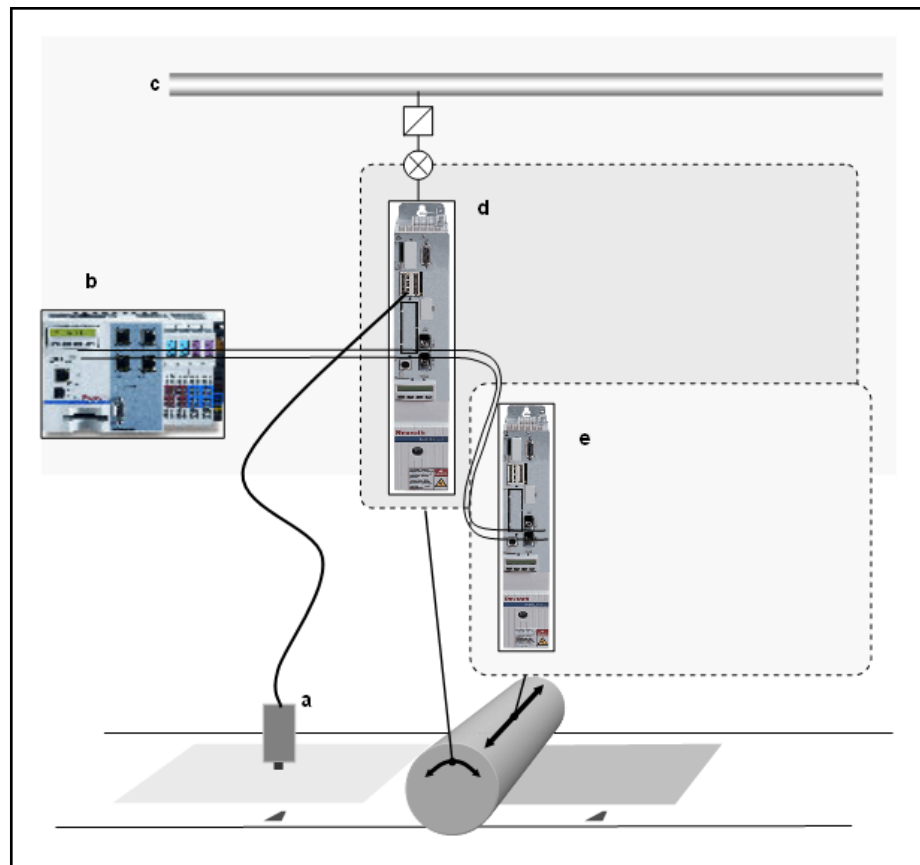
- a: Seitenregister
- b: Bahnaufrichtung
- c: Gemessener Wert
- d: Registermarke
- e: Sensor
- f: Servomotor

Abb. 7-41: Seitenregisterkonfiguration

Verwendung der Seitenregisterregelung

Die nachfolgende Abbildung zeigt eine mögliche Konfiguration eines Seitenregisterreglers für eine IndraMotion MLC.

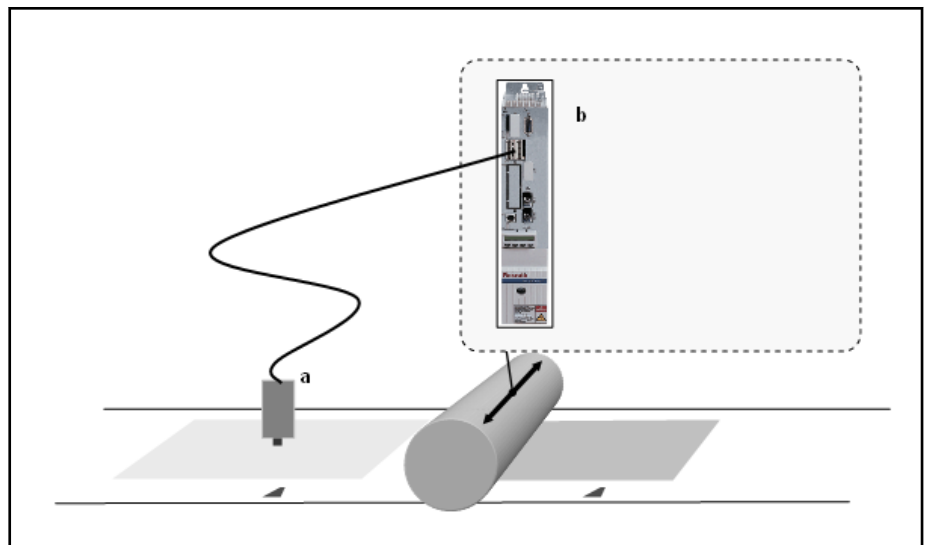
ML_TechRegi.library/MX_TechRegi.lib



- a: Sensor
- b: IndraMotion MLC
- c: Leitachse
- d: Registerregler, Messtasterfunktion, Winkelsynchronisation
- e: Seitenregisterregler, absolutes Positionieren

Abb.7-42: Überblick über Registerregler und Seitenregisterregler auf der IndraMotion MLC

Die nachfolgende Abbildung zeigt die Konfiguration eines Seitenregisterreglers für eine IndraMotion MLD.



- a: Sensor
b: IndraMotion MLD: Messtasterfunktion, Seitenregisterregler, absolutes Positionieren

Abb. 7-43: Überblick über den Seitenregisterregler in einer IndraMotion MLD



Die axiale Achse muss eine absolute Linearachse sein.

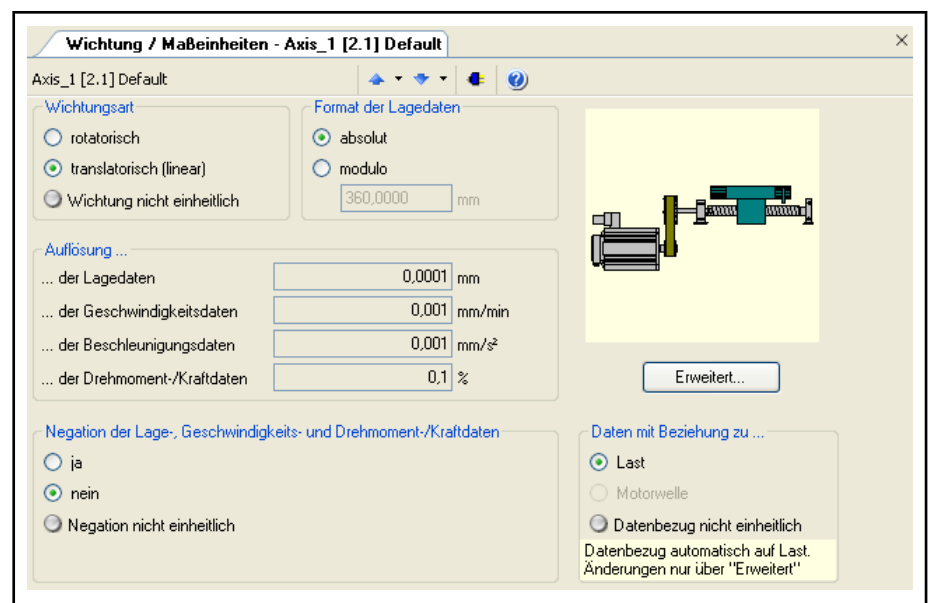


Abb. 7-44: Wichtung der geregelten Achse des Seitenregisterreglers

Beispielkonfiguration

Die schnellste Möglichkeit eine Seitenregisterreglerfunktionalität in einer Anwendung einzusetzen, ist die Kombination des Messtaster-FB mit dem AchsInterface. Der Messtaster-FB liefert den aktuellen Wert an den Seitenregister. Der Seitenregisterregler berechnet die Ziellposition für die Achse. Der Seitenregister verfügt über keine interne Funktionalität um die Lage auf einen Sollwertlageparameter der Achse zu schreiben. Daher wird das AchsInterface verwendet, um die Sollage auf den entsprechenden Achsparameter zu schreiben.

Dieses Beispiel zeigt den Seitenregisterregler MB_RegisterControllerSide-Type01 mit dem Funktionsbaustein MB_TouchProbeContinuous und den Funktionsbausteinen des AchsInterface.

ML_TechRegi.library/MX_TechRegi.lib

1. Ausgang "PositionDifference" zu Eingang "RecordedPositionWidth"
2. Ausgang "DetectedPositionDifference" zu Eingang "Step"

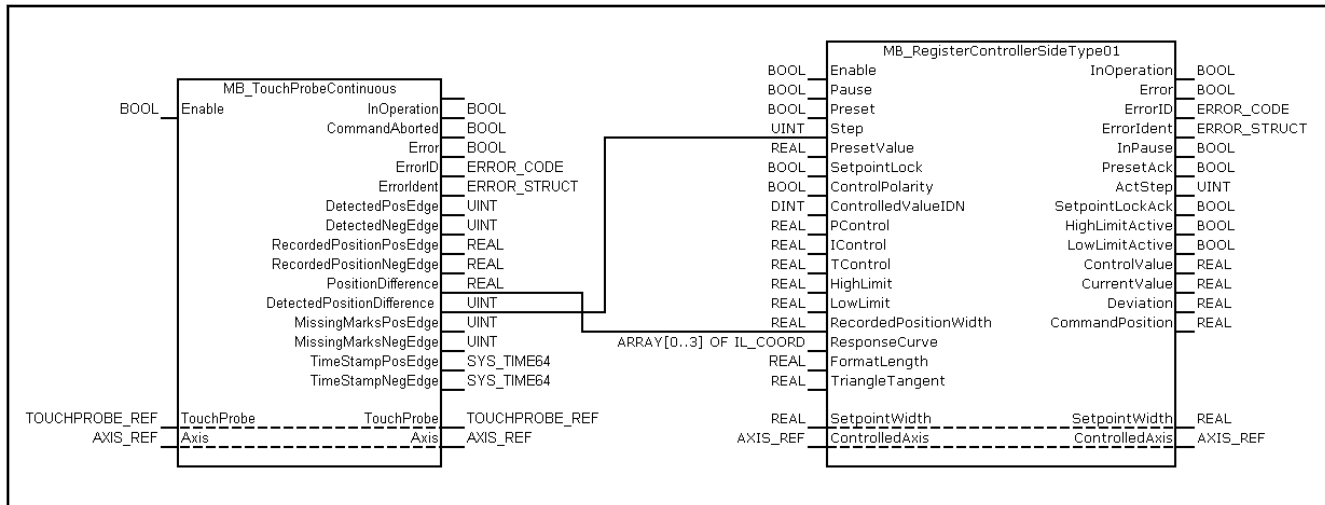


Abb.7-45: Schematische Darstellung der FB *MB_TouchProbeContinuous* und *MB_RegisterControllerSideType01*

3. Weisen Sie "CommandPosition" dem Array `arAxisCtrl[ControlledAxis].PosMode.Position` zu



AchsInterface-Einstellungen

- Betriebsmodus: Absolutes Positionieren
- Folgendes Kommando aktiviert den Betriebsmodus:
`arAxisCtrl[ ].Admin._OpMode := ModePosAbs;` bzw. `arAxisCtrl[ ].Admin.MODE_POS_ABS := TRUE;`
- EnableCyclicScanning muss aktiviert werden

Mehr Informationen finden Sie in der Dokumentation "Rexroth IndraMotion xxVRS Generic Application Template" in Kapitel "ML_TechInterface.library".

xx - Version

Programm:

```
fbMB_RegisterControllerSideType01(
    Enable:= bEnableRegi,
    Pause:= bPauseRegi,
    Preset:= bPresetRegi,
    Step:= uiDetectedPositionDifference,
    PresetValue:= bPresetValueRegi,
    SetpointLock:= bSetpointLock,
    ControlPolarity:= bControlPolarity,
    ControlledValueIDN:= diControlledValueIDN,
    PControl:= rPControl,
    IControl:= rIControl,
    TControl:= rTControl,
```

ML_TechRegi.library/MX_TechRegi.lib

```

HighLimit:= rHighLimit,
LowLimit:= rLowLimit,
RecordedPositionWidth:= rPositionDifference,
ResponseCurve:= stResponseCurve,
FormatLength:= rForamtLength,
TriangleTangent:= rTriangleTangent,
SetpointWidth:= rSetpointWidth,
ControlledAxis:= Drive5,
InOperation=> bInOperationRegi,
Error=> bErrorRegi,
ErrorID=> stErrorIDRegi,
ErrorIdent=> stErrorIdentRegi,
InPause=> bInPauseRegi,
PresetAck=> bPresetAckRegi,
ActStep=> uiActStepRegi,
SetpointLockAck=> bSetpointLockAck,
HighLimitActive=> bHighLimitActive,
LowLimitActive=> bLowLimitActive,
ControlValue=> rControlValue,
CurrentValue=> rCurrentValue,
Deviation=> rDeviation,
CommandPosition=>CommandPosition );

fbMB_AxisInitType01(
    Execute:= bExecuteAxisInit,
    ReadModuloValue:= FALSE,
    EnableCyclicScanning:= TRUE,
    AdminCtrl:= arAxisCtrl[Drive5.AxisNo].Admin,
    PosModeCtrl:= arAxisCtrl[Drive5.AxisNo].PosMode,
    VelModeCtrl:= arAxisCtrl[Drive5.AxisNo].VelMode,
    SyncModeCtrl:= arAxisCtrl[Drive5.AxisNo].SyncMode,
    AxisDataPtrStatus:= arAxisStatus[Drive5.AxisNo].PtrAxisData,
    AdminStatus:= arAxisStatus[Drive5.AxisNo].Admin,
    DiagStatus:= arAxisStatus[Drive5.AxisNo].Diag,
    Axis:= Drive5,
    Done=> bDoneInit,
    Active=> bActiveInit,
    Error=> bErrorInit,
    ErrorID=> stErrorIDInit,
    ErrorIdent=> stErrorIdent);

(* Lagebefehl zum Antrieb der Seitenregisterregelung *)

```

ML_TechRegi.library/MX_TechRegi.lib

```

arAxisCtrl[Drive5.AxisNo].PosMode.Position := CommandPosition;
fbMB_AxisInterfaceType01(
    AdminCtrl:= arAxisCtrl[Drive5.AxisNo].Admin,
    PosModeCtrl:= arAxisCtrl[Drive5.AxisNo].PosMode,
    VelModeCtrl:= arAxisCtrl[Drive5.AxisNo].VelMode,
    SyncModeCtrl:= arAxisCtrl[Drive5.AxisNo].SyncMode,
    AdminStatus:= arAxisStatus[Drive5.AxisNo].Admin,
    DiagStatus:= arAxisStatus[Drive5.AxisNo].Diag,
    PosStatus:= arAxisStatus[Drive5.AxisNo].PosMode,
    VelStatus:= arAxisStatus[Drive5.AxisNo].VelMode,
    SyncModeStatus:= arAxisStatus[Drive5.AxisNo].SyncMode );

```

7.4 Registerregelungs-, Mess- und Stellfunktionsbausteine

7.4.1 Überblick

Dieser Abschnitt beschreibt die Funktionsbausteine zum Messen, Stellen und zur Sollwertübernahme, die dazu eingesetzt werden können kundenspezifische Registerregler-FB, Teile der Bibliothek ML(X)_TechRegi, herzustellen.

7.4.2 MB_RegiMeasuringType01

Kurzbeschreibung

Der Funktionsbaustein MB_RegiMeasuringType01 wird zur Bearbeitung des gemessenen Messtastersignals verwendet.

Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechRegi.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechRegi.lib

Abb. 7-46: Referenztabelle MB_RegiMeasuringType01

Schnittstellenbeschreibung

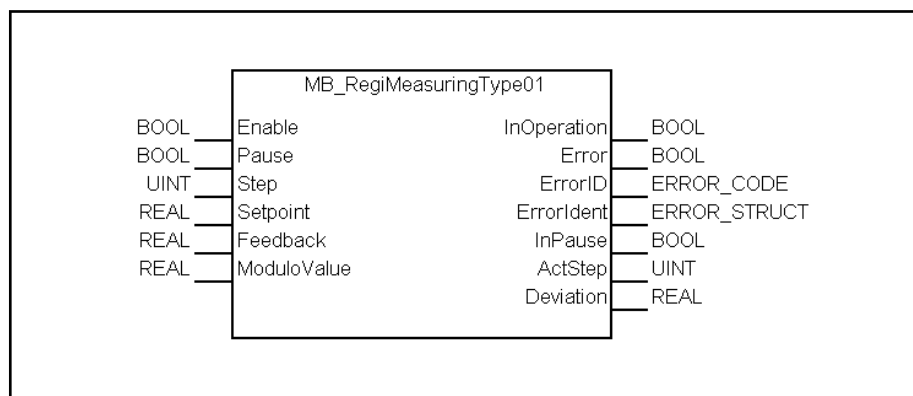


Abb. 7-47: Funktionsbaustein MB_RegiMeasuringType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist

E/A-Typ	Name	Datentyp	Beschreibung
	Step	UINT	Neue Werte werden berechnet wenn sich der "Step"-Eingang ändert
	Setpoint	REAL	Sollwert
	Feedback	REAL	Gemessener Wert
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
VAR_OUTPUT	InOperation	BOOL	Messfunktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt an, dass es zu einem Fehler während des Funktionsbausteinbetriebes kam
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "Deviation" behält seinen Wert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "Deviation" hoch
	Deviation	REAL	"Setpoint" - "Feedback"

Abb. 7-48: Schnittstellenvariablen Funktionsbaustein MB_RegiMeasuringType01

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Baustein-
neingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
Setpoint	REAL	0	ModuloValue	0.0	Änderung von "Step"
Feedback	REAL	0	ModuloValue	0.0	Änderung von "Step"
ModuloValue	REAL	0	n.def	0.0	Steigende Flanke an "Enable"

Abb. 7-49: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Zeitdiagramm gemäß der Spezifikation PLCopen.

ML_TechRegi.library/MX_TechRegi.lib

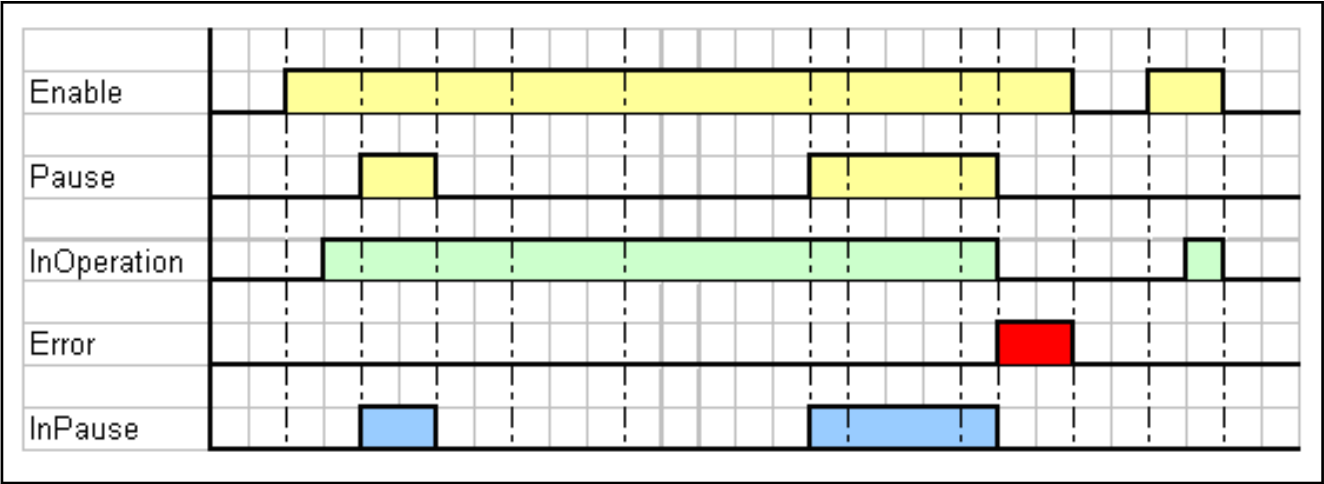


Abb. 7-50: Zeitdiagramm des MB_RegiMeasuringType01

- Funktionsbeschreibung

Der Funktionsbaustein MB_RegiMeasuringType01 subtrahiert die Eingänge "Setpoint" und "Feedback" voneinander ("Setpoint" - "Feedback"). Das Ergebnis der Subtraktion ist der Wert des Ausgangs "Deviation".
- Pause

Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "Deviation"-Ausgang behält seinen Wert.
- Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("Axis_Ref") wurde geändert während der FB in Betrieb war
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0002	Eingang "Setpoint" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0003	Eingang "Setpoint" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0004	Eingang "Feedback" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0005	Eingang "Feedback" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0006	Eingang "ModuloValue" < 0

Abb. 7-51: Fehlercodes Funktionsbaustein MB_RegiMeasuringType01

7.4.3 MB_RegiMeasuringType02

- Kurzbeschreibung

Der Funktionsbaustein MB_RegiMeasuringType02 wird zur Bearbeitung des gemessenen Messtastersignals verwendet.

ML_TechRegi.library/MX_TechRegi.lib

Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechRegi.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechRegi.lib

Abb. 7-52: Referenztable MB_RegiMeasuringType02

Schnittstellenbeschreibung

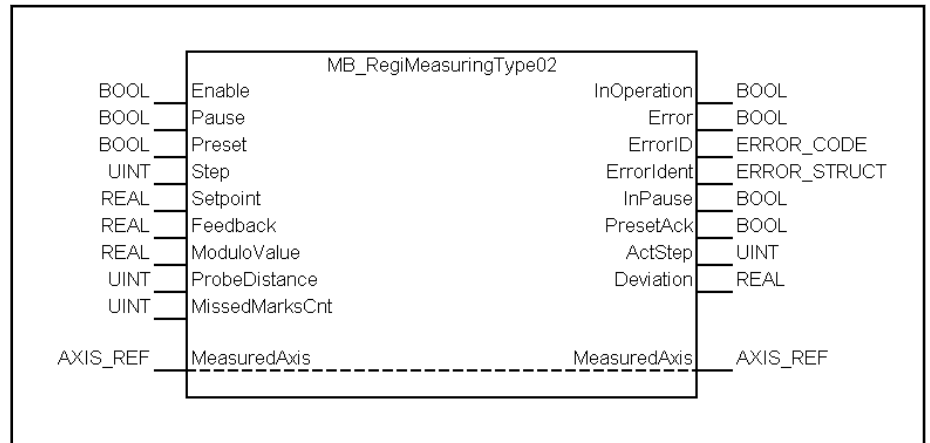


Abb. 7-53: Funktionsbaustein MB_RegiMeasuringType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt den Speicher des Schieberegisters zwangsweise auf "0"
	Step	UINT	Neue Werte werden berechnet wenn sich der "Step"-Eingang ändert
	Setpoint	REAL	Sollwert
	Feedback	REAL	Gemessener Wert
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
	ProbeDistance	UINT	Der Abstand zwischen dem Markensensor und der Mess- (= geregelten) Achse in Leitachsumdrehungen (Formatlängen)
	MissedMarksCnt	UINT	Zähler für Markenausfälle, wenn ein Erwartungsfenster verwendet wird
VAR_OUTPUT	InOperation	BOOL	Messfunktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt an, dass es zu einem Fehler während des Funktionsbausteinbetriebes kam
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

ML_TechRegi.library/MX_TechRegi.lib

E/A-Typ	Name	Datentyp	Beschreibung
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "Deviation" behält seinen Wert
	PresetAck	BOOL	Preset-Funktion ist abgeschlossen
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "Deviation" hoch
	Deviation	REAL	"Setpoint" - "Feedback" verzögert wenn Schieberegister aktiv ist

Abb. 7-54: Schnittstellenvariablen Funktionsbaustein MB_RegiMeasuringType02

Min.- Max.- und Default-Werte der Eingänge
In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	"Enable" TRUE (kontinuierlich)
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
Setpoint	REAL	0	ModuloValue	0.0	Änderung von "Step"
Feedback	REAL	0	ModuloValue	0.0	Änderung von "Step"
ModuloValue	REAL	0	n.def	0.0	Steigende Flanke an "Enable"
ProbeDistance	UINT	0	30	0	Kontinuierlich
MissedMarksCnt	UINT	0	65.535	0	Kontinuierlich

Abb. 7-55: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Zeitdiagramm gemäß der Spezifikation PLCopen.

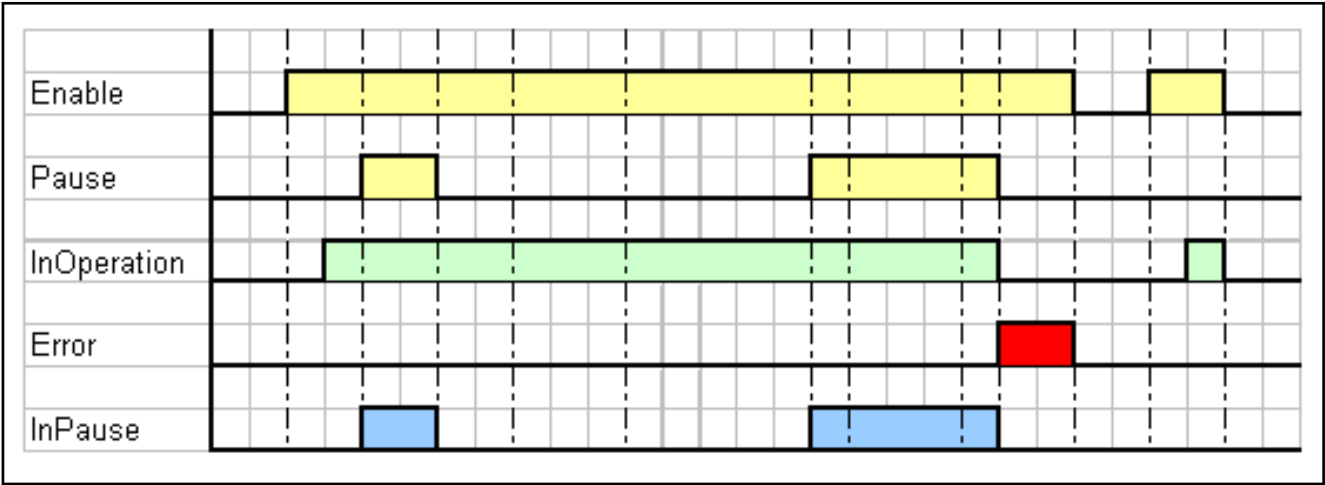


Abb. 7-56: Zeitdiagramm des MB_RegiMeasuringType02

Funktionsbeschreibung Der Funktionsbaustein MB_RegiMeasuringType02 subtrahiert die Eingänge "Setpoint" und "Feedback" voneinander ("Setpoint" - "Feedback"). Das Ergebnis der Subtraktion ist der Wert des Ausgangs "Deviation" wenn das Schieberegister deaktiviert ist. Wenn das Schieberegister aktiviert ist, wird es um die Anzahl der "ProbeDistance" verzögert.

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "Deviation"-Ausgang behält seinen Wert.

Preset Das Schieberegister kann vorab mit dem Eingang "Preset" beschrieben werden. Damit der Schieberegisterspeicher vorab beschrieben wird, muss "Enable" aktiv sein. Ein Nullwert wird in den Speicher bei ansteigender Flanke des Eingangs "Preset" geschrieben wenn "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt.

ProbeDistance Der Eingang "ProbeDistance" ist der Abstand zwischen dem Markensensor und der Mess- (= geregelten) Achse, in Leitachsumdrehungen (Formatlängen). Die maximale Formatlänge beträgt 30. Ist "ProbeDistance" 0, ist das Schieberegister deaktiviert. Der Ausgang "Deviation" wird auf "Setpoint" - "Feedback" gesetzt.

Wenn "ProbeDistance" nicht 0 ist, wird das Schieberegister aktiviert. Das bedeutet, dass die Werte beim Ausgang "Deviation" um die Anzahl der Leitachsdrehungen (Formatlängen) am Eingang "ProbeDistance" verzögert werden.

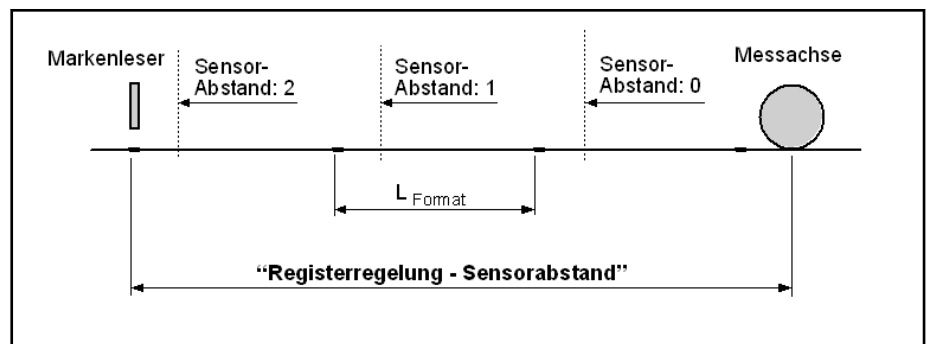


Abb. 7-57: Eingang „ProbeDistance“ setzen

MissedMarkCnt Das ist der Markenausfallüberwachungszähler der Messtasterfunktion wenn das Erwartungsfenster aktiv ist. Wenn eine Marke fehlt, wird eine Null in das Schieberegister geschrieben wenn es aktiviert ist. Im Falle eines deaktivierten Schieberegisters wird die Null in den Ausgang "Deviation" geschrieben.

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("Axis_Ref") wurde geändert während der FB in Betrieb war

ML_TechRegi.library/MX_TechRegi.lib

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0001	Eingang "ProbeDistance" ist außerhalb des gültigen Wertebereichs (0 <= "ProbeDistance" <= 30)
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0002	Eingang "Setpoint" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0003	Eingang "Setpoint" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0004	Eingang "Feedback" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0005	Eingang "Feedback" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0006	Eingang "ModuloValue" < 0

Abb. 7-58: Fehlercodes Funktionsbaustein MB_RegiMeasuringType02

7.4.4 MB_RegiMeasuringType03

Kurzbeschreibung

Der Funktionsbaustein MB_RegiMeasuringType03 wird zur Bearbeitung des gemessenen Messtastersignals verwendet.

Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechRegi.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechRegi.lib

Abb. 7-59: Referenztabelle MB_RegiMeasuringType03

Schnittstellenbeschreibung

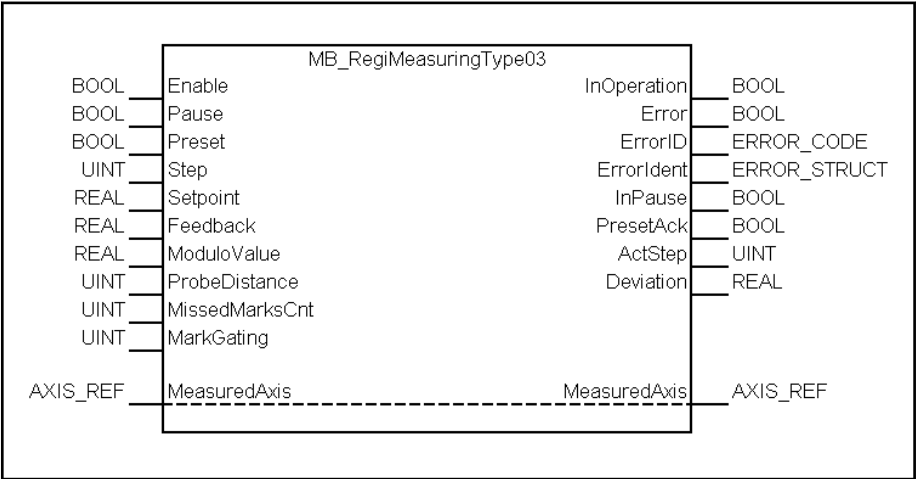


Abb. 7-60: Funktionsbaustein MB_RegiMeasuringType03

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist

E/A-Typ	Name	Datentyp	Beschreibung
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt den Speicher des Schieberegisters zwangsweise auf "0"
	Step	UINT	Neue Werte werden berechnet wenn sich der "Step"-Eingang ändert
	Setpoint	REAL	Sollwert
	Feedback	REAL	Gemessener Wert
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
	ProbeDistance	UINT	Der Abstand zwischen dem Markensensor und der Messachse in Leitachsumdrehungen (Formatlängen)
	MissedMarksCnt	UINT	Zähler der fehlenden Marken wenn ein Erwartungsfenster verwendet wird
	MarkGating	UINT	Jede n-te Marke wird für die Registrierung verwendet
VAR_OUTPUT	InOperation	BOOL	Messfunktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt an, dass es zu einem Fehler während des Funktionsbausteinbetriebes kam
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "Deviation" behält seinen Wert
	PresetAck	BOOL	Preset-Funktion ist abgeschlossen
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "Deviation" hoch
	Deviation	REAL	"Setpoint" - "Feedback" verzögert, wenn Schieberegister aktiv ist

Abb. 7-61: Schnittstellenvariablen Funktionsbaustein MB_RegiMeasuringType03

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	"Enable" TRUE (kontinuierlich)
Step	UINT	0	65.535	0	Kontinuierlich Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
Setpoint	REAL	0	ModuloValue	0.0	Änderung von "Step"
Feedback	REAL	0	ModuloValue	0.0	Kontinuierlich

ML_TechRegi.library/MX_TechRegi.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ModuloValue	REAL	0	n.def	0.0	Steigende Flanke an "Enable"
ProbeDistance	UINT	0	60	0	Kontinuierlich
MissedMarksCnt	UINT	0	65.535	0	Kontinuierlich
MarkGating	UNIT	0	65.535	1	Steigende Flanke an "Enable". Ein Wert von "0" deaktiviert den Markentakt (identisch mit dem Wert "1")

Abb. 7-62: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Zeitdiagramm gemäß der Spezifikation PLCopen.

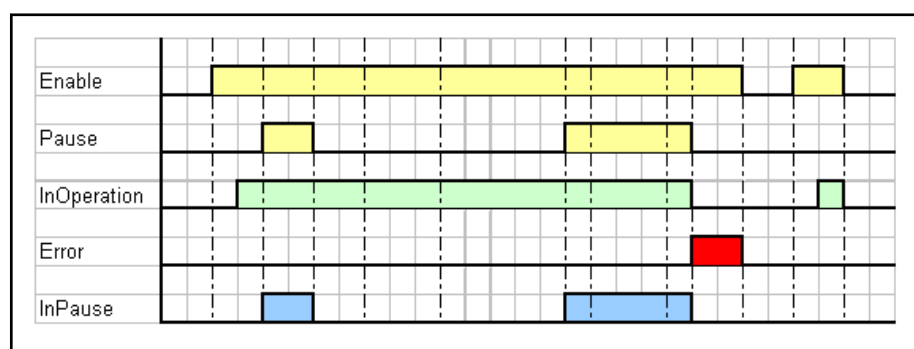


Abb. 7-63: Zeitdiagramm des MB_RegiMeasuringType03

Funktionsbeschreibung

Der Funktionsbaustein MB_RegiMeasuringType03 subtrahiert die Eingänge "Setpoint" und "Feedback" ("Setpoint" - "Feedback"). Das Ergebnis der Subtraktion ist der Wert des Ausgangs "Deviation" wenn das Schieberegister deaktiviert ist. Wenn das Schieberegister aktiviert ist, wird es um die Anzahl der "ProbeDistance" verzögert. Nur jede n-te Marke wird für die Registrierung mit der Markenausfensterungsfunktion des FB verwendet.

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "Deviation"-Ausgang behält seinen Wert.

Preset Das Schieberegister kann vorab mit dem Eingang "Preset" beschrieben werden. Damit der Schieberegisterspeicher vorab beschrieben wird, muss "Enable" aktiv sein. Ein Nullwert wird in den Speicher bei ansteigender Flanke des Eingangs "Preset" geschrieben wenn "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt.

ProbeDistance Der Eingang "ProbeDistance" ist der Abstand zwischen dem Markensensor und der Mess- (= geregelten) Achse, in Leitachsumdrehungen (Formatlängen). Die maximale Formatlänge beträgt 60. Ist "ProbeDistance" 0, ist das Schieberegister deaktiviert. Der Ausgang "Deviation" wird auf "Setpoint" - "Feedback" gesetzt.

Wenn "ProbeDistance" nicht 0 ist, wird das Schieberegister aktiviert. Das bedeutet, dass die Werte beim Ausgang "Deviation" um die Anzahl der Leitachsumdrehungen (Formatlängen) am Eingang "ProbeDistance" verzögert werden.

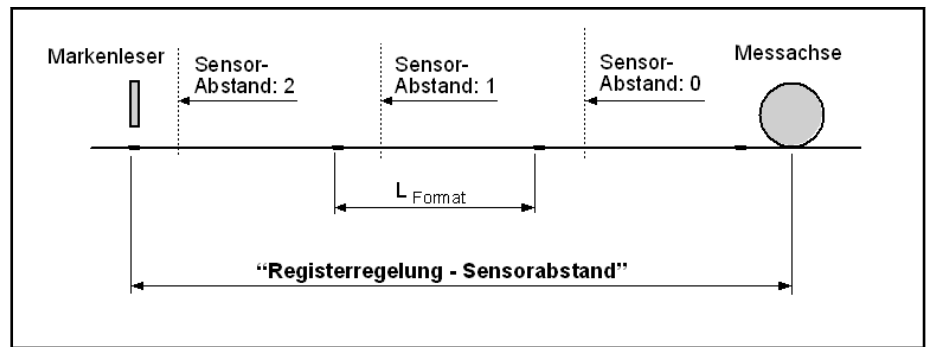


Abb.7-64: Eingang „ProbeDistance“ setzen

MissedMarkCnt

Das ist der Markenausfallüberwachungszähler der Messtasterfunktion wenn das Erwartungsfenster aktiv ist. Wenn eine Marke fehlt, wird eine Null in das Schieberegister geschrieben wenn es aktiviert ist. Im Falle eines deaktivierten Schieberegisters wird die Null in den Ausgang "Deviation" geschrieben.

MarkGating

Mit diesem Feature kann der Anwender die Registrierung nur bei jeder n-ten Marke regeln, indem er ungültige Marken filtert. Man kann dies z.B. verwenden, wenn die Marken auf der Papierbahn mit einer Druckplatte gedruckt wurden, auf dem das gedruckte Bild mehrfach erscheint. Dabei sind die Markenabstände zwischen aufeinanderfolgenden Marken nicht konstant. Die Markenabstände zwischen jeder n-ten aufeinanderfolgenden Marke sind jedoch konstant. Der RegiMeasuring-FB verwendet nur jede n-te Marke für die Registerregelung.

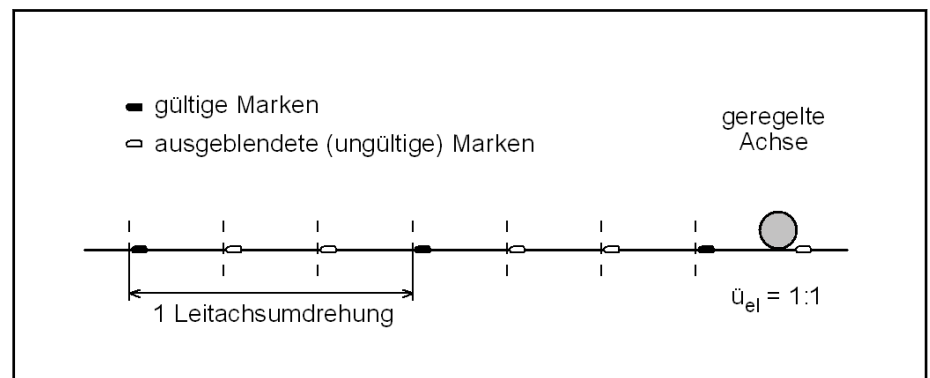


Abb.7-65: Markenausfensterung

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("Axis_Ref") wurde geändert während der FB in Betrieb war
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0001	Eingang "ProbeDistance" ist außerhalb des gültigen Wertebereichs ($0 \leq \text{"ProbeDistance"} \leq 30$)

ML_TechRegi.library/MX_TechRegi.lib

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0002	Eingang "Setpoint" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0003	Eingang "Setpoint" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0004	Eingang "Feedback" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0005	Eingang "Feedback" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0006	Eingang "ModuloValue" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0007	Eingang "MarkGating" < 0

Abb. 7-66: Fehlercodes Funktionsbaustein MB_RegiMeasuringType03

7.4.5 MB_RegiRegulateType01

Kurzbeschreibung Der Funktionsbaustein MB_RegiRegulateType01 wird verwendet, um einen Stellwert an den Antrieb zu schreiben.

Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechRegi.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechRegi.lib

Abb. 7-67: Referenztable MB_RegiRegulateType01

Schnittstellenbeschreibung

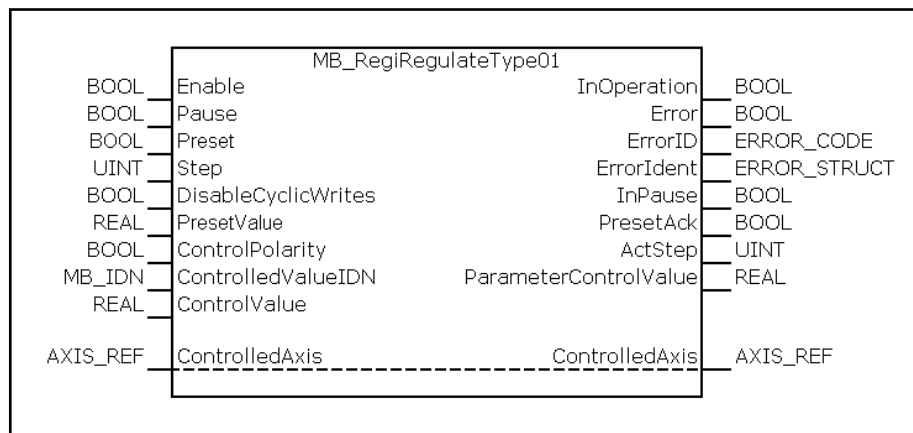


Abb. 7-68: Funktionsbaustein MB_RegiRegulateType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt "ParameterControlValue" zwangsweise auf "PresetValue"
	Step	UINT	Verarbeitungsschritt

E/A-Typ	Name	Datentyp	Beschreibung
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlledValueIDN" wenn TRUE
	PresetValue	REAL	Wert an den Ausgang "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist
	ControlledValueIDN	MB_IDN	SERCOS-IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: IndraMotion MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 IndraMotion MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	ControlValue	REAL	Berechneter Stellwert von z.B. IL_PIType02
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "ParameterControlValue" behält seinen Wert
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in den Ausgang "ParameterControlValue" kopiert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ParameterControlValue" hoch
	ParameterControlValue	REAL	Der Registerregler-FB berechnet einen Wert, der auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang basiert und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parameter, Eingang "ControlledValueIDN", geschrieben, wenn der Eingang "DisableCyclicWrites" = FALSE ist

Abb. 7-69: Schnittstellenvariablen Funktionsbaustein MB_RegiRegulateType01

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"

ML_TechRegi.library/MX_TechRegi.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
ControlValue	REAL	n.def	n.def	0.0	Kontinuierlich
ModuloValue	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"

Abb. 7-70: Übersicht über Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Zeitdiagramm gemäß der Spezifikation PLCopen.

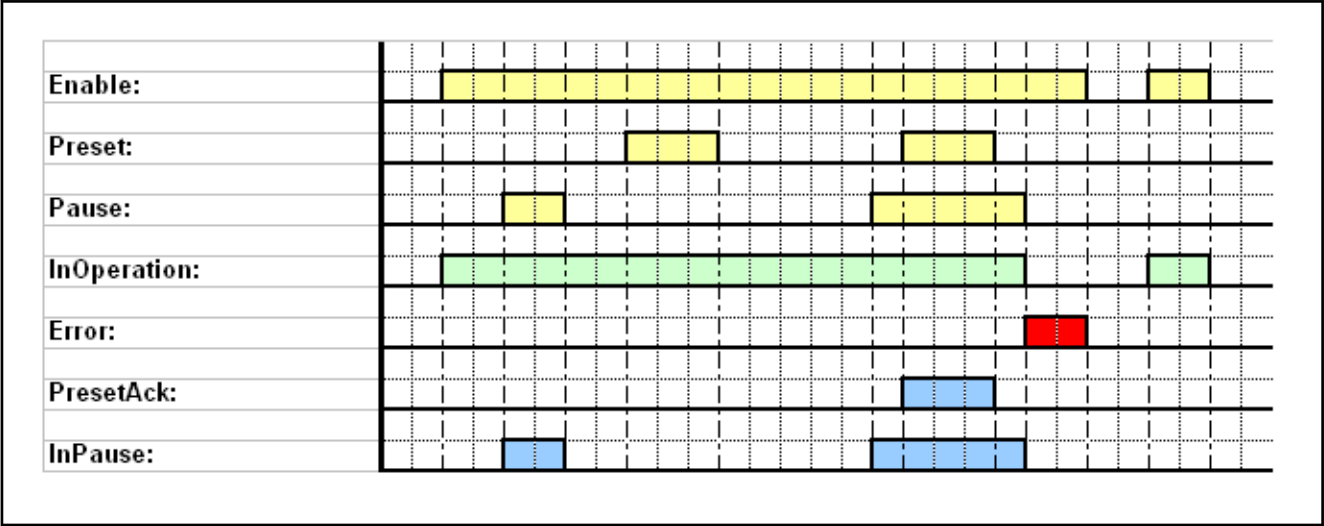


Abb. 7-71: Zeitdiagramm des MB_RegiRegulateType01

Funktionsbeschreibung

Der Funktionsbaustein wird verwendet, um Stellwerte auf einen gewünschten Parameter in einer Achse zu schreiben. Je nach dem gewünschten Zielparame- ter, "ControlledValueIDN", umfasst der Funktionsbaustein eine Integration des Eingangs "ParameterControlValue". Die folgenden Werte von "Control- ledValueIDN" werden integriert:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Ein- gang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "ParameterControlValue"-Ausgang behält seinen Wert.

Preset Der Ausgang "PresetAck" ist aktiv wenn der Wert an "PresetValue" berück- sichtigt wird. Der Eingang "Preset" hat Priorität über den "Pause"-Eingang. Der Wert in "PresetValue" wird bei einer ansteigenden Flanke von "Preset" berücksichtigt wenn der Eingang "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt und der Wert in "PresetVa- lue" an den geregelten Parameter geschrieben.

DisableCyclicWrites Deaktiviert zyklisches Schreiben auf den geregelten Parameter (Eingang "ControlledValueIDN") vom FB.

Normalbetrieb Der RegiRegulateType berechnet nur dann einen neuen Wert wenn sich der "Step"-Eingang ändert. Die geregelte Achse muss eine Modulo-Achse sein.

Notwendige Parametrierung IndraMotion MLC-Parametrierung

Wird eine reale Achse als geregelte Achse ("ControlledAxis"-Eingang) benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695



Zusätzlich muss dieser P-Parameter im MDT des zyklischen Datenkanals im Antrieb konfiguriert sein.

Wird eine virtuelle Achse als geregelte Achse benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten A-Parameter sein:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

IndraMotion MLD-Parametrierung

Der Parameter, der als geregelter Parameter für den "ControlledValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

INPUT_RANGE_ERROR (16#0006)

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("AXIS_REF") wurde geändert während der FB in Betrieb war
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"AXIS_REF" ist außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000A	S-0-0103 oder A-0-0045 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000B	S-0-0091 oder A-0-0032 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000C	P-0-0750 der geregelten Achse außerhalb des gültigen Wertebereichs

ML_TechRegi.library/MX_TechRegi.lib

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Ausgewählte "ControlledValueIDN" wird nicht unterstützt
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00A0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (A-Parameter für Achse mit Interpolation im Antrieb verwendet)
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00B0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (P-Parameter für Achse mit Interpolation in der Steuerung verwendet)
ACCESS_ERROR (16#0003)	16#0815	16#0002	Benötigter Parameter P-0-0061 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0003	Benötigter Parameter P-0-0690 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0004	Benötigter Parameter P-0-0691 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0005	Benötigter Parameter P-0-0692 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0006	Benötigter Parameter P-0-0694 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0007	Benötigter Parameter P-0-0695 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert

Abb. 7-72: Fehlercodes Funktionsbaustein MB_RegiRegulateType01

7.4.6 MB_RegiRegulateType02

Kurzbeschreibung Der Funktionsbaustein MB_RegiRegulateType02 wird verwendet, um einen Stellwert in den Antrieb zu schreiben. Wenn das Korrekturfenster aktiv ist, werden die Abweichungen der Regelung innerhalb eines parametrierten Korrekturfensters korrigiert.

ML_TechRegi.library/MX_TechRegi.lib

Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechRegi.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechRegi.lib

Abb. 7-73: Referenztable MB_RegiRegulateType02

Schnittstellenbeschreibung

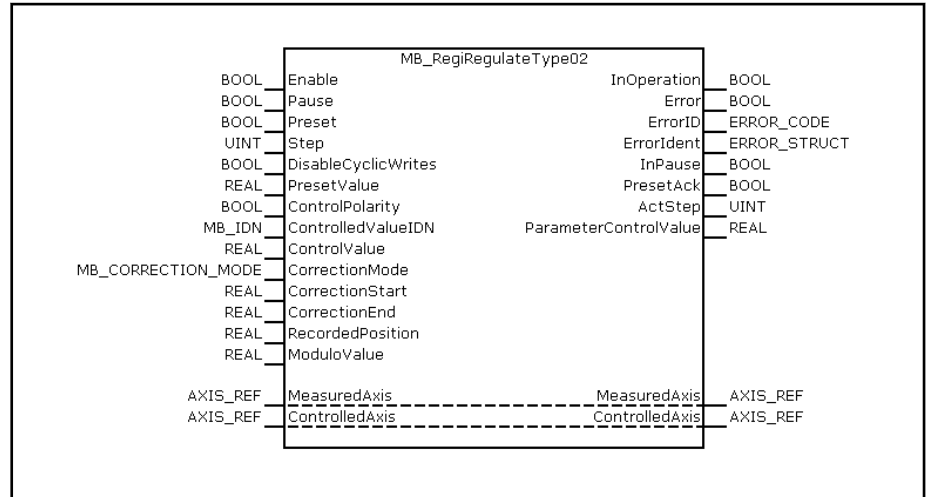


Abb. 7-74: Funktionsbaustein MB_RegiRegulateType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse
	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt "ParameterControlValue" zwangsweise auf "PresetValue"
	Step	UINT	Verarbeitungsschritt
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlledValueIDN" wenn TRUE
	PresetValue	REAL	Wert an den Ausgang "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist

ML_TechRegi.library/MX_TechRegi.lib

E/A-Typ	Name	Datentyp	Beschreibung
	ControlledValueIDN	MB_IDN	SERCOS-IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: IndraMotion MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 IndraMotion MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	ControlValue	REAL	Berechneter Stellwert von z.B. IL_PIType02
	CorrectionMode	MB_CORRECTION_MODE	CORR_ABS = absolutes Korrekturfenster [CorrectionStart .. CorrectionEnd]. CORR_REL = relatives Korrekturfenster [Anfang = "RecordedPosition" + "CorrectionStart" .. Ende = "RecordedPosition" + "CorrectionEnd"] CORR_REL_ABS = relatives Anfangs... absolutes Endkorrekturfenster [Anfang = "RecordedPosition" + "CorrectionStart" .. Ende = "CorrectionEnd"] CORR_ABS_REL = absolutes Anfangskorrekturfenster... relatives Endkorrekturfenster [Anfang = "CorrectionStart" .. Ende = "RecordedPosition" + "CorrectionEnd"] CORR_WIN_DISABLED = Korrekturfenster ist deaktiviert
	CorrectionStart	REAL	Anfang des Korrekturfensters
	CorrectionEnd	REAL	Ende des Korrekturfensters
	RecordedPosition	REAL	Gemessener Wert
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "ParameterControlValue" behält seinen Wert
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in den Ausgang "ParameterControlValue" kopiert

ML_TechRegi.library/MX_TechRegi.lib

E/A-Typ	Name	Datentyp	Beschreibung
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ParameterControlValue" hoch
	ParameterControlValue	REAL	Der Registerregler-FB berechnet einen Wert, der auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang basiert und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parametereingang "ControlledValueIDN" geschrieben wenn der Eingang "DisableCyclicWrites" = FALSE ist

Abb.7-75: Schnittstellenvariablen Funktionsbaustein MB_RegiRegulateType02

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteinneingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
ControlValue	REAL	n.def	n.def	0.0	Kontinuierlich
CorrectionMode	MB_CORRECTION_MODE	0	4	CORR_REL	Kontinuierlich
CorrectionStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich
RecordedPosition	REAL	0.0	ModuloValue	0.0	Änderung des "Step"-Eingangs
ModuloValue	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"

Abb.7-76: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm

Zeitdiagramm gemäß der Spezifikation PLCopen.

ML_TechRegi.library/MX_TechRegi.lib

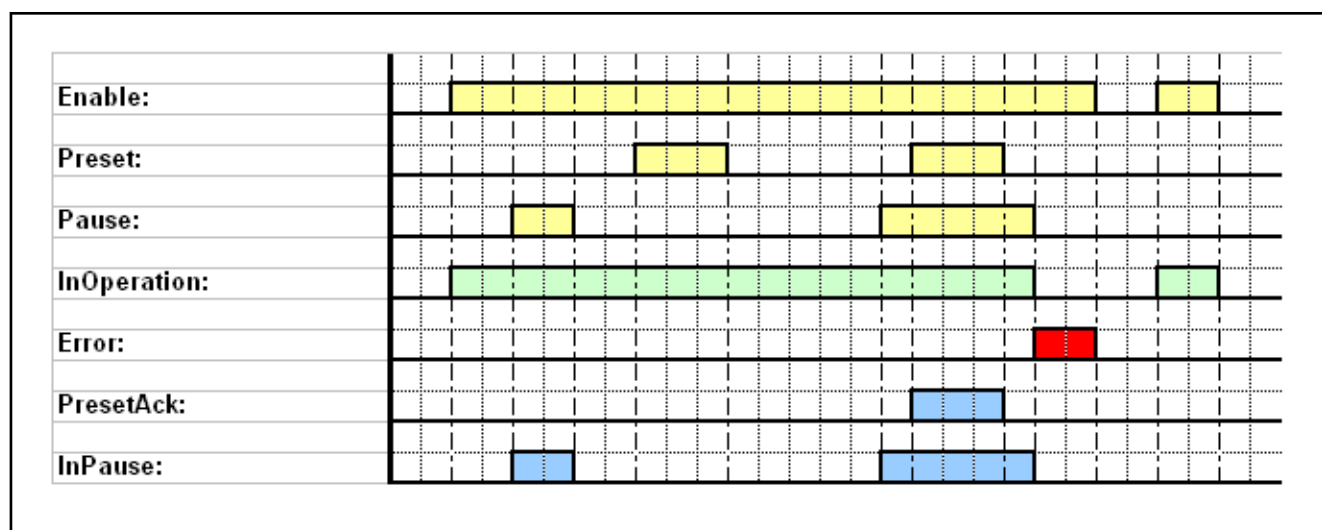


Abb. 7-77: Zeitdiagramm des MB_RegiRegulateType02

Funktionsbeschreibung

Der Funktionsbaustein wird verwendet, um Stellwerte auf einen gewünschten Parameter in einer Achse zu schreiben. Je nach dem gewünschten Zielparame- ter, "ControlledValueIDN", umfasst der Funktionsbaustein eine Integration des Eingangs "ParameterControlValue". Die folgenden Werte von "Control- ledValueIDN" werden integriert:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Pause

Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Ein- gang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "ParameterControlValue"-Ausgang behält seinen Wert.

Preset

Der Ausgang "PresetAck" ist aktiv wenn der Wert an "PresetValue" berück- sichtigt wird. Der Eingang "Preset" hat Priorität über den "Pause"-Eingang. Der Wert in "PresetValue" wird bei einer ansteigenden Flanke von "Preset" berücksichtigt wenn der Eingang "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt und der Wert in "PresetVa- lue" an den geregelten Parameter geschrieben.

DisableCyclicWrites

Deaktiviert zyklisches Schreiben auf den geregelten Parameter (Eingang "ControlledValueIDN") vom FB.

Korrekturfenster

Das Korrekturfenster ist ein auf die Messachse bezogener Winkelbereich. Der Registerregler regelt die Regelungsabweichungen innerhalb dieses Fensters. Die Lage und Größe des Korrekturfensters werden mithilfe der fol- genden Eingänge gesetzt:

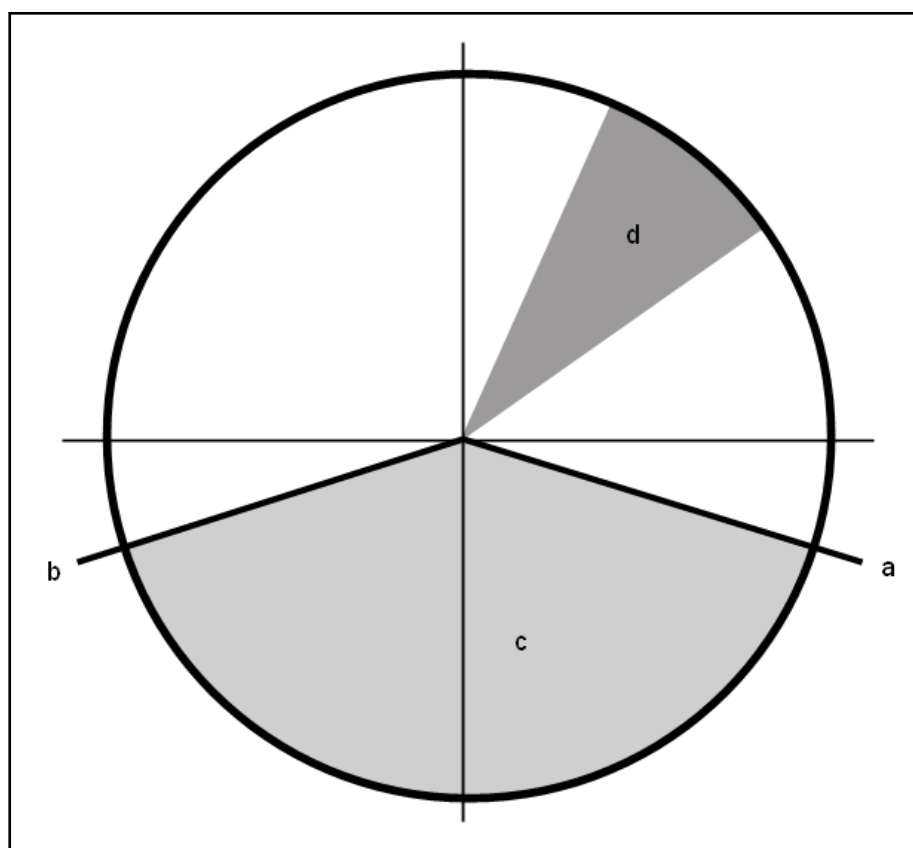
Der Eingang "CorrectionMode" stellt das Korrekturfenster entweder auf abso- lut oder relativ ein. Wenn "CorrectionMode" auf CORR_WIN_DISABLED ge- setzt wird, ist das Korrekturfenster inaktiv.

Der Eingang "CorrectionStart" legt den Anfang des Korrekturfensters fest.

Der Eingang "CorrectionEnd" legt das Ende des Korrekturfensters fest.

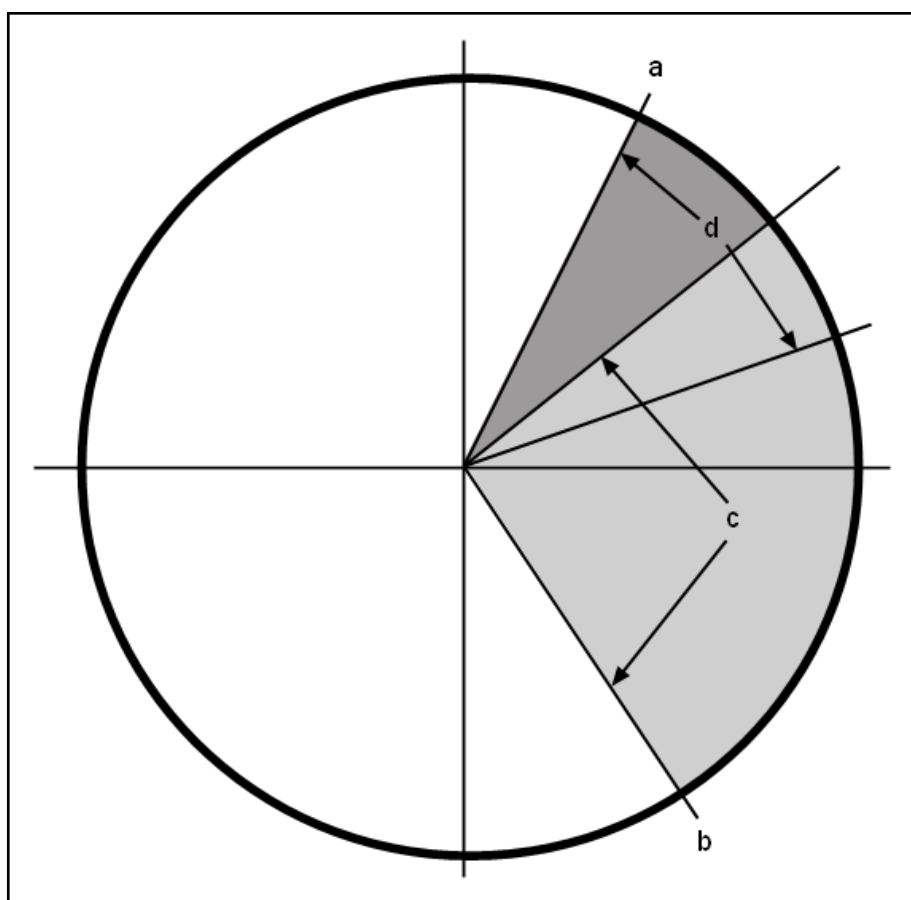
Das Korrekturfenster unterstützt nur einen positiven Modulo-Bereich (z.B. 0...+360 Grad) der Messachse.

ML_TechRegi.library/MX_TechRegi.lib



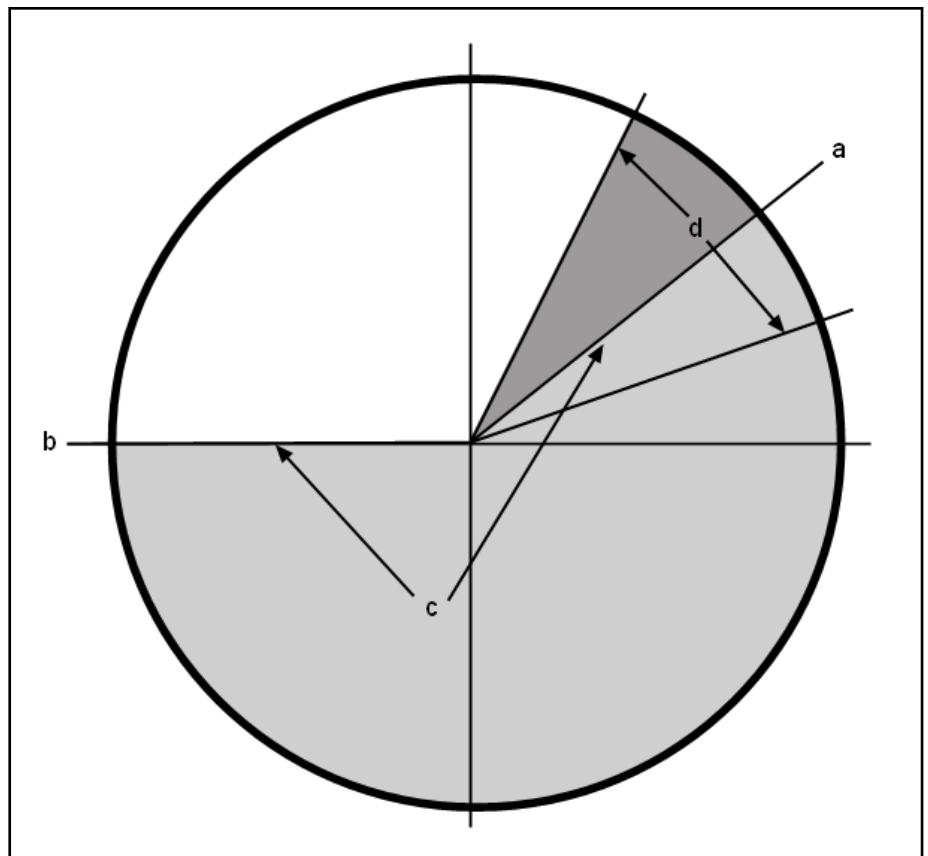
a: Beginn Korrekturfenster
b: Ende Korrekturfenster
c: Korrekturfenster
d: Erwartungsfenster
Abb. 7-78: Korrekturfenster absolut

ML_TechRegi.library/MX_TechRegi.lib



- a: Anfang Korrekturfenster = Messende
 b: Relatives Ende des Korrekturfensters bezogen auf das Messende
 c: Korrekturfenster
 d: Erwartungsfenster

Abb. 7-79: Korrekturfenster relativ



- a: Anfang Korrekturfenster = Messende
b: Ende des Korrekturfensters = perm. Winkel
c: Korrekturfenster
d: Erwartungsfenster

Abb. 7-80: Korrekturfenster, relativer Start und absolutes Ende

Normalbetrieb

Der Funktionsbaustein RegiRegulateType berechnet nur einen neuen Wert wenn sich der Eingang "Step" ändert. Die geregelte und gemessene Achse müssen Modulo-Achsen sein.

Notwendige Parametrierung

IndraMotion MLC-Parametrierung

Wird eine reale Achse als geregelte Achse ("ControlledAxis"-Eingang) benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695



Zusätzlich muss dieser P-Parameter im MDT des zyklischen Datenkanals im Antrieb konfiguriert sein.

Wird eine virtuelle Achse als geregelte Achse benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten A-Parameter sein:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

IndraMotion MLD-Parametrierung

Der Parameter, der als geregelter Parameter für den "ControlledValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt:

ML_TechRegi.library/MX_TechRegi.lib

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

Datentyp**MB_CORRECTION_MODE**

Modus des Korrekturfensters.

Programm:

```

TYPE MB_CORRECTION_MODE:
  (CORR_WIN_DISABLED :=0, (* Korrekturfenster ist ausgeschaltet *)
   CORR_ABS          :=1, (* Korrekturfenster ist absolut *)
   CORR_REL          :=2, (* Korrekturfenster ist relativ *)
   CORR_REL_ABS      :=3, (* Korrekturfenster ist relativer Anfang
                           und absolutes Ende *)
   CORR_ABS_REL      :=4); (* Korrekturfenster ist absoluter Anfang
                           und relatives Ende *)

```

END_TYPE

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

INPUT_RANGE_ERROR (16#0006)

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("AXIS_REF") wurde geändert während der FB in Betrieb war
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"AXIS_REF" ist außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0002	Eingang "CorrectionStart" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0003	Eingang "CorrectionEnd" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0004	Wert < 0 Eingang "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0005	Wert < 0 Eingang "RecordedPosition"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0006	Eingang "CorrectionStart" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0007	Eingang "CorrectionEnd" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0008	Eingang "RecordedPosition" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0009	Korrekturbahn > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000A	S-0-0103 oder A-0-0045 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000B	S-0-0091 oder A-0-0032 der geregelten Achse außerhalb des gültigen Wertebereichs

ML_TechRegi.library/MX_TechRegi.lib

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000C	P-0-0750 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Ausgewählte "ControlledValueIDN" wird nicht unterstützt
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0002	Eingang "CorrectionMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0003	"CorrectionStart" = "CorrectionEnd"
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00A0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (A-Parameter für Achse mit Interpolation im Antrieb verwendet)
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00B0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (P-Parameter für Achse mit Interpolation in der Steuerung verwendet)
ACCESS_ERROR (16#0003)	16#0815	16#0002	Benötigter Parameter P-0-0061 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0003	Benötigter Parameter P-0-0690 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0004	Benötigter Parameter P-0-0691 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0005	Benötigter Parameter P-0-0692 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0006	Benötigter Parameter P-0-0694 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0007	Benötigter Parameter P-0-0695 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0009	Benötigter Parameter P-0-0753 ist nicht in den optionalen zyklischen AT-Daten der gemessenen Achse konfiguriert

Abb. 7-81: Fehlercodes Funktionsbaustein MB_RegiRegulateType02

ML_TechRegi.library/MX_TechRegi.lib

7.4.7 MB_RegiRegulateType03

Kurzbeschreibung Der Funktionsbaustein MB_RegiRegulateType03 wird verwendet, um einen Stellwert mit Hilfe einer parametrierten Korrekturrampe in den Antrieb zu schreiben.

Zielsystem/Bibliothek	Zielsystem	Bibliothek
	IndraMotion MLC 12VRS	ML_TechRegi.compiled-library
	IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechRegi.lib

Abb. 7-82: Referenztabelle MB_RegiRegulateType03

Schnittstellenbeschreibung

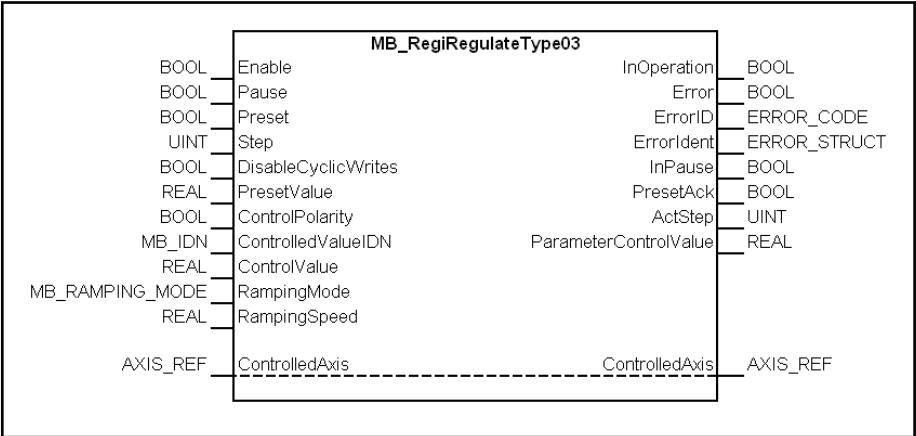


Abb. 7-83: Funktionsbaustein MB_RegiRegulateType03

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt "ParameterControlValue" zwangsweise auf "PresetValue"
	Step	UINT	Verarbeitungsschritt
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlledValueIDN" wenn TRUE
	PresetValue	REAL	Wert an den Ausgang "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist

E/A-Typ	Name	Datentyp	Beschreibung
	ControlledValueIDN	MB_IDN	SERCOS-IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: IndraMotion MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 IndraMotion MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	ControlValue	REAL	Berechneter Stellwert von z.B. IL_PIType02
	RampingMode	MB_RAMPING_MODE	Die folgenden Verrampungsmodi werden unterstützt: 0 = RAMPING_CONSTANT Verrampungsgeschwindigkeit ist konstant 1 = RAMPING_ADAPTIVE Die Verrampungsgeschwindigkeit wird basierend auf der Zeitdifferenz im "Step"-Eingang berechnet
	RampingSpeed	REAL	Dieser Wert verwendet einen Verrampungsgradienten
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "ParameterControlValue" behält seinen Wert
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in den Ausgang "ParameterControlValue" kopiert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ParameterControlValue" hoch
	ParameterControlValue	REAL	Der Registerregler-FB berechnet einen Wert, der auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang basiert und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parametereingang "ControlledValueIDN" geschrieben wenn der Eingang "DisableCyclicWrites" = FALSE ist

Abb. 7-84: Schnittstellenvariablen Funktionsbaustein MB_RegiRegulateType03

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich

ML_TechRegi.library/MX_TechRegi.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
ControlValue	REAL	n.def	n.def	0.0	Kontinuierlich
RampingMode	MB_RAM-PING_MODE	0.0	1	RMPING_CONS	Steigende Flanke an "Enable"
RampingSpeed	REAL	0.0	n.def	1.0	Kontinuierlich

Abb. 7-85: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Zeitdiagramm gemäß der Spezifikation PLCopen.

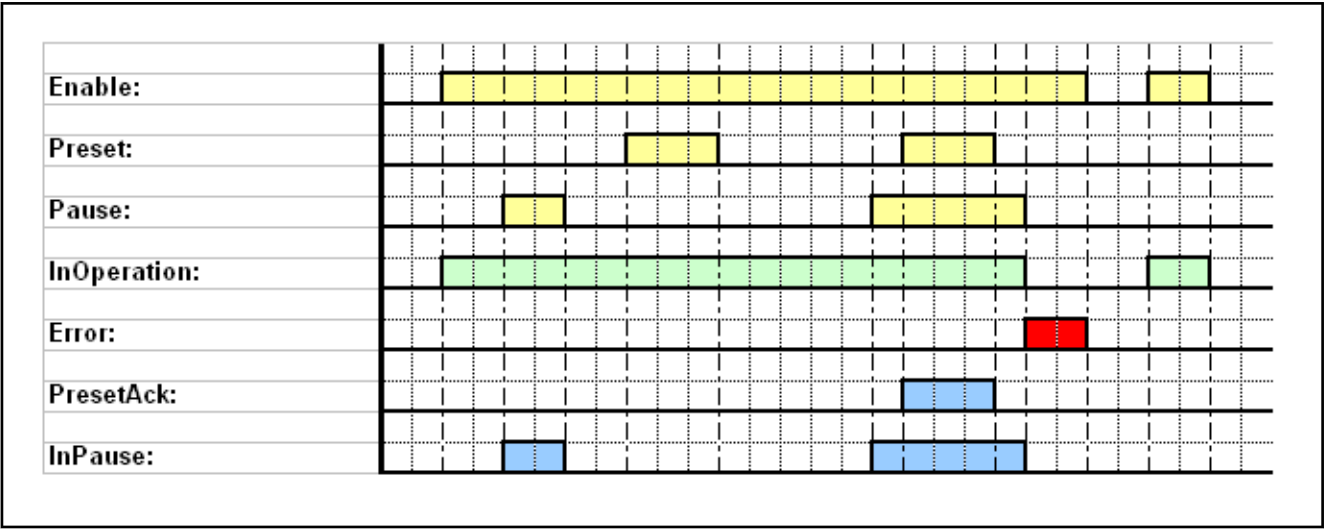


Abb. 7-86: Zeitdiagramm des MB_RegiRegulateType03

Funktionsbeschreibung

Der Funktionsbaustein wird verwendet, um Stellwerte auf einen gewünschten Parameter in einer Achse zu schreiben. Je nach dem gewünschten Parameter, "ControlledValueIDN", umfasst der Funktionsbaustein die Integration des Eingangs "ParameterControlValue". Die folgenden Werte von "ControlledValueIDN" werden integriert:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Pause

Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "ParameterControlValue"-Ausgang behält seinen Wert.

- Preset** Der Ausgang "PresetAck" ist aktiv wenn der Wert an "PresetValue" berücksichtigt wird. Der Eingang "Preset" hat Priorität über den "Pause"-Eingang. Der Wert in "PresetValue" wird bei einer ansteigenden Flanke von "Preset" berücksichtigt wenn der Eingang "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt und der Wert in "PresetValue" an den geregelten Parameter geschrieben.
- DisableCyclicWrites** Deaktiviert zyklisches Schreiben auf den geregelten Parameter (Eingang "ControlledValueIDN") vom FB.
- Verrampungsmodus** Es gibt 2 verschiedene Verrampungsmodi. Ein Modus ist das konstante Verrampen mit dem Gradienten des "RampingSpeed" ("RAMPING_CONSTANT")-Eingangs. Der andere Modus hängt von der Änderung des "Step"-Eingangs ab. Je schneller sich der "Step"-Eingang ändert, desto steiler ist das Verrampen des "ParameterControlValue" ("RAMPING_ADAPTIVE").
- **RAMPING_CONSTANT:**
In diesem Modus hängt die Änderung am "ParameterControlValue"-Ausgang vom "RampingSpeed"-Eingang ab. Ist der "RampingSpeed"-Eingang beispielsweise auf 1 gesetzt, dann ist das Verrampen des "ParameterControlValue"-Ausgangs 1 Grad / Sekunde für den Parameter P-0-0691.

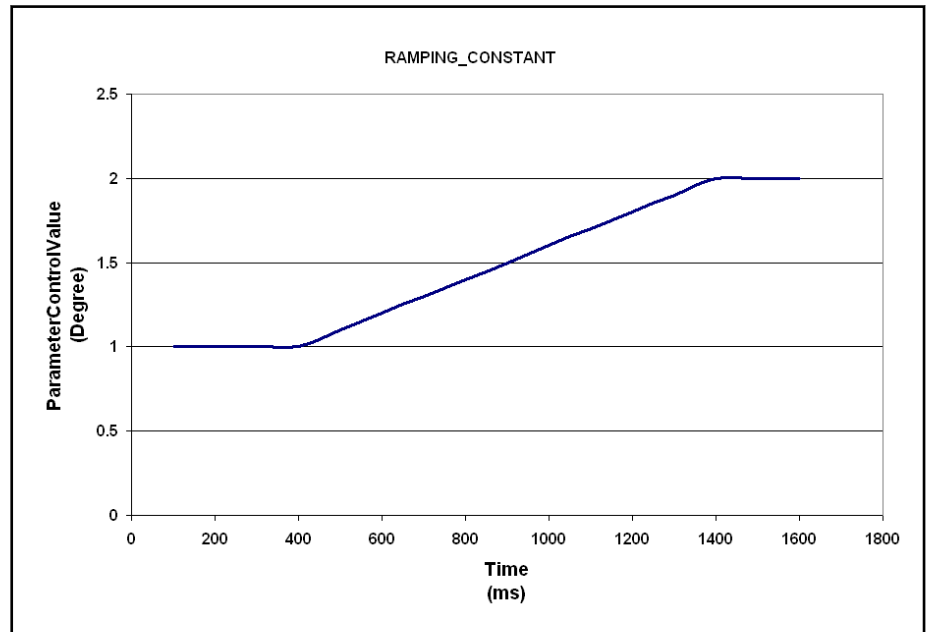


Abb. 7-87: Verrampungskonstante

- **RAMPING_ADAPTIVE:**
In diesem Modus hängt die Änderung des "ParameterControlValue" von der Zeitdifferenz der Änderungen ein "Step"-Eingangs ab. Beispielsweise verursacht eine Änderung nach jeden 0.6 (0.3) Sekunden am "Step"-Eingang, eine Verrampung des "ParameterControlValue"-Ausgang von 1(2) Grad / Sekunden für den Parameter P-0-0691.

ML_TechRegi.library/MX_TechRegi.lib

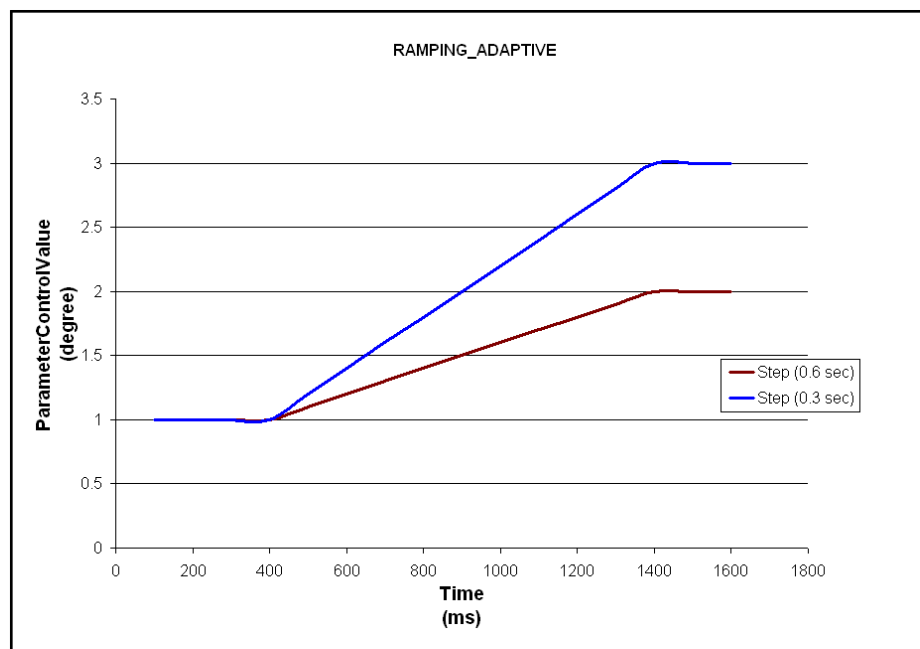


Abb.7-88: Adaptive Verrampung

Verrampungsgeschwindigkeit

Die Verrampungsgeschwindigkeit ist nur in Verbindung mit dem Verrampungsmodus RAMPING_CONSTANT gültig.

Normalbetrieb

Der RegiRegulateType berechnet nur einen neuen Wert wenn sich der "Step"-Eingang ändert. Die geregelte Achse muss eine Modulo-Achse sein.

Notwendige Parametrierung**IndraMotion MLC-Parametrierung**

Wird eine reale Achse als geregelte Achse ("ControlledAxis"-Eingang) benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695



Zusätzlich muss dieser P-Parameter im MDT des zyklischen Datenkanals im Antrieb konfiguriert sein.

Wird eine virtuelle Achse als geregelte Achse benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten A-Parameter sein:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

IndraMotion MLD-Parametrierung

Der Parameter, der als geregelter Parameter für den "ControlledValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

INPUT_RANGE_ERROR (16#0006)

ML_TechRegi.library/MX_TechRegi.lib

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("AXIS_REF") wurde geändert während der FB in Betrieb war
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000A	S-0-0103 oder A-0-0045 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000B	S-0-0091 oder A-0-0032 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000C	P-0-0750 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Ausgewählte "ControlledValueIDN" wird nicht unterstützt
INPUT_INVALID_ERROR (16#0001)	16#0812	16#000C	Wert < 0 für den "RampingSpeed"-Eingang
INPUT_INVALID_ERROR (16#0001)	16#0812	16#000D	Modulowert der geregelten Achse konnte nicht gelesen werden oder ist 0
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00A0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (A-Parameter für Achse mit Interpolation im Antrieb verwendet)
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00B0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (P-Parameter für Achse mit Interpolation in der Steuerung verwendet)
ACCESS_ERROR (16#0004)	16#0815	16#0002	Benötigter Parameter P-0-0061 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0004)	16#0815	16#0003	Benötigter Parameter P-0-0690 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0004)	16#0815	16#0004	Benötigter Parameter P-0-0691 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert

ML_TechRegi.library/MX_TechRegi.lib

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#0815	16#0005	Benötigter Parameter P-0-0692 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0004)	16#0815	16#0006	Benötigter Parameter P-0-0694 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0004)	16#0815	16#0007	Benötigter Parameter P-0-0695 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert

Abb. 7-89: Fehlercodes Funktionsbaustein MB_RegiRegulateType03

7.4.8 MB_RegiRegulateType04

Kurzbeschreibung

Der Funktionsbaustein MB_RegiRegulateType04 wird verwendet, um einen Regelwert an bis zu acht Antriebe zu schreiben. Wenn das Korrekturfenster aktiv ist, werden die Abweichungen der Regelung innerhalb eines parametrisierten Korrekturfensters korrigiert.

Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechRegi.compiled-library

Abb. 7-90: Referenztabelle MB_RegiRegulateType04

Schnittstellenbeschreibung

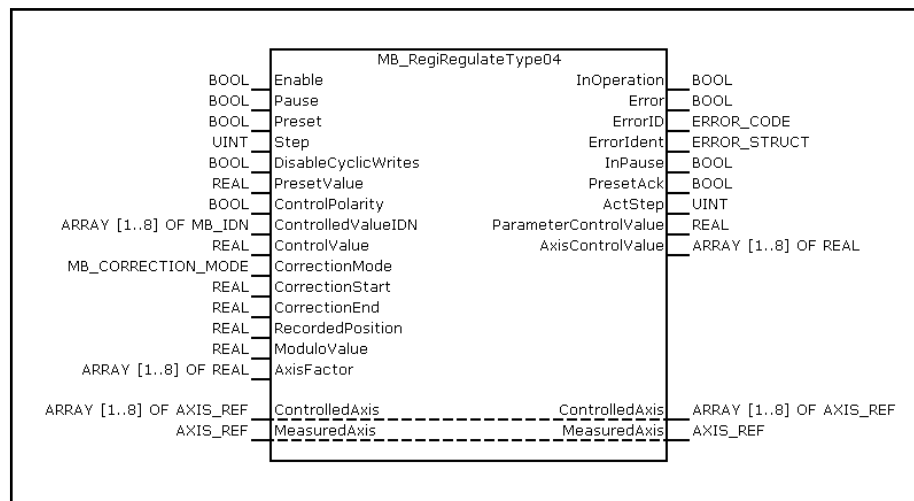


Abb. 7-91: Funktionsbaustein MB_RegiRegulateType04

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse
	ControlledAxis	ARRAY [1..8] OF AXIS_REF	Bezug auf die geregelten Achsen
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist

E/A-Typ	Name	Datentyp	Beschreibung
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt "ParameterControlValue" zwangsweise auf "PresetValue"
	Step	UINT	Verarbeitungsschritt
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlledValueIDN" wenn TRUE
	PresetValue	REAL	Wert an den Ausgang "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist
	ControlledValueIDN	ARRAY [1..8] OF MB_IDN	SERCOS-IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: IndraMotion MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 IndraMotion MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	ControlValue	REAL	Berechneter Stellwert von z.B. IL_PIType02
	CorrectionMode	MB_CORRECTION_MODE	CORR_ABS = absolutes Korrekturfenster [CorrectionStart .. CorrectionEnd]. CORR_REL = relatives Korrekturfenster [Anfang = "RecordedPosition" + "CorrectionStart" .. Ende = "RecordedPosition" + "CorrectionEnd"] CORR_REL_ABS = relatives Anfangs... absolutes Endkorrekturfenster [Anfang = "RecordedPosition" + "CorrectionStart" .. Ende = "CorrectionEnd"] CORR_ABS_REL = absolutes Anfangskorrekturfenster... relatives Endkorrekturfenster [Anfang = "CorrectionStart" .. Ende = "RecordedPosition" + "CorrectionEnd"] CORR_WIN_DISABLED = Korrekturfenster ist deaktiviert
	CorrectionStart	REAL	Anfang des Korrekturfensters
	CorrectionEnd	REAL	Ende des Korrekturfensters
	RecordedPosition	REAL	Gemessener Wert
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
	AxisFactor	ARRAY [1..8] OF REAL	Faktor zum Archivieren einer achsabhängigen Wichtung. Der Wert von "ParameterControlValue" wird mit diesem achsabhängigen Faktor (ARRAY) multipliziert und an jede geregelte Achse verteilt
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein ist in Betrieb

ML_TechRegi.library/MX_TechRegi.lib

E/A-Typ	Name	Datentyp	Beschreibung
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	InPause	BOOL	"Pause" ist aktiv. Die Ausgänge "ParameterControlValue" und "AxisControlValue" behalten ihre Werte.
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in den Ausgang "ParameterControlValue" kopiert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ParameterControlValue" hoch
	ParameterControlValue	REAL	Der Registerregler-FB berechnet einen Wert, der auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang basiert und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parametereingang "ControlledValueIDN" geschrieben wenn der Eingang "DisableCyclicWrites" = FALSE ist
	AxisControlValue	ARRAY [1..8] OF REAL	Berechneter Wert des Registerregler-FB wird in den "ControlledValueIDN"-Eingang geschrieben. "AxisControlValue" ist der gewichtete "ParameterControlValue". Zur Diagnose oder weiteren Verarbeitung verwendet, wenn "DisableCyclicWrites" = TRUE

Abb. 7-92: Schnittstellenvariablen Funktionsbaustein MB_RegiRegulateType04

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	ARRAY [1..8] OF MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
ControlValue	REAL	n.def	n.def	0.0	Kontinuierlich
CorrectionMode	MB_CORRECTION_MODE	0	4	CORR_REL	Kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
CorrectionStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich
RecordedPosition	REAL	0.0	ModuloValue	0.0	Änderung des "Step"-Eingangs
ModuloValue	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"
AxisFactor	ARRAY [1..8] OF REAL	n.def	n.def	1.0	Kontinuierlich

Abb. 7-93: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Zeitdiagramm gemäß der Spezifikation PLCopen.

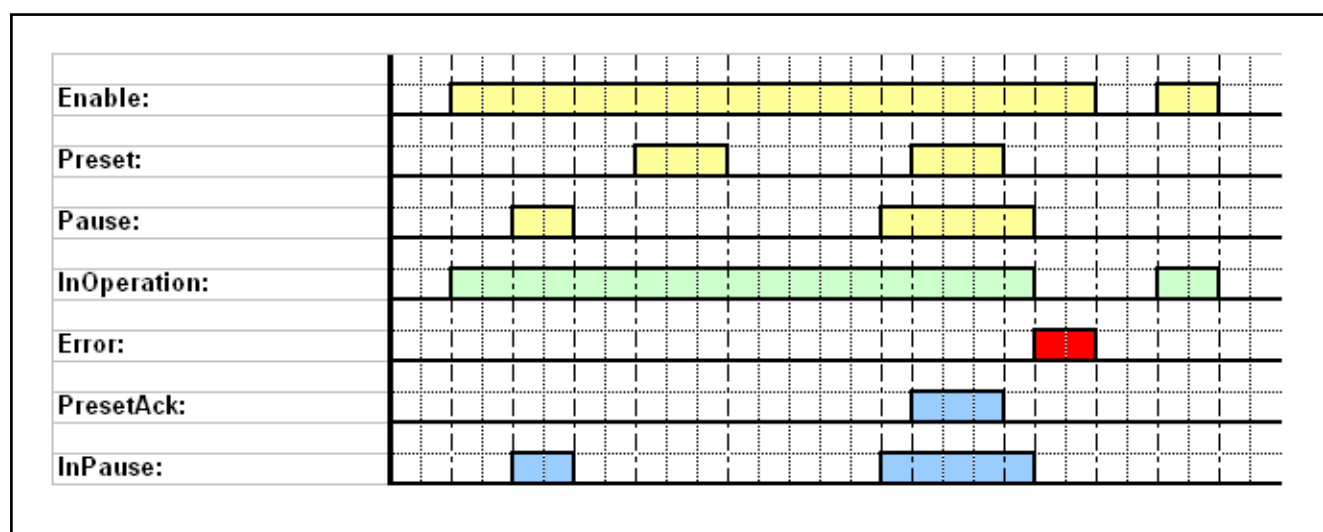


Abb. 7-94: Zeitdiagramm des MB_RegiRegulateType04

Funktionsbeschreibung

Der Funktionsbaustein wird verwendet, um Stellwerte auf einen gewünschten Parameter in einer Achse zu schreiben. Je nach dem gewünschten Parameter, "ControlledValueIDN", umfasst der Funktionsbaustein die Integration des Eingangs "ParameterControlValue". Für die folgenden Werte von ControlledValueIDN wird diese Integration vollzogen:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und die Ausgänge "ParameterControlValue" und "AxisControlValue" behalten ihre Werte.

Preset Der Ausgang "PresetAck" ist aktiv wenn der Wert an "PresetValue" berücksichtigt wird. Der Eingang "Preset" hat Priorität über den "Pause"-Eingang. Der Wert in "PresetValue" wird bei einer ansteigenden Flanke von "Preset" berücksichtigt wenn der Eingang "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt und der Wert in "PresetValue" an den geregelten Parameter geschrieben.

ML_TechRegi.library/MX_TechRegi.lib

DisableCyclicWrites Deaktiviert zyklisches Schreiben auf den geregelten Parameter (Eingang "ControlledValueIDN") vom FB.

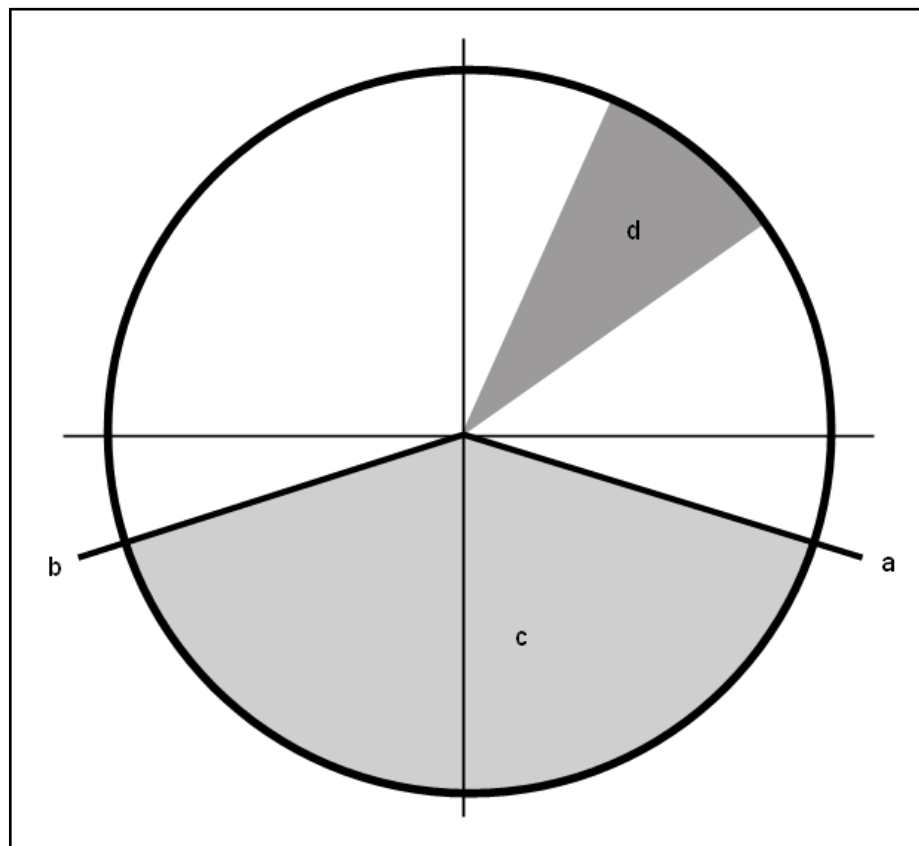
Korrekturfenster Das Korrekturfenster ist ein auf die Messachse bezogener Winkelbereich. Der Registerregler regelt die Regelungsabweichungen innerhalb dieses Fensters. Die Lage und Größe des Korrekturfensters werden mithilfe der folgenden Eingänge gesetzt:

Der Eingang "CorrectionMode" stellt das Korrekturfenster entweder auf absolut oder relativ ein. Wenn "CorrectionMode" auf CORR_WIN_DISABLED gesetzt wird, ist das Korrekturfenster inaktiv.

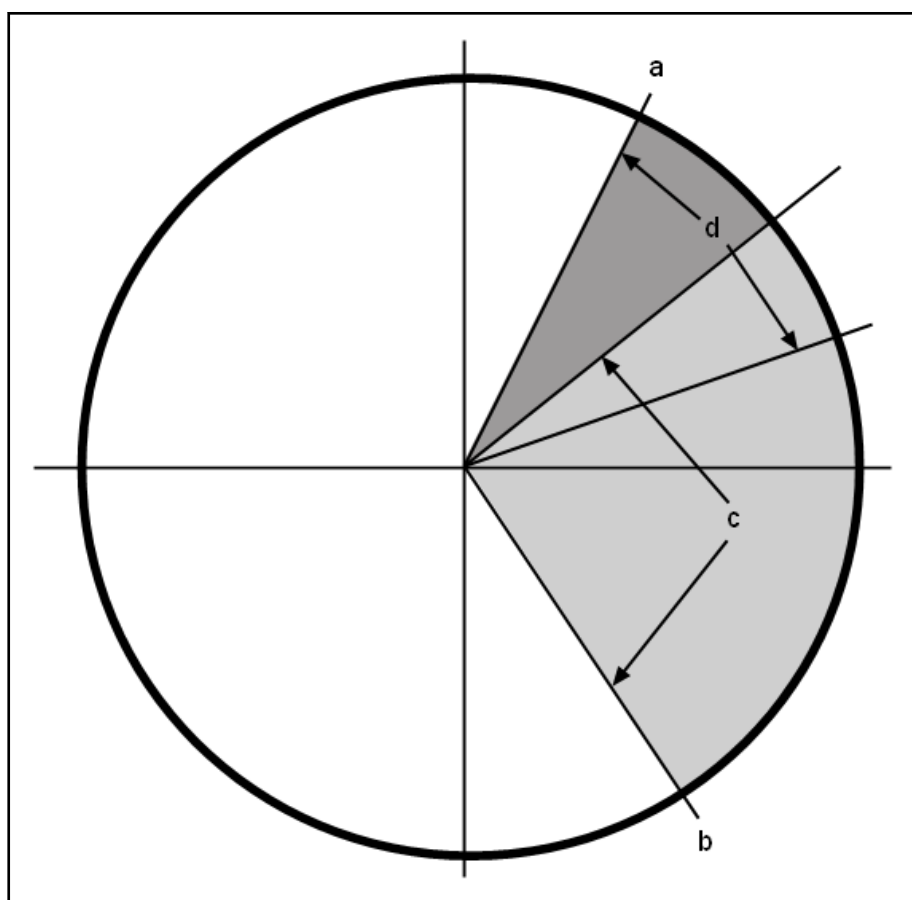
Der Eingang "CorrectionStart" legt den Anfang des Korrekturfensters fest.

Der Eingang "CorrectionEnd" legt das Ende des Korrekturfensters fest.

Das Korrekturfenster unterstützt nur einen positiven Modulo-Bereich (z.B. 0...+360 Grad) der Messachse.

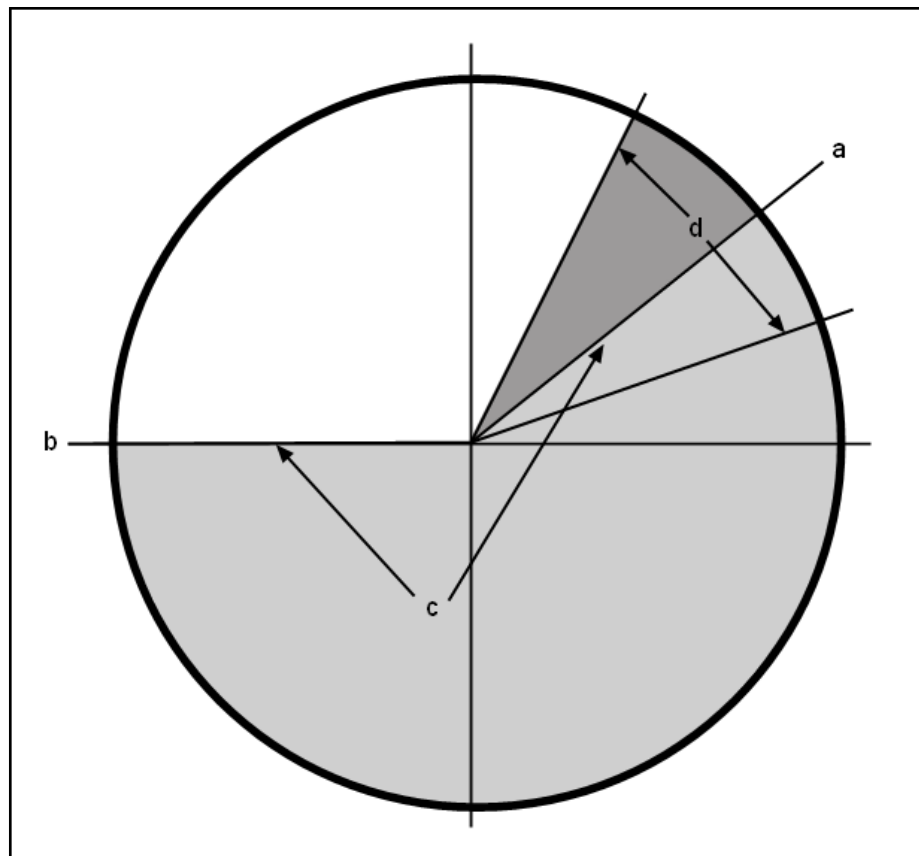


a: Beginn Korrekturfenster
 b: Ende Korrekturfenster
 c: Korrekturfenster
 d: Erwartungsfenster
 Abb. 7-95: Korrekturfenster absolut



- a: Anfang Korrekturfenster = Messende
 b: Relatives Ende des Korrekturfensters bezogen auf das Messende
 c: Korrekturfenster
 d: Erwartungsfenster
 Abb. 7-96: Korrekturfenster relativ

ML_TechRegi.library/MX_TechRegi.lib



a: Anfang Korrekturfenster = Messende

b: Ende des Korrekturfensters = perm. Winkel

c: Korrekturfenster

d: Erwartungsfenster

Abb. 7-97: Korrekturfenster, relativer Start und absolutes Ende

Mehrere Achsen

Die berechneten Stellwerte werden in die entsprechenden Parameter ("ControlValueIDN") auf bis zu 8 geregelte Achsen geschrieben. Die Achsen sind mit dem Eingang "ControlledAxis[x]" referenziert.

Individuelle Stellwertwichtung

Der Stellwert jeder einzelnen Achse kann mit dem Eingang "AxisFactor[x]" gewichtet werden.

Normalbetrieb

Der Funktionsbaustein RegiRegulateType berechnet nur einen neuen Wert wenn sich der Eingang "Step" ändert. Die geregelte und gemessene Achse müssen Modulo-Achsen sein.

Notwendige Parametrierung**IndraMotion MLC-Parametrierung**

Wird eine reale Achse als geregelte Achse ("ControlledAxis"-Eingang) benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695



Zusätzlich muss dieser P-Parameter im MDT des zyklischen Datenkanals im Antrieb konfiguriert sein.

Wird eine virtuelle Achse als geregelte Achse benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten A-Parameter sein:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

IndraMotion MLD-Parametrierung

Der Parameter, der als geregelter Parameter für den "ControlledValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:
INPUT_RANGE_ERROR (16#0006)

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (AXIS_REF) wurde geändert, während der FB in Betrieb ist
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AXIS_REF ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0002	Eingang "CorrectionStart" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0003	Eingang "CorrectionEnd" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0004	Wert < 0 Eingang "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0005	Wert < 0 Eingang "RecordedPosition"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0006	Eingang "CorrectionStart" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0007	Eingang "CorrectionEnd" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0008	Eingang "RecordedPosition" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0009	Korrekturbahn > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0x10	S-0-0103 oder A-0-0045 der Controlled Axis[x] außerhalb des Gültigkeitsbereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0x20	S-0-0091 oder A-0-0032 der Controlled Axis[x] außerhalb des Gültigkeitsbereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0x30	P-0-0750 der Controlled Axis[x] außerhalb des Gültigkeitsbereichs
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0x01	Gewählte "ControlledValueIDN[x]" wird nicht unterstützt
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0002	Eingang "CorrectionMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0003	"CorrectionStart" = "CorrectionEnd"

ML_TechRegi.library/MX_TechRegi.lib

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0xA0	Der Typ des zu regelnden Parameters "ControlledValueIDN[x]" passt nicht zum Typ der geregelten Achse "Controlled Axis[x]" (A-Parameter für Achse mit Interpolation im Antrieb verwendet)
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0xB0	Der Typ des zu regelnden Parameters "ControlledValueIDN[x]" passt nicht zum Typ der geregelten Achse "Controlled Axis[x]" (P-Parameter für Achse mit Interpolation in der Steuerung verwendet)
ACCESS_ERROR (16#0003)	16#0815	16#0x02	Benötigter Parameter P-0-0061 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0x03	Benötigter Parameter P-0-0690 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0x04	Benötigter Parameter P-0-0691 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0x05	Benötigter Parameter P-0-0692 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0x06	Benötigter Parameter P-0-0694 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0x07	Benötigter Parameter P-0-0695 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0009	Benötigter Parameter P-0-0753 ist nicht in den optionalen zyklischen AT-Daten der gemessenen Achse konfiguriert

Abb. 7-98: Fehlercodes Funktionsbaustein MB_RegiRegulateType04

7.5 Registersollwertübernahme-FB

7.5.1 Überblick

Dieser Abschnitt beschreibt den Registersollwertübernahme-FB, der Teil der Bibliotheken ML_TechRegi (IndraMotion MLC) und MX_TechRegi (IndraMotion MLD) ist.

7.5.2 MB_SetpointLockType01

Kurzbeschreibung Der Funktionsbaustein MB_RegiSetpointLockType01 dient zum Verriegeln eines Werts bei einem Registerregler.

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechRegi.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechRegi.lib

Abb. 7-99: Referenztabelle MB_SetpointLockType01

Schnittstellenbeschreibung

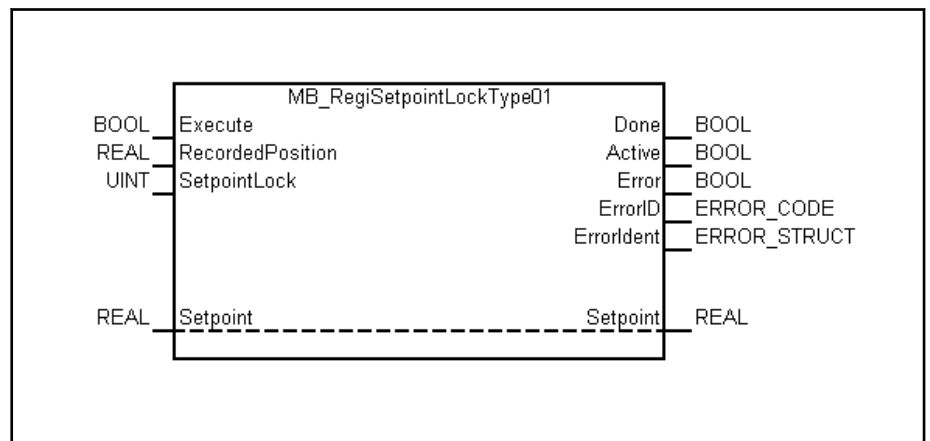


Abb. 7-100: Funktionsbaustein MB_SetpointLockType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Setpoint	REAL	Setpoint
VAR_IN	Execute	BOOL	Der Registersollwertübernahme-FB wird bei einer ansteigenden Flanke aktiv
	RecordedPosition	REAL	Die aufgenommene Position wird in den Eingang "Setpoint" kopiert wenn sich "SetpointLock" ändert
	SetpointLock	UINT	Der Wert muss sich erhöhen wenn der Wert in "RecordedPosition" übernommen wird. Bei 65,535 schlägt der Wert nach 0 um (Wert vom Modulo-Typ). Normalerweise wird ein "actStep"-Ausgang eines Technologiefunktionsbausteins als Eingang verwendet
VAR_OUT	Done	BOOL	Der Funktionsbaustein ist abgeschlossen und der Wert im "RecordPosition"-Eingang wird zu "Setpoint" kopiert
	Active	BOOL	Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass es zu einem Fehler während des Funktionsbausteinbetriebes kam
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

Abb. 7-101: Schnittstellenvariablen Funktionsbaustein MB_RegiSetpointLockType01

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteingänge aufgelistet.

ML_TechRegi.library/MX_TechRegi.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
RecordedPosition	REAL			0.0	Änderung von "SetpointLock"
SetpointLock	UINT	0	65.535	0	Kontinuierlich. Initialisiert bei ansteigender Flanke von "Execute"

Abb. 7-102: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Das folgende Diagramm zeigt den Signal-Zeit-Verlauf des Funktionsbausteins MB_RegiSetpointLockType01.

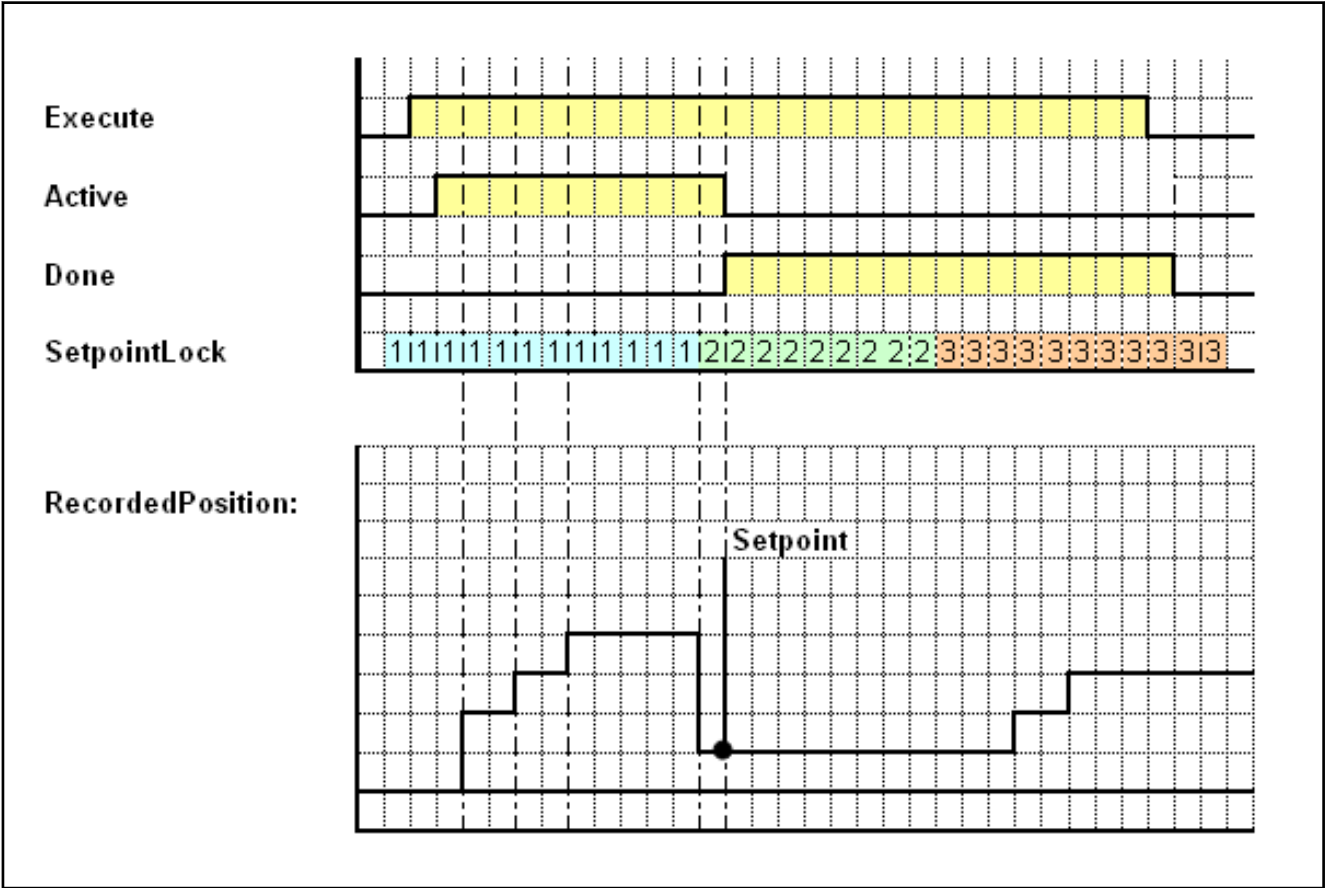


Abb. 7-103: Zeitdiagramm des MB_RegiSetpointLockType01

- Funktionsbeschreibung

Der Funktionsbaustein setzt "Setpoint" auf den als nächsten auftretenden Wert am "RecordedPosition"-Eingang. Das nächste Auftreten wird erkannt wenn sich der Eingang "SetpointLock" ändert. Man kann dies verwenden, um einen neuen Sollwert z.B. für einen Registerregler zu setzen.
- Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins

Abb. 7-104: Fehlercodes Funktionsbaustein MB_RegiSetpointLockType01

7.6 Registerqualitäts-FB

7.6.1 Überblick

Dieser Abschnitt beschreibt den Registerqualitäts-FB, der Teil der Bibliotheken ML_TechRegi (IndraMotion MLC) und MX_TechRegi (IndraMotion MLD) ist.

7.6.2 IL_RegiQualityType01

Kurzbeschreibung

Dieser Funktionsbaustein, der eventgesteuert aufgerufen wird, wertet bis zu 16 Registerabweichungen aus. Anhand von zwei Grenzwerten werden die Druckbilder in "gute/mittlere/schlechte" Qualität eingeteilt und aufsummiert. Entscheidend für das Druckbild ist hierbei der schlechteste Wert einer Farbmarke.

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechRegi.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechRegi.lib

Abb.7-105: Referenztabelle IL_RegiQualityType01

Schnittstellenbeschreibung

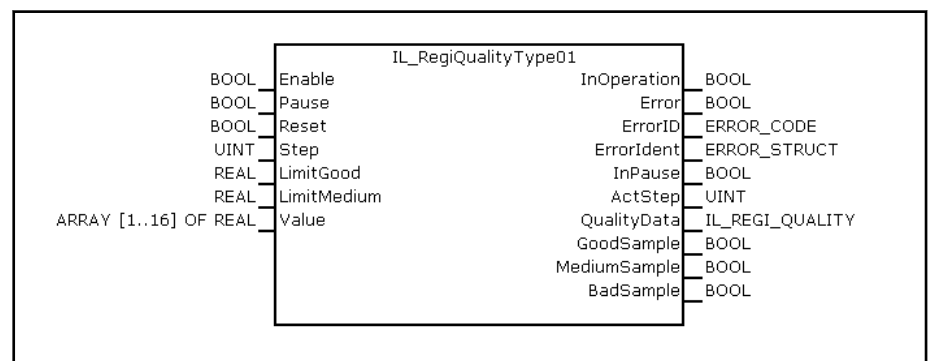


Abb.7-106: Funktionsbaustein IL_RegiQualityType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Berechnung der Daten in jedem Zyklus solange Enable = TRUE
	Pause	BOOL	Die Daten der Ausgänge werden nicht verändert, solange Pause = TRUE
	Reset	BOOL	Reset des FB
	Step	UINT	Aktueller Berechnungsschritt
	LimitGood	REAL	Absolutwert [mm], den eine Registerabweichung maximal haben kann, damit das Druckbild noch mit "guter" Qualität gezählt wird
	LimitedMedium	REAL	Liegt der absolute Wert einer Registerabweichung [mm] zw. "LimitGood" und "LimitMedium", zählt das Druckbild zu "mittlerer" Qualität; Registerabweichungen größer als "LimitMedium" haben "schlechte" Qualität
	Value	ARRAY [1..16] OF REAL	Enthält die Registerabweichungen eines Druckbildes (bis zu 16 Farbmarken)

ML_TechRegi.library/MX_TechRegi.lib

E/A-Typ	Name	Datentyp	Beschreibung
VAR_OUTPUT	InOperation	BOOL	Fehlerfreie Bearbeitung, Ausgangsdaten sind gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorIdent)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung, siehe Fehlercodes , Seite 385
	InPause	BOOL	Zeigt an, dass der Eingang "Pause" aktiv ist und sich die Werte an den Ausgängen nicht ändern
	ActStep	UINT	Letzter verarbeiteter Schritt
	QualityData	IL_REGI_QUALITY	Enthält die berechneten Qualitätsdaten zum Druckprozess
	GoodSample	BOOL	Ist TRUE, wenn das aktuelle Druckbild "gute" Qualität hat
	MediumSample	BOOL	Ist TRUE, wenn das aktuelle Druckbild "mittlere Qualität" hat
	BadSample	BOOL	Ist TRUE, wenn das aktuelle Druckbild "schlechte" Qualität hat

Abb. 7-107: Schnittstellenvariablen Funktionsbaustein IL_RegiQualityType01

IL_REGI_QUALITY		
GoodSampleCount	UDINT	Anzahl der Druckbilder mit "guter" Qualität
MediumSampleCount	UDINT	Anzahl der Druckbilder mit "mittlerer" Qualität
BadSampleCount	UDINT	Anzahl der Druckbilder mit "schlechter" Qualität
GoodSampleRatio	REAL	Prozentualer Anteil der Druckbilder "guter" Qualität
MediumSampleRatio	REAL	Prozentualer Anteil der Druckbilder "mittlerer" Qualität
BadSampleRatio	REAL	Prozentualer Anteil der Druckbilder "schlechter" Qualität

Abb. 7-108: Definition verwendeter Strukturen bei IL_RegiQualityType01

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Pause	BOOL			FALSE	kontinuierlich
Reset	BOOL			FALSE	kontinuierlich
Step	UINT	0	65535	0	kontinuierlich
LimitGood	REAL	0.0	LimitMedium	1.0	steigende Flanke an "Enable"
LimitMedium	REAL	LimitGood	n. def.	2.0	steigende Flanke an "Enable"
Value	ARRAY [1..16] OF REAL	n. def.	n. def.	0.0	bei Änderung an "Step"

Abb. 7-109: Min.- Max.- und Default-Werte der Eingänge IL_RegiQualityType01

Funktionsbeschreibung

Um die Qualität eines Druckprozesses zu beurteilen, ist ein wichtiges Merkmal die Registerfehler der Druckbilder auszuwerten. Bei einem Druckbild ist der (betragsmäßig) größte Registerfehler einer Farbmarke entscheidend für die Qualität.

Über die Eingänge "LimitGood" und "LimitMedium" lassen sich entsprechende Fenster parametrieren, die die Beurteilung eines Druckbildes abhängig von der größten Registerabweichung in Value in "gute", "mittlere" und "schlechte" Qualität vornehmen. Wird für "LimitGood" und "LimitMedium" der gleiche Wert angegeben, findet nur die Einteilung in "good" und "bad" statt (siehe nachfolgende Abbildung).

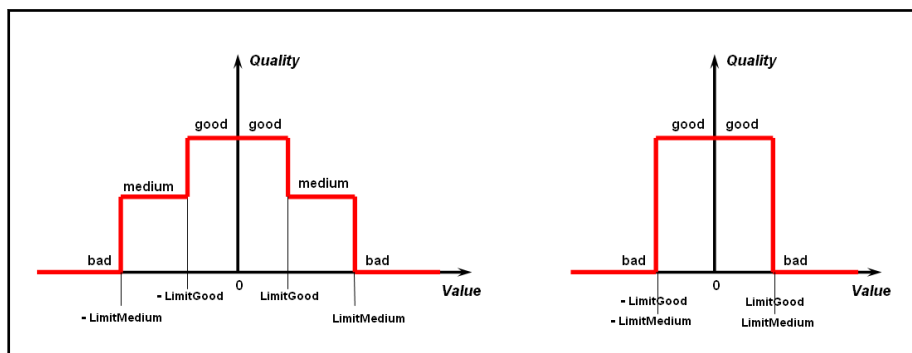


Abb.7-110: Prinzip der Qualitätsbeurteilung bei IL_RegiQualityType01

In der Datenstruktur am Ausgang "QualityData" werden die Druckbilder der drei Qualitätskategorien aufsummiert, sowie deren Anteil an der Gesamtstückzahl ermittelt. Mit einer steigenden Flanke am Eingang „Reset“ werden alle Werte auf '0' gesetzt.

Die Abarbeitung erfolgt kontinuierlich bei gesetztem **Enable**-Eingang. Der Ausgang "ActStep" zeigt an, dass ein neues Datenpaket mit Messwerten erhalten und ausgewertet wurde. Die booleschen Ausgänge "Good-/Medium-/BadSample" zeigen zusätzlich an, welche Qualität das aktuelle Druckbild hat. Ist der Eingang "Pause" gesetzt, so wird der aktuelle Zustand eingefroren, der Ausgang "InPause" wird gesetzt und es findet keine weitere Auswertung statt.

Fehlercodes

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_INVALID_ERROR	16#0817	16#0001	Wert von "LimitGood" < 0.0
INPUT_INVALID_ERROR	16#0817	16#0002	Wert von "LimitMedium" < "LimitGood"

Abb.7-111: Fehlercodes Funktionsbaustein IL_RegiQualityType01

8 ML_TechCrank.library

8.1 Einführung und Übersicht

Allgemeines Technologie-Funktionsbausteine (Tech-FB) erweitern die Grundfunktionalität des Zielsystems IndraMotion MLC und stellen applikationsspezifische Funktionalitäten zur Verfügung. Die hier beschriebene Bibliothek beinhaltet Funktionalitäten für Kurbel und Kniehebelkinematiken.

Die hier beschriebenen Funktionsbausteine werden über die Bibliothek "ML_TechCrank.compiled-library" für das Zielsystem IndraMotion MLC zur Verfügung gestellt.

Diese Dokumentation beschreibt die Funktionalität, sowie die Ein- und Ausgänge der Technologie-Funktionsbausteine.

Voraussetzungen Technologie-Funktionsbausteine der Bibliothek "ML_TechCrank.compiled-library" benötigen die Firmware-Unterstützung des Zielsystems IndraMotion MLC. Spezifische Voraussetzungen der Technologie-Funktionsbausteine sind im Kapitel der entsprechenden Funktionsbausteine dokumentiert.

Übersicht über die Funktionsbausteine Die folgende Tabelle gibt einen Überblick über die verfügbaren Funktionsbausteine.

Funktionsbaustein	Beschreibung
Kurbelkinematik	
MB_CamTableCrank	Berechnet eine Kurvenscheibe zur Transformation einer translatorischen virtuellen Leitachseposition in rotatorische Werte (Kurbelwinkel)
MB_CamTableCrankSuperimposed	Überlagert eine Anwenderkurvenscheibe mit der Transformationskurvenscheibe zu einer überlagerten Kurvenscheibe
MB_PhiToXvirt	Wandelt Kurbelwinkel (Phi) in virtuelle translatorische Positionswerte (Xvirt) zur Verwendung durch das SPS-Programm um
MB_MasterToPhi	Gibt Kurbelwinkel aus, die anhand der Leitachseposition und der überlagerten Kurvenscheibe berechnet worden sind
MB_XvirtToXmech	Wandelt virtuelle translatorische Positionswerte (Xvirt) in mechanische translatorische Positionswerte (Xmech) um
Kniehebelkinematik	
MB_BellCrankData	Berechnet die Y-Positionen und die Kurbelwinkel Phi an der unteren und oberen Polstelle
MB_CamTableBellCrank	Berechnet eine Kurvenscheibe zur Transformation für einen Kniehebelantrieb
MB_CamTableBellCrankSuperimposed	Überlagert eine Anwenderkurvenscheibe mit der Transformationskurvenscheibe zu einer überlagerten Kurvenscheibe
MB_PhiToYvirt	Wandelt den Kurbelwinkel Phi in die mechanische und virtuelle translatorische Position um
MB_YvirtToPhi	Wandelt die virtuelle translatorische Position in den Kurbelwinkel Phi um
MB_YvirtToPhiPoly5	Wandelt die virtuelle translatorische Position in den Kurbelwinkel Phi mittels eines Ersatzpolynoms 5. Ordnung um

ML_TechCrank.library

Funktionsbaustein	Beschreibung
MB_YvirtToYmech	Wandelt eine virtuelle translatorische Position in die mechanische Position um
MB_YVirtToWorkRange	Gibt den Arbeitsbereich der Kurbel 1 zurück

Abb.8-1: ML_TechCrank.compiled-library

8.2 Funktionen und Funktionsbausteine für die Kurbelkinematik

8.2.1 Einführung und Übersicht

Die in diesem Abschnitt beschriebenen Funktionen und Funktionsbausteine dienen zur Konvertierung von translatorischer in rotatorische Bewegung (Kurbelwinkel) und umgekehrt und sind für Kurbelkinematik-Applikationen bestimmt.

Kurbelkinematiken werden häufig zum Antrieb von Quersiegelbacken in Schlauchbeutelmaschinen oder von Form- und Stanzwerkzeugen in Thermoformmaschinen eingesetzt. Der translatorische Schlitten in dieser Kinematik wird durch die Drehbewegung einer Kurbel bewegt. Die Achse des Servomotors, der die Kurbel antreibt, ist gegenüber der Achse des translatorischen Schlittens versetzt angeordnet.

Die unten dargestellte Kurbelkinematik gibt Soll- und Istwerte (Lage, Geschwindigkeit usw.) in translatorischen Einheiten aus, obwohl die Ausgabe des Messsystems in rotatorischen Einheiten erfolgt. Deshalb müssen die Soll- und Istwerte von translatorischen in rotatorische Einheiten umgewandelt werden und umgekehrt.

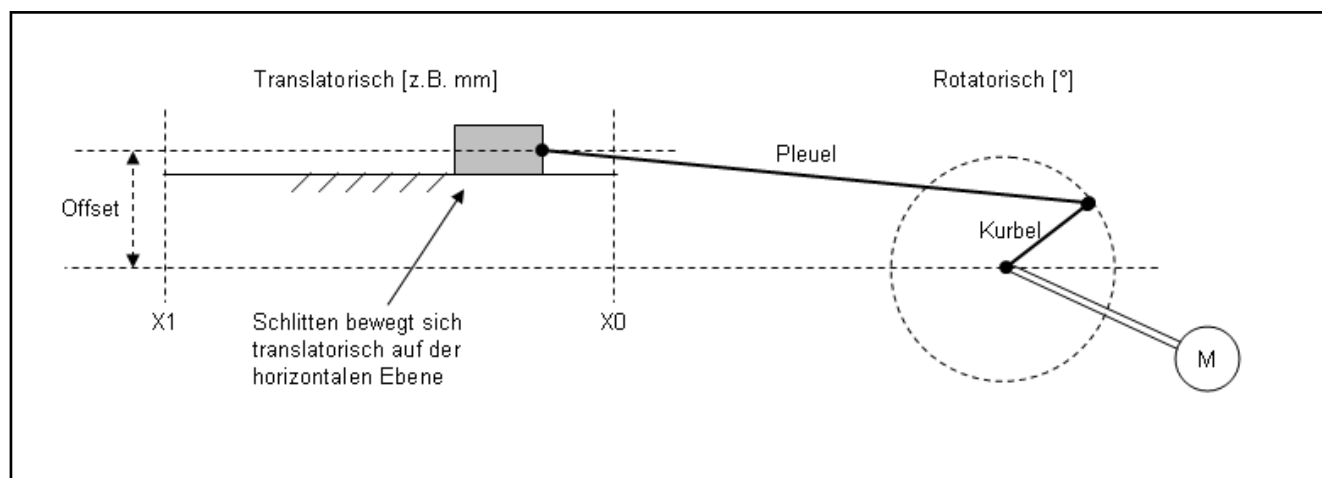


Abb.8-2: Einzelachs-Kurbelkinematik mit versetzter Kurbelachse

Die folgenden Funktionen und Funktionsbausteine werden unterstützt:

Funktion	Beschreibung
MB_MasterToPhi	Gibt Kurbelwinkel aus, die anhand der Leitachseposition und der überlagerten Kurvenscheibe berechnet worden sind

Abb.8-3: Kurbelkinematik-Funktion

Funktionsbaustein	Beschreibung
MB_CamTableCrank	Berechnet eine Kurvenscheibe zur Transformation einer translatorischen virtuellen Leitachseposition in rotatorische Werte (Kurbelwinkel)
MB_CamTableCrankSuperimposed	Überlagert eine Anwenderkurvenscheibe mit der Transformationskurvenscheibe zu einer überlagerten Kurvenscheibe
MB_PhiToXvirt	Wandelt Kurbelwinkel (Phi) in virtuelle translatorische Positionswerte (Xvirt) zur Verwendung durch das SPS-Programm um
MB_XvirtToXmech	Wandelt virtuelle translatorische Positionswerte (Xvirt) in mechanische translatorische Positionswerte (Xmech) um

Abb.8-4: Kurbelkinematik-Funktionsbausteine

Typische Anwendungsbeispiele sind:

Anwendungsfall 1: Betriebsart Einzelachse

Der Begriff Einzelachsbetrieb bezieht sich auf den lage- oder geschwindigkeitsgeregelten Betrieb der translatorischen Achse. In der vorliegenden Applikation wird eine virtuelle Leitachse (mit translatorischer Wichtung) durch Lage- oder Geschwindigkeitsregelungs-Funktionsbausteine bewegt (z. B. MC_MoveAbsolute, MC_MoveRelative oder MC_MoveVelocity).

Die virtuelle Leitachseposition wird durch die Transformationskurvenscheibe von translatorischen Positionswerten in Kurbelwinkelwerte umgerechnet. Die Transformationskurvenscheibe wird durch den Funktionsbaustein MB_CamTableCrank berechnet und mit dem Funktionsbaustein MC_CamIn aktiviert. Die Kurbelachse folgt dann der virtuellen Leitachse (die translatorische Einheiten ausgibt), wobei die Transformationskurvenscheibe die translatorischen Positionswerte in Kurbelwinkelwerte umwandelt. Einzelheiten entnehmen Sie bitte [Kap. 8.2.3 "MB_CamTableCrank" auf Seite 393](#).

Anwendungsfall 2: Synchronbetrieb mit Kurvenscheibe

In dieser Applikation folgt eine translatorische Achse mit einem nutzerdefinierten Kurvenscheibenprofil einer Leitachse. Die anwenderdefinierte Kurvenscheibe bezieht sich nur auf die translatorische Achse. Das Übertragungsverhalten der Kurbel muss daher nicht in der anwenderdefinierten Kurvenscheibe berücksichtigt werden.

Statt dessen wird die anwenderdefinierte Kurvenscheibe durch den Funktionsbaustein MB_CamTableSuperimposed mit einer Transformationskurvenscheibe überlagert, und die resultierende Kurvenscheibe wird durch das SPS-Programm an den Antrieb übertragen. Einzelheiten entnehmen Sie bitte [Kap. 8.2.4 "MB_CamTableCrankSuperimposed" auf Seite 396](#).

Umgekehrte Transformation

In den beiden oben beschriebenen Anwendungsbeispielen werden translatorische in rotatorische Einheiten umgewandelt (Sollwerterstellung). Damit die SPS jedoch anschließend Lage- bzw. Geschwindigkeitsbefehle verarbeiten kann, müssen die rotatorischen Einheiten wieder in translatorische Einheiten umgewandelt werden. Hierzu dient der Funktionsbaustein MB_PhiToXvirt. Einzelheiten entnehmen Sie bitte [Kap. 8.2.5 "MB_PhiToXvirt" auf Seite 399](#).

ML_TechCrank.library

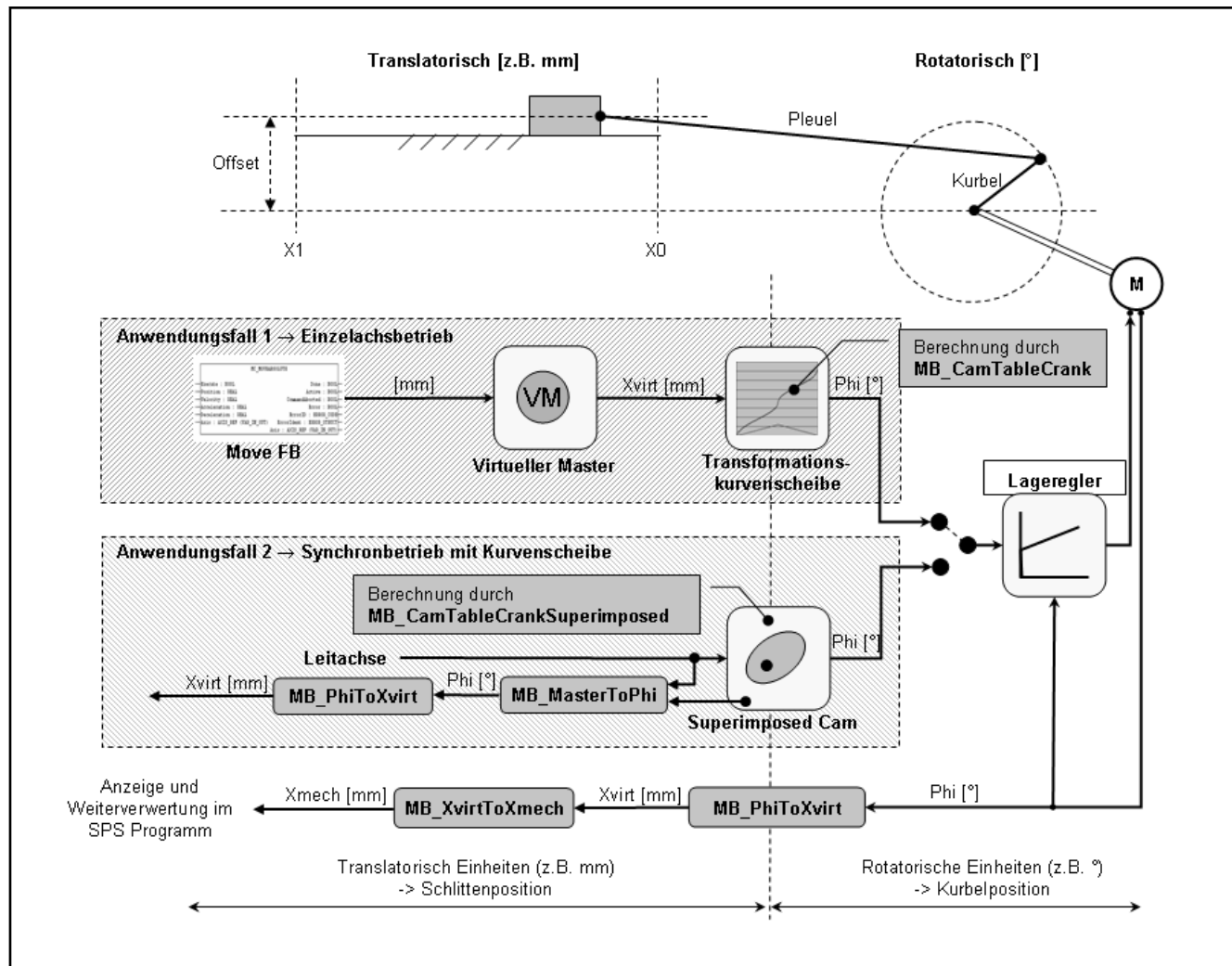
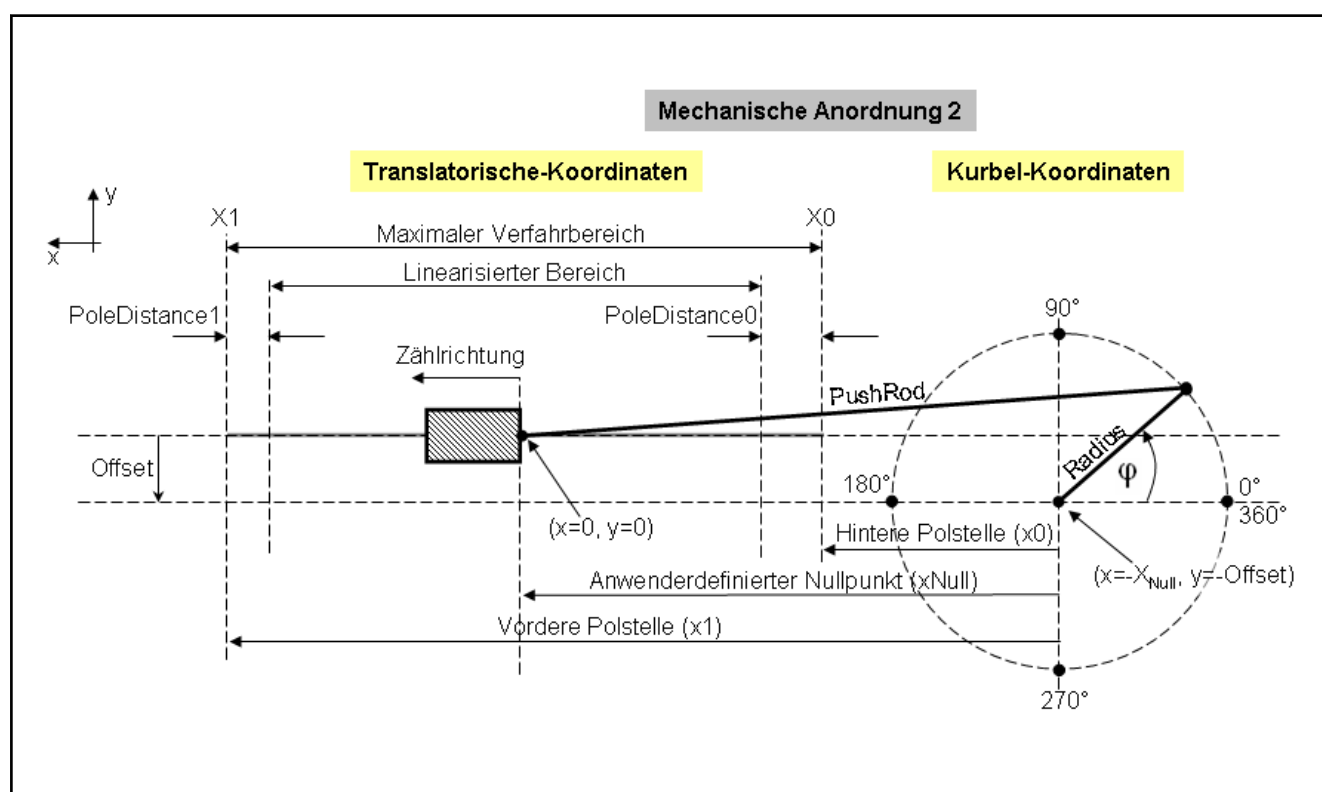
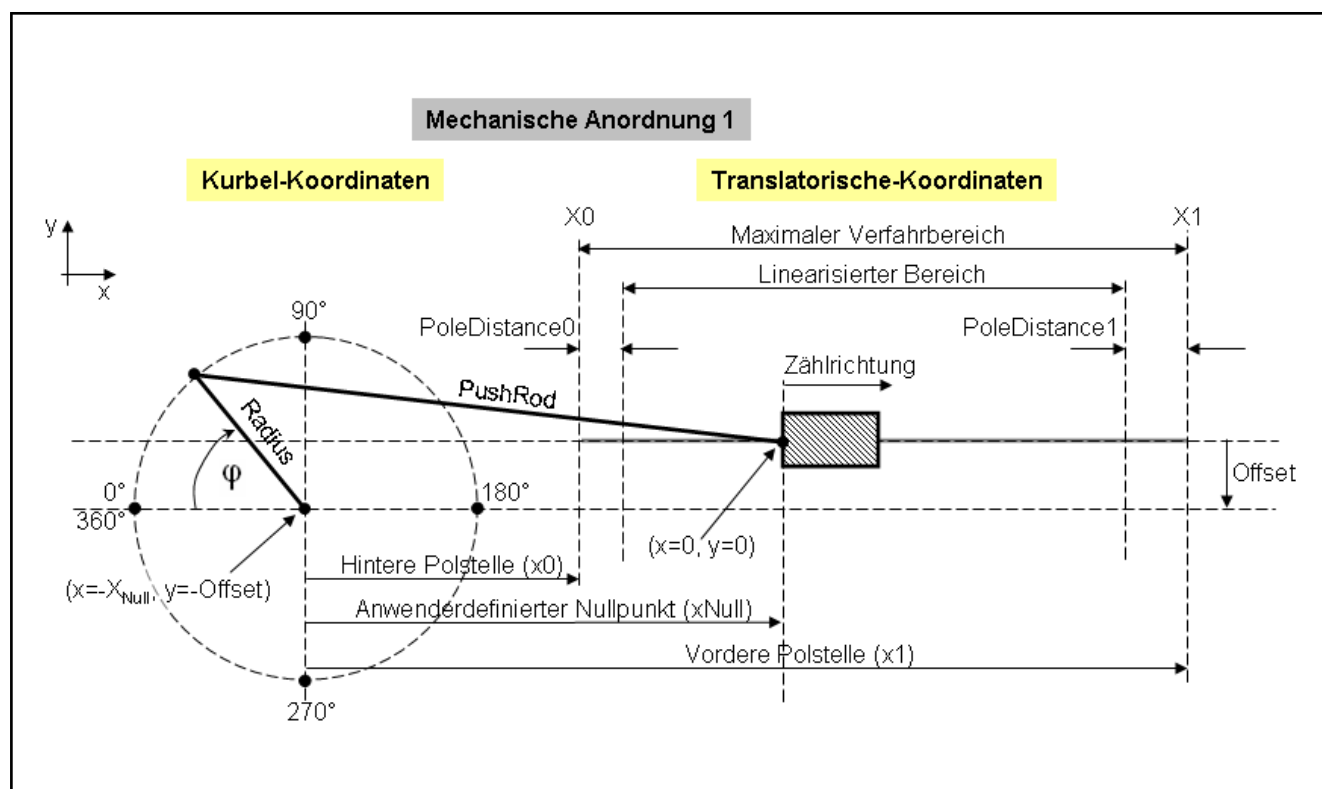


Abb.8-5: Schematische Darstellung der Kurbelkinematik-Technologie-Funktionsbausteine

8.2.2 Allgemeine Definitionen

Definition der grundlegenden Variablen bei der Kurbelkinematik

Die nachstehenden Abbildungen stellen die Zusammenhänge zwischen Kurbel- und translatorischen Koordinaten sowie die Bedeutung verschiedener mechanischer Parameter dar.



ML_TechCrank.library

Variable/Begriff	Definition
XNull	Abstand zwischen Kurbelachse und Schlitten-Nullpunkt
PoleDistance0 PoleDistance1	Abstand von der Polstelle X0 / X1 (jenseits des linearisierten Bereichs wird eine Kompensationsfunktion angewendet)
Maximaler Verfahrbereich	Verfahrbereich des translatorischen Koordinatensystems = X1-X0
Linearisierter Bereich	Im linearisierten Bereich ist kein Ersatzpolynom wirksam
Transformationskurvenscheibe	Wandelt translatorische (mm) Koordinaten in Kurbelwinkel (Grad) um

Abb.8-8: Variable und Begriffe

Zählrichtung

Für die in [Abb. 8-6 " Mechanische Anordnung 1: Zusammenhang von Kurbel- und translatorischen Koordinaten bei der Kurbelkinematik"](#) auf Seite 391 und [Abb. 8-7 " Mechanische Anordnung 2: Zusammenhang von Kurbel- und translatorischen Koordinaten bei der Kurbelkinematik"](#) auf Seite 391 abgebildeten mechanischen Anordnungen gelten allgemein die folgenden Zählrichtungen:

- Zählrichtung des Kurbelwinkels:
 - Mechanische Anordnung 1: Im Uhrzeigersinn (Rechtsdrehung), beginnend links mit 0° und endend mit 360°
 - Mechanische Anordnung 2: Im Gegenuhrzeigersinn (Links-drehung), beginnend rechts mit 0° und endend mit 360°
- Zählrichtung des translatorischen Schlittens:
 - Mechanische Anordnung 1: Nach rechts hin zunehmende numerische Werte
 - Mechanische Anordnung 2: Nach links hin zunehmende numerische Werte

Mechanische (Xmech) und virtuelle (Xvirt) translatorische Position

Mechanisch kann sich der Schlitten (translatorische Achse) zwischen der hinteren (X0) und der vorderen (X1) Polstelle bewegen. Zwischen diesen beiden Begrenzungen liegt der "maximale Verfahrbereich". Jeder mechanischen translatorischen Position (Xmech) entsprechen zwei verschiedene Kurbelwinkel. Deshalb dient die virtuelle translatorische Position (Xvirt) dazu, jedem Kurbelwinkel eine eindeutige translatorische Position zuzuordnen. Die virtuelle translatorische Position bewegt sich daher in positive Richtung, auch wenn die mechanische Position (Xmech) durch Überfahren der Polstelle die Richtung umkehrt. Der Moduloüberlauf von Xvirt wird vom Nutzer über den Nullpunkt (Xnull) definiert. Des weiteren entspricht der Modulowert von Xvirt dem doppelten Verfahrbereich der Kurbelkinematik. Der Verfahrbereich wird wie folgt berechnet:

$$\text{Travelrange} = (\text{FrontPolePoint} - \text{RearPolePoint})$$

$$\text{Travelrange} = \sqrt{(\text{Radius} + \text{PushRod})^2 - \text{Offset}^2} - \sqrt{(\text{Radius} - \text{PushRod})^2 - \text{Offset}^2}$$

Abb.8-9: Verfahrbereichsgleichung für die Kurbelkinematik

Die folgende Abbildung stellt Xvirt und Xmech an einem Beispiel dar:

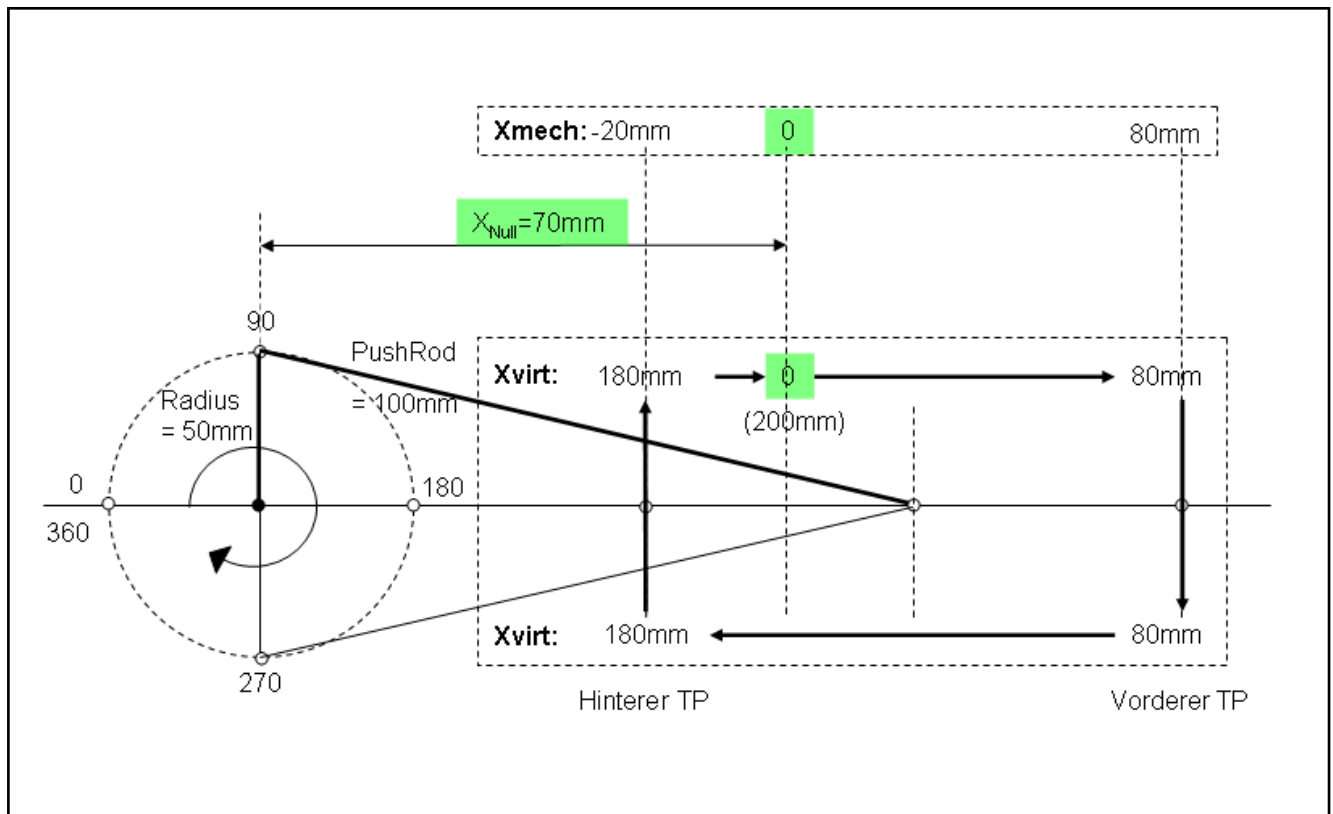


Abb.8-10: Zählrichtung der Schlittenpositionen (X_{virt} und X_{mech})

8.2.3 MB_CamTableCrank

Kurzbeschreibung

Der Funktionsbaustein MB_CamTableCrank berechnet anhand der kurbel-spezifischen Eingangswerte eine Transformationskurvenscheibe (mit 1024 Datenpunkten). Diese Transformationskurvenscheibe wandelt translatorische Positionswerte in rotatorische Positionswerte (Kurbelwinkel) zur Verwendung in einer Kurbelkinematik um. Sie ermöglicht die Kopplung einer translatorischen virtuellen Leitachse mit einem rotatorischen Kurbelantrieb. Einzelheiten entnehmen Sie bitte [Kap. 8.2.1 "Einführung und Übersicht" auf Seite 388](#).

Im Einzelachs-betrieb bewegt sich die virtuelle Achse in translatorischen Einheiten, während der Antrieb der Transformationskurvenscheibe folgt.

Eine Bewegung über den linearisierten Bereich hinaus kann durch Vorgabe je eines Wertes PoleDistance0 und PoleDistance1 realisiert werden. Innerhalb des Bereiches "PoleDistance" wird auf die Position eine Ausgleichsfunktion (Ersatzpolynom 5. Ordnung) angewendet, die den Kurbelantrieb approximiert und die resultierende Antriebsdynamik begrenzt. Beim Übergang in den und aus dem linearisierten Lagebereich sind Geschwindigkeit und Beschleunigung konstant.

Der Funktionsbaustein stellt die berechnete Transformationskurvenscheibe über den VAR_IN_OUT "CamTable" zur Verfügung. Das SPS-Programm muss diese Kurvenscheibentabelle in den Antrieb laden (z. B. über MB_WriteListeParameter), damit sie verwendet werden kann (z. B. von MC_CamIn Funktionsbaustein).

ML_TechCrank.library

Schnittstellenbeschreibung

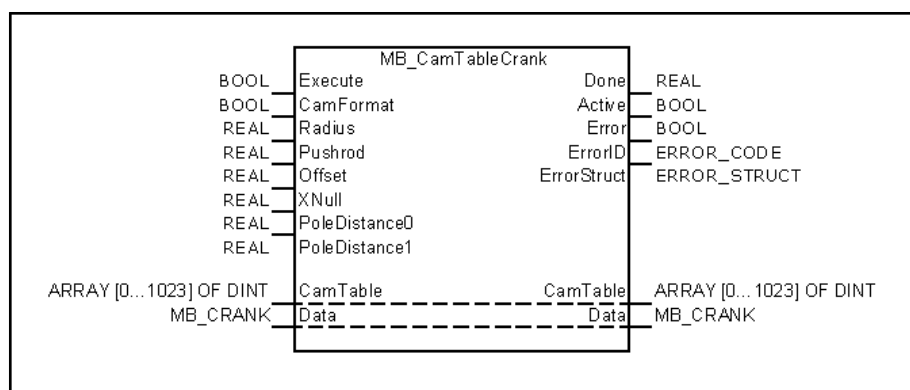


Abb.8-11: Funktionsbaustein MB_CamTableCrank

E/A-Typ	Name	Datentyp	Erläuterung
VAR_IN_OUT	CamTable	ARRAY[0..1023] OF DINT	Array mit den Daten der errechneten Transformationskurvenscheibe
	Data	MB_CRANK	Struktur mit intern berechneten Daten der Kurbelkinematik. Die Daten werden in diesem Funktionsbaustein berechnet und an andere Funktionsbausteine übergeben
VAR_INPUT	Execute	BOOL	Durch eine positive Flanke wird die Berechnung der Transformationskurvenscheibe ausgelöst
	CamFormat	BOOL	TRUE = Neues Format (letzter Punkt = erster Punkt = 360°) FALSE = altes Format (letzter Punkt = 360° - d)
	Radius	REAL	Kurbellänge in [mm]
	Pushrod	REAL	Länge des Pleuels in [mm]
	Offset	REAL	Versatz der Schlittenebene gegenüber dem Kurbelmittelpunkt [mm]
	XNull	REAL	Abstand zwischen Kurbelmittelpunkt und Schlitten-Nullpunkt [mm]
	PoleDistance0	REAL	Abstand zwischen der hinteren Polstelle X0 [mm] und dem linearisierten Bereich, in dem die Bewegung durch die begrenzte Antriebsdynamik beeinflusst wird
	PoleDistance1	REAL	Abstand zwischen der vorderen Polstelle X1 [mm] und dem linearisierten Bereich, in dem die Bewegung durch die begrenzte Antriebsdynamik beeinflusst wird
VAR_OUTPUT	Done	BOOL	Berechnung abgeschlossen, Kurvenscheibentabellen- und Ausgangsdaten (Data) sind gültig
	Active	BOOL	Funktionsbaustein wird gerade ausgeführt
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

Abb.8-12: Schnittstellenvariablen Funktionsbaustein MB_CamTableCrank

Datenstruktur MB_Crank

Die Datenstruktur MB_Crank ist für den internen Datenaustausch zwischen Funktionsbausteinen und Funktionen zuständig. Der Inhalt der Datenstruktur

MB_CRANK wird von den Funktionsbausteinen MB_CamTableCrank und MB_CamTableCrankSuperimposed berechnet und gilt für alle anderen relevanten Funktionsbausteine und Funktionen.

Randbedingung Die folgenden Randbedingungen müssen erfüllt werden, um SPS-Fehlermeldungen zu vermeiden:

Randbedingung	Grund
$Pleuel \geq Radius + Offset $	Wird diese Randbedingung nicht erfüllt, kann der translatorische Schlitten mechanisch nicht auf einer konstanten y-Koordinate verfahren
$(XNull \geq \text{hinterer TP}) \ \& \ (XNull \leq \text{hinterer TP} + 2 \times \text{Verfahrbereich})$	Der Nullpunkt muss innerhalb des Verfahrbereichs liegen

Abb.8-13: Randbedingungen

Um den translatorischen Schlitten im Einzelachsbetrieb verfahren zu können, ist die folgende Konfiguration in IndraWorks durchzuführen:

- Translatorische Wichtung der virtuellen Achse mit einem Modulowert von $(2 \times \text{Verfahrbereich})$.
- Rotatorische Wichtung des Kurbelantriebs (reale Achse) mit einem Modulowert von 360° .



Zur Konfiguration der Skalierungsfaktoren muss sich IndraWorks im Onlinebetrieb und der Antrieb im Parametriermodus befinden.

Anwendungsbeispiel für MB_CamTableCrank

Das folgende Ablaufschema gilt für Einzelachsbetrieb:

1. Der Funktionsbaustein MB_CamTableCrank berechnet die Transformationskurvenscheibe.
2. Die berechnete Transformationskurvenscheibe wird durch den Funktionsbaustein MB_WriteListParameter auf den Kurbelantrieb geschrieben.
3. Über den Funktionsbaustein MC_Power wird der Kurbelantrieb eingeschaltet.
4. Mit dem Funktionsbaustein MC_Home den Kurbelantrieb referenzieren. Dieser Schritt ist erforderlich, wenn sich am Kurbelantriebsmotor kein Absolutgeber befindet.
5. Eingabe der Kurbelantriebs-Istposition in den Funktionsbaustein MB_PhiToXvirt. Die Kurbel-Istposition wird von Kurbelwinkelgraden in einen virtuellen translatorischen Wert in mm umgerechnet.
6. Um die virtuelle Leitachse in die Istposition des Kurbelantriebs zu bewegen, muss nun der Ausgangswert Xvirt des Funktionsbausteins MB_PhiToXvirt an den Funktionsbaustein MC_MoveAbsolute für die virtuelle Achse übergeben werden. Nach dieser Abfolge befindet sich die virtuelle Leitachse in der Istposition des Kurbelantriebs, und der Kurbelantrieb ist – ohne eine Aufsynchonisierbewegung ausgeführt zu haben – mit der virtuellen Leitachse synchronisiert.
7. Umschaltung des Kurbelantriebs in den Kurvenscheiben-Betriebsmodus über MC_CamIn. Einstellen von CamShaftDistance = 360, Getriebeübersetzung auf 1:1, Auswählen der Transformationskurvenscheibe und der virtuellen Leitachse als Leitachse. Der Kurbelantrieb führt zur Synchronisation mit der virtuellen Leitachse keine Bewegung aus, da die virtuelle Leitachse in Schritt 6 auf die Kurbelposition eingestellt wurde.

ML_TechCrank.library

Wurde Schritt 6 nicht ausgeführt, führt der Kurbelantrieb eine dynamische Synchronisationsbewegung aus.

8. Nun kann die virtuelle Achse mit den Funktionsbausteinen MC_MoveAbsolute und MC_MoveVelocity im Einzelachsbetrieb bewegt werden (die Kurbelachse folgt nun der Leitachse und wendet dabei die Transformationskurvenscheibe an).

8.2.4 MB_CamTableCrankSuperimposed

Kurzbeschreibung Der Funktionsbaustein MB_CamTableSuperimposed überlagert die nutzerdefinierte Kurvenscheibe (CamInput) auf die Transformationskurvenscheibe und gibt die resultierende überlagerte Kurvenscheibe über den Ausgang CamOutput aus. Die Transformationskurvenscheibe wird innerhalb des Funktionsbausteins berechnet, vergleichbar mit dem Funktionsbaustein MB_CamTableCrank. Das Überlagerungsprinzip ist in [Abb. 8-14 "Prinzip der Kurvenscheiben-Überlagerung" auf Seite 397](#) gezeigt. Die nutzerdefinierte Kurvenscheibe muss 1024 Datenpunkte enthalten und die Bewegung der translatorischen Achse in Bezug auf die Leitachse definieren (ungeachtet der Kurbelkinematik). Der 100-Prozent-Wert der Tabelle entspricht der Bewegung (2 x Verfahrbereich = Modulowert der virtuellen translatorischen Achse).

Die berechnete überlagerte Kurvenscheibe muss mit Hilfe des Funktionsbausteins MB_WriteListParameter auf den Kurbelantrieb geschrieben und vom SPS-Programm über den Funktionsbaustein MC_CamIn aktiviert werden.

Hinweise für den Anwender

Abhängig vom jeweiligen Endpunkt der nutzerdefinierten Kurvenscheibe werden die folgenden Fälle unterschieden:

- Liegt der Endpunkt der nutzerdefinierten Kurvenscheibe nahe bei 100%, führt die Kurbel keine Richtungsumkehr durch (sondern dreht sich kontinuierlich in die gleiche Richtung). Dies ist energetisch der günstigste Ablauf, da die natürliche Bewegung der Kurbel ausgenutzt wird.
- Beträgt der Endpunkt der nutzerdefinierten Kurvenscheibe 0%, findet eine Hin- und Herbewegung der translatorischen Achse statt, bei der die Kurbel fortwährend ihre Drehrichtung ändert (siehe Abbildung unten).

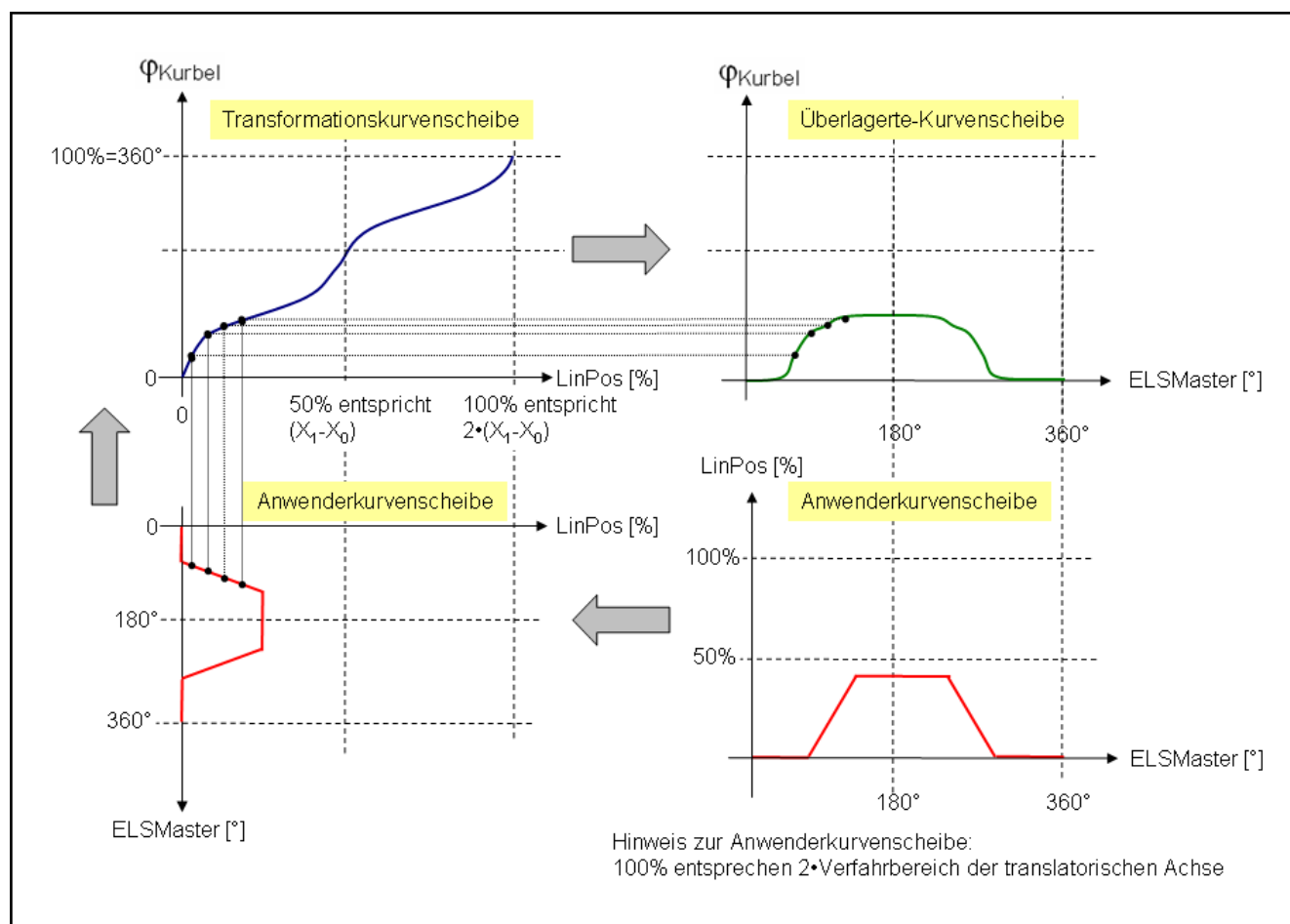


Abb.8-14: Prinzip der Kurvenscheiben-Überlagerung

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-15: Referenztable MB_CamTableCrankSuperimposed

ML_TechCrank.library

Schnittstellenbeschreibung

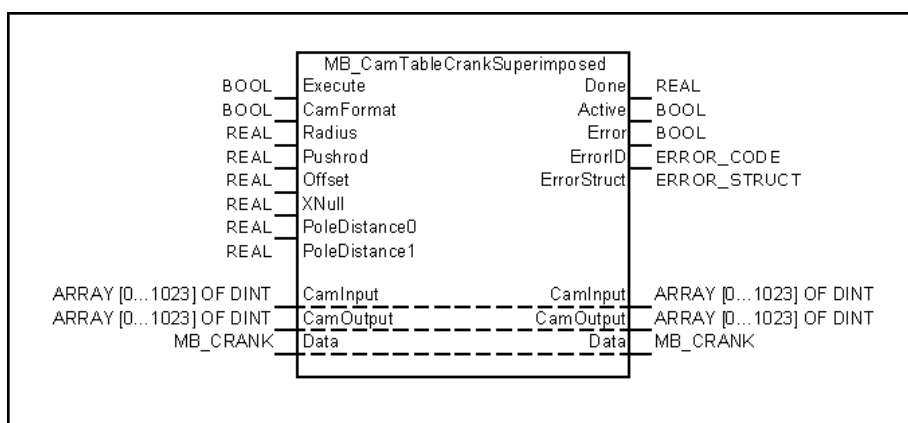


Abb.8-16: Funktionsbaustein MB_CamTable_Superimposed

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	CamInput	ARRAY[0..1023] OF DINT	Array mit den Daten der nutzerdefinierten Kurvenscheibe zur Voreinstellung des translatorischen Bewegungsprofils
	CamOutput	ARRAY[0..1023] OF DINT	Array mit den Daten der berechneten überlagerten Kurvenscheibe zur Aktivierung der Kurbel
	Data	MB_CRANK	Struktur mit intern berechneten Daten der Kurbelkinematik. Die Daten werden in diesem Funktionsbaustein berechnet und dann von anderen Funktionsbausteinen verwendet
VAR_INPUT	Execute	BOOL	Durch eine positive Flanke wird die Berechnung der überlagerten Kurvenscheibe ausgelöst
	CamFormat	BOOL	TRUE = Neues Format (letzter Punkt = erster Punkt = 360°) FALSE = altes Format (letzter Punkt = 360° - d)
	Radius	REAL	Kurbellänge in [mm]
	Pushrod	REAL	Länge des Pleuels in [mm]
	Offset	REAL	Versatz der Schlittenebene gegenüber dem Kurbelmittelpunkt [mm]
	XNull	REAL	Abstand zwischen Kurbelmittelpunkt und Schlitten-Nullpunkt [mm]
	PoleDistance0	REAL	Abstand zwischen der hinteren Polstelle X0 [mm] und dem linearisierten Bereich, in dem die Bewegung durch die begrenzte Antriebsdynamik beeinflusst wird
	PoleDistance1	REAL	Abstand zwischen der hinteren Polstelle X1 [mm] und dem linearisierten Bereich, in dem die Bewegung durch die begrenzte Antriebsdynamik beeinflusst wird
VAR_OUTPUT	Done	BOOL	Berechnung abgeschlossen, Kurvenscheibentabellen- und Ausgangsdaten (Data) sind gültig
	Active	BOOL	Funktionsbaustein wird gerade ausgeführt
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)

E/A-Typ	Name	Datentyp	Beschreibung
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

Abb. 8-17: Schnittstellenvariablen Funktionsbaustein MB_CamTableSuperimposed

Randbedingung Einzelheiten entnehmen Sie bitte "[Randbedingung](#)" auf Seite 395.

Um den translatorischen Schlitten im Synchronbetrieb mit einer vorgegebenen, nutzerdefinierten Kurvenscheibe verfahren zu können, ist die folgende Konfiguration in IndraWorks durchzuführen:

- Rotatorische Wichtung des Kurbelantriebs (reale Achse) mit einem Modulwert von 360°.



Zur Konfiguration der Skalierungsfaktoren muss sich IndraWorks im Onlinebetrieb und der Antrieb im Parametriermodus befinden.

Anwendungsbeispiel überlagerte Kurvenscheibe

Das folgende Ablaufschema gilt für eine kurvenscheibensynchrone Applikation:

1. Eingabe einer nutzerdefinierten Kurvenscheibe über den Eingang CamInput des Funktionsbausteins (z. B. durch Einlesen einer Antriebs-Kurvenscheibe oder Auslesen aus einer Datei usw.)
2. Ausgabe einer überlagerten Kurvenscheibe durch den Funktionsbaustein MB_CamTableCrankSuperimposed, die durch Überlagerung einer nutzerdefinierten Kurvenscheibe in eine Transformationskurvenscheibe berechnet wurde (siehe [Abb. 8-14 "Prinzip der Kurvenscheiben-Überlagerung"](#) auf Seite 397).
3. Die berechnete überlagerte Kurvenscheibe wird durch den Funktionsbaustein MB_WriteListParameter auf den Kurbelantrieb geschrieben.
4. Über den Funktionsbaustein MC_Power den Kurbelantrieb einschalten.
5. Mit dem Funktionsbaustein MC_Home den Kurbelantrieb referenzieren. Dieser Schritt ist erforderlich, wenn kein Absolutgeber verwendet wird.
6. Der Lagesollwert der überlagerten Kurvenscheibe und die aktuelle Kurbelantriebsposition können auseinander liegen. Zur Zuschaltung der Kurvenscheibenbetriebsart (über MC_CamIn, mit den Einstellwerten CamShaftDistance=360, Getriebeübersetzung 1:1 und Auswahl der überlagerten Kurvenscheibe) bestehen zwei Möglichkeiten:
 - Zuschaltung der Kurvenscheibenbetriebsart am Kurbelantrieb ohne vorherige Positionskalibrierung. Der Kurbelantrieb führt dann eine dynamische Aufsynchronisation durch.
 - Synchronisieren des Kurbelantriebs mit der Leitachse durch Eingabe der Leitachseposition und der überlagerten Kurvenscheibe in den Funktionsbaustein MB_MasterToPhi. Anschließend – vor Umschalten auf Synchronbetrieb – muss der Kurbelantrieb mit dem Funktionsbaustein MC_MoveAbsolute so positioniert werden, dass er der Ausgangsposition des Funktionsbausteins MB_MasterToPhi entspricht.

8.2.5 MB_PhiToXvirt

Kurzbeschreibung Der Funktionsbaustein MB_PhiToXvirt wandelt Kurbelwinkelwerte (Phi) in virtuelle translatorische Positionswerte (Xvirt) um.

ML_TechCrank.library

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-18: Referenztable MB_PhiToXvirt

Schnittstellenbeschreibung

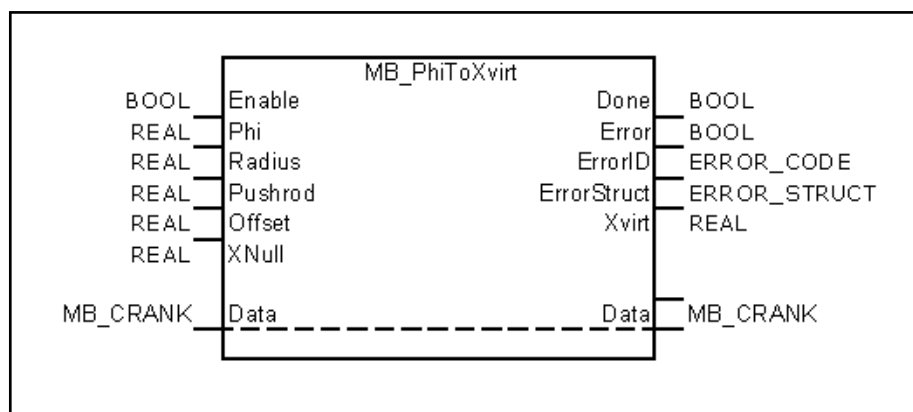


Abb.8-19: Funktionsbaustein MB_PhiToXvirt

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Data	MB_CRANK	Struktur mit den internen Daten der Kurbelkinematik (zu berechnen vor MB_CamTableCrank bzw. MB_CamTableCrankSuperimposed)
VAR_INPUT	Enable	BOOL	Berechnung von Xvirt in jedem Zyklus, während Enable=TRUE
	Phi	REAL	Kurbelposition in [°]
	Radius	REAL	Kurbellänge in [mm]
	Pushrod	REAL	Länge des Pleuels in [mm]
	Offset	REAL	Versatz der Schlittenebene gegenüber dem Kurbelmittelpunkt [mm]
	XNull	REAL	Abstand zwischen Kurbelmittelpunkt und Schlitten-Nullpunkt [mm]
VAR_OUTPUT	Done	BOOL	Berechnung abgeschlossen, Xvirt ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	Xvirt	REAL	Virtuelle translatorische Position (Xvirt)

Abb.8-20: Schnittstellenvariablen Funktionsbaustein MB_PhiToXvirt

8.2.6 MB_MasterToPhi

Kurzbeschreibung Die Funktion MB_MasterToPhi liefert den Kurbelwinkel (Phi), der anhand der Leitachsposition (Leitwert) und der (Überlagerungs-) Kurvenscheibe berechnet wurde. Die überlagerte Kurvenscheibe muss vor dem Aufruf dieser Funktion berechnet werden (durch MB_CamTableCrankSuperimposed). Anhand des Ausgabewertes der Funktion MB_MasterToPhi kann der Kurbelantrieb vor der Umschaltung in den Synchronbetrieb so positioniert werden, dass er der Position der Leitachse entspricht.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-21: Referenztable MB_MasterToPhi

Schnittstellenbeschreibung

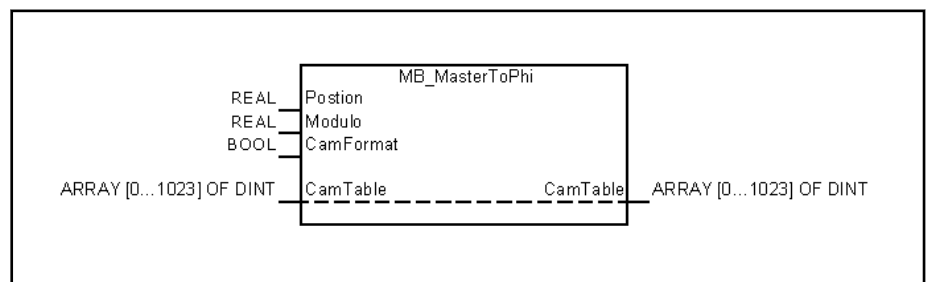


Abb.8-22: Funktion MB_MasterToPhi

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	CamTable	ARRAY[0..1023] OF DINT	Array mit den Daten der überlagerten Kurvenscheibe
VAR_INPUT	Position	REAL	Leitachsposition
	Modulo	REAL	Modulowert der Leitachse
	CamFormat	BOOL	TRUE = Neues Format (letzter Punkt = erster Punkt = 360°) FALSE = altes Format (letzter Punkt = 360° - d)

Abb.8-23: Schnittstellenvariablen Funktion MB_MasterToPhi

8.2.7 MB_XvirtToXmech

Kurzbeschreibung Der Funktionsbaustein MB_XvirtToXmech wandelt die virtuelle translatorische Position (Xvirt) in die mechanische translatorische Position (Xmech) des Schlittens um. Xmech-Werte können zu Anzeigezwecken verwendet werden (z. B. auf einem HMI-Gerät).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib

ML_TechCrank.library

IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-24: Referenztabelle MB_XvirtToXmech

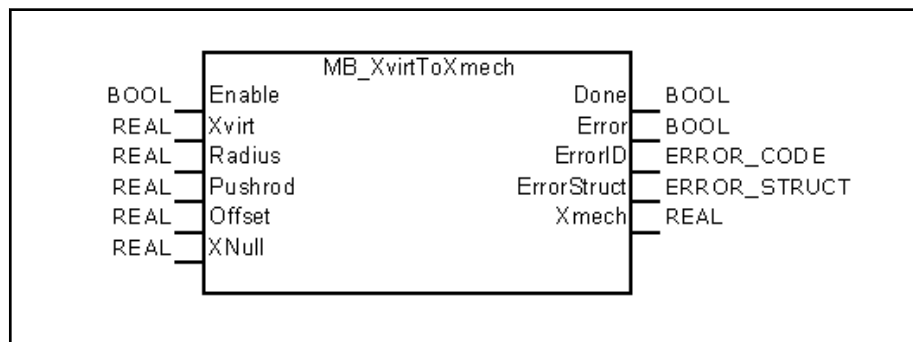
Schnittstellenbeschreibung

Abb.8-25: Funktionsbaustein MB_XvirtToXmech

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Funktionsbaustein führt die Berechnung durch, solange Enable=TRUE
	Xvirt	REAL	Virtuelle translatorische Position in [mm]
	Radius	REAL	Kurbellänge in [mm]
	Pushrod	REAL	Länge des Pleuels in [mm]
	Offset	REAL	Versatz der Schlittenebene gegenüber dem Kurbelmittelpunkt [mm]
	XNull	REAL	Abstand zwischen Kurbelmittelpunkt und Schlitten-Nullpunkt [mm]
VAR_OUTPUT	Done	BOOL	Ausgangswert "Phi" ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	Xmech	REAL	Mechanische translatorische Position (Xmech) des Schlittens

Abb.8-26: Schnittstellenvariablen Funktionsbaustein MB_XvirtToXmech

8.3 Funktionen und Funktionsbausteine für die Kniehebelkinematik

8.3.1 Einführung und Übersicht

Kniehebelkinematiken werden zum Beispiel zum Antrieb von Formwerkzeugen bei Thermoformmaschinen oder Pressen eingesetzt. Die nachfolgende Abbildung zeigt den mechanischen Aufbau der im folgenden verwendeten Kniehebelkinematik (Bell Crank Kinematic).

Bei dieser Kinematik bewirkt die Drehung der Kurbel 1 eine translatorische Bewegung in y-Richtung. Die Kurbel wird durch einen Servomotor angetrieben. Das Messsystem ist am Servomotor angebracht und arbeitet damit in rotatorischen Einheiten.

Bei der gezeigten Kniehebelkinematik werden Soll- und Istwerte (wie z.B. Position, Geschwindigkeit...) in translatorischen Einheiten vorgegeben bzw. angezeigt, obwohl das Messsystem rotatorisch arbeitet. Die im folgenden beschriebenen Tech-FBs führen entsprechende Transformationen durch.

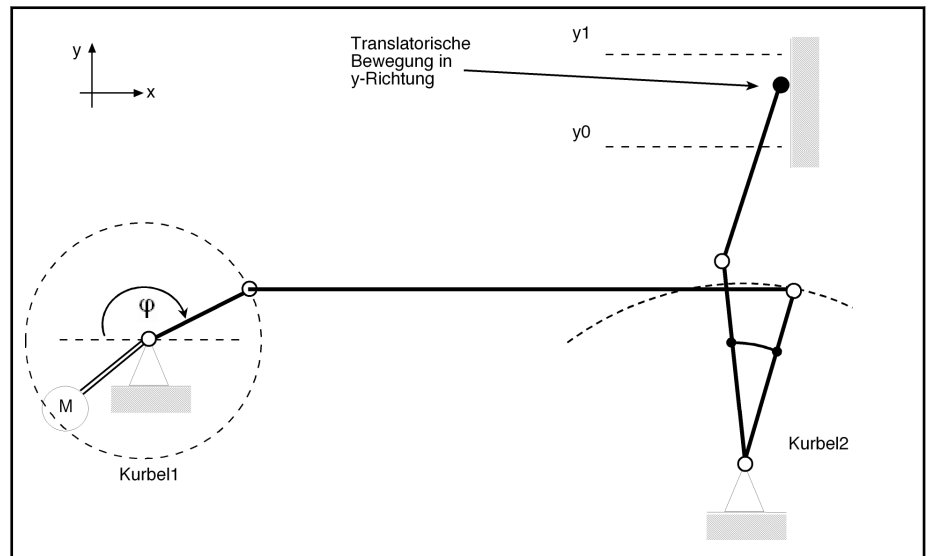


Abb.8-27: Kniehebelkinematik (Bell Crank Kinematik)

8.3.2 Allgemeine Definitionen

Grundlegenden Variablen

Die nachfolgende Abbildung zeigt den Zusammenhang zwischen Kurbel- und Translatorischen Koordinaten, sowie die Bedeutung der einzelnen Mechanikparameter.

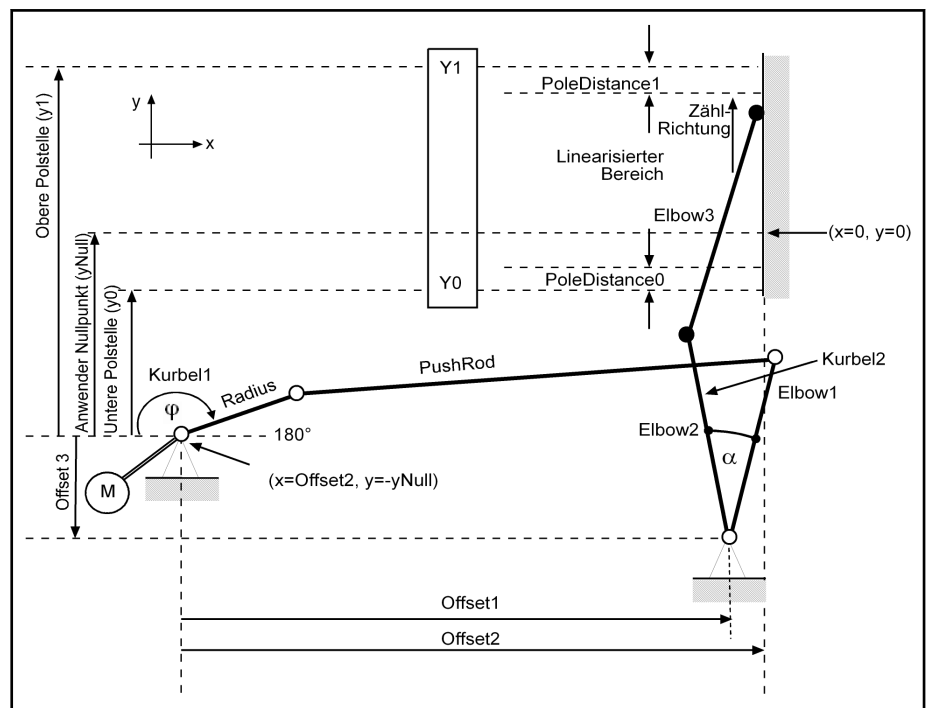


Abb.8-28: Zusammenhang von Kurbel- und translatorischen Koordinaten an der Kniehebelkinematik

ML_TechCrank.library

Zählrichtung

Allgemein gelten die folgenden Zählrichtungen für die in [Abb. 8-28 "Zusammenhang von Kurbel- und translatorischen Koordinaten an der Kniehebelkinematik"](#) auf Seite 403 gezeigte Anordnung:

- Zählrichtung des Kurbel-Winkels:
Im Uhrzeigersinn, links beginnend mit 0° und endend mit 360°
- Zählweise der translatorischen y-Position:
Nach oben grösser werdende Zahlenwerte

Mechanische (Y_{mech}) und virtuelle (Y_{virt}) translatorische Position

Eine mechanische translatorische Position (Y_{mech}) kann typischerweise über zwei verschiedene Winkelstellungen der Kurbel 1 erreicht werden. Damit ist die mechanische translatorische Position (Y_{mech}) nicht eindeutig einem Kurbelwinkel zuzuordnen. Um diese Mehrdeutigkeit zu lösen, wird die virtuelle translatorische Position (Y_{virt}) eingeführt, welche auch bei Richtungsumkehr der translatorischen Achse weiter in positive Richtung zählt.

Der Anwender kann über den Nullpunkt (Y_{null}) festlegen wo auf dem Verfahrbereich der mechanische Nullpunkt und damit auch der Moduloüberlauf liegt. Der Modulowert von Y_{virt} muss hierbei dem doppelten Verfahrbereich der translatorischen Achse entsprechen. Der Verfahrbereich (S_v) sei hierbei als Strecke zwischen der oberen (Y_1) und unteren (Y_0) Polstelle definiert.

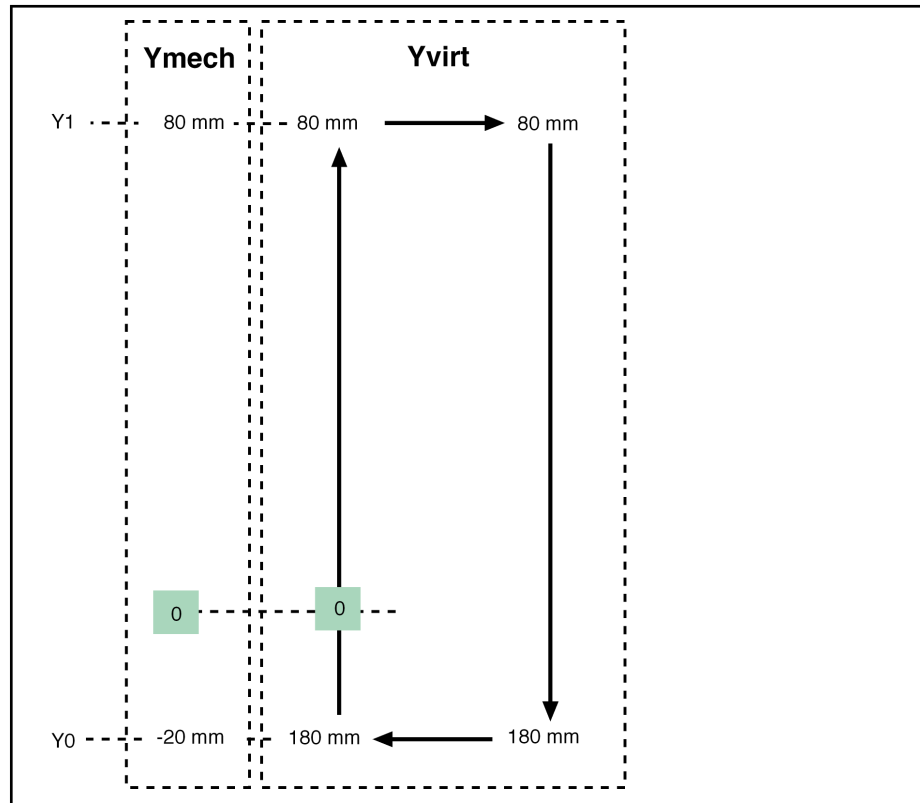


Abb.8-29: Zusammenhang zwischen Y_{virt} und Y_{mech}

Grundprinzip der Kurvenscheiben-transformationen

Die von den Kniehebelkinematik FB berechneten Kurvenscheiben transformieren eine Leitachsposition in einen Kurbelwinkel des Kniehebelantriebes. Grundsätzlich werden dabei die folgenden zwei Anwendungsfälle unterschieden:

- Im ersten Fall wird die reale Kurbelachse über eine Transformationskurvenscheibe an eine translatorisch gewichtete virtuelle Achse gekoppelt. Die virtuelle Achse kann damit in anwenderfreundlichen translatorischen Einheiten (z.B. mm, mm/s...) in allen verfügbaren Betriebsarten betrie-

ben werden und die Kurbel folgt gemäß der Transformationskurvenscheibe. Die notwendige Transformationskurvenscheibe wird hierbei durch den FB "MB_CamTableBellCrank" berechnet. Beim Zielsystem IndraMotion MLD unterstützt die virtuelle Achse keine Synchronbetriebsarten, wodurch dieser Anwendungsfall dort nur eingeschränkt sinnvoll ist.

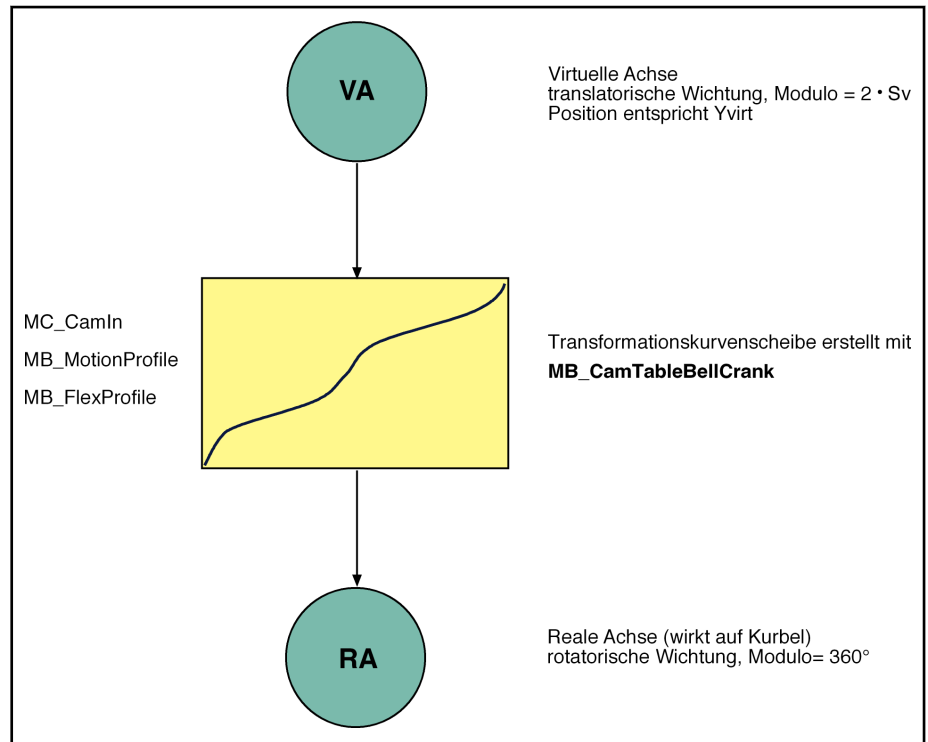


Abb.8-30: Funktionsweise der Transformationskurvenscheibe mit einer translatorischen virtuellen Achse

- Im zweiten Fall wird die reale Kurbelachse über eine überlagerte Anwenderkurvenscheibe an eine beliebige Leitachse mit frei wählbarer Modulowichtung gekoppelt.

Die Anwenderkurvenscheibe beschreibt hierbei die Bewegung der translatorischen Achse im Bezug zur Leitachse ohne das nichtlineare Übersetzungsverhalten der Kinematik zu berücksichtigen. Der FB "MB_CamTableBellCrankSuperimposed" überlagert dafür die Anwenderkurvenscheibe mit den Transformationsgleichungen der Kinematik und berechnet damit die überlagerte Kurvenscheibe. Damit kann der Kniehebelantrieb mit einer Anwenderkurvenscheibe an eine beliebige Leitachse mit frei wählbarem Moduluwert gekoppelt werden. Dieser Anwendungsfall ist speziell für das Zielsystem IndraMotion MLD sinnvoll, da der Kniehebelantrieb damit an eine Leitachse gekoppelt werden kann. Beim Zielsystem IndraMotion MLC hat dieser Anwendungsfall geringere Bedeutung, es kann jedoch damit eine zusätzliche virtuelle Achse eingespart werden.

ML_TechCrank.library

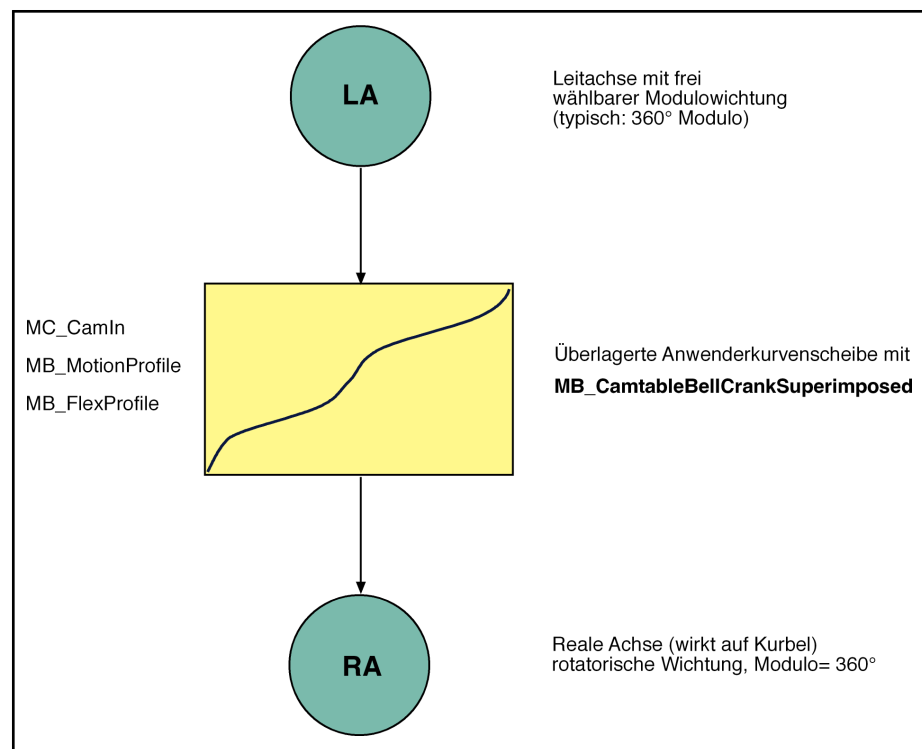


Abb.8-31: Funktionsweise der überlagerten Kurvenscheibe an einer beliebigen Leitachse

Vereinfachungen und Randbedingungen

Die folgenden Vereinfachungen werden festgelegt und müssen erfüllt sein. Die mechanischen Kinematikdaten (wie z.B. Radius, Pushrod...) sind entsprechend zu wählen.

- An der oberen Polstelle (Y1) erreichen beide Kurbeln gleichzeitig ihre Totpunkte. An der oberen Polstelle bildet damit Radius und Pushrod, sowie Elbow2 und Elbow3 eine Gerade
- Die zweite Kurbel bewegt sich nur im Arbeitsbereich 1. Im Arbeitsbereich 1 bewegt sich die mechanische Position der y-Achse in positive Richtung wenn sich die zweite Kurbel im Uhrzeigersinn bewegt. Die erste Kurbel darf sich hingegen in beiden Arbeitsbereichen bewegen und auch endlos in eine Richtung drehen

8.3.3 MB_BellCrankData

Funktionsbeschreibung

Der Funktionsbaustein MB_BellCrankData berechnet die y-Positionen (Y0, Y1) und die Kurbelwinkel Phi (Phi0, Phi1) an der unteren und oberen Polstelle, sowie den maximalen Verfahrbereich (Sv) der y-Achse.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-32: Referenztable MB_BellCrankData

Schnittstellenbeschreibung

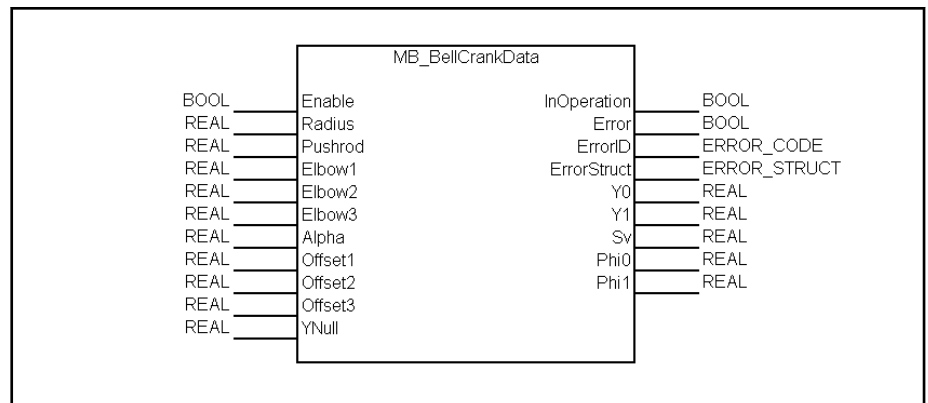


Abb.8-33: Funktionsbaustein MB_BellCrankData

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Berechnung der Daten in jedem Zyklus solange Enable=TRUE
	Radius	REAL	Länge der Kurbel1 in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
VAR_OUTPUT	InOperation	BOOL	Fehlerfreie Bearbeitung, Ausgangsdaten sind gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	Y0	REAL	Mechanische y-Position der unteren Polstelle [mm]
	Y1	REAL	Mechanische y-Position der oberen Polstelle [mm]
	Sv	REAL	Maximaler Verfahrbereich der y-Achse = ABS(Y1-Y0) [mm]
	Phi0	REAL	Winkel Phi der Kurbel 1 an der unteren Polstelle [°]
	Phi1	REAL	Winkel Phi der Kurbel 1 an der oberen Polstelle [°]

Abb.8-34: Schnittstellenvariablen Funktionsbaustein MB_BellCrankData

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle „F_RELATED_TABLE“, 16#0170, folgende Fehlermeldungen:

ML_TechCrank.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Abb.8-35: Fehlercodes Funktionsbaustein MB_BellCrankData

8.3.4 MB_CamTableBellCrank

Funktionsbeschreibung

Der Funktionsbaustein MB_CamTableBellCrank berechnet die notwendige Transformationskurvenscheibe (mit 1024 Stützstellen) für einen Kniehebelantrieb. Die Transformationskurvenscheibe dient zur Umrechnung von translatorischen Einheiten in den Kurbelwinkel Phi. Damit kann der rotatorische Kurbelantrieb über die Transformationskurvenscheibe an eine translatorisch gewichtete virtuelle Achse gekoppelt werden.

Die virtuelle Achse kann dann in translatorischen Einheiten bewegt werden, und der Kurbelantrieb folgt entsprechend der Transformationskurvenscheibe.

Um auch die Polstellen zu durchfahren, kann der Abstand zu den Polstellen (PoleDistance0 und PoleDistance1) definiert werden. Innerhalb der PoleDistance wird eine Ausgleichsfunktion (Ersatzpolynom 5ter Ordnung) verwendet, welche den Kniehebelantrieb approximiert und die resultierende Antriebsdynamik begrenzt. Bei dem Übergang in und aus dem linearisierten Bereich sind Position, Geschwindigkeit und Beschleunigung stetig.

Der FB stellt die berechnete Kurvenscheibe über den Ein/Ausgang "CamTable" dem SPS Programm zur Verfügung.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS ohne MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS ohne MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS ohne MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-36: Referenztable MB_CamTableBellCrank

Schnittstellenbeschreibung

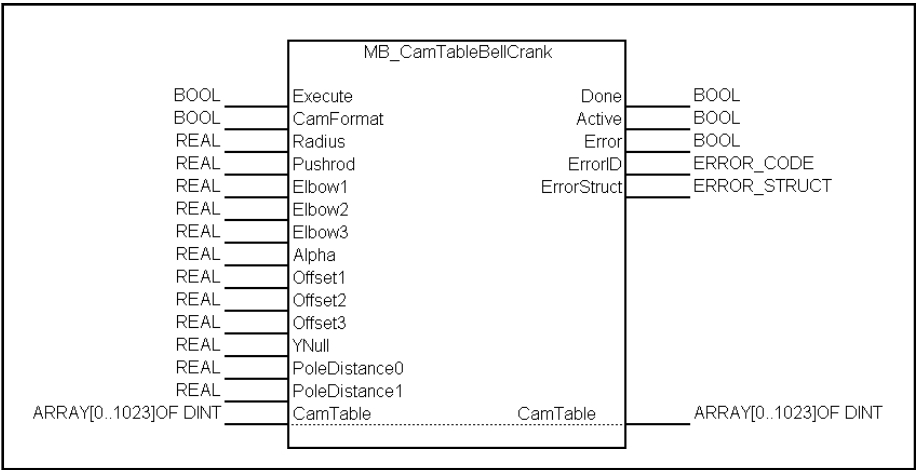


Abb.8-37: Funktionsbaustein MB_CamTableBellCrank

ML_TechCrank.library

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	CamTable	ARRAY [0..1023] OF DINT	Array mit den Daten der berechneten Transformations-Kurvenscheiben
VAR_INPUT	Execute	BOOL	Positive Flanke startet die Berechnung der Transformationskurvenscheibe
	CamFormat	BOOL	TRUE: Neues Kurvenscheiben Format -> letzter Tabellenpunkt entspricht erstem Tabellenpunkt
	Radius	REAL	Länge der Kurbel1 in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
	PoleDistance0	REAL	Entfernung von der unteren Polstelle Y0 (in [mm]), um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniehebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
	PoleDistance1	REAL	Entfernung von der oberen Polstelle Y1 (in [mm]), um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniehebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
VAR_OUTPUT	Done	BOOL	Berechnung abgeschlossen -> Kurvenscheibentabelle ist gültig
	Active	BOOL	FB ist in Bearbeitung
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_ CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_ STRUCT	Detaillierte Fehlerbeschreibung

Abb.8-38: Schnittstellenvariablen Funktionsbaustein MB_CamTableBellCrank

Notwendige Randbedingungen und Voraussetzungen

- An der oberen Polstelle (Y1) erreichen beide Kurbeln gleichzeitig ihre Totpunkte. An der oberen Polstelle bildet damit Radius und Pushrod, sowie Elbow2 und Elbow3 eine Gerade

- Die zweite Kurbel bewegt sich nur im Arbeitsbereich 1. Im Arbeitsbereich 1 bewegt sich die mechanische Position der y-Achse in positive Richtung wenn sich die zweite Kurbel im Uhrzeigersinn bewegt. Die erste Kurbel darf sich hingegen in beiden Arbeitsbereichen bewegen und auch endlos in eine Richtung drehen
- Die für die Kurvenscheibe (als Master) verwendete virtuelle Achse muss translatorisch, modulo gewichtet sein.
Der Modulowert soll hierbei wie folgt eingestellt werden: Modulowert = 2
• Verfahrbereich (Sv). Der Verfahrbereich (Sv) sei hierbei als Strecke zwischen der oberen (Y1) und unteren (Y0) Polstelle definiert.
Der Verfahrbereich Sv kann über den FB MB_BellCrankData ermittelt werden
- Der Kurbelantrieb muss rotatorisch, modulo 360° gewichtet sein

Anwendungsbeispiel von MB_CamTableBellCrank

Mit der folgenden Sequenz kann die translatorische Achse im Einzelachsbetrieb verfahren werden:

- Berechnung der Transformationskurvenscheibe durch den Baustein MB_CamTableBellCrank
- Berechnete Transformationskurvenscheibe auf den Kurbelantrieb laden (z.B. durch MB_WriteListParameter, falls diese nicht bereits geladen ist)
- Zuschaltung der Leistung (MC_Power)
- Herstellen des absoluten Massbezuges, falls kein Absolutwertgeber verwendet wird. Hierzu wird das Antriebskommando "Antriebsgeführtes Referenzieren" C0600 verwendet
- Virtuelle Achse auf die aus dem Kurbelwinkel resultierende translatorische Position positionieren. Die resultierende translatorische Position "Yvirt" kann aus dem Kurbelwinkel (Phi) mit dem FB MB_PhiToYvirt ermittelt werden. Die virtuelle Achse wird dann auf die ermittelte Position positioniert
- Zuschalten der Kurvenscheibenbetriebsart am Kurbelantrieb (z.B. über MC_CamIn, mit CamShaftDistance=360,1:1 Getriebe und Auswahl der zuvor berechneten und geladenen Transformationskurvenscheibe). Der Kurbelantrieb wird keine Aufsynchronisier-Bewegung ausführen, da die virtuelle Achse bereits an der "Synchronposition" steht
- Die virtuelle Achse kann jetzt über die Bausteine MC_MoveAbsolute und MC_MoveVelocity oder AxisInterface im Einzelachsbetrieb verfahren werden (die Kurbelachse folgt dann der Transformationskurvenscheibe). Die virtuelle Achse kann auch in einer Synchronbetriebsart betrieben werden



Der Kurbelantrieb führt eine dynamische Aufsynchronisierbewegung bei Zuschalten der Kurvenscheibenbetriebsart aus, wenn die virtuelle Achse vorher nicht auf die Kurbelposition positioniert wurde.

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0006	PoleDistance0 <= 0

ML_TechCrank.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0007	PoleDistance1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0008	PoleDistance0+PoleDistance1 > Verfahrbereich
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Abb.8-39: Fehlercodes Funktionsbaustein MB_CamTableBellCrank

8.3.5 MB_CamTableBellCrankSuperimposed

Funktionsbeschreibung

Dieser Baustein überlagert die vorgegebene Anwenderkurvenscheibe (CamInput) mit der Transformationskurvenscheibe und liefert das Resultat über den Ein/Ausgang "CamOutput". Die Transformationskurvenscheibe wird innerhalb des FB nach den gleichen Gesetzmässigkeiten, wie im FB "MB_CamTableBellCrank" berechnet. Das Prinzip der Überlagerung ist in [Abb. 8-40 "Prinzip der Kurvenscheiben Überlagerung" auf Seite 413](#) dargestellt. Die Anwenderkurvenscheibe muss 1024 Stützpunkte enthalten und definiert hierbei die Bewegung der translatorischen Achse im Bezug zur Leitachse (ohne Berücksichtigung der Kniehebelkinematik). Der Tabellenwert 100% entspricht dabei der Bewegung $2 \cdot \text{Verfahrbereich} = \text{Modulowert der virtuellen translatorischen Achse}$.

Die berechnete überlagerte Kurvenscheibe muss zur Nutzung von dem SPS Programm auf den Antrieb geladen werden, und über die Kurvenscheiben Betriebsart genutzt werden.



Je nach vorgegebenem Endpunkt der Anwenderkurvenscheibe werden folgende Fälle unterschieden:

- Ist der Endpunkt der Anwenderkurvenscheibe nahe 100%, so führt die Kurbel 1 keine Richtungsumkehr aus (Kurbel 1 "dreht durch") -> Energieoptimales Verfahren, da natürliche Bewegung der Kurbel genutzt wird
- Ist der Endpunkt der Anwenderkurvenscheibe 0%, entsteht eine Vor-Rück-Bewegung der translatorischen Achse mit Richtungsumkehr der Kurbel 1 (wie in [Abb. 8-40 "Prinzip der Kurvenscheiben Überlagerung" auf Seite 413](#) dargestellt)

Weiterführender Hinweis

Ab MLC03VRS unterstützt die verwendete virtuelle Achse Synchron-Betriebsarten. Dadurch ist das Überlagern der Anwenderkurvenscheibe mit der Transformationskurvenscheibe typischerweise für MLC03 Anwendungen nur

dann notwendig, wenn eine zusätzliche virtuelle Achse eingespart werden soll.

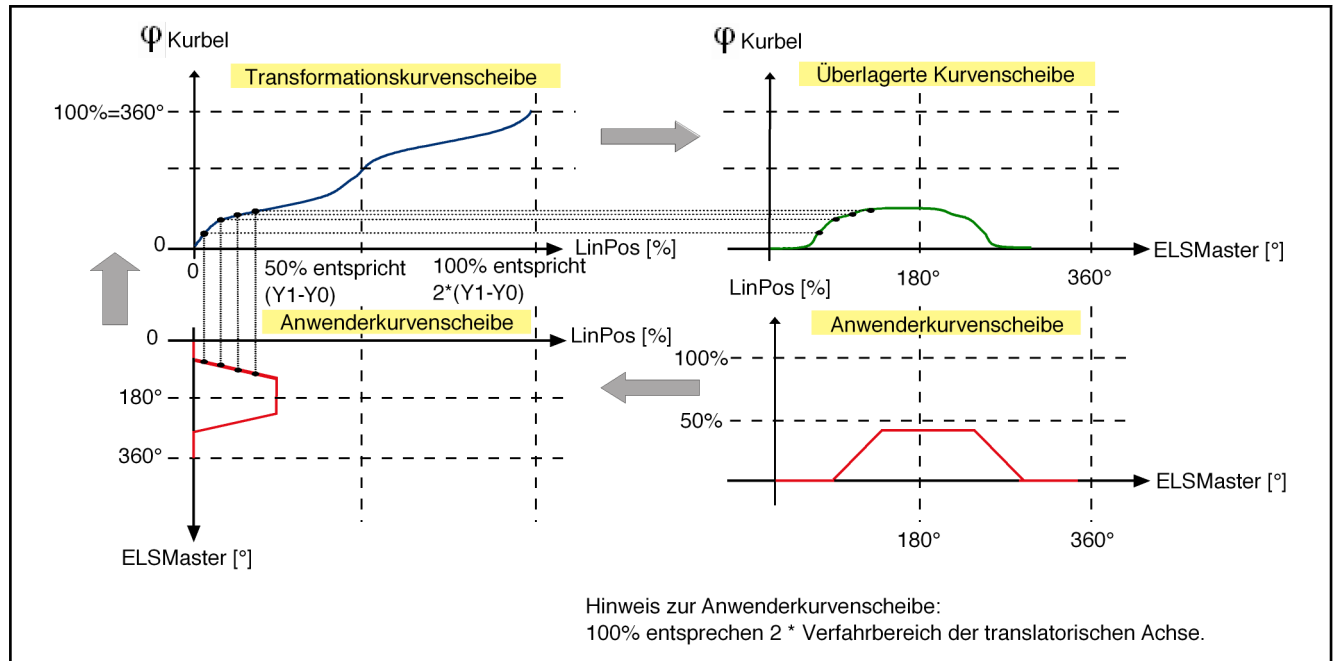


Abb.8-40: Prinzip der Kurvenscheiben Überlagerung

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-41: Referenztable MB_CamTableBellCrankSuperimposed

Schnittstellenbeschreibung

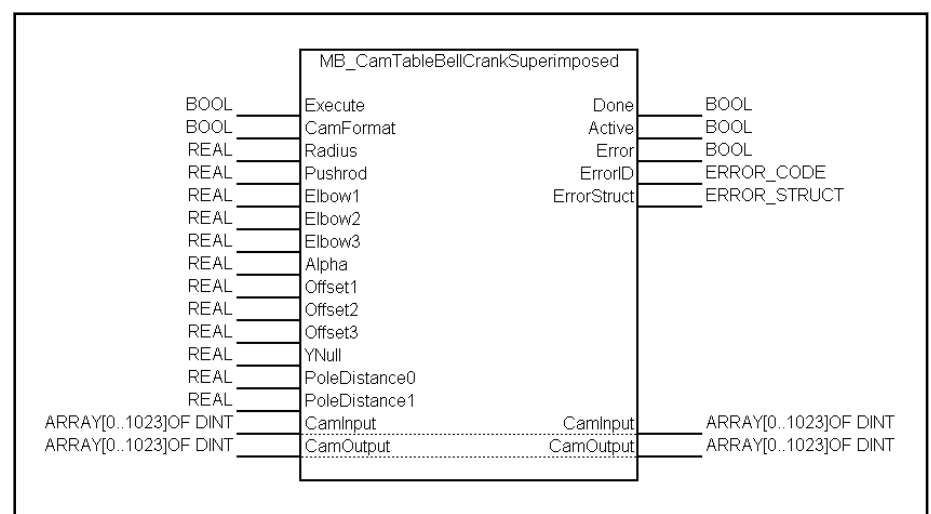


Abb.8-42: Funktionsbaustein MB_CamTableBellCrankSuperimposed

ML_TechCrank.library

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	CamInput	ARRAY [0..1023]OF DINT	Array mit den Daten der Anwenderkurvenscheibe zur Vorgabe der translatorischen Bewegung
	CamOutput	ARRAY [0..1023]OF DINT	Array mit den Daten der berechneten überlagerten Kurvenscheibe zur Ansteuerung der Kurbelachse
VAR_INPUT	Execute	BOOL	Positive Flanke startet die Berechnung der überlagerten Kurvenscheibe
	CamFormat	BOOL	TRUE: Neues Kurvenscheiben Format -> letzter Tabellenpunkt entspricht erstem Tabellenpunkt
	Radius	REAL	Länge der Kurbel1 in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
	PoleDistance0	REAL	Entfernung von der unteren Polstelle Y0 (in [mm]), um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
	PoleDistance1	REAL	Entfernung von der oberen Polstelle Y1 (in [mm]), um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
VAR_OUTPUT	Done	BOOL	Berechnung abgeschlossen, Ausgangs-Kurvenscheibentabelle ist gültig
	Active	BOOL	FB ist in Bearbeitung
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_STRUCT	Detaillierte Fehlerbeschreibung

Abb.8-43: Schnittstellenvariablen Funktionsbaustein MB_CamTableBellCrank-Superimposed



zur Anwendung der überlagerten Kurvenscheibe

Soll die translatorische Achse auf die überlagerte Kurvenscheibe synchronisiert werden, ist wie folgt vorzugehen:

- Anwenderkurvenscheibe wird über den Ein/Ausgang "CamInput" dem FB zur Verfügung gestellt (z.B. durch Auslesen einer Antriebskurvenscheibe, oder Auslesen eines Files...)
- Die Berechnung der überlagerten Kurvenscheibe wird durch den Baustein MB_CamTableBellCrankSuperimposed ausgeführt
- Nach der Kurvenscheiben-Überlagerung lädt das Anwenderprogramm die überlagerte Kurvenscheibe "CamOutput" auf den Antrieb oder die Steuerung (MB_WriteListParameter)
- Zuschaltung der Leistung (MC_Power) am Kurbelantrieb
- Herstellen des absoluten Massbezuges der Kurbelachse, falls kein Absolutwertgeber verwendet wird. Hierzu muss das Antriebskommando C0600 "Antriebsgeführtes Referenzieren" verwendet werden
- Die aktuelle Kurbelposition und der Lagesollwert aus der überlagerten Kurvenscheibe können auseinander liegen. Ein Umschalten der Kurvenscheibenbetriebsart (z.B. über MC_CamIn, mit CamShaftDistance= 360,1:1 Getriebe und Auswahl der überlagerten Kurvenscheibe) kann damit über die folgenden 2 Optionen durchgeführt werden:
 - Umschalten der Kurvenscheibenbetriebsart am Kurbelantrieb ohne vorherigen Positions-Abgleich -> Der Kurbelantrieb wird ein dynamisches Aufsynchronisieren ausführen
 - Kurbelachse wird vor Umschalten der Kurvenscheibenbetriebsart auf die "Synchronposition" positioniert. Hierzu kann über die Funktion "MB_MasterToPhi" der Kurbelwinkel aus der überlagerten Kurvenscheibe ermittelt werden. Vor Umschalten der Synchronbetriebsart kann dann der Kurbelantrieb auf diese Position positioniert werden. Im Anschluss kann dann die überlagerte Kurvenscheibe ohne Aufsynchronisierungsbewegung aktiviert werden



zur Wichtung der Achsen

- Die verwendete Leitachse muss Modulo gewichtet werden (translatorisch oder rotatorisch)
- Die Achse zum Antrieb der Kurbel muss rotatorisch, modulo 360° gewichtet sein

Notwendige Randbedingungen

Siehe ["Notwendige Randbedingungen und Voraussetzungen"](#) auf Seite 410 von MB_CamTableBellCrank.

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, folgende Fehlermeldungen:

ML_TechCrank.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0006	PoleDistance0 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0007	PoleDistance1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0008	PoleDistance0+PoleDistance1 > Verfahrbereich
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Abb.8-44: Fehlercodes Funktionsbaustein MB_CamTableBellCrankSuperimposed

8.3.6 MB_PhiToYvirt

Funktionsbeschreibung

Der Funktionsbaustein MB_PhiToYvirt wandelt den Kurbelwinkel (Phi) in die mechanische - und virtuelle translatorische Position (Ymech, Yvirt) um.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-45: Referenztable MB_PhiToYvirt

Schnittstellenbeschreibung

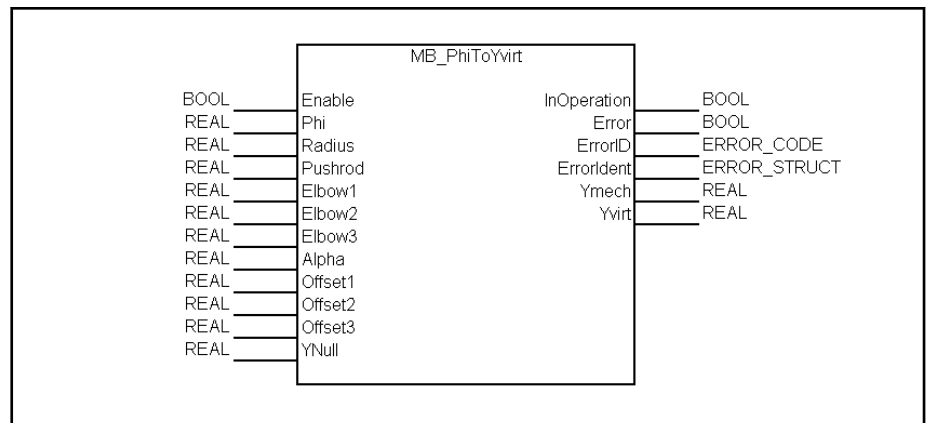


Abb.8-46: Funktionsbaustein MB_PhiToYvirt

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Berechnung von Yvirt in jedem Zyklus solange Enable=TRUE
	Phi	REAL	Kurbelposition in [°]
	Radius	REAL	Länge der Kurbel in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
VAR_OUTPUT	InOperation	BOOL	Berechnung ist fertig -> Yvirt ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	Ymech	REAL	Mechanische translatorische Position [mm]
	Yvirt	REAL	Virtuelle translatorische Position [mm]

Abb.8-47: Schnittstellenvariablen Funktionsbaustein MB_PhiToYvirt

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, folgende Fehlermeldungen:

ML_TechCrank.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Abb.8-48: Fehlercodes Funktionsbaustein MB_PhiToYvirt

8.3.7 MB_YvirtToPhi

Funktionsbeschreibung

Der Funktionsbaustein MB_YvirtToPhi wandelt die virtuelle translatorische Position (Yvirt) in den Kurbelwinkel (Phi) um.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-49: Referenztabelle MB_YvirtToPhi

Schnittstellenbeschreibung

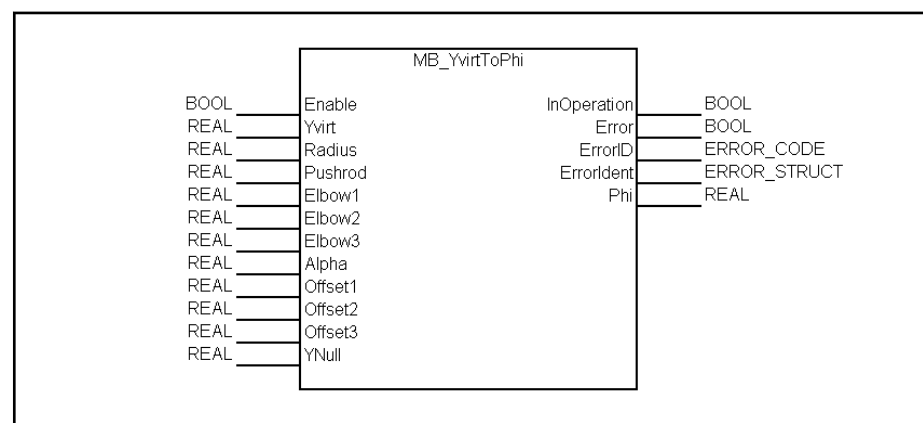


Abb.8-50: Funktionsbaustein MB_YvirtToPhi

E/A-Typ	Name	Typ	Beschreibung
VAR_INPUT	Enable	BOOL	Berechnung von Phi in jedem Zyklus solange Enable=TRUE
	Yvirt	REAL	Virtuelle translatorische Position [mm]
	Radius	REAL	Länge der Kurbel in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
VAR_OUTPUT	InOperation	BOOL	Berechnung ist fertig -> Phi ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	Phi	REAL	Kurbelwinkel in [°]

Abb.8-51: Schnittstellenvariablen Funktionsbaustein MB_YvirtToPhi

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)

ML_TechCrank.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Abb.8-52: Fehlercodes Funktionsbaustein MB_YvirtToPhi

8.3.8 MB_YvirtToPhiPoly5

Funktionsbeschreibung Der Funktionsbaustein MB_YvirtToPhiPoly5 wandelt die virtuelle translatorische Position (Yvirt) in den Kurbelwinkel (Phi) um. Innerhalb der PoleDistance wird eine Ausgleichsfunktion (Ersatzpolynom 5ter Ordnung) verwendet, welche den Kniehebelantrieb approximiert und die resultierende Antriebsdynamik begrenzt. Bei dem Übergang in und aus dem linearisierten Bereich sind Position, Geschwindigkeit und Beschleunigung an der Kurbel 1 stetig.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-53: Referenztabelle MB_YvirtToPhiPoly5

Schnittstellenbeschreibung

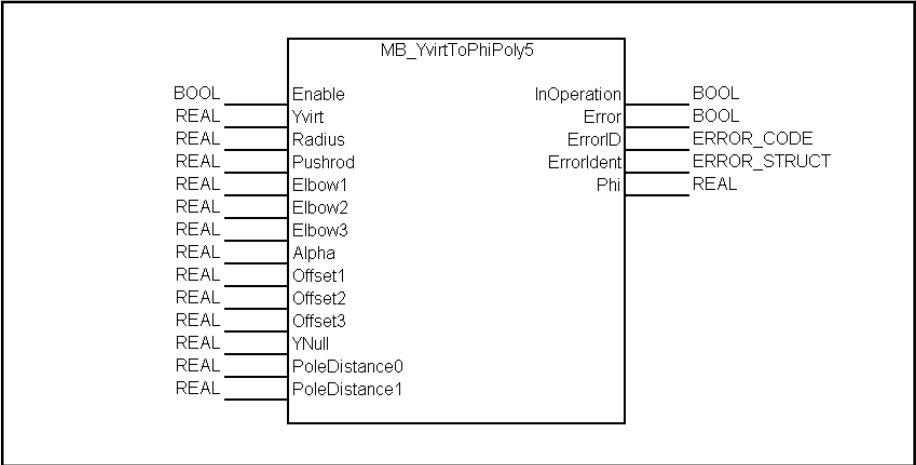


Abb.8-54: Funktionsbaustein MB_YvirtToPhiPoly5

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Berechnung von Phi in jedem Zyklus solange Enable=TRUE
	Yvirt	REAL	Virtuelle translatorische Position [mm]
	Radius	REAL	Länge der Kurbel in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
	PoleDistance0	REAL	Entfernung von der unteren Polstelle Y0 (in [mm]) um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniehebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
	PoleDistance1	REAL	Entfernung von der oberen Polstelle Y1 (in [mm]) um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniehebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
VAR_OUTPUT	InOperation	BOOL	Berechnung ist fertig -> Phi ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	Phi	REAL	Kurbelwinkel in [°]

Abb.8-55: Schnittstellenvariablen Funktionsbaustein MB_YvirtToPhiPoly5

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0006	PoleDistance0 <= 0

ML_TechCrank.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0007	PoleDistance1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0008	PoleDistance0+PoleDistance1 > Verfahrbereich
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Abb.8-56: Fehlercodes Funktionsbaustein MB_YvirtToPhiPoly5

8.3.9 MB_YvirtToYmech

Funktionsbeschreibung

Der Funktionsbaustein MB_YvirtToYmech wandelt die virtuelle translatorische Position (Yvirt) in die mechanische translatorische Position (Ymech) um. Ymech kann z.B. für Anzeigezwecke genutzt werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-57: Referenztabelle MB_YvirtToYmech

Schnittstellenbeschreibung

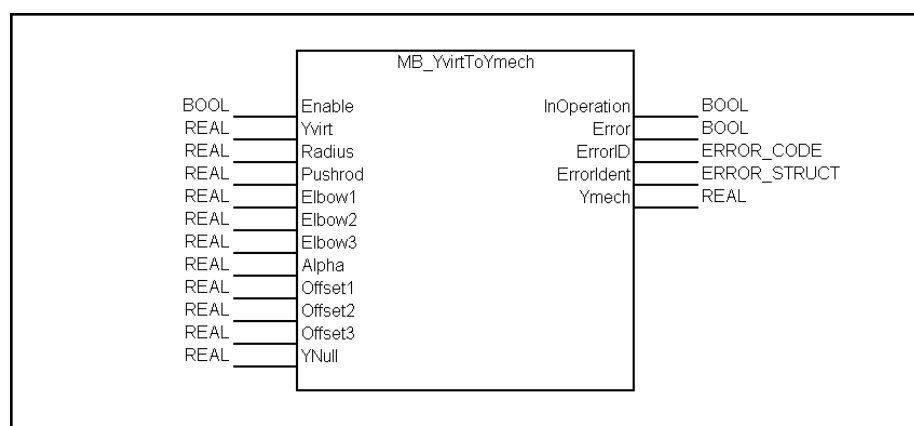


Abb.8-58: Funktionsbaustein MB_YvirtToYmech

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	FB führt Berechnung aus, solange Enable=TRUE ist
	Yvirt	REAL	Virtuelle Translatorische Position in [mm]
	Radius	REAL	Länge der Kurbel in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
VAR_OUTPUT	InOperation	BOOL	Ausgangswert (Ymech) ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	Ymech	REAL	Mechanische translatorische Position (Ymech) [mm]

Abb.8-59: Schnittstellenvariablen Funktionsbaustein MB_YvirtToYmech

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)

ML_TechCrank.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Abb.8-60: Fehlercodes Funktionsbaustein MB_YvirtToYmech

8.3.10 MB_YVirtToWorkRange

Funktionsbeschreibung

Die Funktion MB_YVirtToWorkRange liefert als Rückgabewert den Arbeitsbereich (1 oder 2) der Kurbel 1, welche sich aus der virtuellen translatorischen Position ergibt. Im Arbeitsbereich 1 bewegt sich die mechanische Position der y-Achse in positive Richtung wenn sich die Kurbel 1 im Uhrzeigersinn bewegt. Im Arbeitsbereich 2 bewegt sich die mechanische Position der y-Achse in negative Richtung wenn sich die Kurbel 1 im Uhrzeigersinn bewegt. Diese Funktion wird intern von anderen Funktionsbausteinen verwendet.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechCrank.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology_07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb.8-61: Referenztabelle MB_YVirtToWorkRange

Schnittstellenbeschreibung

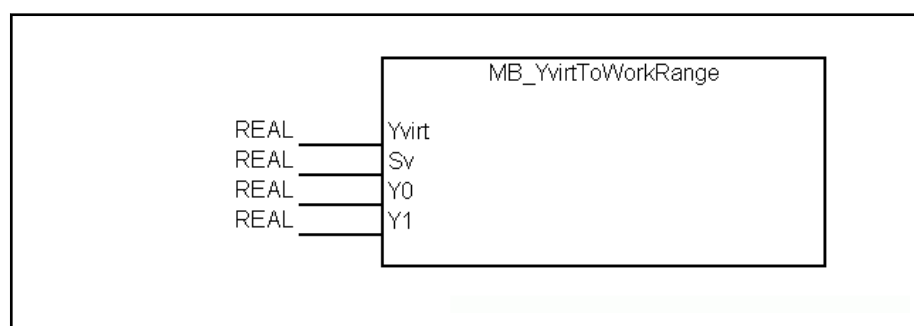


Abb.8-62: Funktion MB_YVirtToWorkRange

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Yvirt	REAL	Virtuelle Translatorische Position [mm]
	Sv	REAL	Maximaler Verfahrbereich der y-Achse = ABS(Y1-Y0) [mm]
	Y0	REAL	Mechanische y-Position der unteren Polstelle [mm]
	Y1	REAL	Mechanische y-Position der oberen Polstelle [mm]

Abb.8-63: Schnittstellenvariablen Funktion MB_YVirtToWorkRange

Fehlerbehandlung

Die Funktion generiert keine Fehler.

9 RMB_TechWinder.library/MX_TechWinder.lib

9.1 Einführung und Übersicht

In den Bibliotheken RMB_TechWinder.compiled-library (IndraMotion MLC) und MX_TechWinder.lib (IndraMotion MLD) sind verschiedene Funktionsbausteine zusammengefasst, die Funktionalitäten für das Auf- und Abwickeln von Stoffbahnen (Wicklern) bereitstellen.

Mit Hilfe dieser Funktionsbausteine können die Funktionen eines

- Tänzerlagereglers
- Wickelrechners mit Tänzer
- Wickelrechners mit Zugkraftregelung und
- eines Wickelrechners mit Zugkraftsteuerung

umgesetzt werden.

Im Folgenden werden die jeweils zur Umsetzung dieser Funktionalitäten erforderlichen Funktionsbausteine beschrieben.

In den folgenden Tabellen wird eine Übersicht über alle dokumentierten Funktionsbausteine und ihre jeweilige Anwendung gegeben.

Vorab verschaltete komplette Wickler-Funktionsbausteine

Funktionsbaustein	Beschreibung
MB_WinderDancerCtrlType01	Funktionsbaustein für Wickler mit Tänzer beinhaltet unter anderem: MB_DancerControlType03 und MB_DiameterCalculatorType03 MB_DiameterMeasurementType01
MB_WinderTensionCtrlType02	Funktionsbaustein für Wickler mit Zugkraftregelung bzw. Zugkraftsteuerung (sensorlos) beinhaltet unter anderem: MB_DiameterCalculatorType03 MB_DiameterMeasurementType01
MB_WinderTensionCtrlSpeed-Type01	Funktionsbaustein für Wickler mit Zugkraftregelung und Drehzahlkorrektur beinhaltet unter anderem: MB_DiameterCalculatorType03 MB_DiameterMeasurementType01

Abb.9-1: Vorab verschaltete Wickler-FB



Die vorab verschalteten Funktionsbausteine beinhalten alle Funktionen der einzelnen Bausteine und nehmen dem Anwender die richtige Zusammenschaltung der einzelnen Funktionsbausteine ab.

Funktionsbausteine für einzelne Funktionalitäten

RMB_TechWinder.library/MX_TechWinder.lib

Funktionsbaustein	Beschreibung
MB_DancerControlType03	Funktionsbaustein zur Berechnung einer Tänzerlageregelung für eine Zugachse mit Tänzer bzw. Wickelachse
MB_DiameterCalculatorType03	Funktionsbaustein zur Berechnung des aktuellen Durchmessers für Wickelachsen mit Geschwindigkeitsnachführung
MB_DiameterMeasurementType01	Funktionsbaustein zum Einlesen eines gemessenen Durchmessers für Wickelachsen mit Geschwindigkeitsnachführung

Abb.9-2: Einzelfunktionsbausteine

Funktionsbausteine für zusätzliche Wicklerfunktionen

Funktionsbaustein	Beschreibung
MB_CalcInertiaLimitType02	Funktionsbaustein zur Berechnung des maximalen Trägheitsmomentes des Wickelgutes auf einem Wickel
MB_UnwindMaterialType02	Funktionsbaustein zur Berechnung der Restlänge, Restlaufzeit, ... der Stoffbahn für einen Abwickler
MB_WindTaperProfileType01	Funktionsbaustein zur Berechnung eines Wickelhärteprofils für einen Aufwickler mit Zugspannungsregelung bzw. -steuerung
MB_WindSpeedControlAdaptionType01	Funktionsbaustein zur automatischen Anpassung der K_p -Verstärkung des Drehzahlreglers

Abb.9-3: Funktionsbausteine für zusätzliche Funktionalitäten

Bibliotheksabhängigkeiten**Steuerung IndraMotion MLC - RMB_TechWinder**

Die benötigten Bibliotheken werden automatisch mit eingebunden.

Steuerung IndraMotion MLD - MX_TechWinder

Folgende Bibliotheken werden benötigt:

- MX_Technology0x (MPX0x Antriebsfirmware) bzw.
- RIL_LoopControl

Funktionspakete für IndraMotion MLD

Für IndraMotion MLD wird für die Bausteine der MX_TechWinder Bibliothek das Antriebsfunktionspaket MA "IndraMotion MLD Advanced" benötigt.

Dieses Funktionspaket ist mit der entsprechenden Antriebsfirmware separat zu bestellen.

9.2 Funktionsbaustein zur Tänzerlageregelung

9.2.1 Einführung und Übersicht

Der Tänzerlageregler wird z.B. an Zugachsen oder Wicklern mit Tänzern eingesetzt. Der Tänzerlageregler regelt die Lage eines Tänzers. Die Zugspannung wird durch eine Kraft vorgegeben, die am frei beweglichen Teil des Tänzers angreift. Diese Kraft kann durch eine Gewichtskraft, eine Feder oder durch Pneumatik aufgebracht werden. Der Istwert der Tänzerlage wird z.B. über einen Analogkanal erfasst, an die Steuerung übergeben und im Prozessregler verarbeitet. Die folgende Abbildung zeigt das Prinzip einer Tänzerlageregelung.

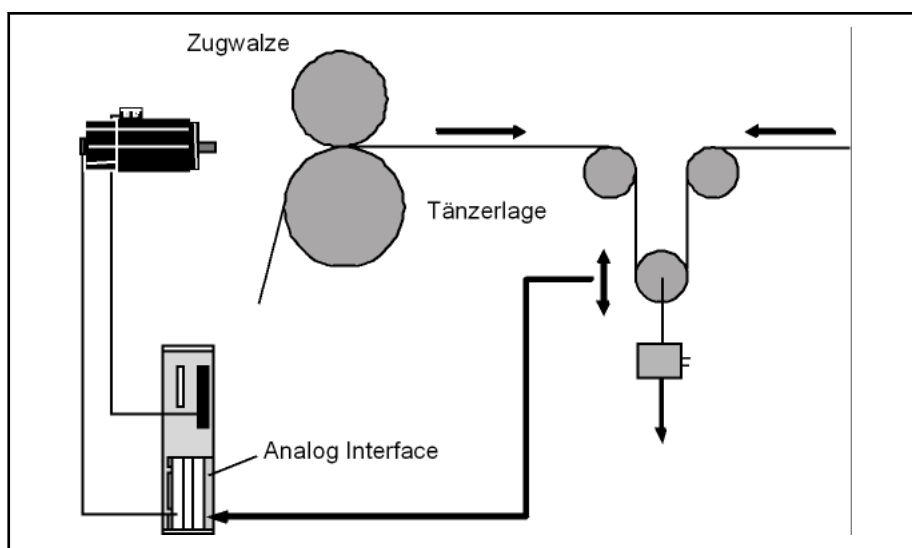


Abb.9-4: Prinzip der Tänzerlageregelung

Der Tänzerlageregler ist im Funktionsbaustein MB_DancerControlType03 implementiert und als PID-Regler ausgeführt. Durch Setzen der Reglerparameter können die einzelnen Reglerfunktionen (P-Anteil: Proportionalanteil, I-Anteil: Integralanteil, D-Anteil: differenzieller Anteil) angesteuert werden. Wird ein Parameter zu Null gesetzt, ist der entsprechende Regleranteil ebenfalls nicht wirksam.

Nachfolgende Tabelle zeigt die möglichen und erlaubten Reglerkonfigurationen.

Nr.	P-Anteil	I-Anteil	D-Anteil	erlaubt	Bemerkung
1	0	0	0	ja	Regler unwirksam, Ausgang = 0.0
2	0	0	1	nein	D-Regler nicht definiert, Fehler
3	0	1	0	ja	reiner I-Regler
4	0	1	1	nein	DI-Regler nicht definiert, Fehler
5	1	0	0	ja	reiner P-Regler
6	1	0	1	ja	PD-Regler
7	1	1	0	ja	PI-Regler
8	1	1	1	ja	PID-Regler

Abb.9-5: Erlaubte und nicht erlaubte Reglerkonfigurationen (1 entspricht Anteil gesetzt (Wert ≥ 0.0), 0 entspricht Anteil nicht gesetzt (Wert = 0.0))

Funktionsdiagramm des Tänzerlagereglers

Anhand des folgenden Funktionsdiagramms wird die Funktionsweise des Tänzerlagereglers schematisch erläutert.

RMB_TechWinder.library/MX_TechWinder.lib

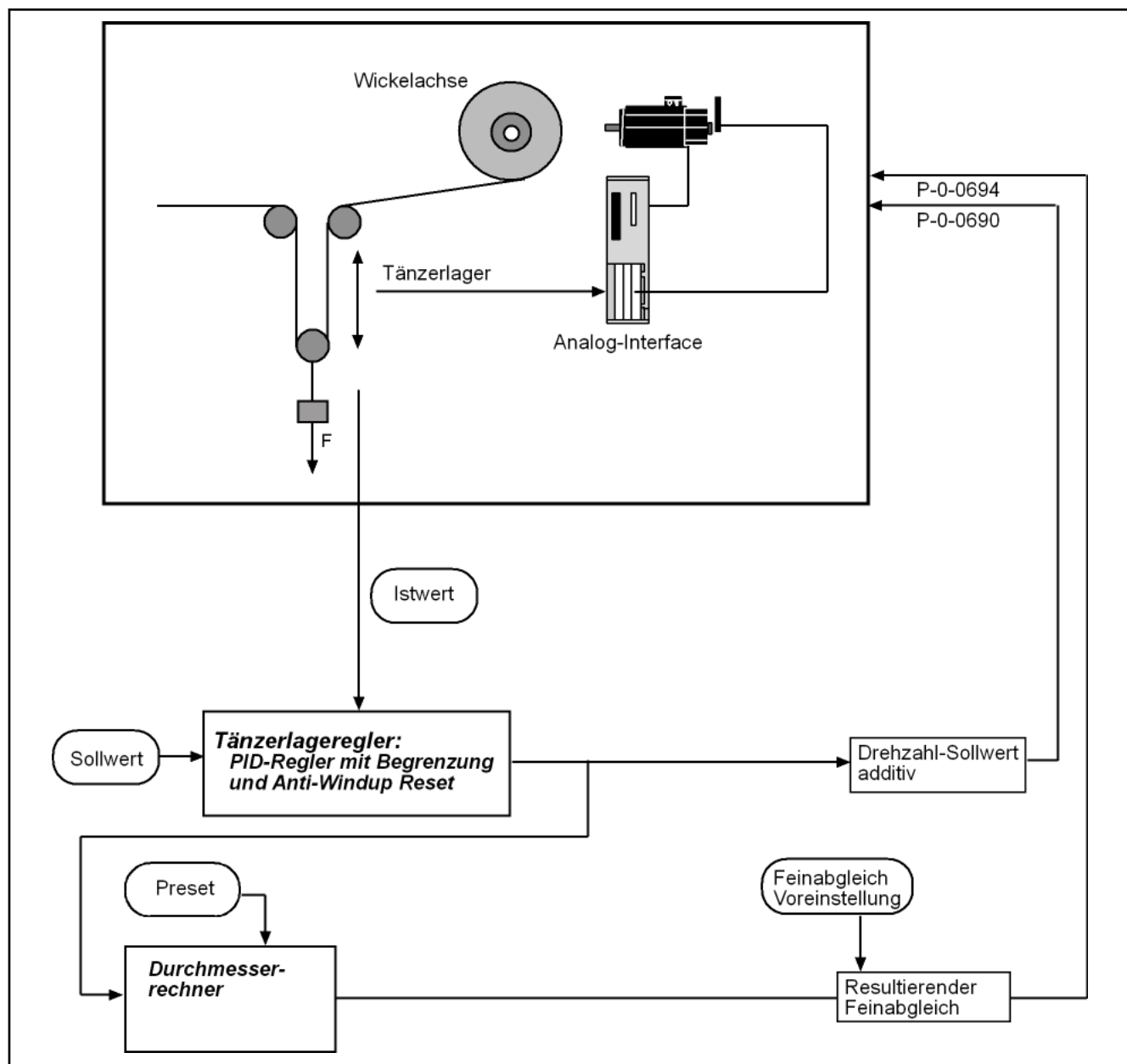


Abb.9-6: Funktionsdiagramm des Tänzerlageregers

9.2.2 MB_DancerControlType03

Kurzbeschreibung

Der Funktionsbaustein berechnet zyklisch eine Tänzerlageregelung für eine Zugwalze mit Tänzer oder für eine Wickelachse mit Tänzer. Zyklische Ausgangsgröße des Tänzerlageregers ist der Drehzahlversatz für den jeweiligen Antrieb einer Zugwalze oder Wickelachse. Des Weiteren wird die aktuelle Regelabweichung des Tänzerlageregers ausgegeben.

Der Ausgang "AddVelocityDancer" des Funktionsbausteins wirkt auf den Parameter P-0-0690 "Geschwindigkeits-Sollwert additiv, Prozessregler". Ist dies nicht gewünscht, kann dies mit dem Eingang "DisableCyclicWrites" verhindert werden. Es wird dann nur auf den Ausgang "AddVelocityDancer" geschrieben.

RMB_TechWinder.library/MX_TechWinder.lib

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechWinder.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechWinder.lib

Schnittstellenbeschreibung

Abb.9-7: Referenztabelle MB_DancerControlType03

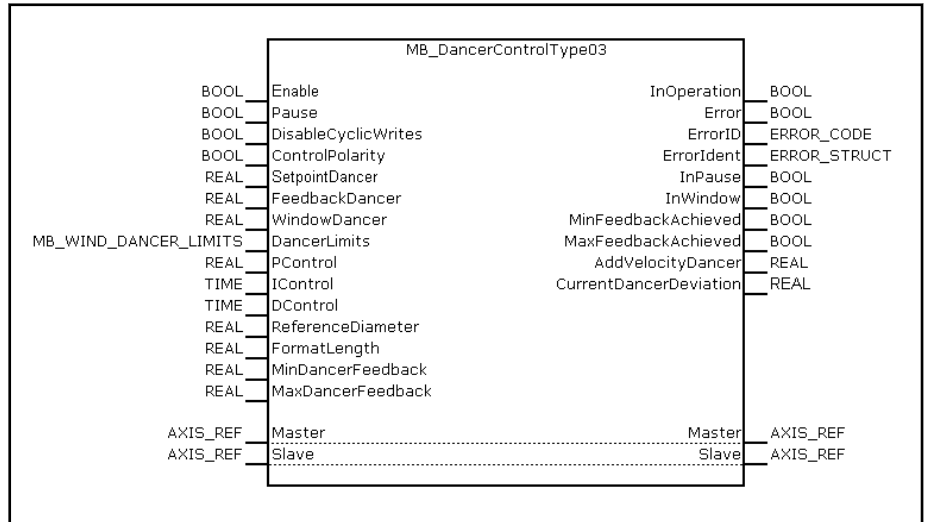


Abb.9-8: Funktionsbaustein MB_DancerControlType03

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (Zugachse)
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	Pause	BOOL	Tänzerregler pausiert, keine Geschwindigkeitsanpassung (pegelgesteuert)
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	ControlPolarity	BOOL	Polarität der Tänzerregelung wird invertiert (Übernahme bei pos. Flanke von Enable)
	SetpointDancer	REAL	Sollwert für die Tänzerregelung
	FeedbackDancer	REAL	Istwert der Tänzerlage
	WindowDancer	REAL	Fenster zur Überwachung des Tänzerlageistwertes [%], bezogen auf den Tänzerlagesollwert. Entfernt sich der Tänzer vom Lagesollwert, um mehr als den angegebenen Wert, meldet dies der Baustein durch Rücksetzen des Ausgangs "InWindow"
	DancerLimits	MB_WIND_DANCER_LIMITS	Struktur für die automatische Stellgrößenbegrenzung des Tänzers
	PControl	REAL	Wert für die Kp-Verstärkung der Tänzerregelung
	IControl	TIME	Wert für die Nachstellzeit der Tänzerregelung

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Typ	Kommentar
	DControl	TIME	Wert für die Vorhaltezeit der Tänzerregelung
	ReferenceDiameter	REAL	Bezugsdurchmesser, (Wert dient zur Normierung von PControl und der Stellgrößenbegrenzung DancerLimits, d.h. diese Werte werden intern angepasst) Bei reinem Tänzerlageregler: Durchmesser der Zugwalze [mm] Bei Wickler mit Tänzer: Aktueller Wickeldurchmesser [mm] vom Durchmesserrechner Baustein zuweisen
	FormatLength	REAL	Formatlänge [mm], (Wert dient zur Normierung von PControl) (1 Umdrehung der Referenzachse abgewickelt)
	MinDancerFeedback	REAL	Untere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
	MaxDancerFeedback	REAL	Obere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	InPause	BOOL	Tänzerregler pausiert
	InWindow	BOOL	Tänzeristwert innerhalb des Tänzerfensters
	MinFeedbackAchieved	BOOL	Istwert der Tänzerlage ist unterhalb des Eingang MinDancerFeedback
	MaxFeedbackAchieved	BOOL	Istwert der Tänzerlage ist oberhalb des Eingang MaxDancerFeedback
	AddVelocityDancer	REAL	Drehzahloffset für den Wickel- oder Zugwalzenantrieb [U/min]
	CurrentDancerDeviation	REAL	Aktuelle Regelabweichung des Tänzerreglers

Abb.9-9: Schnittstellenvariablen Funktionsbaustein MB_DancerControlType03

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Pause	BOOL			FALSE	kontinuierlich
DisableCyclicWrites	BOOL			FALSE	steigende Flanke an Enable
ControlPolarity	BOOL			FALSE	steigende Flanke an Enable
SetpointDancer	REAL	0.0	n. def.	0.0	kontinuierlich
FeedbackDancer	REAL	n. def.	n. def.	0.0	kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
WindowDancer	REAL	0.0	n. def.	0.0	kontinuierlich
DancerLimits	STRUCT				
PControl	REAL	0.0	n. def.	0.0	kontinuierlich
IControl	TIME	0s	n. def.	0s	kontinuierlich
DControl	TIME	0s	n. def.	0s	kontinuierlich
ReferenceDiameter	REAL	0.0	n. def.	100	kontinuierlich
FormatLength	REAL	0.0	n. def.	0.0	steigende Flanke an Enable
MinDancerFeedback	REAL	0.0	MaxDancerFeedback	0.0	kontinuierlich
MaxDancerFeedback	REAL	MinDancerFeedback	n. def.	0.0	kontinuierlich

Abb.9-10: Min-/Max- und Default-Werte der Eingänge MB_DancerControlType03

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
LowLimitControl	REAL	0.0	n. def.	0.0	kontinuierlich	untere Begrenzung des Reglerausganges (bei Drehzahl Null) bipolar [U/min] (Der Wert wird intern mit dem Durchmesserverhältnis von "ReferenceDiameter" und "RepeatLength" normiert)
HighLimitControl	REAL	0.0	n. def.	0.0	kontinuierlich	obere Begrenzung des Reglerausganges (bei Drehzahl VelocityHighLimit) bipolar [U/min] (der Wert wird intern mit dem Durchmesserverhältnis von "ReferenceDiameter" und "RepeatLength" normiert)
VelocityHighLimit	REAL	0.0	n. def.	0.0	kontinuierlich	Drehzahl des Masters für HighLimitControl [U/min]

Abb.9-11: Schnittstellenbeschreibung Datenstruktur MB_WIND_DANCER_LIMITS

Die in der Tabelle zusammengefasste Datenstruktur MB_WIND_DANCER_LIMITS beschreibt den im Diagramm dargestellten Begrenzer der Reglerstellgröße.

RMB_TechWinder.library/MX_TechWinder.lib

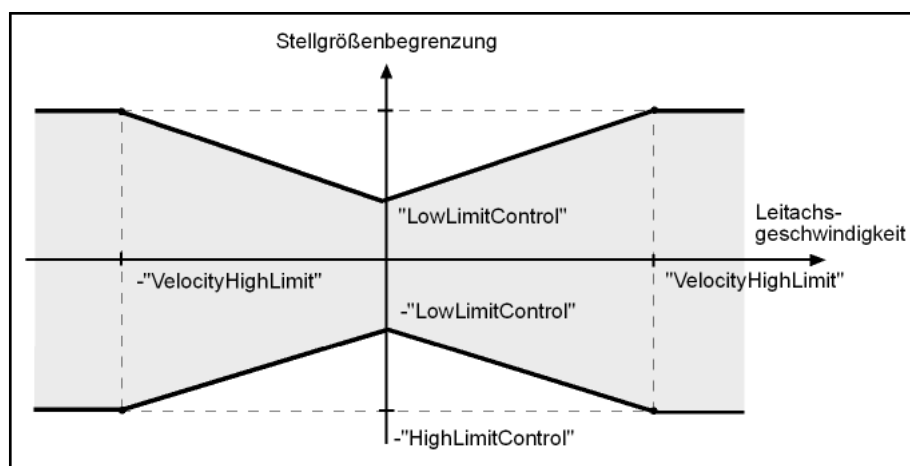


Abb.9-12: Stellgrößenbegrenzung des Tänzerlagereglers

Stellgrößenbegrenzung

Die Stellgrößenbegrenzung des Tänzerlagereglers wird in Abhängigkeit der Referenzachsgeschwindigkeit und dem Durchmesser Verhältnis ("Formatlänge" / π * "ReferenceDiameter") dynamisch angepasst. D.h. bei höherer Geschwindigkeit ist die Begrenzung der additiven Geschwindigkeit vom Tänzerlageregler automatisch größer als bei niedriger Geschwindigkeit. Die Berücksichtigung des Durchmesser Verhältnisses führt dazu, dass die Begrenzung bei Wickelanwendungen automatisch einem sich ändernden "ReferenceDiameter" anpasst wird.

Signal-Zeit-Diagramm

Im folgenden Signal-Zeit-Diagramm sind ausgewählte Signale des Funktionsbausteins dargestellt. Mit dem Eingang "Enable", der am Ausgang "InOperation" signalisiert wird, wird der Baustein in Betrieb genommen. Die Betriebsbereitschaft kann durch einen Fehler, der am Ausgang "Error" signalisiert wird, unterbrochen werden. Der interne Reglerprozess kann mit dem Eingang "Pause" angehalten werden. "Pause" wird am Ausgang "InPause" signalisiert.

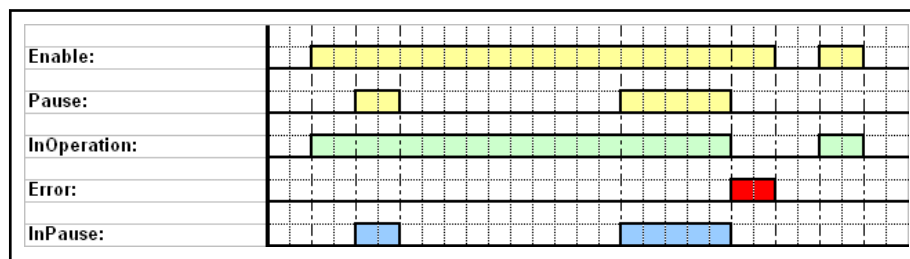


Abb.9-13: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_DancerControlType02

Funktionsbeschreibung

Der Funktionsbaustein berechnet zyklisch eine Tänzerlageregelung für eine Zugwalze mit Tänzer oder für eine Wickelachse mit Tänzer. Es wird der Funktionsbaustein IL_PIDType01 aus der Bibliothek RIL_LoopControl als Regler verwendet. Der Regler ist als PID-Regler ausgeführt. Zyklische Ausgangsgrößen des Tänzerlagereglers sind der Drehzahlversatz des jeweiligen Antriebs für die Zugwalze oder Wickelachse, sowie die aktuelle Regelabweichung des Tänzerlagereglers. Der Ausgang "AddVelocityDancer" des Funktionsbausteins sollte zur Tänzerlageregelung auf den Parameter P-0-0690 "Geschwindigkeits-Sollwert additiv, Prozessregler" wirken (externe Verschaltung). Die aktuelle Regelabweichung "CurrentDancerDeviation" steht zu Diagnosezwecken zur Verfügung. Die Proportionalverstärkung (Faktor K_P) und die Low- und High-Werte für die Stellgrößenbegrenzung "DancerLimits" werden automatisch mit dem Eingang "ReferenceDiameter", bezogen auf die Formatlänge "FormatLength", angepasst. Für den Fall einer Tänzerlagerege-

lung für eine Zugwalze ist der Durchmesser der Zugwalze "ReferenceDiameter" anzugeben. Bei einem Wickler mit Tänzerlageregelung ist "ReferenceDiameter" der aktuelle Wickeldurchmesser. Sind die Werte für den integralen Anteil (Nachstellzeit T_n , "IControl") und das Vorhaltglied (Vorhaltezeit T_D , "DControl") gleich Null, sind die internen Speichergrößen des PID-Reglers gelöscht. Am Eingang "FeedBackDancer" muss ein Wert anliegen, der proportional zur Tänzerwalzenverstellung ist (z.B. ein analoger Spannungswert). Sollwert und Istwert müssen die gleiche Einheit besitzen.



Der Funktionsbaustein verarbeitet nur positive Sollwerte und Istwerte.

Negative Sollwerte führen zu einem Fehler.

Negative Istwerte werden auf Null begrenzt.

Tänzerlageregelung

Zur Erfassung der Tänzerposition wird ein Messglied benötigt, das die Tänzerlage in ein proportionales Signal umsetzt. Die Ist-Position des Tänzers wird von einem gegebenen Sollwert abgezogen und als Regelabweichung auf einen intern eingebundenen PID-Regler gegeben. Die Stellgröße des Reglers wirkt auf den Drehzahlsollwert mittels Addition. Somit dreht sich die Wickelachse im Vergleich zur Leitachse schneller oder langsamer. Hiermit wird die Lage des Tänzers geregelt.

Unter normalen Bedingungen (z.B. keine hohen Drehzahlsollwertsprünge) erfolgt die Grobeinstellung der Umfangsgeschwindigkeit einer Wickelachse über den Getriebefeinabgleich, die Grobeinstellung einer Zugachse über das elektronische Getriebe. Der Tänzerlageregler sollte somit lediglich kleine Abweichungen ausregeln.

Drehrichtung der Zug- bzw. Wickelachse, Regelsinn der Tänzerregelung

Die Drehrichtung der Zug- bzw. Wickelachse kann mit Hilfe des Antriebsparameters P-0-0108 "Polarität Leitantrieb" bzw. des Achsparameters A-0-2798 "Polarität Leitachsposition" eingestellt werden. Dreht sich die Wickelachse nicht in die gewünschte Richtung, ist der Parameter P-0-0108 bzw. der Achsparameter A-0-2798 entsprechend umzustellen. Bei einer Invertierung der Drehrichtung des Antriebes ist auch der Regelsinn (ControlPolarity) zu invertieren.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins aufgelistet und beschrieben.

Es werden die Fehlercodes aus der "F_RELATED table (16#0170)" verwendet. Zusätzlich können die Fehlercodes des Bausteines IL_PIDType01 aus der Bibliothek RIL_LoopControl (siehe Dokumentation "Rexroth IndraWorks 12VRS Basis-Bibliotheken"), sowie MB_GetCyclicParameterHandle (siehe [Kap. 5.9.5 "MB_GetCyclicParameterHandle" auf Seite 133](#)) auftreten.

RMB_TechWinder.library/MX_TechWinder.lib

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1160	16#0020	Fehler in der State machine
INPUT_RANGE_ERROR	16#1161	16#0002	HighLimit < LowLimit
		16#0003	FormatLength < 0.0
		16#0004	ReferenceDiameter < 0.0
		16#0005	ProcessWindow < 0.0
		16#0006	HighLimit < 0.0
		16#0007	LowLimit < 0.0
		16#0008	VelocityHighLimit < 0.0
		16#0009	P-Gain < 0.0
		16#000A	Setpoint < 0.0
		16#000B	MinFeedback < 0.0
		16#000C	MaxFeedback < 0.0
		16#000D	MaxFeedback < MinFeedback
		16#000E	Nummer der Slaveachse falsch
		16#000F	Nummer der Masterachse falsch
DEVICE_ERROR	16#1162	16#0001	Achse nicht im Betriebsmodus
RESSOURCE_ERROR	16#0004	16#0000	Falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Abb.9-14: Fehlercodes Funktionsbaustein MB_DancerControlType03

Notwendige Hardware

- IndraMotion MLC Hardware CML65, CML45, CML25
- IndraDrive

Notwendige Firmware

- IndraMotion MLC Firmware FWA-CMLXX*-MLC-12VRS
XX - Gerätevariante, z.B. CML65
- IndraDrive Antriebsfirmware

Notwendige Software

- IndraWorks 12VRS für IndraMotion MLC oder IndraMotion MLD

Besonderheiten für IndraMotion MLC

Für die Tänzerreglerachse muss Interpolation im Antrieb aktiviert sein.
(reale Achsen → entspr. Achse → Eigenschaften)

Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Im zyklischen SERCOS-Kanal des entsprechenden Antriebs muss der **Parameter P-0-0690 ("Geschwindigkeits-Sollwert additiv, Prozessregler")** eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.

Besonderheiten für IndraMotion MLD

Bei Verwendung der Steuerung IndraMotion MLD-S mit übergeordneter Steuerung IndraMotion MLC muss für die Tänzerreglerachse "Interpolation im Antrieb" aktiviert sein (reale Achsen → entspr. Achse → Eigenschaften). Bei Verwendung der Steuerung IndraMotion MLD-M gibt es diese Einstellung nicht.

Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

RMB_TechWinder.library/MX_TechWinder.lib

Im zyklischen Kanal zwischen Steuerung und Antrieb des entsprechenden Antriebs muss der **Parameter P-0-0690 ("Geschwindigkeits-Sollwert additiv, Prozessregler")** eingetragen sein ("UserCmdDataX")

Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. (Der Dialog ist im Projektbaum unterhalb des MLD-Knotens. → AxisData)

Die "AxisData-Struktur" des Antriebes muss parametrierung sein (P-0-1367 "SPS-Konfiguration" Bit 6 ist gesetzt, bzw. Dialog SPS-Konfiguration)

9.2.3 Inbetriebnahme Tänzerlageregelung

Antriebsregelkreis	Es wird vorausgesetzt, dass der Drehzahlregelkreis des Zugwalzenantriebs eingestellt wurde. D.h. die Parameter S-0-0100 "Geschwindigkeitsregler-Proportionalverstärkung" und S-0-0101 "Geschwindigkeitsregler-Nachstellzeit" müssen der Applikation entsprechend eingestellt sein.
Kommandierung der Tänzerreglerachse	Für die Tänzerreglerachse muss die richtige Betriebsart des Antriebes sowie die Reglerfreigabe des Antriebes gesetzt werden. Dies kann auf zwei unterschiedliche Arten erfolgen: • Setzen der Betriebsart "ModeSyncVel" mit Hilfe des Achsinterfaces (dies wird empfohlen) • Benutzung der PLC-Open Funktionsbausteine "MC_Power", "MC_GearIn", "MC_Stop" zur Kommandierung der Achse
Grundsätzliche Einstellungen	Zunächst wird der Regelsinn des Tänzerlagereglers, abhängig von der Drehrichtung der Zugspannungsachse, festgelegt. Abhängig hiervon wirkt der Regler in entgegengesetzten Richtungen. Der Regelsinn wird durch Setzen des Eingangs "ControlPolarity" festgelegt. Als weitere wichtige Grundeinstellung gilt die Festlegung des Bezugsdurchmessers (Eingang "ReferenceDiameter"). Beim Wickler mit Tänzer (siehe Kapitel 9 Funktionsbausteine zur Wicklerfunktionalität, Seite 425) wird hier der aktuell berechnete Wickeldurchmesser angeschlossen. Beim Tänzerlageregler wird der Durchmesser der Zugachse eingetragen. Mit den Eingängen "MinDancerFeedback" und "MaxDancerFeedback" können Werte für den Verstellbereich des Tänzers angegeben werden, die dann mittels der Ausgänge "MinFeedbackAchieved" und "MaxFeedbackAchieved" angezeigt werden. Diese sollten innerhalb der Endlagen des Tänzers liegen. Als nächstes wird das Prozessgrößenfenster ("WindowDancer") definiert. Mit dem Fenster wird die Überwachung der Tänzersollposition parametrierung. Entfernt sich der Tänzerlageistwert vom Lagesollwert ("SetpointDancer"), bezogen auf den Sollwert um mehr als den angegebenen Prozentsatz, wird der binäre Ausgang "InWindow" zurückgesetzt ("InWindow": = FALSE). Dies ermöglicht der übergeordneten SPS entsprechend auf den Vorgang zu reagieren. Ein Verlassen des parametrierung Fensters hat zunächst keinen Einfluss auf die Funktion des Tänzerlagereglers. Beträgt beispielsweise der Tänzer-sollwert "SetpointDancer" = 500.0 und die Fenstergröße ist auf "WindowDancer" = 10% festgelegt, liegen die Grenzen des Fensters bei 450.0 und 550.0. Innerhalb dieses Wertebereichs ist der Ausgang "InWindow" = TRUE. Schließlich muss die Formatlänge angegeben werden. Die Formatlänge entspricht einer abgewickelten Umdrehung der Referenzachse (Eingang "FormatLength"). Parametrierung des Tänzerlagereglers In einem nächsten Schritt werden die Reglerparameter eingestellt. Diese sind die • Proportionalverstärkung ("PControl")

RMB_TechWinder.library/MX_TechWinder.lib

- Nachstellzeit ("IControl")
- Vorhaltezeit ("DControl")

Zur Einstellung der Größen kann z.B. die Methode von Ziegler und Nichols eingesetzt werden. Der Regelkreis wird geschlossen und die Proportionalverstärkung ("PControl") solange erhöht, bis der Ausgang des Regelkreises bei konstantem Eingang eine Dauerschwingung mit der Periode T_{krit} bei der Proportionalverstärkung $K_{p,krit}$ ausführt. Die Einstellregeln für die Verstärkung K_p ("PControl"), die Nachstellzeit T_n ("IControl") und die Vorhaltezeit T_v ("DControl") lauten, wie in folgender Tabelle angegeben.

Regler	PControl	IControl	DControl
P-Regler	$0,5 \cdot K_{p,krit}$		
PI-Regler	$0,45 \cdot K_{p,krit}$	$0,85 \cdot T_{krit}$	
PD-Regler	$0,55 \cdot K_{p,krit}$		$0,15 \cdot T_{krit}$
PID-Regler	$0,6 \cdot K_{p,krit}$	$0,5 \cdot T_{krit}$	$0,12 \cdot T_{krit}$

Abb.9-15: Reglerparametrierung nach Ziegler und Nichols

Schließlich werden die Eingänge der Struktur MB_WIND_DANCER_LIMITS belegt. Hier wird der Arbeitsbereich des Tänzerlagereglers durch Angabe der Unter- und Obergrenze ("LowLimitControl", "HighLimitControl") des internen Begrenzers festgelegt.

Erster Anlauf des Tänzerlagereglers

Zunächst wird am Eingang "SetpointDancer" ein Wert als Sollwert vorgegeben. Es ist in der Regel zweckmäßig, dass dieser Wert in etwa der Mittellage des Tänzers entspricht. In diesem Fall hat der Tänzer in beide Richtungen einen vergleichsweise großen Bahnweg.

Nach dem Einschalten des Tänzerlagereglers ("Enable" = TRUE) fährt der Tänzer in die vorgegebene Sollwertlage. Die Schwankungen der Bahngeschwindigkeit verursachen Schwankungen in der Bahnspannung, die auf die Tänzerlage und somit auf die Regelabweichung des Tänzerlagereglers wirken. Dies versucht der Tänzer über den Tänzerlageregler durch Anpassung der Antriebsgeschwindigkeit der Zugachse bzw. Wickelachse auszugleichen. An den Ausgängen "CurrentDancerDeviation" und "AddVelocityDancer" werden diese Größen angezeigt und können hiermit überwacht werden.

9.3 Funktionsbaustein zur Durchmesserberechnung

9.3.1 Einführung und Übersicht

Der Funktionsbaustein MB_DiameterCalculatorType03 stellt einen vielseitigen Funktionsbaustein für die Umsetzung von Wickelrechnern dar. Er kommt in dieser Dokumentation in allen beschriebenen Wickelrechnern zum Einsatz. Aus diesem Grund wird er zuerst beschrieben. Im Folgenden wird der Funktionsbaustein erläutert.

9.3.2 MB_DiameterCalculatorType03

Kurzbeschreibung

Der Baustein berechnet zyklisch den aktuellen Durchmesser, die Geschwindigkeitsnachführung für die Wickelachse und das aktuelle Trägheitsmoment des Wickelgutes auf einem Wickel. Der Funktionsbaustein wird für jede Wickelachse aufgerufen. Zyklische Ausgangsgrößen sind der aktuelle Wickeldurchmesser "CurrentDiameter", der resultierende Getriebefeinabgleich

RMB_TechWinder.library/MX_TechWinder.lib

für den zugehörigen Wickelantrieb "GearRatioFineAdjust" und das aktuelle Trägheitsmoment des Wickelgutes "CurrentInertia".

Der Getriebefeinabgleich "GearRatioFineAdjust" sollte auf den Parameter P-0-0694 ("Feinabgleich Getriebeübersetzung, Prozessregler") wirken. Dieser Parameter wird im entsprechenden zyklischen SERCOS-Kanal eingetragen. Die Parametrierung erfolgt im Parametriermodus.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechWinder.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechWinder.lib

Schnittstellenbeschreibung

Abb.9-16: Referenztabelle MB_DiameterCalculatorType03

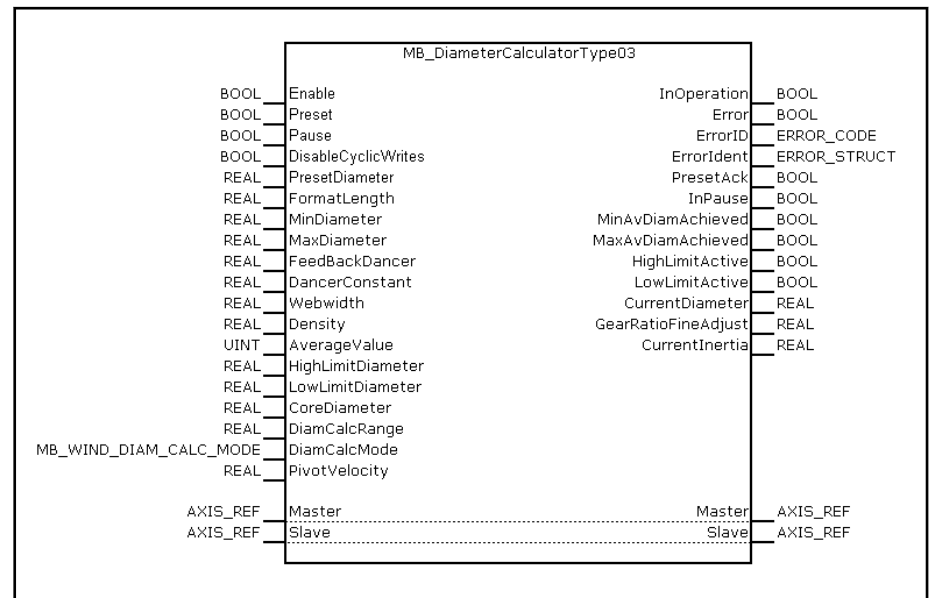


Abb.9-17: Funktionsbaustein MB_DiameterCalculatorType03

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (Zugachse)
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	Preset	BOOL	Preset des aktuellen Wickeldurchmessers
	Pause	BOOL	Durchmesserrechner pausiert (pegelgesteuert)
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	PresetDiameter	REAL	CurrentDiameter wird auf PresetVal gesetzt, GearRatioFineAdjust entsprechend angepasst (flanken-gesteuert)
	FormatLength	REAL	Formatlänge [mm] (entspricht einer abgewickelten Umdrehung der Referenzachse)

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Typ	Kommentar
	MinDiameter	REAL	minimaler Durchmesser [mm] Wird lediglich für die Anzeige durch den Ausgang "MinAvDiamAchieved" verwendet
	MaxDiameter	REAL	maximaler Durchmesser [mm] Wird lediglich für die Anzeige durch den Ausgang "MaxAvDiamAchieved" verwendet
	FeedbackDancer	REAL	Tänzerlageistwert (nur für die Funktionalität "Wickler mit Tänzer")
	DancerConstant	REAL	Bahnweg bei Verstellung der Tänzerwalze, bezogen auf die Einheit der Tänzerverstellung [mm] Vorzeichen ist abhängig vom Aufbau des Tänzers einzustellen! (nur für die Funktionalität "Wickler mit Tänzer")
	Webwidth	REAL	Bahnbreite [mm]
	Density	REAL	Materialdichte [kg/dm ³]
	AverageValue	UINT	Anzahl der Durchmesserwerte, die gemittelt werden. Eingang wird auf maximal 30 begrenzt
	HighLimitDiameter	REAL	Obere Grenze für den Durchmesser [mm]. Auf diesen Wert wird der Durchmesser begrenzt
	LowLimitDiameter	REAL	Untere Grenze für den Durchmesser [mm]. Der Durchmesser unterschreitet diesen Wert nicht
	CoreDiameter	REAL	Hülsendurchmesser des Wickels [mm]
	DiamCalcRange	REAL	Bereich für die Durchmesserberechnung [Grad]. Gibt an, nach wieviel Grad der Durchmesser regelmäßig berechnet werden soll. Bezieht sich auf die Wickelachse (lastseitig)
	DiamCalcMode	MB_WIND_DI-AM_CALC_MODE	Modus für die Durchmesserberechnung 0x0000: CALC_DIAM_MANUAL Durchmesserberechnung abhängig von CalcRange 0x0001: FIRST_REV_AUTO_CALC Durchmesserberechnung erfolgt für 1. Umdrehung alle 5 Grad, danach abhängig vom CalcRange
	PivotVelocity	REAL	Schwenkgeschwindigkeit des Wickels (Relativgeschwindigkeit zur Wickelbewegung) [mm/s] Damit wird die Durchmesserberechnung bei einer Relativbewegung der Wickelachse korrigiert
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	PresetAck	BOOL	Presetdurchmesser wurde übernommen (ist gesetzt, solange Preset ansteht)

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Typ	Kommentar
	InPause	BOOL	Durchmesserrechner pausiert
	MinAvDiamAchieved	BOOL	minimaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	MaxAvDiamAchieved	BOOL	maximaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	HighLimitActive	BOOL	Aktuell berechneter Durchmesser hat obere Grenze erreicht (Durchmesserwert wird verworfen)
	LowLimitActive	BOOL	Aktuell berechneter Durchmesser hat untere Grenze erreicht. (Durchmesserwert wird verworfen)
	CurrentDiameter	REAL	aktueller Wickeldurchmesser
	GearRatioFineAdjust	REAL	Getriebefeinabgleich zur Drehzahlachsführung des Antriebes [%]
	CurrentInertia	REAL	aktuelles gesamtes Trägheitsmoment (bezogen auf die Motorseite) [kg/m ²]

Abb.9-18: Schnittstellenvariablen Funktionsbaustein MB_DiameterCalculator-Type03

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Preset	BOOL			FALSE	kontinuierlich
Pause	BOOL			FALSE	kontinuierlich
DisableCyclicWrites	BOOL			FALSE	steigende Flanke an Enable
PresetDiameter	REAL	> 0.0	n. def.	100.0	kontinuierlich
FormatLength	REAL	> 0.0	n. def.	100.0	steigende Flanke an Enable
MinDiameter	REAL	> 0.0	n. def.	100.0	kontinuierlich
MaxDiameter	REAL	> 0.0	n. def.	2000.0	kontinuierlich
FeedbackDancer	REAL	0.0	n. def.	0.0	kontinuierlich
DancerConstant	REAL	n. def.	n. def.	0.0	kontinuierlich
Webwidth	REAL	0.0	n. def.	0.0	steigende Flanke an Enable
Density	REAL	0.0	n. def.	1.0	steigende Flanke an Enable
AverageValue	UINT	1	30	1	kontinuierlich
HighLimitDiameter	REAL	0.0	n. def.	2000.0	kontinuierlich
LowLimitDiameter	REAL	0.0	n. def.	100.0	kontinuierlich

RMB_TechWinder.library/MX_TechWinder.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
CoreDiameter	REAL	0.0	n. def.	100.0	steigende Flanke an Enable
DiamCalcRange	REAL	0.0	n. def.	360.0	kontinuierlich
DiamCalcMode	ENUM			0	steigende Flanke an Enable
PivotVelocity	REAL	0.0	n. def.	0.0	kontinuierlich

Abb.9-19: Min-/Max- und Default-Werte MB_DiameterCalculatorType03

Name	Wert	Beschreibung
CALC_DIAM_MANUAL	0	Durchmesserberechnung abhängig von CalcRange
FIRST_REV_AUTO_CALC	1	Durchmesserberechnung erfolgt für 1. Umdrehung alle 3 Grad, danach abhängig vom CalcRange

Abb.9-20: Elemente des Aufzählungstyps MB_WIND_DIAM_CALC_MODE

Signal-Zeit-Diagramm

Im folgenden Signal-Zeit-Diagramm werden die Ein- bzw. Ausgänge des Funktionsbausteins betrachtet, die dessen Betrieb betreffen. Ohne Setzen des Eingangs "Enable" ist ein Betrieb des Bausteins nicht möglich. Die Betriebsbereitschaft wird durch den Ausgang "InOperation" signalisiert und lediglich durch einen Fehler, gekennzeichnet durch den Ausgang "Error", unterbrochen. Bei positiver Flanke am Eingang "Preset" wird der Wert, der am Eingang "PresetDiameter" anliegt als Ausgang des Durchmesserrechners geschrieben. Dies ist unabhängig vom Eingang "Pause" möglich. Das Setzen von "Preset" wird durch den Ausgang "PresetAck" bestätigt. Ein Setzen des Eingangs "Pause", der die Berechnung des Durchmessers einfriert, wird am Ausgang "InPause" angezeigt.

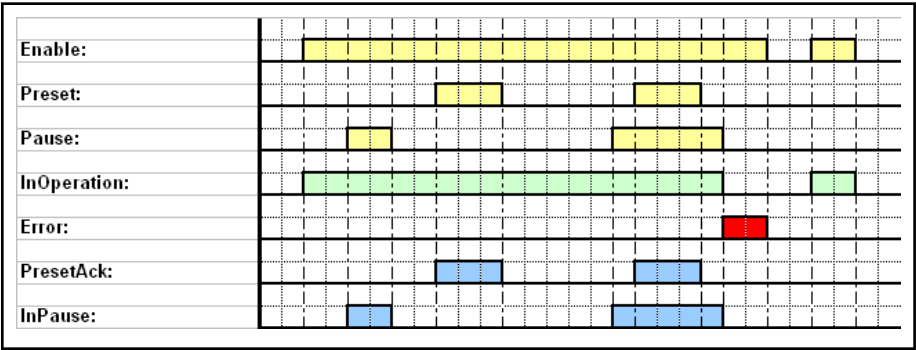


Abb.9-21: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_DiameterCalculatorType03

Funktionsbeschreibung

Preset des Durchmessers

Da der berechnete Durchmesser über den resultierenden Feinabgleich direkt auf die Wickelachsgeschwindigkeit wirkt, muss zu Beginn der Berechnung bei Aktivierung des Wickelrechners ein sinnvoller Wert für den Durchmesser vorgegeben werden. Daher sollte am Eingang "PresetDiameter" ein Wert für den Durchmesser angegeben werden, der nahe dem des tatsächlichen Durchmessers liegt und ein Preset durchgeführt werden. Im Falle eines Aufwicklers sollte der Wert etwas kleiner und für Abwickler etwas größer gewählt werden. Damit wird jeweils ein leichter Anlauf ermöglicht.

Mit jeder neuen Rolle sollte einmalig zu Beginn ein Preset des Wickeldurchmessers durchgeführt werden. Dies erfolgt über den Eingang "Preset". Der Durchmesser wird auf den Wert des Eingangs "PresetDiameter" gesetzt. Das

Setzen des Durchmessers mit Hilfe von "Preset" ist zu jeder Zeit ohne Setzen des Eingangs "Pause" möglich.

Nach dem Einschalten der Steuerung ist der Wickeldurchmesser, der Getriebefeinabgleich und das Trägheitsmoment gleich Null. Dies bedeutet, dass nach jedem Einschalten der Steuerung ein Preset durchgeführt werden sollte, um die entsprechenden Größen mit sinnvollen Werten vorzubelegen. Erfolgt dies nicht, kann ein nicht sinnvoller Wert für den aktuellen Wickeldurchmesser berechnet werden. Dies kann zur Fehlfunktion des Wickelrechners und damit zum Halt des Wicklers führen.

Ist es nicht möglich einen korrekten Presetwert für den Anfangswickels zu bestimmen, wird der "auto-calc-mode" für die Durchmesserberechnung empfohlen (FIRST_REV_AUTO_CALC). Hierbei wird für die erste Wickelachsumdrehung nach dem Einschalten der Bereich für die Durchmesserberechnung (DiamCalcRange) automatisch angepasst, um schneller einen vernünftigen Anfangsdurchmesser zu bekommen.



Soll der aktuelle Durchmesser, nach dem Ausschalten der Steuerung erhalten bleiben, muss der Ausgang "CurrentDiameter" auf eine remanente Variable kopiert werden. Mit diesem Wert kann nach dem Einschalten wieder ein Preset ausgeführt werden.

Durchmesserberechnung

Die Durchmesserberechnung ist proportional zum Quotienten aus der Referenzachsengeschwindigkeit n_{Ref} und der Wickelachsgeschwindigkeit n_{Wickel} und zur Formatlänge F (entspricht einer abgewickelten Umdrehung der Referenzachse).

$$d \sim \frac{n_{Ref}}{n_{Wickel}} * F$$

Abb.9-22: Proportionalität bei der Durchmesserberechnung



Das Verhältnis "Formatlänge" / "CoreDiameter" darf nicht größer als 25 sein.

Ansonsten reicht die Getriebeanpassung über den Getriebefeinabgleich nicht aus!

Der berechnete Wickeldurchmesser wirkt direkt auf den resultierenden Feinabgleich des Getriebes. Damit wird eine durchmesserabhängige Geschwindigkeitsanpassung des Wickels umgesetzt.

Mittelwertbildung

Der berechnete aktuelle Durchmesser "CurrentDiameter" kann mit Setzen des Eingangs "AverageValue" gemittelt werden.

An "AverageValue" wird die Anzahl der Durchmesserwerte eingetragen, über die gemittelt werden soll. Zur Mittelwertbildung tragen nur die intern berechneten aktuellen Durchmesser bei, die innerhalb der Grenzen von "HighLimitDiameter" und "LowLimitDiameter" liegen. Außerhalb liegende Werte werden verworfen. Für "AverageValue" sind Werte zwischen 1 und 10 zweckmäßig. Aus diesem Grund wird "AverageValue" auf maximal 30 begrenzt. Wird ein größerer Wert eingegeben, wird "AverageValue" auf 30 begrenzt. Je höher die Zahl gewählt wird, desto glatter wird der berechnete Durchmesser. Allerdings nimmt mit zunehmenden Werten für "AverageValue" die Genauigkeit des Durchmessers ab, da dieser aufgrund der "Filterung" nachläuft. Daher ist der Wert für "AverageValue" geeignet zu wählen. Bei kleineren Wickeldurch-

RMB_TechWinder.library/MX_TechWinder.lib

messern ist die Änderung der berechneten Durchmesserwerte höher. Aus diesem Grund darf der Wert des Filters nicht zu groß gewählt werden. Die durch die Mittelung verursachte Verzögerung des berechneten Durchmessers hätte sonst ein deutliches "Hinterherlaufen" des berechneten Durchmessers zur Folge. Bei einem Wickler mit Tänzer beispielsweise kann dies zu unerwünschten Auslenkungen des Tänzers führen.

Berechnung des Trägheitsmoments

Mit Hilfe der Eingänge "Webwidth" und "Density" wird intern im Durchmesserrechner das Trägheitsmoment des Wickels berechnet und am Ausgang "CurrentInertia" angezeigt. Mit Hilfe des, vom Durchmesser abhängigen, Trägheitsmoments kann, falls erforderlich, eine Adaption der K_P -Verstärkung des Drehzahlreglers (S-0-0100) im SPS-Projekt in IndraLogic implementiert werden.

Vorbereitung Rollenwechsel

Schließlich können die Eingänge "MinDiameter" und "MaxDiameter" belegt werden. Die hier angegebenen Größen wirken auf die Ausgänge "MinAvDiamAchieved" und "MaxAvDiamAchieved" und melden das Überschreiten der angegebenen Grenzen durch den berechneten aktuellen Wickeldurchmesser. Die jeweiligen Werte sollten innerhalb des erlaubten Durchmesserbereichs liegen, der durch die Eingänge "HighLimitDiameter" und "LowLimitDiameter" definiert wird. So können z.B. für einen Rollenwechsel oder die Anpassung des Eingangs "DiamCalcRange" anhand einer Vorgabe des minimalen/maximalen Durchmessers der Zeitpunkt für bestimmte Prozessschritte in der SPS detektiert und angezeigt werden.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins aufgelistet und beschrieben. Es werden die Fehlercodes aus der "F_RELATED table (16#0170)" verwendet. Zusätzlich können die Fehlercodes des Bausteins MB_GetCyclicParameterHandle (siehe [Kap. 5.9.5 "MB_GetCyclicParameterHandle" auf Seite 133](#)) auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1100	16#0020	Fehler in der StateMachine
INPUT_RANGE_ERROR	16#1101	16#0001	HighLimitDiameter < LowLimitdiameter
		16#0002	FormatLength < 0.0
		16#0003	Webwidth < 0.0
		16#0004	Density < 0.0
		16#0005	Core diameter < 0.0
		16#0006	DiamCalcRange < 0.0
		16#0007	Preset diameter < 0.0
		16#0008	Minimum diameter < 0.0
		16#0009	Maximum diameter < 0.0
		16#000A	HighLimitDiameter < 0.0
		16#000B	LowLimitDiameter < 0.0
		16#000C	DiamCalcMode ungültig
		16#000D	Nummer der Slaveachse falsch
		16#000E	Nummer der Masterachse falsch

ErrorID	Additional1	Additional2	Text
DEVICE_ERROR	16#1102	16#0001	kein Betriebsmodus (keine SERCOS-Phase 4)
RESSOURCE_ERROR	16#0004	16#0000	falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Abb.9-23: Fehlercodes Funktionsbaustein MB_DiameterCalculatorType03

Notwendige Hardware	• IndraMotion MLC Hardware CML65, CML45, CML25 • IndraDrive
Notwendige Firmware	• IndraMotion MLC Firmware FWA-CMLXX*-MLC-12VRS XX - Gerätevariante, z.B. CML65 • IndraDrive Antriebsfirmware
Notwendige Software	• IndraWorks 12VRS für IndraMotion MLC bzw. IndraMotion MLD
Besonderheiten für IndraMotion MLC	Für die Wickelachse muss Interpolation im Antrieb aktiviert sein. (reale Achsen → entspr. Achse → Eigenschaften) Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Im zyklischen SERCOS-Kanal des entsprechenden Antriebs muss der Parameter P-0-0694 ("Feinabgleich Getriebeübersetzung, Prozessregler") eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter S-0-0121 ("Lastgetriebe-Eingangsumdrehungen") und S-0-0122 ("Lastgetriebe-Ausgangsumdrehungen") eingetragen werden. Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z.B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!
Besonderheiten für IndraMotion MLD	Bei Verwendung der Steuerung IndraMotion MLD-S mit übergeordneter Steuerung IndraMotion MLC muss für die Wickelachse Interpolation im Antrieb aktiviert sein (reale Achsen → entspr. Achse → Eigenschaften). Bei Verwendung der Steuerung IndraMotion MLD-M gibt es diese Einstellung nicht. Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Im zyklischen Kanal zwischen Steuerung und Antrieb des entsprechenden Antriebs muss der Parameter P-0-0694 ("Feinabgleich Getriebeübersetzung, Prozessregler") eingetragen sein ("UserCmdDataX") Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus (der Dialog befindet sich im Projektbaum unterhalb des MLD-Knotens → AxisData). Die "AxisData-Struktur" des Antriebes muss parametrierbar sein (P-0-1367 "SPS-Konfiguration" Bit 6 ist gesetzt, bzw. Dialog SPS-Konfiguration). Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter S-0-0121 ("Lastgetriebe-Eingangsumdrehungen") und S-0-0122 ("Lastgetriebe-Ausgangsumdrehungen") eingetragen werden. Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z.B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!

RMB_TechWinder.library/MX_TechWinder.lib

9.3.3 Inbetriebnahme des Durchmesserrechners

Antriebsregelkreis Es wird vorausgesetzt, dass der Drehzahlregelkreis des Wickelantriebes eingestellt wurde. D.h. die Parameter S-0-0100 "Geschwindigkeitsregler-proportionalverstärkung" und S-0-0101 "Geschwindigkeitsregler-Nachstellzeit" müssen der Applikation entsprechend eingestellt sein.

Für die Inbetriebnahme des Durchmesserrechners sind die folgenden Schritte zweckmäßig:

- Festlegung der Stoffbahngrößen
- Festlegung der Wickelgrößen
- Festlegung der zyklisch veränderlichen Eingänge
- Festlegung der Eingänge zur Manipulation der Durchmesserberechnung und
- weitere Festlegungen

Festlegung der Stoffbahngrößen

Als Stoffbahngrößen werden die Eingänge und Werte bezeichnet, die sich auf die Abmaße der Stoffbahn beziehen und üblicherweise während des Betriebs als konstant angenommen werden können. Ausnahme bilden Stoffbahnen mit veränderlichem Material oder variablen Abmessungen.

Folgende Eingänge beschreiben Stoffbahngrößen:

- FormatLength
- Webwidth
- Density

Da diese Größen nicht zyklisch eingelesen werden, müssen sie vor der Aktivierung des Bausteins als Eingangswerte an den Eingängen anliegen.

Festlegung der Wickelgrößen

Wickelgrößen beschreiben die räumlichen, allgemeinen Abmaße des Wikkels. Diese sind der Hülsendurchmesser "CoreDiameter", die maximal und minimal zulässigen Wickeldurchmesser "LowLimitDiameter" und "HighLimitDiameter". Über das Verhältnis von Referenz- und Wickelachsdrehzahl wird mit dem Hülsendurchmesser der aktuelle Durchmesser bestimmt. Der aktuelle Durchmesser "CurrentDiameter" wird auf diese angegebenen Grenzen begrenzt.

Festlegung zyklisch veränderlicher Eingänge

Zu Beginn und während der aktiven Durchmesserberechnung, kann es erforderlich sein, den Wert des Durchmessers manuell zu setzen. Hierzu wird der Eingang "PresetDiameter" beschrieben. Er wird zyklisch eingelesen und somit innerhalb eines Zyklus wirksam.

Festlegung der Eingänge zur Manipulation der Durchmesserberechnung

Die Methode zur Berechnung des Durchmessers ist so implementiert, dass mittels zweier Eingänge des Funktionsbaustein auf das Resultat der Berechnung Einfluss genommen werden kann. "AverageValue" gibt an, über wie viele berechnete Durchmesserwerte gleitend gemittelt werden soll. "DiamCalcRange" erlaubt es, die Häufigkeit der Berechnung zu beeinflussen, was wiederum Einfluss auf die Genauigkeit des Durchmesserwertes hat. Beide Größen können, da sie zyklisch eingelesen und verarbeitet werden, während des Betriebs manipuliert und angepasst werden.

Weitere Festlegungen

Abhängig von der jeweiligen Anwendung werden die Eingänge "DancerConstant", "FeedBackDancer" (beide nur für den Fall der Funktionalität eines

RMB_TechWinder.library/MX_TechWinder.lib

Wicklers mit Tänzer), "MinDiameter" und "MaxDiameter" beschaltet. Sollten diese Eingänge für die zu implementierenden Funktionen im SPS-Projekt nicht benötigt werden, ist es empfehlenswert, die Eingänge "MinDiameter" und "MaxDiameter" mit den gleichen Werten, wie "LowLimitDiameter" und "HighLimitDiameter" zu belegen. Hierdurch wird eine entsprechende Fehlermeldung vermieden.

Kommandierung der Wicklerachse

Für die Wicklerachse muss die richtige Betriebsart des Antriebes, sowie die Reglerfreigabe des Antriebes gesetzt werden. Dies kann auf zwei unterschiedliche Arten erfolgen:

- Setzen der Betriebsart "ModeSyncVel" mit Hilfe des Achsinterfaces (dies wird empfohlen)
- Benutzung der PLC-Open Funktionsbausteine "MC_Power", "MC_GearIn", "MC_Stop" zur Kommandierung der Achse

9.4 Funktionsbaustein zur Erfassung eines extern gemessenen Durchmessers

9.4.1 Einführung und Übersicht

Der Funktionsbaustein MB_DiameterMeasurementType01 liest einen gemessenen Durchmesser ein und führt die Geschwindigkeit des Wickelantriebes nach. In diesem Fall ersetzt er dann den Baustein MB_DiameterCalculatorType03. Er wird in den Bausteinen MB_WinderDancerCtrlType01 und MB_WinderTensionCtrlType02 verwendet.

9.4.2 MB_DiameterMeasurementType01

Kurzbeschreibung

Der Baustein liest einen gemessenen Durchmesser ein und berechnet die Geschwindigkeitsnachführung für die Wickelachse und das aktuelle Trägheitsmoment des Wickelgutes auf einem Wickel. Der Funktionsbaustein wird für jede Wickelachse aufgerufen, bei der der Durchmesser gemessen wird. Zyklische Ausgangsgrößen sind der resultierende Getriebefeinabgleich für den zugehörigen Wickelantrieb "GearRatioFineAdjust" und das aktuelle Trägheitsmoment des Wickelgutes "CurrentInertia".

Der Getriebefeinabgleich "GearRatioFineAdjust" sollte auf den Parameter P-0-0694 ("Feinabgleich Getriebeübersetzung, Prozessregler") wirken. Dieser Parameter wird im entsprechenden zyklischen SERCOS-Kanal eingetragen. Die Parametrierung erfolgt im Parametriermodus.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechWinder.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechWinder.lib

Abb.9-24: Referenztable MB_DiameterMeasurementType01

RMB_TechWinder.library/MX_TechWinder.lib

Schnittstellenbeschreibung

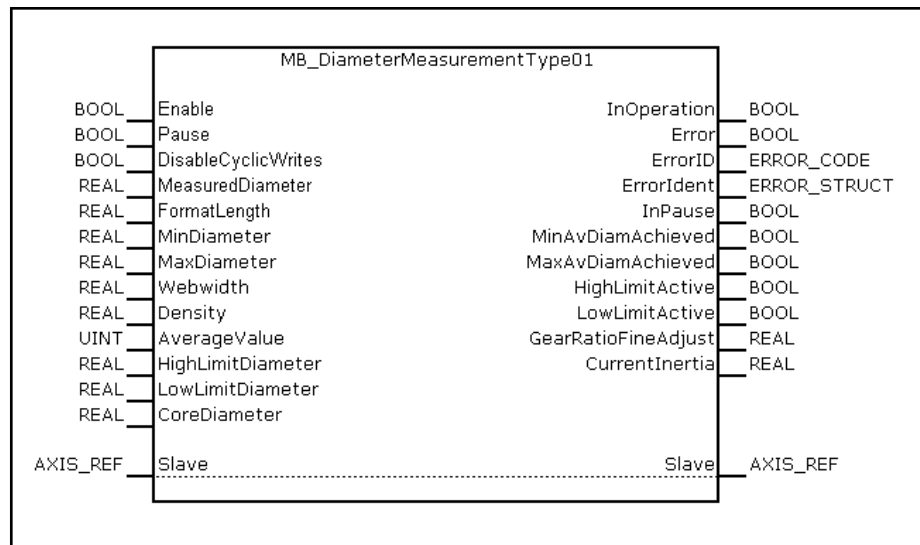


Abb.9-25: Funktionsbaustein MB_DiameterMeasurementType01

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Slave	AXIS_REF	Referenz zur Slaveachse
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	Pause	BOOL	FB pausiert, Getriebefeinabgleich bleibt "eingefroren"
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	MeasuredDiameter	REAL	gemessener Durchmesser [mm]
	FormatLength	REAL	Formatlänge [mm] (1 Umdrehung der Masterachse abgewickelt)
	MinDiameter	REAL	minimaler Durchmesser [mm], wird lediglich für die Anzeige durch den Ausgang "MinAvDiamAchieved" verwendet
	MaxDiameter	REAL	maximaler Durchmesser [mm], wird lediglich für die Anzeige durch den Ausgang "MaxAvDiamAchieved" verwendet
	WebWidth	REAL	Bahnbreite [mm]
	Density	REAL	Materialdichte [kg/dm³]
	AverageValue	UINT	Anzahl der Durchmesserwerte, die gemittelt werden. Eingang wird auf maximal 30 begrenzt
	HighLimitDiameter	REAL	obere Grenze für den Durchmesser [mm] Auf diesen Wert wird der Durchmesser begrenzt
	LowLimitDiameter	REAL	untere Grenze für den Durchmesser [mm] Auf diesen Wert wird der Durchmesser begrenzt
	CoreDiameter	REAL	Hüssendurchmesser des Wickels [mm]
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Typ	Kommentar
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	InPause	BOOL	Tänzerregler pausiert
	MinAvDiamAchieved	BOOL	minimaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	MaxAvDiamAchieved	BOOL	maximaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	HighLimitActive	BOOL	aktuell berechneter Durchmesser hat obere Grenze erreicht (Durchmesserwert wird verworfen)
	LowLimitActive	BOOL	aktuell berechneter Durchmesser hat untere Grenze erreicht (Durchmesserwert wird verworfen)
	GearRatioFineAdjust	REAL	Getriebefeinabgleich zur Drehzahlachsführung des Antriebes [%]
	CurrentInertia	REAL	aktuelles gesamtes Trägheitsmoment (bezogen auf die Motorseite) [kg/m²]

Abb.9-26: Schnittstellenvariablen Funktionsbaustein MB_DiameterMeasurement-Type01

Min-/Max- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Pause	BOOL			FALSE	kontinuierlich
DisableCyclicWrites	BOOL			FALSE	steigende Flanke an Enable
MeasuredDiameter	REAL	> 0.0	n. def.	100.0	kontinuierlich
FormatLength	REAL	> 0.0	n. def.	100.0	steigende Flanke an Enable
MinDiameter	REAL	> 0.0	n. def.	100.0	kontinuierlich
MaxDiameter	REAL	> 0.0	n. def.	2000.0	kontinuierlich
Webwidth	REAL	0.0	n. def.	0.0	steigende Flanke an Enable
Density	REAL	0.0	n. def.	1.0	steigende Flanke an Enable
AverageValue	UINT	0	30	1	kontinuierlich
HighLimitDiameter	REAL	0.0	n. def.	2000.0	kontinuierlich

RMB_TechWinder.library/MX_TechWinder.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
LowLimitDiameter	REAL	0.0	n. def.	100.0	kontinuierlich
CoreDiameter	REAL	0.0	n. def.	100.0	steigende Flanke an Enable

Abb.9-27: Min-/Max- und Default-Werte der Eingänge MB_DiameterMeasurementType01

Signal-Zeitdiagramm

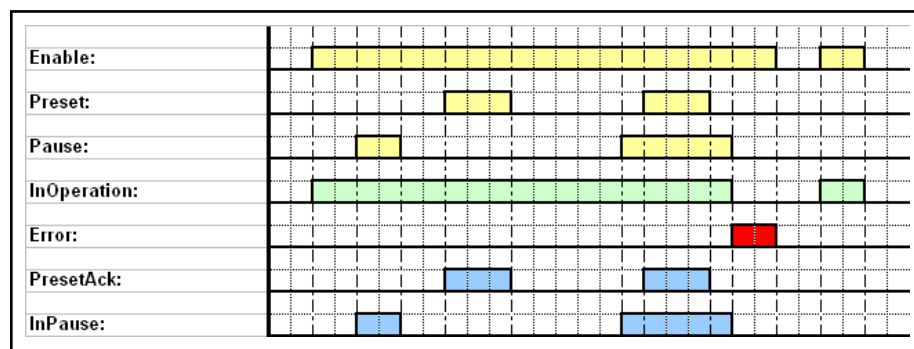


Abb.9-28: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_DiameterMeasurementType01

Funktionsbeschreibung

Der Funktionsbaustein MB_DiameterMeasurementType01 liest einen gemessenen Durchmesserwert (z.B. mit einem Ultraschallsensor) ein und sorgt für die richtige Geschwindigkeitsanpassung der jeweiligen Wickelsachse bezogen auf den gemessenen Durchmesser.

Der Baustein wird in den Bausteinen MB_WinderDancerCtrlType01 und MB_WinderTensionCtrlType02 verwendet.



Das Verhältnis "Formatlänge" / "CoreDiameter" darf nicht größer als 25 sein.

Ansonsten reicht die Getriebearpassung über den Getriebefeinabgleich nicht aus!

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins aufgelistet und beschrieben. Es werden die Fehlercodes aus der "F_RELATED table (16#0170)" verwendet. Zusätzlich können die Fehlercodes des Bausteins MB_GetCyclicParameterHandle (siehe [Kap. 5.9.5 "MB_GetCyclicParameterHandle" auf Seite 133](#)) auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1180	16#0020	Fehler in der Statemachine
INPUT_RANGE_ERROR	16#1181	16#0001	HighLimit < LowLimit
INPUT_RANGE_ERROR	16#1181	16#0002	FormatLength < 0.0
INPUT_RANGE_ERROR	16#1181	16#0003	Webwidth < 0.0
INPUT_RANGE_ERROR	16#1181	16#0004	Density < 0.0
INPUT_RANGE_ERROR	16#1181	16#0005	CoreDiameter < 0.0
INPUT_RANGE_ERROR	16#1181	16#0008	MinDiameter < 0.0
INPUT_RANGE_ERROR	16#1181	16#0009	MaxDiameter < 0.0
INPUT_RANGE_ERROR	16#1181	16#000A	HighLimit < 0.0
INPUT_RANGE_ERROR	16#1181	16#000B	Lowlimit < 0.0

ErrorID	Additional1	Additional2	Text
INPUT_RANGE_ERROR	16#1181	16#000C	falsche Slave-Achs Nummer
DEVICE_ERROR	16#1182	16#0001	kein Betriebsmodus (Sercosphase 4)
RESSOURCE_ERROR	16#0004	16#0000	falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Abb.9-29: Fehlercodes Funktionsbaustein MB_DiameterMeasurementType01

Notwendige Hardware	- IndraMotion MLC Hardware CML65, CML45, CML25 - IndraDrive
Notwendige Firmware	- IndraMotion MLC Firmware FWA-CMLXX*-MLC-12VRS XX - Gerätevariante, z.B. CML65 - IndraDrive Antriebsfirmware
Notwendige Software	IndraWorks 12VRS für IndraMotion MLC bzw. IndraMotion MLD
Besonderheiten für IndraMotion MLC	Für die Wickelachse muss Interpolation im Antrieb aktiviert sein. (reale Achsen → entspr. Achse → Eigenschaften) Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Im zyklischen SERCOS-Kanal des entsprechenden Antriebs muss der Parameter P-0-0694 ("Feinabgleich Getriebeübersetzung, Prozessregler") eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter S-0-0121 ("Lastgetriebe-Eingangsumdrehungen") und S-0-0122 ("Lastgetriebe-Ausgangsumdrehungen") eingetragen werden.
Besonderheiten für IndraMotion MLD	Bei Verwendung der Steuerung IndraMotion MLD-S mit übergeordneter Steuerung IndraMotion MLC muss für die Wickelachse Interpolation im Antrieb aktiviert sein (reale Achsen → entspr. Achse → Eigenschaften). Bei Verwendung der Steuerung IndraMotion MLD-M gibt es diese Einstellung nicht. Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Im zyklischen Kanal zwischen Steuerung und Antrieb des entsprechenden Antriebs muss der Parameter P-0-0694 ("Feinabgleich Getriebeübersetzung, Prozessregler") eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. (Der Dialog ist im Projektbaum unterhalb des MLD-Knotens → AxisData.) Die "AxisData-Struktur" des Antriebes muss parametrierbar sein (P-0-1367 "SPS-Konfiguration" Bit 6 ist gesetzt, bzw. Dialog SPS-Konfiguration) Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter S-0-0121 ("Lastgetriebe-Eingangsumdrehungen") und S-0-0122 ("Lastgetriebe-Ausgangsumdrehungen") eingetragen werden.

9.5 Wickelrechner mit Tänzer

9.5.1 Einführung und Übersicht

Der Wickelrechner mit Tänzer wird an Zentrumswicklern eingesetzt. Die Zugspannung wird durch eine Kraft vorgegeben, die am frei beweglichen Teil des

RMB_TechWinder.library/MX_TechWinder.lib

Tänzers angreift. Diese Kraft kann durch eine Gewichtskraft, eine Feder oder durch Pneumatik aufgebracht werden. Die Regelung der Tänzerlage erfolgt im Tänzerlageregler. Der Istwert der Tänzerlage wird z.B. über einen Analogkanal erfasst und der Regelung zugeführt.

Der aktuelle Wickeldurchmesser wird vom intern eingebundenen Wickelrechner berechnet. Mit Hilfe des berechneten Wickeldurchmessers wird die Wickelachsgeschwindigkeit nachgeführt. Somit ersetzt der berechnete Wickeldurchmesser einen über eine Sensorik erfassten realen Istdurchmesser.

Der Wickelrechner mit Tänzer ist somit eine Kombination aus einem Wickler, dargestellt durch die Wickelachse und den Durchmesserrechner, und einem Tänzerlageregler. Der Durchmesserrechner wirkt auf den Getriebefeinabgleich und der Tänzerlageregler auf den additiven Drehzahlsollwert.

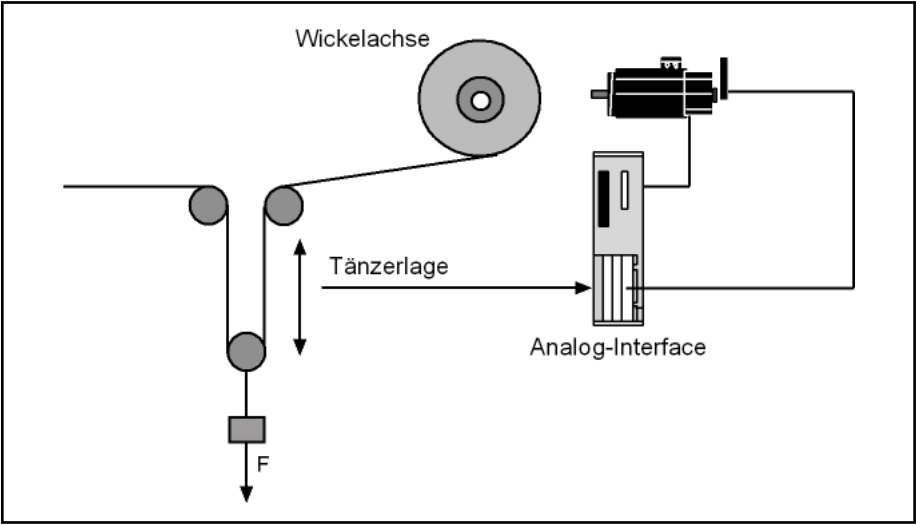


Abb.9-30: Prinzip des Wickelrechners mit Tänzerlageregler

9.5.2 MB_WinderDancerCtrlType01

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechWinder.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechWinder.lib

Abb.9-31: Referenztabelle MB_WinderDancerCtrlType01

RMB_TechWinder.library/MX_TechWinder.lib

Schnittstellenbeschreibung

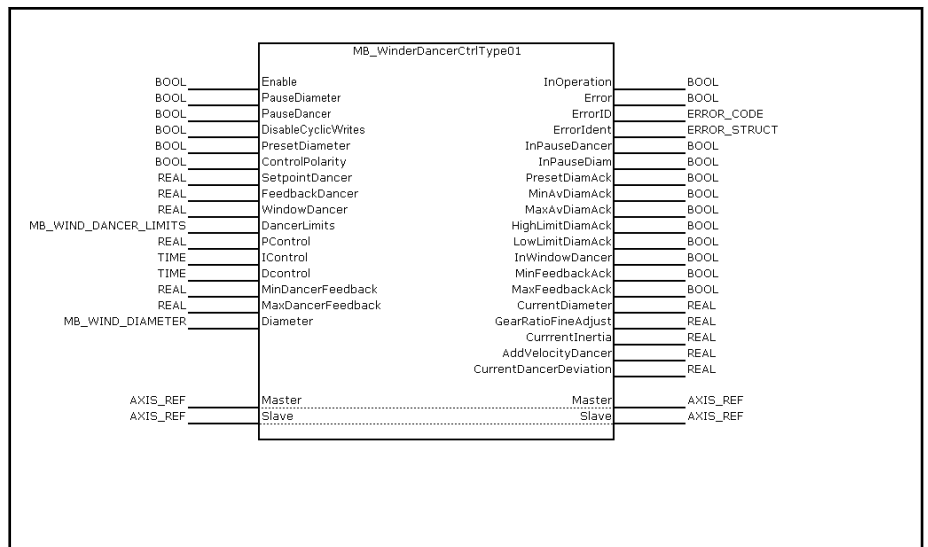


Abb.9-32: Funktionsbaustein MB_WinderDancerCtrlType01

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (Wickelachse)
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegel-gesteuert)
	PauseDiameter	BOOL	Durchmesserrechner pausiert, keine weitere Durch-messerberechnung und Geschwindigkeitsanpassung (pegelgesteuert)
	PauseDancer	BOOL	Tänzerregler pausiert (pegelgesteuert)
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	PresetDiameter	BOOL	CurrentDiameter wird auf PresetVal gesetzt, GearRa-tioFineAdj entsprechend angepasst (flankengesteuert)
	ControlPolarity	BOOL	Polarität der Tänzerregelung wird invertiert (Übernah-me bei pos. Flanke von Enable)
	SetpointDancer	REAL	Sollwert für die Tänzerregelung
	FeedbackDancer	REAL	Istwert der Tänzerlage
	WindowDancer	REAL	Fenster zur Überwachung des Tänzerlageistwertes [%] bezogen auf den Tänzerlagesollwert. Entfernt sich der Tänzer vom Lagesollwert um mehr als den ange-genenen Wert, meldet dies der Baustein durch Rück-setzen des Ausganges "InWindow"
	DancerLimits	MB_WIND_DAN-CER_LIMITS	Struktur für die automatische Stellgrößenbegrenzung des Tänzers
	PControl	REAL	Wert für die Kp-Verstärkung der Tänzerregelung
	IControl	TIME	Wert für die Nachstellzeit der Tänzerregelung
	DControl	TIME	Wert für die Vorhaltezeit der Tänzerregelung

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Typ	Kommentar
	MinDancerFeedback	REAL	untere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
	MaxDancerFeedback	REAL	obere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
	Diameter	MB_WIND_DIAMETER	Struktur für Durchmesserrechner
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	InPauseDancer	BOOL	Tänzerregler pausiert
	InPauseDiam	BOOL	Durchmesserrechner pausiert
	PresetDiamAck	BOOL	Presetdurchmesser wurde übernommen (ist gesetzt, solange Preset ansteht)
	MinAvDiamAchieved	BOOL	minimaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	MaxAvDiamAchieved	BOOL	maximaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	HighLimitDiamActive	BOOL	aktuell berechneter Durchmesser hat obere Grenze erreicht (Durchmesserwert wird verworfen)
	LowLimitDiamActive	BOOL	Aktuell berechneter Durchmesser hat untere Grenze erreicht (Durchmesserwert wird verworfen)
	InWindowDancer	BOOL	Tänzeristwert innerhalb des Tänzerfensters
	MinFeedbackAchieved	BOOL	Istwert der Tänzerlage ist unterhalb des Eingang Min-DancerFeedback
	MaxFeedbackAchieved	BOOL	Istwert der Tänzerlage ist oberhalb des Eingang Max-DancerFeedback
	CurrentDiameter	REAL	aktueller Wickeldurchmesser
	GearRatioFineAdjust	REAL	Getriebefeinabgleich zur Drehzahlachführung des Antriebes [%]
	CurrentInertia	REAL	Aktuelles Trägheitsmoment des Wickelgutes + Mechanik (bezogen auf die Motorseite) [kg m ²]
	AddVelocityDancer	REAL	Drehzahloffset für den Wickelantrieb [U/min]
	CurrentDancerDeviation	REAL	aktuelle Regelabweichung des Tänzerreglers

Abb.9-33: Schnittstellenvariablen Funktionsbaustein MB_WinderDancerCtrl-Type01

Min-/Max- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

RMB_TechWinder.library/MX_TechWinder.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
PauseDiameter	BOOL			FALSE	kontinuierlich
PauseDancer	BOOL			FALSE	kontinuierlich
DisableCyclicWrites	BOOL			FALSE	steigende Flanke an Enable
PresetDiameter	BOOL			FALSE	kontinuierlich
ControlPolarity	BOOL			FALSE	steigende Flanke an Enable
SetpointDancer	REAL	0.0	n. def.	0.0	kontinuierlich
FeedbackDancer	REAL	n. def.	n. def.	0.0	kontinuierlich
WindowDancer	REAL	0.0	n. def.	0.0	kontinuierlich
DancerLimits	STRUCT				
PControl	REAL	0.0	n. def.	0.0	kontinuierlich
IControl	TIME	0s	n. def.	0s	kontinuierlich
DControl	TIME	0s	n. def.	0s	kontinuierlich
MinDancerFeedback	REAL	0.0	MaxDancerFeedback	0.0	kontinuierlich
MaxDancerFeedback	REAL	MinDancerFeedback	n. def.	0.0	kontinuierlich
Diameter	STRUCT				je nach Strukturelement

Abb.9-34: Min-/Max- und Default-Werte der Eingänge MB_WinderDancerCtrl-Type01

MB_WIND_DIAMETER

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
PresetVal	REAL	> 0.0	n. def.	100.0	kontinuierlich	Vorgabewert für den Presetdurchmesser [mm]
CoreVal	REAL	> 0.0	n. def.	100.0	steigende Flanke an Enable	Durchmesser Hülse [mm]
CalcRange	REAL	0.0	n. def.	360.0	kontinuierlich	Bereich für Durchmesserberechnung [Grad] (lastseitig)
RepeatLength	REAL	> 0.0	n. def.	100.0	steigende Flanke an Enable	Formatlänge [mm] (1 Umdrehung der Referenzachse abgewickelt)
Webwidth	REAL	0.0	n. def.	0.0	steigende Flanke an Enable	Bahnbreite [mm] wird für das Trägheitsmoment benötigt
Density	REAL	0.0	n. def.	1.0	steigende Flanke an Enable	Materialdichte [kg/dm ³] wird für das Trägheitsmoment benötigt

RMB_TechWinder.library/MX_TechWinder.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
AverageValue	UINT	0	20	1	kontinuierlich	Anzahl der Durchmesserwerte die gemittelt werden [-]
HighLimit	REAL	LowLimit	n. def.	2000.0	kontinuierlich	obere Grenze für den Durchmesser [mm]. Auf diesen Wert wird der Durchmesser begrenzt
LowLimit	REAL	> 0.0	HighLimit	100.0	kontinuierlich	untere Grenze für den Durchmesser [mm] Der Durchmesser unterschreitet diesen Wert nicht
MinDiameter	REAL	> 0.0	n. def.	100.0	kontinuierlich	minimaler Durchmesser [mm]
MaxDiameter	REAL	> 0.0	n. def.	2000.0	kontinuierlich	maximaler Durchmesser [mm]
MechanicInertia	REAL	0.0	n. def.	0.0	kontinuierlich	Trägheitsmoment Mechanik [kg m ²] bezogen auf die Motorseite Wird für das resultierende Trägheitsmoment benötigt
DancerConstant	REAL	0.0	n. def.	0.0	kontinuierlich	Verstellung des Tänzers bezogen auf die Einheit der Tänzerverstellung [mm/-] Wird nur bei einem Wickler mit Tänzer benötigt
MeasuredVal	REAL	0.0	n. def.	0.0	kontinuierlich	gemessener Durchmesser [mm]
PivotVelocity	REAL	0.0	n. def.	0.0	kontinuierlich	Schwenkgeschwindigkeit (für Relativbewegung der Wickelachse) [mm/s]

RMB_TechWinder.library/MX_TechWinder.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
DiamCalcMode	MB_WIND_DIAM_CALC_MODE				steigende Flanke an Enable	Modus für die Durchmesserberechnung CALC_DIAM_MANUAL Durchmesserberechnung abhängig von CalcRange FIRST_REV_AUTO_CALC Durchmesserberechnung erfolgt für 1. Umdrehung alle 5 Grad, danach abhängig vom CalcRange
DiamSource	BOOL				steigende Flanke an Enable	Durchmesserermittlung: FALSE: Durchmesser wird berechnet TRUE: Durchmesser wird gemessen (MeasuredVal)

Abb.9-35: Schnittstellenbeschreibung Datenstruktur MB_WIND_DIAMETER

MB_WIND_DANCER_LIMITS

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
LowLimitControl	REAL	0.0	n. def.	0.0	kontinuierlich	unterer Grenzwert des Reglerausganges (bei Drehzahl Null) bipolar [U/min]
HighLimitControl	REAL	0.0	n. def.	0.0	kontinuierlich	oberer Grenzwert des Reglerausganges (bei Drehzahl VelocityHighLimit) bipolar [U/min]
VelocityHighLimit	REAL	0.0	n. def.	0.0	kontinuierlich	Drehzahl des Masters für HighLimitControl [U/min]

Abb.9-36: Schnittstellenbeschreibung Datenstruktur MB_WIND_DANCER_LIMITS

Funktionsbeschreibung

Durchmesserberechnung

Die Durchmesserberechnung stellt über den zyklisch aufgerufenen Funktionsbaustein MB_DiameterCalculatorType03 den aktuellen, berechneten Durchmesser zur Verfügung. Abhängig vom Durchmesser wird der additive Feinabgleich der Wickelachse eingestellt. Diese wird im Verhältnis zur Referenzachse schneller oder langsamer bewegt, um eine konstante Zugspannung der Stoffbahn zu erzielen.

Eine Relativbewegung der Tänzerwalze wird bei der Durchmesserberechnung mit Hilfe des Einganges "DancerConstant" berücksichtigt. Hierzu ist diese Tänzerkonstante einmalig zu ermitteln.

Folgende Vorgehensweise wird empfohlen:

1. Bei TänzerAuslenkung von der einen Endlage in die andere Endlage, den Bahnweg ermitteln (Einheit in mm).
2. Differenz zwischen den beiden Tänzeristwerten der jeweiligen Endlage ermitteln.

RMB_TechWinder.library/MX_TechWinder.lib

3. Quotienten der beiden Werte berechnen.
4. Vorzeichen der Tänzerkonstante durch praktischen Versuch bestimmen. Hierzu wird bei langsam drehender Wickel- u. Referenzachse die Durchmesserberechnung beobachtet. Sollwertsprünge auf den Tänzer Sollwert sollten bei richtigem Vorzeichen keinen signifikanten Einfluss auf die Durchmesserberechnung haben.
5. Berechneten Wert für die Tänzerkonstante wie im Schritt 4, wenn nötig, iterativ "nachoptimieren".



Soll der aktuelle Durchmesser, nach dem Ausschalten der Steuerung erhalten bleiben, muss der Ausgang "CurrentDiameter" auf eine remanente Variable kopiert werden. Mit diesem Wert kann nach dem Einschalten wieder ein Preset ausgeführt werden.

Rollenwechsel

Nach einem Rollenwechsel ändert sich in der Regel der Wickeldurchmesser schlagartig. Der zuletzt berechnete interne Wert für den Wickeldurchmesser entspricht dann nicht mehr dem aktuellen Durchmesser. Damit die Durchmesserberechnung korrekt aufsetzen kann, ist ein Preset des Wickeldurchmessers sinnvoll. Der Eingangswert für den Preset sollte annähernd dem tatsächlichen Wert des Wickeldurchmessers entsprechen, um einen möglichst genauen resultierenden Feinabgleich zu erzielen.



Wird der Preset nicht durchgeführt, kann es zu Fehlfunktionen kommen. Daher sollte der Preset durchgeführt werden, bevor der Durchmesserrechner wieder aktiv wird.

Bei einem fliegenden Rollenwechsel muss sichergestellt werden, dass die Umfangsgeschwindigkeit der neuen Rolle, der aktuellen Rolle entspricht. Dies ist durch ein entsprechendes SPS-Projekt in IndraLogic sicherzustellen. Beim Zuschalten des Durchmesserrechners für die neue Rolle kann mit dem Eingang "Pause" gearbeitet werden. Z.B. kann nach dem Abschlagen der alten Rolle "Pause" wieder weggenommen werden.

Tänzerlageregelung

Zur Erfassung der Tänzerposition wird ein Messglied benötigt, das die Tänzerlage in ein proportionales Signal umsetzt. Die Ist-Position des Tänzers wird von einem gegebenen Sollwert abgezogen und als Regelabweichung auf einen internen PID-Regler gegeben. Die Stellgröße des Reglers wirkt auf den Drehzahlsollwert mittels Addition. Somit dreht sich die Wickelachse im Vergleich zur Leitachse schneller oder langsamer. Hiermit wird die Zugkraft des Tänzers geregelt (vgl. Kapitel [Kap. 9.6 "Funktionsbaustein zum Wickelrechner mit Zugkraftsteuerung bzw. Zugkraftregelung"](#) auf Seite 460).

Unter normalen Bedingungen (z.B. keine hohen Drehzahlsollwertsprünge oder plötzliche Durchmesseränderungen des Wickels) erfolgt die Grobeinstellung der Umfangsgeschwindigkeit der Wickelachse über den Getriebefeinabgleich. Hierzu wird der zuvor beschriebene Durchmesserrechner eingesetzt. Der Tänzerlageregler muss somit lediglich kleine Abweichungen ausregeln.

Drehrichtung der Wickelachse, Regelsinn der Tänzerregelung

Die Drehrichtung der Zug- bzw. Wickelachse kann mit Hilfe des Antriebsparameters P-0-0108 "Polarität Leit Antrieb" bzw. des Achsparameters A-0-2798 "Polarität Leitachseposition" eingestellt werden. Dreht sich die Wickelachse nicht in die gewünschte Richtung, ist der Parameter P-0-0108 bzw. der Achsparameter A-0-2798 entsprechend umzustellen. Bei einer Invertierung der

Drehrichtung des Antriebes ist auch der Regelsinn (ControlPolarity) zu invertieren.

Funktionsdiagramm des Tänzerlagereglers

Anhand des folgenden Funktionsdiagramms wird die Funktionsweise des Wickelrechners mit Tänzer schematisch erläutert.

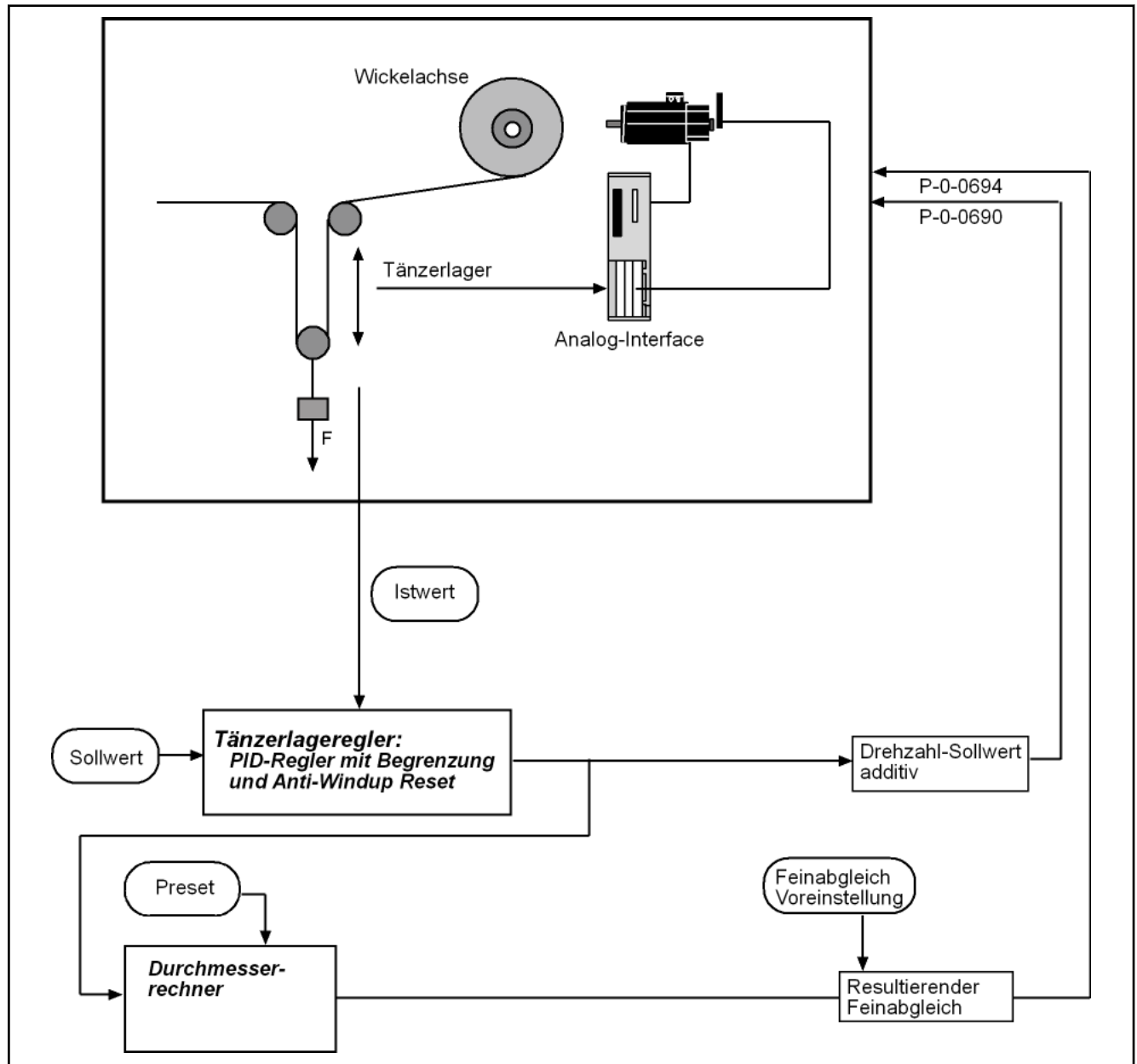


Abb.9-37: Funktionsdiagramm des Wickelrechners mit Tänzer

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins aufgelistet und beschrieben. Es werden die Fehlercodes aus der "F_RELATED table (16#0170)" verwendet. Zusätzlich können die Fehlercodes der Bausteine MB_DancerControlType03 (siehe [Kap. 9.2.2 "MB_DancerControlType03" auf Seite 428](#)), MB_DiameterCalculatorType03 (siehe [Kap. 9.3.2 "MB_DiameterCalculatorType03" auf Seite 436](#)), MB_DiameterMeasurementType01 (siehe [Kap. 9.4.2 "MB_DiameterMeasurementType01" auf Seite 445](#)), IL_PIDType01 aus der Bibliothek RIL_LoopControl (siehe Dokumentation "Rexroth IndraWorks 12VRS Basis-Bibliotheken") sowie MB_GetCyc-

RMB_TechWinder.library/MX_TechWinder.lib

licParameterHandle (siehe [Kap. 5.9.5 "MB_GetCyclicParameterHandle"](#) auf [Seite 133](#)) auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1190	16#0020	Fehler in der Statemachine

Abb.9-38: Fehlercodes Funktionsbaustein MB_WinderDancerCtrlType01

Notwendige Hardware

- IndraMotion MLC Hardware CML65, CML45, CML25
- IndraDrive

Notwendige Firmware

- IndraMotion MLC Firmware FWA-CMLXX*-MLC-12VRS
XX - Gerätevariante, z.B. CML65
- IndraDrive Antriebsfirmware

Notwendige Software

- IndraWorks 12VRS für IndraMotion MLC bzw. IndraMotion MLD

Notwendige Parametrierung für Steuerung IndraMotion MLC

Für die Wickelachse muss Interpolation im Antrieb aktiviert sein.

(reale Achsen → entspr. Achse → Eigenschaften)

Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Im zyklischen SERCOS-Kanal des entsprechenden Antriebs müssen die **Parameter P-0-0694 ("Feinabgleich Getriebeübersetzung, Prozessregler")** und **Parameter P-0-0690 ("Geschwindigkeits-Sollwert additiv, Prozessregler")** eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.

Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter S-0-0121 ("Lastgetriebe-Eingangsumdrehungen") und S-0-0122 ("Lastgetriebe-Ausgangsumdrehungen") eingetragen werden.

Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z.B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!

Notwendige Parametrierung für Steuerung IndraMotion MLD

Bei Verwendung der Steuerung IndraMotion MLD-S mit übergeordneter Steuerung IndraMotion MLC muss für die Wickelachse Interpolation im Antrieb aktiviert sein (reale Achsen → entspr. Achse → Eigenschaften). Bei Verwendung der Steuerung IndraMotion MLD-M gibt es diese Einstellung nicht.

Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Im zyklischen SERCOS-Kanal des entsprechenden Antriebs müssen die **Parameter P-0-0694 ("Feinabgleich Getriebeübersetzung, Prozessregler")** und **Parameter P-0-0690 ("Geschwindigkeits-Sollwert additiv, Prozessregler")** eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.

Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus (der Dialog ist im Projektbaum unterhalb des MLD-Knotens → AxisData).

Die "AxisData-Struktur" des Antriebes muss parametrierbar sein (P-0-1367 "SPS-Konfiguration" Bit 6 ist gesetzt, bzw. Dialog SPS-Konfiguration)

Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter S-0-0121 ("Lastgetriebe-Eingangsumdrehungen") und S-0-0122 ("Lastgetriebe-Ausgangsumdrehungen") eingetragen werden.

Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z.B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!

9.5.3 Inbetriebnahme des Wickelrechners mit Tänzer

Antriebsregelkreis Es wird vorausgesetzt, dass der Drehzahlregelkreis des Wickelantriebes eingestellt wurde. D.h. die Parameter S-0-0100 "Geschwindigkeitsregler-proportionalverstärkung" und S-0-0101 "Geschwindigkeitsregler-Nachstellzeit" müssen der Applikation entsprechend eingestellt sein.

Es wird eine Inbetriebnahme eines Wickelrechners mit Tänzer erläutert. Hierzu sind die folgenden Schritte zweckmäßig:

- Parametrierung des Tänzerlagereglers
- Parametrierung des Durchmesserrechners
- Erster Anlauf der Wickelachse

Kommandierung der Wicklerachse Für die Wicklerachse muss die richtige Betriebsart des Antriebes, sowie die Reglerfreigabe des Antriebes gesetzt werden. Dies kann auf zwei unterschiedliche Arten erfolgen:

- Setzen der Betriebsart "ModeSyncVel" mit Hilfe des Achsinterfaces (dies wird empfohlen)
- Benutzung der PLC-Open Funktionsbausteine "MC_Power", "MC_GearIn", "MC_Stop" zur Kommandierung der Achse

Parametrierung des Tänzerlagereglers

Da die Funktion des Wickelrechners mit Tänzer auf die Funktion des Tänzerlagereglers aufbaut, kann die Parametrierung des Tänzerlagereglers wie in Kapitel 9.2 [Funktionsbaustein zur Tänzerlageregelung, Seite 426](#) beschrieben, durchgeführt werden.

Parametrierung des Durchmesserrechners

Es werden, wie zuvor beschrieben, die einzelnen Eingänge des Durchmesserrechners MB_DiameterCalculatorType03 beschaltet. Hierzu sind für die Inbetriebnahme des Wickelrechners mit Tänzer einige Besonderheiten zu beachten.

Aus der Parametrierung des Tänzerlagereglers sind bereits die Eingänge "FormatLength" und "FeedBackDancer" des Durchmesserrechners vorgelegt. Eine weitere, den Tänzer betreffende Größe, ist die Tänzerkonstante "DancerConstant". Diese gibt an, welchen Bahnweg die volle Auslenkung von negativen bis positiven Anschlag, bezogen auf die Einheit der Tänzerstellung, des Tänzers beschreibt. Der Eingang "ReferenceDiameter" sollte mit dem aktuell berechneten Durchmesser "CurrentDiameter" verknüpft sein.

Da der berechnete Durchmesser über den resultierenden Feinabgleich direkt auf die Wickelachsgeschwindigkeit wirkt, ist zu Beginn der Berechnung bei Aktivierung des Wickelrechners ein sinnvoller Wert für den Durchmesser vorzugeben. Daher ist am Eingang "PresetDiameter" ein Wert für den Durchmesser anzugeben, der nahe dem des tatsächlichen Durchmessers liegt. Im Falle eines Aufwicklers kann der Wert etwas kleiner und für Abwickler etwas größer gewählt werden.

Schließlich können die Eingänge "MinDiameter" und "MaxDiameter" belegt werden. Die hier angegebenen Größen wirken auf die Ausgänge "MinAvDiamAchieved" und "MaxAvDiamAchieved". Die jeweiligen Werte sollten innerhalb des erlaubten Durchmesserbereichs liegen, der durch die Eingänge "HighLimitDiameter" und "LowLimitDiameter" definiert wird. So können z.B. für einen Rollenwechsel oder die Anpassung des "DiamCalcRange" anhand

RMB_TechWinder.library/MX_TechWinder.lib

einer Vorgabe des minimalen/maximalen Durchmessers der Zeitpunkt für bestimmte Prozessschritte in der SPS detektiert und angezeigt werden.



Das Verhältnis "Formatlänge" / "CoreDiameter" darf nicht größer als 25 sein.

Ansonsten reicht die Getriebearpassung über den Getriebefeinabgleich nicht aus!

Erster Anlauf der Wickelachse

Der Tänzerlageregler und der Wickelrechner werden in dieser Reihenfolge nacheinander aktiviert. Nach der Aktivierung der Tänzerlageregelung läuft der Tänzer in seine Sollposition. Wie zuvor beschrieben, sollte diese in etwa der Mittellage des Tänzers entsprechen.

Die Winkellagen von Wickel- und Referenzachse werden aufsummiert und abhängig vom Wert des Eingangs "DiamCalcRange" wird regelmäßig der Durchmesser berechnet und als aktueller Durchmesserwert am Ausgang "CurrentDiameter" angezeigt. Nach dem Einschalten des Durchmesserrechners erscheint an "CurrentDiameter" zunächst der Wert Null, bis das erste Mal ein Durchmesser berechnet worden ist. Soll ein anderer Wert als Null verwendet werden, muss ein Preset durchgeführt werden. Der Wert "PresetDiameter" wird bei einer positiven Flanke an "Preset" als aktueller Durchmesserwert übernommen.

Die Dauer zwischen zwei Berechnungen ist abhängig von der Wickelachsdrehzahl und dem Wert im Eingang "DiamCalcRange".

Der Wickelrechner wirkt auf den Getriebefeinabgleich. Der entsprechende Wert wird am Ausgang "GearRatioFineAdjust" ausgegeben.

9.6 Funktionsbaustein zum Wickelrechner mit Zugkraftsteuerung bzw. Zugkraftregelung

9.6.1 Einführung und Übersicht

Der Funktionsbaustein stellt zwei verschiedene Wickelrechner zur Verfügung:

- Wickelrechner mit Zugkraftsteuerung
 - entspricht einem sensorlosen Wickelrechner
- Wickelrechner mit Zugkraftregelung
 - entspricht einem Wickelrechner mit Kraftmessdose

Die Wickelrechner sind im Funktionsbaustein MB_WinderTensionCtrlType02 implementiert.

Der Funktionsbaustein berechnet zyklisch den aktuellen Wickeldurchmesser, das aktuelle Trägheitsmoment des Wickels, die entsprechende Geschwindigkeitsnachführung über den Getriebefeinabgleich, einen additiven Drehzahloffset zum Spannen der Stoffbahn und die erforderliche Drehmomentgrenze.



Bei Anwendung dieses Wickelkonzeptes sollte die mechanische Reibung (Motorankopplung, Getriebe...) so gering wie möglich gehalten werden. Die mechanische Reibung sollte wenn möglich immer kleiner sein als das kleinste Zugmoment, das sich bei der minimalen Zugspannung und dem minimalen Wickeldurchmesser ergibt.

Wickelrechner mit Zugkraftsteuerung

Der Wickelrechner mit Zugkraftsteuerung wird an Zentrumswicklern eingesetzt. Der Antrieb der Wickelachse arbeitet generell an seiner Drehmomentgrenze. Die Drehmomentgrenze ist abhängig vom aktuellen Wickeldurchmesser und der eingestellten Zugspannung.

Die Zugkraft wird durch Einstellung des Drehmoments der Wickelachse erzeugt. Zusätzliche Messglieder werden nicht benötigt.

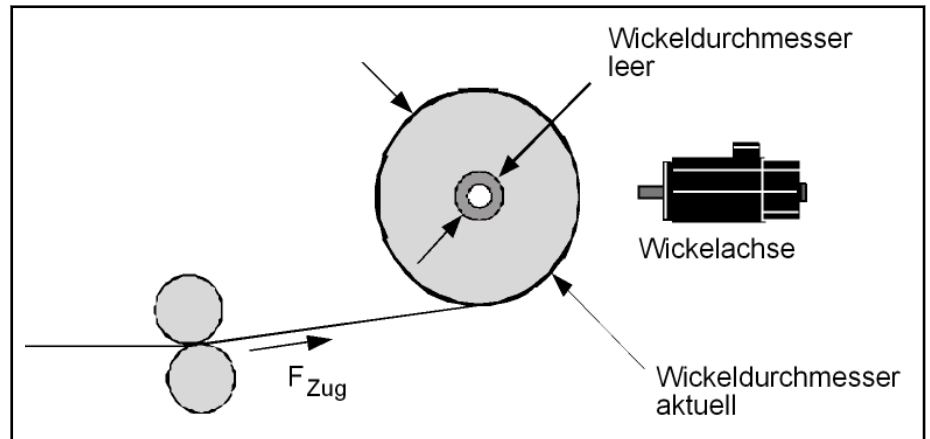


Abb.9-39: Wickelrechner mit Zugkraftsteuerung

Wickelrechner mit Zugkraftregelung

Der Wickelrechner mit Zugkraftregelung wird an Zentrumswicklern eingesetzt. Der Antrieb der Wickelachse arbeitet generell an seiner Drehmomentgrenze. Die Drehmomentgrenze ist abhängig vom aktuellen Wickeldurchmesser und der eingestellten Zugspannung. Die Drehmomentgrenze entspricht dem Wert des Motormoments, das benötigt wird, um den Wickel mit der vorgegebenen Drehzahl zu drehen. Es setzt sich aus dem Last-, dem Reib- und dem Beschleunigungsmoment zusammen.

Der Zugspannungsregler wird als PI-Regler realisiert. Dieser beinhaltet den Funktionsbaustein IL_PIDType01 aus der Bibliothek RIL_LoopControl. Die Stellgröße des Reglers und der aktuelle Wickeldurchmesser bestimmen den Drehmomentgrenzwert des Antriebs. Der Sollwert der Zugspannung sollte zusätzlich gewichtet auf den Reglerausgang gegeben werden (PreControl-Weighting ungleich Null). Bei einer Wichtung ungleich Null, besteht die Zugspannung für die Drehmomentbildung aus Reglerausgangsgröße addiert mit dem gewichteten Sollwert. Wichtung Null bedeutet, dass nur der Zugspannungsregler wirkt. Der Zugspannungsregler sollte daher nur additiv arbeiten (z.B. mit der Wichtung 1.0).

Der Zugspannungswert wird direkt mittels einer Kraftmessdose gemessen und über einen Analogkanal oder über einen Feldbus dem Zugspannungsregler als Istwert zugeführt. Bei gespannter Bahn kann der Durchmesserrechner den aktuellen Wickeldurchmesser berechnen.

RMB_TechWinder.library/MX_TechWinder.lib

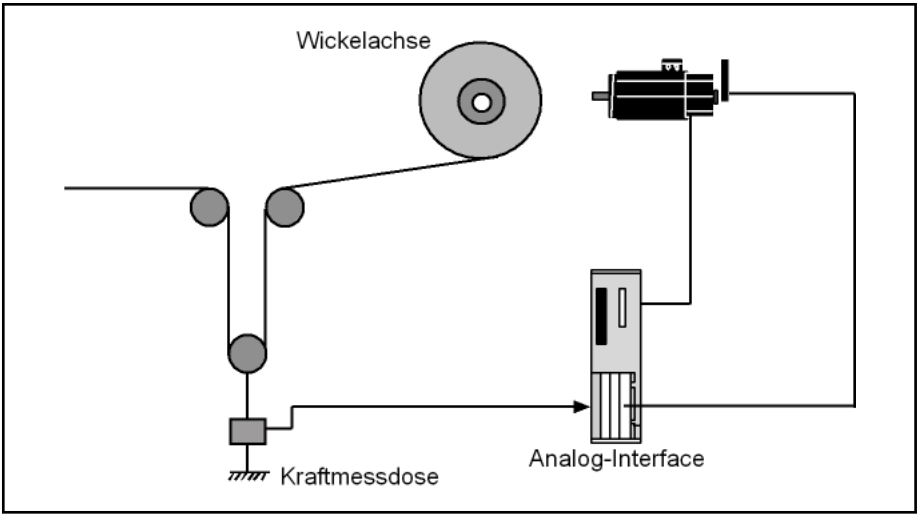


Abb.9-40: Wickelrechner mit Zugkraftregelung

9.6.2 MB_WinderTensionCtrlType02

Kurzbeschreibung

Der Funktionsbaustein berechnet zyklisch den aktuellen Wickeldurchmesser, das aktuelle Trägheitsmoment des Wickels, die entsprechende Geschwindigkeitsnachführung über den Getriebefeinabgleich, einen additiven Drehzahloffset zum Spannen der Bahn und die erforderliche Drehmomentgrenze.

Zielgrößen des Funktionsbausteins sind die Parameter P-0-0694 "Feinabgleich Getriebeübersetzung, Prozessregler" P-0-0690 "Geschwindigkeits-Sollwert additiv, Prozessregler" und S-0-0092 "Drehmoment/Kraft-Grenzwert bipolar". Diese Parameter müssen in den zyklischen Datencontainern der IndraMotion MLC eingebunden werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechWinder.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechWinder.lib

Abb.9-41: Referenztabelle MB_WinderTensionCtrlType02

Schnittstellenbeschreibung

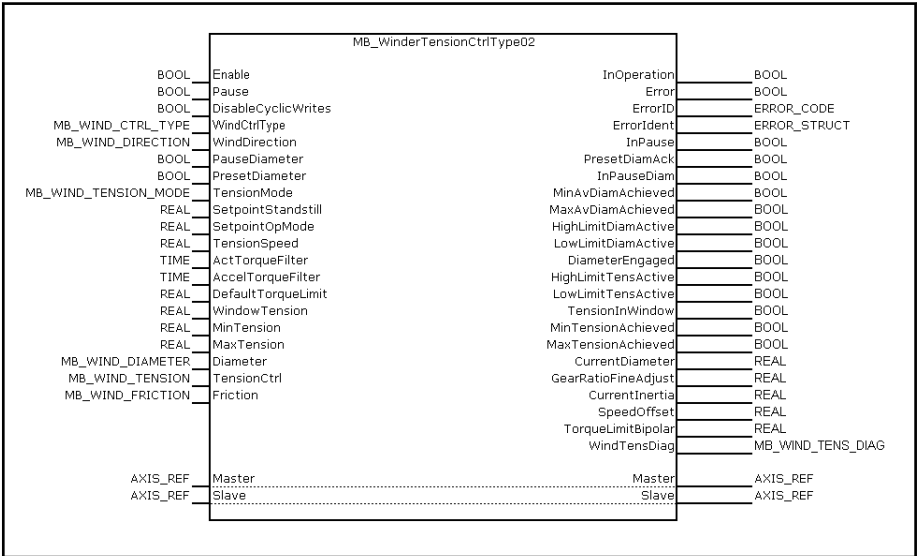


Abb.9-42: Funktionsbaustein MB_WinderTensionCtrlType02

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (Wickelachse)
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegel-gesteuert)
	Pause	BOOL	Pausieren des Zugspannungsreglers und des Wickel-rechners (Werte werden eingefroren)
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	WindCtrlType	MB_WIND_CTRL_TY- PE	Wickelstruktur Zugkraftsteuerung (sensorlos) OPEN_LOOP (16#0000) Zugkraftregelung (Messdose) CLOSED_LOOP (16#0001)
	WindDirection	MB_WIND_DIRECTI- ON	Wickelrichtung Aufwickler REWIND (16#0000) Abwickler UNWIND (16#0001)
	PauseDiameter	BOOL	Anhalten der Durchmesserberechnung Regelung/Steuerung der Drehmomentgrenze bleibt in Betrieb
	PresetDiameter	BOOL	CurrentDiameter wird auf PresetVal gesetzt, GearRa-tioFineAdj entsprechend angepasst (flankengesteuert)
	TensionMode	MB_WIND_TENSI- ON_MODE	Modus zum Spannen der Materialbahn Spannen mit konstanter Drehzahl CONST_SPEED Spannen mit konstanter Bahngeschwindigkeit CONST_WEB_VEL
	SetpointStandstill	REAL	Vorgabe des Stillstandszuges [N] (Umschaltung zwischen Stillstands- und Laufzug an-hand der Stillstandsmeldung des Antriebes)
	SetpointOpMode	REAL	Vorgabe des Laufzuges [N] (Umschaltung zwischen Stillstands- und Laufzug an-hand der Stillstandsmeldung des Antriebes)
	TensionSpeed	REAL	Spanndrehzahl bei leerem Wickel [min ⁻¹]
	ActTorqueFilter	TIME	Filterzeitkonstante für das Istdrehmoment S-0-0084 "Drehmoment-/Kraft-Istwert"
	AccelTorqueFilter	TIME	Filterzeitkonstante für das Beschleunigungsmoment
	DefaultTorqueLimit	REAL	Wert für die bipolare Drehmomentgrenze nach Weg-nahme von "Enable" [%]
	WindowTension	REAL	Fenster für den Zugspannungsiswert [%], bezogen auf den Zugspannungssollwert (Ausgang "TensionIn-Window")

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Typ	Kommentar
	MinTension	REAL	untere Schwelle für den Zugspannungswert (nur für die Anzeige mit binärem Ausgang "MinTensionAck")
	MaxTension	REAL	obere Schwelle für den Zugspannungswert (nur für die Anzeige mit binärem Ausgang "MaxTensionAck")
	Diameter	MB_WIND_DIAMETER	Struktur für Durchmesserrechner
	TensionCtrl	MB_WIND_TENSION	Struktur für Zugspannungsregler
	Friction	MB_WIND_FRICTION	Struktur für Reibmoment
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	InPause	BOOL	Zugspannungsregler und Wickelrechner pausieren
	PresetDiamAck	BOOL	Presetdurchmesser wurde übernommen (ist gesetzt, solange Preset ansteht)
	InPauseDiam	BOOL	Pause Durchmesserrechner
	MinAvDiamAchieved	BOOL	minimaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	MaxAvDiamAchieved	BOOL	maximaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	HighLimitDiamActive	BOOL	Aktuell berechneter Durchmesser hat obere Grenze erreicht (Durchmesserwert wird verworfen)
	LowLimitDiamActive	BOOL	Aktuell berechneter Durchmesser hat untere Grenze erreicht (Durchmesserwert wird verworfen)
	DiameterEngaged	BOOL	Durchmesser "eingerastet" TRUE: eingerastet FALSE: ausgerastet
	HighLimitTensActive	BOOL	Ausgang des Zugspannungsreglers hat obere Grenze erreicht
	LowLimitTensActive	BOOL	Ausgang des Zugspannungsreglers hat untere Grenze erreicht
	TensionInWindow	BOOL	Zugspannungswert ist innerhalb des Zugspannungsfensters
	MinTensionAchieved	BOOL	Zugspannung unterhalb des Einganges "MinTension"
	MaxTensionAchieved	BOOL	Zugspannung oberhalb des Einganges "MaxTension"
	CurrentDiameter	REAL	aktuell berechneter Durchmesser [mm]
	GearRatioFineAdjust	REAL	Resultierender Getriebefineabgleich zur Drehzahl-nachführung des Antriebes [%]
	CurrentInertia	REAL	aktuelles Trägheitsmoment des Wickelgutes + Mechanik (bezogen auf die Motorseite) [kg m²]
	SpeedOffset	REAL	Spanndrehzahl [U/min]

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Typ	Kommentar
	TorqueLimitBipolar	REAL	Bipolare Drehmomentengrenze [%]
	WindTensDiag	MB_WIND_TENS_DIAG	aktuelle Regelabweichung des Tänzerreglers

Abb.9-43: Schnittstellenvariablen Funktionsbaustein MB_WinderTensionCtrl-Type02

Min-/Max und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Pause	BOOL			FALSE	kontinuierlich
DisableCyclicWrites	BOOL			FALSE	steigende Flanke an Enable
WindCtrlType	MB_WIND_CTRL_TYPE				steigende Flanke an Enable
WindDirection	MB_WIND_DIRECTION				steigende Flanke an Enable
PauseDiameter	BOOL			FALSE	kontinuierlich
PresetDiameter	REAL			FALSE	kontinuierlich
TensionMode	MB_WIND_TENSION_MODE				steigende Flanke an Enable
SetpointStandstill	REAL	0.0	n. def.	0.0	kontinuierlich
SetpointOpMode	REAL	n. def.	n. def.	0.0	kontinuierlich
TensionSpeed	REAL	0.0	n. def.	0.0	kontinuierlich
ActTorqueFilter	TIME	0s	n. def.	0s	kontinuierlich
AccelTorqueFilter	TIME	0s	n. def.	0s	kontinuierlich
DefaultTorqueLimit	REAL	0.0	n. def.	100.0	kontinuierlich
WindowTension	REAL	0.0	n. def.	0.0	kontinuierlich
MinTension	REAL	0.0	n. def.	0.0	kontinuierlich
MaxTension	REAL	0.0	n. def.	0.0	kontinuierlich
Diameter	MB_WIND_DIAMETER				je nach Strukturelement
TensionCtrl	MB_WIND_TENSION				kontinuierlich
Friction	MB_WIND_FRICTION				kontinuierlich

Abb.9-44: Min-/Max- und Default-Werte der Eingänge der Eingänge MB_WinderTensionCtrlType02

MB_WIND_CTRL_TYPE

RMB_TechWinder.library/MX_TechWinder.lib

Name	Wert	Beschreibung
OPEN_LOOP	0	Wickeltyp "open loop", sensorlos
CLOSED_LOOP	1	Wickeltyp "closed loop", mit Messdose

Abb.9-45: Elemente des Aufzählungstyps MB_WIND_CTRL_TYPE

MB_WIND_DIRECTION

Name	Wert	Beschreibung
REWIND	0	Aufwickler
UNWIND	1	Abwickler

Abb.9-46: Elemente des Aufzählungstyps MB_WIND_DIRECTION

MB_WIND_TENSION_MODE

Name	Wert	Beschreibung
CONST_SPEED	0	Spannverfahren: konstante Spanndrehzahl
CONST_WEB_VEL	1	Spannverfahren: konstante Bahngeschwindigkeit

Abb.9-47: Elemente des Aufzählungstyps MB_WIND_TENSION_MODE

MB_WIND_DIAMETER

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
PresetVal	REAL	> 0.0	n. def.	100.0	kontinuierlich	Vorgabewert für den Presetdurchmesser [mm]
CoreVal	REAL	> 0.0	n. def.	100.0	steigende Flanke an Enable	Durchmesser Hülse [mm]
CalcRange	REAL	0.0	n. def.	360.0	kontinuierlich	Bereich für Durchmesserberechnung [Grad] (lastseitig)
RepeatLength	REAL	> 0.0	n. def.	100.0	steigende Flanke an Enable	Formatlänge [mm] (1 Umdrehung der Referenzachse abgewickelt)
Webwidth	REAL	0.0	n. def.	0.0	steigende Flanke an Enable	Bahnbreite [mm] (wird für das Trägheitsmoment benötigt)
Density	REAL	0.0	n. def.	1.0	steigende Flanke an Enable	Materialdichte [kg/dm ³] (wird für das Trägheitsmoment benötigt)
AverageValue	UINT	0	20	1	kontinuierlich	Anzahl der Durchmesserwerte die gemittelt werden [-]
HighLimit	REAL	LowLimit	n. def.	2000.0	kontinuierlich	obere Grenze für den Durchmesser [mm] Der Durchmesser wird auf diesen Wert begrenzt

RMB_TechWinder.library/MX_TechWinder.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
LowLimit	REAL	> 0.0	HighLimit	100.0	kontinuierlich	untere Grenze für den Durchmesser [mm] Der Durchmesser unterschreitet diesen Wert nicht
MinDiameter	REAL	> 0.0	n. def.	100.0	kontinuierlich	minimaler Durchmesser [mm]
MaxDiameter	REAL	> 0.0	n. def.	2000.0	kontinuierlich	maximaler Durchmesser [mm]
MechanicInertia	REAL	0.0	n. def.	0.0	kontinuierlich	Trägheitsmoment Mechanik [kg m ²] bezogen auf die Motorseite (wird für das resultierende Trägheitsmoment benötigt)
DancerConstant (wird nur bei einem Wickler mit Tänzer benötigt)	REAL	0.0	n. def.	0.0	kontinuierlich	Verstellung des Tänzers bezogen auf die Einheit der Tänzerverstellung [mm/-] (wird nur bei einem Wickler mit Tänzer benötigt)
MeasuredVal	REAL	0.0	n. def.	0.0	kontinuierlich	gemessener Durchmesser [mm]
PivotVelocity	REAL	0.0	n. def.	0.0	kontinuierlich	Schwenkgeschwindigkeit (für Relativbewegung der Wickelachse) [mm/s]
DiamCalcMode	MB_WIND_DIAM_CALC_MODE				steigende Flanke an Enable	Modus für die Durchmesserberechnung CALC_DIAM_MANUAL Durchmesserberechnung abhängig von CalcRange FIRST_REV_AUTO_CALC Durchmesserberechnung erfolgt für 1. Umdrehung alle 5 Grad, danach abhängig vom CalcRange
DiamSource	BOOL				steigende Flanke an Enable	Durchmesserermittlung: FALSE: Durchmesser wird berechnet TRUE: Durchmesser wird gemessen (MeasuredVal)

Abb.9-48:

Schnittstellenbeschreibung Datenstruktur MB_WIND_DIAMETER

MB_WIND_TENSION

RMB_TechWinder.library/MX_TechWinder.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
PControl	REAL	0.0	n. def.	0.0	kontinuierlich	Kp-Verstärkung des Zugspannungsreglers
IControl	REAL	0.0	n. def.	0.0	kontinuierlich	Tn Nachstellzeit des Zugspannungsreglers
FeedbackVal	REAL	0.0	n. def.	0.0	kontinuierlich	Istwert der Zugspannung
PreControlWeighting	REAL	0.0	n. def.	1.0	kontinuierlich	Wichtung der Sollwertaufschaltung
HighLimit	REAL	0.0	n. def.	0.0	kontinuierlich	obere Grenze des Zugspannungsreglers
LowLimit	REAL	0.0	n. def.	0.0	kontinuierlich	untere Grenze des Zugspannungsreglers

Abb.9-49: Schnittstellenbeschreibung Datenstruktur MB_WIND_TENSION

MB_WIND_FRICTION

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
StandstillTorque	REAL	0.0	n. def.	0.0	kontinuierlich	Reibmoment im Stillstand
MaxTorque	REAL	0.0	n. def.	0.0	kontinuierlich	Reibmoment bei S-0-0091 "Geschwindigkeitsgrenzwert bipolar"
MinTorque	REAL	0.0	n. def.	0.0	kontinuierlich	minimales Reibmoment
MinTorqueVelocity	REAL	0.0	n. def.	0.0	kontinuierlich	Geschwindigkeit beim minimalen Reibmoment

Abb.9-50: Schnittstellenbeschreibung Datenstruktur MB_WIND_FRICTION

MB_WIND_TENS_DIAG

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
TensionCtrlOutput	REAL	n. def.	n. def.	0.0	kontinuierlich	Ausgang des Zugspannungsreglers [N]
TorqueLimit	REAL	n. def.	n. def.	0.0	kontinuierlich	vorzeichenbehaftetes Torquelimit [%]
AccelerationTorque	REAL	n. def.	n. def.	0.0	kontinuierlich	aktuelles Beschleunigungsmoment [%]
FrictionTorque	REAL	n. def.	n. def.	0.0	kontinuierlich	aktuelles Reibmoment [%]

Abb.9-51: Schnittstellenbeschreibung Datenstruktur MB_WIND_TENS_DIAG

Signal-Zeit-Diagramm

Im folgenden Signal-Zeit-Diagramm sind ausgewählte Signale des Funktionsbausteins dargestellt. Mit einer positiven Flanke am Eingang "Enable", der am Ausgang "InOperation" signalisiert wird, wird der Baustein in Betrieb genommen. Die Betriebsbereitschaft kann durch einen Fehler, der am Ausgang "Error" bestätigt wird, unterbrochen werden. Der interne Reglerprozess kann mit dem Eingang "Pause" angehalten werden. "Pause" wird am Ausgang "In-Pause" signalisiert. Das Belegen des Ausgangs "CurrentDiameter" mit einem Vorgabewert (Eingang "PresetVal" in der Struktur MB_WIND_DIAMETER) wird durch eine positive Flanke am Eingang "PresetDiameter" unabhängig vom Eingang "Pause" vorgenommen. Es ist jedoch empfehlenswert, den Funktionsbaustein in "Pause" zu schalten. Dadurch lässt sich der Durchmes-

RMB_TechWinder.library/MX_TechWinder.lib

ser-Preset kontrollierter und übersichtlicher durchführen. Das Setzen des Durchmessers wird am Ausgang "PresetDiamAck" angezeigt.

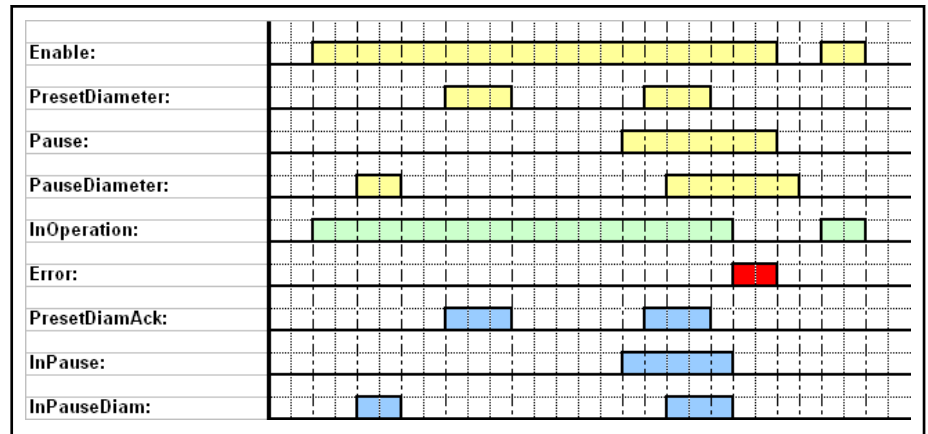


Abb.9-52: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_WinderTensionCtrlType01

Funktionsbeschreibung

Wickelrechner mit Zugkraftsteuerung

Der Funktionsumfang des Wickelrechners mit Zugkraftsteuerung beinhaltet:

- Spannen des ungespannten Wickelgutes
- Drehzahlabhängige Reibkompensation
- Kompensation des Beschleunigungsmoments
- Getrennte Parametrierung der Laufzugkraft und der Stillstandszugkraft und
- Berechnung durchmesserabhängiger Größen
(mit Funktionsbaustein MB_DiameterCalculatorType03)

Funktionsdiagramm des Wickelrechners mit Zugkraftsteuerung

Anhand des folgenden Funktionsdiagramms wird die Funktionsweise des Wickelrechners mit Zugkraftsteuerung schematisch erläutert.

RMB_TechWinder.library/MX_TechWinder.lib

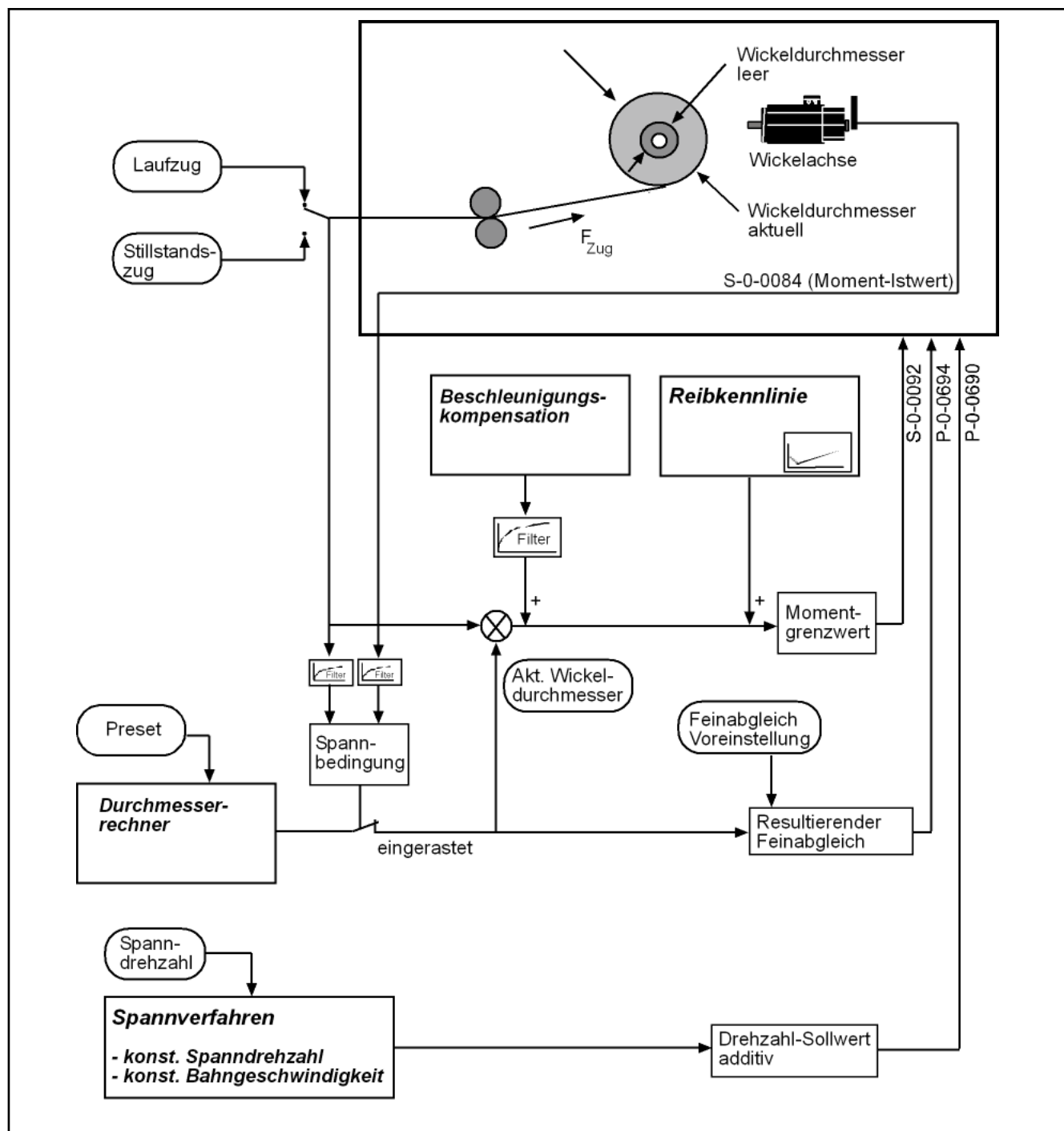


Abb.9-53: Funktionsdiagramm des Wicklers mit Zugkraftsteuerung

Wickelrechner mit Zugkraftregelung

Der Funktionsumfang des Wickelrechners mit Zugkraftregelung beinhaltet:

- Spannen des ungespannten Wickelgutes
- Drehzahlabhängige Reibkompensation
- Kompensation des Beschleunigungsmomentes
- Getrennte Parametrierung der Laufzugkraft und der Stillstandszugkraft
- PI-Regler für die Zugspannung (über IL_PIDType01) und

- Berechnung durchmesserabhängiger Größen (über MB_DiameterCalculatorType02)

Funktionsdiagramm des Wickelrechners mit Zugkraftregelung

Anhand des folgenden Funktionsdiagramms wird die Funktionsweise des Wickelrechners mit Zugkraftregelung schematisch erläutert.

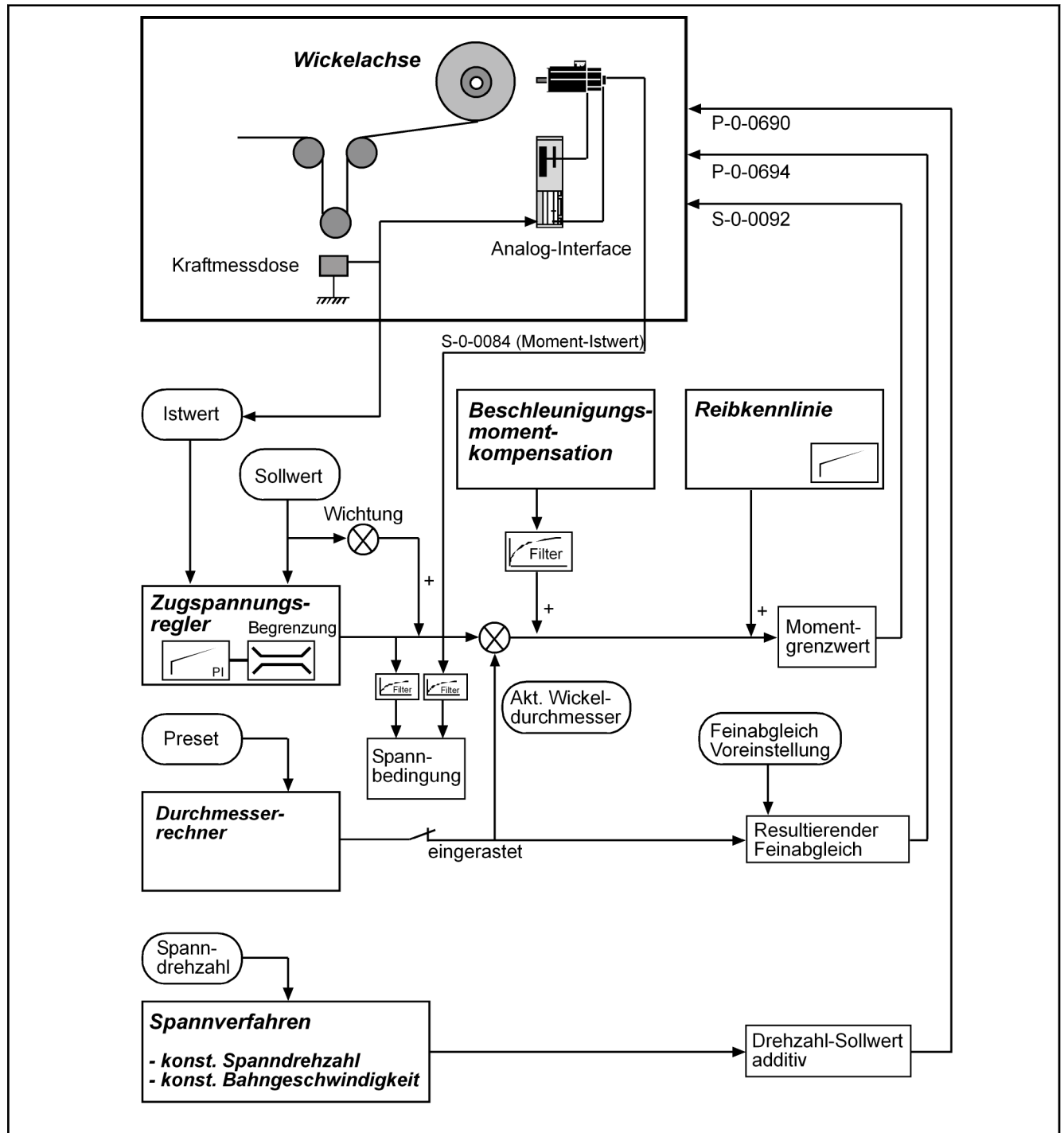


Abb.9-54: Funktionsdiagramm des Wicklers mit Zugkraftregelung
Drehrichtung des Wickelantriebes, Auf-/Abwickler

RMB_TechWinder.library/MX_TechWinder.lib

Mit Hilfe eines Eingangs des Funktionsbausteins wird der Wicklertyp Auf- bzw. Abwickler eingestellt. Die Drehrichtung des Antriebs ist entweder über die Wichtungparameter oder über die Polarität vom Leitantrieb einzustellen.



Die Drehrichtung der Masterachse des Wickelantriebes (d.h. die Masterachse des Bausteins MC_GearIn bzw. Masterachse des Achs-Interface) muss positiv sein, d.h. positiver Drehzahlsollwert. Eine Invertierung des Wickelantriebes muss über die Polaritätenparameter erfolgen.

Durchmesserberechnung

Das Drehmoment für den Wickelantrieb ergibt sich aus dem Produkt von Zugkraft und Wickelradius. Daher muss der Radius bzw. der Durchmesser des Wickels laufend bestimmt werden. Im Funktionsbaustein ist daher ein interner Durchmesserrechner (Funktionsbaustein MB_DiameterCalculator-Type03, siehe [Kap. 9.3.2 "MB_DiameterCalculatorType03" auf Seite 436](#)) eingebunden.

Die Durchmesserberechnung wird erst bei gespanntem Wickelgut und keiner Stillstandsmeldung der Referenzachse aktiv und wird zyklisch in Abhängigkeit von der Wickelachsdrehzahl bzw. -lage durchgeführt. Zu Diagnosezwecken kann der aktuelle Durchmesser am Funktionsbaustein ausgelesen werden.

Der berechnete Wickeldurchmesser wirkt direkt auf den resultierenden Feinabgleich des Getriebes. Wird die Bahnspannung zu gering oder negativ (Schlaufenbildung), setzt der Durchmesserrechner aus, da dann keine sinnvolle Durchmesserberechnung mehr möglich ist. Liegt das Istmoment des Antriebs 25% unter dem Sollwert, rechnet der Durchmesserrechner nicht weiter (Einrastbedingung).



Soll der aktuelle Durchmesser, nach dem Ausschalten der Steuerung erhalten bleiben, muss der Ausgang "CurrentDiameter" auf eine remanente Variable kopiert werden. Mit diesem Wert kann nach dem Einschalten wieder ein Preset ausgeführt werden.



Das Verhältnis "Formatlänge" / "CoreDiameter" darf nicht größer als 25 sein.

Ansonsten reicht die Getriebeanpassung über den Getriebefeinabgleich nicht aus!

Spannen des Wickelgutes

Bei Freigabe des Wickelrechners ("Enable" = TRUE) wird das Wickelgut zunächst mit der Spanndrehzahl gespannt. Um eine Schlaufenbildung im Wickelgut zu vermeiden, sollte die Leitachse nicht drehen. Das Wickelgut gilt als gespannt, wenn der Drehmoment-Sollwert des Antriebs eine vorgegebene Grenze überschreitet und die relative Abweichung zwischen Drehmoment-Sollwert und -Istwert kleiner einer bestimmten Schranke ist.

Für das Spannen des Wickelgutes kann zwischen zwei verschiedenen Verfahren gewählt werden:

- Spannen mit konstanter Drehzahl und
- Spannen mit konstanter Bahngeschwindigkeit.

Beim Spannen mit konstanter Drehzahl wird die Spanndrehzahl konstant gehalten. Dadurch steigt mit zunehmendem Wickeldurchmesser die Bahngeschwindigkeit an.

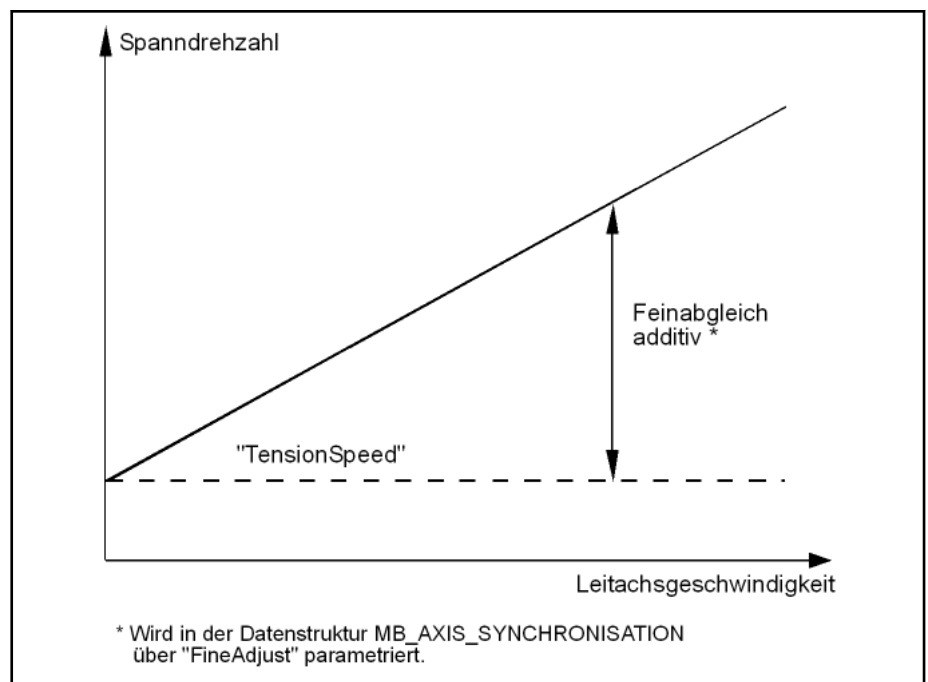


Abb.9-55: Spannen mit konstanter Spanndrehzahl

Beim Spannen mit konstanter Bahngeschwindigkeit sinkt dagegen die Spanndrehzahl reziprok zum Wickeldurchmesser. Die Berechnung der Wickelachsdrehzahl bezieht sich auf die Spanndrehzahl bei leerem Wickel. Diese muss entsprechend dem Funktionsbaustein übergeben werden.

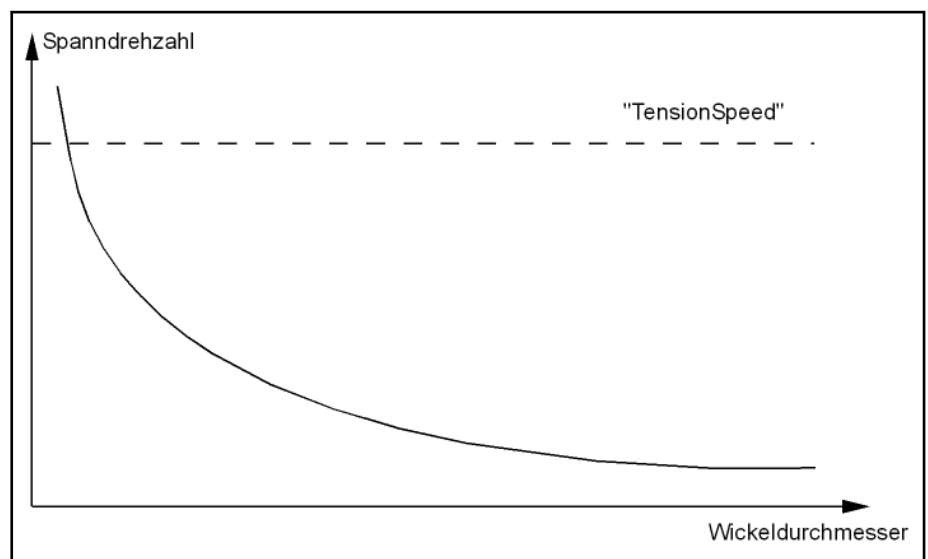


Abb.9-56: Spannen mit konstanter Bahngeschwindigkeit

Reibkompensation

Mit Hilfe verschiedener Eingänge wird eine Reibmomentkompensation umgesetzt. Die Annäherung des Reibungsverhaltens ist wie folgt definiert:

Die Reibung nimmt vom Stillstandsreibmoment zunächst bis zu einer bestimmten Drehzahl ab. Danach steigt das Reibmoment linear mit der Wickelachsdrehzahl bis zur maximalen Drehzahl (S-0-0091 "Geschwindigkeits-Grenzwert, bipolar", bzw. A-0-0032 "Geschwindigkeits-Grenzwert positiv", A-0-0033 "Geschwindigkeits-Grenzwert negativ") an. Für negative Drehzahlen gilt das Verhalten analog. Die Kurve wird an der y-Achse (Reibmoment)

RMB_TechWinder.library/MX_TechWinder.lib

gespiegelt. Mit Hilfe dieses Reibmomentverhaltens, kann dem Effekt der Reibung durch eine Kompensation entgegen gewirkt werden.

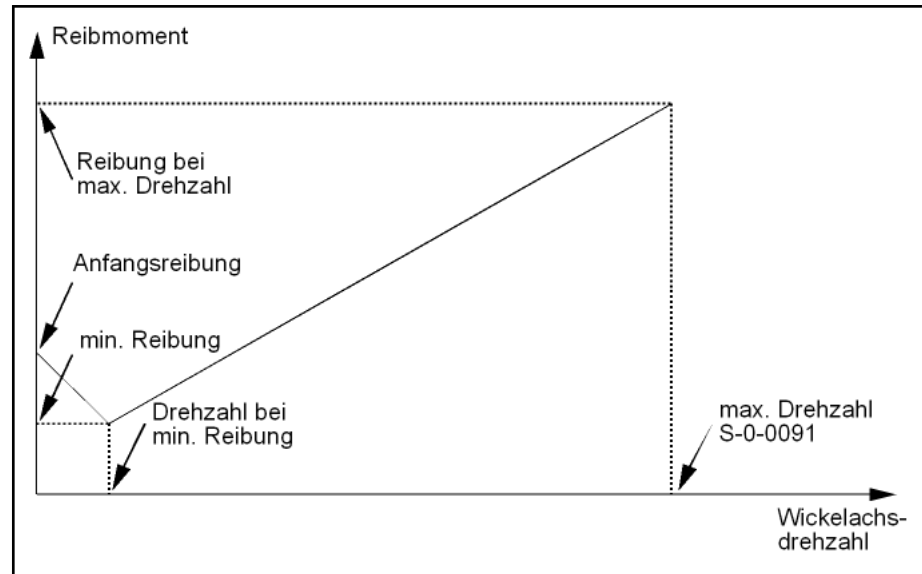


Abb.9-57: Reibmomentkompensation

Kompensation des Beschleunigungsmoments

Für die Beschleunigung der Wickelachse ist ein zusätzliches Beschleunigungsmoment erforderlich. Dieses ist abhängig vom Wickeldurchmesser. Mit der Angabe von Material- und Stoffbahndaten wird eine durchmesserabhängige Trägheit des Wickels berechnet. Das berechnete Beschleunigungsmoment kann zudem gefiltert werden.

Rollenwechsel

Nach einem Rollenwechsel ändert sich in der Regel der Wickeldurchmesser schlagartig. Der zuletzt berechnete interne Wert für den Wickeldurchmesser entspricht dann nicht mehr dem aktuellen Durchmesser. Damit die Durchmesserberechnung korrekt aufsetzen kann, ist ein Preset des Wickeldurchmessers erforderlich. Der Eingangswert für den Preset sollte annähernd dem tatsächlichen Wert des Wickeldurchmessers entsprechen.



Wird der Preset nicht durchgeführt, kann es zu Fehlfunktionen kommen. Daher sollte der Preset abgeschlossen sein, bevor der Durchmesserrechner wieder aktiv wird.

Der Wert für den Preset sollte wie folgt gewählt werden:

- beim Aufwickler etwas kleiner bzw. gleich dem tatsächlichen Wickeldurchmesser und
- beim Abwickler etwas größer bzw. gleich dem tatsächlichen Wickeldurchmesser

Bei einem fliegenden Rollenwechsel muss sichergestellt werden, dass die Umfangsgeschwindigkeit der neuen Rolle, der aktuellen Rolle entspricht. Dies sollte entsprechend im SPS-Projekt in IndraLogic umgesetzt werden.

Filterung des Momentenistwerts

Bei niedrigen Zugspannungen kann ein ungefilterter Momentenwert zu einem Ausrasten des Wickelrechners führen. Daher sollte eine Filterung vorgenommen und der entsprechende Eingang beschaltet werden. Hierzu werden Ist- und Sollmoment des Antriebs miteinander verglichen (z.B. 20ms).

Zusätzliche Vor- bzw. Nacheilung der Wickelachse

Eine zusätzliche Vor- bzw. Nacheilung über den Getriebefeinabgleich (Voreinstellung Feinabgleich "FineAdjust") kann über die Strukturvariable "qrAxisCtrl[AxisNummer].Admin.SyncMode.FineAdjust" der Datenstruktur MB_AXIS_SYNCHRONISATION der Bibliothek ML_TechInterface eingestellt werden. Bei einem Aufwickler sollte ein kleiner positiver Wert (z.B. 5%), bei einem Abwickler ein kleiner negativer Wert eingestellt werden. Dies führt dann bei der jeweiligen Wickelrichtung zu einem besseren Anlaufverhalten.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins aufgelistet und beschrieben. Es werden die Fehlercodes aus der "F_RELATED table (16#0170)" verwendet. Zusätzlich können die Fehlercodes der Funktionsbausteine MB_DiameterCalculatorType03 (siehe [Kap. 9.3.2 "MB_DiameterCalculatorType03" auf Seite 436](#)) und IL_PIDType01 aus der Bibliothek RIL_LoopControl (siehe Dokumentation "Rexroth IndraWorks 12VRS Basis-Bibliotheken") sowie MB_GetCyclicParameterHandle (siehe [Kap. 5.9.5 "MB_GetCyclicParameterHandle" auf Seite 133](#)) auftreten.

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#11B0	16#20	Fehler in der Statemachine
INPUT_RANGE_ERROR	16#11B1	16#01	Diameter.MechanicInertia < 0.0
INPUT_RANGE_ERROR	16#11B1	16#02	TensionCtrl.PreControlWeighting < 0.0
INPUT_RANGE_ERROR	16#11B1	16#03	MaxTension < 0.0
INPUT_RANGE_ERROR	16#11B1	16#04	MaxTension < MinTension
INPUT_RANGE_ERROR	16#11B1	16#05	Friction.StandstillTorque < 0.0
INPUT_RANGE_ERROR	16#11B1	16#06	Friction.MaxTorque < 0.0
INPUT_RANGE_ERROR	16#11B1	16#07	Friction.MinTorque < 0.0
INPUT_RANGE_ERROR	16#11B1	16#08	Friction.MinVelocity < 0.0
INPUT_RANGE_ERROR	16#11B1	16#09	WindowTension < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0A	MinTension < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0B	MaxTension < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0C	Wickelachse nicht in "BB"
INPUT_RANGE_ERROR	16#11B1	16#0D	Drehmomentwichtung falsch (A-0-0050) Wichtung für Motormoment muss motorseitig bezogen sein
INPUT_RANGE_ERROR	16#11B1	16#0E	Achsnummer der Slaveachse falsch
INPUT_RANGE_ERROR	16#11B1	16#0F	Achsnummer der Masterachse falsch
RESSOURCE_ERROR	16#0004	16#0000	falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Abb.9-58: Fehlercodes Funktionsbaustein MB_WinderTensionCtrlType01

Notwendige Hardware

- IndraMotion MLC Hardware CML65, CML45, CML25
- IndraDrive

Notwendige Firmware

- IndraMotion MLC Firmware FWA-CMLXX*-MLC-12VRS
XX - Gerätevariante, z.B. CML65
- IndraDrive Antriebsfirmware

RMB_TechWinder.library/MX_TechWinder.lib

Notwendige Software	IndraWorks 12VRS für IndraMotion MLC bzw. IndraMotion MLD
Besonderheiten für IndraMotion MLC	Für die Wickelachse muss Interpolation im Antrieb aktiviert sein. (reale Achsen → entspr. Achse → Eigenschaften) Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Im zyklischen SERCOS-Kanal des entsprechenden Antriebs müssen die Parameter P-0-0694 ("Feinabgleich Getriebeübersetzung, Prozessregler") und Parameter P-0-0690 ("Geschwindigkeits-Sollwert additiv, Prozessregler") und S-0-0092 "Drehmoment-/Kraft-Grenzwert-bipolar" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter S-0-0121 ("Lastgetriebe-Eingangsumdrehungen") und S-0-0122 ("Lastgetriebe-Ausgangsumdrehungen") eingetragen werden. Den Parameter A-0-0032 "Geschwindigkeitsgrenzwert positiv" bzw. A-0-0033 "Geschwindigkeitsgrenzwert negativ" entsprechend der maximalen Drehzahl des Wickelantriebes setzen (dieser Wert muss im Parameter S-0-0091 "Geschwindigkeitsgrenzwert bipolar" auftauchen). Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z.B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!
Besonderheiten für IndraMotion MLD	Bei Verwendung der Steuerung IndraMotion MLD-S mit übergeordneter Steuerung IndraMotion MLC muss für die Wickelachse Interpolation im Antrieb aktiviert sein (reale Achsen → entspr. Achse → Eigenschaften). Bei Verwendung der Steuerung IndraMotion MLD-M gibt es diese Einstellung nicht. Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Im zyklischen SERCOS-Kanal des entsprechenden Antriebs müssen die Parameter P-0-0694 ("Feinabgleich Getriebeübersetzung, Prozessregler") und Parameter P-0-0690 ("Geschwindigkeits-Sollwert additiv, Prozessregler") und S-0-0092 "Drehmoment-/Kraft-Grenzwert-bipolar" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus (der Dialog ist im Projektbaum unterhalb des MLD-Knotens → AxisData). Die "AxisData-Struktur" des Antriebes muss parametrierbar sein (P-0-1367 "SPS-Konfiguration" Bit 6 ist gesetzt, bzw. Dialog SPS-Konfiguration). Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter S-0-0121 ("Lastgetriebe-Eingangsumdrehungen") und S-0-0122 ("Lastgetriebe-Ausgangsumdrehungen") eingetragen werden. Den Parameter S-0-0091 "Geschwindigkeitsgrenzwert bipolar" entsprechend der maximalen Drehzahl des Wickelantriebes setzen. Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z.B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!

9.6.3 Inbetriebnahme des Wickelrechners mit Zugkraftsteuerung bzw. Zugkraftregelung

Antriebsregelkreis	Es wird vorausgesetzt dass der Drehzahlregelkreis des Wickelantriebes eingestellt wurde. D.h. die Parameter S-0-0100 "Geschwindigkeitsregler-propor-

tionalverstärkung" und S-0-0101 "Geschwindigkeitsregler-Nachstellzeit" müssen der Applikation entsprechend eingestellt sein.

Im Folgenden wird die Inbetriebnahme des Wickelrechners beschrieben. Die Inbetriebnahme erfolgt in sieben Schritten:

1. Festlegung des Wicklertyps
2. Festlegung der Drehrichtung der Wickelachse
3. Ermittlung des Reibmoments im Stillstand
4. Ermittlung des Reibmoments bei maximaler Drehzahl
5. Ermittlung des minimalen Reibmoments
6. abschließende Einstellungen und
7. erster Anlauf der Wickelachse

Kommandierung der Wicklerachse

Für die Wicklerachse muss die richtige Betriebsart des Antriebes, sowie die Reglerfreigabe des Antriebes gesetzt werden. Dies kann auf zwei unterschiedliche Arten erfolgen:

- Setzen der Betriebsart "ModeSyncVel" mit Hilfe des Achsinterfaces (dies wird empfohlen)
- Benutzung der PLC-Open Funktionsbausteine "MC_Power", "MC_GearIn", "MC_Stop" zur Kommandierung der Achse

Festlegung des Wicklertyps

Zunächst muss festgelegt werden, welcher Wicklertyp vorliegt bzw. eingesetzt werden soll. Dazu wird am Eingang "WindDirection" die Wicklerrichtung festgelegt (TRUE: Abwickeln, FALSE: Aufwickeln). Zudem wird der Wickelrechner mit dem Eingang "WindCtrlType" ausgewählt (TRUE: Wickelrechner mit Zugkraftregelung, FALSE: Wickelrechner mit Zugkraftsteuerung).

Festlegung der Drehrichtung der Wickelachse

Die Drehrichtung des Wicklers kann mit Hilfe des Antriebsparameters P-0-0108 "Polarität Leitantrieb" bzw. des Achsparameters A-0-2798 "Polarität Leitachsposition" eingestellt werden. Dreht sich die Wickelachse nicht in die gewünschte Richtung, ist der Parameter P-0-0108 bzw. der Achsparameter A-0-2798 entsprechend umzustellen. Die Polarität der Leitachsposition wird lediglich beim Einschalten des Funktionsbausteins überprüft ("Enable" = TRUE). Eine Änderung der genannten Parameter während des Betriebs des Funktionsbausteins zeigt somit keine Wirkung.

Alternativ kann die Drehrichtung über die Wichtungsart des Wickelantriebs geändert werden. Dies kann durch Drehen des Vorzeichens der Lage-Beschleunigungs- und der Drehmoment-Kraftdaten erfolgen (Parameter S-0-0055, A-0-0029, S-0-0043, S-0-0085).

Die folgende Abbildung zeigt die prinzipiellen Zusammenhänge bei Auf- bzw. Abwickeln.

RMB_TechWinder.library/MX_TechWinder.lib

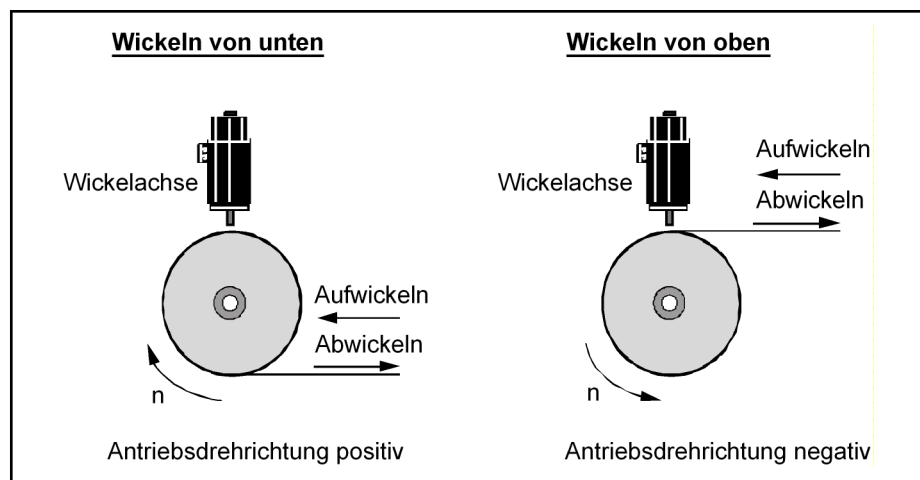


Abb.9-59: Prinzipien von Auf- und Abwickeln

Ermittlung des Reibmoments im Stillstand

1. Das "Reibmoment im Stillstand" (Variable "StandstillTorque" in der Datenstruktur MB_WIND_TENSION) wird bei stehender Leitachse und mit leerem Wickel (nur Hülse) ermittelt. Der Wickelbaustein ist ausgeschaltet. Für die Wickelachse (Folgeachse) ist die Betriebsart "Geschwindigkeitsregelung" zu wählen (im Achs-Interface der Bibliothek ML_TechInterface "MODE_VEL" anwählen), alternativ kann die Achse mit dem PLCopen-Funktionsbaustein MC_MoveVelocity kommandiert werden. Als Sollwert kann z.B. 10 U/min eingestellt werden.
2. Nun wird der Wert Parameter P-0-0109 "Spitzendrehmoment-/Kraft-Begrenzung" auf den Wert Null gesetzt. Die Achse muss nun stehen bleiben. Der Wert für diesen Parameter wird nun schrittweise erhöht, bis sich der Antrieb zu drehen beginnt. Der letzte Wert entspricht dem Reibmoment im Stillstand.
3. Die Achsfreigabe "MODE_VEL" wird wieder zurückgenommen (alternativ mit dem PLCopen-Funktionsbaustein MC_Stop).
Der Parameter P-0-0109 "Spitzendrehmoment-/Kraft-Begrenzung" wird wieder auf seinen alten Wert (z.B. 400%) gesetzt.
Die Einstellung des Reibmoments im Stillstand ist somit abgeschlossen.

Ermittlung des Reibmoments bei maximaler Drehzahl

1. Das "Reibmoment bei maximaler Drehzahl" (Variable "MaxTorque" in der Datenstruktur MB_WIND_TENSION) wird bei stehender Leitachse und mit leerem Wickel (nur Hülse) ermittelt. Der Funktionsbaustein ist ausgeschaltet.
2. Für die Wickelachse (Folgeachse) ist die Betriebsart "Geschwindigkeitsregelung" zu wählen (im Achs-Interface der Bibliothek ML_TechInterface "MODE_VEL" anwählen, alternativ PLCopen-FB).
Der Sollwert für die angewählte Betriebsart ist schrittweise bis zum Wert des Parameters S-0-0091 "Geschwindigkeits-Grenzwert bipolar" zu erhöhen. Der Wert für den Parameter S-0-0091 "Geschwindigkeits-Grenzwert bipolar" muss entsprechend Applikation und Motordaten eingetragen sein.
3. Im nächsten Schritt wird der Wert des Parameters P-0-0109 "Spitzendrehmoment-/Kraft-Begrenzung" schrittweise verringert (z.B. von 100% Richtung 0%), bis die Drehzahl des Antriebs ebenfalls zu sinken beginnt. Die Drehzahl des Antriebs kann mittels des Antriebsparameters

RMB_TechWinder.library/MX_TechWinder.lib

S-0-0040 "Geschwindigkeits-Istwert" bzw. des Achsparameters A-0-0102 "Geschwindigkeits-Istwert" verfolgt werden. Der letzte Wert von P-0-0109 "Spitzendrehmoment-/Kraft-Begrenzung" entspricht dem maximalen Reibmoment bei maximaler Drehzahl.

4. Die Achsfreigabe "MODE_VEL" wird wieder zurückgenommen (alternativ mit dem PLCopen-Funktionsbaustein MC_Stop).

Der Parameter P-0-0109 "Spitzendrehmoment-/Kraft-Begrenzung" wird wieder auf seinen alten Wert (z.B. 400%) gesetzt.

Die Einstellung des Reibmoments maximaler Drehzahl "MaxTorque" ist somit abgeschlossen.

Ermittlung des minimalen Reibmoments

Das minimale Reibmoment wird mit Hilfe eines Bremsversuchs mit leerem Wickel ermittelt. Der Wickelantrieb wird von einer konstanten Geschwindigkeit (z.B. 50 min⁻¹) bis zum Stillstand abgebremst. Gleichzeitig werden die Antriebsparameter S-0-0080 "Drehmoment-/Kraftsollwert" (Achsparameter A-0-0038 "Drehmoment/Kraft-Grenzwert bipolar") und S-0-0040 "Geschwindigkeits-Istwert" (Achsparameter A-0-0102 "Geschwindigkeits-Istwert") mit einem Oszilloskop bzw. mit der Tracefunktion in IndraWorks aufgezeichnet.

Aus dem Drehmomentverlauf wird der Minimalwert abgelesen und kann der Struktur MB_WIND_FRICTION als "MinTorque" übergeben werden. Die korrespondierende Drehzahl wird als "MinTorqueVelocity" in MB_WIND_FRICTION eingetragen.

Abschließende Einstellungen

Schließlich werden, wie in der folgenden Tabelle gezeigt, die verbleibenden Eingangsgrößen festgelegt.

Eingangsgröße	Bemerkung/Einstellung
SetpointOpMode	Einstellung des gewünschten Laufzugs
SetpointStandstill	Einstellung des Stillstandzugs
TensionMode	Auswahl der Spanndrehzahlcharakteristik: • TRUE: Spannen mit konstanter Bahngeschwindigkeit • FALSE: Spannen mit konstanter Bahngeschwindigkeit
Offset im Feinabgleich	In der Datenstruktur MB_AXIS_SYNCHRONISATION aus der Bibliothek ML_TechInterface kann eine Voreinstellung des Feinabgleichs durch die Variable "FineAdjust" mit einem Wert abhängig von der Wickelrichtung belegt. Diese Variable wird zyklisch verarbeitet, wenn "EnableCyclicScanning" = TRUE gesetzt ist. Die Wickelrichtung wird, wie zuvor beschrieben, mit "WindDirection" festgelegt. Bei einem Abwickler wird ein positiver Wert für "FineAdjust" und für einen Aufwickler ein negativer Wert eingesetzt. Diese Werte sollten ≤ 5 % betragen

RMB_TechWinder.library/MX_TechWinder.lib

Eingangsgröße	Bemerkung/Einstellung
ActTorqueFilter	Soll das Istdrehmoment (als Messwert) gefiltert werden, muss an diesem Eingang des Funktionsbaustein ein Wert ungleich Null als Zeitkonstante angelegt werden. Dieser Wert wirkt auf den Standardparameter S-0-0084 "Drehmoment/Kraft-Istwert"
AccelTorqueFilter	Soll das Beschleunigungsmoment gefiltert werden, muss an diesem Eingang des Funktionsbausteins ein Wert ungleich Null als Zeitkonstante angelegt werden

Abb.9-60: Abschließende Festlegung der verbleibenden Eingangsgrößen

Erster Anlauf der Wickelachse

Das Wickelgut wird ungespannt (leicht durchhängend) aufgelegt. Die Wickelachse muss entweder über das Achsinterface oder über die PLCopen Funktionsbausteine kommandiert werden. Danach wird der Wickelrechner mit Setzen des Eingangs "Enable" aktiviert. Jetzt sollte sich das Wickelgut mit der vorgegebenen Spanndrehzahl (Eingang "TensionSpeed") spannen. Der Ausgang "SpeedOffset" stellt die Spanndrehzahl zur Verfügung.

Das Prozessgrößenfenster "WindowTension" kann auf einen Wert von 10 - 20% eingestellt werden.

Die Leitachse wird mit einer vergleichsweise niedrigen Drehzahl (z.B. 10 min^{-1}) freigegeben. Nach einigen Umdrehungen muss der Ausgang "DiameterEngaged" gesetzt sein. Hiermit wird angezeigt, dass die Durchmesserberechnung aktiv ist ("eingerastet"). Diese ist abhängig von der Bahnspannung bzw. dem Istmoment an der Wickelachse und der Drehmomentgrenze (Antriebsparameter S-0-0092 "Drehmoment/Kraft-Grenzwert bipolar" bzw. Achsparameter A-0-0038 "Drehmoment/Kraft-Grenzwert bipolar"). Die Drehmoment/Kraft-Grenze wird am Ausgang "TorqueLimitBipolar" ausgegeben.

Bei eingerastetem Wickelrechner können Stillstands- und Laufzug überprüft werden. Dazu wird der Wert "CurrentTension" aus der Datenstruktur ML_WIND_TENS_DIAG beobachtet. In dieser Datenstruktur können zudem der aktuelle Drehmomentgrenzwert "TorqueLimit", das aktuelle Beschleunigungsmoment "AccelerationTorque" und das aktuelle angenommene Reibmoment "FrictionTorque" betrachtet werden.

Schließlich stellt der Funktionsbaustein an seinen Ausgängen verschiedene Prozessgrößen zur Verfügung. Der berechnete aktuelle Durchmesser des Wickels (Ausgang "CurrentDiameter") sollte mit geringfügiger, vernachlässigbarer Abweichung dem tatsächlichen Durchmesser des Wickels entsprechen. Ferner werden der Getriebefeinabgleich zur Drehzahlachsführung der Antriebsachse des Wickels (Ausgang "GearRatioFineAdjust"), das aktuelle Trägheitsmoment des Wickels (Ausgang "CurrentInertia") und die Spanndrehzahl (Ausgang "SpeedOffset") ausgegeben.

Momentengrenzwertüberwachungen

Beim ersten Hochlaufen der Steuerung kann es zu einer Warnung des Antriebes "E2056 Momenten-Grenzwert = 0" kommen, wenn der Wicklerbaustein noch nicht eingeschaltet ist.

Um dies zu verhindern, kann in der Anlaufphase des SPS-Programmes (z.B. Warmstart) der entsprechende zyklische Datencontainer, indem sich der Wert für den Parameter S-0-0092 "Drehmoment/Kraftwert bipolar" befindet, mit einem Wert ungleich Null beschrieben werden, z.B. 100%.

Kommt es bei eingeschaltetem Baustein zur Warnung "E8260 Moment-/Kraft-Sollwertbegrenzung aktiv", kann diese Warnung durch Setzen des Bit 4 des Parameters P-0-0556 "Achsregler-Konfiguration" beseitigt werden.

In seltenen Fällen (z.B. mehrfach F8078 "Fehler im Drehzahlregelkreis") kann es notwendig sein, die Geschwindigkeitsreglerüberwachung zu deaktivieren (Setzen des Bit 1 des Parameters P-0-0556 "Achsregler-Konfiguration").

9.7 Funktionsbaustein zum Wickelrechner mit Zugkraftregelung und Drehzahlkorrektur

9.7.1 Einführung und Übersicht

Der Wickelrechner mit Zugkraftregelung und Drehzahlkorrektur wird an Zentrumswicklern eingesetzt. Der Antrieb der Wickelachse arbeitet generell NICHT an seiner Drehmomentengrenze. Aus Sicherheitsgründen wird eine Drehmomentenbegrenzung berechnet die zusätzlich noch einstellbar ist (diese sollte oberhalb des benötigten Motormomentes liegen und den Antrieb im normalen Betrieb nicht begrenzen).

Dieser Wickler wird hauptsächlich für elastisches, dehnbares Material eingesetzt.

Der Zugspannungswert wird direkt mittels einer Kraftmessdose gemessen und über einen Analogkanal oder über einen Feldbus dem Zugspannungsregler als Istwert zugeführt. Bei gespannter Bahn kann der Durchmesserrechner den aktuellen Wickeldurchmesser berechnen.

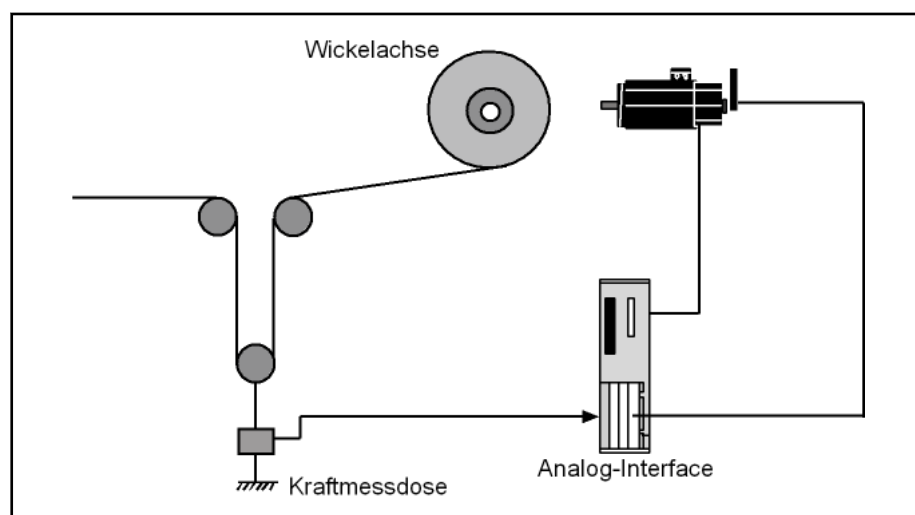


Abb.9-61: Wickelrechner mit Zugkraftregelung

9.7.2 MB_WinderTensionCtrlSpeedType01

Kurzbeschreibung

Der Funktionsbaustein berechnet zyklisch den aktuellen Wickeldurchmesser, das aktuelle Trägheitsmoment des Wickels, die entsprechende Geschwindigkeitsnachführung über den Getriebefeinabgleich, einen additiven Drehzahloffset zum Spannen der Bahn und eine parametrierbare Drehmomentenbegrenzung.

Zielgrößen des Funktionsbausteines sind die Parameter P-0-0694 "Feinabgleich Getriebeübersetzung, Prozessregler" P-0-0690 "Geschwindigkeitssollwert additiv, Prozessregler", S-0-0081 "Drehmoment-/Kraftsollwert additiv" und S-0-0092 "Drehmoment/Kraft-Grenzwert bipolar".

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechWinder.compiled-library

Abb.9-62: Referenztable MB_WinderTensionCtrlSpeedType01

RMB_TechWinder.library/MX_TechWinder.lib

Schnittstellenbeschreibung

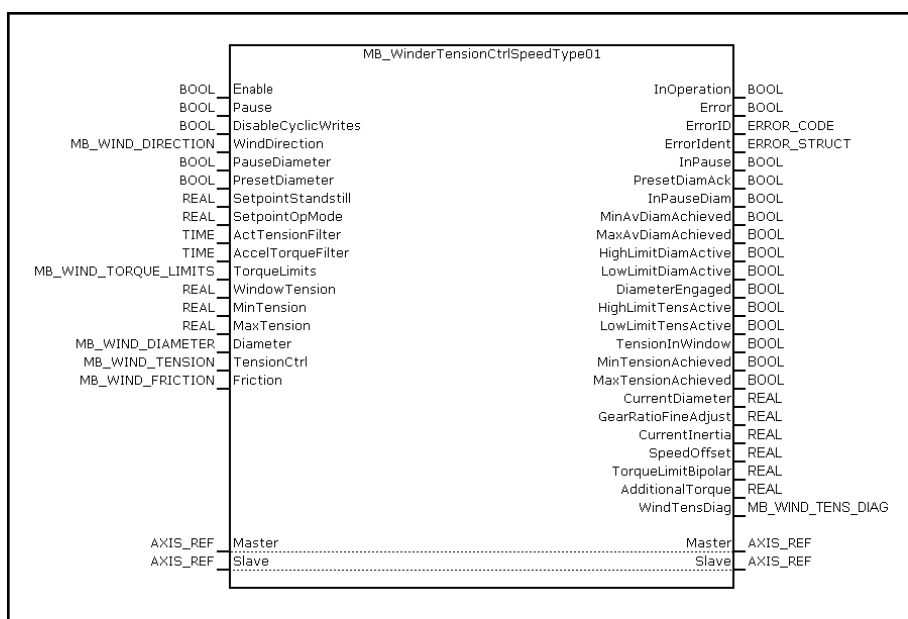


Abb.9-63: Funktionsbaustein MB_WinderTensionCtrlSpeedType01

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (Wickelachse)
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	Pause	BOOL	Pausieren des Zugspannungsreglers und des Wickelrechners (Werte werden eingefroren)
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	WindDirection	MB_WIND_DIRECTION	Wickelrichtung Aufwickler REWIND (16#0000) Abwickler UNWIND (16#0001)
	PauseDiameter	BOOL	Anhalten der Durchmesserberechnung.
	PresetDiameter	BOOL	CurrentDiameter wird auf PresetVal gesetzt, GearRatioFineAdjust entsprechend angepasst (flankengesteuert)
	SetpointStandstill	REAL	Vorgabe des Stillstandszuges [N] (Umschaltung zwischen Stillstands- u. Laufzug anhand der Stillstandsmeldung des Antriebes)
	SetpointOpMode	REAL	Vorgabe des Laufzuges [N] (Umschaltung zwischen Stillstands- und Laufzug anhand der Stillstandsmeldung des Antriebes)
	ActTensionFilter	TIME	Filterzeitkonstante für die Istzugspannung
	AccelTorqueFilter	TIME	Filterzeitkonstante für das Beschleunigungsmoment

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Typ	Kommentar
	TorqueLimits	MB_WIND_TORQUE_LIMITS	Struktur für die bipolare Drehmomentbegrenzung (S-0-0092)
	WindowTension	REAL	Fenster für den Zugspannungswert [%] bezogen auf den Zugspannungssollwert("TensionInWindow")
	MinTension	REAL	Untere Schwelle für den Zugspannungswert (nur für die Anzeige mit binärem Ausgang "MinTensionAck")
	MaxTension	REAL	Obere Schwelle für den Zugspannungswert (nur für die Anzeige mit binärem Ausgang „MaxTensionAck“)
	Diameter	MB_WIND_DIAMETER	Struktur für Durchmesserrechner
	TensionCtrl	MB_WIND_TENSION	Struktur für Zugspannungsregler
	Friction	MB_WIND_FRICTION	Struktur für Reibmoment
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	InPause	BOOL	Zugspannungsregler und Wickelrechner pausieren
	PresetDiamAck	BOOL	Presetdurchmesser wurde übernommen (ist gesetzt, solange Preset ansteht)
	InPauseDiam	BOOL	Pause Durchmesserrechner
	MinAvDiamAck	BOOL	Minimaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	MaxAvDiamAck	BOOL	Maximaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	HighLimitDiamAck	BOOL	Aktuell berechneter Durchmesser hat obere Grenze erreicht (Durchmesserwert wird verworfen)
	LowLimitDiamAck	BOOL	Aktuell berechneter Durchmesser hat untere Grenze erreicht (Durchmesserwert wird verworfen)
	DiameterEngaged	BOOL	Durchmesser "eingerastet" TRUE: eingerastet FALSE: ausgerastet
	HighLimitTensActive	BOOL	Ausgang des Zugspannungsreglers hat obere Grenze erreicht
	LowLimitTensActive	BOOL	Ausgang des Zugspannungsreglers hat untere Grenze erreicht
	TensionInWindow	BOOL	Zugspannungswert ist innerhalb des Zugspannungsfensters
	MinTensionAchieved	BOOL	Zugspannung unterhalb des Einganges "MinTension"
	MaxTensionAchieved	BOOL	Zugspannung oberhalb des Einganges "MaxTension"
	CurrentDiameter	REAL	Aktuell berechneter Durchmesser [mm]
	GearRatioFineAdjust	REAL	Resultierender Getriebefineabgleich zur Drehzahl-nachführung des Antriebes [%]

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Typ	Kommentar
	CurrentInertia	REAL	Aktuelles gesamtes Trägheitsmoment (bezogen auf die Motorseite) [kg/m²]
	SpeedOffset	REAL	Spanndrehzahl [U/min]
	TorqueLimitBipolar	REAL	Bipolare Drehmomentengrenze [%]
	AdditionalTorque	REAL	Additives Drehmoment [%]
	WindTensDiag	MB_WIND_TENS_DIAG	Diagnosestruktur

Abb.9-64: Schnittstellenvariablen Funktionsbaustein MB_WinderTensionCtrl-SpeedType01

Min-/Max und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an Enable
WindDirection	MB_WIND_DIRECTION			FALSE	Steigende Flanke an Enable
PauseDiameter	BOOL			FALSE	Kontinuierlich
PresetDiameter	BOOL			FALSE	Kontinuierlich
SetpointStandstill	REAL	0.0	n.def.	0.0	Kontinuierlich
SetpointOpMode	REAL	0.0	n.def.	0.0	Kontinuierlich
ActTensionFilter	TIME	0s	n.def.	0s	Kontinuierlich
AccelTorqueFilter	TIME	0s	n.def.	0s	Kontinuierlich
TorqueLimits	MB_WIND_TORQUE_LIMITS				Kontinuierlich
WindowTension	REAL	0.0	n.def.	0.0	Kontinuierlich
MinTension	REAL	0.0	n.def.	0.0	Kontinuierlich
MaxTension	REAL	0.0	n. def.	0.0	Kontinuierlich
Diameter	MB_WIND_DIAMETER				Je nach Strukturelement
TensionCtrl	MB_WIND_TENSION				kontinuierlich
Friction	MB_WIND_FRICTION				kontinuierlich

Abb.9-65: Min-/Max- und Default-Werte der Eingänge MB_WinderTensionCtrl-SpeedType01

MB_WIND_DIRECTION

RMB_TechWinder.library/MX_TechWinder.lib

Element	Wert	Beschreibung
REWIND	0	Aufwickler
UNWIND	1	Abwickler

Abb.9-66: Elemente des Aufzählungstyps MB_WIND_DIRECTION

MB_WIND_TORQUE_LIMITS

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
DefaultTorqueLimit	REAL	0.0	400.0	100.0	kontinuierlich	Wert für die bipolare Drehmomentengrenze nach Wegnahme von "Enable" [%]
TorqueLimitFactor	REAL	0.0	400.0	50.0	kontinuierlich	Faktor um den das berechnete TourqueLimit erhöht wird [%]
MinTorqueLimit	REAL	0.0	400.0	10.0	kontinuierlich	Minimales TourqueLimit um den das berechnete TourqueLimit erhöht wird [%]

Abb.9-67: Schnittstellenbeschreibung Datenstruktur MB_WIND_TORQUE_LIMITS

MB_WIND_DIAMETER

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
PresetVal	REAL	> 0.0	n. def.	100.0	kontinuierlich	Vorgabewert für den Presetdurchmesser [mm]
CoreVal	REAL	> 0.0	n. def.	100.0	steigende Flanke an Enable	Durchmesser Hülse [mm]
CalcRange	REAL	0.0	n. def.	360.0	kontinuierlich	Bereich für Durchmesserberechnung [Grad]
RepeatLength	REAL	> 0.0	n. def.	100.0	steigende Flanke an Enable	Formatlänge [mm] (1 Umdrehung der Referenzachse abgewickelt)
Webwidth	REAL	0.0	n. def.	0.0	steigende Flanke an Enable	Bahnbreite [mm] wird für das Trägheitsmoment benötigt
Density	REAL	0.0	n. def.	1.0	steigende Flanke an Enable	Materialdichte [kg/dm³] wird für das Trägheitsmoment benötigt
AverageValue	UINT	0	20	1	kontinuierlich	Anzahl der Durchmesserwerte die gemittelt werden [-]
HighLimit	REAL	LowLimit	n. def.	2000.0	kontinuierlich	obere Grenze für den Durchmesser [mm]
LowLimit	REAL	> 0.0	HighLimit	100.0	kontinuierlich	untere Grenze für den Durchmesser [mm]

RMB_TechWinder.library/MX_TechWinder.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
MinDiameter	REAL	> 0.0	n. def.	100.0	kontinuierlich	minimaler Durchmesser [mm]
MaxDiameter	REAL	> 0.0	n. def.	2000.0	kontinuierlich	maximaler Durchmesser [mm]
MechanicInertia	REAL	0.0	n. def.	0.0	kontinuierlich	Trägheitsmoment Mechanik [kg/m ²] bezogen auf die Motorseite Wird für das resultierende Trägheitsmoment benötigt
DancerConstant	REAL	0.0	n. def.	0.0	kontinuierlich	Verstellung des Tänzers bezogen auf die Einheit der Tänzerverstellung [mm/-] Wird nur bei einem Wickler mit Tänzer benötigt
MeasuredVal	REAL	0.0	n. def.	0.0	kontinuierlich	gemessener Durchmesser [mm]
PivotVelocity	REAL	0.0	n. def.	0.0	kontinuierlich	Schwenkgeschwindigkeit (für Relativbewegung der Wickelachse) [mm/s]
DiamCalcMode	MB_WIND_DIAM_CALC_MODE				steigende Flanke an Enable	CALC_DIAM_MANUAL Durchmesserberechnung abhängig von CalcRange FIRST_REV_AUTO_CALC Durchmesserberechnung erfolgt für 1. Umdrehung alle 5 Grad, danach abhängig vom CalcRange
DiamSource	BOOL			False	steigende Flanke an Enable	Durchmesserermittlung: FALSE: Durchmesser wird berechnet TRUE: Durchmesser wird gemessen (MeasuredVal)

Abb.9-68: Schnittstellenbeschreibung Datenstruktur MB_WIND_DIAMETER

MB_WIND_DIAM_CALC_MODE

Element	Wert	Beschreibung
CALC_DIAM_MANUAL	0	Durchmesserberechnung abhängig von CalcRange
FIRST_REV_AUTO_CALC	1	Durchmesserberechnung erfolgt für 1. Umdrehung alle 3 Grad, danach abhängig vom CalcRange

Abb.9-69: Elemente des Auszählungstyps MB_WIND_DIAM_CALC_MODE

MB_WIND_TENSION

RMB_TechWinder.library/MX_TechWinder.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
PControl	REAL	0.0	n. def.	0.0	kontinuierlich	Kp-Verstärkung des Zugspannungsreglers
IControl	REAL	0.0	n. def.	0.0	kontinuierlich	Tn Nachstellzeit des Zugspannungsreglers
FeedbackVal	REAL	0.0	n. def.	0.0	kontinuierlich	Istwert der Zugspannung
PreControlWeigh- ting	REAL	0.0	n. def.	1.0	kontinuierlich	Wichtung der Sollwertauf- schaltung
HighLimit	REAL	0.0	n. def.	0.0	kontinuierlich	obere Grenze des Zugspan- nungsreglers
LowLimit	REAL	0.0	n. def.	0.0	kontinuierlich	untere Grenze des Zug- spannungsreglers

Abb.9-70: Schnittstellenbeschreibung Datenstruktur MB_WIND_TENSION

MB_WIND_FRICTION

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
StandstillTorque	REAL	0.0	n. def.	0.0	kontinuierlich	Reibmoment im Stillstand
MaxTorque	REAL	0.0	n. def.	0.0	kontinuierlich	Reibmoment bei S-0-0091, Geschwindigkeitsgrenzwert bipolar
MinTorque	REAL	0.0	n. def.	0.0	kontinuierlich	minimales Reibmoment
MinTorqueVeloci- ty	REAL	0.0	n. def.	0.0	kontinuierlich	Geschwindigkeit beim mini- malen Reibmoment

Abb.9-71: Schnittstellenbeschreibung Datenstruktur MB_WIND_FRICTION

MB_WIND_TENS_DIAG

Strukturelement	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
TensionCtrlOutput	REAL	n. def.	n. def.	0.0	kontinuierlich	Ausgang des Zugspan- nungsreglers [N]
TorqueLimit	REAL	n.def.	n. def.	0.0	kontinuierlich	Vorzeichenbehaftetes Tor- quelimit [%]
AccelerationTor- que	REAL	n. def.	n. def.	0.0	kontinuierlich	Aktuelles Beschleunigungs- moment [%]
FrictionTorque	REAL	n. def	n. def	0.0	kontinuierlich	Aktuelles Reibmoment [%]
CurrentTension	REAL	0.0	n. def.	0.0	kontinuierlich	Aktueller Zugspannungsist- wert [N]

Abb.9-72: Schnittstellenbeschreibung Datenstruktur MB_WIND_TENS_DIAG

RMB_TechWinder.library/MX_TechWinder.lib

Signal-Zeitdiagramm

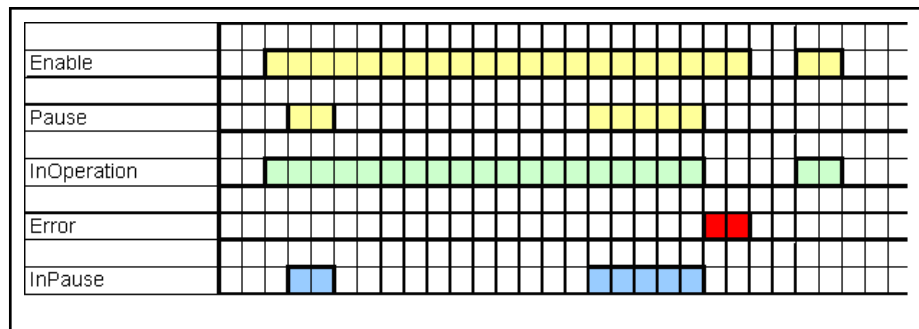


Abb.9-73: Signal-Zeitdiagramm für MB_WinderTensionCtrlSpeedType01

Funktionsbeschreibung

Der Funktionsbaustein stellt einen Wickelrechner mit Zugkraftregelung dar, der normalerweise nicht an der Drehmomentengrenze arbeitet.

Der Funktionsbaustein berechnet zyklisch den aktuellen Wickeldurchmesser, das aktuelle Trägheitsmoment des Wickels, die entsprechende Geschwindigkeitsnachführung über den Getriebefeinabgleich und einen additiven Drehzahloffset zum Spannen der Stoffbahn, sowie den benötigten Drehmomentensollwert zur Vorsteuerung.

Intern werden die Bausteine MB_DiameterCalculatorType03 und MB_DiameterMeasurementType01 verwendet.

Zielgrößen des Funktionsbausteins sind die Parameter P-0-0694 "Feinabgleich Getriebeübersetzung, Prozessregler", P-0-0690 "Geschwindigkeits-Sollwert additiv, Prozessregler", S-0-0081 "Drehmoment-/Kraft-Sollwert additiv" und S-0-0092 "Drehmoment-/Kraft-Grenzwert bipolar".

Die Geschwindigkeitsanpassung aufgrund des sich ändernden Wickeldurchmessers erfolgt über den Getriebefeinabgleich, der Zugspannungsregler wirkt auf den Geschwindigkeitssollwert additiv, das berechnete Drehmoment (Moment für die Zugspannung, Reibmoment, Beschleunigungsmoment) wird über den additiven Drehmomentensollwert vorgesteuert.

Aus Sicherheitsgründen wird der Drehmomentengrenzwert auf den aktuell berechneten Drehmomentensollwert multipliziert mit einem Wichtungsfaktor, sowie einem zusätzlichen additiven Anteil gesetzt. Im normalen Wickelbetrieb soll die Drehmomentenbegrenzung nicht wirken und der Wickelantrieb drehzahl geregelt betrieben werden.

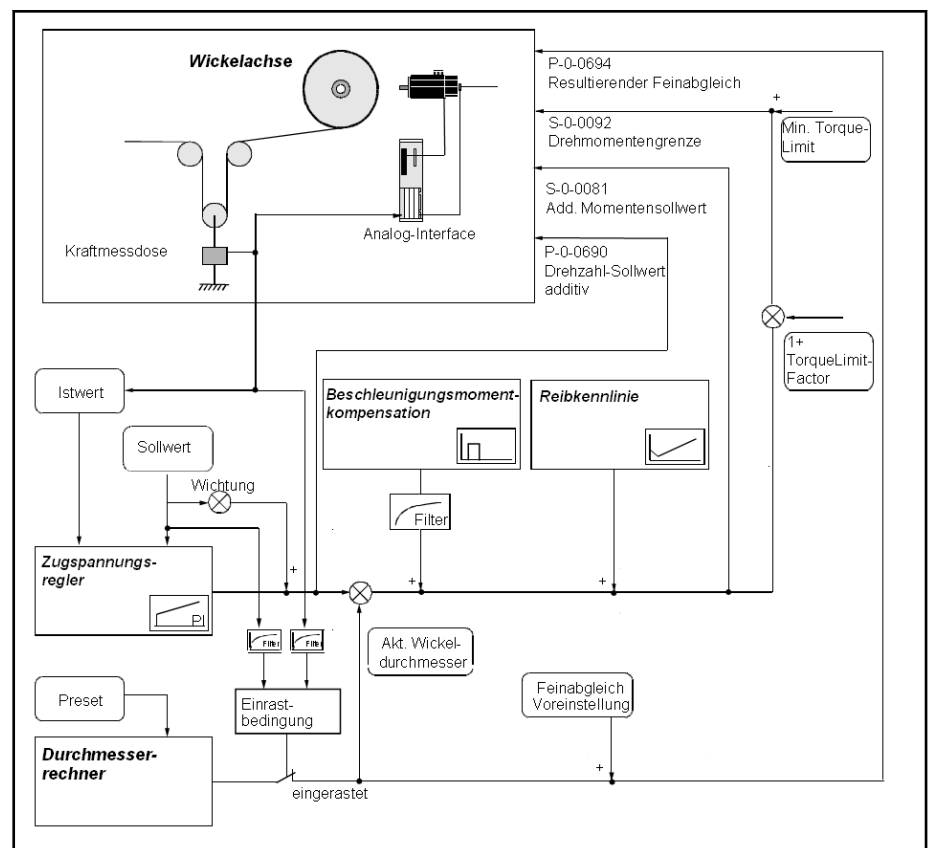


Abb.9-74: Schematische Darstellung der Regelung bei einer Wickelachse

Drehrichtung des Wickelantriebes, Auf-/Abwickler

Mit Hilfe der Enumeration "MB_WindDirection" des Funktionsbausteins wird der Wicklertyp Auf- bzw. Abwickler eingestellt. Die Drehrichtung des Antriebs ist entweder über die Wichtungsparameter oder über die Polarität vom Leit-antrieb einzustellen.



Die Drehrichtung der Masterachse des Wickelantriebes (d.h. die Masterachse des Bausteins MC_GearIn bzw. Masterachse des AxisInterface) muss positiv sein, d.h. positiver Drehzahlsollwert. Eine Invertierung des Wickelantriebes muss über die Polaritätenparameter erfolgen.

Durchmesserberechnung

Das Drehmoment für den Wickelantrieb ergibt sich aus dem Produkt von Zugkraft und Wickelradius. Daher muss der Radius bzw. der Durchmesser des Wickels laufend bestimmt werden. Im Funktionsbaustein ist daher ein interner Durchmesserrechner (Funktionsbaustein MB_DiameterCalculator-Type03, siehe [Kap. 9.3.2 "MB_DiameterCalculatorType03" auf Seite 436](#)) eingebunden.

Die Durchmesserberechnung wird erst bei gespanntem Wickelgut und keiner Stillstandsmeldung der Referenzachse aktiv und wird zyklisch in Abhängigkeit von der Wickelachsdrehzahl bzw. -lage durchgeführt. Zu Diagnosezwecken kann der aktuelle Durchmesser am Funktionsbaustein ausgelesen werden.

Der berechnete Wickeldurchmesser wirkt direkt auf den resultierenden Feinabgleich des Getriebes. Wird die Bahnspannung zu gering oder negativ

Kompensation des Beschleunigungsmoments

Für die Beschleunigung der Wickelachse ist ein zusätzliches Beschleunigungsmoment erforderlich. Dieses ist abhängig vom Wickeldurchmesser. Mit der Angabe von Material- und Stoffbahndaten wird eine durchmesserabhängige Trägheit des Wickels berechnet. Das berechnete Beschleunigungsmoment kann zudem gefiltert werden.

Drehmomentenbegrenzung und Vorsteuerung

Der Antrieb der Wickelachse arbeitet generell NICHT an seiner Drehmomentengrenze. Aus Sicherheitsgründen wird ein berechnetes Drehmoment zusätzlich eines einstellbaren additiven Anteils als Wert für die Drehmomentengrenze eingestellt. (dieser Wert sollte überhalb des benötigten Motormomentes liegen und den Antrieb im normalen Betrieb nicht begrenzen).

Eine Drehmomentengrenze, die proportional der Zugspannung und dem jeweiligen Wickeldurchmesser ist wird zyklisch berechnet.

$$M_{Grenz} \sim F_{Zug} * r_{Wickel} + M_{Beschl.} + M_{Reib}$$

Abb.9-76: Drehmomentengrenze

Dieser zyklische Wert für die Drehmomentengrenze wird als additiver Drehmomentensollwert vorgesteuert.

Desweiteren kann dieser Wert um einen proportionalen Anteil (TorqueLimitFactor) und einen additiven Anteil (MinTorqueLimit) erhöht werden, damit die Drehmomentengrenze im normalen Wickelbetrieb nicht wirkt.

$$M_{ResGrenz} \sim M_{Grenz} * (1 + TorqueLimitFactor) + MinTorqueLimit$$

Abb.9-77: resultierende Drehmomentengrenze

Diese resultierende Drehmomentenbegrenzung entspricht dem Ausgang TorqueLimitbipolar. Standardmäßig wird die berechnete Drehmomentengrenze um den TorqueLimitFactor von 50% und einem minimalen TorqueLimit von 10% erhöht.

Während des Spannvorgangs ist es evtl. sinnvoll die Drehmomentenbegrenzung niedriger zu setzen (z.B. TorqueLimitFactor 10%) , wenn der Spannvorgang mit zuviel Drehmoment erfolgt. Nach erfolgtem Spannvorgang sollte der TorqueLimitFactor wieder erhöht werden.

Rollenwechsel

Nach einem Rollenwechsel ändert sich in der Regel der Wickeldurchmesser schlagartig. Der zuletzt berechnete interne Wert für den Wickeldurchmesser entspricht dann nicht mehr dem aktuellen Durchmesser. Damit die Durchmesserberechnung korrekt aufsetzen kann, ist ein Preset des Wickeldurchmessers erforderlich. Der Eingangswert für den Preset sollte annähernd dem tatsächlichen Wert des Wickeldurchmessers entsprechen.



Wird der Preset nicht durchgeführt, kann es zu Fehlfunktionen kommen. Daher sollte der Preset abgeschlossen sein, bevor der Durchmesserrechner wieder aktiv wird.

Der Wert für den Preset sollte wie folgt gewählt werden:

- beim Aufwickler etwas kleiner bzw. gleich dem tatsächlichen Wickeldurchmesser und
- beim Abwickler etwas größer bzw. gleich dem tatsächlichen Wickeldurchmesser

RMB_TechWinder.library/MX_TechWinder.lib

Bei einem fliegenden Rollenwechsel muss sichergestellt werden, dass die Umfangsgeschwindigkeit der neuen Rolle, der aktuellen Rolle entspricht. Dies sollte entsprechend im SPS-Projekt in IndraLogic umgesetzt werden.

Filterung des Zugspannungswerts

Bei niedrigen Zugspannungen kann ein ungefilterter Zugspannungswert zu einem Ausrasten des Wickelrechners führen. Daher sollte eine Filterung vorgenommen und der entsprechende Eingang beschaltet werden (z.B. 20ms).

Zusätzliche Vor- bzw. Nacheilung der Wickelachse

Eine zusätzliche Vor- bzw. Nacheilung über den Getriebefeinabgleich (Voreinstellung Feinabgleich "FineAdjust") kann über die Strukturvariable "qrAxisCtrl[AxisNummer].Admin.SyncMode.FineAdjust" der Datenstruktur MB_AXIS_SYNCHRONISATION der Bibliothek ML_TechInterface eingestellt werden. Bei einem Aufwickler sollte ein kleiner positiver Wert (z.B. 5%), bei einem Abwickler ein kleiner negativer Wert eingestellt werden. Dies führt dann bei der jeweiligen Wickelrichtung zu einem besseren Anlaufverhalten.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins aufgelistet und beschrieben. Es werden die Fehlercodes aus der "F_RELATED table (16#0170)" verwendet. Zusätzlich können die Fehlercodes der Bausteine MB_DancerControlType03 (siehe [Kap. 9.2.2 "MB_DancerControlType03" auf Seite 428](#)), MB_DiameterCalculatorType03 (siehe [Kap. 9.3.2 "MB_DiameterCalculatorType03" auf Seite 436](#)), MB_DiameterMeasurementType01 (siehe [Kap. 9.4.2 "MB_DiameterMeasurementType01" auf Seite 445](#)), IL_PIDType01 aus der Bibliothek RIL_LoopControl (siehe Dokumentation "Rexroth IndraWorks 12VRS Basis-Bibliotheken") sowie MB_GetCyclicParameterHandle (siehe [Kap. 5.9.5 "MB_GetCyclicParameterHandle" auf Seite 133](#)) auftreten.

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#1120	16#0020	Fehler in der State machine
INPUT_RANGE_ERROR	16#1121	16#0001	Diameter.MechanicInertia < 0.0
INPUT_RANGE_ERROR	16#1121	16#0002	TensionCtrl.PreControlWeighting < 0.0
INPUT_RANGE_ERROR	16#1121	16#0003	MaxTension < 0.0
INPUT_RANGE_ERROR	16#1121	16#0004	MaxTension < MinTension
INPUT_RANGE_ERROR	16#1121	16#0005	Friction.StandstillTorque < 0.0
INPUT_RANGE_ERROR	16#1121	16#0006	Friction.MaxTorque < 0.0
INPUT_RANGE_ERROR	16#1121	16#0007	Friction.MinTorque < 0.0
INPUT_RANGE_ERROR	16#1121	16#0008	Friction.MinVelocity < 0.0
INPUT_RANGE_ERROR	16#1121	16#0009	WinderTension < 0.0
INPUT_RANGE_ERROR	16#1121	16#000A	MinTension < 0.0
INPUT_RANGE_ERROR	16#1121	16#000B	MaxTension < 0.0
INPUT_RANGE_ERROR	16#1121	16#000C	Drehmomentwichtung falsch (A-0-0050 bzw. S-0-0086) Wichtung für Motormoment muss motorseitig sein
INPUT_RANGE_ERROR	16#1121	16#000E	Achsnummer der Slaveachse falsch
INPUT_RANGE_ERROR	16#1121	16#000F	Achsnummer der Masterachse falsch
INPUT_RANGE_ERROR	16#1121	16#0010	SetpointStandstill < 0.0

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR	16#1121	16#0011	SetpointOpMode < 0.0
INPUT_RANGE_ERROR	16#1121	16#0012	DefaultTorqueLimit < 0.0
INPUT_RANGE_ERROR	16#1121	16#0013	MinTorqueLimit < 0.0
INPUT_RANGE_ERROR	16#1121	16#0014	TorqueLimitFactor < 0.0
INPUT_RANGE_ERROR	16#1122	16#0001	Wickelachse nicht in „BB“
RESSOURCE_ERROR	16#0004	16#0000	Falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert
RESSOURCE_ERROR	16#0012	16#00xx	AxisRef.CntrlNo falsch

Abb.9-78: Fehlercodes Funktionsbaustein MB_WinderTensionCtrlSpeedType01

Besonderheiten für IndraMotion MLC

Für die Wickelachse muss Interpolation im Antrieb aktiviert sein. (reale Achsen → entspr. Achse → Eigenschaften)

Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Im zyklischen SERCOS-Kanal des entsprechenden Antriebs müssen die Parameter P-0-0694 "Feinabgleich Getriebeübersetzung, Prozessregler", Parameter P-0-0690 "Geschwindigkeits-Sollwert additiv, Prozessregler", S-0-0092 "Drehmoment-/Kraft-Grenzwert-bipolar" und S-0-0081 "Drehmoment-/Kraft-sollwert additiv" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.

Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter S-0-0121 "Lastgetriebe-Eingangsumdrehungen" und S-0-0122 "Lastgetriebe-Ausgangsumdrehungen" eingetragen werden. Den Parameter A-0-0032 "Geschwindigkeitsgrenzwert positiv" bzw. A-0-0033 "Geschwindigkeitsgrenzwert negativ" entsprechend der maximalen Drehzahl des Wickelantriebes setzen (dieser Wert kann im Parameter S-0-0091 "Geschwindigkeitsgrenzwert bipolar" ausgelesen werden).

Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z.B. 0.1 U/min), da bei Stillstandsmeldung der Referenzachse der Durchmesserrechner angehalten wird, d.h. nicht rechnet.

9.7.3 Inbetriebnahme des Wicklers mit Zugregelung und Drehzahlkorrektur

Antriebsregelkreis

Es wird vorausgesetzt dass der Drehzahlregelkreis des Wickelantriebes eingestellt wurde. D.h. die Parameter S-0-0100 "Geschwindigkeitsregler-proportionalverstärkung" und S-0-0101 "Geschwindigkeitsregler-Nachstellzeit" müssen der Applikation entsprechend eingestellt sein.

Im Folgenden wird die Inbetriebnahme des Wickelrechners beschrieben. Die Inbetriebnahme erfolgt in sieben Schritten:

1. Festlegung des Wicklertyps
2. Festlegung der Drehrichtung der Wickelachse
3. Ermittlung des Reibmoments im Stillstand
4. Ermittlung des Reibmoments bei maximaler Drehzahl
5. Ermittlung des minimalen Reibmoments
6. abschließende Einstellungen und

RMB_TechWinder.library/MX_TechWinder.lib

7. erster Anlauf der Wickelachse

Kommandierung der Wicklerachse

Für die Wicklerachse muss die richtige Betriebsart des Antriebes, sowie die Reglerfreigabe des Antriebes gesetzt werden. Dies kann auf zwei unterschiedliche Arten erfolgen:

- Setzen der Betriebsart "ModeSyncVel" mit Hilfe des Achsinterfaces (dies wird empfohlen)
- Benutzung der PLCopen Funktionsbausteine "MC_Power", "MC_GearIn", "MC_Stop" zur Kommandierung der Achse

Festlegung des Wicklertyps

Zunächst muss festgelegt werden, welcher Wicklertyp vorliegt bzw. eingesetzt werden soll. Dazu wird am Eingang "WindDirection" die Wicklerrichtung festgelegt (TRUE: Abwickeln, FALSE: Aufwickeln).

Festlegung der Drehrichtung der Wickelachse

Die Drehrichtung des Wicklers kann mit Hilfe des Antriebsparameters P-0-0108 "Polarität Leittrieb" bzw. des Achsparameters A-0-2798 "Polarität Leitachseposition" eingestellt werden. Dreht sich die Wickelachse nicht in die gewünschte Richtung, ist der Parameter P-0-0108 bzw. der Achsparameter A-0-2798 entsprechend umzustellen. Die Polarität der Leitachseposition wird lediglich beim Einschalten des Funktionsbausteins überprüft ("Enable" = TRUE). Eine Änderung der genannten Parameter während des Betriebs des Funktionsbausteins zeigt somit keine Wirkung.

Alternativ kann die Drehrichtung über die Wichtungsart des Wickelantriebs geändert werden. Dies kann durch Drehen des Vorzeichens der Lage-Beschleunigungs- und der Drehmoment-Kraftdaten erfolgen (Parameter S-0-0055, A-0-0029, S-0-0043, S-0-0085).

Die folgende Abbildung zeigt die prinzipiellen Zusammenhänge bei Auf- bzw. Abwickeln.

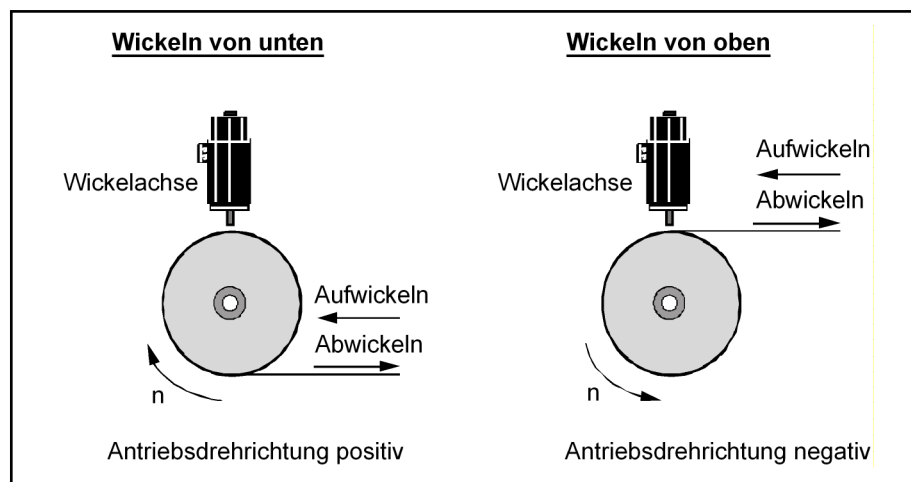


Abb.9-79: Prinzipien von Auf- und Abwickeln

Ermittlung des Reibmoments im Stillstand

1. Das "Reibmoment im Stillstand" (Variable "StandstillTorque" in der Datenstruktur MB_WIND_TENSION) wird bei stehender Leitachse und mit leerem Wickel (nur Hülse) ermittelt. Der Wickelbaustein ist ausgeschaltet. Für die Wickelachse (Folgeachse) ist die Betriebsart "Geschwindigkeitsregelung" zu wählen (im Achs-Interface der Bibliothek ML_TechInterface "MODE_VEL" anwählen), alternativ kann die Achse mit dem PLCopen-Funktionsbaustein MC_MoveVelocity kommandiert werden. Als Sollwert kann z.B. 10 U/min eingestellt werden.

RMB_TechWinder.library/MX_TechWinder.lib

2. Nun wird der Wert Parameter P-0-0109 "Spitzendrehmoment-/Kraft-Begrenzung" auf den Wert Null gesetzt. Die Achse muss nun stehen bleiben. Der Wert für diesen Parameter wird nun schrittweise erhöht, bis sich der Antrieb zu drehen beginnt. Der letzte Wert entspricht dem Reibmoment im Stillstand.
3. Die Achsfreigabe "MODE_VEL" wird wieder zurückgenommen (alternativ mit dem PLCopen-Funktionsbaustein MC_Stop).
Der Parameter P-0-0109 "Spitzendrehmoment-/Kraft-Begrenzung" wird wieder auf seinen alten Wert (z.B. 400%) gesetzt.
Die Einstellung des Reibmoments im Stillstand ist somit abgeschlossen.

Ermittlung des Reibmoments bei maximaler Drehzahl

1. Das "Reibmoment bei maximaler Drehzahl" (Variable "MaxTorque" in der Datenstruktur MB_WIND_TENSION) wird bei stehender Leitachse und mit leerem Wickel (nur Hülse) ermittelt. Der Funktionsbausteins ist ausgeschaltet.
2. Für die Wickelachse (Folgeachse) ist die Betriebsart "Geschwindigkeitsregelung" zu wählen (im Achs-Interface der Bibliothek ML_TechInterface "MODE_VEL" anwählen, alternativ PLCopen-FB).
Der Sollwert für die angewählte Betriebsart ist schrittweise bis zum Wert des Parameters S-0-0091 "Geschwindigkeits-Grenzwert bipolar" zu erhöhen. Der Wert für den Parameter S-0-0091 "Spitzendrehmoment-/Kraft-Begrenzung" muss entsprechend Applikation und Motordaten eingetragen sein.
3. Im nächsten Schritt wird der Wert des Parameters P-0-0109 "Spitzendrehmoment-/Kraft-Begrenzung" schrittweise verringert (z.B. von 100% Richtung 0%), bis die Drehzahl des Antriebs ebenfalls zu sinken beginnt. Die Drehzahl des Antriebs kann mittels des Antriebsparameters S-0-0040 "Geschwindigkeits-Istwert" bzw. des Achsparameters A-0-0102 "Geschwindigkeits-Istwert" verfolgt werden. Der letzte Wert von P-0-0109 "Spitzendrehmoment-/Kraft-Begrenzung" entspricht dem maximalen Reibmoment bei maximaler Drehzahl.
4. Die Achsfreigabe "MODE_VEL" wird wieder zurückgenommen (alternativ mit dem PLCopen-Funktionsbaustein MC_Stop).
Der Parameter P-0-0109 "Spitzendrehmoment-/Kraft-Begrenzung" wird wieder auf seinen alten Wert (z.B. 400%) gesetzt.
Die Einstellung des Reibmoments maximaler Drehzahl "MaxTorque" ist somit abgeschlossen.

Ermittlung des minimalen Reibmoments

Das minimale Reibmoment wird mit Hilfe eines Bremsversuchs mit leerem Wickel ermittelt. Der Wickelantrieb wird von einer konstanten Geschwindigkeit (z.B. 50 min⁻¹) bis zum Stillstand abgebremst. Gleichzeitig werden die Antriebsparameter S-0-0080 "Drehmoment-/Kraftsollwert" (Achsparemeter A-0-0038 "Drehmoment/Kraft-Grenzwert bipolar") und S-0-0040 "Geschwindigkeits-Istwert" (Achsparemeter A-0-0102 "Geschwindigkeits-Istwert") mit einem Oszilloskop bzw. mit der Tracefunktion in IndraWorks aufgezeichnet.

Aus dem Drehmomentverlauf wird der Minimalwert abgelesen und kann der Struktur MB_WIND_FRICTION als "MinTorque" übergeben werden. Die korrespondierende Drehzahl wird als "MinTourqueVelocity" in MB_WIND_FRICTION eingetragen.

Abschließende Einstellungen

RMB_TechWinder.library/MX_TechWinder.lib

Schließlich werden die verbleibenden Eingangsgrößen festgelegt.

Die Tabelle führt diese auf.

Eingangsgröße	Bemerkung/Einstellung
SetpointOpMode	Einstellung des gewünschten Laufzugs
SetpointStandstill	Einstellung des Stillstandzugs
Offset im Feinabgleich	In der Datenstruktur MB_AXIS_SYNCHRONISATION aus der Bibliothek ML_TechInterface kann eine Voreinstellung des Feinabgleichs durch die Variable "FineAdjust" mit einem Wert abhängig von der Wickelrichtung belegt. Diese Variable wird zyklisch verarbeitet, wenn "EnableCyclicScanning" = TRUE gesetzt ist. Die Wickelrichtung wird, wie zuvor beschrieben, mit "WindDirection" festgelegt. Bei einem Abwickler wird ein positiver Wert für "FineAdjust" und für einen Aufwickler ein negativer Wert eingesetzt. Diese Werte sollten $\leq 5\%$ betragen
ActTensionFilter	Soll die Ist-Spannung (als Messwert) gefiltert werden, muss an diesem Eingang des Funktionsbausteins ein Wert ungleich Null als Zeitkonstante angelegt werden
AccelTorqueFilter	Soll das Beschleunigungsmoment gefiltert werden, muss an diesem Eingang des Funktionsbausteins ein Wert ungleich Null als Zeitkonstante angelegt werden

Abb.9-80: Abschließende Festlegung der verbleibenden Eingangsgrößen

Erster Anlauf der Wickelachse

Das Wickelgut wird ungespannt (leicht durchhängend) aufgelegt. Die Wickelachse muss entweder über das Achsinterface oder über die PLCopen Funktionsbausteine kommandiert werden. Danach wird der Wickelrechner mit Setzen des Eingangs "Enable" aktiviert. Jetzt sollte das Wickelgut über die Bahnspannungsregelung gespannt werden. Der Ausgang "SpeedOffset" stellt die Geschwindigkeit dar, mit der gespannt wird.

Das Prozessgrößenfenster "WindowTension" kann z.B. auf einen Wert von 10 - 20% eingestellt werden.

Die Leitachse wird mit einer vergleichsweise niedrigen Drehzahl (z.B. 10 min^{-1}) freigegeben. Nach einigen Umdrehungen muss der Ausgang "DiameterEngaged" gesetzt sein. Hiermit wird angezeigt, dass die Durchmesserberechnung aktiv ist ("engerastet"). Diese ist abhängig von der Bahnspannung bzw. dem Istmoment an der Wickelachse.

Bei eingerastetem Wickelrechner können Stillstands- und Laufzug überprüft werden. Dazu wird der Wert "CurrentTension" aus der Datenstruktur MB_WIND_TENS_DIAG beobachtet. In dieser Datenstruktur können zudem der aktuelle Drehmomentgrenzwert "TorqueLimit", das aktuelle Beschleunigungsmoment "AccelerationTorque" und das aktuelle angenommene Reibmoment "FrictionTorque" betrachtet werden.

Schließlich stellt der Funktionsbaustein an seinen Ausgängen verschiedene Prozessgrößen zur Verfügung. Der berechnete aktuelle Durchmesser des Wickels (Ausgang "CurrentDiameter") sollte mit geringfügiger, vernachlässigbarer Abweichung dem tatsächlichen Durchmesser des Wickels entsprechen. Ferner werden der Getriebefeinabgleich zur Drehzahlachsführung der Antriebsachse des Wickels (Ausgang "GearRatioFineAdjust"), das aktuelle Trägheitsmoment des Wickels (Ausgang "CurrentInertia") und der Drehzahloffset (Ausgang "SpeedOffset"), das additive Drehmoment "AddTorque" und der Drehmomentengrenzwert (Ausgang "TorqueLimitBipolar") ausgegeben.

Momentengrenzwertüberwachungen

Beim ersten Hochlaufen der Steuerung kann es zu einer Warnung des Antriebes "E2056 Momenten-Grenzwert = 0" kommen, wenn der Wicklerbaustein noch nicht eingeschaltet ist.

Um dies zu verhindern, kann in der Anlaufphase des SPS-Programmes (z.B. Warmstart) der entsprechende zyklische Datencontainer, indem sich der Wert für den Parameter S-0-0092 "Drehmoment/Kraftwert bipolar" befindet, mit einem Wert ungleich Null beschrieben werden, z.B. 100%.

Kommt es bei eingeschaltetem Baustein zur Warnung "E8260 Moment-/Kraft-Sollwertbegrenzung aktiv", kann diese Warnung durch Setzen des Bit 4 des Parameters P-0-0556 "Achsregler-Konfiguration" beseitigt werden.

In seltenen Fällen (z.B. mehrfach F8078 "Fehler im Drehzahlregelkreis") kann es notwendig sein, die Geschwindigkeitsreglerüberwachung zu deaktivieren (Setzen des Bit 1 des Parameters P-0-0556 "Achsregler-Konfiguration").

9.8 Ergänzende Funktionsbausteine

9.8.1 Einführung und Übersicht

In diesem Abschnitt sind Funktionsbausteine zusammengefasst, die Funktionalitäten der verschiedenen Wicklerrechner und des Tänzerlageregler ergänzen können. Sie können zusätzlich in das SPS-Projekt in IndraLogic eingebunden werden.

Für zusätzliche Funktionalitäten (z.B. Restlaufzeit für einen Rollenwechsel) stehen weitere folgende Bausteine zur Verfügung.

Bezeichnung	Typ	Beschreibung
MB_CalcInertiaLimitType02	FB	Berechnet das maximale Trägheitsmoment des Wickelgutes auf dem Wickel
MB_UnwindMaterialType02	FB	Berechnet für einen Abwickler Prozessgrößen, wie die abgewickelte - und die restliche Länge der Stoffbahn, Materialstärke und Restlaufzeit für einen Rollenwechsel.
MB_WindTaperProfileType01	FB	Berechnet eine Wickelhärtekennlinie für einen Aufwickler
MB_WindSpeedControlAdaption-Type01	FB	Berechnet für einen Wickler eine Kp-Adaption für den Drehzahlregler

Abb.9-81: Kurzbeschreibung zusätzlicher Funktionsbausteine für verschiedene Wicklertypen aus der Bibliothek RMB_TechWinder

9.8.2 MB_CalcInertiaLimitType02

Kurzbeschreibung

Der Funktionsbaustein stellt, im Gegensatz zu den anderen in dieser Dokumentation beschriebenen Funktionsbausteinen, einen nicht zyklischen Baustein dar. Er wird gezielt, z.B. zur Initialisierung eines anderen Funktionsbausteins aufgerufen.

In diesem Funktionsbaustein werden die Materialeigenschaften und Abmessungen der Stoffbahn ausgewertet und das maximale Trägheitsmoment beim maximalen Durchmesser "MaxDiameter" für das Wickelgut auf dem Wickel berechnet. Das berechnete maximale Trägheitsmoment "MaxInertia" kann z.B. für eine Drehzahlregler-Adaption (Adaption des Parameters S-0-0100 "Geschwindigkeitsregler-Proportionalverstärkung") verwendet werden. Hier-

RMB_TechWinder.library/MX_TechWinder.lib

bei kann die Proportionalverstärkung zwischen einem Minimal- und Maximalwert angepasst werden. Es kann beispielsweise eine lineare Zunahme des Faktors der Proportionalverstärkung eingestellt werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechWinder.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_TechWinder.lib

Abb.9-82: Referenztabelle MB_CalcInertiaLimitType02

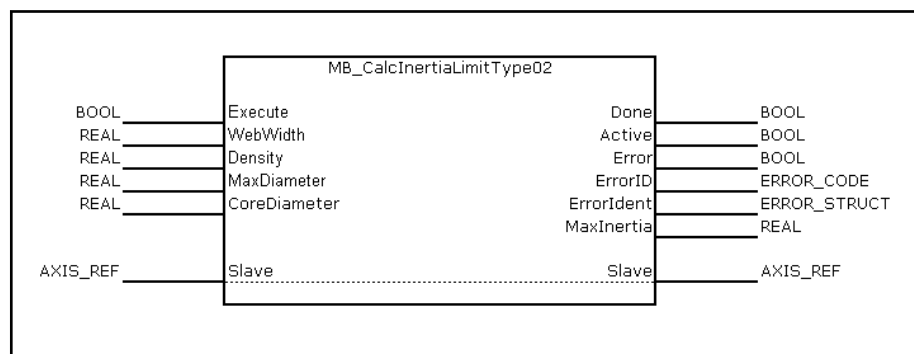
Schnittstellenbeschreibung

Abb.9-83: Funktionsbaustein MB_CalcInertiaLimitType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Einschalten des Bausteins (flankengesteuert)
	WebWidth	REAL	Breite der Stoffbahn [mm]
	Density	REAL	Materialdichte der Stoffbahn [kg/dm³]
	MaxDiameter	REAL	maximaler Durchmesser des Wickels [mm]
	CoreDiameter	REAL	Hülsendurchmesser des Wickels [mm]
VAR_OUTPUT	Done	BOOL	maximales Trägheitsmoment "MaxInertia" berechnet
	Active	BOOL	Verarbeitung, Baustein aktiv
	Error	BOOL	Fehler aufgetreten
	ErrorID	ERROR_CODE	Fehlercode
	ErrorIdent	ERROR_STRUCT	erweiterte Fehleranzeige
	MaxInertia	REAL	maximales Trägheitsmoment [kg/dm³]
VAR_IN_OUT	Slave	AXIS_REF	Achsstruktur der Wickelachse

Abb.9-84: Schnittstellenvariablen Funktionsbaustein MB_CalcInertiaLimitType02

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
Webwidth	REAL	0.0	n. def.	0.0	steigende Flanke an Execute
Density	REAL	0.0	n. def.	1.0	steigende Flanke an Execute

RMB_TechWinder.library/MX_TechWinder.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
MaxDiameter	REAL	> 0.0	n. def	2000.0	steigende Flanke an Execute
CoreDiameter	REAL	> 0.0	n. def.	100.0	steigende Flanke an Execute

Abb.9-85: Min-/Max- und Default-Werte MB_CalcInertiaLimitType02

Signal-Zeit-Diagramm

Das nachfolgend dargestellte Signal-Zeit-Diagramm für die Betriebsein- und -ausgänge zeigt das flankengesteuerte Verhalten. Erfolgt am Eingang "Execute" eine positive Flanke, wird der Baustein aktiv. Die Verarbeitung, die hier beispielhaft mit einer Dauer von zwei Zyklen (zwei Kästchen in der Darstellung) angesetzt wird, wird durch den Ausgang "Active" angezeigt. Nach der Verarbeitung wird der Ausgang "Done" gesetzt. Dieser Zustand bleibt erhalten, wenn "Execute" gesetzt bleibt und kein Fehler ("Error") auftritt. Ist "Execute" nicht gesetzt, wird "Done" nach einem Zyklus gelöscht. Der Ausgang "Error" wird im Falle eines Fehlers gesetzt. "Error" bleibt gesetzt, bis "Execute" zurückgesetzt wird. Ist "Execute" nicht gesetzt, wird die Fehlermeldung am Ausgang "Error" nach einem Zyklus gelöscht.

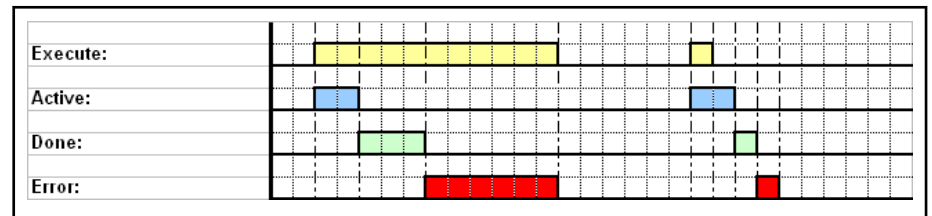


Abb.9-86: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_CalcInertiaLimitType02

Funktionsbeschreibung

Mit Hilfe des Hülsendurchmessers des Wickels "CoreDiameter", der Bahnweite "WebWidth" und der Dichte der Stoffbahn "Density" wird für den angegebenen maximalen Durchmesser "MaxDiameter" das Trägheitsmoment "MaxInertia" berechnet.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins aufgelistet und beschrieben. Es werden die Fehlercodes aus der "F_RELATED table (16#0170)" verwendet.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1140	16#0020	Fehler in der Statemachine
INPUT_RANGE_ERROR	16#1141	16#0001	Webwidth < 0.0
INPUT_RANGE_ERROR	16#1141	16#0002	Density < 0.0
INPUT_RANGE_ERROR	16#1141	16#0003	MaxDiameter < 0.0
INPUT_RANGE_ERROR	16#1141	16#0004	CoreDiameter < 0.0
INPUT_RANGE_ERROR	16#1141	16#0005	MaxDiameter < CoreDiameter
INPUT_RANGE_ERROR	16#1141	16#0006	Nummer der Slaveachse falsch
RESSOURCE_ERROR	16#0004	16#0000	Falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Abb.9-87: Fehlercodes Funktionsbaustein MB_CalcInertiaLimitType02

Notwendige Hardware

- IndraMotion MLC Hardware CML65, CML45, CML25
- IndraDrive

RMB_TechWinder.library/MX_TechWinder.lib

Notwendige Firmware

- IndraMotion MLC Firmware FWA-CMLXX*-MLC-12VRS
XX - Gerätevariante, z.B. CML65

Notwendige Software

- IndraDrive Antriebsfirmware

Notwendige Parametrierung Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter S-0-0121 ("Lastgetriebe-Eingangsumdrehungen") und S-0-0122 ("Lastgetriebe-Ausgangsumdrehungen") eingetragen sein.

9.8.3 MB_UnwindMaterialType02

Kurzbeschreibung Der Funktionsbaustein stellt einen Materialzähler dar. Er berechnet für einen Abwickler die zyklischen Prozessgrößen abgewickelte Länge der Stoffbahn "UnrolledLength", restliche Länge der Stoffbahn "RestLength", die Materialstärke "MaterialThickness" sowie die Restlaufzeit "RemainRunTime".

Mit Hilfe dieser Daten kann ein Rollenwechsel vorbereitet werden.

Als Eingangsgrößen dienen der aktuelle Wickeldurchmesser "CurrentDiameter" und der Restrollendurchmesser "RestRollDiameter".

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechWinder.compiled-library

Abb.9-88: Referenztabelle MB_UnwindMaterialType02

Schnittstellenbeschreibung

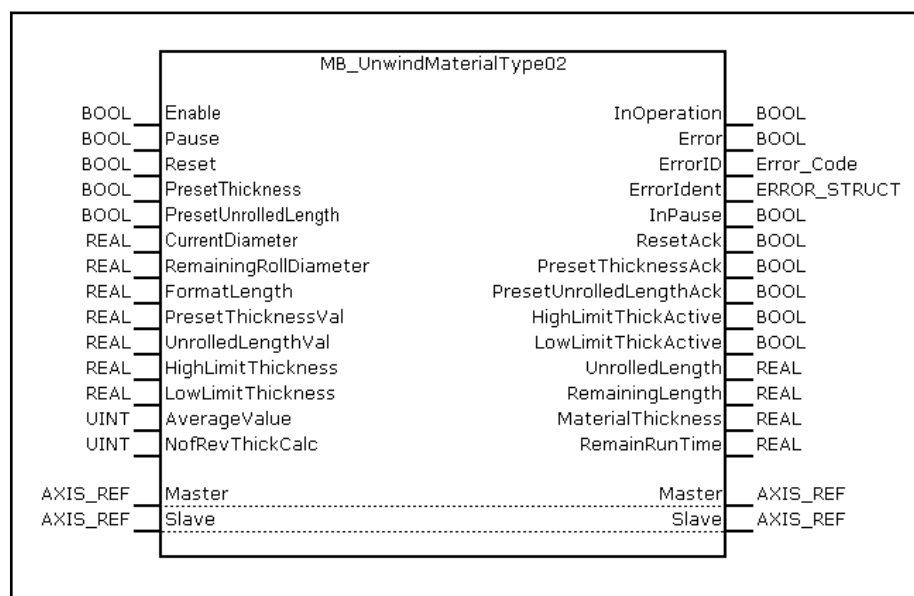


Abb.9-89: Funktionsbaustein MB_UnwindMaterialType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Einschalten des Bausteins (pegelgesteuert)
	Pause	BOOL	Materialzählung angehalten
	Reset	BOOL	Zurücksetzen der Werte für abgewickelte Länge der Stoffbahn, restliche Länge der Stoffbahn
	PresetThickness	BOOL	Preset für die Materialstärke flankengesteuert
	PresetUnrolledLength	BOOL	Preset für die abgewickelte Länge der Stoffbahn flankengesteuert

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Datentyp	Beschreibung
	CurrentDiameter	REAL	Aktueller Wickeldurchmesser [mm] vom Durchmesser-rechner-Baustein
	RemainingRollDiameter	REAL	Restrollendurchmesser [mm]
	FormatLength	REAL	Formatlänge [mm] (entspricht einer abgewickelten Umdrehung der Referenzachse)
	PresetThicknessVal	REAL	Preset-Wert für die Materialstärke [µm]
	UnrolledLengthVal	REAL	Preset-Wert für die abgewickelte Länge der Stoffbahn [mm]
	HighLimitThickness	REAL	obere Grenze für die Materialstärke [µm]
	LowLimitThickness	REAL	untere Grenze für die Materialstärke [µm]
	AverageValue	UINT	Anzahl der Materialstärkewerte, die gemittelt werden. Eingang wird auf maximal 30 begrenzt
	NofRevThickCalc	UINT	Anzahl der Umdrehungen, nach der eine Materialstärkeberechnung erfolgt
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Fehler aufgetreten
	ErrorID	ERROR_CODE	Fehlercode
	ErrorIdent	ERROR_STRUCT	Erweiterte Fehleranzeige
	InPause	BOOL	Materialberechnung angehalten
	ResetAck	BOOL	Reset ausgeführt
	PresetThicknessAck	BOOL	Materialstärke vorbesetzt
	PresetUnrolledLengthAck	BOOL	abgewickelte Länge vorbesetzt
	HighLimitThickAchieved	BOOL	Aktuell berechnete Materialstärke hat obere Grenze erreicht (Wert wird verworfen)
	LowLimitThickAchieved	BOOL	Aktuell berechnete Materialstärke hat untere Grenze erreicht (Wert wird verworfen)
	RemainingLength	REAL	Restlänge der Stoffbahn auf dem Wickel [mm]
	MaterialThickness	REAL	Materialstärke der Stoffbahn [µm]
	UnrolledLength	REAL	abgewickelte Länge der Stoffbahn [mm]
	RemainRunTime	REAL	Restlaufzeit bis zum Rollenwechsel (bezogen auf den Restrollendurchmesser) [s]
VAR_IN_OUT	Master	AXIS_REF	Achsstruktur der Referenzachse
	Slave	AXIS_REF	Achsstruktur der Wickelachse

Abb.9-90: Schnittstellenvariablen Funktionsbaustein MB_UnwindMaterialType02

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

RMB_TechWinder.library/MX_TechWinder.lib

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Pause	BOOL			FALSE	kontinuierlich
Reset	BOOL			FALSE	kontinuierlich
PresetThickness	BOOL			FALSE	kontinuierlich
PresetUnrolled- Length	BOOL			FALSE	kontinuierlich
CurrentDiameter	REAL	0.0	n. def.	0.0	kontinuierlich
RestRollDiameter	REAL	0.0	n. def.	0.0	steigende Flanke an Enable
FormatLength	REAL	0.0	n. def.	0.0	steigende Flanke an Enable
PresetThickness- Val	REAL	> 0.0	n. def.	0.0	kontinuierlich
UnrolledLength- Val	REAL	0.0	n. def.	0.0	kontinuierlich
HighLimitThick- ness	REAL	0.0	n. def.	500.0	kontinuierlich
LowLimitThick- ness	REAL	0.0	n. def.	10.0	kontinuierlich
AverageValue	UINT	0	30	1.0	kontinuierlich
NofRevThickCalc	UINT	0	n. def.	50	kontinuierlich

Abb.9-91: Min-/Max- und Default-Werte des Funktionsbausteins MB_Unwind-MatrialType02

Signal-Zeit-Diagramm

Im Folgenden wird das Signal-Zeit-Diagramm erläutert.

Mit dem Eingang "Enable" wird der Funktionsbaustein in Betrieb genommen. Dies wird am Ausgang "InOperation" angezeigt. Tritt ein Fehler auf, wird der Ausgang "Error" gesetzt und die Betriebsbereitschaft unterbrochen. Die Ausführung des Funktionsbausteins kann durch den Eingang "Pause" unterbrochen werden. Dies wird am Ausgang "InPause" signalisiert. Das Zurücksetzen des Bausteins (Eingang "Reset") oder ein Setzen von Materialparametern (Eingänge "PresetThickness" und "PresetUnrolledLength") erfolgt unabhängig vom Eingang "Pause". Jedoch ist es empfehlenswert, den Funktionsbaustein in "Pause" zu schalten. Dadurch lassen sich Preset und Reset kontrollierter und übersichtlicher durchführen. Das Setzen und Zurücksetzen wird an den Ausgängen "ResetAck", "PresetThicknessAck" und "PresetUnrolledLengthAck" angezeigt.

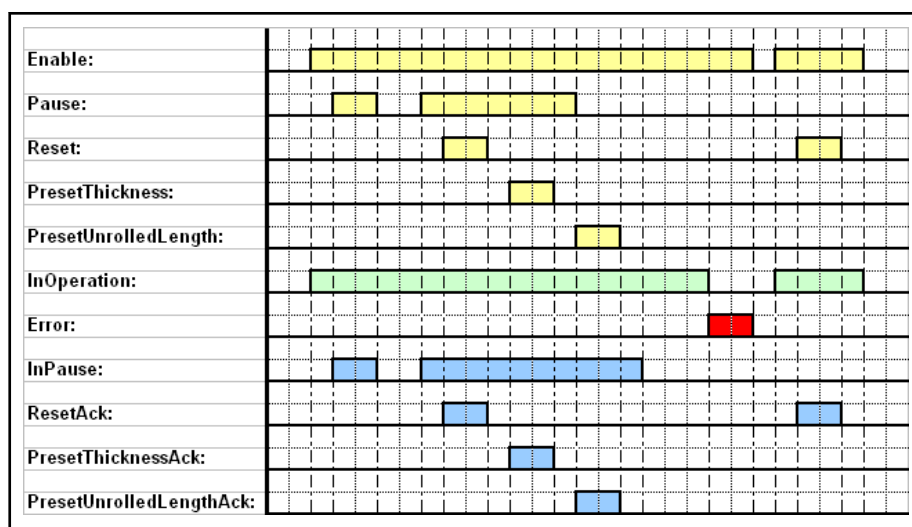


Abb.9-92: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_UnwindMaterialType02

Funktionsbeschreibung

Berechnung der Materialgrößen

Die restliche Länge der Stoffbahn "RestLength" wird mit Hilfe des aktuellen Durchmessers berechnet. D.h., Schwankungen im aktuellen Durchmesser sind auch in der Restlängenberechnung ("RestLength") zu verfolgen. Gleiches gilt für die Materialstärkenberechnung ("MaterialThickness"), die wiederum mit dem Wert der Restlänge arbeitet, sowie für die Restlaufzeitberechnung ("RemainRunTime"), die Materialstärke und Geschwindigkeit der Referenzachse verwendet. Je genauer der aktuelle Durchmesser ermittelt wird, desto genauer sind die zu berechnenden Ausgänge des Bausteines "UnrolledLength", "RestLength", "MaterialThickness" und "RemainRunTime".

Weitere Eingangsgrößen werden verwendet. Um physikalisch sinnvolle Werte für die Materialstärke zu erhalten, wird diese mit "HighLimitThickness" und "LowLimitThickness" begrenzt. Diese Begrenzung wirkt bei der Mittelung der Materialstärke. Eine feste Anzahl von Werten "AverageValue" wird gemittelt. Eine Materialstärkeberechnung wird alle "NofRevThickCalc" Umdrehungen der Wickelachse berechnen.

Preset der Materialgrößen

Zu Beginn des Materialzählers werden die Ausgänge initialisiert. Dies erfolgt mit den Werten "PresetThicknessVal" und "UnrolledLengthVal". Diese Werte können zusätzlich während des Betriebs als Preset-Werte eingesetzt werden. Durch die Eingänge "PresetThickness" und "PresetUnrolledLength" werden diese gesetzt.

Reset des Materialrechners

Ferner kann der Materialzähler durch den Eingang "Reset" zurückgesetzt werden. Das Zurücksetzen ("Reset") als auch das Vorbesetzen der Ausgänge (Preset) kann jederzeit erfolgen, wenn der Funktionsbaustein aktiv ist.

Mittelwertbildung der Materialstärke

Die berechnete Materialstärke kann mit Setzen des Eingangs "AverageValue" gemittelt werden. An "AverageValue" wird die Anzahl der Materialstärkewerte eingetragen, über die gemittelt werden soll.

Zur Mittelwertbildung tragen nur die intern berechneten aktuellen Durchmesser bei, die innerhalb der Grenzen von "HighLimitThickness" und "LowLimitThickness" liegen. Außerhalb liegende Werte werden verworfen. Für "AverageValue" sind Werte zwischen 1 und 10 zweckmäßig. Aus diesem Grund wird

RMB_TechWinder.library/MX_TechWinder.lib

"AverageValue" auf maximal 30 begrenzt. Wird ein größerer Wert eingegeben, wird "AverageValue" auf 30 gesetzt. Es wird allerdings keine Fehlermeldung ausgegeben, da es sich nicht um einen kritischen Vorgang handelt. Je höher die Zahl gewählt wird, desto glatter wird der berechnete Wert der Materialstärke. Allerdings nimmt mit zunehmenden Werten für "AverageValue" die Genauigkeit ab, da der berechnete Wert aufgrund der "Filterung" hinterherläuft. Daher sollte der Wert für "AverageValue" geeignet gewählt werden.



Nach dem Ausschalten der Steuerung werden die Speicher gelöscht. Somit sind die Werte für "UnrolledLength", "RestLength", "Materialthickness" und "RemainRunTime" beim Wiedereinschalten der Steuerung gleich Null.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins aufgelistet und beschrieben. Es werden die Fehlercodes aus der "F_RELATED table (16#0170)" verwendet.

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#11A0	16#20	Fehler in der Statemachine
INPUT_RANGE_ERROR	16#11A1	16#01	CurrentDiameter < 0.0
INPUT_RANGE_ERROR	16#11A1	16#02	RestRollDiameter < 0.0
INPUT_RANGE_ERROR	16#11A1	16#03	FormatLength <= 0.0
INPUT_RANGE_ERROR	16#11A1	16#04	PresetThicknessVal < 0.0
INPUT_RANGE_ERROR	16#11A1	16#05	UnrolledLengthVal < 0.0
INPUT_RANGE_ERROR	16#11A1	16#06	HighLimitThickness < 0.0
INPUT_RANGE_ERROR	16#11A1	16#07	LowLimitThickness < 0.0
INPUT_RANGE_ERROR	16#11A1	16#08	HighLimitThickness < LowLimitThickness
INPUT_RANGE_ERROR	16#11A1	16#09	AverageValue < 0
INPUT_RANGE_ERROR	16#11A1	16#0A	NofRevThickCalc < 0
INPUT_RANGE_ERROR	16#11A1	16#0B	Nummer der Slaveachse falsch
INPUT_RANGE_ERROR	16#11A1	16#0C	Nummer der Masterachse falsch
RESSOURCE_ERROR	16#0004	16#0000	falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Abb.9-93: Fehlercodes Funktionsbaustein MB_UnwindMaterialType02

Notwendige Hardware

- IndraMotion MLC Hardware CML65, CML45, CML25
- IndraDrive

Notwendige Firmware

- IndraMotion MLC Firmware FWA-CMLXX*-MLC-12VRS
XX - Gerätevariante, z.B. CML65

Notwendige Software

- IndraWorks 12VRS für IndraMotion MLC bzw. IndraMotion MLD

Notwendige Parametrierung

Für die Wicklerachse muss Interpolation im Antrieb aktiviert sein.

(reale Achsen → entspr. Achse → Eigenschaften)

Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

9.8.4 MB_WindTaperProfileType01

Kurzbeschreibung Dieser Funktionsbaustein realisiert eine durchmesserabhängige Anpassung der Wickelhärte für einen Aufwickler.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechWinder.compiled-library

Abb.9-94: Referenztable MB_WindTaperProfileType01

Schnittstellenbeschreibung

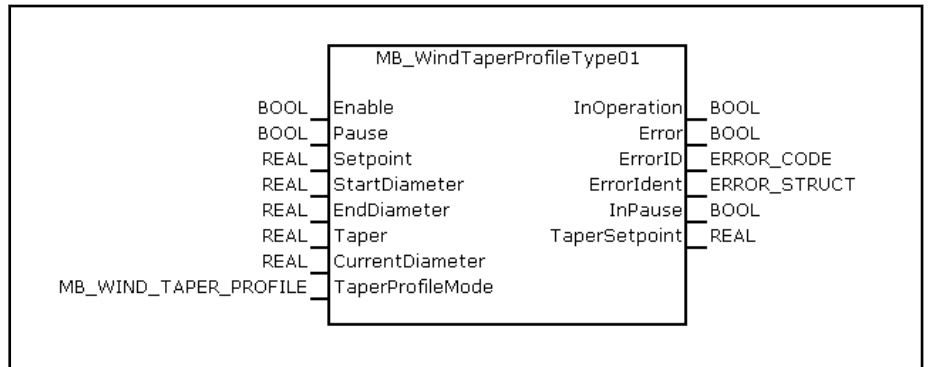


Abb.9-95: Funktionsbaustein MB_WindTaperProfileType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	Pause	BOOL	FB pausiert, TaperTensionSetpoint bleibt "eingefroren"
	Setpoint	REAL	Sollwert [-] Zugspannungssollwert für einen Wickler mit Drehmomentenbegrenzung, Sollwert für Pneumatische Kraft für einen Wickler mit Tänzer
	StartDiameter	REAL	Startdurchmesser [mm], ab dem die Zugspannung angepasst werden soll
	EndDiameter	REAL	Durchmesser [mm], ab dem die Zugspannung nicht mehr angepasst wird (konstant bleibt) Gilt nur für die lineare Kennlinie
	Taper	REAL	Wickelhärte [%] bestimmt den Verlauf der Wickelhärtekennlinie. Ist der Wert zwischen 0 und 100% nimmt die Zugkraft mit steigendem Durchmesser ab. Ist der Wert zwischen 0 und -100% so nimmt die Zugkraft mit steigendem Durchmesser zu
	CurrentDiameter	REAL	Aktueller Durchmesser [mm] des Aufwicklers. (z.B. vom FB MB_WinderTensionCtrlType02)
	TaperProfileMode	MB_WIND_TAPER_PROFILE	Auswahl des Wickelhärteprofils
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Datentyp	Beschreibung
	InPause	BOOL	Wickelhärte pausiert (Zugspannungssollwert wird "eingefroren")
	TaperSetpoint	REAL	Sollwert mit Berücksichtigung der Wickelhärtekennlinie

Abb.9-96: Schnittstellenvariablen Funktionsbaustein MB_WindTaperProfileType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Setpoint	REAL	0.0	n.def.	0.0	Kontinuierlich
StartDiameter	REAL	> 0.0	n.def.	100.0	Kontinuierlich
EndDiameter	REAL	StartDiameter	n.def.	1000.0	Kontinuierlich
Taper	REAL	-100.0	100.0	0.0	Kontinuierlich
CurrentDiameter	REAL	0.0	n.def.	0.0	Kontinuierlich
TaperProfileMode	MB_WIND_TAPER_PROFILE			NO_PROFILE	Steigende Flanke an Enable

Abb.9-97: Min-/Max- und Default-Werte des Funktionsbausteins MB_WindTaperProfileType01

Element	Wert	Beschreibung
NO_PROFILE	0	Kein Profil
PROFILE_LINEAR	1	Lineares Kurvenprofil
PROFILE_HYPERBOLIC	2	Hyperbolisches Kurvenprofil

Abb.9-98: Elemente des Aufzählungstyps, MB_WIND_TAPER_PROFILE

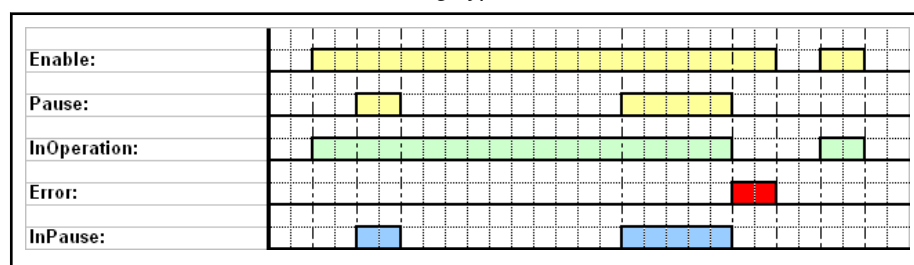
Signal-Zeit-Diagramm

Abb.9-99: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_WindTaperProfileType01

Funktionsbeschreibung

Beim Aufwickeln von Materialien ist es häufig wünschenswert, die Zugspannung mit der aufgewickelt wird, in Abhängigkeit des Wickeldurchmesser zu beeinflussen. Sehr häufig soll bei einem Aufwickelvorgang zu Beginn mit höherer Zugspannung aufgewickelt werden und mit zunehmenden Wickeldurchmesser soll die Zugspannung dann reduziert werden.

Der Baustein MB_WindTaperProfileType01 beeinflusst einen vorgebbaren Sollwert nach einer auswählbaren Kennlinie und einem vorgebbaren Wickeldurchmesser. Die Wickelhärtekennlinie kann über den Eingang Taper (Wickelhärte) noch beeinflusst werden. Ist der Wert für die Wickelhärte zwischen

RMB_TechWinder.library/MX_TechWinder.lib

0% und 100%, so nimmt die Zugspannung mit steigendem Durchmesser ab, ist der Wert zwischen 0% und -100% so nimmt die Zugspannung mit steigendem Durchmesser zu. Je größer der absolute Wert für die Wickelhärte, desto stärker ist die Ab- bzw. Zunahme der Zugspannung.

Der Baustein kann für einen Wickler mit Drehmomentbegrenzung verwendet werden, er wird dann dem Baustein MB_WinderTensionCtrlType02 vorgeschaltet, d.h. der Eingang SetpointOpMode des Bausteins MB_WinderTensionCtrlType02 wird mit dem Ausgang TaperSetpoint verschaltet.

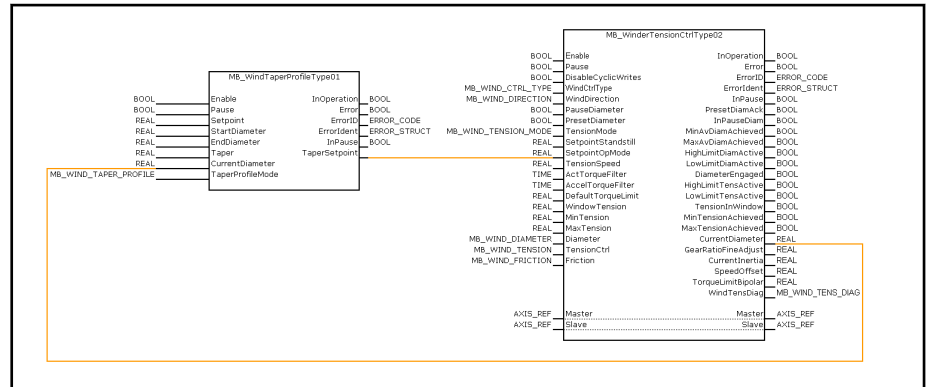


Abb.9-100: MB_WindTaperProfileType01, Verschaltung

Wird der Baustein für einen Wickler mit Tänzer verwendet, so muss der Ausgang TaperSetpoint auf die pneumatische Kraft, mit der die Tänzerwalze beaufschlagt wird, wirken (z.B. analoger Spannungswert). Der Eingang Setpoint entspricht dann auch einem Wert für die pneumatische Kraft.

Lineare Wickelhärtekenlinie:

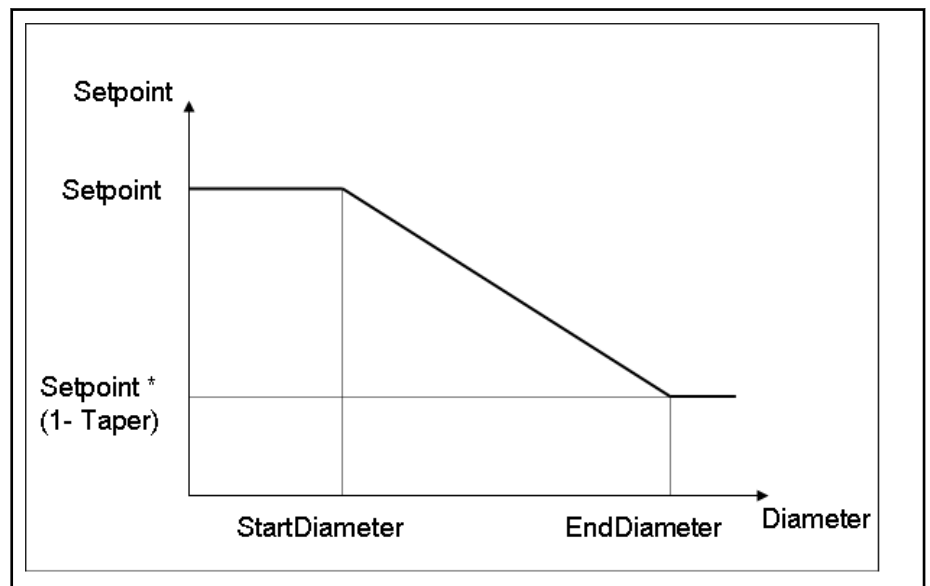


Abb.9-101: MB_WindTaperProfileType01, lineare Wickelhärtekenlinie

Hyperbolische Wickelhärtekenlinie:

RMB_TechWinder.library/MX_TechWinder.lib

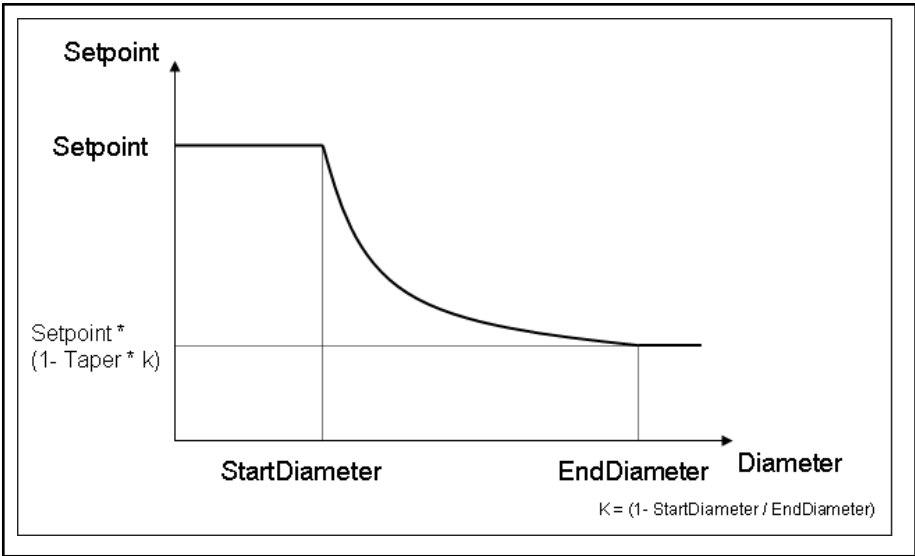


Abb.9-102: MB_WindTaperProfileType01, hyperbolische Wickelhärtekennlinie

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins aufgelistet und beschrieben. Es werden die Fehlercodes aus der "F_RELATED table (16#0170)" verwendet.

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#1150	16#0020	Fehler in der State machine
INPUT_RANGE_ERROR	16#1151	16#0001	TensionSetpoint < 0.0
INPUT_RANGE_ERROR	16#1151	16#0002	StartDiameter <= 0.0
INPUT_RANGE_ERROR	16#1151	16#0003	EndDiameter <= 0.0
INPUT_RANGE_ERROR	16#1151	16#0004	EndDiameter <= StartDiameter
INPUT_RANGE_ERROR	16#1151	16#0005	Taper < -100.0 bzw. Taper > 100.0
INPUT_RANGE_ERROR	16#1151	16#0006	CurrentDiameter <= 0.0
INPUT_RANGE_ERROR	16#1151	16#0007	falscher TaperProfile-Mode

Abb.9-103: Fehlercodes Funktionsbaustein MB_WindTaperProfileType01

- Notwendige Hardware
- IndraMotion MLC Hardware CML65, CML45, CML25
 - IndraDrive
- Notwendige Firmware
- IndraMotion MLC Firmware FWA-CMLXX*-MLC-12VRS
XX - Gerätevariante, z.B. CML65
 - IndraDrive Antriebsfirmware
 - IndraWorks 12VRS für IndraMotion MLC

Besonderheiten für IndraMotion MLC

Besonderheiten für die Steuerung IndraMotion MLC

Keine besondere Parametrierung erforderlich.

9.8.5 MB_WindSpeedControlAdaptionType01

Kurzbeschreibung

Der Baustein führt eine Anpassung der Kp-Verstärkung des Drehzahlreglers des jeweiligen Wickelantriebes in Abhängigkeit des aktuellen Trägheitsmoments des Wickels und einem minimalen und maximalen Kp-Wert durch.

RMB_TechWinder.library/MX_TechWinder.lib

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechWinder.compiled-library

Abb.9-104: Referenztable MB_WindSpeedControlAdaptionType01

Schnittstellenbeschreibung

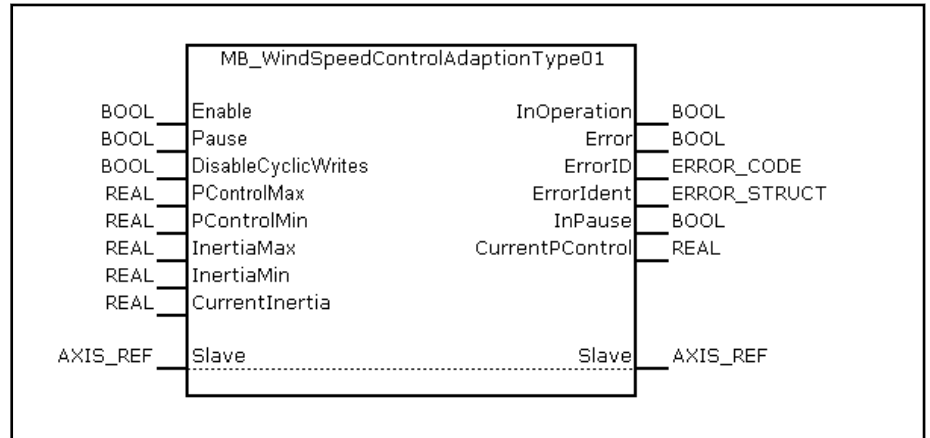


Abb.9-105: Funktionsbaustein MB_WindSpeedControlAdaptionType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Slave	AXIS_REF	Achsstruktur Wickelachse
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	Pause	BOOL	FB pausiert, CurrentPControl wird "eingefroren"
	DisableCyclicWrites	BOOL	Schreiben auf den Antriebsparameter S-0-0100 kann abgeschaltet werden
	PControlMax	REAL	Kp-Verstärkung für den maximalen Durchmesser [Nm/(rad/s)]
	PControlMin	REAL	Kp-Verstärkung für den minimalen Durchmesser [Nm/(rad/s)]
	InertiaMax	REAL	Trägheitsmoment beim maximalen Durchmesser bzw. Trägheitsmoment [kg*m²] (bezogen auf die Motorwelle)
	InertiaMin	REAL	Trägheitsmoment beim minimalen Durchmesser (Hülse) bzw. Trägheitsmoment [kg*m²] (bezogen auf die Motorwelle)
	CurrentInertia	REAL	Aktuelles Trägheitsmoment des Wickels [kg*m²] (bezogen auf die Motorwelle)
VAR_OUTPUT	InOperation	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose

RMB_TechWinder.library/MX_TechWinder.lib

E/A-Typ	Name	Datentyp	Beschreibung
	InPause	BOOL	FB pausiert (CurrentPCControl "eingefroren")
	CurrentPCControl	REAL	Aktueller Kp-Wert (Drehzahlregler) [Nm/(rad/s)]

Abb.9-106: Schnittstellenvariablen Funktionsbaustein MB_WindSpeedControlAdaptionType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an Enable
PControlMax	REAL	PControlMin	n.def.	0.0	Steigende Flanke an Enable
PControlMin	REAL	0.0	n.def.	0.0	Steigende Flanke an Enable
InertiaMax	REAL	InertiaMin	n.def.	0.0	Steigende Flanke an Enable
InertiaMin	REAL	0.0	n.def.	0.0	Kontinuierlich
CurrentInertia	REAL	0.0	n.def.	0.0	Kontinuierlich

Abb.9-107: Min-/Max- und Default-Werte des Funktionsbausteins MB_WindSpeedControlAdaptionType01

Signal-Zeit-Diagramm

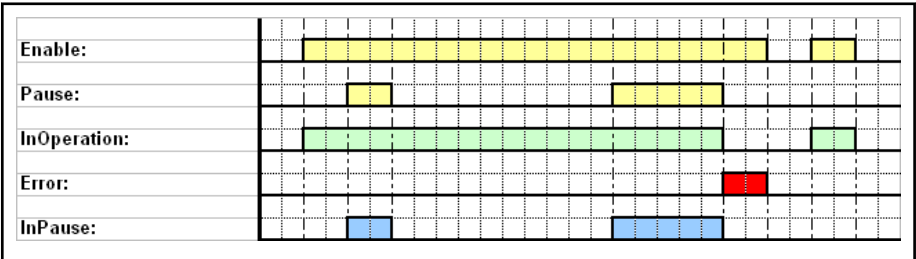


Abb.9-108: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_WindSpeedControlAdaptionType01

Funktionsbeschreibung

Der Baustein berechnet zwischen einem minimalen und maximalen Kp-Wert für den Drehzahlregler linear den Wert für das Kp des Drehzahlreglers in Abhängigkeit des aktuellen Trägheitsmomentes. Das minimale und maximale Kp müssen jeweils bei leerer Hülse und bei vollem Wickeldurchmesser ermittelt werden. Desweiteren werden der Werte für das minimale und maximale Trägheitsmoment (bei der Hülse bzw. bei vollem Wickeldurchmesser) benötigt. Diese können mit dem Funktionsbaustein MB_CalcInertiaLimitType02 ermittelt werden.

Aus diesen Werten wird in Abhängigkeit des aktuellen Trägheitsmoments des Wicklers der aktuelle Kp-Wert des Drehzahlreglers berechnet.

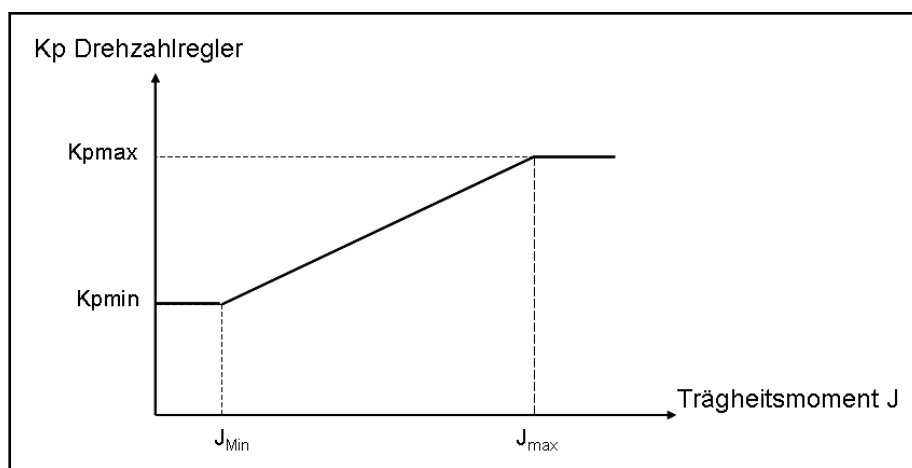


Abb.9-109: MB_WindSpeedControlAdaptionType01

Dieser aktuelle Kp-Wert wird zyklisch in den Antriebsparameter S-0-0100 "Geschwindigkeitsregler-Proportionalverstärkung" geschrieben. Dazu muss der Parameter S-0-0100 in den zyklischen Datenkanälen zwischen Steuerung und Antrieb konfiguriert werden. Wird der Baustein ausgeschaltet, bleibt der letzte Wert von "CurrentPControl" anstehen.

Mit dem Eingang "DisableCyclicWrites" kann dieses Schreiben abgeschaltet werden. Der Ausgang "CurrentPControl" wird dann nicht vom Baustein intern geschrieben, sondern muss vom SPS-Programm in den Antriebsparameter transferiert werden.



Beim Einschalten der Steuerung werden die zyklischen Datenkanäle zwischen Steuerung und Antrieb immer mit Null initialisiert. Ist der Parameter S-0-0100 "Geschwindigkeitsregler-Proportionalverstärkung" zyklisch konfiguriert, muss nach dem Einschalten der Steuerung dafür gesorgt werden, dass der entsprechende zyklische Kanal mit einem sinnvollen Wert vorinitialisiert wird. Dieser gilt dann solange, bis der Baustein eingeschaltet wurde.

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins aufgelistet und beschrieben. Es werden die Fehlercodes aus der "F_RELATED table (16#0170)" verwendet.

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR	16#11E0	16#0020	Fehler in der State machine
INPUT_RANGE_ERROR	16#11E1	16#01	PControlMax < 0.0
INPUT_RANGE_ERROR	16#11E1	16#02	PControlMin < 0.0
INPUT_RANGE_ERROR	16#11E1	16#03	InertiaMax < 0.0
INPUT_RANGE_ERROR	16#11E1	16#04	CurrentInertia < 0.0
INPUT_RANGE_ERROR	16#11E1	16#05	InertiaMin < 0.0
INPUT_RANGE_ERROR	16#11E1	16#06	PControlMax <= PControlMin
INPUT_RANGE_ERROR	16#11E1	16#07	InertiaMax <= InertiaMin
DEVICE_ERROR	16#11E2	16#01	Achse nicht in BB

Abb.9-110: Fehlercodes Funktionsbaustein MB_WindSpeedControlAdaptionType01

Notwendige Hardware • IndraMotion MLC Hardware CML65, CML45, CML25

RMB_TechWinder.library/MX_TechWinder.lib

Notwendige Firmware

- IndraDrive
- IndraMotion MLC Firmware FWA-CMLXX*-MLC-12VRS
XX - Gerätevariante, z.B. CML65

Notwendige Software

- IndraDrive Antriebsfirmware
- IndraWorks 12VRS für IndraMotion MLC

Besonderheiten für IndraMotion MLC

Im zyklischen SERCOS-Kanal des entsprechenden Antriebs muss der **Parameter S-0-0100 ("Geschwindigkeitsregler-Proportionalverstärkung")** eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.



Die Initialwerte für Parameter im zyklischen SERCOS-Kanal sind immer '0'. D.h. beim Einschalten der Steuerung ist der Wert für den Parameter S-0-0100 "Geschwindigkeitsregler-Proportionalverstärkung" Null wenn dieser im zyklischen SERCOS-Kanal eingetragen ist, und der Funktionsbaustein nicht enabled ist. Um dies zu verhindern, kann beim Einschalten der Steuerung der entsprechende Datencontainer einmalig mit einem Wert initialisiert werden, z.B.

```
AxisData[Antriebsname.AxisNo].dwUserCmdDataX_q.REAL_ := 10.0;
```

9.9 Parameterübersicht

In diesem Abschnitt werden die in der Dokumentation verwendeten Parameter aufgelistet.

Für die Steuerung IndraMotion MLC gibt es Achsparameter (A-Parameter), sowie für den jeweiligen Antrieb Produktspezifische Parameter (P-Parameter) und Standardparameter (S-Parameter).

Für die Steuerung IndraMotion MLD gibt es nur Produktspezifische Parameter (P-Parameter) und Standardparameter (S-Parameter).

Parameter	Bezeichnung
A-0-0029	Lage-Polaritäten
A-0-0038	Drehmoment/Kraft-Grenzwert bipolar
A-0-0050	Wichtungsart für Drehmoment-/Kraftdaten
A-0-0102	Geschwindigkeits-Istwert
A-0-0500	Konfiguration benutzerdefinierter Sollwert A
A-0-0502	Konfiguration benutzerdefinierter Sollwert B
A-0-0504	Konfiguration benutzerdefinierter Sollwert C
A-0-0506	Konfiguration benutzerdefinierter Sollwert D
A-0-0520	Konfiguration benutzerdefinierter Istwert A
A-0-0522	Konfiguration benutzerdefinierter Istwert B
A-0-0524	Konfiguration benutzerdefinierter Istwert C
A-0-0526	Konfiguration benutzerdefinierter Istwert D
A-0-2798	Polarität Leitachsposition

Abb.9-111: IndraMotion MLC-Achsparameter (A-Parameter)

RMB_TechWinder.library/MX_TechWinder.lib

Parameter	Bezeichnung
S-0-0040	Geschwindigkeits-Istwert
S-0-0043	Geschwindigkeitspolaritäten-Parameter
S-0-0055	Lage-Polaritäten
S-0-0080	Drehmoment/Kraft-Sollwert
S-0-0084	Drehmoment/Kraft-Istwert
S-0-0085	Drehmoment/Kraft-Polaritäten-Parameter
S-0-0091	Geschwindigkeits-Grenzwert bipolar
S-0-0092	Drehmoment/Kraft-Grenzwert bipolar
S-0-0100	Geschwindigkeitsregler-Proportionalverstärkung
S-0-0121	Lastgetriebe-Eingangsumdrehungen
S-0-0122	Lastgetriebe-Ausgangsumdrehungen

Abb.9-112: Standardparameter (S-Parameter)

Parameter	Bezeichnung
P-0-0108	Polarität Leitantrieb
P-0-0665	Achsregler-Konfiguration
P-0-0690	Geschwindigkeits-Sollwert additiv
P-0-0694	Feinabgleich Getriebeübersetzung, Prozessregler

Abb.9-113: Produktspezifische Parameter (P-Parameter)

Parameter	Bezeichnung
P-0-1623	CCD: Konfiguration Master Sollwerte
P-0-1624	CCD: Konfiguration Slave Istwerte
P-0-1625	CCD: Konfigurationsliste Slave-Sollwerte
P-0-1626	CCD: Konfigurationsliste Slave-Istwerte
P-0-1730	CCD: MDT-Echtzeitcontainer 1, Master
P-0-1731-39	CCD: MDT-Echtzeitcontainer 1, Slave n (n = 1-9)
P-0-1740	CCD: MDT-Echtzeitcontainer 2, Master
P-0-1741-49	CCD: MDT-Echtzeitcontainer 2, Slave n (n = 1-9)
P-0-1750	CCD: MDT-Echtzeitcontainer 3, Master
P-0-1751-59	CCD: MDT-Echtzeitcontainer 3, Slave n (n = 1-9)
P-0-1760	CCD: MDT-Echtzeitcontainer 4, Master
P-0-1761-69	CCD: MDT-Echtzeitcontainer 4, Slave n (n = 1-9)
P-0-1770	CCD: AT-Echtzeitcontainer 1, Master
P-0-1771-79	CCD: AT-Echtzeitcontainer 1, Slave n (n = 1-9)
P-0-1780	CCD: AT-Echtzeitcontainer 2, Master
P-0-1781-89	CCD: AT-Echtzeitcontainer 2, Slave n (n = 1-9)
P-0-1790	CCD: AT-Echtzeitcontainer 3, Master

RMB_TechWinder.library/MX_TechWinder.lib

Parameter	Bezeichnung
P-0-1791-99	CCD: AT-Echtzeitcontainer 3, Slave n (n = 1-9)
P-0-1800	CCD: AT-Echtzeitcontainer 4, Master
P-0-1801-09	CCD: AT-Echtzeitcontainer 4, Slave n (n = 1-9)

Abb.9-114: Produktspezifische Parameter (P-Parameter)

10 RMB_TechCrossCutCrossSeal.library

10.1 MB_CrossCutSealType01

Kurzbeschreibung Der Funktionsbaustein MB_CrossCutSealType01 berechnet, lädt und aktiviert ein Bewegungsprofil für Quersiegler- oder Querschneider-Anwendungen. Der Funktionsbaustein nutzt intern für die Berechnung den Funktionsbaustein [MB_CrossCutterCalcType04](#), Seite 530.

Mit dem errechneten Bewegungsprofil bewegt sich die Slaveachse innerhalb des Siegel- bzw. Schneidebereiches mit Produktgeschwindigkeit. Innerhalb dieses Bereiches können zusätzliche Funktionen, wie z.B. OverSpeed, PushOut oder Kosinus-Korrektur optional zugeschaltet werden.

Außerhalb des Siegel- bzw. Schneidebereiches wird eine nahezu beschleunigungs- und verlustenergieoptimale Ausgleichsbewegung ausgeführt, wenn die vorgegebene Formatlänge nicht dem Messer bzw. Siegelbackenabstand entspricht. Weiterhin kann ein Pendeln der Quersiegler- bzw. Querschneiderachse in frei wählbaren Grenzen vorgegeben werden, um bei großen Formatlängen das Beschleunigungsvermögen des Antriebes optimal zu nutzen.

Im Betrieb können Eingangsgrößen (wie z.B. Formatlänge) verändert werden, wodurch im Hintergrund automatisch ein neues Bewegungsprofil errechnet wird. Das neu errechnete Bewegungsprofil wird anschließend aktiviert und am Umschaltwinkel wirksam.

Bei negativer Flanke an Enable wird der Quersiegler- bzw. Querschneider deaktiviert. Hierbei wird die Achse sofort über die vorgegebene Verzögerungsrampe angehalten.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechCrosscutCrossseal.compiled-library
IndraMotion MLD/MPx07 mit MA Funktionspaket	MX_Technology07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb. 10-1: Referenztable MB_CrossCutSealType01

RMB_TechCrossCutCrossSeal.library

Schnittstellenbeschreibung

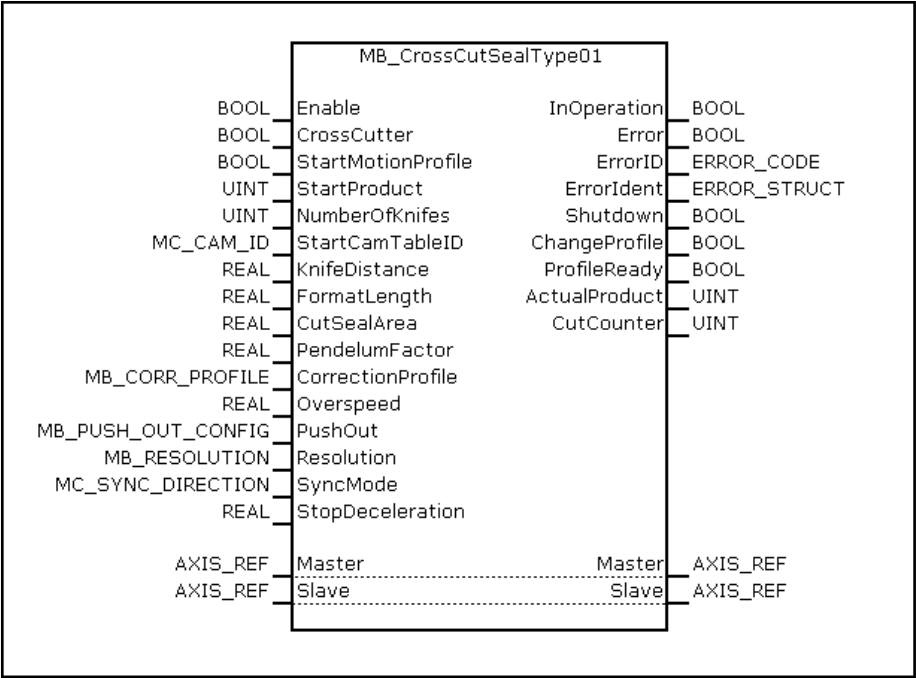


Abb. 10-2: Funktionsbaustein MB_CrossSealType01

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (= Querschneider- / Quersieglerachse) Der Modulowert beträgt 360° elektrisch und ist mechanisch definiert als Bereich zwischen zwei Messern/ Siegelbacken
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins

E/A-Typ	Name	Typ	Kommentar
	CrossCutter	BOOL	Ist CrossCutter = TRUE, wird der Querschneider-Anwendungsfall (siehe Detailbeschreibung unten) ausgewählt. In diesem Anwendungsfall ist die Masterachse so zu normieren, dass in einer Umdrehung das zu schneidende Material die Strecke zurücklegt, die dem Messerabstand entspricht. In diesem Fall wird das verwendete elektronische Getriebe des Bewegungsprofils mit dem Übersetzungsverhältnis "FormatLength/ KnifeDistance" eingestellt. Weiterhin wirkt der Eingang CutSealArea als Schnittwinkel des Querschneiders und die Eingangsgrößen ProductsPerCut und StartProduct wirken nicht. Ist CrossCutter = FALSE, wird der Quersiegler-Anwendungsfall (siehe Detailbeschreibung unten) ausgewählt. In diesem Anwendungsfall ist die Masterachse so zu normieren, dass eine Umdrehung einem Folien-Vorschub von einer Produktlänge entspricht. In diesem Fall wird das verwendete elektronische Getriebe des Bewegungsprofils mit dem Übersetzungsverhältnis "1 / ProductsPerCut" eingestellt. Weiterhin entspricht der Eingang CutSealArea der Siegellänge und wird damit in translatorischen Einheiten (z.B. mm oder Inch) vorgegeben. Die Eingänge ProductsPerCut und StartProduct werden in diesem Fall ausgewertet
	StartMotionProfile	BOOL	Die Aktivierung eines neuen Bewegungsprofils wird blockiert, solange dieser Eingang = FALSE ist. StartMotionProfile = TRUE aktiviert das neue Bewegungsprofile, wenn der Ausgang ProfileReady = TRUE ist
	ProductsPerCut	UINT	Eingang ist nur bei CrossCutter = FALSE wirksam. Anzahl der Produkte pro Siegelvorgang. Entspricht der Anzahl der Masterumdrehungen pro Siegel- bzw. Schneidvorgang
	StartProduct	UINT	Eingang ist nur bei CrossCutter = FALSE wirksam. Bei mehr als einem Produkt pro Siegel- bzw. Schneidvorgang (ProductsPerCut > 1) kann über diesen Eingang das Start-Produkt bei Aktivierung des FB angegeben werden
	NumberOfKnives	UINT	Anzahl der Siegelbacken bzw. Messer die auf der Siegel- bzw. Schneidwalze am Kreisumfang vorhanden sind. Die Siegelbacken bzw. Messer sind in gleichen Abständen angeordnet
	StartCamTableID	MC_CAM_ID	Startnummer der ersten Kurvenscheibe Punktetabelle. Es wird ein Block von maximal 6 Punkttabellen benötigt

RMB_TechCrossCutCrossSeal.library

E/A-Typ	Name	Typ	Kommentar
	KnifeDistance	REAL	Messer- bzw. Siegelbackenabstand. Entspricht dem Kreisumfang dividiert mit der Anzahl der Messer oder Siegelbacken [mm oder Inch]
	FormatLength	REAL	Vorgegebene Schnittlänge bzw. Siegelabstand/Packungslänge [mm oder Inch]
	CutSealArea	REAL	Innerhalb dieses Bereiches bewegt sich das Messer bzw. die Siegelbacke mit der Produktgeschwindigkeit. Die Geschwindigkeit innerhalb dieses Fensters kann durch zusätzliche Eingänge, wie z.B. OverSpeed, PushOut, Kosinus Korrektur... beeinflusst werden. Ist der Eingang CrossCutter = FALSE, entspricht dieser Eingang der Siegellänge und wird damit in translatorischen Einheiten vorgegeben (z.B. in mm oder Inch). Ist der Eingang CrossCutter = TRUE, wirkt dieser Eingang rotatorisch als Winkel in [°]
	PendelumFactor	REAL	Maximaler Pendelweg in [%] 0%: kein Pendeln (Stillstand bei großer Formatlänge) 100%: maximales Pendeln (Pendeln bis an die Grenze des Siegel- bzw. Scheidebereiches möglich)
	CorrectionProfile	MB_CORR_PROFILE	Auswahl eines Spezialprofils. Zur Parametrierung des Spezialprofils müssen die Eingänge "OverSpeed" oder "PushOut" verwendet werden (siehe Kap. 10.2.4 "MB_CORR_PROFILE" auf Seite 541)
	Overspeed	REAL	Konstante Geschwindigkeitsüberhöhung bzw. Verringerung im Schnittbereich in [%] der Leitachsgeschwindigkeit 0%: kein Overspeed 100%: doppelte Leitachsgeschwindigkeit -50%: halbe Leitachsgeschwindigkeit Dieser Eingang ist nur wirksam wenn die Option CORR_PROFILE_OVERSPEED an dem Eingang CorrectionProfile angewählt ist
	PushOut	MB_PUSH_OUT_CONFIG	Struktur zur Definition einer PushOut Bewegung während des Schnitts bzw. Siegelung (siehe Kap. 10.2.6 "MB_PUSH_OUT_CONFIG" auf Seite 543). In CorrectionProfile muss die Option CORR_PROFILE_PUSH_OUT oder CORR_PROFILE_COS_COMP_AND_PUSH_OUT angewählt sein. Ansonsten ist PushOut deaktiviert und somit unwirksam
	Resolution	MB_RESOLUTION	Angabe der Auflösung der Kurvenscheibenberechnung über die Festlegung der Anzahl an Stützstellen in 3 Stufen (Details siehe Kap. 10.2.5 "MB_RESOLUTION" auf Seite 542)
	SyncMode	MC_SYNC_DIRECTION	Aufsynchronisieren der Richtung (Shortest_Way, Catch_Up, Slow_Down)

RMB_TechCrossCutCrossSeal.library

E/A-Typ	Name	Typ	Kommentar
	StopDeceleration	REAL	Mit negativer Flanke von Enable wird die Slaveachse mit der hier definierten Verzögerung angehalten. Einheiten gemäß Achswichtung in rotatorischen Einheiten
VAR_OUTPUT	InOperation	BOOL	Querscheider läuft im fehlerfreien Betrieb
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Fehlerbehandlung, Seite 529
	Shutdown	BOOL	Quersiegler wird nach negativer Flanke von Enable definiert mit der Verzögerungsrampe StopDeceleration angehalten. In dieser Zeit ist der ShutDown-Ausgang gesetzt um das definierte Anhalten anzuzeigen
	ChangeProfile	BOOL	Werden Eingangsgrößen wie z.B. FormatLength im laufenden Betrieb verändert, wird im Hintergrund ein neues Bewegungsprofil gerechnet und geladen. Anschließend wird auf das neue Bewegungsprofil umgeschaltet. Da dieser Vorgang mehrere Zyklen dauert, wird in dieser Zeit dieser Ausgang gesetzt. Werden während diesem Vorgang erneut Eingangsgrößen (z.B. FormatLength) verändert, wird der aktuelle Vorgang abgebrochen und erneut mit den neuen Daten durchgeführt. Der Ausgang wird zurückgesetzt, sobald das neu berechnete Bewegungsprofil wirksam wird und damit auch der Umschaltwinkel durchlaufen wurde
	ProfileReady	BOOL	Dieser Ausgang wird gesetzt, sobald das Bewegungsprofil berechnet geladen und aktiviert ist. Dieser Ausgang wird gelöscht sobald das Bewegungsprofil wirksam ist (nach Durchlaufen des Umschaltwinkels)
	ActualProduct	UINT	Bei mehr als einem Produkt pro Siegelvorgang (ProductsPerCut > 1) wird über diesen Ausgang das aktuell bearbeitete Produkt angezeigt. Bei steigender Flanke von Enable wird dieser Ausgang auf die Eingangsgröße StartProduct zurückgesetzt
	CutCounter	UINT	Bei jedem Moduloüberlauf (360° → 0°) der Slaveachse wird der CutCounter erhöht. Mit steigender Flanke von Enable wird der CutCounter auf 0 zurückgesetzt

Abb. 10-3: Schnittstellenvariablen Funktionsbaustein MB_CrossSealType01

Min-/Max-Werte und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
CrossCutter	BOOL			FALSE	Steigende Flanke an Enable
StartMotionProfile	BOOL			TRUE	kontinuierlich

RMB_TechCrossCutCrossSeal.library

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ProductsPerCut	UINT	1	32	1	Steigende Flanke an Enable
StartProduct	UINT	1	32	1	Steigende Flanke an Enable
NumberOfKnives	UINT	1	8	1	Steigende Flanke an Enable
StartCamTableID	MC_CAM_ID	1	94 (bei MLC) 3 (bei MLD)	1	Steigende Flanke an Enable
KnifeDistance	REAL	> 0.0mm/Inch	n. def.	100.0mm/Inch	Steigende Flanke an Enable
FormatLength	REAL	> 0.2* KnifeDistance	30* KnifeDistance	100.0mm/Inch	kontinuierlich*
CutSealArea	REAL	0° 0mm	< 360° < KnifeDistance	10° 10mm/Inch	kontinuierlich*
PendulumFactor	REAL	0.0%	100%	0.0%	kontinuierlich*
CorrectionProfile	MB_CORR_PROFILE	0	4	0	kontinuierlich*
Overspeed	REAL	- 50.0%	+100.0%	0.0%	kontinuierlich*
PushOut	MB_PUSH_OUT_CONFIG				kontinuierlich*
Resolution	MB_RESOLUTION	RESOLUTION_LOW	RESOLUTION_HIGH	RESOLUTION_MEDIUM	Steigende Flanke an Enable
SyncMode	MC_SYNC_DIRECTION	0	2	Catch_Up	Steigende Flanke an Enable
StopDeceleration	REAL	> 0.0	n. def.	1000.0	kontinuierlich

* Übernahme der Eingänge dauert einige Zyklen und wird über den Ausgang "ChangeProfile" angezeigt

Abb. 10-4: Min-/Max- und Default-Werte Funktionsbaustein MB_CrossSeal-Type01

Funktionsbeschreibung

Nach Bearbeitungs freigabe mittels **Enable** berechnet der Funktionsbaustein ein Bewegungsprofil und Punktetabellen für die Applikation Quersiegler/Querschneider gemäß den angelegten Eingangsdaten. Der Funktionsbaustein unterstützt hierbei auch mehrnutzige Quersiegler/Querschneider mit gleichmäßig verteilten Messern bzw. Siegelbacken.



Für mehrnutzige Querschneider bzw. Quersiegler:

Bei mehrnutzigen Querschneider bzw. Quersiegleranwendungen muss die Parametrierung des mech. Getriebes angepasst werden (siehe auch [Notwendige Randbedingungen und Voraussetzungen](#), Seite 528).

Der errechnete Bewegungsablauf ist dazu in die folgenden zwei Abschnitte unterteilt:

- Der **Schneid- bzw. Siegelbereich** ist durch die Eingangsgröße **CutSealArea** festgelegt. In diesem Bereich bewegt sich das Messer bzw. die Siegelbacke mit Produktgeschwindigkeit. Zusätzlich können

Funktionen wie zum Beispiel **OverSpeed** für Geschwindigkeitsüberhöhung oder Reduzierung, **PushOut** für eine Abtrennbewegung, oder die **Kosinus-Kompensation** optional hinzugeschaltet werden.

In diesem Bereich sind folgende Eingangsgrößen wirksam:

- **CorrectionProfile**
bietet Spezialprofile im Schneid- und Siegelbereich an (Overspeed, Pushout, Kosinus Korrektur, PushOut + Kosinus Korrektur).
- **OverSpeed**
gibt die Geschwindigkeitserhöhung bzw. Reduzierung im **CutSealArea** an [%]
- **PushOut**
legt eine spezielle PushOut-Bewegung während des Schneid- bzw. Siegelvorganges fest
- **Außerhalb** des Schneid- bzw. Siegelbereich wird eine ruck- und energieoptimale Ausgleichsbewegung durchgeführt, wenn die vorgegebene Formatlänge **FormatLength** nicht dem Messer bzw. Siegelbackenabstand **KnifeDistance** entspricht. Weiterhin kann ein **Pendeln** der Quersiegler- bzw. Querschneiderachse in einstellbaren Grenzen vorgegeben werden, um bei großen Formatlängen das Beschleunigungsvermögen des Antriebes optimal zu nutzen.

In diesem Bereich ist folgende Eingangsgröße wirksam:

- **PendelumFactor** gibt an, wieviel Pendelweg maximal zugelassen wird [%]

Das Bewegungsprofil und die Punktetabellen werden nach der Berechnung automatisch geladen und aktiviert. In dieser Zeit ist der Ausgang **ChangeProfile** gesetzt. Ist das Bewegungsprofil wirksam, und die Slaveachse aufsynchronisiert, wird der Ausgang **InOperation** gesetzt. Die Slaveachse bewegt sich in diesem Zustand synchron zu dem Master unter Verwendung des zuvor berechneten Bewegungsprofiles. In diesem Zustand werden mit jeder Umdrehung der Querschneider- bzw. Quersieglerachse der Schnitt bzw. Siegelzähler **CutCounter** erhöht.

Werden im laufenden Betrieb Eingangsgrößen wie zum Beispiel **FormatLength** verändert, wird im Hintergrund ein neues Bewegungsprofil berechnet. Da dieser Vorgang mehrere Zyklen dauert, wird in dieser Zeit der Ausgang **ChangeProfile** gesetzt. Der Ausgang **ProfileReady** wird gesetzt, wenn das neue Bewegungsprofil aktiviert ist, jedoch der Umschaltwinkel noch nicht durchlaufen wurde. Wird anschließend der Umschaltwinkel erreicht, wird das berechnete Bewegungsprofil wirksam und der Ausgang **ChangeProfile** und **ProfileReady** zurückgesetzt. Werden während der laufenden Änderung des Bewegungsprofiles erneut Eingangsdaten verändert, wird die Änderung verworfen und die Berechnung mit den neuen Eingangsdaten neu aufgesetzt.

Die Stützstellen der verwendeten Punktetabellen innerhalb des Bewegungsprofiles kann über den Eingang **Resolution** beeinflusst werden. Die hohe Auflösung führt zur maximalen Genauigkeit, erfordert jedoch den höchsten Zeitbedarf bei der Berechnung. Der Zeitaufwand lässt sich mit den Einstellungen Medium sowie Low um den Faktor 3 bzw. 9 reduzieren. Die Reduzierung der Auflösung ist insbesondere dann sinnvoll wenn bei hohen Taktraten (> 500 Takte/Minute) ständig verschiedene Produktlängen bearbeitet werden.

Eine negative Flanke an **Enable** führt zum definierten Herunterfahren des Querschneiders/Quersieglers. Hierbei wird die Slaveachse mit der einstellbaren Rampe **StopDeceleration** sofort bis zum Stillstand verzögert. Während des definierten Herunterfahrens ist der Ausgang **Shutdown** gesetzt. Der Aus-

RMB_TechCrossCutCrossSeal.library

gang **Shutdown** wird zurückgesetzt, wenn die Slaveachse den Stillstand erreicht hat oder ein Fehler aufgetreten ist.

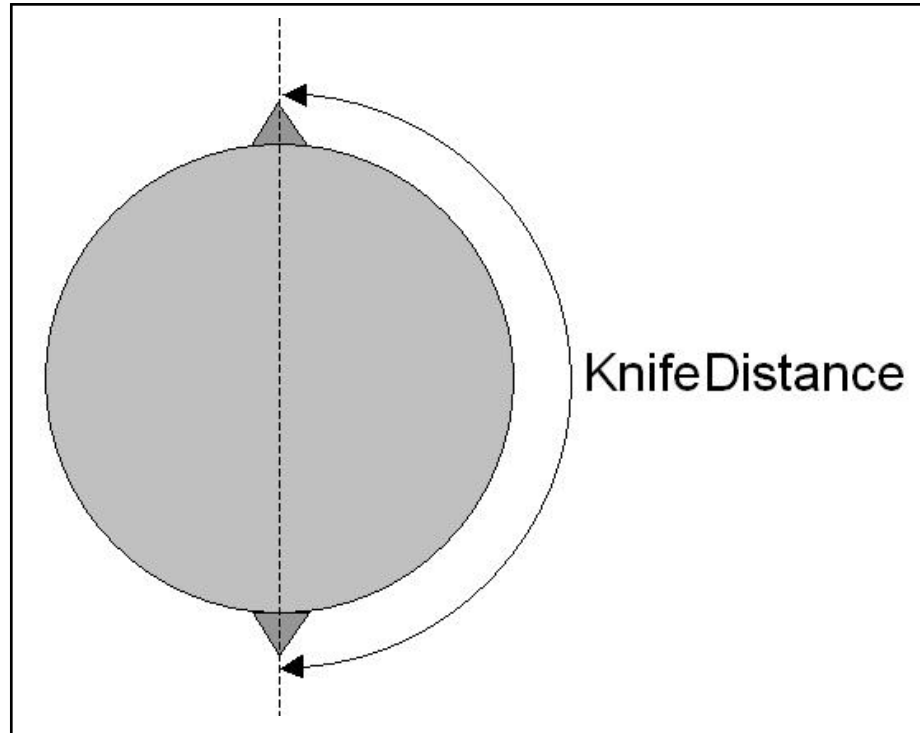


Abb. 10-5: Definition des KnifeDistance Einganges



zur Wichtung:

Die translatorischen FB-Eingangsgrößen (z.B. FormatLength) können wahlweise in "mm" oder "Inch" vorgegeben werden. Jedoch ist zu beachten, dass alle translatorischen Eingangsgrößen in der gleichen Einheit (mm oder Inch) vorgegeben werden.

Quersiegler-Anwendungsfall

Über den Eingang **CrossCutter** unterscheidet der Funktionsbaustein grundsätzlich zwischen dem Querschneider- und Quersiegler-Anwendungsfall:

Ist der Eingang **CrossCutter** = FALSE, wird von einer Quersiegelungs-Anwendung ausgegangen. Die Ansteuerung einer Quersiegelwalze von Schlauchbeutelmaschinen ist ein typisches Anwendungsbeispiel für diesen Anwendungsfall. Hierbei soll sich die Siegelbacke während des Siegelvorganges mit Produktgeschwindigkeit und optionalem Korrekturprofil bewegen. Außerhalb des Siegelbereiches soll eine Ausgleichsbewegung durchgeführt werden, wenn die Formatlänge **FormatLength** nicht dem Siegelbackenabstand **KnifeDistance** entspricht. Weiterhin kann der Quersiegler optional durch ein vorgeschaltetes taktsynchrones Ein- und Auskoppeln (LockOn/LockOff) angesteuert werden, um die Funktion "NoGap/NoSeal" und "NoProduct/NoBag" umzusetzen.

Der Quersiegler-Anwendungsfall ist durch folgende Eigenschaften gekennzeichnet:

- Der Eingang **CutSealArea** entspricht der Länge der Siegelnaht und wird damit in translatorischen Einheiten (mm oder Inch) vorgegeben
- Eine Masterumdrehung (360°) entspricht dem Folienvorschub von einer Produktlänge
- Ein Siegelvorgang kann mehrere Produkte enthalten. Die Anzahl der Produkte pro Siegelvorgang wird über den Eingang **ProductsPerCut** vor-

RMB_TechCrossCutCrossSeal.library

gegeben. Dieser Eingang wirkt direkt auf das elektronische Getriebe des verwendeten Bewegungsprofiles

- Über den Eingang **StartProduct** wird das erste Produkt nach positiver Flanke von Enable eingestellt. Diese Größe wirkt intern auf einen Positions-Offset des verwendeten Bewegungsprofiles. Nach positiver Flanke von Enable wird der Ausgang **ActualProduct** auf den Eingang **StartProduct** gesetzt
- Optional kann das taktsynchrone Ein- und Auskoppeln (LockOn/LockOff) über einen weiteren Funktionsbaustein MB_CamLock vorgeschaltet werden

**zum Start des FB:**

Stehen Master- und Slaveachse bei Start des FB an einer beliebigen Position, wird die Slaveachse ein dynamisches Aufsynchronisieren ausführen, um sich auf die Masterachse auszurichten. Ist dies nicht gewünscht, müssen vor dem Start des FBs die Master- und Slaveachse auf die Position 0° oder 180° gestellt werden.

Die nachfolgende Abbildung zeigt die Leitachsstruktur und das Wirkschema des FB für den Quersiegler-Anwendungsfall mit optionalem taktsynchronen Ein- und Auskoppeln (LockOn/LockOff):

RMB_TechCrossCutCrossSeal.library

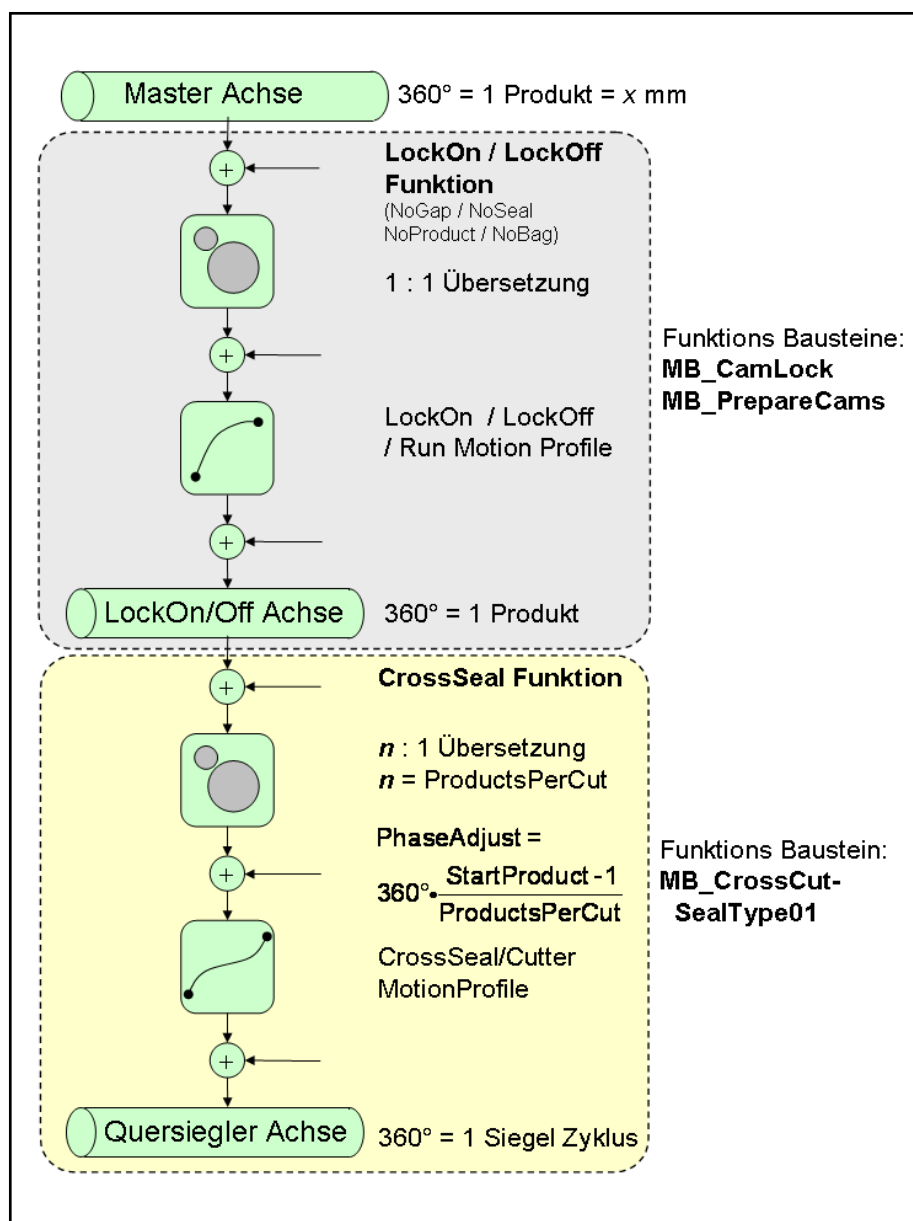


Abb. 10-6: Leitachsstruktur für Quersiegler-Anwendung mit LockOn/LockOff

Die nachfolgende Abbildung zeigt ein exemplarisches Signal-Zeit-Diagramm der Master- und Quersieglerachse für ein Produkt pro Packung:

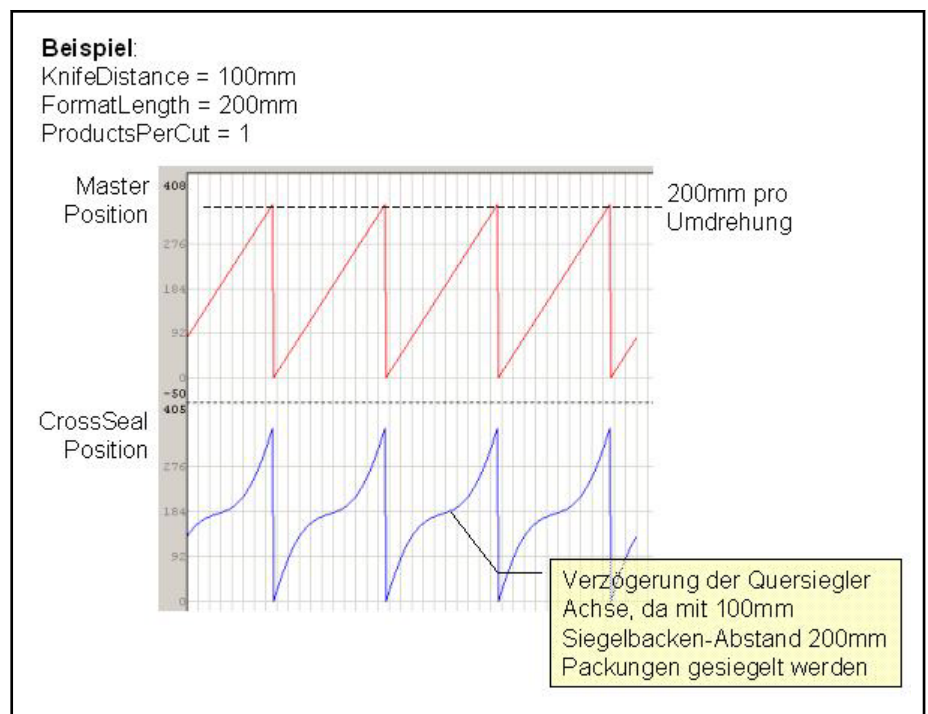


Abb. 10-7: Beispiel Signal-Zeit-Diagramm mit einem Produkt pro Siegelvorgang

Die nachfolgende Abbildung zeigt ein exemplarisches Signal-Zeit-Diagramm der Master- und Quersiegler-Achse für drei Produkte pro Packung:

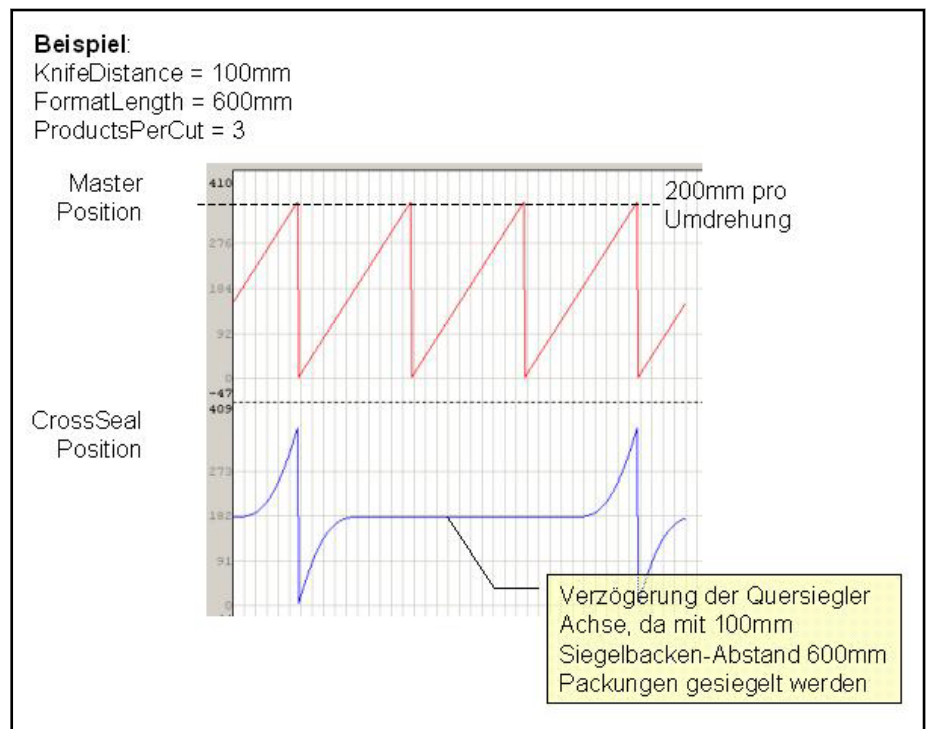


Abb. 10-8: Beispiel Signal-Zeit-Diagramm mit drei Produkten pro Siegelvorgang

Querschneider-Anwendungsfall

Über den Eingang **CrossCutter** unterscheidet der Funktionsbaustein grundsätzlich zwischen dem Querschneider- und Quersiegler-Anwendungsfall.

Ist der Eingang **CrossCutter** = TRUE, wird von einer Querschneider-Anwendung ausgegangen. Während des Schneidevorganges soll sich hierbei das

RMB_TechCrossCutCrossSeal.library

Messer mit Produktgeschwindigkeit und optionalem Korrekturprofil bewegen. Außerhalb des Schneiderbereiches soll eine Ausgleichsbewegung durchgeführt werden, wenn die Formatlänge **FormatLength** nicht dem Siegelbackenabstand **KnifeDistance** entspricht.

Der Querschneider-Anwendungsfall ist durch folgende Eigenschaften gekennzeichnet:

- Der Eingang **CutSealArea** entspricht dem Winkelbereich des Querschneiders indem er sich mit Produktgeschwindigkeit und dem optionalem Korrekturprofil bewegt
- Eine Master-Umdrehung entspricht der Material-Vorschublänge, die dem Messerabstand **KnifeDistance** entspricht
- Es wird immer von einem Schnitt pro Produkt ausgegangen. Der Eingänge **ProductsPerCut** und **StartProduct** sind nicht wirksam (diese Eingänge werden intern fest auf 1 gelegt)
- Der Ausgang **ActualProduct** ist immer auf "1" festgelegt
- Das verwendete elektronische Getriebe des Bewegungsprofiles wird mit dem Übersetzungsverhältnis **FormatLength / KnifeDistance** eingestellt.

Dieses Getriebe lässt nur ganzzahlige Werte von 1 bis 65535 für Zähler und Nenner der Übersetzung zu. Somit ergibt sich, bedingt durch die Quantisierung, eine maximale Schnittlängenabweichung.

Tritt durch die Quantisierung eine Schnittlängenabweichung auf, wird ein entsprechender Getriebe-Feinabgleich berechnet und vorgegeben. In diesem Falle entsteht jedoch systembedingt eine Drift zwischen Master- und Slave-Achse.

Die nachfolgende Abbildung zeigt die Leitachsstruktur und das Wirkschema des FB für die Querschneider-Anwendung:

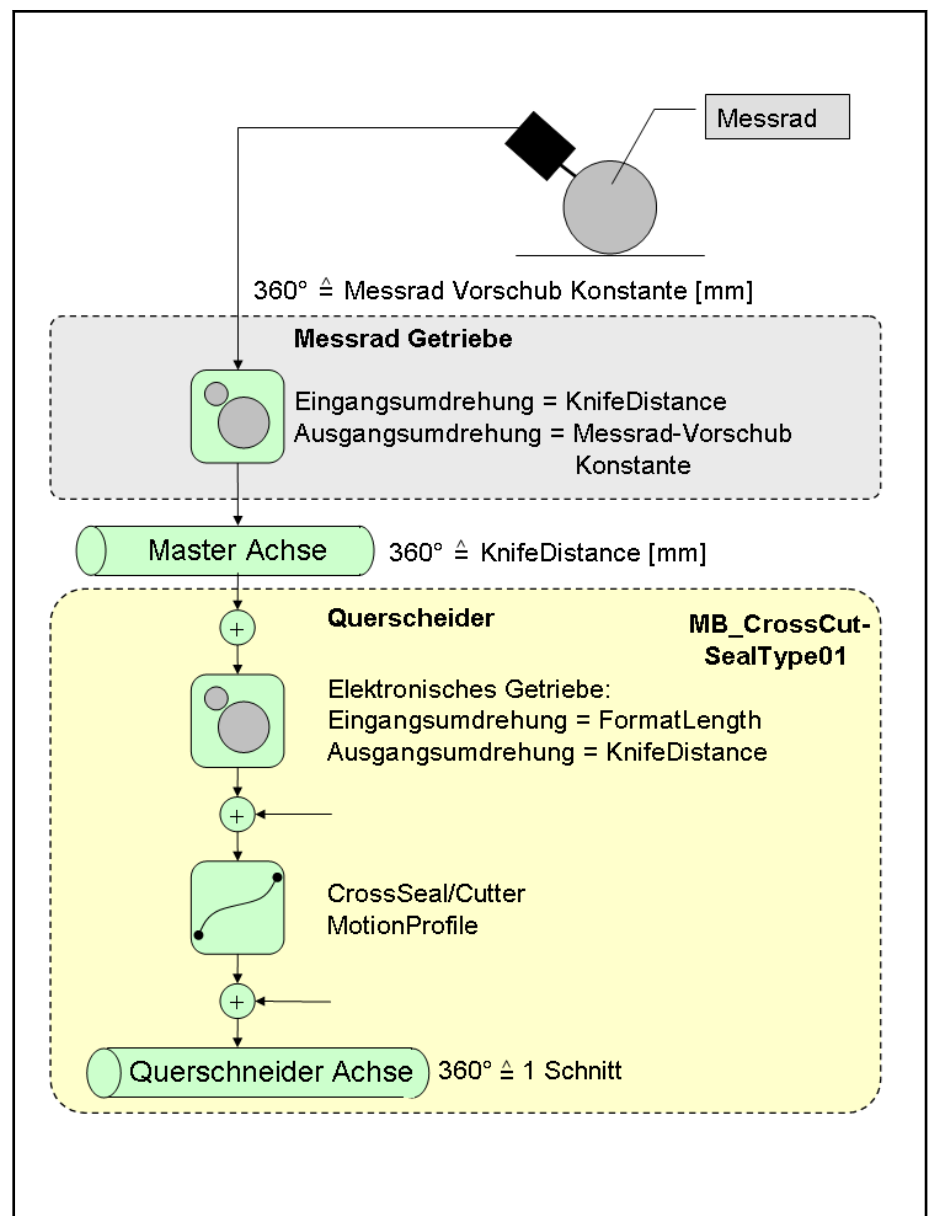


Abb. 10-9: Leitachsstuktur für die Querschneider-Anwendung

Die nachfolgende Abbildung zeigt ein exemplarisches Signal-Zeit-Diagramm der Messrad-, Master- und Querschneiderachse für ein kurzes Format (Formatlänge = 0.5 * Messerabstand):

RMB_TechCrossCutCrossSeal.library

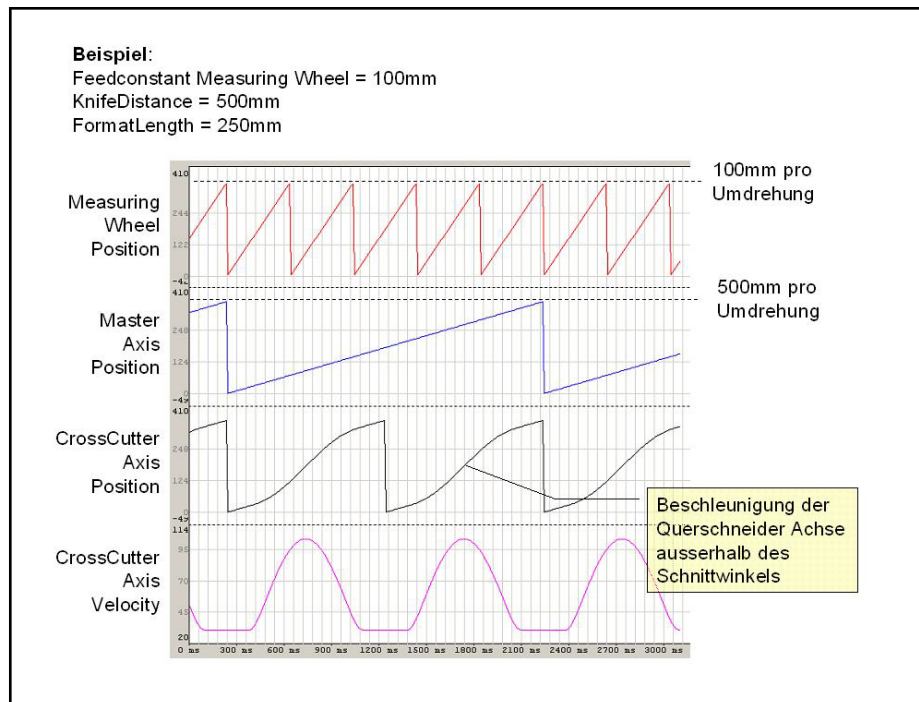


Abb. 10-10: Beispiel Signal-Zeit-Diagramm für Formatlänge = 0.5 * Messerabstand

Die nachfolgende Abbildung zeigt ein exemplarisches Signal-Zeit-Diagramm der Messrad-, Master- und Querschneiderachse für ein langes Format (Formatlänge = 2 * Messerabstand):

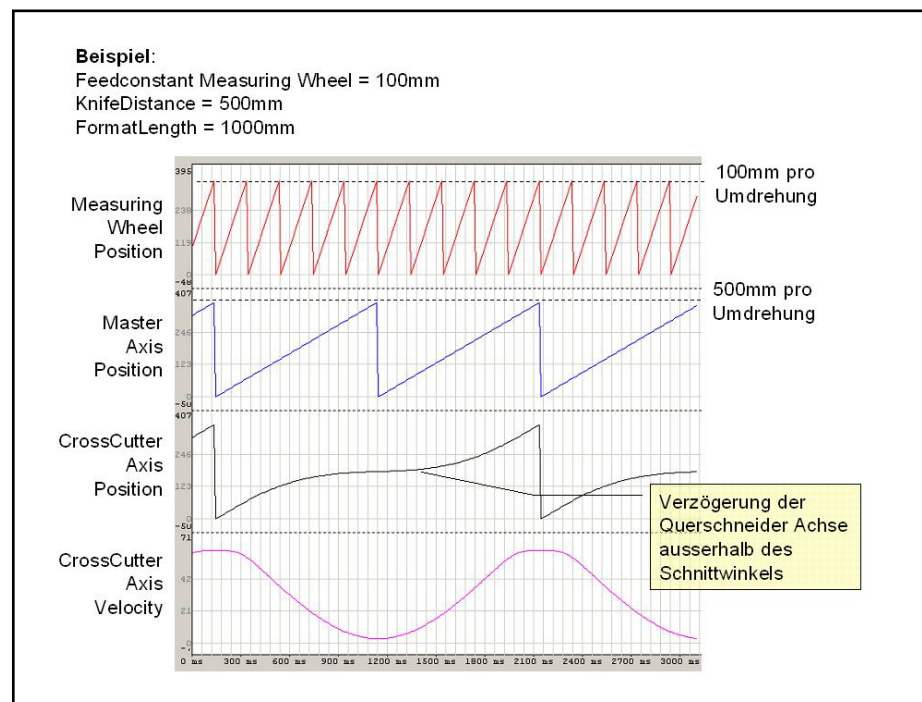


Abb. 10-11: Beispiel Signal-Zeit-Diagramm für Formatlänge = 2 * Messerabstand

Notwendige Randbedingungen und Voraussetzungen

- Siegelbacken/Messer sind symmetrisch über den Umfang verteilt
- Die 0°-Position der Slaveachse liegt immer genau in der Mitte des CutSealArea



zur Antriebs-Parametrierung:

Bei mehr als einer Siegelbacke bzw. Messer pro Kreisumfang müssen die Einstellungen für das mechanische Getriebe (Antriebsparameter: S-0-0121, S-0-0122) angepasst werden. Diese Parameter sind so einzustellen dass 360° elektrisch (= Modulowert der Slaveachse) dem Abstand zwischen zwei Messern bzw. Siegelbacken entspricht.

Beispiel Getriebeprojektierung

Konfiguration des mechanischen Getriebes bei Verwendung eines Querschneiders mit vier äquidistant auf der Querschneiderachse verteilten Messern

Eingangsgrößen:

Mechanisches Getriebe am Motor, Eingangsumdrehungen $n_{\text{Eingang}} = 10$

Mechanisches Getriebe am Motor, Ausgangsumdrehungen $n_{\text{Ausgang}} = 1$

Messeranzahl pro Kreisumfang $n_{\text{Messer}} = 4$

Berechnung der notwendigen Getriebeeinstellung:

$$\text{mechanisches Getriebe} = \frac{\text{Eingangsumdrehungen}}{\text{Ausgangsumdrehungen}} =$$

$$= \frac{n_{\text{Eingang}}}{n_{\text{Ausgang}} \cdot n_{\text{Messer}}} = \frac{10}{1 \cdot 4} = \frac{10}{4}$$

- S-0-0121 = 10
- S-0-0122 = 4

Bewegt sich bei dieser Parametrierung die Schneidwalze um 90° mechanisch, entspricht dies 360° elektrisch (= Modulowert der Slave-Achse).

Fehlerbehandlung

Der FB führt keine eigenständige motionseitige Fehlerreaktion aus. Bei Fehler-Erkennung wird kein weiteres Motion- Kommando an die Achse gegeben und der FB über den Ausgang **Error** verlassen. Eine negative Flanke von Enable löscht den Fehlerausgang und den Fehlerzustand im FB.

Der FB verwendet die F_RELATED-TABLE (16#0170) Fehlertabelle.

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#0003	16#0000	Funktionsbaustein wurde von einem anderen Funktionsbaustein unterbrochen
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0001	Der Eingang ProductsPerCut liegt außerhalb des zulässigen Bereiches (1..32)
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0002	Der Eingang StartProduct liegt außerhalb des zulässigen Bereiches (1..32)
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0003	Der Eingang NumberOfKnives liegt außerhalb des zulässigen Bereiches (1..8)

RMB_TechCrossCutCrossSeal.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0004	Der Eingang StartCamTableID liegt außerhalb des zulässigen Bereiches (1..94)
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0005	Der Eingang KnifeDistance liegt außerhalb des zulässigen Bereiches
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0006	Der Eingang FormatLength liegt außerhalb des zulässigen Bereiches
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0007	Der Eingang CutSealArea liegt außerhalb des zulässigen Bereiches
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0008	Der Eingang PendelumFactor liegt außerhalb des zulässigen Bereiches
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0009	Der Eingang Overspeed liegt außerhalb des zulässigen Bereiches
INPUT_RANGE_ERROR (16#0006)	16#0620	16#000A	Der Eingang Resolution liegt außerhalb des zulässigen Bereiches
INPUT_RANGE_ERROR (16#0006)	16#0621	16#0000	CutSealArea ist zu klein in Verbindung mit der Masterachsgeschwindigkeit und der SERCOS-Zykluszeit

Abb. 10-12: Fehlercodes Funktionsbaustein MB_CrossCutSealType01

10.2 MB_CrossCutterCalcType04

10.2.1 Allgemeines

Kurzbeschreibung

Der Funktionsbaustein MB_CrossCutterCalcType04 stellt an seinen Ausgängen alle notwendigen Daten bereit, die in das Bewegungsprofil einer Achse geladen werden müssen, um einen Querschneider oder Quersiegler mit den in den Parametern des Funktionsbausteins ausgewählten Funktionen zu betreiben. Alle Eingänge des Bausteins werden einmalig bei steigender Flanke von "Execute" übernommen. Der Baustein schreibt zudem während seiner Laufzeit direkt in die durch die Pointer spezifizierten Arrays. Die geschriebenen Daten sind erst dann gültig, wenn der Baustein fehlerfrei durchlaufen wurde und "Done" meldet.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	RMB_TechCrosscutCrossseal.compiled-library
IndraMotion MLD/MPx07VRS mit MA Funktionspaket	MX_Technology07.lib
IndraMotion MLD/MPx08VRS mit MA Funktionspaket	MX_Technology_08.lib (in Vorbereitung)
IndraMotion MLD/MPx17VRS mit MA Funktionspaket	MX_Technology_17.lib (in Vorbereitung)

Abb. 10-13: Referenztable Funktionsbaustein MB_CrossCutterCalcType04

Schnittstellenbeschreibung

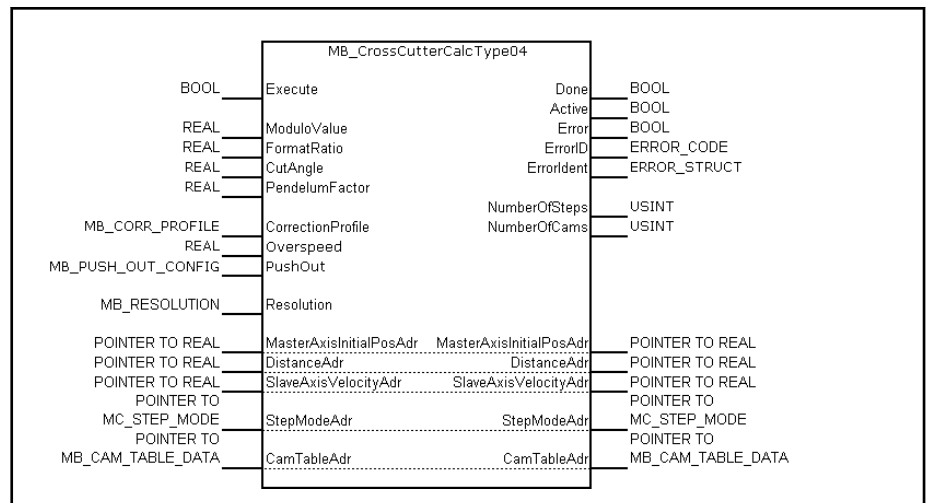


Abb. 10-14: Funktionsbaustein MB_CrossCutterCalcType04

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	MasterAxisInitialPos-Adr	POINTER TO REAL	Zeiger auf ein vom Benutzer anzulegendes Array mit drei Elementen, in das der Baustein die Leitachsenanfangspositionen der generierten Bewegungsprofilsegmente ablegt
	DistanceAdr	POINTER TO REAL	Zeiger auf ein vom Benutzer anzulegendes Array mit drei Elementen, in das der Baustein die Hübe der generierten Bewegungsprofilsegmente ablegt
	SlaveAxisVelocityAdr	POINTER TO REAL	Zeiger auf ein vom Benutzer anzulegendes Array mit drei Elementen, in das der Baustein die Geschwindigkeiten der Folgeachse an den Segmentgrenzen der Bewegungsprofilsegmente ablegt
	StepModeAdr	POINTER TO MC_STEP_MODE	Zeiger auf ein vom Benutzer anzulegendes Array mit drei Elementen, in das der Baustein die Bewegungsschrittmodi der generierten Bewegungsprofilsegmente ablegt
	CamTableAdr	POINTER TO MB_CAM_TABLE_DATA	Zeiger auf ein vom Benutzer anzulegendes Array mit drei Elementen, in das der Baustein Kurvenscheibentabellen, sowie Metadaten zu den Kurvenscheiben ablegt, die als Segment im Bewegungsprofil eingesetzt werden
VAR_INPUT	Execute	BOOL	Freigabe des Funktionsbausteins (einmalig, flankengesteuert)
	ModuloValue	REAL	Modulowert der Querschneiderachse in [°]
	FormatRatio	REAL	Formatverhältnis des zu schneidenden Produktes (einheitenlos)
	CutAngle	REAL	Länge des Schnittbereichs in [°]
	PendulumFactor	REAL	Maximaler Pendelweg in [%] (0%: kein Pendeln, 100%: maximales Pendeln (Pendeln bis an die Grenze des Schnittbereichs))

RMB_TechCrossCutCrossSeal.library

E/A-Typ	Name	Typ	Kommentar
	CorrectionProfile	MB_CORR_PROFILE	Auswahl eines Spezialprofils. Zur Parametrierung des Spezialprofils müssen die Eingänge "Overspeed" oder "PushOut" benutzt werden. Für die Auswahlmöglichkeiten des Aufzählungstyps MB_CORR_PROFILE siehe den, diesem Typ gewidmeten Abschnitt
	Overspeed	REAL	Konstante Geschwindigkeitsüberhöhung oder -reduzierung im Schnittbereich in [%] der Leitchsgeschwindigkeit (0%: kein Overspeed 100%: doppelte Leitchsgeschwindigkeit -50%: halbe Leitchsgeschwindigkeit) In "CorrectionProfile" muss die Option CORR_PROFILE_OVERSPEED angewählt sein. Ansonsten ist "Overspeed" deaktiviert und somit unwirksam
	PushOut	MB_PUSH_OUT_CONFIG	Struktur zur Definition einer PushOut Bewegung während des Schnittes. In "CorrectionProfile" muss die Option CORR_PROFILE_PUSH_OUT oder CORR_PROFILE_COS_COMP_AND_PUSH_OUT angewählt sein. Ansonsten ist "PushOut" deaktiviert und somit unwirksam. Für eine genauere Beschreibung der Struktur MB_PUSH_OUT_CONFIG siehe den dieser Struktur gewidmeten Abschnitt
	Resolution	MB_RESOLUTION	Angabe der Auflösung der Kurvenscheibenberechnung über die Festlegung der Anzahl an Stützstellen in 3 Stufen. Für die Auswahlmöglichkeiten des Aufzählungstyps MB_RESOLUTION siehe den diesem Typ gewidmeten Abschnitt
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang detaillierte Informationen über den Fehler
	NumberOfSteps	USINT	Anzahl der erzeugten und herunterzuladenden Bewegungsprofilsegmente
	NumberOfCams	USINT	Anzahl der erzeugten und herunterzuladenden Kurvenscheiben

Abb. 10-15: Schnittstellenvariablen Funktionsbaustein MB_CrossCutterCalc-Type04

Min-/Max- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
ModuloValue	REAL	>0.0°	n.def.	360.0°	bei steigender Flanke an "Execute"
FormatRatio	REAL	0.2	30.0	1.0	bei steigender Flanke an "Execute"
CutAngle	REAL	0.0°	< "ModuloValue"	12.0°	bei steigender Flanke an "Execute"
PendelumFactor	REAL	0.0%	100.0%	0.0%	bei steigender Flanke an "Execute"
CorrectionProfile	MB_CORR_PROFILE			CORR_PROFILE_NONE	bei steigender Flanke an "Execute"
Overspeed	REAL	-50.0%	100.0%	0.0%	bei steigender Flanke an "Execute"
PushOut	MB_PUSH_OUT_CONFIG				bei steigender Flanke an "Execute"
Resolution	MB_RESOLUTION			RESOLUTION_HIGH	bei steigender Flanke an "Execute"

Abb. 10-16: Min-/Max- und Default-Werte Funktionsbaustein MB_CrossCutterCalcType04

Signal-Zeit-Diagramm

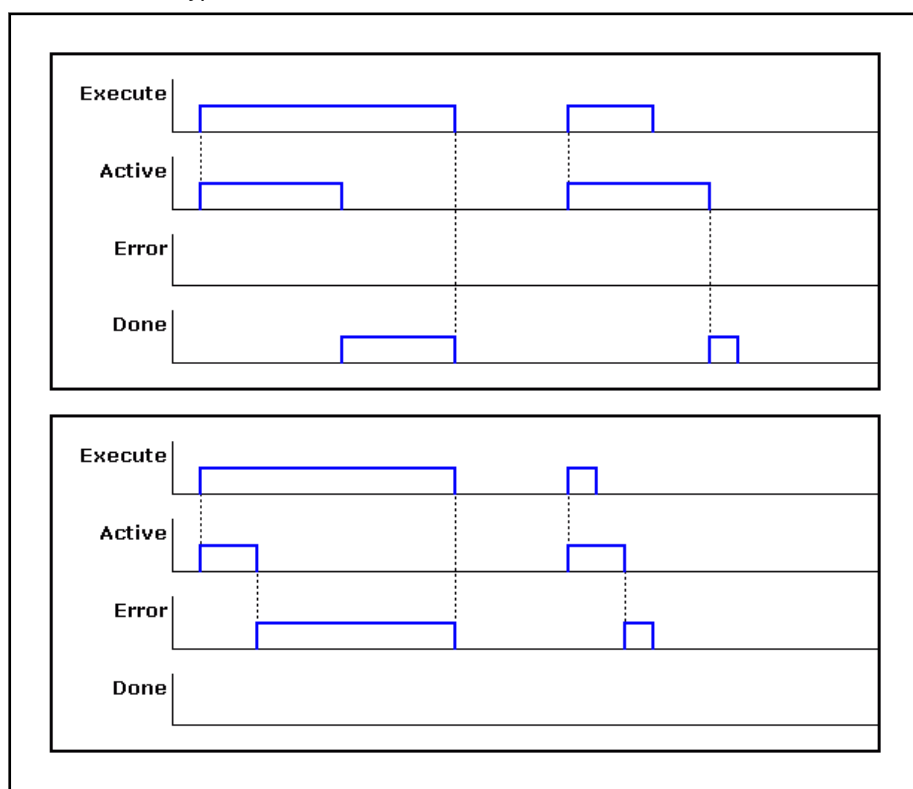


Abb. 10-17: Signal-Zeit-Diagramm Funktionsbaustein MB_CrossCutterCalcType04

Funktionsbeschreibung

Der Funktionsbaustein berechnet auf Basis der Betriebsart "Elektronisches Bewegungsprofil" Profile für die Applikation "Querschneider". Die einzelnen Segmente bestehen dabei entweder aus analytischen Segmenten oder aus Kurvenscheibentabellen. Nach der Bearbeitungsfreigabe mittels "Execute"

RMB_TechCrossCutCrossSeal.library

wird auf Basis der Eingangsgrößen einmalig ein "Elektronisches Bewegungsprofil" aus 3 Segmenten berechnet. Die Kennwerte, die das Profil beschreiben, werden in die jeweiligen Arrays abgelegt, die durch Pointer an den Eingängen des Bausteins "MB_CrossCutterCalcType04" angelegt werden müssen. Die Arrays müssen groß genug sein, um die Anzahl an berechneten Bewegungsschritten (zur Zeit drei) aufnehmen zu können. Das Handling der Pointer ist ähnlich der FBs "MB_ChangeProfileStep"/"MB_ChangeProfileSet", da diese im Anschluss an die Berechnung vom Anwender aufgerufen werden müssen, um das berechnete Profil auf den jeweils freien Satz des "Elektronischen Bewegungsprofils" zu schreiben.

Die folgende Auflistung beschreibt die Bedeutung der einzelnen Eingänge des Funktionsbausteins "MB_CrossCutterCalcType04" genauer:

- Der Eingang "ModuloValue" gibt den Modulowert der Querschneiderachse an. Standardmäßig sind hier 360.0° vorgesehen
- Der Eingang "FormatRatio" kennzeichnet die zu schneidende Produktlänge. Er beschreibt das Verhältnis aus gewünschter Produktlänge und Umfang des Querschneidezylinders. D.h., bei "FormatRatio==1" wird das geschnittene Produkt die selbe Länge haben, wie der Umfang des Querschneidezylinders (Synchronformat). Bei "FormatRatio==5" hingegen wird das geschnittene Produkt die fünffache Länge des Umfangs des Querschneidezylinders haben
- Der Eingang "CutAngle" kennzeichnet den Schnittbereich des Querschneiders. Im Schnittbereich wird entweder mit Materialvorschubgeschwindigkeit oder mit einer (auswählbaren) Korrekturbewegung gedreht
- Der Eingang "PendulumFactor" kennzeichnet das maximale Überspringen der Querschneiderachse in Prozent. D.h., bei "PendulumFactor==0%" wird die Querschneiderachse niemals mit negativer Geschwindigkeit (also rückwärts) drehen, bei "PendulumFactor==100%" wird die Querschneiderachse bis an den Rand des Schnittbereiches drehen, um die Energieverluste und die nötigen Beschleunigungen zu minimieren. Dies ist insbesondere bei größeren Werten für "FormatRatio" vorteilhaft. Bei der Einstellung "PendulumFactor==100%" wird jedoch nicht grundsätzlich bis an den Rand des Schnittbereiches gedreht, da dies nicht immer vorteilhaft ist. Die jeweils optimale Ausnutzung des verfügbaren Weges wird vom FB stets automatisch berechnet
- Der Eingang "CorrectionProfile" ermöglicht verschiedene Korrekturbewegungen im Schnittbereich, die abhängig vom jeweiligen Anwendungsfall benötigt werden. Auswählbar sind hierbei:
 - "CorrectionProfile==CORR_PROFILE_NONE"
(keine Korrektur: Querschneiderachse dreht mit Geschwindigkeit der Master-Achse)
 - "CorrectionProfile==CORR_PROFILE_OVERSPEED"
(konstante Überhöhung der Geschwindigkeit der Querschneiderachse im Schnittbereich)
 - "CorrectionProfile==CORR_PROFILE_COS_COMP"
(konstanter Geschwindigkeitsvektor in Materialrichtung, z.B. für Guillotineschneider)
 - "CorrectionProfile==CORR_PROFILE_PUSH_OUT"
(PushOut des Materials), sowie

- CorrectionProfile==CORR_PROFILE_COS_COMP_AND_PUSH_OUT
(konstanter Geschwindigkeitsvektor in Materialrichtung sowie gleichzeitigem PushOut, z.B. für Guillotineschneider)
- Der Eingang "Overspeed" ermöglicht die exakte Einstellung der Geschwindigkeitsüberhöhung in [%] im Schnittbereich bei
"CorrectionProfile==CORR_PROFILE_OVERSPEED"
Die Eingabe von 10 [%] erhöht somit die Geschwindigkeit der Querschneiderachse im Schnittbereich um 10% gegenüber der Materialgeschwindigkeit
- Der Eingang "PushOut" parametrisiert die PushOut-Korrektur
Der Eingang wird nur verarbeitet bei
"CorrectionProfile==CORR_PROFILE_PUSH_OUT"
oder
"CorrectionProfile==CORR_PROFILE_COS_COMP_AND_PUSH_OUT"
- Der Eingang "Resolution" erlaubt die Auswahl der Auflösung der verwendeten Kurvenscheibentabellen in 3 Stufen. Durch Wahl einer geringeren Auflösung kann Rechenzeit eingespart werden, allerdings auf Kosten der Genauigkeit. Entsprechend kann mit einer hohen Auflösung auf Kosten der Rechenzeit eine höhere Genauigkeit erzielt werden

Die erfolgreiche Berechnung eines Profils wird durch den Ausgang "Done" signalisiert. Die Kennwerte werden in die durch die angelegten Pointer adressierten Datenfelder geschrieben. Diese müssen im Anschluss noch mittels "MB_ChangeProfileSet"/"MB_ChangeProfileStep" auf ein freies Bewegungsprofil geschrieben werden. Ebenso ist mit den Kurvenscheiben zu verfahren. Dabei ist darauf zu achten, dass in der Struktur "MB_CAM_TABLE_DATA" festgelegt ist, in welche Tabelle die Kurvenscheibe geladen werden muss. Anschließend muss das geladene Bewegungsprofil mittels "MB_MotionProfile" aktiviert werden. Die exakte Schnittlänge wird mittels des elektronischen Getriebes am "MB_MotionProfile" eingestellt. Die Getriebeübersetzung entspricht dabei dem Wert von "FormatRatio". Das heißt, für jede mögliche Produktlänge wird ein neues Profil berechnet, das exakt die Eingangsparameter (insbesondere "CutAngle" und "PendelumFactor" berücksichtigt. Die nachfolgende Abbildung zeigt ein Sequenzdiagramm, das die Verwendung des Bausteins MB_CrossCutterCalcType04 beschreibt.

RMB_TechCrossCutCrossSeal.library

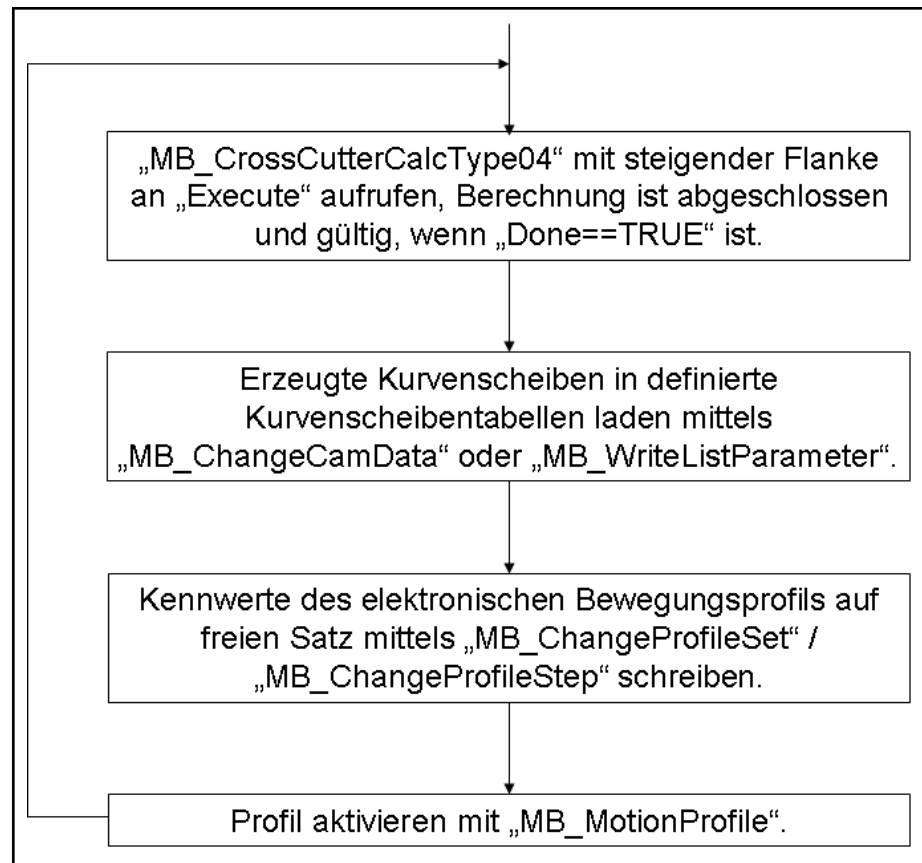


Abb. 10-18: Sequenzdiagramm zur Verwendung von MB_CrossCutterCalcType04

Zur Verwendung mehrerer gleichmäßig (äquidistant) verteilter Messer muss ein mechanisches Getriebe in IndraWorks mit Bezug zur Lastseite projiziert werden. Die Anzahl der Eingangsumdrehungen wird dabei auf 1, die Anzahl der Ausgangsumdrehungen auf die Anzahl der verwendeten Messer n eingestellt. Da jedoch kein reales mechanisches Getriebe vorhanden ist, wird der Antrieb nur jeweils eine n -tel Umdrehung zurücklegen. Somit ist die Geschwindigkeit der Folgeachse in einer Umdrehung n -mal identisch mit der Geschwindigkeit der Leitachse.

Beispiel:

Konfiguration des mechanischen Getriebes bei Verwendung eines Querschneiders mit vier äquidistant auf der Querschneiderachse verteilten Messern

Eingangsgrößen:

Mechanisches Getriebe am Motor, Eingangsumdrehungen $n_{\text{Eingang}} = 10$

Mechanisches Getriebe am Motor, Ausgangsumdrehungen $n_{\text{Ausgang}} = 1$

Messeranzahl pro Kreisumfang $n_{\text{Messer}} = 4$

Berechnung der notwendigen Getriebeeinstellung:

$$\text{mechanisches Getriebe} = \frac{\text{Eingangsumdrehungen}}{\text{Ausgangsumdrehungen}} =$$

$$= \frac{n_{\text{Eingang}}}{n_{\text{Ausgang}} \cdot n_{\text{Messer}}} = \frac{10}{1 \cdot 4} = \frac{10}{4}$$

- S-0-0121 = 10
- S-0-0122 = 4

Bewegt sich bei dieser Parametrierung die Schneidwalze um 90° mechanisch, so entspricht dies 360° elektrisch (= Modulwert der Slave-Achse).

Um die Geschwindigkeit von Leit- und Folgeachse im Schnittbereich auf den gleichen Wert zu bringen, muss noch die Vorschubkonstante zwischen Messrad und interner virtueller Leitachse angepasst werden. Dabei wird der Umfang der internen virtuellen Leitachse mit der Länge zwischen zwei Messern der realen Folgeachse angenommen. Mit dieser Parametrierung ist es möglich, bei entsprechender Vorlaufzeit, jedes Produkt mit anderer Länge zu schneiden.

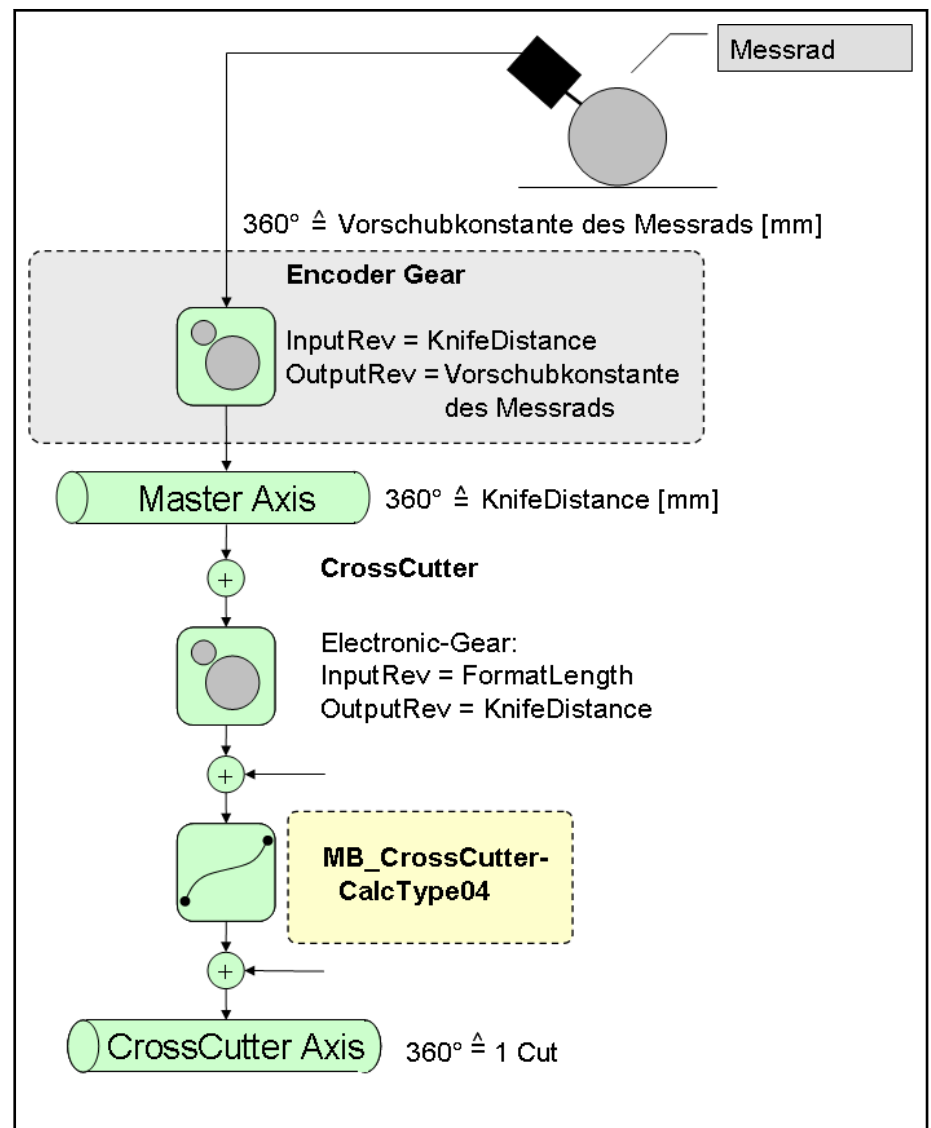


Abb. 10-19: Leitachstruktur bei Betrieb eines Querschneiders

RMB_TechCrossCutCrossSeal.library

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#061x. Er kann in Additional1 und Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0001	Ungültiger Wert des Eingangs "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0002	Ungültiger Wert des Eingangs "FormatRatio"
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0003	Ungültiger Wert des Eingangs "CutAngle"
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0004	Ungültiger Wert des Eingangs "CutAngle" in Kombination mit Cosinus Kompensation
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0005	Ungültiger Wert des Eingangs "PendulumFactor"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0006	Ungültiger Wert des Eingangs "CorrectionProfile"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0007	Ungültiger Wert des Eingangs "Overspeed"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0008	Ungültiger Startpunkt für die PushOut Funktion
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0009	Ungültiger Wert für "Overspeed_At_Cut" in der Struktur "MB_PUSH_OUT_CONFIG"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000A	Ungültiger Wert für "Overspeed_Max" in der Struktur "MB_PUSH_OUT_CONFIG"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000B	Ungültige Kombination von "Overspeed_At_Cut" und "Overspeed_Max"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000C	Zu kleiner Wert des Eingangs "CutAngle" in Kombination mit PushOut
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000D	Ungültiger Wert des Eingangs "Resolution"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000E	Ungültige Werte der "POINTER"-Eingänge
STATE_MACHINE_ERROR (16#0005)	16#0006	xxyy	Ungültiger Zustand der Zustandsmaschine xx gibt den ungültigen Schritt an yy gibt den ungültigen Zustand an

Abb. 10-20: Fehlercodes Funktionsbaustein MB_CrossCutterCalcType04

10.2.2 Besonderheiten der IndraMotion MLC

Inbetriebnahmehinweise Zum Betrieb des Technologie-Funktionsbausteins MB_CrossCutterCalcType04, dem Herunterladen der erzeugten Bewegungsprofile und Kurven-

scheiben, sowie zum Verknüpfen der Querschneiderachse mit der Leitachse sind für das System IndraMotion MLC folgende Inbetriebnahmehinweise zu beachten:

- Für die Querschneiderachse muss Modulowichtung mit dem Modulowert 360° aktiv sein
- Der Funktionsbaustein MB_MotionProfile muss stets mit der Einstellung "0: absolute Synchronisation, ohne erneutes Aufsynchronisieren" für den Eingang "StartMode" betrieben werden, um das exakte Auswechseln der Profile zu gewährleisten
- Der Funktionsbaustein MB_MotionProfile lässt für das elektronische Getriebe nur ganzzahlige Werte von 1 bis 65535 für Zähler und Nenner der Übersetzung zu. Somit ergibt sich, bedingt durch die Quantisierung, eine maximale Schnittlängenabweichung. Diese lässt sich durch Multiplikation mit einer großen Konstanten minimieren, um den Wertebereich möglichst voll ausschöpfen zu können
- Das Herunterladen des erzeugten Bewegungsprofils kann auf der IndraMotion MLC mit dem Funktionsbaustein MB_ChangeProfileSet, das Herunterladen der erzeugten Kurvenscheiben mit dem Funktionsbausteinen MB_ChangeCamData oder MB_WriteListParameter erfolgen. Im Unterschied zum MB_WriteListParameter ist MB_ChangeCamData schneller, die Kurvenscheibe wird jedoch nicht persistent gespeichert und muss nach einem Neustart der Steuerung erneut geladen werden

Beim Herunterladen der Kurvenscheiben ist darauf zu achten, dass die entsprechende Kurvenscheibe an die ihr zugeordnete Position (siehe Element "CamTableID" in Tabelle "Schnittstellenvariablen Funktionsbaustein MB_CAM_TABLE_DATA" in [Kap. 10.2.3 "MB_CAM_TABLE_DATA" auf Seite 539](#)) geladen wird

- Die Auswahl eines kleinen Formatverhältnisses (< 1) sollte mit Bedacht gewählt werden, da die Querschneiderachse im Ausgleichsbereich sehr hohe Geschwindigkeiten (bis zum 8-fachen der Leitachsgeschwindigkeit) erreicht
- Im Parameter A-0-2930 der Querschneiderachse muss Bit 1 gesetzt werden, damit das Getriebe analog zum Bewegungsprofil am Umschaltwinkel von 0° ausgetauscht wird und nicht beim Aufruf des Funktionsbausteins MB_MotionProfile

Formatverhältnis	Verhältnis von Folgeachsgeschwindigkeit zur Leitachsgeschwindigkeit
1	1
0,8	1,5
0,6	2,2
0,4	3,7
0,2	8

Abb. 10-21: Auswirkung des Formatverhältnisses auf die maximale Geschwindigkeit der Querschneiderachse im Ausgleichsbereich

10.2.3 MB_CAM_TABLE_DATA

Kurzbeschreibung

Die Struktur MB_CAM_TABLE_DATA wird vom Baustein MB_CrossCutterCalcType04 verwendet. Sie enthält ein Array mit einer Kurvenscheibentabelle, sowie Meta-Daten, die die Anzahl der Stützpunkte des Arrays, sowie die

RMB_TechCrossCutCrossSeal.library

Position, an die die Tabelle in die Steuerung/den Antrieb geladen werden muss, beschreiben.

Schnittstellenbeschreibung In der folgenden Tabelle sind die Schnittstellenvariablen des Funktionsbausteins aufgelistet.

Strukturelement	Typ	Beschreibung
CamTableID	DINT	Nummer der Kurvenscheibentabelle in die die beschriebene Kurvenscheibe geladen werden muss
NumberOfElements	UINT	Anzahl an Stützstellen der beschriebenen Kurvenscheibe
CamTable	ARRAY[0..1023] OF DINT	Array mit den Stützpunkten der Kurvenscheibe

Abb. 10-22: Schnittstellenvariablen Struktur MB_CAM_TABLE_DATA

Min-/Max- und Default-Werte der Eingänge In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
CamTableID	DINT	0		0
NumberOfElements	UINT	0	1024	0

Abb. 10-23: Min-/Max- und Default-Werte der Struktur MB_CAM_TABLE_DATA

Funktionsbeschreibung Im Strukturelement "CamTableID" ist die Nummer der Kurvenscheibentabelle in dem Format abgelegt, wie es die Funktionsbausteine MB_ChangeCamData, MB_ChangeProfileStep und MB_ChangeProfileSet benötigen. [Abb. 10-24 "Zuordnung von CamTableID zu Parameternummern"](#) auf Seite 540 zeigt die Mappings zwischen "CamTableID" und der entsprechenden Parameternummer im System MLC und MLD.

CamTableID	Parameternummer IndraMotion MLC	Parameternummer IndraMotion MLD
1	C-0-2001	P-0-0072
2	C-0-2002	P-0-0092
3	C-0-2003	P-0-0780
4	C-0-2004	P-0-0781
5	C-0-2005	P-0-0784
6	C-0-2006	P-0-0785
7-99	C-0-2007 - C-0-2099	nicht verfügbar

Abb. 10-24: Zuordnung von CamTableID zu Parameternummern

Die folgende Grafik zeigt, wie die Daten der Struktur MB_CAM_TABLE_DATA verwendet werden müssen, um die Kurvenscheibe in die Steuerung bzw. den Antrieb zu laden.

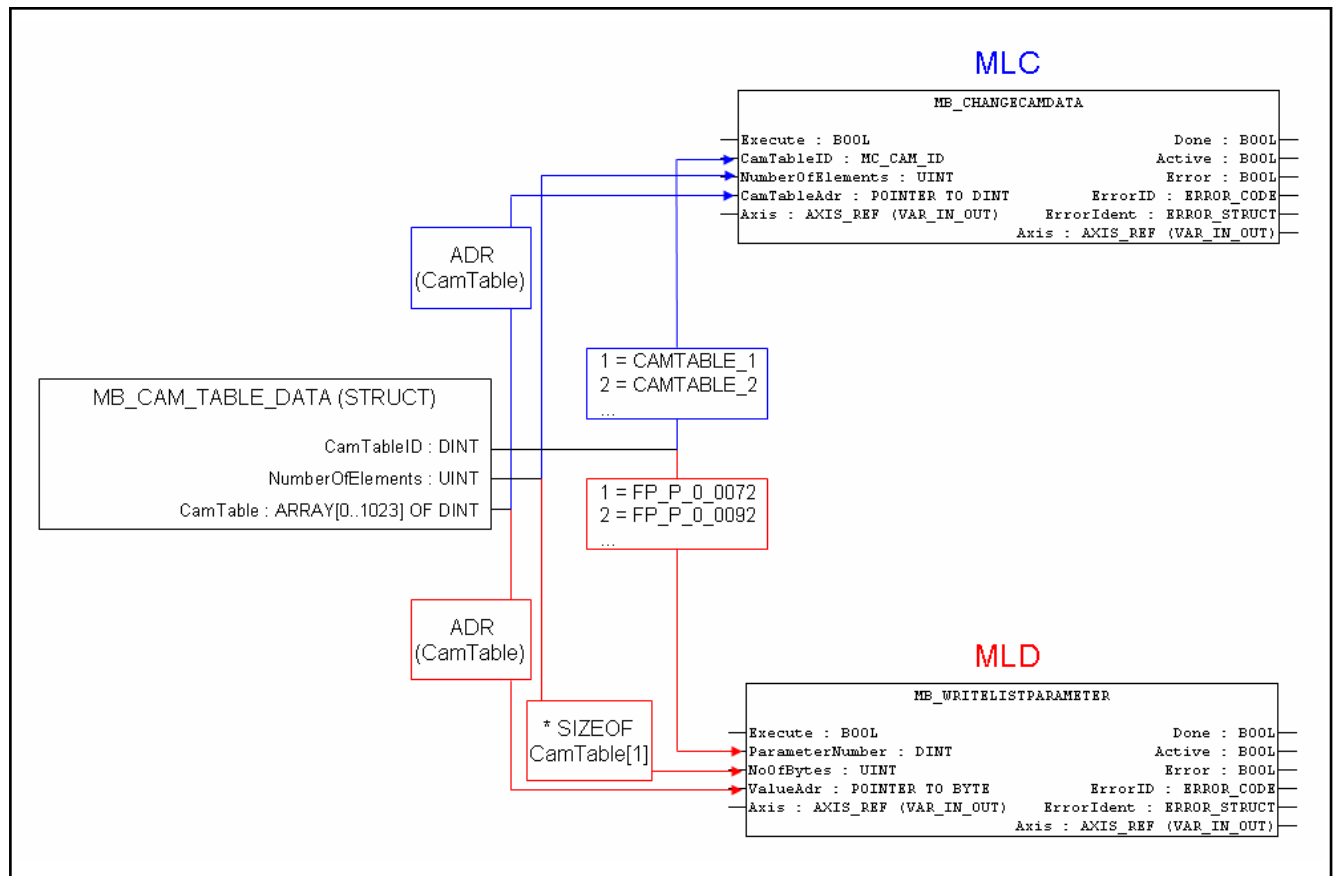


Abb. 10-25: Verschaltungsplan Struktur MB_CAM_TABLE_DATA

10.2.4 MB_CORR_PROFILE

Kurzbeschreibung Die Enumeration MB_CORR_PROFILE wird vom Funktionsbaustein MB_CrossCutterCalcType04 verwendet um eine Korrekturbewegung im Schnittbereich zu definieren. In der folgenden Beschreibung sind die möglichen Elemente des Aufzählungstyps MB_CORR_PROFILE und deren Bedeutung beschrieben.

Elemente des Aufzählungstyps In der folgenden Tabelle sind die Elemente des Aufzählungstyps aufgelistet.

Element	Wert	Beschreibung
CORR_PROFILE_NONE	0	Keine Korrektur im Schnittbereich. Geschwindigkeit der Querschneiderachse entspricht der Geschwindigkeit der Masterachse
CORR_PROFILE_OVERSPEED	1	Konstante Überhöhung oder Reduzierung der Geschwindigkeit der Querschneiderachse im Schnittbereich. Die Geschwindigkeit der Querschneiderachse ist dabei um den am Eingang "Overspeed" des Funktionsbausteins MB_CrossCutterCalcType04 parametrisierten Wert schneller oder langsamer als die Geschwindigkeit der Masterachse

RMB_TechCrossCutCrossSeal.library

Element	Wert	Beschreibung
CORR_PROFILE_COS_COMP	2	Korrektur der Geschwindigkeit der Querschneiderachse. Die Änderung erfolgt so, dass die Querschneiderachse in Materialrichtung (also die x-Komponente des Geschwindigkeitsvektors) mit konstanter Geschwindigkeit dreht. Dabei muss die Drehzahl der Querschneiderachse in Abhängigkeit der aktuellen Lage im Schnitt angepasst werden
CORR_PROFILE_PUSH_OUT	3	Durchführen einer PushOut-Bewegung am Ende des Schnittes. Parametrierung über den Eingang "PushOut" des Funktionsbausteins MB_CrossCutterCalcType04. Dabei muss die Drehzahl der Querschneiderachse in Abhängigkeit der aktuellen Lage im Schnitt angepasst werden. Siehe auch Kap. 10.2.6 "MB_PUSH_OUT_CONFIG" auf Seite 543
CORR_PROFILE_COS_COMP_AND_PUSH_OUT	4	Kombination aus den Korrekturen "CORR_PROFILE_COS_COMP" und "CORR_PROFILE_PUSH_OUT". Parametrierung über den Eingang "PushOut" des Funktionsbausteins MB_CrossCutterCalcType04. Siehe auch Kap. 10.2.6 "MB_PUSH_OUT_CONFIG" auf Seite 543

Abb. 10-26: Elemente des Aufzählungstyps MB_CORR_PROFILE

10.2.5 MB_RESOLUTION

Kurzbeschreibung

Die Enumeration MB_RESOLUTION wird vom Baustein MB_CrossCutterCalcType04 verwendet. Mit ihr kann die Anzahl der Stützpunkte der Kurvenscheibentabellen und somit die benötigte Rechenzeit variiert werden. In der folgenden Beschreibung sind die Elemente der Enumeration MB_RESOLUTION und deren Bedeutung beschrieben.

Elemente des Aufzählungstyps

In der folgenden Tabelle sind die Elemente des Aufzählungstyps aufgelistet.

Element	Wert	Beschreibung	Anzahl Stützstellen (Schnittsegment + Ausgleichssegment + Schnittsegment)
RESOLUTION_LOW	1	Niedrige Auflösung, geringe Rechenzeit nötig (um ca. Faktor 3 schneller als bei mittlerer Auflösung)	10 + 100 + 10
RESOLUTION_MIDDLE	2	Mittlere Auflösung, mittlere Rechenzeit nötig (um ca. Faktor 3 schneller als bei hoher Auflösung)	25 + 300 + 25
RESOLUTION_HIGH	3	Hohe Auflösung, hohe Rechenzeit nötig	50 + 1000 + 50

Abb. 10-27: Elemente des Aufzählungstyps MB_RESOLUTION

Funktionsbeschreibung

Mit der Enumeration MB_RESOLUTION kann die Anzahl der Stützpunkte der Kurvenscheibentabellen und somit die benötigte Rechenzeit variiert werden.

Für kurze Formate und wenig Vorlaufzeit für einen Formatwechsel kann die Rechenzeit des Bausteins durch Auswahl der Option "RESOLUTION_LOW" verkürzt werden. Dabei ist allerdings darauf zu achten, dass die Auflösung der erzeugten Kurvenscheiben sinkt und die Maschine unruhig laufen kann.

Bei langen Formaten und großen Vorlaufzeiten für Formatwechsel kann die Option "RESOLUTION_HIGH" gewählt werden, womit die erzeugten Kurvenscheiben in größtmöglicher Auflösung generiert werden. Grundsätzlich ist diese Einstellung, also "RESOLUTION_HIGH", zu bevorzugen.

Welche Option hier gewählt werden muss, ist stark abhängig von der eingesetzten Applikation und sollte bei der Inbetriebnahme durch Versuche bestimmt werden.

10.2.6 MB_PUSH_OUT_CONFIG

Kurzbeschreibung Die Struktur MB_PUSH_OUT_CONFIG wird vom Baustein MB_CrossCutter-CalcType04 verwendet. Durch sie kann ein definierter PushOut am Ende des Schnittes parametrisiert werden. In der folgenden Schnittstellenbeschreibung sind die Elemente der Struktur MB_PUSH_OUT_CONFIG sowie deren Minimal-, Maximal- und Defaultwerte beschrieben.

Schnittstellenbeschreibung In der folgenden Tabelle sind die Schnittstellenvariablen des Funktionsbausteins aufgelistet.

Strukturelement	Typ	Beschreibung
PushOutBegin	REAL	Beginn des PushOut-Bereichs (Parameter A). Wert muss ungleich 0 sein, Einheit in [°]
Overspeed_At_Cut	REAL	Überhöhung der Geschwindigkeit während des Schnittes (Parameter B). Wert muss ungleich 0 sein, Einheit in [%]
Overspeed_Max	REAL	Maximale Übergeschwindigkeit (Parameter C), Einheit in [%]

Abb. 10-28: Schnittstellenvariablen Struktur MB_PUSH_OUT_CONFIG

Min-/Max- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
PushOutBegin	REAL	$-\frac{\text{"CutAngle"}}{2}$	$\frac{\text{"CutAngle"}}{2}$	-5.0°
Overspeed_At_Cut	REAL	-100.0%	100.0%	5.0%
Overspeed_Max	REAL	>0.0%	100.0%	10.0%

Abb. 10-29: Min-/Max- und Default-Werte der Struktur MB_PUSH_OUT_CONFIG

Funktionsbeschreibung

Die folgenden Grafiken zeigen den Geschwindigkeitsverlauf der Querschneiderwalze im Schnittbereich. Die einzelnen Parameter können durch die Elemente der Struktur MB_PUSH_OUT_CONFIG angepasst werden.

Für die Grafiken ist ein Kurvenprofil mit einem Schnittwinkel von 20° angenommen. Die Steilheit der Geschwindigkeitsänderung wird dabei über die beiden Variablen "Overspeed_At_Cut" und "Overspeed_Max" bestimmt. Der Beginn der Geschwindigkeitserhöhung wird mittels "PushOutBegin" festgelegt. Zur eindeutigen Parametrierung ist es notwendig, dass "PushOutBegin" und "Overspeed_At_Cut" ungleich Null sind. Die beiden Abbildungen zeigen exemplarisch die Parametrierung. Die grüne Linie kennzeichnet die Steilheit des Anstiegs und lässt sich durch Verbinden der Punkte A, B und C bestimmen.

RMB_TechCrossCutCrossSeal.library

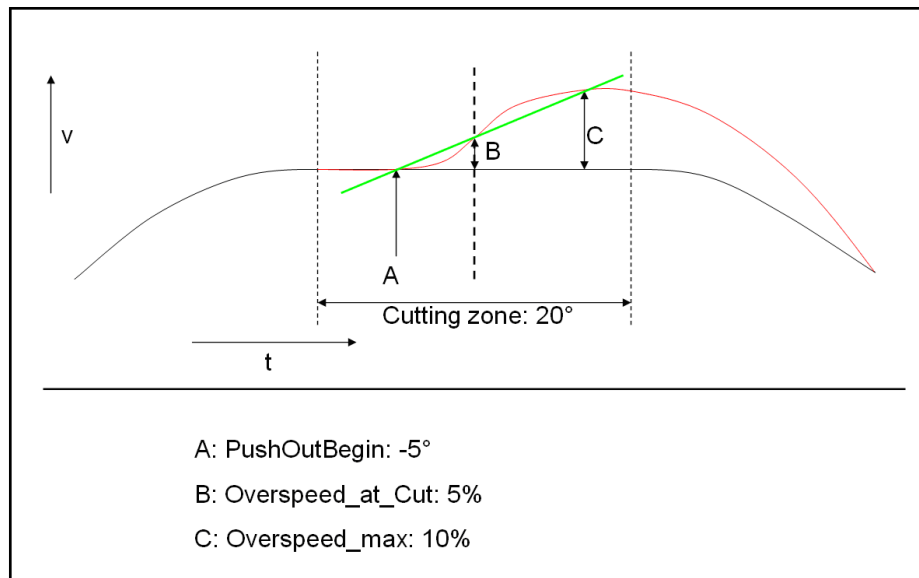


Abb. 10-30: Definition des PushOut.PushOutBegin < 0

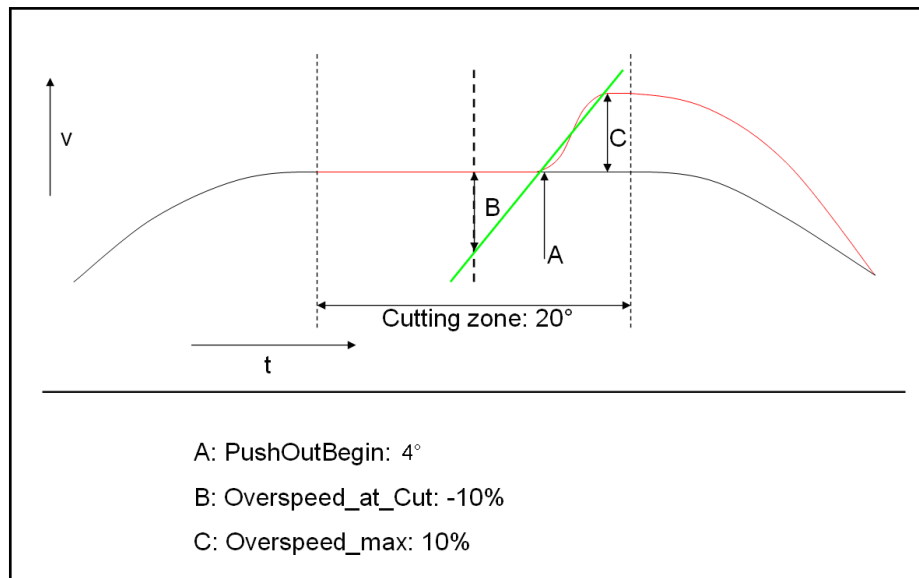


Abb. 10-31: Definition des PushOut.PushOutBegin > 0

11 RMB_TechCam.library

11.1 Einführung und Übersicht

Die hier beschriebene Bibliothek beinhaltet Funktionsbausteine zur Erstellung von Kurvenscheibentabellen.

Die Funktionsbausteine sind für die unterschiedlichen Anwendungen wie Vorschubbewegung, Querschneider etc. verfügbar.

Die Funktionsbausteine [Kap. 11.2 "MB_FeedMovementCalcType01" auf Seite 545](#), [Kap. 11.3 "MB_FeedMovementCalcType02" auf Seite 551](#) und [Kap. 11.4 "MB_CoordinatePairCamType01" auf Seite 558](#) arbeiten mit der Enumeration [Kap. 5.14.3 "MB_CAMPROF" auf Seite 163](#) und der Struktur [Kap. 5.14.4 "MB_CAMTABLE" auf Seite 163](#), welche sich in der Bibliothek "ML_TechBase.compiled-library" befinden. Des Weiteren können die Kurvenscheibentabellen der genannten Funktionsbausteine mittels Funktionsbaustein [Kap. 5.14.2 "MB_LoadCamData" auf Seite 160](#) der Bibliothek "ML_TechBase.compiled-library" in eine Kurvenscheibe in der Steuerung oder in den Antrieb geladen werden.

Übersicht über die Funktionsbausteine

Funktionsbaustein	Beschreibung
MB_FeedMovementCalcType01	Übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen für eine Vorschubbewegung
MB_FeedMovementCalcType02	Übernimmt, unter Benutzung eines Polynom 7. Ordnung, die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunkten, bei der der Ruck-, Beschleunigungs- und Geschwindigkeits-Eingang begrenzt werden kann um eine optimale Ausführung zu gewährleisten
MB_CoordinatePairCamType01	Übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen, die abhängig des ausgewählten Koordinatenpaartyps ermittelt wird

Abb. 11-1: RMB_TechCam.compiled-library

11.2 MB_FeedMovementCalcType01

Kurzbeschreibung

Der Funktionsbaustein berechnet eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen für eine Vorschubbewegung.

Zuordnung Zielsystem/Bibliothek

IndraMotion MLC 12VRS	RMB_TechCam.compiled-library

Abb. 11-2: Referenztable MB_FeedMovementCalcType01

Schnittstellenbeschreibung

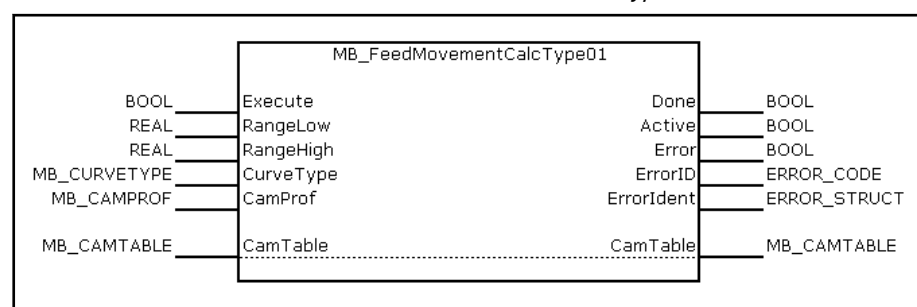


Abb. 11-3: Funktionsbaustein MB_FeedMovementCalcType01

RMB_TechCam.library

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellen-Elementen und Information über die Anzahl der Elemente
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	RangeLow	REAL	Anfangspunkt der Vorschubbewegung; erster Punkt des Bewegungsprofils
	RangeHigh	REAL	Endpunkt der Vorschubbewegung; letzter Punkt des Bewegungsprofils
	CurveType	MB_CURVETYPE	Auswahl des Kurventyps
	CamProf	MB_CAMPROF	Auswahl des Kurvenscheibenprofils
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe "Fehlerbehandlung" auf Seite 551)

Abb. 11-4: Schnittstellenvariablen Funktionsbaustein MB_FeedMovementCalcType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
RangeLow	REAL	0.0	<360.0	0.0	Steigende Flanke an Execute
RangeHigh	REAL	>0.0	360.0	360.0	Steigende Flanke an Execute
CurveType	MB_CURVETYPE			CURVE-TYPE_POLY_5	Steigende Flanke an Execute
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Steigende Flanke an Execute

Abb. 11-5: Min-/Max- und Default-Werte Funktionsbaustein MB_FeedMovementCalcType01

Die Enumeration "MB_CURVETYPE" wird verwendet um zwischen verschiedenen Ausgleichsbewegungen von Rast in Rast zu wählen.

Element	Wert	Beschreibung
CURVETYPE_QUADRATIC_PARABLE	0	Quadratische Parabel
CURVETYPE_SIMPLE_SINUSLINE	1	Einfache Sinuslinie

Element	Wert	Beschreibung
CURVETYPE_POLY_5	2	Polynom 5. Grades
CURVETYPE_SLOPING_SINUSLINE	3	Geneigte Sinuslinie
CURVETYPE_MODIFIED_ACCELERATIONTRAPEZE	4	Modifiziertes Beschleunigungstrapez
CURVETYPE_MODIFIED_SINUSLINE	5	Modifizierte Sinuslinie

Abb. 11-6: Elemente des Aufzählungstyps MB_CURVETYPE

Die Enumeration [Kap. 5.14.3 "MB_CAMPROF" auf Seite 163](#) wird verwendet um zwischen verschiedenen Kurvenscheibenprofilen zu wählen.



Bei der Auswahl des Kurvenscheibenprofils "CAMPROF_STANDARD" muss der Linearpfad deaktiviert sein, da es sich um ein Profil ohne Untersetzung handelt. Während bei den Profilen "CAMPROF_ADDITIVE" und "CAMPROF_ADDITIVE_SEGMENTABLE" der Linearpfad über ein Häkchen aktiviert werden muss und die Reduktion auf den Wert "1" zu stellen ist, da es sich um Profile mit Untersetzung handelt.

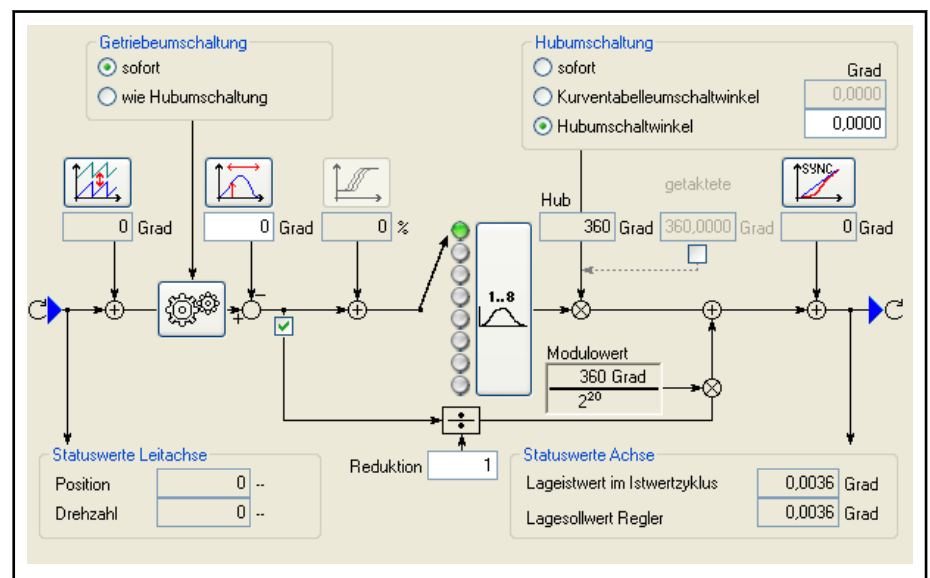


Abb. 11-7: Aktivierung des Linearpfades

Die Struktur [Kap. 5.14.4 "MB_CAMTABLE" auf Seite 163](#) besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Funktionsbeschreibung

Mit der Vorgabe eines durch die Grenzen "RangeLow" und "RangeHigh" definierten Vorschubbereiches berechnet der Funktionsbaustein MB_FeedMovementCalcType01 nach der Bearbeitungsfreigabe mittels "Execute" einmalig auf Basis der Eingangsgrößen eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen. Hierbei wird die Anzahl der Stützpunktelemente einer Kurvenscheibentabelle durch das Element "NumberOfElements" der Struktur "CamTable" eingestellt.

Die Bewegung von Rast in Rast ist mit verschiedenen Ausgleichsbewegungen möglich. Die einstellbaren Bewegungen sind:

- Quadratische Parabel

RMB_TechCam.library

- Einfache Sinuslinie
- Polynom 5. Grades
- Geneigte Sinuslinie
- Modifiziertes Beschleunigungstrapez
- Modifizierte Sinuslinie

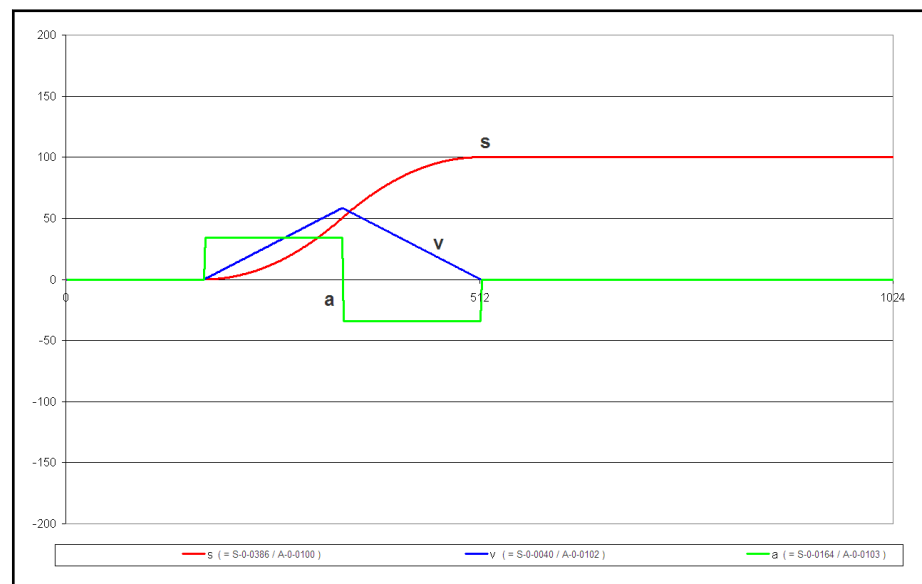
Die zu berechnende Kurvenscheibentabelle wird in der Struktur "CamTable" mit Wichtung in Prozent abgelegt. Die Gültigkeit der Kurvenscheibenelemente wird durch den Ausgang "Done" signalisiert. Die Kurvenscheibentabelle kann mittels des FB [Kap. 5.14.2 "MB_LoadCamData" auf Seite 160](#) in eine Kurvenscheibe in der Steuerung oder in den Antrieb geladen werden.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" werden im Fehlerfall nicht aktualisiert.

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins MB_FeedMovementCalc-Type01 weist folgende Eigenschaften auf:

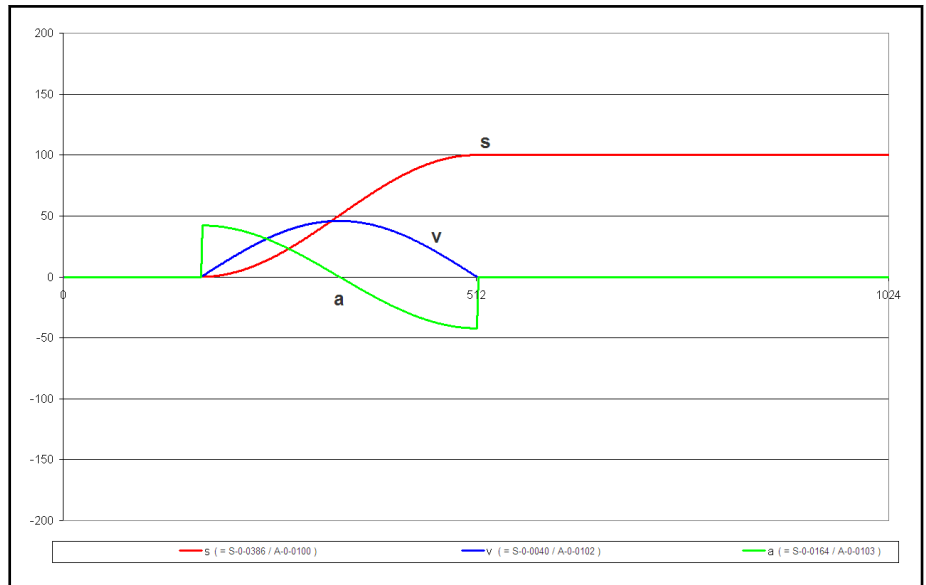
- Die vom Funktionsbaustein erzeugte Kurvenscheibentabelle beschreibt eine Rast-in-Rast-Bewegung
- Der Algorithmus zur Berechnung der Kurvenscheibenstützpunkte ist abhängig von der Differenz (RangeHigh-RangeLow) auf max 1026 Zyklen, d. h. Aufrufe der Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden
- Das Kurvenscheibenprofil ist für die zyklische Abarbeitung im Antrieb als Parameter P-0-0072, P-0-0092 etc. vorgesehen
- CurveType: CURVETYPE_QUADRATIC_PARABLE
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf



RangeLow = 60; RangeHigh = 180

Abb. 11-8: MB_FeedMovementCalcType01 Bewegungsprofil QUADRATIC_PARABLE

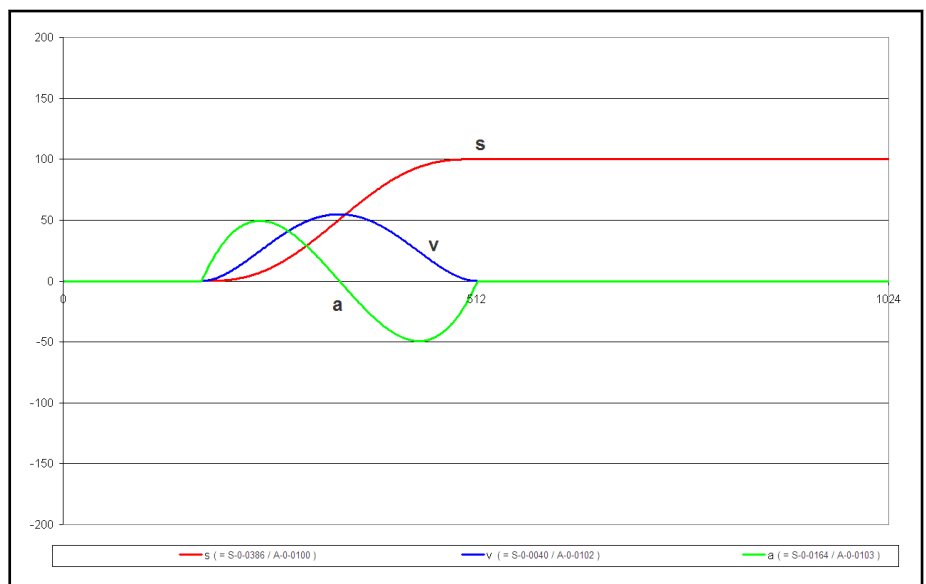
- CurveType: CURVETYPE_SIMPLE_SINUSLINE
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf



RangeLow = 60; RangeHigh = 180

Abb. 11-9: MB_FeedMovementCalcType01 Bewegungsprofil SIMPLE_SINUSLINE

- CurveType: CURVETYPE_POLY_5
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zum Ruck (s, v, a, j)

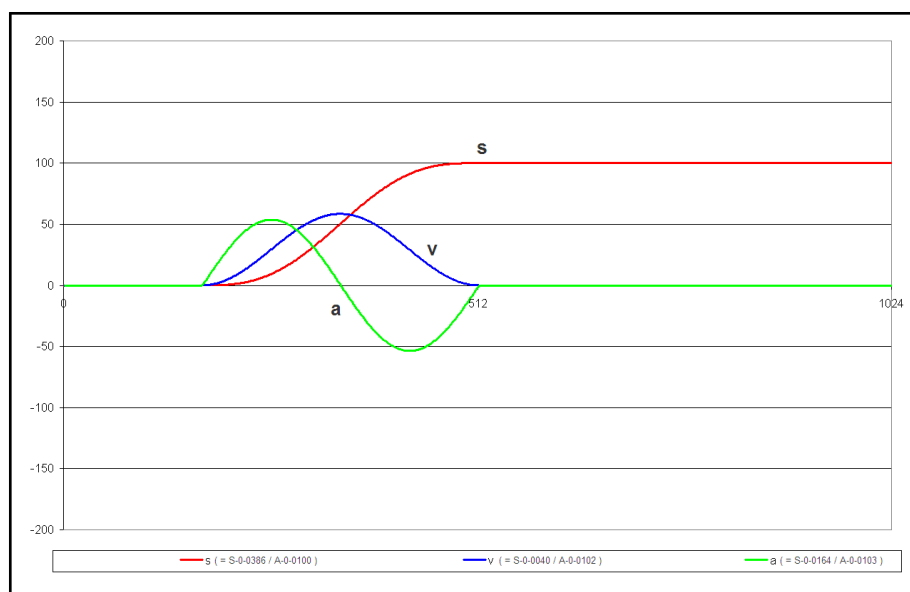


RangeLow = 60; RangeHigh = 180

Abb. 11-10: MB_FeedMovementCalcType01 Bewegungsprofil POLY_5

- CurveType: CURVETYPE_SLOPING_SINUSLINE
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zum Ruck (s, v, a, j)

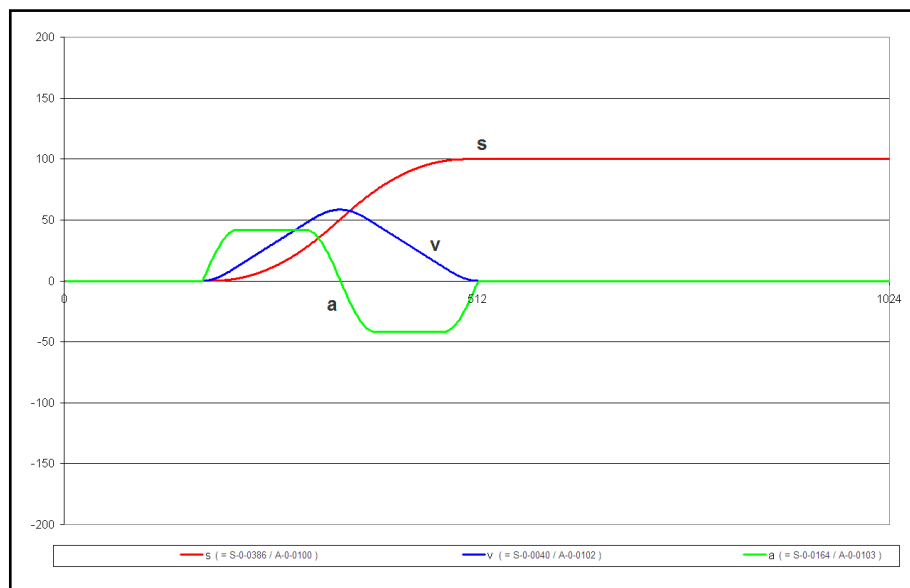
RMB_TechCam.library



RangeLow = 60; RangeHigh = 180

Abb.11-11: MB_FeedMovementCalcType01 Bewegungsprofil SLOPING_SINUSLINE

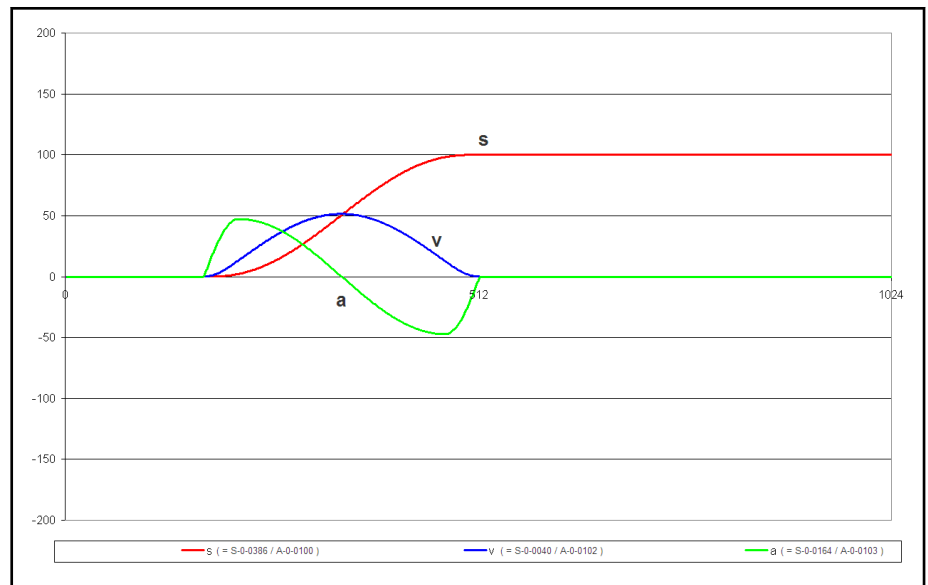
- CurveType: CURVETYPE_MODIFIED_ACCELERATIONTRAPEZE
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zum Ruck (s, v, a, j)



RangeLow = 60; RangeHigh = 180

Abb.11-12: MB_FeedMovementCalcType01 Bewegungsprofil MODIFIED_ACCELERATIONTRAPEZE

- CurveType: CURVETYPE_MODIFIED_SINUSLINE
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zum Ruck (s, v, a, j)



RangeLow = 60; RangeHigh = 180

Abb. 11-13: MB_FeedMovementCalcType01 Bewegungsprofil MODIFIED_SINUSLINE

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170x. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#00000006	16#00000000	Ungültiger Zustand des Funktionsbausteins
CALCULATION_ERROR	16#0000000C	16#00000000	Division durch Null
INPUT_RANGE_ERROR	16#00002200	16#00000001	Eingang NumberOfElements ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR	16#00002220	16#00000001	Eingang RangeLow ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR	16#00002220	16#00000002	Eingang RangeHigh ist kleiner als Minimum oder größer als Maximum
INPUT_INVALID_ERROR	16#00002201	16#00000001	Adresse von Feld CamTable ist null
INPUT_INVALID_ERROR	16#00002221	16#00000001	Eingang RangeLow ist größer oder gleich als RangeHigh

Abb. 11-14: Fehlercodes Funktionsbaustein MB_FeedMovementCalcType01

11.3 MB_FeedMovementCalcType02

Kurzbeschreibung Der Funktionsbaustein berechnet unter Benutzung eines Polynom 7. Ordnung eine Kurvenscheibentabelle mit bis zu 1024 Stützpunkten, bei der der Ruck-, Beschleunigungs- und Geschwindigkeits-Eingang begrenzt werden kann um eine optimale Ausführung zu gewährleisten.

Zuordnung Zielsystem/Bibliothek

IndraMotion MLC 12VRS	RMB_TechCam.compiled-library

Abb. 11-15: Referenztable MB_FeedMovementCalcType02

RMB_TechCam.library

Schnittstellenbeschreibung

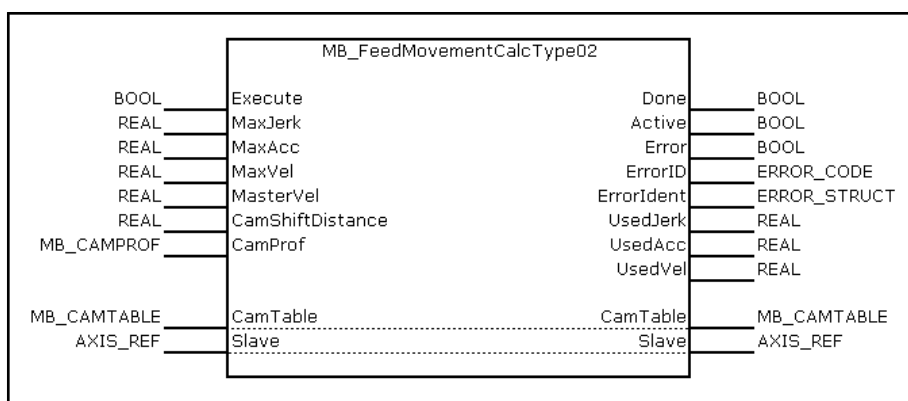


Abb. 11-16: Funktionsbaustein MB_FeedMovementCalcType02

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellen-Elementen und Information über die Anzahl der Elemente
	Slave	AXIS_REF	Referenz zur Folgeachse
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	MaxJerk	REAL	Maximaler Ruck
	MaxAcc	REAL	Maximale Beschleunigung
	MaxVel	REAL	Maximale Geschwindigkeit
	MasterVel	REAL	Geschwindigkeit der Leitachse
	CamShiftDistance	REAL	Länge der Kurvenscheibe
	CamProf	MB_CAMPROF	Auswahl des Kurvenscheibenprofils
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe "Fehlerbehandlung" auf Seite 557)
	UsedJerk	REAL	Verwendeter Ruck in Relation zum Eingang 'MaxJerk' in [%]
	UsedAcc	REAL	Verwendete Beschleunigung in Relation zum Eingang 'MaxAcc' in [%]
	UsedVel	REAL	Verwendete Geschwindigkeit in Relation zum Eingang 'MaxVel' in [%]

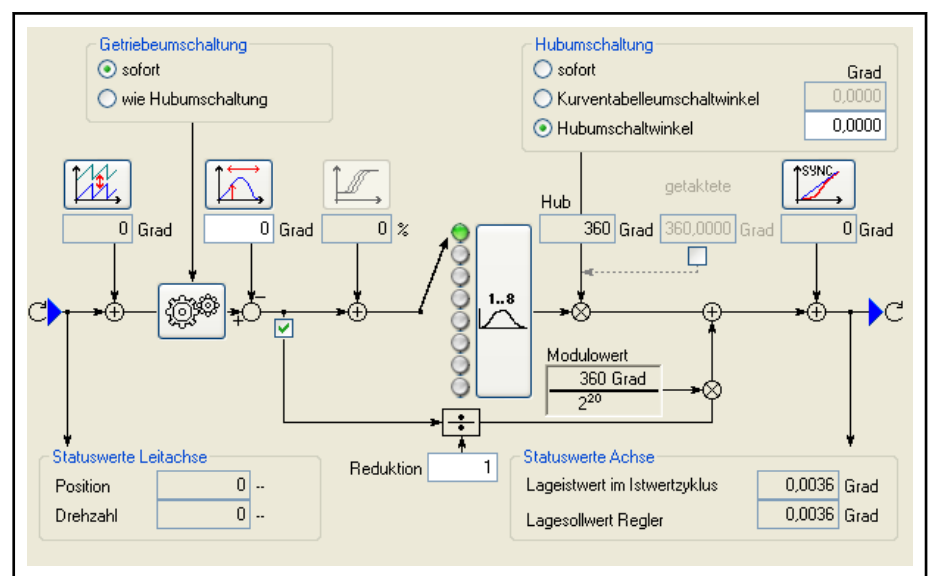
Abb. 11-17: Schnittstellenvariablen Funktionsbaustein MB_FeedMovementCalcType02

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
MaxJerk	REAL	>0.0	n.def.	20.0	Steigende Flanke an Execute
MaxAcc	REAL	>0.0	n.def.	20.0	Steigende Flanke an Execute
MaxVel	REAL	>0.0	n.def.	20.0	Steigende Flanke an Execute
MasterVel	REAL	>0.0	n.def.	10.0	Steigende Flanke an Execute
CamShaftDis- tance	REAL	>0.0	n.def.	360.0	Steigende Flanke an Execute
CamProf	MB_CAMPROF			CAM_PROF_STA NDARD	Steigende Flanke an Execute

Die Enumeration **Kap. 5.14.3 "MB_CAMPFROF"** auf Seite 163 wird verwendet um zwischen verschiedenen Kurvenscheibenprofilen zu wählen.



Bei der Auswahl des Kurvenscheibenprofils "CAMPROF_STANDARD" muss der Linearpfad deaktiviert sein, da es sich um ein Profil ohne Untersetzung handelt. Während bei den Profilen "CAMPROF_ADDITIVE" und "CAMPROF_ADDITIVE_SEGMENTABLE" der Linearpfad über ein Häkchen aktiviert werden muss und die Reduktion auf den Wert "1" zu stellen ist, da es sich um Profile mit Untersetzung handelt.



Die Struktur [Kap. 5.14.4 "MB_CAMTABLE" auf Seite 163](#) besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die

RMB_TechCam.library

Funktionsbeschreibung

Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibepunkte des Arrays "CamTable" ersichtlich ist.

Der Funktionsbaustein MB_FeedMovementCalcType02 berechnet eine Kurvenscheibentabelle mit bis zu 1024 Stützpunkten, bei der Ruck, Beschleunigung und Geschwindigkeit begrenzt werden können. Hierbei wird die Anzahl der Stützpunktelemente einer Kurvenscheibentabelle durch das Element "NumberOfElements" der Struktur "CamTable" eingestellt.

Der Anwender stellt Leitachsgeschwindigkeit, Kurvenscheibenhub, maximale Geschwindigkeit, Beschleunigung und Ruck ein. Aus diesen Vorgabewerten berechnet der Funktionsbaustein dann ein Bewegungsprofil in Form einer Kurvenscheibe, welche eine Rast in Rast Bewegung darstellt. Das Ruckprofil wird entweder aus einzelnen Parabelstücken zusammengesetzt oder als Polynom 7. Ordnung berechnet. Aus diesem Ruckprofil errechnet sich das Bewegungsprofil. Das Bewegungsprofil ermöglicht eine Endlosbewegung der Slave-Achse.

Anmerkung:

Die Variante des Funktionsbausteins für Antriebskurvenscheiben holt weitere für die Berechnung des Bausteins benötigte Informationen, wie maximaler Ruck des Antriebs oder Modulowert, aus den Parametern des betrachteten Antriebs. D.h. während der Baustein aktiv ist, werden Parameter aus dem Antrieb gelesen. Falls die Vorgabewerte möglich sind, wird eine Kurvenscheibe berechnet oder ein Fehler ausgegeben, dass die Bewegung mit diesen Vorgabewerten nicht möglich ist.

Der Modulowert des Antriebs muss dem Wert der Eingangsvariablen CamShaftDistance entsprechen, ansonsten wird der FB mit Fehler beendet. Der Parameter CamShaftDistance des Befehls MC_CamIn muss deshalb genauso groß eingestellt werden, wie der Modulowert der Slaveachse, ansonsten ist keine Endlosbewegung möglich.

Die zu berechnende Kurvenscheibentabelle wird in der Struktur "CamTable" mit Wichtung in Prozent abgelegt. Die Gültigkeit der Kurvenscheibenelemente wird durch den Ausgang "Done" signalisiert. Die Kurvenscheibentabelle kann mittels des FB [Kap. 5.14.2 "MB_LoadCamData" auf Seite 160](#) in eine Kurvenscheibe in der Steuerung oder in den Antrieb geladen werden.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" sowie die Ausgänge "UsedJerk", "UsedAcc" und "UsedVel" werden im Fehlerfall nicht aktualisiert.

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins MB_FeedMovementCalcType02 weist folgende Eigenschaften auf:

- Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist stetig bis zum Ruck (s, v, a, j)
- Der Algorithmus zur Berechnung der Kurvenscheibenstützpunkte ist auf mehrere Zyklen, d. h. Aufrufe der Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden

Übersicht möglicher Berechnungsverfahren

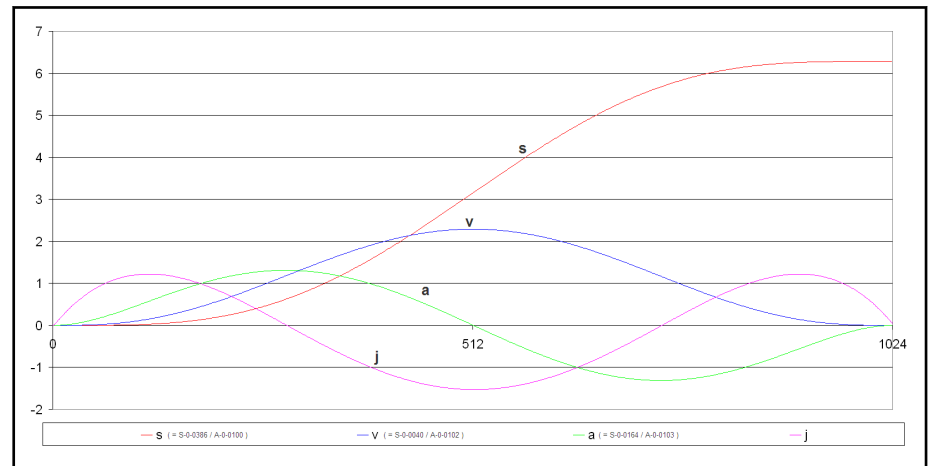
Die Umschaltung zwischen den einzelnen Varianten übernimmt der Funktionsbaustein, wobei bevorzugt eine Kurvenscheibe mit Polynom 7. Ordnung berechnet wird. Falls dies aufgrund der eingestellten Grenzwerte nicht möglich ist, wird auf die Berechnung aus Parabelstücken ausgewichen. Falls

auch diese Möglichkeit die Einhaltung der Grenzwerte nicht erlaubt, wird der Ausgang "Error" des Bausteins gesetzt.

Anmerkung:

Bei der Berechnung aus Parabelstücken kann es bei ungünstigem Verhältnis aus Ruck zu Beschleunigung (hoher Ruck bei geringer Beschleunigung) zu Ungenauigkeiten kommen, da die Parabel für den Ruck sehr hoch, aber gleichzeitig sehr schmal, wird. Aufgrund der begrenzten Genauigkeit der Berechnung kann dies zu Ungenauigkeiten führen.

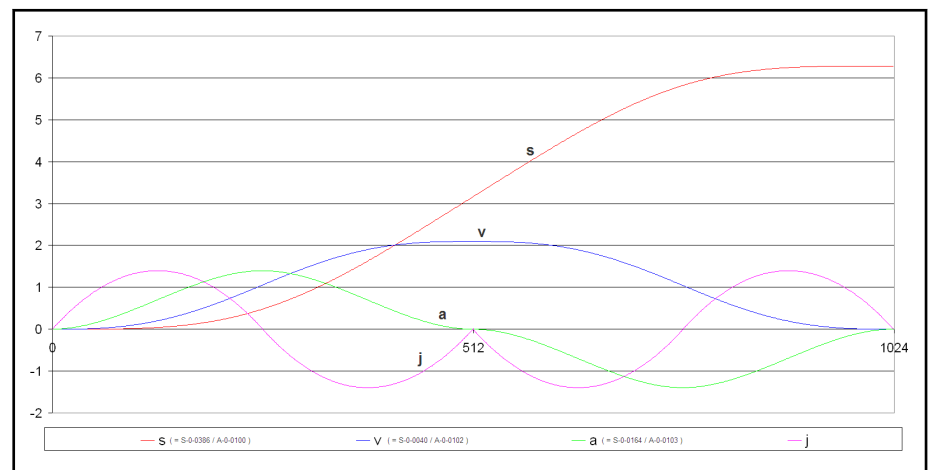
Variante 1: Polynom 7. Ordnung



MaxJerk = 10 rad/s³; MaxAcc = 10 rad/s²; MaxVel = 10 rad/s; MasterVel = 10 U/min; Modulowert = 2*Pi rad
Berechnung als Polynom 7. Ordnung

Abb. 11-20: MB_FeedMovementCalcType02 Kurvenscheibenprofil (Slave = f(Master))

Variante 2: Geschwindigkeitsgrenzwert und Beschleunigungsgrenzwert werden nicht erreicht (Variante kommt in der Praxis nur selten vor)

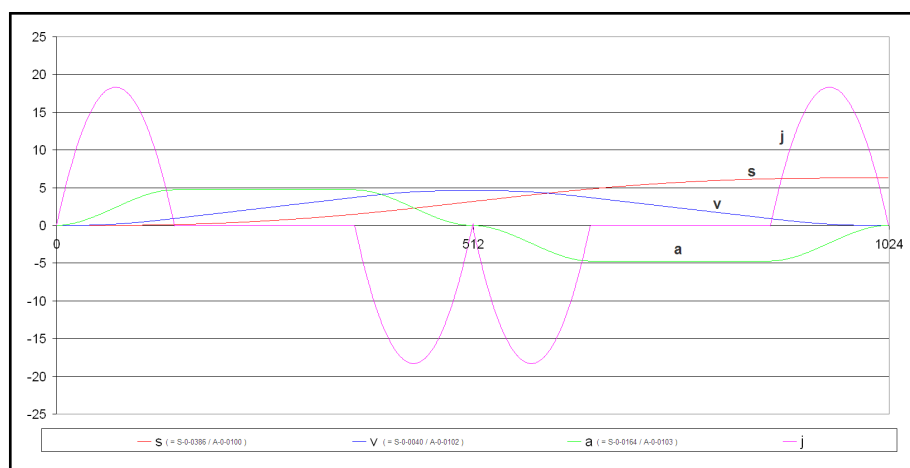


MaxJerk = 10 rad/s³; MaxAcc = 10 rad/s²; MaxVel = 10 rad/s; MasterVel = 10 U/min; Modulowert = 2*Pi rad
Berechnung aus Parabelstücken

Abb. 11-21: MB_FeedMovementCalcType02 Kurvenscheibenprofil (Slave = f(Master))

Variante 3: Geschwindigkeitsgrenzwert wird nicht erreicht, Beschleunigungsgrenzwert wird erreicht (Variante kommt in der Praxis nur selten vor)

RMB_TechCam.library

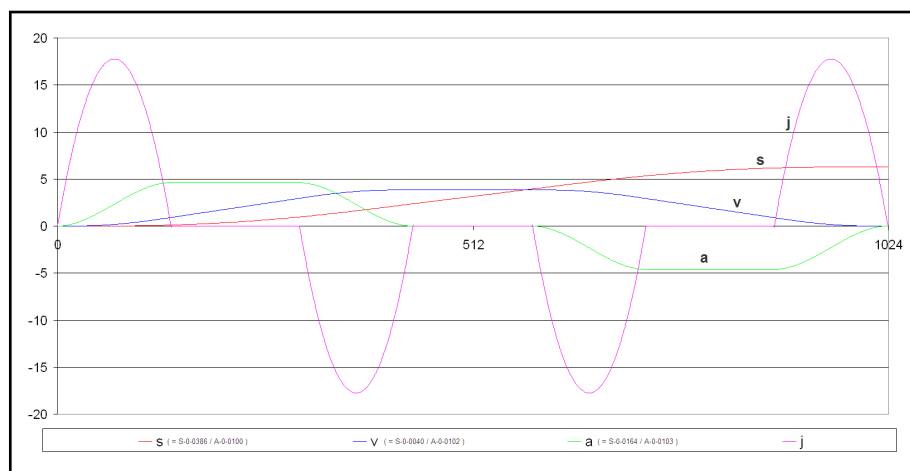


MaxJerk = 20 rad/s³; MaxAcc = 5 rad/s²; MaxVel = 5 rad/s; Master-Vel = 22 U/min; Modulowert = 2*Pi rad

Berechnung aus Parabelstücken

Abb. 11-22: MB_FeedMovementCalcType02 Kurvenscheibenprofil (Slave = f(Master))

Variante 4: Geschwindigkeitsgrenzwert und Beschleunigungsgrenzwert werden erreicht (Variante kommt in der Praxis nur selten vor)

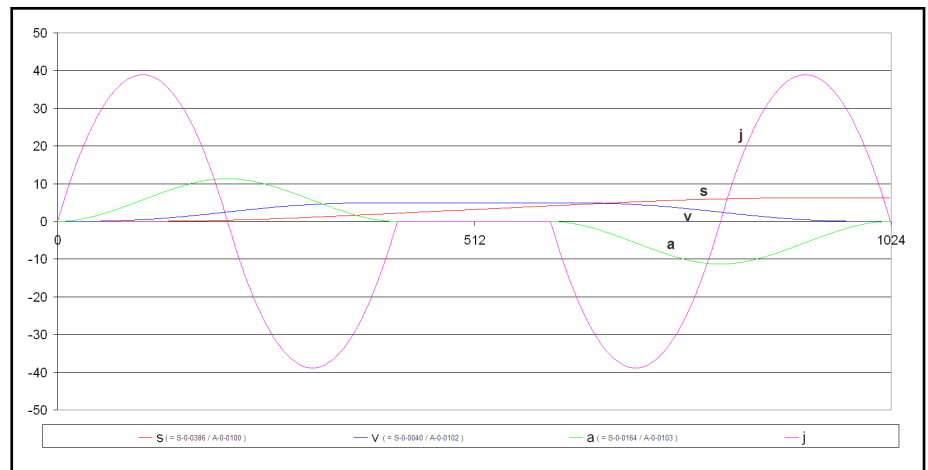


MaxJerk = 20 rad/s³; MaxAcc = 5 rad/s²; MaxVel = 4 rad/s; Master-Vel = 21 U/min; Modulowert = 2*Pi rad

Berechnung aus Parabelstücken

Abb. 11-23: MB_FeedMovementCalcType02 Kurvenscheibenprofil (Slave = f(Master))

Variante 5: Geschwindigkeitsgrenzwert wird erreicht, Beschleunigungsgrenzwert wird nicht erreicht (Variante kommt in der Praxis nur selten vor)



MaxJerk = 40 rad/s³; MaxAcc = 20 rad/s²; MaxVel = 5 rad/s; MasterVel = 28 U/min; Modulowert = 2*Pi rad
Berechnung aus Parabelstücken

Abb. 11-24: MB_FeedMovementCalcType02 Kurvenscheibenprofil (Slave = f(Master))

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#00000006	16#00000000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR	16#00002200	16#00000001	Eingang NumberOfElements ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR	16#00002230	16#00000001	Eingang MaxJerk ist kleiner als Minimum
INPUT_RANGE_ERROR	16#00002230	16#00000002	Eingang MaxAcc ist kleiner als Minimum
INPUT_RANGE_ERROR	16#00002230	16#00000003	Eingang MaxVel ist kleiner als Minimum
INPUT_RANGE_ERROR	16#00002230	16#00000004	Eingang MasterVel ist kleiner als Minimum
INPUT_RANGE_ERROR	16#00002230	16#00000005	Eingang CamShaftDistance ist kleiner als Minimum
INPUT_INVALID_ERROR	16#00002201	16#00000001	Adresse von Feld CamTable ist null
INPUT_INVALID_ERROR	16#00002231	16#00000001	CamShaftDistance ungleich dem Modulowert der Achsreferenz

RMB_TechCam.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#00002231	16#00000002	Eingangswerte für Ruck, Beschleunigung oder Geschwindigkeit sind größer als in den Achsparametern voreingestellt
CALCULATION_ERROR	16#00002232	16#00000001	Geschwindigkeit der Leitachse zu groß oder maximaler Ruck, maximale Beschleunigung oder maximale Geschwindigkeit zu klein

Abb. 11-25: Fehlercodes Funktionsbaustein MB_FeedMovementCalcType02

11.4 MB_CoordinatePairCamType01

Kurzbeschreibung

Der Funktionsbaustein berechnet eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen, die abhängig des ausgewählten Koordinatenpaartyps ermittelt wird.

Zuordnung Zielsystem/Bibliothek

IndraMotion MLC 12VRS	RMB_TechCam.compiled-library

Abb. 11-26: Referenztable MB_CoordinatePairCamType01

Schnittstellenbeschreibung

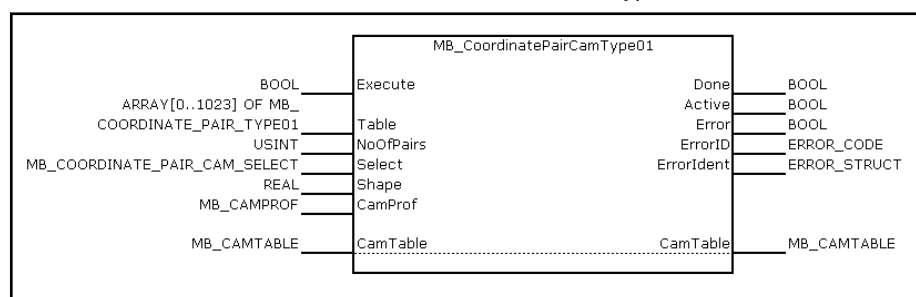


Abb. 11-27: Funktionsbaustein MB_CoordinatePairCamType01

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellenelementen und Information über die Anzahl der Elemente
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	Table	ARRAY[0..1023] OF MB_COORDINATE_PAIR_TYPE01	Liste der Koordinatenpaare
	NoOfPairs	USINT	Anzahl gültiger Koordinatenpaare
	Select	MB_COORDINATE_PAIR_CAM_SELECT	Auswahl der Koordinatenpaartypen
	Shape	REAL	Faktor für Ruck-Begrenzung
	CamProf	MB_CAMPROF	Auswahl des Kurvenscheibenprofils

E/A-Typ	Name	Typ	Kommentar
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe "Fehlerbehandlung" auf Seite 565)

Abb. 11-28: Schnittstellenvariablen Funktionsbaustein MB_CoordinatePairCamType01

Min-/Max- und Default-Werte In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
NoOfPairs	USINT	1 bzw. 2	1023	0	Steigende Flanke an Execute
Select	MB_COORDINATE_PAIR_CAM_SELECT			COORDINATE_PAIR_POS_VEL_TYPE01	Steigende Flanke an Execute
Shape	REAL	0.0	1.0	0.0	Steigende Flanke an Execute
CamProf	MB_CAMPF			CAM_PROF_STANDARD	Steigende Flanke an Execute

Abb. 11-29: Min-/Max- und Default-Werte Funktionsbaustein MB_CoordinatePairCamType01

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
X	LREAL	0.0	<360.0 bzw. 360.0	0.0	Steigende Flanke an Execute	X-Koordinate des Koordinatenpaares (Position)
Y	LREAL	n.def.	n.def.	0.0	Steigende Flanke an Execute	Y-Koordinate des Koordinatenpaares (Position, Geschwindigkeit oder Beschleunigung)

Abb. 11-30: Min-/Max- und Default-Werte der Struktur MB_COORDINATE_PAIR_TYPE01

Die Enumeration "MB_COORDINATE_PAIR_CAM_SELECT" wird verwendet um zwischen verschiedenen Typen (Position, Geschwindigkeit u. Beschleunigung) der Koordinatenpaare zu wählen.

RMB_TechCam.library

Element	Wert	Beschreibung
COORDINATE_PAIR_POS_VEL_TYPE01	0	Koordinatenpaare aus Position (X) und Geschwindigkeit (Y)
COORDINATE_PAIR_POS_VEL_TYPE02	1	Koordinatenpaare aus Position (X) und Geschwindigkeit (Y) und Beachtung des Eingangs-Shape
COORDINATE_PAIR_POS_ACC	2	Koordinatenpaare aus Position (X) und Beschleunigung (Y) und Beachtung des Eingangs-Shape
COORDINATE_PAIR_POS_POS_TYPE01	3	Koordinatenpaare aus Position (X) und Position (Y)
COORDINATE_PAIR_POS_POS_TYPE02	4	Koordinatenpaare aus Position (X) und Position (Y) und Beachtung des Eingangs-Shape

Abb. 11-31: Elemente des Aufzählungstyps MB_COORDINATE_PAIR_CAM_SELECT

Die Enumeration [Kap. 5.14.3 "MB_CAMPFROF" auf Seite 163](#) wird verwendet um zwischen verschiedenen Kurvenscheibenprofilen zu wählen.



Bei der Auswahl des Kurvenscheibenprofils "CAMPFROF_ADDITIVE_SEGMENTABLE" sind folgende Hinweise zu beachten:

- Bei Kurventabellen für Bewegungen mit unbegrenztem (unendlichem) Verfahrbereich, deren erstes und letztes Element identisch sind, ist der Linearpfad über ein Häkchen zu aktivieren und die Reduktion auf den Wert "1" einzustellen, da es sich um ein Profil mit Untersetzung handelt. Bei allen weiteren Kurventabellen ($|\text{letztes\_Element} - \text{erstes\_Element}| = 100\%$) muss der Linearpfad deaktiviert sein, da es sich um Profile ohne Untersetzung handelt.
- Bei Kurventabellen für Bewegungen mit begrenztem (endlichem) Verfahrbereich muss der Linearpfad deaktiviert sein.

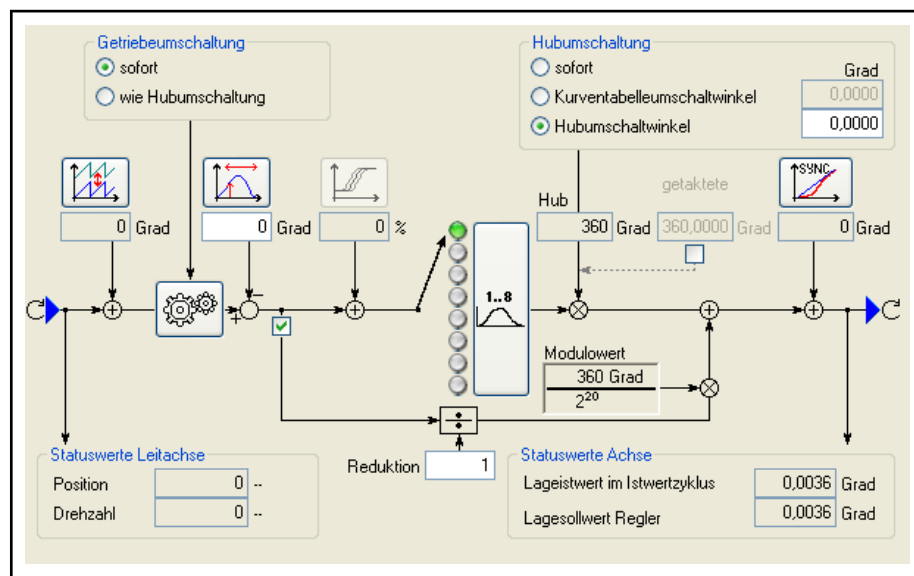


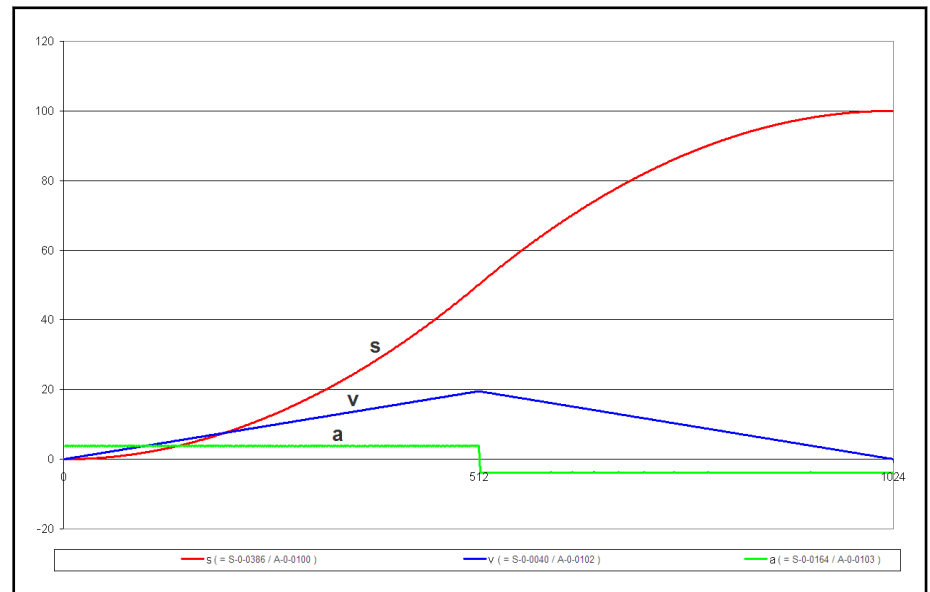
Abb. 11-32: Aktivierung des Linearpfades

Die Struktur [Kap. 5.14.4 "MB_CAMTABLE" auf Seite 163](#) besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Funktionsbeschreibung

Mit der Vorgabe der Koordinatenpaare berechnet der Funktionsbaustein MB_CoordinateCamPairType01 nach der Bearbeitungsfreigabe mittels "Execute" einmalig auf Basis der Eingangsgrößen eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen. Hierbei wird die Anzahl der Stützpunktelemente einer Kurvenscheibentabelle durch das Element "NumberOfElements" der Struktur "CamTable" eingestellt.

- Select: COORDINATE_PAIR_POS_VEL_TYPE01
 - Ist der Stützpunkt (0; 0) kein vorgegebenes Koordinatenpaar, wird dieser Punkt automatisch bei der Berechnung der Kurvenscheibentabelle berücksichtigt.

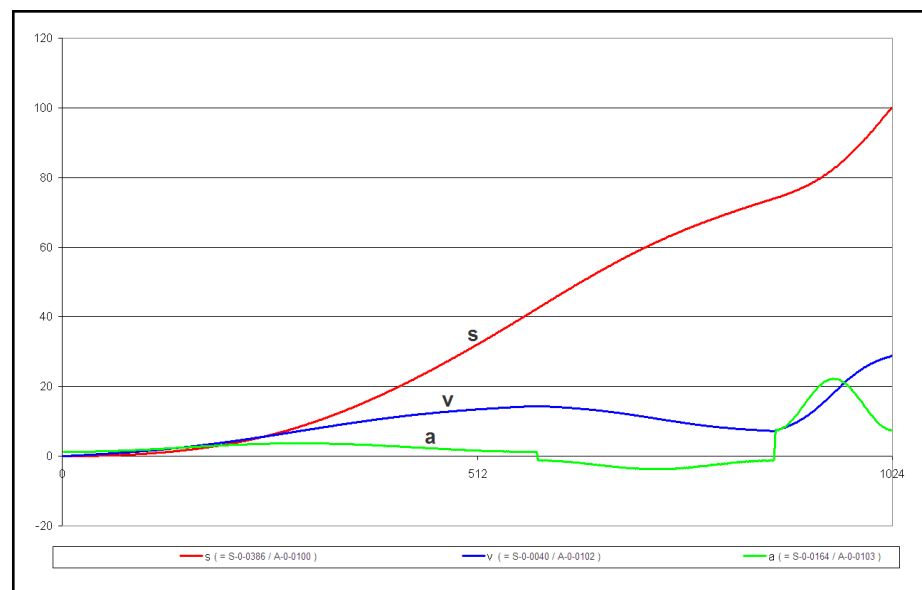


Koordinatenpaare: (180;100)

Abb. 11-33: MB_CoordinatePairCamType01 Bewegungsprofil COORDINATE_PAIR_POS_VEL_TYPE01

- Select: COORDINATE_PAIR_POS_VEL_TYPE02
 - Zur Berechnung der Kurvenscheibentabelle müssen mindestens zwei Koordinatenpaare vorliegen.
 - Durch den Faktor Shape, der zur Begrenzung des Rucks vorgesehen ist, erfolgt ein Verschleifen der Geschwindigkeit.
 - Die berechnete Kurvenscheibe ist lediglich als Endloskurvenscheibe verwendbar, wenn der erste Stützpunkt (0; 0) und der letzte Stützpunkt (360; 0) beträgt. Ist das nicht der Fall, kommt es bei den Übergängen der Kurvenscheiben zu Unstetigkeiten in der Position.
 - Werden die oben genannten Stützpunkte (0; 0 und 360; 0) eingefügt und der Faktor Shape auf 0% gesetzt, ist die erstellte Kurvenscheibe mit der durch die Auswahl COORDINATE_PAIR_POS_VEL_TYPE01 erzeugten Kurvenscheibe identisch.

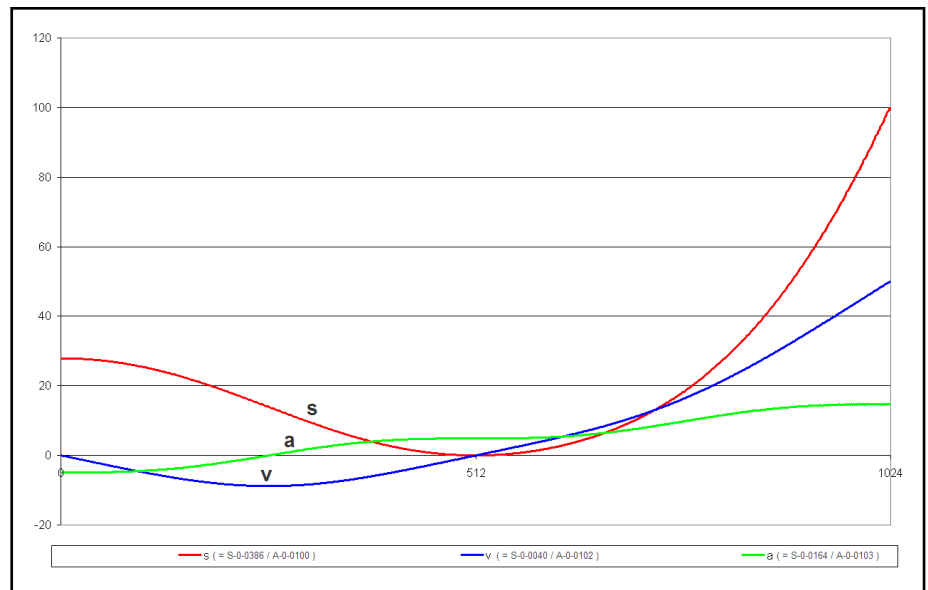
RMB_TechCam.library



Koordinatenpaare: (0;0),(200;200),(300;100),(350;400);
Shape = 0,5

Abb. 11-34: MB_CoordinatePairCamType01 Bewegungsprofil COORDINATE_PAIR_POS_VEL_TYPE02

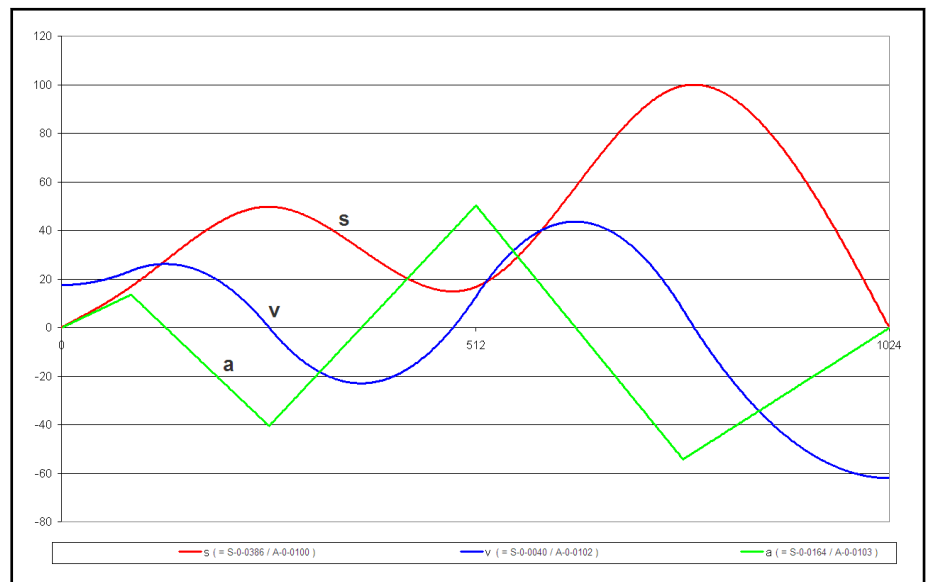
- Select: COORDINATE_PAIR_POS_ACC
 - Diese Auswahl ist einzusetzen, wenn das gewünschte Kurvenscheibenprofil durch einzelne Stützpunkte vorgegeben ist, die mittels einer Rast-in-Rast Bewegung abgefahren werden sollen. Dabei werden die Zielpunkte mit einem Polynom 5. Grades verbunden. Dies hat zur Folge, dass jeder Zielpunkt aus den Eingabedaten mit einer Beschleunigungs-/Bremsbewegung angefahren wird, d.h. an jedem Zielpunkt die Geschwindigkeit = 0 ist.
 - Zur Berechnung der Kurvenscheibentabelle müssen mindestens zwei Koordinatenpaare vorliegen.
 - Durch den Faktor Shape, der zur Begrenzung des Rucks vorgesehen ist, erfolgt ein Verschleifen der Geschwindigkeit.



Koordinatenpaare: (180;100),(360;300); Shape = 1

Abb. 11-35: MB_CoordinatePairCamType01 Bewegungsprofil COORDINATE_PAIR_POS_ACC

- Select: COORDINATE_PAIR_POS_POS_TYPE01
 - Zur Berechnung der Kurvenscheibentabelle müssen mindestens zwei Koordinatenpaare vorliegen.
 - Es werden die einzelnen Wertepaare ohne Rast durchfahren. Dabei werden die Zielpositionen mit einem Spline-Polynom 3. Grades verbunden.



Koordinatenpaare: (0;0),(30;10),(90;30),(180;10),(270;60),(360;0)

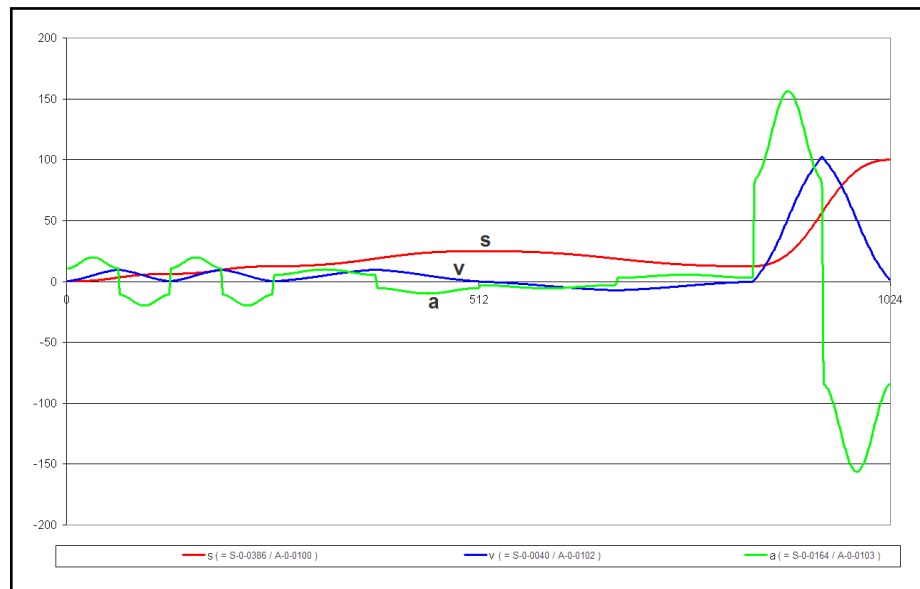
Abb. 11-36: MB_CoordinatePairCamType01 Bewegungsprofil COORDINATE_PAIR_POS_POS_TYPE01

- Select: COORDINATE_PAIR_POS_POS_TYPE02
 - Diese Auswahl ist einzusetzen, wenn das gewünschte Kurvenscheibenprofil durch einzelne Stützpunkte vorgegeben ist, die mit-

RMB_TechCam.library

tels einer Rast-in-Rast Bewegung abgefahren werden sollen. Dabei werden die Zielpunkte mit einem Polynom 5. Grades verbunden. Dies hat zur Folge, daß jeder Zielpunkt aus den Eingabedaten mit einer Beschleunigungs-/Bremsbewegung angefahren wird, d.h. an jedem Zielpunkt die Geschwindigkeit = 0 ist.

- Zur Berechnung der Kurvenscheibentabelle müssen mindestens zwei Koordinatenpaare vorliegen.
- Durch den Faktor Shape, der zur Begrenzung des Rucks vorgesehen ist, erfolgt ein Verschleifen der Geschwindigkeit.



Koordinatenpaare: (0;0),(45;50),(90;100),(180;200),
(300;100),(360;800); Shape = 0,3

Abb.11-37: MB_CoordinatePairCamType01 Bewegungsprofil COORDINATE_PAIR_POS_POS_TYPE02

Die zu berechnende Kurvenscheibentabelle wird in der Struktur "CamTable" mit Wichtung in Prozent abgelegt. Die Gültigkeit der Kurvenscheibenelemente wird durch den Ausgang "Done" signalisiert. Die Kurvenscheibentabelle kann mittels des FB [Kap. 5.14.2 "MB_LoadCamData" auf Seite 160](#) in eine Kurvenscheibe in der Steuerung oder in den Antrieb geladen werden.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" werden im Fehlerfall nicht aktualisiert.

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins MB_CoordinatePairCamType01 weist folgende Eigenschaften auf:

- Der Algorithmus zur Berechnung der Kurvenscheibenstützpunkte ist auf mehrere Zyklen, d. h. Aufrufe der Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden.
- Select: COORDINATE_PAIR_POS_VEL_TYPE01
 - Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist stetig bis zu Geschwindigkeit (s, v) und weist einen Sprung in der Beschleunigung (a) auf
- Select: COORDINATE_PAIR_POS_VEL_TYPE02

- Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist in Abhängigkeit des Faktors Shape stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf
- Select: COORDINATE_PAIR_POS_ACC
 - Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist in Abhängigkeit des Faktors Shape stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf
- Select: COORDINATE_PAIR_POS_POS_TYPE01
 - Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf
- Select: COORDINATE_PAIR_POS_POS_TYPE02
 - Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist in Abhängigkeit des Faktors Shape stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170x. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#00000006	16#00000000	Ungültiger Zustand des Funktionsbausteins
CALCULATION_ERROR	16#0000000C	16#00000000	Division durch Null
INPUT_RANGE_ERROR	16#00002200	16#00000001	Eingang NumberOfElements ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR	16#00002270	16#00000001	Eingang NoOfPairs ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR	16#00002270	16#00000002	Eingang Shape ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR	16#00002270	16#00000003	Eingangselement von Table ist kleiner als Minimum oder größer als Maximum
INPUT_INVALID_ERROR	16#00002201	16#00000001	Adresse von Feld CamTable ist null
INPUT_INVALID_ERROR	16#00002271	16#00000001	Eingangselement von Table ist kleiner als vorheriges Eingangselement

Abb. 11-38: Fehlercodes Funktionsbaustein MB_CoordinatePairCamType01

12 RIL_ParameterChannel.library

12.1 Einführung und Übersicht

Die Bibliothek RIL_ParameterChannel.compiled-library mit ihrem Baustein IL_ParameterChannel ermöglicht eine azyklische Kommunikation über den zyklischen Kanal einer Feldbusverbindung.

12.2 IL_ParameterChannel

Kurzbeschreibung Dieser Funktionsbaustein ermöglicht eine azyklische Kommunikation über den zyklischen Datenkanal. Dabei erfolgt der Datenaustausch über einen Funktionsbaustein auf der Steuerung, sowie einen korrespondierenden Funktionsbaustein im Feldbus-Master.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC/IndraLogic XLC 12VRS	RIL_ParameterChannel.compiled-library

Abb. 12-1: Referenztabelle IL_ParameterChannel

Schnittstellenbeschreibung

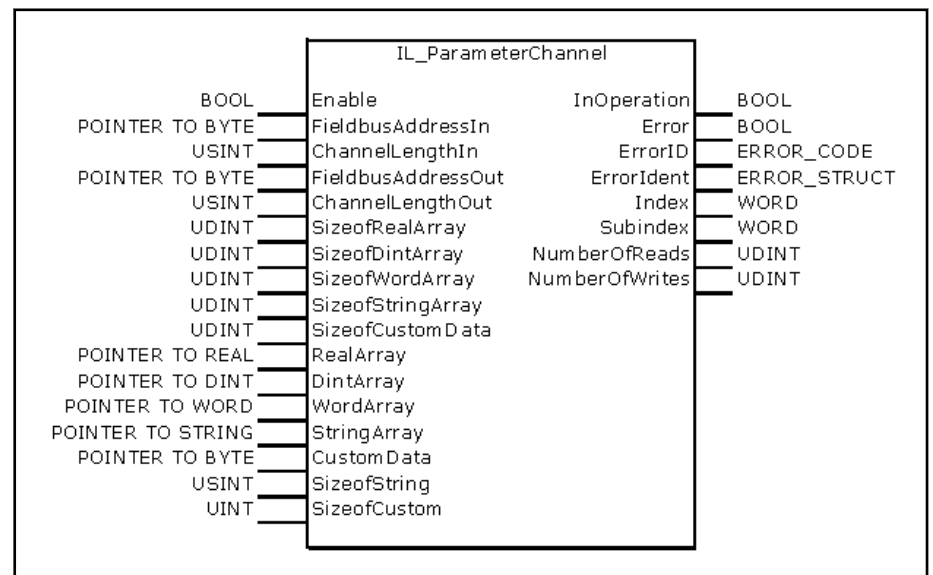


Abb. 12-2: Funktionsbaustein IL_ParameterChannel

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	FieldbusAddressIn	POINTER TO BYTE	Adresse des Eingangsbereiches für den zyklischen Datenverkehr
	ChannelLengthIn	USINT	Länge der zyklischen Eingänge für den Parameterkanal in Bytes
	FieldbusAddressOut	POINTER TO BYTE	Adresse des Ausgangsbereiches für den zyklischen Datenverkehr
	ChannelLengthOut	USINT	Länge der zyklischen Ausgänge für den Parameterkanal in Bytes
	SizeofRealArray	UDINT	Anzahl der REAL-Daten die über die Adresse RealArray zugänglich sind in Bytes

RIL_ParameterChannel.library

E/A-Typ	Name	Datentyp	Beschreibung
	SizeofDintArray	UDINT	Anzahl der DINT-Daten die über die Adresse DintArray zugänglich sind in Bytes
	SizeofWordArray	UDINT	Anzahl der WORD-Daten die über die Adresse WordArray zugänglich sind in Bytes
	SizeofStringArray	UDINT	Anzahl der STRING-Daten die über die Adresse StringArray zugänglich sind in Bytes
	SizeofCustomData	UDINT	Anzahl der gesamten Custom-Daten die über die Adresse CustomData zugänglich sind in Bytes
	RealArray	POINTER TO REAL	Ein Zeiger auf ein Feld von REAL-Werten. Das Feld kann eine Größe von 4845 Elementen besitzen
	DintArray	POINTER TO DINT	Ein Zeiger auf ein Feld von DINT-Werten. Das Feld kann eine Größe von 4845 Elementen besitzen
	WordArray	POINTER TO WORD	Ein Zeiger auf ein Feld von WORD-Werten. Das Feld kann eine Größe von 2295 Elementen besitzen
	StringArray	POINTER TO STRING	Ein Zeiger auf ein Feld von STRING-Werten Das Feld kann eine Größe von 2040 Elementen besitzen
	CustomData	POINTER TO BYTE	Ein Zeiger auf ein Feld von BYTE-Werten. Das Feld kann eine Größe von bis zu 255 * 8 KByte Daten besitzen
	SizeofString	USINT	Größe eines STRINGS im StringArray. Vorsicht: dies ist die Größe, die bei der Definition des Strings angegeben wird (ohne Ende-Kennung)
	SizeofCustom	UINT	Größe eines Datenfeldes (1 – 8192 Bytes)
VAR_OUTPUT	InOperation	BOOL	Der Baustein arbeitet korrekt, Ausgangsdaten gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	Index	WORD	Letzter erfolgreich verwendeter Index
	Subindex	WORD	Letzter erfolgreich verwendeter Subindex
	NumberOfReads	UDINT	Anzahl der erfolgreichen Lesezugriffe durch den Master. Der Wert kann überlaufen (von 65535 auf 0)
	NumberOfWrites	UDINT	Anzahl der erfolgreichen Schreibzugriffe durch den Master. Der Wert kann überlaufen (von 65535 auf 0)

Abb. 12-3: Schnittstellenvariablen Funktionsbaustein IL_ParameterChannel

Min-/Max- und Default-Werte und Übernahme der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
FieldbusAddressIn	POINTER TO BYTE	0x00000000	0xFFFFFFFF	0x00000000	Steigende Flanke an Enable
ChannelLengthIn	USINT	6	18	12	Steigende Flanke an Enable

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
FieldbusAddressOut	POINTER TO BYTE	0x00000000	0xFFFFFFFF	0x00000000	Steigende Flanke an Enable
ChannelLengthOut	USINT	6	18	12	Steigende Flanke an Enable
SizeofRealArray	UDINT	0	19380(4845 * 4)	0	Steigende Flanke an Enable
SizeofDintArray	UDINT	0	19380(4845 * 4)	0	Steigende Flanke an Enable
SizeofWordArray	UDINT	0	4590(2295 * 2)	0	Steigende Flanke an Enable
SizeofStringArray	UDINT	0	522240(2040 * 256)	0	Steigende Flanke an Enable
SizeofCustomData	UDINT	0	2088960(255 * 8192)	0	Steigende Flanke an Enable
RealArray	POINTER TO REAL	0x00000000	0xFFFFFFFF	0x00000000	Steigende Flanke an Enable
DintArray	POINTER TO DINT	0x00000000	0xFFFFFFFF	0x00000000	Steigende Flanke an Enable
WordArray	POINTER TO WORD	0x00000000	0xFFFFFFFF	0x00000000	Steigende Flanke an Enable
StringArray	POINTER TO STRING	0x00000000	0xFFFFFFFF	0x00000000	Steigende Flanke an Enable
CustomData	POINTER TO BYTE	0x00000000	0xFFFFFFFF	0x00000000	Steigende Flanke an Enable
SizeofString	USINT	0	255	80	Steigende Flanke an Enable
SizeofCustom	UINT	0	8192	128	Steigende Flanke an Enable

Abb. 12-4: Min-/Max- und Default-Werte Funktionsbaustein IL_ParameterChannel

Funktionsbeschreibung

Der Baustein ermöglicht eine azyklische Kommunikation über einen zyklischen Kanal. Dazu wird ein Bereich aus dem zyklischen Kanal für die Kommunikation genutzt. Für Daten vom Master zum Slave werden die Eingänge "FieldbusAddressIn" und "ChannelLengthIn" verwendet. Für die Richtung vom Slave zum Master werden die Eingänge "FieldbusAddressOut" und "ChannelLengthOut" verwendet. Die Begriffe "out" und "in" beziehen sich hier auf die Richtung aus Sicht des Slaves. Die Datenbereiche können an beliebiger Stelle innerhalb der zyklischen Daten liegen und auch unterschiedlich lang sein. Die minimale Länge des Parameterkanals beträgt 6, die maximale Länge in beide Richtungen 18 Byte.

SPS-Daten können nur über Datenfelder ausgetauscht werden. Dabei müssen die unterschiedlichen Datentypen berücksichtigt werden. Es gibt fünf verschiedene Datentypen:

- REAL (bis zu 4845 Datenfelder)
- DINT (bis zu 4845 Datenfelder)
- WORD (bis zu 2295 Datenfelder)

RIL_ParameterChannel.library

- STRING (bis zu 2040 Datenfelder)
- Custom (bis zu 255 Datenfelder)

Alle Datenbereiche werden als Zeiger dem Baustein zugewiesen. Die zur Verfügung stehende Größe muss dann jeweils auch angegeben werden. Ist die Datengröße 0 oder ist der zugewiesene Zeiger 0, so interpretiert der Baustein dies als nicht vorhandenen Datenbereich und wird Anfragen auf diese Daten mit Fehler beantworten. Die Datengröße muss in Bytes erfolgen, d.h. der SIZEOF() Operator kann verwendet werden.

Für Strings muss zusätzlich zur Größe des Feldes auch noch die Größe des Strings selbst am Eingang "SizeofString" angegeben werden. Dies ist deshalb erforderlich, da ein String mit unterschiedlicher Größe definiert werden kann. Die Größe der Strings muss innerhalb des Feldes einheitlich sein und darf maximal aus 255 Zeichen bestehen.



die Größe des Strings ist nur die Größe, die bei der Definition des Strings angegeben wird. Indralogic wird aber noch ein weiteres Byte für die Ende Kennung anhängen. Für die Definition

```
strExample: STRING(20);
```

werden in der SPS 21 Bytes belegt, denn nach den 20 Zeichen muss noch die Ende Kennung angefügt werden. Der Funktionsbaustein IL_ParameterChannel berücksichtigt dies automatisch bei der Überprüfung der Größen, so dass im vorherigen Beispiel "SIZEOF(strExample)" als Wert für "SizeofString" angegeben werden muss.

Die CUSTOM-Daten können dazu verwendet werden eine größere Anzahl an Daten konsistent zu übertragen. Dabei kann man sich die CUSTOM-Daten z.B. als Datenstruktur vorstellen, die komplett übertragen werden soll. Es kann bis zu 255 solcher Strukturen geben. Um diese Funktionalität zu nutzen, muss wiederum ein Zeiger zugewiesen werden ("CustomData"). Ebenso muss die Größe einer einzelnen Struktur bekannt sein. Dies erfolgt im Eingang "SizeofCustom". Schließlich wird noch die Gesamtgröße benötigt ("SizeofCustomData").

Da der Funktionsbaustein keine Kenntnis über den Inhalt der Daten besitzt, werden die Daten byteweise kopiert. Es ist Aufgabe der jeweiligen Applikation sicherzustellen, dass das Datenformat auf Slave und Master konsistent ist.



Da bei den CUSTOM-Daten ein Datensatz immer vollständig kopiert wird, nimmt die Kopierzeit einen signifikanten Stellenwert in der Laufzeit des Bausteins ein. Messungen haben ergeben, dass die Laufzeit für das Kopieren und Bearbeiten des Protokolls im maximalen Fall bei:

- 256 Bytes ca. 12us
- 4 KBytes ca. 20us und
- 8 KBytes ca. 40us auf einer IndraControl L65 benötigt werden

Dies muss bei der Taskkonfiguration berücksichtigt werden.



Werden Funktionen zur Prüfung der Gültigkeit von Zeigern und Feldern Grenzen verwendet (z.B. RIL_CheckRtv.library), so nimmt die Laufzeit des Bausteins drastisch zu. Bei CUSTOM-Daten von 4 KByte Daten werden dann anstatt Laufzeiten von 20µs Laufzeiten bis zu 30ms benötigt!

Die Ausgänge "NumberOfReads" und "NumberOfWrites" dienen dazu im Slave eine Aktion durch den Master zu erkennen und darauf reagieren zu können. So kann in der SPS auf dem Slave der jeweilige Wert von "NumberOfReads" und "NumberOfWrites" gespeichert werden. Bei einer Änderung kann dann eine Aktion erfolgen. So ist es z.B. möglich nach einer Änderung von "NumberOfWrites" über den ebenso angegebenen "Index" und "Subindex" den vom Master gerade geschriebenen Wert zu verarbeiten.

Die Ausgänge werden jeweils nur dann aktualisiert, wenn das Lesen, bzw. Schreiben erfolgreich war. Trat allerdings ein Fehler auf, so werden die Zähler nicht erhöht. Die Zähler sind jeweils vom Typ UDINT. Damit haben sie nur einen Wertebereich von 0 bis 65535. Steht der Zähler bei dem Maximalwert von 65535 und es muss erneut inkrementiert werden, so läuft der Zähler über und wird wieder bei 0 beginnen.

Randbedingungen und Einschränkungen:

- Der Baustein darf nur einmal instanziiert werden
- Für SERCOS-Parameter kann nur der Parametersatz 0 gelesen werden
- Es ist nur möglich SERCOS-Parameter als 16-Bit "Ident Number" (IDN) zu lesen. Das Lesen von "Extended Ident Number" (EIDN) ist nicht möglich
- Es kann nur das SERCOS-Element 7 (Daten) bei SERCOS-Parametern gelesen werden

Die Daten können vom Master mittels Index und Subindex angefragt werden. Folgende Liste gibt die entsprechenden Werte für Index/Subindex an.

Startindex (16Bit)	Endindex (16Bit)	Subindex (8Bit)	Rechenvorschrift
0x0000	0x1FFF		Reserviert
0x2000	0x2FFF	Drive # (1-64)	Index = 0x2000 + S-Parameter (nur Parametersatz 0 möglich, keine EIDN)
0x3000	0x3FFF	Drive # (1-64)	Index = 0x3000 + P-Parameter (nur Parametersatz 0 möglich, keine EIDN)
0x4000	0x4FFF	Axis # (1-64)	Index = 0x4000 + A-Parameter (nur Parametersatz 0 möglich, keine EIDN)
0x5000	0x5AEC		Frei
0x5AED	0x5AED	0x01 – 0xFF	Index = 0x5AED; Subindex = Custom Data Array Element; (Element 1 - 255)
0x5AEE	0x5AF7	0x01 – 0xFF	Index = (0x5AEE + [(WORD Array Element - 1) / 255]; Subindex = (WORD Array Element) – [(Index - 0x5AEE) * 255]; (Element 1 - 2295)
0x5AF8	0x5AFF	0x01 – 0xFF	Index = (0x5AF8 + [(STRING Array Element - 1) / 255]; Subindex = (STRING Array Element) – [(Index - 0x5AF8) * 255]; (Element 1 - 2040)
0x5B00	0x5CEB	0x00 – 0xFF	Frei

RIL_ParameterChannel.library

0x5CEC	0x5CFF	0x01 – 0xFF	Index = (0x5CEC + [(REAL Array Element - 1) / 255]; Subindex = (REAL Array Element) – [(Index - 0x5CEC) * 255]; (Element 1 - 4845)
0x5D00	0x5D13	0x01 – 0xFF	Index = (0x5D00 + [(DINT Array Element - 1) / 255]; Subindex = (DINT Array Element) – [(Index - 0x5D00) * 255]; (Element 1 - 4845)
0x5D14	0x5FFF	0x00 – 0xFF	Frei
0x6000	0x6FFF	0x00	Index = 0x6000 + C-Parameter (nur Parametersatz 0 möglich, keine EIDN)
0x7000	0x7FFF	Kinematic # (1-16)	Index = 0x7000 + K-Parameter (nur Parametersatz 0 möglich, keine EIDN)
0x8000	0x8FFF	Probe # (1-99)	Index = 0x8000 + M-Parameter (nur Parametersatz 0 möglich, keine EIDN)
0x9000	0x9FFF	PLS # (1-4)	Index = 0x9000 + N-Parameter (nur Parametersatz 0 möglich, keine EIDN)
0xA000	0xAFFF	Osci # (1-16)	Index = 0xA000 + O-Parameter (nur Parametersatz 0 möglich, keine EIDN)
0xB000	0xFFFF		Frei

Abb. 12-5: Zugewiesene Adressbereiche für Index/Subindex

Fehlerbehandlung

Bei der Fehlerbehandlung muss man zwischen zwei verschiedenen Fehlertypen unterscheiden.

- Fehler beim Datenzugriff und der Datenübertragung
- Fehler in der Konfiguration des Bausteins

Fehler in der Konfiguration (z.B. Parameterkanal zu kurz/lang) werden am Baustein selbst als Fehler angezeigt und müssen durch das SPS-Programm ausgewertet werden.

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#00000001	"FieldbusAddressIn" ist nicht belegt oder hat den Wert 0
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#00000002	"ChannelLengthIn" ist nicht belegt oder hat einen ungültigen Wert (gültige Werte liegen im Bereich 6 <= "ChannelLengthIn" <= 18)
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#00000003	"FieldbusAddressOut" ist nicht belegt oder hat den Wert 0
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#00000004	"ChannelLengthOut" ist nicht belegt oder hat einen ungültigen Wert (gültige Werte liegen im Bereich 6 <= "ChannelLengthOut" <= 18)
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#00000005	"SizeofRealArray" hat einen ungültigen Wert (gültige Werte liegen im Bereich 0 <= "SizeofRealArray" <= 19380)

RIL_ParameterChannel.library

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#00000006	"SizeofDintArray" hat einen ungültigen Wert (gültige Werte liegen im Bereich 0 <= "SizeofDintArray" <= 19380)
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#00000007	"SizeofWordArray" hat einen ungültigen Wert (gültige Werte liegen im Bereich 0 <= "SizeofWordArray" <= 4590)
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#00000008	"SizeofStringArray" hat einen ungültigen Wert (gültige Werte liegen im Bereich 0 <= "SizeofStringArray" <= 522240)
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#00000009	"SizeofCustomData" ist größer als der zulässige Wert (2088960)
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000A	"SizeofRealArray" hat einen ungültigen Wert. "SizeofRealArray" muss durch 4 ohne Rest teilbar sein
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000B	"SizeofDintArray" hat einen ungültigen Wert. "SizeofDintArray" muss durch 4 ohne Rest teilbar sein
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000C	"SizeofWordArray" hat einen ungültigen Wert. "SizeofWordArray" muss durch 2 ohne Rest teilbar sein
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000D	"SizeofStringArray" oder "SizeofString" hat einen ungültigen Wert. "SizeofStringArray" muss durch "SizeofString" + 1 ohne Rest teilbar sein
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000E	"SizeofCustomData" oder "SizeofCustom" hat einen ungültigen Wert. "SizeofCustomData" muss durch "SizeofCustom" ohne Rest teilbar sein
INPUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000F	"SizeofCustom" hat einen ungültigen Wert (gültige Werte liegen im Bereich 0 <= "SizeofCustom" <= 8192)
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#0000000x	Es wurde ein ungültiger Zustand innerhalb des Bausteins erkannt. Baustein muss neu gestartet werden

Abb. 12-6: Fehlercodes Funktionsbaustein IL_ParameterChannel

Fehler beim Datenzugriff (z.B. nicht vorhandenen Index lesen/schreiben) und der Datenübertragung (z.B. falsche Längenangabe) werden nicht am Baustein Ausgang sichtbar. Diese werden innerhalb des Parameterkanals als Fehler gekennzeichnet und Fehlerwerte angegeben. Der Master muss diese Fehlerwerte auswerten.

RIL_ParameterChannel.library

ErrorCode	Text
16#8500	Die übertragene Datenmenge ist zu groß für den internen Zwischenpuffer
16#8B00	Die Formatangabe im Steuerwort ist falsch (muss 16#0C sein)
16#8C00	Die angegebene Länge ist größer als der Parameterkanal selbst
16#9500	Es wurden Daten mitgesendet, obwohl eine Leseanforderung gemacht wurde
16#9600	Bit "C1" ist im Steuerwort nicht gesetzt
16#9700	Das "Last" Bit wurde nicht gesendet, obwohl es sich um eine Leseanforderung handelt
16#1F00	Steuerungsspezifischer Fehler. Weitere Fehlercodes (siehe Fehlertabelle der Steuerung) folgen in den nächsten vier Worten. Im Wort 2 und 3 ist der Fehlercode "Additonal1" und in Wort 4 und 5 der Fehlercode "Additional2" enthalten
16#F200	Der angegebene Index ist unbekannt
16#F300	Der angegebene Subindex ist unbekannt

Abb. 12-7: Fehlercodes innerhalb des Parameterkanals, die vom Funktionsbaustein IL_ParameterChannel erzeugt werden können

12.3 IL_ParameterChannel – Protokoll, Fehlerbehandlung und Inbetriebnahme

12.3.1 Beschreibung des unterlagerten Protokolls

Parameterkanal im zyklischen Datenkanal

Das verwendete Protokoll im Parameterkanal verwendet lediglich ein Steuerwort, um vom Master Daten an den Slave zu senden.

Damit der Slave Daten an den Master senden kann, verwendet er das Statuswort.

Der Master kann entweder Daten schreiben oder Daten lesen. Dabei kann nur der Master eine Datenübertragung anstoßen. Steuer- und Statuswort sind immer fest im Parameterkanal enthalten, wie dies im folgendem Bild zu sehen ist:

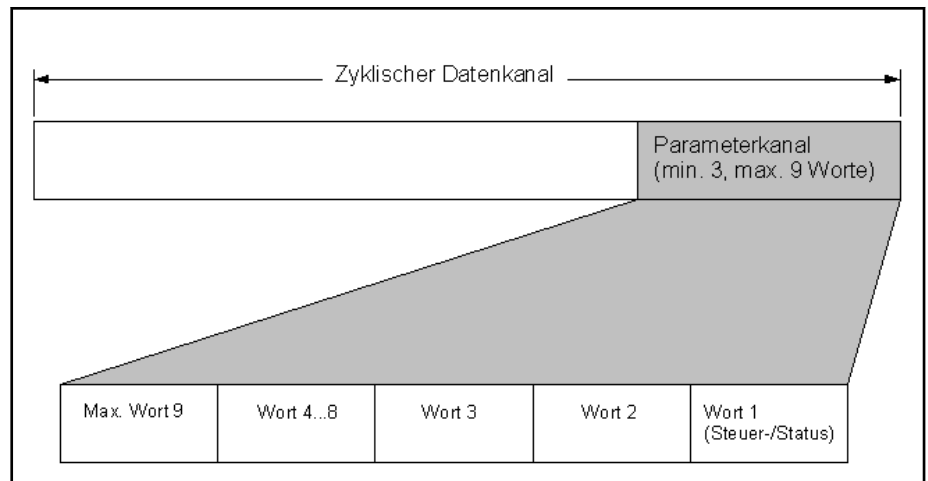


Abb. 12-8: Dateninhalt des Feldbus mit Parameterkanal

Steuer-/Statuswort

Das Steuerwort hat folgende Belegung:

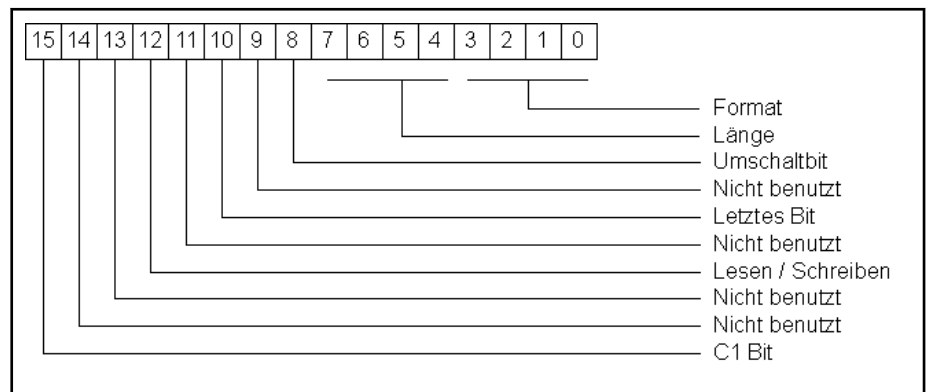


Abb. 12-9: Belegung des Steuerwortes im Parameterkanal

Das Statuswort hat folgende Belegung:

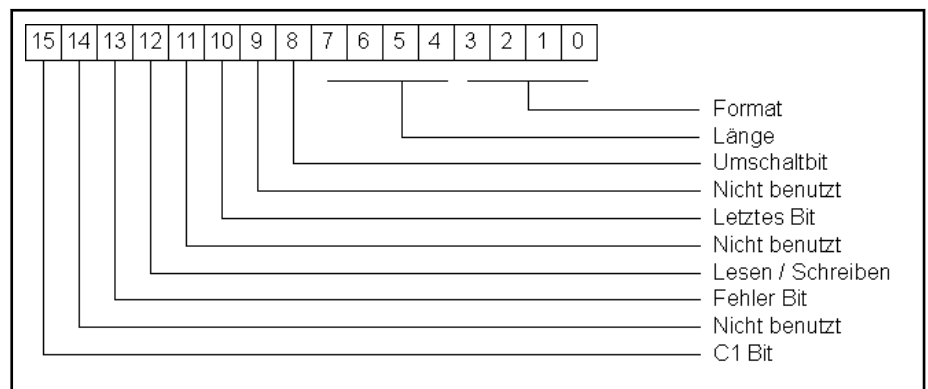


Abb. 12-10: Belegung des Statuswortes im Parameterkanal

Die einzelnen Bits haben folgende Bedeutung:

Format:

Diese Bits geben Verwendungszweck und Bedeutung der folgenden Datenworte im Parameterkanal an. Ihr Wert ist, außer bei Abbruch und Initialisierung der Kommunikation, auf 1100b fixiert.

RIL_ParameterChannel.library

Länge:	Diese vier Bits spezifizieren die Länge der gültigen Daten in Bytes, das Steuerwort nicht mitgerechnet. Die Daten im Rest des Parameterkanals sind nicht definiert.
Umschaltbit (T):	Dieses Bit wird immer dann umgeschaltet, wenn ein neuer Satz Daten gesendet wird. Es stellt einen Handshake zwischen Master und Slave her. Der Master darf dieses Bit nur dann umschalten, wenn das Umschalt-Bit im Statuswort denselben Zustand hat wie das im Steuerwort gesendete Umschalt-Bit.
Letztes Bit (L):	Das letzte Bit wird gesetzt, wenn der letzte Teil des Datenblocks gesendet wird. Es wird verwendet, um ein Fragmentieren zu ermöglichen.
Lesen-Schreiben Bit (R/W):	"Lesen" entspricht dem Bitwert 1, der besagt, dass der Master Daten lesen will. Soll geschrieben werden, muss dieses Bit 0 sein.
C1:	Dieses Bit muss aus Kompatibilitätsgründen zu Altsystemen fest auf 1 gesetzt werden.
Fehlerbit (E):	Dieses Bit signalisiert das Auftreten eines Fehlers im Slave. Die Fehlerursache (Fehlercode) wird mit den folgenden Daten angegeben.



Nicht benutzte Bits haben den Wert 0.

Index und Subindex

Bei der ersten Anforderung durch den Master ist nach dem Steuerwort ein Wort für den Index und ein weiteres Wort für den Subindex vorgesehen. Darüber können alle verfügbaren Daten adressiert werden. Die Zuordnung von Index/Subindex zu den Steuerungsparametern ist wie folgt:

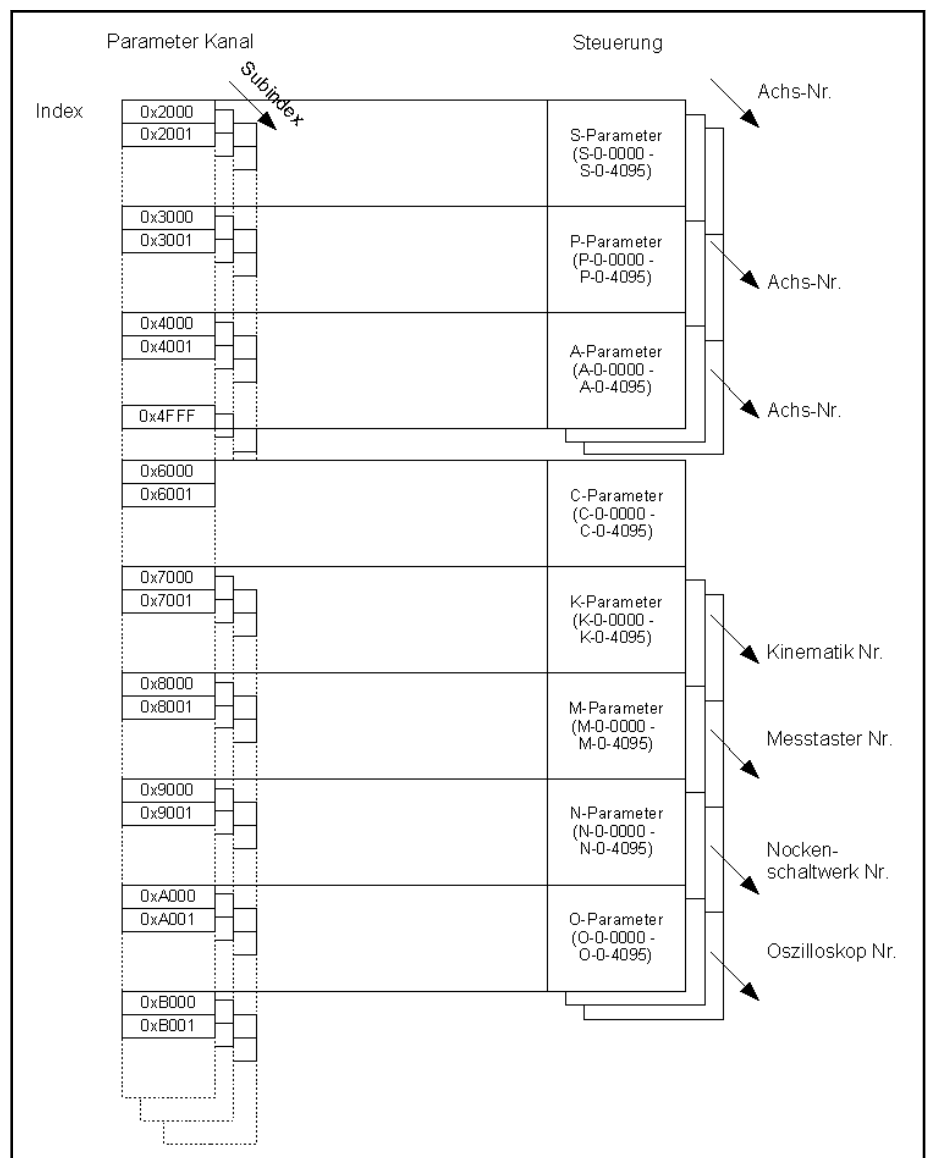


Abb.12-11: Zuordnung von Index/Subindex zu Parametern

Die Zuordnung ist sehr einfach. Um von einem Parameter auf den gewünschten Index/Subindex zu kommen, ist lediglich die Parameternummer zum Startoffset des Indexes zu addieren. Der Subindex ergibt sich aus der jeweiligen Nummer der Achse, Kinematik, etc.

Beispiel:

Es soll der Achsparameter A-0-0100 der Achse 22 gelesen werden.

In diesem Fall muss für den Index $0x4000 + 100 = 0x4064$ und für den Subindex die Achsnummer = 22 gewählt werden.

Die Zuordnung von Index/Subindex zu den SPS-Variablen ist nachfolgend dargestellt. Wichtig hierbei ist, dass der Subindex0 nicht benutzt wird und in der Graphik auch nicht dargestellt ist.

RIL_ParameterChannel.library

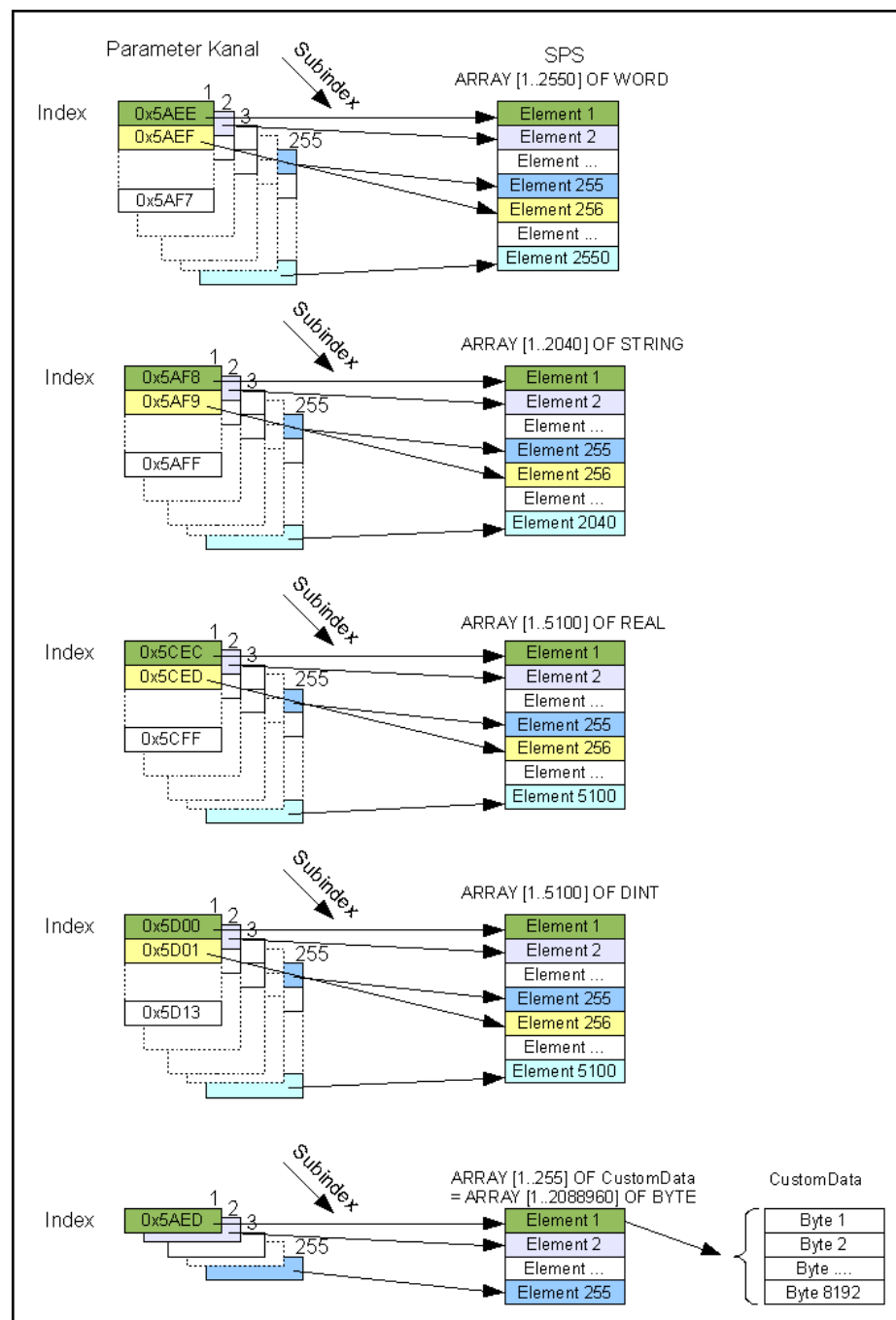


Abb. 12-12: Zuordnung von Index/Subindex zu SPS-Variablen

Hier ist die Abbildungsvorschrift etwas umfangreicher. Um von einer gewünschten Elementennummer auf den Index/Subindex zu kommen, können folgende Umrechnungsformeln für STRING, REAL, DINT und WORD Datentypen verwendet werden:

Index = Startindex + [(Elementnummer - 1) / 255] (die Division ist eine ganzzahlige Division, d.h. Ergebnis wird ohne Nachkommastellen berechnet)

Subindex = Elementnummer - [(Index - Startindex) * 255]

Beispiel:

Es soll das 257. Element des REAL-Feldes gelesen werden.

In diesem Fall ist der Startindex 0x5CEC und die Elementnummer ist 257.

Damit ergibt sich für den Index:

$$\text{Index} = 0x5CEC + (257 - 1) / 255 = 0x5CEC + (256 / 255) = 0x5CEC + 1 = 0x5CED$$

Für den Subindex ergibt sich:

$$\text{Subindex} = 257 - [(0x5CED - 0x5CEC) * 255] = 257 - [1 * 255] = 2$$

Um vom Index/Subindex auf die Elementnummer umzurechnen gilt folgende Formel:

$$\text{Elementnummer} = [(\text{Index} - \text{Startindex}) * 255] + \text{Subindex}$$

Beispiel:

Welches Element wird durch Index 0x5D02 und Subindex 3 angefordert?

Der Index liegt im Bereich 0x5D00 bis 0x5D13, deshalb handelt es sich um ein Element im DINT Feld. Die Elementnummer lautet:

$$\text{Elementnummer} = [(0x5D02 - 0x5D00) * 255] + 3 = 2 * 255 + 3 = 513$$

Für die CUSTOM-Daten gilt die gleiche Abbildungsvorschrift wie oben, da jedoch nur ein Index verwendet wird, kann sie entsprechend vereinfacht werden:

$$\text{Index} = \text{Startindex} = 0x5AED$$

$$\text{Subindex} = \text{Elementnummer}$$

Um im SPS-Programm die Berechnung in umgekehrter Reihenfolge zu ermöglichen (Ermitteln der Elementnummer aus Index und Subindex), kann folgende Formeln verwendet werden:

$$\text{Elementnummer} = (\text{Index} - \text{Startindex}) * 255 + \text{Subindex}$$

Beispiel:

Der Funktionsbaustein zeigt an seinem Ausgang, dass auf Index = 0x5AF9 und Subindex 5 geschrieben wurde. Auf welches Element wurde dadurch geschrieben?

0x5AF9 liegt im Bereich 0x5AF8 bis 0x5AFF. Es handelt sich daher um einen STRING.

Die Elementnummer ergibt sich dann aus:

$$\text{Elementnummer} = (0x5AF9 - 0x5AF8) * 255 + 5 = 0x1 * 255 + 5 = 260$$

Es wurde auf das 260. Element innerhalb des "STRING-Arrays" geschrieben.

Initialisierung der Kommunikation

Um die Kommunikation zu initialisieren, oder auch eine bestehende Anfrage abubrechen, kann der Master eine spezielle Anforderung senden. Diese hat im Steuerwort den Inhalt 1000 000x 0000 1111b. Der Wert des Umschaltbit (T) ist dabei unerheblich. Sendet der Master dieses Steuerwort, so bricht der Slave jegliche Abarbeitung ab und versetzt sich in den Zustand, dass er eine neue Anforderung bearbeiten kann. Er antwortet im Statuswort mit Inhalt 1000 000x 0000 1111b.

Fehlertelegramme

Wenn der Slave einen Fehler ermittelt, so sendet er ein Antworttelegramm an den Master, in dem das Fehlerbit (E) im Statuswort gesetzt ist. Im zweiten Wort ist dann ein Fehlercode enthalten, der die Ursache spezifiziert (siehe dazu auch die [Fehlertabelle, Seite 572](#)).

RIL_ParameterChannel.library

Eine Besonderheit hierbei ist der Fehlercode 0x1F00. Dieser spezifiziert einen Fehler der innerhalb der Steuerung auftrat. In diesem Fall enthält Wort 3 und 4 einen Fehlercode der Steuerung, der in der Dokumentation der Steuerung enthalten ist. In Wort 5 und 6 sind zusätzliche Informationen zum Fehler enthalten. Sollte der Parameterkanal kürzer als 6 Worte sein, so wird das Telegramm fragmentiert.

Beispiel:

Ablauf einer Leseanforderung

Im nachfolgenden Beispiel wird von folgenden Voraussetzungen ausgegangen:

- Parameterkanal ist in beiden Richtungen 8 Byte = 4 Worte lang
- Das Umschaltbit zu Beginn den Wert 0 hat
- Es soll eine SPS-STRING-Variable (Element 1) angefordert werden. Damit muss als Index 0x5AF8 und Subindex 1 gewählt werden
- In diesem Fall antwortet der Slave mit nur einem Telegramm. Ist die Antwort länger, so wäre "Letztes Bit" im Statuswort nicht gesetzt und der Master müsste mit weiteren Telegrammen solange anfordern bis "Letztes Bit" im Statuswort gesetzt ist

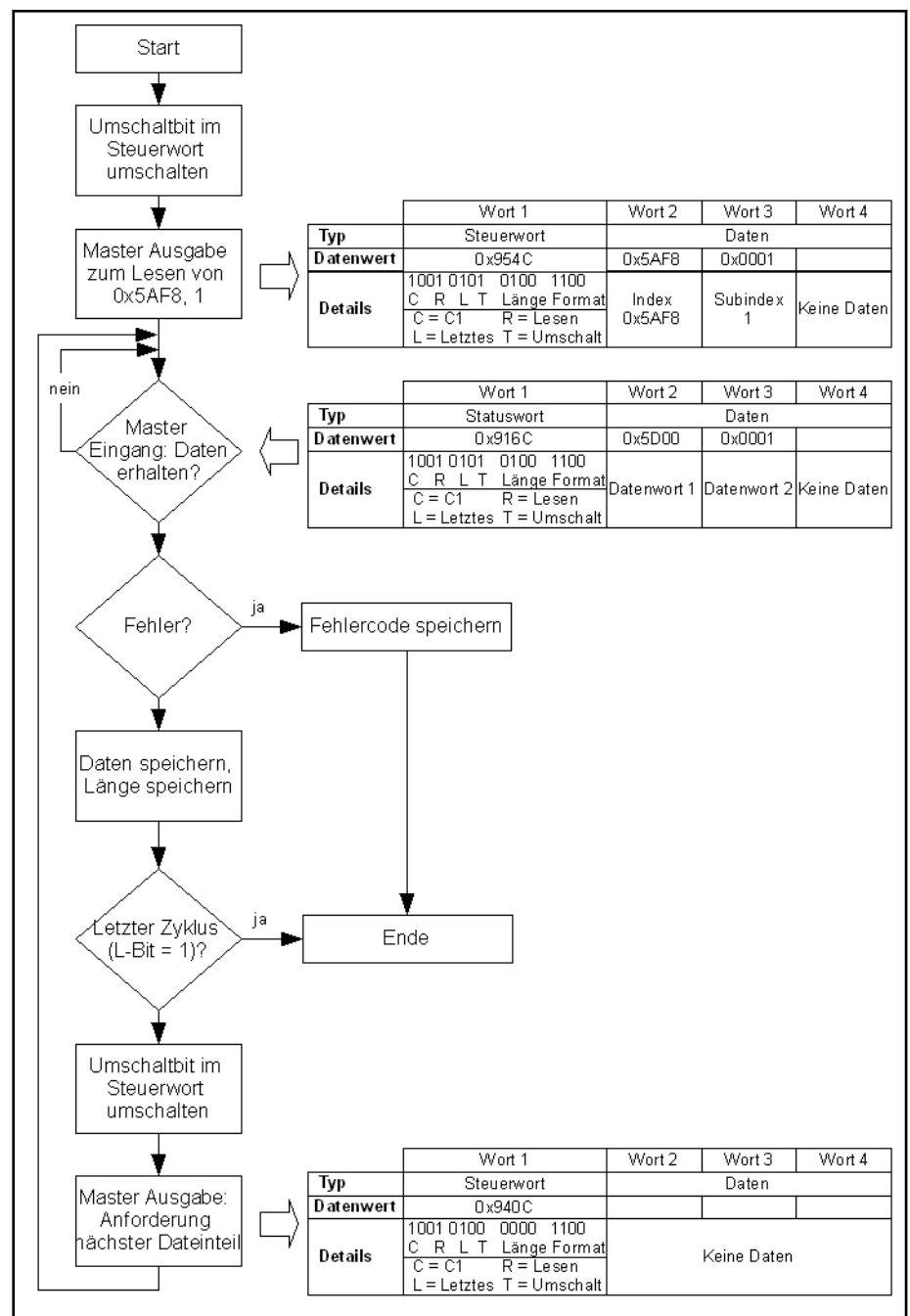


Abb. 12-13: Beispiel zum Lesen einer SPS-STRING-Variablen

Beispiel:

Ablauf einer Schreibanforderung

Im nachfolgenden Beispiel wird von folgenden Voraussetzungen ausgegangen:

- Parameterkanal ist in beiden Richtungen 8 Byte = 4 Worte lang
- Das Umschaltbit zu Beginn den Wert 0 hat
- Es soll der Achsparameter "A-0-0002, Achsname" der Achse 22 geschrieben werden. Damit muss als Index 0x4002 und Subindex 22 gewählt werden

RIL_ParameterChannel.library

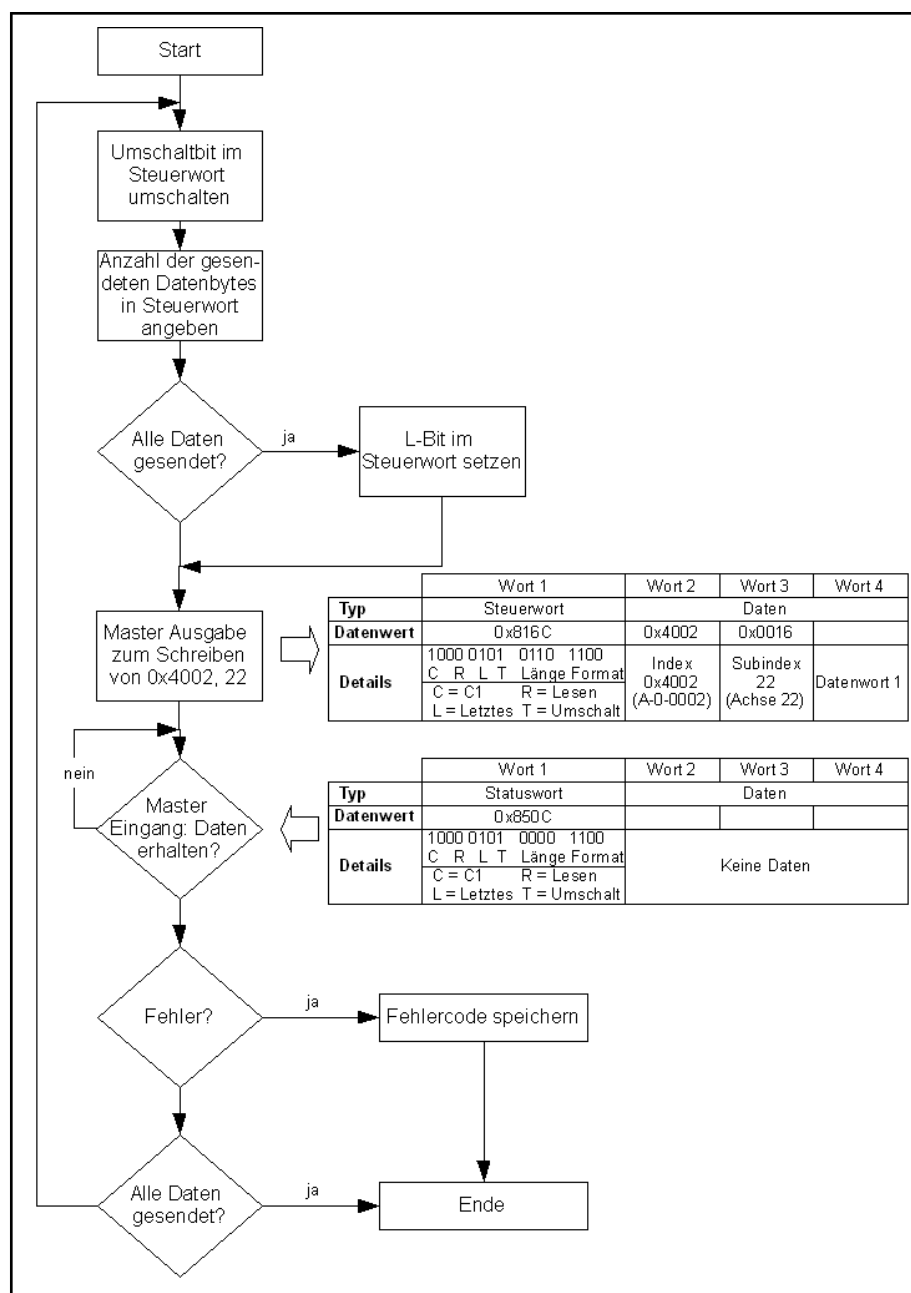


Abb. 12-14: Beispiel zum Schreiben eines Achsparameters

12.3.2 Inbetriebnahme des Bausteins

Bevor der Baustein benutzt werden kann, muss eine Feldbusverbindung bestehen. Am Beispiel des PROFIBUS wird hier gezeigt, wie dieser zum Betrieb konfiguriert wird.

Bei PROFIBUS werden einzelne Module konfiguriert. Für den Parameterkanal wird empfohlen, eigene Module für den Parameterkanal zu verwenden um auch hier eine Strukturierung zu erreichen. Es bieten sich daher Module der Größe 8 und 16 Byte an (Parameterkanal kann 6 bis 18 Bytes lang sein). Alternativ können auch nur Teile eines Moduls verwendet werden. Dies ist dann aber bei der Beschaltung des Parameterkanal-Bausteins zu berücksichtigen. Wichtig ist, dass es sich bei den eingefügten Modulen um Module mit Byte Ausrichtung handelt. Im nachfolgenden Bild sind für den Parameterka-

nal je ein Modul mit 16 Byte als Ein- und Ausgänge definiert ("ParameterChannel_In" und "ParameterChannel_Out").

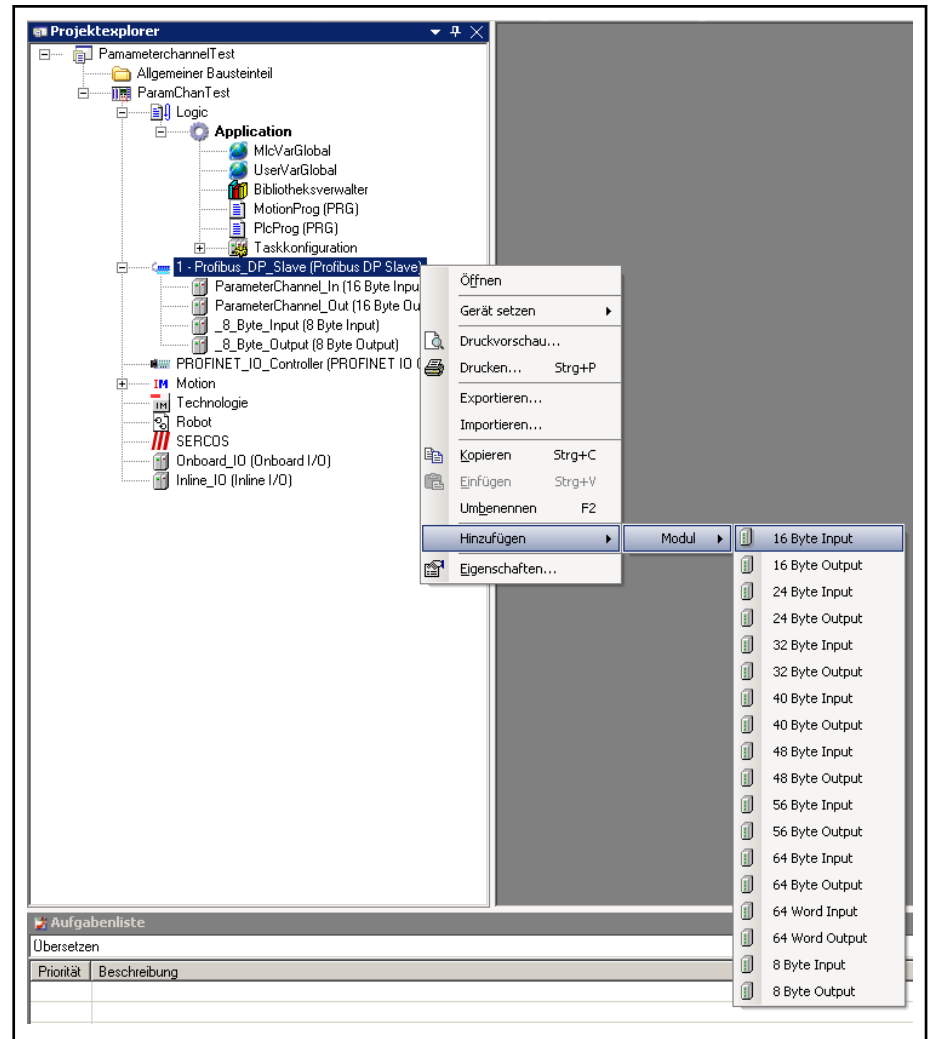


Abb. 12-15: PROFIBUS-Module zur Verwendung als Parameterkanal

Um dem Baustein betreiben zu können, müssen die Adressen für den Feldbus und die jeweilige Länge angegeben werden. Dazu gibt es mehrere Möglichkeiten. Es kann ein symbolischer Name vergeben werden, oder die Adresse des IO-Bereiches benutzt werden.

Im folgenden Bild wurde für die Ausgangsdaten des Feldbusmoduls "ParameterChannel_Out" der Name "myFieldbusOut" vergeben. Dies geschieht durch Doppelklick auf das gewünschte Modul und im sich öffnenden Fenster im Reiter "DP-Module I/O Abbild". Bei anderen Feldbussen hat der Reiter ggf. andere Namen. Alternativ kann auch die Adresse verwendet werden (hier "%QB1").

RIL_ParameterChannel.library

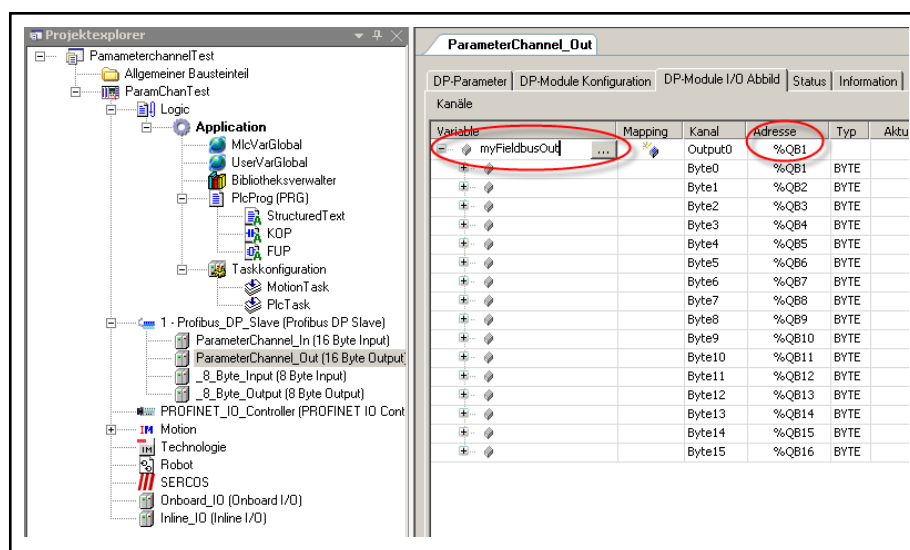


Abb. 12-16: Name und Adresse eines Profibus Moduls

Für die Beschaltung des IL_ParameterChannel Bausteins sind neben den Standard Ein- und Ausgängen noch die Eingänge für SPS-Objekte zu zuweisen. Dabei wird jeweils ein Zeiger sowie die Länge der zugewiesenen Daten benötigt. Eine Besonderheit sind hier die STRING- und CUSTOM-Daten, da diese wiederum selbst anwenderspezifische Längen haben können. Deshalb muss hier auch noch die Länge eines Strings angegeben werden. Wichtig ist in jedem Fall, dass die Felder jeweils bei 1 beginnen. Im nachfolgenden Quelltext sind Beispiele für die Deklaration der SPS-Variablen dargestellt. Für die Felder sind jeweils die maximalen Längen angegeben. Auch sind die Standard-Ein- und Ausgänge, sowie der Funktionsbaustein selbst deklariert.

Deklaration eines FB "IL_ParameterChannel" mit Datensatz

```

TYPE MY_DATA_STRUCTURE :
STRUCT
    // Custom structure, here 2048 * 4 = 8kB Data
    StructureData: ARRAY [1..2048] OF UDINT;
END_STRUCT
END_TYPE

VAR
    // Function block declaration
    Fb: IL_ParameterChannel;
    // PLC Data
    myRealArray:    ARRAY [1..4845] OF REAL;
    myDintArray:    ARRAY [1..4845] OF DINT;
    myWordArray:    ARRAY [1..2295] OF WORD;
    myStringArray:  ARRAY [1..2040] OF STRING(255);
    myCustomData:   ARRAY [1..255] OF MY_DATA_STRUCTURE;
    // In-/outputs of function block
    En:             BOOL;
    InOp:           BOOL;
    Error:          BOOL;
    ErrorID:        ERROR_CODE;
    ErrorIdent:     ERROR_STRUCT;
    Index:          WORD;
    Subindex:       WORD;
    Reads:          UDINT;
    Writes:         UDINT;
END_VAR

```

Der Baustein-Aufruf erfolgt dann wie folgt:

```

1
2  Fb( Enable:= En,
3      FieldbusAddressIn:= %IB1,
4      ChannelLengthIn:= 16,
5      FieldbusAddressOut:= ADR(myFieldbusOut),
6      ChannelLengthOut:= 16,
7      SizeofRealArray:= SIZEOF(myRealArray),
8      SizeofDintArray:= SIZEOF(myDintArray),
9      SizeofWordArray:= SIZEOF(myWordArray),
10     SizeofStringArray:= SIZEOF(myStringArray),
11     SizeofCustomData:= SIZEOF(myCustomData),
12     RealArray:= ADR(myRealArray[1]),
13     DintArray:= ADR(myDintArray[1]),
14     WordArray:= ADR(myWordArray[1]),
15     StringArray:= ADR(myStringArray[1]),
16     CustomData:= ADR(myCustomData[1]),
17     SizeofString:= 255,
18     SizeofCustom:= SIZEOF(MY_DATA_STRUCTURE),
19     InOperation=> InOp,
20     Error=> Error,
21     ErrorID=> ErrorID,
22     ErrorIdent=> ErrorIdent
23     Index=> Index,
24     Subindex=> Subindex,
25     NumberOfReads=> Reads,
26     NumberOfWrites=> Writes);
27

```

Abb. 12-17: Verwendung des Funktionsbausteins IL_ParameterChannel im strukturierten Text

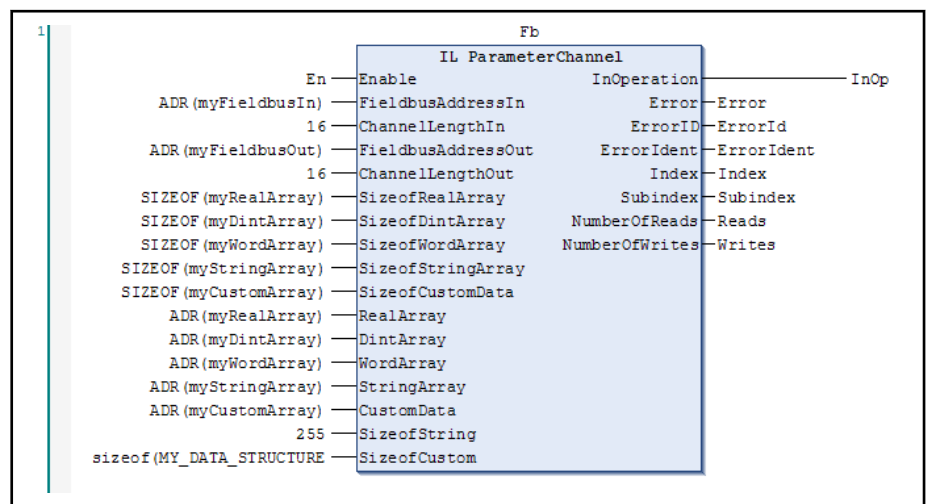


Abb. 12-18: Verwendung des Funktionsbausteins IL_ParameterChannel im FUP

RIL_ParameterChannel.library

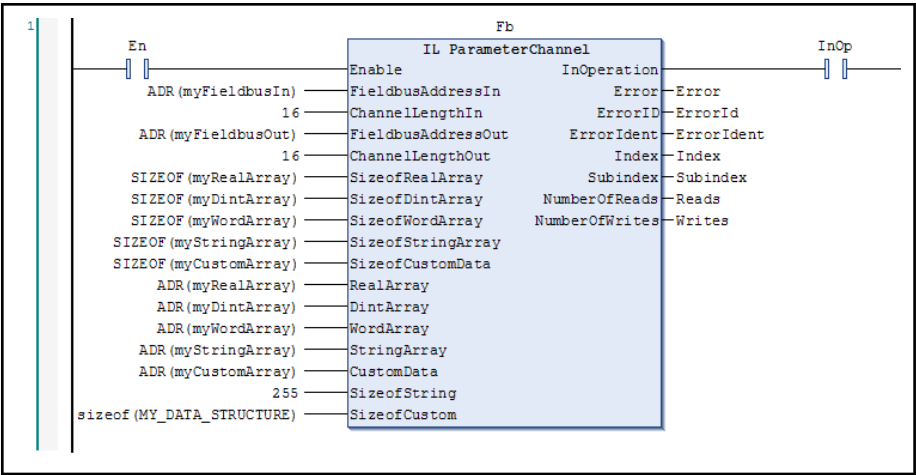


Abb. 12-19: Verwendung des Funktionsbausteins IL_ParameterChannel im KOP
In den obigen Beispielen wurden für die Adressen der Feldbusdaten jeweils im strukturierten Text die Methode mit symbolischer Adresse ("myFieldbusOut") bzw. mit direkter Adressierung ("%IB1") verwendet.

12.3.3 Unterschiede zu Altsystemen

Allgemeines

Der Baustein dient als Ersatz für das bereits bestehende System VisualMotion. Bestehende Unterschiede werden hier beschrieben.

Auf dem Feldbus-Slave

Um den Parameterkanal betreiben zu können, muss auf dem Slave zunächst der Feldbus so eingerichtet werden, dass zusätzliche zyklische Daten für den Parameterkanal konfiguriert werden. Dies war auch bei dem Altsystem der Fall.
Zusätzlich müssen bei der IndraMotion MLC/IndraLogic XLC der Funktionsbaustein IL_ParameterChannel instanziiert, sowie unterschiedliche Felder angelegt werden.
Der Baustein darf nur einmal instanziiert werden. Er wird mit Setzen des "Enable"-Eingangs gestartet und verhält sich im Bezug auf die Standardausgänge ("InOperation", "Error", etc.) genauso wie andere IndraMotion MLC/IndraLogic XLC-Funktionsbausteine.
Die anzulegenden Felder repräsentieren die Variablen und Register in VisualMotion. Wichtig dabei ist, dass die Felder beim Index 1 beginnen! Die Felder von Strings und CustomData sind zusätzlich vorhanden und hatten keine vergleichbaren Variablen in VisualMotion.

IndraMotion MLC/IndraLogic XLC		VisualMotion	
Deklaration	Datentyp	Name	Datentyp
myReals: ARRAY [1..4845] of REAL	REAL (32-bit)	Program Floats (4845 Stück)	FLOAT (32-bit)
myDints: ARRAY [1..4845] of DINT	DINT (32-bit)	Program Ints (4845 Stück)	INT (32-bit)

IndraMotion MLC/IndraLogic XLC		VisualMotion	
myWord: ARRAY [1..2295] of WORD	WORD (16-bit)	Register (2295 Stück)	INT (16-bit)
myString: ARRAY [1..255] of STRING	STRING (myStringSize)	--	--

Abb. 12-20: Gegenüberstellung der SPS-Daten IndraMotion MLC/IndraLogic XLC / VisualMotion

Auf dem Feldbus-Master

Auf der Seite des Masters können bestehende Bausteine weiter verwendet werden. Allerdings müssen Anpassungen in der Beschaltung des Bausteins erfolgen.

Die Adressierung über "Index" und "Subindex" wurde geändert, bzw. erweitert.

Die S- und P-Parameter sind identisch mit dem System VisualMotion.

Für Register, Program Floats und Program Ints ändern sich lediglich die Dateninhalte:

- "Index" 0x5AEE – 0x5AF7 adressierte in VisualMotion die Register, jetzt wird auf das WORD-Feld in der SPS zugegriffen
- "Index" 0x5CEC – 0x5CFF adressierte in VisualMotion die Program Floats, jetzt wird auf das REAL-Feld in der SPS zugegriffen
- "Index" 0x5D00 – 0x5D13 adressierte in VisualMotion die Program Integers, jetzt wird auf das DINT-Feld in der SPS zugegriffen

Datentypen und die Berechnung von "Index" und "Subindex" bleiben aber identisch.

Die A-Parameter wurden kompatibel erweitert. Hier waren bisher nur A-Parameter bis zur Nummer 2047 möglich. Mit diesem Funktionsbaustein kann auch auf A-Parameter bis Nummer 4096 zugegriffen werden.

Alle anderen Daten sind freie oder zusätzliche Daten, die neu hinzu gekommen sind.

Startindex (16Bit)	Endindex (16Bit)	Subindex (8Bit)	Rechenvorschrift
0x0000	0x1FFF		reserviert
0x2000	0x2FFF	Drive # (1-64)	Index = 0x2000 + S-Parameter
0x3000	0x3FFF	Drive # (1-64)	Index = 0x3000 + P-Parameter
0x4000	0x4FFF	Axis # (1-64)	Index = 0x4000 + A-Parameter
0x5000	0x5AEC		Frei
0x5AED	0x5AED	0x01 – 0xFF	Index = 0x5AEDSubindex = Custom Data Array Element (Element 1 - 255)
0x5AEE	0x5AF7	0x01 – 0xFF	Index = (0x5AEE + [(WORD Array Index - 1) / 255]Subindex = (WORD Array Index) – [(Index - 0x5AEE) * 255] (Index 1 - 2295)
0x5AF8	0x5BEC	0x01 – 0xFF	Index = (0x5AF8 + [(STRING Array Index - 1) / 255]Subindex = (STRING Array Index) – [(Index - 0x5AF8) * 255] (Index 1 - 244)
0x5BED	0x5CEB	0x00 – 0xFF	Frei

RIL_ParameterChannel.library

0x5CEC	0x5CFF	0x01 – 0xFF	Index = (0x5CEC + [(REAL Array Index - 1) / 255]) Subindex = (REAL Array Index) – [(Index - 0x5CEC) * 255] (Index 1 - 4845)
0x5D00	0x5D13	0x01 – 0xFF	Index = (0x5CEC + [(DINT Array Index - 1) / 255]) Subindex = (DINT Array Index) – [(Index - 0x5CEC) * 255] (Index 1 - 4845)
0x5D14	0x5FFF	0x00 – 0xFF	Frei
0x6000	0x6FFF	0x00	Index = 0x6000 + C-Parameter
0x7000	0x7FFF	Kinematic # (1-16)	Index = 0x7000 + K-Parameter
0x8000	0x8FFF	Probe # (1-99)	Index = 0x8000 + M-Parameter
0x9000	0x9FFF	PLS # (1-4)	Index = 0x9000 + N-Parameter
0xA000	0xAFFF	Osci # (1-16)	Index = 0xA000 + O-Parameter
0xB000	0xFFFF		Frei

Abb. 12-21: Zugewiesene Adressbereiche für Index/Subindex

Die Fehlercodes innerhalb des Parameterkanals haben sich (teilweise) geändert. Die Unterschiede werden in der folgenden Tabelle dargestellt:

ErrorCode IndraMotion MLC/ IndraLogic XLC	ErrorCode VisualMotion	Beschreibung
16#8500	16#8500	Daten zu lang (VisualMotion: >128 Byte; MLC > 64kByte)
--	16#8800	Fehler auf IndraMotion MLC/IndraLogic XLC nicht vorhanden
16#8B00	16#8B00	Identisch (die Formatangabe im Steuerwort ist falsch (muss 16#0C sein))
--	16#8D00	Fehlerfall wird auf der IndraMotion MLC/IndraLogic XLC bereits beim Start des Bausteins geprüft
16#8C00	16#8C00	Identisch (die angegebene Länge ist größer als der Parameterkanal selbst)
--	16#9000	Format-Bits 0 bis 3 des Steuerwortes haben sich während der Übertragung geändert. Dieser Fehler wird auf der IndraMotion MLC/IndraLogic XLC durch die Fehlernummer 16#8B00 angezeigt
16#9500	16#9500	Identisch (es wurden Daten mit gesendet, obwohl eine Leseanforderung gemacht wurde)
16#9600	--	C1 bit ist nicht gesetzt. Fehler ist auf VisualMotion nicht vorhanden, da hier die alten Formate des Parameterkanals noch unterstützt werden

RIL_ParameterChannel.library

ErrorCode IndraMotion MLC/ IndraLogic XLC	ErrorCode VisualMotion	Beschreibung
16#9700	--	Das "Last" Bit wurde nicht gesendet, obwohl es sich um eine Leseanforderung handelt. Dieser Fehler kann nur auftreten, wenn eine Leseanforderung innerhalb eines Fragmentes übertragen werden kann. Bei VisualMotion war dies nicht garantiert, da der Parameterkanal hier auch nur 2 Worte groß sein durfte. Auf der IndraMotion MLC/IndraLogic XLC ist allerdings festgelegt, dass der Parameterkanal mindestens 3 Worte groß sein muss. Damit kann eine Leseanforderung immer mit einem Telegramm übertragen werden
16#1F00	16#1Fxx	Steuerungsspezifischer Fehler. Bei VisualMotion wird der steuerungsspezifische Fehler im unteren Byte angegeben. Bei der IndraMotion MLC/IndraLogic XLC folgen nach dem Fehlercode noch 8 weitere Bytes. Diese beinhalten die Fehlercodes der unterlagerten Funktionsbausteine zum Lesen/Schreiben von Parametern. In Wort 3 und 4 des Parameterkanals ist der "Additional1" Fehlercode enthalten, im Wort 5 und 6 der "Additional2"
16#F200	16#F200	Identisch (der angegebene Index ist unbekannt)
16#F300	16#F300	Identisch (der angegebene Subindex ist unbekannt)

Abb. 12-22: Unterschiede von IndraMotion MLC/IndraLogic XLC- und VisualMotion-Fehlercodes innerhalb des Parameterkanals

13 ML_TechTensionAdvanced.library

13.1 Einführung und Übersicht

Die Bibliothek ML_TechTensionAdvanced stellt **lizenzpflichtige** Funktionsbausteine für Multiachs Zugspannungsregler mit Entkopplungsnetzwerk zur Verfügung.

Diese Dokumentation beschreibt die Ein- und Ausgänge der einzelnen Funktionsbausteine und gibt Hinweise für deren Verwendung.

Nachfolgende Übersicht zeigt alle Funktionsbausteine der Bibliothek ML_TechTensionAdvanced:

Die Funktionsbausteine von ML_TechTensionAdvanced

Funktionsbaustein	Funktionsbaustein
MB_TensionControlLoadCellType02	Multiachs Zugspannungsregler Funktionsbaustein mit dynamischer Vorsteuerung für bis zu 8 Achsen
MB_TensionDecouplingNetworkType01	Entkopplungsnetzwerk Funktionsbaustein für Multiachs- Zugspannungsregler

Abb. 13-1: Funktionsbausteine der lizenzpflichtigen Bibliothek ML_TechTensionAdvanced



Die Verwendung der Funktionsbausteine der Bibliothek ML_TechTensionAdvanced unterliegen einer Lizenzpflicht. Die entsprechende Lizenz FWS-IM*MLC*-TEC-NNVRS-NN-TENS*ADV muss zur Nutzung der Funktionsbausteine erworben werden. Pro Steuerung, auf der die Funktionsbausteine verwendet werden, ist jeweils eine Lizenz erforderlich (Runtime-Lizenz).

Pro Steuerung muss ein Lizenzblatt erworben werden, das zur Verwendung der Funktionsbausteine der Bibliothek ML_TechTensionAdvanced nur auf diesem Steuerungssystem berechtigt.

13.2 Multiachs Zugspannungsregler MB_TensionControlLoadCellType02

Kurzbeschreibung

Der Zugspannungsregler regelt, abhängig vom Messwert einer Kraftmessdose, die Zugspannung von bis zu 8 Achsen auf den gewünschten Sollwert.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechMotion.compiled-library
IndraMotion MLD/MPx18 mit Zusatzpaket MA (in Vorbereitung)	in Vorbereitung

Abb. 13-2: Referenztablelle Funktionsbaustein MB_TensionControlLoadCellType02



Die Verwendung des Funktionsbausteines unterliegt einer Lizenzpflicht. Details zu der Lizenzpflicht sind in der Einleitung der Dokumentation zu der Bibliothek beschrieben ([siehe Seite 591](#)).

ML_TechTensionAdvanced.library



Ab IndraMotion MLC 12VRS werden zusätzlich auch Slave-Achsen mit Interpolation auf der Steuerung unterstützt.

Schnittstellenbeschreibung

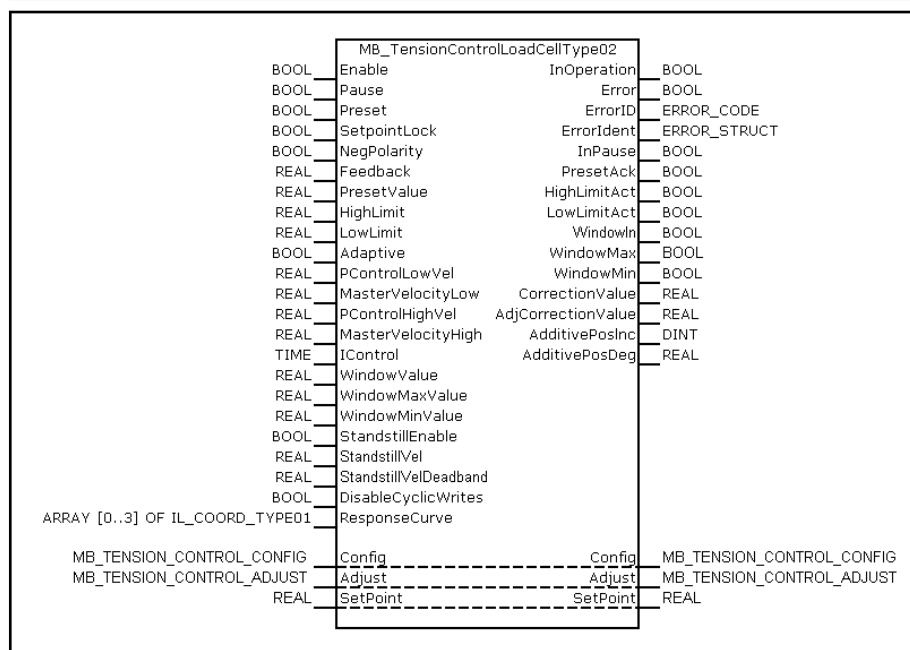


Abb. 13-3: Funktionsbaustein MB_TensionControlLoadCellType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	Config	MB_TENSION_CONTROL_CONFIG	Konfiguration des Zugspannungsreglers (Achsen, Zugspannungsverteilung)
	Adjust	MB_TENSION_CONTROL_ADJUST	Struktur für die Folgeachseinstellungen im laufenden Betrieb
	Setpoint	REAL	Sollwert der Zugspannung Diese wird in der Regel durch den Anwender gesetzt, kann aber auch vom Funktionsbaustein selbst geändert werden (siehe " SetpointLock " auf Seite 602)
VAR_INPUT	Enable	BOOL	Wird dieser Eingang gesetzt, arbeitet der Funktionsbaustein
	Pause	BOOL	Bei gesetztem Eingang wird der Zugspannungsregler angehalten und der I-Anteil des Ausgangswertes eingefroren. Der Ausgang "InPause" signalisiert ein aktives "Pause"
	Preset	BOOL	Bei der steigenden Flanke dieses Eingangs wird der Reglerausgang auf den in PresetValue parametrierbaren Wert gesetzt und bei Erfolg der Ausgang "PresetAck" gesetzt
	SetpointLock	BOOL	Bei steigender Flanke am Eingang wird der aktuelle Zugspannungswert an "Feedback" als neuer Sollwert für den Zugspannungsregler übernommen

E/A-Typ	Name	Datentyp	Beschreibung
	NegPolarity	BOOL	Bei gesetztem Eingang wird die Stellgröße "CorrectionValue" und "AdditivePosInc"/"AdditivePosDeg" mit einem negativen Vorzeichen versehen
	Feedback	REAL	aktueller Istwert der Zugspannung
	PresetValue	REAL	Wert auf den der Reglerausgang gesetzt wird, wenn der Eingang "Preset" gesetzt ist
	HighLimit	REAL	Maximalwert den der Reglerausgang annehmen kann. Wird die vom Regler berechnete Stellgröße größer als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	LowLimit	REAL	Minimalwert den der Reglerausgang annehmen kann. Wird die vom Regler berechnete Stellgröße kleiner als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	Adaptive	BOOL	Bei gesetztem Eingang wird eine adaptive P-Verstärkung für die Regelung verwendet
	PControlLowVel	REAL	P-Verstärkung des Reglers wenn "Adaptive" = FALSE. Ansonsten entspricht der Wert der P-Verstärkung bei niedrigen Leitachsgeschwindigkeiten
	MasterVelocityLow	REAL	Nur wirksam wenn "Adaptive" = TRUE. Leitachsgeschwindigkeit bis zu der "PControlLowVel" verwendet wird
	PControlHighVel	REAL	P-Verstärkung des Reglers bei hohen Leitachsgeschwindigkeiten. Wenn "Adaptive" = FALSE hat dieser Eingang keine Wirkung
	MasterVelocityHigh	REAL	Nur wirksam wenn "Adaptive" = TRUE. Leitachsgeschwindigkeit ab der "PControlHighVel" verwendet wird
	IControl	TIME	Nachstellzeit des I-Reglers. Wird der Wert auf 0 gesetzt, so wird der I-Regler unwirksam
	WindowValue	REAL	Prozessgrößenfenster in [%]
	WindowMaxValue	REAL	Maximalwert der Prozessgröße in [%]
	WindowMinValue	REAL	Minimalwert der Prozessgröße in [%]
	StandstillEnable	BOOL	StandstillEnable = FALSE: Zweig "InVelocity Control" ist aktiv, Stellgröße: Getriebefeinabgleich StandstillEnable = TRUE: Zweig "Regelung im Stillstand" ist aktiv, Stellgröße: Leitachspannung additiv Die in "StandstillVel" angegebenen additive Geschwindigkeit ist wirksam
	StandstillVel	REAL	Additive Geschwindigkeit für die Regelung im Stillstand

ML_TechTensionAdvanced.library

E/A-Typ	Name	Datentyp	Beschreibung
	StandstillVelDeadband	REAL	Dieser Wert bestimmt den Wert der Regelabweichung unterhalb dessen die Spanngeschwindigkeit bei der Regelung im Stillstand reduziert wird
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben. Der Wert wird nur bei steigender Flanke von "Enable" ausgewertet
	ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	Kennlinie zur Gewichtung der Regelabweichung im Zweig "InVelocity Control"
VAR_OUTPUT	InOperation	BOOL	Der Zugspannungsregler arbeitet und die Ausgänge sind gültig
	Error	BOOL	Signalisiert einen Fehler. Mit "Enable" = FALSE wird der Fehler gelöscht
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung gemäß Fehlertabelle (siehe Fehlercodes auf Seite 606)
	InPause	BOOL	"Pause" ist aktiv
	PresetAck	BOOL	Ist dieser Ausgang aktiv, wurde der "Preset" Eingang ausgewertet und der Ausgang "CorrectionValue" hat den Wert "PresetValue" angenommen
	HighLimitAct	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße größer als der in "HighLimit" angegebenen Wert ist
	LowLimitAct	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße kleiner als der in "LowLimit" angegebenen Wert ist
	WindowIn	BOOL	Wenn TRUE, befindet sich die Prozessgröße innerhalb des parametrisierten Fensters
	WindowMax	BOOL	Wenn TRUE befindet sich die Prozessgröße oberhalb des parametrisierten Maximalwertes
	WindowMin	BOOL	Wenn TRUE, befindet sich die Prozessgröße unterhalb des parametrisierten Minimalwertes
	CorrectionValue	REAL	Stellgröße des Zugspannungsreglers
	AdjCorrectionValue	REAL	Stellgröße des Zugspannungsreglers mit dynamischer Vorsteuerung
	AdditivePosInc	DINT	Additive Leitachseposition in Inkrementen je SERCOS-Zyklus, genutzt für reale Slave-Achse mit Interpolation auf dem Antrieb (Zielparameter: P-0-0692)
	AdditivePosDeg	REAL	Ab Version 12VRS: Additive Leitachseposition in Grad je SERCOS-Zyklus, genutzt für Slave-Achse mit Interpolation auf der Steuerung (z.B. für virtuelle Achsen oder reale Achse mit Interpolation auf der Steuerung, Zielparameter: A-0-2600)

Abb. 13-4:

Schnittstellenvariablen Funktionsbaustein MB_TensionControlLoadCellType02

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Pause	BOOL			FALSE	kontinuierlich
Preset	BOOL			FALSE	kontinuierlich
SetPointLock	BOOL			FALSE	kontinuierlich
NegPolarity	BOOL			FALSE	kontinuierlich
Feedback	REAL	n.def.	n.def.	0.0	kontinuierlich
PresetValue	REAL	n.def.	n.def.	0.0	steigende Flanke an "Preset"
HighLimit	REAL	>LowLimit	n.def.	0.00001	kontinuierlich
LowLimit	REAL	-95.0	<HighLimit	-0.00001	kontinuierlich
Adaptive	BOOL			FALSE	kontinuierlich
PControlLowVel	REAL	>0.0	n.def.	0.0001	kontinuierlich
MasterVelocity-Low	REAL	>0.0	<=MasterVelocity-High	0.0	kontinuierlich
PControlHighVel	REAL	>0.0	n.def.	0.0	kontinuierlich
MasterVelocity-High	REAL	>=MasterVelocity-Low	n.def.	1.0	kontinuierlich
IControl	TIME	0s	n.def.	20ms	kontinuierlich
WindowValue	REAL	0.0	100.0	0.0	kontinuierlich
WindowMaxValue	REAL	>=WindowMinValue	n.def.	0.0	kontinuierlich
WindowMinValue	REAL	0.0	<=WindowMaxValue	0.0	kontinuierlich
StandstillEnable	BOOL			FALSE	kontinuierlich
StandstillVel	REAL	0.0	n.def.	0.0	kontinuierlich
StandstillDeadband	REAL	0.0	n.def.	1.0	kontinuierlich
DisableCyclicWrites	BOOL			FALSE	steigende Flanke an "Enable"
ResponseCurve	ARRAY [0..3] OF IL_COORD_TY-PE01	siehe RIL_Loop-Control	siehe RIL_Loop-Control	siehe Datenstruktur	steigende Flanke an "Enable"
Config	MB_TENSION_CONTROL_CONFIG	siehe Datenstruktur	siehe Datenstruktur	siehe Datenstruktur	steigende Flanke an "Enable"

ML_TechTensionAdvanced.library

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Adjust	MB_TENSION_CONTROL_ADJUST	siehe Datenstruktur	siehe Datenstruktur	siehe Datenstruktur	kontinuierlich
Setpoint	REAL	n.def.	n.def.	0.0	kontinuierlich

Abb. 13-5: Min- Max- und Default-Werte Funktionsbaustein MB_TensionControl-LoadCellType02

Min.- Max.- und Default-Werte der Struktur MB_TENSION_CONTROL_CONFIG

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Beschreibung
Master	AXIS_REF			NoAxis	Leitachse
Slave[1]	AXIS_REF			NoAxis	Zugwalze 1
Slave[2]	AXIS_REF			NoAxis	Zugwalze 2
Slave[3]	AXIS_REF			NoAxis	Zugwalze 3
Slave[4]	AXIS_REF			NoAxis	Zugwalze 4
Slave[5]	AXIS_REF			NoAxis	Zugwalze 5
Slave[6]	AXIS_REF			NoAxis	Zugwalze 6
Slave[7]	AXIS_REF			NoAxis	Zugwalze 7
Slave[8]	AXIS_REF			NoAxis	Zugwalze 8
Circumference	REAL	0.0	n.def.	0.0	Achsumfang der ersten Zugwalze
DistanceRoll1To2	REAL	0.0	n.def.	0.0	Bahnlänge zwischen Vorschubwalze und erster Zugwalze

Abb. 13-6: Strukturbeschreibung MB_TENSION_CONTROL_CONFIG

Strukturelement	Typ	Default-Wert	Beschreibung
SlavePreCtrl[1]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 1
SlavePreCtrl[2]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 2
SlavePreCtrl[3]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 3
SlavePreCtrl[4]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 4
SlavePreCtrl[5]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 5
SlavePreCtrl[6]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 6
SlavePreCtrl[7]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 7
SlavePreCtrl[8]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 8

Abb. 13-7: Strukturbeschreibung MB_TENSION_CONTROL_ADJUST

Element	Beschreibung
NO_PRE_CTRL	keine Vorsteuerung/Regelung der Achse
PRE_CTRL_100	Vorsteuerung der Achse ohne Zeitkonstante
PRE_CTRL_DYNAMIC	Dynamische Vorsteuerung

Abb. 13-8: Aufbau des Aufzählungstyps MB_TENSION_PRE_CTRL

Funktionsbeschreibung

Wie in der nachfolgenden Abbildung dargestellt, werden mehrere Zugwalzen durch den Funktionsbaustein MB_TensionControlLoadCellType02 gesteuert. Die Messwerte der Kraftmessdose werden auf den "Feedback"-Eingang des Funktionsbausteins gelegt. Die Sollzugspannung wird über den Eingang "SetPoint" eingestellt. Im normalen Betrieb (Zweig "InVelocity Control" ist aktiv) wird die Zugspannung durch den Getriebefeinabgleich der Zugwalzen geregelt. Je nach Achstyp wird hierbei der entsprechende Parameter beschrieben (P-0-0694 bei Interpolation auf dem Antrieb bzw. A-0-2605 bei Interpolation auf der Steuerung). Bei stehender Leitachse werden die Zugwalzen über die additive Leitachsposition bewegt, da in diesem Fall der Getriebefeinabgleich keinen Einfluss auf die Zugspannung hat. Je nach Achstyp wird hierbei der entsprechende Parameter beschrieben (P-0-0692 bei Interpolation auf dem Antrieb bzw. A-0-2600 bei Interpolation auf der Steuerung).

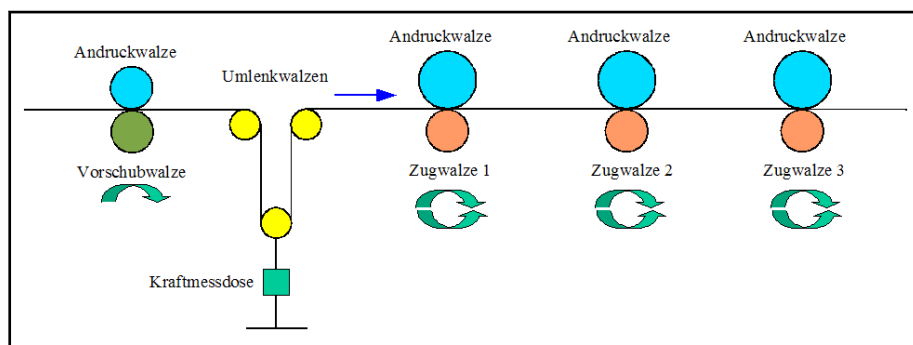


Abb. 13-9: Multiachs-Zugspannungsregler

Im Folgenden werden die einzelnen Funktionen der Eingänge des Funktionsbausteins näher erläutert. Allgemein gilt für alle Eingänge, dass sie kontinuierlich bei gesetztem "Enable"-Eingang ausgewertet werden. Das heißt, Änderungen der Eingänge werden sofort ohne erneutes Toggeln des "Enable"-Eingangs wirksam.

Ausnahme dieser Regel bildet der "PresetValue"-Eingang, der nur bei steigender Flanke am "Preset"-Eingang übernommen wird. Ebenso wird der Eingang "DisableCyclicWrites" nur bei steigender Flanke am "Enable"-Eingang übernommen.

Die gesamte Wirkungsweise des Zugspannungsreglers ist in den folgenden Flussdiagrammen dargestellt. Je nach Eingang "StandstillEnable" ist entweder der Pfad "InVelocity" oder "Regelung im Stillstand" (Standstill Control) wirksam.

ML_TechTensionAdvanced.library

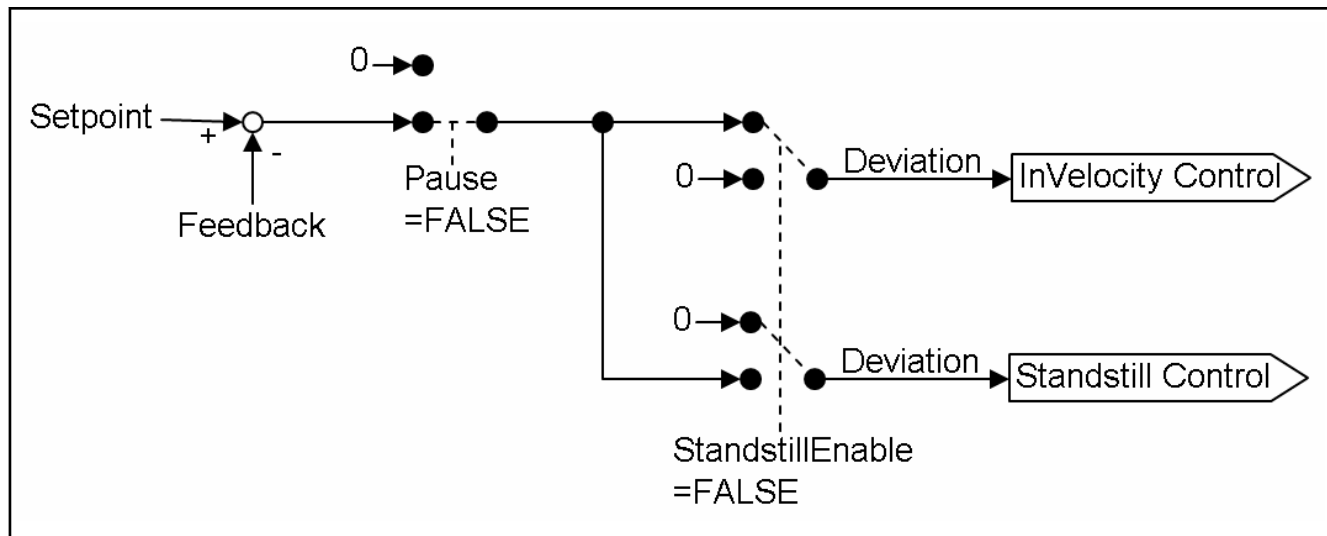


Abb. 13-10: Flussdiagramm des Zugspannungsreglers mit Abzweig für Regelung im Stillstand / In Geschwindigkeit

Der Pfad "InVelocity Control" ist wirksam, wenn der Eingang "Standstill-Enable" nicht gesetzt ist. Die Zugspannung wird hierbei über den Parameter "Getriebefeinabgleich" P-0-0694 (Slave-Achse mit Interpolation auf dem Antrieb / IPO Drive) bzw. A-0-2605 (Slave-Achse mit Interpolation auf der Steuerung/IPO Control) geregelt. Es wird direkt auf die Parameter geschrieben, wenn der Eingang "DisableCyclicWrites" nicht gesetzt ist. Bei realen Achsen mit Interpolation auf dem Antrieb ist daher auch die Konfiguration des Parameters P-0-0694 in den optionalen zyklischen SERCOS-Daten (MDT) erforderlich.

Die Stellgröße wird zusätzlich auch über den Ausgang "CorrectionValue" ausgegeben.

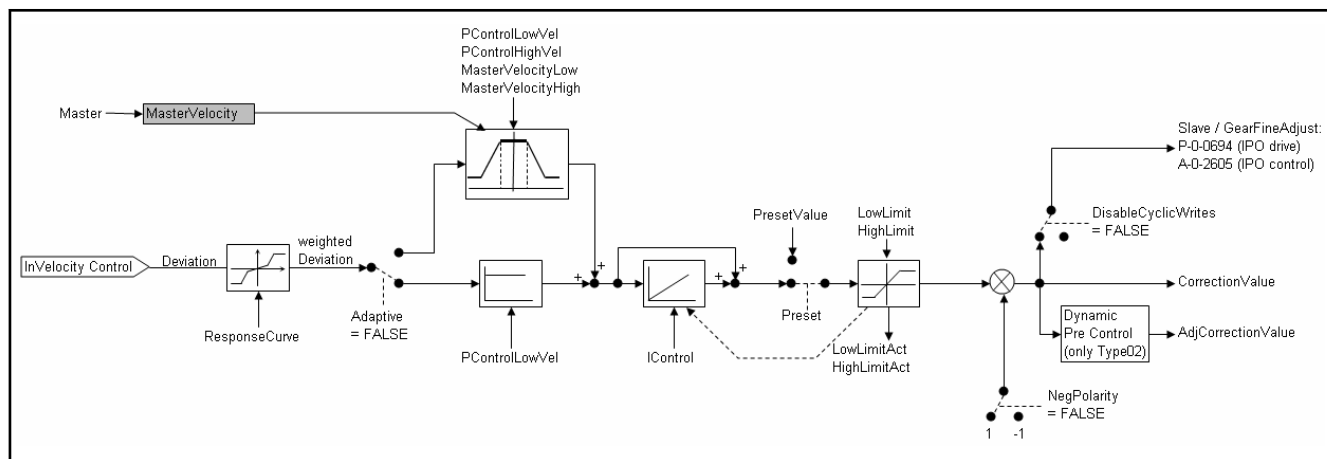


Abb. 13-11: Flussdiagramm des Zugspannungsreglers für den Zweig "InVelocity Control"

Der Pfad "Regelung bei Stillstand der Leitachse" (Standstill Control) ist wirksam, wenn der Eingang "StandstillEnable" gesetzt ist. Dieser Pfad wirkt auf die additive Leitachseposition der Slave-Achse, um damit eine Geschwindigkeit vorzugeben. Es wird direkt auf die Parameter geschrieben, wenn der Eingang "DisableCyclicWrites" nicht gesetzt ist. Bei Slave-Achsen mit Interpolation auf dem Antrieb wird als additive Leitachseposition der Zielparameter P-0-0692 beschrieben. Bei Slave-Achsen mit Interpolation auf der Steuerung wird als additive Leitachseposition der Zielparameter A-0-2600 beschrieben. Bei realen Achsen mit Interpolation auf dem Antrieb ist daher auch die Konfi-

guration des Parameters P-0-0692 in den optionalen zyklischen SERCOS-Daten (MDT) erforderlich.

Gleichzeitig steht die additive Leitachsposition immer über den Ausgang "AdditivePosInc" (bei Interpolation auf dem Antrieb) oder "AdditivePosDeg" (bei Interpolation auf der Steuerung) zur Verfügung.

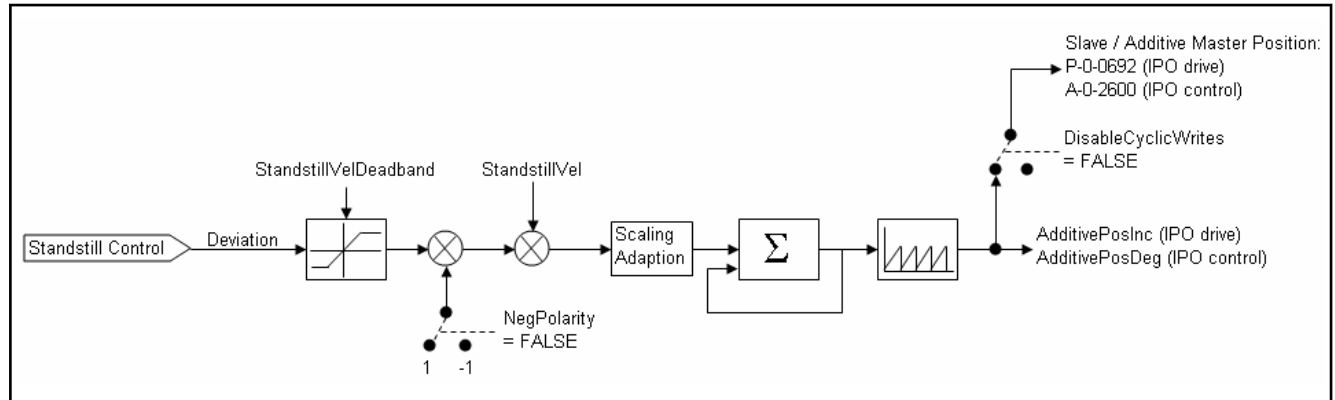


Abb. 13-12: Flussdiagramm des Zugspannungsreglers für die Regelung im Stillstand (Standstill Control)

Das Flussdiagramm enthält aus Gründen der Übersichtlichkeit nicht die Begrenzungen "WindowValue", "WindowMaxValue", "WindowMinValue" sowie deren Statusanzeigen.

Inbetriebnahme

Ist "DisableCyclicWrites" inaktiv, müssen die beiden Parameter P-0-0694 (Getriebefeinabgleich) und P-0-0692 (additive Leitachsposition) im zyklischen Sollwertkanal der Folgeachsen aufgelegt werden, wenn reale Slave-Achsen (Config.Slave[1..8]) mit Interpolation auf dem Antrieb verwendet wird. Bei steigender Flanke an "Enable" werden zyklische Datenkanäle überprüft. Wird einer der beiden Parameter nicht gefunden, erfolgt eine entsprechende Fehlermeldung an den Fehlerausgängen des Funktionsbausteins. Ist "DisableCyclicWrites" aktiv, wird die Konfiguration der zyklischen Sollwertdatenkanäle nicht überprüft.

Bei Slave-Achsen mit Interpolation auf der Steuerung (z.B. virtuelle Achse) ist die Konfiguration der zyklischen Datenkanäle nicht notwendig.



- Die verwendeten Slave-Achsen müssen vor dem Aktivieren des "Enable"-Eingangs in eine synchrone Betriebsart gebracht werden. Dies kann durch das AchsInterface oder die PLCopen-Funktionsbausteine "GearIn" oder "GearInPos" geschehen. Ausnahme: Die synchrone Betriebsart ist nicht erforderlich, wenn der "Pause" Eingang gesetzt ist
- Die Master Achse muss Modulo 360 Grad gewichtet sein
- Es ist möglich die Slave-Achsen gemischt zu konfigurieren (z.B. Slave-Achse 1 mit Interpolation auf der Steuerung, Slave-Achse 2 mit Interpolation im Antrieb). **Bei einer gemischten Konfiguration muss der Parameter P-0-0750 jeder realen Slave-Achse mit Interpolation auf dem Antrieb auf dem Defaultwert = 1 belassen werden**

Config

Über die Struktur "MB_TENSION_CONTROL_CONFIG" werden statische Konfigurationsparameter wie die Achskonfiguration und die Geometrie festgelegt. Der Eintrag "Master" enthält die Achsreferenz auf die Leitachse. Alle Zugwalzen werden im Arrayeintrag "Slave" angegeben. Eine nicht vorhandene Achse wird durch die Konstante "NoAxis" repräsentiert. Für die dynamische Vorsteuerung wird der Umfang der ersten Zugwalze im Struktureintrag



Die Preset Funktion sollte im Stillstand ausgeführt werden (StandstillEnable=TRUE), da ansonsten Geschwindigkeitssprünge durch das Setzen des ControlValue auf den PresetValue entstehen können. Der Betrag der Geschwindigkeitssprünge ist proportional zu der Geschwindigkeit.

Der "Preset"-Eingang wirkt nur bei der steigenden Flanke.

DisableCyclicWrites

Der Eingang "DisableCyclicWrites" bestimmt, ob die vom Funktionsbaustein berechneten Werte direkt über den zyklischen Kanal geschrieben werden (FALSE) oder nur am Bausteinausgang verfügbar sind (TRUE).

Damit ist es möglich, mehrere Bausteine zu kaskadieren, indem man "DisableCyclicWrites" auf TRUE setzt. Dann kann man lediglich die Ausgänge ("CorrectionValue" und "AdditivePosInc" bzw. "AdditivePosDeg") verwenden und weiter modifizieren. Allerdings ist es dann noch notwendig, dass der Anwender die Werte selbst in den Zielparameter schreibt.

Ist "DisableCyclicWrites" auf TRUE gesetzt, wird außerdem die Prüfung, ob die optional zyklischen Parameter auch konfiguriert sind, nicht durchgeführt.

ResponseCurve

Mit Hilfe des Eingang "ResponseCurve" kann die Regelabweichung gewichtet werden. Dies wird durch eine Kennlinie mit bis zu 4 Stützstellen erreicht. Wird der Eingang unbeschaltet gelassen, wird als Default-Wert eine 1-zu-1 Kennlinie verwendet, d.h. es wird keinerlei Gewichtung durchgeführt.

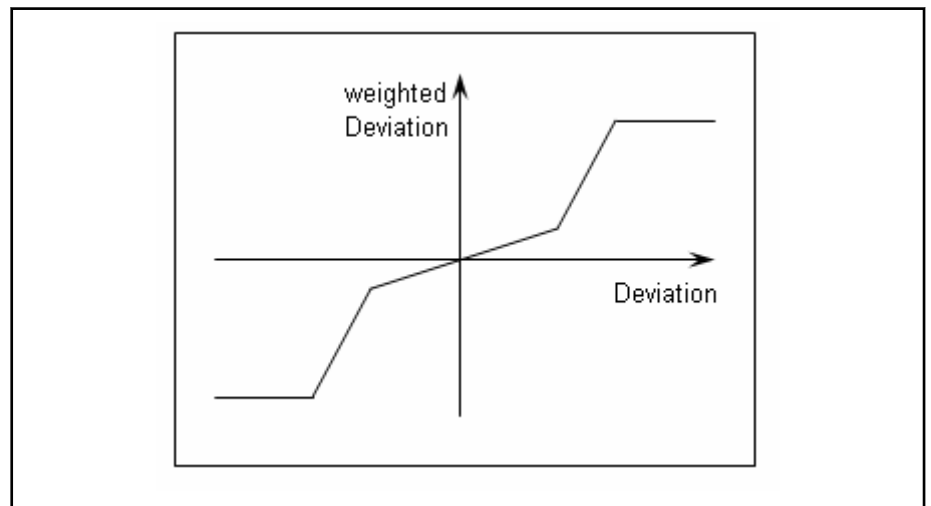


Abb. 13-14: Wichtung der Regelabweichung mit dem Eingang ResponseCurve

Die Funktion kann beispielsweise dafür verwendet werden, bei kleinen Regelabweichungen eine geringere Gewichtung der Regelabweichung vorzugeben. Dies ergibt quasi eine geringere Proportionalverstärkung und dadurch ein ruhigeres Regelverhalten im eingeschwungenen Zustand. Bei der Parametrierung des Eingangs sollten die Reglerparameter zunächst ohne Bewertung eingestellt werden und erst danach eine geringere Gewichtung vorgegeben werden. Hierdurch wird erreicht, dass die Reglerparameter zunächst anhand der Systemstabilität eingestellt werden, und danach die Optimierung für den eingeschwungenen Zustand durchgeführt werden.

Adaptive, PControlLowVel, PControlHighVel, MasterVelocityLow, MasterVelocityHigh

Die P-Verstärkung kann wahlweise als konstante P-Verstärkung oder als adaptive P-Verstärkung vorgegeben werden. Über den Eingang "Adaptive" wird zwischen adaptiver oder konstanter P-Verstärkung gewählt. Durch Setzen von "Adaptive" auf TRUE wird die adaptive P-Verstärkung ausgewählt.

Ist eine konstante P-Verstärkung gewählt, ist nur der Wert in "PControlLowVel" für die P-Verstärkung wirksam.

ML_TechTensionAdvanced.library

Wird die adaptive P-Verstärkung gewählt, müssen die Eingänge "PControlLowVel", "PControlHighVel", "MasterVelocityLow" und "MasterVelocityHigh" beschaltet werden. Die Kennlinie sieht dann folgendermaßen aus:

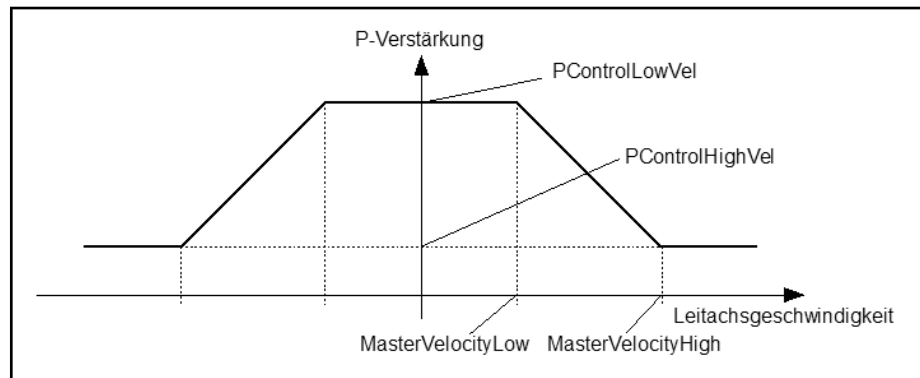


Abb. 13-15: Kennlinie der adaptiven P-Verstärkung

Die einzelnen Eingänge beschreiben die Kennlinie für positive Leitachsgeschwindigkeiten. Die Kennlinie für negative Leitachsgeschwindigkeiten ergibt sich aus der Spiegelung an der Achse der P-Verstärkung. D.h. es sind nur symmetrische Verläufe möglich. Die P-Verstärkung zwischen den Leitachsgeschwindigkeiten "MasterVelocityLow" und "MasterVelocityHigh" wird linear interpoliert. Dadurch ergeben sich einige Bedingungen für die Wertebereiche der Eingaben:

- Die Werte für "MasterVelocityLow" und "MasterVelocityHigh" müssen positiv sein
- Die Werte für "MasterVelocityLow" und "MasterVelocityHigh" dürfen nicht gleich sein
- Der Wert von "MasterVelocityLow" muss kleiner als der Wert von "MasterVelocityHigh" sein

Ist eine dieser Bedingungen verletzt, wird ein Fehler am Ausgang des Bausteines generiert.

SetpointLock Durch Setzen des Eingangs "SetpointLock" wird der aktuelle Zugspannungswert als neuer Sollwert für den Zugspannungsregler übernommen. Die Funktion erlaubt es, z.B. für den Sollwert der Zugspannung einen sinnvollen Startwert zu erhalten, der den momentanen Zustand der Maschine widerspiegelt.

Pause Der Eingang "Pause" ermöglicht das Anhalten des Zugspannungsreglers. Wird der Eingang auf TRUE gesetzt, wird die Regelabweichung auf "0" gesetzt, d.h. der Reglerausgang wird auf dem aktuellen I-Anteil eingefroren. Ist "Pause" aktiv, wird der Ausgang "InPause" gesetzt. Bei aktiver Pause ist die Synchronbetriebsart nicht erforderlich, und es kann die Leistung (MC_Power) abgeschaltet werden.



Wenn keine Zugregelung im Stillstand genutzt wird (Standstill-Enable ist immer FALSE), muss im Maschinenstillstand der Eingang "Pause" gesetzt werden.

NegPolarity Der Eingang "NegPolarity" beeinflusst den Regelsinn. Abhängig von der Lage der Zugwalze, bezüglich der Durchlaufrichtung des Materials und der Lage der Vorschubwalze, muss der Regelsinn entweder positiv oder negativ sein. Ist dieser Eingang FALSE, wird der Regelsinn positiv und eine positive Regelabweichung führt zu einer positiven Stellbewegung. Der Eingang wirkt auch auf die Stillstandsregelung.

Feedback	Dieser Eingang stellt den Istwert der Zugspannung dar. Dieser kann z.B. durch einen Analogeingang am Antrieb oder den Inline I/O eingelesen werden. Der Anwender ist dabei für eine sinnvolle Skalierung verantwortlich.
PresetValue	Durch Setzen des Eingangs "Preset" wird ein Anfangswert über die Variable "PresetValue" für die Stellgröße vorgegeben. Gleichzeitig werden interne Variablen des Reglers so gesetzt, dass ein ruckfreies Anfahren des Reglers erfolgt.
HighLimit, HighLimitAct	Mit diesem Eingang wird die Stellgröße des Reglers auf einen Maximalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der maximale Wert wird durch den im Antrieb definierten Maximalwert für den Getriebefeinabgleich bestimmt. Tritt eine Begrenzung ein, wird der Ausgang "HighLimitAct" gesetzt.
LowLimit, LowLimitAct	Mit diesem Eingang wird die Stellgröße des Reglers auf einen Minimalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der minimale Wert liegt bei -95 unabhängig von ggf. kleineren Werten im Antrieb (siehe dazu auch den Abschnitt "Getriebefeinabgleich"). Tritt eine Begrenzung ein, wird der Ausgang "LowLimitAct" gesetzt.
WindowValue, WindowIn	Mit dem Wert "WindowValue" kann der Anwender ein symmetrisches Fenster definieren innerhalb dessen sich der aktuelle Zugspannungswert befinden soll. Das Fenster ist wie folgt definiert:

$$\left(\text{SetPoint} - |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right) \leq \text{Feedback} \leq \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right)$$

Abb. 13-16: WindowValue, WindowIn

Ist diese Bedingung erfüllt, wird der Ausgang "WindowIn" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Eine Begrenzung wird damit nicht durchgeführt.

WindowMaxValue, WindowMax	Mit diesem Wert kann der Anwender einen oberen Wert für den Zugspannungswert festlegen. Der Wert wird wie folgt ausgewertet:

$$\text{Feedback} > \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowMaxValue} - 100}{100} \right)$$

Abb. 13-17: WindowMaxValue, WindowMax

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMax" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Eine Begrenzung wird damit nicht durchgeführt.

WindowMinValue, WindowMin	Mit diesem Wert kann der Anwender einen unteren Wert für den Zugspannungswert festlegen. Der Wert wird wie folgt ausgewertet:

$$\text{Feedback} < \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowMinValue} - 100}{100} \right)$$

Abb. 13-18: WindowMinValue, WindowMin

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMin" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Eine Begrenzung wird damit nicht durchgeführt.

StandstillEnable	Durch Setzen des Eingangs "StandstillEnable" wird die in "StandstillVel" angegebene Geschwindigkeit additiv auf die zu regelnden Achsen aufgeschaltet. Damit ist eine Zugspannungsregelung auch im Stillstand möglich.

	Der Getriebefeinabgleich verliert bei niedrigen Geschwindigkeiten immer mehr an Einfluss. Bei Stillstand der Leitachse besteht sogar ein offener Regelkreis, da der Getriebefeinabgleich dann keinen Einfluss mehr hat. Daher muss der Eingang "Standstill-Enable" bei Erreichen des Stillstandsfensters gesetzt werden, damit im Pfad "InVelocity Control" der I-Anteil durch den offenen Regelkreis nicht bis zur Grenze ansteigt.
	Bei Achsen mit Interpolation auf der Steuerung wird der Ausgang "AdditivePosInc" im Moduloformat 2 ²⁰ ausgegeben (ab Version 12).
	Wird dieser Eingang verwendet, muss sichergestellt sein, dass der Zugspannungsregler nur in der SERCOS synchronen Task verwendet wird. Ist dies nicht der Fall, kann es zu unruhigem Lauf des Motors, oder Oszillieren um die Stillstands-lage führen.
StandstillVel	Dieser Wert gibt die additive Geschwindigkeit an, die beim Setzen des Eingangs "StandstillEnable" an den Antrieb gesendet wird. Die Einheiten der Geschwindigkeit entsprechen dabei der im Antrieb eingestellten Wichtung.
	Im Stillstand wird die Geschwindigkeit sprunghaft erzeugt. Da die Geschwindigkeit innerhalb eines SERCOS-Zyklus erreicht werden muss, können keine beliebig hohen Werte vorgegeben werden. Wird der Wert der Stillstandsgeschwindigkeit zu hoch gewählt, kann dies zur Oszillation des Antriebes führen. Der Maximalwert richtet sich dabei nach der Wichtung der Geschwindigkeit, sowie der eingestellten maximalen Beschleunigung und angeschlossenen Masse am Antrieb. Um zu große Geschwindigkeitssprünge zu vermeiden, kann eine Filterzeitkonstante im Antriebsparameter P-0-0693 definiert werden, wodurch der Sprung geglättet wird.

Hintergrundinformationen

Getriebefeinabgleich	Der Zugspannungsregler arbeitet in der Regel mit Geschwindigkeiten größer "0". Dabei ist der Getriebefeinabgleich ein probates Mittel um die Zugspannung aufrechtzuerhalten. Allerdings gibt es dabei auch einige Randbedingungen, die hier erläutert werden. Der Getriebefeinabgleich wird intern auf ein Minimum von -95% begrenzt. Wird ein kleinerer Wert eingegeben, erzeugt der Baustein einen Fehler. Hintergrund dieser Begrenzung ist der errechnete Ausgangswert der bei gegebenem Eingangswert wie folgt definiert ist:

$$\text{Ausgangswert} = \text{Eingangswert} * \left(1 + \frac{\text{Getriebefeinabgleich}}{100} \right)$$

Abb. 13-19: Ausgangswert

Anhand der Gleichung kann man erkennen, dass - wenn der Getriebefeinabgleich gegen -100% geht - sich der Ausgangswert gegen "0" bewegt. Damit würde der Regelkreis praktisch geöffnet werden, da eine Änderung des Eingangswertes keinerlei Reaktion am Ausgang bewirkt. Schlimmer noch, bei einem Getriebefeinabgleich der kleiner als -100% wäre, kehrt sich das Vorzeichen des Ausgangswertes um. Im Regelkreis würde eine Mitkopplung entstehen, die das System instabil machen kann.

**StandstillEnable, StandstillVel,
StandstillVelDeadband**

Die Regelung im Stillstand kann durch Verwendung der Eingänge "StandstillEnable" und "StandstillVel" erfolgen. Dabei wird eine additive Geschwindigkeit aufgeschaltet. Um eine bessere Regelgüte zu erreichen, wird diese um den Sollwert herum verringert. Dies geschieht mit Hilfe einer abgewandelten Signum-Funktion, die nachfolgende Übertragungsfunktion besitzt:

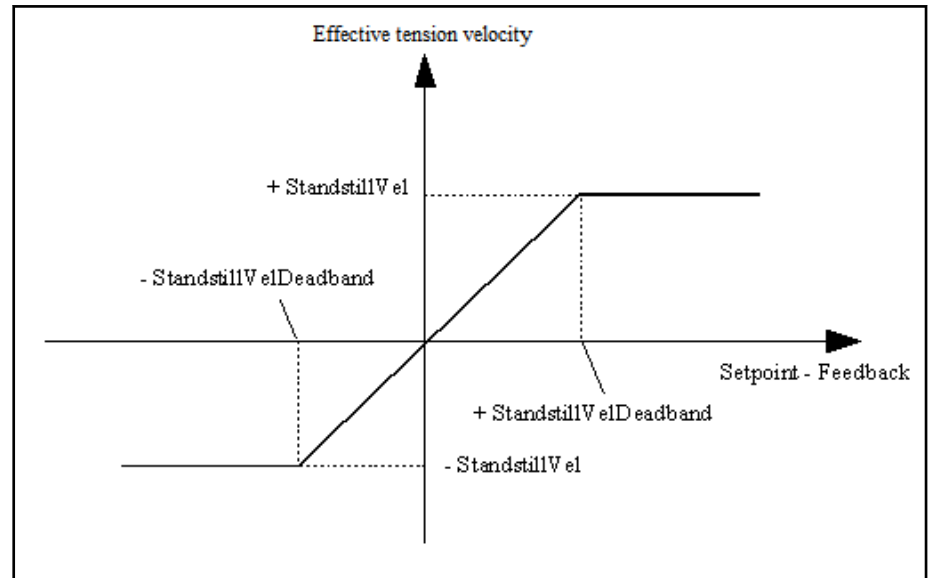


Abb. 13-20: Abgewandelte Signum-Funktion für die Begrenzung der aufgeschalteten StandstillVel

Die Regelabweichung wird dadurch im Bereich des "Setpoint" relativiert. Nähert sich der "Feedback" dem "Setpoint", wird die Regelung "feinfühligere" als ein reiner Zweipunktregler. Der Wert bei dem diese Begrenzung stattfindet kann man am Eingang "StandstillVelDeadband" einstellen. Die Werte besitzen dabei dieselben Einheiten wie der "Setpoint", bzw. "Feedback" Eingang. Zwischen den Extremwerten "+StandstillVel" und "-StandstillVel" wird linear interpoliert, so dass sich obige Übertragungsfunktion für Regelabweichung und Spanngeschwindigkeit ergibt.

Anschließend wird der ermittelte Faktor mit der vorgegebenen "StandstillVel" multipliziert. Die Geschwindigkeit wird dabei als Drehzahl, in der im Antrieb eingestellten Wichtung, verarbeitet. Möchte man anstatt der Drehzahl eine Bahngeschwindigkeit eingeben, ist diese vorher selbst zu berechnen.

Um eine umgekehrte Bewegungsrichtung für die Regelung im Stillstand zu erreichen, ist der Eingang "NegPolarity" zu setzen.

Nachfolgend wird noch mit einem Korrekturfaktor multipliziert. Dieser Korrekturfaktor setzt sich aus dem Getriebeübersetzungsverhältnis zusammen. Jedoch ist der Getriebefeinabgleich nicht berücksichtigt. In der Praxis ist der Getriebefeinabgleich in der Regel sehr klein, so dass dieser nicht ins Gewicht fällt. Das Getriebeübersetzungsverhältnis wird nur bei einer steigenden Flanke des "Enable"-Eingangs übernommen.

Im Flussdiagramm ist ersichtlich, dass die angegebene Geschwindigkeit in die additive Leitachsposition in Inkrementen bzw. Grad umgerechnet wird. Dabei sind einige Regeln zu beachten:

- Die angegebene Geschwindigkeit kann nicht beliebig hoch gewählt werden, denn der Antrieb muss die Geschwindigkeit innerhalb eines SERCOS-Zyklus erreichen können. Damit sind nur sehr kleine Werte zulässig. Die Größe hängt dabei von der verwendeten Wichtung und der Masse am Motor ab. Die errechneten Inkremente/Position werden über

ML_TechTensionAdvanced.library

den Parameter P-0-0692 (Slave-Achse mit Interpolation auf dem Antrieb) bzw. A-0-2600 (Slave-Achse mit Interpolation auf der Steuerung) vorgegeben, dieser kann zudem über Filter P-0-0693 (Slave-Achse mit Interpolation auf dem Antrieb) bzw. A-0-2601 (Slave-Achse mit Interpolation auf der Steuerung) geglättet werden, so dass auch größere Sprünge möglich sind

- Die errechneten Inkremente werden noch vor dem Getriebe aufaddiert. Das Getriebeübersetzungsverhältnis (A-0-2720 und A-0-2721 bzw. P-0-0156 und P-0-0157) wird dabei mit eingerechnet. Der Getriebefeinabgleich (P-0-0083/A-0-2722 und P-0-0694/A-0-2605) wird jedoch nicht berücksichtigt, da dieser in der Regel sehr klein ist. Das kann dazu führen, dass die Achse sich mit einer geringfügig anderen, als der eingestellten Geschwindigkeit bewegt
- Es wird eine Skalierung der Einheiten durchgeführt. Außerdem wird der eingegebene Geschwindigkeitswert nicht nach Maximum- und Minimumwerten überprüft. Dadurch können sich sehr große Werte für die Anzahl der Inkremente/Position je SERCOS-Zyklus ergeben. Überschreitet der errechnete Zwischenwert die Maximalwerte von $\pm$ halber Modulowert, werden diese automatisch begrenzt. Die Folge ist eine niedrigere additive Geschwindigkeit, als am Eingang "StandstillVel" vorgegeben wird
- Bei sehr kleinen Geschwindigkeiten ist außerdem zu beachten, dass die aktuelle Geschwindigkeit von der angegebenen Geschwindigkeit abweichen kann. Dies ist durch die Granulierung der Inkremente je SERCOS-Zyklus zu erklären: entspricht z.B. 1 Inkrement einer Geschwindigkeit von 0,03 U/min, so können nur Geschwindigkeiten die ein Vielfaches von 0,03 U/min sind, wirksam werden. Ist dies nicht der Fall, wird zum nächstgelegenen Wert gerundet.
- Die errechneten Inkremente werden mit jedem SERCOS-Zyklus verarbeitet. Deshalb muss bei Verwendung der Regelung im Stillstand der Baustein in der Motion/SERCOS synchronen Task abgearbeitet werden. Ist dies nicht der Fall, wird sich der Antrieb einen SERCOS-Zyklus um die errechnete Anzahl an Inkrementen bewegen, und in den folgenden SERCOS-Zyklen wieder still stehen, was eine ruckende Bewegung zur Folge hat. Der Einsatz des Filters in P-0-0693 kann auch hier Abhilfe schaffen.



Bei Slave-Achsen mit Interpolation auf dem Antrieb wird empfohlen, die Filterzeitkonstante des P-0-0693 mit der 4-fachen SERCOS-Zykluszeit einzustellen. Ist die Filterzeitkonstante = "0", so wird der Antrieb bei Spannen im Stillstand die Warnung E2070 "Beschleunigungsbegrenzung aktiviert" ausgeben, da die im SERCOS-Takt übertragene additive Leitachspannung nicht innerhalb des Lageregeltaktes im Antrieb feininterpoliert wird.

Fehlercodes

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle **F_RELATED_TABLE**, 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#00000A00	16#00000001	Der Parameter P-0-0694 ist nicht im optional zyklischen Telegramm der Folgeachse parametrierbar
ACCESS_ERROR (16#0004)	16#00000A00	16#00000002	Der Parameter P-0-692 ist nicht im optional zyklischen Telegramm der Folgeachse parametrierbar

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000001	Die Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") ist < 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000002	Die Leitachsgeschwindigkeit im oberen Betriebspunkt ("MasterVelocityHigh") ist < 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000003	Die Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") und oberen Betriebspunkt ("MasterVelocityHigh") besitzen die gleichen Werte
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000004	Der Wert der Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") ist größer als der Wert im oberen Betriebspunkt ("MasterVelocityHigh")
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000005	Der Wert des Eingangs "LowLimit" ist kleiner als der minimal zulässige (-95.0)
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000006	Der Wert der Eingänge "PGainLowVel" oder "PGainHighVel" ist kleiner oder gleich 0
DEVICE_ERROR (16#0008)	16#00000A02	16#00000007	Bei dem in "Slave" angegebenen Antrieb ist keine Leistung zugeschaltet. Eine Zugspannungsregelung ist daher nicht möglich
DEVICE_ERROR (16#0008)	16#00000A02	16#00000008	Der in "Slave" angegebene Antrieb befindet sich nicht in einer synchronen Betriebsart
INPUT_INVALID_ERROR (16#0001)	16#00000A03	16#00000000	Ein SlavePreCtrl-Eingang hat einen ungültigen Wert
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000000	"Circumference" <= 0.0
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000001	"Distance" <= 0.0

Abb. 13-21: Fehlercodes Funktionsbaustein MB_TensionControlLoadCellType02

13.3 Funktionsbausteine zur Anwendung "Entkopplungsnetzwerk"

13.3.1 MB_TensionDecouplingNetworkType01

Kurzbeschreibung Ein Zugspannungsregler regelt abhängig vom Messwert einer Kraftmessdose die Zugspannung in einem Bahnabschnitt auf den gewünschten Sollwert (MB_TensionControlLoadCellType01/02). Über das Entkopplungsnetzwerk MB_TensionDecouplingNetworkType01 besteht nun die Möglichkeit, eine Stellsignaländerung eines beliebigen Bahnzugkraftreglers der kontinuierlichen Fertigungsanlage an alle, zur Entkopplung benötigten Achsen, weiterzuleiten. Durch diese entkoppelnden Vorsteuerungen wird eine temporäre und stationäre Änderung aller in der Maschine vorhandenen Bahnzugkräfte, ausgenommen der lokal zu regelnden, verhindert.

ML_TechTensionAdvanced.library

Zielsystem	Bibliothek
IndraMotion MLC 12VRS	ML_TechTensionAdvanced.compiled-library

Abb. 13-22: Referenztabelle Funktionsbaustein MB_TensionDecouplingNetworkType01



Die Verwendung des Funktionsbausteines unterliegt einer Lizenzpflicht. Details zu der Lizenzpflicht sind in der Einleitung der Dokumentation zu der Bibliothek beschrieben (siehe Seite 591).

Schnittstellenbeschreibung

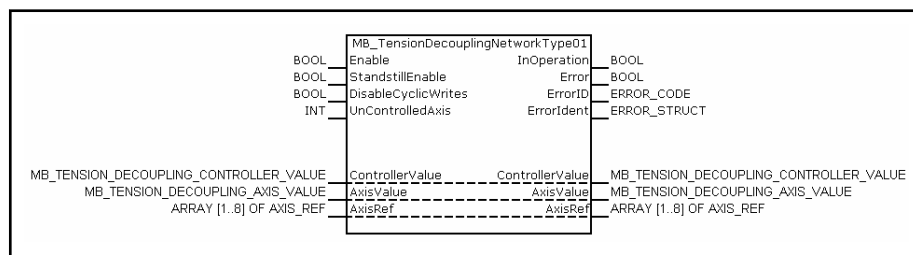


Abb. 13-23: Funktionsbaustein MB_TensionDecouplingNetworkType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	ControllerValue	MB_TENSION_DECOUPLING_CONTROLLER_VALUE	Struktur der Reglerausgänge (CorrectionValue, Adj-CorrectionValue, AdditivePosInc und AdditivePosDeg)
	AxisValue	MB_TENSION_DECOUPLING_AXIS_VALUE	Struktur der Achseingänge (CorrectionValue, AdditivePosInc und AdditivePosDeg)
	AxisRef	ARRAY [1..8] OF AXIS_REF	Stellachsen des Systems (ohne "UnControlledAxis")
VAR_INPUT	Enable	BOOL	Freigabe des Funktionsbausteins (einmalig, flankengesteuert)
	StandstillEnable	BOOL	Bei gesetztem Eingang ist die Stillstandsregelung aktiv
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	UnControlledAxis	INT	Achse im System, die nicht für die Regelung und Entkopplung der Bahnzugkräfte genutzt wird
VAR_OUTPUT	InOperation	BOOL	Das Entkopplungsnetzwerk arbeitet und die Ausgänge sind gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose

Abb. 13-24: Schnittstellenbeschreibung MB_TensionDecouplingNetworkType01

Strukturbeschreibung

In der folgenden Tabelle ist die Struktur des MB_TENSION_DECOUPLING_CONTROLLER_VALUE aufgelistet.

ML_TechTensionAdvanced.library

Name	Typ	Min.-Wert	Max.-Wert	Beschreibung
ControllerCorrectionValue [1]	REAL			CorrectionValue des ersten Bahnzugkraftreglers
ControllerCorrectionValue [2]	REAL			CorrectionValue des zweiten Bahnzugkraftreglers
ControllerCorrectionValue [3]	REAL			CorrectionValue des dritten Bahnzugkraftreglers
ControllerCorrectionValue [4]	REAL			CorrectionValue des vierten Bahnzugkraftreglers
ControllerCorrectionValue [5]	REAL			CorrectionValue des fünften Bahnzugkraftreglers
ControllerCorrectionValue [6]	REAL			CorrectionValue des sechsten Bahnzugkraftreglers
ControllerCorrectionValue [7]	REAL			CorrectionValue des siebten Bahnzugkraftreglers
ControllerCorrectionValue [8]	REAL			CorrectionValue des achten Bahnzugkraftreglers
ControllerAdjCorrectionValue [1]	REAL			AdjCorrectionValue des ersten Bahnzugkraftreglers
ControllerAdjCorrectionValue [2]	REAL			AdjCorrectionValue des zweiten Bahnzugkraftreglers
ControllerAdjCorrectionValue [3]	REAL			AdjCorrectionValue des dritten Bahnzugkraftreglers
ControllerAdjCorrectionValue [4]	REAL			AdjCorrectionValue des vierten Bahnzugkraftreglers
ControllerAdjCorrectionValue [5]	REAL			AdjCorrectionValue des fünften Bahnzugkraftreglers
ControllerAdjCorrectionValue [6]	REAL			AdjCorrectionValue des sechsten Bahnzugkraftreglers
ControllerAdjCorrectionValue [7]	REAL			AdjCorrectionValue des siebten Bahnzugkraftreglers
ControllerAdjCorrectionValue [8]	REAL			AdjCorrectionValue des achten Bahnzugkraftreglers
ControllerAdditivePosInc [1]	DINT			AdditivePosInc des ersten Bahnzugkraftreglers
ControllerAdditivePosInc [2]	DINT			AdditivePosInc des zweiten Bahnzugkraftreglers
ControllerAdditivePosInc [3]	DINT			AdditivePosInc des dritten Bahnzugkraftreglers
ControllerAdditivePosInc [4]	DINT			AdditivePosInc des vierten Bahnzugkraftreglers
ControllerAdditivePosInc [5]	DINT			AdditivePosInc des fünften Bahnzugkraftreglers
ControllerAdditivePosInc [6]	DINT			AdditivePosInc des sechsten Bahnzugkraftreglers
ControllerAdditivePosInc [7]	DINT			AdditivePosInc des siebten Bahnzugkraftreglers
ControllerAdditivePosInc [8]	DINT			AdditivePosInc des achten Bahnzugkraftreglers
ControllerAdditivePosDeg [1]	REAL			AdditivePosDeg des ersten Bahnzugkraftreglers
ControllerAdditivePosDeg [2]	REAL			AdditivePosDeg des zweiten Bahnzugkraftreglers
ControllerAdditivePosDeg [3]	REAL			AdditivePosDeg des dritten Bahnzugkraftreglers
ControllerAdditivePosDeg [4]	REAL			AdditivePosDeg des vierten Bahnzugkraftreglers
ControllerAdditivePosDeg [5]	REAL			AdditivePosDeg des fünften Bahnzugkraftreglers
ControllerAdditivePosDeg [6]	REAL			AdditivePosDeg des sechsten Bahnzugkraftreglers

ML_TechTensionAdvanced.library

Name	Typ	Min.-Wert	Max.-Wert	Beschreibung
ControllerAdditivePosDeg [7]	REAL			AdditivePosDeg des siebten Bahnzugkraftreglers
ControllerAdditivePosDeg [8]	REAL			AdditivePosDeg des achten Bahnzugkraftreglers

Abb. 13-25: Strukturbeschreibung MB_TENSION_DECOUPLING_CONTROL-
LER_VALUE

Strukturbeschreibung In der folgenden Tabelle ist die Struktur des MB_TENSION_DECOUPLING_AXIS_VALUE aufgelistet.

Name	Typ	Min.-Wert	Max.-Wert	Beschreibung
AxisCorrectionValue [1]	REAL			CorrectionValue der ersten Achse
AxisCorrectionValue [2]	REAL			CorrectionValue der zweiten Achse
AxisCorrectionValue [3]	REAL			CorrectionValue der dritten Achse
AxisCorrectionValue [4]	REAL			CorrectionValue der vierten Achse
AxisCorrectionValue [5]	REAL			CorrectionValue der fünften Achse
AxisCorrectionValue [6]	REAL			CorrectionValue der sechsten Achse
AxisCorrectionValue [7]	REAL			CorrectionValue der siebten Achse
AxisCorrectionValue [8]	REAL			CorrectionValue der achten Achse
AxisAdditivePosInc [1]	DINT			AdditivePosInc der ersten Achse
AxisAdditivePosInc [2]	DINT			AdditivePosInc der zweiten Achse
AxisAdditivePosInc [3]	DINT			AdditivePosInc der dritten Achse
AxisAdditivePosInc [4]	DINT			AdditivePosInc der vierten Achse
AxisAdditivePosInc [5]	DINT			AdditivePosInc der fünften Achse
AxisAdditivePosInc [6]	DINT			AdditivePosInc der sechsten Achse
AxisAdditivePosInc [7]	DINT			AdditivePosInc der siebten Achse
AxisAdditivePosInc [8]	DINT			AdditivePosInc der achten Achse
AxisAdditivePosDeg [1]	REAL			AdditivePosDeg der ersten Achse
AxisAdditivePosDeg [2]	REAL			AdditivePosDeg der zweiten Achse
AxisAdditivePosDeg [3]	REAL			AdditivePosDeg der dritten Achse
AxisAdditivePosDeg [4]	REAL			AdditivePosDeg der vierten Achse
AxisAdditivePosDeg [5]	REAL			AdditivePosDeg der fünften Achse
AxisAdditivePosDeg [6]	REAL			AdditivePosDeg der sechsten Achse
AxisAdditivePosDeg [7]	REAL			AdditivePosDeg der siebten Achse
AxisAdditivePosDeg [8]	REAL			AdditivePosDeg der achten Achse

Abb. 13-26: Strukturbeschreibung MB_TENSION_DECOUPLING_AXIS_VALUE

Min-/Max- und Defaultwerte In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
ControllerValue	MB_TENSION_DECOUPLING_CONTROLLER_VALUE			0	kontinuierlich
AxisValue	MB_TENSION_DECOUPLING_AXIS_VALUE			0	kontinuierlich
AxisRef	ARRAY [1..8] OF AXIS_REF			0	steigende Flanke an "Enable"
Enable	BOOL			FALSE	kontinuierlich
StandstillEnable	BOOL			FALSE	kontinuierlich
DisableCyclicWrites	BOOL			FALSE	steigende Flanke an "Enable"
UnControlledAxis	INT	1	9	1	steigende Flanke an "Enable"

Abb. 13-27: Eingangsverhalten MB_TensionDecouplingNetworkType01

Funktionsbeschreibung

Im Folgenden werden die einzelnen Funktionen der Eingänge des Funktionsbausteins näher erläutert. Allgemein gilt für alle Eingänge, dass sie kontinuierlich bei gesetztem "Enable"-Eingang ausgewertet werden. D.h. Änderungen der Eingänge werden sofort wirksam ohne erneutes Toggeln dieses Eingangs. Ausnahme dieser Regel bilden die Eingänge "UnControlledAxis", "DisableCyclicWrites" und das Ein- Ausgangsarray "AxisRef" die nur bei steigender Flanke des "Enable"-Eingangs ausgewertet werden. Alle Eingabewerte bekommen intern den Wert "0" bzw. FALSE zugewiesen, wenn deren Eingänge nicht angeschlossen werden. Ausnahme dieser Regel ist der Eingang "UnControlledAxis", dieser wird bei nicht Belegung auf den Wert 1 gesetzt. Wie schon beim Bahnzugkraftregler MB_Tension_ControlLoadCellType02 wird das Stellsignal im normalen Betrieb auf den Getriebefeinabgleich geschrieben. Je nach Achstyp wird hierbei der entsprechende Parameter (P-0-0694 bei Interpolation auf dem Antrieb bzw. A-0-2605 bei Interpolation auf der Steuerung), falls "DisableCyclicWrites" inaktiv ist, direkt beschrieben. Außerdem stehen die Stellsignale aller Achsen immer über die Struktur "MB_TENSION_DECOUPLING_AXIS_VALUE" am Ausgang des Entkopplungsnetzwerks zur Verfügung. Liegen einheitliche Achstypen zugrunde, ist es ausreichend, lediglich das entsprechende Stillstandsstellsignal (AdditivePosInc bei Interpolation im Antrieb oder AdditivePosDeg bei Interpolation in der Steuerung) von den Reglern auf das Entkopplungsnetzwerk aufzuschalten. Liegen gemischte Achstypen (Interpolation auf dem Antrieb und Interpolation in der Steuerung) zugrunde, ist es zwingend notwendig, bei allen Reglern beide Stillstandsstellsignale (AdditivePosInc und AdditivePosDeg) auf des Entkopplungsnetzwerk aufzuschalten. Da die Ausgänge "CorrectionValue", "AdjCorrectionValue", "AdditivePosInc" und "AdditivePosDeg" der Bahnspannungsregler bei der Nutzung des Entkopplungsnetzwerk nicht mehr direkt auf die Achsen geschrieben werden dürfen, ist es zwingend notwendig in allen beteiligten Bahnzugkraftreglern das zyklische Schreiben der Werte aufzuheben (DisableCyclicWrites:=TRUE). Die gesamte Wirkungsweise des Entkopplungsnetzwerks ist in den folgenden Diagrammen dargestellt.

ML_TechTensionAdvanced.library

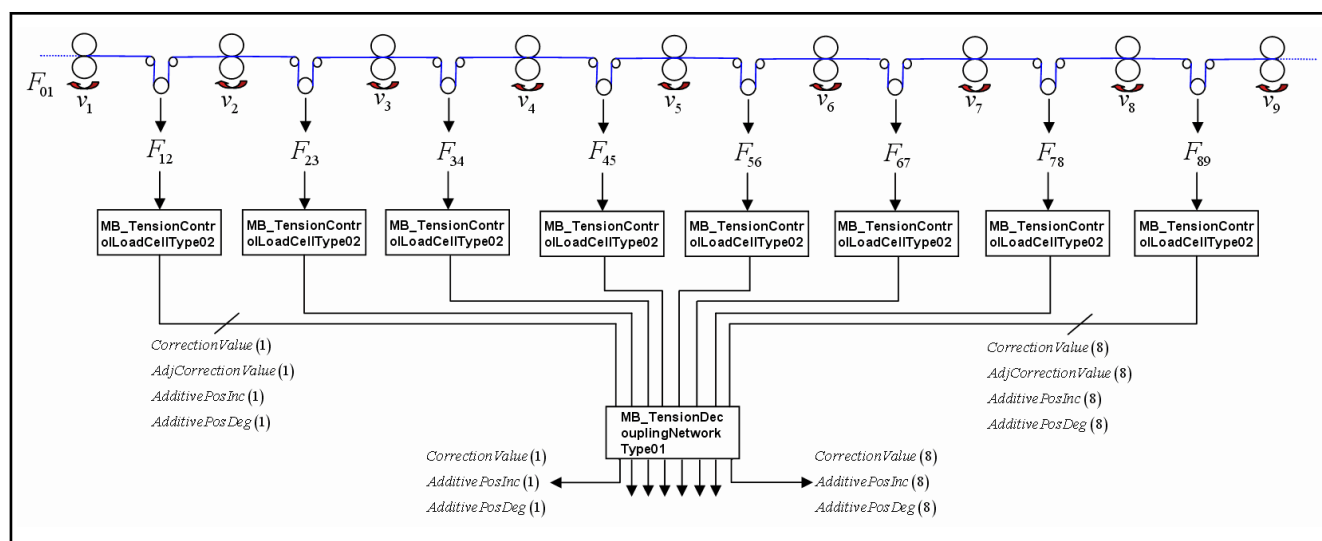


Abb. 13-28: Allgemeines Diagramm des Funktionsbausteins MB_TensionDecouplingNetworkType01

Es ist ersichtlich, dass jeder zu regelnde Bahnabschnitt mit einem Bahnzugkraftregler MB_TensionControlLoadCellType02 ausgestattet werden muss. Hierbei ist zu beachten, dass alle Bahnzugregler im positiven Sinn arbeiten müssen (NegPolarity:=FALSE). Die Reglerausgänge "CorrectionValue", "AdjCorrectionValue", "AdditivePosInc" und "AdditivePosDeg" werden dann dem Entkopplungsnetzwerk MB_TensionDecouplingNetworkType01 zugeführt. In dem dargestellten Beispiel soll in acht Bahnabschnitten die Bahnzugkraft unabhängig voneinander geregelt werden. Somit sind mindestens acht Stellglieder nötig. Da n zu regelnde Bahnabschnitte immer durch $n+1$ Achsen begrenzt werden, kann eine beliebige Achse gewählt werden, die ausschließlich mit Masterachsgeschwindigkeit betrieben wird und keine zusätzlichen Stellbewegungen ("CorrectionValue", "AdditivePosInc" oder "AdditivePosDeg") ausführen muss. In nachfolgender Darstellung wurde die erste Achse des Systems für die entkoppelte Regelung der Bahnzugkräfte ausgeschlossen (UnControlledAxis:=1). Diese Entkopplungsvariante entspricht zugleich der Grundfunktion des MB_TensionControlLoadCellType02, wenn nur der erste Bahnzugkraftregler aktiv ist.

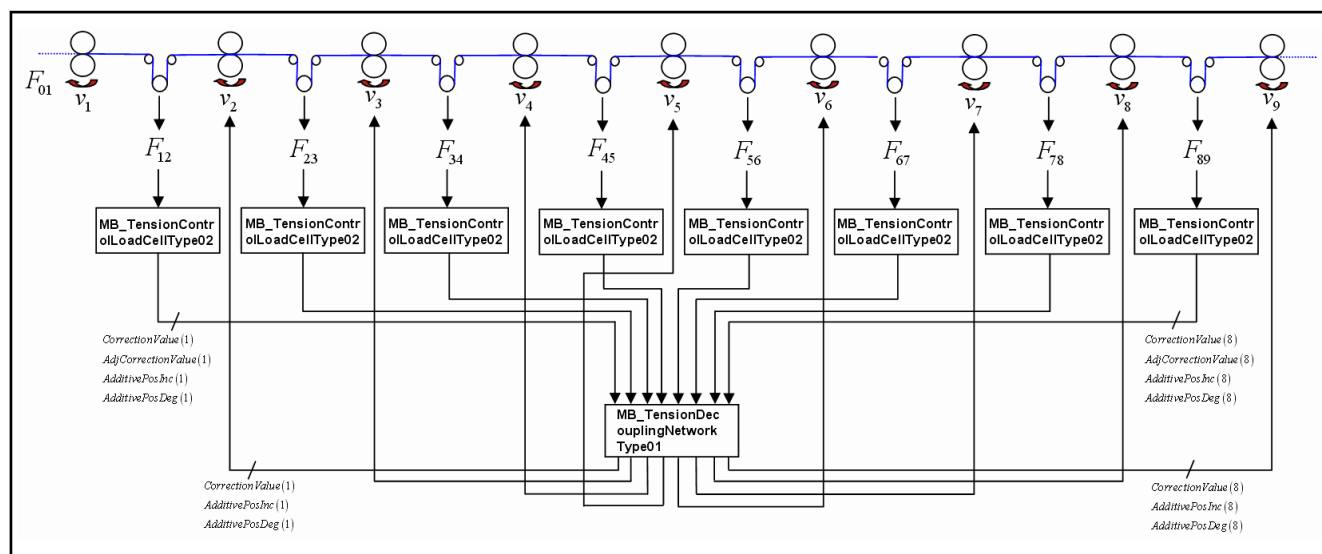


Abb. 13-29: Entkopplung der Bahnzugkräfte durch Vorsteuerung ausschließlich in Bahnlauftrichtung (führende Klemmstellen werden nicht für die Regelung und Entkopplung benötigt)

Wird diese Variante für ein System mit weniger als acht zu regelnden Bahnabschnitten gewählt, so können einfach die hinteren in Bahnlaufrichtung befindlichen Bahnzugkraftregler unberücksichtigt bleiben.

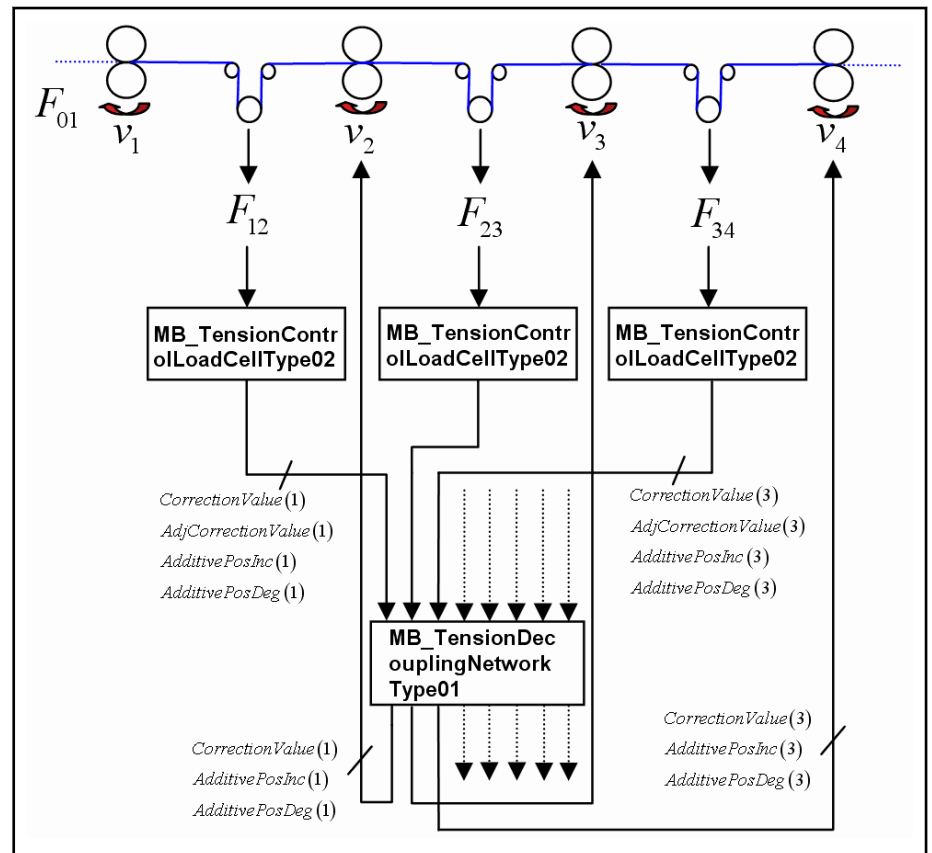


Abb. 13-30: Entkopplung der Bahnzugkräfte durch Vorsteuerung ausschließlich in Bahnlaufrichtung (Darstellung bei drei zu regelnden Bahnabschnitten)

Eine weitere Möglichkeit ist nachfolgend dargestellt. Hierbei wird die letzte Achse des Systems für die Regelung der Bahnzugkräfte ausgeschlossen (UnControlledAxis:=9).

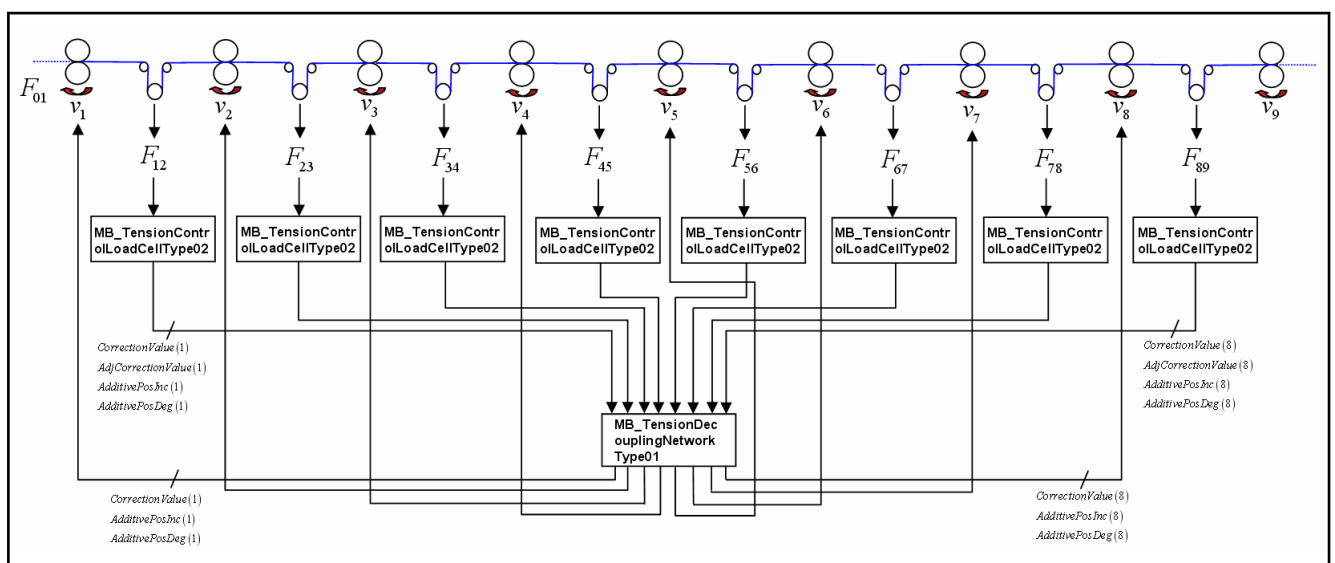


Abb. 13-31: Entkopplung der Bahnzugkräfte durch Vorsteuerung ausschließlich gegen die Bahnlaufrichtung (nachfolgende Klemmstellen werden nicht für die Regelung und Entkopplung benötigt)

ML_TechTensionAdvanced.library

Wird diese Variante für ein System mit weniger als acht zu regelnden Bahnabschnitten gewählt, so können einfach die führenden gegen die Bahnlaufrichtung befindlichen Bahnzugkraftregler unberücksichtigt bleiben.

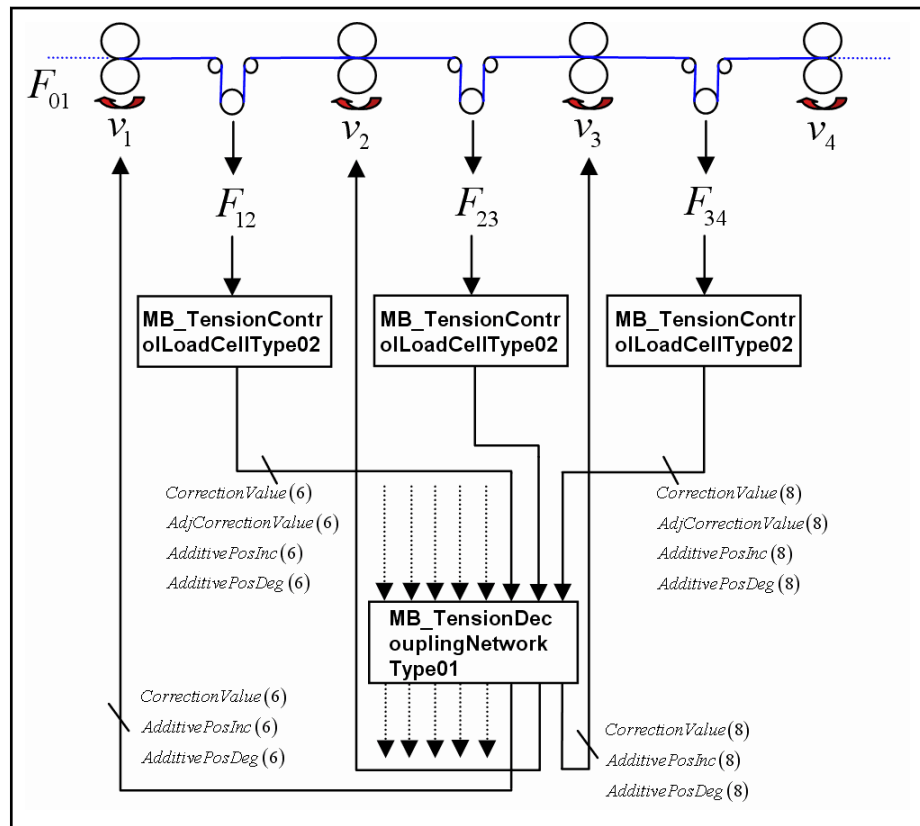


Abb. 13-32: Entkopplung der Bahnzugkräfte durch Vorsteuerung ausschließlich gegen die Bahnlaufrichtung (Darstellung bei drei zu regelnden Bahnabschnitten)

Da jede beliebige Achse des Systems für die Regelung der Bahnzugkräfte ausgeschlossen werden kann, besteht weiter die Möglichkeit, eine mittlere Achse auszuschließen. Dies entspricht den Varianten "UnControlledAxis":=2-8 und ist beispielhaft für die fünfte Achse (UnControlledAxis:=5) dargestellt.

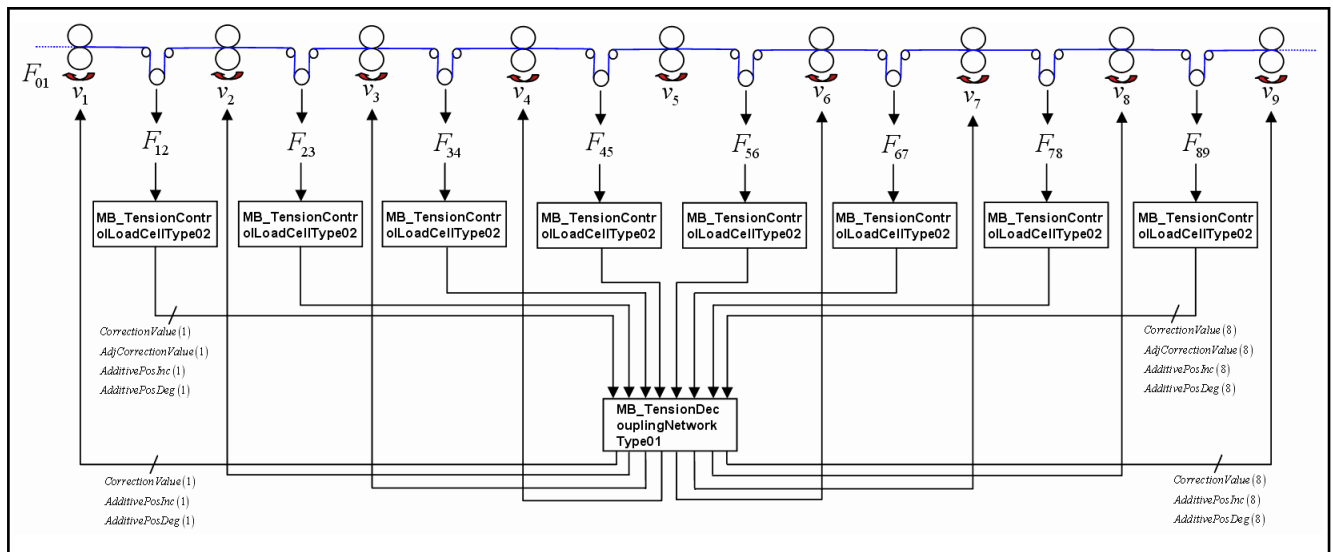


Abb. 13-33: Entkopplung der Bahnzugkräfte durch Vorsteuerung in und gegen die Bahnaufrichtung (eine mittlere Klemmstellen, in Abb. "v5", wird nicht für die Regelung und Entkopplung benötigt)

Wird diese Variante für ein System mit weniger als acht zu regelnden Bahnabschnitten gewählt, so können entsprechend den beiden vorhergehende Varianten einfach die führenden und/oder die in Bahnaufrichtung folgenden Bahnzugkraftregler unberücksichtigt bleiben. Wählt man zweckmäßig die nicht zu regelnde Achse entsprechend des realen Systems in Bahnaufrichtung, so ist es lediglich erforderlich, die folgenden real nicht vorhandenen Bahnzugkraftregler unberücksichtigt zu lassen.

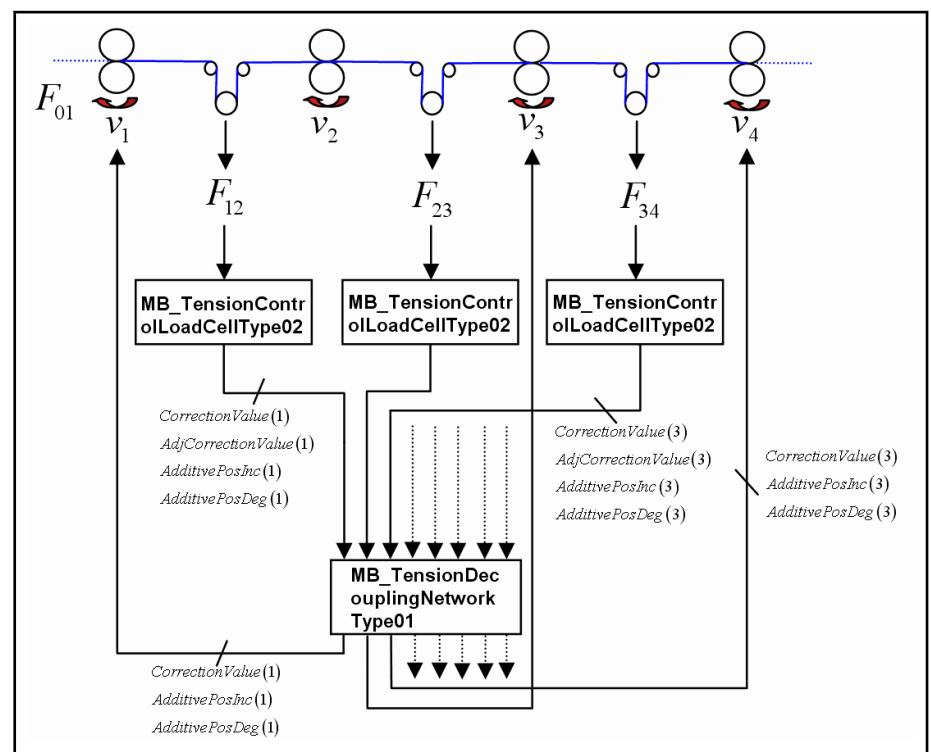


Abb. 13-34: Entkopplung der Bahnzugkräfte durch Vorsteuerung in und gegen die Bahnaufrichtung (Darstellung bei drei zu regelnden Bahnabschnitten)

Neben diesen drei Varianten der Entkopplung, bei denen jeweils eine Achse nicht zur Regelung und Entkopplung benötigt wird, kann das Entkopplungs-

ML_TechTensionAdvanced.library

netzwerk weiter nahezu beliebig eingesetzt werden. Beispielhaft sei der Fall einer Anlage mit bis zu 15 zu regelnden Bahnabschnitten und der Forderung nach einem minimalen Stellaufwand erwähnt.

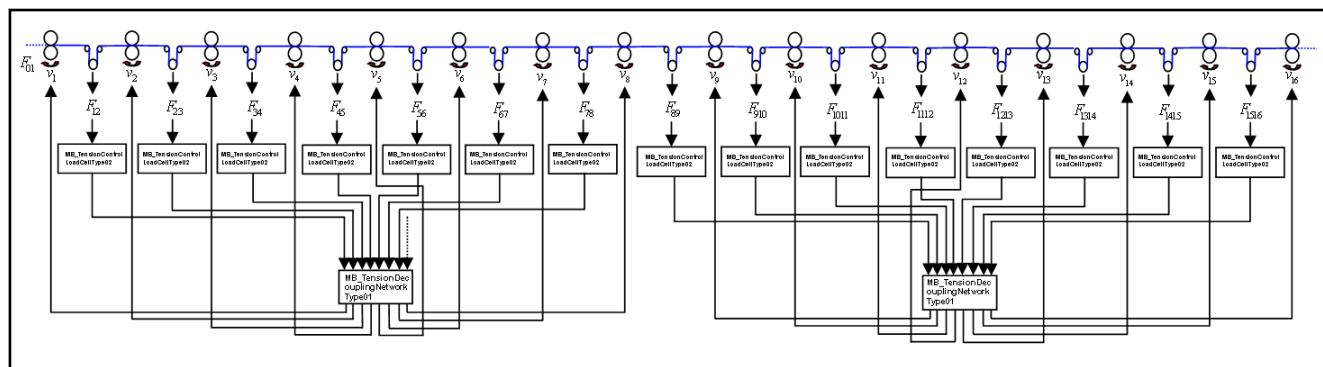


Abb. 13-35: Expansion des Entkopplungsnetzwerks auf 15 zu regelnde Bahnabschnitte bei zusätzlicher Forderung nach minimalem Stellaufwand

Es ist in dieser Darstellung zu erkennen, dass zwei Funktionsbausteine MB_TensionDecouplingNetworkType01 eingesetzt werden, und eine Stellsignaländerung eines Bahnzugkraftreglers maximal acht ((geregelter Abschnitte +1)/2) Stellbewegungen verursacht. Weiter ist zu erkennen, dass in dem führenden Entkopplungsnetzwerk eine Upstream-Entkopplung (UnControlledAxis:=9) und dem hinteren Entkopplungsnetzwerk eine Downstream-Entkopplung (UnControlledAxis:=1) eingesetzt wird. Lediglich der letzte Bahnzugkraftregler des führenden Entkopplungsnetzwerks wurde freigelassen. Da der letzte Bahnabschnitt des führenden Entkopplungsnetzwerks komplett entkoppelt von allen davorliegenden Bahnabschnitten ist, kann diese Kombination der beiden Entkopplungsnetzwerke eingesetzt werden, ohne dass eine Querkommunikation bzw. Quervorsteuerung benötigt wird. Setzt man zwei oder mehr Entkopplungsnetzwerke ein, so kann man diese in zahllosen Varianten miteinander kombinieren. Ist für eine spezielle Anlage eine weitere Variante mit mehreren Entkopplungsnetzwerken zweckmäßig, so können die Ein- und Ausgänge der Entkopplungsnetzwerke extern kombiniert werden. Je nach gewünschtem Verhalten kann es hierbei außerdem nötig sein, das zyklische Schreiben (DisableCyclicWrites:=TRUE) zu unterbinden. Beispielhaft für diese zahllosen Varianten sind Anlagen mit mehr als acht zu regelnden Bahnabschnitten und der Forderung, dass die erste, letzte oder eine beliebige Achse nicht geregelt und/oder vorgesteuert werden darf. Hierbei ist es dann sinnvoll, ein oder mehrere Ausgänge eines Entkopplungsnetzwerks mit einem oder mehreren Eingängen eines anderen Entkopplungsnetzwerks zu kombinieren. Diese Umsetzungen setzen jedoch detaillierte Kenntnisse voraus.

Inbetriebnahme Ist "DisableCyclicWrites" inaktiv, müssen die beiden Parameter P-0-0694 (Getriebefeinabgleich) und P-0-0692 (additive Leitachspolition) im zyklischen Sollwertkanal der Folgeachse aufgelegt werden. Bei steigender Flanke an "Enable" werden zyklische Datenkanäle überprüft. Wird einer der beiden Parameter nicht gefunden, erfolgt eine entsprechende Fehlermeldung an den Fehlerausgängen des Funktionsbausteins. Ist "DisableCyclicWrites" aktiv, wird die Konfiguration der zyklischen Sollwertdatenkanäle nicht überprüft.

Enable Der Eingang „Enable“ aktiviert den Funktionsbaustein.

StandstillEnable Durch Setzen des Eingangs „StandstillEnable“ wird die Zugspannungsregelung im Stillstand aktiviert. Somit werden die Eingänge „AdditivePosInc“ und „AdditivePosDeg“ ausgewertet und über diese die gesamten Stellsignale "AdditivePosInc" und "AdditivePosDeg" der einzelnen Achsen berechnet und ausgegeben.

DisableCyclicWrites Der Eingang "DisableCyclicWrites" bestimmt, ob die vom Funktionsbaustein berechneten Werte direkt über den zyklischen Kanal geschrieben werden (FALSE) oder nur am Baustein Ausgang verfügbar sind (TRUE). Mit "DisableCyclicWrites":=TRUE ist es somit möglich, mehrere Entkopplungsnetzwerke miteinander zu kombinieren, indem man die Ausgänge mehrerer Entkopplungswerke manuell verknüpft.

UnControlledAxis Der Eingang "UnControlledAxis" bestimmt die Entkopplungsvariante sowie die Position der nicht zu regelnden und vorzusteuern Achse. Standardmäßig wird dieser Eingang auf 1 gesetzt, was für das Entkopplungsnetzwerk bedeutet, dass ausschließlich in Bahnaufrichtung vorgesteuert werden soll (Downstream-Entkopplung). Wird dieser Eingang auf 9 gesetzt, so wird die letzte Achse des Systems für die Regelung und Entkopplung ausgeschlossen (Upstream-Entkopplung). Eine Vorsteuerung wird in diesem Falle ausschließlich gegen die Bahnaufrichtung erfolgen. Des Weiteren kann eine beliebige Achse des Systems für die Bahnzugkraftregelung und Entkopplung ausgeschlossen werden (UnControlledAxis:=2-8).

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle **F_RELATED_TABLE**, 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR, 16#0004	16#00000A00	16#00000001	Der Parameter P-0-0694 ist nicht im optional zyklischen Telegramm der vorzusteuern Achse parametrier
ACCESS_ERROR, 16#0004	16#00000A00	16#00000002	Der Parameter P-0-0692 ist nicht im optional zyklischen Telegramm der vorzusteuern Achse parametrier
INPUT_RANGE_ERROR, 16#0006	16#00000A01	16#00000007	Der Eingang „UnControlledAxis“ ist außerhalb des gültigen Bereichs
DEVICE_ERROR, 16#0008	16#00000A02	16#00000007	Bei einem Antrieb ist keine Leistung zugeschaltet. Eine Zugspannungsregelung ist daher nicht möglich
DEVICE_ERROR, 16#0008	16#00000A02	16#00000008	Ein Antrieb befindet sich nicht in einer synchronen Betriebsart

Abb. 13-36: Fehlercodes von MB_TensionDecouplingNetworkType01

14 Service und Support

Unser Kundendienst-Helpdesk im Hauptwerk Lohr am Main und der weltweite Service stehen Ihnen mit Rat und Tat zur Seite. Sie erreichen uns täglich **rund um die Uhr - auch am Wochenende und an Feiertagen.**

	Helpdesk	Service-Hotline Welt
Telefon	+49 (0) 9352 40 50 60	Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit unserem für Sie nächstgelegenen Ansprechpartner auf. Die Hotline-Rufnummern entnehmen Sie bitte den Vertriebsadressen im Internet.
Fax	+49 (0) 9352 40 49 41	
E-Mail	service.svc@boschrexroth.de	
Internet	http://www.boschrexroth.com Hier finden Sie auch ergänzende Hinweise zu Service, Reparatur (z.B. Anlieferadressen) und Training.	

Vorbereitung der Informationen

Wir können Ihnen schnell und effizient helfen, wenn Sie folgende Informationen bereithalten:

- detaillierte Beschreibung der Störung und der Umstände
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern
- Telefon-, Faxnummern und E-Mail-Adresse, unter denen Sie für Rückfragen zu erreichen sind

Index

A

Abwickler.....	471, 477, 489, 494
Analogkanal.....	426
Antriebsparameter.....	141, 144
AntriebsparameterEinstellungen.....	269
Antriebssystem.....	19
Aufwickler.....	471, 477, 489, 494

B

Bahnspannung.....	436
Bahnweg.....	436
Bausteine zur allgemeinen Verwendung.....	210
Beschleunigungsmoment.....	474, 491
Beschleunigungsmomentkompensation... ..	474, 491
Besonderheiten für IndraMotion MLC	
MB_DancerControlType03	434
MB_DiameterCalculatorType03	443
MB_DiameterMeasurementType01	449
MB_WinderTensionCtrlType01	476
MB_WindSpeedControlAdaptionType01	512
MB_WindTaperProfileType01	508
Besonderheiten für IndraMotion MLD	
MB_DiameterCalculatorType03	443
MB_DiameterMeasurementType01	449
MB_WinderTensionCtrlType01	476
Besonderheiten von IndraMotion MLD.....	31
bestimmungsgemäßer Gebrauch.....	17
Einsatz- und Anwendungsbereiche	17
Stand der Technik	17
Bezeichnungen und Abkürzungen.....	15
Bibliothek	
ML_TechBase	33
ML_TechCrank	387
ML_TechInterface.lib	479, 496
ML_TechMotion	165
RIL_ParameterChannel.library	567
RMB_TechCam	545
RMB_TechCrossCutCrossSeal.library	515
Bibliotheken	
ML_TechRegi.library/MX_TechRegi.lib	297
ML_TechTensionAdvanced.library	591
RMB_TechWinder/MX_TechWinder	425

C

CamLock.....	196
Applikationsbeispiel	197
CamLock Funktionsbaustein	
MB_CamLock	202
CamLock Funktionsbausteine	
Ablaufbeispiele	206
MB_PrepareCams	200
Common - Application FBs.....	515
CrossCutter - Calculation FB	
MB_CrossCutterCalcType04	530

C

CurrentDancerDeviation.....	432

D

Datenstruktur	
MB_AXIS_SYNCHRONISATION	479, 496
MB_WIND_DANCER_LIMITS	436
Defaultwerte	
MB_CalcInertiaLimitType02	499
MB_CAM_TABLE_DATA	540
MB_DancerControlType03	431
MB_DiameterCalculatorType03	440
MB_DiameterMeasurementType01	448
MB_PUSH_OUT_CONFIG	543
MB_UnwindMaterialType02	502
MB_WinderDancerCtrlType01	453
MB_WinderTensionCtrlType02	465
Drehmoment für Wickelantrieb	
Durchmesserberechnung	472, 489
Durchmesserberechnung.....	436
Eingänge zur Manipulation der Durch-	
messerberechnung festlegen	444
Inbetriebnahme	444
Mittelwertbildung	441
Preset des Durchmessers	440
Proportionalität bei der Durchmesserbe-	
rechnung	441
Rollenwechsel	442
Stoffbahngrößen festlegen	444
Trägheitsmoment berechnen	442
Wickelgrößen festlegen	444
Wicklerachse kommandieren	445
zyklisch veränderliche Eingänge festlegen	444
Durchmesser extern messen.....	445
Durchmesserrechner	
MB_WinderDancerCtrlType01	455

E

E/A-Belegung	
MB_CalcInertiaLimitType02	498
MB_UnwindMaterialType02	501
Einführung in die SmartBelt-Verwendung.....	263
Elektrisches Antriebssystem.....	19
Entkopplungsnetzwerk.....	607
Erfassung eines extern gemessenen Durch-	
messers.....	445

F

Fehlercodes	
IL_ParameterChannel	573
IL_ParameterChannel, Parameterkanal	574
IL_TempControlType01	77
MB_AllocCyclicParameter	130
MB_CalcInertiaLimitType02	499

Index

F

...Fehlercodes

MB_ContinuousAdjustType01	42
MB_ContinuousAdjustType02	45
MB_CrossCutSealType01	530
MB_DancerControlType03	433
MB_DiameterCalculatorType03	442
MB_DiameterMeasurementType01	448
MB_FreeCyclicParameter	132
MB_GetCyclicParameterHandle	135
MB_GetLastCyclicParameterError	139
MB_GetSystemInfo	160
MB_IncrementalAdjustType01	48
MB_RegiMeasuringType01	344
MB_RegiMeasuringType02	348
MB_RegiMeasuringType03	352
MB_RegiRegulateType01	356
MB_RegiRegulateType02	365
MB_RegiRegulateType03	372
MB_RegiRegulateType04	380
MB_RegiSetpointLockType01	382
MB_RegisterControllerSideType01	336
MB_RegisterControllerType02	308
MB_RegisterControllerType04	321
MB_TensionControlLoadCellType02	607
MB_UnwindMaterialType02	504
MB_WinderDancerCtrlType01	457
MB_WinderTensionCtrlSpeedType01	493
MB_WinderTensionCtrlType01	475
MB_WindSpeedControlAdaptionType01	511
MB_WindTaperProfileType01	508
MC_DigitalCamSwitch	71
ML_InterpolationLinear	53
ML_SagControlA	195
ML_SagControlC	191

Feinabgleich

resultierender	440, 441

Filter

Beschleunigungsmoment	480, 496
Momentistwert	474, 480, 496
Zugspannungsiswert	492

Firmware, notwendig

MB_CalcInertialLimitType02	500
MB_DancerControlType03	434
MB_DiameterCalculatorType03	443
MB_DiameterMeasurementType01	449
MB_UnwindMaterialType02	504
MB_WinderDancerCtrlType01	458
MB_WinderTensionCtrlType01	475
MB_WindSpeedControlAdaptionType01	512
MB_WindTaperProfileType01	508
ML_FlyingShear	173
ML_SagControlA	195
ML_SagControlC	191

Fliegender Rollenwechsel..... 474, 492

Fliegende Säge..... 166

Funktion..... 136

F

...Funktion

MB_GetLastCyclicParameterError	138
MB_GetModuloType01	152
MB_GetModuloType02	153
MB_GetModuloType03	154
MB_GetModuloType04	155
MB_MasterToPhi	401
MB_ReadCyclicParameter	135
MB_ReadCyclicRealParameter	136
MB_WriteCyclicRealParameter	137
MB_YVirtToWorkRange	424
ML_DegToInc	122
ML_DegToRad	123
ML_IncToDeg	123
ML_IncToRad	124
ML_RadToDeg	125
ML_RadToInc	125

Funktionsbaustein

IL_LinkedList	147
IL_ParameterChannel	567
IL_RegiQualityType01	383
IL_TempControlType01	71
MB_AbortTrigger	118
MB_AllocCyclicParameter	128
MB_AxisDistanceAccumulator	211
MB_BellCrankData	406
MB_CalcInertialLimitType02	497
MB_CamLock	202
MB_CamTableBellCrank	408
MB_CamTableBellCrankSuperimposed	412
MB_CamTableCrank	393
MB_CamTableCrankSuperimposed	396
MB_ContinuousAdjustType01	38
MB_ContinuousAdjustType02	42
MB_CoordinatePairCamType01	558
MB_CrossCutSealType01	515
MB_CrossCutterCalcType04	530
MB_DancerControlType03	428
MB_DiameterCalculatorType03	436
MB_DistanceAccumulator	210
MB_FeedMovementCalcType01	545
MB_FeedMovementCalcType02	551
MB_FreeCyclicParameter	130
MB_GetCyclicChannelConfig	132
MB_GetCyclicParameterHandle	133
MB_GetSystemInfo	157
MB_IncrementalAdjustType01	46
MB_InitTouchProbe	102
MB_LoadCamData	160
MB_MagicBeltType01	223
MB_PhiToXvirt	399
MB_PhiToYvirt	416
MB_PilgrimStepCalcType01	280
MB_PositionMonitoring_GantryCant	216
MB_PositionMonitoring_PosCollision	218
MB_PrepareCams	200

F

...Funktionsbaustein	
MB_PrintLengthCorrCalcType01	290
MB_RegiMeasuringType01	342
MB_RegiMeasuringType02	344
MB_RegiMeasuringType03	348
MB_RegiRegulateType01	352
MB_RegiRegulateType02	356
MB_RegiRegulateType03	366
MB_RegiRegulateType04	372
MB_RegisterControllerSideType01	329
MB_RegisterControllerType02	298
MB_RegisterControllerType04	308
MB_SetpointLockType01	381
MB_SmartBeltType01	256
MB_TensionControlLoadCellType01	175
MB_TensionControlLoadCellType02	591
MB_TouchProbe	109
MB_TouchProbeContinuous	112
MB_UnwindMaterialType02	497, 500
MB_WinderDancerCtrlType01	450
MB_WinderTensionCtrlSpeedType01	481
MB_WinderTensionCtrlType02	462
MB_WindSpeedControlAdaptionType01	497, 508
MB_WindTaperProfileType01	497, 505
MB_WriteExpectWindow	115
MB_XvirtToXmech	401
MB_YvirtToPhi	418
MB_YvirtToPhiPoly5	420
MB_YvirtToYmech	422
MC_DigitalCamSwitch	56
ML_AxisBackup	141
ML_AxisRestore	144
ML_InterpolationLinear	51
ML_MaxValue	53
ML_SagControlA	192
ML_SagControlC	188
ML_TwoPosCtrlType01	49
MX(L)_FlyingShear	166
Tänzerlageregelung	426
Funktionsbausteine zum Sichern und Wiederherstellen von Antriebsparametern.....	141
Funktionsbausteine zum Vertippen von Variablen	
Berechnung der Änderungsgeschwindigkeit	36
Einführung und Übersicht	35
Grundregeln	36
Parametrierung	48
Preset	36
Stellgrenzen	36
Funktionsbausteine zur Anwendung	
Fliegende Säge	166
Nockenschaltwerk	54
Positionsüberwachung	216
Schlaufenregelung	187
Vertippen von Variablen	35

F

Funktionsbaustein MB_CamTableCrank	
Applikationshinweise	395
Datenstruktur	394
Randbedingungen	395
Funktionsbaustein zum Vertippen von Variablen	
Komponenten	48
Funktionsbaustein zur Anwendung	
Durchmesserberechnung	436
Durchmesser extern messen	445
Tänzerregler	426
Wickelrechner mit Kraftmessdose	460
Wickelrechner mit Zugkraftregelung	460
Wickelrechner mit Zugkraftsteuerung	460
Wickelrechner sensorlos	460
Funktionsdiagramm	
Tänzerlageregler	427
Wickelrechner mit Kraftmessdose	471
Wickelrechner mit Zugkraftregelung	471
Wickelrechner sensorlos	469

G

Getriebeeinstellung - Berechnung.....	536
Getriebefeinabgleich.....	436, 437, 475, 492
Getriebeübersetzung.....	437
Gliederung der Dokumentation.....	9

H

Hardware, notwendig	
MB_CalcInertiaLimitType02	499
MB_DancerControlType03	434
MB_DiameterCalculatorType03	443
MB_DiameterMeasurementType01	449
MB_UnwindMaterialType02	504
MB_WinderDancerCtrlType01	458
MB_WinderTensionCtrlType01	475
MB_WindSpeedControlAdaptionType01	511
MB_WindTaperProfileType01	508
ML_FlyingShear	173
ML_SagControlA	195
ML_SagControlC	191
Hilfsfunktionen.....	157
Hülsendurchmesser.....	499

I

IL_LinkedList.....	147
IL_ParameterChannel.....	567
Fehlerbehandlung	572
Fehlertelegammme	579
Inbetriebnahme des Bausteins	582
Index und Subindex	576
Initialisierung der Kommunikation	579
Parameterkanal im zyklischen Datenkanal	574
Steuer-/Statuswort	575
IL_ParameterChannel – Protokoll, Fehlerbehandlung und Inbetriebnahme.....	574

Index

I

IL_RegiQualityType01.....	383
IL_TempControlType01.....	71
Abtastzeit	83
Identifikation	84
Regelkreisüberwachung	81
Regler	79
Regleralgorithmus	80
Reglerstruktur	80
Taskintervallzeit	82
Verwendung	81
Inbetriebnahme	
Durchmesserrechner	444
Tänzerlagereger	435
Wickelrechner mit Kraftmessdose	476
Wickelrechner mit Tänzer	459
Wickelrechner mit Zugkraftregelung	476
Wickelrechner mit Zugkraftsteuerung	476
Wickelrechner sensorlos	476
IndraLogic-Schritte, notwendig	
ML_FlyingShear	175
IndraWorks.....	173, 434, 476, 508, 512

K

Kniehebelkinematik.....	402
Funktionsbausteine	402
Kompensation	
Beschleunigungsmoment	474, 491
Reibmoment	473, 490
Konfiguration des Seitenregisters.....	336
Kurbelkinematik	
Anwendungsfall 1: Einzelachsbetrieb	389
Anwendungsfall 2: Synchronbetrieb mit	
Kurvenscheibe	389
Funktion MB_MasterToPhi	401
Funktionsbaustein MB_CamTableCrank	393
Funktionsbaustein MB_PhiToXvirt	399
MB_CamTableCrankSuperimposed	396
MB_XvirtToXmech	401
Mechanische Anordnung 1	390
Mechanische Anordnung 2	391
mechanische translatorische Position	392
Übersicht	388
Variable und Begriffe	392
Verfahrenbereichsgleichung	392
Xmech	392
Xvirt	392
Zählrichtung	392

L

Logik.....	147

M

MagicBelt.....	221
Materialgrößen berechnen.....	503

M

Materialrechner	
Mittelwertbildung der Materialstärke	503
Preset der Materialgrößen	503
Reset	503
Materialzähler.....	500
Mathematische Funktionen.....	51
MB_AbortTrigger.....	118
MB_AllocCyclicParameter.....	128
MB_AxisDistanceAccumulator.....	211
MB_BellCrankData.....	406
MB_CalcInertiaLimitType02.....	497
MB_CAM_TABLE_DATA.....	539
Verschaltungsplan	541
Zuordnung von CamTableID	540
MB_CamLock.....	202
MB_CamLock Parametrierung.....	209
MB_CAMPROF.....	163
MB_CAMTABLE.....	163
MB_CamTableBellCrank.....	408
MB_CamTableBellCrankSuperimposed.....	412
MB_CamTableCrank.....	393
MB_CamTableCrankSuperimposed.....	396
Anwendungsbeispiel	399
MB_ContinuousAdjustType01.....	38
MB_ContinuousAdjustType02.....	42
MB_CoordinatePairCamType01.....	558
MB_CORR_PROFILE.....	541
Elemente	542
MB_CrossCutSealType01.....	515
MB_CrossCutterCalcType04.....	530
MB_PUSH_OUT_CONFIG	543
MB_RESOLUTION	542
MB_CrossCutterCalcType04, Besonderheiten der IndraMotion MLC.....	538
MB_CrossSealType01	
Querschneider-Anwendungsfall	525
Quersiegler-Anwendungsfall	522
MB_CYCLIC_CHANNEL_CONFIG.....	140
MB_CYCLIC_CHANNELS.....	140
MB_CYCLIC_PARAM_REF.....	139
MB_CYCLIC_TYPES.....	139
MB_DancerControlType03.....	428
MB_DiameterCalculatorType03.....	436
MB_DiameterMeasurementType01.....	445
MB_DistanceAccumulator.....	210
MB_FeedMovementCalcType01.....	545
MB_FeedMovementCalcType02.....	551
MB_FreeCyclicParameter.....	130
MB_GetCyclicChannelConfig.....	132
MB_GetCyclicParameterHandle.....	133
MB_GetLastCyclicParameterError.....	138
MB_GetModuloType01.....	152
MB_GetModuloType02.....	153
MB_GetModuloType03.....	154
MB_GetModuloType04.....	155
MB_GetSystemInfo.....	157

M

MB_IncrementalAdjustType01.....	46
MB_InitTouchProbe.....	102
MB_LoadCamData.....	160
MB_MagicBeltType01.....	223
MB_MasterToPhi.....	401
MB_PhiToXvirt.....	399
MB_PhiToYvirt.....	416
MB_PilgrimStepCalcType01.....	280
MB_PositionMonitoring_GantryCant.....	216
MB_PositionMonitoring_PosCollision.....	218
MB_PrepareCams.....	200
MB_PrintLengthCorrCalcType01.....	290
MB_PUSH_OUT_CONFIG.....	543
MB_ReachVelocityType02.....	212
Fehlercodes	215
Min-/Max- und Defaultwerte	213
Schnittstelle	212
MB_ReadCyclicParameter.....	135
MB_ReadCyclicRealParameter.....	136
MB_RegiMeasuringType01.....	342
MB_RegiMeasuringType02.....	344
MB_RegiMeasuringType03.....	348
MB_RegiRegulateType01.....	352
MB_RegiRegulateType02.....	356
MB_RegiRegulateType03.....	366
MB_RegiRegulateType04.....	372
MB_RegiSetpointLockType01.....	381
MB_RegisterControllerSideType01.....	329
MB_RegisterControllerType02.....	298
MB_RegisterControllerType04.....	308
MB_RESOLUTION.....	542
Elemente	542
MB_SmartBeltType01.....	256
Einführung in die SmartBelt-Verwendung ..	263
MB_TENSION_CONTROL_ADJUST	
Strukturbeschreibung	596
MB_TENSION_CONTROL_CONFIG	
Strukturbeschreibung	596
MB_TENSION_PRE_CTRL	
Aufbau - Aufzählungstyp	597
MB_TensionControlLoadCellType01.....	175
Adaptive	182
DisableCyclicWrites	182
Feedback	183
Fehlercodes	186
Getriebefeinabgleich	184
HighLimit	183
HighLimitAct	183
Inbetriebnahme	181
LowLimit	183
LowLimitAct	183
Min-/Max- und Defaultwerte	178
NegPolarity	183
Pause	183
PControlLowVel	183
Preset / PresetValid	181, 600

M

...MB_TensionControlLoadCellType01	
PresetValue	183
Schnittstelle	176
SetpointLock	182
StandstillEnable	184, 603
StandstillVel	184
Stillstand	185
WindowIn	183, 603
WindowMax	183, 603
WindowMaxValue	183, 603
WindowMin	184, 603
WindowMinValue	184, 603
WindowValue	183, 603
MB_TensionControlLoadCellType02.....	591
Adaptive	601
Adjust	600
Config	599
DisableCyclicWrites	601
Eingangsverhalten	596
Feedback	603
Getriebefeinabgleich	604
HighLimit	603
HighLimitAct	603
Inbetriebnahme	599
LowLimit	603
LowLimitAct	603
NegPolarity	602
Pause	602
PresetValue	603
SetpointLock	602
Standstill	605
StandstillVel	604
MB_TensionDecouplingNetworkType01.....	607
DisableCyclicWrites	617
Enable	616
Fehlercodes	617
Inbetriebnahme	616
MB_TENSION_DECOUPLING_CON-	
TROLLER_VALUE	608
Min-/Max- und Defaultwerte	610
Schnittstelle	608
StandstillEnable	616
Strukturbeschreibung	610
UnControlledAxis	617
MB_TouchProbe.....	109
MB_TouchProbeContinuous.....	112
MB_UnwindMaterialType02.....	497, 500, 503
Preset der Materialgrößen	503
Reset des Materialrechners	503
MB_WIND_CTRL_TYPE.....	465
MB_WIND_DANCER_LIMITS.....	431, 455
MB_WIND_DIAM_CALC_MODE	486
MB_WIND_DIAMETER.....	453, 466, 485
MB_WIND_DIRECTION.....	466, 484
MB_WIND_FRICTION.....	468, 487
MB_WIND_TENS_DIAG.....	468, 487

Index

M

MB_WIND_TENSION.....	467, 486
MB_WIND_TENSION_MODE.....	466
MB_WIND_TORQUE_LIMITS.....	485
MB_WinderDancerCtrlType01.....	450
MB_WinderTensionCtrlSpeedType01.....	481
MB_WindSpeedControlAdaptionType01. .	497, 508
MB_WindTaperProfileType01.....	497, 505
MB_WriteCyclicParameter.....	136
MB_WriteCyclicRealParameter.....	137
MB_WriteExpectWindow.....	115
MB_XvirtToXmech.....	401
MB_YvirtToPhi.....	418
MB_YvirtToPhiPoly5.....	420
MB_YVirtToWorkRange.....	424
MB_YvirtToYmech.....	422
MC_CAMSWITCH_REF.....	54
MC_DigitalCamSwitch.....	56
MC_OUTPUT_REF.....	55
MC_TRACK_REF.....	55
Messtaster	
Fehlercodes des MB_AbortTrigger	120
Fehlercodes des MB_InitTouchProbe	109
Fehlercodes des MB_TouchProbe	112
Fehlercodes des MB_TouchProbeConti- nuous	115
Fehlercodes des MB_WriteExpectWindow	118
Komponenten	92
Parametrierung	93
Überblick	91
Messtasterdatentypen	
MB_DIFF_EVAL	121
MB_EXPECT_WINDOW_MODE	120
MB_POSITION_EVAL	121
MB_PROBE_DATA	121
MB_PROBE_EXECUTION_MODE	120
TOUCHPROBE_REF	122
Methode von Ziegler und Nichols.....	436
Mittelwertbildung	
Materialstärke	503
ML_AxisBackup.....	141
ML_AxisRestore.....	144
ML_BACKUP_OPTIONS.....	146
ML_BACKUP_PARAMS.....	146
ML_BASIC_CAMSWITCH_REF.....	54
ML_BASIC_TRACK_REF.....	55
ML_DegToInc.....	122
ML_DegToRad.....	123
ML_FlyingShear.....	166
Besonderheiten der IndraMotion MLC	173
ML_IncToDeg.....	123
ML_IncToRad.....	124
ML_InterpolationLinear.....	51
ML_MaxValue.....	53
ML_RadToDeg.....	125
ML_RadToInc.....	125
ML_SagControlA.....	192

M

ML_SagControlC.....	188
ML_TechBase	
Einführung und Übersicht	33
ML_TechBase.library.....	33
ML_TechCrank.library.....	387
ML_TechMotion.library.....	165
ML_TechRegi/MX_TechRegi.....	297
ML_TechTensionAdvanced.library.....	591
ML_TwoPosCtrlType01.....	49
Modulo.....	152
Momentengrenzwertüberwachungen.....	480, 496
MX_FlyingShear.....	166
MX_TechWinder.....	425
MX_TechWinder.lib.....	425
MX(L)_FlyingShear.....	167
Fehlercodes	171
Signal-Zeit-Diagramm	170

N

Nachstellzeit.....	433, 436
nicht-bestimmungsgemäßer Gebrauch.....	18
Folgen, Haftungsausschluss	17

P

Parameterübersicht.....	512
Achsparemeter	512
MLC-Achsparemeter	512
P-Parameter	513
Produktspezifisch	513, 514
Produktspezifische Parameter für den zyklischen Kanal der Steuerung	
IndraMotion MLD	514
Standardparameter	513
Parametrierung	
Reibmoment	473, 490
Tänzerlagereger	435
Parametrierung, notwendig	
MB_CalcInertiaLimitType02	500
MB_UnwindMaterialType02	504
ML_FlyingShear	173
Parametrierung für Steuerung MLC	
MB_WinderDancerCtrlType01	458
Parametrierung für Steuerung MLD	
MB_DancerControlType03	434
MB_WinderDancerCtrlType01	458
PELV.....	24
PID-Regler.....	427
Preset	
Wickeldurchmesser	440
Presetwert	
Abwickler	474, 491
Aufwickler	474, 491
Proportionalverstärkung.....	435
Prozessregler.....	426

R

Rastbedingung.....	472, 490
RegiMeasuring	
Fehlerbehandlung	351
Funktionsbeschreibung	350
Zeitdiagramm	350
RegiSetpointLock	
Fehlerbehandlung	382
Funktionsbeschreibung	382
Zeitdiagramm	382
Registerregelungs-, Mess- und Stellfunktionsbausteine.....	342
Registerregler	
Anwendungsbeispiel Insetter-Regelung	326
Anwendungsbeispiel Schlauchbeutelmaschine	327
Beispiel Stanzen beim Etikettendruck	326
Registerregler-FB.....	298
Registersollwertübernahme-FB.....	380
Regler	
D-Anteil	427
I-Anteil	427
P-Anteil	427
Reglerkonfigurationen.....	427
Reglerparametrierung nach Ziegler und Nichols.....	436
Reibkompensation.....	473, 490
Reibmomentkompensation.....	473, 490
RIL_ParameterChannel.....	567
RMB_TechCam.library.....	545
RMB_TechCrosscutCrossseal.....	515
Common - Application FBs	515
RMB_TechWinder.....	425
RMB_TechWinder/MX_TechWinder	
Durchmesserberechnung	436
Ergänzende Funktionsbausteine	497
Funktionsbaustein für Wickler mit Zugkraftregelung und Drehzahlkorrektur	481
Parameterübersicht	512
Tänzerlageregelung	426
Wickelrechner mit Tänzer	449
Wickelrechner mit Zugkraftregelung	460
Wickelrechner mit Zugkraftsteuerung	460
RMB_TechWinder.compiled-library.....	425
RMB_TechWinder.library/MX_TechWinder.lib..	425
Rollenwechsel.....	474, 491
MB_WinderDancerCtrlType01	456

S

Schnittstelle	167
MB_CAM_TABLE_DATA	540
MB_PUSH_OUT_CONFIG	543
MB_TensionControlLoadCellType02	594
ML_SagControlA	194
ML_SagControlC	189
Schutzkleinspannung.....	24
Seitenregisterregler-FB.....	329

S

Sicherheitshinweise.....	14
Sicherheitshinweise für elektrische Antriebe und Steuerungen.....	19
Signal-Zeit-Diagramm	
MB_CalcInertiaLimitType02	499
MB_DancerControlType03	432
MB_DiameterCalculatorType03	440
MB_DiameterMeasurementType01	448
MB_UnwindMaterialType02	502
MB_WinderTensionCtrlType01	468
SmartBelt	
Abfolgegrad	265
Anforderungen an Band und Maschine	267
Antriebsanforderungen	268
Antriebsparametereinstellungen	269
Berechnung des "ProbeSetpoint"	274
Berechnung des "SmartBeltPitch"	273
Projektplanung	267
Reibung	266
SmartBelt-Beschränkungen und -Verhalten	275
SPS-Programmierung	271
Stauförderer	264
Tipps und Tricks	277
Verbreitete Fehler und Missverständnisse ..	278
Software, notwendig	
MB_CalcInertiaLimitType02	500
MB_DancerControlType03	434
MB_DiameterCalculatorType03	443
MB_DiameterMeasurementType01	449
MB_UnwindMaterialType02	504
MB_WinderDancerCtrlType01	458
MB_WinderTensionCtrlType01	476
MB_WindSpeedControlAdaptionType01	512
MB_WindTaperProfileType01	508
ML_FlyingShear	173
ML_SagControlA	195
ML_SagControlC	191
Spanndrehzahl.....	472
Spannen	
des Wickelgutes	472, 490
mit konstanter Bahngeschwindigkeit	472
mit konstanter Drehzahl	472
Spannen mit konstanter Bahngeschwindigkeit.	473
SPS-Projekt in IndraLogic.....	442
Support	
siehe Service-Hotline	619
Symbole.....	14

T

Taktsynchrones Ein- und Auskopplung.....	196
Tänzerkonstante ermitteln	
MB_WinderDancerCtrlType01	455
Tänzerlage.....	426
Tänzerlageregelung.....	433
MB_WinderDancerCtrlType01	456

Index

T

Tänzerlagereger	
Funktionsdiagramm	427
Inbetriebnahme	435
Parametrierung	435
Stellgrößenbegrenzung	432
Tänzerregelung.....	426
Temperaturregler.....	71
Anwendung	78
Identifikation	78
Trägheitsmoment	437, 442
Trigonometrische Funktionen.....	122

U

Übersicht Bibliotheken und Zielsysteme.....	31
Übersicht Technologie-Bibliotheken.....	31

V

Verkettete Liste.....	147
Verwendete Symbole.....	14
Vorhaltezeit.....	433, 436

W

Wichtige Gebrauchshinweise.....	17
Wickelachse.....	449
Wickelantrieb.....	437
Wickeldurchmesser.....	436, 472, 489
Wickelrechner	
mit Kraftmessdose	461
mit Tänzer	435
mit Zugkraftregelung	461
mit Zugkraftsteuerung	460
Rastbedingung	472, 490
sensorlos	460
Wickelrechner mit Tänzer.....	449
Durchmesserrechner parametrieren	459
Erster Anlauf der Wicklerachse kommandieren	460
Funktionsdiagramm	457
Inbetriebnahme	459
Tänzerlagereger parametrieren	459
Wicklerachse kommandieren	459
Wickelrechner mit Zugkraftregelung.....	460, 470

W

...Wickelrechner mit Zugkraftregelung	
Inbetriebnahme	476
Wickelrechner mit Zugkraftsteuerung.....	460, 469
Beschleunigungsmoment kompensieren	469, 470
Durchmesserabhängige Größen berechnen	471
Inbetriebnahme	476
Laufzugkraft parametrieren	469, 470
PI-Regler für die Zugspannung	470
Reibkompensation, drehzahlabhängig	469, 470
Stillstandszug parametrieren	469, 470
Wickelgut spannen	469, 470
Wicklerrechner mit Zugkraftregelung	
Drehrichtung der Wickelachse festlegen	477, 494
minimalen Reibmoment ermitteln	479, 495
Reibmoment bei maximaler Drehzahl ermitteln	478, 495
Reibmoment im Stillstand ermitteln	478, 494
Wicklerachse kommandieren	477, 494
Wicklertyp festlegen	477, 494
Wicklerrechner mit Zugkraftsteuerung	
Drehrichtung der Wickelachse festlegen	477, 494
minimalen Reibmoment ermitteln	479, 495
Reibmoment bei maximaler Drehzahl ermitteln	478, 495
Reibmoment im Stillstand ermitteln	478, 494
Wicklerachse kommandieren	477, 494
Wicklertyp festlegen	477, 494
Z	
Zentrumswickler.....	461
Zu dieser Dokumentation.....	9
Darstellung von Informationen	14
Gültigkeit der Dokumentation	9
Zugkraft.....	472, 489
Zugspannung.....	426
Zugspannungsregler.....	175
Zweipunktregler.....	49
Zyklische Datenkanäle.....	126

Notizen

Bosch Rexroth AG
Electric Drives and Controls
Postfach 13 57
97803 Lohr, Deutschland
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Deutschland
Tel. +49 (0)93 52-40-0
Fax +49 (0)93 52-40-48 85
www.boschrexroth.com/electrics



R911333867